

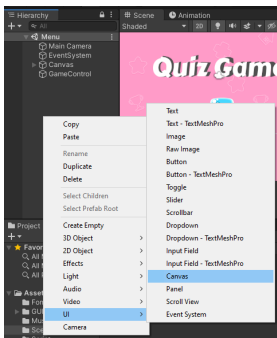
➤ **Criação do projeto na Unity**

1. Abrir o Unity Hub;
2. Selecionar o botão 'NEW';
3. Escolher o **Template 2D**;
4. Escrever "QUIZ GAME" em 'Project Name';
5. Clicar em 'CREATE' e aguardar o projeto inicializar.

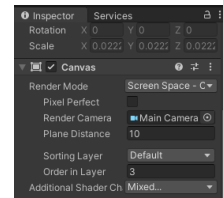
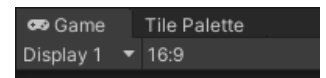
➤ **Importação dos Assets**

1. Fazer o download da pasta 'Assets' compartilhada no drive;
2. Abrir a pasta do projeto (**Unity Projects/QUIZ GAME**) no Explorador de Arquivos;
3. Acessar o diretório **QUIZ GAME/Assets**;
4. Adicionar a pasta **GUI** localizada na pasta compartilhada do *drive* do projeto.

➤ **Preparação do ambiente**



- Alterar proporção da tela na aba *Game* de *Free Aspect* para 16:9;
- Na aba *Hierarchy* no espaço vazio clicar com o botão direito do mouse e selecionar **UI -> Canvas**;
- Selecionar **Canvas** e na guia *Canvas* mudar a opção *Render Mode* de *Screen Space - Overlay* para *Screen Space - Camera*;
- Na guia *Canvas Scaler* mudar a opção *UI Scale Mode* de *Constant Pixel Size* para *Scale With Screen Size* e arrastar o objeto *Main Camera* para dentro do *Render Camera* e alterar *Plane Distance* para 10;



➤ **Frontend do Jogo**

- Menu:
 - Navegar no diretório **Unity Projects\QUIZ GAME\Assets\Scenes**;
 - Clicar com o botão direito na cena e em *Rename* para renomear *Sample Scene* para *Menu*;
 - Na caixa de diálogo, que será aberta, selecionar *Reload*;
 - Clicar com o botão direito em **Canvas -> UI -> Image** e selecionar *Image* e clicar na tecla F2 para renomear para 'BG';
 - Arrastar a imagem 'BG.png' do diretório **Unity Projects\QUIZ GAME\Assets\GUI\png** na guia *Image* arrastar a imagem 'BG.png' dentro de *Source Image*;
 - Selecionar a ferramenta *Scale* e clicar duas vezes em BG na aba *Hierarchy* e usar a ferramenta para aumentar a escala da imagem até que *Width* e *Height* na aba *Inspector* estejam equivalentes a 100;
 - Para criar um título, clicar com o botão direito em **Canvas -> UI -> Text - TextMeshPro** e selecionar *Text (TMP)* e clicar na tecla F2 para renomear para 'Title';
 - Para criar novas fontes clicar em *Window* no menu superior e selecionar **TextMeshPro -> Font Asset Creator** e arrastar a fonte a ser

convertida para dentro do Source Font File e clicar em *Generate Font Atlas* e na sequência clicar em *Save* e salvar na pasta *Font*;

- Escrever em *Text Input* o título do jogo 'QUIZ GAME';
- Para criar o botão Play, clicar com o botão direito em **Canvas -> UI -> Legacy -> Button** e selecionar *Button* e clicar na tecla F2 para renomear para 'BtnPlay';
- Deletar o Text (TMP) do botão, pois será apenas uma imagem;
- Na aba Inspector, alterar *Width* e *Height* para 100;
- E mover com a ferramenta *Move Tool* o botão para baixo do título, centralizando-o;
- Arrastar a imagem 'Button_45' para a guia *Image* dentro de *Source Image*.
- Para criar o botão de sair, duplicar o objeto 'BtnPlay' (Ctrl+c Ctrl+v) e renomear para 'BtnClose';
- Na aba Inspector, alterar *Width* e *Height* para 50 e arrastar a imagem 'Button_35' para dentro de *Source Image*;
- E mover com a ferramenta *Move Tool* o botão para a região superior direita;
- Por fim, clicar no menu superior em **File -> Build Profiles** e em *Add Open Scenes*;
- Deve-se ter uma tela semelhante à Figura 1.

