

CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO – CC4Mb

Davi Conde

Felipe Yuki

Matheus Pastore

**TRABALHO PROGRAMAÇÃO ORIENTADA A OBJETOS 2**

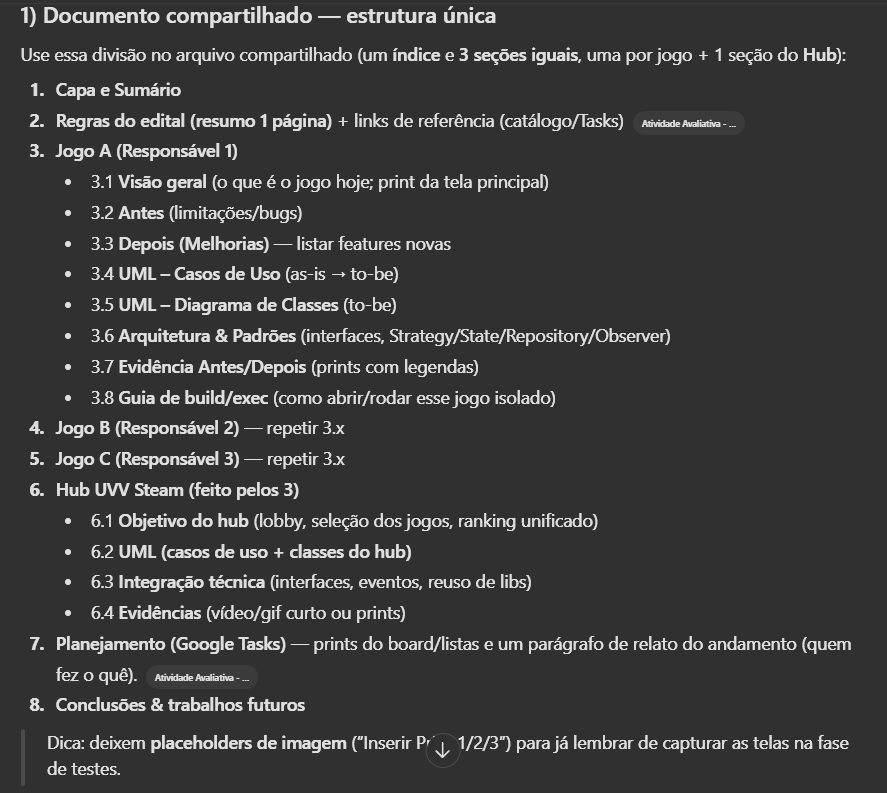
**1º BIMESTRE  
 2025/2**

**VILA VELHA - ES**

**2025**

**DOCUMENTAÇÃO**

**Resumo do Projeto (1 parágrafo)**

**Planejamento  
**

**Jogo 13**

**Título: Helicopter Shooter Game MOO ICT**

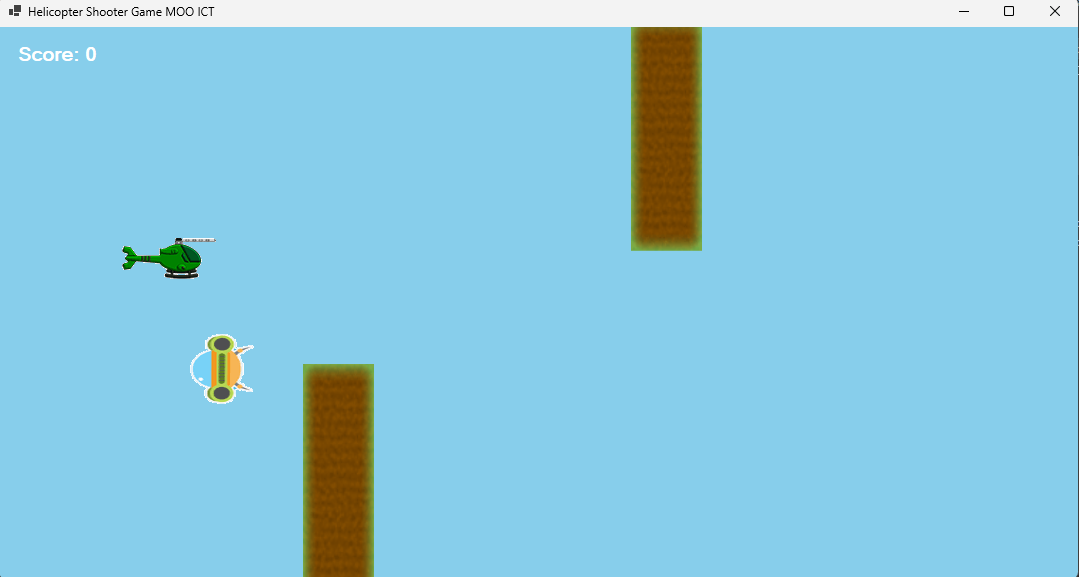
**Descrição:** O jogador controla um helicóptero que se move verticalmente e pode atirar para destruir um UFO inimigo. O cenário inclui obstáculos fixos (pilares) que devem ser evitados. O jogo termina quando o helicóptero colide com um obstáculo ou com o UFO. O jogador pode reiniciar o jogo pressionando Enter.

