



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Projeto Orientado em Computação 2: Relatório final

Aluno: Matheus Farnese Lacerda Senna

Orientador: Lucas Nascimento Ferreira

Tipo de pesquisa: Tecnológica

Data: Dezembro 2025

“A Lenda dos Guardiões”: criação de um jogo RPG indie

Matheus Farnese Lacerda Senna, Lucas Nascimento Ferreira

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brasil

senna.matheus@dcc.ufmg.br, lferreira@dcc.ufmg.br

Resumo. *Nesse trabalho foi implementado um jogo digital 3D estilo RPG utilizando a engine Unity. O jogo é pequeno e simples, mas possui mecânicas coesas e uma progressão bem definida, contendo 3 missões sequenciais, sendo a primeira apresentada ao jogador no início do jogo e as outras após a conclusão da anterior. O jogo possui uma narrativa completa com começo, meio e fim e foi projetado com o objetivo de entreter o jogador por meio de uma narrativa simples com personagens carismáticas e de uma gameplay fluída e intuitiva.*

Palavras-chave— Unity, RPG, jogos 3D, indie, lógica de componentes

1. Introdução

Jogos sempre estiveram presentes na humanidade, sendo um dos mais antigos que se tem registro datando de 3000 a.C. Jogos estão no dia a dia de diversas pessoas, independentemente da classe social, e são utilizados em diversos contextos: competições, demonstrações de inteligência, apostas ou simplesmente como *hobbys*. Em tempos recentes, a partir das décadas de 1950 e 1960 com a criação dos computadores, o ramo de jogos se expandiu para o mundo virtual e, desde então, vem sendo uma área em rápida evolução, com jogos cada vez mais complexos, acompanhando a evolução dos computadores e algoritmos. A indústria de jogos é extensa e seus produtos afetam milhões de pessoas todos os dias, não apenas com a diversão que os jogos em si proporcionam, mas também indiretamente, fomentando um comércio de mercadorias associadas e promovendo interações em redes sociais.

A escolha do desenvolvimento de um jogo como o tema do presente trabalho é justificada não só por trabalhar áreas estudadas durante o curso (e.g. computação gráfica, autômatos, otimização, dentre outras) mas também por exercitar a habilidade de criação de algo interativo e que almeja entreter o usuário final. O entretenimento como um objetivo é algo incomum em trabalhos acadêmicos, sendo que jogos certamente se destacam por sua capacidade de combinar teoria e prática de maneira envolvente. Ao desenvolver um jogo, é possível aplicar conceitos técnicos de forma criativa e, ao mesmo tempo, entender como as tecnologias podem ser utilizadas para proporcionar uma experiência interativa que, além de desafiadora e envolvente, também estimula o raciocínio lógico e estratégico do jogador. Por fim, a escolha por esse tema visa não só aprofundar o conhecimento adquirido ao longo do curso, mas também contribuir para a área de desenvolvimento de jogos, que tem se mostrado como uma indústria crescente tanto no Brasil quanto no exterior.

Dessa forma, “A Lenda dos Guardiões” é um jogo estilo RPG que foi desenvolvido ao longo do semestre, sendo a narrativa e as personagens do jogo baseadas em um livro de fantasia e aventura de mesmo nome [1]. O jogo é pequeno e simples, sendo possível completá-lo em menos de 1 hora. Não obstante, as mecânicas são coesas e há um sistema de progressão claro, com uma narrativa com começo, meio e fim e missões a serem cumpridas pelo jogador. Ademais, o jogo foi feito de tal forma que permite a sua expansão no futuro, tal como a criação de novas missões e de novas personagens jogáveis, mantendo a mesma lógica de projeto e a mesma ambientação.

O restante desse relatório está organizado da seguinte forma: primeiramente serão dadas algumas definições de termos específicos de jogos que serão necessárias para a compreensão do restante do relatório. Em seguida, será discutido brevemente o contexto atual da indústria de jogos no Brasil e o ferramental utilizado no trabalho. Após isso será explicado o funcionamento do jogo, primeiramente sob a perspectiva da *gameplay* (explicando a progressão do jogo) e depois sob a perspectiva da implementação (explicando algumas decisões de projeto e lógicas utilizadas). Por último, encerra-se o relatório detalhando algumas expansões planejadas para serem acrescentadas ao jogo em um futuro próximo.

