

Engenharia de Software na Criação de Jogos Eletrônicos

Introdução

O interesse pela área de criação de jogos digitais pela grande maioria das vezes vem porque a pessoa gosta muito de jogar. A indústria de jogos eletrônicos tem experimentado um crescimento significativo nos últimos anos, impulsionada pelo avanço tecnológico e pelo aumento do interesse dos jogadores. Com isso o desenvolvimento de jogos eletrônicos vem se tornando uma área cada vez mais predominante na área da engenharia de software. Com a ampliação do mercado de entretenimento virtual, cada vez mais demanda-se profissionais nesta área, que exige conhecimento técnico específico e diferente daquele encontrado em processos de engenharia. A criação de jogos eletrônicos é uma tarefa complexa que envolve uma série de etapas e processos. Neste texto, exploraremos como a engenharia de software desempenha um papel fundamental no desenvolvimento de jogos eletrônicos, abordando desde a concepção até a implementação e testes.

Metodologias de Desenvolvimento de Jogos Eletrônicos

A concepção de um jogo eletrônico é o primeiro passo no processo de desenvolvimento. Nesta etapa, os conceitos e ideias iniciais são definidos e refinados. A equipe de desenvolvimento, composta por designers, artistas e engenheiros de software, trabalha em conjunto para definir a mecânica do jogo, a história, os personagens e os objetivos. O uso de técnicas como brainstorming, prototipagem rápida e design thinking é comum nessa fase.

Após a fase de concepção, inicia-se o processo de design e desenvolvimento do jogo eletrônico. Nesta etapa, os engenheiros de software utilizam linguagens de programação, frameworks e ferramentas específicas para implementar as funcionalidades do jogo. De acordo com CLUA (2004), a estrutura de um projeto de desenvolvimento de jogos seria a etapa em que o jogo é criado em si, ou seja, anexa-se a história aos personagens no mundo com sua fantasia própria, além de envolver diversos outros passos como criação da IA, por exemplo. O design de software é essencial para garantir uma arquitetura sólida e modular, facilitando a manutenção e a extensibilidade do código fonte. Além disso, os artistas criam os elementos visuais do jogo, como cenários, personagens e objetos, utilizando softwares de modelagem 3D e edição de imagens.



Etapas de produção de um jogo

- 1 – Pesquisa de mercado. A primeira fase de todas é a mais importante, porém muitos desenvolvedores iniciantes a pulam por achar que não é necessário.
- 2 – Game Design. A fase 2 é o game design.
- 3 – Prototipagem.
- 4 – Produção.
- 5 – Lançamento e divulgação.

Exemplo de Ferramentas de Desenvolvimento de Jogos Eletrônicos

Categoria	Ferramenta
Linguagem	C++, C#, Java
Framework	Unity, Unreal Engine
Modelagem 3D	Blender, Maya, 3ds Max
Edição de Imagens	Photoshop, GIMP
Controle de Versão	Git, SVN

Principais Desafios da Engenharia de Software em Jogos Eletrônicos

Complexidade dos sistemas de software: Os jogos eletrônicos são projetos de software altamente complexos, com uma ampla variedade de componentes interdependentes, como gráficos, física, inteligência artificial, som e rede. Gerenciar

essa complexidade é um dos principais desafios da engenharia de software em jogos.

Otimização de desempenho: Os jogos eletrônicos exigem um alto desempenho para garantir uma experiência de jogo suave e imersiva. A otimização do desempenho é um desafio constante, envolvendo a melhoria do uso dos recursos de hardware, a redução do tempo de carregamento e a otimização de algoritmos e processos.

Integração de tecnologias de ponta: Os jogos eletrônicos frequentemente incorporam tecnologias de ponta, como realidade virtual, realidade aumentada e inteligência artificial. Integrar essas tecnologias de forma eficaz e harmoniosa no jogo é um desafio técnico significativo.