**Controles:**

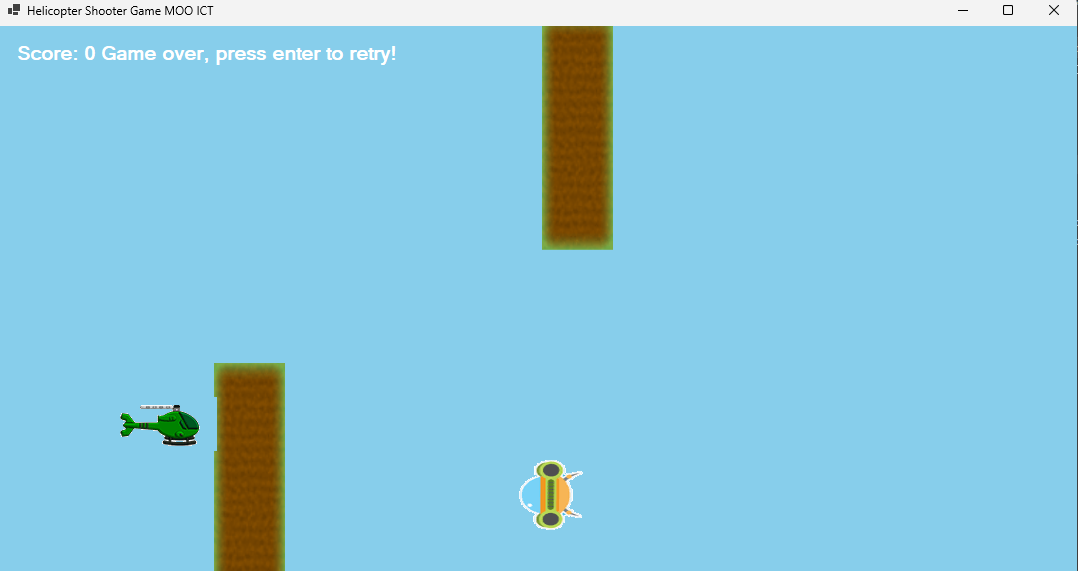
* **Seta para cima (↑):** move o helicóptero para cima
* **Seta para baixo (↓):** move o helicóptero para baixo
* **Espaço:** dispara um projétil
* **Enter:** reinicia o jogo após o "Game Over"

**Objetivo:**Destruir o maior número possível de UFOs e sobreviver o máximo possível, evitando colisões com obstáculos e o inimigo.

