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

YURI DA SILVA CASTELO BRANCO

DESENVOLVIMENTO DE JOGO DIGITAL DE RPG

NATAL - RN

2022

YURI DA SILVA CASTELO BRANCO

DESENVOLVIMENTO DE JOGO DIGITAL DE RPG

Trabalho de Conclusão de Curso apresentado ao Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: Prof.ª M.ª Alyana Canindé Macêdo de Barros

NATAL - RN

2022

Branco, Yuri da Silva Castelo.
B816d Desenvolvimento de jogo digital de RPG / Yuri da Silva Castelo Branco. – 2022.
46 f.: il. Color.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. Natal, 2022.
Orientadora: Profa. Ma. Alyana Canindé Macêdo de Barros.

1. Desenvolvimento de sistemas. 2. Jogos digitais 2D RPG – Desenvolvimento. 3. Tiled (software). 4. Construct 2 (software). 5. Metodologia ágil Scrum. 6. Metodologia MDA. I. Barros, Alyana Canindé Macêdo de. II. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. III. Título.

CDU: 004.41

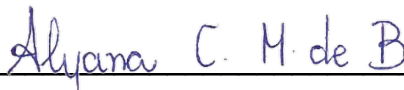
Catalogação na Publicação elaborada pelo Bibliotecário Joel de Albuquerque Melo Neto – CRB-15/320
Biblioteca Central Sebastião Fernandes (BCSF) – IFRN

YURI DA SILVA CASTELO BRANCO

DESENVOLVIMENTO DE JOGO DIGITAL DE RPG

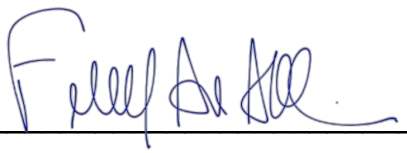
Trabalho de Conclusão de Curso apresentado ao Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Trabalho de Conclusão de Curso aprovado em 16/03/2022 pela seguinte Banca Examinadora:



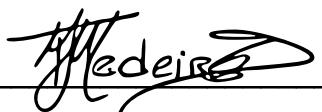
Prof.ª M.ª Alyana Canindé Macêdo de Barros - Orientadora

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



Prof. Dr.Fellipe Araújo Aleixo - Examinador

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



M.ª Tainá Jesus Medeiros - Examinadora Convidada

Thoughtworks

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a todos os professores da instituição que me proporcionaram o conhecimento, um caminho e maturidade para seguir em minha carreira profissional, além da própria instituição que me proporcionou muitas oportunidades. A orientadora deste trabalho, que me ajudou muito na explicação de conceitos essenciais para a elaboração. A todos os meus colegas que conheci no curso, que já me ajudaram a expandir meus conhecimentos, seja retirando dúvidas como ajudando com o manuseio de tecnologias das quais eu desconhecia ou era inexperiente. Por fim agradecer também a minha família que me ajudou e me apoiou nos estudos, me proporcionando uma oportunidade de ter um futuro profissional.

RESUMO

O presente trabalho trata-se de um projeto de desenvolvimento de software, mais especificamente de um jogo digital 2D do gênero RPG, desenvolvido com uso da engine Construct 2 e de outras ferramentas de apoio ao desenvolvimento do software, tais como softwares de edição de imagem, áudio, dentre outros. Tem como principal objetivo o desenvolvimento e entrega do protótipo jogável em alta fidelidade, da primeira fase do jogo digital. Para tal, o trabalho apresenta o processo de desenvolvimento de software, baseado nas fases de construção de engenharia de software e aplicação da metodologia ágil Scrum para a organização nas etapas de construção do software, além de apresentar o processo de desenvolvimento do ponto de vista de um jogo digital, utilizando como base a metodologia MDA, aplicada ao desenvolvimento de jogos.

Palavras-chave: Jogo digital; desenvolvimento; protótipo; Scrum; MDA.

ABSTRACT

The present work is a software development project, more specifically a 2D digital game of the RPG genre, developed using the Construct 2 engine and other software development support tools, such as image editing software, audio, among others. Its main objective is the development and delivery of the high-fidelity playable prototype of the first stage of the digital game. To this end, the paper presents the software development process, based on the phases of software engineering construction and application of the agile Scrum methodology for the organization in the stages of software construction, in addition to presenting the development process from the point of view of a digital game, based on the MDA methodology, applied to game development.

Keywords: Digital game; development; prototype; Scrum; MDA.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ATK	Attack
D&D	Dungeons & Dragons
DEF	Defense
HP	Health Points
HUD	Heads-up display
IA	Inteligência Artificial
IFRN	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
MDA	Mechanics-Dynamics-Aesthetics
MP	Magic Points
RPG	Role-playing Game
SPD	Speed
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1 INTRODUÇÃO

Jogo é um conceito antigo baseado em uma competição de duas ou mais pessoas na qual ambas obedecem às regras impostas para que se defina o vencedor. Esse é um conceito bastante presente na cultura de povos antigos como os da Grécia Antiga, com os jogos olímpicos. Segundo Huizinga (1980, p. 28, tradução própria)[1]:

> [...] jogo é uma atividade voluntária exercida dentro de certos limites fixos de tempo e lugar, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo e acompanhado de um sentimento de tensão, alegria e a consciência de que é "diferente" da "vida comum".

Ao longo do tempo, os jogos foram se tornando mais complexos e elaborados, desde uma simples competição para um jogo de tabuleiro baseado em estratégia, apresentando-se visualmente mais artístico com as grandes produções de jogos digitais da atual geração, voltadas para consoles, computadores e *smartphones*.

Os jogos digitais, ou vídeo games, são softwares nos quais o jogador interage por meio de dispositivos de entrada, como controle, teclado ou mouse, conectado a um console ou computador que produz e transmite imagens e sons a outro(s) dispositivo(s) de saída. Segundo Salen (2003, tradução própria)[2]:

> Jogos digitais são sistemas, assim como todos os outros jogos discutidos até agora. O meio físico do computador é um elemento que compõe o sistema do jogo, mas não representa todo o jogo. O hardware e software são apenas os materiais dos quais o jogo é composto.

Os jogos digitais podem ser subdivididos em categorias, tais como: aventura, ação, esporte, corrida, estratégia, RPG, entre outras. Esta última categoria de jogo digital, também conhecida como *Role-playing game* (RPG) ou jogo de interpretação

---

[1] No original: "[...] play is a voluntary activity or occupation executed within certain fixed limits of time and place, according to rules freely accepted but absolutely binding, having its aim in itself and accompanied by a feeling of tension, joy and the consciousness that it is 'different' from 'ordinary life'." (HUIZINGA, 1980, p. 28).

[2] No original: "Digital games are systems, just like every other game discussed so far. The physical medium of the computer is one element that makes up the system of the game, but it does not represent the entire game. The computer hardware and software are merely the materials of which the game is composed." (SALEN; ZIMMERMAN, 2003).

de papéis, é um tipo de jogo no qual o jogador assume o papel de determinado personagem e interpreta uma narrativa podendo realizar ações com regras predeterminadas impostas sobre o jogo. Segundo Sales (2022), “O RPG surgiu nos EUA em 1971, com a criação do The Fantasy Game, rebatizado em 1974 de Dungeons & Dragons (D&D) – algo como ‘Masmorras e Dragões’”.

Com isso em mente, junto com a experiência e conhecimentos adquiridos na graduação e o interesse pessoal na área de desenvolvimento de jogos, surgiu a ideia de desenvolver um software que seja um jogo digital de gênero RPG, um protótipo de alta fidelidade da primeira fase do jogo digital, em 2D usando pixel arte como estilo artístico e para navegadores, construído com o uso da engine Construct 2 para codificação, a fim de proporcionar um jogo de entretenimento, aos moldes de jogos retrôs como, por exemplo, *Final Fantasy*, *Gargoyle’s Quest*, *Chrono Trigger*, entre outros.

1.1 OBJETIVO GERAL

O objetivo principal é a construção e entrega de um protótipo jogável em alta fidelidade de uma fase de um jogo digital, especificamente um do jogo digital 2D para desktop, do gênero RPG, intitulado de *A Role Play Adventure.*

1.2 OBJETIVOS ESPECÍFICOS

- Elaborar o conhecimento teórico explicando os principais conceitos de base do projeto;
- Apresentar as ferramentas utilizadas para o desenvolvimento do jogo;
- Desenhar a arquitetura do sistema, incluindo os requisitos e casos de uso, que para definem as funcionalidades;
- Elaborar as especificações do jogo;
- Desenvolvimento de um nível do jogo.

