

Unidade 2 | Seção 2

Roteiro Aula Prática



Introdução aos motores gráficos

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: Introdução aos motores gráficos

Unidade 2

Seção 2.2

OBJETIVOS

Definição dos objetivos da aula prática:

- Desenvolvimento do cenário do jogo;
- Desenvolvimento dos aspectos visuais do jogo (adição de personagem à cena, adição das moedas e obstáculos);
- Desenvolvimento da UI (interface de pontuação e fim de jogo);
- Desenvolvimento da lógica do jogo (todos os scripts necessários para realizar a lógica de controle do jogo);
- Teste do jogo.

INFRAESTRUTURA

Instalações:

Laboratório de Informática

Materiais de consumo:

Descrição	Quantid. de materiais por procedimento/atividade
-----------	--

Não se aplica

Software:

Sim (X) Não ()

Em caso afirmativo, qual?

Pago () Não Pago (X)

Tipo de Licença: Gratuita.

Descrição do software:

Unity: motor de jogo (*game engine*) usado para criar e desenvolver jogos 2D e 3D para diversas plataformas.

Equipamento de Proteção Individual (EPI):

Não se aplica.

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade Nº 1

Atividade proposta:

O aluno deverá desenvolver um jogo 2D usando o motor de jogos Unity. O jogador irá controlar um personagem em um ambiente bidimensional, que deverá pegar moedas. A jogabilidade consiste em coletar moedas ou itens semelhantes para ganhar pontos. O jogador vence quando coleta uma quantidade 'X' de moedas, determinada pelo desenvolvedor.

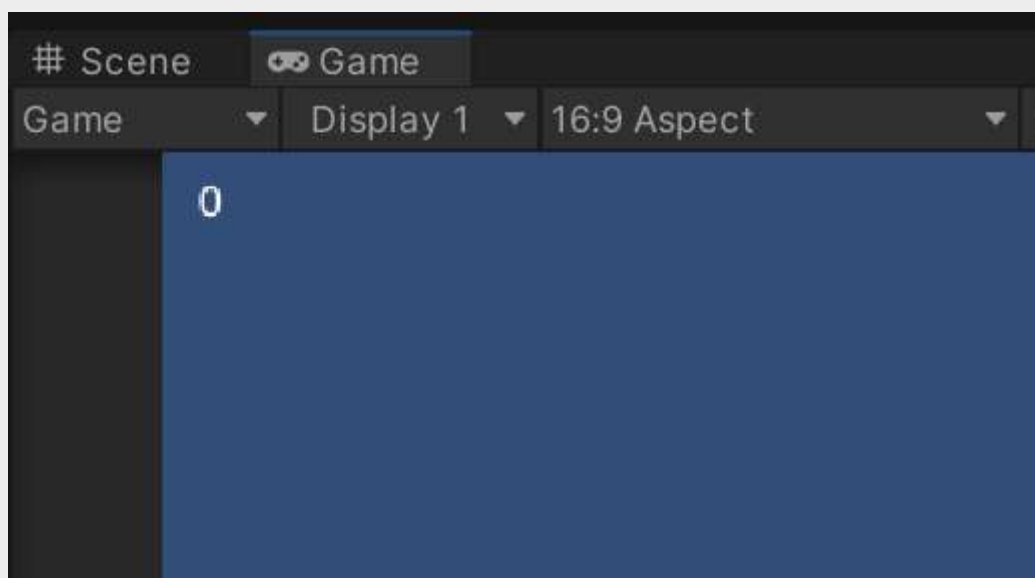
Procedimentos para a realização da atividade:

Para a realização desta atividade o aluno deverá ter o software **Unity** devidamente instalado e configurado na sua máquina. Ademais, o aluno deverá baixar e importar os seguintes conteúdos para o seu Assets da Unity:

- <https://assetstore.unity.com/packages/2d/characters/hero-knight-pixel-art-165188>
- <https://assetstore.unity.com/packages/2d/free-2d-mega-pack-177430>

