

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO RIO
GRANDE DO NORTE

JOÃO PAULO NUNES DE LIMA

MARIA EMANUELE DA COSTA BERNARDO

PLUME: ROLE-PLAYING GAME

JOÃO CÂMARA – RN

2016

JOÃO PAULO NUNES DE LIMA

MARIA EMANUELE DA COSTA BERNARDO

PLUME: ROLE-PLAYING GAME

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em informática.

Orientador: Prof. Me. Diego Henrique Oliveira de Souza

JOÃO CÂMARA – RN

2016

JOÃO PAULO NUNES DE LIMA

MARIA EMANUELE DA COSTA BERNARDO

PLUME: ROLE-PLAYING GAME

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em informática.

Orientador: Prof. Me. Diego Henrique Oliveira de Souza

Aprovado em __/__/____

BANCA EXAMINADORA

Prof. Me. Diego Henrique Oliveira de Souza

Orientador – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Prof. Me. Igor Augusto de Carvalho Alves

Coordenador do Curso Técnico Integrado em informática – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Prof. Me.

Professor – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Dedicamos este trabalho aos que acreditaram que conseguiríamos e aos que nos inspiraram à ideia e ao resultado.

AGRADECIMENTOS

A Junior Nunes que nos ajudou em um momento complicado do trabalho.

A nossos familiares, que mesmo sem compreender ou gostar de jogos digitais, permitiram que nosso trabalho fosse feito.

Aos amigos que nos deram expectativa de como ficaria nosso trabalho.

A todos os professores que nos ensinaram o que foi necessário saber-se na construção de nosso trabalho.

“Eu pensei: se eu conseguisse uma forma de assumir o controle desse jogo, poderia transformá-lo no que quisesse. O melhor de tudo é que se todas as minhas ideias falhassem, ninguém se importaria – ele já era um fracasso”

Hedetaka Miyazaki (宮崎英高) – 2009

RESUMO

Palavras-chave: jogos, jogos digitais, informática

ABSTRACT

Keywords: games, digital games, informatic

SUMÁRIO

1. INTRODUÇÃO.....	11
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1 UNITY 3D	17
2.2 BLENDER.....	19
2.3 PHOTOSHOP	21
2.4 PHOTOSCAPE.....	23
2.5 JAVASCRIPT	24
3. METODOLOGIA	26
4. DESENVOLVIMENTO	28
4.1 APRESENTAÇÃO E ANÁLISE DE RESULTADOS.....	30
5. CONCLUSÃO	45
5.1 PROBLEMAS E LIMITAÇÕES	46
6. REFERÊNCIAS.....	48

LISTA DE FIGURAS

IMAGEM 1: INTERFACE <i>UNITY3D</i>	18
IMAGEM 2: INTERFACE <i>MONO DEVELOP</i>	19
IMAGEM 3: INTERFACE DO <i>BLENDER</i>	20
IMAGEM 4: INTERFACE DO <i>PHOTOSHOP</i>	22
IMAGEM 5: INTERFACE <i>PHOTOSCAPE</i>	23
IMAGEM 7: SAKURA 2D (ROSTO, CORPO E ROUPAS).....	31
IMAGEM 8: SAKURA 2D (EXPRESSÕES).....	31
IMAGEM 9: MARY BLOOD 2D (ROSTO, CABELO, CORPO E ROUPAS)	32
IMAGEM 10: MODELO PRÉ-PRONTO.....	33
IMAGEM 11: SAKURA (MODELO 3D)	33
IMAGEM 12: MARY BLOOD (MODELANDO)	34
IMAGEM 13: MARY BLOOD (MODELO 3D)	34
IMAGEM 14: HOTEL 01 2D	35
IMAGEM 15: KATANA 2D.....	35
IMAGEM 16: <i>PEACE BOW</i> E FLECHA 2D	36
IMAGEM 17: CENA 2D.....	36
IMAGEM 18: <i>PEACE BOW</i>	37
IMAGEM 19: FLECHA.....	38
IMAGEM 20: KATANA	38
IMAGEM 21: HOTEL	39
IMAGEM 22: MAPA BIDIMENSIONAL ORIGINAL.....	40
IMAGEM 23: MAPA 1 TRIDIMENSIONAL (VISÃO PLANIFICADA)	41
IMAGEM 24: MAPA 1 TRIDIMENSIONAL (VISTA DO AR).....	41
IMAGEM 25: MAPA 1 TRIDIMENSIONAL (MONTANHAS)	42
IMAGEM 26: MAPA 1 TRIDIMENSIONAL (MAR).....	42
IMAGEM 27: MAPA 1 TRIDIMENSIONAL (HORIZONTE).....	43
IMAGEM 28: MAPA 1 TRIDIMENSIONAL (ENTRADA DA VILA).....	43
IMAGEM 29: MAPA 1 TRIDIMENSIONAL (FONTE)	44

1. INTRODUÇÃO

É uma convenção mundial que jogos digitais são hoje a maior forma de descontração entre os jovens, seja em computadores *desktop*, *notebooks* ou celulares, *single* ou *multiplayer*, os jogos digitais ganharam a mente e os computadores de muitos usuários. Jogos digitais constituem-se em um conjunto de arte e ciência, vemos a arte na história, nos personagens, nas animações, e nos mesmo lugares vemos a ciência, ao dar vida a todos eles, ao torna-los reais por meio de códigos.

Este trabalho traz um exemplo de como pode ser construído um objeto virtual que é capaz de prender pessoas a uma tela, traz os elementos necessários para um jogo e como podemos montá-los. Construir um jogo sem nenhuma experiência anterior é uma trabalho desafiador, mas que mostra competência em termina-lo e apresentá-lo.

Segundo Schuytema, jogos são atividades, com regras, que servem para descontrair e divertir determinada pessoa ou determinado grupo, desenvolvido por decisões e ações dos jogadores que resultam em uma condição final que será melhor para o resultado do grupo ou da pessoa individual. Jogo digital é algo que contém todas essas especificações, mas pode ser demonstrado virtualmente, em vários tipos de computadores ou em apenas um específico. Um jogo que funcionará apenas em ambiente virtual.

Gilliard Lopes nos diz que existem várias classificações de jogos virtuais, desde o seu tipo de história ao tipo de plataforma, entre outras coisas, para criar uma identidade entre jogos semelhantes. Uma forma de classificar é por dimensionalidade, se é bidimensional ou tridimensional. Do ponto de vista, quando o jogo é em primeira pessoa, o jogador tem a mesma visão do personagem, ou terceira pessoa, o jogador tem uma visão diferente do personagem. Existem jogos que disponibilizam as duas possibilidades, como os de corrida. Por número de jogadores, podendo ser *singleplayer*, onde apenas um jogador participa, ou *multiplayer*, onde é formada uma equipe para o jogo.

É também possível classificar jogos também por seu gênero, que pode ser ação, aventura, estratégia, RPG (*Role-Playing Game*), esporte, simulação, tabuleiro, quebra-cabeça, entre muitos outros. Alguns dos gêneros apresentados tem subcategorias, como o de ação, que pode ser de tiro ou de plataforma, com o objetivo de matar os adversários e conseguir chegar ao final do mapa, respectivamente, de qualquer forma, necessita-se de certa habilidade com o controle.

Os jogos de aventura não exigem tanta habilidade, mas precisa de estratégias pensadas e geralmente cobra muita exploração do jogador.

Jogos de estratégias, com já dito no próprio nome, buscam que o jogador monte uma estratégia para dominar certo desafio, que pode ser um desafio geral para todo o jogo ou vários pequenos desafios que serão substituídos à medida que forem cumpridos. Os RPG são jogos que o jogador precisa viver a vida do personagem, que geralmente tem uma certa linha temporal que precisa ser completa com o jogador, durante esse período da vida do personagem pode existir várias pequenas *quests*, que servem de base para o objetivo principal do jogo, esse gênero de jogo necessita ser muito rico em história para poder ser atrativo aos jogadores. Existe o estilo ocidental e o oriental do mesmo, onde um foca no desenvolvimento do personagem e história enquanto o outro tem um leve enfoque em diálogos, respectivamente.

Os jogos de esportes pretendem simular uma partida de determinado esporte real, mantendo as regras e algumas vezes, praticantes reais desse esporte no mesmo. Jogos de simulação buscam simular, o mais próximo possível da realidade, alguma situação real, principalmente em manuseamento de sistemas complexos, os mais famosos são os simuladores de pilotagem de avião, alguns são usados profissionalmente para pilotos em treinamento. Jogos de tabuleiro são os *boardgames* virtuais, representam jogos tradicionais de tabuleiro, mas representados em máquinas. Jogos de quebra-cabeça são geralmente os educativos, o jogador precisa usar de um pouco de lógica para resolver um ou mais *puzzles*.

É perceptível que o mercado mundial da área só faz crescer desde que foi criado, em 2016 foram gerados cerca de US\$ 99,6 bilhões de lucro com esse mercado, diz a *Newzoo*. A maior parte desse dinheiro provem de jogos para computadores *desktop*. O maior produtor e consumidor desse tipo de produto e a Ásia, em especial o China como consumidor e o Japão como produtor, que deve apresentar US\$ 24,3 bilhões para o setor até o fim de 2016. No Brasil, o mercado de *games* também tem um valor considerável, o país está em décima primeira na colocação mundial e primeira na América Latina, estimando uma receita de US\$ 1,27 bilhões até o fim de 2016.

De acordo com a *Newzoo* numa pesquisa feita em 2015, dos 33,6 milhões de usuários brasileiros, 56% investem dinheiro em jogos. Este ano, mundo dos *games* no Brasil já alcança um público com cerca de 59,2 milhões de brasileiros, onde a maioria consome jogos por smartphone, mas o maior gasto da população referente a jogos digitais, continua sendo nos computadores de mesa ou *notebooks*. Nessas condições, as grandes empresas de *games* estão

buscando novos atrativos para o mercado, investindo em eventos que agradem o público *gamer*, o que conta com a presença de outras áreas interessam ao mesmo público, como quadrinhos, filmes e séries, mas sem perder o foco dos *games*, trazendo várias novidades e em alguns casos, competições.

Segundo a Saga, o Brasil apresenta apenas 3% do desenvolvimento geral no mundo dos *games*, com a maioria dos seus jogos sendo *Advergames* – jogos feitos para divulgação – ou jogos educativos. Embora o mercado ainda não tenha tanta atenção e investimento quanto precisa, é um setor que promete ascender, pois o necessário para expansão e venda dos produtos deste nicho deixou de ser interdito por condições físicas e passou a ser atingido por meio digital. Quando precisaríamos de uma fita ou um CD de 5 a 10 anos atrás, hoje é somente fazer um download de um banco de dados depois de ter feito o pagamento por cartão de crédito diretamente no site, isso quando os jogos não são baixados gratuitamente em *smartphones* ou computadores de mesa ou não.

O perfil do *gamer* brasileiro é traçado a cada ano pela Pesquisa Game Brasil em parcerias com empresas do ramo de *games* e de pesquisas de mercado. Começou o seu trabalho em 2013, parou em 2014, mas voltou em 2015 e também fez no ano seguinte. O modelo consumidor vem variado um tanto de 2013 até os dias atuais. O público *gamer* feminino aumentou de 41% para 52,6% de 2013 para 2016. Assim como a plataforma dos jogos também. O computador que levava o primeiro lugar em 2015 com 85% de preferência deu lugar ao *smartphone*, que hoje tem 82,7% de usuários. Mesmo que os jogos digitais estejam presente na vida de vários brasileiros, todos os dias e tenha um mercado tão grande, estimasse que apenas 11% dos cidadãos online se consideram reais *gamers*.

A mesma pesquisa também faz referência quanto ao tipo de jogos preferidos, eles variam de plataforma a plataforma e em alguns casos, de sexo para sexo. Em dispositivos móveis, por exemplo, os usuários preferem jogos de estratégia, enquanto em vídeo *games* de console, existe uma preferência nos jogos de esportes. Mas no geral, os três tipos de jogos preferidos são de estratégia, aventura e ação, que se fazem presente em posições destacadas de ambos os sexos. Outro ponto a se observar no perfil dos jogadores é o período e local onde jogam, o momento de maior acesso aos *games* entre os usuários é quando estão em deslocamento no trânsito, seja em ônibus, metrô ou carro. Em segundo lugar vem o trabalho, cerca de 74,5% dos jogadores tem contato com jogos no horário de trabalho.

De acordo com Marcelo Teixeira do *TechinBrazil*, em 2014 existiam cerca de 130 empresas no ramo, produzindo *games* para as mais variadas plataformas, mas com foco em jogos para *desktops* ou navegadores de internet e plataformas de *smartphones* e *tablets*. Para promover essa indústria e atrair mais pessoas interessadas, a prefeitura de São Paulo, pelo SPCultura em conjunto com o ABragames (Associação Brasileira dos Desenvolvedores de Jogos Digitais) resolveram criar o SPCine, um edital que patrocina a produção de 16 *games* brasileiros com 35 mil para pessoa física e 150 mil para empresas do Ramo. O edital tem como objetivo principal o incentivo na área e dar oportunidade aos iniciantes na área.

Rafael Dias mostra, a partir do Produção de Jogos, a construção de um jogo digital. Que se dá inicialmente por meio da imaginação, alguém deve ter a base de como será o jogo, sua história – se exigir história –, os personagens, e a partir daí o jogo poderá ser construído. Dependendo da plataforma escolhida, teremos gráficos e qualidades diferente para cada *game*, mas para todos, existe um processo que passa por várias áreas, que podem ser controladas por vários profissionais ou um só. Começa-se essas fases com o roteiro do jogos, onde estará toda a história que será mostrada a o jogador, depois vem os protótipos de personagens e cenário, que serão digitalizados logo após. Depois disse vem as animações e, se for de escolha do produtor, os sons e músicas dos jogos. De modo geral, o desenvolvimento de um jogo se divide em pré-produção e produção, onde existe o planejamento e a produção em si.

Em empresas profissionais da área, existe um profissional para cada fase de desenvolvimento do *game*, dentre eles o *game designer*, que acompanha o jogo do início ao fim, o que faz precisar dividir a função para que não haja sobrecarga em cima de apenas um funcionário, normalmente encontra-se quatro. O *designer*-chefe que monitora todos os outros, o *designer* de sistema, que elabora as regras e funcionamento do jogo, o *designer* de fases, que tem a função de criar fases e missões, e o redator, que cuida da parte não jogável, como cenas e diálogos dos personagens. Ambos devem manter uma comunicação ativa entre si e entre os produtores do jogo, se houver falta de comunicação, pode existir falhas no jogo ou o mesmo não sair como o esperado.

Outros importantes profissionais que estão presente na construção de jogos são os roteiristas, que assim como o nome diz, cuidam do roteiro do jogo, sua linha cronológica ele tem que trabalhar sempre para criar algo que envolva o *gamer*. O produtor, que existe tanto como um supervisor quando como um programador, ele tem que trabalhar no projeto a medida que assegura que o *game* cumpra com seus prazos e orçamento. Temos também o *designer* de

áudio, que compõe a trilha sonora e os sons que farão parte e darão vida ao *game*. Depois de todo o jogo estar funcionando, teoricamente, depois do primeiro protótipo estar completo, faz-se necessário na equipe um testador de jogos, que vai jogá-lo do início ao fim, mais de uma vez, a procura de *bugs*.

O escritor do Produção de Jogos também fala que no Brasil também temos um grande número de jogos *indie*, que são produzidos com independência de apoio de empresas, apenas um grupo de desenvolvedores independentes que fazem tudo existente no jogo. Encontra-se muitos desses jogos em dispositivos móveis, aplicativos para *Android*, *IOS* ou *Windows Phone*, mas também encontra-se muito em jogos para computadores *desktop* ou *notebooks*. Como os jogos são feitos por pessoas físicas, nem sempre se tem registros de quanto foi gasto ou ganho no projeto, é uma parte com resultado imprevisível. Tanto para a economia geral do mercado quanto para a equipe que constrói o jogo.

João Mattar nos mostra que existem também inúmeros *engines* que podem ajudar na construção de um jogo digital, seja para modelar os personagens ou cenário ou para animar e dar vida aos jogos. Uma das mais usadas é o *Unity*, que é um motor gráfico ou *game engine* desenvolvida pela empresa de mesmo nome a fim disponibilizar uma ferramenta mais simples e acessível para os desenvolvedores independentes, iniciantes e avançados. Existe também o *Blender*, desenvolvido pela Blender Foundation em 2002, este é um software livre de modelagem tridimensional sem fins lucrativos.

Também existem ferramentas para jogos bidimensionais, tais como o *Construct2*, desenvolvido pela *Scirra*, é muito bom para iniciantes por ser uma ferramenta que permite criar um jogo sem nenhuma linha de código feita pelo desenvolvedor, ele usa *HTML5* como base para seus programas e é uma ferramenta muito fácil de se usar. Existe também a *Stencyl*, que desenvolve para *Android*, além de computadores. Assim como o citado anterior, a *Stencyl* também permite criar jogos simples sem a necessidade de códigos, é uma ferramenta intuitiva, que permite o usuário criar jogos sem nenhum conceito de programação, embora também busque explorar a lógica do mesmo.

Além das ferramentas de modelagem do jogo em si, existem ferramentas não pertencentes à área que podem ajudar no desenvolvimento de um jogo digital, seja para fazer as texturas usadas, seja para construir algumas bases para a modelagem tridimensional ou para montar uma base de como será aspectos do jogo. Pode-se dizer que são essenciais para tal construção. Uma das ferramentas muito usadas é o *Photoshop*, criado pela *Adobe* e considerado o melhor

aplicativo de *design* do mundo, pode ser utilizada para construir texturas e planificações. Outras ferramentas podem ser usadas para o mesmo objetivo, como o *Core Draw* e o *Photoscape*.

