

CENTRO UNIVERSITÁRIO DO DISTRITO FEDERAL - UDF
CIÊNCIA DA COMPUTAÇÃO

Eduardo de Brito da Silva - 30127599

Hugo Borges Pereira da Silva - 30253471

Lucas José da Silva Germano - 29418101

Matheus Fernandes Duda - 29985684

Victor dos Santos Silva - 29789613

JOGO ZATURA
ZATHURA GAME

Brasília - DF

2024

RESUMO

O projeto Zatura é um jogo 2D desenvolvido em Python, com o tema de defesa da Terra contra ameaças espaciais. Nele, o jogador controla uma nave que orbita o planeta e tem como missão destruir meteoros para que não atinjam a Terra. Os meteoros surgem em ondas progressivas, tornando-se mais rápidos, numerosos e resistentes conforme o jogo avança. A nave conta com armas e habilidades que podem ser melhoradas, exigindo estratégia e precisão do jogador. O projeto combina entretenimento com o uso de conceitos de programação e algoritmos, reforçando habilidades práticas de desenvolvimento.

ABSTRACT

The Zatura project is a 2D game developed in Python, themed around defending Earth from space threats. In the game, the player controls a spaceship orbiting the planet with the mission of destroying meteors before they hit Earth. The meteors appear in progressive waves, becoming faster, more numerous, and more resistant as the game progresses. The spaceship is equipped with weapons and abilities that can be upgraded, requiring strategy and precision from the player. The project combines entertainment with the application of programming and algorithm concepts, reinforcing practical development skills.

INTRODUÇÃO

O projeto Zatura foi desenvolvido com o objetivo de integrar ao jogo os conceitos aprendidos na disciplina de Computabilidade e Complexidade de Algoritmos. Durante a criação, foram aplicados estudos sobre análise de complexidade algorítmica, avaliando a eficiência dos algoritmos utilizados no comportamento dos meteoros e no controle da nave. Também foram empregadas técnicas de programação, como refinamento passo a passo, iteração e recursão, para implementar a lógica de geração das ondas de meteoros e a movimentação orbital da nave. A abordagem de divisão e conquista foi utilizada para dividir o problema em subproblemas menores, como o cálculo de colisões e a atribuição de atributos variáveis aos meteoros, como tamanho, velocidade e resistência. Por fim, métodos de análise de eficiência foram aplicados para otimizar o desempenho do jogo, garantindo fluidez mesmo com o aumento progressivo da complexidade das ondas de meteoros.

PROCESSO DE CRIAÇÃO DO JOGO

Planejamento

O desenvolvimento do jogo começou com a definição do conceito central, que é a defesa da Terra contra meteoros. Durante essa fase, também foram especificadas as mecânicas fundamentais do jogo, como o controle da nave, o disparo de projéteis e as melhorias que podem ser adquiridas ao longo da partida.

Desenvolvimento

O ambiente de desenvolvimento foi configurado utilizando a biblioteca Pygame, escolhida por sua facilidade de uso e recursos para desenvolvimento de jogos 2D. As classes principais, como Game, Ship e Sound, foram criadas para modularizar o código e garantir uma estrutura organizada. A lógica do jogo foi implementada para permitir o controle da nave, o disparo de projéteis e o gerenciamento dos eventos no jogo. Também foram desenvolvidas interfaces de usuário (UI), incluindo menus e a tela de jogo, que exibem informações como pontuação e vidas. A implementação de sons e música de fundo foi realizada para melhorar a imersão do jogador.

Testes

Durante todo o processo de desenvolvimento, foram realizados testes contínuos para garantir que as mecânicas de jogo funcionassem corretamente. A dificuldade do jogo foi ajustada com base no feedback dos testes, buscando equilibrar a experiência e proporcionar um desafio progressivo, mas acessível.

Envio final

A fase final do projeto envolveu a preparação do jogo para entrega do trabalho, o que incluiu a organização do código e a documentação completa do projeto.

