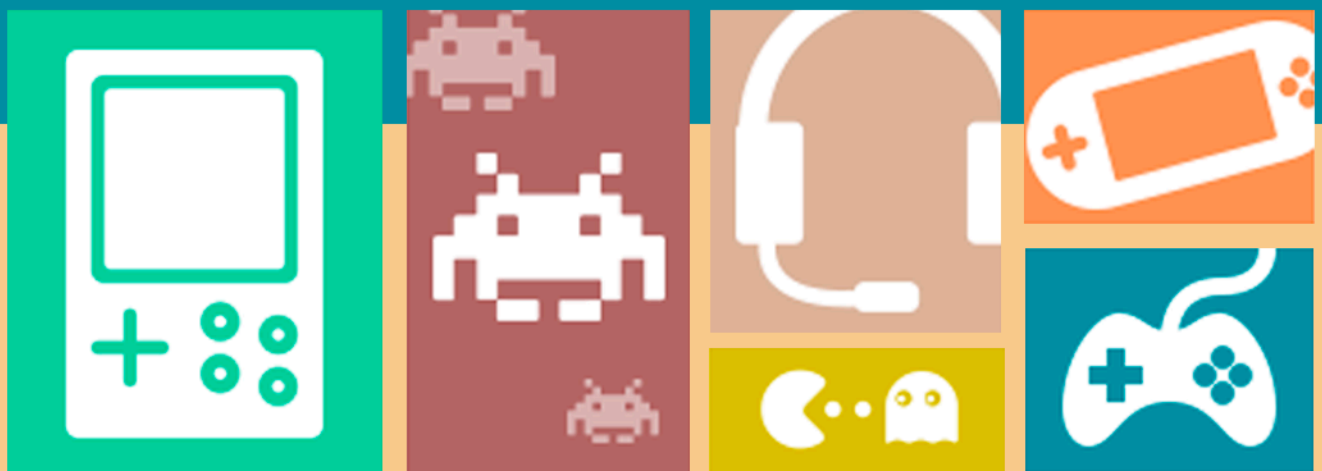


ANÁLISE COMPARATIVA DE METODOLOGIAS USADAS NO DESENVOLVIMENTO DE JOGOS DIGITAIS



Davi Gomes de Melo Silva
Alana Marques de Morais
Aline Marques de Morais

ISBN: 978-85-5597-033-7

**Análise comparativa de metodologias usadas no desenvolvimento
de jogos digitais**

**Davi Gomes de Melo Silva
Alana Marques de Moraes
Aline Marques de Moraes**
(Autores)

Instituto de Educação Superior da Paraíba - IESP

Cabedelo
2018



INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAÍBA – IESP

Diretora Geral

Érika Marques de Almeida Lima Cavalcanti

Diretora Acadêmica

Iany Cavalcanti da Silva Barros

Diretor Administrativo/Financeiro

Richard Euler Dantas de Souza

Editores

Cícero de Sousa Lacerda

Hercilio de Medeiros Sousa

Jeane Odete Freire Cavalcante

Josemary Marcionila Freire Rodrigues de Carvalho Rocha

Corpo editorial

Antônio de Sousa Sobrinho – Letras

Daniel Vitor da Silveira da Costa – Publicidade e Propaganda

Hercilio de Medeiros Sousa – Computação

José Carlos Ferreira da Luz – Direito

Marcelle Afonso Chaves Sodré – Administração

Maria da Penha de Lima Coutinho – Psicologia

Rafaela Barbosa Dantas – Fisioterapia

Rogério Márcio Luckwu dos Santos – Educação Física

Thiago Bizerra Fideles – Engenharia de Materiais

Thiago de Andrade Marinho – Mídias Digitais

Thyago Henriques de Oliveira Madruga Freire – Ciências Contábeis

Copyright © 2018 – Editora IESP

É proibida a reprodução total ou parcial, de qualquer forma ou por qualquer meio. A violação dos direitos autorais (Lei nº 9.610/1998) é crime estabelecido no artigo 184 do Código Penal.

O conteúdo desta publicação é de inteira responsabilidade do(os) autor(es).

**Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Padre Joaquim Colaço Dourado (IESP)**

A532 Análise comparativa de metodologias usadas no desenvolvimento de jogos digitais[recurso eletrônico]/ organizadores, Davi Gomes de Melo Silva, Alana Marques de Moraes, Aline Marques de Moraes. - Cabedelo, PB: Editora IESP, 2018.

56 p.

Formato: E-book

Modo de Acesso: World Wide Web

ISBN 978-85-5597-033-7

1.Computação. 2. Jogos digitais. 3.Computação - Desenvolvimento.4. Computação – Metodologias. I. Silva, Davi Gomes de Melo.II. Moraes, Alana Marques de. III. Moraes, Aline Marques de.

CDU 004

Bibliotecária: Elaine Cristina de Brito Moreira – CRB-15/053

Editora IESP

Rodovia BR 230, Km 14, s/n,
Bloco Central - 2 andar - COOPERE
Morada Nova. Cabedelo - PB.
CEP 58109-303

Prefácio

Este livro se refere ao primeiro exemplar de uma série organizada pela professora e pesquisadora Dr^a. Alana Marques de Moraes – membro do corpo docente dos cursos de Sistemas de Informação e Sistemas para Internet do Instituto de Ensino Superior da Paraíba -IESP.

Com o auxílio da docente Dr^a Aline Marques de Moraes da instituição referida e de alunos vinculados ao IESP, alguns trabalhos de destaque apresentados como trabalhos de conclusão de curso foram convidados a serem publicados no formato de livro eletrônico.

Agradecimentos

Agradeço primeiramente a Deus que nos deu energia e benefícios para concluir todo o trabalho. Nossa gratidão a todos os envolvidos neste projeto, que dedicaram noites e muito trabalho, especificamente ao tecnólogo Davi Silva. Agradecemos ainda a Professora Doutora Erika Marques, diretora da Instituição de Ensino Superior da Paraíba- IESP, pelo apoio incondicional para a concretização desta obra. Ao Professor Doutor Marcelo Fernandes, Coordenador de Sistemas, pelo suporte técnico, confiança e disponibilidade que permitiram a construção deste livro. Por fim, um último agradecimento ao professor Mestre Hercilio de Medeiros pelo apoio e suporte na edição e publicação deste livro.

Lista de Figuras e Gráficos

Figura 1- Fluxo do <i>Scrum</i>	15
Figura 2 - Metodologia de <i>Design</i> de Jogos Sérios para Treinamento.	19
Figura 3 - Documento de <i>design</i> do jogo.	20
Figura 4 - Ambiente do <i>Unity</i> 2017.	23
Figura 5 - Avatar principal do jogo.	25
Figura 6 – <i>Animator</i>	26
Figura 7 - Implementação 1.	27
Figura 8 - Função Pulando.	28
Figura 9 - Função Chão.	28
Figura 10 - Função de movimentação.	29
Figura 11- Função de ataque.	30
Figura 12 - Implementação da câmera.	31
Figura 13 - Função de verificação de plataforma.	32
Figura 14 -Função de patrulhamento.	33
Figura 15 - Documento de <i>design</i>	37
Figura 16 - Cenário do jogo.	40
Figura 17 - Diagrama de estrutura do “Desvia da mala”.	41
Figura 18 - Diagrama de classe do jogo “Desvia da mala”.	42
Figura 19 - Fluxo do processo do <i>Scrum</i> e do <i>Gamescrum</i>	43
Gráfico 1 - Comparação de uso das metodologias.	44

Lista de Quadros

Quadro 1 - Avanços históricos da Engenharia de <i>Software</i>	14
Quadro 2 – Aplicação das metodologias.	35
Quadro 3 - Comparação das metodologias.....	45

Sumário

Capítulo 1. INTRODUÇÃO	9
1.1. OBJETIVO GERAL	11
1.1.1 OBJETIVOS ESPECÍFICOS	11
1.2 CONTRIBUIÇÃO	11
1.3 ESTRUTURA DO LIVRO	12
Capítulo 2. REFERENCIAL TEÓRICO	13
2.1 ENGENHARIA DE <i>SOFTWARE</i>	13
2.2 METODOLOGIAS ÁGEIS	14
2.3 <i>SCRUM</i>	15
2.3.1 PRINCIPAIS PRÁTICAS DO <i>SCRUM</i>	16
2.3.1.1 SPRINTS (ITERAÇÕES)	16
2.3.1.2 PAPÉIS E RESPONSABILIDADES	16
2.3.1.3 PRODUCT SPRINT BACKLOG	17
2.3.1.4 KANBAN	17
2.3.1.5 DAILY SCRUM	18
2.3.1.6 SPRINT REVIEW MEETING	18
2.3.1.7 BURN DOWN CHART	18
2.4 <i>GAMESCRUM</i>	18
2.4.1 PRÉ-PRODUÇÃO	19
2.4.2 PRODUÇÃO	21
2.4.3 PÓS-PRODUÇÃO	21
2.5 <i>ENGINE</i>	22
2.5.1 <i>UNITY 3D</i>	22
Capítulo 3. DESENVOLVIMENTO DO PROTÓTIPO	24
3.1 ESTUDO DE CASO DO JOGO “DESVIA DA MALA”	24
3.2 CONSIDERAÇÕES INICIAIS	25
3.3 PRODUÇÃO DO JOGO	25
Capítulo 4. METODOLOGIA	34
4.1 PLANEJAMENTO DO JOGO	34
4.2 DEFINIÇÃO DAS METODOLOGIAS	35
4.3 APLICAÇÃO DAS METODOLOGIAS NO ESTUDO DE CASO	36
4.4 CONCEITO DO JOGO	36

4.5 PRÉ-PRODUÇÃO OU PLANEJAMENTO	36
4.6 PROTÓTIPO	37
4.7 PRODUÇÃO	38
4.8 ALFA	38
4.9 BETA	38
4.10 OURO	39
4.11 PÓS-PRODUÇÃO	39
Capítulo 5. RESULTADOS	40
5.1 JOGO	40
5.2 DIAGRAMAS	41
5.3 COMPARAÇÃO DE USABILIDADE	43
5.4 QUADRO COMPARATIVO	44
Capítulo 6. CONCLUSÃO	46
Capítulo 7. REFERÊNCIAS	47
APÊNDICE A	49

Capítulo 1.

INTRODUÇÃO

Nos tempos atuais, os avanços tecnológicos relacionados aos jogos digitais possibilitaram um grande crescimento de sua indústria. O estudo publicado pelo instituto SuperData, afirmou que o mercado de jogos movimentou US\$ 65 bilhões em 2015. Neste contexto, a China foi líder mundial com uma movimentação de US\$ 22,2 bilhões e os Estados Unidos ficou em segundo lugar com US\$ 22 bilhões faturados no setor. Diante desse crescimento em tempos de crise e movimentação de grandes quantias era inevitável que a qualidade dos jogos não aumentasse.

Entretanto o crescimento da indústria de jogos se deu pelos avanços tecnológicos e investimentos gerenciados de forma correta e com processos definidos. Segundo Salen e Zimmerman (2004), os jogos são considerados sistemas capazes de envolver seus jogadores em situações de conflito que são definidas por regras e geram um resultado quantificável. Com base nesse conceito, Salen e Zimmerman (2004) afirmaram que os jogos são sistemas, eles também descreveram que jogos são definidos por regras e, por fim, geram resultados.

Esse conceito só confirmou o que os profissionais da tecnologia já sabiam: não adianta grandes investimentos sem gerenciamento e processos para o desenvolvimento do produto. Percebe-se isso nitidamente quando se comenta sobre investimentos, sistemas, regras e resultados, partindo dessa visão resolveu-se fazer um estudo analisando e comparando as metodologias usadas para gerenciar esses ambientes.

Este estudo entende como jogos digitais as aplicações 2D ou 3D; voltadas para consoles, celulares ou para computadores; que possuam os mais variados públicos-alvo, e que tenha o objetivo de ensinar alguma habilidade ou apenas

entreter. Neste sentido, foram listados alguns exemplos de jogos: *For Honor*, *Fifa 17*, *Just Dance*, *QuinckBrain*, *League of LegendseDream League Soccer*.

Alguns desses jogos podem ser utilizados com outras finalidades e não só o divertimento como o *Just Dance* que ajuda adultos e crianças a sair do sedentarismo, fazendo o jogador se movimentar com as danças no intuito de somar pontos com acertos dos passos, tudo isso sendo analisado pelo *Kinect* (sensor de movimentos), outro exemplo é o *QuinckBrain* que é um jogo para celulares baseado em psicologia cognitiva e se propõe a ajudar o jogador a melhorar suas habilidades cerebrais fazendo-o exercitar sua atenção, velocidade, reação, concentração, lógica entre outros.

Para fazer uma análise do gerenciamento de projetos de jogos digitais, foram estudadas a fundo as metodologias ágeis como *Extreme Programming* (XP), *Scrum Gamescrum* que é um híbrido desenvolvido com a junção do XP com *Scrum*. O presente estudo desenvolveu um protótipo para a execução de um estudo de caso exploratório o mesmo é um jogo 2D que tem como enredo o cenário atual da política brasileira. A estrutura do jogo é constituída de uma dimensão 2D como mencionado, com um cenário da esplanada dos ministérios de Brasília em preto e branco para proporcionar uma maior harmonia entre os elementos do jogo que são o avatar principal, inimigo e plataforma. A foco principal do jogo é fazer o avatar desviar do inimigo até a linha de chegada. Uma grande motivação para este estudo foi a carência de pesquisas e documentações na área, a confusão que as empresas fazem entre processos de gerenciamento de desenvolvimento e processos de *design* o que é um grande erro tendo em vista que são criados para executarem tarefas distintas.

