



## **DOCUMENTAÇÃO COMPLETA – GAME NERITH**

Desenvolvimento de Jogos Digitais\_Eng.Soft07

### **Orientador e Product Owner**

Professor: Fabrício Dias

### **Grupo de desenvolvimento (abordagem ágeis)**

Angélica Gomes da Silva (Dev.) - [angelik.gomes27@hotmail.com](mailto:angelik.gomes27@hotmail.com)

Caio Cezar Jotta Nogueira (Dev.) - [caiojotta18@gmail.com](mailto:caiojotta18@gmail.com)

Denis Gomes Bomfim (Dev.) – [denisgomesbomfim@hotmail.com](mailto:denisgomesbomfim@hotmail.com)

Luan Santos Brandão da Silva (Scrum Master) - [brandaoluan506@gmail.com](mailto:brandaoluan506@gmail.com)

Marcos Luiz Sousa Reis (Project Manager) – [marcossouzareis@gmail.com](mailto:marcossouzareis@gmail.com)

Nícolas Vítor Carvalho de Oliveira (Dev.) – [vcarvalho797@gmail.com](mailto:vcarvalho797@gmail.com)

Paulo Victor Melo Da Silva Ferreira (Dev.) – [Paaulomsf@gmail.com](mailto:Paaulomsf@gmail.com)

Yago da Costa Jardim Alves Braga - [yagoniuyagoniu@gmail.com](mailto:yagoniuyagoniu@gmail.com)

Ano 2025

## PROJETO - GAME NERITH 1.0

Esta é uma documentação oficial do projeto **Game NERITH**, desenvolvido pelo **Time de graduandos em Engenharia de Software** no contexto da disciplina **Laboratório de Jogos Digitais**, ministrada pelo Professor Fabrício Dias na Universidade de Vassouras, Campus Maricá. O projeto foi conduzido com base em práticas das **Metodologias Ágeis**, utilizando o framework **Scrum** para organização das Sprints e divisão das tarefas. As ferramentas e tecnologias adotadas foram:

- Engine de desenvolvimento: **Unity**.
- Linguagem de programação: **C#**.
- Gerenciamento de tarefas: **Trello**.
- Ambiente de comunicação e reuniões: **Google Meet**.
- Repositório de versionamentos **GitHub**

Nossa equipe seguiu uma abordagem iterativa e incremental, com foco em entregas funcionais a cada ciclo semanal de desenvolvimento (Sprints), promovendo a colaboração contínua e a evolução do produto com base no feedback da própria equipe e testes.

## **AGRADECIMENTOS**

A equipe de desenvolvimento do **Game NERITH** agradece ao nosso **Product Owner, Professor Fabrício Dias**, por sua orientação, apoio constante e liderança ao longo de todas as etapas do projeto. Sua atuação foi essencial para que pudéssemos transformar ideias iniciais em um produto funcional e significativo.

Estendemos nossos agradecimentos a todos os **docentes do curso de Engenharia de Software**, cujos ensinamentos e dedicação foram determinantes para a consolidação dos conhecimentos técnicos e para a superação dos inúmeros desafios enfrentados durante o desenvolvimento deste trabalho.

Reconhecemos ainda a importância da atuação da **Coordenação do curso de Engenharia de Software**, bem como de toda a equipe **administrativa e de suporte da Universidade de Vassouras**, que contribuíram de forma direta e indireta para a viabilização de mais um projeto em nossa jornada acadêmica e profissional.

## **DESAFIOS - Projeto Game NERITH**

### **RESUMO**

O projeto Game NERITH representa um marco no percurso formativo da equipe de estudantes do curso de Engenharia de Software, desenvolvedores iniciantes no universo dos jogos digitais. Desde a proposta inicial apresentada pelo professor e Product Owner Fabrício Dias, até a entrega do Produto Mínimo Viável (MVP), o desenvolvimento foi marcado por desafios técnicos, intensa curva de aprendizado e fortalecimento do trabalho em equipe. Com o apoio contínuo do professor orientador, que introduziu a engine Unity e acompanhou de perto todas as etapas do projeto, a equipe teve liberdade para escolher sua plataforma e estruturação. A aplicação de práticas ágeis, uso de C#, controle de versões com GitHub, organização com Trello e comunicação remota via Google Meet foram essenciais para a coordenação das tarefas e evolução do trabalho, todas devidamente registradas em atas e organizadas na pasta de documentação do projeto. Optando por um jogo em 2D como entrega viável dentro do tempo e das capacidades disponíveis, a equipe desenvolveu um sistema funcional com lógica de jogo, controle de fases, HUD e comportamento de inimigos, implementando conceitos como modularização, reaproveitamento de assets e boas práticas de codificação.