Ao terminar o trabalho, precisa-se mostrar a construção de um jogo RPG tridimensional de primeira pessoa, buscando entretenimento e colaboração para a comunidade artística e científica do gênero, visando buscar sempre o melhor desempenho do programa em um computador que atenda suas especificações. Pretende-se montar o jogo por partes e mostrando os resultados dessas partes, e como os conhecimentos adquiridos ao longo do curso ajudaram para o alcance do objetivo.

Desejou-se fazer partes distintas do jogo, que se encontrariam ao término do trabalho, montar o personagem, o cenário e fazer a cena, e todas as cenas se juntariam ao final, formando o projeto completo, pronto para ser jogado. O trabalho será feito prioritariamente no *Unity*, tendo várias outras ferramentas para auxiliar nas funções necessárias para a finalização do projeto, tais como *Blender*, *Photoshop* e *Photoscape*.

É de grande importância que seja mostrado a competência para concluir o trabalho, junto com o conhecimento adquirido durante o mesmo. Contribuiu-se para o crescimento cultural e científico do Instituto e de seus discentes, disponibilizando o jogo e o documento, se necessário, aos próximos alunos que precisarem para fins acadêmicos. Compreende-se também a realização pessoal dos realizadores do trabalho ao finalizar o projeto.

2. FUNDAMENTAÇÃO TEÓRICA

As ferramentas usadas para ajudar na construção deste *game* em sua maioria foram ferramentas de construção e manutenção de objetos tridimensionais. Um programa de modelagem tridimensional é uma aplicação que permite ao usuário criar desenhos em 3 dimensões, ou esculturas, como também é conhecido. E isto se dá basicamente pelo fato de o programa ser capaz de gerar formatos geométricos de duas dimensões em três eixos: X, Y e Z, de tal forma que juntos formam imagens geométricas de aparentes três dimensões. Quando estes modelos estão prontos existem basicamente duas formas de usa-los: renderização em tempo real (real time CG) ou pré-renderização (CGI).

CG é o termo universal utilizado para designar uma aplicação tridimensional e traduzida significa gráficos de computador (*Computer Graphics*). Sendo a primeira usada em jogos eletrônicos pois este necessita de uma interação em tempo real com o usuário, no entanto a pré-renderização, como o nome já diz, é renderizada antes do resultado final e por isso permite uma qualidade maior em sua execução e detalhamento pois não necessita de urgência como a executada em tempo real.

2.1 UNITY 3D

Uma das ferramentas usadas foi o *Unity*, usado como motor gráfico para desenvolver o *script* do jogo e também como base de modelagem de uma parte do cenário usado no programa. Esta ferramenta está disponível em duas versões, a PRO que é paga e dispõe mais configurações e melhorias, e a versão básica que foi usada para desenvolver este trabalho, que é grátis e conta com opções reduzidas, porém, nada impede o usuário de desenvolver métodos para alcançar os mesmo resultados da versão completa.

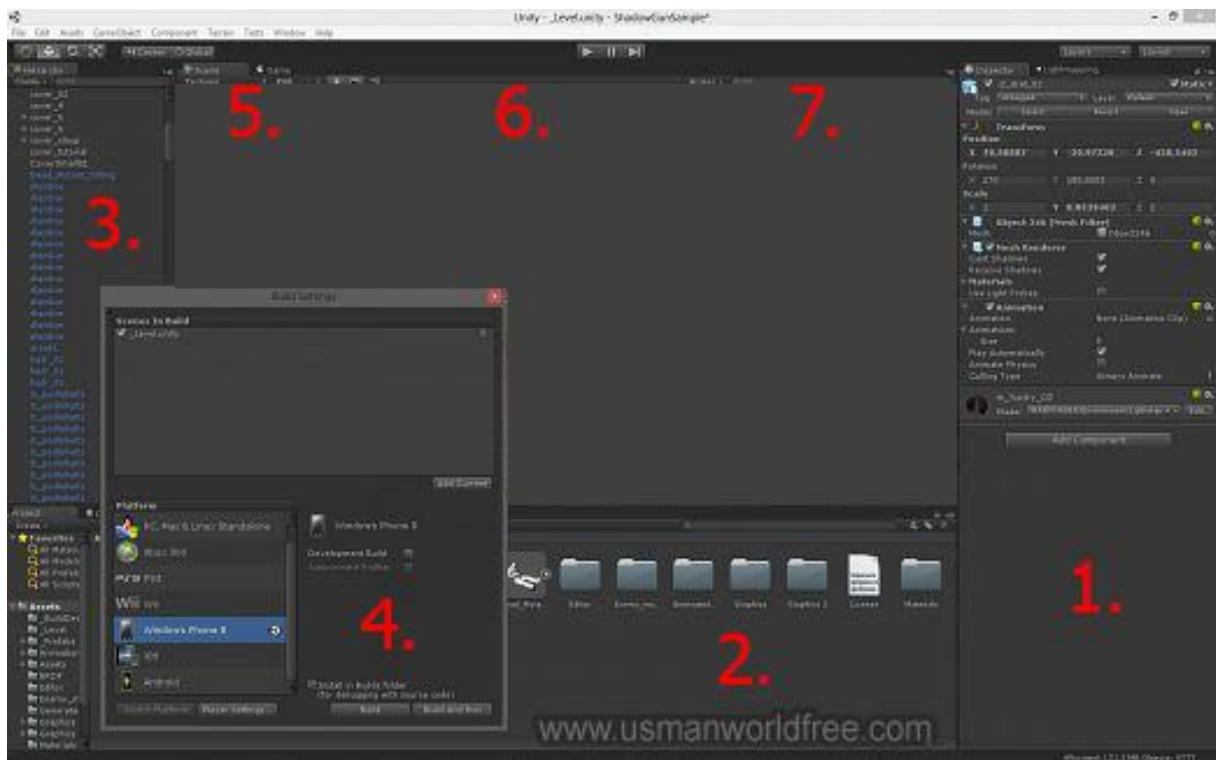
Esta ferramenta já é uma das mais populares entre empresas e desenvolvedores independentes, atualmente cerca de 700 milhões de jogadores utilizam jogos desenvolvidos neste software segundo o próprio site da empresa. Ela tem suporte as mais variadas e utilizadas plataformas da atualidade, como por exemplo, *Android*, *Windows*, *Linux* e todos os consoles de vídeo game atualmente no mercado. Além de permitir que usuários iniciantes tenham acesso a ferramentas que empresas como *Coca-Cola* e *Ubisoft* usam, ainda traz suporta a novas tecnologias como o VR e AR (Realidade virtual e Realidade aumentada, respectivamente).

Este motor gráfico dispõe também de um editor de texto onde é possível desenvolver o código fonte que será utilizado no projeto, chamado de *Mono Develop*, este *software*, que

também pode ser adquirido separadamente, conta com ferramentas que ajudam no desenvolvimento do código, tais como, dicas de comandos, diferenciação de palavras por cores, que ajudam a diferenciar comandos de variáveis ou seja qual for a ocasião. Também permite salvar, editar arquivos de várias extensões suportadas pelo motor. Este software ainda é adaptado para o *Unity* de forma a facilitar o desenvolvimento.

Uma característica forte desta adaptação se dá pelo fato de existirem duas áreas de código, sendo uma designada para executar o comando unicamente no momento em que o jogo é iniciado (carregar o mapa é um exemplo), a outra executa o código continuamente ao decorrer da execução da aplicação, de tal forma a permitir, por exemplo, que o jogador possa mover o personagem a qualquer momento da execução.

Imagem 1: Interface *Unity3d*



1 – Inspetor: Onde ficam todas as características, funções, opções, scripts entre outras coisas que sejam valores atribuídos a o objeto selecionado.

2 – Explorador: Onde é possível ver, editar e mover todos os arquivos presentes no projeto.

3 - *Assets* do cenário: Como o nome diz, todos os *assets* sejam eles objetos, personagens, terrenos etc. ficam presentes nesta área para que sejam selecionados.

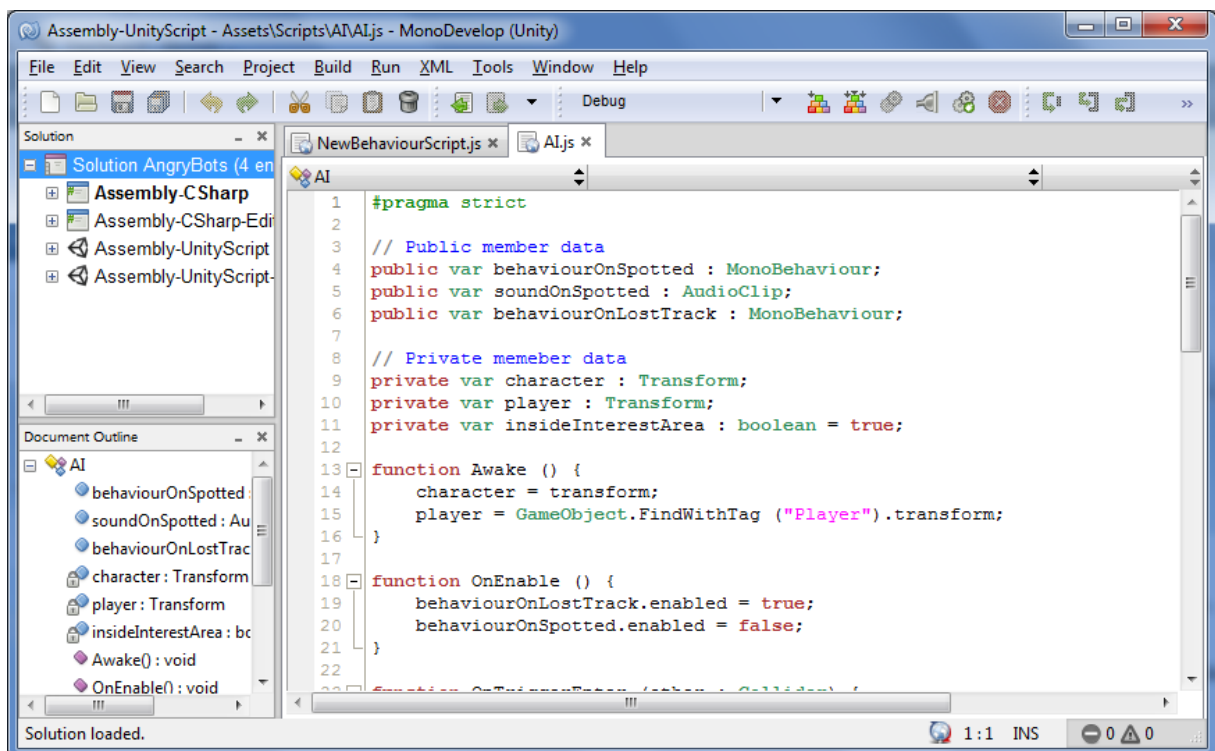
4 - Nesta janela é possível escolher previamente qual a plataforma de destino do projeto, fazendo com que, de acordo com a seleção, todo o programa seja adaptado para a plataforma e os códigos gerados de forma automática também sejam.

5 - Aba cenário: Esta área permite a alteração do cenário, como por exemplo, mover objetos, adicionar, ter uma visualização prévia, coisas deste tipo.

6 - Aba *game*: Quando o jogo é executado a fim de testar ou qualquer motivo, é nesta aba que ele é executado e mostrado da mesma forma que será apresentado ao usuário final.

7 - Aba animação: Onde é possível criar arquivos de animação, importa-los de programas externos, criar uma hierarquia e fluxo que denominam a sequência das animações.

Imagem 2: Interface *Mono Develop*



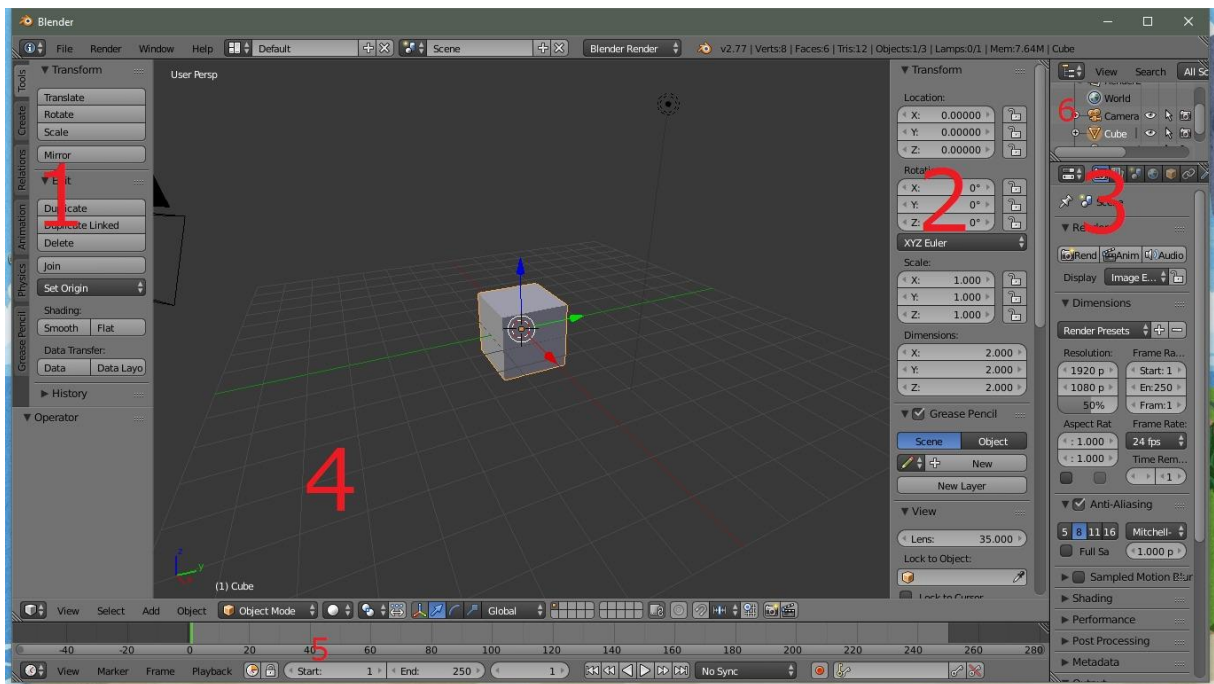
2.2 BLENDER

Outro programa usado para o desenvolvimento do jogo foi o *Blender*. De interface relativamente simples, este programa permite ao usuário desenvolver modelagens a nível profissional, atingindo uma qualidade igualável a de programas pagos usados por empresas. Com uma ferramenta versátil que permite ao desenvolvedor construir os modelos a partir de vértices, linhas ou faces, usando ou não imagens bidimensionais como base a criação.

A modelagem neste programa se baseia na extrusão, movimentação e recorte dos vértices, permitindo assim, criar desde objetos extremamente simples até objetos de uma complexidade elevada de forma que o único limite para tal é a potência da máquina usada. Além de modelagem, outra utilidade do programa se dá no fato de poder-se animar os modelos e inserir as texturas, permitindo assim, que os mesmo já sejam levados ao motor gráfico de desenvolvimento já pronto, precisando apenas de um código que ative suas animações conforme a necessidade e os comandos desejados.

O *Blender* pode estender-se até onde for desejado pelo usuário desde que ele saiba programar e tenha noção de como funciona o programa, pois o mesmo é um programa de *software* livre e isso abre ainda mais as possibilidades no desenvolvimento utilizando esta plataforma.

Imagem 3: Interface do *Blender*



1 - *Tool Shelf*: É onde ficam as ferramentas de edição, que permitem, cortar, extruir, alterar a escala e funções do tipo na modelagem do objeto.

2 - *Propriedades do Programa*: Nessa aba é possível alterar algumas opções de como será feita a modelagem. Por exemplo, travar a visão em algum objeto específico, adicionar imagens bidimensionais para ajudar na modelagem, entre outras.

3 - Propriedades do Objeto: É onde pode ser definido algum tipo de parentesco entre objetos, adicionar modificadores e alterar materiais por exemplo.

4 - Visualização 3d (*3d view*): Essa tela permite a visualização do objeto modelado, não somente durante sua criação, mas também em sua edição. Pode alternar entre modos como: objeto, edição, animação, textura e ainda outros.

5 - Linha do Tempo: Nessa área são adicionados os *keyframes*, que quando executados em sequência, darão vida animação. Que por sua vez é feita, basicamente, pela alteração do modelo seguida da inserção de um *keyframe* que salvará a alteração naquele momento específico, assim dando continuidade ao movimento.

