

Dissertação

Re _ Crowdplay

Design de uma aplicação web de testes e otimização de jogos

Elisabete Gonçalves Simões

Mestrado em Design e Multimédia

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

2016 / 2017

Dissertação

Re _ Crowdplay

Design de uma aplicação web de testes e otimização de jogos

Elisabete Gonçalves Simões

Mestrado em Design e Multimédia

Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Orientador

Rui Alexandre Neves Craveirinha

Co-Orientador

Jorge Carlos dos Santos Cardoso

Júri Arguente

Nuno Miguel Cabral Carreira Coelho

Júri Vogal

João Paulo da Silva Machado Vilela

2016 / 2017



Agradecimentos

Quero começar por agradecer aos meus orientadores, em especial ao Professor Rui Craveirinha, pela orientação, apoio e incentivo constantes para a concretização desta dissertação.

Aos docentes do curso, um obrigada pelos conhecimentos transmitidos e aprendizagem que proporcionaram ao longo destes anos de estudo.

Agradeço a todos aqueles que de alguma forma contribuíram para esta dissertação, nomeadamente aqueles que testaram o meu trabalho.

Obrigada a todos os meus amigos, a todos aqueles com quem partilhei bons momentos e a todos aqueles que fizeram parte deste percurso académico.

Um obrigada especial à Mariana, pelo o que aprendemos e evoluímos juntas e por ser a melhor colega de curso; e à Adriana, pelos conselhos, incentivos, motivação e tudo mais.

Aos meus colegas do “lab”, Pedro e João, obrigada pelo companheirismo e por todo o apoio, sem deixar de frisar os nossos momentos de diversão. Ao grupo 8-Bit Forgers, quero agradecer pelo desafio e pela experiência no mundo dos jogos.

Por fim, e não menos importante, quero agradecer à minha família. Ao meu Pai, obrigada por todo o apoio e por me proporcionares as melhores oportunidades para “desanuviar” e fugir à rotina.

Um obrigada muito especial à minha Mãe, simplesmente por tudo; um obrigada não chega para agradecer tudo aquilo que fazes por mim.

Obrigada.

Resumo

Esta dissertação descreve o processo de design de uma interface web para uma aplicação de testes e otimização de jogos, o Crowdplay.

A indústria dos jogos tem aumentado consideravelmente e o extenso processo de desenvolvimento e avaliação são grandes desafios na prática de design de jogos. Embora haja progressos, no que toca a métodos para avaliar e melhorar os jogos, o problema persiste e foi com este objetivo que surgiu a ferramenta Crowdplay. Esta plataforma web foi desenhada para a experimentação e otimização de artefactos vídeo lúdicos. Permite a evolução semiautomática de protótipos e a coleção e processamento automatizado de métricas e indicadores de experiência.

Na sua versão anterior, a aplicação era extremamente complexa no seu funcionamento, e a interface carecia de um design agradável e de uma boa experiência por parte do utilizador. Uma ferramenta para melhorar o design de experiências tem que, ela própria, oferecer uma experiência agradável, donde resulta o objetivo fulcral desta dissertação de mestrado.

Foi elaborado o design de uma nova interface web para o Crowdplay, com o objetivo de melhorar a ferramenta existente, de modo a torná-la fácil e agradável de utilizar e amplificar a experiência dos utilizadores.

Utilizou-se uma metodologia de design de interação, que envolveu a criação de protótipos de baixa e alta fidelidade, e a elaboração de um protótipo funcional. Este protótipo foi submetido a testes de usabilidade com utilizadores e posteriormente refinado. Desta dissertação resultou um protótipo interativo do Crowdplay preparado para funcionar na web, com um design que proporciona uma melhor experiência ao utilizador.

Palavras-chave:

design de interação; design de interface; experiência de utilizador; UX/UI design; usabilidade; design de jogos; otimização de jogos.

Abstract

This dissertation describes the process of designing the web interface for a game testing and optimization application, Crowdplay.

The video game industry has increased considerably and its extensive development and evaluation process are major challenges in the game design practice. Although there has been progress, in terms of methods to evaluate and improve video games, the problem persists and it was with this goal that the Crowdplay tool was created. This web platform was designed for the experimentation and optimization of video game artifacts. It allows the semiautomatic evolution of prototypes and the automated collection and processing of metrics and experience indicators.

In its previous version, the application was extremely complex in its operation, and the interface lacked a pleasant design, failing to offer a good experience to the user. A tool to improve the design of experiences should offer a pleasant experience, which is the main objective of this master's thesis.

The design of a new web interface for Crowdplay has been developed with the aim of improving the existing tool so as to make it easy and enjoyable to use and improve user experience.

We used an interaction design methodology, which involved the creation of low and high fidelity prototypes, and the design of a functional prototype. This prototype was submitted to usability tests with users and later refined. From this dissertation resulted a web-based interactive prototype of Crowdplay, with a better user experience.

Keywords:

interaction design; interface design; user experience; UI/UX design; usability; game design; game optimization.

Índice

Agradecimentos	v
Resumo	vii
Abstract	ix
Índice	xi
Lista de Figuras	xiii
1. Introdução	3
1.1 Âmbito	4
1.2 Enquadramento	5
1.3 Motivação	5
1.4 Objetivo e Metodologia	6
1.5 Plano de Trabalho	7
1.6 Estrutura do Documento	9
2. Estado de Arte	13
2.1 Crowdplay	15
2.1.1 Geração de Conteúdo Procedimental	19
2.1.2 Design e Desenvolvimento	20
2.1.3 Visualização	21
2.1.4 Aplicações Semelhantes	26
2.2 Design de Interação	37
2.2.1 Design de Interface	42
2.2.2 Usabilidade	44
2.2.3 Avaliação	48
2.3 Análise Transversal do Estado de Arte	51
3. Avaliação Preliminar	55
3.1 Público Alvo	55
3.2 Cenários e Personas	56
3.3 Requisitos e Funcionalidades	57
3.4 Análise da Interface	61
3.5 Casos de uso	69
3.6 Testes de Usabilidade	72

4. Proposta de Design	89
4.1 Proposta	89
4.2 Novas Funcionalidades	97
4.3 Prototipagem de Baixa Fidelidade	99
4.4 Prototipagem de Alta Fidelidade	104
5. Desenvolvimento da Proposta	119
5.1 Software Usado - JustinMind	120
5.2 Protótipo Interativo	123
5.3 Testes de Usabilidade	142
5.4 Alterações	155
6. Considerações Finais	163
6.1 Conclusão	163
6.2 Trabalho Futuro	165
Glossário e Siglas	169
Bibliografia	173
Anexos	177
A - Testes de Usabilidade (Documentação)	179
B - Guião de Tarefas (1 ^{os} Testes de Usabilidade)	181
C - Questionário	183
D - Esboços e Protótipos em Papel	185
E - Grelhas	201
F - Screenshots do Software JustinMind	203
G - Documento de Especificações	207
H - Ecrãs do Protótipo Interativo	221
I - Mapa do Website	239
J - Guião de Tarefas (2 ^{os} Testes de Usabilidade)	243
K - Tutorial	245

Lista de Figuras

Figura 1.1 - Plano de Trabalho previsto.	7
Figura 1.2 - Plano de Trabalho realizado.	8
Figura 2.1 - Diagrama das etapas do processo de funcionamento do Crowdplay.	16
Figura 2.2 - Ecrã inicial - Sessões por Projeto.	22
Figura 2.3 - Ecrã de Eventos por Sessões.	23
Figura 2.4 - Ecrã de Eventos por Sessões com filtragem de 2 categorias.	23
Figura 2.5 - Ecrã de Eventos por Sessões com seleção de 2 sessões de jogo.	24
Figura 2.6 - Ecrã da Visualização Particular das Sessões.	24
Figura 2.7 - Ecrã da Visualização Particular das Sessões com a funcionalidade <i>mouse-over</i> .	25
Figura 2.8 - Ecrã da Visualização Particular das Sessões com uso do <i>slider</i> .	25
Figura 2.9 - <i>Dashboard</i> do <i>Game Analytics</i> .	28
Figura 2.10 - <i>Dashboard</i> do <i>Google Analytics</i> .	29
Figura 2.11 - <i>Dashboard</i> do <i>Unity Analytics</i> .	30
Figura 2.12 - <i>Dashboard</i> do <i>DeltaDNA</i> .	31
Figura 2.13 - <i>Dashboard</i> do <i>Swrve</i> .	32
Figura 2.14 - <i>Dashboard</i> do <i>Upsight</i> .	33
Figura 2.15 - Análise comparativa de funcionalidades das aplicações semelhantes.	35
Figura 2.16 - Disciplinas de Experiência de Utilizador.	38
Figura 3.1 - Diagrama UED.	60
Figura 3.2 - Ecrã <i>Design Problem</i> .	61
Figura 3.3 - Ecrã <i>Design Problem</i> com a secção <i>Game Variation</i> expandida.	62
Figura 3.4 - <i>Popup</i> para seleccionar a variável de jogo.	63
Figura 3.5 - Ecrã <i>Design Problem</i> com a secção <i>Design Goal</i> expandida.	64
Figura 3.6 - <i>Popup</i> para seleccionar a fórmula.	65
Figura 3.7 - <i>Popup</i> para adicionar/editar fórmula.	65
Figura 3.8 - Ecrã <i>Test Setup</i> .	65
Figura 3.9 - Ecrã <i>Results</i> .	66
Figura 3.10 - Ecrã <i>Results</i> com uma vista.	67
Figura 3.11 - Ecrã <i>Results</i> com duas vistas.	67
Figura 3.12 - Ecrã <i>New Experiment</i> .	68
Figura 3.13 - Diagrama de Casos de Usos.	70
Figura 3.14 - Tabela de utilizadores (<i>testers</i>).	76

Figura 3.15 - Gráfico do tempo por tarefa.	77
Figura 3.16 - Gráfico da dificuldade por tarefa.	77
Figura 3.17 - Respostas ao questionário.	84
Figura 4.1 - Referência Visual 4.	91
Figura 4.2 - Referência Visual 3.	92
Figura 4.3 - Referência Visual 1.	92
Figura 4.4 - Referência Visual 2.	93
Figura 4.5 - Fonte Open Sans.	94
Figura 4.6 - Fonte PT Sans.	94
Figura 4.7 - Fonte Source Sans Pro.	94
Figura 4.8 - Paleta de Cores.	95
Figura 4.9 - <i>Layout</i> Conceptual.	96
Figura 4.10 - Página do caderno de esboços 1.	100
Figura 4.11 - Página do caderno de esboços 2.	101
Figura 4.12 - Página do caderno de esboços 3.	102
Figura 4.13 - Página do caderno de esboços 4.	103
Figura 4.14 - Protótipo do ecrã de <i>New Test</i> .	104
Figura 4.15 - Protótipo do ecrã de <i>Recent Tests</i> .	105
Figura 4.16 - Protótipo do ecrã de <i>Game Variations</i> .	105
Figura 4.17 - Protótipo do ecrã para editar/adicionar uma variação de jogo.	106
Figura 4.18 - Protótipo do ecrã para selecionar a variável de jogo.	106
Figura 4.19 - Protótipo do ecrã de <i>Design Goals</i> .	107
Figura 4.20 - Protótipo do ecrã para editar/adicionar um objetivo de design.	107
Figura 4.21 - Protótipo do ecrã para definir fórmula.	108
Figura 4.22 - Protótipo do ecrã para editar/adicionar fórmula.	108
Figura 4.23 - Protótipo do ecrã para definir teste.	109
Figura 4.24 - Protótipo do ecrã <i>Results</i> .	109
Figura 4.25 - Fonte Source Sans Pro.	110
Figura 4.26 - Paleta de cores usada na plataforma.	111
Figura 4.27 - Imagens relativas a cada tipo de teste.	112
Figura 4.28 - Ícones representativos dos tipos de gráficos.	113
Figura 4.29 - Outros ícones.	113
Figura 4.30 - Grelha.	114
Figura 4.31 - Diagrama UED atualizado.	115
Figura 5.1 - <i>Screenshot</i> do software <i>JustinMind</i> 1.	122
Figura 5.2 - <i>Screenshot</i> do software <i>JustinMind</i> 2.	122
Figura 5.3 - Ecrã de <i>Login</i> .	124
Figura 5.4 - Ecrã de <i>New Test</i> .	125
Figura 5.5 - Ecrã de <i>Recent Tests</i> .	125
Figura 5.6 - Ecrã de <i>Game Variations</i> .	126
Figura 5.7 - Ecrã para editar/adicionar uma variação de jogo.	127
Figura 5.8 - Ecrã para selecionar a variável de jogo.	128

Figura 5.9 - Ecrã de <i>Design Goals</i> .	129
Figura 5.10 - Ecrã para editar/adicionar um objetivo de design.	129
Figura 5.11 - Ecrã para definir fórmula.	130
Figura 5.12 - Ecrã para editar/adicionar fórmula.	130
Figura 5.13 - Ecrã para definir teste.	131
Figura 5.14 - Ecrã com o <i>status</i> do teste.	132
Figura 5.15 - Ecrã de <i>Results</i> para selecionar as variáveis.	133
Figura 5.16 - Ecrã de <i>Results</i> com um gráfico de barras.	133
Figura 5.17 - Ecrã de <i>Results</i> com a vista em grelha.	134
Figura 5.18 - Ecrã de <i>Results</i> com uma tabela e uma área para selecionar as variáveis.	134
Figura 5.19 - Ecrã de <i>Settings</i> relativo a <i>projects (admin)</i> .	135
Figura 5.20 - Ecrã de <i>Settings</i> para editar/adicionar <i>projects (admin)</i> .	136
Figura 5.21 - Ecrã de <i>Settings</i> relativo a <i>users (admin)</i> .	136
Figura 5.22 - Ecrã de <i>Settings</i> para editar/adicionar <i>users (admin)</i> .	137
Figura 5.23 - Ecrã de <i>Settings</i> para editar <i>account (designer)</i> .	137
Figura 5.24 - Ecrã de <i>Settings</i> relativo a <i>projects (designer)</i> .	138
Figura 5.25 - Ecrã de <i>Settings</i> relativo a um projeto (designer).	138
Figura 5.26 - Ecrã de <i>Projects (gamer)</i> .	139
Figura 5.27 - Ecrã <i>Account (gamer)</i> .	139
Figura 5.28 - Tabela de utilizadores (2º testes de usabilidade).	143
Figura 5.29 - Gráfico do tempo por tarefa de todos os utilizadores.	144
Figura 5.30 - Gráfico da dificuldade por tarefa de todos os utilizadores.	144
Figura 5.31 - Gráfico do tempo por tarefa dos utilizadores iniciais.	150
Figura 5.32 - Gráfico do tempo por tarefa dos novos utilizadores.	150
Figura 5.33 - Gráfico da dificuldade por tarefa dos utilizadores iniciais.	151
Figura 5.34 - Gráfico da dificuldade por tarefa dos novos utilizadores.	151
Figura 5.35 - Respostas ao questionário de todos os utilizadores.	152
Figura 5.36 - Respostas ao questionário dos utilizadores iniciais.	152
Figura 5.37 - Respostas ao questionário dos novos utilizadores.	152
Figura 5.38 - Ecrã <i>Test Setup</i> antes das alterações.	156
Figura 5.39 - Ecrã <i>Test Setup</i> com as alterações referentes ao <i>Design Problem</i> .	157
Figura 5.40 - Ecrã <i>Settings - Projects</i> com as alterações referentes às permissões de utilizadores.	157
Figura 5.41 - Ecrã <i>Settings - Users</i> com as alterações referentes às permissões dos projetos.	158
Figura 5.42 - Tutorial.	159

1

- 1.1 Âmbito
- 1.2 Enquadramento
- 1.3 Motivação
- 1.4 Objetivo e Metodologia
- 1.5 Plano de Trabalho
- 1.6 Estrutura do Documento

Introdução

1. Introdução

Assistimos nos dias de hoje a uma grande evolução tecnológica, tanto a nível da internet como a nível de jogos.

A par da evolução tecnológica, é necessário também o desenvolvimento do design implícito, pois a tecnologia só ganha valor acrescentado quando proporciona aos seus utilizadores verdadeiras experiências, em vez de apenas um uso de ferramentas. Para tal, é necessário compreender os utilizadores atuais e desenvolver produtos de acordo com os seus objetivos e desejos.

Vários estudos têm demonstrado que as transações experienciais, isto é, compras de férias, experiências gastronómicas, culturais, musicais entre outras, têm maior impacto positivo nas vidas dos seus consumidores que as compras puramente materiais. Tendo isto em mente, os designers estão a mudar a sua perspetiva, transferindo o foco do design de produto para o design de experiência, criando experiências em vez de meros artefactos. (Hassenzahl, s.d.)

A indústria dos jogos também tem aumentado consideravelmente; o extenso processo de desenvolvimento e avaliação são grandes desafios nesta prática de design de jogos. Os protótipos são repetidamente refinados, testados e ajustados de diferentes maneiras, e embora haja progressos, no que toca a métodos para avaliar e melhorar os jogos, o problema persiste em como tratar e aplicar a informação. Foi com este objetivo que surgiu o Crowdplay, uma plataforma de experimentação e otimização de artefactos vídeo lúdicos que permite a evolução semiautomática de protótipos e a coleção e processamento automatizado de métricas e indicadores de experiência.

Como ferramenta, visa a facilitação do design de experiências interativas no meio vídeo lúdico, transportando o design de jogo de um domínio material para um domínio experiencial. No entanto, no seu estado atual, a aplicação é extremamente complexa no seu funcionamento, e tem uma interface descuidada e ruidosa, o que a torna pouco usável pelos seus utilizadores alvo. Seria irónico que uma ferramenta para design de experiências não oferecesse ela própria uma experiência agradável, pelo que importa resolver a contradição atual. Donde, o objetivo fulcral desta tese de mestrado é precisamente o redesign da interface da aplicação Crowdplay, de forma a torná-la mais funcional e usável. Note-se que este projeto não se pode esgotar num embelezamento da interface, sendo necessária uma reformulação de raiz do seu funcionamento (ainda que mantendo as suas funcionalidades nucleares).

A versão existente da plataforma Crowdplay, implementada num suporte web, serve como base de trabalho. Para melhorar a ferramenta em termos de usabilidade e experiência do utilizador, usou-se uma metodologia de design de interação, organizada em ciclos de prototipagem e avaliação.

Numa primeira fase, foram efetuados testes de usabilidade e avaliação da experiência da interface na sua materialização atual. Seguiu-se uma segunda fase, de design, que objetivou a criação de protótipos de baixa e alta fidelidade. Após isto, desenvolveu-se um protótipo funcional e a respetiva avaliação com testes de usabilidade. Para finalizar, foram incorporadas as correções para conseguir um melhor aperfeiçoamento do design.

1.1 Âmbito

Esta dissertação surge no âmbito do Mestrado em Design e Multimédia, para a sua conclusão, decorrendo no Departamento de Engenharia Informática da Universidade de Coimbra; na sequência de uma proposta do Professor Rui Craveirinha, orientador desta dissertação.

Nesta dissertação enquadra-se a área de web design, ou design para suportes web, adequando-se assim ao contexto do mestrado e licenciatura deste mesmo curso, que resulta do cruzamento entre design e tecnologia.

Design de interação está relacionado com a interação entre o utilizador e o dispositivo. Tem o objetivo de facilitar as ações que queremos usufruir com qualquer sistema, sendo que o produto deve ser intuitivo, fácil de aprender, tornando a tecnologia cativante e agradável de interagir. Ora tal é precisamente o problema que se quer resolver no que concerne a este projeto.

1.2 Enquadramento

O Crowdplay é uma aplicação web desenhada para a experimentação e otimização de videojogos, implementada pelo aluno João Soares durante a dissertação de mestrado em Engenharia Informática em 2015 no mesmo departamento. Devido à sua complexidade e ao design fraco ao nível da interface e interação, esta foi alvo de um novo design seguindo uma metodologia de design de interação.

Esta é uma ferramenta para os designers de jogos, que analisa, avalia e testa os seus projetos, e ainda os otimiza segundo uma evolução semiautomática. Permite ainda a coleção e processamento automatizado de métricas e indicadores de experiência.

No decurso desta dissertação, foi elaborado o design e implementação de uma nova interface web para o Crowdplay, pretendendo melhorar a ferramenta, de forma a torná-la fácil e agradável de utilizar e amplificar a experiência dos utilizadores; permitindo assim aos designers de jogos tirar partido desta ferramenta promissora.

1.3 Motivação

Como já referido, a aplicação carece de uma interface adequada e tem um funcionamento complexo; resolver tais problemas constitui a motivação principal para este trabalho. Por ser uma ferramenta promissora e potencialmente útil para os designers de jogos, é necessário ter uma boa interface, fácil de usar para que estes consigam tirar o máximo proveito da ferramenta.

Existem muitas aplicações para analisar jogos, assim como para analisar websites, mas estas focam-se mais noutros aspetos, e não propriamente nos de análise e otimização. Assim, podemos concluir que o Crowdplay é uma ferramenta que traz vantagens inovadoras para o desenvolvimento de jogos.

A nível pessoal constitui um enorme desafio poder contribuir para o desenvolvimento desta plataforma bem como a aquisição e consolidação de conhecimentos, tanto ao longo do curso como ao longo deste processo de dissertação.

1.4 Objetivo e Metodologia

O objetivo principal desta dissertação é desenhar uma nova interface da aplicação web do Crowdplay. Pretende-se no final deste projeto obter um protótipo funcional de toda a interface. Não obstante, não é objetivo deste projeto integrar a mesma interface com a aplicação Crowdplay na sua forma existente, dado que isso requererá trabalho de engenharia informática substancial, algo que se apresenta fora do âmbito desta dissertação.

Para a concretização deste objetivo procedeu-se a uma pesquisa bibliográfica relacionada tanto com a aplicação Crowdplay, como com o domínio de design de interação, crucial para atingir os objetivos delineados para o projeto.

Durante o decurso desta dissertação, foi necessário compreender o funcionamento do Crowdplay, bem como entender a abordagem de geração de conteúdo procedimental, o design e desenvolvimento da ferramenta atual, assim como a aplicação paralela da visualização.

Foi ainda descrito o design de interação, onde damos relevo às áreas de design de interface, usabilidade e avaliação, referindo conceitos e práticas relevantes para a elaboração deste projeto.

Numa segunda fase desenvolveu-se um estudo e análise da ferramenta no seu estado inicial. Este estudo serviu para obter uma melhor compreensão da aplicação em si, bem como para formalizar os seus requisitos e listar os seus principais problemas. Como resultado desta análise, foram delineados testes de usabilidade para verificar, não apenas empiricamente, esses mesmos problemas, e levantar quaisquer outros que tenham ficado por analisar.

Durante o processo de desenvolvimento de design, foram efetuados testes de usabilidade e avaliação da experiência da interface na sua materialização atual. Seguiu-se uma segunda fase, de design, que objetivou a criação de protótipos de baixa fidelidade e alta fidelidade. Após isto, desenvolveu-se um protótipo funcional e a respetiva avaliação com testes de usabilidade. Após a avaliação desse novo design, foram incorporadas as correções para conseguir um melhor aperfeiçoamento do design.

1.5 Plano de Trabalho

Para uma melhor gestão e controlo do tempo útil para a elaboração desta dissertação foi elaborado, ainda no decorrer do 1º semestre, um plano de trabalho previsto para o 2º semestre (figura 1.1), tendo em conta a metodologia inicial, bem como as etapas planeadas.

De referir que, embora não tenha sido elaborado um cronograma para o 1º semestre, foi nesse período que foi feita uma pesquisa relativa ao design de interação, e ao Crowdplay e a respetiva avaliação preliminar. Como já foi referido anteriormente, este projeto foi desenvolvido utilizando uma metodologia de design de interação cíclica, composta pelo design e a avaliação.

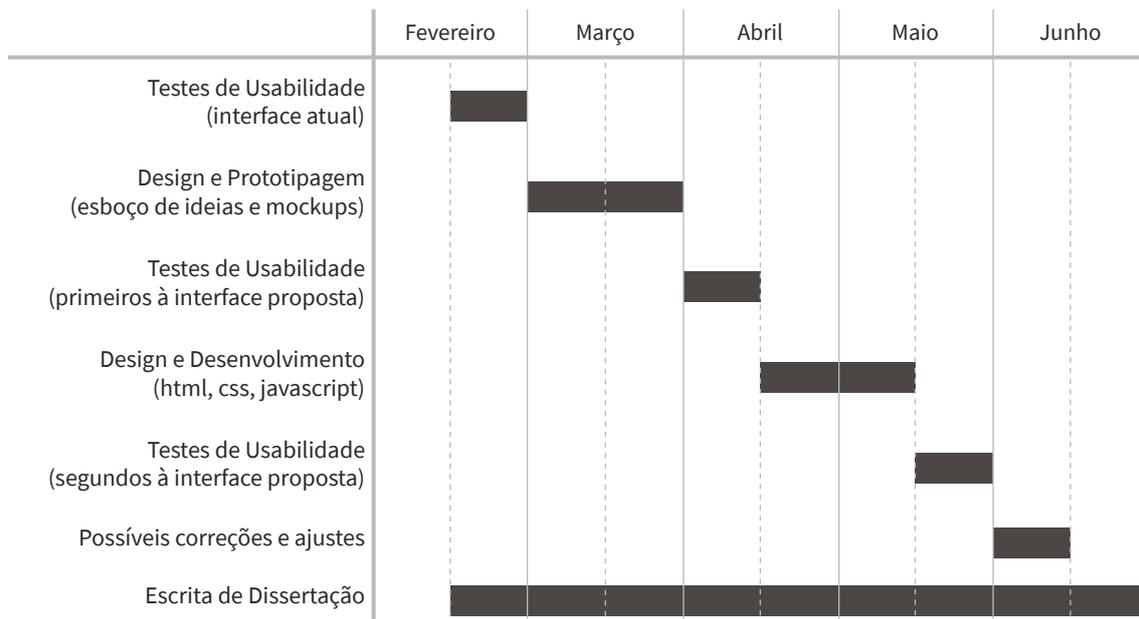


Figura 1.1 - Plano de Trabalho previsto.

Finda a fase de design e prototipagem, fase esta que demorou mais do que o esperado, a metodologia e o consequente plano de trabalho foram reformulados. Foi então elaborado um novo cronograma (figura 1.2) com as devidas alterações e a representação do tempo real.

O trabalho do 2º semestre iniciou-se com a elaboração dos testes de usabilidade à plataforma existente, cuja preparação decorreu ainda no decurso do 1º semestre. Finda esta, iniciou-se a fase de design e prototipagem, na qual foram desenvolvidos desde esboços de ideias, a prototipagem de baixa fidelidade, bem como foi reestruturada toda a plataforma definindo as novas funcionalidades.

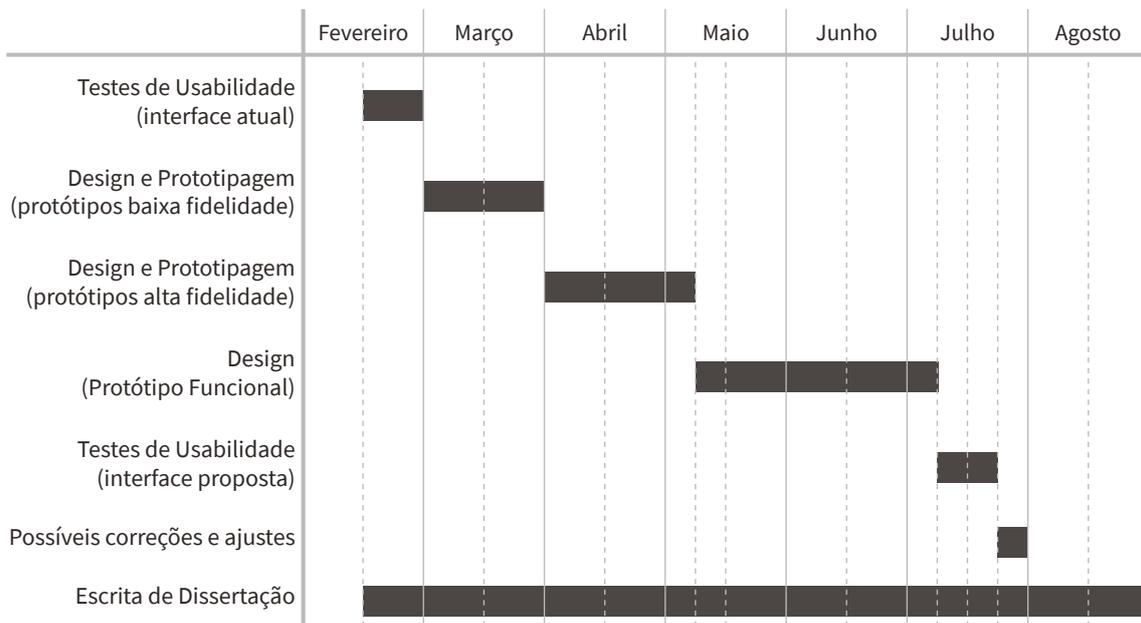


Figura 1.2 - Plano de Trabalho realizado.

Seguiu-se uma fase de desenvolvimento dos protótipos de alta fidelidade, os quais tiveram em conta avaliações breves feitas com recurso às heurísticas do design de interação.

Depois de desenvolver os protótipos de alta fidelidade, era esperado fazer testes de usabilidade e desenvolver a interface funcional com recurso à programação (*html*, *css* e *javascript*). Por entender que não era possível testar a aplicação nos seus aspetos fulcrais, e implementar a plataforma com recurso à programação, decidiu-se alterar a metodologia, para uma melhor gestão do tempo que restava. Optou-se por elaborar protótipos interativos de modo a colmatar esse constrangimento, obtendo-se um protótipo funcional apto a ser testado, mas também uma alternativa à programação.

Para o desenvolvimento do protótipo funcional surgiu então o *software JustinMind* (ferramenta de prototipagem de alta fidelidade com interatividade), de modo a elaborar toda a plataforma o mais próxima do real e pronta a ser testada por completo. Desta ferramenta resultou um protótipo interativo preparado a funcionar na web. Esta fase foi demorada, devido, não só à complexidade do *software*, com uma curva de aprendizagem elevada, mas também por se ter desenvolvido a interface na íntegra. De seguida, foram desenvolvidos os testes de usabilidade com os utilizadores e posteriormente realizadas as correções dos problemas e erros encontrados.

Todo este processo é documentado na dissertação; deste modo, a tarefa de escrita da dissertação é contínua.

1.6 Estrutura do Documento

Para uma melhor compreensão, este documento está organizado por cinco capítulos, para além desta **Introdução**.

O segundo capítulo será dedicado ao **Estado de Arte**, na qual abordamos os subcapítulos do Crowdplay e Design de Interação.

Dentro do subcapítulo Crowdplay, referimos o seu funcionamento, assim como a abordagem de geração de conteúdo procedimental que lhe está subjacente, o design e desenvolvimento da ferramenta existente assim como a aplicação paralela da visualização. É ainda analisado um conjunto de aplicações semelhantes.

Já no subcapítulo de Design de Interação, damos relevo às áreas de design de interface, usabilidade e avaliação, referindo conceitos, regras, práticas e heurísticas dos diferentes temas. Podemos encontrar ainda uma análise transversal a este capítulo .

O terceiro capítulo refere-se à **Avaliação Preliminar**, avaliação esta feita à aplicação existente atualmente, na qual referimos o público alvo, criamos cenários e personas, analisamos requisitos e funcionalidades, desenvolvemos os casos de uso, e ainda referimos e analisamos os testes de usabilidade.

No quarto capítulo, **Proposta de Design**, podemos ler sobre o conceito e ideia desenvolvido ainda no decorrer do primeiro semestre. Após isto, são apresentadas as primeiras fases desenvolvidas já durante o segundo semestre, descrevendo as novas funcionalidades e o processo dos protótipos de baixa e alta fidelidade.

O **Desenvolvimento da Proposta** surge no quinto capítulo, no qual é referido o *software* usado, e apresentada a plataforma na íntegra, bem como todo o processo de desenvolvimento. São ainda descritos os testes de usabilidade e as respetivas alterações efetuadas.

O sexto capítulo, refere-se às **Considerações Finais**, onde são apresentadas as principais conclusões, bem como as dificuldades encontradas ao longo deste processo de dissertação.

Por último, serão apresentados **Glossário e Siglas**, bem como alguns **Anexos** considerados relevantes para o contexto desta dissertação.

2

- 2.1 Crowdplay
 - 2.1.1 Geração de Conteúdo Procedimental
 - 2.1.2 Design e Desenvolvimento
 - 2.1.3 Visualização
 - 2.1.4 Aplicações Semelhantes
- 2.2 Design de Interação
 - 2.2.1 Design de Interface
 - 2.2.2 Usabilidade
 - 2.2.3 Avaliação
- 2.3 Análise transversal do Estado de Arte

Estado de Arte

2. Estado de Arte

Neste capítulo vamos abordar os dois principais temas que se inserem no âmbito desta dissertação. O primeiro tema foca-se na plataforma que será a base do nosso trabalho, o Crowdplay; enquanto que o segundo tema se refere à área prática de enquadramento e desenvolvimento do trabalho, o design de interação.

O capítulo do Crowdplay trata de explicar essa plataforma, partindo do seu objetivo, da abordagem que tem como base - a geração de conteúdo procedimental - até chegar à sua forma de funcionamento. É aqui também referenciado o processo de desenvolvimento da aplicação Crowdplay e de uma aplicação de visualização dos dados que foi prototipada com vista à integração com a mesma. Foi ainda realizada uma pesquisa e análise sobre aplicações semelhantes, na busca de referências sobre como organizar e repensar o design da aplicação.

No capítulo de Design de Interação, abordamos três temas: interação humano-computador, o design de interface e usabilidade. É apresentada uma breve contextualização histórica, conceitos e abordagens para compreender esta componente de design e tecnologia.

2.1 Crowdplay

O Crowdplay é uma plataforma desenhada para a experimentação e otimização de objetos videolúdicos, “teoricamente aplicável a uma vasta gama de géneros de jogo, problemas de design e contextos de uso”. (Craveirinha & Roque, 2016) Esta ferramenta foi implementada pelo aluno João Soares durante a sua dissertação de mestrado em Engenharia Informática no ano letivo de 2014/2015, com o título “Crowdplay - Crowdsourcing Gameplay Data”, segundo orientação do Professor Licínio Roque.

A indústria dos jogos apresenta atualmente um desafio nas práticas de design de jogos que é o extenso processo de desenvolvimento e avaliação, no qual os protótipos são repetidamente refinados, testados de várias modos e ajustados. (Craveirinha & Roque, 2016) Embora haja progressos, no que toca a métodos para melhorar os jogos e avaliar a sua jogabilidade, mesmo com toda a informação existente, persiste o problema de como tratar e aplicar a informação melhorando os videojogos. (Soares, 2015)

Tipicamente, o designer de jogos, no processo inicial de design, necessita de experimentar um grande número de protótipos, e avaliá-los em busca de uma determinada experiência de jogador desejada. (Craveirinha & Roque, 2016) A aplicação do Crowdplay contribui para uma solução ao oferecer um método de evolução semiautomática de protótipos, que visa garantir que, findo o processo, uma determinada experiência do jogador é mediada ao mesmo. Como o processo automatiza o redesign e avaliação do videojogo, este será potencialmente mais eficiente que a prática tradicional.

O Crowdplay é uma aplicação que instância a abordagem *Authorial Game Evolution* (AGE), que recorre a métodos de geração de conteúdo procedimental (PCG - *Procedural Content Generation*).

Quanto ao seu funcionamento, a nível da lógica de base é simples, contudo o funcionamento em si é complexo (figura 2.1).

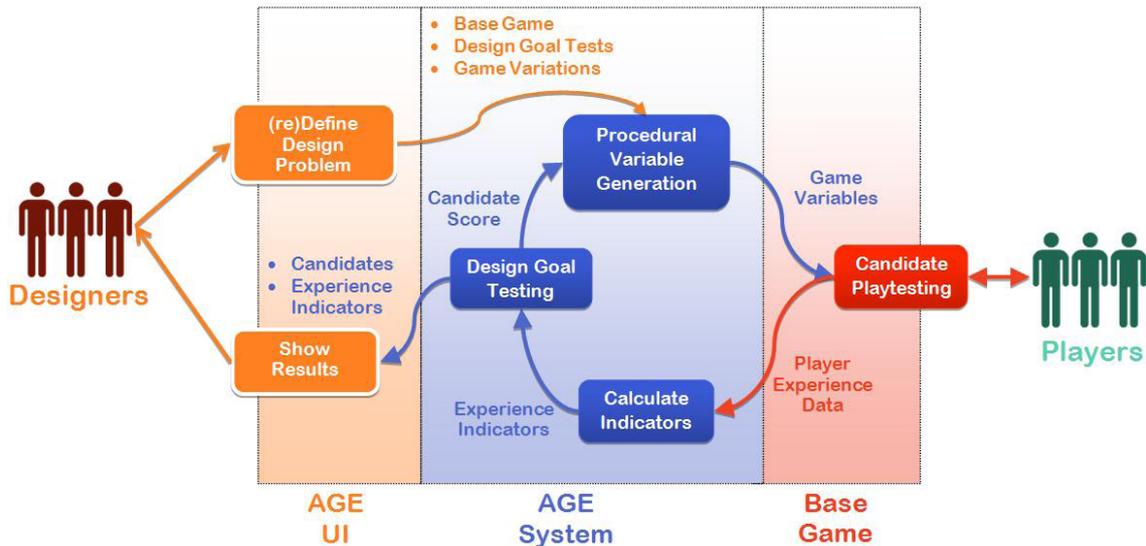


Figura 2.1 - Diagrama das etapas do processo de funcionamento do Crowdplay. (Craveirinha & Roque, 2016)

O processo começa com os designers a partirem de um jogo base, que deve ser previamente integrado com o Crowdplay, para que este possa assim ser evoluído. O designer começa por definir o seu problema de design e um objetivo que pretende resolver. Para exemplificar, imaginemos o jogo do *Super Mario*, na qual temos as “estrelas” para apanhar antes de concluir cada nível. Um objetivo de design hipotético poderia ser “o número ideal de estrelas” que pretendemos que os jogadores consigam apanhar. Para alcançar esse objetivo o designer apenas precisa definir um conjunto de testes (os “Design Goal Tests”) que traduzam esse objetivo.

Ainda no exemplo do *Super Mario*, suponhamos que o ideal seria ter cerca de 30 estrelas colecionadas durante o nível, então para isso define-se um teste que atribui uma pontuação a cada jogo consoante quantas estrelas o jogador coleciona (por exemplo, no intervalo entre 20 e 40), e quanto mais próximo de 30, maior a pontuação. O que isso vai fazer, é levar o motor de geração procedimental a tentar encontrar o jogo candidato que leve os jogadores a colecionar um número de estrelas tão próximo de 30 quanto possível.

Para permitir que o método de geração procedimental gere novos candidatos é, no entanto, necessário definir a forma como ele pode variar o jogo base. Para isso, o designer tem que definir as variações de jogo. No nosso caso hipotético, poder-se-ia definir uma variação para o número de estrelas passíveis de serem colecionadas pelo jogador, e estabelecer que se poderia variar esse valor entre 20 e 50. Tal fará com que o sistema procedimental gere diferentes jogos, cada um com um número de estrelas diferentes, entre 20 e 50, procurando o número ideal para que o jogador coleccione, no decorrer da experiência, 30 estrelas.

Mais detalhadamente, as variações de jogo representam todas as possibilidades de exploração que um algoritmo de geração procedimental pode atingir. Por outras palavras, o designer pode definir um conjunto de variações de jogo, ou seja, o espaço de busca para novos jogos. Cada variação de jogo aplica-se sobre uma variável de jogo, que é um parâmetro do jogo (no exemplo anterior, é o número de estrelas que um nível tem). Quando os jogadores experimentarem o jogo, vão jogar uma variação (do jogo base), consoante as variações predefinidas.

Os objetivos de design são uma materialização dos objetivos de experiência do jogador que os designers pretendem alcançar, sob a forma de testes sobre indicadores de experiência. Um indicador de experiência é um valor calculado a partir de fórmulas com base em diferentes fontes de dados (como métricas de jogabilidade ou respostas a questionários), que serve para avaliar um determinado aspeto da experiência do jogador. Servem de base para a avaliação dos protótipos existentes, medindo o quão próximo está cada um da experiência de jogador pretendida, ou seja, resolvendo o problema de design. Este é um passo crucial para guiar a evolução dos jogos candidatos existentes e tentar assim resolver o problema otimizando as variáveis de jogo.

Há também que estabelecer um conjunto de objetivos de design, que consiste na atribuição de um índice de qualidade (positiva ou negativa), que irá pontuar cada solução de jogo candidata em função do valor dos indicadores de experiência.

Assim que os designers configurem todas as variações de jogo e todos os objetivos de design, o processo de evolução começa. O algoritmo PCG inicia com uma fase de geração de soluções candidatas, gerando novos valores de variáveis de jogo usando um algoritmo de inteligência artificial.

Depois, segue-se uma fase de teste de jogo, em que um conjunto de *playtesters* jogam as soluções candidatas. Os dados dessas sessões de jogo são publicados na ferramenta. Assim que os dados compilados sejam suficientes, são calculados indicadores de experiência que os designers definiram nos objetivos de design.

Voltemos ao exemplo descrito anteriormente (jogo do *Super Mario*) onde foram definidos as variações de jogo e os objetivos de design relativamente às estrelas. Com o algoritmo de PCG geram-se os diferentes jogos, com variações no número de estrelas (entre 20 e 50), e posteriormente, consoante o resultado dos jogadores nos jogos candidatos, é calculado o indicador de experiência “número de estrelas colecionadas”; esse indicador é depois testado: se obtiver 30 estrelas é pontuado com pontuação máxima, se não, mas se se encontrar no intervalo de 20 a 40, é mapeado consoante o valor máximo e mínimo definido.

Cada jogo candidato é avaliado, e o processo repete-se ciclicamente até que atinja a condição definida pelo designer, ou seja, até que uma determinada pontuação ou número máximo de iterações sejam alcançadas. Posto isto, os designers têm acesso a um painel de resultados que permite a visualização de todos os dados da experiência de jogador, em tabelas ou em gráficos.

Durante o desenvolvimento do Crowdplay surgiram novas abordagens para o uso desta plataforma; nomeadamente, a possibilidade de, antes de otimizar os seus jogos, os designers poderem explorar as suas múltiplas variações geradas procedimentalmente. Assim, a ferramenta não serve apenas para a otimização de jogos, mas também para a exploração do espaço de design em redor do jogo base, em busca de configurações que poderão depois ser melhoradas iterativamente. Quanto ao funcionamento, para a exploração é apenas necessário definir as variações de jogo, deixando em aberto os objetivos de design.

Adicionalmente, para além da otimização e da exploração é possível usar o Crowdplay somente para avaliar (ou testar) o jogo, onde apenas se verifica se a experiência dos jogadores vai de encontro ao esperado pelo designer. Nesta vertente o designer avaliará essa experiência pelos resultados em bruto, não havendo a necessidade de definir variações de jogo nem objetivos de design *a priori*.

Com isto, resulta que a utilização do algoritmo de PCG não seja a única adição de valor na ferramenta, mas sim todo o processo de sistematizar o desenvolvimento; exploração, avaliação e teste de protótipos para auxiliar os designers de jogos.

A abordagem centrada no autor para a geração procedimental permite aos designers de jogos, tanto quanto possível, o controle sobre o processo criativo, evitando que se afastem do seu objetivo de design de jogo. (Soares, 2015) Para esse efeito, faz recurso ao processamento automatizado de indicadores de experiência que pode ser usada pelos designers tanto para explorar como para evoluir o seu protótipo. (Craveirinha & Roque, 2016)

2.1.1 Geração de Conteúdo Procedimental

Para melhor compreender o conceito de experimentação, é dada uma explicação sobre as metodologias que estão subjacentes a esta plataforma. O Crowdplay funciona através de uma materialização de métodos de geração de conteúdo procedimental (PCG), especificamente, numa abordagem centrada no autor (ACPCG - *Author-Centric Approach to Procedural Content Generation*). (Soares, 2015)

PCG é um processo no qual um método de algoritmo, de forma automática ou semiautomática, é capaz de gerar conteúdo de videojogo. Definimos conteúdo como qualquer componente do videojogo ou um artefato de videojogo ou ainda um protótipo. (Craveirinha et al 2016)

Um exemplo de PCG usado num jogo é o conhecido *Minecraft*. Neste videojogo a criação do mundo, constituído por cubos texturizados com diferentes materiais, é gerado aleatoriamente de forma parcial. Os cubos são organizados consoante um algoritmo de geração procedimental, criando um mundo praticamente infinito, que difere entre jogos. Os cubos podem assumir diferentes materiais (água, areia, pedra, madeira, carvão, metais ou até pedras preciosas) e criam tipologias de terrenos diferentes (praia, floresta, deserto, entre outros). A geração do mundo é semialeatória, pois há que obedecer a determinadas condições, evitando por exemplo desnivelamentos do terreno (que o tornem impossível de alcançar) ou incoerências na tipologia de terreno (como por exemplo aparecer água no deserto).

A abordagem focada no autor é uma variação de uma metodologia idêntica, focada na experiência do jogador, designada de geração de conteúdo procedimental mediada pela experiência (EDPCG - *Experience-Driven Procedural Content Generation*). (Soares 2015)

Esta metodologia de geração de conteúdo procedimental gera o conteúdo de forma a otimizar a experiência do jogador, usando indicadores recolhidos a partir da interação deste com o jogo, como referência para avaliação. Nesse caso, são as necessidades do jogador que dominam a forma como o conteúdo é gerado.

Quando se trata de jogos, cujo conteúdo é baseado apenas na experiência do jogador, testá-los pode dar origem a resultados indesejados, tais como, alteração da dificuldade do jogo ou dispersão nos componentes mais importantes, levando a uma potencial perda da experiência de jogo. (Soares, 2015)

A principal característica diferenciadora da abordagem centrada no autor é deixar os designers definirem que experiência pretendem que os jogadores obtenham, conseguindo assim ter maior controlo no resultado final. O ACPCG tenta adaptar o jogo consoante o conceito do designer (autor) para a experiência do jogador. Desta forma, a metodologia pode extrair a experiência do jogador mantendo o objetivo alvo do designer. Enquanto com a metodologia EDPCG o jogador pode desviar-se pelo modo de jogo que mais lhe agrada; na metodologia ACPCG é encaminhado para um jogo protótipo consoante a intenção do autor. É nesta última abordagem (ACPCG) que se foca e que consiste a plataforma Crowdplay, e o seu funcionamento base.

2.1.2 Design e Desenvolvimento

A interface desta plataforma surge no decorrer da dissertação do aluno João Soares, e este capítulo serve para descrever o seu processo de desenvolvimento.

O principal objetivo no projeto de tese do aluno João Soares era responder à questão: Como é que a ACPCG pode ajudar os designers de jogos no seu processo de design criativo?

O Crowdplay surge para responder a essa questão, já que para isso, era necessário desenvolver um protótipo da aplicação ACPCG, com uma interface própria e um protótipo de videojogo para trabalhar com a aplicação.

A interface do Crowdplay foi desenhada a partir de protótipos em papel já existentes. Estes protótipos resultaram de um processo de design participativo, desenvolvido pelos (futuros) utilizadores. Estes já estavam familiarizados com o contexto, e poderiam desenhar a interface consoante as suas necessidades e preferências de uso. Depois de desenvolvida a interface do utilizador, foram feitos testes de usabilidade para descobrir falhas no protótipo projetado. Estes testes de usabilidade foram gravados em vídeo e posteriormente analisados, onde se encontraram alguns problemas na interface.

Quanto à terminologia, esta também foi alvo de testes e alterações de modo a facilitar a sua compreensão. Em aplicações complexas, a terminologia deve ser direta e clara, especialmente se a quantidade de termos usados é elevada. Devido à complexidade do algoritmo e da aplicação os termos usados eram, e ainda são, um obstáculo para os potenciais utilizadores desta aplicação. Durante o projeto do aluno João Soares, foi então repensada uma nova terminologia, resultando uma lista de expressões mais intuitivas e mais fáceis de os utilizadores entenderem.

No decorrer do processo de design participativo surgiram duas questões que precisavam de ser resolvidas. Numa primeira fase em que os utilizadores testaram a aplicação, sentiram a necessidade de uma opção para exploração do jogo, antes de o submeter à otimização.

Este foi um problema em parte resolvido, pois é possível no estado atual usar a aplicação para essa tal necessidade de exploração; no entanto essa funcionalidade está apenas subentendida. Pelo referido, considera-se que esta funcionalidade pode ainda ser melhorada de modo a permitir ao utilizador escolher a forma como usa a plataforma (se é para otimização, exploração ou apenas para avaliação).

A outra questão centrava-se no facto de o conceito geral de PCG e a sua utilidade ser difícil de compreender por parte dos designers; apesar de já ter sido alvo de alterações a terminologia carece ainda de alguns ajustes.

Uma vez implementada a aplicação e a sua interface, foi adaptado o código de um jogo (*Dune Legacy*) para integrar e testar com o Crowdplay. Desse processo resultou um caso de estudo onde foi avaliada a experiência de um designer com a aplicação em si. (Craveirinha & Roque, 2016)

Os dados da experiência dos jogadores que resultaram deste teste foram ainda alvo de trabalho noutra tese, que será descrita no capítulo a seguir.

2.1.3 Visualização

No âmbito da dissertação de mestrado em Design e Multimédia do aluno André Cabral, foi desenvolvida uma interface para a visualização dos dados provenientes do Crowdplay.

A proposta era a criação de uma visualização de informação para a plataforma online Crowdplay com o objetivo de auxiliar os designers de jogos.

Esta visualização possibilita a exploração de grandes quantidades de dados com uma representação através da interatividade, permitindo analisar e identificar regularidade ou irregularidade nos dados gerados a partir das ações dos jogadores. (Cabral, 2016)

O método de recolha de dados do Crowdplay gera grandes quantidades de informação, tornando a análise difícil sem uma visualização adequada. Com esta ferramenta adjacente ao Crowdplay é possível visualizar e analisar como os jogadores se comportam dentro dos videojogos testados. Para os cenários em que um designer de jogos quer avaliar ou explorar os resultados é bastante importante a sua representação de forma clara, para que o designer consiga identificar dados pertinentes e obter resultados para melhorar o seu jogo.

Na implementação atual da plataforma Crowdplay, os designers podem explorar e analisar os resultados dos jogos candidatos, usando um conjunto de ferramentas gráficas padrão, podendo visualizar os dados a partir de gráficos como o *pie chart*, gráfico de barras, histogramas e ainda por tabelas. O projeto de visualização de dados não veio substituir os ecrãs que dão essas funcionalidades, tendo sido criada uma ferramenta em *Processing* que trabalha paralelamente ao Crowdplay.

Esta nova ferramenta de visualização tem como objetivo ajudar os designers a identificar dados relevantes, de forma mais fácil e de modo a conseguirem tirar conclusões e ajustar os seus jogos. (Cabral, 2016)

De seguida serão apresentadas as imagens referentes a esta ferramenta de visualização, demonstrando o protótipo funcional e o seu estado de funcionamento.

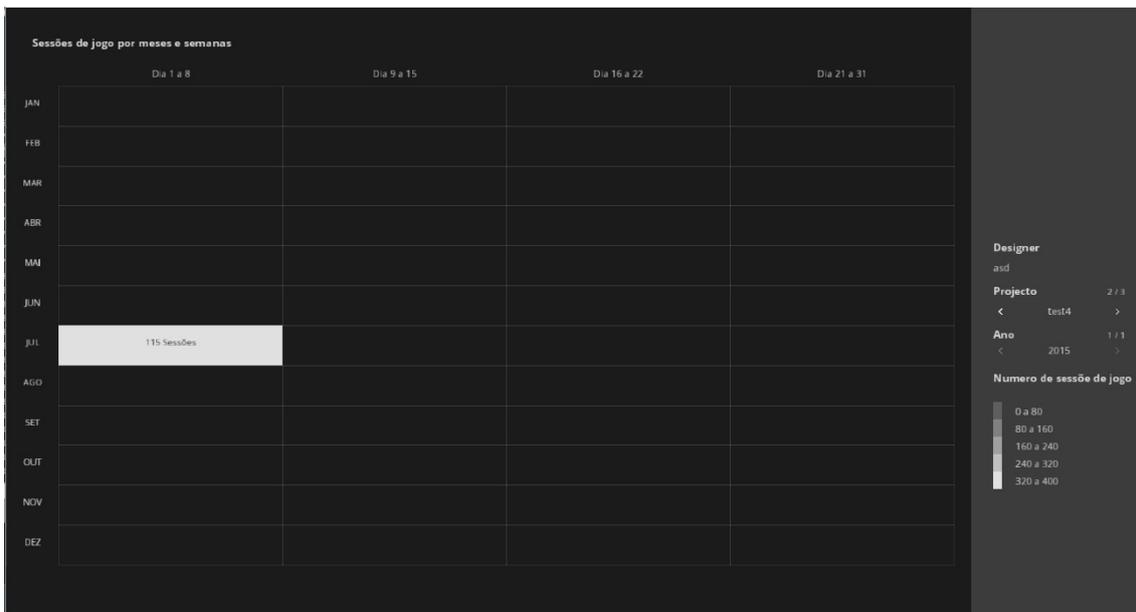


Figura 2.2 - Ecrã inicial - Sessões por Projeto.

Este é o ecrã inicial (figura 2.2), onde é possível selecionar qualquer conjunto de dados (sessões) existentes. Caso um designer tenha mais do que um projeto é possível alterar na barra lateral, e o mesmo acontece para a seleção do ano. Quando existem vários conjuntos de sessões, conseguimos observar quais os que contém mais e menos sessões, com recurso a uma escala (legenda na barra lateral).

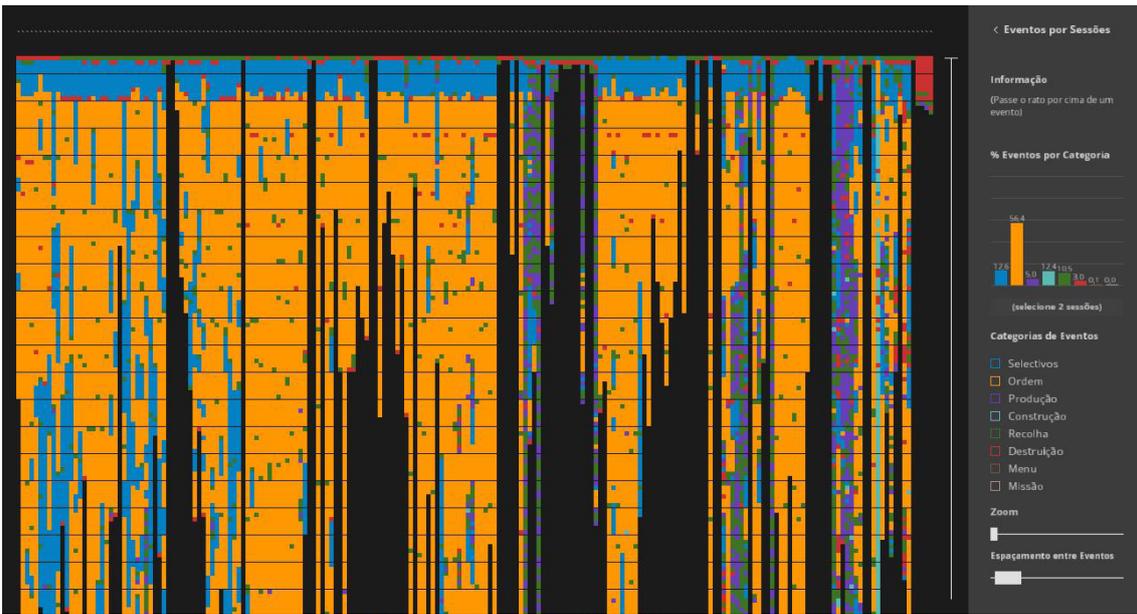


Figura 2.3 - Ecrã de Eventos por Sessões.

Neste ecrã, (figura 2.3), podemos ver as categorias dos eventos jogados, observando a quantidade e sequência das ações do jogador, sendo possível filtrar categoria de eventos (figura 2.4), de maneira a obter um destaque maior dos mesmos. Ainda relativo à interatividade, é permitido fazer um zoom na representação para observar melhor uma zona, e, ao passar o rato sobre um evento, é possível obter informação sobre esse evento (nome do evento e número da sessão). (Cabral, 2016)

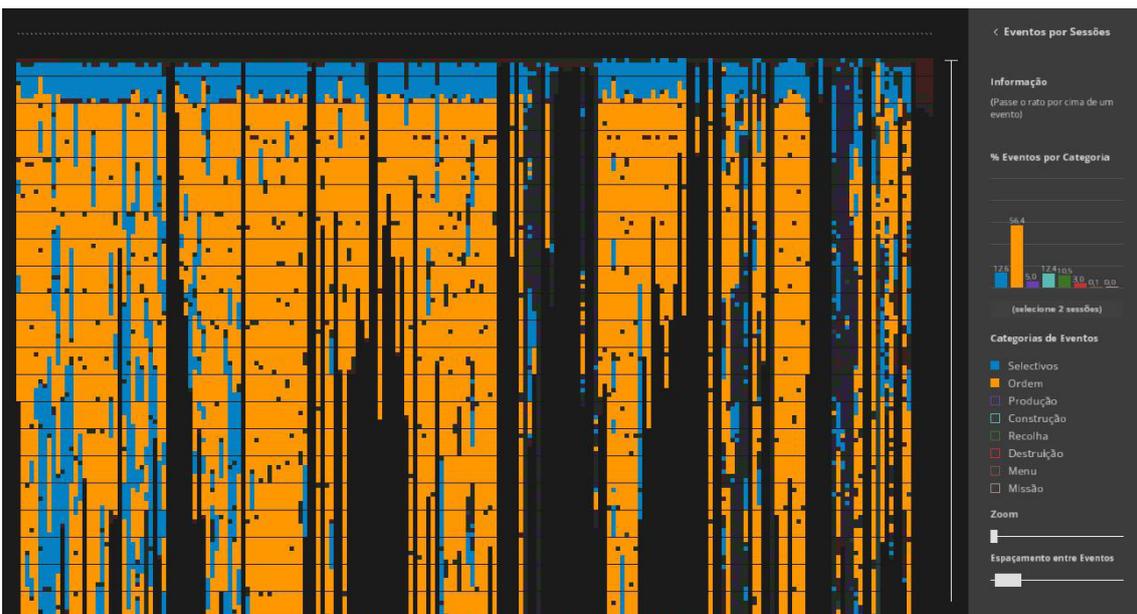


Figura 2.4 - Ecrã de Eventos por Sessões com filtragem de 2 categorias.

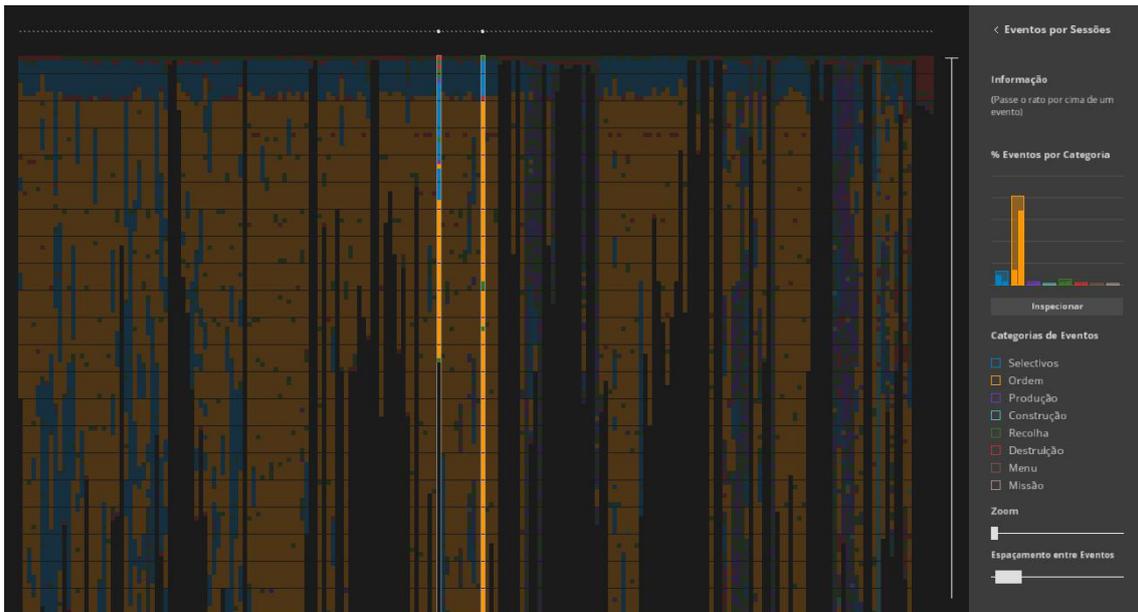


Figura 2.5 - Ecrã de Eventos por Sessões com seleção de 2 sessões de jogo.

Para avançar para o terceiro nível de representação, é necessário selecionar as duas sessões que desejamos observar em detalhe para posterior comparação (figura 2.5).



Figura 2.6 - Ecrã da Visualização Particular das Sessões.

No ecrã do terceiro nível de representação (figura 2.6) é possível comparar duas sessões de jogo e analisá-las. É representada a categoria jogada consoante a sua coordenada, e na parte inferior podemos obter essa informação numa linha temporal. Ao passar o rato por cima de uma dessas coordenadas obtém-se mais informações (podem estar categorias sobrepostas) e interligar a linha temporal, para saber quando é que esse evento ocorreu (figura 2.7). Neste mesmo ecrã podemos ver que também é possível destacar categorias a partir da barra lateral.



Figura 2.7 - Ecrã da Visualização Particular das Sessões com a funcionalidade *mouse-over*.



Figura 2.8 - Ecrã da Visualização Particular das Sessões com uso do *slider*.

Por fim, no ecrã da figura 2.8, é visível o uso do slider para escolher um determinado intervalo na linha temporal, observando em detalhe os dados nesse período de tempo. Também é possível selecionar uma subcategoria, destacando-a no gráfico.

O projeto final da visualização tinha como objetivo ser implementado na plataforma online Crowdplay, mas no âmbito da dissertação do aluno André Cabral, resultou a elaboração de um protótipo com os dados do jogo de estratégia *Dune Legacy*.

Apesar desta visualização funcionar e realmente ajudar os designers de jogos a entender como os *testers* os jogaram, levanta algumas questões. Será que funcionaria noutro tipo de jogo, sem ser estratégia? Como seria adaptado a um jogo que não funcione sobre um mapa ou matriz?

Na segunda página, o ecrã de eventos por sessões (figuras 2.3, 2.4 e 2.5), é adaptável a diferentes géneros de jogos, visto que os eventos podem ser traduzidos em diferentes tipos de categorias, ações ou outro tipo de característica dos jogos. No entanto, a terceira página (figuras 2.6, 2.7 e 2.8) teria de ser reajustada consoante o género de jogo, sendo que em substituição do gráfico da matriz poderíamos ter outro tipo de gráfico consoante os dados.

2.1.4 Aplicações Semelhantes

Existem várias ferramentas que testam e analisam jogos e outros tipos de *softwares* (como websites), no entanto apenas o Crowdplay permite definir e mediar a agenda para a experiência de jogador. Não foi encontrada nenhuma ferramenta com as características de otimização do Crowdplay, com a possibilidade de definir parâmetros como as variações de jogo e objetivos de design, otimizando assim o jogo de modo semiautomático.

As ferramentas semelhantes centram-se apenas na análise e nos testes, devolvendo dados em bruto e em forma de gráficos, com recurso às *dashboards*, para o utilizador analisar e tirar partido disso, melhorando assim os seus projetos. Com isto, as ferramentas analisadas vão de encontro ao uso mais simples da plataforma Crowdplay, na qual apenas avaliamos sem ser necessário definir as variações de jogo.

As ferramentas mais semelhantes destacam a funcionalidade de testes A/B (comparação entre duas opções, ou dois candidatos), o que já se assemelha ao segundo patamar de uso do Crowdplay, a exploração, na qual se definem variações de jogo, podendo estas assumir variadíssimas possibilidades. É de notar que a maioria destas ferramentas parece focar-se na área das aplicações móveis, dando especial ênfase ao marketing e à gestão de anúncios e compras (comprar componentes para os jogos).

Posto isto, de seguida são referenciadas as ferramentas que serão alvo de uma breve análise.

- **Game Analytics** - disponível em www.gameanalytics.com;
- **Google Analytics** - disponível em www.google.com/analytics;
- **Unity Analytics** - disponível em unity3d.com/services/analytics;
- **DeltaDNA** - disponível em deltadna.com;
- **Swrve** - disponível em www.swrve.com;
- **Upsight** - disponível em www.upsight.com.

Game Analytics

O *Game Analytics* é uma ferramenta de análise gratuita, que fornece informações sobre o comportamento do jogador, desde a jogabilidade à gestão de compras, comparando também com o desempenho de outros jogos.

Esta ferramenta funciona partir de um SDK (*Software Development Kit*) em português Kit de Desenvolvimento de Software, na qual é possível integrar vários sistemas operativos (IOS, tvOS e *Android*) ou plataformas de desenvolvimento de jogos (*Unity*, *Unreal* e *Xamarin*). Após a integração do jogo com a ferramenta, é possível, através das *dashboards* (painéis de controlo), a análise com ferramentas de visualização e mineração de dados.

Os dados e eventos suportados por esta plataforma são variados; eventos de compra (para rastrear as transações de dinheiro real); de recursos (para controlar quando os jogadores ganham ou perdem recursos); de progresso (para rastrear quando os jogadores começam e terminam níveis); de erro (para registar erros ou avisos gerados pelo comportamento no jogo); e de design (para rastrear elementos gráficos ou etapas do tutorial). A partir do SDK, e adaptando o código do jogo, é possível recolher métricas e dimensões personalizadas. Esta ferramenta, na última versão, tem a possibilidade de fazer testes A/B, lançando conteúdo personalizado para diferentes grupos.

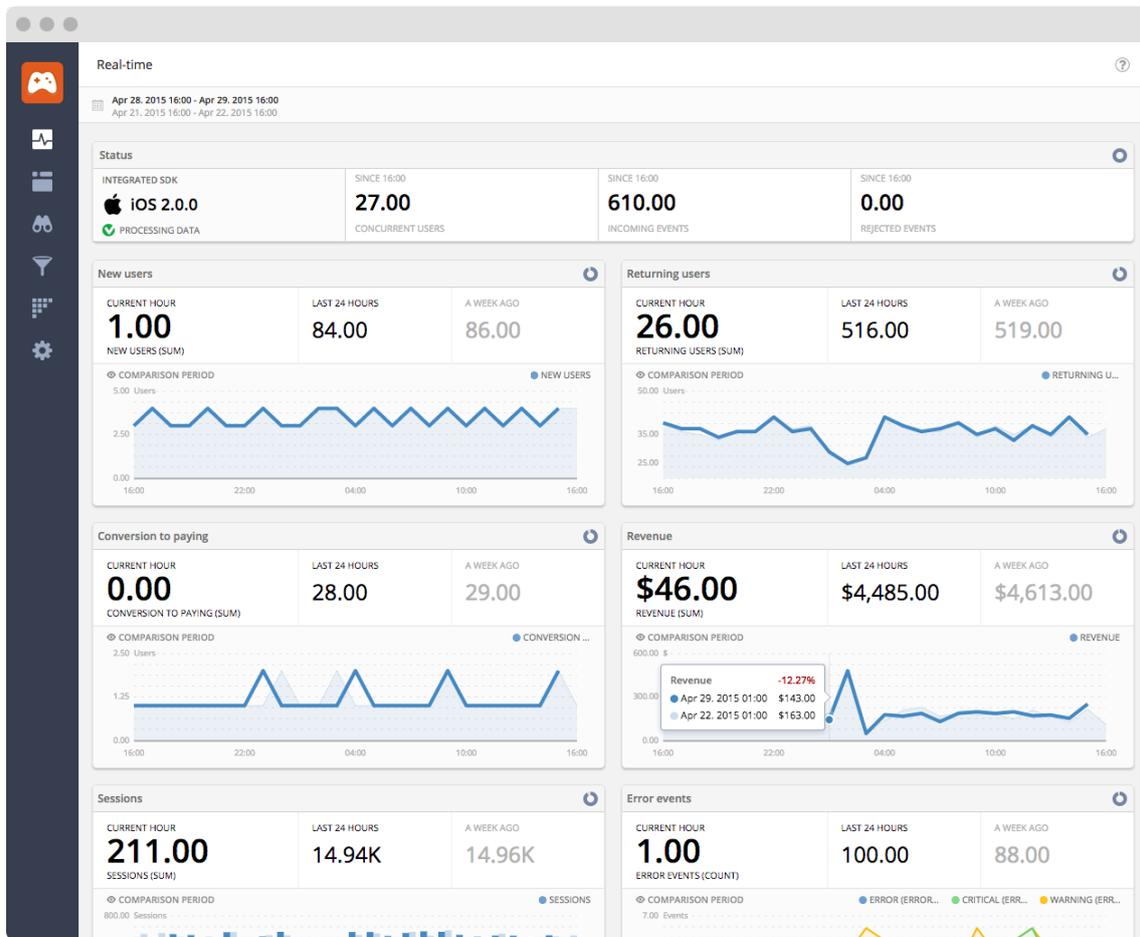


Figura 2.9 - Dashboard do Game Analytics.

As dashboards do Game Analytics (figura 2.9) têm um conjunto de visualizações predefinidas, no entanto estas podem também ser personalizadas de acordo com o objetivo do utilizador, obtendo assim os relatórios e as análises desejadas.

Google Analytics

O *Google Analytics* é uma ferramenta de estatísticas mais vocacionada para websites com diferentes serviços, a qual também oferece um conjunto de ferramentas para estudar o desempenho de jogos. Com o *Google Analytics* é possível analisar o conteúdo dos websites, sabendo com que frequência as pessoas visitam cada página, o tempo que permanecem nela e a frequência com que efetuam ações. Nos serviços do *Google Analytics* encontramos também ferramentas para analisar os websites em dispositivos móveis, relatórios das redes sociais e análise da publicidade.

Relativamente aos jogos, o Google disponibiliza o *Google Play Games Services* (disponível em developers.google.com/games/services/android/stats), um serviço relativo aos jogos em *Android*. A partir da API (*Application Programming Interface*) é possível adaptar as experiências dos jogos em segmentos específicos e em diferentes fases do ciclo de vida do jogador, criando experiências adaptadas para cada conjunto de jogadores com base no seu progresso.

O *Google Analytics* é gratuito e fácil de configurar, suportando métricas personalizadas. Permite alterar a informação das *dashboards* personalizando-as (figura 2.10). Apesar de não ser uma ferramenta de análise própria para jogos, é possível utilizá-la para jogos (em *flash*) na web.

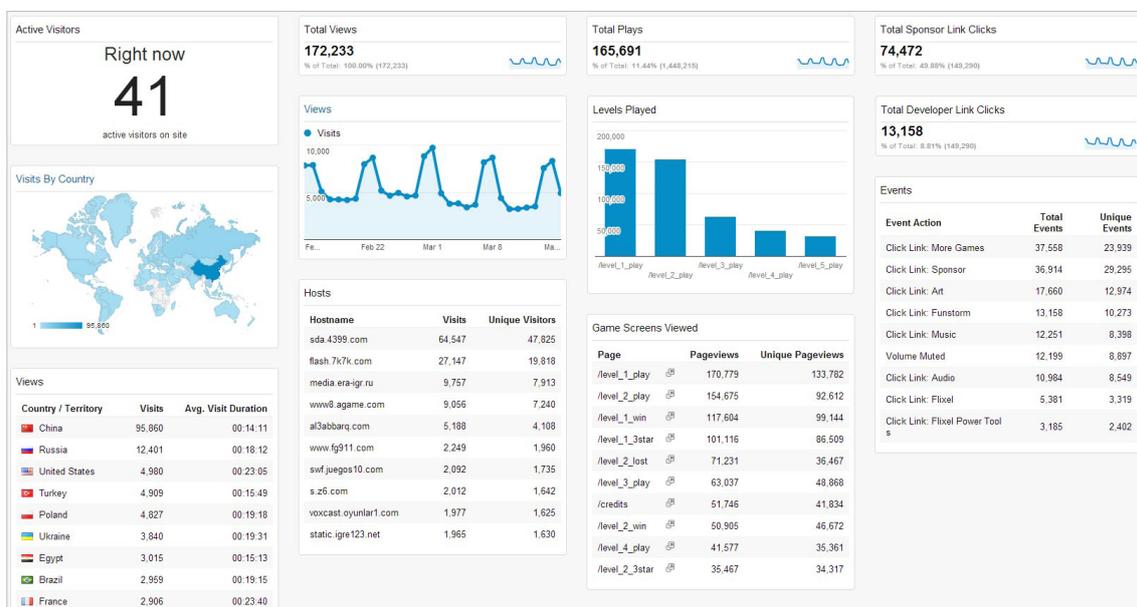


Figura 2.10 - Dashboard do Google Analytics.

Unity Analytics

O *Unity* é uma plataforma de desenvolvimento de jogos e como tal tem a sua própria ferramenta de testes, o *Unity Analytics*.

Dentro do *Unity Analytics* existem vários serviços para ajudar no desenvolvimento, entre eles o *Core Analytics*, dando acesso rápido e fácil a informações importantes que ajudam a melhorar a economia de jogo e a experiência do jogador (figura 2.11). Outra funcionalidade dentro do *Unity Analytics* é o *Heatmaps* (Mapas Térmicos), que se traduzem numa visualização dos eventos espaciais que ocorrem no jogo, criando uma imagem com padrões que refletem as ações. De referir ainda que é possível obter os resultados em tempo real com o *Livestream*, e também exportar os dados em bruto. Por ser uma ferramenta própria do *Unity*, não necessita de SDK, tornando o seu uso mais simples e direto.