1.3 METODOLOGIA CIENTÍFICA

Este trabalho tem como finalidade desenvolver uma pesquisa aplicada, pois utiliza o conhecimento adquirido durante a graduação para o desenvolvimento de um

software na prática, mais especificamente de um jogo digital. Segundo Gil (2019), a pesquisa aplicada, "abrange estudos elaborados com a finalidade de resolver problemas identificados no âmbito das sociedades em que os pesquisadores vivem". Os procedimentos técnicos seguem a classificação de uma pesquisa de desenvolvimento experimental, que, segundo o Decreto nº 5.798/06, conhecido como “Lei do Bem” e voltada para incentivos fiscais à inovação tecnológica, "são os trabalhos sistemáticos delineados a partir de conhecimentos pré-existentes, visando a comprovação ou demonstração da viabilidade técnica ou funcional de novos produtos, processos, sistemas e serviços ou, ainda, um evidente aperfeiçoamento dos já produzidos ou estabelecidos".

Com a finalidade de alcançar os objetivos apresentados, o projeto se deu com uso das etapas de um desenvolvimento de software e utilizou-se de duas metodologias para sua elaboração. Durante a etapa de planejamento foi utilizado a metodologia MDA, que visa definir e estruturar aspectos referentes ao jogo, desde suas mecânicas, jogabilidades e certas funcionalidades, gerando assim artefatos referentes ao jogo e seu funcionamento. Durante a etapa de projeto, se deu definições de arquitetura e funcionalidades, previamente definidas diretamente e/ou indiretamente pelo uso da metodologia MDA, além de definir as ferramentas utilizadas.

Durante a etapa de elaboração, foi utilizado a metodologia Scrum, com uso de sprints semanais, para desenvolvimento ágil e organização do desenvolvimento do projeto. O uso inicial do Scrum se deu com estudos, tanto das ferramentas definidas, como de sistemas similares para sua aplicação, além de definições a respeito de jogabilidade e narrativa. Nos momentos seguintes ocorreram uso de prototipagem das telas e seus respectivos desenvolvimentos. Nos momentos finais houve o desenvolvimento das regras lógicas de movimentação e fluxo do jogo, bem como codificação da inteligência artificial (IA) dos inimigos.

1.4 ESTRUTURA DO TRABALHO

A estrutura do presente trabalho está segmentada em 5 capítulos, incluindo este de introdução, no qual apresenta uma ideia geral, contendo também os objetivos principais e específicos, para o desenvolvimento deste trabalho. Além de contextualizar a respeito do trabalho e informar sobre a estrutura do documento.

O Capítulo 2 apresenta a fundamentação teórica, que contém as principais áreas de conhecimento aplicadas tanto para o desenvolvimento do software quanto o conhecimento da área de desenvolvimento de jogos, que auxiliaram na elaboração e entendimento do projeto. Apresenta também as ferramentas de desenvolvimento de jogos utilizadas.

O Capítulo 3 apresenta a modelagem e desenvolvimento do software. Apresenta as metodologias aplicadas ao projeto e ao desenvolvimento, o levantamento dos requisitos, a construção da arquitetura e o game design do protótipo, de um level do jogo digital.

O Capítulo 4 apresenta os resultados alcançados, as telas, os personagens e o fluxo do jogo. E por fim o Capítulo 5, com as conclusões do trabalho, com as considerações finais e expectativas para futuros trabalhos.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conhecimentos teóricos de base sobre jogos digitais, que são essenciais para o entendimento e o desenvolvimento deste trabalho, e em seguida as principais ferramentas utilizadas em sua elaboração.

2.1 CONHECIMENTO TEÓRICO

2.1.1 Jogos digitais

Jogos são qualquer atividade que exista pelo menos um jogador, e esse jogador está limitado a regras impostas pelo próprio jogo de forma que torne o desafio divertido, não muito diferentes dos jogos digitais, sendo estes um sistema como qualquer outro, com o diferencial de se utilizarem de hardwares e softwares. Segundo Salen (2003, tradução própria)[3], “Um jogo é um sistema no qual os jogadores se envolvem em um conflito artificial, definido por regras, que resulta em um resultado quantificável”. A autora acrescenta ainda que “Esta definição se assemelha estruturalmente à de Avedon e Sutton-Smith, mas contém conceitos de muitos dos outros autores também.” (SALEN; ZIMMERMAN, 2003, tradução própria)[4].

> Os jogos digitais podem e fazem bom uso dos dados: eles geralmente estão repletos de texto, imagens, vídeo, áudio, animações, conteúdo 3D e outras formas de dados armazenados. Na verdade, é justo dizer que os jogos digitais sobrecarregam as capacidades de renderização de dados dos computadores muito mais do que qualquer outro gênero de software de consumo. (SALEN; ZIMMERMAN, 2003, tradução própria)[5].

[3] No original: “A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.” (SALEN; ZIMMERMAN, 2003).
[4] No original: “This definition structurally resembles that of Avedon and Sutton-Smith, but contains concepts from many of the other authors as well.” (SALEN; ZIMMERMAN, 2003).
[5] No original: “Digital games can and do make good use of data: they are often filled to bursting with text, images, video, audio, animations, 3D content, and other forms of stored data. In fact, it is fair to say that digital games tax the data-rendering capabilities of computers far more than any other genre of consumer software.” (SALEN; ZIMMERMAN, 2003).

2.1.2 Jogabilidade

A jogabilidade de um jogo é um conjunto de regras que delimitam o que o jogador poderá ou não fazer dentro do jogo, em sua maior parte ela é influenciada pelo seu próprio gênero em que o jogo se encaixa. De acordo com Novak (2017, p. 187), “A jogabilidade pode ser definida como as escolhas, os desafios ou as consequências enfrentados pelos jogadores ao navegar em um ambiente virtual”.

Como um exemplo disso temos os jogos do gênero de plataforma, uma das funcionalidades mais marcantes em sua jogabilidade é a possibilidade de que o jogador, controlando um personagem, pode realizar ações como pular, correr, se mover, coletar itens pelo cenário e interagir com outros objetos do jogo em tempo real. No famoso jogo chamado “Super Mario Bros.", desenvolvido para o Famicom no ano de 1985 e classificado como um jogo do gênero de plataforma, você tem todas as funcionalidades anteriormente citadas, além de também existir condições de vitória e derrota.

Essas condições controlam o fluxo do jogo e levam o jogador cada vez mais perto ou mais distante de finalizar o jogo. Segundo Novak (2017, p. 186), "As condições de vitória definem como um game pode ser vencido pelos jogadores”. A autora acrescenta ainda que “As condições de derrota especificam como os jogadores perdem um game” (NOVAK, 2017, p. 188).

2.1.3 Heurísticas de jogos

Durante o desenvolvimento dos jogos, existem questões levantadas a respeito do nível de aceitação do público com o jogo, esses questionamentos estão relacionados diretamente tanto ao nível de entretenimento, engajamento e diversão que são proporcionados ao jogador.

> Nos jogos digitais os fatores chave para o sucesso são o prazer e a diversão, que dependem de uma boa usabilidade. Portanto, problemas relacionados aos elementos de entretenimento devem ser detectados e descritos numa primeira avaliação, pois precisam ser corrigidos assim como os problemas de usabilidade. (CUPERSCHMID; HILDEBRAND, 2013, p. 371).

Para chegar a um nível maior de polimento e se certificar da satisfação do jogador com relação ao jogo e questionamentos levantados a respeito do próprio jogo, são elaboradas avaliações heurísticas que visam analisar e avaliar o jogo dentro destes aspectos. Segundo Cuperschmid (2013, p. 372), “A avaliação heurística é um método de inspeção, que trata da avaliação da interface baseada numa lista de heurísticas pré-estabelecidas”. A autora acrescenta ainda que “[...] a heurística pode ser entendida como um conjunto de regras e métodos que conduzem à descoberta, à resolução de problemas e ajudam a traçar diretrizes para a concepção deste tipo de produção”. (CUPERSCHMID; HILDEBRAND, 2013, p. 372).

2.1.4 Design de níveis

Um jogo digital é composto por níveis, sendo estas partes do jogo onde é explorado e utilizado mecânicas projetadas para o mesmo, além da própria jogabilidade do jogo.

> Níveis são a estrutura dentro da qual os jogadores irão experimentar a jogabilidade que você projetou. Eles podem incluir história ou elementos do personagem que são cruciais para o desenvolvimento do jogo. Como os níveis são tão críticos, às vezes os designers de jogos podem se tornar um pouco controladores de como eles são implementados. (FULLERTON, 2014, p. 397, tradução própria)[6].

O Design de níveis é toda e qualquer elaboração de desafios para cenários no qual o jogador terá que percorrer e superar, para assim poder progredir. Segundo Novak (2017, p. 214), "define-se *design de níveis* como a criação de ambientes, cenários ou missões em um game eletrônico”. A autora acrescenta ainda que “os níveis podem ser utilizados para estruturar um game em subdivisões eficazes, organizar a progressão e aprimorar o modo de jogar. Ao projetar níveis, considere meta, fluxo, duração, disponibilidade, relações e dificuldade” (NOVAK, 2017, p. 214).

