A close up of a black sign

Description automatically generatedINSTITUTO SUPERIOR MANUEL TEIXEIRA GOMES

***Desenvolvimento de um Jogo Auto-Shooter utilizando a Engine Godot.***

## **Relatório de Metodologias de Investigação Científica**

**Cayan Moraes Prola**

Trabalho orientado por:

Prof. Cristiano Soares

2024

Desenvolvimento de um Jogo Auto-Shooter utilizando a Engine Godot.

# Declaração de autoria

Declaro ser o (a) autor(a) do trabalho apresentado neste relatório, sendo original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

(Assinatura)

O Instituto Superior Manuel Teixeira Gomes tem o direito, perpétuo e sem limites geográficos, de arquivar e publicitar este trabalho através de quaisquer meios, de o divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos não comerciais, desde que seja dado crédito ao autor.

(Assinatura)

# Agradecimentos

# Resumo

Este projeto teve como objetivo a criação de um jogo do subgênero auto-shooter, inspirado nas tendências modernas de jogos que evoluíram a partir do gênero shoot 'em up e, mais especificamente, do bullet hell. Inspirado em títulos como *Vampire Survivors* e *Deep Rock Galactic: Survivor*, utilizando a engine open-source Godot. O jogo incorpora características típicas do subgênero, como um sistema de upgrades dinâmico, spawn de hordas de oponentes, um mapa infinito e um sistema de armas com características únicas.

A escolha da engine Godot foi feita levando em conta diversas variáveis, como sua flexibilidade e facilidade de adaptação. A engine oferece suporte a múltiplas linguagens de programação, como C#, C++ e GDScript, sendo esta última uma linguagem muito semelhante ao Python, com a qual já possuo experiência. Essa familiaridade acelerou o processo de desenvolvimento, permitindo a implementação eficiente de funcionalidades complexas. Além disso, o fato de Godot ser open-source foi um fator decisivo, proporcionando maior transparência e confiabilidade no desenvolvimento do projeto.

O desenvolvimento enfrentou desafios técnicos e de design, incluindo a integração dos sistemas do jogo, a criação e otimização das armas, e a implementação de múltiplos upgrades. Embora o projeto esteja concluído, há oportunidades para futuros trabalhos, como a adição de novos inimigos, armas, upgrades e mapas que ampliariam a experiência do jogo.

Tabela de Conteúdos

[***Desenvolvimento de um Jogo Auto-Shooter utilizando a Engine Godot.*** 1](#_Toc176292274)

[**Relatório de Metodologias de Investigação Científica** 1](#_Toc176292275)

[Declaração de autoria 2](#_Toc176292276)

[Agradecimentos 3](#_Toc176292277)

[Resumo 4](#_Toc176292278)

[Introdução 6](#_Toc176292279)

[Estado da arte 7](#_Toc176292280)

[Desenvolvimento 7](#_Toc176292281)

[Escolha de assets 7](#_Toc176292282)

[Criação do personagem e sua movimentação 7](#_Toc176292283)

[Implementação da câmera 8](#_Toc176292284)

[Sistema de armas 8](#_Toc176292285)

[Discussão 8](#_Toc176292286)

[Conclusão 9](#_Toc176292287)

[Glossário 10](#_Toc176292288)

[Bibliografia 11](#_Toc176292289)

# Introdução

A indústria de jogos está passando por um momento peculiar no qual jogos produzidos por gigantes do mercado estão cada vez mais sofrendo com orçamentos gigantes que não geram retorno, falta de inovação nos lançamentos, jogos de serviço que não são atraentes ao público. E naturalmente, o crescimento de jogos indie acabou ocorrendo, projetos que são feitos com carinho e cuidado e que lembram a indústria de que jogos são arte, e não apenas um produto comercial.

