

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

MARTIM FERNANDES RIBEIRO LIMA

**INVESTIGANDO IMPLEMENTAÇÃO DE
MECÂNICAS DO GÊNERO RTS EM JOGOS
DIGITAIS EM REALIDADE VIRTUAL**

São Carlos, SP
2025

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

MARTIM FERNANDES RIBEIRO LIMA

**INVESTIGANDO IMPLEMENTAÇÃO DE MECÂNICAS DO GÊNERO RTS EM
JOGOS DIGITAIS EM REALIDADE VIRTUAL**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Ciência de Computação.

Orientador: Prof. Dr. Daniel Lucrédio

São Carlos, SP
2025

Dedico esse trabalho a todos os meus colegas, professores, amigos e família, sem os quais eu não chegaria perto de alcançar tudo que consegui até hoje e especialmente a todos os membros passados, presentes, e futuros do grupo GAMSo que acreditaram nos mesmos valores que me levaram até aqui.

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais que ao longo de todos esses anos me guiaram e puderam prover tudo que eu precisei durante essa longa jornada.

Também agradeço meus amigos, por sempre estarem comigo, me inspirarem, incentivarem e me proverem um lugar caloroso para voltar mesmo nos tempos mais difíceis. Agradeço em particular Romano Amadio, Felipe Borges e Vicente Lima por acompanharem, me darem *feedback*, e discutirem este longo trabalho comigo.

Deixo minha gratidão também ao meu orientador Daniel Lucrédio, que confiou no meu potencial ajudou a nutri-lo de varias maneiras. Nunca esquecerei as oportunidades de realizar coisas que eu não imaginava possíveis que o professor me deu.

Estendo também o agradecimento a todos os professores e funcionários do Departamento de Computação da UFSCar que eu tive o prazer de conhecer. Foram sempre muito pacientes e prestativos, sem o acompanhamento e ajuda de todos eu não teria chegado até aqui.

Por fim, mas não menos importante, agradeço profundamente ao grupo GAMSo: Ter a oportunidade de começa-lo, participar dele, aprender nele, criar nele e lidera-lo foram as experiencias mais marcantes que tive durante a graduação, espero que o grupo possa ser uma influencia tão grande para todos que passarem por ele quanto foi para mim.

“—I think that in life, as in game design, you have to find the fun. There is joy out there waiting to be discovered, but it might not be where you expected. You can’t decide what something’s going to be before you embark on it, and you shouldn’t stick with a bad idea just because you’re fond of it. Take action as quickly and repeatedly as possible, take advantage of what you already know, and take liberties with tradition. But most importantly, take the time to appreciate the possibilities, and make sure all of your decisions are interesting ones.”

(Sid Meier, Sid Meier’s Memoir: A Life in Computer Games)

Resumo

Jogos de realidade virtual permitem uma ampla exploração de implementações de interatividade com usuários, no entanto não há muita diversidade nos tipos de produto que esta mídia tem mostrado: A maioria adapta os mesmos tipos de gêneros e experiências de jogos digitais tradicionais. Essa monografia busca propor algumas soluções de implementação de uma mecânica central de um gênero de jogo digital convencional que quase nunca é visto adaptado para *RV* (Realidade Virtual), o *RTS* (*Real Time Strategy*); e então testá-las por meio de avaliações de usuários para determinar sua qualidade e efetividade. Assim foram determinadas características de várias possibilidades de implementação de mecânicas básicas de movimentação e seleção de unidades para jogos *RTS* em realidade virtual e foi produzido e disponibilizado um protótipo de implementação destas na *Godot Engine*.

Palavras-chave: Realidade Virtual, Estratégia em Tempo Real, Motor de jogos Godot, Mecânicas de jogos.

Abstract

Virtual reality video games allow for a wide exploration of new implementations of user interactions; however, there isn't much variety with the types of genres and experiences the medium has shown so far: most games have adapted the same types of genres and experiences of traditional "flat" video games. This dissertation aims to propose a few new implementation solutions for a central mechanic of a traditional video game genre that is seldom adapted to *VR* (*Virtual Reality*), the *RTS* (Real Time Strategy); and then test them through user evaluations to determine their quality and effectiveness. With that various characteristics of various basic *RTS* game mechanic implementations of unit selection and movement were determined and a prototype of those mechanics implemented under the Godot game engine was produced and made available.

Keywords: Virtual Reality, Real Time Strategy, Godot Engine, Game Mechanics.

Sumário

1	INTRODUÇÃO	9
2	TRABALHOS RELACIONADOS	11
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Jogos VR	13
3.1.1	Componentes do Hardware Moderno de Realidade Virtual	13
3.1.2	Padrões e APIs (OpenXR e Outras)	14
3.1.3	Interação em Jogos VR	14
3.1.4	Princípios de Design para Jogos VR	15
3.1.5	Desafios Técnicos e Limitações Atuais	16
3.2	Jogos RTS	16
3.2.1	História	16
3.2.2	Unidades	17
3.2.3	Construções	17
3.2.4	Recursos	18
4	METODOLOGIA	19
4.1	Mapeamento Inicial	19
4.2	Desenvolvimento do Protótipo	19
4.2.1	Hardware	19
4.2.2	Software	20
4.3	Experimento	21
5	RESULTADOS	22
5.1	Mapeamento Inicial	22
5.2	Desenvolvimento do Protótipo	22
5.2.1	Jogador	24
5.2.2	Mecânicas de seleção	25
5.2.2.1	Laser	25
5.2.2.2	Lente	25
5.2.2.3	Interface	26
5.2.3	Mecânicas de Movimentação	26
5.2.3.1	Laser	26
5.2.3.2	Direto	26
5.2.3.3	Traçar caminho	26

5.2.4	Entidades principais	27
5.2.4.1	Unidade	27
5.2.4.2	Recurso	28
5.2.4.3	Estrutura	28
5.2.4.4	Inventario	28
5.2.5	Objetivos	29
5.3	Realização dos Testes	30
5.4	Resultados	32
5.4.1	Processamento de Entrevistas	33
5.4.2	Análise Quantitativa e Síntese	35
5.4.3	Análise Final	36
5.4.4	Síntese dos Resultados	37
6	CONCLUSÃO	40
6.1	Trabalhos Futuros	40
	REFERÊNCIAS	41
	APÊNDICE A – LINKS IMPORTANTES	43

1 Introdução

Os jogos digitais têm um papel cada vez maior na sociedade, tanto culturalmente quanto como indústria. Este papel pode se tornar especialmente relevante no Brasil a partir da aprovação da Lei 14.852/24 que cria o marco legal para a indústria de jogos eletrônicos e regulariza a profissão de desenvolvedor de jogos, conforme notícia publicada no site camara.leg.br (2024).

Um dos campos nos quais se vê muita inovação tecnológica, tanto nos jogos digitais, quanto na ciência da computação, é a criação de software para realidade virtual (*VR*), no qual o usuário, através de um *HMD* - (*head-mounted display*) equipado de vários sensores como giroscópio, acelerômetro, câmeras ou emissores de luz infravermelha para detecção por aparelhos externos - pode experimentar a interação humano-computador por meio da imersão em um ambiente virtual tridimensional que acompanha seus movimentos.

Tal ambiente abre muitas possibilidades para explorar novas formas de interação com *software* Truman et al. (2018), visto que qualquer elemento funcional de uma dada interface pode ser apresentado de qualquer forma tridimensional, disposto em qualquer coordenada do espaço e ter uma interação programada que pode ir muito além do toque, digitação ou clique de um *mouse*.

De acordo com Vlahovic, Suznjevic e Skorin-Kapov (2022), para explorar essas novas possibilidades de interação, os jogos digitais em realidade virtual são uma ferramenta de pesquisa ideal por serem uma mídia que propõe diversas interações com *software* para fins de entretenimento não vistos em aplicações do dia a dia, tais como editores de texto, planilhas ou navegadores *web*, assim abrindo amplo espaço para experimentação de abordagens menos convencionais. Estes jogos têm muito potencial, tanto como entretenimento, arte digital ou *serious games* - jogos criados para aplicações além do entretenimento. Assim, faz sentido explorar tal meio para criar inovações e avançar o estado da arte na área de interação humano-computador.

Entre os jogos digitais, um dos gêneros mais conhecidos é o *RTS*, ou *Real Time Strategy*, popularizado pelo jogo de computador *Warcraft*, que foi publicado em 1994 por *Blizzard Entertainment*. O gênero continuou se proliferando e expandindo, gerando diversos subgêneros. Entre esses, o *MOBA* (*multiplayer online battle arena*) encontrou massiva popularidade no meio dos esportes eletrônicos, por meio do título *League of Legends*, publicado por *Riot Games*, o que demonstra a relevância deste formato de jogo até hoje.

Apesar de não ser comum encontrar definições formais para gêneros de jogos digitais, como explicado por Apperley (2006), visto que se pode escolher manter ou abandonar

seus elementos identificadores arbitrariamente e se misturar com outros gêneros em uma única obra, jogos *VR* não têm muita variedade no tipo de experiência que oferecem. Muitos deles usam os mesmos sistemas de interação, limitando o espaço de possibilidades do que é comum de se fazer e, portanto, havendo pouca oportunidade para experimentação e inovação. Na plataforma de jogos para computador mais prolífica do mercado - a *Steam*, cuja base de dados pode ser acessada pelo site *Steamdb*, pode-se averiguar que, tomando os marcadores comuns atribuídos aos jogos catalogados, tem-se 21% de jogos de simulação, 20% de *RPGs* (*Role Playing Games*), 18% *Puzzle* e 11% *Shooters*, enquanto que, para jogos que requerem *hardware VR*, essa mesma distribuição é 35% Simulação, 8% *RPGs*, 10% *Puzzle*, 15% *Shooters*. Quando se trata de jogos *RTS*, essa proporção é de 2% dos jogos comuns e 0,6% dos jogos *VR*.

Essa monografia tem como objetivo investigar implementações de interações com o usuário para realizar uma das mecânicas centrais de jogos *RTS*: a movimentação de unidades no espaço do jogo; e assim prover uma base da qual podem partir futuras explorações de como adaptar este gênero à realidade virtual para que possa se gerar mais conhecimento sobre esses campos.

