

## **Keepers of Intheris:**

Uma produção audiovisual de animação em 3D

Modelação 3D, captura de movimento e efeitos visuais

PROJETO DE MESTRADO

**Erika Carmen Fernandes da Silva**

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

*A Nossa Universidade*

[www.uma.pt](http://www.uma.pt)

Setembro | 2017



## **Keepers of Intheris**

Uma produção audiovisual de animação em 3D.  
Modelação 3D, captura de movimento e efeitos visuais

PROJETO DE MESTRADO

**Erika Carmen Fernandes da Silva**

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR

Sergi Bermúdez i Badia

CO-ORIENTADOR

Eduardo Leopoldo Fermé

Pau Pascual Galbis







# ABSTRATO

A indústria cinematográfica habituou as pessoas a grandes produções 3D, com várias técnicas inovadoras para a captura de movimentos juntamente com a utilização de vários *softwares* pagos, que envolvem grandes investimentos a nível financeiro. Atualmente, no mercado, existem alguns sistemas de captura de movimentos a preços acessíveis como também pacotes de software, alguns gratuitos e outros com preços mais acessíveis, que em conjunto são capazes de criar soluções de vídeo 3D com captura de movimentos a um valor bastante aliciante. Este projeto consistiu na utilização de uma solução mais económica para a produção de um filme de animação 3D, mais propriamente um *trailer* para o jogo “Keepers of Intheris”. Seguindo uma *pipeline* de produção, atravessámos várias fases. Na fase inicial, denominada de pré-produção, foi idealizado todo o projeto e o enredo a adotar, seguindo pela criação do *storyboard* e acabando no design dos conceitos. Na fase de produção, todos os conceitos foram concretizados por modelação e texturização. Ainda nesta fase, as personagens, além da modelação e texturização, passaram pelas fases de *rigging* e animação, onde depois foram inseridas nos cenários, tendo sido realizados também os efeitos visuais. Na ultima fase, a pós-produção do projeto, foram realizados os vários efeitos visuais 2D e edição de vídeo e de som. No fim da produção deste projeto obtivemos como resultado um *trailer* para o jogo referido. Este projeto segue uma *pipeline* que poderá ser útil para qualquer pessoa que queira realizar qualquer tipo de filme ou vídeo 3D aliado a um baixo custo quando comparado com as soluções existentes na indústria cinematográfica.

# ABSTRACT

When it comes to the cinematographic industry, people have come to expect large 3D productions with ground breaking techniques for motion capture and very high costs from the required software. Currently, the consumer segment has access to some motion capture systems as well as software packages for 3D motion capture at very competitive prices. On this project, this equipment and software are used to produce a 3D animation video, specifically, a trailer for the game “Keepers of Intheris”. Following the standard production pipeline, there are three main stages in this project. In pre-production, the entire project was idealized, planned and the story was written. This was followed by the creation of the storyboard and then, the concepts were designed. During the production stage, all concepts were modelled and textured. During this stage, characters were also rigged and animated before being placed on the set/scenes after which, visual effects were added. On the last stage of the pipeline, post-production, 2D visual effects were added, the video was edited and sound was added to the video. The result of this project’s production was the trailer for the game Keepers of Intheris. This project follows a pipeline that is useful for anyone who wants to produce any type of video or 3D animation at a low cost when compared to the current solutions of the cinematographic industry



# PALAVRAS-CHAVE

Animação 3D

Kinect

Captura de movimento

# KEYWORDS

3D Animation

Kinect

Motion Capture



# AGRADECIMENTOS

Quero começar por agradecer aos meus pais, que são muito importantes para mim, que sempre acreditaram em mim e me concederam todo o seu apoio durante todo o meu percurso académico e que com muito esforço me ajudaram a concretizar esta fase da minha vida.

Agradeço ao meu namorado Filipe Abreu, por todo o seu apoio e dedicação, por todas as horas de sono perdidas e toda a sua persistência. Pela sua vontade de ajudar e por estar presente nos piores e melhores momentos.

Um agradecimento especial ao meu orientador Sergi Bermúdez i Badia pela sua disponibilidade e toda ajuda disponibilizada tendo sido um dos pilares deste projeto, sem esquecer os meus co-orientadores, Eduardo Leopoldo Fermé e Pau Pascual Galbis por todo o apoio, dicas, tempo, compreensão e dedicação.

Aos meus amigos, Tatiana Vieira, Yuri Almeida e João Serina pela amizade incondicional, cumplicidade, pelas gargalhadas e por todo o apoio e paciência que tiveram durante a realização deste projeto.

Não querendo esquecer ninguém, agradeço a todos os meus amigos, professores e familiares que me apoiaram e ajudaram direta ou indiretamente em todo o meu percurso académico.



# ÍNDICE

1. Introdução.....	1
1.1. Motivação e objetivos.....	1
2. Estado da arte.....	5
2.1. Historia da animação 3D.....	5
2.2. Processo de produção da animação 3D.....	6
2.2.1. Pré-produção.....	6
Ideia / História.....	7
Guião.....	7
Storyboard.....	7
Animatic/Pre-visualization.....	8
Design.....	8
2.2.2. Produção.....	9
Layout.....	9
Investigação e desenvolvimento.....	9
Modelação.....	10
Texturização.....	10
Rigging.....	11
Animação.....	11
Efeitos Visuais 3D.....	12
Luzes.....	12
2.2.3. Pos produção.....	12
Composição.....	13
Edição de Som e Vídeo.....	13
Correção de Cor.....	13
Vídeo Final.....	13
2.3. Ferramentas.....	13
2.3.1. Software.....	14
2.3.1.1. Software de modelação – 3d max/maya/Zbrush.....	14
2.3.1.2. Software de animação e mocap.....	15
2.3.1.3. Software de texturização – Photoshop – Substance painter.....	17
2.3.1.4. Software de criação de personagens – fuse – Make human.....	19
2.3.2. Captura de movimento (MoCap).....	21

2.3.2.1.	Evolução da Captura de Movimentos antes dos computadores .....	21
2.3.2.2.	Evolução da captura de movimento para animações de personagens de computador .....	23
2.3.2.3.	Tipos de captura de movimentos .....	26
	Captura magnética.....	27
	Captura Mecânica .....	27
	Captura ótica.....	27
2.3.2.4.	Kinect.....	28
	Evolução da Kinect .....	29
	Kinect 1 vs. Kinect 2? .....	29
	Esqueletos e corpos v1 vs. v2 .....	30
	Características de rosto .....	31
	Usos da Kinect na indústria.....	31
2.4.	Conclusão.....	35
3.	Metodologia .....	39
3.1.	Pré-produção .....	39
3.1.1.	História .....	39
3.1.2.	Storyboard.....	40
3.1.3.	Design .....	42
3.2.	Produção.....	45
3.2.1.	Criação do 3d.....	45
3.2.1.1.	Criação do mundo .....	45
	Criação da base .....	45
	Esculpir.....	47
	Re-topologia.....	51
	Unwrap UVW .....	51
	Projeção .....	52
	Texturização.....	53
3.2.1.2.	Criação das personagens.....	56
	Provas para escolha do método de produção de uma personagem.....	57
	Criação das personagem – Modelação .....	59
	Rigging/Skinning .....	61
3.2.1.3.	Montagem das cenas .....	64
	Divisão das cenas, escolha do software e importação dos modelos.....	64

Efeitos visuais.....	65
Câmaras .....	79
Luzes .....	80
3.2.2. Animação.....	82
3.2.2.1. Captura da animação.....	83
Animação corporal.....	83
Animação facial.....	86
3.2.2.2. Criação de animações.....	89
3ds Max.....	89
Maya .....	93
3.2.3. Render .....	93
3.3. Pós-produção .....	94
4. Implementação .....	97
5. Conclusão .....	105
6. Referencias .....	107
7. Anexos .....	117
7.1. Tabela com as características da Pen tablet.....	117
7.2. Storyboard .....	118
7.3. Conceito dos mundos .....	134
7.4. Opções de render para realizar a projeção .....	137
7.5. Tabelas com os passos de todos os Efeitos visuais.....	138
7.6. Tabela de horas de renders .....	140
7.7. Tabela de renders .....	141



# ÍNDICE DE IMAGENS

Imagem 1 - Filme Toy Story (1995) [6].	5
Imagem 2 - Demonstração gráfica do pipeline de uma animação [8].	6
Imagem 3 - Frame do storyboard do filme Toy Story [143].	7
Imagem 4 - Animatic do filme Toy Story [144].	8
Imagem 5 - Ideias do visual que Woody personagem do Toy Story [13].	8
Imagem 6 - Pipeline de produção [8].	9
Imagem 7 - Layout da personagem Buzz do filme Toy Story [145].	9
Imagem 8 - Modelação de personagens do filme Toy Story; (A) Woody; (B) Buzz Lightyear [14]	10
Imagem 9 - Texturização de cenário e personagens do filme Tpy Story [15].	10
Imagem 10 – (A) Personagens do filme Toy Story com rigging e controladores [17]; (B) Rigging da personagem Buzz [14].	11
Imagem 11 - Animação de cena do Toy Story [146].	11
Imagem 12 - Efeitos visuais adicionados no filme Toy Story [15].	12
Imagem 13 - Luzes adicionas a cena do Toy Story [15].	12
Imagem 14 - Pen tablet HUION H610, utilizado no projeto.	15
Imagem 15 - Rostos com morphs/blendshapes de uma personagem [147].	17
Imagem 16 - Substance Painter com as duas visualizações 3D e 2D [52].	18
Imagem 17 – Baking Substance Painter[53].	19
Imagem 18 - Janela com os vários materiais do Substance Painter.	19
Imagem 19 - Fluxo de trabalho da criação de uma personagem oferecido pelo Mixamo [64].	20
Imagem 20 – Evolução da captura de movimento. 1832 – Phenakistoscope; 1879 – Prova de Eadweard Muybridge (Equipamento Zoopraxiscópio); 1882 – Etienne – Jules Marey (Fatos MoCap); 1915 – Rostocopia.	21
Imagem 21 – (A) Phenakistoscope [70]; (B) Prova de Eadweard Muybrige em 1879 [72]; (C) – Zoopraxiscópio [73].	22
Imagem 22 – (A) Fato com marcas laterais e na cabeça, Etienne-Jules Marey , 1884 [72]; (B) Captura de movimento de andamento com o fato de Etienne-Jules Marey [74].	22
Imagem 23 – (A) Rotoscope [75]; (B) Max Fleischer's e o palhaço Koko [75]; (C) A utilização do rotoscope no palhaço Koko [75].	23
Imagem 24 – (A) Um dos primeiros equipamentos fotográficos que usa uma luz intermitente com uma câmara Kodak de 35mm [77]; (B) Exemplo da captura de movimento do serviço do tenista Gussie Moran realizado por Edgerton, 1952 [77].	23
Imagem 25 - Pontos processados computacionalmente com ajuda de LEDs durante uma captura de movimento e o Stick Figure criada computacionalmente com recurso aos dados analisados [148] ...	24

Imagem 26 - (A) Robô Brilliance; (B) Atriz por detrás do robô.....	24
Imagem 27 - (A) Waldo C. Graphic [19]; (B) Equipamento de captura de movimento [86] .....	25
Imagem 28 - Mike the Talking Head [88]. .....	25
Imagem 29 - (A) Ator com o capacete de captura de movimentos a representar a personagem Mário; (B) Output final com a personagem Mário [89]. .....	26
Imagem 30 - Captura de movimento a imitar um jogador de futebol [91]. .....	26
Imagem 31 - Câmara Vicon equipada com LEDs para captura ótica de movimentos [72] .....	28
Imagem 32 - Sensores da Kinect 2 da Windows [99]. .....	29
Imagem 33 - (A) Esqueleto detetado pela Kinect1; (B) Esqueleto detetado pela Kinect2.....	31
Imagem 34 - Utilização da Kinect na medicina [149]. .....	32
Imagem 35 - Melhoramento do raio-X da mão. (A)Raio-X; (B) Kinect.[150].....	32
Imagem 36 - Reabilitação utilizando a Kinect [110] .....	32
Imagem 37 – (A) Interface do projeto que ajuda ao medico a saber o estado do paciente; (B) Interface do projeto que mede as habilidades de toques do paciente através da Kinect [112]. .....	33
Imagem 38 - Projeto Kinect Sign Language Translator [113]. .....	33
Imagem 39 - Jogo educativo que utiliza a Kinect. In Body and Brain Connection [115]. .....	34
Imagem 40 - Robotics Developer Studio [116]. .....	34
Imagem 41 - – Interactive Solution [117]. .....	34
Imagem 42 - Virtual Style Sense [119]. .....	35
Imagem 43 - Path Tracker [151]. .....	35
Imagem 44 - Pipeline de desenvolvimento .....	39
Imagem 45 - Pipeline de pré-produção.....	39
Imagem 46 – Miniatura do storyboard da cidade. (A) Campo onde é elaborado o desenho da cena; (B) Informação sobre a cena. ....	41
Imagem 47 – Miniaturas do storyboard no instante em que o templo de ilumina e se ativas os portais. (A) Abertura dos portais; (B) Guerreiro sai do portal. ....	41
Imagem 48 – Inspiração e desenhos do mundo dos curandeiros.....	43
Imagem 49 -Inspirações. (A) Poções mágicas dos curandeiros; (B) Guerreiros curandeiros.....	44
Imagem 50 - Pipeline de criação 3D.....	45
Imagem 51 - Pipeline da criação do cenário e os seus acessórios .....	45
Imagem 52 - (A) Base dos portais modelado no 3ds Max; (B) Base do símbolo do portal dos curandeiros, modelada a partir de uma imagem no 3ds Max .....	46
Imagem 53 – Opção de exportação para o Zbrush. ....	47
Imagem 54 –(A) Detalhe de pedra esculpido no portal; (B) Pincéis com algumas texturas de ranhuras de pedras que podem ser adicionados ao modelo. ....	48
Imagem 55-Criação da corda do Zbrush .....	48

Imagem 56 - Portal dos curandeiros e opções que foram utilizadas. ....	49
Imagem 57 – Esculpir a navalha utilizando a mascara. ....	49
Imagem 58 – Esculpir um modelo com uma imagem alpha e utilizando o Backface mask. ....	50
Imagem 59 – (A) Modelo da poção do portal dos curandeiros esculpido: (B) Utilização da técnica SliceCurve no bastão. ....	50
Imagem 60 - Re-topologia do bloco do portal. ....	51
Imagem 61 - Unwrap de um dos blocos que fazem parte do portal. ....	52
Imagem 62 – Projeção de um bloco do portal. ....	52
Imagem 63 - Comparação de o número de polígonos e vértices entre um bloco do portal low poly com um high poly. ....	53
Imagem 64 – (A) Gizmo cilíndrico das lianas; (B) Textura adicionada através do modificador UVW Map nas lianas e nas flores. ....	54
Imagem 65 – (A) Textura de relva; (B) Textura de terra; (C) Textura final.....	54
Imagem 66 –(A) Textura Diffuse Map; (B) Textura Lighting Map; (C) Textura Normal Map; (D) Textura final.....	55
Imagem 67 – (A) Adicionar texturas a um material no 3ds Max; (B) Portal dos curandeiros com a texturização final. ....	55
Imagem 68 – (A) Janelas 3D e 2D do material atribuído ao modelo; (B) Janela 3D e 2D com o mapa Normal, onde podemos ver os relevos adicionados ao modelo.....	56
Imagem 69 – (A) Parte da interface onde podemos escolher qual é o modelo que queremos texturizar; (B) Interface em 3D e 2D do modelo escolhido; (C) As varias camadas de pintura que estão aplicadas ao modelo.....	56
Imagem 70 - Pipeline de criação de uma personagem .....	57
Imagem 71 - Modelo escolhido para elaborar o rigging. ....	57
Imagem 72 – (A) Duplicação de rostos para realizarmos varias expressões; (B) Depois da junção dos rostos obtemos o menu Blend Shapes como todas as expressões que podemos realizar; (C) Controladores do pescoço e aos olhos e ossos adicionados a cabeça.....	58
Imagem 73 –(A) Expressões faciais incorporadas no Makehuman; (B) Opções de esqueletos.....	58
Imagem 74 – (A) Característica para criação personagens; (B) Opções de roupa do Makehuman.....	59
Imagem 75 - Interface do Fuse com algumas opções de cabeças e de vestuário.....	60
Imagem 76 – Opções de modificação do modelo. ....	60
Imagem 77 – Tipos de texturas existentes no Fuse .....	61
Imagem 78 – Representação da topologia no rosto e no corpo. ....	61
Imagem 79 – Auto-rigger do Mixamo, onde escolhemos se queremos Blendshapes e a quantidade de ossos. ....	62
Imagem 80 – Skinning utilizando o Weight Tool.....	63
Imagem 81 -Pipeline do procedimento que realizamos para a montagem das cenas. ....	64

Imagem 82 – Alguns dos cenários divididos (A) Espaço; (B) Vista total da vila e floresta; (C) Floresta; (D) Vila. ....	65
Imagem 83 - Galáxia azul.....	66
Imagem 84 – Elaboração das estrelas através do pincel space.mel oferecido pelo Maya. ....	66
Imagem 85 - Galáxias. (A)Imagem colorida da galáxia; (B) Imagem da galáxia preto e branco; (C) Sistema de partículas. ....	67
Imagem 86 – Expressão que dá diferentes tempos de vida a cada partícula. ....	68
Imagem 87 – Janela Hypershade; (A) Opção Particle Cloud; (B) Input and Output connection.....	68
Imagem 88 - Particle Cloud .....	69
Imagem 89 - Create render node opção Ramp .....	69
Imagem 90 – (A) Rasto da nave criado através de um sistema de partículas; (B) Render final.....	69
Imagem 91 – (A) Container; (B) Janela Graph Editor onde mostramos a inclinação provocada na densidade e no aquecimento através da opção Step tangentes. ....	70
Imagem 92 – (A) Antes de calcular os valores da temperatura e combustível; (B) depois de calcular as cores da temperatura e combustível. ....	71
Imagem 93 – Processo intermédio da criação da explosão. ....	71
Imagem 94 - Sequencia da explosão final. ....	72
Imagem 95 – Criação do sol com a textura preta e as definições do material da textura.....	72
Imagem 96 – (A) Opções da textura fractal; (B) Render do sol.....	73
Imagem 97 – (A) Formula para animar a superfície do sol; (B) Sol com efeito de brilho; (C) Editor de expressões para o threshold. ....	74
Imagem 98 – (A) Partículas no sol com gravidade; (B) Partículas a volta do sol com a gravidade desligada; (C) Definições do sol sem gravidade. ....	74
Imagem 99 – (A) Sol verdadeiro (esquerda) e sol realizado no Maya (direita) com sistema de partícula; (B) Sol realizado no Maya com sistema de partículas em roxo. ....	75
Imagem 100 – Sistema de partículas Super Spray. (A) Emissor de partículas a volta do portal; (B) emissor de partículas central. ....	75
Imagem 101 – Identificação de ID do material. (A) Janela Material Editor; (B) Janela Environment and Effects.....	76
Imagem 102 - Render com sistema de partículas Super Spray.....	76
Imagem 103 – Ligação do PFSource com a linha através do Path Constraint. ....	77
Imagem 104 – Janela Particle View com todas as configurações dos vários sistemas de partículas PFSource.....	77
Imagem 105 –(A) Imagem com a padrão criado no Photoshop; (B) Material do sistema de partícula PFSource; (C) Prova do material no sistema de partículas. ....	78
Imagem 106 - Render do Particle Systems.....	78
Imagem 107 – Partícula Particle Systems. (A) Render do fumo; (B) Render rasto da caída do robô. ..	79

Imagem 108 – Camara target no cenário dos portais.....	79
Imagem 109 - Ligação da camara a uma animação em andamento, dando sensação de movimento na primeira pessoa.....	80
Imagem 110 – (A) Cenário sem luz ambiente; (B) Cenário com luz ambiente;(C)Luz ambiente das cenas.....	81
Imagem 111 –Rendes de Animação facial da personagem Isene. (A) Sem luz Target Direct ;(B)Com luz Target Direct.....	81
Imagem 112 – Renders do portal dos Assassinos. (A) Sem luz Target spot; (B) Com luz Target spot. .	82
Imagem 113 - Pipeline de animação. ....	82
Imagem 114 - Pipeline de animação com captura de movimento. ....	83
Imagem 115 - Captura de movimento através do Brekel Pro Body v2 e Kinect 2. ....	84
Imagem 116 - Control rig da personagem Isene no MotionBuilder.....	85
Imagem 117 - Ligação do Brekel Pro Body ao MotionBuilder.....	85
Imagem 118 – Barra Transport Controls. ....	85
Imagem 119 - Curver editor da captura de animação corporal. ....	86
Imagem 120 - Pontos faciais capturados pelo Brekel Pro Face v2.....	87
Imagem 121 – Separador Character Face Definition. (A) Pontos faciais do MotionBuilder; (B) Pontos faciais da personagem.....	88
Imagem 122 – (A) Pontos capturados pelo Brekel no software Motion Builder; (B) Captura de movimento facial.....	88
Imagem 123 - Corrigir imperfeições faciais através do FCurves no MotionBuilde. ....	89
Imagem 124 - Animação de andamento no Mixamo [140]. ....	90
Imagem 125 – Animação da personagem utilizando o Path constraint. ....	90
Imagem 126 - Animação de andamento do esqueleto no 3ds Max, utilizando o footstep. ....	91
Imagem 127 – (A) Animação manual de uma personagem; (B) Animação do urso realizada manualmente. ....	92
Imagem 128 – Personagem Isene. (A) Antes do modificador cloth; (B) Depois do modificador cloth.	92
Imagem 129 - Animação no Maya da orbita dos planetas.....	93
Imagem 130 – Discrepância de cor na junção de frames de computadores diferentes.....	94
Imagem 131 - Pipeline de pós-produção .....	95
Imagem 132 - Como importar sequências de imagens no Premiere .....	95
Imagem 133 – Montagem de imagens no Premiere.....	96
Imagem 134 - Linha temporal de vídeo final. ....	96
Imagem 135 - Linha temporal de músicas, sons e diálogo.....	96
Imagem 136 – Evolução do processo de criação dos portais.....	98

Imagem 137 – Evolução do processo de criação de alguns objetos das cenas (Amuleto, centro do templo).....	99
Imagem 138 -Todas as personagens realizadas no Fuse.....	100
Imagem 139 - Comparação desenho vs render das galáxias boa e má. ....	101
Imagem 140 - Comparação desenho vs render da orbita planetária dos bons. ....	101
Imagem 141 - Comparação desenho vs render da floresta onde bateu a nave. ....	101
Imagem 142 - Comparação desenho vs render em que Nifast mata o urso.....	102
Imagem 143 - Comparação desenho vs render da vila. ....	102
Imagem 144 - Comparação desenho vs render final da cena em que é encontrado o robô.....	103
Imagem 145 - Comparação desenho vs render do templo onde se encontra os portais.....	103
Imagem 146 - Comparação desenho vs render do portal dos assassinos. ....	103
Imagem 147 - Inspiração para o mundo dos assassinos e o seu respetivo desenho.....	134
Imagem 148 - Inspirações. (A) Arma dos assassinos; (B) Guerreiros assassinos. ....	134
Imagem 149 - Inspiração para o mundo dos inventores e o seu respetivo desenho. ....	135
Imagem 150 - Inspirações. (A) Arma dos Inventores; (B) Guerreiros Inventores. ....	135
Imagem 151 - Inspiração para o mundo dos feiticeiros e o seu respetivo desenho.....	136
Imagem 152 - Inspirações. (A) Arma dos feiticeiros; (B) Guerreiros feiticeiros.....	136
Imagem 153 - Inspiração para o mundo dos protetores e o seu respetivo desenho. ....	136
Imagem 154 – Inspirações. (A) Arma dos protetores; (B) Guerreiros protetores.....	137

# ÍNDICE DE TABELAS

Tabela 1 - Custos dos softwares de animação MoCap.....	17
Tabela 2 - Especificações da Kinect1 e Kinect2 [102] .....	29
Tabela 3 - Conceitos de cada satélite e dos seus guerreiros [120]. .....	42
Tabela 4 – Diferentes tipos de esqueletos existentes no Mixamo [59]. .....	62
Tabela 5 - Características e funções de cada computador utilizado em este projeto. ....	97
Tabela 6 - Características e funcionalidades da Pen tablet HUION H610 [31]. .....	117
Tabela 7 - Tabela com os passos necessários para chegar a uma determinada opção na elaboração do rasto da nave. ....	138
Tabela 8 – Tabela com os passos necessários para chegar a uma determinada opções na elaboração da explosão.....	138
Tabela 9 - Tabela de como foram divididas as cenas e como foram distribuídas pelos computadores. ....	141



# ACRÓNIMO

Fbx – Formato que facilita o intercâmbio de dados com maior fidelidade entre *softwares*.

MoCap – Motion Capture

MH – Makehuman

OBJ – Formato que possuem todos os dados da geometria 3D



## 1. INTRODUÇÃO

### 1.1. MOTIVAÇÃO E OBJETIVOS

A motivação para a realização deste projeto surgiu de uma disciplina de desenho 3D do Mestrado de Engenharia de Informática. Esta cativou-nos com o grande número de possibilidades que podemos realizar nesta área e também pelo realismo ao qual podemos chegar. Aos poucos, fomos notando que elaborar pequenas cenas modeladas e animadas em 3D, com um aspeto minimamente realista, demora imenso tempo a ser implementado e a chegar ao um estado final.

Neste tipo de produção, apesar de demorar imenso tempo, por vezes é necessário despender muito dinheiro em *softwares* e licenças, tanto de modelação como de animação. Além disto, são ainda necessários recursos para a realização das animações, sejam eles equipamentos ou sensores. Daqui surgiu a ideia de produzir um pequeno vídeo de animação 3D e mostrar que não é necessário despender de muitos recursos ou até de ter equipamentos de topo para o realizar.

Atualmente podemos ver todo o tipo de filmes em 3D, sejam eles a nível de animação, curtas-metragens, *trailers* de jogos, filmes, entre outros. Os custos para produzir este tipo filmes, desde o material necessário (ex. fatos com marcadores refletivos) até os mais caros *softwares* de produção são sempre um obstáculo durante a realização deste tipo de projetos.

O objetivo deste trabalho é então, mostrar uma maneira de realizar um projeto de animação, mais propriamente um *trailer* do jogo Keepers of Intheris [1], apenas utilizando um pequeno número de recursos associado a um baixo custo. Tentando evitar as soluções dispendiosas para a captura de movimento (como os fatos com marcadores), mostramos que é possível, através da utilização da Microsoft Kinect [2], captar todos os movimentos desejados e incorporá-los nos modelos 3D.

O que me levou a realizar um trailer para o jogo Keepers of Intheris foi o facto de ter acompanhado este projeto desde o seu início, tendo como criadores os meus colegas e amigos Tatiana Severim, Yuri Almeida e João Serina. Sendo que nesta altura já estava a adquirir um carinho especial pela área de 3D, decidi então seguir em frente com a ideia.

Uma das grandes diferenças entre os trailers de jogos e os trailers de filmes é que o primeiro está mais focado na parte visual do jogo enquanto que o segundo está mais focado na parte narrativa e na interpretação.

O porquê de um trailer é simples de explicar. Os trailers são considerados uma das ferramentas mais prolíficas e influentes do marketing de um jogo, sendo que são uma forma de criar a imagem do jogo e, com isto, criar a primeira impressão às pessoas.

Num nível pessoal, um dos objetivos deste projeto é obter mais experiência a trabalhar com *softwares* de produção 3D e assim conseguir aprender novas técnicas e adquirir mais conhecimento na área.



## PARTE I – ESTADO DA ARTE



## 2. ESTADO DA ARTE

### 2.1. HISTORIA DA ANIMAÇÃO 3D

Nos últimos anos, com os avanços tecnológicos que existiram, o mundo dos jogos sofreu várias mudanças e melhorias contínuas, tanto que uma criança que começa a jogar nesta altura irá acostumar-se aos gráficos de alta definição e às animações 3D. Para estas crianças, será difícil acreditar que os jogos começaram nas máquinas arcade, nos salões de jogos, e que um dos jogos inovadores dessa altura era o Pong, o qual possuía uma animação pixelizada em 2D [3]–[5].

Há 14 anos atrás era impossível ver em qualquer jogo uma simulação de cabelo dinâmico nas personagens em tempo-real, mas atualmente já existe essa realidade sem que a experiência de jogo seja sacrificada. À medida que a tecnologia vai avançando e os gráficos dos jogos vão melhorando, existe a necessidade também de melhorar as animações das personagens. Animações fluidas e realistas necessitam de ser criadas para realçar a complexidade das personagens [3].

Na década de 80, as animações começaram a surgir nos jogos 2D. Um dos primeiros jogos a aparecer com animações foi o Donkey Kong, onde as animações passavam por realizar algumas poses da personagem, que variavam entre os lançamentos de barris do Donkey Kong e também nas batidas do peito deste. Um outro jogo a levar as animações para o nível seguinte foi o The Legend of Zelda (1987) no qual já havia algumas animações de ataque [3].

Na década de 90 começaram a aparecer as animações nos jogos 3D. Em 1996, no jogo Super Mario 64, foram implementados vários tipos de animação, desde a animação da personagem em repouso, a andar, a correr, saltar, morrer, entre outras. Estas animações foram um grande avanço na altura, tanto na sua variedade como na criatividade [3], [5].

Durante o período do ano 2000 até o presente, várias gerações de consolas foram lançadas. Com isto foram também introduzidas animações mais complexas nos jogos. Jogos como o Grand Theft Auto 3, e sucessores, já deram a possibilidade de andar livremente dentro de uma cidade inteira, permitindo ao jogador fazer praticamente qualquer coisa, desde lutar contra pedestres, a roubar e conduzir carros. À medida que os jogos possuem mais possibilidades para o jogador, mais animações são necessárias para dar um grau de realismo aos jogos [3].

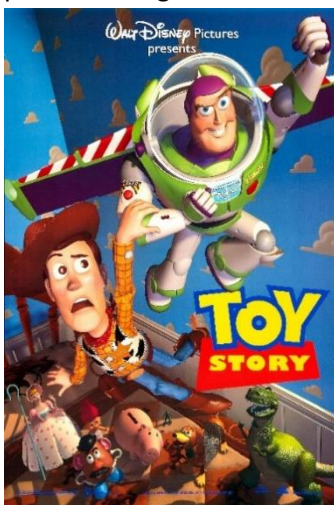


Imagem 1 - Filme Toy Story (1995) [6].

No que toca aos filmes, um dos grandes passos realizados relacionados com a animação foi no Toy Story (1995) (Imagem 1). Este filme foi a primeira longa metragem realizada apenas utilizando imagens geradas por computador [6]. Alguns dos desafios durante a realização deste filme foi trabalhar com uma equipa em que para cerca de metade nunca tinha utilizado um computador anteriormente. Com uma equipa de 110 pessoas e com um orçamento limitado, foi possível superar, conseguindo realizar animações bastante detalhadas [6], [7].

Com uma equipa de 27 animadores, estes usaram 400 modelos de computador para conseguir animar as personagens. A personagem mais complexa foi o Woody, onde foram necessários 723 controlos de movimento, dos quais 212 no rosto e 58 para a boca. Para renderizar o

filme, foram usados 300 processadores de computador de forma a chegar ao produto final. No total, foram necessárias 800.000 horas/máquina, 114,240 frames de animação, tendo sido gasto à volta de duas a 15 horas gastas por frame [6], [7].

## 2.2. PROCESSO DE PRODUÇÃO DA ANIMAÇÃO 3D

As indústrias de entretenimento utilizam uma *pipeline* semelhante à Imagem 2. Os filmes de produção 3D têm três etapas de desenvolvimento: pré-produção, produção e pós-produção.

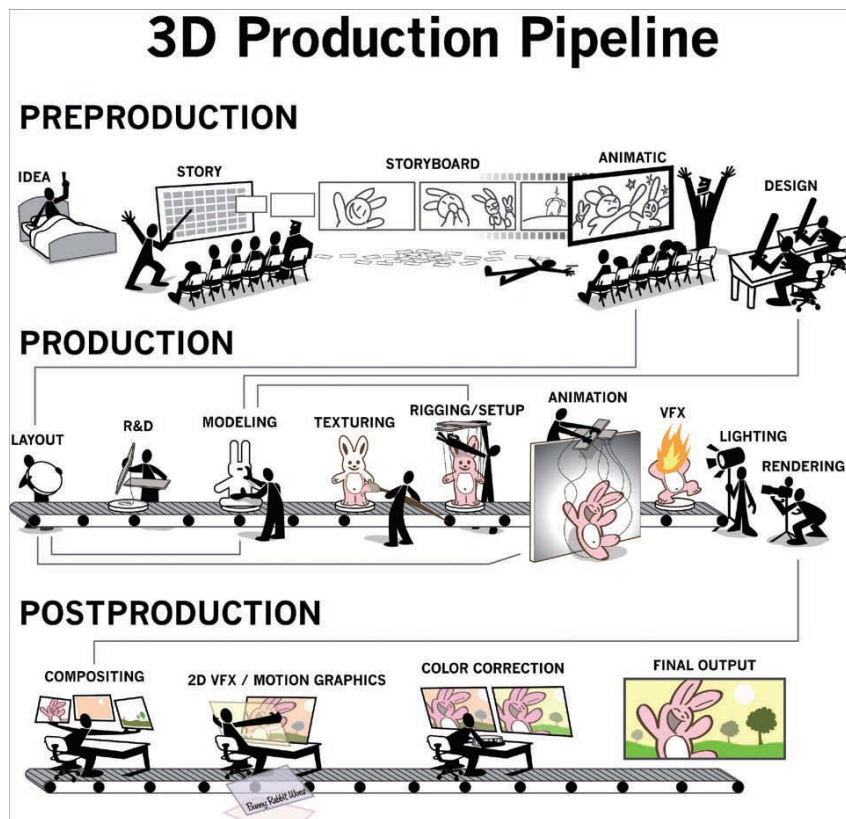


Imagem 2 - Demonstração gráfica do *pipeline* de uma animação [8].

### 2.2.1. PRÉ-PRODUÇÃO

A Pré-Produção é a fase de planeamento, design e pesquisa de todo o projeto. Esta é uma fase indispensável porque é onde as boas ideias e planos de produção são gerados, podendo assim compreender como gerir o projeto. Uma boa ideia com um plano de produção sólido, tem uma maior chance de ser completado do que uma grande ideia, mas sem nenhum plano [8].

As várias indústrias de entretenimento tratam da fase da pré-produção mais ou menos da mesma forma. Os artistas usam esta fase para gerar ideias, escrevendo-as numa história envolvente. Depois é decidido como contar essa história visualmente. Estas indústrias gastam mais de metade do seu tempo de produção na fase de pré-produção [8]. Esta fase é dividida em cinco etapas e, dependendo do tipo de projeto que está a ser produzido, pode ser necessário utilizar todas as etapas ou apenas parte delas. A ordem das etapas da fase de pré-produção não tem de ser definida e estas podem ser completadas em paralelo ou uma de cada vez para ajudar todo o processo criativo [8], [9].

A ordem típica destas etapas costuma ser a seguinte: Ideia / história, guião, *storyboards*, *animatic*/Pré-Visualização e design

---

## IDEIA / HISTÓRIA

A ideia para um projeto pode ser originada em qualquer coisa, desde o decorrer de uma conversa com alguém, de uma cor, um som ou até um cheiro. Esta ideia apenas necessita de ser capaz de iniciar um diálogo connosco mesmos ou com outras pessoas que queiram trabalhar na mesma ideia. Depois de trabalhar nesta mesma ideia, caso seja boa, é necessário transformá-la numa narrativa. Neste ponto temos apenas uma ideia básica de todo o arco da história. É também nesta etapa que se definem os detalhes básicos que desejamos incluir no projeto, tal como quem são as personagens ou até criar alguns dos momentos importantes na história [8][10].

---

## GUIÃO

O guião é a forma de escrita formal da história final e será a base escrita para o resto da produção. O guião é projetado para as equipas de pré-produção e produção para criarem uma ideia visual de toda a história. As equipas de pré-produção e produção, ao olhar para o guião, têm que ser capazes de apanhar a informação rapidamente. O formato de um guião escrito normalmente possui a duração de um minuto de tempo de ecrã por cada página. Este formato é normalmente padrão nas indústrias de animação 3D e filmes [8], [11].

---

## STORYBOARD



Imagem 3 - Frame do storyboard do filme *Toy Story* [143].

O *storyboard* é uma forma visual de representar uma história, juntando ideias, câmaras, efeitos visuais, poses importantes das personagens e eventos marcantes nas cenas, onde cada imagem representa um momento na história, como podemos ver no exemplo do filme *Toy Story* (Imagem 3). Não é necessário realizar um *storyboard* pormenorizado, ou seja, desenhar as personagens detalhadamente ou cada movimento realizado. Um *storyboard* pode ir desde um pequeno esboço até um desenho

extremamente profissional. Nestes desenhos, o que é realmente importante é transmitir a informação necessária para compreender a visão do artista naquele momento da história [8].

Este é o modo de pré-produção mais barato, pois os gastos de modificar alguma parte da história são praticamente nulos, visto que nesta fase ainda não foi inicializada nenhuma parte do trabalho prático [8], [9], [12].

---

## ANIMATIC/PRE-VISUALIZATION

Um *animatic* é uma forma animada do *storyboard*. O *animatic* pode ser criado de qualquer forma semelhante ao ato de criar um simples filme no computador. Na sua forma mais simples, um *animatic*, é apenas uma sequência de imagens do *storyboard* sincronizadas a um simples diálogo e efeitos de som com o intuito de mostrar o ritmo e ângulos das câmaras utilizados na sequência do projeto. Na Imagem 4 podemos ver toda a sequência do *storyboard* do filme *Toy Story*. Um dos objetivos da realização de um *animatic* é perceber se o *storyboard* possui uma fluidez visual no decorrer da história. Cortar partes ou sequências da animação na fase de pós-produção é muito dispendioso e só deverá ser realizada caso seja absolutamente necessário. Realizando um *animatic* na fase de pré-produção ajuda a evitar imprevistos dispendiosos ao longo do projeto [8], [9].



Imagem 4 - *Animatic* do filme *Toy Story* [144].

---

## DESIGN

No design é onde o aspeto final do projeto é decidido. Para as indústrias de entretenimento, esta fase inclui o design das personagens, adereços, roupas e cenários. Nesta fase, não importa o que é utilizado para criar a arte, apenas importa que o conceito e o design sejam totalmente transmitidos, como foi realizado com a personagem *Woody* do filme *Toy Story* (Imagem 5).

A razão pela qual a etapa de design está na fase de pré-produção deve-se ao facto de que o artista que realiza o design pode rapidamente desenhar várias opções para o design de uma personagem num dia e depois finalizar o design de maneira a agradar o gosto do diretor ou realizador do projeto. Em contrapartida, um artista de modulação 3D pode demorar até uma semana para criar um modelo final, e o diretor ou realizador do projeto poderá não querer esperar tanto devido às possíveis alterações ao modelo que possam existir após a criação 3D do artista [8], [10].



Imagem 5 - Ideias do visual que *Woody* personagem do *Toy Story* [13].

## 2.2.2. PRODUÇÃO

Assim que o *storyboard* é aprovado, o projeto entra para a parte de produção. É na produção que o trabalho realmente começa, baseado nos princípios e orientações definidas durante as fases anteriores. A Imagem 6 mostra um exemplo de *pipeline* normalmente seguido pelas empresas de entretenimento durante a fase de produção [8].

### PRODUCTION

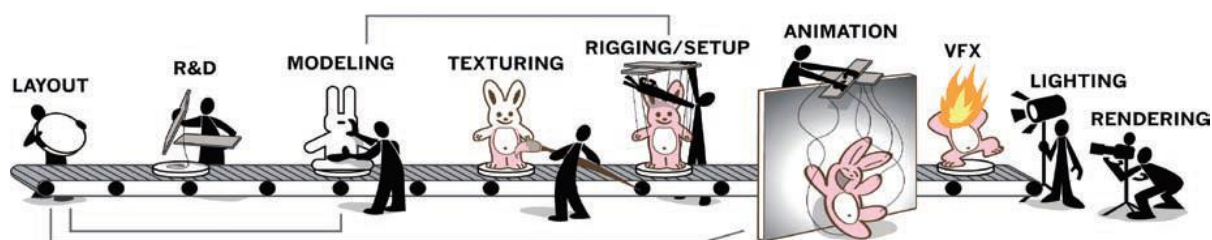


Imagem 6 - Pipeline de produção [8].

### LAYOUT

Na fase de *Layout* 3D é criada uma versão 3D do *animatic*. O *layout* 3D deve começar assim que sejam criadas as representações de *low poly* dos modelos finais. Estas representações não precisam de atributos muito detalhados, tais como cara ou até mesmo dedos para descrever a história visual (Imagem 7). O objetivo desta fase é mostrar os ângulos das câmaras, a escala das personagens e o seu campo de ação e enquadramento espacial no cenário, a personagem a se mover de um ponto para outro ou a direção para a qual a personagem está a olhar [8], [9], [12].



Imagem 7 - Layout da personagem Buzz do filme Toy Story [145].

### INVESTIGAÇÃO E DESENVOLVIMENTO

Nesta fase, os artistas de desenvolvimento visual exploram os conceitos da história do projeto, desde o design das personagens ao design do ambiente [12]. Por exemplo, no filme da Pixar “À Procura de Nemo”, a equipa responsável pela investigação e desenvolvimento teve que descobrir como criar a aparência da água, incluindo os pequenos objetos/efeitos flutuantes na água. Na altura em que este filme foi produzido, com uma data de lançamento de 2003, nenhum filme animado 3D tinha antes retratado a água, pois era considerada muito difícil de controlar e renderizar com eficiência. No entanto, “À Procura de Nemo” teve elementos de água em quase todas as cenas. As equipas de Investigação e Desenvolvimento estão sempre a trabalhar na próxima grande revolução para a indústria. A experimentar diversos elementos na apresentação do filme para transmitir emoções a quem vê o filme e também criar uma ligação emocional com a história. Desta forma, esta é a equipa que tem maior probabilidade de inovar ao criar um efeito nunca visto antes [8], [12].

---

## MODELAÇÃO

Um modelo é uma representação numa superfície geométrica de um objeto que pode ser rodado e visto num *software* de animação 3D [8]. Nesta fase, os artistas responsáveis pela modelação têm o trabalho de criar os modelos complexos que são necessários para a animação das personagens, adereços e cenários, como foram realizados com as personagens do filme *Toy Story* (Imagem 8) [12]. Alguns dos cuidados que os artistas necessitam durante o processo de modelação é garantir que os modelos são funcionais para poderem ser trabalhados no resto da *pipeline* de produção. Além disto, os modelos necessitam de ter uma boa estética visual e também necessitam de funcionar corretamente quando deformados, só assim poderão ser animados sem dificuldades acrescentadas [10]. Após os modelos estarem aprovados, podem então passar para as fases seguintes [9].

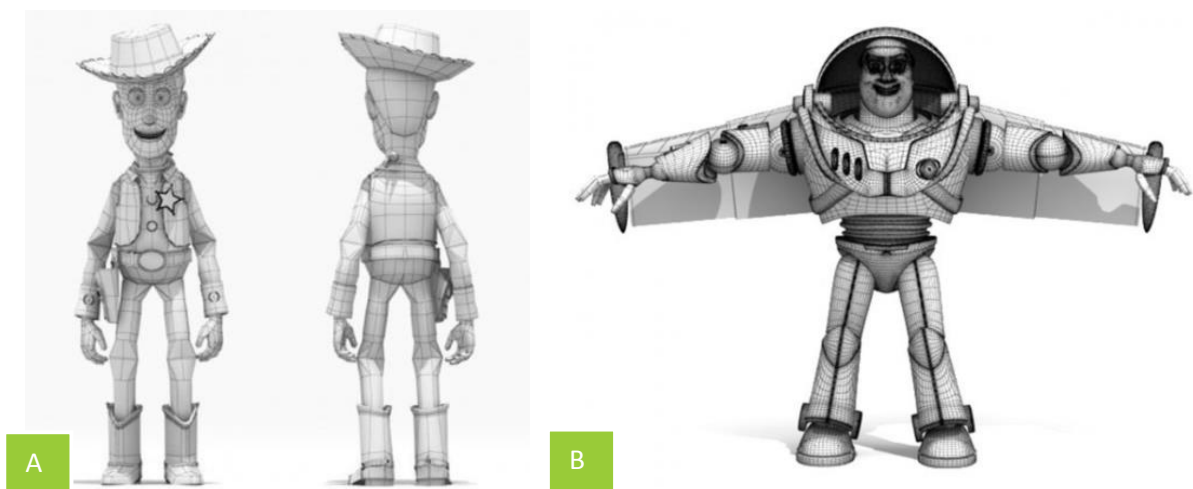


Imagem 8 -Modelação de personagens do filme *Toy Story*; (A) *Woody*; (B) *Buzz Lightyear* [14]

---

## TEXTURIZAÇÃO

Nesta fase, o artista de texturas aplica propriedades de cor e superfícies aos modelos (Imagem 9). O artista pode criar a textura do zero ou então através da edição de uma imagem já existente. A responsabilidade destes artistas é fazer com que a superfície do modelo seja igual ao conceito visual previamente aprovado [8], [9].



Imagem 9 - Texturização de cenário e personagens do filme *Toy Story* [15].

---

## RIGGING

Esta fase é composta por duas etapas, o *rigging* e o *skinning*. Estas duas etapas consistem em adicionar ossos e articulações a uma personagem (Imagem 10). Isto permite que os modelos realizem movimentos de acordo com os ossos colocados. Sem os controlos de *rigging* corretos, os animadores não conseguem criar as poses que desejam para aquele modelo [16]. Na fase de *rigging*, é da responsabilidade da pessoa que trata do *rigging* ajudar os animadores a criar um sistema de controlos que permita lhes trabalhar o mais rapidamente e eficientemente possível [8].

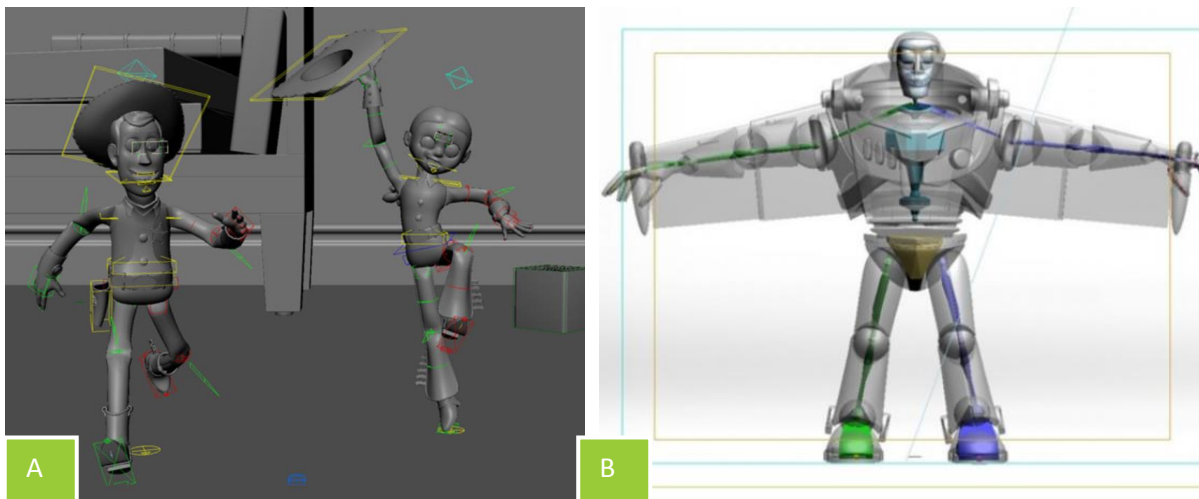


Imagem 10 – (A) Personagens do filme *Toy Story* com *rigging* e controladores [17]; (B) *Rigging* da personagem Buzz [14].

---

## ANIMAÇÃO

A animação é a realização dos movimentos de todos os elementos, desde as personagens (cara e corpo) aos outros elementos da cena, por exemplo, edifícios, água, céu ou vegetação. Nesta fase, os animadores são responsáveis por dar vida a todos estes elementos [10], [12]. Os animadores usam os ficheiros da fase *layout* como ponto de partida para obter as personagens, cenário e a câmara para a cena, e depois adicionam a encenação final ou o movimento às personagens (Imagem 11). Existem três



Imagem 11 - Animação de cena do *Toy Story* [146].

tipos de animação: *keyframes*, *motion capture* e animação gerada automaticamente (*procedural animation*). Animação por *keyframes* é onde o animador cria à mão cada momento da animação. *Motion capture* é onde o animador transfere o movimento capturado de um ator através de um mecanismo de captura de movimentos para um modelo 3D. Na animação gerada automaticamente o programador cria um conjunto de regras e a personagem move-se de acordo com essas regras [8]. Este último tipo de animação é muito usado em jogos para criar diversidade nas animações. Por exemplo, é capaz de gerar milhões de animações únicas para o movimento de uma personagem.

---

## EFEITOS VISUAIS 3D

Os artistas de efeitos visuais 3D (VFX) são animadores de elementos naturais, tais como a água, fogo, fumo, neve, chuva, vento e até explosões e também elementos como cabelos e roupas. A maior parte dos efeitos visuais 3D são baseados num motor dinâmico de física de um *software* que utilize a física natural, tal como o ar ou a gravidade, de forma a manipular estes sistemas. Os tipos de efeitos que um artista de 3D VFX trabalha é normalmente chamado de “*invisible effects*” (efeitos invisíveis) porque o objetivo destes efeitos é melhorar as gravações das cenas em que os efeitos se inserem, e não para tirar o protagonismo da cena [8], [10]. No filme *Toy Story* um dos efeitos visuais realizados foi o pelo da personagem que se caracterizava como um urso (Imagem 12).



Imagem 12 - Efeitos visuais adicionados no filme *Toy Story* [15].

melhorar as gravações das cenas em que os efeitos se inserem, e não para tirar o protagonismo da cena [8], [10]. No filme *Toy Story* um dos efeitos visuais realizados foi o pelo da personagem que se caracterizava como um urso (Imagem 12).

---

## LUZES

A luz é uma parte essencial na visualização de uma história e é a última etapa na produção (Imagem 13). Nesta fase, os artistas adicionam luzes de forma a fazer com que as personagens ou cenários pareçam realistas. A luz conduz e foca o olhar do espectador para onde olhar e transmite as emoções de cada cena [12]. Tomando como exemplo o filme “*À Procura de Nemo*”, a luz ajudou a criar uma sensação de estar submerso [18]. Os artistas de luzes 3D têm acesso a múltiplos tipos de luzes que imitam as luzes no mundo real – por exemplo focos de luz, lâmpadas e luz do sol. É necessário que estes artistas definam como é que as luzes interagem com os diferentes tipos de materiais, tendo de ter sempre o



Imagem 13 - Luzes adicionadas a cena do *Toy Story* [15].

Os artistas de luzes 3D têm acesso a múltiplos tipos de luzes que imitam as luzes no mundo real – por exemplo focos de luz, lâmpadas e luz do sol. É necessário que estes artistas definam como é que as luzes interagem com os diferentes tipos de materiais, tendo de ter sempre o

cuidado de manter a qualidade e realismo das texturas usadas nos objetos nos quais a luz irá incidir. No fundo, o seu trabalho é criar uma harmonia de cores e luminosidade na reprodução das cenas do filme [8], [9].

---

### 2.2.3. POS PRODUÇÃO

A Pós-Produção é a terceira e última etapa na criação de um projeto 3D. As indústrias de entretenimento utilizam esta etapa para fazer com que o seu projeto se destaque através de efeitos

visuais e correções de cor e áudio [8], [9]. A etapa de pós-produção inclui as seguintes fases: Composição; Edição de Som e Vídeo; Correção de cor e vídeo final

---

## COMPOSIÇÃO

Na fase de composição, o artista de pós-produção junta todos os elementos 3D produzidos, pelos outros artistas anteriores, no decorrer da *pipeline* para criar uma imagem renderizada final pronta para utilizar no filme [8], [9].

---

## EDIÇÃO DE SOM E VÍDEO

Os artistas de edição de som têm como responsabilidade selecionar e juntar as gravações de som para obter o áudio final. Estes artistas têm o cuidado de assegurar a sincronização dos lábios com as falas e também de adicionar todos os efeitos de som necessários [8], [9].

Por outro lado, a edição de vídeo trata-se do processo de manipular e rearranjar as gravações de forma a obter um resultado final sem falhas, removendo também as gravações ou filmagens que não sejam necessárias. A edição de vídeo é um passo crucial para obter um vídeo que flua de acordo com o objetivo inicial da história. É também nesta fase que são incluídos os títulos e outros efeitos visuais ao vídeo final [8], [9].

---

## CORREÇÃO DE COR

A correção de cores é quando todo o projeto é ajustado para garantir que todas as cores das imagens são consistentes e correspondem ao resultado final. Todos os projetos de animação 3D são trabalhados como gravações individuais e, em seguida, olhados como sequências e, finalmente, como um projeto inteiro. A correção de cor pode ser muito artística e altamente técnica ao mesmo tempo [8].

---

## VÍDEO FINAL

O resultado final da animação 3D pode vir em muitas formas diferentes, como um filme, uma imagem parada, filme 3D estereoscópico e até em mídia impressa. Cada uma destas formas têm *workflows* diferentes e limitações técnicas. O tipo de resultado final mais comum é o vídeo digital, o qual pode ser reproduzido no computador, numa televisão ou na Internet. A maior limitação técnica para os vídeos digitais é a correção de cores, porque nem todos os computadores e monitores são calibrados da mesma maneira e possivelmente terão cores diferentes uns dos outros [8].

## 2.3. FERRAMENTAS

Para escolher as ferramentas adequadas é necessário saber quais são as ferramentas que existem no mercado para elaboramos o projeto. E para isso pesquisamos as ferramentas e não só a nível de modelação 3D mas também para realizar captura de movimentos. Saber o que é a captura de movimentos, como surgiu e que tipos de captura existiam era imprescindível.

---

### 2.3.1. SOFTWARE

Para a realização da *pipeline*, o primeiro passo a dar é a escolha de todo o *software* necessário. Para isso realizámos uma pesquisa dos *softwares* de modelação, animação e produção. Um dos pontos fundamentais nesta pesquisa foi garantir que o *software* em questão tinha as funcionalidades necessárias para efetuar as operações pretendidas e que os *softwares* escolhidos eram compatíveis entre si.

---

#### 2.3.1.1. SOFTWARE DE MODELAÇÃO – 3D MAX/MAYA/ZBRUSH

Hoje em dia existem muitos *software* de modelação 3D, não só para o ramo empresarial, mas também educacional ou pessoal. Para escolher os pacotes de *software* foi necessário realizar alguma pesquisa. Para este projeto apenas escolhemos três ferramentas, o 3ds Max [19], Maya [20] e Zbrush [21].

A Autodesk [22] (a empresa que desenvolveu o 3ds Max e o Maya), permite a interligação entre vários *softwares* da empresa, facilitando assim a exportação/importação de ficheiros sem nos preocuparmos com a sua extensão. Uma vantagem destes *softwares* é que ambos suportam o formato FBX [19].

O 3ds Max é conhecido por ser uma ferramenta de modelação tridimensional que permite criar modelos 3D complexos. É considerado um dos principais *softwares* de animação 3D e utilizado pelas grandes empresas de animação e design. É utilizado na área de jogo, produção de *trailers*, filmes de curta e longa metragens e publicidades [19]. O 3ds Max incorpora uma grande biblioteca de modificadores. Se juntamos a isto a facilidade da interface e a excelente capacidade de visualização arquitetónica, podemos dizer que o seu manuseamento será mais fácil, logo permite que qualquer pessoa inexperiente possa criar um modelo 3D [23]. Alguns utilizadores preferem o 3ds Max, pela facilidade que este oferece na configuração dos materiais [24]

Apesar do Maya ter algumas funcionalidades parecidas ao 3ds Max, o Maya é mais complexo e completo, sendo o *rigging* o seu melhor aspeto. Em qualquer um dos *softwares* de modelação, criar um *rigging* facial pronto para animar é um dos aspetos mais desafiantes e demorado na criação de uma personagem [23]. A modelação no Maya é mais difícil de realizar que no 3ds Max devido à sua complexa interface gráfica. No entanto, é um *software* muito mais poderoso quando se trata de animação graças à sua grande biblioteca de ferramentas de animação. A mesma animação pode ser realizada em qualquer um dos *softwares*, mas quando falamos em facilidade e no número de ferramentas disponíveis, o Maya é o mais forte [23]. Devido às dificuldades na modelação (uma das componentes básicas do 3D), a curva de aprendizagem do Maya é mais longa [23], [25].

O Maya é compatível com vários plug-ins que ajudam na criação de efeitos realistas [25]. A linguagem de script Maya Embedded Language (MEL) é personalizável e faz com que este *software* seja também usado no sector de cinematografia, visto que estes profissionais conseguem adaptar este *software* às suas necessidades, facilitando o seu trabalho [23], [26]

Comparando um ponto importante entre estes dois *softwares*, o Maya é mais utilizado para efeitos visuais e animação nas indústrias de jogo, enquanto o 3ds Max é mais adequado para o desenvolvimento de jogos e para a arquitetura [27].

O *Zbrush* é um *software* de esculpir estruturas orgânicas e uma ferramenta artística poderosa. Embora o 3ds Max e o Maya sejam bons para modelar, não permitem ter o detalhe desejado que o *Zbrush*

permite. Para isso é utilizado *software* de esculpir. O 3ds Max e o Maya são ideais para modelar superfícies rígidas, enquanto que o Zbrush para além disto, permite trabalhar com modelos mais orgânicos e detalhados [28]. O Zbrush é conhecido por lidar com milhões de polígonos com uma grande facilidade e pela ferramenta *ZRemesher* que reorganiza os polígonos do modelo em instantes. Contudo, trabalhar com uma grande quantidade de polígonos não significa que o modelo seja melhor [28]. O Zbrush permite esculpir e texturizar ao mesmo tempo através de uma grande gama de opções para modificar parâmetros de pincéis e de tipos de traço. Existem vários tipos de pincéis e alguns deles são para extrair formas [29].