Um exemplo notável é o subgênero dos *auto-shooters*, que evoluiu a partir dos jogos de bullet hell, dentro do gênero *shoot 'em up*. Um exemplo de um jogo desse é Vampire Survivors, que foi desenvolvido por apenas um desenvolvedor inicialmente, e revolucionou a indústria com uma jogabilidade simples, porém extremamente viciante e relaxante, disponibilizada em múltiplas plataformas e com um preço acessível, contrariando muito das práticas usuais das grandes empresas da indústria. O sucesso do jogo foi tão grande que inspirou vários outros jogos como Deep Rock Galactic: Survivors, e até sendo incorporado no universo de jogos da Riot Games, uma das maiores empresas da indústria, com o modo de jogo chamado “Swarm”, lançado recentemente no cliente do jogo League of Legends, possivelmente o maior *MOBA* da atualidade.

O objetivo deste projeto é desenvolver um jogo *auto-shooter* inspirado por essas tendências, utilizando a *engine* *open-source* Godot. O jogo busca ter a mesma jogabilidade que encantou milhões de jogadores pelo mundo e gerar uma sensação descontraída e relaxante de diversão.

A escolha do projeto se deu feita pela minha paixão em jogos, cultivada desde a infância, unida ao desejo de um desafio que nunca havia enfrentado: desenvolver um jogo completo. Juntamente com a oportunidade de expandir meus conhecimentos em programação e design de jogos e ter a experiencia de fazer algo que utilizei durante muito tempo.

Durante o desenvolvimento do jogo, foram empregadas diversas técnicas de otimização e design de gameplay, utilizando as funcionalidades da Godot para garantir um desempenho eficiente e uma experiência de jogo fluida. A escolha da *engine* Godot foi estratégica, considerando sua flexibilidade e suporte ao GDScript, que facilitou o desenvolvimento rápido e eficiente do jogo.

# Estado da arte

## Introdução ao gênero

O gênero *shoot 'em up* é um dos gêneros mais antigos da história dos videogames, tendo suas raízes na década de 1960. São caracterizados por colocarem o jogador em controle de uma espaçonave ou um personagem enfrentando ondas de inimigos diferentes, tendo como objetivo a sobrevivência e a destruição dos inimigos, necessitando boa coordenação motora para se mover em movimento constante, desviando dos projeteis inimigos e acertando os seus próprios disparos.

No final da década de 1970 e início da década de 1980, o gênero *shoot 'em up* começou a crescer em popularidade com jogos como Space Invaders (1978), Asteroids (1979) e Galaga (1981), jogos esses que estabeleceram conceitos e padrões de design utilizados até os dias de hoje, como sistemas de power-ups e batalhas contra chefes.

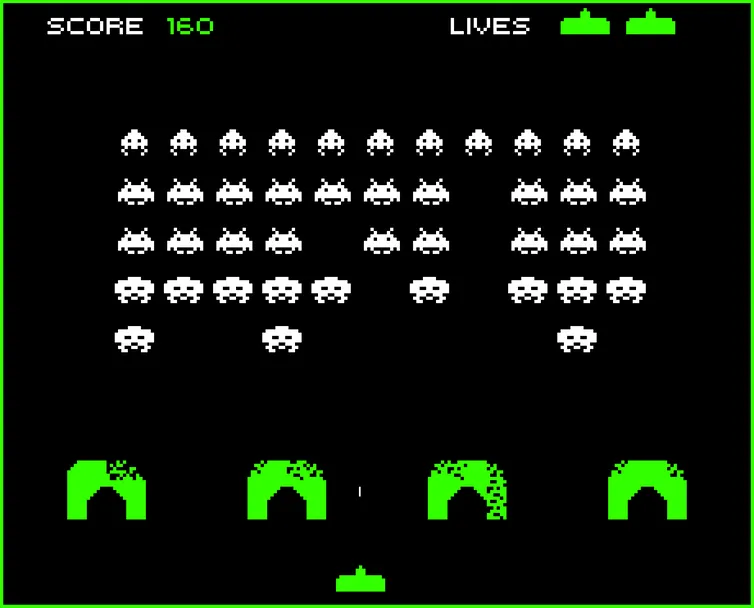


Figura 1: Imagem do jogo Space Invaders (1978).

