



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Mestrado em Engenharia Informática

Dissertação

Relatório Final

Porting e Melhoria do Videojogo “Falcao vs Aliens”

Daniel Francisco Lopes de Carvalho
dflopes@student.dei.uc.pt

Orientador do Departamento: Tiago Baptista

Orientador da Empresa: Ricardo Lopes

Coimbra
02 de Setembro de 2015

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologias
Universidade de Coimbra

Pólo II, Pinhal de Marrocos
3030-290 Coimbra



WingzStudio

Bairro Sousa Pinto, 35 - 1º
3030-393 Coimbra



Porting e Melhoria do Videojogo “Falcao Vs Aliens”

Daniel Francisco Lopes de Carvalho

dflopes@student.dei.uc.pt

Dissertação de Mestrado em Engenharia Informática

Relatório Final

Júri Arguente:

Alcides Fonseca

amaf@dei.uc.pt

Júri Vogal:

Licínio Roque

lir@dei.uc.pt

Coimbra

02 de Setembro de 2015

Abstract

Software porting has a big role on video games scene, the reason behind is the high number of platforms available. To make a video game able to run on most platforms it's necessary to perform the porting or during the original development ensure the maximum compatibility between platforms.

This document describes the project of porting and improvement of the video game "Falcao vs Aliens", created for mobile devices by WingzStudio. The need to attract more players and increase the retention rate, make it necessary to increase the compatibility with new platforms like Windows, Mac OS X, browsers and Facebook, as well as developing new features and improvements.

The first phase will include the study around porting and the analysis of several tools that simplifies that process. Afterwards the analysis of the video game with the intent of finding new improvement opportunities and issues. At last the development process inspired by waterfall model.

Key-Words: Porting of Video Games, Porting Process, Video Game Development, Game Engine, Falcao vs Aliens, WingzStudio.

Resumo

O *porting* de *software* tem grande peso na área dos videojogos, isto porque o número de plataformas disponíveis é cada vez maior. De forma a que um videojogo execute no máximo de plataformas é necessário efetuar o *porting* do mesmo ou, durante o processo de desenvolvimento original garantir a maior compatibilidade possível entre plataformas.

Neste documento é retratado o projeto de *porting* e melhoria do videojogo “Falcao vs Aliens”, o qual foi desenvolvido pela empresa WingzStudio para dispositivos móveis. Devido à necessidade de captar mais jogadores e aumentar a taxa de retenção é importante torná-lo compatível com mais plataformas, principalmente as nativas (*Windows* e *Mac OS X*) e *web* (navegadores atuais e *Facebook*), assim como melhorar e adicionar novas funcionalidades.

Numa primeira fase foi elaborado um estudo sobre o *porting* e uma análise de ferramentas que facilitam esse processo. Posteriormente realizou-se uma análise ao videojogo de forma a descobrir oportunidades de melhoria e problemas da versão original. Seguidamente efetuou-se o processo de desenvolvimento, assente na metodologia *waterfall*.

Palavras-Chave: Porting de Videojogos, Processo de Porting, Desenvolvimento de Videojogos, Motor de Jogo, Falcao vs Aliens, WingzStudio.

Índice de Figuras

Figura 1 - <i>Unity Editor</i>	11
Figura 2 - Menu de Seleção da Plataforma a Compilar.....	13
Figura 3 - <i>Unity Asset Store (Unity Editor)</i>	14
Figura 4 - <i>Cocos Studio</i>	18
Figura 5 - Esquema representativo das fases da metodologia <i>waterfall</i>	28
Figura 6 - Planeamento do Projeto - 1º Semestre.....	29
Figura 7 - Planeamento do Projeto - 2º Semestre.....	30
Figura 8 - Planeamento do Projeto - Implementação - Parte 1/2.....	30
Figura 9 - Planeamento do Projeto - Implementação - Parte 2/2.....	31
Figura 10 - “Falcao vs Aliens” - Exemplo de Nível.....	33
Figura 11 - Arquitetura - Visão Global.....	45
Figura 12 - Arquitetura - Mecanismo de Implementação do Core.....	46
Figura 13 - Arquitetura - Componentes do Core.....	47
Figura 14 - Arquitetura - Scenes.....	48
Figura 15 - <i>Design</i> de um nível do videojogo (com barreira a bloquear a baliza).....	51
Figura 16 - <i>Design</i> de um nível do videojogo (sem barreira a bloquear a baliza).....	53
Figura 17 - Fluxo Menus - “Falcao vs Aliens”.....	54
Figura 18 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo Inicial - Parte 1/5.....	55
Figura 19 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo da Seleção de Modos de Jogo - Parte 2/5.....	56
Figura 20 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo da Sessão de Jogo - Parte 3/5.....	57
Figura 21 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo do Menu de Opções - Parte 4/5.....	58
Figura 22 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo do Menu de <i>Achievements</i> - Parte 5/5.....	59
Figura 23 - Mapa do Mundo - “Falcao vs Aliens” - Nova Versão.....	67

Figura 24 - <i>Leaderboard</i> - “Falcao vs Aliens” - Nova Versão.....	68
Figura 25 - Seleção de Modos de Jogo - “Falcao vs Aliens” - Nova Versão.....	69
Figura 26 - Submenu do Estádio 4 - Mapa do Mundo - “Falcao vs Aliens” - Nova Versão.....	70
Figura 27 - <i>Leaderboard</i> (à esquerda) e Início de Nível (à direita) - Modo História - “Falcao vs Aliens” - Nova Versão.....	71
Figura 28 - Menu Modo Desafio - “Falcao vs Aliens” - Nova Versão.....	74
Figura 29 - Gráfico de Percentagem de Sucesso por Nível - 1º Teste “Falcao vs Aliens” - Versão de Desenvolvimento.....	76
Figura 30 - Gráfico de Classificação da Dificuldade por Níveis - 1º Teste “Falcao vs Aliens” - Versão de Desenvolvimento.....	76
Figura 31 - Gráfico de Percentagem de Sucesso por Nível - 2º Teste “Falcao vs Aliens” - Versão de Desenvolvimento.....	77
Figura 32 - Gráfico de Classificação da Dificuldade por Níveis - 2º Teste “Falcao vs Aliens” - Versão de Desenvolvimento.....	78
Figura 33 - Gráfico de Percentagem de Sucesso por Nível - Comparação entre o 1º e 2º Testes “Falcao vs Aliens” - Versão de Desenvolvimento.....	78
Figura 34 - Gráfico de Classificação da Dificuldade por Níveis - Comparação entre o 1º e 2º Testes “Falcao vs Aliens” - Versão de Desenvolvimento.....	79
Figura 35 - Comparação dos Ícones do Menu de Seleção de Modos de Jogo - “Falcao vs Aliens”.....	80

Índice de Tabelas

Tabela 1 - Comparação entre Motores de Jogo - <i>Unity Personal Edition 5.1.2f</i> vs <i>Cocos2d-x 3.7</i>	23
Tabela 2 - Comparação entre os Requisitos Funcionais - “Falcao vs Aliens”	42
Tabela 3 - Comparação entre os Requisitos Não-funcionais - “Falcao vs Aliens” ..	44
Tabela 4 - Comparação entre a versão original e a nova - “Falcao vs Aliens”	83
Tabela 5 - Comparação da nova versão nas várias plataformas - “Falcao vs Aliens”	84

Glossário

Achievements	Desafios que ao serem cumpridos oferecem recompensas ao jogador, como por exemplo troféus ou medalhas.
API	Conjunto de instruções e padrões de programação para acesso a uma aplicação ou <i>software</i> .
Assets	Conteúdo artístico que é inserido num videojogo, como por exemplo modelos 3D e efeitos sonoros.
Bottlenecks	Parte do <i>software</i> em que a performance é lenta, podendo em casos extremos, causar a interrupção do processo.
Bug	Comportamento incorreto do sistema, normalmente devido a um erro no código do mesmo.
Debugging	Processo que envolve a procura e redução de falhas no código de um sistema.
Designer	Indivíduo responsável pela criação de material artístico.
Design Patterns	Padrões que oferecem soluções abstratas de alto nível com o objetivo de resolver determinados problemas de implementação.
FPS	Número de imagens processadas/renderizadas por segundo.
Gameover	No contexto deste projeto significa derrota, por exemplo quando o jogador perde todas as vidas.
Gamepad	Dispositivo usado como controlador para jogar videojogos.
Indie	Significa independente. No contexto dos videojogos significa sem o apoio de uma produtora.
Input	Entrada caracterizada pelo fornecimento de dados, os quais serão processados para produzir informações de saída.
IDE	<i>Software</i> que reúne um conjunto de funcionalidades com o intuito de ajudar e agilizar o processo de desenvolvimento de <i>software</i> .
JSON	Formato de troca de mensagens que oferece uma facilidade de utilização/leitura tanto para máquinas como humanos.

Loading	Carregamento para a memória dos componentes necessários para a execução da próxima parte de um videojogo.
Middleware	<i>Software</i> que se encontra entre sistemas e converte informações entre eles, por exemplo um sistema que gere base de dados.
Multithreading	Mecanismo que permite que um processo seja dividido em várias linhas de execução, podendo executar em diferentes processadores, sem a necessidade de criar múltiplos processos.
Plugin	<i>Software</i> que serve para adicionar funções a outro <i>software</i> , ou seja que fornece determinada(s) funcionalidade(s).
Porting	Processo que consiste em adaptar um videojogo a plataformas para as quais este não foi inicialmente desenvolvido.
Power ups	Bónus possíveis de apanhar pelo jogador, oferecendo-lhe vantagem. Quando são mencionados <i>power ups</i> negativos significa o oposto, ou seja que provocam desvantagem.
OpenGL	<i>API</i> que permite a renderização, isto é, processamento digital, de gráficos 2D e 3D.
Open source	<i>Software</i> de código aberto e distribuição livre.
Reset	Apagar as configurações atuais e voltar às de origem.
Save Files	Ficheiros criados por um sistema com o objetivo de guardar informações relacionadas ao uso do mesmo. No caso dos videojogos estes normalmente guardam o progresso do jogador.
Scripting	Programação baseada em linguagens que normalmente são interpretadas em vez de compiladas.
Sistema de Partículas	Sistema que permite a representação de objetos cuja superfície não estão bem definida, como por exemplo fumo e água.
Singleton	É um <i>design pattern</i> que garante a existência de apenas uma instância de uma classe/componente.
Slider	Objeto que desliza.
Socket	Primitiva de comunicação pela rede através da qual processos comunicam.
Spritesheet	Conjunto de imagens num único ficheiro.

TCP	Conjunto de protocolos de comunicação entre sistemas em rede.
Thread	Linha de execução de um processo.
Tile Map	Representação de um cenário 2D que contém armazenada informação sobre as células que o compõem, assim como a posição de cada célula e a textura correspondente.
Touch	Termo usado para dispositivos em que a tela é sensível ao toque, ou seja, <i>touch</i> significa tocar na tela.

Acrónimos

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
FPS	Frames Per Second
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
LLVM	Low Level Virtual Machine
TCP	Transmission Control Protocol

Índice

1. Introdução.....	1
2. Estado da Arte.....	3
2.1. <i>Porting</i> de Videojogos.....	3
2.1.1 Desenvolvimento do Videojogo.....	4
2.1.2 Processo de <i>Porting</i>	7
2.2. Motores de Jogo.....	10
2.2.1. <i>Unity Personal Edition 5.2.1f1</i>	10
2.2.1.1. <i>Unity Editor</i>	11
2.2.1.2. Renderização.....	12
2.2.1.3. <i>Scripting</i>	12
2.2.1.4. Suporte a Plataformas.....	12
2.2.1.5. Documentação, Suporte e Tempo de Aprendizagem.....	13
2.2.1.6. <i>Open Source</i>	14
2.2.1.7. Preços e Licenças.....	15
2.2.1.8. Capacidades de Desenvolvimento.....	15
2.2.1.9. Conclusão.....	16
2.2.2. <i>Cocos2d-JS 3.1</i>	16
2.2.2.1. Renderização.....	17
2.2.2.2. Performance.....	17
2.2.2.3. <i>Open Source</i>	17
2.2.2.4. Editor Visual.....	17
2.2.2.5. Documentação, Suporte e Tempo de Aprendizagem.....	18
2.2.2.6. Comunidade e Extras.....	19
2.2.2.7. Capacidades de Desenvolvimento.....	19
2.2.2.8. Conclusão.....	19
2.2.3. Comparação dos Motores de Jogo.....	20
2.2.3.1. Suporte a Plataformas.....	20
2.2.3.2. Preços e Licenças.....	20
2.2.3.3. Suporte 2D.....	21

2.2.3.4. Física 2D.....	21
2.2.3.5. Áudio.....	21
2.2.3.6. Editor Visual.....	21
2.2.3.7. Linguagens de Programação.....	22
2.2.3.8. <i>Open Source</i>	22
2.2.3.9. Documentação e Suporte.....	22
2.2.3.10. Loja Integrada.....	23
2.3. Outras Ferramentas/Abordagens.....	24
3. Objetivos e Método de Abordagem.....	25
3.1. Objetivos.....	25
3.2. Abordagem de Desenvolvimento.....	27
3.3. Planeamento de Atividades.....	28
3.3.1. Primeiro Semestre.....	28
3.3.2. Segundo Semestre.....	29
4. Conceito, Requisitos e Arquitetura.....	32
4.1. Conceito.....	32
4.2. História/Enredo.....	32
4.3. Descrição Geral do Videojogo.....	33
4.4. Requisitos Funcionais e Não-funcionais.....	34
4.5. Arquitetura Geral.....	45
4.6. Arquitetura Detalhada.....	47
4.6.1. <i>Core</i>	47
4.6.2. <i>Scenes</i>	48
4.7. <i>Design</i>	50
4.7.1. <i>Design</i> de um Nível do Videojogo.....	51
4.7.2. Organização Geral do Fluxo.....	53
4.7.3. Organização Detalhada do Fluxo.....	54
5. Implementação, Testes e Resultados.....	60
5.1. Considerações Acerca da Versão Original.....	60
5.2. Fases de Implementação.....	60
5.2.1. Implementação do Conteúdo Original.....	61
5.2.1.1. Principais Mecânicas e Jogabilidade.....	61

5.2.1.2. Menu Principal e de Opções.....	64
5.2.1.3. Menu de <i>Achievements</i>	65
5.2.1.4. Coleção de Cromos.....	66
5.2.1.5. Sequência Animada de Introdução.....	66
5.2.1.6. Mapa do Mundo.....	66
5.2.1.7. <i>Leaderboard</i>	67
5.2.2. Implementação de Novas Funcionalidades/Melhorias.....	68
5.2.2.1. Menu de Seleção de Modos de Jogo.....	68
5.2.2.2. Modos de Jogo Arcade e História.....	69
5.2.2.3. Melhoria e Introdução de Novos <i>Power Ups</i>	71
5.2.2.4. Integração com o <i>Facebook</i>	72
5.2.2.5. Alterações à Base de Dados.....	73
5.2.2.6. Animação de Conclusão dos Modos Arcade e História.....	73
5.2.2.7. Modo de Jogo Desafio.....	74
5.3. Testes.....	75
5.4. Comparação Entre Versões.....	80
5.4.1. Versão Original vs Nova Versão.....	81
5.4.2. Comparação da Nova Versão entre Plataformas.....	83
6. Conclusão.....	86
7. Referências.....	88
8. Anexos.....	98
8.1. Anexo A - Documento de Propostas de Melhoria - Videojogo “Falcao vs Aliens”.....	98
8.2. Anexo B - Documento de Requisitos - Porting do Videojogo “Falcao vs Aliens”.....	110
8.3. Anexo C - Modelo de Mensagens JSON Trocadas com o Servidor - “Falcao vs Aliens”.....	133
8.4. Anexo D - Diagramas de Entidade e Relacionamento - Bases de Dados Original e da Nova Versão - “Falcao vs Aliens”.....	138

1. Introdução

O primeiro videojogo surgiu em Outubro de 1958, criado pelo físico William Higinbotham [1][2]. Desde então, a área dos videojogos tem ganho cada vez mais destaque, sendo atualmente uma das maiores indústrias de entretenimento no mundo [3][4][5]. Consequentemente, o número de videojogos lançados é cada vez maior, mas para usufruir dos mesmos é necessário uma plataforma que os execute, hoje em dia não existe apenas uma capaz de o fazer, mas sim um conjunto variado de opções. No entanto, um videojogo nem sempre é compatível com todas as plataformas, seja devido a limitações de *hardware* das mesmas ou pelo facto do videojogo não ter sido desenvolvido especificamente para as mesmas. Quando se deve apenas a esta última limitação, existe a possibilidade de torná-lo compatível com outras plataformas capazes de o executar, sendo para isso necessário efetuar o *porting* do mesmo.

Este projeto insere-se na área do *porting* de videojogos e, surge da necessidade de fazer com que um videojogo execute em novas plataformas, mais especificamente *Windows*, *Facebook* e navegadores atuais, como por exemplo *Firefox* e *Chrome*. O videojogo mencionado foi desenvolvido pela empresa WingzStudio, denominado por "Falcao vs Aliens", o qual foi inicialmente desenvolvido e lançado para *Android*, *iOS* e *Windows Phone 8*, como uma versão de demonstração (quatro níveis jogáveis). Além do *porting*, é ainda necessário terminar o desenvolvimento da versão original, corrigir eventuais defeitos e adicionar novas funcionalidades.

A WingzStudio Lda, fundada em 2011, tem como principal atividade a criação de videojogos para dispositivos móveis [6]. Encontra-se sediada em Portugal, mais propriamente na cidade de Coimbra. Esta é responsável por este projeto, que tem como principal finalidade atingir novos mercados, principalmente *Facebook*, pois atualmente é a rede social mais popular no mundo [7]. Como o videojogo em causa tem como protagonista o profissional de futebol Radamel Falcao, que conta com milhões de seguidores na sua página no *Facebook* [8], esta é uma oportunidade para atingir esse público.

Atualmente a empresa usa uma ferramenta criada internamente para desenvolver videojogos [9], tendo esta suporte a um número limitado de plataformas (*Mac OS X*, *iOS*, *Android* e *Windows Phone 8*) [9], o que dificulta o processo de *porting* dos seus videojogos para novas plataformas, assim há a necessidade de usar uma ferramenta com maior potencial e com um maior número de plataformas suportadas.

O projeto encontra-se dividido em dois períodos, coincidindo com cada um dos semestres da disciplina. O primeiro período tem como objetivos:

- Estudo do processo de *porting* e de ferramentas que o facilitam, sendo necessária a escolha de uma ferramenta onde será efetuado o processo de *porting*;
- Escolha da metodologia de desenvolvimento.
- Realização de alguns processos de desenvolvimento do videojogo, como a definição do conceito e da arquitetura geral.

Durante o segundo período os objetivos são:

- Análise e levantamento de melhorias do “Falcao vs Aliens”;
- Conclusão dos processos de desenvolvimento.

No final deste estágio é esperado que esta nova versão do videojogo esteja fiel ao conceito original, seja capaz de executar num maior número de plataformas, incluindo integração completa com o *Facebook* e que ofereça um novo conjunto de funcionalidades. Ao atingir essas metas é esperado que o número de jogadores do “Falcao vs Aliens” aumente.

Este documento serve como meio de apresentação de todo o trabalho realizado assim como os resultados obtidos, o primeiro capítulo inicia-se com as motivações, objetivos e abordagem a utilizar. No segundo capítulo é apresentado um estudo sobre o processo de *porting* e uma análise a ferramentas que facilitam esse processo. É também apresentada uma análise comparativa entre algumas ferramentas, resultando na escolha de qual será utilizada para o desenvolvimento. O terceiro capítulo incidirá sobre: os objetivos; a metodologia a ser utilizada durante o desenvolvimento, assim como as razões que levaram à sua escolha; planeamento de todo o trabalho. O capítulo quatro apresenta a primeira parte do processo de desenvolvimento, do videojogo “Falcao vs Aliens”, no qual serão descritos: conceito, requisitos e arquitetura. Posteriormente, no capítulo cinco irá ser detalhada a implementação do videojogo, assim como os testes realizados ao mesmo.

2. Estado da Arte

O principal objetivo deste capítulo é fazer uma introdução à área do *porting* de videojogos, pretendendo explicar em que consiste e abordar alguns temas que a envolvem. Seguidamente serão analisados e comparados dois motores de jogo. Por fim, são apresentadas outras abordagens ao *porting*, assim como algumas ferramentas que facilitam as mesmas. Assim, o capítulo encontra-se dividido em:

- **Porting de Videojogos:** é feita uma introdução ao tema e são abordados vários aspetos fundamentais a ter em conta durante o desenvolvimento de um videojogo e o durante o processo de *porting*.
- **Motores de Jogo:** é feita a análise e comparação de dois motores de jogo, *Unity Personal Edition* e *Cocos2d-JS*.
- **Outras Ferramentas/Abordagens:** análise de outras abordagens e de ferramentas que facilitam as mesmas.

2.1. Porting de Videojogos

Ao longo do tempo foram criadas diversas plataformas para jogar, o que levou a uma divisão dos utilizadores pelas mesmas. Desta forma para aumentar a visibilidade e o número de potenciais utilizadores de um videojogo, é necessário levá-lo a diferentes plataformas. De maneira a atingir esse objetivo é necessário fazer o *porting* do mesmo, este é um processo que consiste em adaptar um videojogo a plataformas para as quais este não foi inicialmente desenvolvido. Por vezes, este processo é encarado de forma incorreta, principalmente pelo facto das equipas/empresas não terem consciência da dificuldade e complexidade do mesmo, aumentando a probabilidade do resultado final ser um fracasso [12]. Os problemas mais comuns são: mau controlo da câmara, falha nos *save files*, gráficos de qualidade inferior à versão original, performance e *bugs* [13].

O elevado número de casos que não tiveram resultados satisfatórios [12], faz com que os jogadores mais exigentes não tenham uma opinião positiva relativamente a versões resultantes deste processo [13], havendo consequências para as produtoras responsáveis pelos fracassos, como a perda de credibilidade e uma reputação negativa. No entanto, existem também casos cujo processo de *porting* foi executado com sucesso, onde a qualidade das versões originais foi superada [10], no geral o resultado mínimo exigido deste processo é um produto ao nível da versão original. Para diminuir os riscos de fracasso, é necessário ter um conhecimento aprofundado acerca

dos diferentes aspetos envolvidos, como: plataformas, linguagens de programação e *software*, como por exemplo motores de jogo e bibliotecas. Por vezes, esses conhecimentos só são alcançáveis com anos de prática, contudo existem inúmeros aspetos fundamentais a ter em conta, os quais serão aprofundados na próxima secção.

O aumento do nível de exigências nesta área, teve como consequência a criação de equipas/empresas especializadas no *porting* de videojogos, um exemplo é a *Aspyr* que há 17 anos que se dedica quase exclusivamente a esta atividade [11]. Além dos motivos mencionados para a existência destas, há ainda o facto dos criadores de videojogos, por vezes, optarem por deixar a tarefa do *porting* para terceiros, dedicando-se unicamente a novos projetos [14].

A decisão de realizar o processo de *porting* deve ser ponderada, principalmente em equipas/empresas pequenas, pois tornar um videojogo compatível com múltiplas plataformas é uma tarefa complexa, visto que normalmente estas possuem *hardware* e *software* (como sistemas operativos) diferentes [13].

Atualmente efetuar este processo de forma rápida e sem planeamento não é suficiente, tal como pensar nas várias versões de um videojogo como simples processos de *porting*, pois é necessário garantir que todas são um bom produto e que tiram proveito das funcionalidades oferecidas por cada plataforma [18]. Deve-se assegurar a qualidade de todas as versões, garantindo o interesse das pessoas, mesmo aquelas que já tenham adquirido o videojogo em outra plataforma [18].

Portanto, para que esse processo não seja tão complexo, há algumas práticas que podem ser seguidas para garantir uma maior facilidade e rapidez num futuro processo de *porting*, as quais podem ser seguidas desde o início do desenvolvimento da versão original do videojogo. Práticas essas que serão abordadas a seguir.

2.1.1 Desenvolvimento do Videojogo

Antes do começo do desenvolvimento, convém existir uma fase de planeamento, onde além da especificação da estrutura do videojogo e dos diversos componentes, também são tomadas diversas decisões como as ferramentas e linguagens de programação a usar [33].

As equipas pequenas normalmente são obrigadas a usar motores de jogo ou bibliotecas pré-concebidas, podendo dedicar-se apenas ao desenvolvimento do videojogo em si, visto que programar um motor de jogo demora bastante tempo [14]. Este tipo de escolha normalmente tem impacto no futuro processo de *porting*, principalmente quando estes são de código fechado e oferecem suporte a poucas

plataformas. Nesses casos, provavelmente será necessário converter certos componentes da versão original para um novo motor de jogo ou bibliotecas, o que aumenta a duração do processo. Assim, quando é necessário escolher as tecnologias a usar deve-se ter especial atenção se estas são de código aberto e se têm uma licença de uso apropriada, caso contrário é necessário garantir que suportam plataformas para as quais no futuro existe a possibilidade de desenvolver [14].

A linguagem de programação a usar também é uma escolha importante, sendo conveniente garantir que esta compile para as plataformas pretendidas e que execute de forma adequada, isto é, que permita obter uma boa performance sem limitações. O uso de linguagens recentes no mercado poderá apresentar alguns riscos, porque mesmo que as funcionalidades sejam muito apelativas, por vezes estas não suportam todas as plataformas desejadas ou apresentam problemas/instabilidades ao executar nas mesmas [14]. Um exemplo é a linguagem de programação *Swift*, lançada em 2014 pela *Apple*, que apesar de ser uma linguagem apelativa para o desenvolvimento de aplicações *iOS*, esta oferece pouco suporte a diferentes plataformas [17].

Contudo, todas as linguagens, ferramentas, bibliotecas ou motores de jogo têm vantagens e desvantagens, devendo-se ter especial atenção às suas limitações antes de se tomar uma decisão sobre quais usar, caso contrário estas poderão vir a constituir problemas graves no desenvolvimento.

Após a fase de planeamento, inicia-se a fase de implementação, nesta fase há vários aspetos que convém ter em atenção de forma a facilitar um futuro processo de *porting*.

É boa prática garantir o uso eficiente dos recursos, pois nem todas as plataformas têm as mesmas capacidades. Apesar deste aspeto não ser uma grande barreira, visto que normalmente é possível resolver-se mais tarde com optimizações de código, convém ter em mente os seguintes pontos:

- Usar os recursos de memória com moderação, visto que no futuro poderá haver a necessidade de levar o videojogo a plataformas que não têm tanta capacidade, portanto quanto mais otimizado estiver mais fácil será processo de *porting*. Um exemplo é a libertação de recursos que não estão a ser necessários no momento.
- Manter sob controlo o uso da unidade central de processamento (*CPU*), ou seja, o processador. Ao desenvolver-se em plataformas de última geração certas limitações que poderão existir acabam por ser esquecidas, é então aconselhável a utilização de uma máquina com características inferiores, assim ao testar-se

na mesma consegue-se ter uma maior percepção de potenciais problemas de desempenho.

- O código deve ser feito a pensar no uso *multithreaded* desde o começo porque, em algumas plataformas, um único processador pode não ser suficiente, existindo a necessidade de dividir em duas ou mais *threads* que corram em processadores diferentes. Portanto, ter o código *multithreaded* poderá reduzir trabalho futuro [14].

Além do uso eficiente dos recursos, há outros aspetos a ter em conta:

- Não usar código específico de determinada plataforma ou sistema operativo, pois estes provavelmente terão que ser alterados [16].
- Deve-se garantir que a camada de apresentação seja flexível, porque existem inúmeras configurações diferentes, como por exemplo a resolução de ecrã. Convém também ter atenção ao tamanho da fonte usada no texto, de forma a garantir que este é legível [16].
- Tendo em conta a variedade de dispositivos é aconselhável oferecer diversas alternativas de *input*, desta forma além de permitir uma maior escolha ao jogador sobre qual controlador usar, também poderá evitar problemas num futuro processo de *porting* [16].
- Apesar da linguagem universal ser inglês nem todas as pessoas a dominam [74][75][76], então é conveniente manter o videojogo capaz de lidar com a adição de novas localizações. De forma a cumprir essa tarefa, devemos ter em atenção os seguintes elementos: texto, áudio de voz e alguma arte com conteúdo específico de determinada localização. Quanto ao texto devemos garantir que se encontra em ficheiros independentes que possam ser chamados através do código, consoante a preferência do jogador. Para o áudio de voz existem duas opções: um sistema de legendas, mantendo o áudio inalterado; a segunda solução é gravar o áudio nas línguas alvo, à semelhança do texto, deve-se também manter em ficheiros separados e serem chamados do código consoante a preferência. Por fim em relação à arte, não é conveniente adicionar texto ou algo específico de determinada localização, porque apesar de não trazer complicações na programação, visto que basta chamar as diferentes imagens, é uma sobrecarga de trabalho para os artistas, pois têm que recriar a mesma arte múltiplas vezes (normalmente uma por localização).

- Ao serem fornecidas várias localizações, existe um problema que é o espaço que o videojogo irá ocupar, pois este irá ter mais ficheiros de som, imagem e texto, normalmente os ficheiros de som e imagem ocupam muito espaço [16]. Uma técnica usada para diminuir este problema é fornecer a opção de escolher quais localizações se pretende instalar, descartando assim ficheiros correspondentes às outras localizações.

Ao atingir a fase final de desenvolvimento, ainda há certas práticas que poderão poupar trabalho a quem reutilizar os componentes do videojogo, seja para efetuar um processo de *porting* ou simplesmente para efetuar melhorias. Essas práticas passam por assegurar uma boa documentação dos diversos aspetos do videojogo, sejam eles técnicos, como por exemplo sobre o código e funcionamento das tecnologias, ou gerais (tema, conceito, intenção...).

Outro aspeto a considerar, é a facilidade de acesso aos vários componentes, sendo aconselhável o uso de um repositório que esteja disponível a qualquer hora. Apesar de parecer simples, por vezes ocorre a perda de certos materiais, principalmente artísticos (imagens, som, entre outros), o que, durante o processo de *porting*, obriga a uma recriação dos mesmos ou ao uso de engenharia inversa.

2.1.2 Processo de *Porting*

Anteriormente foram apresentados alguns aspetos a considerar durante o desenvolvimento de um videojogo, com o intuito de facilitar um futuro *porting*. Seguidamente serão apresentados aspetos a ter em conta durante esse processo.

À semelhança do desenvolvimento original, também no começo deste processo é recomendado passar pela fase de planeamento, onde além da possível necessidade de fazer uma nova arquitetura, é vital assegurar que as plataformas alvo têm todas as condições/capacidades para executar o videojogo com a qualidade e funcionalidades pretendidas. Assim, é necessário um estudo tanto sobre as plataformas como do próprio videojogo e de todas os componentes que o envolvem (código, tecnologias, ...). Neste último caso, quando não existe documentação ou é limitada, o estudo do mesmo torna-se mais complicado. Um aspeto ainda mais problemático é a indisponibilidade de certos componentes, o que obriga à utilização de engenharia inversa ou processos semelhantes, aumentando a complexidade do processo.

Por fim, é também necessário garantir que se possui os recursos suficientes para levar o videojogo a outras plataformas, como: tempo, dinheiro, ferramentas e mão de obra.

Posteriormente à fase de planeamento, temos a implementação, a qual inclui vários passos, cada um tendo os seus próprios desafios:

- Em primeiro deve-se fazer o videojogo executar na plataforma pretendida, ou seja, que este compile e execute mesmo que o faça com limitações ou defeitos. Este passo por vezes é complexo e é onde são encontrados os problemas mais difíceis, maioritariamente devido a dependências de bibliotecas ou *middleware* incompatíveis com a plataforma. Nesse caso, em que foram usadas ferramentas de código fechado que não suportam a plataforma alvo, há a necessidade de refazer partes ou até a totalidade do videojogo, o que dificulta o processo de *porting*.
- O próximo passo é adicionar um suporte adequado à parte gráfica, extraíndo as chamadas gráficas para uma interface comum a todas as plataformas. Videojogos criados a pensar num futuro *porting*, normalmente já têm esta camada separada do código principal, mas há situações em que estas chamadas são intercaladas diretamente no código, sendo necessário extraí-las para uma biblioteca em separado, que por sua vez deve ser implementada para cada *Application Programming Interface (API)* gráfica das plataformas alvo.

Ao concluir os passos anteriores, nem sempre a performance está ao nível desejado, principalmente quando a plataforma alvo não é tão poderosa em recursos quanto a plataforma para a qual o videojogo original foi desenvolvido. Novas plataformas apresentam diferentes desafios, como por exemplo a nível de performance da *CPU* ou da Unidade de Processamento Gráfico (*GPU*), que podem afetar o videojogo em diversas formas que nem sempre são fáceis de prever. Esta é uma das partes mais desafiantes neste processo, visto que é necessário uma boa compreensão do *hardware* subjacente e como este é diferente daquele para o qual a versão original foi concebida. Estes conhecimentos são essenciais de forma a encontrar onde os *bottlenecks* estão e como resolvê-los, garantindo a execução em *hardware* modesto. É nesta fase aquando do uso de ferramentas de código fechado que estas podem causar problemas, porque não existe uma maneira fácil de saber o que está a ser feito internamente, não sendo possível otimizá-las.

Após estar tudo a funcionar com uma performance adequada, é preciso lidar com um conjunto de pequenos problemas que consomem uma quantidade considerável de tempo. Os mais comuns são: a necessidade de alterar os controlos para a plataforma alvo; as interfaces de utilizador terem de ser adaptadas ao tamanho da janela e aos métodos de *input*. Um exemplo é a necessidade de mudar uma interface de utilizador baseada em *touch* para outra que faça uso de um *gamepad*.

Outro problema ocorre quando as plataformas têm diferentes resoluções de ecrã, sendo boa prática testar e rever todos os textos para garantir que estão ajustados ao espaço disponível, assim como se são legíveis. Para além do que já foi referido, cada plataforma tem os seus próprios requisitos de certificação que deverão ser cumpridos, como exemplo temos o caso do *Windows*, em que é obrigatório que as aplicações suportem os mecanismos de segurança do mesmo e, uma instalação limpa e reversível. Este tipo de situações normalmente exigem mudanças nas funcionalidades do videojogo.

Por fim, atinge-se a última fase do processo de *porting*, em que o principal objetivo deverá ser a garantia da qualidade. Uma equipa que serve como exemplo é a *Aspyr*, que dos seus 50 funcionários, 38 dedicam-se ao controlo de qualidade [15].

Existem várias formas de garantir a qualidade, em primeiro lugar deve-se assegurar que o videojogo executa sem nenhum problema, ou seja: performance adequada, inexistência de interrupções abruptas ou de falhas gráficas/sonoras, jogabilidade funcional e *bugs* corrigidos (pelo menos aqueles considerados críticos). De forma a garantir o que foi referido, além dos testes internos, existe ainda a possibilidade de efetuar testes beta, reunindo um grupo de pessoas externas para realizarem essa tarefa. Estes, além de permitirem descobrir novos *bugs* ou outros problemas, permite também conhecer a perspectiva de outras pessoas em relação ao produto. Ao desenvolver-se para uma plataforma, que não é a primária ou da qual não se tem muito conhecimento, podem surgir problemas inesperados, principalmente de usabilidade/jogabilidade. Até porque, por vezes, a potencialidade da plataforma não foi explorada ao máximo e ao realizar-se estes testes, várias ideias/perspectivas diferentes aparecem através destes ou da observação de como as pessoas jogam [19].