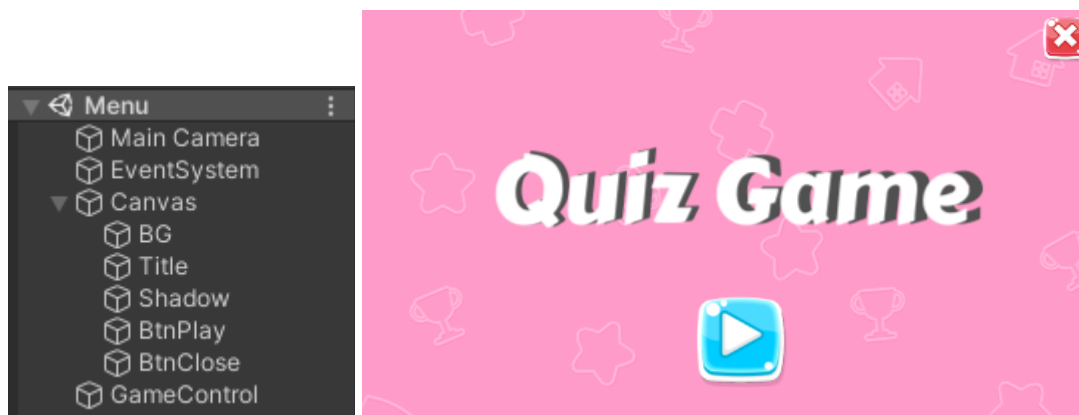


Figura 1. Menu do jogo

○ Question:

- Clicar no menu superior em **File -> Save as** e salvar no diretório **Unity Projects\QUIZ GAME\Assets\Scenes** com o nome de 'Question';
- Na aba *Hierarchy* em *Canvas* remover o *Title*;
- Para criar um objeto vazio para armazenar a questão, clicar com o botão direito em **Canvas -> Create Empty** e clicar na tecla F2 para renomear para 'Question';
- Clicar com o botão direito em **Canvas -> UI -> Image** e selecionar *Image* e clicar na tecla F2 para renomear para 'Panel';
- Arrastar a imagem 'Window_31' para dentro de *Source Image* e alterar *Width* e *Height* para que fique semelhante ao painel de 'Statement' na Figura 2;

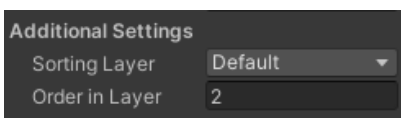
- Para criar o enunciado da questão, clicar com o botão direito em **Canvas -> UI -> Legacy -> Text**, selecionar *Text* e clicar na tecla F2 para renomear para 'Statement', em seguida, arrastar para dentro do objeto de imagem 'Panel';
- Selecionar *Statement* e na guia *Paragraph* marcar a opção *BestFit* e definir *Min* com o valor 12 e *Max* com o valor 23 e na guia *Text Input* escrever "Statement";
- Duplicar o objeto 'Panel' (Ctrl+c Ctrl+v), renomear para 'A';
- Selecionar *Statement* e renomear para 'AlternativeA', na guia *Paragraph* com a opção *BestFit* marcada, definir *Min* com o valor 10 e *Max* com o valor 25 e na guia *Text Input* escrever "Alternative A";
- Duplicar o objeto 'A' (Ctrl+c Ctrl+v) e renomear o objeto para 'B' e o texto 'AlternativeA' para 'AlternativeB', e na guia *Text Input* escrever "Alternative B";
- Repetir a etapa anterior para 'C' e 'D';
- Para criar o texto que informará o índice da questão, clicar com o botão direito em **Canvas -> UI -> Legacy -> Text**, selecionar *Text*, renomear para 'InfoQuestion', e na guia *Text Input* escrever "Respondendo 1 de 5";
- Clicar no menu superior em **File -> Build Profiles** e em *Add Open Scenes*;
- Por fim, deve se ter uma tela semelhante à Figura 2.

Figura 2. Estrutura das questões



○ Final Grade:

- Clicar no menu superior em **File -> Save as** e salvar no diretório **Unity Projects\QUIZ GAME\Assets\Scenes** com o nome de 'FinalGrade';
- Na aba *Hierarchy* em *Canvas* arrastar os objetos *Panel* e *InfoQuestion* para fora do objeto *Question* e deletar o objeto *Question*;
- Alterar a imagem do objeto 'Panel' em *Source Image* para 'Window_4', usar a ferramenta *Scale* para redimensionar o quadro e na guia *Sprite Renderer* alterar *Order in Layer* para 1;
- Deletar o objeto *Statement*;
- Clicar com o botão direito em **Canvas -> UI -> Image** e selecionar *Image*, renomear para 'Title', alterar a imagem em *Source Image* para 'Window_46', usar a ferramenta *Scale* para redimensionar o quadro e na guia *Sprite Renderer* alterar *Order in Layer* para 2;