Juntamente com a *pen tablet*, foram utilizados programas tais como o *software* de modelação *Zbrush* e o Photoshop que será apresentado mais à frente. Tendo acesso a um periférico deste tipo, fez sentido utilizá-lo ao longo deste projeto para modelar, esculpir e texturizar os cenários da história, conseguindo desta forma obter um maior nível de detalhe, trabalhando de uma forma mais simples o que permitiu economizar tempo.

A *pen tablet* é utilizada em muitos lugares e para diferentes finalidades, desde um pequeno risco ou pequeno esboço até esculturas e desenhos profissionais. A caneta é utilizada para desenhar na superfície da tablet. Esta controla a pressão realizada, transmitindo linhas com pouca ou mais espessura e dando assim uma sensação mais orgânica e natural aos traços. Estes *pen tablets* são utilizados em conjunto com programas de edição de imagem, desenhos técnicos ou modelações [8].



Imagem 14 - Pen tablet HUION H610, utilizado no projeto.

A *pen tablet* utilizada foi o modelo HUION H610 (Imagem 14). As suas características e funcionalidades encontram-se nos anexos na Tabela 6 [30]. Uma das desvantagens do 3ds Max, é que este apenas é suportado pelo sistema operativo Windows, enquanto que o Maya e o Zbrush são suportados não só pelo Windows, mas também por macOS. Em relação ao preço, os dois *softwares* disponibilizados pela Autodesk (3ds Max e Maya) têm uma licença de estudante, que permite usufruir de três anos gratuitos [19], [26]. O mesmo não acontece com o Zbrush mas, em contrapartida, oferecem um desconto de 50% para os estudantes no valor total do *software* e também oferecem uma versão gratuita de 45 dias [31].

Com recurso a este periférico, é ainda possível entrar no mundo da modelação 3D, pois permite modelar todo o tipo de personagens ou cenários e esculpi-las com um enorme nível de detalhe. Uma das grandes vantagens deste periférico em relação ao rato, é que depois de algum tempo de uso, torna-se mais fácil e rápido de utilizar para modelar ou esculpir algo [32].

---

### 2.3.1.2. SOFTWARE DE ANIMAÇÃO E MOCAP

De tal modo como a modelação 3D, onde existem pacotes de *software* específicos, para realizar a captura de movimento isto também acontece. Para isso precisamos de *software* que utilize a Kinect como sensor de captura de movimento e de um *software* de animação 3D.

O MotionBuilder [33] da Autodesk, é um *software* de animação de personagens 3D que oferece uma vasta coleção de ferramentas de animação em tempo-real, realizando desde as tradicionais animações

*keyframe* às complexas edições de *captura de movimento*. Permite também a criação eficiente de animações para jogos ou filmes. O Motion Builder, permite criar, editar e reproduzir animações de personagens complexas, num ambiente interativo que é ideal para grandes volumes de animação, tais como, cinematografia virtual, pré-visualização e animação de desempenho. O MotionBuilder é construído sobre uma arquitetura de exibição em tempo-real, que permite trabalhar sem a necessidade de perder tempo a gerar pré-visualizações das animações produzidas. Para permitir a produção de animações de qualidade é disponibilizada uma lista repleta de ferramentas para criar animações por *keyframes* avançadas e animações em camadas. O manuseamento da informação é simplificado através de ferramentas integradas para editar e misturar animações. O MotionBuilder possui ainda uma tecnologia para a extração de fonemas o que permite gerar animações faciais realistas e a sincronização com os lábios independentemente da linguagem utilizada [33].

As funcionalidades do MotionBuilder podem ser expandidas facilmente através de scripts na linguagem *Python* ou através do *software development kit* (SDK) da ferramenta, o *Open Reality*, baseado em C++. O MotionBuilder é facilmente integrável com os ambientes de produção existentes pois é compatível com sistemas de gestão de produção e motores de *render* baseados em *hardware*. Tudo isto é possível graças ao suporte nativo do formato FBX, garantindo ainda a transferência de objetos do 3ds Max, Maya, ou qualquer outro *software* 3D que suporte este formato. A ligação entre o MotionBuilder e a Kinect para capturar uma animação através de MoCap não é direta e, para realizar esta ligação, é necessário um intermediário que os comunique [33].

Hoje em dia as grandes empresas que trabalham e desenvolvem em *software* 3D têm que pagar para ter acesso às melhores ferramentas de trabalho para poderem desenvolver os seus projetos 3D. Estes projetos podem ir desde a simples modelação de um objeto ou personagem até a projetos mais complexos, onde além da modelação pode ainda ser necessário realizar efeitos visuais e animações, o que implica a existência de muitas mais etapas técnicas, etapas essas que serão explicadas mais à frente.

Uma das grandes preocupações das empresas é o custo do *software* e custos de licenciamentos associados. Por isso testamos três *softwares* de captura de movimento o Brekel (Brekel Pro Face [34] e Brekel Pro Body [35]), IpiSoft [36] e o iClone [37]. A escolha do *software* de animação através da Kinect 2 foi baseada nos requisitos pedidos por cada uma das empresas, os descontos associados para estudantes e claro o que cada *software* incluía.

O Brekel se for utilizado juntamente com o sensor Kinect permite detetar até seis pessoas e seguir o rasto do estado das mãos e a da sua inclinação. Com uma vertente de seguir o rasto do rosto, o Brekel Pro Face v2 permite capturar expressões também até seis pessoas. Rastreia 20 formas de rostos diferentes e ainda a posição e rotação da cabeça [34], [35].

O IpiSoft suporta a captura de movimento através de dois sensores de profundidade e sem necessidade de marcadores, incluindo todos os recursos básicos para uma captura de movimento de alta precisão. Este *software* permite rastrear múltiplas pessoas e as das suas mãos, mas os movimentos são simples e sem rotação. Infelizmente este não permite a gravação de animações em tempo real o que o desvaloriza em relação aos outros dois *softwares* apresentados [36],[38].

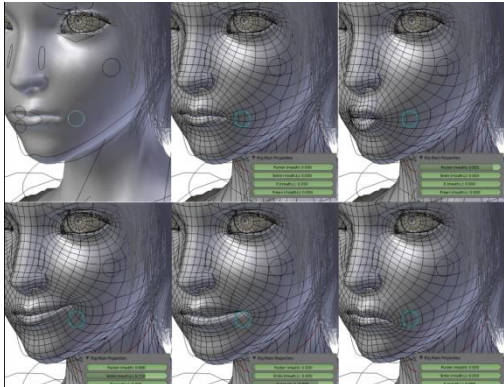


Imagem 15 - Rostos com morphs/blendshapes de uma personagem [147].

O iClone permite realizar movimentos em tempo real utilizando o seu plug-in ou inserir movimentos pré-definidos existentes na sua biblioteca. Através do 3DXchange podem ser importados os esqueletos de personagens animadas para *softwares* 3D. O iClone permite elaborar o design das personagens com uma grande qualidade visual combinada com movimento, sincronização de lábios e animações faciais através de *morphs/blendshapes*. [39], [40]. O morphs ou blendshapes é uma maneira de criar expressões faciais em uma personagem através das varias deformações no rosto (Imagem 15) [41].

Qualquer um dos *softwares* de captura de movimento pode ter ligação direta ou podem ser importados os dados para o MotionBuilder. O ponto forte do Brekel é que pode ser utilizado em tempo real e grava diretamente no MotionBuilder através de um plugin, exportando uma mesh com *morphs/blendshapes*. Permite ainda a gravação de áudio através do microfone da Kinect [35],[34].

Qualquer um dos *softwares* de captura de movimento está disponível apenas para Windows e qualquer um pode ser usado para realizar captura de movimento (MoCap) sem necessidade de marcadores. Para poder utilizar o Brekel é necessário cumprir com alguns requisitos mínimos, tal como ter uma porta USB 3.0 tipo A, pelo menos 4 GB de memória RAM e um processador i5 ou i7 Dual Core. Para o iClone o único requisito necessário é ter um processador x64 Dual core ou superior [34], [35], [42].

Tabela 1 - Custos dos *softwares* de animação MoCap

Programa	Versão Trial	Valor	Desconto	Características
Brekel[43]	30 Dias	\$239	30% se for aluno	ProBody v2 e ProFace v2
iPi MocapStudio[36]	30 Dias	\$45 (3 meses) \$256.50 (3 anos)	30% se a universidade estiver credenciada pela secretaria de educação dos EUA	Animação corporal. Movimentos simples e sem rotação
iClone6[37]	30 Dias	\$497.95	50% para a educação	Animação facial e corporal

Devido ao preço e por todas as suas funcionalidades, o *software* de captura que foi selecionado foi o Brekel. A versão trial apenas serviu para realizarmos as provas necessárias para ter a certeza que o *software* escolhido era o mais adequado para o projeto. O valor referido na Tabela 1 para o Brekel, incorpora dois *softwares* de captura de animação, o facial e o corporal.

### 2.3.1.3. SOFTWARE DE TEXTURIZAÇÃO – PHOTOSHOP – SUBSTANCE PAINTER

Para poder texturizar os modelos 3D foi necessário encontrar *softwares* que permitissem a manipulação de imagens e também a visualização em tempo real das suas alterações. Entre outros *softwares*, o Adobe Photoshop [44] e o Substance Painter da Allegorithmic [45] são conhecidos na

indústria de entretenimento como as ferramentas de referência [46], [47]. Os dois *softwares* têm preços semelhantes através de um sistema de subscrição, mas o Substance Painter oferece ainda uma licença de um ano para educação [48], [49].

O Photoshop é um *software* de processamento de imagem extremamente poderoso. Podem ser realizadas múltiplas coisas nele, entre elas, a produção de texturas. Ao utilizar as camadas (*layers*), é possível combinar várias imagens como se estas estivessem sobrepostas. Usando a funcionalidade de máscaras, é possível criar os vários mapas de texturas num único ficheiro [50]. Através do plugin NVIDIA Normal Map Filter, o Photoshop permite ainda a geração de Normal Maps e permite pré-visualizar como esta textura irá ficar [51]. O Normal Maps permitem adicionar detalhes de alta resolução a objectos com baixo número de polígonos.

O Substance Painter é um *software* que, ao contrário do Photoshop, só pode ser utilizado para texturizar modelos 3D. Em contrapartida, tem todas as funcionalidades de processamento de imagem que o Photoshop também possui, permitindo ainda trabalhar com as camadas e as máscaras. A grande vantagem do Substance Painter em relação ao Photoshop no que toca a texturização, é que permite visualizar simultaneamente a imagem 2D da textura e o resultado final no modelo 3D (Imagem 16). O Photoshop é um pouco mais limitado neste aspeto, mas, é possível visualizar a textura no modelo 3D se for instalado a versão Photoshop Essentials.

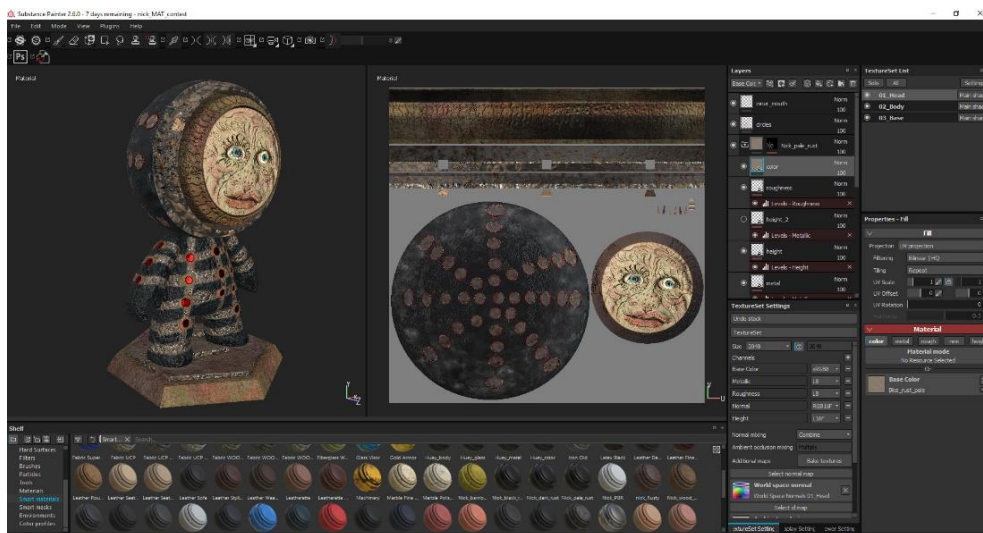


Imagem 16 - Substance Painter com as duas visualizações 3D e 2D [52].

Antes da texturização do modelo 3D, é necessário fazer a projecção do detalhe do modelo *high poly* para o modelo *low poly*. Ao usar o Photoshop, este processo tem de ser feito no programa de modelação escolhido ou noutro programa à parte, algo que não é necessário quando se usa o Substance Painter. Uma outra grande vantagem de utilizar o Substance Painter para a projecção, é que este permite dividir o modelo *high poly* em vários segmentos e associá-los aos mesmos segmentos do modelo *low poly*, criando todos os mapas de texturas necessários. Isto reduz o tempo necessário para esta etapa (pelo menos 3x mais rápido) e também reduz a probabilidade de haver erros durante a projecção (Imagem 17) [53].

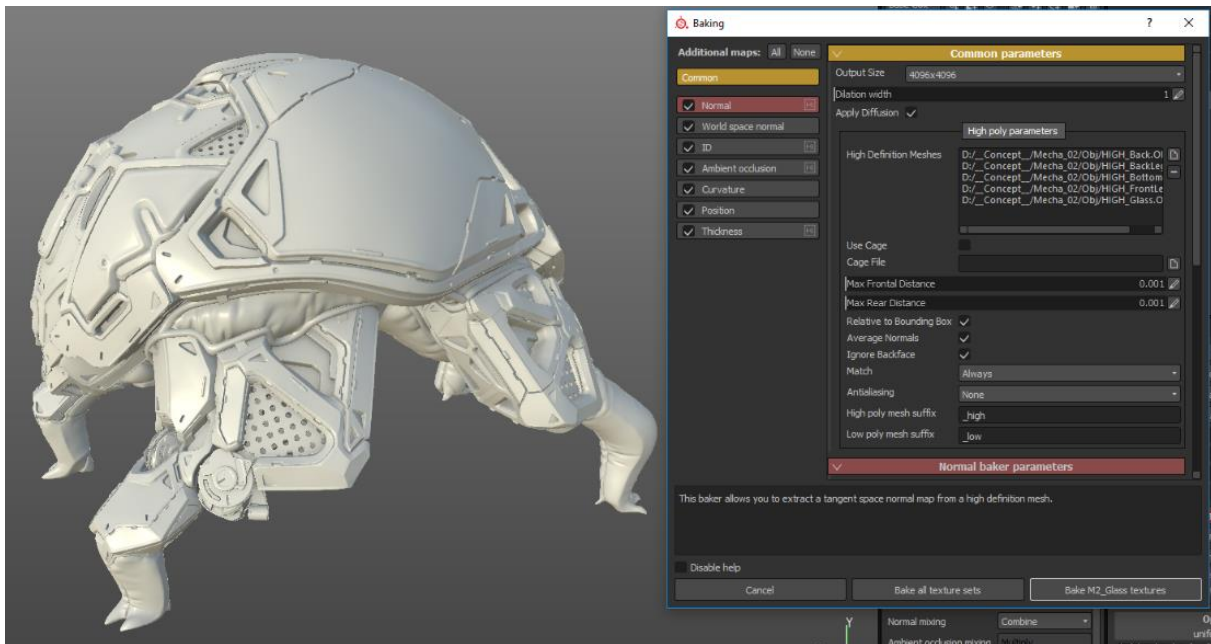


Imagem 17 – Baking Substance Painter[53].

Para modificar uma textura no Photoshop, só é possível modificar um mapa de cada vez (diffuse, normal, specular ou lightmap). No Substance Painter, é possível modificar todos os mapas de textura simultaneamente. Isto deve-se ao facto de que cada vez que se usa o pincel no Substance Painter, é possível definir a quantidade de cor (diffuse), relevo (normal/height), reflexão da luz (specular) e sombras (lightmap) que é aplicada quando se pinta a textura (Imagem 18) [54]. Uma outra grande vantagem é que alguns materiais frequentemente usados já estão criados (*smart materials*), o que aumenta a produtividade, pois não é necessário criá-los para usar no modelo 3D, basta apenas arrastar para uma camada do projeto [55].

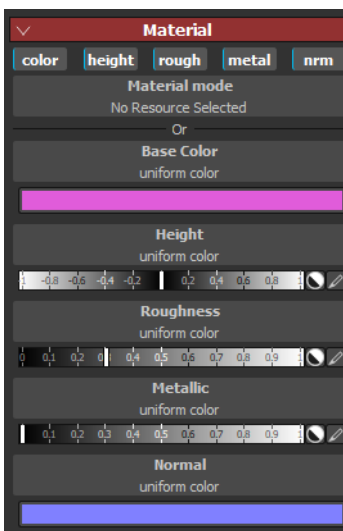


Imagem 18 - Janela com os vários materiais do Substance Painter.

Ambos os *softwares* permitem a exportação das texturas para poderem ser usadas nos modelos 3D. Mais uma vez, o facto do Substance Painter ser uma ferramenta desenvolvida a pensar só em texturização de modelos 3D, tem a vantagem de poder exportar todas as texturas num único clique. Por outro lado, no Photoshop é necessário exportar todos os mapas individualmente. Os dois *softwares* são capazes de fazer todo este processo. Mas o facto do Substance Painter ter sido criado especificamente para esta tarefa, faz com que todas as suas funcionalidades tenham como prioridade melhorar e tornar mais eficiente este processo. No entanto, o Photoshop é extremamente versátil e as suas funcionalidades adicionais poderão ser usadas noutras fases do processo de produção e também para trabalhar sobre os ficheiros criados no Substance Painter [46]. O Substance Painter foi utilizado para texturizar coisas mais orgânicas e com uma maior dimensão, que se fosse realizado no Photoshop seria muito mais complicado de realizar. Por outro lado o Photoshop foi utilizado para a edição de texturas pequenas.

#### 2.3.1.4. SOFTWARE DE CRIAÇÃO DE PERSONAGENS – FUSE – MAKE HUMAN

Para criar as personagens foi necessário encontrar *software* específicos para esta funcionalidade. Através de *softwares*, tais como o Adobe Fuse [56] ou o Make Human [57] podemos modelar personagens sem a necessidade de ser um artista experiente no ramo de 3D.

O Fuse cria personagens em 3D sem necessidade de ter vastos conhecimentos na área. Neste, pode ser criada qualquer tipo de personagem, permitindo também a alteração da anatomia da personagem que esta a ser modelada, de maneira a definir o seu peso, a cor da sua roupa e cabelo, ficando assim com uma aparência mais próxima ao que se pretende e mantendo o realismo. Independentemente das dimensões definidas para a personagem, a roupa ajusta-se ao modelo de forma a que não se deforme. No Fuse, existe a opção de salvar o modelo na biblioteca Creative Cloud [58] e, desta forma, exportar para outro *software* como Mixamo [59] para realizar o rigging, ficando assim pronto para realizar qualquer tipo de animação [60]–[62]. O Fuse vem incorporado com uma biblioteca com mais de 70 partes do corpo (humanos e humanoides), mais de 150 peças de roupa e 42 texturas dinâmicas. Adicionalmente, possui um Auto-rigger [63] e também disponibiliza animações grátis e ilimitadas (Imagem 19). O Auto-rigger é um modo online da Mixamo onde a personagem criada no Fuse será animada. É possível definir qual o tipo de esqueleto pretendido para a personagem a ser modelada e também a possibilidade de colocar blendshapes na personagem [64].



Imagem 19 - Fluxo de trabalho da criação de uma personagem oferecido pelo Mixamo [64].

A interface deste software é muito simples e trabalhar com o Fuse é fácil pois o seu processo não é confuso, mesmo possuindo uma vasta gama de opções. É intuitiva e sabemos sempre onde encontrar o que desejamos. Para a criação de uma personagem temos sempre de seguir passos: Montagem, Vestuário e Personalização [65]. Através de controladores deslizantes simples podemos ajustar expressões e modificar partes do corpo, o que também pode ser realizado com um simples arrastamento de rato da zona pretendida, mantendo a sua topologia limpa [61], [62]. Para usufruir deste produto não é necessário comprar nenhuma licença adicional ou qualquer tipo de direitos, é gratuito para qualquer pessoa que tenha conta na Adobe [62]. Por outro lado, o Make Human também é um gerador de personagens realistas. Neste caso, já existe um modelo base de um humano e a partir dele realizamos todas as mudanças necessárias (cabelo, músculo, tamanho). As personagens podem ser editadas através de arquivos de texto, alterando as suas dimensões [66], [67]. O Make Human é um software gratuito e de código fonte aberto, tendo sido criado para complementar o Blender [68]. Uma desvantagem neste software é a quantidade limitada de roupas e acessórios. Incorpora um grande número de morph, mais propriamente 1170 e uma biblioteca com todas as topologias existentes [57], [66], [67]. Pela desvantagem que o Make Human tinha com a quantidade de roupa que continha, este software foi eliminado, dando assim a oportunidade ao Fuse.

### 2.3.2. CAPTURA DE MOVIMENTO (MOCAP)

A captura de movimentos não foi algo que surgiu há pouco tempo, muito pelo contrário. É fácil cometer o erro de assumir que para poder fazer uma captura de movimentos é necessário um computador, mas, na verdade, a captura de movimentos já surgiu muito antes desta tecnologia estar disponível (Imagem 20). Em esta seção será explicado como surgiu a captura de movimento, a sua evolução e quais são os tipos de capturas existem.

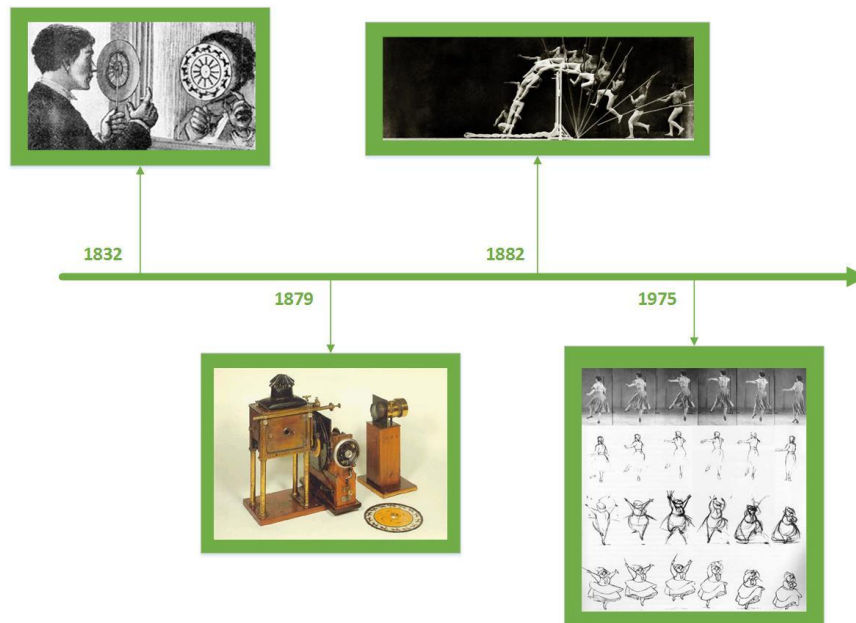


Imagem 20 – Evolução da captura de movimento. 1832 – *Phenakistoscope*; 1879 – Prova de Eadweard Muybridge (Equipamento *Zoopraxiscópico*); 1882 – Etienne – Jules Marey (Fatos MoCap); 1915 – Rostocopia.

#### 2.3.2.1. EVOLUÇÃO DA CAPTURA DE MOVIMENTOS ANTES DOS COMPUTADORES

Recuando até 1832, em Bruxelas, Joseph Plateau, ao inventar o *Phenakistoscope*, tornou-se na primeira pessoa a demonstrar a ilusão de uma imagem em movimento [69]. Este aparelho consiste em dois discos montados sobre o mesmo eixo (Imagem 21 -A). O primeiro disco, situado na aba do aparelho, contém ranhuras em seu redor e o segundo disco, situado no centro, contém uma série de desenhos sucessivos de um movimento que irá servir de base para a ilusão. O truque, para visualizar o movimento das imagens do segundo disco do *Phenakistoscope*, é ver, com a ajuda de um espelho, através das ranhuras existentes no primeiro disco enquanto este gira [70].

Eadweard Muybridge, em 1879, à conta de uma aposta sobre se um cavalo a trotar levantava as quatro patas do solo, conseguiu provar a veracidade deste facto recorrendo à captura de movimento deste animal, recorrendo a cerca de uma dúzia de câmaras, que efetivamente o cavalo levantava as quatro patas do solo simultaneamente, enquanto trotava. Inspirado no *Phenakistoscope*, o equipamento que Muybridge inventou e utilizou para provar, o *Zoopraxiscópico* (Imagem 21 - C), projetava imagens sequenciais do movimento do cavalo num disco numa rápida sucessão (Imagem 21 - B) [71], [72].



Imagem 21 – (A) *Phenakistoscope* [70]; (B) Prova de *Eadweard Muybrige* em 1879 [72]; (C) – *Zoopraxiscópio* [73].

Marey, um fisiologista, conheceu Muybrige em Paris em 1882 e, inspirado por este, inventou uma câmara que capturava imagens para uma tira de papel, introduzindo assim a utilização de rolos de filme para as imagens com movimento. Estas imagens eram capturadas em sequência com recurso a apenas uma câmara. Marey, numa das capturas de imagens que realizou, utilizou um fato com algumas marcas laterais a acompanhar os membros superiores e inferiores, como também uma marca na cabeça (Imagem 22 - A). Com este fato e utilizando a câmara que inventou, realizando um conjunto de fotos sequenciais, conseguiu extrair o movimento da pessoa em movimento numa perspetiva lateral (Imagem 22 - B) [74]. Esta foi a primeira abordagem aos fatos de *MoCap* que foi introduzida por Etienne-Jules Marey em 1884, tendo sido este também a criação da base de cena da nossa versão de *MoCap* contemporâneo [74].

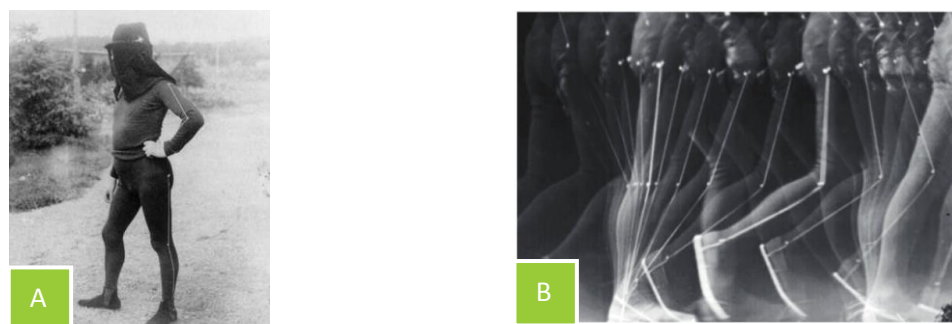


Imagem 22 – (A) Fato com marcas laterais e na cabeça, Etienne-Jules Marey , 1884 [72]; (B) Captura de movimento de andamento com o fato de Etienne-Jules Marey [74].

Alguns anos mais tarde, em 1915, Max Fleischer, um editor de arte, desenvolveu uma técnica que consistia na realização de um desenho animado manualmente. Esta animação era desenhada à mão *frame a frame*. A esta técnica foi dada o nome de *Rotoscopia (Rotoscoping)*. Nesta altura os filmes eram gravados a 16 *frames* por segundo fazendo com que fosse necessário que o animador tivesse que criar 16 desenhos por cada segundo de animação. A técnica utilizada passou por arranjar um aparelho que permitisse ao animador projetar um filme para uma superfície de vidro. Isto permitia a visualização do filme de forma a que o animador pudesse desenhar a sua animação por cima do que era projetado. Desta forma, o animador, conseguia utilizar a projeção como uma espécie de “guia” para os seus desenhos (Imagem 23 - A) [75]. A primeira série animada, da autoria de Max Fleischer, a utilizar esta técnica foi lançada no ano 1919 e foi intitulada de *Out of the Inkwell* (Imagem 23 – B e C) [75].

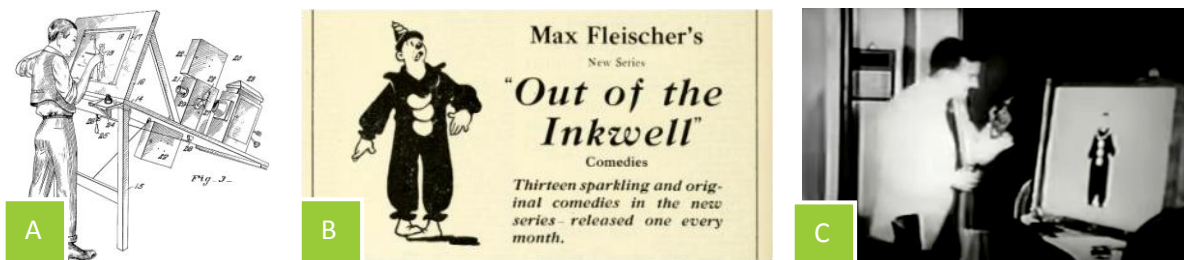


Imagem 23 – (A) Rotoscope [75]; (B) Max Fleischer's e o palhaço Koko [75]; (C) A utilização do rotoscópio no palhaço Koko [75].

Com o sucesso desta técnica, não levou muito tempo a que a Disney começasse a explorar também este processo de animação. Apesar de Fleischer ter desenvolvido vários filmes experimentais utilizando esta técnica, a Disney foi a pioneira no uso de rotoscopia como um estudo do movimento humano e animal.

Os filmes “Branca de Neve e os Sete Anões (1937)” e “Bambi (1942)” foram os primeiros grandes filmes lançados pela Disney utilizando a rotoscopia [74], [76].

Um marco importante na captura de movimento foram as fotografias em alta velocidade. Harold Edgerton, apercebeu-se que conseguia observar uma peça de um motor que se encontrava em movimento como se ela estivesse parada, sendo que para isso tinha que combinar a frequência de uma luz intermitente à velocidade da rotação do motor em movimento. Este fenómeno trata-se do efeito estroboscópico e a técnica fotográfica designa-se por técnica estroboscópica (Imagem 24)[72], [77].



Imagem 24 – (A) Um dos primeiros equipamentos fotográficos que usa uma luz intermitente com uma câmara Kodak de 35mm [77]; (B) Exemplo da captura de movimento do serviço do tenista Gussie Moran realizado por Edgerton, 1952 [77].

### 2.3.2.2. EVOLUÇÃO DA CAPTURA DE MOVIMENTO PARA ANIMAÇÕES DE PERSONAGENS DE COMPUTADOR

No final da década de 1960 e na década seguinte, com a evolução tecnológica e o aparecimento dos computadores, iniciaram-se investigações e desenvolvimentos de tecnologia digital de *MoCap* e *Computer-generated imagery* (CGI – Imagens Geradas por Computador). Na área militar, foi desenvolvido um sistema proprietário de captura de movimento com a finalidade de capturar o movimento da cabeça de um piloto para fins de investigação. Este sistema utilizava um transmissor magnético juntamente com sensores que eram colocados no piloto [78], [79]. Focando um pouco mais nas animações geradas por computador, foi no início da década de 80 que estes tipos de animação começaram a ser produzidos.

Em 1982-1983, já com a existência de alguns sistemas óticos de captura de movimento no mercado, tais como os sistemas *Op-Eye* e *Selspot*. Estes sistemas, começaram a convergir de forma a interligar os sistemas de captura de movimento juntamente com a comunidade de computação gráfica. Esta junção teve início a partir de uma parceria entre o *Massachusetts Institute of Technology (MIT) Architecture Machine Group* e o *New York Institute of Technology Computer Graphics Lab*, os quais realizaram experiências para conseguir capturar o movimento do corpo humano. Os marcadores óticos eram marcadores pequenos colocados no corpo, tipicamente díodo emissores de luz (*LEDs*) ou pontos refletores, juntamente com duas ou mais câmaras que capturavam o movimento de um ator (Imagem

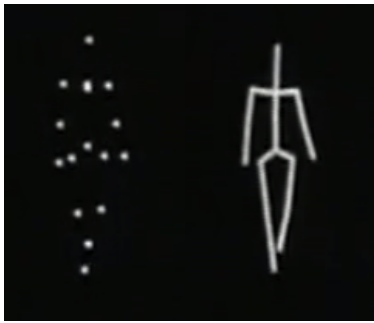


Imagem 25 - Pontos processados computacionalmente com ajuda de LEDs durante uma captura de movimento e o *Stick Figure* criada computacionalmente com recurso aos dados analisados [148]

25). Nesta altura, uma das grandes dificuldades era conseguir arranjar um sistema com resolução suficiente para conseguir captar os marcadores colocados no corpo, LEDs ou pontos refletores, que estivessem bastante juntos. Nesta altura, quanto mais resolução era necessária, mais a câmara iria custar. Em 1983 no *MIT*, Ginsberg e Maxwell apresentaram um sistema baseado nesta tecnologia chamado de *Graphical Marionette*. Este sistema tinha como base a captura ótica de movimentos *Op-Eye* com recurso a um fato com vários LEDs nas articulações e noutras partes do corpo. O computador usava a informação da posição do ator, captada por duas câmaras, compilava a informação do movimento em tempo real e espelhava estes movimentos numa *stick figure* no computador [80].

Ao longo dos anos, foram também surgindo alguns *softwares* de modelação 3D proprietários que eram desenvolvidos para um projeto em específico e não se tornavam em versões comerciais. O *software* DP3D, desenvolvido por Larry Yeager, foi um exemplo de um *software* criado para um projeto em particular, tendo sido utilizado no filme *The Last Starfighter* (1984) [81]. A primeira utilização a nível de produção foi num anúncio da *Superbowl*, em 1985 com o robô *Brilliance* de Robert Abel (Imagem 26)[82]. O método de captura utilizado para realizar os movimentos do robô consistiu em pintar pontos pretos nas articulações de uma modelo feminina e fotografá-la em movimento numa cadeira giratória [72]. Nesta altura, o tempo de renderização de 8 pontos, captados por 4 câmaras num período de 3 segundos, era de 17 horas [83].

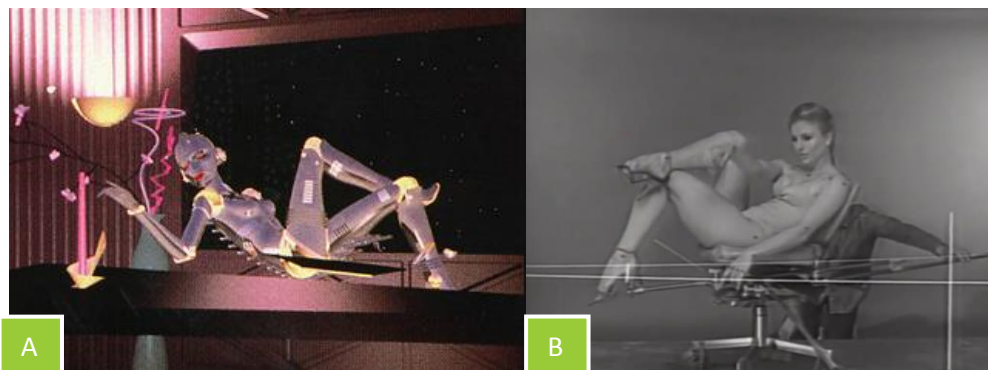


Imagem 26 - (A) Robô Brilliance; (B) Atriz por detrás do robô.

O primeiro *software* desenvolvido e comercializado para animações 3D nos computadores surgiu pela empresa *Wavefront Technologies* em 1985. Este *software* chamava-se *PreView* e foi projetado para ser utilizado nas estações de trabalho da *Silicon Graphics* [84], [85]. Os computadores pessoais ainda

não possuíam poder de processamento suficiente para conseguir processar e renderizar as animações e por isso era necessário recorrer a supercomputadores para realizar este trabalho [72], [77].

Em 1988, foi utilizada uma técnica de captura de movimentos para animar uma personagem 3D com um baixo nível de detalhe. Esta personagem chamava-se *Waldo C. Graphic* (Imagem 27- A). Este tipo de animação era possível ser realizado em tempo real e utilizava um equipamento semelhante a um braço mecânico para poder controlar a personagem (Imagem 27 - B). Este equipamento possuía uns acessórios que permitiam o controlo da posição e movimentos da boca da personagem virtual [80].

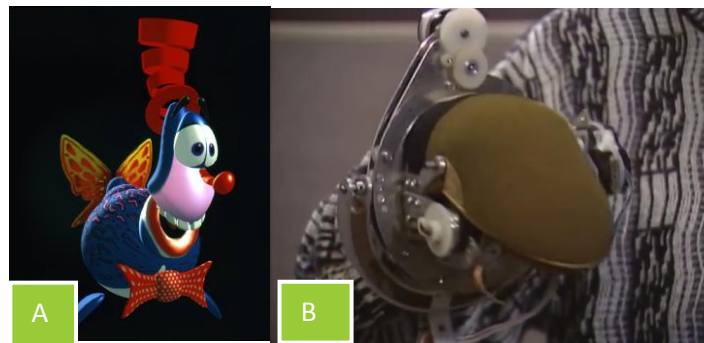


Imagem 27 - (A) Waldo C. Graphic [19]; (B) Equipamento de captura de movimento [86]

Também em 1988, com o avanço tecnológico existente na altura, nomeadamente as estações de trabalho *Silicon Graphics 4D series workstation*, foi possível realizar proezas no âmbito da animação 3D em tempo real que nunca tinham sido feitas em público. Brad deGraff e Michael Wahrman, criaram uma demonstração para um dos seus modelos 4D. Esta demonstração foi denominada com o nome *Mike the Talking Head* e foi apresentada no evento *SIF-GRAPH '88* (Imagem 28). Tratava-se de uma cabeça em 3D que era animada em tempo real por um ator com ajuda de um sistema de captura de movimentos faciais. Pela primeira vez, uma personagem animada foi capaz de interagir em tempo real com uma plateia de pessoas [80],[87].



Imagem 28 - Mike the Talking Head [88].

Mais tarde, em 1992, a empresa *SimGraphics* desenvolveu um sistema de captura de movimento facial chamado *Face Waldo*. Através de sensores mecânicos colocados em várias zonas da face do ator (queixo, lábios, bochechas e sobrancelhas) e sensores eletromagnéticos colocados na estrutura do capacete, era possível copiar em tempo real, com mais detalhe, os movimentos faciais e criar personagens animadas. Um dos maiores sucessos do sistema *Face Waldo* foi a representação de um ator, no qual realizou o papel da mítica personagem da Nintendo, Mário (Imagem 29) [80].



Imagem 29 - (A) Ator com o capacete de captura de movimentos a representar a personagem Mário; (B) Output final com a personagem Mário [89].

Em 1996, Tom Molet, R. Boulic e D. Thalmann propuseram um método eficiente de captura de movimento humano baseado numa tecnologia com sensor magnético (Imagem 30). Estes dados capturados eram convertidos, em tempo real, e processados para uma personagem animada. Esta tecnologia foi utilizada numa vasta quantidade de aplicações, desde teleconferências até ao conceito de animação comportamental [90].



Imagem 30 - Captura de movimento a imitar um jogador de futebol [91].

Três anos mais tarde, em 1999, as mesmas pessoas melhoraram esta tecnologia de forma a incluir um controlo de várias articulações com apenas um sensor, desde que as articulações fossem dependentes umas das outras (isto é, quando a variação de uma articulação também implica variação de uma outra articulação). Com isto, foi possível obter capturas mais realistas dos movimentos conseguindo manter o mesmo custo a nível de número de sensores utilizados [90].

Desde estes esforços pioneiros na captura de movimentos, esta área tem sofrido vários avanços de forma a conseguir fornecer dados às diversas aplicações que requerem captura de movimentos. Nos anos mais recentes, são várias as áreas que requerem este tipo de tecnologias, desde as áreas de medicina, militar, desporto e entretenimento [72].

### 2.3.2.3. TIPOS DE CAPTURA DE MOVIMENTOS

Atualmente são várias as tecnologias que permitem a captura de movimentos e, de forma a conseguir diferenciar, foram separados em três grupos principais, captura ótica, mecânica e magnética. Nestes três grupos podemos encontrar várias tecnologias diferentes, desde tecnologias com recurso a marcadores, sem marcadores às capturas magnéticas e até a capturas de movimento com recurso a fatos e exosqueletos. Das tecnologias sem marcadores, realça-se o sistema da Microsoft, o Kinect [92], que foi o sistema escolhido para a realização deste projeto.

## CAPTURA MAGNÉTICA

---

Os sistemas de captura magnética surgiram nos sensores colocados nos capacetes dos pilotos de aviões, de forma a conseguir monitorizar a posição e orientação da cabeça do piloto. Neste tipo de sistema, são utilizados 12 a 20 sensores de movimento, que são utilizados para medir a relação espacial em relação a um transmissor magnético. Sendo que os sistemas magnéticos de captura de movimentos são sistemas com aquisição direta de dados, estes não necessitam de qualquer tipo de pós-processamento [93]. Este facto faz com seja possível utilizar este tipo de captura de movimentos em aplicações em tempo real. A principal vantagem destes sistemas em relação aos sistemas de captura ótica, é que não necessitam de estar à vista para que o movimento seja gravado. No entanto, estes sistemas são propensos a interferências magnéticas ou elétricas causadas por objetos metálicos ou eletrónicos no espaço onde se realizaram as capturas de movimento [72].

## CAPTURA MECÂNICA

---

Os sistemas de captura de movimentos mecânica funcionam com recurso a exosqueletos. Estes exosqueletos são um equipamento colocado no ator e é capaz de medir os movimentos realizados pelo ator com recurso a acelerómetros. Este tipo de sistema é imune a interferências magnéticas ou elétricas e também é um sistema facilmente transportável [72]. Sendo que é um sistema que utiliza acelerómetros para capturar o movimento, isto induz um problema no que toca aos movimentos num espaço. Se o ator saltar ou subir umas escadas, os dados capturados não mostrarão isso. Os dados capturados simulam o movimento do ator como se este nunca tivesse saído do chão. Para combater isto, sensores magnéticos são normalmente adicionados aos sistemas mecânicos de captura de movimentos para corrigir este problema [72]. Os movimentos possíveis de serem executados são limitados, pois o volume do exosqueleto poderá impedir certos movimentos que envolvam rolar no chão, por exemplo [72].

## CAPTURA ÓTICA

---

Os sistemas de captura de movimentos ótico, por sua vez, estão subdivididos em dois grupos principais. Por um lado, temos os sistemas que recorrem à colocação de marcadores (*Marker-Based*) nos atores ou nos objetos nos quais serão alvo de captura de movimento e, por outro lado, temos os sistemas que são livres de marcadores, normalmente chamados de sistemas *Markerless* [94], [95].

### COLOCAÇÃO DE MARCAS PARA DETERMINAR PONTOS ESPECÍFICOS DO CORPO

---

Tipicamente, um sistema ótico de captura de movimentos com recurso a marcadores utiliza entre 4 a 32 câmaras juntamente com um computador pessoal (PC) que controla as respetivas câmaras. Neste tipo de captura de movimentos, podem ser abordadas duas vertentes no que toca à escolha dos sensores. Temos a captura de movimentos com recurso a refletores, denominada de captura de movimentos passiva, e a captura de movimentos com recurso a emissores de luz, denominadas de captura de movimentos ativa [72].

Na captura ótica de movimentos ativa, os marcadores são *LEDs*. Existem alguns métodos diferentes para poder identificar os *LEDs* deste tipo de sistemas. Alguns sistemas iluminam um *LED* de cada vez, podendo desta forma conseguir identificar cada marcador. Outros sistemas iluminam todos os *LEDs*

ao mesmo tempo e, de forma a identificar cada *LED*, alteram a frequência ou amplitude de cada LED permitindo assim a fácil identificação por parte do equipamento recetor [72].

Na captura de movimentos passiva, os marcadores refletores possuem normalmente um formato esférico, semiesférico ou circular e são feitos com base num material refletor [72].



Imagem 31 - Câmera Vicon equipada com LEDs para captura ótica de movimentos [72]

Os tamanhos e formatos podem variar dependendo do objetivo da captura de movimento, sendo que são normalmente utilizados marcadores mais pequenos para capturas de movimento faciais ou capturas do movimento das mãos. Estes marcadores são colocados diretamente na pele do ator ou então colocados num fato próprio para o efeito, sendo normalmente feito de materiais com elasticidade, tal como o *spandex* (também conhecido por elastano) [72]. As câmaras utilizadas neste tipo de captura são equipadas com LEDs de forma a que a luz emitida seja refletida pelos marcadores (Imagem 31) [72].

Estes métodos de captura com marcadores têm alguns problemas. Um deles é a limitação no movimento do ator, causada pela colocação de um marcador num certo lugar que acaba por dificultar a realização de algum movimento. Outro problema com a utilização dos marcadores refletores é que é necessário um ambiente controlado para conseguir uma captura de movimentos sem interferência. Mesmo com estes problemas esta é a técnica utilizada mais comum e tem muita popularidade [94][91].

#### CAPTURA SEM MARCAS - MARKERLESS

---

A captura de movimentos sem qualquer tipo de marcador, também chamada de captura *markerless*, tem vindo a ser uma importante área de investigação. Neste tipo de captura de movimentos, não é necessário vestir nenhum fato. Considerando que não existe nenhum tipo de sensores junto ao corpo, há uma maior dificuldade em capturar o movimento de um ator ou um objeto e conseguir uma melhor recolha de dados. Para ultrapassar estas limitações, são utilizadas múltiplas câmaras de forma a melhorar a captura de movimento. O lançamento da Microsoft Kinect com os sensores de profundidade, revolucionou esta área pois facilitou a tarefa da captura de movimento sem que fossem necessárias várias câmaras, diminuindo assim os custos em equipamentos [94].

#### 2.3.2.4. KINECT

---

O Kinect é um dispositivo lançado pela Microsoft para ser utilizado juntamente com a Xbox360 em 2010, permitindo ao utilizadores uma interação com a consola através de captura de movimentos, gestos e comandos de voz, sem necessidade de comandos de jogo, até para ligar a Xbox basta dizer “Xbox on” e estamos pronto a jogar. É um sensor de movimento desenvolvido principalmente para Xbox 360 e Xbox One. O Kinect tem como objetivo aumentar a utilização da Xbox 360 [96][97]. Este foi de facto o maior lançamento da Xbox 360 [98].

Este dispositivo incorpora uma câmara RGB para capturar imagens e reconhecimento faciais, microfones para reconhecimento de voz e detetar onde o utilizador se encontra e sensores de infravermelhos para detenção de profundidade (Imagem 32) [99]. A Microsoft apresentou uma nova câmara com sensores de movimento Kinect para acompanhar a consola Xbox. A Kinect 2 optou por

realizar algumas melhorias importantes. A Kinect não permite torções musculares, o que provoca um pouco de irrealismo nas animações, visto que os seres humanos podem realizar esses movimentos.

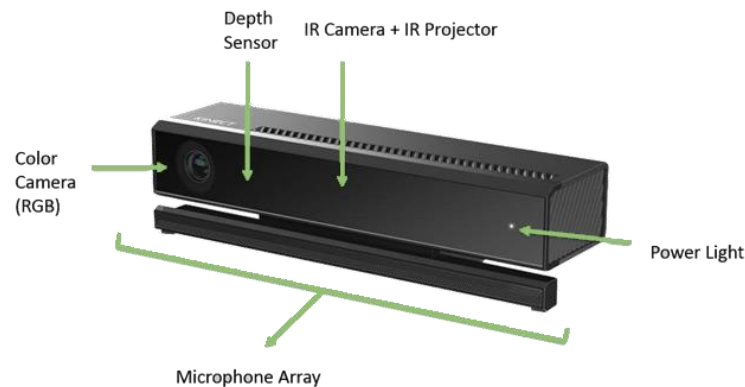


Imagem 32 - Sensores da Kinect 2 da Windows [99].

## EVOLUÇÃO DA KINECT

“Projeto Natal” como inicialmente era conhecida o Kinect, foi lançada em junho de 2009. Através de uma parceria entre a empresa Israelita *PrimeSense* e a Microsoft da qual nasceu o sensor Kinect para a Xbox 360 em novembro de 2010. Em julho de 2011 foi lançado um *software development kit* (SDK) pela Microsoft, mas não foi comercializada. Só em fevereiro de 2012 o SDK foi comercializado para Windows, no mesmo ano foram lançadas duas versões diferentes da Kinect a 1.5 e a 1.6. Em novembro de 2013, a Microsoft lançou a Kinect 2 para a sua nova consola, a Xbox One [97], [100], [101].

## KINECT 1 VS. KINECT 2?

Escolher o hardware adequado para o projeto em questão é muito importante, por isso, temos de analisar as vantagens e desvantagens de cada um e certificar-nos que não fazemos a escolha errada. Na Tabela 2 podemos ver algumas especificações importantes sobre cada uma das versões da Kinect (v1 vs. v2), de maneira a facilitar a escolha do hardware que virá a ser utilizado no projeto.

Tabela 2 - Especificações da Kinect1 e Kinect2 [102] .

Característica	Kinect 1	Kinect 2
<b>Câmara</b>	640 x 480 @ 30 fps	1920 x 1080 @ 30 fps
<b>Câmara de profundidade</b>	320 x 240	512 x 424
<b>Câmara de infravermelhos</b>	Nenhuma	512 x 424
<b>Distância máxima profundidade</b>	~4 metros	~4,5 metros
<b>Tipo de luz</b>	Light Coding	ToF
<b>Distância mínima profundidade</b>	80 cm (40 cm em modo near) -> Modo “perto”	50cm
<b>Campo de visão horizontal</b>	57 graus	70 graus
<b>Campo de visão vertical</b>	43 graus	60 graus
<b>Microfone</b>	4-mic array a 16 kHz	4-mic array a 48 kHz

<b>Motor Tilt</b>	Motorizado	Manuel
<b>Articulações Esqueleto</b>	20 articulações	25 articulações
<b>Total de esqueletos monitorizados</b>	2	6
<b>Seguimento da mão</b>	Ferramentas Externas	Sim
<b>Seguimento da cara</b>	Sim	Sim + Expressões
<b>Tipo de USB</b>	2.0	3.0
<b>Sistema Operativo Suportado</b>	Win7, Win8	Win8-8.1 (WSA)
<b>Preço (sensor + adaptador)</b>	~160€	~ 200€ [103]

Apesar de a Kinect 2 ser mais cara, as suas vantagens são impressionantes. Logo, a melhor escolha será a Kinect 2. A maior vantagem da Kinect 2 em relação a Kinect 1 é a qualidade dos seus sensores, o que permite uma captura com muito maior qualidade e detetar a localização e o movimento de pessoas e as suas vozes. Destes sensores, destacam-se os seguintes:

- **Cor** - Incorporada com uma câmara de 1080p RGB, proporciona uma entrada estável.
- **Profundidade** - Com uma maior fidelidade e capacidade para detetar objetos mais pequenos e com muita mais clareza e estabilidade para rastreio de pessoas. Mede a diferença de tempo em que a luz é emitida e refletida para poder obter um valor de profundidade.
- **Infravermelhos** - Com valor de intensidade de 16 bits, permite que o sensor veja ao escuro. Podendo também agora utilizar o IR e a cor ao mesmo tempo.
- **Rastreio de corpo** (esqueleto, seguimentos da mão, inclinação) - Com a fidelidade do sensor de profundidade, juntamente com as melhorias que se tem vindo a ter, agora podemos rastrear até 6 esqueletos, o que corresponde a 25 articulações por pessoa. Assim criar personagens mais realistas, precisos e nítidos.
- **Áudio** - Incorporado com 4 microfones, para gravação de áudio, mas também localizar a origem do som [2], [102]

## ESQUELETOS E CORPOS V1 VS. V2

Como foi referido no ponto anterior, a Kinect 2 trouxe melhorias no rastreio dos esqueletos. Na comparação da Imagem 33-A e Imagem 33 - B, podemos notar um aumento de articulações. Adicionando à Kinect 2, articulações nas extremidades das mãos, polegar e ombro. Outra das atualizações que houve na Kinect 2 foi a captura da mão, esta deteta se a mão está aberta ou fechada e a que lado pertence (direita ou esquerda) [99].

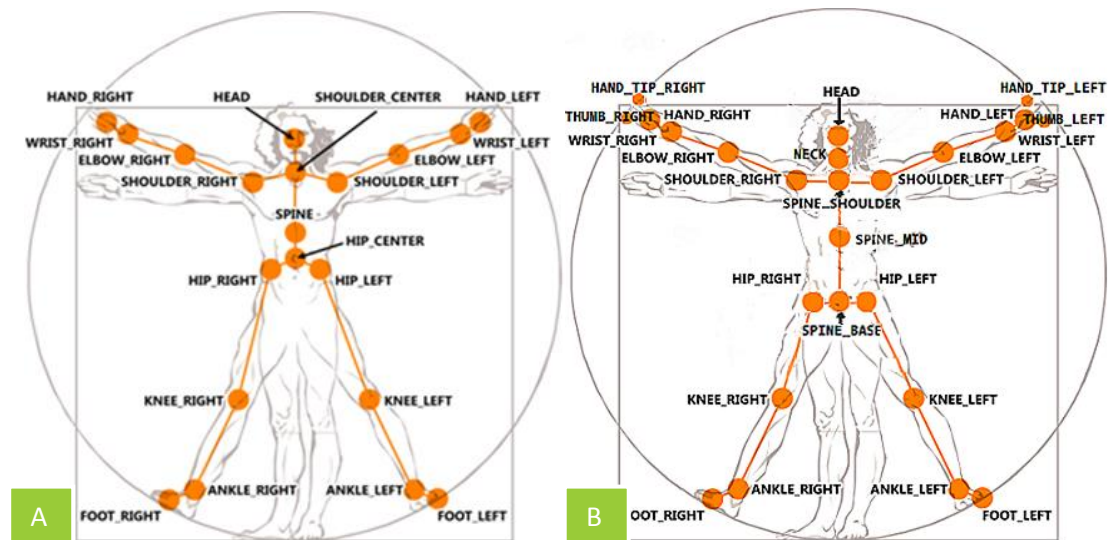


Imagem 33 - (A) Esqueleto detetado pela Kinect1; (B) Esqueleto detetado pela Kinect2

## CARACTERÍSTICAS DE ROSTO

A Kinect não serve só para o corpo, mas também para o rosto. Com uma resolução 20 vezes superior, garantindo mais de 1000 pontos na representação do rosto de uma pessoa [104]. Tal como no corpo, onde a Kinect deteta se as mãos estão fechadas ou não, na cara é capaz de detetar também várias atividades, como, se os olhos estão abertos ou fechados, se olhar foi desviado, se a boca está aberta ou fechada ou se realiza algum movimento de deslocação. Juntamente com isso capta expressões faciais (Feliz, neutro, etc.) [99], [105].

## USOS DA KINECT NA INDÚSTRIA

Uma vez que faz parte do âmbito desta tese demonstrar como é possível realizar um *Cinematic 3D* com o auxílio de captura de movimentos através da Kinect, nada melhor como mostrar alguns dos trabalhos existentes que usaram este sensor. Nesta secção irá ser apresentada informações sobre trabalhos realizados que interagem com a Kinect, podendo assim dar uma ideia do que é possível fazer com este dispositivo. Cada vez mais, a Kinect tem vindo a ganhar popularidade em várias áreas e tem-se vindo a demonstrar o potencial das suas capacidades.

## MEDICINA

Na Medicina já encontramos algumas soluções inovadoras que utilizam a Kinect.



Imagem 34 - Utilização da Kinect na medicina [149].

Alguns exemplos disso são as salas de cirurgias com recurso à Kinect (Imagem 34). Estas salas de cirurgia são salas onde os médicos e enfermeiros obtêm mais autonomia através de reconhecimento de gestos conseguem aceder a vários ficheiros de pacientes, exames, entre outros sem ter nunca de tocar fisicamente em algo. A este método de auxílio, é adicionado uma maior segurança e menor risco de infeções para os pacientes que se encontram num ambiente esterilizado [106], [107].

Um outro exemplo, ainda em estudo, que poderá mudar o futuro na medicina utilizando a Kinect é melhorar a eficácia dos Raios-X (Imagem 35) juntamente com uma diminuição da radiação emitida aos pacientes. Alguns investigadores da Escola de Medicina da Universidade de Washington, adaptaram um sistema, utilizando a Kinect, que consegue colmatar esta necessidade [108].

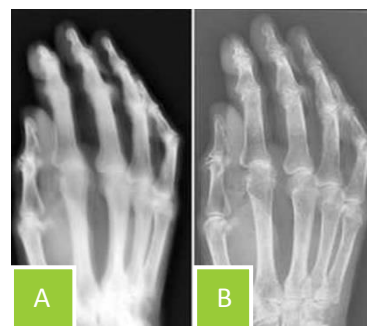


Imagem 35 - Melhoramento do raio-X da mão. (A) Raio-X; (B) Kinect. [150].

Ainda na área da medicina, a Kinect também pode ser utilizada como forma de reabilitação com pacientes que tenham sofrido vários tipos de trauma (Imagem 36). De notar, que segundo uma sondagem realizada em 2014 [109], cerca de 65% dos pacientes, submetidos a uma reabilitação após um acidente vascular cerebral (AVC), falha em seguir o programa delineado. Para combater estes resultados foi necessário pensar numa abordagem um pouco diferente na realização dos programas de reabilitação.

Tomando como exemplo este tipo de pacientes, que tenham sofrido um AVC, foi desenvolvido um



Imagem 36 - Reabilitação utilizando a Kinect [110].