Se o principal objetivo do *porting* é fazer com que o resultado seja o mais semelhante ao videojogo original, as duas versões devem ser executadas lado a lado, de forma a garantir que nada ficou esquecido ou desajustado [20]. Por vezes, os responsáveis pelo desenvolvimento assumem que está tudo a funcionar correctamente, mas ao testar lado a lado apercebem-se de que, por exemplo, a física ou a iluminação não têm o mesmo comportamento da versão original. Existem ainda casos mais extremos de conteúdo em falta, como por exemplo níveis, personagens, veículos, armas, entre outros elementos. Retirar pequenos pormenores em prol de algo, sem causar grande impacto na experiência é aceitável, mas quando se trata de uma situação em que até um jogador casual percebe a ausência de conteúdo, torna-se significativo e inaceitável [12].

Para concluir, tal como na fase final do desenvolvimento da versão original, deve-se também assegurar uma boa documentação e fácil acesso aos vários componentes, pois poderá haver necessidade de voltar a usá-los.

2.2. Motores de Jogo

No caso particular deste estágio, o videojogo destinado a passar pelo processo de *porting* foi desenvolvido com um motor de jogo criado pela própria empresa, o qual foi desenvolvido em C++, não suportando navegadores e consequentemente *Facebook* (apenas integração externa). Assim, há a necessidade de converter este videojogo para outro motor de jogo, de forma a escolher o mais indicado foi feita uma pré-seleção de ferramentas que cumprem com as exigências da empresa:

- compatibilidade com *iOS*, *Android*, *Windows*, *Mac OS*, navegadores atuais e *Facebook* (integração interna, ou seja, *Facebook App*);
- mínimo custo monetário possível;
- suporte 2D, física e áudio.

Após esta pré-seleção, apenas sobraram dois motores que cumprem com todas as exigências mencionadas, *Unity Personal Edition 5.1.2f1*¹ e *Cocos2d-x 3.7*², os quais serão analisados e comparados de seguida.

2.2.1. Unity Personal Edition 5.1.2f1

O *Unity* é um motor de jogo criado em 2004 pela *Unity Technologies*, com o intuito inicial de ser apenas uma ferramenta de desenvolvimento para o videojogo *GooBall*, criado pela própria empresa [26]. Mas em 2005, este motor foi lançado para o público, sendo inicialmente necessária a compra de uma licença, posteriormente passou a estar disponível uma versão gratuita, tendo esta algumas limitações em relação à versão paga [26]. Nesta análise, decidiu-se optar pela versão gratuita visto que a *WingzStudio*, de momento, não pretende investir dinheiro em ferramentas.

Atualmente é um dos motores de jogo mais conhecidos e utilizados para o desenvolvimento de videojogos no mundo [21][22][23][24], por exemplo, uma pesquisa realizada em 2012 pela revista de videojogos "*Game Developer*" afirma que 53,1% das equipas de desenvolvimento para dispositivos móveis usavam *Unity* para criar os seus videojogos, este é também referenciado como um dos melhores [25][26].

¹ <https://unity3d.com/pt/unity/personal-edition>

² <http://cocos2d-x.org/products#cocos2d-x>

O motor suporta o uso de três linguagens de programação: *C#*, *Boo* e *JavaScript*. Esta última acaba por ser mencionada como *UnityScript*, visto que não é *JavaScript* puro e sim uma adaptação da mesma às necessidades do *Unity*, logo existem inúmeras diferenças entre elas [27]. Assim, existe a necessidade do utilizador ter algum conhecimento básico em alguma destas linguagens de forma a poder iniciar rapidamente o desenvolvimento.

2.2.1.1. Unity Editor

O objetivo principal deste motor é facilitar o fluxo de trabalho no desenvolvimento de videojogos, sendo o *Unity Editor* (Figura 1) um dos pontos fortes que demonstram exatamente esse objetivo. O editor já vem integrado com o motor, sendo ele bastante visual, oferecendo funcionalidades como arrastar e alterar objetos, o que permite uma rápida construção e modificação de projetos. Além destes, permite também executar o projeto através do editor, sendo possível ajustar as dimensões da janela, simulando vários dispositivos diferentes, facilitando o teste em diferentes resoluções [41]. Este editor permite também a quem não seja programador, construir e testar os níveis com a interface gráfica, sendo uma mais valia em equipas que contêm artistas e/ou *designers* [40].

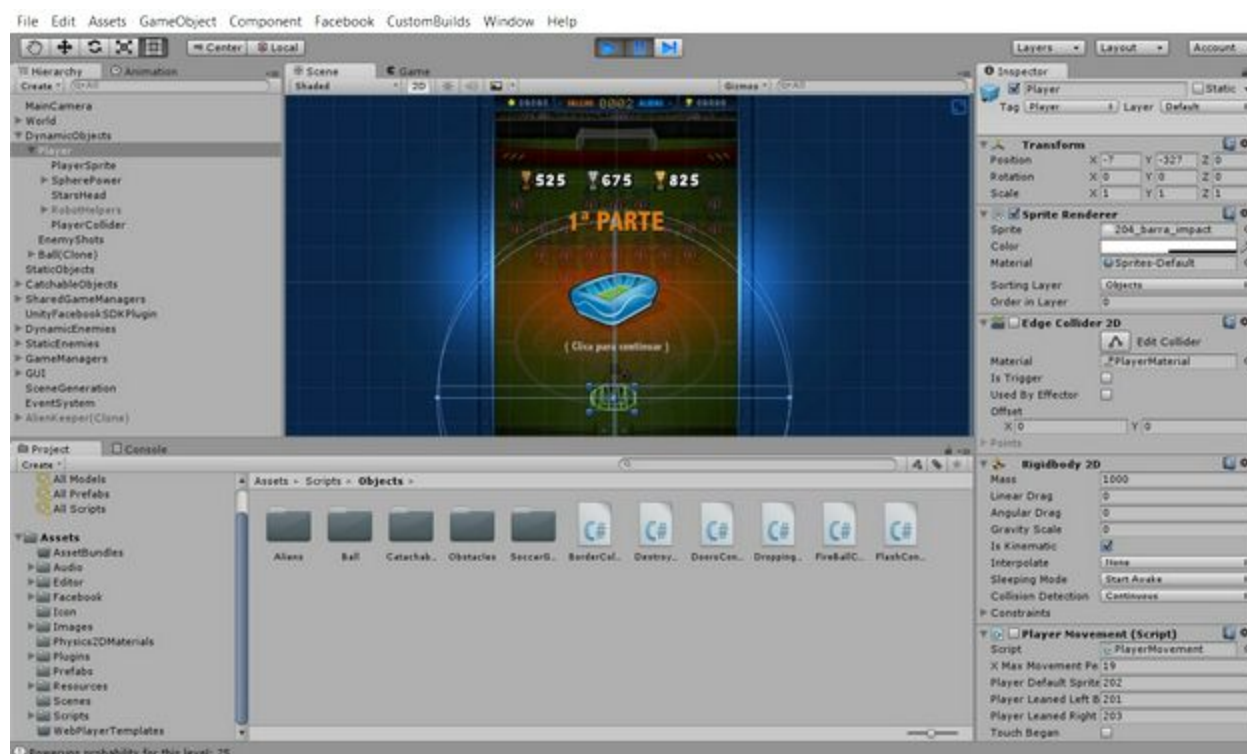


Figura 1 - Unity Editor

Uma particularidade muito interessante do *Unity*, é o facto de permitir efetuar alterações aos objetos e cenário, enquanto o videojogo é executado no editor, o que acelera o processo de desenvolvimento.

Por fim, é ainda possível aumentar as capacidades do editor consoante as necessidades, visto que é possível integrar ferramentas ou *plugins* [26], além do facto de que é também possível alterar o código fonte da própria interface do editor [28].

2.2.1.2. Renderização

O *Unity Personal Edition* faz uso de diferentes *API's* gráficas para efetuar a renderização consoante a plataforma alvo, este usa:

- *OpenGL ES* 1.1, 2.0 e 3.0 para os dispositivos móveis [43][44];
- *OpenGL* para *Windows*, *Mac OS X* e *Linux* [47];
- *Direct3D* 9 e 11 para *Windows* [45][46].

2.2.1.3. Scripting

Mesmo tendo uma forte componente visual, o *Unity* não está limitado a esta, oferecendo também a possibilidade de elaborar *scripts* complexos, com o auxílio das três linguagens de programação mencionadas anteriormente. É possível usá-las em conjunto no mesmo projeto, podendo assim tirar o máximo proveito de cada uma delas.

O *Unity* traz consigo um *Integrated Development Environment (IDE)* com o nome *MonoDevelop*³, que conta com as capacidades necessárias para tratar de toda a parte de programação incluindo *debugging* [29]. Caso o utilizador não esteja satisfeito com este *IDE*, pode sempre integrar outros *IDE's* externos com *Unity* [26].

2.2.1.4. Suporte a Plataformas

Provavelmente será difícil encontrar um motor de jogo que tenha suporte a tantas plataformas quanto o *Unity*. Este suporta as seguintes: *Windows*, *Linux*, *Mac OS X*, *iOS*, *Android*, *Windows Phone 8*, *BlackBerry 10*, *Tizen*, *Windows Store Apps*, navegadores e *Facebook*, estes dois últimos através do uso do *plugin Unity Web Player*⁴ [30] ou através de *WebGL*. O *Unity Web Player* permite visualizar conteúdo 3D criado com o *Unity* diretamente nos navegadores [31], este no caso das aplicações *Facebook* encontra-se assinado pelo próprio *Facebook*, permitindo assim que os utilizadores

³ <http://www.monodevelop.com>

⁴ <https://unity3d.com/pt/webplayer>

tenham confiança no *plugin* e o aceitem instalar. No caso do *WebGL*, este executa em qualquer navegador que ofereça suporte à tecnologia, não sendo necessária a instalação de *plugins*, sendo uma excelente alternativa ao *Unity Web Player*.

O *Unity* oferece ainda suporte a consolas, mas para poder compilar um projeto para estas é necessário comprar licenças e ser aprovado pelo detentor dos direitos da consola em questão [32]. É notável o facto de que para compilar para estas, basta apenas seleccionar a plataforma desejada através de um menu (Figura 2), que o *Unity* encarrega-se de finalizar a tarefa, não sendo necessário efetuar alterações ao projeto [26].



Figura 2 - Menu de Seleção da Plataforma a Compilar (*Unity Editor*)

2.2.1.5. Documentação, Suporte e Tempo de Aprendizagem

O *Unity* conta com uma comunidade bastante ativa e com um suporte relativamente rápido, os fóruns são um bom local para colocar questões, receber orientação, discutir as melhores práticas, entre outros assuntos. O serviço *Unity Answers*⁵ é outra fonte que facilita a aprendizagem, oferecendo recompensas às boas perguntas/respostas, tendo uma abordagem mais social do que os tradicionais *websites* com este propósito. Fora estes, existe ainda uma abundante documentação para o *Unity* disponível na *Internet*, como tutoriais em vídeo ou texto de boa qualidade.

⁵ <http://answers.unity3d.com>

Dentro do próprio *Unity* é possível acessar à *Unity Asset Store*⁶ (Figura 3), que fornece os meios necessários para a partilha e venda de conteúdo feito pela comunidade. Isto permite aos utilizadores adquirirem *assets*, poupando o tempo que necessário para os fazerem, acelerando o desenvolvimento [26].

Todos estes aspetos, resultam numa diminuição do tempo de aprendizagem, sendo esta mais rápida do que na maioria dos motores de jogo.

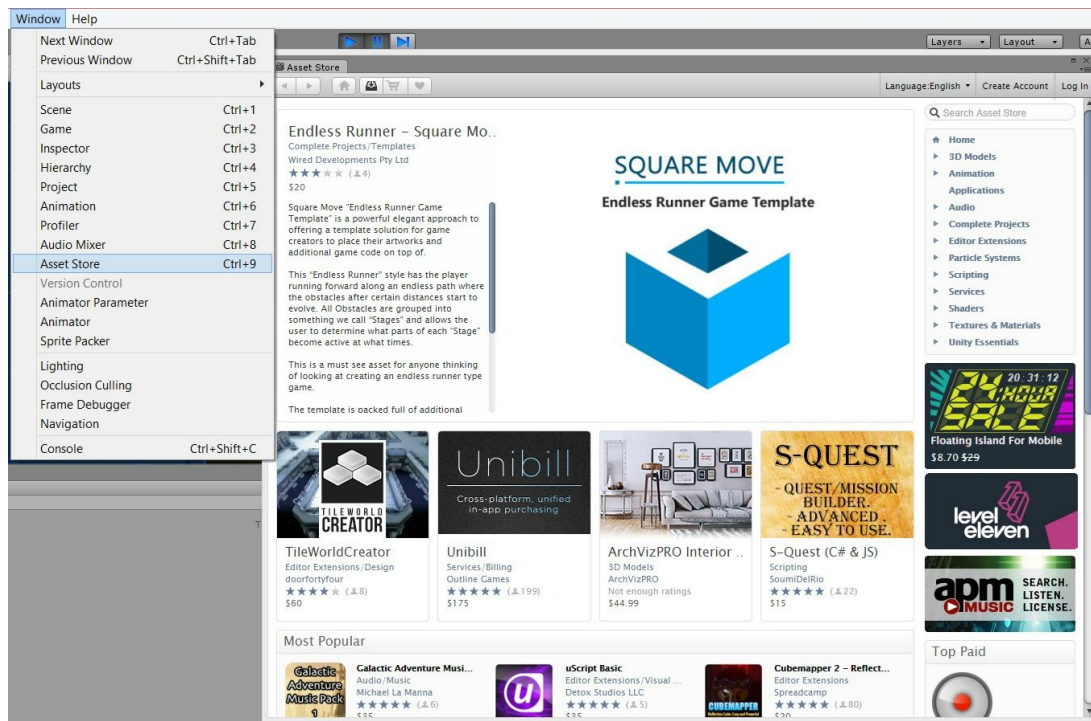


Figura 3 - *Unity Asset Store (Unity Editor)*

2.2.1.6. Open Source

O *Unity* já conta com alguns componentes *open source*, mais especificamente as ferramentas de testes e o sistema de interface gráfica do utilizador [42]. Contudo, a maioria dos componentes não têm o código disponibilizado, podendo ser problemático no caso de existir a necessidade de melhorar ou modificar internamente o motor. No entanto, prevê-se que nas futuras versões os criadores irão ter mais atenção a esta questão, fornecendo o código fonte de mais componentes [42].

⁶ <https://www.assetstore.unity3d.com/en/>

2.2.1.7. Preços e Licenças

Como já foi referido anteriormente, o *Unity* conta com duas versões, *Unity Personal Edition* que é gratuita e *Unity Professional Edition* que tem custos associados [34]. O *Unity Personal Edition* ao contrário da maioria das ferramentas, mesmo sendo uma versão gratuita conta com a maioria das ferramentas da versão paga, sendo elas suficientes para criar um produto com fins comerciais. A versão *Professional Edition* conta com alguns extra, como *Unity Analytics Pro*⁷, *Unity Cloud Build Pro*⁸ e *splash screen* customizável [35]. Uma desvantagem da versão gratuita, é o facto de não permitir personalizar a *splash screen* inicial, a qual por padrão apresenta o logotipo do *Unity*.

Há ainda um detalhe a considerar, que é a obrigatoriedade de comprar a *Professional Edition* caso as receitas sejam acima de 100.000,00 dólares no último ano fiscal [36].

2.2.1.8. Capacidades de Desenvolvimento

O *Unity* é um motor muito poderoso e versátil, este permite desenvolver os mais variados estilos de videojogos nas várias dimensões: 2D, 2.5D e 3D [39]. Atualmente conta com uma base de utilizadores bastante grande, desde as *indie* até às grandes empresas de videojogos, como por exemplo, a *Ubisoft* (*Assassin's Creed Identity* para *iOS*) e *Square Enix* (*Hitman Go* para *Android* e *iOS*) [37][38]. Contudo, não só estas empresas usam este motor, existem outras dedicadas a diferentes áreas que o usam para investigação, simulação e demonstração [26]. Outro aspeto que merece ser referido é o facto de oferecer a possibilidade de integrar ferramentas de edição gráfica e de animação, como por exemplo: *Photoshop*, *Blender* e *Maya*. Assim, é possível importar animações, imagens, cenários, entre outros componentes, diretamente das várias ferramentas gráficas. O *Unity* também oferece suporte a física, tanto 2D como 3D [65], no primeiro caso é orientado pelo *Box2D*, enquanto que em relação ao 3D usa *NVIDIA PhysX*. Este motor mostra-se uma ferramenta muito completa, capaz de gerir praticamente todos, senão todos, os aspetos que envolvem o desenvolvimento de um videojogo [40].

⁷ <https://unity3d.com/pt/services/analytics>

⁸ <https://build.cloud.unity3d.com/landing/>

2.2.1.9. Conclusão

O motor de jogo *Unity* é uma ferramenta extremamente útil no desenvolvimento de projetos. Funcionalidades como o editor visual, *scripting* completo e robusto, o sistema e suporte a multi-plataformas tornam este motor um dos melhores da atualidade. Além destes, o apoio fornecido pela comunidade e a *Unity Asset Store* elevam-no ainda mais [26]. Todos estes aspetos resultam numa diminuição do tempo necessário para o desenvolvimento de projetos.

2.2.2. Cocos2d-x 3.7

O *Cocos2d* é um motor de jogo 2D *open source* e gratuito [48]. Foi inicialmente criado em 2008 pelo Ricardo Quesada [49], juntamente com outros programadores. O motor é usado para o desenvolvimento de: videojogos, aplicações e outro *software* interativo. Este oferece suporte a várias plataformas como: *Windows*, *Linux* e *Mac* [50].

Entretanto o motor sofreu imensas adaptações, existindo actualmente várias versões, por exemplo: *Cocos2d* (versão original), *Cocos2D-swift* (também conhecida por *Cocos2d-iPhone*), *Cocos2d-x* e *Cocos2d-XNA* [50][51]. Estas versões são bastante populares e utilizadas para o desenvolvimento de videojogos, principalmente para dispositivos móveis [22].

Tendo em conta as necessidades da WingzStudio, decidiu-se analisar apenas a versão *Cocos2d-x*, pois suporta um maior número de plataformas, principalmente integração completa com o *Facebook* [53]. Esta é desenvolvida pela *Chukong Technologies*, tendo sido lançada a primeira versão em 2010 [66], apesar da equipa de desenvolvimento não ser a mesma do *Cocos2d* original, esta versão continua a ser totalmente *open source* e gratuita [52].

O *Cocos2d-x* oferece suporte a três linguagens de programação: *C++*, *Lua* e *JavaScript* (adicionado recentemente com a integração do *Cocos2d-JS* [67]). Este permite desenvolver aplicações em 2D, 2.5D e 3D [55][56][68], oferecendo suporte às seguintes plataformas: *Mac OS X*, *Windows*, *Linux*, *iOS*, *Android*, *Blackberry 10*, *Windows Phone 8*, *Tizen*, *Facebook* e navegadores que suportem *HTML5* [52][56][69], como por exemplo, *Firefox* e *Chrome*. No entanto, só é possível compilar para todas as plataformas se a linguagem escolhida tiver sido *JavaScript*, caso contrário navegadores deixam de ser suportados. Contudo, mesmo programando em *JavaScript*, nas plataformas nativas este é ligado a objetos *C++*, garantindo assim uma maior performance [56]. No caso das plataformas *web*, estas executam o código *JavaScript* e usam *Canvas* ou *WebGL* para a renderização, não existindo a necessidade de instalar

outras aplicações [56]. Uma vantagem do código ser desenvolvido com o uso de uma linguagem de *script* é que, como este não é compilado, pode ser modificado em *run-time* o que permite testar de forma mais rápida [64].

Como as exigências deste projeto incluem suporte a navegadores, nesta análise será dado maior destaque à vertente *JavaScript*.

2.2.2.1. Renderização

Este motor de jogo faz uso de diferentes *API's* gráficas para efetuar a renderização consoante a plataforma alvo, este usa:

- *OpenGL ES 2.0* para os dispositivos móveis [52];
- *OpenGL 2.1* para *Windows*, *Mac OS X* e *Linux* [52];
- *WebGL* com *auto canvas fallback* para *web* [52].

2.2.2.2. Performance

Com o *Cocos2D-x* é possível criar aplicações com uma performance muito boa [60], desde que os programadores tenham um conhecimento adequado e mantenham o código otimizado, é possível atingir *60 Frames Per Second (FPS)* mesmo em dispositivos antigos [60].

2.2.2.3. Open Source

Um dos aspetos importantes deste motor é o facto de ser totalmente *open source* [52][56], trazendo várias vantagens. A principal vantagem é o facto dos programadores terem acesso ao código, o que permite: descobrir e corrigir eventuais *bugs* [55][56], efetuar melhorias em certos processos [55][56], integrar bibliotecas externas [40], ou simplesmente modificar alguns componentes de forma a adaptar o motor às nossas necessidades [56].

Existe também um vasto número de extensões, ferramentas e outro código *open source* disponíveis [62], o que permite aumentar as capacidades originais do motor.

2.2.2.4. Editor Visual

O *Cocos2d* não possui um editor visual integrado por padrão, assim o *design* de janelas e interfaces é feito através de programação [57], ou seja, os programadores serão também responsáveis pelo *design*, como o posicionamento dos objetos, entre outros componentes artísticos [40]. No entanto, é possível o uso de editores gráficos

externos, permitindo que os *designers* tenham um papel mais participativo, como é o caso do *Cocos Studio*.

O *Cocos Studio* (Figura 4) é o editor oficial, também gratuito, este tem como principais funcionalidades: editar interfaces, editar cenários e editar animações [58]. Apesar deste facilitar o desenvolvimento, o facto de não ser integrado de raiz apresenta algumas limitações quando comparado a outros editores integrados, como o caso do *Unity Editor*, que permite uma fácil inserção de objetos, *scripts*, física, efeitos de luz, entre outros [57]. Outro aspeto negativo do *Cocos Studio*, é o facto deste apresentar incompatibilidades com alguns computadores [57].

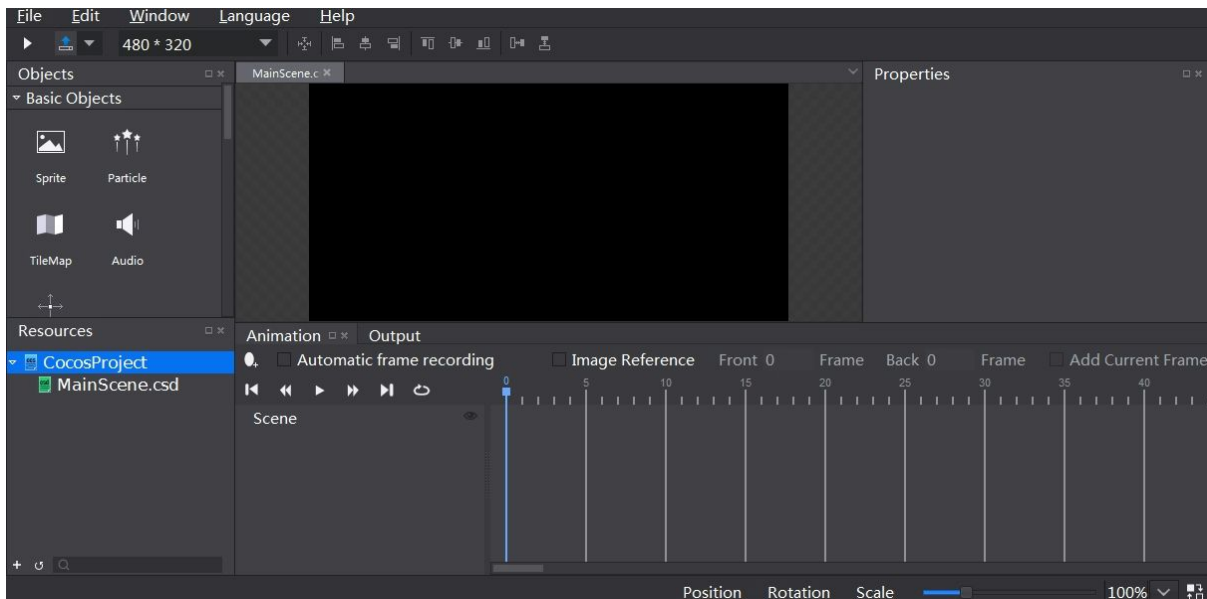


Figura 4 - Cocos Studio

2.2.2.5. Documentação, Suporte e Tempo de Aprendizagem

No geral, tal como em outras ferramentas *open source*, a documentação sobre o *Cocos2d-x* é um dos principais pontos negativos [55][56], principalmente em relação à vertente *JavaScript*, pois esta é escassa e de qualidade inferior [57]. A maioria encontra-se em Inglês, ainda assim alguma documentação está disponível apenas em Chinês [57][41], o que pode dificultar a aprendizagem. Além destas ainda há outras adversidades, como o facto do suporte ser quase inexistente e por vezes muitas das questões colocadas no fórum oficial [59] não obtêm resposta [57]. Contudo, espera-se que sejam publicados novos livros sobre esta família de motores [41], o que provavelmente irá combater um pouco estas lacunas, no entanto é provável que estes não sejam gratuitos, podendo não ser acessíveis a todas as empresas/equipas.

Tendo em conta estes problemas, o tempo de aprendizagem necessário para começar a desenvolver neste motor poderá ser longo, visto que é necessário possuir conhecimentos básicos acerca das linguagens e bibliotecas suportadas pelo mesmo [57], além do facto de que a falta de documentação e suporte, poderão tornar frustrante a aquisição desses conhecimentos.

2.2.2.6. Comunidade e Extras

Uma das lacunas da família *Cocos2d*, é a falta de uma loja integrada [57], como por exemplo, a mencionada na secção anterior (*Unity Asset Store*), que permite à comunidade partilhar diversos conteúdos, acelerando o tempo de desenvolvimento.

Outra lacuna que merece destaque, é a não existência de uma forma fácil de integrar ferramentas que permitam gerir gráficos e animações, tendo estes que ser criados externamente, adicionados ao projeto e feita a referência através do código [40].

2.2.2.7. Capacidades de Desenvolvimento

O *Cocos2d-x*, como já foi referido, permite desenvolver aplicações 2D, 2.5D e 3D [55][56][68], sendo este bastante versátil, o que permite desenvolver uma variedade de estilos [61]. Tal como o *Unity*, este também foi/é usado por profissionais, desde pequenas equipas a grandes empresas, como por exemplo a *Konami* (*Contra: Evolution* para *iOS* e *Android*) e a *Zynga* (*Matching with Friends*) [63][64][77].

Uma das funcionalidades interessantes deste motor, é o facto de poder criar todo o cenário de um videojogo 2D através da leitura de um ficheiro de *Tile Maps* [52], em que este contém toda a informação necessária, como é o caso das texturas a utilizar e a localização dos objetos no cenário [9].

Neste motor é também possível implementar física, tanto 2D [9] como 3D [67]. No caso de físicas 2D, conta com dois motores integrados: *Box2d* e *Chipmunk* [70]. Quanto a físicas 3D, este passou a oferecer suporte às mesmas na versão 3.7, com a integração da *Bullet Physics Library*⁹ [67].

2.2.2.8. Conclusão

O *Cocos2d-x* apresenta-se como uma ferramenta viável no desenvolvimento de projetos. Além do leque de funcionalidades que este oferece e do número de plataformas suportadas, o facto de ser *open source* torna o potencial do mesmo quase

⁹ <http://bulletphysics.org/wordpress/>

ilimitado. No entanto, a documentação e o apoio da comunidade não são o ideal, podendo resultar num aumento do tempo de desenvolvimento.

2.2.3. Comparação dos Motores de Jogo

Após uma análise individual dos dois motores, conseguiu-se identificar alguns dos pontos fortes e fracos de cada um deles. Contudo, tal como foi referido anteriormente, a escolha destes deveu-se ao facto de serem os únicos que cumpriam com as exigências da empresa:

- compatibilidade com *iOS*, *Android*, *Windows*, *Mac OS*, navegadores atuais e *Facebook* (integração interna, ou seja, *Facebook App*);
- mínimo custo monetário possível;
- suporte 2D, física 2D e áudio.

Assim, esta comparação terá como principal foco essas mesmas exigências, mas irá também abordar outros aspetos importantes, como por exemplo disponibilização do código fonte e documentação.

2.2.3.1. Suporte a Plataformas

Ambos os motores oferecem suporte a um diverso número de plataformas, apesar da versão paga do *Unity* suportar algumas consolas, isto não acontece na versão gratuita, logo em comparação com o *Cocos2d-x*, estes estão em igualdade. No entanto, a maioria das aplicações desenvolvidas com o *Unity* necessitam de recorrer a um *plugin*, *Unity Web Player*, para conseguirem executar em certas plataformas, como é o caso dos navegadores e *Facebook*. Contudo recentemente foi adicionado ao *Unity* o suporte a *WebGL*, sendo possível compilar para navegadores que suportem essa tecnologia, sem a necessidade de recorrer a *plugins*. No caso do *Cocos2d-x* não há necessidade da utilização de qualquer *plugin*, então neste aspeto considera-se que este motor tem uma ligeira vantagem em relação ao *Unity Personal Edition*.

2.2.3.2. Preços e Licenças

As opções analisadas são gratuitas, podendo ser usadas para desenvolvimento e comercialização de aplicações. Contudo, convém referir que existe uma restrição no caso do *Unity Personal Edition*, que é a impossibilidade de usar esta versão caso se tenha obtido no último ano fiscal um volume de negócios superior a 100.000,00 dólares. Tendo em conta que no *Cocos2d-x* não existe esta restrição, este tem novamente vantagem quando comparado ao *Unity Personal Edition*.

2.2.3.3. Suporte 2D

Os dois motores de jogo têm total suporte ao 2D e oferecem todas as funcionalidades necessárias para o projeto, desta forma ambos estão em posição de igualdade.

2.2.3.4. Física 2D

As opções analisadas suportam física 2D, mais especificamente recorrendo a bibliotecas integradas, o *Unity Personal Edition* usa apenas *Box2D*, enquanto que o *Cocos2d-x* usa *Box2D* e *Chipmunk*. Como ambas as soluções são eficazes na implementação de física 2D, considera-se que nenhuma é superior.

2.2.3.5. Áudio

Este é o último aspeto exigido pela WingzStudio, ambos os motores analisados suportam áudio, no caso do *Unity Personal Edition* este recorre à biblioteca *FMOD*¹⁰, enquanto que o *Cocos2d-x* conta com duas bibliotecas: *WebAudio*¹¹ para plataformas *web* e *CocosDenshion* para as restantes. Quanto a formatos de áudio suportados, o *Unity Personal Edition* suporta *MPEG(1/2/3)*, *Ogg Vorbis*, *WAV*, *AIFF*, *MOD*, *IT*, *S3M* e *XM*; o *Cocos2d-x* suporta *Ogg Vorbis*, *WAV*, *MP3 (MPEG1/2 Audio Layer 3)*, *CAF* e *MIDI*. Contudo, há uma limitação relativamente a estes formatos no *Cocos2d-x*, pois estão restringidos ao facto da plataforma alvo os suportar ou não. No *Unity Personal Edition*, caso a plataforma alvo não suporte determinado formato, este é convertido de forma automática. Assim, neste aspeto o *Unity Personal Edition* tem vantagem em relação ao *Cocos2d-x*.

Após a comparação entre os motores, com base nos aspetos exigidos pela empresa, pode-se concluir que o *Cocos2d-x* tem vantagem em relação ao *Unity Personal Edition*. No entanto, esta vantagem não é muito significativa, não sendo suficiente para afirmar qual destes motores é melhor. Portanto, seguidamente é apresentada a comparação tendo em conta outros aspetos.

2.2.3.6. Editor Visual

O *Unity* conta com um editor completamente integrado, o que fornece um número elevado de funcionalidades, as quais foram referidas na análise individual deste motor, que facilitam o fluxo de trabalho (Secção 2.2.1.1). Por outro lado, o

¹⁰ <http://www.fmod.org>

¹¹ <https://webaudio.github.io/web-audio-api/>

Cocos2d-x não possui um editor integrado por padrão, o que dificulta o processo de *design*. Contudo, é possível o uso de editores externos, entre as opções disponíveis temos o *Cocos Studio*, que é o editor oficial, o qual apresenta limitações quando comparado com, por exemplo, o editor do *Unity*. Neste aspecto o *Unity* tem uma clara vantagem quando comparado com o *Cocos2d-x*.

2.2.3.7. Linguagens de Programação

O *Cocos2d-x*, oferece suporte a três linguagens de programação: *C++*, *Lua* e *JavaScript*. No entanto, somente com o uso de *JavaScript* é possível compilar para navegadores. No caso do *Unity Personal Edition*, este suporta: *C#*, *Boo* e *UnityScript* (baseada em *JavaScript*). É possível usar estas três linguagens no mesmo projeto, podendo tirar o máximo proveito de cada uma delas e independentemente da linguagem é possível compilar para todas as plataformas suportadas pelo *Unity*. Assim, como o *Unity Personal Edition* oferece uma maior versatilidade, este apresenta vantagem em relação ao *Cocos2d-x*.

2.2.3.8. Open Source

O *Cocos2d-x* é totalmente *open source*, enquanto que o *Unity* conta apenas com alguns componentes *open source*, mais especificamente as ferramentas de teste e o sistema de interface gráfica do utilizador. Contudo, nas futuras versões do *Unity*, prevê-se que será disponibilizado código fonte de mais componentes. Portanto, neste aspecto o *Cocos2d-x* é superior em relação ao *Unity*.

2.2.3.9. Documentação e Suporte

O *Unity* tem um suporte relativamente rápido e conciso, este conta com imensa documentação disponível, desde video tutoriais a textos de boa qualidade, incluindo em diferentes idiomas. Existe também o serviço *Unity Answers*, que facilita a aprendizagem, oferecendo premiações às boas perguntas e respostas, tendo uma abordagem mais social do que o típico *website* de perguntas e respostas.

Por outro lado, a documentação sobre *Cocos2d-x* é escassa, principalmente em relação à vertente *JavaScript*, esta encontra-se normalmente apenas disponível em Inglês ou Chinês. O suporte é quase inexistente, havendo imensas questões no fórum oficial sem resposta.

Tendo em conta estes aspetos, o *Unity* mostra-se superior ao *Cocos2d-x* em documentação e suporte.

2.2.3.10. Loja Integrada

Uma das lacunas da família *Cocos2d*, é a falta de uma loja integrada, como a *Unity Asset Store* no caso do *Unity*. Esta permite à comunidade compartilhar um variado leque de conteúdo, que pode acelerar o tempo de desenvolvimento. O *Unity* revela-se também superior neste ponto em relação ao *Cocos2d-x*.

