LIL'WITCH

A Design Case Study of Animation in Games

Universidade de Coimbra
Mestrado em Design e Multimédia
Faculdade de Ciências e Tecnologia
Gustavo Campos de São Pedro Barroso
Fevereiro de 2017

Lil'Witch
A Design Case Study of Animation in Games

Universidade de Coimbra
Mestrado em Design e Multimédia
Faculdade de Ciências e Tecnologia
Gustavo Campos de São Pedro Barroso
Fevereiro de 2017

Orientação
Antonio Manuel Sucena Silveira Gomes
Licinio Gomes Roque

Júri
Maria Alice Barriga Geirinhas dos Santos
Juri Vogal: Bruna Raquel Santos Sousa

Índice

1. Introdução

Com a evolução da tecnologia dos computadores nos recentes anos a animação 3D tornou-se mais viável e adquiriu maior complexidade exigindo múltiplas e mais profundas competências que requerem a quem quer abordar esta nova tecnologia, um estudo/formação aprofundado e permanente, bem como recursos dispendiosos. Já a Animação 2D, apesar de ter uma coleção de técnicas refinadas, é mais acessível a animadores principiantes. Recentemente, têm sido publicados, com sucesso, produtos cinematográficos ou jogos, com combinações de Animação 3D com Animação 2D. Estas combinações abrem novos horizontes no campo de Animação, temática que esta dissertação pretende explorar, estudar e aplicar num jogo.

1.1. Enquadramento

A dissertação pertence ao tema de Design de Jogos, com planeamento e aplicação de design, regras e estética na criação de um jogo, por forma a que este ofereça a melhor experiencia possível ao jogador.
Um dos objetivos de design de jogos é que se estabeleça uma conexão entre o jogo e os seus jogadores; para tal a sua animação exige cuidados extras. Tornando-se consequentemente os jogadores mais sensíveis à animação do jogo, que se transformará numa plataforma ideal para explorar e testar animação.

1.2. Motivação

A pesquisa e estudo de combinação 2D com 3D é por si só extremamente motivante para os desenvolvedores do jogo, dado que os ajuda a adaptar o seu processo de desenvolvimento, às suas habilidades e recursos por forma a alcançarem novos resultados estéticas e mecânicas de *gameplay*.
Motivante também, é constatar que com a execução deste trabalho, surge-nos a oportunidade de estudar extensivamente os temas de animação e multimédia 3D, tema que não tive durante a frequentação do curso. Assim, surge-nos uma proposta verdadeiramente desafiante, a combinação de 2D com 3D num jogo, aventura graficamente, para mim como designer de multimédia.

1.3. Objectivo

O objectivo desta dissertação é a exploração da combinação da animação 2D com 3D, como esta oferece o melhor do 2D e 3D, tendo em atenção como pode ser adaptável à equipas de diferentes competências, superar estética sem volume e até criar novos estilos de *gameplay.* Para tal pretende-se colecionar e processar informação de animação 2D, 3D e analisar casos de estudo de produtos com técnicas que usufruem animação 2D com 3D. Tem como propósito ajudar e orientar a construção do design de um jogo, designado por Lil' Witch, e a sua aplicação num protótipo composto por um nível curto.

1.4. Estrutura do Documento

A Estrutura desta dissertação é dividida nos seguintes cápitulos:

- A Introdução que apresenta uma pequena explicação da Tese, incluindo os seus objectivos, a motivação e execução prática.

- O Estado da Arte, que contém a informação pesquisada ao longo da dissertação, relevante à elaboração do design do jogo e o seu processo de implementação, tal como conceitos-chave, tecnologias e técnicas.

- A abordagem Metodológica, explicita o processo para atingir os objetivos propostos pela Dissertação, incluindo a calendarização e desafios do desenvolvimento da prova de conceito, o jogo.

- A Proposta de Trabalho, indica o método inicial de implementação, o conceito, design e as técnicas que se pretende pôr em prática, para criar o protótipo do jogo.

- A Implementação do jogo, contem experiências que reformataram o design do jogo e o seu processo de criação, tal como o design das animações e personagens, as técnicas e tecnologias implementadas.

Os Testes de apreciação que disponibilizam os testes de avaliação do jogo pelo publico, incluindo a análise dos seus resultados e sua contribuição para o aperfeiçoamento do jogo

- A Conclusão, que encerra esta dissertação com a reflexão do seu resultado, as dificuldades encontradas e as perspectivas futuras.

2. Estado de Arte

Neste capítulo, serão abordados vários temas que servirão como base de compreensão das técnicas de animação usadas nas áreas 2D, 3D e jogos. Este capítulo está dividido em cinco subcapítulos:

- O primeiro, aborda os princípios básicos de animação, e começa com o procedimento básico da criação de animação. São definidos os 12 Princípios Básicos de Animação e mencionadas algumas ferramentas de suporte do planeamento da animação.

- O segundo, contém técnicas básicas de animação e analisa as características de animação de jogos que a distingue da animação de cinema.

- O terceiro, é dedicado ao procedimento de construção do projeto 3D - *pipeline* 3, especificamente da análise dos vários passos da fase de Produção, tais como Modelação, Textura, *Rigging*, Animação, *VFX*, Iluminação e Renderização. Nestes passos também se incluíram algumas das características especificas dos jogos.

- O quarto, inclui informação sobre o estado da indústria de animação, e as vantagens e desvantagens entre animação 2D e 3D.

- O quinto, é composto pela recolha e a análise de técnicas de animação, incluindo as suas vantagens, desvantagens e exemplos de projetos que exibem o resultado destas técnicas.

2.1 Princípios Básicos de Animação

> *"For some presumptous reason, man feels the need to create something of his own that appears to be living, that has an inner strenght, a vitality, a separate identity - something that speaks out with authority-- a creation that gives the ilusion of live."*
> **Thomas F. & Johnston O. (1981) The illusion of life. Italy, Walt Disney Productions**

Animação é dar vida às personagens e objetos do ecrã. Tecnicamente é uma sequência de imagens, designadas por fotograma, que ilustram uma personagem em diferentes poses. Esta sequência engana a perceção do público levando-o a acreditar que a personagem está em movimento. Tipicamente um vídeo animado tem 24 *fps*. Como tal um vídeo de 90 minutos teria 129,600 imagens que têm que ser criadas e analisadas para se certificar que estão consistentes. Portanto, animação é um processo moroso e que exige muito trabalho. Na procura de manter a consistência da ilustração da personagem ao longo dos fotogramas, estabeleceu-se um conjunto de princípios para guiar os animadores, que são os 12 Princípios Básicos de Animação, analisados e refinados ao longo do tempo pelos estúdios da Walt Disney. A saber:

1. Squash and Stretch - É um dos princípios mais importantes, por introduzir certas deformações na personagem ao longo da animação que ajuda a exibir peso e flexibilidade no seu movimento. Um dos exercícios básicos para novos animadores, é animar os saltos numa bola, porque é considerado perfeito e simples para testar este principio. Quando a bola cai no chão, por breves momentos fica espalmada no chão, ou seja *squashed*, assim que a bola está prestes a saltar, a sua forma começa a esticar, ou seja *stretch*.

2. Slow In and Slow Out - Este principio indica que nenhum objeto para imediatamente, ou seja, o animador tem que ter em conta a aceleração e desaceleração em movimentos, que por norma, se exageram em animação. Este principio também pode ser facilmente observado no teste básico dos saltos numa bola. Por exemplo, quando a bola cai, a distância da bola entre os fotogramas não é muito grande, logo o seu movimento ainda é lento, mas quanto maior for a distância da bola entre fotogramas maior é sua velocidade (aceleração).

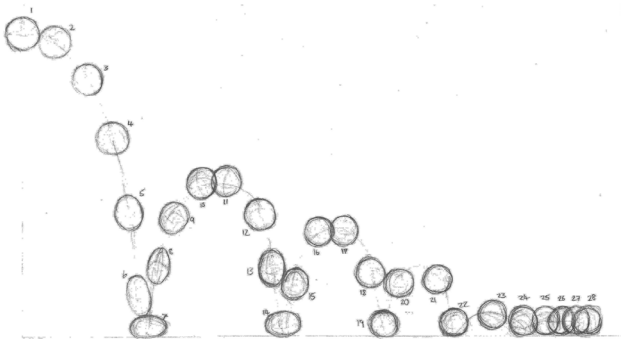

Fig. 1- Teste básico de animação. Recuperado em https://indiaholdenap.wordpress.com/2014/09/30/bouncing-balls/

3. Anticipation - Indica que no início da animação de uma personagem, a audiência tem que conseguir perceber e antecipar o seu movimento. A criação da antecipação, é possível ao preceder qualquer ação da personagem com a introdução de um conjunto de movimentos ou poses no início do seu movimento. Neste exemplo, antes de um jogador de basebol atirar a bola, ele primeiro posiciona-se.

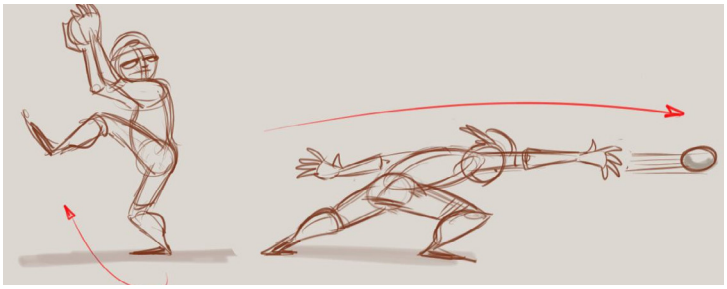

Fig. 2- Exemplo de Anticipation em basebol. Recuperado em http://br260097-sp.blogspot.pt/2015/11/12-principles-of-animation.html

4. Staging - É um principio genérico, presente em diversas áreas, tal como o antigo teatro, televisão... Indica que a apresentação de uma ideia por meios visuais deve ser completamente e inequivocamente clara. De que são exemplo: o artista deve ter cuidado na forma como organiza e compõe o cenário, certificando-se que não existe nenhum elemento que possa roubar a atenção do público em relação às ações da personagem. A câmara deve focar-se na conversa entre personagens e não em elementos do cenário irrelevantes para a narrativa. Em momentos cruciais da narrativa, a pose das personagens deve corresponder à sua expressão, atitude, estado emocional. O sucesso deste exemplo pode ser verificado se a silhueta da pose expressa claramente a emoção da personagem.

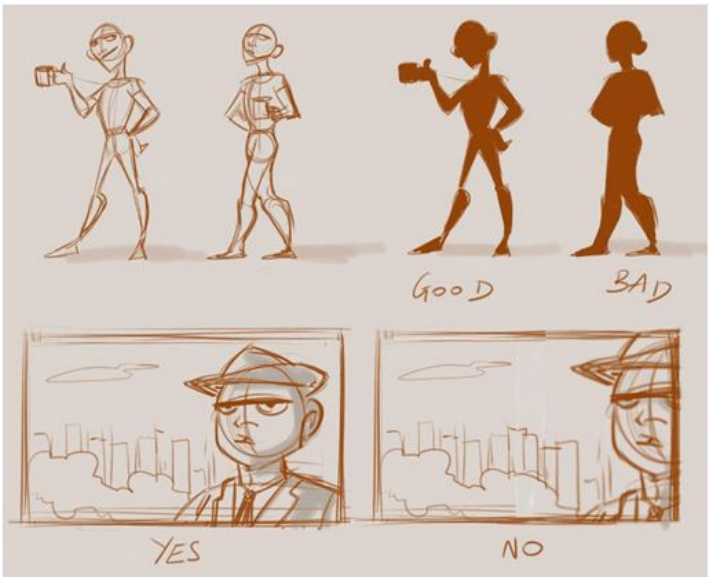

Fig. 3- Exemplos de Staging. Recuperado em http://anisfitrisya.blogspot.pt/2012/10/12-principles-of-animation.html

5. Timing- É calcular quanto tempo é necessário para fazer uma ação e representar-la em fotogramas. Por exemplo, supondo que uma pessoa demora 1,5 segundos para se sentar, numa animação de 24 *fps*, teríamos que desenhar a ação em 36 *ilustrações.*

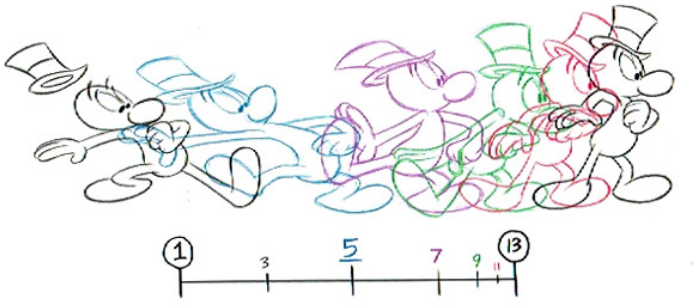

Fig. 4- Exemplo de Timing, acompanhado com uma *timeline* cujos segundos correspondem as poses. Recuperado em http://animatedviews.com/2008/eric-goldberg-animation-book-interview/

6. Straight Ahead Action and Pose to Pose - São abordagens diferentes ao estilo de animação tradicional. *Straight Ahead Action* impõe que os fotogramas sejam ilustrados desde o início até ao fim. Enquanto *Pose to Pose* impõe que os *key poses* da personagem sejam ilustrados primeiro, só depois é que os movimentos entre as *key* poses são ilustrados. Naturalmente cada estilo tem vantagens e desvantagens. Enquanto o *Straight Ahead Action* cria ações mais dinâmicas, fluidas, espontâneas, por exemplo permitindo erros que adicionam personalidade à animação. Enquanto *Pose to Pose*, a animação pode ser facilmente planeada, controlada de modo a ter um bom sincronismo, não sendo necessário descartar a animação inteira se esta tiver um erro e capaz de ser avaliada e aprovada com antecedência, porque é possível demonstrar uma ideia genérica da animação através das suas *key* poses.

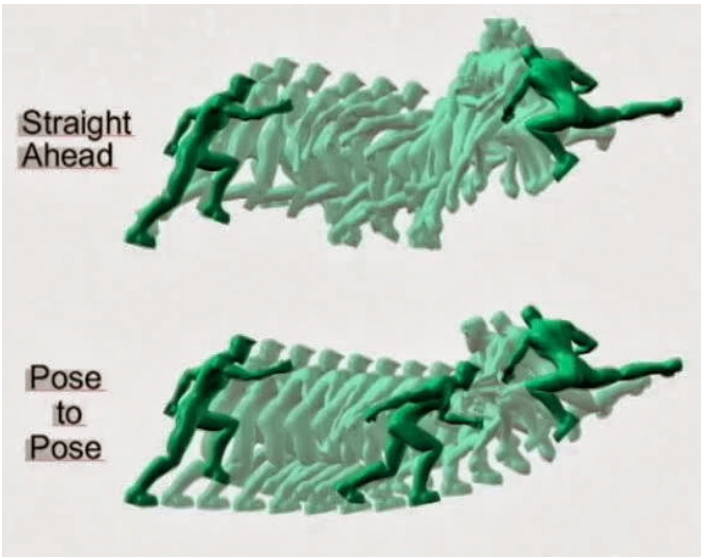

Fig. 5- Exemplo de comparação entre Straight Ahead Action e Pose to Pose. Recuperado em http://anetematvejeva3d.blogspot.pt/2013/10/reflecting-on-12-principles-of-animation.html

7. Follow Throught and Overlapping Action

Follow Throught indica que quando um objeto em movimento para bruscamente, partes do seu corpo continuam em movimento antes de pararem completamente. Por exemplo, quando uma pessoa com um cabelo longo para de correr, o seu cabelo continua a mover-se, e só mais tarde fica parado.

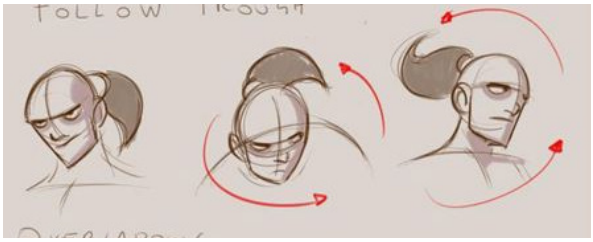

Fig. 6- Exemplo de Follow Throught com cabelo. Recuperado em https://owenrye.wordpress.com/2014/02/17/comm-345-animation-communication-blog-3-principles/

Overlapping Action relembra que partes num corpo podem ter velocidades diferentes. Por exemplo, enquanto uma personagem gorda está a correr, a sua barriga agita-se a uma velocidade diferente da do corpo.

Fig. 7- Exemplo de Overlapping Action com cabelo. Recuperado em https://owenrye.wordpress.com/2014/02/17/comm-345-animation-communication-blog-3-principles/

8. Arcs - As ações de pessoas, criaturas, e algumas vezes objetos, têm um movimento curvilíneo.

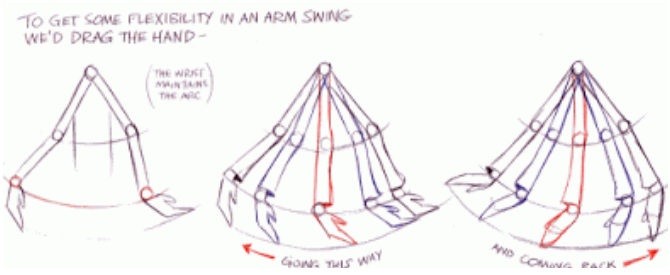

Fig. 8- Exemplo de Arcs Recuperado em http://idearocketanimation.com/2592-principles-of-animation-arcs/

9. Secondary Action - Enquanto há uma ação principal, esta será suportada, fortificada por uma ação secundaria. Por exemplo, quando a ação principal for chorar, a secundaria pode ser limpar as suas lágrimas com um lençol.

Fig. 9- Exemplo de ação com e sem Secondary Action. Recuperado em https://jordannwharton.wordpress.com/year-2/unit-67-3d-animation/12-principles-of-animation/

10. Exaggeration - É a intenção de tornar o movimento da animação mais extrema, exagerada, de modo a ser mais vibrante e viva, como tal as ações das personagens tornam-se mais claras e visíveis ao público.

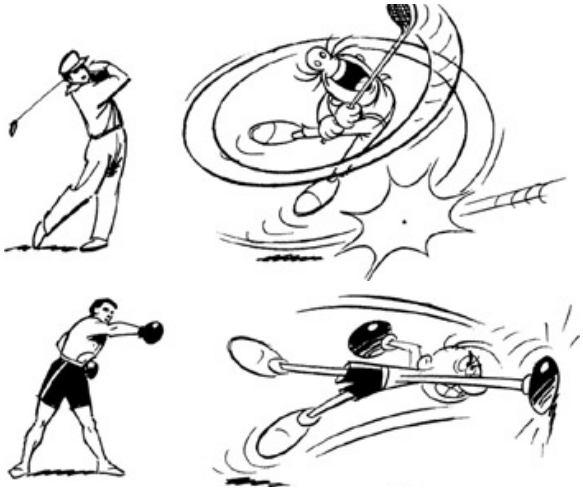

Fig. 10- Exemplos de Exaggeration. Recuperado em https://caworld3.wordpress.com/2013/04/21/12-principle-of-animation-examples/

11. Solid Drawing - É a importância do conhecimento de conceitos básicos e técnicos necessários para animação, tal como conhecer anatomia dos corpos, conseguir desenhar uma personagem de várias perspetivas, o que era antigamente priorizado nos estudos da Disney. Afinal, conhecer estes conceitos permite uma animação rápida e compreensível.

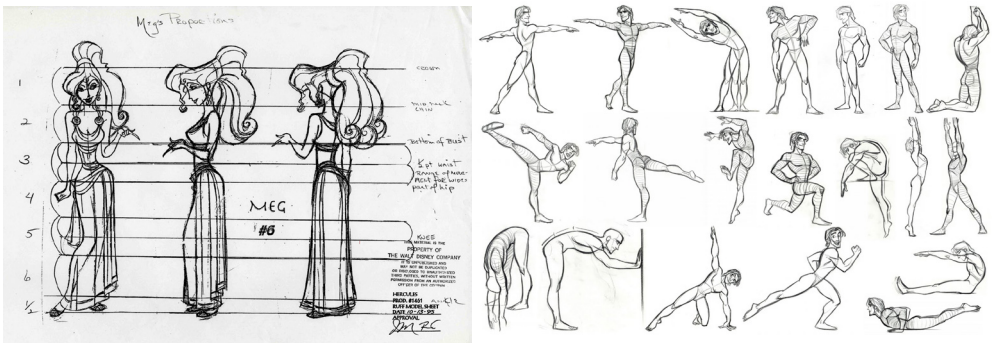

Fig. 11- Exemplos de Solid Drawing. Recuperado em http://livlily.blogspot.pt/2010/10/hercules-1997.html e http://www.chroniquedisney.fr/dossier/2011-confraiponce.htm

12. Appeal - É um princípio importante em animação, a personagem tem que ser apelativa, a audiência tem que ter a capacidade de conectar-se com a personagem. Os movimentos da personagem e o seu design visual devem conseguir representar a sua personalidade. Por exemplo, personagens com cabeças e olhos de maiores proporções em relação ao corpo parecem ser mais jovens e ganham mais simpatia da audiência, ou vilões cujo design tem que os tornar antipáticos à audiência.

"This new art of animation had the power to make the audience actually feel the emotions of a cartoon figure"
Thomas F. & Johnston O. (1981) The illusion of life. Italy, Walt Disney Productions

Fig. 12- Exemplo de Appeal do filme Hercules. Recuperado em http://www.hypable.com/once-upon-a-time-season-5-cast-hades/ e http://difundir.org/2016/05/08/11-personajes-de-disney-que-podrian-ser-gay-lo-habias-pensado/

Estes princípios foram utilizados do estudo e criação eficiente da animação de movimentos realistas das personagens do jogo.

2.2. Animação: Análise Comparativa do Cinema com os Jogos

O objetivo de um animador não é criar uma obra de arte mas criar movimento, como tal é aceitável deformar o modelo em ordem a obter um movimento fluido. Estas deformações podem ser por exemplo:

Breaking body- alterar a composição do modelo de tal modo que pareça que estamos a deslocar/ partir o corpo do modelo.

Smear- substituir partes do corpo do modelo por borrões, de modo a representar a trajetórias de movimento demasiado rápidos ao olhos da audiência.

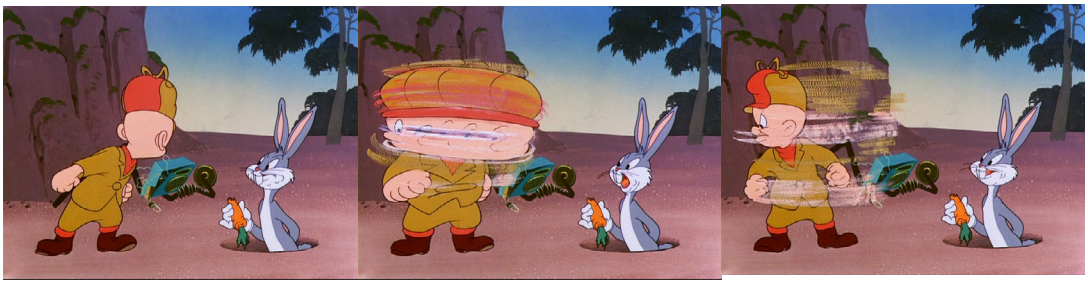

Fig. 13- Exemplo de *Smear* de looney tunes. Recuperado em http://tralfaz.blogspot.com/2012/12/hare-do-smears.html

"Remember your foundations as an artist but don´t be afraid to break a few bones if it´s gonna help your motion look better, remember you're creating movement, not individual pieces of art"
Mariel Cartwright (2016) Making Fluid and Powerful Animations For Skullgirls from https://www.youtube.com/watch?v=Mw0h9WmBlsw

Em animação cinemática, o animador controla a posição e direção da câmera, conhece a história do filme e possui controlo total das personagens, de modo a que consigam criar e refinar animações com personalidade, apelo visual, emocional. Mas em jogos, os animadores têm de ter em consideração a presença de um jogador e das ferramentas, mecânicas do jogo que são afetadas ou afetam a animação.

Ao contrário dos filmes em que a animação tem que ser visualmente agradável e parecer natural, em jogos prioriza-se que o jogador se sinta bem com a animação. O que resulta que o jogador sinta que a animação seja "*responsive*", flexível, instantânea às suas intenções. No entretanto, o jogador necessita que as animações dos *npc* sejam fáceis de ler, de analisar, de modo a que ele consiga rapidamente prever e reagir à ação do *npc*.

Utilizando a animação de um golpe como exemplo, podemos estudar as diferenças entre um soco num *npc* em relação ao soco do protagonista, a personagem controlada pelo jogador. Antecipação do soco do Protagonista, os movimentos do corpo necessários para formar o soco, tem que ser quase instantânea, de forma a que o soco coincida com o momento preciso em que o jogador queria dar o soco. Mas isto torna a animação muito rápida e difícil de ser processada pelo jogador, como tal usa-se o princípio de "follow through" e *smear* na animação para completar a informação do movimento ao jogador. Enquanto que a Antecipação do soco do *npc* deve ser suficientemente longa de modo a que seja possível com que o jogador consiga notar a ação e prever que é um soco.

