

Gilberto Martini de Oliveira

Realidade virtual em apresentação de projeto arquitetônico

Projeto de Conclusão de Curso do Departamento de Expressão Gráfica, Centro de Comunicação e Expressão da Universidade Federal de Santa Catarina, como requisito final para a obtenção do grau de bacharel em Design.

Orientador: Prof. Me. Flávio Andaló

Florianópolis
2016

Ficha de identificação da obra elaborada pelo autor
através do Programa de Geração Automática da Biblioteca Universitária
da UFSC.

A ficha de identificação é elaborada pelo próprio autor
Maiores informações em:
<http://portalbu.ufsc.br/ficha>

Gilberto Martini de Oliveira

Realidade virtual em apresentação de projeto arquitetônico

Projeto de Conclusão de Curso julgado e aprovado pelo departamento de Expressão Gráfica, Centro de Comunicação e Expressão da Universidade Federal de Santa Catarina para a obtenção do grau de Bacharel em Design.

Florianópolis 23 de novembro de 2016.

Prof. Dr. Luciano Patrício Souza de Castro
Coordenador do Curso

Banca Examinadora:

Prof. Me. Flávio Andaló
Orientador
Universidade Federal de Santa Catarina

Prof.^a Dr.^a Regiane Trevisan Pupo
Universidade Federal de Santa Catarina

Prof. Dr. William Andrade
Universidade Federal de Santa Catarina

Este trabalho é dedicado para a minha família, em especial para a minha mãe que tanto me incentiva a ser forte e seguir sempre em frente.

AGRADECIMENTOS

Quero agradecer primeiramente a Deus e minha família, pelo apoio incondicional. Especialmente minha mãe, que sempre me incentivou a seguir em frente não importando a dificuldade.

Agradeço meus amigos, por sempre estarem presentes e tornarem a minha vida e a jornada deste projeto de conclusão de curso mais leve e divertida.

Agradeço ao povo brasileiro por ter subsidiado esta graduação, e o intercâmbio que fiz nos Estados Unidos em 2014 por meio do programa ciência sem fronteiras. Foi a partir deste intercâmbio que descobri a paixão pela área de realidade virtual.

RESUMO

O objetivo deste projeto de conclusão de curso, é guiar o leitor no processo de produção de uma aplicação de Realidade Virtual para a apresentação de um projeto de arquitetura, usando a tecnologia de jogos digitais. Por meio de um óculos de Realidade Virtual e a sensação de imersão, o usuário pode sentir-se em um projeto que ainda não existe, possibilitando uma maior compreensão do espaço projetado. Como resultado final, foi criado um apartamento com uma decoração convidativa no qual o usuário com um *Oculos Rift*, conseguirá movimentar-se livremente e será capaz de interagir com a decoração trocando revestimentos.

Palavras-chave: realidade virtual, jogos digitais, sensação de imersão, interação.

ABSTRACT

This project's objective is to walk the user through the production process of architectural design in a 3D virtual environment using the game technology. The use of a head mounted display enables the user to navigate the virtual space, creating the sensation of being within the design. Furniture, everyday household accessories, and art are added making the environment feel warm and inviting. The final result will be a virtual reality application allowing the user to interact with the environment as designed.

Keywords: Virtual Environment, game technology, Virtual Reality,

LISTA DE FIGURAS

Figura 01 - Exemplo de uma maquete eletrônica.	27
Figura 02 – Metodologia do Duplo diamante.	30
Figura 03 - A batalha de Borodino.	33
Figura 04 - Estereoscópio.	34
Figura 05 - Sensorama.	34
Figura 06 - Sega Genesis.	35
Figura 07 - Nintendo Virtual Boy.	36
Figura 08 - John D. Carmack II apresentando o oculus rift na E3 de 2012.	37
Figura 09 - Exemplo de dispositivos que compõe um sistema de RV. .	38
Figura 10 - Aplicativo PokemonGo.	39
Figura 11 - Capacete de RV F-35 Gen III (HMDS).	41
Figura 12 - HTC vive.	42
Figura 13 - Google Cardboard.	42
Figura 14 - kit de construção do google cardboard.	43
Figura 15 - Oculus Rift DK2.	43
Figura 16 - Níveis de resolução em um objeto 3D.	44
Figura 17 - Programa 3D Max.	45
Figura 18 - Programa Maya.	46
Figura 19 - Programa Modo.	46
Figura 20 - Imagem da interface do programa Blender 3D.	47
Figura 21 - Da esquerda para a direita, exemplifica-se vértice, aresta, triângulo e polígono ou face.	48
Figura 22 - Exemplo de modelo 3D criado com a técnica box modeling.	49
Figura 23 - Na esquerda o modelo de um cilindro, na direita a malha poligonal do cilindro planificada.	50
Figura 24 - Nesta imagem após a edição no photoshop, criou-se uma textura de concreto tendo como base a malha planificada anteriormente.	50
Figura 25 - Ponto de pivot do objeto.	51
Figura 26 - Objeto posicionado no centro do mundo virtual.	52
Figura 27 - Modelo 3D da esquerda antes do processo de triangulação, e da direita ja somente com triângulos.	52
Figura 28 - Interface da UE4.	53
Figura 29 - Exemplo de programação visual Blue Print.	54
Figura 30 - Imagem da tela de importação de modelos 3D na UE4.	54
Figura 31 - Modelo 3D importado na pasta principal do projeto.	55

Figura 32 - Modelo 3D com material aplicado e textura usada para dar a aparência desejada.....	55
Figura 33 - Referência quarto.....	56
Figura 34 - Referência da sala de estar e cozinha.....	57
Figura 35 - Referência do banheiro e lavanderia.....	58
Figura 36 - Referência da sacada.....	58
Figura 37 - Referência área de jantar.....	59
Figura 38 - Unidade de medidas em centímetros.....	60
Figura 39 - Evolução da modelagem das paredes.....	60
Figura 40 - Faces externas separadas das internas.....	61
Figura 41 - Vista 01 dos ambientes com demais objetos.....	61
Figura 42 - Vista 02 dos ambientes com demais objetos.....	62
Figura 43 - Diferentes materiais aplicados no modelo 3D.....	63
Figura 44 - Modelo 3D mostrando canais de mapeamento.....	64
Figura 45 - Exemplo de mapeamento do sofá.....	65
Figura 46 - Modelo 3D posicionado na cena do mundo virtual do blender.....	66
Figura 47 - Modelo 3D posicionado no centro do mundo virtual do blender.....	67
Figura 48 - Selecionar opção fbx para exportar.....	67
Figura 49 - Configuração para exportar em fbx.....	68
Figura 50 - Página inicial do aplicativo <i>Epic Games Launcher</i>	69
Figura 51 - Modelos de templates oferecidos pela UE4.....	70
Figura 52 – Estrutura de pastas e template de primeira pessoa.....	70
Figura 53 – Arquivos importados na pasta estruturas.....	71
Figura 54 – Selecionar e arrastar modelos para o mundo virtual da UE4.....	72
Figura 55 – Posicionar modelos 3D e operações de escala e rotação....	72
Figura 56 – Modelos importados e cena completa na UE4.....	73
Figura 57 – Explo de luz do tipo point e spot light.....	74
Figura 58 – Parâmetros das cores.....	74
Figura 59 – Modelos envolvidos pelo lightmass importance volume...	75
Figura 60 – Níveis de <i>lighting quality</i>	76
Figura 61 – Criar material.....	77
Figura 62 - Material branco.....	78
Figura 63 – Material bulbo da lâmpada.....	79
Figura 64 – Todos os materiais aplicados.....	79
Figura 65 – Volume envolvendo o modelo 3D para desencadear evento de interação.....	80
Figura 66 – Sistema inicial de programação para habilitar interatividade.....	81

Figura 67 – Evento troca de materiais.	82
Figura 68 – Requisitos de configuração de computador para ter boa performance em aplicações de RV.	82
Figura 69 – Tela do local para baixar os drivers do Oculus Rift.	83
Figura 70 – Selecionando o modo de preview em RV.	84
Figura 71 - Aplicação final do apartamento sendo testada.	85
Figura 72 – Salvar projeto como uma aplicação de RV.	86

Sumário

1	Introdução	27
1.1	Apresentação do tema	27
1.2	Problematização	28
1.3	Objetivo Geral	29
1.4	Objetivo Específico	29
1.5	Justificativa	29
1.6	Metodologia Projetual	30
1.7	Descobrir (<i>Discover</i>)	31
1.8	Definir (<i>Define</i>)	31
1.9	Desenvolver (<i>Develop</i>)	32
1.10	Entregar (<i>deliver</i>)	32
2	Coleta de dados e análise	33
2.1	Histórico da realidade virtual	33
2.2	O que é a Realidade Virtual	37
2.3	Tipos de sistemas de Realidade Virtual	39
2.4	Características da Realidade Virtual imersiva	40
2.5	Dispositivos de Realidade Virtual	41
2.6	Criação de mundos virtuais	44
2.6.1	Programa de modelagem 3D	45
2.6.2	Modelagem 3D (box modeling)	48
2.6.3	Mapeamento 3D	49
2.6.4	Salvar e exportar modelos	51
2.6.5	Integração com o motor de jogos Unreal Engine 4	53
3	Desenvolvimento	56
3.1	Escolha de referências	56
3.2	Modelagem 3D	59
3.3	Mapeamento 3D	62
3.4	Exportar modelos 3D	66
3.5	Importar na Unreal Engine 4	69
3.6	Iluminação	73
3.7	Criação de materiais	76
3.8	Progamação de Interatividades (Blueprint)	80
3.9	Testes	82
3.10	Exportar aplicação	85
4	Conclusão	87
	REFERÊNCIAS	88

LISTA DE ABREVIATURAS E SIGLAS

2D - Duas dimensões

3D - Três dimensões

CAD – Computer-Aided Design

RV - Realidade Virtual

PCC - Projeto de conclusão de curso

HMD - Head Mounted Display

UE4 - Unreal Engine 4

1 Introdução

1.1 Apresentação do tema

É notável o crescimento e sofisticação da tecnologia de jogos digitais nos últimos anos. A qualidade gráfica de muitos jogos para desktop ou mesmo consoles, ironicamente exigem mais capacidade computacional que a maioria dos programas CAD¹ conhecidos. A evolução das placas de vídeo, por meio da aceleração gráfica, tem permitido o uso de texturas com alta resolução possibilitando assim, ricas simulações em tempo real de ambientes interativos (HOON; KEHOE, 2003). Aproveitando este avanço tecnológico, a realidade virtual (RV) ressurgiu em 2012 com o lançamento de um dispositivo chamado *oculus Rift* e trouxe a possibilidade de criar, além de jogos digitais, simulações de ambientes em tempo real com um grau de imersão nunca visto antes.

Por mais de duas décadas, arquitetos vem fazendo uso do computador para criar representações de seus projetos e mostrar aos clientes, visando melhor entendimento e interação do espaço desenvolvido (Figura 01). Estas representações geralmente são impressas ou visualizadas em monitores o que já oferece ao cliente uma ideia mais realista do projeto (SHNEIDERMAN; PLAISANT, 2009).



Figura 01 - Exemplo de uma maquete eletrônica.

Fonte: próprio autor

¹ - Computer-Aided Design – Desenho assistido por computador. Fonte: <<http://dicionario.sensagent.com/CAD/pt-pt/>>

Na medida em que a tecnologia está evoluindo, outros recursos começam a ser utilizados e sempre com intenção de facilitar o entendimento de informações entre o arquiteto (especialista) e o cliente (não especialista). Neste sentido, a tecnologia de RV possibilita ao cliente experimentar e interagir um ambiente virtual, com um grau de imersão altamente realista. Estas novas possibilidades de interação, tais como poder controlar uma câmera virtual por meio das setas do teclado do computador, interagir com objetos virtuais trocando revestimentos, ou poder movimentar-se virtualmente num ambiente, possibilitam perceber e entender projetos antes mesmo de existirem no mundo físico.

A cada avanço, as possibilidades de interação fazem com que o usuário não mais sinta-se simplesmente olhando para algo, mas sim, parte de algo. Neste projeto de conclusão de curso (PCC), será criado um ambiente virtual imersivo em que será possível interagir e movimentar-se livremente, mostrando novas possibilidades e capacidades de interação no entendimento de um ambiente ainda não construído.

1.2 Problematização

A produção de aplicações de RV, é uma tarefa que exige conhecimento de diversas áreas. Algumas das quais pode-se destacar:

- Produção de ambientes 3D,
- Mapeamento de modelos 3D,
- Integração de modelos 3D em motores de jogos ou outros programas de RV,
- Programação de objetos interativos,

Para isso, foi usado como base teórica autores nacionais e internacionais, os quais nos dão uma visão geral de como este tipo de produção é feita. Primeiramente será abordado a seleção de imagens que servirão de base para um ambiente 3D e demais objetos como, sofás, mesas, cadeiras, objetos de decoração, etc. Após isso, serão criados os modelos 3D para a composição de um apartamento completo. Etapas de como importar e construir um ambiente virtual no motor de jogos Unreal Engine 4 serão exemplificadas por meio de imagens para facilitar o entendimento. Por fim será abordado o processo e ajustes para exportar uma aplicação de RV para as plataformas *Windows* e *Mac*.

1.3 Objetivo Geral

O objetivo geral deste projeto, é criar uma aplicação de Realidade Virtual para a apresentação de um projeto de arquitetura usando a tecnologia de jogos digitais.

1.4 Objetivo Específico

Como objetos específicos destacam-se:

- Escolha imagens de ambientes que servirão de referência;
- Modelagem 3D de um apartamento, juntamente com objetos de decoração;
- Texturização de todos os objetos modelados em 3D;
- Integração dos modelos 3D com o programa de realidade virtual.
- Gerar aplicação em realidade virtual para ser visualizada com o *Oculus Rift DK2*² para as plataformas Windows e Mac.

1.5 Justificativa

O avanço tecnológico que foi alcançado ao longo dos últimos dez anos, favoreceu para que a tecnologia de Realidade Virtual (RV) amadurecesse o suficiente para tornar-se viável nos dias de hoje. Em 2012 a *Oculus VR*, empresa responsável pela produção do *oculus rift*, arrecadou mais de dois milhões de dólares em uma campanha de financiamento coletivo no site *Kickstarter*³. Em 2014 o *Facebook* comprou a *Oculus VR* por dois bilhões de dólares. Desde então grandes empresas como a *Microsoft*, *HTC* e a *Google*, estão investindo na produção de óculos e produtos que usam o potencial da RV.

Atualmente muitas empresas como *Epic Games*⁴, *Crytec*⁵, e *Unity*⁶ disponibilizam versões gratuitas de seus motores de jogos (*game*

² **Oculus Rift DK2** é um óculos de RV produzido pela Oculus VR. Fonte: <<https://www.oculus.com>> acesso em agosto de 2016.

³ **Kickstarter** é o maior site de financiamento coletivo do mundo e que busca apoiar projetos inovadores. Fonte: <<https://www.kickstarter.com>> Acesso em agosto de 2016.

⁴ **Epic Games** é a desenvolvedora da engine de jogos *Unreal engine*. Fonte: <<https://epicgames.com>> Acesso em agosto de 2016.

engines) para integração e disponibilização jogos digitais e aplicações em RV, de modo que seja necessário pagar *royalties* somente após a geração de lucro. Este tipo de iniciativa favorece para que estudantes e profissionais sejam capazes de criar soluções em RV antes mesmo de ter um orçamento disponível.

Em 2016, a RV apareceu pela primeira vez na lista de desejos dos brasileiros, o que mais uma vez aponta, para uma grande demanda de aplicações usando esta tecnologia. Para tanto, cada vez mais será necessário o apoio à pesquisas e projetos que auxiliem na produção de aplicações em RV, assim como este PCC.

1.6 Metodologia Projetual

A metodologia projetual abordada para a realização deste projeto será a do *Double Diamond* conforme figura 02.

