

# Table of Contents

BRD .....	2
Space Invaders .....	6
Architeture .....	7
Enginnering Requeriments .....	8
MidTerm .....	10
Final .....	11
Resources .....	12
Sprites .....	13
Sounds .....	14
Uno Plataform .....	15

# BRD

## 1. Introdução

### 1.1. Objetivo do Projeto

O objetivo deste projeto é desenvolver uma recriação do clássico jogo "Space Invaders" como uma aplicação de desktop. O projeto servirá como avaliação final para a disciplina de Programação 3, focando na aplicação prática de conceitos de Programação Orientada a Objetos (OOP), Estruturas de Dados e manipulação de eventos em C#.

### 1.2. Escopo

O projeto será desenvolvido individualmente, utilizando C# e a Uno Platform, sem o auxílio de qualquer motor de jogo (game engine). A interface e a lógica do jogo serão construídas do zero, com entregas parciais e uma entrega final.

## 2. Requisitos Funcionais

### 2.1. Tela Inicial

- O jogo deve apresentar uma tela inicial com as seguintes opções:
  - **Iniciar um novo jogo:** Começa uma nova partida.
  - **Ver a tabela com os placares:** Exibe a lista de pontuações salvas.
  - **Ver os controles do jogo:** Mostra as instruções de como jogar.

### 2.2. Jogabilidade

#### 2.2.1. Nave do Jogador

- O jogador controla uma nave localizada na parte inferior da tela.
- A nave pode se mover horizontalmente (esquerda e direita).
- A nave pode disparar um projétil (laser) para cima.
- O jogador só pode ter um projétil ativo na tela por vez.

### **2.2.2. Inimigos (Alienígenas)**

- Múltiplas ondas de alienígenas são posicionadas na parte superior da tela.
- Os alienígenas se movem em um bloco coeso, da esquerda para a direita.
- Ao atingir a borda da tela, o bloco de alienígenas desce uma linha e inverte sua direção de movimento.
- A velocidade de movimento dos alienígenas e a frequência de seus disparos aumentam a cada vez que o bloco desce.
- Apenas o alienígena de 40 pontos pode atirar.
- Um alienígena especial (vermelho) aparece periodicamente no topo da tela, movendo-se horizontalmente e oferecendo uma pontuação variável.

### **2.2.3. Barreiras de Proteção**

- Haverá 4 blocos de proteção (escudos) localizados acima da nave do jogador.
- Os escudos servem para bloquear tiros, tanto do jogador quanto dos alienígenas.
- Os escudos se degradam visualmente (mudando de cor de branco para chumbo) ao serem atingidos e desaparecem após sofrerem uma certa quantidade de dano.

### **2.2.4. Sistema de Pontuação e Vidas**

- O jogador começa com um número fixo de vidas.
- A pontuação é incrementada ao destruir um alienígena, com valores diferentes para cada tipo:
  - Alienígena tipo 1: 10 pontos
  - Alienígena tipo 2: 20 pontos
  - Alienígena tipo 3: 40 pontos
  - Alienígena especial: Pontuação misteriosa, o jogador ganha uma vida extra, até um máximo de 6 vidas.

- A pontuação atual é sempre visível no canto superior esquerdo da tela.

## **2.3. Fim de Jogo**

- O jogo termina se o(a) (???)
- A cada 10 uma das seguintes condições for atendida:
  - O jogador perde todas as suas vidas.
  - O bloco de alienígenas alcança a parte inferior da tela (posição do jogador).
- Após o fim do jogo, o jogador tem a opção de:
  - Salvar sua pontuação, associando-a a um apelido.
  - Voltar a jogar.
  - Retornar à tela inicial.

## **2.4. Persistência de Dados**

- As informações do placar (pontuação e apelido) devem ser salvas em um arquivo de texto.

# **3. Requisitos Não-Funcionais**

## **3.1. Plataforma**

- O jogo deve ser uma aplicação de desktop, construída com a Uno Platform para garantir a portabilidade.

## **3.2. Tecnologia**

- A linguagem de programação será C#.
- A interface do usuário (UI) será definida usando XAML.
- Nenhuma game engine externa (ex: Unity, Godot) será utilizada.

