



# Roadmap Capstone - Programming 3

# Table of Contents

BRD .....	2
Space Invaders .....	6
Arquitetura .....	12
Requisitos de Engenharia .....	15
Apresentação Parcial (MidTerm) .....	19
Entrega Final .....	22
Uno Plataform .....	25
Recursos de Áudio .....	28
Recursos de Sprites .....	30
Recursos de Desenvolvimento .....	33

# 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

## O que é o jogo?

Space Invaders é um jogo eletrônico arcade que se tornou um marco na história dos videogames, revolucionando a indústria do entretenimento digital no final dos anos 1970. Criado pela empresa japonesa Taito em 1978 e projetado por Tomohiro Nishikado, o jogo rapidamente se transformou em um fenômeno global que definiria muitos elementos fundamentais dos jogos de nave e shoot 'em up.

## Conceito do Jogo

No jogo, o jogador controla uma nave de defesa terrestre posicionada na parte inferior da tela, movendo-se horizontalmente e disparando contra ondas de invasores alienígenas que descem gradualmente em direção à Terra. O objetivo é simples, porém desafiador: destruir todos os invasores antes que eles alcancem a superfície do planeta.

### Características Principais:

- **Jogabilidade Simples:** Movimento lateral e tiros verticais
- **Progressão de Dificuldade:** Invasores ficam mais rápidos conforme são eliminados
- **Elementos Gráficos Básicos:** Sprites pixelados que se tornaram um ícone da era dos videogames retrô
- **Mecânica de Defesa:** Uso de "barricadas" que protegem temporariamente a nave do jogador

## Explicação Detalhada do Jogo

**Objetivo Principal:** O jogador deve destruir todas as naves alienígenas que formam uma falange na parte superior da tela antes que elas consigam descer e alcançar a parte inferior, onde a nave do jogador está posicionada.

### Mecânicas de Jogo:

#### 1. Controle do Jogador:

- O jogador controla um canhão laser (nave) que só pode se mover para a esquerda e para a direita na parte inferior da tela.
- O disparo é sempre vertical, para cima. Só pode haver um tiro do jogador na tela por vez.

## 2. Comportamento dos Invasores:

- Os alienígenas se movem em bloco, horizontalmente. Quando um alienígena na borda da formação atinge o limite da tela, todo o bloco desce um nível e inverte a direção do movimento.
- À medida que o jogador destrói os invasores, a velocidade de movimento e a frequência dos disparos dos alienígenas restantes aumentam. Isso acontece porque, originalmente, o hardware tinha dificuldade para processar muitos sprites, e com menos inimigos na tela, o jogo rodava mais rápido. Este "bug" se tornou uma característica fundamental da jogabilidade.
- Ocasionalmente, uma nave especial (um "disco voador") cruza o topo da tela, oferecendo mais pontos se for destruída.

## 3. Sistema de Defesa (Barricadas):

- Existem quatro barricadas posicionadas acima da nave do jogador.
- Elas servem como escudos, absorvendo tiros tanto dos inimigos quanto do próprio jogador.
- Cada tiro destrói uma pequena parte da barricada, que vai se deteriorando até desaparecer completamente.