Seguidamente é apresentada uma tabela (Tabela 1), onde são comparados vários aspetos entre estes dois motores de jogo:

	Unity Personal Edition 5.1.2f1	Cocos2d-x 3.7
Suporte ao 3D	Sim	Sim
Suporte ao 2D	Sim	Sim
Custo	Gratuito *1	Gratuito
Open Source	Alguns Componentes (licença MIT/X11) *2	Sim (licença MIT/X11)
Editor Visual Integrado	Sim (Unity Editor)	Não
Store Integrada	Sim (Unity Asset Store)	Não
Plataformas Suportadas:		
Android	Sim	Sim
iOS	Sim	Sim
Windows Phone 8	Sim	Sim
BlackBerry 10	Sim	Sim
Tizen	Sim	Sim
Windows	Sim	Sim
Linux	Sim	Sim
Mac OS X	Sim	Sim
Navegadores Modernos (HTML5)	Sim *3	Sim *4
Facebook	Sim *3	Sim *4
Áudio	Sim (FMOD)	Sim (CocosDenshion e WebAudio)
Formatos de Áudio Suportados	MPEG(1/2/3), Ogg Vorbis, WAV, AIFF, MOD, IT, S3M, XM	Ogg Vorbis, WAV, MP3 (MPEG1/2 Audio Layer 3), CAF, MIDI *5
Física 2D	Sim (Box2D)	Sim (Box2D e Chipmunk)
Física 3D	Sim (NVIDIA PhysX)	Sim (Bullet Physics Library)
Rendering	OpenGL ES 3.0 (dispositivos móveis), OpenGL ES 2.0 (dispositivos móveis), OpenGL ES 1.1 (dispositivos móveis), OpenGL (Windows, Mac OS X, Linux), Direct3D 9 (Windows), Direct3D 11 (Windows)	OpenGL ES 2.0 (dispositivos móveis), OpenGL 2.1 (desktop), WebGL com auto canvas fallback (web)
Linguagens de Programação	C#, UnityScript (inspirado em JavaScript) e Boo (inspirado em Python)	C++, Lua e JavaScript
Sistema de Partículas	Sim	Sim
Sistema de Interface de Utilizador	Sim	Sim

*1 - Somente para empresas que no último ano fiscal obtiveram um volume de negócios inferior a 100.000,00 USD.
 *2 - Test Tools e Graphical User Interface system.
 *3 - Através do uso do plugin Unity Web Player ou de WebGL.
 *4 - Somente se a linguagem usada for JavaScript.
 *5 - No Cocos2d-x estes formatos de áudio dependem se as plataformas os suportam.

Tabela 1 - Comparação entre Motores de Jogo - Unity Personal Edition 5.1.2f vs Cocos2d-x 3.7
Legenda: os aspetos marcados a negrito representam os que são exigidos pela WingzStudio.

Após a comparação entre os dois motores, conclui-se que o *Unity Personal Edition* é a melhor opção para este projeto. O facto deste fornecer toda a tecnologia

necessária para a execução do *porting* do videojogo mencionado na Secção 1 (Introdução), aliado ao facto de oferecer toda a documentação necessária e apoio da comunidade, fazem com que o tempo necessário de aprendizagem seja menor. Assim, *Unity Personal Edition* revela-se como uma escolha mais segura.

2.3. Outras Ferramentas/Abordagens

No caso particular deste projeto existem outras abordagens válidas ao *porting*, como reescrever o motor original e o videojogo para *JavaScript*, o que permitiria executar nas plataformas exigidas. De forma a atingir esse propósito, podem ser utilizados compiladores de C++ para *JavaScript*, o que diminui o tempo necessário para o fazer em comparação à conversão manual.

Atualmente os compiladores de C++ para *JavaScript* mais populares são: *Emscripten*¹² e *Cheerp*¹³. O primeiro é um projeto *open source* baseado no *LLVM Project*¹⁴, uma infraestrutura de compilador concebida em C++, este compila qualquer linguagem em *JavaScript*, desde que seja possível traduzí-las em *Low Level Virtual Machine (LLVM) bitcode*, os principais exemplo são as linguagens C e C++ [71]. O código *JavaScript* que resulta da compilação, em alguns casos, é capaz de executar praticamente à mesma velocidade que as versões nativas [71]. Um dos pontos fortes é a quantidade de plataformas em que é possível executar o código, como navegadores, dispositivos móveis e PC's. Existem inúmeros projetos que usaram este compilador, como a conversão do *Unreal Engine 3* e do *Bullet Physics Engine* [73].

Cheerp à semelhança do *Emscripten* também é *open source* e baseado no *LLVM Project*, este é compatível com quase todo o código C ou C++(11), sendo capaz de o compilar em *JavaScript*, resultando código otimizado compatível com qualquer navegador em qualquer plataforma [72]. A grande diferença deste compilador para o anterior é que o *Cheerp* oferece a possibilidade de alocar memória dinamicamente, ao contrário do *Emscripten* que obriga a definir previamente um valor máximo de memória. Por outro lado, esta funcionalidade do *Cheerp* obriga a um esforço maior por parte dos programadores, pois exige mais alterações ao código original, no entanto os criadores afirmam que a performance obtida é superior [72].

Apesar de ambos os compiladores serem potenciais alternativas para este projeto de *porting*, a WingzStudio tem preferência pela abordagem de se usar um motor de jogo com suporte a multiplas plataformas.

¹² <https://kripken.github.io/emscripten-site/>

¹³ <http://leaningtech.com/cheerp/>

¹⁴ <http://llvm.org>

3. Objetivos e Método de Abordagem

Este capítulo está dividido em três subsecções: objetivos, abordagem de desenvolvimento e planeamento de atividades. Na primeira subsecção são apresentados os objetivos deste projeto, na subsecção seguinte é descrita a abordagem de desenvolvimento escolhida e por último é apresentado o planeamento elaborado para o projeto.

3.1. Objetivos

Os principais objetivos deste projeto são o *porting* do “Falcao vs Aliens” para novas plataformas (como *Windows*, *Facebook* e navegadores atuais) e implementação de melhorias no mesmo. No entanto, o projeto inicialmente tinha outros objetivos, sendo eles o *porting* de dois videojogos, “Falcao vs Aliens” e “StringZ HD”, porém após uma análise concluiu-se que seria mais enriquecedor para o projeto, se além do exercício de *porting*, também fossem implementadas melhorias tornando o videojogo mais atrativo aos jogadores.

O videojogo mencionado foi desenvolvido pela empresa WingzStudio, com um motor de jogo produzido internamente, o qual tem suporte a um número limitado de plataformas. Assim, existe a necessidade de desenvolver num novo motor, o qual foi escolhido no último capítulo, o *Unity Personal Edition*. Após o desenvolvimento, espera-se que o videojogo seja capaz de executar tanto nas plataformas originais (*Android*, *iOS* e *Windows Phone 8*), como em novas plataformas como: *Windows*, *Facebook* e navegadores atuais.

Tendo em conta a necessidade de desenvolver o “Falcao vs Aliens” de raiz num novo motor de jogo, surgiu a oportunidade de adicionar novas funcionalidades e melhorar as existentes na versão original. Desta forma, efetuou-se uma análise ao videojogo com o objetivo de detectar lacunas e eventuais *bugs*, resultando um conjunto de potenciais melhorias/correções (Anexo A). Estas foram discutidas internamente na WingzStudio de forma a serem aprovadas e priorizadas de acordo com a preferência e o tempo disponível. As principais melhorias a implementar são:

- **Maior suporte a *aspect ratios*:** a versão original oferece suporte apenas a *aspect ratios* de orientação vertical (10:16, 9:16 e 3:4). De forma a executar em plataformas de orientação horizontal, existe a necessidade de suportar novos *aspect ratios*, tais como: 16:10, 16:9 e 4:3.

- **Modo história:** a versão original tem apenas um modo de jogo, baseado no estilo arcade, o que obriga a completar todos os níveis numa única sessão. Com o modo história, passa a ser possível desbloquear níveis ao completar os anteriores, evitando a frustração de jogadores que não têm tempo suficiente para completar a totalidade de níveis numa única sessão. Além do referido, os níveis passariam a ter pontuação individual e, consoante a pontuação obtida seria atribuído um número de taças conquistadas (0 a 3) nesse nível.
- **Modo desafio (multijogador):** este modo permitirá desafiar amigos ou outros para um duelo, o qual consistirá numa sessão de jogo semelhante a um nível do modo história. Essa sessão será jogada pelos dois jogadores, cada um terá apenas visibilidade sobre o seu jogo e não será realizada em tempo real, a razão de não ser em tempo real deve-se ao facto de obrigar as duas pessoas a estarem online no mesmo momento, o que poderia causar um longo tempo de espera por adversário. Com este modo pretende-se adicionar uma maior componente social, o que poderá resultar num aumento de jogadores e da sua taxa de retenção.
- **Suporte a rato e teclado:** como serão suportadas novas plataformas que fazem uso destes periféricos, é necessário suporte aos mesmos.
- **Adição de idiomas:** português e espanhol. Sendo este videojogo desenvolvido em Portugal por uma empresa portuguesa, é importante incluir o idioma português. Por outro lado, como a personagem principal é o colombiano Radamel Falcao, é essencial incluir também o idioma oficial da Colômbia (espanhol).
- **Sincronização entre dispositivos:** como irão ser suportadas várias plataformas, considerou-se importante fornecer uma funcionalidade que permita aos jogadores guardar o progresso também no servidor, através desta espera-se reduzir a frustração de começar de raiz ao trocar de dispositivo/plataforma.
- **Novos power ups:** após a análise da versão original, uma das conclusões obtidas é o facto de um dos elementos que torna o videojogo apelativo são os power ups, no entanto devido a não serem muito numerosos tornam-se repetitivos a curto prazo, então será aumentado o número de power ups diferentes.

Com o cumprimento dos objetivos mencionados espera-se que o videojogo atinja novos mercados e ofereça uma experiência mais cativante e variada, de forma a aumentar tanto o número de jogadores como a taxa de retenção.

3.2. Abordagem de Desenvolvimento

Para o desenvolvimento deste projeto foi escolhida uma metodologia de desenvolvimento *waterfall*. Esta metodologia é composta por diversas fases de desenvolvimento, as quais são visíveis na Figura 5: conceito, recolha e análise de requisitos, arquitetura geral do sistema, arquitetura detalhada do sistema, implementação, testes, lançamento do sistema.

A primeira fase consiste na definição do conceito, onde é definida a ideia geral do projeto e o seu propósito. Após a fase do conceito é efetuada a recolha e análise de requisitos, onde são recolhidos todos os possíveis requisitos sobre o sistema. De seguida estes passam por uma triagem onde são selecionados os que realmente importam, por fim esses requisitos são documentados.

A terceira e quarta fase consistem na definição da arquitetura geral e detalhada do sistema, as quais têm como objetivo dividir o sistema em componentes e descrevê-las detalhadamente, incluindo as vertentes *hardware* e *software*, estas deverão também explicar como o sistema deverá ser implementado. Com as fases anteriores concluídas, o projeto está pronto a entrar na fase de implementação, a qual corresponde ao desenvolvimento de código, material artístico, entre outros componentes.

Além da concepção são também realizados testes unitários consoante se implementa as diferentes componentes. Seguidamente ocorre a fase de testes, onde o sistema deverá ser testado como um todo, ou seja, com todas as componentes integradas. Caso sejam encontradas falhas, estas deverão ser corrigidas e o sistema deverá ser testado novamente, repetindo este processo até o sistema atingir a maturidade pretendida.

Após estas fases é esperado que o sistema esteja pronto, assim chegamos à última fase, que consiste no lançamento do sistema para o mercado. A metodologia *waterfall* segue um processo sequencial, ou seja, só quando uma fase está completa é que se avança para a próxima.

Uma das razões que levou à escolha desta metodologia é o facto do projeto ser de pequena dimensão e o videojogo alvo de *porting* já se encontrar no mercado, sendo grande parte dos requisitos bem conhecidos, claros e fixos. Outro facto que justifica o

uso desta metodologia é a baixa necessidade de interação com o cliente, visto que neste caso o cliente é a própria empresa responsável pelo projeto, pelo que as interações com a mesma ocorrem internamente e com o principal propósito de esclarecimento de dúvidas em relação às componentes do videojogo em causa.

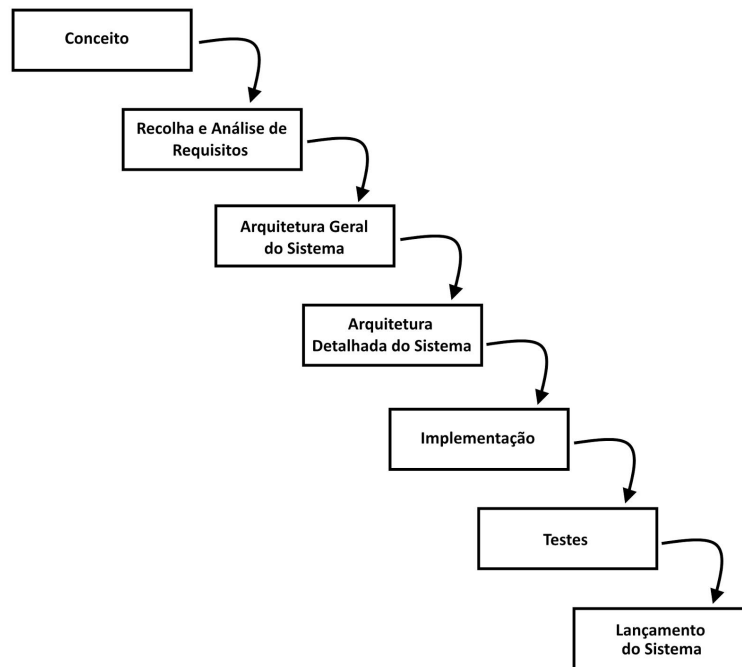


Figura 5 - Esquema representativo das fases da metodologia *waterfall*

3.3. Planeamento de Atividades

O plano de atividades deste projeto tem como finalidade organizar todos os passos do desenvolvimento e definir prazos para a conclusão dos mesmos. De forma a efetuá-lo de forma organizada e de fácil interpretação. Foram elaborados quatro diagramas de Gantt, dois deles correspondentes a cada um dos semestres e os restantes relativos às principais tarefas de implementação.

3.3.1. Primeiro Semestre

O início deste projeto estava previsto para Setembro de 2014, no entanto devido a problemas burocráticos só foi possível iniciar-se a 24 de Novembro do mesmo ano, desta forma o tempo para a sua elaboração, principalmente no 1º Semestre, foi reduzido.

Assim, o plano para o 1º Semestre encontra-se dividido em oito partes, como é visível na Figura 6. A primeira parte corresponde ao estado da arte, onde foi efetuada uma pesquisa sobre a área do *porting* de videojogos, como técnicas e processos

envolvidos, assim como boas práticas que facilitam esses processos. Foram ainda analisadas duas ferramentas que facilitam o processo de *porting*, *Unity Personal Edition* e *Cocos2d-x*. Na segunda parte, foi elaborado o planeamento de atividades para o 1º Semestre. Posteriormente foram definidos os objetivos para este projeto e de seguida a metodologia de desenvolvimento a usar. Como a metodologia escolhida foi a *waterfall*, prosseguiu-se para a recolha e análise de requisitos do “Falcao vs Aliens”, sendo elaborado um documento de requisitos. Com a análise de requisitos finalizada, o próximo passo foi a definição da arquitetura geral. Seguidamente foi efetuado o planeamento de atividades para o 2º Semestre (Figuras 7, 8 e 9). Por fim, iniciou-se a elaboração da arquitetura detalhada, esta tarefa foi interrompida pela necessidade de finalizar o relatório intermédio.

1º Semestre

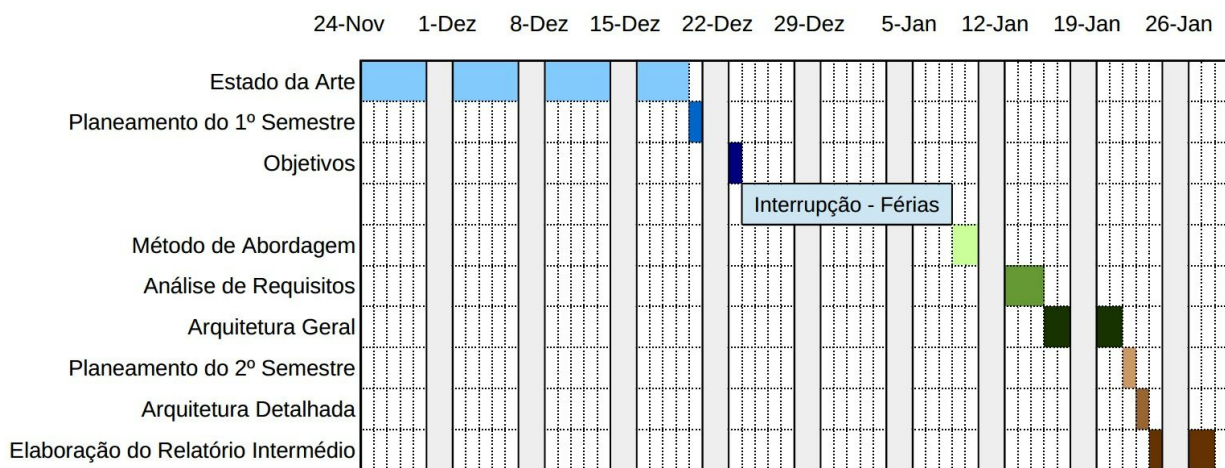


Figura 6 - Planeamento do Projeto - 1º Semestre

3.3.2. Segundo Semestre

No 2º Semestre o planeamento encontra-se dividido em sete partes, como se pode observar na Figura 7. Os objetivos iniciais para o projeto foram alterados no final do primeiro semestre, deste modo a primeira tarefa a ser realizada neste período consistiu na reformulação dos objetivos. Para o levantamento de melhorias, efetuou-se uma análise ao “Falcao vs Aliens” de forma encontrar possibilidades de melhoria. Seguidamente, foi reformulado o planeamento do segundo semestre, para estar em conformidade com os novos objetivos. Após a conclusão do planeamento, procedeu-se à continuação da elaboração da arquitetura detalhada, iniciada no primeiro semestre. Com a arquitetura finalizada, deu-se início à implementação do videojogo, ao mesmo tempo que foram efetuados testes informais, tanto pela equipa de desenvolvimento,

como por pessoas externas. Concluído o desenvolvimento, procedeu-se à elaboração do relatório, tendo os testes se mantido.

2º Semestre

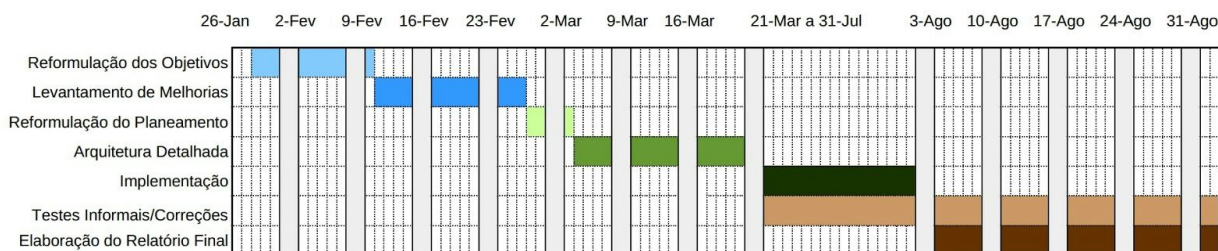


Figura 7 - Planeamento do Projeto - 2º Semestre

De forma a detalhar a fase de implementação, elaborou-se mais dois diagramas de Gantt (Figuras 8 e 9), onde é visível o tempo planeado para as principais tarefas de implementação.

A Figura 8 é referente às tarefas realizadas até ao final do *porting* do conteúdo original, sendo a primeira tarefa a implementação das funcionalidades presentes na versão original do “Falcao vs Aliens”. À medida que forem implementadas essas funcionalidades, também será incluído o suporte a novos *aspect ratios*. Nas últimas duas semanas do *porting* do conteúdo original, serão implementados os novos idiomas (português e espanhol), tendo como base o idioma Inglês, o qual se encontra presente na versão original.

Implementação (Parte 1 de 2)

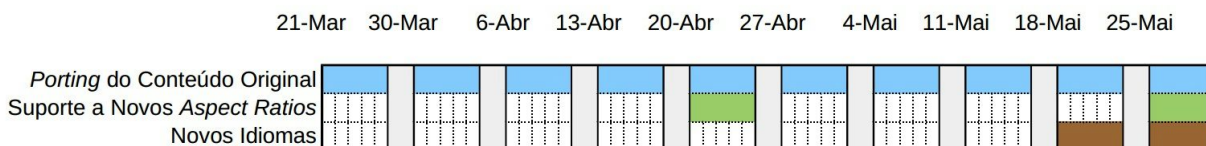


Figura 8 - Planeamento do Projeto - Implementação - Parte 1/2

Na Figura 9 são visíveis as tarefas de melhoria, as quais começam com a implementação do modo de jogo história, sendo necessário atualizar os vários idiomas com os textos referentes a este modo. Seguidamente, será desenvolvido um sistema de sincronização de dados entre os dispositivos: *Android*, *iOS*, *Windows Phone*, *Windows*, *Facebook* e navegadores atuais. Após a sincronização, será implementado outro modo de jogo, neste caso o modo desafio/multijogador. Por fim, serão criados novos *power ups* e será feita uma revisão geral do videojogo, de forma a confirmar se todos os textos estão corretamente traduzidos e se todas as secções são apresentadas corretamente nos vários *aspect ratios* suportados.

Implementação (Parte 2 de 2)

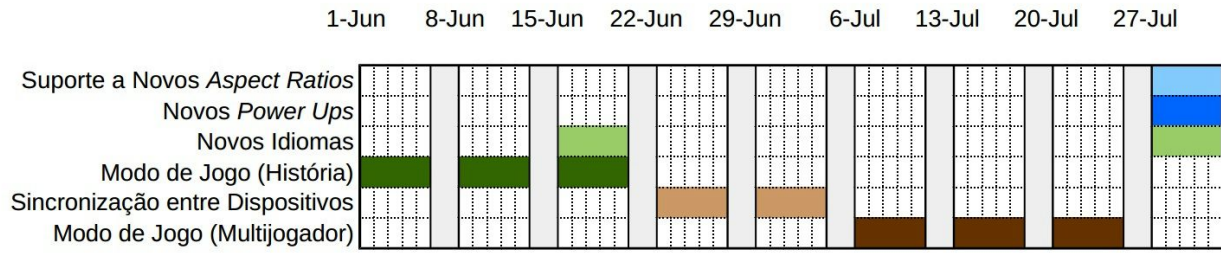


Figura 9 - Planejamento do Projeto - Implementação - Parte 2/2

4. Conceito, Requisitos e Arquitetura

Este capítulo está dividido em 7 subsecções: Conceito, História/Enredo, Descrição Geral do Videojogo, Requisitos Funcionais e Não-funcionais, Arquitetura Geral, Arquitetura Detalhada e *Design*. Na subsecção Conceito irá ser abordada a origem deste videojogo e a ideia geral por detrás do mesmo. Seguidamente será apresentado o enredo que serve de base ao videojogo. Posteriormente serão abordados os requisitos funcionais e não-funcionais, que ajudam a definir o comportamento do sistema. Na subsecção Arquitetura Geral, o sistema será dividido em componentes, as quais serão analisadas de uma forma geral. De seguida, será detalhada a arquitetura, desta forma serão descritos os vários componentes da mesma. Por fim, serão analisadas algumas partes do *design*, como menus e níveis do videojogo.

4.1. Conceito

O “Falcao vs Aliens” foi desenvolvido não apenas com o objetivo de aumentar o portefólio da empresa, mas também como resultado de uma parceria entre a WingzStudio e uma empresa responsável por gerir os direitos de imagem de atletas mundiais. Este videojogo foi publicado em 2014 e é o primeiro videojogo exclusivo que conta com a presença da estrela de futebol Radamel Falcao. Houve dois conceitos que foram discutidos entre as partes envolvidas, um seria baseado num estilo arcade semelhante ao conceito do clássico *Arkanoid*, que consiste na destruição de blocos com uma bola ao mesmo tempo que o jogador controla uma barra para evitar que a bola saia do ecrã. O outro conceito levaria à criação de um videojogo que se iria incluir na categoria de *puzzle*. Depois da apresentação dos conceitos à entidade externa, a decisão recaiu sobre o conceito arcade.

4.2. História/Enredo

Um grupo de alienígenas conseguiu interceptar uma transmissão de um jogo de futebol que ocorreu no planeta Terra. Essa transmissão despertou a curiosidade desses seres para com o futebol, decidiram então visitar o planeta Terra para aprender mais sobre esse desporto. No entanto, o facto de não terem conhecimento das regras fez com que estes seres não fossem bem recebidos pelos humanos, o que levou a que se apoderassem de estádios espalhados por todo o planeta ao mesmo tempo que afugentavam os espectadores. Vendo o seu desporto favorito ser destruído, o herói deste videojogo (Radamel Falcao), decide começar uma longa aventura com o objetivo

de recuperar todos os estádios controlados pelos alienígenas. De forma atingir o objetivo, Falcao terá de vencer estes seres em diversos jogos de futebol sem regras e resistir a todas as estratégias utilizadas por estas criaturas.

4.3. Descrição Geral do Videojogo

O videojogo é composto por 18 níveis, uma vez que o tema do jogo é futebol cada nível corresponderá a um estádio de futebol, havendo um total de 6 estádios diferentes. A ação decorrerá no relvado (Figura 10), estando a personagem principal na parte inferior do ecrã, que se encontra aproximadamente a meio do campo. Na zona oposta à personagem principal encontra-se a baliza adversária.

A versão original conta apenas com um único modo de jogo, no qual para acabar o videojogo era necessário completar todos os níveis sem perder. Existe um único jogador humano, este controla a personagem principal: Radamel Falcao. O jogador nesse modo começa com quatro vidas, correspondendo a quatro bolas, perde-se uma vida quando o Falcao deixa a bola passar. Na nova versão, serão adicionados dois modos de jogo, modo história e modo desafio. O modo história será semelhante ao modo arcade presente na versão original, mas o progresso será guardado, sendo possível desbloquear níveis e continuar a partir dos mesmos após a sessão terminar. O modo desafio, permite convidar um jogador humano para ambos competirem no mesmo nível, o vencedor é quem atingir a maior pontuação.



Figura 10 - "Falcao vs Aliens" - Exemplo de Nível

Existem inimigos alienígenas controlados por inteligência artificial, estes têm características diferentes entre eles, por exemplo alguns são fixos/móveis ou estão armados/desarmados. Ao acertar com a bola nos inimigos faz com que estes percam vida, se ficar a zero morrem e desaparecem do ecrã. Os alienígenas encontram-se na zona frontal à baliza, protegendo-a.

Os jogadores terão em cada nível que destruir os diversos alienígenas espalhados pelo cenário, abrindo desta maneira o caminho para a baliza. A baliza encontra-se inicialmente bloqueada com uma barreira, a qual só desaparece após serem apanhadas três estrelas, estas encontram-se escondidas atrás de alguns alienígenas, pelo que é necessário destruí-los para que estas sejam reveladas e possam ser apanhadas.

Após a baliza estar desbloqueada, um alienígena de maior estatura aparece a defendê-la, sendo então necessário destruí-lo para que a baliza fique aberta e seja possível marcar golo. O nível é completado assim que o jogador consegue marcar golo, permitindo ao jogador prosseguir para o nível seguinte. Além do que já foi referido, é também de destacar que: existem *power ups* que são possíveis de apanhar, estes vão aparecendo aleatoriamente consoante eliminamos os alienígenas, podendo ter efeitos positivos ou negativos; há também garrafas de energia que se podem apanhar, as quais protegem a personagem principal dos ataques dos inimigos; existem *achievements* colecionáveis. O videojogo contará com a integração com o *Facebook*, o que permitirá aos jogadores partilhar os seus *achievements* e pontuações obtidas.

4.4. Requisitos Funcionais e Não-funcionais

Nesta subsecção serão apresentados os requisitos funcionais e não-funcionais para este projeto, no entanto foi elaborado um documento onde estes são abordados com maior detalhe, o qual poderá ser consultado no Anexo B.

Para estabelecer o grau prioridade dos mesmos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial:** requisito imprescindível sem o qual o sistema não entra em funcionamento.
- **Importante:** requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória.

- **Desejável:** requisito que não compromete as funcionalidades do sistema, estes poderão ser adicionados posteriormente, caso não exista tempo para os implementar nesta versão.

Como forma de distinguir os requisitos que foram mantidos, introduzidos ou alterados em relação à versão original, estes serão classificados com os seguintes termos: “novo”, “alterado” e “mantido”.

- **Novo:** requisito introduzido nesta versão do sistema, não estando presente na original.
- **Alterado:** requisito que será incluído na nova versão, com algumas alterações em relação à original.
- **Mantido:** requisito existente na versão original e que será incluído também nesta nova versão.

Seguidamente serão apresentados os requisitos funcionais, os quais se dividem em três categorias.

➤ **Registo/Login:**

[RF001] Efetuar <i>login</i>	
Prioridade: Importante	Classificação: Alterado
Descrição: Este caso permite que o utilizador efetue o <i>login</i> utilizando os dados de uma conta do <i>Facebook</i> .	

[RF002] Efetuar registo	
Prioridade: Importante	Classificação: Alterado
Descrição: Proporciona que o utilizador efetue o registo de uma conta no <i>Facebook</i> .	

➤ **Interface:**

[RF001] Menu Principal	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso uso viabiliza que o utilizador acesse o menu principal.	

[RF002] Aceder ao Menu de Seleção do Modo de Jogo	
Prioridade: Essencial	Classificação: Novo
Descrição: Permite que o utilizador acesse o menu de seleção do modo de jogo.	

[RF003] Entrar no Modo História	
Prioridade: Essencial	Classificação: Novo
Descrição: Este garante que o utilizador entre no modo de jogo história.	

[RF004] Entrar num nível do Modo História	
Prioridade: Essencial	Classificação: Novo
Descrição: Possibilita que o utilizador entre num nível do modo de jogo história.	

[RF005] Entrar no Modo Arcade	
Prioridade: Essencial	Classificação: Alterado
Descrição: Este caso de uso permite que o utilizador entre no modo de jogo arcade.	

[RF006] Entrar no Modo Desafio	
Prioridade: Desejável	Classificação: Novo
Descrição: Este proporciona que o utilizador entre no modo de jogo desafio.	

[RF007] Desafiar um Jogador	
Prioridade: Desejável	Classificação: Novo
Descrição: Permite ao utilizador que convide outra pessoa para jogar no modo desafio.	

[RF008] Aceitar Desafios	
Prioridade: Desejável	Classificação: Novo
Descrição: Este caso possibilita que o utilizador aceite desafios.	

[RF009] Consultar Estatísticas de Desafios	
Prioridade: Desejável	Classificação: Novo
Descrição: Permite que o utilizador visualize as estatísticas relativas ao modo desafio, mais especificamente o número de vitórias e derrotas.	

[RF010] Apresentar Tutorial	
Prioridade: Importante	Classificação: Mantido
Descrição: Possibilita que o utilizador visualize o tutorial que ensina como jogar.	

[RF011] Apresentar Pontuação Atual	
Prioridade: Essencial	Classificação: Mantido
Descrição: Permite ao utilizador visualizar a pontuação atual numa sessão de jogo.	

[RF012] Apresentar Pontuação Recorde	
Prioridade: Essencial	Classificação: Mantido
Descrição: Proporciona que o utilizador visualize a pontuação recorde enquanto se encontra numa sessão de jogo.	

[RF013] Aceder ao <i>In-game</i> Menu	
Prioridade: Essencial	Classificação: Mantido
Descrição: Possibilita ao utilizador aceder a um menu durante uma sessão de jogo.	

[RF014] Voltar ao Menu Principal	
Prioridade: Essencial	Classificação: Mantido
Descrição: Permite ao utilizador voltar ao menu principal durante a sessão de jogo.	

[RF015] Desativar/ativar Som	
Prioridade: Essencial	Classificação: Mantido
Descrição: Permite ao utilizador desativar/ativar o som durante uma sessão de jogo.	

[RF016] Apresentar Pontuações após <i>Gameover</i>	
Prioridade: Essencial	Classificação: Mantido
Descrição: Possibilita ao utilizador visualizar a pontuação obtida numa sessão de jogo do modo arcade e o seu recorde, após derrota.	

[RF017] Apresentar Pontuações após Completar o Jogo	
Prioridade: Essencial	Classificação: Mantido
Descrição: Proporciona ao utilizador visualizar a pontuação obtida e o seu recorde, após a conclusão da sessão de jogo.	

[RF018] Apresentar Pontuações após Concluir um Nível do Modo História	
Prioridade: Essencial	Classificação: Novo
Descrição: Possibilita ao utilizador visualizar a pontuação obtida numa sessão de jogo do modo história e o seu recorde, após concluir o nível em causa.	

[RF019] Voltar ao Menu Principal após Terminar a Sessão de Jogo	
Prioridade: Essencial	Classificação: Mantido
Descrição: Viabiliza que o utilizador volte ao menu principal após a sessão de jogo.	

[RF020] Aceder Opções	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso de uso permite que o utilizador aceda ao menu de opções.	

[RF021] Alterar Opções	
Prioridade: Essencial	Classificação: Mantido
Descrição: Possibilita que o utilizador altere as opções de jogo.	

[RF022] Efetuar <i>Reset</i>	
Prioridade: Importante	Classificação: Mantido
Descrição: Este permite que o utilizador faça <i>reset</i> a todos os dados do jogo (<i>achievements</i> , opções, se já viu o tutorial alguma vez, ...).	

[RF023] Ativar Sincronização	
Prioridade: Importante	Classificação: Novo
Descrição: Viabiliza que o utilizador ative a sincronização de dados.	

[RF024] Desativar Sincronização	
Prioridade: Importante	Classificação: Novo
Descrição: Possibilita ao utilizador desativar a sincronização de dados.	

[RF025] Mostrar Créditos	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso permite que o utilizador veja os créditos.	

[RF026] Aceder ao Menu de <i>Achievements</i> (cacifo)	
Prioridade: Essencial	Classificação: Mantido
Descrição: Possibilita que o utilizador aceda ao menu de <i>achievements</i> .	

[RF027] Consultar <i>Achievements</i>	
Prioridade: Essencial	Classificação: Mantido
Descrição: Proporciona que o utilizador aceda à descrição/informação de cada <i>achievement</i> .	

[RF028] Aceder ao Menu de Cromos Colecionáveis	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso de uso permite que o utilizador aceda ao menu de cromos colecionáveis.	

[RF029] Consultar Cromos Colecionáveis	
Prioridade: Essencial	Classificação: Mantido
Descrição: Permite que o utilizador aceda à descrição/informação de cada cromo.	

[RF030] Partilhar Cromos Colecionáveis no <i>Facebook</i>	
Prioridade: Importante	Classificação: Mantido
Descrição: Possibilita que o utilizador partilhe um cromo no <i>Facebook</i> .	

[RF031] Sair do Jogo na Totalidade	
Prioridade: Importante	Classificação: Mantido
Descrição: Este caso de uso permite que o utilizador saia por completo do jogo.	

➤ **Jogabilidade:**

[RF001] Controlar a Personagem	
Prioridade: Essencial	Classificação: Mantido
Descrição: Permite que o utilizador controle a personagem principal.	

[RF002] Acertar e Causar Dano nos Inimigos	
Prioridade: Essencial	Classificação: Mantido
Descrição: Proporciona que o utilizador acerte e cause dano nos alienígenas.	

[RF003] Impedir Personagem de Sair do Ecrã	
Prioridade: Essencial	Classificação: Mantido
Descrição: Impossibilita a personagem principal saia do terreno de jogo.	

[RF004] Impedir a Bola Saia do Ecrã	
Prioridade: Essencial	Classificação: Mantido
Descrição: Impedir que a bola saia do terreno de jogo.	

[RF005] Restantes Colisões da Bola	
Prioridade: Essencial	Classificação: Mantido
Descrição: Colisões da bola com alienígenas, barreira da baliza, postes da baliza, personagem principal.	