Levando em consideração que o XP e *Scrum* são modelos criados para o gerenciamento de equipes na construção de sistemas web como aplicativos móveis, sites entre outros, contudo os seus princípios básicos não facilitam a construção de games e para adaptá-los a essa nova realidade foram feitas modificações importantes como adicionar novos papéis, e incorporar outros tipos de atividades mais voltada para a área de *design* e modelagens de games.

De acordo com o andamento desta análise, foram discutidos pontos fortes e fracos de cada metodologia usada na construção de jogos digitais tendo em vista documentar novas soluções para os problemas que acontecem no cotidiano dos desenvolvedores de jogos.

1.1. OBJETIVO GERAL

O **objetivo geral** deste livro foi apresentar e discutir metodologias interativas e integradoras para analisar, projetar, implementar e avaliar jogos digitais, para atingir este objetivo central alguns objetivos específicos foram delineados e percorridos.

1.1.1 OBJETIVOS ESPECÍFICOS

- Elencar as metodologias ágeis e suas vertentes;
- Adequar metodologias de gerenciamento e construção de *software* a realidade dos jogos;
- Construir e modelar um protótipo que seja capaz de fornecer análises precisas em cada atividade do processo;
- Executar uma análise comparativa dos resultados obtidos.

1.2 CONTRIBUIÇÃO

De acordo com a problemática apresentada e os objetivos elucidados anteriormente, é possível destacar que as principais contribuições a fazer este estudo estão pontuadas abaixo:

- Há uma carência de estudos que explorem a engenharia de *software* no contexto dos jogos. Por isso este estudo se propôs a executar uma análise comparativa entre metodologias de desenvolvimento;
- Este estudo é uma abordagem introdutória prática sobre metodologias ágeis que pode ser útil para iniciantes na área de engenharia de *software* em jogos;

- Ao final deste documento, o presente estudo apresentou uma análise comparativa entre as metodologias *Gamescrum Scrum*, para concepção de uma abordagem híbrida incorporando os pontos fortes de cada uma delas;
- O presente estudo comparativo pode ainda ajudar empresas que não seguem uma metodologia a conhecer e aplicar uma abordagem em suas rotinas de desenvolvimento de jogos.

1.3 ESTRUTURA DO LIVRO

Este livro foi organizado em 7 capítulos, incluindo o introdutório, que discorreu sobre a apresentação do projeto, a descrição do problema, os objetivos do trabalho, contribuição e a organização do mesmo. O capítulo 2 apresentou os conceitos relativos ao *Scrum Gamescrum*, suas práticas e pontos importantes a serem destacados, também foram abordados a *Engine* para a construção do jogo e a linguagem de programação usada na implementação do mesmo. O capítulo 3 descreveu o desenvolvimento do jogo protótipo planejado e desenvolvido neste estudo. A seção seguinte (capítulo 4) apresentou a metodologia utilizada na análise comparativa entre o *Scrum Gamescrum* durante as principais fases de construção do jogo. Por fim, o último capítulo apresentou os resultados obtidos da comparação das metodologias e os gráficos resultantes das comparações.

Capítulo 2.

REFERENCIAL TEÓRICO

Este capítulo discutiu sobre abordagens da Engenharia de *Software* descrevendo sua história e objetivos, além de apresentar uma discussão sobre o surgimento das metodologias ágeis. Por fim, a presente seção apresenta duas metodologias relevantes para este relatório: o *Scrum* e o *Game Scrum*.

2.1 ENGENHARIA DE SOFTWARE

No início dos anos 70, a indústria de hardware teve um progresso significativo e os avanços tecnológicos possibilitaram a construção de grandes máquinas (DIJKSTRA, 1972). De fato, a indústria de hardware estava em um momento muito bom. Contudo, no ano de 1972, um cientista da computação Neerlandês chamado Edsger Dijkstra analisava o rápido progresso do *hardware* e das demandas por sistemas cada vez mais complexos em um ambiente nada favorável para os desenvolvedores (DIJKSTRA, 1972).

Segundo Edsger Dijkstra (1972), a engenharia de *software* era tratada como uma disciplina sem muita relevância no meio da computação e por decorrência disso alguns problemas aconteciam no desenvolvimento dos projetos, são eles:

- Projetos que não cumprem o cronograma;
- Projetos que excedam o orçamento;
- Produto de baixa qualidade ou que não atenda aos requisitos;
- Produtos não gerenciáveis e difíceis de manter e evoluir;

Após pontuar tais problemas, de acordo com Edsger Dijkstra, houveram grandes avanços científicos na área da Engenharia de *Software*, pesquisadores desenvolvedores e empresários enxergaram que se conseguissem desenvolver métodos capazes de resolver esses problemas a indústria de *software* poderia ser

muito lucrativa (DIJKSTRA, 1972). Como consequência, muitos estudos e modelos foram feitos ao longo da história como retrata o Quadro 1.

De modo geral, considerou-se que os objetivos primários da engenharia de *software* foram o aprimoramento da qualidade dos produtos de *software* e o aumento da produtividade dos engenheiros de *software*, além dos entendimentos dos requisitos de eficácia, ou seja, efetividade (MAFFEO, 1992).

Quadro 1 - Avanços históricos da Engenharia de Software.

Ano	Tema
1975 - década de 70	Programação Estruturada Metodologia de Desenvolvimento de Sistemas
1981 - década de 80	Estrutura de análise de Sistemas e metodologia de projetos Sistemas de investigação de aprendizagem
1990 1991 1995 1999	Programação orientada a objeto (OOP) Desenvolvimento rápido de aplicações <i>Scrum</i> Programação extrema
2000	Processo ágil Unificado

Fonte: Adaptação Sommerville(2007).

A engenharia de *software* visa melhorar a produção, manutenção e recuperação de sistemas de forma organizada e dentro dos prazos e custos determinados.

2.2 METODOLOGIAS ÁGEIS

As abordagens de desenvolvimento de *software* vêm mudando drasticamente na última década. Existem várias desvantagens acerca das metodologias tradicionais e bem documentadas, consideradas muito “pesadas” para algumas abordagens. Em resposta a essas abordagens, um novo grupo de metodologias têm aparecido nos últimos anos. Estas metodologias são as conhecidas como metodologias ágeis (AGARWAL e UMPHRESS,2008).

As metodologias ágeis começaram a surgir no final dos anos 90 devido à ineficiência das metodologias pesadas e projetos de *software* que extrapolavam o

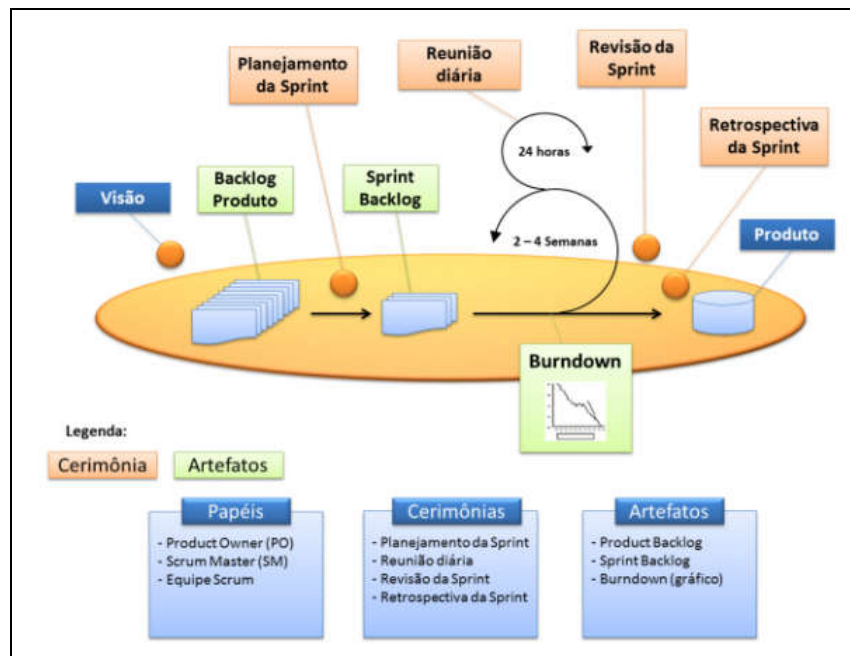
tempo de entrega, para resolver esses problemas foi definitivamente integrado aos projetos de *software* a Metodologia ágil com o objetivo de acelerar o desenvolvimento dos sistemas visando a melhoria contínua do processo (AGARWAL e UMPHRESS, 2008).

Neste livro, foi utilizado para a comparação as metodologias *Gamescrum* e *Scrum*. Os principais fundamentos dessas metodologias foram abordados nas próximas seções durante a discussão sobre a execução do estudo de caso do presente documento, construção de um jogo protótipo.

2.3 SCRUM

O *Scrum* é uma abordagem benéfica para as empresas que estão se debatendo e tentando seguir uma metodologia cascata, ou que simplesmente não estão seguindo nenhuma metodologia de desenvolvimento de *software*. É uma estrutura que transforma a criação de *software* complexo e sua entrega num prazo muito mais fácil (DIMES, 2014).

Figura 1- Fluxo do Scrum.



Fonte: <http://blog.contratado.me/wp-content/uploads/2017/01/processo-Scrum1.png>.

O *Scrum* é de fácil adequação a novos projetos e a projetos já em andamento, e tudo isso foi planejado desde do início pelos seus criadores que descrevem o *Scrum* como um processo incremental e interativo para desenvolvimento de produtos ou para gerenciamento. Pode-se observar o funcionamento do *Scrum* foi apresentado na Figura 1.

2.3.1 PRINCIPAIS PRÁTICAS DO SCRUM

No *Scrum* todos os papéis, atividades e práticas são previamente definidos para o início de cada projeto, abaixo foram detalhadas as principais práticas de tal metodologia.

2.3.1.1 SPRINTS (ITERAÇÕES)

O *Scrum* é dividido por ciclos chamados de *Sprints* ou interações, cada *Sprint* representa uma ou várias tarefas a serem executadas num período que pode durar entre 2 e 4 semanas dependendo da forma que o time decidir com o *ScrumMaster* e o *ProductOwner*.

2.3.1.2 PAPÉIS E RESPONSABILIDADES

Na metodologia *Scrum*, segundo Sabbagn (2014), existem três papéis definidos são eles:

a) ProductOwner

- Responsável por fazer o produto gerar lucro;
- Responsável por oferecer ao *Scrum Master* um bom ambiente para o desenvolvimento dos produtos;
- Responsável por aceitar ou rejeitar as tarefas executadas em cada *Sprint*;
- Responsável em decidir as datas de entrega;
- Responsável por aprovar ou reprovar os requisitos levantados;

- Na maioria dos casos é quem representa a equipe diretamente com clientes;
- Tem poder para inverter a ordem das *Product Sprint Backlog*.

b) *Scrum Master*

- Responsável pelo time;
- Responsável pelo funcionamento e produção do time;
- Protege o time de problemas externos;
- Responsabilidade com os *hardwares*;
- Responsável em garantir que o processo e metodologia estão sendo cumpridos.

c) *Scrum Team (Time)*

- Responsável pelo desenvolvimento tal como codificação, testes, configuração, modificações entre outro;
- O time é formado inicialmente por 5 membros podendo ter no máximo 9 membros multidisciplinares.

2.3.1.3 PRODUCT SPRINT BACKLOG

No início de cada Sprint todas as equipes são encarregadas de implementar uma funcionalidade do projeto, essas funcionalidades são mantidas em uma lista chamada de *product Backlog* (SABBAGN, 2014).

2.3.1.4 KANBAN

O kanban é um quadro onde fica organizada as atividades e itens do *Backlog*, também é utilizado para controle de tarefas que já foram implementadas (SABBAGN, 2014).

2.3.1.5 DAILY SCRUM

Entende-se por *Daily Scrum* a reunião diária com duração de no máximo de 15 minutos com todos os participantes em pé. O intuito da reunião é fazer três perguntas: o que fez no dia anterior? o que pretende fazer hoje? e se há algum impedimento para a execução da tarefa? A *Daily Scrum* pode ser realizada no início da manhã ou na parte da tarde (SABBAGN, 2014).

2.3.1.6 SPRINT REVIEW MEETING

O *Sprint Review Meeting* é uma reunião onde serão apresentados os resultados de cada Sprint. Nessa fase, também é relatado os erros que atrapalham o fluxo do projeto, e por fim analisar o que pode ser melhorado para que em novos projetos não aconteça erros acontecidos em projetos anteriores (SABBAGN, 2014).

2.3.1.7 BURN DOWN CHART

É um gráfico com dois eixos x e y que controlam as atividades que ultrapassam o tempo de entrega.