- Renomear o objeto InfoQuestion para 'TxtInfo', alterar *FontSize* para 30 e escrever em *TextInput* "Acertos: 5 de 5 perguntas";
- Duplicar o objeto 'TxtInfo' (Ctrl+c Ctrl+v), renomear para 'TxtGrade', alterar *Font Size* para 50 e escrever em *TextInput* "10 pts";
- Alterar a imagem do objeto 'Panel' em Source Image para 'Window_31';
- Para criar o botão Home, clicar com o botão direito em **Canvas -> UI -> Legacy -> Button** e selecionar *Button* e clicar na tecla F2 para renomear para 'BtnHome';
- Deletar o Text (TMP) do botão, pois será apenas uma imagem;
- Adicionar a imagem em *Source Image* para 'Button_49' e alterar *Width* e *Height* para 60;
- Mover com a ferramenta *Move Tool* o botão de acordo com a Figura 3;
- Duplicar o objeto 'BtnHome' (Ctrl+c Ctrl+v), renomear para 'BtnReplay' e alterar a imagem em *Source Image* para 'Button_73';
- Clicar no menu superior em **File -> Build Profiles** e em *Add Open Scenes*;
- Por fim, deve se ter uma tela semelhante à Figura 3.



Figura 3. Pontuação final

➤ **Backend do Jogo**

- Na aba *Project* no diretório **Unity Projects\QUIZ GAME\Assets**, criar uma pasta com o nome de '*Scripts*';
- Selecionar a nova pasta e clicar com o botão direito na aba *Project* em **Create -> Scripting -> Empty C# Script** e renomear o script para **Controls**, onde serão descritos os principais controladores do jogo;
- Abrir o novo script **Controls** com o editor de texto *Visual Studio Code* e implementar o código a seguir:

```
using System.Collections;
```

```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class Controls : MonoBehaviour
{
    public GameObject btnMusicOff;
    public AudioSource audioSource;

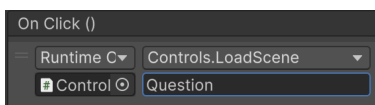
    public void LoadScene(string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }

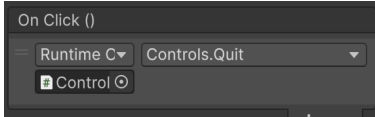
    public void Quit()
    {
        Application.Quit();
    }

    public void AudioOff()
    {
        audioSource.Stop();
        btnMusicOff.gameObject.SetActive(true);
    }

    public void AudioOn()
    {
        audioSource.Play();
        btnMusicOff.gameObject.SetActive(false);
    }
}
```

- Agora, basta integrar o código implementado aos objetos do jogo;
- Na cena **Menu**:
 - Arrastar o *script* 'Controls' para dentro da aba Hierarchy;
 - Selecionar o objeto *BtnPlay* na aba **Hierarchy->Canvas->BtnPlay** e na função *OnClick()* clicar no '+' e arrastar o objeto *Controls* para dentro do campo que diz 'None (object)' e em 'No Function' selecionar a função **Controls -> LoadScene (string)**;
 - Em seguida, deve-se preencher o campo vazio com o nome da próxima cena a ser carregada, nesse caso, será 'Question';





- Selecionar o objeto *BtnClose* na aba **Hierarchy->Canvas->BtnPlay** e na função *OnClick()* clicar no '+' e arrastar o objeto *Controls* para dentro do campo que diz 'None (object)' e em 'No Function' selecionar a função **Controls ->Quit ()**;
- Novamente, na aba *Project* em **Create -> Scripting -> Empty C# Script** e renomear o *script* para **Question**, onde serão definidas as questões e o funcionamento do jogo;
- Abrir o novo *script Question* com o VSCode e implementar o código a seguir:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class Question : MonoBehaviour
{
    public Text statement;
    public Text alternativeA;
    public Text alternativeB;
    public Text alternativeC;
    public Text alternativeD;
    public Text infoQuestion;

    public string[] statements;
    public string[] alternativesA;
    public string[] alternativesB;
    public string[] alternativesC;
    public string[] alternativesD;
    public string[] correctAnswers;

    private int questionId;

    private float correctQuestions;
    private float questions;
    private float average;
    private int finalGrade;

    public void InitializeQuestion()
    {
        statement.text = statements[questionId];
```

```
        alternativeA.text = alternativesA[questionId];
        alternativeB.text = alternativesB[questionId];
        alternativeC.text = alternativesC[questionId];
        alternativeD.text = alternativesD[questionId];
        infoQuestion.text = "Respondendo " + (questionId+1) + " de "
+ questions.ToString();
    }

    // Start is called before the first frame update
    void Start()
    {
        questionId = 0;
        questions = statements.Length;
        InitializeQuestion();
    }

    public void Answer(string alternative)
    {
        if (alternative == "A")
        {
            if (alternativesA[questionId] ==
correctAnswers[questionId])
            {
                correctQuestions += 1;
            }
        }
        else if (alternative == "B")
        {
            if (alternativesB[questionId] ==
correctAnswers[questionId])
            {
                correctQuestions += 1;
            }
        }
        else if (alternative == "C")
        {
            if (alternativesC[questionId] ==
correctAnswers[questionId])
            {
                correctQuestions += 1;
            }
        }
    }
}
```

