

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**DESENVOLVIMENTO DE UM JOGO DIGITAL 3D DO GÊNERO RPG**

**ÉRICO JUNIOR DE MORAIS  
SAMUEL DE SOUZA SILVA**

**ANÁPOLIS  
2018**

**ERICO JUNIOR DE MORAIS  
SAMUEL DE SOUZA SILVA**

**DESENVOLVIMENTO DE UM JOGO DIGITAL 3D DO GÊNERO RPG**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientadora: Prof. Me. Luciana Nishi.

Anápolis  
2018

**ERICO JUNIOR DE MORAIS**  
**SAMUEL DE SOUZA SILVA**

## **DESENVOLVIMENTO DE UM JOGO DIGITAL 3D DO GÊNERO RPG**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 11 de Dezembro de 2018, composta por:



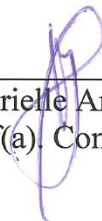
---

Luciana Nishi  
Presidente da Banca



---

Alexandre Moraes Tannus  
Prof(a). Convidado(a)



---

Millys Fabielle Araujo Carvalhaes  
Prof(a). Convidado(a)

Dedicamos a todos que sonham em criar jogos,  
profissionalmente ou como hobby.

## **AGRADECIMENTOS**

Agradecemos a Deus por nos acompanhar nessa jornada nos dando saúde e força para superar as dificuldades.

Aos pais, namoradas e familiares pelo amor, apoio e incentivo dado.

A nossa orientadora pela confiança, suporte e orientações.

A todos que direta ou indiretamente contribuíram para que concluíssemos este trabalho, nosso muito obrigado.

*“A percepção do desconhecido é a mais fascinante das experiências. O homem que não tem os olhos abertos para o misterioso passará pela vida sem ver nada. ”*

*(Albert Einstein)*

## RESUMO

Este trabalho discorre sobre o processo de desenvolvimento de jogos digitais, através de um projeto real de criação de um jogo, abordando desde sua concepção até a implementação, destacando atividades a serem desempenhadas e metodologias de desenvolvimento importantes na criação de jogos. Ferramentas especializadas para construção de jogos são definidas, técnicas de modelagem e animação são expostas e artefatos necessários ao desenvolvimento são criados.

**Palavras-chave:** Jogos 3D. Desenvolvimento de Jogos Digitais. RPG. *Game Design*.

## **ABSTRACT**

This work discusses the process of developing digital games, through a real game creation project, from conception to implementation, highlighting activities to be performed and important development methodologies in game creation. Specialized tools for building games are defined, modeling and animation techniques are exposed, and necessary artifacts for the development are created.

**Keywords:** 3D Games. Development of Digital Games. RPG. *Game Design*.



## LISTA DE ILUSTRAÇÕES

Figura 01 – Game Waterfall Process, adaptada.....	16
Figura 02 - Fluxo <i>Extreme Programming</i> – Adaptado pelos autores para XGD. ....	18
Figura 03 - Processo de design de jogos .....	22
Figura 04 - Malha de um carro gerada pela modelagem NURBS.....	23
Figura 05 - Modelo poligonal sendo subdividido várias vezes. ....	24
Figura 06 - Arquitetura de Sistemas de Jogo do Projeto. ....	34
Figura 07 - Modelagem com o auxílio de imagem como referência.....	37
Figura 08 - Edição da textura de um personagem com o Photoshop. ....	38
Figura 09 - Processo de mapeamento UV dentro do Blender. ....	39
Figura 10 - Processo de animação utilizando <i>bones</i> e quadros-chaves com o Blender.....	40
Figura 11 - Montagem de cenário com o Unity.....	42

## LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões
3D	Três dimensões
GDD	<i>Game Design Document</i>
GPL	<i>General Public License</i>
HUD	Heads-Up Display
MMORPG	<i>Massive Multiplayer Online Role Playing Game</i>
NPC	<i>Non-Player Character</i>
NURBS	<i>Non-Uniform Rational Basis Spline</i>
PC	<i>Personal Computer</i>
RPG	<i>Role-Playing Game</i>
XDG	<i>Extreme Game Development</i>
XP	<i>Extreme Programming</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	14
2.1	JOGOS RPG ( <i>ROLE PLAYING GAME</i> )	14
2.2	JOGOS DIGITAIS	15
2.3	METODOLOGIAS DE DESENVOLVIMENTO DE JOGOS	15
<b>2.3.1</b>	<b>Game Waterfall Process</b>	16
<b>2.3.2</b>	<b><i>Extreme Game Development</i></b>	17
2.4	ARQUITETURA DE SOFTWARE	19
2.5	ATIVIDADES NO DESENVOLVIMENTO DE JOGOS	19
<b>2.5.1</b>	<b>Game Design</b>	20
<b>2.5.2</b>	<b>Modelagem 3D</b>	22
2.5.2.1	Modelagem NURBS	23
2.5.2.2	Modelagem poligonal	24
<b>2.5.3</b>	<b>Animação</b>	24
<b>2.5.4</b>	<b>Texturização</b>	25
2.6	FERRAMENTAS DE AUXÍLIO	26
<b>2.6.1</b>	<b>Criação Artística</b>	26
2.6.1.1	Blender 3D	26
2.6.1.2	Maya	27
2.6.1.3	Photoshop	27
<b>2.6.2</b>	<b>Game Engine</b>	28
2.6.2.1	Unity 3D	28
2.6.2.2	Unreal Engine	29
<b>3</b>	<b>DESENVOLVIMENTO</b>	31
3.1	METODOLOGIA DE DESENVOLVIMENTO	31
3.2	CONCEPÇÃO	32
3.3	PREPARAÇÃO	33
3.4	IMPLEMENTAÇÃO	36
<b>3.4.1</b>	<b>Modelagem 3D</b>	36
<b>3.4.2</b>	<b>Texturização</b>	38
<b>3.4.3</b>	<b>Animação</b>	39
<b>3.4.4</b>	<b>Sonorização</b>	40

<b>3.4.5 Codificação .....</b>	<b>40</b>
<b>3.4.6 Construção do mundo do jogo.....</b>	<b>41</b>
<b>4 CONSIDERAÇÕES FINAIS.....</b>	<b>44</b>

## 1 INTRODUÇÃO

O desenvolvimento de jogos digitais possui algumas diferenças em relação ao desenvolvimento de *softwares* tradicionais como sites e sistemas de gestão por possuir atividades específicas a serem desempenhadas e conhecimentos específicos sobre jogos necessários ao desenvolvedor. De acordo com Esper (2012), inúmeras etapas fazem parte do processo de desenvolvimento de jogos, dentre elas, a concepção do enredo contendo a história, criação de arte envolvendo imagens, construção de cenários e modelagens 2D (Bidimensional) ou 3D (Tridimensional), músicas e sons para o ambiente do jogo.

Dentre os vários tipos de jogos, existem os do gênero RPG (*Role-Playing Game*), em que jogadores interpretam papéis de personagens de uma história narrada (ARRUDA, 2014, p. 50), vivem aventuras em mundos de fantasias em ambientes medievais, futuristas, entre outros (PAVÃO, 2000). Jogos de RPG geralmente necessitam de estratégia por parte dos jogadores, cada ação realizada, cada caminho tomado pode acarretar em um evento diferente e impactar positivamente ou negativamente o personagem do jogador. As aventuras dos jogos de RPG contêm enigmas, desafios, e situações que exigem que os jogadores tomem decisões estratégicas (PAVÃO, 2000). Como o desenvolvimento de jogos possui atividades além das já desempenhadas na criação de softwares tradicionais e existem vários tipos de jogos com diversas características, então como é o desenvolvimento de um jogo digital 3D do gênero RPG?

Tem-se como objetivo apresentar conceitos, ferramentas utilizadas, descrever atividades envolvidas, e artefatos gerados durante o desenvolvimento de um jogo digital 3D. Tem como estilo o gênero RPG, mas pode ser utilizado como base para qualquer tipo de jogo.

De acordo com a empresa PricewaterhouseCoopers (2014 apud GEDIGames e NPGT), os jogos digitais em relação a outras alternativas de entretenimento estão apresentando as maiores taxas de crescimento. Estimando que suas vendas sejam maiores que as da indústria fonográfica e com um crescimento médio previsto de 6,2% ao ano, crescendo com mais rapidez que o setor cinematográfico até o ano de 2018. Com a expansão do mercado de jogos, a demanda por profissionais desenvolvedores de jogos aumenta, gerando oportunidades para ingressar na área tanto como um colaborador em uma empresa como um empreendedor em sua *startup*.

Este trabalho foi elaborado com base em pesquisas bibliográficas realizadas sobre o processo de desenvolvimento de jogos digitais, envolvendo diversas fontes de pesquisa, como livros de autores conceituados, artigos publicados na internet, vídeo aulas e fóruns. Alguns

jogos digitais de RPG, foram utilizados como fundamento para compreensão de suas características através da experiência de jogo, a fim de que esse conhecimento possa ser utilizado na construção deste trabalho.

Na primeira parte deste trabalho se encontra a fundamentação teórica, em que tópicos sobre a temática do trabalho são abordados com base em referências, que agregam ao leitor conhecimento necessário para o melhor entendimento de todo o trabalho e que tornam possível aos autores seu desenvolvimento. A segunda parte é a seção de Desenvolvimento, onde é descrito todo o processo realizado no desenvolvimento deste projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Desenvolver um jogo exige muito conhecimento, não somente de codificação e *design*, mas também de vários outros como efeitos especiais e sonoros, modelagem e animação, metodologia de desenvolvimento e definição e especificação da ideia do jogo e suas características.

Com a finalidade de conhecer as necessidades para o desenvolvimento do jogo objetivo deste trabalho, sua fundamentação teórica aborda assuntos relacionados a jogos digitais do estilo RPG, para que haja entendimento sobre o tipo de jogo que será construído; Metodologias de desenvolvimento de jogos, que são uma forma de conduzir o processo de desenvolvimento de maneira profissional e que auxilia na construção de um produto de qualidade; *Game design*, assunto que permitirá conhecer como criar um guia para os desenvolvedores durante a implementação; *Softwares* que podem ajudar na obtenção de melhor performance na implementação do projeto; E conceitos abordados sobre jogos que possibilitarão conhecer melhor as necessidades para o desenvolvimento de um jogo.

### 2.1 JOGOS RPG (*ROLE PLAYING GAME*)

O tipo de jogo que será construído neste trabalho será o de gênero RPG, um estilo de jogo com características bem específicas que poderão ser notadas no decorrer dos assuntos abordados neste trabalho.

Segundo Arruda (2014, p. 50), o gênero de jogo RPG permite que os participantes criem narrativas e assumam papéis de personagens por exemplo de guerreiros, magos, reis e joguem colaborativamente. Neste gênero existe um sistema de regras que devem ser respeitadas e fluxos que devem ser seguidos durante a partida, mas também há aberturas para que o jogador realize improvisações e interprete o papel de seu personagem.

Arruda (2014, p. 50) relata que um dos principais títulos de jogos que deu origem ao gênero RPG que pode ser citado é o jogo de tabuleiro *Dungeons & Dragons* que teve sua origem em 1974. O gênero RPG evoluiu com o tempo, saindo um pouco dos jogos de mesa e ganhando espaço no ambiente virtual com os jogos digitais, primeiro com jogos de um único jogador vivenciando aventuras, logo se expandiu e também virou multijogador sendo jogado em rede em um mundo virtual com milhares de jogadores, nascendo então os MMORPGs (*Massive Multiplayer Online Role Playing Game*).