NERITH não é apenas um jogo: é o reflexo do engajamento de uma equipe que, sob orientação competente e com espírito colaborativo, foi capaz de transformar uma ideia acadêmica em um produto digital acessível, funcional e com potencial de expansão e evolução.

## Sumário

<b>AGRADECIMENTOS .....</b>	<b>3</b>
<b>Casos de Teste Realizados: .....</b>	<b>6</b>
<b>Padrões e Arquitetura .....</b>	<b>9</b>
<i>Fluxo de Execução do Jogo NERITH .....</i>	<i>10</i>
<i>Estrutura Assets .....</i>	<i>11</i>
Conclusão .....	12
<i>Objetivo do Jogo Nerith: .....</i>	<i>12</i>
<i>Requisitos: .....</i>	<i>12</i>
<i>Comandos (Jogabilidade): .....</i>	<i>13</i>
<i>Como jogar: .....</i>	<i>13</i>
<i>Telas do sistema: .....</i>	<i>13</i>
<b>Apresentação: .....</b>	<b>13</b>
<i>Perfil do usuário: .....</i>	<i>13</i>
<i>Tela inicial (Main Menu) - Elementos: .....</i>	<i>13</i>
<i>Fases jogáveis (Ex: savePoint.unity, Fases.unity) - Navegação: .....</i>	<i>14</i>
<i>Interface (HUD) - Elementos visíveis: .....</i>	<i>14</i>
<i>Funcionalidades gerais (sistema de checkpoint automático): .....</i>	<i>14</i>
<i>Níveis de acesso e perfis de usuário: .....</i>	<i>14</i>
<i>Recomendações: .....</i>	<i>14</i>
<b>Conclusão .....</b>	<b>14</b>
<i>Pré-Requisitos (mínimos para execução do jogo): .....</i>	<i>15</i>
<i>Para execução correta do jogo, recomenda-se .....</i>	<i>15</i>
<i>Passos para a instalação e execução, local (desenvolvedores) .....</i>	<i>15</i>
<b>Passo a passo (Configuração do ambiente de desenvolvimento com Unity e WebGL): ..</b>	<b>16</b>
<i>Problemas comuns (Soluções para possíveis erros de execução): .....</i>	<i>16</i>
<i>Recomendações: .....</i>	<i>16</i>
<i>Este documento pode ser complementado conforme a evolução e o .....</i>	<i>16</i>
<i>Execução em ambiente Web (usuário final) .....</i>	<i>16</i>
<i>Para executar o jogo via navegador: .....</i>	<i>16</i>
<b>Suporte: .....</b>	<b>17</b>
Conclusão – Manual de Instalação .....	17

## **Manual Técnico – Game NERITH**

Este manual técnico apresenta a documentação detalhada do projeto NERITH, abordando sua arquitetura de software, estrutura interna, padrões de codificação, tecnologias utilizadas e os testes realizados para garantir a confiabilidade e o funcionamento adequado da aplicação. Este documento destina-se a desenvolvedores, mantenedores e profissionais técnicos que desejam compreender, manter ou expandir o sistema no futuro. O projeto NERITH foi desenvolvido com base em tecnologias modernas e boas práticas de engenharia de software, e seu código-fonte está disponível em repositório público no GitHub: <https://github.com/2000Paulo/GameNerith>. A seguir, apresenta-se um panorama das principais tecnologias e componentes adotados no desenvolvimento do jogo:

### **Engine de Desenvolvimento:**

Utilizou-se a Unity, com suporte ao Universal Render Pipeline (URP), motor gráfico amplamente consolidado no setor e responsável por cerca de 79% de domínio no mercado global de engines para jogos digitais.