**JOGO RODANDO:**

****

**JOGO QUANDO GAME OVER:**

****

**Visão Geral / Gameplay**

mecânicas do jogo:

* O helicóptero se move verticalmente dentro da área da janela.
* Os pilares se movem da direita para a esquerda simulando movimento.
* O UFO também se move da direita para a esquerda e muda de imagem periodicamente.
* Quando o UFO é atingido por um projétil:
  + O UFO muda de posição e imagem.
  + O jogador ganha 1 ponto.
* Após 10 pontos:
  + A velocidade dos obstáculos e do UFO aumenta (dificuldade progressiva).
* Colidir com um pilar ou com o UFO termina o jogo.
* O texto da pontuação é atualizado a cada frame.
* Para começar novamente, basta apertar **Enter**.

telas do jogo:

* **Início do jogo**: o jogo começa diretamente quando a aplicação é executada, sem tela de menu.
* **HUD de pontuação**: no canto superior (Label txtScore), é exibida a pontuação atual do jogador.
* **Game Over**:
  + Ao colidir com um UFO ou com os pilares, o jogo é interrompido.
  + A Label txtScore mostra a mensagem **"Score: X Game Over, press Enter to retry!"**.
  + O jogador pode reiniciar o jogo pressionando a tecla **Enter**.

**Antes (limitações)**

Foi possível observar várias limitações que diminuem a qualidade da experiência do jogador, sendo elas:

* **Poucos tipos de inimigos e obstáculos**: atualmente o jogador enfrenta apenas UFOs e pilares fixos, o que deixa a experiência repetitiva com o tempo.
* Não há sistema de ranking ou persistência de pontuação: quando o jogo fecha, todos os pontos são perdidos; não existe histórico nem Top-5.
* Não existe opção de pausa: o jogador não consegue parar o jogo no meio; precisa perder ou fechar para sair.
* Dificuldade fixa: a velocidade dos obstáculos e a gravidade não aumentam ao longo do tempo, então o desafio não escala.
* Sem feedback sonoro ou visual extra: não há sons de colisão, pontuação ou efeitos visuais quando o jogador acerta/erra.
* Somente uma tela (MainWindow): não existe menu inicial, tela de opções ou de game over separada; tudo acontece em uma única janela.
* Colisão simples e sem variação: o hitbox é um retângulo fixo; em alguns momentos parece injusto ou pouco natural.
* Sem power-ups ou itens coletáveis: não há mecânicas extras que mudem a dinâmica (ex.: escudo, pontos dobrados, velocidade extra).
* Debug visual intrusivo: no game over, os retângulos aparecem com borda vermelha/preta; é útil para programador, mas quebra a imersão do jogo

**Possíveis Melhorias**

Algumas possíveis melhorias que podem ser implementadas:

* Ranking persistente (Top-5 em JSON)

Salvar as maiores pontuações em um arquivo JSON.

Mostrar uma tela de ranking ao final de cada partida.

O jogador insere iniciais quando bate recorde.

* Dificuldade progressiva.

A cada certo tempo ou pontuação, aumenta a velocidade dos obstáculos.

Deixar o jogo mais desafiador quanto mais tempo o jogador sobreviver

* Novos tipos de obstáculos.

Obstáculos com tamanhos/velocidades diferentes.

Obstáculos móveis (ex.: que pulam ou caem).

* Sistema de power-ups

Itens coletáveis que dão vantagem temporária:

Escudo (ignora 1 colisão).

Pontuação em dobro por alguns segundos.

Boost de velocidade.

* Menu inicial e tela de Game Over mais amigáveis

Menu com botões “Iniciar”, “Ranking”, “Sair”.

Tela final com score, recorde e botão de reinício.

* Função de pausa

Pressionar **P** para pausar/despausar.

Mostrar o overlay “Game Paused”.

* Feedback sonoro e visual

Sons para pulo, colisão e pontuação.

Efeitos visuais ao pegar power-up ou bater recorde.

* HUD aprimorado

Mostrar além do score: vidas, nível de dificuldade atual, tempo jogado.