*software* onde os pacientes, interagindo com a Kinect, têm como objetivo realizar certos movimentos de forma a conseguir recuperar e obter total mobilidade dos seus membros superiores e inferiores. Utilizando a câmara tridimensional da Kinect, é possível capturar os movimentos do paciente que está a ser alvo da terapia. Desta forma é possível que o sistema consiga interpretar os dados obtidos, medindo e avaliando o processo do paciente através de vários exercícios disponibilizados. Com este sistema é

possível medir a coordenação dos pacientes, a sua destreza nos movimentos, habilidades motoras, reflexos e tempos de reação [110]. Além da reabilitação em pacientes que tenham sofrido AVCs, a Kinect também é utilizada noutros tipos de reabilitação, tal como na reabilitação de pacientes com paralisia cerebral. Foi notado que as crianças com paralisia cerebral tendem a ter dificuldades em estender os seus membros, tais como os pulsos, cotovelos e ombros. A reabilitação nestes casos é

muitas vezes realizada com práticas repetitivas de movimentos que por vezes tornam-se aborrecidas e podendo ser desmotivadoras para os pacientes. O uso da Kinect nestes casos entra para tornar esta reabilitação menos monótona e mais divertida, levando o paciente a fazer os mesmos movimentos, mas desta vez com recurso de jogos ou atividades com a ajuda da Kinect de forma a que este consiga manter a motivação durante todo o processo de reabilitação [111].

Além da reabilitação, a Kinect também pode ser utilizada como forma de monitorização. Dando o exemplo de pacientes que sofram da doença de *Parkinson*, estes têm que realizar consultas de rotina junto de um médico fisioterapeuta, uma ou mais vezes por ano. Com recurso a uma aplicação (Imagem 37) baseada na Kinect, é possível fazer uma monitorização remota aos pacientes de uma forma contínua. A monitorização consiste na captura de várias ações por parte do paciente, sendo algumas delas as frequências do toque físico entre o dedo polegar e o indicador, agilidade e movimentos com os pés e movimentos alternados com as mãos. Após as ações por parte dos pacientes, todos os dados que foram capturados são processados e analisados para depois serem disponibilizados aos médicos fisioterapeutas responsáveis. Desta forma é possível que estes médicos consigam acompanhar de uma forma mais regular o paciente em questão podendo até fazer alterações na medicação caso seja necessário [112].

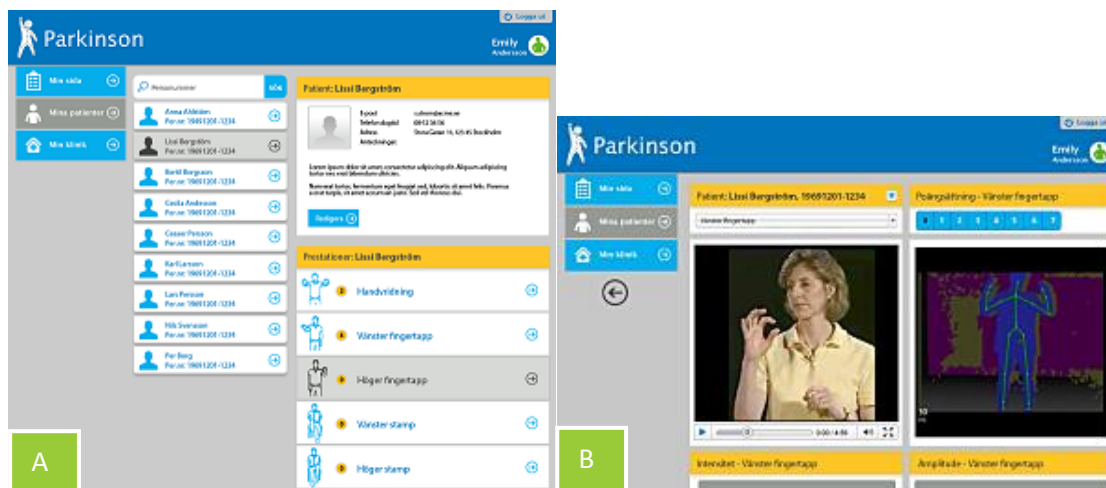


Imagem 37 – (A) Interface do projeto que ajuda ao medico a saber o estado do paciente; (B) Interface do projeto que mede as habilidades de toques do paciente através da Kinect [112].

## ENSINO

Uma das grandes barreiras na comunicação por vezes, é a incapacidade de algumas pessoas poderem



Imagem 38 - Projeto Kinect Sign Language Translator [113].

falar. Nestas situações, estes locutores recorrem a linguagem gestual. Quando o segundo locutor não compreende linguagem gestual, torna a comunicação quase impossível. Para combater este obstáculo linguístico, através da Kinect, foi possível desenvolver um sistema que permite servir de tradutor entre dois locutores numa situação semelhante (Imagem 38). Através de gestos de linguagem gestual, o sistema reconhece, lê e traduz para linguagem falada e vice-versa [113].

Ainda no ensino, passando para dentro das salas de aula, alguns professores acharam uma boa opção juntar a tecnologia atual aos métodos de ensino já existentes. Tendo como objetivo dar uma maior



Imagem 39 - Jogo educativo que utiliza a Kinect. In *Body and Brain Connection* [115].

imersão no ensino dado aos alunos, a opção de incorporar a Kinect nas salas de aula mostrou-se bastante recetiva pelos alunos, tornando, de uma maneira criativa, a aprendizagem mais eficiente e mais divertida. Através de jogos educativos (Imagem 39), em várias disciplinas, foi possível fazer com que os alunos tirassem mais dúvidas sobre o que estavam a aprender aumentando assim a motivação em aprender[114], [115].

## ROBÓTICA

Uma das áreas também em grande crescimento atualmente é a robótica. Com uma necessidade crescente de automatizar as coisas, os robôs têm vindo a aumentar. Para que um robô consiga ter alguma autonomia é sempre necessário que este consiga “ler” o ambiente que o rodeia. Para isso nada melhor que o uso de sensores.

Os sensores da Kinect permitem que esta necessidade seja cumprida. A própria Microsoft desenvolveu o *Robotics Developer Studio* (Imagem 40) que permite projetar e construir robôs com base na Kinect [116].



Imagem 40 - *Robotics Developer Studio* [116].

## COMERCIAL

Passando à área comercial, também esta é beneficiada com a utilização da Kinect.



Imagem 41 -- *Interactive Solution* [117].

Tomando como exemplo o produto *Kinect Interactive Solution* (Imagem 41), criado pela *IntuiLab*, este permite a iteração com um catálogo comercial apresentado num ecrã ao cliente. Este é um sistema que é ativado com recurso à proximidade entre o cliente e o ecrã. Através de gestos, o cliente poderá navegar no catálogo e obter informações sobre os produtos que lhe interessam [117].

## RETALHO

Na área das vendas e retalho, cada vez mais as lojas procuram maneiras de atrair clientes, sejam elas com promoções, com algum tipo de animação e/ou por vezes com soluções inovadoras.

Assim que o potencial da Kinect foi notado, além das várias soluções que já vimos anteriormente, na área do retalho foram também criadas soluções inovadoras pensadas com o objetivo de cativar e facilitar as compras em vários centros e lojas comerciais [118].

Em particular, uma solução virada para os clientes de sinalização digital possibilita ao cliente experimentar vários tipos de vestuário que se encontra à venda naquele estabelecimento comercial sem ter o trabalho de se dirigir a um provador e experimentar efetivamente aquela peça de roupa.



Imagem 42 - Virtual Style Sense [119].

Através do sensor de profundidade da Kinect juntamente com um *software* de reconhecimento demográfico, é possível mostrar a partir de um ecrã o próprio cliente com a peça de roupa vestida. Esta solução, o *Virtual Style Sense* (VSS) (Imagem 42), foi trazida para o mercado pela Zugara com o objetivo de aumentar o espaço nas lojas para expor o seu inventário, fazer com que os clientes usufruam de uma melhor experiência personalizada, entre outros [119].

As soluções com Kinect não passam apenas por facilitar e ajudar os clientes. As próprias lojas também podem beneficiar com este sistema, com soluções de recolha e análise de dados, podendo analisar o comportamento dos clientes nas suas próprias lojas. Um produto que foi apresentado na *Expo National Retail Federation* (NRF) em 2015, facultava precisamente este tipo de análises. Com o nome de *Path Tracker* (Imagem 43), esta tecnologia utiliza os sensores da Kinect V2 de forma a conseguir obter, as rotas que os clientes efetuam dentro da loja.

Os sensores ficam montados a uma altura de forma que seja possível abranger toda a área de circulação de clientes da loja. À medida que os clientes vão circulando dentro da loja, os sensores vão capturando os movimentos e vão registrando a afluência dos vários clientes aos vários caminhos/áreas da loja. Desta forma é possível obter um mapa térmico com a atividade dos clientes naquele espaço. Com esta informação, os lojistas conseguem saber em que áreas os clientes têm mais interesse/passam mais tempo e podem utilizar essa informação de forma a poder otimizar as vendas [118].

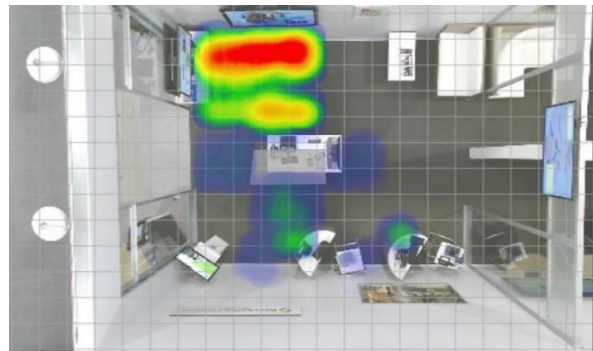


Imagem 43 - Path Tracker [151].

Com esta informação, os lojistas conseguem saber em que áreas os clientes têm mais interesse/passam mais tempo e podem utilizar essa informação de forma a poder otimizar as vendas [118].

## 2.4. CONCLUSÃO

Toda a pesquisa realizada para produzir o estado da arte ajudou-nos a tomar escolhas importantes para a elaboração deste projeto. Conhecer todo o processo de produção da animação 3D foi o primeiro passo para definirmos uma *pipeline* que se ajustasse ao que pretendíamos realizar e delinear todos os passos que tínhamos de fazer. Só assim teríamos um bom resultado após a produção do nosso filme final.

Saber quais as vantagens, desvantagens e funcionalidades de alguns dos *softwares* desta área também tornou a escolha um pouco mais fácil, contudo, em alguns casos, ficámos na dúvida de qual seria o melhor, tendo sido necessário realizar testes de forma a poder avaliar um pouco mais as funcionalidades do software, tomando como exemplo os *softwares* MakeHumen e o Fuse. O resto dos *softwares* que foram escolhidos foram devido às suas vantagens e funcionalidades. O 3ds Max foi o

escolhido para ser o protagonista do projeto, pois é nele onde serão modelados todos os objetos em 3D e onde realizaremos toda a montagem das cenas. O Zbrush é onde daremos todos os detalhes aos nossos objetos 3D e no Photoshop e Substance Painter é onde daremos as cores a esses modelos. O Maya é um software ótimo para criar efeitos especiais e, por isso, escolhemos este software para criar as cenas que contenham mais efeitos.

Por último escolhemos o Brekel e o MotionBuilder para ser realizada a captura de movimento. A compatibilidade entre os dois *softwares* é transparente e a vantagem do Brekel é que se trata de um software muito simples, facilitando a fase da captura de animação, que só por si é um processo complexo e demorado.

## PARTE II - DESENVOLVIMENTO



### 3. METODOLOGIA E DESENVOLVIMENTO

Depois de saber como uma empresa de criação de vídeos 3D funciona, foi necessário criar uma *pipeline* que se ajustasse ao nosso projeto. Baseada na *pipeline* (Imagem 44), o desenvolvimento foi dividido em 3 partes: Pré-produção, Produção e Pós-produção.

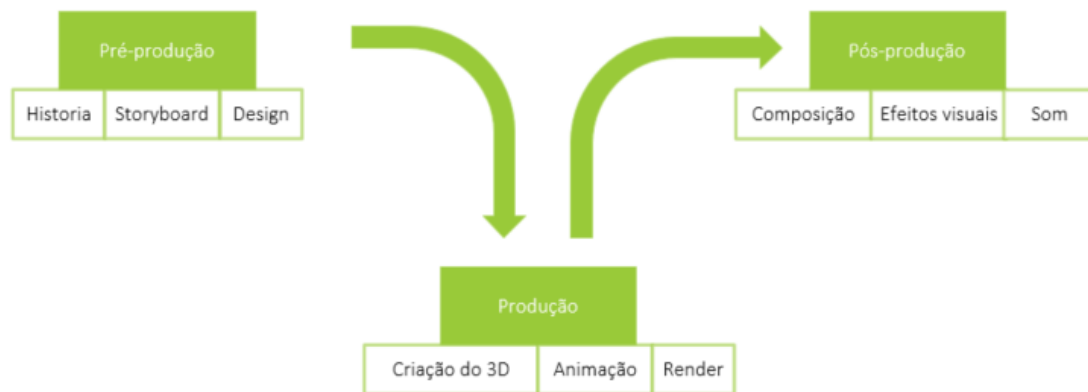


Imagem 44 - Pipeline de desenvolvimento

O desenvolvimento foi realizado seguindo esta *pipeline* e delineando uma linha de produção para criação do produto final.

#### 3.1. PRÉ-PRODUÇÃO

Esta fase consiste na realização dos planos iniciais do projeto. É a parte mais indispensável, pois é onde são definidos os conceitos essenciais para poder mostrar e descrever todo o trabalho. Esta parte foi subdividida em quatro (Imagem 45): História, *Storyboard*, Guião e Design.



Imagem 45 - Pipeline de pré-produção

##### 3.1.1. HISTÓRIA

A ideia para este projeto surgiu de um projeto de mestrado onde foi criado um jogo de estratégia online. O objetivo deste projeto é a produção de um *trailer* para o jogo existente utilizando os recursos que estão ao alcance do público comum e com custo reduzido, sem ter que recorrer a equipamentos e tecnologias utilizadas pelas grandes empresas.

O enredo do jogo é baseado na história de dois povos de duas galáxias diferentes, a galáxia vermelha e a azul.

Na galáxia vermelha, os colonos são de uma raça maquiavélica e tecnologicamente superiores, cujas intenções são meramente destruidoras. Esta raça tem como principal objetivo a destruição de todos

os planetas da galáxia azul e a captura do maior número de recursos existentes nessa galáxia, que são essenciais para manter a vida deste povo maquiavélico.

É no momento em que as galáxias se cruzam, a cada 50000 anos, que esta raça destruidora executa o seu plano, plano este que consiste no envio de um dos seus melhores espiões, Nifast, para o planeta Theris da galáxia azul. Este espião tem como missão obter informações sobre qual a melhor forma de invadir e dominar aquele planeta. Um fator que é desconhecido desta raça invasora é que, na galáxia azul, existem poderosos e misteriosos satélites, os quais eram utilizados para treinar os guerreiros nas mais variadas disciplinas de combate.

Chegado o momento, Nifast parte na missão de descoberta em direção ao planeta Theris, um planeta com uma civilização antiga e em decadência. Ao fazer a aproximação ao planeta, surge uma avaria o que faz a nave despenhar-se no meio de uma floresta nesse planeta. Depois da aproximação desastrosa ao planeta, as coisas continuam a não acontecer como tinha sido planeado. Nifast acaba por se apaixonar por uma habitante daquele planeta, Isene, e em vez de recolher informações sobre a melhor maneira de invadir aquele planeta, começa a criar uma família e uma vida naquele local, mas sempre com o receio em mente de que aquele planeta e as pessoas que lá habitam pudessem sofrer um ataque da galáxia vermelha.

O espião não se deixou ficar por aqui, tomou medidas preventivas para proteger aquela galáxia, galáxia essa que já tinha desenvolvido um vasto carinho e amor, e onde agora habitava com a sua família e amigos. Ele chamou os melhores guerreiros das mais variadas disciplinas de combate (Feiticeiros, Curandeiros, Guerreiros, Arqueiros e Guardiões) que se encontravam nos misteriosos satélites, para serem os mestres dos clãs que iriam ser criados para defender o planeta, ensinando e treinando os novos recrutas de cada clã e tornando-os capazes de defender a galáxia azul da invasão.

Um explorador intergaláctico de uma civilização avançada vai visitar uma civilização antiga em decadência com o propósito de preparar uma invasão para recolher recursos importantes para eles. Mas ao chegar a essa civilização ele apaixona-se por uma rapariga. Como preparação para o dia da invasão ele decide recuperar os desaparecidos clãs de guerreiros da civilização e os treinar reativando os portais secretos dos clãs.

---

### 3.1.2. STORYBOARD

Uma das grandes preocupações foi certificarmo-nos que é escolhida a melhor forma de mostrar e descrever o que foi idealizado na história. É necessário ter em conta vários fatores, começando pelo correto posicionamento das câmaras. Para isto, uma das formas de controlar esta preocupação é realizar um *storyboard*, cujo objetivo é conseguir mostrar a história visualmente através de desenhos simples (miniaturas) em papel que mostram o seu desenvolvimento. Estes desenhos têm ainda em conta o posicionamento das câmaras escolhido para cada uma das cenas.

Na elaboração do *storyboard* para este projeto foram selecionados os pontos mais marcantes da história que deu a criação do jogo, de maneira a que os jogadores percebam um pouco os objetivos do jogo em questão.

No decorrer da realização do *storyboard*, até chegar a versão final que se encontra no anexo 7.2, foram realizadas várias modificações, de forma a melhorar a coerência e a fluência da história. Adicionámos uma previsão de tempo relativa a cada miniatura, uma pequena descrição e em alguns casos que tipo

de transição seria realizada. Cada miniatura foi desenhada tendo em conta a posição em que a câmara iria se encontrar e como se iria deslocar.

A história foi dividida em oito blocos essenciais. Qualquer alteração feita, a história tinha de rodar à volta destes pontos. 1 - Missão; 2 - Acidente; 3 - Apaixona-se; 4 - Mensagem de ataque; 5 - Revelação da verdade; 6 - Entrega do amuleto; 7 - Abrir portais; 8 - Guerreiros saem dos portais prontos para treinar e defender o planeta.

Uma miniatura é dividida em duas partes, a miniatura onde é desenhado a cena em questão (Imagem 46 - A) e um quadro de informação, onde podemos encontrar qual é a cena que pertence, onde é elaborado o cenário, o tempo que demora e que tipo de transição vai ser efetuada (Imagem 46 - B). Na miniatura da Imagem 46, podemos ver a cidade onde a camera seguira o ângulo desenhado na miniatura quando percorre a cidade. Outra informação que podemos ver é o deslocamento da personagem e a direção que ira tomar.

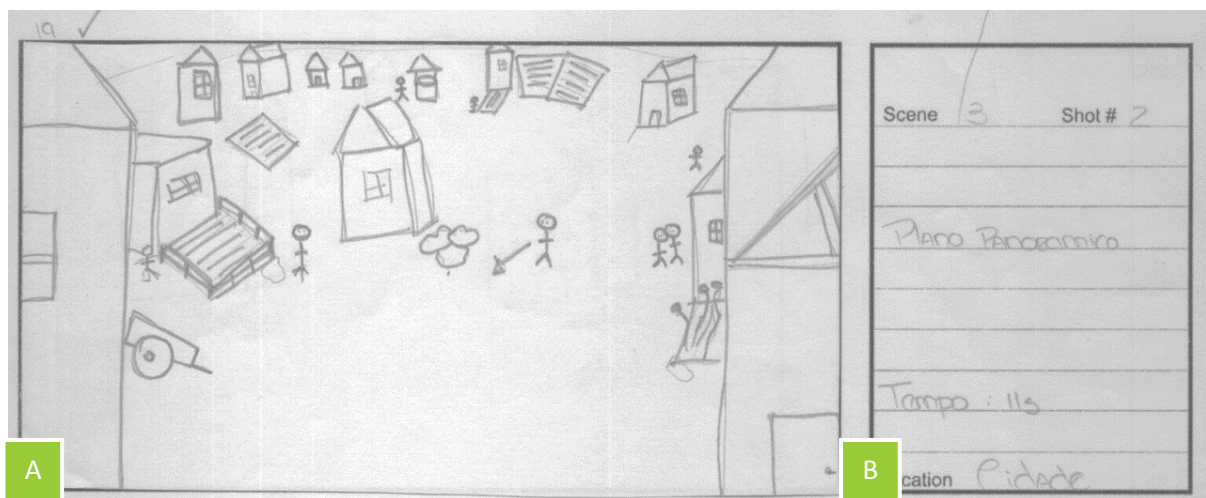


Imagem 46 – Miniatura do *storyboard* da cidade. (A) Campo onde é elaborado o desenho da cena; (B) Informação sobre a cena.

Na Imagem 47 podemos ver duas miniaturas, na qual as duas pertencem à mesma cena (cena dos portais). Na Imagem 47 – A podemos ver a abertura dos portais no templo e o brilho que é refletido por cada um deles. Quando todos os portais estão abertos, passamos para a próxima miniatura (Imagem 47 - B) onde sai cada guerreiro do seu respetivo portal e estes são apresentados.

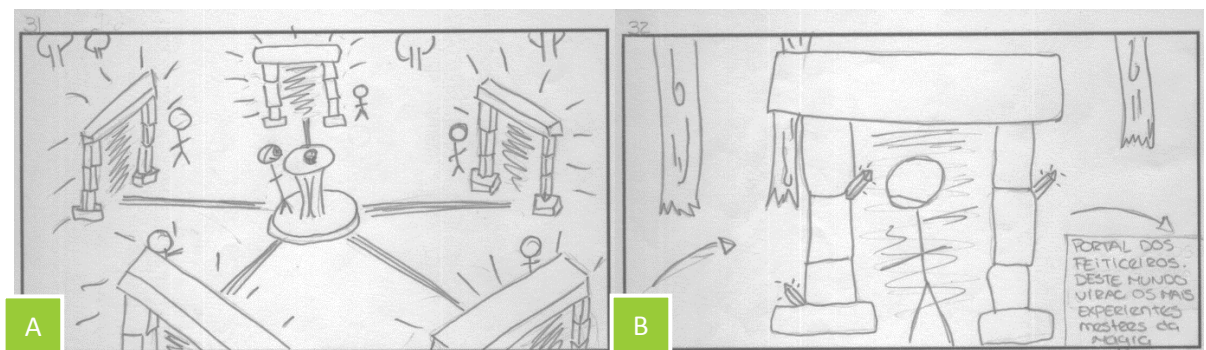


Imagem 47 – Miniaturas do *storyboard* no instante em que o templo de ilumina e se ativa os portais. (A) Abertura dos portais; (B) Guerreiro sai do portal.

### 3.1.3. DESIGN

Nesta parte foi realizada toda a pesquisa para podermos concretizar o design necessário à execução dos cenários. Esta fase é importante pois contribui para termos uma noção de todos os elementos que podemos vir a necessitar, bem como contibui para a desição relativa ao processo de design que devemos seguir. No decorrer deste trabalho falaremos no método de criação do portal dos curandeiros de maneira a poder explicar todo o seu processo e os passos seguidos.

Na história existem cinco mundos completamente diferentes. Cada um desses mundos está ligado a um portal mágico que se encontra no planeta Theris. Para cada satélite, ou mundo, foi necessário delinear as suas características, o conceito, as cores, as particularidades das personagens e as suas armas.

Na Tabela 3 estão descritas todas as características que foram tidas em conta para a criação de cada satélite e para criação de cada portal. Esta tabela foi a que deu inspiração aos criadores do jogo Keepers of Intheris [120] para criar as personagens do jogo. Com a informação recolhida da tabela, conseguimos juntar imagens que serviram de inspiração na hora de idealizar o nosso projeto.

Tabela 3 - Conceitos de cada satélite e dos seus guerreiros [120].

Satélite	Conceito	Mundo	Cores	Características	Armas
<b>Assassinos</b>	Assassin's Creed[121], Indiana Jones	Ruínas azteca, lianas, armadilha	Castanho e verde (predomina)	Sorrateiros, ágeis e espias	Navalhas, cordas
<b>Inventores</b>	Steampunk, BioShock	Tecnologia da era vitoriana, <i>steampunk</i> , vapor e dirigíveis	Ferrugem e cinza metálico	Atacam ao longe utilizando a tecnologia	Armas de fogo e arcos
<b>Feiticeiros</b>	Dalaran – WOW [122], elf cities e sky cities	Torres flutuantes, cristais mágicos e criaturas místicas	Lilas e azul	Com a magia arcana, são os mais poderosos e ao mesmo tempo os mais frágeis pela sua pouca defesa	Bastões mágicos
<b>Protetores</b>	Jurassic Park e Yeti - WOW [123]	Montanhas com neve, vales, castelos gigante e criaturas gigantes	Verde, Castanho e Cinza	São os mais corajosos, são os únicos que se defendem e protegem todos	Escudos
<b>Curandeiro</b>	Floresta proibida (Harry Potter) e a Maléfica	Vida: Floresta e vida animal	Cores vivas	Manipuladores da força vital e decidem quem vive e quem morre	Poções
		Morte: Trevas, esqueletos, espinhos	Cores escuras		

Conforme a informação da tabela fomos pesquisando imagens para obter inspiração para a criação dos nossos mundos, personagens e acessórios.

O mundo dos curandeiros é um mundo dividido pela vida e pelas trevas. No lado da vida tudo é bonito e cheio de vida, o verde expande-se de maneira a propagar a vida e a tranquilidade dos que lá vivem. Por outro lado, temos o lado das trevas, onde tudo é muito sombrio e onde a morte se instalou, deixando rastros de esqueletos e secando toda a flora à sua volta. O contraste das cores entre os dois lados é enorme, o que ajuda a revelar os poderes dos guerreiros que aí se encontram (Imagem 48).

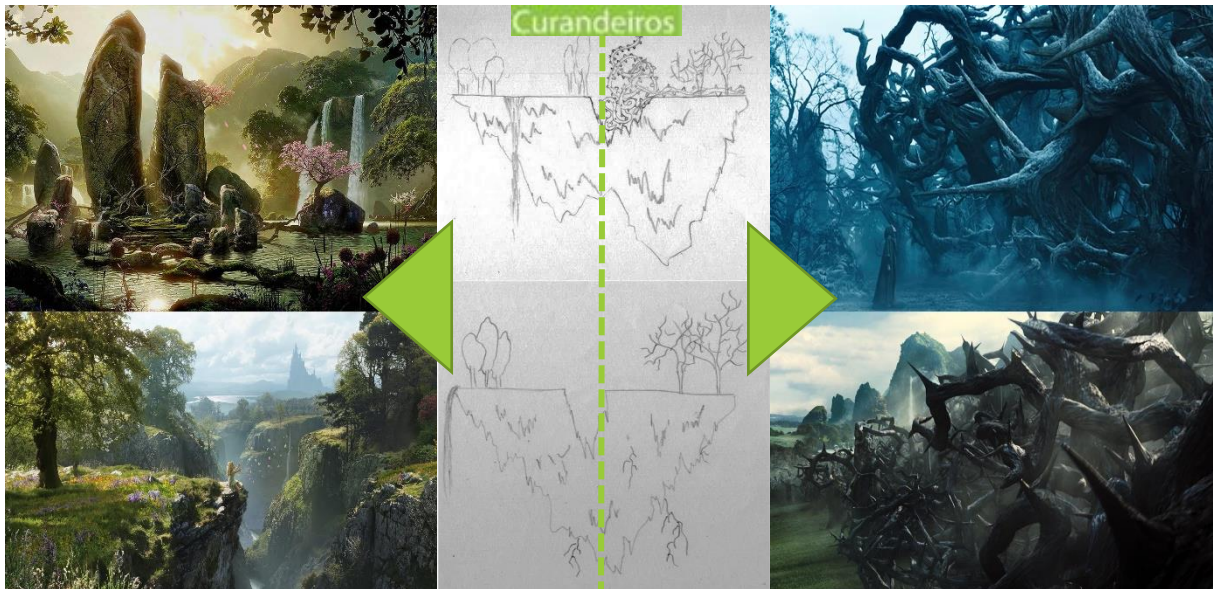


Imagem 48 – Inspiração e desenhos do mundo dos curandeiros.

As personagens deste mundo possuem o poder da manipulação da vida, decidindo quem morre e quem vive através das suas grandes variedades de poções (Imagem 49 -A). A criação deste mundo foi inspirada na floresta proibida (*Harry Potter*)[124] e na floresta do filme *Maleficent* [125].

No que toca às personagens, inicialmente também foram realizadas algumas pesquisas e obtidas algumas ideias para criar e modelar as personagens. No entanto, no decorrer do projeto, tendo em conta todo o trabalho de modelação e tempo que seria necessário despendido para a criação de cada uma delas, decidimos elaborar as personagens num software mais simples, o qual não possui material suficiente para implementar as nossas ideias e inspiração (Imagem 49 - B).

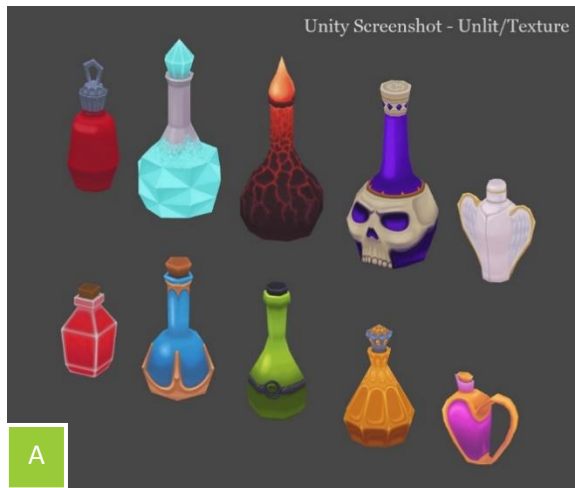


Imagem 49 -Inspirações. (A) Poções mágicas dos curandeiros; (B) Guerreiros curandeiros.

Através da tabela anterior e da pesquisa de imagens, fomos obtendo inspiração para criarmos todos os nossos mundos e objetos.

O mundo dos assassinos é um mundo repleto de ruínas aztecas, onde a mãe natureza tomou conta de tudo o que lá existia. Num mundo com abismos, a existência de muitas lianas ajuda os habitantes deste mundo a ultrapassarem todas as armadilhas e a decifram todos os puzzles que lá existem. A inspiração para a criação deste mundo surgiu através do jogo *Assassin's Creed* [126] e dos filmes *Indiana Jones* [127] e *A Múmia* [128]. O castanho e o verde são as cores que mais representam este mundo (Imagem 147 - anexo). As pessoas que lá vivem, normalmente são espiões ágeis e sorrateiros com armas pequenas, consistindo na maioria em navalhas e cordas, de forma a que a agilidade destes habitantes não seja reduzida pelo tamanho das armas que transportam (Imagem 148).

O mundo dos inventores é um mundo completamente tecnológico. Baseado na era vitoriana e no *steampunk*, está repleto de rodas dentadas, dirigíveis e muitas chaminés com fumo, o cinza é a cor que mais se familiariza com este mundo (Imagem 149). Todas as pessoas que aqui vivem atacam de longe e normalmente utilizam muita tecnologia, o que os leva a construir as suas próprias armas, através da tecnologia que conhecem, tais como arcos e flechas (Imagem 150).

O mundo dos feiticeiros, é um mundo mágico, cheio de torres flutuantes gigantes, edifícios pontiagudos, muitas casas pequenas e rodeado de magia e cristais. Neste mundo podemos encontrar criaturas mágicas. O mundo dos feiticeiros é um mundo invadido pelas cores roxo e azul e foi inspirado na cidade dos elfos, na cidade dalaran [129], *crystal cities* e *sky cities* (Imagem 151). As pessoas que aqui vivem usufruem da magia arcana através do seu bastão. São extremamente poderosos, mas em contrapartida são os guerreiros mais frágeis (Imagem 152).

O último mundo é o mundo dos protetores. Um mundo onde tudo o que existe é grande, com muitas montanhas, vales e criaturas gigantes, tais como dinossauros ou dragões. Este mundo foi inspirado no *Jurassic Park* [130] e os seus monstros nos *Yetis* do jogo *World Of Warcraft* [123]. Este mundo é caracterizado pelas cores cinza, castanho e verde (Imagem 153).

As pessoas que aqui vivem são extremamente corajosas, podendo defender e proteger qualquer outra pessoa através das suas fortes armas, como a espada e os seus escudos (Imagem 154).

Finalizando esta fase de design, conseguimos ficar com a ideia de como ficariam os nossos objetos 3D e assim poder seguir para a próxima fase com uma ideia concreta.

## 3.2. PRODUÇÃO

A produção deste projeto está dividida em três partes. A criação de todo o 3D existente no projeto, de todo o tipo de animações necessárias para cada cenário e da fase dos renders e definições necessárias para a sua execução.

### 3.2.1. CRIAÇÃO DO 3D

A fase de criação do 3D, foi dividida em três subprocessos: criação do mundo, personagens e montagem de cenas (Imagem 50).

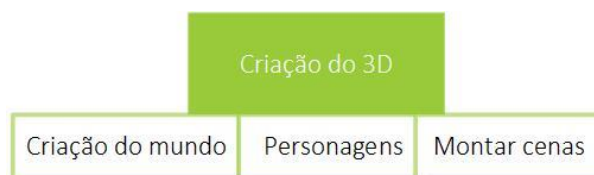


Imagem 50 - Pipeline de criação 3D.

Nesta secção será descrito todo o processo da criação 3D dos mundos, dos seus acessórios e como foram criadas as personagens. Iremos falar também de todos os passos necessários para montar e preparar todos os efeitos visuais 3D, camaras e luzes de cada cena.

#### 3.2.1.1. CRIAÇÃO DO MUNDO

Antes de começarmos com qualquer modelação 3D relacionada com o mundo foi criada uma *pipeline* de produção para a criação dos mundos e dos seus acessórios. No decorrer desta secção serão explicados todos os procedimentos realizados para a elaboração dos acessórios para o mundo (Imagem 51).

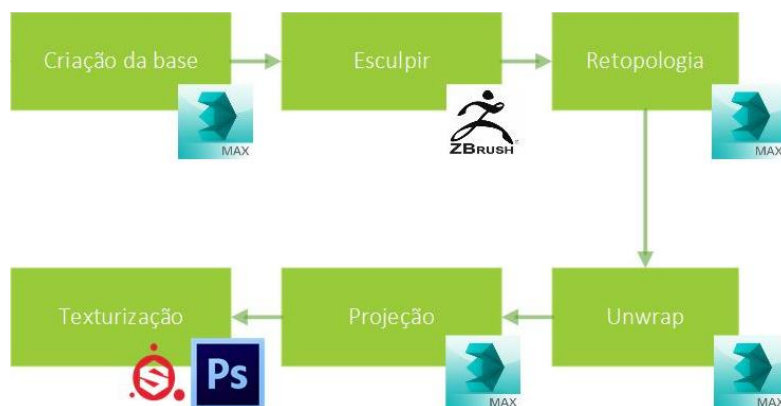


Imagem 51 - Pipeline da criação do cenário e os seus acessórios

#### CRIAÇÃO DA BASE

O processo de criação começa com a criação de uma base no 3ds Max respeitando a topologia. Nesta secção iremos mostrar como foi feito o processo de criação dos portais. Existem cinco tipos de portais

diferentes, mas com uma base semelhantes, logo aproveitámos a base do primeiro portal a ter sido realizado para a elaboração de todos os outros (Imagem 52 - A). A característica inicial para a criação desta base foi pela sua dimensão, de maneira a que todos os portais ficassem com as suas medidas correspondentes. O seu formato surgiu através de uma junção de cubos empilhados de maneira a formar a sua estrutura. No caso de outros modelos, por exemplo a poção (símbolo do portal dos curandeiros), criámos o seu formato através de uma imagem a partir do seu conceito e modelámos o modelo através dela (Imagem 52 - B).

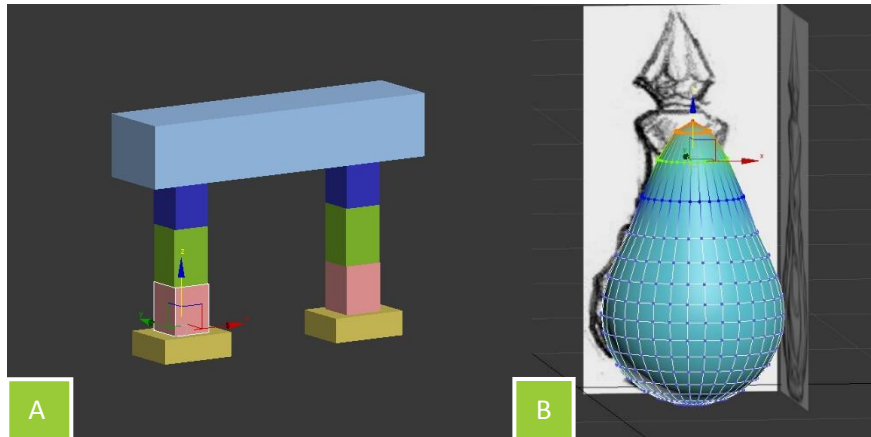


Imagem 52 - (A) Base dos portais modelado no 3ds Max; (B) Base do símbolo do portal dos curandeiros, modelada a partir de uma imagem no 3ds Max

Depois de concluir esta fase, passámos à fase de adição de detalhe. Para isto, foi utilizado o *software Zbrush 4R7*. Embora o 3ds Max também possua capacidades de adicionar detalhes, a escolha recaiu pelo *Zbrush* pelo facto de ser muito mais simples e também por fornecer muitas mais opções de esculpir e adição de detalhes. Poderíamos também criar o modelo base do portal no *Zbrush*, mas devido a uma maior familiaridade com o 3ds Max, optamos por o utilizar. Apesar das bases dos portais serem todas a mesma e o seu processo ser semelhante, cada portal tem características diferentes e foram esculpidos separadamente. Após concluir esta fase, exportámos o ficheiro com formato *.obj*, diretamente para o *Zbrush* (Imagem 53).

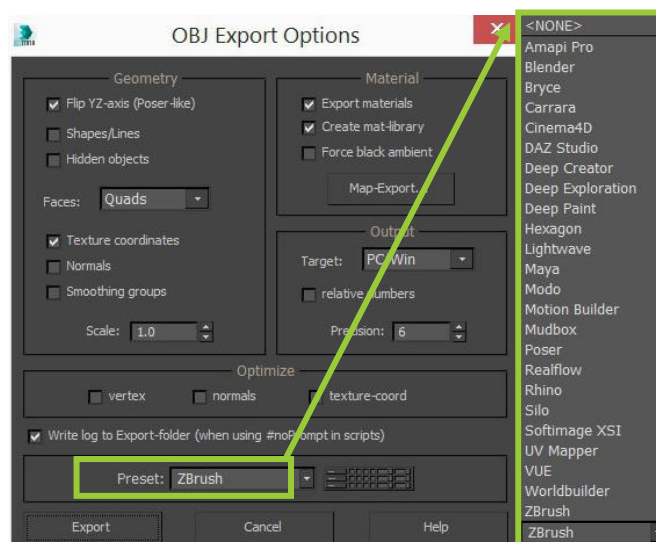


Imagem 53 – Opção de exportação para o Zbrush.

## ESCULPIR

Nesta parte será explicado como esculpir um modelo no Zbrush. O primeiro passo a tomar é a transformação do modelo *low poly* para um modelo *high poly*. Um modelo *high poly* é um modelo com um alto número de polígonos, que nos permite criar todo o tipo de detalhe que desejamos. Por outro lado, um modelo *low poly* é o inverso do *high poly*, este contém um número reduzido de polígonos e é mais difícil aplicar detalhes. Num modelo 3D, os detalhes são o que faz a diferença entre os vários estilos (cartoon, realista, hiper-realista, minimalista, *low poly*, etc).

Para poder esculpir o modelo é necessário conhecer um pouco do que o programa nos oferece, o que podemos fazer com essas ferramentas e se conseguimos realizar os que pretendemos. Felizmente podemos encontrar facilmente informações, tutoriais, uma comunidade e até inúmeras bibliotecas na Internet [131], [132]. Para poder esculpir no Zbrush, de uma forma mais fácil, é essencial utilizar a *pen tablet*, sem este periférico, o processo para esculpir torna-se mais difícil. Com o rato, não conseguimos ter a precisão da força aplicada e, conseqüentemente, ao utilizar é aplicada a força máxima durante a criação da escultura. Outro dos benefícios ao utilizar a *pen tablet* foi a rapidez e eficiência na sua utilização após atingir a familiaridade com o periférico, visto que se assemelha a algo tão natural como desenhar numa folha de papel. Sem isto, o trabalho com recurso a um rato, para fazer pequenos retoques ou até para desenhar é extremamente complicado.

O primeiro passo na criação do portal foi dividir o modelo para permitir a criação de detalhes mais pequenos. A divisão que o Zbrush realiza apenas dividirá o modelo em polígonos. Utilizando o ZRemesher podemos escolher o número de polígonos que o nosso modelo terá sem alterar a forma do modelo. Para podermos dar a textura de pedra ao portal e também as irregularidades da pedra, foram utilizados um conjunto de pincéis. O pincel *move elastic* foi utilizado para modelar o portal e também para dar aquela irregularidade da pedra. Outros vários pincéis, com texturas de pedras, também foram utilizados para dar algumas peculiaridades à pedra, tais como ranhuras e para criar a ilusão de termos pedras partidas em alguns sítios (Imagem 54).

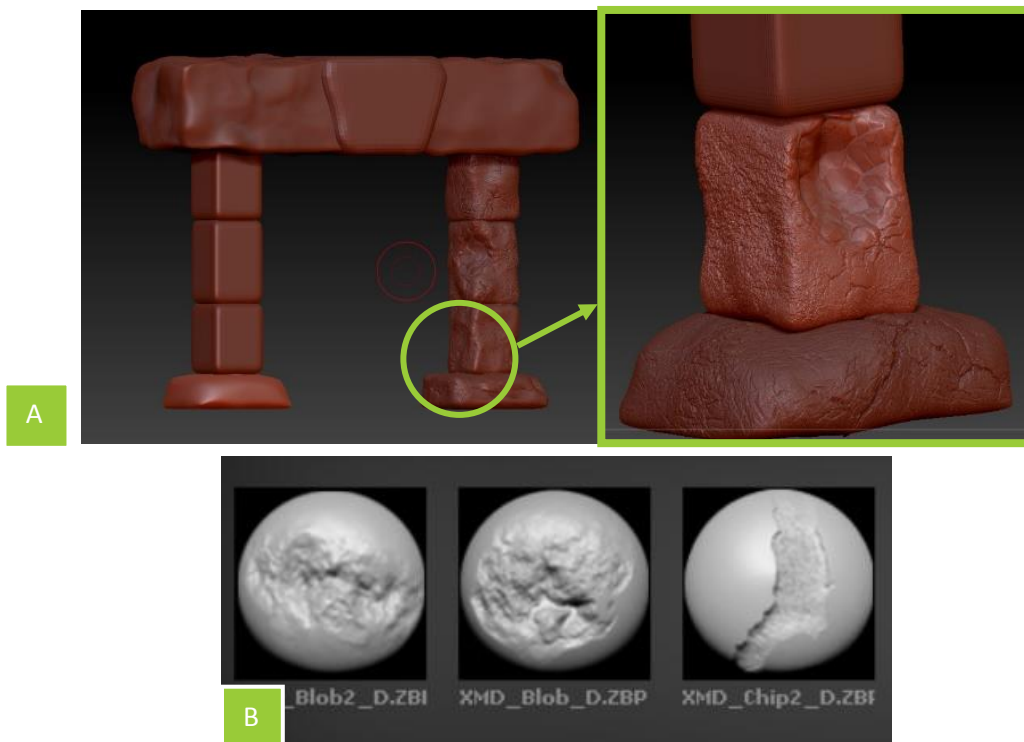


Imagem 54 –(A) Detalhe de pedra esculpido no portal; (B) Pincéis com algumas texturas de ranhuras de pedras que podem ser adicionados ao modelo.

O próximo passo foi a caracterização de cada portal. Com a ajuda de um pincel de criação de corda (Imagem 55), criámos uma corda para colocar em redor dos portais onde desejávamos que estivessem lianas ou lianas com espinhos (inspirado na árvora Acácia Espinhosa).



Imagem 55-Criação da corda do Zbrush

Esta técnica serviu para dois portais, Portal dos Assassinos e dos Curandeiros. Para que este tipo de lianas possuísse um aspeto mais realista utilizamos então três tipos de deformações, o *Inflate Ballon* para colocar os fios da corda um pouco mais grossos, o *Inflate* que separou cada fio da corda e por ultimo o *noise* que foi o que deu um pouco de realismo à liana criando uma deformação. Com isto, ficamos com a liana terminada. No entanto foi ainda necessário realizar algumas alterações de forma a conseguirmos ter uma liana com um especto mais malévolo, isto é, com espinhos. Para isto, utilizamos o pincel *Move Elastic*, que permitiu-nos esticar pequenos pontos da liana e assim formar os

espinhos. Da mesma maneira que foi utilizado um pincel para criar as lianas, também foram utilizados pincéis para criar as folhas e as flores destas mesmas lianas (Imagem 56).

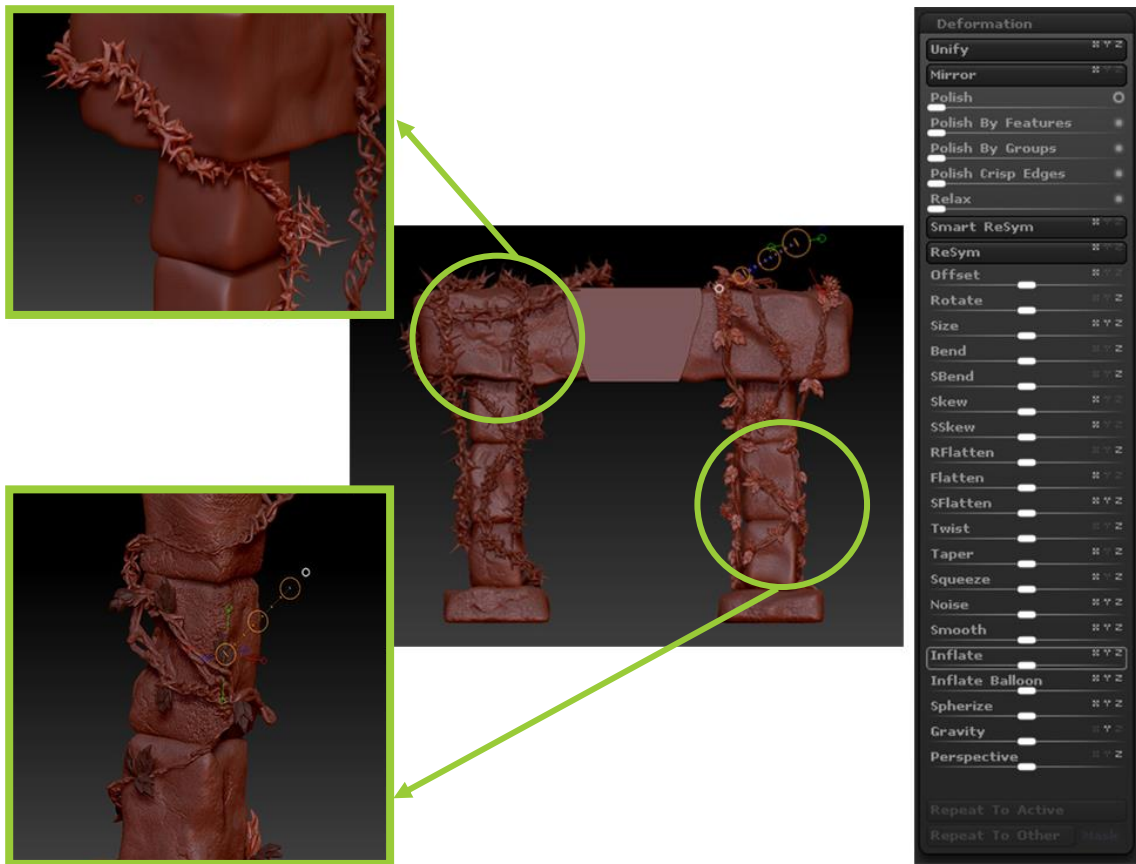


Imagem 56 - Portal dos curandeiros e opções que foram utilizadas.

Uma ferramenta muito útil no *Zbrush* é a máscara que permite selecionar a zona que será ignorada no momento que estamos a esculpir (Imagem 57).

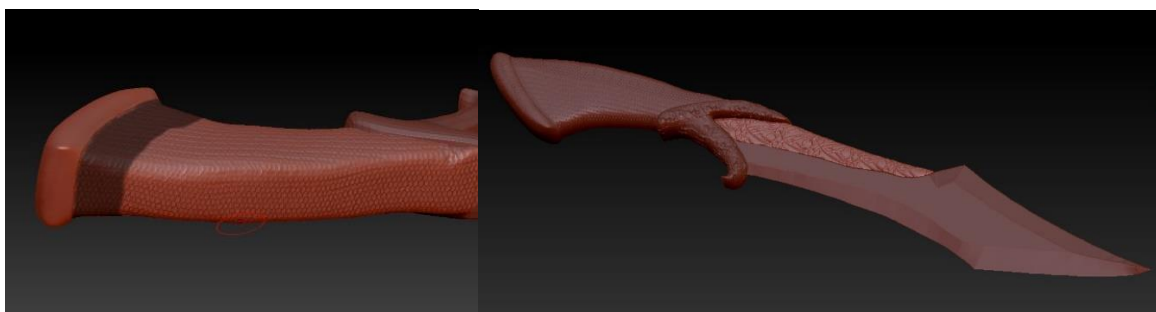


Imagem 57 – Esculpir a navalha utilizando a máscara.

Para a criação do símbolo do portal dos curandeiros (Imagem 58) pegamos na base realizada no 3D's Max e, no *Zbrush*, com o pincel standard juntamente com uma imagem *Alpha* e com a opção *DragRect* ativa, a imagem foi esculpida no modelo. Ao realizarmos esta técnica é necessário ter em atenção se queremos que modelo seja esculpido em apenas um dos lados ou em ambos os lados. Neste caso, para impedir que o modelo fosse esculpido em ambas as faces, ligámos a opção *Backface mask*. A técnica de

esculpir através de uma imagem de referência foi utilizada em muitos modelos, tal como na poção e no chão do templo (Imagem 58).

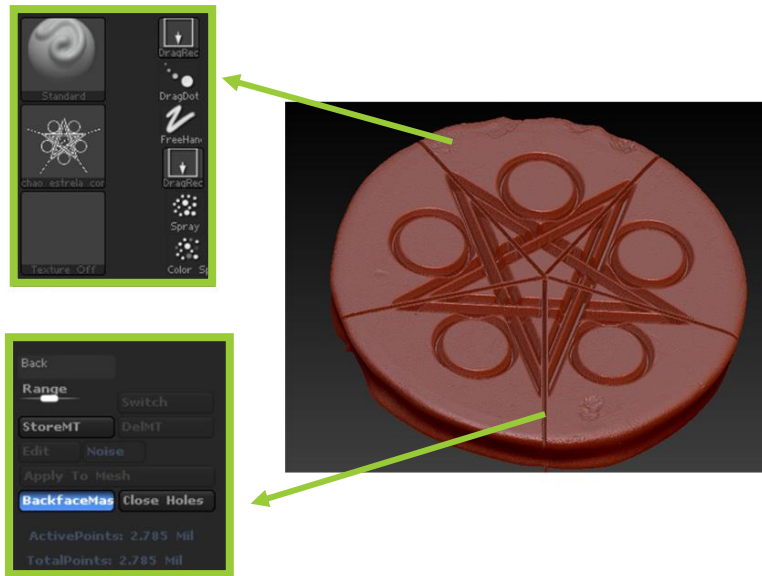


Imagem 58 – Esculpir um modelo com uma imagem *alpha* e utilizando o *Backface mask*.

Na criação do bastão foi adotada uma nova técnica, utilizando o pincel *SliceCurve*, (Imagem 59 - B) o que nos permitiu cortar o modelo em grupos permitindo-nos, desta forma, obter diferentes fitas formando assim o detalhe das cordas à volta do bastão. No entanto, para adicionar detalhes, utilizámos pincéis como *Slash3*, *ClayBuildup*, *Layer*, *Clay* e o *Smooth* que suaviza um pouco a superfície.

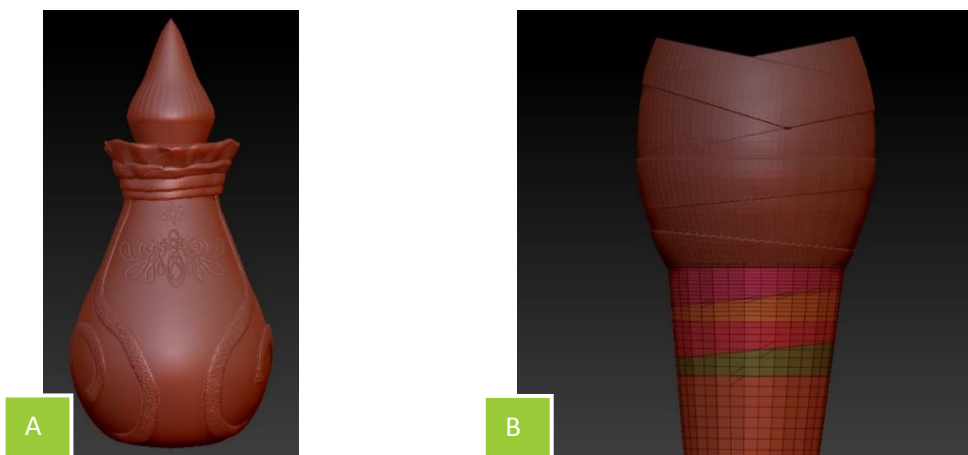


Imagem 59 – (A) Modelo da poção do portal dos curandeiros esculpido; (B) Utilização da técnica *SliceCurve* no bastão.

Depois de adicionamos todos os detalhes aos modelos, é boa prática criar os modelos *low poly* para podermos trabalhar na secção que se segue. Para isso utilizámos o plugin do Zbrush, *Decimation Master* que permite reduzir o número de polígonos sem perder detalhes, mas cria triângulos o que limita o que se pode fazer posteriormente no modelo. Um exemplo do número de polígonos que o *Decimation Master* pode reduzir é num dos blocos do portal, antes de realizar o *Decimation Master* o bloco tinha 1.556 mil polígonos, e depois disso passou a ter 311.286 polígonos. É importante passarmos todos os modelos por esta fase porque ao reduzir o número de polígonos estamos a melhorar a eficiência e rapidez do programa enquanto estivermos a trabalhar nas fases seguintes,

sendo que as que têm mais impacto são a de montagem e de animação. Depois de realizar o passo com o *Decimation Master*, é exportado o ficheiro com a extensão *.obj* para continuarmos para a fase de re-topologia no 3ds Max.

## RE-TOPOLOGIA

---

A re-topologia foi realizada no 3ds Max. Depois de importar o modelo *high poly* do *Zbrush*, dependendo das modificações que foram feitas a nível de silhueta, podemos utilizar a base anteriormente criada ou não. Caso o modelo *high poly* criado difira do modelo base inicial, será necessário criar um novo modelo base com uma forma semelhante ao modelo *high poly*.

Para realizarmos a re-topologia do portal, pegamos na base inicial e sobrepomos ao modelo *high poly* como se pode ver na Imagem 60, para cada um dos blocos do portal. Um ponto importante na re-topologia é que o modelo criado deve ter a mesma silhueta que o modelo *high poly*, de modo a facilitar os próximos passos do processo.

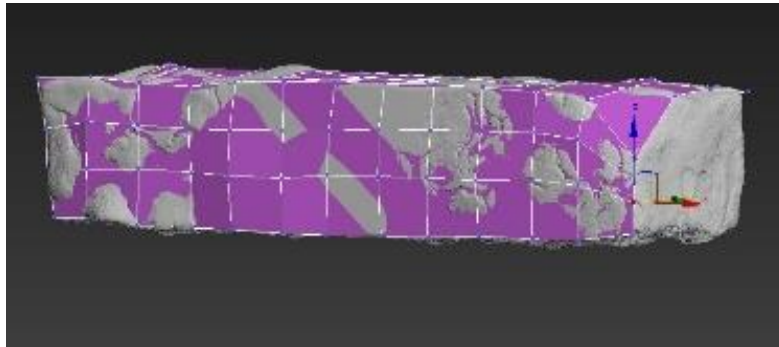


Imagem 60 - Re-topologia do bloco do portal.

## UNWRAP UVW

---

O passo seguinte foi utilizarmos o modificador *Unwrap UVW* para realizarmos o mapeamento dos modelos. Para isso adicionámos o modificador no modelo *low poly* criando assim um mapa para cada bloco do portal. Uma das preocupações que tivemos foi realizar os cortes do mapa em lugares estratégicos (parte de trás do bloco e na junção dos dois blocos) para caso houvesse alguma falha na texturização esta não se notasse.

Este mapeamento pode ser efetuado por dois métodos diferentes. O primeiro, é o manual. No mapeamento manual podemos seleccionar quais são as arestas (*edges*) do objeto que irão abrir de maneira a que o *unwrap* fique bem feito (Imagem 61). Desta forma, conseguimos ter mais controlo sobre toda a superfície do objeto e é a forma mais fiável para realizar um *unwrap*, embora mais trabalhosa. O segundo método para realizar o *unwrap* é usar a forma automática. Neste método, o 3ds Max irá realizar o mapeamento do objeto que estamos a trabalhar de uma forma automática, mas, por vezes, este algoritmo não realiza o *unwrap* da forma mais correta. Em qualquer um dos métodos apresentadas é necessário ter em atenção que nenhum polígono ou vértice se sobreponha, caso contrario o *unwrap* não estará bem feito e causará problemas nos passos seguintes.