[RF006] Perder uma Vida	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso de uso descreve o cenário em que ocorre a perda de vidas por parte do personagem principal.	

[RF007] <i>Gameover</i>	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso detalha a situação em que o jogador perdeu todas as vidas numa sessão de jogo, obtendo <i>gameover</i> .	

[RF008] <i>Nível Completado com Sucesso</i>	
Prioridade: Essencial	Classificação: Mantido
Descrição: Retrata o cenário em que o jogador obteve sucesso no nível atual.	

[RF009] <i>Jogo Completado com Sucesso</i>	
Prioridade: Essencial	Classificação: Mantido
Descrição: Este caso descreve a situação em que o jogador obteve sucesso em todos os níveis atingindo o final do jogo.	

Como forma de analisar estes requisitos funcionais, de acordo com as modificações introduzidas em relação à versão original, é apresentada na Tabela 2 o número de requisitos novos, alterados e mantidos. Também será apresentada a percentagem dos mesmos em relação ao total.

Comparação entre Requisitos Funcionais			
Novos	Alterados	Mantidos	Total
10 (24%)	3 (7%)	29 (69%)	42 (100%)

Tabela 2 - Comparação entre os Requisitos Funcionais - "Falcao vs Aliens"

De seguida serão apresentados os requisitos não-funcionais, estando divididos em duas categorias.

➤ **Interface:**

[RNF001] Interface Gráfica	
Prioridade: Essencial	Classificação: Mantido
Descrição: O videojogo deverá oferecer uma interface gráfica.	

[RNF002] Interface Intuitiva	
Prioridade: Essencial	Classificação: Mantido
Descrição: O videojogo deverá oferecer ao jogador a possibilidade de usufruir das suas funcionalidades principais de forma rápida e fácil.	

[RNF003] Resolução	
Prioridade: Importante	Classificação: Alterado
Descrição: O videojogo deverá oferecer suporte todas as resoluções.	

➤ **Outros Aspectos:**

[RNF001] Performance	
Prioridade: Importante	Classificação: Mantido
Descrição: O videojogo deverá ser capaz de executar a 60 <i>frames</i> por segundo sem quebras, a não ser que a plataforma não suporte.	

[RNF002] Plataformas suportadas	
Prioridade: Essencial	Classificação: Alterado
Descrição: O videojogo deverá executar pelo menos nas seguintes plataformas: <i>Android, iOS, Windows Phone, Windows, Facebook</i> , navegadores atuais.	

[RNF003] Espaço físico	
Prioridade: Essencial	Classificação: Mantido
Descrição: O <i>apk</i> do videogame para <i>Android</i> não poderá exceder os 50Mb de espaço físico.	

[RNF004] Memória	
Prioridade: Essencial	Classificação: Mantido
Descrição: O videogame não poderá exceder os 200Mb de memória em dispositivos móveis.	

[RNF005] Ferramenta de desenvolvimento	
Prioridade: Importante	Classificação: Alterado
Descrição: O videogame deverá ser desenvolvido utilizando o motor de jogo <i>Unity</i> .	

Na Tabela 3 é contabilizada a totalidade dos requisitos não-funcionais novos, alterados e mantidos. Também será apresentada a percentagem dos mesmos em relação ao total.

Comparação entre Requisitos Não-funcionais			
Novos	Alterados	Mantidos	Total
0 (0%)	3 (37.5%)	5 (62.5%)	8 (100%)

Tabela 3 - Comparação entre os Requisitos Não-funcionais - "Falcao vs Aliens"

4.5. Arquitetura Geral

A Figura 11 representa a visão geral do sistema, o qual é composto pelo *core*, que corresponde ao núcleo do videojogo, este troca informação com os seguintes componentes/sistemas: *Facebook*, servidor, e armazenamento local de dados.

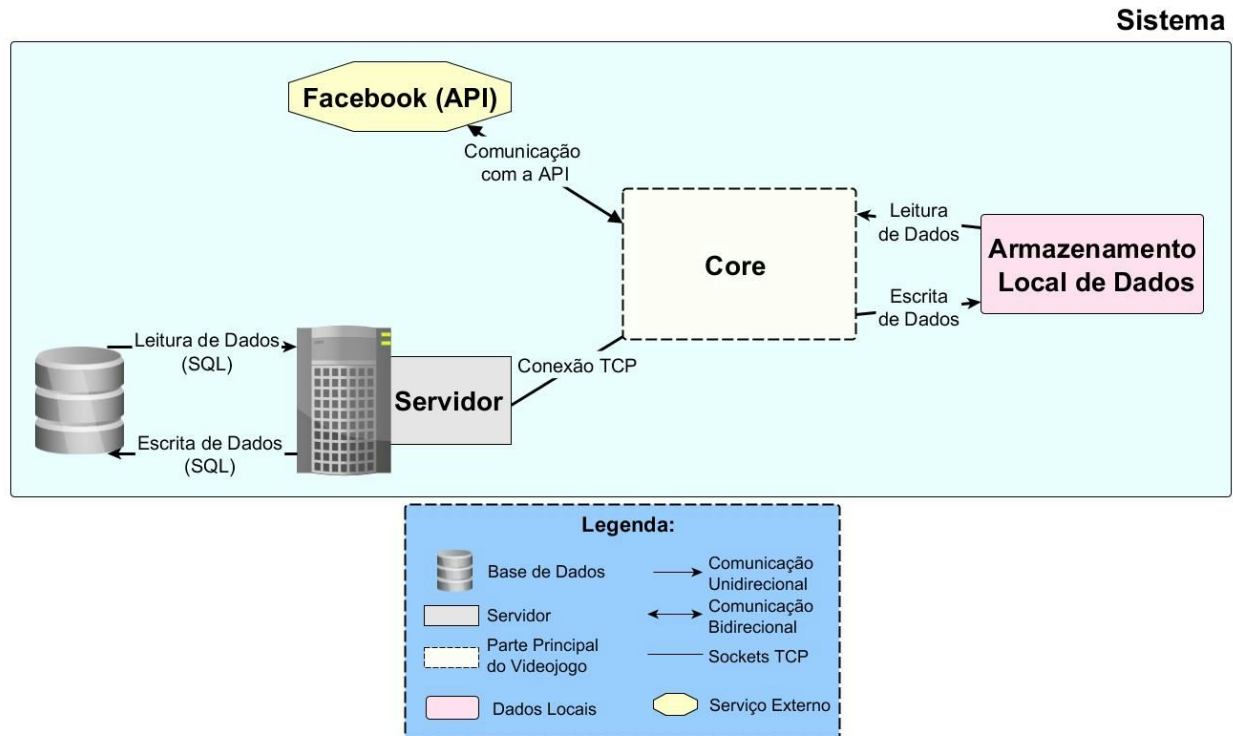


Figura 11 - Arquitetura - Visão Global

A comunicação com o *Facebook* é feita através da *API* disponibilizada para *Unity*, esta é utilizada com os principais objetivos de permitir aos utilizadores partilhar conteúdo na rede social acerca do videojogo (pontuação, cromos) e convidar outras pessoas para jogar.

O servidor serve de acesso à base de dados, apesar de ser possível aceder diretamente à base de dados, com um servidor consegue-se aumentar a segurança, além de que este ainda implementa outras funcionalidades, como por exemplo a conversão de nomes com caracteres não suportados pelo código *American Standard Code for Information Interchange (ASCII)*¹⁵. A comunicação com o servidor é efetuada através de *sockets TCP (Transmission Control Protocol)*, sendo as mensagens enviadas no formato *JSON (JavaScript Object Notation)*¹⁶ e encriptadas através de um

¹⁵ <http://www.asciitable.com>

¹⁶ <http://json.org>

algoritmo simples, o qual não será detalhado a pedido da WingzStudio. A decisão de optar por este algoritmo, deve-se ao facto dos dados não serem confidenciais, não sendo necessária a utilização de uma encriptação complexa. Os modelos das várias mensagens *JSON* são apresentados no Anexo C.

O armazenamento local de dados é usado como fonte de gravação e leitura das preferências e progresso do utilizador, este é efetuado através da classe *PlayerPrefs*¹⁷ do motor *Unity*. Os dados são gravados em diferentes formatos e locais consoante a plataforma usada.

A base de dados é responsável por guardar dados relativos ao progresso e pontuações dos utilizadores, esta encontra-se implementada através do *MySQL*¹⁸. A comunicação com a base de dados (leitura e escrita), tal como mencionado é feita através do servidor.

O *core* é responsável por toda a interface gráfica e jogabilidade do videojogo, este é implementado através da utilização do motor *Unity* em conjunto com o *Mono Runtime*, como é visível na Figura 12.

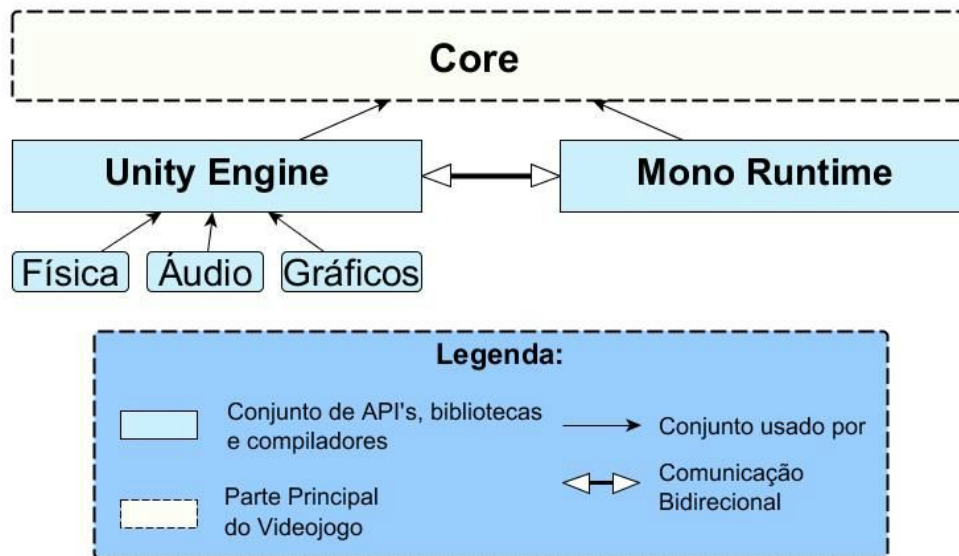


Figura 12 - Arquitetura - Mecanismo de Implementação do Core

¹⁷ <http://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

¹⁸ <https://www.mysql.com>

4.6. Arquitetura Detalhada

Nesta subsecção serão analisados em detalhe o *core* do “Falcao vs Aliens” e os vários componentes que o compõem. Através desta análise, será descrito o funcionamento interno do videogame, assim como a implementação do mesmo.

4.6.1. Core

Na Figura 13 são visíveis três grupos de componentes: *resources*, *scripts* e *scenes*. *Resources* é o conjunto de ficheiros composto por: *assets* (como áudio e *spritesheets*), *fonts*, ficheiros com os textos nos vários idiomas e com informação sobre os níveis. As *scenes* contêm os vários cenários de jogo, sendo constituídas por objetos e componentes, os quais fazem uso dos *resources* e *scripts*. Os *scripts* contêm o código relativo à parte central do videogame (interface, jogabilidade), estes têm direta influência sobre as várias *scenes* ao mesmo tempo que são utilizados como componentes de objetos dentro das *scenes*.

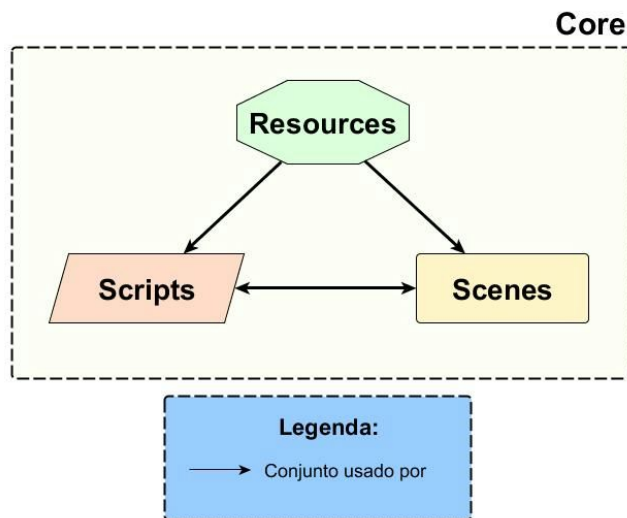


Figura 13 - Arquitetura - Componentes do Core

4.6.2. Scenes

A arquitetura das *scenes* que compõem o videogame está detalhada na Figura 14, estas são implementadas em *Unity* e, são compostas por objetos e componentes dos mesmos.

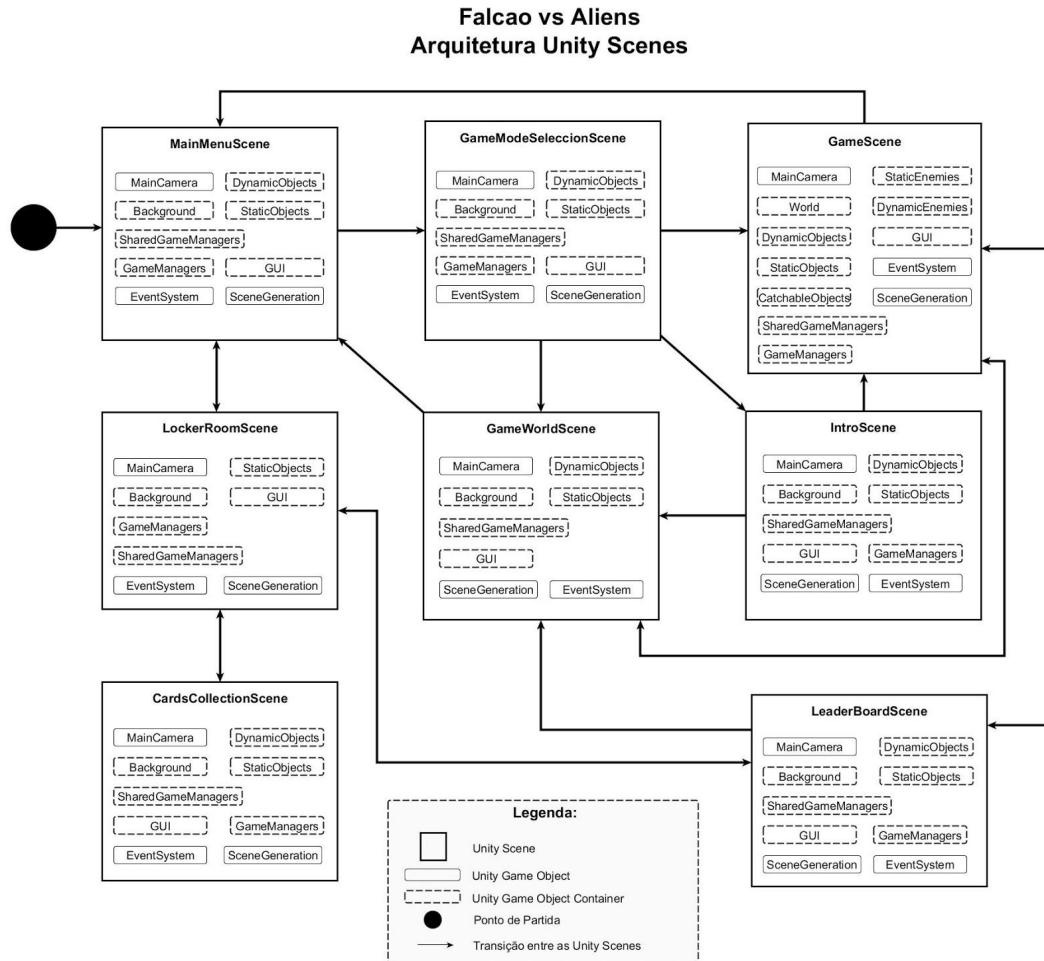


Figura 14 - Arquitetura - Scenes

Estas *scenes* têm uma organização semelhante entre elas, contendo: câmera, objetos (estáticos e dinâmicos), plano de fundo e elementos que o complementam, interface de utilizador, *game managers*, sistema de eventos e, por último um objeto responsável por configurar/gerar diversos objetos e componentes de cada *scene*. A câmera é ortogonal, mantendo-se fixa durante decorrer de toda ação, sendo configurada quando se executa o videogame de forma a adaptar-se aos vários *aspect ratios*. Cada *scene* tem os seus próprios objetos estáticos e dinâmicos, estes atuam como contentores de outros objetos. Os planos de fundo, que também são contentores, são compostos pelos vários elementos que definem o fundo de cada *scene*. A interface

de utilizador, contém todos os elementos relacionados com a mesma ou seja, botões, notificações e deteção de *inputs*. O sistema de eventos está diretamente relacionado com a interface, sendo este uma forma de enviar eventos relacionados com a deteção de *inputs* para outros objetos. Os *game managers* são partilhados entre as diversas *scenes* e são implementados através do *design pattern, singleton*. Estes *game managers* são compostos por um leitor de recursos, configurações globais e dois gestores de comunicação, um com o *Facebook* e outro com o servidor. O leitor de recursos é responsável por carregar *spritesheets*, áudio e textos. As configurações globais, são responsáveis pela gestão de todas as configurações, como: opções gerais (música, sons, linguagem e sincronização), progresso atual do modo história, *achievements*, cromos obtidos e pontuações. Por último, os gestores de comunicação, são responsáveis por toda a comunicação com o *Facebook* e com o servidor, as principais funcionalidades destes são: criação/formatação de mensagens, encriptação/descriptação e envio/recepção de mensagens.

No total existem oito *scenes* diferentes (Figura 14), sendo elas: “MainMenuScene”, “LockerRoomScene”, “CardsCollectionScene”, “GameModeSeleccionScene”, “IntroScene”, “GameScene”, “GameWorldScene” e “LeaderBoardScene”.

Quando se inicia o videojogo a primeira *scene* que executa é a “MainMenuScene”, a qual consiste no menu principal juntamente com as opções gerais. A partir desta *scene* é possível transitar para as seguintes: “GameModeSeleccionScene” e “LockerRoomScene”, como é visível na Figura 14.

A “LockerRoomScene” é responsável por mostrar os *achievements* conquistados pelo jogador, assim como algumas hiperligações para páginas *web*, como por exemplo a página da WingzStudio. Esta *scene* tem ligação às seguintes: “MainMenuScene”, “CardsCollectionScene” e “LeaderBoardScene” (Figura 14).

“CardsCollectionScene” contém o album de cromos colecionáveis, sendo possível ver em detalhe cada um e partilhá-los no *Facebook*. A partir desta *scene* só é possível transitar para a “LockerRoomScene” (Figura 14).

A “LeaderBoardScene” consiste na apresentação das pontuações obtidas pelo jogador, assim como a tabela de maiores pontuações obtidas pelos vários jogadores. Esta *scene* dependendo de qual foi a origem, apresenta informações e acesso a *scenes* diferentes, pois esta é acedida através da “LockerRoomScene” e da “GameScene”. No primeiro caso, esta apresenta as pontuações obtidas no modo arcade e um botão para retornar à “LockerRoomScene”. No segundo caso, se o jogador estiver numa sessão de jogo do modo história, esta *scene* irá apresentar as

pontuações relativas ao nível atual, oferecendo a possibilidade de repetir o nível, voltar para o mundo de jogo ou voltar para o menu principal; se o jogador estiver numa sessão do modo arcade, esta *scene* irá apresentar pontuações relativas a este modo e a possibilidade de recomeçar a sessão de jogo ou voltar ao menu principal.

“GameModeSeleccionScene” contém os menus de seleção de modos de jogo e do modo desafio, sendo o primeiro composto por botões para os três modos de jogo (história, arcade e desafio) e o segundo apresenta as opções de aceitar desafios ou desafiar um jogador, são também visíveis as estatísticas relacionadas com este modo. Partindo desta *scene* é possível navegar para a “IntroScene”, “GameScene”, “GameWorldScene” e voltar para a “MainMenuScene” (Figura 14).

A “IntroScene” é responsável por apresentar uma sequência animada, que serve como introdução aos modos de jogo, caso seja a primeira vez o jogador é obrigado a assistir, caso contrário poderá fazer *skip* a esta. Dependendo do modo de jogo escolhido, esta *scene* encaminhará o jogador para a “GameWorldScene” (modo história) ou para a “GameScene” (modo arcade).

“GameWorldScene” contém o mundo de jogo, o qual é usado para o modo história, sendo visíveis os seis estádios que contêm três níveis cada um. Através desta *scene* é possível transitar para a “GameScene” ou voltar para a “MainMenuScene”.

A “GameScene” consiste na parte jogável, sendo responsável por gerar o nível atual, assim como todos os elementos do mesmo. A partir desta *scene* é possível navegar para as seguintes *scenes*: “MainMenuScene”, “GameWorldScene” (no caso da sessão atual ser do modo história) e “LeaderBoardScene”.

4.7. Design

Nesta subsecção serão descritas alguns componentes do “Falcao vs Aliens”, como a interface, fluxo de menus e objetos que compõe o cenário de jogo.

O videojogo, tal como foi mencionado nas Subsecções 4.2 e 4.3, é inspirado no clássico *Arkanoid*, no entanto não é uma cópia exata do mesmo, até porque o tema geral é o futebol, o que levou à adição de uma baliza e da mudança do objetivo central, que no caso do *Arkanoid* é destruir todos os blocos, mas no “Falcao vs Aliens” é marcar golo.

4.7.1. Design de um Nível do Videojogo

Um nível apresenta um aspeto idêntico à Figura 15, na Secção 4.3 foi feita a descrição de como se joga e como completar um nível, pelo que nesta subsecção irão ser focados os elementos que constituem cada nível.



Figura 15 - Design de um nível do videojogo (com barreira a bloquear a baliza)

Legenda: 1-Pontuação atual; 2-Resultado global (progresso); 3-Melhor pontuação; 4-Botão para pausar e aceder ao menu; 5-Baliza (inicialmente bloqueada por uma barreira); 6-Barreira (necessário apanhar três estrelas para que esta seja removida); 7-Garrafa de energia; 8-Estrela; 9-Alienígena (inimigo); 10-Bola; 11-Power ups (o da esquerda tem efeito negativo enquanto que o da direita tem efeito positivo); 12-Falcao (personagem controlável); 13-Barra de energia (cheia e vazia); 14-Bola representativa de uma vida (no caso da imagem o jogador possui duas vidas além da atual); 15-Slider (indicador do movimento);

A Figura 15 representa uma situação inicial de jogo onde a baliza ainda se encontra bloqueada pela barreira, sendo então necessário apanhar três estrelas para que esta barreira seja removida do cenário. Essa barreira é um objeto rígido, pelo que quando existe uma colisão da bola com o mesmo, a bola faz ricochete. Além dessa barreira, ainda existem barreiras laterais invisíveis, que aquando da colisão da bola

com estas ocorre a mesma reação, estas barreiras servem para impedir que a bola saia do terreno de jogo.

Existem vários elementos que pertencem à interface, um deles é a barra superior. Esta contém várias informações, como: pontuação atual, pontuação recorde e o progresso atual. Além das informações mencionadas, nesta barra também se situa o botão responsável por pausar o jogo e aceder a um menu, esse menu (apesar de não ser visível na Figura 15) contém como principais opções: ligar/desligar som (música e efeitos sonoros), continuar jogo, reiniciar ou sair. Outros elementos que fazem parte da interface são: a quantidade de estrelas apanhadas, visível no centro da barreira protetora da baliza; o número de vidas restantes, que se encontram no canto inferior direito; a barra de energia do Falcao, situada no canto inferior esquerdo; *slider* que representa o movimento, semelhante a um botão mas não é obrigatório o seu uso.

Os restantes elementos têm um impacto maior na ação de jogo e são constituídos por: Falcao, alienígenas, bola, *power ups*, estrelas e garrafas de energia. O Falcao é a personagem principal, esta é controlada pelo jogador através do rato, teclado ou dedo (em plataformas que suportam *touch*), sendo possível movimentá-la horizontalmente. O jogador deve manter a bola dentro do terreno de jogo, para isso terá que impedir que esta passe para as suas costas, caso contrário perde uma vida. Os alienígenas são os adversários, estes têm diferentes características, como a possibilidade de disparar e de se movimentar (visível na Figura 16), existe também outro tipo de alienígena que tem a função de guarda-redes (Figura 16), pelo que é necessário destruí-lo para poder marcar golo. Os alienígenas têm uma barra que representa a vida restante, quando estes colidem com uma bola essa barra diminui, ao desaparecer estes são destruídos. As estrelas estão escondidas atrás de certos alienígenas e só são visíveis após a destruição dos mesmos, estas podem ser capturadas, para isso deverá colidir com a bola. As garrafas de energia são visíveis desde o início do nível, estas são capturadas da mesma forma que as estrelas e enchem a barra de energia, servindo de defesa contra disparos alienígenas (Figura 16). Os *power ups* são outro elemento capturável, estes aparecem aleatoriamente consoante se destrói alienígenas.

Por último, existem ainda os painéis publicitários (Figura 16), que possuem um conjunto de imagens, as quais vão alternando entre si através do desaparecimento e aparecimento gradativo.

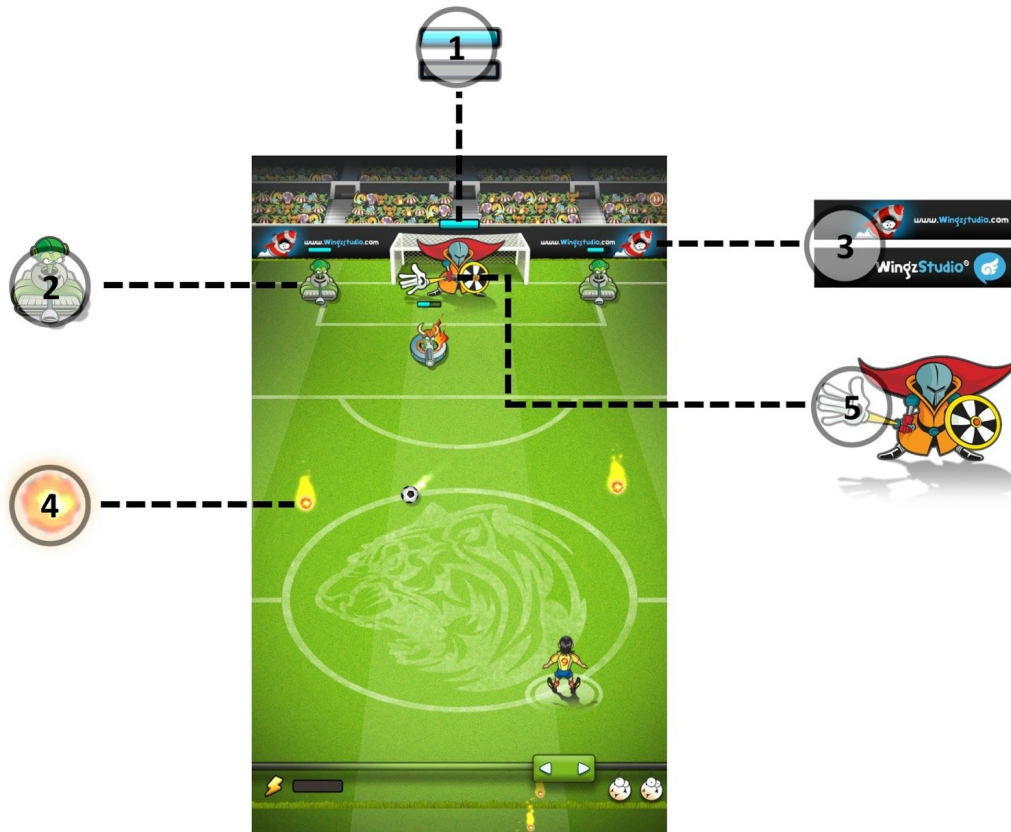


Figura 16 - *Design* de um nível do videojogo (sem barreira a bloquear a baliza)

Legenda: 1-Indicador de vida do alienígena; 2-Alienígena que dispara; 3-Painéis publicitários; 4-Disparo do alienígena; 5-Guarda-redes alienígena (necessário destruí-lo para poder marcar golo);

4.7.2. Organização Geral do Fluxo

Na Subsecção 4.7.1 foi descrito o *design* de um nível de jogo, nesta seção serão descritos os seguintes: menu principal, menu de opções, cacifo, menu de seleção de modo de jogo, menu modo desafio, mundo de jogo e o menu acessível durante a sessão de jogo. Estes menus possuem ligação entre si como é possível verificar na Figura 17.

O menu principal é o primeiro menu com o qual o jogador tem contato ao entrar no videojogo, a partir deste menu é possível sair da aplicação e navegar para os menus de opções, cacifo e de seleção de modos. No menu de opções, é possível alterar certas configurações, como: ligar/desligar a música e os efeitos sonoros; efetuar *reset* a todos os dados do jogador (conquistas, pontuações, entre outros); ativar/desativar sincronização de dados; e alterar o idioma (inglês, português e espanhol). É também possível visualizar os créditos e voltar ao menu principal. No cacifo é possível: visualizar as cartas e troféus adquiridos; aceder a páginas *web* externas, como por exemplo a página da WingzStudio; aceder às pontuações; retornar

ao menu principal. No menu de seleção de modo, existem três à escolha: história, arcade e desafio. O menu de desafio apresenta todas as opções relacionadas com este modo, ou seja, aceitar desafios e desafiar jogadores, assim como também consultar o número de vitórias e derrotas. O mundo de jogo apresenta os vários estádios e níveis bloqueados/desbloqueados, no caso de desbloqueados é visível as taças obtidas pelo jogador.

No caso de iniciar uma sessão de jogo, durante a mesma pode-se aceder a um menu, que conta com as principais opções: desligar/ligar som, continuar jogo, reiniciar e sair. No caso de *gameover* ou de completar o videojogo é apresentada a informação da pontuação e volta-se ao menu principal ou ao mundo de jogo (no caso do modo história).

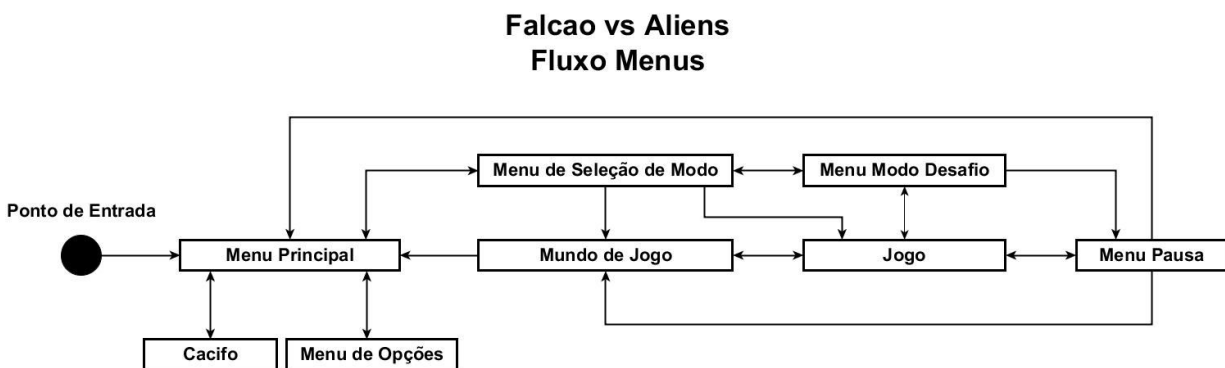


Figura 17 - Fluxo Menu - "Falcao vs Aliens"

4.7.3. Organização Detalhada do Fluxo

Na Subsecção 4.7.2 foi descrito o fluxo geral do videojogo, nesta secção este será descrito com mais detalhe, para isso serão apresentadas cinco imagens (Figuras 18, 19, 20, 21 e 22) que representam detalhadamente o menu principal, modos de jogo, menu de opções, cacifo e sessão de jogo.

Na Figura 18 é possível observar o fluxo inicial da aplicação, ou seja, a abertura, *loading* inicial, menu principal e janela de confirmação de encerramento da aplicação.



Figura 18 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo Inicial - Parte 1/5
Legenda: Seta contínua- apenas há um caminho possível e a transição é automática ou através do *touch* no ecrã; Setas continua- necessário pressionar botão específico; Seta curva- retroceder.

A Figura 19 representa o fluxo originário do botão *play* no menu principal, sendo direcionado para o menu de seleção de modos de jogo, o qual contém quatro botões, três destes correspondem aos modos (história, arcade e desafio) e o último serve para retornar ao menu principal.

Ao clicar no botão do modo história (mapa), inicia-se um *loading* e seguidamente é mostrada uma introdução animada, onde é visível uma nave espacial a desembarcar alienígenas num estádio de futebol, do qual pessoas fogem desesperadas, após esta sequência a camera afasta-se e mostra o Radamel Falcao no cimo de uma colina. Após esta introdução, o jogador é encaminhado para o mundo de jogo, o qual conta com um botão para retornar ao menu principal e um mapa com os vários estádios clicáveis, tendo cada um deles três níveis, após clicar no nível desejado, inicia-se um *loading* e começa a sessão de jogo.

O botão do modo desafio direciona o jogador para o menu deste modo, o qual é composto pelo botão de voltar ao menu de seleção de modos, nome e foto de perfil do *Facebook*, estatísticas de vitórias e derrotas, lista de desafios pendentes e dois botões: jogar/desafiar e histórico. O primeiro abre uma janela externa do *Facebook* onde é possível escolher que amigo desafiar, de seguida é apresentado o mundo de jogo de forma a selecionar o nível pretendido e por fim inicia-se um *loading* que leva à sessão de jogo. O segundo mostra uma lista com informação acerca dos últimos cinco jogos realizados (adversário, resultado, pontuações e data).

Por fim, o botão do modo arcade encaminha o jogador para um *loading* e à semelhança do modo história, também neste é apresentada a introdução animada. Seguidamente é apresentado um tutorial que, caso seja a primeira vez, é obrigatório

assistir até ao fim. Após o tutorial, o jogador encontra-se em posição de iniciar a sessão de jogo.



Figura 19 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo da Seleção de Modos de Jogo - Parte 2/5

Legenda: Seta contínua- apenas há um caminho possível e a transição é automática ou através do *touch* no ecrã; Seta contínua- necessário pressionar botão específico; Seta curva- retroceder.

A Figura 20 representa o fluxo originário de uma sessão de jogo, este pode ter um de três pontos de origem: clicar no botão do modo arcade, seleccionar nível no modo história ou desafiar um jogador. No cenário de jogo poderão ocorrer três situações: aceder ao menu de pausa, *gameover* ou completar o nível. A primeira, ocorre quando se pressiona o botão de pausa (ou através de teclas de atalho), ao fazê-lo abre-se um menu que tem como principais opções: ligar/desligar som, continuar jogo, repetir e sair.

A segunda ocorre quando o jogador fica sem vidas, o que origina o cenário de *gameover*, apresentando ao jogador a informação de que perdeu. Caso a sessão atual seja do modo arcade é apresentada a pontuação obtida e, após um *loading* é redirecionado para o menu inicial. No caso do modo história, após o *gameover* o

jogador terá três botões à disposição, sendo um deles para reiniciar o nível, outro para voltar ao mundo de jogo e outro para o menu principal.

A última situação ocorre quando se consegue marcar golo, ao fazê-lo é mostrada uma animação de comemoração, caso seja o modo arcade começa o nível seguinte, caso contrário é apresentada a pontuação obtida assim como a taça conquistada e de seguida é apresentado o mundo de jogo onde é mostrada a informação de desbloqueio de estádios/níveis e as taças conquistadas.