A elaboração dos níveis geralmente é atrelada a uma ferramenta de design de níveis ou editor de nível. Designers de níveis podem criar essas ferramentas e usá-las para definir os níveis de acordo com os elementos de história, mecânicas de

[6] No original: “Levels are the structure within which the players will experience the gameplay you have designed. They might include story or character elements that are crucial to the development of the game. Because levels are so critical, sometimes game designers can become somewhat controlling of how they are implemented.” (FULLERTON, 2014, p. 397).

jogo, ambientação e missões. Esses elementos são testados e melhorados para que se torne um ambiente interessante de se jogar.

2.1.5 Inteligência artificial

De acordo com Kaplan (2018, tradução própria)[7], define-se inteligência artificial (IA) como “a capacidade de um sistema de interpretar corretamente dados externos, aprender com esses dados e usar esses aprendizados para atingir metas e tarefas específicas por meio de adaptação flexível”. Dentro de um ambiente de desenvolvimento de jogos, a inteligência artificial (IA) é uma programação utilizada para gerar comportamentos a personagens e/ou inimigos que não são jogáveis, ou seja, não são conduzidos pelo jogador, podendo proporcionar maiores desafios ao jogador. Conforme Novak (2017, p. 322), “o programador de inteligência artificial (IA) dedica-se à criação de comportamentos que dão a impressão de comportamento inteligente”.

Ao longo dos anos de evolução no campo da inteligência artificial, algumas técnicas foram desenvolvidas, para diversos propósitos, tais como: raciocínio baseado em casos, máquina de estados finita, árvore de decisão, sistema de produção, lógica da primeira ordem, etc. De acordo com Rabin (2002, p. 3, tradução própria)[8], “[...] IA tem se tornado cada vez mais um dos fatores críticos para o sucesso de um jogo, decidindo quais jogos se tornam mais vendidos e determinando o destino de vários estúdios de jogos”.

A inteligência artificial é um elemento fundamental para o funcionamento das interações entre personagem, jogador e inimigos, além de também poder funcionar como um elemento para a elaboração de design de níveis, para controlar objetos e cenários. E segundo Novak (2017, p. 322):

> A IA de um game envolve respostas a estímulos, localização de caminhos, planejamento estratégico e até mesmo diálogos. Certos elementos de design relacionados ao modo de jogar, como o equilíbrio e o fluxo do game, também são incorporados pelo

[7] No original: “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation” (KAPLAN; HAENLEIN, 2018).
[8] No original: “[...] AI has increasingly become one of the critical factors in a game's success, deciding which games become bestsellers and determining the fate of more than a few game studios” (RABIN, 2002, p. 3).

> programador de IA para que o game seja desafiador e divertido de jogar.

2.2 TECNOLOGIAS UTILIZADAS

2.2.1 Tiled

O Tiled é um editor de nível 2D para ajudar a desenvolver mapas para jogos utilizando de tiles para a construção do cenário, tendo também suporte para separação de camadas. Utilizando de uma folha/imagem, também chamada de *tileset*, são separados pedaços da mesma em tamanhos iguais, cada pedaço é um tile que tem seu próprio número de identificação. Segundo Brinkmann (2018, tradução própria)[9]:

> O Tiled oferece a opção de importar conjuntos de peças e construir mundos com eles, para você usar com uma variedade de mecanismos de jogo. O próprio Tiled não constrói os jogos, apenas os mapas, mas os mapas podem ser usados até com programas como o RPGMaker para fazer jogos comerciais (o RPGMaker tem seu próprio editor de mapas, mas o Tiled também pode ser usado).

Na construção dos mapas, os dados são informações de posição e referências dos tiles, armazenadas em um array, não possuindo informações adicionais para cada posição, exceto flags que permitem a rotação de um determinado tile. Brinkmann acrescenta que (2018, tradução própria)[10]:

> Tiled é um editor de mapas flexível que você pode usar para diferentes propósitos. Embora você possa usá-lo para criar mapas para jogos de RPG - offline ou online - você também pode usá-lo para mapear sua casa, escola ou qualquer outra área real ou imaginativa.

---

[9] No original: “Tiled gives you the option to import tilesets, and build worlds with them, for you to use with a variety of game engines. Tiled itself does not build the games, just the maps, but the maps can be used with even programs like RPGMaker to make commercial games (RPGMaker has its own Map editor, but Tiled can also be used.)” (BRINKMANN, 2018).

[10] No original: “Tiled is a flexible map editor that you can use for different purposes. While you may use it to create maps for roleplaying games -- offline or online -- you may also use it to map your house, school, or any other area real or imaginative.” (BRINKMANN, 2018).

2.2.2 Construct 2

O Construct 2 é uma *game engine*, um programa com um conjunto de bibliotecas para simplificar o desenvolvimento de jogos eletrônicos, voltada para o desenvolvimento de jogos 2D. Utilizando uma interface com elementos de arrasta e solta, simples e intuitiva junto a um sistema de codificação baseado em folhas de eventos, o Construct 2 se torna uma ferramenta fácil de manuseio e de rápido aprendizado.

> O Construct 2 é uma ferramenta que não é necessário codificar, pois tudo é feito de forma automática. O usuário apenas irá criar o enredo do jogo, o restante da aplicação é feito praticamente através do mouse, com a ação de arrastar e soltar os objetos no cenário principal. (MEDEIROS; SILVA; ARANHA, 2013).

A codificação por meio de eventos, no qual cada evento é um código em JavaScript que está sendo reutilizado, se torna mais rápida e de fácil manutenção para desenvolvimento de jogos. Isso proporciona um melhor desenvolvimento e organização para o projeto, visto em conta destas qualidades esse motor de jogo foi escolhido para a construção e desenvolvimento do jogo *A Role Play Adventure*.

2.2.3 Scrum

O Scrum é uma ferramenta metodológica de desenvolvimento ágil que permite a elaboração de atividades incrementais, ou seja, periódicas e cíclicas, para o gerenciamento e organização de projetos de software. As atividades são executadas em etapas chamadas de sprints. Ao início de cada sprint, são planejadas as atividades a serem desenvolvidas durante um curto período de tempo.

> Eventos prescritos são usados no Scrum para criar uma rotina e minimizar a necessidade de reuniões não definidas no Scrum. Todos os eventos são eventos time-boxed, de tal modo que todo evento tem uma duração máxima. Uma vez que a Sprint começa, sua duração é fixada e não pode ser reduzida ou aumentada. Os eventos restantes podem terminar sempre que o propósito do evento é alcançado, garantindo que uma quantidade adequada de tempo seja gasta sem permitir perdas no processo. (SCHWABER; SUTHERLAND, 2013, p. 8).

Ao término de cada sprint, são entregues partes do sistema e é iniciado um planejamento para o ciclo de uma nova sprint, até a finalização do produto.

2.2.4 MDA

O framework MDA, também conhecido como Mechanics-Dynamics-Aesthetics, trata-se de uma ferramenta de desenvolvimento de jogos usada para analisar e estruturar o jogo. Segundo Hunicke (2004, tradução própria)[11], “MDA é uma abordagem formal para a compreensão de jogos – uma que tenta preencher a lacuna entre design e desenvolvimento de jogos, crítica de jogos e pesquisa técnica de jogos”.

O MDA é constituído por três aspectos, sendo estes as Mecânicas, as Dinâmicas e os aspectos Estéticos do jogo. Essas três camadas da ferramenta, ajudam na compreensão das heurísticas envolvidas e a definir uma estratégia na escolha e construção dos elementos do jogo, proporcionando assim uma melhor experiência para o jogador e atingindo os objetivos do jogo.

As Mecânicas são os componentes básicos do jogo, suas regras e ações no qual o jogador pode realizar, também podendo conter dados numéricos e algoritmos em suas descrições. As Dinâmicas descrevem os comportamentos das mecânicas em tempo real, agindo sobre as entradas e saídas de dados fornecidas pelo jogador ao decorrer do jogo. E por último a Estética, essa descreve a resposta emocional desejada, esperada no jogador, quando o mesmo interage com o sistema do jogo.

[11] No original: “MDA is a formal approach to understanding games - one which attempts to bridge the gap between game design and development, game criticism, and technical game research.” (HUNICKE, 2018).

3 MODELAGEM DO PROJETO

Neste capítulo são apresentadas fases de desenvolvimento do software, com o uso da metodologia Scrum, estrutura do sistema desenvolvido, sua arquitetura e funcionalidades, e em paralelo, fases de desenvolvimento do software como um jogo, apresentando o uso da metodologia MDA e especificações do jogo. Sendo elaborado utilizando os jogos “Final Fantasy”, desenvolvido para o Famicom no ano de 1987, e "Pokémon", desenvolvido para o Game Boy no ano de 1996, como uma referência para construção e idealização do jogo, tanto em funcionalidades quanto em mecânicas.