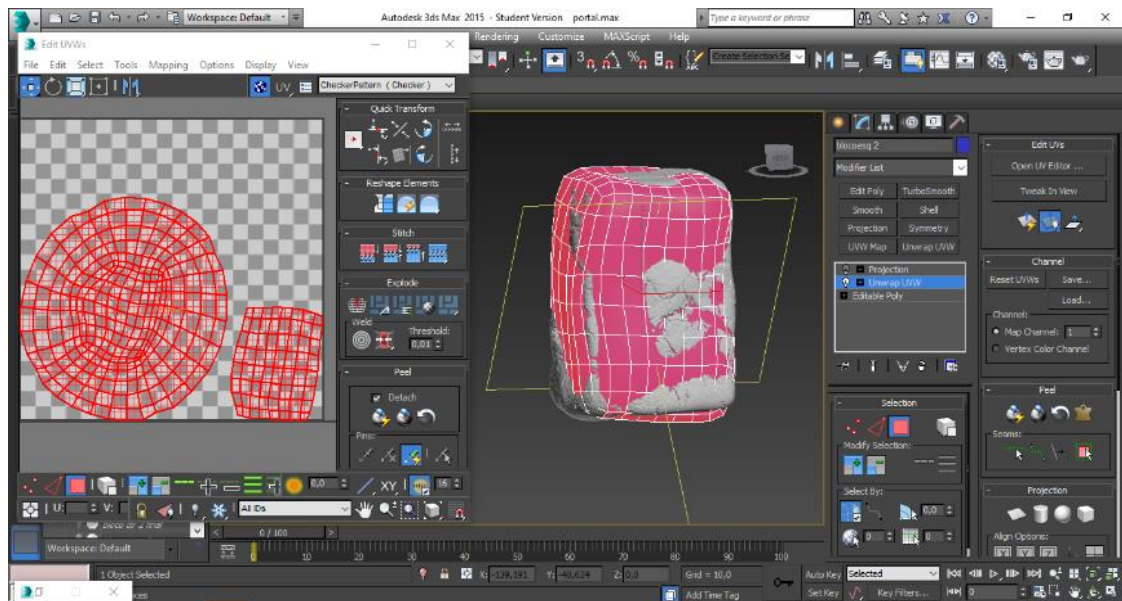


Imagem 61 - Unwrap de um dos blocos que fazem parte do portal.

## PROJEÇÃO

Com o mapeamento do modelo realizado, passamos para a fase de projeção. Nesta fase foram projetados todos os detalhes do modelo *high poly* para o mapa criado no passo anterior do modelo *low poly* criando assim a ilusão de detalhe. Todo este processo foi feito no 3ds Max.

Usamos então o modificador *Projection*. Este modificador permite-nos criar uma rede (*cage*) à volta do modelo (Imagem 62), podendo ajustar esta “rede” de maneira a que todos os polígonos do modelo *high poly* fiquem dentro. Caso existam polígonos fora desta “rede”, a projeção para estes polígonos não se realizará. Com o recurso à opção *Render to Texture*, seleccionámos os mapas que queríamos que fossem produzidos: *Diffuse* (Mapa de cor), *Normal Map* (Mapa de profundidade/detalhe) e *Lighting Map* (Mapa de sombras) com uma resolução de 1024x1024.

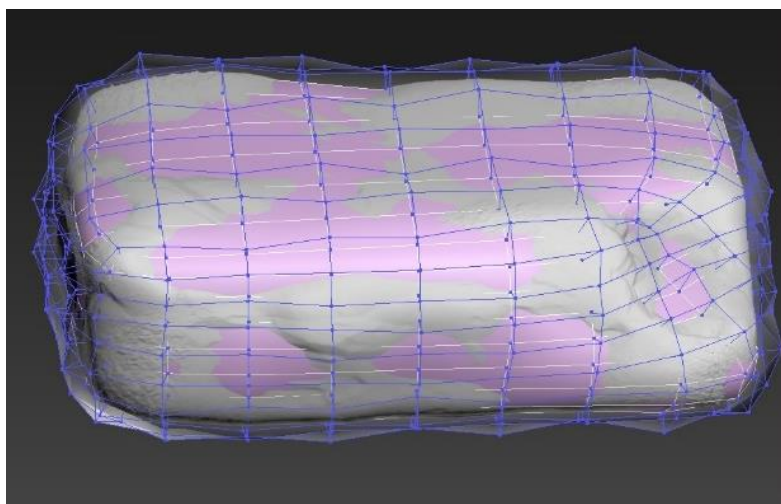


Imagem 62 – Projeção de um bloco do portal.

Visto que precisávamos do mapa *Lighting Map*, foi ainda necessário adicionar uma luz *Skylight* (cuja posição na cena não é importante), e configurar as opções de render para que as sombras produzidas

no modelo passem para este mapa. Todas as configurações encontram-se no anexo -7.4. Depois de termos realizados todas as configurações anteriores, o modelo está concluído e pronto para ser renderizado. O resultado desta renderização é um conjunto de imagens que ao serem colocadas nos vários mapas de texturização do modelo *low poly*, dão-lhe um aspeto igual ao modelo *high poly*. Todo este processo foi realizado em todos os modelos existentes e em todas as suas partes (todos os blocos do portal).

A projeção é um processo importante, como podemos ver na Imagem 63, a diferença do número de polígonos do modelo *high poly* para o de *low poly* é muito grande (aproximadamente 100x) e graças à projeção a diferença de detalhe entre as duas é praticamente nula.

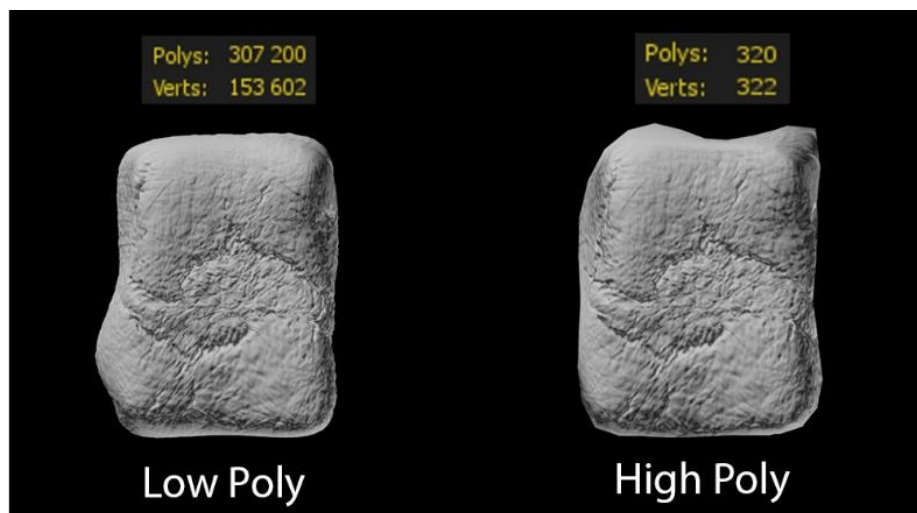


Imagem 63 - Comparação de o número de polígonos e vértices entre um bloco do portal *low poly* com um *high poly*.

## TEXTURIZAÇÃO

---

Depois de todo o processo de modelação e projeção, é hora de dar cor aos modelos. A texturização ao longo do projeto varia conforme o que fizemos. Ao todo, existem três tipos de texturização neste projeto.

### TEXTURIZAÇÃO UTILIZANDO UVW MAP

---

Caso o modelo não passe pelo processo acima descrito, ou seja, um modelo que não seja necessário realizar um mapeamento através do *Unwrap UVW*, podemos realizar o mapeamento através do *UVW Map*.

Por exemplo, nas flores e nas lianas normais e com espinhos do portal dos curandeiros utilizamos esta técnica. É adicionado o modificador *UVW Map* ao modelo que queremos texturizar. No caso das lianas, após adicionarmos o modificador, utilizámos um *Gizmo* cilíndrico (Imagem 64 - A), adicionámos a textura (Imagem 64 - B) que pretendemos e modificámos os *tiles* U e V conforme o seu aspeto até que se adequem ao pretendido.

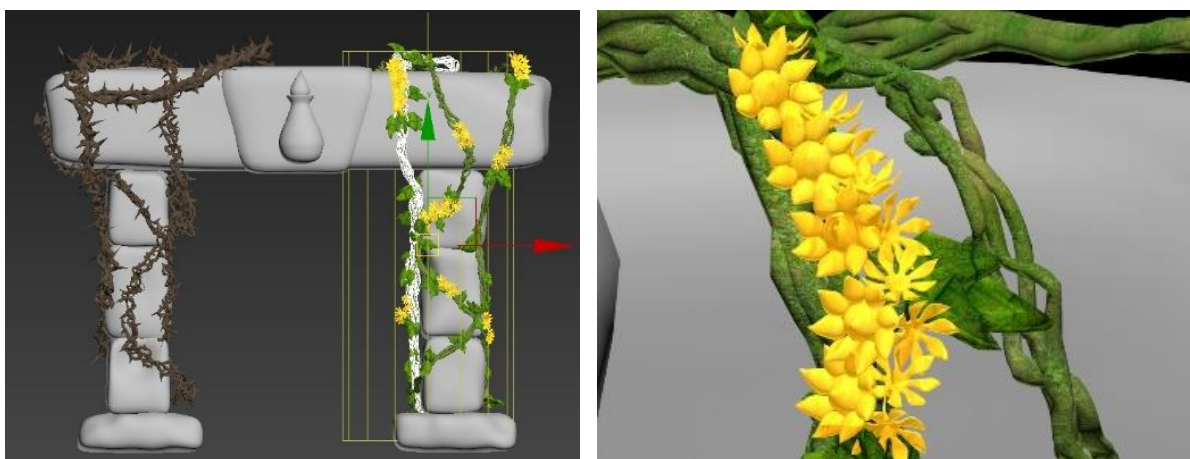


Imagem 64 – (A) Gizmo cilíndrico das lianas; (B) Textura adicionada através do modificador *UVW Map* nas lianas e nas flores.

Um caso diferente, mas no qual também foi utilizado o *UVW Map*, foi na texturização do terreno da floresta. A criação desta textura levou-nos à criação de um material diferente, tendo sido uma junção de texturas através do 3ds Max. Como queríamos uma textura um pouco verde (Imagem 65 - A) e um pouco castanho (Imagem 65 - B), decidimos então utilizar este método. A partir da opção *Blend* foi possível obter misturas de materiais, podendo definir a quantidade de cada textura e o seu ruído (*noise*). Depois da textura ter sido criada e como referido nos passos anteriores, arrastámos o material realizado (Imagem 65 - C) para o modelo pretendido.



Imagem 65 – (A) Textura de relva; (B) Textura de terra; (C) Textura final.

## TEXTURIZAÇÃO COM O PHOTOSHOP CS6

Para realizar este tipo de texturização, utilizados o Photoshop CS6 e os mapas que foram criados anteriormente, nomeadamente o *Diffuse Map*, *Normal Map* e *LightingMap*.

Começamos por adicionar a textura do *diffuse* (Imagem 66 - A) a uma camada, em seguida adicionámos uma nova camada com a textura que queríamos, neste caso uma textura de pedra e aplicámos a esta textura uma mistura de sobreposição (*Overlay*) com a camada anterior. A próxima camada que adicionámos foi a de *lighting map* (Imagem 66 - B) onde aplicámos um misturador *Multiply* que introduz um efeito de escurecimento. A última camada a ser aplicada foi o *normal map* (Imagem 66 - C). A esta camada é aplicada a um misturador de *Multiply* e adicionámos um efeito preto e branco o que transforma o *normal map* numa imagem em tons de cinza. Após estes passos, surge a textura *diffuse* que seria colocada no 3ds Max no passo seguinte (Imagem 66 - D).

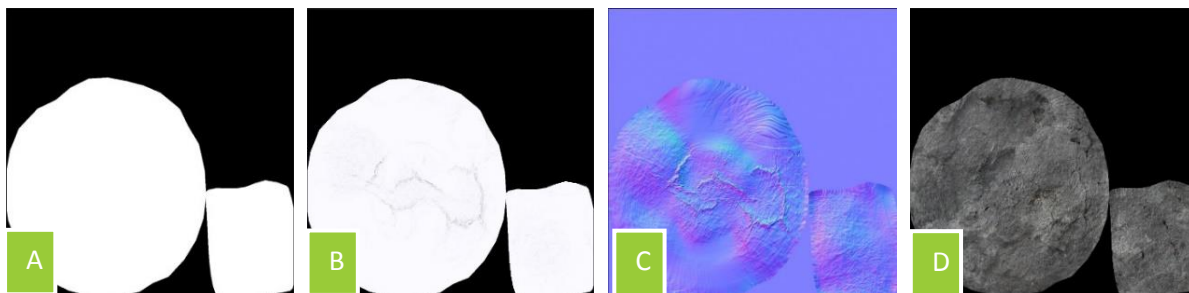


Imagem 66 –(A) Textura Diffuse Map; (B) Textura Lighting Map; (C) Textura Normal Map; (D) Textura final.

Ao terminar a textura realizada anteriormente, podemos adicioná-la ao modelo final. No 3ds Max criámos um novo material que pode ser visto na Imagem 67 - A. Assim que a textura esteja ligada ao novo material, basta arrastar a textura para a parte do modelo correspondente (Imagem 67 - B).

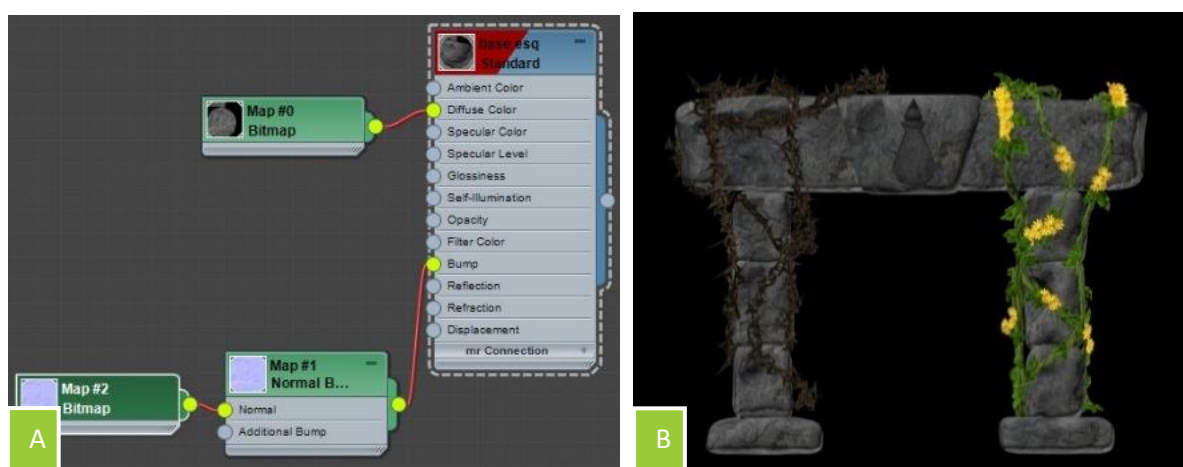


Imagem 67 – (A) Adicionar texturas a um material no 3ds Max; (B) Portal dos curandeiros com a texturização final.

## TEXTURIZAÇÃO COM O SUBSTANCE PAINTER

Para poder texturizar utilizando o *Substance Painter 2* é necessário ter cuidados com o modelo ainda no 3ds Max. No 3ds Max colocamos em cada modelo um material com uma cor sólida e identificamo-lo com um nome para poder mais tarde identifica-lo no *Substance Painter*. Outro cuidado a ter, e o mais importante, é que todo o modelo tem de ter passado pelo processo de *Unwrap*, tal como foi explicado anteriormente. Depois de tudo isto, podemos guardar o arquivo com uma extensão *.fbx* de maneira a este possa ser importado para o *Substance Painter*.

Para texturizar qualquer modelo no *Substance Painter* é fundamental ter uma *pen tablet* para poder ter uma pintura com mais detalhes e precisão. Através da força realizada na caneta podemos definir as zonas que queremos uma textura ou relevo mais leve ou profundo.

Ao importar mais do que um modelo do 3ds Max, tal como no exemplo que vamos dar, em que temos o terreno das montanhas e da água, surge a opção de qual o modelo que queremos texturizar no *Substance painter* (Imagem 69 - A). O *software* permite-nos ainda configurar a interface de maneira a podermos texturizar o modelo em 3D ou em 2D (Imagem 69 - B), para obtemos um melhor *feedback* do que está a acontecer. Neste projeto, trabalhamos sempre com as duas janelas ativas, assim podemos pintar na janela 3D e ter uma ideia de como estava a ficar a textura na janela de 2D (Imagem 68).

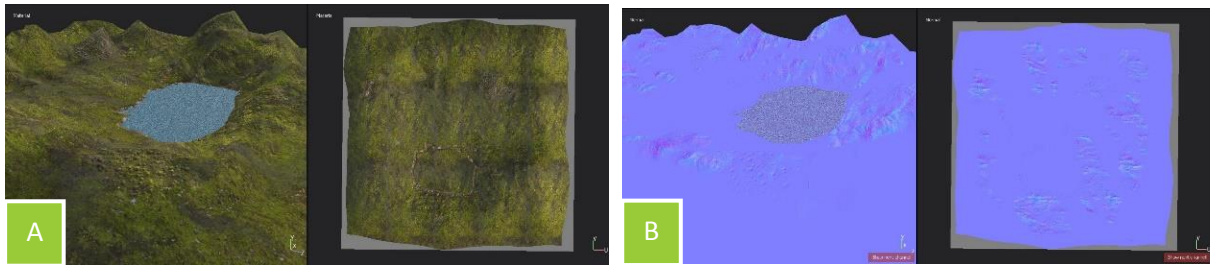


Imagem 68 – (A) Janelas 3D e 2D do material atribuído ao modelo; (B) Janela 3D e 2D com o mapa *Normal*, onde podemos ver os relevos adicionados ao modelo

Tal como no *Photoshop*, o *Substance Painter* também trabalha com camadas de maneira a facilitar o processo de texturização (Imagem 69 - C). Para texturizarmos o terreno das montanhas, foi necessário adicionar uma camada com uma textura base (verde). Já com uma base, são adicionadas outras camadas, mas desta vez com máscaras aplicadas. Estas máscaras, servem para adicionar ao modelo a textura, apenas nas áreas pintadas da máscara. Este processo foi repetido até que a textura satisfizesse o que era pretendido.

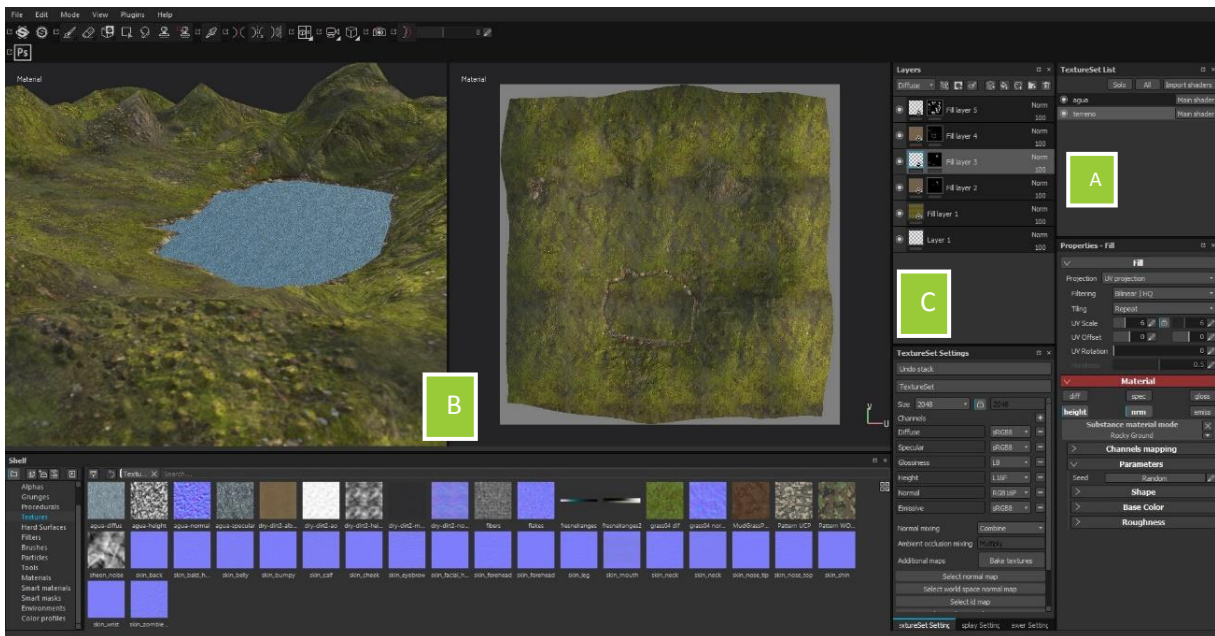


Imagem 69 – (A) Parte da interface onde podemos escolher qual é o modelo que queremos texturizar; (B) Interface em 3D e 2D do modelo escolhido; (C) As varias camadas de pintura que estão aplicadas ao modelo.

Após a conclusão do processo da textura do passo anterior é necessário exportar os mapas que necessitamos. O *Substance Painter* cria vários mapas, tais como *Diffuse*, *Specular*, *Glossiness*, *Normal*, *Height* e *Emissive*. Mas para o âmbito deste processo, os únicos mapas necessários foram o *Diffuse* e o *Normal Map*. Os mapas foram exportados com o tamanho 4096x4096 pixéis.

### 3.2.1.2. CRIAÇÃO DAS PERSONAGENS

A criação das personagens sofreu algumas iterações até chegar à *pipeline* final de produção de personagens (Imagem 70). Existem dois tipos de personagens no projeto, os humanos e um animal.



Imagem 70 - Pipeline de criação de uma personagem

## PROVAS PARA ESCOLHA DO MÉTODO DE PRODUÇÃO DE UMA PERSONAGEM

A ideia inicial para este projeto era a criação e modelação de todas as personagens necessárias, mas com os vários testes que realizámos e, visto que um dos focos deste projeto é utilizar ferramentas gratuitas ou economicamente vantajosas e também uma rápida produção, esta ideia foi descartada.

Um dos testes realizados foi o quão complicado seria realizarmos o *rigging* facial de uma personagem, já que precisaríamos de ter um bom *rigging* para podermos criar uma boa base para depois realizar a animação. Para realizarmos o *rigging* facial, procurámos um modelo que tivesse um rosto adequado para animar (Imagem 71). O programa escolhido para realizarmos o *rigging* facial do modelo foi o Maya 2015. O Maya, pela experiência que obtivemos, é dos melhores programas para a realização deste tipo de trabalhos.

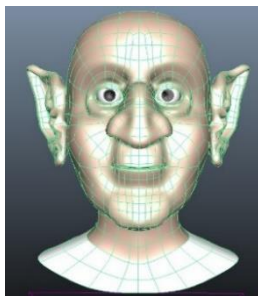
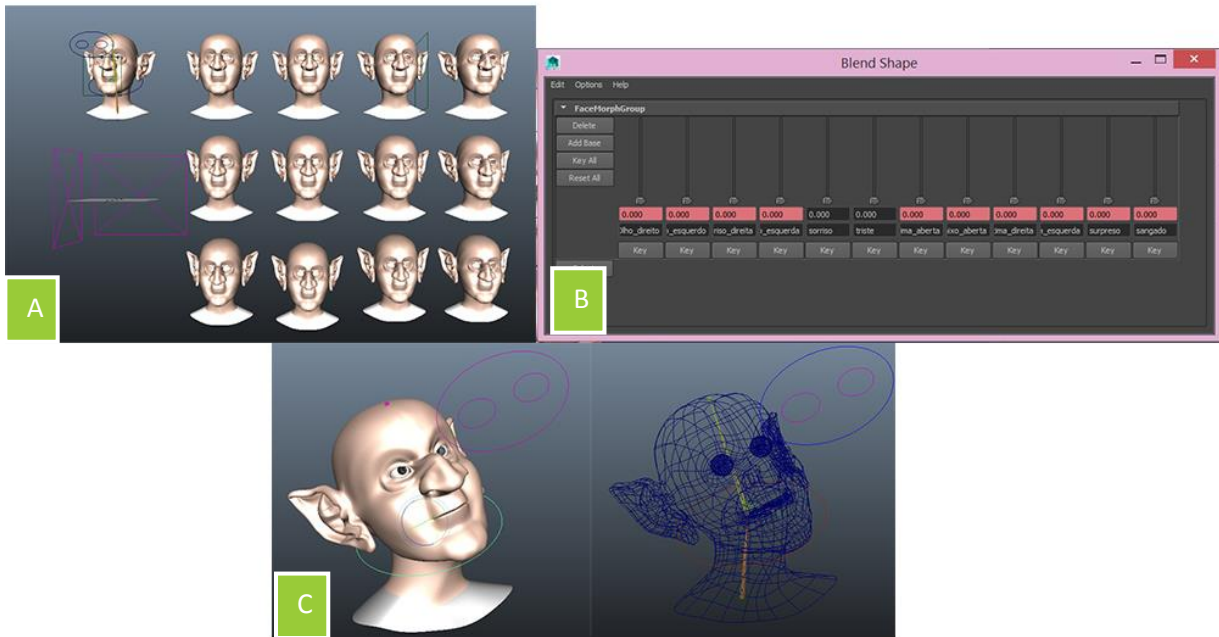


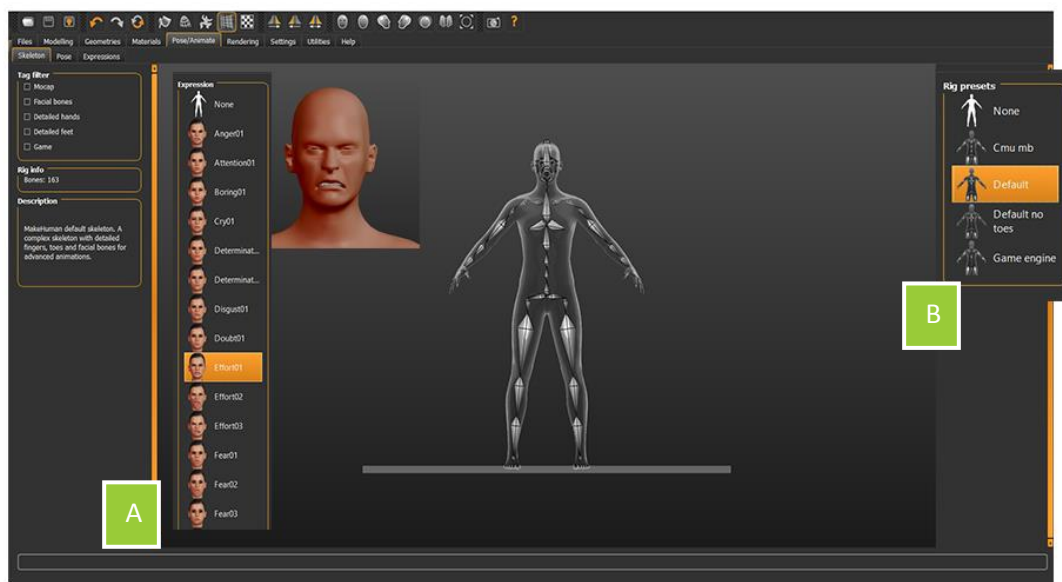
Imagem 71 - Modelo escolhido para elaborar o *rigging*.

A técnica utilizada para elaborarmos este teste foi o *Blendshape* (Imagem 72 - B), que consiste na multiplicação do rosto base, podendo assim deformar cada um dos rostos duplicados em várias outras expressões faciais (Imagem 72 - A). Depois de realizar todas as expressões, juntamos todas com o modelo base do rosto. Para o movimento dos olhos e cabeça utilizámos os controladores tal como ilustrado na Imagem 72 - C. A utilizando os ossos adicionados à cabeça.

Seguindo todo este processo foi possível terminar o *rigging* facial e, com isto, concluímos que era um processo complicado e demorado pois o rosto humano é extremamente complexo, tendo em conta que seria necessário realizar o mesmo processo para as oito personagens. O *rigging* corporal não entrou na fase de teste porque já sabíamos como era possível realizá-lo e tínhamos uma noção de tempo e dificuldades que poderíamos vir a encontrar, visto que já se tinha alguma experiência nesta parte.



Foi necessário arranjar uma nova alternativa. As alternativas passaram pelos *softwares* de criação de personagens, o que nos levou a encontrar o *Makehuman*. O *Makehuman* permite-nos contruir as personagens através de características humanas (género, idade, musculo, altura, tipo de pele) (Imagem 74 - A) e ainda ajustar o modelo e cores. Este *software* incorpora a *rigging* tanto facial, onde incorpora uma variedade de expressões (Imagem 73 - A), como o corporal, dando a escolha do tipo de esqueleto variando o número de ossos (Imagem 73 - B). Mas a sua desvantagem e fator limitador, é a existência de poucos adereços para as personagens, tanto em roupa como acessórios.



Apesar de existir uma comunidade [133] onde podemos realizar download de vários acessórios, roupas e esqueletos compatíveis com vários programas, não era suficiente para diferenciar todas as personagens e caracterizar as personagens de acordo com os estilos medievais inicialmente pensados (Imagem 74 - B).

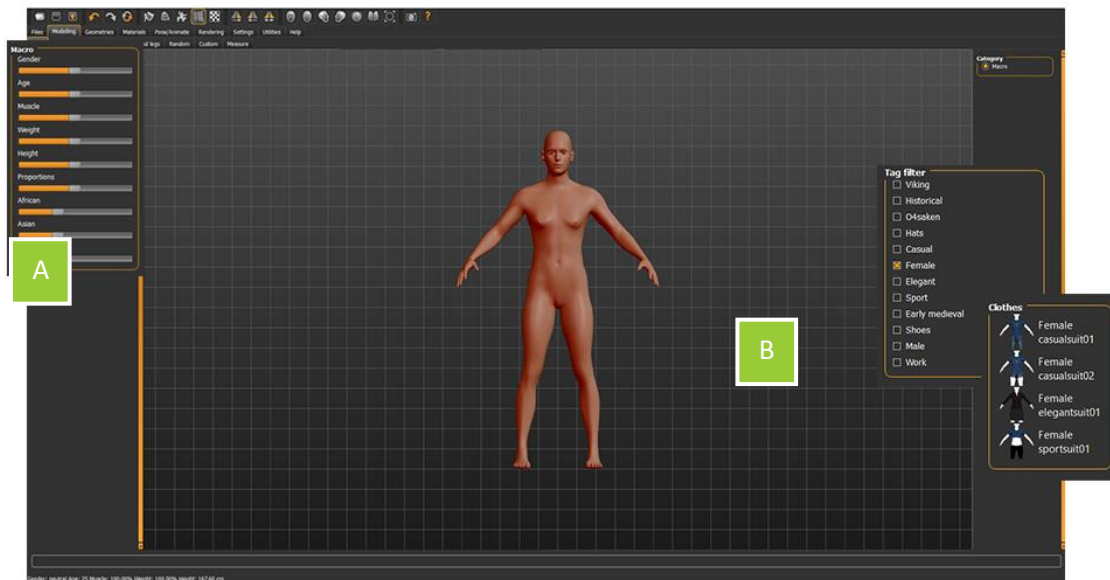


Imagem 74 – (A) Característica para criação personagens; (B) Opções de roupa do *Makehuman*

Por esta razão, a utilização do *Makehuman* foi descartada. Depois do *Makehuman*, conhecemos o *software Fuse*. Este foi o *software* escolhido e utilizado para elaborarmos todas as personagens que encontramos no projeto.

## CRIAÇÃO DAS PERSONAGEM – MODELAÇÃO

Criar uma personagem no *Fuse* torna-se fácil e as múltiplas funcionalidades ainda tornam esta criação mais simples. Para demonstrar a criação de uma personagem, iremos apresentar a criação de uma personagem deste projeto, a Isene.

O *Fuse* oferece uma grande variedade de personalização relativamente às várias partes do corpo (cabeça, braços, pernas e tronco) de maneira a construir uma personagem que vai de encontro com as nossas necessidades. A criação de uma personagem é simples, basta selecionarmos as várias partes do corpo que queremos e o modelo vai-se transformando com as escolhas realizadas. Colocar o vestuário é um processo semelhante à personalização do corpo. No separador de roupa podemos encontrar acessórios, sapatos e várias partes de roupas, existindo aqui vários estilos à escolha (Imagem 75).



Imagem 75 - Interface do Fuse com algumas opções de cabeças e de vestuário.

Este *software* também permite modificar um pouco o modelo, como mostra a imagem (Imagem 76). Podemos selecionar a parte do corpo que queremos modificar e arrastamo-la de maneira a provocar uma deformação ou podemos também realizar esta deformação através do menu *Customizer*, onde cada parte do corpo está dividida em pequenos pontos de deformação. Estas opções ajudaram-nos a ajustar o corpo que se sobreponha à roupa e ao cabelo.

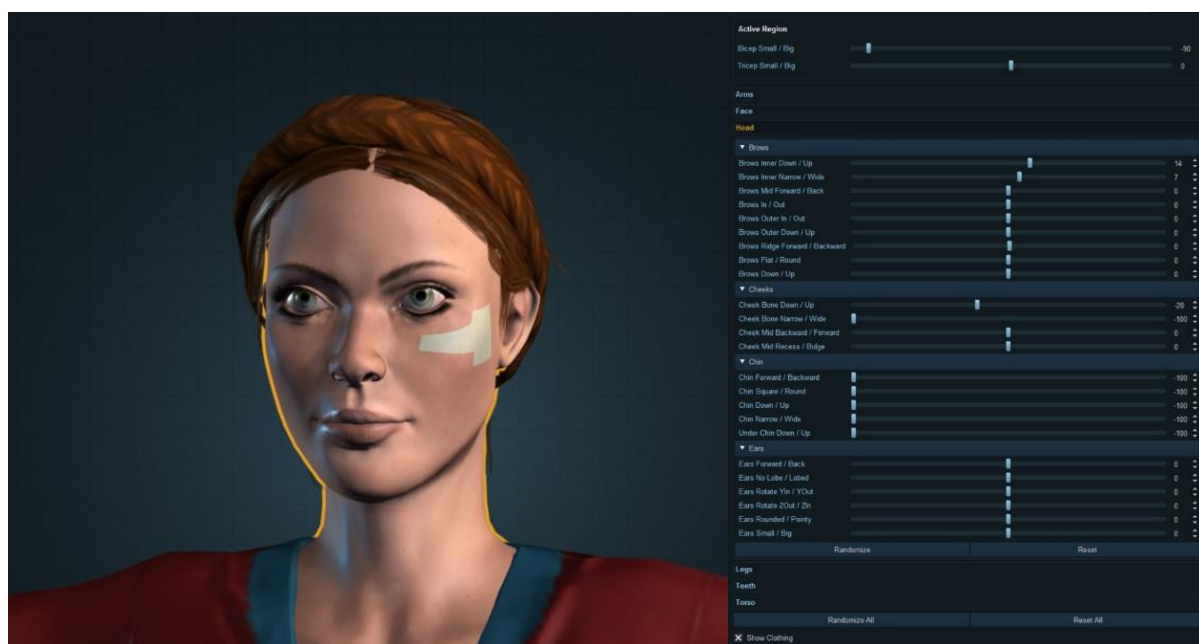


Imagem 76 – Opções de modificação do modelo.

Tal como editamos o corpo da personagem, o mesmo pode ser realizado com a roupa através do separador *Texture*. Mas desta vez pode ser editado a cor e as texturas que a roupa pode ter. Além das texturas de tecidos, podemos também adicionar metais e cores sólidas á roupa, personalizando assim as personagens (Imagem 77). Algo muito positivo, foi a liberdade de criar uma personagem tal e qual como a imaginamos e de até termos as opções de adicionamos características humanas, tal como a idade da personagem em questão.

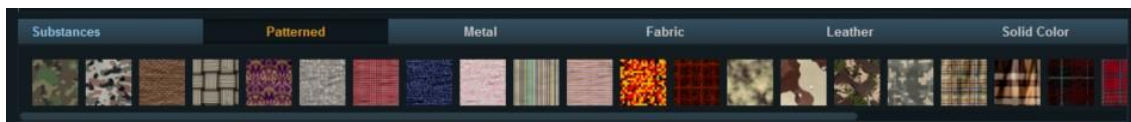


Imagem 77 – Tipos de texturas existentes no Fuse

A sua topologia é pensada para a animação tanto corporal como facial, o que facilita na parte da animação graças à sua boa deformação e realismo da mesma (Imagem 78).



Imagem 78 – Representação da topologia no rosto e no corpo.

O *Fuse* foi o *software* com o qual realizámos testes e o achámos mais completo a nível de criação de personagens. O *Fuse* tem uma ligação direta com o *auto-rigger*, o que nos facilita este procedimento, já que é com o *auto-rigger* que toda a parte do *rigging* da personagem é criada.

Dos programas testados, este, por ser o mais completo e o que nos entrega uma personagem com *rigging* corporal e facial, foi o programa escolhido para a realização das oito personagens que incorporam a nossa história

### RIGGING/SKINNING

O *rigging* e o *skinning* são dois processos diferentes, sem os quais a animação de uma personagem seria impossível. Com o *rigging* podemos criar o esqueleto ou ossos que sejam necessários para que uma personagem se possa mexer, redimensionando cada osso de maneira a ocupar a maior parte do modelo. Mas o *rigging* não é suficiente para que uma personagem esteja pronta para passar para o seguinte passo, a animação. Antes de seguir para a animação, adicionamos o modificador *skin*. O *skinning* permite fazer a ligação entre os vértices do modelo 3D e os ossos criados no *rigging*. Esta ligação é quantificada através de um número de 0 a 1 (chamado de pesos) que traduz a deformação que o modelo 3d sofre quando o osso se mexe. É graças a estes pesos, que é possível movimentar o osso e desta forma realizar o movimento do modelo.

O *Mixamo* é uma plataforma que complementa o *Fuse*. Aqui encontramos personagens que já foram desenvolvidas e uma grande variedade de animações que foram realizadas através de *MoCap*. Como referido anteriormente, nesta plataforma também está incorporada a opção de *auto-rigging*,

bastando, para isso, carregar no botão *Animate* que se encontra no *Fuse*. Ao carregar no *Animate* o *Fuse* realiza uma ligação direta com o *auto-rigger* do *Mixamo*, carregando assim a personagem que estava a ser modelada.

O *auto-rigger* permite-nos escolher a quantidade de ossos que desejamos na personagem e se queremos que o modelo tenha ou não *Blendshapes* (Imagem 79).

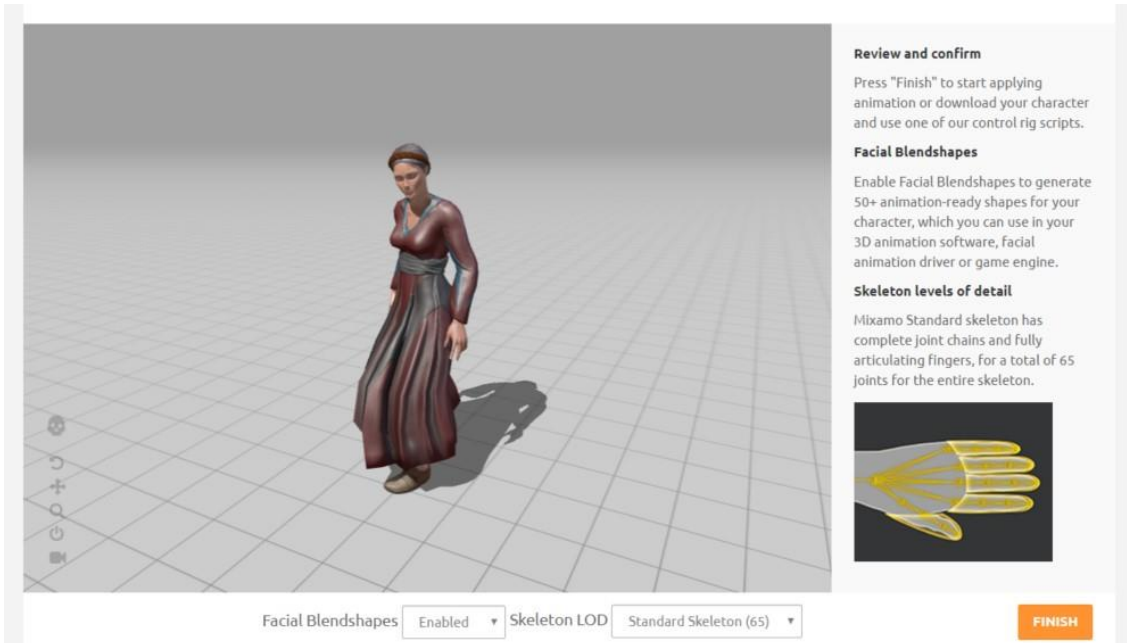


Imagem 79 – *Auto-rigger* do *Mixamo*, onde escolhemos se queremos *Blendshapes* e a quantidade de ossos.

Os esqueletos do *Mixamo* variam conforme o número de ossos que queremos na mão do nosso modelo. Existem quatro tipos diferentes de esqueletos, apresentados na seguinte tabela.

Tabela 4 – Diferentes tipos de esqueletos existentes no *Mixamo* [59].

<p><b>Standard Skeleton (65)</b></p>		<p>É o esqueleto mais completo que existe no <i>Auto-Rigger</i> do <i>Mixamo</i>, com 65 articulações. Este contém articulações completas, abrangendo todo o corpo e também todos os dedos das mãos. Pela sua complexidade foi o esqueleto escolhido para a elaboração do <i>Auto-Rigger</i></p>
<p><b>3 Chain Fingers (49)</b></p>		<p>Tem um conjunto de articulações no polegar, no dedo indicador e no dedo anelar. O conjunto de articulações do dedo anelar controla também os dedos mindinho e médio. No total este esqueleto incorpora 49 articulações.</p>
<p><b>2 Chain Fingers (41)</b></p>		<p>Tem um conjunto de articulações no polegar e também no dedo médio. O conjunto do dedo médio controla também todos os outros dedos (indicador, anelar e mindinho). Tornando assim um esqueleto com 41 articulações.</p>

### No Fing (25)



O número de articulações é reduzido, com apenas 25 articulações. Nesta opção é removida todas as articulações dos dedos.

O esqueleto que escolhemos para todas as personagens foi o *Standard Skeleton* pois é o mais completo e é compatível com os *softwares* da Autodesk. Em relação ao *Blendshapes*, o *Mixamo* dá-nos a escolher se queremos ou não incorporar no modelo. Esta opção oferece-nos mais de 50 formas de animação facial. Depois de selecionarmos o tipo de *rigging* que queremos, a personagem está pronta para fazer *download* na plataforma do *Mixamo*.

Outro processo de *rigging* e *skinning* foi utilizado com o único animal que existe na história, o urso. O urso foi um modelo cedido pelo *Cubebrush* [134]. Este disponha do *rigging* e do *skinning* do corpo inteiro do urso, à exceção da boca, o que nos levou a realizar o *rigging* e do *skinning* dessa parte. Vamos demonstrar em seguida como foi realizado esse processo.

Primeiramente foi adicionado um osso na boca do urso de maneira a que cobrisse a maior área da boca do modelo. Se o osso cobrir a área adequada, a fase de *skinning* é muito mais fácil pois os pesos estarão adequadamente distribuídos. Em seguida, foi adicionado o modificador *Skin*. Sendo que o osso adicionado não possuía todos os pesos incluídos necessários, estes foram adicionados. Selecionando o osso e os vértices que queremos adicionar/remover do determinado osso, bastou apenas abrir a opção *Weight Tool* (Imagem 80) no 3ds Max. Esta opção é onde tratamos dos pesos que cada vértice tem sobre o osso alvo.

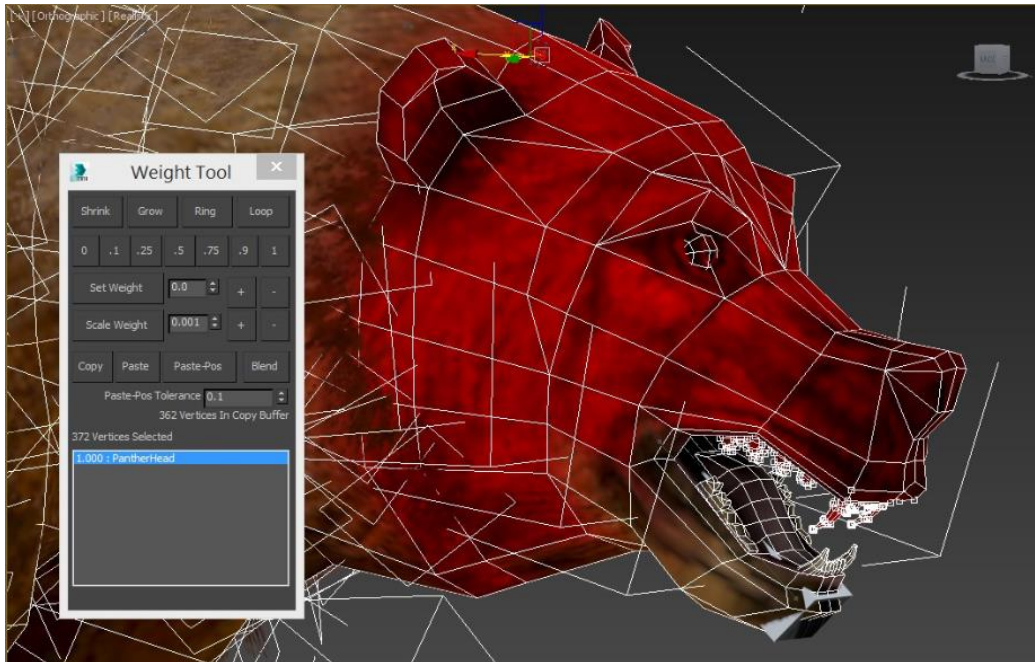


Imagem 80 – Skinning utilizando o *Weight Tool*.

Estes foram os dois processos de *rigging* e *skinning* que realizados no projeto. O processo utilizado no urso é o processo que também poderia ter sido realizado nas personagens. No entanto, com a utilização do *auto-rigger* do *Mixamo*, todo este processo tornou-se mais simples.

---

### 3.2.1.3. MONTAGEM DAS CENAS

Nesta secção abordaremos como foi realizada a montagem das cenas. Montar as cenas de uma história não é simplesmente importar os modelos e colocá-los nos lugares adequados. Na *pipeline* abaixo (Imagem 81) podemos ver o procedimento que realizámos para a montagem das cenas.

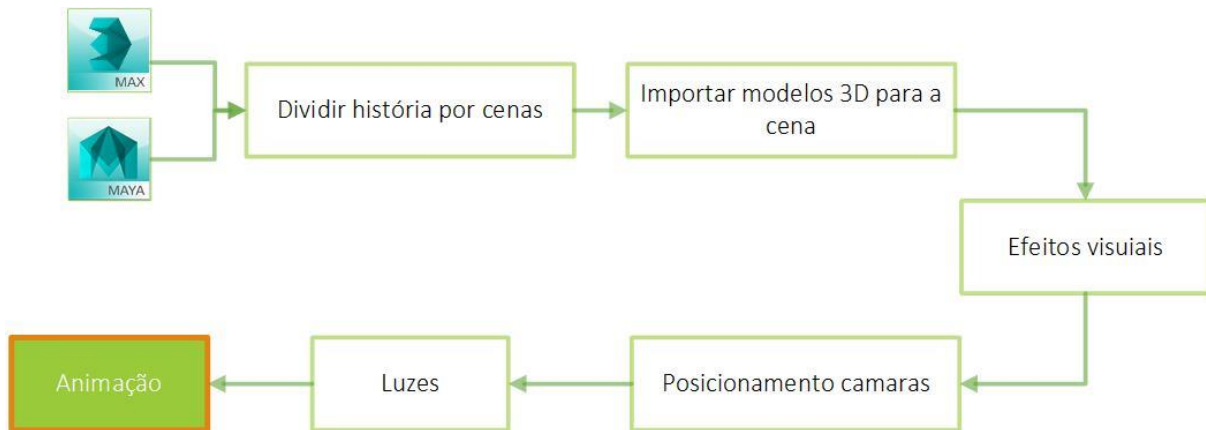


Imagem 81 -Pipeline do procedimento que realizamos para a montagem das cenas.

---

#### DIVISÃO DAS CENAS, ESCOLHA DO SOFTWARE E IMPORTAÇÃO DOS MODELOS

O primeiro passo é dividir a história em cenas. A primeira questão que surgiu foi como dividir a história em cenas. Existem vários fatores importantes que nos levam a dividir as cenas. Um deles é o *software* que vamos utilizar para aquela cena e quais são as vantagens de o utilizar, no nosso caso utilizamos dois *softwares* diferentes, o 3ds Max e o Maya. A escolha do Maya, como já foi referido, deveu-se à facilidade e ao realismo das partículas e os seus efeitos visuais. Então, todas as cenas que faziam parte dos cenários das galáxias foram criadas neste *software*. Todas as outras cenas foram desenvolvidas no 3ds Max.

Outro fator é o cenário, dividimos sempre as cenas conforme o cenário em que nos encontramos, assim podemos trabalhar em uma cena com um só cenário. Dividir as cenas por cenários (Imagem 82) por vezes não é suficiente e há necessidade de realizarmos mais uma divisão, isto porque, o tamanho do arquivo também é um fator importante. Ao começar a importar os modelos para a cena, o ficheiro pode-se tornar muito pesado, o que dificulta o trabalho. Depois de dividir todas as cenas podemos importar todos os modelos criados para reproduzir cada uma das cenas. Neste passo, é importante confirmar que todos os modelos têm a mesma escala.



Imagem 82 – Alguns dos cenários divididos (A) Espaço; (B) Vista total da vila e floresta; (C) Floresta; (D) Vila.

## EFEITOS VISUAIS

Os efeitos visuais são o que dão vida e cor às cenas. Através de efeitos visuais, como partículas ou efeitos de fumo, é possível captar a atenção de quem está a visualizar o filme. Neste tópico explicaremos quais foram os efeitos visuais utilizados, tanto para o *software* Maya como para o 3Ds Max.

### MAYA

Como foi descrito no tópico anterior, o Maya foi escolhido pelas suas capacidades na criação de efeitos visuais, apesar da sua complexidade e quantidade de opções disponíveis. No Maya foram criadas todas as cenas das galáxias e todos os efeitos necessários. Para realizar estas cenas, precisávamos de realizar efeitos visuais para o sol, para a explosão da nave, o rasto que esta deixa ao se deslocar no espaço e para todas as galáxias que existem no espaço. Para criarmos as galáxias utilizamos duas maneiras diferentes.

A primeira forma de criarmos uma galáxia foi utilizando o “Paint Effects Brushes”. Esta técnica permite-nos escolher uma variedade de pincéis com vários efeitos. Para realizar esta galáxia começamos por escolher um pincel que mais se adequasse ao pretendido. O pincel eleito, escolhido de uma gama de pincéis da secção *galactic*, foi o *spaceGasBlue.mel*.

Com este pincel selecionado, começámos a desenhar a galáxia, realizando circunferências mais intensas no centro e desvanecendo-se conforme nos afastamos do centro, obtendo assim a forma de galáxia. Em seguida, com o pincel *galaxy.mel* desenhámos uma pequena concentração de estrelas no centro da galáxia para dar um efeito estrelado (Imagem 83). Algumas das galáxias foram criadas desta maneira, tendo sido apenas necessário mudar a cor e formato.

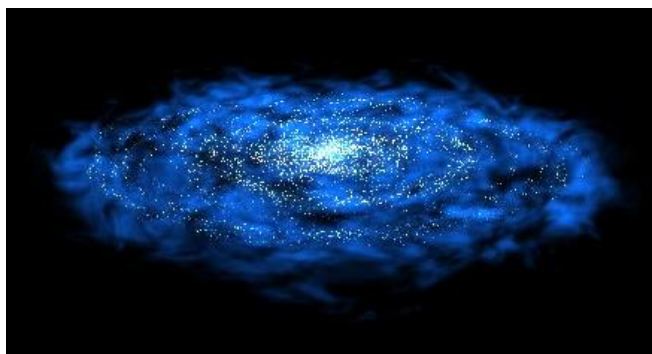


Imagem 83 - Galáxia azul.

Esta mesma técnica foi utilizada para realizarmos as estrelas do espaço, mas desta vez utilizamos o pincel *space.mel* (Imagem 84)

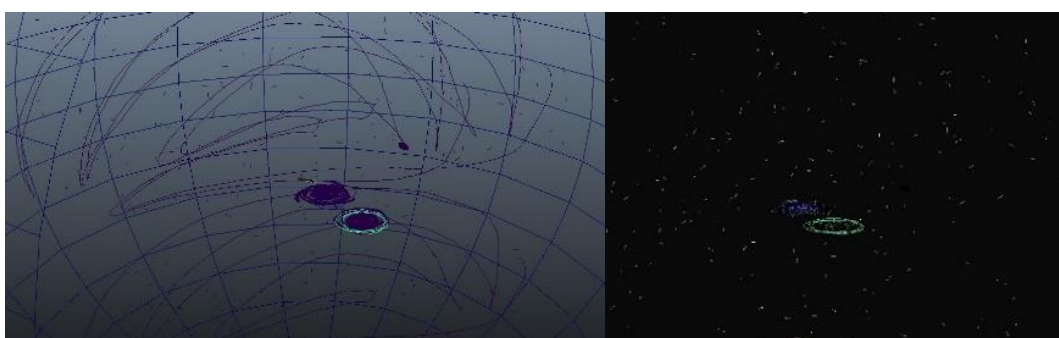


Imagem 84 – Elaboração das estrelas através do pincel *space.mel* oferecido pelo Maya.

A outra maneira que utilizamos para criar as galáxias foi com o *Emitter from object*. A partir do *Emitter from object* podemos criar partículas através duma imagem. Estas partículas são criadas tal e qual à imagem, tanto na sua forma como nas cores. A Imagem 85 mostra-nos os dois tipos de imagens que foram necessários para a criação das partículas. Uma das imagens representa o que queremos que seja reproduzido e a outra, a preto e branco, representa onde serão e não serão produzidas partículas (na parte preta não são produzidas partículas). A última imagem representa o sistema de partículas que criámos através da imagem inicial.

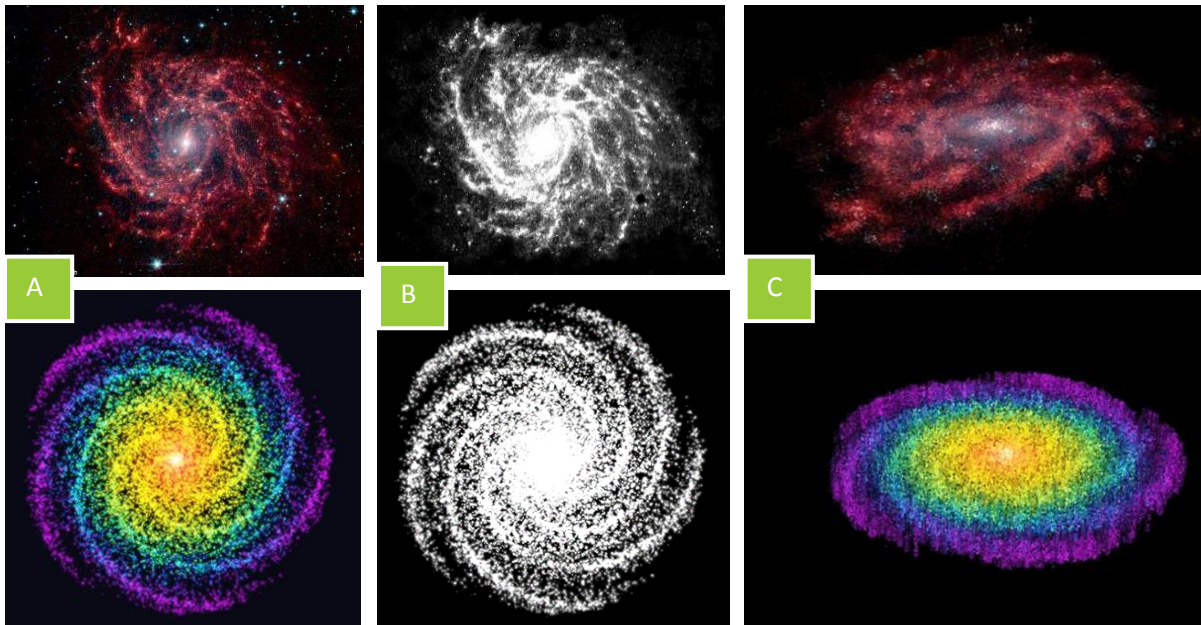


Imagem 85 - Galáxias. (A) Imagem colorida da galáxia; (B) Imagem da galáxia preto e branco; (C) Sistema de partículas.

Para criarmos esta galáxia tivemos de recorrer à opção *particle*, no menu *Dynamics*. Nesta opção escolhemos a superfície onde as partículas são emitidas. Depois disto, foi necessário escolher as imagens que vão ser reproduzidas pelas partículas (Imagem 85 – A e B). Isto faz com que, passemos de duas imagens, em plano 2D, para um sistema de partículas fiel as imagens escolhidas (Imagem 85 - C). Esta transformação pode ser configurada na opção *Emitter*, na janela *Attribute Editor*. De forma a finalizar este processo, foi necessário adicionar a opção *Add Per Particle Attribute*, presente no menu anterior, fazendo com que as partículas sejam finalmente criadas. Este método apenas foi utilizado para criarmos algumas galáxias.

Outros dos efeitos visuais que criámos foram para a nave. Um dos efeitos, foi o rasto que a mesma deixava no espaço ao se deslocar e o outro foi a explosão num dos motores da nave. De forma a replicar o efeito do rasto da nave é possível seguir todos os passos na Tabela 7 - anexos.