DESCRIÇÃO DO JOGO

O jogador controla uma nave espacial com a missão de interceptar meteoros que chegam em ondas progressivas e ameaçam colidir com a Terra. Para evitar o

impacto, é necessário mover a nave utilizando propulsão e realizar disparos que fragmentam os meteoros até destruí-los. Ao longo do jogo, *power-ups* aparecem aleatoriamente pelo mapa, e ao coletá-los, o jogador recebe melhorias temporárias que ajudam a destruir os meteoros ou a proteger a nave, proporcionando uma vantagem estratégica.

ESTRUTURA DO CÓDIGO

O código do jogo "Zatura" é organizado em vários arquivos, cada um responsável por diferentes aspectos do jogo. Abaixo estão os principais componentes.

Arquivos principais

O projeto iniciou com a criação do arquivo README.md, onde foram definidas a descrição do jogo e o conceito do projeto. Após isso, foi criado o arquivo principal game.py, responsável por controlar a lógica geral do jogo, incluindo o estado do jogo, eventos e atualizações das entidades. Para organizar o código, outros arquivos foram desenvolvidos com responsabilidades específicas. O arquivo ship.py define a classe Ship, que representa a nave do jogador, com mecânicas de movimento e tiro. O arquivo sound.py gerencia os efeitos sonoros e a música de fundo, enquanto asteroid.py define a classe para os meteoros que o jogador deve interceptar. O arquivo galaxy.py organiza o espaço de jogo e gerencia as entidades presentes, e countdown.py é responsável pelas contagens regressivas, como o tempo para power-ups ou mudanças de fases. Por fim, score.py gerencia a pontuação e as vidas do jogador, registrando o desempenho ao longo das partidas. Essa estrutura modular favorece a organização e manutenção do código.

Classes e Funções Principais

A estrutura do jogo é organizada em várias classes e funções principais, cada uma com responsabilidades específicas.

Classe Game

A classe Game é responsável por inicializar o jogo e configurar a tela. Ela gerencia o estado do jogo, alternando entre os modos MENU e PLAYING. Essa

classe também controla a lógica de início de um novo jogo e gerencia os eventos relacionados aos power-ups, como a ativação e uso das melhorias durante a partida.

Classe Ship

A classe Ship controla a nave do jogador, permitindo o movimento, o disparo de projéteis e a detecção de colisões com meteoros. Além disso, ela implementa mecânicas de power-ups, como o tiro rápido e o tiro triplo, que melhoram a capacidade de ataque da nave. A classe também gerencia a proteção da nave, incluindo o escudo que pode absorver impactos e garantir a sobrevivência do jogador.

Classe Sound

A classe Sound segue o padrão Singleton, garantindo que apenas uma instância de gerenciamento de som seja criada. Ela é responsável por carregar e tocar os efeitos sonoros, além de gerenciar a música de fundo. A classe permite que o volume de diferentes categorias de som (efeitos, música) seja ajustado, oferecendo uma experiência auditiva personalizada.

Eventos e Interações

O jogo utiliza eventos do Pygame para capturar as entradas do jogador, como pressionamento de teclas para mover a nave ou disparar projéteis. Além disso, eventos personalizados são usados para gerenciar interações específicas, como a ativação de power-ups e a detecção de colisões entre a nave e os meteoros. Essa abordagem permite uma maior flexibilidade na resposta do jogo aos inputs do jogador.

CONCLUSÃO

A execução do projeto Zatura permitiu uma rica experiência de aprendizado, integrando teoria e prática. Por meio do desenvolvimento do jogo, foi possível aprofundar os conhecimentos em computabilidade e complexidade de algoritmos, aplicando técnicas como análise de eficiência, refinamento passo a passo e divisão e conquista. Além disso, o projeto destacou a importância de atitudes fundamentais para o sucesso acadêmico e profissional, como a valorização do trabalho em equipe, a cooperação e o respeito às opiniões dos colegas. A interação durante o processo de aquisição do conhecimento mostrou-se essencial, evidenciando que a troca de ideias e a colaboração enriquecem tanto o produto final quanto o desenvolvimento individual dos integrantes. Essa estrutura permitiu criar um jogo bem desenvolvido, equilibrado e com uma experiência de usuário envolvente, dessa forma, o projeto reforçou a compreensão prática dos conceitos teóricos, demonstrando sua aplicabilidade em um contexto interativo e desafiador.