**JOGO 16**

**Título:** Space Battle Shooter WPF

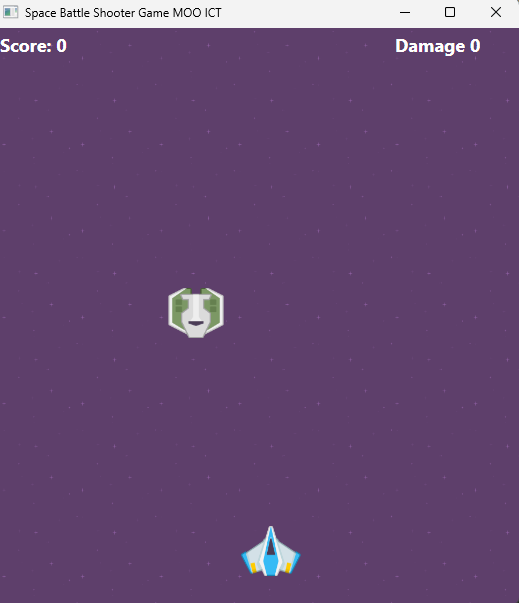
**Descrição:** Jogo 2D desenvolvido em C# onde o jogador controla uma nave espacial para defender a galáxia contra ondas de inimigos alienígenas. O jogo combina gráficos simples e envolventes com uma jogabilidade rápida e desafiadora.

**Controles:**

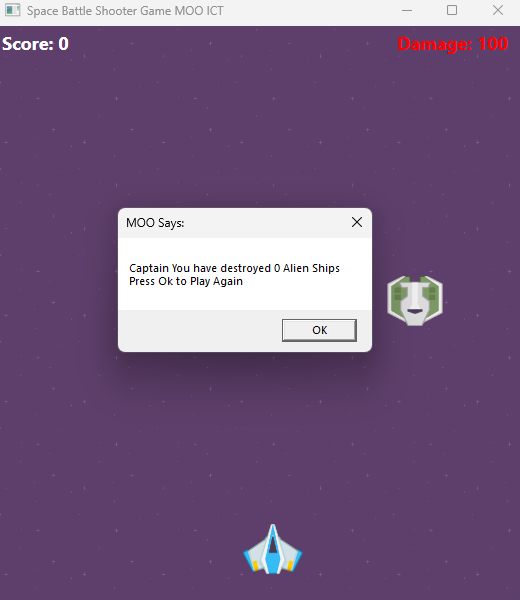
* Espaço: atirar.
* Setas: Move horizontalmente a nave(esquerda e direita).

**Objetivo:** Sobreviver o máximo possível e conquistar a maior pontuação possível, destruindo todas as naves inimigas que aparecem na tela antes que elas atinjam a sua nave ou a parte inferior da tela.

**Jogo rodando:**

****

**Jogo quando Game Over:**

****

### **Visão Geral / Gameplay**

**Mecânicas do jogo:**

* O personagem (nave espacial) se movimenta de acordo com o pressionar das teclas “setas” horizontalmente.
* Caso nenhuma tecla seja acionada a nave permanece parada na tela.
* Ao pressionar a tecla Espaço, o jogador faz a nave realizar um disparo, que pode destruir os inimigos que aparecem na tela.
* Quando o inimigo atinge a parte inferior da tela sem ser abatido o dano aumenta em 10. Caso atinja a nave, o dano passa a ser 5.
* Obstáculos (como asteroides ou minas espaciais) aparecem sempre à frente da nave e se movem em direção ao final da tela.
* Quando um obstáculo sai da tela, ele reaparece mais à frente em diferentes posições, para variar o desafio.
* A cada inimigo abatido o valor da pontuação(score) aumenta de um em um.
* O jogo termina imediatamente quando o dano atinge o valor 100.
* Para reiniciar o jogo, basta clicar em OK

**Telas do jogo:**

* O jogo inicia direto (sem menu).
* HUD de score no canto superior esquerdo (Label scoreText).
* HUD de dano no canto superior direito da tela.
* Game Over exibe dano total recebido na nave e o botão OK para jogar novamente.