### **3.3. Som**

- Cada ação significativa no jogo (disparo, destruição de inimigo, morte do jogador, etc.) deve ser acompanhada por um efeito sonoro correspondente.

## **4. Fases do Projeto**

### **4.1. Entrega Intermediária (Midterm - Semana 4)**

- **Escopo Reduzido:**
  - Alienígenas não precisam se mover.
  - O jogador deve ser capaz de se mover e atirar.
  - Os blocos de proteção devem ser destrutíveis pelo jogador.
  - A pontuação deve ser atualizada ao destruir um alienígena.
  - O jogo termina quando o jogador atinge 500 pontos.
  - A pontuação é mantida apenas em memória (perdida ao fechar o jogo).
  - Deve haver uma tela inicial e sons para as ações.

### **4.2. Entrega Final**

- Implementação de todos os requisitos funcionais e não-funcionais descritos neste documento.

# Space Invaders

Start typing here...

# Architeture

Technology stack

Wireframe

Database schema

# Enginnering Requeriments

## Requisitos Funcionais

- O sistema deve permitir que o usuário controle uma nave, podendo movimentar no sentido horizontal, sem sair da tela, usando as setas direcionais para esquerda e direita
- O usuario deve conseguir disparar lasers na direção vertical com o botão de espaço
- O sistema deve atribuir pontos ao usuário disparar e acertar laser nos inimigos, com base na tabela de inimigos ([Enginnering Requeriments](#))
- O sistema deve fazer com que os inimigos (do tipo 1, 2 ,3) apareçam na tela de forma aleatória e eles não podem sair do quadro
- O sistema deve fazer com que os alienigenas vermelhos se movem aleatoriamente começando da esquerda para a direita ou vice-versa e saindo do quadro, aparecendo 1 vez por 2 minutos aproximadamente, os pontos que eles lhe dão ao atirar variam
- O usuario deve ter 3 vidas iniciais, sendo que se for atingido ou encostar em um alienigena ele morre
- O sistema deve finalizar o jogo se o jogador perder todas as vidas ou terminara quando as naves alcançarem o jogador
- O sistema deve possuir 4 blocos de proteção, contra os alienígenas, eles mudarão de cor conforme forem recebendo danos e eventualmente desaparecendo
- O sistema deve fazer com que o movimento das naves alienigenas aumente um pouco e a velocidade de fogo também assim que se moverem para baix
- O sistem deve fazer com que a unica nave que ataca é o que valo 40 (veja mais na: tabela de inimigos ([Enginnering Requeriments](#)))
- O usuario deve disparar novamento quando seu tiro anterior acertar ou de outro forma quando exceder o limite supeprior do quadro do jogo
- O sistema deve fazer com que os alienigenas não colidam com os blocos de proteção



- O usuario deve conseguir destruir naves alienigenas com 1 tiro
- O sistema deve fazer aparecer uma nova onda assim que todas as naves alienigenas forem destruidas, essa nova onda gerada move a posção para baixo os primeiros 5 e aumentando um pouco a velocidade a partir da quinta em diant
- O sistema deve aumentar o número de vidas a cada 1000 pontos que o jogador conseguir
- O sistema deve exibir a pontualçai no canto superior esquerdo
- O sistema deve incrementar seguindo a tabela de inimigos ([Enginnering Requirements](#)) a cada vez que um inimigo é destruido
- O sistema deve ao fim do jogo, exibir uma opção para o jogador de salvar a pontuação e adicionar um apelido, deve tambem ofertar ao jogador a opção de voltar a jogar e se o jogador nao quiser jogar novamente a página inicial será exibida
- O sistema deve exibir a tela inicial, com: um botao de iniciar um novo jogo, um botao de ver a tabela com os placare, botao de ver os controles do jogo
- O sistema deve possuir um som representativo para cada ação do jogo
- O sistema deve salvar as informações do painel de pontuação em um arquivo de texto
- ..

# MidTerm

Start typing here...

# Final

Start typing here...

# Resources

Start typing here...

# Sprites

Start typing here...

# Sounds

Start typing here...

# Uno Plataform

Start typing here...