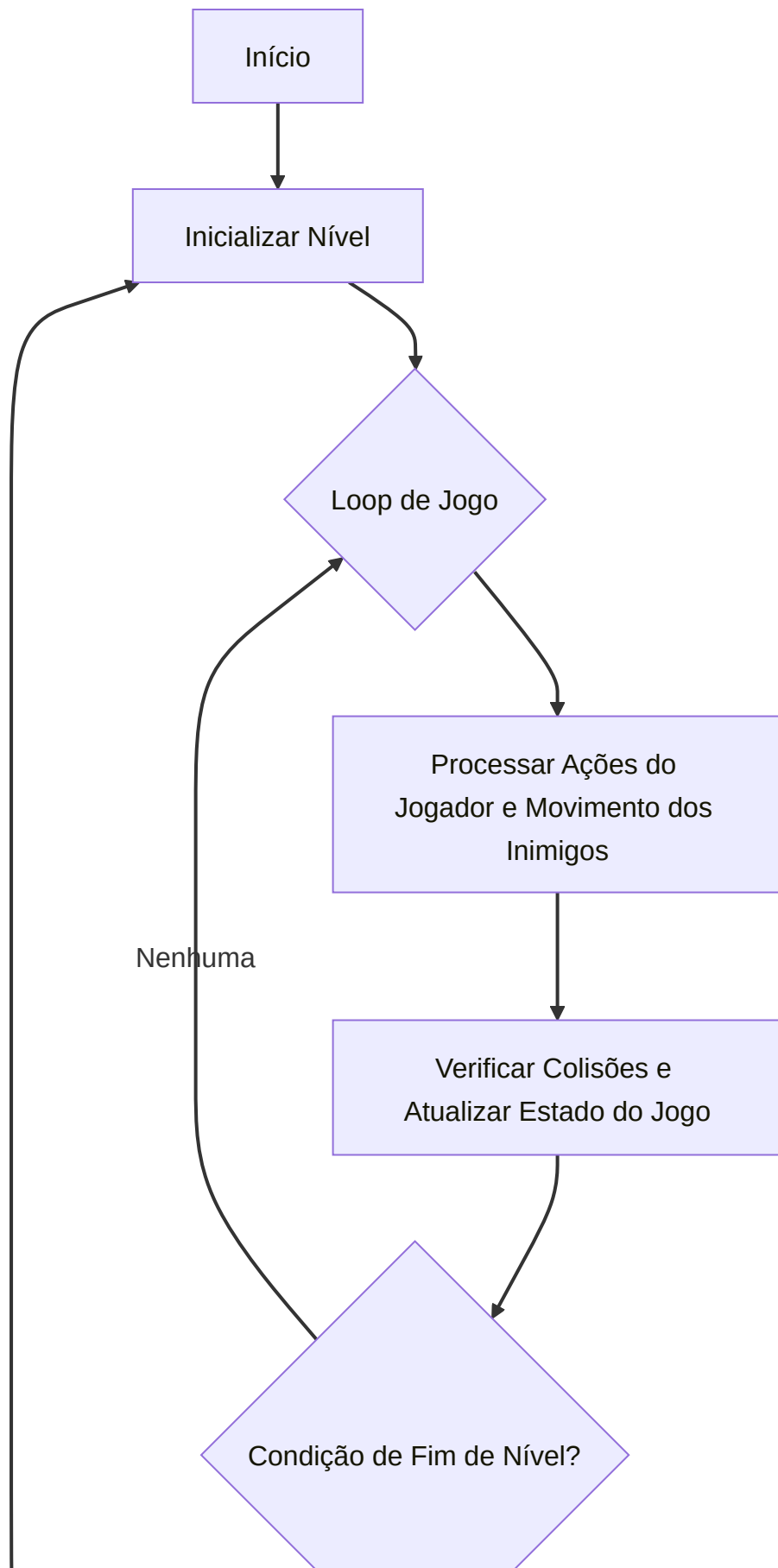
## Condições de Fim de Jogo:

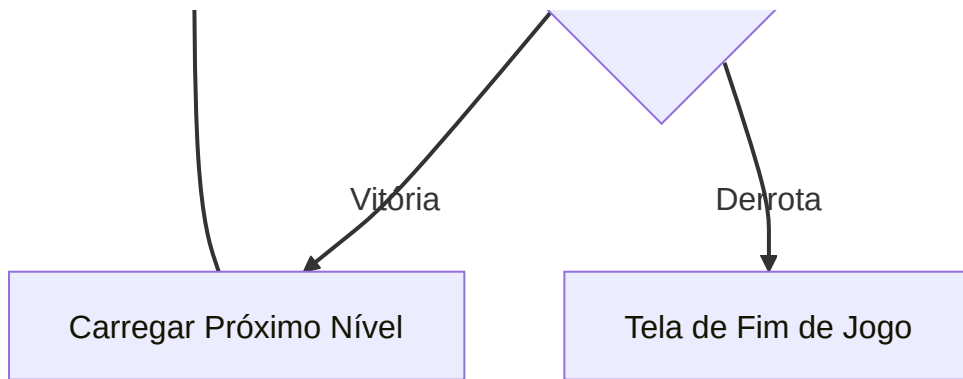
- **Vitória (Nível):** O jogador vence o nível ao destruir todos os invasores da onda. O jogo então avança para o próximo nível, geralmente com os inimigos começando em uma posição mais baixa.
- **Derrota:** O jogo termina se:
  1. Os invasores conseguirem alcançar a parte inferior da tela.



