



CES-22 - Programação Orientada a Objetos

Recriação do jogo River Raid com a biblioteca pygame

Bruno Monteiro Franzini

Luís Henrique Zuin Ruiz

William Ryu Gyotoku

COMP 24

Professores

Dr. Edgar Yano

Dra. Karla Donato Fook

Instituto Tecnológico de Aeronáutica – ITA

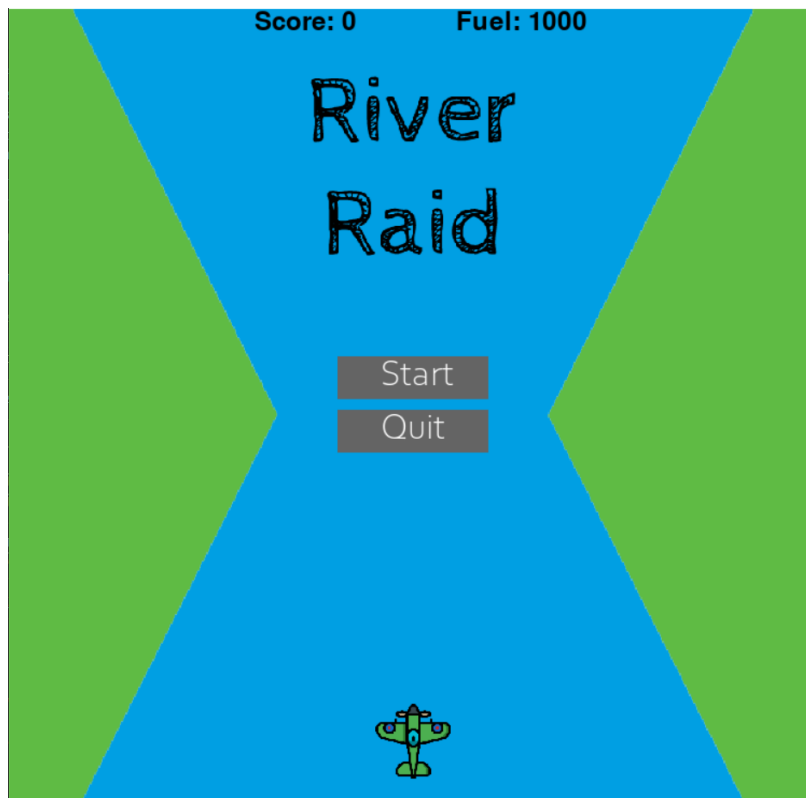
Departamento de Computação

Descrição do Jogo:

O jogo escolhido foi o clássico River Raid, jogado inicialmente no console Atari. O objetivo principal é sobreviver o máximo de tempo possível e assim pontuar cada vez mais. Para isso, o jogador controla um avião, o qual tem que usar para destruir os inimigos, desviar dos obstáculos e, com o tempo, reabastecer para não parar de voar. O jogo acaba se o jogador colidir com obstáculos, for atingido por inimigos, ou ficar sem combustível. Como forma de intensificar a jogabilidade, a velocidade do jogador aumenta gradativamente com o tempo, tornando cada vez mais difícil a sobrevivência.