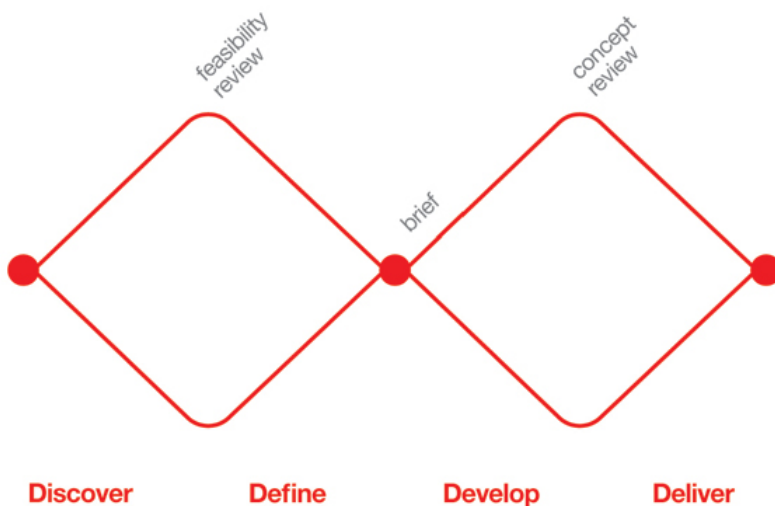


Figura 02 – Metodologia do Duplo diamante.

Fonte <<http://www.designcouncil.org.uk>>. Acesso em jun de 2016

⁵ Fundada em 1999, **Crytek** é a desenvolvedora da engine de jogos *CryEngine*. Fonte: <http://www.crytek.com> Acesso em agosto de 2016.

⁶ A **Unity 3D** é a desenvolvedora do motor de jogos *Unity*. Fonte: <<https://unity3d.com>> Acesso em agosto de 2016.

Esta metodologia descrita como “*Double Diamond*” foi criada no ano de 2005 pelo *Design Council* por meio de uma pesquisa com departamentos de design de onze empresas de nível global. Este modelo aborda quatro estágios, convergentes e divergentes de um projeto de design que são: Descobrir, definir, desenvolver e entregar.

1.7 Descobrir (*Discover*)

A primeira etapa do diamante duplo, demonstra o início do projeto, o qual tem como finalidade adotar uma ideia ou necessidade de algum assunto baseado nas necessidades de um grupo de usuários. No caso deste projeto, foi pensado na necessidade de estruturar o processo de produção de aplicações em realidade virtual imersiva para apresentação de projetos de arquitetura. De acordo com isto, essa fase fica estruturada da seguinte forma:

- Escolha do tema do PCC;
- Pesquisa sobre RV;
- Tecnologias de RV;
- Busca por imagens de ambientes, móveis, cores, decoração, etc;
- Análise e escolha de referências;

1.8 Definir (*Define*)

A segunda etapa do duplo diamante pode ser considerada uma das mais importantes, já que é neste momento que deve se escolher as tecnologias a serem utilizadas.

- Imagens de referências de um ambiente;
- Técnicas de modelagem 3D;
- Programa de modelagem 3D;
- Programa de RV;

1.9 Desenvolver (*Develop*)

A terceira etapa do duplo diamante é o momento da produção técnica, as quais são:

- Modelagem 3D;
- Mapeamento dos modelos 3D;
- Integração dos modelos 3D com o programa de RV;
- Programação das interatividades;
- Teste da aplicação;

1.10 Entregar (*deliver*)

A última etapa do duplo diamante refere-se a entrega do projeto, que neste caso é:

- Aplicação de realidade virtual para dois sistemas operacionais: *Windows* e *Mac*;

2 Coleta de dados e análise

2.1 Histórico da realidade virtual

Acredita-se que as primeiras tentativas de proporcionar experiências imersivas, foram definitivamente os murais panorâmicos do século 19 (figura 03). Estes grandes murais eram pintados em salas, e cobrindo todas as paredes tinham a intenção de proporcionar a quem olhasse a sensação de presenciar uma cena ou evento histórico (RIBEIRO, 2007).



Figura 03 - A batalha de Borodino.

Fonte: <http://moscow.touristgems.com/history/17787-battle-of-borodino-1812/>.
Acesso em junho de 2016

Em 1838 o físico *Sir Charles Wheatstone*⁷ criou o estereoscópio (figura 04), que veio a ser o primeiro dispositivo mais tecnológico a proporcionar a sensação de imersão. Este dispositivo, no formato de um óculos, que através de suas lentes juntava duas imagens que posicionadas em pontos diferentes davam a impressão de tridimensionalidade.

⁷ **Sir Charles Wheatstone** (Gloucester, 6 de fevereiro de 1802 — Paris, 19 de outubro de 1875) foi um cientista britânico.

Inventor de muitas das inovações científicas da era vitoriana, incluindo a concertina inglesa, o estereoscópio (um dispositivo para exibir imagens tridimensionais), e a cifra Playfair (uma técnica de criptografia).



Figura 04 - Estereoscópio.

Fonte: <https://pt.wikipedia.org/wiki/Estereoscópio>. Acesso em jun de 2016

Entretanto foi na década de 50 e 60 que a RV começou a tomar a forma que conhecemos hoje, através de um dispositivo chamado *sensorama* (figura 05), inventado por *Morton Heilig*⁸ que recebeu o título de “pai da realidade virtual” (PEREIRA; NOGUEIRA, 20--?). O sensorama funcionava como uma cabine que oferecia tanto experiências imersivas, por meio de imagens e sons, quanto sensoriais com aromas e movimentos.



Figura 05 - Sensorama.

Fonte: <http://web.ist.utl.pt/ist170613/>. Acesso em junho de 2016.

⁸ **Morton Heilig** – acessado em 24/05/2016 <<http://www.mortonheilig.com>>.

Porém, o conceito de realidade virtual, foi introduzido somente no início da década de oitenta por *Jaron Lanier*⁹, o qual foi criado “para descrever os sistemas que possibilitavam a realização de simulações por vários utilizadores num ambiente partilhado“ (RIBEIRO, 2007).

Somente na década de 90 com a popularização dos vídeo jogos que empresas como *Sega*¹⁰, e *Nintendo*¹¹, começaram a aplicar os conceitos de RV em seus produtos. Porém dispositivos como *Sega VR* (figura 06), e *Nintendo virtual boy* (figura 07), foram considerados um fiasco, porque eram pesados, desconfortáveis e não ofereciam boa qualidade gráfica ao usuário (NOGUEIRA, 2016).



Figura 06 - Sega Genesis.

Fonte: < <http://www.ign.com/articles/2016/01/14/the-evolution-of-virtual-reality>>. Acesso em junho de 2016

⁹ **Jaron Zepel Lanier** Pioneiro na indústria de RV.

Fonte: < https://en.wikipedia.org/wiki/Jaron_Lanier> Acesso em outubro de 2016.

¹⁰ **Sega**. É uma empresa desenvolvedora de programa para Video games, e uma antiga produtora de consoles.

¹¹ **Nintendo** empresa japonesa fabricante de videogames.

Fonte: < <http://www.nintendo.com/corp/index.jsp>> Acesso em outubro de 2016



Figura 07 - Nintendo Virtual Boy.

Fonte: < <http://www.digitalspy.com/gaming/retro-corner/feature/a562419/virtual-boy-retrospective-nintendos-disastrous-foray-into-vr/>>. Acesso em jun de 2016

Durante mais de duas décadas a tecnologia de RV pouco evoluiu. Foi então que em 2010, a empresa *Oculus VR* apresentou o *oculus rift*. Este dispositivo foi considerado revolucionário, não somente pelo custo acessível comparado aos outros no mercado, mas também por retomar o assunto da RV num momento em que a tecnologia estava mais apta a oferecer aplicações com qualidade gráfica incomparáveis.

Em 2012 durante a feira mais importante na área de jogos eletrônicos, a *Electronic Entertainment Expo*¹², a empresa *Oculus VR* apresentou o primeiro protótipo do *oculus rift* que mostrado por *John D. Carmack II* (figura 08) rodando o jogo *Doom 3* impressionou o público pela qualidade gráfica oferecida. Ainda em 2012, por meio do *website* de financiamento colaborativo *Kickstarter*, a *Oculus VR* em sua campanha superou a meta de financiamento e atingiu de mais de dois milhões de dólares, podendo assim dar continuidade ao desenvolvimento do *oculus rift* (DAN, 2011).

Diante de todos os fatos relatados, é possível inferir que a tecnologia de RV voltou com mais qualidade e apoio de empresas importantes no setor tecnológico global, o que faz acreditar que agora a RV voltou para cada vez mais estar presente no nosso dia a dia.

¹² A **Electronic Entertainment Expo**, mais conhecida Como **E3**, é uma feira internacional dedicada a jogos eletrônicos. É considerada a mais importante do gênero, por reunir novidades relativas a lançamentos e tendências de mercado de várias empresas do setor. Fonte: <https://pt.wikipedia.org/wiki/Electronic_Entertainment_Expo> acessado em junho de 2016



Figura 08 - John D. Carmack II apresentando o oculus rift na E3 de 2012.
Fonte <<http://onovonerd.blogspot.com.br/2013/02/oculus-rift-o-oculos-que-esta.html>>. acessado em junho de 2016.

2.2 O que é a Realidade Virtual

A RV é um campo de estudo que visa a criação de um sistema que proporcione uma experiência sintética, ilusória ou virtual para o usuário. Estas experiências são denominadas assim, pois são geradas pelo sistema de RV que é composto basicamente de três componentes: um ou mais monitores, sensores que captam os movimentos do usuário, e um computador que é responsável por controlar toda a experiência (KIM, 2005).

Na figura 09, é possível ter uma ideia de um sistema de RV, no qual o usuário utiliza um *head-mounted display* HMD¹³, juntamente com um pedal e um volante, para virtualmente dirigir um carro.

¹³ **Head-mounted display**, ou HMD, é um dispositivo de vídeo usado na cabeça como um capacete e com fones de ouvido, como uma interface através da qual o usuário pode experimentar um ambiente de realidade virtual. É provavelmente, o mais amplo visor visual usado em sistemas de Realidade Virtual.

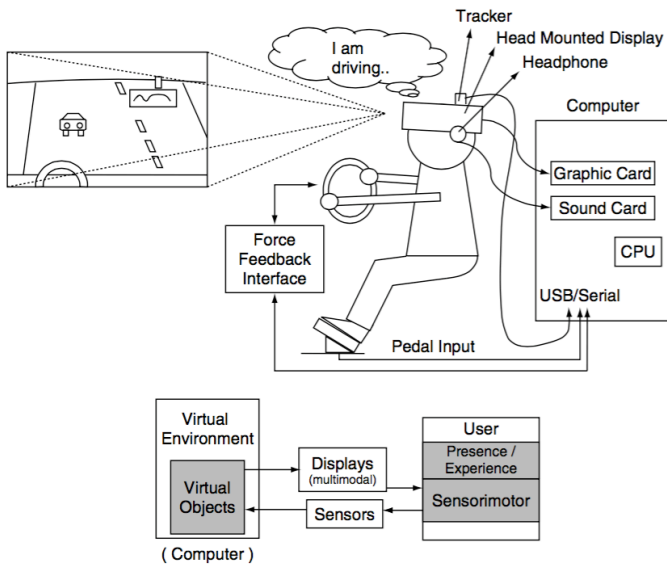


Figura 09 - Exemplo de dispositivos que compõe um sistema de RV.

Fonte: Gerard Jounghyun Kim, pg. 4

A RV é uma tecnologia multidisciplinar. Este recurso tecnológico tem levado a interação humano computador a um nível nunca antes alcançado. Na área médica existem aplicações que possibilitam médicos realizar cirurgias virtuais com o intuito de otimizar o número de procedimentos quando a cirurgia for feita no paciente. Na área de jogos e entretenimento, a RV tem proporcionado novas maneiras de interagir com ambientes tridimensionais, pois o óculos de RV proporciona um grau de imersão que um monitor não consegue.

De fato, a RV é um assunto que tem um potencial enorme, pois a possibilidade de experienciar aplicações criadas por computador e sentir-se completamente imerso em experiências, como se você realmente estivesse lá (presente), abre um precedente enorme para novos modos de interação e comunicação (PARISI, 2015).

2.3 Tipos de sistemas de Realidade Virtual

De acordo com Tori e Kirner (2006, p.8):

A realidade virtual pode ser classificada, em função do senso de presença do usuário, em imersiva ou não-imersiva. A realidade virtual é imersiva, quando o usuário é transportado predominantemente para o domínio da aplicação, através de dispositivos multisensoriais, que capturam seus movimentos e comportamento e reagem a eles (capacete, caverna e seus dispositivos, por exemplo), provocando uma sensação de presença dentro do mundo virtual. A realidade virtual é categorizada como não-imersiva, quando o usuário é transportado parcialmente ao mundo virtual, através de uma janela (monitor ou projeção, por exemplo), mas continua a sentir-se predominantemente no mundo real.

Existe também a Realidade Aumentada. Esta tecnologia possibilita a projeção de conteúdos virtuais no mundo físico o qual se faz necessário o uso de um celular ou *tablet* para reproduzir o conteúdo. Um bom exemplo do uso desta tecnologia, é o aplicativo Pokémon Go, que por meio do uso da câmera do celular, projeta no mundo real personagens da nintendo, as quais o usuário precisa capturar para tornar-se treinador (Figura 10).



Figura 10 - Aplicativo PokemonGo.

Fonte: <http://exame.abril.com.br/tecnologia/noticias/o-que-e-realidade-aumentada-chave-do-sucesso-de-pokemon-go>> Acesso em outubro de 2016

2.4 Características da Realidade Virtual imersiva

A ideia principal da RV é proporcionar experiências ilusórias, para nos fazer sentirmos em lugares que na verdade não estamos.

De uma forma geral, os sistemas de realidade virtual combinam a imersão, a interactividade, o realismo e o envolvimento do utilizador, de modo a proporcionar experiências com um ambiente virtual que gera sensações mais ou menos intensas de presença. Os sistemas de realidade virtual podem, pois, ser caracterizados através dos graus de imersão, interactividade, realismo e envolvimento que proporcionam ao utilizador (RIBEIRO, 2007, p 393).

Geralmente as interfaces de aplicações deste género oferecem ao usuário diversos graus de interação e seis graus de liberdade:

1. Para frente/ para trás
2. Para cima/ para baixo
3. Da direita/ para a esquerda
4. Inclinação para cima/ e para baixo
5. Rotação para a direita/ e para a esquerda
6. Inclinação para a direita/ e para a esquerda

2.5 Dispositivos de Realidade Virtual

Os dispositivos de RV estão em constante desenvolvimento, e por serem de tecnologia de ponta, ainda são considerados caros e inacessível para a grande massa de consumidores. Existe uma grande quantidade de dispositivos para RV, alguns podendo custar mais de quatrocentos mil dólares, como é o caso do *F-35 Gen III Helmet Mounted Display System*, figura 11. Este dispositivo de RV tem inúmeras vantagens, como o fato de não precisar ser ligado a um computador para funcionar, pois todo o sistema e *hardware* fica dentro do capacete.



Figura 11 - Capacete de RV F-35 Gen III (HMDS).

Fonte: < <https://www.rockwellcollins.com>>. Acesso em jun de 2016

Entretanto, pode-se destacar outro dispositivo que também oferece qualidade para experiências com RV como é o caso do *HTC vive* (figura 12), desenvolvido pela empresa HTC¹⁴. O *vive* é um concorrente direto do *Oculus Rift*, existem apenas algumas diferenças em questão ao desenvolvimento de aplicações mas, no quesito qualidade de experiência para o usuário ambos são equivalentes.

¹⁴ **HTC Corporation**, formalmente **High-Tech Computer Corporation**, é uma empresa multinacional taiwanêsa de smartphones e tablets com sede na cidade de Nova Taipé. Fundada em 1997, HTC iniciou como fabricante e desenvolvedora equipamentos, desenhando e fabricando dispositivos como celulares, celulares com touchscreen e PDAs e Brew MP.



Figura 12 - HTC vive.

Fonte: <<https://www.htcvive.com/us/>>. Acesso em jun de 2016

Diante de tantas empresas investindo na RV, a *Google* não poderia ficar de fora, e em 2014 lançou o *google cardboard* (figura 13). Diferindo dos outros dois dispositivos apresentados anteriormente este é um suporte feito de papelão que apenas serve para colocar um *smartphone* dentro, o qual é responsável por executar as aplicações de RV como jogos, exibição de vídeos imersivos, etc. A *Google* disponibiliza esse kit de forma gratuita (figura 14) e incentiva as pessoas a construírem o próprio óculos de RV.