## 2.2 JOGOS DIGITAIS

Jogos digitais são jogados em dispositivos eletrônicos como computadores, *tablets*, consoles e smartphones, criados geralmente com computadores (ARRUDA, 2014, p. 3). Os jogos estão em constante evolução, em cada lançamento percebe-se que as histórias estão mais envolventes, ficam mais divertidos e com o ininterrupto aprimoramento de *hardware*, pode-se notar que a qualidade gráfica sempre está melhorando.

Os jogos digitais são compostos por diversos elementos dos quais podem ser destacados as histórias, os personagens, cenários, sons e efeitos especiais. Mitterofhe (2016, p. 16), diz que "Sua criação requer método, disciplina, ideias, roteiros, recursos tecnológicos, *softwares* específicos e a capacidade de transformar a experiência interativa do ato de jogar a mais divertida possível." Tendo isto em vista, é correto dizer que os desenvolvedores possuem a responsabilidade de entregar um produto final de qualidade para o público, e para que isso seja possível é importante que a produção de jogos digitais assim como a de *softwares*, utilizem processos de produção adequados. Esses processos podem ser definidos com metodologias de desenvolvimento, algumas tradicionais que podem se adequar a projetos específicos e demandar mais tempo de desenvolvimento e outras ágeis para um desenvolvimento rápido.

## 2.3 METODOLOGIAS DE DESENVOLVIMENTO DE JOGOS

Segundo Clinton (2010, p. 6), a indústria de jogos digitais começou pequena, as equipes eram reduzidas e a necessidade de metodologias não era tão grande, mas com a grande expansão da área isso mudou e fez com que a utilização de uma metodologia se tornasse essencial para os desenvolvedores. Com a utilização de metodologias os desenvolvedores passam a ter um roteiro a ser seguido, com métodos específicos, visando produtividade no desenvolvimento e a criação de um produto de qualidade.

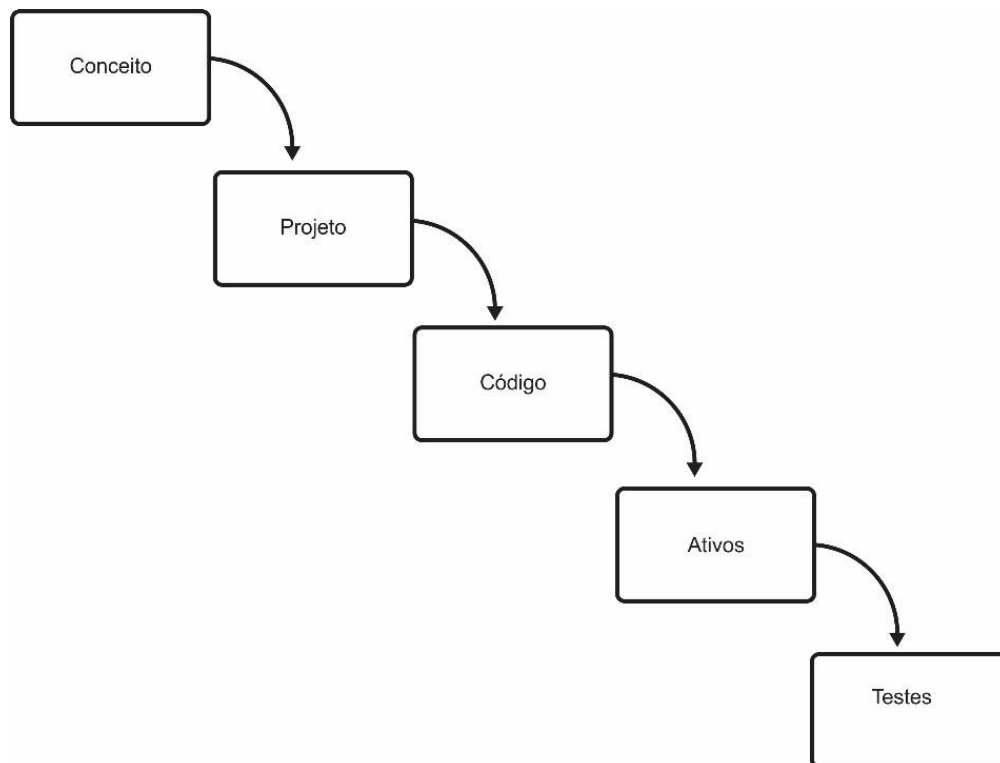
De acordo com Carvalho (2013, p. 27), as metodologias utilizadas no desenvolvimento de *softwares* devem ser empregadas aos jogos e seu uso ajuda a garantir uma maior qualidade final do produto. Porém ele alerta que tais metodologias devem sofrer alterações para se aplicar a projetos de jogos eletrônicos, pois jogos são tipos de softwares que possuem algumas características específicas que devem ser trabalhadas diferentemente das gerais. Algumas metodologias de desenvolvimento de jogos foram criadas a partir de adaptações de metodologias usadas comumente em *softwares* tradicionais, em que pode-se destacar a *Game Waterfall Process* e a *Extreme Game Development*.



### 2.3.1 Game Waterfall Process

A *Game Waterfall Process* é uma metodologia derivada do modelo de desenvolvimento de *software* cascata. Segundo Clinton (2010, p. 6), muitos projetos de jogos usam uma abordagem em cascata para o desenvolvimento. No processo de desenvolvimento de *software* cascata, todas as atividades acontecem de forma sequencial, uma atividade só é iniciada ao terminar a anterior, conforme pode ser visualizado na figura 01. No *Game Waterfall Process* além de serem incluídas mais atividades a serem trabalhadas em relação ao cascata, desenvolvedores realizam adaptações no modelo cascata para trazer mais flexibilidade ao projeto, utilizam uma cascata incremental e até voltam em uma fase anterior enquanto outra está sendo realizada.

Figura 01 – Game Waterfall Process, adaptada.



Fonte: Clinton (2010).

Dependendo do projeto do jogo, um processo sequencial não seria a melhor solução a ser utilizada, pois se imaginar um jogo com alto volume de conteúdo sendo criado por uma grande equipe, chegando na fase de testes que seria a última e constatarem que houve um erro na interface que afetaria os fluxos de vários conteúdos criados, seria um grande prejuízo monetário e de tempo caso precisassem realizar alterações para a correção. Em projetos assim outros tipos de metodologias poderiam ser mais ideais, como uma metodologia ágil,

metodologia que trabalha com um processo incremental, que não necessita aguardar a finalização de todo o processo para poder retornar a uma etapa específica e que permita trabalhar em duas etapas em paralelo.

### 2.3.2 *Extreme Game Development*

No desenvolvimento de jogos digitais, a XP é popularmente conhecida como *Extreme Game Development*, metodologia que herda suas características, mas é aplicada ao desenvolvimento de jogos. Se baseia nos mesmos cinco princípios ou valores da XP que são comunicação, simplicidade, *feedback*, coragem e respeito (DEMACHY, 2003), e utilizada como uma alternativa para a diminuição no tempo de desenvolvimento de jogos. A XGD trabalha com a adaptação do modelo XP ao desenvolvimento de jogos, envolvendo a criação de conteúdo, mecânica de jogo, elementos visuais e sonoros além da parte de programação em que há mais ênfase na XP.

A *Extreme Programming* (programação extrema), é uma metodologia ágil de desenvolvimento criada por Kent Beck em seu trabalho na Crysler, (STEPHENS e ROSENBERG, 2003, p. 39) e (TELES, 2014, p. 286-287), possui a orientação a objetos como seu paradigma de desenvolvimento preferido, envolvendo regras e práticas no contexto de suas atividades metodológicas também chamadas de atividades-chave: planejamento, projeto, codificação e testes (PRESSMAN, 2011, p. 88-90) que são listadas a seguir e podem ser visualizadas no fluxo da XP conforme figura 02:

**Planejamento:** É a etapa em que ocorre o levantamento de requisitos e a busca pelo entendimento do ambiente de negócio do software. Realizando também a criação de histórias de usuários que mapeiam as características e as funcionalidades dos requisitos levantados para o software.

**Projeto:** No projeto, a XP segue o princípio de preservar a simplicidade, é preferível um projeto simples do que um complexo. Produz poucos artefatos, pode e deve ser continuamente modificado à medida em que o desenvolvimento evolui.

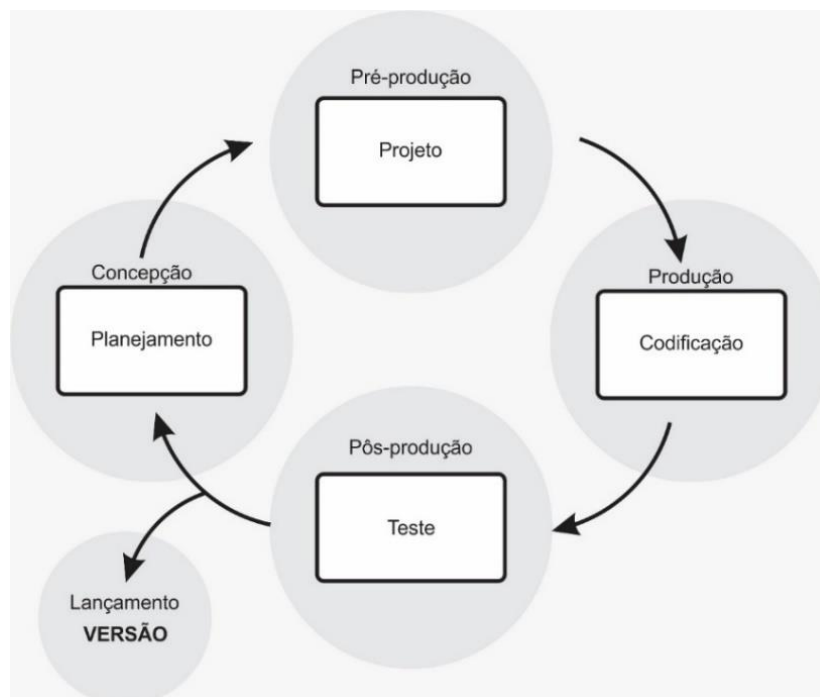
**Codificação:** Com as histórias de usuários criadas e o projeto pré-elaborado, é realizado a criação de testes de unidades para cada uma das histórias definidas para a iteração atual do processo de desenvolvimento, que é incremental. Com os testes planejados, inicia-se a codificação e teste da unidade codificada, que provê um rápido *feedback* para os desenvolvedores.

A XP aconselha a programação em dupla, a conhecida programação em pares, que proporciona uma forma de resolução de problemas e garantia da qualidade em tempo real, pois duas mentes pensando é melhor que uma.

**Testes:** Um elemento chave na abordagem da metodologia XP é a criação de testes de unidade antes da codificação, que devem ser implementados de forma a serem automatizados.

Na XP também ocorrem os testes de aceitação, criados conforme as histórias de usuários implementadas. São os testes de cliente e mantém o foco na funcionalidade do sistema.

Figura 02 - Fluxo *Extreme Programming* – Adaptado pelos autores para XGD.



Fonte: Pressman (2010, p. 88).

As metodologias ágeis de desenvolvimento foram criadas para agilizar o processo de desenvolvimento de *software*, possibilitando entregar soluções em menor tempo e de forma gradual. De acordo com Albino, Souza e Prado (2014, p. 16), a proposta das metodologias de desenvolvimento ágil é promover a adaptação e o fortalecimento do trabalho em equipe aliado a organização e agilidade em entregas.