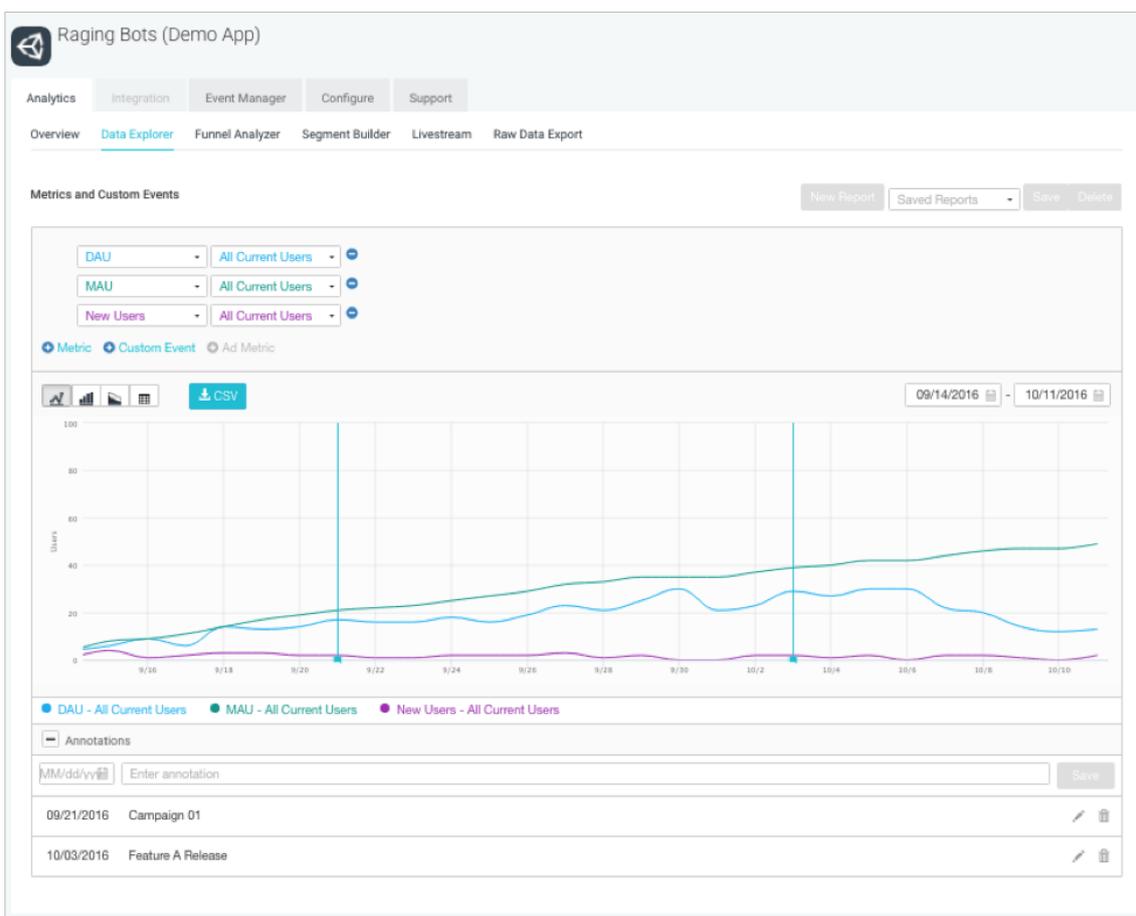


Figura 2.11 - Dashboard do Unity Analytics.

DeltaDNA

Esta ferramenta oferece desde uma análise de dados, consultoria e gestão de marketing até serviços de suporte, tanto a videojogos como para jogos de dinheiro real (jogos de casino). O *DeltaDNA* aposta na profunda análise de dados e numa plataforma de marketing em tempo real.

É possível uma integração rápida do jogo com a plataforma a partir de SDKs oferecendo suporte para implementações nativas para iOS e *Android*, bem como a versão para *Unity* (que apoia iOS, *Android*, *Windows Mobile* e PC). Quando um jogo é adicionado à plataforma *DeltaDNA*, esta tem a capacidade de enviar uma ampla gama de dados de eventos predefinidos e personalizados, assim como também suporta segmentações e mensagens redirecionadas. As *dashboards* personalizadas (figura 2.12) fornecem as informações necessárias num só lugar, podendo configurar e partilhar painéis. Para além da análise dos dados, com esta ferramenta é também possível fazer teste A/B e segmentar os jogos e as mensagens de marketing, redirecionando para um público específico, conseguindo assim uma experiência personalizada a cada tipo jogador.

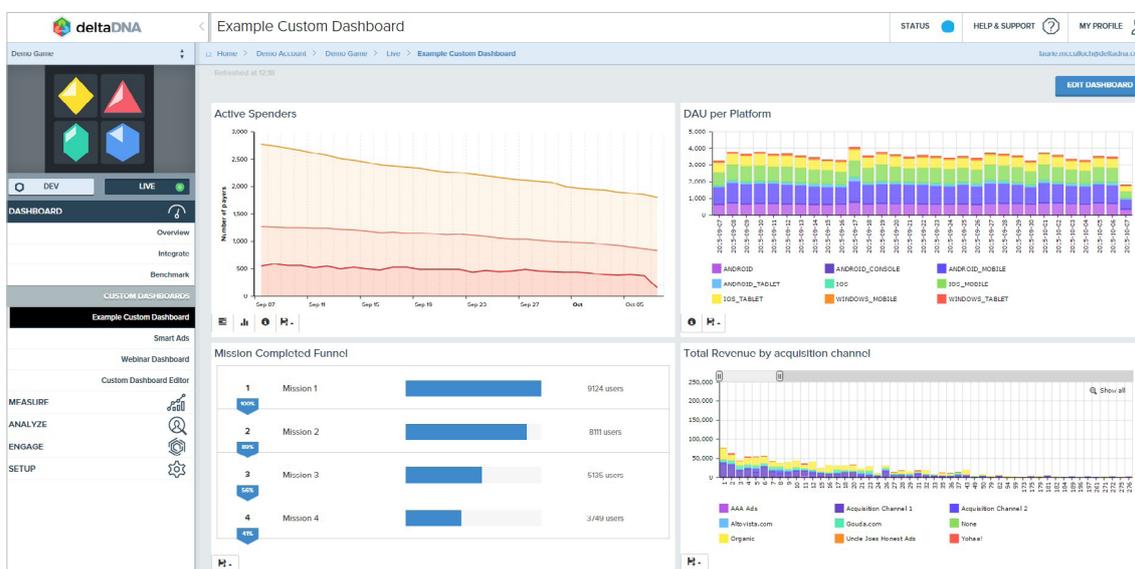


Figura 2.12 - Dashboard do DeltaDNA.

Swrve

O *Swrve* está vocacionado para o marketing em aplicações móveis, nomeadamente jogos. Esta ferramenta tem a possibilidade de segmentação tanto da análise de dados como a segmentação de utilizadores, criando grupos distintos com conteúdo específico.

A plataforma *Swrve* em termos de funcionalidades é muito idêntica à *DeltaDNA* (acima descrita), na qual é ainda possível fazer testes A/B com o conteúdo da própria aplicação ou de mensagens.

Quanto aos jogos, o *Swrve* gere o ciclo de vida do jogador, desde a instalação a uma determinada compra, não avaliando propriamente o modo como os jogadores interagem com o jogo, mas personalizando o conteúdo (publicidade) adequado a cada jogador.

Todas estas ferramentas funcionam com uma interface do tipo de *dashboard* (figura 2.13), no entanto esta não permite a visualização de dados com a mesma potencialidade do que as outras, mostrando simplesmente a informação sem recorrer aos diferentes tipos de gráficos.

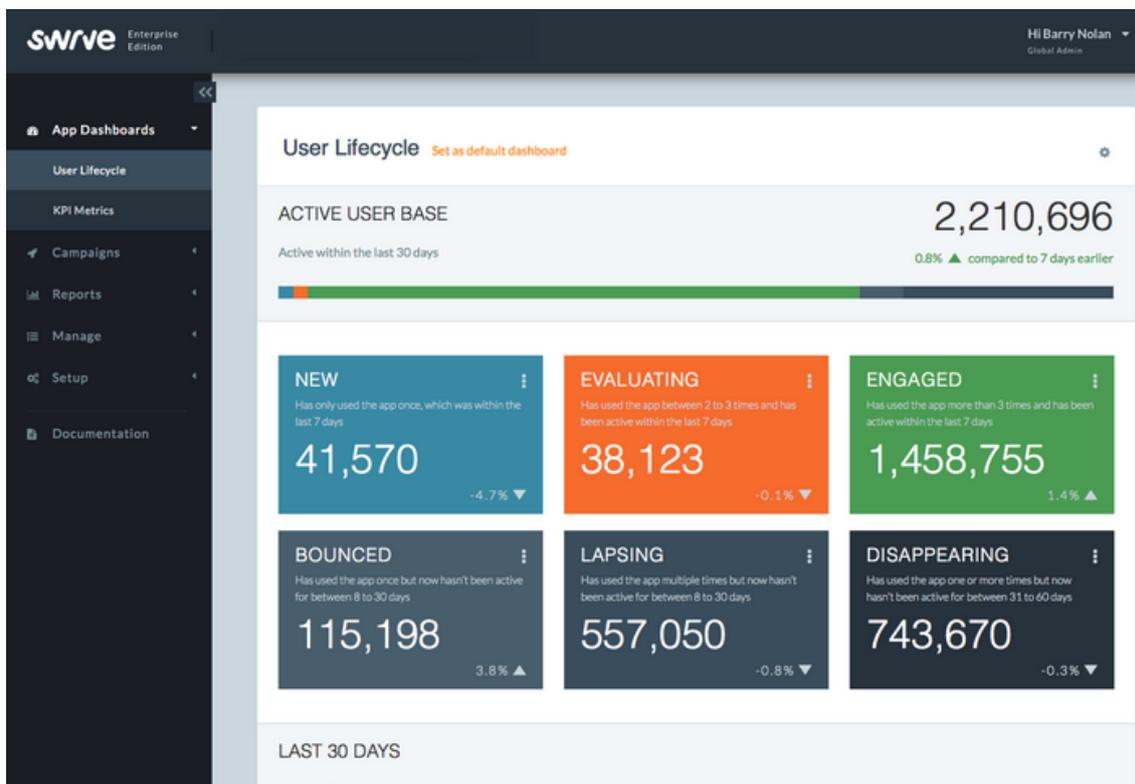


Figura 2.13 - Dashboard do Swrve.

Upsight

A ferramenta *Upsight* direciona-se para a exploração, segmentação e otimização de marketing, sendo bastante idêntica às duas ferramentas descritas anteriormente. Com esta plataforma é possível, a partir de atributos padrão ou personalizados, criar segmentos de utilizadores significativos para as campanhas de marketing ou experiências de utilizador. Esta ferramenta também possibilita testes A/B aos anúncios, assim como a sua otimização; para além disso, esta ferramenta parece possuir mais suporte técnico com acompanhamento no projeto.

A *dashboard* (figura 2.14) potencia a exploração de dados, para que os utilizadores tirem partido dos resultados e de toda a informação com os painéis personalizáveis.

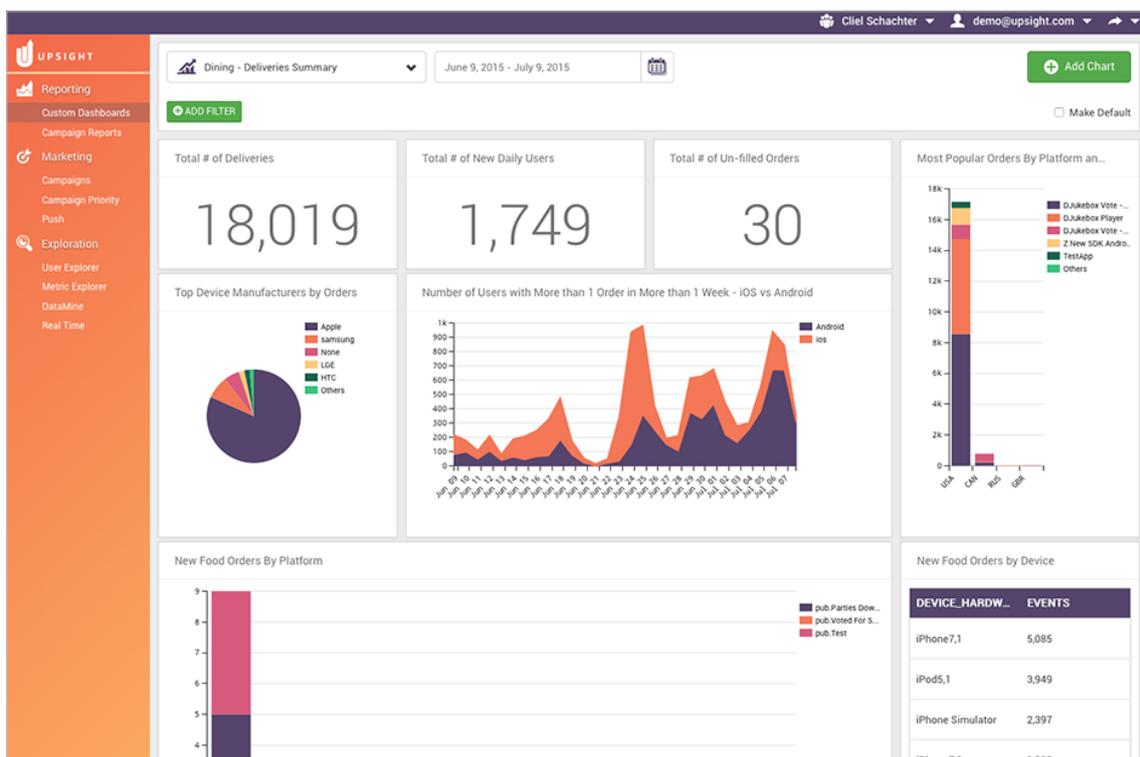


Figura 2.14 - Dashboard do Upsight.

Análise Comparativa

Uma característica bastante relevante, e comum entre todas as aplicações em análise, é a apresentação dos dados a partir das *dashboards*. Estas, traduzem-se em painéis de informação, e possibilitam a visualização de dados a partir dos diferentes tipos de gráficos, desde gráficos de barras, a histogramas, entre outros. A maioria das *dashboards* são personalizáveis, o que possibilita ao utilizador visualizar a informação desejada e de acordo com o seu objetivo. O design de *dashboard* parece funcionar neste tipo de ferramentas, sendo que será uma mais valia pensar nesta característica para o Crowdplay.

Todas as aplicações analisadas são bastante vocacionadas para o marketing e para a gestão de utilizadores (rastrear utilizadores), sendo que os fatores mais importantes são o seu ciclo de vida (desde que instalam, o quanto usam, até que deixam de jogar), e a monetização de compras. Neste sentido existem métricas predefinidas bastante comuns deste tipo de análise: DAU é o número de utilizadores ativos por dia; MAU é o número de utilizadores ativos por mês; ARPU é a receita média por utilizador, ARPPU é a receita média por utilizador que compra e retenção é a medida de quantas pessoas voltam ao aplicativo, com base num determinado período de tempo; entre outras.

É de notar que nas aplicações pagas, o preço é definido consoante a métrica MAU da aplicação a ser testada, ou seja, consoante o número de utilizadores ativos durante o mês.

Nestas aplicações também é possível analisar a experiência de jogador; apesar de não ser uma característica principal, consegue-se recolher informação a partir de métricas personalizáveis. Quando referimos otimização nestas aplicações, isso não se refere ao mesmo tipo de otimização do Crowdplay, pois nestas aplicações a otimização de conteúdo está relacionado com o marketing, publicidade e compras de componentes dos jogos. O mesmo acontece para os teste A/B e a segmentação de conteúdo personalizado para os diferentes grupos de utilizadores, já que o Crowdplay gera as variações de jogo definidas pelo designer para analisar a experiência de jogador. Estas ferramentas estão focadas em objetivos diferentes do Crowdplay, e são mais vocacionadas para os aplicativos móveis, no entanto o seu objetivo base é bastante semelhante, analisar e testar jogos. Apesar de os objetivos serem diferentes, os métodos são semelhantes e existem características importantes a ter em conta.

Para resumir todas as características e funcionalidade analisadas neste capítulo, desenvolvemos uma tabela (figura 2.15) para identificar as características presentes em cada aplicação. Incluímos nesta análise características como: a visualização de dados com a mineração e filtragem de dados; a possibilidade de personalizar as *dashboards*; os teste A/B; a segmentação de utilizadores e conteúdo; resultados em tempo real e ainda a possibilidade de analisar métricas predefinidas e personalizadas, entre outras.

	Game Analytics	Google Analytics	Unity Analytics	Delta DNA	Swrve	Upsight
Visualização de dados (c/ mineração de dados)	✓	✓	✓	✓	✗	✓
Dashboards configuráveis	✓	✓	?	✓	?	✓
Teste A/B	✓	✗	✗	✓	✓	✓
Segmentação de Utilizadores / Conteúdo	✓	✓	✓	✓	✓	✓
Otimização de conteúdo (inclusivé publicidade)	✓	✓	?	✓	✓	✓
Resultados em tempo real	✓	✓	✓	✓	?	?
Métricas predefinidas	✓	✓	✓	✓	✓	✓
Métricas personalizáveis	✓	✓	?	✓	✗	✓
Resultados em tempo real	✓	✓	✓	✓	?	?
Exportar dados em bruto	✓	✓	✓	✓	✓	✓
Documentação / Suporte Técnico / Consultoria	✓/✓/✗	✓/✓/✗	✓/?/✗	✓/?/✓	✓/✓/✗	✓/✓/✓
Custo	Grátis	Grátis	Pago/#MAU (c/ Demo)	Pago/#MAU (c/ Demo)	Pago (c/ Demo)	Pago/#MAU (c/ Demo)
Características relevantes (integração, entre outros)	Integração com Unity, Unreal, Marmalade, Xamarin	API. Foco em testes de Websites	Ferramenta própria do Unity. Mapas Térmicos	API, iOS, Android, Unity. Notificações	iOS, Android, Unity. Email, Notificações	iOS, Android, Unity, tvOS. Notificações

Legenda:

✓ - Característica presente

✗ - Característica não presente

? - Sem informação

Figura 2.15 - Análise comparativa de funcionalidades das aplicações semelhantes.

2.2 Design de Interação

A disciplina de Interação Humano-Computador (IHC), não é propriamente recente, existindo oficialmente desde a década de 1980, com afiliação na maior associação na área da informática, a *Association of Computing Machinery* (ACM). No entanto, as origens desta disciplina antecedem essa altura, surgindo inovações na década de 1960, como a manipulação direta de objetos em 1963 por Ivan Sutherland; a invenção de dispositivos como o rato em 1965 por Douglas Englebart e outras como interfaces gráficas inventadas por Alan Kay e colegas, na Xerox em 1972. (Fonseca, Campos, & Gonçalves, 2012)

Interação Humano-Computador (IHC) é uma área de investigação e prática, que engloba várias disciplinas, incorporando diversos conceitos e abordagens. (Carroll, s.d.)

Esta disciplina de carácter multidisciplinar, tem como bases sólidas e fortes influências, áreas bem reconhecidas, tais como a psicologia cognitiva, a psicologia social e organizacional, a ergonomia, a linguística, a inteligência artificial, a filosofia, entre outras. (Fonseca et al., 2012)

Esta ciência da computação estuda o design, a avaliação, e a implementação de sistemas computacionais interativos para a utilização humana, bem como a interação que resulta desta utilização. Por se tratar de um campo multidisciplinar, como já referido, é baseado em determinadas especialidades da ciência da computação que se tornaram vitais para o seu desenvolvimento, tais como, a ciência cognitiva e os fatores humanos. (Fonseca et al., 2012)

As primeiras interfaces e os primeiros computadores, surgiram na década de 1950, onde os únicos utilizadores nessa altura eram os programadores técnicos do hardware primordial, sendo que a única interface existente era baseada em código e cartões perfurados.

Durante as décadas de 1960 e 1970 evoluiu-se para uma interface a nível da programação, onde aprender a programar era um requisito para saber utilizar um computador e até à altura os únicos humanos que interagem com computadores eram apenas os profissionais da tecnologia.

Apenas em meados de 1980 as interfaces se massificaram ao nível do diálogo da interação; primeiro baseadas em linhas de comando (UNIX e MS-DOS) e posteriormente em grafismos (GUI - *Graphical User Interface*).

Mais tarde, na década de 1990, surgem as interfaces ao nível do trabalho, baseados em sistemas em rede, com suporte ao trabalho colaborativo, denominado de CSCW (*Computer-Supported Cooperative Work*). (Fonseca et al., 2012)

Esta é uma área onde a multidisciplinaridade não é um problema, mas sim uma vantagem; na figura 2.16 podemos observar como se relacionam entre si.

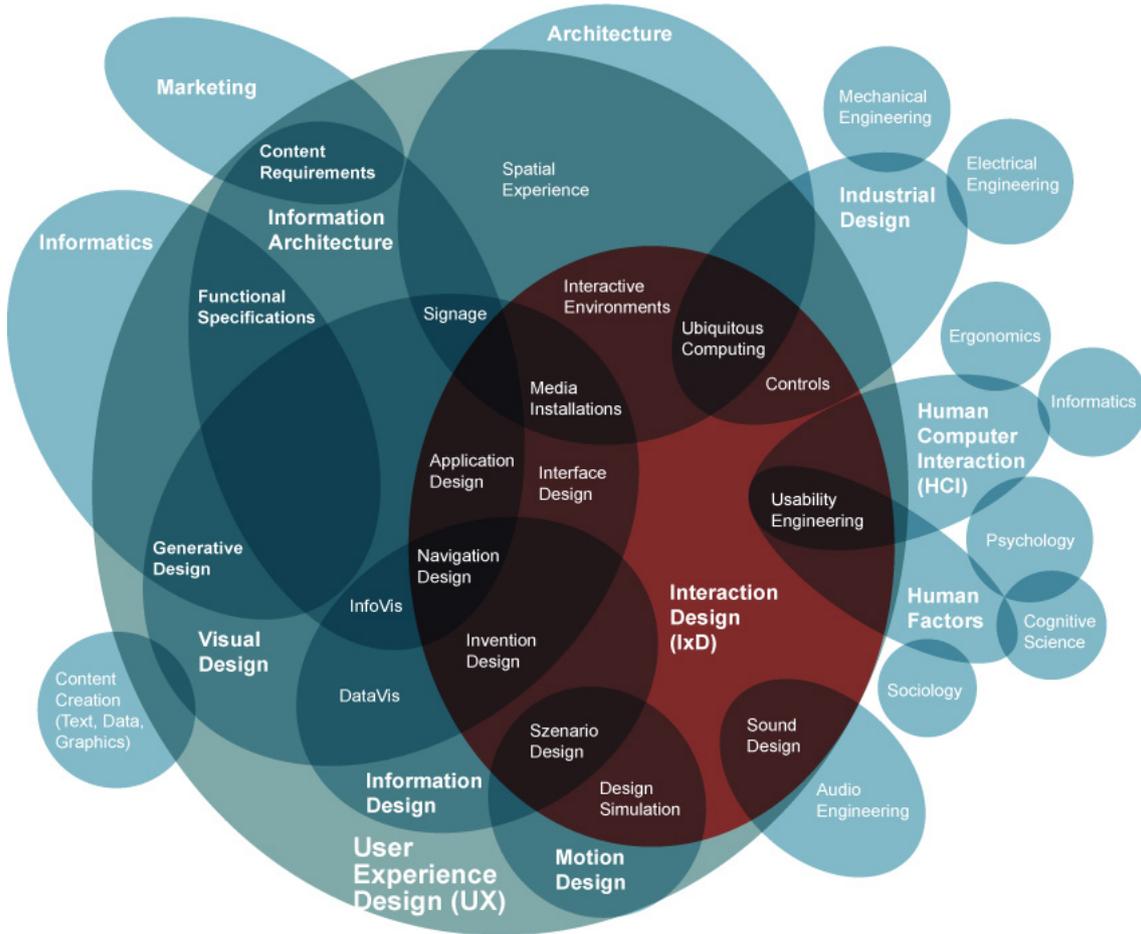


Figura 2.16 - Disciplinas de Experiência de Utilizador. (Carroll, s.d.)

Nesta equação é necessário compreender áreas como, a Experiência do Utilizador (UX), a Interface do Utilizador (UI) e o Design de Interação (IxD), definidas por Miller (2016); no entanto existem algumas variações entre outros autores.

Experiência do Utilizador - (UX - *User Experience*) - abrange todos os aspetos da interação do utilizador com a tecnologia, focando-se em proporcionar melhores experiências nessa relação, aumentando a sua satisfação e melhorando a eficácia e a eficiência.

Interface do Utilizador - (UI - *User Interface*) - é o espaço onde a interação entre o ser humano e o produto ocorre, a qual deve responder às suas necessidades e refletir as suas capacidades, dando mais destaque à informação em si do que ao próprio design. O objetivo é o utilizador estar preocupado com a informação e na tarefa que pretende executar e não nos mecanismos utilizados para apresentar as informações, ou seja, uma interface praticamente invisível, tentando antecipar as necessidades e as ações do utilizador.

Design de Interação - (IxD - *Interaction Design*) - está relacionado com a interação entre o utilizador e o dispositivo. Tem o objetivo de facilitar as ações que queremos usufruir com qualquer sistema, sendo que o produto deve ser intuitivo, fácil de aprender, tornando a tecnologia cativante e agradável de interagir.

De referir que estas áreas estão bastante relacionadas entre si, pois todas elas acabam por se influenciar e depender umas das outras. Interligado-se com a interação humano-computador, o design de interação abrange uma parte do campo desta disciplina.

Até meados da década de 1990 o design de interação não era objeto de grande desenvolvimento, apenas a usabilidade e a engenharia de fatores humanos eram explorados, com foco na psicologia e na ergonomia, para criar interações eficientes e sem erros. Com o aumento da internet e das tecnologias, as duas culturas de design e engenharia desenvolveram interesses comuns, particularmente na área do design de interação e da experiência do utilizador. No virar do século, a noção de design de interação começou a ganhar popularidade, reconhecendo uma abordagem mais criativa do tema. (Lowgren, s.d.)

A nova disciplina de design nasce com o intuito de criar soluções atrativas para um mundo virtual, tendo em consideração as necessidades e desejos das pessoas que usam determinados produtos ou serviços.

Lowgren em 2013 enunciou as cinco principais características do design de interação:

1. O design envolve a mudança de situações, moldando e implementando artefactos - por outras palavras, o design é sobre a transformação e os meios que o designer dispõe para iniciar essa mudança numa situação concreta; ou seja, em última análise, desenvolver o seu projeto digital.

2. O design é sobre exploração de futuros possíveis - isto significa que é necessário explorar e gastar tempo nas fases iniciais, divergindo e analisando diversas possibilidades antes de se comprometer com uma determinada direção. Para além disso é necessário também intervir com os (futuros) utilizadores, fazendo estudos colaborativos e avaliações aos utilizadores. Isto não significa que os testes de utilizadores e a sua avaliação constituam projetos de design de interação, mas sem dúvida fazem parte deles.

3. O design implica enquadrar o “problema” em paralelo com a criação de possíveis “soluções” - tendo em conta a noção de mudança e de exploração, chega-se à conclusão que ao criar algo numa dada situação, esta altera-se. No entanto, se só se enquadrar o problema, qualquer solução de design encontrada será sempre limitada, pelo que se deve explorar e criar várias soluções.

4. O design envolve pensar através de esboços e outras representações tangíveis - é importante esboçar (com papel e lápis) as ideias; mas para o design de interação há implicações particulares, pois ao projetar técnicas inovadoras de interação, por vezes é necessário “esboçar” em *software* (e *hardware*), em vez de ficar apenas pelos esboços de baixa fidelidade.

5. O design aborda aspetos instrumentais, técnicos, estéticos e éticos num todo - cada uma das possíveis soluções exploradas num projeto de design tem considerações e implicações em todas estas dimensões, sem ser possível determinar qual a ordem de influência de cada uma delas. No design de interação passa-se o mesmo: decisões técnicas influenciam a qualidade estética da interação, assim como escolhas instrumentais podem ter repercussões éticas, entre outras.

As restrições ou conveniências tecnológicas, assim como os caprichos do cliente ou utilizador, não devem influenciar as soluções de design. Um bom projeto de design deve proporcionar soluções simples e elegantes que traduzem verdadeiras experiências de utilizador e não apenas uso de ferramentas por mais versáteis que elas sejam. (Fonseca et al., 2012)

O design de interação é sobre o comportamento, e este é uma interação, uma transação entre duas entidades, tipicamente uma troca de informações. A própria interação ocorre entre pessoas, máquinas e sistemas, numa variedade de combinações. (Saffer, 2010)

Dan Saffer (2010) definiu aspetos a ter em conta quando projetamos interfaces:

1. Focar no utilizador

Os designers devem preocupar-se em fazer as tarefas e alcançar os objetivos dentro dos limites, das necessidades e dos objetivos do utilizador. Compreender o utilizador é a chave do sucesso.

2. Encontrar alternativas

Ao projetar novos produtos devem-se explorar múltiplas opções. Quando se tem duas soluções, deve-se encontrar sempre uma terceira alternativa.

3. Explorar ideias e fazer prototipagem

Deve-se ter várias ideias para encontrar boas soluções e através da prototipagem consegue-se encontrar a melhor opção, excluindo assim as ideias menos boas logo numa fase inicial.

4. Colaborar e resolver restrições

Os designers devem saber trabalhar de forma colaborativa e tirar partido disso para resolver as restrições e rentabilizar vários recursos.

5. Criar soluções apropriadas

Devem ser criadas soluções adequadas a cada problema, em cada projeto. A solução definitiva deve abordar exclusivamente as questões desse problema específico, tendo em conta o contexto em que está inserido o produto e os utilizadores.

6. Desenvolver com um amplo campo de influências

O designer procura inspirações e soluções nas diversas áreas que o design abrange (áreas como a psicologia, ergonomia, engenharia, arquitectura, arte e muito mais).

7. Saber incorporar a emoção

A emoção é normalmente vista como um impedimento da lógica e de tomar escolhas certas, mas um produto sem uma componente emocional é um produto sem vida, e não cria um elo de ligação com as pessoas.

Depois de analisar as boas práticas sugeridas por Saffer, há também que referir o ciclo de interação de Norman. Este modelo permitiu a identificação das fases da interação entre o utilizador e a interface. O ciclo é interativo e pode ser dividido em duas fases principais: execução e avaliação; sendo que estas podem ainda ser subdivididos nas designadas sete etapas da interação: definir o objetivo, formar a intenção, especificar a ação, executar a ação, perceber o estado do sistema, interpretar o estado do sistema e avaliar o resultado. (Dix, Finlay, Abowd, & Beale, 2004)

As etapas de execução traduzem o objetivo inicial numa intenção para executar, e este precisa de ser enquadrado no domínio da aplicação e posteriormente na interface. Já as etapas de avaliação iniciam-se com a perceção do sistema, que é interpretada de acordo com as expectativas e avaliada em função das intenções e dos objetivos. Há também que referir que as etapas de execução estão inseridas num nível físico, enquanto que as etapas de avaliação referem-se a um nível mental. Este modelo de Norman serve essencialmente para lidar com os problemas de interação de um modo estruturado, podendo extrair-se ainda um conjunto de heurísticas, para aplicar no projeto de sistemas interativos.

2.2.1 Design de Interface

Enquanto o design de interação trata do desenvolvimento do fluxo da aplicação para facilitar as tarefas do utilizador, definindo como este interage com o sistema, o design de interface define os componentes da interface, o que permite ao utilizador realizar as tarefas no sistema.

O design de interface deve ser desenvolvido para apoiar as tarefas que os utilizadores querem realmente fazer e não ter apenas um aspeto agradável e bonito.

Para desenvolver uma boa interface com foco na satisfação do utilizador, Shneiderman (2005) enunciou um conjunto de regras, designadas regras de ouro do design de interface.

Estas regras são aplicáveis à maioria dos sistemas interativos, mas devem ser validados e ajustados para domínios específicos. Destinam-se a serem utilizados durante o design, mas também podem ser aplicados, tal como as heurísticas de Nielsen (ver capítulo 2.2.2) para a avaliação de sistemas.

Em seguida, são descritas as oito regras de ouro de Shneiderman:

1. Manter consistência - situações semelhantes exigem sequências de ação consistentes, ou seja, terminologia, cor, layout, fonte e outros parâmetros devem ser definidos de forma coerente e consistente. Exceções a esta regra devem ser compreendidas e limitadas. Esta regra é a mais frequentemente violada, dado que a consistência pode assumir várias formas.

2. Atender à usabilidade universal - a aplicação deve ser suficientemente flexível para atender aos diferentes tipos de utilizadores. Adequar conteúdos e introduzir recursos apropriados às necessidades de cada utilizador (principiantes, especialistas, faixas etárias, conhecimento tecnológico...) enriquece o design de interface e melhora a qualidade do sistema.

3. Oferecer feedback informativo - o sistema deve fornecer *feedback* informativo de todas as ações realizadas pelo utilizador. A resposta pode ser mais modesta e subtil, ou mais substancial e explícita conforme o tipo de ação.

4. Criar diálogos para fechar ações - sequências de ações devem ser organizadas (com início, meio e fim). O retorno informativo ao término de um grupo de ações dá aos utilizadores a satisfação de realização, podendo assim progredir para a próxima etapa.

5. Evitar erros - sempre que possível, deve se desenvolver um sistema que evite o utilizador cometer erros sérios. E caso um utilizador cometa um erro, a interface deve oferecer instruções simples, construtivas e específicas para a recuperar a ação.

6. Permitir a fácil reversão das ações - tanto quanto possível, as ações devem ser reversíveis. Esse recurso alivia a ansiedade, uma vez que o utilizador sabe que as ações ou erros podem ser desfeitos, incentivando assim a exploração de opções desconhecidas.

7. Suportar o controle do utilizador - os utilizadores desejam a sensação de que eles são responsáveis pela interface e que a interface responde às suas ações. Deve-se evitar por parte dos utilizadores a incapacidade de produzir a ação desejada ou dificuldades em obter determinadas informações necessárias, pois geram ansiedade e insatisfação.

8. Reduzir a carga de memória de curta duração - as interfaces devem evitar levar o utilizador a memorizar grande quantidade de informação, para não sobrecarregar a memória. Isto exige que as interfaces sejam simples, com movimentos reduzidos e um tempo de treinamento suficiente para códigos, mnemónicas e sequências de ações.

Não é possível desenvolver uma interface sem antes a testar, assim como também não é possível testar sem antes desenvolver protótipos. Estes podem ser divididos em duas categorias, os protótipos de baixa e de alta fidelidade. Numa primeira fase deve-se desenvolver os protótipos de baixa fidelidade, que envolvem na maioria dos casos esboços rápidos no papel. Estes, são simples e não polidos, com poucos atributos visuais, mas com validade para testar o conceito global e o fluxo do produto. Uma vez que os conceitos e a navegação geral tenham sido prototipados com recurso a métodos de baixa fidelidade, segue-se a fase de prototipagem de alta fidelidade, procurando obter uma aproximação à experiência do produto final. (Saffer, 2010)

Os protótipos de alta fidelidade são na sua maioria interativos (ou clicáveis), no entanto isto não é uma característica obrigatória. A diferença deste último, é que funciona (na maioria das vezes) como era suposto; o sistema comporta-se como seria no real.

Estes princípios subjacentes têm as suas limitações, mas fornecem um bom ponto de partida para os designers. Não só aumentam a produtividade dos utilizadores como também ajudam no desenvolvimento, e com isto na produtividade dos designers.

2.2.2 Usabilidade

A usabilidade aumenta a produtividade através de designs mais eficientes, que suportam melhor as tarefas dos seus utilizadores.

Como referido por Cockton (s.d), a ISO 9241-11 (*International Organization for Standardization*) distingue três componentes importantes da usabilidade: eficácia, eficiência e satisfação; e define-a como: “a medida em que um produto pode ser utilizado por utilizadores específicos para atingir objetivos específicos com eficácia, eficiência e satisfação num dado contexto de utilização”.

(Cockton, s.d)

No entanto, de acordo com Nielsen (1993), é importante compreender que a usabilidade não é uma propriedade simples e unidimensional da interface do utilizador, pois tem múltiplas componentes, de que se destacam os cinco atributos seguintes que devem constituir o foco da atenção do designer de interface.

1. Há que ter em conta a **facilidade de aprendizagem**, ou seja, o sistema deve ser suficientemente fácil de aprender para que o utilizador possa rapidamente efetuar as tarefas desejadas.
2. A **eficiência** relaciona-se com a capacidade do sistema atingir um alto nível de produtividade, uma vez aprendido. Quanto mais eficiente for o sistema, maior a velocidade com que os utilizadores realizam as suas tarefas.
3. Um sistema deve ser **fácil de memorizar**, por forma a que os utilizadores casuais possam reutilizar o sistema, ao fim de algum tempo sem o usar, sem ter que o aprender novamente.
4. Um sistema com **prevenção de erros** não só permite que o utilizador cometa poucos erros ou se os produzir permite recuperar o erro. Este atributo não inclui erros de sistema, apenas se refere aos erros cometidos pelo utilizador.
5. Por último, a **satisfação** depende de um sistema agradável na sua utilização propiciando uma satisfação subjectiva ao utilizador.

São estes atributos que transformam o conceito abstrato de usabilidade numa verdadeira engenharia de usabilidade que pode ser aplicada, avaliada e melhorada. Mesmo não se fazendo estudos formais sobre a usabilidade, estes atributos permitem um exercício fácil de medir a usabilidade de um sistema. (Nielsen, 1993)

Nielsen desenvolveu ainda os 10 princípios com base no trabalho anterior sobre avaliação heurística em colaboração com Rolf Molich, em 1990. Estes 10 princípios para design de interação são chamados de heurísticas, pois são processos cognitivos para decisões não racionais e não devem ser tomados como orientações específicas de usabilidade. (Nielsen, 1995a)

1. Visibilidade do estado do sistema

O sistema deve manter sempre os utilizadores informados sobre o que está a acontecer, através de *feedback* adequado em tempo útil.

2. Correspondência entre o sistema e o mundo real

O sistema deve falar o idioma dos utilizadores, usando palavras, frases e conceitos familiares, em vez de termos orientados ao sistema, e deve seguir as convenções do mundo real. A comunicação deve ser natural e intuitiva.

3. Controle e livre arbítrio do utilizador

Os utilizadores devem ter a liberdade para selecionar e realizar as tarefas pela ordem que desejarem, em vez de ser o sistema a impor. Frequentemente os utilizadores escolhem funções do sistema por engano, necessitando de “saídas de emergência” bem assinaladas para poder sair do estado indesejado sem ter de percorrer um caminho longo, com suporte de desfazer e refazer.

4. Consistência e padrões

Os utilizadores não devem ter que se preocupar se diferentes palavras, situações ou ações significam a mesma coisa em contextos diferentes. Opções semelhantes devem ser tratadas de igual forma seguindo as convenções da plataforma.

5. Prevenção de erros

Melhor do que uma boa mensagem de erro é um design cuidado que previne a ocorrência de erros. A interface deve fazer com que seja difícil cometer erros, eliminando as condições propensas ao erro, verificando-as, e apresentando ao utilizador uma opção de confirmação antes que este se comprometa com a ação.

6. Reconhecimento em vez de recordação

Deve-se minimizar a carga de memória do utilizador, tornando visíveis os objetos, as ações e as opções. O utilizador não deve ter de se lembrar de informações de uma parte da interface para outra, devendo as instruções de utilização do sistema serem visíveis ou facilmente recuperáveis.

7. Flexibilidade e eficiência

A interface deve permitir ao utilizador a personalização das ações mais frequentes, disponibilizando aceleradores, invisíveis para os utilizadores principiantes, que acelerem as ações dos peritos. Deste modo o sistema ganha em eficiência e adapta-se a todos os tipos de utilizador, sendo flexível.

8. Estética e design minimalista

Os diálogos da interface não devem conter informação irrelevante ou raramente necessárias. Cada unidade de informação extra num diálogo compete com a informação verdadeiramente relevante e diminui a sua visibilidade relativa.

9. Ajudar o utilizador a reconhecer, diagnosticar e recuperar erros

Mensagens de erro devem ser expressas em linguagem simples e clara (sem códigos), indicando com precisão o problema e sugerindo construtivamente uma solução.

10. Ajuda e documentação

Poucos sistemas podem ser usados sem instruções, mesmo que isso fosse o ideal. Portanto, pode ser necessário fornecer ajuda e documentação. A informação deve ser fácil de pesquisar, focada na tarefa do utilizador, listar as etapas a realizar de modo concreto e ser breve.

Existem vários conjuntos de heurísticas de usabilidade, por vezes usadas em simultâneo. Todas têm uma visão válida do que é uma boa interface de utilizador, diferem apenas na forma de organizar essa visão em regras práticas. Além do conjunto de heurísticas de Nielsen, existem também as oito regras de ouro de Shneiderman (já descritas anteriormente), a lista de 16 princípios de Bruce Tognazzini ou os princípios de design de Norman, por exemplo. (Fonseca et al., 2012)

As heurísticas podem ser utilizadas tanto durante o processo de design da interface, guiando os designers na escolha das melhores alternativas, como durante a avaliação para encontrar e justificar os problemas de usabilidade das interfaces.

2.2.3 Avaliação

Como temos vindo a falar, a avaliação é um ponto fulcral no desenvolvimento de interfaces e design de interação. Nos capítulos anteriores discutimos processos para apoiar o desenvolvimento de sistemas interativos, no entanto, mesmo usando todos esses processos e/ou princípios, é necessário avaliá-los e testá-los. A avaliação não deve ser apenas pensada como uma fase única no processo de design, muito menos como uma atividade final desse processo; deve sim ocorrer ao longo de todo o ciclo de desenvolvimento do projeto. Nem sempre é possível realizar os testes extensivos continuamente ao longo de todo o processo, mas as técnicas analíticas e informais podem e devem ser sempre usadas e mantidas em mente. Relativo a este aspeto, existe uma estreita ligação entre a avaliação e os princípios e técnicas de desenvolvimento, técnicas essas que ajudam a garantir a avaliação contínua do projeto. Assim os problemas podem ser resolvidos com um mínimo de esforço e recursos envolvidos na implementação, pois é muito mais fácil corrigir um projeto nas fases iniciais do desenvolvimento do que nas fases posteriores.

A avaliação tem como objetivos principais, avaliar a usabilidade e a experiência dos utilizadores, assim como identificar os problemas de design e erros específicos do sistema, e pode ser efetuada de diferentes modos. Os diferentes tipos de avaliação vão desde avaliações analíticas e preditivas, em que não se envolvem utilizadores, nomeadamente a avaliação heurística, passando também pela avaliação com os utilizadores, nomeadamente testes e questionários. (Dix et al., 2004)

A avaliação heurística, ou avaliação por peritos, é uma técnica desenvolvida por Nielsen e Molich, a qual envolve ter um pequeno grupo de avaliadores examinando a interface de acordo com os princípios da usabilidade reconhecidos como heurísticas. Durante a avaliação os avaliadores ou peritos examinam a interface colocando-se no lugar do utilizador e registando os resultados em relatórios escritos. (Nielsen, 1995b) Um dos problemas deste tipo de avaliação, centra-se no facto de o avaliador não ser o utilizador, pois embora o perito tente passar por um utilizador comum ele sabe bastante sobre interfaces e usabilidade para se comportar como um. A avaliação heurística também pode ser pelo próprio designer, se não existirem recursos para a alocação de peritos.

Os testes de usabilidade aos utilizadores baseiam-se numa técnica de avaliação que envolve medir o desempenho e a satisfação de utilizadores com diferentes métodos desde: métodos empíricos ou experimentais, métodos observacionais, inquéritos ou até métodos que utilizam monitorização fisiológica.

Neste tipo de avaliação, os testes de utilizadores mais comuns são os métodos empíricos, onde se pede para efetuarem um conjunto de tarefas, ou os métodos experimentais onde os utilizadores são livres de explorar o sistema. Nestes dois casos, mas principalmente no primeiro modelo, é importante recolher as medidas e métricas de desempenho, como o tempo para concluir cada tarefa e o número de erros cometidos. Posto isto, há que analisar todos esses dados para recolher informações, para melhor caracterizar os problemas (desde os mais aos menos graves/importantes) e assim poder resolvê-los. Quando planeamos os testes, existem ainda outros fatores a ter em conta, desde o planeamento (desenvolver um guião de tarefas), a preparação (onde, quando e o que que é necessário) e preparar e introduzir os utilizadores (explicar o processo, promover o conforto e a privacidade).

Segundo Nielsen (2000) o número ideal de utilizadores para desenvolver os testes é de cinco, de acordo com a sua pesquisa. Os resultados dos seus testes apontam que, a partir de cinco utilizadores, os dados se tornam redundantes, pelo que ter mais do que cinco utilizadores em cada teste é um desperdício de recursos e tempo. Com cinco utilizadores conseguimos encontrar cerca de 85% dos problemas, e acima deste valor (cinco utilizadores) a probabilidade de encontrar novos erros é cada vez mais reduzida.

Isto aplica-se tanto aos testes com utilizadores como à avaliação heurística com peritos. (Nielsen, 1995b, 2000)

Estes dois tipos de avaliação não se sobrepõem. Podem ser como alternativa (ou um ou outro), mas podem ser usados em complemento na avaliação. Isto permite encontrar problemas e erros de diferentes naturezas. Enquanto uma avaliação com utilizadores encontra problemas mais relacionados com a interação e usabilidade, a avaliação heurística pode encontrar erros relativos ao design da interface.

Existem ainda outros modos de avaliação, com ou sem recurso a utilizadores. Os testes *cognitive walkthrough*, referem-se, como o nome indica, a uma avaliação passo a passo; a sequência de ações refere-se às etapas que uma interface exige de um utilizador para realizar a tarefa.

Os avaliadores ou peritos percorrem essa sequência para verificar se existem potenciais problemas relativos à usabilidade, tendo como foco principal, estabelecer o quão fácil é aprender o sistema. Para elaborar uma avaliação *cognitive walkthrough*, referindo-se ao procedimento cognitivo, é necessário ter em conta os seguintes quatro aspetos: (1) uma especificação ou protótipo do sistema a avaliar; (2) uma descrição da(s) tarefa(s) que o utilizador deve executar no sistema; (3) uma sequência completa de ações necessárias para completar a(s) tarefa(s) e por fim (4) uma descrição do tipo de utilizadores e de experiência destes. (Dix et al., 2004)

Outro método de avaliação são os questionários, podendo variar quanto ao objetivo (qual a informação procurada) como também no modo de mensuração (comentários específicos ou pontuação mensurável). Um exemplo disto é o questionário S.U.S. (*System Usability Scale*) com uma escala tipo *Likert* composta por dez itens que dá uma visão global da avaliação subjetiva da usabilidade. Estes são respondidos pelos utilizadores e normalmente servem para complementar outros métodos de avaliação, como por exemplo os testes formais de usabilidade.

Para este questionário (S.U.S.) foram selecionadas um conjunto de declarações que cobrem uma variedade de aspetos da usabilidade do sistema tais como, necessidade de ajuda, utilidade do sistema, complexidade, entre outros, apresentando validade para medir a usabilidade de um sistema. (Brooke, 1996)

Como já referimos, design de interação é um processo iterativo, onde a cada novo ciclo de desenvolvimento e testes, vamos melhorando o sistema. Para acompanhar este processo existem técnicas de prototipagem, desde as de baixa fidelidade às de alta fidelidade, ou seja, inicialmente podemos começar com protótipos em papel e ir evoluindo até protótipos funcionais, rentabilizando melhor o tempo.

2.3 Análise Transversal do Estado de Arte

Este capítulo permitiu compreender o Crowdplay, desde a abordagem teórica subjacente até ao seu funcionamento.

Compreendendo o seu pressuposto principal, a abordagem de geração de conteúdo procedimental, torna-se mais fácil entender e tirar partido do funcionamento da plataforma. Foi também fundamental para conhecer o processo de desenvolvimento anterior do Crowdplay.

A análise das aplicações semelhantes ao Crowdplay, não deixa de ser importante para identificar o que existe nesta área. No entanto as ferramentas encontradas não se focam propriamente nos mesmos aspetos de análise e otimização do Crowdplay, estando mais focadas de um modo geral para o marketing. Apesar de não corresponderem completamente às mesmas funcionalidade e estarem mais focadas para outro tipo de teste e análise de jogos, servem de inspiração para o design, as *dashboards*.

O capítulo de design de interação foi, sem dúvida, bastante importante para consolidar todos os aspetos e assim melhorar todo o desenvolvimento prático deste trabalho.

É fundamental compreender e conhecer os conceitos de design de interação, design de interface e usabilidade, assim como todos os fatores e princípios a ter em conta no desenvolvimento de um projeto de design. Há ainda que referenciar a importância da avaliação e os testes de usabilidade com a finalidade de melhorar o produto e a experiência do utilizador.

3

- 3.1 Público Alvo
- 3.2 Cenários e Personas
- 3.3 Requisitos e Funcionalidades
- 3.4 Análise da Interface
- 3.5 Casos de uso
- 3.6 Testes de Usabilidade

Avaliação Preliminar

3. Avaliação Preliminar

Neste capítulo é descrita a análise desenvolvida à plataforma inicial do Crowdplay. Foi revista e avaliada a plataforma tendo em conta vários aspetos de usabilidade, desenvolvendo cenários, requisitos, funcionalidades, e ainda os casos de uso. Por fim foram preparados e realizados os testes de usabilidade.

3.1 Público Alvo

O público alvo do Crowdplay são todos os criadores envolvidos na produção de um videojogo - sejam eles designers ou programadores de jogos. Dentro do público alvo podemos classificar três tipologias de criadores - designer de jogo, programador e designer de interação - que por via de percursos académico-profissionais diversificados, terão um diferente leque de competências e objetivos. Estas diferentes competências resultarão numa potencial divergência de necessidades no que corresponde à plataforma Crowdplay.

Estipulamos que num exemplo típico, os designers de jogo estarão mais interessados em explorar diferentes variações de um determinado jogo, utilizando a plataforma como ferramenta integrada num processo de design de jogo, como forma de explorar o espaço de design, obtendo dados sobre um conjunto de objetos (jogos) gerados pela plataforma, tentando mapear como pequenas oscilações nos mesmos influenciam a experiência do jogador. Chamamos a este caso de uso - exploração. Já um programador poderá querer afinar mecânicas e outros elementos de baixo nível do design, de forma a que estes se equilibrem e otimizem a experiência do jogador como imaginada pelo designer, no que denominamos otimização. Finalmente, o designer de interação estará focado num processo de teste (avaliação ou análise) de um jogo cujo design e implementação final estão fechados, e por isso apenas terá de verificar se a experiência dos jogadores está alinhada com o design de jogo (como visionado pelo designer).

Note-se que a atribuição é artificial, e serve apenas para guiar o design da aplicação de forma a que contemple estas diferentes tipologias de interação, não havendo qualquer discriminação do seu uso face ao tipo de utilizador; isto é, a plataforma deve estar aberta a que um designer de jogo faça otimização, um programador teste o jogo e por aí em diante.

3.2 Cenários e Personas

De seguida são apresentados os cenários construídos para exemplificar os diferentes modos de utilizar o Crowdplay, cada um associado a um tipo diferente de utilizador/persona.

O João é designer de interação e quer testar o jogo da sua equipa. Considerando que o jogo já está submetido na plataforma, ele apenas precisa de definir o número máximo de sessões e iniciar o teste. Após os testers terem jogado esse número de vezes, ele pode observar os resultados obtidos.

O Miguel é designer de jogo e pretende usar a aplicação para explorar variações de jogo. Para Miguel explorar as variações de jogo necessita de definir as variações de jogo, escolhendo a função probabilística e o seu tipo. Depois de os testers jogarem os jogos candidatos, o Miguel pode visualizar os resultados obtidos e tirar conclusões para variar e melhorar o seu jogo.

O Pedro é programador e necessita de otimizar a experiência de jogo. Para otimizar o jogo, Pedro precisa de definir as variações de jogo e os objetivos de design. Nas variações de jogo, define as diferentes variáveis, definindo o seu intervalo e a função da probabilidade de variação. Nos objetivos de design, define a pontuação para cada qualidade a partir de uma determinada fórmula (criada ou editada por ele próprio). Após ativar todas as características/atributos que ele deseja testar, define que o teste acaba quando uma solução chegar a uma determinada pontuação. Depois dos testes com os jogadores estarem concluídos, o Pedro pode obter o jogo otimizado.

3.3 Requisitos e Funcionalidades

Requisitos não funcionais

Tendo em conta o projeto desta dissertação, os requisitos não funcionais mais importantes são, sem dúvida, os requisitos de usabilidade. A plataforma web tem de ser fácil de usar e bastante intuitiva, aliando sempre com um design simples e agradável à vista.

Deve-se ter em conta a facilidade de aprendizagem, de modo a que o utilizador possa rapidamente efetuar as tarefas e ser, de igual modo, fácil de memorizar, para que, ao fim de algum tempo sem utilização do sistema, se recordem sem ter que aprender de novo.

Para ajudar na usabilidade do sistema é também importante estudar a terminologia, de modo a fazer uma correspondência coerente entre o sistema e o mundo real, para que a interface comunique com o utilizador de uma forma natural e intuitiva.

Para além desses requisitos de usabilidade, a aplicação deve responder rapidamente, sem falhas, funcionando em todos os *browsers* mais comuns.

Requisitos funcionais

Quando falamos de requisitos funcionais, estes aproximam-se das funcionalidades, que por sua vez já se encontram implementadas na versão atual. No entanto, determinadas funcionalidades sofreram alterações no decorrer desta dissertação, apresentadas no capítulo 5.2 - Protótipo Interativo.

Os requisitos funcionais estipulados são os seguintes:

- *Login e registo*
- *Página de Design Problem*
 - *Definir Variações de Jogo*
 - *Selecionar variável, função probabilística, intervalo (...)*
 - *Criar, Editar e/ou Apagar Variações de Jogo*
 - *Definir Objetivos de Design*
 - *Selecionar nome, função de pontuação, fórmula (...)*
 - *Criar, editar e/ou apagar fórmulas*
 - *Criar, editar e/ou apagar Objetivos de Design*
- *Página de Test Setup*
 - *Definir (características do) Teste*
 - *Iniciar, pausar e parar Teste*
 - *Sumário do Design Problem*
- *Página Results*
 - *Selecionar vistas e tipo de gráfico*
 - *Escolher dados a visualizar*
- *Página New Experiment (admin)*
 - *Registar novos utilizadores*
 - *Editar permissões de utilizadores*
- *Logout*

Diagrama UED

Para uma melhor análise e compreensão da navegação entre páginas e das suas funcionalidades, desenvolveu-se um diagrama UED (*User Environment Design*) que demonstra o fluxo da plataforma, com informação das funções, *links* e objetos de cada página (figura 3.1).

É de notar que é possível navegar entre as quatro páginas (*New Experiment*, *Design Problem*, *Test Setup* e *Results*), apenas com um clique a partir do menu, mas no diagrama preferiu-se representar o fluxo mais natural e comum de todo o processo ao usar a plataforma. A tracejado estão representados áreas que se expandem dentro da página e as janelas *popups*.

Este diagrama foi desenvolvido de acordo com a plataforma atual, e serve para encontrar possíveis erros e planificar melhor a nova proposta. Uma das falhas mais significativas encontradas ao desenvolver este diagrama encontra-se na página de *New Experiment*, a qual, para além de conter informação que apenas devia ser visível para o *admin*, está mal estruturada.

Ao pensar numa resolução para este problema, surgiu uma nova ideia; a de criar três tipos distintos de utilizadores, ou seja, distinguir o administrador de um utilizador comum e ainda acrescentar uma nova conta de jogadores, os *testers*. Assim, a nova versão tem a possibilidade de fazer *login* com três tipos distintos de utilizadores, na qual o administrador tem acesso a um ecrã para gerir os utilizadores e criar novos projetos, enquanto o utilizador comum, o designer de jogos, pode visualizar os vários projetos em que está envolvido. Foi ainda acrescentada uma nova conta para os jogadores, os que testam os jogos, para que estes sejam notificados quando existir novos testes e possam fazer o *download* do jogo candidato.

O utilizador, ao entrar na plataforma, deveria ter a possibilidade de escolher o projeto, sem ter que fazer *logout*, e esta informação deveria estar na primeira página. No estado atual o designer não tem a possibilidade de mudar de projeto, assim como não pode escolher o tipo de teste (caso de uso) que pretende fazer, pois isso está subentendido consoante os parâmetros que definimos. Na nova versão, estes dois problemas foram corrigidos, de modo a que seja possível ao utilizador escolher desde o projeto ao tipo de teste que pretende efetuar.

Outro problema detetado relativo à navegação, é a existência de excessivas janelas *popups* que abrem para escolher fórmula e adicionar/editar fórmula, para a qual foi repensada toda uma nova navegação entre páginas.

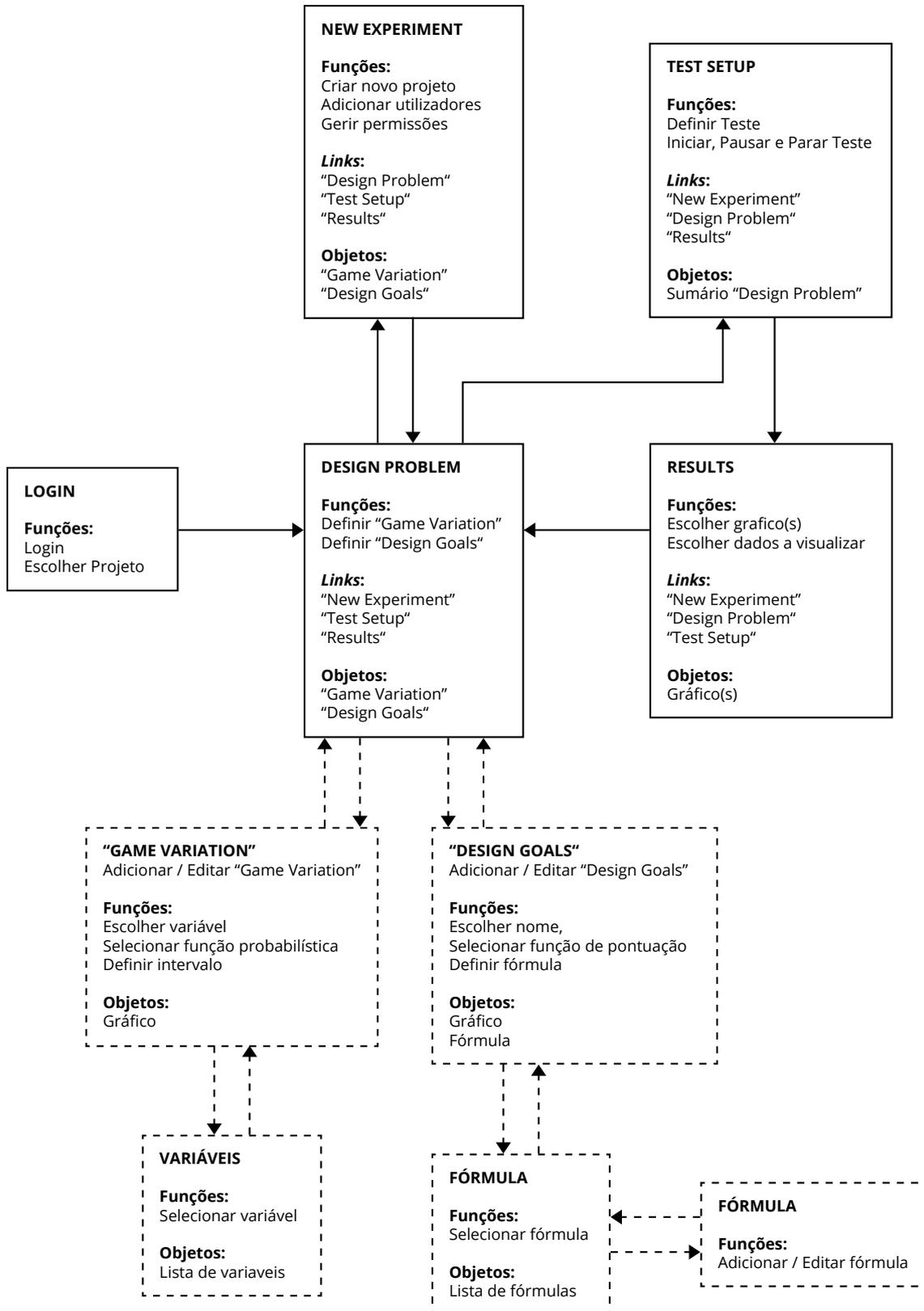


Figura 3.1 - Diagrama UED.

3.4 Análise da Interface

Para um melhor conhecimento do estado atual da interface do Crowdplay serão de seguida apresentados e analisados os ecrãs. Serve isto também de auxílio visual para melhor compreensão do diagrama UED (apresentado na secção 3.3).

Ao efetuar o *login* entramos na página de *Design Problem* (figura 3.2), e nesta encontramos as duas secções: *game variations* e *design goals*. Tanto nas variações de jogo como nos objetivos de design podemos ver as respetivas características e se estes se encontram ativos para o teste.

Na tabela das variações de jogo consta a informação do intervalo e do tipo de função para variar no teste.

CrowdPlay New Experiment **Design Problem** Test Setup Results Logout

Game Variation [Add/Remove]

Vary	Game Variation	Test Range	Variable Function
<input checked="" type="checkbox"/>	Life Lost Stopped	1.0 < Life Lost Stopped < 3.0	step
<input checked="" type="checkbox"/>	Enemy Life Giving	5.0 < Enemy Life Giving < 50.0	step
<input checked="" type="checkbox"/>	Moving Life Lost	-3.0 < Moving Life Lost < -1.0	step
<input checked="" type="checkbox"/>	PlayerUnits_Weapon Damage	5.0 < PlayerUnits_Weapon Damage < 50.0	step
<input checked="" type="checkbox"/>	Spice Quota	10000.0 < Spice Quota < 100000.0	step
<input type="checkbox"/>	IX_HitPoints	0.0 < IX_HitPoints < 100.0	step
<input type="checkbox"/>	Tank_Weapon Damage	0.0 < Tank_Weapon Damage < 100.0	triangle

Design Goals [Add/Remove]

Optimization	Target Experience Quality	Test Condition	Test score	Score Function
<input checked="" type="checkbox"/>	Selects	0.0 < Selects < 200.0	100.0	triangle
<input checked="" type="checkbox"/>	Cenas	0.0 < Cenas < 100.0	100.0	triangle
<input checked="" type="checkbox"/>	GameEndExit	0.0 < GameEndExit < 3.0	100.0	triangle
<input type="checkbox"/>	GameEndWin	0.0 < GameEndWin < 2.0	100.0	triangle
<input type="checkbox"/>	GameEndLose	0.0 < GameEndLose < 3.0	100.0	triangle

Figura 3.2 - Ecrã *Design Problem*.

CrowdPlay New Experiment **Design Problem** Test Setup Results Logout

Game Variation [Add/Remove]

Vary	Game Variation	Test Range	Variable Function
<input checked="" type="checkbox"/>	Life Lost Stopped	1.0 < Life Lost Stopped < 3.0	step
<input checked="" type="checkbox"/>	Enemy Life Giving	5.0 < Enemy Life Giving < 50.0	step
<input checked="" type="checkbox"/>	Moving Life Lost	-3.0 < Moving Life Lost < -1.0	step
<input checked="" type="checkbox"/>	PlayerUnits_Weapon Damage	5.0 < PlayerUnits_Weapon Damage < 50.0	step
<input checked="" type="checkbox"/>	Spice Quota	10000.0 < Spice Quota < 100000.0	step
<input type="checkbox"/>	IX_HitPoints	0.0 < IX_HitPoints < 100.0	step
<input type="checkbox"/>	Tank_Weapon Damage	0.0 < Tank_Weapon Damage < 100.0	triangle

Game Variation details

Get Registered Game Variables

PlayerUnits_Weapon Dan

Name

PlayerUnits_Weapon Damage

Type

Float/Rational

Probability Function

Step

Increment

5

Description

Description

Probability Function

Design Goals [Add/Remove]

Figura 3.3 - Ecrã *Design Problem* com a secção *Game Variation* expandida.

Ao seleccionar uma variação de jogo para a editar podemos ver mais informação em detalhe, podendo redefinir cada componente (figura 3.3). Um dos problemas encontra-se nesta ação, pois para editar uma variável é necessário seleccioná-la primeiro, o que não é muito intuitivo, pois tem-se a tendência para clicar primeiro no botão de editar. O mesmo acontece quando queremos apagar uma variação de jogo, é necessário seleccioná-la antes de clicar no botão para a eliminar.

Na secção que mostra em detalhe cada variação de jogo, podemos ver o gráfico que demonstra o intervalo e o tipo de função, sendo este importante para compreender como cada variação vai variar dentro de um teste. Para adicionar uma nova variação de jogo abre-se uma secção semelhante que permite definir todas as características pretendidas. Uma outra dificuldade encontrada nesta página é o facto de não termos a possibilidade de alterar a visualização em detalhe para outra variação de jogo. Devia ser possível alterar sem ter que fechar ou guardar a variação de jogo expandida, assim como, poder visualizar em detalhe todas as variações e não apenas no momento da edição.

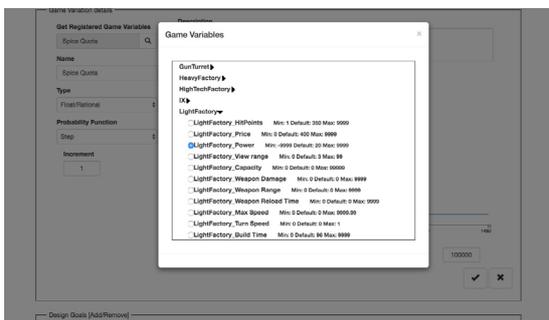


Figura 3.4 - *Popup* para seleccionar a variável de jogo.

Nesta secção, para seleccionar a variável abre-se um *popup*, como podemos ver na figura 3.4, na qual, cada variável se encontra dentro da sua categoria (expansível).

No ecrã seguinte (figura 3.5) podemos observar um exemplo de um objetivo de jogo expandido com as funcionalidades bastante semelhantes às descritas anteriormente para o ecrã de uma variação de jogo. Os mesmos problemas e falhas encontrados também ocorrem nesta área.

Nesta secção podemos igualmente ver mais informação sobre cada objetivo de design, visualizando o gráfico que representa a fórmula e o tipo de função, demonstrando como cada objetivo é pontuado (ou avaliado).

Os problemas e falhas recorrentes, descritas anteriormente na secção de uma variável de jogo, repetem-se aqui, na secção de objetivos de design, pois o funcionamento é igual.

Desde o facto de se ter a tendência para clicar primeiro no botão de editar ou apagar antes de seleccionar o objetivo pretendido; assim como a falta de possibilidade de alterar o objetivo de design expandido.

CrowdPlay New Experiment **Design Problem** Test Setup Results Logout

Game Variation [Add/Remove]

Design Goals [Add/Remove]

Optimization	Target Experience Quality	Test Condition	Test score	Score Function
<input checked="" type="checkbox"/> ON	Selects	0.0 < Selects < 200.0	100.0	triangle
<input checked="" type="checkbox"/> ON	Cenas	0.0 < Cenas < 100.0	100.0	triangle
<input checked="" type="checkbox"/> ON	GameEndExit	0.0 < GameEndExit < 3.0	100.0	triangle
<input type="checkbox"/> OFF	GameEndWin	0.0 < GameEndWin < 2.0	100.0	triangle
<input type="checkbox"/> OFF	GameEndLose	0.0 < GameEndLose < 3.0	100.0	triangle

Design Goal details

Name

Type

Score Function

Increment

Description

Score

Penalty

Figura 3.5 - Ecrã *Design Problem* com a secção *Design Goal* expandida.

Nesta secção, abre-se um *popup* para seleccionarmos uma fórmula (figura 3.6), e ainda editar, apagar ou criar. Para criar ou editar uma fórmula abre um novo *popup* como podemos ver na figura 3.7. Há que referir que uma fórmula final (figura 3.5), pode ser um cálculo entre várias e funciona de forma semelhante aos cálculos do excel, sendo possível tanto escrever o nome da fórmula como seleccioná-la.

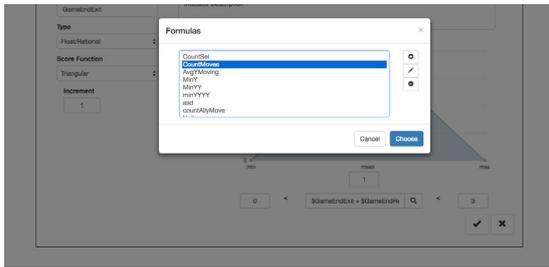


Figura 3.6 - *Popup* para selecionar a fórmula.

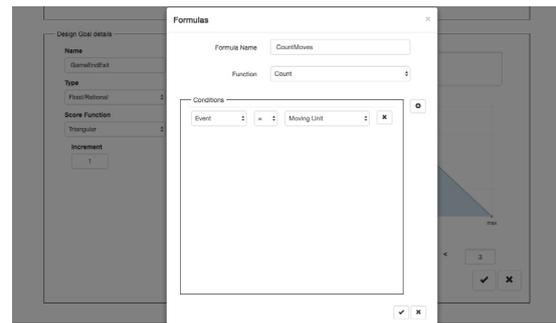


Figura 3.7 - *Popup* para adicionar/editar fórmula.

No ecrã de *Test Setup* (figura 3.8) podemos definir parâmetros gerais do teste, desde o número de soluções, ao número de sessões de cada solução e ainda a condição para terminar a otimização. É neste ecrã que iniciamos, pausamos ou paramos o teste, podendo ver o seu estado atual, sendo visível um resumo do problema de design estabelecido.

CrowdPlay New Experiment Design Problem **Test Setup** Results Logout

Number of Solutions: Current State: Setup Running Completed Type:

Minimum Play Sessions:

End Condition details

End Condition:

Target Score:

Genetic Algorithm details

Generation Algorithm: Selection:

Crossover Type: Mutation Type:

Crossover Rate: Mutation Rate:

Design Problem Summary

Artifact Qualities		Design Goals		
Game Variation	Test Range	Target	Test Condition	Test score
Life Lost Stopped	1.0 < Life Lost Stopped < 3.0	Selects	0.0 < Selects < 200.0	100.0
Enemy Life Giving	5.0 < Enemy Life Giving < 50.0	Cenas	0.0 < Cenas < 100.0	100.0
Moving Life Lost	-3.0 < Moving Life Lost < -1.0	GameEndExit	0.0 < GameEndExit < 3.0	100.0
PlayerUnits_Weapon Damage	5.0 < PlayerUnits_Weapon Damage < 50.0	GameEndWin	0.0 < GameEndWin < 2.0	100.0
Spice Quota	10000.0 < Spice Quota < 100000.0	GameEndLose	0.0 < GameEndLose < 3.0	100.0
IX_HitPoints	0.0 < IX_HitPoints < 100.0			
Tank_Weapon Damage	0.0 < Tank_Weapon Damage < 100.0			

Clear game's data
After Crash
Start
Save Project

Figura 3.8 - Ecrã *Test Setup*.



Figura 3.9 - Ecrã Results.

O ecrã Results (figura 3.9) permite ver toda a informação, a partir de quatro gráficos distintos (*pie chart*, gráfico de barras, histograma e tabela), sendo também possível escolher a vista de um, dois ou quatro desses gráficos (figuras 3.10 e 3.11). Para seleccionar os dados a serem mostrados, é necessário seleccionar as variáveis para o *x* e para o *y*, podendo ainda acrescentar mais variáveis para o *y* (figura 3.11).

Um problema nesta página, para além da complexidade de possíveis variáveis a serem apresentadas, é o facto de, para se seleccionar o número de vistas ser necessário apenas um clique, enquanto para seleccionar o tipo de gráfico é com a função de arrastar (*drag and drop*). Dado que estes botões se apresentam com o mesmo estilo, o utilizador assume que funcionam do mesmo modo. O sistema deve ser coerente e consistente com as diferentes funcionalidades e componentes de um sistema.

Um outro problema deve-se à falta da opção de editar os dados seleccionados para visualizar no gráfico, sendo que, para o fazer é necessário voltar a arrastar o gráfico. Isto vai contra a regra que permite ao utilizador a fácil reversão das ações e/ou recuperação de erros.

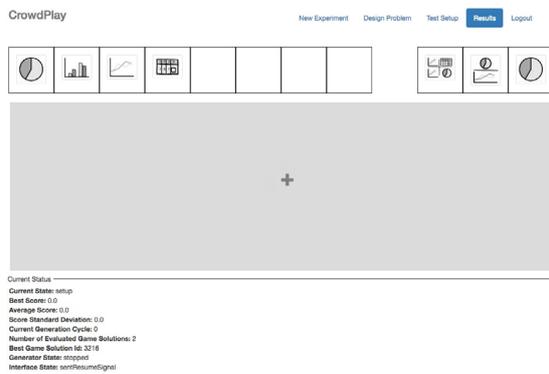


Figura 3.10 - Ecrã *Results* com uma vista.

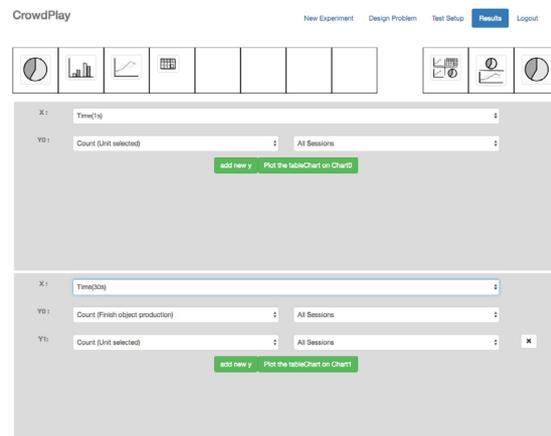


Figura 3.11 - Ecrã *Results* com duas vistas.

No ecrã *New Experiment* (figura 3.12) é possível gerir as permissões dos utilizadores a um projeto assim como adicionar um novo utilizador. Esta é uma página destinada ao administrador pelo que sofreu alterações nesse sentido, criando contas distintas para diferenciar um utilizador comum de um administrador (e ainda a do jogador, que foi adicionada).

CrowdPlay

New Experiment Design Problem Test Setup Results Logout

Prototype dunegen - Dune x 1000

Name Name

User Permissions

- asd
- myname
- Luis
- jfms
- Jorge Vieira
- Jorge Vieira
- Joao Ferreira
- luis
- luis
- Mariana
- Ana
- ana
- Andre
- Nuno Barreto
- elisabete

Server Access user credentials for this Project

Login Login

Password Password

☑ ☒

Figura 3.12 - Ecrã *New Experiment*.

Esta plataforma pode ser vista a partir do endereço crowdplay.dei.uc.pt/login.

3.5 Casos de uso

A aplicação web Crowdplay pode ser usada de três modos diferentes, consoante o objetivo ou experiência do autor (designer/programador). Aos três modos distintos de usar a plataforma podemos designá-los de casos de uso, e, são os seguintes:

- **Testar Jogo;**
- **Explorar Variações de Jogo;**
- **Otimizar Experiência de Jogo.**

Numa primeira fase a plataforma pode servir para **testar** o jogo, por outra palavras, para analisar ou avaliar, explorando os resultados. Este é o processo mais simples, não sendo automatizado e sem recurso a geração procedural. Pode ser utilizado para facilitar a geração de novos artefactos com combinações de características incomuns - contando que haja um conjunto de experiências que os designers procuram.

Já com a utilização do AGE (*Authorial Game Evolution*), podemos usar a aplicação web para **explorar** as variações de jogo ou para **otimizar** experiência de jogo.

No primeiro uso definimos as variações possíveis de jogo e exploramos e/ou analisamos os resultados. Neste caso é facultativo definir os objetivos de design e é usado para a produção de conteúdos - de níveis, mundos, missões - tomando um protótipo existente e depois variando as características até que certas tipologias experienciais sejam mediadas por jogadores.

No caso de uso de **otimizar** experiência de jogo, não só definimos as variações de jogo como também os objetivos de design, para que com recurso ao AGE e de acordo com os dados inseridos, a plataforma otimize e melhore o jogo equilibrando e afinando as qualidades/características.