**Antes (limitações)**

Foi possível observar várias limitações que diminuem a qualidade da experiência do jogador, sendo elas:

* Apenas um tipo de obstáculo: o jogador sempre enfrenta o mesmo inimigo (sendo somente uma placa escrita “sem correr”) , mudando só um pouco a altura. Isso torna o jogo repetitivo rapidamente.
* Não há sistema de ranking ou persistência de pontuação: quando o jogo fecha, todos os pontos são perdidos; não existe histórico nem Top-5.
* Não existe opção de pausa: o jogador não consegue parar o jogo no meio; precisa perder ou fechar para sair.
* Dificuldade fixa: a velocidade dos obstáculos e a gravidade não aumentam ao longo do tempo, então o desafio não escala.
* Sem feedback sonoro ou visual extra: não há sons de colisão, pontuação ou efeitos visuais quando o jogador acerta/erra.
* Somente uma tela (MainWindow): não existe menu inicial, tela de opções ou de game over separada; tudo acontece em uma única janela.
* Colisão simples e sem variação: o hitbox é um retângulo fixo; em alguns momentos parece injusto ou pouco natural.
* Sem power-ups ou itens coletáveis: não há mecânicas extras que mudem a dinâmica (ex.: escudo, pontos dobrados, velocidade extra).
* Debug visual intrusivo: no game over, os retângulos aparecem com borda vermelha/preta; é útil para programador, mas quebra a imersão do jogo

**Possíveis Melhorias**

Algumas possíveis melhorias que podem ser implementadas:

* Ranking persistente (Top-5 em JSON)

Salvar as maiores pontuações em um arquivo JSON.

Mostrar uma tela de ranking ao final de cada partida.

O jogador insere iniciais quando bate recorde.

* Dificuldade progressiva.

A cada certo tempo ou pontuação, aumenta a velocidade dos obstáculos.

Deixar o jogo mais desafiador quanto mais tempo o jogador sobreviver

* Novos tipos de obstáculos.

Obstáculos com tamanhos/velocidades diferentes.

Obstáculos móveis (ex.: que pulam ou caem).

* Sistema de power-ups

Itens coletáveis que dão vantagem temporária:

Escudo (ignora 1 colisão).

Pontuação em dobro por alguns segundos.

Boost de velocidade.

* Menu inicial e tela de Game Over mais amigáveis

Menu com botões “Iniciar”, “Ranking”, “Sair”.

Tela final com score, recorde e botão de reinício.

* Função de pausa

Pressionar **P** para pausar/despausar.

Mostrar o overlay “Game Paused”.

* Feedback sonoro e visual

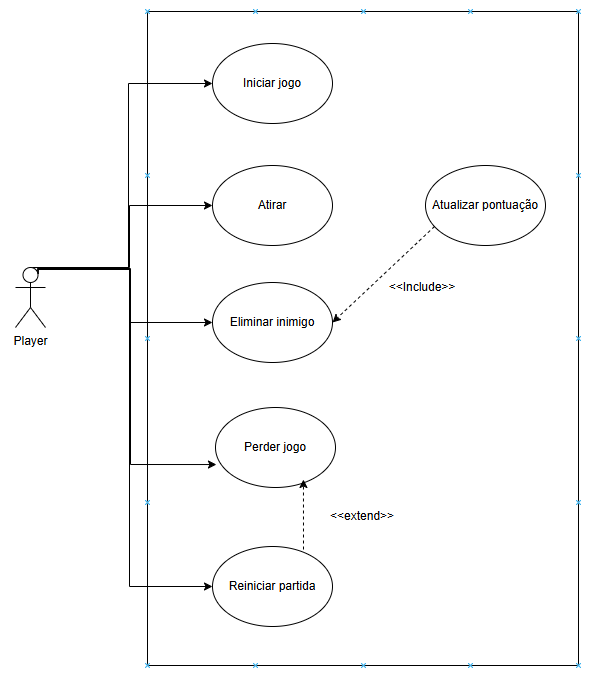
Sons para pulo, colisão e pontuação.