Inicialmente foi feita uma pesquisa exploratória para conhecer o que é ofertado no mercado de jogos *VR* e o que é pouco representado e pode ser explorado. Essa pesquisa levou à escolha do gênero *RTS*. Com isso, após buscar a mecânica mais fundamental e praticável com objetivo de adaptar ao contexto de realidade virtual no escopo possível para a pesquisa em jogos do gênero, foi selecionada a movimentação de unidades. A fim de experimentar diferentes implementações dessa mecânica com usuários, foi desenvolvido um protótipo para avaliação. Este protótipo é composto por uma série de tarefas comuns realizadas em jogos do gênero, implementadas em um ambiente de realidade virtual; o usuário realiza cada uma dessas tarefas, podendo utilizar essas mecânicas. Por fim, o usuário é entrevistado, são feitas perguntas relacionadas à intuitividade, facilidade de uso e outros aspectos das variações de mecânicas testadas para determinar as qualidades e defeitos de cada abordagem.

Com a realização do teste com 33 usuários, foi determinado que as mecânicas desenvolvidas são adequadas para jogos *RTS* em Realidade Virtual, apesar da variação na preferência dos usuários, cada uma delas evidenciou claras vantagens e desvantagens em diferentes contextos possíveis em um jogo.

O projeto de jogo desenvolvido para o experimento foi disponibilizado publicamente na plataforma *github* e, por ser desenvolvido em uma plataforma acessível e de código aberto, pode ser facilmente usado como base para o desenvolvimento de um jogo.

2 Trabalhos Relacionados

Existem vários escritos que tratam de temas tangenciais como a proposta de um *framework* em *Godot* para jogos *RTS* por Malý (2024). Este é um trabalho que implementa também um protótipo do mesmo gênero e utilizando a mesma ferramenta principal de desenvolvimento, porém sem a experimentação com realidade virtual. O protótipo criado pelo autor tem o objetivo de servir como base para a criação rápida de jogos *RTS* com mínima necessidade de escrita de novo código utilizando o padrão de arquitetura de projeto *ECS* (*Entity Component System*).

Já a implementação de um ambiente de realidade virtual por Santucci et al. (2020) é um protótipo também utilizando o motor *Godot* em busca de uma abordagem de código aberto e baixo custo; no entanto, trata-se de uma aplicação primariamente para uso educacional que não procura explorar mecânicas de jogos *VR*, focando muito mais na interação do ambiente com os movimentos feitos pelo usuário, com um foco na imersão dentro do ambiente virtual.

Por fim, o trabalho mais próximo identificado foi o de Truman et al. (2018), que propõe abordagens para criação de jogos em realidade virtual do gênero *RTS* por meio de um protótipo de jogo. O artigo identifica a necessidade de mais pesquisa nesse tópico. Além disso, determina que a tecnologia de realidade virtual tem grande potencial para aprimorar o gênero *RTS*. No entanto, não há uma busca por comparar diferentes abordagens de implementação de mecânicas e testar com usuários.

Houve também tentativas de adaptação de mecânicas de jogos tradicionais para a realidade virtual, como a de Mußmann, Truman e Mammen (2021), que, apesar de não contemplar o gênero *RTS*, segue um processo de exploração e iteração de implementação para sistemas de inventário em jogos *VR* de aventura.

Link et al. (2016) experimentou com usuários um protótipo de jogo *RTS* em realidade mista, por meio de um jogo de tabuleiro digital com o qual se interagia usando uma mesa com tela de toque. Essa pesquisa também realizou testes com usuários com um gênero semelhante de jogo e tecnologia de realidade aumentada.

Em termos mais gerais, também existem diversos estudos analisando a realidade virtual como meio de *IHC*, por meio de pesquisa exploratória. Como os analisados na meta-análise de Azofeifa et al. (2022). Esta buscou novas avenidas para a pesquisa de *IHC* multimodal em diversos domínios de conhecimento que utilizassem essa tecnologia.

No trabalho de Phillips e Piekarski (2005), os autores investigam técnicas de interação para jogos *RTS* em ambientes de Realidade Aumentada (*AR*), abordando a limitação

imposta pela necessidade de correspondência entre o espaço físico e o ambiente do jogo. Para contornar essa restrição, é proposta uma mecânica que permite ao jogador assumir a perspectiva de qualquer unidade sob seu controle, visualizando e interagindo com o ambiente a partir desse ponto de vista e testa essa mecânica em um protótipo de jogo. A relação com esta pesquisa reside na adaptação de mecânicas de *RTS* para ambientes imersivos e na abordagem semelhante de exploração mecânica; no entanto, foca em realidade aumentada e não virtual.

Por fim, Vlahovic, Suznjevic e Skopin-Kapov (2022) explora a adaptação de jogos *RTS* para Realidade Virtual, propondo uma taxonomia do gênero aplicada a títulos em *VR*. O trabalho identifica desafios específicos, como a necessidade de novos métodos de navegação em ambientes imersivos e também cria um protótipo como proposta de solução. A principal contribuição dos autores é uma análise sistemática de elementos fundamentais de *RTS* (como controle de unidades, visão do campo de batalha e interface) no contexto de *VR*, oferecendo uma base teórica para futuras adaptações do gênero. Essa pesquisa relaciona-se diretamente com o presente trabalho, mas traz uma abordagem mais teórica e não realiza testes de sua solução com usuários.

3 Fundamentação teórica

3.1 Jogos VR

Hardware moderno de realidade virtual que segue o padrão da *API OpenXR* - a principal usada no desenvolvimento de *software* que interage com tal *hardware* - é composto de um visor utilizado na cabeça do usuário e dois controles para serem segurados nas mãos, todos com sensores de movimento para rastrear sua posição e orientação relativas ao ambiente. Os controles também possuem *inputs* semelhantes aos dos controles padrões para jogos digitais modernos, com botões digitais ou analógicos com uma ou duas dimensões de vetores.

Este formato é categoricamente diferente de um *mouse* e teclado que são normalmente usados como dispositivos de entrada em jogos *RTS*. Não há possibilidade de interagir rapidamente com coordenadas em uma tela com um *mouse* ou acessar múltiplas de funções e atalhos que possam ser usadas para realizar comandos diferentes em um teclado. É uma tecnologia que cria ambientes digitais imersivos, nos quais o usuário interage por meio de dispositivos como *head-mounted displays (HMDs)* e controladores de mão, substituindo completamente a percepção do mundo real. Essa imersão distingue-se de outras abordagens tecnológicas, como a Realidade Aumentada (*AR*), que sobrepõe elementos digitais ao ambiente físico (ex.: *Pokémon GO*), e a Realidade Mista (*MR*), que combina objetos virtuais com interação contextual no espaço real (ex.: *Microsoft HoloLens*). Enquanto *AR* e *MR* estendem a realidade, a realidade virtual exige maior capacidade computacional para gerar mundos autônomos, tornando-a especialmente relevante para jogos e simulações complexas.

A evolução da realidade virtual em jogos remonta a dispositivos pioneiros como o *Sensorama* (1962) e o *Sword of Damocles* (1968), mas seu uso comercial só ganhou impulso nos anos 1990, com o fracasso do *Virtual Boy* (Nintendo, 1995), que revelou as limitações técnicas da época. A virada ocorreu na década de 2010, com o *Oculus Rift* (2016), que popularizou *HMDs* acessíveis e de alta qualidade, impulsionando avanços em *tracking*, resolução de tela e padronização de *APIs*. Atualmente, a realidade virtual consolidou-se como plataforma tanto para entretenimento quanto para aplicações profissionais, com *frameworks* como o *OpenXR* facilitando o desenvolvimento multiplataforma.

3.1.1 Componentes do Hardware Moderno de Realidade Virtual

Os visores modernos (*HMDs*) empregam tecnologias de exibição como painéis *OLED*, que oferecem cores vibrantes e pretos profundos, ou painéis *LCD*, mais acessí-

veis, mas com menor contraste. A resolução varia entre 2K e 4K por olho, enquanto taxas de atualização superiores a 90 Hz são essenciais para reduzir sensação de enjoo e vertigem. O rastreamento de movimento pode ser *inside-out*, onde câmeras no próprio visor mapeiam o ambiente (ex.: *Meta Quest*), ou *outside-in*, que depende de sensores externos (ex.: *Valve Index*). As lentes, geralmente do tipo *Fresnel* ou *Pancake*, ajustam-se à distância focal do usuário, e um campo de visão (*FOV*) acima de 100° busca aproximar-se da visão humana para maior imersão.

Já os controladores de mão utilizam sensores de *6DoF* (seis graus de liberdade), capazes de detectar não apenas rotação, mas também posicionamento espacial. Seus sistemas de *input* incluem gatilhos analógicos para pressão variável, botões capacitivos que respondem ao toque sem necessidade de pressionamento físico e, em modelos mais antigos, *trackpads* para navegação alternativa.

3.1.2 Padrões e APIs (OpenXR e Outras)

A falta de padronização nos primórdios da realidade virtual levou à fragmentação do desenvolvimento, com *APIs* proprietárias como *Oculus SDK* e *OpenVR* exigindo adaptações específicas para cada plataforma. Essa limitação foi superada com a introdução do *OpenXR* (*Khronos Group*, 2019), um padrão aberto que unifica o desenvolvimento, permitindo que um mesmo código seja executado em dispositivos de diferentes fabricantes. Suas principais vantagens incluem interoperabilidade entre *hardwares* e suporte nativo em motores como *Unity*, *Godot* e *Unreal Engine*, reduzindo a necessidade de reimplementações e facilitando a portabilidade de projetos.

3.1.3 Interação em Jogos VR

A locomoção em realidade virtual apresenta desafios únicos, pois métodos tradicionais como movimento contínuo via *joystick* podem induzir enjoo em usuários sensíveis. Alternativas como teleportação (deslocamento instantâneo entre pontos) e movimento natural (*room-scale*, onde o jogador anda fisicamente no espaço) buscam equilibrar imersão e conforto. Técnicas como *snap turning* (rotações em ângulos fixos) e *comfort modes* (redução do campo de visão durante deslocamento) são estratégias comuns para mitigar enjoos.

A manipulação direta de objetos com física realista, como em *Half-Life: Alyx*, é uma das grandes vantagens da realidade virtual, permitindo ações como pegar, arremessar e montar estruturas de forma intuitiva. Já ponteiros virtuais (como *laser pointers*) são frequentemente utilizados para seleção à distância em interfaces. A *UI/UX* adaptada também é crucial, com menus tridimensionais posicionados no espaço do jogo, evitando sobreposições imersivas que quebram a ilusão de realidade.