Podemos associar cada um destes casos de uso a um tipo de autor. Não que esta associação seja direta, ou mesmo implícita, mas de algum modo cada autor está mais familiarizado, ou seja, associado a um caso de uso.

Testar Jogo - Designer de interação

Explorar Variações de Jogo - Designer

Otimizar Experiência de Jogo - Programador

No diagrama seguinte (figura 3.13) apresenta-se mais detalhadamente estes casos de uso.

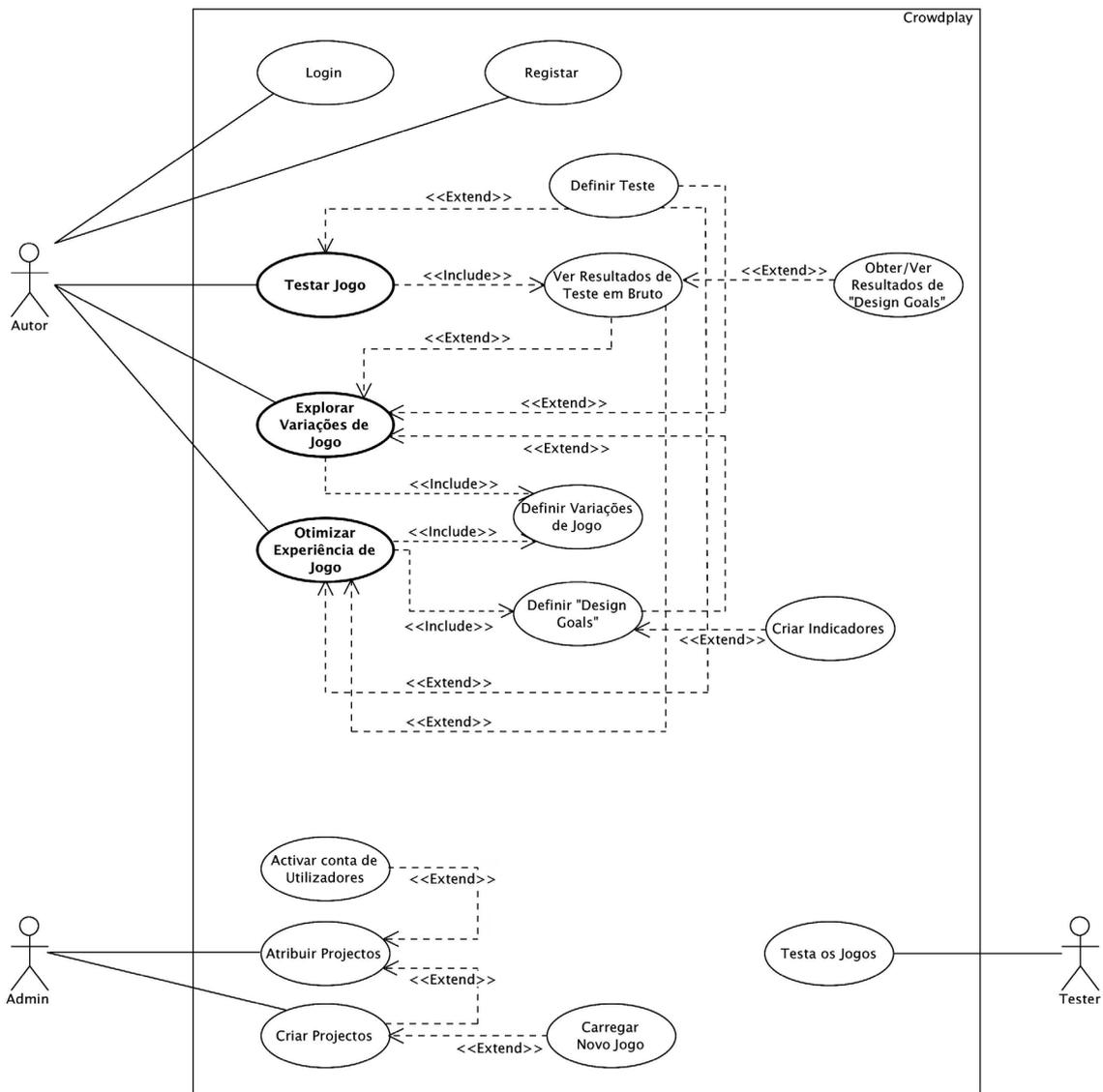


Figura 3.13 - Diagrama de Casos de Usos.

Como podemos ver no diagrama, existem três tipos de utilizadores que interagem com a aplicação; o **autor**, o **admin** e o **tester**. Relativamente ao autor, este pode ser, como já referido em cima, um designer, um programador e/ou um designer de interação, no entanto como cada um pode usar o Crowdplay em qualquer um dos casos, não fazemos distinção entre eles.

Ao iniciar, o **autor** pode fazer *login*, ou caso ainda não tenha conta, pode-se registar; esta ação requer autorização por parte do *admin* (ativar conta de utilizadores).

Em **testar** o jogo, não existe a necessidade de definir os parâmetros de variações de jogo e de objetivo de design, pois neste cenário apenas interessa visualizar os resultados em bruto, podendo ainda, posteriormente, observar mais detalhadamente filtrando os dados.

No cenário em que **exploramos** as variações de jogo, definimos os atributos variáveis (obrigatório), e opcionalmente podemos definir também os *design goals*. Com este cenário conseguimos gerar diferentes variações do jogo para, a partir da observação dos resultados, tirarmos conclusões.

Já no cenário de **otimizar** a experiência de jogo, necessitamos obrigatoriamente de definir tanto as variações de jogo como os *design goals*. Estes dois parâmetros neste cenário são obrigatórios para que a plataforma consiga proceder à otimização dos jogos candidatos. Nestes dois últimos casos temos a possibilidade de ver os resultados em bruto e posteriormente a opção de os visualizar mais detalhadamente pelos *design goals*.

Em qualquer um dos três casos, o **autor** tem a possibilidade de definir um teste, ou seja, sempre que um jogo é carregado podemos definir as características do teste, ou definir um novo teste para um jogo já carregado (um mesmo jogo pode ter vários testes com características diferentes).

O **admin** pode criar projetos, e caso o jogo ainda não esteja na plataforma pode carregar um novo. É também trabalho do **admin**, atribuir projetos aos utilizadores (autores), assim como ativar a conta a um utilizador novo.

O **tester**, assim como o nome indica, testa os jogos, ele não tem conhecimento de qual o método a ser usado, simplesmente joga.

3.6 Testes de Usabilidade

Durante o decorrer da dissertação do aluno João Soares foram desenvolvidos testes de usabilidade nos quais foram encontrados alguns problemas, sendo que apenas uma parte foi resolvida.

Exemplo disto são as tarefas de editar e criar novas variações de jogo e objetivos de design, que se traduzem nas tarefas principais mais complexas e complicadas por parte dos utilizadores. Nomeadamente, as tarefas como definir o intervalo de uma variação de jogo ou editar uma fórmula nos objetivos de design foram, sem dúvida, as tarefas onde surgiram mais dúvidas e por vezes sem sucesso a concluir. A tarefa de definir o intervalo de uma variação de jogo foi corrigida, no entanto o editor de fórmulas, onde os utilizadores sentem grande dificuldade, não sofreu alterações. Referimos ainda alguns dos restantes problemas e erros encontrados na seguinte lista:

- tentativa de editar gráfico diretamente da tabela;
- tentativa de criar uma fórmula diretamente na caixa de texto;
- não identificação do botão para criar uma nova fórmula;
- adicionar condições desnecessárias na fórmula;
- criação de uma fórmula errada.

Muitos problemas foram encontrados com estes testes de usabilidade, no entanto ao analisá-los surgem dúvidas acerca do que foi realmente corrigido. Tendo isto em conta, e para uma análise mais correta, optou-se por fazer novos testes de usabilidade à aplicação.

Foram então preparados testes de usabilidade para uma análise focada nos verdadeiros problemas de usabilidade. Para isso pensou-se em desenvolver estes testes com utilizadores que não estejam familiarizados com o Crowdplay, para que estes não conheçam as suas “peculiaridades” e não influenciem os resultados. Para a seleção dos *testers*, são critérios de inclusão, ter conhecimentos sólidos no uso do computador, e preferencialmente ter algum conhecimento na área de desenvolvimento de jogos, para uma melhor compreensão do funcionamento base do Crowdplay. Idealmente num teste com cinco utilizadores, dois ou três teriam conhecimentos em desenvolvimento, quer como programador quer como designer de jogos.

Há ainda que referir a importância da privacidade e do conforto dos utilizadores. É importante clarificar que o que está a ser avaliado é o sistema e não o utilizador, e ainda informá-los que não serão identificados pelos resultados divulgados. Apesar de no questionário se pedir o nome, entre outras informações, a identificação da pessoa não é revelada na exposição dos resultados.

Os testes seguiram a ordem de etapas descrita: primeiramente é dado aos *testers* uma breve explicação do que é o Crowdplay e do seu funcionamento base, ou seja, a documentação para a preparação (anexo A), depois passa-se à fase de realização das tarefas e no final os *testers* respondem a um questionário.

A realização das tarefas seguiu um guião (anexo B), desenvolvido para abranger uma diversidade de funcionalidades e problemas. Foi pensado para testar a fundo alguns dos problemas de usabilidade. Na realização das tarefas foi pedido aos *testers* para pensarem em voz alta, o que ajudou a identificar e reportar dificuldades. As tarefas foram desenvolvidas uma a uma, e foram planeadas 12, algumas com subtarefas (resultando um total de 30). Todas foram temporizadas e foi pedido ao utilizador para reportar a dificuldade da sua execução numa escala de um a cinco. No fim das tarefas principais foi questionado a existência de algum comentário, e foram anotadas algumas observações.

O questionário (anexo C) foi elaborado com base no questionário S.U.S (*System Usability Scale*), usado também nos testes do aluno João Soares. É um questionário que cobre uma variedade de aspetos de usabilidade dos sistemas, tais como a complexidade, a facilidade de usar, a frequência de uso, necessidade de ajuda, entre outros. No entanto, para a realização destes testes de usabilidade o questionário foi adaptado, focando em questões importantes para avaliar o Crowdplay.

Antes de desenvolver os testes de usabilidade com os utilizadores, os mesmos foram analisados. Foi calculado o tempo para cada tarefa e, a partir disso, o tempo máximo expectável, a dificuldade expectável, assim como descrito os eventos que poderiam ocorrer (os erros e problemas esperados). Estes problemas foram analisados a partir de um teste do género do *cognitive walkthrough*, que consiste na análise passo a passo do website.

De seguida apresentamos a lista de tarefas com os erros mais previsíveis:

T1 - Registrar com a seguinte conta no projeto “Dune Nuno”

- Preencher os campos no *Login* antes de clicar em *Register*.

T1.1 - Entrar com a mesma conta no respetivo projeto (“Dune Nuno”)

- O utilizador pode esquecer-se de seleccionar o projeto pretendido

T4 - Adicionar uma variação de jogo no Game Variation

- O utilizador pode não se aperceber que abriu uma nova área.

T4.2 - Seleccionar o gráfico do tipo “Step”

- Dificuldade em perceber ou confundir com outro parâmetro.

T5.2 - Adicionar uma nova fórmula com o nome “N_eventos”

- O utilizador pode não perceber como se adiciona uma fórmula na caixa inicial.

T5.3 - Seleccionar a condição “Event” igual a “Unit Selected” e guardar fórmula

- Dificuldade em entender onde se acrescentam as condições.

T6 - Editar o objetivo de design com o nome “HasWon”

- Não seleccionar o objetivo antes de clicar em editar.

T6.1 - Definir a função de pontuação para o tipo “Triangular Step Right”

- Dificuldade em perceber ou confundir com outro parâmetro.

T6.2 - Definir a pontuação máxima para 200

- O utilizador pode confundir com o intervalo da pontuação.

T6.3 - Alterar a fórmula para calcular metade (“\$HasWon/2”)

- O utilizador pode não perceber que se pode escrever diretamente no campo da fórmula.

T6.4 - Definir que é dada pontuação máxima a partir 60 e guardar as alterações

- Dificuldade em compreender a tarefa.

T7 - Apagar o objetivo de design “QualityTeste”

- Não seleccionar o objetivo antes de clicar em apagar.

T8 - Ir para a página do Test Setup e iniciar o teste

- Dificuldade em encontrar o botão para iniciar o teste, por se encontrar no fim da página, e ainda a possibilidade de confundir com o estado do teste.

T11 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

- O utilizador pode não perceber que o ícone do gráfico é para arrastar.

T12 - Visualizar dois gráficos, escolher o Histograma (Line Chart) e uma Tabela (Data Table)

- O utilizador pode assumir que para visualizar duas vistas seja também por arrastar. Pode ainda não perceber que já abriu uma nova área para o gráfico.

Para além dos problemas encontrados em cada tarefa existem outros nas seguintes páginas:

Na página de Resultados

- Depois de seleccionar os diferentes gráficos não existe indicação de qual foi seleccionado.
- Na visualização dos gráficos os parâmetros seleccionados não são visíveis.
- Para alterar as variáveis para a visualização é necessário seleccionar de novo o gráfico (e recomeçar de novo).

Na página de *Design Problem*

- Os gráficos nas variações de jogo e objetivos de design por vezes não correspondem ao seleccionado (erro de interface)

Depois de definir os tempos e dificuldades expectáveis, bem como, realizar o teste de usabilidade piloto para confirmação, passou-se aos testes de usabilidade em si. Durante o seu desenvolvimento foram feitas gravações de ecrã com áudio, para facilitar e melhorar a análise dos testes. Como já referido, foram seleccionados cinco utilizadores para desenvolver os testes de usabilidade que não estavam familiarizados com o Crowdplay mas sim familiarizados com desenvolvimento de jogos, quer como programador quer como designer. Os utilizadores são estudantes de Mestrado em Design e Multimédia (MDM), Mestrado em Engenharia Informática (MEI) e Licenciatura em Engenharia Informática (LEI), com idades compreendidas entre os 19 e os 28. A tabela seguinte (figura 3.14) representa a lista de *testers*, com indicação da formação e ocupação, se está ou não familiarizado com o Crowdplay e com o desenvolvimento de jogos.

Utilizador	Formação / Ocupação	Crowdplay	Desenvolvimento de Jogos
U1	MDM	×	✓
U2	MEI	×	✓
U3	MEI	×	✓
U4	MDM	×	✓
U5	LEI	×	✓

Legenda:

✓ - Familiarizado com

× - Não familiarizado

Figura 3.14 - Tabela de utilizadores (*testers*).

Os dados do tempo e da dificuldade de cada tarefa foram analisados e comparados com o tempo e dificuldade expectáveis. Para aprofundar a análise, calculou-se ainda a média do tempo/dificuldade dos utilizadores.

As figuras 3.15 e 3.16 representam o tempo e a dificuldade de cada tarefa respetivamente. Nos gráficos, o tempo e a dificuldade expectáveis estão representados pela barra.

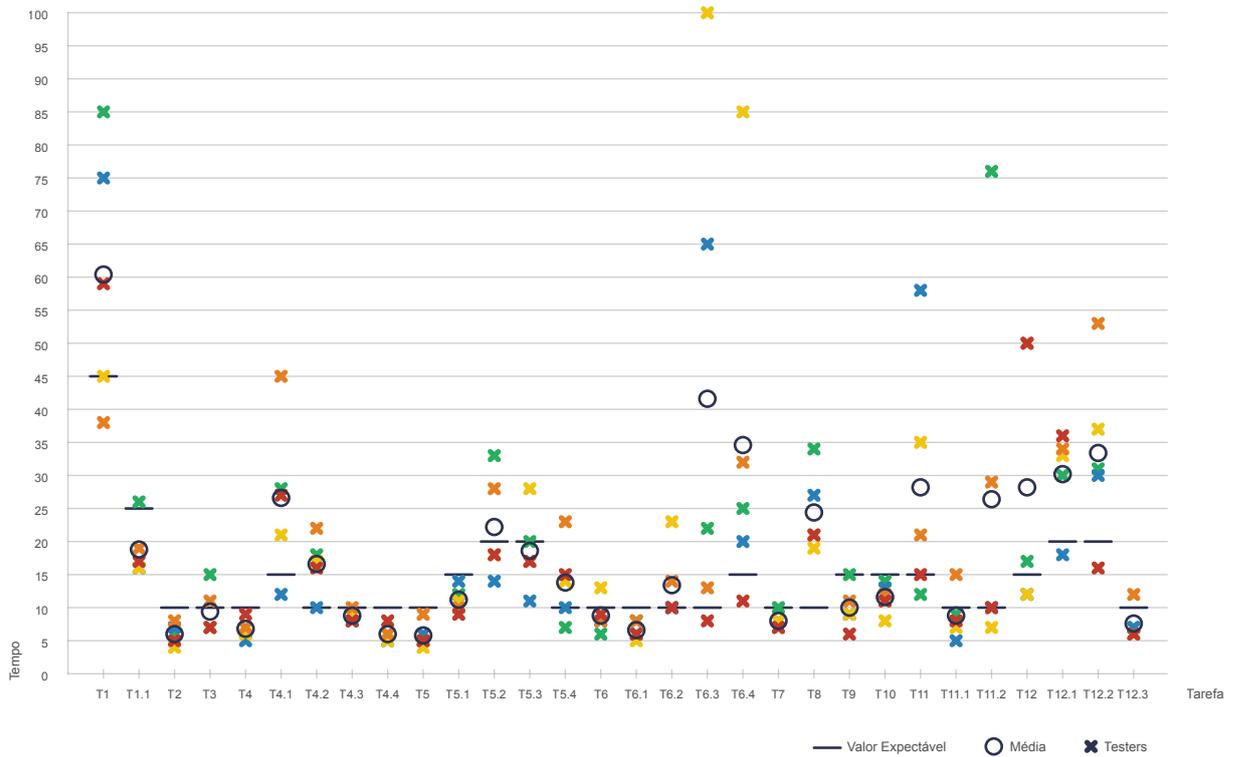


Figura 3.15 - Gráfico do tempo por tarefa.

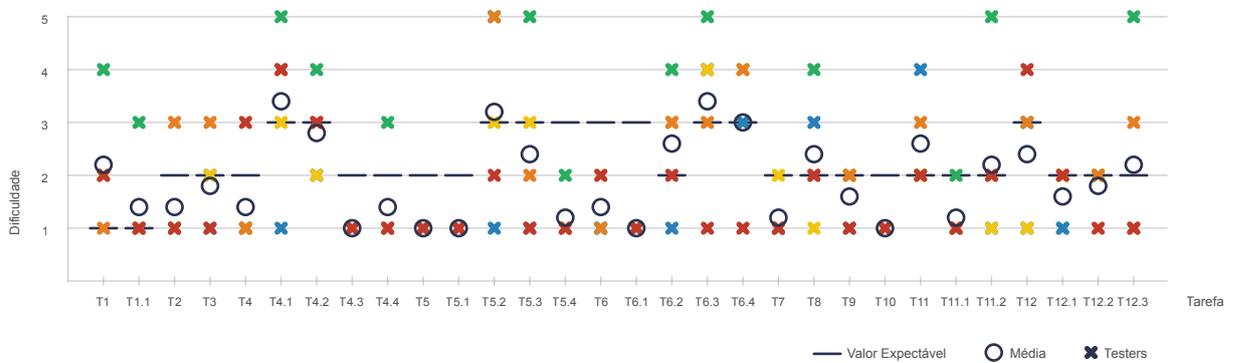


Figura 3.16 - Gráfico da dificuldade por tarefa.

É de referir que estes gráficos foram elaborados com recurso ao *Processing*, cujo código foi desenvolvido com este propósito. Optou-se por usar esta linguagem por permitir desenvolver os gráficos como pretendido e gerá-los de forma automática.

Após a elaboração e análise dos gráficos foi feita uma avaliação posterior a cada tarefa individualmente, na qual foram descritas as ocorrências durante o teste, nomeadamente os problemas e uma possível solução, já a pensar na futura interface. Apresentamos de seguida a listagem de tarefas com a respetiva informação:

T1 - Registrar com a (seguinte) conta no projeto “Dune Nuno”

Nesta tarefa, três utilizadores preencheram os dados no *Login*, sendo que dois deles chegaram a completar os dados todos até perceber que não estavam na área para registar.

Problema: Os dois botões de *Login* e *Register* são iguais.

Solução: Distinguir os botões, dando destaque ao botão de *Login* (quando os campos visíveis são do correspondente) e ter um botão para *Register* distinto.

T1.1 - Entrar com a mesma conta no respetivo projeto (“Dune Nuno”)

Esta tarefa, foi clara e simples para a maioria dos utilizadores, apenas um hesitou e questionou se tinha de voltar a preencher os dados, clicando no botão de *Login* antes de preencher os dados.

Problema: Assim que o registo é feito com sucesso passa automaticamente para a página de *Login*, não sendo muito intuitivo.

Solução: Diferenciar os ecrãs do *Login* e *Register*, ou mostrar os dois no mesmo ecrã.

T2 - Desativar a variação “Soldier_Weapon Damage” nas variações de jogo (“Game Variation”)

Tarefa conseguida com sucesso por parte de todos utilizadores.

T3 - Ativar o teste “HasLost” nos objetivos de design (“Design Goals”)

Nesta tarefa, muito semelhante à anterior, surgiu alguma dificuldade por ser necessário fazer *scroll*.

Problema: A secção dos objetivos de design é muito pouco visível (sem a ação de *scroll*).

Solução: Evitar a necessidade de *scroll*, ou pelo menos tornar mais visível a existência de mais secções.

T4 - Adicionar uma variação de jogo no “Game Variation”

Todos os utilizadores adicionaram a variação sem dificuldade, no entanto, dois não perceberam rapidamente se a tinham realmente adicionado (questionando, se já estava).

Problema: A área para adicionar a variação surge no final da página, de forma pouco visível (sem a ação de *scroll*).

Solução: Evitar a necessidade de *scroll*, ou, ao surgir a nova área do ecrã, centrar-se nela.

T4.1 - Escolher a variável "Radar_Power"

Nesta tarefa surgiram algumas dificuldades, uma surgiu na tentativa de pesquisa, escrevendo no campo inicial (onde é mostrada a variável selecionada). A outra dificuldade surgiu ao encontrar a variável pedida, visto a lista não se encontrar por ordem alfabética. Estas dificuldades ocorreram em 4 utilizadores.

Problema: Apesar da barra de pesquisa estar a *disable* para escrever, os utilizadores tentaram escrever. O modo para seleccionar a variável é pouco intuitivo, a expansão e o clique não funcionam sobre o texto (só dentro da *checkbox*). E a listagem não se encontra ordenada alfabeticamente.

Solução: Pesquisar mais intuitivo, com possibilidade de pesquisar a partir da escrita e com o sistema de seleção mais intuitivo e ordenado alfabeticamente (como por exemplo, do género de pesquisa e navegação entre pastas).

T4.2 - Selecionar o gráfico do tipo "Step"

Todos os utilizadores sentiram alguma hesitação, pois não associaram o "Probability Function" ao tipo de gráfico e alguns confundiram com o "Type" (Tipo de conjuntos numéricos).

Problema: Terminologia pouco esclarecedora e ainda o facto de não existir ligação do gráfico com o campo para seleccionar o tipo.

Solução: Corrigir a terminologia e tornar intuitivo a escolha do gráfico com algum tipo de associação ao "Probability Function".

T4.3 - Definir para variar entre 100 e 200

Tarefa conseguida com sucesso por parte de todos utilizadores.

T4.4 - Com um incremento de 5 e guardar a variação de jogo

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5 - Adicionar um objetivo de design no "Design Goals"

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5.1 - Com o nome "QualityTeste"

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5.2 - Adicionar uma nova fórmula com o nome "N_eventos"

Nesta tarefa surgiram muitas dúvidas, desde a tentativa de apenas escrever a fórmula no campo de texto inicial. Posteriormente, ao clicar em pesquisar (na lupa), os *testers* perceberam que tinham errado, e que ainda não tinham adicionado a fórmula. Ao adicionar a fórmula surgiram mais dúvidas e hesitações sobre se realmente estavam a fazer bem e se tinham concluído a tarefa.

Problema: Opção para adicionar nova fórmula só visível depois de clicar em pesquisar as fórmulas.

Solução: Tornar claro e distinguir a área para selecionar as fórmulas da área para adicionar nova fórmula.

T5.3 - Selecionar a condição "Event" igual a "Unit Selected" e guardar fórmula

Nesta tarefa as dúvidas e hesitações mantiveram-se, no entanto foi mais simples para a maioria dos utilizadores.

Problema: Toda a área para editar, adicionar ou definir as fórmulas é confusa devido às próprias funcionalidades e a todo o processo.

Solução: Reformular todo o processo relacionado com as fórmulas, para melhorar a compreensão de como funciona na prática e assim facilitar todo o processo.

T5.4 - Selecionar essa fórmula e guardar o objetivo de design

Esta tarefa foi fácil por parte de todos os utilizadores, no entanto, ao questionar se tinham percebido o que tinham feito, surgiram dúvidas.

Problema: Não se torna fácil perceber o que na prática acontece, desde o início do processo das fórmulas.

Solução: Reformular toda a secção de editar, adicionar ou definir fórmulas, para a tornar mais intuitiva, tanto na interação como na compreensão do que realmente acontece na prática.

T6 - Editar o objetivo de design com o nome "HasWon"

Dois dos utilizadores clicaram no botão para adicionar antes de selecionar o objetivo.

Problema: Pouco intuitivo de que é possível selecionar cada objetivo.

Solução: Alterar o modo de editar, tornando-o mais simples e apenas com um único clique.

T6.1 - Definir a função de pontuação para o tipo “Triangular Step Right”

À semelhança do que aconteceu na tarefa T4.2, houve hesitações na procura pelo campo para alterar o tipo de gráfico, no entanto os utilizadores já não confundiram com o “Type” (Tipo de conjuntos numéricos).

Problema: Terminologia pouco esclarecedora e o facto de não haver ligação do gráfico com o campo para seleccionar o tipo (“Score Function”).

Solução: Corrigir a terminologia e tornar intuitivo a escolha do gráfico com algum tipo de associação do gráfico ao “Score Function”.

T6.2 - Definir a pontuação máxima para 200

Nesta tarefa, 4 utilizadores inicialmente confundiram com o limite máximo do intervalo, ao olhar com mais atenção, perceberam o erro e corrigiram.

Problema: Os diferentes campos não estão bem identificados e não se percebe o que realmente implicam na realidade.

Solução: Tornar os diferentes campos visualmente perceptíveis do que implicam realmente na prática.

T6.3 - Alterar a fórmula para calcular metade (“\$HasWon/2”)

Surgiram dificuldades e hesitação por parte da maioria dos utilizadores. Dois utilizadores tentaram editar a fórmula, não percebendo que se podia escrever no campo inicial. Todos os utilizadores ao concluir a tarefa mostraram algumas dúvidas sobre se a tinham concluído e alguns assumiram ter chegado lá a partir da dica “\$HasWon/2”.

Problema: Pouco intuitivo de que se pode escrever no campo e que realmente funciona. Não existe nada que verifique se a fórmula está bem escrita.

Solução: Reformular a secção, dando *feedback* e maior intuitividade ao definir as fórmulas.

T6.4 - Definir que é dada pontuação máxima a partir 60 e guardar as alterações

Nesta tarefa era pedido alguma interpretação por parte dos utilizadores. Ao que dois utilizadores alteraram o mínimo do intervalo, e outros dois hesitaram bastante entre o mínimo do intervalo e o *mean*, acabando por concluir a tarefa, mas sem entender se o estavam a fazer correctamente.

Problema: Terminologia pouco esclarecedora; os diferentes campos não estão bem identificados e não se percebe o que realmente implicam na realidade.

Solução: Corrigir a terminologia e tornar os campos intuitivos e mais perceptíveis do que implicam realmente na prática.

T7 - Apagar o objetivo de design "QualityTeste"

Tarefa conseguida com sucesso por parte de todos utilizadores. No entanto houve comentários como "já sabia que tinha que selecionar o objetivo antes, se não clicava primeiro no " - " (sinal menos)". Mesmo problema sucedido na tarefa T6.

Problema: Pouco intuitivo para o facto de que é possível selecionar cada objetivo.

Solução: Alterar o modo de apagar, tornando-o mais simples e apenas com um único clique ou movimento.

T8 - Ir para a página do Test Setup e iniciar o teste

Nesta tarefa as dúvidas surgiram ao iniciar o teste. Foi fácil ir para a página mas houve confusões com os estado do teste, e dificuldades em encontrar os botões no final da página.

Problema: Botões pequenos e no fim da página.

Solução: Tornar os botões mais visíveis.

T9 - Pausar o teste e mudar o tipo de teste para "Sample"

Tarefa conseguida com sucesso pela maioria dos utilizadores. Apenas um sentiu dificuldade em encontrar o campo para alterar o tipo de teste.

T10 - Reiniciar o teste e Para-lo

Tarefa conseguida com sucesso por parte de todos utilizadores.

T11 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

Nesta tarefa surgiu a dificuldade esperada, nomeadamente, a de os utilizadores não perceberem que era para arrastar. Os utilizadores tentaram clicar várias vezes e inclusive houve quem experimentasse clicar no "+".

Problema: *Popup* de ajuda demora a aparecer, e ao tentar sempre com cliques não chega sequer a aparecer. O sinal "+" não é intuitivo de que é para arrastar.

Solução: Tornar a seleção dos gráficos com o arrastar mais intuitivo.

T11.1 - Escolher para o y "Count (Order unit to Attack)"

Tarefa conseguida com sucesso por parte de todos utilizadores.

T11.2 - Escolher para todas as Sessões e visualizar gráfico

Nesta tarefa surgiram algumas dificuldades ao visualizar os dados.

Problema: Terminologia do botão "Plot the BarChart on the Chat0" não intuitiva e não compreendida por parte de todos os utilizadores.

Solução: Diferenciar o botão de "Plot" com o botão de adicionar novo y e corrigir a terminologia.

T12 - Visualizar dois gráficos, escolher o Histograma (Line Chart) e uma Tabela (Data Table)

Nesta tarefa houve dificuldade em perceber como se escolhia as vistas múltiplas para os gráficos, visto que se seleciona a partir do clique, diferentemente do modo para selecionar o tipo de gráfico a partir do arrastar. Houve a tentativa de arrastar o botão das duas vistas, assim como ao explorar um utilizador tentou adicionar os dois gráficos pedidos para a mesma área.

Problema: Incoerência na interação pelo facto de botões idênticos terem interatividades diferentes.

Solução: Reformular toda a área para selecionar as vistas múltiplas e a escolha do tipo de gráfico.

T12.1 - Para o histograma escolher para o x “Session” e para o y “Indicator Value” e “Time per Session”

Esta tarefa foi relativamente simples por parte dos utilizadores, no entanto um problema que surgiu foi o facto de existirem muitas variáveis para procurar e selecionar.

Problema: O *select list* ser muito longo.

Solução: Reformular os *select lists* dividindo por algum tipo de categoria, ou acrescentar algum tipo de procura.

T12.2 - Para a tabela escolher para o x “Time(10s)” e com dois y, um “Unit selected” e outro “Moving Unit” (ambos para a sessão 889)

Nesta tarefa, para além do problema mencionado na tarefa anterior (T12.1), houve dificuldades ao adicionar o segundo y, por parte de dois utilizadores.

Problema: O facto do botão de adicionar novo y ser muito idêntico ao de visualizar os dados.

Solução: Distinguir os dois botões de “Plot” e adicionar um novo y.

T12.3 - Visualizar os dois gráficos

Mesmo problema mencionado na tarefa T12; no entanto nesta tarefa os utilizadores já sabiam como lidar com a dificuldade. Um outro problema que surgiu nesta sub-tarefa, foi o facto de um utilizador ter selecionado os dados em desacordo com o tipo de gráfico (dados para a tabela, selecionados para o histograma, e vice-versa).

Problema: Depois de selecionar os gráficos para cada área, não é identificado qual foi selecionado.

Solução: Reformular toda a área para selecionar as vistas múltiplas e a escolha do tipo de gráfico. E ainda quanto ao erro que surgiu, mostrar qual o tipo de gráfico selecionado.

Posto isto, conclui-se que as tarefas mais problemáticas são as seguintes, tendo em conta também as médias dos tempos que ficaram acima do tempo expectável, bem como as dificuldades percebidas:

- T1 - Registrar com a conta no projeto "Dune Nuno"*
- T4.1 - Escolher a variável "Radar_Power"*
- T4.2 - Selecionar o gráfico do tipo "Step"*
- T5.2 - Adicionar uma nova fórmula com o nome "N_eventos"*
- T5.4 - Selecionar essa fórmula e guardar o objetivo de design*
- T6.2 - Definir a pontuação máxima para 200*
- T6.3 - Alterar a fórmula para calcular metade ("\$HasWon/2")*
- T6.4 - Definir que é dada pontuação máxima a partir 60 e guardar as alterações*
- T8 - Ir para a página do Test Setup e iniciar o teste*
- T11 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras*
- T11.2 - Escolher para todas as Sessões e visualizar gráfico.*
- T12 - Visualizar dois gráficos, escolher o Histograma (Line Chart) e uma Tabela (Data Table)*
- T12.2 - Para a tabela escolher para o x "Time(10s)" e com dois y, um "Unit selected" e outro "Moving Unit" (ambos para a sessão 889)*

Ainda relativamente aos testes de usabilidade com os utilizadores foi pedido para responder ao questionário adaptado de Questionário S.U.S.. Este questionário é composto por sete perguntas com resposta com pontuação de 1 a 5 (1 - discordo totalmente e 5 - concordo totalmente) e duas perguntas de resposta aberta. Quanto às perguntas fechadas com a pontuação de 1 a 5, nem todos estão classificadas pela mesma ordem (exemplo: nem sempre pontuação máxima é o ideal). Os dados são apresentados na figura 3.17, onde o valor expectável indica qual a pontuação mais próxima do ideal (ou 1 ou 5).

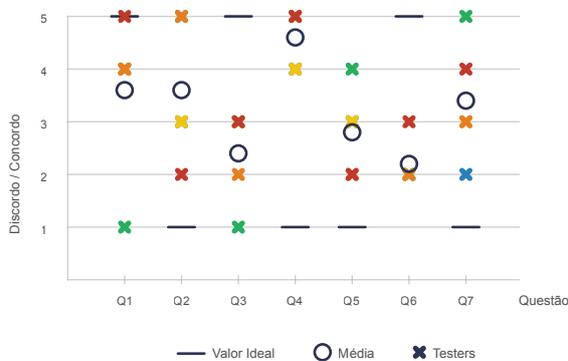


Figura 3.17 - Respostas ao questionário.

Conclui-se então que as questões, cuja resposta em média se afasta mais da resposta teoricamente ideal (dois pontos de diferença), são as seguintes:

Q2 - O sistema é demasiado complexo

Q3 - O sistema é fácil de usar

Q4 - Precisava de aprender algumas coisas antes de começar a usar este sistema

Q6 - A maioria das pessoas iria aprender a usar este sistema muito rapidamente

Q7 - Senti dificuldade em perceber o significado de cada tarefa

Quanto à pergunta de resposta aberta “Qual a tarefa mais difícil ou a maior dificuldade?”, concluímos que as tarefas onde surgiram mais dificuldade são as seguintes:

T4.1 - Escolher a variável “Radar_Power”

T5.2 - Adicionar uma nova fórmula com o nome “N_eventos”

T6.3 - Alterar a fórmula para calcular metade (“\$HasWon/2”)

T11 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

T12 - Visualizar dois gráficos, escolher o Histograma (Line Chart) e uma Tabela (Data Table)

Relativamente a comentários ou sugestões, quatro dos dos cinco utilizadores fizeram observações. Foi descrito que a plataforma não era coerente; não existe uniformidade nos botões; página de *Test Setup* saturada e que deveria adotar um visual mais coerente e apelativo.

Conclusões dos Testes

Com estes testes de usabilidade podemos observar que a interface apresenta problemas significativos. Salientam-se, por exemplo, aspectos de coerência e consistência não preservados, com a existência de botões e/ou componentes muito semelhantes para funções distintas, que induzem os utilizadores em erro. De referir também problemas no que diz respeito à terminologia pouco esclarecedora, à informação pouco visível e/ou clara, questões de acessibilidade e ainda falta de informação de progresso.

Estas questões aqui levantadas vão de encontro às perceções dos próprios utilizadores que referem: “A plataforma não é coerente...”, e “Não há uniformidade nos botões...”, entre outras observações.

Todos os problemas detetados e aqui descritos foram tidos em conta para a sua possível resolução.

4

4.1 Proposta

4.2 Novas Funcionalidades

4.3 Prototipagem de Baixa Fidelidade

4.4 Prototipagem de Alta Fidelidade

Proposta de Design

4. Proposta de Design

Neste capítulo, dividido em quatro subcapítulos, é descrito o processo de desenvolvimento da proposta de design.

Podemos ler sobre a proposta em si (capítulo 4.1), elaborado ainda no decorrer do primeiro semestre, onde é referido o conceito e algumas características do design pensadas para desenvolver a nova plataforma do Crowdplay (utilizado-se o tempo futuro representando a intencionalidade da ação no futuro próximo).

Nos restantes capítulos, elaborados já no decorrer do segundo semestre, são apresentadas as novas funcionalidades do Crowdplay e as primeiras fases do desenvolvimento da interface, da qual fazem parte os protótipos de baixa e alta fidelidade. Faz-se referência ainda a algumas características do design, tais como, a tipografia e a paleta de cores selecionadas para a interface, bem como alguns ícones e imagens.

4.1 Proposta

Como já referimos, a tecnologia ganha mais valor quando proporciona aos utilizadores verdadeiras experiências, em vez de apenas o uso das ferramentas, e esse será o desafio principal a ter em conta. Promover aos utilizadores do Crowdplay uma experiência fácil, intuitiva e divertida, será o objetivo fulcral para o redesign desta plataforma que até agora funciona apenas como uma ferramenta. Para promover a melhor experiência possível, há que entender o público alvo, e como já foi referido, o Crowdplay é usado tanto por programadores como por designers, mas o que eles têm em comum são, sem dúvida, os jogos. Todos os futuros utilizadores estão de algum modo familiarizados com os jogos, quer seja a desenvolvê-los ou simplesmente a jogar, e conhecem bem as suas características de interação. É daqui que nasce o conceito, trazer parte da experiência dos jogos para a ferramenta do Crowdplay, usando características da interação dos jogos e adaptando-as ao Crowdplay.

O conceito não se traduz propriamente numa metáfora ou representação visual, mas sim numa ideia para a interatividade.

Ao falar em características de interação, referimo-nos ao modo como o utilizador interage, desde um simples clique, duplo clique, função de arrastar (*drag and drop*) e ainda teclas e atalhos.

Estas características não são exclusivas dos jogos, podendo ser encontradas em ferramentas de *software* (programas da *Adobe*, por exemplo) ou até no próprio *software* de computador. Exemplos disto são a função de duplo clique para abrir uma pasta nos nossos computadores, ou os inúmeros atalhos com teclas presentes nos programas da *Adobe*. Esta é a nossa prova de conceito, e apesar de não estarmos habituados a encontrar estas características na web, no entanto, apesar do Crowdplay funcionar na web, não deixa de ser uma ferramenta de *software*.

As diversas características da interação serão pensadas e implementadas de acordo com as funcionalidades, de modo a que essa interatividade seja intuitiva. É de esperar que estas diferenças possam dificultar o entendimento inicial de funcionamento, mas com uma ajuda inicial, pode-se tornar fácil e intuitivo; desde que seja de fácil aprendizagem e memorização, cumpre o objetivo.

Quanto à representação visual, pretende-se que seja simples e minimalista, reduzindo o *layout* ao essencial, valorizando assim o conteúdo. O design das *dashboards* servirá de inspiração para o redesign do Crowdplay, aproveitando características que valorizem a usabilidade desta plataforma.

Com um design simplista pretende-se minimizar a complexidade da ferramenta, e ainda criar experiências agradáveis em vez de simples artefactos.

As seguintes imagens servem como referências visuais e inspirações (figuras 4.1, 4.2, 4.3 e 4.4).

Nas figuras 4.1 e 4.2 podemos observar dois exemplos com a funcionalidade de arrastar itens, nas quais, mas principalmente a figura 4.1 revela um design bastante agradável e intuitivo.

Na figura 4.3, destaca-se como referência a organização e a paleta de cores; já na figura 4.4 podemos observar uma tabela com um item que se expande, o que pode ser interessante para as variações de jogo e os objetivos de design.

Apesar de na maioria das *dashboards* o menu aparecer na lateral, isto ainda não está definido para o novo design do Crowdplay, e será alvo de experiências e testes no início do desenho da interface.

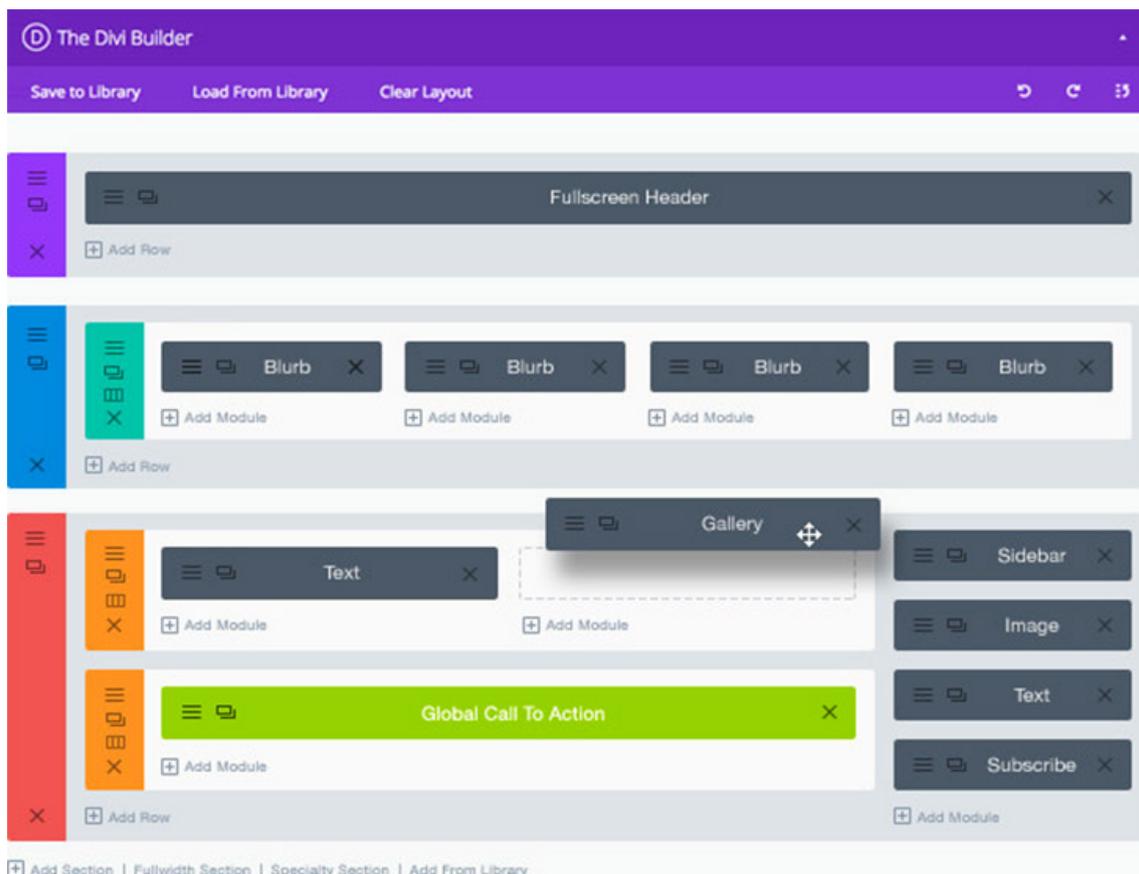


Figura 4.1 - Referência Visual 4. (Premium WP, 2017. *A Drag & Drop Page Builder Plugin for any WordPress Theme*)

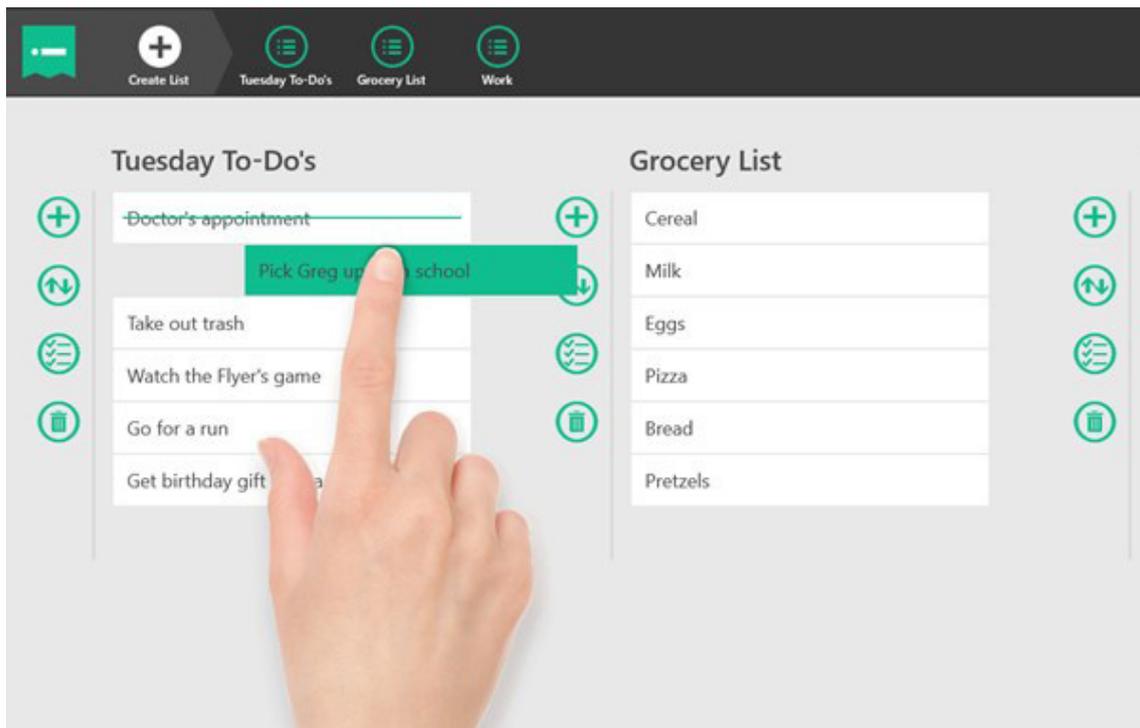


Figura 4.2 - Referência Visual 3. (Dribbble, 2015. *Freebie: Course Dashboard (Sketch)*)

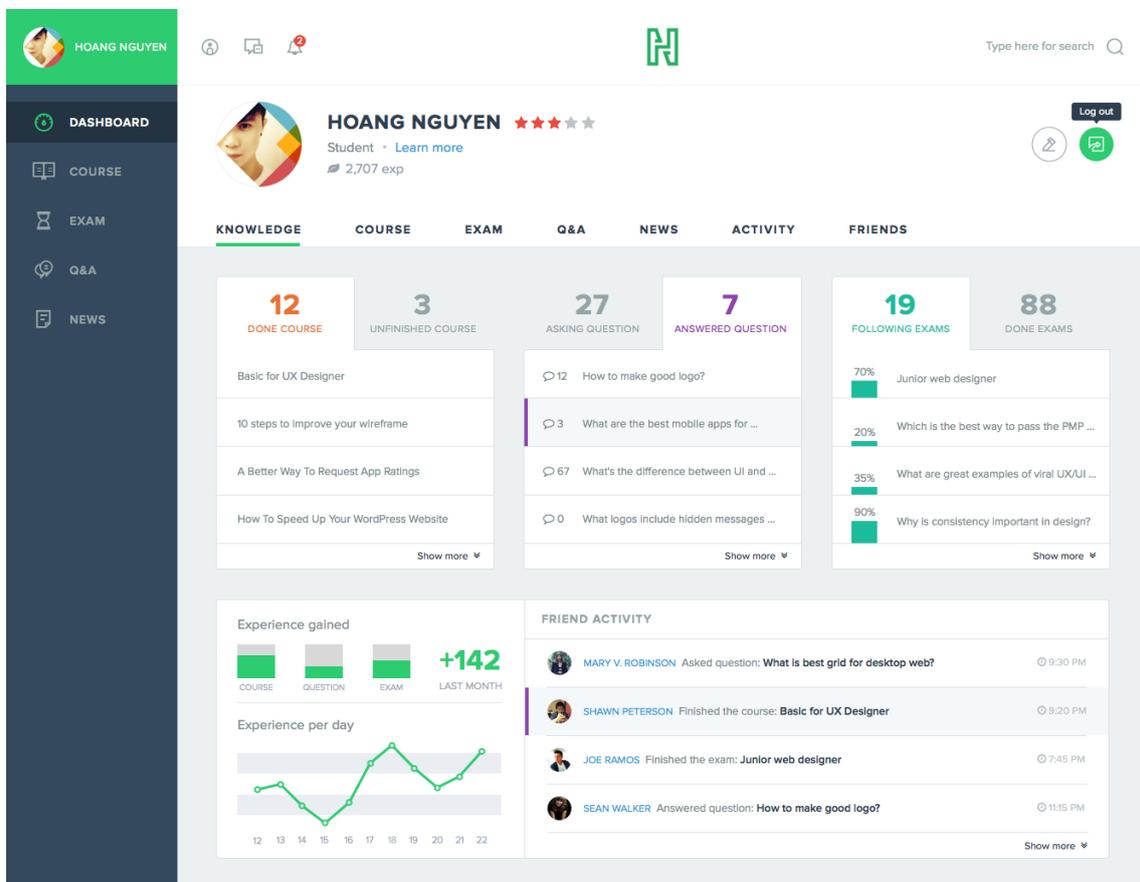


Figura 4.3 - Referência Visual 1. (Dribbble, 2013. *Minimalist is live!*)

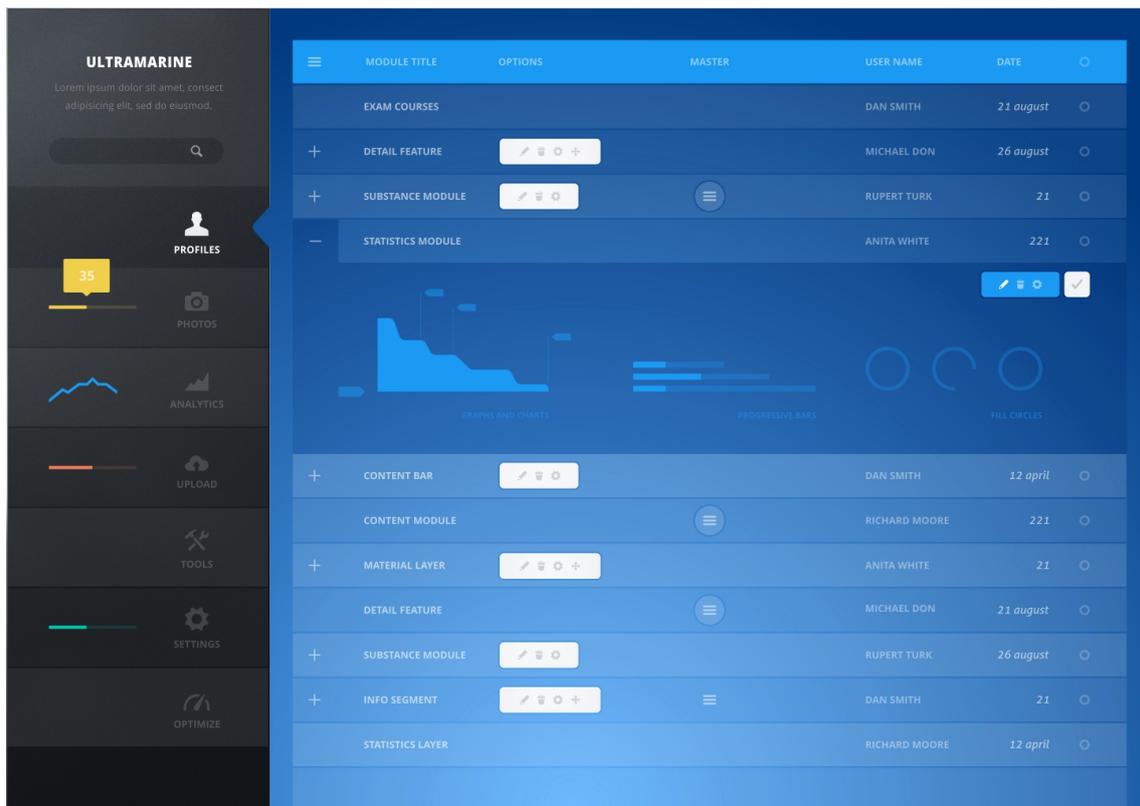


Figura 4.4 - Referência Visual 2. (Dribbble, 2013. *Ultramarine Admin*)

Quanto à tipografia, será escolhida uma fonte não serifada com características humanistas e contemporâneas, tendo em conta o seu funcionamento em interfaces e ecrãs. Nesta seleção destacam-se os seguintes tipos de letras: Open Sans (figura 4.5), PT Sans (figura 4.6) e Source Sans Pro (figura 4.7), que serão testados no desenho da nova interface.

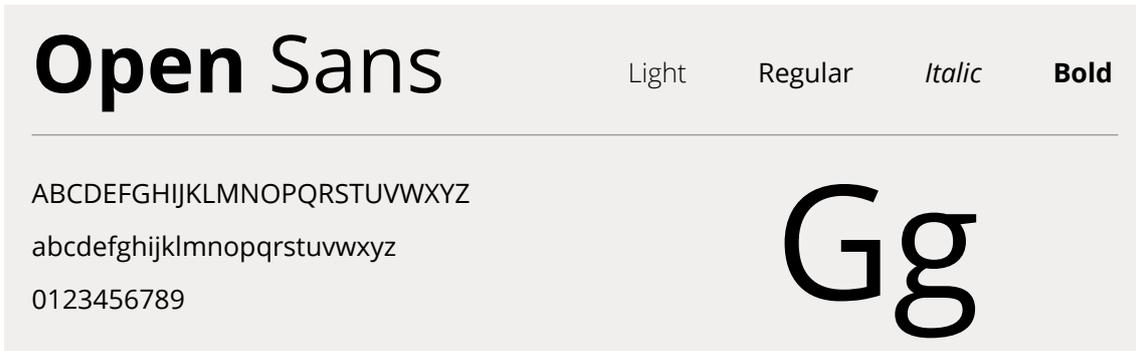


Figura 4.5 - Fonte Open Sans.

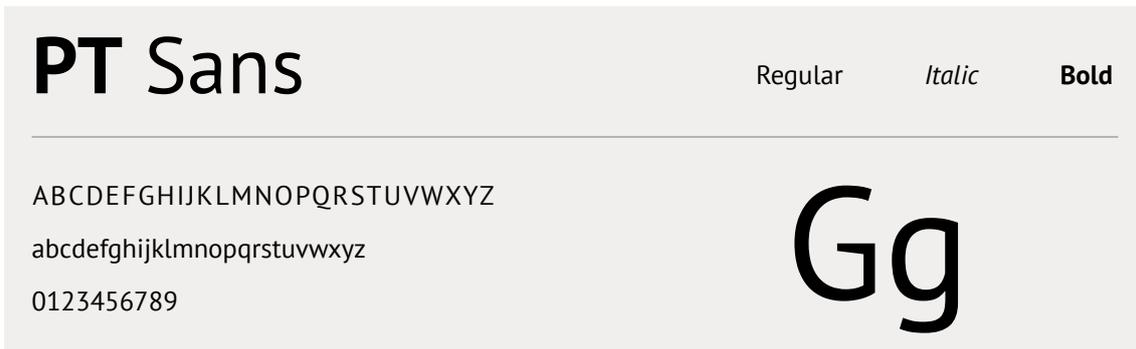


Figura 4.6 - Fonte PT Sans.

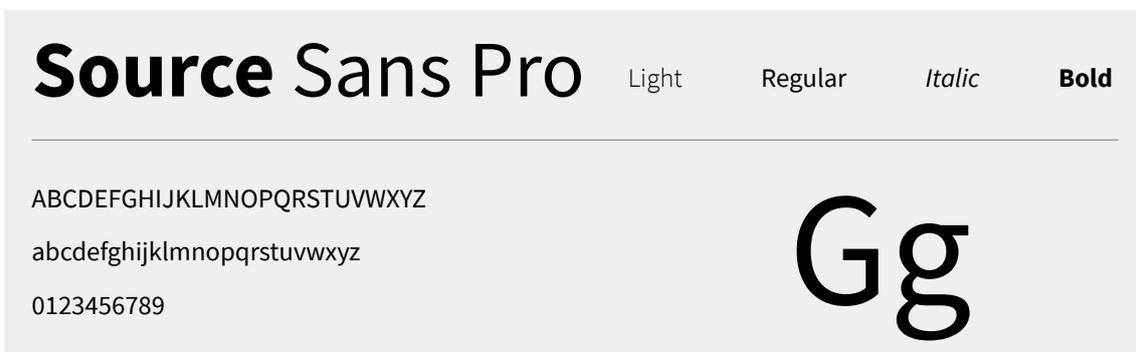


Figura 4.7 - Fonte Source Sans Pro.

Para a paleta de cores pensa-se numa escala de cinzentos claros para a base (*background* e componentes de divisão), contrastando com o tom cinzento escuro ou preto para a informação e uma ou duas cores vivas para destaques. A figura 4.8 mostra possíveis paletas de cores a serem testadas no desenvolvimento dos primeiros *mockups* em computador.



Figura 4.8 - Paleta de Cores.

A figura 4.9 é um *mockup* conceptual de uma abordagem possível para o design da nova interface. Idealmente teríamos lado a lado as variações de jogo e os objetivos de design; na qual, em cada uma destas áreas, teríamos um separador para selecionar a variável ou para selecionar/editar/criar uma fórmula, respetivamente.

Na área inferior teríamos as variações de jogo e os objetivos de design do problema, arrastando para a “caixa” o que desejaríamos submeter ao teste. Esta é apenas uma das ideias a abordar no segundo semestre, pelo que foram ainda pensadas e testadas novas abordagens.



Figura 4.9 - *Layout* Conceptual.

4.2 Novas Funcionalidades

A plataforma Crowdplay foi alvo não só de um redesign, como também de uma reformulação a nível de plataforma, acrescentando algumas funcionalidades e características ao seu funcionamento.

Uma das primeiras significativas alterações, foi relativa aos projetos. Na plataforma anterior para se mudar de projeto era necessário fazer logout e selecionar o projeto pretendido aquando do preenchimento dos dados de *login*. Agora, na plataforma proposta, depois de se fazer *login* com a conta, podemos selecionar o projeto pretendido, podendo sempre alterar de projeto sem se ter que sair da conta.

Como já foi referido anteriormente no capítulo 3 (Avaliação Preliminar), a nova proposta do Crowdplay, terá três modos distintos de efetuar os testes. Apesar de na plataforma anterior ser possível, consoante o que se definia na secção de *Design Problem*, efetuar os diferentes testes, isso não era direto nem visível. Ou seja, o utilizador poderia fazer um teste do tipo **Experiment** sem o ter selecionado ou sem se dar conta, apenas definindo as variações de jogo (sem objetivos de design). Na nova interface, podemos selecionar previamente o tipo de teste que pretendemos efetuar em conjunto com o projeto. Consoante o tipo de teste selecionado, a interface adapta-se dando informação do que se pode definir, ou seja, caso seja selecionado o tipo de teste **Evaluation**, por exemplo, a interface redireciona diretamente para os objetivos de design, mas as variações de jogo ficam bloqueadas. Nos testes do tipo **Experiment** e **Optimize**, em que tanto podemos definir as variações de jogo como os objetivos de design, temos a indicação de quais são obrigatórios ou opcionais.

Resumindo, estas três diferentes formas de testar os jogos referem-se a:

Evaluation - testar um protótipo de jogo único e analisar os indicadores de experiência.

Experiment - criar variações aleatórias do protótipo de jogo e observar os resultados.

Optimize - otimizar o protótipo de jogo para que atinja a experiência desejada. Estes termos foram anteriormente designados por **testar** jogo, **explorar** variações de jogo e **otimizar** experiência de jogo, respetivamente, referidos na secção 3.5 - Casos de uso.

A outra nova funcionalidade refere-se às contas de utilizadores, sendo que na nova proposta podemos distinguir três tipos de utilizadores, o **admin**, o **designer** e o **tester**. Para cada um deles a plataforma alterna, principalmente na secção *Settings*.

O **admin**, tal como o nome indica, é o administrador e tem acesso total; na área *Settings* pode visualizar todos os projetos e todos os utilizadores, podendo atribuir-lhes permissões de acesso dos projetos e vice-versa. Este utilizador pode também adicionar e editar projetos (nome do projeto e nome do jogo), bem como adicionar ou editar utilizadores (nome, email e seleccionar o tipo de conta).

O **designer**, anteriormente designado como autor, é o utilizador comum, e tem acesso à plataforma na íntegra, apenas havendo alterações relativamente à página *Settings*. Este utilizador não tem permissões para apagar projetos, mas pode visualizá-los todos, pedindo permissões para aqueles que não tem. Relativamente à secção *Settings*, o utilizador pode editar os seus dados, desde o nome, email, e *password*, podendo também ver qual o tipo de conta e para que projetos tem permissões. Este utilizador pode ainda ver a listagem de todos os utilizadores registados.

O **tester** é o jogador que irá testar os jogos, pelo que a plataforma para ele contém apenas a área para fazer o *download* dos diferentes jogos (projetos), e a secção *Settings*, semelhante ao do designer.

4.3 Prototipagem de Baixa Fidelidade

A primeira fase de desenvolvimento prático passou pela elaboração dos protótipos de baixa fidelidade, protótipos esses em papel com recurso ao lápis e borracha, uma técnica fácil e rápida para fazer rascunhos, esboçar as ideias, e iterar rapidamente os designs. Esta técnica permitiu estudar e testar o melhor fluxo de navegação, a arquitetura da informação e de um modo geral a sua disposição no ecrã. Esta fase foi também importante para o planeamento das novas funcionalidades, discutindo e analisando o que se poderia implementar no novo sistema. Para tal, recorreu-se a uma metodologia de diálogo constante com o orientador no sentido de se obterem os consensos necessários para validar e iterar o design e as suas funcionalidades. Destas reuniões foram elaborados apontamentos e esboços que serviram de orientação em todo o processo.

Antes de desenvolver alguns esboços, ou *mockups*, foram anotadas as ideias em modo de brainstorming, de que são exemplo as figuras 4.10 e 4.11.

Após esta fase prévia, desenvolveram-se os primeiros protótipos, protótipos de baixa fidelidade, novamente com recurso ao lápis no caderno de esboços, como podemos observar nas figuras 4.12 e 4.13. Para ver mais dos esboços desenvolvidos, ver anexo D.

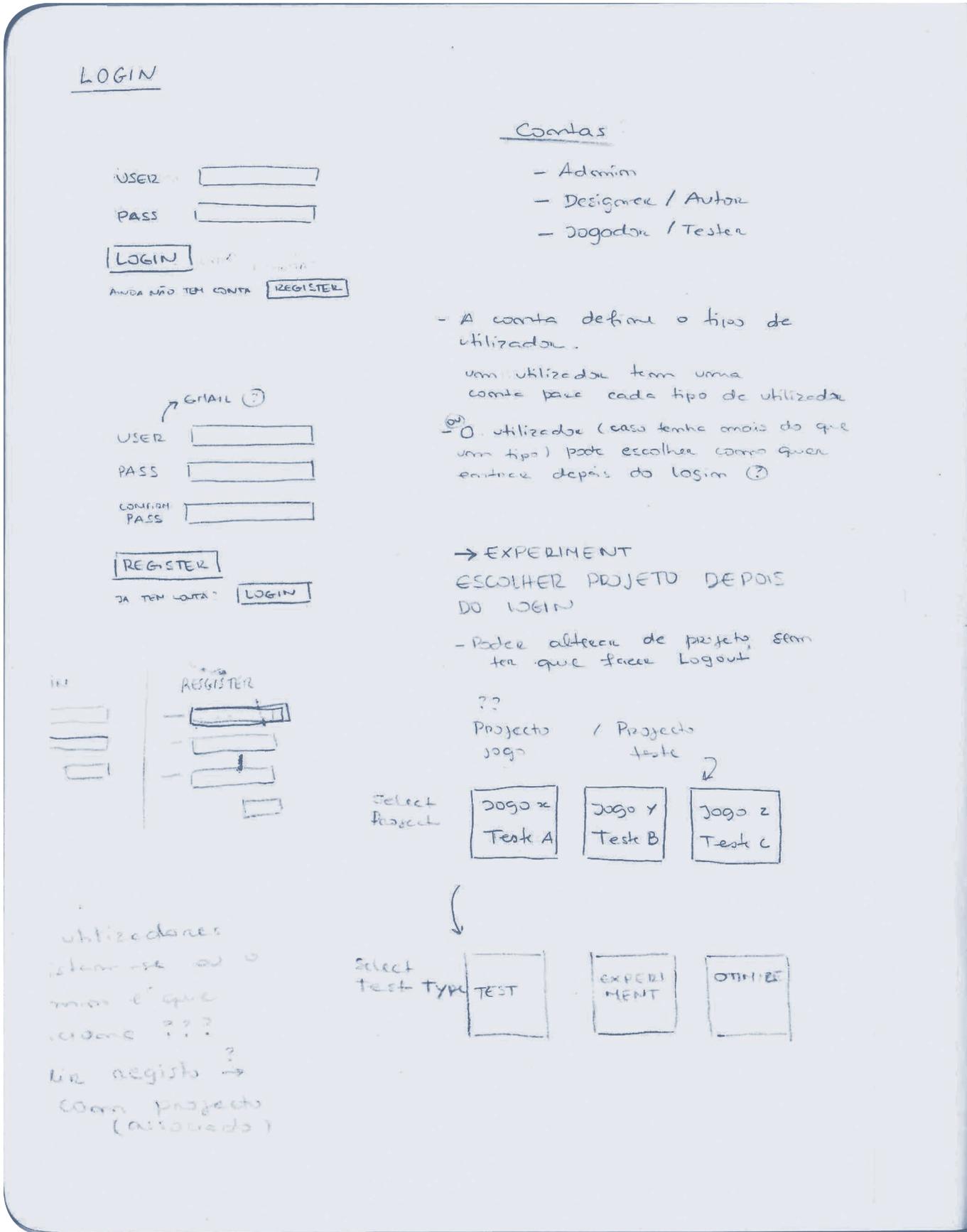
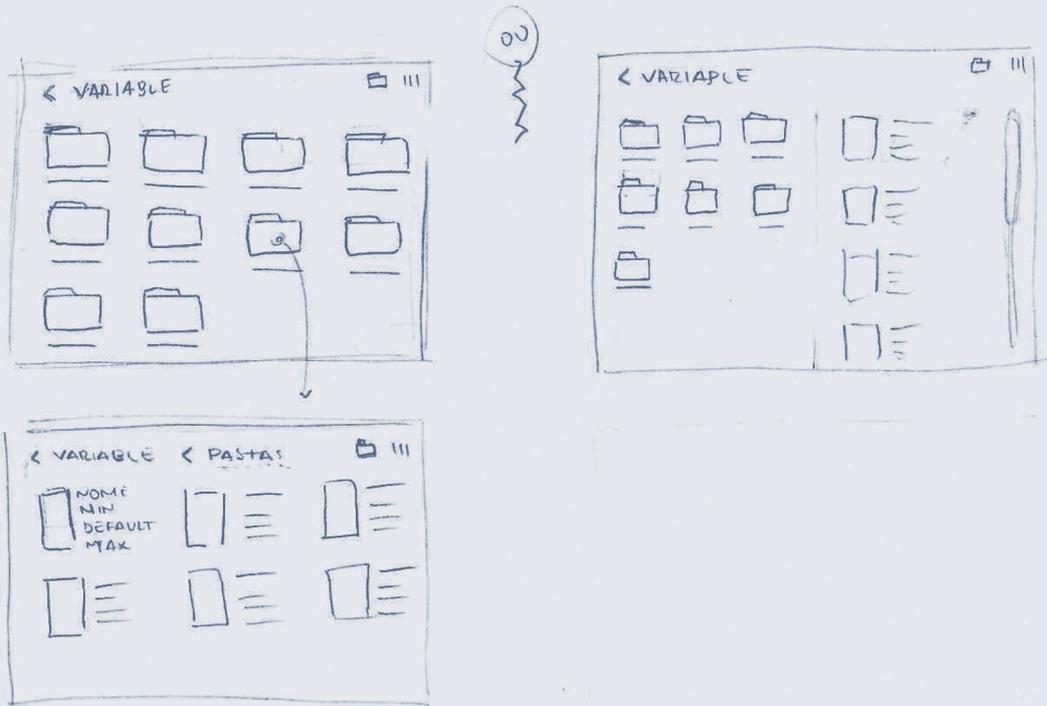


Figura 4.10 - Página do caderno de esboços 1.



EDITAR / APAGAR / ADICIONAR



⊖ Ao abrir uma variável é logo para editar

? - Pode-se apagar esse variável?

GUARDAR / CANCELAR / APAGAR

(ou mais)

GUARDAR / CANCELAR



⊖ Ao abrir uma variável apenas para ver (sem editar)

- EDITAR / APAGAR

↘ Ao clicar em editar a info torna-se editável → cliques desnecessários

↘ Ao apagar volta para a vista das variáveis todas.

Figura 4.11 - Página do caderno de esboços 2.

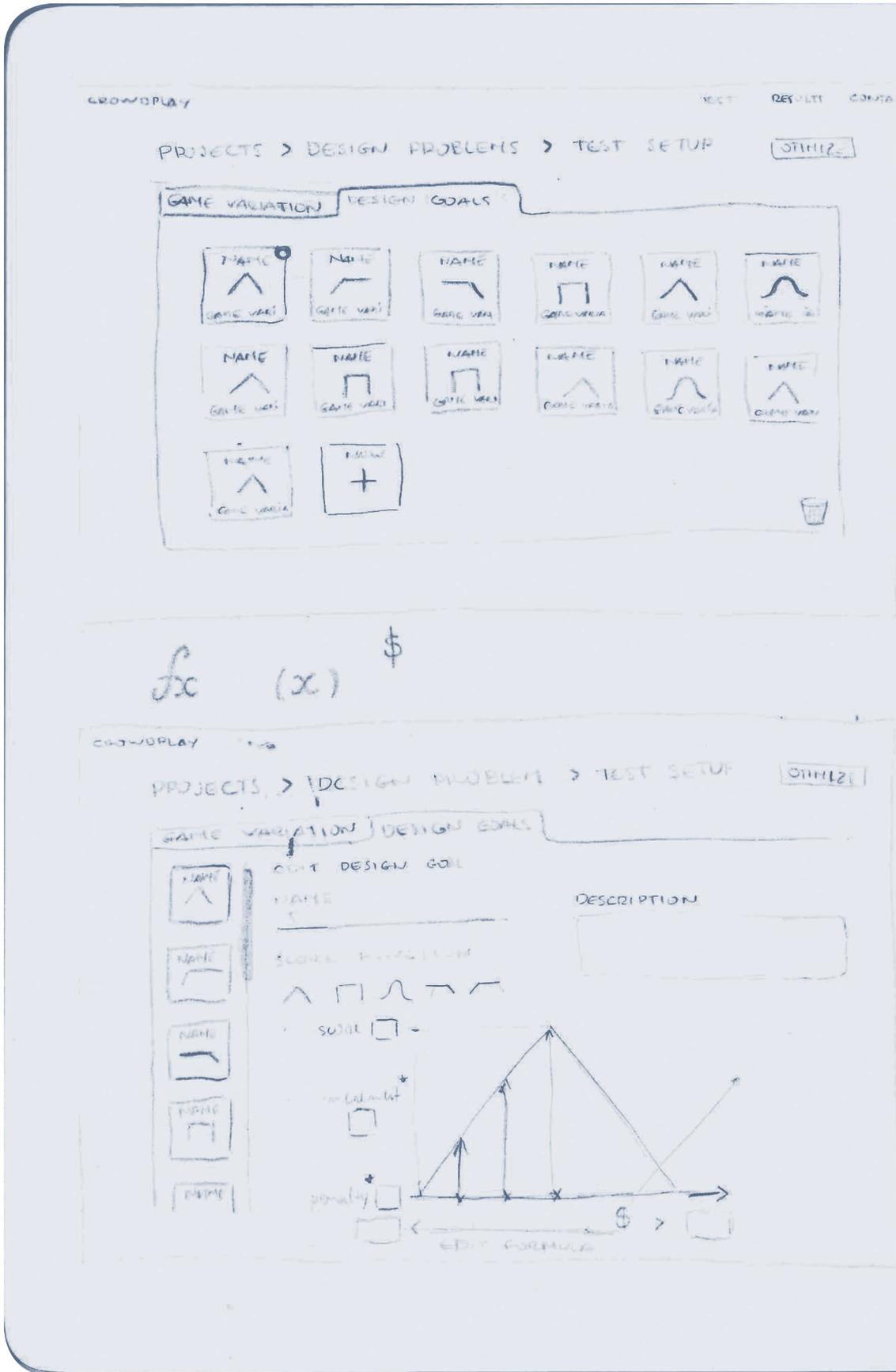


Figura 4.12 - Página do caderno de esboços 3.