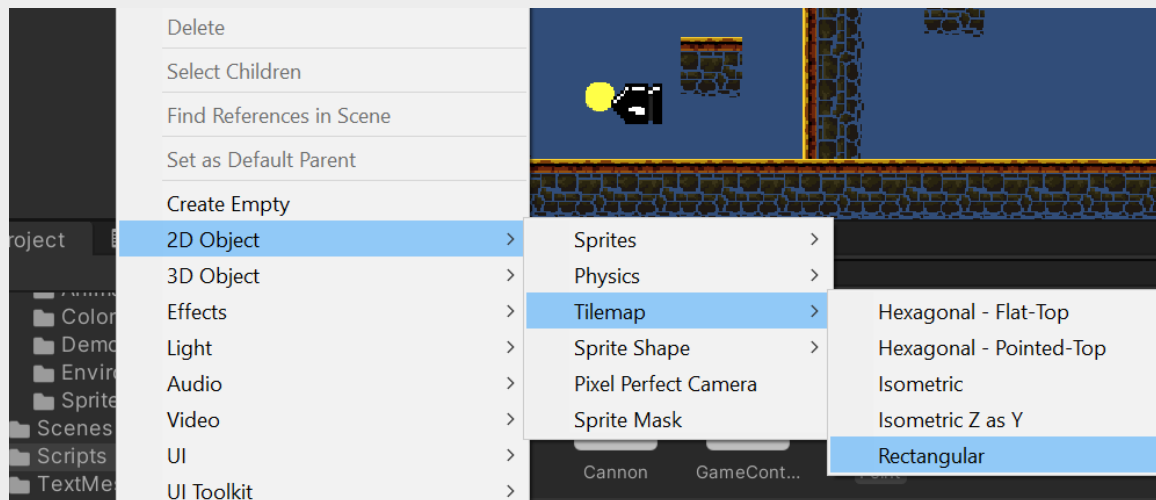
Obs.: Os *assets* acima deverão ser carregados e importados na ferramenta Unity.

1. Crie um novo projeto do tipo 2D na Unity.
2. Importe os *assets* baixados para o seu projeto.
3. Defina a proporção da tela para 16:9, conforme segue a imagem abaixo:

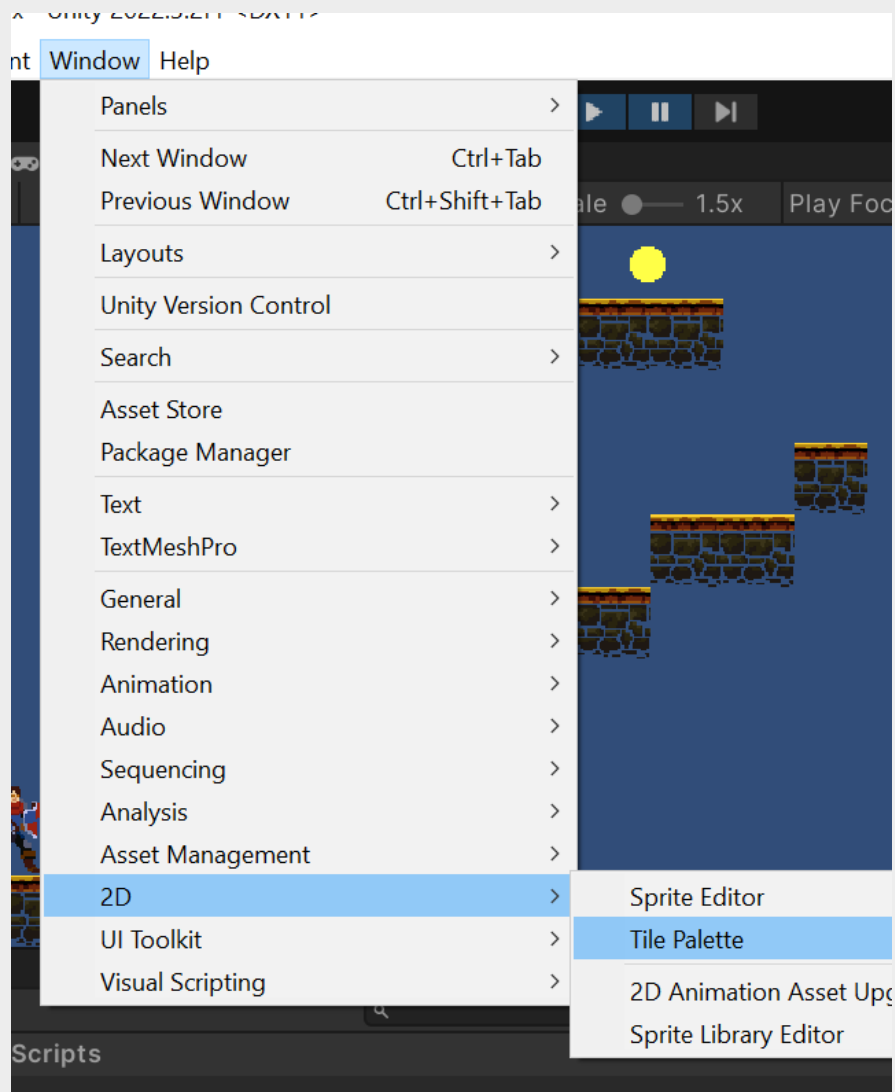


Fonte: elaborada pelo autor.

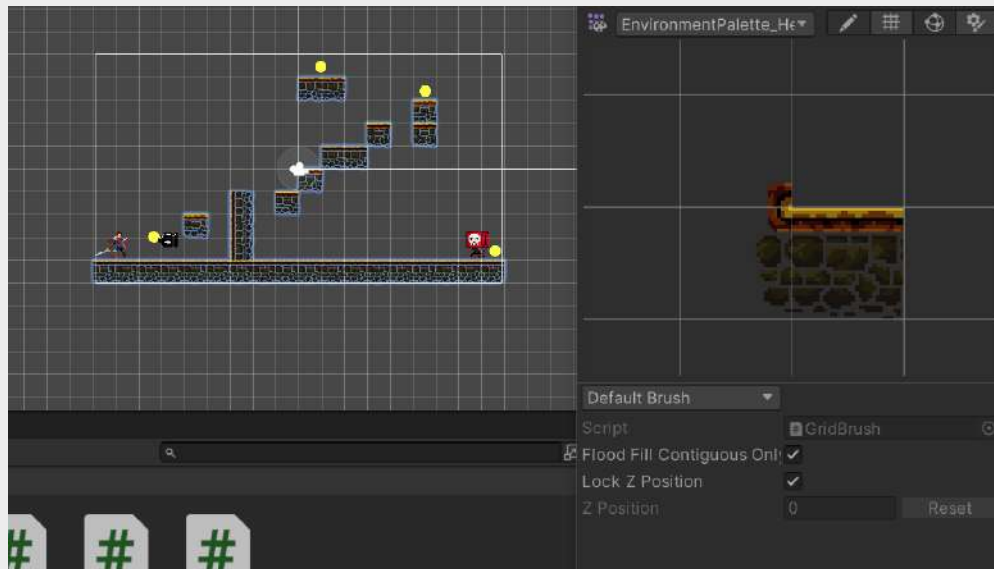
4. Selecione uma área livre na janela *Hierarchy* com o botão direito do mouse e adicione uma *Tilemap* retangular:



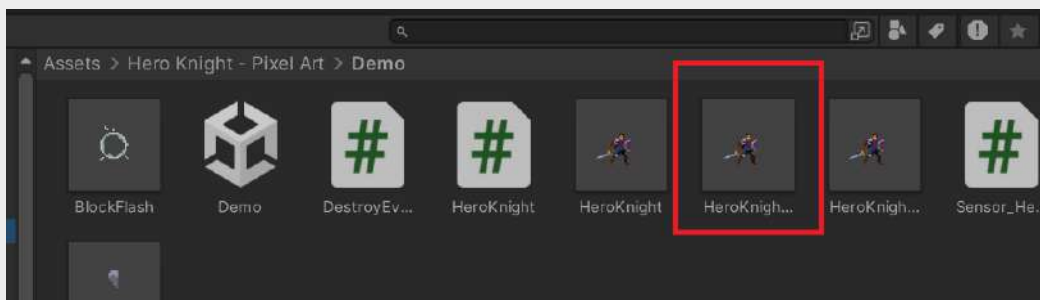
5. Selecione *Windows > 2D > Tile Palette*:



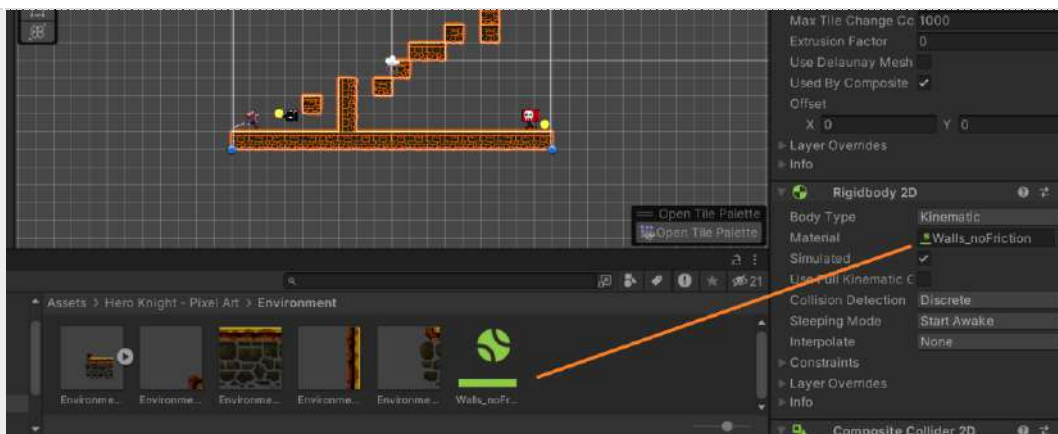
6. Os *tiles* criados no pacote serão exibidos. Utilizando a ferramenta de pincel, você pode escolher um bloco específico e pintar o cenário conforme sua preferência. Crie obstáculos que o personagem consiga passar. O tile apresentado na Figura pertence ao conjunto importado no início da atividade.



- Para usar esta ferramenta, selecione a opção “Pincel” e selecione os quadrados que deseja criar o terreno ou a parede. Você poderá “rotacionar” os *tiles* formando paredes.
7. Adicione o personagem à sua cena. Para isto arraste HeroKnight para dentro da cena:

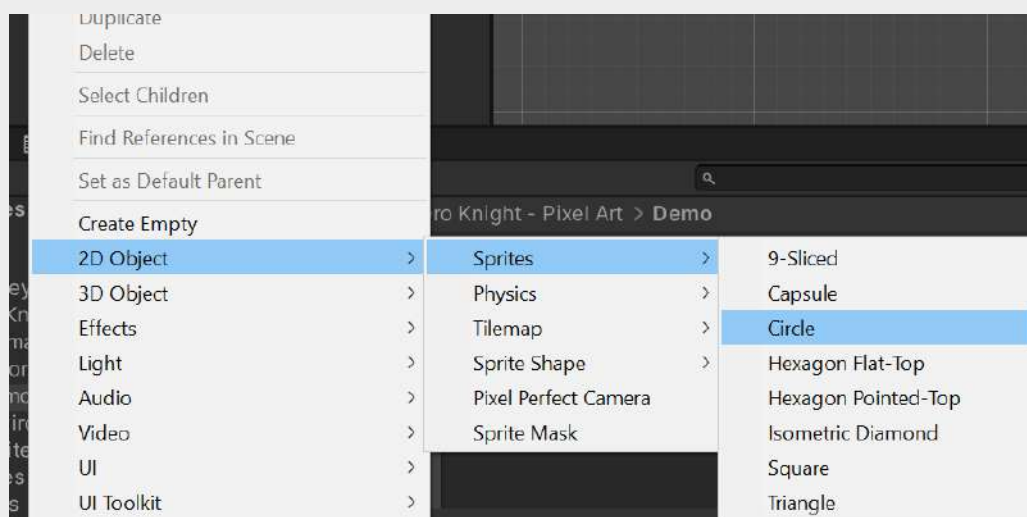


8. Adicione um *Tilemap Collider 2D* ao cenário criado. Para isto clique em *Add Component*.
- Marque a opção *Used By Composite* dentro do *Tilemap Collider 2D*.
9. Adicione o componente *Composite Collider 2D* ao cenário.
- Altere o *Body Type* do *Rigidbody 2D* criado para *Kinematic*.
 - Adicione o material *Walls_noFriction* ao *Composite Collider 2D*.



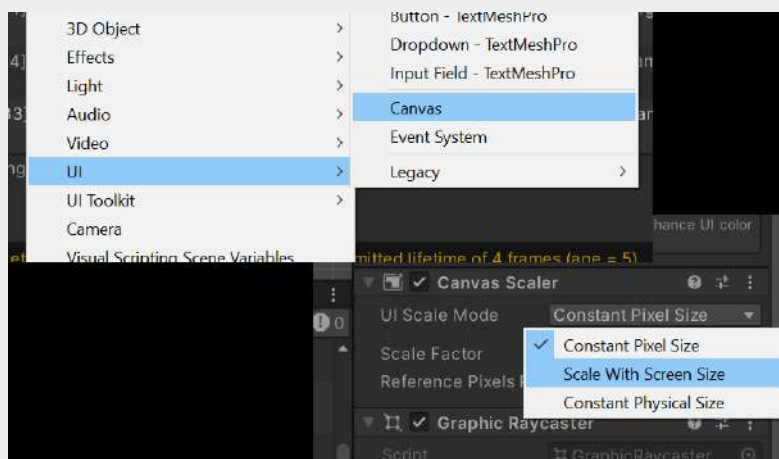
10. Crie a pontuação. Para isto crie uma *Sprite* de círculo e adicione o componente *Circle Collider 2D*, ajuste seu tamanho para 0.5, mude a cor, defina a estética como preferir.

