**Space Rush**

Alunos:

Eduardo Oliveira Almeida – 7210995

João Pedro Ribeiro de Menezes - 7211248

**Introdução**

Durante o curso de Computação Gráfica foram apresentados diversos conceitos, como Sintese de Imagens, Transformações Geométricas, Iluminção e Textura. O projeto da matéria visa desenvolver as habilidades adquiridas durante as aulas e desenvolver as ferramentas apresentadas – como o Blender e o Processing.

A computação gráfica está presente em diversos elementos na tecnologia, desde de representações até interações, seja para entretenimento ou para educação, os elementos estão no dia a dia das pessoas, cada vez é mais comum, por exemplo, pessoas em uma fila compenetradas em um jogo de celular para passar o tempo.

**Objetivo**

O objetivo do projeto é desenvolver um jogo 3D com todos os elementos e conhecimentos obtidos durante o curso – como sistema de colisão, partículas, texturas e modelagem de objetos 3D. No jogo, interativo, o usuário controla uma nave no espaço e deve escapar dos asteroides que encontra pelo caminho. A nave irá ter 2 graus de liberdade – Horizontal e Vertical – enquanto os asteróides movimentam-se apenas horizontalmente, da esquerda para direita, mas aparecem em lugares aleatórios verticalmente. O ambiente do jogo é fixo, mas todos os elementos serão 3D.

**Implementação**

O projeto teve duas fases principais após definido o escopo, a modelagem dos objetos – asteroide e nave – e o desenvolvimento do código fonte. Durante a fase de modelagem do objeto, desenvolvidos em Blender, foram pesquisadas texturas que poderiam ser aplicadas e formatos que fossem simples mas ao mesmo tempo que a experiência do jogador não fosse pobre devido aos elementos gráficos. Além disso, logo no começo da implementação, percebemos que a imagem de fundo (background) tinha um grande impacto na performance do jogo, optamos então em remover o background durante a implementação e usar uma cor apenas.

A segunda parte do desenvolvimento foi toda dedicada ao código – Sistema de colisão, criação dos objetos, aplicação das texturas, sistema de partículas, pontuação, menu e movimentação dos objetos. Os maiores desafios foram o sistema de colisão, não por sua implementação em Java, mas pela decisão de qual algoritmo usar. Outro ponto de discussão foi o sistema de partículas, pois o projeto deveria balancear a quantidade de partículas na tela e a performance do jogo.

Os únicos softwares utilizados foram o Blender e o Processing.

**Resultados**

Os resultados mostrados abaixo são da versão apresentada no final do curso. Esses resultados estão dentro do proposto inicialmente pelo projeto, mas em termos de produto, ainda faltariam diversas implementações, que serão exploradas na seção Conclusão