A limitação de botões em controladores de realidade virtual exige soluções criativas, como combinações de gestos (ex.: segurar gatilho + movimento de punho) e menus radiais contextuais (*Beat Saber*). Jogos como *Boneworks* e *The Walking Dead: Saints & Sinners* demonstram como sistemas avançados de física e inventários corporais podem superar essas restrições.

3.1.4 Princípios de Design para Jogos VR

A escolha entre esqueumorfismo (interfaces realistas, como as de *Job Simulator*) e abstração (elementos simplificados, como em *Superhot VR*) depende do contexto do jogo. Neste trabalho, optou-se por um ponteiro-mouse virtual no protótipo de *RTS*, equilibrando precisão estratégica e usabilidade em um ambiente tridimensional.

As interfaces de usuário em realidade virtual demandam uma reformulação completa dos paradigmas tradicionais de design. A abordagem diegética, onde elementos de interface são integrados organicamente ao ambiente tridimensional, tem se mostrado particularmente eficaz para manter a imersão. Exemplos incluem relógios de pulso virtuais, como em *Pistol Whip*, ou painéis de informação que aparecem contextualmente nas superfícies do mundo virtual. O *feedback* multimodal - combinando estímulos visuais, auditivos e hápticos - é outro elemento crucial para comunicar eficazmente as respostas do sistema às ações do usuário sem recorrer a elementos de interface tradicionais que quebram a ilusão de realidade.

O conforto do usuário deve ser considerado como prioridade absoluta em qualquer projeto de realidade virtual. Isso envolve não apenas a mitigação de *motion sickness* através de técnicas como *snap turning* e redução controlada do campo de visão durante movimentos, mas também a atenção à duração das sessões interativas. Estudos sugerem que experiências contínuas além de 20-30 minutos podem levar à fadiga visual e física, indicando a necessidade de estruturas de jogo que permitam pausas naturais ou sessões mais curtas.

A manutenção da sensação de presença constitui outro pilar essencial do design para realidade virtual. Ambientes virtuais devem exibir coerência comportamental, com objetos respondendo de maneira previsível e física às interações do usuário. O áudio espacial desempenha papel fundamental nesse aspecto, não apenas como elemento imersivo, mas como ferramenta de navegação e orientação. Títulos como *Resident Evil 7 VR* demonstraram como a sonorização tridimensional pode guiar intuitivamente o usuário através de espaços complexos.

3.1.5 Desafios Técnicos e Limitações Atuais

Apesar dos avanços, a realidade virtual ainda enfrenta obstáculos significativos. A latência, quando superior a 20 ms entre movimento e renderização, pode causar desconforto e quebrar a imersão. Além disso, a demanda por *hardware* robusto (como *GPUs RTX 3060+*) limita a acessibilidade. Outro desafio é a escassez de grandes produções de jogos nativos para realidade virtual, com a maioria das experiências sendo adaptações (ex.: *Skyrim VR*) ou títulos curtos, refletindo um mercado ainda em consolidação.

3.2 Jogos RTS

Para os fins dessa pesquisa, serão adotados os seguintes elementos mecânicos básicos que compõem a maioria dos títulos identificados popularmente como *RTS* e serão elaborados a seguir após uma explicação do surgimento do gênero.

O termo *RTS* (*Real-Time Strategy*) se refere a um subgênero de jogos de estratégia onde as ações, decisões e consequências destas ocorrem em tempo real.

3.2.1 História

O veículo Gamespot (2001) relata que um dos primeiros jogos a aparecer no mercado possuindo as mecânicas comuns de um *RTS* foi *Herzog Zwei* (1989 Japão/ 1990 lançamento global). O jogo se resume a uma partida de um jogador contra o outro com o objetivo de destruir a base inimiga, apesar da presença de diversas unidades ao seu comando, o jogador pode controlar apenas uma, tendo as outras como suporte para sua conquista. Esta é a primeira aparição de uma mecânica nesta escala entre os jogos e é uma das mecânicas que define o gênero atualmente. *Herzog Zwei*, apesar de ser um jogo apenas para *consoles*, e não ser uma inspiração direta para o jogo que é notoriamente intitulado como o primeiro *RTS*, mostra que os desenvolvedores da época já se interessavam na possibilidade de usar estratégia em tempo real em seus jogos.

Três anos depois do lançamento de *Herzog Zwei*, é lançado *Dune II*, jogo popularmente mencionado como o primeiro *RTS* por grande parte do público principal do gênero, como documentou Winstanley (2023). *Dune II* não necessariamente apresentou novas mecânicas ao público, mas sim uma coletânea delas que, juntas, se popularizaram como o gênero *RTS*. As inspirações para o jogo vêm de outros jogos da época, como: *Eye of the Beholder*, *Military Madness*, *Sid Meyer's Civilization* e *Populous*. Estes jogos foram utilizados como base para vários dos sistemas presentes em *Dune II*, como, por exemplo, o combate estratégico, a interface e os combates rápidos, o gerenciamento de recursos e o mundo explorável, e, por fim, a falta de controle de seus aliados, que agiam com base em seus comandos, mas de maneira própria.

Após o lançamento e considerável sucesso comercial de *Dune II*, o mercado abriu as portas para outros jogos do gênero, vindo logo em sequência o jogo *Warcraft* e *Command & Conquer* que expandem os conceitos apresentados inicialmente em *Dune II* de maneiras diferentes, crescendo o gênero. Ambos os jogos tiveram uma recepção variada, sendo *Command & Conquer* um sucesso aclamado e *Warcraft* um sucesso mediano, mas que garantiu a produção de sua sequência, *Warcraft II*, que, por sua vez, obteve uma grande onda de sucesso e se tornou um dos jogos mais populares do gênero até hoje.

Por volta dos anos 2000, o gênero era apreciado por uma base estável de jogadores, mas pouca inovação nos conceitos, com os jogos, em sua maioria, recebendo apenas refinamentos dos aspectos já presentes nos títulos do passado. Apenas em 2002 foi adotada como padrão a construção dos jogos em ambiente 3D, descreve Toronto (2008).

Poucos subgêneros surgiram do próprio *RTS*, mas os mais notáveis são o *RTT* (*Real-Time Tactics*), que possui um foco menor na gestão de recursos, e o gênero *MOBA* (*Multiplayer Online Battle Arena*), onde os jogadores têm controle sobre uma unidade mais forte que possui níveis e habilidades diferentes. O gênero mantém o objetivo de destruir a base inimiga, mas abandonou a construção de estruturas e o controle de outras unidades. Jogos famosos destes gêneros incluem títulos como a série de jogos *Total War* e *League of legends*, respectivamente.

3.2.2 Unidades

Dentro de um jogo *RTS* uma das mecânicas mais importantes é o uso de outras unidades controladas pelo computador para o seu auxílio. Essas unidades agem a favor do jogador, ajudando na coleta, ataque e descoberta do mapa, ajudando a completar tarefas básicas mais facilmente e aumentando seu controle sobre o mapa, assim podendo produzir mais unidades e ganhando mais força contra seu oponente. Durante a partida, um dos principais objetivos imediatos deste tipo de jogo é, progressivamente, acessar unidades mais poderosas para levar o jogador à vitória.

3.2.3 Construções

Construções possuem diferentes utilidades dentro do jogo, existem construções que produzem unidades novas, construções que liberam a construção de outras construções mais avançadas, e construções que produzem itens. Todas estas construções fazem parte do ciclo de gerenciamento do jogo, no qual o jogador monitora a produção e o consumo de itens.

3.2.4 Recursos

Recursos funcionam essencialmente como uma variedade de moedas que o jogador e suas construções utilizam para produzir qualquer tipo de ação no jogo. Eles são necessários para gerar construções e unidades, pois cada uma destas possui um pré-requisito para ser produzida ou construída. Esses recursos, geralmente, podem ser obtidos em diversos pontos do ambiente do jogo, às vezes necessitando de construções para serem extraídos ou de unidades específicas para serem coletados.

4 Metodologia

4.1 Mapeamento Inicial

Dada a baixa frequência de jogos *RTS* em realidade virtual discutida no Capítulo 1, esse gênero foi escolhido como alvo para experimentar implementação de mecânicas por ser pouco explorado, assim aumentando a chance de que novas soluções pudessem ser descobertas na pesquisa.

No início da pesquisa, antes de serem decididas as mecânicas como alvo de exploração de implementações, foi feito um mapeamento dos jogos classificados como *RTS* na plataforma *Steam*, para que fossem identificados seus elementos mais comuns.

Com a decisão do foco em mecânicas de seleção e movimentação de unidades, foi determinado que, para avaliar as abordagens de implementação, seria necessário um teste com usuários.

4.2 Desenvolvimento do Protótipo

Com a mecânica decidida, a abordagem selecionada para avaliar possíveis implementações foi criar um protótipo de jogo com diversas variações dessa mecânica e testar tal protótipo com usuários para determinar sua qualidade. Foi decidido que seria usado o motor de desenvolvimento de jogos *Godot* pela sua praticidade e natureza de código aberto. O motor conta com *plugins* criados para desenvolvimento de software de realidade virtual.

4.2.1 Hardware

Foi utilizado o *headset* de realidade virtual *Pico 4*, este *hardware* é desenvolvido pela empresa chinesa *PICO*, subsidiária da *Bytedance*, e conta com processador de arquitetura *arm* customizado para realidade virtual: o *Snapdragon XR2*. O sistema operacional deste *headset* é uma versão customizada para realidade virtual do *Android* e é completamente autocontido sem necessidade de ser acoplado a outro dispositivo com seu próprio *CPU*, assim como um *smartphone*. O *headset* tem bateria própria e é capaz de monitorar constantemente sua posição no espaço e a de seus controles associados por meio de câmeras externas.

O sistema operacional do dispositivo permite a utilização da *Android Debug Bridge* (*ADB*), que possibilita iteração e depuração rápida de *software*, além de função nativa de

gravação da tela e acesso direto ao sistema de arquivos por meio da interface *USB*. Assim, foi planejada a gravação dos testes para futura análise.

Figura 1 – HMD Pico 4



4.2.2 Software

Para implementar o protótipo em si foi escolhido a *Godot Engine* versão 4.4.1, um motor de jogo de código aberto que utiliza a sua própria linguagem de programação - *GScript*. A escolha desta ferramenta é interessante para este projeto pois seus recursos e *workflows* conduzem o desenvolvimento da aplicação para ser modular e poder ser facilmente reaproveitada por qualquer desenvolvedor interessado.