### **Linguagem de Programação:**

O jogo foi inteiramente programado em C#, seguindo a API de Scripting da Unity, com separação clara de responsabilidades entre os scripts e aderência aos princípios de Clean Code.

### **Sistema de Entrada (Input):**

Implementado com o Unity Input System, permitindo controle refinado e personalizável para teclado, mouse ou gamepad.

### **Pipeline de Renderização:**

Utilização da Universal Render Pipeline (URP) para garantir alta performance e compatibilidade gráfica com múltiplas plataformas.

### **Tipo de Software:**

A versão atual do projeto é uma aplicação web, exportada para WebGL, executável diretamente em navegadores modernos sem necessidade de instalação local.

### **Casos de Teste Realizados:**

#### **✓Verificação das Mecânicas Principais**

- Movimentação fluida (WASD/setas), pulo e ataque com barra de espaço
- Detecção de colisão com inimigos, obstáculos e plataformas

#### **✓Testes de Vida e Combate**

- Atualização em tempo real da barra de vida ao sofrer dano
- Reinício do jogo e zeramento do score em caso de morte (modo roguelite)

- Validação da opção "Tentar novamente" com reinício a partir da última fase

### ✓Transição e Progressão

- Funcionamento correto dos portais invisíveis entre fases
- Checkpoints ativados automaticamente ao final de cada estágio
- Transição visual e lógica entre cenas (Scenes) no Unity

### ✓Telas e Menus

- Exibição da tela de **Game Over** com botões funcionais ("Tentar novamente" e "Menu Principal")
- Teste da tecla "P" com pausa narrativa (caixa de diálogo)
- Teste da tecla "ESC" com menu funcional (retorno, comandos e controle de volume)

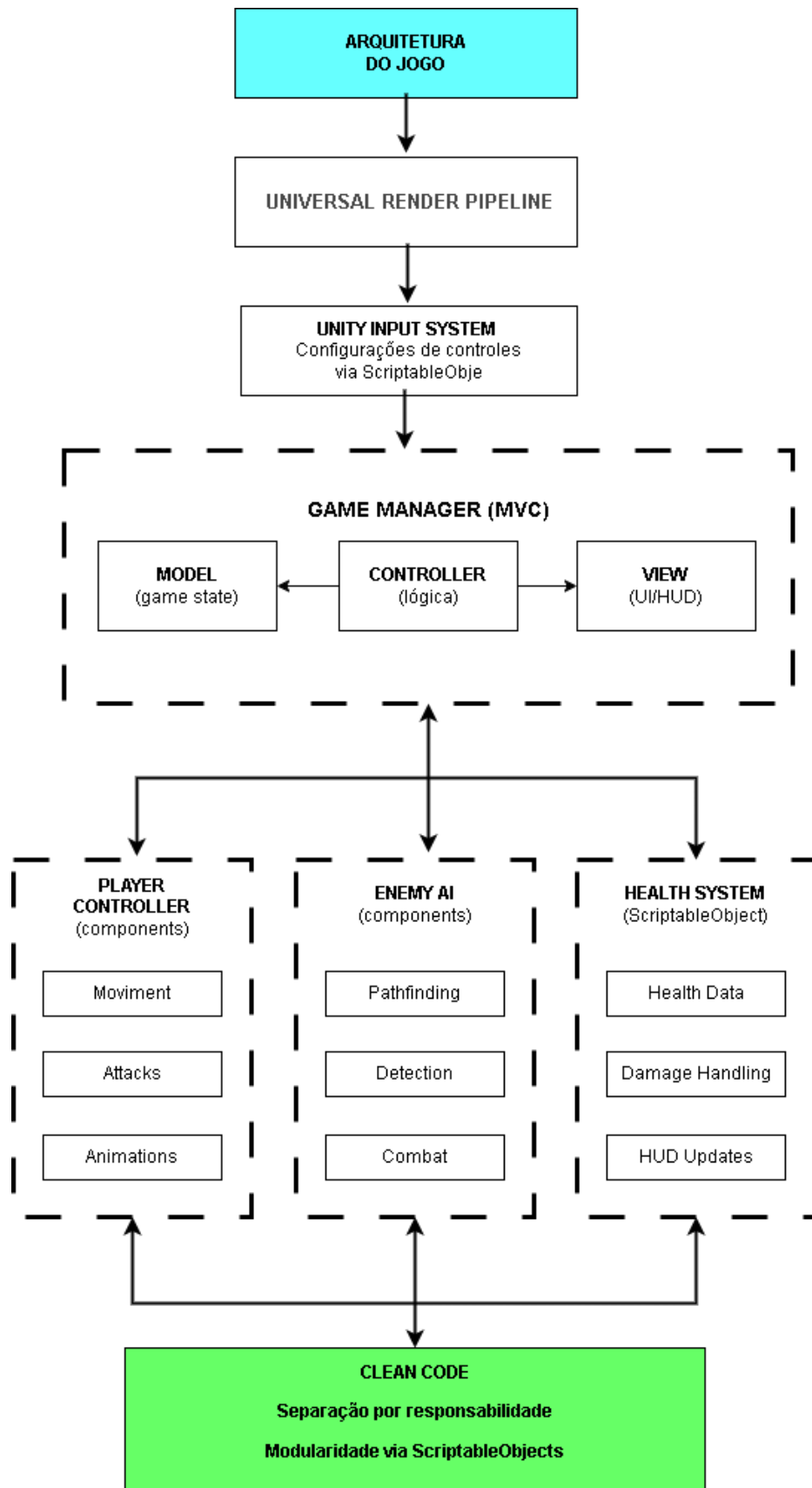
### ✓Executável Testado (.exe)