Tal como nas galáxias, a primeira coisa a ser realizada foi a criação de um *Emitter* de partículas em formato *box*. Foi definido o aspeto que as partículas iriam tomar (passo 1 - Tabela 7) e em seguida definimos o tipo de vida que as partículas adquiriram (passo 2 - Tabela 7). O passo 3 - Tabela 7 levou-nos à configuração de vida das partículas, clicando com o botão direito na opção *Lifespan PP* criamos uma expressão. Na nova janela adicionámos a expressão (Imagem 86) a qual fará com que cada partícula tenha diferentes tempos de vida.

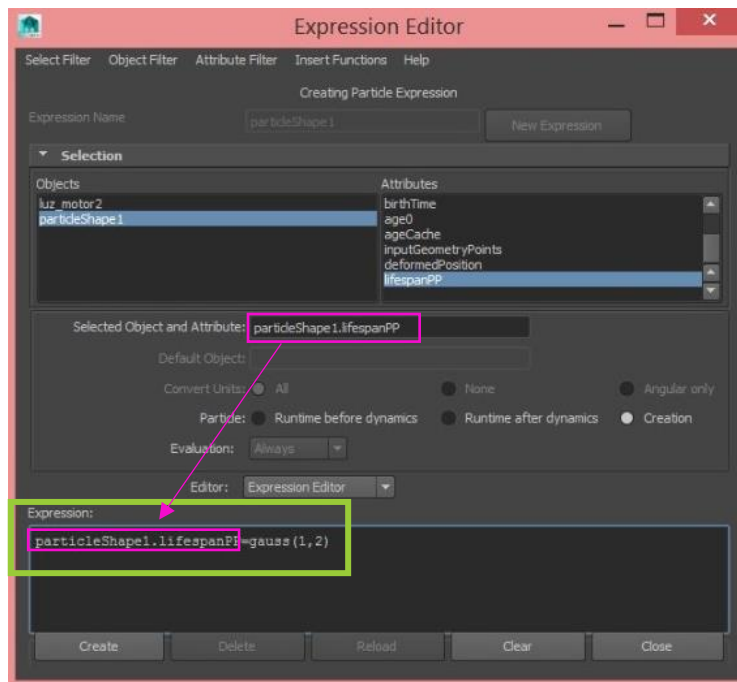


Imagem 86 – Expressão que dá diferentes tempos de vida a cada partícula.

De seguida adicionámos atributos dinâmicos às partículas individualmente (passo 4 - Tabela 7). Em seguida voltámos a executar o passo 3 - Tabela 7, clicando com o botão direito na opção *Create Ramp* que permitiu modificar as cores do fumo da nave para uns tons de cinza e branco.

Seguindo o passo 8 - Tabela 7 chegámos à janela *Hypershade* (Imagem 87). Aqui foi criada a cor que as partículas iriam ter. Puxando o *Particle Cloud* (Imagem 87 - A) que se encontra no separador *Create* para a *Work Area*, criamos uma textura base automática. Selecionando a textura que acabou de ser criada e seguindo o passo 5 - Tabela 7, temos três opções (Imagem 88) para definir a cor que queremos: *Color*, *Transparency* e *Incandescence*.

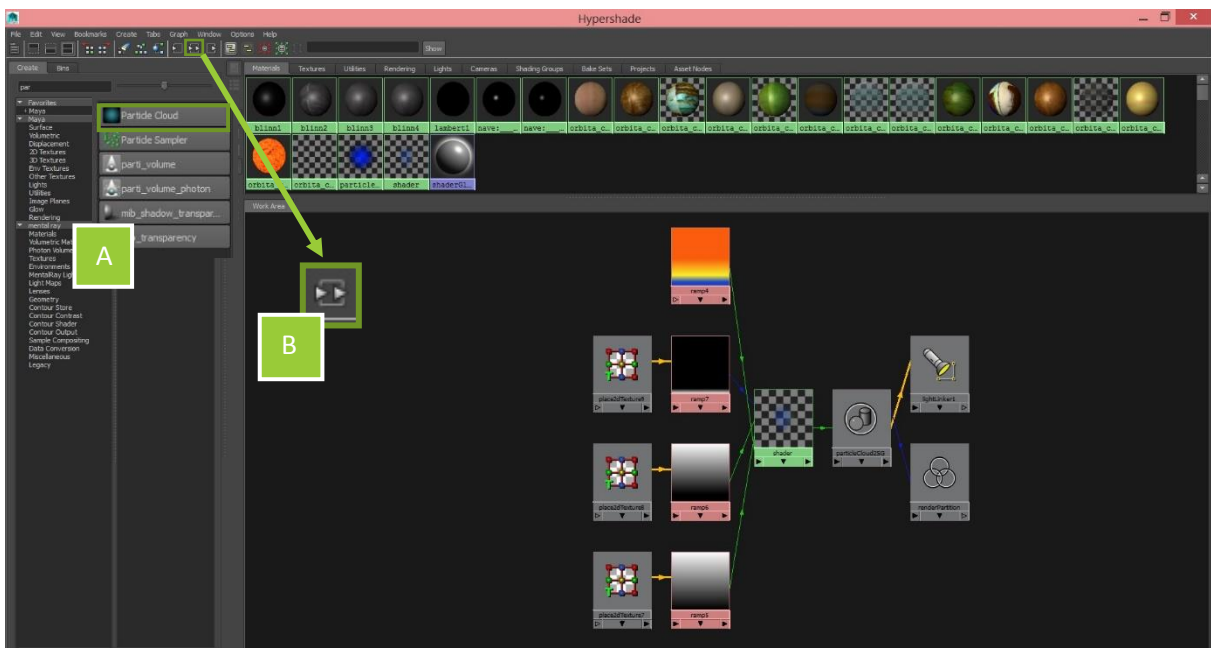


Imagem 87 – Janela *Hypershade*; (A) Opção *Particle Cloud*; (B) *Input and Output connection*

Em todas as opções carregamos no quadrado xadrez que se encontra à sua direita e selecionamos a opção *Ramp* (Imagem 89). Na opção *Color* escolhemos a cor que queremos dar às partículas, no *Transparency* com as cores preto e branco definimos a transparência das partículas, sendo o branco a cor que representa a transparência. Da mesma maneira que o *Transparency* realizamos o *Incandescence*, só que desta vez o branco representa a Incandescência das partículas.

Na Imagem 87 – B mostra a opção a clicar para que seja realizada a conexão de entrada e saída. Das muitas coisas que são criadas ao realizar este passo, foi criada uma luz. Com esta luz selecionada realizamos os últimos ajustes para finalizar o fumo da nave.

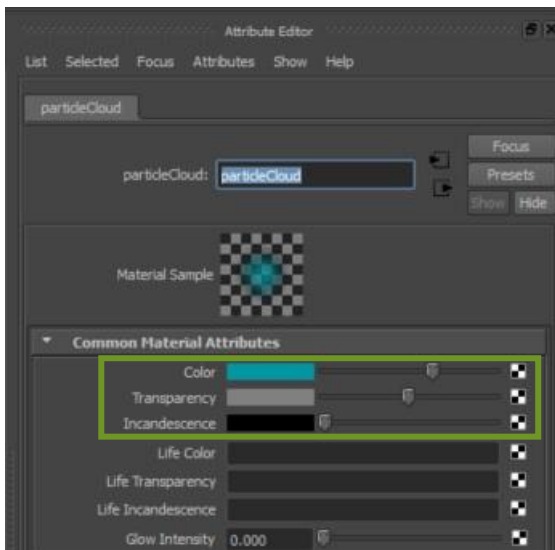


Imagem 88 - Particle Cloud

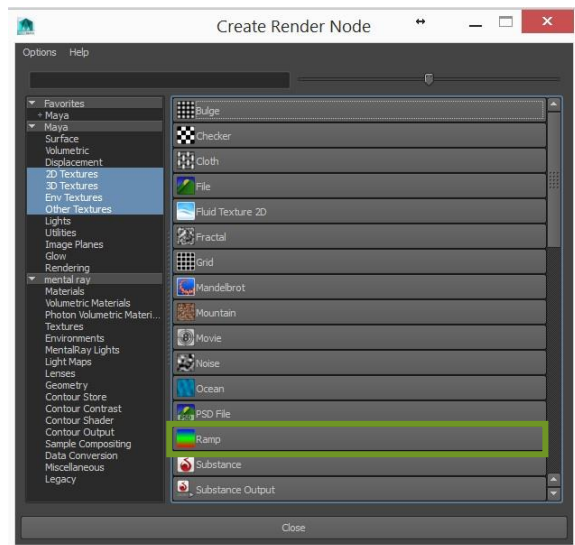


Imagem 89 - Create render node opção Ramp

Seguimos o passo 6 e 7 - Tabela 7 para poder mudar o *Glow intensity* e a transparência. Desta forma concluímos o processo de criação do rasto da nave (Imagem 90).

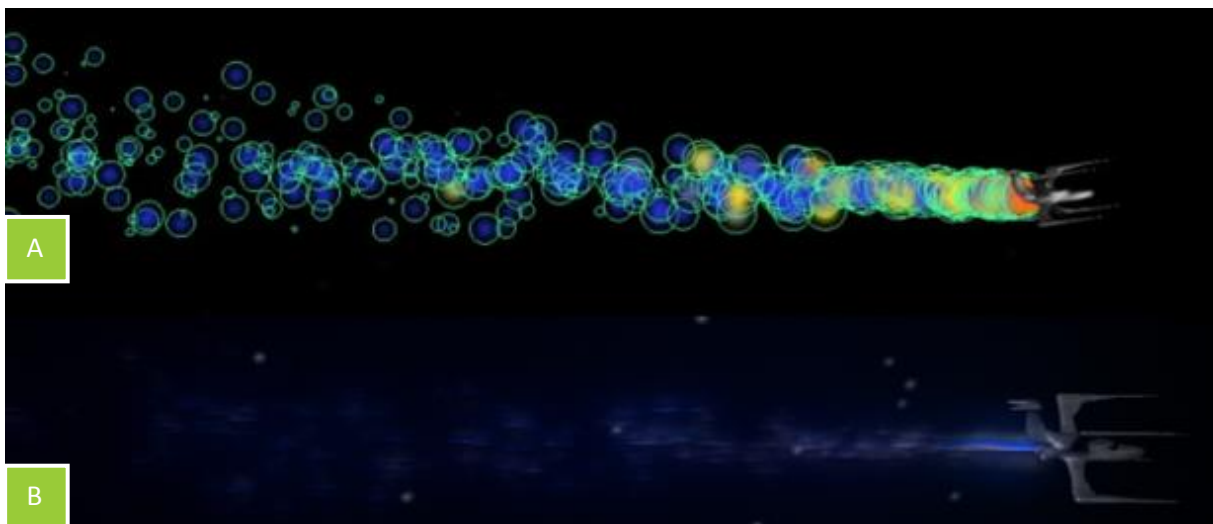


Imagem 90 – (A) Rasto da nave criado através de um sistema de partículas; (B) Render final.

A explosão foi o segundo efeito visual associado à nave e para isso utilizamos o *Fluid Effect* e recorremos a um tutorial [135]. Fluid Effect é uma simulação de fluidos que podem ser líquidos ou gases. Neste exemplo utilizamos gases e calculou-se o valor do deslocamento do fluido dentro do

*container* criado. Na descrição deste efeito visual iremos seguir os passos que se encontram na Tabela 8 - anexos.

Para realizar este efeito o único menu que utilizámos foi o *Dynamic*. Começamos por criar um *container* (Imagem 91) através da opção *Create 3D container with Emitter*. Será dentro deste *container* que é criado o fluido que por sua vez criará a explosão. Em seguida, aumentamos a resolução da base do *container* e o seu tamanho de maneira a que a explosão ficasse mais visível e com uma melhor resolução (passo 1 - Tabela 8). Assim, podemos ajustar o emissor do fluido para que este emita o fluido a partir do volume de uma esfera o que fará com que o fluido se espelhe pelo *container* e não seja projetado apenas numa direção (passo 2 e 3 - Tabela 8).

Toda a explosão necessita de ter densidade (*density*), aquecimento (*heat*) e combustível (*fuel*). Para isso, seguimos o passo 4 - Tabela 8, e desta forma podemos reproduzir a explosão com estas três componentes. Este passo consiste em três fases. Como uma explosão não emite sempre a mesma densidade, aquecimento e combustível conforme vai ocorrendo, então devemos criar *key frames* de maneira a marcar este valor na linha do tempo. Mas isto não é suficiente para mostrar o impacto da explosão. Ao vermos o gráfico (passo 5 - Tabela 8) (Imagem 91 - B) podemos ver que as duas opções que criámos em cima de densidade e aquecimento vão decrescendo pouco a pouco ao passar o tempo e isto não acontece na realidade com uma explosão. Por isso, devemos selecionar estas duas componentes no gráfico e clicar na opção *Step tangentes* (Imagem 91 - B), e deste modo eles decrementarão bruscamente imitando o efeito do início da explosão.

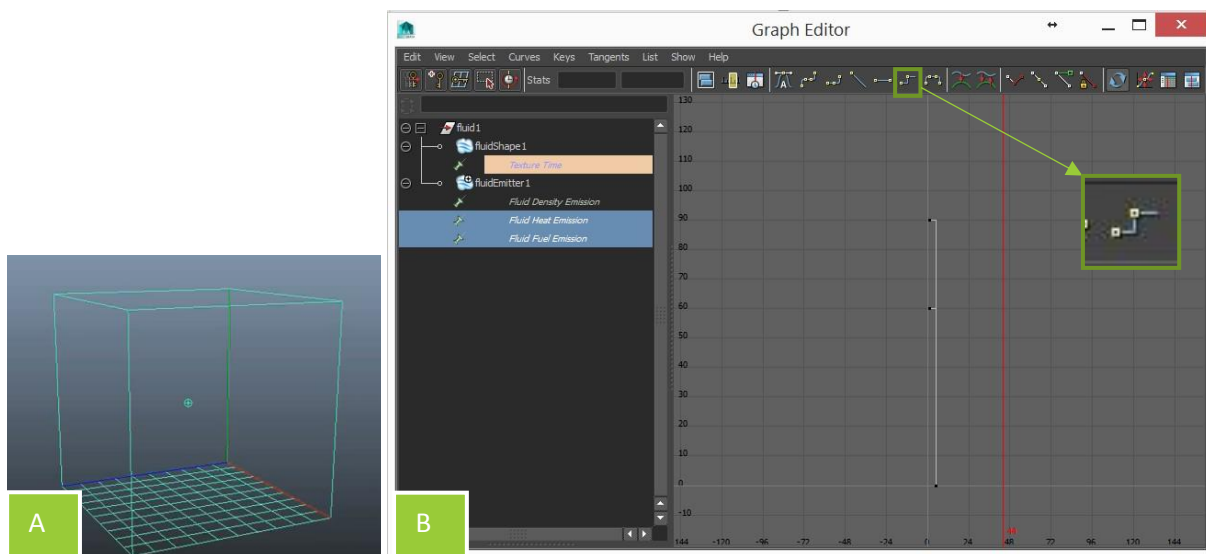


Imagem 91 – (A) Container; (B) Janela Graph Editor onde mostramos a inclinação provocada na densidade e no aquecimento através da opção *Step tangentes*.

Como a explosão foi calculada dinamicamente foi necessário ativar o cálculo da temperatura e do combustível e, para que esta se pareça com uma explosão verdadeira, também ativámos a cor (passo 6 - Tabela 8). Assim, sempre que o Maya calcular estes valores, as cores estarão de acordo com a temperatura e o combustível. Com isto, e ao ajustarmos a transparência do fluido, este fica com um melhor aspeto (passo 7 - Tabela 8). (Imagem 92)

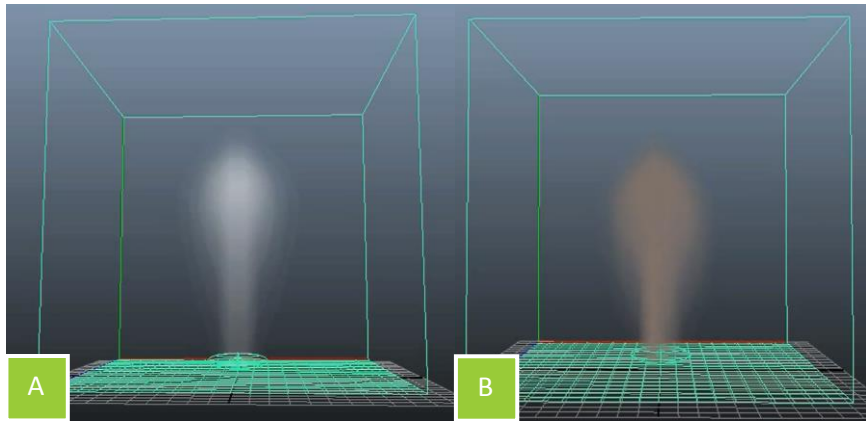


Imagem 92 – (A) Antes de calcular os valores da temperatura e combustível; (B) depois de calcular as cores da temperatura e combustível.

De seguida devemos ajustar todos os valores necessários para os cálculos do fluido, tais como o combustível, a temperatura, a turbulência, a velocidade e a densidade do fluido. O combustível é onde definimos o calor libertado e a velocidade da reação (passo 8 - Tabela 8), a temperatura é onde definimos a velocidade em que a temperatura cai e a maneira que o fluido flutua (passo 9 - Tabela 8), a turbulência, a velocidade e a densidade do fluido é onde podemos colocar o fumo um pouco caótico e com perturbações em forma de remoinhos (passo 10 e 11 - Tabela 8). Já que uma explosão não é sempre igual, é necessário definir que o fluido que existe no *container* irá desaparecer conforme se for espalhando (passo 12 - Tabela 8) e não se irá concentrar dentro do *container*.

Depois de todos estes passos já podemos ver a explosão com um melhor aspeto, mas ainda falta definir algumas opções de maneira a aperfeiçoá-la. (Imagem 93) Um dos aspetos que notamos foi a necessidade de aumentar a sua velocidade (passo 13 - Tabela 8), já que na hora de reprodução estava muito lenta e um pouco desfocado e, por isso, aumentamos um pouco a resolução do fluido (passo 14 - Tabela 8). Neste momento o fluido ainda tem um aspeto 2D, o que faz com que a explosão não pareça realista. Para resolver este problema, foi necessário ligar a opção que cria um efeito 3D (passo 15 - Tabela 8), tendo sido também necessário suavizar a borda do fluido (passo 16 - Tabela 8) para que se deixasse de notar a baixa resolução e tornar a sua densidade um pouco mais opaca (passo 17 -Tabela 8).

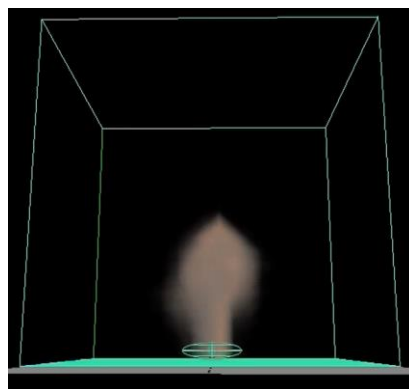


Imagem 93 – Processo intermédio da criação da explosão.

De seguida, aplicámos uma textura 3D no fluido, de forma a dar um pouco mais de detalhes sem que fosse necessário aumentar a sua resolução (passo 18 - Tabela 8). De seguida realizámos o mapeamento

dos valores da densidade e da opacidade (passo 20 - Tabela 8) e por fim, demos um pequeno movimento na textura do fluido para parecer mais realista (passo 19 - Tabela 8).

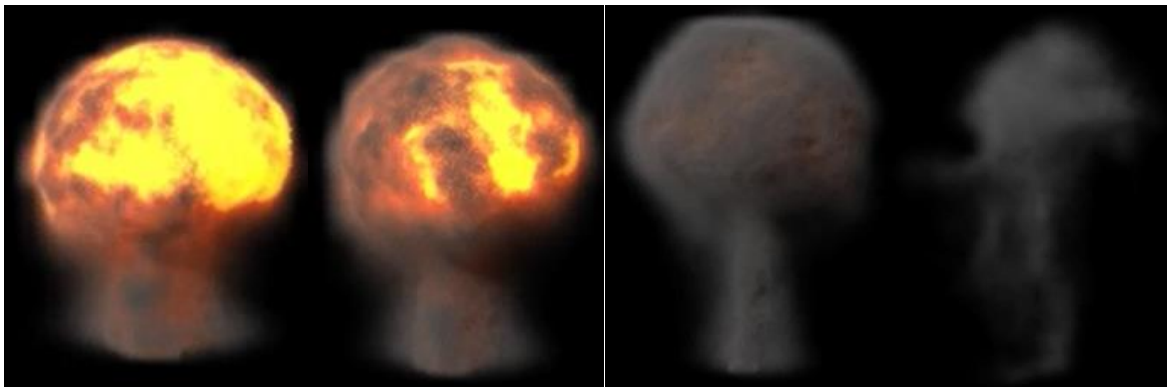


Imagem 94 - Sequencia da explosão final.

O último efeito visual que utilizámos no Maya foi a realização do sol. Consistiu na criação de uma circunferência que emitia fogo, acompanhada de uma textura. Para realizar este efeito recorreremos a um tutorial [136].

Começamos por criar uma esfera para representar o sol e adicionámos um novo material com cor preta. Foi ligada a opção *incandescente* para poder mapear uma textura do tipo *fractal* nesse canal (*texture channel*) (Imagem 95).

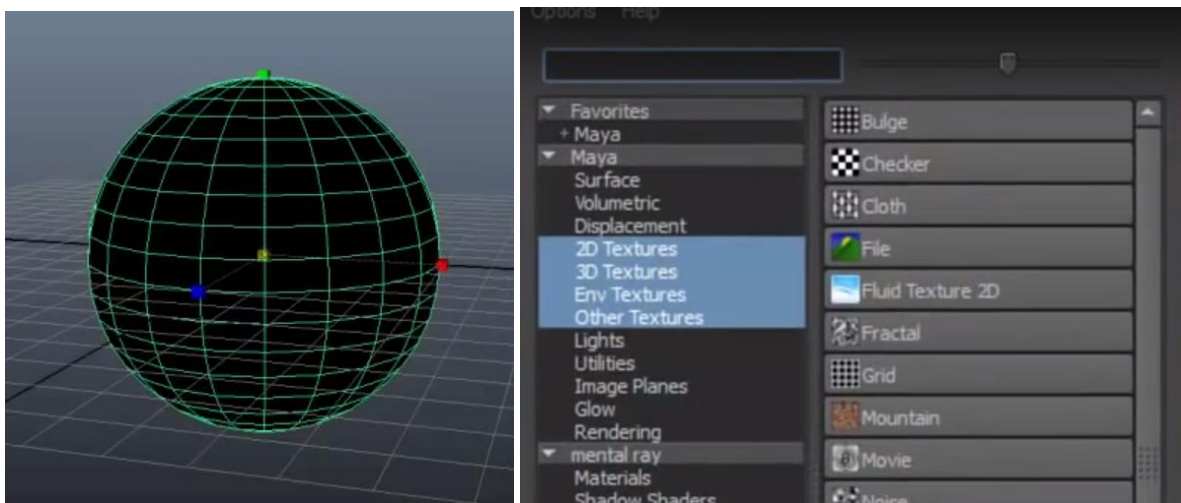


Imagem 95 – Criação do sol com a textura preta e as definições do material da textura.

Para podermos animar a textura do sol foi necessário escolher uma cor principal para a superfície (laranja) e ligar as opções *inflexion* (no menu *Fractal Attributes*) e *invert* (no menu *Effects*). Depois de ajustarmos os valores nas opções da textura *fractal* (Imagem 96 - A) e ajustar também a cor principal conseguimos obter o resultado final para a textura da superfície do sol (Imagem 96 - B).

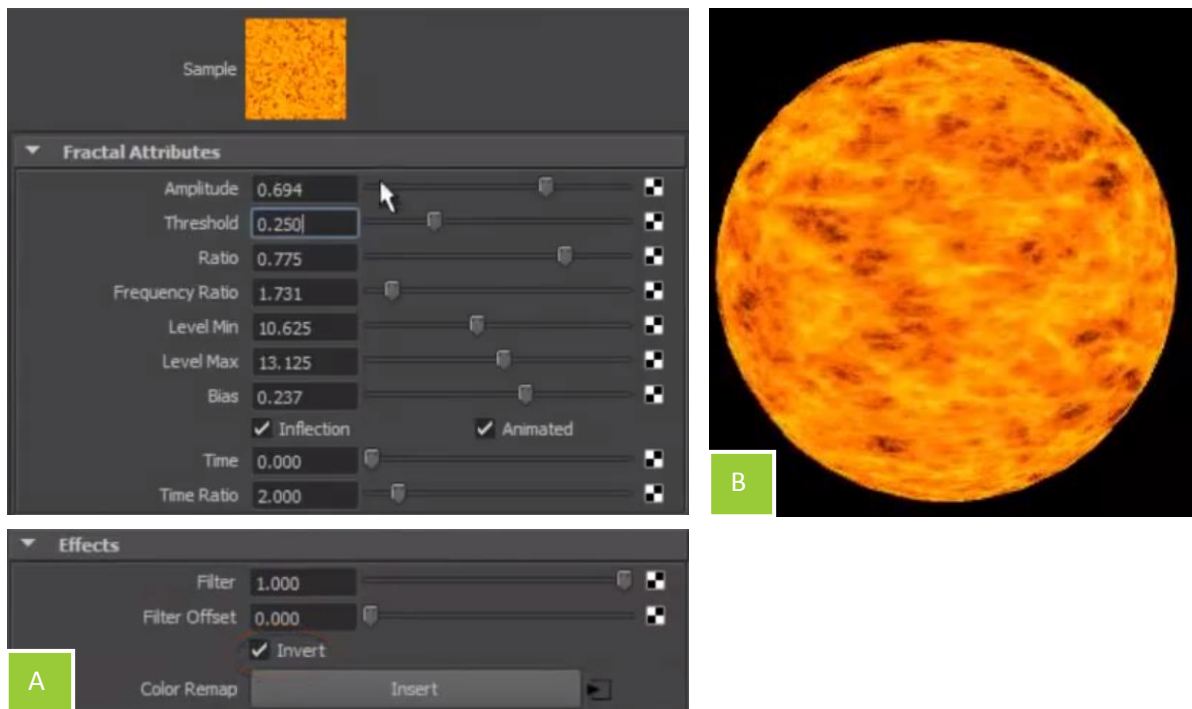


Imagem 96 – (A) Opções da textura *fractal*; (B) Render do sol.

Para podermos animar a superfície do sol de maneira a conseguirmos imitar o movimento dos gases e as zonas de maior calor, foi necessário adicionarmos uma fórmula à variável do tempo (*time*) no editor de expressões (Imagem 97 - A) para permitir a manipulação do tempo da animação ( $time/72$ ). Ao dividir por 72, garantimos que a animação iria ser 3x mais lenta ( $3 \times 24 = 72$ ; 24 representa o número de frames).

Manipulando a variável *threshold* conseguimos fazer com que o brilho da superfície solar fosse mudando conforme o tempo passava (Imagem 97 - B), tornando o sol ainda mais realista. Para tal, usámos uma expressão conhecida como '*noise(time)*' que gera valores aleatórios entre 0 e 1 com diferenças pequenas entre eles. Ao adicionar este valor aleatório ao nosso *threshold* (*threshold* era igual a 0.25) conseguimos ter valores aleatórios entre 0.25 e 1.25, o que era demasiado para a criação de um brilho suave. Para resolver este problema, dividimos o valor gerado por 10, o que retornava valores entre 0 e 0.1 fazendo com que o *threshold* não variasse acima do necessário. Ficamos assim com a expressão final tal como podemos ver na Imagem 97 – C.

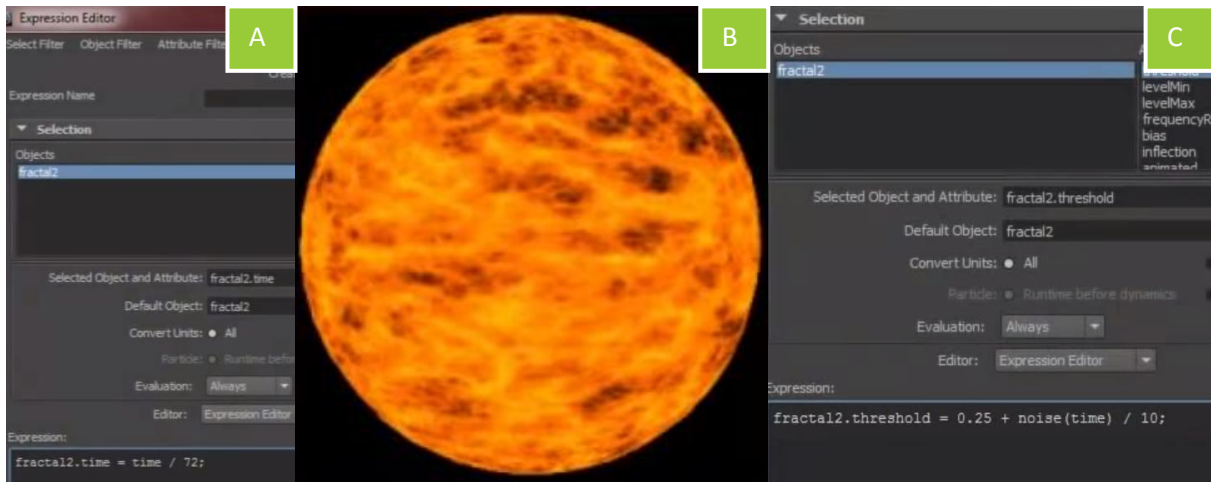


Imagem 97 – (A) Formula para animar a superfície do sol; (B) Sol com efeito de brilho; (C) Editor de expressões para o *threshold*.

Finalmente, criámos os detalhes finais da superfície do sol, tais como as explosões solares e a turbulência solar que podemos observar no sol. Para tal foi usado um sistema de partículas com efeito de fogo (Menu *Dynamics – Create Fire effect*) fazendo com que a esfera se tornasse num emissor de partículas. Infelizmente este efeito fazia com que todas as partículas se movessem para cima (no eixo do Y), criando um efeito irrealista (Imagem 98 - A). Para resolver este problema, removemos a gravidade das partículas (Imagem 98 - B) fazendo com que estas pairassem à volta da esfera movendo-se nas várias direções (Imagem 98 - C).

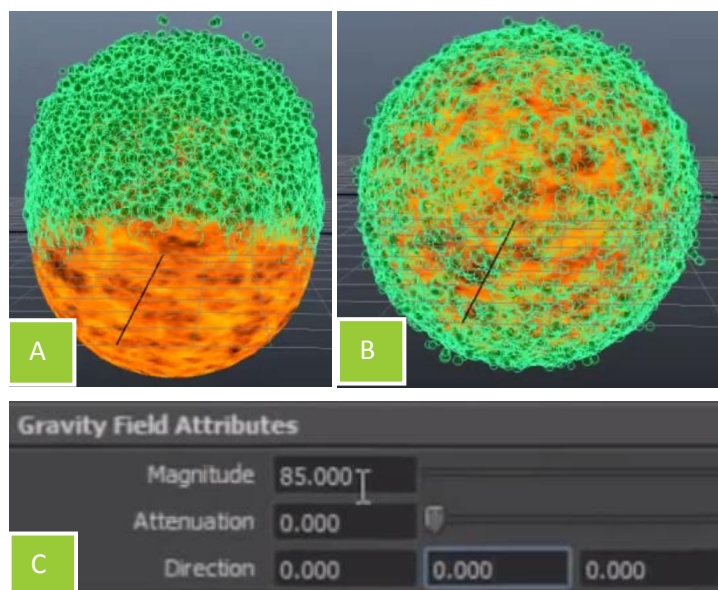


Imagem 98 – (A) Partículas no sol com gravidade; (B) Partículas a volta do sol com a gravidade desligada; (C) Definições do sol sem gravidade.

Por fim, adicionámos um efeito de turbulência às partículas de modo a que os cantos à volta do sol não fossem tão achatados, ficando semelhante a um sol verdadeiro (Imagem 99 - A). Para além do sol laranja criado usando este processo, foi criado ainda um sol roxo que pertence à outra galáxia da história como podemos ver na Imagem 99 - B.



Imagem 99 – (A) Sol verdadeiro (esquerda) e sol realizado no Maya (direita) com sistema de partícula; (B) Sol realizado no Maya com sistema de partículas em roxo.

### 3DS MAX

Tal como no Maya, os cenários criados no 3ds Max também têm efeitos visuais. Estes efeitos são todos criados através de sistemas de partículas. Neste caso só utilizámos dois tipos diferente de partículas, o *Super Spray* e o *Particle Systems*.

Utilizámos o *Super Spray* como efeito visual nos portais. O *Super Spray* é um sistema de partículas que emite a pulverização de um spray. Cada um dos portais contém quatro emissores de partículas deste tipo. Três dos emissores que criámos foram para criar a formação do portal e o quarto para permanecer no centro do portal e dar ilusão de um portal mágico.

O *Super Spray* é um sistema de partículas simples e fácil de utilizar. Neste sistema de partículas devemos definir o tipo de partículas (esferas, triângulos, cubos, etc.), a quantidade de partículas que queremos emitir, quais são os frames em que a emissão de partículas começa e acaba e o seu tempo de vida. Ao ajustar estes valores obtemos as partículas que formam o portal.

A diferença dos emissores de partículas do contorno do portal para o emissor do centro do portal é a maneira que estas são formadas. O emissor que forma o portal cria as partículas do tipo *tetra* com uma abertura de 14 graus, enquanto que o emissor central do portal cria esferas com uma abertura de 180 graus (Imagem 100).



Imagem 100 – Sistema de partículas *Super Spray*. (A) Emissor de partículas a volta do portal; (B) emissor de partículas central.

Estes quatro emissores foram iguais para os cinco portais. Para os diferenciar, utilizámos materiais com cores diferentes que correspondem a cada portal. Para criar o brilho das partículas utilizamos os efeitos

do *Environment and Effect*. Aqui adicionamos o efeito *Lens Effects* onde colocamos o brilho (*Glow*) das partículas e um efeito de anel (*Ring*) à volta de cada uma delas. Para identificarmos qual o material ao qual queríamos adicionar o brilho, foi criado um ID no material de maneira a poder diferenciar onde é que o efeito iria atuar (Imagem 101).

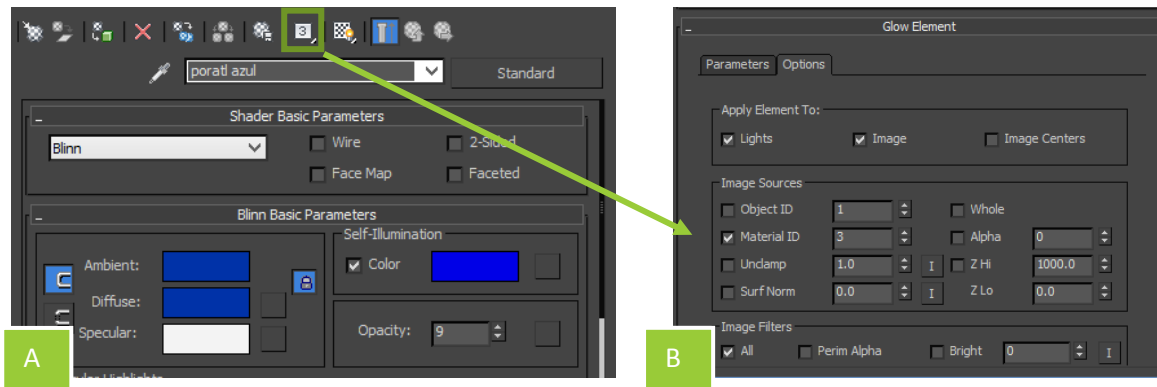


Imagem 101 – Identificação de ID do material. (A) Janela *Material Editor*; (B) Janela *Environment and Effects*.

Renderizando as partículas, podemos ver um ambiente com brilho. O efeito das partículas centrais lembra-nos um portal mágico (Imagem 102).



Imagem 102 - Render com sistema de partículas *Super Spray*.

O ultimo efeito visual do projeto, também realizado no 3ds Max, chama-se *Particle Systems (PFSource)*. O *Particle Systems* funciona como um emissor de um padrão, que por defeito vem com um retângulo, tendo a possibilidade de alterar a forma e aparência. Este sistema de partículas foi utilizado em algumas situações no 3ds Max, tais como no efeito luminoso no chão, no centro do templo, no fumo da nave e no rasto do robô. Apenas as texturas e as definições variam para cada uma destas cenas. Para realizarmos o sistema de partículas que cria ilusão de luz no centro e chão do templo recorreremos a um tutorial [137]. A primeira coisa que fizemos foi criar uma linha que delineasse o movimento e a direção que as partículas iriam tomar. Logo, criamos um *PFSource* e definimos a sua localização no *pivot*. Para que o sistema de partícula seguisse a linha, foi necessário utilizar o *Path Constraint* e definir o tempo de movimento através de *key frames*.



Imagem 103 – Ligação do *PFSource* com a linha através do *Path Constraint*.

Este sistema de partículas vem com uma opção (*Particle View*) onde definimos, adicionamos ou removemos todos os argumentos para ajustar as partículas. Aqui definimos que o sistema de partículas que estávamos a implementar iria seguir o tipo geométrico quadrado e em 2D. Nesta opção também podemos definir em que *frame* o emissor de sistema de partículas irá começar e acabar e o tempo de vida que cada partícula tem. Uma preocupação importante nesta fase é colocarmos com que frequência as partículas nascem, sendo que quanto maior for o número, mais definido será o seu efeito (Imagem 104).

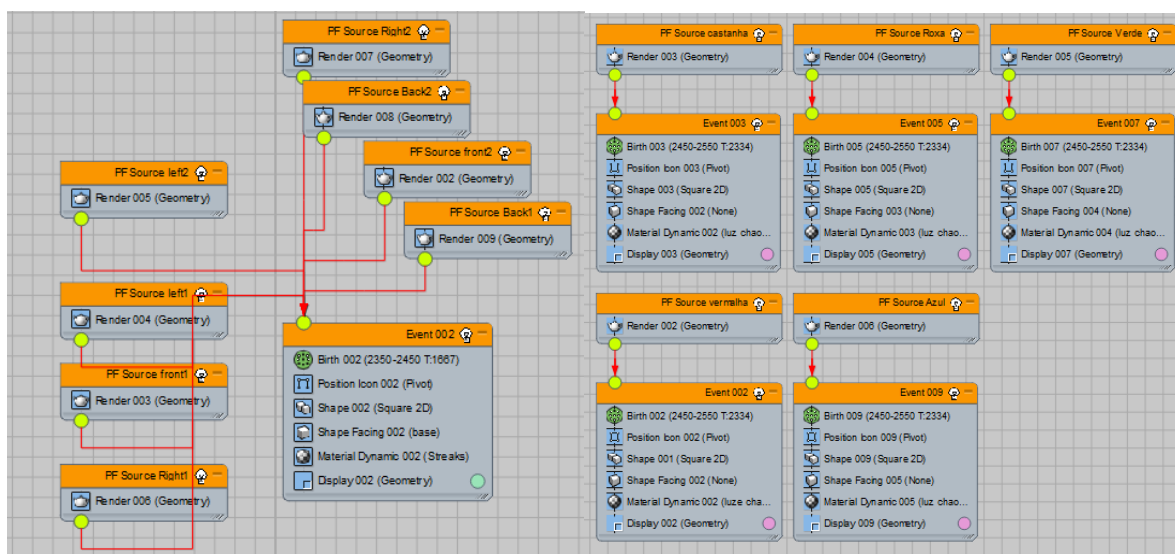


Imagem 104 – Janela *Particle View* com todas as configurações dos vários sistemas de partículas *PFSource*.

Em seguida demos cor às nossas partículas. Como o *PFSource* pode ser utilizado para produzir um padrão, decidimos então criar um padrão no Photoshop. A imagem que define o padrão a seguir deve ser simples, de cor branca e fundo preto. Colocamos esta imagem no material do 3ds Max e definimos a cor e a opacidade que queríamos que ela emitisse (Imagem 105).

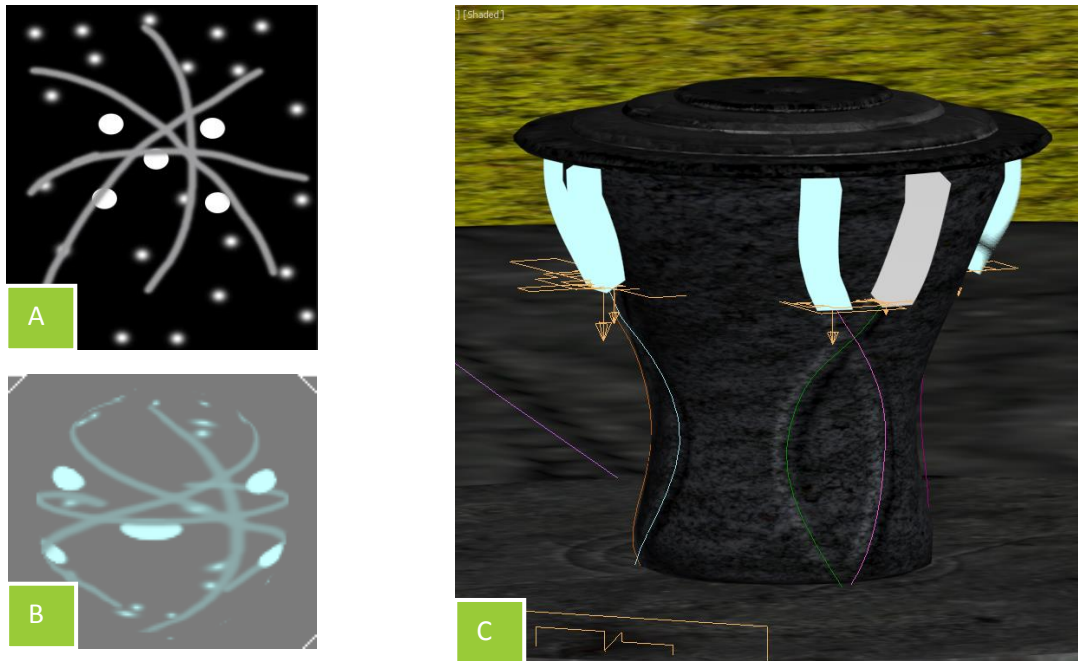


Imagem 105 –(A) Imagem com a padrão criado no Photoshop; (B) Material do sistema de partícula *PFSource*; (C) Prova do material no sistema de partículas.

Adicionámos este material ao sistema de partículas e, a partir desse momento, já se consegue visualizá-las. À semelhança do que foi feito no sistema de partículas dos portais, aqui também foi necessário adicionar brilho. A única diferença entre o sistema de partículas anterior e este, é que neste apenas adicionámos o brilho (Glow) ao material.

Todo este procedimento foi realizado para as cinco linhas que se encontram no chão do templo e no centro (Imagem 106).

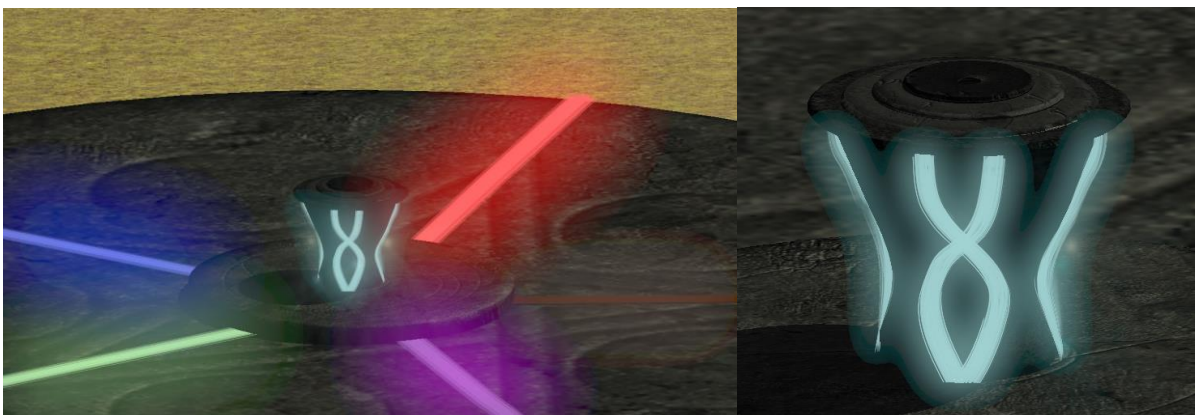


Imagem 106 - Render do *Particle Systems*.

Este mesmo sistema de partículas também serviu para produzir o fumo da nave e o rasto do robô a cair. Para elaborar estes dois sistemas de partículas bastou-nos ajustar alguns parâmetros e substituir a textura por uma que se adequasse a cada uma das situações (Imagem 107).

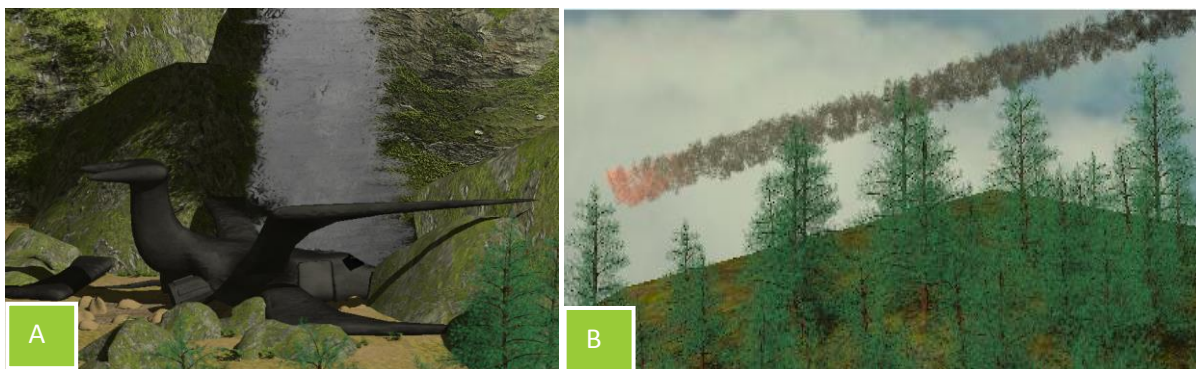


Imagem 107 – Partícula *Particle Systems*. (A) Render do fumo; (B) Render rasto da caída do robô.

## CÂMARAS

Todos os programas de animação 3D contêm câmaras para podermos animar e capturar várias cenas dos mais variados ângulos. No 3ds Max existem dois tipos de câmaras, *target* e *free*. Neste projeto utilizamos os dois tipos. A câmara *target* foi utilizada para capturar a animação dos guerreiros a sair no portal. Esta câmara tem a vantagem de mover apenas o alvo da câmara sem que seja necessário deslocar a câmara, o que permite que a posição da mesma não seja afetada. Esta funcionalidade ajudou-nos a realizar a animação dos portais. Centrando a câmara no centro do templo, apenas foi necessário mover o alvo para realizar a animação (Imagem 108).

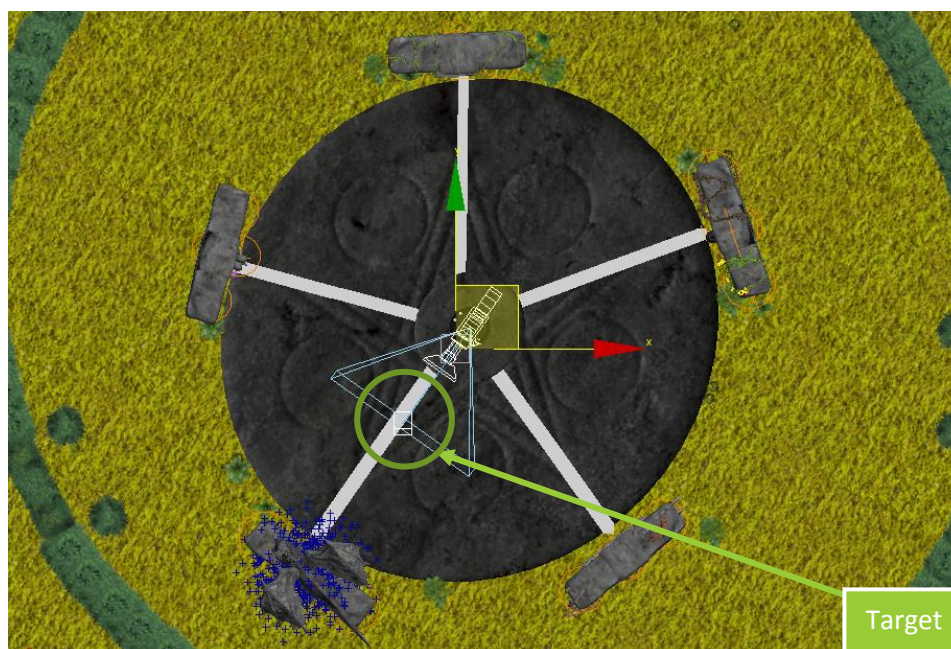


Imagem 108 – Câmara *target* no cenário dos portais.

A câmara *free* foi escolhida para fazer todas as outras gravações das animações. Uma das animações realizadas com esta câmara foi a apresentação da vila. Adicionar uma animação a uma câmara é muito simples. Apenas devemos posicioná-la na posição inicial da animação e, com ela selecionada, criamos um *key frame*. Desta forma, apenas é necessário movimentar a câmara ao longo da cena e na linha temporal criamos os *keys frames*.

Outro tipo de animação de camara foi quando a associámos a um esqueleto em andamento. Ao fazer esta associação a camara fica na perspectiva da primeira pessoa, imitando o movimento de andamento (Imagem 109). A animação de andamento é explicada na secção 3.2.2.2.



Imagem 109 - Ligação da camara a uma animação em andamento, dando sensação de movimento na primeira pessoa.

## LUZES

As luzes são uma parte fundamental nas cenas. Geralmente são colocadas na última etapa da produção. Neste projeto colocámos as luzes conforme fomos montando a cena de maneira a tê-la o mais completa possível e só ser necessário incorporar as animações das personagens. Durante a montagem das cenas definimos a luz ambiente e a luz solar, mas depois de adicionarmos as personagens e termos realizados alguns *renders* de prova, constatamos que estas luzes não eram suficientes para iluminar as cenas em questão. Mais à frente é explicado o processo que foi realizado para ultrapassar este problema.

As luzes no projeto foram criadas no software 3ds Max, visto que as cenas realizadas no Maya eram no espaço e não precisavam de iluminação. Todas as luzes criadas no 3ds Max são do tipo *standard*, estas luzes podem simular vários tipos de luzes tais como lâmpadas ou até o próprio sol. A luz *Omni* foi uma das luzes *standard* utilizadas no projeto e a escolhida para imitar o sol. Esta luz foi utilizada para iluminar todas as cenas do nosso projeto porque lança raios para todas as direções utilizando apenas uma única fonte de luz. Configurando esta luz foi possível projetar, nos nossos cenários, sombras. Para as sombras escolhemos o tipo de sombra *Ray Traced Shadows*.

Depois disso, reparamos que o nosso cenário estava mais escuro que o pretendido e as sombras projetadas também estavam um pouco mais escuras do que o esperado, apenas da luz *Omni* estar ligada (Imagem 110 - A). Para corrigir este problema, foi necessário adicionar uma luz ambiente ao cenário (Imagem 110 - C). Aqui definimos a cor que a luz ambiente iria refletir na cena tornando o cenário e as sombras um pouco mais claras (Imagem 110 - B).

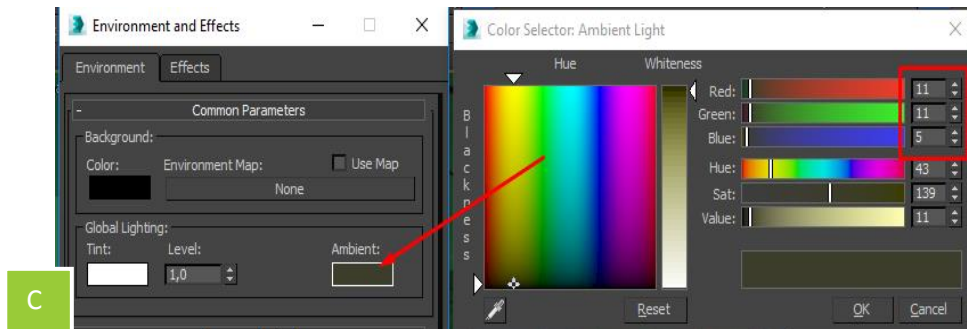


Imagem 110 – (A) Cenário sem luz ambiente; (B) Cenário com luz ambiente;(C)Luz ambiente das cenas.

Esta foi a única luz colocada na hora da montagem dos cenários. Quando chegámos à última parte da etapa de produção, reparámos que existia uma falta de luz tanto nas personagens como em alguns objetos. Uma das coisas que detetámos na hora da realização dos *renders* de prova foi a dificuldade em ver as linhas do rosto das personagens pois, como estava escuro, perdíamos muito detalhe do rosto e, conseqüentemente, da animação. Por isso, adicionámos uma luz *Target Direct* para as iluminar (Imagem 111). Ao utilizarmos esta luz não estávamos a iluminar apenas a personagem, mas também tudo o que se encontrava ao alcance da luz, o que se tornava um pouco estranho. Para solucionar isso, configurámos a luz de maneira e excluir a iluminação de todos os objetos que não interessavam.



Imagem 111 –Rendes de Animação facial da personagem Isene. (A) Sem luz *Target Direct* ;(B)Com luz *Target Direct*.

A última luz que adicionamos no 3ds Max foi a *Target Spot*. Esta luz realiza o mesmo efeito que uma lanterna ou um farol, lançando um feixe de luz. Utilizamos esta luz principalmente para iluminar os detalhes dos portais que, pelo posicionamento da camera contra a luz *Omni*, encontravam-se mais escuros e, conseqüentemente, dificultando a visualização dos detalhes dos portais (Imagem 112). E da mesma maneira que excluimos todos os objetos que não queremos iluminar com a luz anterior, também foi necessário realizar o mesmo procedimento aqui.



Imagem 112 – Renders do portal dos Assassinos. (A) Sem luz *Target spot*; (B) Com luz *Target spot*.

Com estes três tipos de luzes iluminamos todo o cenário, deixando bem visíveis todos os detalhes da modelação e da animação realizada nas personagens.

### 3.2.2. ANIMAÇÃO

Há vários tipos de animações que podem ser realizadas num projeto 3D. Neste caso iremos dividir as animações em duas partes. Na captura de animação através da Kinect e todos os outros tipos de animações manuais (Imagem 113).



Imagem 113 - Pipeline de animação.

### 3.2.2.1. CAPTURA DA ANIMAÇÃO

Para realizarmos a captura da animação para este projeto foram necessários alguns dias de pesquisa e provas para chegar à *pipeline* que se encontra abaixo (Imagem 114).

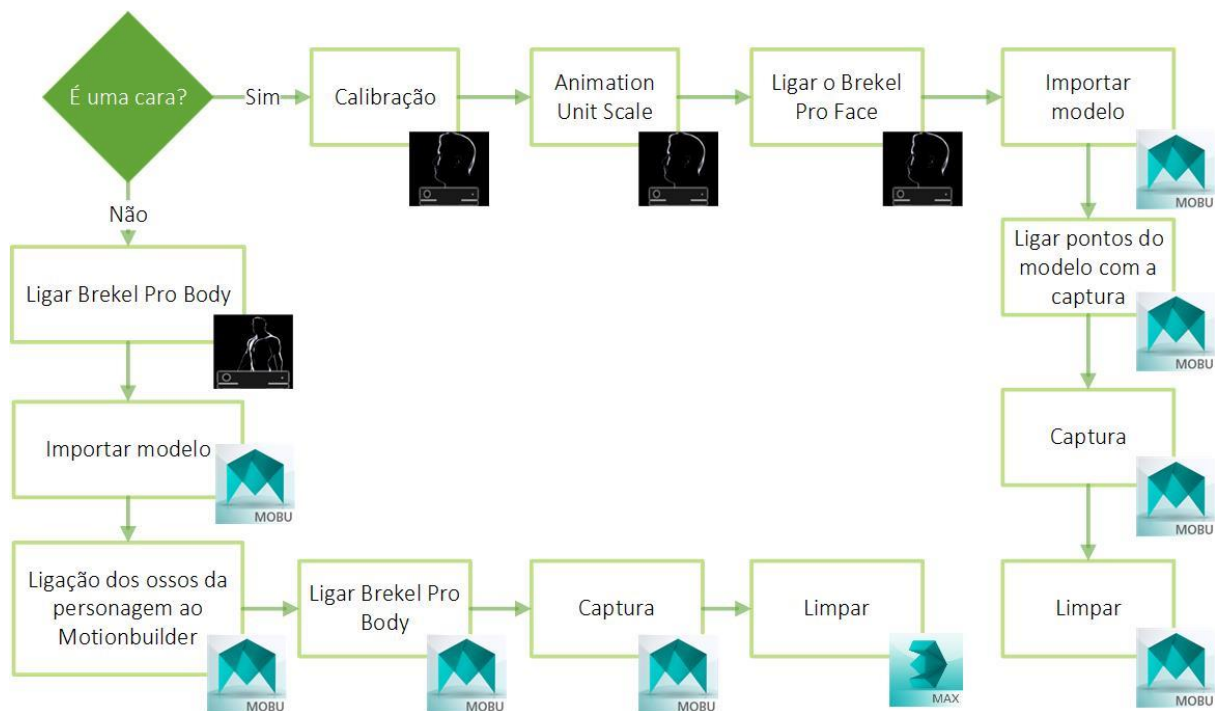


Imagem 114 - Pipeline de animação com captura de movimento.

Como podemos ver na Imagem 114, a animação de captura de movimento foi dividida em duas partes diferentes: animação corporal e animação facial, isto porque os *softwares* de captura são diferentes e os seus procedimentos também. Os dois tipos de capturas de animação são realizados com a Kinect.

Qualquer um dos *softwares* onde foram realizadas as animações finais trabalham com *frames*. Transformar estes *frames* em segundos, varia com as configurações que definimos. O nosso projeto está configurado com o formato *Phase Alternating Line* (PAL) que consiste em 25 *frames* por segundo, sendo este o mais utilizado para reproduzir vídeos na Europa. Outro formato muito utilizado é o *National Television System Committee* (NTSC), utilizado para reproduzir vídeos na América e no Japão. Este formato consiste em ter 30 *frames* por segundo.