2. Definições

“O que é um jogo?” Várias respostas já foram dadas a essa pergunta, desde frases simples e genéricas como a clássica frase de Sid Meier “*a game is a series of interesting choices*” até definições mais complexas, como a presente no livro de Raph Koster [2] que, traduzida para o português, lê-se como: “Um jogo é uma experiência interativa que apresenta ao jogador uma sequência de padrões e ações progressivamente mais difíceis que ele(a) aprende e, eventualmente, domina”. Em A Lenda dos Guardiões, o jogador interage com o jogo ao controlar as ações da personagem principal, utilizando mecânicas pré definidas e obtendo itens cada vez melhores ao longo do jogo para derrotar inimigos progressivamente mais fortes. “Mecânicas”, “inimigo” e “item” são termos específicos do contexto de jogos, cujas definições estão expressas a seguir, juntamente com outros termos que serão utilizados ao longo desse texto.

- Gameplay: É a experiência de jogar o jogo, abrangendo as regras, mecânicas, objetivos e a interação do jogador com o jogo.
- Condições de vitória e derrota: São acontecimentos que encerram o jogo com, respectivamente, a vitória do jogador ou sua derrota.
- Personagens: Entidades do jogo que interagem entre si na narrativa.
- Personagem jogável: Personagem dinâmica controlada pelo jogador para alcançar a condição de vitória.
- Inimigos: Personagens dinâmicas, não controladas pelo jogador e que agem buscando a condição de derrota do jogador.
- Gamepad: Um dispositivo portátil de entrada com botões, direcionais e alavancas, utilizado pelo jogador para interagir com o jogo.
- Mecânicas: Conjunto de ações que podem ser realizadas pelo jogador ao apertar botões do *gamepad*. Cada botão realiza uma ação pré definida de acordo com o estado atual do jogo.
- Skills: Um subconjunto das mecânicas. É um termo utilizado para designar ações que necessitam de mana.
- Mana: Um recurso do jogo. Mana regenera automaticamente ao longo do jogo e é gasta ao utilizar uma *skill*. Diferentes *skills* possuem diferentes custos de mana.
- Stats: Conjunto de valores associados a personagens e inimigos e que definem parâmetros do jogo (e.g. taxa de regeneração de mana). Esses parâmetros afetam a *gameplay*.
- Itens: São obtidos ao longo do jogo e modificam *stats* da personagem jogável, podendo também adicionar alguma mecânica nova. Itens só funcionam se estiverem equipados pela personagem, que pode carregar quantos itens quiser, mas pode equipar apenas uma certa quantidade ao mesmo tempo (no caso desse jogo, 3).

3. Contextualização

Com relação ao estado atual do mercado brasileiro de jogos, de acordo com uma pesquisa da Newzoo [3], que é uma empresa de pesquisa de mercado que fornece dados

e análises sobre a indústria global de *games*, *e-sports* e mídia digital, o Brasil é o terceiro país em termos de números absolutos de consumidores de jogos, atrás apenas dos Estados Unidos e China. Entretanto, apesar do grande mercado consumidor, não há muitas empresas brasileiras produtoras de jogos, como pode-se observar na figura 1, onde o lucro do Brasil com jogos está muito aquém dos outros países. Apesar disso, de acordo com a Abragames [4], a indústria brasileira de jogos cresceu nos últimos anos, mais do que duplicando o número de empresas desenvolvedoras nos últimos 5 anos, sendo a maioria delas presente na região sudeste.





	Market	Revenue (USD)	Players
	China	\$49.8B	723.0M
	United States	\$49.6B	224.8M
	Brazil	\$2.7B	123.3M
	Japan	\$16.8B	74.1M

Figura 1: Extraída da pesquisa da Newzoo, mostrando os países que mais produzem e consomem jogos.

Com relação ao ferramental utilizado para o desenvolvimento de jogos, a Unity [5] é uma *engine* multiplataforma amplamente utilizada tanto por grandes empresas quanto por desenvolvedores de jogos *indie*. De acordo com a pesquisa da Abragames [4], 75% dos desenvolvedores de jogos no Brasil utilizam a Unity. Essa engine possui várias funcionalidades que facilitam a criação de um jogo, tais como modelos de física e de iluminação já implementados, interfaces gráficas para a criação e modificação de objetos na tela, sistemas para executar o jogo e detectar as ações do jogador, dentre várias outras, abstraindo essas camadas para o desenvolvedor e permitindo que ele foque sua atenção na lógica do jogo. Por essas razões, a Unity foi escolhida como a *engine* utilizada nesse trabalho.