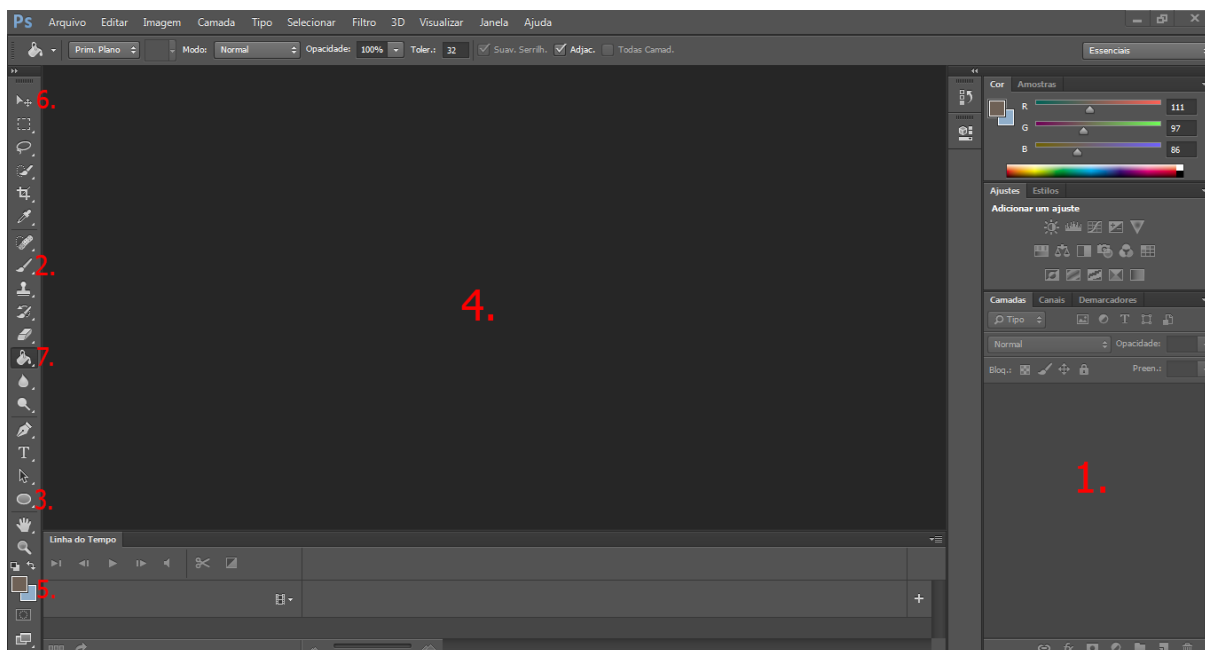
6 - *Outliner*: Funciona como um registro dos objetos presentes no ambiente, permitindo que se possa alternar entre eles, deixá-los visíveis ou invisíveis e alterar seu parentesco, por exemplo.

2.3 PHOTOSHOP

Saindo do cenário tridimensional, outra ferramenta essencial no desenvolvimento do *game* foi o *Photoshop*, criado pelos irmãos Thomas e John Knoll, a ferramenta de manipulação de imagens foi originalmente chamada de *ImaginePro*, cerca de um ano depois, foi apresentado ao *Adobe* e à *Apple*, então mudou para *Photoshop*, lançado exclusivamente para *Macintosh*, a partir daí, cresceu até se tornar a melhor ferramenta do ramo.

O programa trabalha com o sistema de *layers*, onde existem várias camadas com imagens ou configurações diferentes que tentem a se juntar a formar a imagem final. Nesse modo, não altera-se a imagem ao todo, mas sim uma camada, alterando essa camada, altera-se o resultado final. O número de camadas que pode ser usado depende apenas da capacidade do computador, podendo atingir um número infinito. Como não existe um formato popular que aceite saltar o documento em camadas, o *Photoshop* o fez, guardando o documento no formato *.psd* (*PhotoShop Document*).

Usado principalmente para fazer texturas e a versão bidimensional da ilha na primeira versão, que foi usada como base da ilha construída no Unity. Foram feitas todas as texturas da Sakura e da Mary Blood no programa, corpo, cabelo e roupas. Como também todas as texturas utilizadas na ilha. Para ambas foi usada basicamente a ferramenta pincel e a criação de formas geométrica, junto com a paleta de cores para escolher a cor de determinada característica.

Imagem 4: Interface do *Photoshop*

1 – *Layer*: Onde ficam as camadas apresentadas, podendo-se organizá-las, modificá-las e agrupá-las;

2 – Ferramenta Pincel: Como o nome já diz, é onde localiza-se o pincel, ferramenta que serve para dar cor ou forma a pequenos detalhes, podendo-se escolher o formato, tamanho e dureza;

3 – Formas Geométricas: Localiza-se todas as formas geométricas planas e algumas formas especiais, que podem ser usadas para várias funções;

4 – Área de trabalho: Onde mostra-se a soma das *layers*, o resultado final e onde pode alterar as mesmas

5 – Seletor de Cores: Essa ferramenta nos dá uma paleta de cores para escolher a que queremos usar. Tem cor de primeiro e segundo plano, que podem se alterar. Existem as duas opções para casos onde precisa-se fazer um *dégradé*, por exemplo, o programa já oferece a função e faz-se apenas escolher as cores que serão usadas para o mesmo;

6 – Ferramenta Mover: O nome já é auto explicativo, com essa ferramenta, você pode mover a camada;

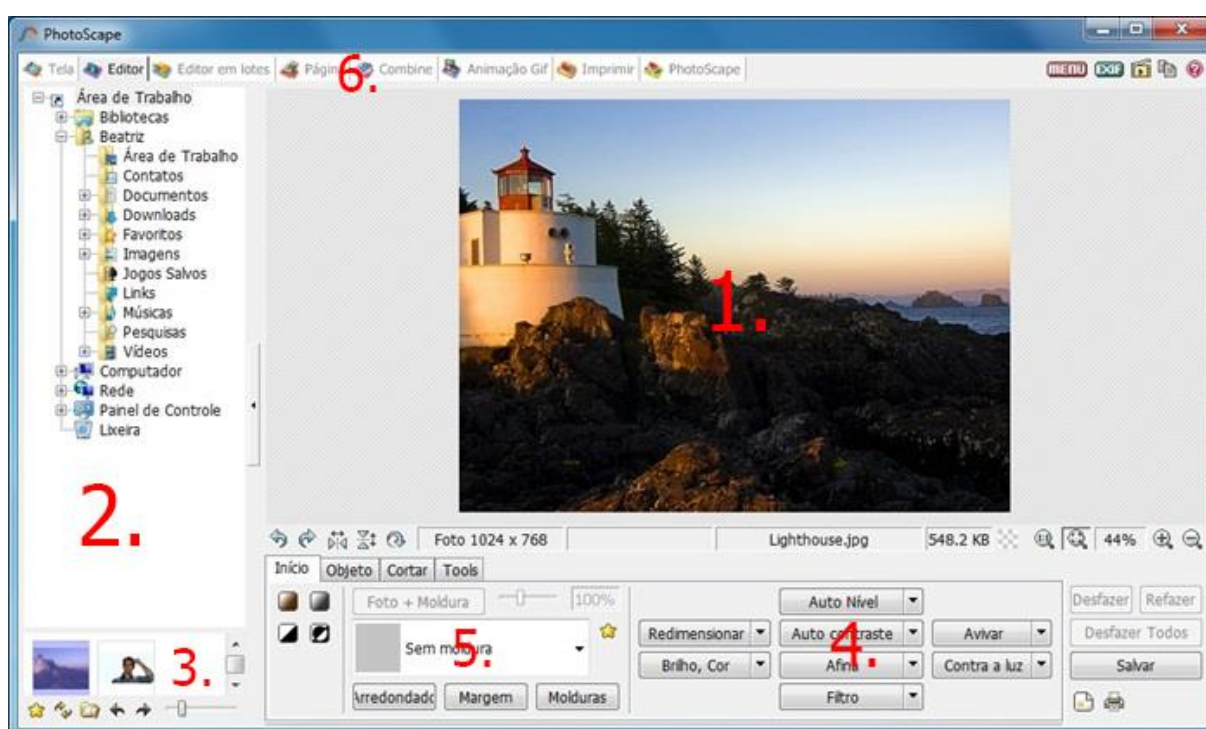
7 – Balde de Tinta: Com essa ferramenta, pode-se colorir toda uma área da cor selecionada, também pode-se alterar a mesma para ferramenta *dégradé*.

2.4 PHOTOSCAPE

O *Photoscape* também foi utilizado nesse projeto, só que numa frequência muito menor que todos os outros, é disponibilizado gratuitamente pela *Mooii Tech*, tal organização oferece o *software* e o mantém em funcionamento por meio de doações. O mesmo é um editor de fotos que permite trabalhar com fotos individuais ou em lotes, um conjunto de fotos, o que facilita o trabalho caso se queira fazer a mesma configuração para mais de uma foto.

Ele foi usado somente para alterar a resolução das imagens para poder adequá-las depois ao que o usuário puder suportar no computador. As resoluções variam de 440 a 720, permitindo vários tipos de qualidade para adequar a melhor ao computador onde está sendo processado.

Imagem 5: Interface *Photoscape*



- 1 – Área de trabalho: Onde se localiza a foto a ser editada;
- 2 – Navegador: Local onde pode-se encontrar todas as pastas do computador;
- 3 – Visualizador de miniaturas: Pode-se ver a miniatura de todas as fotos localizadas na pasta selecionada, onde escolhe-se a foto que será mostrada na área de trabalho e editada;

4 – Opções de Configurações: Onde localiza-se todas as opções que podem ser escolhidas para fazer alteração na imagem. Algumas delas são redimensionar, que altera o tamanho da imagem, brilho e cor, dando ou tirando brilho da imagem ou alterando seu enfoque para determinada cor, avivar, que dá um toque maior de brilho e definição a imagem e filtro, podendo-se escolher algum efeito que fará parte da mesma;

5 – Moldura: O programa já vem com algumas molduras predefinidas que podem ser escolhidas pelo usuário a fim de dar uma beleza a mais na imagem;

6 – Abas de edições: Abas que contém janelas, cada uma tem um método de edição diferente, inclusive a edição em lotes.

2.5 JAVASCRIPT

O *JavaScript* foi usado como linguagem principal na codificação do jogo. A linguagem teve início quando a *Netscape* criou um sistema de linguagem para *web* usando o *server-side* (lado do servidor). A *Microsoft* então fez sua própria versão do sistema, criando o *JScript*, dando a chance da *Netscape* desenvolver um sistema *cliente-side* (lado do cliente), que executa o *script* no computador no usuário, que foi o *JavaScript*.

Embora tenha surgido com o nome de *Mocha*, depois se transformado em *LiveScript* a linguagem recebeu o nome de *JavaScript*, por uma jogada de *marketing*, pois a empresa fazia uso de programas com a linguagem *Java*. O que fez o *JavaScript* ganhar popularidade.

O *Unity*, ferramenta onde se é usada a linguagem, existe também a opção de programação em C#, mas foi escolhida a linguagem apresentada tanto por ser mais simples de se trabalhar com quanto de aprender, como também por um dos componentes já ter conhecimento relativamente avançado da linguagem, tornando-se mais fácil de ser usada.

Imagem 6: JavaScript

```
49     m_Camera = Camera.main;
50     m_OriginalCameraPosition = m_Camera.transform.localPosition;
51     m_FovKick.Setup(m_Camera);
52     m_HeadBob.Setup(m_Camera, m_StepInterval);
53     m_StepCycle = 0f;
54     m_NextStep = m_StepCycle/2f;
55     m_Jumping = false;
56     m_AudioSource = GetComponent();
57     m_MouseLook.Init(transform , m_Camera.transform);
58 }
59
60
61 // Update is called once per frame
62 private void Update()
63 {
64     RotateView();
65     // the jump state needs to read here to make sure it is not missed
66     if (!m_Jump)
67     {
68         m_Jump = CrossPlatformInputManager.GetButtonDown("Jump");
69     }
70
71     if (!m_PreviouslyGrounded && m_CharacterController.isGrounded)
72     {
73         StartCoroutine(m_JumpBob.DoBobCycle());
74         PlayLandingSound();
75         m_MoveDir.y = 0f;
76         m_Jumping = false;
77     }
78     if (!m_CharacterController.isGrounded && !m_Jumping && m_PreviouslyGrounded)
79     {
80         m_MoveDir.y = 0f;
81     }
82
83     m_PreviouslyGrounded = m_CharacterController.isGrounded;
84 }
85
```

3. METODOLOGIA

O trabalho foi feito com poucos encontros para reunião entre os participantes, com foco completo na construção do projeto e documentação sobre o mesmo, a comunicação baseou-se em pedir algo necessário para a construção de determinada configuração do jogo apresentado. O maior foco, até o momento foi na modelagem tridimensional dos personagens e cenário, com suas respectivas texturas.

A primeira coisa a ser feita foi a estipulação dos objetivos, o que se espera do jogo, com a montagem do roteiro de jogo, quais vão ser os níveis, os personagens, como eles vão interagir, como se dará a história, toda a linha temporal que será seguida pelo jogador. Também, fez-se um modelo bidimensional de tudo que será usado para compor o cenário e de todos os personagens planejados até o momento. Depois disso, começou-se a digitalização dos personagens principais – Mary Blood e Sakura –, modelos tridimensionais, assim como suas texturas, que logo foram aplicadas.

Logo após isso, foi começado a modelagem do espaço onde iria se passar a maior parte do jogo, a primeira ilha de Plume, a ilha da natureza, que foi modelada em uma ferramenta diferente da anterior, assim como sua textura também. Depois disso, começou-se a animar a primeira personagem, Mary Blood, para coloca-la funcionando no jogo. Quando pronto, será feito o mesmo com a próxima personagem, Sakura. Enquanto isso, também estava sendo feito modelagem e animação de alguns objetos que serão usados no jogo, armas, acessórios, objetos necessários para a conclusão do jogo.

Depois das animações concluídas, elas serão colocadas no cenário do jogo, onde são manipulado os códigos que farão as animações funcionarem como modo de jogo, apenas quando o *player* der o comando. Quando pronto, modelou-se um inimigo compositor de uma parte jogável do projeto. Depois de modelado, o mesmo foi animado e posto no cenário, então a primeira cena jogável montou-se. Uma luta entre a primeira personagem principal animada, que foi a Mary Blood, e o inimigo criado. Pretende-se fazer depois todos os personagens, animá-los, modelar a segunda ilha de Plume, a ilha da civilização, que será tão importante para a história do jogo quanto a primeira, por todos os personagens e programar tudo para seguir o roteiro.

A ferramenta usada para construir os modelos tridimensionais dos personagens e dos objetos foi o *Blender*, programa feio especificamente para modelagem tridimensional. Foi

usado o modo de modelagem por cubo onde há a distorção de cubos pré-definidos e disponibilizados pelo programa, dando os formatos necessários, seja para as personagens ou para os objetos. São alterados as linhas, faces e pontos do cubo com os comandos *extrude*, *space* e *rotation*.

As ferramentas usadas para as texturas de todos os componentes tridimensionais são o *Photoshop* e o *Photoscape*, ambos ferramentas de design e edição de imagens. O *Photoshop* usado para fazer a textura em si, a imagem que servirá de base para inserir no personagem ou objeto, enquanto o *Photoscape* é usado para alterar a resolução da imagem, por ser uma ferramenta mais fácil de usar para essa função. As resolução foram alteradas para que o jogador possa escolher qual é melhor para suas especificações de computador.

Já o cenário foi feito no *Unity*, que é um *engine* específico para jogos, com funções específicas que ajudam na montagem do mesmo. Ela foi usada na construção do jogo como um todo, pois nela foi adicionado todo o conteúdo já produzido a fim de tornar tudo em algo homogêneo e fluido. O *Unity* é utilizado para a criação do cenário, porém, como não é um ambiente desenvolvido para a modelagem tridimensional, maior parte dos objetos constituintes do mesmo são desenvolvidos em softwares paralelos, no nosso caso o *Blender*.

Essa ferramenta também é onde se dá vida ao jogo, pois é nela que desenvolve-se o código que será responsável pela lógica do jogo. As linguagens aceitas pelo programa são basicamente *JavaScript* e *C#*, sendo escolhida primordialmente o *JavaScript*, no entanto, nada impede de em algum ponto utilizar-se o *C#* por comodidade ou necessidade da ocasião.

O Plume contará uma parte da história de Sakura e Mary Blood, duas garotas de universos diferentes, mas que são a mesma pessoa. Elas se encontrarão em um espaço entre os universos chamado vácuo, nesse vácuo estarão as ilhas de Plume, onde vai ocorrer a maior parte do jogo. O Objetivo do mesmo é fazer com que as duas personagens percebam quem são, através das missões do jogo e diálogos que terão com os NPCs (*non-player character*).

4. DESENVOLVIMENTO

O jogo foi planejado para ter momentos variados, dividindo isso entre mapas que apresentarão enredo e jogabilidade diferentes. No momento da apresentação, apenas um desses mapas será apresentado, o mapa principal onde maior parte do jogo será desenvolvido, chamado de Plume, porém já será visível a forma de funcionamento destes ao todo, pois tratam-se apenas de variações de um mesmo modelo base já definido previamente.

As fases são divididas em duas partes, “introdução e continuidade”, sendo três fases: duas na introdução e uma na continuidade. Como cada personagem viverá em um mundo diferente antes de ser levado ao centro do jogo, ou seja, na introdução do mesmo, sendo assim, um mapa para cada personagem.

A primeira personagem, chamado de Sakura, viverá em uma cidade, este ambiente não possuirá liberdade de locomoção do jogador, possuindo um caminho predefinido no qual o levará a conclusão do mesmo. Como características principais este mapa possui:

- Caminho pré definido;
- Tamanho pequeno;
- Tempo de jogo curto;
- Ambiente noturno;
- Algumas interações relevantes para o jogador com objetos do cenário.

O segundo personagem, chamado de Mary Blood, encontra-se inicialmente em um ambiente ainda mais simplório, uma prisão de pedras da qual este irá escapar por meio de um portal. Como características relevantes este possui:

- Caminho pré definido;
- Visual simples e não muito detalhado;
- Ambiente fechado e escuro;
- Nenhuma interação importante com objetos do cenário.

O mapa principal é onde se desenrolará todo o jogo. Ele se trata de um tipo de ilha flutuante com pequenas ilhas ao seu redor e terá um reservatório de água que ocupa uma grande parte de seu território. Este mapa será dividido em quatro partes, sendo essas: *Mega Town*, que

é uma cidade capitalista baseada no nosso sistema atual de sociedade e economia. *Peace Village*, é uma civilização imponente e rival da primeira por ter um sistema que as diferem muito, baseado em uma filosofia naturalista. *Tri*, que é uma civilização antiga, porém que sofre influência moderna e por fim, a Floresta, que possui peculiaridades e personagens que a torna um ambiente tão influente quanto os outros três.