Gerenciamento de ativos de conteúdo: Os jogos eletrônicos dependem de uma variedade de ativos de conteúdo, como gráficos, animações, áudio e níveis de jogo. Gerenciar e integrar esses ativos de forma eficiente, garantindo sua qualidade e consistência, é um desafio importante.

Testes e garantia de qualidade: Os jogos eletrônicos precisam passar por extensos testes e garantia de qualidade para garantir que funcionem corretamente em diferentes plataformas e configurações de hardware. Isso inclui testes de funcionalidade, testes de desempenho, testes de usabilidade e depuração de bugs.

Colaboração em equipe: O desenvolvimento de jogos eletrônicos envolve equipes multidisciplinares, incluindo programadores, artistas, designers e engenheiros de som. Coordenar e colaborar efetivamente entre essas diferentes disciplinas é um desafio, especialmente quando se lida com prazos apertados e requisitos em constante evolução.

Manutenção e suporte contínuos: Os jogos eletrônicos modernos são frequentemente atualizados com novo conteúdo, correções de bugs e recursos adicionais após o lançamento. Gerenciar a manutenção e o suporte contínuos ao longo do ciclo de vida do jogo requer um planejamento cuidadoso e uma estratégia de engenharia de software bem definida.

Tendências Futuras

Realidade Virtual e Realidade Aumentada: Com o contínuo avanço da tecnologia de realidade virtual (VR) e realidade aumentada (AR), espera-se que essas tecnologias sejam cada vez mais utilizadas na criação de jogos eletrônicos. Isso inclui a criação de experiências imersivas e interativas que permitem aos jogadores se envolverem diretamente com o ambiente virtual.

Inteligência Artificial (IA): A IA tem desempenhado um papel cada vez mais importante na indústria de jogos eletrônicos, seja na criação de personagens não jogáveis (NPCs) com comportamentos realistas, no aprimoramento da jogabilidade ou no desenvolvimento de algoritmos de aprendizado de máquina para personalizar

a experiência do jogador. Espera-se que a IA continue a desempenhar um papel significativo na engenharia de software de jogos eletrônicos no futuro.

Jogos em Nuvem: Com a melhoria da infraestrutura de internet e o aumento da velocidade de conexão, os jogos em nuvem estão se tornando mais viáveis. Essa abordagem permite que os jogadores acessem jogos de alta qualidade sem a necessidade de um hardware poderoso, pois o processamento é realizado nos servidores em nuvem. Essa tendência pode levar a uma maior disponibilidade de jogos para dispositivos móveis e uma experiência de jogo mais acessível em diferentes plataformas.

Jogos Sociais e Multijogador: Os jogos eletrônicos com foco em interações sociais e experiências multijogador estão se tornando cada vez mais populares. Essa tendência provavelmente continuará no futuro, com a engenharia de software se concentrando em recursos de compartilhamento, colaboração e comunicação entre os jogadores.



Conclusão

Com esse trabalho conclui-se que a Engenharia de Software desempenha um papel essencial na criação de jogos eletrônicos e se difere dos processos da engenharia de software convencionais principalmente por existir a necessidade de inspiração artística para um bom game designer, mas também um bom game designer, que são aqueles que se dedicam ao estudo das técnicas propostas neste trabalho, desde a concepção das ideias até o desenho do ambiente, da interface homem máquina, e do balanceamento do jogo oferecendo abordagens e técnicas para o desenvolvimento eficiente e de alta qualidade. Com o crescimento da indústria de jogos e o avanço da tecnologia, é fundamental acompanhar as tendências e os desafios em constante evolução. A aplicação adequada da Engenharia de Software no desenvolvimento de jogos contribui para a criação de experiências envolventes e cativantes para os jogadores.