3.1 METODOLOGIAS DE DESENVOLVIMENTO

3.1.1 Scrum

No desenvolvimento deste projeto, foi utilizado a metodologia Scrum, para acompanhamento e agilidade na produção e codificação do sistema, contendo apenas um único integrante e utilizando sprints com duração de uma semana. Mediante a isto e ao contexto de um desenvolvimento de um software de um jogo digital, a metodologia foi adaptada inicialmente com uma etapa de estudo e prática, tanto das ferramentas quanto de sistemas similares, e posteriormente sendo utilizada para prototipagens, codificação das interfaces e sistemas do jogo.

Na elaboração deste projeto foram realizadas as seguintes sprints e suas respectivas atividades, podendo ser visualizadas na Tabela 1.

Tabela 1 - Sprints e atividades

Sprint	**Atividades**
[SP01] Idealização e pesquisa de jogos-modelo	[SP01.1] Pesquisa sobre jogos retrôs de gênero RPG para servir como exemplo [SP01.2] Definição da narrativa [SP01.3] Definição de gameplay e classificação
[SP02] Estudo das ferramentas	[SP02.1] Estudo e prática com o Construct 2 para codificação do jogo [SP02.2] Estudo e prática com o Tiled para construção de mapas
[SP03] Planejamento inicial para o	[SP03.1] Definição das funcionalidades para a tela de título

jogo	
[SP04] Prototipagem inicial	[SP04.1] Prototipagem para a tela de título [SP04.2] Coleta e criação do material audiovisual para a tela de título [SP04.3] Prototipagem do menu de opções [SP04.4] Coleta e criação do material audiovisual para o menu de opções
[SP05] Implementação da tela inicial	[SP05.1] Montagem e codificação do menu da tela de título [SP05.2] Montagem e codificação do menu de opções
[SP06] Desenvolvimento da primeira fase	[SP06.1] Codificação da lógica de movimentação do personagem no mapa [SP06.2] Construção do design de nível
[SP07] Desenvolvimento da primeira fase	[SP07.1] Codificação dos objetos interativos presentes no mapa
[SP08] Desenvolvimento do menu de inventário	[SP08.1] Prototipagem para a tela do menu de inventário [SP08.2] Montagem e codificação do menu de inventário [SP08.3] Codificação da listagem de itens do menu de inventário
[SP09] Desenvolvimento da batalha	[SP09.1] Prototipagem para a tela de batalha [SP09.2] Codificação do menu de batalha [SP09.3] Codificação da lógica de batalha [SP09.4] Codificação da IA dos inimigos

Fonte: Autoria própria (2022)

3.1.2 MDA

Para que fosse possível desenvolver o software do ponto de vista de um jogo, a metodologia MDA foi aplicada para definir os três aspectos estabelecidos pela ferramenta, que compõem o jogo, estruturando assim o sistema do jogo e seus elementos, tanto de jogabilidade quanto de narrativa. Com o uso dessa ferramenta foram gerados artefatos que definissem cada um dos aspectos abordados e suas relações, esses artefatos foram elaborados em forma de tabelas que podem ser encontradas logo abaixo. Na Tabela 2, é apresentada todas as mecânicas projetadas para o jogo, bem como suas respectivas descrições.

Tabela 2 - MDA - Mecânicas

MECÂNICAS DO JOGO		
Mecânica	**Descrição**	**Observação**

Níveis	O protótipo do jogo tem 1 nível projetado com as mecânicas mais importantes, que estariam presentes nesse nível em uma versão completa.	
Movimentação	Personagem anda em 4 direções (cima, baixo, esquerda e direita), utilizando as setas do teclado.	
Iniciar uma batalha	Enquanto estiver se movimentando, exceto dentro de áreas projetadas para não iniciar uma batalha, a cada 16 pixels caminhado, é chamado uma função no qual vai definir se uma batalha vai ou não acontecer, tendo um total de 10% de chance de se iniciar uma batalha.	
Exibir menu de batalha	Durante o turno de um personagem jogável, um menu será exibido contendo 5 itens para se navegar, sendo esses: - Ataque - Habilidade - Magia - Item - Fugir	
Exibir vida dos personagens	Durante a tela de batalha, um contador aparece em cima de cada personagem, mostrando sua vida atual e a sua vida máxima.	
Usar ataque	Durante o turno de um personagem jogável, usando a tecla *enter* no item *Ataque*, o personagem do turno atual tentará atacar o inimigo.	
Fugir da batalha	Durante o turno de um personagem jogável, usando a tecla *enter* no item Fugir, os personagens tentaram fugir da batalha atual, tendo uma chance de 10% para escapar da batalha e voltar para o mapa. Caso não se tenha sucesso, o turno será passado para o próximo personagem vivo.	
Tentar ataque	Se a (velocidade do atacante > velocidade do alvo*2) o atacante tem 15% de chance de errar o ataque, caso contrário ele terá 30% de chance de errar o ataque. Caso o ataque não erre, um dano será aplicado na vida do alvo.	

Receber dano	O dano que o alvo receberá será equivalente a (ataque do atacante - defesa do alvo). Caso o ataque do atacante seja menor que a defesa do alvo, o dano aplicado será equivalente a 0.	
Abrir um baú	Usando a tecla *espaço*, um item de ID correspondente ao número armazenado no objeto baú será adicionado ao inventário.	É preciso estar ao lado de um baú e olhando diretamente para o mesmo para que se obtenha o item do baú.
Abrir uma porta	Usando a tecla *espaço*, uma chave será descontada do contador de chaves para abrir a porta. Caso não tenha nenhuma chave, nenhuma interação vai acontecer.	É preciso estar ao lado de uma porta e olhando diretamente para o mesmo para que se tente abrir.
Abrir menu de inventário	Usando a tecla *enter*, exceto durante uma batalha, um menu será mostrado contendo as informações estatísticas dos personagens, além de 4 itens para se navegar, sendo esses: - Arma - Amuleto - Item - Sair	O Inventário também não é aberto caso você esteja se movendo.
Sair do Inventário	Usando a tecla *enter* no item *Sair* no menu de inventário, o menu vai desaparecer e o controle voltará para a movimentação do personagem. *backspace* enquanto estiver navegando dentro dos primeiros itens,	Também é possível sair do inventário usando a tecla *backspace* caso o jogador esteja com a navegação dentro dos itens *Arma*, *Amuleto*, *Item* e *Sair*.
Visualizar lista de itens consumíveis	Usando a tecla *enter* no item *Item* no menu de inventário, uma lista com os itens que o jogador possui será mostrada, contendo seus nomes, quantidades e suas respectivas descrições.	
Sair da lista de itens consumíveis	Usando a tecla *backspace* caso a navegação esteja dentro da lista de itens consumíveis, a lista de itens consumíveis vai desaparecer e o controle voltará a navegação para os itens do inventário.	
Exibir chaves	Na tela do mapa e no inventário, um ícone de uma chave dourada estará visível e ao	

	seu lado terá um contador que mostra a quantidade de chaves que o jogador possui.	

Fonte: Autoria própria (2022)

Por fim, a Tabela 3 descreve a estética e dinâmicas aplicadas, bem como suas relações, finalizando assim os aspectos definidos pela metodologia.

Tabela 3 - MDA - Estética de Dinâmicas

	Estética	**Dinâmicas**
1	Contemplação e descoberta	**Conversão e aprendizagem:** explorando o cenário e obtendo itens o jogador vai aprendendo e descobrindo as mecânicas do jogo, como um tutorial mas sem informar ao jogador explicitamente.
2	Descoberta	A exploração gera a descoberta de trechos da estrutura do mapa/cenário além de localização de baús e portas.
3	Recompensa	Explorar o cenário dá mais recursos ao jogador como itens e chaves que facilitam e ajudam durante o percurso do jogo. Se tratando de um protótipo possíveis recompensas com relação a ganhar uma batalha estão ausentes.
4	Tensão	**Possíveis danos:** durante a batalha, quando a vida de um personagem é zerada ele se torna incapacitado, diminuindo as chances de derrotar o inimigo. Se tratando de um protótipo possíveis penalidades com relação a perder batalhas e zerar a vida estão ausentes.

Fonte: Autoria própria (2022)

3.2 LEVANTAMENTO DE REQUISITOS

3.2.1 Requisitos funcionais

Os requisitos funcionais são todas e quaisquer tarefas a serem executadas pelo sistema, definindo assim as funcionalidades a serem desempenhadas pelo mesmo. Os requisitos funcionais que foram projetados e elaborados para este sistema estão presentes na Tabela 4, bem como suas descrições e prioridades.