Segundo Demachy (2003), os motivos pelos quais os jogos sempre estão atrasados são porque a tecnologia evolui muito rapidamente, mas leva tempo para aprender e adaptar em projetos; Desenvolvedores sempre querem fazer o melhor, criar o código mais otimizado, os modelos 3D mais sofisticados, querem entregar o jogo apenas quando está totalmente completo e os editores e ou clientes sempre mudam de ideia no meio do projeto. Reagir a mudanças

segundo o autor, é quase impossível se você estiver seguindo uma bíblia de design de jogos de centenas de páginas ou que todas suas entregas de funcionalidades do jogo sejam definidas antecipadamente. Como o XGD é um dos métodos ágeis de desenvolvimento de software, possui um princípio importante que é responder a mudanças mais que seguir um plano, então ao surgir alguma necessidade de mudança no projeto não haveria frustrações pois já se está esperando e trabalhando com essas possibilidades. E como entregar o jogo somente quando está totalmente concluído pode causar atrasos e gerar problemas como mudanças no escopo, o XGD herdando princípios do XP prega a entrega de resultados iterativa e incremental, entregando *features* regularmente e com frequência.

## 2.4 ARQUITETURA DE SOFTWARE

Softwares podem sofrer várias alterações em seus ciclos de vida. Uma arquitetura bem desenvolvida ajuda a reduzir o acoplamento de componentes, permitindo que novas funcionalidades sejam implementadas sem problemas, além de que com uma boa documentação a manutenção se torna muito mais fácil.

Uma arquitetura de software é um modelo abstrato que tem como objetivo descrever a estrutura de elementos que compõe um software e o seu funcionamento (LEITE, 2000). A utilização de um modelo abstrato torna possível visualizar e documentar a estrutura e o funcionamento de um software independente da linguagem de programação utilizada.

A arquitetura é definida através do projeto de arquitetura de software, em que ocorre a identificação dos principais componentes estruturais e os relacionamentos entre eles (SOMMERVILLE, 2011, p. 103). É importante haver harmonia entre todos os componentes do sistema, nenhum pode ser prejudicado por outro. Os dados precisam chegar ao destino final com integridade.

## 2.5 ATIVIDADES NO DESENVOLVIMENTO DE JOGOS

Assim como em qualquer processo, na criação de jogos há muitas atividades envolvidas, algumas presentes no desenvolvimento de softwares tradicionais como levantamento de requisitos, codificação e testes, mas também há atividades mais voltadas ao desenvolvimento de jogos e simulações virtuais.

### 2.5.1 Game Design

O desenvolvimento de um bom jogo começa com uma boa ideia, mas nem sempre é fácil tê-la, para isso é necessário muito esforço e dedicação dos desenvolvedores. Segundo Schuytema (2008, p. 22) “Não importa como elas surgem, a geração de ideias é uma das tarefas centrais[...]”.

O *game design* é a principal fase no desenvolvimento de jogos e é considerada a espinha dorsal do projeto, é nela que são definidos todos os pontos do jogo e serve como guia para todo o time de desenvolvimento como cita Schell, (2015, p. 24) "*O game design* é o ato de decidir o que o jogo deve ser".

Nessa etapa do desenvolvimento são descritas as informações mais importantes do jogo, jogabilidade, controles, personagens, interfaces, movimentos, objetos, fases e outras características importantes para o jogo. De acordo com Perucia et, al. (2005, p. 30) o *game design*:

[...] em geral, é o que determina a jogabilidade, as escolhas que o jogador terá dentro do mundo do jogo e as ramificações que suas escolhas vão ter no resto do jogo. Inclui o que faz o jogador vencer ou perder, como ele vai controlar o jogo, as informações que o jogador deverá receber.

Assim como no desenvolvimento de *softwares* institucionais e comerciais, no desenvolvimento de jogos também se utiliza a engenharia de *software*, porque um jogo digital não deixa de ser um *software*, necessita de todos os processos da engenharia de *software* para ser construído corretamente, processos que são introduzidos na criação do *Game Design Document* (GDD). Vasconcelos e Ribeiro (2013, p. 46) diz que este documento deverá permitir e facilitar converter os requisitos e o design do software numa construção efetiva que determinará a fase de codificação, implementação e integração do projeto de *software*. Alguns tópicos que devem conter em um GDD são a descrição inicial do jogo e dos seus objetivos, de forma que toda a história do jogo e seus objetivos sejam apresentadas. A apresentação das características do jogo para a sua potencial audiência e apresentar o desenvolvimento conforme um processo de engenharia de *software*.

Segundo Mastrocola (2012, p. 34), “se o desafio criado não divertir ou valer a pena, com certeza será descartado pelos jogadores”. Então o jogo deve ser bem planejado e seus aspectos bem definidos para que possa satisfazer as expectativas dos jogadores. É importante desenvolver um jogo com boa qualidade gráfica, controles de personagens eficientes, conteúdo que agrade e envolva o jogador no mundo do jogo.

McGonigal (2011, p. 21), apresenta uma perspectiva sobre *game design* bastante interessante ao citar que todos os *games* são definidos por quatro aspectos essenciais: 1) O objetivo: pelo que os jogadores deve buscar durante o jogo; 2) As regras e mecânicas: ditam o funcionamento do jogo e do ambiente de jogo, definindo como os jogadores podem atingir os objetivos; 3) O sistema de resposta: a comunicação do *game* que mostra aos jogadores seu status dentro da busca pelos objetivos; 4) A participação voluntária: que tem como definição que os jogadores conhecem o jogo e o aceitam. Essa visão mostra pontos importantes a serem considerados no desenvolvimento de um jogo e que devem ser mapeados no *game design* a fim de ser implementados.

De acordo com Adams (2014, p, 31), a criação de um jogo digital além de necessitar do lado criativo também envolve planejamento, processos e métodos. A abordagem defendida pelo autor é centrada no jogador e acredita-se que essa seja a base para o desenvolvimento de um bom *game design*, nessa abordagem o designer se coloca no papel de jogador ou cria um jogador imaginário do jogo que se deseja criar e dessa forma ele cria um jogo de acordo com os seus anseios enquanto jogador. Essa abordagem demonstra a importância de se desenvolver o produto pensando como seu usuário final, buscando-se eliminar problemas como por exemplo descobrir que o jogo proposto não tem chances de aceitação.

Adams (2014, p. 45), propõe um processo de design de jogos composto por três estágios: a concepção e a elaboração, fases em que são extraídos mais detalhes da ideia do jogo e a fase de detalhamento ou refinamento onde os ajustes são feitos para melhorar as ideias. Seguindo este processo, as ideias estarão melhor formadas, bem definidas, evitando perda de tempo em uma ideia mal elaborada. A figura 03 ilustra o processo.

No estágio de concepção são feitas escolhas de sensações estéticas e é criada o *high concept*, o *game design* deve ter sua primeira versão criada nessa fase, além disso outros documentos devem ser criados nessa fase como cronograma. Durante essa fase em projetos de jogos é comum a realização de reuniões para captação de ideias.

A fase de elaboração representa boa parte do desenvolvimento, é nela que terá início o desenvolvimento de documentos importantes e o refinamento do *game design*, geralmente nesse momento os problemas de planejamento, as complicações na comunicação entre a equipe começam a aparecer.

No estágio de detalhamento são realizados detalhamentos nas atividades que fazem parte do desenvolvimento para que essa possa ser entregue.

Figura 03 - Processo de design de jogos



Fonte: Adams (2014, p. 45)

## 2.5.2 Modelagem 3D

Algo que contribuiu para a grande popularização de jogos digitais foi a modelagem tridimensional ou modelagem 3D. De acordo com Pereira (2008, p. 6), “A modelagem 3D é uma subárea da computação gráfica que tem como objetivo a criação de objetos em três dimensões”. As técnicas utilizadas para produção de objetos 3D são capazes de trazer bons resultados na simulação de objetos realísticos e é esse fator que a faz tão usada atualmente, tendo em vista que um dos objetivos dos produtores de jogos é trazer realidade a seus jogos.

Em 1992 com o surgimento de novas tecnologias e com a redução dos preços de *hardwares* necessários para o desenvolvimento de alto nível, houve a quebra dimensional dos jogos 2D para os 3D, tendo como primeiro sucesso para plataforma de PCs o jogo *Wolfenstein 3D* (CLUA e BITTENCOURT, 2005, p. 15). Apesar de até hoje jogos 2D ainda serem criados, os grandes títulos estão sendo desenvolvidos em 3D.

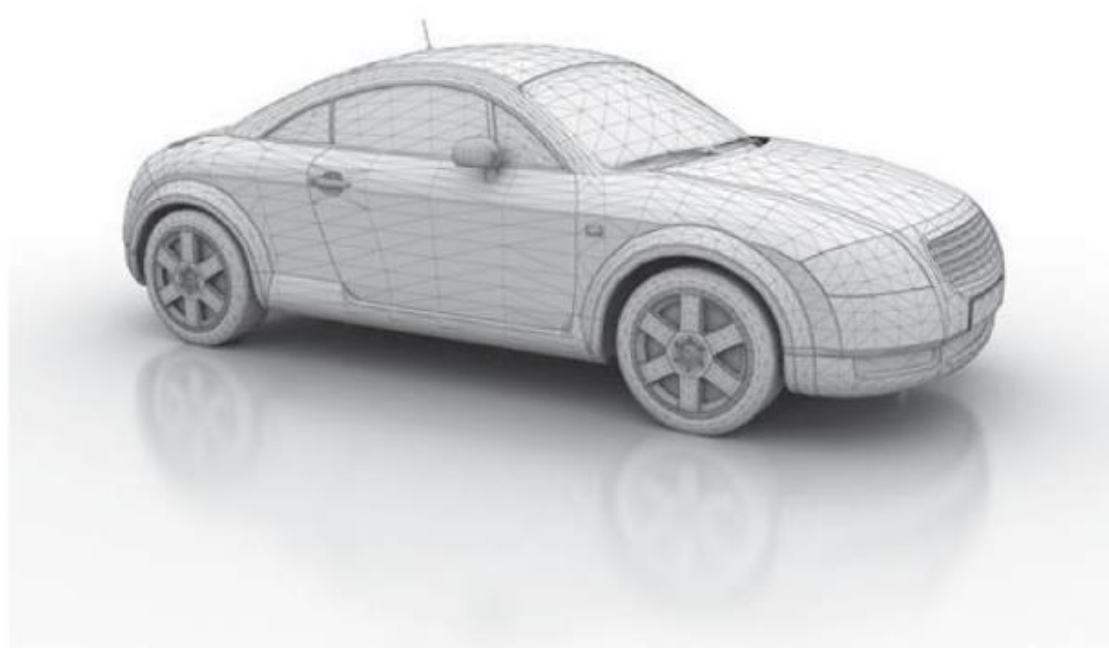
O surgimento de *softwares* cada vez mais robustos alavancados pela melhora computacional proporcionou ao cinema muita evolução, dando vida a novos formatos de filmes, unindo o real a modelos digitais, tais como *Exterminador do Futuro 2* em 1991, *Jurassic Park* em 1993 e o primeiro filme totalmente 3D *Toy Story* em 1995. (PEREIRA, 2008, p. 7). Essa última produção foi revolucionária, as técnicas de modelagem 3D subiram de nível e a criação de personagens tanto para a indústria cinematográfica como para a de jogos digitais também melhorou agregando muito do conteúdo desenvolvido inicialmente na *Pixar Animation*.

Segundo Andaló (2015, p. 20), existem atualmente diversas técnicas para modelagem, mas essas variam conforme as ferramentas. O autor nos apresenta dois caminhos para produção de malhas 3D, a modelagem NURBS e modelagem poligonal.

### 2.5.2.1 Modelagem NURBS

Modelagem NURBS (*Non-Uniform Rational Basis Spline*) surgiu nos anos de 1950 com o objetivo de representar matematicamente curvas complexas, em que linhas são geradas tornando sua curva uma constante, não dependente de segmentos (ANDALÓ, 2015, p. 20). A superfície gerada é sempre curva e suave devido não haver subdivisões e sua curva ser constante, diferente da modelagem poligonal que necessita de várias subdivisões para se conseguir uma superfície curva.