Vários dos elementos essenciais da *engine* são pensados desde sua concepção para se adequar a padrões de projeto de boa qualidade, como a hierarquia de cenas e objetos que funciona seguindo o padrão *ECS (Entity-Component System)*, que se baseia em criar funcionalidades agregando objetos modulares com os quais compõem comportamentos diferentes. O protocolo mais básico de comunicação entre objetos em *Godot* também é construído sobre o padrão *Observer*, requerendo que objetos emitam e se inscrevam em sinais ao invés de terem referências diretas uns aos outros, assim minimizando o acoplamento entre classes.

Essa ferramenta também implementa a API *OpenXR* possibilitando o desenvolvimento de programas que podem ser usados com equipamento *VR* moderno e pode compilar programas utilizando *Gradle* permitindo a criação de programas executáveis em Óculos *VR Android*, como é o caso do *Pico 4*.

Uma variedade de *plugins* e *addons* também é criada e disponibilizada pela comunidade de desenvolvedores de *Godot*. Para este projeto foi essencial o *plugin XR tools*

que implementa algumas *features* padrão de jogos *VR*, compondo uma base robusta para o desenvolvimento da aplicação. Para depuração e testes, também foi usado o *addon XRSimulator*, que permite testar aplicações *VR* sem a necessidade de utilizar um visor.

4.3 Experimento

Com o protótipo desenvolvido e as mecânicas implementadas, o próximo passo foi testar com usuários para determinar seu funcionamento, qualidades e casos de uso no contexto para o qual foram projetadas. Os usuários testaram as mecânicas e depois foram entrevistados com o objetivo de coletar dados para uma análise qualitativa por meio de codificação.

A proposta do protótipo é apresentar ao usuário alguns cenários comuns de tarefas performadas neste tipo de jogo, como levar unidades a um ponto no ambiente, ou coletar recursos, por exemplo. Cada um destes cenários seria repetido 3 vezes, cada vez com uma implementação diferente da mecânica de seleção e movimentação de unidades.

5 Resultados

5.1 Mapeamento Inicial

Para iniciar a investigação de como expandir o estado da arte em interações em jogos de realidade virtual, foi determinado que era necessário escolher algum tipo de mecânica que não era muito representada neste universo. Com isso, um dos gêneros que chamou a atenção por sua falta de representatividade foi o *RTS*: de dezenas de milhares de jogos publicados na plataforma *Steam* na data de publicação desta pesquisa, apenas 39 tinham os marcadores “*Only for VR*” e “*RTS*”, a URL da consulta está disponível no Apêndice A.

Tendo decidido o gênero *RTS* para pesquisa, o próximo passo foi determinar qual mecânica deste gênero seria interessante avaliar possíveis adaptações para realidade virtual. Para este fim, foram analisados os 10 jogos mais populares do gênero e enumeradas as mecânicas e interações mais comuns entre eles. Assim, foi selecionada uma mecânica que é a base de interação em muitos destes jogos e que não tem um análogo claro no contexto de realidade virtual: a seleção e comando de unidades.

5.2 Desenvolvimento do Protótipo

Assim, para que fosse possível interagir com essas mecânicas de forma já familiar aos usuários, buscou-se seguir o princípio de design de familiaridade, seguindo as recomendações de Piaskiewicz (2019), por exemplo, foi acoplado um “ponteiro” à mão do usuário que se poderia usar tal qual um *mouse* para selecionar opções em menus e “clicar” em objetos ou coordenadas do ambiente, apontando para eles diretamente com a mão, assim buscando criar uma interface intuitiva que usuários pudessem entender mesmo sem experiência com realidade virtual. Todas as mecânicas implementadas buscaram seguir este princípio conforme possível.

Em contrapartida, adaptar funcionalidades de um teclado para realidade virtual foi mais complexo devido à pouca quantidade de botões. Por um lado, isso pode ser um facilitador ao *design*, pois dificilmente usuários têm que procurar qual botão apertar, o que nem seria possível em um ambiente *VR* devido à limitação visual do ambiente real, mas, para atender aos requisitos de implementação de uma variedade de funcionalidades, necessita da programação de combinações de botões que podem se apresentar complexas demais.

Com essas considerações, foram elaboradas as seguintes *features* para o protótipo

publicado no *github*, vide Apêndice A. A seguir, serão listados os objetos de jogo, mecânicas e tarefas para usuários implementadas no protótipo e, em seguida, explicados em detalhes um a um:

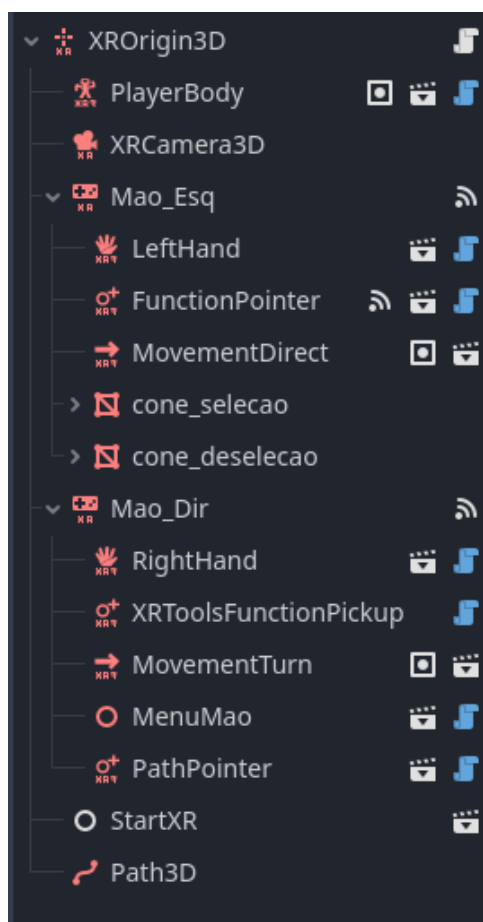
- Implementados três tipos de entidades que interagem entre si e com o jogador:
 - **Recursos:** “*Containers*” no qual o jogador ou as unidades podem pegar objetos para transportar
 - **Estruturas:** “*Containers*” no qual o jogador ou as unidades podem deixar objetos que estão carregando
 - **Unidades:** Podem ser comandadas para irem a um dado ponto no ambiente, pegar itens em recursos e deixar itens em estruturas
- Implementados três tipos de mecânicas de seleção:
 - **Laser:** É utilizado o ponteiro já ligado a mão esquerda do jogador que também é usado para outras interações para selecionar unidades
 - **Lente:** Permite seleção ou de-seleção de todas as unidades dentro do campo de visão do jogador
 - **Interface:** Permite interação com um menu no espaço do jogo ligado a mão do jogador que permite selecionar unidades pelo nome
- Implementados três tipos de mecânicas de movimento:
 - **Laser:** É utilizado o ponteiro já ligado a mão esquerda do jogador que também é usado para outras interações para mandar unidades para um dado ponto no ambiente
 - **Direto:** Permite movimentar unidades selecionadas diretamente por meio do *joystick* no controle direito
 - **Traçar-caminho:** Permite que o jogador escolha vários pontos no chão para que as unidades sigam em ordem
- Implementadas três atividades que testam interações comuns presentes neste gênero de jogo
 - **Atividade 1:** Levar unidades a um ponto do ambiente
 - **Atividade 2:** Fazer unidades pegarem itens nos recursos
 - **Atividade 3:** Fazer unidades deixarem itens em uma estrutura

A seguir será explicado como o projeto para testar as mecânicas foi construído.

5.2.1 Jogador

A classe central do projeto é a do jogador, apresentada na Figura 2, esta emprega uma abordagem multimodal que combina gestos manuais, interfaces espaciais e *feedback* visual imediato. A arquitetura proposta busca trazer elementos fundamentais de interação em ambientes virtuais tridimensionais através de um conjunto integrado de mecanismos que preservam a imersão enquanto mantêm a precisão necessária para jogos do gênero *RTS*.

Figura 2 – Arquitetura do jogador



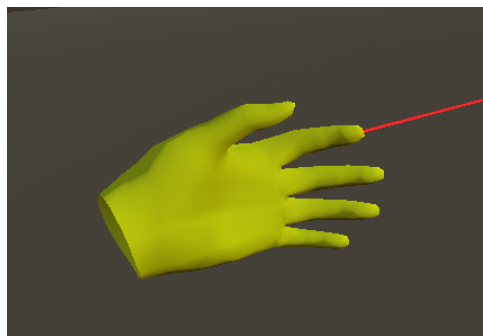
As mecânicas foram elaboradas por processo *ad hoc*, pensando em que tipos de comandos poderiam ser realizados com os *inputs* disponíveis nos controles do equipamento que seriam relevantes às ações comuns de jogos *RTS* mapeadas anteriormente, procurou-se primeiro implementar ideias mais simples de interação pensadas intuitivamente e avançar para tentativas de ideias mais complexas e elaboradas ao longo da pesquisa, como ações que requeriam que o usuário apertasse múltiplos botões ou fizesse diferentes movimentos.

5.2.2 Mecânicas de seleção

5.2.2.1 Laser

O laser visível na Figura 3 é o modo padrão de seleção com o propósito de imitar um *mouse* de computador - o método primário de seleção de unidades em jogos *RTS*. Essa funcionalidade emite um *raycast* constantemente da mão do jogador que, ao colidir com uma unidade, permite ao jogador apertar o gatilho do controle para selecioná-la. Em um jogo conhecido do gênero, como *Age of Empires*, esta ação seria análoga a clicar em uma unidade.