Tabela 4 - Requisitos funcionais

Código	Nome	Descrição	Prioridade
F01	Iniciar jogo	Inicia o jogo atualizando variáveis globais necessárias para o funcionamento do mesmo.	Alta
F02	Alterar volume de música de fundo	Aumenta ou diminui o volume da música de fundo no jogo.	Média
F03	Alterar volume de efeitos sonoros	Aumenta ou diminui o volume dos efeitos sonoros no jogo.	Média
F04	Alterar resolução	Troca entre tela cheia ou janela.	Média
F05	Mover personagem	Movimenta o personagem nos eixos X e Y do cenário, aplicando também uma lógica para não colidir com sólidos.	Alta
F06	Iniciar Batalha	A cada 16 pixels caminhados pelo personagem, a funcionalidade é chamada para decidir se uma batalha vai se iniciar.	Alta
F07	Exibir menu de batalha	Um menu é gerado na tela de batalha para que o jogador possa navegar no mesmo e escolher suas ações.	Média
F08	Exibir vida dos personagens	Um contador é posto em cima de cada personagem, mostrando os valores de suas vidas	Alta
F09	Usar ataque	Ao selecionar a opção *Ataque* do menu de batalha, é chamada uma função para checar se o ataque vai ou não falhar.	Alta
F10	Fugir da batalha	Ao selecionar a opção *Fugir* do menu de batalha, é chamada uma função para gerar um valor aleatório entre 0 a 9, caso o valor seja 0 o sistema volta para a o mapa da fase, caso contrário o turno da batalha é passado para o próximo personagem vivo.	Média
F11	Tentar Ataque	Recebendo os valores do atributo *spd* dos personagens envolvidos na batalha é checado se o atributo do atacante é maior que o dobro do atributo do alvo, ao depender do resultado o ataque pode ter entre 15% a 30% de chance de errar, assim não aplicando dano ao alvo.	Alta

F12	Receber dano	Dano é aplicado à vida do alvo, sendo este uma subtração entre o atributo *atk* do atacante e o atributo *def* do alvo.	Alta
F13	Abrir um baú	Ao interagir com um baú, o número do item, armazenado dentro do objeto, será adicionado a um vetor, sendo este a lista de itens do jogador.	Alta
F14	Abrir uma porta	Ao interagir com uma porta, uma chave será subtraída para que o objeto seja destruído e o caminho fique livre para se locomover.	Média
F15	Abrir menu de inventário	É gerado e exibido um menu para navegação, além de informações dos valores de cada atributo dos personagens.	Alta
F16	Sair do inventário	Ao selecionar a opção *Sair* do menu de inventário, ou ao utilizar a tecla backspace, o menu é destruído e a navegação volta para o personagem.	Alta
F17	Visualizar lista de itens consumíveis	Ao selecionar a opção *Item* do menu de inventário, uma lista é exibida contendo os itens que o jogador já obteve, com seus respectivos nomes, descrições e quantidade.	Alta
F18	Sair da lista de itens consumíveis	Ao utilizar a tecla backspace na navegação dentro da lista de itens consumíveis, a lista é destruída e a navegação volta para o menu de inventário.	Alta
F19	Exibir chaves	Um contador é exibido na tela do mapa e no inventário, mostrando a quantidade de chaves que o jogador possui.	Média

Fonte: Autoria própria (2022)

3.2.2 Requisitos não funcionais

Os requisitos não funcionais são todas as necessidades que o software precisa e que não podem ser obtidas através de funcionalidades, não estando assim diretamente relacionados às funcionalidades do sistema. Os requisitos não funcionais que foram listados para este sistema estão presentes na Tabela 5, bem como suas descrições e prioridades.

Tabela 5 - Requisitos não funcionais

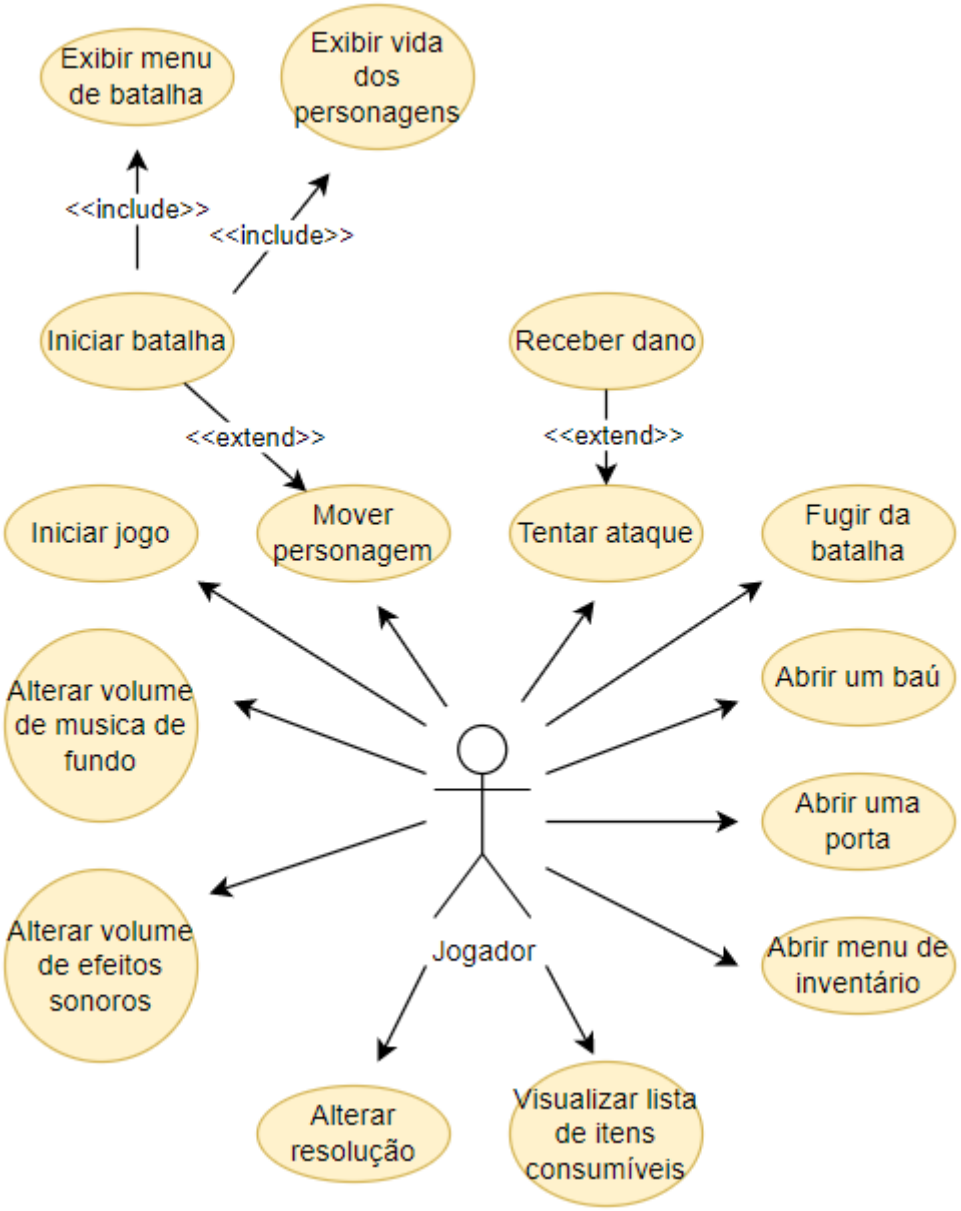
Código	Nome	Descrição	Prioridade
NF01	Baixo tempo de resposta	Por ser um sistema offline e pelo programa ser pequeno, o tempo de resposta em execução se torna baixo	Alta
NF02	Segurança	Por ser um sistema fechado e totalmente offline, não oferece riscos ao usuário.	Alta

Fonte: Autoria própria (2022)

3.3 MODELO DE CASOS DE USO

O diagrama de casos de uso tem o objetivo de detalhar as interações entre os atores do sistema e as funcionalidades. O sistema conta apenas com um único ator, sendo este o próprio jogador. Na Figura 1 é possível visualizar de forma ampla o ator, as interações do mesmo com as funcionalidades do sistema e as interações entre funcionalidades.

Figura 1 - Casos de uso



Fonte: Autoria própria (2022)

É possível notar que a quantidade de ações que o ator pode realizar no sistema é limitada somente às interações que lhe são permitidas. Todas as interações presentes são realizadas pelo próprio jogador, exceto as que apresentam o texto *include* e *extend* nas suas setas. Estas são realizadas pelo próprio sistema de forma automática. As interações que são *extend*, são funcionalidades que podem ser realizadas juntamente com a funcionalidade que as inclui, já nas interações que são *include*, são funcionalidades obrigatórias realizadas pelo próprio sistema e vão sempre serem realizadas juntamente com a funcionalidade que as estende.

3.4 ESPECIFICAÇÕES DO JOGO

O jogo tem como objetivo principal aplicar os conhecimentos adquiridos no curso e conhecimentos específicos da área de jogos para desenvolver um protótipo de alta fidelidade de uma fase do jogo e suas principais mecânicas, que foram projetada com uso do framework MDA, além de demonstrar e servir como um exemplo de aplicação desenvolvida com o foco em jogos. Possibilitando ajudar pessoas interessadas no processo de desenvolvimento de jogos, aprendendo sobre conceitos e como são feitos.

