**SENAC/SESC ESCOLA  CURSO JOGOS DIGITAIS**

**Desenvolvimento de Jogos Digitais para pc**

**CUIABÁ**

**2024**

**MARCOS VINICIUS PEREIRA**

**Desenvolvimento de Jogos Digitais para pc z**

**trabalho feito para**

**o curso do senac**

**jogos   digitais no sesc escola**

**Orientador:                                                                                              Wanderson Timóteo**

**CUIABÁ**

**2024**

**RESUMO:**

A pesquisa sobre o desenvolvimento de jogos digitais para PC aborda as principais áreas envolvidas na criação de jogos. Ela explora como os dispositivos de entrada, como teclado, mouse e gamepads, são integrados com game engines para melhorar a interação do jogador. Também analisa os princípios de design de jogabilidade, como as ferramentas das engines podem ser usadas para criar uma experiência envolvente e como os desafios dentro do jogo influenciam as escolhas do jogador. Além disso, estuda o design de interfaces de usuário adaptáveis, a construção de jogos redistribuíveis, o versionamento de código para um gerenciamento eficiente de projetos, as técnicas de depuração de erros e a criação de inteligência artificial para personagens que interagem com as mecânicas do jogo. Cada um desses aspectos contribui para o desenvolvimento de jogos mais imersivos e tecnicamente bem-feitos.

**ABSTRACT**

The research on digital game development for PC covers the main areas involved in game creation. It explores how input devices, such as keyboard, mouse, and gamepads, are integrated with game engines to enhance player interaction. It also examines gameplay design principles, how game engine tools can be used to create an engaging experience, and how in-game challenges influence player choices. Additionally, it studies adaptable user interface design, the creation of redistributable game executables, version control for efficient project management, debugging techniques, and the creation of artificial intelligence for characters that interact with game mechanics. Each of these aspects contributes to the development of more immersive and technically well-crafted games.

**SUMÁRIO**

[**1. Interação do Jogador através dos Dispositivos de Entrada** 7](#_Toc193696883)

[ 1.1 Tecnologias de Game Engine e suas Capacidades 7](#_Toc193696884)

[ 1.2 Tipos de Dispositivos de Entrada para PC 7](#_Toc193696885)

[ 1.3 Integração de Dispositivos de Entrada com Game Engines 7](#_Toc193696886)

[ 1.4 Exemplos de Jogos com Interação Inovadora 7](#_Toc193696887)

[**2. Desenvolvimento da Jogabilidade Central** 7](#_Toc193696888)

[ 2.1 Princípios de Design de Jogabilidade 7](#_Toc193696889)

[ 2.2Interação Jogador-Jogo e Impacto na Experiência do Usuário 7](#_Toc193696890)

[ 2.3 Ferramentas de Game Engines para Desenvolver Jogabilidade 7](#_Toc193696891)

[ 2.4 Estudos de Caso de Jogos com Jogabilidade Central Bem-Sucedida 7](#_Toc193696892)

[3. Desafios e Consequências no Jogo 7](#_Toc193696893)

[ 3.1 Tipos de Desafios em Jogos 7](#_Toc193696894)

[ 3.2 Teoria das Mecânicas de Jogo 8](#_Toc193696895)

[ 3.3 Impacto das Consequências nas Decisões dos Jogadores 8](#_Toc193696896)

[ 3.4 Exemplos de Jogos com Sistemas de Desafio Complexos 8](#_Toc193696897)

[**4. Elementos de Interface de Usuário Adaptáveis** 8](#_Toc193696898)

[ 4.1 Princípios de Design de Interface de Usuário (UI) 8](#_Toc193696899)

[ 4.2 Ferramentas e Técnicas para Criar UI Adaptável em PCs 8](#_Toc193696900)

[ 4.3 Integração de UI com Game Engines 8](#_Toc193696901)

[ 4.4 Estudos de Caso de Interfaces de Jogos que Melhoram a Experiência do Usuário 8](#_Toc193696902)

[**5. Construção de Executáveis Redistribuíveis** 8](#_Toc193696903)

[ 5.1 Processos de Construção de Jogos para PC 8](#_Toc193696904)

[ 5.2 Otimização de Desempenho e Compatibilidade 8](#_Toc193696905)

[ 5.3 Ferramentas e Técnicas para Criar Executáveis Redistribuíveis 8](#_Toc193696906)

[ 5.4 Exemplos de Pipelines de Construção de Jogos 8](#_Toc193696907)

[**6. Procedimentos de Versionamento** 9](#_Toc193696908)

[ 6.1 Importância do Versionamento no Desenvolvimento de Jogos 9](#_Toc193696909)

[ 6.2 Ferramentas de Versionamento Populares 9](#_Toc193696910)

[ 6.3 Melhorias Práticas para Versionamento de Jogos 9](#_Toc193696911)