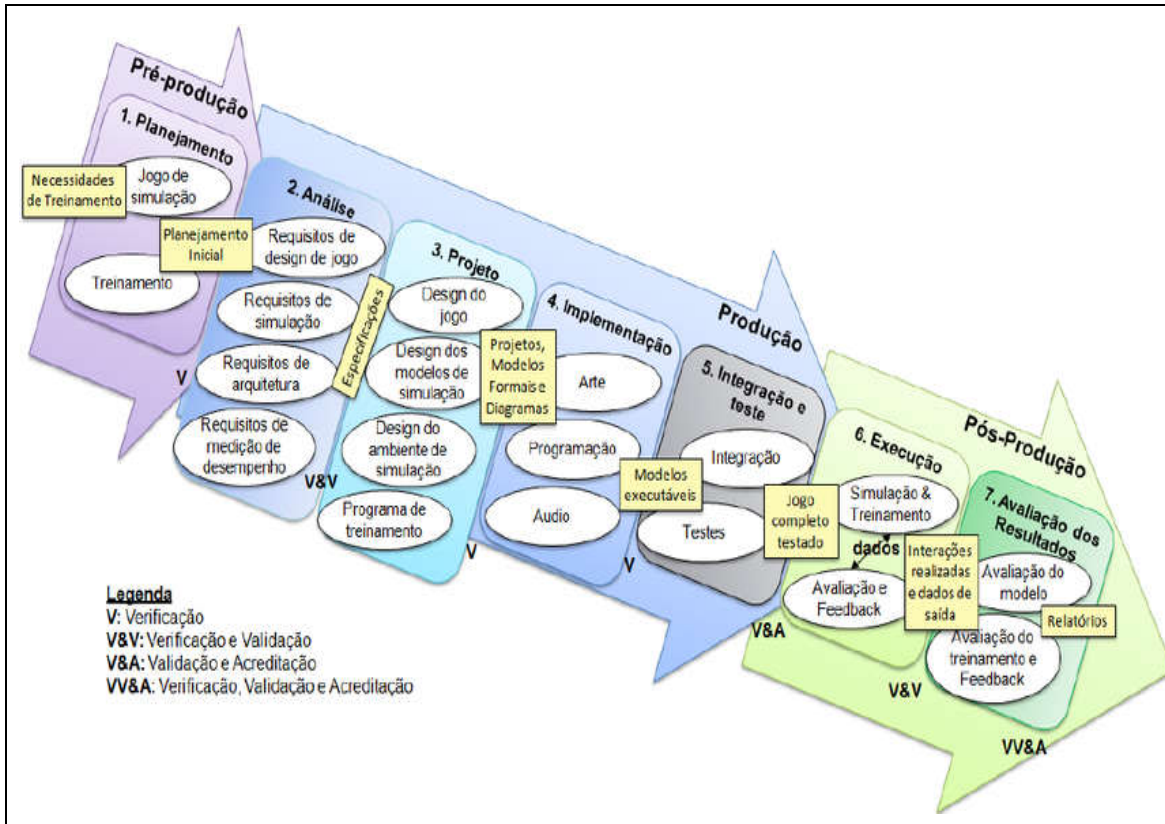
2.4 GAME SCRUM

O *Gamescrum* foi uma metodologia criada a partir da união do *Scrum* com *Extreme Programming* (XP) e teve como intuito resolver problemas recorrentes no gerenciamento de desenvolvimento dos jogos. O *Gamescrum* uniu as principais características das duas metodologias que lhe deram origem para o melhor funcionamento de suas *Sprints*, do *Scrum* foi utilizado o Gerenciamento de projetos e do XP a engenharia de projetos.

No *Scrum* as *Sprints* foram igualmente divididas e suas tarefas tem aproximadamente as mesmas características pelo motivo de ser voltado para o desenvolvimento de *software*. Diferentemente do *Gamescrum* que é voltado para jogos digitais e suas distribuições de trabalho no desenvolvimento não é igualmente distribuído durante suas iterações, para apoiar esta realidade

Gamescrumé dividido em três fases, discutidas abaixo (KEITH, 2010). A Figura 2 ilustra o fluxo do Gamescrum.

Figura 2 -Metodologia de Design de Jogos Sérios para Treinamento.



Fonte: Araújo (2013).

2.4.1 PRÉ-PRODUÇÃO

Na pré-produção, todos os fatores e contextos do jogo devem ser escolhidos tal como sua narrativa, fator de divertimento, público alvo, tipo de iteração, simulador, *serious game*, entre outros. Para o melhor funcionamento e esclarecimento do jogo um *brainstorming* é recomendado para desenvolver ideias e agregar novas sugestões da equipe.

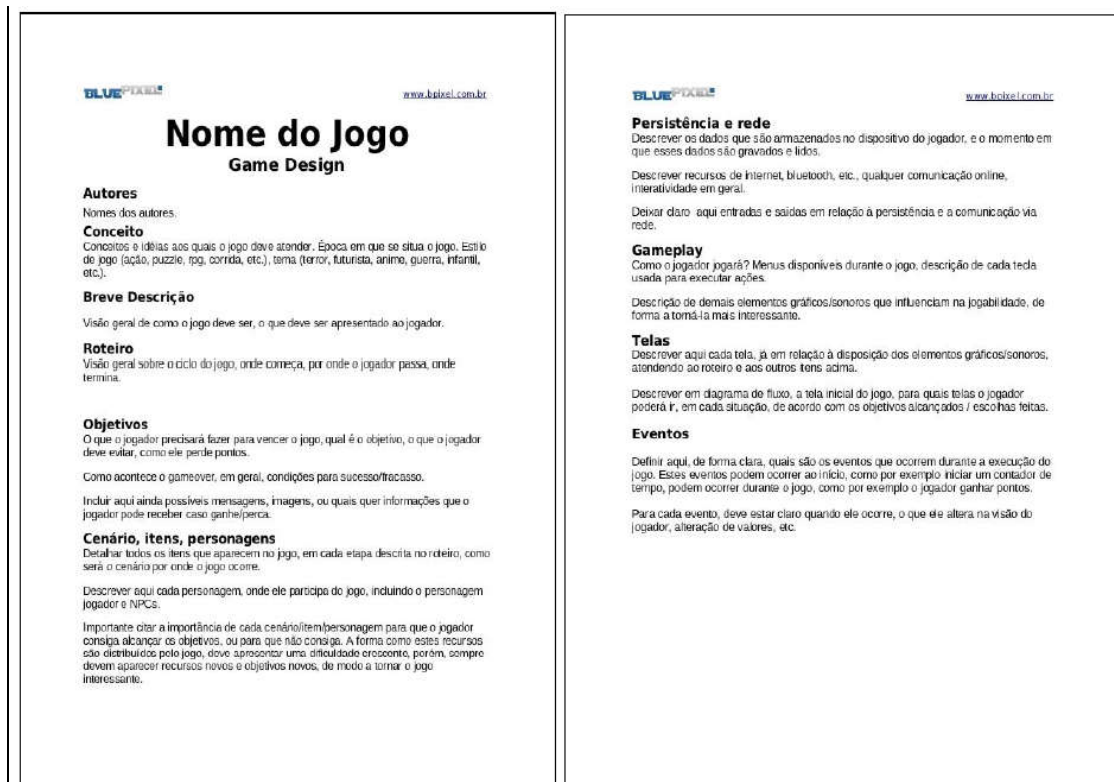
Uma pré-produção eficiente reduz a necessidade do encontro do elemento “diversão” durante o estágio de produção e permite que a equipe se concentre na implementação do jogo, em vez de experimentar com isso (KANADE e HADDAD, 2009).

Um protótipo é gerado na pré-produção do presente estudo. Esta aplicação é responsável pela navegação dos profissionais na interface do jogo, normalmente no fim dos testes o protótipo é descartado e o código não é reaproveitado.

Documento de *design* do jogo, esse documento é o resultado mais importante da pré-produção, ele é responsável por orientar o tamanho do projeto e toda parte de desenvolvimento e testes.

Não existe padrão para a elaboração do documento de *design* do jogo, contudo Schuytema (2008) afirma que este documento deve ter uma descrição abrangente do jogo em todos os seus aspectos, então que o time de desenvolvimento pode construí-lo (SCHUYTEMA, 2008). A Figura 3 ilustra um exemplo documento de *design* do jogo.

Figura 3 - Documento de *design* do jogo.



Fonte: <https://image.slidesharecdn.com/vgp-modelodedocumentodegamedesign-121105083609-phppapp01/95/modelo-de-documento-de-game-design-1-638.jpg?cb=1352105406>

No Apêndice A do presente livro, de *design* do jogo “Desvia da mala” foi apresentado em sua totalidade. Este documento contém a ficha completa de todas as informações do jogo e serviu de documentação base para sua construção.

2.4.2 PRODUÇÃO

Na produção é importante que o propósito do jogo esteja concluído, tal como o fator divertimento e o escopo do projeto. Nesta fase, o documento de *design* do jogo deve ser utilizado para a divisão das *Sprints*, pois as sprints serão definidas e as maiores e atrasadas serão divididas em peças menores com tarefas estabelecida

A criação do conteúdo artístico geralmente é organizada pelo Kanban possibilitando que a equipe responsável pela arte continue as tarefas gradativamente nas próximas sprints (GODOY; BARBOSA, 2010). Na fase de produção, o fator determinante é conciliar a programação com processo artístico para que os testes sejam bem-sucedidos.

2.4.3 PÓS-PRODUÇÃO

A pós-produção é iniciada quando o jogo está pronto, uma equipe é responsável em fazer os testes de jogabilidade e fator diversão, quando um erro é detectado na pós-produção dependendo no nível de criticidade o jogo pode voltar a produção. Na pós-produção é criado um post-mortem formado a partir dos comentários feitos durante todo o processo de construção do jogo (GODOY e BARBOSA, 2010).

Os postmortems são importantes porque permitem conhecer os pontos fortes e fracos do processo de desenvolvimento usado, problemas que ocorreram e sugestões para melhorias. Por meio desse feedback pode-se obter uma estimativa melhor para projetos futuros e ajustes necessários ao processo com maior assertividade (GODOY e BARBOSA, 2010, p. 294).

2.5 ENGINE

Diante de um cenário de muita competitividade como é o de jogos, ter uma *Engine* que possibilita à equipe um melhor aproveitamento no processo de desenvolvimento é primordial.

Diversas tecnologias utilizadas no desenvolvimento de jogos digitais têm sido geradas para fornecer uma base para desenvolvedores produzirem seus *games* com maior rapidez e eficiência, sem a preocupação com detalhes como aceleração de *hardware*, física, inteligência artificial, gerenciamento do som, etc. (BENIN e ZAMBIASI, 2010, p.1).

Há uma grande variedade de *Engine* para a construção de jogos, todas com o mesmo intuito de integrar as diferentes fases de desenvolvimento de um jogo e assim potencializar a produção de empresas e desenvolvedores independentes.

Para o desenvolvimento do protótipo deste livro foi usado a *EngineUnity 3D* que é uma ferramenta paga, que disponibiliza uma versão gratuita para estudantes, no tópico seguinte a *Unity 3D* é apresentada de uma maneira mais detalhada.

2.5.1 UNITY 3D

A *EngineUnity 3D* foi criado pela empresa *Unity Technologies*, e atualmente está na versão 2017. A *Unity 3D* foi elogiada no meio dos desenvolvedores graças a sua fácil adaptação de plataformas, ou seja, é uma *Engine* generalista que funciona bem com todos os públicos seja aqueles que desenvolvem em 2D, 3D *mobile* e *console*.

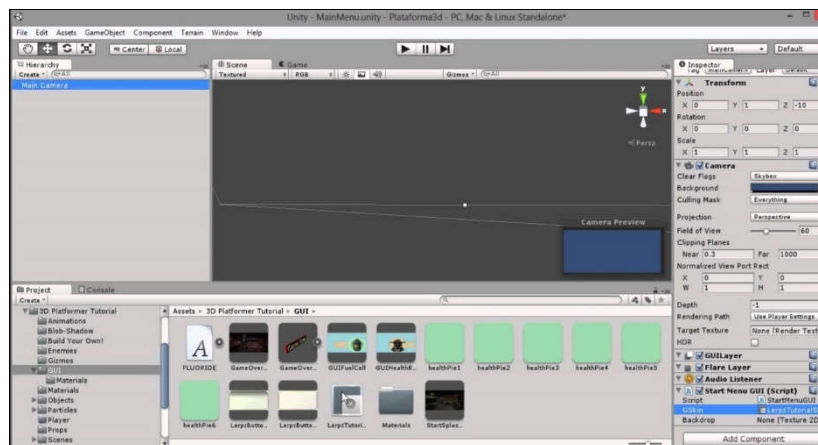
Outra novidade da *Unity 2017* é a integração de melhorias nos recursos artísticos, segundo o site oficial da empresa *Unity Technologies*, a *Unity 2017* apresenta novos recursos que ajudam equipes de artistas e desenvolvedores a criar experiências juntos. Novas ferramentas potentes, como *Timeline* e *Cinemachine*, permitem aos artistas criar conteúdo cinematográficos e sequências de jogo sem a necessidade de um engenheiro. Agora os criadores têm menos tempo de espera e implementação mais rápida (*Unity*, 2017).

A *Unity 2017* é um ambiente onde a criatividade de *designers*, artistas e desenvolvedores é somada a ferramentas potentes que permitem a reprodução instantânea e possibilita a edição rápida e melhor compreensão da equipe, funcionalidades com esses objetivos foram inseridas na *Unity 2017* segundo a documentação da *EngineUnity* que se encontra no site da *Unity Technologies*(2017), são elas:

- **Storytelling:** A ferramenta *Timeline* fornece aos artistas o poder de criar conteúdo cinematográficos deslumbrantes e sequências de jogabilidade.
- **Conteúdo cinemático:** Com o conjunto de câmeras inteligentes da *Cinemachine*, é possível controlar as tomadas como um diretor de cinema dentro do Editor.
- **Ferramentas pathfinding IA:** Seu PNJ irá navegar pelo mundo num abrir e fechar de olhos. As ferramentas avançadas de criação de perfil oferecem aos desenvolvedores as informações necessárias para otimizar sua performance.
- **Ferramentas personalizadas:** É possível ampliar o Editor com as ferramentas necessárias para combinar o fluxo de trabalho da sua equipe.

É possível visualizar o ambiente de desenvolvimento da *Unity* na Figura 4.

Figura 4 - Ambiente do *Unity 2017*.



Fonte: <https://i.ytimg.com/vi/LkmVaOS98Bs/maxresdefault.jpg>

Capítulo3.

DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo, foram apresentadas as estratégias e etapas percorridas para executar a comparação de forma coerente, com o objetivo de obter resultados válidos, além de um detalhamento acerca do jogo. Inicialmente foi necessário destacar a estratégia de utilização de um estudo de caso para analisar comparativamente as metodologias supracitadas de desenvolvimento de *software* em jogos.

3.1 ESTUDO DE CASO DO JOGO “DESVIA DA MALA”

Para executar a análise comparativa, foi fundamental delinear um estudo de caso experimental. A ideia do estudo de caso foi definir um contexto para verificação de efeitos de um determinado fenômeno ou produto frente a aplicação de metodologias distintas.

Para desenvolver um estudo de caso foi proposto o desenvolvimento de jogo intitulado “Desvia da mala”. O objetivo do jogo proposto foi ter um avatar que desviasse objetos que vem em sua direção. O jogador ganhará o jogo se conseguir desviar os objetos até a linha de chegada.

A ideia do jogo foi verificar mediante as etapas das metodologias, como cada uma delas lidou com o mesmo contexto de jogo. Esta análise permitiu perceber limitações e vantagens de cada uma das metodologias e obter conclusões sobre o processo.

3.2 CONSIDERAÇÕES INICIAIS

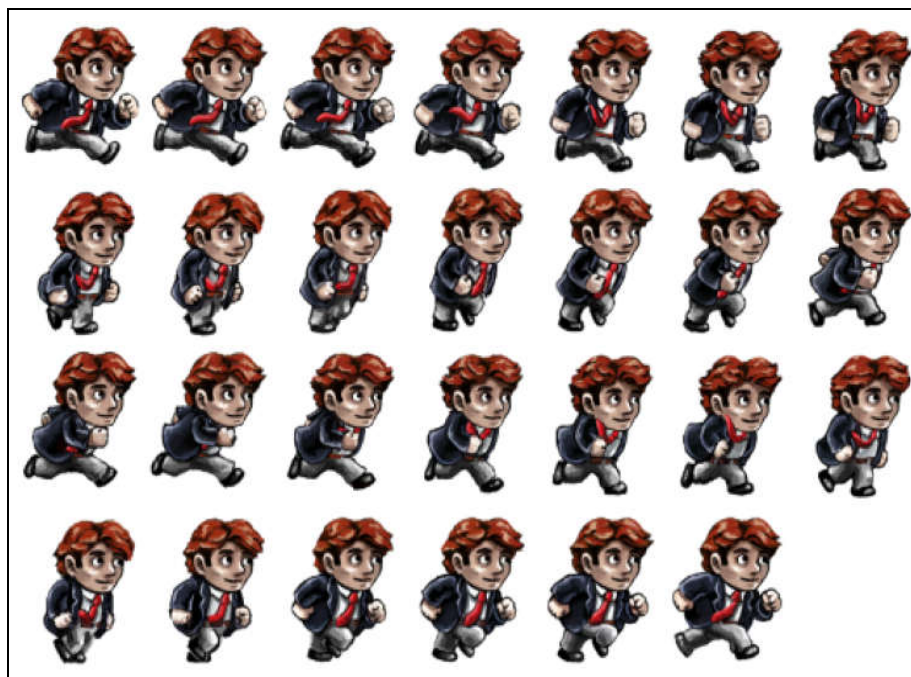
O jogo digital, chamado “Desvia da mala”, está compreendido como um jogo 2D, ou seja, significa que suas interações se limitam a caminhar em uma direção no cenário. Este tipo de jogo foi escolhido pelos seguintes motivos: pela viabilidade do tempo para a execução do projeto e por fornecer uma boa base para comparação das metodologias, foco principal do estudo de caso.

3.3 PRODUÇÃO DO JOGO

O tema corrupção está em alta no Brasil, pois toda semana surge um escândalo diferente relacionado ao tema. Por isso, este tema foi usado como enredo do jogo de forma irônica e intuitiva quando o mesmo propõe o jogador a pensar em política mesmo que seja lúdica.

Como já foi citado na seção 3.1.1 o jogo “Desvia da mala” é um jogo 2D com o objetivo principal de desviar dos objetos que vêm em sua direção. O jogo foi composto por um personagem principal que foi ilustrado na Figura 5.

Figura 5 - Avatar principal do jogo.

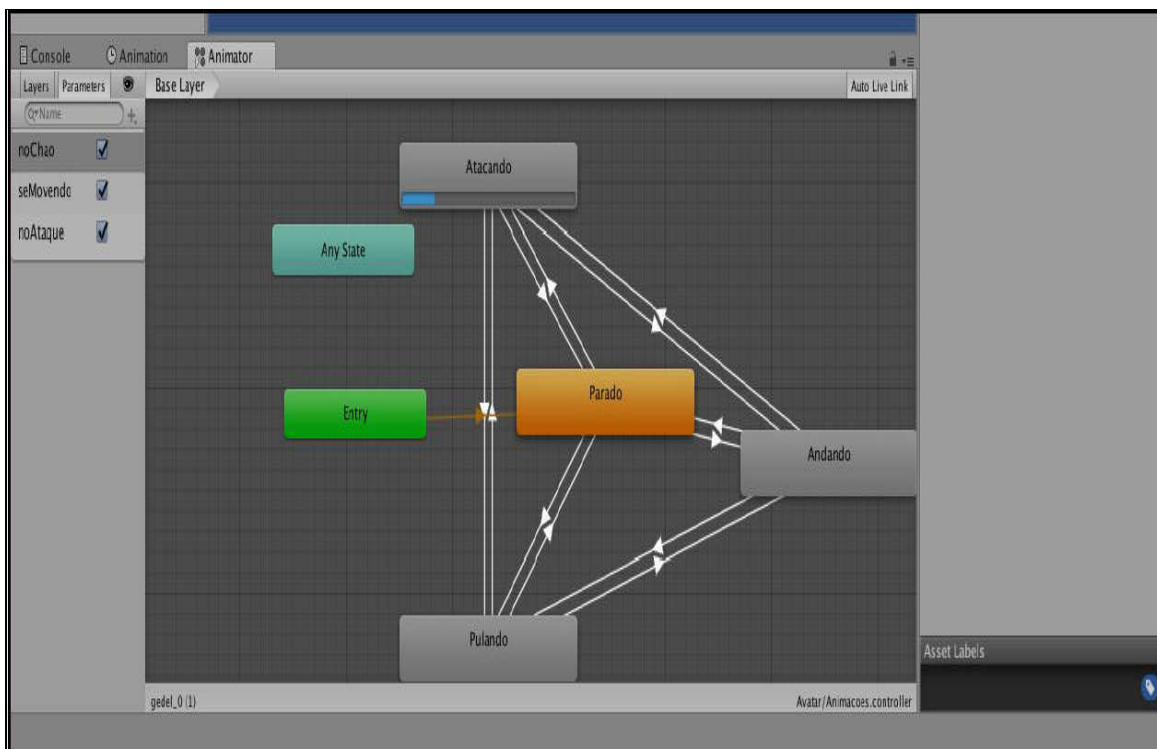


Fonte: https://cdn-images-1.medium.com/max/800/0*v-9KSDYtRcYloCHv.png

O avatar do jogo descrito na Figura 5 possui três variáveis principais, que foram: parado, correndo e atacando essas variáveis fizeram o avatar responder de forma programada o que o jogador determinou. Para implementação do jogo foi usado a linguagem de programação JavaScript, pelo domínio prévio da tecnologia.

A primeira iteração feita pelo avatar foi implementada na própria *Engine* descrita na (seção 2.5) pela funcionalidade do *Animator*. Este processo está ilustrado na Figura 6.

Figura 6 –Animator.



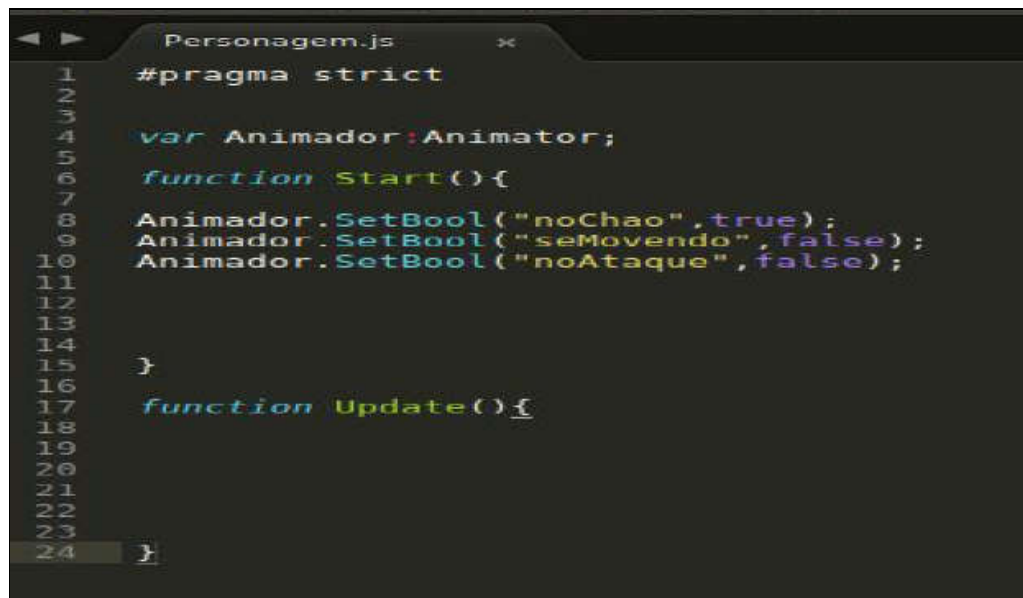
Fonte: Próprio Autor.

Já a segunda iterações foi implementada no Sublime text 3, usando o *JavaScript* para criação de Funções e métodos do jogo, na Figura 7 a seguir foi ilustrado a criação de duas funções que são: *Start* e *Update*.

- Função Start: Seta variáveis e guarda valores nelas, podendo iniciar novos valores pelo comando do usuário;

- Função Update: atualiza para novas iterações do avatar, podendo também conter laços de repetição e condicionais.

Figura 7 - Implementação 1.



```
1  #pragma strict
2
3
4  var Animador:Animator;
5
6  function Start(){
7
8  Animador.SetBool("noChao",true);
9  Animador.SetBool("seMovendo",false);
10 Animador.SetBool("noAtaque",false);
11
12
13
14
15 }
16
17 function Update(){
18
19
20
21
22
23
24 }
```

Fonte: Próprio Autor.

Ainda na classe Personagem foram criadas novas funções são elas: Pulando, Chão, Movendo e Atacando, funções essas que foram implementadas com o objetivo de dar mobilidade ao avatar.

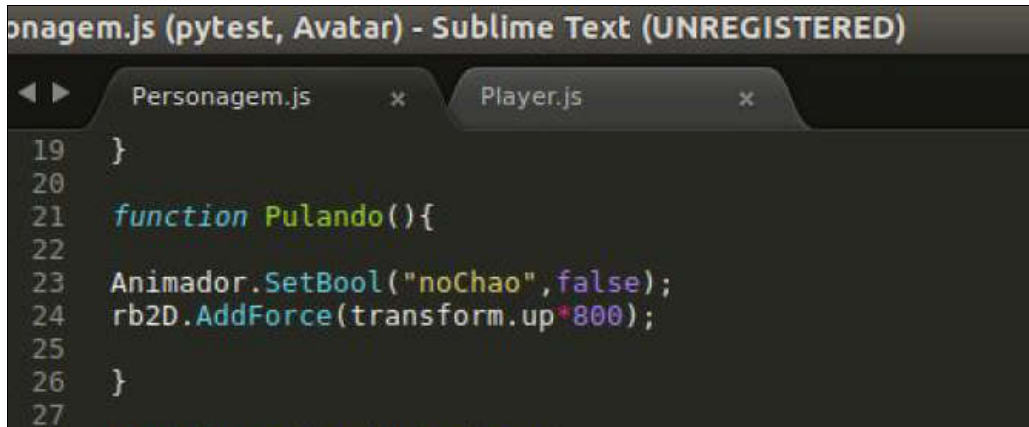
Uma parte muito importante do jogo é a escolha dos botões que irão invocar as funções de movimentação e ataque, uma escolha errada pode prejudicar a jogabilidade do mesmo. Quando a escolha dos botões é feita a fase seguinte é integrar os botões as funções escolhidas de forma que os botões possam invocar as funções em tempo real de jogo.

A integração do jogo “Desvia da mala” foi feita por meio das funções Pulando, Chão, Movendo e Atacando, porque até então nas funções Start e Update, o avatar só obedecia comandos implementados diretamente da Engine não havia uma interação entre jogador e avatar.

Diante dos pontos comentados, será apresentado e explicado telas e códigos das funções Pulando, Chão, Movendo e Atacando:

- **Pulando:** Como já diz o nome a função Pulando tem objetivo de fazer o avatar pular, e para que isso fosse alcançado foi adicionado uma força por meio do método *AddForce* na variável de *rb2d* que é uma variável do corpo do avatar. a Figura 8 a seguir ilustra bem isso que foi explicado neste tópico:

Figura 8 - Função Pulando.