Com o passar do tempo, o gênero se diversificou mais ainda, pois os jogos já não eram considerados desafiadores o suficiente, com isso, houve o surgimento do subgênero *bullet hell*, na década de 1990. Este nome vem da expressão japonesa “*danmaku*”, que traduzida ao português significa “barragem”, referindo-se a imensa quantidade de projeteis que o jogador deve se esquivar, aumentando a dificuldade e tornando a capacidade de completar os jogos uma verdadeira conquista. Este subgênero teve seu nascimento e ascensão no mercado de jogos japonês, e acabou tendo algumas de suas características aplicadas em jogos de outros gêneros, como Nier: Automata e Undertale, ambos jogos imensamente aclamados pela crítica especializada.

A screen shot of a video game

Description automatically generated

Figura 2: Imagem do jogo Touhou 2: The Story of Eastern Wonderland (1997)

Devido à grande dificuldade dos jogos *bullet hell,* o gênero acabou se tornando nichado e de difícil acesso, o que levou a continuação desse processo de evolução. O próximo passo evolucionário foi o surgimento do subgênero *auto-shooter*, que também pode ser chamado de *bullet* *heaven,* contrastando diretamente com *bullet* *hell*, pois quem atira todos os projeteis são o jogador, e seu objetivo principal é desviar e sobreviver dos inimigos, focando-se apenas na movimentação. Com os avanços tecnológicos ocorridos durante os anos 2000 e o grande crescimento do mercado de jogos, a possibilidade de jogos estarem disponíveis a qualquer momento através de um dispositivo móvel como um celular ou um vídeo game portátil como o *Steam* *Deck*, ocasionou com que houvesse uma facilitação a entrada para novos jogadores. Em união com a simplificação dos controles, a automatização dos disparos e a baixa necessidade computacional dos jogos *auto-shooter*, surgiu um cenário perfeito para eles fazerem muito sucesso.

## Análise de jogos relevantes

### Vampire Survivors

*Vampire Survivors* é um dos jogos mais influentes dentro do subgênero auto-shooter, e foi a grande inspiração para o desenvolvimento deste projeto. Desenvolvido inicialmente por um único desenvolvedor, este jogo destacou-se pela sua simplicidade e jogabilidade viciante. O jogo coloca o jogador em um ambiente 2D, onde deve sobreviver até um determinado tempo, lutando contra ondas de inimigos contínuas.

A mecânica de *auto-shooting* é a essência do jogo: o jogador apenas controla a movimentação do personagem, enquanto as armas disparam automaticamente. Porém, com um pequeno detalhe que torna o jogo diferente, que são elementos *roguelike*. *Roguelike* é um outro subgênero que cresceu bastante na década de 2010, que consiste em jogos no qual são necessárias várias tentativas para terminar uma missão ou mapa, e a cada tentativa o jogador recebe novas habilidades e até aumentos de estatísticas do personagem. Criando um loop de constante melhora, no qual quanto mais o jogador joga, mais poderoso se torna. Estes elementos *roguelike* acabaram se tornando algo muito comum nos jogos *auto-shooter,* aumentando a dificuldade inicial dos jogos, mas também criando uma re-jogabilidade muito maior.

No caso de Vampire Survivors, temos melhorias permanentes, que devem ser compradas com o ouro recebido em cada partida, e dentro de cada partida temos 6 espaços para escolhermos diversas opções de armas com atuações diferentes, e mais 6 opções de estatísticas com atuações diferentes, totalizando 12 espaços totais. Isso torna cada partida do jogo diferente, devido a grande possibilidade de escolhas possíveis. Unido a isso, existem combinações únicas para cada arma, que fazem com que as armas evoluam ainda mais, aumentando em muito seu poder e mudando a dinâmica do jogo.