O jogo terá uma ambientação e enredo que será uma mistura de mundos de fantasia e *cyber punk*. Sendo uma característica forte desta mistura, um visual que junte natureza e máquinas, o contraste do verde com as cores fortes das luzes e a diferença relevante entra o dia e a noite.

No centro deste ambiente haverá uma grande construção que na qual todo enredo se desenrola sobre. É um lugar construído por ancestrais e que levará muito mistério e tempo de jogo ao usuário. Este mapa, por fim, apresenta como relevantes as seguintes características:

- Mundo aberto;
- Sistema *Sand Box* (caixa de areia);
- Tamanho expressivo;
- Tempo de jogo elevado;
- Muita interação com objetos e personagens;
- Nível de detalhamento elevado.

O objetivo do jogo é descobrir os segredos que aquele mundo guarda e encontrar respostas a questionamentos que são deixados durante o *gameplay*. Como os personagens não se sentem tão presos a sua velha vida, voltar ao mundo anterior não será um objetivo, mas sim descobrir uma existência nova, que é o grande segredo guardado pela velha construção.

Porém para encontrar a resposta do grande segredo eles precisam se tornar dignos de tal feito, e a dignidade se busca através do conhecimento e esse conhecimento é adquirido da própria natureza e das criaturas que vive nela. De certo modo, os personagens terão de entender como funciona o mundo e só ai eles poderão muda-lo de forma a conseguir chegar a seu destino. No caso, um dos personagens se tornará Deus, através do conhecimento. Esse será o objetivo. O outro personagem falhará. Ainda terá uma segunda alternativa, onde os dois personagens se tornam um só afim de alcançar o ideal que dividem.

4.1 APRESENTAÇÃO E ANÁLISE DE RESULTADOS

No começo do trabalho foi feito o enredo e desenhos de como seria o visual do jogo. Basicamente escreveu-se ideias aleatórias que vinham a mente e então depois eram discutidas a fim de encontrar um ponto que conectasse as ideias e deixasse um objetivo mais nítido a ser desenvolvido futuramente. Pesquisando na internet, tomando outros jogos, filmes ou qualquer forma de entretenimento servente de base ou inspiração para o desenvolvimento do trabalho, apresentamos ideias e discutimos até chegar a um consenso final do que seria produzido.

Após essa etapa, foi discutido o que estaria pronto para a entrega do trabalho, que seria aquilo com maior prioridade a ser trabalhado. Todo o enredo, arte, modelos e especificações técnicas de como funcionará o jogo foram registradas e guardadas para eventual consulta, alteração e entre outras coisas que venham a tornar necessária a existências de tais documentos.

Esta por sua vez, foi a parte menos dificultosa do desenvolvimento, pois, tratava-se apenas de colocar ideias e usar da criatividade para criar algo que nos agradasse, no entanto, sempre visando nossas limitações como desenvolvedores e nossa limitação no tempo estipulado até a entrega, para que não fosse criado algo que extrapolasse nossos limites e acabasse por não ser completo por falta de capacidade.

Mesmo com o jogo não concluído, os resultados alcançados foram satisfatórios diante do tempo disponibilizado e dos problemas apresentados ao o grupo. Como dito anteriormente, foi iniciado pela construção do roteiro detalhado do mesmo, que se seguiu apenas até a sétima cena, onde decidiu-se seguir para outra função para não perder muito tempo em apenas uma parte, embora a história completa e detalhada não esteja completa, a base dela está e pretende-se voltar para o roteiro o mais rápido possível.

Após isso foram desenhados no papel os modelos que serviriam de base para a primeira personagem principal idealizada, Sakura. Suas expressões e roupas tentam acompanhar sua personalidade, dócil demais que chega a ser passiva.

Imagem 7: Sakura 2D (rosto, corpo e roupas)

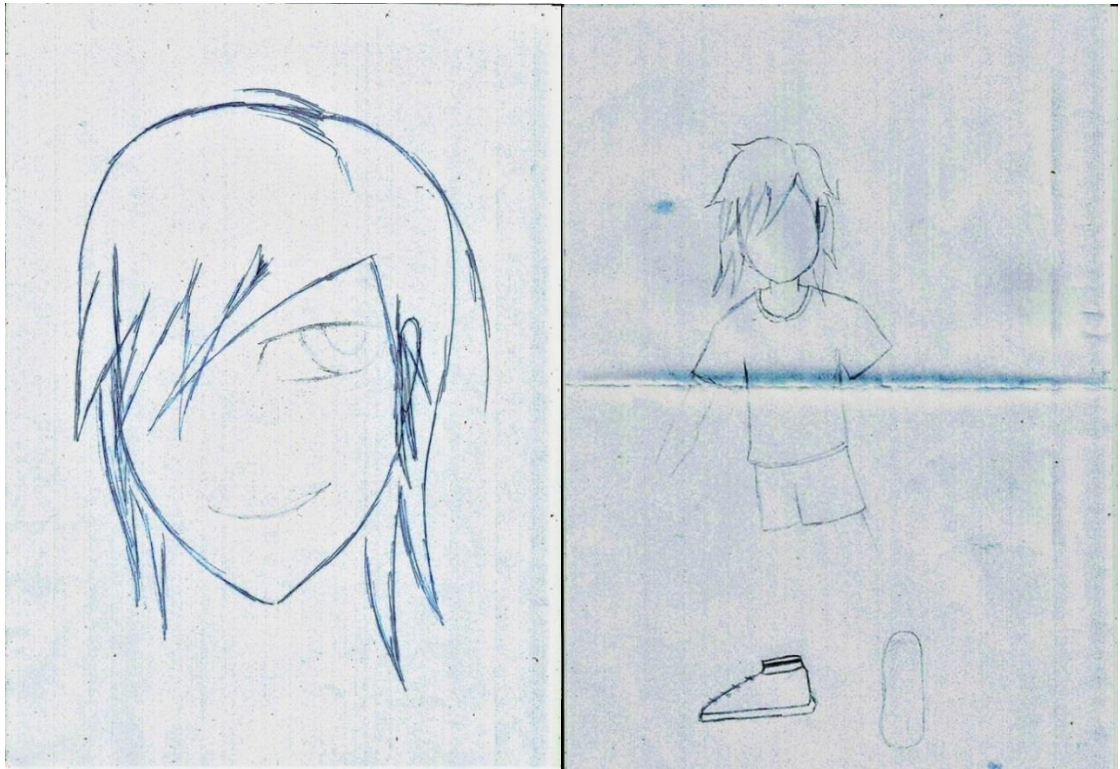
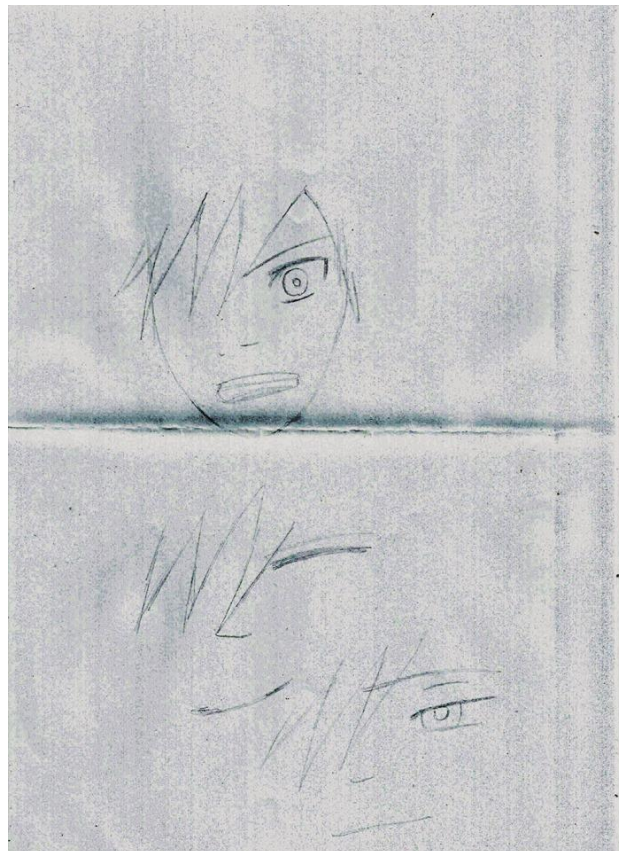


Imagem 8: Sakura 2D (Expressões)



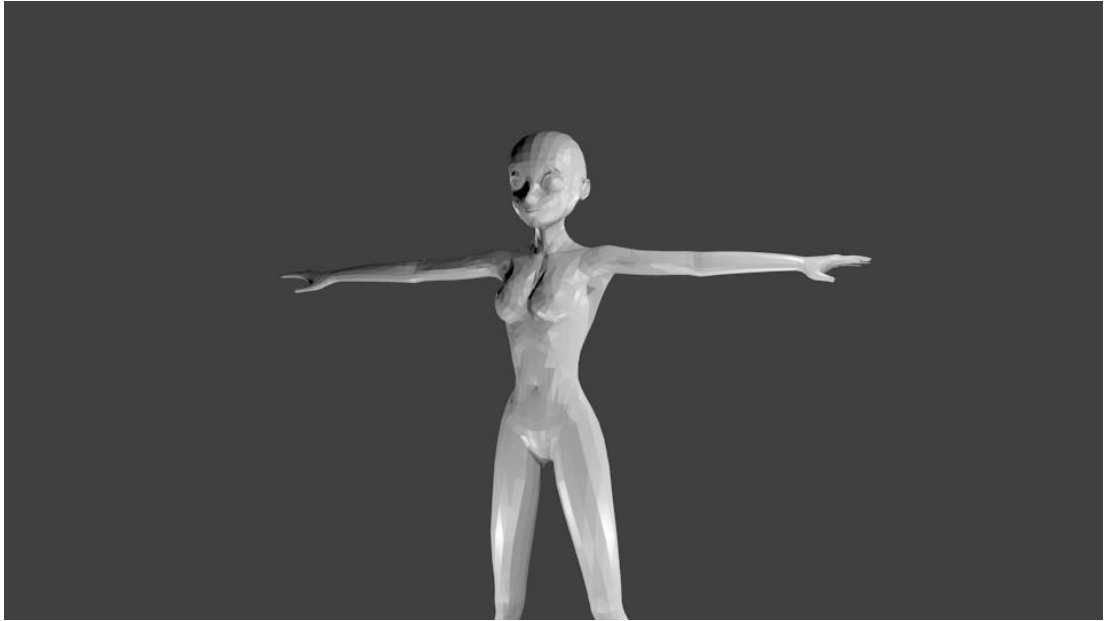
Depois de um resultado satisfatório com a personagem apresentada, aconteceu o mesmo com a segunda personagem imaginada, Mary Blood. Sua aparência também tem a ver com sua personalidade, que se assemelha mais a rebeldia apresentada por muitos jovens com a mesma mente.

Imagem 9: Mary Blood 2D (Rosto, cabelo, corpo e roupas)



Após a montagem bidimensional de ambos os modelos, começou-se a modelagem tridimensional dos mesmos, na mesma sequência que foram desenhadas, as personagens foram modeladas. Ao começar-se a modelar um corpo completamente novo, os resultados não foram os esperados, haviam bugs e problemas associados ao mesmo. Já que o problema não foi resolvido, então foi pego um modelo base que iria ser alterado para atender às características de cada personagem.

Imagem 10: Modelo pré-pronto



Depois de alteradas a partir do modelo bidimensional desenhado no papel, corrigimos os poucos *bugs* com detalhes que são quase imperceptíveis, mas que não deixam o modelo deformado, e sim mais parecido com algo real.

Imagem 11: Sakura (modelo 3D)



Imagem 12: Mary Blood (modelando)

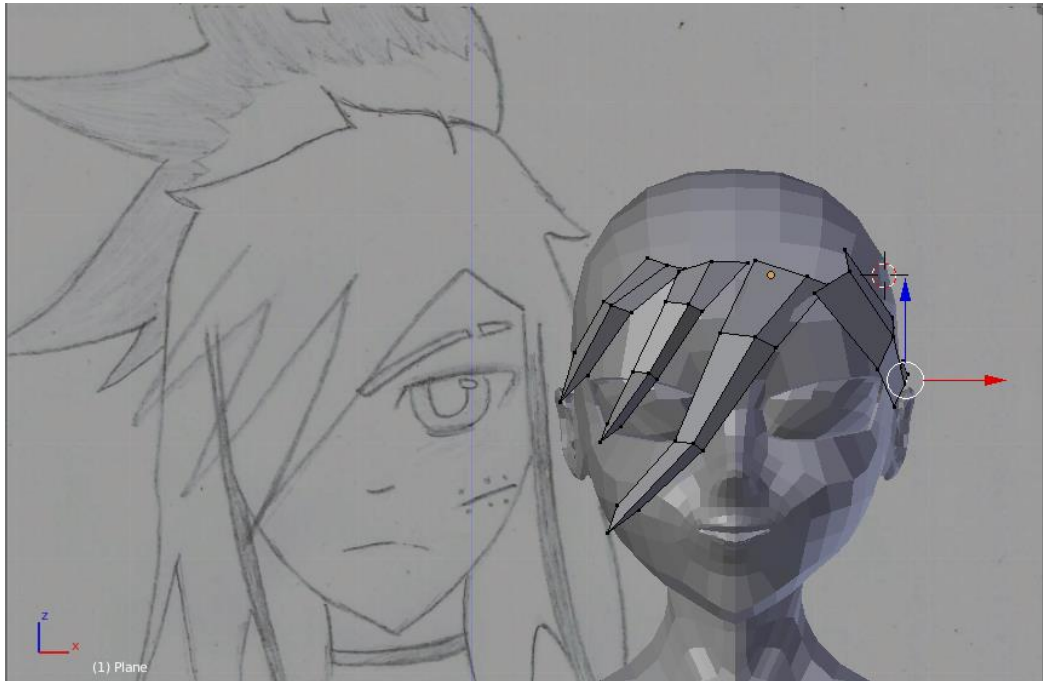
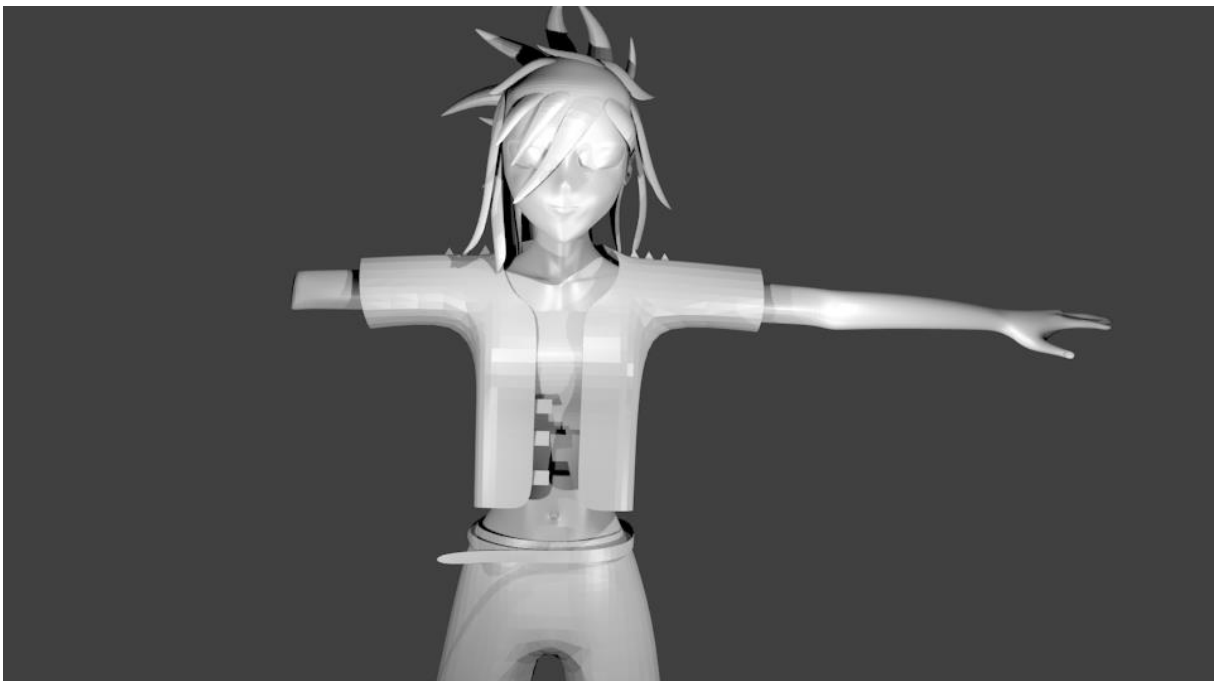


Imagem 13: Mary Blood (modelo 3D)



Paralelo a isso, foram sendo desenhados no papel alguns objetos, paisagens e personagens que iriam ser vistos e utilizados no *game*. Cenários pelos quais os personagens iriam passar, objetos que seriam usados pelos mesmos e personagens que iriam agir como inimigos ou NPC.