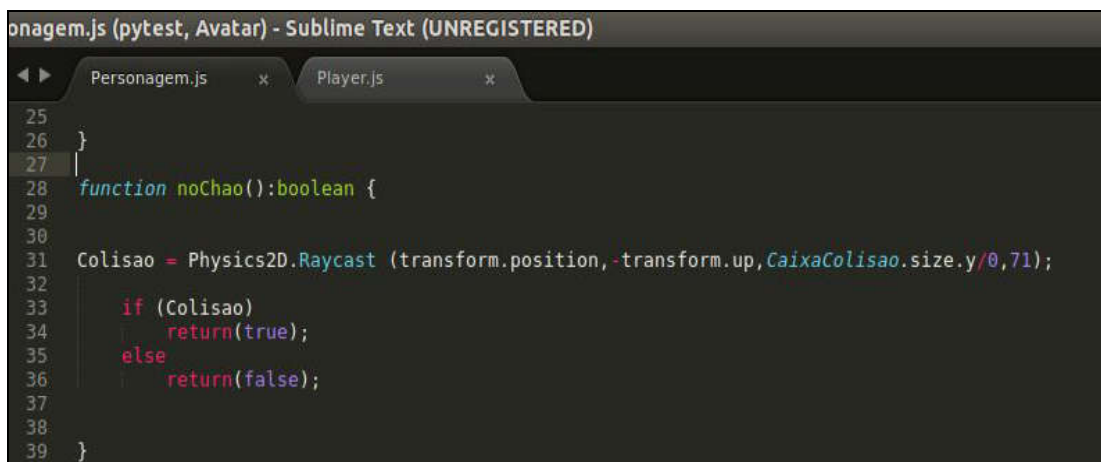


```
onagem.js (pytest, Avatar) - Sublime Text (UNREGISTERED)
Personagem.js x Player.js x
19 }
20
21 function Pulando(){
22
23   Animador.SetBool("noChao", false);
24   rb2D.AddForce(transform.up*800);
25
26 }
27
```

Fonte: Próprio Autor.

- **Chão:** A função Chão foi implementada com o objetivo de proporcionar a o avatar controle da queda depois do pulo, controle velocidade quando e controle de direção. A Figura 9 ilustra esta implementação.

Figura 9 - Função Chão.

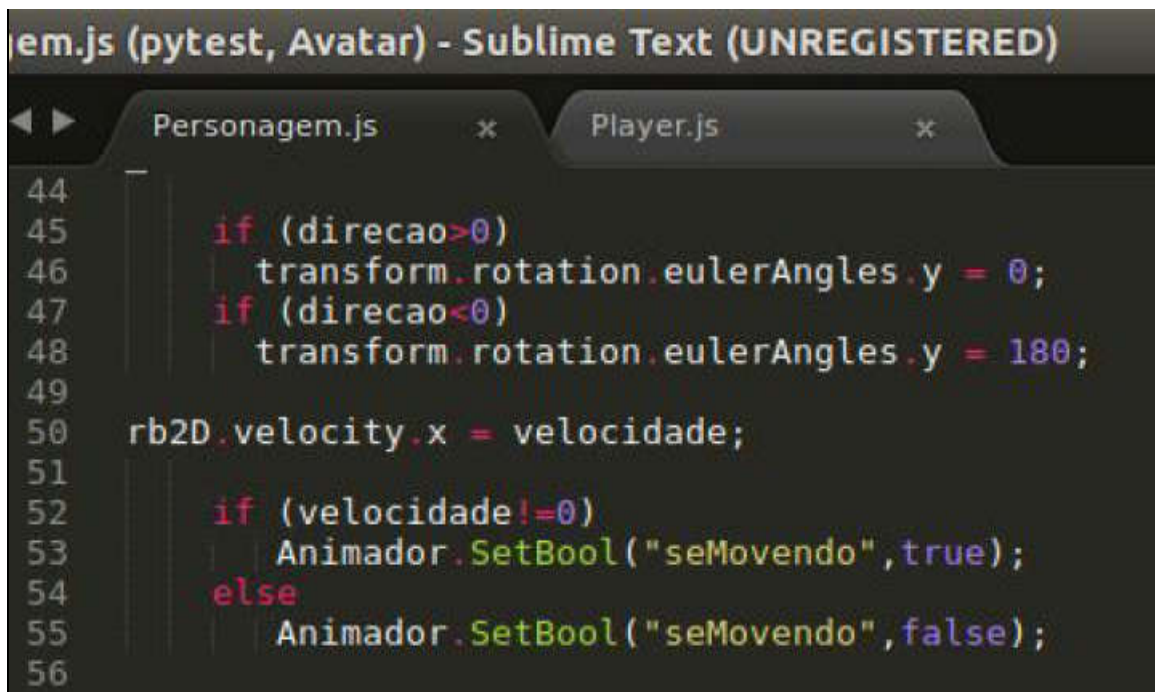


```
onagem.js (pytest, Avatar) - Sublime Text (UNREGISTERED)
Personagem.js x Player.js x
25 }
26 }
27
28 function noChao():boolean {
29
30
31   Colisao = Physics2D.Raycast (transform.position, -transform.up, CaixaColisao.size.y/0,71);
32
33   if (Colisao)
34     return(true);
35   else
36     return(false);
37
38
39 }
```

Fonte: Próprio Autor.

- **Movendo:** A função movendo é responsável pela parte de movimentação do avatar, ela contém duas variáveis chamadas de direção e velocidade, que são implementadas de forma que o jogador consiga direcionar o avatar.

Figura 10 - Função de movimentação.



```
em.js (pytest, Avatar) - Sublime Text (UNREGISTERED)
Personagem.js x Player.js x
44
45     if (direcao>0)
46         transform.rotation.eulerAngles.y = 0;
47     if (direcao<0)
48         transform.rotation.eulerAngles.y = 180;
49
50     rb2D.velocity.x = velocidade;
51
52     if (velocidade!=0)
53         Animador.SetBool("seMovendo",true);
54     else
55         Animador.SetBool("seMovendo",false);
56
```

Fonte: Próprio Autor.

- **Atacando:** A função atacando foi implementada de forma que o avatar pudesse fazer ataques repetitivos com pausas entre cada ataque de 0.5 segundos, o método *Time.deltaTime* permite o gerenciamento do tempo em funções desse *JavaScript*. A Figura 11 ilustra a função ataque.

Figura 11- Função de ataque.

```
em.js (pytest, Avatar) - Sublime Text (UNREGISTERED)
Personagem.js x Player.js x
61
62 function Ataque()
63 {
64
65 tAtaque+= Time.deltaTime;
66 if (tAtaque<0.5)
67
68 {
69
70 rb2D.velocity = Vector2(transform.forward.z, transform.forward.y)*20;
71
72 }
73
74 else
75
76 {
77
78 Animador.SetBool("noAtaque", false);
79 tAtaque= 0;
80 tSemAtaque = 0;
81
82 }
83
84
85
86 }
```

Fonte: Próprio Autor.

Em relação a Implementação da câmera, essa etapa é uma das mais importantes no processo de desenvolvimento de jogos digitais, pois a forma que o jogador interage com o avatar está diretamente ligada a MainCamera.

Há diversas formas de interação de câmera com avatar, como jogos de tiro onde a câmera é em primeira pessoa fazendo com que o jogador tenha a impressão que está dentro do jogo, outro exemplo são os jogos de futebol onde a câmera fica longe do avatar para que o jogador tenha visão de campo/cenário do jogo.

A câmera do jogo “Desvia da mala” foi implementada de forma parecida com a do jogo Super Mario, como cenário é 2D o avatar tem opção de ir aos eixos x e y dessa forma a implementação é feita a partir da distância entre o eixo escolhido e a variável do tipo câmera, no caso do “Desvia da mala” a variável é MCamera. a Figura 12 ilustra a implementação da câmera.

Figura 12 - Implementação da câmera.

```
(pytest, Avatar) - Sublime Text (UNREGISTERED)
Personagem.js x Player.js x
51
52 function MoveCamera()
53 {
54
55 // Eixo x
56
57
58 var DisX:float = Mathf.Abs (transform.position.x - MCamera.position.x);
59 if (!Mx && DisX > 2)
60 Mx = true;
61
62 if (Mx)
63 {
64
65 MCamera.position.x = Mathf.Lerp(MCamera.position.x, transform.position.x, Time.deltaTime*5);
66 if (DisX < 0.1)
67 Mx = false;
68
69 }
70
71 // Eixo Y
72
73 var Disy:float = Mathf.Abs (transform.position.y - MCamera.position.y);
74 if (!My && Disy > 4)
75
76 My = true;
77
78 if (My)
79 {
80
81 MCamera.position.y = Mathf.Lerp(MCamera.position.y, transform.position.y, Time.deltaTime*5);
82 if (Disy < 0.1)
83 My = false;
84
85 }
86
87
88 }
```

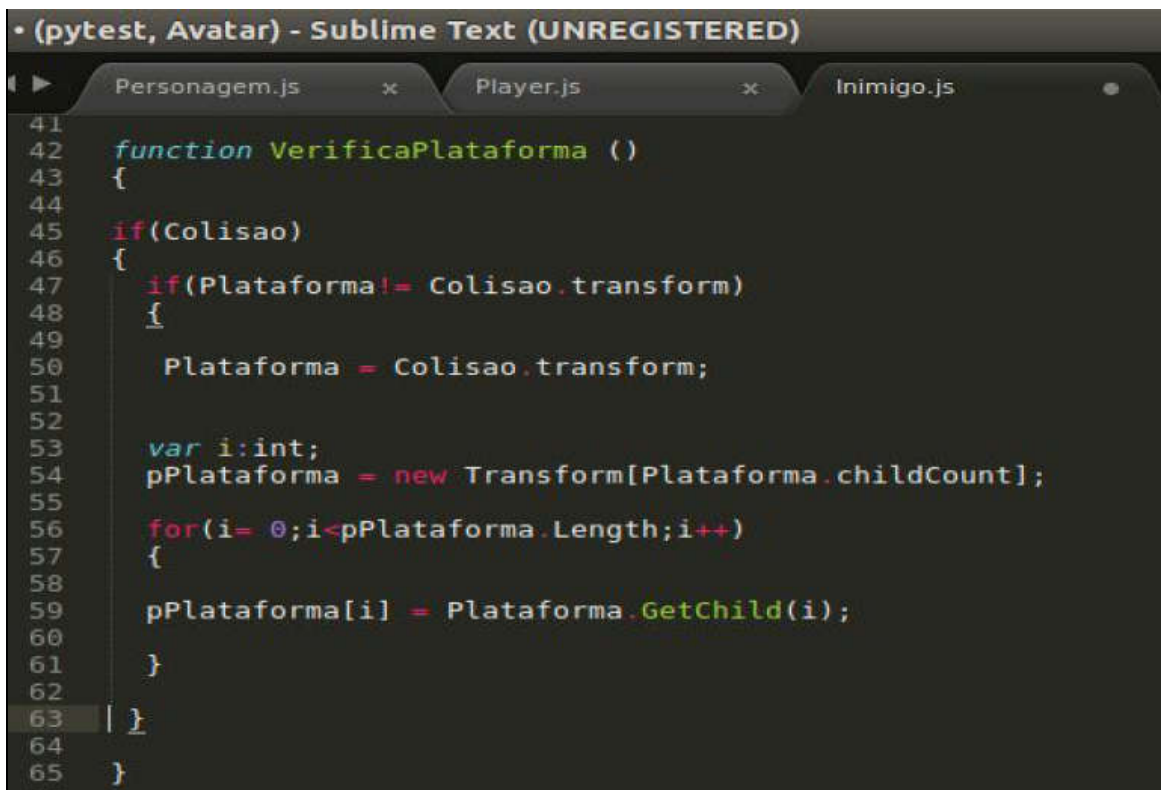
Fonte: Próprio Autor.

Implementação do adversário, esta parte do jogo “Desvia da mala” tem como objetivo construir o oponente do avatar principal usando técnicas de Inteligência Artificial(IA).

Para a implementação do oponente chamado de Mala foi aproveitado todas as funções Pulando, Chão, Movendo e Atacando do avatar principal, tendo em vista que o Mala vai ser controlado pela IA. Contudo para a implementação da IA foi criado duas Funções chamadas VerificarPlataformas e Patrulha, farei uma apresentação detalhada das mesmas a seguir.

- **VerificarPlataformas:** A implementação dessa função foi feita para o oponente verificar as plataformas do jogo, e isso só foi possível graças à criação de pontos nas plataformas, é basicamente uma condicional igual a plataforma que o oponente está, se for diferente disso ele percorre o vetor da plataforma que são os pontos criados. A Figura 13 a seguir ilustra bem o texto acima.

Figura 13 - Função de verificação de plataforma.