Figura 13 - Google Cardboard

Fonte: <<https://vr.google.com/cardboard/>>. Acesso em junho de 2016

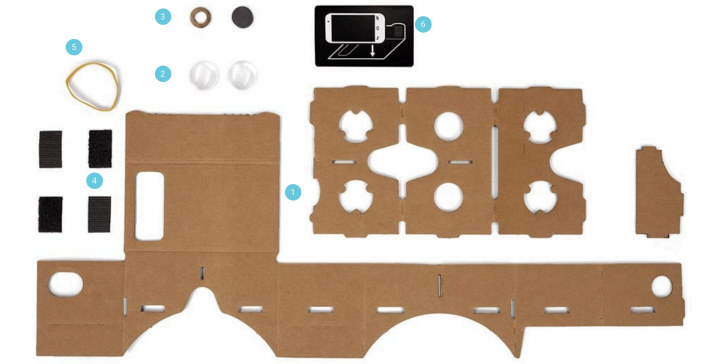


Figura 14 - kit de construção do google cardboard

Fonte: < <https://vr.google.com/cardboard/get-cardboard/>>. Acesso jun 2016.

Este PCC foi desenvolvido em parceria com o DesignLab¹⁵, que no momento de desenvolvimento deste projeto tinha um *oculus rift DK2* (figura 15) à disposição. O fato de existir mais material técnico disponível favoreceu para a escolha deste dispositivo de RV.



Figura 15 - Oculus Rift DK2.

Fonte: <<http://www.vrbites.com/reviews/oculus-rift-dk2-the-world-is-ready-for-virtual-reality/>>. Acesso em outubro de 2016

¹⁵ O **DesignLab** é um laboratório vinculado ao programa de Graduação e Pós-Graduação (Mestrado e Doutorado) em Design da Universidade Federal de Santa Catarina. O objetivo é realizar projetos de pesquisa com ênfase em tecnologia, bem como a extensão desse estudo para o desenvolvimento de produtos de caráter social, econômico e cultural.

2.6 Criação de mundos virtuais

Dentro do processo de criação de uma aplicação de RV, existe a etapa de construção dos mundos virtuais, que podem ser representações fiéis ao mundo físico ou não. Estas etapas englobam: modelagem 3D, texturização, exportar e integrar os modelos no programa de RV. Esta etapa é muito semelhante ao processo de produção de um ambiente 3D para jogos, já que recomendações como baixo número de polígonos, o uso de variações do mesmo objeto com menos geometria *Level of detail*¹⁶ (figura 16), também se aplicam. Estas recomendações são necessárias, pois o processo de renderização¹⁷ em aplicações de RV assim como nos jogos, é feito em tempo real. Portanto tudo deve ser otimizado ao máximo, para que a performance da aplicação não seja comprometida e consequentemente a experiência do usuário.

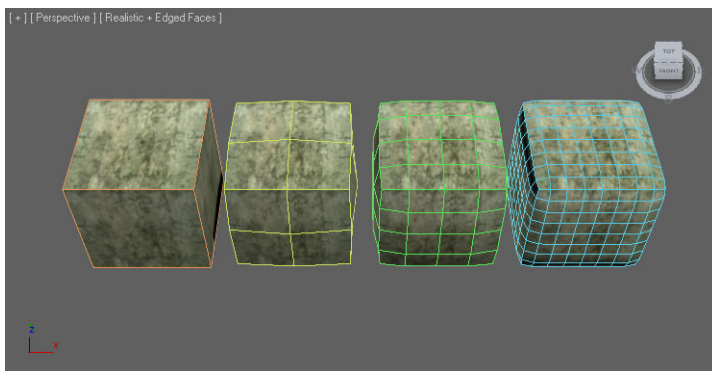


Figura 16 - Níveis de resolução em um objeto 3D.

Fonte: <<https://udn.epicgames.com/Three/FBXStaticMeshPipeline.html>>. Acesso em jun 2016

¹⁶ **Level of Detail** (Português: **Nível de detalhe**) é uma técnica que permite várias representações para um objecto gráfico, que vão sendo activadas de acordo com as diferentes distâncias a que o utilizador se encontra. Fonte: <https://pt.wikipedia.org/wiki/nivel_de_detalhe> acesso em jun de 2016.

¹⁷ **Renderização** é o processo pelo qual se pode obter o produto final de um processamento digital qualquer. Este processo aplica-se essencialmente em programas de modelagem 2D e 3D. Fonte <<https://pt.wikipedia.org/wiki/Renderização>> acessado em jun de 2016.

2.6.1 Programa de modelagem 3D

A partir desta etapa deve-se escolher um programa de modelagem 3D para a construção do mundo virtual. Existe uma gama de programas que podem ser usados para esta finalidade, muitos dos quais são usados para produzir, filmes, jogos e animações em 3D, como é o caso do *3ds Max* (figura 17) programa produzido pela *Autodesk*¹⁸. Este programa é muito utilizado na indústria de desenvolvimento de jogos, e consequentemente pode ser usado para produzir os modelos 3D para uma aplicação de RV.



Figura 17 - Programa 3D Max.

Fonte: <<http://www.autodesk.com.br/products/3ds-max/>>. Acesso em jun 2016.

Existem outros programas como *Maya* (figura 18), *Modo* (figura 19) todos com plena capacidade de gerar conteúdos para RV. Vale salientar que o programa para modelagem 3D é uma questão de escolha, pois praticamente todos os disponíveis no mercado são capazes de produzir conteúdo para aplicações de RV.

¹⁸ **Autodesk, Inc.** é uma empresa de programa de design e de conteúdo digital.
Fonte: <<https://pt.wikipedia.org/wiki/Autodesk>> acesso em jun 2016.

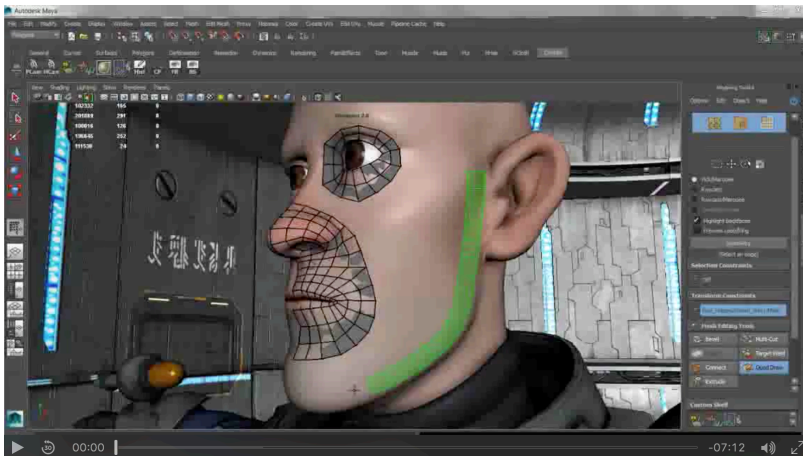


Figura 18 - Programa Maya.

Fonte: <<http://www.autodesk.com.br/products/maya/>> Acesso em jun 2016.

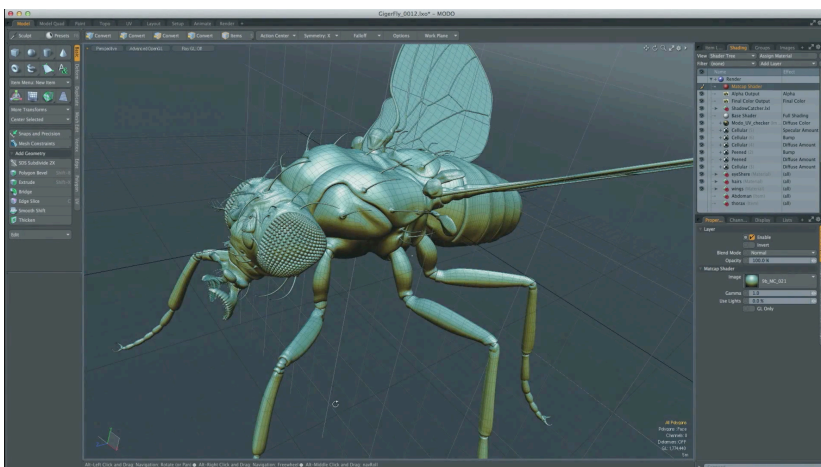


Figura 19 - Programa Modo.

Fonte: <<https://www.thefoundry.co.uk/products/modo/features/#Modeling>>. Acesso em jun 2016.

Para quem deseja uma opção gratuita, existe o *Blender 3D*¹⁹ (figura 20), um programa tão poderoso quando os demais aqui destacados, só que sob a licença livre, o que possibilita o uso acadêmico ou profissional sem a necessidade de pagar pelo programa.

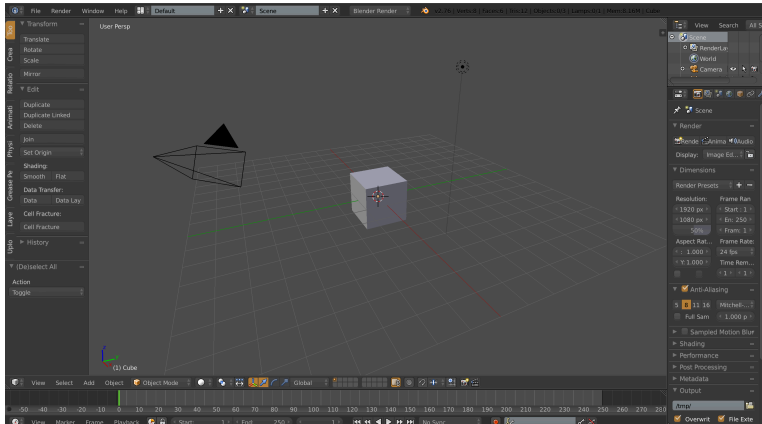


Figura 20 - Imagem da interface do programa Blender 3D.

Fonte: próprio autor

Como mencionado anteriormente, escolha do programa é uma questão pessoal quando se trabalha sozinho. O profissional deve experimentar e concluir qual torna-se mais eficaz no dia a dia, porém a indústria acaba definindo certos programas como sendo “perfeitos” para tal área, surgindo assim a necessidade de o profissional saber trabalhar com mais de um programa. Para este projeto foi escolhido o programa *Blender*, pelo fato de ser um software livre e pela experiência prévia do autor neste programa.

¹⁹ **Blender**, também conhecido como **blender3d**, é um programa de computador de código aberto, desenvolvido pela Blender Foundation, para modelagem, animação, texturização, composição, renderização, edição de vídeo e criação de aplicações interativas em 3D, tais como jogos, apresentações e outros, através de seu motor de jogo integrado, o Blender Game Engine. Fonte: < <https://www.blender.org> > Acesso em jun 2016.

2.6.2 Modelagem 3D (box modeling)

Existem diversas metodologias de modelagem 3D, e uma delas se chama *box modeling*, ou modelagem por meio de um cubo. No *Blender* ou em outros programas de modelagem 3D, os modelos são compostos por vértices (*Vertices*), arestas (*edges*), triângulos (*triangles*) e polígonos (*polygon*) ou faces (figura 21).

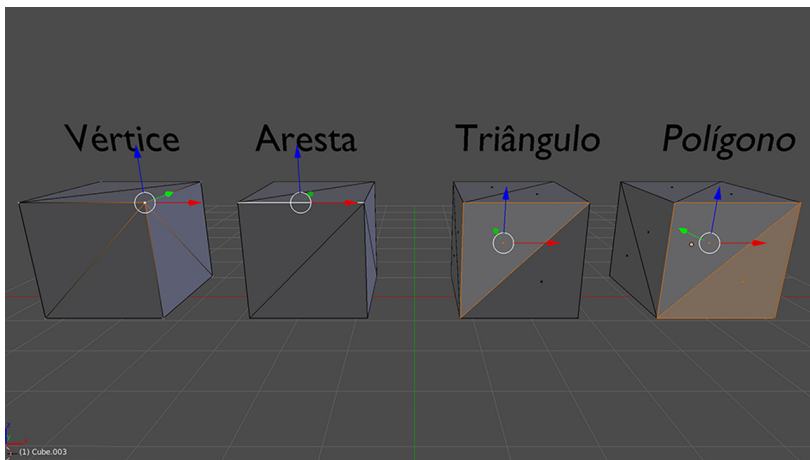


Figura 21 - Da esquerda para a direita, exemplifica-se vértice, aresta, triângulo e polígono ou face.

Fonte: próprio autor

Nesta técnica, os modelos 3D são iniciados a partir de cubos que após operações de extrusão, escala e rotação, adicionam outros detalhes e mais complexidade ao modelo 3D (figura 22).

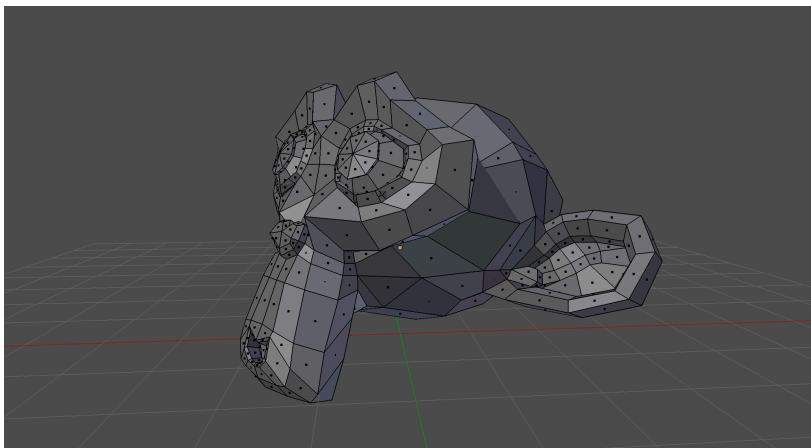


Figura 22 - Exemplo de modelo 3D criado com a técnica box modeling.
Fonte: próprio autor

2.6.3 Mapeamento 3D

Após a etapa de modelagem 3D, os modelos precisam ser mapeados. Este processo se caracteriza pela criação de texturas (imagens) que serão aplicadas aos materiais para proporcionar uma aparência mais realista ao modelo 3D, se esta for a intenção.

Este processo é facilitado através de programas de edição de imagens, como por exemplo o *adobe photo shop*. Para iniciar este processo, os modelos 3D precisam ter a sua malha de polígonos planificada, em um processo chamado *uvunwrap* (figura 23).

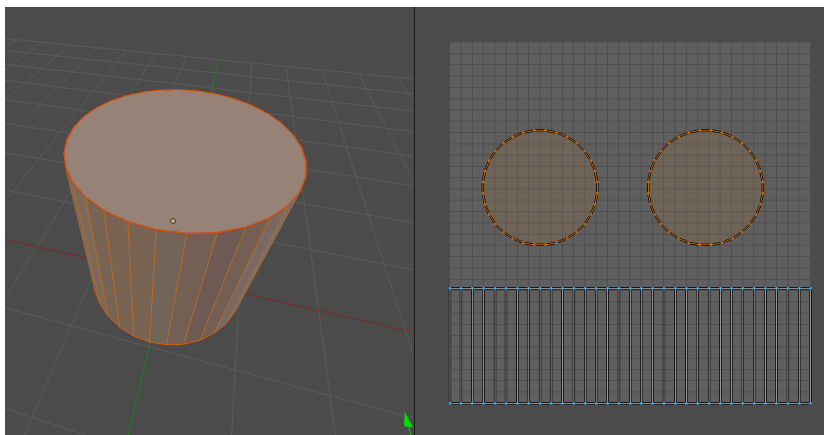


Figura 23 - Na esquerda o modelo de um cilindro, na direita a malha poligonal do cilindro planificada.

Fonte: próprio autor

Esta forma planificada é exportada para o *photoshop* e servirá como base para a criação de uma textura de metal por exemplo (figura 24). Outra possibilidade, é usar uma textura sem emendas, chamadas de *seamless*. Estas texturas podem ser multiplicadas infinitamente, e no resultado final o modelo 3D não apresenta sinais de emendas.