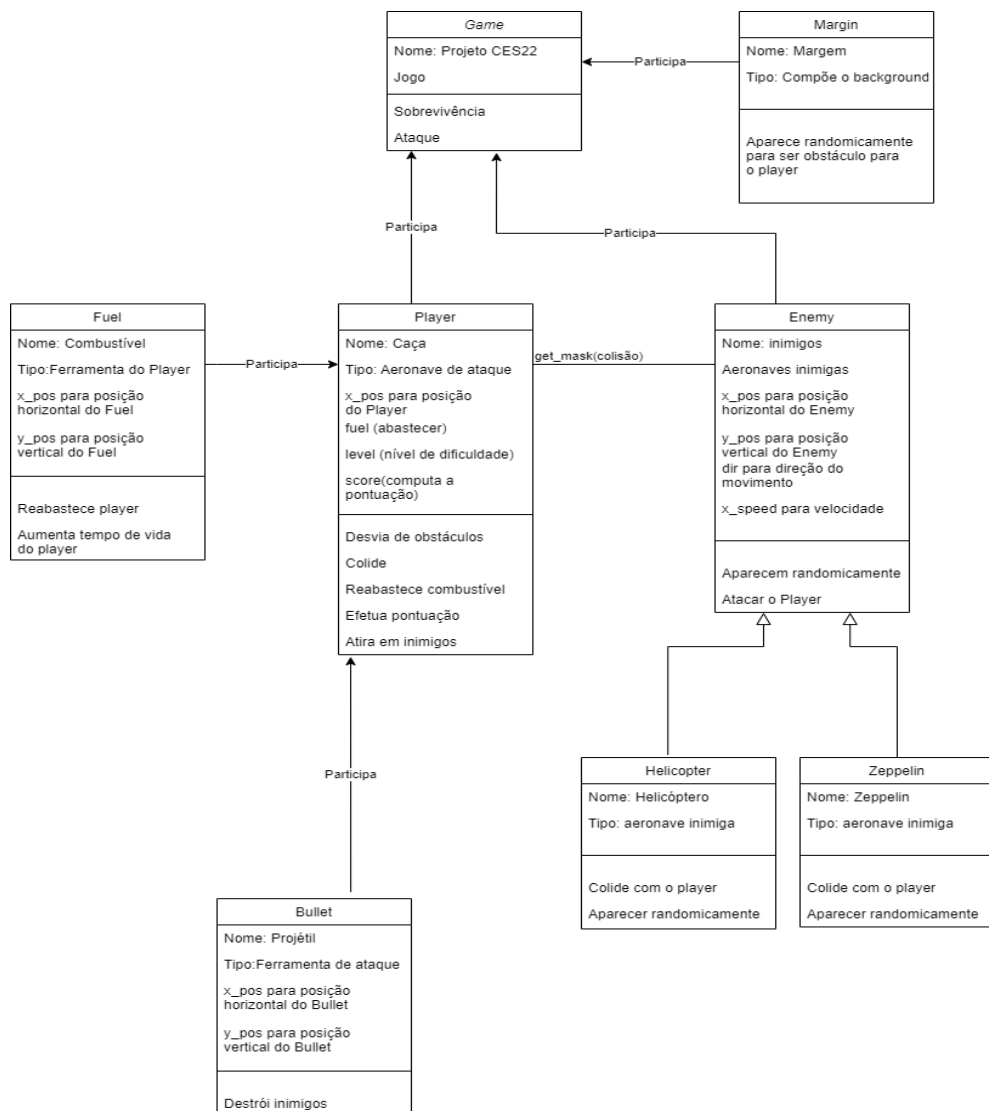
Entre os requisitos identificados no projeto estavam a randomização da aparição e movimento dos inimigos, a randomização da aparição do cenário, a implementação de uma dinâmica de controle que fosse semelhante ao jogo original, a criação de uma representação numérica para o combustível do jogador, e a implementação da física de colisões entre o jogador e obstáculos e inimigos e entre projéteis e inimigos.

Arquitetura



O jogo foi programado na linguagem Python, utilizando da biblioteca Pygame, além de bibliotecas auxiliares, como a Time, que foram usadas durante o desenvolvimento dos métodos. A classe Player representa o humano que está jogando, e é representada graficamente pela imagem de um avião. A sua posição na tela, o score atual, o combustível restante são alguns de seus atributos representados. Foi preciso também analisar a forma como essa classe interage com as demais.

Por fim, destaca-se o diagrama de classes UML criado para representar as interações entre os diferentes objetos criados, que pode ser visto abaixo:



Arquivos Entregues

A entrega consiste em uma pasta, contendo os arquivos de áudio utilizados, outra pasta contendo as imagens utilizadas e três arquivos .py: main, objects e physics.

O arquivo main implementa a linha temporal do jogo, criando os objetos iniciais, adicionando os elementos da tela e atualizando os diferentes componentes do cenário, além de controlar a aparição e funcionamento do menu.

Por sua vez, o arquivo objects define as classes utilizadas e funções e variáveis necessárias para sua execução.

Por fim, o arquivo physics implementa as interações de colisão entre as diferentes classes empregadas, ou seja, define como será feita a interação entre classes distintas.