

**FUNDAÇÃO VALEPARAIBANA DE ENSINO**  
**COLÉGIOS UNIVAP – UNIDADE CENTRO**

**CURSO TÉCNICO EM INFORMÁTICA**

**NOME DO AUTOR**

**LISTA DE EXERCÍCIOS 2º BIMESTRE**

**DESENVOLVIMENTO DE JOGOS**

Lista apresentada ao Curso Técnico de informática  
como composição de nota.  
Prof. Me. Hélio Lourenço Esperidião Ferreira

**SÃO JOSÉ DOS CAMPOS**  
**2023**

1. Quais as formas de colisão que existem e podem ser representadas no unity?

Existe a colisão de objetos tocados, podendo ser representado no unity com o: OnCollisionEnter2D.  
Existe a colisão de dois objetos continuam sendo tocados, representado com o: OnCollisionStay2D.  
E existe a colisão de quando um objeto para de tocar no outro, representado com o: OnCollisionExit2D.

2. Crie um pequeno jogo contendo partes do chão coberta com gelo. Aumente a velocidade do jogador quando personagem passar pelo gelo. Ao sair do gelo retorne a velocidade normal.

```
using System.Collections;
```

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

```
using UnityEngine;
```

```
public class ex2 : MonoBehaviour
```

```
{
```

```
float Vx; //velocidade em x
```

```
float Vy; //velocidade em y
```

```
float VelocidadeHorizontal;
```

```
float VelocidadeVertical;
```

```
float VelocidadePular;
```

```
float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
```

```
float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
```

```
Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
```

```
PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.
```

```
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
```

```
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
```

```
void Start()
```

```
{
```

```
Vx = 0; //inicia a velocidade em x
```

```

Vy = 0; //inicia a velocidade em y
DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")
DirecaoHorizontal = 0; // inicia a direção horizontal(andar)
VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)
VelocidadePular = 7; //inicia a velocidade de pular

CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento
que possui o script(corpo rígido do personagem)

Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
possui o script(colisor do personagem)

CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
= 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

```

```

//"void Update ()" Executado uma vez por frame

```

```

//"void Update ()" Executado uma vez por frame.

```

```

void Update()

```

```

{

```

```

MovimentoHorizontal();

```

```

Pular();

```

```

//MovimentoVertical ();

```

```

}

```

```

void Pular()

```

```

{

```

```

//verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false

```

```

bool apertou = Input.GetButtonDown("Jump");

//verdadeiro(true) se o personagem estiver tocando outro collider
bool tocou = Colisor.IsTouchingLayers();

if (apertou == true && tocou == true)
{
    // esse bloco só é executado se for apertado o botão de Jump
    // e se o personagem estiver tocando em alguma coisa com collider
    Vx = CorpoRigido.velocity.x;
    Vy = VelocidadePular;
    Vector2 pulo = new Vector2(Vx, Vy);
    CorpoRigido.velocity = pulo;
}
}

void MovimentoHorizontal()
{
    //DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.
    Quando parado Recebe 0.

    DirecaoHorizontal = Input.GetAxis("Horizontal");

    // Gera uma nova velocidade em x
    Vx = VelocidadeHorizontal * DirecaoHorizontal;

    //Recupera a velocidade em y que o personagem já possui
    Vy = CorpoRigido.velocity.y;

    //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
    Vector2 andar = new Vector2(Vx, Vy);

    // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
    CorpoRigido.velocity = andar;
}

