

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO DE EXPRESSÃO GRÁFICA
CURSO DE GRADUAÇÃO EM ANIMAÇÃO - BACHARELADO

PEDRO EURICO RODRIGUES MARTINS

**ESTUDO SOBRE O USO DE MÓDULOS NA CRIAÇÃO DE CENÁRIOS PARA JOGOS
DIGITAIS**

Florianópolis

Dezembro de 2019

Pedro Eurico Rodrigues Martins

**ESTUDO SOBRE O USO DE MÓDULOS NA CRIAÇÃO DE CENÁRIOS PARA JOGOS
DIGITAIS**

Trabalho Conclusão de Curso de Graduação em Animação do Centro de Comunicação e Expressão da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Animação.

Orientador: Prof. Flávio Andaló

Florianópolis

Dezembro de 2019

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

Eurico Rodrigues Martins, Pedro

Estudo sobre o uso de módulos na criação de cenários para jogos digitais / Pedro Eurico Rodrigues Martins ; orientador, Flávio Andaló, 2019.

41 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro de Comunicação e Expressão, Graduação em Animação, Florianópolis, 2019.

Inclui referências.

1. Animação. 2. 3D. 3. Modelagem. 4. Animação. 5. Níveis. I. Andaló, Flávio. II. Universidade Federal de Santa Catarina. Graduação em Animação. III. Título.

Pedro Eurico Rodrigues Martins

Estudo sobre o uso de módulos na criação de cenários para jogos digitais

Este Trabalho de Conclusão de Curso (TCC) foi julgado adequado para obtenção do Título de Bacharel em Animação e aprovado em sua forma final pelo Curso de Animação da Universidade Federal de Santa Catarina.

Florianópolis, 26 de novembro de 2020.

Prof. Flávio Andaló, Dr. Coordenador do Curso de Animação UFSC

Banca Examinadora:

Professor Gabriel de Souza Prim, Ms. UFSC.

Professor Renan de Paula Binda, Ms. UFSC.

Professor Flávio Andaló, Dr. UFSC. Orientador.

Professor/a Orientador/a

Universidade Federal de Santa Catarina

Resumo

Durante o desenvolvimento deste projeto buscou-se inicialmente uma compreensão sobre conceitos estabelecidos a respeito de *game design*, mais especificamente os referentes a *level design* envolvendo técnicas e práticas aplicadas no processo de criação de níveis para jogos digitais e animações. A partir deste entendimento foi possível elaborar uma conexão entre estes aspectos e a ideia de construção de blocos modulares, pois os profissionais que trabalham no desenvolvimento de níveis precisam das “ferramentas” para tornar seu trabalho mais eficaz e a construção através de módulos serve exatamente como tal, revelando assim um meio de aceleração do processo de concepção de grandes cenas, além de otimizar requisitos de processamento e vídeo para estes cenários.

Com este conhecimento, traçou-se o cerne deste projeto: a modelagem de um pacote modular utilizando o software *Autodesk 3DS Max*, contendo uma gama de *assets* que viabilizem a produção de diferentes tipos de cenários, aplicar este conteúdo no software *Unreal Engine 4*, para então ser criado um *level* que siga os conceitos estudados e sirva como demonstração da aplicabilidade desta prática. Como produtos finais deste trabalho seriam apresentados, além de um nível jogável, uma animação cinemática (neste mesmo nível) apontando os principais pontos abordados.

Palavras-chave: Animação, *Level Design*, Modularidade, *Unreal Engine 4*, 3D, Modelagem.

Abstract

During the development of this project the initial objective was to understand established concepts about game design, more specifically those that focus on level design, like techniques and good practices that are applied to the process of level creation for digital games and animations. After acquiring knowledge about those topics it was possible to create a connection between those aspects and the idea of modular kits. Professionals that work in the industry require tools to make their work more efficient and the process of creating levels using this technique of modular blocks is exactly that, a way to speed up the time necessary to create huge scenes and optimize the graphics requirements.

With those ideas in mind, it was possible to set the focus of this project: the concepting and modeling of a modular kit using the software Autodesk 3DS Max, this package should contain a big pool of assets in order to enable the creation of different types of levels. This content would be exported to the Unreal Engine 4. After these steps have been completed the kit would be applied in a complete level, this one being a demonstration scene that showcases the most important topics addressed in this project. The final product would be a playable level and a short animation that travels through it.

Keywords: Animation, Level Design, Modular, Unreal Engine 4, 3D, Modeling.

Sumário

1	Introdução.....	8
2	Referencial Teórico.....	9
3	Procedimento Metodológicos.....	11
4	Desenvolvimento.....	13
4.1	Primeira Etapa.....	13
4.1.1	Pré-Produção.....	13
4.1.2	Modelagem.....	15
4.1.3	Revisões.....	16
4.2	Segunda Etapa.....	16
4.3	Terceira Etapa.....	20
4.3.1	Pré-Produção.....	20
4.3.2	Produção.....	21
4.4	Quarta Etapa.....	23
5	Resultados.....	24
5.1	Resultados - Modelagem.....	24
5.2	Resultados - Pacote de Conteúdo.....	27
5.3	Resultados - Aplicação em Cenário.....	33
5.4	Resultados - Criação de Cinematic.....	36
6	Conclusão.....	38
7	Referências.....	40

Índice de Figuras

Figura 1 - Cronograma quando foi iniciado o segundo semestre de 2019.....	12
Figura 2 - Painel semântico criado para etapa de modelagem.....	15
Figura 3 - Pacote de Modelos recém-importados.....	17
Figura 4 - Amostra dos materiais utilizados.....	18
Figura 5 - Amostras dos <i>decals</i> criados.....	19
Figura 6 - Amostra dos sistemas de partículas desenvolvidos.....	20
Figura 7 - Segundo painel semântico.....	21
Figura 8 - Etapas do progresso do cenário.....	22
Figura 9 - Exemplo de enquadramento escolhido para <i>cinematic</i>	23
Figura 10 - Resultado 1 do pacote de modelos.....	24
Figura 11 - Resultado 2 do pacote de modelos.....	25
Figura 12 - Resultado dos módulos criados ainda no 3DS Max.....	26
Figura 13 - Primeiro estudo de iluminação e composição.....	26
Figura 14 - Pacote de conteúdo pronto na <i>Unreal Engine</i>	27
Figura 15 - Demonstração das possibilidades ao se usar instância.....	28
Figura 16 - Demonstração dos <i>Merged Assets</i>	29
Figura 17 - Visualização do resultado ao se combinar um dos módulos pré-moldados.....	29
Figura 18 - Exemplos dos módulos.....	30
Figura 19 - Demonstração de todos módulos criados.....	31
Figura 20 - Comparação na resolução das sombras projetadas.....	33
Figura 21 - Cenário utilizando o filtro de complexidade de iluminação.....	34
Figura 22 - Resultado de se usar um sistema de partículas com característica emissiva.....	35
Figura 23 - Cenário com filtro da complexidade de <i>shaders</i>	36

Figura 24 - Cenário completo.....	36
Figura 25 - Cenário pela lente da câmera utilizada.....	37

1 Introdução

A indústria de games está em ascensão não somente no Brasil, mas sim em uma escala global (WIJMAN, 2019). Para suprir este mercado voraz as empresas têm que manter um constante fluxo de adições e atualizações de conteúdo para os seus jogos para prender o interesse do público. Essa demanda requer uma eficiente e rápida linha de produção, além de um conteúdo de alta qualidade. Com essas necessidades este projeto buscou primeiramente analisar uma destas etapas de produção, a criação de cenários. Mais especificamente a utilização de pacotes ou kits modulares para desenvolvimento destes ambientes, tentando entender como esta técnica serve como um acelerador eficaz, além de possibilitar que diferentes times que trabalhem simultaneamente em um mesmo jogo consigam manter um padrão coeso nas suas peças criadas, estas também conhecidas na indústria de jogos pelo termo em inglês *asset*.

Para realizar este estudo foi definido o objetivo do projeto, sendo ele a criação de um destes pacotes modulares e posteriormente a implementação deste kit em um cenário original, tornando desta forma possível a avaliação deste processo através da experiência e coleta de dados.

2 Referencial Teórico

Durante as diferentes fases de produção deste projeto os conhecimentos envolvidos foram baseados em três alicerces: gestão de projetos aplicada para a indústria; a estruturação dos passos e práticas na criação de níveis jogáveis; e por último o domínio do *software* a ser utilizado na etapa de produção, a *Unreal Engine 4*.

Abordando estes tópicos respectivamente, quanto à parte relativa ao gerenciamento do trabalho, utilizaram-se as informações obtidas na disciplina de gestão de projetos, disponibilizada pelo curso, bem como o material de referência coletado, com a menção especial para o artigo “*How to meet deadlines without micromanaging*” (SIMON, 2018), onde são abordados os principais guias que levaram este projeto adiante, sendo eles: o estabelecimento das metas e as expectativas a serem alcançadas como um primeiro passo, a comunicação, confiança e honestidade entre os membros da equipe, neste caso por se tratar de um projeto desenvolvido individualmente estes aspectos podem ser traduzidos como o entendimento dos limites que são impostos ao se enfrentar um desafio como tal, com uma única pessoa trabalhando, sejam eles relativos à restrição de tempo (um semestre), a quantidade de trabalho que pode ser feita no período de uma *sprint*, ou até a impossibilidade de realizar duas, ou mais tarefas em paralelo. Citado ainda dentro deste artigo, a necessidade de se estipular uma carga de trabalho saudável para que se possa manter um ritmo produtivo durante todas as etapas de produção.

Quanto ao método utilizado para gerenciamento das atividades a serem realizadas no projeto, a técnica de *Scrum*, que é um sistema de organização de projetos que subdivide as etapas de desenvolvimento, posicionando-as em uma espécie de “esteira industrial”, onde cada atividade passa por diferentes processos até poder ser entregue como uma fração do produto final, ao se vencer todas as subdivisões têm-se o trabalho completo. O Scrum ainda propõe reuniões de equipe diárias para que possam ser discutidos os conhecimentos absorvidos no dia anterior, possíveis barreiras que nas atividades a serem realizadas e o que deve ser produzido no dia atual.

Como segundo alicerce do conhecimento necessário para este projeto, utilizou-se as informações reunidas no artigo *A beginner’s guide to designing video game levels*” (STOUT, 2016) em conjunto com as experiências relatadas no *e-book* “*How to create a map in 11 days*” (GALUZIN, 2011). Ambas as referências propunham que deveriam ser respondidas perguntas sobre o nível a ser criado, para que as respostas servissem de diretrizes a serem seguidas durante a produção. Exemplos destes questionamentos: duração, se tratando de tempo o quão longo este

nível deve ser; para qual plataforma este nível, ou jogo, seria direcionado; qual o público-alvo; ou ainda qual a história que seria contada dentro daquele espaço.