2. A nave do jogador for destruída por um tiro inimigo e não houver mais vidas restantes.

## **Fluxograma do Jogo**





## Impacto Cultural

Space Invaders não foi apenas um jogo, mas um fenômeno cultural que:

- Popularizou os jogos arcade
- Inspirou gerações de desenvolvedores de jogos
- Criou um novo gênero de jogos de nave e tiro
- Ajudou a estabelecer o Japão como potência na indústria de jogos eletrônicos

## Curiosidades:

1. O jogo foi tão popular no Japão que causou uma "escassez de moedas" nas máquinas arcade
2. Inspirou diversos jogos e franquias posteriores
3. Seus gráficos pixelados se tornaram um símbolo da cultura pop dos anos 80

Space Invaders representa mais do que um simples jogo: é um marco tecnológico que ajudou a definir a cultura dos videogames como a conhecemos hoje.

## Links sobre o jogo em ação:

- Space Invaders 1978 - Arcade Gameplay (<https://www.youtube.com/watch?v=MU4psw3ccUI>)
- Space Invaders Creator Thinks Video Games Are Made The Wrong Way (<https://www.youtube.com/watch?v=M7cQoRXjPEY>)

- How Space Invaders Birthed Japanese Games (<https://www.youtube.com/watch?v=Jbn8IRmSq8M>)



Jogue online: Versão online (<https://freeinvaders.org/>)

# Arquitetura

## Technology Stack

A pilha de tecnologia para este projeto foi escolhida para focar em tecnologias .NET modernas e garantir a portabilidade da aplicação de desktop.

- **Linguagem de Programação: C#**
  - Uma linguagem robusta, orientada a objetos, ideal para o desenvolvimento de aplicações complexas como jogos.
- **Plataforma: Uno Platform**
  - Utilizada para construir a aplicação de desktop. Garante que a base de código possa ser potencialmente portada para outras plataformas (Web, Mobile) no futuro.
- **Definição de UI: XAML**
  - Usado para definir de forma declarativa a interface do usuário, separando a lógica da apresentação.
- **Persistência de Dados: Arquivo de Texto (.txt)**
  - Para o salvamento dos placares, será utilizado um arquivo de texto simples, seguindo os requisitos do projeto.

## Wireframe

A seguir, uma representação textual das principais telas do jogo.

### 1. Tela Inicial

```
+-----+
|                                     |
|          CATSTEROIDS               |
|      (Space Invaders Clone)        |
|                                     |
|          [ Iniciar Jogo ]          |
|                                     |
+-----+
```

```

|           [  Placares   ]           |
|           [  Controles  ]           |
|                                     |
+-----+

```

## 2. Tela de Jogo

```

+-----+
| PONTOS: 00420          VIDAS: ❤️❤️ |
+-----+
|                                     |
|           🌸🌸🌸🌸🌸                 |
|               | (laser do jogador) |
|           🌸🌸 🌸🌸🌸                 |
|                                     |
|  _=_*      _=_  _**_      _=_      |
|  * (laser inimigo)                                     |
|                                     |
|           🤖                                     |
+-----+

```


## 3. Tela de Fim de Jogo

```

+-----+
|                                     |
|           FIM DE JOGO                 |
|                                     |
|       Sua pontuação: 01230             |
|                                     |
|  Apelido: [_____] [ Salvar ]           |
|                                     |
|           [ Jogar Novamente ]         |
|           [ Voltar ao Menu ]          |
|                                     |
+-----+

```

## Database Schema

 Isso é apenas um exemplo básico.