4. Gameplay

A Lenda dos Guardiões é um jogo estilo RPG cuja narrativa e história das personagens foi baseada em um livro de mesmo nome [1]. A personagem jogável é a protagonista do livro: Talim, uma órfã que perdeu seus pais durante a infância, foi adotada e treinada pelo Clã dos Assassinos, tornando-se uma lutadora ágil e extremamente competente. O jogador irá assistir a história pela perspectiva da Talim e, durante o combate contra inimigos, irá controlar a personagem por meio de ações como mover e atacar, além de ter acesso a um conjunto de 4 *skills* que podem ser utilizadas para auxiliar no combate. As *skills* também foram baseadas nas habilidades de Talim no livro.

4.1. Fluxo do jogo

O jogo possui 3 missões, sendo que a condição de vitória é cumprir todas as 3. A condição de derrota é morrer a qualquer momento do jogo. A morte se dá quando os pontos de vida (HP) da Talim chegam a zero. HP é um recurso do jogo intrínseco a agentes (Talim e inimigos) e que, quando chega a zero, o respectivo agente morre. Cada agente possui uma quantidade inicial de HP e, a cada ataque sofrido, esse valor é reduzido de acordo com a força do ataque. Inimigos não regeneram HP e Talim regenera HP ao final de cada missão e a depender dos itens equipados. Cada missão possui uma condição de vitória que é derrotar um certo inimigo pré definido.

A *gameplay* é dividida em 4 modos que se intercalam ao longo do jogo: narrativa, base, exploração, combate. O modo narrativa conta a história ao jogador por meio de textos. Para esse modo, foi adotada uma abordagem parecida com uma *visual novel*, onde a narrativa é apresentada ao jogador por meio de diálogos (em forma de texto) entre as personagens e com imagens ao fundo representando a presente cena, como pode ser verificado na figura 2.



Figura 2: Um momento da narrativa, onde Talim conversa com outro personagem em uma floresta.

O modo base é um menu onde o jogador pode comprar itens em uma loja. Cada item custa uma certa quantidade de moedas, sendo que o jogador começa com moedas suficientes para comprar de 2 a 3 itens, a depender do preço, e ganha mais moedas ao derrotar inimigos. Ademais, também é possível verificar as *skills* de Talim e quais são seus efeitos em batalha.

O modo exploração é quando o jogador controla Talim livremente pelo mundo. Cada missão se passa em um mapa diferente, onde o jogador pode movimentar Talim para onde quiser dentro da área da missão. Há também inimigos espalhados pelo mapa, que irão perseguir Talim caso ela se aproxime muito. Ao entrar em contato com um inimigo, o jogo muda para o modo combate.



Figura 3: Menu principal do modo base.



Figura 4: Modo base. À direita a tela explicando as *skills* a que o jogador tem acesso. À esquerda é mostrado uma parte da loja de itens.



Figura 5: Modo exploração. No centro da tela está a Talim. Os inimigos vagueiam pelo mapa e foram circutados na imagem (os círculos foram editados no *screenshot* tirado e não estão presentes no jogo).

O modo combate é quando Talim luta contra inimigos. Nesse modo, a área de movimentação é limitada a um quadrado pequeno e o jogador deve utilizar os ataques e *skills* de Talim para derrotar os inimigos. É nesse modo que o jogador terá que usar suas habilidades manuais e reflexos para apertar botões do *gamepad* e executar uma boa sequência de ações para matar os inimigos e evitar que Talim morra. Ou seja, é nesse modo que o jogo irá desafiar o jogador.

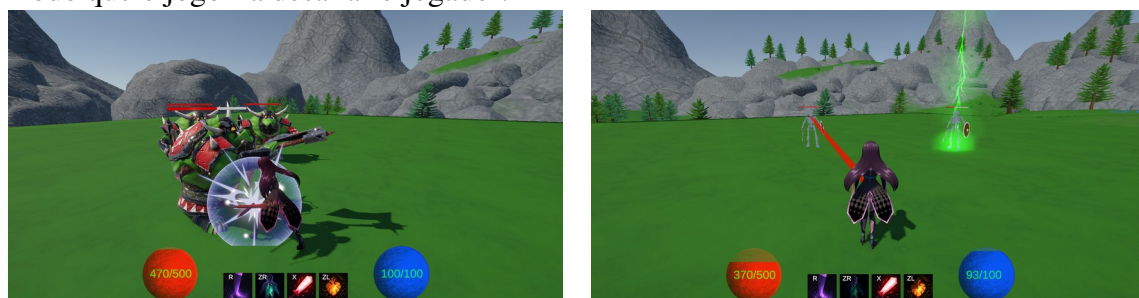


Figura 6: Modo combate. À esquerda é demonstrado um ataque da Talim e à direita a utilização de uma *skill* de dano (o relâmpago verde caindo sobre o esqueleto). Verifica-se também o HUD, que indica os pontos de vida e mana do jogador (respectivamente círculos vermelho e azul) e os ícones das 4 *skills*, mostrando se elas podem ser utilizadas ou se estão recarregando.