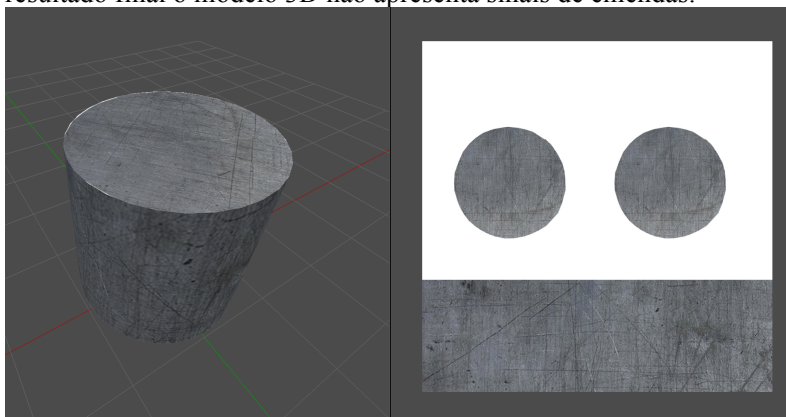


Figura 24 - Nesta imagem após a edição no *photoshop*, criou-se uma textura de concreto tendo como base a malha planificada anteriormente.

Fonte: próprio autor

Neste caso foi criada uma textura para dar cor ao modelo 3D porém, existem texturas específicas que controlam brilho, sensação de relevo e transparência. Desta forma o processo continua até que todos os modelos estejam texturizados e possam ser exportados para o programa de RV.

2.6.4 Salvar e exportar modelos

Após finalizar a criação dos modelos 3D, é preciso prepará-los para salvar e exportar no formato Autodesk FBX²⁰ e importar programa de RV. Dependendo da necessidade do projeto, certos modelos 3D podem ser animados dentro do programa escolhido e serem exportados com estas animações, opção possível de ser feita com o formato do arquivo FBX. Neste projeto apenas serão exportados modelos 3D estáticos, ou seja, não serão animados no *Blender*, mas sim no programa de RV.

Todos os modelos devem ter o ponto de pivot acertados antes do processo de exportação. O ponto pivot é o eixo central do objeto de onde todas as deformações, como escala, rotação e movimentação partirão (figura 25). Após o eixo central do objeto ser definido, este deve estar posicionado nas coordenadas 0,0,0 no mundo virtual (figura 26).

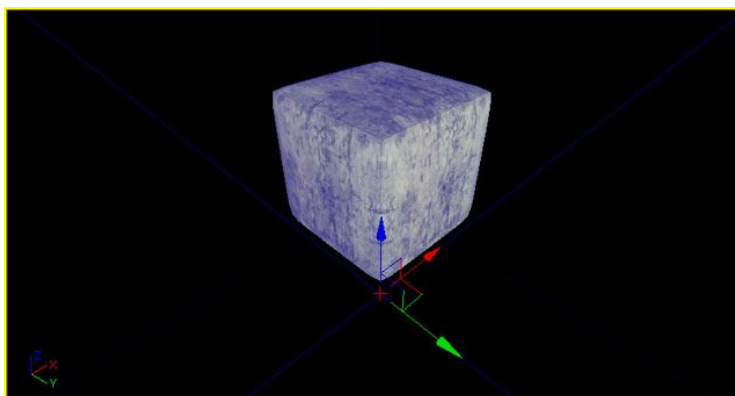


Figura 25 - Ponto de pivot do objeto.

Fonte: <https://docs.unrealengine.com/>. Acesso em jun de 2016

²⁰ O **Autodesk FBX**, formato de arquivo 3D, padrão aberto e independente de plataforma, permite o acesso a todo conteúdo produzido em qualquer pacote de programa suportado.

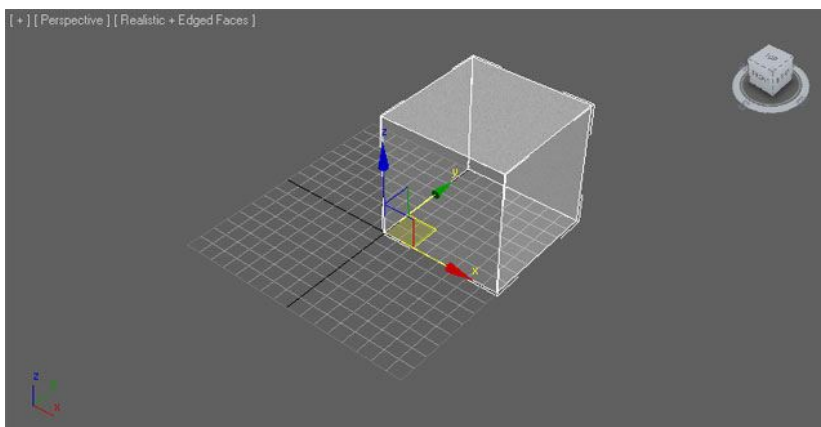


Figura 26 - Objeto posicionado no centro do mundo virtual.
Fonte: <https://docs.unrealengine.com/>. Acesso em jun de 2016

Como já explicado anteriormente, no mundo virtual os modelos 3D são compostos por vértices, arestas, triângulos, e polígonos ou faces. Antes de o objeto ser exportado é necessário converter todas os polígonos para triângulos, caso esta operação não seja feita, será possível optar na hora de exportar o objeto, ou próprio programa de RV fará esta operação automaticamente (figura 27). Após estes ajustes o modelo 3D está pronto para ser salvo no formato FBX.

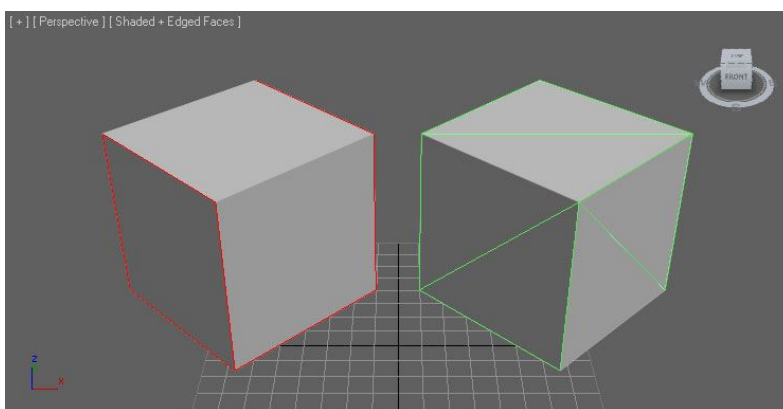


Figura 27 - Modelo 3D da esquerda antes do processo de triangulação, e da direita já somente com triângulos.

Fonte: https://docs.unrealengine.com. Acesso em jun de 2016

2.6.5 Integração com o motor de jogos Unreal Engine 4

Como já abordado na introdução deste projeto, existem diversos programas que possibilitam o desenvolvimento de aplicações para RV. Neste projeto será usada a *Unreal Engine 4*²¹ (figura 28), um dos motores de jogos mais potentes do mercado.

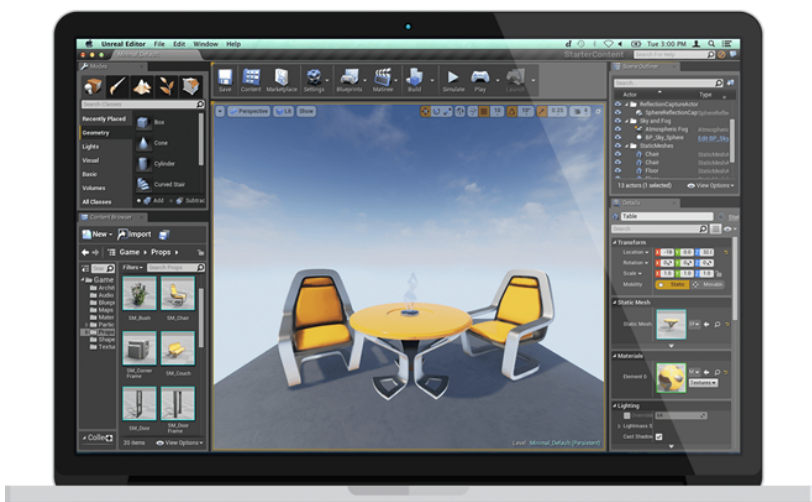


Figura 28 - Interface da UE4.

Fonte: <https://www.unrealengine.com/what-is-unreal-engine-4>. Acesso em jun de 2016

A UE4 foi criada por desenvolvedores de jogos e pensada para artistas, pois oferece uma interface de fácil entendimento e aprendizado. Outra vantagem é a versatilidade no quesito programação de interatividades, já que oferece um formato de programação visual chamado *blue print*²² (figura 29), que baseado em eventos e conexões entre blocos de lógica, possibilita profissionais que não são programadores criar interatividades complexas.

²¹ Unreal Engine 4 (UE4). Fonte: <www.epicgames.com> Acesso outubro de 2016

²² Blue Print – Formato de programação visual desenvolvido pela epic games para tornar o processo de programação de interatividades mais intuitivo. Fonte: <www.epicgames.com/blueprint> Acesso em outubro de 2016

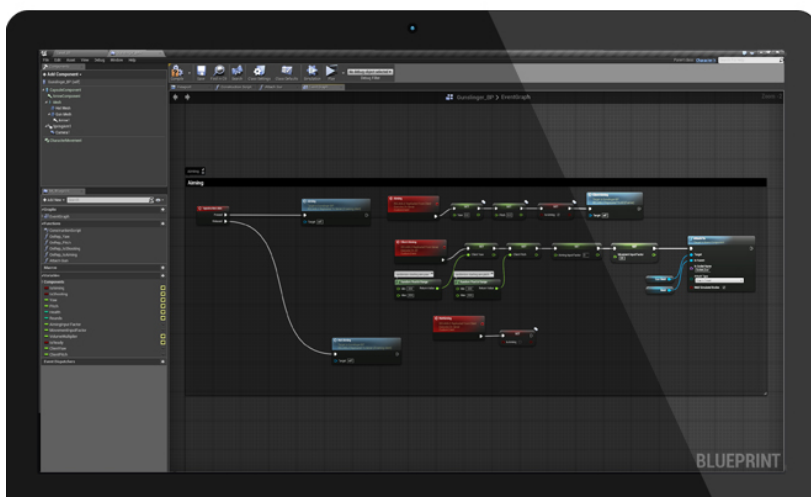


Figura 29 - Exemplo de programação visual Blue Print.

Fonte: <<https://www.unrealengine.com/what-is-unreal-engine-4>>. Acesso em jun de 2016

Para importar os modelos 3D, o processo é bem similar a qualquer outro programa 3D. Conforme a figura 30, basta selecionar a opção *file* e depois *import*, navegar até a pasta em que os modelos 3D foram salvos, selecionar e importar.

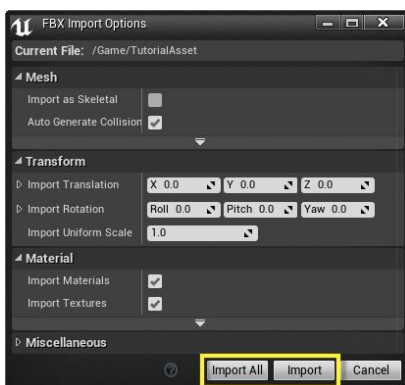


Figura 30 - Imagem da tela de importação de modelos 3D na UE4.

Fonte: <<https://www.unrealengine.com/what-is-unreal-engine-4>>. Acesso em jun de 2016

Neste caso o modelo 3D foi importado para a pasta raiz do diretório principal da UE4 (figura 31), porém é recomendado que desde o início seja respeitado uma hierarquia de pastas, para facilitar a navegação e identificação dos elementos que compõe toda a aplicação de RV.

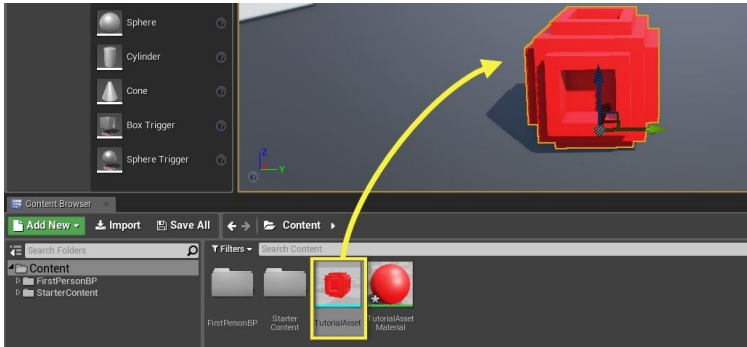


Figura 31 - Modelo 3D importado na pasta principal do projeto.

Fonte: <<https://www.unrealengine.com/what-is-unreal-engine-4>>. Acesso em jun de 2016

Após o modelo 3D ser importado para a UE4, é possível criar o material que dará o visual desejado. Na figura 32, é possível notar a esfera representando o material criado, o cubo com o material aplicado e a textura de tijolos criada no programa de edição de imagens.

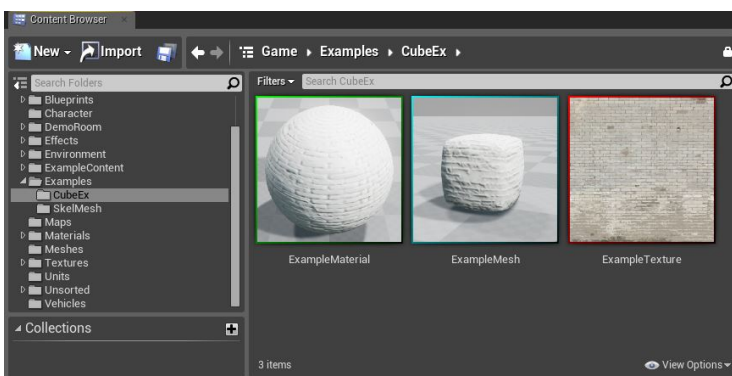


Figura 32 - Modelo 3D com material aplicado e textura usada para dar a aparência desejada.

Fonte: <<https://www.unrealengine.com/what-is-unreal-engine-4>>. Acesso em jun de 2016

3 Desenvolvimento

3.1 Escolha de referências

Para este projeto foi utilizado o site *Pinterest*²³, para a escolha de referências de ambientes e mobiliário para representação em 3D e consequentemente em RV. Quando se trata de ambientes ou projetos de arquitetura, é importante ter além de imagens o próprio projeto arquitetônico que possibilitará uma representação virtual mais fiel em questão a medidas e proporção. Entretanto, é possível fazer a modelagem 3D baseando-se em imagens, apenas será preciso ajustar a escala do que foi modelado no mundo 3D da melhor forma possível.

O site *Pinterest*, é considerado um catálogo mundial de ideias, pois é onde muitos designers e arquitetos publicam os seus projetos. Para tanto, foi escolhido como principal fonte de referências e inspiração para este PCC. Dentre as milhares publicações foi escolhida uma na qual tinha referências de um apartamento completo, com os vários ambientes e mostrando bons detalhes de decoração assim como mobiliários e utensílios. (Figuras 33, 34, 35, 36 e 37)



Figura 33 - Referência quarto.

Fonte: <<https://br.pinterest.com/pin/493284965410968700/>> acesso em agosto de 2016

²³ Pinterest – Maior catálogo mundial de ideias. Fonte: <www.pinterest.com> Acesso em outubro de 2016



Figura 34 - Referência da sala de estar e cozinha.

Fonte: <<https://br.pinterest.com/pin/493284965410968700/>> acesso em agosto de 2016



Figura 35 - Referência do banheiro e lavanderia.

Fonte: <<https://br.pinterest.com/pin/493284965410968700/>> Acesso em agosto de 2016



Figura 36 - Referência da sacada.

Fonte: <<https://br.pinterest.com/pin/493284965410968700/>> Acesso em agosto de 2016



Figura 37 - Referência área de jantar.

Fonte: <<https://br.pinterest.com/pin/493284965410968700/>> Acesso em agosto de 2016

3.2 Modelagem 3D

Por meio das imagens escolhidas é possível iniciar a modelagem 3D da estrutura do apartamento. Neste caso, como não existe a disposição o projeto arquitetônico deste ambiente, será feita uma estimativa de medidas e proporção para chegar o mais próximo possível de uma representação ideal. As imagens deste ambiente foram escolhidas para servirem como inspiração, tanto de estrutura como decoração, o que não significa que a representação será totalmente fiel, deste modo é possível que algumas adaptações sejam feitas.