- Desempenho estável em diferentes configurações de hardware (Windows)
- Áudio, animações e HUD renderizados corretamente na versão final do build

## ARQUITETURA DO SOFTWARE

O projeto utiliza o padrão de arquitetura baseada em Componentes (Unity) e MVC (Model-View-Controller) nas lógicas de scripts.

A separação clara entre lógicas visuais (View), controle (Controller) e dados (Model) permite uma melhor e maior manutenibilidade.





## Padrões e Arquitetura

**GameManager:** Gerencia os estados globais do jogo (como início, pausa, derrota) e as transições entre cenas. Centraliza o fluxo da aplicação com base no padrão **MVC**.

**HealthSystem:** Sistema de vida modular e reutilizável, implementado com uso de **ScriptableObject**, permitindo facilmente ajuste e reaproveitamento em diferentes personagens.

**EnemyAI:** Comportamento básico de IA para inimigos, com lógica de perseguição e ataque programada em C#, associada a colisores e triggers para interação.

**ScoreSystem:** Gerencia a pontuação do jogador com base em tempo, número de inimigos derrotados e dano recebido. Responsável também pela comunicação com o sistema de ranking interno.

**Fases:** Estruturadas de forma modular com uso de **Prefabs**, permitindo facilmente adição, replicação ou ajuste de cenários e obstáculos.

**PlayerController:** Responsável por controlar todas as ações do personagem NERITH, incluindo movimentação (WAD/setas), pulo, ataque (barra de espaço) e interação com a física do ambiente. Integra animações, colisões e resposta ao input com precisão, promovendo fluidez à jogabilidade.

## Principais características do fluxograma:

### 1. Hierarquia de Dependências:

- URP como base de renderização
- Input System como camada de entrada
- GameManager como núcleo MVC

### 2. Componentes Principais:

- PlayerController (combina componentes de movimento, ataque e animação)
- EnemyAI (lógica de IA simplificada)
- HealthSystem (modular, baseado em ScriptableObject)

### 3. Padrões Destacados:

- MVC híbrido no GameManager
- Componentes para comportamentos específicos
- ScriptableObjects para dados compartilhados

#### **4. Fluxo de Dados:**

- Setas indicam direção das dependências
- Comunicação centralizada via GameManager
- Este diagrama mostra a separação de responsabilidades e a modularidade através dos ScriptableObjects, conforme especificado na arquitetura proposta padrão MVC.