Tendo em conta que jogadores são muito sensíveis à animação, é importante garantir que eles sentem peso, impacto dos movimentos das personagens. Isto pode ser realizado estendendo o momento de *hitspot* e introduzindo *overshoot* na animação

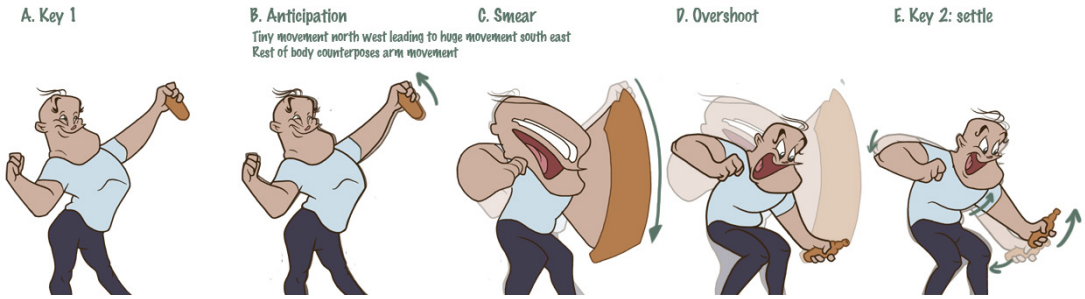

Fig. 14- Exemplo com hitspot e overshoot. Recuperado em http://toonamir.blogspot.pt/2011/08/influences-on-barley-way-more-on.html

2.3. A Animação no Desenvolvimento de Jogos

Com a evolução da área de animação, naturalmente começaram aparecer projetos de animação grandes e complexos, e como tal, eram necessários novas ferramentas para os planear e novos modelos de produção.

O Storyboard é uma ferramenta que permite a representação visual da história do vídeo, como se fosse uma banda desenhada. Inclui ideias iniciais de ângulos da câmara, esboços iniciais dos *fotogramas* importantes do vídeo, com os seus respetivos segmentos do *script*. Em qualquer projeto de animação, é sempre útil a fabricação de um *storyboard*, não só porque permite uma análise do produto final das primeiras fases do projeto, como também é ideal como um guia para as próximas fases de produção, aumentando a coesão das diversas áreas do projeto (por exemplo: som, diálogo, imagens...)

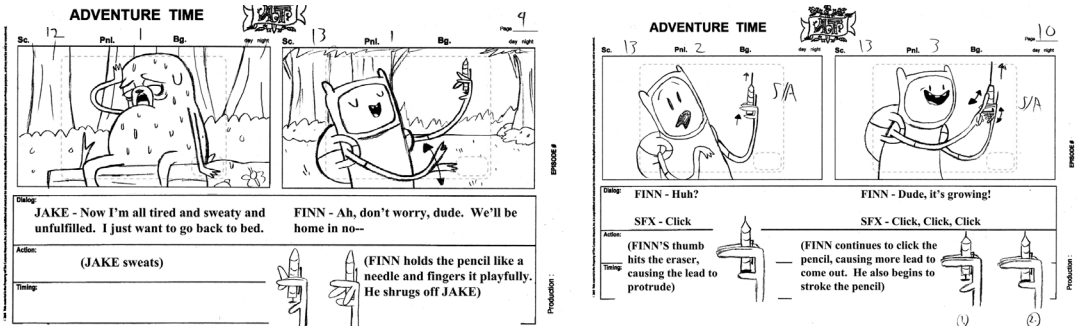

Fig. 15- Exemplos de *storyboard* de Adventure Time. Recuperado em http://andothernonesuch.blogspot.pt/2012/08/adventure-time-storyboard-test.html

A ferramenta Animatic é uma versão animada num *storyboard* em formato de vídeo. Ideal para certificar que ao longo do vídeo a animação tenha um bom fluxo visual. Porque normalmente a animação do vídeo é dividida e trabalhada por diferentes artistas, logo esta ferramenta permite averiguar com antecedência a coesão e consistência das várias animações ao longo do vídeo total.

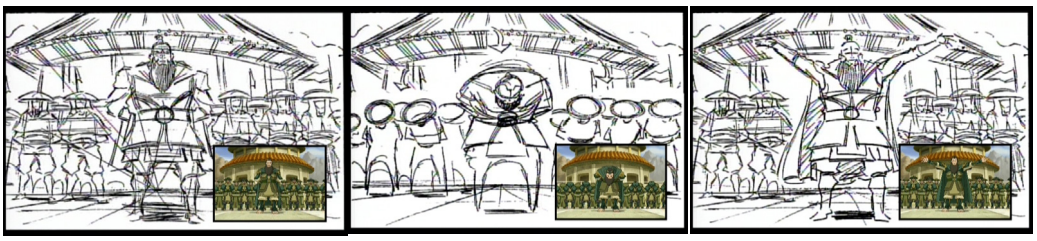

Fig. 16- Exemplos de Animatic de Avatar: The last Airbender. Recuperado em http://piandao.org/screenshots/specials/earth1-animatic/earth1-animatic-4.html

O modelo de 3D *pipeline* que é um processo normalmente utilizado pela indústria para produzir projetos 3D, é composto por três fases principais na seguinte ordem: Pré-produção, Produção e Pos-Produção,

Na produção de projetos 3D, a indústria costuma aderir ao modelo de 3D *pipeline* que é composto por Pré-produção, Produção e Pos-Produção. A qual pretendo implementar na construção do jogo da dissertação.

A **Pré-produção** de todos os projetos começam com uma ideia, história, são construídos *scripts* e *storyboards*, testados com um *animatic* e finalmente decide-se o design final. Estes passos são normalmente realizados por uma equipa de artistas, responsáveis pelo projeto e uma equipa de gestão que cria o plano de produção. Esta fase é importante para o sucesso do projeto, quanto melhor for a preparação, definição das ideias, mais facilmente se previnem complicações ao longo da *pipeline*. De tal modo que existem projetos em que metade do seu calendário é ocupado com a Pré-produção.

Na **Produção** são desenvolvidos os vários elementos do projeto. Se a Pré-produção foi realizada com sucesso, então Produção torna-se mais eficiente; por exemplo, seria contra produtivo se modelos, animações, sons tivessem que ser descartados porque tinha havido uma mudança da história ou design. Naturalmente poderá haver mudanças do design do projeto durante a fase de Produção, mas normalmente são incrementos ou pequenas alterações.

A **Pós-produção** tem como objetivo dar os últimos retoques ao produto final, introduzindo alguns efeitos especiais 2D, corrigir as cores, corrigir erros e ser avaliado, com o propósito de certificar que o produto está pronto para ser publicado.

Considerando que o processo de criação do protótipo do jogo da dissertação recebeu uma grande influência da fase de produção, analisei os seus vários elementos *Layout, R&D, Modeling, Texturing, Rigging, Animation, VFX, Lighting e Rendering*. O estudo destes elementos foram fundamentais para perceber os termos técnicos necessários para o uso de software, e, deteção e correção de erros.

Layout

Criação de *storyboards* e a versão 3D do *animatic* com modelos *proxy*. Esta última componente é vital porque, ao contrário de 2D, esta é necessária para averiguar se os ângulos da câmera a escala das personagens em relação a câmera, ou a distância entre objetos, estão corretos dado que a sua correção no final da *pipeline* não é fácil. Portanto o 3D *animatic* é utilizado como guia para as escalas entre objetos e serve como base para as diversas equipas do projeto ao longo da fase de Produção.
Esta fase serviu como orientação para a a criação de *storyboards* para as animações do jogo.

Research and development (R&D)

Em certos projetos, os artistas podem defrontar-se com dificuldades, ou incapacidades em produzir o projeto, porque ou são limitados pelo software, hardware, ou necessitam de um efeito cujas técnicas, métodos, desconhecem ou necessitam reproduzir um efeito que nunca foi feito na sua indústria. Estes desafios são abordados em *research and development* com o objetivo de encontrar soluções. R&D é abordado ao longo da *pipeline* 3D, onde artistas de diversas áreas em conjunção com diretores técnicos, tentam resolver futuros desafios técnicos do projeto. Por exemplo, no jogo "Rayman legends" (2013) tiverem que conceptualizar e produzir software para facilitar a criação de níveis digitais para artistas. Na produção do vídeo "Paperman" (2012)

tiveram que criar um software para preservar o traço de artista dos esboços nos modelos 3D. No filme "Tangled" (2010), a equipa R&D descobriu um método para que a equipa de animadores conseguisse controlar o cabelo da Rapunzel, de modo a que se comportasse e movimentasse como um cabelo real.

Fig. 17- O cabelo em Tangled [Video]. Recuperado em https://www.youtube.com/watch?v=9K-Gv4XVb10

> *"We're interested in taking the artist's original intent with the stroke they lay down on paper and making sure we can faithfully get that into a computer. So we looked at all the drawing tools out there. All of them. All the commercial tools, all kind of under wraps R&D things for drawing—and we were left unsatisfied. A lot of the tools don't faithfully record what the artist. Almost all of our artists found themselves drawing and re-drawing to try to basically beat the computer into submission—beat the line into what they wanted it to be.."*
> **(Hendrickson A. & Whited B. entrevista por Kaganskiy J. Março 5, 2013, recuperado de www.fastcolabs.com/3006276/open-company/trying-woo-animators-disney-accidentally-invents-paperman-method)**

Modeling

Modelação é responsável por esculpir os modelos 3D em corpos de personagens, objetos, cenários, que serão utilizados ao longo do Projeto. Estes podem ser construídos a partir de referências, com um scanner 3D, através de escultura digital ou através da combinação de simples figuras geométricas, tais como esferas, cubos, cilindros, superfícies planas. Mas antes de um modelador começar a modelar, ele tem que primeiro escolher o tipo de geometria ou seja a constituição dos modelos, superfícies de subdivisão, NURBS (non-uniform rational B-splines) ou polígonos.

Existe muita informação para cada um dos tipos de geometria, mas nesta dissertação vamo-nos só focar nos polígonos, porque estes são normalmente os mais utilizados em jogos e por modeladores 3D. Portanto a superfície num objeto 3D é composto por polígonos interligados, designado por *Polygon mesh*. Cada polígono é composto por 3 ou mais vértices, as arestas, e a face. A forma mais básica num polígono é Tri (3 arestas, triângulo), a mais preferida e usada é a Quad (4 arestas, quadrangulares). Modeladores evitam usar qualquer forma com mais do que 4 arestas. Porque abre a possibilidade a polígonos com reentrâncias, cavidades o que pode causar problemas no projeto, mais especificamente pode facilmente complicar ou inviabilizar algoritmos de cálculo de intersecção, como tal, não têm suporte direto na base computacional.

Topologia é a composição, organização dos polígonos no *Polygon mesh*. Se este tiver uma boa topologia pode ser deformada de várias maneiras sem se rasgar ou ficar cortado. Uma das razões porque os polígonos Quad são os mais usados deve-se à composição do seu agrupamento ser ordenado, simples, como uma grelha, A introdução num polígono Tri numa malha de Quad causaria uma deformação, inchaço no modelo. *Polycount* é o número de polígonos na superfície num modelo, quanto mais polígonos tiver maior o seu detalhe, mas maior poder de processamento é necessário para o escultar e maior tempo necessário para o renderizar. Aliás um jogo pode ter uma personagem com um modelo com maior número de polígonos (grande detalhe) quando é precisa em vídeos cinemáticos, e um modelo com menor número de polígonos ideal para *gameplay*, onde a sua manipulação depende do jogador e o computador precisa renderizar em tempo real. O mesmo aplica-se para modelos que se encontram longe ou escondidos da câmera, logo não necessitam muito detalhe, polígonos.

Na criação e design dos modelos, principalmente quando se pretende atingir realismo, deve-se ter em atenção os aspectos de Uncanny valley, fenômeno estudado originalmente no campo da robótica. Segundo esta teoria, quando um robô, modelo fica parecido e quase atua como um ser humano mas não convence o observador na sua integra, é percepcionado como desagradável e repulsivo. O que é extremamente problemático em animação, principalmente porque vai contra o princípio do appeal, um dos 12 principles of animation referidos anteriormente.
Estas informações são importantes para perceber a constituição dos modelos, corpos das personagens no jogo e como esta pode influenciar o seu comportamento com os outros elementos do jogo, por exemplo com a iluminação do jogo.

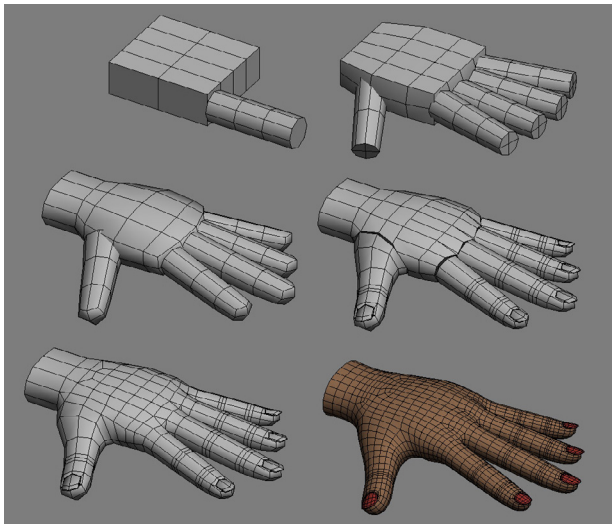

Fig. 18- Exemplos de *Polygon mesh* constituido por *Quad.* Recuperado em http://www.3dtotal.com/tutorial/1868-3ds-max-character-creation-chapter-1-photoshop-v-ray-by-andrew-hickinbottom-female-pin-up-olivia?page=3#.VFEJa_nF_84

Texturing

Normalmente os modelos são automaticamente criados com uma cor básica uniforme e neutra (normalmente cinzento), portanto os artistas de textura têm a responsabilidade de transformar a superfície do modelo igual às superfícies representadas nos conceitos de arte ou igual à superfície do seu correspondente real. As texturas ou são criadas a partir da combinação de fotografias, por exemplo padrão de parede de tijolos (através do Adobe Photosop) ou são pintadas à mão, e mais tarde aplicadas nos modelos. Embora com os avanços de software, também é possível pintar as texturas à mão diretamente nos modelos (por exemplo através de Autodesk's Mudbox, Maxon's BodyPaint 3D, ou Pixologic's Zbrush).

Para um artista de textura conseguir aplicar uma textura 2D ao modelo 3D necessita de criar um UV *map*, estes são responsáveis por projetar, embrulhar a textura 2D no modelo 3D. Basicamente atribui-se coordenadas UV / marcadores dos polígonos no modelo e a seguir desembrulha-se o modelo numa imagem 2 dimensões, que servira como um mapa das coordenadas dos polígonos, o UV *map*. Portanto, quando um artista pinta a textura no UV *map*, automaticamente estamos a associar as coordenadas da textura com as coordenadas UV dos polígonos, tornando possível a aplicação da textura do modelo. Este processo é designado por UV *mapping* . Ainda não existe software que consiga criar automaticamente um excelente UV *map* com apenas um clique num botão, normalmente é necessário modificar manualmente o UV *map* gerado pelo computador.

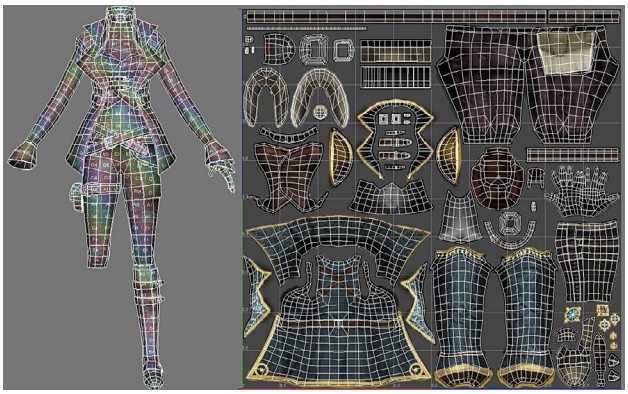

Fig. 19- Exemplo de *UV map*. Recuperado em http://lart3d.com/3dgallery/3d12mg.html

Shaders (também conhecidos por material e *surface*) são instruções implementadas pelo artista de textura na superfície do modelo, em como se deve comportar com o ambiente (o que o rodeia). Alguns atributos básicos de *shaders* são:

- Cor- atribuir uma cor ou textura ao modelo.

- *Ambience* - como é que a quantidade de luz de ambiente afeta a superfície do objeto.

- Transparência- a quantidade de luz capaz de atravessar pelo objeto, tornando-o transparente, normalmente utilizado para recriar vidro.

- Refletivo - o quanto refletivo o objeto é, ou seja a quantidade de luz que a superfície do objeto consegue refletir. Translucidez - Quantidade de luz atravessada pelo objeto, ideal para recriar papel.

- Refração - Este atributo indica se a direção da luz muda quando atravessada pelo objeto, incluindo o ângulo de desvio, por exemplo o efeito de quando se olha através dum copo com água e os objetos estão distorcidos.

- Brilho - simula o efeito de auto iluminação, por exemplo a luz emitida pelo ecrã num monitor de computador. Incandescência - determina a qualidade de auto iluminação.

- Realce especular (*Speclar highlight*) - cria manchas brilhantes ao longo da superfície do objeto, que na vida real corresponde à reflexão da fonte de luz.

- Colisão (*Bump*) - simula textura ao longo da superfície através de *bumps maps*. São imagens monocromáticas que representam mudanças ao longo da superfície, a cor preto representa declives, descensões na superfície e a cor branco representa elevações, ascensões na superfície. Mas o *bump map* não altera a geometria do modelo, altera as normais da superfície do modelo e consequentemente como esta reage à luz, criando a ilusão de elevações e declives ao longo da sua superfície. Por exemplo consegue simular uma superfície rugosa.

Existem vários tipos de *shaders*, por exemplo:

- *Flat shaders* - um dos *shaders* mais básicos, cujo resultado é sempre uma cor uniforme, sem gradientes (exemplo de uso: imagem de fundo do céu).

- Lambert - com o mais básico de iluminação sem efeitos de refração e Realce especular, indicado para materiais que não brilham muito, papel, madeira inacabada.

- Blinn - permite reflexão e Realce especular, próprio para plásticos, metais, cabedal.

- Phong - cria um aspeto brilhante, com um Realce especular intenso, próprio para plástico.

Assim que o artista de textura configurar os atributos do *shader* num modelo, precisa introduzir a informação visual nos atributos através da atribuição mapas, tais como:

- *Bump maps*. Mapas de cor- com a informação da distribuição das cores ao longo da superfície no modelo.

- *Specular maps* - Afeta o comportamento de Realce especular (*Speclar highlight*) ao longo da superfície, criando a ilusão de arranhões, perturbações.

- Mapas de transparência - Cria um gradiente de transparência ao longo da superfície, bons para recriar vidros sujos, gelado ou colorido.

- Mapas de reflexão- Controla o degrau de reflexão ao longo da superfície.

- *Displacement maps* - Depois de ser aplicado no modelo e renderizado, altera a geometria da superfície do modelo de modo a criar uma nova forma.

- *Ambient occlusion map*- São imagens monocromáticas que criam sombras suaves, desfocadas ao longo da superfície do modelo

- *Normal maps* - Tal como *bump maps*, cria a ilusão de textura, mas em vez de se basear numa imagem de tons de cinza, utiliza uma imagem com as cores verde, azul e vermelho (RGB). Tipicamente utilizado em jogos.

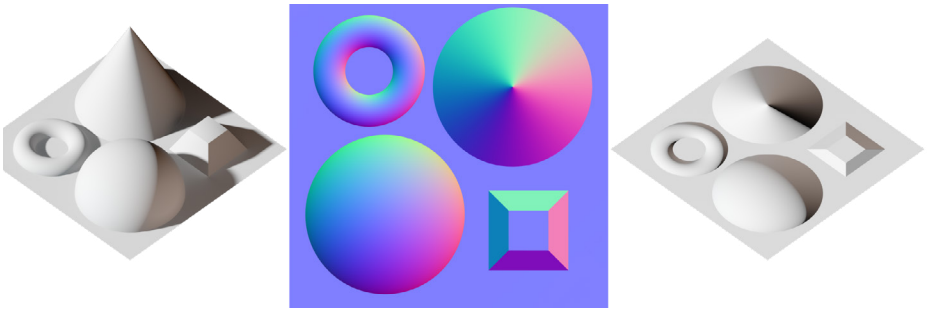

Fig. 20- Exemplo de *Normal Map*. Recuperado em https://en.wikipedia.org/wiki/Normal_mapping

Esta informação, ajudou-me a compreender os vários tipos de *maps* e *shaders* e como estas ajudam a criar a ilusão de volume em imagens 2D.

Rigging/setup

Rigging é responsável pela criação e implementação de um *esqueleto* (também conhecido por rig), num modelo, permitindo a sua manipulação, controlo e animação, do modo mais rápido e eficiente possível. O artista de *rigging* implementa um esqueleto, composto por articulações e ossos. A seguir cria controlos que permite aos animadores animar o modelo, através da translação e rotações das articulações. Depois usam *deformers* para conectar o esqueleto com a geometria do modelo, de modo a este acompanhar com os movimentos dos *rigs* e finalmente para garantir o realismo do movimento do modelo implementa-se mais *deformers*. Para facilitar o processo de rigging, convém certificar que o modelo têm uma boa topologia, garantindo que com a sua manipulação, o seu corpo deforma-se corretamente sem se partir.

A implementação desta área no projeto é bastante complexa e normalmente o artista de *rigging* necessita de ser acompanhado com a equipa de R&D.
Deformers são responsáveis pela ligação entre o modelo e o *rig*. Por exemplo:

- *Skining/Binding é* um *deformer* que atribui valores à cada vértice do modelo que especifica o quanto são afetados pela articulação do joint.

- *Constraints* permitem criar um sistema em que certos objetos partilham o mesmo tipo de movimento. Os *constrains* mais utilizados são:

- *Point*- A translação do objeto x é partilhado com o objeto y.

- Aim- A rotação do objeto x segundo um alvo é partilhado com objeto y, ideal para um sistema de controlo de olhos .

- *Orient*- A rotação do objeto x é partilhado com objeto y.

- *Scale*- A mudança de escala do objeto x é partilhado com o objeto y.

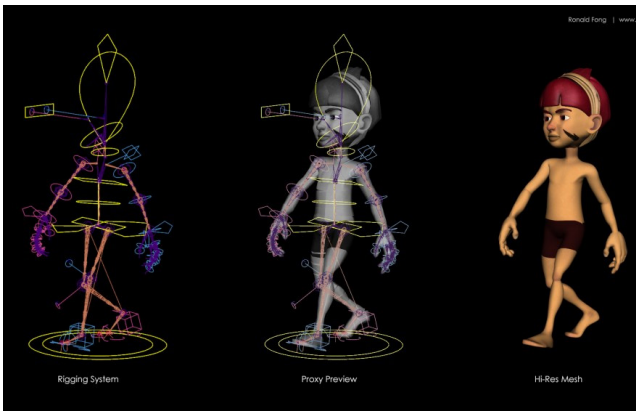

Fig. 21- Exemplo de *rig*. Recuperado em http://www.ronald-fong.com/blog/portfolio-item/rigging-for-animation-demo-reel/

Todos os esqueletos são sistemas com hierarquia, de modo a definir como os movimentos de partes do modelo influenciam outras partes. O mais básico é hierarquia Parent/child, em que categoriza os ossos do *esqueleto* por parent ou child, ou seja, um osso *child*, pode mover, rodar sem afetar o osso *parent*, mas se o osso *parent* mover então o osso *child* acompanha com o seu movimento. Esta relação chama-se *parenting*. É possível haver vários *child* com o mesmo *parent*, chamados *siblings*. Embora existam aplicações que conseguem customizar a relação *parenting*.

Kinematics permite estabelecer a ordem na hierarquia do *rig*, basicamente fornece aos animadores duas diferentes abordagens de criar uma animação. A Forward Kinematics (FK) estabelece a ordem de hierarquia predefinida do *rig* e Inverse Kinematics (IK) inverte a ordem de hierarquia predefinida do *rig*, portanto se a ultima articulação na hierarquia mover-se, o resto dos joints, bones acompanham com o movimento. Normalmente o IK é excelente para animar pessoas a andar, por exemplo, um pé ao tomar um passo permanece na nova localização e consequentemente a anca acompanha. Inverse Kinematics é mais rápido que Forward Kinematics, mas a aplicação de ambos é útil para alcançar um bom movimento natural.

Normalmente os *rigs* são baseados à partir do esqueleto humano, animal, como tal é conveniente que os artistas de textura tenham no minimo um conhecimento básico da anatomia do corpo que pretende simular. Contudo, deve-se ter em conta que a anatomia de todos os corpos têm limites, uma pessoa não consegue rodar a sua cara 180º. Como tal é necessário implementar limites na rotação dos joints. Alias a adição destes limites é fundamental para o uso de Inverse kinematics.