Assim sendo, o jogo começa no modo narrativa, contando a história dos primeiros capítulos do livro, dando uma visão geral do mundo e apresentando as personagens que serão importantes para a narrativa. Então, é apresentado ao jogador (sob a perspectiva da Talim) um problema que precisa ser resolvido. Após isso, Talim é instigada a resolvê-lo e o jogo muda para o modo base. Depois de aprender o que suas *skills* fazem e de comprar os itens iniciais, o jogador é então colocado na área da primeira missão e o modo exploração é ativado, dando o controle sobre a movimentação da Talim ao jogador. Ao entrar em contato com um inimigo durante a exploração, o modo combate é ativado. Ao derrotar todos os inimigos, Talim ganha uma certa quantidade de moedas, volta para onde ela estava no mundo e o jogo volta ao modo exploração. O objetivo da missão é matar um inimigo específico. Ao derrotá-lo, o modo narrativa é novamente ativado para progredir a história, onde é apresentado uma nova missão. Novamente o jogador é colocado no modo base, onde pode comprar mais itens e, quando estiver pronto, é então colocado na área da segunda missão. Analogamente, ao derrotar certo inimigo, o jogador é então apresentado à terceira e última missão, que consiste em apenas uma batalha contra o inimigo mais desafiador do jogo. Ao derrotá-lo, o jogador vence o jogo.

4.2. Arte

A arte utilizada foi no estilo JRPG, caracterizado por uma forte influência de anime e de outros elementos da cultura pop japonesa, resultando em gráficos menos focados no realismo e mais na expressividade e estilo. O jogo é 3D, sendo que todos os modelos 3D das personagens (Talim e inimigos) e dos elementos do mundo (e.g. montanhas, árvores, grama, pedras, etc) foram obtidos de fontes gratuitas de *assets* disponíveis na internet com permissão para utilização em projetos não comerciais. As imagens 2D de personagens e *backgrounds* utilizados no modo narrativa foram, em sua maioria, gerados por IA utilizando ferramentas como a *Meshy AI*.

5. Metodologia

Nessa seção serão explicadas a arquitetura, a lógica e as decisões de projeto tomadas durante a criação de A Lenda dos Guardiões. O jogo foi feito utilizando a *engine* Unity, seguindo uma lógica de implementação em componentes, que se baseia no princípio de que os objetos do jogo (*GameObjects*) são meros *containers* para componentes, que por sua vez definem o seu comportamento e funcionalidade. Essencialmente, os objetos são criados por composição em vez de herança de classes rígida. Um objeto é qualquer entidade presente no jogo que influencia o seu comportamento, seja ele visível (e.g. personagem do jogador, inimigos, peças de cenário, etc) ou invisível (e.g. *triggers*, *spawners*, barreiras invisíveis, etc). A seguir serão definidos alguns termos específicos da Unity que serão necessários para o entendimento dessa seção.

- *GameObject*: É a entidade básica que representa qualquer coisa presente no jogo. Um *GameObject* por si só não possui funcionalidades, sendo apenas um *container* vazio que só ganha comportamentos por meio dos componentes adicionados a ele.
- *Objeto*: Sinônimo de *GameObject* e utilizado no texto para evitar repetições.
- *Componente*: Elementos modulares que implementam uma funcionalidade e podem ser anexados a *GameObjects* para prover a ele a funcionalidade implementada.
- *Script*: Um componente criado pelo desenvolvedor do jogo.
- *Cena*: Um conjunto de *GameObjects*. Unity possui comandos para carregar e descarregar cenas, respectivamente instanciando ou destruindo todos os objetos da cena.

Um componente na Unity pode ser definido como uma unidade modular de funcionalidade que, quando associada a um *GameObject*, atribui um comportamento a ele dentro do ambiente de jogo. Existem diversos componentes já implementados fornecidos pela Unity, tais como componentes para identificar a posição e orientação (*Transform*) ou para controlar interações físicas (*Rigidbody*). Além dos componentes já implementados, o desenvolvedor também pode criar os seus próprios componentes por meio de *scripts* escritos em C#, que irão introduzir lógicas específicas do jogo aos objetos. Idealmente, um componente deve respeitar alguns princípios: responsabilidade única, reutilização, flexibilidade e modularidade.