#### **Pastas:**

- Scenes: contém as fases e ambientes do jogo (.Unity)
- Scripts: lógicas do jogador, inimigos, HUD, GameManager etc.
- Player, Enemies, Barra\_de\_Vida: assets e controladores
- Resources: carregamento dinâmico
- ProjectSettings: configurações do projeto

#### **Códigos-chave:**

- PlayerController.cs: movimentação, pulo, ataque
- EnemyAI.cs: lógica básica de inimigos
- HealthSystem.cs: barra de vida e sistema de dano
- GameManager.cs: controle da lógica global de fases

#### **Testes realizados:**

- Testes manuais e automatizados com Unity Test Runner.
- Casos de teste de movimentação, detecção de colisão, interação com objetos, salvamento de progresso, HUD e ataques.

#### **Fluxo de Execução do Jogo NERITH**

Flowchart TD

A[Início do Jogo] --> B[Menu Principal\n (Main Menu Unity) ]

B --> C [Fase Inicial\n (jogo começa após botão iniciar) ]

C --> D[Checkpoint 1\n (transição automática) ]

D --> E[Próxima Fase\n (desafio aumentado) ]

E --> F[Checkpoint 2\n (transição automática) ]

F --> G[Fase Avançada\n (dificuldade elevada) ]

G --> H[Final do Jogo\n→ Game Over ou Vitória]

H -->|Tentar Novamente| G

H -->|Menu Principal| B

C --> I [se NERITH morrer → Tela de Game Over]

E --> I

G --> I

I -->|Tentar Novamente| última Fase [Última Fase Jogada]

I -->|Menu Principal| B

Subgraph “Pausas e Interações”

P1[“Tecla 'P': Caixa de Diálogo\n→ Mensagem Estética e Narrativa”]

P2[“Tecla 'ESC': Menu Pausado\n→ Voltar ao Jogo\n→ Lista de Comandos\n→ Menu Principal\n→ Controle de Volume”]

End

C --> P1 & P2      E --> P1 & P2      G --> P1 & P2

## Estrutura Assets

Assets/

|— BarraDeVida/=> HUD com barra de vida do jogador(HealthSystem.cs)

|— Player/ => Sprites, animações e PlayerController.cs

|— Scripts/ => lógica principal (GameManager.cs, EnemyAI.cs, ScoreSystem.cs, etc.)

|— Scenes/ => Cenas jogáveis, tela inicial, fase 1, fase 2 e tela de derrota

|— Resources/ => Prefabs, áudios, imagens e ScriptableObjects

|— Monster\_Creatures/ => Sprites e IA dos inimigos (Inimigos por fase)

|— craftpix-\* => Tilesets e assets visuais das fases (pixel art)

\*Plataforma .net especializada em Assets confiáveis

Através do link do Repositório GitHub outro desenvolvedor terá acesso ao código e toda documentação, na ocasião em que for implementar a manutenção no programa e precisar fazer qualquer alteração no código. Este profissional, no entanto, pode não ter participado do projeto ou ser novo na empresa. Então é só seguir o link: [colocar aqui o link do GitHub]

## Conclusão

Este manual técnico apresenta de forma clara a estrutura e os aspectos funcionais do **Game NERITH**, desenvolvido com base em uma arquitetura híbrida (Componentes + MVC) integrada à Unity Engine (URP e Input System). O projeto priorizou modularidade, escalabilidade e desempenho, com códigos organizados, reutilizáveis e aderentes às boas práticas do **Clean Code**.

A estrutura foi planejada com divisão de responsabilidades, uso de **ScriptableObjects**, versionamento via **GitHub**, organização ágil no **Trello** e colaboração remota contínua. Scripts principais como **GameManager.cs**, **PlayerController.cs**, **EnemyAI.cs** e **HealthSystem.cs** foram testados com foco em robustez, fluidez e usabilidade.

Os testes funcionais e exploratórios confirmaram o bom desempenho do MVP, validando mecânicas, HUD, transições e o executável final. Este documento registra não só o aspecto técnico do projeto, mas também representa um marco na jornada de aprendizado da equipe, sendo referência para manutenções e futuras expansões.

## MANUAL DO USUÁRIO

### Objetivo do Jogo Nerith:

Percorrer as fases do mundo de Nerith e fazer o maior número de pontos possíveis e no menor tempo, enfrentando e eliminando inimigos diversos em um ambiente 2D com arte em pixel art.