Animação

> *"poor animation can kill any 3d project. Even if the models are perfect and the lighting is great, unrealistic or distracting motion will pull your audience out of watching your project"*
> **Beane A. (2012). 3D animation essentials**

Muitas das técnicas, ferramentas de animação tradicional foram adaptadas para animação 3D. Mas ao longo do tempo a área de animação 3D tem evoluído, amadurecido, criando a sua própria linguagem e ferramentas. Como tal um animador tradicional sempre conseguiu transferir-se para a indústria de animação 3D, mas ao longo do tempo a curva de aprendizagem para animação 3D tem aumentando.

Os 12 princípios básicos de animação também foram influenciados pela indústria 3D, digital: O principio de Solid Drawing requer que um animador deva ter um boa compreensão do seu software de animação 3D, mas não significa que não precisa de saber desenhar, afinal animação 3D continua a ser uma comunicação visual, e a maneira mais rápida de demostrar o produto final é a partir de desenhos. Tendo em conta ao principio de Straight Ahead Action and Pose to Pose, especificamente Pose to Pose. Em vez de ilustrar todos os *fotogramas* na animação entre *key frames*, a animação pode ser calculada e produzida por computadores.

Motion-Capture Animation é um novo processo, sistema, de capturar a posição e os movimentos num ator. O sistema processa e transforma os dados do movimento em dados de animação 3D, permitindo aos *rigs* dos modelos replicar os movimentos do ator. Tipicamente para capturar os dados de movimento, os atores usam fatos colados à pele com marcadores ao longo do corpo, o computador consegue capturar a posição dos marcadores à partir de varias câmeras durante a performance e consequentemente deduzir, calcular os dados de movimento.

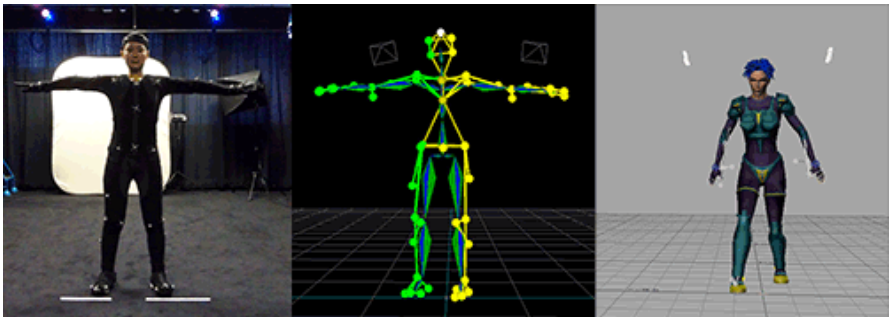

Fig. 22- Exemplo de Motion Capture Animation. Recuperado em https://www.behance.net/gallery/16734909/Motion-Capture

Efeitos visuais 3D (VFX)

Efeitos especiais 3D é a animação de tudo exceto as personagens e os *props* com que interagem, por exemplo cabelo, tecidos, pó, água e fogo. Normalmente estes efeitos especiais dependentes de software com motores de física que simulam ar, gravidade e vento, porque quanto mais natural e complexo for o seu aspeto e movimento, mais difícil é de os animar à mão. Portanto é necessário que o artista de efeitos especiais 3D tenha um conhecimento básico de físicas e matemática e as consiga combinar com o seu senso artístico. Mas tem que ter sempre em conta que o seu objetivo final é com que os efeitos melhoram o cenário, em vez de roubarem involuntariamente a atenção do público.

A simulação dos efeitos 3D produz muitos dados que precisam ser calculados e processados. Portanto, não é fácil manipulá-los e exigem uma alta carga de trabalho ao processador. Os efeitos especiais 3D podem dividir-se nas seguintes especializações:

- *Particles* - Um emissor, posicionado no espaço pelo artista, liberta um fluxo de pontos, partículas, cujo movimento respeita as leis de física, gravidade, vento, fricção. Ideal para simular, poeira, fogo, chuva, neve ou enxames.

- *Hair and Fur* - São sistemas que criam cabelo e pelos sujeitos a perturbações, que se comportam de modo fluído e realista. Este sistema também pode ser utilizado para criar antenas, caudas, tentáculos, e é um dos sistemas mais difíceis de controlar e cuja renderização requer muito tempo, sendo difícil obter resultados realistas.

- *Fluids* - São simulações especiais de partículas que conseguem simular o movimento de fluidos através de uma equação. Não se restringe à simulação de líquidos, como também ao fumo, fogo e plasma.

- *Rigid Bodies* - São modelos que representam sólidos, que ao colidir com outros objetos não deformam, cujo movimento é condicionado pelas leis da física. Normalmente utilizado para colisão básica de objeto rígidos, fracionamento de objetos, por exemplo a simulação de cacos de vidro a cair.

- *Soft Bodies* - São modelos que representam sólidos, que devido à colisão com outros objetos, deformam-se. Estes tipos de simulações podem exercer muita carga de trabalho ao computar. São normalmente utilizadas para criar tecido realístico, músculo, cartoon *hair*.

Iluminação

Iluminação é a área que simula os vários tipos de luz do mundo real num cenário 3D, por exemplo *spotlights*, luz de lâmpadas, luz do sol... Portanto permite manipular o posicionamento, direção e atributos das luzes (Intensidade, color, atenuação), podendo também pode incluir o das propriedades das sombras do objetos de modo a condizer com a intensidade das luzes de modo alcançar o cenário, ambiente predeterminado. Alguns exemplos de Iluminação são:

- *Spotlight* - Emite uma luz numa direção e localização específica.

- *Omni/Points Lights* - Emite luz numa localização em todas as direções

- *Infinite/Directional lights* - Emite raios de luz que são paralelos entre si, produzindo sombras paralelas entre si criando a sensação que a fonte luz está muito longe. Ideal para simular a luz do Sol ou da Lua.

- *Ambient Light* - Estas luzes são usadas para simular iluminação global

- *Area light* - Uma das luzes mais complexas, em vez de emitir a luz num ponto de localização, emite numa área, por exemplo a área numa janela ou ecrã num monitor.

Rendering

O passo final da fase de produção, que combina os modelos, *rigs*, animações, *shaders*, texturas efeitos especiais 3D e iluminação e os transforma, renderiza-os em imagens. Dois métodos básicos de renderização são:

- Scaline - Renderização mais rápida, consequentemente esta não calcula reflexões, refrações ou sistemas complexos de iluminação global. Sendo ideal para renderização de cenários *cell-shaded*, parecido com cartoons, mas principalmente para as pré-visualizações do cenário ao longo do projeto para observar o seu progresso.

- Raytracing - É o método de renderização mais completo; renderiza reflexões, refrações e sistemas complexos de iluminação global, o que consequentemente, exerce mais carga ao computador.

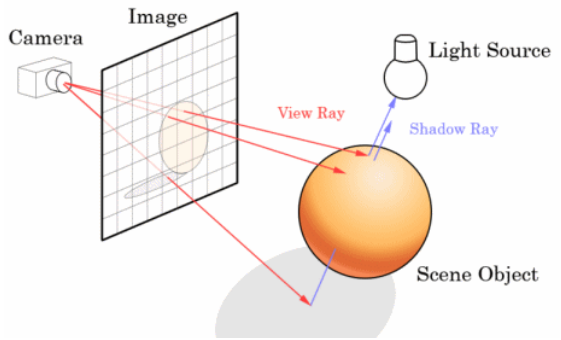

Fig. 23- Exemplo de Raytracing. Recuperado em https://blog.codinghorror.com/real-time-raytracing/

Naturalmente nos jogos, o cenário tem que reagir às ações dos jogadores e estas bem com as suas consequências, necessitam ser renderizadas em tempo real. Um método para obter renderização, em tempo real, é *baking texture maps*, que envolve a incorporação da informação de sombras e luzes no mapa de textura, pois a renderização da reflexão de luz não é fácil.

Apesar de *Raytracing* renderizar imagens muito realistas, este é extremamente difícil aplicar em renderização em tempo real, pois é um processo complexo que exerce muita carga de trabalho e como tal necessita de muito tempo para renderizar.
Os *engines* de jogos que nos permitem jogar jogos de vídeo tais como Unreal e unity engine, também são bem capazes de renderizar em tempo real. Mas tipicamente os designers de jogos preferem implementar os programas de renderização DirectX e OpenGL, porque tem algoritmos rápidos de *Scaline* e são suportados por placas gráficas .

2.4. Estado da Indústria de Animação

Com o propósito de aprofundar o meu conhecimento de animação e determinar se a base do meu jogo deve ser 3D ou 2D, realizei uma pesquisa do vários *input* de pessoas com mais experiencia da área de animação e uma investigação dos motivos pelo crescimento de produtos de animação 3D, consequentemente determinando as vantagens e desvantagens entre 2D e 3D.

Desde do inicio do milênio o custo e tempo de produção de animação 3D têm diminuindo enquanto a sua qualidade têm aumentado, tal como têm aumentado a sua complexidade e portando sendo necessário um estudo profundo antes de o praticar. Entretanto animação 2D é um processo simples e pessoal que depende especialmente em experiencia da sua prática. No entretanto recentemente os grandes estúdios de animação têm optado por animação 3D na produção dos seus filmes. Apesar da recepção positiva do público e à consciencialização de que a animação 3D é mais difícil e complexa, o processo de produção de animação desenhada à mão é mais estrita, sendo pouco flexível à mudanças. Este aspecto foi observado na construção do protótipo do jogo, como é explicado em detalhe na página 55.

> *"I've worked on CG features and I've worked on hand-drawn features. And hand-drawn features are harder to make. Hand-drawn cartoons take a year to produce. Once you've produced sequences, it's hard to change the work. You have to go back and do everything over."*
>
> *"But with CG, you can animate the movie in three or four months, change things close to the release date. You can't do that in hand-drawn animation. If you find out the story doesn't work when you're two-thirds done, you're stuck. With CG, we change the story and rework sequences until late in the process."*
>
> **http://www.rotoscopers.com/2015/02/27/former-disney-veteran-explains-why-big-studios-have-abandoned-2d-animation/**
>
> Steve Huelett

Apesar de tudo, os estúdios têm uma maior tendência a continuar produzir animações 3D porque se estes fizeram o grande investimento financeiro para obter o equipamento necessário para produzir animações 3D, sentem-se compelidos a continuar a utilizar o equipamento. Afinal a indústria é primeiro um negócio antes der ser uma arte.

Com a publicação constante de filmes de animação 3D, membros do público, incluindo eu, começaram a formar a opinião que animação 2D estava a ser esquecida, estava a morrer. Esta é uma opinião precipitada, afinal ainda existe a produção de novas animações 2D na televisão, internet e cinemas. Simplesmente a animação 2D deixou de ser a escolha dominante, desde que deixou de ser a opção mais viável em termos de criatividade. O que não é necessariamente um mau desenvolvimento para o futuro do 2D, Dado que os seus limites são constantemente expandidos com o suporte das novas tecnologias digitais, e, está livre para evoluir e redefinir-se sem ser influenciado pelos grandes estúdios, por exemplo Disney.

> *"So I've got a lot of love for computer animation, but I think it's also important to have an alternative. It would be a pity for the tradition of hand-drawn animation to die out. Yet with technology, we're offered the opportunity to make hand-drawn animation in a way that we weren't even fifteen years ago. With today's computers, we can make hand-drawn animation on a feature scale with much smaller teams and lower budgets, and still make it more personal than high-level CG, which still requires a lot of money, a lot of technology and a lot of people. "*
>
> *"But I think today 2D animation has a responsibility, much like painters had after photography was invented, to reinvent what it is. It can't go after realism, because there is no point; it has to do something only 2D can do. ... In the whole history of visual arts, there is still so much that we can explore."*
> **http://www.cartoonbrew.com/award-season-focus/tomm-moore-on-song-of-the-sea-reinventing-2d-and-dodging-the-studio-system-107389.html**

Com o uso hábil da tecnologia 3D, esta oferece soluções que facilitam o processo de animação. Tais como:

- Consegue produzir mais facilmente produtos de grande detalhe ou complexidade com maior perfeição.

- Têm mais facilidade em criar animação com subtileza, por exemplo animar uma personagem que mal se move e que pareça estar viva.

- Obter um estilo visual mais realista, em que se obscura a linha entre fantasia e realidade, a ideia de fantasia tornar uma realidade pode ser apelativa.

Segundo opinião dos entusiastas da animação 3D, os maiores desafios desta área são:

- Fazer o *rigging* das caras das personagens.

- Construção de uma boa topologia e a aplicação de *deformers* de um modelo , de modo a garantir uma animação fluida sem erros do modelo.

- Utilizar o mesmo ficheiro em diferentes softwares, programas, sem que este sofre erros ou perda de dados.

2.5. Técnicas de Animação em Jogos

Tendo em conta o âmbito desta dissertação e o estudo, compreensão dos conceitos básicos da animação 3D e 2D, fiz a pesquisa e análise de vários jogos com animações, que evidenciam técnicas que influenciam a sua animação, tais como, a ilusão de volume em imagens 2D. A partir de entrevistas, palestras dos seus criadores consegui recolher uma lista de técnicas de animação dos seus pontos fortes e fracos.

Animação modular/ bone animation

Esta técnica envolve a animação de *sprites* como se fossem marionetas, ou seja as personagens são compostas por varios bocados de imagens, cujo movimento depende de translações, rotações e transformações dos bocados. Possibilita igualmente, inúmeras e diversas animações com um número pequeno, limitado de fragmentos de imagens, o que exige menor memória do disco rígido. Normalmente esta animação é utilizada em jogos com uma câmera fixa, ou seja um jogo "Side Scroller". Se o jogo tiver uma câmera com mais liberdade, o jogador terá acesso a mais ângulos de perspetiva das personagens e por cada novo ângulo é necessário um novo perfil da personagem, o que envolve a produção de mais fragmentos de imagens para o novo perfil.

Tipicamente em animações de *sprites* com pequeno orçamento, os animadores têm que ilustrar rapidamente uma quantidade imensa de fotogramas, e para tal ser possível, convém que os desenhos sejam simples ou de menor qualidade. Em contrapartida a animação modular permite animar ilustrações ricas em pormenor de alta qualidade.

Contudo animação modular continua a utilizar imagens 2D, e como tal, o movimento da personagem está restrito a duas dimensões, mas é possivel criar a ilusão de movimento de 3 dimensões com a animação tradicional de sprites dos pedaços da marioneta. Por exemplo, o estúdio Vanillaware (2002) abusa desta combinação de animações, como se pode verificar no seu jogo Dragon Crown (2013) em que na animação de andar do Wizard, nota-se a troca dos sprites do ombro e da extremidade do sobretudo.

Fig. 24- Sprites da animação de caminhar do Wizard de Dragon Crown (2013). Recuperado em http://www.gamespot.com/forums/system-wars-314159282/dragons-crown-gamespot-reviewsome-offensive-and-un-29430571/?page=5

Fig. 25- Exemplos de Muramasa (2009), incluindo um dos seus *texture map.* Recuperado em http://www.gamasutra.com/view/feature/132486/king_of_2d_vanillawares_george_.php?print=1 e http://danfessler.com/blog/thoughts-on-modular-animation

Fig. 26- Exemplo de Dragon's Crown (2013) e Rayman Legends (2013). Recuperado em http://psnow.es/reportajes/guia-de-dragons-crown/ e http://www.polygon.com/2012/11/15/3651300/rayman-legends-to-have-playable-demo-on-wii-u-launch-day-new#5

Animação desenhada à mão

Envolve o uso de animação tradicional em jogos, ou seja a animação do jogo é composto por sprites que são desenhados à mão. Esta técnica permite uma animação de movimentos mais fluidos e suaves, principalmente tendo em consideração que é mais fácil implementar os 12 princípios de animação. Normalmente é utilizado por animadores, equipas com experiência e paixão por animação 2D. Todavia, apesar de ser um processo simples, requer muito trabalho e tempo para desenhar todas os sprites.

> *"...each character has essentially 8 animations each, filmed from two directions, northeast and southeast. With each animation coming in between 30-60 frames, that means each character class can have up to 1000 unique frames of animation"*
>
> *"Finally, making a finished animation in 2D can be time-intensive so we always start prototyping with rough animations. It's much quicker to iterate on programming and game design elements this way "*
> **E3_2011 (2011, Junho 11) Supergiant Games' Bastion. recuperado em http://www.cgsociety.org/index.php/CGSFeatures/CGSFeatureSpecial/supergiant_games_bastion**

Fig. 27- Exemplos de Jotun (2015), Banner Saga (2014), Cuphead e Castle Crashers (2008). Recuperado em http://jotungame.com/ e http://www.gamespot.com/reviews/the-banner-saga-review/1900-6415626/ e http://cupheadgame.com/ e https://en.wikipedia.org/wiki/Castle_Crashers

MultiPlane Camera effect

Método inventado pela animadora alemã Lotte Reininger nas suas animações, tais como The Adventures of Prince Achmed (1926). Uma década mais tarde foi explorada por Bill Garity, técnico dos estúdios Walt Disney, no desafio em incorporar volume, profundidade nos seus filmes de animação 2D.

> *"Imagine you're traveling by car on a long road trip. The road stretches out ahead of you into a wide-open landscape. In the distance you see a range of tall, hazy mountains. As you drive, you see trees and telephone poles whiz past you. At first, they appear tiny in the distance. Gradually, they get larger as you travel closer to them and pass them. Strangely, the mountains still appear to be very far away and hazy. They don't appear to have gotten larger despite being closer to them the way the trees and the telephone poles did. Why? Because the mountains are so large and far away, the change in their apparent size is negligible."*
>
> **Multiplane educator guide, recuperado em http://www.waltdisney.com/sites/default/files/MultiplaneGuideCurriculumPacket_Final.pdf**

O MultiPlane Camera normalmente cria este efeito dividindo o cenário em três diferentes planos, o plano frontal, plano médio, e plano de fundo. Alterando a posição deste níveis relativamente a câmera, é possível criar a sensação de profundidade. Por exemplo, aproximando o plano frontal e plano médio em direção a câmera, enquanto o plano de fundo permanece na mesma posição, cria-se a ilusão de que estamos a atravessar o cenário.

Fig. 28- Exemplos de Rayman Legends (2013), The Adventures of Prince Achmed (1926). Recuperado em https://www.youtube.com/watch?v=y-chi097uV4 e https://raymanpc.com/wiki/en/Mechanical_Dragon e https://www.theguardian.com/music/2015/jan/19/phillip-johnston-review-mofo-2015 e https://cizgilimasallar.blogspot.pt/2012/02/lotte-reiniger-adventures-of-prince.html

Efeitos especiais 2D

Uso de animação de *sprites*, num espaço 3D, para representar efeitos especiais, poeira, raios, fogo. Ajuda a alcançar o estilo 2D, "cartoon" e não exerce muita carga de trabalho no processador, ao contrário da alternativa efeitos especiais 3D. Mas em certos jogos cuja câmera tem muita liberdade, os *sprites* podem ser facilmente reconhecidos como bidimensionais, arruinando a experiencia de profundidade, portanto nessas situações convêm que os efeitos sejam pequenos (por exemplo: nuvens de poeira) ou rápidos (por exemplo: flash de luzes)

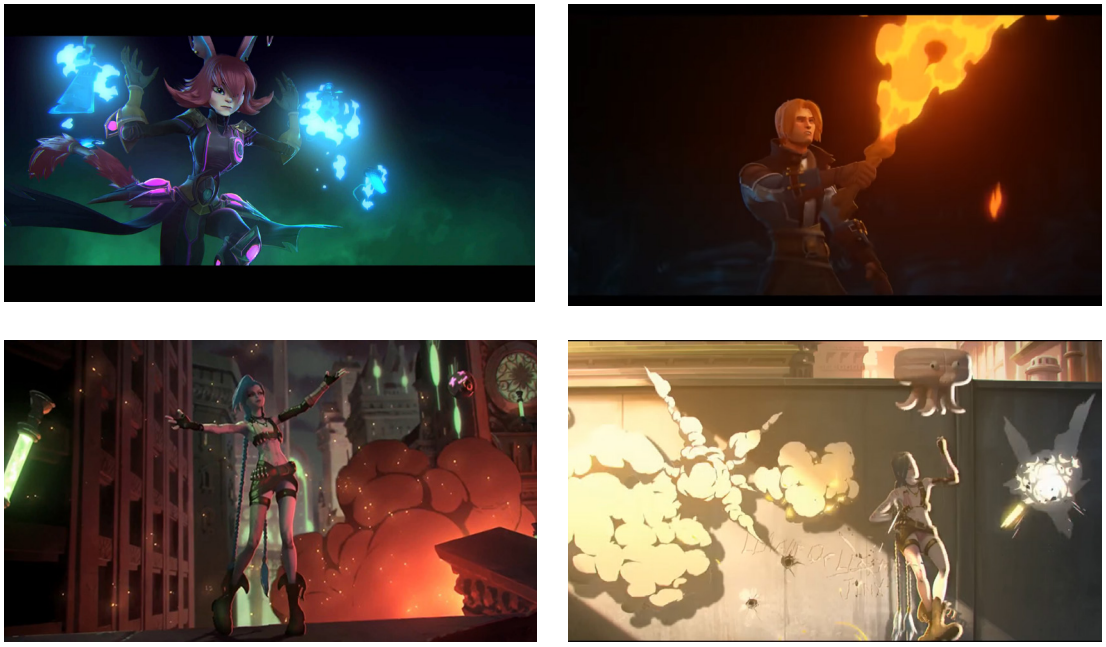

Fig. 30- Exemplos de Wildstar - Free-to-Play Trailer (2015) e League of Legends Music: Get Jinxed (2013) [Video]. Recuperado em https://www.youtube.com/watch?v=5dvIuXj4Gm4 e http://www.arkadian.vg/league-legends-music-get-jynxed/

Iluminação num jogo 2D (normal map)

Em vez de aplicar *normal map* em modelos 3D, aplica-se em sprites, imagens de cenário, de modo a estes serem afetados pela luz e sombras criando a ilusão de profundidade e volume.

Habitualmente, *Normal maps* são gerados pelo computador a partir de um modelo 3D e mais tarde são modificados manualmente. Mas no caso desta técnica não há modelos 3D, portanto os *Normal maps* têm que ser sempre criados manualmente.

Existem vários processos para obter *Normal map* a partir de imagens 2D, através de *plugins*, por exemplo NVidia Photoshop Plugin, ou criando 4 imagens, em que cada uma tem a personagem com sombras em que a luz está à sua esquerda, em cima, à sua direita e em baixo. Através do programa Sprite Lamp, ou seguindo o tutorial Normal Map Photography (2007) de Ryan Clark, as 4 imagens transformam-se no normal map.

A desvantagem deste método é que para criar estes processos manualmente com resultados realistas, é necessário muita experiencia e treino.

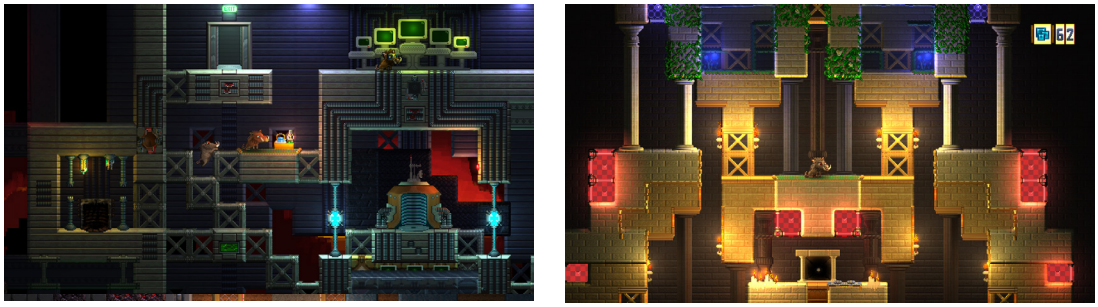

Fig. 29- Exemplo de Full Bore (2014). Recuperado em http://www.spawnfirst.com/reviews/dig-pig-full-bore-review/

Cel shading

É um tipo de renderização que aproxima os modelos 3D dos modelos 2D, exagerando o tratamento da superfície como na banda desenhada. A partir de cálculos matemáticos entre as normais dos modelos e a fonte de luz, os meios-tons, sombras e gradientes dos modelos são respetivamente substituídas por cores únicas . Mas sem controlo e alterações manuais nas normais dos modelos, este processo pode reproduzir resultados diferentes do desejado.