Apesar de não estar ilustrado na Figura 20, após completar todos os níveis é apresentada informação a felicitar pelo desempenho em salvar o mundo dos alienígenas. De seguida é apresentada a informação sobre a pontuação obtida e o jogador é redirecionado para o menu inicial.

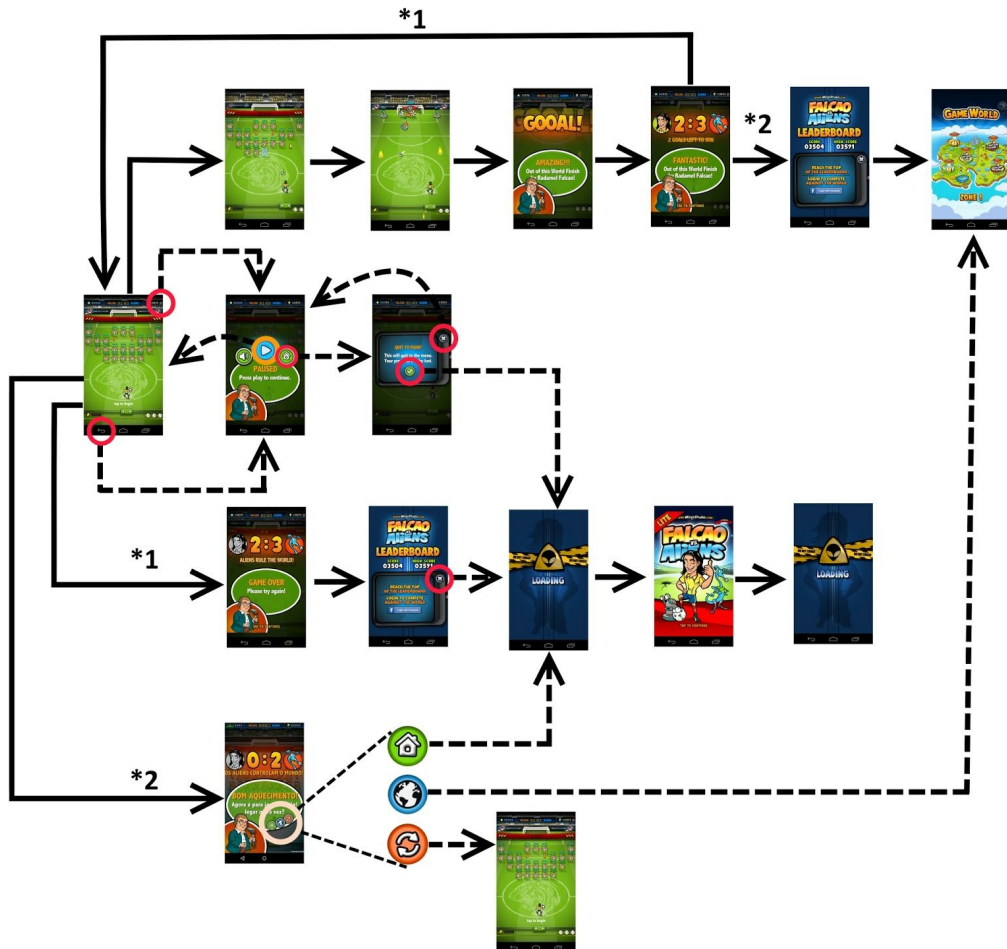


Figura 20 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo da Sessão de Jogo - Parte 3/5
 Legenda: Seta contínua- apenas há um caminho possível e a transição é automática ou através do *touch* no ecrã; Seta curva- necessário pressionar botão específico; *1- Modo Arcade; *2- Modo História.

A Figura 21 representa o fluxo originário do botão de opções. Este inicia-se no menu principal quando o jogador pressiona o botão mencionado, sendo então direcionado para o menu de opções. Neste menu é possível: ligar/desligar a música e os efeitos sonoros; ativar/desativar sincronização; efetuar *reset* a todos os dados do jogador (conquistas, pontuações, entre outros); e alterar o idioma (de momento só está disponível inglês). É também possível visualizar os créditos e voltar ao menu principal.



Figura 21 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo do Menu de Opções - Parte 4/5
 Legenda: Seta contínua- apenas há um caminho possível e a transição é automática ou através do *touch* no ecrã; Setas contínua- necessário pressionar botão específico; Seta curva- retroceder.

A Figura 22 representa o fluxo originário do botão de *achievements*, o mesmo inicia-se no menu principal quando o jogador pressiona o botão mencionado, sendo então direcionado para o menu de *achievements*. Neste menu é possível: ver as pontuações obtidas e comparar com outros jogadores (autenticação necessária com conta *Facebook*); é possível visualizar as cartas e troféus adquiridos; é possível aceder a páginas *web* externas, como por exemplo a página da empresa; por fim, é possível retornar ao menu principal.

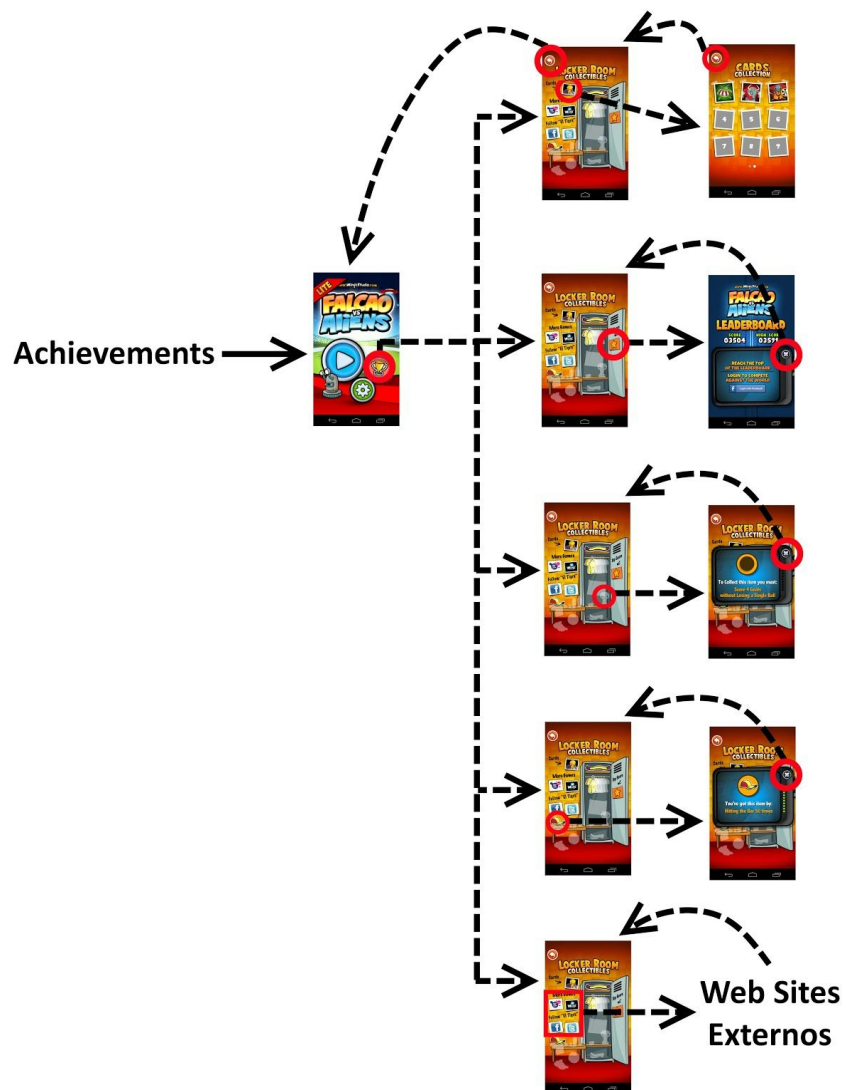


Figura 22 - Organização Detalhada do Videojogo “Falcao vs Aliens” - Fluxo do Menu de *Achievements* - Parte 5/5

Legenda: Seta contínua- apenas há um caminho possível e a transição é automática ou através do *touch* no ecrã; Setas contínua- necessário pressionar botão específico; Seta curva- retroceder.

5. Implementação, Testes e Resultados

Este capítulo está dividido em 4 subsecções: Considerações Acerca da Versão Original, Implementação, Testes e Comparação Entre Versões. Na primeira subsecção serão apresentados algumas lacunas da versão original, principalmente as que foram descobertas após a análise do código e *assets* originais. Seguidamente será detalhado todo o processo de implementação da nova versão do “Falcao vs Aliens”. Na subsecção Testes serão descritos os testes realizados durante o desenvolvimento. Por fim, será efetuada uma comparação entre a nova versão e a original, assim como também serão comparadas as diferenças apresentadas pela nova versão, entre as várias plataformas.

5.1. Considerações Acerca da Versão Original

O “Falcao vs Aliens” original apresentava-se como uma versão de demonstração, sendo apenas jogável em um único modo (arcade) e quatro dos dezoito níveis implementados. Em relação aos não jogáveis, apesar de implementados não foram testados, o que levou à necessidade de efetuar um balanceamento de dificuldade para a nova versão, resultando na reformulação da maioria dos mesmos. Além do referido, a dificuldade geral foi considerada excessiva, sendo também necessária a revisão/alteração de outros elementos como: velocidade da bola e do Falcao, probabilidade do aparecimento de *power ups*, velocidade dos disparos alienígenas e tempo de inibição de movimento do Falcao após sofrer um disparo.

A versão original apresentava ainda outras lacunas, como a falta de diversos sons e *sprites*, assim como uma animação de conclusão do videogame. Desta forma, foi necessária a procura de novos sons, a criação/edição de *sprites* e a implementação da animação mencionada. Por último, esta versão também apresentava vários *bugs*, um total de onze conhecidos, tornando-se a correção dos mesmos uma prioridade.

5.2. Fases de Implementação

A nova versão do videogame, desenvolvida no motor *Unity*, dividiu-se em duas principais fases de implementação, uma referente ao conteúdo original enquanto a outra a novas funcionalidades e melhorias. Como forma de melhorar a organização de trabalho e percepção de progresso, dividiram-se estas fases nos seguintes conjuntos de tarefas:

- Implementação do conteúdo original (com a adição de novos idiomas):
 - Principais mecânicas e jogabilidade;
 - Menus principal e de opções;
 - Menu de achievements (cacifo);
 - Coleção de cromos;
 - Sequência animada de introdução;
 - Mapa do mundo;
 - *Leaderboard*;
 - Novos idiomas;

- Implementação de novas funcionalidades/melhorias;
 - Menu de seleção de modos de jogo;
 - Modos de jogo: arcade e história;
 - Melhoria e introdução de novos *power ups*;
 - Integração com o *Facebook*;
 - Alterações à base de dados;
 - Animação de conclusão dos modos história e arcade;
 - Modo de jogo desafio (multijogador).

Estas fases serão detalhadas nas subsecções que se seguem, assim como os conjuntos de tarefas que constituem as mesmas.

5.2.1. Implementação do Conteúdo Original

Nesta primeira fase foram implementadas as funcionalidades existentes na versão original. Como mencionado anteriormente, esta fase foi dividida em oito conjuntos de tarefas, os quais serão explicados nesta subsecção.

5.2.1.1. Principais Mecânicas e Jogabilidade

O primeiro conjunto coincidiu com a parte jogável, contendo os elementos que fazem parte de um nível e jogabilidade do mesmo.

Desta forma, começou-se por criar o cenário (Figura 15), ou seja, o plano de fundo (relvado e bancadas), painéis publicitários, baliza, barreira protetora da baliza, limites do cenário e as respectivas configurações físicas, de forma detectar e tratar colisões.

Seguidamente, foi implementada a personagem principal (Figura 15, elemento 12), a qual é composta por uma base de contacto com a bola e o Falcao, tendo ambos várias animações. A base mencionada, quando entra em colisão com a bola a cor é

realçada, após algum tempo esta volta à cor original. Em relação ao Falcao, foram criadas as animações de correr e chutar. Por fim, foi também implementada a lógica relacionada com a detecção de *inputs (touch)* e controlo da personagem. No entanto, como para esta nova versão era necessário o suporte a rato e teclado, aproveitou-se para introduzir também esse suporte nesta fase.

Em seguida, implementou-se o elemento correspondente à bola de futebol (Figura 15, elemento 10), este conta com o auxílio do motor de física, tanto para a movimentação no cenário, como para as colisões com os diversos objetos. Além do referido, este possui: um sistema de partículas (simulando o rasto de movimentação), sombra, animação de rotação e a lógica. Esta última é responsável pelo comportamento do antes/depois do jogador efetuar a acção de chutar, detecção da perda de bola, variações na velocidade e cálculo do ângulo de rotação.

A próxima tarefa de implementação foi a criação dos alienígenas estáticos e dinâmicos, os quais partilham alguns elementos em comum, como a barra de energia e determinadas animações (respirar e teletransporte ao serem destruídos). No entanto, os inimigos dinâmicos e os guarda-redes, têm algumas animações e comportamentos bastante diferentes dos estáticos, nomeadamente o disparo de bolas de fogos ou raios laser, movimentação dos mesmos. Alguns estão limitados a uma determinada área enquanto outros não, podendo entrar/sair do cenário.

Uma característica da jogabilidade, é a vulnerabilidade do Falcao aos disparos alienígenas, que ao colidir com estes ocorre uma animação de queda, tendo como consequência impedir a movimentação durante alguns segundos. Portanto, essa tarefa foi realizada a seguir.

Como este videojogo foi concebido a pensar nos dispositivos móveis, contém grupos de *spritesheets* com diferentes resoluções, havendo três variações, os quais dependendo da resolução são carregados para a memória. Desta forma, criou-se um algoritmo de decisão, o qual tem como argumento a resolução nativa da plataforma em que é executado e consoante o valor da altura é escolhido o grupo com a resolução mais próxima. No caso da altura ser inferior a 768 píxeis é escolhido o grupo com resolução mais baixa, se for maior que 1536 píxeis é utilizada a variação com maior resolução, caso contrário é escolhido o grupo entre essas resoluções. Através desta abordagem é possível poupar espaço em memória e aumentar a performance em dispositivos de capacidades inferiores.

A implementação das estrelas (Figura 15, elemento 8), foi a tarefa seguinte, estas aparecem após a destruição de certos alienígenas, ficando a rodar sobre si (animação) no lugar onde eles se encontravam, no total existem três estrelas por nível.

Quando a bola entra em colisão com uma estrela, esta é capturada, ocorrendo uma animação de deslocação para a barreira protetora da baliza (Figura 15, elemento 6), deixando um rasto amarelo (sistema de partículas). Ao alcançarem a barreira, estas vão ocupando os três lugares disponíveis, sendo esta desbloqueada após a captura das três estrelas. Assim, foi também implementada a lógica de desbloqueio desta barreira, que além de desaparecer também inclui a animação do aparecimento de alguns alienígenas, como o caso de tanques e guarda-redes (Figura 16).

Posteriormente, decorreu a implementação das bebidas energéticas (Figura 15, elemento 7), estas são visíveis desde o início do nível, sendo também capturadas após colisão com a bola, executando uma animação ao deslocar-se para a barra de energia do Falcao (sistema de partículas). Assim, juntamente com as bebidas foi também implementada a barra de energia mencionada, esta enche assim que o jogador captura uma bebida, esta barra vai diminuindo ao longo de alguns segundos, ficando sem efeito ao chegar ao fim. O efeito desta consiste em proteger o Falcao de disparos alienígenas e outros efeitos negativos (como o *power up* de inversão dos controlos), este efeito é representado por uma bola de energia que rodeia a personagem principal.

O sistema de pontuação foi introduzido a seguir, desta forma adicionou-se uma barra ao topo da tela (Figura 15, elementos 1, 2 e 3) com o objetivo de mostrar duas pontuações: atual e máxima. Esta tarefa teve algumas complicações pelo facto do *Unity* não oferecer suporte a fontes animadas (através de *sprites*), o que levou à criação de um algoritmo para realizar essa função. Para se ter uma melhor percepção da pontuação, sempre que é realizada uma ação que é recompensada com pontos, estes são mostrados no local de origem e somados à pontuação atual (barra no topo da tela). Existem várias fontes de pontos, como alienígenas, *power ups* e postes da baliza, no entanto estas foram ampliadas na nova versão, passando a incluir estrelas, bebidas energéticas e bolas extra. A próxima tarefa foi a implementação do sistema de vidas (bolas restantes) e de deteção de derrota/vitória, assim como as animações envolvidas.

Seguidamente foram implementados os *power ups*, os quais surgem após a destruição de um alienígena, tendo uma probabilidade associada a essa ação. Estes *power ups* descem a tela até atingirem o limite inferior e serem removidos do cenário, se a personagem principal entrar em colisão com um *power up* é ativado o efeito do mesmo. Nesta fase, foram apenas implementados os *power ups* da versão original, totalizando sete: bolas duplas, bola extra, velocidade extra, bolas gigantes, bolas pequenas, controlos invertidos e o poder de fogo.

Após a tarefa anterior criaram-se os obstáculos, estes são objetos estáticos e indestrutíveis, possuem detecção de colisão tendo alguns destes animação de impacto. Os obstáculos existentes são: pneus, paredes e naves espaciais destruídas.

Posteriormente foram implementadas condições climáticas, através da utilização de sistemas de partículas de forma a reproduzir o efeito da queda de chuva e neve. Nem todos os níveis têm as condições climáticas ativas, tendo estas o papel de trazer uma maior variedade aos cenários.

A criação de uma nave espacial foi a tarefa que se seguiu, a qual contém várias animações: entrada no cenário, feixe de luz, largada de alienígenas e saída do cenário.

A construção de níveis tem como base informação guardada em ficheiros, os quais contêm todos os dados necessários para a criação de níveis como: posições dos vários alienígenas e bebidas energéticas, alienígenas que possuem estrelas, estado climático (chuva, neve), posição de obstáculos e se a nave espacial que larga mais inimigos está ou não ativa no nível em questão. De forma a extrair e usar essa informação para a construção dos níveis, foi desenvolvido um algoritmo de *parsing*, assim como a correspondente configuração e criação dos elementos de cada nível.

Por fim, a última tarefa deste conjunto foi a criação do menu de pausa dentro da sessão de jogo, o qual era composto pelos botões de continuar, voltar ao menu principal e ligar/desligar áudio. Estes botões contêm uma animação ao aparecer/desaparecer da tela. Neste menu existe ainda uma personagem que é um comentador de futebol, o qual tem uma animação que simula a fala e um balão onde é apresentado o texto, ambos possuem animações de entrada e saída. Além do referido, foi também implementado um sistema de notificações, o qual consiste numa televisão que entra/sai da tela, esta apresenta texto informativo ou questões, contendo normalmente dois botões: confirmar e fechar. No menu de pausa, este sistema serve para confirmar se o utilizador deseja realmente sair, após clicar no botão de voltar ao menu principal.

5.2.1.2. Menu Principal e de Opções

O segundo conjunto diz respeito à implementação do menu principal e do menu de opções, assim como os elementos que fazem parte destes e a navegação entre eles.

A primeira tarefa consistiu na criação do cenário e os vários objectos que o preenchem, ou seja, uma imagem estática como plano de fundo, uma nave espacial

com uma animação de movimento e rotação, *flashes* a simular câmeras fotográficas, o Falcao, um robô e um alienígena (ambos com animação de entrada).

Em seguida, foram implementados os botões do menu principal: iniciar jogo, opções, cacifo e hiperligação à página da WingzStudio. O sistema de notificações criado para o menu de pausa foi reutilizado, com o objetivo de confirmar se o utilizador tem a certeza que deseja sair do videojogo (após clicar na tecla “esc” ou no botão de sistema *back* em *Android*), assim como também para confirmar se deseja visitar a página da WingzStudio (ao clicar na hiperligação).

Posteriormente implementou-se o menu de opções, contendo cinco botões: ligar/desligar música, ligar/desligar som, alterar idioma, reiniciar dados e créditos. Como uma das tarefas de melhoria era a implementação de novos idiomas, aproveitou-se o facto de implementar o botão de alterar idioma para incluir parcialmente os textos nos novos idiomas (português e espanhol), ou seja, os textos existentes no conteúdo já implementado. Por fim, adicionaram-se os créditos e, as transições entre menus e sessão de jogo.

5.2.1.3. Menu de *Achievements*

O conjunto de tarefas que se seguiu foi a implementação do menu de *achievements* e a transição com o menu principal. À semelhança dos anteriores, iniciou-se pela criação do cenário, contendo o plano de fundo e os objetos que o constituem como: tapete, banco e armário. Seguidamente foram implementados os vários objetos correspondentes ao *achievements*, existindo um total de dezassete, estes têm um funcionamento semelhante ao de um botão, permitindo assim ao jogador clicar nos mesmos de forma a consultar informação de como os obter e o progresso atual. Esta informação é apresentada através do sistema de notificação criado anteriormente, no entanto este sofreu algumas alterações, pois era necessário apresentar também a imagem correspondente ao *achievement*.

Em seguida, foi implementada toda a lógica relacionada com os *achievements*, como a maioria são obtidos através de sessões de jogo, retornou-se ao primeiro conjunto de tarefas de forma a introduzir as alterações necessárias. A tarefa que se seguiu foi a implementação de hiperligações para as páginas das entidades ligadas a este videojogo (WingzStudio e *Marmalade*¹⁹), assim como as páginas do Radamel Falcao no *Facebook* e *Twitter*²⁰. Por fim, implementou-se o título deste menu e os botões de navegação para: o menu principal, a coleção de cromos e a *leaderboard*.

¹⁹ <https://www.madewithmarmalade.com>

²⁰ <https://twitter.com/falcao>

5.2.1.4. Coleção de Cromos

As próximas tarefas consistiram na criação da coleção de cromos, a qual é composta pelo plano de fundo e duas páginas de cromos. Desta forma, começou-se por criar o cenário, sendo para isso adicionada a imagem de fundo.

Seguidamente, foram introduzidos os cromos, os quais são semelhantes ao *achievements*, visto que são também clicáveis e apresentam uma notificação com informação relacionada aos mesmos. Contudo, é possível partilhar cromos no *Facebook*, sendo esta a principal diferença em relação à implementação dos *achievements*.

Após a tarefa anterior, implementou-se o sistema de páginas, tendo cada uma nove cromos, a transição entre elas é obtida arrastando com o dedo/rato. Como nesta nova versão do “Falcao vs Aliens”, o rato e o teclado serão suportados, aproveitou-se para adicionar dois botões de mudança de página, aos quais foram associadas teclas de atalho (“w”/”a” e setas de orientação horizontal), com isto espera-se facilitar a usabilidade. Por fim, foi introduzido o botão de voltar ao menu de *achievements*.

5.2.1.5. Sequência Animada de Introdução

O grupo de tarefas seguinte incidiu sobre a criação de uma sequência animada, com o objetivo de servir de introdução às sessões de jogo.

A primeira foi a criação do cenário, sendo este composto pelo plano de fundo, colinas, edifícios e um estádio. Após a criação destes elementos, seguiram-se os restantes objetos: nave espacial, alienígenas, público do estádio e o Falcao, os quais são os principais personagens desta animação.

A implementação da sequência de animações foi a tarefa que se seguiu, a qual é composta pelas seguintes: entrada da nave alienígena com direção ao estádio, desembarque de vários alienígenas, público em pânico a fugir, afastamento de câmara e destaque do Falcao em cima das colinas, este afastamento é alcançado através da escala e movimentação dos elementos.

5.2.1.6. Mapa do Mundo

As tarefas seguintes consistiram na implementação do mapa do mundo já existente na versão original, no entanto não foi implementado exatamente igual, visto que uma das novas funcionalidades implementadas foi o modo de jogo história, no qual uma das partes principais é o mapa do mundo. Desta forma, a primeira tarefa

executava foi a criação do cenário, tendo sido colocada uma imagem de plano de fundo, inúmeras nuvens (incluindo animações de movimento), mapa e elementos que o complementam como: árvores, edifícios, monumentos e estádios (Figura 23). Na versão original, cada nível correspondia a um círculo no mapa, no entanto estes não foram implementados pois iriam ser alvo de alteração quando fosse implementado o modo história. Por fim, implementou-se naves alienígenas a sobrevoar os estádios e o botão de voltar ao menu principal (inexistente na versão original).

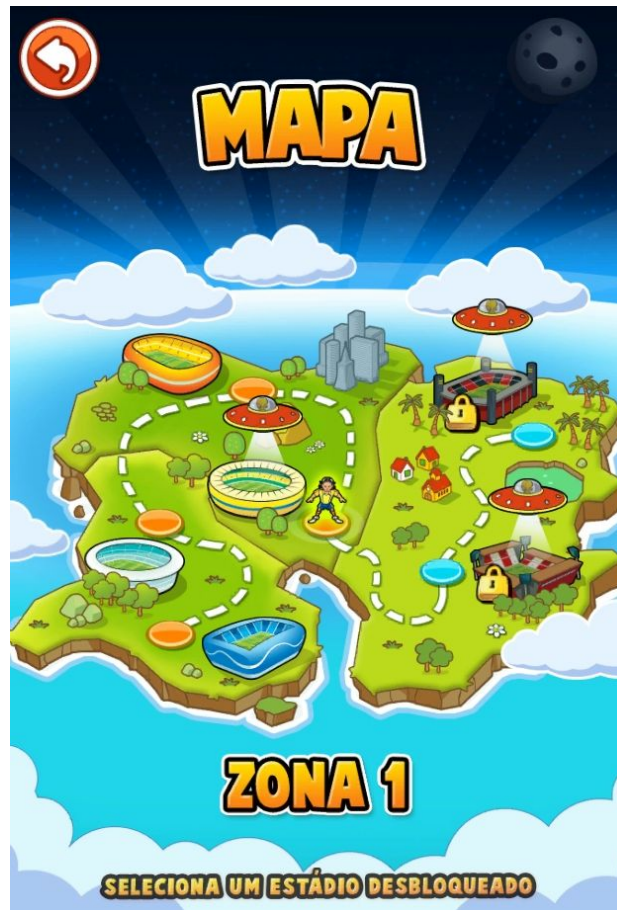


Figura 23 - Mapa do Mundo - "Falcao vs Aliens" - Nova Versão

5.2.1.7. **Leaderboard**

O último conjunto consistiu na concepção de uma *scene* onde são mostradas as pontuações do jogador e a tabela mundial de máximas pontuações. A primeira tarefa incidiu na elaboração do cenário, tendo apenas o plano de fundo. De seguida, foram colocados textos e o sistema de notificação, que neste caso serviu como contendor para a tabela de pontuações (Figura 24). Seguidamente, implementou-se o botão de autenticação no *Facebook*, sendo necessária esta autenticação para submeter a

pontuação do jogador e aceder à tabela mundial. Por fim, adicionou-se o botão de sair dessa *scene*.



Figura 24 - Leaderboard - "Falcao vs Aliens" - Nova Versão

5.2.2. Implementação de Novas Funcionalidades/Melhorias

Nesta segunda fase, foram implementadas novas funcionalidades e melhorias à versão original, estas foram divididas em sete grupos de tarefas, os quais serão detalhados nesta subseção.

5.2.2.1. Menu de Seleção de Modos de Jogo

O primeiro grupo diz respeito à criação do menu de seleção de modos de jogo (Figura 25), o qual conta com três modos à escolha: arcade, história e desafio. A primeira tarefa consistiu na construção do cenário, contendo uma imagem de fundo estática e o Falcao a aparecer através de uma animação. De seguida, foram implementados os botões, tanto os três para cada um dos modos como o botão de voltar ao menu principal, estes sofreram várias alterações ao longo do desenvolvimento, as razões que levaram a isso foram os testes realizados ao videojogo

com um grupo de pessoas que consideraram estes pouco intuitivos, estes testes serão detalhados na seção 5.3. Desta forma, além da alteração das *sprites*, foi implementada a funcionalidade de mostrar o nome do modo quando o rato/dedo estivessem por cima destes botões.



Figura 25 - Seleção de Modos de Jogo - "Falcao vs Aliens" - Nova Versão

5.2.2.2. Modos de Jogo Arcade e História

As tarefas que se seguiram consistiram na implementação do funcionamento dos modos de jogo, arcade e história. Para o primeiro modo não foram necessárias muitas alterações, visto que este se mantém semelhante ao da versão original, tendo apenas sido necessário implementar a navegação do menu de seleção de modos e a transição de níveis.

Na versão original, ao completar um nível do modo arcade era mostrado o mundo de jogo e o Falcao a avançar para o próximo nível antes deste ser carregado, nesta nova versão isso não acontece, sendo feita logo a transição sem animação nenhuma, esta decisão foi tomada devido ao facto do modo história já incluir este mapa.

No caso do modo história houve a necessidade de efetuar inúmeras alterações no mapa do mundo, a primeira foi torná-lo num menu de seleção de estádios e níveis, assim como a apresentação do progresso e taças conquistadas através da pontuação obtida por nível.

A primeira tarefa a ser realizada para este modo foi tornar os estádios clicáveis, sendo implementado um submenu contendo botões para três níveis (Figura 26), o qual abre ao clicar num estádio desbloqueado (é necessário completar os três níveis do estádio anterior para desbloquear o seguinte). De seguida, implementou-se a navegação no mapa de jogo e deste para os vários níveis, bastando clicar no estádio pretendido e de seguida no nível desejado.

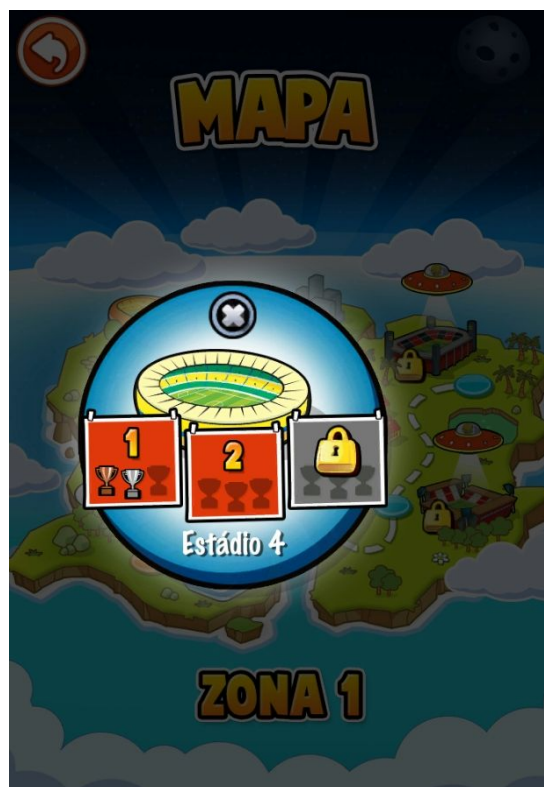


Figura 26 - Submenu do Estádio 4 - Mapa do Mundo - “Falcao vs Aliens” - Nova Versão

Posteriormente, implementou-se as animações da conquista de taças (havendo três: ouro, prata e ouro) e de desbloqueio de estádios, sendo esta última a animação mais complexa, envolvendo um sistema de partículas a simular fogo de artifício, a nave alienígena a abandonar o estádio, a movimentação do Falcao para o próximo estádio, o desaparecimento do aloquete e o clarear desse estádio. Após estas animações o submenu do estádio desbloqueado é aberto de forma a que o jogador perceba que tem acesso a novos níveis.

Terminadas as alterações ao mapa, foi ainda necessário reutilizar a *scene* das pontuações, de forma a que no final de cada nível seja apresentada a pontuação obtida, assim como a pontuação necessária para adquirir cada taça (Figura 27). O início de cada nível também sofreu alterações (Figura 27), sendo apresentadas as pontuações necessárias para cada taça, as mesmas também são apresentadas no menu de pausa. Por fim, foi ainda necessário introduzir novos botões tanto no menu de pausa como na *leaderboard*, os botões adicionados foram: reiniciar nível, sair para o menu principal e sair para o mapa do mundo (Figura 27).



Figura 27 - *Leaderboard* (à esquerda) e Início de Nível (à direita) - Modo História - “Falcao vs Aliens” - Nova Versão

5.2.2.3. Melhoria e Introdução de Novos *Power Ups*

O grupo de tarefas seguinte consistiu na melhoria do sistema de *power ups* e implementação de dois novos *power ups*: dano extra e restauro das bebidas energéticas. Os originais, tinham uma probabilidade fixa de aparecimento (30%), no entanto caso fosse o da bola extra, este só poderia surgir caso o jogador não tivesse bolas extra. A primeira melhoria incidiu sobre a probabilidade, a qual passou a variar consoante o nível, nos primeiros esta tem o valor de 25% e vai aumentando

progressivamente até atingir o valor máximo de 65%, a razão que levou a esta melhoria foi o facto de nos níveis avançados, os alienígenas demorarem demasiado tempo a serem destruídos, resultando em momento monótonos. De forma a aumentar a variedade e utilidade dos *power ups*, introduziram-se novas condições:

- Um *power up* não poderá surgir se o efeito do mesmo já estiver ativo, por exemplo caso as bolas estejam pequenas, não irão aparecer *power ups* que causem esse efeito;
- Impossibilidade de aparecerem consecutivamente dois *power ups* iguais.

A última melhoria efetuada foi permitir também o aparecimento do *power up*, bola extra, quando o jogador ainda tem à disposição uma vida, esta alteração teve como origem o facto da maioria das pessoas que testaram o videojogo, não se aperceberem da existência deste *power up*, devido à sua raridade em aparecer.

Em relação aos novos *power ups*, o primeiro (dano extra) levou à criação de um efeito a simular uma esfera de energia verde em torno da bola, para isso utilizou-se um sistema de partículas. O segundo restaura as bebidas energéticas, colocando-as nas posições iniciais. A decisão que levou à adição destes novos *power ups*, teve como base o facto dos jogadores considerarem que após algumas sessões de jogo, os *power ups* tornavam-se repetitivos.

5.2.2.4. Integração com o Facebook

A integração com o Facebook foi efetuada através da API disponibilizada para Unity, utilizando os métodos de autenticação, recolha de informações do perfil (primeiro e último nome, data de nascimento, email, lista de amigos, localização, foto de perfil e identificador), partilha de conteúdo (cromos e pontuações) e envio de convites a amigos (modo desafio). Além das alterações para a utilização desses métodos, foi necessário também adicionar novas notificações:

- **Menu principal:** caso o utilizador nunca tenha efetuado autenticação, ao clicar para continuar na primeira tela, é apresentada uma notificação a questionar se o deseja fazer.
- **Menu de seleção de modos de jogo:** ao clicar no modo desafio, caso não tenha sido feita a autenticação no Facebook, é apresentada uma notificação para o fazer, caso não o faça não poderá jogar este modo.

5.2.2.5. Alterações à Base de Dados

A versão original, apenas continha três tabelas: “user”, “user_friendship” e “highscores”. A primeira guarda os dados pessoais de cada jogador, como primeiro e último nome, data de nascimento, *email*, nome traduzido (certos nomes extraídos do *Facebook* apresentam caracteres especiais, como por exemplo nomes japoneses, sendo necessária a tradução dos mesmos), sexo e localização, assim como um identificador único. A “user_friendship” contém dois campos, sendo eles identificadores da tabela “user”, mantendo um registo de quais utilizadores são amigos entre si. Por último, a tabela “highscores” armazena as pontuações obtidas pelos vários jogadores, esta é constituída pelos seguintes campos: identificador única de cada entrada, identificador de utilizador e a pontuação.

