

Jogo Patrulha do Espaço

1. Criar três cenas: menu, main e game over
2. No menu, criar o design inicial do jogo e um botão que abrirá a cena “main”
3. Criar script compBtnJogo
4. Criar função para AbrirMain

```
public void AbrirMain()
{
    SceneManager.LoadScene("main");
}
```

5. Inserir nave espacial no Main
6. Adicionar BoxCollider2D – objeto adicionado para tratar a colisão dele com outros objetos no jogo
7. Pode-se criar um objeto vazio OpcoesNave ou associar diretamente com o objeto Nave
8. Criar Script compNave – arraste para o objeto nave ou OpcoesNave
9. Objetos Pré-Fabricados – Criar pasta, arrastar objeto para dentro da pasta
10. Criar função para mudar a posição da Nave

```
public void MudarPosNave ( )
{
    //Verificar no Unity a propriedade máxima e mínima na Cena -
    Transform – Position X, Y, Z

    Float x = Random.Range (-7f,7f); //intervalo aleatório

    Float y = Random.Range (-3f, 3f); //intervalo aleatório

    //O Transform do Unity recebe um vetor, um tipo de objeto que
    representa 3 valores (Vector3)

    Vector3 vetor = new Vector3(x,y,0);

    //Acessar a propriedade Transform do Objeto Nave Espacial

    gameObject.transform.position = vetor;
}
```

11. Acessar a função MudarPosNave

```
void OnMouseDown( )  
{  
    MudarPosNave( );  
    OpcoesNave.quantidade++;  
}
```

12. Criar o contador de Naves

Toda programação até o momento foi para os itens do Objeto, mas para fazer o contador é preciso criar um item para a classe lá no início

//static diz que a variável é da classe – se não especificar, a variável é do objeto

```
Static public int quantidade;
```

13. Programar Mecânica do Tempo – 20 segundos por exemplo

14. Criar novo componente vazio

15. Chamar de GameController

16. Criar um script para ele - GameController

17. Associar esse script ao objeto vazio criado

18. Inserir uma “dica de ferramenta” através do comando

```
[SerializeField, Tooltip (“Duração do Jogo”)]
```

- A propriedade SerializeField permite visualizar e editar variáveis privadas no Inspector da Unity.
- Tooltip significa “Dica de Ferramenta”

19. Criar uma variável para o tempo –

```
[SerializeField, Tooltip(“Duração do Jogo”)]  
private float timer;
```

20. Criar função IrParaGameOver

```
void IrParaGameOver( )  
{  
    SceneManagement.SceneManager.LoadScene(2);  
}
```

21. Em Start, chamar a função que encerra o jogo

```
void Start ( )
{
    //Zera a quantidade de naves interceptadas no jogo anterior
    OpcoesNave.quantidade = 0;
    //Comando que chama uma função depois do tempo
    estabelecido
    Invoke("IrParaGameOver", timer);
}
```

22. Para exibir o Timer na Cena do Jogo (main), adicionar um TMP: Timer: 10;

23. Adicionar a biblioteca using TMPro no script GameController;

24. Adicionar [SerializeField, Tooltip("Texto do Timer do Jogo")]

25. Criar variável TextTimer para retornar timer diretamente no TextMeshPro

```
private TextMeshProUGUI TextTimer;
```

26. Dentro de Start, passar o valor do Timer para a variável TextTimer

```
TextTimer.text = "TEMPO: " + timer.ToString( );
```

27. Criar função para decrementar o tempo

```
private void DecTimer ( )
{
    timer -= 1; // recebe o valor atual menos 1
    TextTimer.text = "TEMPO: " + timer.ToString( ); // Esse comando
    atualiza o texto gerado na tela anteriormente
}
```

28. Para isso, em Start é preciso utilizar o comando InvokeRepeating ("método chamado", repete depois de x tempo, por x tempo)

29. Comando fica assim: InvokeRepeating("DecTimer", 1f, 1f);

30. No objeto vazio gamecontroller, associar Text Timer com Text(TMP)

31. Para programar a mensagem final, onde a tela permanecerá por 5 segundos mostrando o resultado e depois retornará ao Menu Principal, criar um script GameOver

32. Criar objeto vazio GameOverController e associar ao script GameOver

33. No script, criar

```
[SerializeField, Tooltip ("Contagem Final")]  
private TextMeshProUGUI contagemfinal;
```

34. Criar função para voltar ao Menu Principal

```
void IrParaMenu ( )  
{  
    SceneManager.LoadScene (0);  
}
```

35. Em Start, chamar método IrParaMenu

```
void Start ( )  
{  
    Invoke("IrParaMenu",5);  
}
```

36. Acrescentar biblioteca TMPro;

37. Incrementar o texto com a quantidade

```
TextTimer.Text = "Você interceptou " +  
OpcoesNave.quantidade.ToString( ) + " naves espaciais!";
```

38. Associar objeto vazio GameOverController com Inspector TextMesh da contagem final

39. Criar botão Sair e o script compBtnSair

40. Programar a função SairdoJogo()

Lista de Scripts Completos

Script compBotaoJogar.cs

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

public class compBtnJogar : MonoBehaviour
{
    public void AbrirMain()
    {
        SceneManager.LoadScene("main");
    }
}
```

Script comBotaoSair.cs

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

public class compBtnSair : MonoBehaviour
{
    public void SairdoJogo()
    {
        Debug.Log("Jogo Encerrado");
        Application.Quit();
    }
}
```

Script compNave.cs

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

public class OpcoesNave : MonoBehaviour
{
    static public int quantidade;

    void OnMouseDown()
    {
        MudarNave();
        OpcoesNave.quantidade++;
    }

    public void MudarNave()
    {
        float x = Random.Range(-7, 7);
        float y = Random.Range(-3, 3);
        Vector3 vetor = new Vector3(x, y, 0);
        gameObject.transform.position = vetor;
    }
}
```

Script GameController.cs

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

```

public class gamecontroller : MonoBehaviour
{
    [SerializeField, Tooltip ("Duração do Jogo")]
    private float timer;
    [SerializeField, Tooltip ("Timer do Jogo")]
    private TextMeshProUGUI TextTimer;
    // Start is called before the first frame update
    void Start()
    {
        OpcoesNave.quantidade = 0;
        Invoke("IrParaGameOver", timer);
        TextTimer.text="TEMPO: " + timer.ToString();
        InvokeRepeating("DecTimer",1f, 1f);
    }
    public void IrParaGameOver ()
    {
        SceneManager.LoadScene("gameover");
    }
    private void DecTimer()
    {
        timer-=1;
        TextTimer.text="TEMPO: " + timer.ToString();
    }
}

```

Script GameOver.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;

```

```
public class gameover : MonoBehaviour
{
    [SerializeField, Tooltip ("Contagem Final")]
    private TextMeshProUGUI textContagemFinal;
    // Start is called before the first frame update
    void Start()
    {
        textContagemFinal.text = "Você interceptou " +
OpcoesNave.quantidade.ToString() + " naves espaciais!";
        //Invoke("IrParaMenu",5);
    }

    void IrParaMenu()
    {
        UnityEngine.SceneManagement.SceneManager.LoadScene("menu");
    }
}
```