### ANIMAÇÃO CORPORAL

Para realizarmos a captura de movimento corporal foi necessário utilizar três *softwares* diferentes: Brekel Pro Body, MotionBuilder e 3d's Max.

Os passos executados para realizar este processo não têm de ser os definidos na nossa *pipeline*. No entanto, o procedimento seguido foi o mais adequado e eficiente tendo em conta as ferramentas disponíveis e o resultado final pretendido.

Para realizarmos a animação corporal de uma personagem, começámos por ligar a Kinect com o Brekel Pro Body. Com esta ligação pudemos ver na interface do Brekel Pro Body a captura dos nossos

movimentos, sendo representado por um esqueleto e sem ser necessário nenhum tipo de calibração (Imagem 115).

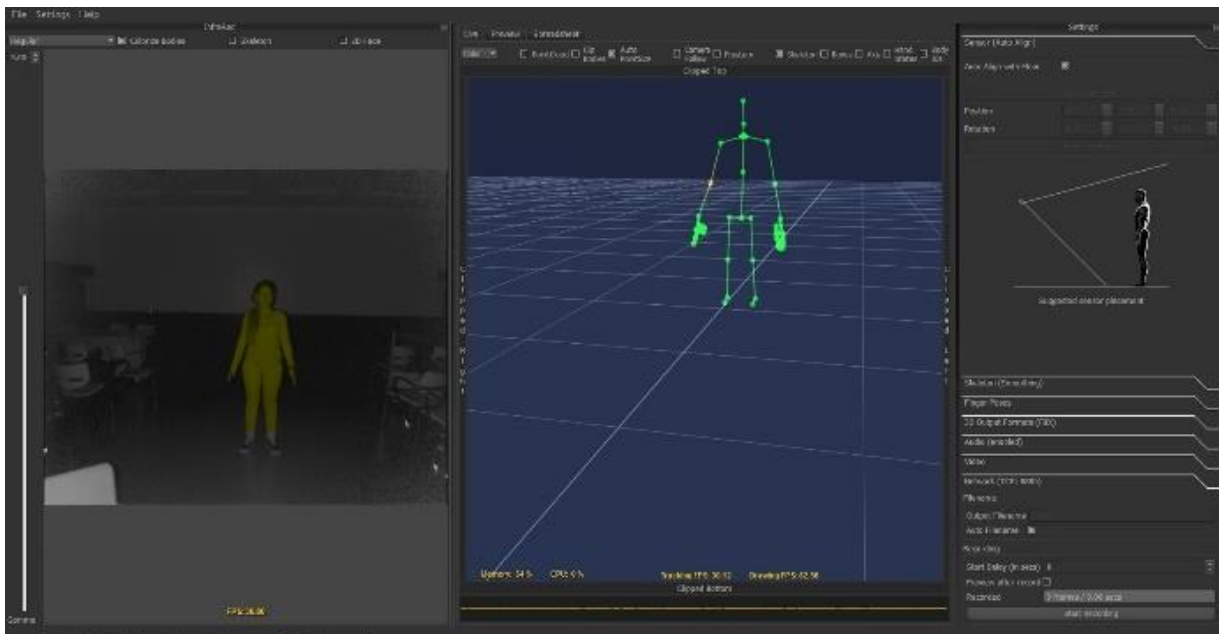


Imagem 115 - Captura de movimento através do Brekel Pro Body v2 e Kinect 2.

O passo seguinte levou-nos a duas maneiras diferentes de realizar a animação. Existe a opção de realizar a animação já na personagem que queremos animar ou, então, podemos capturar a animação que é capturada pela Kinect de maneira a mais tarde juntar a qualquer personagem criada.

No nosso caso, optamos por realizar a animação já na personagem, de maneira a poder identificar mais facilmente algum problema que surja durante a captura de movimento. Depois de escolher a personagem a ser animada, esta foi importada para o MotionBuilder. Esta personagem já deve ter todo o processo de *rigging* e *skinning* realizado, para poder ser animada. Através da janela *Character Control* podemos definir o *control rig* da personagem, interligando todos os ossos da personagem (Imagem 116). Apenas quando o círculo que se encontra no lado superior direito estiver verde o processo de *control rig* está concluído.

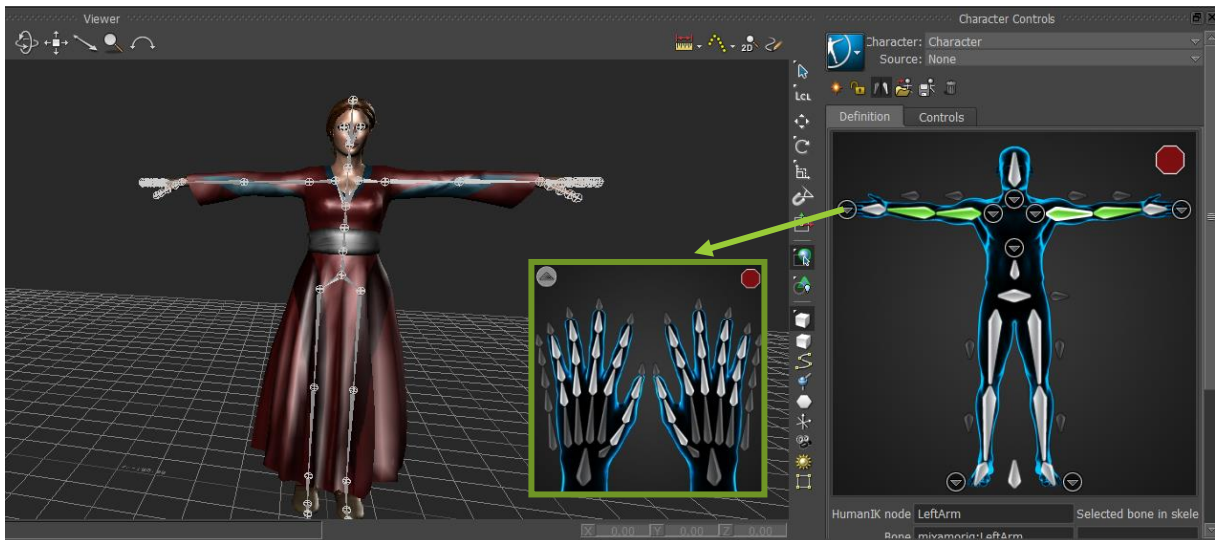


Imagem 116 - Control rig da personagem Isene no MotionBuilder.

O Brekel fornece um plug-in para ser instalado no MotionBuilder que realiza a conexão entre os dois softwares e assim podemos realizar todo o tipo de captura de animação neste. Ao ligar o plug-in do Brekel no MotionBuilder, aparece-nos, na barra de navegação, a opção *Devices*, onde ligamos o MotionBuilder ao Brekel. A primeira coisa que fizemos para ter a ligação entre os dois softwares foi criar um *model binding*, que será a referência que o brekel irá utilizar e, em seguida, ligámos as opções *online*, *live* e *recording* (Imagem 117). Com estes passos o MotionBuilder já está pronto para começar as gravações das animações capturadas.

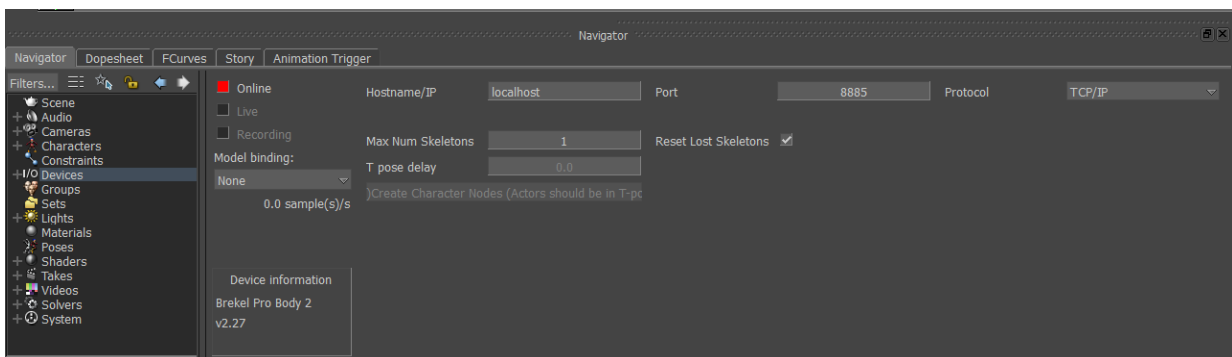


Imagem 117 - Ligação do Brekel Pro Body ao MotionBuilder.

Para gravar a captura de movimento basta verificar que nos encontramos no *frame* um e podemos começar a gravação clicando no botão *recorder* e *play*. Isto irá gravar todos os movimentos realizados até parar a gravação (Imagem 118).

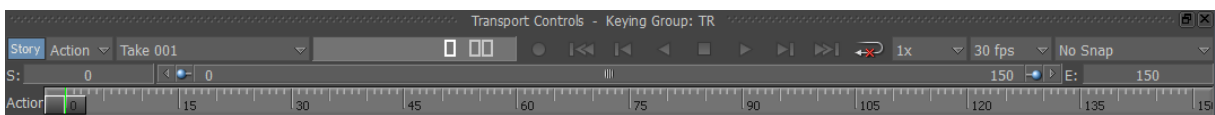


Imagem 118 – Barra Transport Controls.

A captura de movimento é guardada *frame a frame*, ou seja, em todos os frames da nossa gravação é criada uma *key* com todos os movimentos que os ossos fizeram. Como estamos a falar de captura de movimento, há sempre pequenas falhas durante a animação, algum movimento indevido ou até

alguma falha na captura o que implica movimentos involuntários. Para remover estes erros, importámos a animação para o 3ds Max para limpar todas as imperfeições que ocorreram.

Há duas maneiras de limpar estas imperfeições, ou ruídos, uma delas é utilizando o *curver editor* do 3ds Max. Através das curvas podemos ver se existe algum pico na animação que esteja a provocar algum movimento indesejado ou até podemos suavizar algum movimento que a personagem esteja a realizar (Imagem 119).

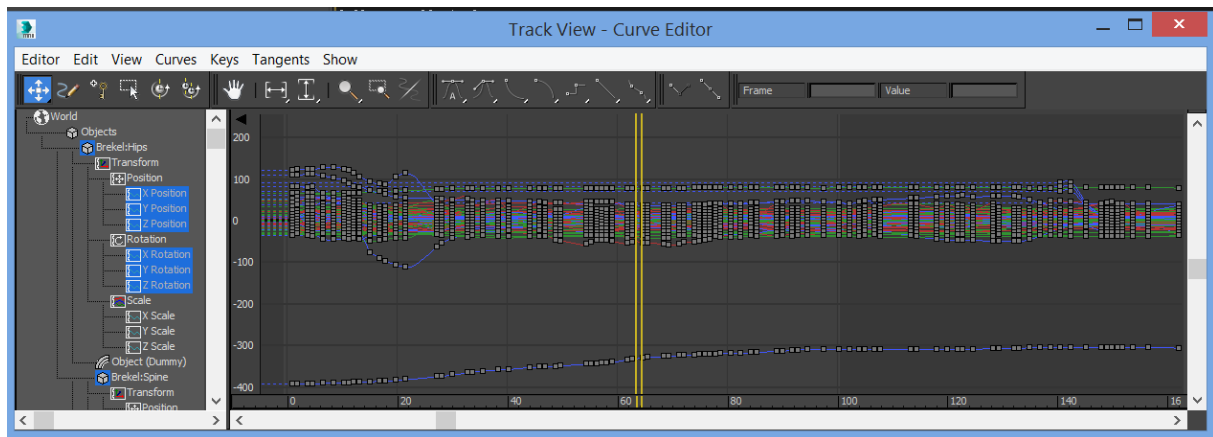


Imagem 119 - Curver editor da captura de animação corporal.

A outra maneira de corrigir a animação é mais trabalhosa. Antes de realizarmos alguma mudança na animação, a primeira coisa que temos de realizar é converter o esqueleto que vem com a personagem no esqueleto do 3ds Max (bipedo) através de um script fornecido pela Mixamo [138].

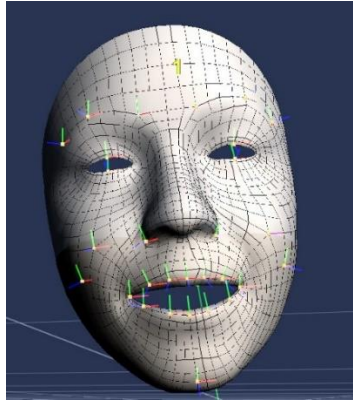
Depois de convertermos o esqueleto, podemos corrigir a animação através do separador *motion* no 3ds Max. Criando uma nova *layer* de animação, podemos corrigir pequenos movimentos indesejados ou até adicionar outras animações ou posições diferentes. Desta forma, a personagem fica pronta para ser colocada no seu cenário correspondente.

## ANIMAÇÃO FACIAL

---

Tal como a animação corporal, para realizarmos a captura da animação facial também é necessário software específico. Neste caso utilizámos o Brekel Pro Face e o MotionBuilder.

A animação facial é um processo trabalhoso e um pouco demorado pela grande quantidade de pontos que são capturados através da Kinect (Imagem 120).



**Imagem 120 - Pontos faciais capturados pelo Brekel Pro Face v2.**

Para a utilização correta do Brekel Pro Face v2, temos de calibrar o nosso rosto para ter um melhor feedback na hora da captura. Outro ponto importante é ajustar os valores da opção *animation unit scale/offset* que nos permite ajustar uma expressão neutra e sincronizar automaticamente. Isto é, caso estejamos a fazer uma expressão neutra e o Brekel Pro Face não a detetar, podemos ajustar os valores ate que estejamos satisfeitos com o rosto que se encontra na interface. Da mesma maneria que tínhamos um plug-in para ligar o Brekel Pro Body ao MotionBuilder, também temos um plug-in para fazer o mesmo com o Brekel Pro Face e assim fazer a ligação entre os dois *softwares* (Imagem 121) Fazer a ligação do Brekel Pro face ao MotionBuilder é parecido ao do corpo. Ativamos as opções *online*, *live* e *recording*. Criamos um *model binding* e um *character face*, isto criará um *character face* para podermos fazer as correspondências dos pontos mais a frente.

De seguida, importámos a personagem criada no Fuse para o MotionBuilder, tal como o corpo a personagem que está a ser importada para o MotionBuilder deve ter o rigging facial.

Ao ligar o Brekel anteriormente, criámos um *character face*. Podemos encontra o *character face* na barra de navegação. É com esta opção que vamos fazer a ligação dos pontos do rosto da personagem ao MotionBuilder. No *character face definition* podemos encontrar todos os pontos que o MotionBuilder deteta (separador *custom*) e também todos os pontos que o rosto da nossa personagem tem (separador *shapes mapping*) (Imagem 121). A única coisa que temos de fazer é arrastar o rosto da personagem para a opção *models with sha* e a tabela será preenchida com todos os pontos que a nosso rosto tem.

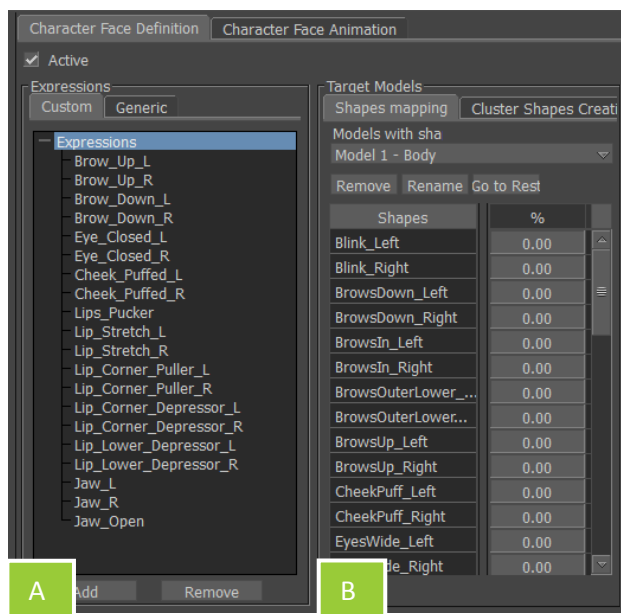


Imagem 121 – Separador Character Face Definition. (A) Pontos faciais do MotionBuilder; (B) Pontos faciais da personagem.

O próximo passo é ligar os pontos do separador *custom* (pontos do MotionBuilder) com o separador *shapes mapping* (pontos do rosto da personagem). A cada ponto do separador *custom* temos de dar uma certa percentagem ao ponto correspondente no separador *shapes mapping*. Por exemplo: para o ponto “*Brow\_Up\_L*” no separador *custom*, demos a percentagem de 100% ao ponto “*BrowsUp\_Left*” no separador *shapes mapping*. Por fim, é só ativar a opção *Active*.

E assim, a nossa personagem está pronta para realizar a captura de movimento. Tal como na animação do corpo, para começarmos a gravação da animação temos de verificar se nos encontramos no frame um e assim começar a gravar clicando no botão *recording* e *play* (Imagem 122).

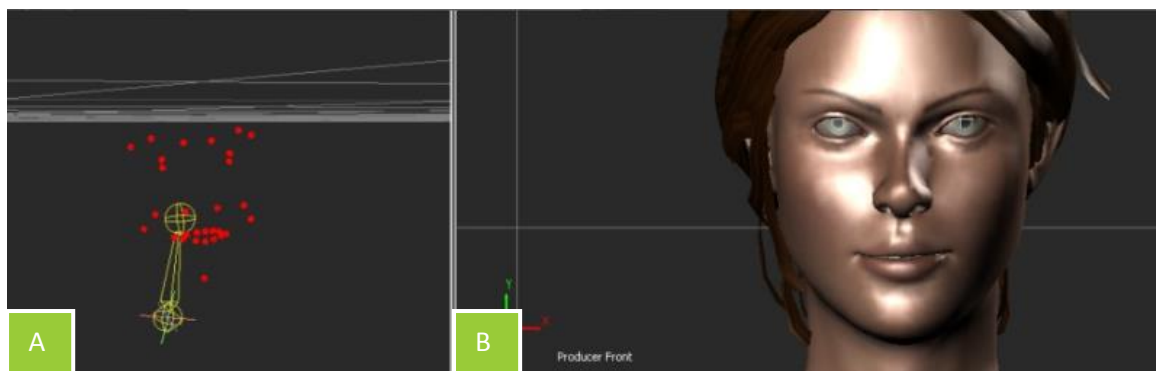


Imagem 122 – (A) Pontos capturados pelo Brekel no software Motion Builder; (B) Captura de movimento facial.

Tanto no rosto como no corpo também existe imperfeições na gravação da animação. Estas distorções ou ruídos são chamados de *Jitter* [139]. Esta interferência é normal nas capturas de movimento que utilizam tecnologia *markerless*, que é o nosso caso com o Kinect. Ao fazer a nossa captura de movimentos encontramos muitas imperfeições por isso tivemos de limpar a animação.

Através do MotionBuilder realizámos todo o procedimento de limpeza da animação facial utilizando o filtro *butterwoth*. O *butterwoth* remove o ruído dos dados sem afetar os valores mínimos e máximos

do gráfico *FCurves* (Imagem 123). O filtro evita os problemas que podem acontecer quando filtramos os dados da captura de movimento.

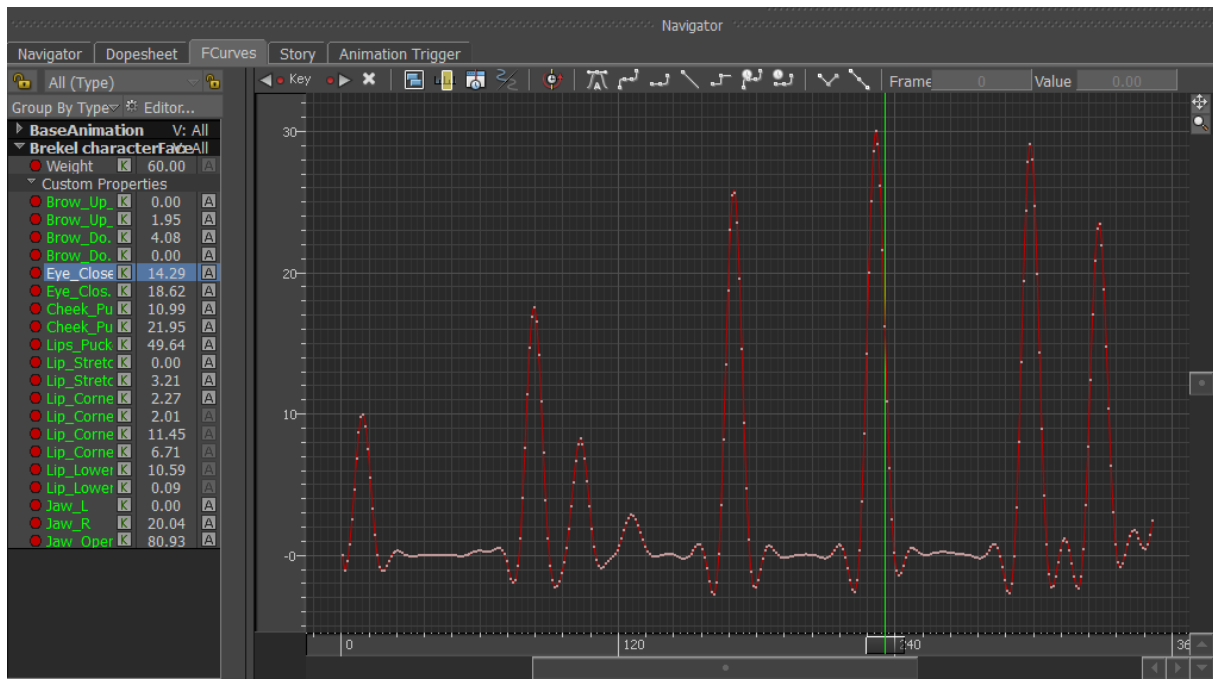


Imagem 123 - Corrigir imperfeições faciais através do *FCurves* no MotionBuilde.

Este procedimento de limpeza deve ser realizado em todos os pontos faciais da nossa personagem. Depois de limparmos todas as imperfeições, a animação facial captada através da captura de movimento está concluída e pronta para ser colocada na cena.

### 3.2.2.2. CRIAÇÃO DE ANIMAÇÕES

Como referido anteriormente, o projeto foi dividido por dois *softwares* de animação 3D, o Maya e o 3Ds Max. Como é de esperar, em cada um deles foi necessário realizar diferentes animações.

#### 3DS MAX

Na animação das personagens, nem todas as animações realizadas com estas personagens foram através da captura de movimentos. Sabendo que existem vários métodos para animar, utilizámos alguns métodos diferentes na animação das personagens e objetos.

Outra das opções que escolhemos para realizar a animação de uma personagem foi importar uma animação pré-feita que a Mixamo fornece (que também foi criada a partir de captura de movimento) [140]. A animação que escolhemos foi uma animação de uma pessoa a andar sempre no mesmo lugar (Imagem 124). Escolhemos uma animação estacionária para facilitar a realização do *loop* no andamento da personagem.

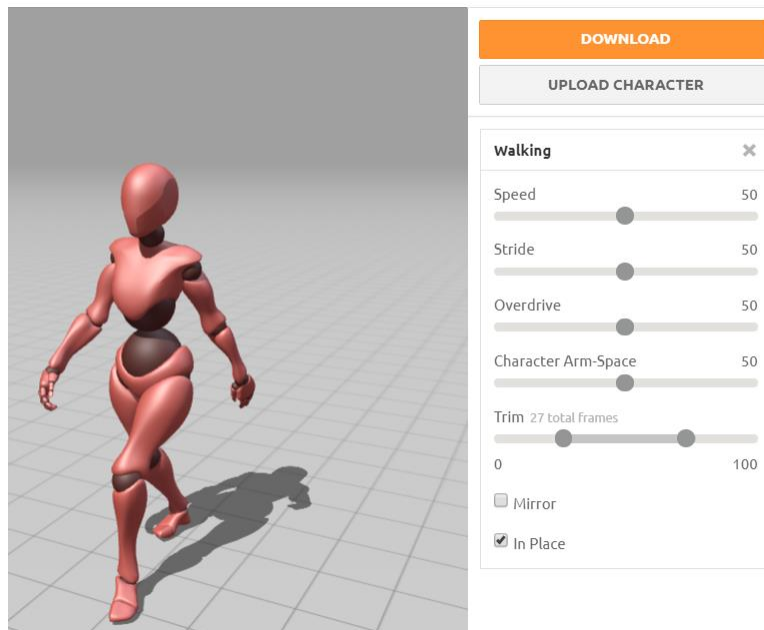


Imagem 124 - Animação de andamento no Mixamo [140].

Depois de realizarmos o *download* e colocarmos a animação na personagem que escolhemos, iniciamos a animação da personagem no cenário, este processo é realizado no software 3D's Max.

Como está a ser usada uma animação estacionária, para dar o efeito de deslocamento, utilizámos o *Path constraint*. O *Path constraint* consiste em restringir um objeto a um percurso e a distância que este irá percorrer. Deste modo, podemos definir o percurso que a nossa personagem irá tomar. Desenhando uma linha à altura da cintura da personagem definimos o seu deslocamento. Selecionando o osso da cintura da personagem (*hips*) e recorrendo ao menu *Motion*, podemos adicionar um controlador do tipo *path constraint*. Nos parâmetros desta opção adicionamos a linha anteriormente criada e selecionamos a opção *follow* de maneira a que a personagem a siga (Imagem 125). Como a personagem se desloca através da linha, juntamente com a animação de andamento, dá a sensação que a personagem está, efetivamente, a andar.



Imagem 125 – Animação da personagem utilizando o *Path constraint*.

Através do esqueleto do 3ds Max também podemos realizar animações, de andamento e corrida, utilizando o *footstep*.

O *footstep* cria vários passos na cena que o esqueleto irá seguir. Cada passo pode ser movimentado e rodado de maneira a que possa mudar de direção e até mudar a distância entre cada passo. O *footstep* permite criar passos que façam com que o esqueleto possa andar normalmente, correr ou até saltitar (Imagem 126).

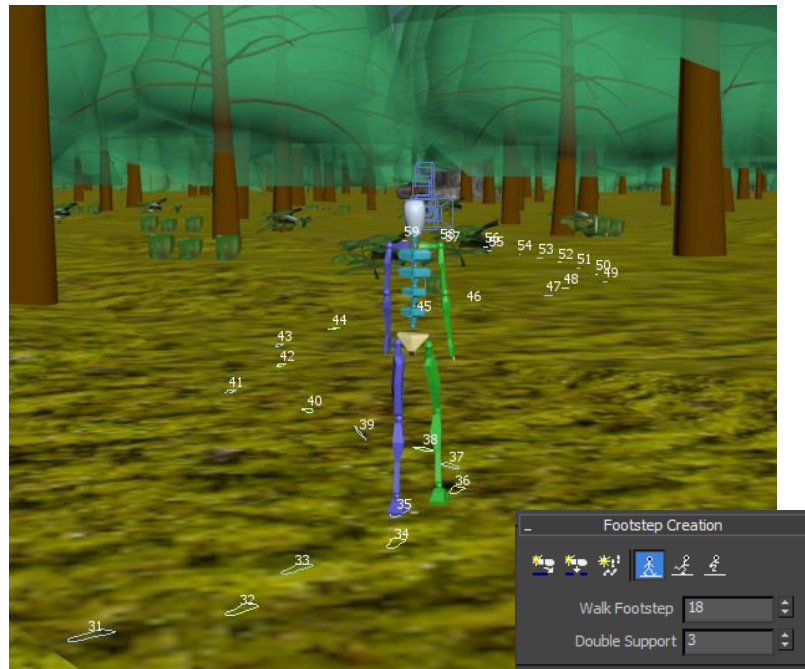


Imagem 126 - Animação de andamento do esqueleto no 3ds Max, utilizando o *footstep*.

Outras animações que realizamos manualmente foram animações de pequenos movimentos das personagens, toda a animação do urso e também algumas animações de objetos. No caso das personagens, animamos manualmente alguns pequenos movimentos, tais como mãos, a cabeça, o equilíbrio do corpo, entre outros. Em relação aos objetos animados temos o exemplo da animação da porta da nave. No caso de quisermos animar o movimento da mão, o que temos de fazer é selecionar o osso da mão da personagem e selecionar o *frame* no qual queremos iniciar a animação. Colocamos a mão na posição inicial e criamos um *key frame* com o posicionamento da mão da personagem (Imagem 127 - A). Desta mesma forma, vamos criando o movimento que queremos que a mão da personagem realize e, simultaneamente, criando *keys frames* com a sua deslocação. Quando acabamos de realizar todo o movimento pretendido, podemos visualizá-lo e modificá-lo.

A animação do urso foi realizada da mesma forma que as animações dos pequenos movimentos das personagens (Imagem 127 - B). Esta era a nossa única opção para animar o urso, visto que esta animação não era possível de realizar com a Kinect. O urso tem um número de ossos maior que o de uma personagem, o que torna a sua animação um pouco mais complicada e não tão natural.



Imagem 127 – (A) Animação manual de uma personagem; (B) Animação do urso realizada manualmente.

Uma das desvantagens que encontramos ao animar a personagem Isene foi o seu vestido. Ao realizar animações simples como o ato de andar podíamos ver como o seu vestido esticava nas pernas e como a textura ficava distorcida (Imagem 128 - A). Por isso, no 3Ds Max, adicionámos o modificador *Cloth* ao vestido de maneira a converte-lo em tecido.

Este modificador simula a roupa e a interação que esta tem sobre um objeto no mundo. Ao adicionarmos o modificador ao vestido, podemos definir quais são os objetos que fazem parte da simulação, sendo esses considerados objetos de colisão. Isto também permite-nos definir o tipo de tecido que queremos utilizar para a simulação (seda, couro, entre outros).

Quando definimos todos os parâmetros, passamos para a fase de simulação do tecido. Antes de realizar qualquer animação e simular ao longo do tempo, realizámos uma simulação local do tecido, para que este se ajustasse ao corpo da personagem (Imagem 128 - B).



Imagem 128 – Personagem Isene. (A) Antes do modificador *cloth*; (B) Depois do modificador *cloth*.

## MAYA

---

O Maya foi o software escolhido para realizar todas as cenas do espaço. E tal como ocorre no 3Ds Max, foi necessário animar os nossos objetos nas cenas.

A diferença de animar no Maya e no 3Ds Max não são muitas. A animação do meteorito e da nave foram realizadas exatamente da mesma maneira que animamos o urso ou os pequenos movimentos das personagens.

Uma animação parecida à que realizamos no 3Ds Max, foi a animação da órbita dos planetas em torno do sol. Os planetas giram em torno do Sol com diferentes velocidades, conseqüentemente cada planeta tem de ser animado separadamente.

Para criarmos a órbita de um planeta, bastou desenhar uma circunferência à volta do sol onde queremos que o planeta orbite, selecionamos o planeta e a linha onde vai orbitar (é importante selecionar por essa ordem) e adicionamos um *Motion Paths* (Imagem 129). O *Motion Paths* é semelhante ao *paths constraint* do 3Ds Max. Depois de adicionarmos cada planeta à sua órbita, definimos velocidades e posicionamentos diferentes.

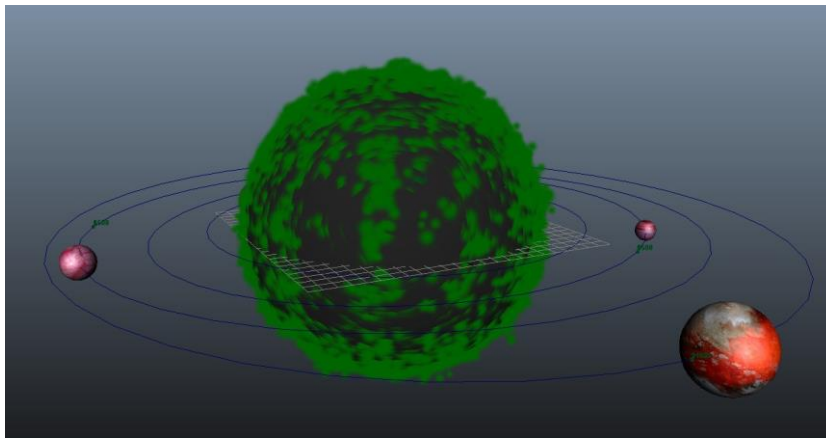


Imagem 129 - Animação no Maya da órbita dos planetas.

---

### 3.2.3. RENDER

Algumas perguntas surgidas no início do projeto foram “Qual é o melhor computador para modelação 3D ou para fazer renders?” e “O que deve ser feito para que o render não demore tanto tempo?”. Chegámos à conclusão que não há uma resposta correta, visto que a indústria dos computadores está sempre a evoluir. Por exemplo, se queremos um computador para realizar renders usando o processador, o melhor será um computador com um bom processador e uma boa quantidade de memória RAM.

A RAM é importante para um bom desempenho do computador, caso a cena 3D tenha muitos polígonos ou muitos mapas de texturas. O sistema operativo de 64 bits também influencia na utilização da memória RAM, já que são capazes de reconhecer memórias acima dos 4GB. O processador é o componente mais importante. Quanto melhor forem os núcleos, melhor será a velocidade do clock. Por exemplo, um processador quad core, onde cada núcleo possui 2 threads, implica que o render de uma cena seja distribuído por 8 threads [141]. No caso de utilizar o render que vem por defeito no 3Ds Max (Backburner), o melhor CPU a utilizar é o da Intel, visto que o Backburner consegue usar todos os núcleos para processar o render. Enquanto que nos processadores da AMD não consegue utilizar todos

os núcleos. Para poder combater esta limitação podemos utilizar outro tipo de render, tal como o Redshift [142]. Isto deve ser pensado desde o início do projeto, já que implica algumas mudanças, tanto nas luzes como noutros fatores importantes.

Para conseguir fazer todos os renders foram necessárias 6089 horas e 12 minutos, que foram divididos pelos PCs acima mencionados. Em anexo, na Tabela 9, encontra-se a divisão dos frames de cada cena, pelos computadores.

Antes de começarmos a fazer o render do projeto, tivemos de dividir as cenas em vários ficheiros para podermos distribuir por todos os computadores existentes de maneira a diminuir o tempo de render que levaria a realizar o render em apenas um computador. Como estamos a trabalhar com um vídeo e não um jogo, que tem de ser renderizado em tempo real, temos a vantagem de podermos dividir as cenas pelos vários computadores e juntar tudo no fim, sem que ninguém se aperceba que o filme foi realizado em vários computadores. Um problema que encontramos ao juntar todos os renders foi uma pequena discrepância de cor, na junção de frames de computadores diferentes numa determinada cena (Imagem 130). Esta discrepância não se fez sentir em todas as junções de frames, apenas num cenário em particular. Não soubemos explicar a razão desta diferença de cor, visto que o ficheiro renderizado foi o mesmo e o método de renderização foi com recurso ao processador e não através da placa gráfica, o que não implica nenhuma mudança nas cores.



Imagem 130 – Discrepância de cor na junção de frames de computadores diferentes.

Escolhemos renderizar as imagens com o tamanho 1280x720, no formato TIFF de 8 bits. A grande qualidade das imagens foi o principal motivo para esta escolha de formato. Em relação ao número de bits, tínhamos duas opções de escolha, 8 e 16 bits, mas devido ao fato do Adobe Premiere não aceitar imagens TIFF de 16 bits, tivemos de optar pelo outro formato.

Uma das configurações realizadas no render foi desligar todas as opções do RayTracer. Deixando esta opção ligada, iria fazer com que um simples frame com poucas árvores e sombras demorasse horas a ser renderizado.

### 3.3. PÓS-PRODUÇÃO

Uma vez acabada a fase de renderização passou-se para a fase de pós-produção. Esta fase foi dividida em 6 etapas: importação de renders, montagem do vídeo, adição de efeitos visuais de transição, efeitos visuais adicionais, adição do som e por fim do diálogo (Imagem 131).

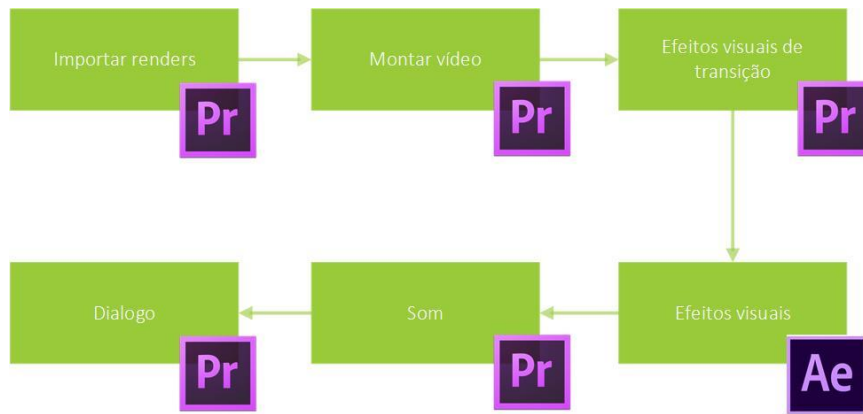


Imagem 131 - Pipeline de pós-produção

Na primeira etapa, foi usado o Adobe Premiere que é um *software* de edição de vídeo e foi utilizado nas primeiras três etapas desta fase [58]. O Premiere permite importar uma sequência de imagens como um único clip de vídeo de modo a poder mais tarde editar cada clip. Para poder usar esta funcionalidade basta ir ao menu *File – Import* e depois de escolher a primeira imagem da sequência escolher a opção *Import Sequence* (Imagem 132). Esta funcionalidade serviu para importar todas as imagens que tinham sido renderizadas na fase de produção.

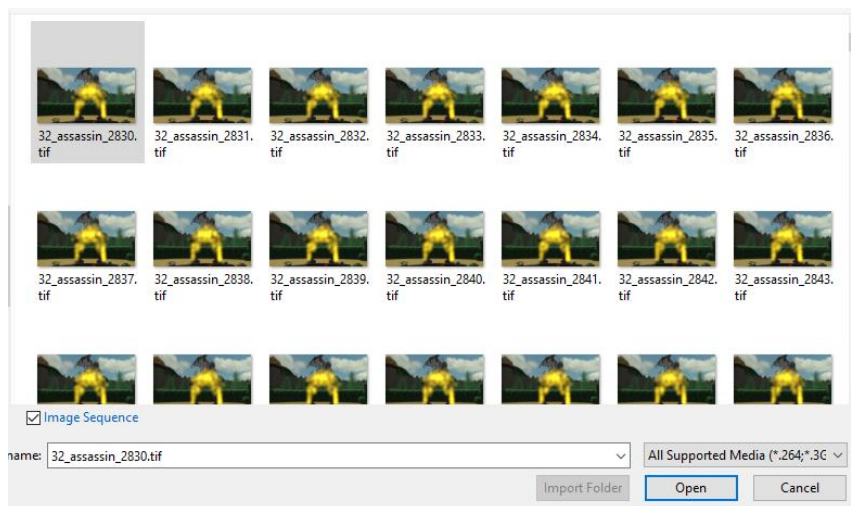


Imagem 132 - Como importar sequências de imagens no Premiere

Assim que todas as sequências de imagens já se encontravam no projeto do Premiere, foi então montado um vídeo completo baseado na linha temporal do *storyboard*. Este vídeo tinha no total 10 minutos de renders. É importante ter tempo de vídeo a mais do que o que se vai usar, pois torna toda a etapa de montagem mais flexível. Isto permite fazer os cortes e as transições necessárias sem nos termos de nos preocupar que estamos a cortar alguma parte importante. Na etapa de montagem do vídeo foi ainda necessário adicionar alguns efeitos de sobreposição (combinação) de vários renders, pois o 3DsMax não permite este tipo de efeitos. Para tal, usou-se o efeito *Chroma Key* que permite retirar uma cor de todo o vídeo, semelhante ao sistema de tela verde usada na televisão. O *Chroma Key* juntamente com efeito de invisibilidade (*Opacity*) permitiram que se conseguisse criar um efeito de aparecimento de cada personagem a sair do portal, tal como podemos ver na Imagem 133. Devido a limitações de *hardware* ao longo da criação do projeto, foi sempre difícil ter a certeza que as velocidades das animações eram as corretas. Como isto acabou por ser um problema em algumas das

cenas renderizadas, foi necessário aumentar a velocidade desses clips. Para isso, o Premiere tem uma opção que permite aumentar a velocidade por uma porcentagem, o que permitiu sincronizar os vários clips.



Imagem 133 – Montagem de imagens no Premiere.

Depois da montagem do filme, passámos à sua edição. Para isso, usámos efeitos tais como *Blur*, que permite desfocar a imagem e *Dip to white* e *Dip to black* para poder fazer transições de desvanecimento para branco e para preto, respetivamente. Foi nesta etapa que foram cortados os vários clips de modo a criar a sequência temporal que tinha sido previamente definida no *storyboard*. No fim desta etapa, a duração do filme passou de 10 para 6 minutos (Imagem 134).

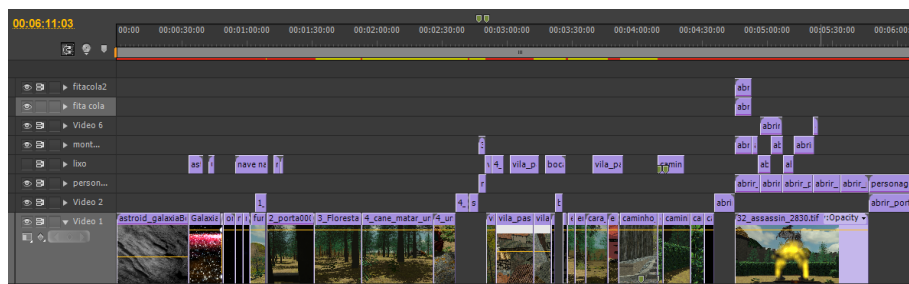


Imagem 134 - Linha temporal de vídeo final.

Depois de terminada toda a edição e montagem do filme, foi necessário adicionar efeitos visuais que eram muito mais simples e menos demorosos de reproduzir se fossem criados na fase de pós-produção e não na fase de produção. Para adicionar estes efeitos foi usado o Adobe After Effects que é um *software* de edição de efeitos visuais nos vídeos [58]. Alguns dos efeitos realizados foram: o efeito de energia à volta do robô, efeito de fumo para melhorar o que foi realizado no 3ds Max e um rasto luminoso que imitasse o disparo da arma.

A última fase do projeto é a importação da faixa de som e do diálogo para o Adobe Premiere de maneira a o sincronizar com o vídeo.

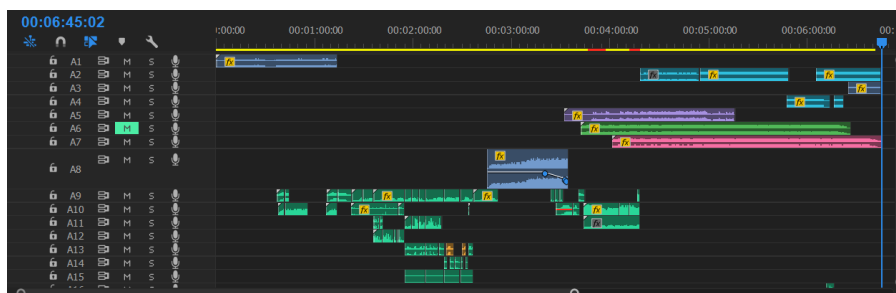


Imagem 135 - Linha temporal de músicas, sons e diálogo.

## 4. IMPLEMENTAÇÃO

Nesta secção mostraremos o processo de criação de alguns objetos 3D, desde o design da ideia até à versão final. Vamos comparar algumas fases do *storyboard* com os renders das cenas finais mostrando as diferenças encontradas e apresentaremos todas as personagens produzidas para o projeto.

O *storyboard* está dividido em 25 cenas, calculando o tempo que levou os oitos computadores a realizarem o render dos 20584 *frames* da historia, chegamos a 6089 horas e 12 minutos. O que corresponde a oito meses, cinco dias, 17 horas e 12 minutos. Estes frames ocupam um total de 81GB de espaço. Tendo em conta isto, se só fosse utilizado um computador para fazer os renders e se este fosse o mais fraco, em relação a processador (exemplo PC4) demoraríamos pelo menos o dobro do tempo de renderização, o que corresponde a 12176 horas e 19 minutos (aproximadamente um ano e cinco meses). Como produto final, chegamos à seguinte produção audiovisual: <https://youtu.be/-syVR4Nt9sQ>

Como sabíamos que o render era um processo que demora muito tempo, então optamos por realizarmos as cenas os mais simples possível, optando por imagens de 1280x720 em vez de 1920x1080 o que nos poupou imenso tempo de render. Isto também nos limitou na hora da escolha das luzes, não podendo utilizar o V-Ray para processar todos o render porque assim demoraríamos ainda mais tempo na fase de renderização. A falta de vegetação nas cenas deve-se também ao tempo de render que esta adiciona a cada frame. Basicamente, o tempo que íamos despendendo na fase de render era tanta, que nos foi limitando conforme fomos avançando no projeto.

A Tabela 5, mostra as características e funções de cada computador utilizado para a realização deste projeto.

**Tabela 5 - Características e funções de cada computador utilizado em este projeto.**

Computadores	Função	Processador	Gráfica	RAM
PC 1	Render	Intel Xeon E3-1230 V2 @ 3.30GHz	NVIDIA quadro 2000	8GB
PC 2	Render	Intel Core i5-4570S @ 2.90GHz	Intel hd graphics 4600	16GB
PC3	Render	Intel Core 2 Quad - Q6600 @ 2.40GHz	NVIDIA GeForce GTX 950	8GB
PC4	Render	Intel Core i7 Q720 @ 1.60GHz	AMD Mobility Radeon HD 5000	4GB
PC5	Render; Modelação 3D	Intel Core i7 - 4720HQ @ 2.60GHz	NVIDIA GeForce GTX 950	16GB
PC6	Render	AMD FX-8320E	4096MB ATI AMD Radeon R9 270X	32GB
PC7	Render	Intel Core i7 6700HQ @ 2.60GHz	2047MB NVIDIA GeForce GTX 960M	16GB
PC8	Render	Intel Core i5 6600K @ 3.50GHz	8192MB AMD Radeon rx480	16GB DDR4

Partindo de um conceito idealizado no design, em primeiro lugar os portais foram desenhados em formato papel. Durante a fase de produção, implementamos as ideias anteriores em 3D. A partir de uma base 3D, os portais foram importados no Zbrush e os seus detalhes foram esculpidos, tendo em conta as características de cada um. Após a inclusão dos detalhes, foi necessário texturizar o respetivo portal como pretendido. Por fim, foram adicionados efeitos visuais usando os sistemas de partículas existentes no 3Ds Max (Imagem 136).

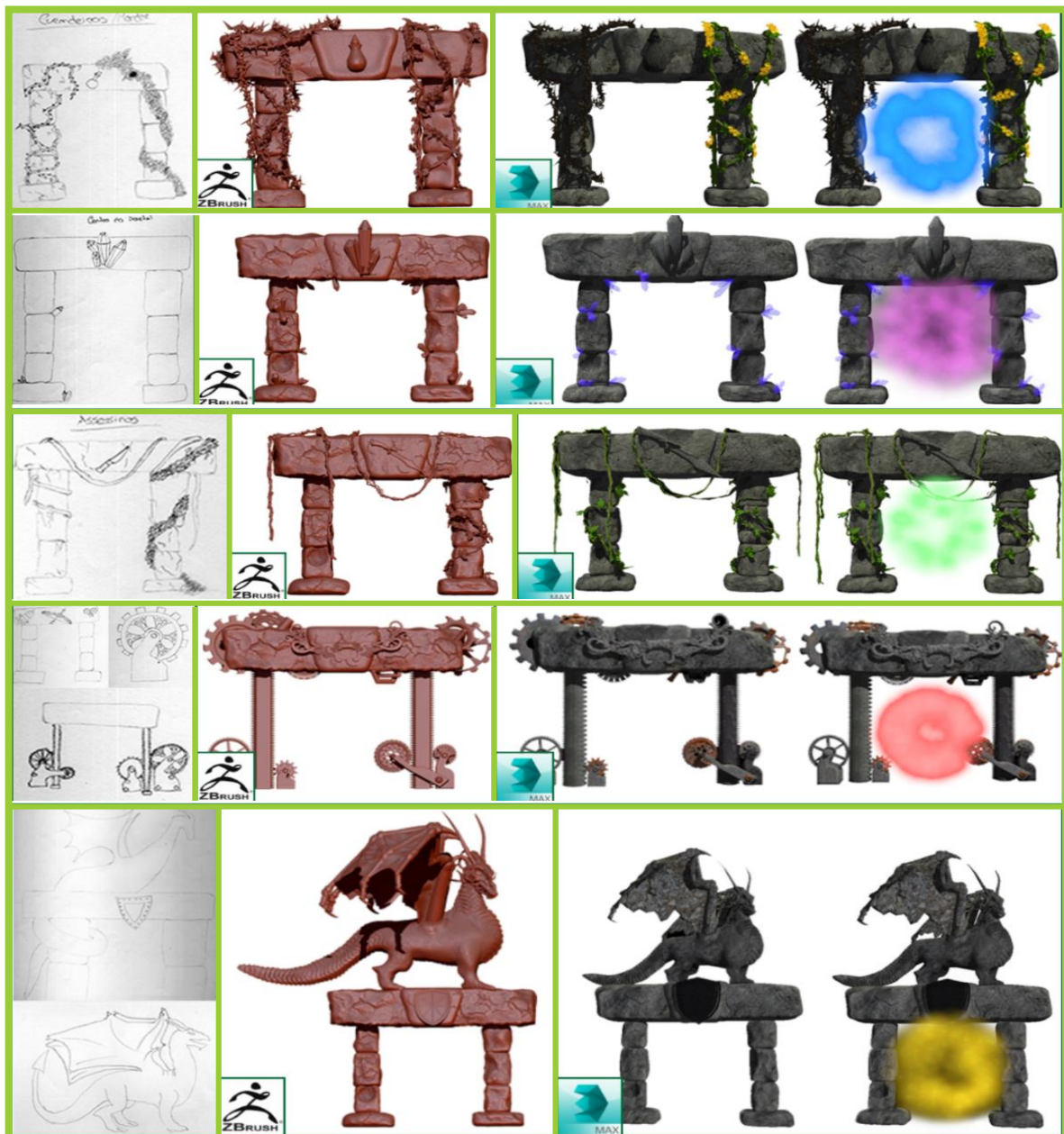


Imagem 136 – Evolução do processo de criação dos portais.

À semelhança do processo realizado com os portais, os objetos das cenas também foram criados seguindo a mesma lógica. Após passar para papel as ideias obtidas, foram criados os modelos base, os detalhes foram adicionados no Zbrush e, posteriormente, texturizados e adicionados efeitos visuais no 3ds Max (Imagem 137).



Imagem 137 – Evolução do processo de criação de alguns objetos das cenas (Amuleto, centro do templo).

Todas as personagens criadas para este projeto foram geradas com o *software* Fuse. Não foi realizado nenhum tipo de design para as personagens, a única característica que as personagens tinham, era que a sua roupa tinha de ser do tempo medieval. Apenas uma personagem tem dois tipos de roupas, o Nifast. Isto deve-se ao facto de que quando ele chega ao planeta Theris, vem de um planeta superior o que se reflete na sua roupa. Com o passar do tempo, ao se adaptar ao mundo, a personagem muda o seu estilo de vestir também para medieval.



Imagem 138 -Todas as personagens realizadas no Fuse.

Iremos fazer a comparação do *storyboard* com os renders finais de modo a explicar as diferenças e estatísticas interessantes em relação ao projeto. Nenhuma cena apresentada teve diferenças significativas, apenas foram adicionados detalhes ou ajustados os objetos e/ou as câmeras.

A cena em que as galáxias se juntam (cena 2) foi realizada no *software* Maya (Imagem 139). As diferenças que podemos encontrar neste *frame* é na nave. No meio de tanta partícula de criação das galáxias e estrelas, ao afastar a câmera da nave acabamos por perde-la de vista. Esta cena é composta por 709 *frames* o que corresponde a um tempo de vídeo de 30 segundos. No total temos 10.2GB de imagens que demorou cinco horas e três minutos de render.



Imagem 139 - Comparação desenho vs render das galáxias boa e má.

A próxima cena acontece quando a nave está a se aproximar do planeta Theris (Imagem 140). Aqui podemos ver que foi desenvolvido exatamente o que estávamos a pensar quando criamos o *storyboard*. Esta cena é composta por 520 *frames* o que corresponde a 22 segundos de vídeo final. Devido à ausência de luzes, de existirem apenas dois sistemas de partículas e as animações serem simples a sua renderização demorou uma hora e ocupou 1.63 GB.

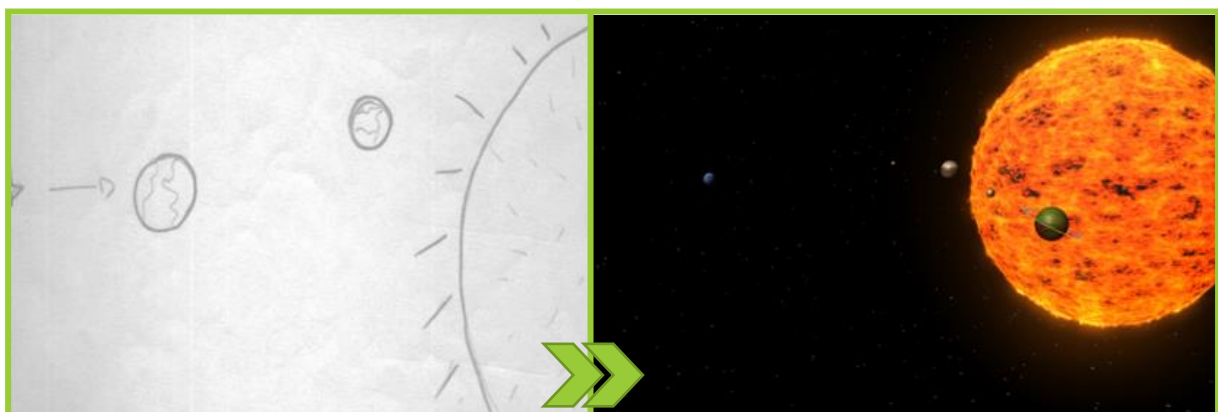


Imagem 140 - Comparação desenho vs render da orbita planetária dos bons.

A cena 6 é representada pela floresta onde Nifast despenhou-se, formando uma nuvem de fumo. Foi montada e animada como foi pensado no *storyboard*. Esta cena representa sete segundos no vídeo final, o que corresponde a 161 *frames* e 20 horas e 31 minutos de tempo de render.

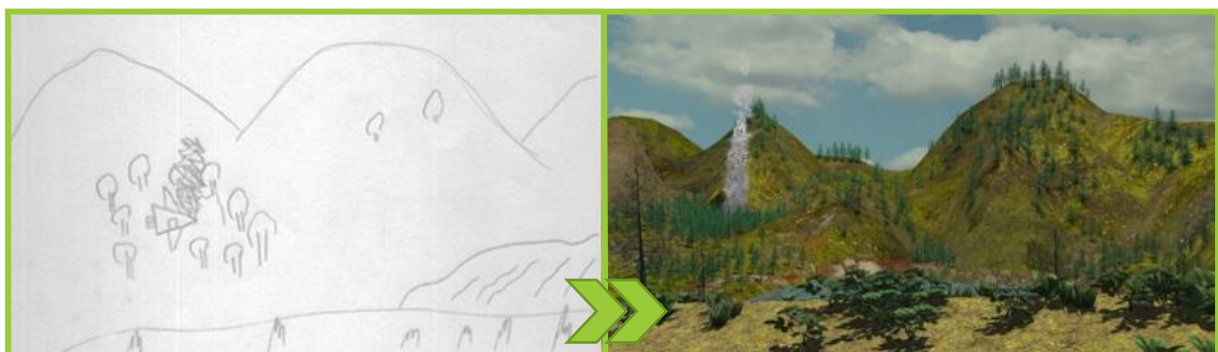


Imagem 141 - Comparação desenho vs render da floresta onde bateu a nave.

A única cena que incorpora um animal é a cena 10. Nesta cena, Isene é salva quando está a ser atacada por um urso feroz e Nifast é o protagonista da cena. A diferença principal nesta cena é o posicionamento da camara. No 3D foi colocar camara numa perspetiva de primeira pessoa vendo a ação através dos olhos de Nifast. Para realizar esta cena foram necessários 1401 *frames* e seis dias, 11 horas e 25 minutos de render, que geraram 58 segundos de vídeo.



Imagem 142 - Comparação desenho vs render em que Nifast mata o urso.

A vila é talvez uma das cenas onde se note mais a diferença entre o *storyboard* e o render final. A ideia no *storyboard* era uma vila medieval um pouco mais agrícola, com terrenos entre os prédios e pessoas a andar pela vila e a cultivar os terrenos (Imagem 143). No fim, decidimos dividir as casas e os terrenos, apenas as casas localizam-se no centro da vila e todos os terrenos de cultivo à volta desta. Como já não podíamos colocar as personagens a cultivar os terrenos, colocamo-las então a passear nas ruas da vila. Esta cena é composta por 1861 *frames* e corresponde a um minuto e 30 segundos de vídeo, demorando 14 dias, nove horas e 41 minutos de render ocupando 4.75GB.



Imagem 143 - Comparação desenho vs render da vila.

A descoberta do robô e o encontro de Nifast com a anciã é o que se desenrola na cena 17 (Imagem 144). As diferenças entre o *storyboard* e a cena final, é que na cena final o Nifast se aproxima até ao robô tentando descobrir de que se trata, enquanto que no *storyboard* a anciã já o estava esperando. Outra diferença, é o movimento da camara que no *storyboard* se encontrava estática e na cena final desloca-se de maneira a poder mostrar mais da cena. Esta cena é composta por 361 *frames*, que foram renderizados em dois dias, cinco horas e 29 minutos, que correspondem a 15 segundos de vídeo.