Ainda seguindo alguns dos conceitos propostos (STOUT, 2016) reunidos na parte do processo de criação e confecção de um *level*, foram estruturadas seis etapas a serem vencidas:

1. Definição dos limites impostos;
2. Conceituação do nível;
3. Blocação;
4. Diagramação do fluxo de jogabilidade;
5. Estruturação do mapa;
6. Finalização;

Dentre estas etapas como por exemplo as limitações em questões de processamento gráfico e quantidade de triângulos em cena, bem como o fluxo de jogabilidade são pontos onde utilizar um software em que podem ser feitos testes de jogabilidade e analisar dados em tempo real se torna uma grande vantagem. Pode ser mencionada ainda a enorme quantidade de material disponível para aprendizado sobre o manuseio do software Unreal Engine, seja este material produzido pela própria EPIC Games na sua seção de Learning, ou pelos vídeos e live-streams¹ periodicamente postados em seus canais de comunicação, que abordam novas ferramentas implementadas no programa, ou como melhor utilizar outras já presentes. Além de fóruns onde outros usuários, de diferentes lugares do planeta, compartilham conhecimento e ajudam-se solucionando problemas e dúvidas.

¹ *Live-stream*: Compartilhamento de dados e informação feita ao vivo em canais como, por exemplo, *YouTube*. Neste caso o compartilhamento se refere a aulas, palestras ou demonstrações de funcionalidade dentro da *Unreal Engine*

3 Procedimento Metodológicos

Como método de gerenciamento de produção, foi optado por se usar a técnica de Scrum. Este sistema quando aplicado ao projeto desenvolvido gerou uma pilha de dezessete atividades a serem desenvolvidas, sendo estas conhecidas como o backlog. Para que fosse estimado o tempo das atividades foram delimitadas as sprints, ou seja, o período de atividades realizadas, neste caso dentro de uma semana, sendo que alguns dos elementos empilhados no backlog foram projetados para serem finalizados em metade de uma sprint enquanto outros requeriam até duas, chegando assim a um total de vinte e uma sprints a serem concluídas. No caso deste projeto optou-se por criar algo semelhante a um “diário”, que cumprisse esta função de conexão, expondo as dificuldades encontradas mantendo em vista os objetivos e metas diárias a serem atingidas e indo além delineando o foco para o dia seguinte.

Durante o projeto os métodos utilizados para se avaliar os conceitos abordados foram principalmente de ordem quantitativa, com alguns pontos sendo levada em consideração a parte qualitativa. Respectivamente a ideia de quantidade, estando presente nas estimativas de tempo demandado por atividade, fornecendo assim uma noção da aceleração de alguns dos processos realizados. Outro exemplo em que os dados quantitativos puderam ser analisados foi na contagem de triângulos de algumas peças modulares e também de algumas cenas do cenário final. A coleta de dados do desempenho do cenário final como um todo serviria como base para análise.

A parte qualitativa de avaliação atrelou-se a decisões artísticas do projeto, em casos onde se tinha as opções de melhorar a aparência de, por exemplo, as sombras projetadas de um objeto sobre o outro.

Para que todas as metas estabelecidas pudessem ser alcançadas com tempo extra para correções e possíveis atrasos em etapas de produção, foi elaborado um cronograma de atividades, este dividido em *sprints* semanais. Seguindo esta organização, o projeto estava seccionado em grandes etapas a serem vencidas, estando elas discriminadas a seguir:

- a. Primeira Etapa: Pré-produção, desenvolvendo um conceito para os modelos a serem criados, com a elaboração de um painel semântico para servir como guia para a etapa de modelagem. Produção de modelos 3D e exportação em pacotes divididos prontos para serem importados para *Unreal Engine*.
- b. Segunda Etapa: Importação de todo o material gerado utilizando o *3DS Max* para *Unreal*, criação de um pacote de materiais que seguissem o conceito estabelecido

na primeira etapa. Montagem de um *level* de demonstração contendo as bibliotecas de materiais e os *assets* produzidos, estes já prontos para serem levados para o próximo estágio de produção.

- c. Terceira Etapa: Reunião de referências para montagem de um segundo painel semântico e a aplicação do pacote desenvolvido na segunda etapa dentro de um nível completo.
- d. Quarta Etapa: Produção de uma *cinematic* com estilo *teaser* dentro do cenário da etapa anterior, salientando os principais pontos desenvolvidos.

A Figura 1 representa o estado do cronograma criado para o desenvolvimento do projeto no decorrer de sua produção, separado em estágios como proposto pela técnica do *Scrum*. Estão discriminadas em cada uma das atividades o ponto de início relativo as *sprints*.

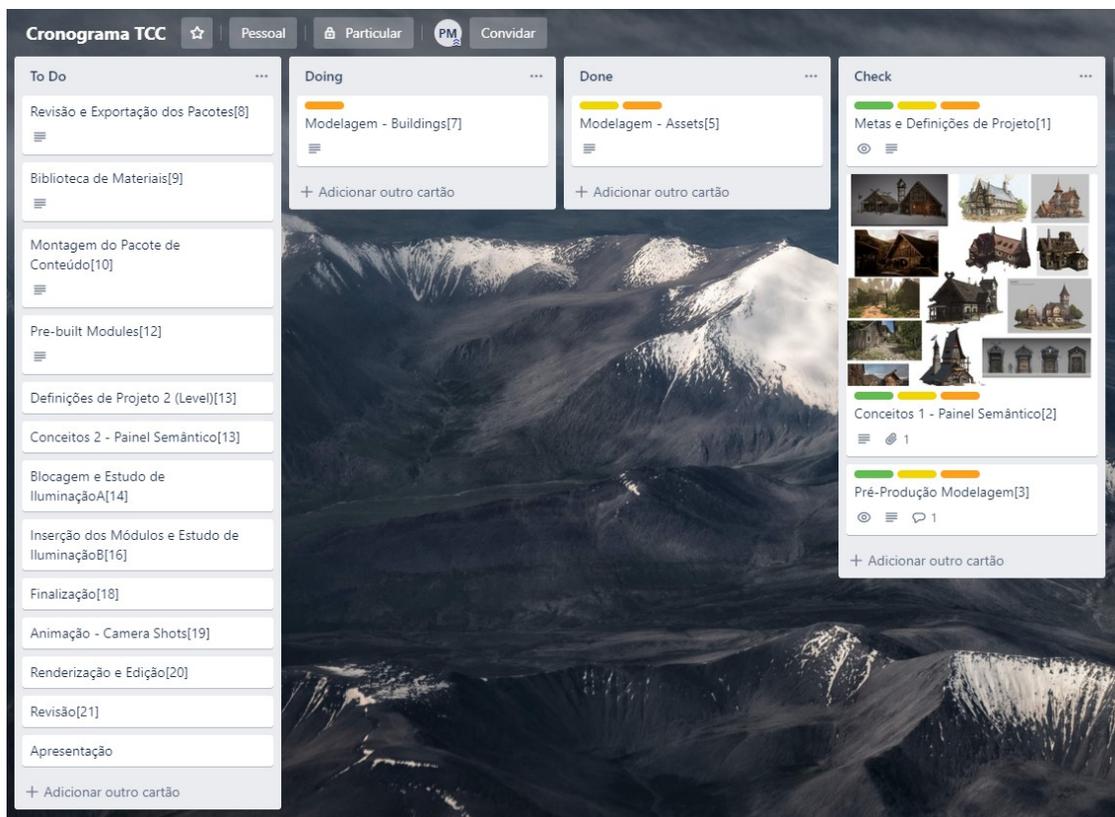


Figura 1 - Cronograma quando foi iniciado o segundo semestre de 2019.

Fonte: montado no *Trello* pelo autor

4 Desenvolvimento

Com o início das atividades de produção do trabalho as tarefas a serem realizadas foram divididas seguindo a metodologia já descrita no item anterior.

Se tratando dos softwares a serem utilizados no decorrer do trabalho, foi decidido utilizar exclusivamente produtos gratuitos, dessa forma: os modelos seriam criados utilizando a licença estudantil do *Autodesk 3DS Max*, para coleta de texturas a página da *web Textures.com*, para criação dos mapas de materiais o programa *Materialize* e para criação do pacote modular e cenário seria utilizado o programa distribuído pela *EPIC Games* a *Unreal Engine 4*, esta que divide espaço no mercado atual com a *Unity* como grandes nomes em questão de plataforma para criação de jogos digitais, abrangendo variados tipos de recursos possibilitando aos desenvolvedores uma oportunidade de gerar um produto original.

4.1 Primeira Etapa

4.1.1 Pré-Produção

O primeiro passo para construção deste trabalho, conceituação, começou a tomar forma com a definição de questões sobre qual tipo de pacote seria montado: se seria algo voltado mais para um estilo futurista? Talvez algo contemporâneo com prédios e automóveis? Ou ainda motivado por outros jogos, literatura e filmes um tema épico de fantasia. Ficando decidido que seria uma mistura entre os possíveis temas: trazendo elementos da época medieval, entrelaçados com algo mais contemporâneo quando aplicado para a arquitetura, além de uma veia nórdica. Esta decisão foi motivada pela grande quantidade de material já presente no mercado que serviria como comparação, além das inspirações em outros setores como literatura e cinema. Com estas decisões estabelecidas seria possível avançar algumas etapas para que também já fossem visualizados os principais materiais que seriam aplicados posteriormente, tendo eles como fundamentos pedras, rochas, ou até concreto. Diversos tipos de madeira e por último metais como ferro, cobre, bronze, com variações de envelhecimento como ferrugem, arranhões e desgaste.

Baseado nestas definições pré-estabelecidas, foi montado um painel semântico² de referências a serem seguidas enquanto fossem criados os *assets*, buscando-se manter uma quantidade relativamente pequena de objetos, mas sem perder a identidade do tema.

² Um painel semântico, ou também conhecido como *Mood Board*, são grupo de imagens, fotos, ilustrações ou qualquer material que sirva como referência para transmitir as ideias do projeto em questão.

Ainda na pré-produção dos modelos foi levantando o questionamento de em qual escala de grandeza o nível final a ser produzida na terceira etapa se encaixaria: se seria algo como uma pequena vila com quatro ou cinco construções modulares; ou talvez uma pequena cidade com a contagem de casas e construções girando em torno de vinte até trinta; ou ainda algo realmente ambicioso como uma zona de larga escala com sessenta blocos. Por fim ficou determinado que um meio termo de cidade seria um objetivo plausível e alcançável dada a limitação de tempo imposta sobre o projeto. Com o estilo, materiais, características e escala passou-se a construção dos *assets* de uma forma que fosse possível gerar uma ideia de que cada módulo seria único em aproximadamente quinze construções.