[ 6.4 Estudos de Caso de Projetos de Jogos com Bom Gerenciamento de Versão 9](#_Toc193696912)

[7. Depuração de Código 9](#_Toc193696913)

[ 7.1 Técnicas de Depuração de Jogos 9](#_Toc193696914)

[ 7.2 Ferramentas de Depuração Suportadas por Game Engines 9](#_Toc193696915)

[ 7.3 Estruturação de Planos de Depuração Eficazes 9](#_Toc193696916)

[ 7.4 Exemplos de Problemas Comuns e Soluções na Depuração de Jogos 9](#_Toc193696917)

**[8. Desenvolvimento de Desafios e Comportamento de Personagens](#_Toc193696918)** [10](#_Toc193696918)

[ 8.1 Criação de IA para Personagens Protagonistas e Antagonistas 10](#_Toc193696919)

[ 8.2 Ferramentas e Técnicas para Desenvolver Comportamento de Personagens 10](#_Toc193696920)

[ 8.3 Integração de Desafios e Comportamento com Mecânicas de Jogo 10](#_Toc193696921)

[ 8.4 Exemplos de Jogos com IA Avançada e Desafiadora 10](#_Toc193696922)

# **1. Interação do Jogador através dos Dispositivos de Entrada**

* 1.1 Tecnologias de Game Engine e suas Capacidades**: Estudo sobre as capacidades das principais engines de jogos (Unity, Unreal Engine, Godot, etc.) no que tange ao suporte a diferentes tipos de dispositivos de entrada, como sensores de movimento, controle por toque, ou até mesmo dispositivos inovadores como óculos de realidade aumentada.**
* 1.2 Tipos de Dispositivos de Entrada para PC**: Aqui, o foco é nos dispositivos comuns usados no PC, como teclado, mouse, gamepads, joysticks, e outros periféricos (ex: controles de realidade virtual ou câmeras de movimento).**
* 1.3 Integração de Dispositivos de Entrada com Game Engines**: A integração de dispositivos de entrada com engines de jogo envolve não apenas a programação para captar os sinais dos dispositivos, mas também a interpretação desses sinais dentro do contexto do jogo. Examinar como as engines lidam com essa integração pode revelar boas práticas e técnicas.**
* 1.4 Exemplos de Jogos com Interação Inovadora**: Analisar jogos que fazem uso inovador desses dispositivos de entrada, como jogos que utilizam sensores de movimento ou interfaces personalizadas, como "Just Dance" ou "Beat Saber", que envolvem uma experiência interativa única.**

# **2. Desenvolvimento da Jogabilidade Central**

* 2.1 Princípios de Design de Jogabilidade**: Explorar os conceitos fundamentais por trás do design de jogabilidade, como o equilíbrio entre desafio e recompensa, e como isso afeta a experiência do jogador.**
* 2.2Interação Jogador-Jogo e Impacto na Experiência do Usuário**: Estudar como a interação entre jogador e jogo pode criar uma experiência envolvente. Isso inclui aspectos como feedback sensorial, ritmo do jogo e narrativa interativa.**
* 2.3 Ferramentas de Game Engines para Desenvolver Jogabilidade**: Investigar ferramentas nas engines de jogo que ajudam no desenvolvimento de mecânicas de jogabilidade, como sistemas de física, inteligência artificial, animações e controle de câmeras.**
* 2.4 Estudos de Caso de Jogos com Jogabilidade Central Bem-Sucedida**: Analisar exemplos como "The Legend of Zelda: Breath of the Wild" ou "Portal", que são amplamente reconhecidos por sua jogabilidade refinada, que mantém os jogadores engajados e desafiados.**

# 3. Desafios e Consequências no Jogo

* 3.1 Tipos de Desafios em Jogos**: Estudo das várias formas de desafios presentes em jogos, como puzzles, combate, exploração ou desafios de estratégia, e como eles podem ser combinados para criar uma experiência diversificada.**
* 3.2 Teoria das Mecânicas de Jogo**: Analisar como as mecânicas de jogo (ações que o jogador pode realizar) interagem para criar uma experiência rica e desafiadora.**
* 3.3 Impacto das Consequências nas Decisões dos Jogadores**: Explorar como as escolhas do jogador dentro do jogo podem levar a consequências impactantes, seja no desenvolvimento da narrativa, evolução dos personagens ou mudanças no mundo do jogo.**
* 3.4 Exemplos de Jogos com Sistemas de Desafio Complexos**: Jogos como "Dark Souls" e "The Witcher 3" são conhecidos por seus sistemas de desafios complexos que influenciam diretamente a maneira como os jogadores abordam o jogo.**