Referências Bibliográficas

Felipe Castanheira Morais¹ Cristiano Maciel Silva – Orientador², DESENVOLVIMENTO DE JOGOS ELETRÔNICOS. Disponível em: [134 \(unibh.br\)](http://134.unibh.br), acessado em: 27/05/2023

Samanta Bueno de Camargo Campana, Eduardo Martins Morgado, Wilson Massashiro Yonezawa, Edriano Carlos Campana, Anderson Rogério Campana, Maria Cristina Gobbi, Edutretenimento e sua relação com jogos digitais educacionais. Disponível em: <https://www.bing.com/ck/a?!&p=879200f6549d1064JmldHM9MTY4NTE0NTYwMCZpZ3VpZD0wOTAyMjNjNS1iNGlyLTYYNjgtMTViZC0zMTA3YjVIMzYzNGQmaW5zaWQ9NTE4MA&pptn=3&hsh=3&fclid=090223c5-b4b2-6268-15bd-3107b5e3634d&psq=clua+2004&u=a1aHR0cHM6Ly9yZXZpc3Rhcy5maWJiYXVydS5ici9tdWx0aXBsaWNpZGFkZWZpYi9hcnRpY2xIL2Rvd25sb2FkLzkwLzcxLzEzNQ&ntb=1> Visitado em 27/05/2023

Sistema de itens estilo Skyrin. Disponível em: [java - Sistema de itens estilo Skyrim - Stack Overflow em Português](#) Visitado em 27/05/2023

Como funciona o processo de criação de um jogo? Disponível em: [Como funciona o processo de criação de um jogo? \(vocepergunta.com\)](#) Visitado em 27/05/2023

Lista de Siglas e Abreviaturas

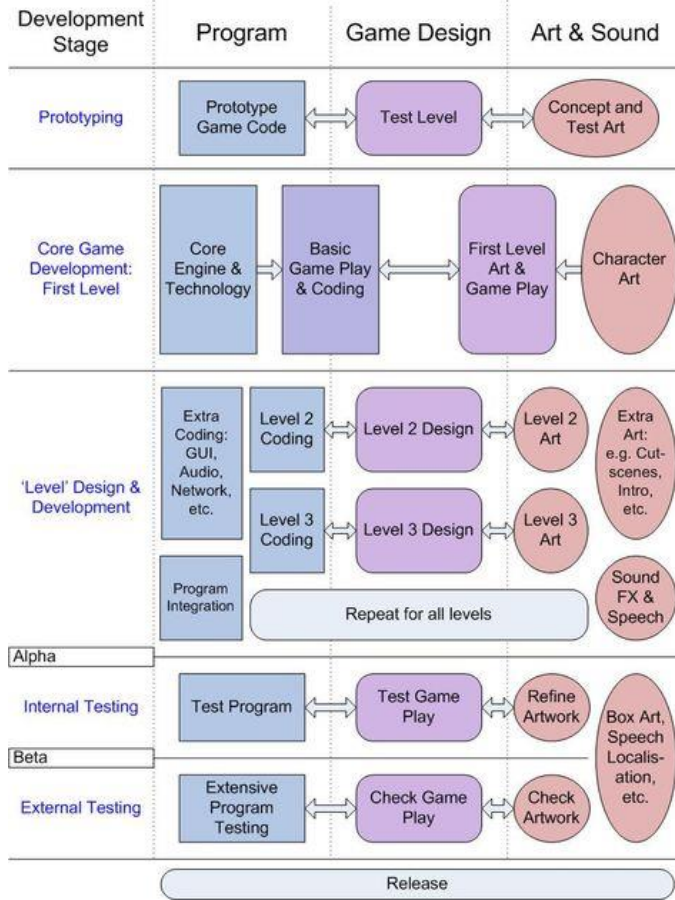
- IA: Inteligência Artificial
- VR: Realidade Virtual
- AR: Realidade Aumentada
- NPCs: Personagens não jogáveis