Como um último ponto de reflexão ainda neste momento relativo à pré-produção, foi levantada ainda mais uma questão sobre o nível final. Para qual tipo de jogo este cenário seria criado: em caso de um jogo de tiro em primeira pessoa, a preocupação seria com as áreas abertas e os *assets* inseridos nesses espaços, bem como construções que seriam interconectadas, fornecendo diferentes meios de movimentação pelo cenário. Neste caso detalhes em questão de telhados e outros pontos muito acima da linha do horizonte não seriam vitais. Por outro lado, um jogo focado em exploração e missões, o jogador não estaria constantemente sob pressão. Tendo a opção de olhar os diferentes aspectos dentro do nível, desta forma, características do ponto de vista artístico teriam um peso maior. A preocupação na criação destes módulos seria voltada pra que eles fizessem sentido de um ponto de vista arquitetônico e físico e ao mesmo tempo agradáveis ao olhar.

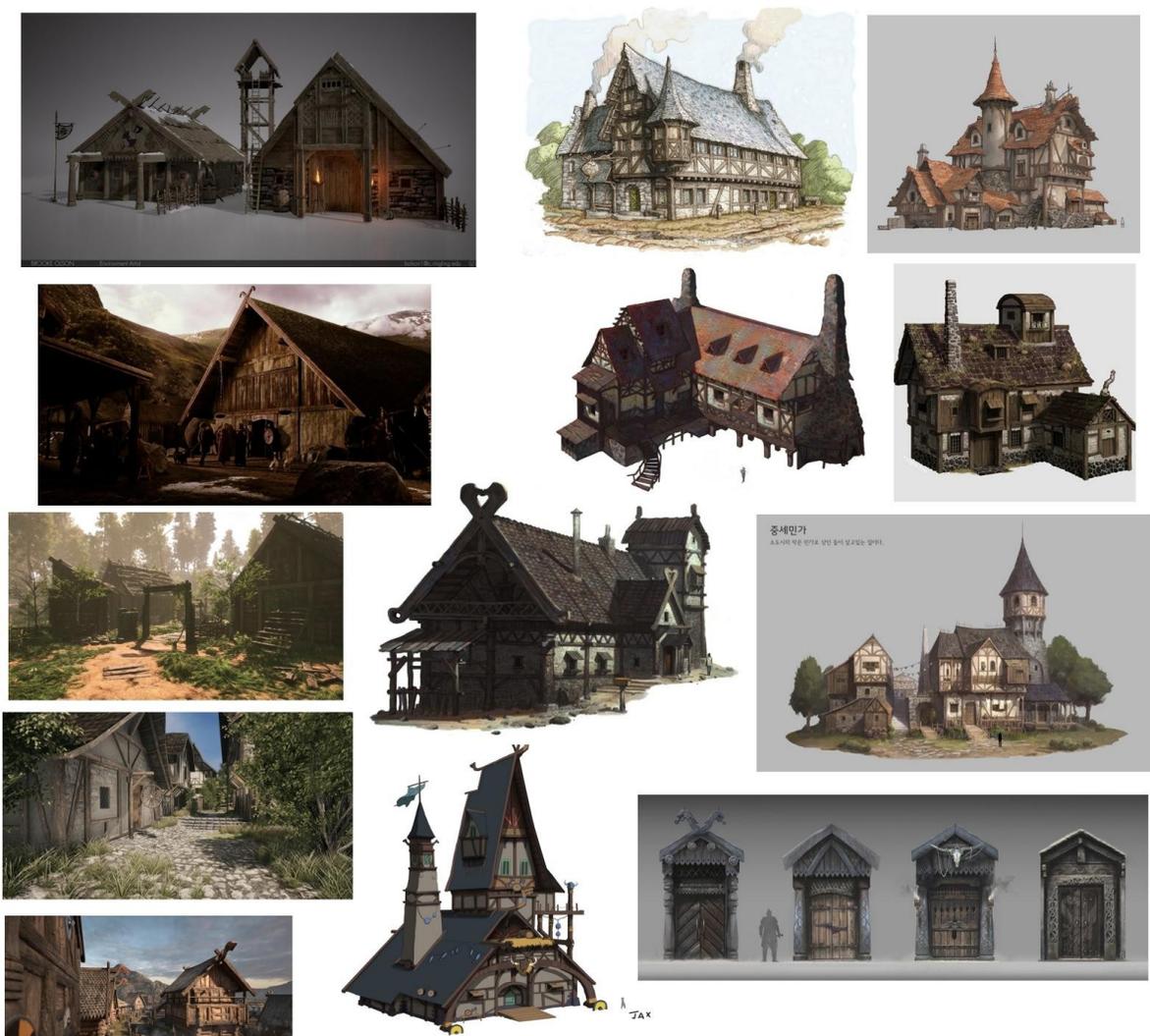


Figura 2 - Painel semântico criado para etapa de modelagem.

Fonte: montagem do autor

4.1.2 Modelagem

Durante esta fase foi estabelecido como objetivo a análise de todo o material de referência que foi acumulado dentro do painel semântico, para que fosse traçada uma linha de intersecção que fornecesse quais objetos dentre todas as imagens transmitiriam as ideias das palavras-chave escolhidas: medieval, fantasia e nórdico.

Sendo assim, alguns exemplos de *assets* desta lista que foram identificados são: o barril; diferentes tipos de caixas e pacotes; braseiros; tochas; prateleiras; variados tipos de mesas; cadeiras; bancos; carroça; bigorna; chaminé e lareira.

Ainda foram divididas outras duas seções dentro da parte da criação de objetos, sendo elas:

- a. Construções
 - i. Primitivos, que seriam os cubos, cilindros, outras formas simples que posteriormente seriam escalados para se tornarem as paredes, colunas, pilares, tábuas, ripas, telhados, etc;
 - ii. Blocos grandes, onde o bloco de construção modular estaria praticamente pronto ainda na parte de modelagem;
- b. *Assets*
 - i. De mesa, esta que compreendem coisas como canecas, pratos, potes, panelas e garrafas.
 - ii. *Props*, como os exemplos citados anteriormente: barril, caixa, mesa, etc.

O último registro desta etapa de produção foi a preocupação em se manter a proporcionalidade entre os diferentes *assets*, para que posteriormente na *Unreal* fossem utilizados na montagem dos blocos modulares. Alguns estudos de iluminação e de texturização foram realizados com intuito de se ter uma pré-visualização do que se teria como resultado posteriormente.

4.1.3 Revisões

Algumas das peças modeladas quando utilizadas em etapas posteriores apresentaram alguma deficiência no sentido de necessitarem que alguns polígonos possuíssem um material diferente dos demais, desta forma modificações foram feitas em objetos já modelados, para que quando exportados estes tivessem mais *material slots*. Outro momento em que iterações e consertos de problemas tiveram de ser feitos foram quando as faces de uma peça ficavam invertidas ou na parte de *UV mapping*. Ou até mesmo em etapas posteriores de desenvolvimento do projeto em que se julgou adequado adicionar *assets* que se mostraram necessários.

4.2 Segunda Etapa

Etapa referente a importar todos os modelos 3D criados e exportados como arquivos de extensão (*.fbx*) para o software *Unreal Engine 4* (versão 4.22), organizá-los como já descrito na etapa de modelagem, em suas subdivisões como demonstrado na Figura 3.

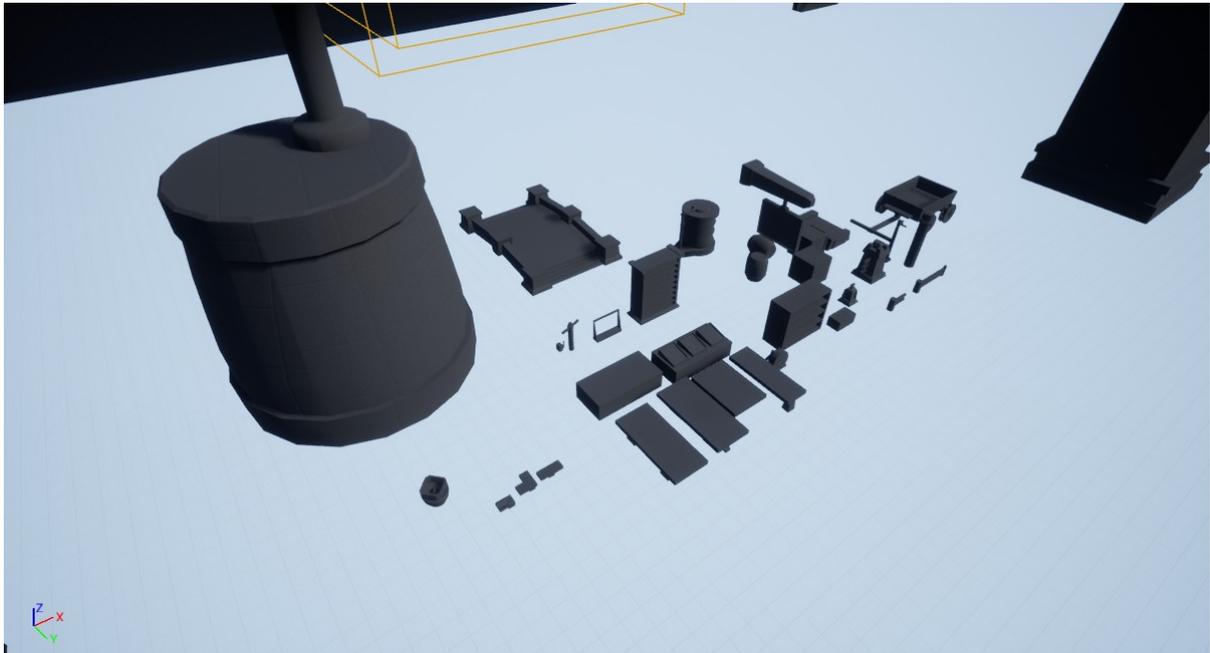


Figura 3 - Pacote de Modelos recém-importados.

Fonte: imagem do autor.

Com todos os modelos prontos para serem adicionados a parte de texturização e materiais e correções de problemas relativos a *lightmaps*³, se tornou possível a confecção do que foram chamados de: “*House Modules*” (blocos grandes como base), “*My Modules*” (criados inteiramente a partir de pequenos *assets*, cada tábuia de madeira, chegando a altas contagem de atores, podendo atingir até 450 em apenas uma construção) “*Primitive Modules*” (fazem um meio termo entre os dois anteriores, utilizando uma mistura entre blocos grandes para criar o “corpo” da estrutura e auxiliando os primitivos para dar identidade a eles). Definindo assim os três tipos de construções pré-moldadas que seriam trabalhados no projeto.

³ Os *lightmaps* são texturas mapeadas onde ficam salvos os dados calculados de iluminação da superfície dos objetos aos quais eles estão atrelados.