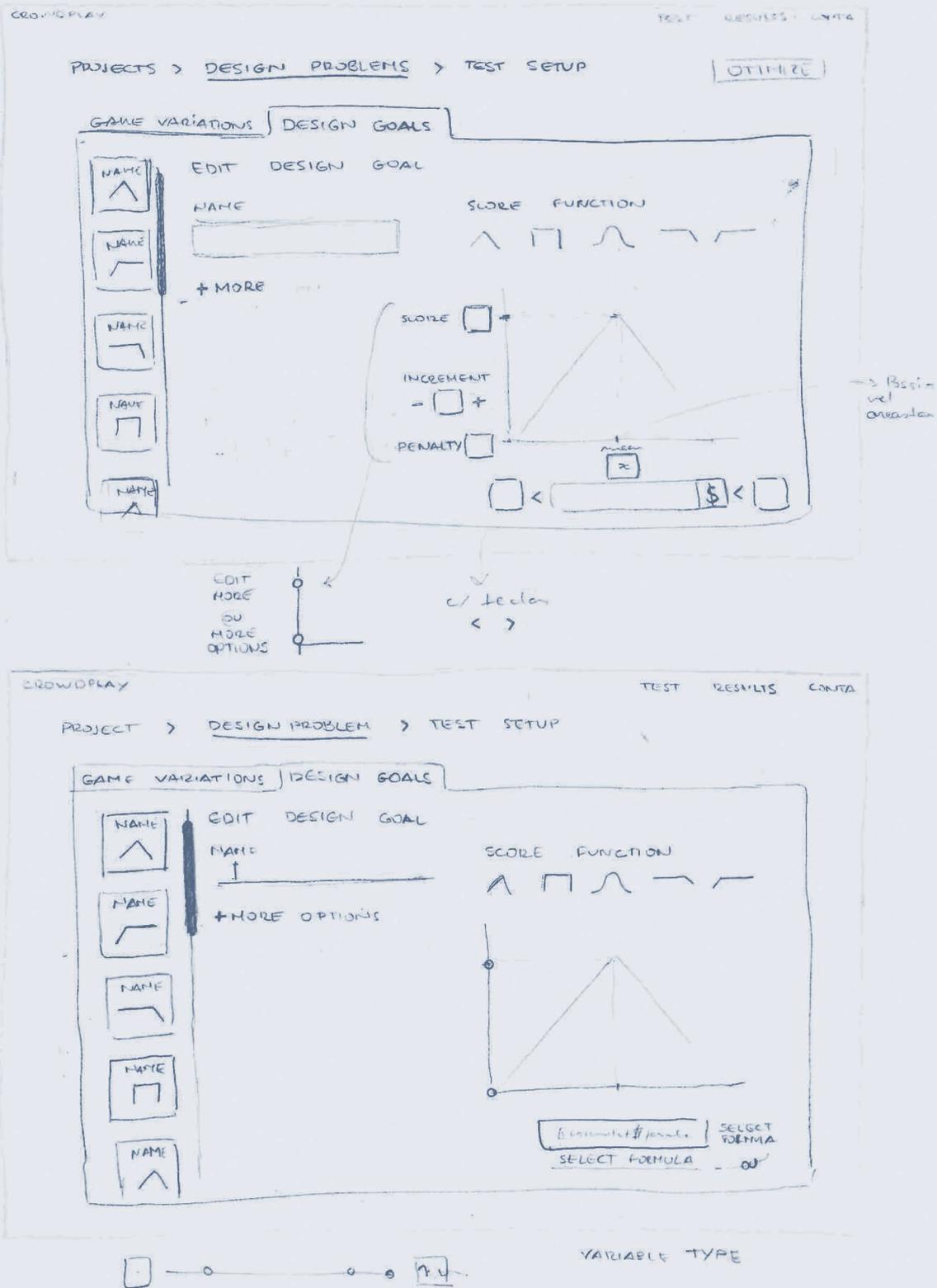


Figura 4.13 - Página do caderno de esboços 4.

4.4 Prototipagem de Alta Fidelidade

Após os esboços e a prototipagem em papel estarem completos, iniciou-se o desenvolvimento dos protótipos de alta fidelidade. Estes foram elaborados com recurso à ferramenta *Adobe Indesign*, por vezes complementada pela ferramenta *Adobe Illustrator* para o desenvolvimento de alguns componentes. Os protótipos desenvolvidos nesta fase são considerados de alta fidelidade, mas não interativos, no entanto, ao desenhá-los esteve sempre presente o cuidado de pensar em toda a interação da interface e na sua navegação.

Nas figuras seguintes (4.14 a 4.24) apresenta-se um conjunto de imagens referentes aos protótipos de alta fidelidade desenvolvidos nesta fase.

Os ecrãs apresentados nas figuras 4.14 e 4.15 são relativos aos projetos, onde se pode escolher o projeto e o tipo de teste para efetuar, adicionar um novo projeto ou ainda abrir um teste já iniciado anteriormente. Na secção dos novos projetos temos a indicação de quais os tipos de testes que já foram iniciados para cada projeto, e na secção de testes recentes temos a indicação de qual o tipo de teste associado, e se tem ou não as variações de jogo e os objetivos de design definidos.

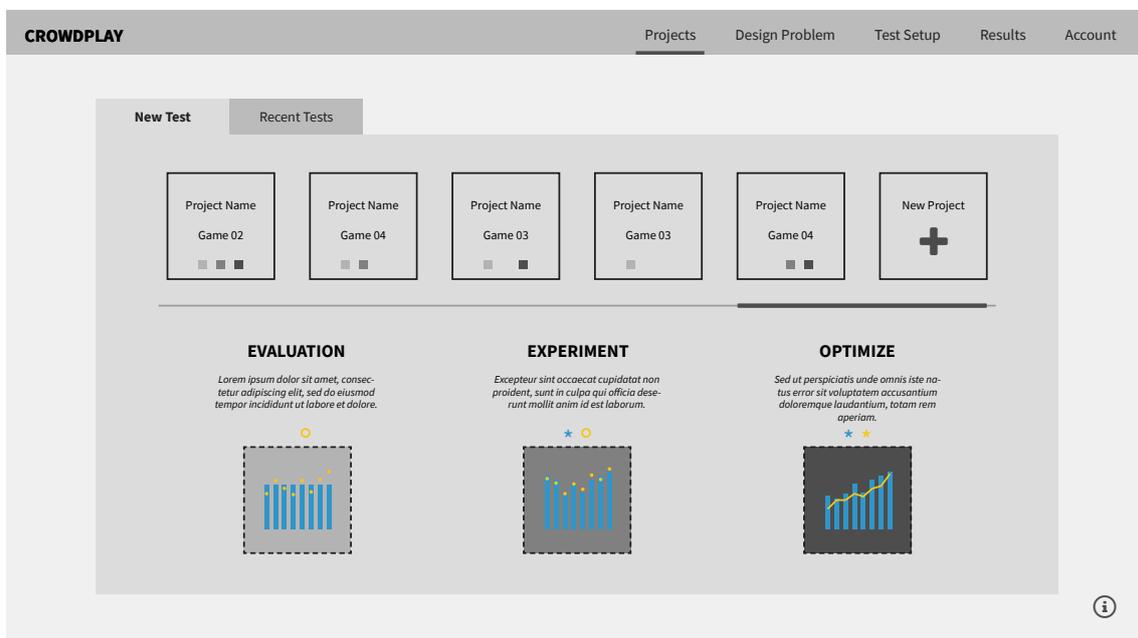


Figura 4.14 - Protótipo do ecrã de *New Test*.

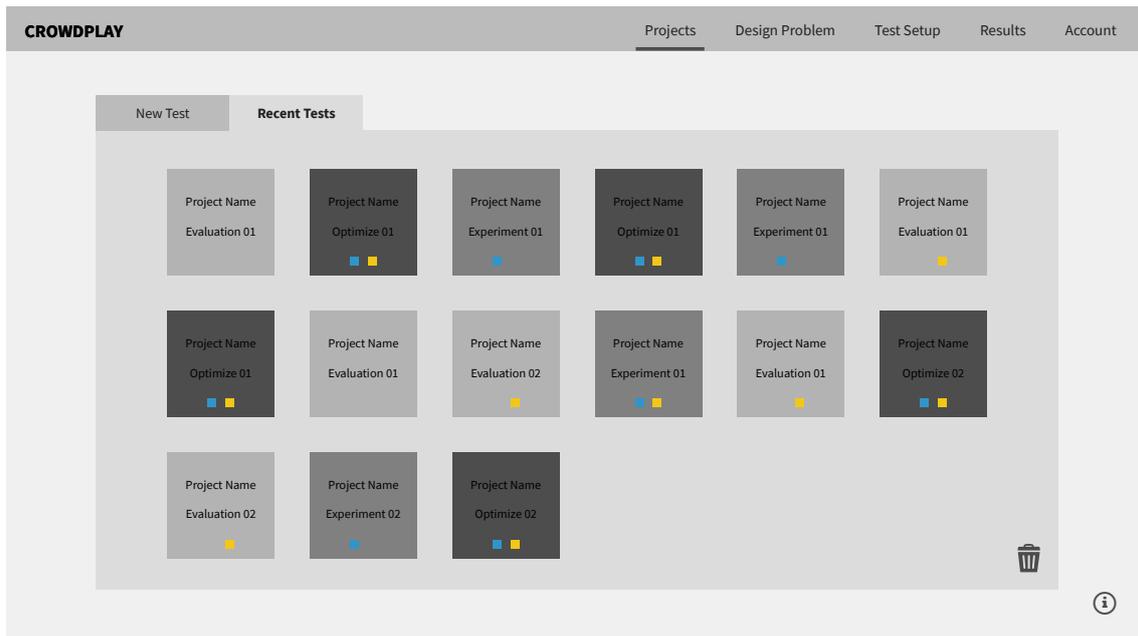


Figura 4.15 - Protótipo do ecrã de *Recent Tests*.

As figuras 4.16, 4.17 e 4.18 são relativas aos ecrãs das variações de jogo, nos quais se pode adicionar e apagar, bem como editar definindo a variável de jogo, o intervalo da variação, a função probabilística, entre outros.

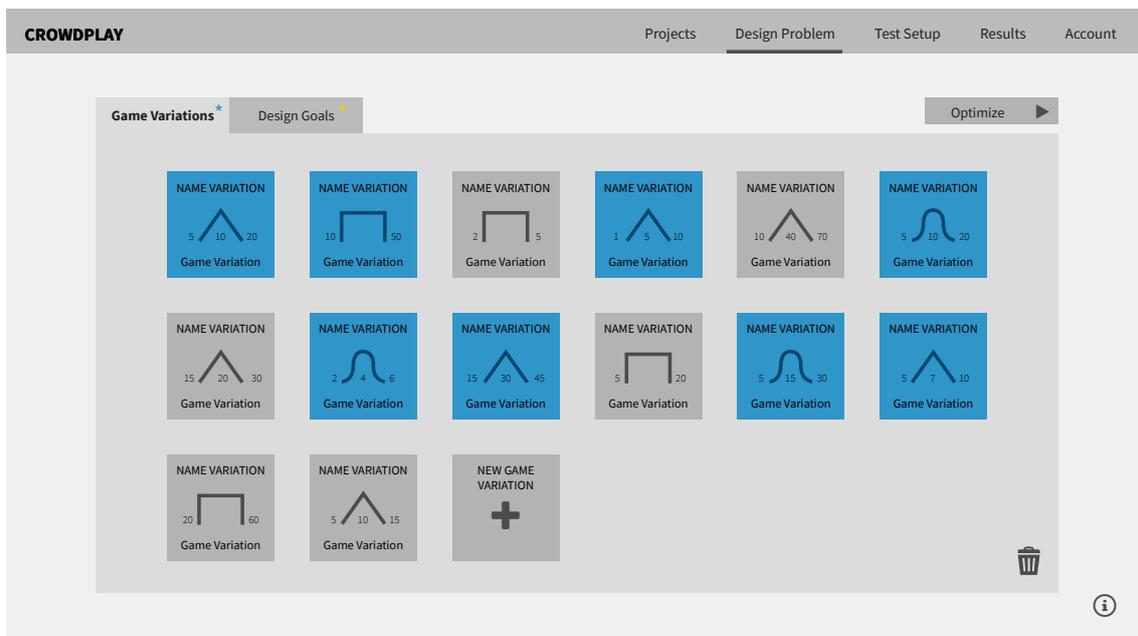


Figura 4.16 - Protótipo do ecrã de *Game Variations*.

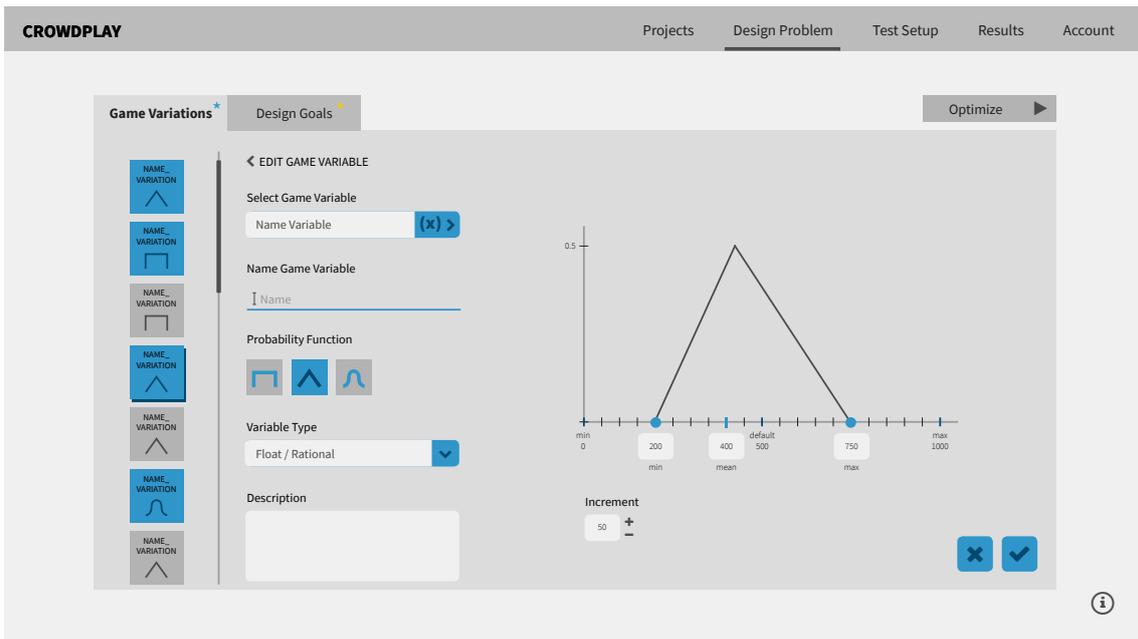


Figura 4.17 - Protótipo do ecrã para editar/adicionar uma variação de jogo.

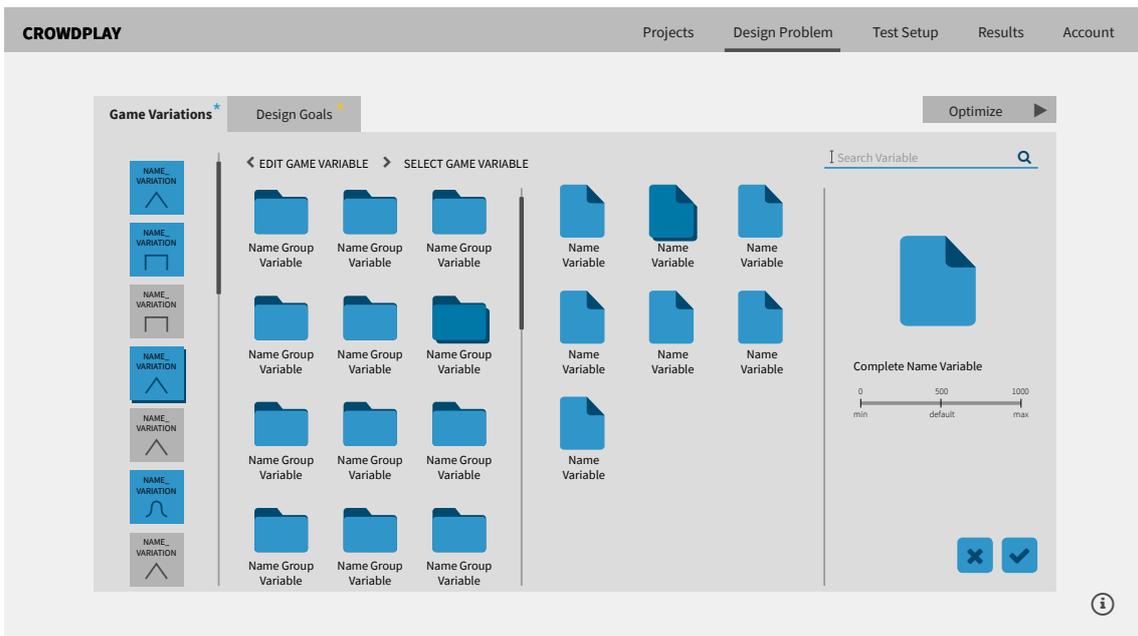


Figura 4.18 - Protótipo do ecrã para selecionar a variável de jogo.

Os ecrãs referentes aos *Design Goals* são apresentados nas figuras 4.19 à 4.22, onde podemos adicionar, apagar e editar os objetivos de design. Ao editar os objetivos de design podemos definir, para além do nome, do intervalo e função de pontuação, a fórmula.

Nas secções relativas às fórmulas (figuras 4.21 e 4.22) é também possível adicionar, apagar ou editá-las.

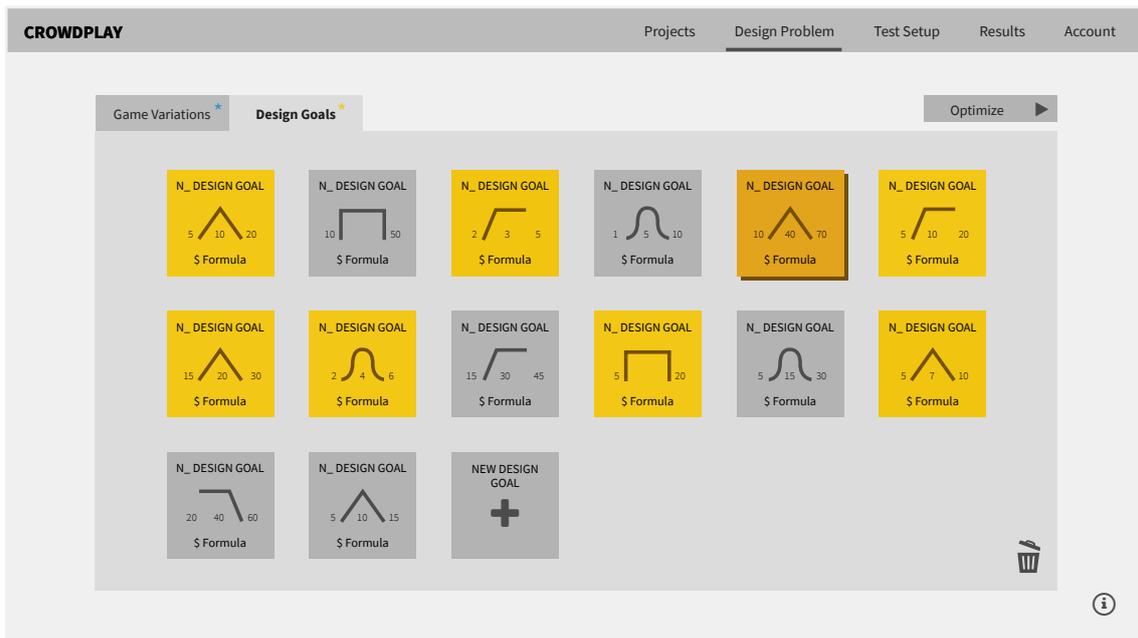


Figura 4.19 - Protótipo do ecrã de *Design Goals*.

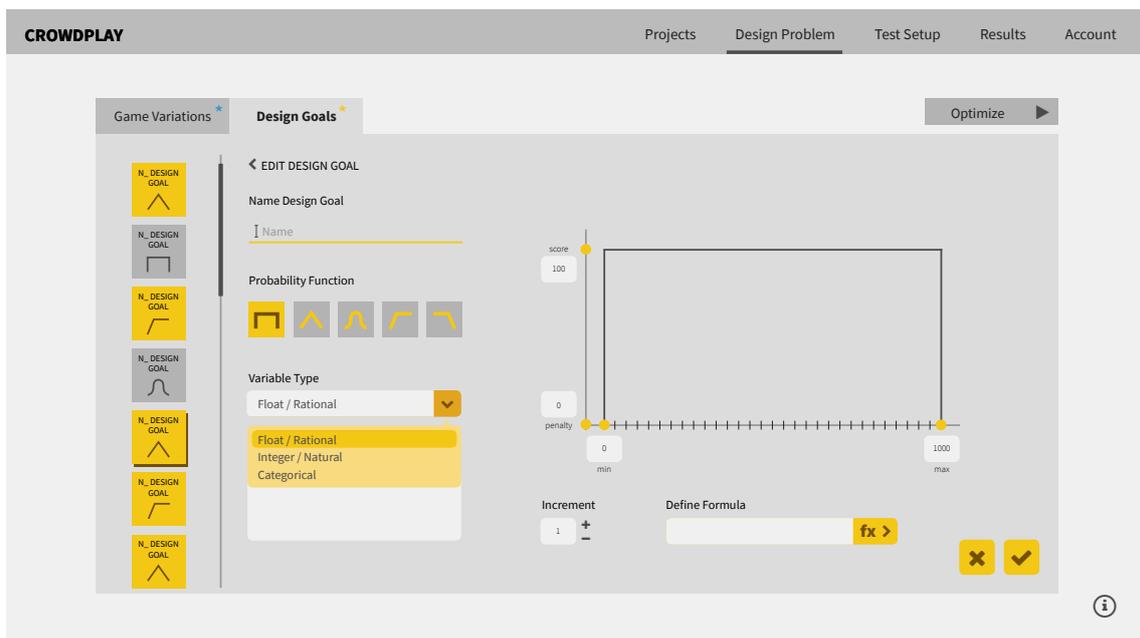


Figura 4.20 - Protótipo do ecrã para editar/adicionar um objetivo de design.

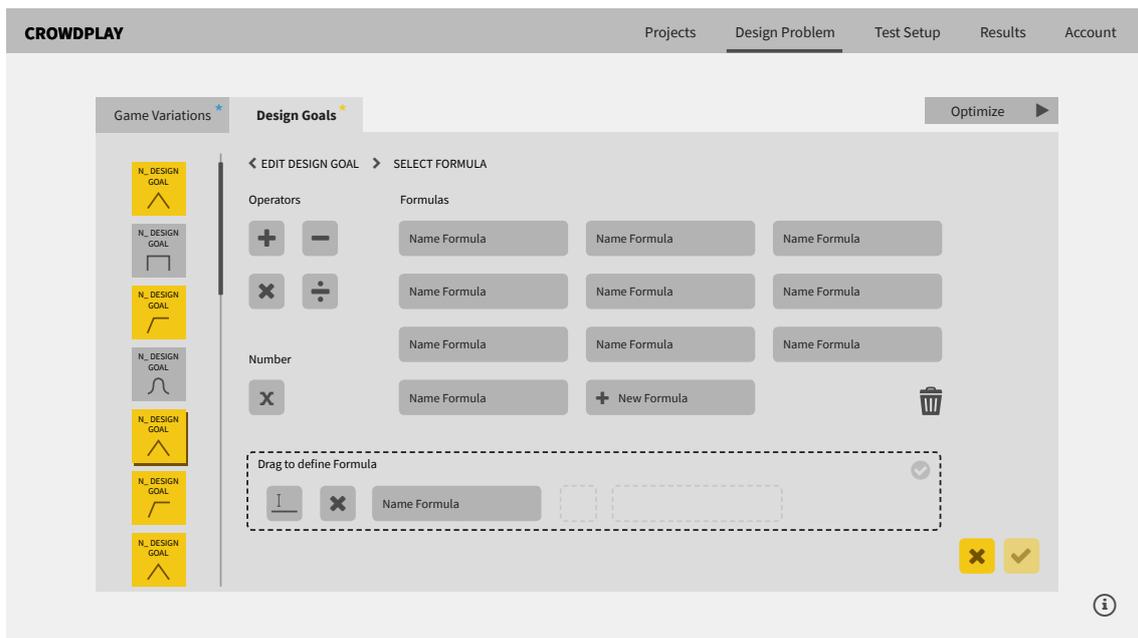


Figura 4.21 - Protótipo do ecrã para definir fórmula.

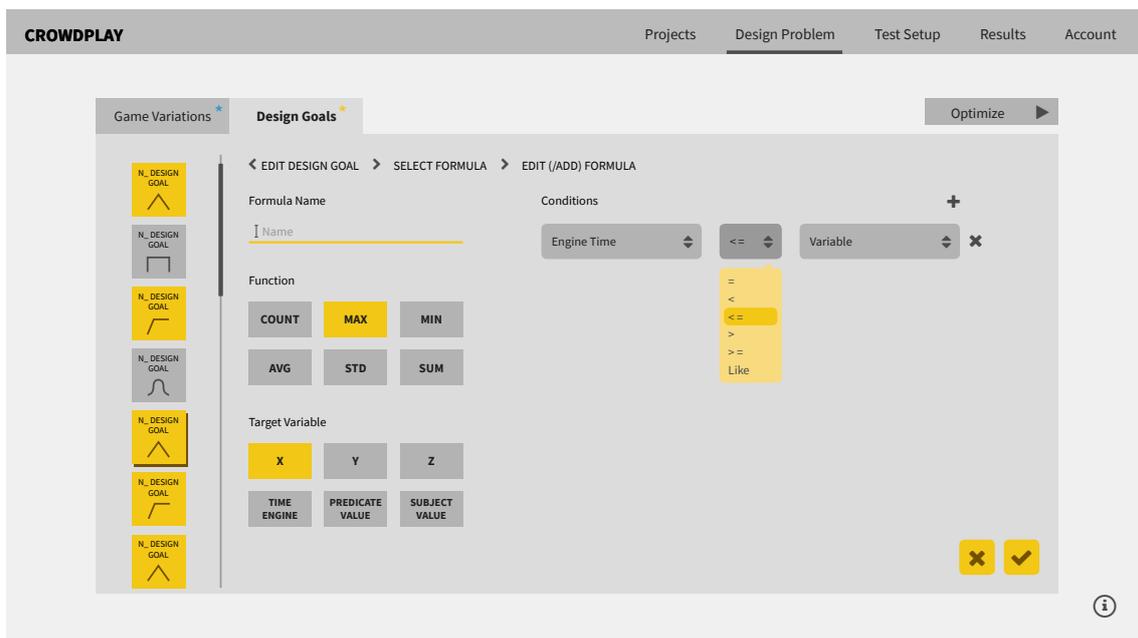


Figura 4.22 - Protótipo do ecrã para editar/adicionar fórmula.

Na figura 4.23, é visível a secção relativa ao *Test Setup*, na qual definimos as condições do teste e podemos novamente ativar ou desativar as variações de jogo e os objetivos de design.

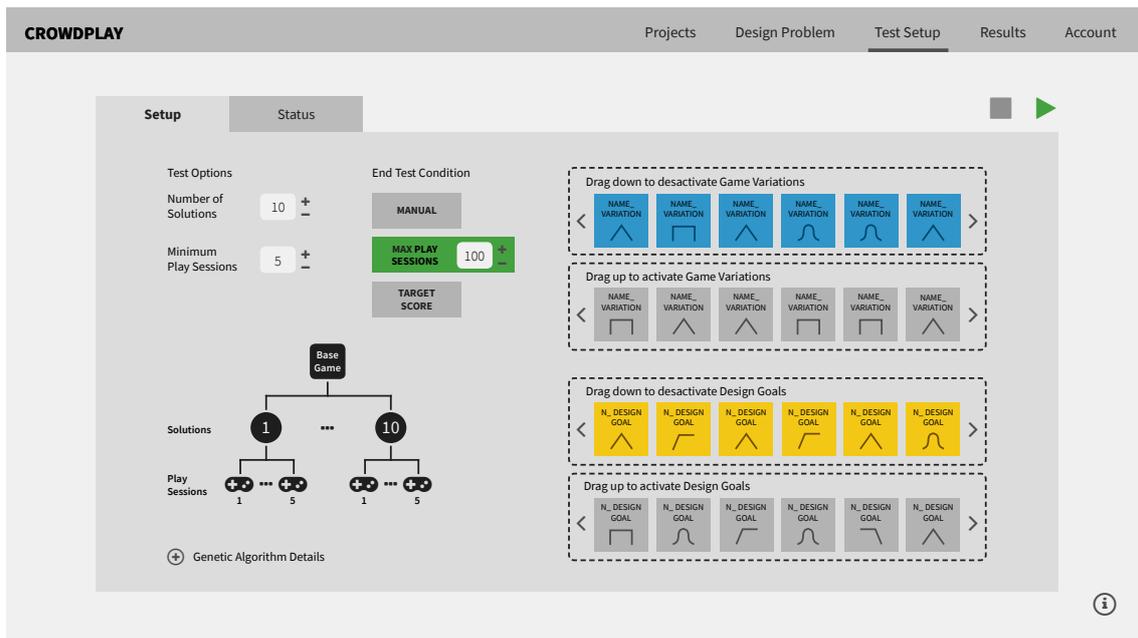


Figura 4.23 - Protótipo do ecrã para definir teste.

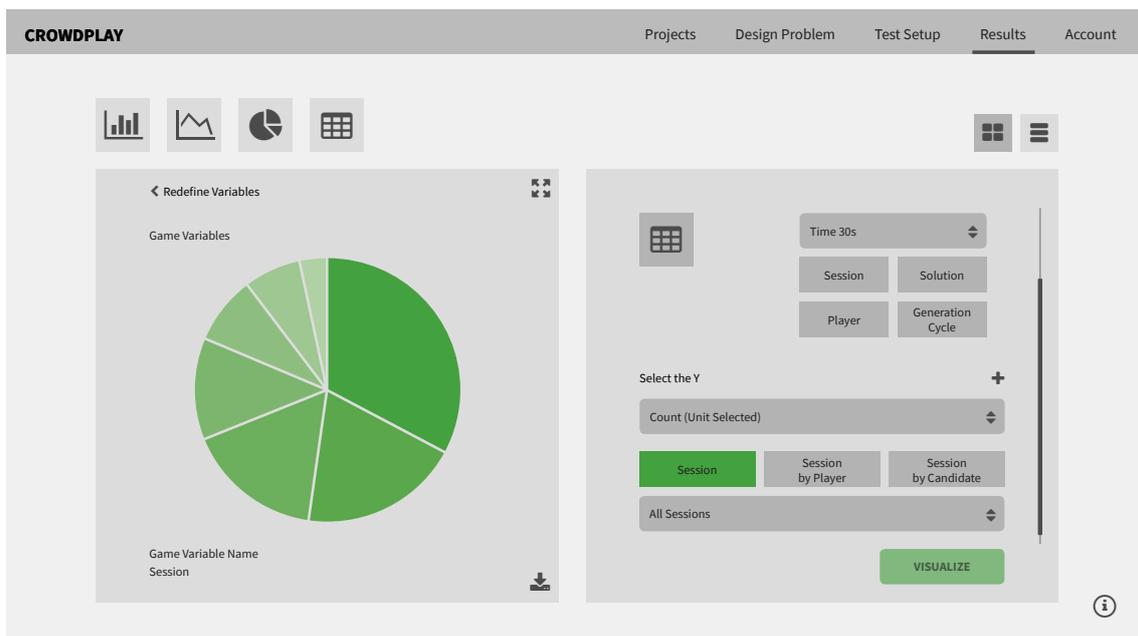


Figura 4.24 - Protótipo do ecrã *Results*.

Por fim, é apresentado um ecrã relativo à página dos resultados (figura 4.24), com a vista em grelha, na qual é possível selecionar o tipo de gráfico e as variáveis que se pretende visualizar.

Ainda nesta fase foram estudadas e analisadas a tipografia e a paleta de cores, bem como foram desenvolvidos alguns componentes e icons.

Para a tipografia da plataforma foi selecionada a fonte Source Sans Pro (figura 4.24), uma tipografia preparada para as interfaces e não serifada, e que fazia parte da seleção inicialmente definida. Na plataforma foram usados os diferentes pesos, Regular, Light, Italic e Bold para as diferentes hierarquias da informação.

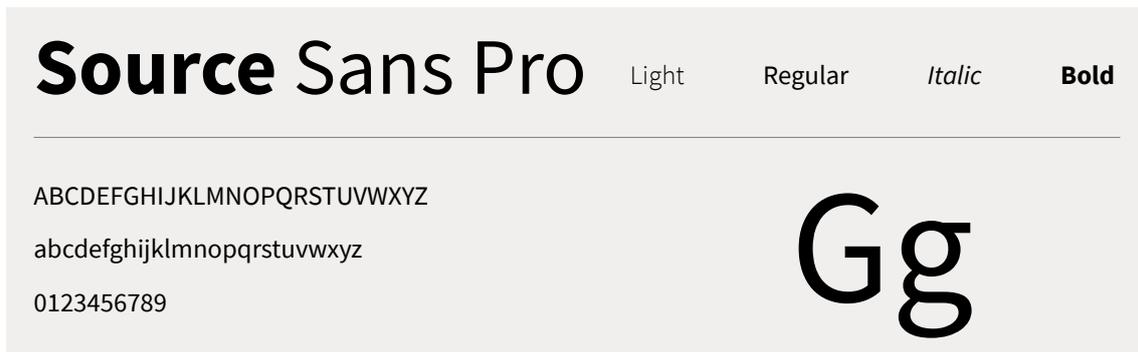


Figura 4.25 - Fonte Source Sans Pro.

Relativamente à paleta de cores, foi definida uma escala de cinzentos para toda a plataforma, desde o *background*, texto e alguns componentes. Foram ainda selecionadas três cores, cada uma delas servindo para fazer a associação a diferentes secções.

O azul é a cor para as variações de jogo, o amarelo para os objetivos de design e o verde foi a cor selecionada para o restante, ou seja, para os componentes que não são de *Design Problem*, mas relativos ao *Test Setup, Results* e *Account*. Podemos associar isso a uma analogia, ou seja, depois de definir o *Design Problem*, que contém as cores azul e amarelo, estas juntam-se, resultando assim no verde para a restante interface.

Este código de cores foi criado de modo a servir de mnemónica, e identificar mais facilmente cada componente, distinguindo-os facilmente.

Na figura 4.25 podemos observar a paleta contendo todas as cores e tons usados na interface com o respetivo código hexadecimal, RGB e CMYK.



Figura 4.26 - Paleta de cores usada na plataforma.

Foram elaboradas imagens e ícones para a interface de modo a representar alguns componentes, visíveis nas figuras 4.27, 4.28 e 4.29. Alguns ícones foram usados ou adaptados da *toolkit Font Awesome* disponível em fontawesome.io/.

Na figura 4.27 são apresentadas as imagens relativas a cada tipo de teste. Estas imagens com os ícones de diferentes gráficos foram criados de modo a identificar e distinguir cada um dos distintos testes. Nas imagens, a cor azul representa as variações dos jogos, a cor amarela os objetivos de design e a escala de cinzentos representa a complexidade dos testes.

Por ordem da imagem, temos:

- a **Evaluation** em que, como não existe variações de jogo, as barras azuis são iguais. Neste tipo de teste podemos analisar os indicadores de experiência, traduzidos nos objetivos de design, que na imagem surgem a amarelo;
- já no **Experiment** temos as variações de jogo (barras azuis de diferentes dimensões) e podemos de igual modo observar os resultados;
- no tipo de teste **Optimize**, temos representado as variações de jogo que evoluem consoante os objetivos de design.



Figura 4.27 - Imagens relativas a cada tipo de teste.

A figura 4.28 apresenta os ícones criados para representar o tipo de gráfico; nas variações de jogo o tipo de gráfico da "Probability Function" (azul) e nos objetivos de design o tipo de gráfico do "Score Function" (amarelo).

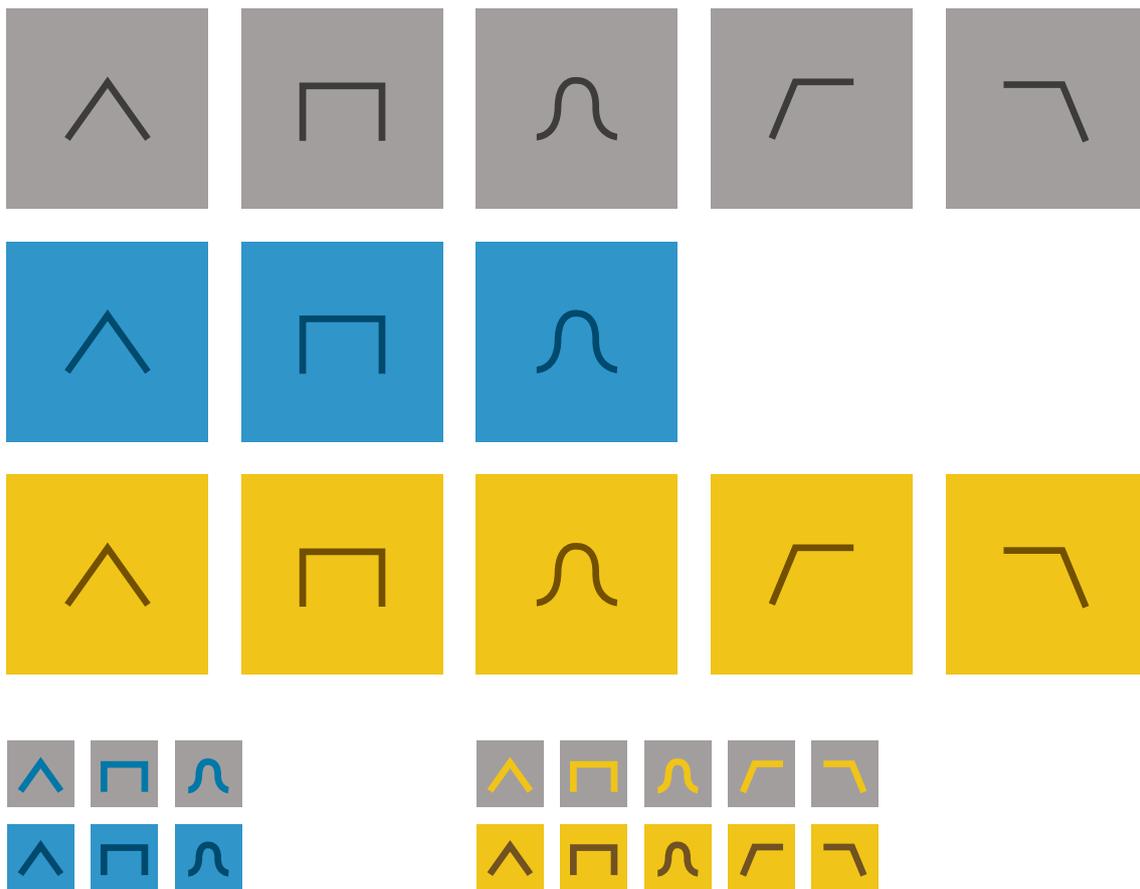


Figura 4.28 - Ícones representativos dos tipos de gráficos.

Por fim, na figura 4.29, podemos ainda visualizar outros ícones criados para representar os diferentes tipos de utilizadores, o ícon da pasta e do ficheiro para a secção das variáveis de jogo e os ícones relativos à página dos resultados.

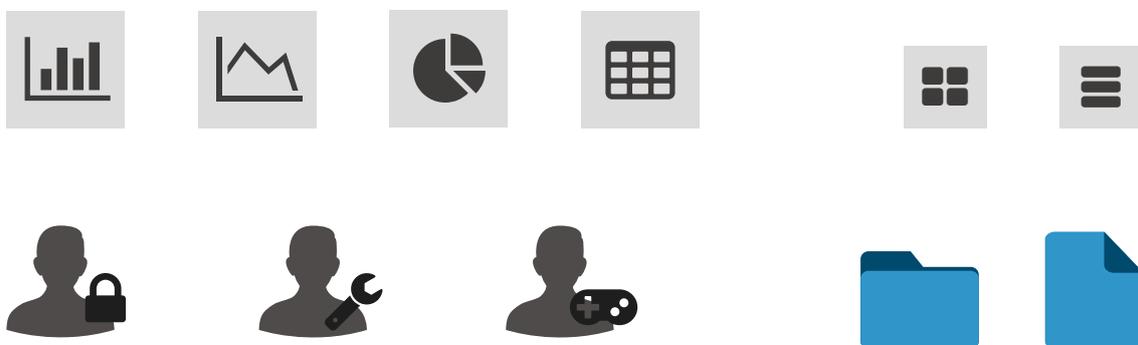


Figura 4.29 - Outros ícones.

Foram ainda desenvolvidas grelhas que serviram de base para a elaboração da interface, com indicação das respetivas medidas, margens e espaçamentos. Esta é um grelha adaptativa, e dado que os componentes da interface variam bastante, desenvolveu-se uma grelha modular, para uma maior correspondência com as características da plataforma. Na figura 4.30 é apresentada a grelha principal relativa às páginas de *Projects*, *Game Variations*, *Design Goals* e ainda às de *Projects* e *Users* na área *Account*. Consultar anexo E para ver outras grelhas desenvolvidas.

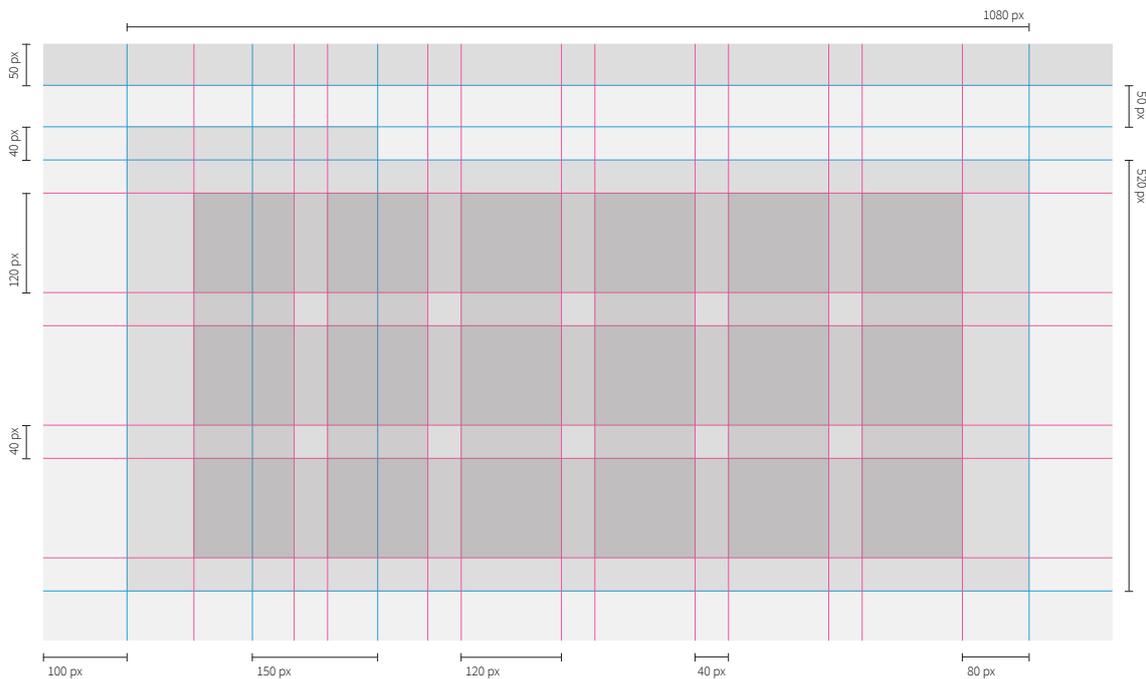


Figura 4.30 - Grelha.

No capítulo 3.3 - Requisitos e Funcionalidades - foi apresentado o diagrama UED (figura 3.1) com o objetivo de proporcionar uma melhor compreensão da navegação e das suas funcionalidades principais. Nesta fase, tendo em conta que a plataforma sofreu alterações ao nível da navegação e fluxo, optou-se por desenvolver o diagrama atualizado (figura 4.31), para compreensão da nova dinâmica.

Na figura 4.31 está apenas representado o fluxo mais natural e comum de todo o processo ao usar a plataforma, pois é possível navegar entre todas os ecrãs, apenas com um clique a partir do menu, para as páginas respetivas *Projects*, *Design Problem*, *Test Setup*, *Results* e *Settings*.

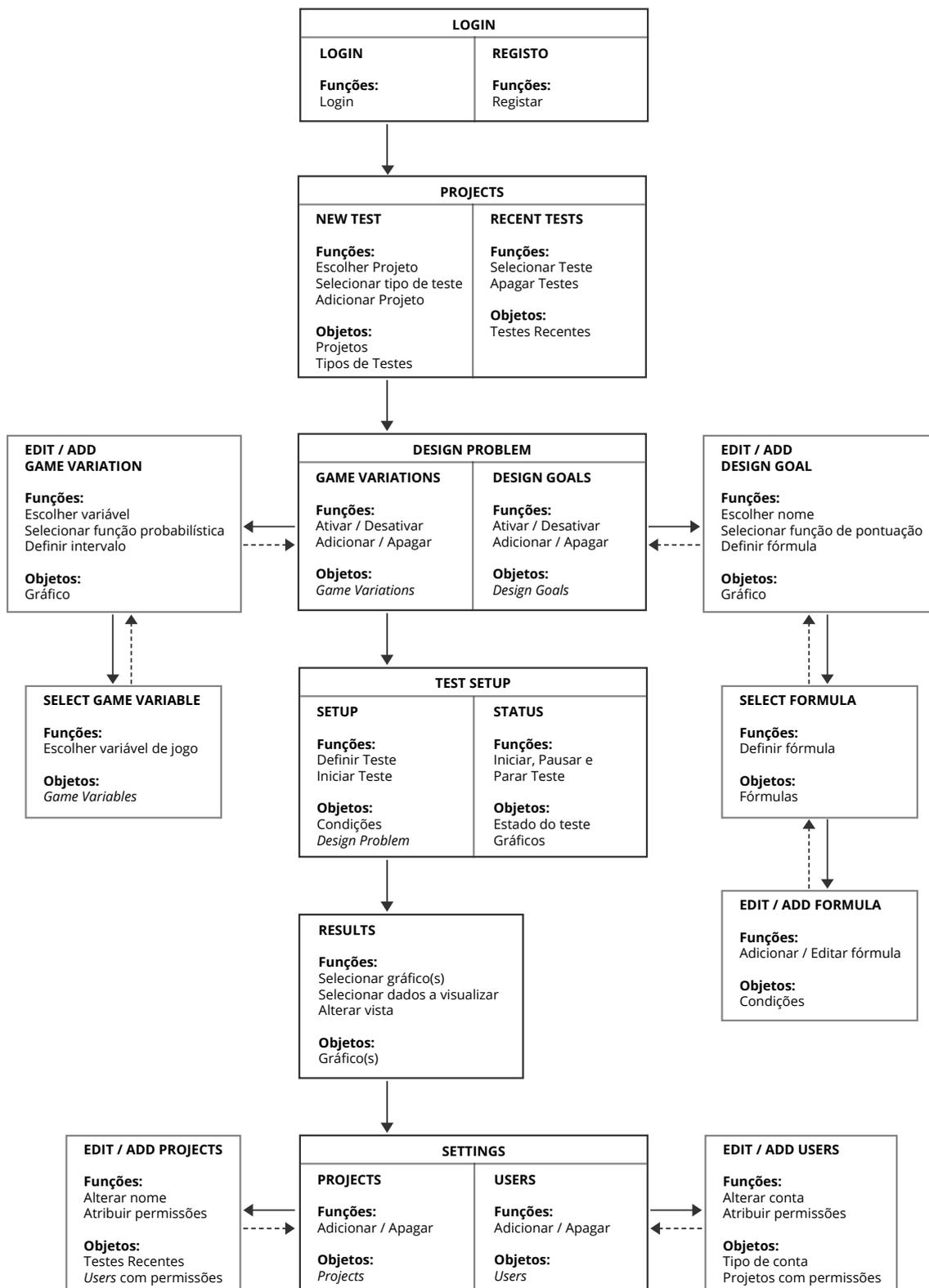


Figura 4.31 - Diagrama UED atualizado.

Um dos aspetos que foi mantido em mente aquando do desenvolvimento destes protótipos foi o facto de a navegação ajudar o utilizador no processo completo do Crowdplay.

Independentemente de no menu as etapas estarem por ordem, a interface guia-nos nesse sentido, pois, no ecrã de *New Test*, ao seleccionar o projeto e o tipo de teste, a interface reencaminhamos para o *Design Problem*. Já nos diversos ecrãs de *Design Problem* é apresentado um botão com o tipo de teste escolhido que nos reencaminha para o *Test Setup*.

Para além de se verificar uma melhoria do fluxo da interface, podemos também observar que a interface está mais completa, com novas funcionalidades, nomeadamente relativas aos *Projects* e *Settings*.

5

5.1 Software usado - JustinMind

5.2 Protótipo Interativo

5.3 Testes de Usabilidade

5.4 Alterações

Desenvolvimento da Proposta

5. Desenvolvimento da Proposta

Depois de desenvolver os protótipos, de baixa e alta fidelidade, era esperado fazer testes de usabilidade e desenvolver a interface funcional com programação (*html*, *css* e *javascript*). Por entender que não era possível testar a aplicação nos seus aspetos fulcrais (tais como o *drag and drop*, o duplo clique, entre outros) decidiu-se alterar a metodologia e o plano de trabalho.

Colocou-se então a hipótese de fazer prototipagem em papel, mas tendo em conta que não seria possível testar as características fulcrais da interface, e por esta ser uma técnica de execução demorada para o projeto em causa, optou-se por elaborar os protótipos interativos de modo a conseguir colmatar esse problema. Preferiu-se ter um protótipo interativo validado, antes de partir para a programação, pois não se poderia passar a esta fase sem a avaliar primeiro. Durante esta etapa, para validar as melhores ideias e propostas, realizaram-se avaliações breves, utilizando as heurísticas do design de interação.

Tendo isto em conta, foi necessário conceber uma alternativa para desenvolver os protótipos de modo a testar a interface por completo.

Foram então analisadas ferramentas de prototipagem de alta fidelidade com interação. Era importante ser uma ferramenta completa para que permitisse desenvolver os protótipos o mais “realistas” possível, conseguindo assim testar a interface na sua totalidade.

Dentro desta análise, foram revistas as ferramentas *InVision* (www.invisionapp.com), *Sketch* (www.sketchapp.com), *Adobe Experience Design* (www.adobe.com/pt/products/experience-design.html) e *JustinMind* (www.justinmind.com).

Para esta análise, foram tidos em conta como critérios, ser uma ferramenta que possibilitasse o desenvolvimento da interface, desde o desenho dos componentes, às interações complexas, e ainda a possibilidade de gerar código *html*.

Optou-se pelo *JustinMind* por se apresentar o mais completo e ter a funcionalidade de criar interações do tipo de *drag and drop*.

Outra característica importante para a decisão por este *software* foi a possibilidade de gerar código *html*, *css* e *javascript*, entre outras funcionalidades descritas no capítulo seguinte.

A opção por este *software* também veio solucionar o problema relativo à programação. Não só a plataforma Crowdplay existente tem características que necessitam de uma programação complexa, como também foram acrescentados alguns componentes na proposta desenhada o que poderia tornar esta tarefa ainda mais difícil. Deste modo, ao desenvolver o protótipo interativo com recurso ao *JustinMind* espera-se obter um resultado capaz de colmatar as complexidades relativas à programação.

Neste capítulo é apresentado o *software* utilizado, bem como o resultado deste, o protótipo funcional. Podemos ainda ler sobre os testes de usabilidade efetuados assim como as correções à plataforma.

5.1 Software Usado - JustinMind

Como foi referido anteriormente, a ferramenta usada para desenvolver a plataforma Crowdplay, foi o *JustinMind*, disponível em www.justinmind.com. É um *software* para prototipagem de alta fidelidade que oferece uma possibilidade vasta de criar diferentes tipos de interação.

O *JustinMind* permite desenhar toda a interface e criar interações a partir de eventos e condições.

Dentro das interações, este suporta os diferentes tipos, desde cliques, duplo cliques, *drag and drop*, ações com o teclado entre outras.

É possível configurar os eventos e as condições a cada elemento individualmente, conseguindo assim criar não só interações bastante reais como também uma interface muito próxima do produto final. Para criar as condições ou regras o *software* oferece determinadas funções e constantes associadas à lógica da programação.

Nesta ferramenta é ainda possível criar e definir variáveis e tabelas (definidas como *Data Masters* que funcionam como uma base de dados simples) o que possibilita armazenar qualquer tipo de informação durante a simulação, e tornar o protótipo com dados dinâmicos.

Este *software* permite não só gerar o código *html*, *css* e *javascript* totalmente funcional e interativo, como gerar um documento *Microsoft Word* com as especificações e recursos visuais.

A ferramenta apresenta-se com um grau elevado de complexidade, e por isso a curva de aprendizagem é moderadamente elevada. Apesar de não ser necessário desenvolver uma única linha de programação ou mesmo ter conhecimentos de programação, no entanto considera-se que deter alguns conhecimentos desta área ajuda na compreensão da ferramenta, pois esta possui uma lógica idêntica. Devido também à complexidade do *software*, o seu desempenho por vezes é irregular, sendo necessário comunicar com o suporte técnico no decorrer deste trabalho.

Para elaborar a interface do Crowdplay foi redesenhado cada elemento no *JustinMind* com base no que se tinha planeado para os protótipos de alta fidelidade com recurso ao *Adobe Indesign*. Apesar de estes últimos protótipos conterem um grande nível de fidelidade, era necessário desenvolver elemento a elemento no próprio *JustinMind* para elaborar um protótipo bastante interativo e próximo do real. Considera-se que esta fase consumiu algum tempo, no entanto, o resultado final é compensatório.

De seguida são apresentadas dois *screenshots* do *workspace* do *JustinMind* (figuras 5.1 e 5.2), nas quais podemos observar algumas características do *software* descritas anteriormente e parte do desenvolvimento da interface do Crowdplay. Concretamente na figura 5.1, destaca-se na zona central o *workspace*, a área onde se desenha a interface com recurso à secção de *Properties*, e na zona inferior, a área onde se define os eventos de um determinado elemento. Na figura 5.2 é visível a caixa referente à zona onde podemos definir expressões relativas às *Data Masters*, neste caso para adicionar uma variação de jogo. Para ver mais dos *screenshots* relativos ao *JustinMind* e desta fase de desenvolvimento ver anexo F.

Re-Crowdplay

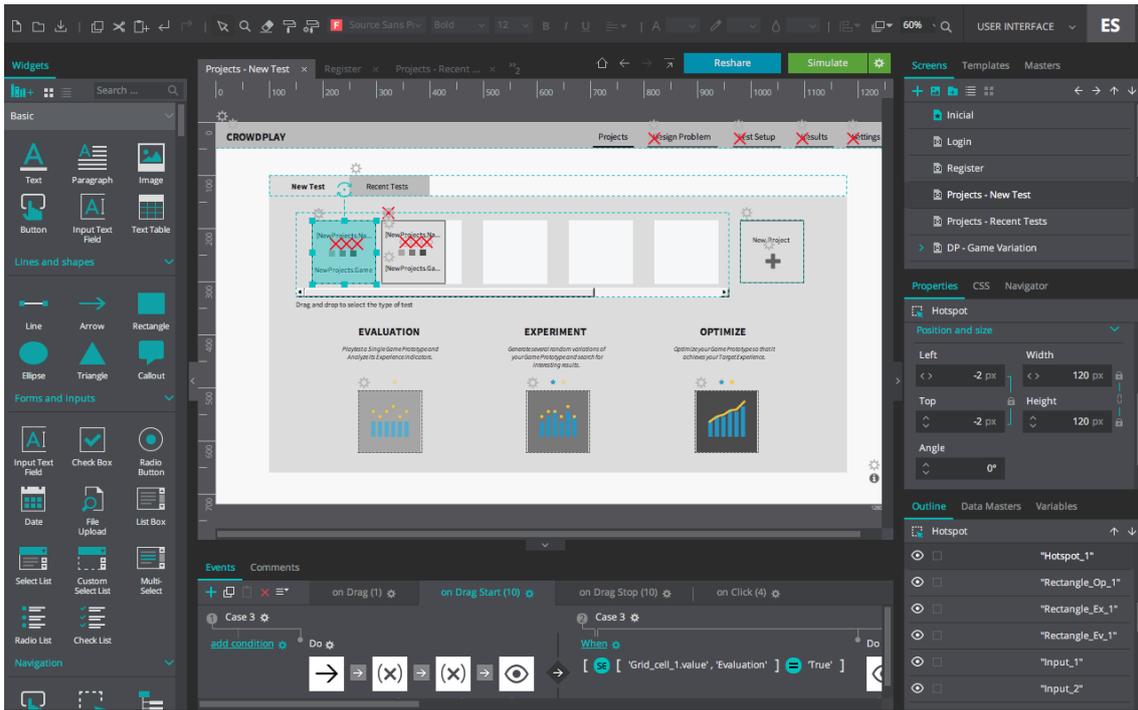


Figura 5.1 - Screenshot do software JustinMind 1.

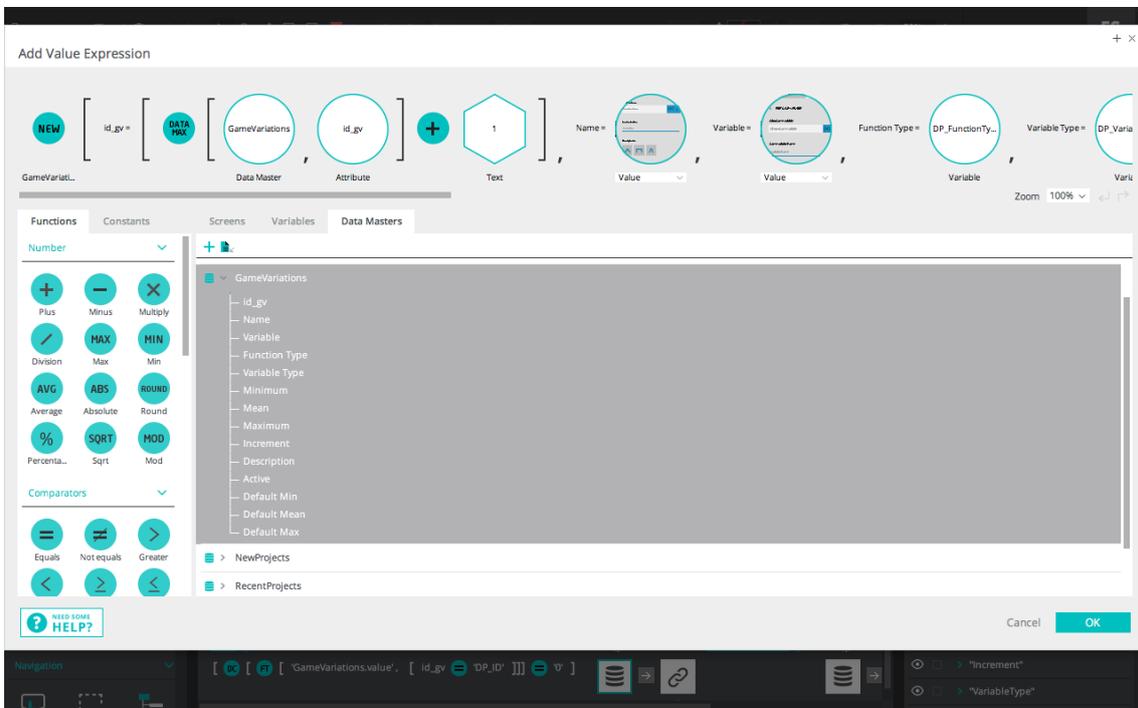


Figura 5.2 - Screenshot do software JustinMind 2.

Para desenvolver um protótipo que mimetize o mais possível as condições do real, foram criadas tabelas (nomeadas de *Data Masters* no *software*) com dados plausíveis. Foram então elaboradas tabelas com informações relativas a *New Test*, *Recent Tests*, *Game Variations*, *Game Variables*, *Design Goals*, *Formulas*, e *Users*. No entanto o *software* não permite associar e relacionar tabelas, pelo que existe apenas um conjunto de dados de cada tabela, ou seja, para qualquer projeto selecionado, aparece a mesma informação de *Game Variables* e *Design Goals* (por exemplo). É ainda de referir que os dados apresentados são apenas dados fictícios preparados para o protótipo.

Como já foi referido anteriormente, este *software* gera, para além do código *html*, um documento com as especificações das ações, eventos, condições e tudo o que foi “programado”. Ao concluir o desenvolvimento deste protótipo foi gerado este documento, que resultou num pdf de 120 páginas. É apresentada parte deste em anexo (anexo G), onde foi selecionada a secção referente aos ecrãs relativos aos objetivos de design.

5.2 Protótipo Interativo

Neste subcapítulo é apresentada e descrita a interface na sua totalidade, explicando o seu funcionamento e algumas considerações relativas ao design da interface. Para esta descrição far-se-á uso das imagens dos ecrãs desenvolvidos com recurso ao *software JustinMind*.

A plataforma redesenhada do Crowdplay pode ser vista a partir do endereço crowdplay.dei.uc.pt:8080/p1, sendo possível a um utilizador comum criar conta, registando-se como designer. Para entrar na plataforma como *admin* ou *gamer*, pode-o fazer pelo *login* com o *username* “Admin” e/ou “Gamer” com as *passwords* “admin” e/ou “play”, respetivamente.

Na figura 5.3 é apresentado o ecrã relativo ao *Login*, onde o utilizador entra na plataforma com o *username* e *password*. Caso o utilizador ainda não tenha conta, este pode preencher o formulário idêntico ao do *Login*, com registo do *username*, email, *password* e confirmação desta.

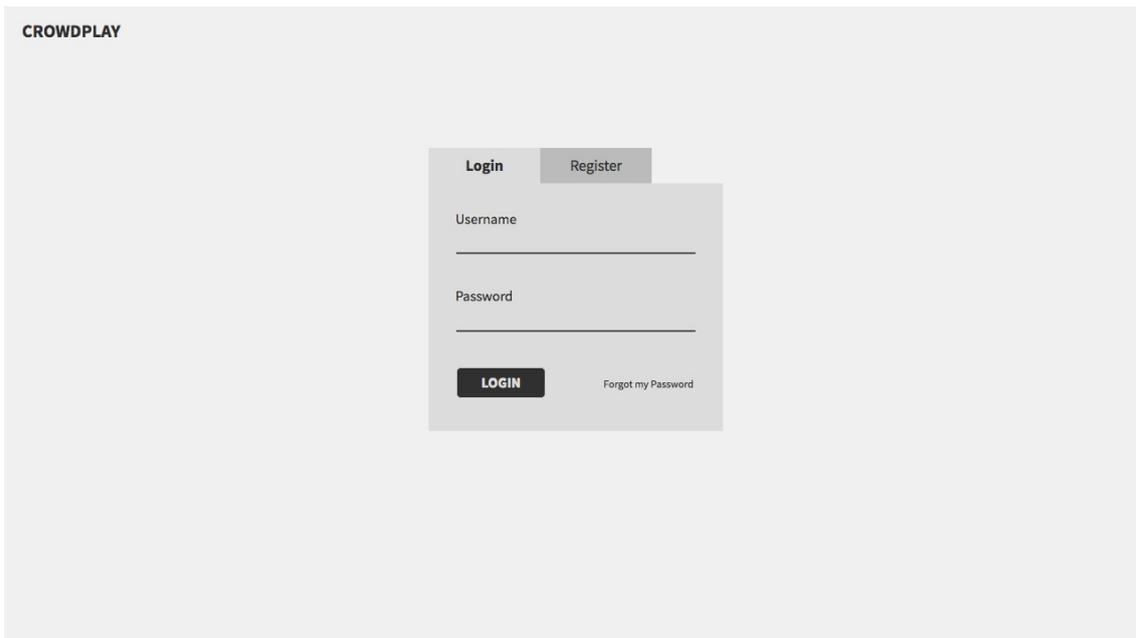


Figura 5.3 - Ecrã de *Login*.

A figura 5.4 apresenta o primeiro ecrã após entrar na ferramenta propriamente dita. Nesta página é possível selecionar o projeto e o tipo de teste que se pretende operar, e, esta ação efetua-se arrastando o projeto para o tipo de teste desejado. Esta ação encaminha para a página de *Design Problem* consoante o tipo de teste definido. Ainda no ecrã de *Projects - New Test* é possível adicionar um novo projeto.

Relativamente a algumas características do desenho, temos, como já foi referido no capítulo 4.4 (Prototipagem de Alta Fidelidade), as imagens referentes a cada tipo de teste, de modo a ajudar o utilizador a associar mais rapidamente aos objetivos de cada teste. Na lista dos novos projetos podemos observar o quadrado com a cor referente ao(s) tipo(s) de teste(s) a que cada projeto já foi submetido. Com esta referência conseguimos visualizar a que testes um determinado projeto já foi submetido sem se ter que verificar no ecrã *Recent Tests*.

Já no ecrã *Recent Tests*, visível na figura 5.5, podemos observar a que tipo de teste se refere um determinado projeto e em que fase este se encontra. Pelos ícones presentes temos a indicação da fase em que se encontra o teste: na fase de definir o *Design Problem*, na fase de teste (em *play* ou em pausa) ou se já se encontra concluído e com os respetivos resultados. Ao abrir um teste, a interface reencaminha para o ecrã referente a essa fase, ou seja, a página onde se ficou na última vez ao trabalhar com esse projeto.

É ainda possível apagar um teste já efetuado, arrastando-o para o ícone do caixote do lixo.

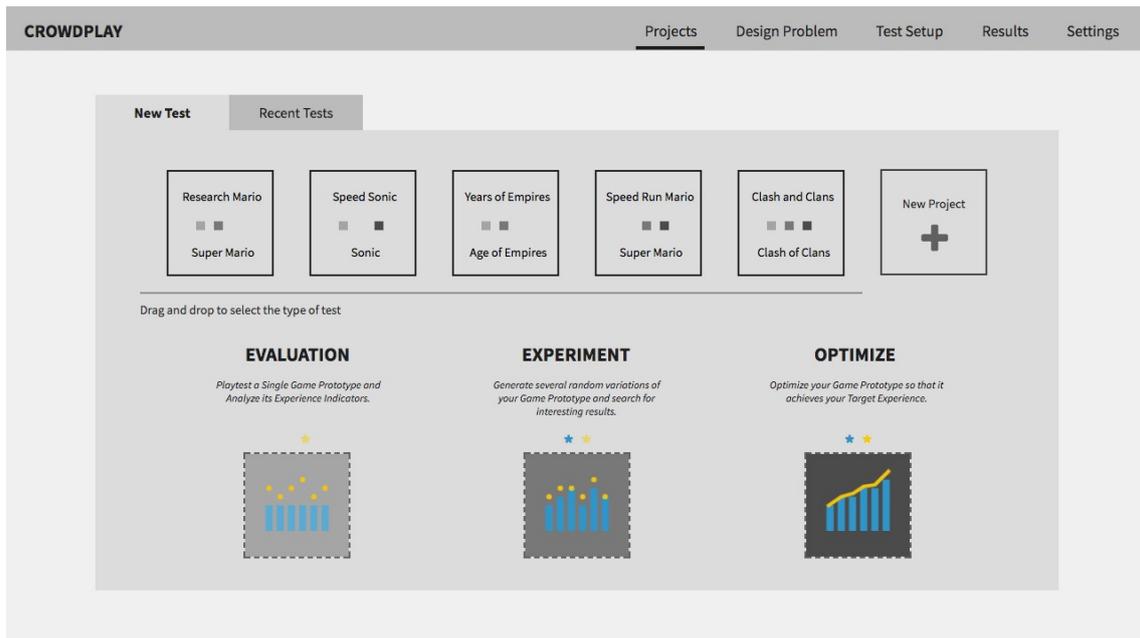


Figura 5.4 - Ecrã de *New Test*.

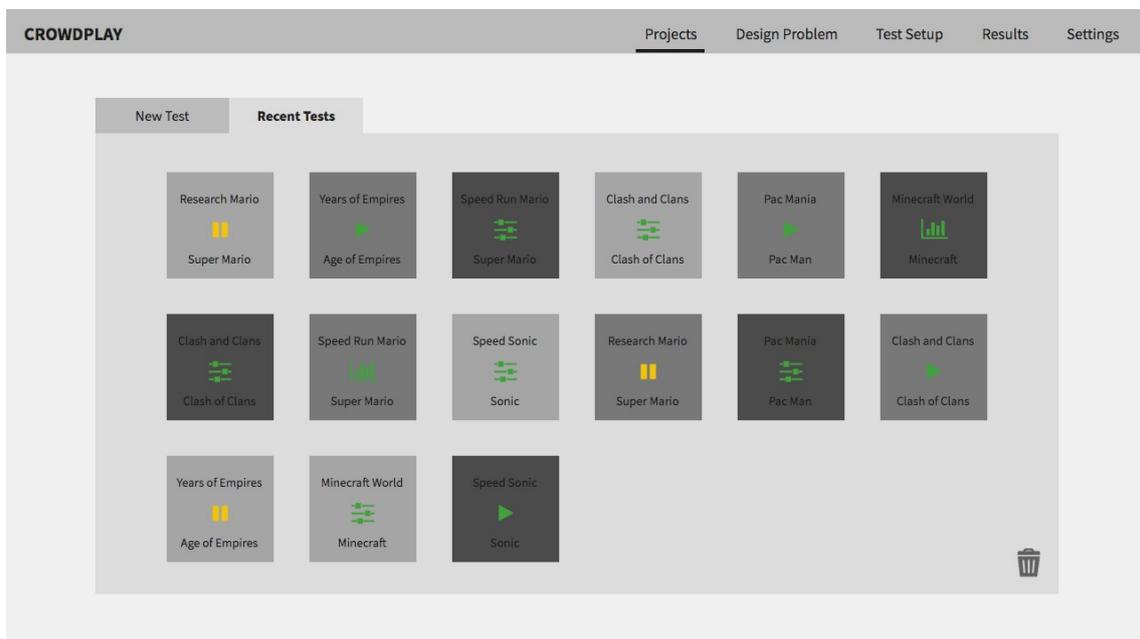


Figura 5.5 - Ecrã de *Recent Tests*.

As figuras 5.6 à 5.12 referem-se à área *Design Problem*, a qual contém as secções relativas a *Game Variations* e *Design Goals*, identificadas pelas cores azul e amarelo respetivamente. Nestes ecrãs é visível no canto superior direito informação do nome do projeto e o tipo de teste selecionado, pelo que o botão relativo à informação do tipo de teste encaminha para a página de *Test Setup*, de modo a simplificar a ação de iniciar o teste após a definição do problema. Esta zona é usada para dar orientação de qual o próximo passo a efetuar, para que o utilizador não se sinta perdido na interface. Consoante o tipo de teste escolhido, a interface guia-nos no sentido de perceber se as variações de jogo e os objetivos de design são obrigatórios ou opcionais, com a indicação do asterisco referente.

Na figura 5.6, é apresentado o ecrã com as *Game Variations*, no qual podemos ativar/desativar com um clique ou apagar arrastando-as para o ícone do lixo. Em cada variação de jogo é apresentada, no respetivo quadrado, o nome, a variável de jogo, o intervalo e a sua função de probabilidade (gráfico), bem como se está ativa ou desativa para o teste consoante a cor azul ou cinzenta, respetivamente.

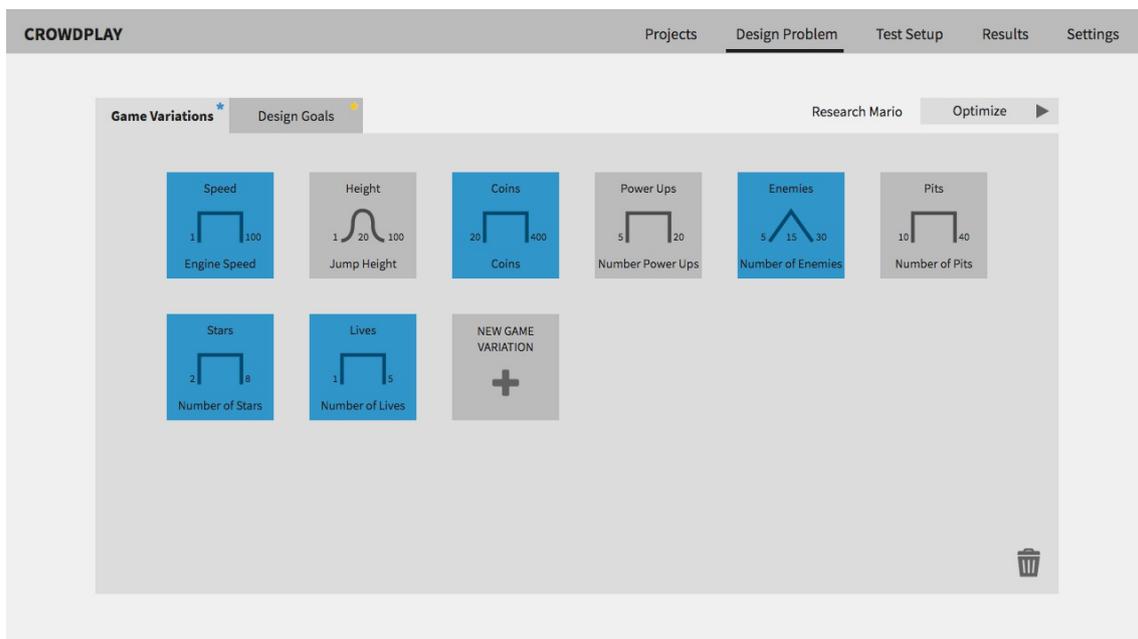


Figura 5.6 - Ecrã de *Game Variations*.

A figura 5.7 apresenta o ecrã no qual se edita ou cria uma variação de jogo. Podemos definir o nome da variação de jogo, e seleccionar a função probabilística e o tipo de variável. Para definir o intervalo da variação, podemos fazê-lo de três modos distintos: escrevendo no campo, arrastando, ou aumentar e diminuir com recurso às setas. Ao utilizar as setas é ainda possível, com recurso à tecla *Tab* (primida), alterar para incrementar dez valores, ao invés de incrementar um. Idealmente, a tecla usada para esta funcionalidade seria o *Shift* (característica familiar aos utilizadores comuns de diferentes *softwares*), no entanto, não foi possível “programar” essa tecla no *software*. De referir que se prevê que no produto final, o gráfico se adapte do mesmo modo que as caixas numéricas, consoante o valor do intervalo, no entanto não foi possível definir isso com recurso ao *software JustinMind*. Na lateral esquerda é possível visualizar “as miniaturas” de todas as variações de jogo, podendo ainda alterar entre elas a partir de um clique. Esta funcionalidade permite ao utilizador visualizar diferentes variações de jogo sem ter que retroceder ao ecrã anterior, minimizando assim os cliques. Esta funcionalidade está presente em vários ecrãs.

Ainda neste ecrã (figura 5.7), ao seleccionar a variação de jogo, esta encaminha para a página apresentada na figura 5.8, onde as variáveis de jogo são apresentadas sob a forma idêntica a pastas de um sistema. As variáveis associadas neste caso a ficheiros estão agrupadas por categorias, ou seja as pastas. Para facilitar mais a procura, esta secção apresenta ainda uma barra de pesquisa. É possível seleccionar a variável pretendida ao clicar no botão *check* ou com o duplo clique sobre a variável.