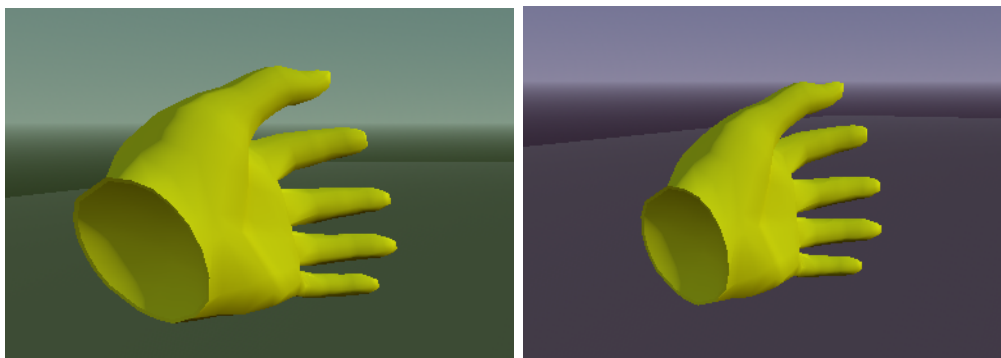
Figura 3 – Modo padrão da mão esquerda



5.2.2.2 Lente

A lente tem a intenção de poder ser usada rapidamente, análoga ao atalho “*ctrl+a*” em um teclado, que geralmente é associado com a ação “selecionar tudo” em vários *softwares* e jogos, permitindo acessar todas as unidades sem a necessidade de apontar precisamente para uma de cada vez. Dentro do espaço tridimensional do jogo, isso é realizado criando um grande cilindro de colisão parcialmente transparente na frente do jogador, o que permite interação com todas as unidades em seu campo de visão, sendo ilustrada por uma leve, mas perceptível mudança na cor da área afetada.

Figura 4 – Modo de seleção visual



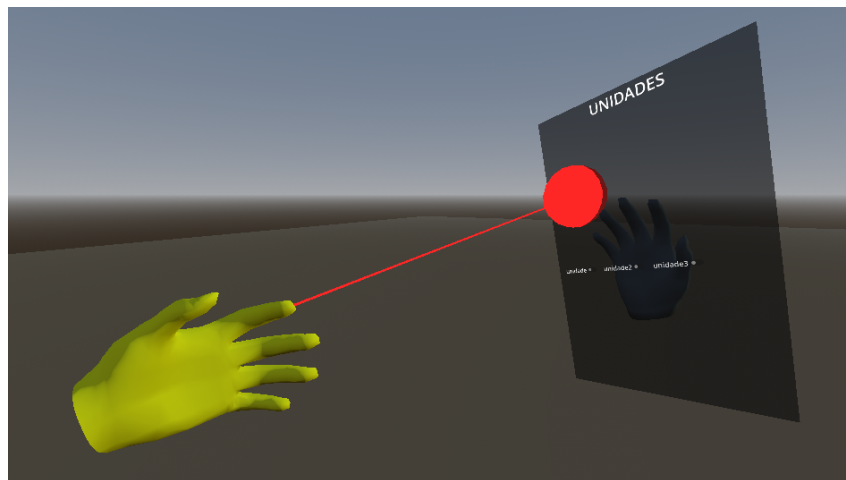
(a) Visual de seleção

(b) Visual de de-seleção

5.2.2.3 Interface

A interface tenta replicar um “*tablet*” trazendo mais um elemento de familiaridade com o mundo real. Usando o mesmo laser de seleção, é possível apontar para essa interface que surge na mão direita do jogador ao pressionar um botão no controle para selecionar unidades, sem a necessidade de olhar para elas diretamente, semelhante a interfaces de listas de unidades em jogos *RTS* tradicionais.

Figura 5 – Modo seleção por UI



5.2.3 Mecânicas de Movimentação

5.2.3.1 Laser

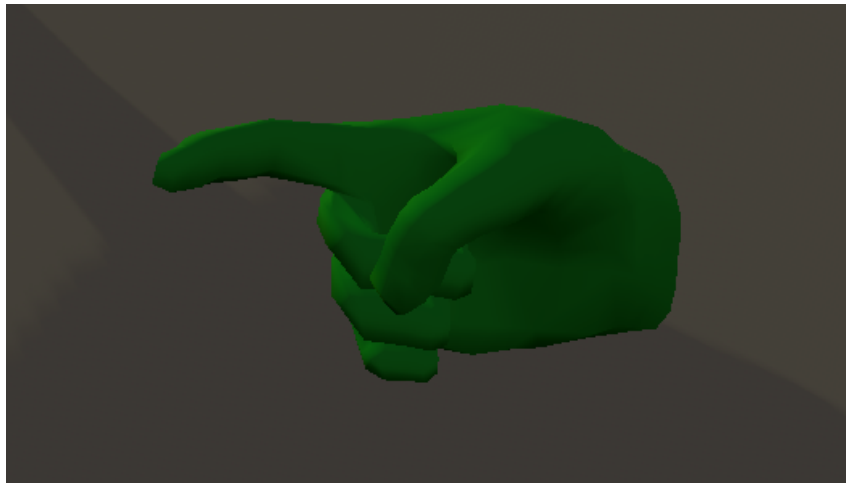
Usa o mesmo laser da seção de seleção discutida anteriormente ilustrado na Figura 5 e o mesmo botão também, ao apontar para o chão com ao menos uma unidade selecionada e apertar o botão, todas as unidades andam para a coordenada apontada no mundo e interagem com objetos compatíveis pelo caminho que percorrem.

5.2.3.2 Direto

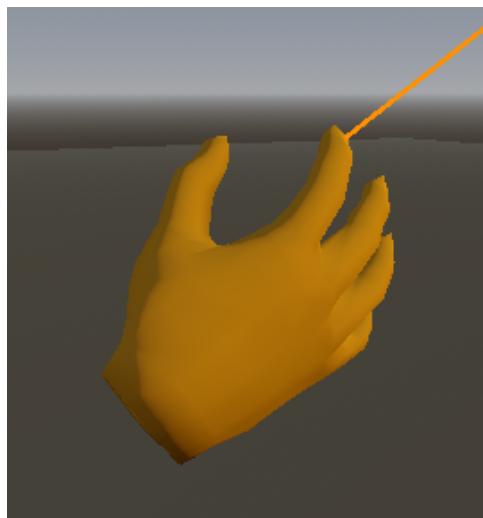
O movimento direto é realizado usando a combinação de segurar o botão *grip* do controle direito e mover o direcional analógico do mesmo controle, assim, todas as unidades selecionadas se movem instantaneamente de acordo com a posição do direcional, de forma semelhante a jogos em primeira ou terceira pessoa que utilizam este tipo de controle, que não é comum ao gênero *RTS*.

5.2.3.3 Traçar caminho

Também é possível acionar um segundo laser na mão direita do jogador indicado com a cor laranja ao apertar o botão que o ativa. Este é utilizado apontando para múltiplos pontos no chão em sequência escolhidos arbitrariamente pelo jogador, clicando em cada

Figura 6 – Modo de movimento direto

um de forma a traçar um caminho, indo de ponto em ponto em ordem. Ao soltar o botão que aciona o modo de traçar, todas as unidades atualmente selecionadas seguem o caminho traçado pelos pontos desenhados pelo jogador, relativo à posição inicial do próprio.

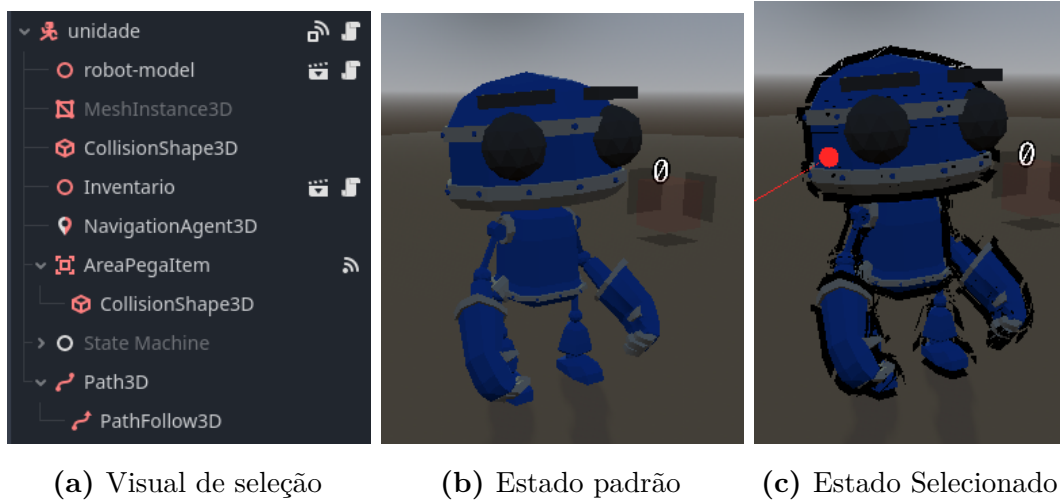
Figura 7 – Modo de movimento por caminho

5.2.4 Entidades principais

5.2.4.1 Unidade

A unidade é o principal tipo de objeto que o jogador tem controle sobre. Ela alterna entre estados de não selecionada e selecionada; caso esteja selecionada, pode receber “ordens” de movimento do jogador. O objeto é composto por componentes responsáveis por atribuir física dinâmica, colisão e movimentação pelo espaço através do algoritmo A*, implementado por padrão na *Godot Engine*. Além disso, também tem um inventário para guardar itens coletados do recurso ou entregues pelo jogador.

Figura 8 – Unidade



5.2.4.2 Recurso

O recurso é a única entidade que tem acesso ao próprio inventário com itens pré-alocados, além dos mesmos tipos de propriedades físicas estáticas da estrutura. Ao ser detectada no caminho de uma unidade, a mesma vai automaticamente pegar um item se houver um no inventário do recurso.



5.2.4.3 Estrutura

A estrutura é uma entidade simples, é composta apenas de elementos físicos estáticos e um inventário no qual unidades ou o jogador podem guardar itens. Ao ser detectada no caminho de uma unidade, a mesma vai automaticamente deixar um item se houver um em seu inventário.

5.2.4.4 Inventario

Inventários são “*containers*” capazes de segurar itens físicos que podem ser guardados ou pelo jogador ou pelas unidades, os itens em si são apenas objetos físicos contabilizados pelo inventário para o cumprimento de objetivos.