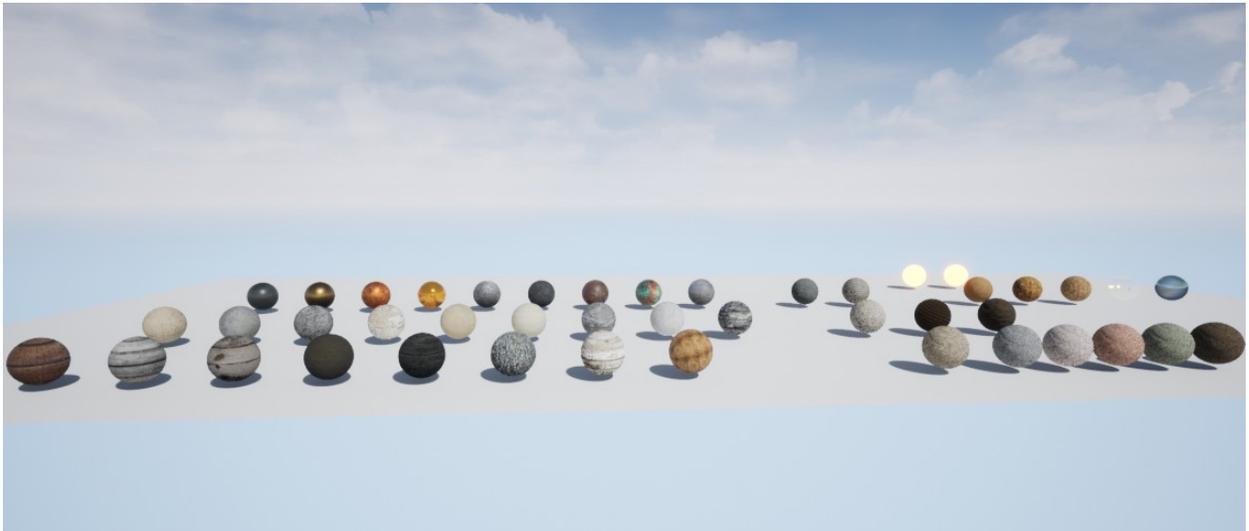


Figura 4 - Amostra dos materiais utilizados.

Fonte: imagem do autor.

Foi desenvolvida uma biblioteca de materiais (Figura 4) para serem usados neste momento de produção, bem como nas etapas que seguiriam, contendo mais uma vez o que seria o cerne do tema abordado no primeiro passo da produção: madeiras, metais, pedras e concreto. Acompanhando esta linha de pensamento foi desenvolvido em paralelo um segundo acervo, sendo este de *decals*⁴ demonstrado na Figura 5, que seriam posteriormente empregados como um meio de quebra de repetição, além de melhorar a comunicação entre o cenário e os módulos em que o *decal* seria aplicado.

⁴ Os *decals* são planos que podem ser posicionados sobre *static meshes* com características de materiais, porém esta sobreposição se utiliza da transparência de certas áreas, possibilitando assim a criar a ilusão de deformação em superfícies. Outra aplicação comum é sugerida pela própria nomenclatura, os adesivos, em placas, texto, etc.

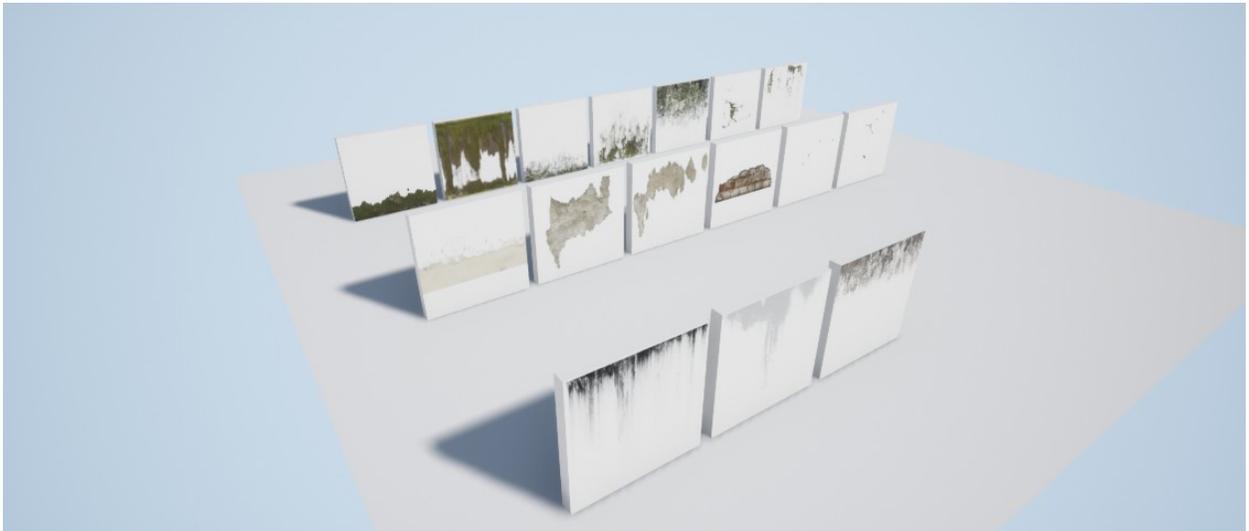


Figura 5 - Amostras dos *decals* criados.

Fonte: imagem do autor.

Finalmente foi criado um simples sistema de partículas⁵ para simular o fogo proveniente das tochas e braseiros, dividindo deste mesmo a parte referente à fumaça para ser posicionada sobre as chaminés acesas. Juntando todo este sistema em uma *blueprint*⁶ (fogo/fumaça) que ainda emitisse a luz de uma forma não uniforme, semelhante a uma chama verdadeira, atrelando o efeito sonoro do crepitar do fogo. Estes sistemas estão representados na Figura 6 a seguir.



Figura 6 - Amostra dos sistemas de partículas desenvolvidos.

⁵ Um sistema de partículas é constituído por uma fonte emissora, a qual serve como ponto de origem para unidades que tem suas características controláveis, como por exemplo a orientação e velocidade em que elas são disparadas do ponto de origem além de controles como aplicar um mapeamento de texturas para cada unidade ou até mesmo uma característica luminosa que varia de acordo com o tempo.

⁶ As *blueprints* são sistemas da *Unreal Engine* aonde podem ser agregados diferentes subsistemas, tais como luzes, emissores de partículas, efeitos sonoros, etc. Além disto é possível vincular às *blueprints* comandos de programação.

Fonte: imagem do autor.

4.3 Terceira Etapa

Prontos todos os *assets*, *props* e módulos, com seus devidos materiais, texturas e um primeiro estudo de iluminação em cenas internas e externas, iniciou-se mais uma etapa do projeto, que seria a aplicação destes dentro de um nível.

Dividiu-se este estágio em duas partes.

4.3.1 Pré-Produção

Segundo momento de conceituação, este referente ao cenário a ser criado: o “tema dentro do tema”, sendo montado um segundo painel semântico (Figura 7) para dar suporte referencial para esta etapa. Dentre as diversas combinações possíveis que poderiam ser desenvolvidas a partir dos módulos e objetos criados, ficou decidido que o nível seria o de uma pequena vila portuária, cercada por uma massa de água, com docas e com uma divisão semelhante à de camadas verticais.

Tendo em mente estas definições foram regredidos alguns passos na produção, para que pudessem ser confeccionados outras peças e ajustes em alguns dos módulos de construções para que estes ficassem mais alinhados com o conceito escolhido.

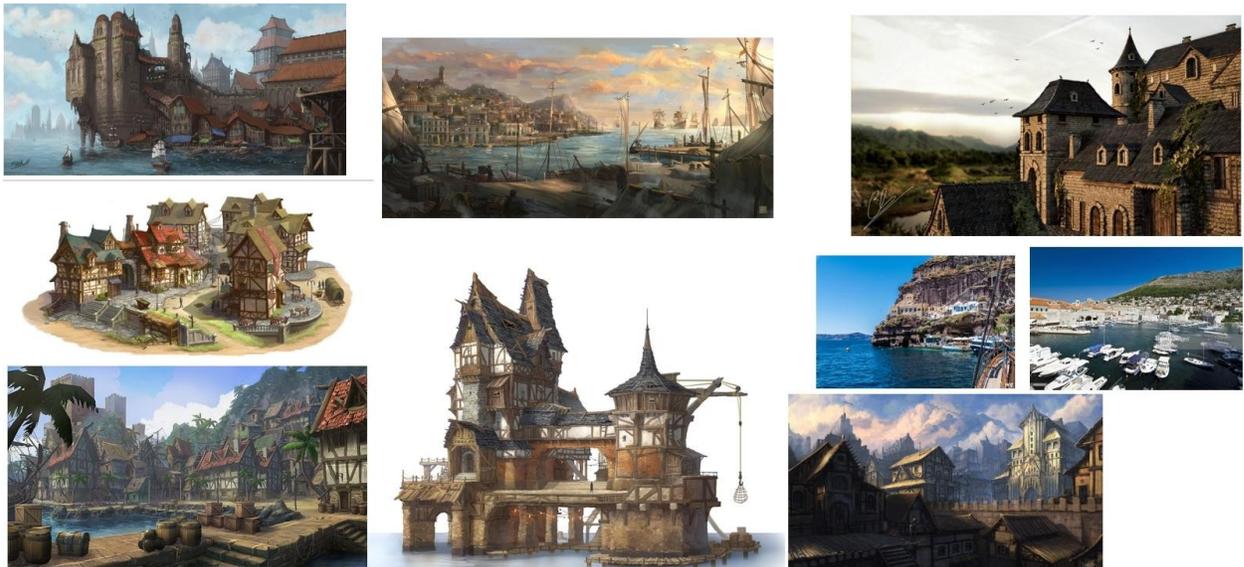


Figura 7 - Segundo painel semântico.

Fonte: montagem do autor.

4.3.2 Produção

Foi elaborada uma sequência de passos baseada em todo o material pesquisado e artigos reunidos como referência a serem seguidos para esta etapa de produção:

1. Baseado no painel semântico criar a blocagem deste cenário, definindo as limitações impostas ao nível. Além disso fazer um primeiro estudo de iluminação global com os blocos posicionados.
2. Trazer as construções do pacote de conteúdo, posicionando-as na cena, fazer a primeira revisão e iluminação global, fazendo alterações nas posições dos módulos e da fonte luminosa caso necessário.
3. Substituição dos blocos por *landscapes*⁷ nas superfícies por onde o jogador estará mais presente adicionando assim um maior nível de detalhamento e *static meshes*⁸ onde formas mais simples não terão um impacto visual negativo.
4. Inserção de detalhes para preenchimento do nível. Segunda revisão de iluminação.
 - a. Criação dos “*Merged Assets*” (basicamente ao invés de voltar para a etapa de modelagem e criar um único *asset*, montar este novo objeto a partir dos primitivos ainda na *Unreal*, como por exemplo, construir uma tenda a partir de ripas, planos, cilindros e cubos).
5. Adicionar os *decals* e folhagem.
6. Terceira revisão de iluminação global e interna.