O "banco de dados" do projeto será um simples arquivo de texto chamado `scores.txt` para armazenar os placares. A estrutura será baseada em valores separados por vírgula (CSV), facilitando a leitura e a escrita.

**Arquivo:** `scores.txt`

**Formato:** Cada linha representará uma pontuação salva.

Nickname,Score

**Exemplo de Conteúdo:**

```
Player1,1550
GamerX,980
CatLover,2100
```

# Requisitos de Engenharia

## Requisitos Funcionais



ID	Requisito	Descrição	Prioridade
RF 01	Controle da Nave	O sistema deve permitir que o usuário controle uma nave, movimentando-a horizontalmente com as setas direcionais.	Alta
RF 02	Disparo de Lasers	O usuário deve conseguir disparar lasers verticalmente com a barra de espaço.	Alta
RF 03	Pontuação por Inimigo	O sistema deve atribuir pontos ao usuário quando um laser acerta um inimigo, com base na tabela de inimigos ( <a href="#">"Tabela de Inimigos" in "Requisitos de Engenharia"</a> ).	Alta
RF 04	Geração de Inimigos	Inimigos (tipos 1, 2, 3) devem aparecer na tela e se mover sem sair do quadro de jogo.	Alta
RF 05	Inimigo Especial	Um alienígena vermelho especial deve cruzar a tela periodicamente, oferecendo pontos variáveis.	Média
RF 06	Sistema de Vidas	O usuário começa com 3 vidas e perde uma ao ser atingido ou tocar em um alienígena.	Alta
RF 07	Condição de Fim de Jogo	O jogo termina se o jogador perder todas as vidas ou se os alienígenas alcançarem a base.	Alta
RF 08	Barreiras de Proteção	O sistema deve ter 4 blocos de proteção que se degradam com o dano.	Alta
RF 09	Aumento de Dificuldade	A velocidade de movimento e de tiro dos alienígenas deve aumentar conforme eles descem.	Média

	de		
RF 10	Ataque Inimigo	Apenas o tipo de alienígena que vale 40 pontos pode atirar.	Média
RF 11	Recarga de Tiro	O usuário só pode disparar novamente após o tiro anterior atingir um alvo ou sair da tela.	Alta
RF 12	Comportamento dos Inimigos	Os alienígenas não devem colidir com as barreiras de proteção.	Baixa
RF 13	Dano em Inimigos	O usuário deve conseguir destruir naves alienígenas com um único tiro.	Alta
RF 14	Novas Ondas	Uma nova onda de inimigos deve aparecer após a anterior ser destruída, com dificuldade aumentada.	Alta
RF 15	Vida Extra	O jogador ganha uma vida extra a cada 1000 pontos.	Média
RF 16	Exibição de Pontuação	A pontuação deve ser exibida no canto superior esquerdo da tela.	Alta
RF 17	Incremento de Pontuação	A pontuação deve ser incrementada de acordo com a tabela de inimigos ( <a href="#">"Tabela de Inimigos" in "Requisitos de Engenharia"</a> ).	Alta
RF 18	Tela de Fim de Jogo	Ao final do jogo, deve ser possível salvar a pontuação com um apelido ou jogar novamente.	Alta
RF 19	Tela Inicial	O jogo deve ter uma tela inicial com opções para "Novo Jogo", "Placares" e "Controles".	Alta

RF 20	Efeitos Sonoros	O sistema deve ter um som representativo para cada ação principal do jogo.	Média
RF 21	Salvamento de Placar	As informações do painel de pontuação devem ser salvas em um arquivo de texto.	Alta

## Tabela de Inimigos

Tipo de Inimigo	Pontuação	Observações
Inimigo Tipo 1	10	Inimigo padrão, não atira.
Inimigo Tipo 2	20	Inimigo padrão, não atira.
Inimigo Tipo 3	40	Inimigo que pode atirar.
Alienígena Vermelho Especial	50-250 (variável)	Cruza a tela periodicamente.