Glossário

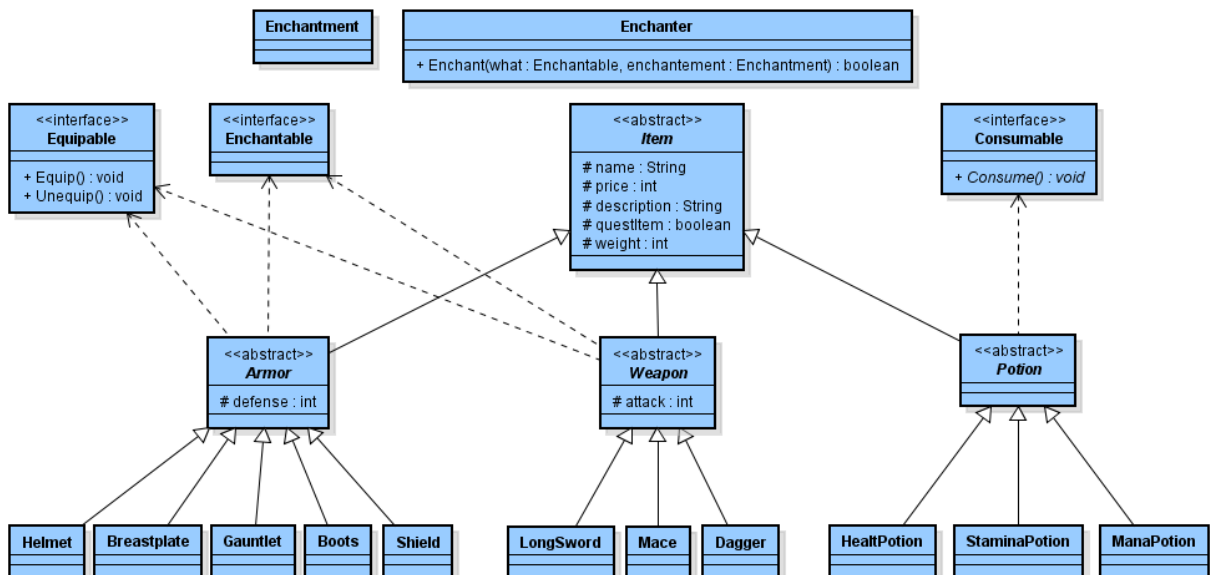
- Colisão: No contexto dos jogos, colisão refere-se à interação entre objetos virtuais no espaço do jogo, onde a detecção de colisão é usada para determinar quando dois objetos se encontram.
- Trilha Sonora: A trilha sonora é a composição musical específica para um jogo, que ajuda a criar a atmosfera e a imersão desejadas.

Anexo:

Development Timeline



Apêndice:



```

public class Player {
    int hp;
}

```

```

        public Player(int hp) {
            this.hp = hp;
        }
    }

    public abstract class Item implements Equipable, Consumable{
        Player possesor;
        String name = "";
        int price = 0;
        boolean questItem = false;
        boolean equiped = false;

        public Item(Player p) {
            possesor = p;
        }
    }

    public interface Equipable {
        void equip();
        void unequip();
    }public interface Consumable {
        void consume();
    }
}
package br.com.interfaces;

import sun.reflect.generics.reflectiveObjects.NotImplementedException;

public class Tester {

    public static void main(String[] args) {

        Item ringOfLife = new Item(new Player(0)) {

            @Override
            public void consume() {
                throw new NotImplementedException();
            }

            @Override
            public void unequip() {
                if (equiped) {
                    possesor.hp -= 10;
                    equiped = false;
                }
            }

            @Override
            public void equip() {
                // Ok, eu sei, isso deveria ser um encantamento ,-,
                if (!equiped) {
                    possesor.hp += 10;
                }
            }
        };
    }
}

```

```
        equipped = true;
    }
};

assert(ringOfLife.possesor.hp == 0);
assert(ringOfLife.equiped == false);
ringOfLife.equip();
assert(ringOfLife.possesor.hp == 10);
assert(ringOfLife.equiped == true);
ringOfLife.unequip();
assert(ringOfLife.possesor.hp == 0);
assert(ringOfLife.equiped == false);
}
}
```