⁷ Ferramenta de softwares de criação de níveis e jogos onde é criado um plano com controle de resolução da malha deste plano, podendo controlar materiais e texturas que são aplicados nos polígonos deste plano, além de um controle de deformação dos vértices gerando assim terrenos com elevações e declives.

⁸ *Static meshes* são peças geométricas que podem ser movimentadas, rotacionadas ou escaladas e suas informações são pré-computadas pela placa-de-vídeo tornando sua renderização mais eficiente. Com a observação que os vértices de *static meshes*, assim como o nome sugere não podem ser alterados.

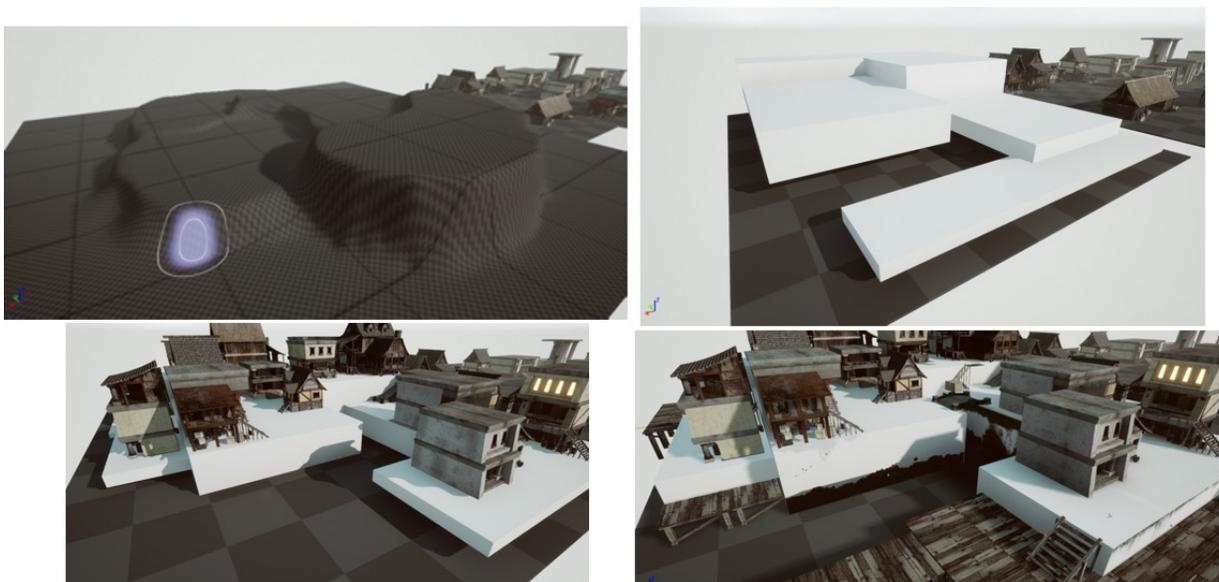


Figura 8 - Etapas do progresso do cenário.

Fonte: imagem do autor.

Entre todos estes passos (Figura 8) de produção foi feito um breve *play-test* do nível, para checar limitações que pudessem passar despercebidas, bem como problemas de escala tendo o personagem padrão da *Unreal* como base.

Neste momento também foi tomada a decisão do nomear este cenário como: *New Port*.

4.4 Quarta Etapa

Com o nível completo, utilizando ferramenta *Sequencer*⁹ seria montada uma animação no estilo *teaser* que viajasse pelo nível, demonstrando os principais pontos estudados no projeto e os toque finais da cena puderam ser feitos com a ferramenta de *post-produce volume*, esta que possibilita aplicar ajustes de cor, exposição, saturação, bem como simular diferentes efeitos projetados na tela do jogador, além de outras possibilidades.

Na Figura 9 é ilustrado uma opção simples presente na *Unreal Engine 4*, porém que foi de grande valor na parte de produção do vídeo, esta que permite separar a área de trabalho em terços.

⁹ Ferramenta do software Unreal Engine 4 que permite a animação de qualquer componente da cena, se tratando de static meshes a posição, rotação e escala podem ser animadas, câmeras possuem outras características que podem também ser animadas como por exemplo o foco da lente.



Figura 9 - Exemplo de enquadramento escolhido para *cinematic*.

Fonte: imagem do autor.

A fase final se resumiu a exportação de vídeo para que pudesse ser adicionada a parte sonora, que apesar do fato que dentro da própria *Unreal* pode ser criada uma ideia de volume e espacialidade com o uso de atores sonoros, a renderização do vídeo contendo áudio torna-se demasiada complexa, com a adição de vinheta, áudio, cortes e créditos seriam as últimas etapas a serem concluídas.

5 Resultados

5.1 Resultados - Modelagem

Terminada a primeira etapa de modelagem os resultados podem ser vistos como ilustrado na Figura 10 e Figura 11, estando os *assets e props* divididos por seções.

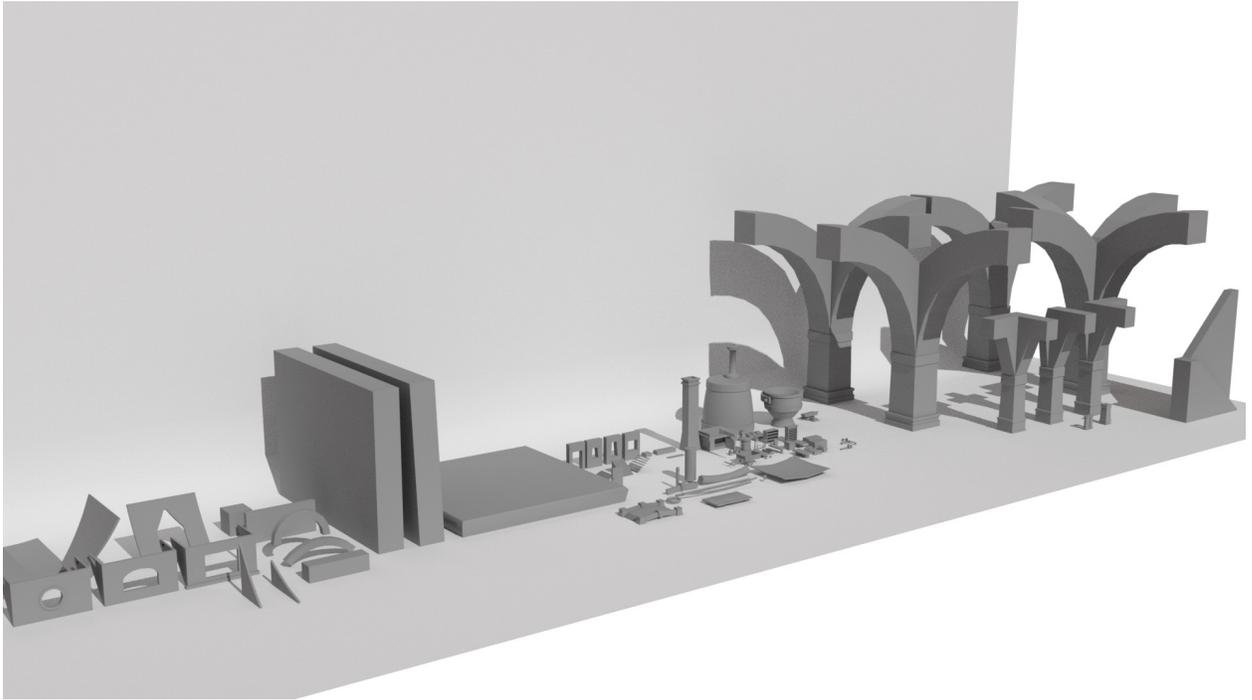


Figura 10 - Resultado 1 do pacote de modelos.

Fonte: imagem do autor.

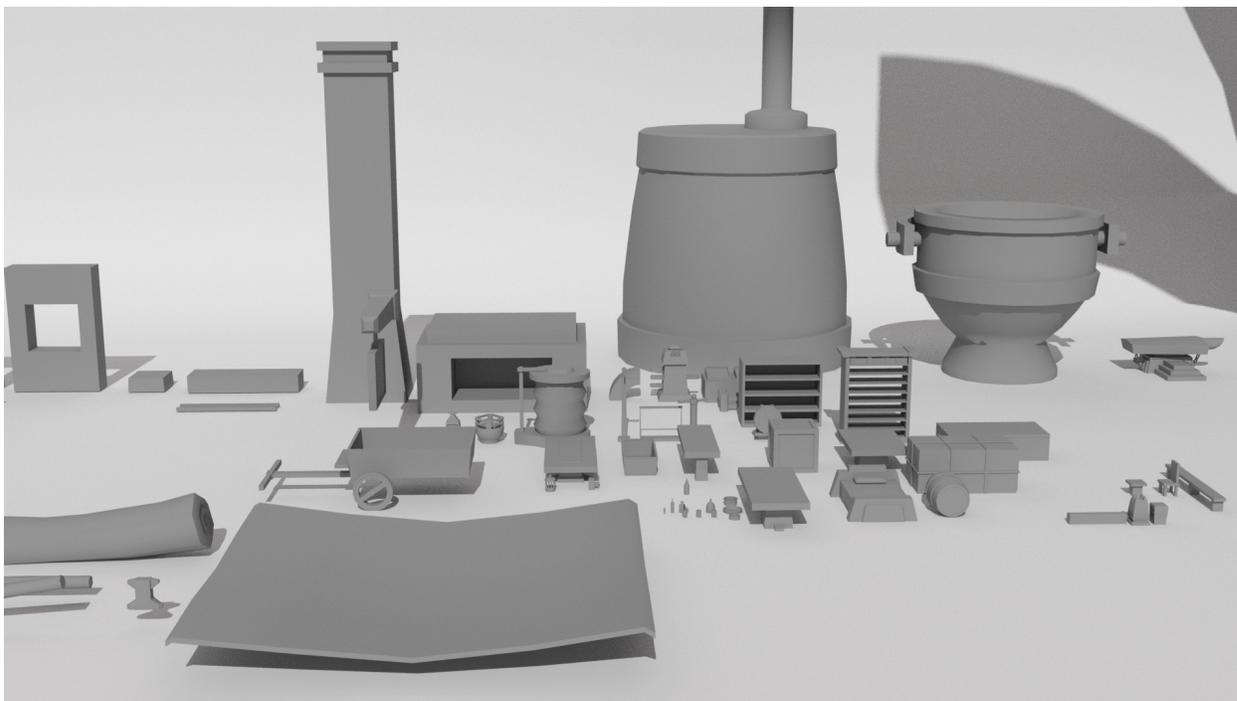


Figura 11 - Resultado 2 do pacote de modelos.