Imagem 14: Hotel 01 2D

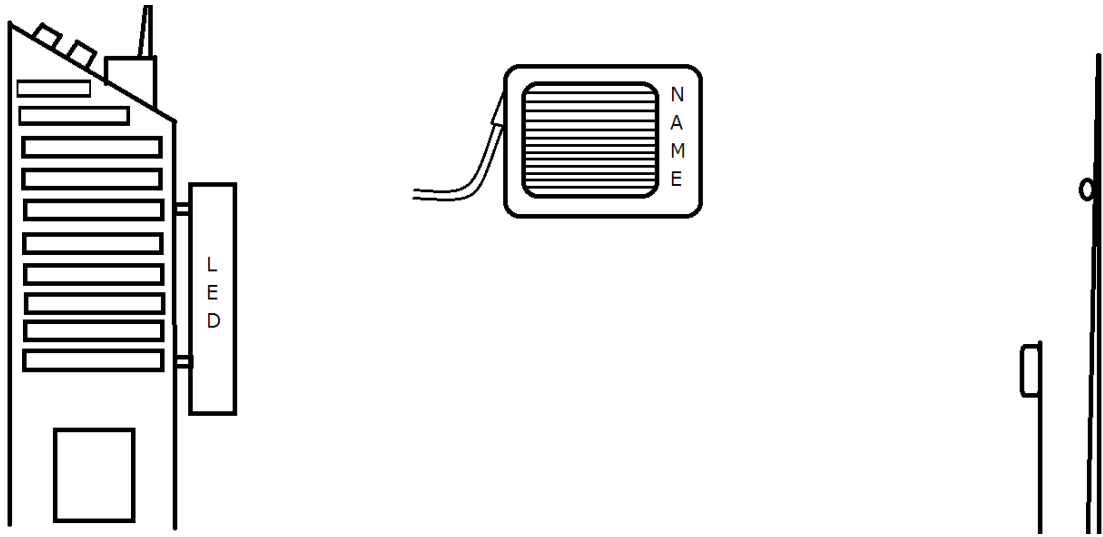


Imagem 15: Katana 2D

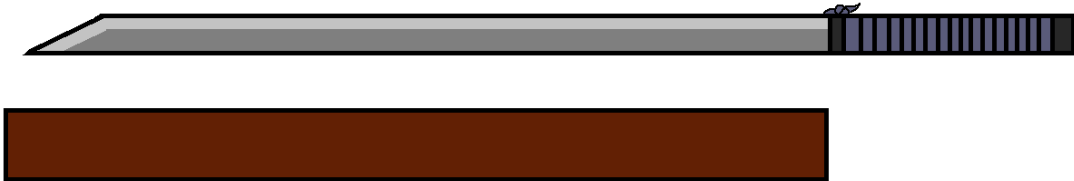
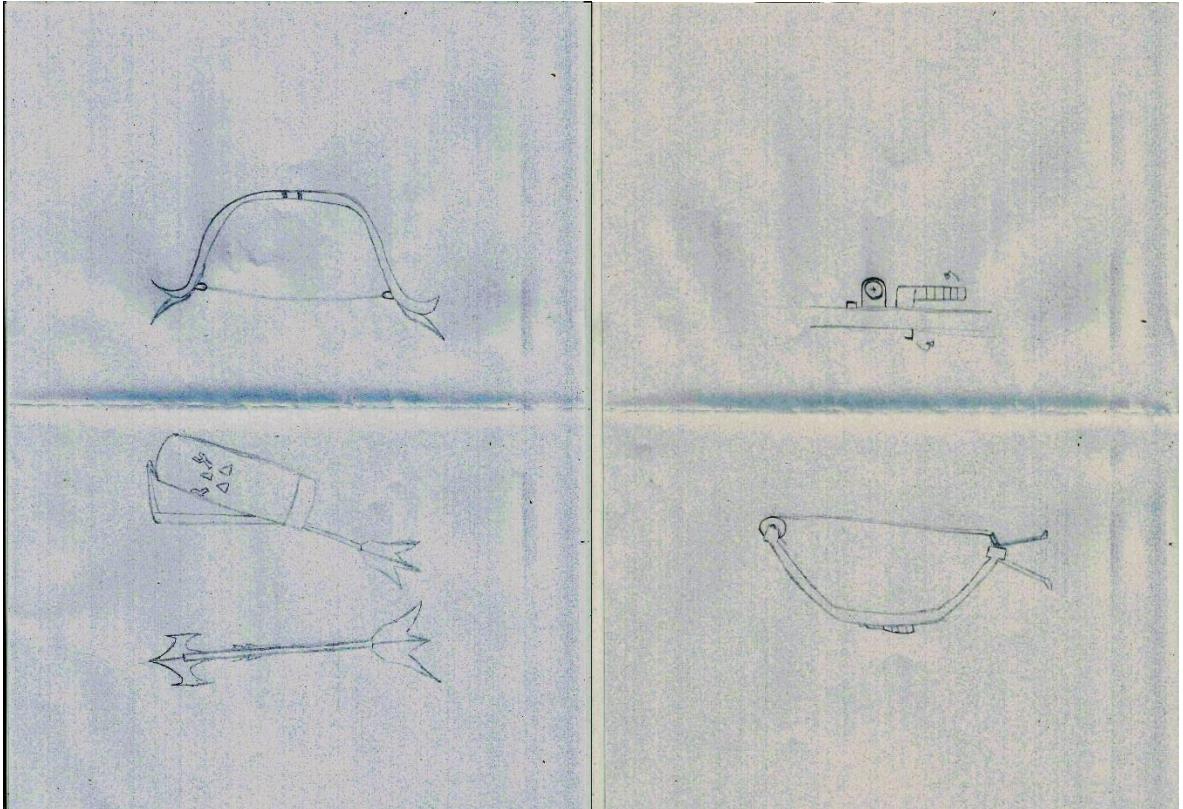
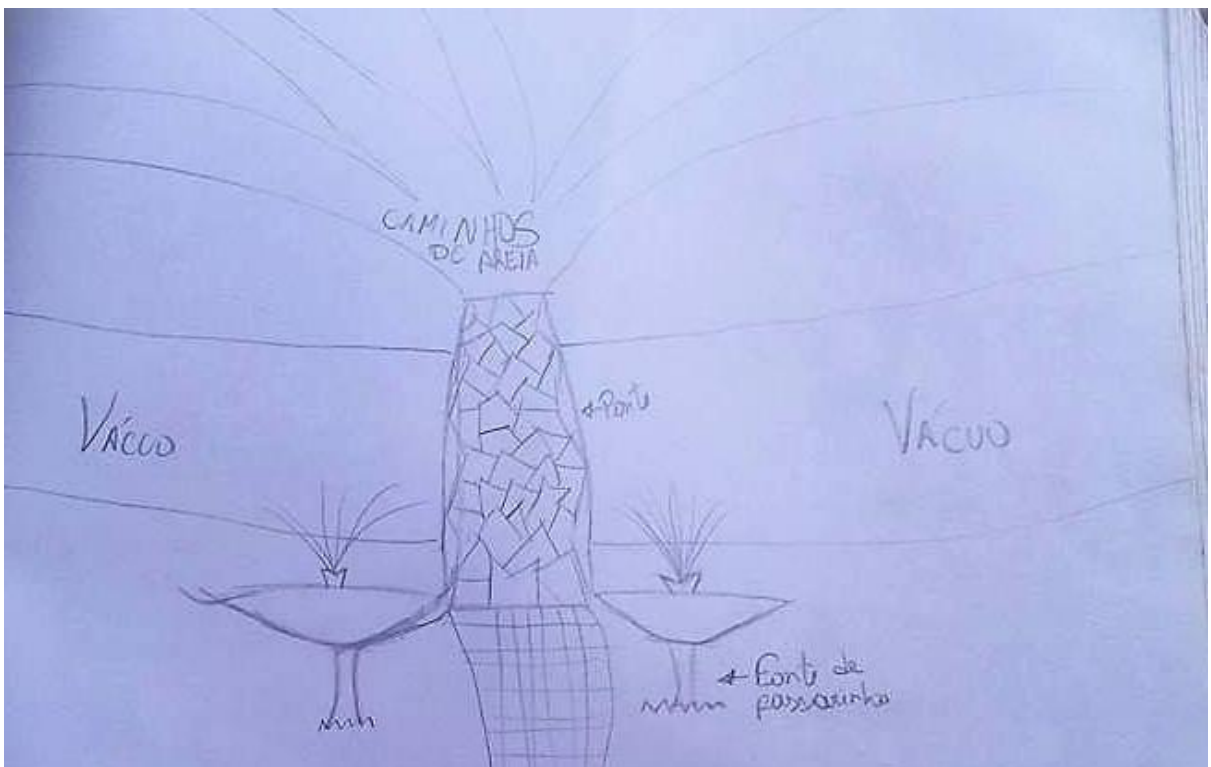


Imagem 16: *Peace Bow e flecha* 2DImagem 17: *Cena* 2D

Quando finalmente esteve pronto os modelos tridimensionais da Sakura e da Mary Blood apresentados, resolveu-se fazer alguns dos objetos, como a Katana, o arco, o hotel e a terceira cena aqui apresentada, tanto para ser possível fazer a cena a ser apresentada tanto por falta de prática e estudo sobre como fazer as animações precisas para fazer o jogo funcionar, estudos estes que foram ajudados principalmente em canais do *Youtube*, como o *GamesIndie*, *GearRecord*, *Game Development*, *Jornada Indie*, *TheGamingInc* e *Yinara100 animation*. Foram alguns meses de prática, já que ambos os componentes do trabalho não tinham experiência com animações feitas no *Blender*.

Imagem 18: *Peace Bow*

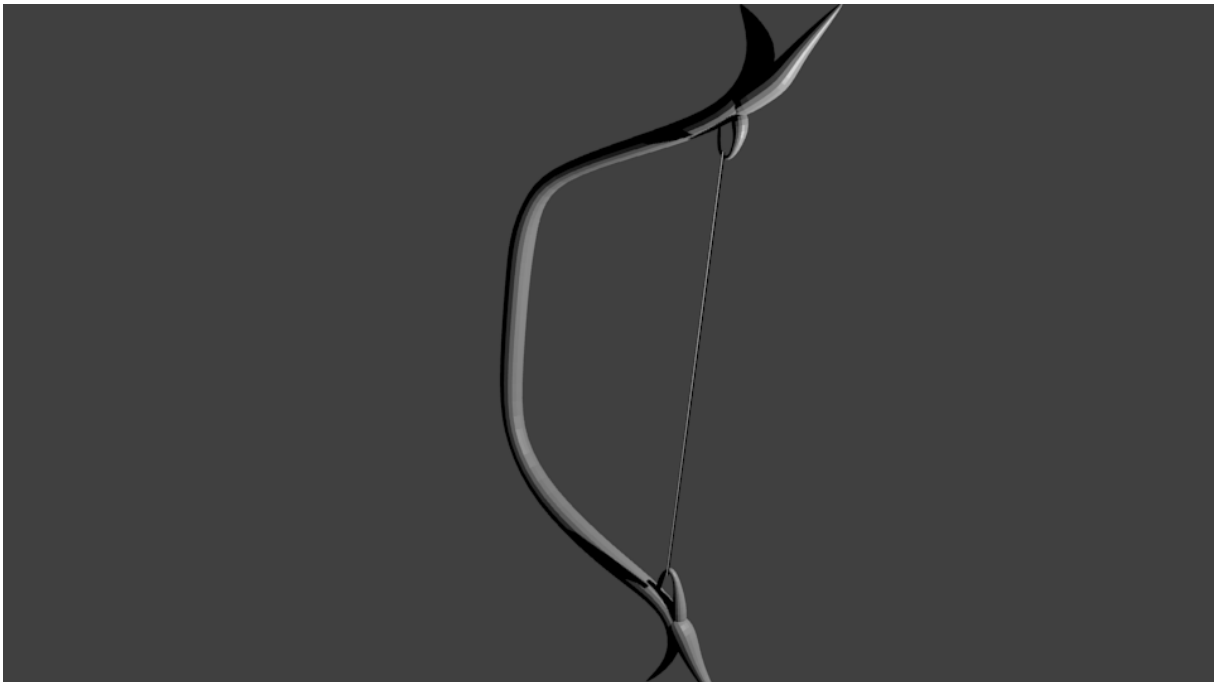


Imagem 19: Flecha

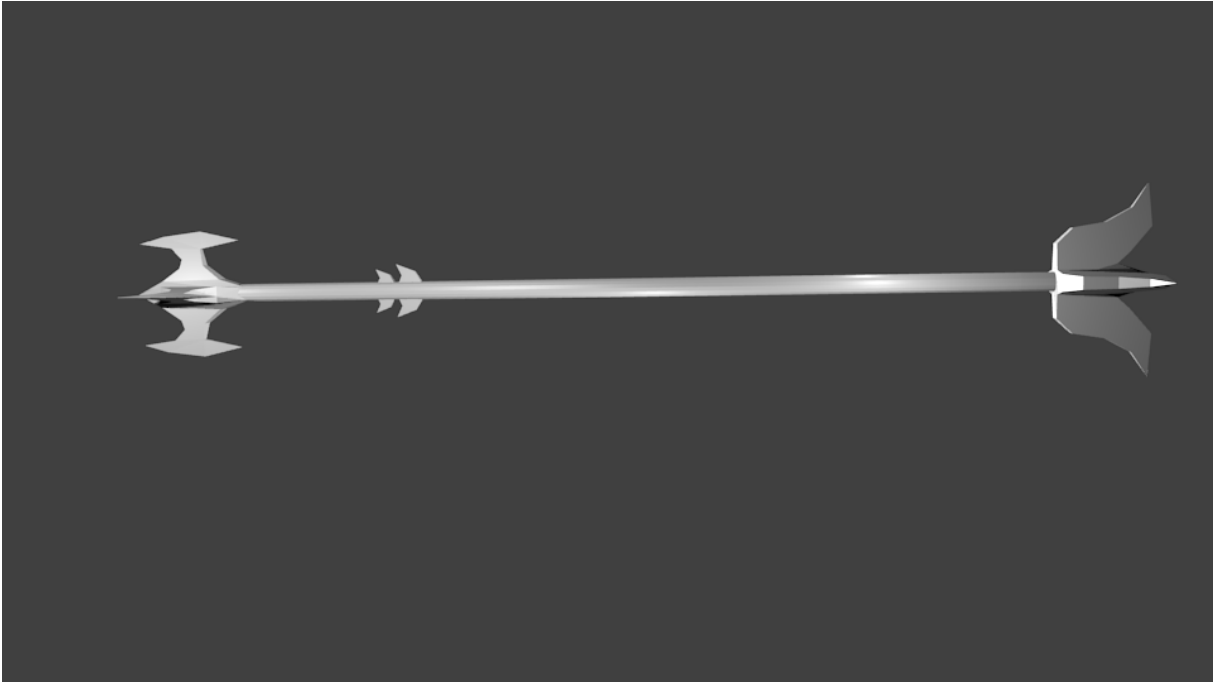


Imagem 20: Katana

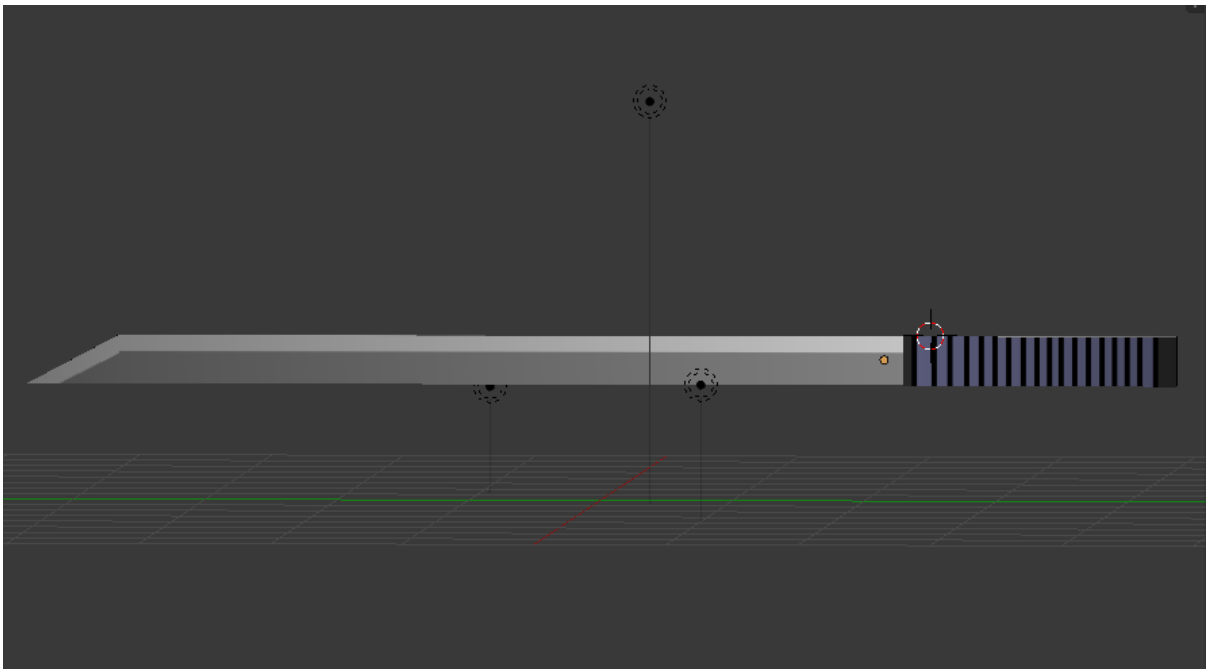
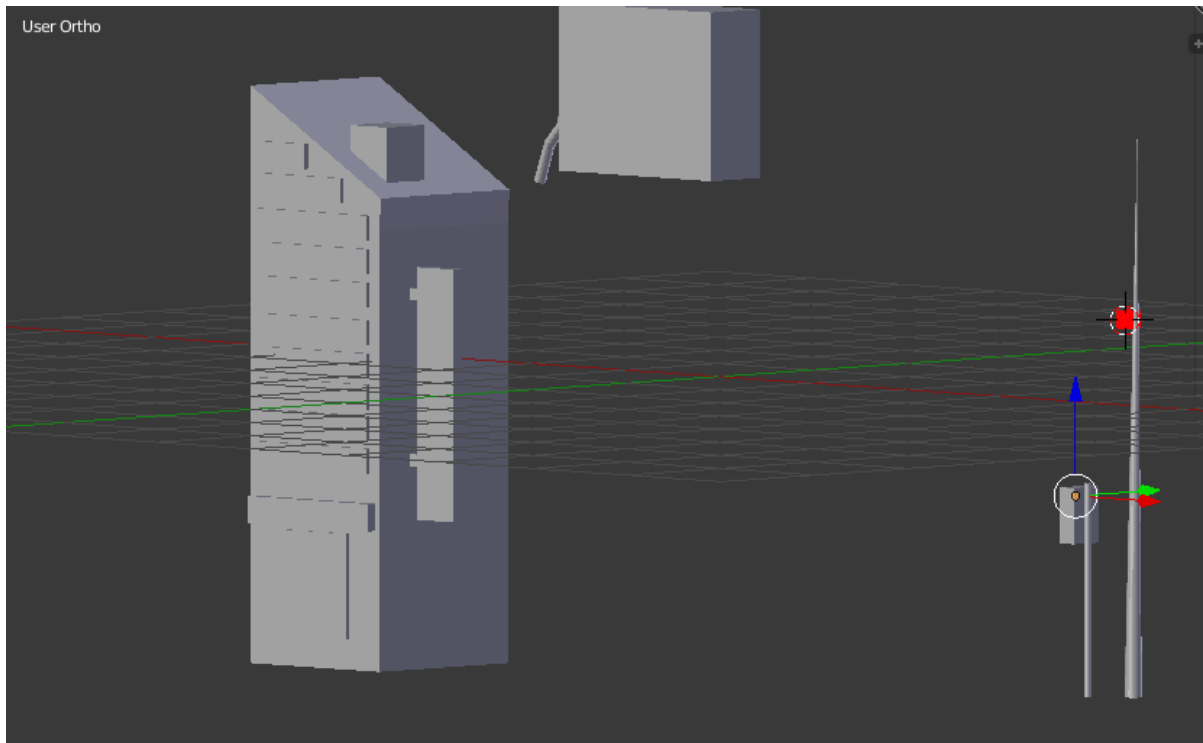


Imagem 21: Hotel



Estes foram alguns objetos que foram e serão usados na construção do jogo, seja para uso dos personagens, seja para construção do cenário ou seja em apenas algum momento específico do jogo. Todos conseguiram ter qualidade condizente ao que esperávamos para os mesmos e tiveram o modelo do esperado. Para pouco tempo e pouca experiência com o *Blender*, os modelos ficaram ideais e conseguiram passar o que era pretendido desde o início. O mesmo aconteceu com os objetos que foram desenhados e modelados, conseguiram passar o que lhes foi cobrado.