Com o intuito de seguir uma proporção métrica para este ambiente, no programa Blender foi necessário configurar a unidade de medidas. A figura 38 mostra a barra de propriedades na área cena, a configuração das unidades de representação para centímetros. Este procedimento deve ser seguido caso a modelagem 3D esteja sendo feita em outros programas, pois a UE4 interpreta a escala em centímetros, deste modo evita-se o processo de escalonamento dos modelos 3D na hora de exportar.

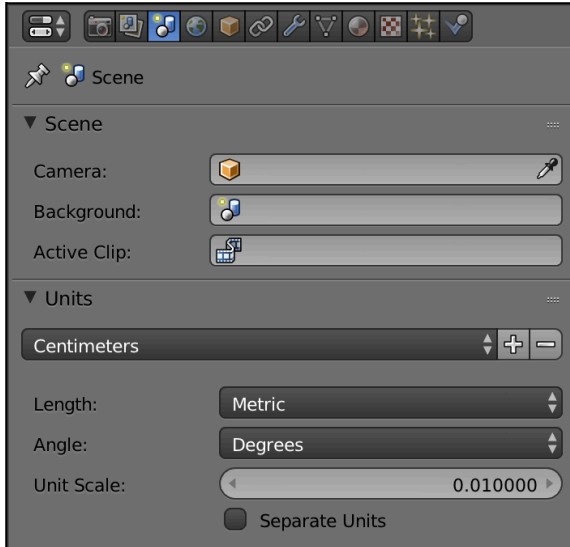


Figura 38 - Unidade de medidas em centímetros.
Fonte: próprio autor

Por meio de um cubo a modelagem foi iniciada, e após operações de extrusão, movimentação de faces, adição de arestas e escalonamento pode-se chegar no resultado da figura 39.

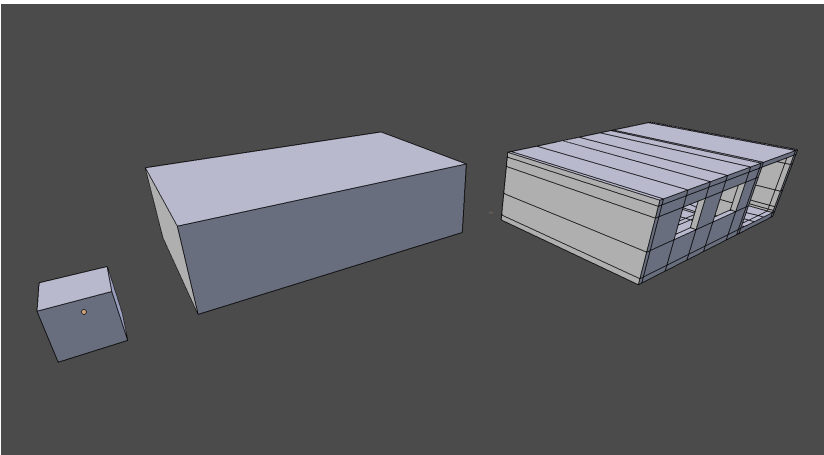


Figura 39 - Evolução da modelagem das paredes.
Fonte: próprio autor

Para evitar problemas com iluminação e geração de sobras na UE4, foi necessário separar o modelo em duas partes, paredes internas e externas (Figura 40)

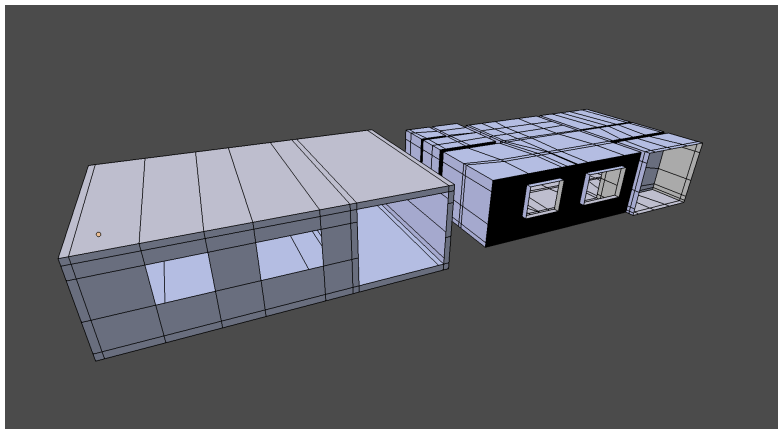


Figura 40 - Faces externas separadas das internas.

Fonte: próprio autor

Após a modelagem 3D da estrutura do apartamento, foi possível dar continuidade nos demais objetos. Da mesma forma, baseando-se na técnica de modelagem a partir de um cubo, foi possível criar os demais modelos e compor os ambientes. (Figura 41 e 42)



Figura 41 - Vista 01 dos ambientes com demais objetos.

Fonte: Próprio autor



Figura 42 - Vista 02 dos ambientes com demais objetos.

Fonte: Próprio autor

3.3 Mapeamento 3D

Esta é uma das etapas mais importantes e complexas na criação de modelos 3D para visualização em tempo real e RV. Cada modelo 3D pode ter diversos canais de mapeamento, porém um canal deve ser criado especificamente para guardar as informações de luz e sombra que o programa de RV calcula e aplica automaticamente, chamado *lightmaps*. Recomenda-se que todas as texturas a serem aplicadas em modelos 3D possuam valores em pixel com números que sejam potência de 2 como, 128x128, 256x256, 512x512 ou mantendo esses valores para criar formatos retangulares como 128x256, 512x1024, por exemplo. Os *lightmaps* devem ser pensados cuidadosamente, pois por questão de otimização e desempenho da aplicação de RV, devem ter o menor número possível de pixels (GAMES, 2016).

No caso da estrutura do apartamento, optou-se por dividir as paredes em externas e internas, como já mencionado no capítulo anterior, Esta decisão foi tomada devido ao fato de que a parte externa do apartamento não vai ser visualizada pelo usuário, podendo assim ter uma resolução de *lightmap* menor que as paredes internas. Caso contrário, se a estrutura fosse formada por um só modelo, seria utilizada uma resolução de *lightmap* maior que a necessária.

Durante esta etapa deve-se criar e aplicar os materiais nas diferentes partes do modelo 3D, se necessário. Se um modelo 3D tiver o chão com material de madeira, e paredes brancas, cada parte deve ter um material diferente aplicado. Estes materiais criados no programa de modelagem 3D servem apenas de base para os materiais definitivos que serão criados e aplicados no programa de RV. Na figura 41, é possível ter uma ideia de como os materiais devem ser configurados e aplicados no modelo 3D antes de ser exportado. Após aplicados os materiais, pode-se passar para a criação do canal de mapeamento destinado ao *lightmap*.

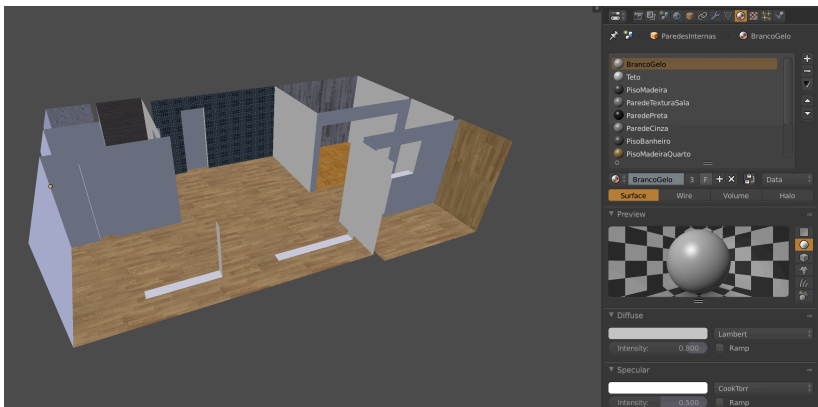


Figura 43 - Diferentes materiais aplicados no modelo 3D.

Fonte: Próprio autor

Na figura 44, é possível ver o modelo 3D das paredes internas com os dois canais de mapeamento configurados. O canal número um do mapeamento, foi configurado de acordo com a necessidade de cada textura de cor, já o canal número dois foi configurado para ser usado no processo de *lightmap*, portando a malha planificada deve estar dentro da área de resolução, com um afastamento de no mínimo de pixels para que não fiquem sobrepostas.

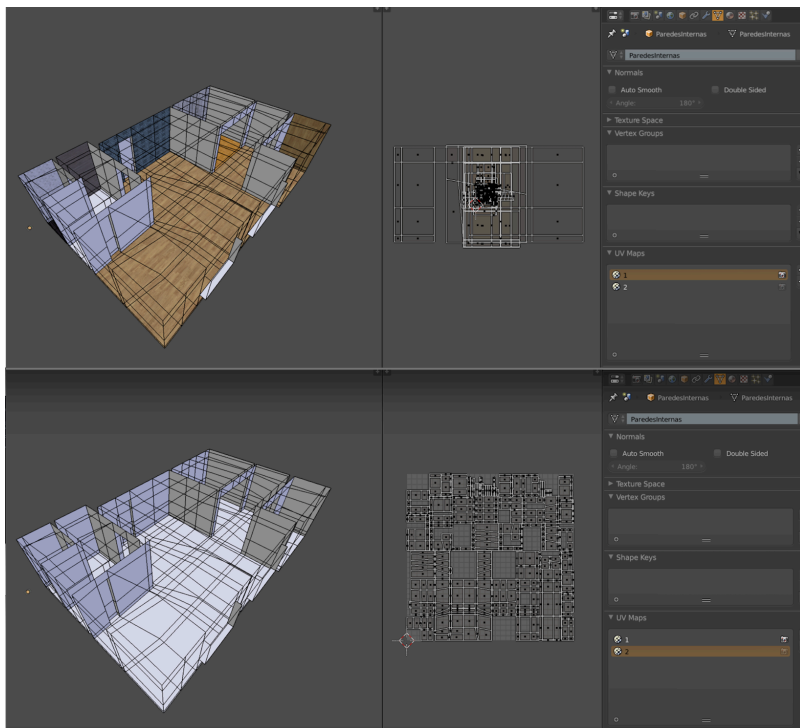


Figura 44 - Modelo 3D mostrando canais de mapeamento.

Fonte: Próprio autor

Este processo de planificação da malha do modelo 3D, configuração dos canais de mapeamento, e aplicação de materiais se estende para todos os modelos criados (figura 44).

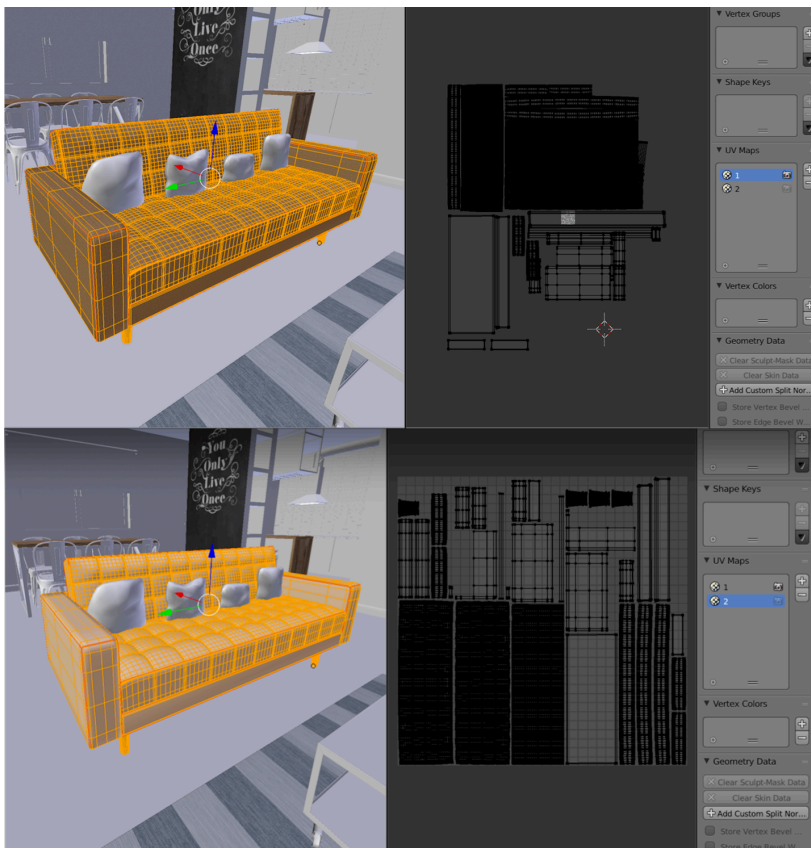


Figura 45 - Exemplo de mapeamento do sofá.
Fonte: Próprio autor

Este processo deve ser replicado aos demais, para poderem ser exportados e integrados com a UE4.

3.4 Exportar modelos 3D

Após finalizar todas as etapas anteriores, os modelos 3D estão prontos para serem exportados. Como já abordado anteriormente o processo de exportação é simples, porém necessita de alguns ajustes. Primeiramente o modelo 3D precisa ter configurado um ponto central que a partir dele o modelo será movimentado, rotacionado ou escalonado no programa de RV. Na figura 46, é possível ver o criado mudo com o ponto central configurado. Neste caso o modelo 3D está posicionado dentro da cena do mundo virtual do programa Blender, e seu ponto central está localizado no canto inferior em uma das rodas.



Figura 46 - Modelo 3D posicionado na cena do mundo virtual do blender.

Fonte: Próprio autor

Se o modelo 3D for exportado a partir deste ponto, ele carregará as informações de posição escala e rotação, e quando for importado no programa de RV, o ponto central será automaticamente alterado para a posição central do mundo virtual do *Blender*, e não o ponto que foi configurado manualmente. Para que isto não aconteça, é necessário mover o modelo 3D para o ponto central do mundo virtual, como mostra a figura 47.

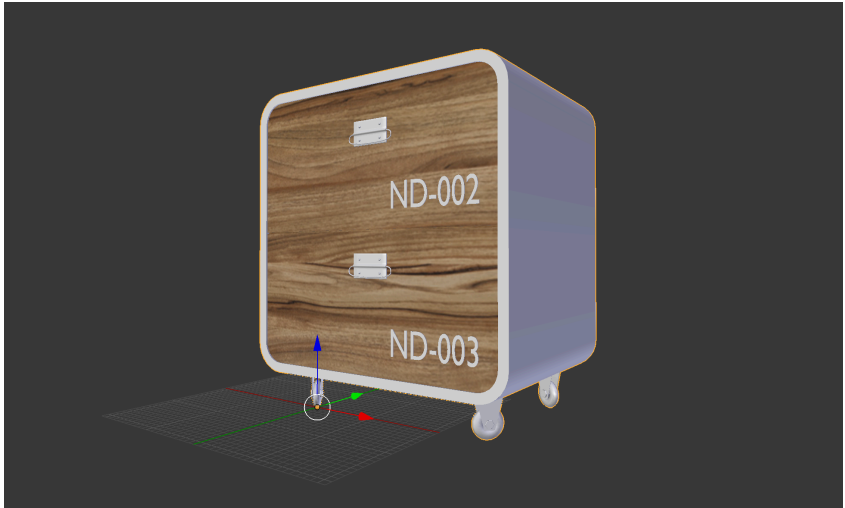


Figura 47 - Modelo 3D posicionado no centro do mundo virtual do blender.

Fonte: Próprio autor

Com o modelo 3D selecionado, basta ir em exportar e escolher a opção fbx, conforme a figura 48.