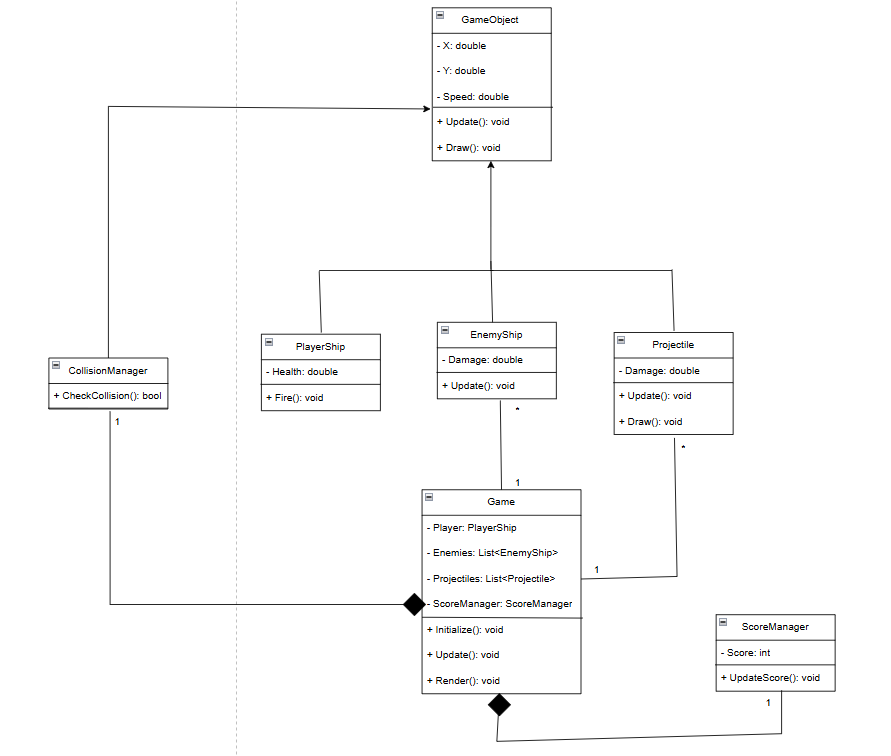
Efeitos visuais ao pegar power-up ou bater recorde.

* HUD aprimorado

Mostrar além do score: vidas, nível de dificuldade atual, tempo jogado.

**UML caso de uso**





**Jogo 14**

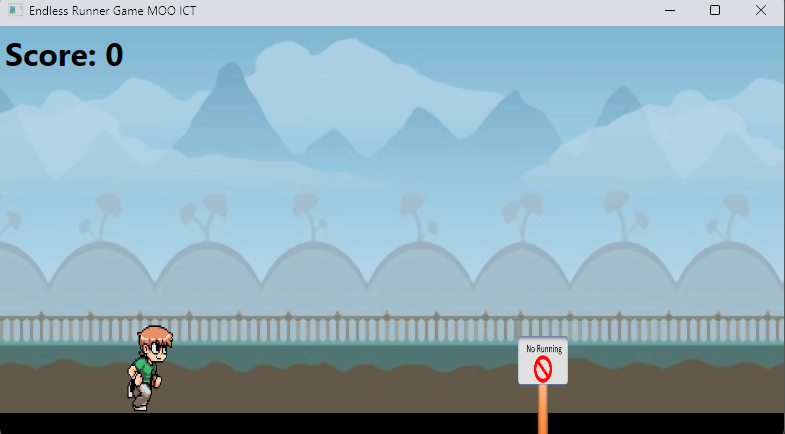
**Título:** Endless Runner WPF **Descrição:** Jogo 2D do tipo endless runner(corrida infinita) desenvolvido em C# com WPF. O personagem corre em um cenário com rolagem contínua, desviando de um obstáculo que surge repetidamente com alturas um pouco diferentes. A cada ciclo completo do obstáculo, o jogador ganha pontos. A partida termina ao colidir com o obstáculo, é possível reiniciar imediatamente.