Fonte: imagem do autor.

Ainda dentro do software *3DS Max*, foram feitos alguns estudos de modelos, seguindo as referências adotadas no painel semântico, com o objetivo que se pudesse ter uma primeira ideia, do que eventualmente seria a etapa de construção de cenário como representado nas Figuras 12 e 13.

Ainda um pouco além do que já havia sido proposto, foi investido tempo adicional para montagem de um pequeno cenário, recriando uma cidade, gerando assim um conceito de composição com os modelos criados bem como a coesão entre os mesmos quando aplicados juntos, dentro de uma mesma cena.

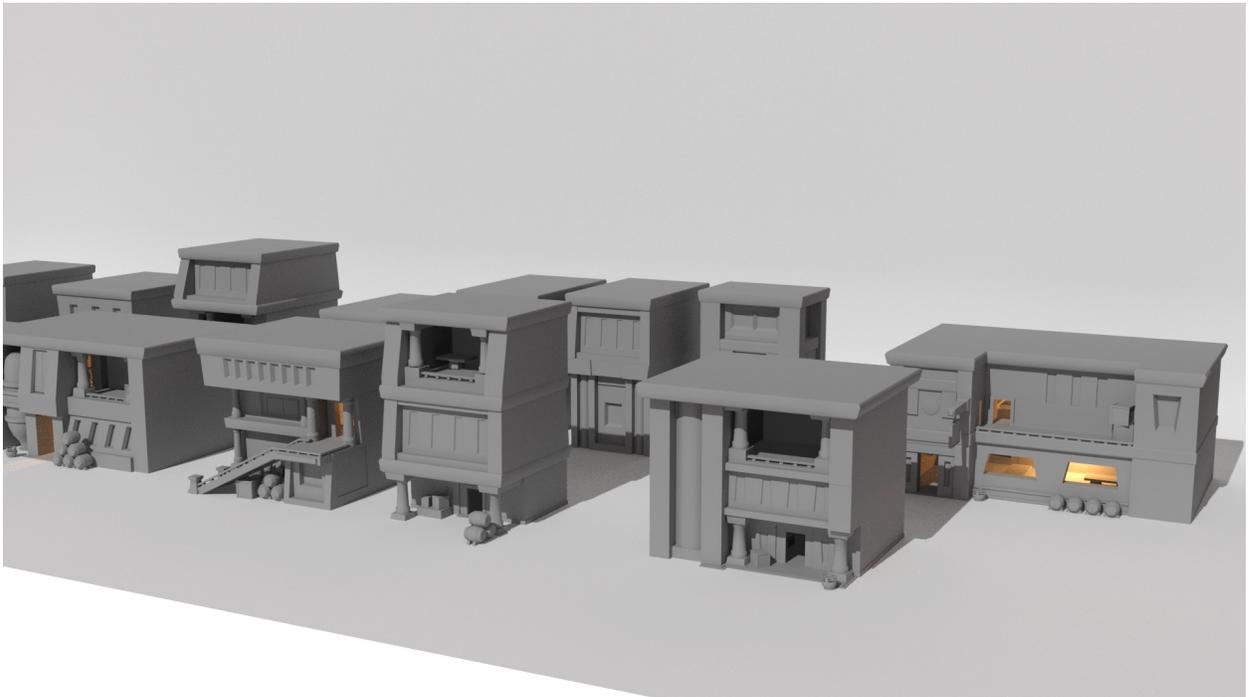


Figura 12 - Resultado dos módulos criados ainda no 3DS Max.

Fonte: imagem do autor.

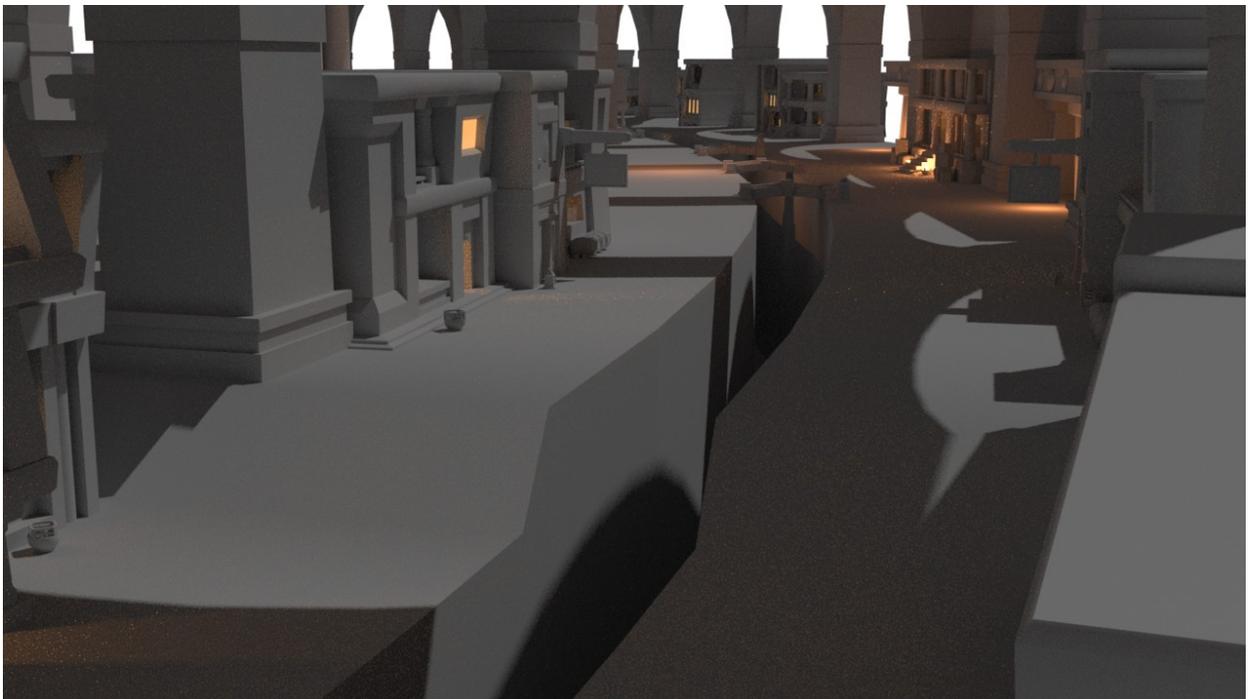


Figura 13 - Primeiro estudo de iluminação e composição.

Fonte: imagem do autor.

5.2 Resultados - Pacote de Conteúdo

Finalizada a biblioteca de materiais e com os *assets* importados foi possível montar o *level* de demonstração (Figura 14). Com a observação que os materiais selecionados para alguns dos objetos servem apenas como preenchimento para que fosse possível se ter noção do resultado da textura aplicada, tendo a possibilidade de corrigir erros na parte de criação das *UV* e *lightmaps* neste momento.



Figura 14 - Pacote de conteúdo pronto na *Unreal Engine*.

Fonte: imagem do autor.

Com a biblioteca de materiais pronta, o fato de utilizar um software como a *Unreal Engine* mostrou-se uma grande vantagem, dada a possibilidade de instanciamento de materiais para que se pudesse reduzir a necessidade de processamento e criação de um novo *draw call*¹⁰ para um material diferente, junto com uma série de outras texturas que dão suporte a este novo material. Desta maneira o objetivo era extrair um número significativo de possibilidades de cada material criado, tendo a biblioteca um total que girava em torno de 40 materiais: nove tipos de metais, nove tipos de materiais para construções como argamassa, gesso, pedra, concreto além de oito diferentes tipos de madeira. O instanciamento de cada material se mostrou vital, parametrizando os *inputs* de coloração, rugosidade, característica metálica e o *tiling*. Gerando o que pode ser visto na Figura 15, têm-se o material de tecido entrelaçado e cinco outras instâncias deste mesmo, apenas utilizando a variação de cor.



Figura 15 - Demonstração das possibilidades ao se usar instância.

Fonte: imagem do autor.

¹⁰ Uma draw call seria uma unidade de informação mandada do processador para a placa de vídeo que deve ser renderizada. No caso da Unreal Engine uma draw call é gerada para cada grupo de polígonos que compartilhem o mesmo material.

Outra funcionalidade dentro do software escolhido foi a possibilidade de combinar múltiplas *static meshes* dentro uma nova que carregue a informação contida nas partes escolhidas. Com o foco nessa opção foi montada outra biblioteca, esta que foi chamada de “*Merged Assets*” como representado na Figura 16, contendo uma variedade de *props* já organizados para serem inseridos dentro do cenário que foi montado. Esta escolha resultaria em consequências: ao se mesclar múltiplas *meshes* em uma única, o novo *lightmap* a ser criado em alguns casos necessita de uma resolução maior, tornando-se notável na definição das sombras ao se usar uma resolução abaixo do ideal. Além deste contratempo, a única possível customização que poderia ser feita a partir do momento em que os objetos fossem combinados, seria a troca de materiais, pois para cada material dentro dos objetos selecionados foi criado um *material slot* no novo. Um exemplo em que essa “limitação” se tornou algo que ajudou, foi na criação de pilhas de *props* mistas, como diferentes caixas e barris, caso combinados com diferentes tipos de materiais, a *mesh* final tinha uma gama ainda maior de combinações possíveis.



Figura 16 - Demonstração dos *Merged Assets*.

Fonte: imagem do autor.

A mesma função de mistura foi utilizada para o desenvolvimento de ainda mais uma biblioteca. Nesse caso foi a chamada: Módulos Pré-Moldados, que consistia basicamente da reunião de todas as construções criadas. Ao invés de conterem centenas de atores cada uma (algo que na etapa de construção se mostrou um verdadeiro problema nos momentos em que era necessário calcular a iluminação, além da organização de todos estes atores) foram convertidas para uma única *mesh* cada exemplificado na Figura 17 e 18.

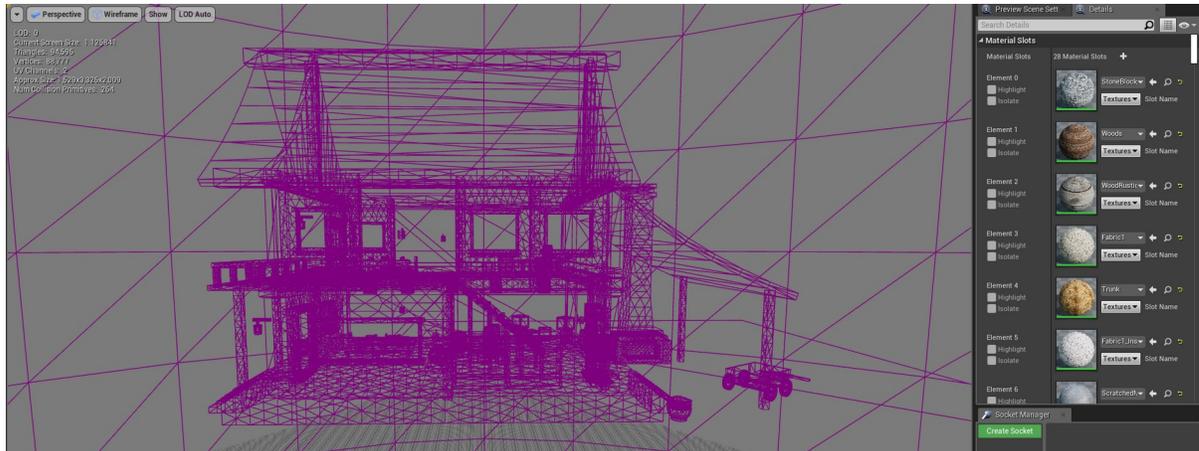


Figura 17 - Visualização do resultado ao se combinar um dos módulos pré-moldados.