O protótipo do jogo, que se chama *A Role Play Adventure*, foi desenvolvido para desktop, sendo assim executado em uma plataforma web. Ele tem uma resolução real de 320 x 240 e uma visão de câmera em terceira pessoa com vista de cima para baixo em relação ao cenário, além de conter um estilo artístico 2D, sendo feito com uso de *pixel art*, e ser baseado nos jogos de gênero RPG.

Expondo um pouco sobre o contexto da narrativa, o jogo tem como personagens principais Serima e Veras, dois guerreiros que são invocados para um mundo de fantasia, que tem que explorar o novo mundo, tendo que enfrentar monstros e criaturas místicas, descobrir o que está acontecendo e finalizar a jornada nesse mundo de fantasia derrotando os seres malignos, para que assim os protagonistas possam voltar para os seus mundos de origem.

O jogador terá que aprender as regras do jogo, coletar itens, explorar o mapa e seguir a história para poder se manter no jogo, para derrotar os monstros e finalizá-lo. Para isso, ele terá o auxílio de informações que são passadas através de uma HUD[12], sendo todo e quaisquer tipos de informações apresentadas ao jogador de forma gráfica e de fácil visualização.

3.4.1 Personagens

O jogo contém três tipos de personagens, sendo o primeiro deles um jogável controlado pelo jogador para se locomover no cenário, esse personagem está presente apenas na tela do mapa do jogo, podendo interagir com objetos do mesmo.

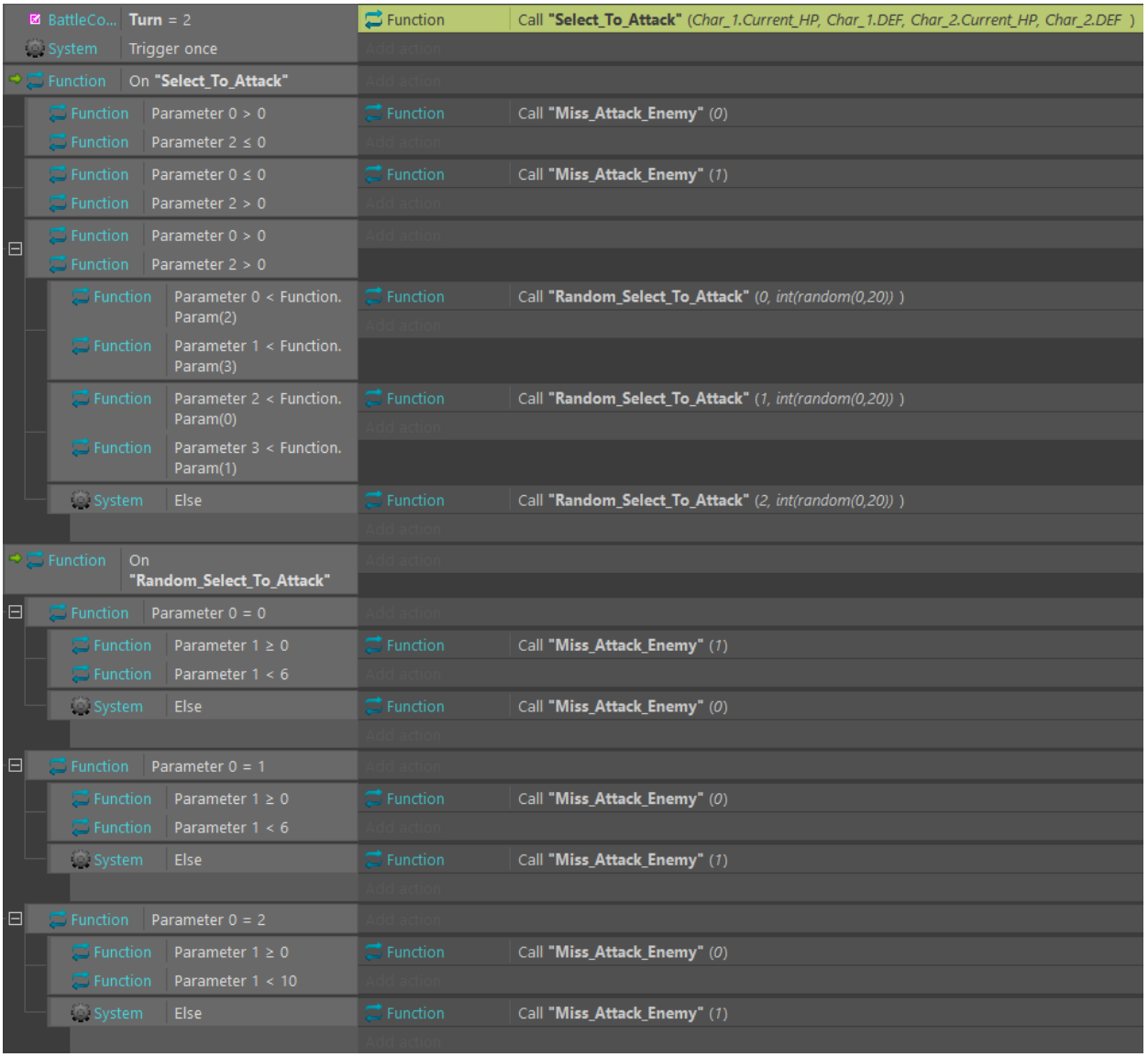
[12] De acordo com Novak (2017, p.246): “As interfaces exibidas na tela incluem painéis de informações transparentes (heads-up displays ou HUDs), que sobrepõem a interface a toda a área de ação do game, e delimitadores, que exibem a interface em uma área menor da tela (geralmente em um dos cantos)”.

O segundo tipo de personagem é encontrado na tela de batalha e suas ações podem ser controladas pelo jogador selecionando opções por meio do menu. Nessa mesma tela o jogador é capaz de controlar as ações de dois personagens dessa categoria, no qual cada um é um objeto contendo atributos, sendo esses *hp* (*health points)*, *mp* (*magic points*), *atk* (*attack*), *def* (*defense*), *spd* (*speed*), *current_hp* e *current_mp*, que são utilizados nas etapas da batalha.

O terceiro e último tipo de personagem é um inimigo que também é encontrado na tela de batalha, porém diferente do tipo anterior, este é controlado por uma IA. Existem 3 tipos diferentes de inimigos e suas respectivas IA foram desenvolvidas baseada nos mesmos tipos usados em jogos retrôs, com uso de aleatoriedade e um certo controle direto nas ações da máquina. Um número é gerado aleatoriamente e dependendo do seu valor ações fixas são realizadas pela máquina, como demonstrado na Figura 2, no qual ao se iniciar o turno do inimigo, a função **Select_To_Attack,** que é usada para a máquina escolher qual alvo do seu ataque, é executada, sendo passado como parâmetros os valores referentes à quantidade de vida dos dois alvos e seus respectivos atributos de *def*.

A máquina apenas é capaz de escolher aleatoriamente caso ambos os dois alvos estejam vivos, tendo mais chances de escolher o alvo mais vulnerável, sendo este o que tem menores valores nos respectivos atributos *current_hp* e *def*. Essa escolha é feita com um número aleatório no qual é gerado por meio de uma função nativa da ferramenta, o *random*, esta função gera um número aleatório dentro de uma faixa que é definida por meio dos parâmetros fornecidos a ela. Ao gerar esse número aleatório, é feita uma comparação do valor gerado, essa comparação vai definir se a máquina irá atacar ou não o personagem mais vulnerável. Após isso é chamada a função **Miss_Attack_Enemy**, passando a ela um valor correspondente ao alvo para ser atacado, esta função vai lidar com as chances de se errar o ataque e os cálculos de dano.

Figura 2 - Aplicação da IA



Fonte: Autoria própria (2022)

Esse inimigo também é um objeto com atributos, porém diferente do anterior ele não detém os atributos *mp* e *current_mp*. Existem 3 tipos de inimigos no qual esse objeto pode ser configurado. No começo da batalha um tipo de inimigo é selecionado aleatoriamente e os valores para os atributos do objeto são inseridos a depender de qual o tipo foi escolhido.

3.4.2 Primeira fase

A primeira fase do jogo é dentro de um templo, no qual o jogador inicia o jogo, sendo uma área curta e introdutório servindo como um tutorial para se aprender sobre o jogo e suas funcionalidades.

Dentro desse mapa contém alguns objetos usados em interação e em regras lógicas do próprio jogo, sendo esses o próprio personagem, um mapa invisível usado como sólido para controlar a movimentação do personagem, um objeto invisível que habilita ou desabilita uma variável global chamada *battle*, após colidir com o personagem, na qual é utilizada para definir se uma batalha vai ou não iniciar, variando entre os valores 0 e 1, e dois objetos interativos, um baú contendo um atributo indicando o item dentro do mesmo, e uma porta sendo um objeto sólido parando alguns caminhos do jogador.