Tendo em conta as novas funcionalidades foi necessário alterar a base de dados, de forma a suportar tabelas de pontuações por nível, sincronização e o modo desafio. Das tabelas já existentes foi necessário alterar a tabela das pontuações, de forma a introduzir um novo campo “level”, correspondendo ao nível do modo história em que a pontuação foi obtida, no caso do modo arcade este campo tem valor -1. Foi também necessária a criação de duas novas tabelas:

- **“User_preferences”**: que contém os dados relativos ao progresso do jogador, para isso como campos possui um identificador unico, identificador do utilizador, nome e valor da chave. Estas chaves, são responsáveis por guardar inúmeras informações, tais como progresso dos *achievements*, máximo nível atingido, número de vitórias/derrotas, pontuações do modo arcade e de cada nível do modo história.
- **“Challenges”**: responsável por guardar a informação relativa aos vários desafios realizados e em progresso. Esta tabela conta com um identificador único, dois identificadores de utilizadores, data do desafio, estado atual do desafio (à espera de aceitação, terminado ou cancelado) e duas pontuações correspondentes à de cada jogador.

Paralelamente a este documento foram elaborados dois diagramas de entidade e relacionamento, os quais representam as bases de dados original e da nova versão, estes podem ser consultados no Anexo D.

5.2.2.6. Animação de Conclusão dos Modos Arcade e História

O próximo agrupamento de tarefas consistiu na criação de uma sequência animada, com o objetivo de dar um desfecho aos modos arcade e história. Desta

forma, decidiu-se reutilizar a animação de introdução de forma invertida, ou seja, em vez de uma nave transportar alienígenas para o estádio, esta trata de os transportar para fora do mesmo, dando a entender ao jogador que estes seres estão a fugir, graças ao jogador ter salvo o planeta.

Após a nave sair, são mostradas pessoas a celebrar com cânticos e fogos de artifício, seguindo-se um efeito de afastamento de câmara (através da escala e movimentação dos objetos do cenário) e um alienígena é destacado em cima da colina, juntamente com um música de suspense.

5.2.2.7. Modo de Jogo Desafio

O último conjunto de tarefas incidiu na criação do modo de jogo desafio (Figura 28), no entanto a implementação do mesmo não foi finalizada até à data da escrita deste documento. O principal motivo da não conclusão deste modo, foi o surgimento de novas tarefas com maior prioridade, tanto por iniciativa da WingzStudio como por sugestões resultantes de testes efetuados por grupos de pessoas, esses testes serão abordados com maior detalhe na próxima secção.



Figura 28 - Menu Modo Desafio - “Falcao vs Aliens” - Nova Versão

Além dos motivos mencionados, existiu também uma falta de consenso em relação ao funcionamento deste modo, porque alguns elementos da equipa queriam

que fossem criados níveis diferentes, de forma a que não fosse revelado o conteúdo dos restantes modos, pois poderia provocar a falta de interesse em jogá-los. Além do referido, foi sugerida ainda outra abordagem, a qual limitaria a escolha dos níveis ao jogador, podendo apenas escolher aqueles que já tivessem sido desbloqueados no modo história, o problema desta solução é que também iria limitar o número de amigos que poderiam ser desafiados, visto que ambos teriam que ter o nível em causa desbloqueado.

Apesar deste modo de jogo ter sido incluído nos requisitos e planeamento deste projeto, este foi considerado secundário pela WingzStudio, preferindo esperar para observar a aceitação desta nova versão e posteriormente, caso seja viável, introduzir este novo modo. Contudo, foram realizadas algumas tarefas de implementação, como a criação do menu de desafio (cenário, botões e parte da lógica) (Figura 28), formatação das mensagens *JSON* a serem trocadas com o servidor e as alterações à base de dados (mencionadas anteriormente).

5.3. Testes

Ao longo da implementação desta nova versão, foram efetuados diversos testes informais, tendo a maioria sido realizada internamente pelos elementos da WingzStudio. No entanto, para uma pequena parte dos testes usou-se um grupo de pessoas externas, com idades entre os 10 e 31 anos, das quais apenas duas do sexo feminino. O nível de experiência em videojogos apresentado por estas pessoas era bastante diferente, assim como a frequência de utilização dos mesmos, o que permitiu obter uma opinião mais diversificada.

O primeiro teste realizado com este grupo foi efetuado após a conclusão da implementação das mecânicas principais e do menu principal. Este teve como principal objetivo a avaliação da dificuldade geral e individual dos níveis, para isso pediu-se que fosse classificado, numa escala de 0 a 10, o grau de dificuldade para cada um dos níveis. Do grupo apenas duas pessoas ultrapassaram o quarto nível, tendo estas atingido os níveis 11 e 8 respectivamente (Figura 29). Em relação às restantes, uma não conseguiu completar o nível 2 e as outras obtiveram derrota no quarto nível. Desta forma, não foi possível uma avaliação da totalidade dos níveis (18), sendo indicativo de uma dificuldade excessiva, principalmente porque apenas 20% do grupo conseguiu completar um terço dos níveis (Figura 29).

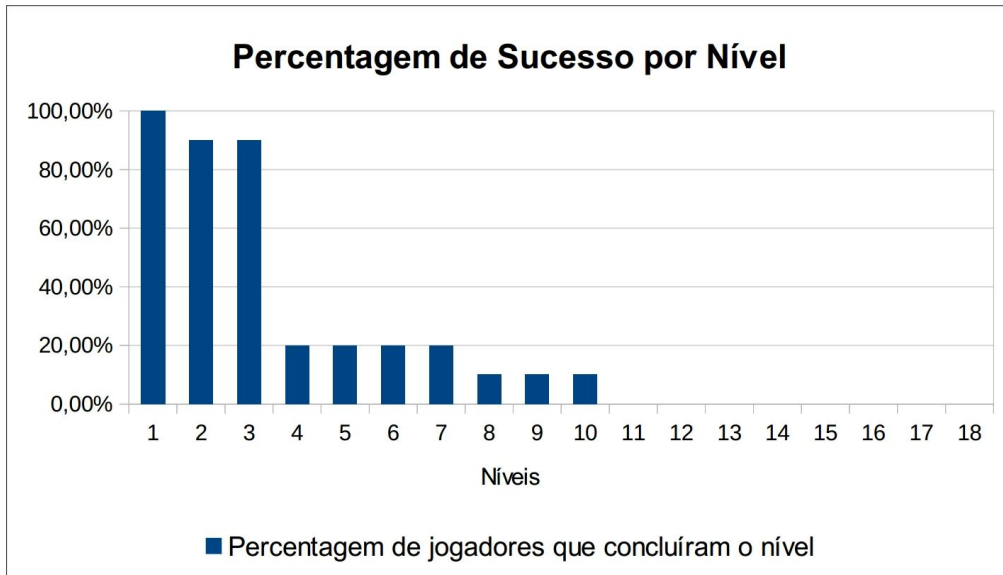


Figura 29 - Gráfico de Percentagem de Sucesso por Nível - 1º Teste “Falcao vs Aliens” - Versão de Desenvolvimento

De seguida, no Gráfico 2 é a apresentada a média da classificação de dificuldade, atribuída pelos jogadores em cada um dos níveis testados.

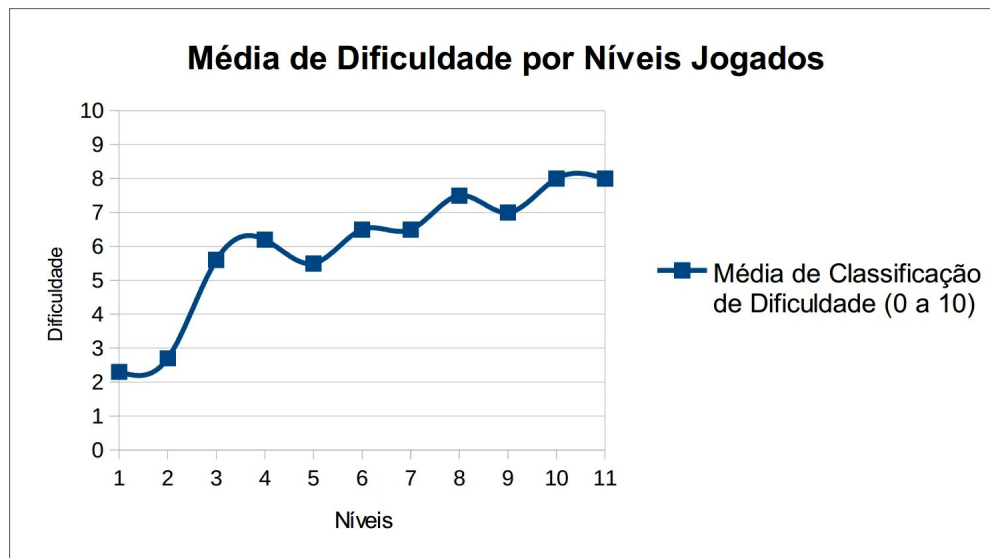


Figura 30 - Gráfico de Classificação da Dificuldade por Níveis - 1º Teste “Falcao vs Aliens” - Versão de Desenvolvimento

Analisando o gráfico da Figura 30, percebe-se que a curva de dificuldade não está de acordo com o esperado, havendo situações em que o nível anterior apresenta maior dificuldade que o seguinte, assim como um aumento exagerado de dificuldade como acontece do segundo para o terceiro nível.

Devido aos resultados obtidos neste primeiro teste, analisaram-se todos os níveis do videojogo com o intuito de diminuir o grau de dificuldade e o balanceamento do mesmo consoante se avança nos níveis, sendo alterados nove dos dezoito existentes. Após estas alterações, foram efetuados testes internos com a equipa da WingzStudio, concluindo que o videojogo ainda apresentava demasiada dificuldade, desta forma foram revistos os seguintes aspetos: velocidade da bola, tempo de inibição de movimento, probabilidade e velocidade de queda de *power ups*, intensidade e rapidez de disparos, movimentação do Falcao e duração da proteção resultante de bebidas energéticas.

Concluídas as alterações à dificuldade, repetiu-se o mesmo teste com o mesmo grupo de pessoas, com o intuito de comprovar se o objetivo de diminuir a dificuldade tinha sido cumprido. Neste teste a maioria, oito das dez pessoas, conseguiu atingir o nível 7, enquanto que das restantes, uma atingiu o nível 5 e a outra ficou pelo terceiro (Figura 31). Contudo, o nível máximo atingido passou a ser 13, tendo sido atingido por uma pessoa, enquanto que outras duas atingiram os níveis 8 e 11 respectivamente (Figura 31). Estes resultados confirmaram que houve uma diminuição significativa no grau de dificuldade, no entanto os inquiridos sugeriram o aumento da variedade de *power ups*, até porque a maioria dos que apareciam eram considerados inúteis à situação de jogo.



Figura 31 - Gráfico de Percentagem de Sucesso por Nível - 2º Teste “Falcao vs Aliens” - Versão de Desenvolvimento

Seguidamente, na Figura 32 será apresentada a média da classificação de dificuldade, atribuída pelos jogadores em cada um dos níveis testados neste segundo teste.

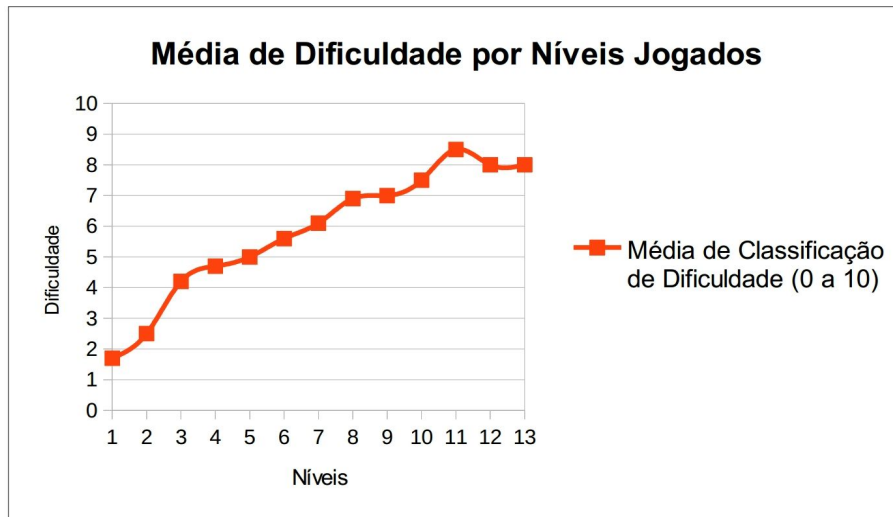


Figura 32 - Gráfico de Classificação da Dificuldade por Níveis - 2º Teste “Falcao vs Aliens” - Versão de Desenvolvimento

Após observar os resultados obtidos neste segundo teste, verifica-se que a dificuldade de facto diminuiu, assim como uma melhoria na curva de dificuldade ao longo dos diversos níveis (Figura 32), tendo esta um crescimento menos acentuado, comprovando o sucesso do balanceamento dos mesmos. De forma, a ter uma melhor percepção da diferença entre os dois testes, são apresentados dois gráficos com a comparação entre os resultados obtidos nos testes realizados (Figuras 33 e 34).

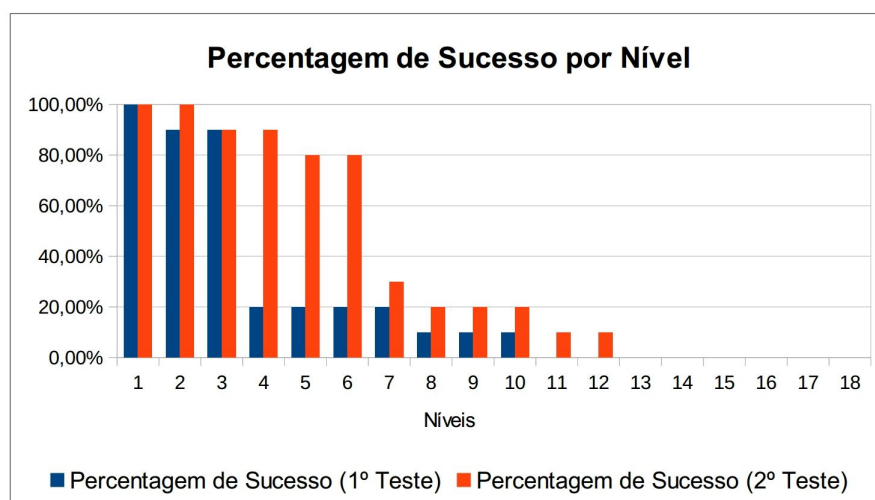


Figura 33 - Gráfico de Percentagem de Sucesso por Nível - Comparação entre o 1º e 2º Testes “Falcao vs Aliens” - Versão de Desenvolvimento

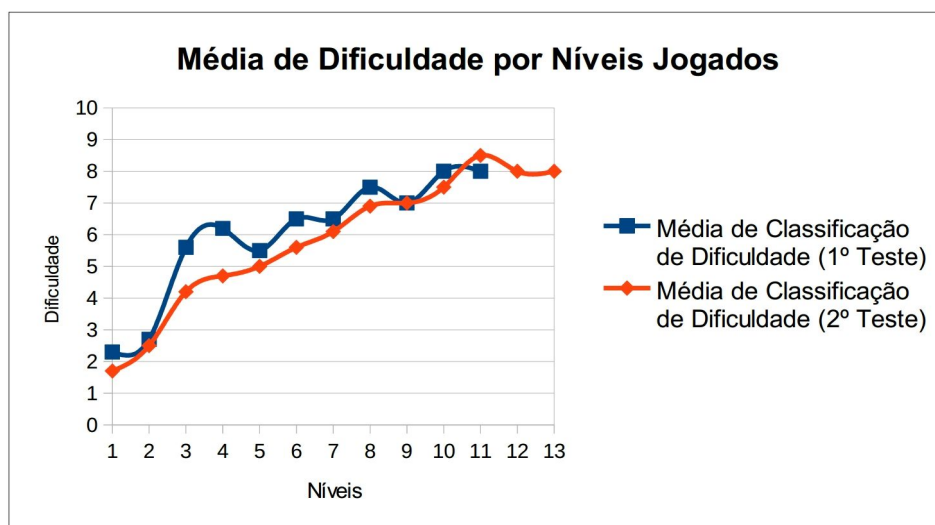


Figura 34 - Gráfico de Classificação da Dificuldade por Níveis - Comparação entre o 1º e 2º Testes “Falcao vs Aliens” - Versão de Desenvolvimento

Como é visível nos gráficos das Figuras 33 e 34, ocorreu uma melhoria significativa tanto no sucesso dos níveis como na sua curva de dificuldade. Considerando também as sugestões recolhidas com o teste anterior, foram implementados dois novos power ups e reformulado o sistema de aparecimento dos mesmos, esta tarefa de implementação foi detalhada anteriormente na subsecção 5.2.2.3.

Por fim, foi realizado mais um teste com o mesmo grupo de pessoas, este decorreu na fase final da implementação, contendo o modo história e as alterações aos *power ups*. Neste teste o principal objetivo era obter a opinião do grupo acerca do modo história, pois uma das críticas apontadas em testes anteriores foi o facto do modo arcade obrigar a começar do primeiro nível quando se iniciava uma nova sessão, impossibilitando os jogadores de atingir os últimos níveis e tornava a jogabilidade extremamente repetitiva. As opiniões recolhidas mostraram que todos os elementos do grupo ficaram satisfeitos com o modo história, tendo atingido mais facilmente níveis avançados, exemplo que o comprova é o facto de todos os jogadores terem ultrapassado o oitavo nível, algo que não se tinha observado nos testes anteriores. Além do referido, como neste novo modo é necessário atingir determinada pontuação por nível para ganhar taças (ouro, prata e bronze), os inquiridos afirmaram que o videojogo tornou-se mais cativante e desafiante. No entanto, foram também recolhidas algumas críticas, tais como:

- O balanceamento das pontuações necessárias para ganhar as taças não se encontrava coerente, havendo níveis em que era extremamente difícil atingir a pontuação para a taça de ouro, enquanto que noutros era trivial;
- Ícones dos modos de jogo não eram intuitivos, sendo difícil para o jogador saber qual botão correspondia a determinado modo.

Com os resultados obtidos no terceiro teste, decidiu-se rever todas as pontuações necessária a cada taça, sendo estas posteriormente testadas internamente pelos elementos da WingzStudio. Os ícones dos botões dos modos de jogo foram também revistos, resultando na alteração dos mesmos, na Figura 35 é possível comparar estes ícones, foi também adicionado uma nova funcionalidade, a qual consiste em mostrar o nome do modo ao passar o rato/dedo por cima dos botões.

Comparação dos Ícones do Menu de Seleção de Modos de Jogo



Figura 35 - Comparação dos Ícones do Menu de Seleção de Modos de Jogo - “Falcao vs Aliens”

Como foi possível observar ao longo desta secção várias melhorias foram feitas no videojogo graças aos testes realizados, porém durante os mesmos alguns elementos ainda não estavam presentes, como o modo desafio, desta forma a WingzStudio planeia efetuar mais testes antes do lançamento deste produto, um dos quais durante o mês de Setembro do corrente ano, que terá como público alvo adolescentes entre os 12 e 18 anos.

5.4. Comparação Entre Versões

Nesta subsecção serão apresentadas as diferenças entre a versão original e a versão resultante deste projeto. Além do referido, serão também comparadas as diferenças que esta nova versão apresenta nas várias plataformas suportadas: nativas

(*Windows, Linux e Mac OS*), móveis (*Android, iOS e Windows Phone*) e *web* (*Facebook*).

5.4.1. Versão Original vs Nova Versão

A versão original oferecia apenas um modo de jogo, arcade, sendo o mesmo alterado nesta nova versão de forma a diminuir os tempos mortos, pois como este modo não permite a continuação após sair do mesmo, considerou-se desnecessária a animação de mudança de nível, passando esta a ser direta. Foram também adicionados dois novos modos, história e desafio, os quais foram detalhados na subsecção 5.2.2.

Em relação aos menus que fazem parte do videojogo original, estes sofreram as seguintes alterações:

- **Menu principal:** adicionou-se a possibilidade de clicar no endereço para a página da WingzStudio, melhoria da animação da nave alienígena, alteração do ícone do menu de *achievements*, textos traduzidos nos novos idiomas (português e espanhol) e a adição de sete novas notificações, as quais relacionadas com *Facebook* e comunicação com o servidor.
- **Menu de opções:** textos traduzidos nos novos idiomas, introdução do botão para ativar/desativar a sincronização e adição de quatro notificações.
- **Coleção de Cromos:** textos traduzidos nos novos idiomas, adição de dois novos botões para mudança de página, introdução de duas novas notificações.
- **Menu de pausa:** textos traduzidos e adição de dois botões (reiniciar e sair para o mundo de jogo).
- **Leaderboard:** adicionou-se a possibilidade de clicar no endereço para a página da WingzStudio, criação de três novos botões (reiniciar, sair para o menu principal e voltar ao mundo de jogo), textos traduzidos, adição das pontuações necessárias para atingir determinada taça e animações das mesmas, quatro novas notificações relacionadas com *Facebook* e comunicação com o servidor.

A todos os menus restantes, foram também adicionadas as traduções para os novos idiomas. Contudo, foram também adicionados dois novos menus: seleção de modos de jogo e o mapa do mundo, os quais foram detalhados nas subsecções 5.2.2.1 e 5.2.2.2.

Em relação ao áudio presente no videojogo original, este sofreu inúmeras alterações e adições, totalizando 1 música, 16 novos sons e 6 alterados. O mesmo ocorreu com as imagens, tendo sido alteradas as seguintes: sombra do porco alienígena, plano de fundo de dois estádios, remoção de um contorno verde existente na bola, sombra dos robôs ajudantes (efeito do *power up* “poder de fogo”), ícone do botão para o cacifo e o mapa do mundo, o qual contém inúmeras alterações as quais foram descritas em 5.2.2.2. Também foram adicionadas novas imagens, principalmente para os modos de jogo que foram introduzidos, tendo sido detalhadas na secção 5.2.2.

A maioria dos níveis presentes na versão original foram reformulados devido à curva de dificuldade não estar de acordo com o desejado, desta forma alteraram-se os seguintes: 5, 6, 7, 11, 12, 13, 14, 15, 17 e 18.

Como referido na subsecção 5.3, houve vários elementos da jogabilidade que também foram revistos de forma a diminuir a dificuldade: velocidade da bola, tempo de inibição de movimento, probabilidade e velocidade de queda de *power ups*, intensidade e rapidez de disparos, movimentação do Falcao e duração da proteção resultante de bebidas energéticas.

Nesta nova versão foram também adicionadas novas fontes de pontos, algumas das quais penalizam o jogador subtraindo um determinado valor à pontuação atual, algo que não acontecia na versão original. As fontes adicionadas foram: perda de vidas (-100 pontos), bebidas energéticas (5 pontos), estrelas (25 pontos), ação do Falcao sofrer um disparo alienígena (-5 pontos) e vidas restantes no final de cada nível do modo história (50 pontos).

O videojogo original apresentava alguns *bugs* (11), tal como foi referido na subsecção 5.1, tendo estes sido corrigidos para a nova versão.

Por fim, em termos de plataformas, além das suportadas na versão original (*Android*, *iOS* e *Windows Phone*), foram adicionadas as seguintes: *Windows*, *Linux*, *Mac OS*, *Facebook* e navegadores atuais. No entanto, como esta nova versão foi implementada no motor *Unity*, existem outras plataformas para as quais é possível compilar, como *Tizen*, *BlackBerry*, *Samsung TV* e diversas consolas, contudo estas não foram testadas.

Seguidamente é apresentada uma tabela (Tabela 4), onde são mostradas as diferenças entre a versão original e a nova:

	Versão Original	Nova Versão
Integração Facebook	Sim	Sim *1
Sincronização	Não	Sim
Total de Níveis	18	18
Total de Níveis Jogáveis	4	18
Total de Power Ups	7	9
Plataformas Suportadas:		
Android	Sim	Sim
iOS	Sim	Sim
Windows Phone 8	Sim	Sim
Windows	Não	Sim
Linux	Não	Sim
Mac OS X	Não	Sim
Navegadores (HTML5)	Não	Sim
Facebook	Não	Sim
Áudio		
Total de Sons	79	95
Total de Músicas	5	6
Modos de Jogo		
Modo Arcade	Sim	Sim
Modo História	Não	Sim
Modo Desafio	Não	Sim *2
*1 – Não disponível para plataformas nativas.		
*2 – Em desenvolvimento.		

Tabela 4 - Comparação entre a versão original e a nova - “Falcao vs Aliens”

Analisando a tabela é visível uma diferença bastante considerável entre as duas versões, além da nova versão completar o que era pretendido atingir com a versão original, esta ainda oferece um conjunto de novas funcionalidades, tornando-se numa experiência mais completa e diversificada.

5.4.2. Comparação da Nova Versão entre Plataformas

A nova versão oferece suporte a várias plataformas, contudo esta apresenta algumas diferenças consoante a plataforma em que está a ser executada, as quais serão apresentadas nesta subsecção.

O videojogo oferece integração com o *Facebook*, como autenticação, partilha e envio de convites. No entanto, esta integração não está disponível nas plataformas nativas (*Windows*, *Linux* e *Mac OS*) devido ao facto da *API* disponibilizada pelo

Facebook, para o motor *Unity*, não oferecer suporte a estas plataformas. Desta forma, seria necessário efectuar a integração de forma manual, obrigando à inclusão dos campos de autenticação. Contudo a empresa considera as plataformas nativas pouco relevantes, afirmando que provavelmente não será efetuado o lançamento do videojogo nas mesmas, pelo que preferiram que o tempo de desenvolvimento fosse dedicado a outras tarefas. Como é necessária a autenticação no *Facebook*, para ter acesso ao modo desafio, tabelas de pontuação e a funcionalidade de sincronização, estes também não estão disponíveis para as plataformas nativas.

Ao executar o videojogo diretamente no *Facebook*, faz com que a autenticação com o mesmo seja automática, pelo que os botões para efetuar esta tarefa não estão visíveis, assim como as respetivas notificações.

Como as plataformas oferecem resoluções diferentes, certos elementos são ou não visíveis baseados no *aspect ratio*, desta forma nos dispositivos móveis não são visíveis as barras laterais e o botão de modo janela/ecrã inteiro. Outra diferença é a barra de pontuação durante uma sessão de jogo, a qual é sempre visível em plataformas nativas, navegadores atuais e dispositivos móveis em que o valor da largura dividida pela altura é inferior a 0,6.

Por último, os controlos também variam consoante a plataforma, pelo que nos dispositivos móveis só é suportado *touch*, enquanto que nas restantes são suportados rato, teclado e *touch*.

De seguida é apresentada uma tabela (Tabela 5), onde são comparadas as principais diferenças, que a nova versão apresenta nas várias plataformas suportadas:

	Plataformas Nativas	Dispositivos Móveis	Navegadores e Facebook
Integração Facebook	Não	Sim	Sim
Sincronização	Não	Sim	Sim
Controlos Suportados:			
Touch	Sim	Sim	Sim
Rato	Sim	Não	Sim
Teclado	Sim	Não	Sim
Modos de Jogo			
Modo Arcade	Sim	Sim	Sim
Modo História	Sim	Sim	Sim
Modo Desafio	Não	Sim	Sim

Tabela 5 - Comparação da nova versão nas várias plataformas - “Falcao vs Aliens”

Analisando a tabela são visíveis algumas diferenças entre os conjuntos de plataformas, no entanto somente o de plataformas nativas ficou limitado, visto que este

não conta com duas funcionalidade e um modo de jogo. Em relação aos dispositivos móveis, apesar destes não suportarem rato e teclado, em nada afeta a experiência de jogo, pois é possível efetuar todas as acções através de *touch*.

6. Conclusão e Trabalho Futuro

O trabalho realizado durante este estágio permitiu criar uma versão melhorada do videogame “Falcao vs Aliens”, contendo novas funcionalidades assim como a capacidade de executar num número maior de plataformas. Desta forma, considera-se que este projeto foi bastante produtivo, pois a nova versão apresenta-se muito superior em relação à original, a qual apresenta inúmeras lacunas, sendo estas a principal razão pela baixa taxa de retenção de jogadores. Com as melhorias efetuadas e tendo em conta a opinião de quem experimentou esta nova versão, é esperado que tanto o número de jogadores como a taxa de retenção dos mesmos aumente consideravelmente após o lançamento.

A WingzStudio não chegou a um consenso sobre o rumo do modo desafio, pelo que a implementação do mesmo ainda não se encontra finalizada. Contudo o lançamento desta nova versão do videogame, já seria possível, pois está planeado que o modo desafio só seja disponibilizado numa fase posterior ao lançamento. Esta versão de momento já oferece suporte a às seguintes plataformas: *Android, iOS, Windows Phones, Windows, Linux, Mac OS X*, navegadores atuais e *Facebook*. Oferece também novas funcionalidades, como o modo história, sincronização entre dispositivos e integração completa com o *Facebook*, que é atualmente a rede social com mais utilizadores [7].

Entretanto, a empresa espera por uma melhor oportunidade para o publicar, pois como este está associado ao jogador de futebol Radamel Falcao, o qual transferiu-se recentemente para um novo clube (*Chelsea Football Club*²¹), aguarda-se por um momento em que este ganhe maior popularidade. Uma versão com o tema “dia das bruxas” está também a ser planeada, tendo como base o produto resultante deste projeto, demonstrando a confiança que a empresa tem no mesmo.

A experiência de uso do motor *Unity* no desenvolvimento desta versão do “Falcao vs Aliens” foi positiva, demonstrando-se uma escolha satisfatória. Este forneceu a maioria das funcionalidades necessárias e mesmo em situações que isso não ocorreu, possibilitou que as mesmas fossem criadas e integradas no mesmo.

Como trabalho futuro será implementado um novo nível de jogo, o qual servirá de desfecho do modo história, este apresentará diversas novidades comparado aos níveis já presentes, pois o objetivo principal do mesmo será a destruição de uma

²¹ <https://www.chelseafc.com>

máquina de guerra alienígena, a qual ocupará grande parte do cenário. Além deste, serão também adicionados novos *power ups* e *achievements*, os quais ainda estão em fase de planeamento. Por fim, será também finalizada a implementação do modo desafio.

Em suma, apesar de nem todas as melhorias planeadas estarem finalizadas até à data, o produto resultante deste estágio foi satisfatório, embora não tenha sido possível analisar a receptividade do mesmo, pois este ainda não se encontra disponível ao público. No entanto, os resultados obtidos com este estágio são promissores em relação ao potencial sucesso do videojogo.

7. Referências

- [1] American Physical Society Sites. This Month in Physics History: October 1958: Physicist Invents First Video Game. Disponível em: <http://www.aps.org/publications/apsnews/200810/physicshistory.cfm>. Acesso em: 1 de Setembro de 2015.

- [2] The First Video Game? Disponível em: <http://www.bnl.gov/about/history/firstvideo.php>. Acesso em: 1 de Setembro de 2015.

- [3] Wilson, Craig. How video games shot their way to the top. 13 de Dezembro de 2011. Disponível em: <http://www.techcentral.co.za/how-video-games-shot-their-way-to-the-top/28060/>. Acesso em: 1 de Setembro de 2015.

- [4] The Economist. All the world's a game. 10 de Dezembro de 2011. Disponível em: <http://www.economist.com/node/21541164>. Acesso em: 1 de Setembro de 2015.

- [5] Richardson, Joshua. What Is Gained From Video Games. 30 de Janeiro de 2010. Disponível em: <https://8adlemon.wordpress.com/research-paper/>. Acesso em: 1 de Setembro de 2015.

- [6] Página web oficial da WingzStudio. Disponível em: <http://www.wingzstudio.com/>. Acesso em: 22 de Janeiro de 2015.

- [7] Top 15 Most Popular Social Networking Sites. Janeiro de 2015. Disponível em: <http://www.ebizmba.com/articles/social-networking-websites>. Acesso em: 1 de Setembro de 2015.

- [8] Página oficial no Facebook do jogador de futebol Radamel Falcao. Disponível em: <https://www.facebook.com/pages/Falcao/221940704484621?fref=ts>.

Acesso em: 1 de Setembro de 2015.

- [9] Lopes, Ricardo F. G. Mobile Game Development. 2013. 114 f. Dissertação (Mestrado em Engenharia Informática) - Departamento de Engenharia Informática, FCTUC, Universidade de Coimbra, Coimbra.
- [10] Polished Port.
Disponível em: <http://tvtropes.org/pmwiki/pmwiki.php/Main/PolishedPort>.
Acesso em: 1 de Setembro de 2015.
- [11] Moss, Richard. The life of a porting house. 5 de Janeiro de 2014.
Disponível em:
<http://www.eurogamer.net/articles/2014-01-05-the-life-of-a-porting-house>.
Acesso em: 1 de Setembro de 2015.
- [12] Porting Disaster.
Disponível em: <http://tvtropes.org/pmwiki/pmwiki.php/Main/PortingDisaster>.
Acesso em: 1 de Setembro de 2015.
- [13] Hardwidge, Ben. The Problem With Porting Games. 18 de Agosto de 2009.
Disponível em:
<http://www.bit-tech.net/gaming/2009/08/18/the-problem-with-porting-games/1>. Acesso em: 1 de Setembro de 2015.
- [14] Wawro, Alex. What exactly goes into porting a video game? BlitWorks explains. 13 de Agosto de 2014. Disponível em:
http://www.gamasutra.com/view/news/222363/What_exactly_goes_into_porting_a_video_game_BlitWorks_explains.php.
Acesso em: 1 de Setembro de 2015.
- [15] Davison, Pete. The Life of a Porting House: Eurogamer's Richard Moss delves into the wild world of Mac gaming. 1 de Julho de 2014.
Disponível em: <http://www.usgamer.net/articles/the-life-of-a-porting-house>.
Acesso em: 1 de Setembro de 2015.
- [16] Till, John. Six Tips for Developing Easily Ported Games. 7 de Setembro de 2010. Disponível em:

https://software.intel.com/sites/default/files/m/7/5/e/f/1/30212-6_Tips_for_Porting_Games.pdf. 1 de Setembro de 2015.

- [17] Summer, Gabe. Why Most Developers Should Avoid Apple's New Swift Language. 5 de Junho de 2014. Disponível em: http://www.huffingtonpost.com/gabe-sumner/why-most-developers-should-avoid-apple-s-new-swift-language_b_5454013.html. Acesso em: 1 de Setembro de 2015.
- [18] Taylor, Paul. The challenges of porting a PC game to iPad. 18 de Setembro de 2012. Disponível em: <http://www.gamesbrief.com/2012/09/the-challenges-of-porting-a-pc-game-to-ipad/>. Acesso em: 1 de Setembro de 2015.
- [19] Gordon, Ryan. The Whys and Hows of Porting Software. 26 de Fevereiro de 2002. Disponível em: http://www.pyrogon.com/about/diary/2_26_2002.php. Acesso em: 1 de Setembro de 2015.
- [20] MacGregor, Kyle. Naughty Dog: Porting The Last of Us to PS4 was hell. 18 de Maio de 2014. Disponível em: <http://www.destructoid.com/naughty-dog-porting-the-last-of-us-to-ps4-was-hell-275048.phtml>. Acesso em: 1 de Setembro de 2015.
- [21] Popular Game Engines. Disponível em: <http://www.moddb.com/engines?sort=ranktoday-asc>. Acesso em: 1 de Setembro de 2015.
- [22] Itterheim, Steffen. Measuring Game Engine Popularity. 25 de Julho de 2013. Disponível em: <http://www.learn-cocos2d.com/2013/07/measuring-game-engine-popularity/>. Acesso em: 1 de Setembro de 2015.
- [23] DeLoura, Mark. Game Engines and Middleware (2011). 3 de Setembro de 2012. Disponível em: <http://pt.slideshare.net/markdeloura/game-engines-and-middleware-2011>. Acesso em: 1 de Setembro de 2015.
- [24] Itterheim, Steffen. The Mobile 2D Game Engine Popularity Index – November 2013. 14 de Novembro de 2013. Disponível em:

<http://www.learn-cocos2d.com/2013/11/mobile-game-engine-popularity-index/>. Acesso em: 1 de Setembro de 2015.