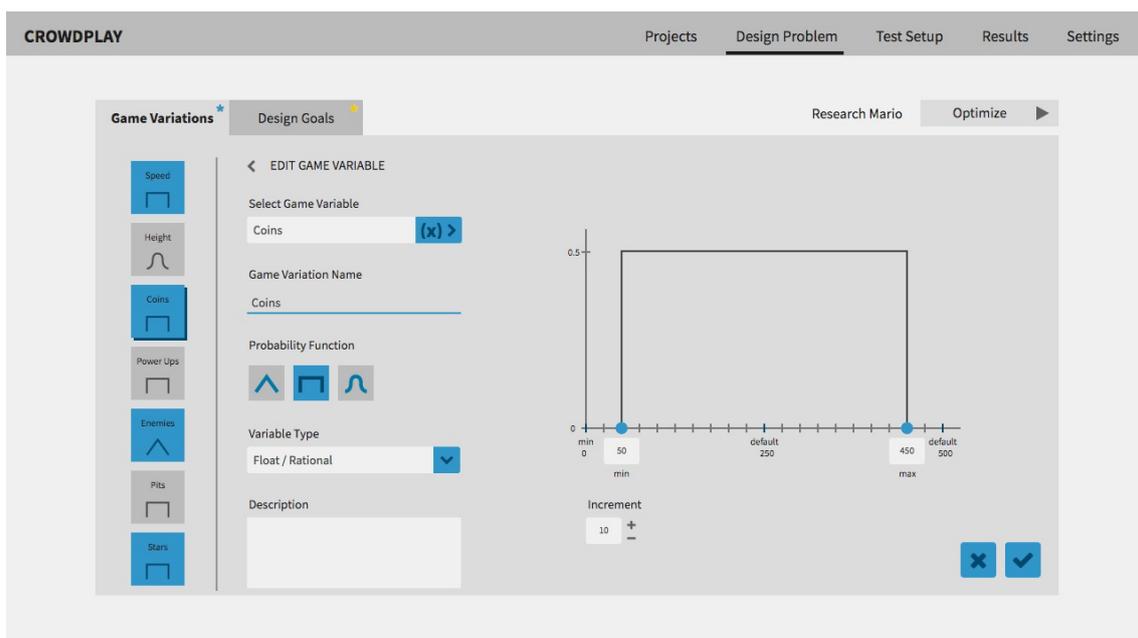


Figura 5.7 - Ecrã para editar/adicionar uma variação de jogo.

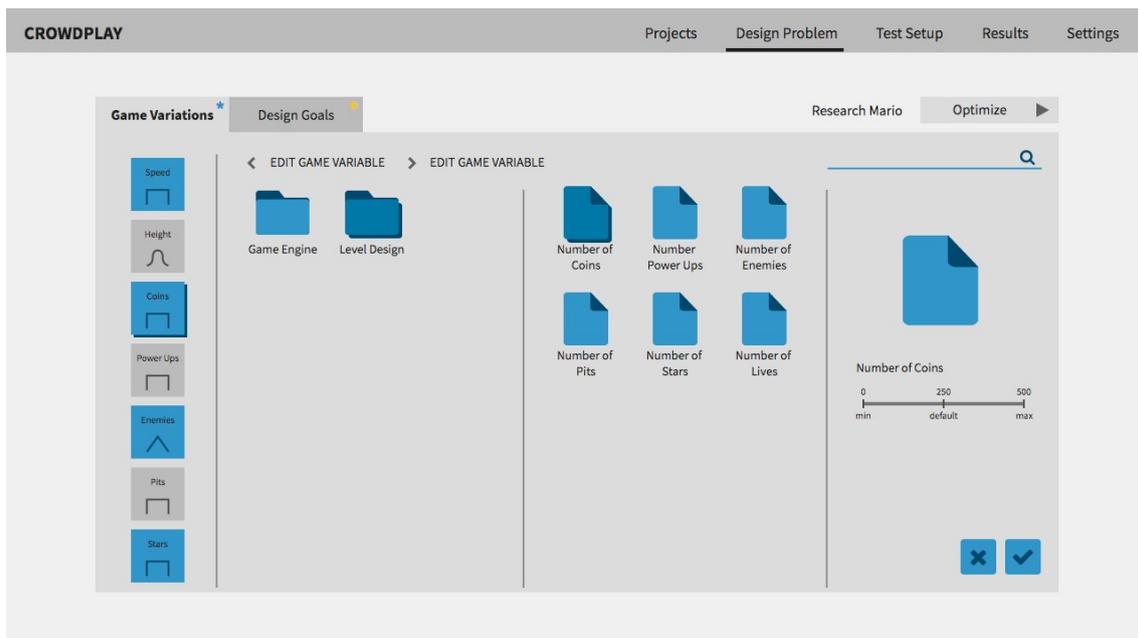


Figura 5.8 - Ecrã para seleccionar a variável de jogo.

A secção referente aos *Design Goals* apresenta-se semelhante às *Game Variations*. O ecrã principal dos objetivos de design (figura 5.9) funciona do mesmo modo que nas variações de jogo. Já no ecrã de editar ou adicionar um objetivo de design (figura 5.10), este é idêntico apesar de algumas funcionalidades diferentes. Para editar os valores relativos à pontuação, podemos, de igual modo, escrever, usar as teclas ou arrastar (os círculos), sendo que neste caso as caixas são fixas (por não existir um valor máximo e mínimo). O ecrã para definir a fórmula (figura 5.11), contém as fórmulas e os operadores para que os utilizadores definam a fórmula pretendida arrastando para a área relativa. É possível o utilizador retroceder à construção da fórmula, assim como é apresentado um *feedback* quando é definida uma fórmula válida.

No produto final, é esperado que, ao arrastar um elemento para fora da caixa, o elimine da fórmula, no entanto, isto não foi possível desenvolver com recurso ao *JustinMind*. Esta página apresenta por vezes algumas falhas na interação, devido à complexidade das ações “programadas”. Neste ecrã é ainda possível apagar as fórmulas, assim como editar ou adicionar, o que direciona para a ecrã seguinte (figura 5.12). Para editar ou adicionar as fórmulas é necessário definir o nome da fórmula, escolher a função e a *target variable*, bem como definir uma ou mais condições.

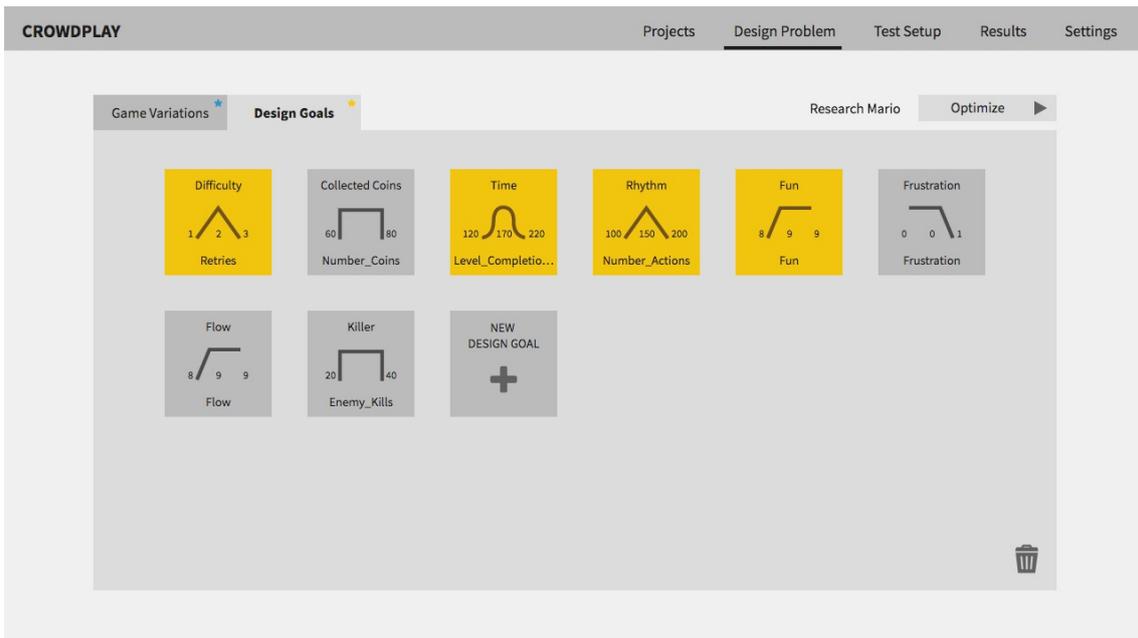


Figura 5.9 - Ecrã de *Design Goals*.

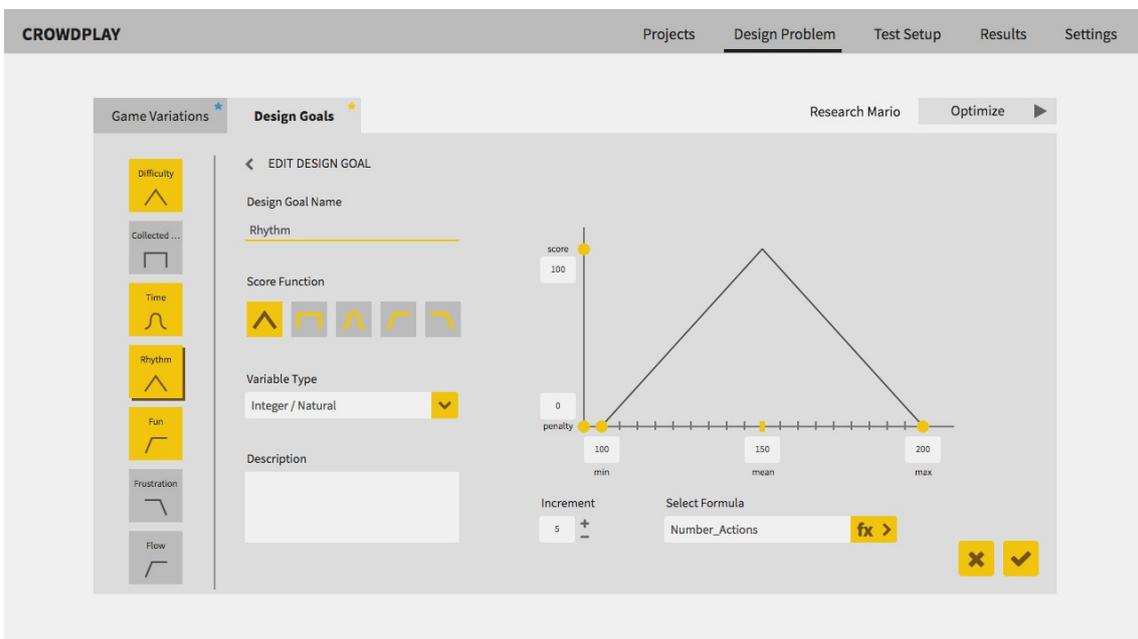


Figura 5.10 - Ecrã para editar/adicionar um objetivo de design.

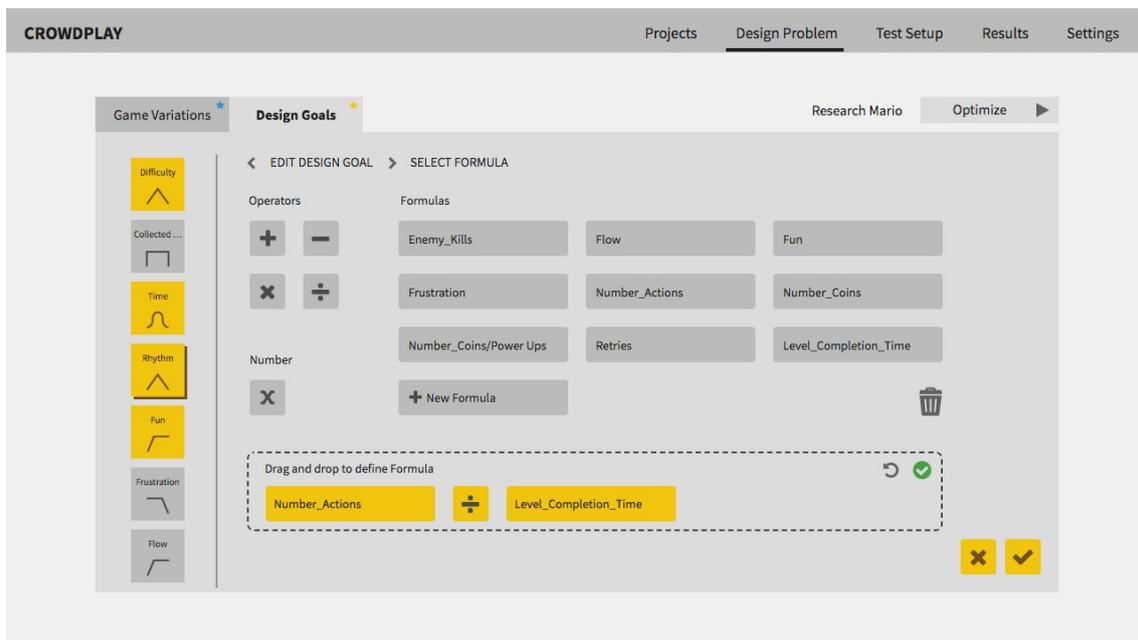


Figura 5.11 - Ecrã para definir fórmula.

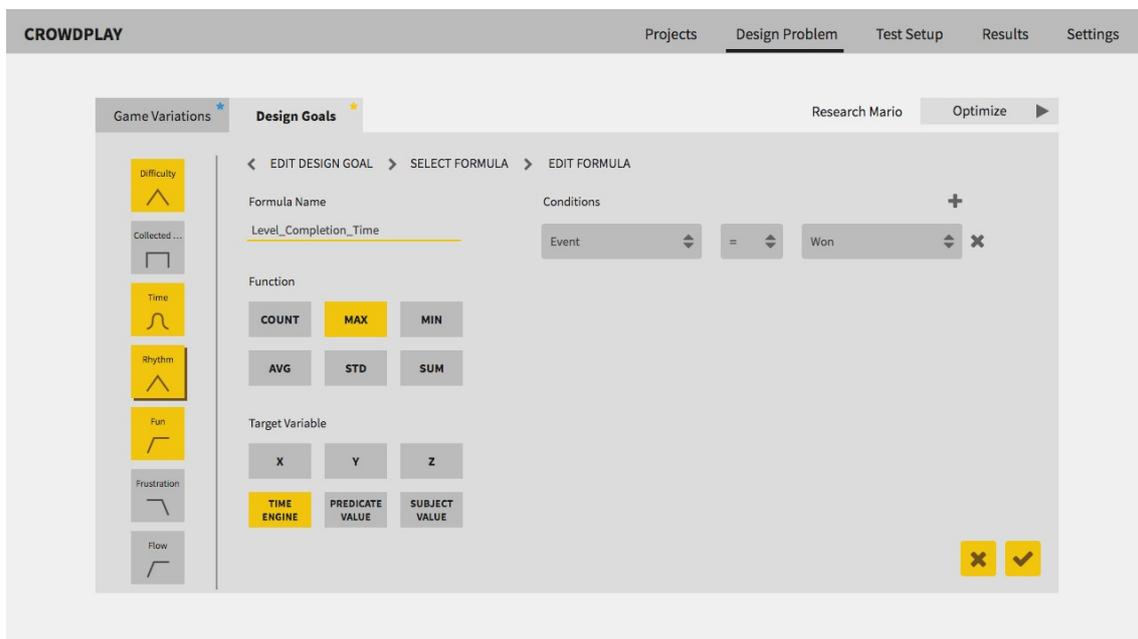


Figura 5.12 - Ecrã para editar/adicionar fórmula.

Na secção referente ao *Test Setup* temos as áreas de *Setup* e *Status*, presentes nas figuras 5.13 e 5.14, respetivamente.

No ecrã para definir o teste (figura 5.13), podemos selecionar as opções e as condições de fim de teste. É apresentado um esquema que visa ajudar os utilizadores na compreensão do teste consoante os valores definidos.

É ainda possível nesta fase redefinir as variações de jogo e os objetivos de design ativos para o teste, o que também serve apenas para visualizar os ativos, como resumo do teste.

No canto superior direito são apresentados os botões para iniciar, pausar e parar o teste. Ao iniciar o teste, clicando no botão *play*, este encaminha para o ecrã *Status* (figura 5.14). Neste ecrã, são apresentadas barras de progresso, consoante a condição de fim de teste anteriormente selecionada, para indicar em que estado está o teste, bem como alguns gráficos (de barras ou histogramas) que informam sobre outras características do teste.

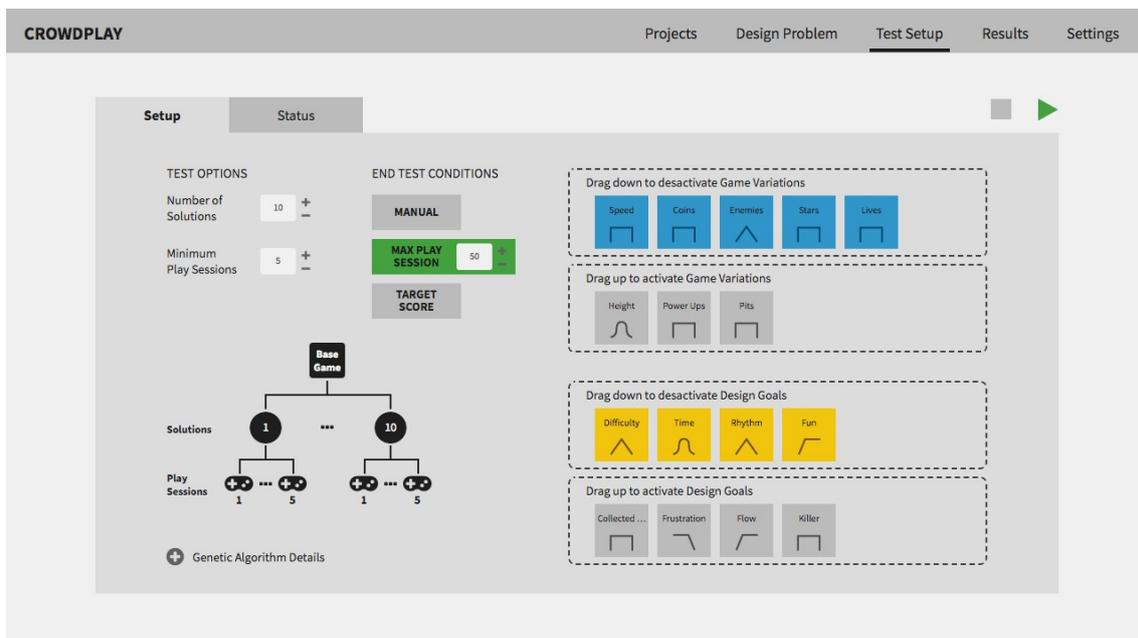


Figura 5.13 - Ecrã para definir teste.

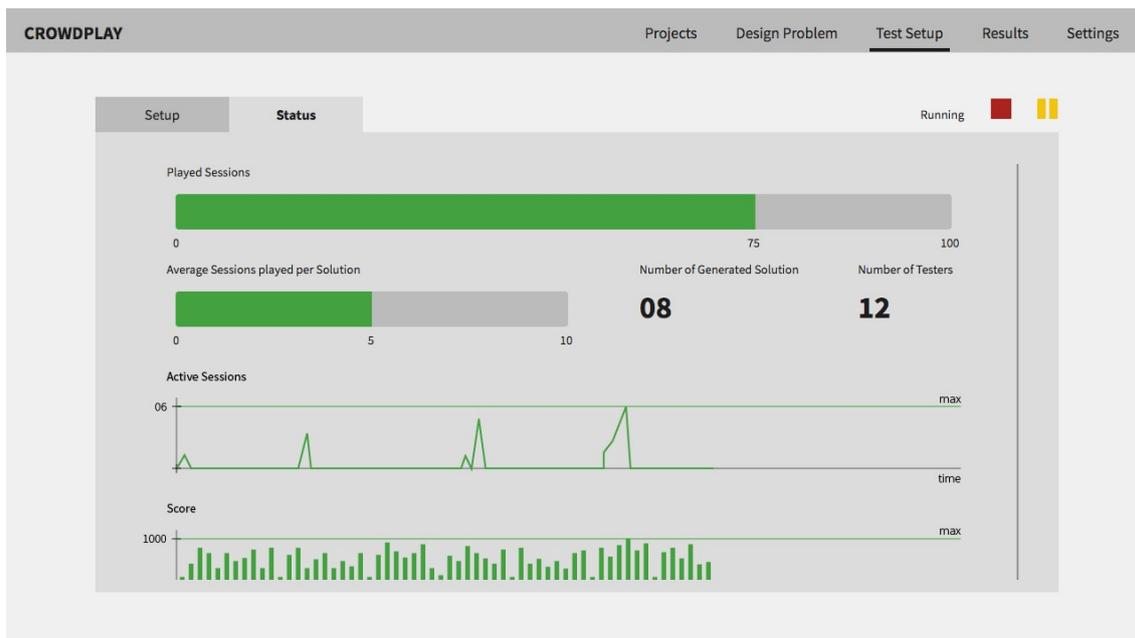
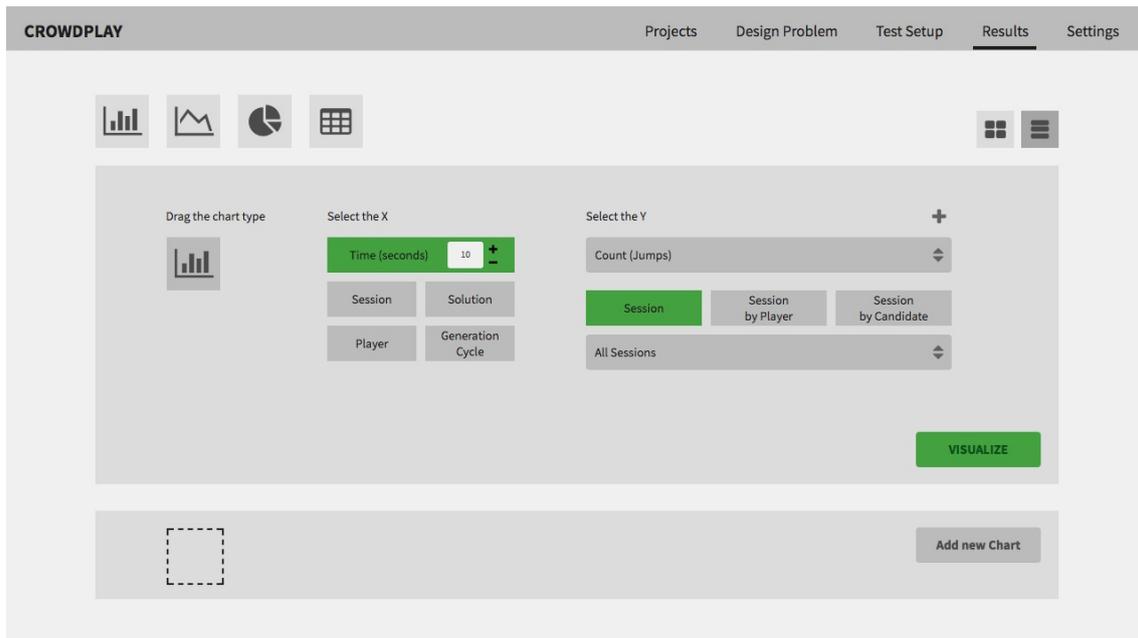
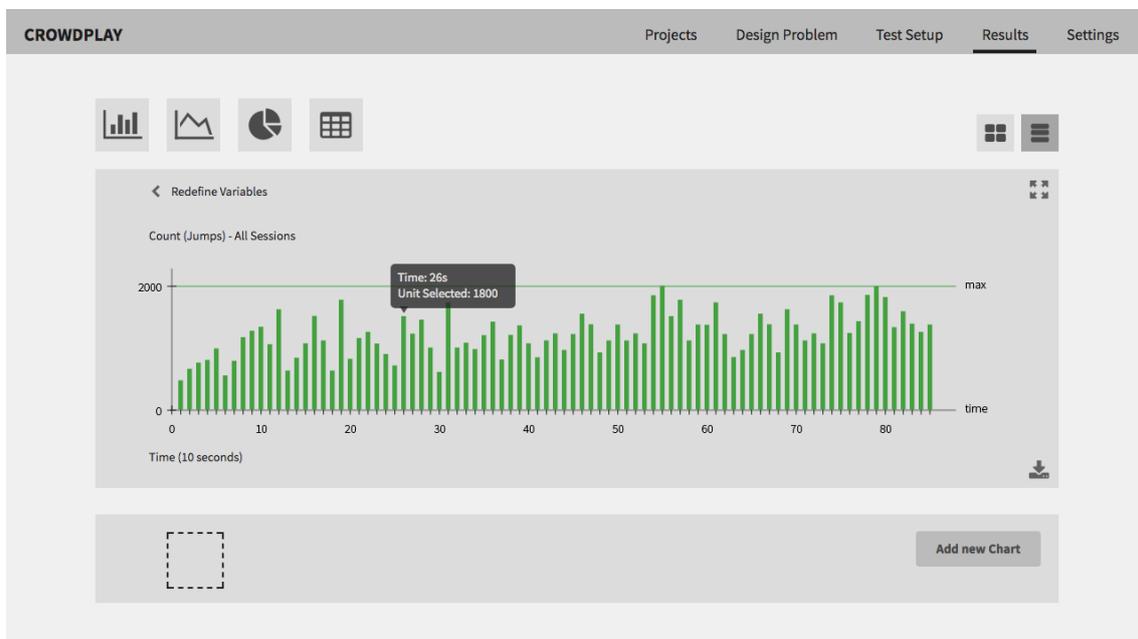


Figura 5.14 - Ecrã com o *status* do teste.

Nas figuras 5.15 a 5.18 podemos observar os ecrãs relativos aos resultados, página esta que mostra os resultados depois de um teste estar concluído. É possível visualizar os resultados pela vista em lista (figuras 5.15 e 5.16) ou pela vista em grelha (figuras 5.17 e 5.18), podendo selecionar o tipo de gráfico entre as seguintes opções; gráfico de barras, histograma, pie chart e tabela. Nestes ecrãs podemos, em primeiro lugar selecionar o tipo gráfico, com a funcionalidade de drag and drop, e depois selecionar as variáveis *x* e *y* que pretendemos visualizar. As opções da variável *y* diferenciam-se consoante a variável *x* escolhida, e é ainda possível adicionar mais do que um *y*, bem como, se a variável *x* for "Time", definir os segundos pretendidos. Ao visualizar o gráfico, é apresentado um ecrã semelhante à figura 5.16, com o gráfico previamente selecionado. Nesta figura é exemplificado o hover para mostrar a informação mais detalhadamente. Já as figuras 5.17 e 5.18 apresentam os ecrãs relativos à vista em grelha com as mesmas funcionalidades.

Por se tratar de um protótipo apenas a segunda área (para um segundo gráfico) está funcional no ecrã com vista em grelha (figura 5.17), demonstrando como aparece sempre um último espaço para adicionar uma nova visualização. Ao preencher esse último espaço, é suposto aparecer uma nova área, para o caso do utilizador pretender visualizar mais dados. É de referir que os dados apresentados nos gráficos são fictícios e meramente exemplificativos.

Figura 5.15 - Ecrã de *Results* para selecionar as variáveis.Figura 5.16 - Ecrã de *Results* com um gráfico de barras.

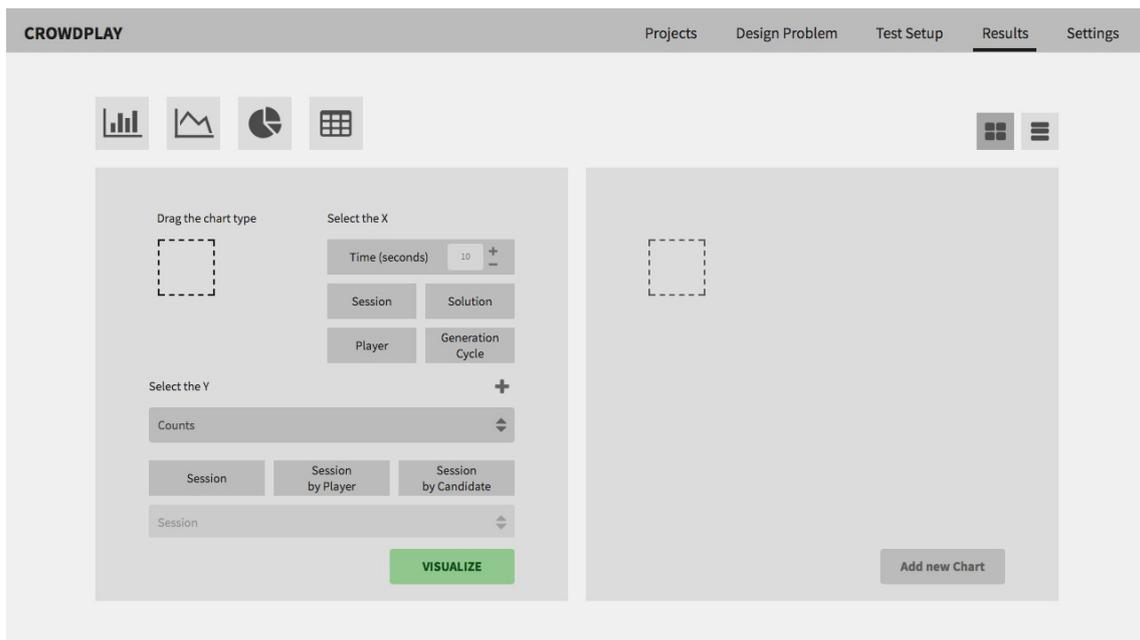


Figura 5.17 - Ecrã de *Results* com a vista em grelha.

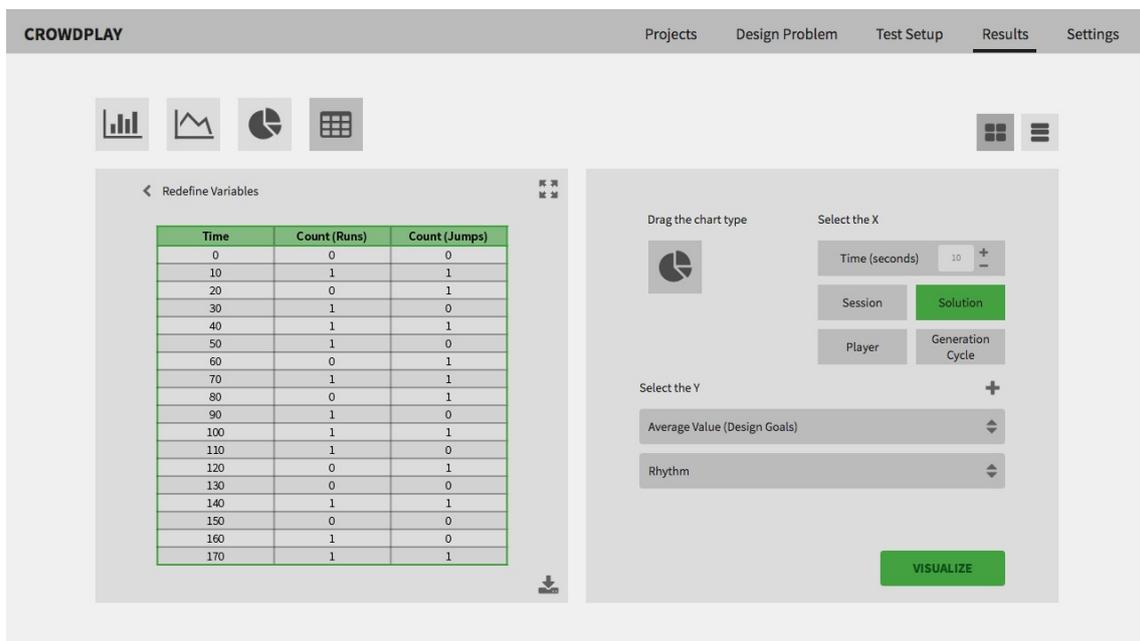


Figura 5.18 - Ecrã de *Results* com uma tabela e uma área para selecionar as variáveis.

As figuras seguintes (5.19 à 5.27) referem-se a secção *Settings*, pelo que é distinta para cada tipo de utilizador.

Os ecrãs relativos ao *admin* apresentam-se nas figuras 5.19 à 5.22, onde é possível ver e editar informações dos projetos e dos utilizadores.

No ecrãs referentes aos projetos (figura 5.19 e 5.20) é possível ver, editar ou adicionar um projeto. São apresentados os testes recentes relativos ao projeto/jogo selecionado, e ainda permite adicionar ou retirar permissões de utilizadores para o projeto em questão.

Nas figura 5.21 e 5.22 são apresentados ecrãs semelhantes relativos aos utilizadores. É de igual modo possível apagar, editar e/ou adicionar utilizadores. Um *admin* pode editar o *username*, o email e o tipo de conta, bem como adicionar ou retirar permissões dos projetos.

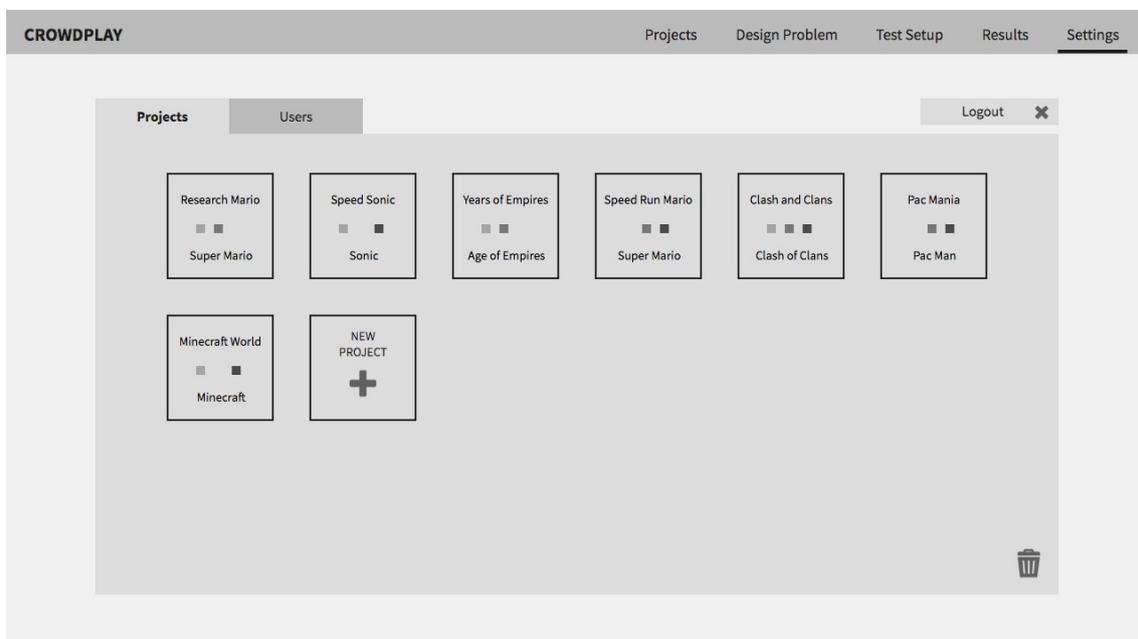


Figura 5.19 - Ecrã de *Settings* relativo a *projects* (*admin*).

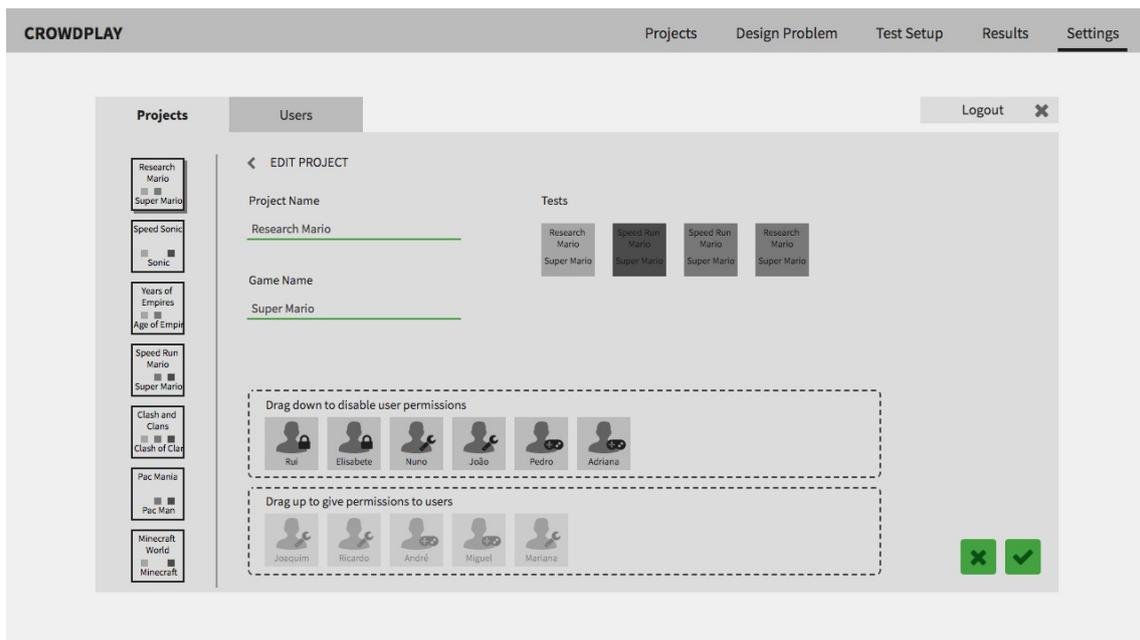


Figura 5.20 - Ecrã de Settings para editar/adicionar projects (admin).

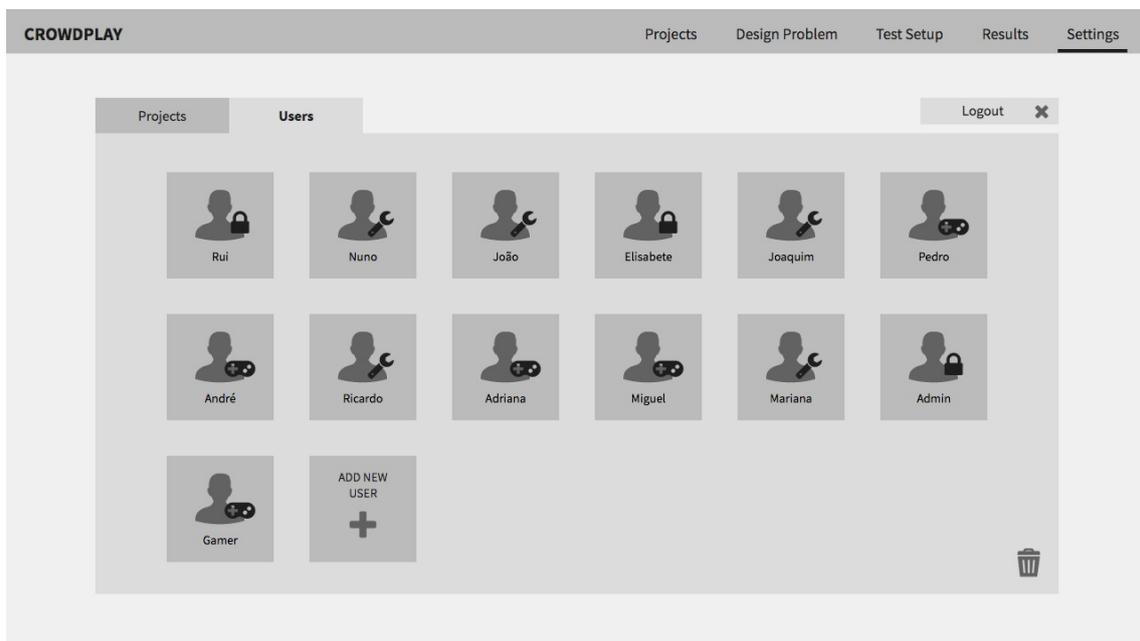


Figura 5.21 - Ecrã de Settings relativo a users (admin).

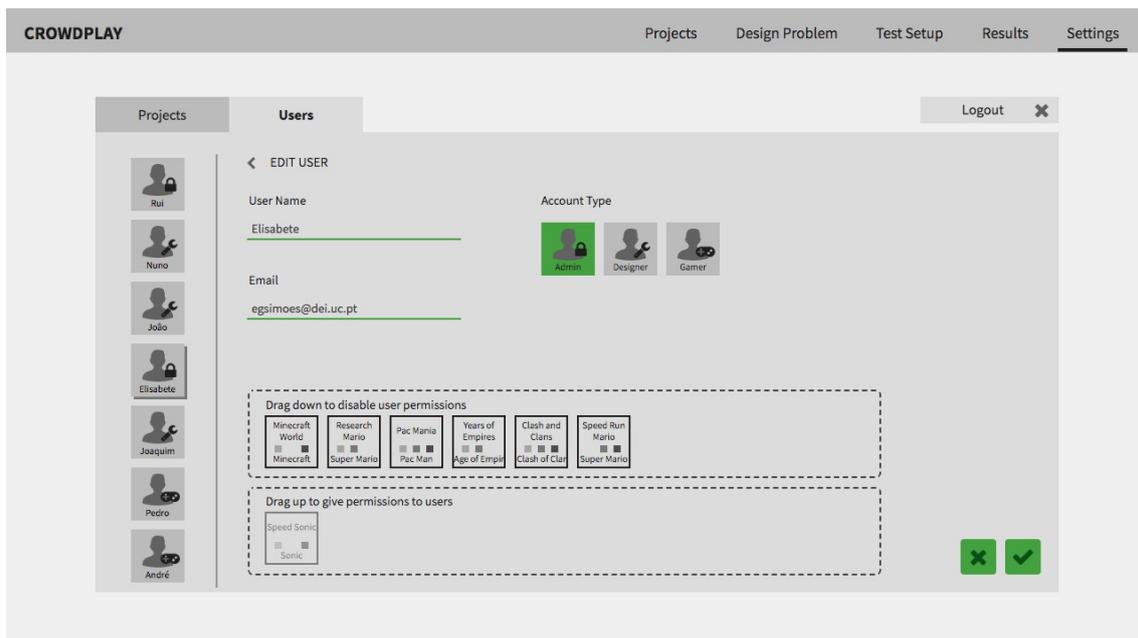


Figura 5.22 - Ecrã de *Settings* para editar/adicionar *users* (*admin*).

Na secção *Settings* destinada aos designers (figuras 5.23 a 5.25), existem algumas diferenças. A primeira página apresentada (figura 5.23) é referente à conta, na qual o designer pode editar o seu nome, email e *password*, e visualizar para que projetos tem permissões. Relativamente a este ecrã é de referir que o utilizador, apesar de poder ver a lista de utilizadores, não tem permissões para os visualizar em mais detalhe.

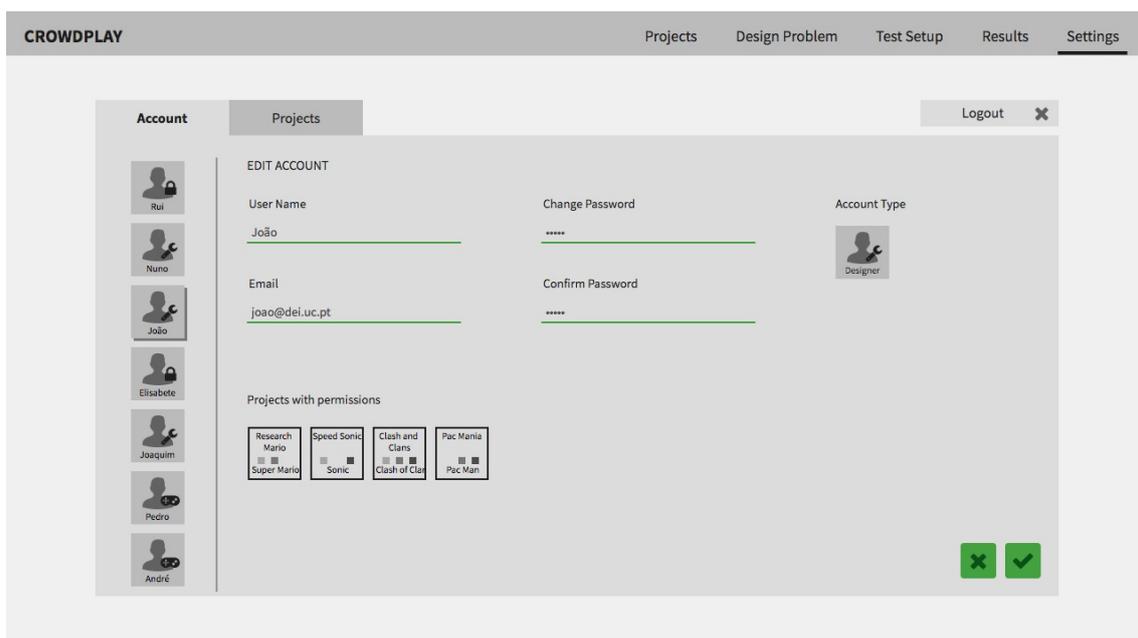


Figura 5.23 - Ecrã de *Settings* para editar *account* (*designer*).

Na página referente aos projetos (figura 5.24), o utilizador pode visualizar os testes recentes relacionados, bem como, pedir permissões (ao *admin*) num determinado projeto, caso não as tenha (figura 5.25).

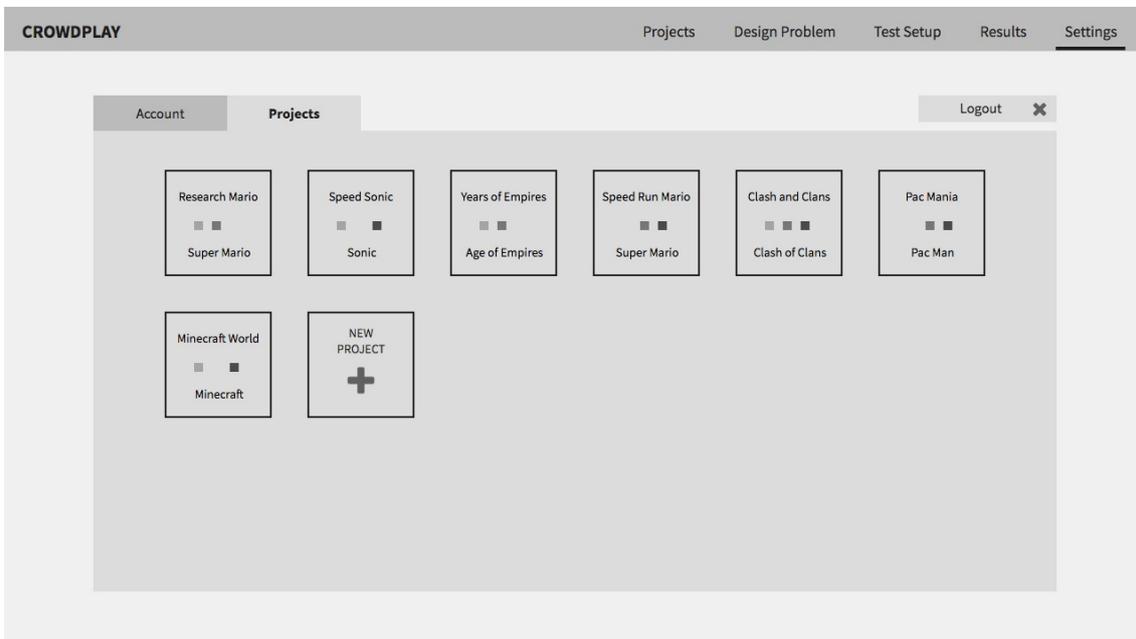


Figura 5.24 - Ecrã de *Settings* relativo a *projects* (designer).

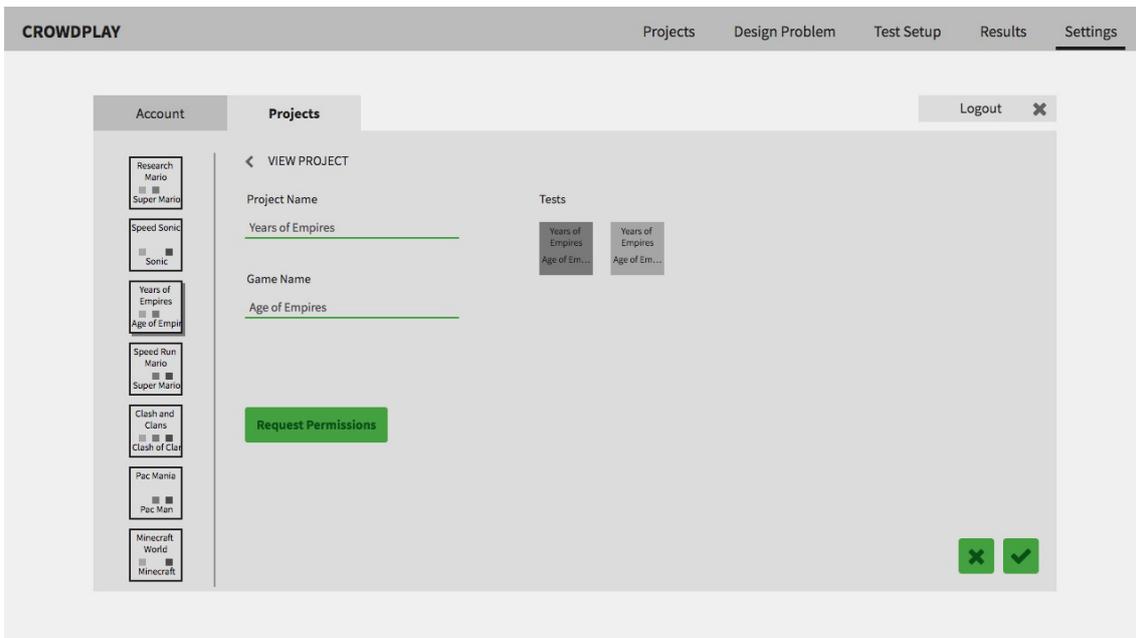


Figura 5.25 - Ecrã de *Settings* relativo a um projeto (designer).

Por fim, nas figuras 5.26 e 5.27 são visíveis os ecrãs relativos ao utilizador *gamer*. Este tem acesso apenas a uma área onde pode descarregar os projetos para os jogar e testar (figura 5.26), e a área *account* idêntica ao do designer, na qual pode editar o nome, email e *password* (figura 5.27).

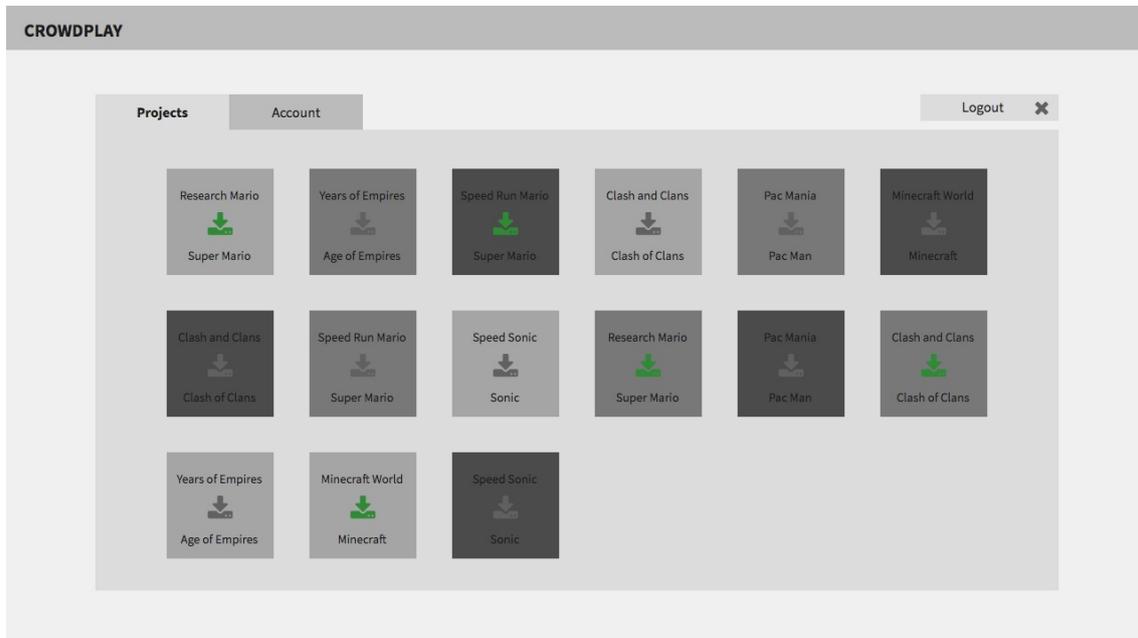


Figura 5.26 - Ecrã de *Projects* (*gamer*).

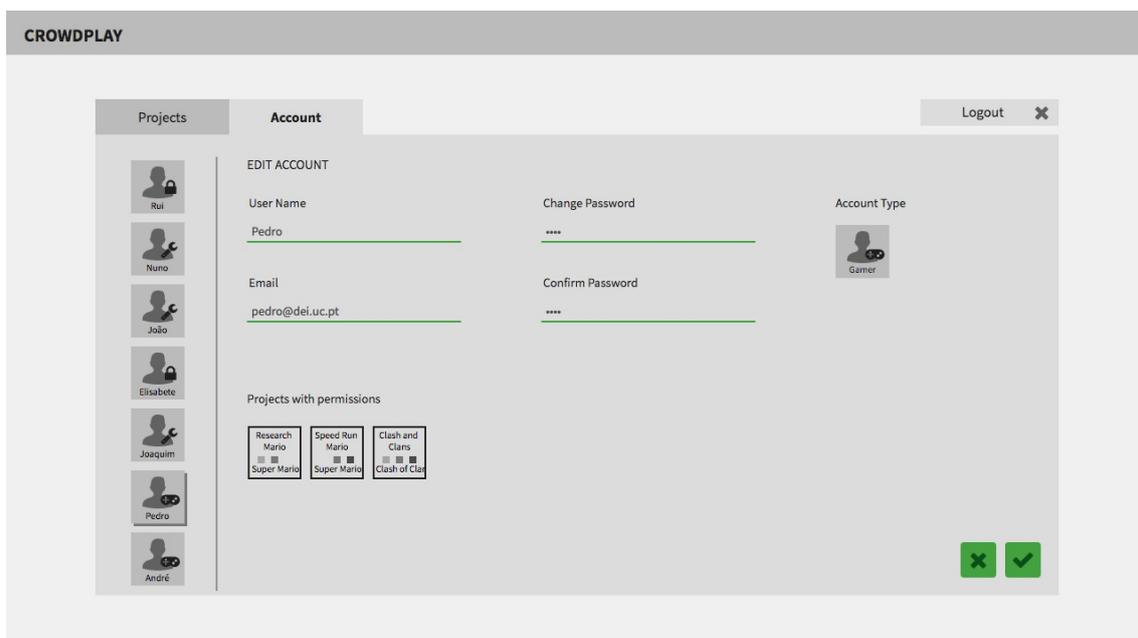


Figura 5.27 - Ecrã *Account* (*gamer*).

Para uma visualização mais abrangente da interface desenvolvida, são apresentados os ecrãs finais no anexo H. De referir que estes se apresentam no estado final, após introduzir as alterações relativas aos testes de usabilidade.

Foi ainda desenvolvido um mapa do website com recurso aos ecrãs desenhados, com o objetivo de se obter uma overview, anexado no final (anexo I). Neste diagrama, foi de igual modo representado apenas o fluxo mais natural e comum ao usar a plataforma.

Em modo de resumo de toda a interface listam-se de seguida os ecrãs com os respetivos requisitos funcionais:

- *Login*
- *Register*
- **Projects**
 - *New Test*
 - *iniciar teste associado a um determinado tipo de teste*
 - *Recent Tests*
 - *continuar teste*
- **Design Problem**
 - *Game Variations*
 - *ativar/desativar e apagar variações de jogo*
 - *editar/adicionar variações de jogo*
 - *selecionar variável, função probabilística, intervalo (...)*
 - *Design Goals*
 - *ativar/desativar e apagar objetivos de design*
 - *editar/adicionar objetivos de design*
 - *selecionar nome, função de pontuação, fórmula (...)*
 - *editar/adicionar e/ou apagar fórmulas*
- **Test Setup**
 - *Setup*
 - *definir (características do) teste*
 - *iniciar, pausar e parar teste*
 - *sumário de Design Problem*
 - *Status*
 - *sumário do teste*
- **Results**
 - *vista List/Grid*
 - *selecionar tipo de gráfico e dados a visualizar*
- **Settings (admin)**

- *Projects*
 - *editar projetos*
 - *editar permissões de utilizadores*
- *Users*
 - *editar utilizadores*
 - *editar permissões a projetos*
- **Settings (designer)**
 - *Account*
 - *editar conta*
 - *Projects*
 - *pedir permissões*
- **(gamer)**
 - *Projects*
 - *descarregar jogos/projetos*
 - *Account*
 - *editar conta*

A interface foi desenhada tendo em conta as dimensões de ecrã de 1280x720, sendo que se ajusta a qualquer tamanho (para uma melhor experiência é preferível manter essa mesma proporção - 16/9, ou seja ecrã completo). Visto a ferramenta Crowdplay ser única e exclusivamente para trabalhar num computador, não foram desenvolvidas mais formas de responsividade da interface para dispositivos móveis.

O protótipo desenvolvido foi testado nos *browsers Google Chrome, Safari e Firefox*, funcionando na totalidade em todos eles.

5.3 Testes de Usabilidade

Após desenvolver o protótipo interativo, foram aplicados os testes de usabilidade afim de testar a interface. Estes testes foram operados de modo idêntico aos testes realizados com a interface anterior iniciando-se com a entrega de documentação explicativa do Crowdplay e do seu funcionamento base (anexo A). Seguiu-se a realização das tarefas de acordo com o guião (anexo J), e por fim a aplicação do questionário adaptado do questionário S.U.S. (anexo C).

Foram planeadas 11 tarefas, algumas com subtarefas (resultando um total de 35) (anexo J). Pelo facto desta interface conter novas funcionalidades e uma navegação diferente, as tarefas sofreram alterações, mantendo-se o mais possível idênticas às anteriores para comparação.

As tarefas foram desenvolvidas uma a uma; todas foram temporizadas e foi pedido ao utilizador para reportar a dificuldade da sua execução numa escala de um a cinco. Questionou-se sobre a existência de comentários, e foram anotadas algumas observações. No desenvolvimento dos testes foram, de igual modo, feitas gravações de ecrã com áudio, para facilitar e melhorar a análise dos testes.

Antes de realizar os testes com os utilizadores, foram analisadas as tarefas, tal como nos testes de usabilidade desenvolvidos à primeira interface.

Foi calculado o tempo máximo expectável, a dificuldade expectável, assim como descrito as tarefas onde poderiam surgir dificuldades. Mesmo antes de desenvolver os testes sabe-se que algumas tarefas podem ser de difícil compreensão por falta de informação ou ajuda inicial.

Tem-se a noção que nesta fase a plataforma necessita de um tutorial ou ajuda inicial para explicar determinadas ações. Estes testes de usabilidade estão em conformidade com este facto para ajudar a definir a informação que deverá conter esse tutorial.

De seguida podemos ver a lista de tarefas nas quais se especula que possam surgir dúvidas e/ou dificuldades:

T2 - Iniciar um teste do projeto “Research Mario” do tipo “Optimize”

- O utilizador não ler e/ou compreender que é pra arrastar.

T4 - Editar a variação de jogo “Coins”

- O utilizador pode demorar a perceber que para abrir a variação de jogo é necessário duplo clique.

T5.4 - Selecionar a fórmula com o nome “Number_Coins/Power Ups” e guardar o objetivo de design

- Dificuldade em entender que é para arrastar a fórmula.

Relativamente aos *testers*, optou-se por voltar a aplicar os testes de usabilidade tanto aos que desenvolveram os primeiros testes, como a novos utilizadores. Os utilizadores/*testers* são estudantes de Mestrado em Design e Multimédia (MDM), Mestrado em Engenharia Informática (MEI), Licenciatura em Engenharia Informática (LEI) e Doutoramento em Ciências e Tecnologias da Informação (PDCTI), com idades compreendidas entre os 20 e os 28, representados na tabela (figura 5.28). Nesta tabela podemos observar que os 5 primeiros utilizadores correspondem aos dos testes de usabilidade iniciais. Alguns utilizadores passaram a estar familiarizados com a plataforma Crowdplay por a testarem e/ou trabalharem com ela.

Utilizador	Formação / Ocupação	Crowdplay	Desenvolvimento de Jogos
U1	MDM	✓*	✓
U2	MEI	✓*	✓
U3	MEI	✓**	✓
U4	MDM	✓*	✓
U5	LEI	✓*	✓
U6	MDM	✗	✗
U7	MDM	✗	✗
U8	PDCTI	✓**	✓
U9	MDM	✓*	✓
U10	MEI	✗	✓
U11	MEI	✗	✓

Legenda:

- ✓ - Familiarizado com
- ✗ - Não familiarizado
- * - Utilizadores que testaram a plataforma (incluindo também na dissertação anterior)
- ** - Utilizadores que trabalham com a plataforma

Figura 5.28 - Tabela de utilizadores (2º testes de usabilidade).

Depois do desenvolvimento dos testes de usabilidade, foram elaborados os gráficos com os tempos e dificuldades de cada tarefa. Podemos então observar os gráficos com os tempos e dificuldades por tarefa com todos os utilizadores, distinguindo os iniciais dos novos utilizadores, nas figuras 5.29 e 5.30 respetivamente. Para aprofundar a análise, inclui-se o tempo/dificuldade expectáveis e a média do tempo/dificuldade dos utilizadores, podendo assim facilmente observar-se quais as tarefas que ultrapassaram o pretendido.

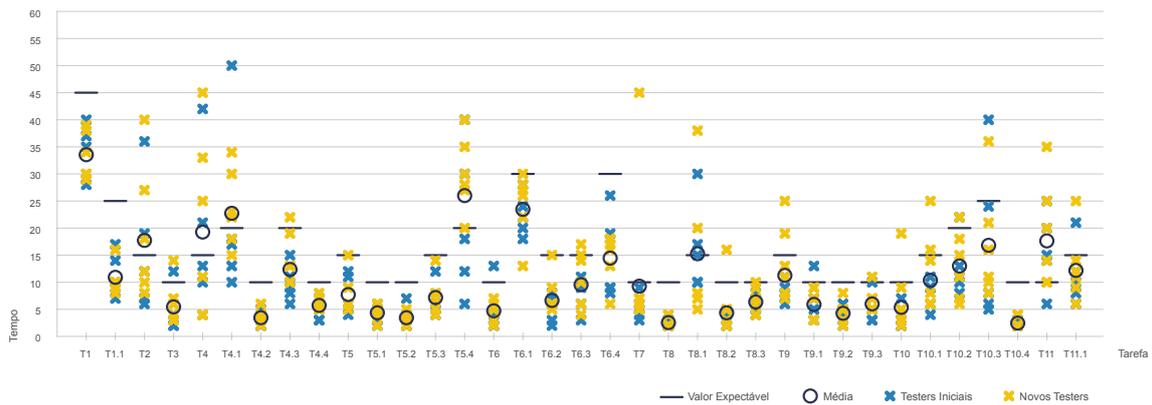


Figura 5.29 - Gráfico do tempo por tarefa de todos os utilizadores.

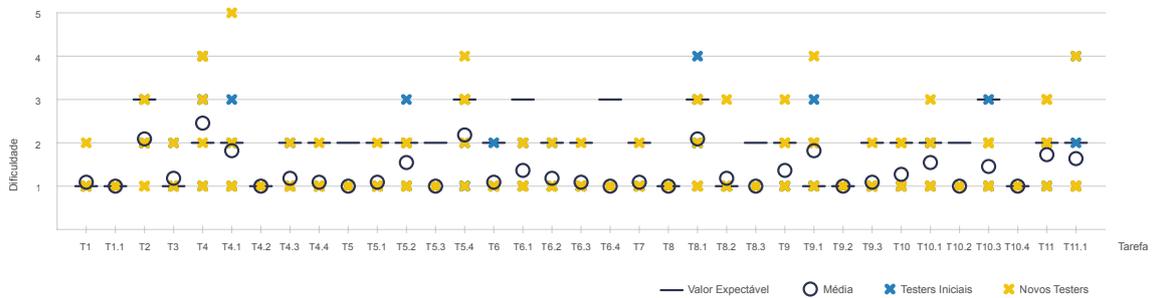


Figura 5.30 - Gráfico da dificuldade por tarefa de todos os utilizadores.

De seguida apresenta-se a análise dos resultados obtidos em cada tarefa, na qual são descritas as ocorrências, assim como o modo de execução da tarefa.

T1 - Registrar com a seguinte conta

Tarefa conseguida com sucesso por parte de todos utilizadores, como podemos observar no gráfico 5.29, pois todos completaram a tarefa dentro do tempo expectável. Podemos concluir que esta tarefa foi corrigida com sucesso, pois nos testes de usabilidade à plataforma anterior, esta era uma tarefa onde surgiram alguns problemas.

T1.1 - Entrar com a respetiva conta

Tarefa conseguida com sucesso por parte de todos utilizadores.

T2 - Iniciar um teste do projeto “Research Mario” do tipo “Optimize”

Como se esperava, nesta tarefa surgiram algumas dúvidas e falhas iniciais. Alguns utilizadores não leram e demoraram a perceber que a tarefa se efetuava arrastando o projeto. Três dos onze utilizadores assumiram que não leram a informação e outros três tentaram clicar de forma a seleccionar. Apesar de alguns utilizadores demorarem mais tempo que o expectável (figura 5.29), não se considera que esta seja uma tarefa difícil, como se comprova na figura 5.30. Considera-se que este problema acontece apenas por ser a primeira vez que os utilizadores lidam com a plataforma, e que possa ser resolvido com o tutorial inicial.

T3 - Ativar a variação “Power Ups” nas variações de jogo (Game Variations)

Tarefa conseguida com sucesso por parte de todos utilizadores.

T4 - Editar a variação de jogo “Coins”

Nesta tarefa surgiu alguma dificuldade como se previa. Alguns utilizadores demoraram a compreender a ação; três deles, ao explorar, ativaram e desativaram a variação; dois tentaram com o clique direito do rato, ou ainda a procura de algum botão para editar.

Tal como na tarefa T2, considera-se que este problema acontece por ser a primeira vez a lidar com a plataforma e que possa ser resolvido com o tutorial inicial e com uma aprendizagem rápida.

T4.1 - Alterar a variável para “Number of Coins” (“Level Design”)

Tarefa conseguida com sucesso por parte de todos os utilizadores; no entanto surgiu alguma hesitação na procura. A dica dada (a dizer que a variável se encontrava em “Level Design”) levou a algumas hesitações na procura pelo pretendido. Dois utilizadores clicaram duas vezes na variável pedida assumindo que a poderiam seleccionar desse modo. Nesta tarefa, apesar de o tempo médio ter ultrapassado o expectável (figura 5.29), considera-se que no geral foi concluída com sucesso. Nos testes de usabilidade à plataforma inicial existiram alguns problemas que agora foram superados.

T4.2 - Selecionar o gráfico do tipo “Triangular”

Tarefa conseguida com sucesso por parte de todos utilizadores. Como podemos observar nas figuras 5.29 e 5.30, o tempo de execução da tarefa foi baixo, assim como a dificuldade atribuída. Conclui-se que esta tarefa foi corrigida com sucesso, pois nos testes de usabilidade à plataforma anterior, esta era uma tarefa onde surgiu alguma dúvida.

T4.3 - Definir para variar entre 50 e 450

Tarefa conseguida com sucesso por parte de todos utilizadores. De referir que sete dos utilizadores utilizaram a escrita para definir os números enquanto os outros quatro usaram o *drag*.

T4.4 - Com um incremento de 5 e guardar a variação de jogo

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5 - Adicionar um objetivo de design no “Design Goals”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5.1 - Com o nome “Collector”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5.2 - Selecionar o gráfico do tipo “Step”

Tarefa conseguida com sucesso por parte de todos utilizadores, no entanto quatro utilizadores sentiram algumas dúvidas sobre qual seria o “Step”. Para resolver este problema inicial sugere-se um *tooltip*; embora não seja um recurso de primeira linha, poderia funcionar como ajuda adicional.

T5.3 - Definir para variar entre 90 e 110

Tarefa conseguida com sucesso por parte de todos utilizadores.

T5.4 - Selecionar a fórmula com o nome “Number_Coins/Power Ups” e guardar o objetivo de design

Nesta tarefa surgiram algumas dificuldades, pois os utilizadores não perceberam imediatamente que era para arrastar e alguns, ao explorar, clicaram na fórmula, abrindo-a. Isto aconteceu a três utilizadores e levou-os a pensar que a tarefa já estava concluída. Apesar do tempo médio ultrapassar o expectável, esta não é uma tarefa problemática.

T6 - Editar o objetivo de design “Rhythm”

Nesta tarefa, muito idêntica a T4, a maioria dos utilizadores tinha aprendido a forma de abrir as variações de jogo. Apenas um utilizador não se lembrou/ associou que se procedia do mesmo modo.

T6.1 - Adicionar uma nova fórmula com o nome “Level_Completion_Time”

Os utilizadores demoraram ligeiramente, sem ultrapassar o tempo pretendido (figura 5.29), a encontrar o botão para adicionar a fórmula, por esta, segundo referido (quatro utilizadores), estar pouco destacável e confundir-se com as outras fórmulas. Comparativamente à tarefa T5.2 dos testes de usabilidade à plataforma anterior onde surgiram algumas dificuldades, atualmente, apesar de ainda surgir alguma dificuldade a encontrar o botão para adicionar nova fórmula, esta é mínima. Conclui-se também que a tarefa agora está mais simples e clara para o utilizador.

T6.2 - Selecionar a função “MAX” e selecionar “Time Engine” para a “Target Variable”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T6.3 - Definir a condição “Event = Won” e guardar a fórmula

Tarefa conseguida com sucesso por parte de todos utilizadores.

T6.4 - Definir a fórmula para calcular “Number_Actions” a dividir por “Level_Completion_Time”

Tarefa conseguida com sucesso por parte de todos utilizadores.

Esta tarefa é similar às tarefas T5.4 e T6.3 dos testes de usabilidade à plataforma anterior, nas quais surgiram dificuldades. Pode concluir-se, portanto, que a tarefa foi corrigida com sucesso, pois a partir do momento em que os utilizadores percebem que a ação é arrastar as fórmulas, concluem a tarefa com sucesso (em termos de tempo e dificuldade - figuras 5.29 e 5.30, respetivamente) e compreendem como criar e definir qualquer tipo de fórmula.

T7 - Apagar o objetivo de design “Killer”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T8 - Ir para a página do Test Setup

Tarefa conseguida com sucesso por parte de todos utilizadores.

T8.1 - Desativar o objetivo de design “Collector”

Nesta tarefa surgiram alguns erros e comentários. Três utilizadores tentaram clicar para desativar e houve comentários sobre o facto de não ser do mesmo modo que nas variações de jogo/objetivos de design. Na figura 5.29, constatou-se que o tempo médio de execução não se afasta do expectável, contudo, esta é uma tarefa que requer atenção futura para a correção de incoerências.

T8.2 - Definir a condição de fim de teste para “Manual”

Tarefa conseguida com sucesso por parte de todos utilizadores. Apesar de um utilizador ter pontuado com dificuldade elevada (figura 5.30), tal resultado não é significativo.

T8.3 - Iniciar teste, pausar e pará-lo

Tarefa conseguida com sucesso por parte de todos utilizadores. Comparativamente à tarefa T8 dos testes de usabilidade à plataforma anterior, esta era uma tarefa onde surgiu alguma dificuldade em encontrar o botão para iniciar o teste.

T9 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

Esta tarefa foi clara para a maioria dos utilizadores, apenas dois deles clicaram inicialmente, não percebendo de imediato que seria para arrastar. Comparando com a tarefa T11 dos testes de usabilidade à plataforma inicial, conclui-se que as dificuldades encontradas foram praticamente corrigidas, surgindo apenas algumas dúvidas por breves segundos.

T9.1 - Escolher para o x “Time (10 seconds)”

Quatro dos onze utilizadores pensaram que a tarefa já estava concluída por estar definido *a priori* 10 segundos, tendo surgido algumas dúvidas sobre se teriam de selecionar o botão “Time”, atribuindo uma dificuldade extra (figura 5.30).

T9.2 - Escolher para o y “Count (Jumps)”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T9.3 - Escolher para todas as Sessões e visualizar gráfico

Tarefa conseguida com sucesso por parte de todos utilizadores. Em comparação com a tarefa T11.2 dos testes de usabilidade da plataforma inicial, conclui-se que o problema foi resolvido.

T10 - Alterar a vista para “Grid” (em vez de “List”)

Esta tarefa foi simples e clara para a maioria dos utilizadores, no entanto, apenas para um utilizador surgiu alguma dúvida e demora a encontrar o pretendido, como se pode conferir na figura 5.29.

Comparando com a tarefa idêntica a esta, a T12 dos testes à plataforma inicial, pensa-se ter resolvido o problema.

T10.1 - Escolher visualizar uma Tabela (Data Table) com o x igual a “Player”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T10.2 - Escolher para o y “Average Value (Design Goals)” para o Design Goal “Time”

Tarefa conseguida com sucesso por parte de todos utilizadores.

T10.3 - Adicionar um segundo y com “Average Value (Design Goals)” para o Design Goal “Rhythm”

Tarefa conseguida com sucesso pela maioria, na qual apenas dois utilizadores necessitaram de mais tempo para encontrar onde adicionar o segundo “y” (figura 5.29). Relativamente à tarefa T12.2 dos testes de usabilidade à plataforma anterior, tarefa semelhante a esta, conclui-se que atualmente ela se encontra muito mais simples e clara para os utilizadores.

T10.4 - Visualizar o gráfico

Tarefa conseguida com sucesso por parte de todos utilizadores.

T11 - Nos Settings, ver para que projetos se tem permissões

Nesta tarefa surgiram alguns problemas também pelo facto de a página demorar a carregar, nomeadamente a zona que continha a informação pedida, levando os utilizadores a procurar no separador *Projects*. Com isto, os utilizadores demoram mais tempo que o expectável, no entanto não se considera uma tarefa problemática.

T11.1 - Pedir permissões para o projeto “Years of Empires”

Esta tarefa foi simples para a maioria dos utilizadores, no entanto, dois utilizadores sentiram algumas dificuldades. Um deles clicou na página inicial dos Projetos e o outro não percebeu que tinha de abrir o projeto para o qual pretendia pedir permissões, pensando que já as tinha.

Optou-se também por apresentar os gráficos com o tempo e dificuldade por tarefa de cada grupo de utilizadores, para uma análise mais discriminativa. Podemos então observar os gráficos dos utilizadores iniciais e dos novos utilizadores, respectivamente nas figuras. 5.31 e 5.32 para os dados do tempo e figuras 5.33 e 5.34 para os dados da dificuldade.

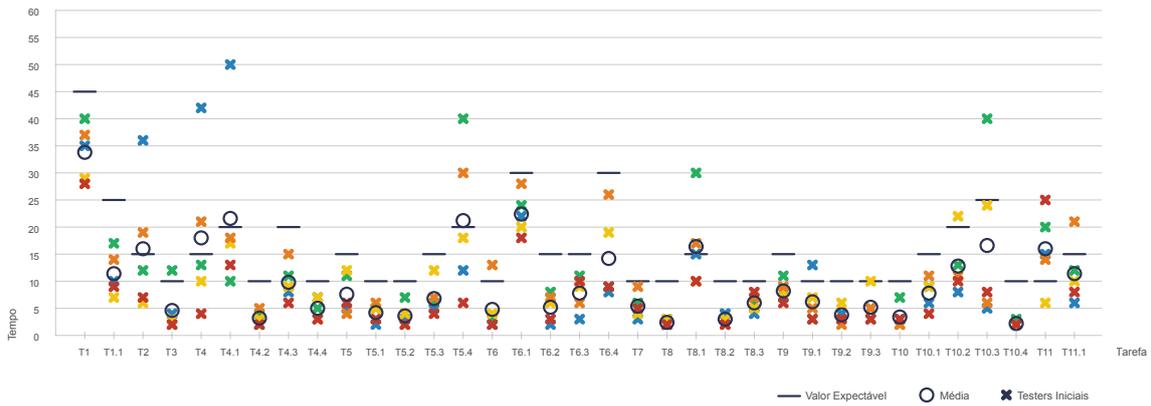


Figura 5.31 - Gráfico do tempo por tarefa dos utilizadores iniciais.