```
    }  
    else if (alternative == "D")  
    {  
        if (alternativesD[questionId] ==  
correctAnswers[questionId])  
        {  
            correctQuestions += 1;  
        }  
    }  
    NextQuestion();  
}  
  
public void NextQuestion()  
{  
    questionId += 1;  
    if (questionId <= (questions-1))  
    {  
        InitializeQuestion();  
    }  
    else  
    {  
        average = 10 * (correctQuestions / questions);  
        finalGrade = Mathf.RoundToInt(average);  
  
        PlayerPrefs.SetInt("finalGradeTemp", finalGrade);  
        PlayerPrefs.SetInt("correctQuestionsTemp",  
(int) correctQuestions);  
        SceneManager.LoadScene("FinalGrade");  
    }  
}  
}
```

- Na cena **Question**:
 - Arrastar o *script* 'Question' para dentro da aba Hierarchy.
 - Na aba Inspector, arrastar todos os objetos que aparecem de acordo com seu respectivo nome, como mostra na Figura 4.
 - Para todos os vetores ('Statements', 'Alternatives A', ... e 'Correct Answers') definir o Size como 5, pois serão cinco questões ao todo e adicionar os enunciados em *Statements*, as alternativas A's em *AlternativesA*, e assim por diante, obedecendo a ordem que aparece ao acessar o link [Questões](#);

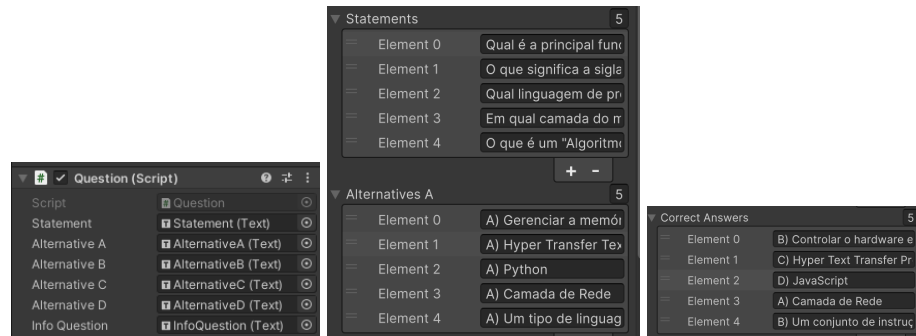


Figura 4. Relacionando objetos do jogo com o código

- Selecionar o objeto A na aba **Hierarchy->Canvas->Question->A**
 - Verificar que existe o componente Button, caso não exista, adicioná-lo;
 - Na função `OnClick()` clicar no '+' e arrastar o objeto *Question* para dentro do campo que diz 'None (object)' e em 'No Function' selecionar a função **Question -> Answer (string)**;
 - Em seguida, deve-se preencher o campo vazio com a resposta, nesse caso, será 'A', pois é o nome do objeto no jogo;
 - Por fim, deve-se repetir as duas etapas anteriores para B, C e D.
- Novamente, na aba *Project* em **Create -> Scripting -> Empty C# Script** e renomear o script para **FinalGrade**, onde serão definidas a nota e a quantidade de acertos;
 - Abrir o novo script **FinalGrade** com o *VSCode* e implementar o código a seguir:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class FinalGrade : MonoBehaviour
{
    public Text txtGrade;
    public Text txtInfo;

    private int grade;
    private int correctQuestions;

    // Start is called before the first frame update
    void Start()
    {
        grade = PlayerPrefs.GetInt("finalGradeTemp");
    }
}
```

```
correctQuestions =  
PlayerPrefs.GetInt("correctQuestionsTemp");  
  
txtGrade.text = grade.ToString() + " pts";  
txtInfo.text = "Acertos: " + correctQuestions.ToString() + "  
de 5 perguntas";  
  
}  
}
```

- Na cena **FinalGrade**:
 - Arrastar o *script* 'FinalGrade' para dentro da aba Hierarchy.
 - Na aba Inspector, arrastar todos os objetos que aparecem de acordo com seu respectivo nome, como mostra na Figura 5.

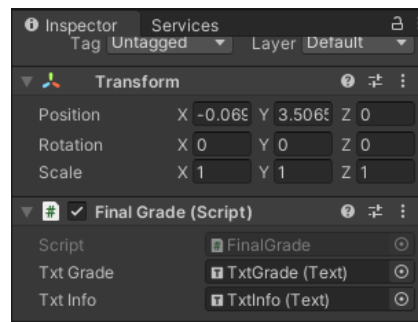
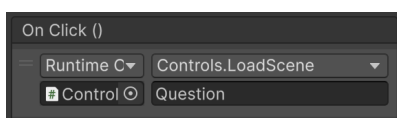
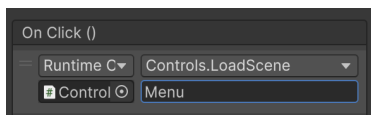