- [25] Chapple, Craig. The top 16 game engines for 2014. 30 de Abril de 2014. Disponível em: <https://www.develop-online.net/tools-and-tech/the-top-16-game-engines-for-2014/0192302>. Acesso em: 1 de Setembro de 2015.
- [26] Geig, Mike. Why You Should Be Using the Unity Game Engine. 4 de Abril de 2013. Disponível em: <http://www.informit.com/articles/article.aspx?p=2031153>. Acesso em: 1 de Setembro de 2015.
- [27] UnityScript versus JavaScript. Disponível em: http://wiki.unity3d.com/index.php/UnityScript_versus_JavaScript. Acesso em: 1 de Setembro de 2015.
- [28] Bard, Na'Tosha. 4.6 IS RELEASED WITH SOURCE FOR UI SYSTEM!. 26 de Novembro de 2014. Disponível em: <http://blogs.unity3d.com/2014/11/26/4-6-is-released-with-source-for-ui-system/>. Acesso em: 1 de Setembro de 2015.
- [29] Unity Documentation: MonoDevelop. Disponível em: <http://docs.unity3d.com/Manual/MonoDevelop.html>. Acesso em: 1 de Setembro de 2015.
- [30] Multiplatform: Effortlessly unleash your game on the world's hottest platforms. Disponível em: <http://unity3d.com/unity/multiplatform>. Acesso em: 1 de Setembro de 2015.
- [31] Unity Web Player. Disponível em: <https://unity3d.com/pt/webplayer>. Acesso em: 1 de Setembro de 2015.
- [32] Get Going With Unity for Consoles. Disponível em: <http://unity3d.com/unity/multiplatform/consoles>. Acesso em: 1 de Setembro de 2015.

- [33] Lambert, Steven. Making Your First Game: A Walkthrough for Game Developers. 19 de Fevereiro de 2013.
Disponível em:
<http://gamedevelopment.tutsplus.com/articles/making-your-first-game-a-walkthrough-for-game-developers--gamedev-5204>.
Acesso em: 1 de Setembro de 2015.
- [34] Unity Store. Disponível em: <https://store.unity3d.com>.
Acesso em: 1 de Setembro de 2015.
- [35] Comparações Entre Licenças.
Disponível em: <http://unity3d.com/pt/unity/licenses>.
Acesso em: 1 de Setembro de 2015.
- [36] Unity Software License Agreement 4.x. 12 de Julho de 2013.
Disponível em: <http://unity3d.com/legal/eula>.
Acesso em: 1 de Setembro de 2015.
- [37] Gallery: Made With Unity.
Disponível em: <http://unity3d.com/showcase/gallery>.
Acesso em: 1 de Setembro de 2015.
- [38] Unity: Public Relations. Disponível em: <http://unity3d.com/public-relations>.
Acesso em: 1 de Setembro de 2015.
- [39] Tuliper, Adam. Unity: Developing Your First Game with Unity and C#. Agosto de 2014. Disponível em:
<https://msdn.microsoft.com/en-us/magazine/dn759441.aspx>.
Acesso em: 1 de Setembro de 2015.
- [40] Is Cocos2d-x or Unity better for use as a cross-platform 2D game engine?.
Disponível em:
<https://www.quora.com/Is-Cocos2d-x-or-Unity-better-for-use-as-a-cross-platform-2D-game-engine>. Acesso em: 1 de Setembro de 2015.
- [41] Unity3D VS Cocos2D-X. 27 de Julho de 2014.
Disponível em: <http://blugri.com/blog/Unity3D+vs+Cocos2d-x>.
Acesso em: 1 de Setembro de 2015.

- [42] Bard, Na'Tosha. More Open-Source Initiatives from Unity. 20 de Agosto de 2014. Disponível em: <http://blogs.unity3d.com/pt/2014/08/20/more-open-source-initiatives-from-unity/>. Acesso em: 1 de Setembro de 2015.
- [43] Unity Documentation: Rendering Paths. Disponível em: <http://docs.unity3d.com/Manual/RenderingPaths.html>. Acesso em: 1 de Setembro de 2015.
- [44] Balasevicius, Vilmantas. Unity 4.2 Has Arrived. 22 de Julho de 2013. Disponível em: <http://blogs.unity3d.com/2013/07/22/unity-4-2-has-arrived/>. Acesso em: 1 de Setembro de 2015.
- [45] Unity Documentation: Using DirectX 11 in Unity 4. Disponível em: <http://docs.unity3d.com/Manual/DirectX11.html>. Acesso em: 1 de Setembro de 2015.
- [46] Unity Documentation: Command line arguments. Disponível em: <http://docs.unity3d.com/Manual/CommandLineArguments.html>. Acesso em: 1 de Setembro de 2015.
- [47] Kim, AeHyun; Bae, JaeHwan. Development of Mobile Game Using Multiplatform (Unity3D) Game Engine. International Journal of Intelligent Information Processing (IJIP), v. 5, n. 1, Março 2014. Disponível em: <http://www.aicit.org/ijiip/ppl/ijiip155ppl.pdf>. Acesso em: 1 de Setembro de 2015.
- [48] Cocos2d 0.6.0. Disponível em: <https://pypi.python.org/pypi/cocos2d>. Acesso em: 1 de Setembro de 2015.
- [49] LinkedIn: Ricardo Quesada. Disponível em: <https://www.linkedin.com/in/ricardoquesada>. Acesso em: 1 de Setembro de 2015.
- [50] Cocos2d. Disponível em: <http://cocos2d.org>. Acesso em: 1 de Setembro de 2015.

- [51] Cocos2d-x: Produtos. Disponível em: <http://www.cocos2d-x.org/products>. Acesso em: 1 de Setembro de 2015.
- [52] Developers Manual: Cocos2d-x. Disponível em: <http://www.cocos2d-x.org/wiki/Cocos2d-x>. Acesso em: 1 de Setembro de 2015.
- [53] Cocos2d-x with Facebook. Disponível em: <http://cocos2d-x.org/docs/manual/framework/html5/facebook-sdk/en>. Acesso em: 1 de Setembro de 2015.
- [54] Outdated Docs: Relationships in Cocos2d Family. Disponível em: http://www.cocos2d-x.org/wiki/Relationships_in_Cocos2d_Family. Acesso em: 1 de Setembro de 2015.
- [55] Birch, Jennifer. Unity Vs. Marmalade Vs. V-Play Vs. Corona Vs. Cocos2D: Five Cross-platform Game Engines Compared. 17 de Junho de 2014. Disponível em: <http://renatus.com/unity-marmalade-v-play-corona-cocos2d-cross-platform-game-engines>. Acesso em: 1 de Setembro de 2015.
- [56] Weiss, Nat. Selecting a Cross-platform Game Engine. 14 de Maio de 2014. Disponível em: <http://www.binpress.com/blog/2014/05/14/selecting-cross-platform-game-engine/>. Acesso em: 1 de Setembro de 2015.
- [57] Ombredane, Edouard. Comparison Between Unity and Cocos2D Platforms for 2D game Developments. 20 de Agosto de 2014. Disponível em: <http://en.yeeply.com/blog/comparison-between-unity-and-cocos2d-platforms-for-2d-game-developments/>. Acesso em: 1 de Setembro de 2015.
- [58] Developers Manual: Cocos Studio. Disponível em: http://www.cocos2d-x.org/wiki/Cocos_Studio. Acesso em: 1 de Setembro de 2015.
- [59] Cocos2d-x: Community Forums. Disponível em: <http://discuss.cocos2d-x.org>. Acesso em: 1 de Setembro de 2015.

- [60] Weiss, Nathanael. What is Cocos2D and Why is it One of the Best iPhone Game Engines?. 3 de Junho de 2011. Disponível em: <http://wizardfu.com/blog/cocos2d-best-iphone-game-engine/>. Acesso em: 1 de Setembro de 2015.
- [61] Comparison, Cocos2d-x, Unity and Sprite Kit. 7 de Abril de 2014. Disponível em: <http://www.justtotaltech.co.uk/blog/comparison-cocos2d-x-unity-sprite-kit/>. Acesso em: 1 de Setembro de 2015.
- [62] Wilcox, Mark. Top Game Development Tools: Pros and Cons. 16 de Setembro de 2014. Disponível em: <http://www.developereconomics.com/top-game-development-tools-pros-cons/>. Acesso em: 1 de Setembro de 2015.
- [63] AndrewH. Konami's Contra: Evolution runs and guns its way onto Android finally. 22 de Agosto de 2013. Disponível em: <http://www.droidgamers.com/index.php/game-news/android-game-news/6656-konami-s-contra-evolution-runs-and-guns-its-way-onto-android-finally>. Acesso em: 1 de Setembro de 2015.
- [64] Lidholt, Viktor. Go Multi-Platform with Cocos2d JS. 1 de Fevereiro de 2013. Disponível em: <https://www.zynga.com/pt/blogs/engineering/go-multi-platform-cocos2d-js-2>. Acesso em: 1 de Setembro de 2015.
- [65] Unity: Physics. Disponível em: <http://docs.unity3d.com/Manual/PhysicsSection.html>. Acesso em: 1 de Setembro de 2015.
- [66] About cocos2d-x. Disponível em: http://www.cocos2d-x.org/wiki/About_Cocos2d-x. Acesso em: 1 de Setembro de 2015.
- [67] Slackmoehrle. COCOS2D-X & JS V3.7 RELEASED. 21 de Julho de 2015. Disponível em: <http://blog.cocos2d-x.org/2015/07/cocos2d-x-js-v3-7-is-released/>.

Acesso em: 1 de Setembro de 2015.

- [68] Cocos2d-x. 3D Graphics. Disponível em:
http://www.cocos2d-x.org/wiki/3D_Graphics.
Acesso em: 1 de Setembro de 2015.
- [69] zhangxm. COCOS2D-X V3.5 FOR TIZEN RELEASED. 24 de Abril de 2015.
Disponível em:
<http://blog.cocos2d-x.org/2015/04/cocos2d-x-v3-5-for-tizen-released/>.
Acesso em: 1 de Setembro de 2015.
- [70] PHYSICS INTEGRATION. Disponível em:
<http://www.cocos2d-x.org/docs/manual/framework/native/v3/physics-integration/en>.
Acesso em: 1 de Setembro de 2015.
- [71] About Emscripten. Disponível em:
https://kripken.github.io/emscripten-site/docs/introducing_emscripten/about_emscripten.html.
Acesso em: 1 de Setembro de 2015.
- [72] Cheerp 1.1 - C++ for the Web with fast startup times, dynamic memory and now, more speed!. Disponível em:
<http://leaningtech.com/cheerp/blog/2015/06/17/Cheerp-1.1/>.
Acesso em: 1 de Setembro de 2015.
- [73] Hiiiragi, Kagami. Porting Examples and Demos. Disponível em:
<https://github.com/kripken/emscripten/wiki/Porting-Examples-and-Demos>.
Acesso em: 1 de Setembro de 2015.
- [74] Quem fala inglês no mundo?. 27 de Junho de 2013. Disponível em:
<http://www.englishtown.com.br/blog/quem-fala-ingles-no-mundo/>.
Acesso em: 1 de Setembro de 2015.
- [75] Amorim, Maíra. Brasileiros não sabem falar inglês: apenas 5% dominam o idioma. 30 de Setembro de 2012. Disponível em:
<http://oglobo.globo.com/economia/emprego/brasileiros-nao-sabem-falar-ingles-apenas-5-dominam-idioma-6239142>.

Acesso em: 1 de Setembro de 2015.

- [76] Pedro, Carla. 51,3% dos portugueses adultos não fala uma única língua estrangeira. 24 de Setembro de 2009. Disponível em:
http://www.jornaldenegocios.pt/economia/detalhe/513_dos_portugueses_adultos_natildeo_fala_uma_uacutenica_liacutengua_estrangeira.html.
Acesso em: 1 de Setembro de 2015.
- [77] Cocos2d-x Showcase - Matching with Friends. Disponível em:
<http://www.cocos2d-x.org/games/1215>.
Acesso em: 1 de Setembro de 2015.

8. Anexos

8.1. Anexo A - Documento de Propostas de Melhoria - Videojogo “Falcao vs Aliens”

Propostas de Alterações/Melhorias

1. Introdução

Neste documento serão apresentadas propostas de alterações e melhorias à versão original do videojogo “Falcao vs Aliens”, assim como detalhes para a implementação das mesmas. Os objetivos principais a atingir com as alterações/melhorias são:

- Oferecer aos jogadores uma variedade de modos de jogo, de forma a satisfazer tanto a quem procura desafios como a quem pretende apenas experienciar uma sessão de jogo descontraída e divertida;
- Aumentar a taxa de retenção de jogadores;
- Explorar a componente social e tentar aproveitá-la de forma a cativar mais jogadores.

O documento encontra-se dividido em dois capítulos, o primeiro apresenta as propostas/melhorias aprovadas pela WingzStudio e a sua prioridade (desejável, importante e essencial), enquanto que o segundo apresenta as que foram rejeitadas.

2. Propostas Aprovadas

Neste capítulo serão apresentadas as propostas que foram aprovadas pela WingzStudio e a prioridade de implementação das mesmas.

2.1. Suporte a Novos *Aspect Ratios* e Adaptação da Interface

Prioridade: Desejável.

Nesta secção são apresentadas propostas relacionadas com o suporte a novos *aspect ratios*. A versão original do “Falcao vs Aliens” oferece suporte aos *aspect ratios* usados pela maioria dos dispositivos móveis, mas apenas na orientação vertical (Figura 1), correspondendo aos seguintes:

- 10:16 - utilizado, por exemplo, por alguns modelos do *iPhone*;
- 9:16 - utilizado, por exemplo, por alguns modelos do *iPhone* e *Moto G*;
- 3:4 - utilizado, por exemplo, por alguns modelos do *iPad*;

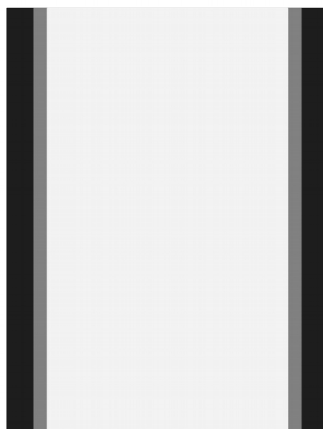


Figura 1 - Sobreposição dos *Aspect Ratios* verticais suportados pelo “Falcao vs Aliens”
Legenda: Preto- 12:16 (3:4); Cinza- 10:16; Branco- 9:16.

Propõe-se a adição do suporte também aos seguintes *aspect ratios* de orientação horizontal (Figura 2):

- 16:10 - usado, por exemplo, por alguns monitores.
- 16:9 - usado, por exemplo, por alguns monitores
- 4:3 - usado, por exemplo, por algumas televisões.

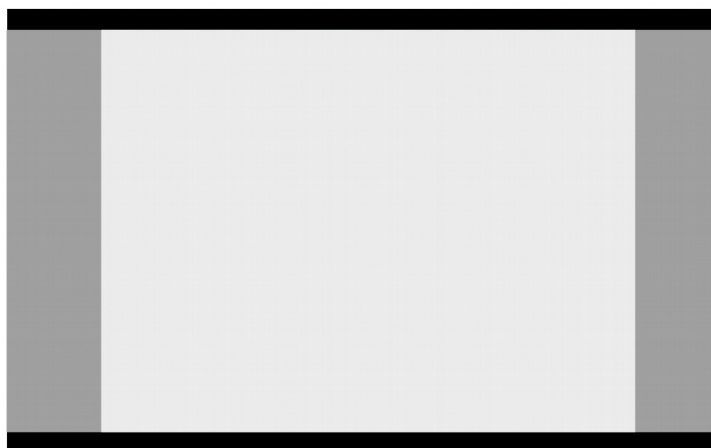


Figura 2 - Sobreposição dos *Aspect Ratios* horizontais propostos a serem suportados pelo "Falcao vs Aliens"

Legenda: Preto- 16:10; Cinza- 16:9; Branco- 12:9 (4:3).

Com a adição do suporte a novos *aspect ratios*, propõe-se que a interface seja reformulada, permitindo que esta se ajuste consoante o *aspect ratio* em uso. Quanto aos que são suportados pela versão original, provavelmente não serão necessárias alterações à interface, quanto aos novos *aspect ratios* a interface terá que ser reformulada quase na totalidade.

Como é visível na Figura 2, o espaço do *aspect ratio* 4:3 é comum aos restantes, pelo que propõe-se adaptar a interface atual de forma a ocupar todo esse espaço. Quanto ao espaço restante propõe-se a extensão do cenário de fundo, de forma a evitar faixas pretas.

2.2. Modos de Jogo

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração de modos de jogo.

2.2.1. Alteração do Modo História

Prioridade: Essencial.

No modo história da versão original é necessário completar todos os níveis numa única sessão, a qual se inicia com quatro vidas. A sugestão seria começar apenas com uma vida, mas quando se completasse um nível desbloquearia o próximo, o qual ficaria disponível sempre. Desta forma, evita-se a frustração dos jogadores que não possuem tempo suficiente para completar o videojogo numa única sessão e são obrigados a começar sempre do primeiro nível. Além dessa alteração, os níveis passariam a ter pontuação individual e, consoante a pontuação obtida seria atribuído um número de estrelas/taças conquistadas (0 a 3) nesse nível. O jogador irá passar a escolher os níveis por estádio (Figura 3), estando

inicialmente disponível apenas o primeiro estágio. Cada estágio terá três níveis (Figura 4) e após um determinado número de estrelas/taças o próximo estágio é desbloqueado.



Figura 3 - Protótipo Mundo de jogo do modo história do “Falcao vs Aliens”

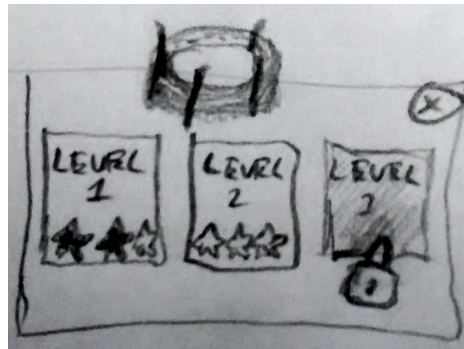


Figura 4 - Protótipo da seleção de níveis por estágio no modo história do “Falcao vs Aliens”

2.2.2. Modo Desafiar Amigos

Prioridade: Importante.

Propõe-se a adição de um modo de jogo que permita desafiar amigos ou outras pessoas para um duelo. Este modo consistirá numa sessão de jogo, a qual poderá ter várias opções diferentes, mas no geral deverá ser semelhante a um nível do modo história da versão original. Essa sessão será jogada pelos dois jogadores, cada um terá apenas visibilidade sobre o seu jogo e não será realizada em tempo real, a justificação para não ser em tempo real deve-se ao facto de não obrigar as duas pessoas a estarem *online* em simultâneo, o que poderia causar um longo tempo de espera por adversário. Desta forma, o sistema funcionará da seguinte forma:

- Jogador escolhe este modo de jogo;
- Desafia um amigo através do *Facebook* ou uma pessoa de forma aleatória (Figura 6);
- É enviado o convite ao adversário, o qual terá três dias para aceitar e jogar, visível no menu deste modo de jogo (Figura 5);
- Inicia-se a sessão de jogo, independentemente do adversário aceitar/recusar;
- Quando o jogador termina a pontuação que obteve é guardada;
- Após ambos os jogadores terminarem as pontuações obtidas são comparadas, o jogador com maior pontuação vence o duelo;
- Após três dias se o adversário não tiver jogado é dado como vencido.

Com este modo pretende-se adicionar uma maior componente social ao “Falcao vs Aliens”, o que poderá resultar num aumento de jogadores e da sua retenção.

Após discussão com a empresa, decidiu-se que os jogadores só poderão jogar em níveis que já desbloquearam no modo história, pois caso fosse possível jogar em qualquer nível, o modo história deixaria de ter o impacto desejado, visto que se estaria a revelar ao jogador os diferentes níveis existentes.

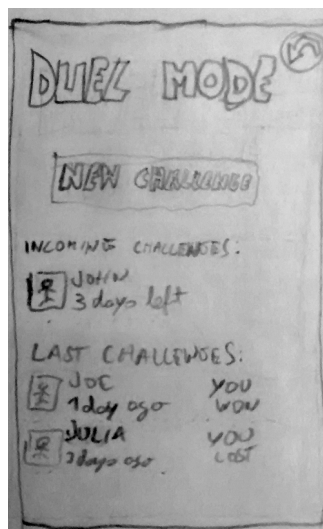


Figura 5 - Protótipo do menu do modo desafiar amigos do “Falcao vs Aliens”

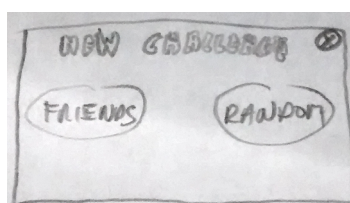


Figura 6 - Protótipo da *pop-up* de novo desafio do modo desafiar amigos do “Falcao vs Aliens”

2.3. Controladores

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração dos controladores suportados pelo videojogo. A versão original apenas suporta *touch*, pelo que com a adição do suporte a novas plataformas será vantajoso oferecer suporte a diferentes controladores.

2.3.1. Suporte a Rato e Teclado

Prioridade: Essencial.

Sugere-se adição do suporte a rato e teclado, as teclas para controlar o Falcao seriam: setas (esquerda e direita); ou as teclas “A” e “D” (esquerda e direita respetivamente). Para chutar a bola seria usado o “espaço”.

No caso do rato, sugerem-se duas opções:

- O rato seria usado de forma semelhante ao *touch*, ou seja, para mover o Falcao teria que se manter premido o botão esquerdo e mover o rato para a esquerda e direita. Para chutar a bola seria um clique no botão esquerdo.
- A outra opção seria o Falcao seguir o movimento do rato, sem ser necessário manter premido o botão do lado esquerdo. Para chutar a bola continuaria a ser um clique no botão esquerdo.

Em discussão com a empresa, decidiu-se que a segunda opção é a mais indicada.

2.4. Novos Idiomas

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração de idiomas.

2.4.1. Lista de Novos Idiomas

Prioridade: Importante.

Propõe-se a adição dos seguintes idiomas:

- Espanhol - como o videojogo tem como personagem principal o jogador de futebol colombiano Radamel Falcao, considera-se importante ter o idioma oficial da Colômbia. Além do referido, as estatísticas de uso do videojogo mostraram que a maioria dos jogadores atuais residem em países que têm Espanhol como idioma principal;

- Português - o videojogo foi desenvolvido em Portugal por uma empresa portuguesa, então é coerente incluir este idioma;

Como nem todos as pessoas dominam o idioma Inglês, com a adição destes idiomas é esperado que a facilidade de uso do videojogo aumente, assim como o seu número de jogadores.

2.5. Interface

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração da interface do “Falcao vs Aliens”.

2.5.1. Novo Menu de Seleção do Modo de Jogo

Prioridade: Essencial.

Sugere-se a adição de um novo menu de selecção dos diferentes modos de jogo, ou seja, quando o jogador clica no botão “play” do menu principal, em vez de ir direto para o modo história (como ocorre atualmente), será apresentado um menu onde poderá escolher qual o modo em que pretende jogar (Figura 7).

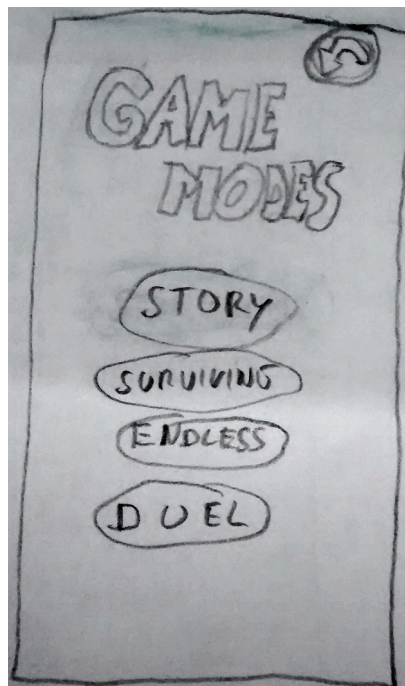


Figura 7 - Protótipo do menu de seleção do modo de jogo do “Falcao vs Aliens”

2.6. Funcionalidades

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração de funcionalidades.

2.6.1. Sincronização dos Dados nos Vários Dispositivos

Prioridade: Essencial.

Esta funcionalidade permitirá aos jogadores poderem ter as suas definições e restante dados sincronizados nos diferentes dispositivos que utilize. Com esta funcionalidade é esperado reduzir a frustração dos jogadores terem que começar tudo do zero sempre que mudam de dispositivo.

2.6.2. Guardar o Estado Atual da Sessão de Jogo

Prioridade: Essencial.

Na versão original do “Falcao vs Aliens”, ao sair da sessão de jogo todo o progresso é perdido, pelo que sugere-se que seja guardado este progresso de forma a permitir ao jogador continuar noutro momento. Caso sejam aceites as alterações sugeridas ao modo história, este passará obrigatoriamente a ter que guardar o estado atual da sessão de jogo, mas também é pretendido que no modo sobrevivência seja possível guardar o estado (vidas, nível e pontuação atual). O motivo desta sugestão deve-se ao facto deste tipo de videojogos serem usados para curtos períodos de utilização, o que poderá nunca ser suficiente para completar todos os níveis, deste modo, começar sempre do primeiro nível poderá causar frustração e desinteresse.

2.6.3. Integração de *GameAnalytics*

Prioridade: Desejável.

De forma a poder retirar o máximo de informação sobre a utilização do videojogo, sugere-se a integração de uma ou mais ferramentas de *gameanalytics*.

2.6.4. Contornar *Unity Plugin*

Prioridade: Desejável.

Atualmente para executar videojogos desenvolvidos em *Unity*, no *Facebook* e *browsers* é necessária a instalação do *Unity Plugin*, o que poderá não ser conveniente para alguns utilizadores. Deste modo, propõe-se um estudo com o intuito de encontrar formas de contornar o uso do *plugin*.

2.6.5. Contornar *Loop Infinito da Bola*

Prioridade: Desejável.

Como um dos principais problemas da jogabilidade da versão original do “Falcao vs Aliens” é a ocorrência de *loops* infinitos ou semi-infinitos em que a bola

fica a colidir com as paredes (direita e esquerda), a sugestão seria detectar esses casos. Para isso, se a bola demorasse mais do que um determinado tempo a avançar no terreno de jogo, ocorreria um evento, como por exemplo, a entrada de um adepto no campo que rematasse a bola ou uma nave alienígena que lançasse um raio em direção à bola.

2.7. Power Ups

Prioridade: Importante.

Nesta secção serão apresentadas propostas relacionadas com a adição ou alteração de *power ups* (Tabela 1).

Power Up	Efeito
	Bola de Chumbo: a bola passa a dar mais dano ao alienígenas.
	Íman: os <i>power ups</i> são atraídos pelo Falcao.
	Diminuição da Velocidade da Bola
	Congelar: os alienígenas ficam sem se poder mover ou disparar durante 10 segundos.
	Tartaruga/Lesão: o Falcao move-se mais lentamente.

Tabela 1 - Lista de Novos *Power Ups* Propostos a adicionar no “Falcao vs Aliens”

2.8. Achievements

Prioridade: Importante.

Nesta secção serão apresentadas propostas relacionadas com a adição (Tabela 2) ou alteração de *achievements* (Tabela 3).

2.8.1. Novos Achievements


Ação	Prémio	Imagem
	Camisola do Manchester United	
Completar o modo história		
Ganhar 5 desafios seguidos contra outros jogadores.		
Convidar 20 amigos.		

Tabela 2 - Lista de Novos *Achievements* Propostos a adicionar no “Falcao vs Aliens”

2.8.2. Alterações aos *Achievements* Existentes

Na versão original do “Falcao vs Aliens” ao recuperar o terceiro estádio o jogador é recompensado com uma suposta camisola do Futebol Clube do Porto, no entanto a cor usado é um azul clarinho semelhante à cor usada na camisola da seleção Argentina, pelo que sugere-se a alteração para uma camisola mais semelhante à do Porto, como pode ser visível na Tabela 3. Deste modo, ou a camisola original seria removida ou, caso se pretenda manter, mudar a descrição para camisola da Argentina e alterar a ação necessária para a desbloquear.



Ação	Prémio	Imagem
Claim back Stadium 3	Camisola do Porto	
	Camisola da Argentina	

Tabela 3 - Lista de Alterações aos *achievements* propostas para o “Falcao vs Aliens”

3. Propostas Rejeitadas

Neste capítulo serão apresentadas as propostas que foram rejeitadas pela WingzStudio.

3.1. Modos de Jogo

3.1.1. Adição de Nova Dificuldade de Jogo

Sugere-se a adição de uma nova dificuldade ao modos história. O modo normal seria como está o modo história atualmente, ou seja, quatro vidas iniciais e pode-se apanhar vidas extras. O modo difícil seria desbloqueado aquando do término da dificuldade normal e as diferenças desta dificuldade seriam:

- Uma única vida e não há vidas extras para apanhar;
- Alienígenas mais resistentes;
- Aumento da velocidade da bola.

3.1.2. Algoritmo de Geração Aleatória de Níveis

De forma a aumentar a diversidade das sessões de jogo, propõe-se a criação de um algoritmo que crie níveis de forma aleatória.

3.1.3. Editor de Níveis

Outra sugestão é a introdução de um editor de níveis, que permitisse aos jogadores criar, partilhar e jogar níveis criados por eles mesmos. O jogador poderia escolher os tipos de alienígenas desejados e a sua posição, também teria à sua disposição outras opções como: velocidade da bola, resistência dos alienígenas, power ups, entre outras.

3.1.4. Modo Praticar

Propõe-se a adição de um modo de jogo onde se podem escolher níveis que já foram completados no modo história. O jogador terá apenas uma vida e o objetivo é atingir o máximo de pontuação, deste modo existirá uma tabela das maiores pontuações em cada nível.

3.1.5. Modo Infinito

Este modo consistirá num nível que não tem fim, o jogador começaria com uma vida e o objetivo seria obter o máximo de pontuação possível, para isso teria que destruir o máximo de alienígenas possível. Neste modo são sugeridas duas abordagens:

Quando se destrói todos os inimigos aparece uma mensagem a dizer que foi completado o turno atual e que se irá iniciar um novo, aparecendo de seguida uma nova vaga de inimigos;

A outra abordagem seria baseado no tempo, após um determinado tempo os inimigos avançariam no terreno e seria adicionada uma nova linha de inimigos nas costas dos atuais.

8.2. Anexo B - Documento de Requisitos - *Porting* do Videojogo “Falcao vs Aliens”

WingzStudio, Lda

Daniel Francisco Lopes de Carvalho
dflopes@student.dei.uc.pt

25 de Fevereiro de 2015

Documento de Requisitos

Porting do Videojogo “Falcao vs Aliens”

Versão do Documento: 1.2

Índice

1. Introdução	2
2. Descrição Geral do Sistema	4
3. Requisitos Funcionais (Casos de Uso)	7
4. Requisitos Não-funcionais	22

1. Introdução

Este documento apresenta a especificação de requisitos para o projeto de *porting* do videojogo “Falcao vs Aliens”, desenvolvido pela WingzStudio. O documento tem como alvo a equipa de desenvolvimento, fornecendo informações necessárias para o projeto e para a sua implementação, assim como para a realização dos testes ao mesmo.

1.1 Visão Geral do Documento

Além desta secção introdutória, as secções seguintes estão organizadas da seguinte forma:

Secção 2 – Descrição Geral do Sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu propósito e descrevendo os seus utilizadores.

Secção 3 – Requisitos Funcionais (Casos de Uso): especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.

Secção 4 – Requisitos Não-funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.

Secção 5 – Referências: apresenta referências para outros documentos utilizados para a confeção deste documento.

1.2 Convenções, Termos e Abreviações

1.2.1 Identificação dos Requisitos

Por convenção, a referência a requisitos é feita através do nome da subsecção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subsecção.identificador do requisito]

Por exemplo, o requisito funcional [Registo/Login.RF001] deve estar descrito em uma subsecção chamada “Registo/Login”, em um bloco identificado pelo número [RF001]. Já o requisito não-funcional [Interface.RNF001] deve estar descrito na secção de requisitos não-funcionais de Interface, em um bloco identificado por [RNF001]. Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [RNF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos, nas Secções 3 e 4, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados obrigatoriamente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser publicado e usado.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo disponível para implementá-los na versão que está a ser especificada.

1.2.3 Classificação dos Requisitos

De forma a distinguir os requisitos que foram introduzidos ou alterados para esta nova versão do sistema, nas Secções 3 e 4, foram adotadas as denominações “novo”, “alterado” e “mantido”.

- **Novo** é um requisito que foi introduzido para esta nova versão do sistema, não existindo na original.
- **Alterado** é um requisito existente na versão original, mas sofreu alterações nesta nova versão.
- **Mantido** é um requisito existente na versão original e que será incluído na nova versão.

2. Descrição Geral do Sistema

O “Falcao vs Aliens” é o primeiro videojogo exclusivo que conta com a presença da estrela de futebol Radamel Falcao. O conceito é adequado a todas as idades, sendo inspirado no estilo do clássico *Arkanoid*, que consistia na destruição de blocos com uma bola ao mesmo tempo que o jogador controlava uma barra para evitar que a bola saísse do ecrã.

Os principais objetivos deste videojogo são proporcionar diversão aos utilizadores e publicitar a estrela de futebol associada. No caso concreto deste projecto os objetivos principais são:

- Adicionar ao videojogo compatibilidade com novas plataformas, principalmente: *Windows*, *Facebook* e navegadores atuais.
- Integração com o *Facebook* nas principais plataformas (*Android*, *iOS* e *Web*).
- Adição de novas funcionalidades: modos de jogo (história e desafio), sincronização entre dispositivos e, os idiomas português e espanhol.
- Melhoria/Balanceamento da jogabilidade, *power ups* e níveis.

Ao atingir esses objetivos a WingzStudio espera aumentar o número de jogadores.

2.1 Abrangência e Sistemas Relacionados

O “Falcao vs Aliens” é um *software* de entretenimento. Este fornece aos fãs da estrela de futebol Radamel Falcao a possibilidade de o controlarem e através das suas capacidades salvar o planeta de invasores alienígenas.

O videojogo é inspirado no estilo do *Arkanoid*, pelo que a jogabilidade é semelhante ao mesmo.