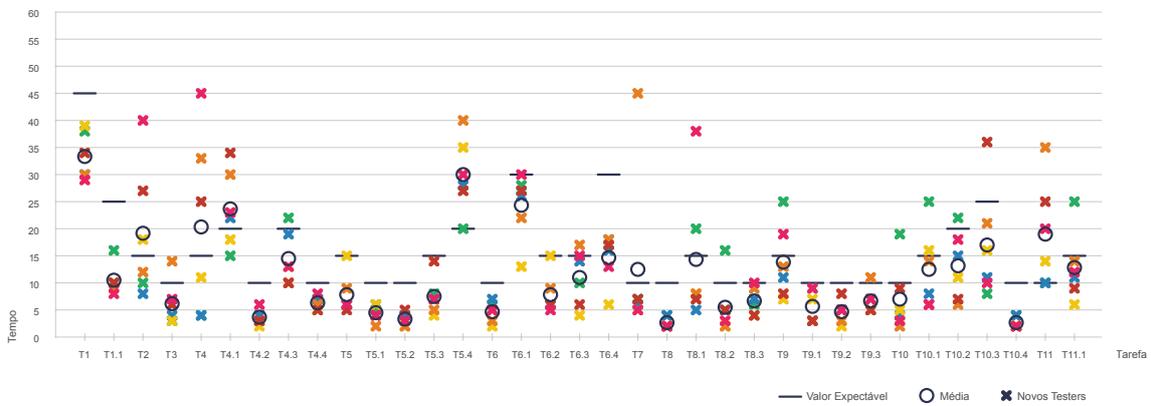


Figura 5.32 - Gráfico do tempo por tarefa dos novos utilizadores.

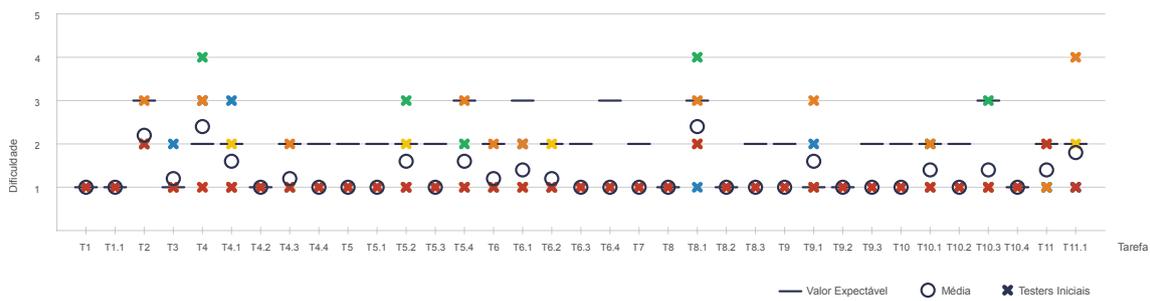


Figura 5.33 - Gráfico da dificuldade por tarefa dos utilizadores iniciais.

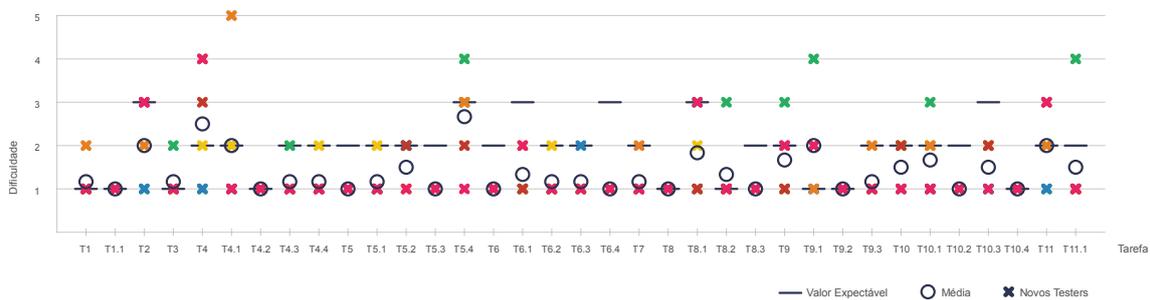


Figura 5.34 - Gráfico da dificuldade por tarefa dos novos utilizadores.

Conclui-se que as tarefas onde surgiram mais problemas, tendo em conta também as médias dos tempos que ficaram acima do expectável (apresentado no gráfico com todos os utilizadores - figura 5.29), são as seguintes:

- T2 - Iniciar um teste do projeto "Research Mario" do tipo "Optimize"*
- T4 - Editar a variação de jogo "Coins"*
- T4.1 - Alterar a variável para "Number of Coins" ("Level Design")*
- T5.4 - Selecionar a fórmula com o nome "Number_Coins/Power Ups" e guardar o objetivo de design*
- T8.1 - Desativar o objetivo de design "Collector"*
- T11 - Nos Settings, ver para que projetos se tem permissões*

Considera-se que as dificuldades associadas às tarefas T2 e T4 se resolvem com um tutorial inicial. Nas tarefas T2, T4, T4.1 e T5.4 as dificuldades detetadas foram sobretudo iniciais e momentâneas, pelo que se considera que o utilizador aprenda facilmente após a primeira experiência.

Na tarefa T8.1, admite-se que existe alguma incoerência na interface e pondera-se uma possível alteração, que será descrita no subcapítulo seguinte.

Por fim, quanto à tarefa T11, o protótipo teria um erro nesta fase, mostrando os projetos com permissões de forma aleatória, o que dificultou a execução da tarefa. Este problema já se encontra resolvido.

Relativamente ao questionário (ver anexo C), no qual era pedido aos utilizadores que respondessem a uma série de questões (sete de resposta fechada com intervalo de 1 a 5, e duas de resposta aberta) podemos agora observar os resultados (figuras 5.35, 5.36 e 5.37).

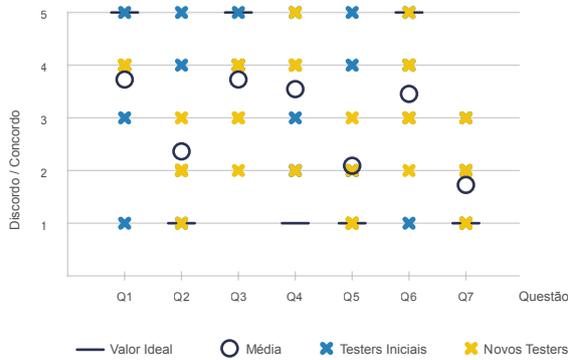


Figura 5.35 - Respostas ao questionário de todos os utilizadores.

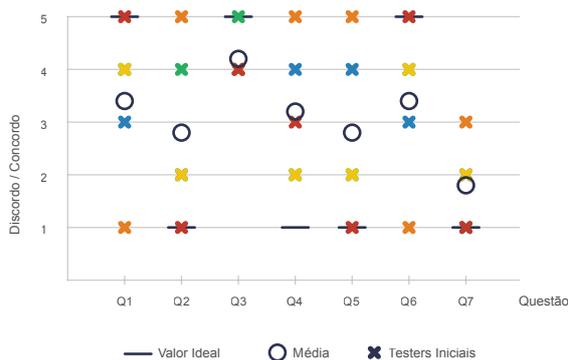


Figura 5.36 - Respostas ao questionário dos utilizadores iniciais.

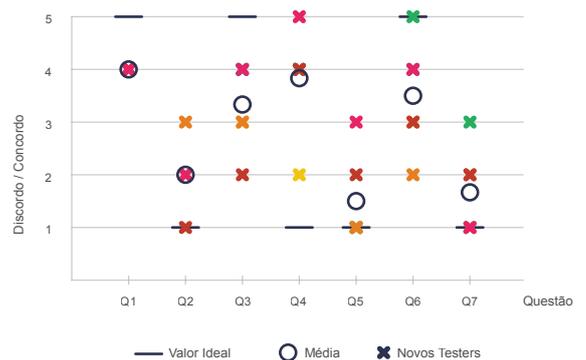


Figura 5.37 - Respostas ao questionário dos novos utilizadores.

Comparativamente ao questionário dos testes de usabilidade à plataforma anterior, em que as questões, cuja resposta média se afasta (mais de dois pontos) da resposta ideal, eram as Q2, Q3, Q4, Q6 e Q7, no questionário atual apenas a questão Q4 se afasta do valor ideal.

A questão número 4 - precisava de aprender algumas coisas antes de começar a usar este sistema - sendo a única que se afasta do ponto ideal, reforça a ideia de que o funcionamento base do sistema é complexo e ainda pouco explorado, reforçando-se a necessidade de tutoriais.

Quanto às perguntas de resposta aberta, na qual era perguntado qual a tarefa mais difícil, as respostas assumem categorias idênticas às descritas anteriormente para as tarefas onde surgiram mais problemas:

T4 - Editar a variação de jogo "Coins"

T4.1 - Alterar a variável para "Number of Coins" ("Level Design")

T5.4 - Selecionar a fórmula com o nome "Number_Coins/Power Ups" e guardar o objetivo de design

T8.1 - Desativar o objetivo de design "Collector"

T11.1 - Pedir permissões para o projeto "Years of Empires"

Analisando as sugestões e comentários referidas identificam-se potencialidades e sugestões de melhoria.

Foi referido que a plataforma "está bastante suave a nível visual", mais intuitiva, apelativa e clara, bem como que houve "uma melhoria ao design". Ainda descrito que "é bastante intuitivo alcançar determinado objetivo mesmo para quem não está familiarizado com jogos nem desenvolvimento dos mesmos" e que "é divertido". Surgiram também algumas sugestões no que diz respeito aos "drags", pois dois dos onze utilizadores, sugeriram minimizá-los, e "resolver erros de incoerências" em determinadas ações (arrastar para cima/baixo para ativar/desativar - tarefa T8.1).

Foi ainda descrito que apesar de ser uma plataforma "fácil de entender, no entanto, é necessário um tutorial no início para se compreender por completo as várias tarefas".

Conclusões dos Testes

Das dificuldades e problemas encontrados nos testes de usabilidade à plataforma inicial, e descritos anteriormente, foram vários os que foi possível corrigir e/ou melhorar tal como mostram os resultados obtidos nos testes de usabilidade a esta proposta. Exemplo disso são as ações de registar, seleccionar o tipo de gráfico, iniciar teste e as diversas tarefas na área dos resultados. Já as tarefas como seleccionar a variável de jogo, editar e definir fórmulas foram, sem dúvida, melhoradas e simplificadas, tornando-as mais claras e simples para o utilizador.

Comparando os dois testes de usabilidade, podemos ainda concluir que tanto o tempo como a dificuldade atribuída a cada tarefa reduziram.

Nos testes à plataforma inicial, o tempo máximo rondava os 100 segundos enquanto nos testes à plataforma proposta ronda os 50 segundos, assim como reduziu o tempo médio e a dificuldade de todas as tarefas iguais ou semelhantes.

Do mesmo modo que conseguimos reduzir o tempo e dificuldade atribuída a cada tarefa dos primeiros testes de usabilidade para os testes à plataforma proposta, conseguimos também melhorar os dados relativamente ao questionário, surgindo também mais sugestões e comentários positivos.

Há ainda que referir que parte dos problemas deve-se ao facto de os utilizadores estarem a lidar com a plataforma pela primeira vez e esta ainda não conter um tutorial ou ajuda inicial.

5.4 Alterações

Após o desenvolvimento e análise dos testes de usabilidade, a plataforma foi sujeita a algumas alterações tendo em conta o resultado dos mesmos. Foram analisadas as tarefas com problemas e nas quais fazia sentido proceder a algumas alterações. Entre as correções, para além das relativas aos testes de usabilidade, foi também desenvolvido o tutorial inicial e ainda algumas outras correções à interface.

A tarefa T8.1 na qual era pedido aos utilizadores para desativar um objetivo de design na página de *Test Setup*, surgiu não só alguma dificuldade como também hesitação.

Pelo facto de na página de *Design Problem* as variações de jogo e objetivos de design se ativarem/desativarem com um clique, na página de *Test Setup* alguns utilizadores assumiram que também seria do mesmo modo; ao invés de arrastar para ativar/desativar as variações de jogo e/ou objetivos de design. Posto isto, decidiu-se então proceder a alteração e desenvolver esta ação com a mesma interação que da página de *Design Problem*, mantendo a coerência e a consistência, tratando funcionalidades semelhantes de forma coerente. Por este sistema estar também implementado em outras seções, procedeu-se de igual modo à correção para que tudo ficasse coerente.

Nas figuras seguintes podemos observar as alterações, na qual a figura 5.38 representa a interface antes dos testes de usabilidade e as restantes (figuras 5.39, 5.40 e 5.41) representam a plataforma depois das alterações.

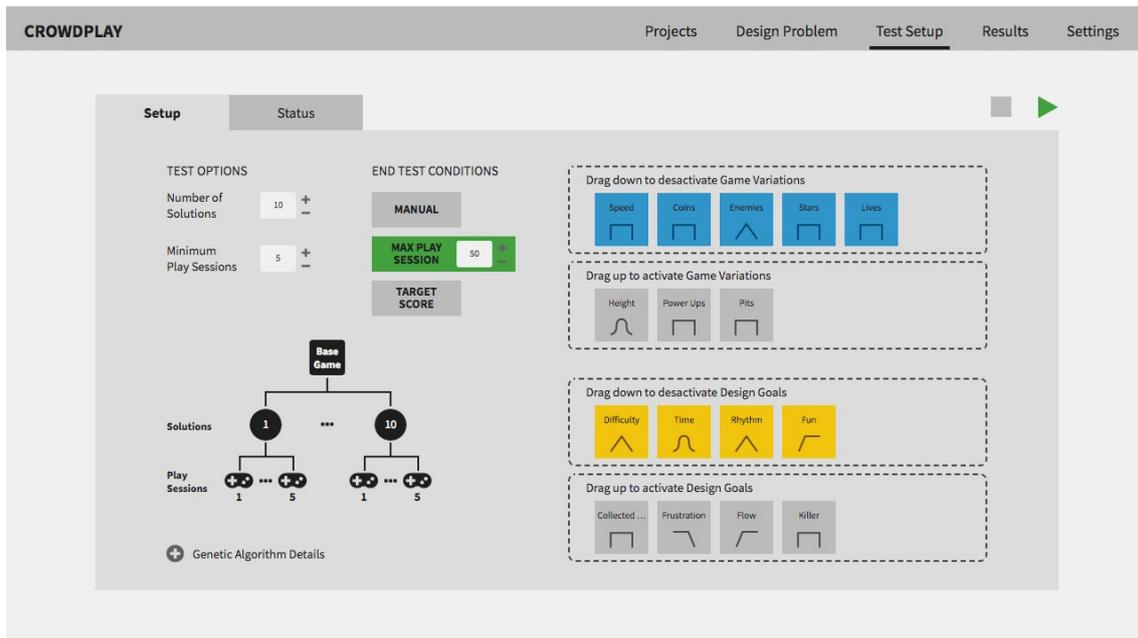


Figura 5.38 - Ecrã *Test Setup* antes das alterações.

Na figura anterior (5.38) podemos observar o sistema para ativar/desativar as variações de jogo e os objetivos de design aquando dos testes de usabilidade, e na figura 5.39 após as alterações efetuadas. Deste modo, no sistema de ativar/desativar as variações de jogo e os objetivos de design procede-se de igual modo às páginas de *Design Problem*, mas neste caso em “miniaturas”, traduzindo-se assim, num modo mais simples e coerente. Para além disso o utilizador já tem conhecimento prévio, adquirido em páginas anteriores.

Visto o mesmo sistema de ativar/desativar, se encontrar em outras páginas, procedeu-se também a essa alteração.

É o caso de ativar/desativar as permissões de utilizadores para um determinado projeto na página de *Settings* em *Projects* e em *Users* para ativar/desativar permissões dos projetos para um determinado utilizador (figuras 5.40 e 5.41).

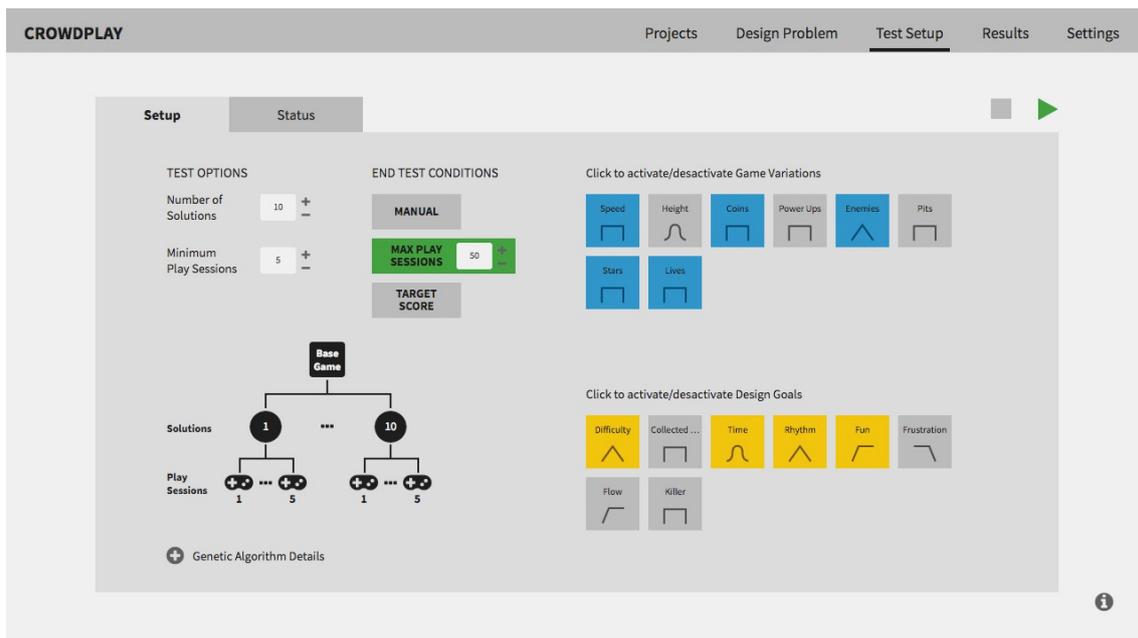


Figura 5.39 - Ecrã *Test Setup* com as alterações referentes ao *Design Problem*.

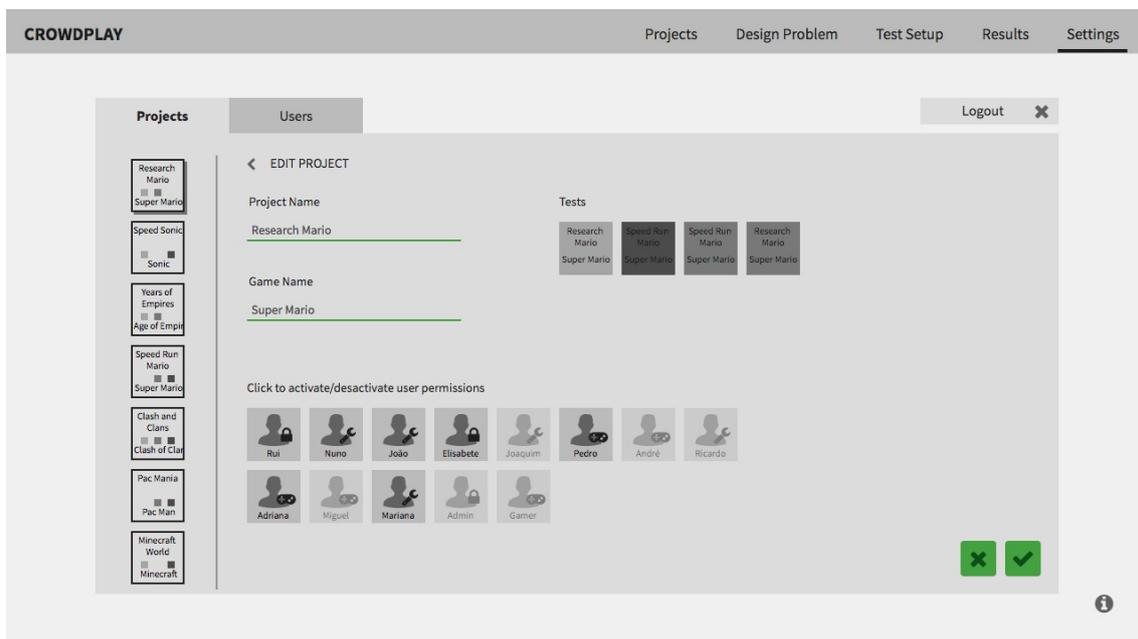


Figura 5.40 - Ecrã *Settings - Projects* com as alterações referentes às permissões de utilizadores.

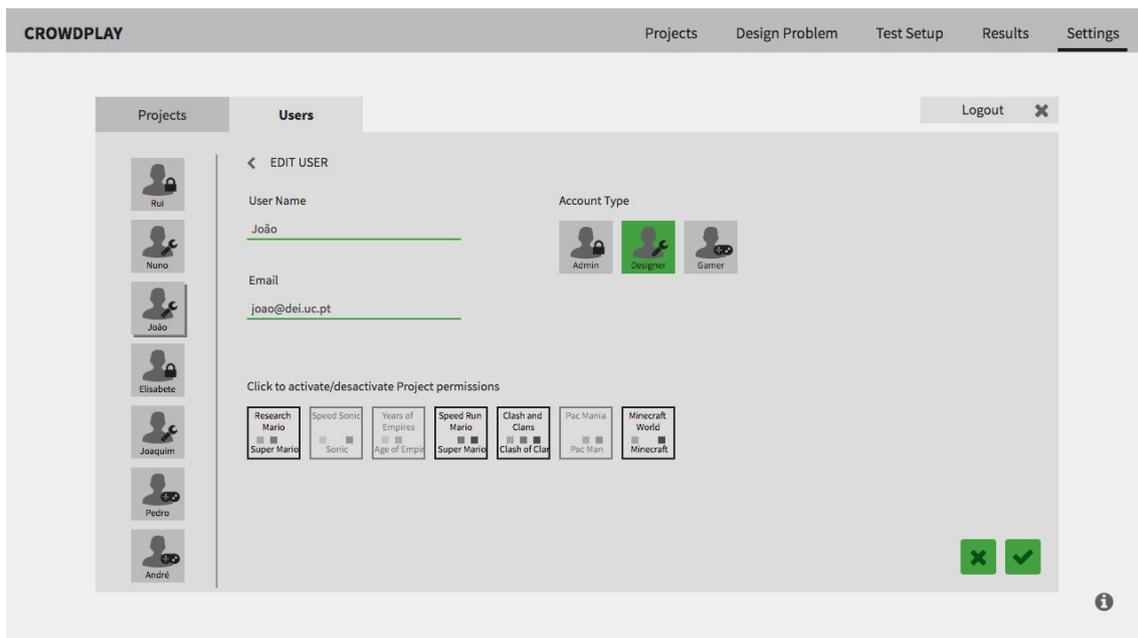


Figura 5.41 - Ecrã *Settings - Users* com as alterações referentes às permissões dos projetos.

Ainda no decorrer dos testes de usabilidade, na observação da execução das tarefas, surgiram algumas dúvidas relativas aos nomes de determinados gráficos. Para resolver esse problema optou-se por apenas acrescentar o *tooltip* (padrão do *justinMind*) para solucionar a dúvida inicial. Como não é uma questão muito problemática pensa-se que este *tooltip* resolva os casos espontâneos de quando algum utilizador ainda não conheça a plataforma na sua totalidade.

Esta informação foi não só adicionada às páginas para editar uma variação de jogo ou um objetivo de design, como também na página de *Results* nos elementos para selecionar o gráfico a visualizar ou nos botões para selecionar a vista dos resultados.

Outra alteração efetuada, para melhorar o *feedback* ao utilizador foi, aquando do registo bem sucedido, aparecer uma pequena notificação com essa informação, "Account created successfully". Isto acontece ainda na página de registo, antes de carregar o ecrã de *Login*.

Por fim, elaborou-se o tutorial, do género de documentação ou ajuda inicial. Serve este guia para descrever ou explicar determinadas informações pouco explícitas. A informação para este tutorial foi selecionada também com base na análise dos testes. Na figura 5.42 podemos observar como o tutorial é apresentado nas páginas, já o tutorial completo é apresentado no anexo K.

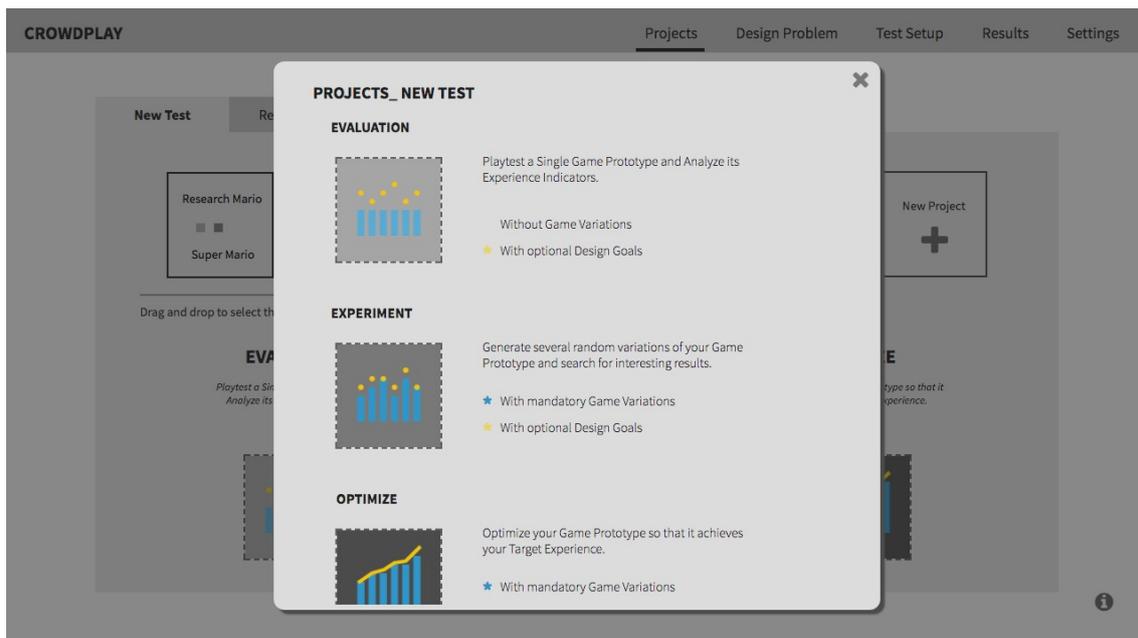


Figura 5.42 - Tutorial.

É possível aceder a esta ajuda a partir de todas as páginas do sistema e, consoante a página onde o utilizador se encontra, a informação é redirecionada (ou seja, a página faz *scroll* para a informação correspondente).

Descrevem-se em seguida as principais informações:

- Descrição dos diferentes tipos de testes;
- Informação de *New Test*;
- Informação de *Recent Tests*;
- Informação de *Design Problem*;

Considera-se que este tutorial seja útil para um novo utilizador compreender a plataforma na sua totalidade, permitindo uma aprendizagem rápida, e o acesso fácil a informações acerca do sistema (funcionamento base) ou a interface (como trabalhar com ela).

6

6.1 Conclusão

6.2 Trabalho futuro

Considerações Finais

6. Considerações Finais

Aqui apresentam-se as conclusões deste trabalho, quer que no que se refere ao trabalho teórico, quer ao trabalho prático, bem como às dificuldades sentidas. Neste último capítulo apontam-se também algumas sugestões de trabalho futuro relacionado com a área em estudo.

6.1 Conclusão

Durante a escrita desta dissertação abordámos temas importantes para a elaboração prática deste projeto; o Crowdplay que foi a base de trabalho e o design de interação a área prática de enquadramento e desenvolvimento. Foi importante conhecer e compreender o Crowdplay mais a fundo, pois não é um simples website, é uma ferramenta complexa.

No estudo da ferramenta foram encontradas algumas dificuldades visto não estar familiarizada com os jogos e o seu desenvolvimento, no entanto com algum investimento pessoal penso ter superado as dificuldades.

Quanto ao tema de design de interação, este foi decisivo para consolidar vários aspetos que contribuíram para um melhor desenvolvimento deste projeto. É fundamental quando desenvolvemos um projeto de design conhecer os fatores, regras e princípios do design de interação e da usabilidade, e ainda ter em conta a importância da avaliação e dos testes de usabilidade para melhorar o produto e a experiência do utilizador.

A avaliação preliminar feita à aplicação anterior tornou-se bastante pertinente, pois ajudou a compreender o funcionamento do Crowdplay, desde o seu uso aos requisitos, assim como a identificar determinados problemas. Os testes de usabilidade desenvolvidos nesta fase foram de igual modo importantes, pois consolidaram a listagem de todos os problemas. Nesta etapa surgiram ideias relevantes para o desenvolvimento da nova aplicação.

Na fase de desenvolvimento da proposta, foi feito um brainstorming para definir as novas funcionalidades do Crowdplay, após o que foram desenvolvidos os protótipos de baixa e alta fidelidade.

Nesta fase surgiram algumas dificuldades relativas à forma de testar, pois não era possível testar a aplicação nas suas características de interação fulcrais. A alteração da metodologia e plano de trabalho inicialmente traçados, revelou-se uma alternativa bem sucedida, sem prejuízo dos objetivos deste trabalho.

A utilização do *JustinMind* para desenvolver o protótipo funcional, revelou-se uma opção bastante útil e com um resultado final que cumpre os objetivos, funcionando na web. Foi desafiante aprender a trabalhar com um novo programa tão complexo, mas também compensador e gratificante, proporcionando novos conhecimentos e habilidades. Com dedicação conseguiu-se ultrapassar as dificuldades iniciais.

Desta dissertação era inicialmente esperado que o protótipo funcional fosse desenvolvido a partir das linguagens web: *html*, *css* e *javascript*, no entanto a alteração da metodologia (recurso ao *JustinMind*) levou a um produto final mais completo. Com recurso à programação não teria sido possível desenvolver algumas funcionalidades e interações, devido à complexidade e exigência da programação, ultrapassando o calendário disponível. É um facto que a interface não foi implementada com recurso à programação, por isso, o produto final resultante desta dissertação não é um protótipo funcional usável, também devido ao *software JustinMind* não gerar código reutilizável.

No entanto, a utilização do *JustinMind* permitiu uma maior dedicação aos aspetos de design, nomeadamente aos de design de interação, sem os problemas associados à programação. O protótipo final é completo, tem todas as interações e foi validado.

Em modo de conclusão pensa-se ter alcançado o objetivo planeado, desenhar uma nova interface para a plataforma Crowdplay, e oferecer uma melhor experiência ao utilizador, mantendo o conceito predefinido de uma interface com características de interação dos jogos e dos *softwares*.

Na interface desenhada estão presentes essas interatividades que traduzem a forma como o utilizador interage, desde um simples clique, duplo clique, função de arrastar (*drag and drop*) e ainda atalhos.

Essas características da interação foram implementadas de acordo com as funcionalidades, de modo a que a interatividade seja intuitiva. Tal como era de esperar, estas características podem dificultar o entendimento inicial, mas apesar disso, pensa-se ter conseguido elaborar uma interface intuitiva e fácil de usar. Podemos também concluir que é uma plataforma de fácil aprendizagem e memorização, e por isso mesmo, cumpre o objetivo.

Quanto à representação visual, conseguiu-se um design simples e minimalista, valorizando assim o conteúdo e reduzindo o *layout* ao essencial, tal como estava planeado.

Desde o início que este projeto se revelou bastante desafiante, em que as dificuldades foram vistas como oportunidades, pois com dedicação conseguiu-se ultrapassar todos os problemas, sendo que o resultado final do trabalho foi positivo e satisfatório. Reforça-se a ideia da importância do trabalho prático para a consolidação de conhecimentos e a obtenção de experiência nesta área do design. A nível pessoal constituiu um enorme desafio poder contribuir, não só para o desenvolvimento desta plataforma, como adquirir e consolidar conhecimentos nesta área do design e multimédia.

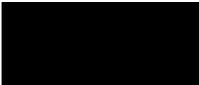
6.2 Trabalho Futuro

A pesquisa teórica e o trabalho prático desenvolvidos no decurso desta dissertação permitem argumentar que, apesar dos desenvolvimentos na área do design de jogos, há ainda espaço para trabalhos futuros neste domínio, nomeadamente na criação de ferramentas para testar e otimizar jogos. O objetivo planeado para esta dissertação foi alcançado, no entanto para a finalização do projeto Crowdplay é necessário proceder à sua implementação.

No entanto, há ainda algum trabalho prévio de design que deveria ser desenvolvido. Trata-se da heurística de prevenção de erros, ou seja, a interface deve prevenir a ocorrência de erros, apresentando ao utilizador uma opção de confirmação antes de se comprometer com a ação. Exemplo disto, seria uma caixa de diálogo para confirmar se desejaria apagar um determinado componente; estas referem-se apenas às ações ou erros do utilizador, e não inclui os erros da interface.

Outro aspeto que poderia ter sido alvo de mais estudos refere-se a paleta de cores. Apesar de se ter tido o cuidado de elaborar uma paleta de cores consistente, pensa-se poder haver espaço para mais estudos, reforçando e melhorando a ideia de associar uma cor a um componente para ajudar o utilizador a identificá-lo mais facilmente.

Apesar de tudo, consideramos que o trabalho desenvolvido nesta dissertação permite e apresenta todas as condições para a fase seguinte, a implementação em contexto real, e a sua posterior avaliação através de novos testes de usabilidade.



Glossário e Siglas

Glossário e Siglas

Abordagem Centrada no Autor de Geração de Conteúdo Procedimental | *Author-Centric Approach to Procedural Content Generation (ACPCG)*

Abordagem centrada no autor para a geração de conteúdo procedimental. Uma metodologia PCG que está mais focada na experiência de jogo consoante o objetivo de design do autor.

Authorial Game Evolution (AGE)

Ferramenta de “Authorial Game Evolution”, a abordagem do Crowdplay.

Indicador de Experiência | *Experience Indicator*

Uma medida sobre um aspeto da experiência do jogador.

Jogo Candidato | *Candidate game*

Uma versão do jogo protótipo com um conjunto de parâmetros gerados pelo algoritmo PCG.

Geração de Conteúdo Procedimental | *Procedural Content Generation (PCG)*

Processo de criação automática de conteúdo usando algoritmos.

Geração de Conteúdo Procedimental Mediada pela Experiência | *Experience-Driven Procedural Content Generation (EDPCG)*

Uma metodologia PCG focada no jogador que gera o conteúdo de forma a otimizar a sua experiência.

Métricas de jogo | *Gameplay Metrics*

Dados de jogo em bruto, não processados e coletados das sessões de jogo.

Objetivos de Design | *Design Goal*

Um objetivo a ser alcançado pelo designer num protótipo de jogo, definido por uma fórmula.

Problema de Design | *Design Problem*

Conjunto de variações de jogo e objetivos de design de um projeto. Representa o conjunto de testes aplicados.

Varição de Jogo | *Game Variation*

O conjunto de parâmetros pelos quais uma variável de jogo pode ser variada nos jogos candidatos.

Variável de Jogo | *Game Variable*

Um parâmetro num jogo a que será dado um valor em cada sessão.



Bibliografia

Bibliografia

Adobe Experience Design CC. Recuperado em maio de 2017 de <https://www.adobe.com/pt/products/experience-design.html>.

Brooke, J.(1996). SUS - A quick and dirty usability scale. In Jordan, P. W., Thomas, B., McClelland, I. L., Weerdmeester, B. (Eds.), *Usability Evaluation In Industry*, (pp. 189-194). London, UK: Taylor & Francis Ltd.

Cabral, A. (2016). *Visualising Gameplay Information for Game Design Insight* (Dissertação de Mestrado). Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

Carroll, J. M. (s.d.). 2. *Human Computer Interaction - brief intro*. In Interaction Design Foundation. Recuperado em outubro de 2016 de <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro>.

Cockton, G. (s.d.). 15. *Usability Evaluation*. In Interaction Design Foundation. Recuperado em novembro de 2016 de <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/usability-evaluation>.

Craveirinha, R., Barreto N., & Roque L. (2016). Towards a Taxonomy for the Clarification of PCG Actors' Roles. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '16, (pp. 244–253). Austin, Texas, USA: ACM.

Craveirinha, R., & Roque L. (2016). Exploring the Design-Space: The Authorial Game Evolution Tool Case-Study. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*, ACE 2016, (pp. 1-10). Osaka, Japan: ACM.

Craveirinha, R., Santos L., & Roque L. (2013). An Author-Centric Approach to Procedural Content Generation. In Reidsma, D., Katayose, H., Nijholt, A. (Eds.), *Advances in Computer Entertainment*. 10th International Conference, ACE 2013, (pp. 14-28). Boekelo, Netherlands: Springer International Publishing.

Crowdplay. Recuperado em setembro de 2016 de <http://crowdplay.dei.uc.pt/login>.

DeltaDNA. Recuperado em dezembro de 2016 de <https://deltadna.com/>.

DeltaDNA, Game Development Tools: Analytics Platform Comparison Table. Recuperado em janeiro de 2017 de <https://deltadna.com/blog/game-dev-tools-analytics-compared/>.

Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-Computer Interaction* (3ªEd.). Essex, England: Pearson Education Limited.

Dribbble (2013). *Minimalist is live!*. Recuperado em janeiro de 2017 de <https://dribbble.com/shots/960497-Minimalist-is-live>.

Dribbble (2013). *Ultramarine Admin*. Recuperado em janeiro de 2017 de <https://dribbble.com/shots/1023229-Ultramarine-Admin>.

Dribbble (2015). *Freebie: Course Dashboard (Sketch)*. Recuperado em janeiro de 2017 de <https://dribbble.com/shots/2050867-Freebie-Course-Dashboard-Sketch>.

Fonseca, M. J., Campos, P., & Gonçalves, D. (2012). *Introdução ao Design de Interfaces* (2ª Ed.). Lisboa, Portugal: FCA - Editora de Informática.

Font Awesome. Recuperado em abril de 2017 de <http://fontawesome.io/>.

GameAnalytics. Recuperado em novembro de 2016 de <http://www.gameanalytics.com/>.

Google Analytics. Recuperado em novembro de 2016 de <http://www.google.com/analytics/>.

Google Play Games Services. Recuperado em novembro de 2016 de <https://developers.google.com/games/services/android/stats>.

Hassenzahl, M. (s.d.). 3. *User Experience and Experience Design*. In Interaction Design Foundation. Recuperado em novembro de 2016 de <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/user-experience-and-experience-design>.

InVision. Recuperado em maio de 2017 de <https://www.invisionapp.com/>.

JustinMind. Recuperado em maio de 2017 de <https://www.justinmind.com/>.

Lowgren, J. (s.d.). *1. Interaction Design - brief intro*. In Interaction Design Foundation. Recuperado em outubro de 2016 de https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/interaction-design-brief-intro#toc_4.

Miller, L. (2016). *The Term Game: UX, UXD, CX, UI, IxD*. In Usabilla Blog. Recuperado em dezembro de 2016 de <http://blog.usabilla.com/the-term-game-ux-cx-ixd/>.

Mobyaffiliates, *What are the Best App Analytics Tools?*. Recuperado em janeiro de 2017 de <http://www.mobyaffiliates.com/blog/best-app-analytics-tools/#8>.

Nielsen, J. (1993). *Usability Engineering*. London, England: Academic Press. Recuperado em outubro de 2016 de <https://books.google.pt/books?id=DBOo-wF7LqIQC&printsec=frontcover&hl=pt-PT#v=onepage&q&f=false>.

Nielsen, J. (1995a). *10 Usability Heuristics for User Interface Design*. In Nielsen Norman Group. Recuperado em novembro de 2016 de <https://www.nngroup.com/articles/ten-usability-heuristics/>.

Nielsen, J. (1995b). *How to Conduct a Heuristic Evaluation*. In Nielsen Norman Group. Recuperado em novembro de 2016 de <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>.

Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. In Nielsen Norman Group. Recuperado em novembro de 2016 de <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.

Premium WP (2017). *Divi Builder – A Drag & Drop Page Builder Plugin for any WordPress Theme*. Recuperado em janeiro de 2017 de <http://www.premiumwp.com/divi-builder-drag-drop-page-builder-plugin-for-wordpress/>.

Saffer D. (2010). *Designing for Interaction - Creating Innovative Applications and Devices* (2ªEd.). Berkeley, USA: New Riders.

Shneiderman, B., & Plaisant C. (2005). *Designing the User Interface: Strategies for effective human-computer interaction* (4ªEd.). Maryland, USA: Pearson Education, Inc.

Sketch. Recuperado em maio de 2017 de <https://www.sketchapp.com/>.

Soares, J. (2015). *Crowdplay - Crowdsourcing Gameplay Data* (Dissertação de Mestrado). Faculdade de Ciências e Tecnologia, Universidade de Coimbra.

Soomla Blog, *The 2016 Definitive Guide to Mobile Analytic Platforms*. Recuperado em janeiro de 2017 de <http://blog.soom.la/2016/01/2016-guide-mobile-analytic-platforms.html>.

Swrve. Recuperado em dezembro de 2016 de <https://www.swrve.com/>.

Unity Analytics. Recuperado em dezembro de 2016 de <https://unity3d.com/services/analytics>.

Upsight. Recuperado em dezembro de 2016 de <http://www.upsight.com/>.



- A - Testes de Usabilidade (Documentação)
- B - Guião de Tarefas (1^{os} Testes de Usabilidade)
- C - Questionário
- D - Esboços e Protótipos em Papel
- E - Grelhas
- F - Screenshots do Software JustinMind
- G - Documento de Especificações
- H - Ecrãs do Protótipo Interativo
- I - Mapa do Website
- J - Guião de Tarefas (2^{os} Testes de Usabilidade)
- K - Tutorial

Anexos

A - Testes de Usabilidade (Documentação)

Nestes testes de usabilidade irá testar a interface de uma aplicação web, o Crowdplay.

Esta plataforma serve para testar e otimizar videojogos, com o objetivo de melhorar a experiência do jogador. Para isso é necessário definir as variações de jogo ("Game Variations") e os objetivos de design ("Design Goals").

O Crowdplay gera diferentes jogos candidatos a partir das variações de jogo, e, consoante os objetivos de design, tenta encontrar a melhor solução que corresponda a esse objetivo para a experiência.

As variações de jogo são parâmetros que podem variar nos diversos jogos candidatos; para isso definimos a variável que queremos sujeitar ao teste, o intervalo de valores, e a "probabilidade" para cada valor, entre outros parâmetros. Cada candidato irá então ter um valor específico, pertencente ao intervalo estipulado, para cada um dos parâmetros definidos.

Os objetivos de design são uma materialização dos objetivos de experiência do jogador que os designers pretendem alcançar. Define-se a pontuação (a forma como é pontuado cada candidato) em função do valor de cada indicador de experiência. Um indicador de experiência é um valor calculado a partir de fórmulas com base em diferentes fontes de dados, que serve para avaliar um determinado aspeto da experiência do jogador.

Imaginemos o exemplo *Super Mario*, no qual temos as "estrelas" para apanhar antes de concluir cada nível. Um objetivo de design hipotético poderia ser "o número ideal de estrelas" que pretendemos que os jogadores consigam apanhar. Para alcançar esse objetivo precisamos de definir um conjunto de testes.

Para permitir que o Crowdplay gere novos candidatos é necessário definir a forma como ele pode variar o jogo base. Para isso, o designer tem que definir as variações de jogo. No nosso caso hipotético, poder-se-ia definir uma variação para o número de estrelas presentes no nível a gerar, e logo, passíveis de serem colecionadas pelo jogador; por exemplo, esse intervalo poderia permitir valores entre 20 e 50.

Suponhamos que o ideal seria colecionar cerca de 30 estrelas durante o nível, então para isso define-se um teste que atribui uma pontuação a cada jogo consoante quantas estrelas o jogador coleciona (por exemplo, no intervalo entre 20 e 40), e quanto mais próximo de 30, maior a pontuação. O que isso vai fazer, é levar o motor de geração procedimental a tentar encontrar o jogo candidato que leve os jogadores a colecionar um número de estrelas tão próximo de 30 quanto possível.

No final terá uma página para observar os resultados dos indicadores de experiência dos jogos testados.

Tenha em consideração que estes testes servem para avaliar a interface e não a pessoa, pelo que se pede que responda a todas as perguntas com sinceridade. Para a divulgação dos resultados será respeitado o anonimato e a confidencialidade dos dados.

Na página seguinte encontra o guião com as tarefas, que peço para ler e realizar uma a uma (inclusive as sub-tarefas). Em cada tarefa será questionado sobre o grau de dificuldade e tem ainda a possibilidade de fazer algum comentário.

Após a realização das tarefas, existe um questionário global para a avaliação da aplicação.

Obrigada!

B - Guião de Tarefas (1^{os} Testes de Usabilidade)

TAREFAS

T1 - Registrar com a seguinte conta no projeto "Dune Nuno"

Login: anonimo__@dei.uc.pt

Nome: Teste __

Password: anonimo__

T1.1 - Entrar com a mesma conta no respectivo projeto ("Dune Nuno")

T2 - Desativar a variação "Soldier_Weapon Damage" nas variações de jogo ("Game Variation")

T3 - Ativar o teste "HasLost" nos objetivos de design ("Design Goals")

T4 - Adicionar uma variação de jogo no "Game Variation"

T4.1 - Escolher a variável "Radar_Power"

T4.2 - Selecionar o gráfico do tipo "Step"

T4.3 - Definir para variar entre 100 e 200

T4.4 - Com um incremento de 5 e guardar a variação de jogo

T5 - Adicionar um objetivo de design no "Design Goals"

T5.1 - Com o nome "QualityTeste"

T5.2 - Adicionar uma nova fórmula com o nome "N_eventos"

T5.3 - Selecionar a condição "Event" igual a "Unit Selected" e guardar fórmula

T5.4 - Selecionar essa fórmula e guardar o objetivo de design

T6 - Editar o objetivo de design com o nome "HasWon"

T6.1 - Definir a função de pontuação para o tipo "Triangular Step Right"

T6.2 - Definir a pontuação máxima para 200

T6.3 - Alterar a fórmula para calcular metade (" $\$HasWon/2$ ")

T6.4 - Definir que é dada pontuação máxima a partir 60 e guardar as alterações

T7 - Apagar o objetivo de design "QualityTeste"

T8 - Ir para a página do Test Setup e iniciar o teste

T9 - Pausar o teste e mudar o tipo de teste para "Sample"

T10 - Reiniciar o teste e Para-lo

T11 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

T11.1 - Escolher para o y "Count (Order unit to Attack)"

T11.2 - Escolher para todas as Sessões e visualizar gráfico

T12 - Visualizar dois gráficos, escolher o Histograma (Line Chart) e uma Tabela (Data Table)

T12.1 - Para o histograma escolher para o x "Session" e para o y "Indicator Value" e "Time per Session"

T12.2 - Para a tabela escolher para o x "Time(10s)" e com dois y, um "Unit selected" e outro "Moving Unit" (ambos para a sessão 889)

T12.3 - Visualizar os dois gráficos

C - Questionário

Questionário dos Testes de Usabilidade

Teste # _____

Nome _____

Formação _____

Idade _____

Ocupação _____

Está familiarizado com a plataforma online Crowdplay? Sim Não

Está familiarizado com desenvolvimento de jogos (quer seja programador ou designer)? Sim Não

Questões

discordo totalmente - concordo totalmente

1 2 3 4 5

Q1. Utilizaria o sistema com frequência (caso queira testar/otimizar algum jogo) 1 2 3 4 5

Q2. O sistema é demasiado complexo 1 2 3 4 5

Q3. O sistema é fácil de usar 1 2 3 4 5

Q4. Precisava de aprender algumas coisas antes de começar a usar este sistema 1 2 3 4 5

Q5. É necessário apoio de uma pessoa técnica para aprender a usar este sistema 1 2 3 4 5

Q6. A maioria das pessoas iria aprender a usar este sistema muito rapidamente 1 2 3 4 5

Q7. Senti dificuldade em perceber o significado de cada tarefa 1 2 3 4 5

Q8. Qual a tarefa mais difícil ou a maior dificuldade?

Q9. Outro comentário ou sugestão?

Obrigada!

D - Esboços e Protótipos em Papel

(digitalizações do caderno)

LOGIN

USER

PASS

LOGIN

ANDA NÃO TEM CONTA

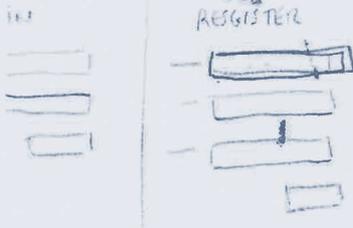
REGISTER

EMAIL (?)
 USER
 PASS
 CONFIRM PASS

REGISTER

JÁ TEM CONTA?

LOGIN



utilizadores
 registam-se ou o
 sistema é que
 regista ???.
 ou regista ?
 com projecto
 (associado)

Contas

- Admin
- Designer / Autor
- Jogador / Tester

- A conta define o tipo de utilizador.
 um utilizador tem uma conta para cada tipo de utilizador
 O utilizador (caso tenha mais do que um tipo) pode escolher como quer entrar depois do login (?)

→ EXPERIMENT

ESCOLHER PROJETO DEPOIS DO LOGIN

- Poder alterar de projeto sem ter que fazer Logout

??

Projecto jogo / Projecto teste

Select Project

Jogo X
Test A

Jogo Y
Test B

Jogo Z
Test C

Select Test Type

TEST

EXPERIMENT

OPTIMIZE

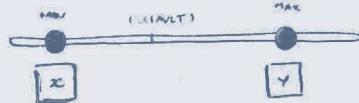
→ NONE ??
ou A VARIÁVEL

ON / OFF
???

APAGAR / ADICIONAR

GAME VARIATION DESIGN GOALS

PLAYER UNIT COSTS 5 \wedge 10			NONE VARIAT 5 \wedge 10 GAME VARIABLE
			+ ?



GAME VARIATION DESIGN GOALS

GAME VARIABLE

NAME
I

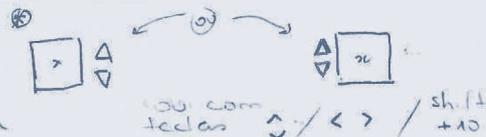
TYPE
o — o — o —

INCREMENT $x \updownarrow$

PROBABILITY FUNCTION

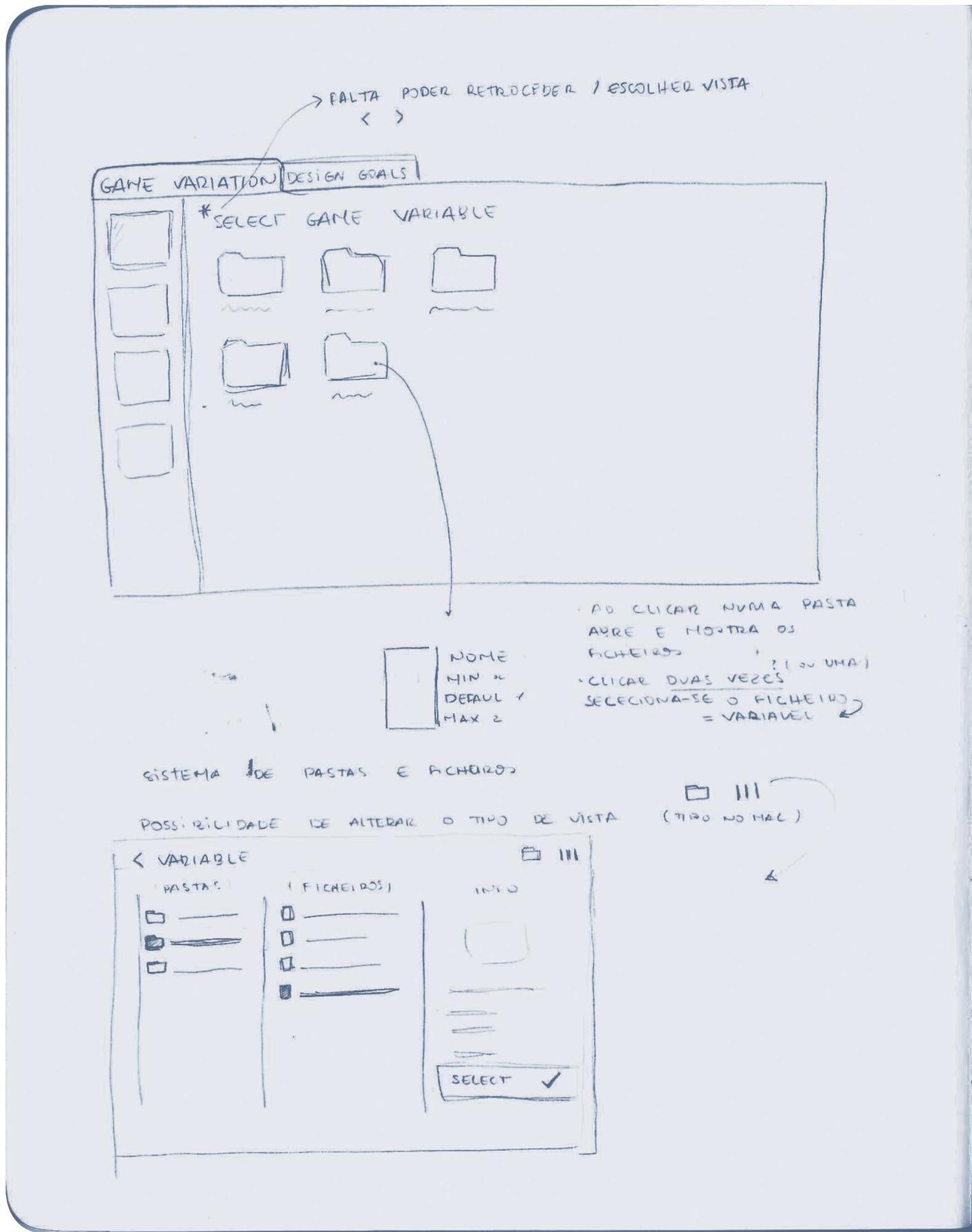
Score
PENALTY

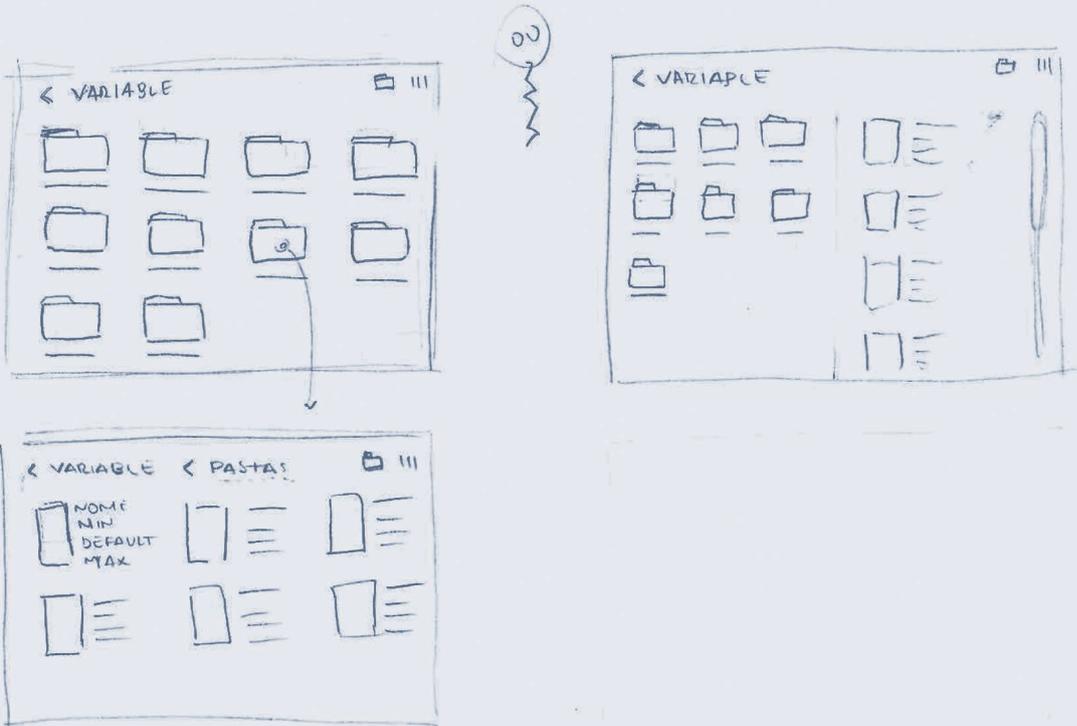
possibilidade de ver todos c/ scroll
pode alterar / editar



???

ou GUARDA AS ALTERAÇÕES QUANDO SE CARREGA AQUI (OU GUARDA SEMPRE QUE SE ALTERA





EDITAR / APAGAR / ADICIONAR



⊖ Ao abrir uma variável é logo para editar

? - Pode-se apagar esse variável?

GUARDAR / CANCELAR / APAGAR
(ou não)
GUARDAR / CANCELAR



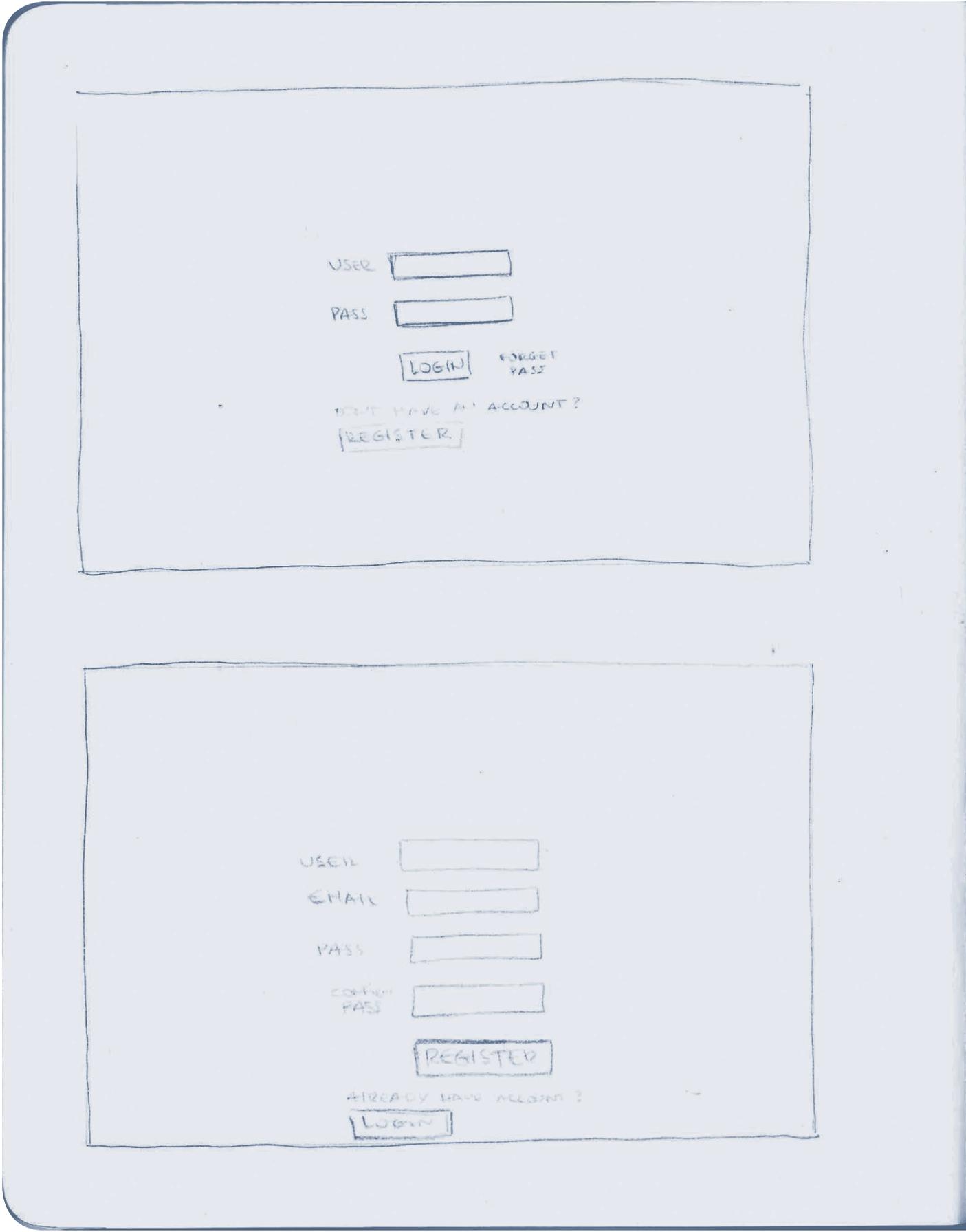
⊖ Ao abrir uma variável apenas para ver (sem editar)

- EDITAR / APAGAR

↙ Ao clicar em editar a info tornava-se editável

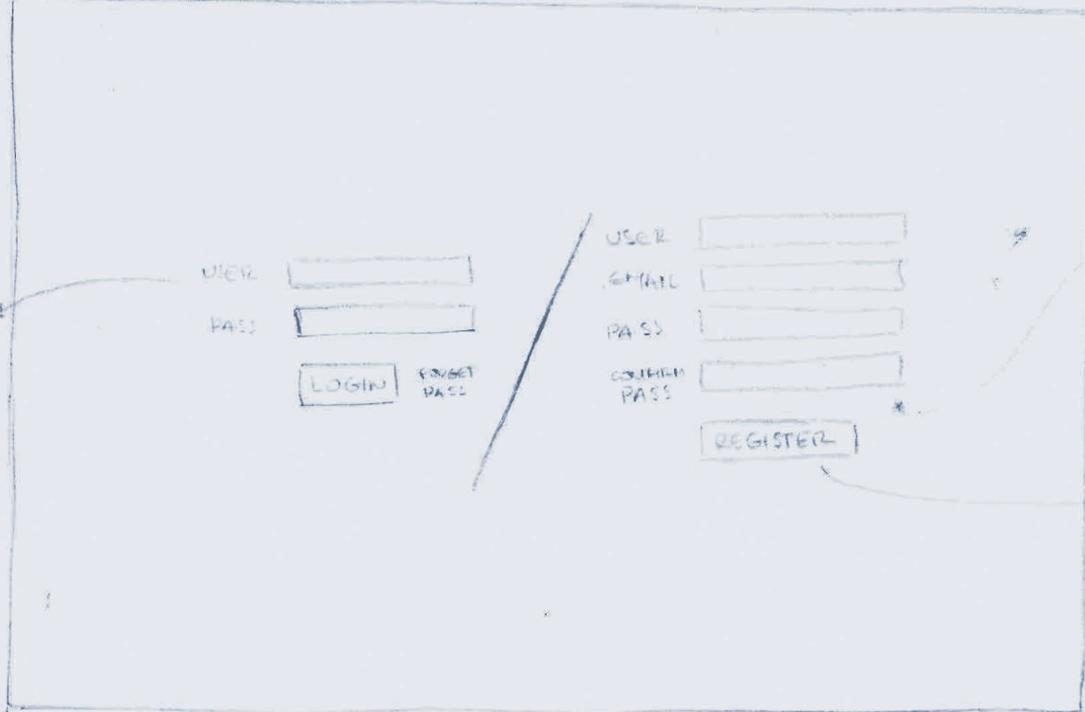
↘ Ao apagar volta para a vista das variáveis todas.

→ cliques desnecessários



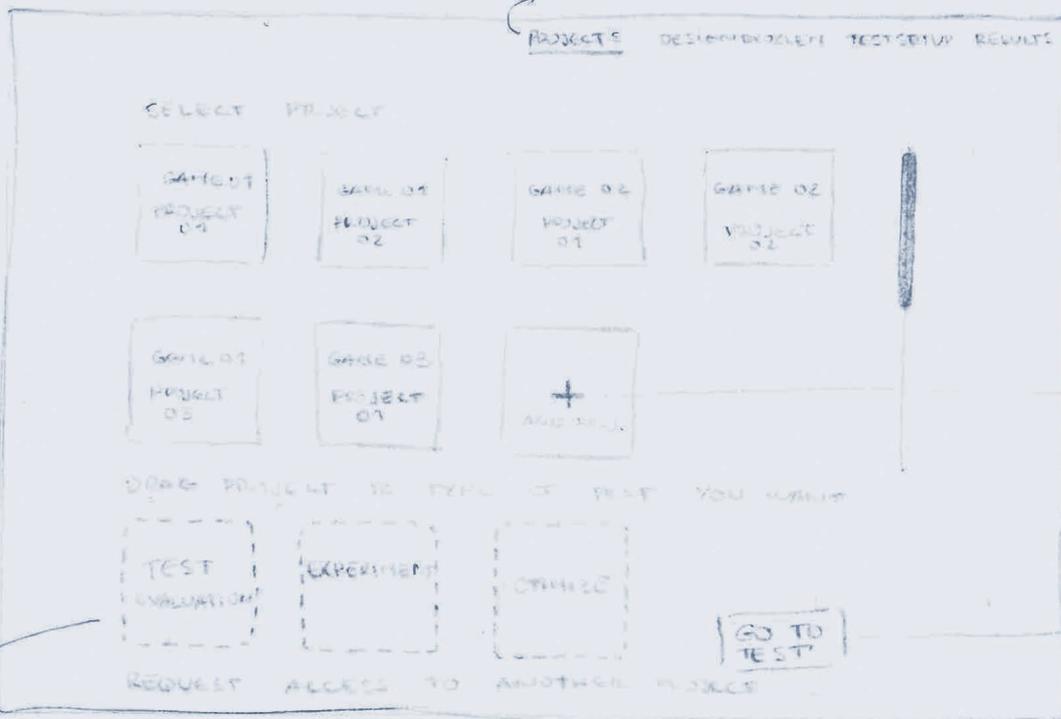
12

02
EMAIL



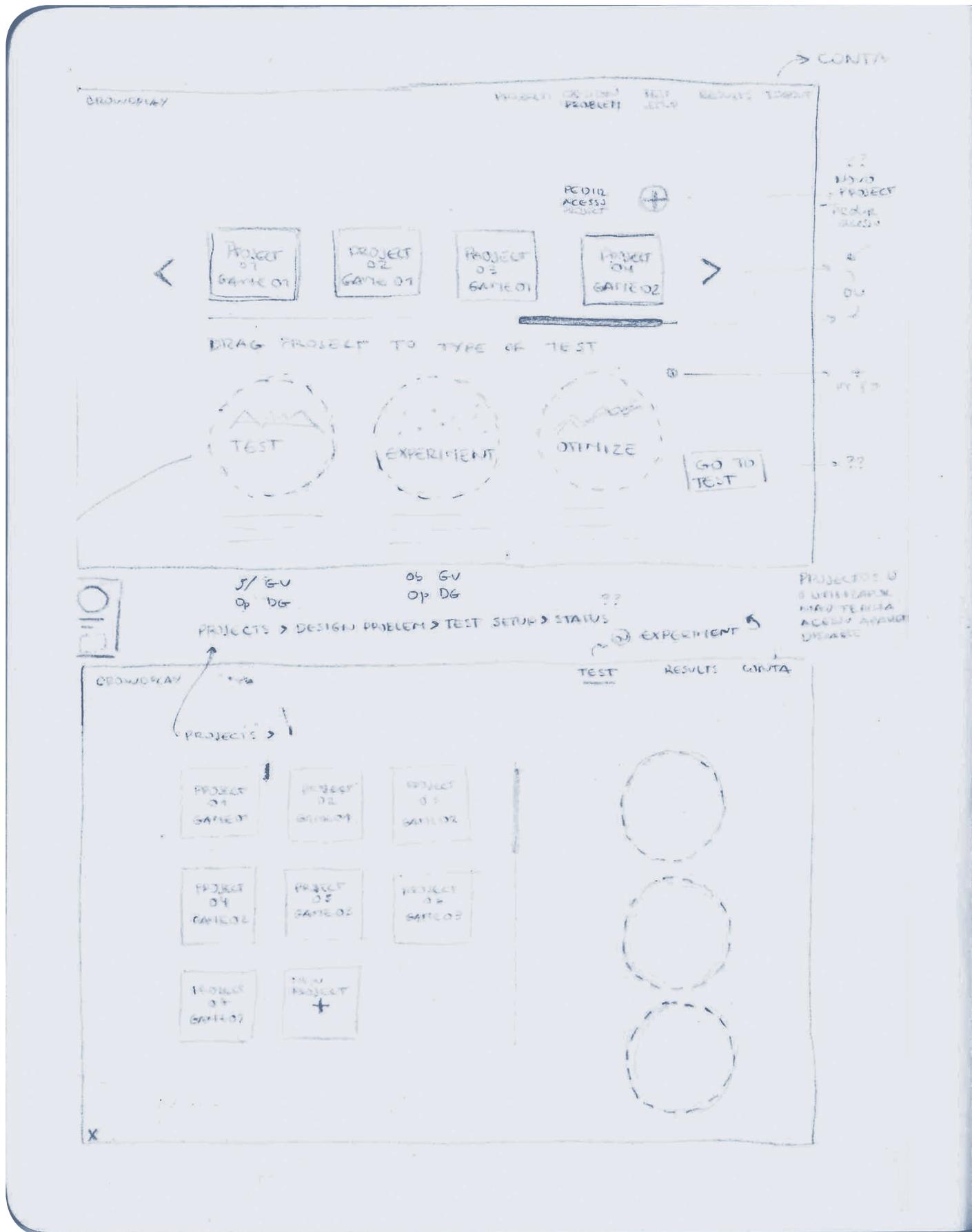
PROJECT

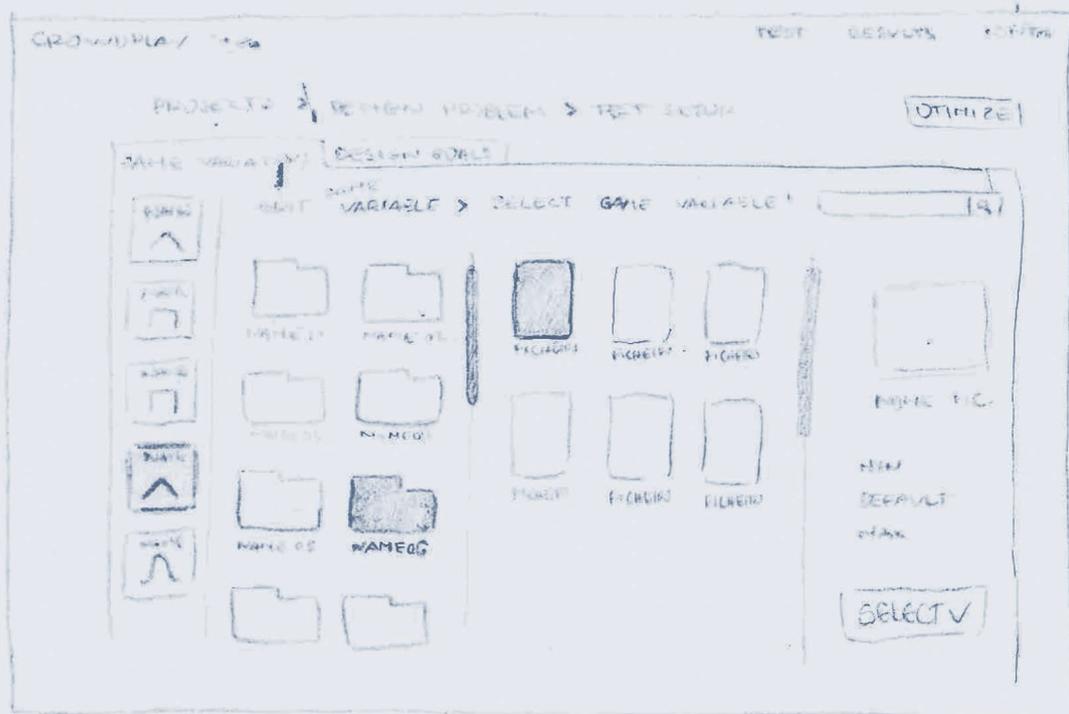
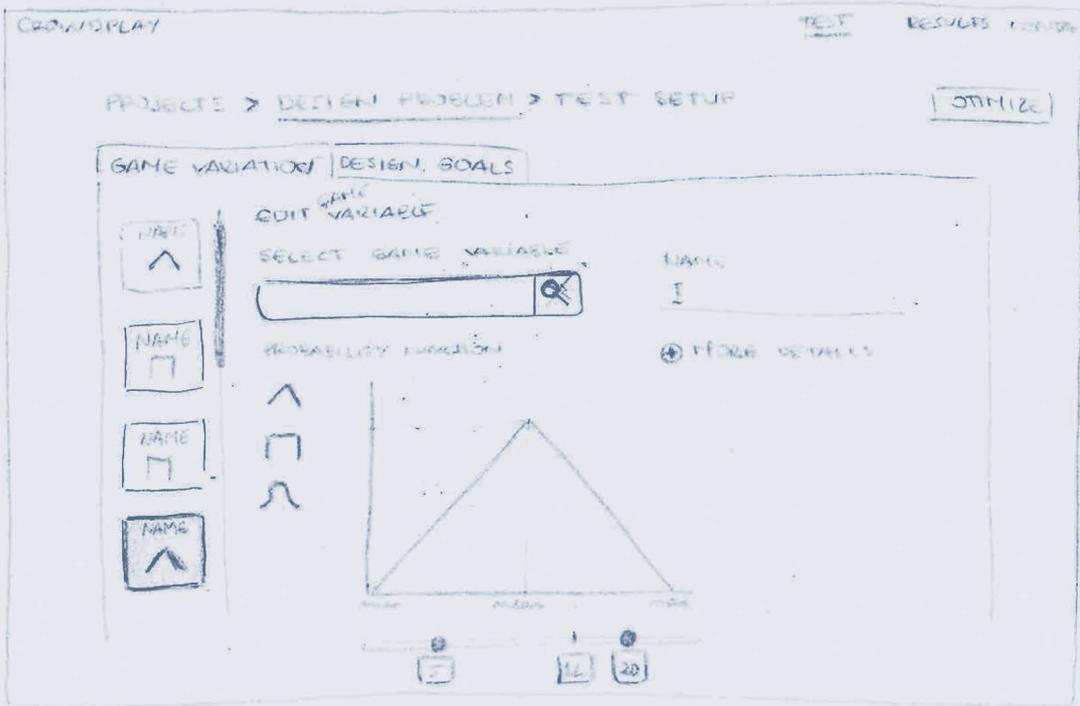
??? FOR AREA OF ADMIN - PROJECTS / USERS