A responsabilidade única dita que o componente deve adicionar apenas um comportamento ao objeto, cumprindo apenas uma função bem definida e evitando acumular lógicas diferentes. Reutilização implica uma escrita do componente de forma a ser possível anexá-lo a diferentes objetos, reduzindo duplicação de código e acelerando o desenvolvimento. Flexibilidade refere-se à capacidade do componente de se adaptar a mudanças no projeto, seja por meio de parâmetros expostos ao programador por meio da interface da Unity ou por uma lógica que permita comportamentos variados sem reescrita de código. Finalmente, a modularidade garante que os componentes funcionem como peças independentes, facilmente combináveis entre si, permitindo que objetos complexos sejam construídos a partir de partes menores e mais simples.

Além disso, Unity permite a criação de diversas cenas para facilitar a organização. Uma cena é um ambiente que reúne e organiza todos os elementos necessários para representar uma parte do jogo. Cada cena contém o conjunto de *GameObjects* e componentes que definem o que acontece naquele trecho do jogo. Cenas podem representar desde menus e telas de interface até fases completas, áreas exploráveis ou ambientes de teste, permitindo que o desenvolvimento seja estruturado de forma modular, organizada e fácil de gerenciar. Nesse projeto, foi criada uma cena para cada um dos 4 modos apresentados na seção 4.1 (narrativa, base, exploração e combate).

5.1. Talim

Em cada uma das 4 cenas diferentes implementadas (referentes aos 4 modos) o jogador interage com o jogo de uma maneira diferente. O modo narrativa é o mais simples, onde os textos de diálogo aparecem na tela e a única ação possível é passar para o próximo texto. Um controle simples foi implementado para controlar essa interação com o jogador. Clicar no botão A do *gamepad* avança o texto, permitindo que o jogador ajuste à sua velocidade de leitura.

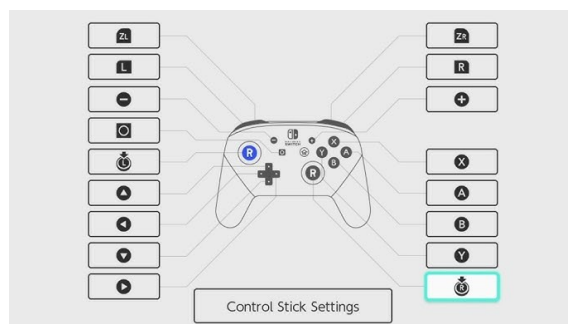


Figura 7: Um exemplo de *gamepad*.

O modo base é um menu sobre o qual o jogador navega utilizando os botões direcionais do *gamepad* para escolher onde clicar, A para confirmar a escolha e B para voltar à seleção anterior. Nesse modo é possível comprar itens e, portanto, há um *GameObject* “inventário” para controlar essa interação. Esse objeto possui um componente que controla a adição e controle (equipar / desequipar) de itens, além de

armazenar a lista de itens que o jogador possui. Esse objeto é instanciado no começo da execução e é sempre mantido ao longo das cenas, pois o inventário pode ser acessado em todas elas.

Para o modo exploração, o jogador precisa ser capaz de movimentar a Talim pelo mundo. Portanto, há um *GameObject* representando a Talim que possui um componente para verificar os *inputs* do *gamepad* (e.g. direcionais para verificar a direção do movimento), outro para lidar com a movimentação em si (e.g. cálculo de velocidades), outro para lidar com a câmera e fazê-la seguir o jogador em 3ª pessoa e um último para lidar com as animações. No modo de exploração a Talim precisa apenas de componentes relacionados à movimentação, pois o combate é feito em outra cena. Além do *GameObject* da Talim, o jogador também tem acesso a um menu de pausa, onde pode equipar e desequipar os itens no seu inventário.

O *GameObject* que representa a Talim no modo combate é o mais complexo dos 4 modos. Além dos componentes de detecção de *inputs*, movimentação e câmeras já citados, são necessários 4 outros componentes para controlar os pontos de vida (HP), a mana, as *skills* e os ataques. Ademais, o componente responsável pelas animações agora também precisa lidar com animações de combate (e.g. ataque, canalização, morte, etc). A figura 8 a seguir ilustra as relações entre os componentes da Talim. Em rosa estão aqueles presentes tanto no modo exploração como no modo combate. Em vermelho estão aqueles presentes apenas no modo combate. O componente de HP é utilizado tanto na Talim quanto nos inimigos, sendo que o componente representado em azul está anexado a um inimigo.