Imagem 144 - Comparação desenho vs render final da cena em que é encontrado o robô.

A chegada ao templo e o início da abertura dos portais é a cena na qual podemos notar uma maior entre o *storyboard* e o render (Imagem 145) final. Inicialmente foi pensado em cortar diretamente para os portais que se localizavam na floresta, onde a cena anterior acontecia. Para criar um pouco de mistério, foi adicionado a esta cena um jardim em forma de labirinto que é percorrido para chegar aos portais. Esta é a cena com o maior número de *frames*, tendo 3526. Consiste em dois minutos e 45 segundos de vídeo e levou 6 meses, 16 dias e duas horas e seis minutos, ocupando 12.2GB.



Imagem 145 - Comparação desenho vs render do templo onde se encontra os portais.

A saída das personagens dos portais foi a cena mais complexa visto que a personagens têm de parecer que saem do portal (Imagem 146). Para isso a cena foi dividida em dois ficheiros. Um deles com todo o cenário e sem a personagem e o outro com apenas a personagem e um pouco do chão. O chão foi necessário já que as duas cenas de iriam ser sobrepostas e necessitámos da sombra da personagem no chão. Juntando as duas partes, no total esta cena é composta por 3596 *frames*, o que corresponde a 1:25 segundos de vídeo. Demorou um dia, sete horas e 41 minutos a renderizar que ocupou um total de 11,7GB. As diferenças entre a ideia e o render final não são muito grandes. Apenas o incorporamos num cenário, mais completo e adicionámos o símbolo correspondente ao portal.

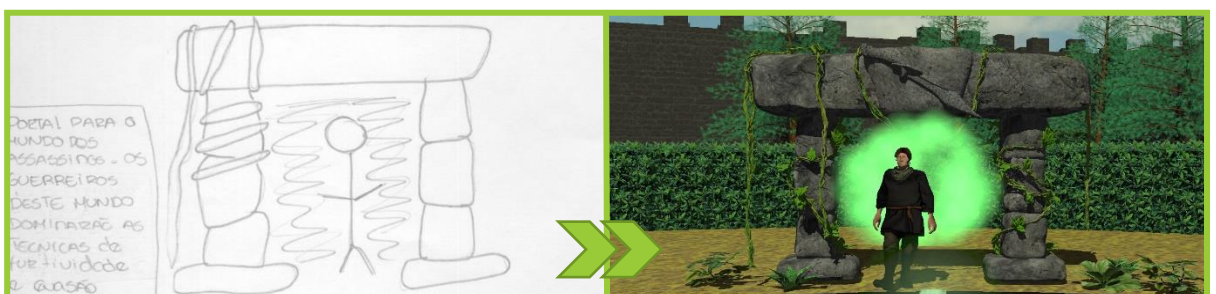


Imagem 146 - Comparação desenho vs render do portal dos assassinos.



## 5. CONCLUSÃO

Durante muitos anos, as produções de filmes 3D estavam limitadas às várias empresas da indústria de entretenimento. Principalmente devido aos custos elevados dos equipamentos necessários, desde os sistemas de captura de movimentos às estações de trabalho, como também aos custos dos *softwares* utilizados. Nestes últimos anos, foram surgindo soluções mais económicas e mais acessíveis ao público geral. No que toca aos sistemas de captura de movimentos, com o lançamento da Microsoft Kinect, tornou-se possível realizar projetos incluindo captura de movimentos por valores bastante competitivos.

Neste projeto demonstramos de que maneira foi possível juntar a Microsoft Kinect, como sistema de captura de movimentos, com um conjunto de *softwares* gratuitos e pagos, de forma a conseguir seguir um processo para a realização de um *trailer* 3D para o jogo *Keepers of Intheris* [1].

Normalmente, este tipo de projetos é realizado na indústria de entretenimento por equipas de várias pessoas, onde cada pessoa é normalmente especialista numa determinada fase do projeto, tomando como exemplo as fases de modelação, rigging, texturização, animação, tratamento de imagem/som, entre outros. Este foi um dos grandes desafios enfrentados, pois foi necessário adquirir vários conhecimentos e um à vontade suficiente para conseguir um resultado satisfatório. Muitas das técnicas utilizadas neste projeto não foram aprendidas durante o percurso universitário na Universidade da Madeira. Estas foram aprendidas através de uma autoaprendizagem, tendo encontrado soluções tanto em tutoriais presentes na Internet e através de tentativa e erro.

Um outro grande desafio, no qual foi necessário despender muito tempo, foi na familiarização dos vários *softwares* que reunimos para a realização deste projeto. Muitos destes *softwares* foram utilizados pela primeira vez no decorrer do projeto. A nível de hardware, embora já possuísse um tablet de desenho, foi necessário também reunir conhecimentos sobre como usá-lo para esculpir os modelos.

Uma dificuldade sentida durante a fase de produção da animação facial e corporal foi conseguir estabelecer a interligação entre o Brekel e o MotionBuilder e, por sua vez, a interligação com a personagem. O problema foi que estas três componentes possuíam vários pontos de interligação diferentes uns dos outros. Não foi possível encontrar nenhum tutorial que explicasse como realizar esta ação. Para contornar este problema, foi necessário extrair informação de vários tutoriais e, através de tentativa e erro, foi possível conseguir interligar estas três componentes.

Como trabalho futuro, gostaríamos de continuar a modelação do mundo de cada um dos guerreiros. Visto que já existe trabalho nesta vertente, nomeadamente todas as ideias desenhadas e imagens que servem de inspiração, há opção para dar continuidade à história, mostrando como cada guerreiro vive e treina no seu mundo.

O resultado deste projeto não foi um resultado perfeito, uma vez que existe margem para melhorar. No entanto, fomos capazes de demonstrar que é possível realizar uma produção de um vídeo 3D, com recurso a um sistema de captura de movimentos, com equipamentos e *softwares* bastante acessíveis à maior parte das pessoas, o que nos levou a alcançar o nosso objetivo.

Alguns aspetos que particularmente achamos que obtiveram um bom resultado foram os sistemas de partículas realizados com o Maya, mais propriamente as galáxias, o sol e o rasto da nave. Estes sistemas

de partículas deram um bom grau de realismo ao início do vídeo. As capturas de movimento usando a Microsoft Kinect, tanto facial como corporal, tiveram bons resultados sendo que, com isto, conseguimos minimizar o tempo passado a animar obtendo também um bom grau de realismo às cenas. Por fim, temos a destacar os efeitos visuais e os sons/músicas utilizados que deram uma outra imersão a quem visualiza o vídeo.

Em contrapartida, existem alguns aspetos que poderiam ter sido melhorados, sendo um deles os sistemas de partículas utilizados nos portais e no templo. Estes sistemas de partículas, em particular, poderiam ter sido criados de uma outra forma, de maneira a que o resultado final parecesse, dentro da temática, mais realista e natural. Um outro aspeto que poderia vir a ser melhorado era a texturização. Um exemplo seria a texturização dos portais usando as técnicas de pintura que o Substance Painter para texturizar todos os modelos existentes, pois esta ferramenta permite adicionar detalhes em zonas específicas e também texturizar no próprio modelo 3D. Por fim, a gravação da voz foi realizada após a captura de movimento facial, o que levou, numa etapa final do projeto, a uma fase de sincronismo entre a animação e a gravação. Uma vez que a própria Kinect dispõe de um microfone integrado, esta fase poderia ter sido simplificada ao capturar a voz juntamente com a captura de movimento facial.

Espero que este projeto possa servir de ajuda a quem pretenda realizar um vídeo ou filme 3D, com ou sem equipamento de captura de movimentos. A *pipeline* utilizada neste projeto, na minha opinião, e embora não seguindo à risca, é uma boa adaptação às *pipelines* utilizadas nas indústrias de entretenimento, sendo que se alguém seguir o processo descrito conseguirá ter mais alguma orientação e uma base para uma produção 3D.

## 6. REFERENCIAS

- [1] Keepers of Intheris, "Keepers of Intheris." [Online]. Available: <http://www.keepersofintheris.com/>. [Accessed: 07-Sep-2017].
- [2] Microsoft, "Hardware Kinect," 2017. [Online]. Available: <https://developer.microsoft.com/pt-pt/windows/kinect/hardware>. [Accessed: 20-Jul-2017].
- [3] Pluralsight, "From the 80s to Now: The Evolution of Animation in Video Games | Pluralsight," 2014. [Online]. Available: <https://www.pluralsight.com/blog/film-games/80s-now-evolution-animation-video-games>. [Accessed: 23-Aug-2017].
- [4] "The Evolution of Gaming and Animation | Centre for Arts and Technology," 2013. [Online]. Available: <https://digitalartschool.com/blogs/animation/2013/06/evolution-gaming-and-animation>. [Accessed: 23-Aug-2017].
- [5] brennanmurraysae, "History of 3D Animation In Video Games – brennanmurraysae," 2016. [Online]. Available: <https://brennanmurraysae.wordpress.com/2016/08/26/history-of-3d-animation/>. [Accessed: 23-Aug-2017].
- [6] FANDOM, "Toy Story | Disney Wiki | FANDOM powered by Wikia." [Online]. Available: [http://disney.wikia.com/wiki/Toy\\_Story](http://disney.wikia.com/wiki/Toy_Story). [Accessed: 26-Aug-2017].
- [7] P. De Semlyen, "A History Of CGI In The Movies , Feature | Movies - Empire," 2015. [Online]. Available: <http://www.empireonline.com/movies/features/history-cgi/>. [Accessed: 26-Aug-2017].
- [8] A. Beane, *3D Animation Essential*. 2012.
- [9] A. Gupta, "3D animation film making," 2014. [Online]. Available: <https://www.slideshare.net/aakashgupta370/3d-animation-film-making>. [Accessed: 21-Aug-2017].
- [10] F. Pellacini, "the 3D production pipeline," 2009. [Online]. Available: [http://pellacini.di.uniroma1.it/teaching/projects10/lectures/01\\_pipeline.pdf](http://pellacini.di.uniroma1.it/teaching/projects10/lectures/01_pipeline.pdf). [Accessed: 21-Aug-2017].
- [11] M. Ray Brown, "Screenplay Format Guide," 2006. [Online]. Available: <http://www.storysense.com/SPFormat.pdf>. [Accessed: 21-Aug-2017].
- [12] "3D Production Pipeline | David Hailes 3D Research." [Online]. Available: <https://davidhailes.wordpress.com/3d-production-pipeline/>. [Accessed: 21-Aug-2017].
- [13] C. E. Ferreira, "8 coisas que você talvez não saiba sobre Toy Story - TecMundo," 2015. [Online]. Available: <https://www.tecmundo.com.br/cinema/81649-8-coisas-voce-nao-saiba-toy-story.htm>. [Accessed: 26-Aug-2017].
- [14] cgriver, "Woody and Buzz Rigged - 3D Model - CGriver." [Online]. Available: <http://www.cgriver.com/fabelar/10058-woody-and-buzz-rigged.html>. [Accessed: 26-Aug-2017].
- [15] M. Seymour, "RenderMan cai no preço | Fxguide," 2010. [Online]. Available: [https://www.fxguide.com/featured/renderman\\_drops\\_in\\_price/](https://www.fxguide.com/featured/renderman_drops_in_price/). [Accessed: 26-Aug-2017].
- [16] pixar, "Rigging | The Science Behind Pixar." [Online]. Available:

- <http://sciencebehindpixar.org/pipeline/rigging>. [Accessed: 21-Aug-2017].
- [17] “Christopher McCabe.” [Online]. Available: [http://christopher-mccabe.com/projects/toy\\_story.html](http://christopher-mccabe.com/projects/toy_story.html). [Accessed: 26-Aug-2017].
- [18] pixar, “Lighting | The Science Behind Pixar.” [Online]. Available: <http://sciencebehindpixar.org/pipeline/lighting>. [Accessed: 21-Aug-2017].
- [19] Autodesk, “3ds Max 3D Modeling, Animation & Rendering Software.” [Online]. Available: <http://www.autodesk.com/products/3ds-max/overview>.
- [20] “Maya | Animação Computer & Modeling Software | Autodesk.” [Online]. Available: <http://www.autodesk.com/products/maya/overview>. [Accessed: 16-Dec-2016].
- [21] Pixologic, “ZBrush - The all-in-one-digital sculpting solution.” [Online]. Available: <http://pixologic.com/>. [Accessed: 25-Jul-2017].
- [22] Autodesk, “Autodesk | Software de projeção, engenharia e entretenimento 3D.” [Online]. Available: <https://www.autodesk.pt/>. [Accessed: 22-Aug-2017].
- [23] Pluralsight, “3ds Max vs. Maya: Is One Better than the Other? | Pluralsight,” 2015. [Online]. Available: <https://www.pluralsight.com/blog/film-games/3ds-max-vs-maya-is-one-better-than-the-other>. [Accessed: 24-Aug-2017].
- [24] Autodesk, “Comparison of 3ds Max and Maya | 3ds Max | Autodesk Knowledge Network,” 2016. [Online]. Available: <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/sfdcarticles/sfdcarticles/Comparison-of-3ds-Max-and-Maya.html#mtc-english>. [Accessed: 24-Aug-2017].
- [25] Udemy, “3DS Max vs Maya: A Friendly Comparison,” 2014. [Online]. Available: <https://blog.udemy.com/3ds-max-vs-maya/>. [Accessed: 24-Aug-2017].
- [26] Autodesk, “Maya | Computer Animation & Modeling Software.” [Online]. Available: <https://www.autodesk.com/products/maya/overview>. [Accessed: 16-Dec-2016].
- [27] cgwhat, “Maya vs 3DS Max.” [Online]. Available: <http://cgwhat.com/maya-vs-3ds-max/>. [Accessed: 24-Aug-2017].
- [28] Pluralsight, “New features and enhancements in ZBrush 4R7 | Pluralsight,” 2015. [Online]. Available: <https://www.pluralsight.com/blog/film-games/new-features-enhancements-zbrush-4r7>. [Accessed: 24-Aug-2017].
- [29] Pluralsight, “ZBrush or Mudbox: Sculpting Showdown | Pluralsight,” 2014. [Online]. Available: <https://www.pluralsight.com/blog/film-games/zbrush-mudbox-sculpting-showdown>. [Accessed: 24-Aug-2017].
- [30] Huion, “Huion H610 Graphic Tablet.” [Online]. Available: <https://www.huiontablet.com/h610.html>. [Accessed: 26-Jul-2017].
- [31] Pixologic, “Pixologic > ZBrush 4R8 > Single User License.” [Online]. Available: <https://store.pixologic.com/zbrush-4r8/single-user-license/>. [Accessed: 24-Aug-2017].
- [32] Wacom, “Escultura em 3D e modelagem | Wacom,” 2016. [Online]. Available: <http://www.wacom.com/en-es/discover/draw/3d-sculpting-and-modeling>. [Accessed: 19-Jul-2017].

- [33] Autodesk, “3D Character Animation Software | MotionBuilder.” [Online]. Available: <https://www.autodesk.com/products/motionbuilder/overview>. [Accessed: 16-Dec-2016].
- [34] Brekel, “Brekel Pro Face 2,” 2008. [Online]. Available: <http://brekel.com/brekel-pro-face-2/>. [Accessed: 21-Jul-2017].
- [35] Brekel, “Brekel Pro Body v2,” 2008. [Online]. Available: <http://brekel.com/brekel-pro-body-v2/>. [Accessed: 21-Jul-2017].
- [36] IpiSoft, “iPi Soft – Markerless Motion Capture.” [Online]. Available: <http://ipisoft.com/>. [Accessed: 16-Dec-2016].
- [37] Reallusion, “Real-time 3D Animation Software - iClone.” [Online]. Available: <https://www.reallusion.com/iclone/>. [Accessed: 16-Dec-2016].
- [38] “Software - Facial Motion Capture.” [Online]. Available: <http://kinectmocap.weebly.com/software.html>. [Accessed: 25-Aug-2017].
- [39] CyBoRgTy, “Mocap into Carrara – by CyBoRgTy | Carrara Cafe,” 2013. [Online]. Available: <http://carraracafe.com/articles/mocap-in-carrara-by-cyborgty/3/>. [Accessed: 25-Aug-2017].
- [40] Reallusion, “3DXchange - Unlimited 3d Content, 3d Character and Motion Conversion - Reallusion.” [Online]. Available: <https://www.reallusion.com/iclone/3dx/>. [Accessed: 25-Aug-2017].
- [41] Wikipédia, “Animação de destino Morph - Wikipedia,” 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Morph\\_target\\_animation](https://en.wikipedia.org/wiki/Morph_target_animation). [Accessed: 28-Aug-2017].
- [42] MicroFilmmaker Magazine, “iClone 6 Pro + Character Creator (Software Review) | MicroFilmmaker Magazine,” 2015. [Online]. Available: [http://www.microfilmmaker.com/2015/10/iclone6\\_rev/](http://www.microfilmmaker.com/2015/10/iclone6_rev/). [Accessed: 25-Aug-2017].
- [43] “Brekel - Tools for Markerless Motion Capture,” 2016. [Online]. Available: <http://brekel.com/>.
- [44] Adobe, “Aplicativos de edição de fotos para Mac, PC e dispositivos móveis | Família Adobe Photoshop.” [Online]. Available: <https://www.adobe.com/pt/products/photoshopfamily.html>. [Accessed: 26-Aug-2017].
- [45] Allegorithmic, “Substance Painter - 3D Painting Software.” [Online]. Available: <https://www.allegorithmic.com/products/substance-painter>. [Accessed: 25-Jul-2017].
- [46] T. Williamson, “Feature preview: Photoshop and the 3D pipeline | Advanced Photoshop - Free Photoshop Tutorials & Online Resources | Advanced PhotoShop Magazine.” [Online]. Available: <https://blog.advancedphotoshop.co.uk/magazine-previews/feature-preview-photoshop-and-the-3d-pipeline/>. [Accessed: 26-Aug-2017].
- [47] allegorithmic, “Games | allegorithmic.” [Online]. Available: <https://www.allegorithmic.com/games>. [Accessed: 26-Aug-2017].
- [48] allegorithmic, “Buy | allegorithmic.” [Online]. Available: <https://www.allegorithmic.com/buy/download>. [Accessed: 26-Aug-2017].
- [49] Adobe, “Preços e planos de associação à Creative Cloud | Adobe Creative Cloud.” [Online]. Available: [https://www.adobe.com/pt/creativecloud/plans.html?single\\_app=photoshop&promoid=KSPGW](https://www.adobe.com/pt/creativecloud/plans.html?single_app=photoshop&promoid=KSPGW). [Accessed: 26-Aug-2017].

- [50] Adobe, "Comprar o Adobe Photoshop CC," 2017. [Online]. Available: [https://www.adobe.com/pt/products/photoshop.html?mv=search&s\\_kwcid=AL!3085!3!188069151803!e!!!!photoshop cs6&ef\\_id=WQodkQAAAZLh7Z60:20170721110038:s](https://www.adobe.com/pt/products/photoshop.html?mv=search&s_kwcid=AL!3085!3!188069151803!e!!!!photoshop%20cs6&ef_id=WQodkQAAAZLh7Z60:20170721110038:s). [Accessed: 21-Jul-2017].
- [51] NVIDIA, "Ferramentas de textura NVIDIA para Adobe Photoshop," 2017. [Online]. Available: <https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>. [Accessed: 21-Jul-2017].
- [52] N. Friess, "ArtStation - MAT Substance Painter Contest, Nick Friess," 2017. [Online]. Available: <https://www.artstation.com/artwork/96rBv>. [Accessed: 26-Aug-2017].
- [53] V. Gault, "Character Look Dev with Substance Painter: The Workflow of Cedric Seaut | allegorithmic," 2016. [Online]. Available: <https://www.allegorithmic.com/blog/character-look-dev-substance-painter-workflow-cedric-seaut>. [Accessed: 26-Aug-2017].
- [54] Pluralsight, "Substance Painter: An Intelligent Approach to Painting Textures | Pluralsight," 2014. [Online]. Available: <https://www.pluralsight.com/blog/film-games/substance-painter-intelligent-approach-painting-textures>. [Accessed: 26-Aug-2017].
- [55] "The rise of substance."
- [56] Adobe, "3D Character Creator, Fuse (Preview) - Mixamo." [Online]. Available: <https://www.mixamo.com/fuse>. [Accessed: 17-Aug-2017].
- [57] MakeHuman, "MakeHuman | Open source tool for making 3d characters." [Online]. Available: <http://www.makehuman.org/>. [Accessed: 17-Aug-2017].
- [58] Adobe, "Adobe Creative Cloud | Software e serviços para profissionais de criação." [Online]. Available: <https://www.adobe.com/pt/creativecloud.html>. [Accessed: 27-Aug-2017].
- [59] Adobe, "3D Animation Online Services, 3D Characters, and Character Rigging - Mixamo." [Online]. Available: <https://www.mixamo.com/>. [Accessed: 17-Aug-2017].
- [60] S. Caplin, "Learning Adobe Fuse CC," 2016. [Online]. Available: <https://www.lynda.com/Fuse-tutorials/Learn-Adobe-Fuse-CC-Basics/473891-2.html>. [Accessed: 27-Aug-2017].
- [61] Adobe, "Recursos do Fuse CC (Preview) | Experimentar o Adobe Fuse CC (Preview)." [Online]. Available: <https://www.adobe.com/pt/products/fuse/features.html>. [Accessed: 27-Aug-2017].
- [62] S. Rosa, "Human characters with Adobe Fuse - Renderosity Magazine," 2016. [Online]. Available: <http://renderositymagazine.com/human-characters-with-adobe-fuse-cms-485>. [Accessed: 27-Aug-2017].
- [63] Adobe, "Rig Your 3D Characters Automatically, Auto-Rigger - Mixamo." [Online]. Available: <https://www.mixamo.com/auto-rigger>. [Accessed: 17-Aug-2017].
- [64] Creative Bloq Staff, "Is Adobe Fuse a worthwhile addition to your design toolkit? | Creative Bloq," 2015. [Online]. Available: <http://www.creativebloq.com/software/adobe-fuse-worthwhile-addition-121518421>. [Accessed: 27-Aug-2017].
- [65] N. Kirk, "How to Use Adobe Fuse," 2016. [Online]. Available: <https://design.tutsplus.com/articles/how-to-use-adobe-fuse--cms-26108>. [Accessed: 27-Aug-2017].
- [66] M. Kathleen, "Evolving Characters | Computer Graphics World," 2014. [Online]. Available:

- <http://www.cgw.com/Press-Center/In-Focus/2014/Evolving-Characters.aspx>. [Accessed: 27-Aug-2017].
- [67] silotto, "MakeHuman - Software Open Source para criação 3D," 2015. [Online]. Available: <http://www.tekzoom.com.br/makehuman-software-open-source-para-criacao-de-personagens-3d/>. [Accessed: 27-Aug-2017].
- [68] blender, "blender.org - Home of the Blender project - Free and Open 3D Creation Software." [Online]. Available: <https://www.blender.org/>. [Accessed: 27-Aug-2017].
- [69] N. Jeremy, "The Phenakistoscope, the First Device to Demonstrate the Illusion of a Moving Image (1832 – 1834)." [Online]. Available: <http://www.historyofinformation.com/expanded.php?id=4131>. [Accessed: 25-Jul-2017].
- [70] North Carolina School of Science and Mathematics, "Phenakistoscope." [Online]. Available: <http://courses.ncssm.edu/gallery/collections/toys/html/exhibit07.htm>. [Accessed: 27-Jun-2017].
- [71] The Royal Borough of Kingston upon Thames, "Eadweard Muybridge."
- [72] M. Kitagawa and B. Windsor, *MoCap for Artists: Workflow and Techniques for Motion Capture*. 2012.
- [73] P. Prodger and T. Gunning, *Time Stands Still: Muybridge and the Instantaneous Photography Movement*. 2003.
- [74] M. Delbridge, *Motion Capture in Performance: An Introduction*. 2015.
- [75] Fleischer Studios, "Rotoscope." [Online]. Available: <http://www.fleischerstudios.com/rotoSCOPE.html>. [Accessed: 19-Jun-2017].
- [76] hereitbegins, "Making Of Disney's Snowwhite or How Disney Cartoons are made.flv - YouTube." [Online]. Available: <https://www.youtube.com/watch?v=OebUzEhSLBI>. [Accessed: 08-Jul-2017].
- [77] Massachusetts Institute of Technology, "Stroboscopic « Harold &quot;Doc&quot; Edgerton." [Online]. Available: <http://edgerton-digital-collections.org/techniques/stroboscope>. [Accessed: 09-Jul-2017].
- [78] AnimationXpress Team, "The Evolution of Motion Capture Technology: Philip Stilgoe," 2015. [Online]. Available: <http://www.animationxpress.com/index.php/latest-news/the-evolution-of-motion-capture-technology>. [Accessed: 08-Jul-2017].
- [79] Polhemus History, "Over 40 Years of Experience." [Online]. Available: <http://polhemus.com/company/history/>. [Accessed: 08-Jul-2017].
- [80] B. Koses, "A Brief History of Motion Capture," 2014. [Online]. Available: <http://www.geekinsider.com/breif-history-motion-capture/>. [Accessed: 12-Jul-2017].
- [81] Cyberbiology, "Digital Biota 2." [Online]. Available: <http://www.cyberbiology.org/Yaeger.html>. [Accessed: 09-Jul-2017].
- [82] V. Berezina-Blackburn, "Motion Capture History, Technologies and Applications."
- [83] M. Zhang, "Infographic: The History of Motion Capture, From 1774 to Today - MovieMaker Magazine." [Online]. Available:

- <https://www.moviemaker.com/archives/moviemaking/other/history-of-motion-capture-infographic/>. [Accessed: 30-Jul-2017].
- [84] P. Roberts, "What kind of 3d programs were used for creating 1960's - 1980's motion graphics and effects for movies/TV? - Quora," 2015. [Online]. Available: <https://www.quora.com/What-kind-of-3d-programs-were-used-for-creating-1960s-1980s-motion-graphics-and-effects-for-movies-TV>. [Accessed: 09-Jul-2017].
- [85] The Ohio State University, "Animation Software Companies and Individuals." [Online]. Available: <http://excelsior.biosci.ohio-state.edu/~carlson/history/tree/ani-software.html>. [Accessed: 09-Jul-2017].
- [86] Fandom, "Waldo C. Graphic | Muppet Wiki | FANDOM powered by Wikia." [Online]. Available: [http://muppet.wikia.com/wiki/Waldo\\_C.\\_Graphic](http://muppet.wikia.com/wiki/Waldo_C._Graphic). [Accessed: 16-Aug-2017].
- [87] A. Menache, *Understanding motion capture for computer animation and video games*. Morgan Kaufmann, 2000.
- [88] "Mike the Talking Head," 2012.
- [89] DigitalNeohuman, "Real Time Mario at the 1992 SCES - YouTube," 2010. [Online]. Available: <https://www.youtube.com/watch?v=R1t6iNG28zl>. [Accessed: 16-Aug-2017].
- [90] C. Lo, "VIRTUAL MIRROR: A real-time motion capture application for virtual-try-on," 2006.
- [91] S. Corazza, L. Mündermann, A. M. Chaudhari, T. Demattio, C. Cobelli, and T. P. Andriacchi, "A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach," *Ann. Biomed. Eng.*, vol. 34, no. 6, pp. 1019–1029, Jun. 2006.
- [92] Microsoft, "Hardware e Jogos - Microsoft." [Online]. Available: <https://www.microsoft.com/pt-pt/search/result.aspx?q=kinect&search=>. [Accessed: 16-Aug-2017].
- [93] P. Nogueira, "Motion Capture Fundamentals A Critical and Comparative Analysis on Real-World Applications."
- [94] A. Shingade and A. Ghotkar, "Animation of 3D Human Model Using Markerless Motion Capture Applied To Sports," *Int. J. Comput. Graph. Animat.*, vol. 4, no. 1, 2014.
- [95] E. Ceseracciu, Z. Sawacha, and C. Cobelli, "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept," *PLoS One*, vol. 9, no. 3, p. e87640, 2014.
- [96] Los Angeles Times, "E3: a Microsoft mostra a tecnologia de controle gestual para Xbox 360," 2009. [Online]. Available: <http://latimesblogs.latimes.com/technology/2009/06/microsofte3.html>. [Accessed: 18-Aug-2017].
- [97] M. M. Correia, "Reconhecimento de Elementos Gestuais com Kinect," 2013.
- [98] J. Loureiro, "Kinect esgota em Portugal no dia de lançamento," 2010. [Online]. Available: <http://www.eurogamer.pt/articles/2010-11-11-kinect-esgota-em-portugal-no-lancamento>. [Accessed: 20-Jul-2017].
- [99] T. Sugiura, "Kinect v2 Introduction and Tutorial," 2014. [Online]. Available: <https://pt.slideshare.net/SugiuraTsukasa/kinect-v2-introduction-and-tutorial>. [Accessed: 20-

Jul-2017].

- [100] Microsoft Décideurs IT, "Kinect en entreprise," 2013. [Online]. Available: <https://www.slideshare.net/DecideursIT/des207-kinect-en-entreprise?next>. [Accessed: 21-Jul-2017].
- [101] V. Guigui, "Kinect et usages professionnels," 2014. [Online]. Available: <https://www.slideshare.net/gcrao78/jdc2014-kinect-et-usages-professionnels>. [Accessed: 20-Jul-2017].
- [102] Matteo Valoriani, "Programming with kinect v2," 2015. [Online]. Available: <https://www.slideshare.net/MatteoValoriani/programming-with-kinect-v2?next>. [Accessed: 02-Dec-2017].
- [103] Microsoft, "Obter o Kinect," 2017. [Online]. Available: <https://developer.microsoft.com/pt-pt/windows/kinect/hardware-reseller>. [Accessed: 02-Jul-2017].
- [104] Microsoft, "Desenvolver com o Kinect," 2017. [Online]. Available: <https://developer.microsoft.com/pt-pt/windows/kinect/develop>. [Accessed: 20-Jul-2017].
- [105] M. Bonanni, "Kinect V2: NUI for dummies!!," 2014. [Online]. Available: <https://pt.slideshare.net/massimobonanni/kinect-v2-nui-for-dummies>. [Accessed: 20-Jul-2017].
- [106] Microsoft, "Microsoft Impacto no Brasil," 2013. [Online]. Available: <https://www.microsoft.com/about/corporatecitizenship/pt-br/impacto-no-brasil/inovacao-e-ped/casos-de-sucesso-inovacao-e-ped/kinect-vai-para-a-sala-de-cirurgia/>. [Accessed: 02-Jun-2017].
- [107] B. Crouse, "Microsoft Kinect Sensor applications in health and medicine – HealthBlog," 2014. [Online]. Available: <https://blogs.msdn.microsoft.com/healthblog/2014/01/10/microsoft-kinect-sensor-applications-in-health-and-medicine/>. [Accessed: 09-Jul-2017].
- [108] K. Saueerwein, "Xbox gaming technology may improve X-ray precision," 2015. [Online]. Available: <https://source.wustl.edu/2015/12/xbox-gaming-technology-may-improve-x-ray-precision/>. [Accessed: 02-Jun-2017].
- [109] L. Tyler, "FDA Approves Use Of Kinect For Stroke Recovery | Ubergizmo," 2014. [Online]. Available: <http://www.ubergizmo.com/2014/05/fda-approves-use-of-kinect-for-stroke-recovery/>. [Accessed: 01-Jul-2017].
- [110] X-TECH CREATIVE STUDIO, "Kinect Game-Based Rehabilitation." [Online]. Available: <http://x-tech.am/kinect-game-based-rehabilitation/>. [Accessed: 01-Jul-2017].
- [111] E. Kee, "Xbox One Kinect Is Therapy Device For Cerebral Palsy Kids | Ubergizmo," 2014. [Online]. Available: <http://www.ubergizmo.com/2014/05/xbox-one-kinect-is-therapy-device-for-cerebral-palsy-kids/>. [Accessed: 01-Jul-2017].
- [112] Microsoft, "Using Kinect to monitor Parkinson's patients – Kinect for Windows Product Blog," 2015. [Online]. Available: <https://blogs.msdn.microsoft.com/kinectforwindows/2015/03/13/using-kinect-to-monitor-parkinsons-patients/>. [Accessed: 01-Jul-2017].
- [113] Microsoft, "Kinect Sign Language Translator - part 1 - Microsoft Research," 2013. [Online]. Available: <https://www.microsoft.com/en-us/research/blog/kinect-sign-language-translator->

- part-1/. [Accessed: 29-Jun-2017].
- [114] Microsoft, "Teachers Are Using Kinect for Xbox 360 to Engage Students and Bring Learning to Life - News Center," 2012. [Online]. Available: <https://news.microsoft.com/2012/03/06/teachers-are-using-kinect-for-xbox-360-to-engage-students-and-bring-learning-to-life/#5zDYKUTShobJDozV.97>. [Accessed: 29-Jun-2017].
- [115] L. Students, "Change the Game with Lean Learning Change the Game with Lean Learning."
- [116] Microsoft, "Welcome to Robotics Developer Studio." [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb648760.aspx?f=255&MSPPErr=-2147217396>. [Accessed: 01-Jul-2017].
- [117] Matt, "Kinect Interactive Catalog - Intuilab | Kinect Hacks." [Online]. Available: <http://www.kinecthacks.com/kinect-interactive-catalog-intuilab/>. [Accessed: 03-Jul-2017].
- [118] Microsoft, "Kinect-enabled solutions offer insights on retail customers – Kinect for Windows Product Blog," 2015. [Online]. Available: <https://blogs.msdn.microsoft.com/kinectforwindows/2015/01/26/kinect-enabled-solutions-offer-insights-on-retail-customers/>. [Accessed: 26-Jun-2017].
- [119] Zugara, "Virtual Style Sense." [Online]. Available: <http://zugara.com/virtual-dressing-room-technology/virtual-style-sense>. [Accessed: 01-Jul-2017].
- [120] T. S. Vieira, "Keepers of Intheris: Story and Aesthetics," 2017.
- [121] Ubisoft, "Assassin's Creed Origins para Xbox One, PS4, PC | Ubisoft (Brazil)." [Online]. Available: <https://assassinscreed.ubisoft.com/game/pt-br/home/>. [Accessed: 31-Jul-2017].
- [122] WOW, "Dalaran-WoW | The Best WOTLK WoW private server with Retail scripts." [Online]. Available: <http://www.dalaran-wow.com/>. [Accessed: 31-Jul-2017].
- [123] Fandom, "Yeti | WoWWiki | FANDOM powered by Wikia." [Online]. Available: <http://wowwiki.wikia.com/wiki/Yeti>. [Accessed: 31-Jul-2017].
- [124] J. . Rowling, "Harry Potter - Pottermore." [Online]. Available: <https://www.pottermore.com/explore-the-story/harry-potter>. [Accessed: 16-Aug-2017].
- [125] Disney, "Maleficent | Official UK Disney Site," 2014. [Online]. Available: <http://movies.disney.co.uk/maleficent>. [Accessed: 16-Aug-2017].
- [126] Ubisoft, "Ubisoft - Assassin's Creed." [Online]. Available: <https://www.ubisoft.com/en-gb/game/assassins-creed/>. [Accessed: 28-Aug-2017].
- [127] Fandom, "Indiana Jones and the Temple of Doom | Indiana Jones Wiki | FANDOM powered by Wikia." [Online]. Available: [http://indianajones.wikia.com/wiki/Indiana\\_Jones\\_and\\_the\\_Temple\\_of\\_Doom](http://indianajones.wikia.com/wiki/Indiana_Jones_and_the_Temple_of_Doom). [Accessed: 28-Aug-2017].
- [128] Universal Pictures, "The Mummy - Official Trailer (HD) - YouTube," 2017. [Online]. Available: <https://www.youtube.com/watch?v=ljHgzkQM2Sg>. [Accessed: 28-Aug-2017].
- [129] FANDOM, "Dalaran | WoWWiki | FANDOM powered by Wikia." [Online]. Available: <http://wowwiki.wikia.com/wiki/Dalaran>. [Accessed: 28-Aug-2017].
- [130] IMDb, "Parque Jurássico (1993) - IMDb." [Online]. Available:

- <http://www.imdb.com/title/tt0107290/>. [Accessed: 28-Aug-2017].
- [131] ZBrushCenter, "Insert Multi Mesh Repository." [Online]. Available: <http://www.zbrushcentral.com/showthread.php?170167-Insert-Multi-Mesh-Repository>. [Accessed: 05-Aug-2017].
- [132] Pixologic, "Pixologic :: Download Center." [Online]. Available: <https://pixologic.com/zbrush/downloadcenter/alpha/>. [Accessed: 05-Aug-2017].
- [133] MakeHuman, "User contributed assets | www.makehumancommunity.org." [Online]. Available: [http://www.makehumancommunity.org/content/user\\_contributed\\_assets.html](http://www.makehumancommunity.org/content/user_contributed_assets.html). [Accessed: 10-Aug-2017].
- [134] Cubebrush, "Cubebrush." [Online]. Available: <https://cubebrush.co/>. [Accessed: 17-Aug-2017].
- [135] VFX Brasil, "Maya fluid effects - Criando uma explosão - Parte 1 - YouTube," 2011. [Online]. Available: <https://www.youtube.com/watch?v=8hX41WEHRkM>. [Accessed: 05-Sep-2017].
- [136] MrToshihir0, "How to create a Sun in Maya - YouTube," 2012. [Online]. Available: <https://www.youtube.com/watch?v=F5C5hDKvSNo>. [Accessed: 05-Sep-2017].
- [137] G. Galer, "Light Streaks with Particle Flow / 3ds Max - YouTube," 2011. [Online]. Available: <https://www.youtube.com/watch?v=G8FCpDI6H7M>. [Accessed: 06-Sep-2017].
- [138] Adobe, "Mixamo Auto Biped Script." [Online]. Available: <https://www.mixamo.com/c/mixamo-auto-biped-script>. [Accessed: 26-Jul-2017].
- [139] B. Rosenhahn, T. Brox, D. Cremers, and H. Seidel, "Online smoothing for markerless motion capture," *Pattern Recognit.*, pp. 163–172, 2007.
- [140] Adobe, "Mixamo | Animations." [Online]. Available: <https://www.mixamo.com/#/?page=1&type=Motion%2CMotionPack>. [Accessed: 04-Sep-2017].
- [141] Blog Tonka3D, "Computador ideal para modelagem, animação 3d e renderização," 2010. [Online]. Available: <http://tonka3d.com.br/blog/computador-ideal-para-modelagem-animacao-3d-e-renderizacao/>. [Accessed: 25-Jul-2017].
- [142] Redshift, "Weblet Importer." [Online]. Available: <https://www.redshift3d.com/>. [Accessed: 25-Jul-2017].
- [143] Wired, "Animando um Blockbuster: Como o Pixar construiu Toy Story 3 | WIRED," 2010. [Online]. Available: [https://www.wired.com/2010/05/process\\_pixar/all/1](https://www.wired.com/2010/05/process_pixar/all/1). [Accessed: 26-Aug-2017].
- [144] "Storyboard - PIXAR." [Online]. Available: <http://pixar-animation.weebly.com/storyboard.html>. [Accessed: 26-Aug-2017].
- [145] AnimationInfinity, "Toy Story Progression Reel - YouTube," 2016. [Online]. Available: <https://www.youtube.com/watch?v=dNdPs6P1HIQ>. [Accessed: 28-Aug-2017].
- [146] ASHO1997, "Pixar Short Film Research," 2017. [Online]. Available: <https://kcimgdjordanaalexisebanks.wordpress.com/2017/02/04/pixar-short-film-research/>. [Accessed: 26-Aug-2017].

- [147] A. Guenette, "File:Sintel-face-morph.png - Wikipedia," 2011. [Online]. Available: <https://en.wikipedia.org/wiki/File:Sintel-face-morph.png>. [Accessed: 28-Aug-2017].
- [148] B. Stein, "Graphical Marionette on Vimeo." [Online]. Available: <https://vimeo.com/43287234>. [Accessed: 28-Jul-2017].
- [149] H. Dingman, "Kinect v2 gets a Windows launch date, \$200 price tag | PCWorld," 2014. [Online]. Available: <http://www.pcworld.com/article/2450755/kinect-v2-gets-a-windows-launch-date-200-price-tag.html>. [Accessed: 09-Jul-2017].
- [150] P. Gutierrez, "Kinect sensors used to improve x-ray process - News - IoT Hub," 2015. [Online]. Available: <https://www.iothub.com.au/news/kinect-sensors-used-to-improve-x-ray-process-412529>. [Accessed: 09-Jul-2017].
- [151] AVA Retail, "AVA Retail - Products." [Online]. Available: <http://www.avaretail.com/avaretail/ProductDetails.aspx#retail-analytics-dashboard>. [Accessed: 10-Jul-2017].

## 7. ANEXOS

### 7.1. TABELA COM AS CARACTERÍSTICAS DA PEN TABLET

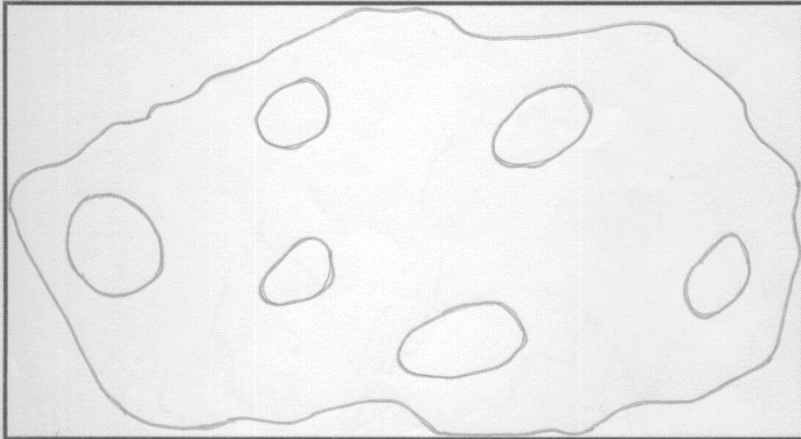
Tabela 6 - Características e funcionalidades da Pen tablet HUIION H610 [30].

<b>Technology</b>	<b>Electromagnetic Digitizer</b>
<b>Active Area</b>	10" x 6.25"
<b>Resolution</b>	4000 LPI (Lines Per Inch)
<b>Report Rate</b>	200 RPS
<b>Pen Pressure</b>	2048 Levels
<b>Reading Height</b>	0.4"
<b>Supply Voltage</b>	DC 5V
<b>Power Consumption</b>	0.35 W
<b>Interface Support</b>	USB 1.1
<b>OS Support</b>	Win 10/Windows 8/Windows 7, Mac OS X10.8.0 and above
<b>Compliance</b>	CE/FCC/CCC
<b>Dimensions/piece</b>	353 x 245 x 10 mm
<b>Net Weight/ piece</b>	595g
<b>Gross Weight/piece</b>	1.2kg
<b>Out Carton Dimensions</b>	407x325x45mm
<b>Quantity per Carton</b>	10 pieces
<b>Net Weight/carton</b>	12kg
<b>USB Cable Length</b>	1.5m
<b>Gross Weight/carton</b>	13.2kg
<b>Type</b>	No
<b>Express Keys</b>	8
<b>Function Keys</b>	16

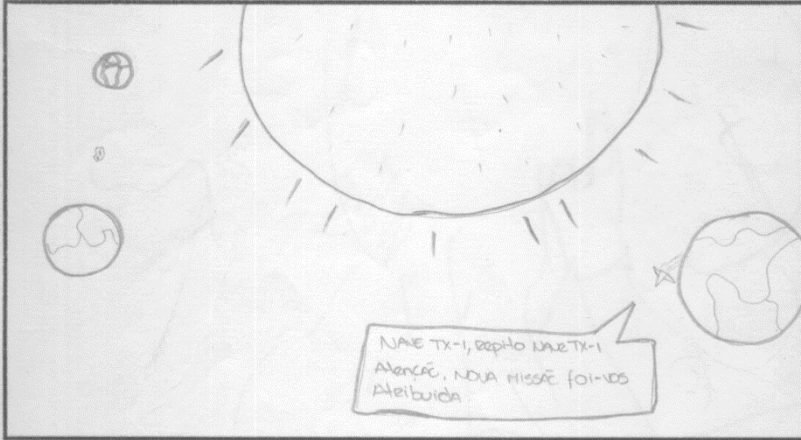
## 7.2. STORYBOARD

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>1</u> / <u>    </u>

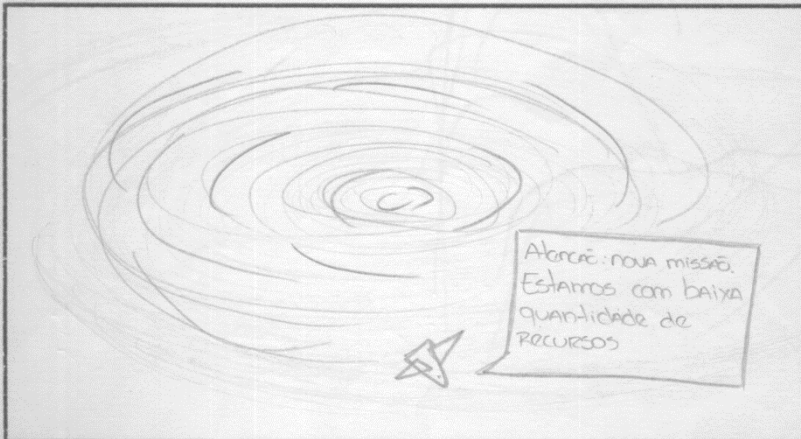
  

<p>1 ✓</p> 	<p>Scene <u>1</u> Shot # <u>1</u></p> <p>- Meteorito</p> <p>- Meteorito se afasta até desaparecer e se vê os planetas e o sol</p> <p>Tempo: 3s</p> <p>Location: <u>Orbita Mars</u></p>
---	--

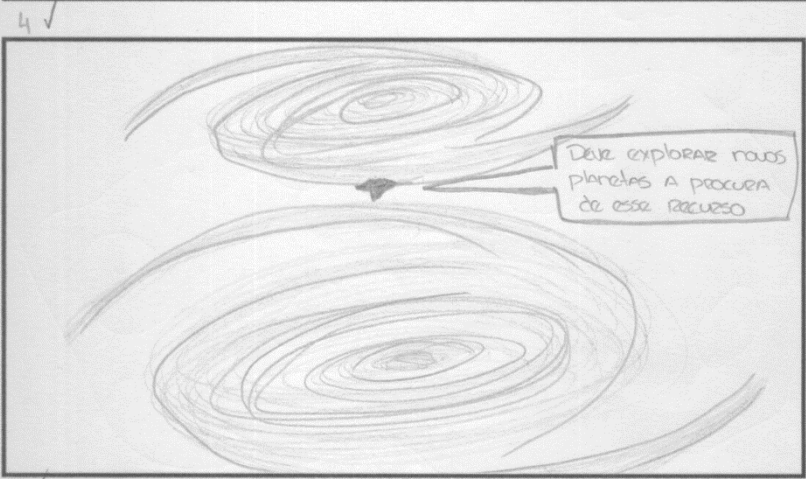
<p>2 ✓</p> 	<p>Scene <u>1</u> Shot # <u>1</u></p> <p>- CAMARA se esta afastando pouco a pouco</p> <p>- NAE vai em direção a camera</p> <p>Transição: <u>Coleta</u></p> <p>Tempo: 17s</p> <p>Location: <u>Orbita Mars</u></p>
---	--

<p>3 ✓</p> 	<p>Scene <u>1</u> Shot # <u>2</u></p> <p><u>Pico Perseu</u></p> <p>Transição: <u>Zoom cut</u></p> <p>Tempo: 12s</p> <p>Location: <u>Galaxia Tróvão (Mars)</u></p>
---	---

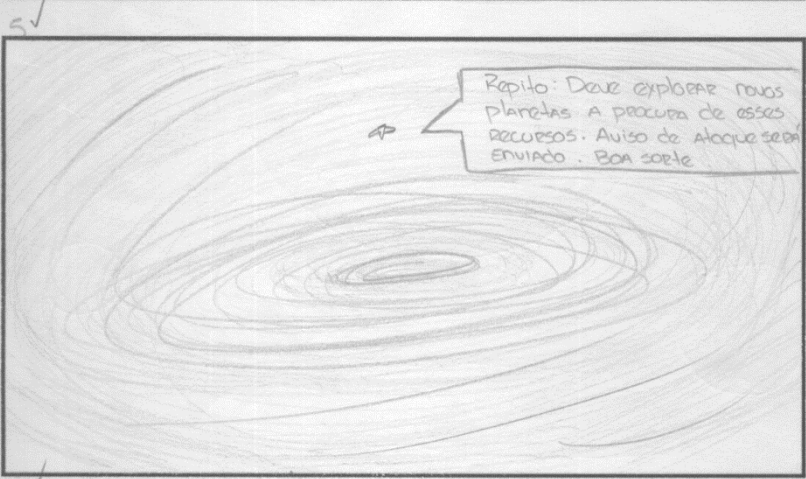
→ COZINHAR o início da Animação

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>2</u> / <u>   </u>



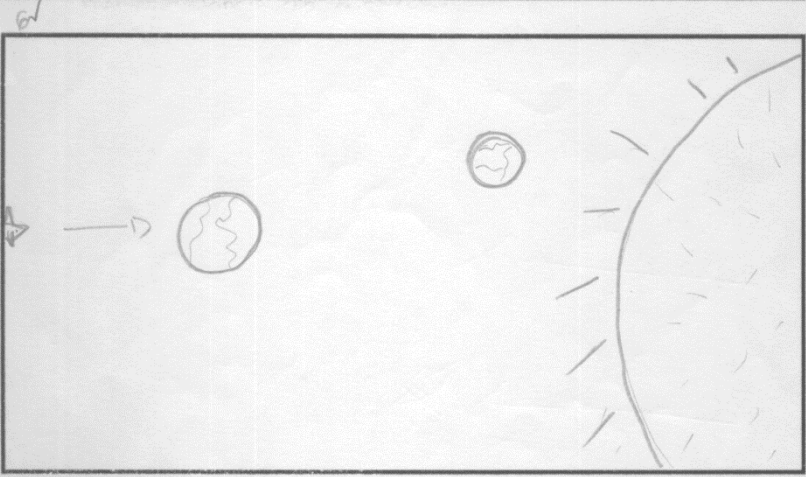
Deve explorar novos planetas a procura de esse recurso

Scene 1 Shot # 2  
 - Nave Alabassa da galaxia Ingrid para a Emax  
 Plano Paralelo  
 Zoom: in  
 Tempo: 11s  
 Location Espaço



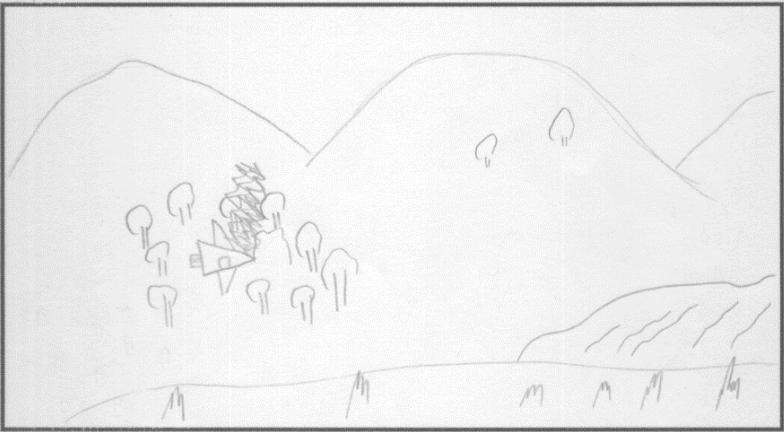
Repeto: Deve explorar novos planetas a procura de esses recursos. Aviso de Alaque será enviado. Boa sorte

Scene 1 Shot # 2  
 Plano Paralelo  
 Transição: Fundido  
 Tempo: 6s  
 Location Galaxia Emax



Scene 1 Shot # 3  
 - Camera aproxima do planeta Ithris  
 - Nave aparece em direção ao planeta Ithris  
 Transição: Fundido P  
 Tempo: 24s  
 Location Oribita Emax

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>31</u>



Scene 2 Shot # 1

---

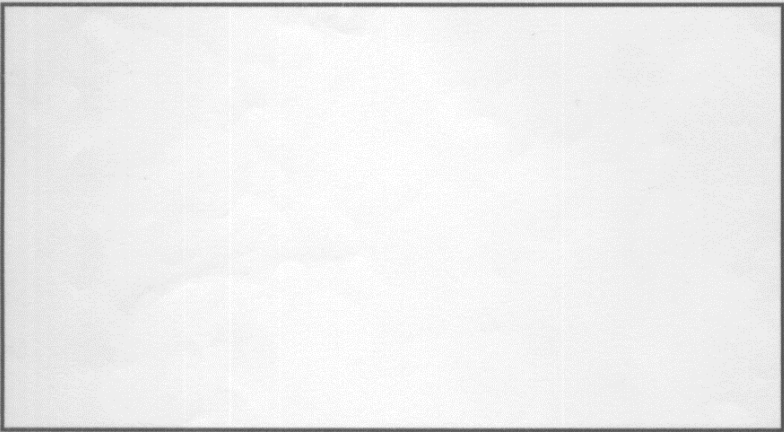


---

Transição: Corte

Tempo: 6s

Location Floresta



Scene Shot #

---



---

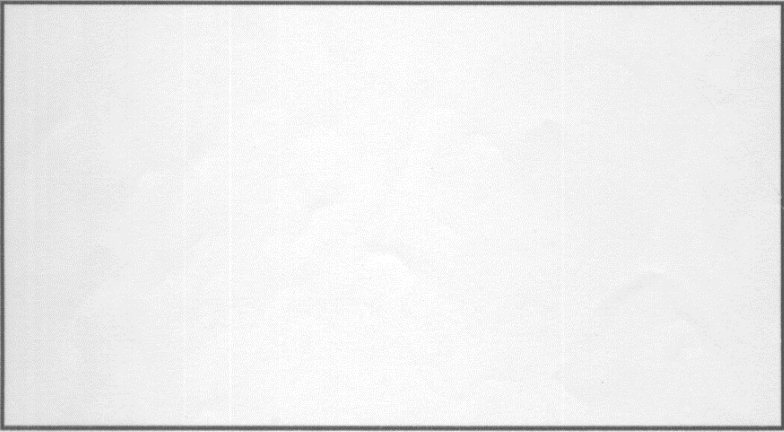


---



---

Location



Scene Shot #

---



---



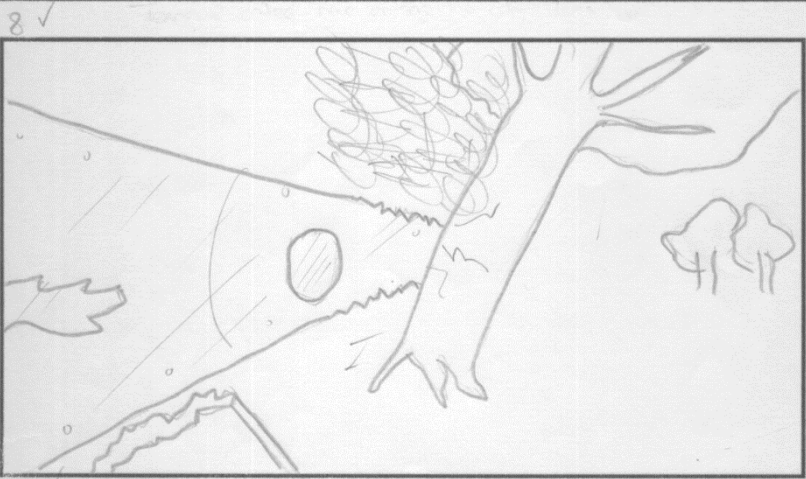
---



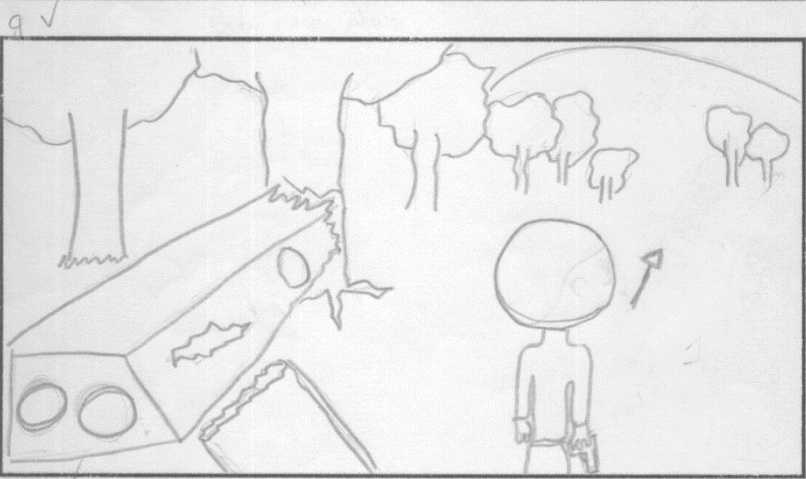
---

Location

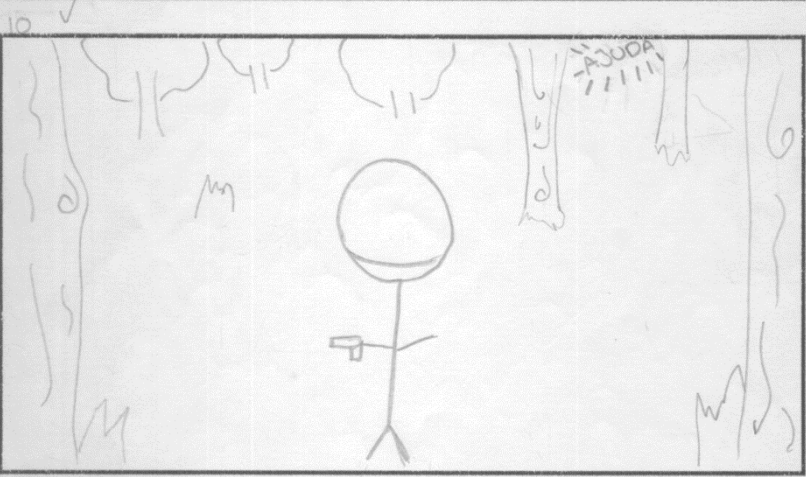
Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: 4 /