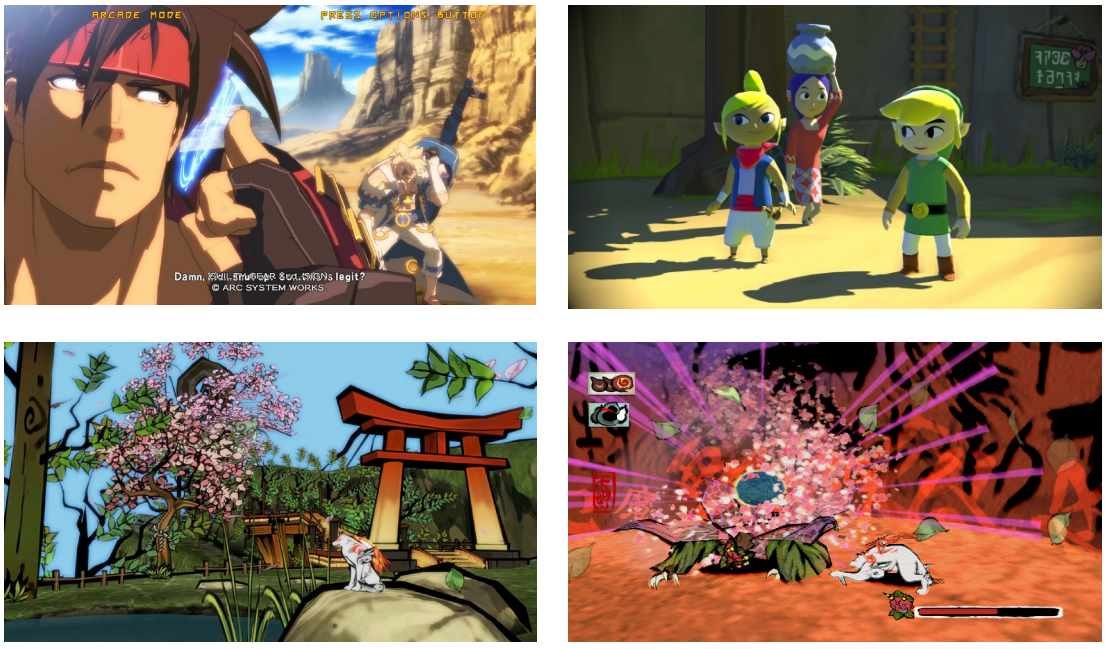

Fig. 31- Exemplo de GuiltyGearXrd (2014), Zelda Wind Walker (2012) e Okami (2006) . Recuperado em http://www.outofmeme.com/review/review-guilty-gear-xrd-sign/ e http://gematsu.com/2013/01/two-new-zelda-games-announced-for-wii-u e http://imgur.com/a/Z87Lu e http://www.meristation.com/playstation-3/okami-hd/analisis-juego/1785727

Animação de sprites a partir de modelos 3D

Esta técnica baseia-se em que as personagens, objetos e cenário são criadas e animadas em 3D, as animação são exportadas em series de imagens e mais tarde são agrupadas em sequência pelo *game engine* 2D para criar as animações. Animação de *sprites* a partir de modelos 3D tanto pode ser ideal para equipas pequenas e sem experiencia em 3D game *engines*, como também pode ajudar a capturar o estilo 2D na animação a partir de modelos 3D.

> *"The reason why they went this route was to keep the whole game running within a 2D engine and to avoid developing the game with a 3D engine which often proves to be a gigantic chore for game developers that don't have an incredibly huge staff."*
> **(2015, Abril 21) GuiltyGearXrd's Art Style : The X Factor Between 2D and 3D [video], GDC recuperado em https://youtu.be/yhGjCzxJV3E?t=1787**

Fig. 32- Exemplo de Bastion (2011), GuiltyGearXrd (2014). Recuperado em https://www.supergiantgames.com/games/bastion/ e https://www.destructoid.com/review-guilty-gear-xrd-sign--284491.phtml

Rotoscopia digital em modelos 3D

Paperman (2012) é um filme curto com um novo processo inovador. Depois da renderização da animação digital, com o uso de um software customizado, o artista desenha à mão sobre os modelos digitais nas *key frames*. O software anexa os desenhos aos modelos e produz a sua animação entre as *key frames*. Portanto, este processo permite implantar ilustrações nos modelos 3d que serão animados automaticamente segundo o movimento dos modelos. Apesar de ainda se encontrar em desenvolvimento, especificamente a animação automática dos desenhos entre *key frames*, já se demonstrou como um excelente processo com resultados híbridos de animação 3D com animação tradicional.

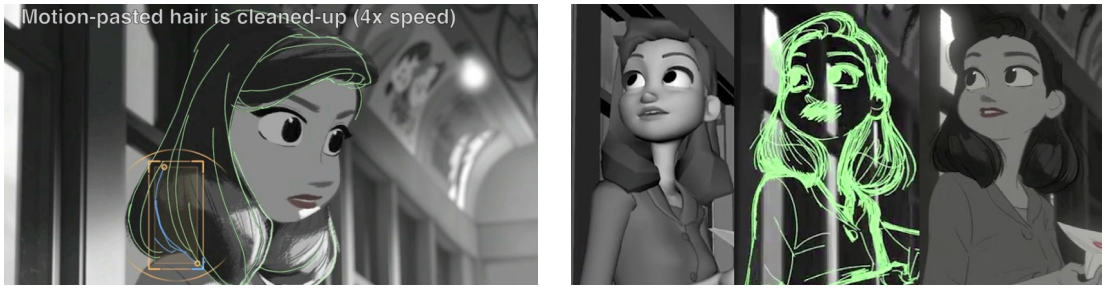

Fig. 33- Exemplo de Paperman (2012) [Video]. Recuperado em https://www.youtube.com/watch?v=TZJLtujW6FY

Outline e inline 3D

Reprodução de traços *outline* e *inline* típicos de ilustrações 2D em modelos 3D Os traços outline são os traços à volta da silhueta do modelo, através do método Inverted Hull Os traços inline são os traços na superfície do modelo, normalmente utilizados em 2D para indicar os contornos, bordas que se encontram dentro da silhueta do modelo.

Fig. 34- Exemplo de GuiltyGearXrd (2014) e Bordelands 2 (2012). Recuperado em http://mugenguild.com/forum/topics/generating-2d-guilty-gear-xrd-sprites-170997.0.html e http://borderlands.wikia.com/wiki/File:Krieg.jpg

Uso de cenário físico

Criação física num set, capturado, filmado por câmera é introduzida como espaço, cenário no jogo. Apesar de facilitar e simplificar a construção do espaço, cenário do jogo, principalmente para equipas sem experiência na criação digital do mundo e a suas interações, são necessárias extensas horas de trabalho manual e filmagem para capturar todos os ângulos necessários do set.

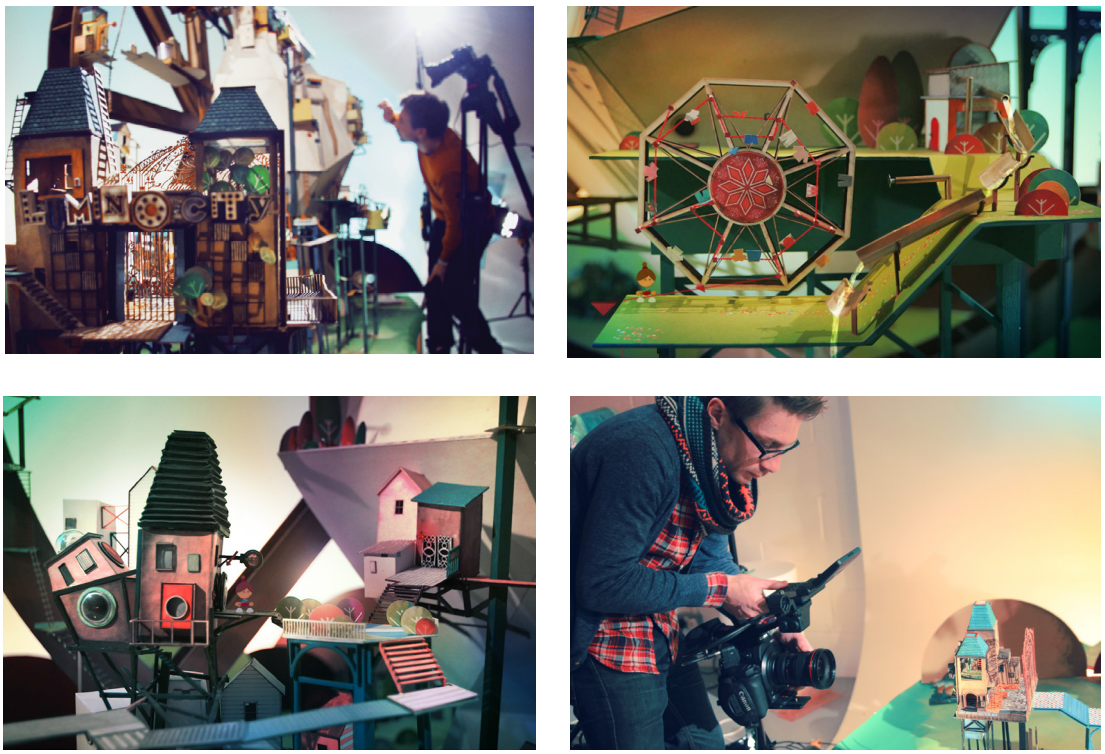

Fig. 35- Exemplo de Lumino City (2014). Recuperado em http://www.littlebeee.co.uk/home/2015/3/22/state-of-play e http://www.luminocitygame.com/ e http://www.theverge.com/2014/11/25/7275969/lumino-city-paper-video-game

3. Abordagem Metodológica

Neste capítulo estabelecemos os desafios a ser explorados para alcançar o objetivo desta dissertação, os processos técnicos a ensaiar para resolver os desafios, de modo a produzir a prova de conceito e por fim a sua calendarização.

3.1. Desafios Futuros

Relembro que o objectivo desta dissertação, é a pesquisa de técnicas que combinem maioritariamente processos 2D com 3D, sem perder a conjunção entre profundidade, volume e toque do artista na animação e o aspecto gráfico em jogos, e que pretendo implementar num protótipo do Jogo como prova de conceito, e consequentemente permitindo a avaliação do resultado da combinação de modo a determinar os ajustamento que necessita. No âmbito desta dissertação serei confrontado com vários desafios para executar o procedimento do 2D com 3D no jogo, tais como:

- Explorar como é que espontâneas fontes de iluminação 3D podem afetar imagens 2D sem estas se tornarem planas, com renderização em tempo real.

- Investigar com experimentação de métodos, formas de dar um toque artístico à textura dos modelos 3D.

- Descobrir os limites no jogo do uso de animações complexas.

- Explorar o uso de efeitos especiais 3D mas com estilo 2D, com a gratificação da diminuição de complexidade dos efeitos especiais 3D.

- Salvaguardar a eficiência do jogo, ou seja, manter um equilíbrio entre carga de trabalho do processador e o uso de memória do disco rígido, ao longo do processo de resolução dos desafios anteriormente mencionados.

- Definir as minhas limitações, permitindo determinar o potencial do uso num game *engine* 3D ou 2D, de modo a escolher o mais qualificado segundo a natureza desta dissertação.

3.2. Processo para Produzir a Prova de Conceito

Os processos adotados na realização da prova de conceito passaram por uma pesquisa sobre estado da indústria, conceitos básicos, técnicas e exemplos de animação 3D, 2D, e a sua combinação, incluindo uma investigação em como esta informação pode ser utilizada na criação de um protótipo de um jogo. Mais tarde, a informação foi analisada e escrita na dissertação.
Para criar o nível do jogo foi necessário uma familiarização com o software 3D, através de tutoriais colecionados ao longo da pesquisa, e através da experimentação. Este também envolveu a reprodução das técnicas de animação, de modo a determinar o seu potencial.

Para uma melhor gestão do processo de produção do nível do jogo foi fundamental determinar os limites da minha capacidade de produção.

O processo de testes de apreciação é responsável pela exposição do nível do jogo a *test users*, de forma a registar a sua experiencia, *bugs* e opinião sobre a estética do jogo, incluindo a animação, com o propósito de analisar os resultados dos testes, determinando o sucesso das técnicas implementadas e consequentemente fazer as alterações necessárias.

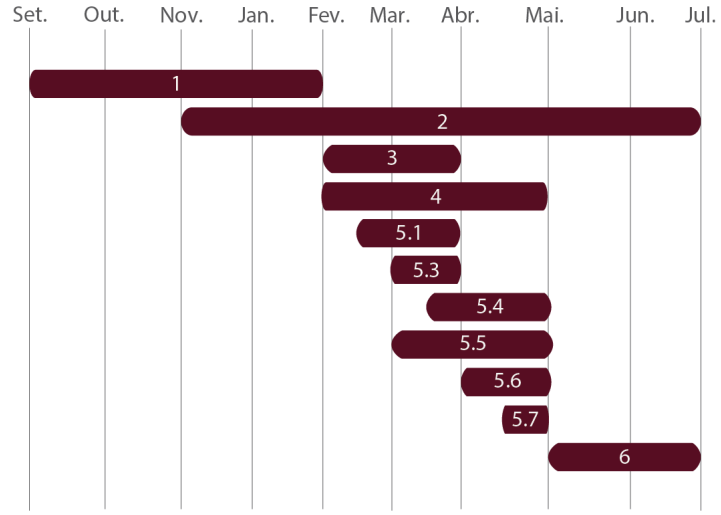

Fig. 36- Primeiro Calendário de trabalho

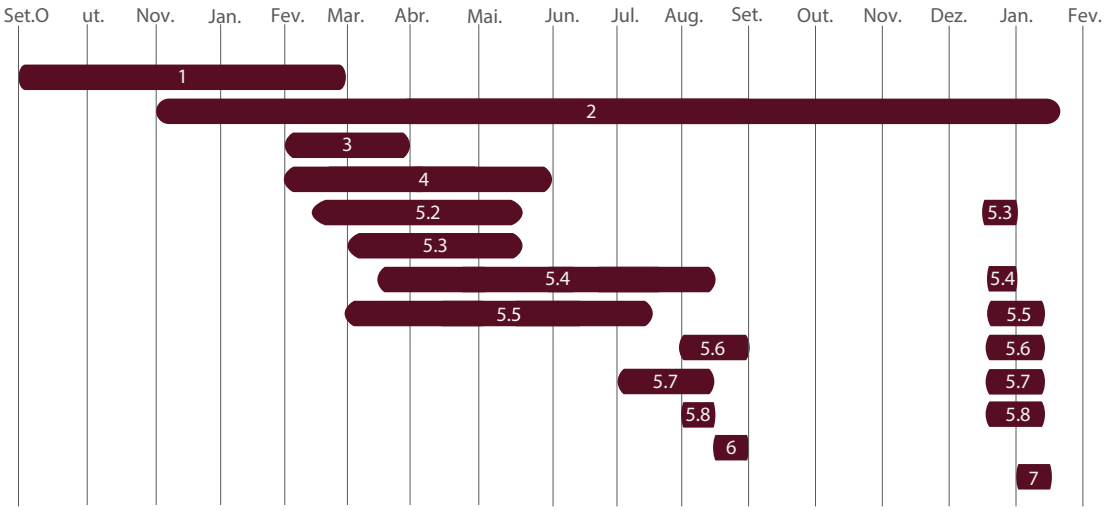

Fig. 37- Calendário de trabalho

1- Pesquisa para dissertação
2- Escrita da dissertação
3-Familiarização com software
4-Experimentação de Técnicas
5- Criação do jogo
5.1- Modelação
5.2- Desenho
5.3- Rigging
5.4- Animação
5.5- Programação
5.6- VFX
5.7- Iluminação
5.8- Áudio
6- 1ª Teste de apreciação
7- 2ª Teste de apreciação

4. Conceptualização e Proposta de Game Design

Este capítulo representa a primeira fase do planeamento do desenvolvimento do projeto prático desta dissertação, nomeadamente do desenvolvimento do nível do jogo, e, estabelece a premissa, mecânicas de *gameplay,* design do jogo, o modelo de animação e as técnicas de animação que se pretenderam implementar.

4.1. Conceito Proposto

Decidi criar um conceito simples de modo a que melhor ilustrasse os objetivos da tese, por forma a ter ação que possibilitasse a animação de diversas ações, e que cuja história permitisse integrar mecânicas que tornassem o jogo mais interessante.

O gênero deste jogo é de Fantasia, com localização e data indefinidas, inspiradas pelas terras célticas na idade do ferro. O ser humano tem acesso à magia da natureza e de deuses a partir do uso de runas, sendo uma prática comum de druidas. As runas podem estar inscritas em armas, armaduras ou livros, mas a sua inscrição necessita de tempo, materiais e técnicas especificas, tornando a sua preparação essencial para combates o que introduz uma componente interessante no *gameplay*. A personagem principal do jogo adquiriu acesso à magia sem ser a partir de runas, *Witchcraft*. Esta habilidade é anormal, inatural, o seu abuso pode corromper a natureza (por exemplo: dias mais escuros) e atrair atenção e castigos dos deuses (por exemplo, dano ou maldições à personagem). A personagem principal, "Witch", no nível do jogo ao explorar uma floresta, vai ter necessidade de travar um combate com bandidos ou cultistas no descampado na proximidade de um castro.

4.2. Proposta de Game Design

A intenção é a de implementar no jogo a alteração do mundo durante o uso de *Witchcraft* em combate. O mundo escurece drasticamente e os seres vivos tornam-se em sombras cobertos por símbolos, que representam o seus espíritos e pistas sobre as suas habilidades.

Estas animações serão criadas tendo em consideração que câmera do jogo vai ter uma vista da terceira pessoa a partir do céu.

Idealmente o sistema de combate, permitia total controlo das personagem, *hack and slash* game, por exemplo: Diablo 3, mas tendo em conta ao minha experiencia e tempo limite para a concretização do protótipo do jogo, o sistema do jogo vai ser dividido por turnos, *turn based* game, por exemplo: Final Fantasy Tactics, Disgaea.

Fig. 38- Exemplo de Disgaea 3 e Diablo 3. Recuperado em http://projetojogatina.org/jogatina/max-time-waster-d2/ e http://www.gamingcfg.com/screenshot/Diablo-3-mobs-1747

4.3. Técnicas a Ensaiar

Para a seleção de animações tive em consideração as animações básicas, necessárias para um jogo com mecânicas de combate e animações complexas, que não só o tornam mais interessante, mas também lhe conferem a capacidade de demonstrar o resultado de 2D com 3D, como por exemplo animações com efeitos especiais.

As animações básicas são:

- Movimento, o que envolve no mínimo animar a personagem em 3 perspetivas, andar para norte, para sul e andar para este ou oeste (estas perspetivas partilham a mesma animação).
- Ataque simples (por exemplo: soco ou ataque com espada).
- Sofrer ferida.
- Cair inconsciente/morrer.
- Estar em pé, pois nenhum ser vivo consegue estar completamente parado, por isso idealmente quando este não está a praticar nenhuma ação deve ter ligeiros movimentos. Esta animação aumenta a ilusão de vida da personagem.

As animações complexas são:

- Ataques com elementos (por exemplo, espada com eletricidade).
- *Dash attack*, ataques físicos com alta mobilidade.
- Magia de evocar criaturas, projéteis (por exemplo: bolas de fogo, picos gelo) ou um laser. Pois cada um destes tipos de magia requer diferentes segmentos de código.

É um jogo 3D que não é fotorrealista, além de não possuir os recursos necessários para produzir um jogo fotorrealista, o estilo cartoon de não fotorrealista combina melhor com os efeitos 2D, oferecendo um contraste de estética interessante.

As técnicas que se pretendem implementar no jogo são *multiplane camera*, Iluminação num jogo 2D com efeitos especiais 2D, em caso de necessidade devido a problemas com animação e compensações pode-se recorrer à animação modular, cell shading e animação de *sprites* a partir de modelos 3D. O Game engine (motor de jogo) para correr o jogo é Unity, sendo o engine em que possuo mais experiência. Mas os programas que tenciono utilizar para as várias fases da construção do jogo são:

Funções	**Programas**
Modelação	Z-brush, Autodesk 3ds Max e Mudbox
Composição de topologia	Autodesk 3ds Max e Mudbox
Baking e texturas	Photoshop, Autodesk 3d Max e Mudbox
Rigging	Autodesk 3d Max ou Cinema 4d
Animação	Autodesk 3d Max ou Cinema 4d

5. Desenvolvimento do Protótipo

Neste capítulo serão descritas as várias fases do processo do desenvolvimento do protótipo do jogo, incluindo as funções que foram implementadas, escolha de tecnologias e como as fases se influenciaram mutuamente.

O primeiro subcapítulo incide na primeira fase do processo, a experimentação das tecnologias e técnicas e como estas induziram às modificações da proposta de design. O Segundo subcapítulo contém o design visual do jogo e a sua evolução. No terceiro, abordamos a produção das personagens, o que envolve o seu desenho, a construção dos seus *rig* e a sua mútua relação. No quarto designaram-se as várias animações implementadas no jogo, incluindo o seu design, amostras das animações e o mapa das suas relações com as outras animações. No quinto, referiremos a iluminação, materiais e *shaders* implementados no jogo. No sexto, explicaremos o comportamento e constituição da câmera de jogo e descreveremos o comportamento e a constituição da câmera de jogo. No sétimo, explicaremos em programação as várias funções que foram implementadas e as que não foram, e, no oitavo, mencionaremos os vários sons implementados.

5.1. Fase de Experimentação

Apercebi-me que a minha proposta inicial de design de jogo não correspondia à proposta da tese, pois a combinação 3D com 2D era só uma conjunção das suas estéticas em vez da combinação das técnicas. Como tal comecei por refletir em métodos que inovassem e simplificassem a combinação de 3D com 2D para artistas, animador com pouca experiencia, tais como:

- Substituir o processo complexo de animação das expressões faciais 3D por simplesmente desenhá-las em 2D no modelo 3D.

- Transformar imagens 2D em 3D, embora já haja aplicações deste método, cujos resultados estão restritos a objetos simples e básicos.

- O movimento dos modelos 3D serem baseados e criados a partir da animação de rigs 2D.

- Animar os modelos 3D fotograma por fotograma, de modo a não ser necessário criar rigs complexos.

- Aplicar o processo 2D de animação de fotograma por fotograma na animação 3D, sendo desnecessário a criação de rigs complexos. Este processo apesar de tornar animação 3D mais acessível para animadores 2D, apresenta contudo o inconveniente de aumentar a quantidade de trabalho a realizar.

Experimentei criar um *rig* em Blender com um modelo grátis, e, criei e pus em prática um cenário MultiPlane Camera effect em papel, e percebi o quão complexo são estes métodos e quanto tempo requerem.

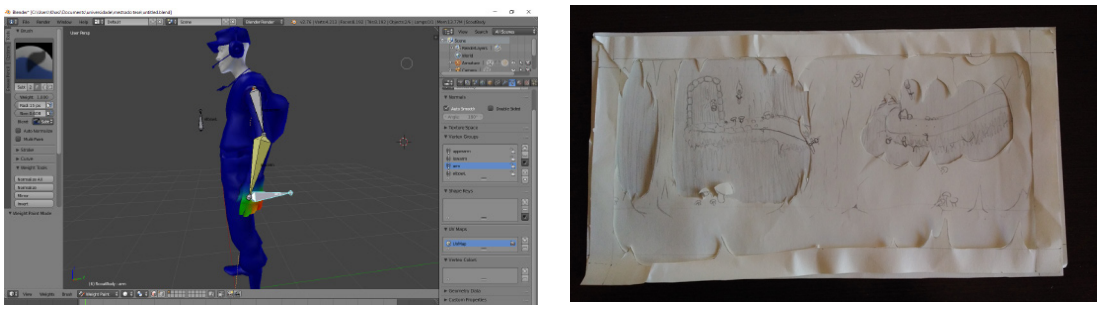

Fig. 39- Teste de rig em Blender e protótipo teste em papel de Multiplane Camera effect

Atendendo que os métodos anteriormente referidos só eram apropriados a certas situações especificas e requeriam conhecimento profundo de programação e ferramentas customizadas que não possuía, e em conjunto à experiencia que recebi com a experimentação com a criação de um rig e um cenário *MultiPlane Camera* effect, optei por progredir com animação modular e aplicar técnicas 3D em recursos 2D, de modo a ganharem profundidade e volume.

Tendo em conta a escolha de usar animação modular, decidi utilizar uma câmera *Side-Scroller* em vez da câmera *Top-view*, pois esta iria restringir ou complicar a galeria de animações possíveis das personagens, além que a câmera *Side-Scroller* permite uma visualização clara das animações.

Com a intenção de maximar o apelo e dinâmica dos eventos e animações do jogo, resolvi implementar ações automáticas na câmera, por exemplo, afastamento da câmera em relação às personagens ou a rotação da câmera. De modo a esta rodar sem denunciar os elementos do jogo como objetos de duas dimensões, pesquisei soluções em jogos com câmeras parecidas, GuiltyGearXrd e Darkest Dungeon. Como resultado aprendi GuiltyGearXrd utiliza modelos 3D que parecem ser 2D, a rotação da câmera de Darkest Dungeon é demasiada subtil, restrita. Como tal decidi simplificar as ações da câmera inspirando-me da câmera do Darkest Dungeon.

Fig. 40- Exemplo da rotação subtil da camera de Darkest Dungeon, este efeito representa a troca do turno do jogador com a dos rivais [Video]. Recuperado em https://www.youtube.com/watch?v=J-spazTn-4U

Simultaneamente experimentei criar um *Skydome* com Homeworld Universe Mod Tools, o que iria permitir criar um efeito de paisagem ideal independente do angulo da câmera, mas com a decisão de simplificar a câmera não procedi a estas experiencias.

Em vez de fazer um jogo turn-based, a acção do jogo é dividida por turnos, valorizando estratégia optei por fazer um jogo *Beat 'em up* , cuja ação é dinâmica, flexível e em tempo real às intenções do jogador, valorizando os reflexos do mesmo. Portanto permite o estudo e implementação das características distintas de animação em tempo real em jogos.