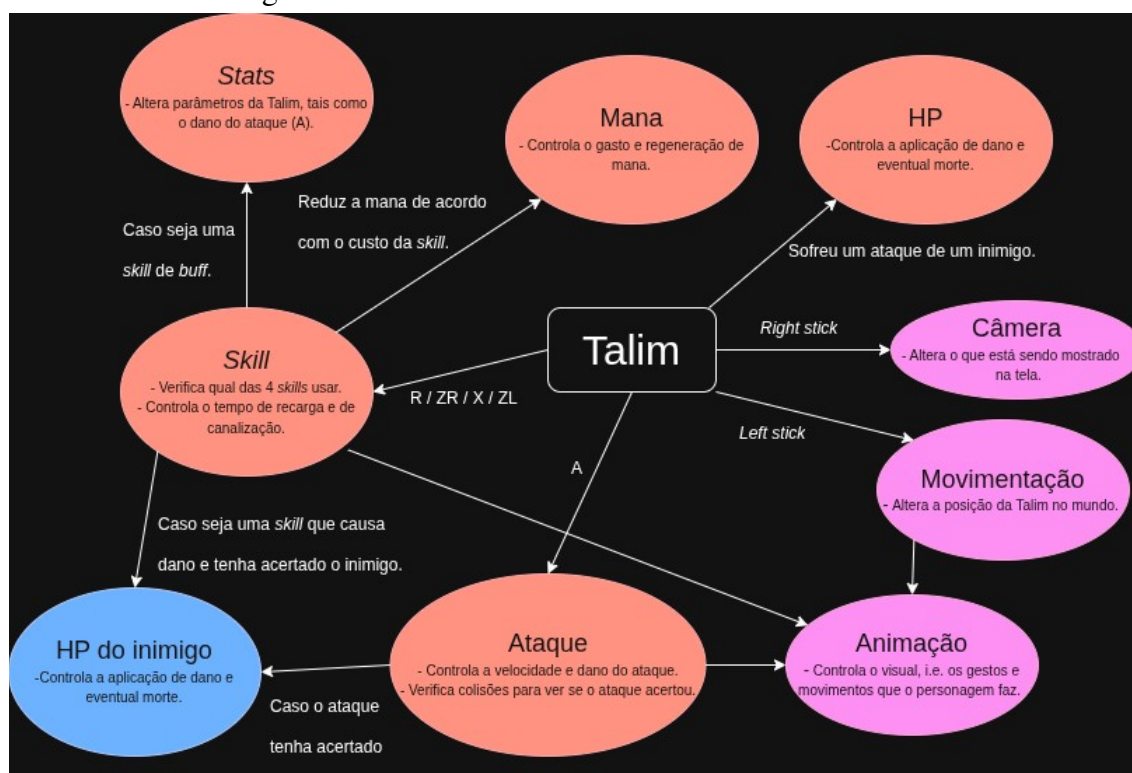


Figura 8: Componentes da Talim com suas respectivas interações.

5.2. Inimigos

Os inimigos foram implementados utilizando uma lógica de máquina de estados. Unity também oferece suporte para esse tipo de lógica, sendo possível criar *scripts* que representam cada estado separadamente e um controlador que decide qual estado está ativo. O controlador chama as funções do estado atual (e.g. *Enter* que faz algo quando

há uma transição para aquele estado, *Update*, que executa o estado a cada *frame* que ele estiver ativo e *Exit* que faz algo ao sair do estado) e muda de estado quando as condições definidas são atendidas. Em outras palavras, cada estado induz um comportamento diferente no inimigo e eventos do jogo (e.g. Talim se aproximando) mudam o estado do inimigo.

Assim como a Talim, os inimigos também funcionam de forma diferente no modo exploração e combate. Dessa forma, foram implementadas duas máquinas de estados, uma para o modo exploração e outra para o modo combate. Para o modo exploração, há 3 estados: ocioso, patrulha e perseguição. O inimigo começa ocioso e, após alguns segundos, começa a patrulhar. Cada inimigo possui uma área de patrulha pré definida no mapa, sobre a qual ele anda aleatoriamente. Após alguns segundos de patrulha, ele volta a ficar ocioso. Esse ciclo se repete infinitamente até que a Talim se aproxime dele. Quando isso ocorre, o inimigo, entra no modo perseguição. Nesse modo, sua velocidade de movimento é aumentada com relação à patrulha e ele corre em direção à Talim. Caso colida com ela, o modo combate é ativado e a batalha começa. Caso a Talim se afaste muito do inimigo, ele retorna à área de patrulha e volta ao ciclo de patrulha e ócio.