Figura 5. Relacionando objetos do jogo com o *Script Final Grade*

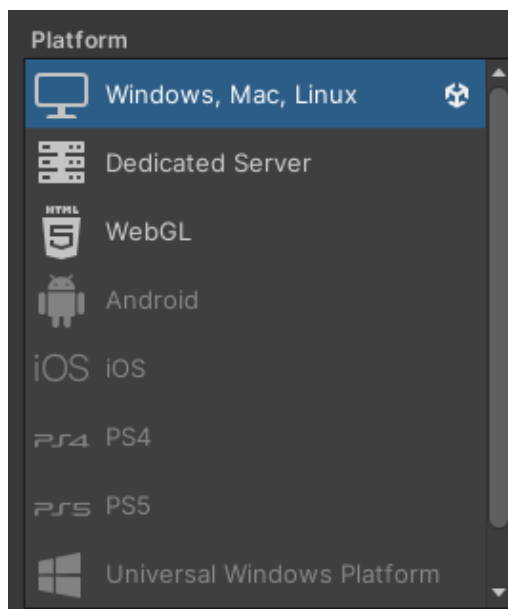
- Arrastar o *script* Controls para dentro da aba Hierarchy.
- Selecionar o objeto *BtnHome* na aba **Hierarchy->Canvas->BtnHome** e na função *OnClick()* clicar no '+' e arrastar o objeto *Controls* para dentro do campo que diz 'None (object)' e em 'No Function' selecionar a função **Controls -> LoadScene (string)**;
- Em seguida, deve-se preencher o campo vazio com o nome da próxima cena a ser carregada, nesse caso, será 'Menu', pois esse botão retorna para o menu do jogo;
- Por fim, selecionar o objeto *BtnReplay* na aba **Hierarchy->Canvas->BtnReplay** e na função *OnClick()* clicar no '+' e arrastar o objeto *Controls* para dentro do campo que diz 'None (object)' e em 'No Function' selecionar a função **Controls -> LoadScene (string)**;
- Em seguida, deve-se preencher o campo vazio com o nome da próxima cena a ser carregada, nesse caso, será 'Question', pois esse botão retorna para as questões do jogo.



Gerando a build do jogo

O processo de *build* (construção, em tradução livre), de um jogo, nada mais é do que gerar a aplicação final do projeto, o que no nosso caso é um arquivo executável, pois estamos trabalhando no Sistema Operacional Windows e queremos gerar uma aplicação para esse mesmo Sistema Operacional.

Dentro da Unity temos diferentes opções de *build* de um projeto, tal como mostra a figura abaixo:



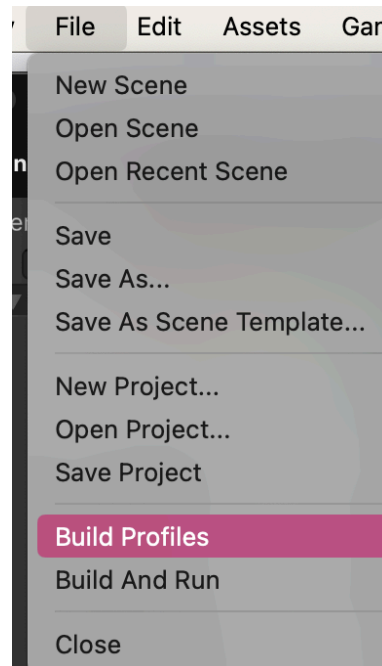
Exemplos de plataformas comportadas pelo Build dentro da Unity

Fonte: Própria

O ícone/logo da Unity ao lado da opção nos diz que o nosso Editor está pronto e configurado para gerar uma aplicação para Windows, Mac e Linux, nesse caso. Ou seja, consegue gerar uma aplicação *standalone* para rodar em uma máquina que possua qualquer um desses 3 Sistemas Operacionais. Para acessar outras opções de plataforma, é necessário adicioná-las no Editor, como se fosse uma dependência ou um pacote que será inserido e ficará disponível ali para escolha.