### Requisitos:

- Sistema: Windows 10 ou superior
- RAM: 4GB
- GPU: Intel HD ou superior

### **Comandos (Jogabilidade):**

- A/D: mover(esquerda/direita)
- W: pular
- Espaço: atacar
- P: Abre caixa de diálogo e pausa momentaneamente – iteragir
- ENTER: Fecha a caixa de diálogo e retorna para a fase onde pausou
- ESC: Abre menu com três botões - Menu Principal, Tentar Novamente, Voltar.

### **Como jogar:**

1. Inicie o jogo pela cena principal `savePoint.unity`.
2. Explore o mapa, enfrente inimigos e colete itens.
3. Use os pontos de salvamento para registrar progresso.

### **Telas do sistema:**

- Tela Inicial: botão iniciar, configurações e sair
- Tela de Fase: HUD de vida, barra de energia, ações
- Tela de Pausa/Menu: opções de salvar, carregar, sair
- Tela de Game Over: opção de reiniciar ou voltar ao menu
- Quando o protagonista: NERITH morre em qualquer fase o jogo volta para a tela inicial e recomeça do zero.

### **Apresentação:**

Este manual tem como objetivo orientar o usuário final na utilização do jogo digital NERITH, desenvolvido como projeto acadêmico por uma equipe de estudantes de Engenharia de Software. O jogo foi projetado com foco em acessibilidade, jogabilidade fluida e imersão narrativa, e está disponível para execução em navegadores web (plataforma WebGL).

### **Perfil do usuário:**

O público-alvo do jogo inclui jogadores iniciantes ou casuais que desejam vivenciar uma experiência 2D de ação e aventura em um ambiente pixelado. O jogo pode ser utilizado em computadores pessoais com navegador moderno e acesso à internet.

### **Tela inicial (Main Menu) - Elementos:**

Botões "Jogar", "Opções" e "Sair".

- Ação esperada: Ao clicar em "Iniciar Jogo", o usuário será direcionado à tela de introdução da fase.
- Acesso rápido e direto com foco em simplicidade.

### **Fases jogáveis (Ex: savePoint.unity, Fases.unity) - Navegação:**

- Teclas direcionais ou WAD: movimentação do personagem.
- Barra de Espaço: pular.
- Mouse ou Barra de espaço =>Tecla de Ataque: atacar inimigos.
- Interações automáticas com plataformas, inimigos e checkpoints.

### **Interface (HUD) - Elementos visíveis:**

- Barra de Vida do Jogador (HUD superior).
- Feedback visual de dano e recuperação.
- A HUD se adapta dinamicamente com base nas ações do jogador.

### **Funcionalidades gerais (sistema de checkpoint automático):**

- Derrota ocorre quando a barra de vida é zerada.
- Possível retorno ao menu principal ao pressionar ?Esc?.
- Avanço automático para a próxima fase após a conclusão

### **Níveis de acesso e perfis de usuário:**

- O jogo é monousuário e não possui diferentes níveis de acesso.
- Todos os usuários têm as mesmas permissões durante a execução.

### **Recomendações:**

- Jogue com fones de ouvido para maior imersão.
- Recarregue a página em caso de travamento inesperado.
- O jogo é compatível com navegadores modernos (Chrome, Firefox, Edge).

### **Conclusão**

A utilização do Game NERITH foi planejada para ser intuitiva, acessível e envolvente, proporcionando ao jogador uma experiência agradável em um ambiente 2D de ação e aventura. Este manual apresenta de forma clara e objetiva todas as funcionalidades disponíveis, a navegação entre as telas, os controles básicos e as interações esperadas dentro do jogo.

Com uma interface simplificada, mecânicas bem definidas e fases progressivamente desafiadoras, o jogo busca atender tanto os jogadores iniciantes quanto àqueles que desejam um entretenimento casual e dinâmico. Espera-se que, com o apoio deste manual, o usuário possa explorar o universo de NERITH com autonomia, imersão e prazer.