Todas essas características foram inspirações diretas no meu projeto.

A video game screen with many people

Description automatically generated Figura 3 Imagem do jogo Vampire Survivors (2022)

### Deep Rock Galactic: Survivors

*Deep Rock Galactic: Survivors* é um spin-off do jogo *Deep Rock Galactic*, adaptando o universo rico e cooperativo do original para o formato auto-shooter. O jogo mantém a ambientação e universo característico do sucessor, porém com uma visão isométrica, elementos *roguelike*, mapas procedurais e com destruição de elementos.

As armas e habilidades disponíveis são as mesmas encontradas no primeiro jogo da série, trazendo um senso de continuidade e familiaridade para os fãs do original.

Além disso, o jogo mantém um foco muito maior em progressão de personagem, não sendo necessário apenas ouro para evoluir uma estatística do jogador, como também a coleta de diferentes ingredientes do universo do jogo para a evolução. O jogo também inclui diferentes missões opcionais dentro de cada partida, aumentando bastante a variedade entre partidas.

O sistema de mapas é uma grande evolução de outros jogos *auto-shooter*, o sistema de destruição do mapa, além de muito bem-feito, faz muito sentido no contexto do jogo, no qual os personagens jogáveis são mineradores. Todos os objetos destrutíveis são minérios, e não servem apenas como itens a serem coletados, mas também como uma maneira de se movimentar pelo mapa e criar caminhos diferentes.

Figura 4 Imagem do jogo Deep Rock Galactic: Survivor (2024)

## Conclusão do estudo da arte

Estudando a evolução dos jogos, principalmente do gênero *shoot 'em up,* produziu muitas informações relevantes na construção do projeto, desde a escolha do tema e conceito do projeto, como também nas diferentes maneiras de executar o desenvolvimento.

Estes jogos não só definem padrões para o gênero, mas também serviram de inspiração direta para o desenvolvimento do projeto, sendo Vampire Survivors a grande motivação por trás da escolha do tema.

# Métodos

O processo de desenvolvimento do jogo começou com uma fase de planejamento, para definir os objetivos e as funcionalidades que o jogo deveria ter. O objetivo era criar um jogo completo, com todas as funcionalidades que jogos do gênero auto-shooter possuem.

O desenvolvimento do projeto foi feito utilizando a engine open-source Godot na sua versão 4.2.2, escolhida pela sua flexibilidade e confiabilidade. Foi utilizado a linguagem GDScript, que está integrada diretamente na engine. Além disso, foi utilizado Git para controle de versão, Itch.io, Kenney e OpenGameArt para assets. Todos os assets utilizados são gratuitos e livres para uso.

## Escolha de assets

Após o processo de planejamento e a definição de objetos, comecei a escolha de assets, para visualizar como poderia fazer a arte do jogo, se as artes se comunicariam bem entre elas e se seria possível atingir a qualidade que gostaria. Todos os assets escolhidos e utilizados possuem estilos parecidos e são feitos em formato de pixel art. Após a escolha e coleta de todos os possíveis assets, comecei com o desenvolvimento do projeto.

## Criação do personagem e sua movimentação

A primeira implementação foi a criação do personagem principal e todos seus nodes e funcionalidades básicas. Os nodes utilizados para o personagem foram um **CharacterBody2D** como root node, e **AnimatedSprite2D** e **CollisionShape2D** como seus filhos. O node **CharacterBody2D** serve para a criação de personagens 2D com física e colisões que se movem através da funcionalidade implementada em scripts. O personagem é puramente controlado pelo utilizador, possui uma velocidade padrão e se move em qualquer eixo. O node **AnimatedSprite2D** é utilizado para utilizar um personagem animado com a utilização dos assets obtidos, o personagem possui animações enquanto anda, quando leva dano e quando morre. Já o node **CollisionShape2D** serve para controlar a colisão com os inimigos e possíveis obstáculos. Para a movimentação do personagem, foram adicionados novos inputs na engine, estes inputs foram as teclas W, A, S, D, teclas estas que são o padrão utilizado no mercado de jogos para teclados do padrão QWERTY. Estes inputs são então recebidos pela engine e lidos na função de física padrão da classe **CharacterBody2D**, onde cada um dos inputs representa uma direção específica.