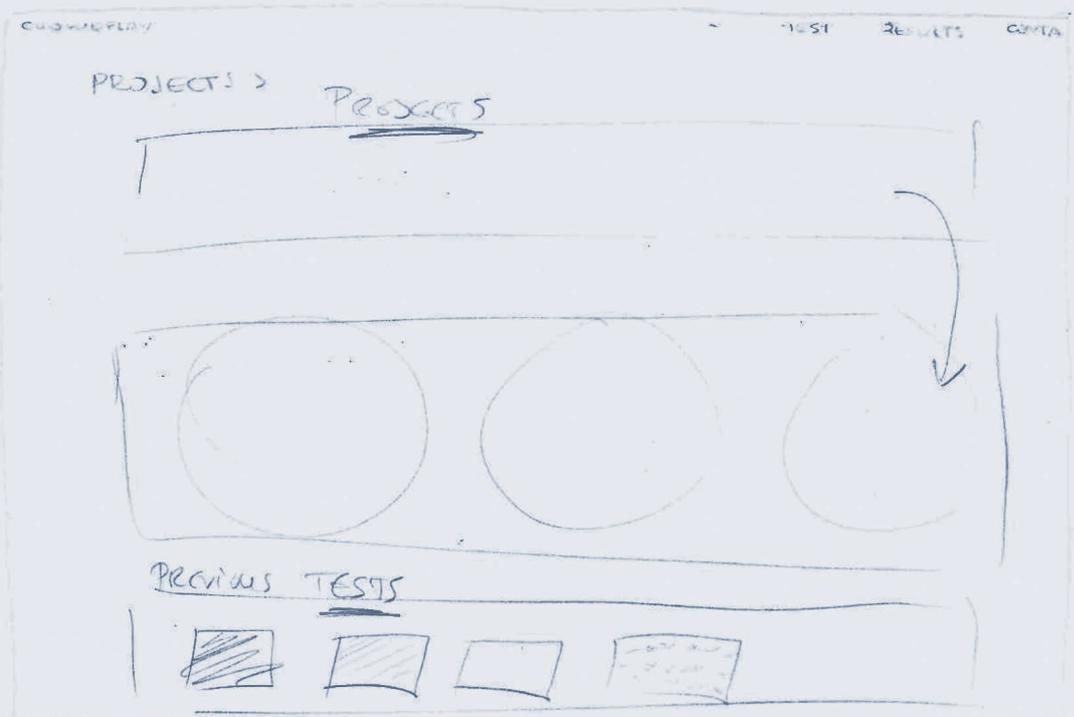
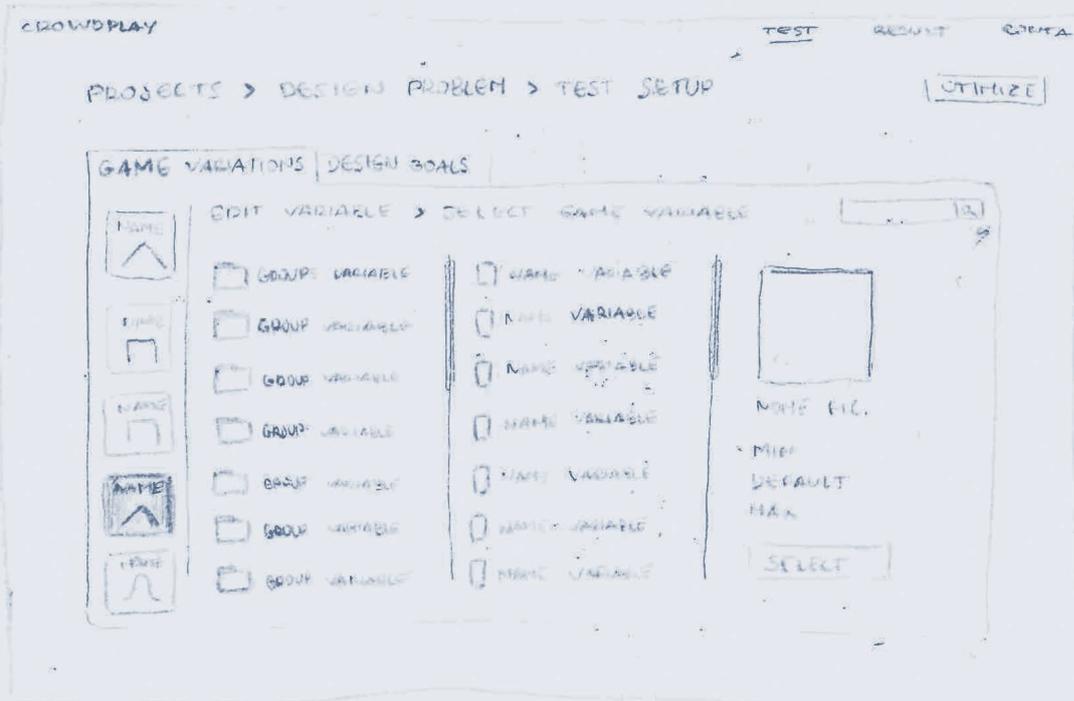


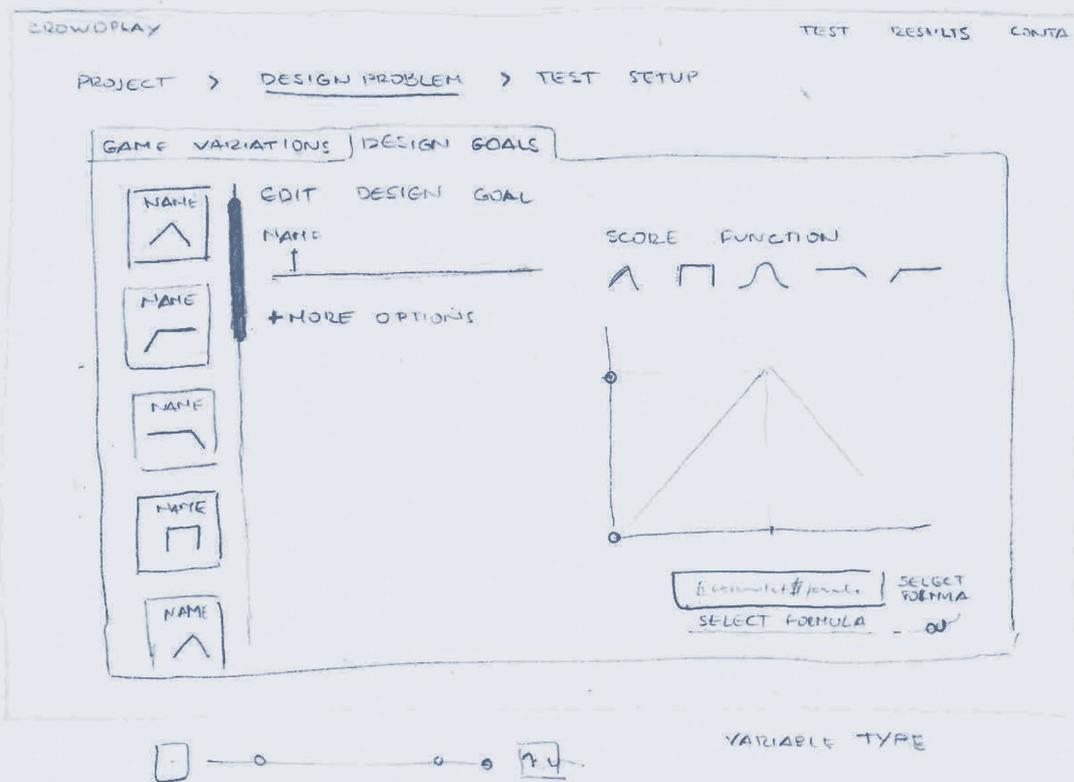
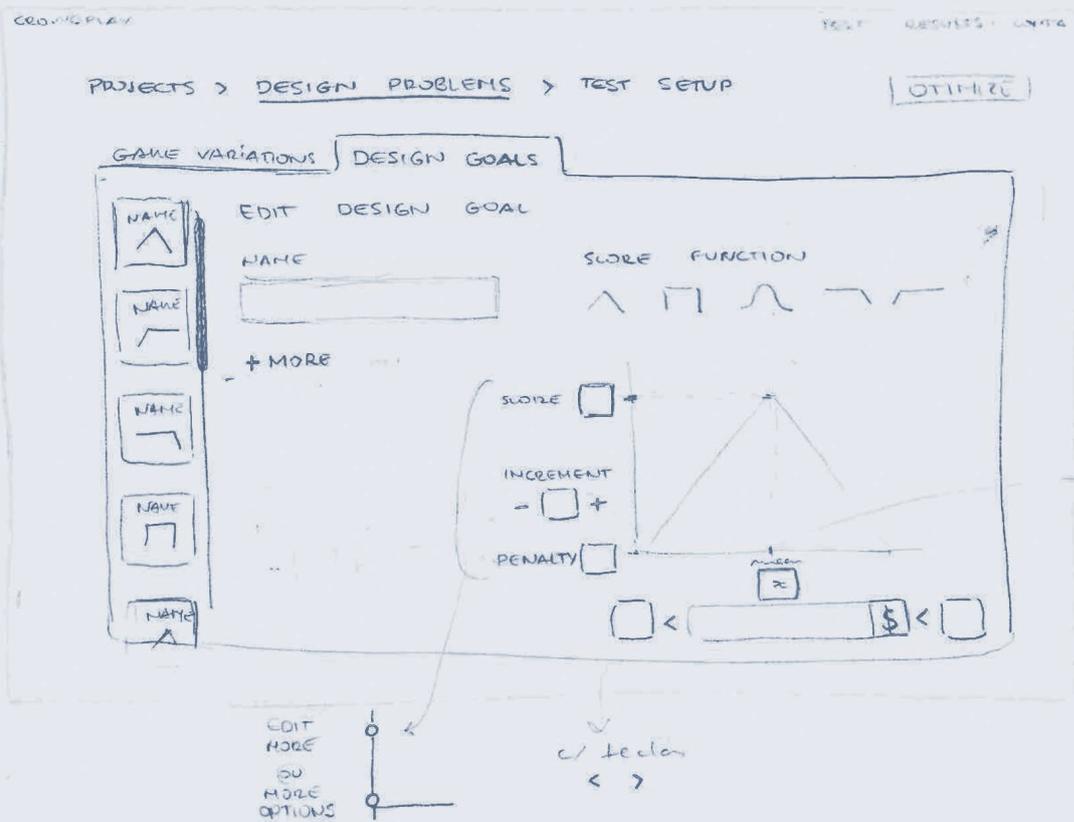
PROYECTOS

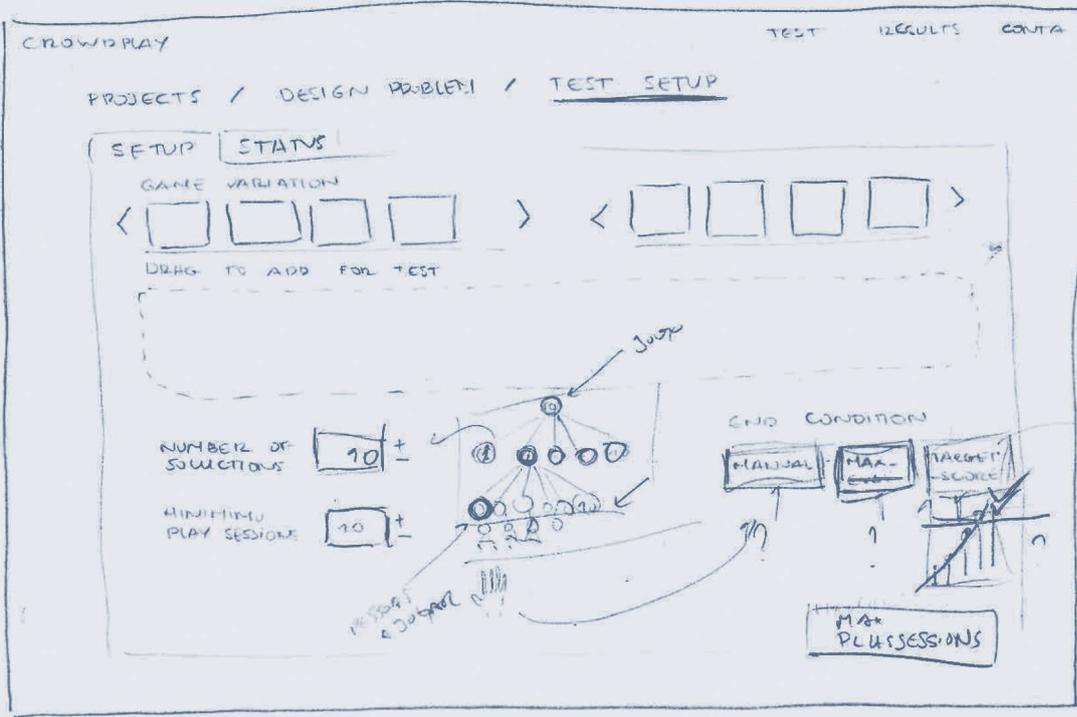
→ 02 → 3 tonalidades
 emails clave → emails de teste + integra





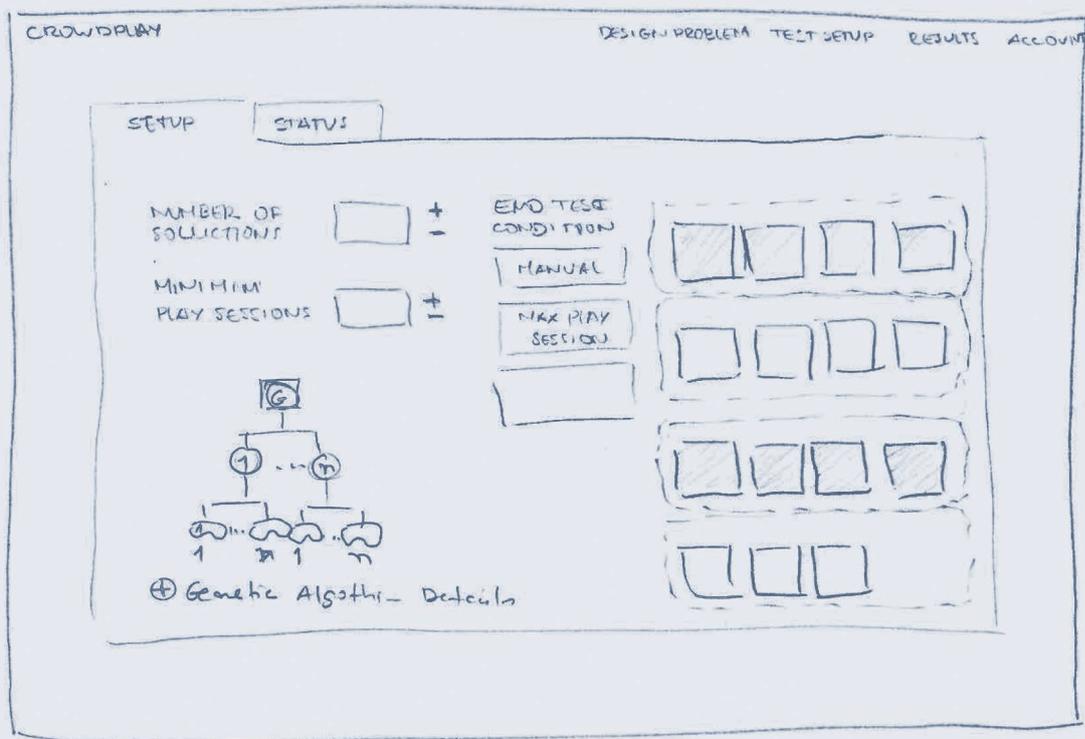




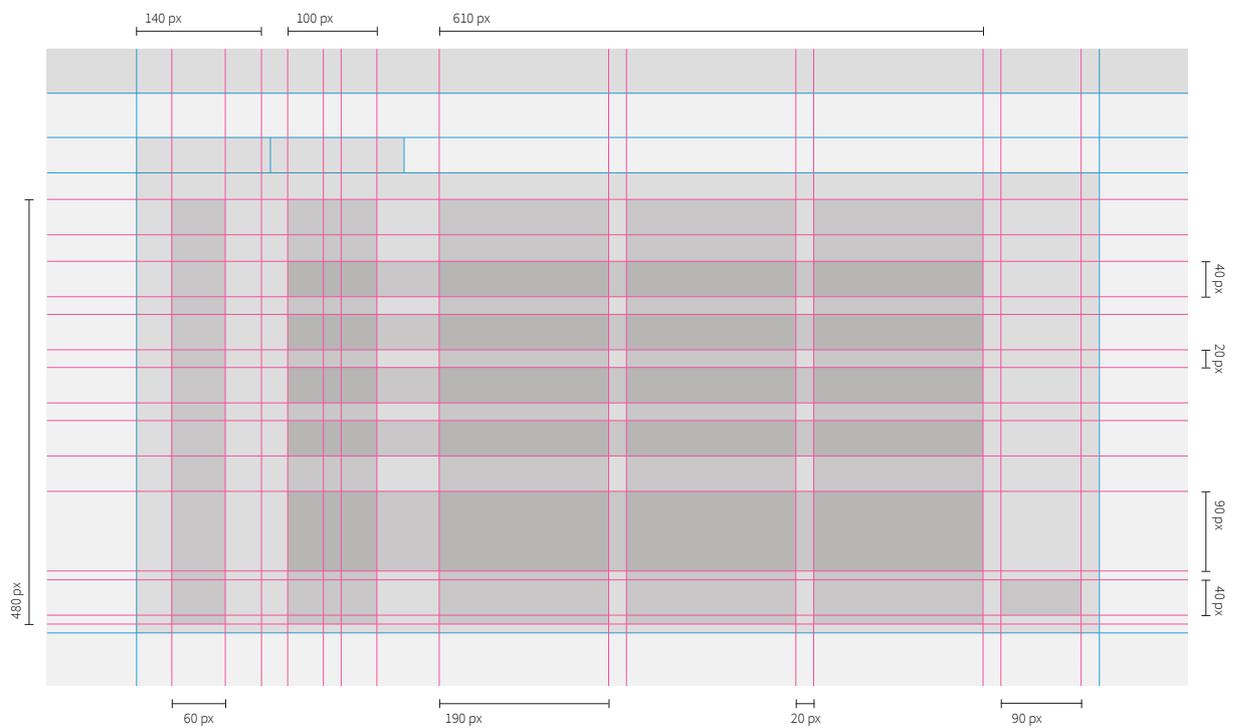
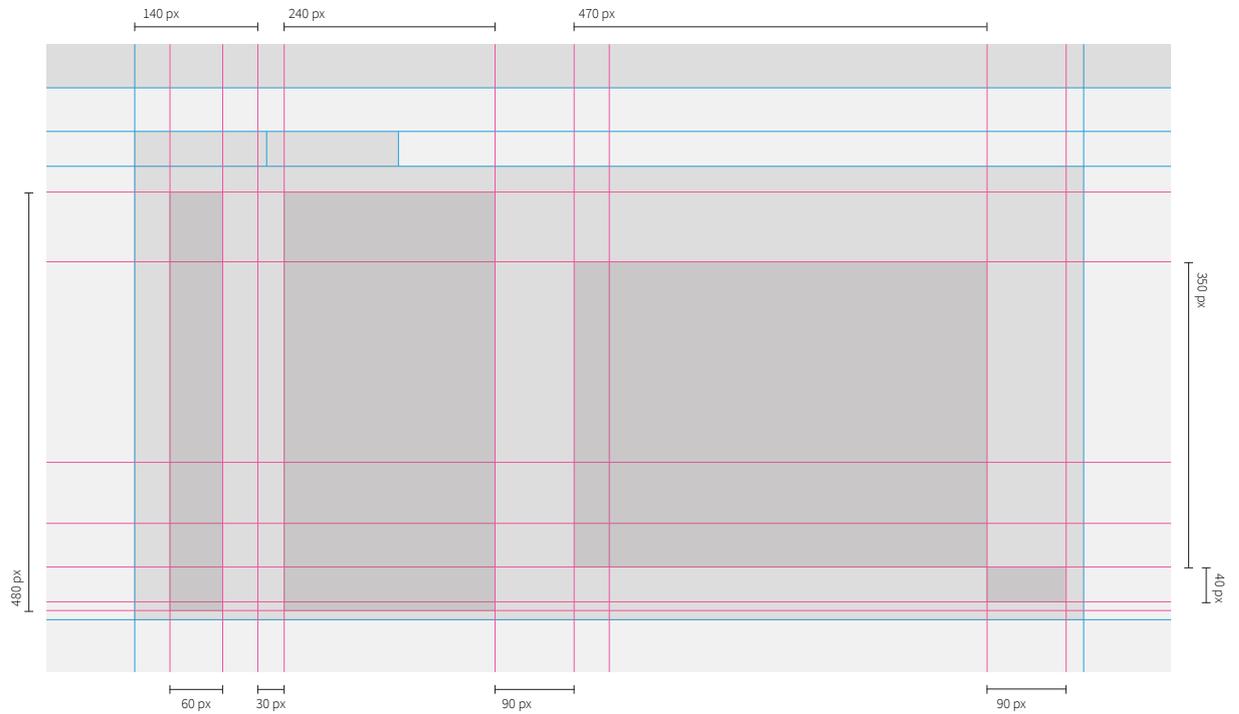


??
TYPE
Expense
eliminate
??

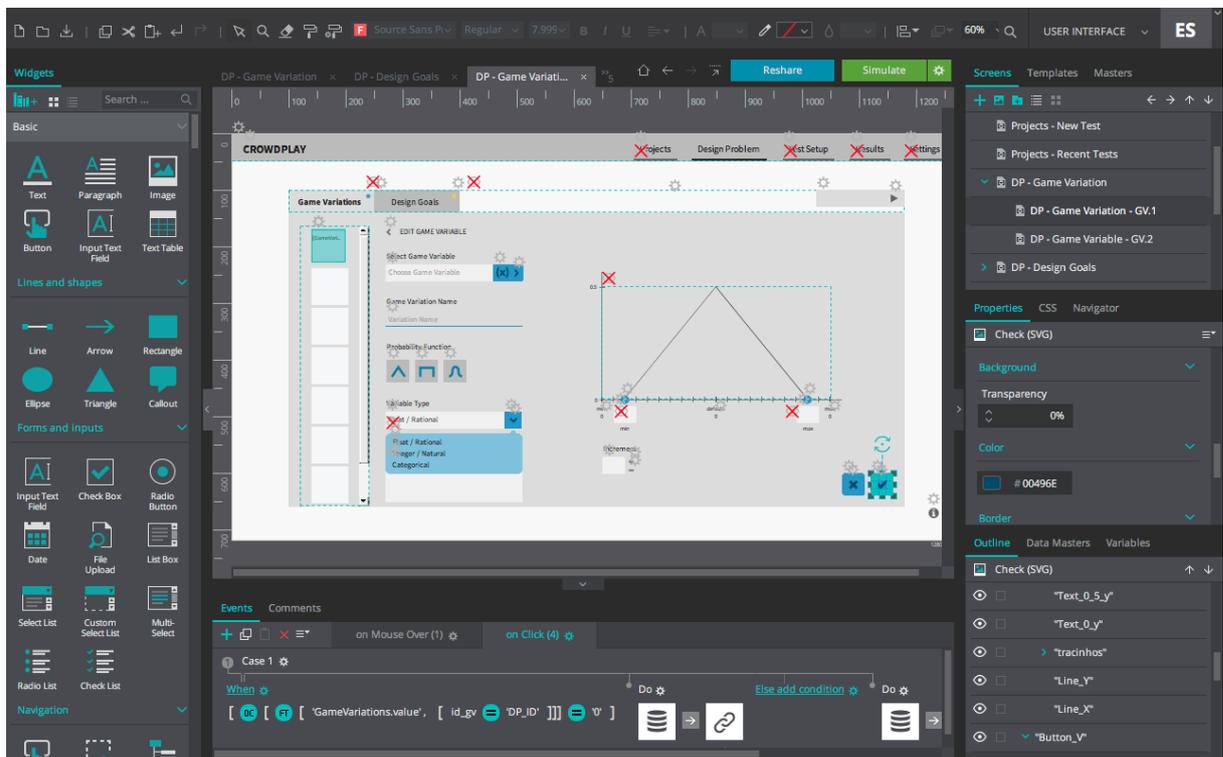
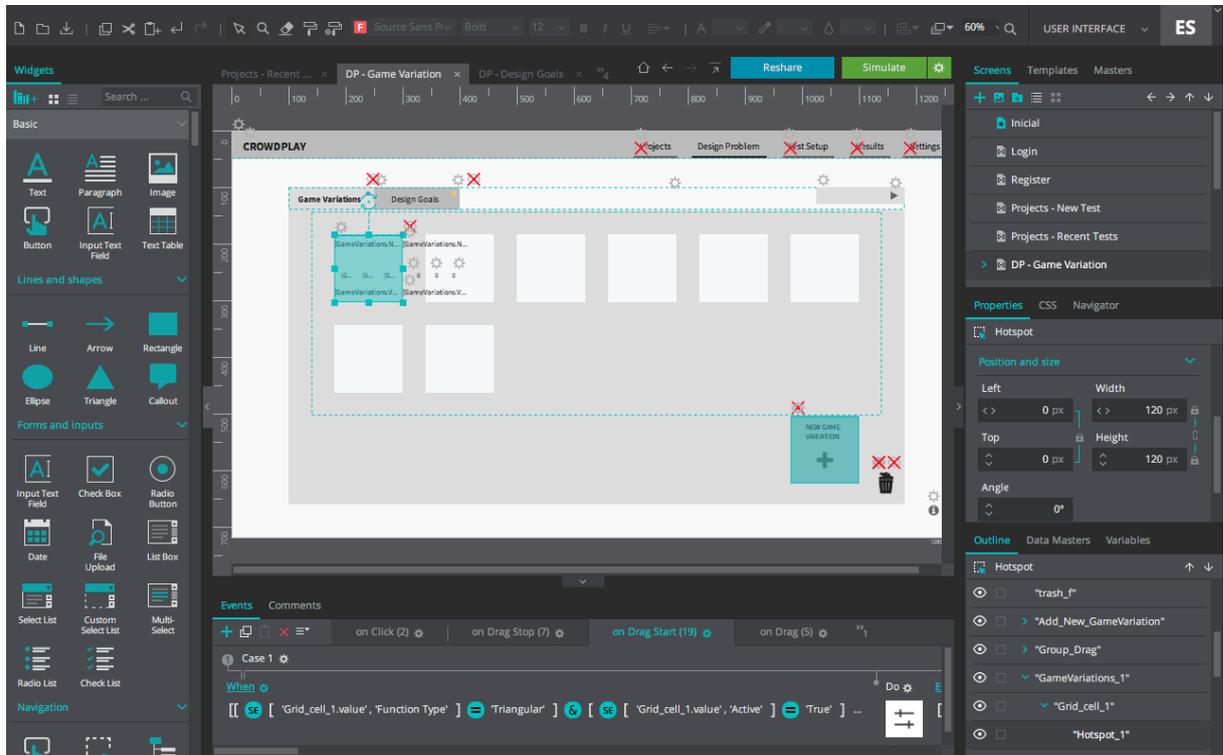
SOLUTIONS ●



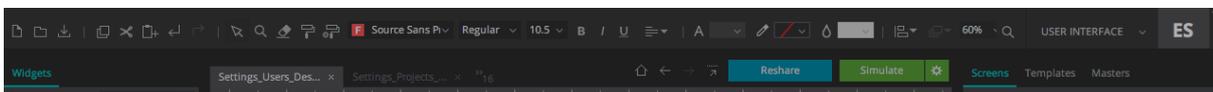
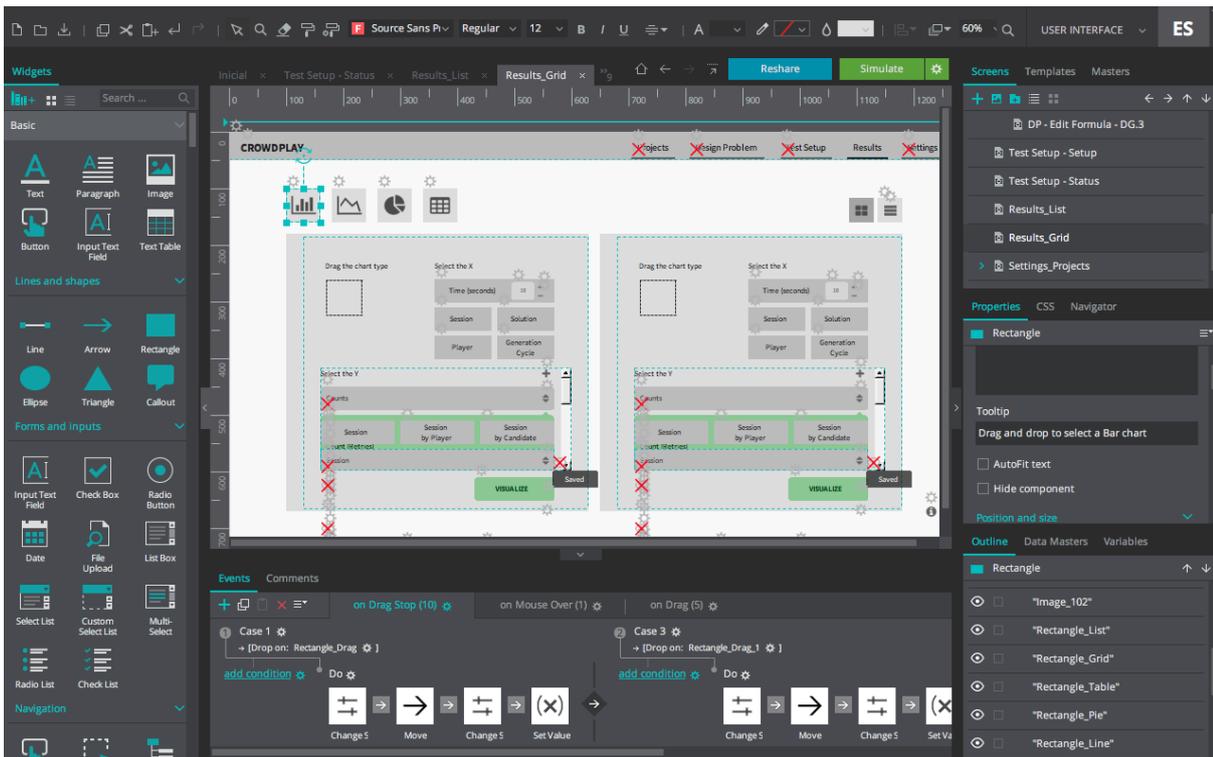
E - Grelhas



F - Screenshots do Software JustinMind



Re-Crowdplay



Data Master Details

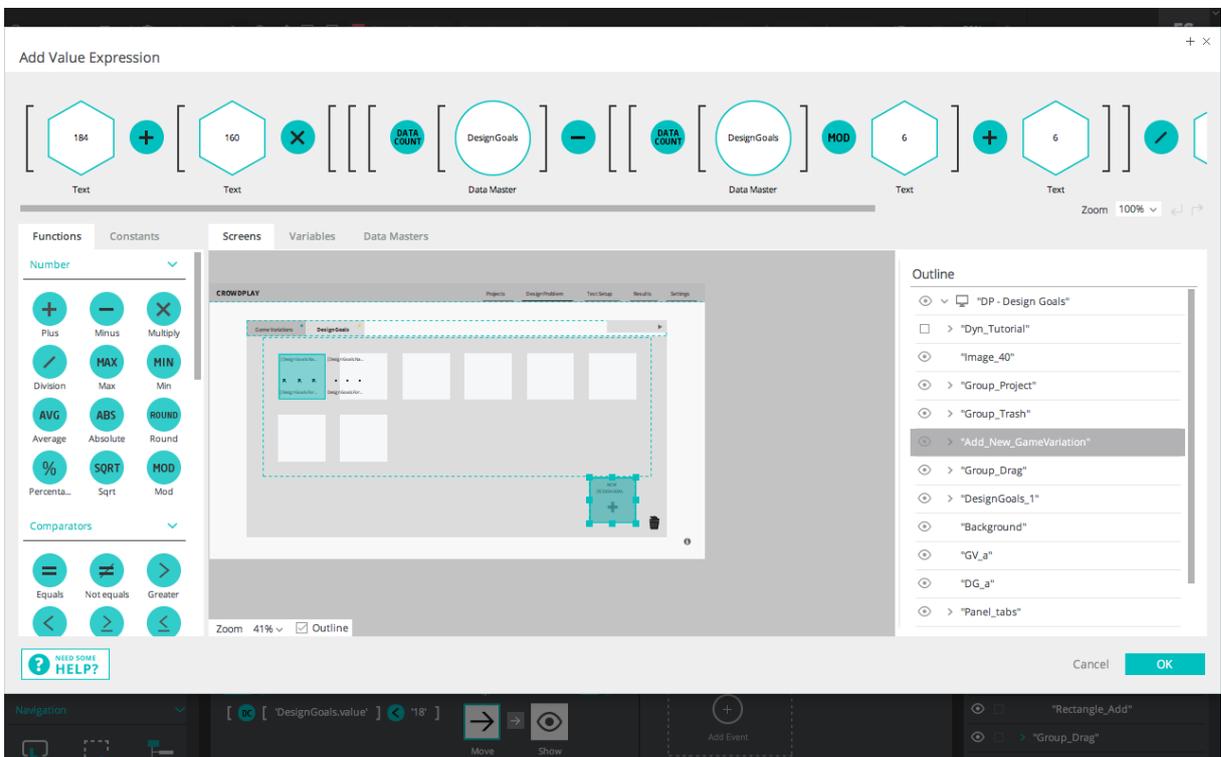
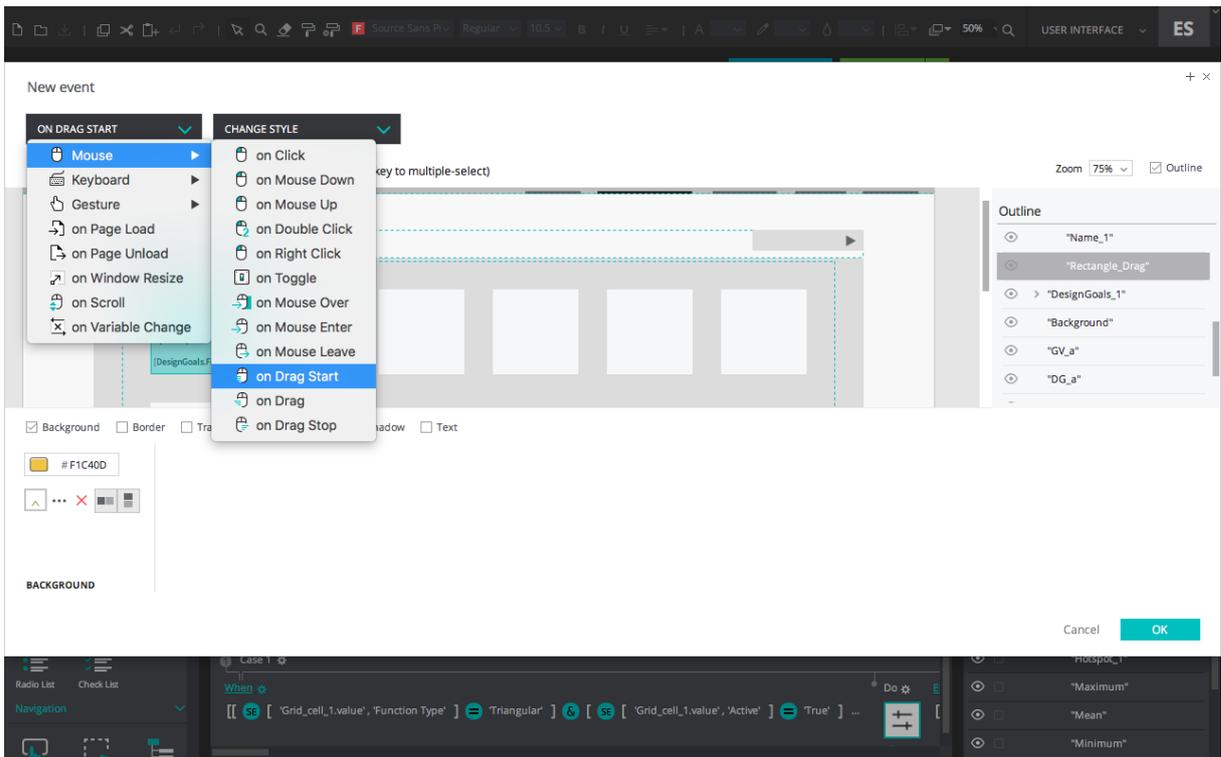
View and edit records

Import/Export Records

Records table

id	gv	Name	Variable	Function Type	Variable Type	Minimum	Mean	Maximum	Increment	Description	Active	Default Min	Default Mean	Default Max	
1		Speed	Engine Speed	Step	Float / Rational	1	30	100	1		True	0	50	100	↑
2		Height	Jump Height	Gaussian	Float / Rational	1	20	100	1		False	0	50	100	↓
3		Coins	Coins	Step	Float / Rational	20	200	400	10		True	0	250	500	✖
4		Power Ups	Number Power Ups	Step	Integer / Natural	5	10	20	1		False	0	20	40	✖
5		Enemies	Number of Enemies	Triangular	Categorical	5	15	30	1		True	0	25	50	
6		Pits	Number of Pits	Step	Float / Rational	10	20	40	2		False	0	25	50	
7		Stars	Number of Stars	Step	Float / Rational	2	5	8	1		True	0	5	10	
8		Lives	Number of Lives	Step	Float / Rational	1	3	5	1		True	0	5	10	

Cancel Ok



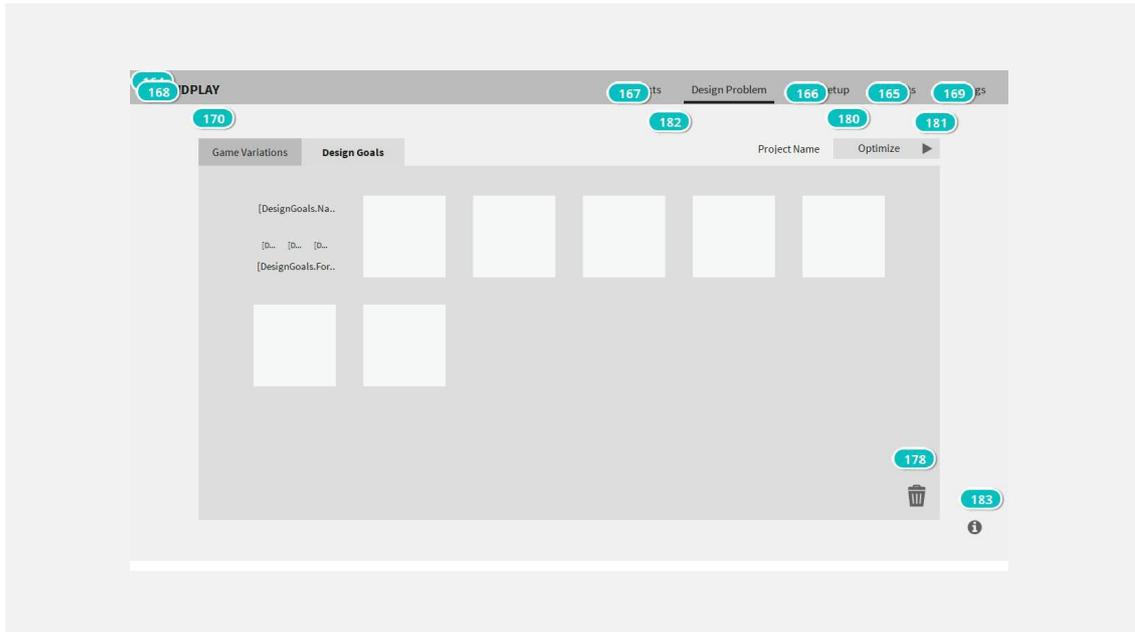
G - Documento de Especificações

(do JustinMind, pp. 22 a 34 de 120)

02. Screens / CrowdPlay_Interactive



DP - Design Goals



Interactions

- 164 **on Page Load:** moves 'Finned-top' to x: current value, y: 'windowScrollY' →
on Scroll: moves 'Finned-top' to x: current value, y: 'windowScrollY' →
- 165 **on Mouse Over:** shows 'Line_12' →
on Click: goes to 'Results_List' →
- 166 **on Mouse Over:** shows 'Line_13' →
on Click: goes to 'Test Setup - Setup' →
- 167 **on Mouse Over:** shows 'Line_15' →
on Click: goes to 'Projects - New Test' →
- 168 **on Click:** goes to 'Inical' →
- 169 **on Mouse Over:** shows 'Line_16' →
on Click: When (User_Type = 'Admin') goes to 'Settings_Projects' → Else goes to 'Settings_Users_Designer' →
- 170 **on Mouse Over:** When (Type_Test <> 'Evaluation') changes style of 'GameVariations': Background →
on Click: When (Type_Test <> 'Evaluation') goes to 'DP - Game Variation' →
- 178 **on Mouse Enter:** hides 'trash_f' → shows 'trash_f_h' →
- 180 **on Mouse Over:** changes style of 'Rectangle_3': Background → changes style of 'Image_130' →
on Click: goes to 'Test Setup - Setup' →
on Page Load: sets value 'Type_Test' to 'Rectangle_3' →
- 181 **on Mouse Over:** changes style of 'Image_130' → changes style of 'Rectangle_3': Background →
on Click: goes to 'Test Setup - Setup' →
- 182 **on Page Load:** sets value 'Name_Test_Project' to 'Text_2' →
- 183 **on Click:** shows 'Dyn_Tutorial' → scrolls to 'DESIGN_PROBLEM' →
on Mouse Over: changes style of 'Image_40' →

02. Screens / DP - Design Goals

Other states

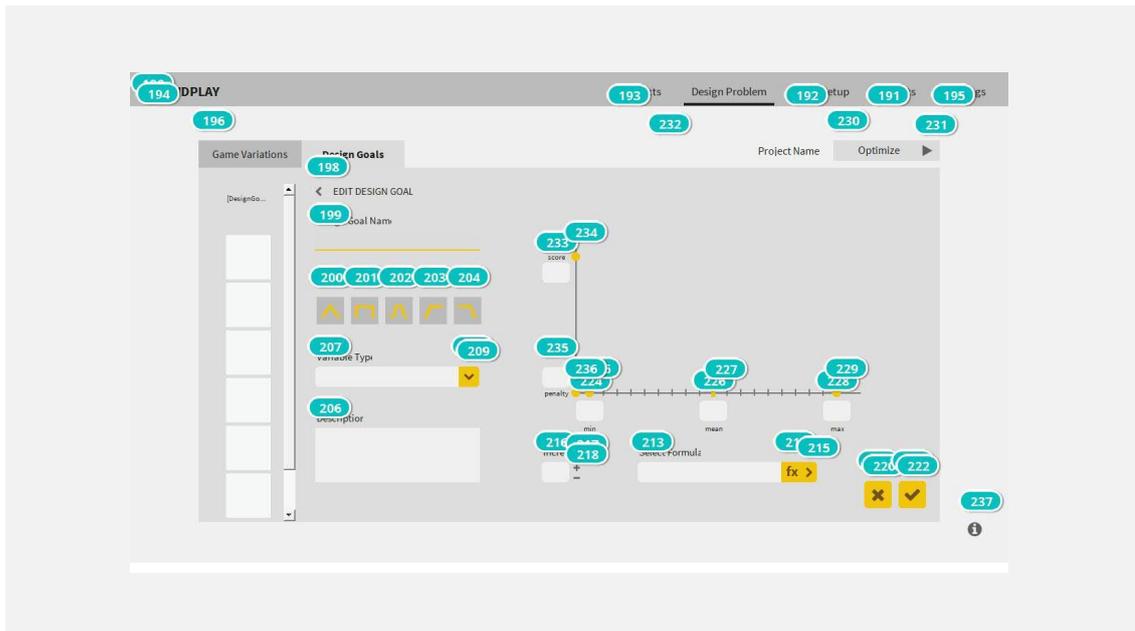


Interactions

- 184 **on Click:** hides 'Dyn_Tutorial' →
- 184 **on Mouse Over:** changes style of 'Image_120' →
- 185 **on Click:** hides 'Dyn_Tutorial' →
- 186 **on Mouse Leave:** changes style of 'Image_5': Border →
- 186 **on Mouse Enter:** changes style of 'Image_5': Border →
- 187 **on Mouse Leave:** changes style of 'Image_6': Border →
- 187 **on Mouse Enter:** changes style of 'Image_6': Border →
- 188 **on Mouse Leave:** changes style of 'Image_7': Border →
- 188 **on Mouse Enter:** changes style of 'Image_7': Border →
- 189 **on Mouse Enter:** moves 'New Project_Tut_6' to x: 116px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px → moves 'New Project_Tut_6' to x: 256px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px → moves 'New Project_Tut_6' to x: 396px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px →

02. Screens / CrowdPlay_Interactive

DP - Goal - DG.1



Interactions

- 190** **on Page Load:** moves 'Finned-top' to x: current value, y: 'windowScrollY' →
on Scroll: moves 'Finned-top' to x: current value, y: 'windowScrollY' →
- 191** **on Mouse Over:** shows 'Line_12' →
on Click: goes to 'Results_List' →
- 192** **on Mouse Over:** shows 'Line_13' →
on Click: goes to 'Test Setup - Setup' →
- 193** **on Mouse Over:** shows 'Line_15' →
on Click: goes to 'Projects - New Test' →
- 194** **on Click:** goes to 'Inical' →
- 195** **on Mouse Over:** shows 'Line_31' →
on Click: When (User_Type = 'Admin') goes to 'Settings_Projects' → Else goes to 'Settings_Users_Designer' →
- 196** **on Mouse Over:** When (Type_Test <> 'Evaluation') changes style of 'GameVariations': Background →
on Click: When (Type_Test <> 'Evaluation') goes to 'DP - Game Variation' →
- 198** **on Click:** goes to 'DP - Design Goals' →
on Mouse Over: changes style of 'Image_back' →
- 199** **on Page Load:** sets value 'DP_Name' to 'input_NameVariation' →
on Change: sets value 'input_NameVariation.value' to variable(s) 'DP_Name' →
- 200** **on Click:** show s'triang_on' → hides 'gaus_on, right_on, left_on, step_on' → sets value 'Triangular' to variable(s) 'DP_FunctionType' →
on Mouse Over: changes style of 'Rect_triang': Background →
- 201** **on Click:** show s'step_on' → hides 'gaus_on, right_on, left_on, triang_on' → sets value 'Step' to variable(s) 'DP_FunctionType' →
on Mouse Over: changes style of 'Rect_step': Background →
- 202** **on Click:** show s'gaus_on' → hides 'right_on, left_on, step_on, triang_on' → sets value 'Gaussian' to variable(s) 'DP_FunctionType' →
on Mouse Over: changes style of 'Rect_gaus': Background →
- 203** **on Click:** show s'right_on' → hides 'left_on, gaus_on, step_on, triang_on' → sets value 'Triangular Step Right' to variable(s) 'DP_FunctionType' →
on Mouse Over: changes style of 'Rect_right': Background →

- 204 **on Click:** shows 'left_on' → hides 'right_on, gaus_on, step_on, triang_on' → sets value 'Triangular Step Left' to variable(s) 'DP_FunctionType' →
on Mouse Over: changes style of 'Rect_left': Background →
- 206 **on Page Load:** sets value 'DP_Description' to 'input_Description' → sets value 'DP_Description' to 'input_Description' →
- 207 **on Mouse Over:** changes style of 'button_x_1': Background →
on Click: When not(VariableType_Drop.isvisible) shows 'VariableType_Drop' → Else hides 'VariableType_Drop' →
on Variable Change: sets value 'DP_VariableType' to 'input_GameVariable_1' →
on Page Load: sets value 'DP_VariableType' to 'input_GameVariable_1' →
- 208 **on Mouse Over:** changes style of 'button_x_1': Background →
on Click: When not(VariableType_Drop.isvisible) shows 'VariableType_Drop' → Else hides 'VariableType_Drop' →
- 209 **on Mouse Over:** changes style of 'button_x_1': Background →
on Click: When not(VariableType_Drop.isvisible) shows 'VariableType_Drop' → Else hides 'VariableType_Drop' →
- 213 **on Mouse Over:** changes style of 'button_x': Background →
on Click: goes to 'DP - Select Formula - DG.2' →
on Page Load: sets value 'DP_Formula_DG' to 'input_GameVariable' →
- 214 **on Mouse Over:** changes style of 'button_x': Background →
on Click: goes to 'DP - Select Formula - DG.2' →
- 215 **on Mouse Over:** changes style of 'button_x': Background →
on Click: goes to 'DP - Select Formula - DG.2' →
- 216 **on Page Load:** sets value 'DP_Increment' to 'input_inc_e' →
on Variable Change: sets value 'DP_Increment' to 'input_inc_e' →
on Change: sets value 'input_inc_e.value' to variable(s) 'DP_Increment' →
on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → sets value '(input_inc_e.value + DP_Increment_Inc)' to 'input_inc_e' → sets value '(input_inc_e.value - DP_Increment_Inc)' to 'input_inc_e' →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →
- 217 **on Click:** sets value '(input_inc_e.value + DP_Increment_Inc)' to 'input_inc_e' →
on Mouse Over: changes style of 'plus_inc' →
- 218 **on Click:** sets value '(input_inc_e.value - DP_Increment_Inc)' to variable(s) 'DP_Increment' →
on Mouse Over: changes style of 'minus_inc' →
- 219 **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Design Goals' →
- 220 **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Design Goals' →
- 221 **on Mouse Over:** changes style of 'Button_v': Border, Background →
on Click: When (count(filter(DesignGoals.value, (id_gv = DP_ID))) = '0') executes 'new(id_gv = (max(DesignGoals.value, id_gv) + '1'), Name = input_NameVariation.value, Formula = input_GameVariable.value, Function Type = DP_FunctionType, Variable Type = DP_VariableType, Minimum = input_min_e.value, Mean = input_mean_e.value, Maximum = input_max_e.value, Penalty = input_penalty_e.value, Score = input_score_e.value, Increment = input_inc_e.value, Description =

Increment = input_inc_e.value, Description = input_Description.value, Active = 'True') → goes to 'DP - Design Goals' → Else executes 'modify(filter(DesignGoals.value, (id_gv = DP_ID)), id_gv = undefined, Name = input_NameVariation.value, Formula = input_GameVariable.value, Function Type = DP_FunctionType, Variable Type = DP_VariableType, Minimum = input_min_e.value, Mean = input_mean_e.value, Maximum = input_max_e.value, Penalty = input_penalty_e.value, Score = input_score_e.value, Increment = input_inc_e.value, Description = input_Description.value, Active = undefined) → goes to 'DP - Design Goals' →

- 222 **on Click:** When (count(filter(DesignGoals.value, (id_gv = DP_ID))) = '0') executes 'new(id_gv = (max(DesignGoals.value, id_gv) + '1'), Name = input_NameVariation.value, Formula = input_GameVariable.value, Function Type = DP_FunctionType, Variable Type = DP_VariableType, Minimum = input_min_e.value, Mean = input_mean_e.value, Maximum = input_max_e.value, Penalty = input_penalty_e.value, Score = input_score_e.value, Increment = input_inc_e.value, Description = input_Description.value, Active = 'True') → goes to 'DP - Design Goals' → Else executes 'modify(filter(DesignGoals.value, (id_gv = DP_ID)), id_gv = undefined, Name = input_NameVariation.value, Formula = input_GameVariable.value, Function Type = DP_FunctionType, Variable Type = DP_VariableType, Minimum = input_min_e.value, Mean = input_mean_e.value, Maximum = input_max_e.value, Penalty = input_penalty_e.value, Score = input_score_e.value, Increment = input_inc_e.value, Description = input_Description.value, Active = undefined) → goes to 'DP - Design Goals' →
on Mouse Over: changes style of 'Button_v': Border, Background →
- 224 **on Page Load:** sets value 'DP_Min' to 'input_min_e' →
on Change: sets value 'input_min_e.value' to variable(s) 'DP_Min' →
on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → sets value '(input_min_e.value + DP_Increment_Inc)' to 'input_min_e' → moves 'Ellipse_min' to x: 18px, y: current value → moves 'Ellipse_min' to x: 13px, y: current value → sets value '(input_min_e.value - DP_Increment_Inc)' to 'input_min_e' → moves 'Ellipse_min' to x: 8px, y: current value → sets value '(input_min_e.value + DP_Increment_Inc)' to 'input_min_e' → moves 'Ellipse_min' to x: 18px, y: current value → moves 'Ellipse_min' to x: 13px, y: current value → sets value '(input_min_e.value - DP_Increment_Inc)' to 'input_min_e' → moves 'Ellipse_min' to x: 8px, y: current value → moves 'Ellipse_min' to x: 13px, y: current value →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →
- 225 **on Drag:** When (cursorX < Ellipse_min.x (absolute)) moves 'Ellipse_min' to x: current cursor 'x', y: current value → sets value '(input_min_e.value - DP_Increment_Inc)' to 'input_min_e' → When (cursorX > Ellipse_min.x (absolute)) moves 'Ellipse_min' to x: current cursor 'x', y: current value → sets value '(input_min_e.value + DP_Increment_Inc)' to 'input_min_e' →
on Mouse Over: changes style of 'Ellipse_min': Background →
on Drag Stop: moves 'Ellipse_min' to x: 13px, y: 0px →
- 226 **on Page Load:** sets value 'DP_Mean' to 'input_mean_e' →
on Change: sets value 'input_mean_e.value' to variable(s) 'DP_Mean' →
on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → When (input_mean_e.value < input_max_e.value) sets value '(input_mean_e.value + DP_Increment_Inc)' to 'input_mean_e' → moves 'Line_mean' to x: 19px, y: current value → moves 'Line_mean' to x: 14px, y: current value → When (input_mean_e.value > input_min_e.value) sets value '(input_mean_e.value - DP_Increment_Inc)' to 'input_mean_e' → moves 'Line_mean' to x: 9px, y: current value → moves 'Line_mean' to x: 14px, y: current value → When (input_mean_e.value < input_max_e.value) sets value '(input_mean_e.value + DP_Increment_Inc)' to 'input_mean_e' → moves 'Line_mean' to x: 19px, y: current value → moves 'Line_mean' to x: 14px, y: current value → When (input_mean_e.value > input_min_e.value) sets value '(input_mean_e.value - DP_Increment_Inc)' to 'input_mean_e' →

moves 'Line_mean' to x: 9px, y: current value → moves 'Line_mean' to x: 14px, y: current value →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →

227 on Drag: When ((cursorX < Line_mean.x (absolute)) and (input_mean_e.value > input_min_e.value)) moves 'Line_mean' to x: current cursor 'x', y: current value → sets value '(input_mean_e.value - DP_Increment_Inc)' to 'input_mean_e' → When ((cursorX > Line_mean.x (absolute)) and (input_mean_e.value < input_max_e.value)) moves 'Line_mean' to x: current cursor 'x', y: current value → sets value '(input_mean_e.value + DP_Increment_Inc)' to 'input_mean_e' →
on Mouse Over: changes style of 'Line_mean': Border →
on Drag Stop: moves 'Line_mean' to x: 13px, y: 3px →

228 on Page Load: sets value 'DP_Max' to 'input_max_e' →
on Change: sets value 'input_max_e.value' to variable(s) 'DP_Max' →
on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → sets value '(input_max_e.value + DP_Increment_Inc)' to 'input_max_e' → moves 'Ellipse_max' to x: 18px, y: current value → moves 'Ellipse_max' to x: 13px, y: current value → sets value '(input_max_e.value - DP_Increment_Inc)' to 'input_max_e' → moves 'Ellipse_max' to x: 8px, y: current value → moves 'Ellipse_max' to x: 13px, y: current value → sets value '(input_max_e.value + DP_Increment_Inc)' to 'input_max_e' → moves 'Ellipse_max' to x: 18px, y: current value → moves 'Ellipse_max' to x: 13px, y: current value → sets value '(input_max_e.value - DP_Increment_Inc)' to 'input_max_e' → moves 'Ellipse_max' to x: 8px, y: current value → moves 'Ellipse_max' to x: 13px, y: current value →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →

229 on Drag: When (cursorX < Ellipse_max.x (absolute)) moves 'Ellipse_max' to x: current cursor 'x', y: current value → sets value '(input_max_e.value - DP_Increment_Inc)' to 'input_max_e' → When (cursorX > Ellipse_max.x (absolute)) moves 'Ellipse_max' to x: current cursor 'x', y: current value → sets value '(input_max_e.value + DP_Increment_Inc)' to 'input_max_e' →
on Mouse Over: changes style of 'Ellipse_max': Background →
on Drag Stop: moves 'Ellipse_max' to x: 13px, y: current value →

230 on Mouse Over: changes style of 'Rectangle_5': Background → changes style of 'Image_130' →
on Click: goes to 'Test Setup - Setup' →
on Page Load: sets value 'Type_Test' to 'Rectangle_5' →

231 on Mouse Over: changes style of 'Image_130' → changes style of 'Rectangle_5': Background →
on Click: goes to 'Test Setup - Setup' →

232 on Page Load: sets value 'Name_Test_Project' to 'Text_6' →

233 on Page Load: sets value 'DP_Score_DG' to 'input_score_e' →
on Change: sets value 'input_score_e.value' to variable(s) 'DP_Score_DG' →

on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → sets value '(input_score_e.value + DP_Increment_Inc)' to 'input_score_e' → moves 'Ellipse_score' to x: current value, y: 8px → moves 'Ellipse_score' to x: current value, y: 13px → sets value '(input_score_e.value - DP_Increment_Inc)' to 'input_score_e' → moves 'Ellipse_score' to x: current value, y: 18px → moves 'Ellipse_score' to x: current value, y: 13px → sets value '(input_score_e.value + DP_Increment_Inc)' to 'input_score_e' → moves 'Ellipse_score' to x: current value, y: 8px → moves 'Ellipse_score' to x: current value, y: 13px → sets value '(input_score_e.value - DP_Increment_Inc)' to 'input_score_e' → moves 'Ellipse_score' to x: current value, y: 18px → moves 'Ellipse_score' to x: current value, y: 13px →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →

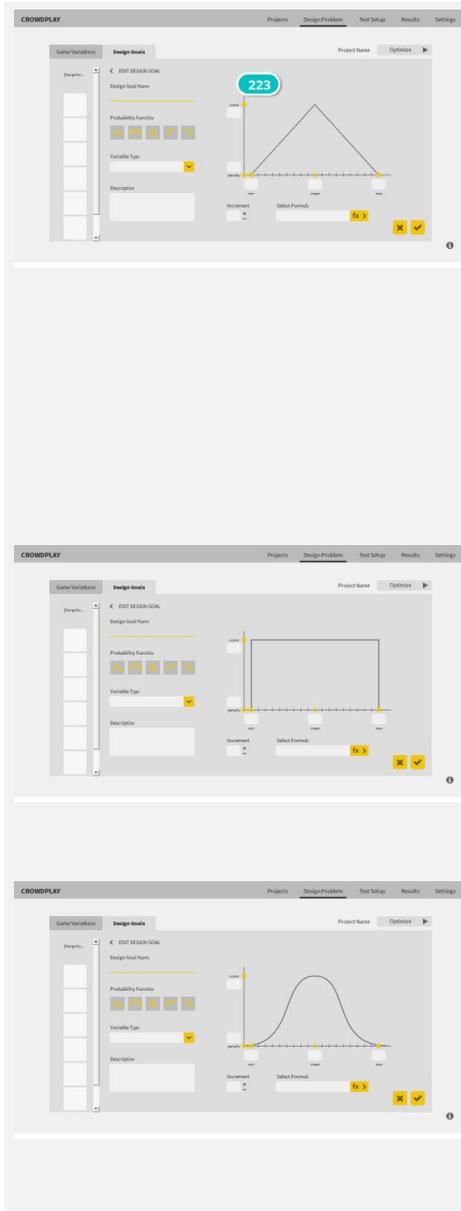
234 on Drag: moves 'Ellipse_score' to x: current value, y: current cursor 'y' → When (cursorY < Ellipse_score.y (absolute)) sets value '(input_score_e.value + DP_Increment_Inc)' to 'input_score_e' → When (cursorY > Ellipse_score.y (absolute)) sets value '(input_score_e.value - DP_Increment_Inc)' to 'input_score_e' →
on Mouse Over: changes style of 'Ellipse_score': Background →
on Drag Stop: moves 'Ellipse_score' to x: 0px, y: 13px →

235 on Page Load: sets value 'DP_Penalty_DG' to 'input_penalty_e' → sets value 'DP_Penalty_DG' to 'input_penalty_e' →
on Change: sets value 'input_penalty_e.value' to variable(s) 'DP_Penalty_DG' →
on Key Down: sets value '10' to variable(s) 'DP_Increment_Inc' → sets value '(input_penalty_e.value + DP_Increment_Inc)' to 'input_penalty_e' → moves 'Ellipse_penalty' to x: current value, y: 9px → moves 'Ellipse_penalty' to x: current value, y: 14px → sets value '(input_penalty_e.value - DP_Increment_Inc)' to 'input_penalty_e' → moves 'Ellipse_penalty' to x: current value, y: 19px → moves 'Ellipse_penalty' to x: current value, y: 14px → sets value '(input_penalty_e.value + DP_Increment_Inc)' to 'input_penalty_e' → moves 'Ellipse_penalty' to x: current value, y: 9px → moves 'Ellipse_penalty' to x: current value, y: 14px → sets value '(input_penalty_e.value - DP_Increment_Inc)' to 'input_penalty_e' → moves 'Ellipse_penalty' to x: current value, y: 19px → moves 'Ellipse_penalty' to x: current value, y: 14px →
on Key Up: sets value '1' to variable(s) 'DP_Increment_Inc' →

236 on Drag: moves 'Ellipse_penalty' to x: current value, y: current cursor 'y' → When (cursorY > Ellipse_penalty.y (absolute)) sets value '(input_penalty_e.value - DP_Increment_Inc)' to 'input_penalty_e' → When (cursorY < Ellipse_penalty.y (absolute)) sets value '(input_penalty_e.value + DP_Increment_Inc)' to 'input_penalty_e' →
on Mouse Over: changes style of 'Ellipse_penalty': Background →
on Drag Stop: moves 'Ellipse_penalty' to x: 0px, y: 14px →

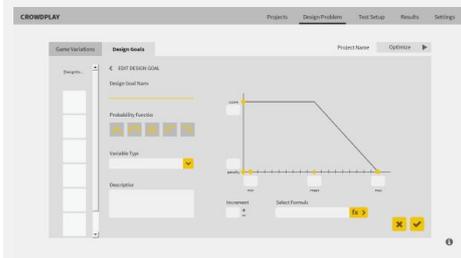
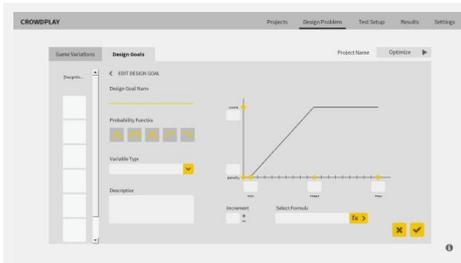
237 on Click: shows 'Dyn_Tutorial' → scrolls to 'DESIGN_PROBLEM' →
on Mouse Over: changes style of 'Image_40' →

Other states



Interactions

- 223 **on Variable Change:** When triang_on.isvisible sets active panel 'Panel_Triangular' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen step_on.isvisible sets active panel 'Panel_Step' → hides 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen gaus_on.isvisible sets active panel 'Panel_Guassian' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen right_on.isvisible sets active panel 'Panel_Right' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen left_on.isvisible sets active panel 'Panel_Left' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' →
- on Page Load:** When triang_on.isvisible sets active panel 'Panel_Triangular' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen step_on.isvisible sets active panel 'Panel_Step' → hides 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen gaus_on.isvisible sets active panel 'Panel_Guassian' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen right_on.isvisible sets active panel 'Panel_Right' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' → Else w hen left_on.isvisible sets active panel 'Panel_Left' → shows 'Group_Mean' → shows 'Dynamic_Type_Graph' →



Interactions

- 238 **on Click:** hides 'Dyn_Tutorial' →
- 239 **on Mouse Over:** changes style of 'Image_120' →
- 239 **on Click:** hides 'Dyn_Tutorial' →
- 240 **on Mouse Leave:** changes style of 'Image_5': Border →
- 240 **on Mouse Enter:** changes style of 'Image_5': Border →
- 241 **on Mouse Leave:** changes style of 'Image_6': Border →
- 241 **on Mouse Enter:** changes style of 'Image_6': Border →
- 242 **on Mouse Leave:** changes style of 'Image_7': Border →
- 242 **on Mouse Enter:** changes style of 'Image_7': Border →
- 243 **on Mouse Enter:** moves 'New Project_Tut_6' to x: 116px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px → moves 'New Project_Tut_6' to x: 256px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 396px, y: 1070px with easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px →

DP - Select Formula - DG.2

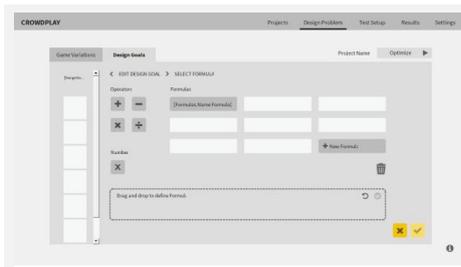


Interactions

- 244 **on Page Load:** moves 'Pinned-top' to x: current value, y: 'window ScrollY' →
on Scroll: moves 'Pinned-top' to x: current value, y: 'window ScrollY' →
- 245 **on Mouse Over:** shows 'Line_12' →
on Click: goes to 'Results_List' →
- 246 **on Mouse Over:** shows 'Line_13' →
on Click: goes to 'Test Setup - Setup' →
- 247 **on Mouse Over:** shows 'Line_15' →
on Click: goes to 'Projects - New Test' →
- 248 **on Click:** goes to 'Initial' →
- 249 **on Mouse Over:** shows 'Line_16' →
on Click: When (User_Type = 'Admin') goes to 'Settings_Projects' → Else goes to 'Settings_Users_Designer' →
- 250 **on Mouse Over:** When (Type_Test <> 'Evaluation') changes style of 'GameVariations': Background →
on Click: When (Type_Test <> 'Evaluation') goes to 'DP - Game Variation' →
- 252 **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Goal - DG.1' →
- 253 **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Goal - DG.1' →
- 254 **on Mouse Over:** When not((DP_Formula_DG = "")) changes style of 'Button_v': Border, Background →
on Click: When not((DP_Formula_DG = "")) goes to 'DP - Goal - DG.1' →
on Variable Change: When not((DP_Formula_DG = "")) changes style of 'Image_100' → changes style of 'Button_v': Border, Background → changes style of 'Image_103': Transparency → When (DP_Formula_DG = "") changes style of 'Image_100' → changes style of 'Button_v': Border, Background → changes style of 'Image_103': Transparency →
on Page Load: changes style of 'Image_100' → changes style of 'Button_v': Border, Background → changes style of 'Image_103': Transparency →
- 255 **on Click:** When not((DP_Formula_DG = "")) goes to 'DP - Goal - DG.1' →
on Mouse Over: When not((DP_Formula_DG = "")) changes style of 'Button_v': Border, Background →

- 256** **on Click:** goes to 'DP - Goal - DG.1' →
on Mouse Over: changes style of 'Text_Walk': Text →
- 257** **on Click:** goes to 'DP - Design Goals' →
on Mouse Over: changes style of 'Image_back' →
- 258** **on Mouse Over:** changes style of 'Rectangle_3': Background →
changes style of 'Image_130' →
on Click: goes to 'Test Setup - Setup' →
on Page Load: sets value 'Type_Test' to 'Rectangle_3' →
- 259** **on Mouse Over:** changes style of 'Image_130' → changes style of 'Rectangle_3': Background →
on Click: goes to 'Test Setup - Setup' →
- 260** **on Page Load:** sets value 'Name_Test_Project' to 'Text_2' →
- 261** **on Mouse Over:** changes style of 'Button_x_5': Border, Background → changes style of 'Image_90' → changes style of 'Button_tap': Background →
on Drag Stop: inserts 'Group_Number_Drag' into 'Panel_fxf' → changes style of 'Button_x_10': Border, Shadow, Background → changes style of 'Button_x_5': Border, Background → changes style of 'Button_tap': Background → changes style of 'Image_90': Transparency → hides 'Dashed_3' → hides 'Dashed_2' → hides 'Dashed_1' → When $\text{not}((\text{Group_Number_Drag.area} \text{ intersects } \text{Panel_fxf.area}))$ moves 'Group_Number_Drag' to x: 273px, y: 420px → hides 'Group_Number_Drag' → changes style of 'Button_x_5': Border, Background → changes style of 'Button_tap': Background → changes style of 'Image_90': Transparency →
on Drag: moves 'Group_Number_Drag' to x: current cursor 'x', y: current cursor 'y' → changes style of 'Image_90': Transparency → changes style of 'Button_x_5': Border, Background → changes style of 'Button_tap': Background → show s'Group_Number_Drag' → changes style of 'Button_x_10': Border, Shadow, Background →
- 262** **on Mouse Over:** changes style of 'Button_x_4': Border, Background → changes style of 'Image_96, Image_163' →
on Drag Stop: inserts 'Button_Division_Drag' into 'Panel_fxf' → changes style of 'Button_x_6': Border, Shadow, Background → changes style of 'Button_x_4': Transparency → changes style of 'Image_96, Image_163': Transparency → hides 'Dashed_3' → hides 'Dashed_2' → hides 'Dashed_1' → When $\text{not}((\text{Button_Division_Drag.area} \text{ intersects } \text{Panel_fxf.area}))$ moves 'Button_Division_Drag' to x: 333px, y: 300px → hides 'Button_Division_Drag' → changes style of 'Button_x_4': Transparency → changes style of 'Image_96, Image_163': Transparency →
on Drag: moves 'Button_Division_Drag' to x: current cursor 'x', y: current cursor 'y' → changes style of 'Image_96, Button_x_4, Image_163': Transparency → shows 'Button_Division_Drag' → changes style of 'Button_x_1, Button_x_3, Button_x_2': Border, Background → changes style of 'Image_95, Image_135, Image_119' → changes style of 'Button_x_6': Border, Shadow, Background →
- 263** **on Mouse Over:** changes style of 'Button_x_3': Border, Background → changes style of 'Image_119' →
on Drag Stop: inserts 'Button_Multiply_Drag' into 'Panel_fxf' → changes style of 'Button_x_7': Border, Shadow, Background → changes style of 'Button_x_3': Transparency → changes style of 'Image_119': Transparency → hides 'Dashed_3' → hides 'Dashed_2' → hides 'Dashed_1' → When $\text{not}((\text{Button_Multiply_Drag.area} \text{ intersects } \text{Panel_fxf.area}))$ moves 'Button_Multiply_Drag' to x: 273px, y: 300px → hides 'Button_Multiply_Drag' → changes style of 'Button_x_3': Transparency → changes style of 'Image_119': Transparency →
on Drag: moves 'Button_Multiply_Drag' to x: current cursor 'x', y: current cursor 'y' → changes style of 'Image_119, Button_x_3': Transparency → shows 'Button_Multiply_Drag' → changes style of 'Button_x_1, Button_x_4, Button_x_2': Border, Background →
- changes style of 'Image_95, Image_96, Image_135, Image_163' → changes style of 'Button_x_7': Border, Shadow, Background →
- 264** **on Mouse Over:** changes style of 'Button_x_2': Border, Background → changes style of 'Image_95' →
on Drag Stop: inserts 'Button_Less_Drag' into 'Panel_fxf' → changes style of 'Button_x_8': Border, Shadow, Background → changes style of 'Button_x_2': Transparency → changes style of 'Image_95': Transparency → hides 'Dashed_3' → hides 'Dashed_2' → hides 'Dashed_1' → When $\text{not}((\text{Button_Less_Drag.area} \text{ intersects } \text{Panel_fxf.area}))$ moves 'Button_Less_Drag' to x: 333px, y: 240px → hides 'Button_Less_Drag' → changes style of 'Button_x_2': Transparency → changes style of 'Image_95': Transparency →
on Drag: moves 'Button_Less_Drag' to x: current cursor 'x', y: current cursor 'y' → changes style of 'Image_95, Button_x_2': Transparency → shows 'Button_Less_Drag' → changes style of 'Button_x_1, Button_x_4, Button_x_3': Border, Background → changes style of 'Image_96, Image_135, Image_119, Image_163' → changes style of 'Button_x_8': Border, Shadow, Background →
- 265** **on Mouse Over:** changes style of 'Button_x_1': Border, Background → changes style of 'Image_135' →
on Drag Stop: inserts 'Button_Plus_Drag' into 'Panel_fxf' → changes style of 'Button_x_9': Border, Shadow, Background → changes style of 'Button_x_1': Transparency → changes style of 'Image_135': Transparency → hides 'Dashed_3' → hides 'Dashed_2' → hides 'Dashed_1' → When $\text{not}((\text{Button_Plus_Drag.area} \text{ intersects } \text{Panel_fxf.area}))$ moves 'Button_Plus_Drag' to x: 273px, y: 240px → hides 'Button_Plus_Drag' → changes style of 'Button_x_1': Transparency → changes style of 'Image_135': Transparency →
on Drag: moves 'Button_Plus_Drag' to x: current cursor 'x', y: current cursor 'y' → changes style of 'Button_x_1, Image_135': Transparency → shows 'Button_Plus_Drag' → changes style of 'Button_x_4, Button_x_3, Button_x_2': Border, Background → changes style of 'Image_95, Image_96, Image_119, Image_163' → changes style of 'Button_x_9': Border, Shadow, Background →
- 266** **on Mouse Over:** changes style of 'Image_136' → changes style of 'Rectangle_4': Border, Background →
on Page Load: When $(\text{count}(\text{Formulas.value}) < '10')$ moves 'Group_New_Formula' to x: $(('440' + ('210' * \text{mod}(\text{count}(\text{Formulas.value}), '3'))))$, y: $(('240' + ('60' * ((\text{count}(\text{Formulas.value}) - (\text{mod}(\text{count}(\text{Formulas.value}), '3') + '3')) / '3') + '1')))$ →
on Click: sets value $(\text{max}(\text{Formulas.value}, \text{id_Formula}) + '1')$ to variable(s) 'DP_ID_Formula_DG' → sets value 'Name Formula' to variable(s) 'DP_Formula_DG' → sets value " to variable(s) 'DP_Function_Formula_DG' → sets value " to variable(s) 'DP_Target_Formula_DG' → goes to 'DP - Edit Formula - DG.3' →
- 268** **on Mouse Enter:** hides 'trash_f' → shows 'trash_f_h' →
- 270** **on Mouse Over:** changes style of 'Image_138' →
on Click: show s'Dashed_3, Dashed_2, Dashed_1' → hides 'Button_Less_Drag, Button_Plus_Drag, Button_Division_Drag, Group_Number_Drag, Button_Multiply_Drag' → moves 'Button_Plus_Drag' to x: 273px, y: 240px → moves 'Button_Less_Drag' to x: 333px, y: 240px → moves 'Button_Multiply_Drag' to x: 273px, y: 300px → moves 'Button_Division_Drag' to x: 333px, y: 300px → moves 'Group_Number_Drag' to x: 273px, y: 420px → changes style of 'Dashed_3, Dashed_1': Border, Background → sets value " to 'Dashed_3, Dashed_1' → sets value " to variable(s) 'DP_Formula_DG' →
- 271** **on Click:** show s'Dyn_Tutorial' → scrolls to 'DESIGN_PROBLEM' →
on Mouse Over: changes style of 'Image_40' →