5.2. Design Visual

5.2.1 . Design das Personagens

Para ilustrar as personagens, fiz uma pesquisa de referencias célticas em conjunto com a produção do design das personagens, ou seja a criação dos seu visuais, de modo a estes refletirem a sua personalidade e passado, e, tendo em conta as necessidades do jogo. Por exemplo, para evitar gastar memória em desenhos da personagem a olhar para diferentes direções, o jogo faz *flip* à personagem, ou seja, roda a personagem 180º em relação a um eixo vertical o que implica que o seu design seja simétrico.

Fig. 41- Exemplos de referencias célticas. Recuperado em https://pt.pinterest.com/pin/454722893610378174/ e https://pt.pinterest.com/pin/287104544967889711/ e https://www.flickr.com/photos/marcinmg42/7065096681 e http://feinar.tumblr.com/private/24840776874/tumblr_m566oxTHR91r0x8pn

A Witch é uma rapariga jovem (14 anos) que sempre viveu na floresta, tendo pouco contacto social com a civilização, e como tal, o seu design inclui os seguintes elementos: Prático, oculto/bruxa, natureza, feminino.

Inspirando-me inicialmente em fantasmas asiáticas de raparigas, desenhei vários esboços da Witch para criar um modelo com uma forte silhueta e determinar o estilo de desenho que estaria confortável a repetir constantemente sem que houvesse inconsistências. Com estes esboços decidi que a Witch iria ter longo cabelo de cor de carvão, acentuando o elemento de oculto, e que iria caminhar descalça, acentuando o elemento de natureza.

Fig. 42- Exemplos de referencia de fantasmas asiáticas de raparigas. Recuperado em https://cinemaknifefight.wordpress.com/tag/ghosts-and-society/ e http://fyasianhorror.tumblr.com/post/52647748411/cheonyeo-gwishin

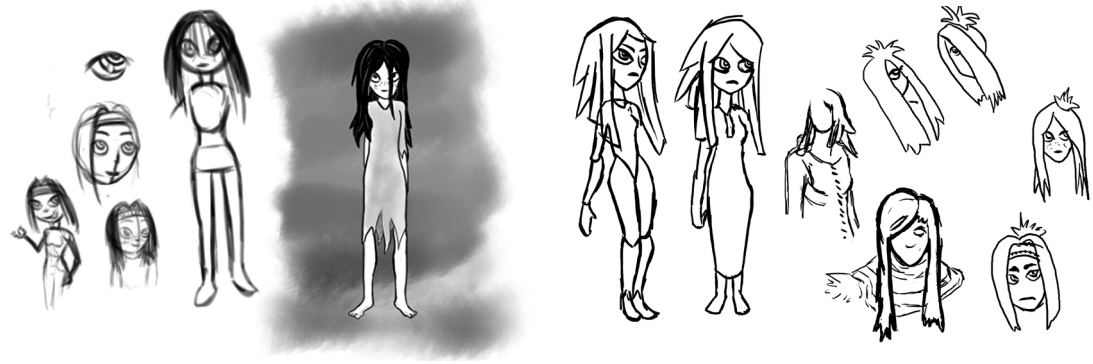

Fig. 43- Esboços da Witch

Ilustrei vária cabeças e roupas com o objectivo de escolher as opções cujo design representa melhor o conceito da personagem. As várias cabeças têm maquilhagens diferentes inspiradas pelas pinturas faciais de guerreiros célticos, portanto segundo *feedback*, selecionei a primeira cabeça.

Decidi utilizar a ultima cabeça em que parte da cara da Witch está escondida pelo seu cabelo, para quando ela está a 30% ou menos de vida, porque não só representa bem a ideia de estar ferida, mas também é uma feição tipica em fantasmas asiáticas de rapariga, reforçando o seu elemento de oculto

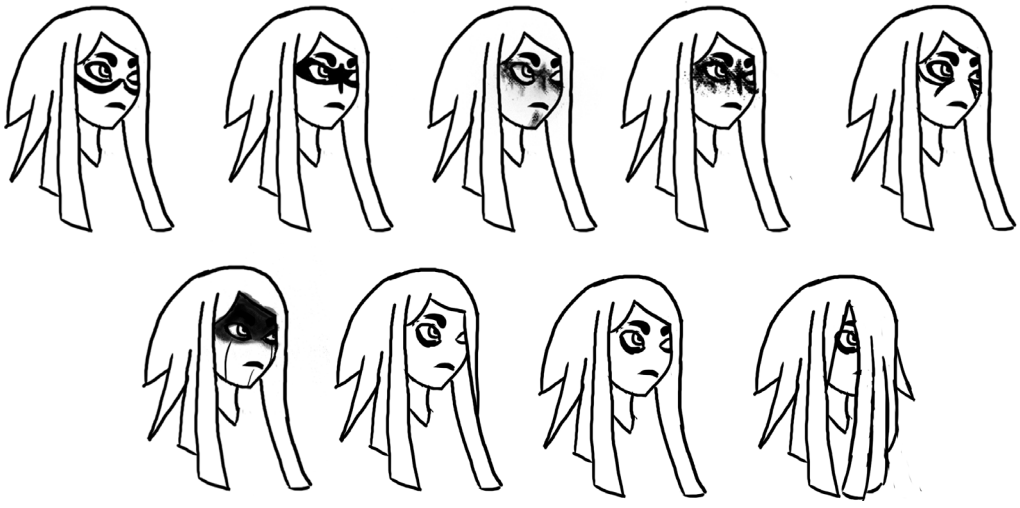

Fig. 44- Designs da cabeça da Witch

Os próximos passos do design das personagens demonstraram ser um desafio, visto que não tinha experiência com a ilustração de vestuário e pintura das mesmas. Portanto apesar do penúltimo vestuário ser básico, o seu desenho ser rápido, os seus rasgos representavam bem o isolamento da Witch numa floresta e pessoalmente projetava uma ideia de fantasia e magia, reconheci que não tinha a experiencia necessária em ilustração para desenhar a animação da saia. como tal escolhi o quarto vestuário visto que este também incorporava bem o conceito da Witch, explicitamente:

- Prático, possuir bolsos no seu cinto, uso de calças para se movimentar com facilidade na floresta.

- Natureza, a sua roupa ser constituída por ingredientes da floresta, por exemplo as penas.

- Oculto/bruxa, as penas dos seus adereços pertenciam à corvos

- Feminino, apesar de ter crescido sozinha na floresta, teve a iniciativa em usar maquilhagem, criar os seus adereços e roupas, evidenciado pelas suas costuras descobertas.

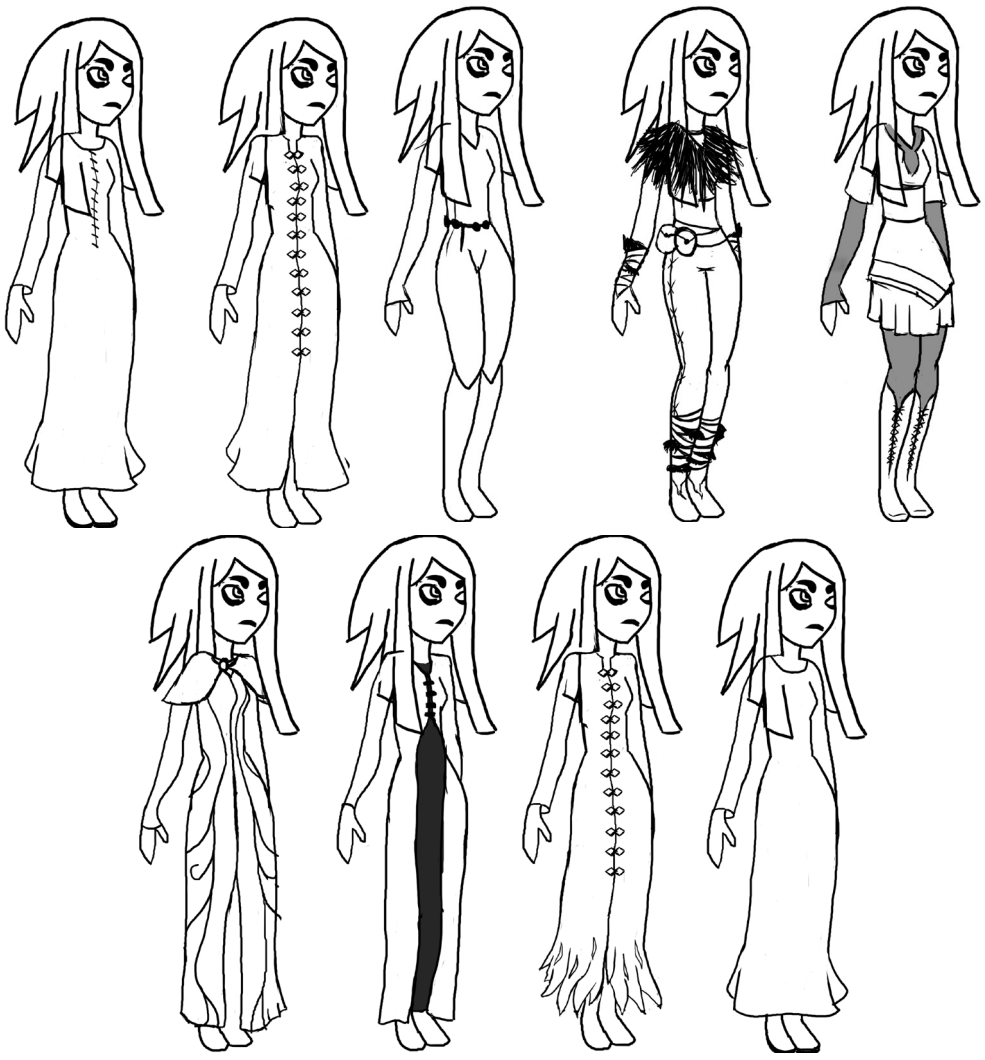

Fig. 45- Testes de design do vestuário da Witch

Realizei os testes cromáticos para escolher o padrão básico de cores e a sua distribuição no vestuário da Witch, de modo a reforçar o seu conceito e tendo em conta que é o protagonista do jogo, criar contraste com o cenário e outras personagens. Portanto escolhi preto que representa o elemento bruxaria e a cor vermelho cereja que não só se encontra na natureza mas também acentua a feminidade da Witch pela sua iniciativa em tingir a sua roupa.

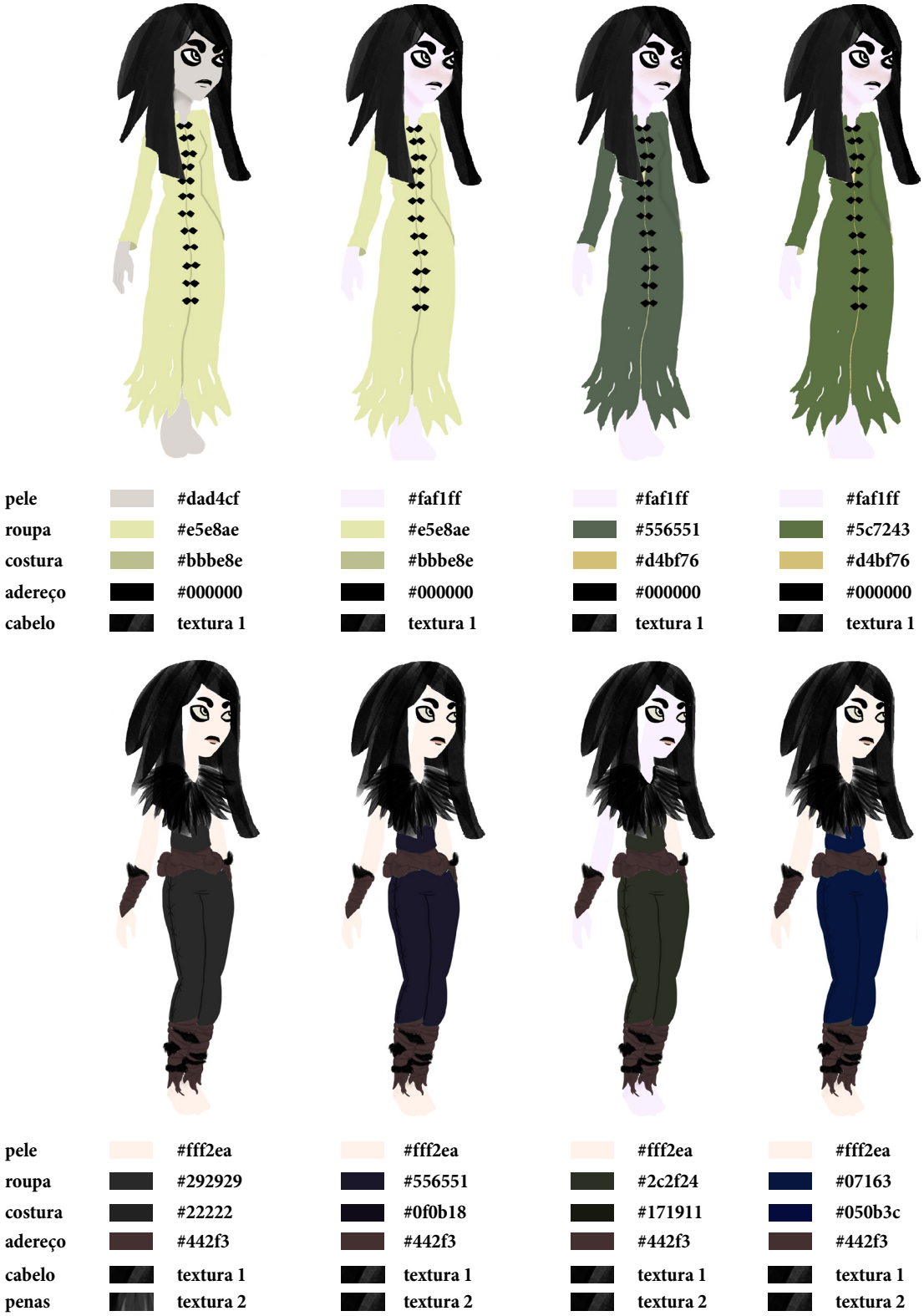

Fig. 46- Testes de cor do vestuário da Witch

Fig. 47- Testes de cor do vestuário da Witch

Fig. 48- Design final da Witch

No jogo, os bandidos são os inimigos da Witch, sendo caçadores e ladrões que invadiram a floresta, como tal o seu design inclui os seguintes elementos:

- Agressivo, parte do seu rosto é obscurecido pelo chapéu, armado..
- Robusto, musculado, calças com proteções de metal.
- Caçador, uso de troféus de caça como adereço.

O propósito dos esboços do bandido foi para definir o balanço entre um design rico de personalidade e uma estética genérica. Foi com a ilustração e animação do protótipo que me apercebi, que se deve evitar desenhar a linha de contorno (*outline*), pois ao longo da animação esta é quebrada nas zonas de articulação da marioneta.

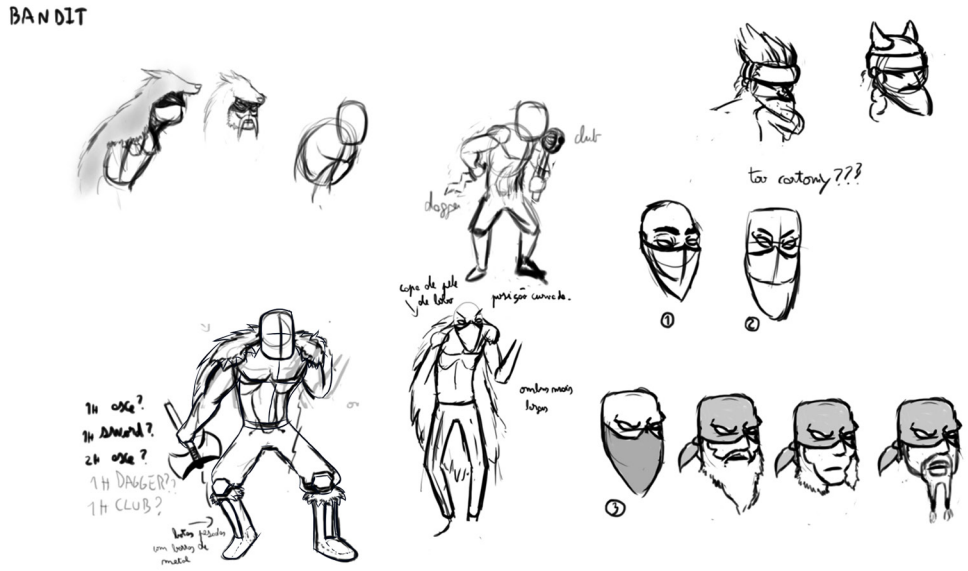

Fig. 49- Esboços do bandido

Fig. 50- Protótipo do bandido

Fig. 51- Design final do bandido

5.2.2. Design do Ambiente

Face à minha inexperiência no desenho de cenários, e considerando o local onde se desenrola o nível de jogo, decidi construir o cenário a partir de uma fotografia de uma floresta. Uma das utilidades da *pallete knife* é que facilita pintar traços mais largos, portanto ao aplicar o filtro *Pallete knife* do Photoshop, desenvolvida pela Adobe no cenário, a sua composição cromática ficou desconstruída em traços mais grossos, mas mantendo a sua forma de modo a ser reconhecível. Como resultado, o cenário ficou menos realista, tornando-se mais consistente com as personagens e mais fácil de o editar sem inconsistências gráficas.

Fig. 52- Foto do cenário sem filtro. Recuperado em http://www.heraldchronicle.com/sherwood-forest-conservation-effort-protects-endangered-wildlife-habitat-and-local-jobs/

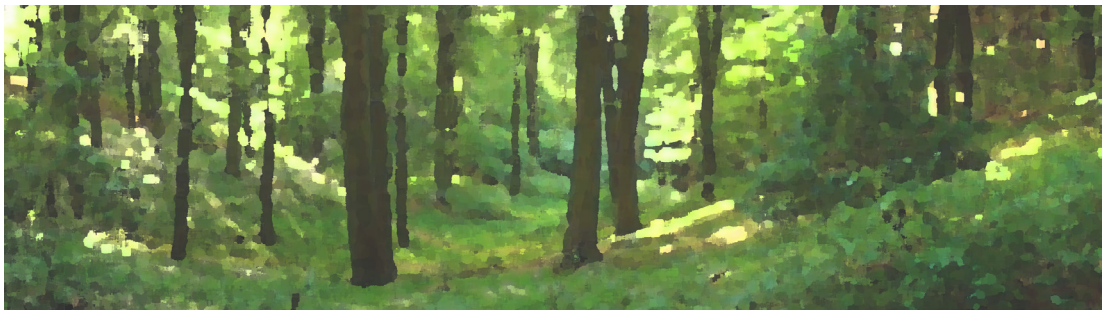

Fig. 53- Protótipo do plano de fundo

Fig. 54- Protótipo do plano médio

Fig. 55- Protótipo do plano frontal

Para aplicar o MultiPlane Camera effect, identifiquei os diferentes planos de profundidade da imagem e editei-as para criar os planos frontal, médio e de fundo.

Assim que implementei o MultiPlane Camera effect apercebi que apesar de ter implementado as *outline* e *inline* do plano frontal e médio para oferecer forma, corpo, volume aos elementos ilustrados, estas quebravam a ilusão de profundidade e criava inconsistência visual entre os planos, e, cada plano precisava de ajustes para reforçar a ilusão de profundidade.

Para que o protótipo do plano de fundo ocupasse completamente o fundo do jogo, tive que aumentar o seu tamanho, consequentemente criava árvores gigantes que quebravam a perspectiva e ilusão de profundidade. Portanto a partir da edição da imagem, tornei-a mais extensa.

Fig. 56- Imagem do plano de fundo do cenário

Tanto como no plano médio como no plano frontal, retirei conteúdo de modo a ser possível visualizar mais claramente os diferentes planos.

Fig. 57- Imagem do plano médio do cenário

Tendo em conta que o plano frontal é o plano mais próximo do jogador, decidi aumentar o diâmetro das árvores e decidi fazer cortes em forma de relva na base do plano frontal, com o propósito a fundir com o terreno, ocultando que este é tecnicamente uma imagem 2D assente perpendicularmente no terreno.

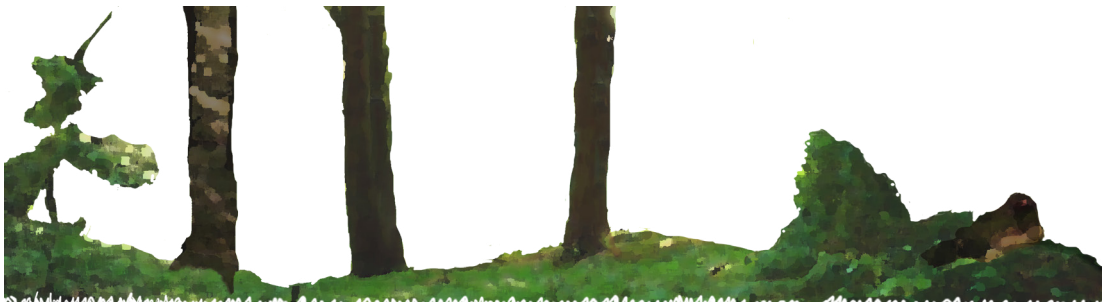

Fig. 58- Imagem do plano frontal do cenário

Criei um plano adicional que o designei por plano de bosque, este localiza-se entre o jogador e as personagens, com o proposito de o conectar com o jogo, pois esta cria a ilusão de que o jogador encontra-se a observar no bosque a ação.

Fig. 59- Imagem do plano de bosque

Construí a textura do terreno do cenário no jogo através da combinação de três texturas, tendo em consideração que o terreno teria que fundir com o plano frontal.

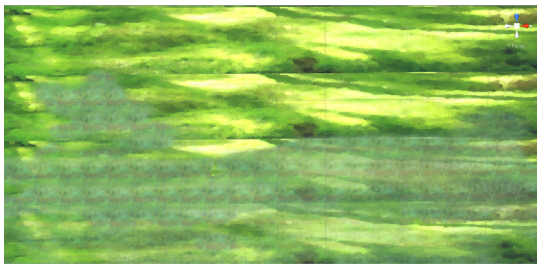

Fig. 60- Imagem do terreno

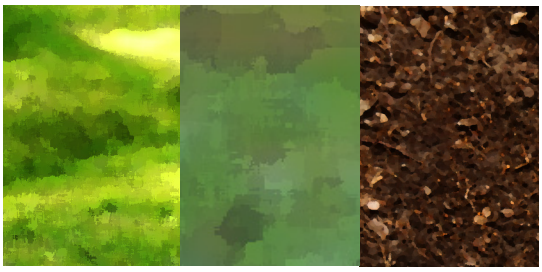

Fig. 61- Amostras das texturas do terreno

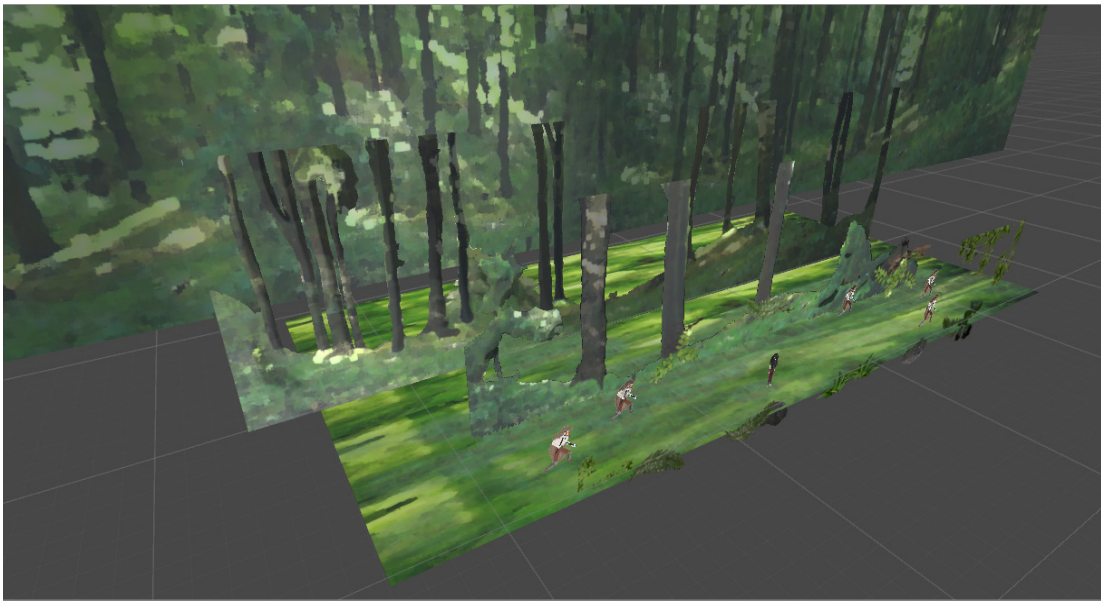

Fig. 62- Resultado final do cenário

Fig. 63- Resultado final do cenário do ponto de vista do jogador

5.2.3. Design de GUI (Graphical User Interface)

O design da barra de vida da Witch foi inspirado pelos ornamentos célticos, especificamente um laço de nós celta, acompanhado de uma base meia translúcida para reforçar a distinção entre a barra de vida e o cenário. Para facilitar o reconhecimento ao jogador da ligação da Witch com a barra de vida, estes dois partilham o mesmo padrão de cores - preto e vermelho cereja.