# Apresentação Parcial (MidTerm)

**Projeto:** Versão Inicial do Jogo "Space Invaders"

**Data da Apresentação:** Semana 4 do Módulo

## Escopo da Entrega

Para a avaliação parcial, o projeto deve implementar um subconjunto dos requisitos funcionais, focando na mecânica central do jogo. A tabela abaixo detalha o que precisa ser entregue.

ID do Requisito	Funcionalidade	Detalhes da Entrega Parcial
RF01	Controle da Nave	A nave do jogador deve se mover horizontalmente.
RF02	Disparo de Lasers	O jogador deve ser capaz de disparar projéteis para cima.
RF04	Geração de Inimigos	Os alienígenas devem ser exibidos na tela, mas podem permanecer estáticos (sem movimento).
RF08	Barreiras de Proteção	As barreiras devem ser exibidas e podem ser destruídas pelos tiros do jogador.
RF13	Dano em Inimigos	Alienígenas devem ser removidos ao serem atingidos por um tiro.
RF16	Exibição de Pontuação	A pontuação deve ser visível na tela.
RF17	Incremento de Pontuação	A pontuação deve ser atualizada sempre que um inimigo for destruído.
RF19	Tela Inicial	O jogo deve apresentar uma tela inicial simples.
RF20	Efeitos Sonoros	Deve haver sons para as ações principais (tiro, destruição).

## Lógica de Jogo Simplificada

- **Condição de Vitória:** Para esta entrega, o jogo termina quando o jogador atingir **500 pontos**.

- **Armazenamento:** A pontuação é mantida apenas em memória durante a sessão de jogo.
- **Inimigos:** O movimento complexo dos inimigos (descida, aumento de velocidade) não é necessário nesta fase.

# Entrega Final

**Projeto:** Versão Completa do Jogo "Space Invaders"

**Data da Apresentação:** Semana 7 do Módulo

## Escopo da Entrega

Para a avaliação final, o projeto deve implementar **todos os requisitos funcionais** definidos na documentação. Isso inclui completar as funcionalidades da entrega parcial e adicionar as mecânicas avançadas listadas abaixo.

## Funcionalidades a Serem Completadas ou Adicionadas

ID do Requisito	Funcionalidade	Detalhes da Entrega Final
RF04	Movimento dos Inimigos	Implementar o movimento em bloco, com descida e inversão de direção nas bordas.
RF05	Inimigo Especial	A nave vermelha deve aparecer periodicamente no topo da tela.
RF06	Sistema de Vidas	O jogador deve começar com 3 vidas e perdê-las ao ser atingido.
RF07	Condições de Fim de Jogo	O jogo deve terminar se as vidas acabarem ou se os inimigos alcançarem a base.
RF09	Aumento de Dificuldade	A velocidade dos inimigos deve aumentar conforme o jogo avança.
RF10	Ataque Inimigo	O inimigo de 40 pontos deve ser capaz de atirar no jogador.
RF14	Novas Ondas	O jogo deve gerar uma nova onda de inimigos após a anterior ser destruída.
RF15	Vida Extra	O jogador deve ganhar uma vida extra a cada 1000 pontos.
RF18	Tela de Fim de Jogo	Implementar a tela de "Game Over" com opções para salvar o placar e jogar novamente.
RF19	Menu Completo	O menu inicial deve ser funcional, com opções para "Novo Jogo", "Placares" e "Controles".