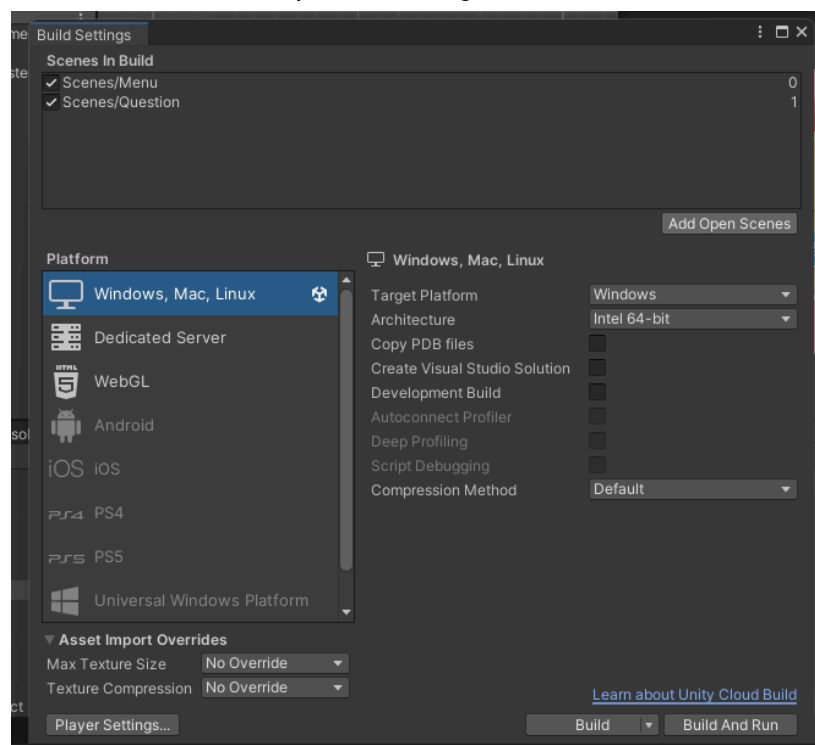
Processo de Build

→ **Passo 1:** Acessar o menu **File -> Build Profiles** (Perfis de Build).



Caminho para o Build Profiles

A janela que irá abrir é semelhante à próxima imagem:



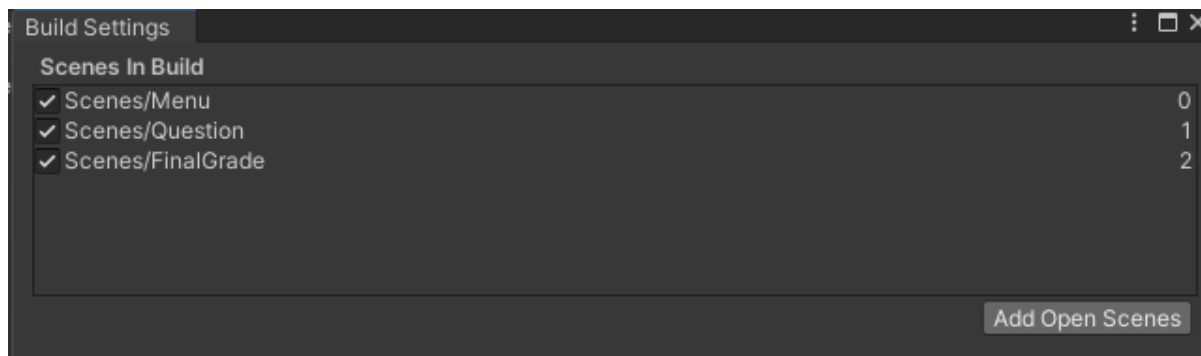
Menu Build Profiles

→ **Passo 2:** Certificar-se de que todas as cenas do jogo estão adicionadas nesse menu.

Se olharmos na parte superior deste menu, temos a janela **Scenes in Build**. Todas as cenas que estão ali, serão as que aparecerão na aplicação final, depois do processo de

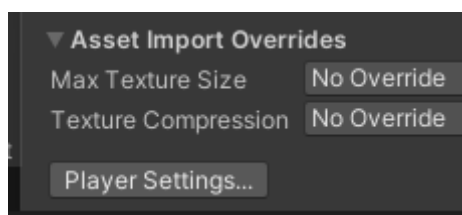
Build. É extremamente importante que a cada vez que uma cena for finalizada no seu projeto, esse menu seja acessado e a cena atual seja adicionada nessa lista.

Para fazer isso, basta clicar na opção **Add Open Scenes**. Com isso, a cena atual deverá aparecer na lista e estará pronta para ir para a Build junto com as demais. A imagem abaixo mostra a cena atual que está aberta no Editor, chamada *FinalGrade*, presente na lista após termos clicado na opção de adicionar a cena.