Experimentei com vários protótipos, com o propósito de encontrar o design da barra de vida, cuja leitura fosse clara e fácil ao jogador, que condissesse com o visual do jogo e também oferecesse contraste com o cenário.

Fig. 64- Amostras dos protótipos da barra de vida

Fig. 65- Elementos da barra de vida

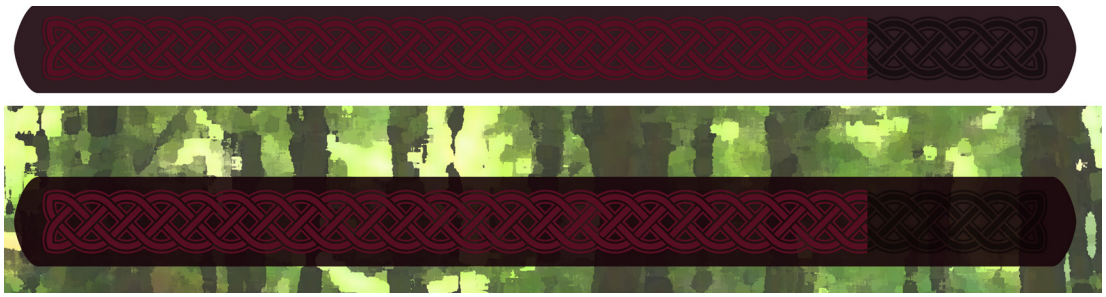

Fig. 66- Design final da barra de vida

5.3. Produção das Personagens

Explorei o programa Spriter, construí e animei um modelo, marioneta teste. Mais tarde, ao aperceber-me que a aplicação responsável para exportar a marioneta do Spriter para Unity foi criada recentemente por fãs e ainda tinha *bugs* e restrições de controlo da marioneta em Unity, optei por construír o rig das marionetas no *engine* 3D do Unity. Assim, não só os *rigs* têm a liberdade de movimentar nas 3 dimensões, como também tenho o seu controlo total. Por exemplo, através de código ou das ferramentas do Unity, consigo criar eventos do jogo que alteram partes da marioneta. Em contrapartida o rig da marioneta é básico, portanto não consegue suportar diferentes sprites numa parte do seu corpo sem exercer muita carga de trabalho ao processador.

As personagens são desenhadas e divididas em partes da marioneta no Photoshop, que mais tarde são guardadas em formato de *texture map*. Através do Unity estas são divididas, em que todos os seus pedaços são categorizados e atribuídos ao rig marioneta, formando a personagem. A desvantagem deste processo, é que só é possível certificar que os desenhos da personagem estão bem construídos para serem animados no final deste processo. O que significa que o menor erro detetado implica corrigir e repetir todo o processo.

Fig. 67- Raio-X das marionetas das personagnes

Fig. 68- *Texture maps* da Witch

Fig. 69- *Texture maps* do bandido

Para perceber como profissionais constroem os *rigs* e ilustram as personagens em animação modular, adquiri os vários segmentos que compõe as personagens do Dragon Crown (2013). Através de uma imagem da personagem Amazon e Fighter como base, e, com os segmentos colecionados, reconstruí as personagens de modo a ajudar-me a compreender a lógica e cuidados que devo ter na criação de animação modular.

Uma das desvantagens de usar *rig* marioneta, é que a ilustração das suas zonas de articulação necessitam de cuidado e atenção. Por exemplo, na linha de costura que atravessa as calças da Witch, não importa o ângulo entre as pernas, as linhas de costura têm de conectar-se entre si. Mas no caso da articulação entre a anca e a perna da Witch, quando esta se levanta, supostamente o tecido e a linha de costura deviam ficar deformados, sendo necessário acrescentar uma peça específica ao rig para representar a deformidade.

Para indicar ao jogador que a Witch tem pouca vida, 30% da barra de vida ou menos, a personagem está programada para ficar com cabelo à frente da cara. Tipicamente isto significa ter que criar um conjunto repetido de animações da Witch com o cabelo à frente da cara, principalmente em animação de *sprites*, mas como estamos a utilizar animação modular, só é necessário acrescentar as peças de cabelo ao rig, poupando muita memória.

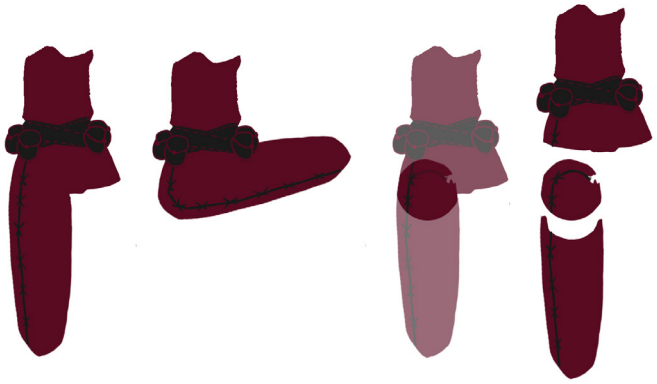

Fig. 70- Exemplo da deformação

Fig. 71- Witch com pouca vida

Para facilitar o processo de animação, tentei incorporar código no jogo para poder utilizar *rigs* IK (Inverse Kinematics), contudo, depois de alguns testes, os rigs IK tiverem que ser descartados porque só funcionariam com eficiência em animações simples.

5.4. Animação das Personagens

Tipicamente jogos preferem ter entre 30 a 60 fps, como tal decidi criar as animações no jogo a 60 fps, sendo o ideal para jogos de computador. Estas animações resultam da combinação de animação modular e animação de sprites, o que possibilita que a da troca sequencial dos fragmento da marioneta, não só tornando o seu movimento mais fluido, mas também cria a ilusão de movimento fluido de sprites 2D no terceiro eixo de coordenadas (3D).

Em jogos, as animações estão naturalmente condicionadas às ações do jogador, portanto o jogo tem que conter mapas que indicam em que condições as animações são iniciadas, quantas vezes se repetem e quando é que estas podem transitar para outras animações.

Os mapas de animação entre a Witch e o bandido são diferentes, visto que ao contrário do bandido que pode ser harmed em qualquer momento, a Witch só pode ser *harmed* quando não está a atacar. Sem esta diferença o jogo ficaria mais difícil.

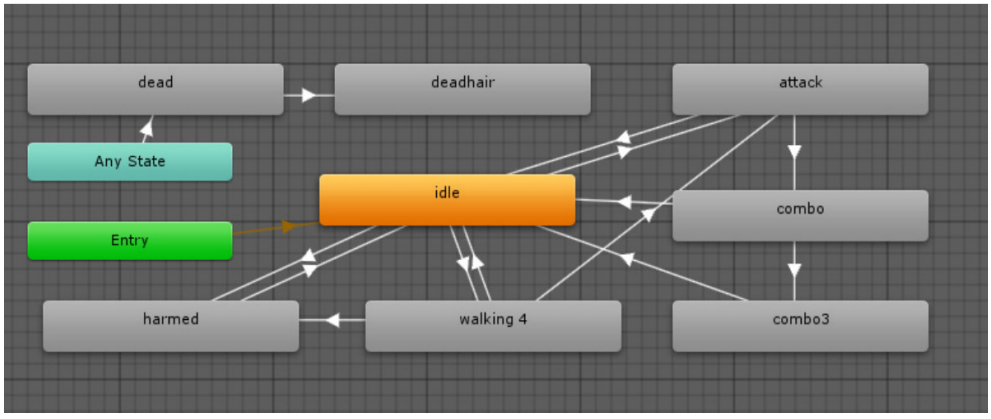

Fig. 72- Mapa de Animação da Witch

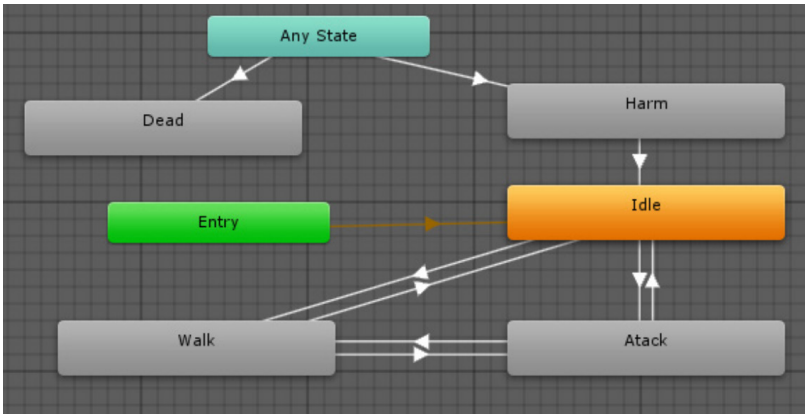

Fig. 73- Mapa de Animação do Bandido

dead- morrer
deadhair- animação de game over
idle- respirar, enquanto espera por ordens
harmed, harm- personagem a ser magoada
walking 4, walk- andar
attack, atack- atacar
combo, combo3- ataque consecutivo

As personagens só andam na direção este e oeste, porque cheguei à conclusão que precisaria no mínimo do triplo do trabalho para as desenhar a andar para o norte e sul, criar os seus rigs e as suas animações, pelo que não teria tempo. Não implementei a animação de *dash*, porque no final desta animação a Witch ficava sempre na posição errada e não consegui corrigir este problema. Apesar de ter escrito um código que corrigia a posição da Witch, havia algum elemento desconhecido no Unity que alterava a posição.

Criei *storyboards* simples das animações, o que me ajudou a reproduzir com eficiência os movimentos das personagens.

Decidi não criar as animações de magia que evocasse criaturas, projéteis e laser, porque requeria adição e modificação do código do jogo, além de diminuir o desafio do jogo retirando-lhe interesse.

Não implementei as animações de salto porque a Inteligência artificial responsável pelo movimento dos bandidos é uma componente do Unity designada por Nav Mesh Agent, mas esta não inclui a habilidade de os fazer saltar.

Para que o jogador se sinta confortável a controlar o protagonista, a Witch, ele necessita sincronização entre as suas intenções e as ações da Witch. Para tal a antecipação destas ações foram encurtadas. Para garantir que a animação do golpe da Witch (Figura. 79) fosse inteligível apesar de ser curta, foi aplicado o efeito de *smear* nas chamas.

Para realçar a agressividade do bandido, os seus passos são violentos como se estivesse constantemente a pisar com força o chão, como observado na Figura 81.

A antecipação do ataque dos bandidos é longo, com o proposito do jogador ter tempo para reconhecer e reagir ao perigo. Da animação de ataque do bandido (Figura 84), aumentei o tamanho dos machados para criar a ilusão de peso e implementei um *smear* do rastro dos machados para tornar a animação mais inteligível.

Animações da Witch

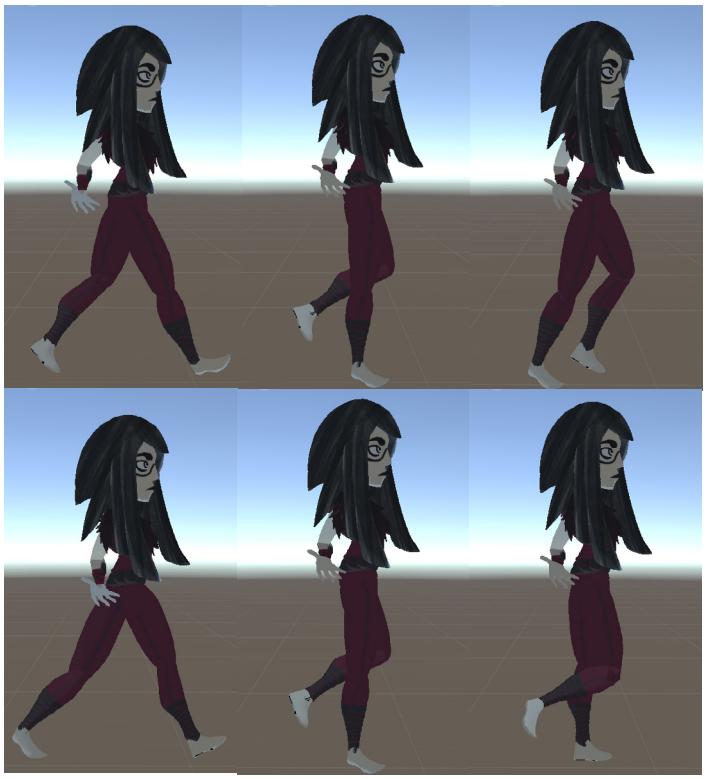

Fig. 74- walking 4: Animação de andar

Fig. 75- dead: Animação da sua morte

Fig. 76- deadhair: Animação loop do seu estado de morta

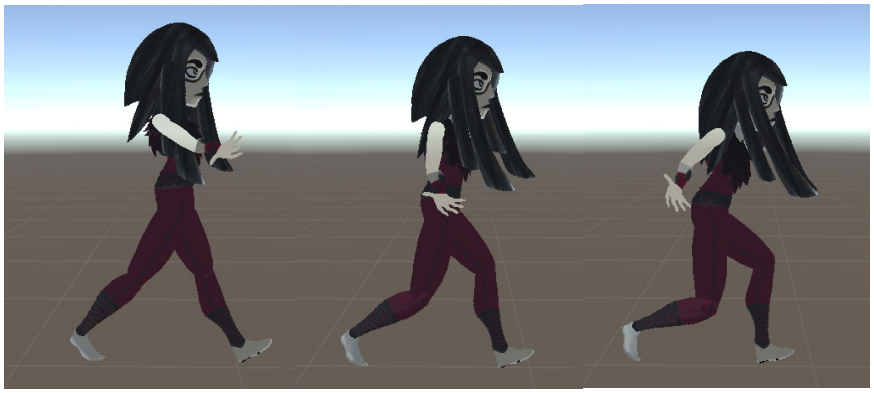

Fig. 77- harmed: Animação de ser magoada

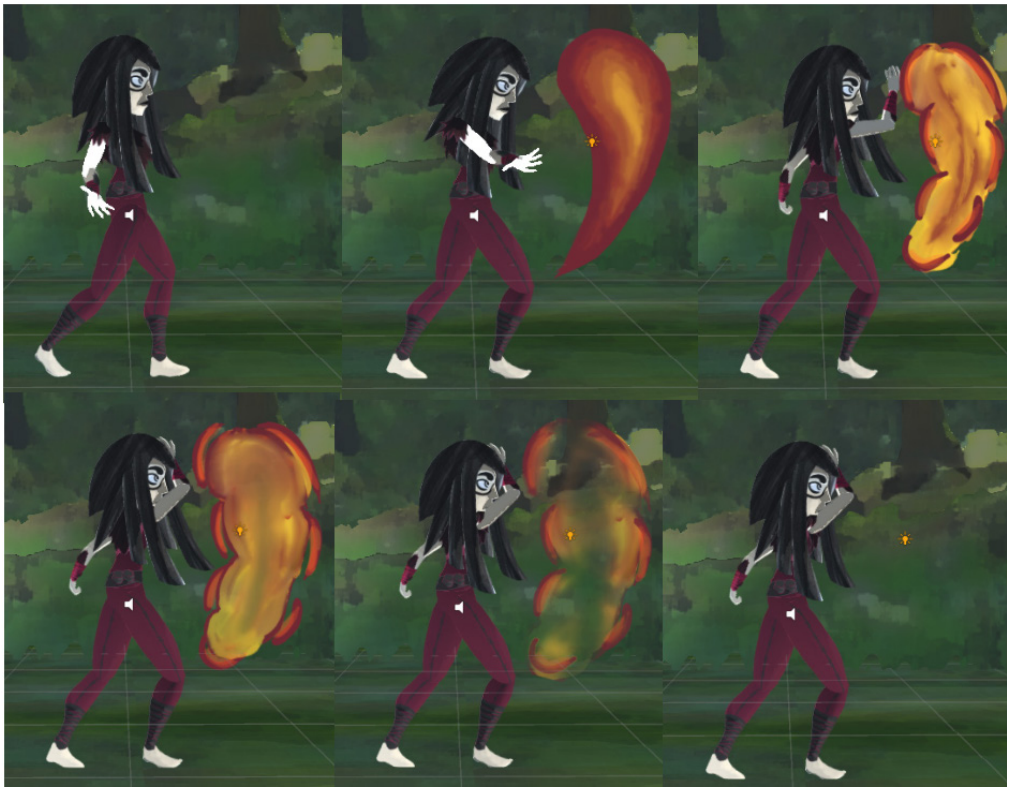

Fig. 78- attack: Animação de ataque

Fig. 79- combo: Animação do primeiro ataque consecutivo

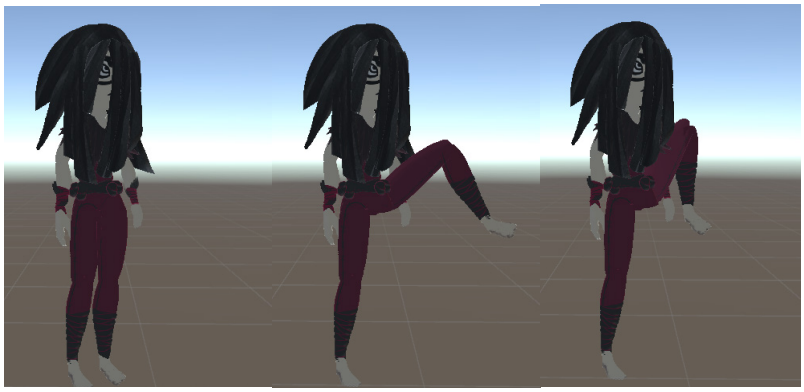

Fig. 80- combo3: Animação do segundo ataque consecutivo

Animações do bandido

Fig. 81- walk: Animação de caminhar

Fig. 82- harm: Animação de receber ferida

Fig. 83- dead: Animação de morte

Fig. 84- atack: Animação de ataque

5.5. Iluminação

No jogo, a luz do sol é simulada por uma *Directional light*, e os efeitos especiais do golpe da Witch, chamas, possuí uma *Point Light*.

Com a intenção de reforçar a ilusão objetos 2D terem volume, decidi aplicar material ao cenário, personagens, assim estes são afetados pelas fontes de iluminação no Unity.

Com a aplicação de um material numa personagem, estamos ao mesmo tempo indicar automaticamente ao Unity que só deve renderizar a face da frente da personagem. Portanto assim que a personagem muda de direção, faz flip, a face de trás da personagem fica revelada mas não é renderizada, ou seja não é visível, fica transparente. Para que a ambas faces sejam renderizadas, tive que alterar o código do *shader* do Unity e aplicá-lo às personagens.

No âmbito de introduzir a técnica de iluminação 3D em objetos 2D, de modo a criar a ilusão de volume em imagens, tentei usar *oclusion maps*, e, criar *normal maps* através da NVidea Photoshop Plugin, conversor automático da Unity e desenho. Como não fiquei satisfeito com os resultados inspirei-me do tutorial Normal Map Photography (2007) de Ryan Clark e desenhei as sombras dos *sprites* das personagens e dos cenários em 4 direções diferentes da fonte de luz, mais tarde estas foram combinadas resultando os *normal maps*.

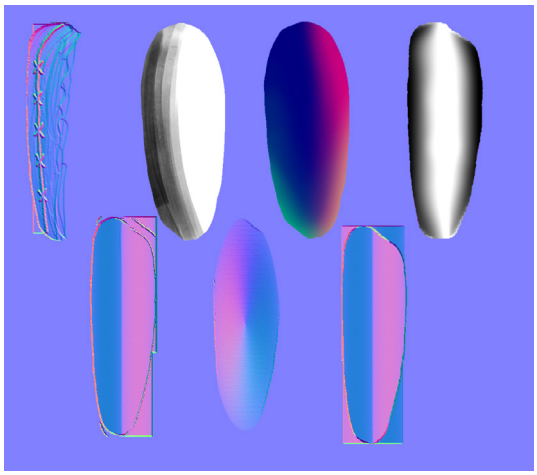

Fig. 85- Tentativas de *Normal maps* e *Oclusion maps*

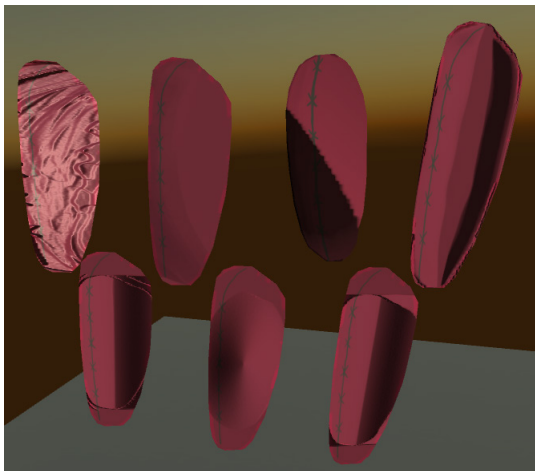

Fig. 86- Resultado das tentativas dos *maps*

Fig. 87- *texture map* original da Witch

Fig. 88- *normal map* dos sprites da Witch

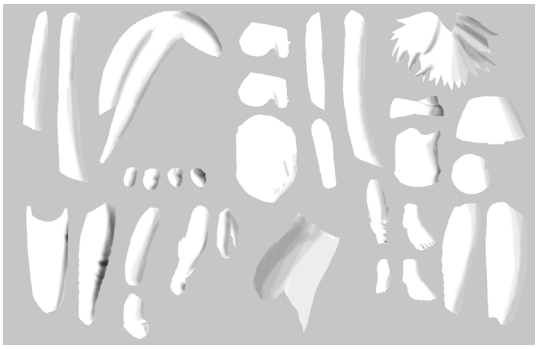

Fig. 89- Imagem das sombras com a luz à esquerda

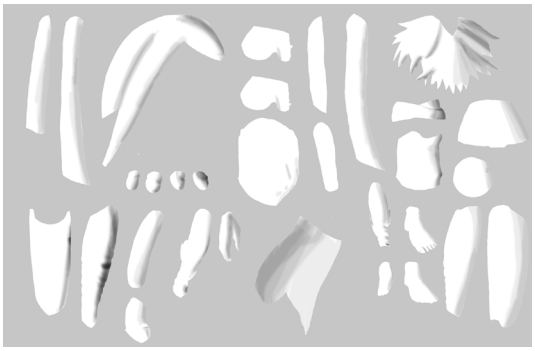

Fig. 90- Imagem das sombras com a luz à direita

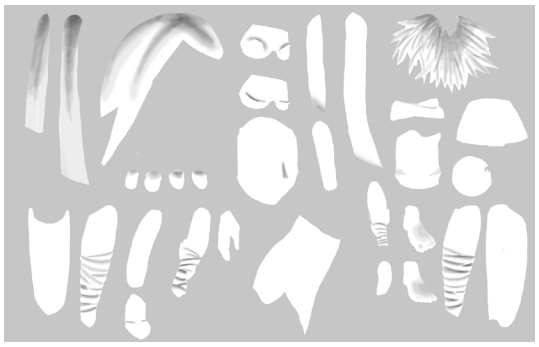

Fig. 91- Imagem das sombras com a luz em baixo

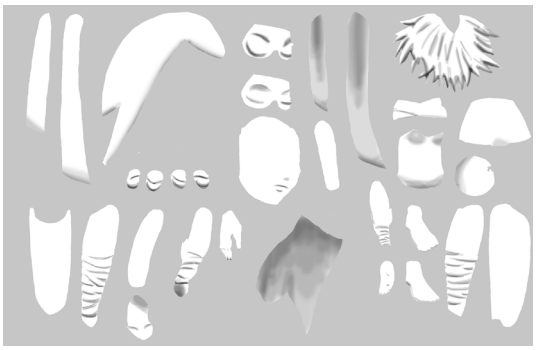

Fig. 92- Imagem das sombras com a luz em cima

Tendo em conta que usamos animação modular, com a aplicação de material à personagem, estamos tecnicamente à aplicar o material às varias componentes da marioneta da personagem. Em conjunto com o uso de *normal maps*, torna-se obvio ao jogador que a personagem é constituído por vários componentes, para evitar tal efeito tive que modificar manualmente os *normal maps* e escolher com que o *shader* desvaneça as extremidades dos *sprites* da marionetas, de modo a melhorar a sua união.

Fig. 93- Witch sem as extremidades dos *sprites* desvanecidas

Quando uma personagem faz *flip* do jogo, a direção das *normal lines,* responsáveis por calcular como a luz afeta a superfície da imagem, ficam no sentido oposto, portanto induz uma reação errada à iluminação. Apesar deste dilema ter várias soluções, não as consigo por em prática. Tal como:

- Customizar o *shader* de modo a corrigir a direção as *normal lines,* mas requer conhecimento que não possuo.

- Criar atrás na personagem uma cópia com as *normal lines* na direção correcta, mas neste caso é uma solução ineficiente, porque exerceria muita carga de trabalho ao processador.