RF21	Salvamento de P lacar	A pontuação deve ser salva em um arquivo de texto (s cores.txt).
------	--------------------------	---

## Requisitos Já Entregues (MidTerm)

As seguintes funcionalidades, já implementadas na entrega parcial, devem estar completamente integradas e funcionais na versão final:

- RF01: Controle da Nave
- RF02: Disparo de Lasers
- RF08: Barreiras de Proteção
- RF13: Dano em Inimigos
- RF16 & RF17: Sistema de Pontuação
- RF20: Efeitos Sonoros

# Uno Platform

A Uno Platform é uma plataforma de código aberto que permite a criação de aplicações multi-plataforma usando C# e XAML. Com ela, você pode desenvolver aplicações que rodam nativamente em WebAssembly, iOS, Android, macOS, Linux e Windows a partir de uma única base de código.

## Instalação e Setup

Para começar a desenvolver com Uno Platform, você precisará instalar algumas ferramentas e SDKs. Siga os passos abaixo para configurar seu ambiente de desenvolvimento.

### Pré-requisitos

Certifique-se de ter os seguintes pré-requisitos instalados em seu sistema:

- **Visual Studio 2022** (Community, Professional ou Enterprise) com as seguintes cargas de trabalho:
  - Desenvolvimento para desktop com .NET
  - Desenvolvimento móvel com .NET
  - Desenvolvimento multiplataforma de interface do usuário .NET
  - Desenvolvimento Web ASP.NET e Web
- **SDK do .NET** (geralmente incluído com o Visual Studio, mas pode ser baixado separadamente se necessário).
- **Node.js** (necessário para o desenvolvimento WebAssembly).

### Instalação do Uno Platform Solution Templates

Os templates de solução da Uno Platform facilitam a criação de novos projetos. Você pode instalá-los via linha de comando:

```
dotnet new install Uno.Templates
```

## Criando um Novo Projeto Uno Platform

Após instalar os templates, você pode criar um novo projeto Uno Platform usando o Visual Studio ou a linha de comando.

### Via Visual Studio

1. Abra o Visual Studio 2022.
2. Selecione "Criar um projeto".
3. Pesquise por "Uno Platform" e escolha o template "Uno Platform App".
4. Siga as instruções para configurar seu projeto (nome, localização, etc.).

### Via Linha de Comando

Você pode criar um novo projeto Uno Platform a partir do terminal:

```
dotnet new unoapp -n MyUnoApp
```

Substitua `MyUnoApp` pelo nome desejado para o seu projeto.

## Executando o Projeto

Após criar o projeto, você pode executá-lo em diferentes plataformas. Por exemplo, para rodar a versão WebAssembly:

```
dotnet run -f net8.0-browser
```

Para rodar a versão Windows (se estiver no Windows):

```
dotnet run -f net8.0-windows
```

Para outras plataformas, você pode usar o Visual Studio para selecionar o target desejado e executar a aplicação.

## Recursos Adicionais

- Documentação Oficial Uno Platform (<https://platform.uno/docs/>)

- Tutoriais e Exemplos da Uno Platform (<https://platform.uno/platform/>)
- Comunidade Uno Platform (Discord, GitHub Discussions)  
(<https://platform.uno/community/>)

# Recursos de Áudio

## 1. Repositório Abrangente de Assets

- **Repositório Space Invaders (Bruno Nascimento):** Este repositório contém uma coleção de sons que podem ser utilizados no projeto.
- Link (<https://github.com/brunotnasc/space-invaders>)

## 2. Músicas e Trilhas Sonoras

- **Space Invaders (Mega Drive) - Trilha Sonora Completa (Archive.org):** A trilha sonora da versão de Mega Drive do jogo, ideal para referência ou como música de fundo.
- Link ([https://archive.org/details/md\\_music\\_space\\_invaders\\_91/01+-+Opening+Theme+--+Noriyuki+Iwadare.flac](https://archive.org/details/md_music_space_invaders_91/01+-+Opening+Theme+--+Noriyuki+Iwadare.flac))

## 3. Efeitos Sonoros (SFX)

*Esta seção pode ser preenchida com links para SFX de tiros, explosões, etc., caso o repositório acima não seja suficiente.*

## 4. Geração de Efeitos Sonoros com IA

Para criar efeitos sonoros personalizados ou preencher lacunas nos assets existentes, ferramentas de inteligência artificial podem ser extremamente úteis. Elas permitem gerar sons a partir de descrições textuais.