O cenário foi elaborado e montado logo em seguida, e embora muito mais trabalhoso que o restante já feito, também obteve resultado agradável. Ocorreu mudanças entre o seu planejamento e o que realmente foi feito, mas não deixou de ter uma ótima aparência final. O projeto inicial seria fazer apenas uma ilha que conteria as florestas e todas as cidades se concentrariam nela também, mas foi feita uma ilha que contém apenas a natureza do universo, está foi feita e será mostrada abaixo, enquanto às cidades, cada uma teria sua ilha, deixando assim o jogo mais organizado e mais chamativo ao *player*.

Imagem 22: Mapa bidimensional original



Imagem 23: Mapa 1 tridimensional (visão planificada)

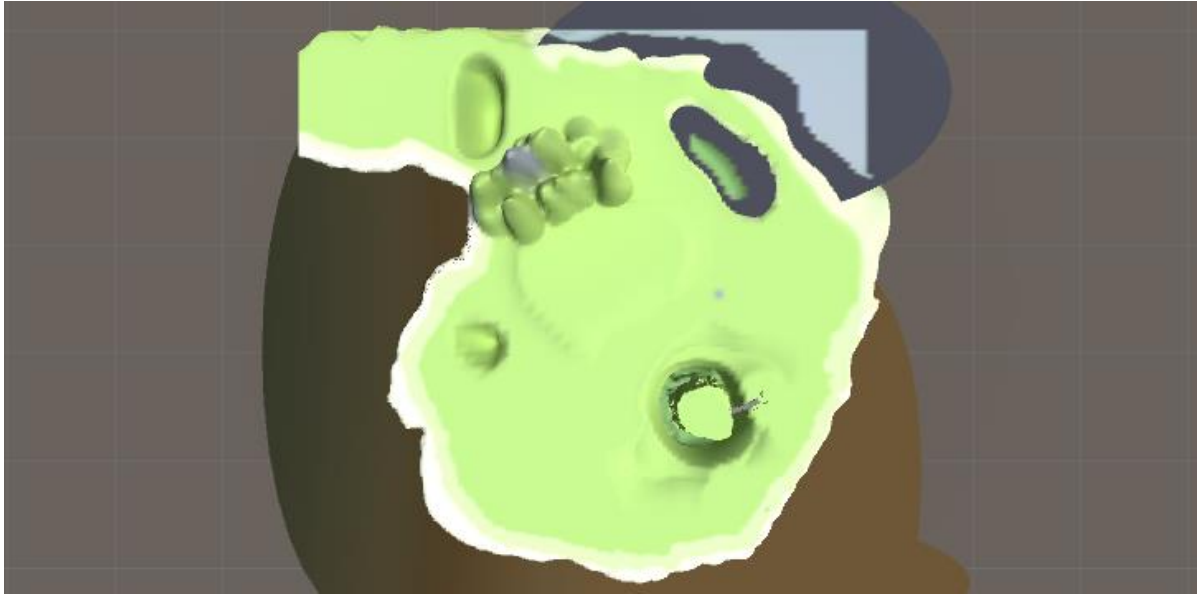


Imagem 24: Mapa 1 tridimensional (Vista do ar)

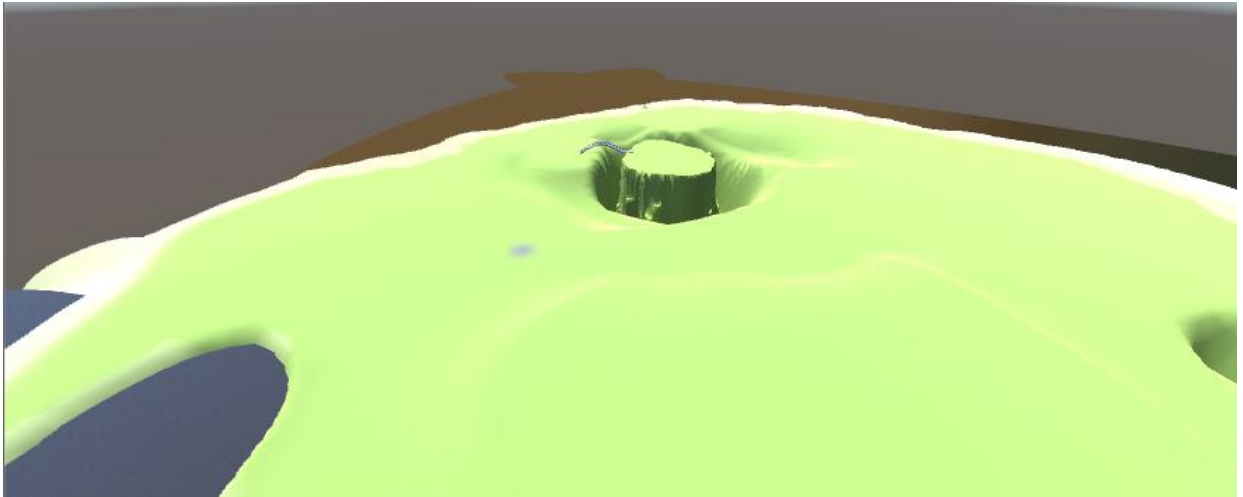


Imagem 25: Mapa 1 tridimensional (Montanhas)

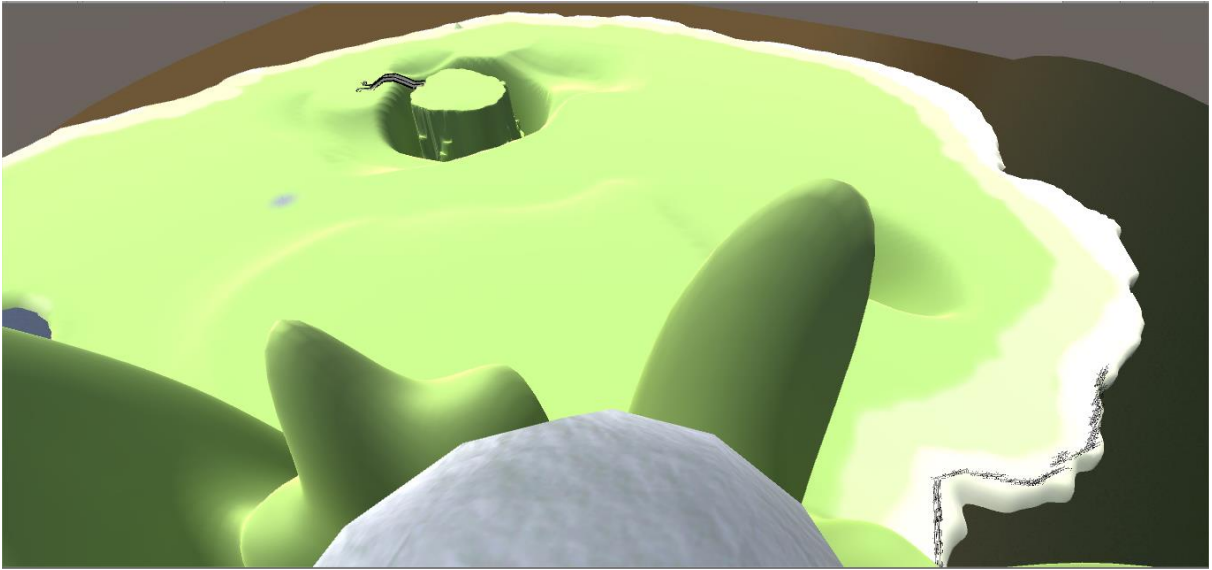


Imagem 26: Mapa 1 tridimensional (Mar)

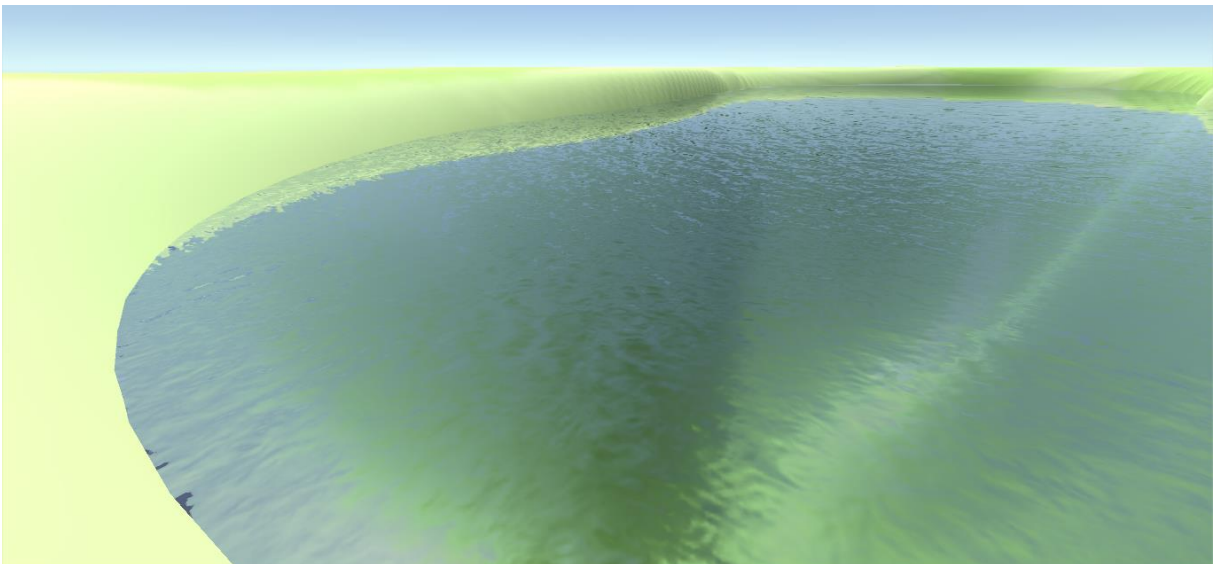


Imagem 27: Mapa 1 tridimensional (Horizonte)

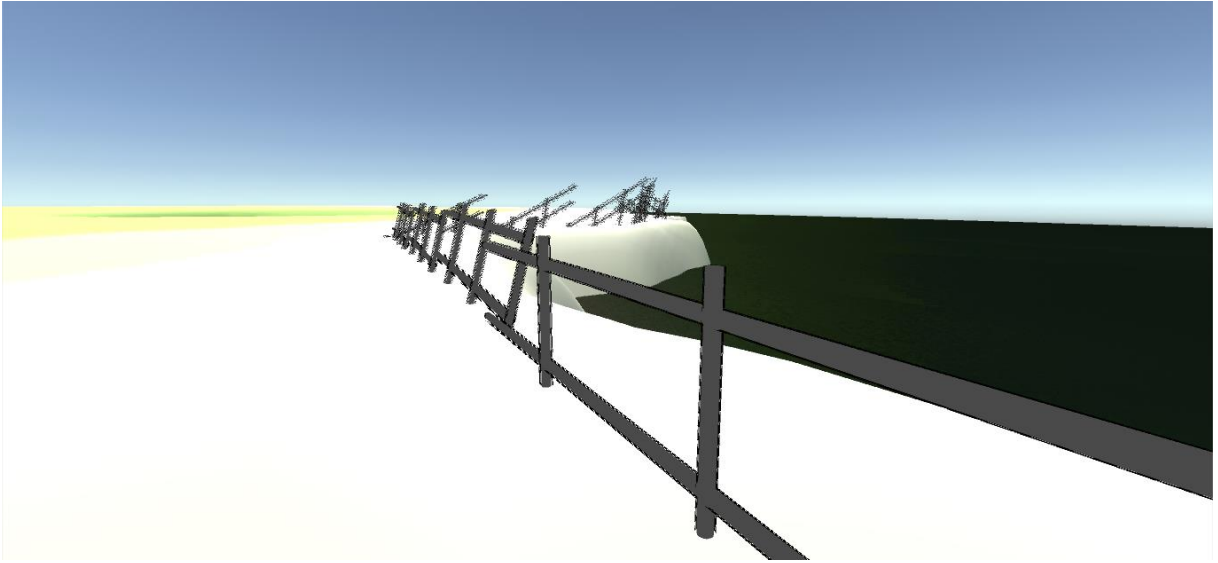


Imagem 28: Mapa 1 tridimensional (Entrada da vila)

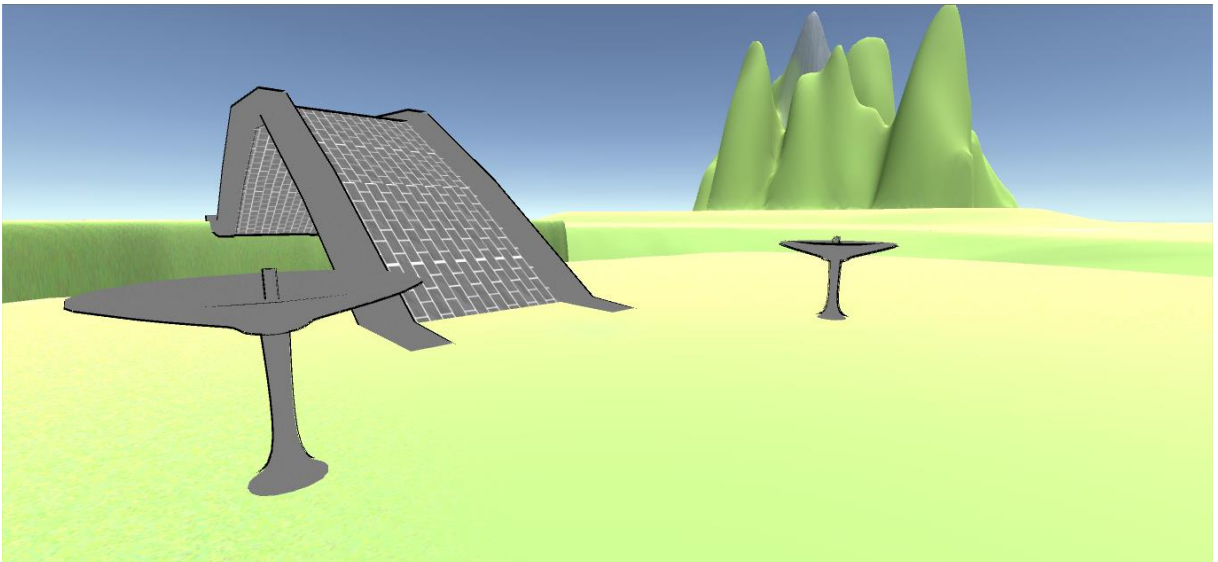
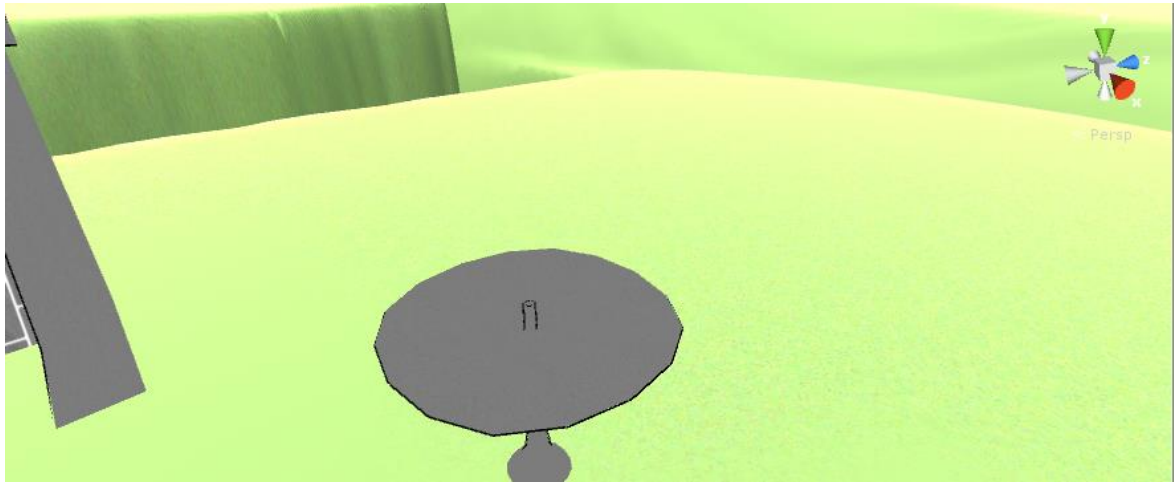


Imagem 29: Mapa 1 tridimensional (Fonte)



5. CONCLUSÃO

Esse trabalho tinha o principal objetivo de mostrar nossa capacidade de desenvolver não apenas um programa, mas também um projeto. Era esperado um jogo fluído, capaz de rodar em máquinas mais básicas, mas que também tivesse um belo visual e jogabilidade.