- Através dum software 3D, tornar os fragmentos das personagens como *double layered objects,* ou seja, em imagens em que ambas superfícies possuem *normal lines.* Apesar desta ser melhor a solução, requeria que repetisse o processo de criação e animação das personagens, para a qual não tenho tempo.

O plano de fundo não tem um normal map, para evitar com que este seja afetado pelas iluminação do golpe da Witch, visto que este supostamente encontra-se longe das personagens.

5.6. Câmera do Jogo

Implementei uma câmera Side-Scroller, que acompanha a Witch no eixo de ordenadas X. A câmera encontra-se próxima da Witch, permitindo ao jogador observá-la bem. Mas, com a presença de inimigos, o jogador necessita de um maior campo de visão para obter informação do espaço e o posicionamento dos bandidos. Portanto, sempre que um destes aparece na visão da câmera, esta afasta-se gradualmente do cenário aumentando o seu campo de visão.

Tecnicamente a câmera é constituída por duas câmeras, uma que é responsável por renderizar o GUI (Graphical user interface) e outra que renderiza o restante. Este método do Unity é de modo a garantir que não importa o que haja entre a câmera e GUI, este aparece sempre sobreposto ao restante .

O modo como a câmera é utilizada durante o jogo tem um impacto sobre a perceção do jogador em relação às animações, tal como reforçar o peso e o valor das animações. Considerei fazer com que câmera rodasse ligeiramente ou que se aproximasse das personagens em determinadas animações, para reforçar o seu peso e torná-las mais dinâmicas, mas conclui que acabaria por distrair e atrapalhar o jogador.

Para reforçar a ilusão de volume das personagens e cenário 2D, criei *oclusion maps* e *normal maps*. Portanto desenhei e apliquei *occlusion maps* nos materiais, o que fez com que as personagens ganhassem sombreamento. Realizei testes com *normal maps*, mas apercebi-me que não tinha a experiência para as criar com boa qualidade para que o resultado fosse realista.

5.7. Progamação do Jogo

Uma das razões para ter escolhido Unity era porque funcionava com código javascript, entretanto preferi aprender e utilizar antes o código C#, por me aperceber que havia um maior apoio neste código da comunidade Unity, com diversos conselhos e tutoriais úteis para o desenvolvimento do jogo.

Ainda implementei um sistema de combos, ou seja uma sequencia de ataques, cuja realização depende do timing do jogador do pressionamento das teclas de ataque. Este tem o propósito de mostrar uma maior galeria de animações e tornar o jogo mais interessante. Todavia requer do jogador muita coordenação para efetuar combos.

Na câmera tentei implementar um código, que, por cada bandido que entrasse na sua visão, esta afastava-se para aumentar o campo de visão. Mas isso envolvia descobrir constantemente as novas coordenadas dos bandidos, o que sobrecarregava o processador, tornando o jogo lento.

A introdução de um sistema de alteração do mundo, corrupção da natureza, maldições no jogo segundo o uso excessivo dos poderes da Witch, tornaria este nível de jogo mais interessante e divertido, contudo tal processo não se mostrou exequível dado exigir demasiado tempo para criar o seu design e código.

Os códigos que implementei e adaptei foram baseados nos tutoriais dos canais de youtube CasanisPlays e C Sharp Accent Tutorials

5.8. Áudio

O áudio ajuda a imergir os jogadores no jogo, portanto também ajuda imergir, sentir a animação. Na pesquisa de sons, só selecionei os que tivessem uma licença creative commons.

Os sons de *Music,* estes ajudam a transmitir emoção:

- Uma música céltica, que fosse feita de modo a que pudesse ser repetida sem que a sua melodia sofre-se disrupções. Transmite uma atmosfera de tranquilidade.

  Slaking-97. (2015, 30 de Dezembro).Fairytale (Harp & Chimes). Em *Freesound.* Recuperado em https://www.freesound.org/people/Slaking_97/sounds/332024/

- Música que só ocorre na morte da Witch .

  Alexnekita. (2015, 3 de Junho). Impact - Lost in the woods. Em *Freesound.* Recuperado em https://www.freesound.org/people/alexnekita/sounds/346891/

A Ambience, contém os sons que ajudam a construir o espaço do jogo:

- Sons da floresta

  Porphyr. (2014, 10 de Junho). Forest Soundscape (Thuringian Forest). Em *Freesound.* Recuperado em https://www.freesound.org/people/Porphyr/sounds/240339/

Os sons de *Foley* normalmente acompanham com as ações das personagens, tornando-as mais inteligíveis. Logo são importantes em animação, porque reforça a percepção dos jogadores no que respeita às movimentos instantâneos:

- Witch magoada

11linda. (2016, 5 de Maio). dying female. Em *Freesound.* Recuperado em https://www.freesound.org/people/11linda/sounds/345049/

- Ataque do Bandido

Tiger_v15. (2014, 30 de Julho). Whooshes Whips and Swooshes. Em *Freesound.* Recuperado em https://www.freesound.org/people/Tiger_v15/sounds/243752/

- Andar da Witch

JanKoehl. (2009, 17 de Dezembro). walk-forest02. Em *Freesound.* Recuperado em https://www.freesound.org/people/JanKoehl/sounds/85602/

- Não implementei os sons dos bandidos a andarem, porque criaria uma superabundância de sons.

Os *Sound Effects (SFX)* tornam os objetos e fenômenos digitais mais realistas:

- Magia do fogo

Niedec. (2015, 6 de Janeiro). Basic Fire whoosh 3. Em *Freesound.* Recuperado em https://www.freesound.org/people/Niedec/sounds/260584/

- Som de uma trompa, ocorre para denunciar o aparecimento, ataque dos bandidos.

Porphyr. (2013, 21 de Maio). Battle Horn. Em *Freesound.* Recuperado em https://www.freesound.org/people/Porphyr/sounds/188815/

6. Teste de Apreciação

No decurso do desenvolvimento do protótipo do jogo, foram disponibilizados ao público dois conjuntos constituídos por uma versão do jogo e o seu questionário de apreciação. Que serão referidos e estudados neste capitulo.

O primeiro subcapítulo contém a natureza da informação que se pretende recolher com os testes de apreciação. Os seguintes subcapítulos correspondem à cada um dos testes de apreciação, incluindo a analise dos seus resultados e que modificações ao jogo estes inspiraram.

6.1. Intenções dos Testes de Apreciação

Com a implementação dos testes de apreciação pretendia-se colecionar a experiencia do publico com o jogo, nomeadamente:

- Se a visualização das animações proporcionaram prazer aos jogadores?

- Nas interações entre as personagens, as animações eram claras? Por exemplo: os bandidos reagiam propriamente as ações da protagonista?

- O jogador teve dificuldades em controlar a Witch? As ações da Witch correspondiam aos comandos do jogador?

- Avaliar se o publico percepcionava a constituição do jogo por elementos planos, de duas dimensões, ou, de três dimensões, com volume e profundidade.

- Averiguar se o jogador seria capaz de identificar as técnicas de animação. Se as técnicas de animação estavam de tal modo bem implementados que eram difíceis de distinguir?

6.2. Resultados do Primeiro Teste de Apreciação

Este questionário foi respondido por 7 pessoas e contém as seguintes questões:

1. Consideras a animação fluída?

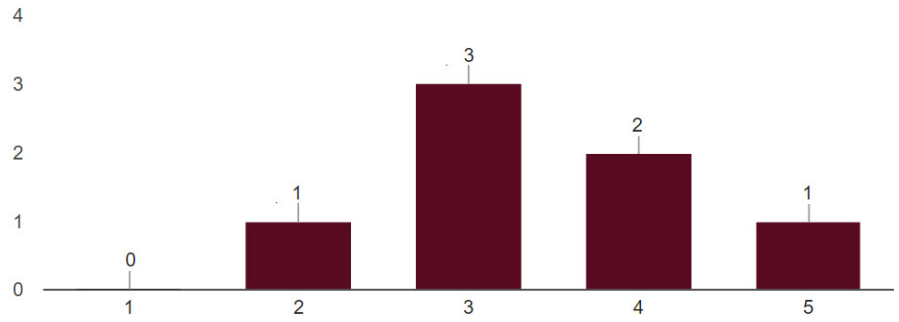

Fig. 94- Gráfico dos resultados da 1º pergunta com escala de valores 1-5

2. Consideras a animação *responsive*?

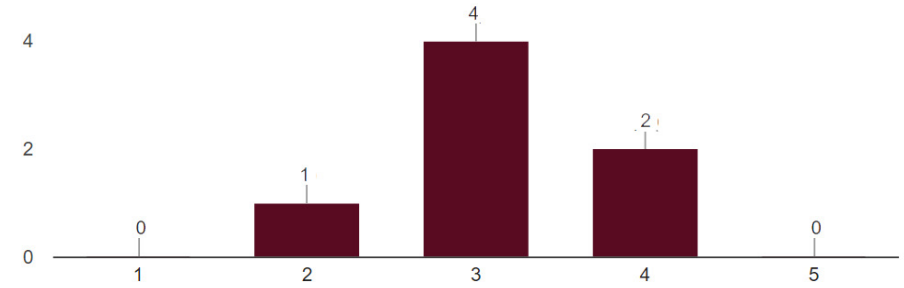

Fig. 95- Gráfico dos resultados da 2º pergunta com escala de valores 1-5

3. Consegues indicar o tipo de animação? Se sim qual?

Quatro das sete pessoas responderam esta questão, uma resposta positiva "3D side scroller. " e as respostas negativas: "3D side scroller.", "A animação dos ataques tanto da personagem como dos inimigos por vezes não aparece.", "Tipo de animação?" e "Não".

4. O jogo tem objectos 3D?

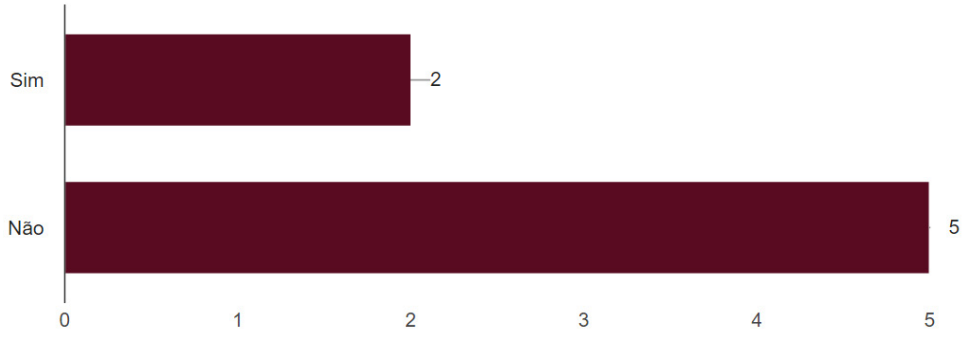

Fig. 96- Gráfico dos resultados da 4º pergunta

5. Se respondeu Sim anteriormente, indique os objectos 3D?

- "Tem a noção de profundidade, tendo movimentos das personagem nos 3 planos"

- "Background em profundidade; as personagens aumentam ou diminuem consoante a distância."

6.2.1. Análise do Primeiro Teste

Devido ao tempo necessário para o desenvolvimento do protótipo do jogo, o período do primeiro teste de apreciação foi curto, portanto apesar das respostas recolhidas não terem sido suficientes para alcançar uma conclusão sobre o sucesso do protótipo, permitiram detectar *bugs*, problemas com animação, com o controlo da Witch e com o formulário.

Em ordem a melhorar o questionário, estes tem que evitar o uso de termos técnicos, pois o publico geral demonstrou dificuldade em os interpretar. Como tal decidi excluir a pergunta Nº3 (identificação do tipo de animação). Percebi que na minha tentativa com que a pergunta Nº4 e 5 não influenciasse a opinião do jogador, tornei-as demasiado ambíguos.

Segundo os resultados dos testes, alguns dos jogadores sentiram que havia falhas na animação de ataque, especificamente que ocasionalmente a animação de ataque eram ininteligíveis. Depois de uma investigação, determinei que este problema foi induzido porque as animações de combo tinham menor impacto do que o primeiro ataque da Witch, como tal estas tornavam-se facilmente despercebidas, e, o código do jogo não permitia o jogador atacar os bandidos enquanto ele pressiona-se um dos botões de movimento, ao qual dificultava o controlo da Witch.

Na investigação da inteligibilidade de animação de ataque, apercebi que como na interação entre personagens estes não sofriam a força de impacto, as animações colidiam, sobrepõem entre si tornado a interação enganosa.

O jogo tinha um *bug* em que no cruzamento entre personagens, partes do corpo de uma personagens sobreponham outras personagens. Este fenômeno sucedeu devido ao modo de construção dos *rigs*, utilizei uma ferramenta que estabelecia a ordem em que as partes dos corpos eram sobrepostas, mas esta ordem também afetava outras personagens.

Fig. 97- Exemplos do bug de sobreposição

6.2.2 Primeiras Modificações ao Jogo

Em ordem a corrigir a intangibilidade da animação de ataque, emendei o código de modo a aceitar comandos de ataque enquanto as teclas de movimento estão pressionadas e para os ataques de combo terem maior impacto criei novas animações.

Utilizei esta oportunidade para recriar o movimento dos ataques de combo, de modo a que se assemelhem mais com uma cadeia de ataques interligados do que 3 ações isoladas. Redesenhei a animação das chamas para as tornar mais fluidas, e, escolhi implementar chamas de diferentes categorias, com o proposito de visualizar como os *normal maps* reagem com iluminações de diferentes cores.

Fig. 98- combo: Nova animação do primeiro ataque consecutivo

Fig. 99- *Texture maps* das chamas de "combo"

Na criação do ultimo ataque consecutivo, tive em consideração que este tinha que recompensar o jogador por concluir a cadeia de ataques, como tal tinha que ser satisfatório e resultar uma forte impressão. Portanto o design das chamas inclui a cor cereja, característico da Witch, cuja estrutura foi inspirada por um laço de nós celta e uma garra, representando agressividade.

Fig. 100- combo3: Nova animação do segundo ataque consecutivo

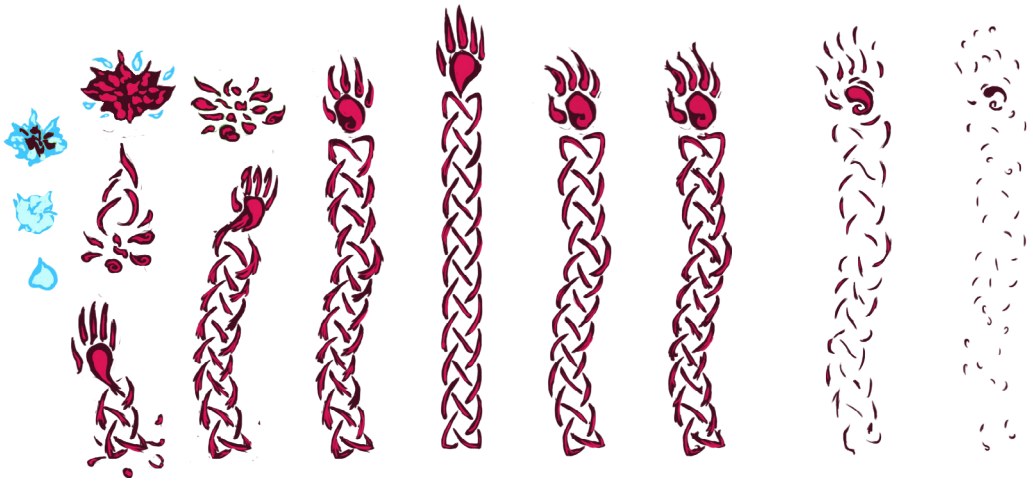

Fig. 101- *Texture maps* das chamas de "combo3"

Com os novos ataques consecutivos tive que redesenhar as chamas do primeiro golpe da Witch, com o objectivo de manter a harmonia visual das animações.

Fig. 102- attack: Nova animação do primeiro golpe

Fig. 103- *Texture maps* das chamas de "attack"

Para evitar sobreposição das animações de interação, aumentei o alcance do ataque do bandido e programei-os a serem vulneráveis à força de impacto dos ataques da Witch, portanto não só cria um maior espaço entre as personagens, como também reforça a ilusão de força das animações de golpes.

Deixei de aplicar a ferramenta que estabelecia a ordem dos *sprites* das personagens, consequentemente estes tornaram-se instáveis, ou seja, o Unity tinha dificuldade em renderizar corretamente a sua ordenação. Para resolver o dilema com renderização, tendo em conta que não tinha um período de tempo razoável para obter os recursos necessários, optei por aumentar o espaço entre os *sprites* e ajustei as animações. Assim o Unity teria menor dificuldade de reconhecer a ordem correta dos *sprites* das personagens, embora como consequência requereu ajuste às animações.

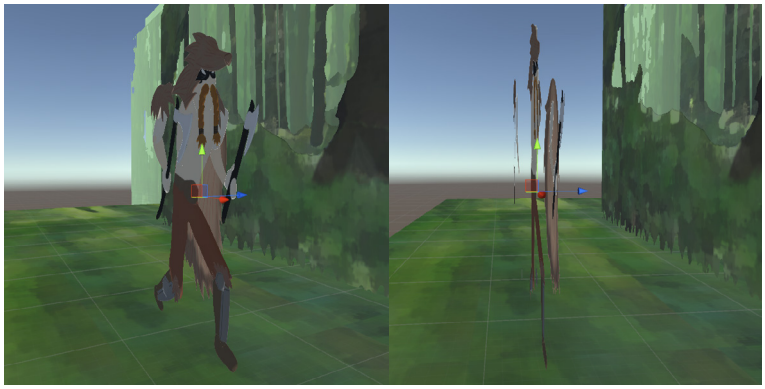

Fig. 103- exemplo do espaço entre sprites

6.3. Resultados do Segundo Teste de Apreciação

Este questionário foi respondido por 17 pessoas e contém as seguintes questões:

1. Gostaste de ver as Animações?

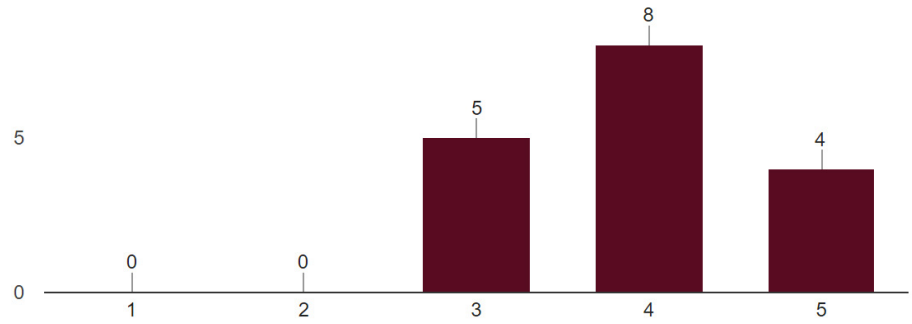

Fig. 104- Gráfico dos resultados da 1º pergunta

2. Ao controlar a protoganista sentiste algum desconforto?

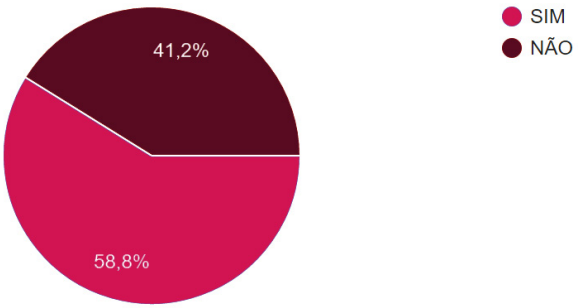

Fig. 105- Gráfico dos resultados da 2º pergunta

3. Em caso positivo, que desconforto?

- "Nada de grave, animações são um pouco lentas. E como vi que tens uma grande variedade nos ataques, mas parece ser algo aleatório é um pouco desperdício, era bom poder usar combos de teclas (ou outras teclas mesmo) para usar os diferentes ataques para se poder ver as animações destes."

- "Outro ponto picuinhas é a transição entre as animações, não é algo fluido para uma personagens tão detalhadas."

- "os inputs (salto e ataque) ocasionalmente não funcionaram, "clipping" de frames"

- O facto de ela ser muito lenta a andar de um lado para o outro. Principalmente por querer se tentar evitar os inimigos uma vez ou outro. Se fosse noutro contexto que não o combate, provavelmente não teria este pequeno desconforto. "

- "Ao chegar aos cantos da zona do jogo, também reparei que a personagem começava a deformar-se, causando assim um pequeno desconforto em vê-la com essa deformação, e indicando no início que o feedback que deveria impor nela seria de voltar ao centro da tela da zona do jogo."

- "A velocidade de movimento é muito lenta, e a string de ataques tem uma janela de tempo tão pequena entre cada ataque que é difícil de executar por inteiro como suposto. O som de ataque tocar de cada vez que se carrega na tecla, independentemente da animação já estar em curso ou não, não ajuda."

- "Complicado de fazer os wombo combos, dificil de saber para onde apontar"

- "A personagem desloca-se de forma lenta, impedindo de se desviar de ataques a tempo"

- "Problema com os combos do ataque"

- "input delay"

- "A tecla de ataque nem sempre respondia.
Aguentar algum tempo para andar e depois largar fazia a personagem continuar por mais um pouco a andar."

- "os movimentos deviam fluir mais suavemente"

4. As animações entre as personagens estavam claras?

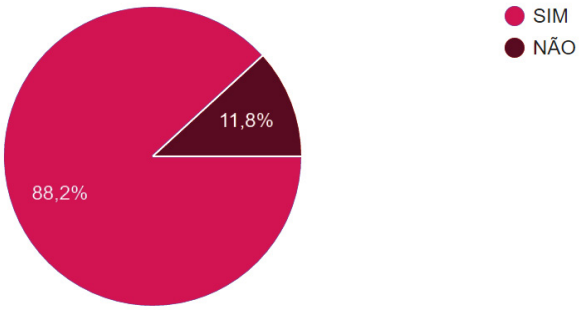

Fig. 106- Gráfico dos resultados da 4º pergunta

5. Em caso negativo, porquê?

- "Falta alguma forma de feedback adicional no inimigo para tornar os hits mais claros"

- "Reparei que havia diferentes formas de "atacar", mas nao sei como o fiz"

6. Consideras que o cenário tem profundidade? Justifique.

Houve 15 pessoas que responderam sim:

- "Sim, pelas diferentes posições das personagens e das arvores."

- "tem profundidade no sentido de existir claramente vários planos, no entanto uma melhor resolução nas texturas do mesmo poderia melhorar a experiência"

- "Considero sim, visto que são apresentadas diferentes "Layers" ou camadas de fundo, permitindo aumentar assim a sensação de profundidade, e intensificar o conceito de 3D. O facto de pudermos mover na vertical em adição ao movimento horizontal permite aumentar a sensação de profundidade, mesmo que todo o jogo apresentado vá de encontro com o género "Sidescrollers" como o Super Mario ou Sonic."

- "Sim, a perspectiva que tenho do cenário transmite profundidade e ao movimentar a personagem ao aproxima-lo e afastar-lo também transmite essa profundidade"

- "A pintura do cenário com as várias tonalidades e planos transmite a ideia de profundidade"

- "Sim. A utilização de parallax nos fundos ajuda a estabelecer uma noção de profundidade, juntamente com o facto das personagens de aproximarem e afastarem do ponto de vista."

- "Sim, mas existem bugs com as fontes luminosas"

- "o cenário podia ter mais profundidade"

- "Sim, as diversas camadas de vegetação dão a sensação de profundidade"

- "Sim, 3 lanes dao a profundidade suficiente para este tipo de projecto."

- "Sim, apesar necessitar de uma diferenciação mais acentuada a nível de cores à medida que a profundidade aumenta."

- "Sim. Bom uso de parallax scrolling ;)"

"Sim devido aos diferente tamanhos e sombras"

"Sim, porque tinha tinha diferentes tamanhos das arvores e cores."

"Sim, porque há medida que a paisagemm esta mais longe tambem escuresse"

Houve uma pessoa que respondeu não:

- "Não, pode ter profundidade a nível físico, mas o facto de estarem todos com o mesmo tom de cor não transmite nada essa ideia:
Exemplo: http://www.advancedphotoshop.co.uk/users/2253/thm1024/1333363189_blue_forest_small.jpg"
A mudança para o azul mais claro há medida que os objectos se encontram mais distantes. Penso que um fix rápido para isso é só mudar a saturação da textura.

Houve uma pessoa que estava incerto:

- "maybe?"

"7. Consideras que as personagens são de 3 dimensões? Justifique."

Houve 10 pessoas que responderam sim:

- "Sim, devido ao design que lhes foi proporcinado"

- "as personagens são "layered" e o cenário (e a profundidade do mesmo) ajuda na "ilusão" do tridimensional"

- "Considero sim. Denotasse que mesmo que o jogo seja apresentado com uma vista bidimensional e camadas de fundos com unicamente um lado, as personagens apresentam moldes semelhantes a uma personagem real, sendo que estes apresentam diferentes direções correspondentes às 3 dimensões. Denotasse mais a questão do 3D quando os inimigos e a personagem principal executa uma certa acção, como lançar o feitiço, em que a personagem principal mexe os braços em direcções diferentes ao invés de uma (como seria de esperar numa personagem 2D)."