### Prompt Eficaz para Geração de SFX

Ao usar IAs generativas de áudio, a clareza e a especificidade do prompt são cruciais. Considere o seguinte modelo:

```
"[tipo de som, ex: tiro de laser, explosão, som de nave se movendo, som de coleta de item], [características, ex: agudo,
```

grave, metálico, eletrônico, curto, longo, com eco], [estilo, ex: retro, 8-bit, futurista, arcade]"

### Exemplos:

- tiro de laser, agudo, eletrônico, curto, estilo arcade
- explosão, grave, com reverberação, estilo retro
- som de nave espacial se movendo, contínuo, futurista, com leve zumbido
- som de coleta de item, brilhante, curto, 8-bit

### Ferramentas de IA Recomendadas

- **ElevenLabs:** Embora seja mais conhecida por geração de voz, possui recursos experimentais para SFX e pode ser útil para sons mais abstratos ou ambientes.
- **AIVA (Artificial Intelligence Virtual Artist):** Focada em música, mas pode gerar trechos curtos que servem como base para SFX.
- **Google Magenta Studio:** Uma coleção de ferramentas de código aberto que exploram o aprendizado de máquina para criação de música e arte, incluindo sons.
- **OpenAI Jukebox:** Gera música com vocais em vários estilos e gêneros, mas pode ser adaptado para criar texturas sonoras.



**Dica:** Muitas vezes, o som gerado por IA pode precisar de edição em um software de áudio (como Audacity ou Ableton Live) para ajustar volume, cortar, adicionar efeitos ou mixar com outros sons.



**Nota sobre Licenças:** Verifique sempre as políticas de uso e licenciamento dos sons gerados por IA, pois elas podem variar entre as plataformas e o uso comercial pode ter restrições.

# Recursos de Sprites

A seguir, uma lista de links verificados para encontrar sprites e assets para o projeto. Lembre-se sempre de verificar a licença de cada ativo antes de usá-lo.

## 1. Repositório Abrangente de Assets

- **Repositório Space Invaders (Bruno Nascimento):** Este repositório contém uma coleção de imagens e sons que podem ser utilizados no projeto.
  - Link (<https://github.com/brunotnasc/space-invaders>)

## 2. Sprites Originais (Referência)

- **Classic Gaming - Space Invaders Graphics:** Rip dos sprites originais do jogo, excelente para referência de estilo e design.
  - Link (<https://classicgaming.cc/classics/space-invaders/graphics>)

## 3. Pacotes Gerais de Assets (Naves, Inimigos, Tiros)

- **Busca Geral no Itch.io por "Space Shooter":** Uma ampla gama de pacotes de assets, incluindo naves para o jogador, inimigos, projéteis e mais.
  - Link 1 (<https://itch.io/search?q=space-shooter-assets>)
  - Link 2 (<https://itch.io/search?q=space+shooter>)
- **Free Spaceship Pixel Art (CraftPix):** Contém 3 naves espaciais animadas que podem servir como inimigos ou para o jogador.
  - Link (<https://craftpix.net/freebies/free-spaceship-pixel-art-sprite-sheets/>)
- **Space Shooter Redux (GDevelop):** Um pacote completo com mais de 140 objetos, ideal para prototipagem rápida.
  - Link (<https://gdevelop.io/asset-store/free/space-shooter-redux-space-shooter>)

## 4. Efeitos de Explosão

- **Pixel Art Explosions by unTied Games (Itch.io):** Um pacote com 5 efeitos de explosão em alta qualidade, perfeitos para destruição de naves.
- Link (<https://untiedgames.itch.io/five-free-pixel-explosions>)

## 5. Cenários e Planos de Fundo

- **Busca por Planos de Fundo no Itch.io:** Busca por planos de fundo com estilo retrô para o cenário do jogo.
- Link (<https://itch.io/search?type=games&q=space+backgrounds+pack+retro>)

## 6. Geração de Sprites com IA

Para criar sprites personalizados ou preencher lacunas nos assets existentes, a inteligência artificial pode ser uma ferramenta poderosa. Abaixo, um prompt eficaz e algumas recomendações de ferramentas.