## **MANUAL DE INSTALAÇÃO - Game NERITH**

O Manual de Instalação do Game NERITH, elaborado pela equipe de graduandos em Engenharia de Software do 7º período com o objetivo de fornecer orientações detalhadas e precisas para a implantação correta do sistema, garantindo sua disponibilidade e funcionamento adequado para o usuário final. Embora o jogo esteja preparado para execução em navegadores via WebGL, este documento foi pensado principalmente para uso por profissionais técnicos das áreas de infraestrutura e suporte, tais como administradores de rede, administradores de banco de dados, gestores de segurança da informação, analistas de suporte técnico e desenvolvedores. O manual contempla as seguintes instruções necessárias para instalar e executar o jogo digital NERITH, desenvolvido na Unity e exportado para WebGL, permitindo que o jogo seja executado diretamente em navegadores modernos:

### **Pré-Requisitos (mínimos para execução do jogo):**

*Para execução correta do jogo, recomenda-se*

- Navegador: Google Chrome, Mozilla Firefox, Microsoft Edge (atualizados).
- Sistema Operacional: Windows 10 ou superior, macOS ou Linux com suporte a WebGL.
- Memória RAM: mínimo de 4 GB.
- Conexão com a internet para carregamento do jogo.
- Resolução de tela recomendada: 1280x720 ou superior.

### **Passos para a instalação e execução, local (desenvolvedores)**

1. Para fins de desenvolvimento e testes locais:
2. Instale o Unity Hub.
3. Baixe e instale o Unity Hub (recomendada versão 2021.3 LTS ou superior).
4. Instale a Unity com URP e suporte a plataforma PC, Mac & Linux Standalone
5. Clone ou baixe o repositório do jogo do GitHub para seu computador.

6. Espere o carregamento e clique em "Open"
7. Abra o Unity Hub => Add => selecione a pasta do projeto.
8. Confirme se a cena "savePoint.unity" está aberta.
9. Certifique-se de que o módulo WebGL está instalado no Unity versão 6000.0.40f1.
10. Vá em File => Build and Run ou pressione Play para testar
11. Vá em File > Build Settings => WebGL => Build and Run.

### **Passo a passo (Configuração do ambiente de desenvolvimento com Unity e WebGL):**

- Input configurado com "InputSystem\_Actions.inputactions"
- URP ativado com "GraphicsSettings.asset"
- Prefabs e assets já referenciados nos scripts
- Scene "savePoint.unity" já adicionada ao "EditorBuildSettings.asset"

### **Problemas comuns (Soluções para possíveis erros de execução):**

- Falta do pacote Input System: instale via Package Manager
- Erros de build: cheque dependências URP e Graphics Settings
- Tela preta: verifique se a cena principal está no Build Settings
- Recomendação de backups regulares em versões do projeto (para desenvolvedores);
- Verifique a garantia de compatibilidade entre as plataformas envolvidas.

### **Recomendações:**

Este documento pode ser complementado conforme a evolução e o crescimento do projeto e a expansão de sua distribuição, colaborando com a manutenção futura e com a integração de novos ambientes (manutenção adaptativa).

### **Execução em ambiente Web (usuário final)**

#### **Para executar o jogo via navegador:**

1. Acesse o link onde o jogo foi hospedado (ex: itch.io ou GitHub Pages).
2. Aguarde o carregamento completo do jogo (pode levar alguns segundos).
3. Pressione =>Iniciar Jogo =>na tela inicial.
4. Jogue normalmente com teclado e mouse.



**Possíveis problemas (solução):**

- O jogo não carrega: verifique se o navegador é compatível com WebGL e se o JavaScript está habilitado.
- Travamentos ou lentidão: feche outras abas e programas pesados, ou tente um navegador diferente.
- Build local não executa: certifique-se de que o WebGL Build Support está instalado no Unity.

**Suporte:**

- Em caso de dúvidas técnicas, entre em contato com a equipe de desenvolvimento por meio do e-mail na capa deste documento.
- Institucional fornecido no repositório oficial GitHub ou pelos canais acadêmicos da Univassouras Campus Marica – Engenharia de Software.

**Conclusão – Manual de Instalação**

**O *Manual de Instalação do Game NERITH*** fornece orientações práticas e bem estruturadas para a configuração e execução local do jogo. Voltado a usuários técnicos, o documento reduz riscos operacionais e garante que o executável `.exe` seja acessado de forma estável e segura.

Com base em boas práticas de desenvolvimento e segurança, o manual também apoia a manutenção, suporte e futura escalabilidade do projeto, servindo como referência para profissionais envolvidos no ciclo de vida do software.