- "Sim, o design e movimento das personagem tranmite-me a sensação de 3D."

- "Sim, mas nas rotações falta volumetria à personagem"

- "Os efeitos de luminusidade e o z-fighting entre os "componentes" das personagens (mesmo que não intencional) sugerem alguma profundidade, mas no máximo parecem múltiplos objectos 3D sobrepostos com a intenção de parecer formar sprites 2D."

- "Sim, porque Deus as fez assim."

- "Sim, porque podemos observar mais do que uma perpectiva e devido a sombras"

- "Sim, devido ao design pareciam 3D e nao 2D e as diferentes tonalidades."

- "Considero, dado que podemos ver diferentes perspectivas e nao so num plano. As diferentes cores/tonalidades e volume das roupas também ajuda."

Houve 6 pessoas que responderam não:

- "Não, apesar de se tentar usar a iluminação de modo a dar profundidade aos elementos das personagens, a animação é demasiado rígida e bidimensional."

- "Não. As personagem aperentavem ter duas dimensoes num cenario com profundidade"

- "Não. As personagem aperentavem ter duas dimensoes num cenario com profundidade"

- "Não, o desenho e pintura das personagens não dão a sensação das 3 dimensões. Talvez aumentar o sombreado e as variações das tonalidades ajudassem a transmitir essa ideia"

- "nope, feels 2D"

- "Consideraria as personagens como sendo 2D e não 3D devido aos sprites e à forma como foram desenhadas apesar da existência de profundidade."

- "Não. As personagens parecem ser apenas de 2 dimensões mas capazes de se mover num cenário devido à perspetiva isométrica."

Houve 1 pessoa que estava incerto:

- "Complicado. O facto da luz se movimentar pelo corpo das personagens durante o seu movimento parece que são 3D, por outro lado, esteticamente, parecem ser completamente 2D."

6.3.1. Análise do Segundo Teste

Este novo teste de apreciação foi um sucesso, pois obtivemos comentários melhor justificados e em geral os jogadores ofereceram comentários positivos, mas alguns dos negativos originaram de *bugs*. Tal como a protagonista conseguir sair do cenário e reencontraram instâncias em que a constituição das personagens ficam instáveis, principalmente nas extremidades do cenário, devido à breve mudança de perspectiva da câmera, além que estas ficam decompostas devido ao grande espaço entre *sprites*.

Tendo em conta às respostas da segunda pergunta, os jogadores sentem-se frustados com a lentidão da Witch, os combos eram de tal modo difíceis de efetuar que alguns dos jogadores nem se aperceberam do sistemas de combos e pensavam que eram aleatórios.

Os jogadores comentaram que em certas instancias não conseguiam atacar, pelo que deduzi que nos momentos em que a Witch não consegue atacar, porque esta já se encontra no meio da animação de um ataque, os jogadores continuam a premir da tecla "Q", que automaticamente emite o som de ataque, induzindo o jogador em erro.

Na quarta pergunta, foi indicado que gostariam de um sinal que confirma que feriram os bandidos

A sexta pergunta revelou o sucesso da liberdade do movimento vertical das personagens e do uso de Multiplane Camera Effect, também designado por Parallax Scrolling, com o objectivo de criar a ilusão de profundidade. Mas o exemplo do comentário negativo inspirou-me a melhorar o plano de fundo do cenário.

Na sétima questão, obtive várias opiniões mistas sobre o volume das personagens. Aos quais se resume nos seguintes tópicos:

- Havia pessoas que consideraram que a anatomia das personagens os tornavam 3D ou denunciava-as como 2D. Apesar de ter implementado o sistema de iluminação para criar a ilusão de volume do cenário e personagens. Contudo os jogadores aperceberam a diferença de tonalidades das personagens e como estas reagiam à luz, sendo índices que a técnica de Iluminação num jogo 2D é plausível com o uso de melhores *normal maps*.

- Tal como alguns se encontravam satisfeitos com os movimentos de três dimensões, outros achavam que estes necessitavam ser mais fluídas e que frequentemente as personagens ainda se encontravam restritos à movimentos de duas dimensões. Apesar de saber que estes problemas podem ser corrigidos com a adição de mais *sprites*, o *rig* atual não é capaz de suportar uma grande variedade de *sprites* sem provocar um grande custo de processamento ao computador.

6.3.2. Segundas Modificações ao Jogo

Os *bugs* foram corrigidos, com a reposição e reforçamento das barreiras invisíveis, a Witch já não consegue sair do cenário. Após uma investigação, identifiquei que o uso da habilidade do *shader* em desvanecer as extremidades nos *sprites*, mencionado na página 65, causava o Unity dificuldade em as renderizar corretamente. Portanto com essa habilidade desativada, não houve mais instabilidades.

Fig. 107- Imagem do das barreiras invisiveis

A respeito do conforto do uso da protagonista, aumentei a sua velocidade e a dos bandidos. Encontrei um novo método para construir um melhor sistema de combos a partir de *input buffering*, basicamente utiliza *queue* para registar temporariamente os comandos do jogador, ao qual procura por padrões que correspondem a combos. Mas no momento não se encontra disponível porque necessito tempo para o estudar e implementar.

Alterei o código de modo a que o som de ataque da Witch não é ativado pela tecla "Q", só ocorre na animação. Deste processo, decide implementar os seguintes sons:

- Music de tambores Porque intensifica o combate, por transmitindo uma atmosfera de energia e agressividade.
  Limetoe. (2016, 11 de Abril). Tribe Drum Loop. Em *Freesound.* Recuperado em https://www.freesound.org/people/limetoe/sounds/342465/

- Foley do bandido magoado, que é o sinal que o jogador o feriu.
  Ryanconway. (2013, 28 de Março). Male Grunts. Em *Freesound.* Recuperado em https://www.freesound.org/people/ryanconway/sounds/182304/

Com o objectivo de melhorar o plano de fundo do cenário, decidi acrescentar uma *layer*, designada por Plano de fundo 2.

Fig. 108- Imagem do *Plano de fundo 2* do cenário

Acrescentei nevoeiro que ao altera a tonalidade entre os planos da Multiplane Camera Effect, reforçando a ilusão de profundidade. Implementei efeitos de imagem do Unity:

- Color Correction (Curves, Saturation) que torna as cores do jogo menos pálidas.

- Bloom que simula incandescência a volta dos objetos, que neste caso melhora as chamas 2D e simula raios solares a penetrar a floresta, intensificando a ilusão de profundidade. O que me obrigou a acrescentar uma terceira câmera somente responsável em renderizar o Bloom.

Fig. 109- Exemplos gráficos Do efeito Bloom nas chamas 2D

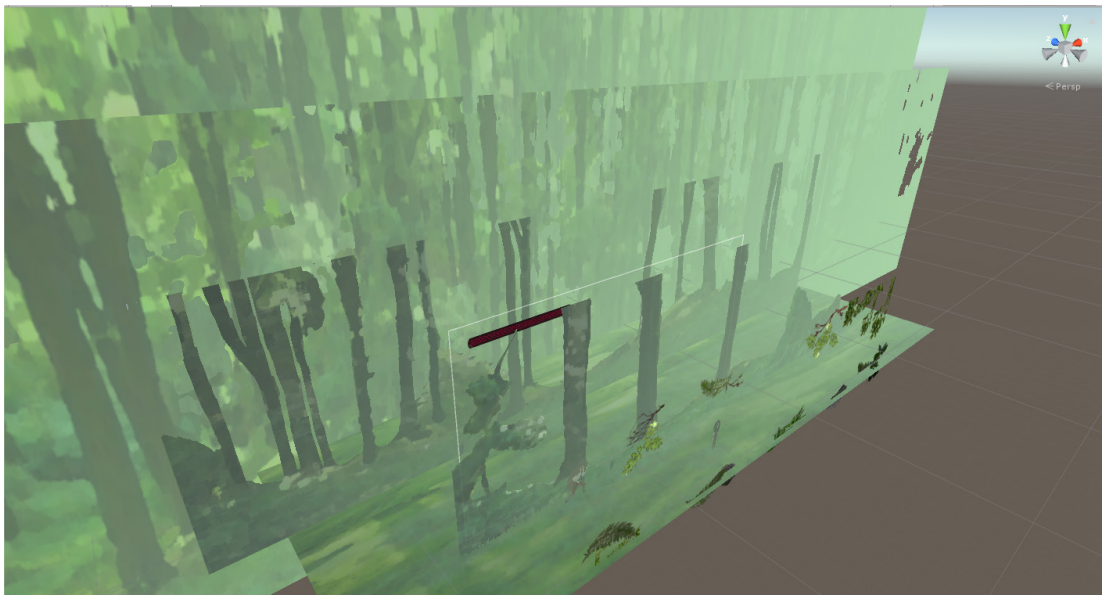

Fig. 110- Resultado final do cenário com as modificações

Fig. 111- Resultado final do cenário com as modificações do ponto de vista do jogador

7. Conclusão

7.1. Desafios na Produção

Durante o processo da dissertação deparei-me me com alguns problemas que dificultaram o seu progresso, nomeadamente a falta de experiência e não ter um conhecimento sólido na área de animação 3D e de pintura digital. Foi necessário aprender a escrever código C# e o código de Unity, que constitui o maior desafio, pois exigiu a aprender como este comunica com a estrutura do editor do Unity e o comportamento das suas várias funções e classes. As dificuldades anteriormente elencadas e as inerentes à construção do jogo, das quais destaco a dificuldade em me concentrar numa só tarefa/área por existir uma ligação natural entre as diversas áreas do jogo, tais como design, desenho, rig, animação e principalmente programação, em que o desenvolvimento de uma condicionava o processo das outras. Assim, os progressos feitos não se traduziam em resultados imediatos e foi necessário aprender a trabalhar de uma forma atenta e globalizante o que se revelou ser um desafio para cumprir com a calendarização. Face ao exposto, apenas foi possível implementar os testes de apreciação quando a estética e a animação do jogo estava completa, pois tendo em conta o âmbito desta dissertação, seria irrelevante aplicá-los com um protótipo de baixa fidelidade.

7.2. Sobre as Técnicas Ensaiadas.

O objetivo desta dissertação era combinar as técnicas simplificadas e refinadas 2D com as técnicas dinâmicas 3D, de modo a superar a estética plana, e criar novos sabores na animação no contexto de jogos, o que conduziu ao longo da pesquisa de animação 3D e 2D, e outros jogos, à coleção de técnicas relevantes ao tema da dissertação. Criou-se um nível do jogo Lil´Witch através da *selecção* e aplicação das técnicas animação modular, iluminação 3D em 2D e MultiPlane Camera effect. A partir deste estudo em conjunto com os resultados dos testes de apreciação, conclui que, MultiPlane Camera effect foi um sucesso em criar a ilusão de profundidade, e, apesar dos indícios positivos do sucesso de animação modular e iluminação 3D em 2D, os seus resultados podem ser melhorados substancialmente com experiencia e tentativa e erro, revelando novos potenciais em jogos, e, principalmente em animação. Estas técnicas permitem modificar o processo de animação criando alternativas o que o torna mais adaptável às necessidades e qualidades dos animadores e dos projetos, e, em consequência, possibilita a aplicação de técnicas que eram antigamente incompatíveis ou ineficientes, ou seja, em animação de *sprites*, seria necessário criar uma *normal map* ou *oclusion map* diferente para cada sprite, ao contrário da animação modular.

7.3. Possiveis Evoluções do Artefacto.

Forem colecionadas várias técnicas para criar um nível num jogo de 2D com 3D, contudo este ainda têm espaço para melhorias. Tais como:

- Com o objectivo de tornar o jogo mais interessante, teria que ser implementado o sistema de runas, corrupção da natureza e a habilidade de salto.

- Melhorar o código do sistema de combate, especificamente tornar o sistema de combos mais acessível a partir do uso de *input buffering*.

- Melhorar o *rig* de animação modular, de modo a conseguir suportar animações mais complexas sem adicionar muita carga de trabalho ao processador, conseguir ser sujeita a todas as ferramentas e funções do Unity, e, habilitar modificações em animações e personagens sem ser necessário repetir completamente o seu processo de produção.

- Aperfeiçoar os *normal maps* de modo a que a reação do cenário e o corpo das personagens com a iluminação, seja mais fiel à realidade.

- Reconstruir as personagens de modo a serem *double layered objects,* ou seja ambos lados dos *sprites* reagem corretamente à iluminação

Glossário

gameplay - É a interação entre os jogadores e o jogo, segundo as regras do jogo.
test users - São usuários que testem um produto, cuja experiencia é recolhida e analisada.
feedback - É uma resposta, consequência numa ação do usuário.
modelos proxy - Modelos de baixa resolução que representa os modelos finais.
software - É o termo geral para os diversos programas usados para operar o computador.
hardware- São as componentes físicas, eletrônicas que constituem o computador.
reentrancias - Angulo ou curva para dentro, por exemplo uma cavidade.
mesh - Coleção de vértices, arestas e faces, que definem a forma num modelo em computação 3D.
Key frame - É um fotograma que indica na linha de tempo o inicio ou final de um movimento, transição
Key pose- fotogramas que contem as poses das personagens, que definem a ação da personagem.
Processador - É uma componente de hardware do computador, responsável pelos cálculos, processamento de dados e execução de tarefas. Portanto a velocidade de processamento do computador depende do seu processador.
Placa gráfica - É a componente de hardware do computador responsável pela construção e apresentação de elementos gráficos, por exemplo a da imagem do ecrã.
Disco rigido - É uma componente de hardware do computador, responsável pelo armazenamento de dados, ou seja a memória do computador.
Bugs - É um erro no funcionamento comum do software ou hardware do computador.
skydome- método para simular o fundo de paisagens, por exemplo a imagem de edificios, montanhas , ceu são projetadas do interior num hemisfério ou esfera, portanto com a camera do jogo dentro do skydome, cria a ilusão do fundo de paisagem estar a uma grande distancia.
npc- non playable chracter, personagens que não são controladas pelo jogador.
plugin- um componente de software que adiciona uma função à um programa preexistente.
fps- Também designado por *frame rate* são o número de fotogramas visualizados por segundo.

Bibliografia

Chopine, A. (2011). *3D art essentials*. Taylor & Francis

Beane, A. (2012). *3D animation essentials. John Wiley & Sons*

Thomas, F. & Johnston O. (1981) *The illusion of life*. Italy: Walt Disney Productions

Bain, G. (1977) *Celtic art the methods of construction. London:*

Multiplane educator guide. Recuperado em http://www.waltdisney.org/calendar/technological-innovation-animation.

Webgrafia

Smith, B. (2015, 27 de Fevereiro). Former Disney Veteran Explains Why Big Studios Have Abandoned 2D Animation. Em *Rotoscopers.* Recuperado em http://www.rotoscopers.com/2015/02/27/former-disney-veteran-explains-why-big-studios-have-abandoned-2d-animation/

Thill, S. (2015, 1 de Fevereiro). Tomm Moore on 'Song of the Sea,' Reinventing 2D, and Dodging the Studio System. Em *Cartoon Brew.* Recuperado em http://www.cartoonbrew.com/award-season-focus/tomm-moore-on-song-of-the-sea-reinventing-2d-and-dodging-the-studio-system-107389.html

Kenny, C. (2014, 11 de Setembro). It's Time To Admit That 2-D Animation Does Not Need 'Saving'. Em *Inde Wire*. Recuperado em http://www.indiewire.com/2014/09/its-time-to-admit-that-2-d-animation-does-not-need-saving-123975/

Character Design.Em *Pixar*. Recuperado em http://pixar-animation.weebly.com/character-design.html

How NOT To Make Normal Maps From Photos Or Images. Em *KatsBits*. Recuperado em http://www.katsbits.com/tutorials/textures/how-not-to-make-normal-maps-from-photos-or-images.php

Oatley, C. The Making Of Paperman: The Idea, The Drawings, The Look. Em ChrisOatley. Recuperado em http://chrisoatley.com/making-of-paperman/

Kaganskiy, J. (2013, 5 de Março). Trying To Woo Animators, Disney Accidentally Invents "The Paperman Method". Em *Fast Company.* Recuperado em https://www.fastcompany.com/3006276/open-company/trying-woo-animators-disney-accidentally-invents-paperman-method

Ortiz, B. (2016, 16 de Julho). What to consider when deciding on 2D vs 3D for a game?. Em *Gamedev Stackexchange*. Recuperado em http://gamedev.stackexchange.com/questions/631/what-to-consider-when-deciding-on-2d-vs-3d-for-a-game

Riki, J.K. (2015, 27 de Junho). Why Should 2D Animation Be Abandoned? Em *Animator Island*. Recuperado em

http://www.animatorisland.com/why-2d-animation-should-be-abandoned-part-1/

Ikinema, J. (2014, 21 de Novembro). What are your biggest issues with 3D animation? Em *Unity Forum*. Recuperado em https://forum.unity3d.com/threads/what-are-your-biggest-issues-with-3d-animation.281449/

Moyes, G. (2014, 6 de Maio). Multidirectional character animation not practical? Em *Spriter Forums*. Recuperado em https://brashmonkey.com/forum/index.php?/topic/3308-multidirectional-character-animation-not-practical/

Chris 2 (2010, 12 de Junho). How many polygons is too much? Em *Unity Forum*. Recuperado em http://answers.unity3d.com/questions/19379/how-many-polygons-is-too-much.html

20 Different Types of Animation Techniques and Styles. Em *Webneel*. Recuperado em http://webneel.com/different-types-of-animation-styles

Massoudi, P. (2015, 11 de Agosto). The Challenge of Having Both Responsiveness and Naturalness in Game Animation. Em *Gamasutra*. Recuperado em http://www.gamasutra.com/blogs/PeymanMassoudi/20150811/250937/The_Challenge_of_Responsiveness_VS_Naturalness_in_Game_Animation.php

2.5D. Em *Wikipedia*. Recuperado em https://en.wikipedia.org/wiki/2.5D

Muzzoid (2013, Maio). Convincing 3d that looks like 2d.... Wow. Em *polycount*. Recuperado em http://polycount.com/discussion/121144/convincing-3d-that-looks-like-2d-wow

Nishikawa, Z. (2014, 7 de Dezembro) Nishikawa Yoshiji's "Examination Game Graphics" (1) "Secret of real-time 3D graphics only visible to animation" realized with "GUILTY GEAR Xrd -SIGN-". Em 4gamer. Recuperado em http://www.4gamer.net/games/216/G021678/20140703095/

Schreibt, S. (2013, 18. de Março). Homeworld 2 – Backgrounds Tech. Em *Simon Schreibt*. Recuperado em https://simonschreibt.de/gat/homeworld-2-backgrounds/

Schreibt, S. (2013, 27 de Setembro). World of Torch Siege – Blended Trunks. Em *Simon Schreibt*. Recuperado em https://simonschreibt.de/gat/world-of-torch-siege-blended-trunks/

Schreibt, S. (2013, 4 de Dezembro). Handmade Normal Maps. Em *Simon Schreibt*. Recuperado em https://simonschreibt.de/gat/handmade-normal-maps/

Lambie, R. (2015, 9 de Novembro). Song Of The Sea: how an animated treat was made. Em *Den of Geek*. Recuperado em http://www.denofgeek.com/movies/song-of-the-sea/36038/song-of-the-sea-how-an-animated-treat-was-made

Sheffield, B. (2009, 3 de Agosto). King of 2D: Vanillaware's George Kamitani. Em *Gamasutra*. Recuperado em http://www.gamasutra.com/view/feature/132486/king_of_2d_vanillawares_george_.php

Vader, D., Nguyen, M. & Nguyen, V. (2015, 2 de Outubro). Beneath the pixels: The art direction of Super Time Force. Em *Gamasutra*. Recuperado em http://www.gamasutra.com/view/news/255187/Beneath_the_pixels_The_art_direction_of_Super_Time_Force.php

(2015, 29 de Maio). 10 Years of Game Animation. Em *Game Anim.* Recuperado em http://www.gameanim.com/2015/05/29/10-years-of-game-animation/

*Fe*ssler, D. (2014, 19 de Fevereiro). Thoughts on Modular Animation. Em *Dan Fessler.* Recuperado em http://danfessler.com/blog/thoughts-on-modular-animation

Stoic, A. (2013, 23 de julho). Animation Process . Em *Banner Saga.* Recuperado em http://stoicstudio.com/animation-process/

Leon, D. (2015, 2 de julho). Next-Gen Cel Shading in Unity 5. Em *Gamasutra*. Recuperado em http://www.gamasutra.com/blogs/DavidLeon/20150702/247602/NextGen_Cel_Shading_in_Unity_5.php

Gauthier, J. & Boyer, A. (2015, 7 de Outubro). Art Design Deep Dive: The hand-drawn art and animation of Jotun. Em *Gamasutra*. Recuperado em http://www.gamasutra.com/view/news/255371/Art_Design_Deep_Dive_The_handdrawn_art_and_animation_of_Jotun.php

CARLIN, C. (2013, 7 de Junho). Lighting a 2D Game. Em *The Complete Swine*. Recuperado em http://www.wholehog-games.com/devblog/2013/06/07/lighting-in-a-2d-game/

(2011, 11 de Junho). Supergiant Games' Bastion. Em *CGsociety*. Recuperado em http://www.cgsociety.org/index.php/CGSFeatures/CGSFeatureSpecial/supergiant_games_bastion

Conzeit (2014, 5 de Fevereiro). Puppet/Modular Animation: how when and why?. Em *Pixelation.* Recuperado em http://pixelation.org/index.php?topic=16284.0

Perton , B.(2014, 5 de Dezembro) I Want to See Less Realistic Videogame Graphics!. Em *Gamasutra.* Recuperado em http://www.gamasutra.com/blogs/BrandonPerton/20140512/217528/I_Want_to_See_Less_Realistic_Videogame_Graphics.php

Webster, A.(2014, 25 de Novembro) The stunning 'Lumino City' is a video game made of paper instead of polygons. Em *The Verge.* Recuperado em http://www.theverge.com/2014/11/25/7275969/lumino-city-paper-video-game

(2014, 2 de Dezembro). *From Paper to Play: How they made Lumino City - Teaser*. [Video]. Recuperado em https://www.youtube.com/watch?v=JO6t6H19CUk

Paperman and the Future of 2D Animation. [Video]. Recuperado em https://www.youtube.com/watch?v=TZJLtujW6FY

(2015, 21 de Maio). *GuiltyGearXrd's Art Style : The X Factor Between 2D and 3D*. [Video]. GCD. Recuperado em https://www.youtube.com/watch?v=yhGjCzxJV3E

(2015, 7 de Julho). *The Animation Process Of Ori & The Blind Forest.* [Video]. GCD. Recuperado em https://www.youtube.com/watch?v=m8lOwrWNbEY

(2016, 26 de Janeiro). *Making Fluid and Powerful Animations For Skullgirls.* [Video]. GCD. Recuperado em https://www.youtube.com/watch?v=Mw0h9WmBlsw

(1957, 13 de Fevereiro). *Walt Disney's MultiPlane Camera.* [Video]. Disney. Recuperado em https://www.youtube.com/watch?v=YdHTlUGN1zw

(2013, 31 de Julho). *How Rayman Legends Is Made!.* [Video]. Recuperado em https://www.youtube.com/watch?t=1&v=y-chi097uV4

(2014, 18 de Dezembro). *Design Club - The Animation of Punch Out!! - How the Wii Remake Scored a KO* [Video]. Extra Credits. Recuperado em https://www.youtube.com/watch?t=1&v=y-chi097uV4

(2016, 15 de Fevereiro). *The Animation of Jak & Daxter - #1 - SQUASH & STRETCH* [Video]. Extra Play. Recuperado em https://www.youtube.com/watch?v=BbP6Jsh8M6Y

(2016, 8 de Março). *Animation Q&A - #1 - Dan Answers Your Animation Questions!* [Video]. Extra Play. Recuperado em https://www.youtube.com/watch?v=0pknmynv3NU

(2016, 25 de Janeiro). *The Animation of Shadow of the Colossus - #1 - TIMING* [Video]. Extra Play. Recuperado em https://www.youtube.com/watch?v=2_bg5D8A2QU

(2016, 26 de Janeiro). *The Animation of Shadow of the Colossus - #2 - Inverse Kinematics & Colossus Climbing* [Video]. Extra Play. Recuperado em https://www.youtube.com/watch?v=LVHM4spXuOM

(2016, 27 de Janeiro). *The Animation of Shadow of the Colossus - #3 - The Camera* [Video]. Extra Play. Recuperado em https://www.youtube.com/watch?v=bdBMA-3Crng

(2015, 16 de Fevereiro). *2D Prototyping in Unity - Tutorial - Platformer* [Video]. CasanisPlays. Recuperado em https://www.youtube.com/watch?v=gC0N6ETulv0&list=PL2cNFQAw_ndyKRiobQ2WqVBBBSbAYBobf

(2015, 20 de Dezembro). *Unity 5 Tutorial 3D Beat 'Em Up* [Video]. CasanisPlays. Recuperado em https://www.youtube.com/watch?v=6OMMA0s2NLY&t=16s