Fonte: imagem do autor.

Mais uma vez outro fator limitante foi encontrado. Algo que deveria ser evidente no princípio acabou passando despercebido: a ferramenta apenas combina *static meshes*, ou seja, *decals*, partículas, iluminação ou *blueprints* não podem ser inseridos. Neste sentido foi “perdida” uma quantidade de tempo considerável para que estas customizações pudessem ser feitas novamente.



Figura 18 - Exemplos dos módulos.

Fonte: imagem do autor.

O resultado de todos estes módulos pré-moldados pode ser verificado na Figura 19, sendo que alguns dos presentes neste nível de demonstração não foram utilizados na construção do cenário de New Port.



Figura 19 - Demonstração de todos módulos criados.

Fonte: imagem do autor.

Ainda se tratando da etapa referente à confecção do cenário, foram coletados dados (Tabela 1) para ilustrar as motivações por trás de algumas decisões tomadas neste processo sendo algumas delas por questões de otimização, outras por limitações ou até mesmo por organização.

Módulo Pré-Moldado	Contagem de Atores (<i>pre-merge</i>)	Contagem de Material Slots (<i>post-Merge</i>)	Contagem de Triângulos
<i>MyModule01</i>	284	28	94.596
<i>MyModule02</i>	358	26	170.986
<i>MyModule03</i>	183	20	84.650
<i>MyModule04</i>	446	30	217.784
<i>HouseModule02</i>	223	20	97.051
<i>HouseModule14</i>	60	16	16.790
<i>PrimitiveModule01</i>	376	28	151.266

<i>PrimitiveModule04</i>	197	22	76.216
--------------------------	-----	----	--------

Tabela 1 - Comparação entre atores, material slots e triângulos.

Ao se posicionar a câmera longe o suficiente de modo que capture o cenário inteiro, tem-se um total aproximado de nove milhões de triângulos, isto sem o corte de objetos à distância e trabalhando com apenas um nível de detalhe, por se tratar de uma cena concentrada. Ao se caminhar pelo cenário, a média de triângulos em cada quadro fica entre um e 1,5 milhões. A título de comparação com os *assets* que compõem o cenário, algumas outras contagens de triângulos: a tenda completa com as caixas, balcão e prateleiras têm 12.824; o telhado com duas janelas ficou com 6.680 e uma sacola com duas cordas amarrando-a 768. A média de triângulos por construção pré-moldada ficou em torno de 55 mil.

Neste mesmo viés, a questão do número de *draw calls* dentro da cena. Foram coletadas algumas amostras: a primeira, como um ponto de estresse máximo, ao se visualizar (o nível como um todo) o ponto limite atingido foi de aproximadamente dois mil. Na segunda foi realizado um *play-test* navegando pelas ruas e construções (buscando ângulos de câmera mais comuns), chegando a uma média entre quinhentos e seiscentos *draw calls*, tornando-se claro neste momento que a opção de criar apenas uma *static mesh* para as casas pré-moldadas realmente trouxe algum benefício na redução deste número.



Figura 20 - Comparação na resolução das sombras projetadas.

Fonte: imagem do autor.

5.3 Resultados - Aplicação em Cenário

Uma das barreiras encontradas nesta terceira etapa do projeto foi a aplicação dos sistemas de partículas e *blueprint* de iluminação criados no cenário, principalmente quando se tratava de ambientes internos com os raios de atenuação de mais de uma fonte luminosa estando interseccionados, como demonstrado na Figura 21. Com a investigação do problema, foi possível detectar que a fonte causadora era a característica emissiva atrelada a partículas que simulavam o fogo de tochas e braseiros. Para contornar esta situação foi tomada a decisão de manter este sistema para a criação da *cinematic*, pois o resultado visual da combinação entre *blueprint* (uma luz pontual que varia o valor de intensidade seguindo uma senóide) e este sistema de partícula se aproximaria ao real. Porém para a versão jogável do nível seria desligado o fator emissivo.

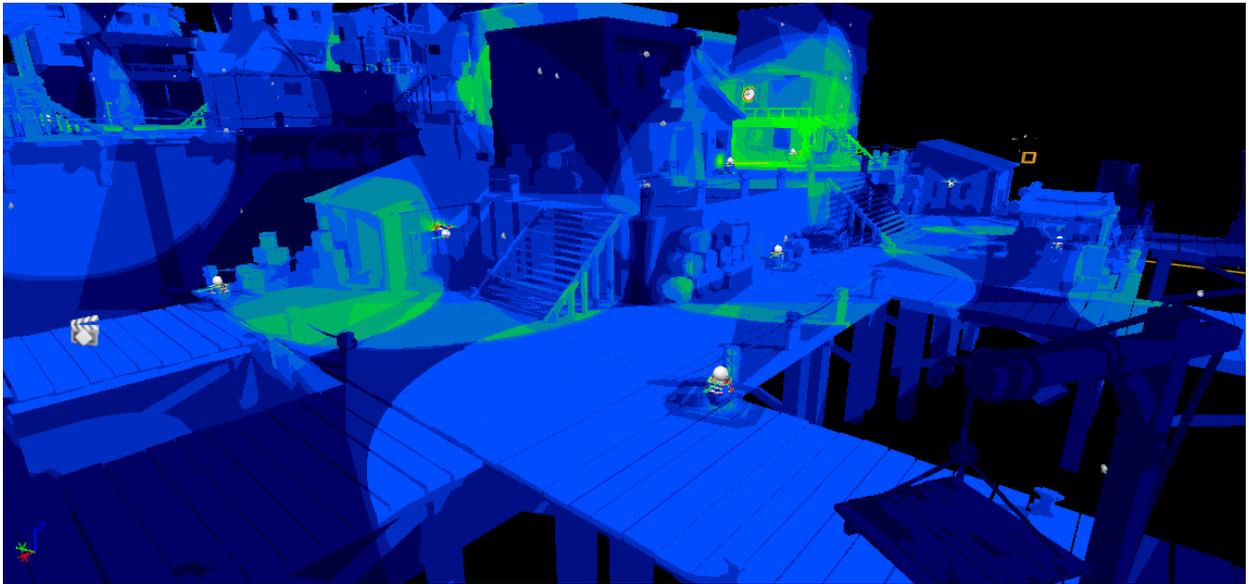


Figura 21 - Cenário utilizando o filtro de complexidade de iluminação.

Fonte: imagem do autor.

Em questão de resultados numéricos, foram analisados os dados do tempo de resposta da *GPU* para que fossem calculados os elementos do cenário, demonstrando neste momento que a maior porção do tempo foi utilizada para montar a iluminação da cena, juntamente com as sombras projetadas pelas fontes luminosas. Entrando em mais detalhes ainda foi possível identificar que a maior porção deste processamento estava sendo direcionado para o *light source*, flutuando entre valores dois até 2.5 *ms*. Além disso, foi detectado que cada *blueprint* de iluminação (“*Flickering Lights*” como foram chamadas) representava um aumento de aproximadamente 0.08 *ms*, sendo que em alguns piores casos estes valores poderiam chegar até 0.25 *ms*, exemplificado na Figura 22. Com estes dados ficou decidido que estas formas de iluminação ainda seriam aplicadas, contanto que fossem reduzidas a um mínimo possível, buscando-se ao máximo utilizar a iluminação proveniente do *light source*, “Sol”.

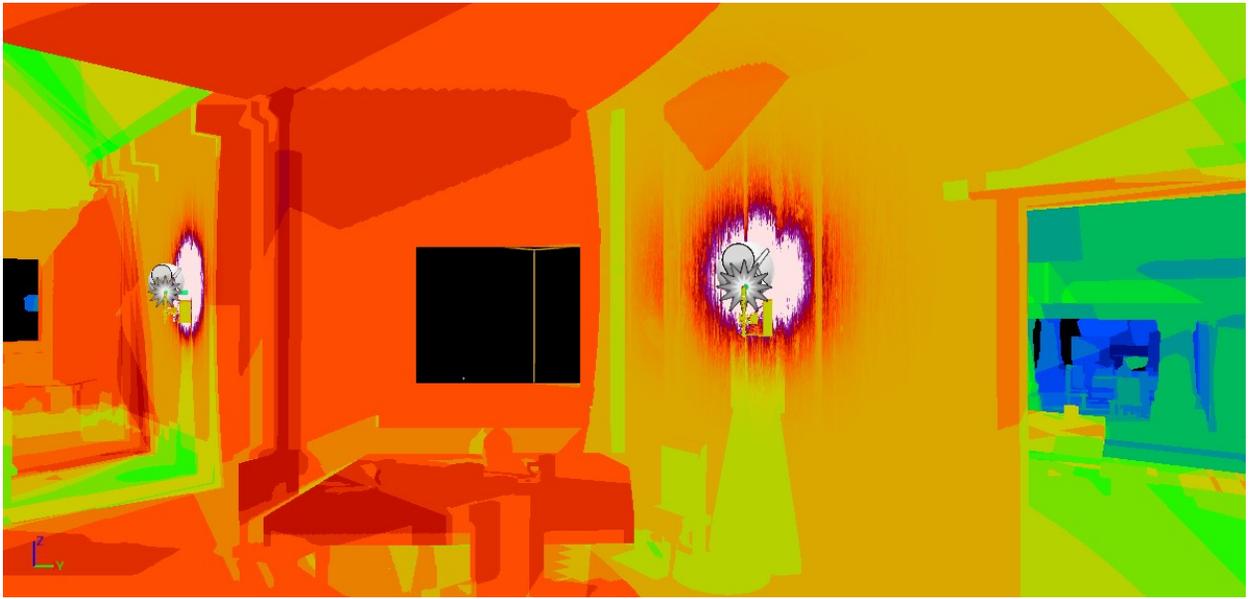


Figura 22 - Resultado de se usar um sistema de partículas com característica emissiva.

Fonte: imagem do autor.