Scene 2 Shot # 2  
 Naive bale  
 Plano Panorâmico  
 Transição: Corte  
 Tempo: 5s  
 Location Floresta

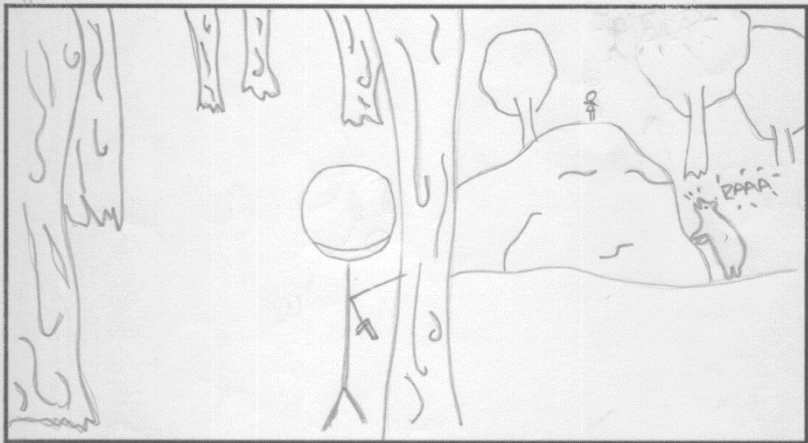


Scene 2 Shot # 3  
 Plano Médio  
 Explora a floresta  
 A procura de recursos  
 Transição: Corte  
 Tempo: 6s  
 Location Floresta



Scene 2 Shot # 4  
 Nifast explora floresta  
 Ouve um pedido de Ajuda -> Isene voz amoff  
 Transição: Corte  
 Tempo: 12s  
 Location Floresta

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>5</u> / <u>1</u>



Scene 2 Shot # 5

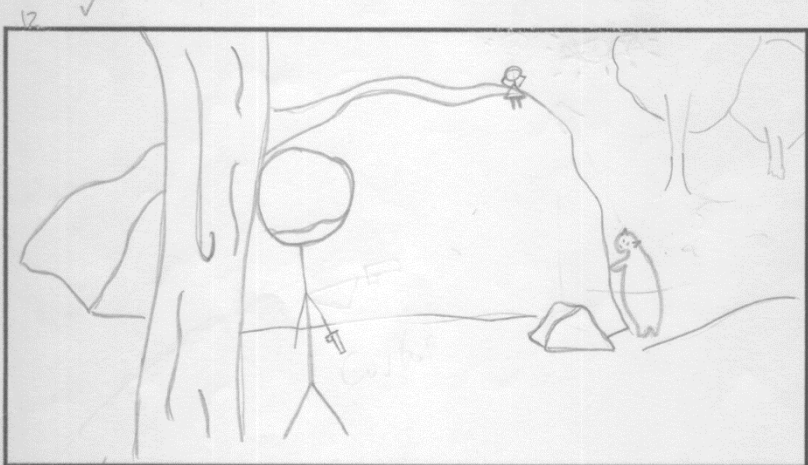
Plano Geral

Nifast vê Isere em perigo

Transição: Corte

Tempo: 6s

Location Floresta



Scene 2 Shot # 6

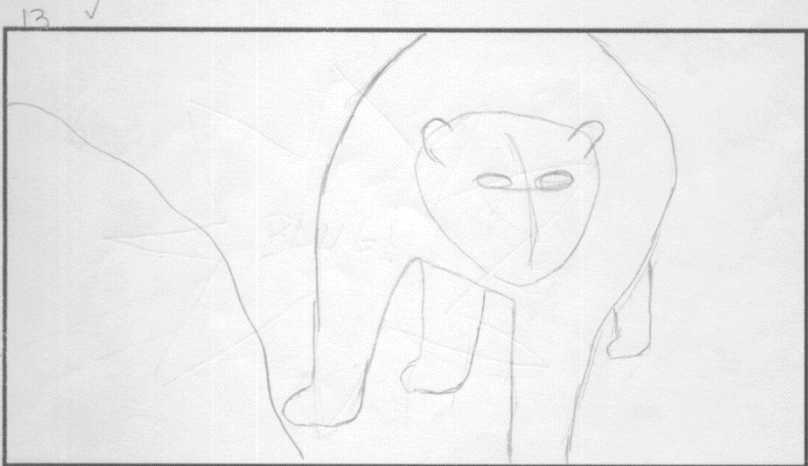
Nifast grita para o Urso

e Levanta a arma para disparar contra ele

Transição: Corte

Tempo: 8s

Location Floresta



Scene 2 Shot # 7

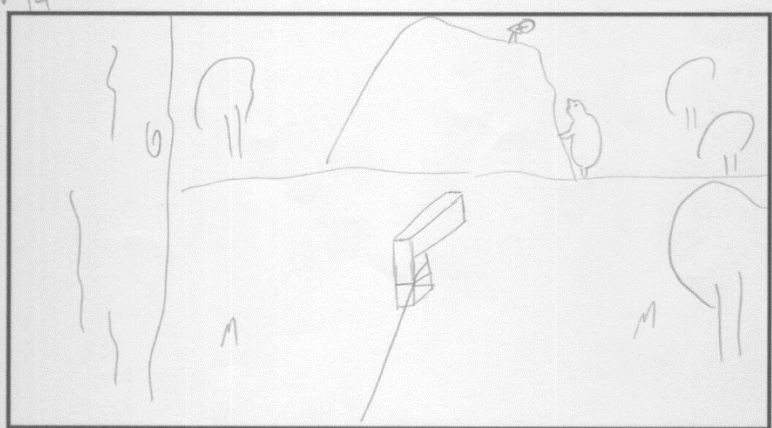
Urso tenta atacar Isere

Tempo: 10s

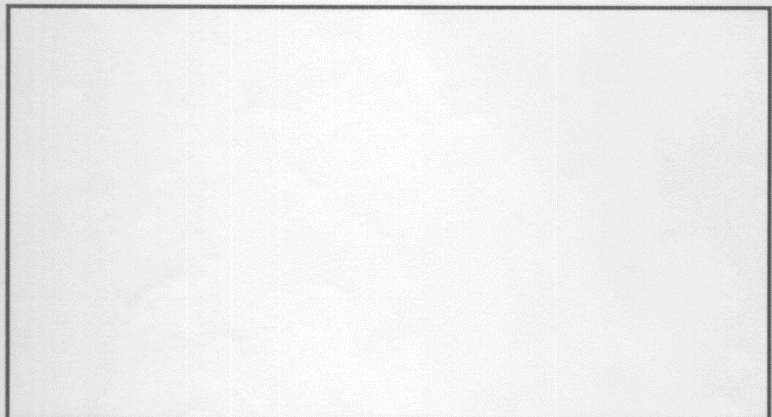
Location Floresta

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>6 /</u>

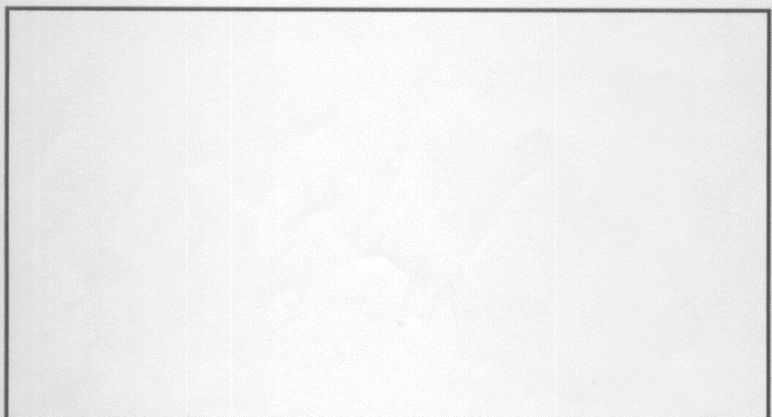
✓ 14



Scene	<u>2</u>	Shot #	<u>8</u>
<u>Nifast dispara</u>			
<u>AO URSO</u>			
Tempo: <u>10s</u>			
Location <u>Floresta</u>			

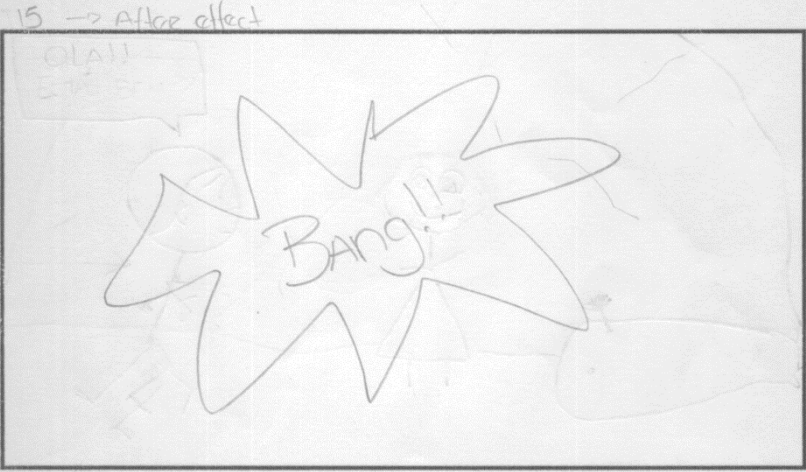


Scene		Shot #	
Location			



Scene		Shot #	
Location			

Project: <i>Projeto de Folhas</i>	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>7</u> / <u>    </u>



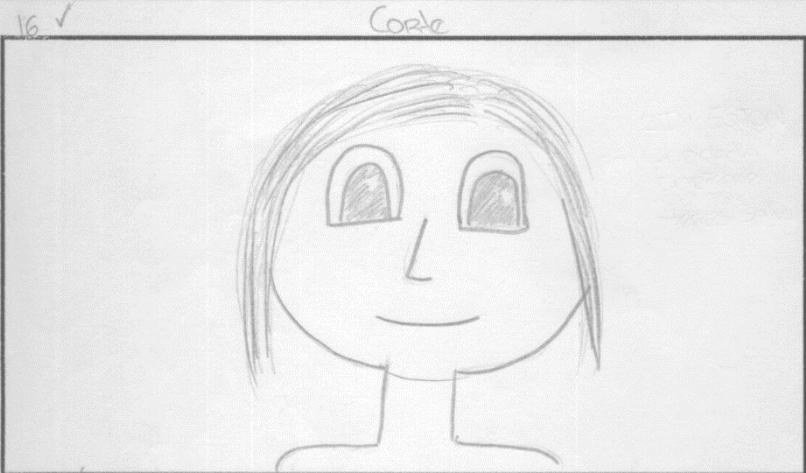
Scene 2 Shot # 9

Fica Branco e  
ouve-se o tiro

Transição: Fundido  
Branco

Tempo: 3s

Location



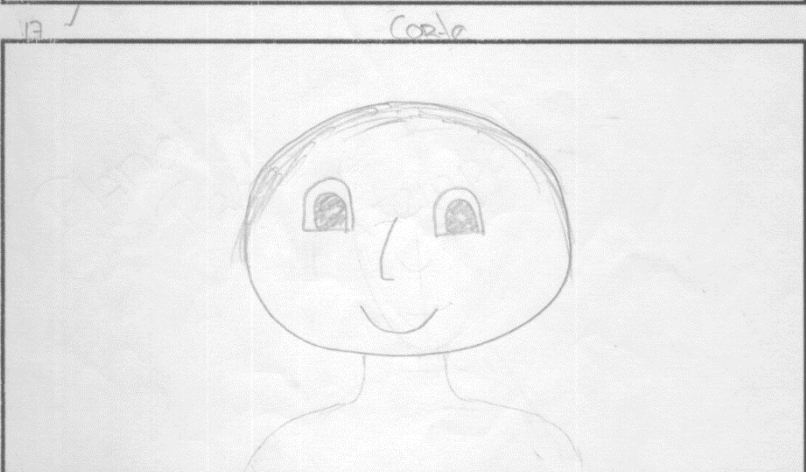
Scene 2 Shot # 10

Plano Médio

Transição: Corte

Tempo: 3

Location Floresta



Scene 2 Shot # 11

Nífast

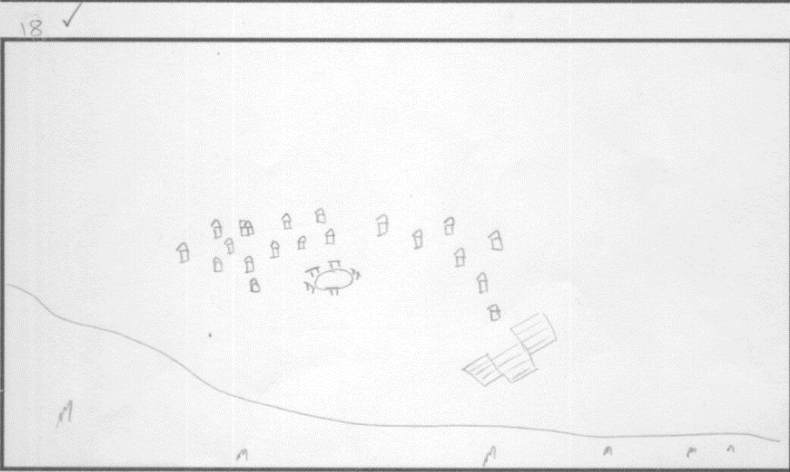
Transição: Fundido

Tempo: 3

Location     

Fundido a Preta (Tempo)

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: 8 /



Scene 3 Shot # 1

---



---

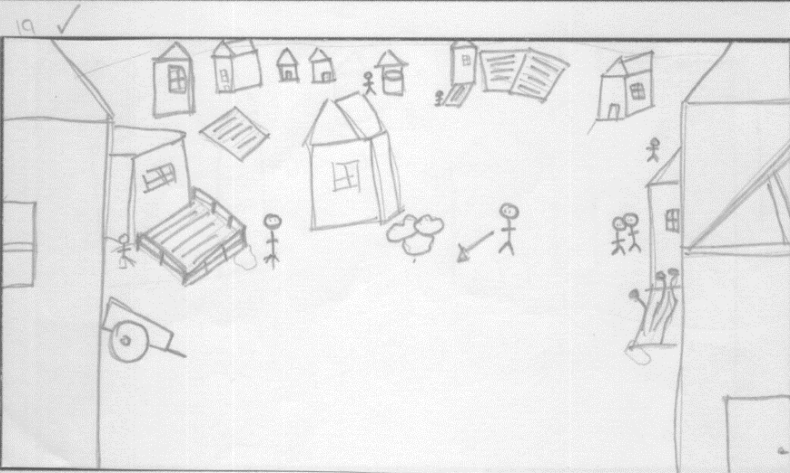


---



---

Location Cidade

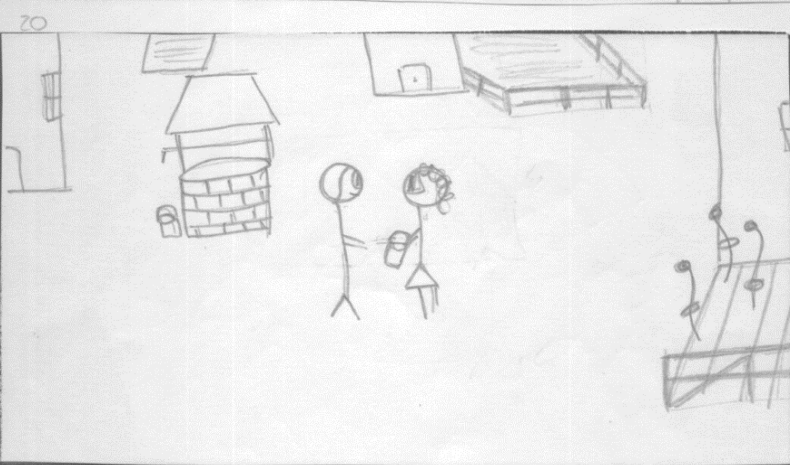


Scene 3 Shot # 2

Plano Panorâmico

Tempo: 11s

Location Cidade

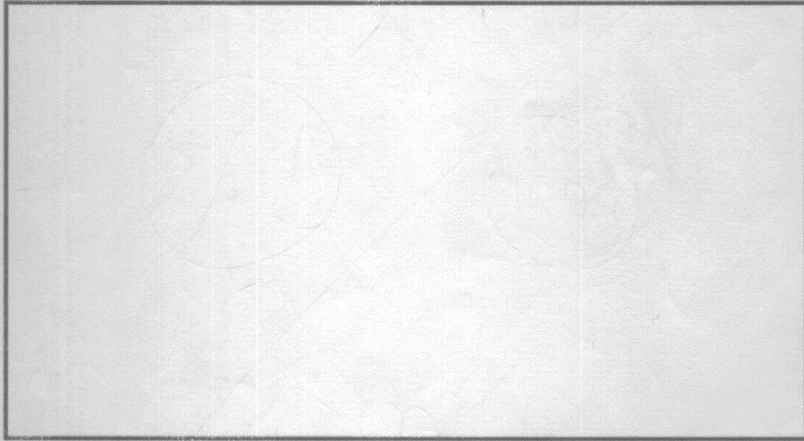


Scene 3 Shot # 3

Tempo: 6s

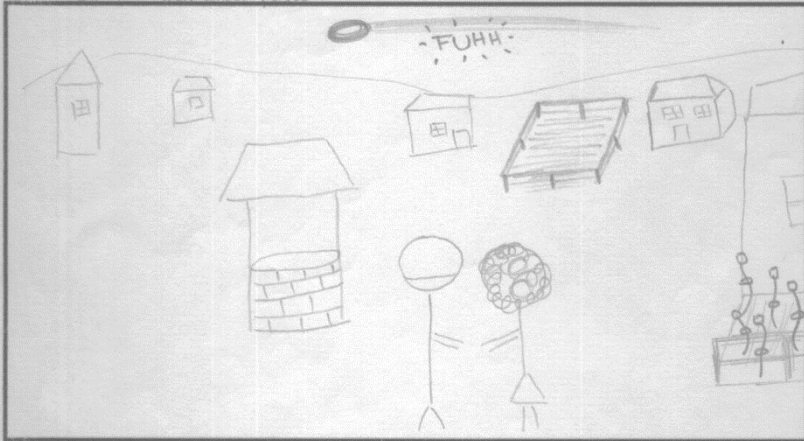
Location Cidade

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>9</u> / <u>    </u>



Scene	Shot #
Location	

21. CAMERA PRIMEIRA PESSOA

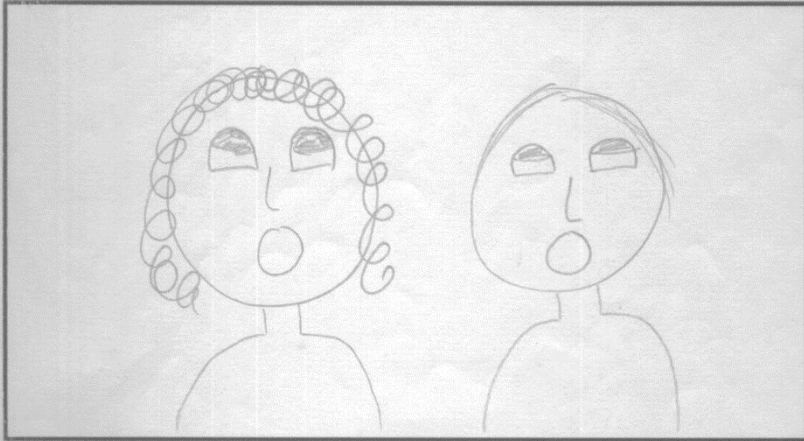


Scene	3	Shot #	4
Location	Vila / Campo		

Transição: Corte

Tempo: 5s

22 ✓

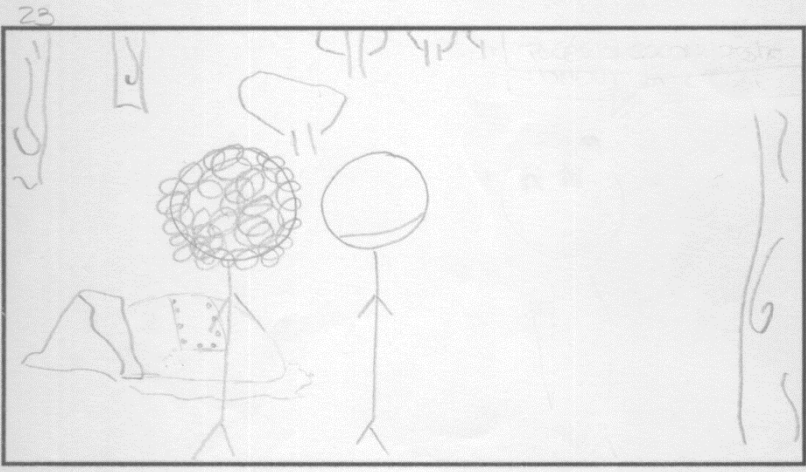


Scene	3	Shot #	5
Location	Vila / Campo		

Transição: Corte

Tempo: 4s

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>10</u> / <u>    </u>

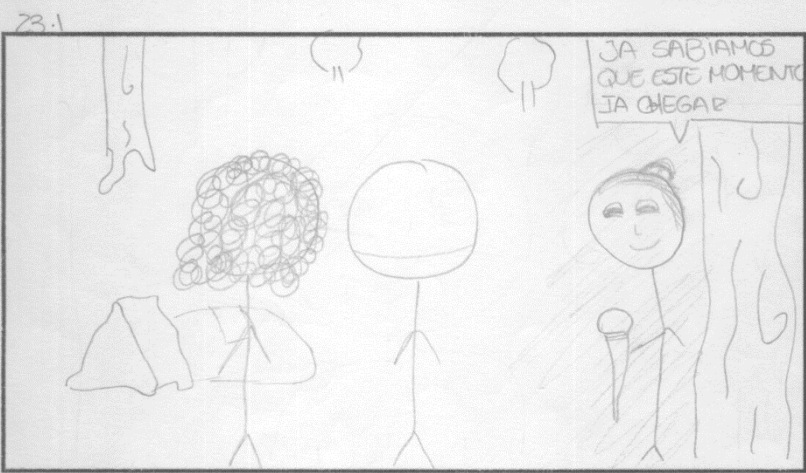


Scene 4 Shot # 1

Transcrip: Coele

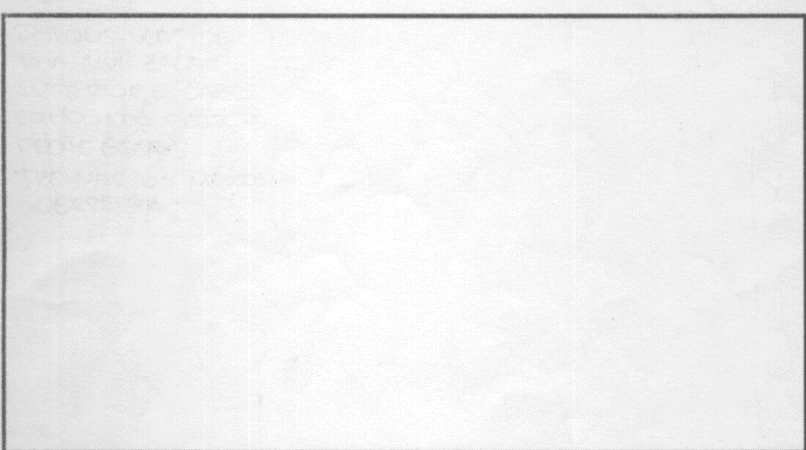
Tempo: 8s

Location Fiesta



Scene Shot #

Location

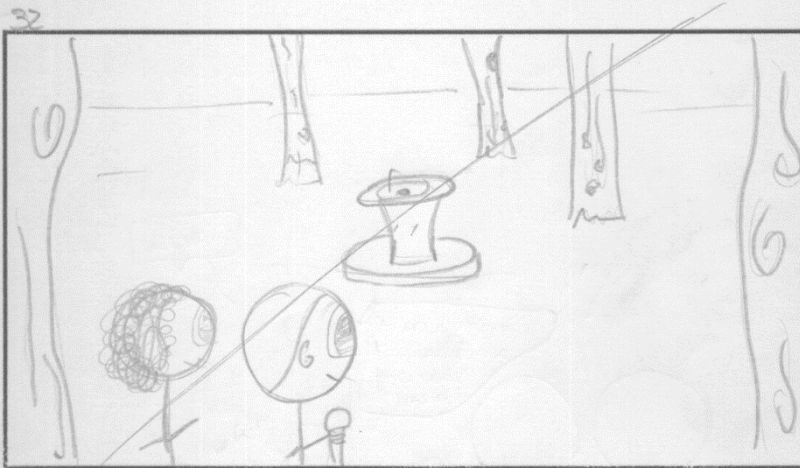


Scene Shot #

Location



Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: <u>12 /</u>



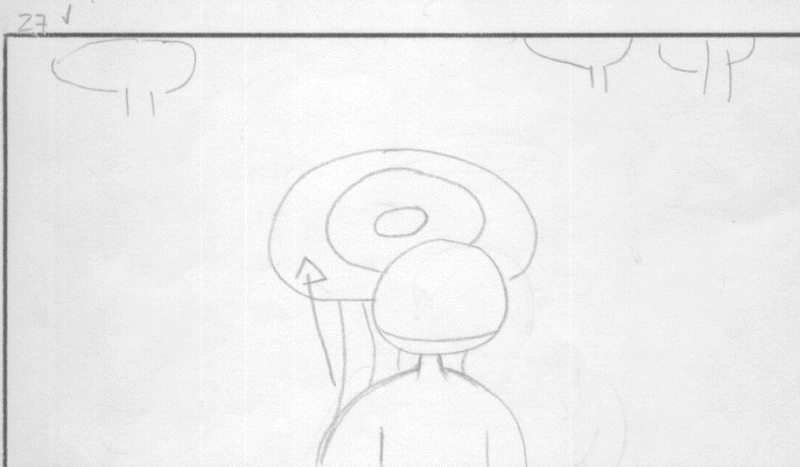
Scene 6 Shot # 4

Plano Americano

Ricado

Transição: Corte

Location

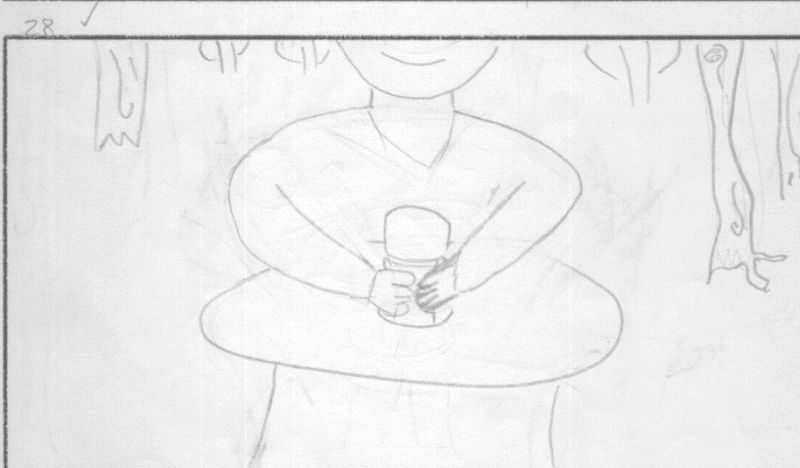


Scene 5 Shot # 3

Transição: Corte

Tempo: 8s

Location Porta / Fiores-la



Scene 5 Shot # 4

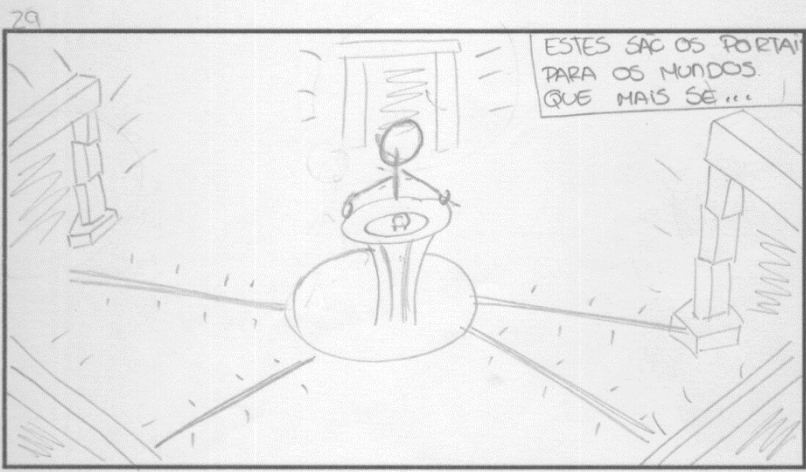
Abertura dos  
portais

Transição: Fudido

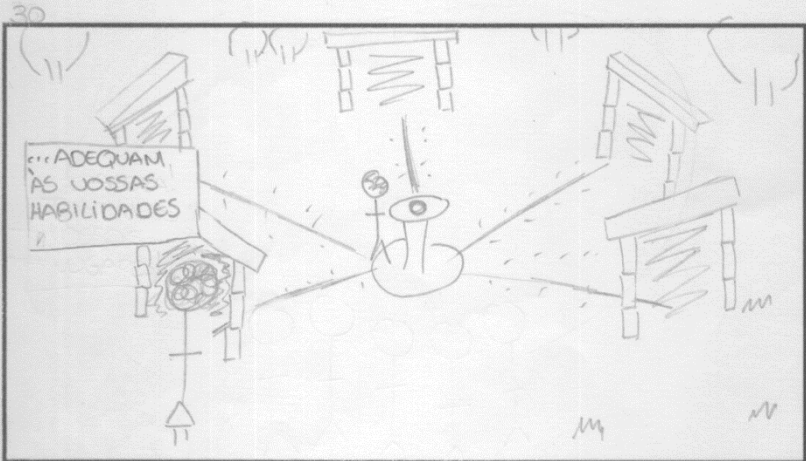
Tempo: 4

Location Porta / Fiores-la

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: 18 /



Scene 5 Shot # 5  
 Nifast abre os portais mágicos  
 Prádo  
 Transição: Corte  
 Tempo: 12  
 Location Templo



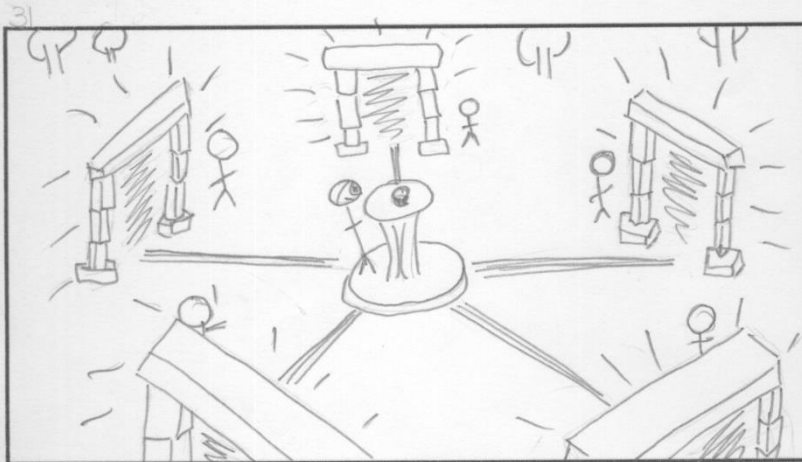
Scene 5 Shot # 6  
 Tem de fazer as escolhas para cada portal  
 Transição: Fundido  
 Tempo: 8s  
 Location Templo

31 -D NARRAÇÃO

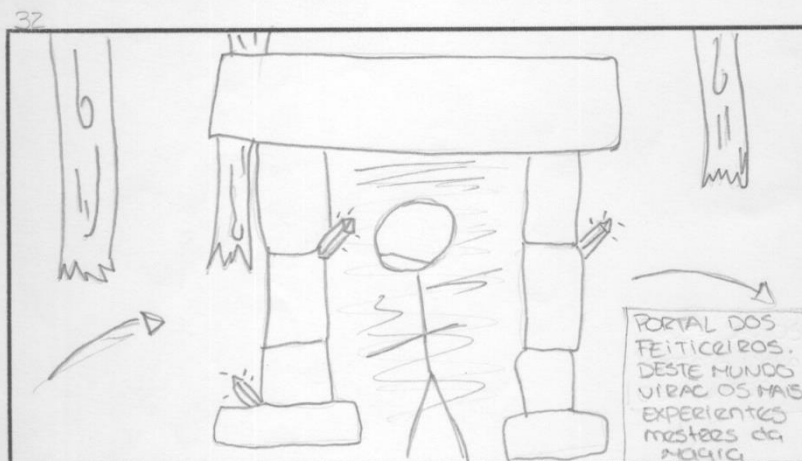
VOCÊS DEVEM TREINAR, APREFEIÇOAR E DOMINAR AS VOSSAS HABILIDADES

Scene Shot #  
 Location

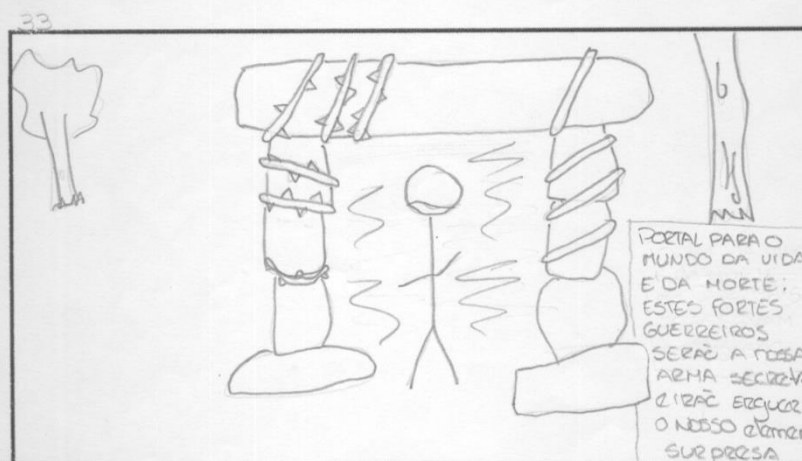
Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: 14 / ____



Scene 5 Shot # 7  
 Picado  
 Transição: Corte  
 Tempo: 05  
 Location: Portais / Floresta

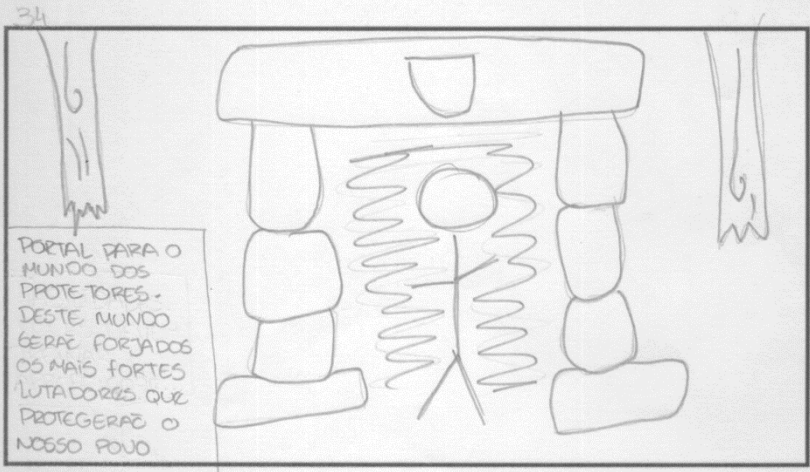


Scene 5 Shot # 8  
 Isone FALA dos portais Normal  
 Tempo: 10s  
 Portal Feiticeiros  
 Transição:  
 CAMARA ROLA A volta dos portais  
 Location: Portal / Floresta



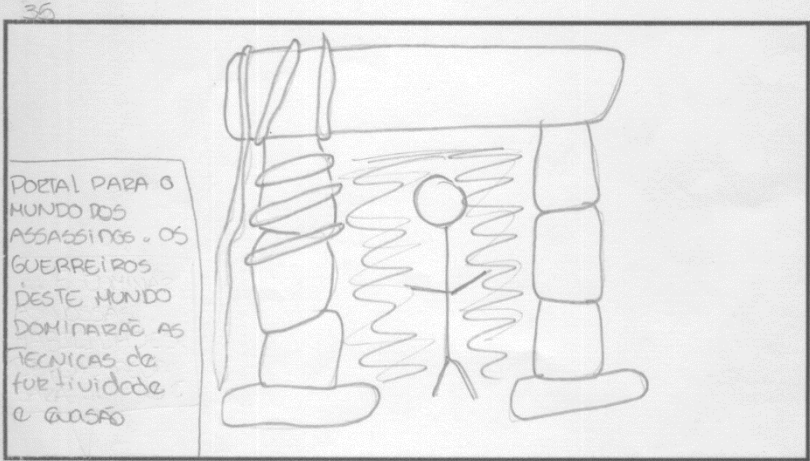
Scene 5 Shot # 9  
 Isone Fala dos portais Portal dos Guerreiros  
 Transição: Rotação  
 Tempo: 10s  
 Location: Portal / Floresta

Project:	Date:
Director:	Storyboards:
1st A.D.:	U.P.M.:
	Page: 15 /



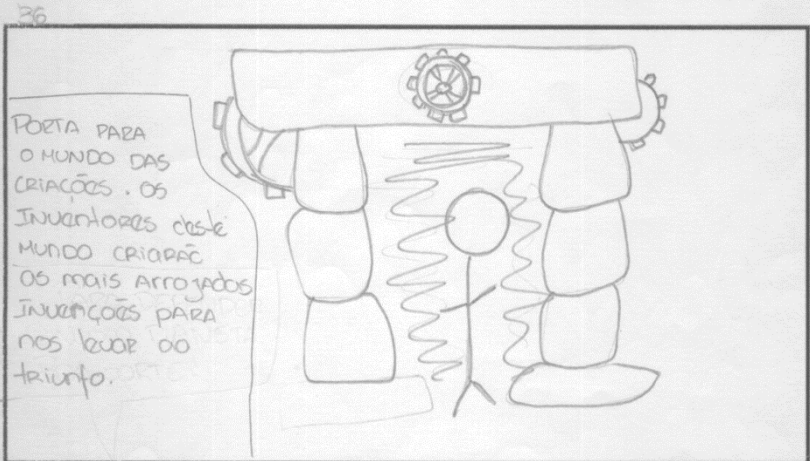
PORTAL PARA O MUNDO DOS PROTETORES. DESTE MUNDO SERÃO FORJADOS OS MAIS FORTES LUTADORES QUE PROTEGERÃO O NOSSO POVO

Scene 5 Shot # 10  
 Isere fala sobre portais  
 Portal dos Protetores  
 Transição: Rotação  
 Tempo: 10s  
 Location Portal/Floresta



PORTAL PARA O MUNDO DOS ASSASSINOS. OS GUERREIROS DESTE MUNDO DOMINARÃO AS TÉCNICAS DE FURTIVIDADE E CAÇÃO

Scene 5 Shot # 11  
 Isere fala sobre portais  
 Portal Assassinos  
 Transição: Rotação  
 Tempo: 10s  
 Location Portal/Floresta



PORTA PARA O MUNDO DAS CRIAÇÕES. OS INVENTORES DESTE MUNDO CRIARÃO OS MAIS ARROJADOS INVENÇÕES PARA NOS LEVAR AO TRIUNFO.

Scene 5 Shot # 12  
 Isere fala sobre portais  
 Portal Inventores  
 Transição: Fundo P  
 Tempo: 10s  
 Location Portal/Floresta



### 7.3. CONCEITO DOS MUNDOS

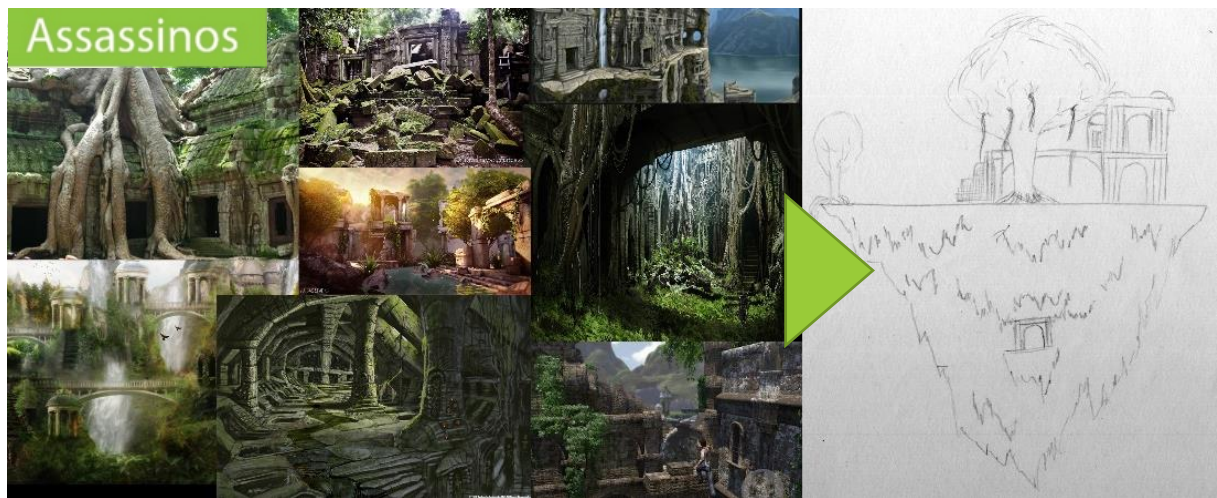


Imagem 147 - Inspiração para o mundo dos assassinos e o seu respetivo desenho.

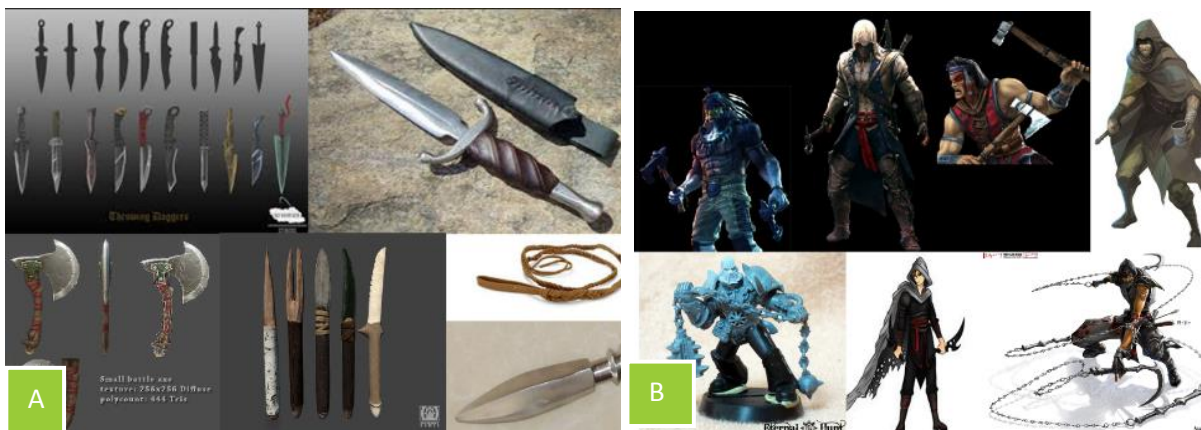


Imagem 148 - Inspirações. (A) Arma dos assassinos; (B) Guerreiros assassinos.





Imagem 151 - Inspiração para o mundo dos feiticeiros e o seu respetivo desenho.

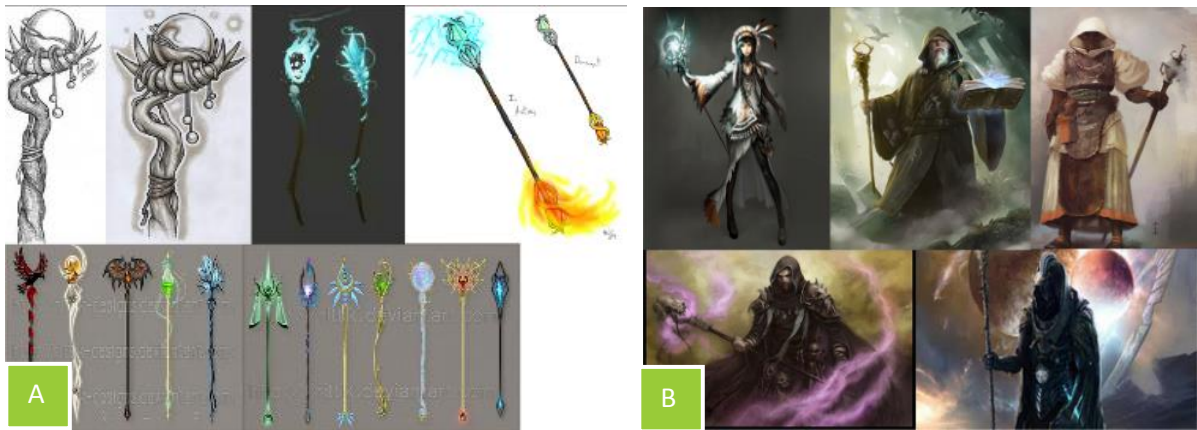


Imagem 152 - Inspirações. (A) Arma dos feiticeiros; (B) Guerreiros feiticeiros

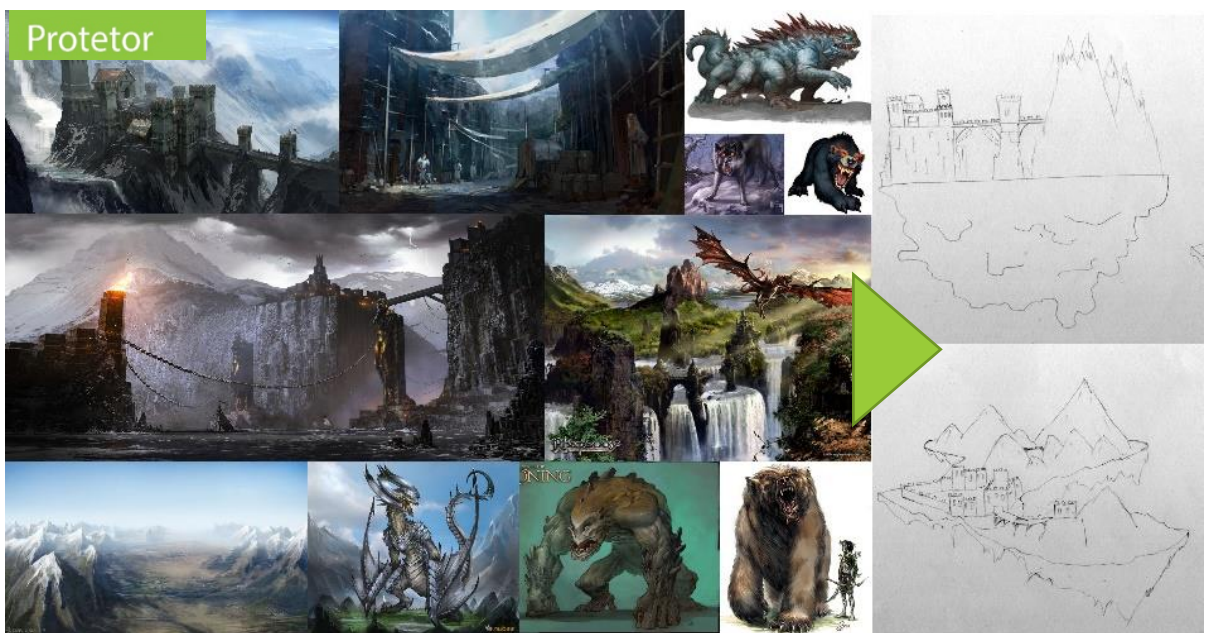


Imagem 153 - Inspiração para o mundo dos protetores e o seu respetivo desenho.

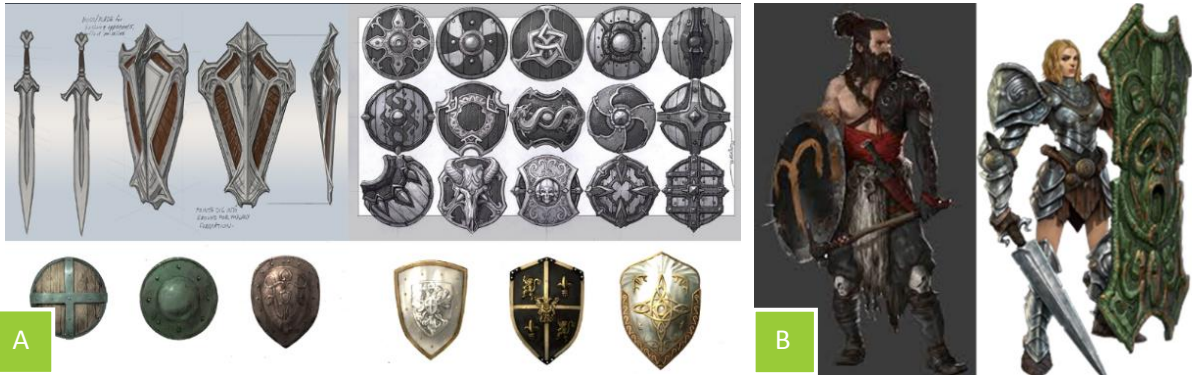
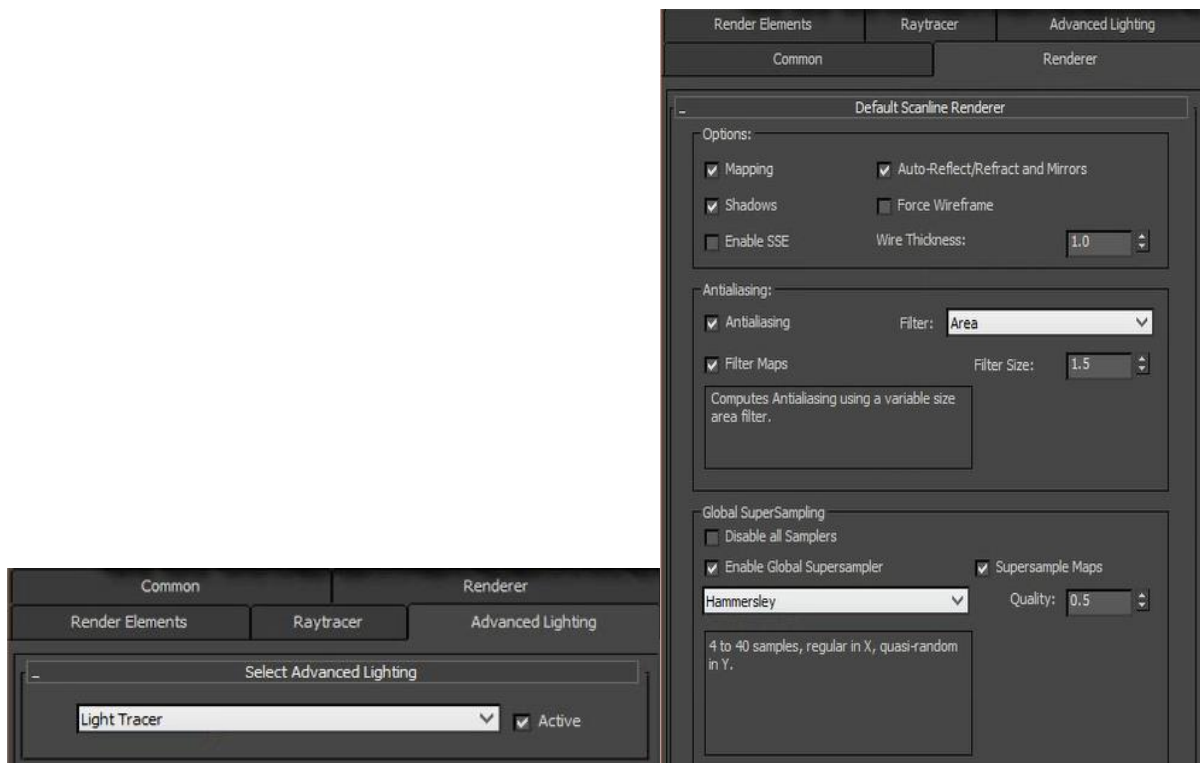


Imagem 154 – Inspirações. (A) Arma dos protetores; (B) Guerreiros protetores

## 7.4. OPÇÕES DE RENDER PARA REALIZAR A PROJEÇÃO



## 7.5. TABELAS COM OS PASSOS DE TODOS OS EFEITOS VISUAIS

Tabela 7 - Tabela com os passos necessários para chegar a uma determinada opção na elaboração do rasto da nave.



Descrição dos passos necessários				Passos
				
Attribute Editor	particleShapes	Render Attributes	Particle Render Type: Cloud(s/w)	1
			Current Render Type -> Radius: 0.850	
		Lifespan Attributes	lifespanPP only	2
		Per Particle		3
	Add Dynamic Attribute	Opacity: Add Per Object Attribute	4	
		Color: Add Per Object Attribute		
		General -> Particle: RadiusPP		
particleCloud	Common Material Attributes		5	
particleShapes	Common Material Attributes		6	
	Transparency		7	
Rendering Editors	Hypershade			8

Tabela 8 – Tabela com os passos necessários para chegar a uma determinada opções na elaboração da explosão.

Descrição dos passos necessários				Passos
				
Attribute Editor	fluidShape	Container Properties	Base Resolution: 20	1
			Size: 20 20 20	
	fluidEmitter	Basic Emitter Attribute	Emitter type: Volume	2
		Volume Emitter Attribute	Volume shape: Sphere	3
		Fluid Attributes	Density: 60 60 0 (frame1 frame5 frame 10)	4
Heat: 60 0 0 (frame1 frame5 frame 10)				
Fuel: 90 0 0 (frame1 frame5 frame 10)				
Animation Editors	Graph Editor			5
	fluidShape		Temperature: Dynamic Grid	6

<b>Attribute Editor</b>	Contents Method	Fuel: Dynamic Grid			
		Color method: Dynamic Grid			
	Shading	Transparency: 0.034		7	
	Contents Details	Fuel	Reaction speed: 3		8
			Ignition temperature: 0.005		
			Heat released: 60		
		Temperature	Bouyancy: 1.3		9
			Dissipation: 4		
			Noise: 0.173		
		Turbulence	Strength: 0.069		10
		Velocity	Swirl: 1.156		11
	Noise: 0.173				
	Density	Dissipation: 0.49		12	
	Dynamic simulation	Simulation rate scale: 2		13	
	Container properties	Base resolution: 50		14	
	Lighting	Self shadow (check)		15	
	Shading Quality	Render interpolation: smooth		16	
	Shading	Opacity	Input bias: 0.150	17	
	Texture	Texture incandescence (check)		18	
		Texture opacity (check)			
Depth max: 4					
Frequency: 2.7					
Inflection (check)					
	Texture time	(botão direito) Create new expression	19		
Shading	Editar grafico		20		

## 7.6. TABELA DE HORAS DE RENDERS

Cena	Nº de Frames total	Horas de render
1	1000	00 meses 00 dias 00 horas 53 minutos
2	709	00 meses 00 dias 05 horas 03 minutos
3	520	00 meses 00 dias 01 horas 00 minutos
4	551	00 meses 00 dias 02 horas 14 minutos
5	110	00 meses 00 dias 00 horas 15 minutos
6	161	00 meses 00 dias 20 horas 37 minutos
7	550	00 meses 00 dias 03 horas 45 minutos
8	1501	00 meses 09 dias 06 horas 09 minutos
9	1001	00 meses 03 dias 08 horas 24 minutos
10	1401	00 meses 06 dias 11 horas 25 minutos
11	301	00 meses 01 dias 00 horas 28 minutos
12	401	00 meses 00 dias 16 horas 20 minutos
13	351	00 meses 00 dias 04 horas 05 minutos
14	1861	00 meses 14 dias 09 horas 41 minutos
15	101	00 meses 00 dias 06 horas 27 minutos
16	471	00 meses 00 dias 00 horas 16 minutos
17	361	00 meses 02 dias 05 horas 29 minutos
18	261	00 meses 00 dias 10 horas 24 minutos
19	248	00 meses 00 dias 11 horas 16 minutos
20	3526	06 meses 12 dias 01 horas 13 minutos
21	350	00 meses 08 dias 11 horas 55 minutos
22	1798	00 meses 00 dias 08 horas 13 minutos
23	1798	00 meses 02 dias 23 horas 28 minutos
24	626	00 meses 01 dias 15 horas 29 minutos
25	626	00 meses 00 dias 00 horas 43 minutos
<b>Total</b>	<b>29584</b>	<b>08 meses 05 dias 17 horas 12 minutos</b>

## 7.7. TABELA DE RENDERS

Tabela 9 - Tabela de como foram divididas as cenas e como foram distribuídas pelos computadores.

Cenas	Frames	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
1	1;1000					1;1000			
2	1;710					1;710			
3	1;261					1;261			
4	0;550					0;550			
5	440;550					440;550			
6	0;160				0;160				
7	0;550				0;550				
8	0;1500						0;750	751;1500	
19	0;1000								0;1000
10	0;1400				0;999	1000;1400			
11	0;300				0;300				
12	0;400				0;24900	250;400			
13	0;350				0;350				
14	0;1860	0;599	600;1860						
15	0;100					0;100			
16	0;470				0;470				
17	0;360				0;360				
18	0;260	0;129	130;260						
19	252;500				252;500				
20	0;2350	631;679 1100;1299	680;999	0;584	2320;232 1000;1059	2000;2319 2333;2350 585;599 1060;1099	1800;1999	600;630	1300;1799

<b>21</b>	2350; 2700						2350; 2700		
<b>22</b>	2830; 4374						2830; 4374		
<b>23</b>	2830; 4374					2830; 4374			
<b>24</b>	4374; 5000	4374;5 000							
<b>25</b>	4374; 5000					4374; 5000			