Figura 10 – Estrutura

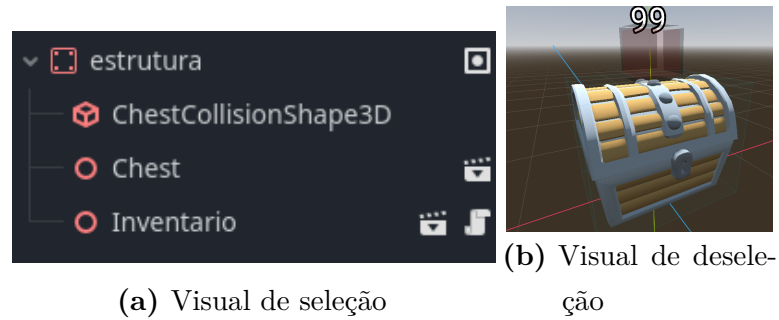
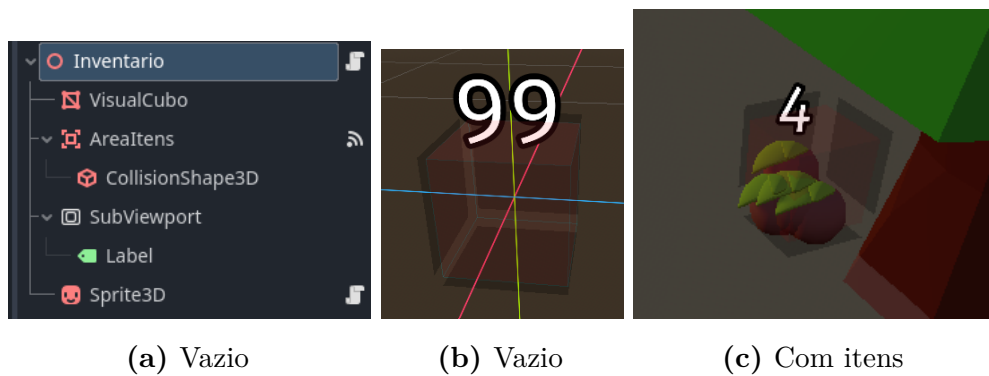


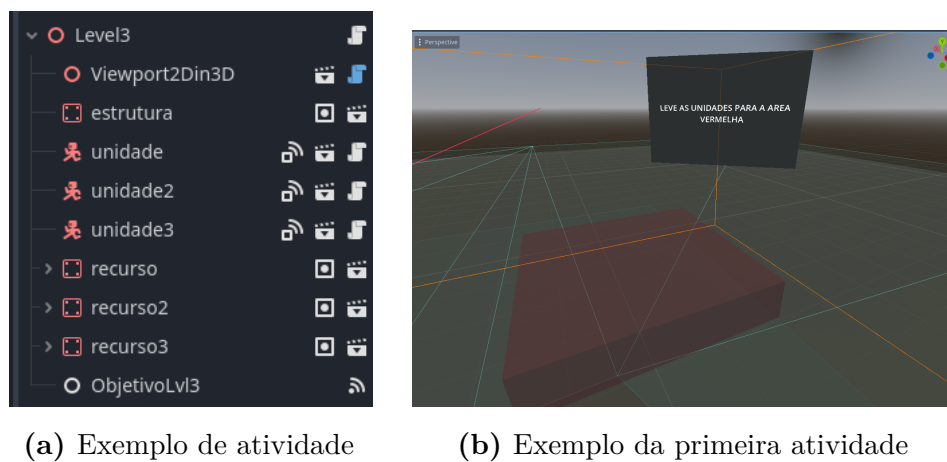
Figura 11 – Inventario



5.2.5 Objetivos

Cada uma das atividades tem um objetivo programaticamente associado que alerta o jogador quando é cumprido para que o experimento possa avançar. Cada objetivo tem suas próprias entidades e condição de encerramento.

Figura 12 – Objetivos



Com a implementação desses objetivos, buscava-se reproduzir de forma simples as ações mais básicas e comuns de jogos *RTS* identificadas no mapeamento.

Na primeira tarefa, os jogadores deveriam levar um dado número de unidades para uma área designada (ilustrada na figura 12b), replicando uma ação básica de comando em diversos jogos, como retornar tropas para a base em *Starcraft*.

Na segunda, os jogadores deviam pegar itens dos inventários de vários recursos; isso poderia ser feito tanto direcionando as unidades para os recursos para elas pegarem automaticamente ou o jogador poderia manualmente pegar os itens com a mão. Esta ação replica uma ação comum em diversos jogos que se trata de espalhar as unidades disponíveis e direcioná-las para lugares diferentes para obter recursos. Como por exemplo no jogo *Pikmin*, onde o jogador pode arremessar unidades em flores ou monstros que contêm recursos para produzir mais unidades.

Por fim na terceira tarefa, cuja hierarquia no *Godot* está ilustrada na figura 12a, haviam todas as entidades disponíveis, e a meta dada foi levar itens à estrutura disponibilizada; isso poderia ser feito tanto levando as unidades primeiro para os recursos e depois para a estrutura. Este tipo de ação de fazer unidades repetirem um percurso para acumular itens que o jogador pode usar também é uma ação básica quando se trata de seleção e movimentação de unidades. No jogo *Factory Town*, um processo muito comum é associar uma unidade a um recurso no mundo a ser obtido e a uma estrutura no mundo para a qual deve ser levado para realizar a coleta.

5.3 Realização dos Testes

Figura 13 – Fotos da execução do experimento



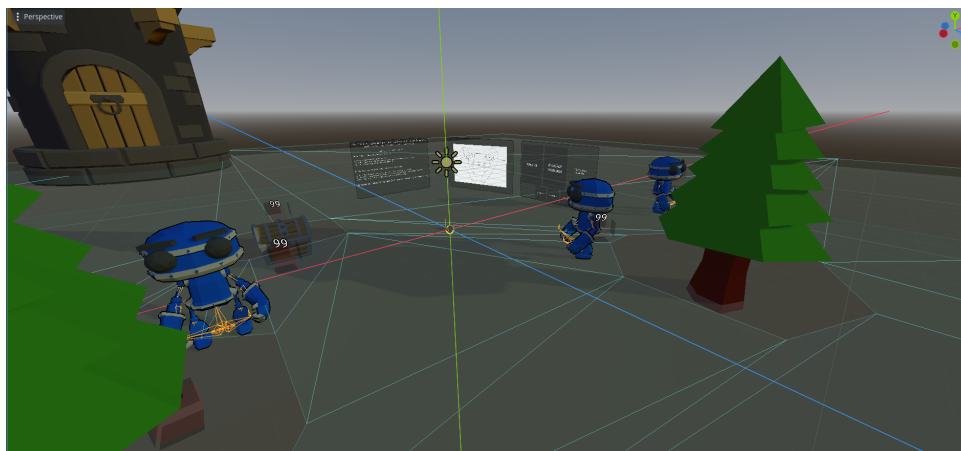
No Departamento de Computação da UFSCar de São Carlos foi montada uma bancada com os equipamentos necessários para o teste e foi aberto espaço para realização segura dos mesmos, visto que usar equipamento de realidade virtual requer que o usuário se mova fisicamente sem enxergar seus reais arredores. Os participantes assinavam

um termo de consentimento, vestiam o equipamento, recebiam orientações de como utilizar as mecânicas implementadas e então tentavam realizar as tarefas propostas durante intervalos de entre 10 e 20 minutos.

Ao colocar o equipamento, o usuário se deparava com o mundo mostrado na figura 14. Neste, haviam painéis flutuantes com instruções e *menus* para acessar as atividades ou reiniciar a aplicação caso houvesse algum problema. Além disso, haviam entidades de jogo arbitrariamente espalhadas sem nenhum objetivo inicial atrelado para que o participante pudesse se familiarizar com os controles e experiência em *VR*. Os jogadores tinham a liberdade de utilizar todas as mecânicas implementadas livremente sem nenhuma restrição.

Todos os usuários foram capazes de realizar as tarefas propostas sem grandes problemas, em algumas ocasiões certos *bugs* dificultaram a execução e havia dificuldade em orientar os participantes sem que o experimentador pudesse ver o que eles viam, mas nada foi considerado grave o suficiente para invalidar os testes. As gravações do ponto de vista de alguns dos experimentos estão disponibilizadas no Apêndice A caso haja necessidade de reconsiderar a validade dos testes ou realizar futuras pesquisas.

Figura 14 – Ambiente do protótipo na visão do editor da *Godot Engine*



Após terminarem e retirarem o equipamento, eram entrevistados pelo experimentador. Nesta, eram feitas diversas perguntas (disponíveis no Apêndice A) sobre a experiência do participante com jogos *VR*, jogos *RTS*, sua perspectiva sobre esses jogos, sua experiência com desenvolvimento de jogos e suas opiniões sobre cada uma das mecânicas presentes no protótipo.

Quanto a questões éticas, os únicos dados pessoais obtidos dos participantes foram seus nomes, idades, e vozes gravadas durante entrevistas. A coleção foi feita de forma consensual e apenas foi usada para organização dos dados. Ao final da pesquisa, as gravações de voz foram destruídas e os nomes foram retirados manualmente das transcrições para que os dados fossem suficientemente anonimizados. Além disso, os usuários foram informados sobre o equipamento de realidade virtual utilizado e os riscos que seu uso

pode acarretar antes de decidirem participar. O termo de consentimento utilizado está disponível no Apêndice A.

A primeira iteração do experimento foi realizada em junho de 2025 e contou com 21 participantes, nessa ocasião apenas duas das mecânicas de movimentação e duas das mecânicas de seleção haviam sido implementadas; a seleção por interface e movimentação por traçar-caminho ainda não haviam sido devidamente completadas. Os testes ocorreram ao longo de três dias com alunos do departamento e demais pessoas interessadas.

A segunda iteração ocorreu durante dois dias no mês seguinte na mesma localização e recebeu 12 participantes distintos da primeira. Para esta segunda ocasião, o protótipo foi devidamente incrementado e aperfeiçoado com *feedback* da iteração anterior, com todas as mecânicas implementadas e vários *bugs* corrigidos.

Para testar o protótipo com usuários, foi decidido buscar participantes no Departamento de Computação da UFSCar, tendo em vista que esse ambiente é frequentado por pessoas já interessadas por jogos digitais, dentre elas várias que desenvolvem jogos, assim fazendo com que o público-alvo da pesquisa pudesse abranger aqueles que ponderariam com mais profundidade as características das mecânicas.

Foi obtida as impressões de 33 participantes no total sobre as mecânicas apresentadas, seus prós/contras, intuitividade, qualidade, diversão e possibilidades de aplicação em outros jogos.

O objetivo das entrevistas era obter dados qualitativos das mecânicas implementadas: Quais aspectos os jogadores apreciavam, em que contextos mais faziam sentido, quais são suas qualidades e seus defeitos. Além disso, também procurou-se entender a relação dos participantes com jogos *RTS* e jogos *VR* para levar em conta ao se considerar suas opiniões sobre a experiência com o protótipo e sua visão sobre a viabilidade do que estava sendo testado.