**Controles:**

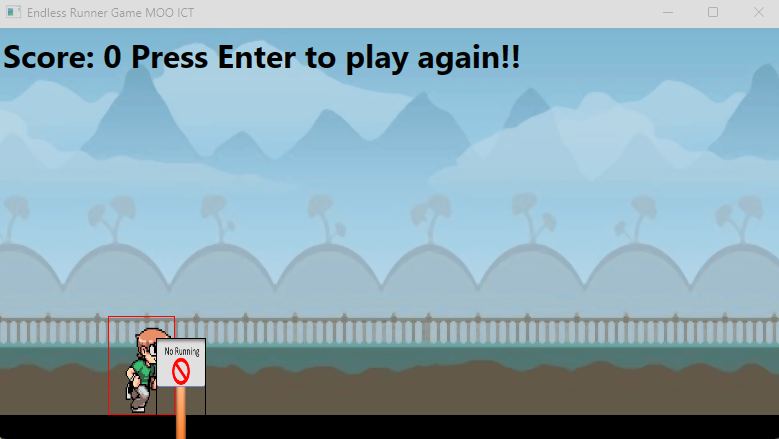
* Espaço: pular
* Enter: reiniciar após Game Over(perder)

**Objetivo:** Sobreviver o máximo de tempo possível, desviando do obstáculo e acumulando pontos a cada ciclo do obstáculo.

**Jogo rodando:**

****

**Jogo quando Game Over:**

****

**Visão Geral / Gameplay**

mecânicas do jogo:

* O personagem corre sozinho e a cada momento a gravidade faz ele voltar para o chão quando não está pulando.
* Ao apertar a tecla **Espaço**, o jogador pula. Ele sobe por um tempo e depois volta a cair naturalmente
* Quando encosta no chão, o personagem continua correndo e a animação de corrida volta ao normal.
* Um obstáculo aparece sempre na frente e se move em direção ao jogador. Quando passa pela tela, ele reaparece mais à frente, em alturas diferentes.
* A cada obstáculo superado, a **pontuação aumenta em 1**.
* O jogo acaba quando o personagem encosta em um obstáculo.
* Para começar novamente, basta apertar **Enter**.

telas do jogo:

* O jogo inicia direto (sem menu).
* HUD de score no canto superior (Label scoreText).
* Game Over exibe bordas da hitbox do personagem e mensagem no scoreText assim pedindo **Enter** para reiniciar.

**Antes (limitações)**

Foi possível observar várias limitações que diminuem a qualidade da experiência do jogador, sendo elas:

* Apenas um tipo de obstáculo: o jogador sempre enfrenta o mesmo inimigo (sendo somente uma placa escrita “sem correr”) , mudando só um pouco a altura. Isso torna o jogo repetitivo rapidamente.
* Não há sistema de ranking ou persistência de pontuação: quando o jogo fecha, todos os pontos são perdidos; não existe histórico nem Top-5.
* Não existe opção de pausa: o jogador não consegue parar o jogo no meio; precisa perder ou fechar para sair.
* Dificuldade fixa: a velocidade dos obstáculos e a gravidade não aumentam ao longo do tempo, então o desafio não escala.
* Sem feedback sonoro ou visual extra: não há sons de colisão, pontuação ou efeitos visuais quando o jogador acerta/erra.
* Somente uma tela (MainWindow): não existe menu inicial, tela de opções ou de game over separada; tudo acontece em uma única janela.
* Colisão simples e sem variação: o hitbox é um retângulo fixo; em alguns momentos parece injusto ou pouco natural.
* Sem power-ups ou itens coletáveis: não há mecânicas extras que mudem a dinâmica (ex.: escudo, pontos dobrados, velocidade extra).
* Debug visual intrusivo: no game over, os retângulos aparecem com borda vermelha/preta; é útil para programador, mas quebra a imersão do jogo

**Possíveis Melhorias**

Algumas possíveis melhorias que podem ser implementadas:

* Ranking persistente (Top-5 em JSON)

Salvar as maiores pontuações em um arquivo JSON.

Mostrar uma tela de ranking ao final de cada partida.

O jogador insere iniciais quando bate recorde.

* Dificuldade progressiva.

A cada certo tempo ou pontuação, aumenta a velocidade dos obstáculos.

Deixar o jogo mais desafiador quanto mais tempo o jogador sobreviver

* Novos tipos de obstáculos.