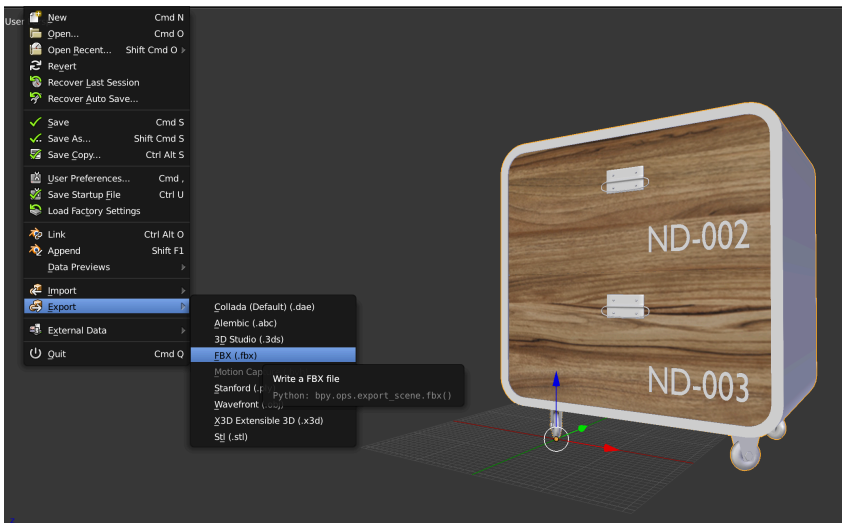


Figura 48 - Selecionar opção fbx para exportar.

Fonte: Próprio autor

Existem dois formatos de exportação em fbx, modelos 3D animados ou estáticos. Neste caso todos os modelos 3D serão estáticos, as animações se necessárias serão feitas diretamente na UE4. Deste modo conforme a figura 49, deve-se selecionar a opção para exportar somente o que estiver selecionado, especificar que é um modelo do tipo *mesh*, e que a opção de suavização da malha será a partir das faces do modelo 3D. Por fim deve-se escolher a pasta de destino a qual o modelo 3D será armazenado e salvar. Este processo deve ser feito para todos os modelos 3D da cena.

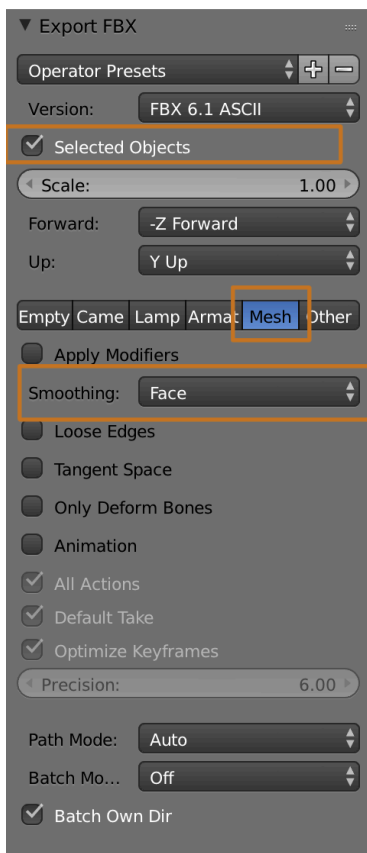


Figura 49 - Configuração para exportar em fbx.
Fonte: Próprio autor

3.5 Importar na Unreal Engine 4

Antes de iniciar o processo de integração com a UE4, é necessário criar uma conta em seu site para ter acesso ao arquivo de instalação para ambas as plataformas: *windows* e *Mac*. A UE4, vem ganhando adeptos tanto de grandes desenvolvedoras de jogos quanto artistas e designers, pelo fato de ser um motor de jogos muito completo e por oferecer material de apoio e uma vasta comunidade online que auxilia durante o desenvolvimento de qualquer aplicação (figura 50).

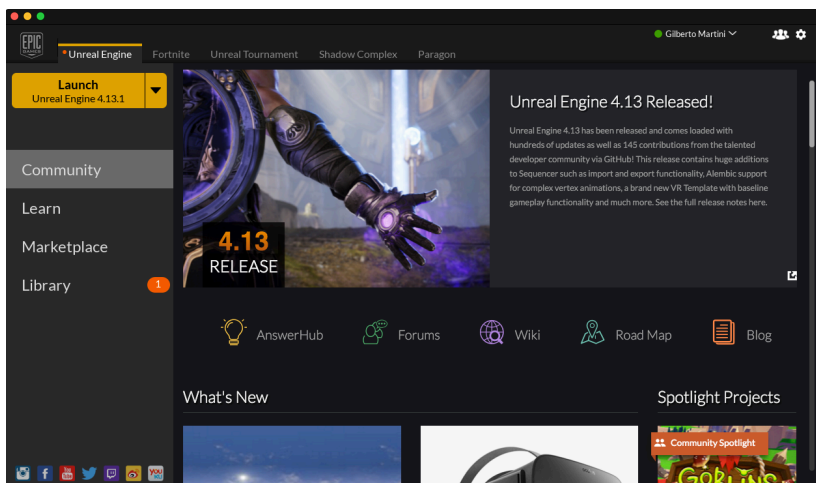


Figura 50 - Página inicial do aplicativo *Epic Games Launcher*.

Fonte: Próprio autor

Outro diferencial em usar a UE4 é o fato de poder iniciar um projeto baseado em um *template*, que nada mais é do que um modelo de mundo virtual pré configurado. Existem diversos tipos de *template*, desde jogos em primeira e terceira pessoa, veículos, otimizados para aparelhos de celular, etc. Para aplicações de RV é necessário escolher o *template* de primeira pessoa (figura 51), que já vem configurado com um personagem e mundo virtual prontos para uso. Como este *template* é baseado em um jogo de tiro em primeira pessoa, é necessário desabilitar a arma virtual pois as outras configurações já servem para VR.

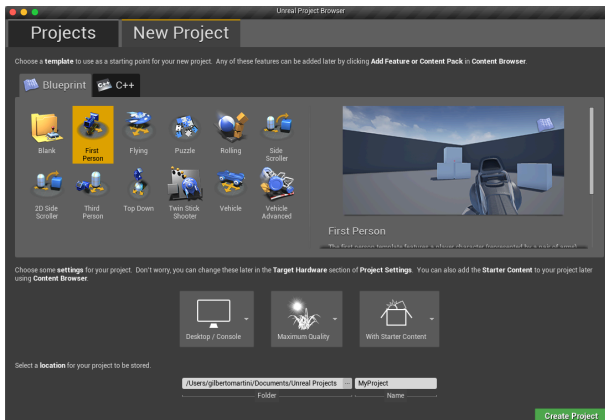


Figura 51 - Modelos de templates oferecidos pela UE4.
 Fonte: Próprio autor

Durante o desenvolvimento de aplicações de RV, assim como qualquer outras, a organização dos diretórios ou pastas são importantes. Isto facilita pois na medida em que o desenvolvimento do projeto torna-se complexo e com muitas pastas, ter uma organização básica facilita para que qualquer colaborador rapidamente localize-se no diretório principal e assim possa contribuir com a produção. Conforme a figura 52, após escolher o template de primeira pessoa ten-se um mundo virtual já configurado com céu e nuvens, e na aba *content* foi criado a estrutura principal de pastas que vão manter o projeto organizado.

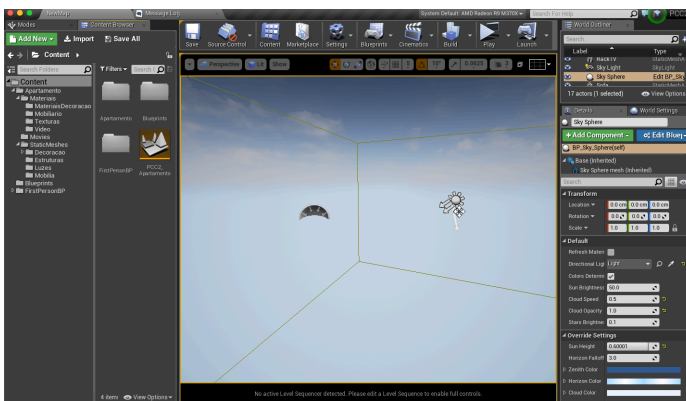


Figura 52 – Estrutura de pastas e template de primeira pessoa.
 Fonte: Próprio autor

O processo de importação dos modelos 3D foi feito em etapas. No caso deste projeto como será representado o interior de um apartamento, iniciou-se pela importação das paredes, janelas, portas e outros modelos que juntos formarão a estrutura principal (figura 53).

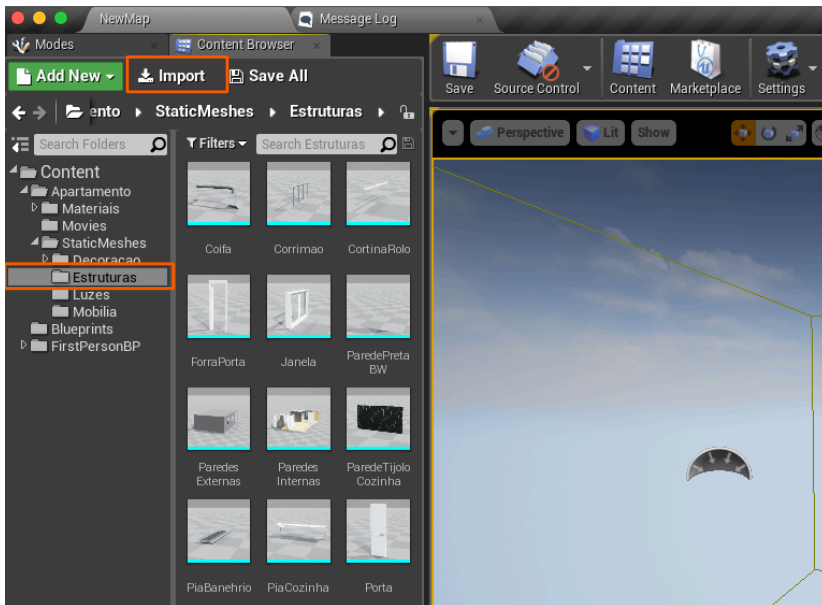


Figura 53 – Arquivos importados na pasta estruturas.

Fonte: Próprio autor

Este processo de importação foi replicado até que todos os arquivos estivessem nas devidas pastas. Uma vez que todos os arquivos foram importados, iniciou-se a construção do apartamento. Os modelos 3D podem ser arrastados para o mundo virtual da UE4 individualmente ou em conjuntos. No caso das paredes, apesar de no processo de exportação terem sido separadas em internas e externas, agora podem ser integradas em conjunto, já que precisam estar perfeitamente alinhadas. Os outros modelos foram inseridos um por um, de modo que a localização de cada um precisou ser ajustada.

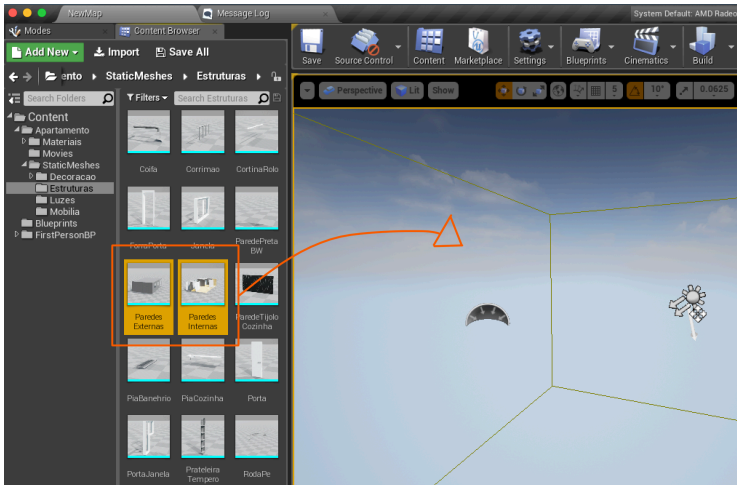


Figura 54 – Selecionar e arrastar modelos para o mundo virtual da UE4.
 Fonte: Próprio autor

Ao arrastar os modelos para a cena, a localização final será determinada pelo cursor do mouse. Ao selecionar o modelo 3D, os eixos x, y e z representados pelas cores vermelho, verde e azul aparecem, permitindo assim move-lo para onde quiser, para mudar para escala ou rotação basta selecionar na barra de ferramentas a opção desejada. (Figura 55)

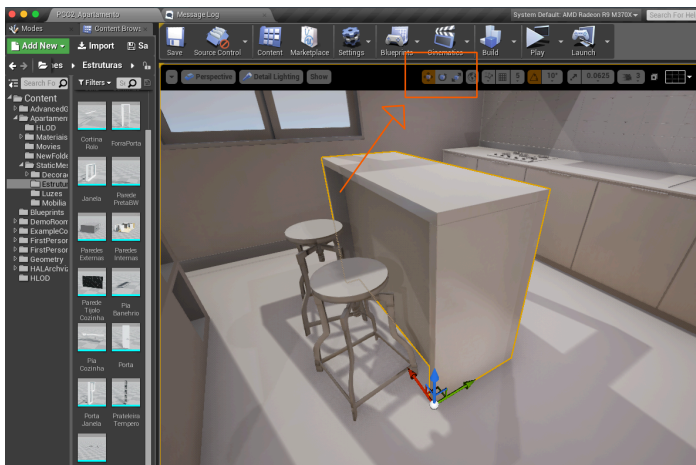


Figura 55 – Posicionar modelos 3D e operações de escala e rotação.
 Fonte: Próprio autor

Seguindo estas etapas de selecionar, arrastar e organizar os modelos no mundo virtual da UE4, foi possível reproduzir a cena antes modelada no programa *Blender* (Figura 56).

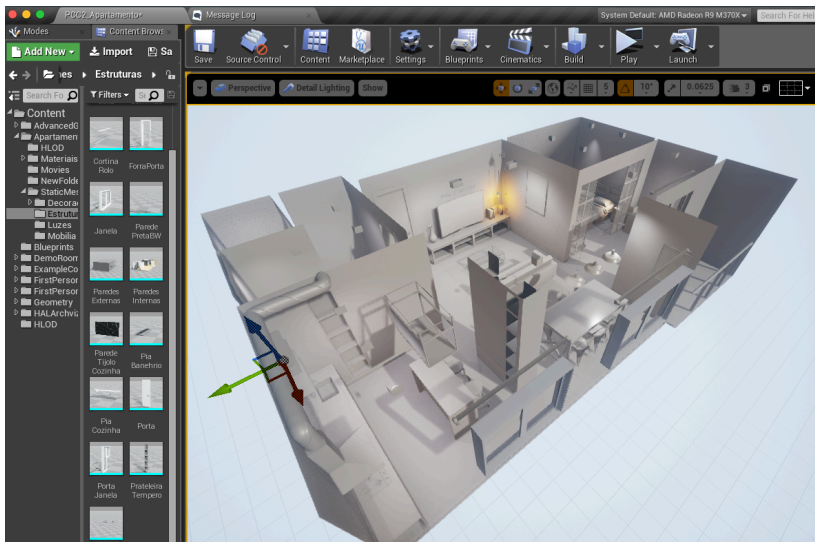


Figura 56 – Modelos importados e cena completa na UE4.

Fonte: Próprio autor

3.6 Iluminação

O *template* inicial que foi escolhido já vem com uma luz *directional light*, que simula o sol e uma luz do tipo *sky light* que simula a luz que vem da atmosfera. O processo de iluminação da cena no mundo virtual é bastante semelhante ao planejar a iluminação de um ambiente real, pois as luzes devem ser colocadas somente onde necessário e dependendo da intenção com diferentes potências, cores e efeitos. Existem outros dois tipos de luzes, as do tipo *point light* que emanam luz de um ponto central iluminando tudo ao redor e as do tipo *spot light* que projetam a partir de um ponto, mas com um formato de cone, que iluminam somente o que estiver perpendicular a direção que o cone aponta (Figura 57).

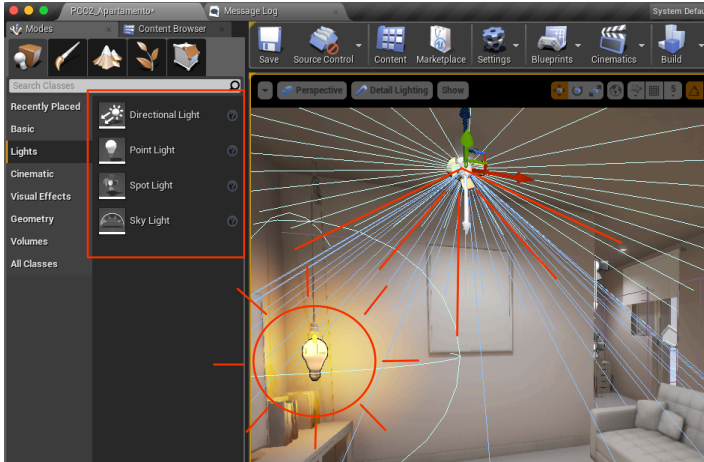


Figura 57 – Exploração de luz do tipo point e spot light.
 Fonte: Próprio autor

Da mesma forma que os modelos 3D, para adicionar as luzes na cena basta selecionar o tipo, na aba *lights* e arrastar para o local desejado. Uma vez na cena, é possível alterar a localização usando como guia os eixos x, y e z. As configurações das luzes são localizadas no canto inferior direito. É possível alterar diversos parâmetros como potência, cor, o quão distante a luz ilumina, e no caso das luzes do tipo *spot light* os ângulos de abertura dos cones de influência (Figura 58).