5.4 Resultados

Após a execução dos testes e gravação de entrevistas, os dados obtidos sobre as mecânicas em cada teste foram analisados e, com isso, foram encontrados padrões nas perspectivas dos jogadores que apareceram com frequência.

Não houve espaço para obter mais de uma perspectiva sobre os dados obtidos, o que pode implicar em viés na avaliação dos resultados. Todas as transcrições e análises estão disponibilizadas (*link* no Apêndice A) para caso haja interesse em verificar ou reproduzir o processamento feito.

5.4.1 Processamento de Entrevistas

As gravações das entrevistas foram transcritas utilizando a ferramenta de código aberto *Vibe AI* que utiliza um pequeno modelo de processamento de linguagem para transformar arquivos de áudio em texto. Foram elaboradas várias perguntas de pesquisa com base no objetivo inicial e estas foram usadas para analisar o conteúdo das transcrições feitas das entrevistas por meio do método de codificação das respostas.

- **QP1** Quais aspectos de *RTS* são mais apreciados pelos jogadores?
- **QP2** Quais aspectos de VR são mais apreciados pelos jogadores?
- **QPs1.1** A opção de seleção por ponteiro/laser é adequada para seleção de unidades em jogos *RTS*?
- **QPs1.2** Quais são os pontos positivos da opção de seleção por ponteiro/laser de seleção de unidades?
- **QPs1.3** Quais são os problemas com a opção de seleção por ponteiro/laser?
- **QPs1.4** Quais foram as interações e dinâmicas proporcionadas pela opção de seleção por ponteiro/laser de seleção de unidades?
- **QPs2.1** A opção de seleção visual/por lente é adequada para seleção de unidades em jogos *RTS*?
- **QPs2.2** Quais são os pontos positivos da opção de seleção visual/por lente de seleção de unidades?
- **QPs2.3** Quais são os problemas com a opção de seleção visual/por lente de seleção de unidades?
- **QPs2.4** Quais foram as interações e dinâmicas proporcionadas pela opção de seleção visual/por lente de seleção de unidades?
- **QPs3.1** A opção de interface de mão de seleção de unidades é adequada para seleção de unidades em jogos *RTS*?
- **QPs3.2** Quais são os pontos positivos da opção de interface de mão de seleção de unidades?
- **QPs3.3** Quais são os problemas com a opção de interface de mão de seleção de unidades?
- **QPs3.4** Quais foram as interações e dinâmicas proporcionadas pela opção de interface de mão de seleção de unidades?

- **QPs0.1** Quais tarefas de *RTS* são mais adequadas para cada opção de seleção de unidades?
- **QPs0.2** Qual opção de seleção de unidades é mais atrativa para se explorar no desenvolvimento de um jogo?
- **QPm1.1** A opção 1 de movimentação de unidades por laser/ponteiro é adequada para movimentação em jogos *RTS*?
- **QPm1.2** Quais são os pontos positivos da opção de movimentação de unidades por laser/ponteiro?
- **QPm1.3** Quais são os problemas com a opção de movimentação de unidades por laser/ponteiro?
- **QPm1.4** Quais foram as interações e dinâmicas proporcionadas pela opção movimentação de unidades por laser/ponteiro?
- **QPm2.1** A opção de movimentação de unidades por controle direto é adequada para movimentação em jogos *RTS*?
- **QPm2.2** Quais são os pontos positivos da opção de movimentação de unidades por controle direto?
- **QPm2.3** Quais são os problemas com a opção de movimentação de unidades por controle direto?
- **QPm2.4** Quais foram as interações e dinâmicas proporcionadas pela opção de movimentação de unidades por controle direto?
- **QPm3.1** A opção de movimentação de unidades por traço de caminho é adequada para movimentação em jogos *RTS*?
- **QPm3.2** Quais são os pontos positivos da opção de movimentação de unidades por traço de caminho?
- **QPm3.3** Quais são os problemas com a opção de movimentação de unidades por traço de caminho?
- **QPm3.4** Quais foram as interações e dinâmicas proporcionadas pela opção de movimentação de unidades por traço de caminho?
- **QPm0.1** Quais tarefas de *RTS* são mais adequadas em cada opção de movimentação de unidades?
- **QPm0.2** Qual opção de movimentação de unidades é mais atrativa para se explorar no desenvolvimento de um jogo?

- **QPi1** A interação direta com itens é adequada em jogos *RTS*?
- **QPi2** Quais são os pontos positivos da interação direta com itens?
- **QPi3** Quais são os problemas com a interação direta com itens?
- **QPi4** A interação direta com itens é atrativa de se explorar no desenvolvimento de um jogo?

Não foram encontrados códigos suficientes para várias das perguntas para que pudessem ser considerados resultados relevantes, seguem os códigos considerados frequentes o suficiente (ao menos três ocorrências) para serem considerados notáveis.

5.4.2 Análise Quantitativa e Síntese

Os participantes demonstraram apreço pelos aspectos estratégicos e de gerenciamento de recursos típicos do gênero *RTS* (QP1), com frases como “Gosto de gerenciar tropas e alocar recursos” e “Complexo, mas gratificante”. No entanto, parte dos respondentes (24%) destacou a complexidade como uma barreira, especialmente para jogadores casuais (“Exige muita paciência”). Apenas dois participantes se declararam indiferentes ao gênero, justificando falta de familiaridade (“Não me cai muito”).

A imersão na realidade virtual (QP2) foi o ponto mais elogiado (57% das respostas), com descrições como “Extensão da realidade” e “Muito mais interativo que o esperado”. Contudo, questões de acessibilidade surgiram como limitações recorrentes: 33% mencionaram o alto custo do *hardware* (“Tecnologia promissora, mas inacessível”), e 19% relataram enjoos ou desconforto físico (“Dá tontura após um tempo”). Apesar disso, a maioria (81%) avaliou a experiência como positiva, sugerindo potencial para jogos *VR-RTS* bem otimizados.

A mecânica de apontar para unidades individuais (QPs1.1-QPs1.4) foi considerada a mais intuitiva por 87% dos participantes, com vantagens como “Precisão rápida” e “Não exige memorização de botões”. Apenas 13% tiveram dificuldades, relacionadas principalmente à curva de aprendizado (“Demorei a entender o controle”).

Embora útil para multi-seleção (75% das citações), a lente (QPs2.1-QPs2.4) foi descrita como “complexa” por 25% dos usuários, que a subutilizaram por falta de clareza (“Esqueci como ativar”). Sugestões incluíram melhorar o *feedback* visual (“Mostrar área de seleção mais claramente”).

Entre as mecânicas de seleção, a seleção por ponteiro foi a mais recomendada (60%), seguida pela lente (28%) e *UI* na mão (12%). A combinação das duas primeiras foi apontada como ideal para balancear precisão e eficiência.

Apesar de funcional, quanto à movimentação de unidades por laser (QPm1.1-QPm1.4) e bem acoplada a seleção também por laser, 29% dos usuários relataram problemas de precisão (“Difícil selecionar o ponto exato”) ou mal-estar (“Movimento repetitivo causa enjoos”).

A utilização do direcional analógico para controle direto das unidades (QPm2.1-QPm2.4) foi preferida por 82% dos participantes por ser “confortável e familiar”, especialmente para ajustes finos (“Microajustes com analógico são perfeitos”).

O método de traçar caminho (QPm3.1-QPm3.4) foi pouco utilizado (apenas 18% o exploraram), foi criticado por falta de intuitividade (“Não sabia como ativar”) e ausência de *feedback* visual (“Faltou representação do caminho traçado”).

O controle direto (*joystick*) foi considerado a forma mais popular de movimento (53%), seguido por apontar no chão (33%) e traço de caminho (14%).

Apesar de elogiada por sua imersão (“Agachar para pegar frutas foi divertido”), a interação direta com os itens apresentou *bugs* frequentes: 40% dos usuários relataram problemas como “itens flutuando” ou “dificuldade em soltar objetos”.

5.4.3 Análise Final

O estudo, apesar de limitado em escopo, revelou que a combinação de mecânicas tradicionais de *RTS* com interações imersivas de *VR* tem potencial, desde que equilibrada com usabilidade e acessibilidade. A implementação dessas melhorias pode tornar jogos *VR-RTS* mais atraentes para um público amplo.

A Seleção por laser destacou-se como a mecânica mais intuitiva e de fácil adoção pelos usuários. Sua principal vantagem reside na simplicidade de operação e na integração natural com o sistema de movimentação por laser. Entretanto, esta abordagem apresenta limitações significativas, particularmente na seleção de múltiplas unidades simultaneamente e na exigência de precisão na pontaria, o que pode representar um desafio para usuários menos experientes.

A seleção visual emergiu como solução eficiente para seleção em massa, operando diretamente através do campo visual do jogador. Embora ofereça rapidez na seleção de múltiplas unidades, esta mecânica mostrou-se complexa em termos de usabilidade, exigindo um conhecimento prévio dos controles em *VR* e apresentando dificuldades na comunicação intuitiva de seu funcionamento aos usuários.

A seleção por interface apresentou-se como alternativa eficaz para unidades fora do campo visual imediato do jogador. Apesar de sua interface compreensível, esta abordagem foi prejudicada pela complexidade de acesso aos controles e pela dificuldade em correlacionar informações textuais com o ambiente tridimensional do jogo.

A movimentação por laser manteve a consistência com a seleção de laser, oferecendo uma experiência coesa e intuitiva. Contudo, suas limitações se manifestaram na restrição a comandos unitários e na mesma exigência de precisão que caracteriza o sistema de seleção correspondente.

A movimentação direta destacou-se pela abordagem mais imersiva e amigável ao usuário, minimizando a necessidade de movimentos complexos ou mira precisa. Paradoxalmente, esta mecânica compartilhou com a seleção visual as mesmas dificuldades na transmissão intuitiva de seu funcionamento e na exigência de familiaridade com controles *VR*.

A movimentação por caminho traçado demonstrou potencial para execução de manobras complexas e ofereceu sensação de controle refinado. No entanto, sua utilização mostrou-se como a mais desafiadora, tanto em termos de complexidade de uso quanto na dificuldade inerente ao traçado de rotas em espaço tridimensional.

Foi observada uma relação inversamente proporcional entre intuitividade e funcionalidade avançada em todas as mecânicas avaliadas. Enquanto sistemas baseados em laser oferecem maior acessibilidade inicial, abordagens mais complexas como a seleção visual e movimentação por caminho apresentam potencial para operações avançadas, embora à custa de uma curva de aprendizado mais elevada.