Como um último ponto a ser salientado em opções artísticas versus aplicações otimizadas, foi a questão de se ter um cenário cercado por uma massa de água, pesquisou-se na biblioteca de aprendizado fornecida pela própria *Unreal*, para que se pudesse entender como funcionam os materiais que simulam oceanos, lagos e rios. Com o entendimento veio a revelação de mais um dilema a ser vencido: este material simula as propriedades da água utilizando translucência, opacidade baseada em altura relativa e também efeitos Fresnel para se obter uma aparência mais realista. Todos estes aspectos aumentavam a complexidade dos *shaders* envolvidos. Neste caso não foi possível se encontrar uma solução mais elegante, ficando definido que para o mar que cerca a área jogável do cenário bem como uma pequena fonte em uma das plataformas, seriam utilizados os materiais que possuíam um nível de complexidade médio-alto, como representado na Figura 23 e 24 a seguir.



Figura 23 - Cenário com filtro da complexidade de *shaders*.

Fonte: imagem do autor.

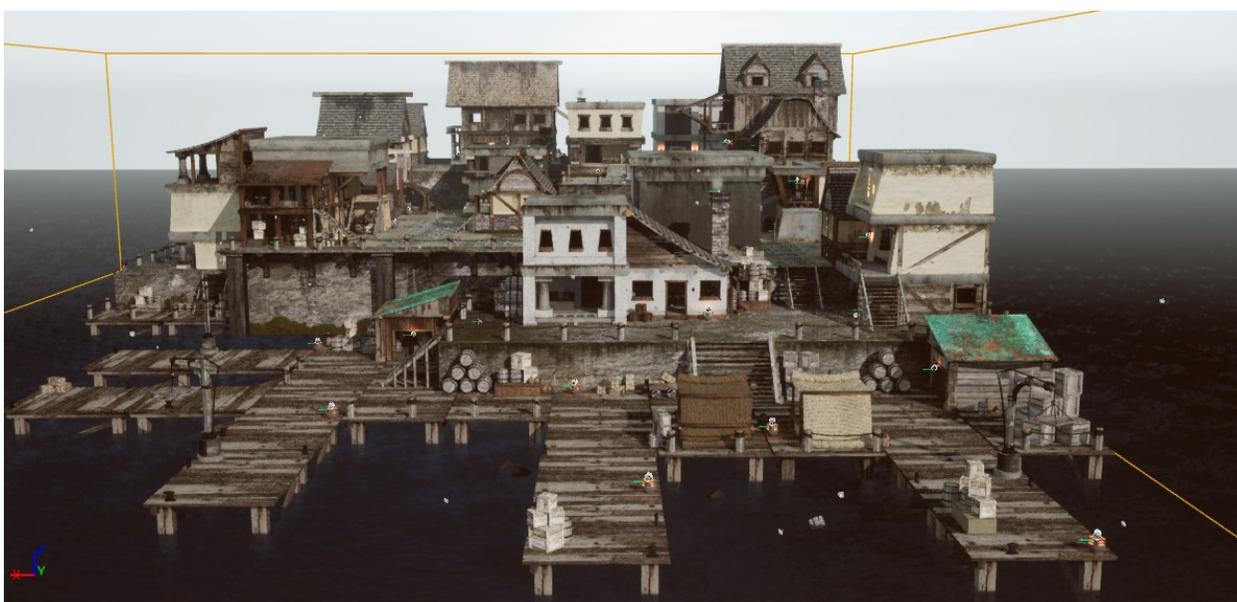


Figura 24 - Cenário completo.

Fonte: imagem do autor.

5.4 Resultados - Criação de Cinematic

O último passo da produção foi o momento em que a decisão de se usar o software da *EPIC Games*, a *Unreal Engine* mostrou-se mais uma vez ser uma enorme vantagem, pois ao se calcular todos os componentes do sistema (iluminação, geometria, reflexos, etc) foi possível montar a *cinematic* e renderizá-la em tempo real, tendo a opção de fazer alterações e ajustes e

receber uma resposta rápida. Ainda se tratando deste aspecto da produção, a ferramenta de *Sequencer*, possibilita uma solução fácil e integrada, com a interface semelhante a maioria dos softwares de edição de vídeo, a animação do movimento de câmera, junto com distância focal e outras ferramentas (Figura 25). Como único ponto negativo, apesar de ser possível adicionar uma trilha de áudio, na parte de renderização ainda há problemas, para que estas faixas sejam exportadas. Sendo assim, a edição final com a inserção da parte sonora, vinheta e créditos foi feita em outro software, o *Adobe Premiere*.



Figura 25 - Cenário pela lente da câmera utilizada.

Fonte: imagem do autor.

6 Conclusão

Com o trabalho finalizado, ao se olhar para trás e analisar os passos da produção é possível se fazer algumas últimas considerações, tais como apontar o quão essencial e determinante foi iniciar o trabalho com antecedência, cerca de três meses anteriores ao semestre letivo, com esta antecipação as etapas iniciais do projeto, tais como a organização do cronograma, conceituação e reunião de referências para o primeiro painel semântico, além de primórdios dos primeiros modelos confeccionados no *3DS Max* estavam completos quando o segundo semestre de 2019 começou.

Um segundo ponto importante a ser analisado, foram as dificuldades técnicas encontradas em diferentes momentos da produção, com menção especial para dois aspectos em específico: sendo o primeiro, configurações específicas no processo de exportar os modelos do *3DS Max*, para que quando importados para *Unreal*, fosse possível aplicar dois, ou mais, tipos de material para uma mesma *static mesh* e o segundo ponto em que o processo de produção foi desacelerado, foram os momentos em que se teve que aprender uma nova ferramenta do software *Unreal*, recorrendo a seção de aprendizado e fóruns da própria *EPIC Games*, exemplos disso sendo as ferramentas: *Merge Actors*, *Sequencer*, *BSP Brushes*, *Blueprints*, Efeitos de Partículas, entre outros.

Outro aspecto que pode ser percebido, olhando-se para as diversas etapas concluídas durante o projeto, foi a presença dos conhecimentos acumulados nas disciplinas oferecidas no decorrer do curso de Animação. De uma forma geral, a partir do primeiro semestre todas as matérias cursadas tiveram em algum momento da produção relevância e contribuíram para que os objetivos pudessem ser alcançados. Exemplos dessa importância são: já citado anteriormente gestão de projetos, *design* de cenários, modelagem *3D*, arte para *games*, *rendering* de animação, entre outros.

Este trabalho de conclusão de curso além de um projeto acadêmico serviu como uma oportunidade de ampliação dos conhecimentos introduzidos durante os quatro anos de graduação, proporcionando uma experiência única que estabeleceu uma ponte entre a base dos ensinamentos transmitidos pelos professores e o processo real de uma etapa do desenvolvimento de jogos digitais.

Como última consideração a ser feita os objetivos propostos ao se iniciar este projeto não poderiam ter sido alcançados com êxito, obviamente, sem o cronograma, mas principalmente sem

a disponibilidade de tempo para ser investido nas etapas de produção, em momentos chegando a carga horária diária de até dez horas para que se pudesse concluir algumas das metas semanais estabelecidas. Desta forma os dois fatores que determinaram o sucesso desta “jornada” foram: o estabelecimento de metas alcançáveis e o planejamento das etapas de produção.

7 Referências

- BOSCO, Laura. **Be more productive as a digital project manager (here's how!)**. 2018. Disponível em: <<https://thedigitalprojectmanager.com/more-productive-dpm/>>. Acesso em 20 de maio de 2019 às 13:25.
- EPIC GAMES. **Actors merging**. 2018. Disponível em: <<https://docs.unrealengine.com/en-US/Engine/Actors/Merging/index.html>>. Acesso em 25 de set. de 2019 às 13:41.
- EPIC GAMES. **Packaging Projects**. 2018. Disponível em: <<https://docs.unrealengine.com/en-US/Engine/Basics/Projects/Packaging/index.html>>. Acesso em 25 de set. de 2019 as 13:43.
- EPIC GAMES. **Perfomance guideline for artists and designers**. 2018. Disponível em: <<https://docs.unrealengine.com/en-US/Engine/Performance/Guidelines/index.html>>. Acesso em 22 de set. de 2019 às 18:34.
- GALUZIN, Alex. **Guide to blocktober**. 2019. Disponível em: <https://www.worldofleveldesign.com/categories/level_design_tutorials/guide-to-blocktober.php>. Acesso em 12 de out. de 2019 às 11:38.
- GALUZIN, Alex. **How to create a map in 11 days**. 2011. Disponível em: <<https://coachstech.files.wordpress.com/2011/06/how-to-create-a-map-in-11-days-using-udk.pdf>>. Acesso em 12 de out. de 2019 às 13:15.
- GRUPO ABRIL. *Mercado de games no Brasil deve crescer 5,3% até 2022 diz estudo*. **EXAME**, 3 ago. 2019. Disponível em: <<https://exame.abril.com.br/negocios/mercado-de-games-no-brasil-deve-crescer-53-ate-2022-diz-estudo/>>. Acesso em 5 de ago. de 2019 às 09:37.
- LITTLEFIELD, Andrew. *The beginner's guide to Scrum and Agile Project management*. 2016. Disponível em: <<https://blog.trello.com/beginners-guide-scrum-and-agile-project-management>>. Acesso em 21 de maio de 2019 às 08:27.

SAKUDA, Luiz Ojima; FORTIM, Ivelise (Orgs.). **II Censo da Indústria Brasileira de Jogos Digitais**. Ministério da Cultura: Brasília, 2018. Disponível em: <<http://tinyurl.com/censojogosdigitais>>. Acesso em 12 de nov. De 2019 às 20:58.

SIMON, Joanna Leigh. **How to meet deadlines without micromanaging**. 2018. Disponível em: <<https://thedigitalprojectmanager.com/how-meet-deadlines-without-micromanaging/>>. Acesso em 20 de maio de 2019 às 14:02.

STOUT, Mike. **A beginner's guide to designing vídeo game levels**. 2016. Disponível em: <<https://gamedevelopment.tutsplus.com/tutorials/a-beginners-guide-to-designing-video-game-levels--cms-25662>>. Acesso em 13 de out. de 2019 às 13:28.

SUTHERLAND, J.; SCHWABER, K. **Scrum Guides**, 2017. Disponível em: <<https://scrumguides.org/index.html>>. Acesso em 12 de nov. de 2019 às 10:53.

UNIVERSITAT POMPEU FABRA – BARCELONA & MUSIC TECHNOLOGY RESEARCH GROUP. **Freesound**, 2019. Página inicial. Disponível em: <<https://freesound.org/>>. Acesso em 30 de set. de 2019 às 09:05.

WIJMAN, Tom. **The Global Games Market Will Generate \$152.1 Billion in 2019 as the U.S. Overtakes China as the Biggest Market**. 2019. Disponível em: <<https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market/>>. Acesso em 12 de nov. De 2019 às 18:26.