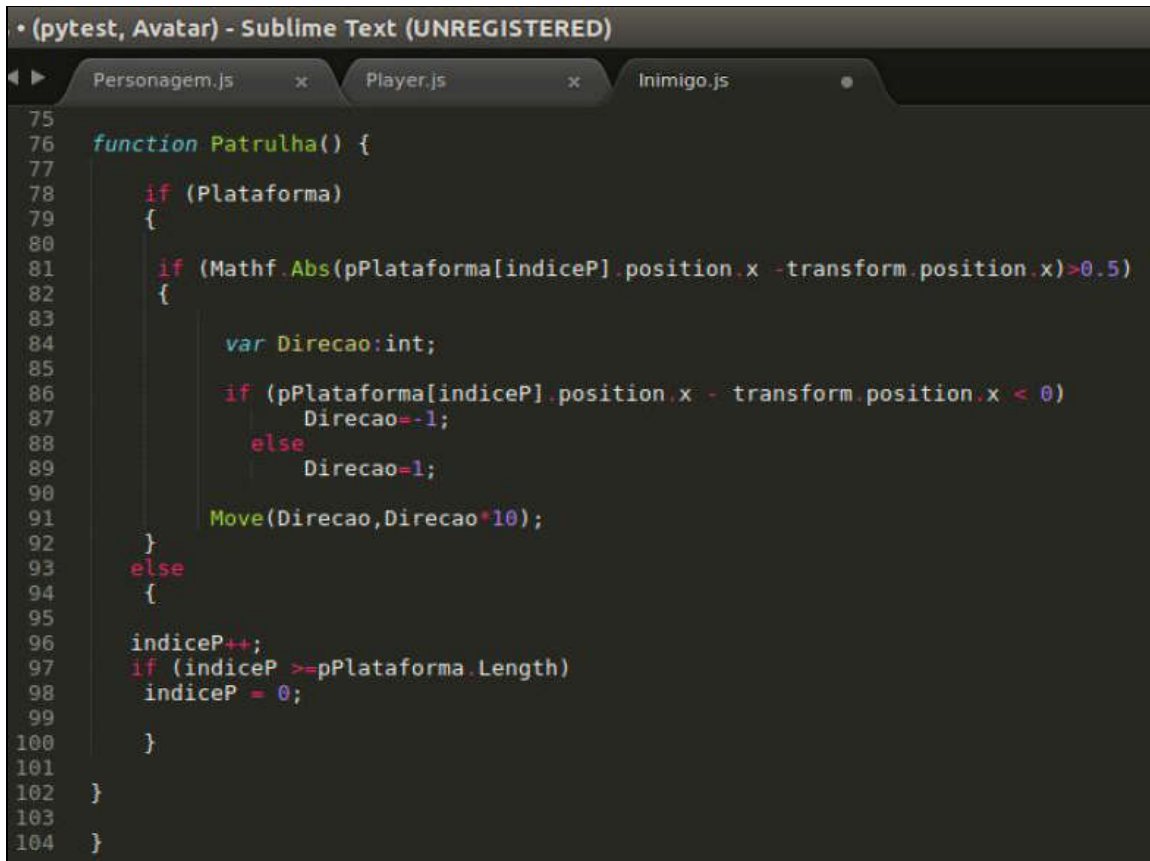
```
41
42 function VerificaPlataforma ()
43 {
44
45 if(Colisao)
46 {
47     if(Plataforma!= Colisao.transform)
48     {
49
50         Plataforma = Colisao.transform;
51
52
53         var i:int;
54         pPlataforma = new Transform[Plataforma.childCount];
55
56         for(i= 0;i<pPlataforma.Length;i++)
57         {
58
59             pPlataforma[i] = Plataforma.GetChild(i);
60
61         }
62
63     }
64
65 }
```

Fonte: Próprio Autor.

- **Patrulha:** A função patrulha pagar os pontos da plataforma e faz a implementação a partir deles, ou seja, se a distância entre a plataforma e a posição do oponente for maior que 0,1 metros de distância ele se moverá em direção avatar principal. A Figura 14 a seguir ilustra bem esse texto.

A implementação do adversário foi a última parte do desenvolvimento, com isso o jogo foi liberado para os testes de jogabilidade, os testes de jogabilidade ocorreram da forma esperada então o desenvolvimento do jogo foi finalizado.

Figura 14 -Função de patrulhamento.



```
75
76 function Patrulha() {
77
78     if (Plataforma)
79     {
80
81         if (Mathf.Abs(pPlataforma[indiceP].position.x -transform.position.x)>0.5)
82         {
83
84             var Direcao:int;
85
86             if (pPlataforma[indiceP].position.x - transform.position.x < 0)
87                 Direcao=-1;
88             else
89                 Direcao=1;
90
91             Move(Direcao,Direcao*10);
92         }
93     }
94     else
95     {
96         indiceP++;
97         if (indiceP >=pPlataforma.Length)
98             indiceP = 0;
99     }
100 }
101
102 }
103
104 }
```

Fonte: Próprio Autor.

Tendo em vista a complexidade envolvida no desenvolvimento do jogo principalmente no que tange os aspectos de processos de engenharia de *software*, considera-se que o processo proposto vem auxiliar a equipes envolvidas com construção de um jogo, no entanto o “Desvia da mala” foi desenvolvido por um único desenvolvedor.

Capítulo4.

METODOLOGIA

A engenharia de *software* por padrão orienta as equipes de desenvolvimento a seguir o fluxo proposto pela metodologia usada para gerenciar o desenvolvimento do projeto.

Como no jogo “Desvia da mala” foi usado duas metodologias para o desenvolvimento, um fluxo específico de uma metodologia não foi seguido e sim foi usado o que tratava melhor a fase em que o projeto estava. Contudo a estrutura base de todo projeto de jogo foi respeitada e foi seguida com o acréscimo de três novas que são alfa, beta e ouro sendo assim a sequência ficou:

- Conceito;
- Pré-produção ou planejamento;
- Protótipo;
- Produção;
- Alfa;
- Beta;
- Ouro;
- Pós-produção.

Essa sequência base foi seguida, e conforme cada uma se apresentava era discutido qual metodologia melhor desempenhava a mesma.

4.1 PLANEJAMENTO DO JOGO

A fase de planejamento do jogo foi tratada no conceito do jogo sendo finalizada na pré-produção, do planejamento foi gerado como resultado os artefatos Documento de *design* do jogo e protótipo, essas etapas vão ser apresentadas detalhadamente na aplicação da metodologia no estudo de caso.

4.2 DEFINIÇÃO DAS METODOLOGIAS

Como mencionado, as duas metodologias foram usadas no processo de desenvolvimento do jogo, sendo aplicada a que melhor executasse a fase em que o projeto se encontrava.

O *Gamescrum Scrum* alternarão entre as fases do desenvolvimento, sendo aplicado quando melhor atingisse os critérios de avaliação que foram propostos para escolhas da aplicação das metodologias neste projeto.

Crítérios esses que são produtividade, agilidade, recurso, técnica, correção de erro e gerenciamento de profissional sendo esse o mais complexo de analisar, pois o projeto foi desenvolvido por apenas um desenvolvedor, contudo foi aplicado no estudo de caso para melhor se aproximar da realidade. As metodologias foram abordadas em seções futuras, contudo a definição destas foram apresentadas no Quadro 2.

Quadro 2 –Aplicação das metodologias.

Fase	Metodologi a	Proposta
Conceito	<i>Scrum</i>	levantamento de requisitos
Pré-produção ou planejamento	<i>Gamescrum</i>	Pré-produção
Prototipo	<i>Gamescrum</i>	Pré-produção
Produção	<i>Gamescrum</i>	Produção(Desenvolvimento)
Alfa	<i>Gamescrum</i>	Produção(Testes)
Beta	<i>Scrum</i>	Qualidade de <i>software</i>
Ouro	<i>Scrum</i>	A fase ouro é a responsável pela distribuição do jogo por mídia física ou digital, diante disso nesta análise o <i>Gamescrum Scrum</i> não se adaptaram a essa fase.
Pós-produção	<i>Gamescrum</i>	Pós-produção

Fonte: Próprio Autor.

4.3 APLICAÇÃO DAS METODOLOGIAS NO ESTUDO DE CASO

Esta etapa descreve a forma que as metodologias foram usadas no processo de desenvolvimento do jogo. É importante salientar que para a aplicação das metodologias alguns pontos foram adaptados levando em conta que havia apenas um desenvolvedor para implementar todo jogo.

4.4 CONCEITO DO JOGO

Em relação a fase do conceito do jogo “Desvia da mala”, esta etapa foi executada durante dois meses, pois o mesmo envolveu um tema muito amplo que é política. Além disso, foi feito um estudo profundo em revistas, jornais entre outros que pudesse fomentar ideias sobre a temática. Público alvo, no conceito também é importante escolher o público alvo do jogo, no caso do “Desvia da mala” são pessoas que gostam de jogos 2D com contexto animado.

É importante salientar que para a elaboração do conceito do jogo foi utilizado a metodologia *Scrum*. O *Scrum* define este trabalho inicial do projeto como levantamento de requisitos.

O *Gamescrum* também foi usado para elaborar o conceito do jogo, ele apresenta a pré-produção para resolver essa atividade, contudo não foi viável usá-lo pelo fato de planejamento para a fase seguinte e com isso não prejudicar o processo de desenvolvimento do projeto, deixando muitas atividades para uma única fase.

4.5 PRÉ-PRODUÇÃO OU PLANEJAMENTO

Para o planejamento foi utilizado o *Gamescrum*, como já foi descrito na (seção 3.2.1), o *Gamescrum* tratou o planejamento na pré-produção fazendo com que o projeto ganhasse a cara do jogo. Com o conceito do jogo pronto foi possível elaborar o documento de *design* do jogo e protótipo. Na pré-produção do jogo “Desvia da mala”, foi escolhido o cenário, personagens, linguagem de programação e motor do jogo descrito na (seção 2.6.1).

O documento de *design* do jogo foi construído nesta parte do processo pois o principal artefato da pré-produção e o mesmo. A Figura 15 ilustra o documento de *design* do jogo “Desvia da mala”.

Figura 15 - Documento de *design*.

Desvia da Mala Game Design	
Autores:	Alana Morais, Davi Gomes.
Conceito:	O Desvia da Mala é um jogo com tema político que retrata a época atual. O estilo do jogo e ação com obstáculos.
Descrição:	O jogo Desvia da mala é 2D, que pretende apresentar para o jogador um cenário rápido exigindo o reflexo do mesmo.
Roteiro:	O jogo começa na plataforma 1, e para vencer o jogador precisa levar seu avatar até a última plataforma sem ser tocado pelo oponente.
Objetivo:	O jogo Desvia da mala tem como objetivo principal desvia ou atacar o oponente.
Cenário, Itens e Personagens:	Cenário é um desenho das esplanada dos ministério em Brasília, os itens são as várias plataformas que o jogo tem, e os personagens são avatar principal e o oponente.

Fonte: Próprio Autor.

4.6 PROTÓTIPO

No planejamento o documento de *design* do jogo e o protótipo é desenvolvido, nesta seção é apresentado um protótipo como resultado do artefato da pré-produção. O protótipo é uma primeira visão do jogo que serve para fins de apresentação a clientes e a equipes que vão desenvolver, geralmente os protótipos não servem para dar continuidade no desenvolvimento do jogo, o *Gamescruma* aconselha a não reaproveitar o código do protótipo.

Nesta fase, foi usado o *Gamescrum* pelo fato da fase protótipo fazer parte da pré-produção. Contudo o projeto estaria mais coerente dessa forma, tratando dois artefatos da mesma fase com a mesma metodologia.

Para a construção do protótipo foi feito um planejamento no conceito do jogo, para ter uma ideia de como o jogo seria, já tendo o conceito do jogo “Desvia da mala” formado, foi feito um protótipo da tela principal, para identificar os pontos de localização dos personagens do jogo, cenário e objetos. Todo esse processo foi seguido conforme a pré-produção aconselha, sendo adaptado a um único programador.

4.7 PRODUÇÃO

Para a produção do jogo foi usado a metodologia *Gamescrum*, ao invés do *Scrum*, pelo fato do *Gamescrum* tratar essa fase mais amplamente que o *Scrum*. O *Gamescrum* trata a produção do jogo com uma fase que leva o mesmo nome produção como está descrito na (seção 2.5.2), e dessa forma que foi gerenciado o jogo “Desvia da mala”, toda parte de implementação, game *design*, integração e teste foram feitos na produção.

4.8 ALFA

Na fase alfa o jogo é jogável do início ao fim, contudo o mesmo ainda apresenta alguns defeitos e a fase alfa foi implementada para tratar os erros do jogo, nesta fase do jogo “Desvia da mala” foi usado a metodologia *Gamescrum* pelo fato de toda parte de testes de código e teste interação ser na produção fazendo com que o projeto ganhe em tempo.

4.9 BETA

A fase beta é a fase que o jogo já se encontra pronto, contudo a necessidade de testes de qualidade antes do produto ir pro consumidor final, a fase que melhor trata qualidade é o *Scrum* com seus processos de qualidade que inclui reunião

defeedback e revisão do projeto, diante disso o jogo “Desvia da mala” gerenciou a fase beta pela metodologia *Scrum*.

4.10 OURO

A fase ouro trata da distribuição do jogo para fabricação, dentro desse trabalho não foi possível gerar resultado dessa fase, pois só as grandes empresas conseguem distribuir seus conteúdos no mercado pelo meio físico.

4.11 PÓS-PRODUÇÃO

A pós-produção do jogo “Desvia da mala”, foi tratado pela metodologia *Gamescrum* pois a mesma trata essa fase com a pós-produção que tem como objetivo de dar mais longevidade ao jogo no mercado lançando novas atualizações e versões.

Para conseguir chegar em um resultado aceitável, o jogo “Desvia da mala” tem como projetos futuros lançar atualizações melhoradas pelas plataformas de loja *online* do Android, conseguindo assim cumprir o objetivo da pós-produção que é a permanência do jogo no mercado e para acontecer isso o produto precisa estar atualizado.

Testes de integração também foram feitos na pós-produção para dar a garantia final para a equipe de desenvolvimento no caso ‘um programador’, de que o jogo está finalizado totalmente.

Capítulo 5.