Para se iniciar uma batalha, sempre que o personagem se move 16 pixels, uma função é chamada, essa função checa se a batalha vai ou não acontecer dependendo do valor dentro de uma variável global chamada *battle*, caso o valor seja 1, é gerado um número aleatório entre 1 e 10, se o valor obtido for 1, o jogo abre a tela de batalha.

Nas interações com os objetos, ao interagir com o baú é checado se ele já foi aberto ou não, usando um atributo dentro do mesmo chamado *status*. Caso ainda não tenha sido aberto, uma função é chamada passando como parâmetro o número do item armazenado no objeto dentro do seu atributo *item.* Esse número será usado na lista de itens do jogador, sendo este um array contendo o número do item e a quantidade do mesmo.

Já na interação com a porta, é checado a quantidade de chaves que o jogador possui, estando essa armazenada em uma variável global *key*, caso este valor seja maior que 0, o objeto *porta* no qual o jogador está interagindo será destruído.

3.4.3 Jogabilidade

Usando um teclado seguindo as normas da ABNT, o jogador utiliza as setas para se locomover no cenário e para navegar nos menus. Com o uso da tecla enter e a tecla *backspace*, o jogador pode respectivamente confirmar e voltar para itens

anteriores durante a navegação nos menus. Durante a locomoção no cenário, o jogador pode usar das teclas *enter* e *space* para, respectivamente, abrir o menu de inventário e abrir baús ou portas.

4 RESULTADOS

Neste capítulo é apresentado as interfaces desenvolvidas para o sistema, bem como seu fluxo de navegação e os personagens elaborados para o jogo. O executável foi gerado para desktop, sendo para sistema operacional windows 64 bits e executado pelo arquivo com nomenclatura “*nw.exe*”, ele pode ser encontrado e baixado no seguinte endereço: <*https://yuridasilva.itch.io/aroleplayadventure/*>.

4.1 INTERFACE GRÁFICA

4.1.1 Fluxo de telas

Na **tela de título** o jogador pode tanto seguir para a **tela do mapa,** onde ele será mandado para a primeira fase ou para abrir a **tela de opções,** para mudar as configurações do sistema. Durante a **tela do mapa,** o jogador terá acesso a **tela do inventário,** através de uma tecla, e a **tela da batalha**, através de um evento controlado pelo próprio sistema. A Figura 3, que se apresenta abaixo, demonstra esse fluxo das telas para uma melhor visualização.

Figura 3 - Fluxo de telas

Fonte: Autoria própria (2022)

4.1.2 Tela de título

Na **tela de título**, demonstrada na Figura 4, é apresentado quatro itens para navegação, podendo assim o jogador iniciar um novo jogo, continuar algum jogo salvo, alterar opções de configuração do jogo e fechar o jogo. Sendo que apenas o item **continuar** não se apresenta funcional nesta versão do protótipo.

Figura 4 - Tela de título

Fonte: Autoria própria (2022)

4.1.3 Tela de opções

Na **tela de opções**, demonstrada na Figura 5, são apresentados quatro itens para se navegar, podendo assim o jogador alterar a resolução de janela para tela cheia, alterar o volume de efeitos sonoros, alterar o volume de músicas de fundo e sair da **tela de opções**.

Figura 5 - Tela de opções

Fonte: Autoria própria (2022)

4.1.4 Tela do mapa

Na **tela do mapa**, demonstrada na Figura 6, é onde uma parte do jogo se passa, nessa tela o jogador pode locomover-se pelo mapa e interagir com os objetos do cenário.

Figura 6 - Tela do mapa

Fonte: Autoria própria (2022)

4.1.5 Tela do inventário

Na **tela do inventário**, demonstrada na Figura 7, é apresentado 4 itens para se navegar, podendo assim o jogador abrir lista de armas, abrir lista de amuletos, abrir uma lista de itens consumíveis e sair da tela do inventário, sendo que apenas os itens **item** e **sair** se apresentam funcionais neste protótipo. Nela também é possível visualizar os atributos dos personagens e seus valores, bem como suas respectivas quantidades de vida.

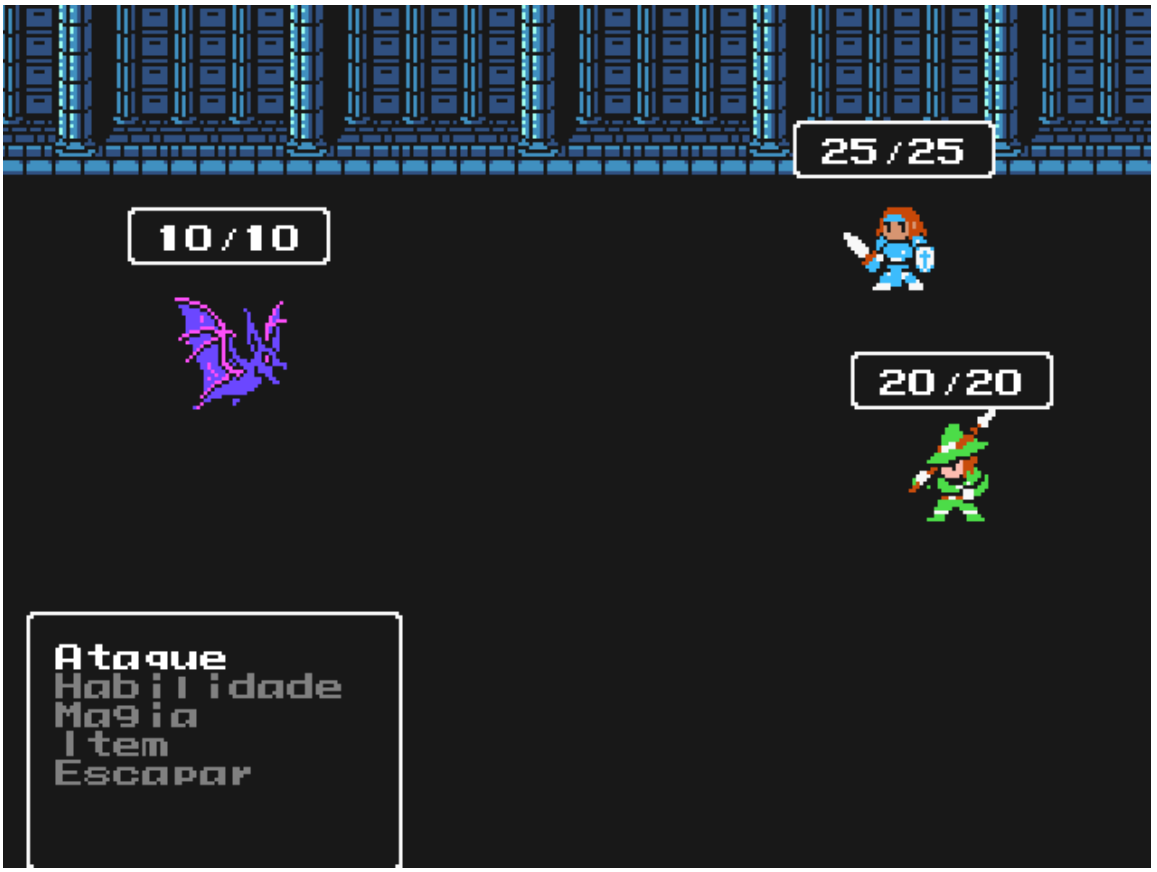
Figura 7 - Tela do inventário

Fonte: Autoria própria (2022)

4.1.6 Tela da batalha

Na **tela de batalha**, demonstrada na Figura 8, é onde acontece a luta entre os inimigos e os personagens jogáveis. Nela é apresentado cinco itens para selecionar durante a batalha, podendo assim o jogador atacar, usar uma habilidade, usar uma magia, usar um item consumível e tentar sair da batalha, sendo que apenas os itens **atacar** e **fugir** se apresentam funcionais neste protótipo.