## Implementação da câmera

Para garantir que o personagem fosse sempre o foco principal, foi adicionado um node **Camera2D** como filho do personagem. A câmera se movimenta juntamente com o personagem pelo mapa, atingindo o resultado necessário.

## Sistema de armas e estatísticas

A seguir, foi implementada outra funcionalidade base do personagem, implementar o sistema de armas e estatísticas. O jogador possui 4 espaços para armas e 4 espaços para estatísticas, a cada nível desbloqueado em uma partida, pode aumentar o nível de uma escolha anterior ou escolher uma opção nova, até estar com todos os espaços ocupados. Cada um dos upgrades possui 5 níveis, no caso das armas, cada nível traz funcionalidades diferentes para cada arma. Inicialmente, a implementação começou com apenas uma arma, o laser roxo, com o propósito de testar outras funcionalidades, como as colisões com os inimigos. E após a implementação de outros sistemas como o loop do jogo, a implementação correta dos inimigos e a implementação de algumas interfaces do jogo, foi feita a continuação do desenvolvimento das outra armas e estatísticas.

A arma padrão é o laser roxo, começando com o jogador desde o início das partidas. Inicialmente, dispara apenas em uma direção, para a direita, com uma velocidade fixa e um dano específico. Para os níveis 1, 3 e 5 é adicionado mais um disparo em uma outra direção, para a esquerda, para cima e para baixo respectivamente. Já para os níveis 2 e 4, os atributos de dano e velocidade da arma são aumentados.

# Discussão

# Conclusão

# Glossário

*Auto*-*Shooter*: Um subgênero de jogos originado de outro subgênero chamdo *Bullet* *Hell*, no qual o jogador apenas movimenta seu personagem pelo mapa, enquanto as armas do personagem disparam automaticamente durante as partidas, tendo como objetivo a sobrevivência de ondas de inimigos. Pode ser chamado também por *Bullet* *Heaven*.

*Shoot 'em up*:

*Bullet Hell:*

*Indie*: Abreviação de "independente", usada no mercado de jogos para se referir a jogos desenvolvidos por pequenos estúdios ou indivíduos, sem o apoio financeiro de grandes publicadoras.

*Engine*: Conjunto de ferramentas e bibliotecas que simplificam o desenvolvimento de jogos ou outros programas gráficos em tempo real, facilitando a criação de elementos como física, gráficos e som.

*Open-Source*: Aplicação com código fonte público, permitindo que desenvolvedores ao redor do mundo façam contribuições.

*MOBA*: Arena de Batalha Online Multijogador, é um gênero de jogos de estratégia onde jogadores controlam um personagem em um time, onde o principal objetivo é derrotar a base do time inimigo.

*Spawn*: Termo em inglês muito utilizado em jogos para se referir a um local de surgimento de objetos, sejam jogadores, objetos do mapa ou inimigos.

*HUD*: Interface gráfica que exibe informações importantes ao jogador durante o jogo, como vida, pontuação, mapa, entre outros.

*Riot* *Games*: Empresa norte-americana conhecida por desenvolver, publicar e organizar campeonatos de jogos extremamente populares, como *League of Legends*.

*GDScript*: Linguagem de programação de alto nível, orientada a objetos e imperativa, semelhante a Python. Otimizada e integrada com a engine Godot.

*Git*: É um sistema de controle de versões, muito utilizado para desenvolvimento de aplicações.

*Itch.io:*

*Kenney*:

*Assets:*

# Bibliografia