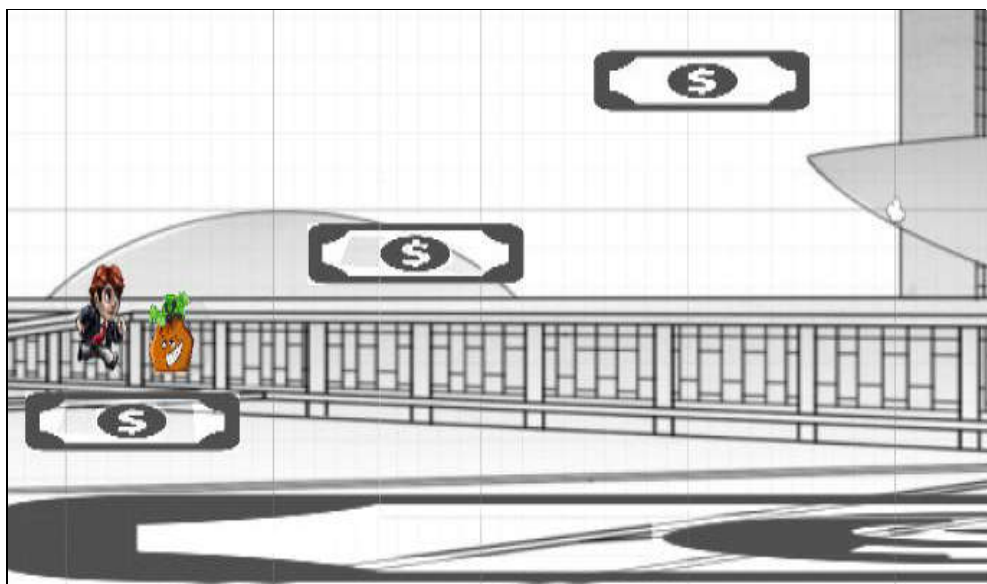
RESULTADOS

Nesta seção foi apresentado o jogo, os diagramas, aplicação das metodologias equadros para apresentar os resultados da comparação no processo dedesenvolvimento do jogo “Desvia da mala”.

5.1 JOGO

O primeiro resultado relevante deste trabalho foi o jogo em si, denominado “Desvia da mala”. Este projeto foi executado em um período de 6 meses. Na introdução e metodologia deste trabalho foi explicado mais detalhadamente o jogo, principalmente para mostrar o resultado até o momento, a Figura 16 ilustra a tela principal do jogo. Esta tela é o início do jogo “Desvia da mala”, a intenção foi mesmo essa de deixar o cenário de fundo em preto e branco para destacar o foco principal no avatar e no oponente.

Figura 16 - Cenário do jogo.



Fonte: Próprio Autor.

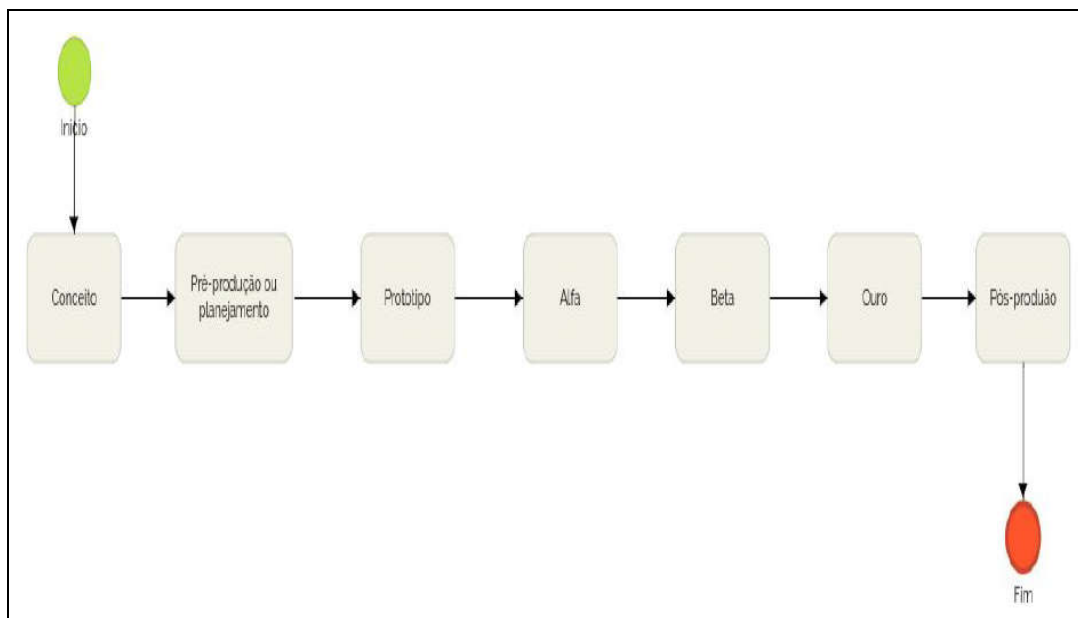
5.2 DIAGRAMAS

Nesta fase foram apresentados alguns diagramas que foram usados no processo de desenvolvimento do jogo “Desvia da mala”, para a gerenciamto do projeto sempre é necessária a diagramação de casos de uso, diagramas de processos, diagramas para o gerenciamto de metodologia.

Para esse projeto, foi elaborado diagramas específicos, pelo fato de ser um trabalho voltado para comparação de metodologias, então foi necessário a elaboração de processos com estratégias que atacasse os pontos principais do trabalho eliminando o que era considerado fraco pela análise de qualidade do próprio autor do trabalho, e usando o que a outra metodologia apresentava de uma forma mais eficiente e produtiva.

Para o início do processo de desenvolvimento, foi elaborado um diagrama base de todas as etapas que vão ser gerenciada pelas metodologias. A seguir, a Figura 17 foi ilustrado como resultado do processo que foi seguido, a mesma descreve todas as fases que foram percorridas no processo de desenvolvimento do jogo.

Figura 17 - Diagrama de estrutura do “Desvia da mala”.

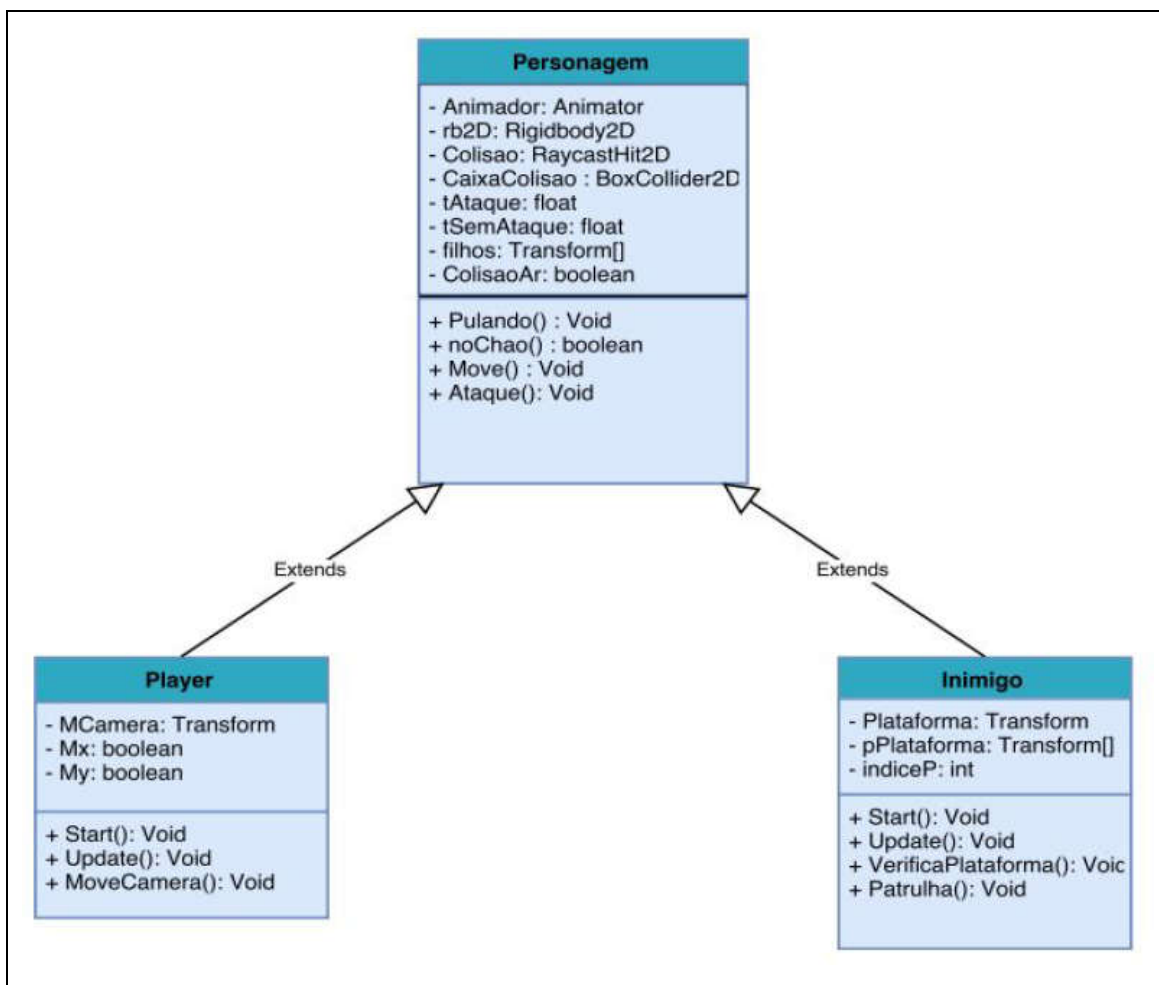


Fonte: Próprio Autor.

Para iniciar a implementação foi usado o diagrama de classe que contém todos as classes, atributos e métodos do jogo “Desvia da mala”. Mesmo será ilustrado no diagrama 2 a seguir, por questões de privacidade do projeto só as três principais classes foram apresentadas no diagrama.

A apresentação de apenas um trecho do diagrama não influenciará no entendimento do leitor, pelo fato do trabalho ser voltado para comparação e não implementação do jogo.

Figura 18 - Diagrama de classe do jogo “Desvia da mala”.

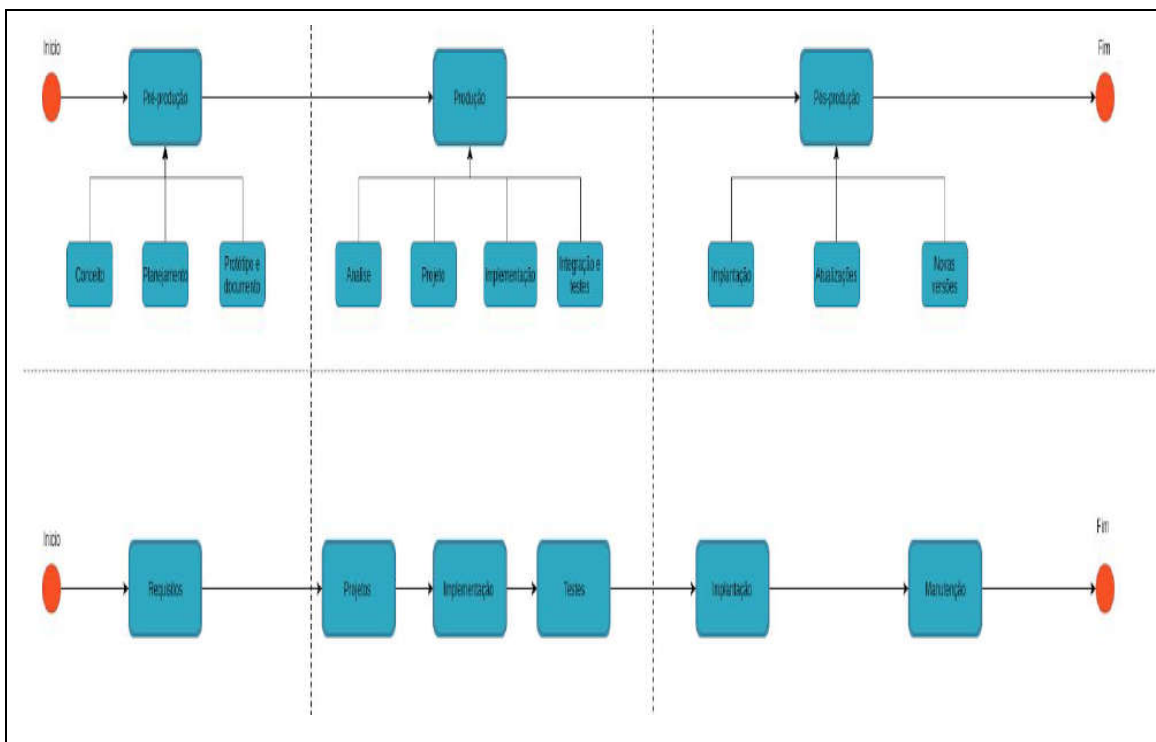


Fonte: Próprio Autor.

Por fim o último diagrama do trabalho vai ser apresentado a seguir, esse com a maior importância do trabalho pois o mesmo ilustra as duas metodologias que foram usadas no processo de desenvolvimento do jogo “Desvia da mala”.

Contudo é importante fazer uma observação a respeito da visualização e entendimento do leitor, para dar uma ideia de comparação no diagrama o processo da metodologia *Gamescom* ficou em cima da *Scrum*, fazendo com que a análise fique mais fácil e direta. A Figura 19 a seguir ilustra na visão de um diagrama o fluxo do processo das duas metodologias usadas no trabalho.

Figura 19 -Fluxo do processo do *Scrum* e do *Gamescrum*.