Já a máquina de estados para o modo combate possui 6 estados e está ilustrada na figura 9. O inimigo começa parado até que um limite de tempo seja atingido. Após isso, o inimigo escolhe estocasticamente se irá atacar Talim de perto (corpo a corpo) ou de longe (atirando projéteis). A probabilidade dessa escolha varia de inimigo para inimigo. Tendo escolhido o comportamento, o inimigo então parte para um estado onde o jogador consegue entender qual ação está sendo planejada. No caso do ataque corpo a corpo, o inimigo simplesmente se move em direção à Talim, dando ao jogador um certo tempo para pensar no que fazer. No caso do ataque a distância, o inimigo fica parado por um curto período de tempo e aparece uma linha pulsante vermelha na tela (como mostrado na figura 6 da seção 4.1) ligando o inimigo à Talim, mostrando ao jogador que um projétil irá ser atirado em breve naquela direção. Após esses estados, o inimigo de fato realiza a ação de atacar ou atirar. No caso do ataque corpo a corpo, se ele não estiver perto da Talim mesmo após a movimentação, ele simplesmente retorna ao estado ocioso. Finalmente, a qualquer momento, caso os pontos de vida do inimigo cheguem a zero, ele entra no estado “morrer”, onde é mostrada uma animação de morte e, após isso, ele é retirado do jogo.

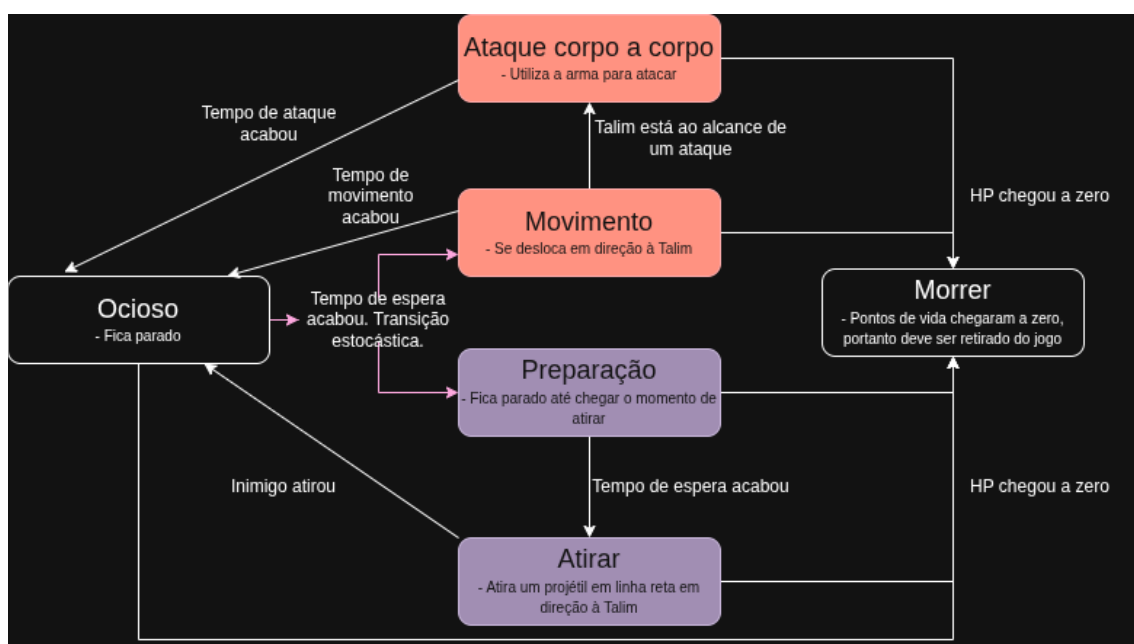


Figura 9: Máquina de estados que controla os inimigos no modo combate.

Finalmente, assim como a Talim, inimigos também são representados por *GameObjects* com um certo conjunto de componentes. A figura 10 ilustra os componentes presentes em um inimigo. Novamente, os componentes em rosa estão presentes tanto no modo exploração quanto no modo combate. Já os azuis, estão presentes apenas no modo combate. Ademais, o componente HP em vermelho é um componente anexado à Talim.