# **4. Elementos de Interface de Usuário Adaptáveis**

* 4.1 Princípios de Design de Interface de Usuário (UI)**: Estudo dos princípios que governam o design de interfaces de usuário eficazes, como simplicidade, clareza e acessibilidade.**
* 4.2 Ferramentas e Técnicas para Criar UI Adaptável em PCs**: Explorar as tecnologias que permitem criar interfaces que se adaptam ao contexto do jogo, como elementos dinâmicos que mudam com base no status do jogador ou do jogo.**
* 4.3 Integração de UI com Game Engines**: Como as UI são integradas nas engines de jogos e quais técnicas e frameworks são utilizados para criar interfaces eficientes.**
* 4.4 Estudos de Caso de Interfaces de Jogos que Melhoram a Experiência do Usuário**: Exemplos de jogos que utilizam interfaces inovadoras para melhorar a imersão e a experiência do usuário, como as interfaces de "Horizon Zero Dawn" ou "Overwatch", que são altamente intuitivas.**

# **5. Construção de Executáveis Redistribuíveis**

* 5.1 Processos de Construção de Jogos para PC**: Investigar os passos necessários para criar e empacotar jogos que possam ser executados em diferentes configurações de PC. Inclui o uso de ferramentas como Unity, Unreal Engine, e técnicas como construção de pipelines automatizados.**
* 5.2 Otimização de Desempenho e Compatibilidade**: A necessidade de otimizar os jogos para garantir boa performance e compatibilidade com uma ampla gama de hardwares e sistemas operacionais.**
* 5.3 Ferramentas e Técnicas para Criar Executáveis Redistribuíveis**: Estudar os melhores métodos para criar pacotes de jogos prontos para distribuição, como usando serviços de distribuição de jogos como Steam ou Epic Games Store.**
* 5.4 Exemplos de Pipelines de Construção de Jogos**: Como estúdios de jogos criam e gerenciam seus pipelines de construção, com foco na eficiência e confiabilidade, usando ferramentas como Jenkins, GitLab CI, ou customizações de scripts de construção.**

# **6. Procedimentos de Versionamento**

* 6.1 Importância do Versionamento no Desenvolvimento de Jogos**: Como o versionamento permite que equipes de desenvolvimento colaborem eficientemente e mantenham o controle de mudanças no código-fonte do jogo.**
* 6.2 Ferramentas de Versionamento Populares**: Ferramentas como Git, Subversion (SVN), ou Perforce são essenciais para o desenvolvimento colaborativo de jogos. Estudo de como essas ferramentas são usadas para gerenciar grandes bases de código.**
* 6.3 Melhorias Práticas para Versionamento de Jogos**: Como otimizar o versionamento de jogos para grandes arquivos, múltiplos desenvolvedores e integrações contínuas.**
* 6.4 Estudos de Caso de Projetos de Jogos com Bom Gerenciamento de Versão**: Analisar como grandes estúdios utilizam o versionamento para gerenciar jogos de grande escala e como as melhores práticas podem ser aplicadas a projetos menores.**

# 7. Depuração de Código

* 7.1 Técnicas de Depuração de Jogos**: Técnicas específicas de depuração aplicadas ao contexto dos jogos, como rastreamento de erros gráficos, de física ou inteligência artificial.**
* 7.2 Ferramentas de Depuração Suportadas por Game Engines**: Ferramentas como debuggers integrados em engines de jogo (ex: Unity Profiler, Unreal Debugger) que ajudam os desenvolvedores a localizar problemas no código e melhorar o desempenho do jogo.**
* 7.3 Estruturação de Planos de Depuração Eficazes**: Como planejar uma abordagem estruturada para depuração, organizando testes, simulando erros e usando dados de usuários para melhorar a qualidade do jogo.**
* 7.4 Exemplos de Problemas Comuns e Soluções na Depuração de Jogos**: Estudo de problemas frequentes de programação de jogos e como solucioná-los, como erros de física, glitches gráficos e falhas de IA.**

# **8. Desenvolvimento de Desafios e Comportamento de Personagens**

* 8.1 Criação de IA para Personagens Protagonistas e Antagonistas**: Como criar inteligência artificial para NPCs (personagens não-jogáveis), sejam aliados ou inimigos, e como a IA pode influenciar a jogabilidade.**
* 8.2 Ferramentas e Técnicas para Desenvolver Comportamento de Personagens**: Estudo das técnicas de desenvolvimento de IA, como máquinas de estados finitos (FSM), comportamento baseado em objetivos (GOAP), e redes neurais para IA mais complexa.**
* 8.3 Integração de Desafios e Comportamento com Mecânicas de Jogo**: Como o comportamento da IA é integrado com o design das mecânicas de jogo, criando desafios adaptativos para o jogador.**
* 8.4 Exemplos de Jogos com IA Avançada e Desafiadora**: Análise de jogos como "Halo", "The Last of Us", ou "Alien: Isolation", que apresentam IA inovadora que desafia o jogador de maneiras imprevisíveis.**