O jogo ainda não estará concluído na entrega deste documento tanto quanto na apresentação final, pois tal desenvolvimento demanda tempo e dedicação, tais quais, transpassam o tempo limite que recebemos como prazo final. Este, por sua vez, não serve apenas como trabalho de conclusão do curso (TCC), mas também como experiência e aprendizado para um mercado que pretendemos seguir.

Algumas metas não foram atingidas, no entanto, do mesmo modo que algumas não foram bem sucedidas outras foram incorporadas e até concluídas com relativo sucesso. Uma dificuldade enfrentada no desenvolvimento deste trabalho se deu no fato de este ser um assunto extra curricular e por tal motivo algumas dúvidas e problemas tiveram de ser sanados por meio de pesquisas que acabaram por consumir bastante tempo.

Os pontos que foram conseguidos com sucesso foram, principalmente, a modelagem tridimensional de várias itens que seriam usados, junto com as personagens. A construção do mapa, embora não tenha sido igual ao modelo inicial, também pode ser apontada como um objetivo alcançado. Contrário a isso, não foram feitos, ainda, muitos personagens e muitas cenas que farão parte do jogo, o enredo também não foi concluído e os mapas secundários ainda não estão prontos, objetivos a serem alcançados posteriormente, para uma conclusão do projeto e experiência para quem o fez.

Por se tratar de alunos de turnos inversos e pela distância, impediu-se as reuniões pessoais acontecerem de maneira frequente, o que acabou por se limitar a contato via internet que por vezes é falho ou complicado de manter-se estável. Outro problema recorrente foram os computadores que tiveram alguns problemas e acabaram por deixar algumas lacunas na sequência do desenvolvimento e que por fim restou apenas uma máquina fixa para o trabalho prosseguir.

É esperado desse projeto, apesar de problemas no desenvolvimento, que ele seja concluído com sucesso para que seja enfim lançado no mercado e traga resultados como: inserção no mercado, *feedback* da comunidade e entre outras coisas, além, claramente, do

sucesso na avaliação deste pela banca. Sendo assim, podendo dar continuidade a conclusão deste curso, que é o início de carreira.

Como primeiro desenvolvimento importante como programadores e técnicos, este permitiu a vivência de como se dá, ao menos por parte, um trabalho deste tipo, trazendo tanto experiências pessoais como profissionais de como é esta área da tecnologia e mercado. Trouxe conhecimento e reforçou as escolhas a respeito de tal área de mercado e escolhas pessoais, pois, além ser uma área da tecnologia, o desenvolvimento de jogos digitais também requer algum tipo de arte e a organização do trabalho para que assim possa ser concluído com sucesso.

5.1 PROBLEMAS E LIMITAÇÕES

Grande parte dos problemas que surgiram ao longo da construção do trabalho envolveram dano físico nos computadores usados para fazê-lo, embora tenham surgido muitos outros ao longo do caminho. O primeiro problema que surgiu foi não conseguir fazer o corpo base dos personagens, sempre surgia um defeito em alguma parte do corpo que não conseguia ter conserto. Por isso foi pego um modelo base para o corpo no *site* cgtrade.com e o alterava nos lugares precisos para dar forma ao corpo dos personagens feitos.

Os problemas principais, como dito anteriormente, foram danos físicos nos computadores. Com poucos dias de diferença, a placa mãe de um computador e a placa de vídeo do outro quebraram, impossibilitando assim todo o funcionamento de um computador, tornando impossível de fazer o trabalho no mesmo, enquanto no outro, houve dificuldade a respeito da renderização, pois os modelos tridimensionais perderam qualidade quando foram criados por não suportar configurações melhores.

Logo após uma nova placa ser comprada, a fonte do mesmo computador queimou, tornando-o impossível de ligar e, conseqüentemente, dar continuidade ao trabalho. Cerca de uma semana depois, a placa mãe do mesmo computador queimou, deixando-o novamente impossibilitado de ligar e deixar o projeto ser feito.

Um pequeno obstáculo no desenvolvimento encontrou-se no momento da programação do jogo, pois, apenas um dos membros dominava a língua utilizada no motor gráfico Unity 3d, que é o JavaScript. Por esse motivo alguma tempo que poderia ser melhor aproveitado acabou sendo usado para o aprendizado da nova linguagem, por meios de vídeo aulas, tutorias em sites, exercícios em sala de aula e em dúvidas que foram resolvidas com o parceiro de desenvolvimento e outros colegas.

Ao animar os personagens e tentar montar a cena que seria apresentada, surgiram alguns problemas, como o código não funcionar do modo que devia, ocorrendo *bugs* na cena, a animação não ficar como esperada. Acontece em mais de um momento na cena que está sendo feita e está tentando ser corrigido um por vez. Alguns demoram mais a ser concertados do que outros. O problema ainda está tentando ser corrigido até o momento atual.

Apesar deste obstáculo o desenvolvimento conseguiu seguir em frente e graças a comunidade do Unity 3d e dos desenvolvedores de jogos independentes na internet, mesmo as partes que tivemos mais dificuldade em programar puderam ser sanadas, tanto por meio de scripts já prontos, quanto por meio de dicas e tutorias focados nesse “departamento” do desenvolvimento.

6. REFERÊNCIAS

- CONSTRUCT 2. *Scirra LTD*. Disponível em: <https://www.scirra.com/construct2>. Acesso em 26 de nov. de 2016.
- DIAS, Rafael. Fazendo carreira no mundo dos jogos digitais: o guia completo!. *Produção de Jogos*, set./2014. Disponível em: <http://producaodejogos.com/fazendo-carreira-mundo-dos-jogos-digitais-o-guia-completo-44/>. Acesso em 15 de nov. de 2016.
- GAME DEVELOPMENT. Disponível em: <https://www.youtube.com/channel/UCCR955dujboBxEN2VJIZwJw>. Acesso em jul., ago., set., out. e nov. de 2016
- GAMESINDIE. Disponível em: <https://www.youtube.com/user/GamesIndie>. Acesso em jul., ago., set., out. e nov. de 2016
- GEARRECORD. Disponível em: <https://www.youtube.com/user/GearRecord>. Acesso em jul., ago., set., out. e nov. de 2016
- JORNADA INDIE. Disponível em: <https://www.youtube.com/user/jornadaindie>. Acesso em jul., ago., set., out. e nov. de 2016
- LIMA, Mariana. Mercado de games avança no Brasil. *Inova.jor*, jun./2016. Disponível em: <http://www.inova.jor.br/2016/06/01/mercado-games-brasil/>. Acesso em 17 de nov. de 2016.
- LOPES, Gilliard. Jogos Eletrônicos: Conceitos Gerais. Disponível em: http://www-usr.inf.ufsm.br/~pozzer/disciplinas/cga_8_classificacao_jogos.pdf. Acesso em: 26 de nov. de 2016
- LUCCHESI, Fabiano; RIBEIRO, Bruno. Conceituação de Jogos Digitais. Disponível em: <http://www.dca.fee.unicamp.br/~martino/disciplinas/ia369/trabalhos/t1g3.pdf>. Acesso em 26 de nov. de 2016.
- MATTAR, João. Ferramentas para Desenvolvimento de Games. *De Mattar*, jan./2015. Disponível em: <http://joaomattar.com/blog/2015/01/03/ferramentas-para-desenvolvimento-de-games/>. Acesso em: 26 de nov. de 2016.
- MENDONÇA, Vinicius Godoy de. Outros gêneros: Variações e jogos importantes. *Ponto V*, set./2015. Disponível em: <http://www.pontov.com.br/site/game-design/67-classificacao-dos-jogos/78-outros-generos-variacoes-e-jogos-importantes>. Acesso em 26 de nov. de 2016.
- MOGNON, Mateus. Mercado de games gera US\$ 99,6 bilhões mundialmente e PC é a plataforma mais lucrativa, indica pesquisa. *Adrenaline*, Florianópolis, mar./2016. Disponível em: <http://adrenaline.uol.com.br/2016/04/30/42730/mercado-de-games-gera-us-99-6-bilhoes->

[mundialmente-e-pc-e-a-plataforma-mais-lucrativa-indica-pesquisa](#). Acesso em 17 de nov. de 2016.

MÜLLEREM, Leonardo. China se torna maior mercado para games do mundo; Brasil sobe para 12°. *TechMundo*, jun./2016. Disponível em <https://www.tecmundo.com.br/video-game-e-jogos/106554-china-supera-eua-torna-maior-mercado-games-mundo.htm>. Acesso em 15 de nov. de 2016.

MUSTAFA, Eduardo. JavaScript: 20 anos de história e construção da web. *iMasters*, jan./2016. Disponível em: <http://imasters.com.br/front-end/javascript/javascript-20-anos-de-historia-e-construcao-da-web/?trace=1519021197>. Acesso em 30 de nov. de 2016.

NEWZOO'S Top 100 Countries By 2015 Game Revenues. *Newzoo*, out./2015. Disponível em: <https://newzoo.com/insights/articles/newzoos-top-100-countries-by-2015-game-revenues/>. Acesso em: 16 de nov. de 2016.

O MERCADO de Games. *SAGA - Escola de desenvolvimento de jogos, arte digital, design e efeitos visuais*, set./2016. Disponível em: <https://www.saga.art.br/o-mercado-de-games/>. Acesso em 17 de nov. de 2016

SCHUYTEMA, P. *Design de games: uma abordagem prática*. São Paulo: Cengage Learning, 2008. 447 p.

STARCKEM, Daniele. A história do Photoshop: o editor de imagens mais usado no mundo. *TechMundo*, mar./2013. Disponível em: <https://www.tecmundo.com.br/photoshop/37907-a-historia-do-photoshop-o-editor-de-imagens-mais-usado-no-mundo.htm>. Acesso em: 30 de nov. de 2016.

TEXEIRA, Marcelo. Desenvolvimento de Jogos no Brasil. *TechinBrazil*, ago./2015. Disponível em: <https://techinbrazil.com.br/desenvolvimento-de-jogos-no-brasil>. Acesso em 15 de nov. de 2016.

THEGAMEINC. Disponível em: <https://www.youtube.com/user/809MasUsacA809>. Acesso em jul., ago., set., out. e nov. de 2016

VELLOSO, Ariel. SPCine: Edital De Produção De Games. Disponível em: <http://spcultura.prefeitura.sp.gov.br/projeto/2208#>. Acesso em 17 de nov. de 2016.

YINARA100 ANIMATION. Disponível em: <https://www.youtube.com/user/yinara100>. Acesso em jul., ago., set., out. e nov. de 2016

GLOSSÁRIO

Cyber punk: Termo usado para designar histórias, filmes, jogos, imagens ou qualquer forma de arte ou entretenimento que se baseie num futuro distópico e avançado tecnologicamente.

Fantasia: Gênero ou termo usado para os mesmos fins que o primeiro, a característica deste por sua vez, é um mundo fantasioso e com uma visão menos realista do mundo e que normalmente remete a tempos passados, ou até mesmo tempos futuros, porém que remetem muito a ideia de natureza, energia e crenças.

Sand Box: Termo designado para um jogo que apresenta muito conteúdo extra, como easter eggs, missões secundárias, histórias ocultas e entre outras coisas.

Open World: Quando é permitido ao jogador que ele tenha a liberdade de movimentar-se e interagir com todo o mundo que é apresentado, sem prendê-lo ao enredo, permitindo que o ambiente seja explorado independente de como se desenrola a o gameplay principal, este jogo leva o título de “open world”, ou “mundo aberto” em português.

Gameplay: Termo em inglês que significa “jogabilidade” ou “jogatina” em uma tradução livre. É usado para expressar quando se está tratando da ação executada pelo jogo.

APÊNDICE A – IMAGENS DOS COMPONENTES

Serão apresentados aqui algumas imagens dos componentes que serão usados ao longo do jogo, tanto alguns bidimensionais, usados como base para modelos tridimensionais quando os tridimensionais já feitos no *Blender*, prontos para ser inseridos nos jogos.