Figura 04 - Malha de um carro gerada pela modelagem NURBS.



Jezper/Shutterstock.com

Fonte: Andaló (2015, p. 21)

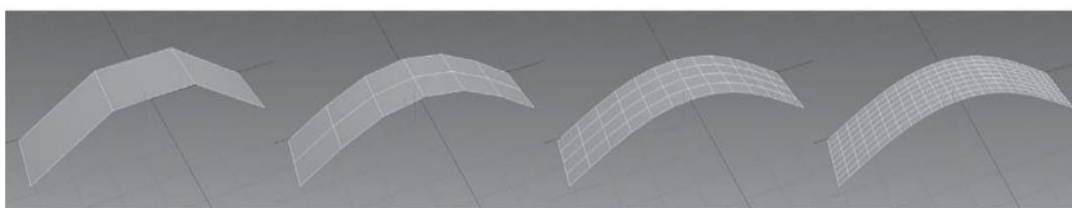
Esta técnica apresenta diversas dificuldades para a modelagem orgânica, e por isso foi sendo abandonada aos poucos na criação de personagens, sendo substituída pela modelagem poligonal, mas ainda é a técnica preferida para modelagem industrial de carros, motos, aviões, telefones, brinquedos e vários outros objetos (ANDALÓ, 2015, p. 20). Estes objetos necessitam de um detalhamento maior, curvas complexas são necessárias para representar o objeto do mundo real no ambiente 3D, para se chegar ao mesmo resultado com a modelagem poligonal a quantidade de polígonos teria que ser enorme, trazendo complexidade ao modelo e deixando-o mais pesado para o processamento gráfico, tornando a modelagem NURBS a melhor escolha.



### 2.5.2.2 Modelagem poligonal

Essa técnica de modelagem é a mais usada atualmente para diversos fins, é conhecida como aquela em que diversos triângulos conectados formam superfícies tridimensionais (ANDALÓ, 2015, p. 21). Surgiu no final dos anos 1950, em que havia baixo poder computacional. É um método de modelagem em que através da manipulação de polígonos é possível obter uma forma de objeto desejada.

Figura 05 - Modelo poligonal sendo subdividido várias vezes.



Fonte: Andaló (2015, p. 22)

Andaló (2015, p. 21) descreve que com a evolução dos computadores, com aumento do poder computacional, foi possível trabalhar com cada vez mais polígonos em um modelo, o que permite um nível de detalhamento maior, mas mesmo assim era complicado criar formas complexas com modelagem poligonal, que acabou perdendo espaço para a NURBS. Porém o autor conclui que com o advento da subdivisão de polígonos a modelagem poligonal voltou a ter destaque.

A modelagem gera um objeto estático, mas quando é preciso que o objeto se mova, como um personagem ou uma árvore mexendo as folhas é necessário a realização da animação. A animação complementa o modelo gerado, dando vida ao objeto.

### 2.5.3 Animação

Além da modelagem 3D um importante aspecto para criação de jogos é a animação digital. A animação digital permite a adição de movimentos nos jogos que junto dos modelos 3D ou 2D dão vida a personagens e ambientes que simulam o mundo real, se aproximando cada vez mais da realidade. A respeito da atividade de animação, segundo Andaló (2015, p. 123), "Seu princípio básico está na transformação ao longo do tempo, ou seja, ela pode ser tanto uma mudança de posição como uma rotação, uma cor, ou qualquer outra propriedade que sofra uma alteração em um determinado tempo".

Em jogos, geralmente uma imagem é apresentada e em seguida é substituída por uma nova imagem trazendo o efeito de movimento, uma sequência de imagens que ao longo de um tempo registra o que aconteceu durante um período, mas que é apresentada para nossa compressão como algo único, em forma de vídeo. Andaló (2015, p. 124) nos apresenta a técnica de animação por quadro-chave, que através da análise de movimento torna capaz a identificação de poses que na passagem de uma a outra mostram a variação do movimento, através da interpolação dessas poses o processo de criação de uma animação se dá início.

O processo de animação por quadro-chave é bastante utilizado para criação de animações para jogos, mas para grandes filmes ou produções em que há a necessidade de animações mais realísticas de personagens humanos são utilizadas várias outras técnicas como a utilização de roupas com sensores que transmitem dados para um software que os processa e cria as animações automaticamente.

#### **2.5.4 Texturização**

A texturização ou mapeamento de textura nasceu da complexidade de se criar mais detalhes na modelagem de polígonos, é realizada através da construção de uma imagem contendo os detalhes desejados e mapeando sobre a superfície do modelo (SOUZA, ARAÚJO e LEE, 2011). Essa técnica possibilita a inclusão de realismo à cenas e seus modelos digitais 2D ou 3D com baixo custo por ser mais fácil renderizar uma imagem do que milhares de polígonos que seriam necessários para se chegar no mesmo resultado.

O mapeamento UV é uma das formas mais utilizadas para mapeamento de texturas em modelos 3D, nela é necessário fazer marcações nos objetos 3D para determinar os pontos em que a malha poligonal deve ser aberta. A partir dessas marcações pode-se manusear as arestas, polígonos, vértices e faces a fim de planificar o objeto 3D gerando por fim um mapa 2D que pode ser exportado para ser utilizado em outros softwares.

Criar um jogo com características bem pensadas, conteúdo interessante aos olhos dos jogadores pode levar um tempo. Desenvolver um jogo com boa qualidade gráfica, modelos 2D ou 3D bem construídos, simular física e criar interações do jogador com o ambiente do jogo também pode ser demorado e exige grande conhecimento dos desenvolvedores. Um ponto importante que os desenvolvedores devem analisar no desenvolvimento de jogos é a utilização de ferramentas de auxílio, que possibilitam obter maior performance no processo e maior qualidade no produto final.

## 2.6 FERRAMENTAS DE AUXÍLIO

Desde o surgimento dos jogos digitais os desenvolvedores buscam criar jogos cada vez mais complexos, com qualidade gráfica superior e em prazos menores. Para isso diversas ferramentas que possuem funcionalidades para modelagem, animação, criação de cenários e codificação são criadas para auxiliarem os desenvolvedores na construção de jogos. Entre as ferramentas destacam-se as *game engines*, *softwares* para edição de imagens, modelagem e animação.

### 2.6.1 Criação Artística

O conteúdo gráfico de um jogo é muito importante para maior imersão do jogador, jogos podem chamar a atenção e atrair jogadores simplesmente pelo visual. Utilizar *softwares* que possibilitem a criação dos elementos gráficos do jogo pode prover performance em seu desenvolvimento e auxiliar na construção artística de um visual que agrada os jogadores.

#### 2.6.1.1 Blender 3D

Segundo Brito (2011, p. 15), o Blender é um *software* que possui um conjunto de ferramentas integradas que ajudam muito no desenvolvimento de um projeto de jogo, dentre elas destacam-se as funcionalidades para modelagem, renderização, animação e pós-produção, além de uma *game engine*. O *software* é desenvolvido pela *Blender Foundation* e distribuído sob a licença GPL (*General Public License*).

Segundo o manual da última versão do Blender, (Blender – Manual v.2.79<sup>a</sup>) o *software* oferece várias funcionalidades, dentre elas estão:

- **Modelagem:** Dispõe de uma ferramenta para modelagem tridimensional completa.
- **Animação:** O Blender 3D possui uma ferramenta capaz de criar animações para personagens e objetos em geral.
- **Simuladores e modificadores:** Conta com uma série de simuladores, que vão desde de simuladores de tecidos até simuladores de fluidos. Além disso possui modificadores que são capazes de criar operações automáticas facilitando o trabalho de artistas e desenvolvedores.
- **Game engine:** Internamente possui uma ferramenta para criação de *games* robusta, disponibilizando muitas funcionalidades.

### 2.6.1.2 Maya

Maya é um *software* de modelagem, animação, simulação e renderização 3D que fornece várias ferramentas integradas. É utilizado para criar personagens, ambientes tridimensionais e realidade virtual (AUTODESK, 2018). Ele se destaca na modelagem e animação de personagens 3D que são as atividades mais realizadas por seus usuários.

A empresa disponibiliza o software através de assinatura mensal, anual, no período de dois e três anos. Mas também é possível utilizá-lo gratuitamente por 30 dias na versão de avaliação. Para fins educacionais pode se solicitar uma licença educacional, mas atentando-se as restrições de uso e contrato de licença.

O Maya possibilita a construção de gráficos de alta qualidade, é um dos softwares mais utilizados por estúdios de cinema e desenvolvedores de jogos para criação de gráficos tridimensionais, possui extensão de funcionalidades através de sua linguagem de *scripting*, a *Maya Embedded Language*, entre outras e importa gráficos de vários outros aplicativos (STUDICA, 2018). Foi utilizada na criação de recursos gráficos em grandes filmes como Guardiões da Galáxia, Interestelar e Como treinar seu Dragão 2, que inclusive foram produções indicadas para o Oscar (ABDALA, 2015). Sem dúvidas é um *software* completo e que fornece ferramentas profissionais para se criar recursos gráficos de qualidade, devido ao preço de licença é mais indicado a projetos com orçamentos maiores.

### 2.6.1.3 Photoshop

O Adobe Photoshop é referência na geração de imagens digitais e fornece uma gama de ferramentas para edição e composição de imagens 2D. O seu nível de poder e controle sobre imagens permite alta eficiência na produção de imagens dos mais variados estilos (SMITH, 2012, p. 2). É uma excelente escolha para criação de ilustrações, tratamentos de imagem e criação de texturas. Este *software* permite através de ferramentas de edição o corte, colagem e clonagem das imagens possibilitando uma fácil manipulação.

O Photoshop oferece um conjunto de ferramentas intuitivas para trabalhar com fotografias e ilustrações, criar pôsteres, embalagens, banners, logotipos e sites, que até iniciantes podem fazer algo incrível (ADOBE, 2018). As possibilidades de criação artística são inúmeras, profissionais conseguem criar belíssimas obras de arte gráficas.

O *Software* pode ser adquirido via planos de assinatura mensal ou anual e também conta com uma versão gratuita de avaliação. A empresa fornece alguns materiais para aprendizado,

mas a disseminação de conhecimento pela comunidade de utilizadores se destaca como fonte de conteúdo.

## 2.6.2 Game Engine

A *game engine* é um motor de jogo, um software composto por bibliotecas criadas para facilitar o processo de construção de um jogo (VASCONCELOS E RIBEIRO, 2013, pag. 62). Possui várias funcionalidades utilizadas em conjunto ou separadas que auxiliam os desenvolvedores na criação de jogos.

Tudo que envolve interação no ambiente, desde iluminação, efeitos especiais e simulação de física do mundo real dentro do jogo é controlado pela *game engine*, (BRITO, 2011, p. 15). Ela renderiza gráficos 2D e 3D, possui motor de física para trabalhar com colisões e animações, suporte a sons, inteligência artificial, programação e gerenciamento de arquivos (DIAS, 2018). Um motor de jogo ainda realiza o gerenciamento de janelas do sistema operacional, proporciona o suporte a rede, monitora eventos de dispositivos de entrada e realiza a integração da codificação com o universo do jogo.

*Game engines* possibilitam o reaproveitamento de código e artefatos desenvolvidos em outros projetos. É uma das ferramentas mais importantes no desenvolvimento de jogos, sua utilização facilita o desenvolvimento, reduz o tempo total de trabalho e possibilita uma produção de jogos de qualidade, à medida que concentra a equipe na implementação da lógica e conteúdo do jogo.