11. Crie uma pasta dentro de *Assets* chamada *Scripts*. Dentro dela iremos salvar nossos *scripts*.

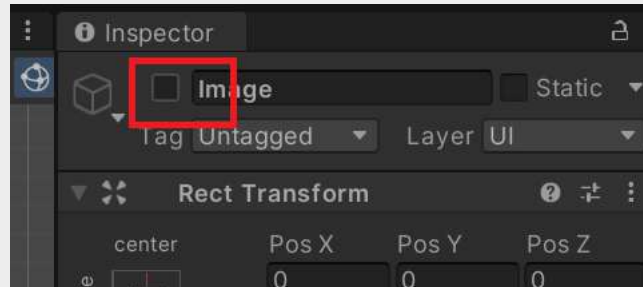


12. Crie a interface que exiba os pontos coletados (moedas coletadas), criando um *Canvas* com um *Text* dentro.

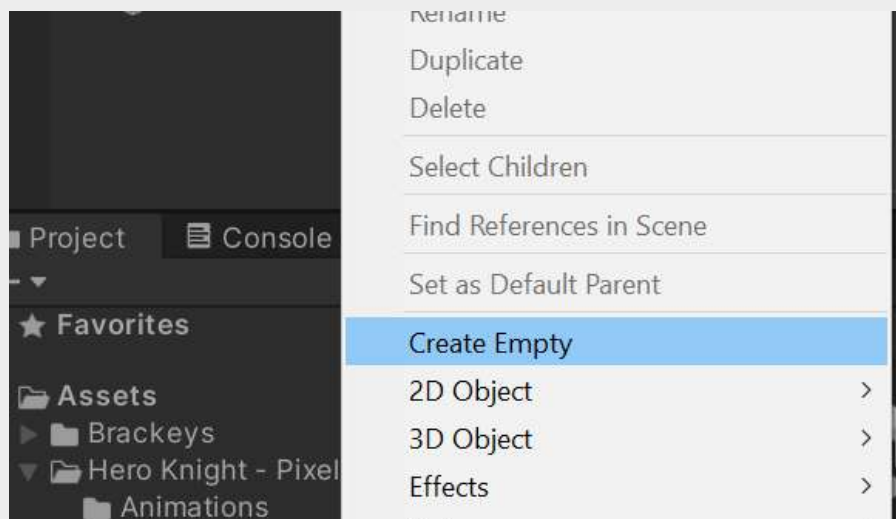
- Altere o *design* como preferir.
- Altere a escala *UI Scale Mode* para *Scale With Screen Size* e altere o *Match* para 1.



- Crie uma imagem com um pouco de opacidade. Dentro dela coloque um texto: “Fim de Jogo”. Desmarque a opção que deixa esta imagem visível para que ela apareça somente quando decidirmos que deverá aparecer (no caso quando o jogador coletar uma quantidade “X” de moedas).



13. Crie um objeto vazio e nomeie como *GameController*. Nele serão armazenadas as informações de logica de controle do jogo.

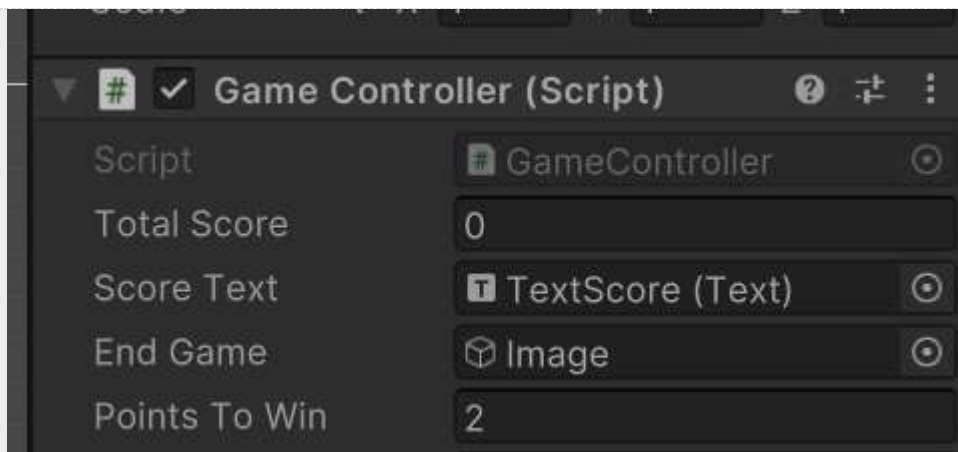


14. Crie um script chamado “*GameController*”. Dentro dele iremos criar a seguinte lógica:

- Crie dois atributos públicos: *totalScore*, *scoreText*, *endGame* e *pointsToWin*;
- Crie uma função pública chamada “*UpdateScoreText*” para atualizar o texto da UI.

```
scoreText.text = totalScore.ToString();
```

- Crie a função pública que irá mostrar a mensagem de fim de jogo caso o *totalScore* for igual ao *pointsToWin*.
- Ao finalizar o *script*, o Unity irá atualizar e você poderá colocar no *Inspect* do *GameController* os valores para cada atributo público criado. Adicione a UI de Fim de jogo (imagem criada) ao atributo *endGame* e adicione quantos pontos deseja para considerar que o jogador finalizou a fase.



15. Renomeie o círculo que foi criado anteriormente como *Point* e ative a opção *isTrigger* em seu "colisor". Isso possibilitará que o jogador passe pelo objeto, que perderá sua física.