### Prompt Eficaz para Geração de Sprites

Ao usar IAs generativas de imagem, a clareza e a especificidade do prompt são cruciais. Considere o seguinte modelo:

```
"pixel art, [tipo de sprite, ex: nave espacial, inimigo, projétil, explosão], [estilo, ex: 8-bit, 16-bit, retro, futurista], [cor principal], [detalhes específicos, ex: com asas pontudas, olhos grandes, rastro de fumaça, animação de voo], fundo transparente, folha de sprites"
```

### Exemplos:


- pixel art, nave espacial do jogador, 8-bit, verde e roxo, com canhões duplos na frente, animação de voo, fundo transparente, folha de sprites




- pixel art, inimigo alienígena, 16-bit, azul e laranja, formato de polvo, com tentáculos se movendo, fundo transparente, folha de sprites
- pixel art, explosão, retro, amarelo e vermelho, com fumaça e faíscas, fundo transparente, folha de sprites

## Ferramentas de IA Recomendadas

- **Midjourney:** Conhecido pela alta qualidade e estilo artístico. Requer assinatura.
- **DALL-E 3 (via ChatGPT Plus ou Bing Image Creator):** Ótimo para gerar imagens a partir de texto, com boa compreensão de prompts complexos. Acesso via assinatura ou gratuito no Bing.
- **Stable Diffusion:** Uma opção de código aberto que pode ser executada localmente ou via serviços online. Oferece grande controle e flexibilidade.
- **Lexica Art:** Um motor de busca e galeria de imagens geradas por IA, útil para encontrar inspiração e prompts.

 **Dica:** Após gerar os sprites, pode ser necessário um pequeno ajuste manual em um editor de pixel art (como Aseprite ou Piskel) para garantir a consistência e otimização para o jogo.

 **Nota sobre Licenças:** Verifique sempre as políticas de uso e licenciamento das imagens geradas por IA, pois elas podem variar entre as plataformas.

# Recursos de Desenvolvimento

Aqui estão links para recursos essenciais para o desenvolvimento do jogo, focando nas tecnologias utilizadas:

## Linguagem de Programação: C#

- Documentação Oficial C# (Microsoft Learn) (<https://learn.microsoft.com/pt-br/dotnet/csharp/>)
- TutorialsPoint C# Tutorial (<https://www.tutorialspoint.com/csharp/index.htm>)
- GeeksForGeeks C# (<https://www.geeksforgeeks.org/c-sharp/csharp-programming-language/>)

## Plataforma: Uno Platform

- Documentação Oficial Uno Platform (<https://platform.uno/docs/>)
- Uno Platform GitHub Repository (<https://github.com/unoplatform/uno>)
- Uno Platform Samples (<https://github.com/unoplatform/uno.samples>)

## Definição de UI: XAML

- XAML Overview (Microsoft Learn) (<https://learn.microsoft.com/pt-br/dotnet/desktop/xaml-overview/?view=netdesktop-8.0>)
- WPF XAML Tutorial (TutorialsPoint) ([https://www.tutorialspoint.com/wpf/wpf\\_xaml\\_tutorial.htm](https://www.tutorialspoint.com/wpf/wpf_xaml_tutorial.htm))
- Introdução ao XAML (Macoratti) ([http://www.macoratti.net/10/05/c\\_xaml1.htm](http://www.macoratti.net/10/05/c_xaml1.htm))

## Ferramentas e IDEs

- Visual Studio (IDE para .NET/C#) (<https://visualstudio.microsoft.com/>)
- Visual Studio Code (Editor de código leve) (<https://code.visualstudio.com/>)
- .NET SDK (Necessário para compilar e rodar aplicações .NET)  
(<https://dotnet.microsoft.com/download>)