Existem diversas *game engines* disponíveis no mercado, algumas são recomendadas para produção de *games* 2D ou para produções em 3D, algumas são *open source*, outras cobram uma porcentagem dos lucros e existem as que só cobram um valor específico. Dentre muitas *engines* existentes, destaca-se neste trabalho o Unity e a Unreal Engine, *engines* completas, que estão ganhando espaço entre os desenvolvedores de jogos no Brasil.

### 2.6.2.1 Unity 3D

A Unity contribui com o desenvolvimento de jogos possibilitando a redução de tempo, criação de conteúdo gráfico de melhor qualidade, mais interações entre o jogo e os jogadores, facilidade de distribuição e integração de tecnologias para o desenvolvimento. Ele tem sido o ambiente de desenvolvimento usado para a construção de vários jogos que tiveram sucesso

globalmente, incluindo jogos *online* e para diferentes plataformas (VASCONCELOS e RIBEIRO, 2013, p. 21).

O Unity é um motor de desenvolvimento integrado que fornece uma funcionalidade pioneira para criação de jogos e outros conteúdos interativos. Você poderá utilizar o Unity para montar sua arte e recursos em cenas e ambientes, adicionar Física, editar e testar simultaneamente seu jogo e, quando finalizado, publicar em suas plataformas escolhidas, tais como computadores, Web, iOS, Android, Wii, PS3 e Xbox 360 (UNITY, 2018).

Dias (2018) cita algumas características que contribuem para que a unity seja lembrada no momento da escolha de uma *game engine* e do porquê ser bastante utilizada em projetos de jogos:

- **Facilidade de aprendizado:** possui uma interface amigável e de fácil usabilidade, com muitas funcionalidades disponíveis. Os usuários da ferramenta têm a sua disposição uma vasta e organizada documentação, possui materiais de apoio em vídeo tutoriais criados pela própria empresa e por seus milhares de usuários e há discussões em fóruns que auxiliam na resolução de problemas. Todo esse conteúdo disponível facilita a jornada de aprendizado dos desenvolvedores iniciantes.
- **Multiplataforma:** o jogo gerado do mesmo projeto de desenvolvimento pode ser publicado em diversas plataformas, não necessitando criar projetos diferentes para plataformas específicas, trazendo facilidades, reduzindo custos e tempo e aumentando o público dos jogos.
- **Quantidade de materiais prontos para uso:** ao longo do tempo a comunidade que utiliza o unity desenvolveu um vasto conteúdo de elementos de jogos prontos para serem utilizados, envolvem modelos 3D, animações, cenários, efeitos especiais, sons e extensões da engine, que podem ser encontrados na própria loja do unity e em sites especializados.

O Unity pode ser obtido em sua versão *Personal* gratuitamente quando a receita não exceder cem mil dólares por ano, e em suas versões *Plus* e *Pro* através de assinaturas que fornecem vários benefícios não disponíveis na *Personal*.

#### 2.6.2.2 Unreal Engine

A Unreal Engine criada pela Epic Games é uma ferramenta bastante presente no mercado de desenvolvimento de jogos. A empresa cobra *royalties* sobre o uso comercial da ferramenta, 5% da receita bruta obtida com a *engine* deve ser repassado à empresa, mas também está disponível gratuitamente para uso não comercial.

Segundo Mallmann (2012, p.30) a Unreal Engine possui um *framework* de desenvolvimento que oferece uma série de tecnologias que vão desde de ferramentas de criação de conteúdo até suporte de programação. Esse motor foi escrito em C++ e foi utilizada no desenvolvimento de muitos *games* de sucesso como Fortnite e Unreal Tournament.

Essa ferramenta foi criada com o intuito de proporcionar o máximo de facilidade na criação de jogos, tendo como objetivo colocar grande parte do poder de decisão nas mãos dos desenvolvedores para desenvolver projetos de jogos com o mínimo de programação (UNREAL ENGINE, 2018).

### 3 DESENVOLVIMENTO

Esta parte do trabalho está dividida em quatro seções que descrevem todo o processo realizado no desenvolvimento do projeto. Na primeira, é discorrido sobre a metodologia de desenvolvimento adotada, na segunda encontra-se o relato de como foi a concepção do jogo, a terceira aborda partes do planejamento e a quarta finaliza descrevendo como o jogo foi construído.

Para o desenvolvimento do jogo, foram necessárias a busca e utilização de grande volume de conhecimento, adquirido de diversas fontes, desde experiências anteriores dos desenvolvedores, pesquisas bibliográficas, cursos especializados e grupos de discussões de profissionais, seguindo e adaptando conceitos compreendidos. Alguns títulos de jogos digitais de RPG como *Perfect World*, *World of Warcraft* e *Dota 2*, foram utilizados como base para compreensão da temática do gênero RPG, possibilitando a obtenção do conhecimento sobre suas mecânicas e interações possíveis aos jogadores.

Desenvolver um jogo com as melhores práticas da engenharia de software é um ideal do projeto. Com esta ideia, uma metodologia ágil de desenvolvimento é definida para ser utilizada e auxiliar os autores na condução de todo o processo de desenvolvimento do jogo.

#### 3.1 METODOLOGIA DE DESENVOLVIMENTO

Como metodologia de desenvolvimento foi selecionada a metodologia ágil *Extreme Game Development* que se baseia na *Extreme Development* muito utilizada na construção de softwares tradicionais. Ela pode auxiliar os desenvolvedores a produzirem um jogo de qualidade em tempo reduzido e se adequa ao projeto pois a equipe de desenvolvimento é pequena, possuindo um prazo muito curto para a construção do jogo e como metodologia ágil, incentiva que as entregas de *features* sejam pequenas, mas frequentes entre as iterações.

Como parte da aplicação da metodologia, histórias de usuários e seus cenários (Apêndice B), foram criados para serem utilizados na análise de requisitos e também em testes, técnicas que se adequam muito bem na criação de funcionalidades para jogos, pois é possível simular fluxos que os jogadores poderiam seguir dentro do jogo durante as interações, possibilitando criações de interações funcionais bem planejadas.

Programação em pares, uma técnica muito utilizada nessa metodologia, está presente na implementação do jogo, não somente na etapa de codificação, mas também em outras atividades



como modelagem e animação, propiciando um trabalho colaborativo, almejando uma produção de qualidade e evitando a geração de bugs.

Após a definição da metodologia de desenvolvimento é iniciada a fase de concepção do projeto do jogo, onde reuniões são realizadas, ideias começam a surgir e uma visão do jogo começa a ser formada.

## 3.2 CONCEPÇÃO

Na fase de concepção houve a captação de ideias através de um *brainstorming*, em que características e funcionalidades que o jogo poderia possuir foram levantadas e documentadas, assim como refinadas no decorrer do projeto.

O estilo gráfico foi um dos primeiros assuntos abordados, sendo estabelecido o uso de gráficos tridimensionais que proporcionaria maior imersão ao jogador, e modelos com características *low poly*<sup>1</sup> e estilização *hand painted*<sup>2</sup>, conjunto de elementos que além de boa aparência visual, não necessita de um alto consumo de hardware para processamento gráfico durante a construção e execução do jogo.

Com o estilo do jogo e suas características em mente deu-se início ao pensamento sobre o contexto em que o jogo estaria inserido. Um jogo precisa possuir um contexto, uma história para embasar todo o seu ambiente. Uma história foi concebida, nasceu então o mundo de Mordrek, acontecimentos foram descritos e uma abertura para conectar o ambiente do jogo com a continuação da história foi criada.

Todas as características, funcionalidades e demais informações relacionadas ao jogo foram organizadas e inseridas no documento de *game design* (Apêndice A), documento que sofre frequentes atualizações e é um guia para os desenvolvedores durante toda a construção do jogo. Foi criado de acordo com vários conceitos levantados por Rogers (2014, p. 64), tendo sua estrutura adaptada para o projeto. O *game design* aborda a história, o mundo do jogo, *gameplay*, sistemas de jogo, características dos personagens e mecânicas em geral.

De acordo com Rogers (2014, p. 68), o GDD deve conter elementos como o título do jogo, os sistemas de jogos pretendidos, detalhes sobre o público alvo, um resumo da história do jogo, mecânica e modos de jogo, pontos de venda exclusivos e produtos da concorrência. Neste projeto são utilizados alguns destes tópicos como título do jogo, sistemas de jogos, detalhes

---

<sup>1</sup> Em modelos 3D representa uma malha poligonal com quantidade de polígonos reduzida.

<sup>2</sup> Estilo de texturização em que os detalhes do modelo 3D são criados na textura e não na malha poligonal.

sobre o público, história ou contexto em que se passa o jogo e detalhes sobre o gameplay, mas outros como pontos de venda e assuntos sobre concorrência não são abordados pois não são de grande relevância para este projeto.

Com várias ideias sobre o jogo definidas e a construção do documento de *game design* em progresso, é iniciada então a fase de preparação para a implementação do jogo que utilizará todos os dados levantados na concepção como base para tomadas de decisões.

### 3.3 PREPARAÇÃO

A fase de concepção foi utilizada para definir elementos que abrem a mente da equipe para as características do jogo, cria uma direção a ser focada e possibilita o prosseguimento do projeto. Na fase de preparação, as ideias continuaram a ser refinadas, arquitetura do software é criada e formas para obtenção de melhor performance no processo de desenvolvimento e qualidade do produto são definidas por meio de ferramentas.

A arquitetura foi criada com o intuito de ser facilmente entendida e gerenciável, provendo facilidade em manutenções de código, reduzindo o acoplamento entre componentes e possibilitando a criação e integração de novos sistemas de jogo.

A arquitetura do jogo é composta por elementos chamados neste trabalho de sistemas de jogo, em que cada sistema é responsável por uma parte do jogo, possui uma coleção de funcionalidades e podem trabalhar em conjunto com outros sistemas. Há sistemas externos disponibilizando interações ao jogador e também internos relacionados à forma de execução do jogo. Entre os sistemas internos estão o *UI Manager*, *Scene Manager* e *Game Manager*, e nos externos estão o sistema de missões, estatísticas, inventário e combate. A comunicação entre os sistemas pode ser observada na figura 06.

Figura 06 - Arquitetura de Sistemas de Jogo do Projeto.



Fonte: Os autores.

O *Game Manager* gerencia todo o fluxo do jogo, controla o carregamento de cenas, gerencia pause, salvamento e carregamento do estado do jogo, inicialização e finalização do jogo. É criado para garantir que todos os sistemas de jogo trabalhem juntos e em ordem, usando o estado do jogo para rastrear e controlar o comportamento do jogo, pode ditar comandos, impor regras, e disponibilizar dados que muitas partes de um jogo podem precisar de acesso (MCELROY, 2018). Incorporado a ele está o *Scene Manager* que recebe instruções para realizar manipulações de cenas, como carregamento e remoção, e ainda disponibiliza eventos de feedback de suas operações, é um sistema interno do motor gráfico mas é utilizado diretamente pelo *Game Manager*.

Além dos cenários e personagens, os jogos também possuem interfaces gráficas contendo menus e o HUD (*Heads-Up Display*), informações sobre elementos do jogo durante as interações, neste projeto estas interfaces fazem parte do sistema *UI Manager*. Ele é um sistema planejado para gerenciar os diversos elementos gráficos presentes na interface do jogo, como exibição do menu de opções, barras de saúde e energia, coordenadas do personagem no mundo do jogo, telas de carregamento, efeitos de transição entre cenas e exibição de inventário do personagem.