Imagem 1: Anji 2D

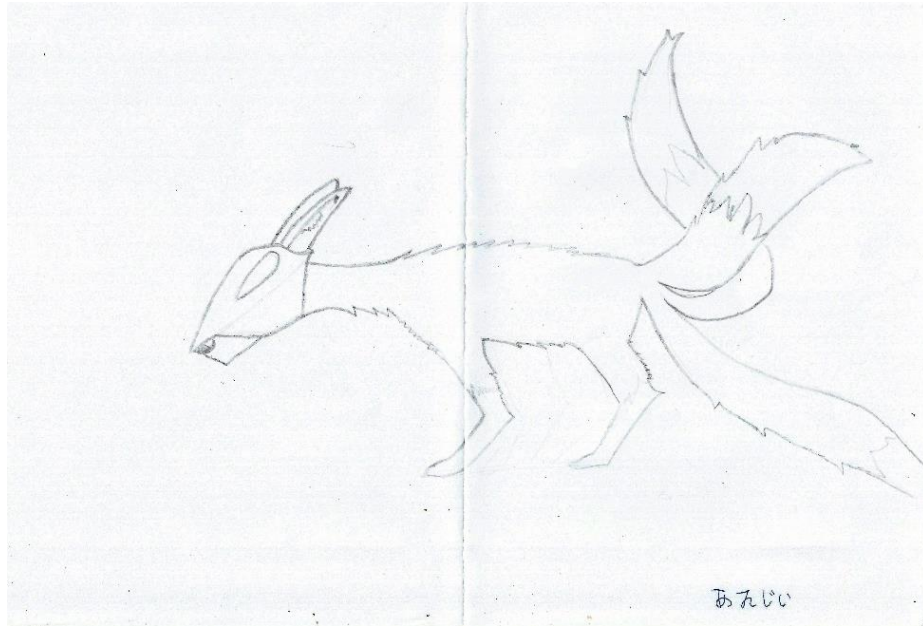


Imagem 2: The Blind One 2D

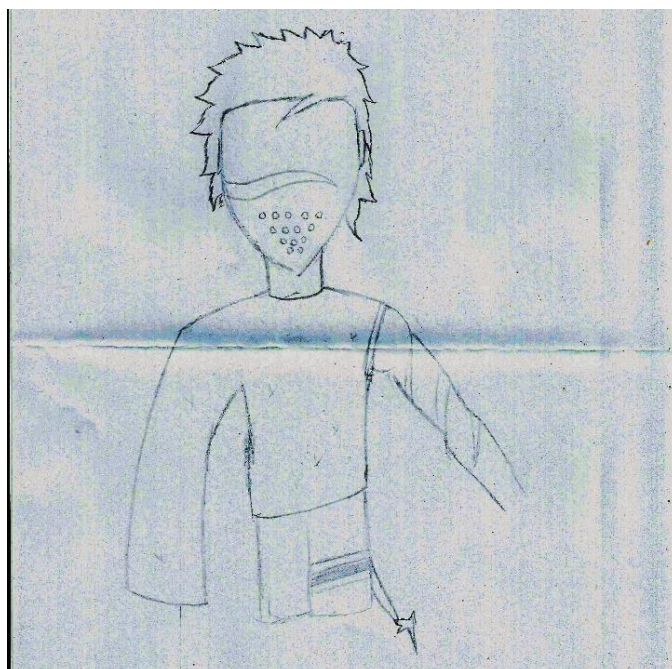


Imagem 3: Homem tentáculo 2D

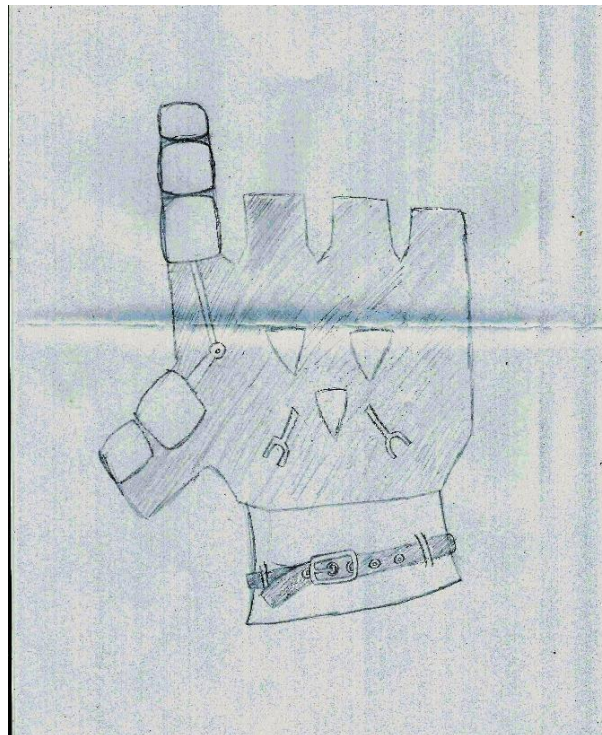
Imagem 4: *Peace Glove* 2D

Imagem 5: Braço mecânico da Mary Blood 2D

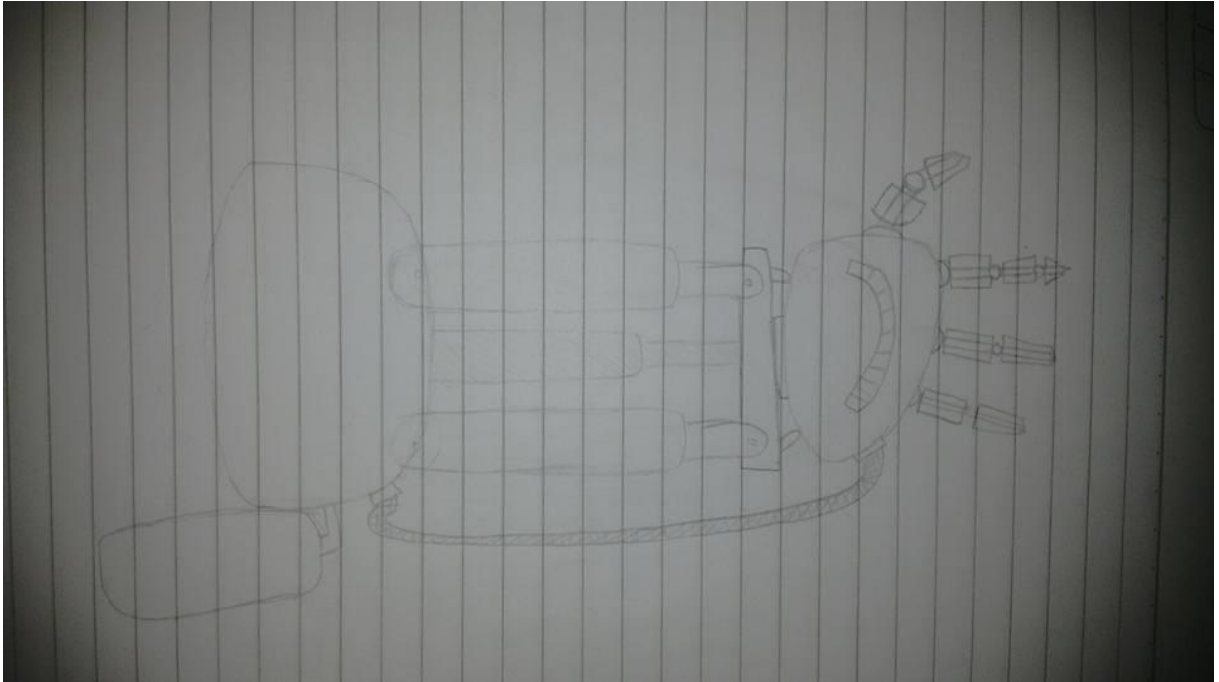


Imagem 6: Cena (1) 2D

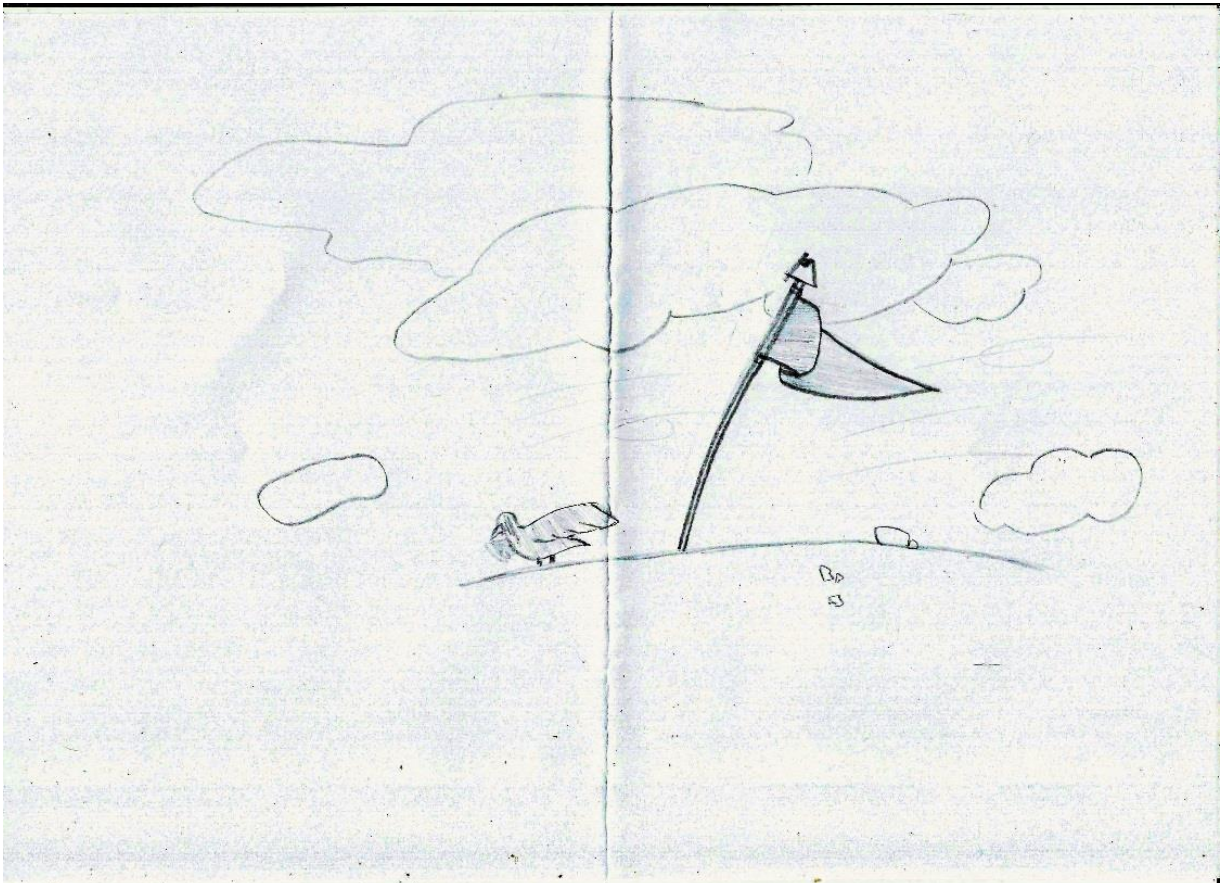


Imagem 7: Cena (2) 2D

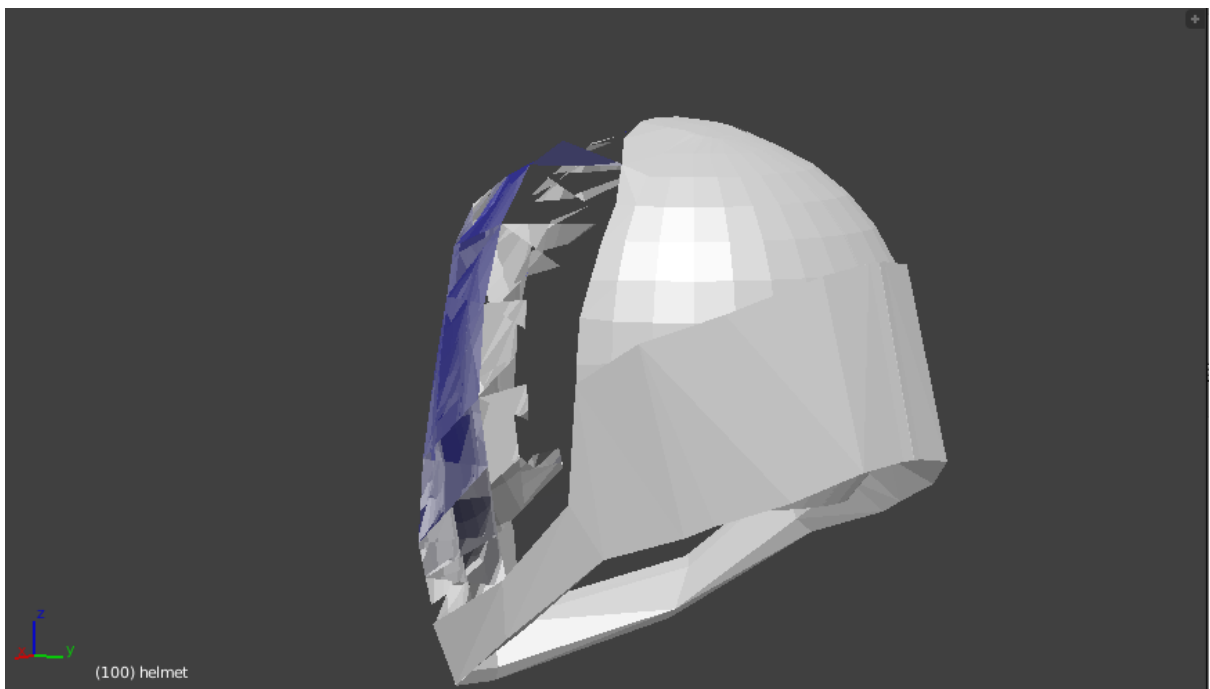
Imagem 8: *Daft-helmet*

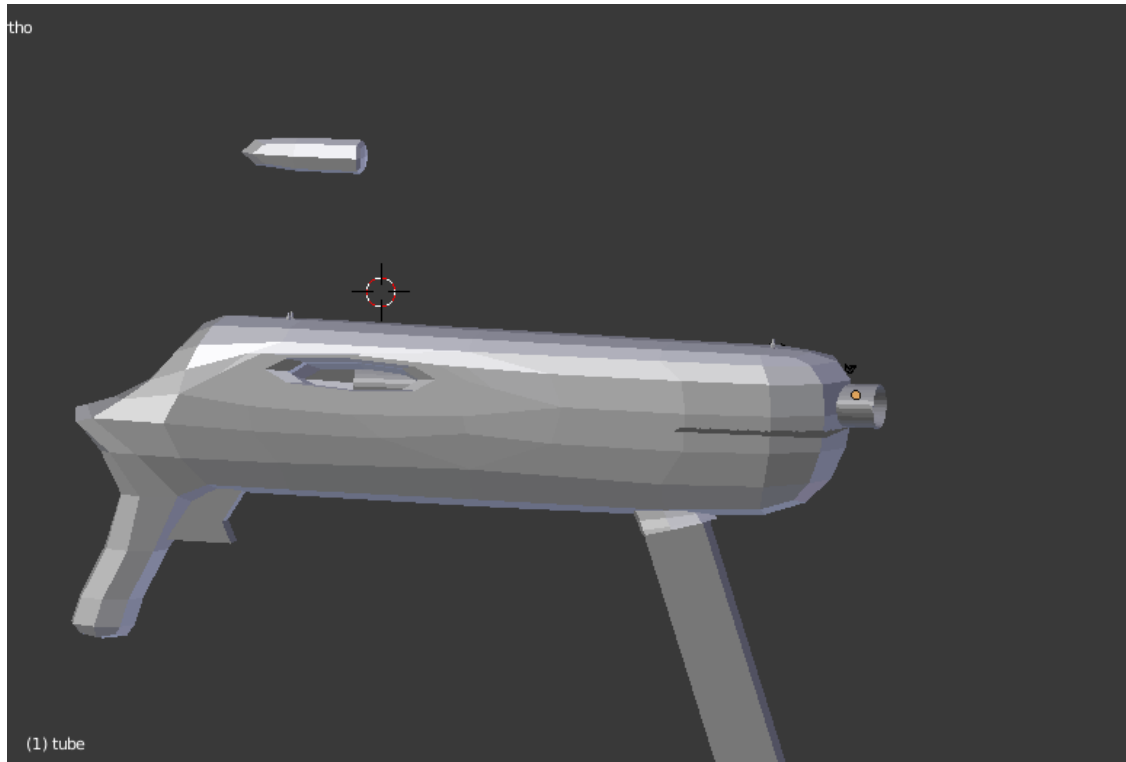
Imagem 9: *Tommy Gun*

Imagem 10: Fonte

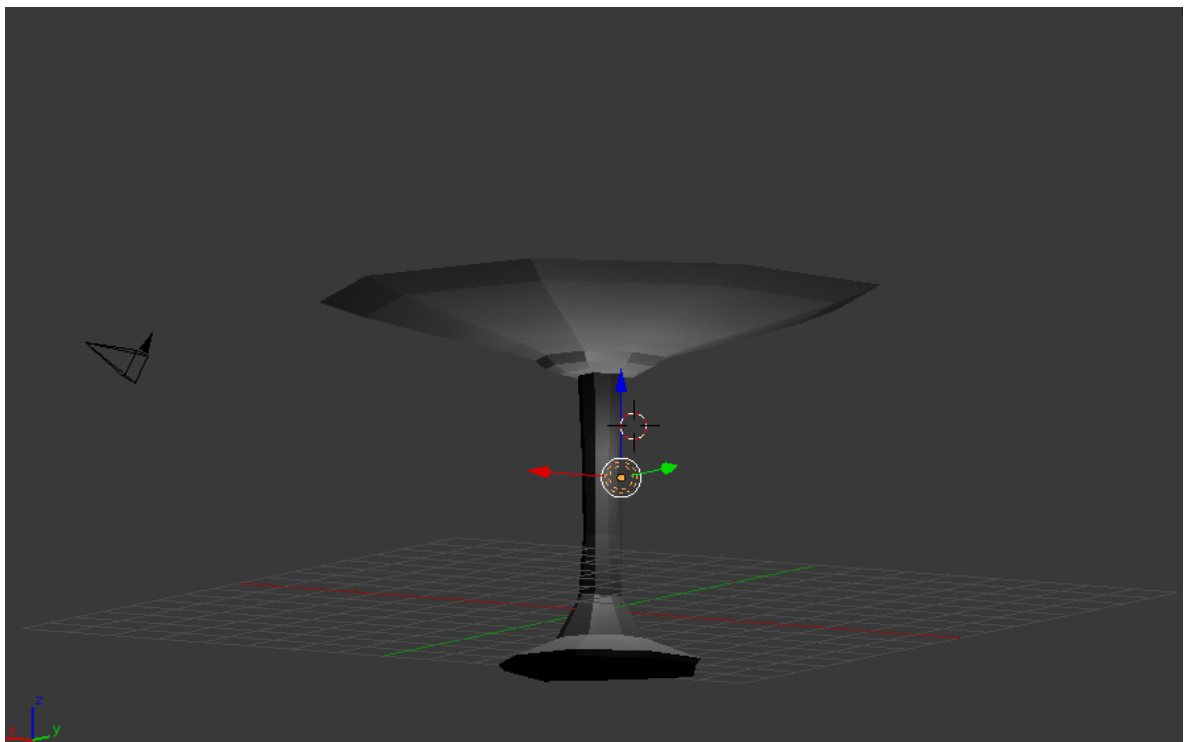
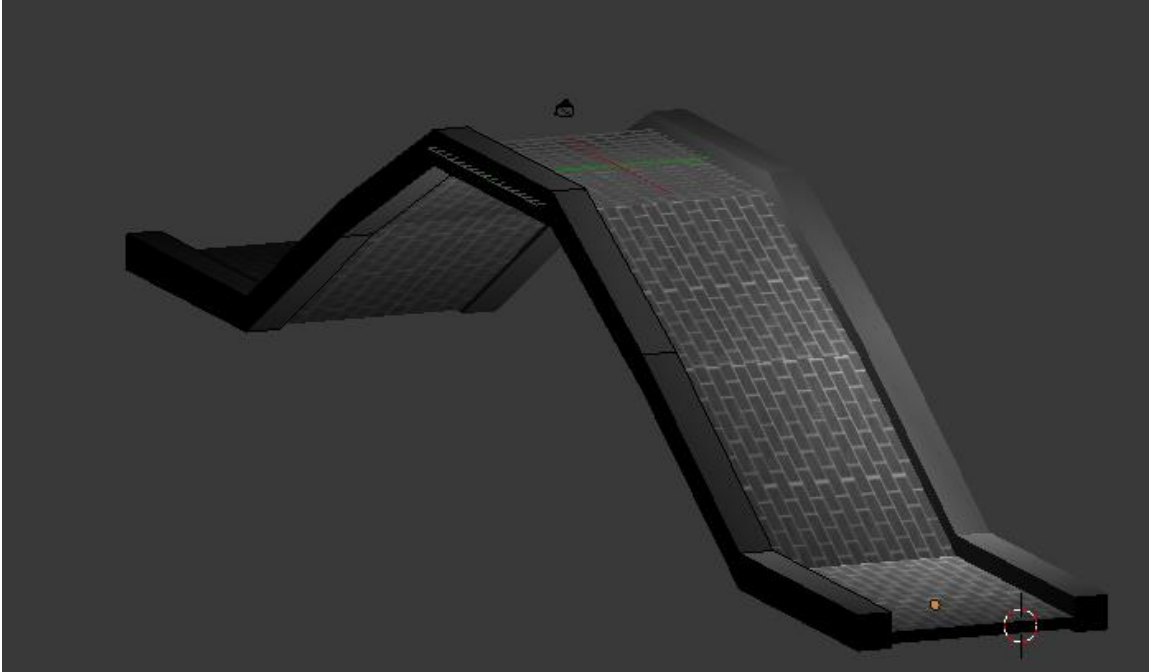


Imagem 11: Ponte

Imagem 12: *Merchandising rocket*