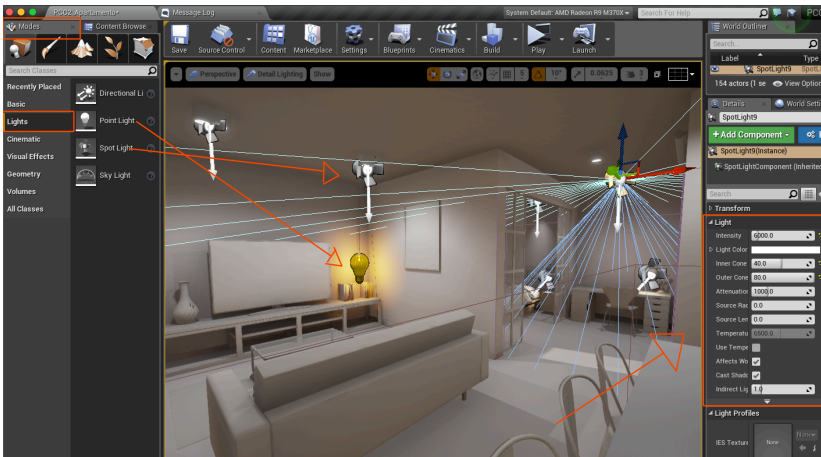


Figura 58 – Parâmetros das cores
 Fonte: Próprio autor

O *template* de jogo em primeira pessoa que após ajustado para servir para projetos de VR, já vem com as configurações do mundo virtual configuradas. Porém deve-se certificar que todos os modelos 3D, como neste caso o apartamento, devem estar dentro do *lightmass importance volume* (figura 59), que é um perímetro no qual a UE4 usa para calcular as informações de luz e sombra, para após salvar em texturas.

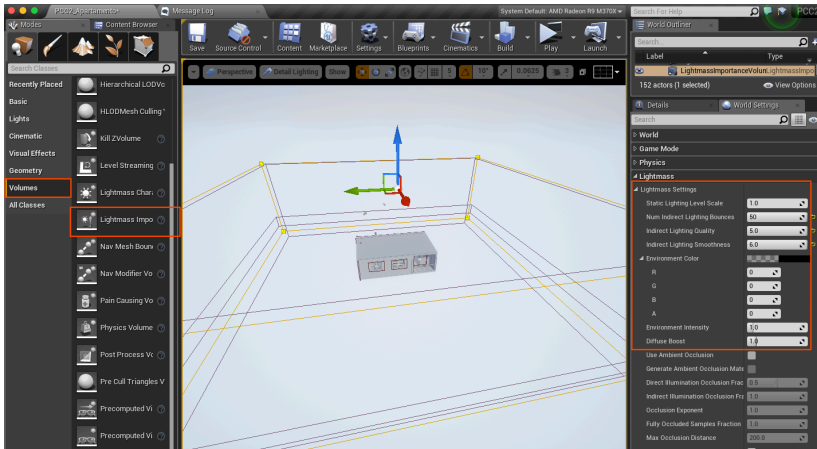


Figura 59 – Modelos envolvidos pelo lightmass importance volume
Fonte: Próprio autor

A UE4 faz os cálculos de luz e sombra previamente em um processo chamado *build* (figura 60), e os salva em texturas que aplicadas nos modelos dão a aparência de estarem iluminados. Toda vez que luzes e objetos tem sua posição, rotação ou escala alterados é necessário fazer o cálculo do *build* novamente para que a UE4, salve as informações novas calculadas.

Durante a execução da aplicação de RV todos as interações que demandem movimento de objetos como abrir portas, mover modelos de lugar, tem as luzes e sombras calculadas em tempo real. Pensando na otimização da aplicação para que não ultrapasse a capacidade que o computador tem de calcular, estas interações devem ser planejadas previamente para que não impactem na performance e consequentemente na experiência do usuário, pois se a aplicação travar ou ficar com a imagem tremendo o usuário pode sentir desconforto.

Durante o desenvolvimento da aplicação de VR deve-se usar a opção de cálculo *lighting quality* como *preview* (figura 60), pois é um cálculo rápido e oferece boa qualidade visual. Na medida em que o desenvolvimento da aplicação avança, testes com qualidade superiores podem ser feitos, apenas o tempo de cálculo será maior. Já o cálculo final da aplicação deve ser no nível *production* para que as informações sejam apresentadas de forma mais precisa e visualmente superiores.

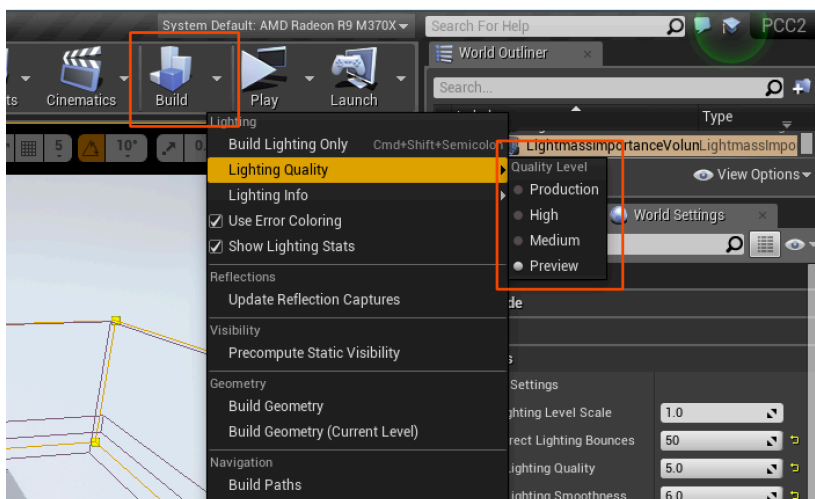


Figura 60 – Níveis de *lighting quality*.

Fonte: Próprio autor.

3.7 Criação de materiais

A etapa de criação de materiais pode ser considerada como uma das mais trabalhosas, pois cada modelo 3D pode ter um ou mais materiais aplicados. Durante a modelagem 3D no programa Blender, alguns modelos tiveram vários materiais aplicados, como foi o caso das paredes internas que teve material de madeira, branco, textura na sala, e a parede preta aplicados como base. Cada material deve ser criado em separado e assim como os modelos 3D podem apenas ser arrastados para cima do objeto que se deseja aplicar o material.

Os materiais criados na UE4 tem parâmetros e configurações baseados em dados físicos do mundo real, chamados de *PBR Physically based material*. Este formato de criação de materiais possibilita uma

melhor interação entre as luzes e os parâmetros, fornecendo assim um aspecto mais condizente ao que vemos no mundo real. Os quatro parâmetros básicos usados na criação dos *PBR materials* são: cor base, nível de rugosidade, metálico, e brilho. Para dar a cor dos materiais poder ser usado valores de *red, green e blue* (rgb), ou uma textura que deve ser importada e armazenada antes do processo de criação de materiais. Neste PCC foram usadas texturas prontas que puderam ser repetidas inúmeras vezes sem aparecer emendas, esta técnica chamada *seamless*, possibilita o uso de texturas pequenas que replicadas podem cobrir a superfície do modelo 3D sem impactar na performance. As texturas foram baixadas no website CGTextures²⁴, que oferece texturas de graça e nesse formato.

Para criar um material basta que na pasta específica seja acessado o painel *add new* e selecionar a opção *material*.

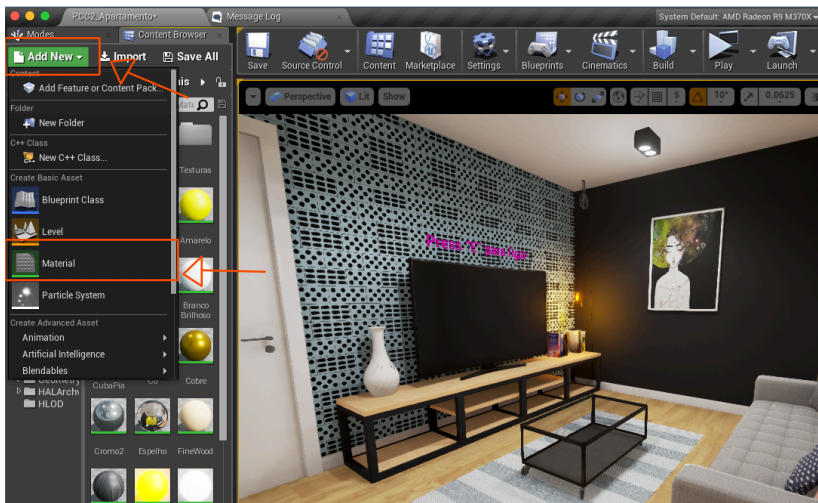


Figura 61 – Criar material

Fonte: Próprio autor

A UE4 usa uma interface baseada em *nodes* que trata os parâmetros num formato de blocos que ligados entre si tornam o processo de criação mais visual e intuitivo. Na figura 61, é possível ver o material branco, criado para ser aplicado em uma das partes do

²⁴ CGTextures. Disponível em: <www.cgtextures.com> Acesso em agosto de 2016

modelo paredes internas. Para criar a cor base foi usado um node *constat 3 vector* que por meio dos canais r,g,b possibilita a criação da cor branca. Como a parede não vai ser reflexiva ou ter brilho, os parâmetros foram deixados em branco, apenas foi adicionado um node *constant* que carrega valores de zero a um, onde valores próximos de zero deixam a parede suavizada e valores próximos de um a deixam áspera.

Desta forma foi possível criar os demais materiais, alguns somente com cor, outros com textura como o tampo da mesa.

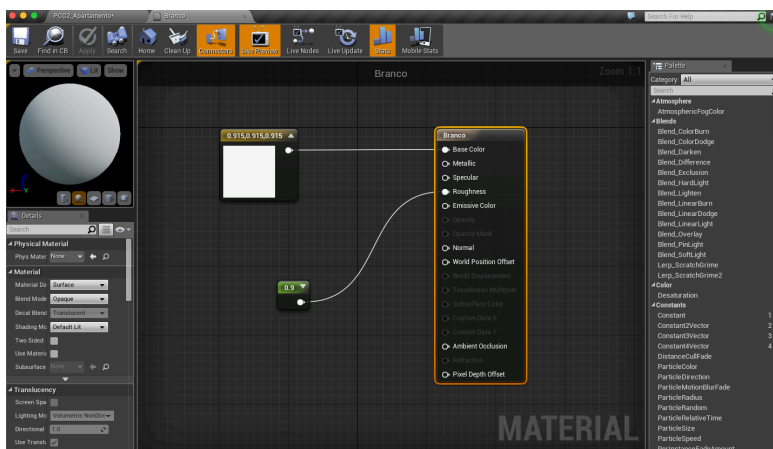


Figura 62 - Material branco

Fonte: Próprio autor

Porém dependendo do efeito que se deseja criar, o material pode ter vários nodes e tornar-se complexo como no caso das lâmpadas (figura 63). Neste caso além ser necessário colocar uma luz próxima ao bulbo da lâmpada, foi necessário criar um material que emitisse luz para dar a ilusão de que ela estava ligada. Este material, assim como os outros teve um parâmetro de cor como base e a partir de um node *multiply* que intensifica os parâmetros nele ligados dá a sensação de luminosidade do material.

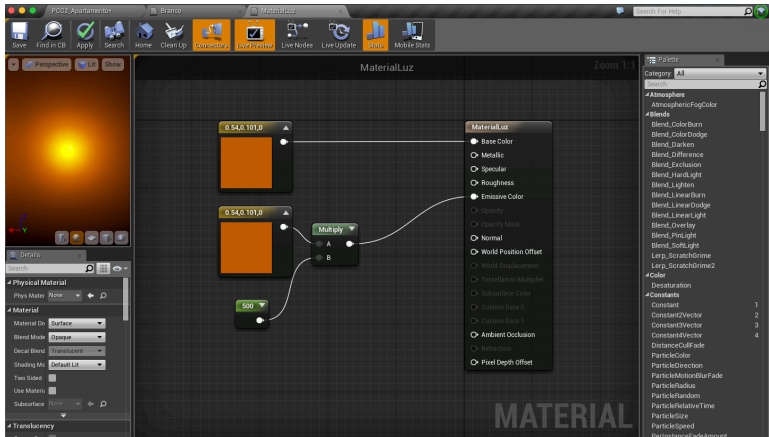


Figura 63 – Material bulbo da lâmpada

Fonte: Próprio autor

Para que os materiais pudessem interagir com a iluminação, foi necessário clicar no processo de *build* várias vezes para ter uma ideia do resultado final. Quando praticamente todos os materiais estavam criados foi feito um último *build* e o resultado pode ser visto na imagem 64.

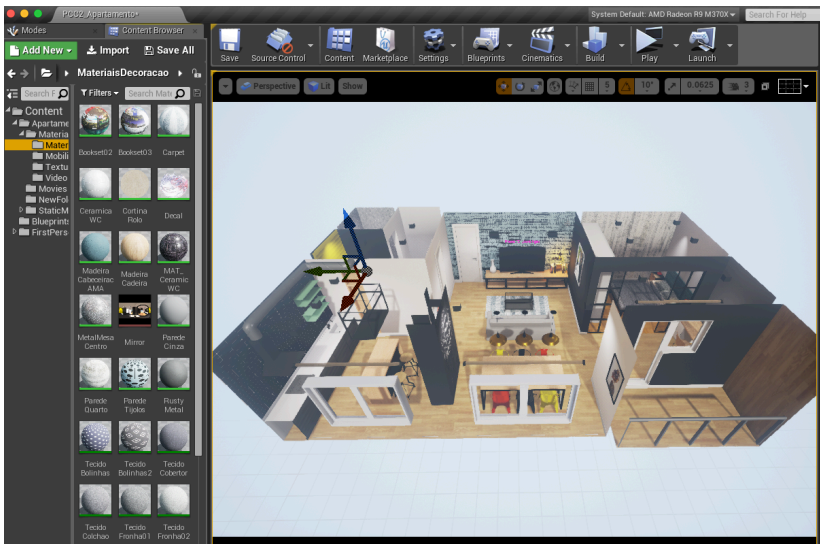


Figura 64 – Todos os materiais aplicados

Fonte: Próprio autor

Após os materiais terem sido criados e aplicados nos modelos 3D foi possível seguir para a criação de interatividades.

3.8 Programação de Interatividades (Blueprint)

Para criar as interatividades a UE4 usa um formato de programação visual chamado *blueprint*. Este formato assim como o editor de materiais é baseado em *nodes*, o que possibilita um formato mais visual e intuitivo. Baseado no mesmo processo de criação de materiais, foi criada uma pasta chamada *blueprint*, e ao selecionar a opção *add new* foi escolhido *blueprint class* para iniciar a programação.

Para este projeto, como se trata de um ambiente virtual imersivo, foi criado um sistema que possibilitou o usuário trocar materiais de alguns modelos 3D. Esta outra possibilidade de interação proporciona ao usuário mais controle sob a aplicação e maior sensação de imersão. Como as interações são baseadas em eventos, foi criado um volume ao redor do modelo 3D em questão e a partir do momento que o personagem virtual colidisse habilitaria a possibilidade de interação mostrada por meio de um texto informando press “E” (Figura 65).