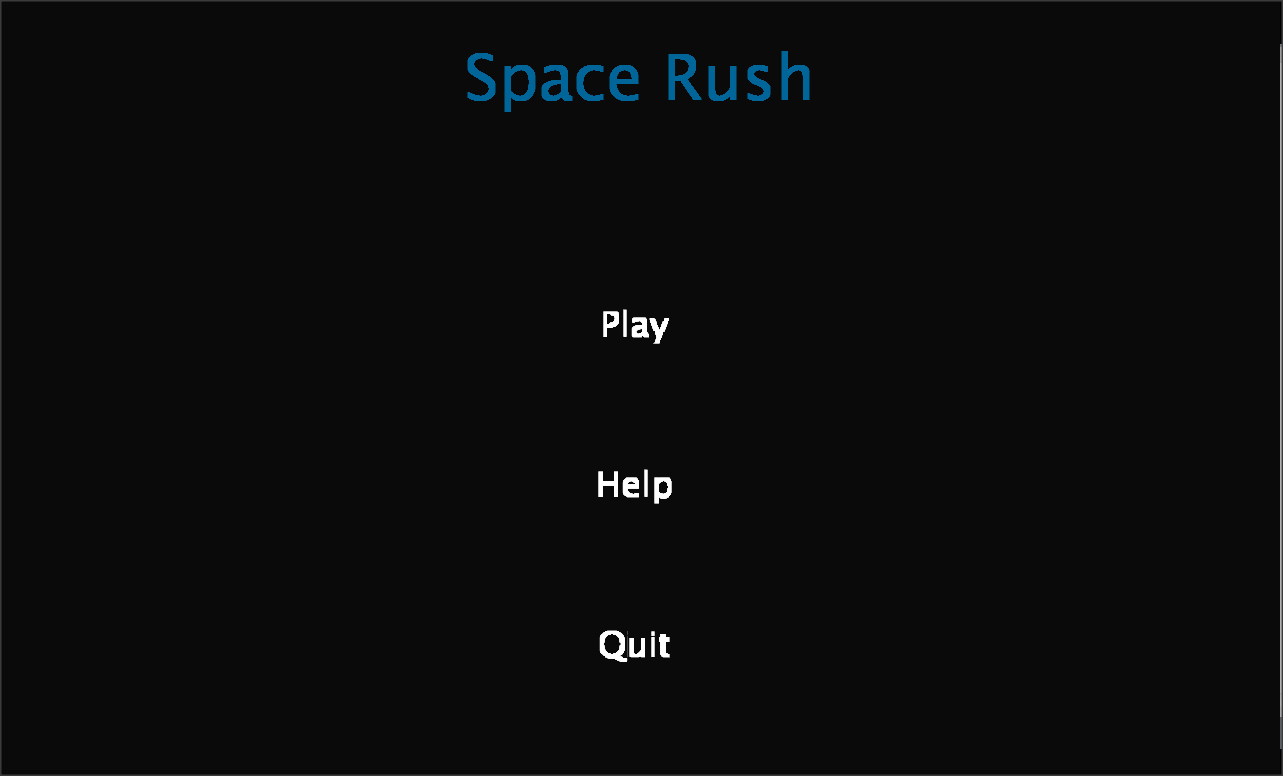
****

Figura - Tela de Início

****

Figura - Nave navegando sozinha

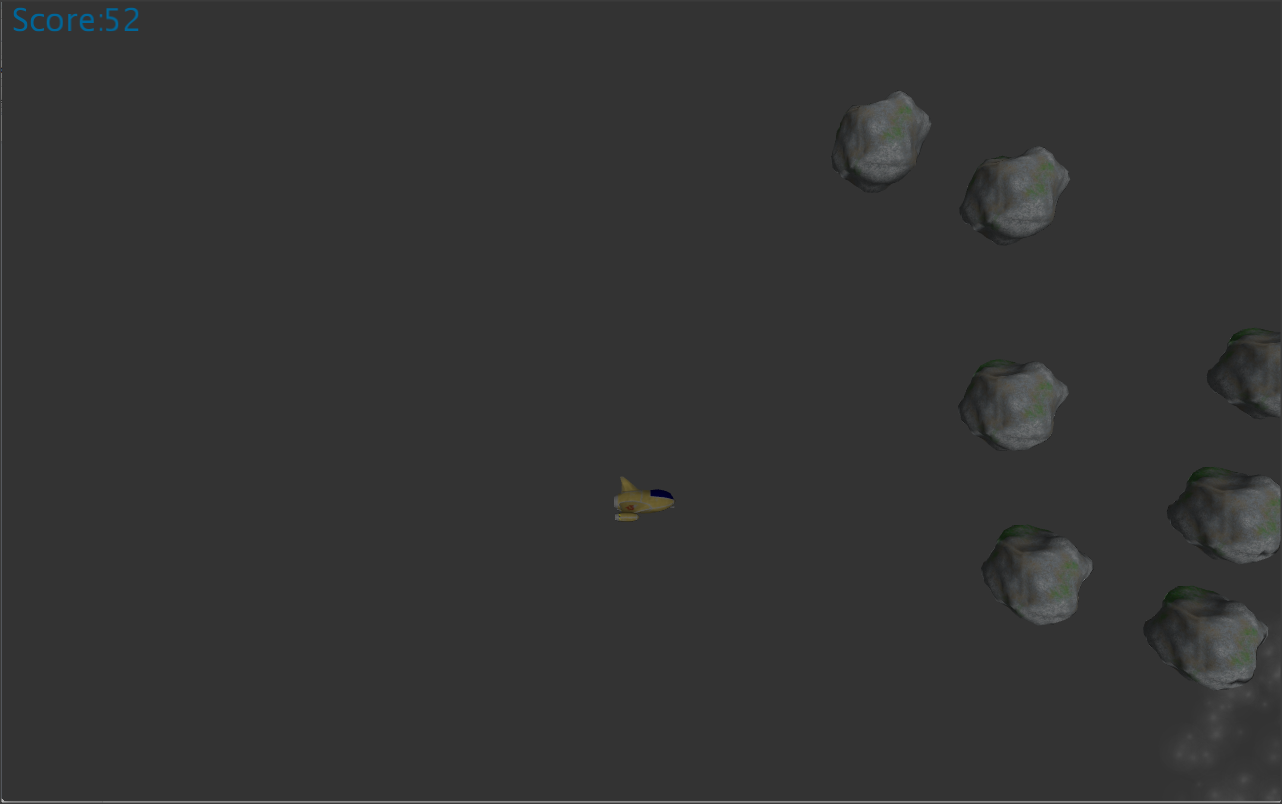
****

Figura - Asteroides no espaço

**Conclusão**

O projeto pode ser melhorado de diversas maneiras, vamos listar e explicar cada uma delas:

1. Melhorias no sistema de criação de Pedras

O sistema implementado tem uma quantidade fixa de pedras – Essa decisão foi feita com o intuito de economizar memória, mas não adiciona dificuldades a jogabilidade, uma vez que novas pedras não podem ser criadas. Para mitigar esse problema, as velocidades das pedras são variáveis e mudam ao voltarem para o começo da tela. Entretanto, para maiores opções de jogabilidade, este sistema precisa ser melhorado.

1. Melhorias no sistema de pontuação

A pontuação é totalmente baseada no tempo – Quanto mais o jogador conseguir manter sem colidir, mais pontos, porém, para uma melhor experiência, poderíamos implementar adicionais, como bônus por desvios complicados ou similares.

1. Maior interação do usuário

Hoje o jogo não permite que a nave atire nos asteroides, esta seria uma grande melhoria na jogabilidade.

1. Elementos diferentes como obstáculos

Outra grande melhoria seria a criação de mais inimigos, não apenas asteroides mas inimigos que façam trajetórias diferentes e possam também contra atacar a nave.

1. Sons

O jogo não possui nenhum som, neste campo podem ser feitas diversas melhorias.

1. Fontes

As fontes utilizadas pelo programa estão serrilhadas, uma

melhoria seria desenhar as próprias fontes para que o look and feel do programa fosse melhor