As opiniões dos jogadores a respeito do protótipo foram unanimemente positivas, implicando que de fato há potencial valor na adaptação de jogos *RTS* para realidade virtual. As tarefas propostas para o experimento podem ter sido excessivamente simples e não testado as mecânicas, em situações de estresse, mas salvo casos de *bugs* com o protótipo todos os jogadores conseguiram realizar as tarefas propostas.

5.4.4 Síntese dos Resultados

Para facilitar a comparação e análise das mecânicas implementadas, seguem tabelas resumindo as características principais encontradas em cada uma.

Tabela 1 – Comparação de mecânicas de seleção implementadas no protótipo

Mecânica de Seleção	Vantagens	Desvantagens
Laser	<ul style="list-style-type: none"> + Mais Intuitiva; + Encaixava com movimentação; 	<ul style="list-style-type: none"> – Não era possível selecionar várias unidades rapidamente; – Precisão de pontaria era necessária;
Lente	<ul style="list-style-type: none"> + Mais Rápida; + Depende apenas do olhar; 	<ul style="list-style-type: none"> – Pouco controle; – Usuários raramente sentiram necessidade de utilizar;
Interface	<ul style="list-style-type: none"> + Não é preciso ver as unidades para selecionar; + Usuários apreciaram interface diegética 	<ul style="list-style-type: none"> – Menos intuitiva; – Era difícil diferenciar as unidades pela interface

Tabela 2 – Comparação de mecânicas de movimentação implementadas no protótipo

Mecânica de Movimentação	Vantagens	Desvantagens
Laser	<ul style="list-style-type: none"> + Mais Intuitiva; + Encaixava com seleção; 	<ul style="list-style-type: none"> – Precisão de pontaria era necessária; – Fácil clicar acidentalmente em lugar errado;
Direto	<ul style="list-style-type: none"> + Jogadores apreciam a sensação tátil de controle; + Fácil de fazer movimentos pequenos e precisos; 	<ul style="list-style-type: none"> – Combinação de botões difícil; – Requer <i>input</i> constante;
Traçar Caminho	<ul style="list-style-type: none"> + Permite movimentos mais complexos; + Permite automatização de tarefas; 	<ul style="list-style-type: none"> – Menos intuitiva; – Não havia como sempre prever o caminho com clareza;

6 Conclusão

Com base nos resultados obtidos, pode-se afirmar que o objetivo da pesquisa de explorar as possibilidades de implementação de mecânicas do gênero *RTS* em um jogo *VR* teve sucesso: foi criada uma base que pode ser reutilizada para o desenvolvimento de jogos desta categoria que comprovadamente causou uma boa impressão nos usuários.

Foi possível criar três mecânicas de seleção de unidades e três mecânicas de movimentação de unidades e validar sua funcionalidade, pontos positivos e pontos negativos na visão dos usuários.

6.1 Trabalhos Futuros

As limitações encontradas durante esta pesquisa abrem caminho para diversas investigações complementares que podem aprimorar e expandir os resultados obtidos.

Embora as mecânicas implementadas tenham sido validadas, algumas delas — especialmente as introduzidas em fases posteriores do desenvolvimento — demandam testes adicionais para otimização. Estudos futuros podem conduzir análises comparativas mais aprofundadas. Além disso, seria valioso investigar como ajustes de *feedback* visual e sonoro podem melhorar a usabilidade, especialmente em cenários complexos com múltiplas unidades simultâneas.

Esta pesquisa focou na movimentação e seleção de unidades, mas jogos *RTS* envolvem outras mecânicas essenciais, como construção de bases, gerenciamento de recursos e combate tático. Futuros trabalhos podem explorar como essas dinâmicas podem ser traduzidas para *VR*, mantendo a imersão sem sacrificar a eficiência. Além disso, gêneros como estratégia por turnos, simuladores de gestão ou até mesmo *MOBA*s em *VR* poderiam se beneficiar das lições aprendidas nesta investigação, ampliando o escopo de aplicação. Um modo de jogo que foi desenvolvido para o protótipo, mas não chegou a ser utilizado na pesquisa, por exemplo, é o modo de “visão aérea”, no qual a escala do ambiente do jogo é diminuída e a perspectiva do jogador se torna mais semelhante à visão da câmera de um jogo *RTS* tradicional.

Os testes realizados focaram em tarefas controladas, mas cenários de alta pressão — como batalhas em larga escala, contra relógio ou com recursos limitados — podem expor limitações não detectadas, assim permitindo a avaliação de como as mecânicas se comportariam sob estresse físico e cognitivo, com a demanda por tomada de decisão rápida dos usuários. Isso também permitiria investigar se a fadiga física (comum em sessões prolongadas de *VR*) impacta a eficácia das interações propostas.

Referências

- APPERLEY, T. H. Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, v. 37, n. 1, p. 6–23, 2006. Disponível em: <<https://doi.org/10.1177/1046878105282278>>. Citado na página 9.
- AZOFEIFA, J. D. et al. Systematic review of multimodal human–computer interaction. *Informatics*, v. 9, n. 1, 2022. ISSN 2227-9709. Disponível em: <<https://www.mdpi.com/2227-9709/9/1/13>>. Citado na página 11.
- CAMARA.LEG.BR. *Entra em vigor marco legal da indústria de jogos eletrônicos*. 2024. <<https://www.camara.leg.br/noticias/1058707-entra-em-vigor-marco-legal-da-industria-de-jogos-eletronicos/>>. Acesso 01/07/2025. Citado na página 9.
- GAMESPOT. *The Genesis of Real-Time Strategy*. 2001. <https://web.archive.org/web/20110721213420/http://www.gamespot.com/gamespot/features/all/real_time/p2_01.html>. Acesso em 10/07/2025. Citado na página 16.
- LINK, S. et al. An intelligent multimodal mixed reality real-time strategy game. In: *2016 IEEE Virtual Reality (VR)*. [S.l.: s.n.], 2016. p. 223–224. Citado na página 11.
- MALÝ, J. *Real-time strategy videogame toolkit for Godot Engine*. Dissertação (Mestrado) — Charles University, Faculty of Mathematics and Physics, Department of Software and Computer Science Teaching, 2024. Disponível em: <<https://dspace.cuni.cz/handle/20.500.11956/193210>>. Citado na página 11.
- MU MANN, M.; TRUMAN, S.; MAMMEN, S. von. Game-ready inventory systems for virtual reality. In: *2021 IEEE Conference on Games (CoG)*. IEEE Press, 2021. p. 1–8. Disponível em: <<https://doi.org/10.1109/CoG52621.2021.9619028>>. Citado na página 11.
- PHILLIPS, K.; PIEKARSKI, W. Possession techniques for interaction in real-time strategy augmented reality games. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA: Association for Computing Machinery, 2005. (ACE '05), p. 2–es. ISBN 1595931104. Disponível em: <<https://doi.org/10.1145/1178477.1178584>>. Citado na página 11.
- PIASKIEWICZ, M. *Level Design Principles - Familiarity*. 2019. <<https://www.gamedeveloper.com/design/level-design-principles---familiarity>>. Acesso em 10/07/2025. Citado na página 22.
- SANTUCCI, F. et al. An immersive open source environment using godot. In: GERVASI, O. et al. (Ed.). *Computational Science and Its Applications – ICCSA 2020*. Cham: Springer International Publishing, 2020. p. 784–798. ISBN 978-3-030-58820-5. Citado na página 11.
- TORONTO, N. *The Future Of The Real-Time Strategy Game*. 2008. <<https://www.gamedeveloper.com/design/the-future-of-the-real-time-strategy-game>>. Acesso em 01/07/2025. Citado na página 17.

TRUMAN, S. et al. Rethinking real-time strategy games for virtual reality. In: *Proceedings of the 13th International Conference on the Foundations of Digital Games*. New York, NY, USA: Association for Computing Machinery, 2018. (FDG '18). ISBN 9781450365710. Disponível em: <<https://doi.org/10.1145/3235765.3235801>>. Citado 2 vezes nas páginas 9 e 11.

VLAHOVIC, S.; SUZNJEVIC, M.; SKORIN-KAPOV, L. A framework for the classification and evaluation of game mechanics for virtual reality games. *Electronics*, v. 11, n. 18, 2022. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/11/18/2946>>. Citado 2 vezes nas páginas 9 e 12.

WINSTANLEY, C. *The Making of Dune II The birth of the real-time strategy game*. 2023. <<https://web.archive.org/web/20230227071053/https://readonlymemory.vg/the-making-of-dune-ii/>>. Acesso em 10/07/2025. Citado na página 16.

APÊNDICE A – Links Importantes

- Repositório do projeto produzido e utilizado nos experimentos - <<https://github.com/Yamartim/framework-vr-rts>>
- Consulta no banco de dados da *Steam* de jogos *VR* e *RTS* - <<https://steamdb.info/instantsearch/?refinementList%5Btags%5D%5B0%5D=RTS&refinementList%5BhardwareCategories%5D%5B0%5D=VR%20Only>>
- Planilha de mapeamento de jogos da *Steam* - <https://docs.google.com/spreadsheets/d/1LwIeZk-gvHTdQsn7Y0FWc-TuNVe1T3SE2suHsKPMB4k/edit?usp=drive_link>
- Formulário de consentimento de participação no experimento - <https://docs.google.com/document/d/1kvflOfA_Z96TnJNrNnI4xRxfeL4iiVuxiugi09TihQ8/edit?usp=drive_link>
- Planilha de experimentos realizados com transcrições das entrevistas - <https://docs.google.com/spreadsheets/d/1knPrW96mYeKIkMacUa_3LDNtwpQaadExfJT-ZpK96lY/edit?usp=sharing>
- Lista de perguntas realizadas nas entrevistas - <https://docs.google.com/document/d/1UmOBKrKXhgDkoHgNngJqB6KcX4oj_KF83oq9O6ZIXdI/edit?usp=drive_link>
- Planilha de codificação: Experimentos x Perguntas de Pesquisa - <https://docs.google.com/spreadsheets/d/1g0Q_JpfTtrbSrev498p-zH7_MRzVcrDBsKciSBPV7bo/edit?usp=drive_link>
- Pasta com vídeos de experimentos gravados <<https://drive.google.com/drive/folders/1ibgcmQxCOBPM0U8X5gHx3WoqlxLg4n1K>>