Obstáculos com tamanhos/velocidades diferentes.

Obstáculos móveis (ex.: que pulam ou caem).

* Sistema de power-ups

Itens coletáveis que dão vantagem temporária:

Escudo (ignora 1 colisão).

Pontuação em dobro por alguns segundos.

Boost de velocidade.

* Menu inicial e tela de Game Over mais amigáveis

Menu com botões “Iniciar”, “Ranking”, “Sair”.

Tela final com score, recorde e botão de reinício.

* Função de pausa

Pressionar **P** para pausar/despausar.

Mostrar o overlay “Game Paused”.

* Feedback sonoro e visual

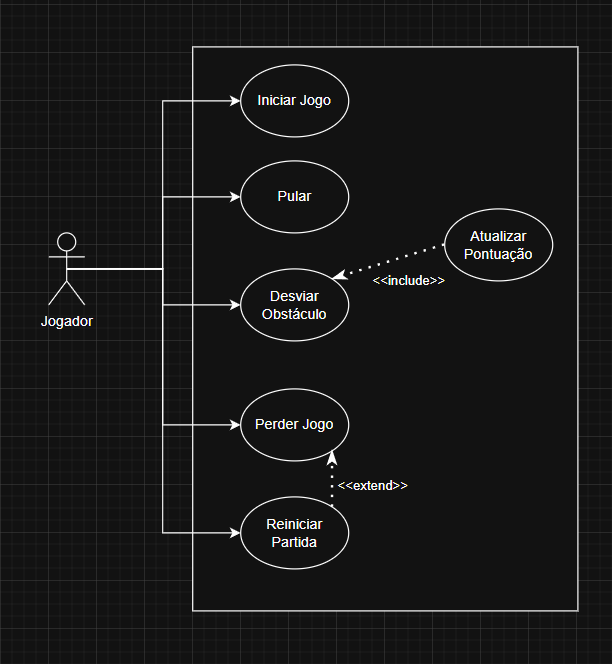
Sons para pulo, colisão e pontuação.

Efeitos visuais ao pegar power-up ou bater recorde.

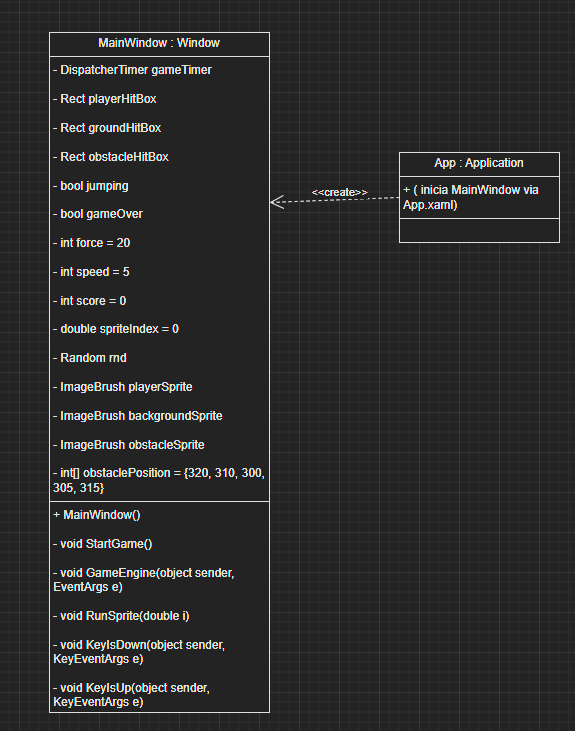
* HUD aprimorado

Mostrar além do score: vidas, nível de dificuldade atual, tempo jogado.

UML caso de uso



UML Diagrama de classes



Melhorias feitas no jogo:

Pequenas melhorias:

* modo de pausa
* organização no código
* hud em pt-br
* tamanho da janela aumentado
* depois de finalizar remover hitbox vermelha

Grandes melhorias/mudanças:

1. **timer de gameplay**

****

1. hub reorganização

**Conclusão e Próximos Passos**

**Créditos (quem fez o quê)**