2.1.1 Descrição do Videojogo

O videojogo é composto por 18 níveis, uma vez que o tema do jogo é futebol cada nível corresponderá a um estádio de futebol, havendo um total de 6 estádios diferentes. A ação decorrerá no relvado, estando a personagem principal na parte inferior do ecrã, que se encontra aproximadamente a meio do campo. Na zona oposta à personagem principal encontra-se a baliza adversária. A versão original conta apenas com um único modo de jogo, no qual para acabar o videojogo era necessário completar todos os níveis sem perder as vidas disponíveis. Existe um único jogador humano, este controla a personagem principal: Radamel Falcao. O jogador nesse modo de jogo começa com quatro vidas, correspondendo a quatro bolas, perde-se uma vida quando o Falcao deixa a bola passar. Na nova versão,

serão adicionados dois modos de jogo, modo história e modo desafio. O modo história que será semelhante ao modo presente na versão original, mas o progresso é guardado, sendo possível desbloquear níveis e continuar a partir deles quando o jogador perde ou termina a sessão. O modo desafio, permitirá desafiar um jogador humano para ambos jogarem o mesmo nível, o vencedor é o jogador que obtiver melhor pontuação.

Existem inimigos alienígenas controlados por inteligência artificial, estes têm características diferentes entre eles, por exemplo alguns são fixos/móveis ou estão armados/desarmados. Ao acertar com a bola nos inimigos faz com que estes percam vida, se ficar a zero morrem e desaparecem do ecrã. Os alienígenas encontram-se na zona frontal à baliza, protegendo-a.

Os jogadores terão em cada nível que destruir os diversos alienígenas espalhados pelo cenário, abrindo desta maneira o caminho para a baliza. A baliza encontra-se inicialmente bloqueada com uma barreira, essa barreira só desaparece após serem apanhadas três estrelas, estas encontram-se escondidas atrás de alguns alienígenas, pelo que é necessário destruí-los para que estas sejam reveladas e possam ser apanhadas.

Após a baliza estar desbloqueada, um alienígena de maior estatura aparece a defendê-la, sendo então necessário destruí-lo para que a baliza fique aberta e seja possível marcar golo. O nível é completado assim que o jogador consegue marcar golo, permitindo ao jogador prosseguir para o nível seguinte. Além do que já foi referido, é também de destacar que: existem *power ups* que são possíveis de apanhar, estes vão aparecendo aleatoriamente consoante eliminamos os alienígenas, podendo ter efeitos positivos ou negativos; existem garrafas de energia que se podem apanhar, protegendo a personagem principal dos ataques dos inimigos; existem *achievements* colecionáveis. O videojogo contará com a integração com o *Facebook*, o que permitirá aos jogadores partilhar os seus *achievements* e pontuações obtidas.

2.1.2 Características do Videojogo

- Videojogo 2D.
- Géneros: arcade, desporto, casual.
- Suporte a várias plataformas, entre as quais: *Android*, *iOS*, *Windows Phone*, *Windows*, *Facebook*, navegadores atuais.
- Integração com o *Facebook* (necessário efetuar *login* com as credenciais de uma conta no *Facebook*).
- 18 níveis e 6 estádios (cenários) diferentes.
- Radamel Falcao como personagem principal.
- Personagem principal controlável através de *touch* nas plataformas móveis e plataformas com suporte a *touch*, nas restantes plataformas através do rato.
- Variedade de inimigos alienígenas.

- *Power ups* positivos e negativos.
- *Achievements* colecionáveis.
- Registo da pontuação mais alta e tabela de pontuações dos jogadores.
- Interface gráfica.
- Menus.
- Opções de jogo.
- Sistema de partículas.
- Animações.
- Efeitos sonoros e música.
- Física.
- Colisões.

2.1.3 Restrições do Videojogo

O videojogo deverá ser capaz de executar pelo menos nas seguintes plataformas: *Android*, *iOS*, *Windows Phone*, *Windows*, *Facebook*, navegadores atuais.

Em sistemas operativos *Android*, o *apk* não poderá exceder os 50Mb de espaço físico, no entanto, caso o espaço necessário seja maior poderão ser usados mecanismos alternativos, como por exemplo, efetuar o *download* de ficheiros adicionais na primeira execução do videojogo. Nas restantes plataformas não há restrição de espaço.

Em relação à memória *RAM*, esta não poderá exceder os 200MB em dispositivos móveis, nos restantes não há restrição.

A resolução base será 640x960, mas é esperado que o videojogo suporte todas as resoluções verticais/horizontais suportadas pelas plataformas.

Deverá ser usada uma base de dados *online* para guardar dados relativos às pontuações de cada jogador nos diversos modos de jogo, assim como o progresso de níveis no modo história e *achievements* conquistados. No entanto, é necessário que o jogador se tenha autenticado com uma conta *Facebook*. Além desta base de dados é necessário um mecanismo de gravação para guardar os dados relativos aos *achievements*, se o jogador já alguma vez viu o tutorial, pontuações do jogador e progresso no modo história.

2.2 Descrição de Utilizadores

O videojogo tem como objetivo abranger um público de todas as idades, estando incluído na categoria de videojogos casuais, ou seja, que são utilizados em momentos de pausa, como são exemplo as viagens de transportes públicos, as esperas para as consultas ou os intervalos no trabalho.

Apesar de o jogo ser destinado a todo o tipo de pessoas, existirão dois grupos que serão representativos da maioria dos jogadores: crianças e adultos com idade entre os 20 e os 40 anos. De acordo com estas características foram elaborados quatro perfis que têm como objetivo perceber quais os objetivos destes utilizadores durante a utilização do videojogo:

- **Samuel - 5 anos:** Frequenta uma escola privada e todos os dias o pai ou a mãe vão buscá-lo e levá-lo. Só gosta das viagens de carro quando os pais o deixam jogar no *iPad*. Gosta de jogos com personagens divertidas e que sejam visualmente apelativas.
- **Sofia (Estudante Universitária) - 20 anos:** Vive num apartamento a três quilómetros da universidade em que estuda. Gosta de utilizar o seu telemóvel para jogar enquanto viaja de autocarro para a universidade. Gosta de jogos com uma jogabilidade fácil e com um visual divertido.
- **José (Contabilista) - 27 anos:** Vive nos subúrbios, mas trabalha numa empresa no centro da cidade. Todos os dias apanha o metro para ir até ao escritório passando as viagens a jogar no *iPhone*. Gosta particularmente de videojogos desafiantes e onde possa superar-se a ele próprio.
- **Rafael (Empresário) - 39 anos:** Vive numa vivenda, a qual partilha com a esposa. Todos os dias à noite usa o *Facebook*, no qual ocasionalmente gosta de jogar para descontrair. Prefere jogos que não exijam muito tempo e com uma baixa curva de aprendizagem.

Com a análise destes perfis foi possível chegar à conclusão que o videojogo deve ter um visual apelativo, uma jogabilidade simples, uma rápida aprendizagem de como jogar e ao mesmo tempo ser desafiante de forma a satisfazer os jogadores. É também possível perceber que enquanto um grupo de utilizadores apenas está interessado na diversão, outro grupo está interessado no desafio apresentado.

3. Requisitos Funcionais (Casos de Uso)

3.1 Registo/Login

[RF001] Efetuar *login*

Descrição do caso de uso: Este caso de uso permite que o utilizador efetue o *login* utilizando os dados de uma conta do *Facebook*.

Prioridade: Importante.

Classificação: Alterado.

Entradas e Pré-condições:

Caso 1 - O utilizador deve estar no menu inicial, ter clicado para continuar. Seguidamente é apresentada uma notificação a perguntar se deseja efetuar *login* no *Facebook*. O utilizador deverá aceitar efetuar o *login* e introduzir os dados da conta *Facebook*.

Caso 2 - O utilizador deve estar no menu de seleção de modo de jogo e ter clicado no modo desafio. Seguidamente é apresentada uma notificação a perguntar se deseja efetuar *login* no *Facebook*. O utilizador deverá aceitar efetuar o *login* e introduzir os dados da conta *Facebook*.

Caso 3 - O utilizador deve estar no menu de *achievements*, deve clicar na pontuação e seguidamente na opção de efetuar *login* no *Facebook*. Introduzir os dados da conta *Facebook*.

Caso 4 - O utilizador completou o videojogo ou obteve *gameover*. É lhe apresentada a pontuação que obteve, deve clicar na pontuação e seguidamente na opção de efetuar *login* no *Facebook*. Introduzir os dados da conta *Facebook*.

Saídas e Pós-condições: O sistema reencaminha para uma página de autenticação no *Facebook*.

No caso 1, se os dados preenchidos forem validados pelo *Facebook*, deverá ser apresentada uma notificação a avisar que a operação foi bem sucedida.

No caso 2, se os dados preenchidos forem validados pelo *Facebook*, deverá ser apresentada uma notificação a avisar que a operação foi bem sucedida e encaminhar o utilizador para o menu do modo desafio.

Nos casos 3 e 4, se os dados preenchidos forem validados pelo *Facebook* deverá ser apresentada a maior pontuação obtida pelo utilizador e a tabela com as pontuações mais altas.

[RF002] Efetuar registo

Descrição do caso de uso: Este caso de uso permite que o utilizador efetue o registo de uma conta no *Facebook*.

Prioridade: Importante.

Classificação: Alterado.

Entradas e Pré-condições:

Caso 1 - O utilizador deve estar no menu inicial, ter clicado para continuar. Seguidamente é apresentada uma notificação a perguntar se deseja efetuar *login* no *Facebook*. O utilizador deverá aceitar efetuar o *login* e seguidamente escolher a opção de criar nova conta. Introduzir os dados da conta *Facebook*.

Caso 2 - O utilizador deve estar no menu de seleção de modo de jogo e ter clicado no modo desafio. Seguidamente é apresentada uma notificação a perguntar se deseja efetuar *login* no *Facebook*. O utilizador deverá aceitar efetuar o *login* e

seguidamente escolher a opção de criar nova conta. Introduzir os dados da conta *Facebook*.

Caso 3 - O utilizador deve estar no menu de *achievements*, deve clicar na pontuação e seguidamente na opção de efetuar *login* no *Facebook*. Escolher a opção de criar nova conta. Introduzir os dados da conta *Facebook*.

Caso 4 - O utilizador completou o videojogo ou obteve *gameover*. É lhe apresentada a pontuação que obteve, deve clicar na pontuação e seguidamente na opção de efetuar *login* no *Facebook*. Escolher a opção de criar nova conta. Introduzir os dados da conta *Facebook*.

Saídas e Pós-condições: O sistema apresenta os campos necessários para preencher com os dados da conta *Facebook*.

No caso 1, se os dados preenchidos forem validados pelo *Facebook*, deverá ser apresentada uma notificação a avisar que a operação foi bem sucedida.

No caso 2, se os dados preenchidos forem validados pelo *Facebook*, deverá ser apresentada uma notificação a avisar que a operação foi bem sucedida e encaminhar o utilizador para o menu do modo desafio.

Nos casos 3 e 4, se os dados preenchidos forem validados pelo *Facebook* deverá ser apresentada a maior pontuação obtida pelo utilizador e a tabela com as pontuações mais altas.

3.2 Interface

[RF001] Menu Principal

Descrição do caso de uso: Este caso de uso permite que o utilizador acesse o menu principal.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deverá iniciar o videojogo, clicar para continuar.

Saídas e Pós-condições: O menu principal é apresentado com três botões/opções disponíveis (*play*, opções e *cacifo*).

[RF002] Aceder ao Menu de Seleção do Modo de Jogo

Descrição do caso de uso: Este caso de uso permite que o utilizador acesse o menu de seleção do modo de jogo.

Prioridade: Essencial.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve estar no menu principal e deve clicar no botão *play*.

Saídas e Pós-condições: O menu de seleção de modo de jogo é apresentado com três botões/opções correspondentes a cada um dos modos (história, arcade e desafio).

[RF003] Entrar no Modo História

Descrição do caso de uso: Este caso de uso permite que o utilizador entre no modo de jogo história.

Prioridade: Essencial.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve estar no menu de seleção de modos de jogo e deve clicar no botão correspondente ao modo história.

Saídas e Pós-condições: Caso o utilizador ainda não tiver iniciado nenhuma vez este modo, deverá ser apresentada uma breve introdução animada. Após esta animação o utilizador será encaminhado para o mundo de jogo, o qual contém os estádios e níveis.

[RF004] Entrar num nível do Modo História

Descrição do caso de uso: Este caso de uso permite que o utilizador entre num nível do modo de jogo história.

Prioridade: Essencial.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve estar no mundo de jogo, selecionar um estádio que esteja desbloqueado e clicar num nível desse estádio que também esteja desbloqueado.

Saídas e Pós-condições: O sistema inicializa uma nova sessão de jogo com as configurações atuais (música, efeitos sonoros e nível escolhido).

[RF005] Entrar no Modo Arcade

Descrição do caso de uso: Este caso de uso permite que o utilizador entre no modo de jogo arcade.

Prioridade: Essencial.

Classificação: Alterado.

Entradas e Pré-condições: O utilizador deve estar no menu de seleção de modos de jogo e deve clicar no botão correspondente ao modo arcade.

Saídas e Pós-condições: O sistema deverá apresentar uma breve introdução animada, seguidamente deverá inicializar uma nova sessão de jogo com as configurações atuais (música, efeitos sonoros).

[RF006] Entrar no Modo Desafio

Descrição do caso de uso: Este caso de uso permite que o utilizador entre no modo de jogo desafio.

Prioridade: Desejável.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve estar no menu de seleção de modos de jogo, deve ter efetuado *login* no *Facebook* e deve clicar no botão correspondente ao modo desafio.

Saídas e Pós-condições: O sistema deverá apresentar o menu do modo desafio com os vários desafios pendentes, estatísticas de resultados e botão/opção de desafiar jogadores.

[RF007] Desafiar um Jogador

Descrição do caso de uso: Este caso de uso permite que o utilizador desafie outra pessoa para jogar no modo desafio.

Prioridade: Desejável.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve ter efetuado *login* no *Facebook*, estar no menu do modo desafio, clicar no botão desafiar. Seguidamente terá a opção de escolher um amigo ou uma pessoa aleatória. Terá ainda que escolher o nível em que quer jogar ou deixar que este seja escolhido de forma aleatória.

Saídas e Pós-condições: O sistema inicializa uma nova sessão de jogo com as configurações atuais (música, efeitos sonoros e nível) e serão enviadas informações relacionadas ao desafio para o servidor, sendo guardadas na base de dados *online*.

[RF008] Aceitar Desafios

Descrição do caso de uso: Este caso de uso permite que o utilizador aceite desafios.

Prioridade: Desejável.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve ter efetuado *login* no *Facebook*, estar no menu do modo desafio. Neste menu terá uma lista de desafios pendentes, tendo o utilizador que clicar no qual pretende aceitar.

Saídas e Pós-condições: O sistema inicializa uma nova sessão de jogo com as configurações atuais (música, efeitos sonoros e nível) e serão enviadas informações relacionadas ao desafio para o servidor, sendo guardadas na base de dados online.

[RF009] Consultar Estatísticas de Desafios

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize as estatísticas relativas ao modo desafio, mais especificamente o número de vitórias e derrotas.

Prioridade: Desejável.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve ter efetuado *login* no *Facebook* e estar no menu do modo desafio.

Saídas e Pós-condições: As estatísticas são visíveis no menu do modo desafio.

[RF010] Apresentar tutorial

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize o tutorial que ensina como jogar.

Prioridade: Importante.

Classificação: Mantido.

Entradas e Pré-condições:

Caso 1 - O utilizador deve estar no menu de seleção de modos de jogo e clicar no modo arcade.

Caso 2 - O utilizador deve estar no mundo de jogo, ser a primeira vez que vai jogar e escolher o nível para jogar.

Saídas e Pós-condições: O sistema após ter iniciado a nova sessão de jogo deverá mostrar o tutorial antes de permitir que o jogador comece a jogar. No caso 2, se for a primeira vez que o jogador vê o tutorial, o sistema deverá impossibilitar que o jogador faça *skip*, caso contrário o sistema deverá apresentar a opção de fazer *skip*.

[RF011] Apresentar pontuação atual

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize a pontuação atual enquanto se encontra numa sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo.

Saídas e Pós-condições: O sistema apresenta um contador de pontos na parte superior do ecrã, o qual começa em zero. Existem diversas formas de obter ou perder pontos: acertar nos alienígenas dá um ponto, matar alienígenas, apanhar *power ups*, apanhar garrafa, apanhar estrelas, postes da baliza, marcar golo, perder vidas e ser derrubado.

[RF012] Apresentar pontuação recorde

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize a pontuação recorde enquanto se encontra numa sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo.

Saídas e Pós-condições: O sistema apresenta o recorde de pontos na parte superior do ecrã.

[RF013] Aceder ao *in-game menu*

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda a um menu enquanto está a decorrer uma sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e, deve clicar num dos seguintes:

- Botão de pausa;
- Tecla de sistema *back* (*Android*, *iOS*).
- Tecla “esc”;
- Tecla “p”;
- Botão central do rato.

Saídas e Pós-condições: O sistema apresenta um menu com as opções: ativar/desativar todo o som, continuar jogo, sair para o menu principal e reiniciar jogo. No caso do modo história é também apresentada a opção de voltar ao mundo de jogo.

[RF014] Voltar ao menu principal

Descrição do caso de uso: Este caso de uso permite que o utilizador voltar ao menu principal enquanto está a decorrer uma sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e deve ter o *in-game menu* aberto. De seguida escolher a opção de sair (*home*) e confirmar.

Saídas e Pós-condições: O sistema termina a sessão de jogo atual e volta para o menu principal.

[RF015] Desativar/ativar som

Descrição do caso de uso: Este caso de uso permite ao utilizador desativar/ativar o som enquanto está a decorrer uma sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e deve ter o *in-game menu* aberto. De seguida escolher a opção de desativar/ativar som.

Saídas e Pós-condições: Caso o som esteja ativo este é desativado, caso contrário é ativado.

[RF016] Apresentar pontuações após *gameover*

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize a pontuação obtida numa sessão de jogo do modo arcade e o seu recorde, após derrota.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve ter perdido todas as vidas na sessão atual de jogo do modo arcade.

Saídas e Pós-condições: O sistema deve apresentar a pontuação obtida e o recorde pessoal do jogador. Se o jogador estiver autenticado com uma conta *Facebook* deverá ser apresentada a tabela de *high-scores*.

[RF017] Apresentar pontuações após completar o jogo

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize a pontuação obtida numa sessão de jogo do modo arcade e o seu recorde, após concluir este modo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve ter completado todos os níveis sem ter ficado sem vidas na sessão atual de jogo do modo arcade.

Saídas e Pós-condições: O sistema deve apresentar a pontuação obtida e o recorde pessoal do jogador. Se o jogador estiver autenticado com uma conta *Facebook* deverá ser apresentada a tabela de *high-scores*.

[RF018] Apresentar pontuações após concluir um nível do modo história

Descrição do caso de uso: Este caso de uso permite que o utilizador visualize a pontuação obtida numa sessão de jogo do modo história e o seu recorde, após concluir o nível em causa.

Prioridade: Essencial.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve ter completado o nível em causa na sessão de jogo do modo história.

Saídas e Pós-condições: O sistema deve apresentar a pontuação obtida e o recorde pessoal do jogador. Se o jogador estiver autenticado com uma conta *Facebook* deverá ser apresentada a tabela de *high-scores* para esse nível.

[RF019] Voltar ao menu principal após terminar a sessão de jogo

Descrição do caso de uso: Este caso de uso permite que o utilizador volte ao menu principal após ter terminado a sessão de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve ter terminado a sessão de jogo. Deve clicar no botão *home* ou no botão de sistema *back*.

Saídas e Pós-condições: O sistema deve apresentar o menu principal.

[RF020] Aceder opções

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda ao menu de opções.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu principal e deve clicar no botão associado às opções.

Saídas e Pós-condições: O sistema apresenta o menu com as opções: ativar/desativar música e efeitos sonoros, seleccionar língua, ativar/desativar sincronização, efetuar *reset*, mostrar créditos.

[RF021] Alterar opções

Descrição do caso de uso: Este caso de uso permite que o utilizador altere as opções de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de opções e deve clicar no botão associado à opção desejada.

Saídas e Pós-condições: O sistema apresenta a opção alterada.

[RF022] Efetuar reset

Descrição do caso de uso: Este caso de uso permite que o utilizador faça *reset* a todos os dados do jogo (*achievements*, opções, se já viu o tutorial alguma vez, ...).

Prioridade: Importante.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de opções, deve clicar na opção associada ao reset e confirmar.

Saídas e Pós-condições: O sistema efetua um reset aos dados.

[RF023] Ativar sincronização

Descrição do caso de uso: Este caso de uso permite que o utilizador ative a sincronização de dados.

Prioridade: Importante.

Classificação: Novo.

Entradas e Pré-condições: O utilizador deve ter efetuado *login* no *Facebook*, estar no menu de opções, ter a opção de sincronização desativa, clicar na opção associada à sincronização e confirmar.

Saídas e Pós-condições: O sistema efetua a sincronização de dados com o servidor.

[RF024] Desativar sincronização

Descrição do caso de uso: Este caso de uso permite que o utilizador desative a sincronização de dados.

Prioridade: Importante.

Classificação: Novo.

Entradas e Pré-condições: O utilizador estar no menu de opções, ter a opção de sincronização ativa e clicar na opção associada à sincronização.

Saídas e Pós-condições: O sistema desativa a opção de sincronização com o servidor.

[RF025] Mostrar créditos

Descrição do caso de uso: Este caso de uso permite que o utilizador veja os créditos.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de opções e deve clicar na opção associada aos créditos (“CREDITS” no caso da língua inglesa).

Saídas e Pós-condições: O sistema apresenta os créditos.

[RF026] Aceder ao menu de *achievements* (cacifo)

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda ao menu de *achievements*.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu principal e deve clicar na opção associada aos *achievements*.

Saídas e Pós-condições: O sistema apresenta o menu de *achievements*.

[RF027] Consultar *achievements*

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda à descrição/informação de cada *achievement*.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de *achivements* e deve clicar no *achievement* que deseja consultar.

Saídas e Pós-condições: O sistema apresenta a descrição/informação sobre o respectivo *achievement*.

[RF028] Aceder ao menu de cromos colecionáveis

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda ao menu de cromos colecionáveis.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de *achievements* e deve clicar na opção associada aos cromos colecionáveis.

Saídas e Pós-condições: O sistema apresenta o menu de *achievements*.

[RF029] Consultar cromos colecionáveis

Descrição do caso de uso: Este caso de uso permite que o utilizador aceda à descrição/informação de cada cromo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu de cartas colecionáveis e deve clicar no cromo que deseja consultar.

Saídas e Pós-condições: O sistema apresenta a descrição/informação sobre o respectivo cromo.

[RF030] Partilhar cromos colecionáveis no *Facebook*

Descrição do caso de uso: Este caso de uso permite que o utilizador partilhe um cromo no *Facebook*.

Prioridade: Importante.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve ter efetuado *login* no *Facebook*, estar no menu de cartas colecionáveis, clicar no cromo que deseja partilhar e clicar na opção de partilhar no *Facebook*.

Saídas e Pós-condições: O cromo é partilhado no *Facebook*.

[RF031] Sair do jogo na totalidade

Descrição do caso de uso: Este caso de uso permite que o utilizador saia por completo do jogo.

Prioridade: Importante.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar no menu principal e, deve clicar num dos seguintes:

- Botão de sistema *back* (*Android*);
- Tecla "Esc".

Saídas e Pós-condições: O sistema fecha por completo o jogo.

3.3 Jogabilidade

[RF001] Controlar a personagem

Descrição do caso de uso: Este caso de uso permite que o utilizador controle a personagem principal: Radamel Falcao.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e, deverá usar um dos seguintes:

- *Touch*;
- Rato;
- Teclado (teclas “w”, “a”, “s”, d” ou as setas).

Saídas e Pós-condições: O sistema movimenta a personagem consoante os comandos de entrada.

[RF002] Acertar e causar dano nos inimigos

Descrição do caso de uso: Este caso de uso permite que o utilizador acerte e cause dano nos alienígenas.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e, deverá acertar na bola com o personagem de forma a que esta colida com o(s) alienígena(s).

Saídas e Pós-condições: O sistema detecta a colisão da bola com o(s) alienígena(s) e retira-lhe(s) a vida correspondente. Ao atingir zero de vida o alienígena morre e desaparece.

[RF003] Impedir personagem de sair do ecrã

Descrição do caso de uso: Este caso de uso impede que o jogador faça a personagem sair do terreno de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e controla o personagem.

Saídas e Pós-condições: O sistema detecta que o movimento do personagem ultrapassará o terreno de jogo e fá-lo parar no limite, mantendo-o assim no terreno.

[RF004] Impedir a bola saia do ecrã

Descrição do caso de uso: Este caso de uso impede que a bola saia do terreno de jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e a bola está em movimento.

Saídas e Pós-condições: O sistema detecta que a bola colidiu com o limite do ecrã e aplica-lhe a física correspondente a ter colidido com uma parede.

[RF005] Restantes colisões da bola

Descrição do caso de uso: Este caso de uso representa as colisões da bola com alienígenas, barreira da baliza, postes da baliza, personagem principal.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e a bola está em movimento.

Saídas e Pós-condições: O sistema detecta que a bola colidiu com os elementos referidos e aplica-lhe a física correspondente a ter colidido com uma parede.

[RF006] Perder uma vida

Descrição do caso de uso: Este caso de uso descreve o cenário em que ocorre a perda de vidas por parte do personagem principal.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e a bola está em movimento.

Saídas e Pós-condições: O sistema detecta que a bola ultrapassou o Falcao, estando nas suas costas e dirigindo-se para o limite inferior do ecrã. É removida a bola e retirada uma vida ao jogador.

[RF007] Gameover

Descrição do caso de uso: Este caso de uso descreve o cenário em que o jogador perdeu todas as vidas numa sessão de jogo, obtendo *gameover*.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo, não lhe restarem vidas extra e o jogador perder a última bola.

Saídas e Pós-condições: É removida a bola e como não há vidas restantes, é apresentado o cenário de *gameover*.

[RF008] Nível completado com sucesso

Descrição do caso de uso: Este caso de uso descreve o cenário em que o jogador obteve sucesso no nível atual.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições: O utilizador deve estar numa sessão de jogo e encaminhar a bola para dentro da baliza.

Saídas e Pós-condições: O sistema detecta que a bola entrou na baliza, apresenta os festejos de golo. O nível é dado como completado. Caso seja o modo arcade o jogador é encaminhado para o nível seguinte. Caso seja o modo história ou o modo desafio, o jogador é encaminhado para a tela de pontuações.

[RF009] Jogo completado com sucesso

Descrição do caso de uso: Este caso de uso descreve o cenário em que o jogador obteve sucesso em todos os níveis atingindo o final do jogo.

Prioridade: Essencial.

Classificação: Mantido.

Entradas e Pré-condições:

Caso 1 - O utilizador deve estar numa sessão de jogo do modo arcade e ter completado todos os níveis.

Caso 2 - O utilizador deve estar numa sessão de jogo do modo história e completar o último nível pela primeira vez.

Saídas e Pós-condições: O sistema apresenta uma animação e de seguida são apresentadas as pontuações obtidas.

4. Requisitos Não-funcionais

4.1 Interface

[RNF001] Interface Gráfica

Descrição do caso de uso: O videogame deverá oferecer uma interface gráfica.

Prioridade: Essencial.

Classificação: Mantido.

[RNF002] Interface Intuitiva

Descrição do caso de uso: O videogame deverá oferecer ao jogador a possibilidade de usufruir das suas funcionalidades principais de forma rápida e fácil.

Prioridade: Essencial.

Classificação: Mantido.

[RNF003] Resolução

Descrição do caso de uso: O videogame deverá oferecer suporte todas as resoluções (verticais e horizontais) suportadas pelas plataformas.

Prioridade: Importante.

Classificação: Alterado.

4.2 Outros Aspectos

[RNF001] Performance

Descrição do caso de uso: O videogame deverá ser capaz de executar a 60 *frames* por segundo sem quebras, a não ser que a plataforma não suporte.

Prioridade: Importante.

Classificação: Mantido.

[RNF002] Plataformas suportadas

Descrição do caso de uso: O videogame deverá executar pelo menos nas seguintes plataformas: *Android, iOS, Windows Phone, Windows, Facebook*, navegadores atuais.

Prioridade: Essencial.

Classificação: Alterado.

[RNF003] Espaço físico

Descrição do caso de uso: O *apk* do videogame para *Android* não poderá exceder os 50Mb de espaço físico.

Prioridade: Essencial.

Classificação: Mantido.

[RNF004] Memória

Descrição do caso de uso: O videogame não poderá exceder os 200Mb de memória em dispositivos móveis.

Prioridade: Essencial.

Classificação: Mantido.

[RNF005] Ferramenta de desenvolvimento

Descrição do caso de uso: O videogame deverá ser desenvolvido utilizando o motor de jogo *Unity*.

Prioridade: Importante.

Classificação: Alterado.

8.3. Anexo C - Modelo de Mensagens *JSON* Trocadas com o Servidor - “Falcao vs Aliens”

- **JSON Template - Pedido de Registo**

```
{
  "type": int,                // 0
  "user" :                    // user information
  {
    "id": string,             // FB user ID
    "first_name": string,     // user first name
    "last_name": string,      // user last name
    "gender": bool,           // user gender
    "email": string,          // user email
    "location": string,       // user current location
    "birthday": string        // user birthday
  },
  "friends": [string, string, ...] // user friends' FB ID
}
```

- **JSON Template - Resposta ao Pedido de Registo**

```
{
  "type": int,                // 0
  "success": bool,            // the success of the request
  "user_id": int,             // user ID on database
  "user_name": string,        // user name on database
  "error": string             // error message if there is any error
}
```

- **JSON Template - Envio de Pontuação**

```
{
  "type": int,                // 1
  "user_id": int,             // user ID on database
  "level": int,               // -1 arcade mode / (0 - 17) story mode levels
  "score": int                // user's top score
}
```

- **JSON Template - Resposta ao Envio de Pontuação**

```
{
  "type": int,                // 1
  "success": bool,            // the success of the request
  "score": int,               // the best score that player did
  "error": string             // error message if there is any error
}
```

- **JSON Template - Pedido do TOP 5 Melhores Pontuações**

```
{
  "type": int,           // 2
  "level": int,         // -1 arcade (0 - 17) levels
  "month": int,        // the month to search for
  "year": int          // the year to search for
}
```

- **JSON Template - Resposta ao Pedido do TOP 5 Melhores Pontuações**

```
{
  "type": int,           // 2
  "success": bool,      // the success of the request
  "error": string,      // error message if there is any error
  "scores":             // list of the 5 highest scores
  [
    {
      "user_id":         // user ID on database
      "name": string,    // user name
      "score": int       // user top score
    }
  ]
}
```

- **JSON Template - Pedido de Sincronização**

```
{
  "type": int,           // 3
  "user_id": int,       // user ID on database
  "settings":
  {
    "total_achievements": int,
    "achievement0": int,
    "achievement1": int,
    (...)
    "achievement16": int,
    "achievement17": int,

    "story_level_reached": int,
    "max_level_reached": int,
    "matches_won": int,
    "matches_lost": int,
  }
  "levels":
  {
    "arcade": int,

    "total_levels": int,
    "level0": int,
    "level1": int,
  }
}
```



```

    (...)
    "level16": int,
    "level17": int
  }
}

```

- **JSON Template - Resposta ao Pedido de Sincronização**

```

{
  "type": int,                // 3
  "success": bool,           // success of the request
  "error": string,           // error message if there is any error
  "settings":
  {
    "total_achievements": int,
    "achievement0": int,
    "achievement1": int,
    (...)
    "achievement16": int,
    "achievement17": int,

    "story_level_reached": int,
    "max_level_reached": int,
    "matches_won": int,
    "matches_lost": int,
  }
  "levels":
  {
    "arcade": int,

    "total_levels": int,
    "level0": int,
    "level1": int,
    (...)
    "level16": int,
    "level17": int
  }
}

```

- **JSON Template - Pedido da Lista de Desafios**

```

{
  "type": int,                // 4
  "user_id": int,            // user ID on database
}

```

- **JSON Template - Resposta ao Pedido da Lista de Desafios**

```
{
  "type": int,                // 4
  "success": bool,           // success of the request
  "error": string,           // error message if there is any error
  "challenges":
  [
    {
      "challenge_id": int,    // challenge ID on database
      "timestamp": long,     // date
      "state": int,          // challenge current state
      "user_id01": int,      // user 1 ID on database
      "user_id02": int,      // user 2 ID on database
      "user_score01": int,   // user 1 score
      "user_score02": int,   // user 2 score
    }
    {
      (...)
    }
    (...)
  ]
}
```

- **JSON Template - Pedido do Histórico de Desafios**

```
{
  "type": int,                // 5
  "user_id": int,             // user ID on database
}
```

- **JSON Template - Resposta ao Pedido do Histórico de Desafios**

```
{
  "type": int,                // 5
  "success": bool,           // success of the request
  "error": string,           // error message if there is any error
  "challenges":
  [
    {
      "challenge_id": int,    // challenge ID on database
      "timestamp": long,     // date
      "state": int,          // challenge current state
      "user_id01": int,      // user 1 ID on database
      "user_id02": int,      // user 2 ID on database
      "user_score01": int,   // user 1 score
      "user_score02": int,   // user 2 score
    }
  ]
}
```

```

        {
            (...)
        }
        (...)
    ]
}

```

- **JSON Template - Novo Desafio**

```

{
    "type": int,                // 6
    "timestamp": long,         // date
    "user_id01": int,          // user 1 ID on database
    "user_id02": int           // user 2 ID on database
}

```

- **JSON Template - Resposta Novo Desafio**

```

{
    "type": int,                // 6
    "challenge_id": int,        // challenge ID on database
    "success": bool,           // success of the request
    "error": string            // error message if there is any error
}

```

- **JSON Template - Envio Resultado do Desafio**

```

{
    "type": int,                // 7
    "challenge_id",            // challenge ID on database
    "user_id": int,           // user ID on database
    "score", int              // user score on the challenge
}

```

- **JSON Template - Resposta Envio Resultado do Desafio**

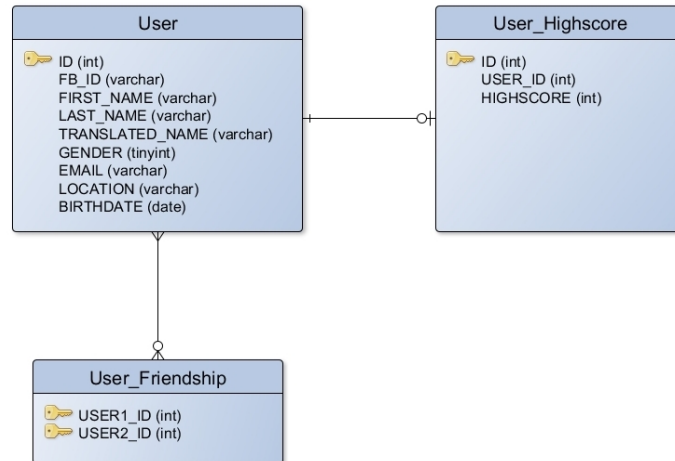
```

{
    "type": int,                // 7
    "challenge_id",            // challenge ID on database
    "success": bool,           // success of the request
    "error": string,          // error message if there is any error
    "state": int,              // challenge current state
    "user_id01": int,          // user 1 ID on database
    "user_id02": int,          // user 2 ID on database
    "user_score01": int,       // user 1 score
    "user_score02": int       // user 1 score
}

```

8.4. Anexo D - Diagramas de Entidade e Relacionamento - Bases de Dados Original e da Nova Versão - “Falcao vs Aliens”

“Falcao vs Aliens” Versão Original - Diagrama ER



“Falcao vs Aliens” Versão Original - Diagrama ER