Other states

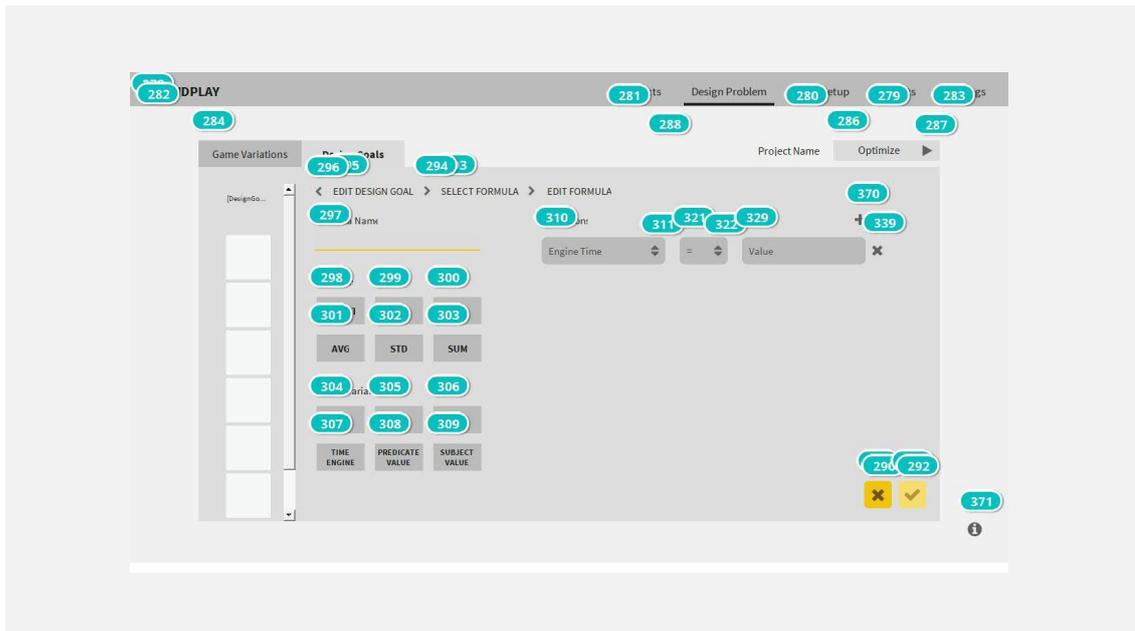


Interactions

- 272 **on Click:** hides 'Dyn_Tutorial' →
- 272 **on Mouse Over:** changes style of 'Image_121' →
- 273 **on Click:** hides 'Dyn_Tutorial' →
- 274 **on Mouse Leave:** changes style of 'Image_5': Border →
- 274 **on Mouse Enter:** changes style of 'Image_5': Border →
- 275 **on Mouse Leave:** changes style of 'Image_6': Border →
- 275 **on Mouse Enter:** changes style of 'Image_6': Border →
- 276 **on Mouse Leave:** changes style of 'Image_7': Border →
- 276 **on Mouse Enter:** changes style of 'Image_7': Border →
- 277 **on Mouse Enter:** moves 'New Project_Tut_6' to x: 116px, y: 1070px w ith easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px → moves 'New Project_Tut_6' to x: 256px, y: 1070px w ith easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px → moves 'New Project_Tut_6' to x: 396px, y: 1070px w ith easing: easeInOutQuad in 1000ms → moves 'New Project_Tut_6' to x: 56px, y: 930px →

02. Screens / CrowdPlay_Interactive

DP - Edit Formula - DG.3



Interactions

- 276** **on Page Load:** moves 'Finned-top' to x: current value, y: 'windowScrollY' →
on Scroll: moves 'Finned-top' to x: current value, y: 'windowScrollY' →
- 279** **on Mouse Over:** shows 'Line_12' →
on Click: goes to 'Results_List' →
- 280** **on Mouse Over:** shows 'Line_13' →
on Click: goes to 'Test Setup - Setup' →
- 281** **on Mouse Over:** shows 'Line_15' →
on Click: goes to 'Projects - New Test' →
- 282** **on Click:** goes to 'Inical' →
- 283** **on Mouse Over:** shows 'Line_16' →
on Click: When (User_Type = 'Admin') goes to 'Settings_Projects' → Else goes to 'Settings_Users_Designer' →
- 284** **on Mouse Over:** When (Type_Test <> 'Evaluation') changes style of 'GameVariations': Background →
on Click: When (Type_Test <> 'Evaluation') goes to 'DP - Game Variation' →
- 286** **on Mouse Over:** changes style of 'Rectangle_3': Background → changes style of 'Image_130' →
on Click: goes to 'Test Setup - Setup' →
on Page Load: sets value 'Type_Test' to 'Rectangle_3' →
- 287** **on Mouse Over:** changes style of 'Image_130' → changes style of 'Rectangle_3': Background →
on Click: goes to 'Test Setup - Setup' →
- 288** **on Page Load:** sets value 'Name_Test_Project' to 'Text_2' →
- 289** **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Select Formula - DG.2' →
- 290** **on Mouse Over:** changes style of 'Button_x': Border, Background →
on Click: goes to 'DP - Select Formula - DG.2' →
- 291** **on Mouse Over:** When not((DP_Function_Formula_DG = "")) changes style of 'Button_v': Border, Background →
on Click: When ((count(filter(Formulas.value, (id_Formula = DP_ID_Formula_DG))) = '0') and (DP_Function_Formula_DG <> "")) executes 'new(id_Formula = (max(Formulas.value, id_Formula) + '1'), Name_Formula = input_NameVariation.value, Function_Formula = DP_Function_Formula_DG, Target Variable =

DP_Target_Formula_DG' → goes to 'DP - Select Formula - DG.2'
 → Else when not((DP_Function_Formula_DG = "")) executes
 'modify(filter(Formulas.value, (id_Formula =
 DP_ID_Formula_DG)), id_Formula = undefined, Name Formula =
 input_NameVariation.value, Function Formula =
 DP_Function_Formula_DG, Target Variable =
 DP_Target_Formula_DG)' → goes to 'DP - Select Formula - DG.2'
 →

on Variable Change: When not((DP_Function_Formula_DG = ""))
 changes style of 'Button_v': Border, Background → changes style
 of 'Image_103': Transparency → When
 (DP_Function_Formula_DG = "") changes style of 'Button_v':
 Border, Background → changes style of 'Image_103':
 Transparency →
on Page Load: When not((DP_Function_Formula_DG = ""))
 changes style of 'Button_v': Border, Background → changes style
 of 'Image_103': Transparency → When
 (DP_Function_Formula_DG = "") changes style of 'Button_v':
 Border, Background → changes style of 'Image_103':
 Transparency →

292 on Click: When ((count(filter(Formulas.value, (id_Formula =
 DP_ID_Formula_DG))) = '0') and (DP_Function_Formula_DG <<
 "")) executes 'new(id_Formula = (max(Formulas.value, id_Formula)
 + '1'), Name Formula = input_NameVariation.value, Function
 Formula = DP_Function_Formula_DG, Target Variable =
 DP_Target_Formula_DG)' → goes to 'DP - Select Formula - DG.2'
 → Else when not((DP_Function_Formula_DG = "")) executes
 'modify(filter(Formulas.value, (id_Formula =
 DP_ID_Formula_DG)), id_Formula = undefined, Name Formula =
 input_NameVariation.value, Function Formula =
 DP_Function_Formula_DG, Target Variable =
 DP_Target_Formula_DG)' → goes to 'DP - Select Formula - DG.2'
 →
on Mouse Over: When not((DP_Function_Formula_DG = ""))
 changes style of 'Button_v': Border, Background →

293 on Click: goes to 'DP - Select Formula - DG.2' →
on Mouse Over: changes style of 'Text_Walk_1': Text →
 changes style of 'Image_back_1' →

294 on Click: goes to 'DP - Select Formula - DG.2' →
on Mouse Over: changes style of 'Image_back_1' → changes
 style of 'Text_Walk_1': Text →

295 on Click: goes to 'DP - Goal - DG.1' →
on Mouse Over: changes style of 'Text_Walk': Text →

296 on Click: goes to 'DP - Design Goals' →
on Mouse Over: changes style of 'Image_back' →

297 on Page Load: sets value 'DP_Formula_DG' to
 'input_NameVariation' →
on Change: sets value 'input_NameVariation.value' to variable(s)
 'DP_Formula_DG' →

298 on Click: sets value 'Count' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_1':
 Background → changes style of 'Rect_button_5, Rect_button_6,
 Rect_button_3, Rect_button_2, Rect_button_4': Background →
 sets value "" to variable(s) 'DP_Target_Formula_DG' → changes
 style of 'Rect_button_8, Rect_button_11, Rect_button_9,
 Rect_button_7, Rect_button_12, Rect_button_10': Background →
on Mouse Over: changes style of 'Rect_button_1': Background
 →
on Page Load: When (DP_Function_Formula_DG = 'Count')
 changes style of 'Rect_button_1': Background → Else when
 (DP_Function_Formula_DG = 'Max') changes style of
 'Rect_button_2': Background → Else when
 (DP_Function_Formula_DG = 'Min') changes style of
 'Rect_button_3': Background → Else when
 (DP_Function_Formula_DG = 'Avg') changes style of
 'Rect_button_4': Background → Else when
 (DP_Function_Formula_DG = 'Std') changes style of
 'Rect_button_5': Background → Else when
 (DP_Function_Formula_DG = 'Sum') changes style of
 'Rect_button_6': Background →

33 1.13.2017

299 on Click: sets value 'Max' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_2':
 Background → changes style of 'Rect_button_5, Rect_button_6,
 Rect_button_3, Rect_button_1, Rect_button_4': Background →
on Mouse Over: changes style of 'Rect_button_2': Background
 →

300 on Click: sets value 'Min' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_3':
 Background → changes style of 'Rect_button_5, Rect_button_6,
 Rect_button_2, Rect_button_1, Rect_button_4': Background →
on Mouse Over: changes style of 'Rect_button_3': Background
 →

301 on Click: sets value 'Avg' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_4':
 Background → changes style of 'Rect_button_5, Rect_button_6,
 Rect_button_3, Rect_button_2, Rect_button_1': Background →
on Mouse Over: changes style of 'Rect_button_4': Background
 →

302 on Click: sets value 'Std' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_5':
 Background → changes style of 'Rect_button_6, Rect_button_3,
 Rect_button_2, Rect_button_1, Rect_button_4': Background →
on Mouse Over: changes style of 'Rect_button_5': Background
 →

303 on Click: sets value 'Sum' to variable(s)
 'DP_Function_Formula_DG' → changes style of 'Rect_button_6':
 Background → changes style of 'Rect_button_5, Rect_button_3,
 Rect_button_2, Rect_button_1, Rect_button_4': Background →
on Mouse Over: changes style of 'Rect_button_6': Background
 →

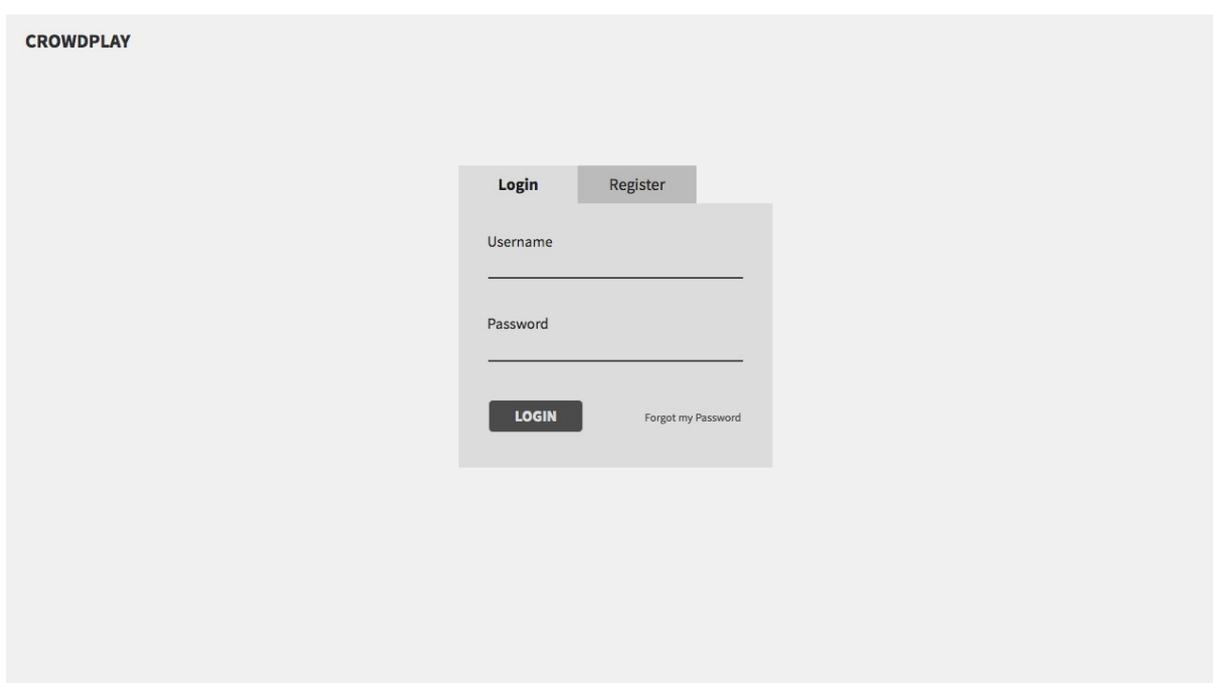
304 on Click: When (DP_Function_Formula_DG << 'Count') sets
 value 'X' to variable(s) 'DP_Target_Formula_DG' → changes style
 of 'Rect_button_7': Background → changes style of
 'Rect_button_8, Rect_button_11, Rect_button_9, Rect_button_12,
 Rect_button_10': Background →
on Mouse Over: When (DP_Function_Formula_DG << 'Count')
 changes style of 'Rect_button_7': Background →
on Page Load: When (DP_Target_Formula_DG = 'X') changes
 style of 'Rect_button_7': Background → Else when
 (DP_Target_Formula_DG = 'Y') changes style of 'Rect_button_8':
 Background → Else when (DP_Target_Formula_DG = 'Z')
 changes style of 'Rect_button_9': Background → Else when
 (DP_Target_Formula_DG = 'Time Engine') changes style of
 'Rect_button_10': Background → Else when
 (DP_Target_Formula_DG = 'Predicate Value') changes style of
 'Rect_button_11': Background → Else when
 (DP_Target_Formula_DG = 'Subject Value') changes style of
 'Rect_button_12': Background → When
 (DP_Function_Formula_DG = 'Count') changes style of
 'Rect_button_8, Text_5, Rect_button_11, Rect_button_9,
 Rect_button_7, Rect_button_12, Rect_button_10': Transparency
 → Else changes style of 'Rect_button_8, Text_5,
 Rect_button_11, Rect_button_9, Rect_button_7, Rect_button_12,
 Rect_button_10': Transparency →
on Variable Change: When (DP_Function_Formula_DG =
 'Count') changes style of 'Rect_button_8, Text_5, Rect_button_11,
 Rect_button_9, Rect_button_7, Rect_button_12, Rect_button_10':
 Transparency → Else changes style of 'Rect_button_8, Text_5,
 Rect_button_11, Rect_button_9, Rect_button_7, Rect_button_12,
 Rect_button_10': Transparency →

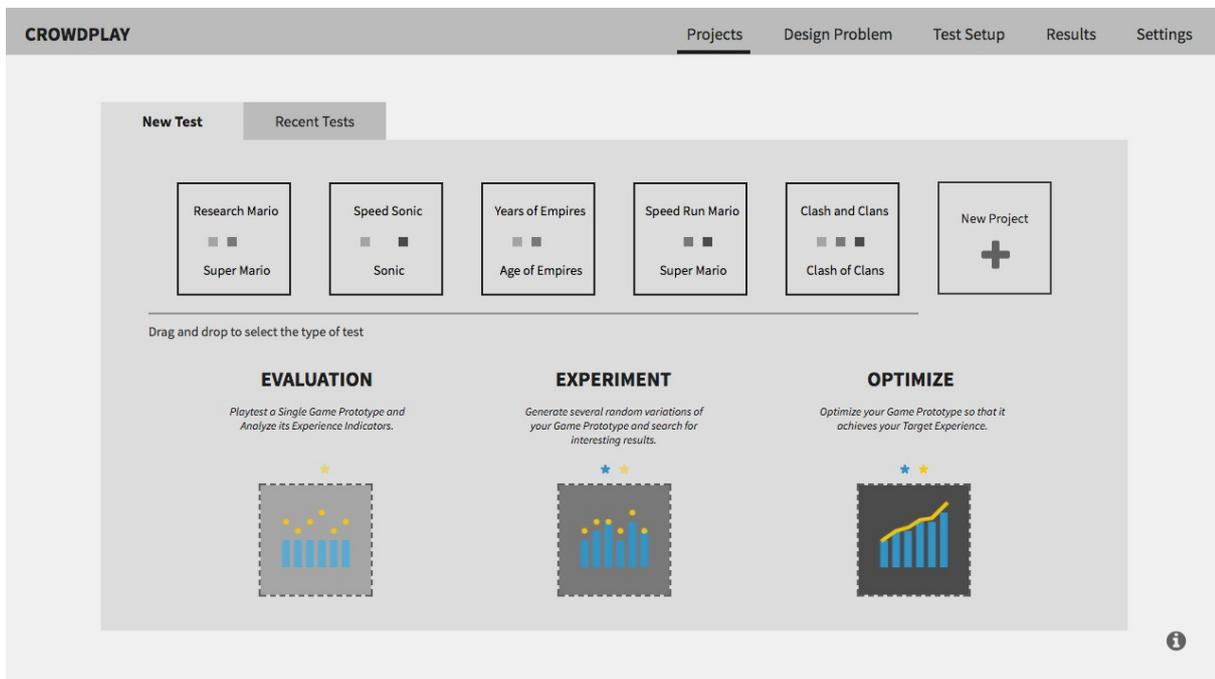
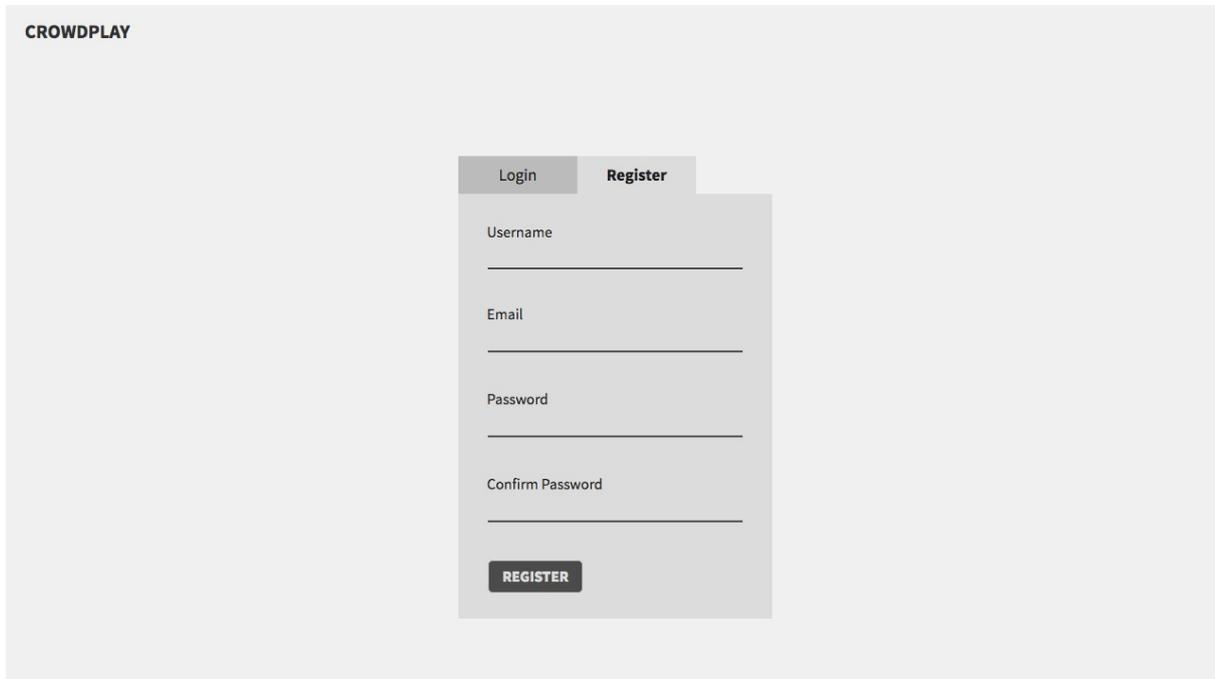
305 on Click: When (DP_Function_Formula_DG << 'Count') sets
 value 'Y' to variable(s) 'DP_Target_Formula_DG' → changes style
 of 'Rect_button_8': Background → changes style of
 'Rect_button_11, Rect_button_9, Rect_button_12, Rect_button_7,
 Rect_button_10': Background →
on Mouse Over: When (DP_Function_Formula_DG << 'Count')
 changes style of 'Rect_button_8': Background →

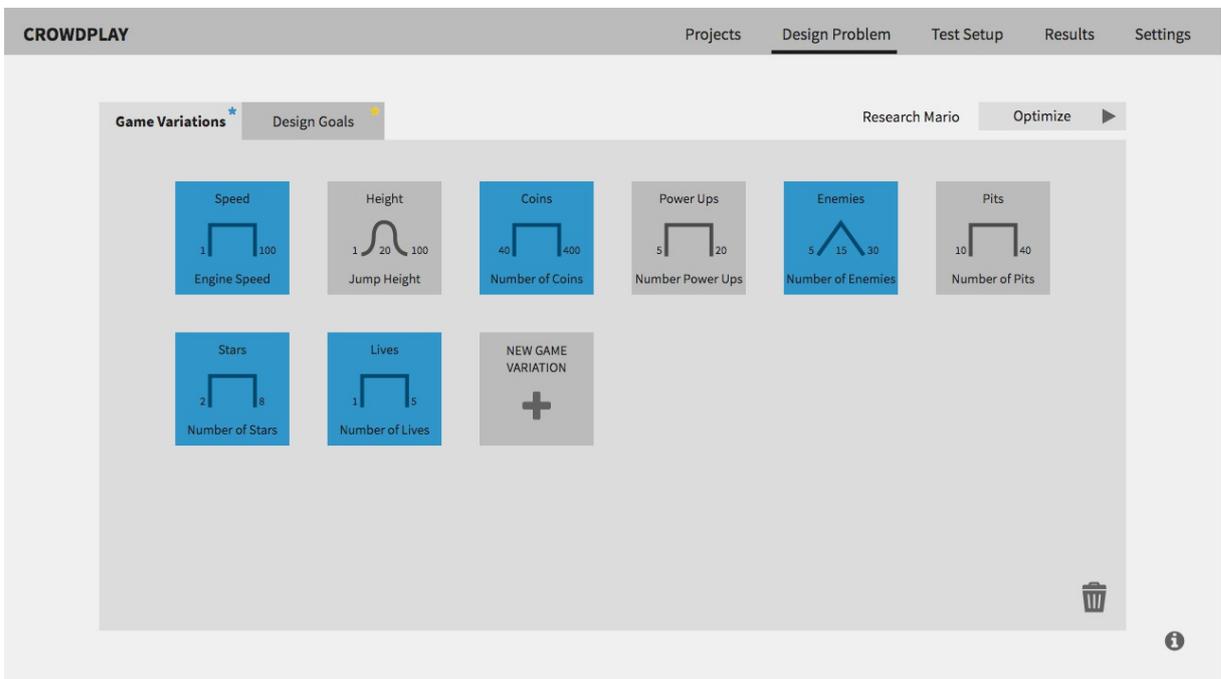
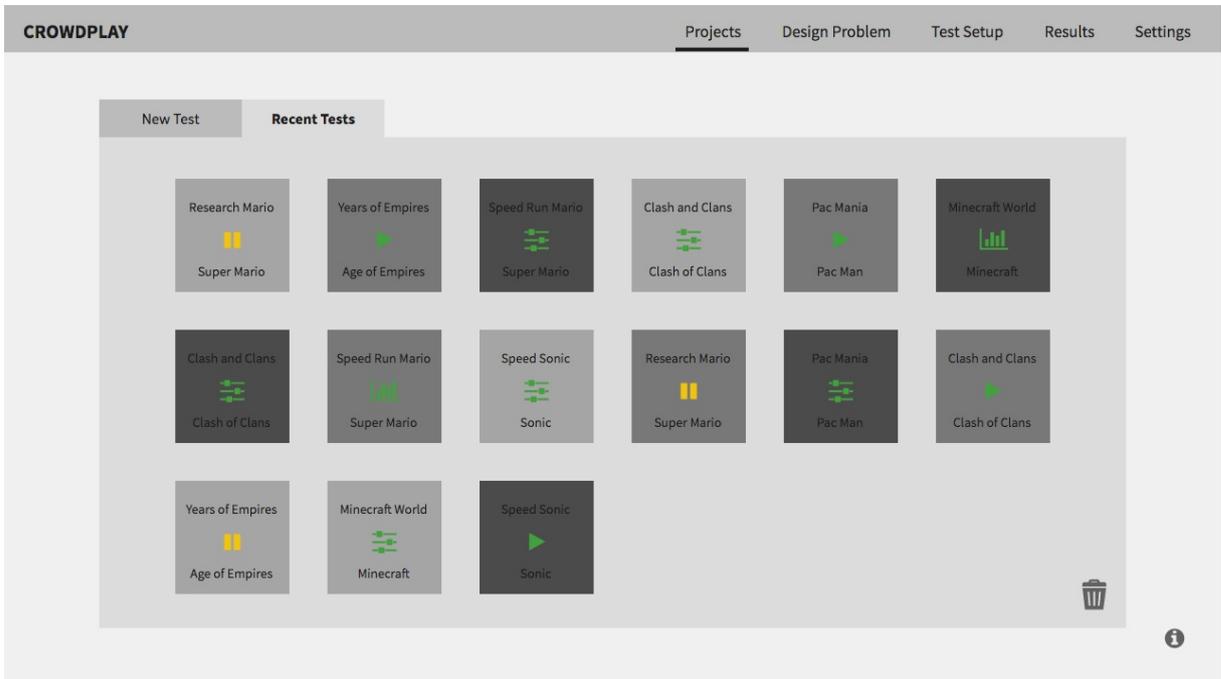
- 306** **on Click:** When (DP_Function_Formula_DG <> 'Count') sets value 'Z' to variable(s) 'DP_Target_Formula_DG' → changes style of 'Rect_button_9': Background → changes style of 'Rect_button_8, Rect_button_11, Rect_button_12, Rect_button_7, Rect_button_10': Background →
on Mouse Over: When (DP_Function_Formula_DG <> 'Count') changes style of 'Rect_button_9': Background →
- 307** **on Click:** When (DP_Function_Formula_DG <> 'Count') sets value 'Time Engine' to variable(s) 'DP_Target_Formula_DG' → changes style of 'Rect_button_10': Background → changes style of 'Rect_button_8, Rect_button_11, Rect_button_9, Rect_button_12, Rect_button_7': Background →
on Mouse Over: When (DP_Function_Formula_DG <> 'Count') changes style of 'Rect_button_10': Background →
- 308** **on Click:** When (DP_Function_Formula_DG <> 'Count') sets value 'Predicate Value' to variable(s) 'DP_Target_Formula_DG' → changes style of 'Rect_button_11': Background → changes style of 'Rect_button_8, Rect_button_9, Rect_button_12, Rect_button_7, Rect_button_10': Background →
on Mouse Over: When (DP_Function_Formula_DG <> 'Count') changes style of 'Rect_button_11': Background →
- 309** **on Click:** When (DP_Function_Formula_DG <> 'Count') sets value 'Subject Value' to variable(s) 'DP_Target_Formula_DG' → changes style of 'Rect_button_12': Background → changes style of 'Rect_button_8, Rect_button_11, Rect_button_9, Rect_button_7, Rect_button_10': Background →
on Mouse Over: When (DP_Function_Formula_DG <> 'Count') changes style of 'Rect_button_12': Background →
- 310** **on Mouse Over:** changes style of 'input_Conditions_1': Background → changes style of 'Image_62' →
on Click: When not(Conditions1_Drop.isvisible) shows 'Conditions1_Drop' → Else hides 'Conditions1_Drop' →
- 311** **on Mouse Over:** changes style of 'input_Conditions_1': Background → changes style of 'Image_62' →
on Click: When not(Conditions1_Drop.isvisible) shows 'Conditions1_Drop' → Else hides 'Conditions1_Drop' →
- 321** **on Mouse Over:** changes style of 'input_Conditions_2': Background → changes style of 'Image_63' →
on Click: When not(Conditions2_Drop.isvisible) shows 'Conditions2_Drop' → Else hides 'Conditions2_Drop' →
- 322** **on Mouse Over:** changes style of 'input_Conditions_2': Background → changes style of 'Image_63' →
on Click: When not(Conditions2_Drop.isvisible) shows 'Conditions2_Drop' → Else hides 'Conditions2_Drop' →
- 329** **on Mouse Over:** changes style of 'input_Conditions_3': Background →
on Click: When xor(xor((input_Conditions_1.value <> 'Event'), (input_Conditions_1.value <> 'Predicate')), (input_Conditions_1.value <> 'Subject')) enables 'input_Conditions_3' → sets value "" to 'input_Conditions_3' → hides 'Conditions3_Drop' → Else when (xor(xor((input_Conditions_1.value = 'Event'), (input_Conditions_1.value = 'Predicate')), (input_Conditions_1.value = 'Subject'))) and not(Conditions3_Drop.isvisible) disables 'input_Conditions_3' → shows 'Conditions3_Drop' → Else when (xor(xor((input_Conditions_1.value = 'Event'), (input_Conditions_1.value = 'Predicate')), (input_Conditions_1.value = 'Subject'))) and Conditions3_Drop.isvisible hides 'Conditions3_Drop' →
- 339** **on Mouse Over:** changes style of 'Delete_Conditions' →
on Click: When Group_Conditions_1.isvisible hides 'Group_Conditions_1' →
- 370** **on Mouse Over:** changes style of 'More_Conditions' →
on Click: When not(Group_Conditions_1.isvisible) shows 'Group_Conditions_1' → Else when not(Group_Conditions_2.isvisible) shows 'Group_Conditions_2' →
- 371** **on Click:** shows 'Dyn_Tutorial' → scrolls to 'DESIGN_PROBLEM' →
on Mouse Over: changes style of 'Image_40' →

H - Ecrãs do Protótipo Interativo

(produto final)







The screenshot shows the CROWDPLAY interface with the 'Design Problem' tab selected. The 'Game Variations' sidebar on the left lists various game elements: Speed, Height, Coins (selected), Power Ups, Enemies, Pits, and Stars. The main area is titled 'EDIT GAME VARIATION' and shows the configuration for the 'Coins' variable. The 'Select Game Variable' dropdown is set to 'Coins'. The 'Game Variation Name' is 'Coins'. The 'Probability Function' is set to a step function. The 'Variable Type' is 'Float / Rational'. The 'Description' field is empty. A graph on the right shows the variable's range from 0 to 500, with a default value of 250. The current value is set to 50. An 'Increment' of 10 is shown below the graph. There are 'X' and checkmark buttons at the bottom right of the graph area.

The screenshot shows the CROWDPLAY interface with the 'Design Problem' tab selected. The 'Game Variations' sidebar on the left lists various game elements: Speed, Height, Coins (selected), Power Ups, Enemies, Pits, and Stars. The main area is titled 'SELECT GAME VARIABLE' and shows a search bar and a grid of variables. The variables are: Game Engine, Level Design, Number of Coins, Number of Power Ups, Number of Enemies, Number of Pits, Number of Stars, and Number of Lives. The 'Number of Coins' variable is selected, and its configuration is shown on the right. The 'Number of Coins' variable has a range from 0 to 500, with a default value of 250. There are 'X' and checkmark buttons at the bottom right of the configuration area.

CROWDPLAY Projects Design Problem Test Setup Results Settings

Game Variations Design Goals Research Mario Optimize ▶

Difficulty



Retries

Collected Coins



Number_Coins

Time



Level_Completi...

Rhythm



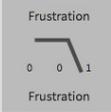
Number_Actions

Fun



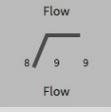
Fun

Frustration



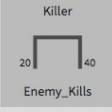
Frustration

Flow



Flow

Killer



Enemy_Kills

NEW DESIGN GOAL

+

🗑️ ⓘ

CROWDPLAY Projects Design Problem Test Setup Results Settings

Game Variations Design Goals Research Mario Optimize ▶

EDIT DESIGN GOAL

Design Goal Name

Rhythm

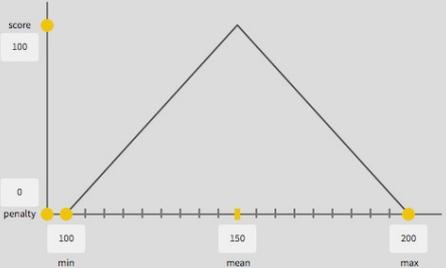
Score Function



Variable Type

Integer / Natural ▼

Description



Increment 5 + -

Select Formula

Number_Actions fx >

✕ ✓

🗑️ ⓘ

CROWDPLAY Projects Design Problem Test Setup Results Settings

Game Variations **Design Goals** Research Mario Optimize ▶

EDIT DESIGN GOAL > SELECT FORMULA

Operators: +, -, ×, ÷

Formulas: Enemy_Kills, Frustration, Number_Coins/Power Ups, Retries, Level_Completion_Time, Flow, Fun, Number_Actions, Number_Coins

Number: X

+ New Formula

Drag and drop to define Formula

Number_Actions ÷ Level_Completion_Time

✕ ✓

i

CROWDPLAY Projects Design Problem Test Setup Results Settings

Game Variations **Design Goals** Research Mario Optimize ▶

EDIT DESIGN GOAL > SELECT FORMULA > EDIT FORMULA

Formula Name: Level_Completion_Time

Conditions: Event = Won

Function: COUNT, MAX, MIN, AVG, STD, SUM

Target Variable: X, Y, Z

TIME ENGINE, PREDICATE VALUE, SUBJECT VALUE

✕ ✓

i

CROWDPLAY Projects Design Problem Test Setup Results Settings

Setup Status ▶

TEST OPTIONS

Number of Solutions: 10

Minimum Play Sessions: 5

END TEST CONDITIONS

MAX PLAY SESSIONS: 50

Click to activate/deactivate Game Variations

Speed <input type="checkbox"/>	Height <input type="checkbox"/>	Coins <input type="checkbox"/>	Power Ups <input type="checkbox"/>	Enemies <input checked="" type="checkbox"/>	Pits <input type="checkbox"/>
Stars <input type="checkbox"/>	Lives <input type="checkbox"/>				

Click to activate/deactivate Design Goals

Difficulty <input checked="" type="checkbox"/>	Collected ... <input type="checkbox"/>	Time <input type="checkbox"/>	Rhythm <input checked="" type="checkbox"/>	Fun <input checked="" type="checkbox"/>	Frustration <input type="checkbox"/>
Flow <input type="checkbox"/>	Killer <input type="checkbox"/>				

Base Game

```

    graph TD
      BaseGame[Base Game] --> S1((1))
      BaseGame --> S10((10))
      S1 --> P1_1[1]
      S1 --> P1_2[5]
      S10 --> P10_1[1]
      S10 --> P10_2[5]
      style S1 fill:#000,color:#fff
      style S10 fill:#000,color:#fff
      style P1_1 fill:#000,color:#fff
      style P1_2 fill:#000,color:#fff
      style P10_1 fill:#000,color:#fff
      style P10_2 fill:#000,color:#fff
    
```

Genetic Algorithm Details

CROWDPLAY Projects Design Problem Test Setup Results Settings

Setup Status ▶

TEST OPTIONS

Number of Solutions: 10

Minimum Play Sessions: 5

END TEST CONDITIONS

MAX PLAY SESSIONS: 50

Click to activate/deactivate Game Variations

Speed <input type="checkbox"/>	Height <input type="checkbox"/>	Coins <input type="checkbox"/>	Power Ups <input type="checkbox"/>	Enemies <input checked="" type="checkbox"/>	Pits <input type="checkbox"/>
Stars <input type="checkbox"/>	Lives <input type="checkbox"/>				

Click to activate/deactivate Design Goals

Difficulty <input checked="" type="checkbox"/>	Collected ... <input type="checkbox"/>	Time <input type="checkbox"/>	Rhythm <input checked="" type="checkbox"/>	Fun <input checked="" type="checkbox"/>	Frustration <input type="checkbox"/>
Flow <input type="checkbox"/>	Killer <input type="checkbox"/>				

Genetic Algorithm Details

Generation Algorithm: GA

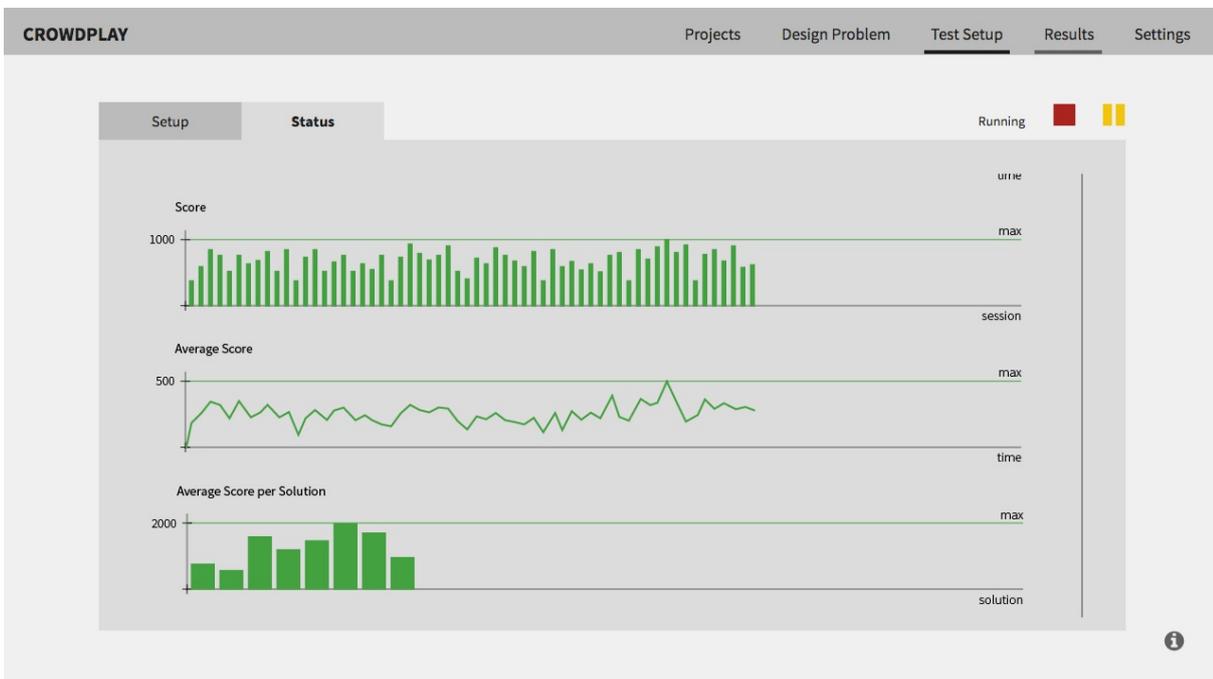
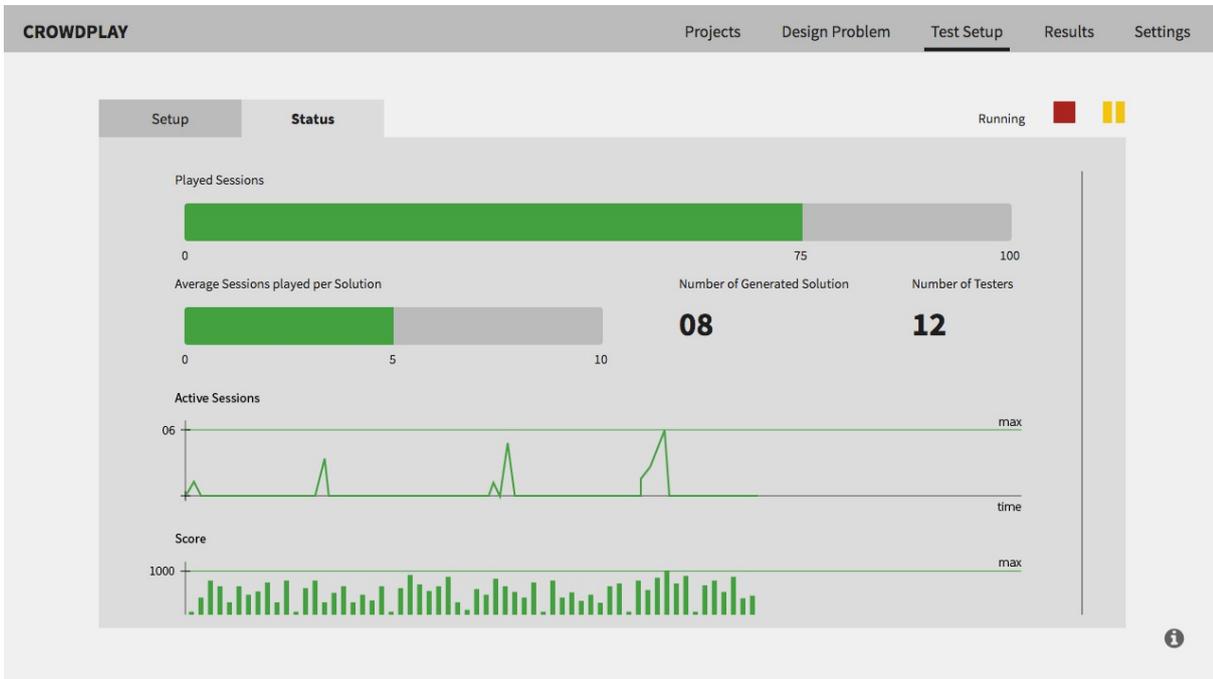
Crossover Type: n_dot

Crossover Rate: 80

Selection: tournament

Mutation Type: random

Mutation Rate: 5



CROWDPLAY Projects Design Problem Test Setup Results Settings

Bar Chart Line Chart Pie Chart Table

Drag the chart type

Select the X

Time (seconds) 10

Session Solution

Player Generation Cycle

Select the Y

Counts

Session Session by Player Session by Candidate

Session

VISUALIZE

Add new Chart

i

CROWDPLAY Projects Design Problem Test Setup Results Settings

Bar Chart Line Chart Pie Chart Table

Drag the chart type

Select the X

Time (seconds) 10

Session Solution

Player Generation Cycle

Select the Y

Count (Jumps)

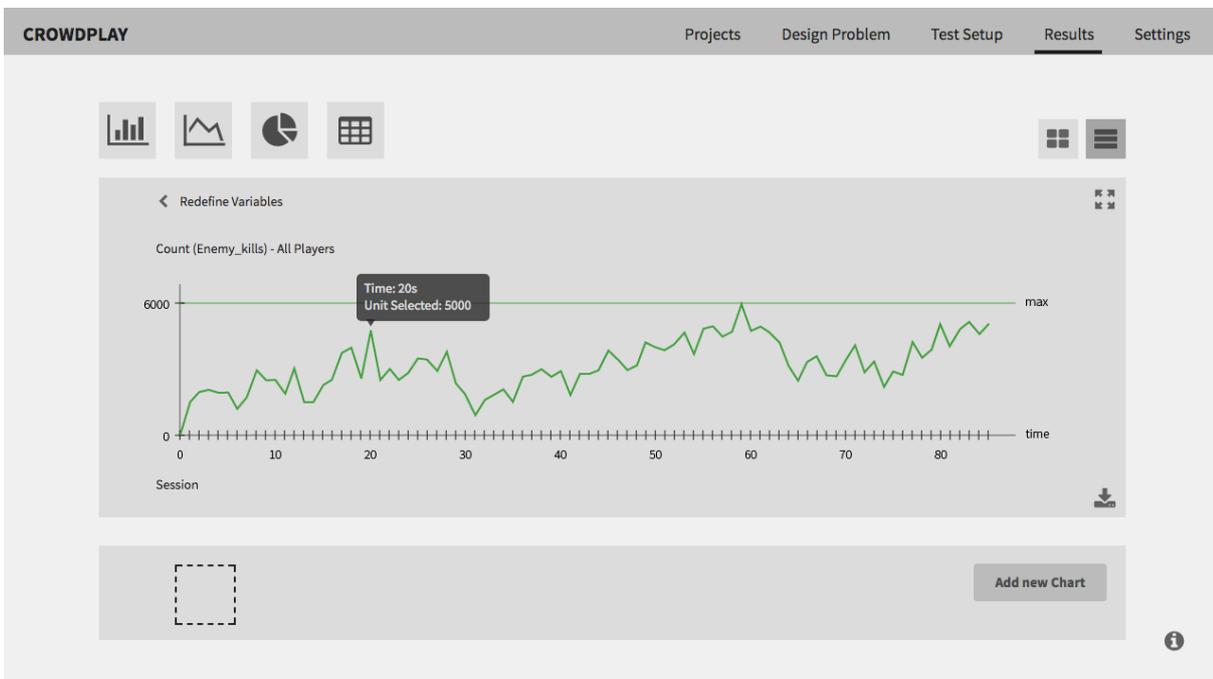
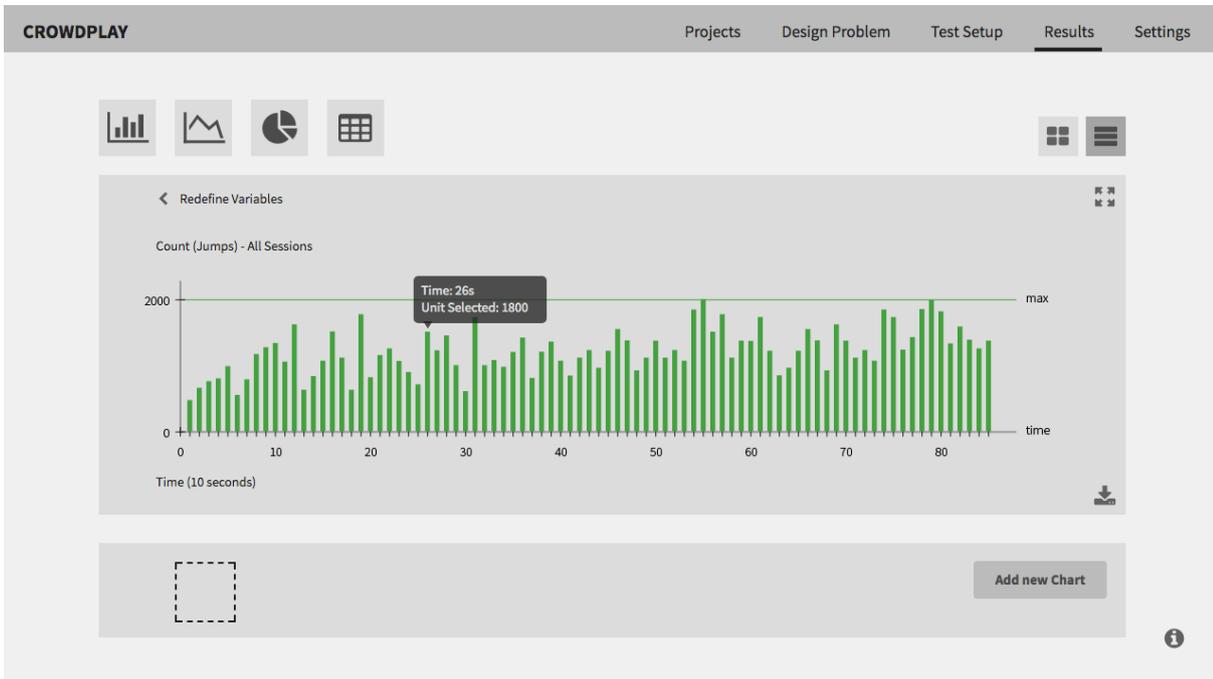
Session Session by Player Session by Candidate

All Sessions

VISUALIZE

Add new Chart

i



CROWDPLAY Projects Design Problem Test Setup Results Settings

Bar chart, Line chart, Pie chart, Table icons

Drag the chart type: [Dashed box]

Select the X:

- Time (seconds) 10
- Session
- Solution
- Player
- Generation Cycle

Select the Y:

- Counts
- Session
- Session by Player
- Session by Candidate

Session

VISUALIZE

Add new Chart

i

CROWDPLAY Projects Design Problem Test Setup Results Settings

Bar chart, Line chart, Pie chart, Table icons

Drag the chart type: [Table icon]

Select the X:

- Time (seconds) 10
- Session
- Solution
- Player**
- Generation Cycle

Select the Y:

- Average Value (Design Goals)
- Time

VISUALIZE

Add new Chart

i

CROWDPLAY Projects Design Problem Test Setup **Results** Settings

Bar Chart Line Chart Pie Chart Table

← Redefine Variables

Time	Count (Runs)	Count (Jumps)
0	0	0
10	1	1
20	0	1
30	1	0
40	1	1
50	1	0
60	0	1
70	1	1
80	0	1
90	1	0
100	1	1
110	1	0
120	0	1
130	0	0
140	1	1
150	0	0
160	1	0
170	1	1

Drag the chart type



Select the X

Time (seconds) 10

Session Solution

Player Generation Cycle

Select the Y

Average Value (Design Goals)

Rhythm

VISUALIZE

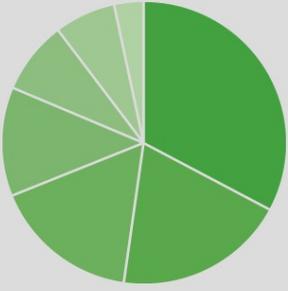
CROWDPLAY Projects Design Problem Test Setup **Results** Settings

Bar Chart Line Chart Pie Chart Table

← Redefine Variables

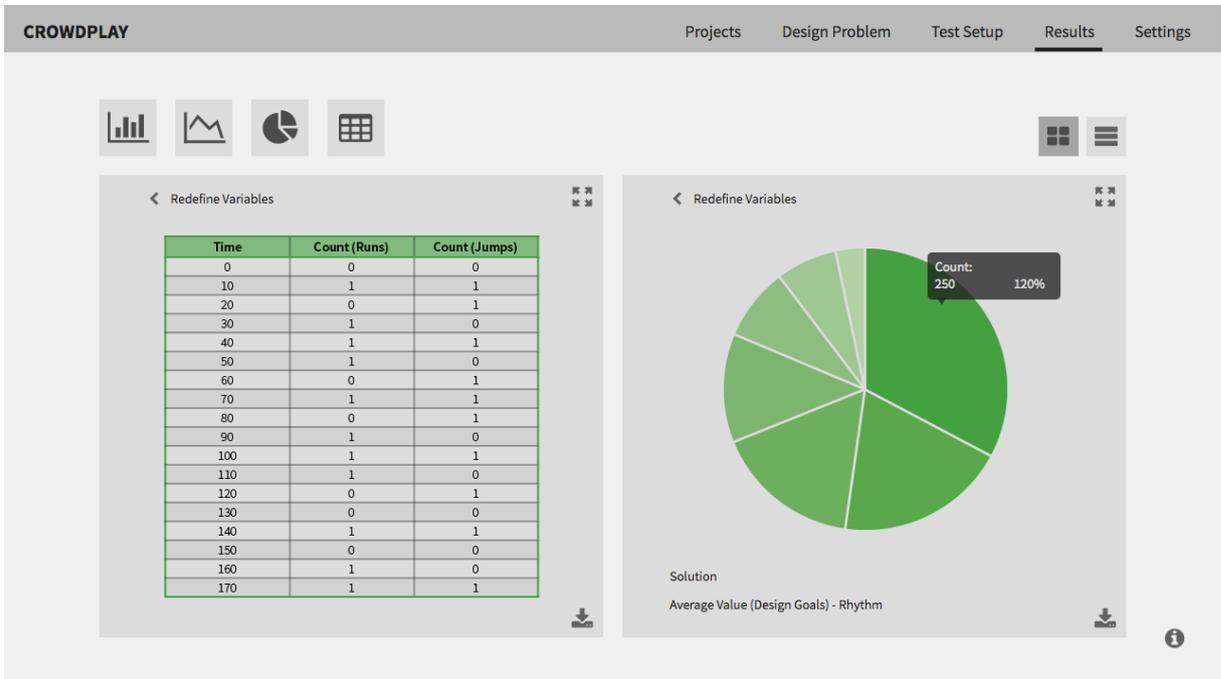
Time	Count (Runs)	Count (Jumps)
0	0	0
10	1	1
20	0	1
30	1	0
40	1	1
50	1	0
60	0	1
70	1	1
80	0	1
90	1	0
100	1	1
110	1	0
120	0	1
130	0	0
140	1	1
150	0	0
160	1	0
170	1	1

← Redefine Variables



Solution

Average Value (Design Goals) - Rhythm



CROWDPLAY Projects Design Problem Test Setup Results Settings

Projects
Users

Logout 

Research Mario

■ ■

Super Mario

Speed Sonic

■ ■ ■

Sonic

Years of Empires

■ ■

Age of Empires

Speed Run Mario

■ ■

Super Mario

Clash and Clans

■ ■ ■ ■

Clash of Clans

Pac Mania

■ ■

Pac Man

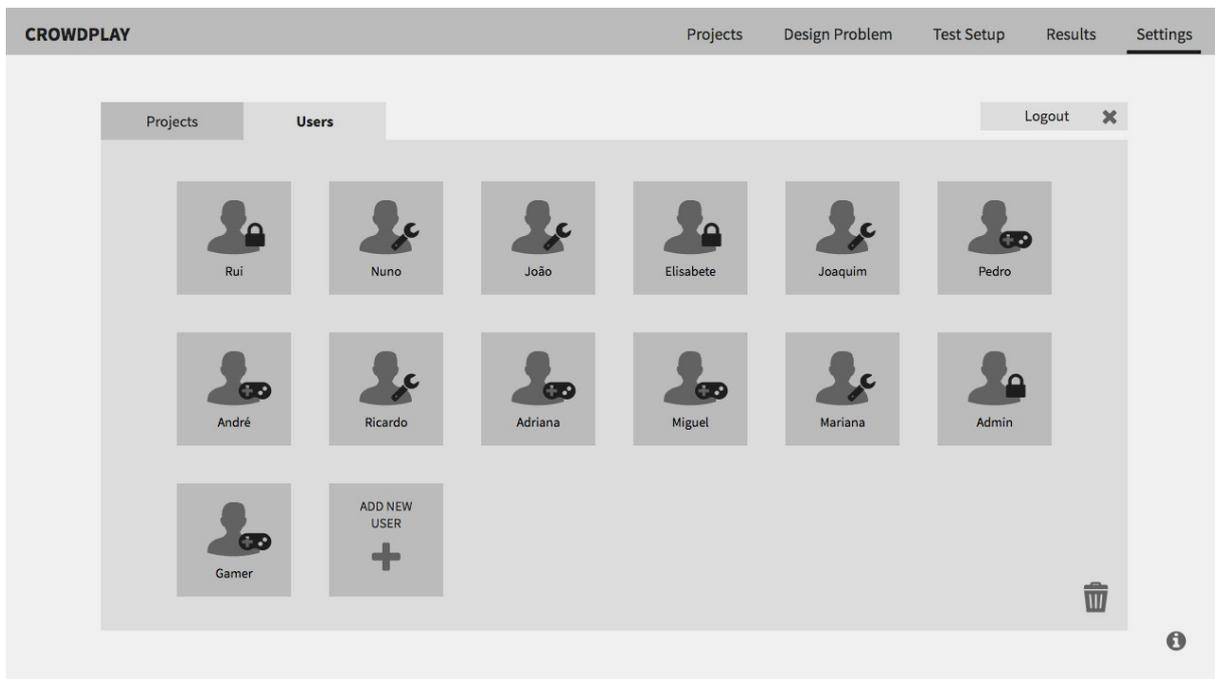
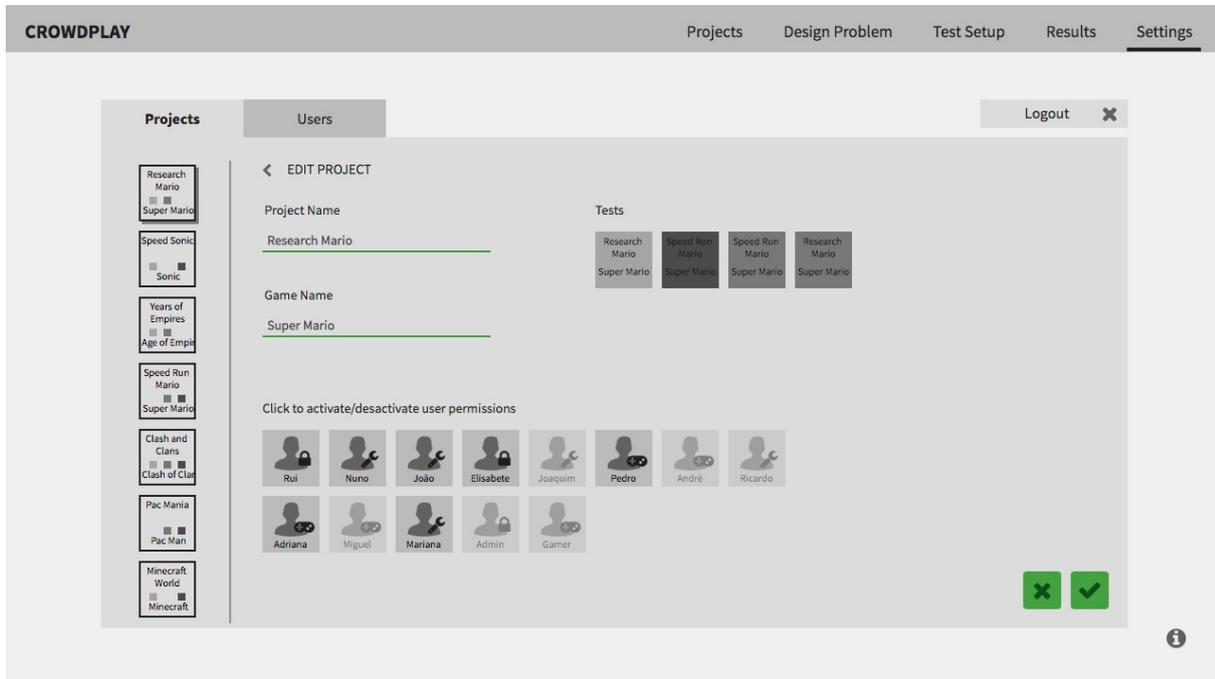
Minecraft World

■ ■

Minecraft

NEW PROJECT

+



CROWDPLAY Projects Design Problem Test Setup Results Settings

Projects **Users** Logout X

EDIT USER

User Name: Elisabete

Account Type: Admin, Designer, Gamer

Email: egsimoes@dei.uc.pt

Click to activate/desactivate Project permissions

Research Mario Super Mario	Speed Sonic Sonic	Years of Empires Age of Empir	Speed Run Mario Super Mario	Clash and Clans Clash of Clas	Pac Mania Pac Man	Minecraft World Minecraft
-------------------------------	----------------------	----------------------------------	--------------------------------	----------------------------------	----------------------	------------------------------

X ✓

i

CROWDPLAY Projects Design Problem Test Setup Results Settings

Account **Projects** Logout X

EDIT ACCOUNT

User Name: João

Change Password: *****

Account Type: Designer

Email: joao@dei.uc.pt

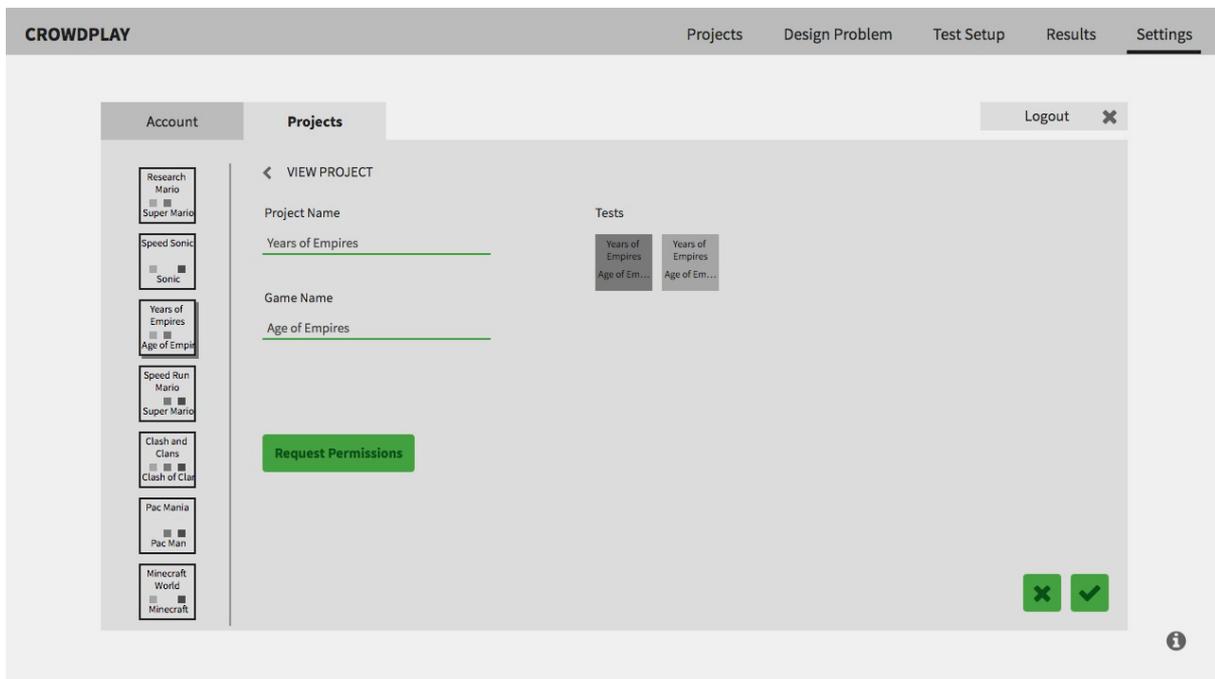
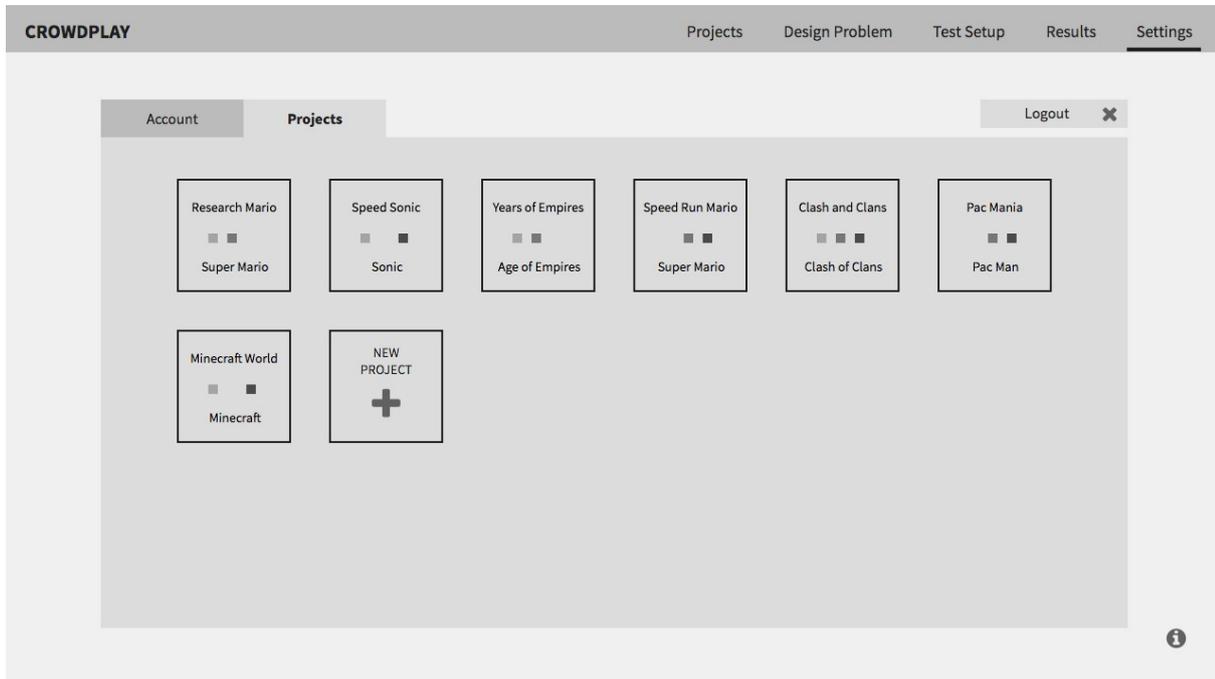
Confirm Password: *****

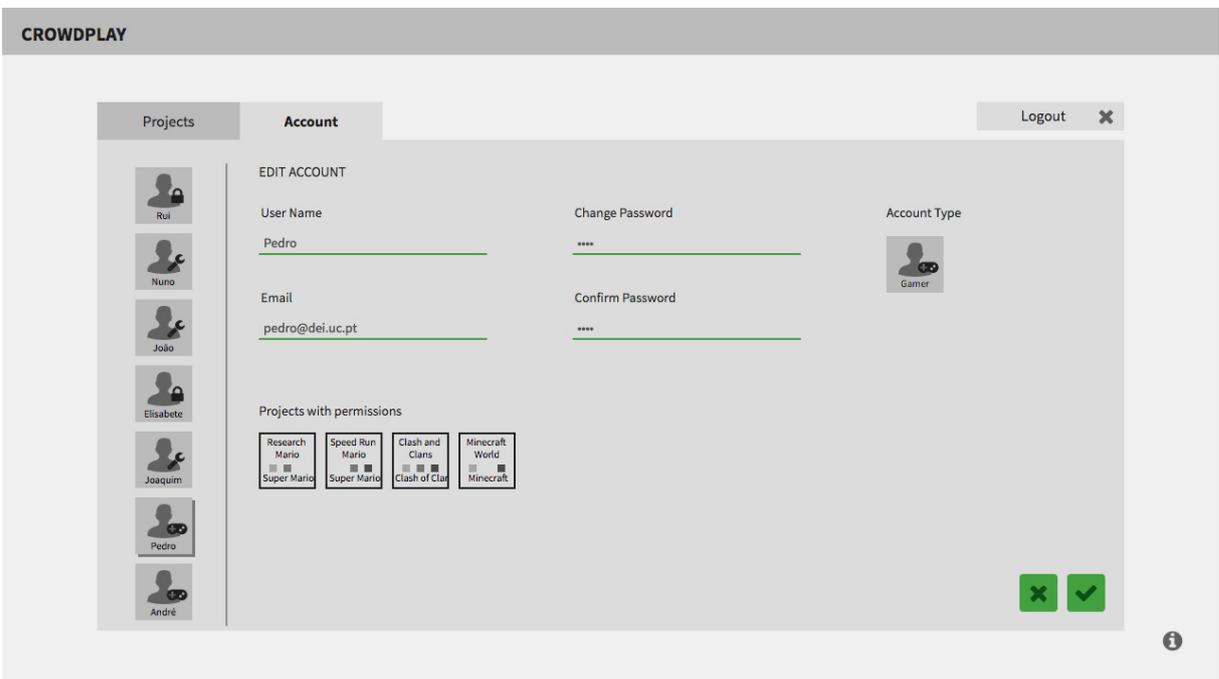
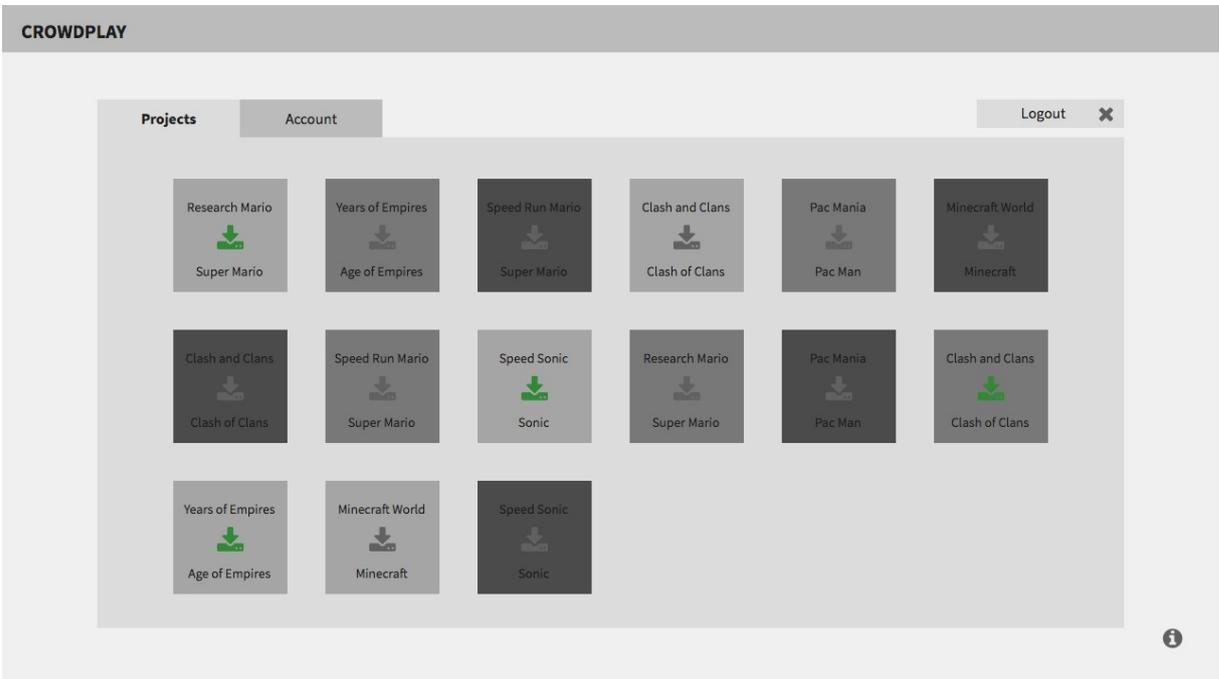
Projects with permissions

Research Mario Super Mario	Speed Run Mario Super Mario	Clash and Clans Clash of Clas	Minecraft World Minecraft
-------------------------------	--------------------------------	----------------------------------	------------------------------

X ✓

i

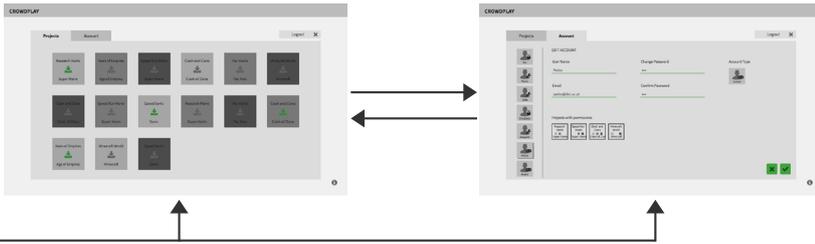




I - Mapa do Website

Re-Crowdplay





J - Guião de Tarefas (2^{os} Testes de Usabilidade)

TAREFAS

T1 - Registrar com a seguinte conta:

Username: _____

Email: _____@dei.uc.pt

Password: crowdplay

T1.1 - Entrar com a respectiva conta

T2 - Iniciar um teste do projeto "Research Mario" do tipo "Optimize"

T3 - Ativar a variação "Power Ups" nas variações de jogo ("Game Variation")

T4 - Editar a variação de jogo "Coins"

T4.1 - Alterar a variável para "Number of Coins" ("Level Design")

T4.2 - Selecionar o gráfico do tipo "Triangular"

T4.3 - Definir para variar entre 50 e 450

T4.4 - Com um incremento de 5 e guardar a variação de jogo

T5 - Adicionar um objetivo de design no "Design Goals"

T5.1 - Com o nome "Collector"

T5.2 - Selecionar o gráfico do tipo "Step"

T5.3 - Definir para variar entre 90 e 110

T5.4 - Selecionar a fórmula com o nome "Number_Coins/Power Ups" e guardar o objetivo de design

T6 - Editar o objetivo de design "Rhythm"

T6.1 - Adicionar uma nova fórmula com o nome "Level_Completion_Time"

T6.2 - Selecionar a função "MAX" e selecionar "Time Engine" para a "Target Variable"

T6.3 - Definir a condição "Event = Won" e guardar a fórmula

T6.4 - Definir a fórmula para calcular "Number_Actions" a dividir por "Level_Completion_Time"

T7 - Apagar o objetivo de design "Killer"

T8 - Ir para a página do Test Setup

T8.1 - Desativar o objetivo de design "Collector"

T8.2 - Definir a condição de fim de teste para "Manual"

T8.3 - Iniciar teste, pausar e pará-lo

T9 - Na página dos Resultados, escolher ver os dados pelo gráfico de barras

T9.1 - Escolher para o x "Time (10 seconds)"

T9.2 - Escolher para o y "Count (jumps)"

T9.3 - Escolher para todas as Sessões e visualizar gráfico

T10 - Alterar a vista para "Grid" (em vez de "List")

T10.1 - Escolher visualizar uma Tabela (Data Table) com o x igual a "Player"

T10.2 - Escolher para o y "Average Value (Design Goals)" para o Design Goal "Time"

T10.3 - Adicionar um segundo y com "Average Value (Design Goals)" para o Design Goal "Rhythm"

T10.4 - Visualizar o gráfico

T11 - Nos Settings, ver para que projetos se tem permissões

T11.1 - Pedir permissões para o projeto "Years of Empires"

K - Tutorial

PROJECTS_NEW TEST

EVALUATION



Playtest a Single Game Prototype and Analyze its Experience Indicators.

- Without Game Variations
- ★ With optional Design Goals

EXPERIMENT



Generate several random variations of your Game Prototype and search for interesting results.

- ★ With mandatory Game Variations
- ★ With optional Design Goals

OPTIMIZE

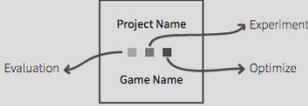


Optimize your Game Prototype so that it achieves your Target Experience.

- ★ With mandatory Game Variations
- ★ With mandatory Design Goals

NEW TEST

Indicates if there is already a test of type



Drag and drop the project to select the type of test

Project Name
Game Name

EXPERIMENT



EVALUATION

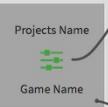


OPTIMIZE



PROJECTS_RECENT TESTS

RECENT TESTS



The icons represent the state where the project is

-  Design Problem
-  Running Test
-  Complete test with
-  Test Paused

The color indicates the type of test. Experiment, Evaluation or Optimize.

DESIGN PROBLEM

Game Variations

G. Variation Name



G. Variable Name

Indicates the type of Probability Function graph and its interval

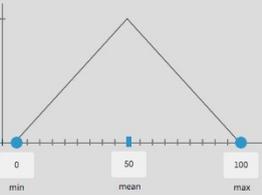
- ☞ One click to activate / deactivate
- ☞ Double click to open / edit

Design Goals

Design Goal Name



Formula Name



To change the values can be by:

- Writing
- Arrows
- Drag
- Plus / Minus

For increment 10 press TAB

+

-