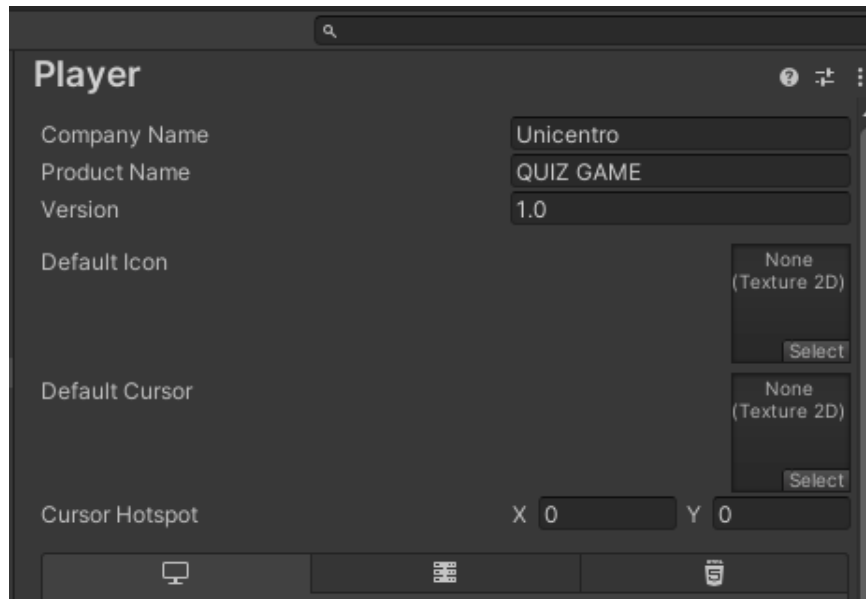
Cena *FinalGrade* na lista *Scene List*

→ **Passo 3 (opcional):** Acessar o menu *Player Settings* para configurar os detalhes de *Build* do jogo.



Opção *Player Settings*

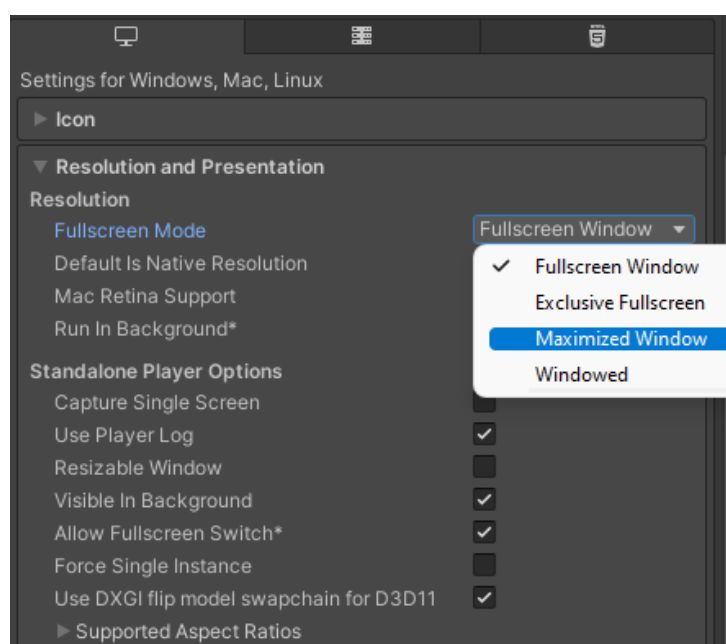
Este menu está presente no final da janela, do lado inferior esquerdo. O *Player Settings* abre outra janela, chamada *Player*. Essa janela também pode ser acessada através do caminho **File -> Project Settings -> Player** (no menu lateral esquerdo).



Menu Player

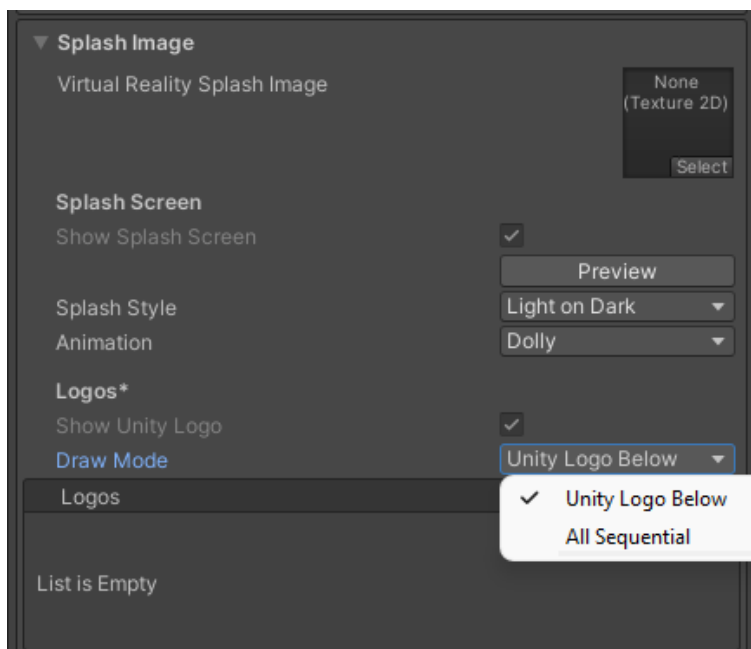
Aqui podem ser configurados detalhes como o **Company Name** (a equipe/empresa/organização que desenvolveu o jogo), o **Product Name** (o nome da aplicação final, do arquivo executável que vamos gerar e que irá aparecer para o usuário) e a **versão** do jogo. Também pode ser adicionado um ícone em **Default Icon**, que será a imagem característica do jogo e irá aparecer no próprio arquivo executável. O **Default Cursor** permite, em caso de jogos mais customizados, substituir o ponteiro do mouse padrão do Sistema Operacional por algo estilizado.

As próximas guias definem outros detalhes da aplicação final. A guia **Resolution and Presentation** (Resolução e Apresentação) é onde podemos definir se o nosso jogo irá abrir, por padrão, em tela cheia ou em janela (onde aparecem as opções de redimensionar e minimizar).