16. Crie um script chamado "*Point*" e adicione a seguinte lógica a ele:

- Crie uma variável pública que irá definir quantos pontos serão atribuídos ao *totalScore* quando o objeto for coletado chamada "*Score*".
- Crie a implementação para a função que é chamada quando algo atravessa o círculo:

```
// Função que é chamada quando algo atravessa o objeto  
void OnTriggerEnter2D(Collider2D collider)  
{
```

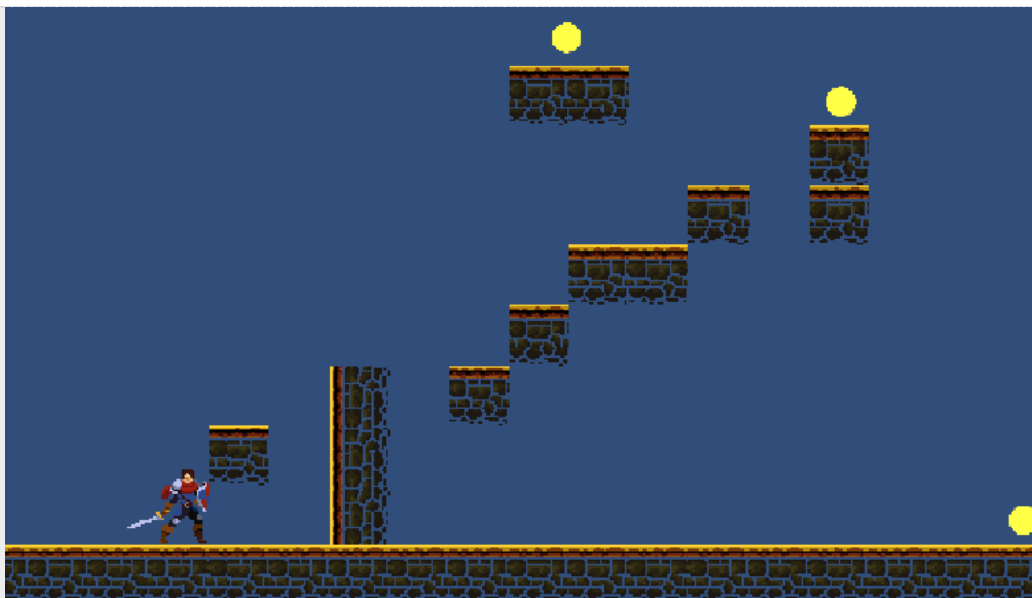
- Implemente uma lógica dentro desta função que verifique se o objeto que esta colidindo com o círculo é o jogador. Em seguida incremente o *totalScore* com os pontos "*Score*" e destrua o objeto círculo.
- Adicione o valor do *Score* como 1 no objeto *Point*.

17. Incorpore o script criado ao objeto vazio *GameController* e atribua a ele o *textScore*.

18. Adicione o script "*Point*" dentro do objeto círculo (para isto arraste o *script* ao seu *Inspector*).

- Adicione o valor do *Score* no objeto *Point* (quantidade de moedas que o jogador deveria coletar para finalizar o jogo).

19. Teste o Jogo.



Checklist:

- Desenvolvimento do cenário do jogo;
- Desenvolvimento dos aspectos visuais do jogo (adição de personagem à cena, adição das moedas e obstáculos);
- Desenvolvimento da UI (interface de pontuação e fim de jogo);
- Desenvolvimento da lógica do jogo: *script Point* e *script GameController*;
- Testar o jogo.
- Adicionar níveis de dificuldade (outros *levels* ou inimigos, como desejar) **opcional.

RESULTADOS

Resultados da aula prática:

É esperado que o aluno ao término da prática o aluno aprenda a utilizar os principais mecanismos do motor de jogo escolhido, desde conceitos de criação de cenário, manipulação de personagem, lógica baseada em frames e UI, até a utilização de conceitos de física como a gravidade, colisão, detecção de objetos e movimentos. Ademais, o aluno deverá entregar uma pasta com todos os scripts em C# utilizados para a criação do jogo e a exportação do executável jogo para Windows e Mac.