```

```

void MovimentoVertical()
{
    //DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando
    parado Recebe 0.

    DirecaoVertical = Input.GetAxis("Vertical");

    Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu
    corpo rígido

    // Gera uma nova velocidade em y

    Vy = VelocidadeVertical * DirecaoVertical;

    Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da
    velocidade em x e da velocidade em y (vx e vy)

    CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade
    do corpo rígido do personagem
}

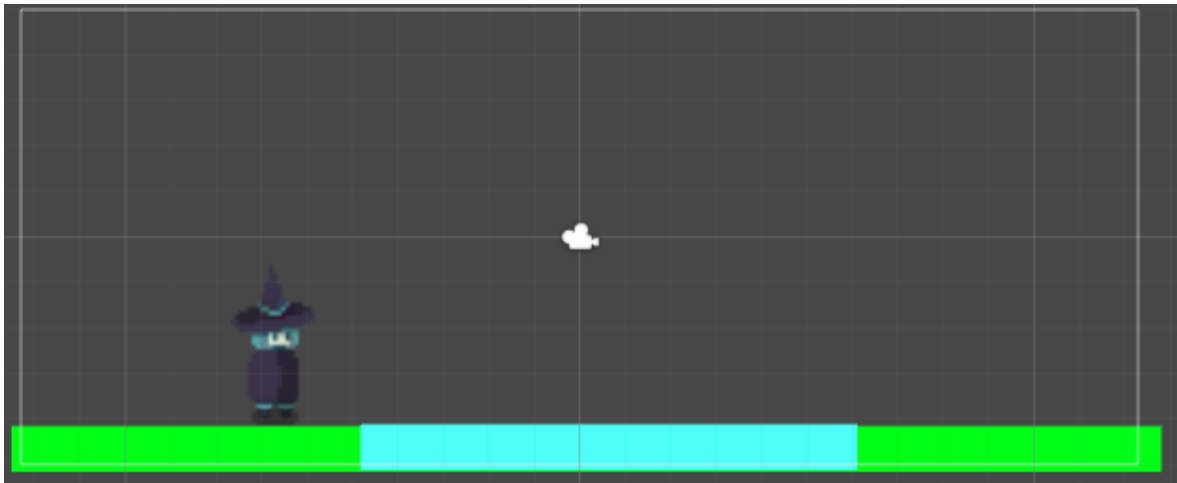
```

```

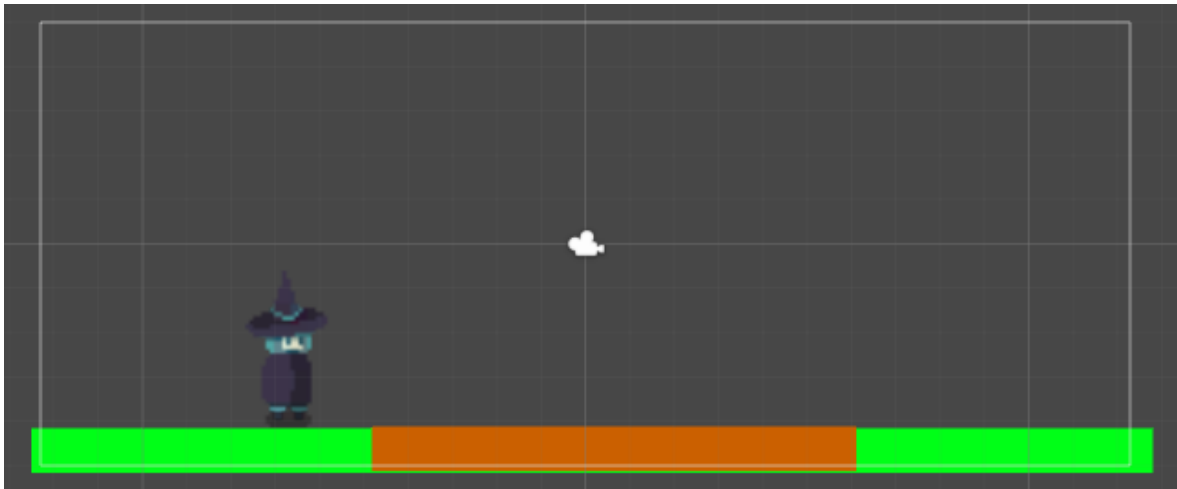
void OnCollisionStay2D(Collision2D tocandoNoObjeto)
{
    string tagTocando = tocandoNoObjeto.gameObject.tag;

    if (tagTocando == "ice")
    {
        VelocidadeHorizontal = VelocidadeHorizontal + 2f;
    }
}

```



3. Crie um pequeno jogo contendo lama no chão. Diminua a velocidade do jogador quando o mesmo permanece na lama.



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

public class ex3 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
```

```

float VelocidadePular;

float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
void Start()
{
Vx = 0; //inicia a velocidade em x
Vy = 0; //inicia a velocidade em y
DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")
DirecaoHorizontal = 0; // inicia a direção horizontal(andar)
VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)
VelocidadePular = 7; //inicia a velocidade de pular

CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rigido inserido no elemento
que possui o script(corpo rígido do personagem)

Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
possui o script(colisor do personagem)

CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
= 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame

```

```

//"void Update ()" Executado uma vez por frame.
void Update()
{
    MovimentoHorizontal();
    Pular();
    //MovimentoVertical ();

}

void Pular()
{
    //verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
    bool apertou = Input.GetButtonDown("Jump");

    //verdadeiro(true) se o personagem estiver tocando outro collider
    bool tocou = Colisor.IsTouchingLayers();

    if (apertou == true && tocou == true)
    {
        // esse bloco só é executado se for apertado o botão de Jump
        // e se o personagem estiver tocando em alguma coisa com collider
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular;
        Vector2 pulo = new Vector2(Vx, Vy);
        CorpoRigido.velocity = pulo;
    }
}

void MovimentoHorizontal()
{

```



//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda. Quando parado Recebe 0.