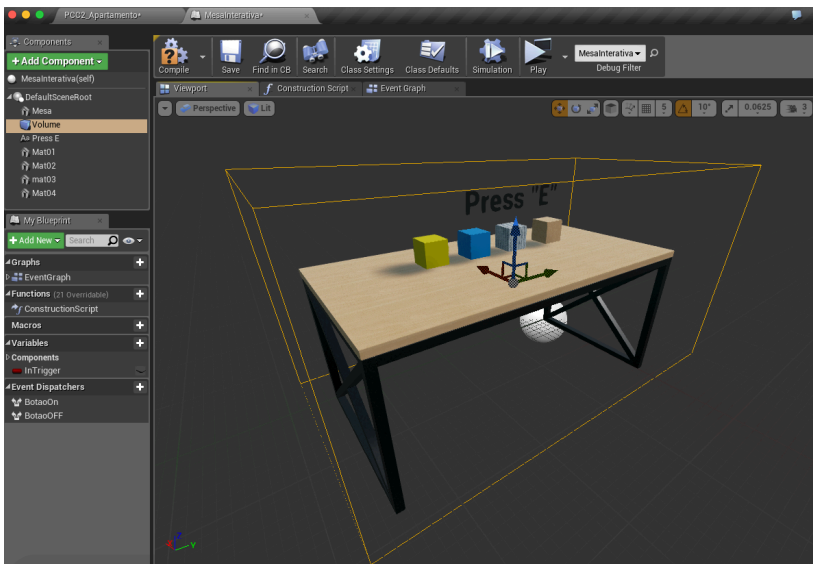


Figura 65 – Volume envolvendo o modelo 3D para desencadear evento de interação.

Fonte: Próprio autor

A programação visual mostrada na figura 66, exemplifica como funcionou este sistema de interação. Basicamente quando o personagem virtual colidisse com o volume habilitaria a opção para press “E” e apareceria os quadrados em cima da mesa com as opções de material disponíveis.

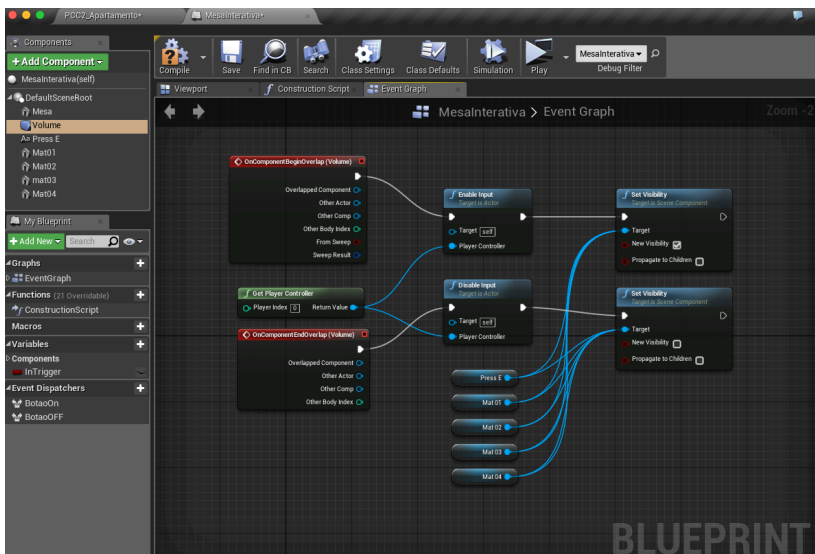


Figura 66 – Sistema inicial de programação para habilitar interatividade.

Fonte: Próprio autor

Outra programação foi feita para desencadear o evento de troca de materiais. Quando o usuário apertasse a letra “E” exemplificado na imagem 67. O node *multigate* que tem um parâmetro de entrada, que é o evento tecla “E” e os eventos de saída que neste caso são as opções de materiais escolhidas. Poderia ser adicionado quantos materiais fossem necessários, bastando apenas clicar na opção *add pin* no node *multigate*. Cada node chamado *set material* precisou ter um material específico criado previamente para ser adicionado e após ligado na opção de saída no node *multigate*. Neste caso no node *set material* está ligado como *target* no modelo mesa, e para criar os demais modelos interativos apenas foi duplicada esta programação e trocada a referência para o modelo desejado.

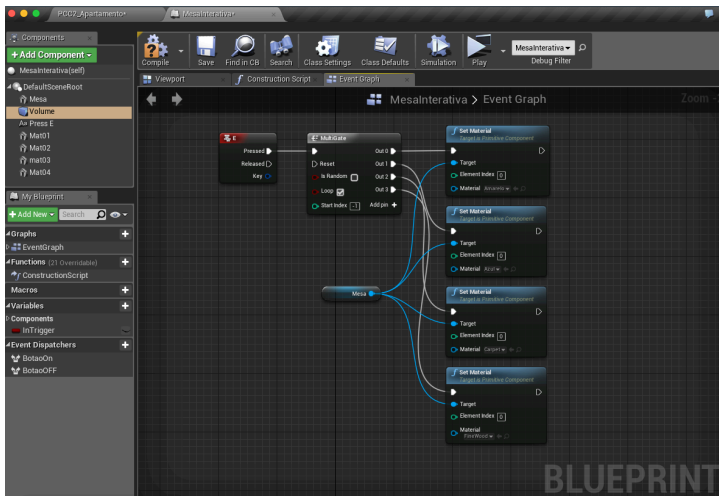


Figura 67 – Evento troca de materiais.
Fonte: Próprio autor

3.9 Testes

Após todos os ajustes referentes a parte visual do apartamento, assim como todas as interatividades, foi possível iniciar os testes da aplicação. Algumas recomendações referentes ao computador a ser utilizado para executar a aplicação foram encontradas nos arquivos de desenvolvimento do Oculus Rift (figura 68).

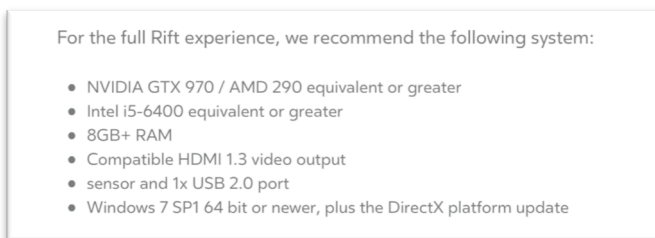


Figura 68 – Requisitos de configuração de computador para ter boa performance em aplicações de RV.

Fonte:

<<https://developer3.oculus.com/documentation/pcsdk/latest/concepts/gsg-intro/#dg-sdk-setup-requirements>> Acesso em outubro de 2016.

Este PCC foi desenvolvido parte em um *notebook* da plataforma *Mac*, como modelagem 3D e integração com a UE4, porém os testes foram feitos em um computador da plataforma *Windows*. O computador em que este PCC foi testado continha uma placa de vídeo *Nvidia GTX 670*, o que é inferior ao recomendado, porém não foi constatado queda significativa de performance da aplicação.

Ainda nesta fase de testes, a aplicação estava sendo executada internamente na UE4 e alguns procedimentos foram necessários para visualizar com o *Oculus Rift*. Primeiramente foi necessário instalar os *drivers*, programas que reconheceriam o *Oculus Rift* quando ligado no computador, e para isto foi criada uma conta em seu site para ter acesso aos arquivos de instalação que uma vez baixados, foram intuitivamente guiando para os procedimentos de instalação (figura 69).

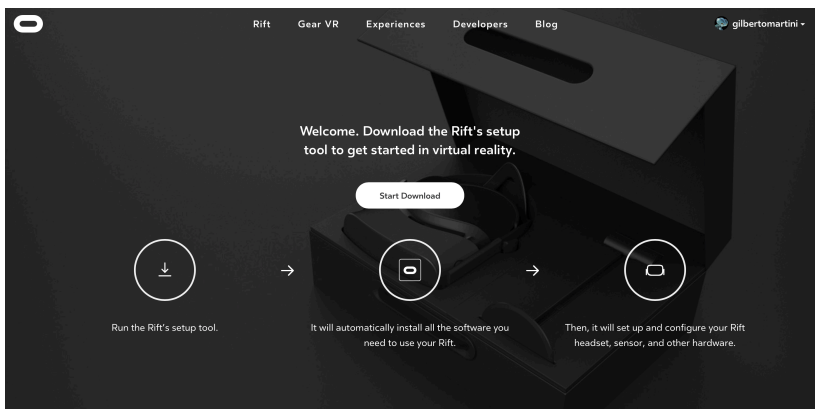


Figura 69 – Tela do local para baixar os drivers do Oculus Rift.

Fonte : Próprio autor

Uma vez que os *drivers* estavam instalados e o *Oculus Rift* ligado ao computador, internamente na UE4 foi possível testar a aplicação. Para isto, foi necessário na aba *play* escolher a opção *VR preview* (figura 70).

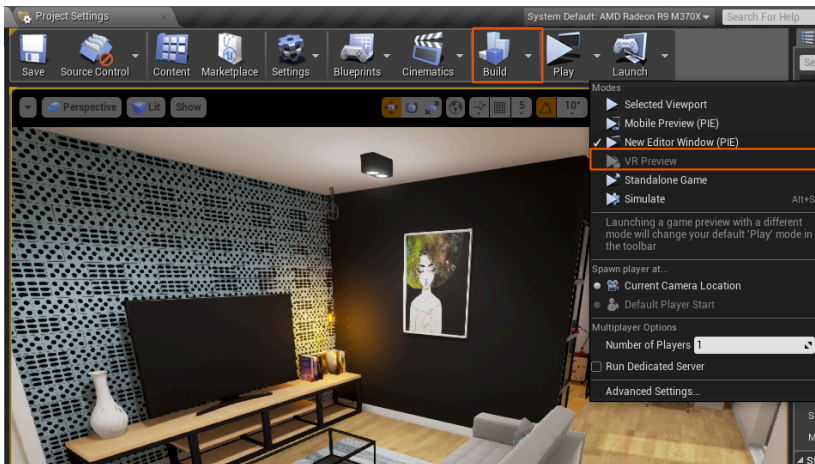


Figura 70 – Selecionando o modo de preview em RV.

Fonte: Próprio autor.

Durante a etapa de testes foi constatado que apesar de a UE4 fornecer basicamente todo o ambiente necessário para uma aplicação de RV, alguns ajustes foram necessários. Primeiramente a escala do personagem padrão que vem como o template precisou ser alterada para um tamanho inferior a 1.90 metros. Outro ponto, é que em RV a movimentação do personagem virtual deve ser fluida e evitar movimentos bruscos, pois isso faz com o que o usuário sinta-se enjoado. Para isto a velocidade de rotação e a velocidade de movimentação do personagem virtual foi diminuída, aproximando-se das velocidades médias que acontecem do mundo físico. Da mesma forma, a opção de *motion blur*, que cria um desfoque de acordo com a movimentação, foi desabilitada e favoreceu para uma menor sensação de tontura. A figura 71, mostra o outor deste projeto interagindo com a aplicação final.



Figura 71 - Aplicação final do apartamento sendo testada.
Fonte : Próprio autor

3.10 Exportar aplicação

Somente após todos os testes referente a parte visual e interativa da aplicação terem sido finalizados, é que foi possível iniciar o processo de exportar o projeto como uma aplicação de VR por meio de um processo chamado *package content*. Este processo junta todos os arquivos criados e salva uma como uma aplicação, porém deve ser o último a ser realizado, pois a partir do momento em que é feito o arquivo torna-se fechado impossibilitando alterações ou correções. A UE4, faz o processo de exportação da aplicação praticamente que automático, basta selecionar qual a plataforma de destino e escolher o lugar para salvar o arquivo final conforme figura 72.

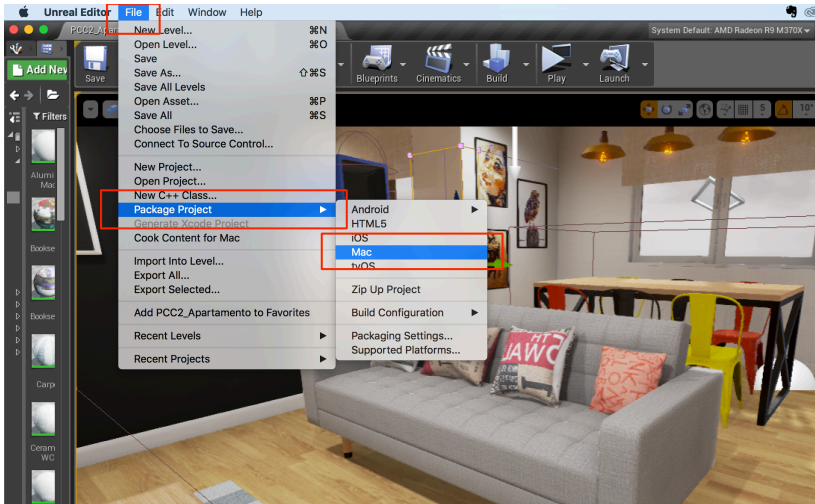


Figura 72 – Salvar projeto como uma aplicação de RV.
Fonte: Próprio autor.

Antes de iniciar a aplicação deve-se ligar e conectar o *oculus rift* no computador, para após abrir o arquivo da aplicação, caso contrário a aplicação não reconhecerá o dispositivo de RV. Para movimentar-se virtualmente por meio do teclado do computador, as teclas W(para frente), A(para a esquerda), S(para trás), e D(para direita) respectivamente realizam o movimento do usuário dentro do mundo virtual. O *oculus rift* dá a liberdade de movimentação virtual da cabeça, possibilitando olhar o ambiente 3D em todas as direções.

4 Conclusão

Diante de todo o processo de produção deste PCC, foi possível ter uma noção das etapas necessárias para a elaboração de uma aplicação de RV para servir como ferramenta na apresentação de um projeto de arquitetura.

Este tipo de aplicação mostrou-se viável tanto em termos de produção quanto em efetividade nos resultados, pois a possibilidade de imersão, grau de realismo, e interação proporcionados superaram as expectativas. Este tipo de ferramenta não substitui os projetos técnicos realizados por um arquiteto, mas auxilia para que o cliente tenha ao seu dispor algo mais visual e interativo.

A RV, faz jus ao seu diferencial de multidisciplinaridade ao ser capaz de suprir a demanda de mais uma área, e diante de tantos avanços tecnológicos, possivelmente será cada vez mais solicitada e inserida no nosso dia dia.

REFERÊNCIAS

- DAN, John. **Oculus Rift: O óculos que está revolucionando toda a história da tecnologia de realidade virtual**. 2011. Disponível em: <<http://onovonerd.blogspot.com.br/2013/02/oculus-rift-o-oculos-que-esta.html>>. Acesso em: 09 nov. 2016.
- DESIGN COUNCIL (Design Council). **Eleven Lessons: managing design in eleven global brands: A study of the design process**. 2007. Disponível em: <[http://www.designcouncil.org.uk/sites/default/files/asset/document/ElevenLessons_Design_Council_\(2\).pdf](http://www.designcouncil.org.uk/sites/default/files/asset/document/ElevenLessons_Design_Council_(2).pdf)>. Acesso em: 09 nov. 2016.
- GAMES, Epic. **Static Meshes - Unwrapping UVs for Lightmaps**. Disponível em: <<https://docs.unrealengine.com/latest/INT/Engine/Content/Types/StaticMeshes/LightmapUnwrapping/>>. Acesso em: 10 out. 2016.
- HOON, Michael; KEHOE, Michael. **Enhancing Architectural Communication with Gaming Engines**. 2003. Disponível em: <http://cumincad.architecturez.net/system/files/pdf/acadia03_045.content.pdf>. Acesso em: 09 nov. 2016.
- KIM, Gerard Jounghyun. **Designing Virtual Reality Systems – The structured approach**. Londres: Springer, 2005. 233 f.
- PARISI, Tony. **Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web and Mobile**. Sebastopol, Ca: OREILLY, 2015. 7 p.
- PEREIRA, Ivo; NOGUEIRA, Nuno. **Realidade Virtual**. 20--?. Disponível em: <<http://web.ist.utl.pt/ist170613/>>. Acesso em: 15 maio 2016.
- RIBEIRO, Nuno. **Multimédia e tecnologias interactivas**. 2. ed. Lisboa: FCA, 2007. p 391 a 395.
- SHNEIDERMAN, Ben; PLAISANT, Catherine. **Designing the user interface: Strategies for effective human-computer interaction**. 5. ed. Maryland: PEARSON, 2009. 210 p.
- TORI, R. ; KIRNER, C. In: **Symposium on Virtual Reality**, 8., 2006, São Paulo. Fundamentos e tecnologia de realidade virtual e aumentada: Fundamentos da Realidade Virtual. Belém: SBC, 2006. 56 p.