Como o jogo é um RPG, um sistema de estatísticas de personagem foi planejado, jogador e NPCs, personagens não jogáveis controlados pelo computador, do mundo do jogo possuem características como saúde, *level*, experiência, energia, força de ataque e defesa. Através deste sistema essas estatísticas podem ser gerenciadas e utilizadas para interatividade com vários sistemas presentes no jogo, como o sistema de inventário que contém itens que o jogador pode equipar e utilizar e o sistema de combate que usará e interferirá diretamente nelas.

O sistema de combate é planejado integrado com o sistema de estatísticas e de missões, tornando possível a interação de luta entre o personagem do jogador e os NPCs e os NPCs entre si. Este sistema proporciona mais ação ao jogo, é uma das principais interações em jogos de RPG, o jogador pode usar estratégias nas batalhas, utilizando equipamentos e habilidades desejados, adequados ou não para a batalha, seja contra um ou vários oponentes.

A construção do sistema de missões possibilita a apresentação de desafios e tarefas para o jogador, os NPCs interagem com o jogador possibilitando que vivencie aventuras e obtenha recompensas ao concluir as missões. Este sistema se comunica com o sistema de inventário ao necessitar obter informações sobre itens armazenados e disponibilizar itens como recompensa, com o sistema de combate para acesso a dados de lutas do personagem e com o sistema de estatísticas do personagem para disponibilizar experiência ao completar missões.

Para obter melhor performance no desenvolvimento, ganhar agilidade e possibilitar os autores a construir um jogo de qualidade, ferramentas foram selecionadas para auxiliar no desenvolvimento, algumas por já serem conhecidas pelos autores, mas todas por proverem funcionalidades que são necessárias na construção do jogo.

O Blender foi selecionado como ferramenta para modelagem e animação, proporcionando à equipe de desenvolvimento a criação de modelos 3D de maneira profissional e sem custo por ser uma ferramenta livre.

Esse software é amplamente utilizado no mundo todo e oferece uma grande gama de funcionalidades úteis para o desenvolvimento de jogos. Foi uma das ferramentas que forneceram mais desafios aos autores por não possuírem muita experiência com ela. Com ela foi possível a modelagem 3D de vários objetos do jogo, assim como a animação de alguns deles e as texturas criadas com o Photoshop puderam ser mapeadas aos modelos.

A parte artística de um jogo é um ponto extremamente importante para agradar e envolver os jogadores. Para se criar um ambiente gráfico de um jogo digital 3D é muito importante utilizar um software de manipulação de imagens que possa auxiliar na criação provendo funcionalidades que tragam velocidade e qualidade.

Neste projeto, o Photoshop foi o *software* utilizado para criação de ilustrações, texturas e no tratamento de imagens. Auxiliou os autores desde a criação de rascunhos de elementos gráficos até a arte final. As artes gráficas produzidas pela ferramenta ganham destinação na *game engine*, onde são acopladas a modelos, aplicadas ao terreno ou inseridas em interfaces gráficas de usuário.

O Unity *game engine* foi uma ferramenta muito importante no desenvolvimento, proporciona aos desenvolvedores diversas funcionalidades que os auxiliam na construção do

*game*. Através dele foi possível montar os cenários simplesmente arrastando e soltando modelos no editor, podendo ainda posicioná-los, rotacioná-los e manipular seu tamanho. Fornece ótimo sistema de construção de interface gráfica, em que torna possível a criação de menus e interface gráfica com apenas alguns cliques. Possibilitou criar efeitos especiais com seu sistema de partículas, configurar animações e texturas de modelos.

Todo o jogo foi criado dentro do Unity, modelos 3D, imagens, texturas e animações foram importados e integrados através de suas ferramentas. Ao decorrer da construção do jogo, foi necessário recorrer à documentação e tutoriais oficiais do Unity que foram de grande ajuda para tirar dúvidas e aprender como trabalhar com determinada funcionalidade. Durante os momentos de dúvidas percebeu-se que a comunidade de utilizadores da ferramenta é muito grande e também produz muito conteúdo para ajudar os novatos. Essas percepções sobre a ferramenta levam a conclusão de que é uma boa escolha para desenvolvedores de jogos iniciantes.

O Unity é a ferramenta que usará todos elementos criados pelas outras ferramentas, e este momento se dará na fase de implementação.

### 3.4 IMPLEMENTAÇÃO

Nesta seção são descritas quais e como foram executadas as atividades presentes no processo de desenvolvimento que possibilitou atingir o objetivo do trabalho, a criação do jogo. A maioria das atividades quando disponível foram desempenhadas paralelamente, somente algumas foram executadas de forma sequencial pois dependia do produto de outra.

Após formar ideias, levantar funcionalidades, definir ferramentas de auxílio e uma metodologia a ser utilizada e planejar uma arquitetura de software, deu-se início a fase de implementação, momento em que o jogo começa a ser formado.

Uma das primeiras atividades na fase de implementação foi a modelagem 3D dos elementos do jogo. Mas devido ao curto prazo para construção do jogo, aquisições de modelos 3D também foram necessárias.

#### 3.4.1 Modelagem 3D

Modelos 3D que fazem parte do cenário e alguns personagens foram adquiridos diretamente da *assets store* do unity, uma loja virtual do *software*, para possibilitar um ganho de agilidade no processo de desenvolvimento. Realizar a compra de alguns componentes do

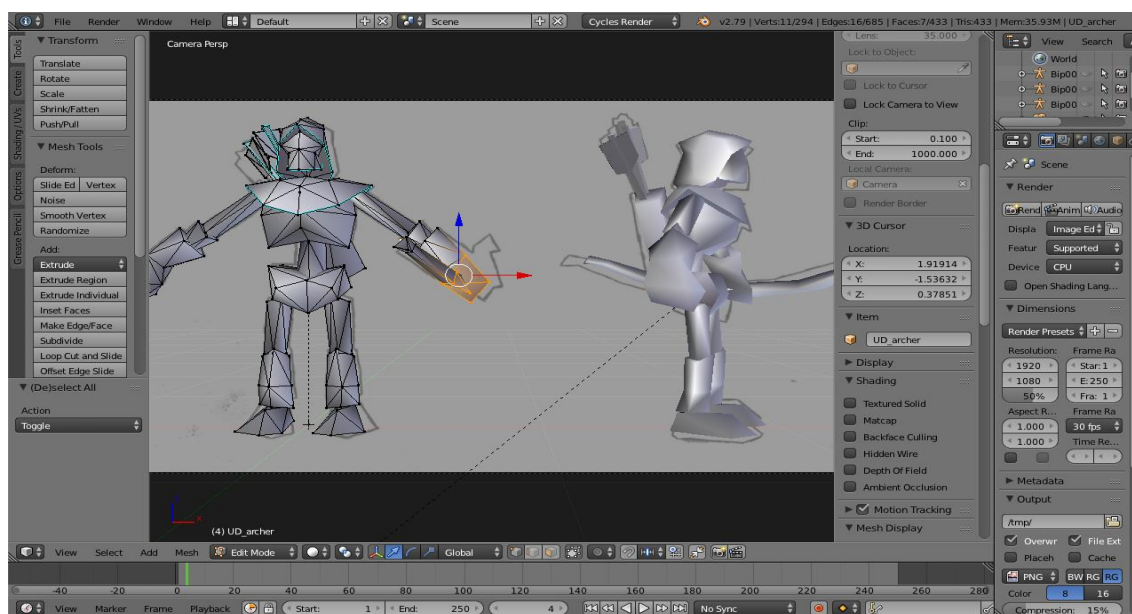
jogo pode ser uma boa alternativa à criação do zero, ainda mais quando o prazo de desenvolvimento é curto ou os criadores não possuem tanta perícia com a atividade exigida, mas em contrapartida aumenta o custo do projeto.

Foram adquiridos modelos que se adequam ao estilo gráfico e temática do jogo, alguns foram utilizados exatamente como foram criados e outros foram modificados para adaptação ao jogo através da ferramenta de modelagem e animação.

A modelagem de objetos 3D realizada pelos autores foi feita através da ferramenta Blender utilizando a técnica de modelagem poligonal. Com funcionalidades da ferramenta de modelagem foi possível construir os modelos através da ligação de vértices (pontos no espaço tridimensional) formando arestas ou segmentos (linhas), que estando três ou mais conectadas formam uma face (polígono). Com vários polígonos conectados é formado a malha do objeto ou superfície, surgindo então o modelo 3D bruto.

Para maior facilidade na modelagem do objeto desejado foram utilizadas referências externas, desenhos do objeto em diferentes ângulos de visualização, que possibilitam a criação dos polígonos em cima da imagem de referência. Para melhor entendimento pode-se fazer uma analogia com o processo de usar uma folha transparente de desenho em cima de uma imagem para fazer uma cópia. Este processo de modelagem pode ser observado através da figura 07.

Figura 07 - Modelagem com o auxílio de imagem como referência.



Fonte: Os autores.

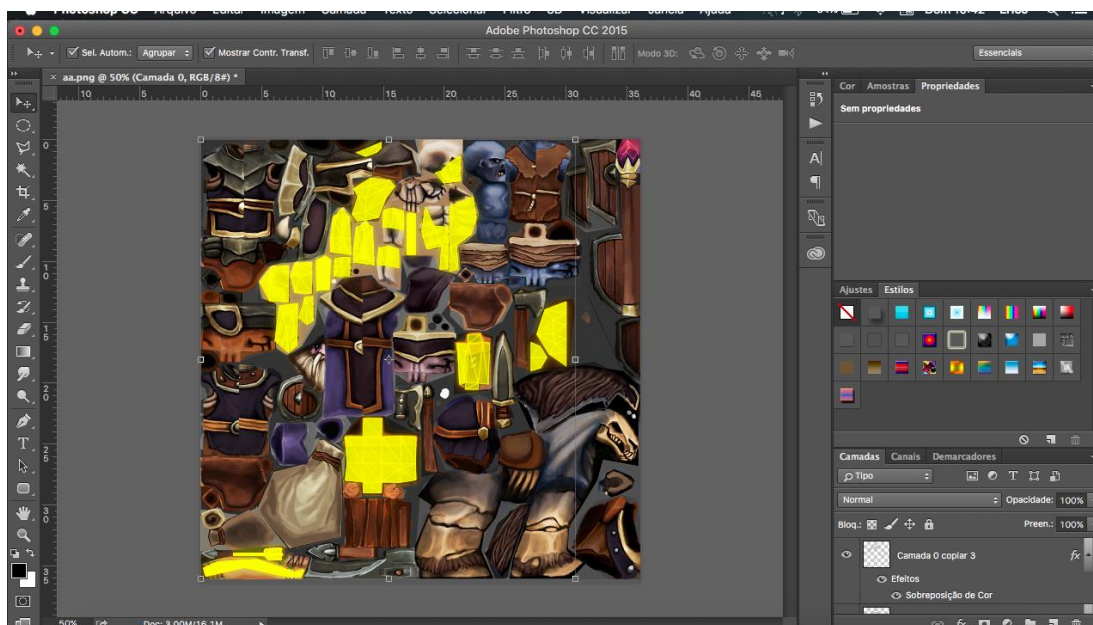
Devido ao estilo gráfico do jogo, os modelos criados pelos autores e os adquiridos foram construídos com a quantidade de polígonos reduzida, deixando os detalhes para serem

implementados por meio da textura do modelo utilizando o estilo de texturização *Hand Painted*, estratégia que se adequa muito bem ao projeto devido ao tempo e orçamento para o desenvolvimento, não necessitando de ambientes gráficos e personagens muito realistas com vários detalhes, e também ao estilo de jogo do projeto que contempla elementos medievais e de fantasia.