Guia *Resolution and Presentation*

A guia seguinte é a **Splash Image**, onde podemos definir detalhes de como o logotipo da Unity vai ser apresentado quando o jogo for executado. Também podemos adicionar outros logotipos que serão exibidos ao usuário no início da execução e até como a animação irá apresentar essa imagem!

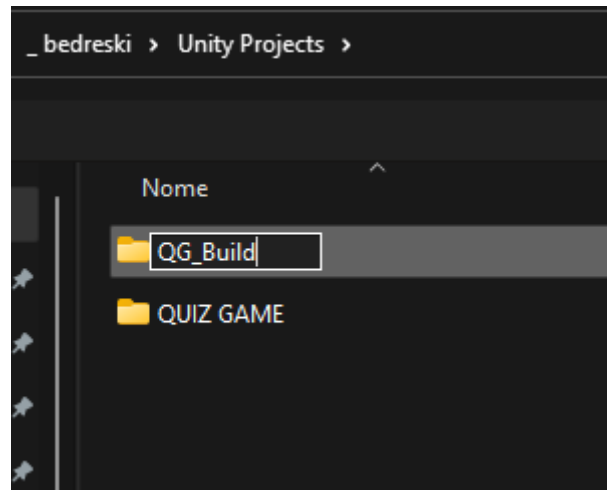


Guia *Splash Image*

Outras guias estão disponíveis abaixo e é possível explorá-las e testar cada uma das opções para ver como é o resultado na aplicação final.

→ **Passo 4:** Criar uma pasta para armazenar os arquivos de Build.

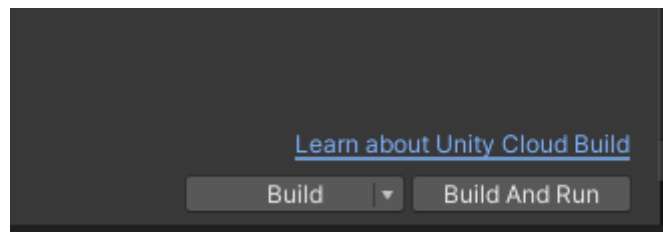
Quando geramos a Build de um projeto pela Unity dentro do Windows, não é criado apenas o executável em si, mas também outros arquivos de especificação e configuração. Ainda que só o executável nos interesse, é importante que os outros arquivos “andem junto” com o arquivo .exe e é por isso que criamos uma pasta apenas para os arquivos de Build. Você pode criá-la onde achar melhor, pode criar uma pasta para guardar todas as outras pastas de Builds dos projetos que fizer. Aqui, está sendo criada uma pasta ao lado da pasta do projeto QUIZ GAME (também dentro do diretório *Unity Projects*), chamada *QG_Build*.



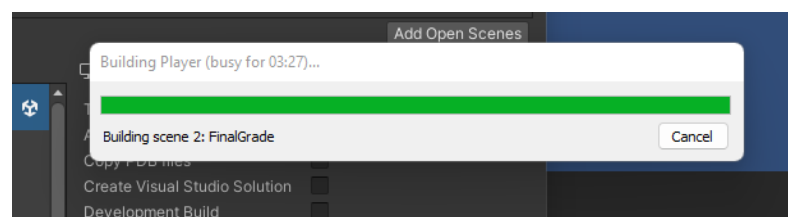
Criação da pasta para os arquivos de Build

→ **Passo 5:** Clicar na opção **Build** e ser feliz!

Com a pasta para Build selecionada, agora podemos realmente gerar a Build do nosso projeto! O processo pode ser um pouco lento e varia de máquina para máquina. Caso você queira gerar a Build e já abrir o jogo, pode fazer isso escolhendo a opção **Build and Run** ao invés da padrão.



Opção *Build*



Barra de progresso da Build

Após isso, se acessarmos a pasta *QG_Build* criada anteriormente, vamos ver os arquivos de configuração gerados pela Unity e, finalmente, o nosso arquivo executável do jogo.

Nome	Data de modificação	Tipo
MonoBleedingEdge	02/08/2022 14:20	Pasta de arquivos
QUIZ GAME_BurstDebugInformation_...	02/08/2022 14:19	Pasta de arquivos
QUIZ GAME_Data	02/08/2022 14:20	Pasta de arquivos
✓ QUIZ GAME	02/08/2022 14:20	Aplicativo
UnityCrashHandler64	02/08/2022 14:20	Aplicativo
UnityPlayer.dll	02/08/2022 14:20	Extensão de aplica...

Arquivo executável do jogo

Materiais Auxiliares

- Plataforma oficial de ensino da Unity: <https://learn.unity.com/>

Referências

- [Tutorial Unity 5 - Quiz Game - Vídeo Aulas Criação de Jogos - YouTube](#)
- Documentação da Unity 2D: <https://docs.unity3d.com/Manual/Unity2D.html>