Figura 8 - Tela da batalha

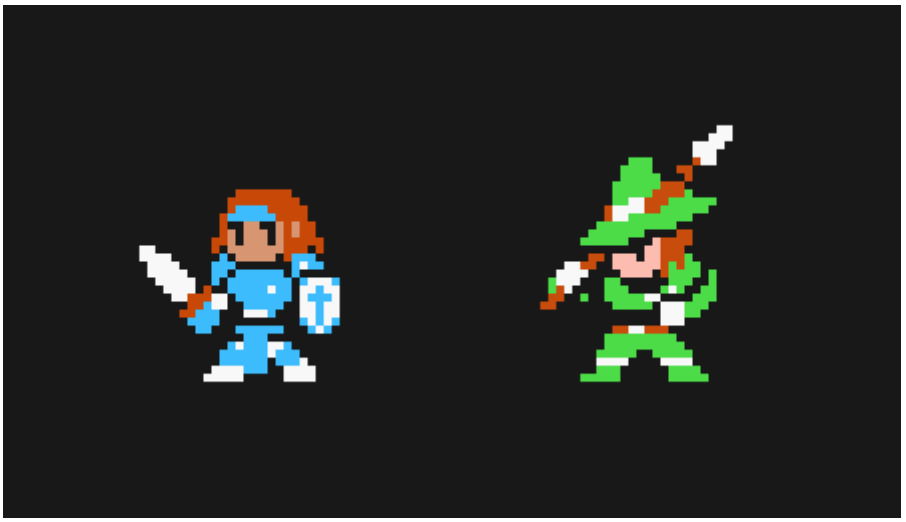


Fonte: Autoria própria (2022)

4.2 PERSONAGENS

Dos personagens que foram desenvolvidos, em termos de game design, dois são os protagonistas, sendo esses Serima, uma cavaleira, focada em maior dano e defesa para resistir e derrotar inimigos mais rapidamente, e Veras, um lanceiro com menor dano, mas maior velocidade, fazendo assim que tenha menos chance de errar seus ataques e maior chances de se esquivar de ataques de inimigos. Esses dois personagens podem ser visualizados na Figura 9.

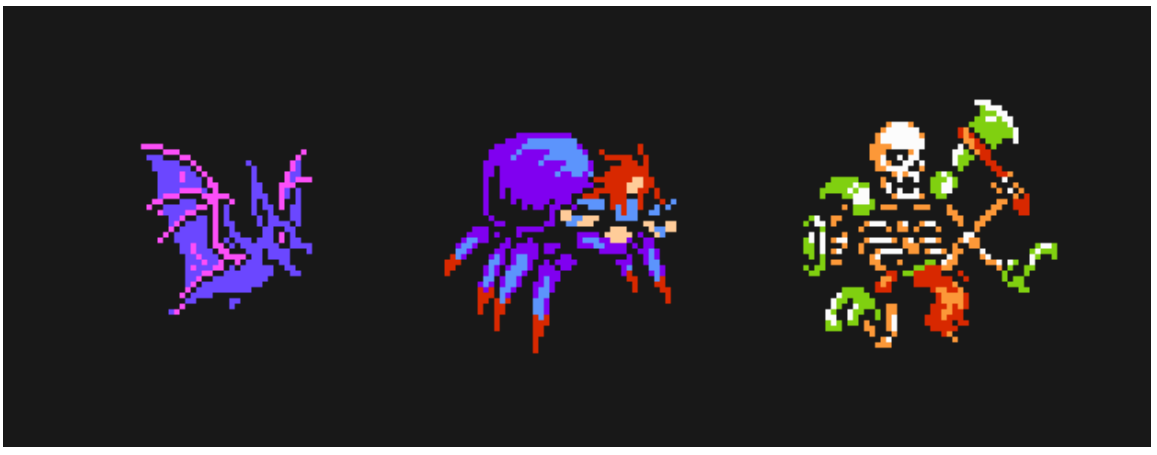
Figura 9 - Personagens



Fonte: Autoria própria (2022)

Já os inimigos foram balanceados em termos de dificuldade, além de seus atributos terem valores diferentes um do outro, suas aparências também diferem, a fim de destacar cada um dos inimigos. Os inimigos desenvolvidos neste protótipo podem ser visualizados na Figura 10.

Figura 10 - Inimigos



Fonte: Autoria própria (2022)

5 CONCLUSÃO

O desenvolvimento deste projeto permitiu uma melhor visualização e aplicação de conceitos das áreas de análise e desenvolvimento de sistemas voltadas à elaboração de jogos digitais, uma vez que um jogo é caracterizado como um tipo de software e sua elaboração contempla as etapas de desenvolvimento de softwares.

Para elaborar esse software, foram utilizados conceitos da área de jogos, como game design, design de níveis e IA, além de conceitos de conhecimento da área de desenvolvimento de softwares abordados na graduação, sendo esses conhecimentos em algoritmos, arquitetura, teste de software e também conhecimentos sobre programação, principalmente programação orientada a objetos. Conhecimentos os quais são de grande relevância e possibilitam a construção de softwares, sua manutenção e qualidade.

Com a conclusão do trabalho, o sistema serve como uma boa experiência para desenvolvimento em áreas mais voltadas a jogos e pode possibilitar até mesmo a gamificação de sistemas. A partir das funcionalidades levantadas no início do projeto, foi possível atingir os objetivos definidos anteriormente, no capítulo 1. Os resultados obtidos caracterizam um protótipo inicial jogável, que é passível de melhorias e continuidade em versões futuras, já que o sistema se trata inicialmente de um protótipo. Para versões futuras o sistema poderia contemplar outros níveis e um melhor polimento nas mecânicas, bem como suas funcionalidades, tornando assim um jogo mais completo e próximo de sistemas que são lançados nessa área.

A fim de fechar o ciclo de desenvolvimento de software, a inclusão da validação do software desenvolvido poderia ser contemplada como um trabalho futuro, já que trata-se de uma etapa importante em um ciclo de desenvolvimento de software. Permitir que o jogo seja testado por pessoas que possam avaliá-lo tecnicamente do ponto de vista de um software e de um jogo é algo desejável, porém não é um impeditivo para alcançar o objetivo deste trabalho, que é entregar um protótipo jogável em alta fidelidade de uma fase de um jogo digital.

REFERÊNCIAS

BRINKMANN, Martin. **A Look at Tiled - Tiled map editor for GNU/Linux**. GHacks Technology News. Disponível em: <https://www.ghacks.net/2018/03/04/tiled-rpg-map-editor-linux/>. Acesso em: 6 mar. 2022.

CONSTRUCT 2 ONLINE MANUAL & DOCUMENTATION. SCIRRA LTD. Disponível em: <https://www.construct.net/en/construct-2/manuals/construct-2/>. Acesso em: 10 mar. 2022.

CUPERSCHMID, Ana Regina Mizrahy; HILDEBRAND, Hermes Renato. **Avaliação Heurística de Jogabilidade**: Counter-Strike: Global Offensive. SBGAMES. Disponível em: <http://www.sbgames.org/sbgames2013/proceedings/artedesign/44-dt-paper.pdf/>. Acesso em: 9 mar. 2022.

FULLERTON, Tracy. **Game Design Workshop**: a playcentric approach to creating innovative games. 3. ed. New York: A K Peters/CRC Press, 2014.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 6. ed. São Paulo: Atlas, 2019.

HUIZINGA, Johan. **Homo Ludens**: Study of the Play Element in Culture. New Edition. London: Routledge, 1980.

HUNICKE, Robin; LEBLANC, Marc; ZUBEK, Robert. **MDA**: A Formal Approach to Game Design and Game Research. Disponível em: <https://users.cs.northwestern.edu/~hunicke/pubs/MDA.pdf>. Acesso em: 4 mar. 2022.

INTRODUCTION - TILED DOCUMENTATION. Disponível em: <https://doc.mapeditor.org/en/stable/manual/introduction/>. Acesso em: 10 mar. 2022.

KAPLAN, Andreas; HAENLEIN, Michael. **Siri, Siri, in my hand**: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. Business Horizons, 2018.

LEI DO BEM: OS INCENTIVOS FISCAIS À INOVAÇÃO TECNOLÓGICA. ABGI. Disponível em: <https://brasil.abgi-group.com/lei-do-bem/>. Acesso em: 26 jan. 2022.

MEDEIROS, Tainá Jesus; SILVA, Thiago Reis da; ARANHA, Eduardo Henrique da Silva. **Ensino de programação utilizando jogos digitais**: Uma revisão sistemática da literatura. 2013. 10f. TCC (Graduação) - Curso de TI, UFRN, Natal, 2013.

MIRANDA, Frederico S; STADZISZ, Paulo C. **Jogo Digital: definição do termo**. SBC - Proceedings of SBGames 2017. Disponível em:

<https://www.sbgames.org/sbgames2017/papers/ArtesDesignShort/173500.pdf/>. Acesso em: 4 mar. 2022.

NOVAK, Jeannie. **Desenvolvimento de Games.** Tradução da 2ª edição norte-americana. São Paulo: Cengage Learning, 2017.

RABIN, Steve. **AI Game Programming Wisdom**. Massachusetts: Charles River Media, 2002.

SALEN, Katie; ZIMMERMAN, Eric. **Rules of Play: Game Design Fundamentals**. The Mit Press, 2003.

SALES, Matheus. **RPG (Role-Playing Game)**. Brasil Escola. Disponível em: <https://brasilescola.uol.com.br/curiosidades/rpg.htm/>. Acesso em: 4 mar. 2022.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum**. **Um guia definitivo para o Scrum: As regras do jogo**. 2013. Disponível em: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 4 mar. 2022.