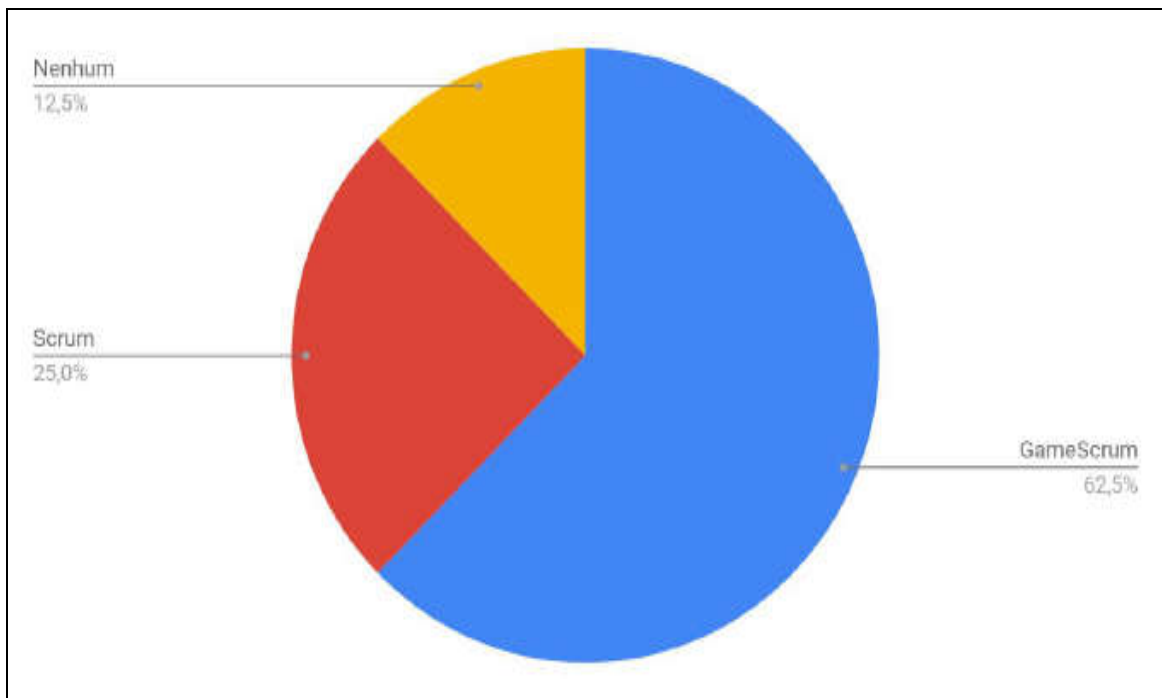
Fonte: Próprio Autor.

5.3 COMPARAÇÃO DE USABILIDADE

A comparação das metodologias foi feita a partir dos critérios de produtividade como já foi mencionado, nesta seção foi apresentado o uso de cada metodologias nas oito fases do projeto.

Nos resultados foi apresentado um quadro detalhando todo processo, no entanto a seguir é apresentado o gráfico 1 que ilustra a porcentagem de uso de cada metodologia no jogo “Desvia da mala”, o gráfico tem o percentual de quanto cada metodologia foi usada no processo de desenvolvimento de jogo, para que o gráfico ficasse mais próximo da realidade do processo usado no desenvolvimento foi incluído o percentual equivalente a uma fase sem o uso de nenhuma metodologia, pois na fase ouro nenhuma metodologia foi utilizada.

Gráfico 1 - Comparação de uso das metodologias.



Fonte: Próprio Autor.

Nas oito fases de processo do jogo foram usadas a metodologia *Gamescrum*, *Scrum* e em uma fase não foi gerenciada por nenhuma como já foi falado na (seção 3.4.5).

5.4 QUADRO COMPARATIVO

Nesta fase, foi apresentada um quadro comparativo das metodologias para desenvolvimento de jogos digitais, com o modelo de processo de engenharia de

software(Quadro 3). O Quadro ilustra a comparação entre as metodologias *Gamescrum Scrum* com o intuito de gerar discussão a respeito dos resultados obtidos neste trabalho.

Quadro3 - Comparação das metodologias.

Proposta		Processo Scrum	Processo Gamescrum
Autor		Takeuchi e Nonaka, 1986	Godoy, 2010
Aplicação		Desvia da mala	Desvia da mala
Física		2D	2D
V&V		Revisão e testes	Revisão e testes
Avaliação		Revisão do Sprint	Postmortem
Limitações		Avaliação e validação da metodologia	Falta de Doc. da mecânica de jogabilidade.
Design da arte do jogo	Planejamento	NA	Pré-produção
	Execução	NA	Produção
Legenda: NA: Não Abordado Doc.: documentação NE: Não Especificado			

Fonte: Rocha (2010).

Capítulo 6.

CONCLUSÃO

O desenvolvimento do presente estudo possibilitou uma análise de resultados da comparação de duas metodologias usadas para o desenvolvimento de jogos digitais. No processo de comparação foi construído um protótipo que foi utilizado para realizar os eventos de cada metodologia fazendo assim o estudo de caso ficar mais próximo da realidade e conseqüentemente minimizar a margem de erro.

Esse trabalho além de comparar duas metodologias também apresentou uma metodologia recente que é o *Gamescrum*, o mesmo foi criado em 2010 por brasileiros.

Os resultados obtidos durante o processo de comparação foram considerados deficientes diante das dificuldades enfrentadas no processo de comparação principalmente quando envolvia equipes, levando em conta que era apenas um desenvolvedor para construir todo jogo e elaborar todas as rotinas de eventos das metodologias.

Outro fator que pode interferir no resultado é o tempo, levando em conta que foi apenas seis meses para levantar os requisitos, elaborar as documentações e desenvolver um protótipo. Em cada fase de desenvolvimento do protótipo foi escolhida uma metodologia diferente, sendo assim possível comparar a que gerenciava melhor a fase.

Para que a análise comparativa atingisse um grau ainda menor de erros o ideal seria fazer dois jogos cada uma com uma metodologia diferente e de preferência com mais de uma pessoa, sendo assim o resultado seria ainda mais satisfatório, contudo esse processo pode ser executado e analisado em trabalhos futuros.

Capítulo 7.

REFERÊNCIAS

AGARWAL, R.; UMPHRESS, D. Extreme programming for a single person team. In: Proceedings of the 46th Annual Southeast Regional Conference on XX. New York, NY, USA: ACM, 2008. (ACM-SE 46), p. 82–87. ISBN 978-1-60558-105-7. Disponível em: <<http://doi.acm.org/10.1145/1593105.1593127> > Acessado em 18 de Outubro de 2017.

KEITH, C. (2010) Kanban for video game development. Disponível em: <<http://www.infoq.com/presentations/kanban-video-game-dev>>. Acessado em: outubro/2010. DIJKSTRA, Edsger Wybe. The humble programmer. 1972, 17p. Disponível em: <<https://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF> >. Acessado em 18 de julho de 2017.

DIMES, T. *Scrum* Essencial. Babelcube Inc., 2014. 48 p. Disponível em: <<http://goo.gl/Xn.TI7a> > Acessado em 18 de julho de 2017.

GODOY, A.; BARBOSA, E.F. (2010). “Game-*Scrum*: an approach to agile game development”. In: **Simpósio Brasileiro de Jogos e Entretenimento Digital**.

SABBAGH, R. (2014) *Scrum*: Gestão ágil para projetos de sucesso. São Paulo: Editora Casa Do Código (Digital). Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=pG-CcwAAQBAJ&oi=fnd&pg=PT9&dq=SABBAGH,+2014+Scrum&ots=ESMtpMLEse&sig=EwPmdvsVbDttSINi4PKeXcpKyQI#v=onepage&q&f=false >. Acessado em 18 de Novembro de 2017.

SALEN, K.; ZIMMERMAN, E. (2004) **Rules of Play: Game Design Fundamentals**. Cambridge, MIT.

MAFFEO, B. (1992), Engenharia de *Software* e Especificação de Sistemas, Rio de Janeiro, Ed. Campus, 1992.

ROCHA, ARAÚJO(2013). Metodologia de *Design* de Jogos Sérios para Treinamento: Ciclo de vida de criação, desenvolvimento e produção. SBC - Proceedings of SBGames .51

SCRUM. (2017) Disponível em: <<http://www.Scrumguides.org/>>. Acessado em: 26/10/2017.

SOMMERVILLE, I. (2007) “Engenharia de *software*” . 8. ed. Tradução de Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa. São Paulo: Pearson Addison-Wesley.

UNITY TECHNOLOGIES,(2017). Documentação da empresa usada na Engine Unity3D. Disponível em: <<https://unity3d.com/pt/unity>>. Acessado em 18 de novembro de 2017.

KANODE, C. M.; HADDAD, H. M. (2009) *Software Engineering Challenges in Game Development*. In: **6th IEEE International Conference on Information Technology**, LA, USA. Proceedings: IEEE Computer Society.

APÊNDICE A

Neste apêndice é apresentado o Documento de design do jogo “**Desvia da Mala**”, o mesmo foi elaborado na pré-produção com a finalidade de ter documentado as especificações do jogo para que ambas as partes envolvidas no processo de desenvolvimento tenham controle de eventuais mudanças no contexto do jogo. A seguir observa-se o documento de *design* do jogo.

Desvia da Mala

Game Design Document

Autores: Davi Gomes

Alana Morais

João Pessoa

Novembro 2017-11-18

1.História

Jogo 2D que tem como enredo o cenário atual da política brasileira. A estrutura do jogo é constituída de dimensões 2D como já foi falado, com um cenário da esplanada dos ministérios de Brasília em preto e branco para proporcionar uma maior harmonia entre os elementos do jogo que são: avatar principal, inimigo e plataforma. A foco principal do jogo é fazer o avatar desviar do inimigo até a linha de chegada.

2.Gameplay

Mecânica do jogo:

A mecânica do jogo desvia da mala e similar a do jogo Mario onde o avatar pula para Desviar de oponentes.

Desafios encontrados pelo jogador:

Os principais desafios encontrados pelo jogador no jogo desvia da mala são rapidez nos ataques de adversários e um cenário cheios de elementos que podem tirar sua atenção.

Como o jogador avança no jogo:

O jogador avança no jogo quando não é atingido pelos inimigos.

Relação entre gameplay e história:

O jogador se relaciona com a história através do avatar principal que

intuitivamente incentiva o jogador a executar os eventos como ataques e pulus.

3. Personagens

Avatar principal:



Inimigo:



4. Controles

Controle dos personagens

1. Tecla espaço: Pula
2. Tecla Direita: Para frente
3. Tecla Esquerda: Para trás
4. Tecla Ctrl: Ataque

5. Câmera

A câmera do jogo **Desvia da mala** é parecida com a do jogo Super Mario, como cenário é 2D o avatar tem opção de ir aos eixos **x** e **y** dessa forma a implementação é feita a partir da distância entre o eixo escolhido e a variável do tipo câmera.

6. Universo do Jogo

Descrição e ilustração do cenário do jogo:

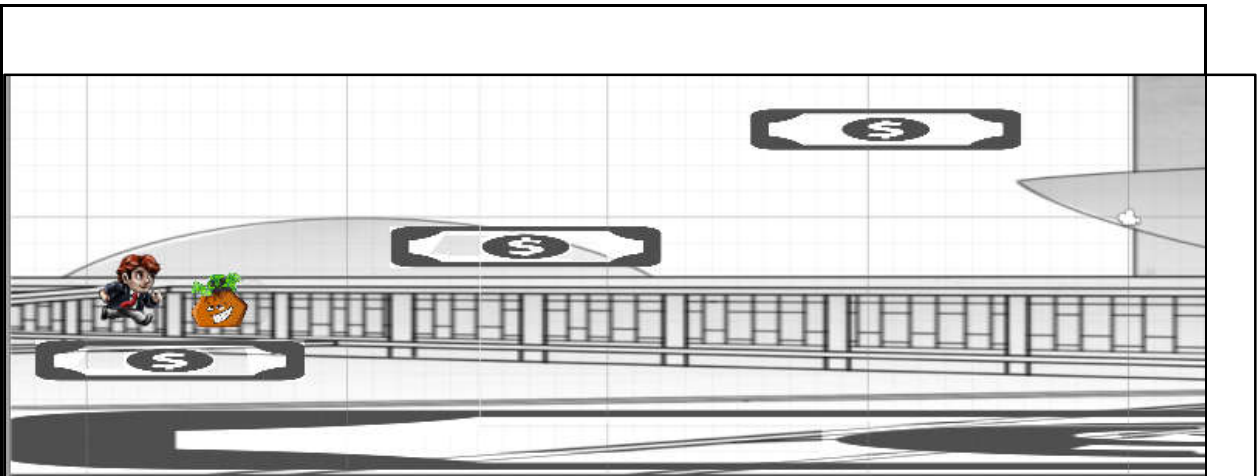
Uma ilustração das Esplanada dos ministérios em Brasília, uma figura no formato **png** em preto e branco.

Qual a estrutura do mundo: 2D

Tipo de música:

Nenhuma

Ilustrações da Primeira fase:



7. Inimigos

Descrição:

O inimigo é uma mala com dinheiro dentro, com um sorriso cínico, fazendo alusão aos roubos e malas de dinheiro que os políticos costumam roubar.

Em qual ambiente/fase cada inimigo vai aparecer:

Uma maior velocidade para atingir o jogador.

Como o jogador supera cada inimigo:

Pulando e atacando.

Comportamento e habilidades de cada inimigo:

Pular, correr e atacar.

8.Interface

Design e ilustração:

Todos os componentes do jogo são de autorias de terceiros.

Menu:

O menu do jogo deve ser algo simples, como o jogo é relacionado a política, não precisa de muitas cores e efeitos.

9.Cutscenes

Filmes serão incluídos no jogo:

Não

10.Cronograma

- Cronograma:

Tarefa/Semana	Junho				Julho				Agosto				Setembro				Progresso
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Escrever o GDD	█	█	█														CP
Apresentar GDD			█														CP
Selecionar/desenhar a arte dos personagens				█													EP
Selecionar/desenhar a arte dos cenários				█													EP
Desenvolver o sistema de controle do jogador					█												PL
Desenvolver sistema de mapas e fases					█												PL
Implementar a detecção de colisão						█											PL
Desenvolver sistema de pontuação						█											EP
Implementar inimigos							█										PL

Abreviações: CP: Completo

EP: Em progresso

PL: Planejamento