### 3.4.2 Texturização

As texturas para os objetos modelados pelos autores foram criadas no Photoshop, gerando uma imagem contendo a textura de todas as partes do modelo conforme pode ser observado na figura 08. Variações de cores das texturas também foram utilizadas em vários modelos para proporcionar a percepção de diferença entre personagens que possuem o mesmo modelo 3D.

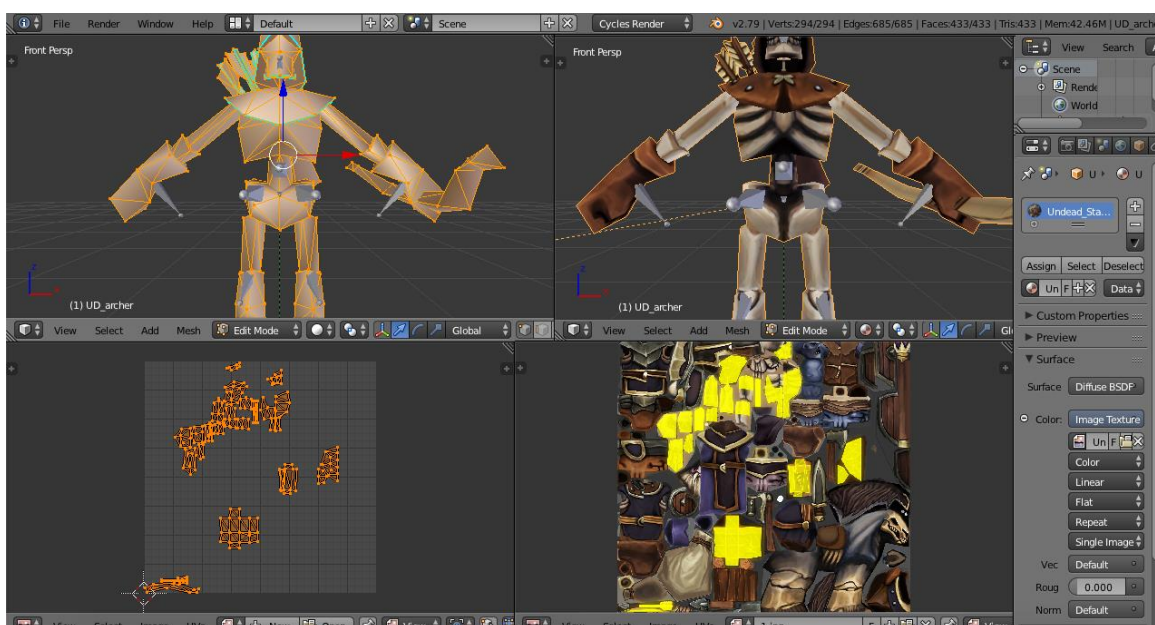
Figura 08 - Edição da textura de um personagem com o Photoshop.



Fonte: Os autores.

A imagem com a textura é importada para o Blender onde se inicia o processo de mapeamento do modelo 3D em relação à textura criada através de mapa UV conforme pode-se visualizar na figura 09. Este processo consistiu em realizar marcações de partes do modelo 3D ligando a partes da textura no editor UV do Blender, formando então um mapa que pode ser importado por softwares de renderização juntamente com o modelo tornando possível a visualização do modelo com a textura aplicada a ele.

Figura 09 - Processo de mapeamento UV dentro do Blender.



Fonte: Os autores.

As texturas utilizadas em objetos que necessitaram de repetições, como texturas de solo e paredes foram editadas seguindo uma proporção de múltiplos de dois. Para isso o Photoshop foi utilizado para criar texturas de tamanhos específicos para o modelo a ser aplicada permitindo sua replicação de forma que a apresentação da textura aplicada seja contínua, reduzindo a percepção de quadrados se repetindo com o mesmo padrão na superfície do modelo.

### 3.4.3 Animação

A animação assim como a texturização é uma atividade que precede a modelagem, com os modelos construídos passa-se então a criar movimento a eles.

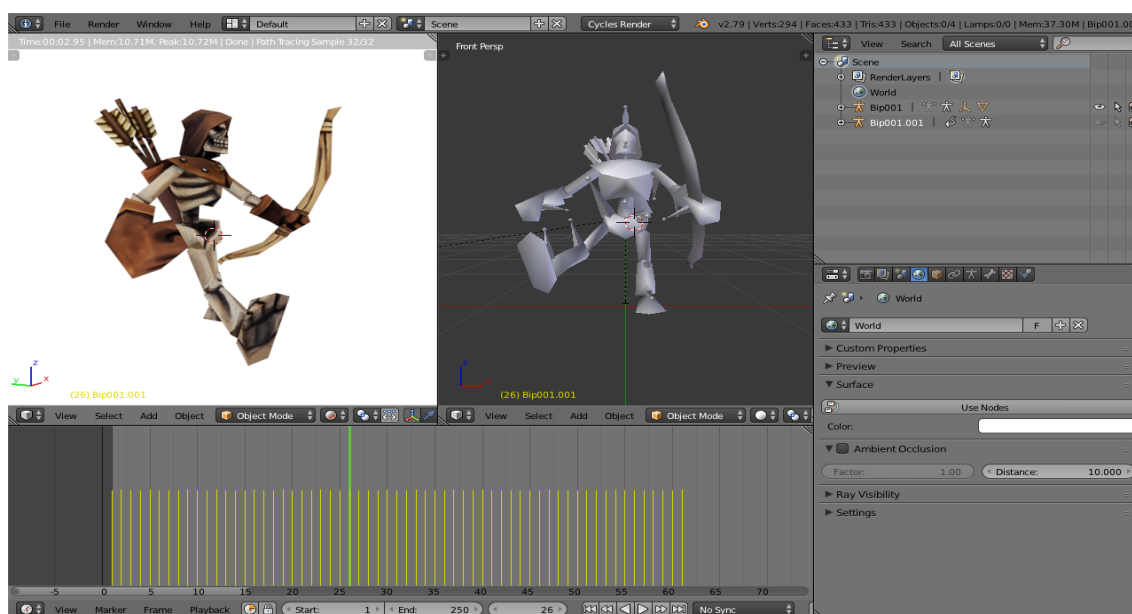
As animações dos modelos 3D foram criadas com o Blender por meio de sua funcionalidade de animação com quadros-chaves. Cada quadro-chave registra uma pose do modelo e o Blender se encarrega de produzir os quadros intermediários de acordo com a posição, escala e rotação do modelo gerando a animação desejada.

As poses dos modelos foram criadas com auxílio de elementos do Blender chamados de *Bones*, que formam um esqueleto que ao ser aplicado a um modelo permite deformar sua malha facilitando a criação das poses. Este processo de animação pode ser observado pela figura 10.

A animação foi a última atividade relacionada à criação do modelo 3D, com o modelo finalizado é possível utilizá-lo na construção do mundo do jogo.



Figura 10 - Processo de animação utilizando *bones* e quadros-chaves com o Blender.



Fonte: Os autores.

### 3.4.4 Sonorização

Para maior imersão do jogador, músicas e efeitos sonoros são essenciais no ambiente do jogo. A sonorização foi implementada de modo que fosse adequada ao ambiente e ao momento em que o jogador está presente. Ao caminhar por uma cidade é possível ouvir pessoas conversando, em uma floresta sons de animais e perto de um rio ouve-se o barulho da água correndo.

A maioria das músicas e os efeitos sonoros foram obtidos em sites especializados de maneira gratuita, mas alguns efeitos sonoros como conversas de NPCs foram gravados pelos autores utilizando suas próprias vozes e de voluntários.

A reunião dos elementos sonoros do jogo foi realizada em paralelo com as atividades de modelagem e codificação, esta que é responsável por programar quando e onde o som será tocado no jogo.

### 3.4.5 Codificação

A codificação foi desempenhada em paralelo com as atividades artísticas. A linguagem utilizada foi o C# que foi integrada muito bem ao Unity possibilitando a criação de qualquer tipo de jogo. A codificação é realizada utilizando o paradigma de orientação a objetos. As funcionalidades são criadas em arquivos fontes chamados no Unity de scripts, cada um

contendo uma funcionalidade específica que para estarem disponíveis na execução do jogo devem ser acoplados a componentes presentes na cena.

Os primeiros códigos criados foram os dos sistemas de jogo internos, que cuidam da execução do jogo como carregamento de cenas, pause, salvamento e finalização do jogo. Logo depois deu-se início a codificação dos sistemas de jogo externos, as funcionalidades disponíveis para interação do jogador com o mundo do jogo como movimentação do personagem, sistema de combate e missões.

### **3.4.6 Construção do mundo do jogo**

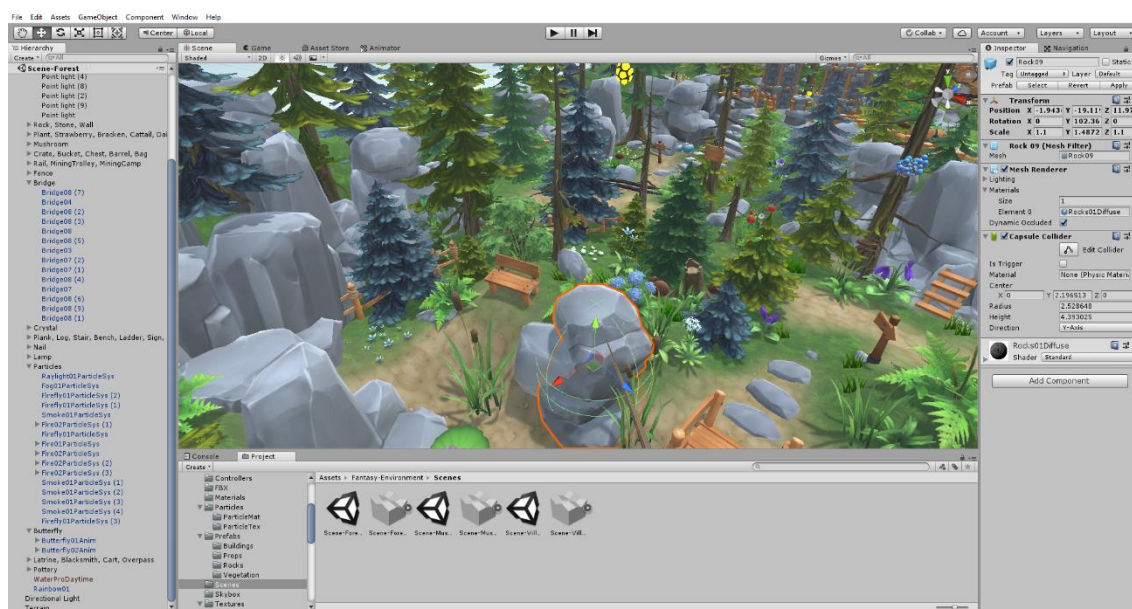
A construção do mundo do jogo ocorre a medida em que os elementos que compõem o jogo ficam disponíveis para utilização. Nesta etapa é o momento de integrar tudo o que foi produzido, modelos 3D, códigos, texturas, animações e sons para construir o mundo do jogo em que o jogador poderá interagir.

Tudo o que foi produzido faz parte do cenário do jogo ou age de determinada maneira sobre ele. A montagem do cenário foi uma das atividades que exigiram mais criatividade aos autores, muito tempo foi gasto a fim de que o conjunto de objetos posicionados e configurados no cenário retratassem a temática de RPG de fantasia que o jogo propõe.

O jogo possui apenas um grande mapa, não possui fases como no jogo *super mário world*, todo o desenrolar do *gameplay* acontece nele, é um mundo aberto disponível para ser desbravado pelo jogador, a paisagem muda ao longo do cenário assim como é no mundo real. Por ser um jogo de mundo aberto, tudo presente no jogo estar na mesma cena do unity, a construção do jogo tende a necessitar de um maior uso de *hardware* para processamento e dependendo de como for construído essa necessidade também se estende para a execução do jogo.