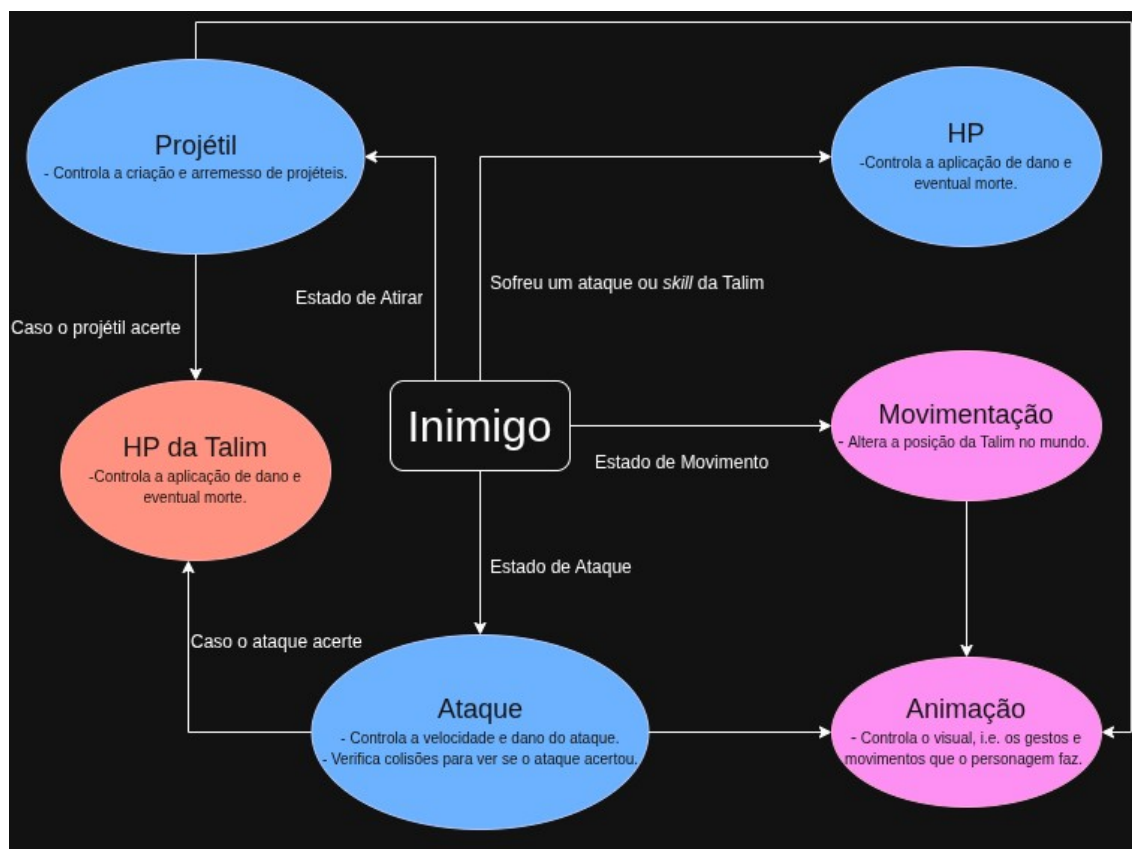


Figura 10: Componentes dos inimigos com suas respectivas interações.

6. Conclusão e trabalhos futuros

O jogo implementado é completo e possui progressão e narrativa com começo, meio e fim. Não obstante, almeja-se expandi-lo em trabalhos futuros, com a adição de novas personagens jogáveis (os outros dois protagonistas do livro, além da Talim) e novas missões para serem concluídas. Tais expansões não alteram a lógica base do jogo e podem ser facilmente implementadas utilizando com base os componentes e *GameObjects* já existentes.

Uma segunda expansão, mais ambiciosa, que será buscada é a implementação de um modo multi jogador, no qual dois ou três jogadores poderão jogar juntos, cada um controlando uma das personagens jogáveis. Essa expansão exige uma mudança na lógica de alguns componentes, pois será preciso controlar variáveis (e.g. inventário e *skills*) de vários jogadores ao mesmo tempo, além de processar simultaneamente *inputs* vindos de 3 *gamepads* diferentes. Assim sendo, devido à facilidade da implementação, será buscado primeiro a expansão mais simples.

Referências

- [1] Farnese, Carolina "A Lenda dos Guardiões, parte 1: O Recomeço de um Coração Despedaçado" [manuscrito não publicado]

- [2] Koster, Raph. *Theory of fun for game design*. " O'Reilly Media, Inc.", 2013.
- [3] Top countries and markets by video game revenues – Newzoo
- [4] Pesquisa da indústria brasileira de games 2023. ABRAGAMES, São Paulo (2023)
- [5] Singh, Swati, and Amanpreet Kaur. "Game development using unity game engine." 2022 3rd International Conference on Computing, Analytics and Networks (ICAN). IEEE, 2022.