O cenário foi montado através da importação dos modelos tridimensionais, texturas, animações e demais componentes para dentro do editor de cenas do Unity, onde o posicionamento, rotação e tamanho dos modelos foram configurados e texturas e animações aplicadas, conforme pode ser visto na figura 11. O unity também fornece ferramentas para a criação de terrenos, em que foi possível criar facilmente planícies, vales, montanhas e rios e aplicar texturas de solo. Com o sistema de partículas foi possível criar efeitos especiais como fumaça, fogo e habilidades de personagens simplesmente através de configurações visuais, não necessitando criar nenhuma linha de código.

Figura 11 - Montagem de cenário com o Unity.



Fonte: Os autores.

Assim como é possível manipular modelos 3D dentro do editor de cena, também é possível adicionar luzes à cena e manipulá-las do mesmo modo. A iluminação é um aspecto de importância em jogos, ela pode fazer toda a diferença se criada corretamente, pode tornar um jogo mais bonito ou mais triste dependendo da escolha de cores, intensidade e tipos de luzes.

O Unity fornece vários tipos de luzes e vários parâmetros para configurá-las, assim como as sombras geradas por elas. No jogo foram utilizadas variações de formas de iluminação, foram criadas luzes que iluminam uma grande área, outras um objeto específico, algumas com cores diferenciadas que fazem parte dos efeitos especiais da cena, luzes estáticas que não sofrem modificações ao longo do jogo e luzes dinâmicas que tem suas propriedades alteradas ao receber uma influência externa.

Um cenário envolvente é muito importante, mas um jogo que possui um mundo sem vida, sem habitantes não é interessante ao jogador, por isso NPCs foram configurados e posicionados em várias localidades do mapa, fornecendo interatividade ao jogador, seja de combate ou informação através de sua inteligência artificial.

Os NPCs são controlados pelo próprio jogo, recebem um comportamento de acordo com resultados de cálculos e tomada de decisões sobre eventos que ocorrem no ambiente do jogo, permitindo-os determinar distâncias, definir coordenadas de destino para moverem-se, desviar de obstáculos e saber o momento certo de executar determinadas ações. Essa inteligência é fornecida através de scripts codificados em C# que são aplicados aos modelos 3D dos NPCs, permitindo comandar os NPCs de acordo com o ambiente ao seu redor.



#### 4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o processo de desenvolvimento de um jogo digital abordando suas atividades e conteúdo teórico necessário para seu entendimento. O jogo e artefatos necessários ao projeto foram criados, muito conhecimento foi adquirido e desafios foram enfrentados pelos autores.

O jogo construído até o momento possui conteúdo e as funcionalidades mais típicas de um jogo RPG, mas pode continuar a ser desenvolvido, adicionando-se mais funcionalidades e interações para o jogador e melhorando os detalhes do ambiente.

Os testes do jogo foram realizados pelos desenvolvedores e voluntários durante e após o desenvolvimento de cada funcionalidade, levando em consideração as informações presentes no GDD e no documento de histórias de usuários.

Através deste trabalho conclui-se que com a tecnologia e ferramentas disponíveis tornou-se possível a criação de jogos de qualidade por uma pequena equipe e até por desenvolvedores independentes em um curto período de tempo. Mas como visto, o desenvolvimento de jogos abrange várias atividades, algumas exigindo muita criatividade e técnica de seus desempenhadores, o que traz a necessidade de profissionais especializados para a criação de grandes jogos.

O mercado de jogos e a área de desenvolvimento tende a continuar crescendo, criando muitas oportunidades para novos desenvolvedores mostrarem seu potencial, colaborando em grandes empresas, trabalhando como desenvolvedor independente ou abrindo *startups* de criação de jogos.

Através deste trabalho, juntamente com o jogo e os artefatos gerados, espera-se contribuir com a comunidade de desenvolvedores de softwares compartilhando o conhecimento obtido pelos autores.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABDALA, Erico. Oscar 2015: **O Maya e a tecnologia por trás dos grandes filmes**. Disponível em: <<https://www.showmetech.com.br/oscar-2015-o-maya-e-tecnologia-por-tras-dos-grandes-filmes/>>. Acesso em: 02 jul. 2018.
- ADAMS, Ernest. **Fundamentals of game design**. 3. ed. New Jersey: New Riders, 2014.
- ADOBE. **Photoshop**. Disponível em: <<https://www.adobe.com/br/products/photoshop.html>>. Acesso em: 10 jul. 2018.
- ALBINO, Raphael Donaire; SOUZA, Cesar Alexandre; PRADO, Edmir Parada Vasques. **Benefícios alcançados por meio de um modelo de gestão Ágil de projetos em uma empresa de jogos eletrônicos**. Revista de Gestão e Projetos - GeP, v.5, 2014.
- ALLAN, Patrick. **The Surprising Benefits of Role-Playing Games (and How to Get Started)**. Disponível em: <<https://lifehacker.com/the-surprising-benefits-of-role-playing-games-and-how-1684582789>>. Acesso em 13 fev. 2018.
- ANDALÓ, Flávio. **Modelagem e animação 2D e 3D para jogos**. São Paulo: Editora Érica Ltda, 2015.
- ARRUDA, Eucídio Pimenta. **Fundamentos para o desenvolvimento de jogos digitais**. Porto Alegre: Editora Bookman, 2014.
- BENITTI, F. B. V.; SEARA, E. F. R.; SCHLINDWEIN, L. M. **Processo de Desenvolvimento de Software Educacional: proposta e experimentação**. Novas Tecnologias na Educação, V. 3 n. 1, 2005.
- BLIZZARD. WORLD OF WARCRAFT – BRASIL. Disponível em: <http://us.blizzard.com/pt-br/company/events/wow-brazil.html>. Acesso em: 13 fev. 2018.
- BRITO, Alan. **Blender 3D: jogos e animações interativas**. São Paulo: Novatec, 2011.
- CARVALHO, Eusyar Alves. **Os 5 Desprezíveis: Desenvolvimento de um Jogo Eletrônico Utilizando os Princípios de Engenharia de Software**. Universidade de Brasília - UnB, 2013.
- CLINTON, Keith. **Agile Game Development with Scrum**. United States of America: Addison Wesley Professional, 2010.
- CLUA, Eteban Walter Gonzales; BITTENCOURT, João Ricardo. **Desenvolvimento de Jogos 3D: Concepção, Design e Programação**. São Leopoldo: UNISINOS, 2005.
- DEMACHY, Thomas. **Extreme Game Development: Right on Time, Every Time**. Disponível em: <[https://www.gamasutra.com/view/feature/131236/extreme\\_game\\_development\\_right\\_on\\_.php](https://www.gamasutra.com/view/feature/131236/extreme_game_development_right_on_.php)>. Acesso em: 15 mai. 2018.

DIAS, Raphael. **Unity – Guia Completo sobre a Game Engine**. Disponível em: <<https://producaodejogos.com/unity/>>. Acesso em: 4 abr. 2018.

DIAS, Raphael. **Game Engine: o que é, para que serve e como escolher a sua**. Disponível em: <<https://producaodejogos.com/game-engine/>>. Acesso em: 4 abr. 2018.

ESPER, Marcio. **Uma pequena introdução ao desenvolvimento de jogos de computador**. Disponível em: <<http://www.hardware.com.br/guias/uma-pequena-introducao-desenvolvimento-jogos-computador/>>. Acesso em: 13 fev. 2018.

GEDIGAMES - Grupo de Estudos e Desenvolvimento da Indústria de Games, USP; NPGT - Núcleo de Política e Gestão Tecnológica, USP. **Mapeamento da Indústria Brasileira de Jogos Digitais**. São Paulo (SP); 2014.

MALLMANN, Eduardo Rafael. **Estudo e Desenvolvimento de um Jogo Utilizando Unreal Development Kit**. Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2012

MASTROCOLA, Vicente Martin **Ludificador**: um guia de referências para o game designer brasileiro. São Paulo: Independente, 2012.

MCELROY, Jon. **Swords and Shovels: Game Managers, Loaders, and the Game Loop**. Disponível em: <<https://www.pluralsight.com/courses/swords-shovels-game-managers-loaders-game-loop>>. Acesso em 01 jul. 2018.

MCGONIGAL, Jane. **The reality is broken**. London: Editora The Penguin Press, 2011

MITTEROFHE, Luiz Felipe de Lima. **Análise Comparativa entre Metodologias de Desenvolvimento de Games**. Universidade Federal de Juiz de Fora, 2016.

PAVÃO, Andrea. **A AVENTURA DA LEITURA E DA ESCRITA ENTRE MESTRES DE Roleplaying Games (RPG)**. 23ª Reunião Anual da Anped. Disponível em: <<http://23reuniao.anped.org.br/textos/1003t.PDF>> Acesso em: 01 jul. 2018.

PEREIRA, Caroline. **Modelagem e animação de personagens humanóides 3D**. Monografia de conclusão de curso apresentada à Universidade do Planalto Catarinense Departamento de Ciências Exatas e Tecnológicas. Lages, 2008.

PERUCIA, Alexandre Souza; BERTHÊM, Antônio Córdova de; BERTSCHINGER, Guilherme Lage; MENEZES, Roberto Ribeiro Castro. **Desenvolvimento de Jogos Eletrônicos Teoria e Prática**. São Paulo: Novatec Editora Ltda, 2007.

PRESSMAN, S. ROGER. **Engenharia de Software**: Uma abordagem profissional. Porto Alegre. 7. ed. 2011.

RAMSEY, Robert. **Guide**: The Best PS4 Open World Games. Disponível em: <[http://www.pushsquare.com/news/2017/11/guide\\_the\\_best\\_ps4\\_open\\_world\\_games](http://www.pushsquare.com/news/2017/11/guide_the_best_ps4_open_world_games)>. Acesso em: 13 fev. 2018.

ROGERS, Scott. **Level Up! the Guide to Great Video Game Design**. 2ª ed. United Kingdom : John Wiley & Sons Ltd, 2014.

ROSENBERG, Doug; STEPHENS, Matt. **Extreme Programming Refactored: The Case Against XP**. Berkeley, CA, Apress, 2003.

SCHELL, J. **The Art of Game Design: a Book of Lenses**. USA: Elsevier, 2015.

SCHUYTEMA, Paul. **Design de Games uma abordagem prática**. São Paulo: Cengage Learning, 2008.

SMITH, Jennifer. **Adobe Photoshop CS6 Digital Classroom**. Indianapolis: John Wiley & Sons, 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª ed. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, Carla Florentino; ARAÚJO, Edson Leite; LEE, Jennifer Chuin. **Texturização**. Disponível em <<http://www.dca.fee.unicamp.br/courses/IA725/1s2011/projetos/lee-araujo-souza/index.html>>. Acesso em 18 jul. 2018.

STUDICA. **Maya**. Disponível em: <<http://studicabrasil.com.br/autodesk/maya/>>. Acesso em: 01 jul. 2018.

TELES, Vinícius Manhães. **Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. São Paulo: Novatec Editora Ltda, 2014.

UNITY. **Termo de uso da Unity**. Disponível em: <<https://unity3d.com/pt/legal/terms-of-use>>. Acesso em: 4 abr. 2018.

UNREAL ENGINE (2018). **Unreal Engine**. Disponível em: <<http://www.unrealengine.com/>> Acesso em: 26 jun. 2018.

VASCONCELOS, José Braga; RIBEIRO, Nuno. **Tecnologias de Programação de Jogos**. Lisboa: FCA - Editora de Informática, Ltda, 2013.