```
DirecaoHorizontal = Input.GetAxis("Horizontal");
```

```
// Gera uma nova velocidade em x
```

```
Vx = VelocidadeHorizontal * DirecaoHorizontal;
```

```
//Recupera a velocidade em y que o personagem já possui
```

```
Vy = CorpoRigido.velocity.y;
```

```
//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
Vector2 andar = new Vector2(Vx, Vy);
```

```
// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
CorpoRigido.velocity = andar;
```

```
}
```

```
void MovimentoVertical()
```

```
{
```

```
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando parado Recebe 0.
```

```
DirecaoVertical = Input.GetAxis("Vertical");
```

```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
}
```

```
void OnCollisionStay2D(Collision2D tocandoNoObjeto)
```

```
{
```

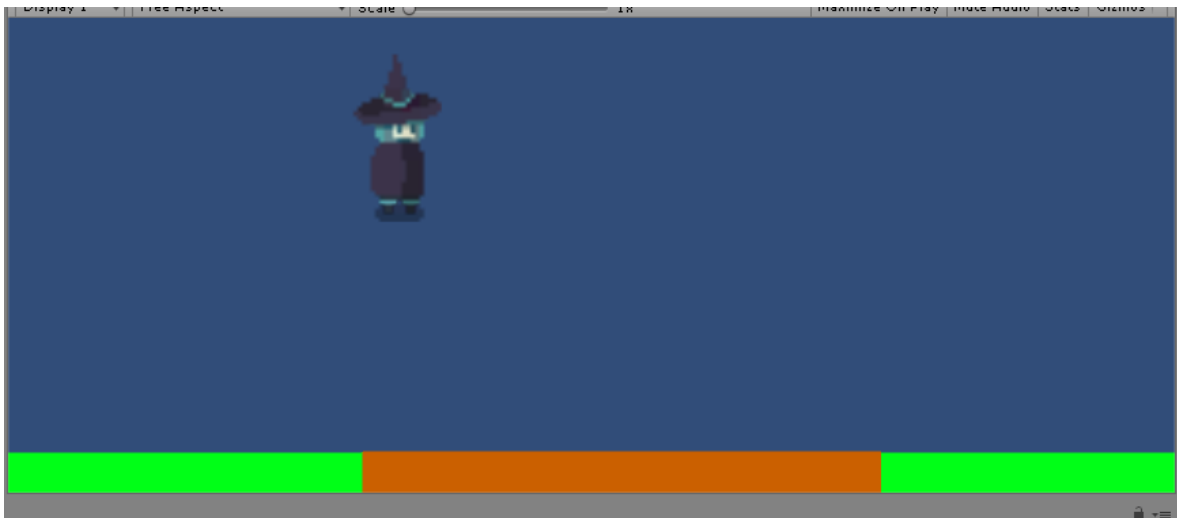
```
string tagTocando = tocandoNoObjeto.gameObject.tag;
```

```

if (tagTocando == "lama")
{
    VelocidadeHorizontal = 2;
}
}
}

```

#### 4. Faça um pulo triplo.



```

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

public class ex4 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;
}

```

```

float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

float ContadorPulos;

//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
void Start()
{
    ContadorPulos = 0;
    Vx = 0; //inicia a velocidade em x
    Vy = 0; //inicia a velocidade em y
    DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")
    DirecaoHorizontal = 0; // inicia a direção horizontal(andar)
    VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
    VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)
    VelocidadePular = 7; //inicia a velocidade de pular

    CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento
    que possui o script(corpo rígido do personagem)

    Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
    possui o script(colisor do personagem)

    CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

    //inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
    = 1 deve ser igual 0

    CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame

```

```

//"void Update ()" Executado uma vez por frame.
void Update()
{
    MovimentoHorizontal();
    PuloTriplo();
    //MovimentoVertical ();

}

void MovimentoHorizontal()
{
    //DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.
    Quando parado Recebe 0.

    DirecaoHorizontal = Input.GetAxis("Horizontal");

    // Gera uma nova velocidade em x
    Vx = VelocidadeHorizontal * DirecaoHorizontal;

    //Recupera a velocidade em y que o personagem já possui
    Vy = CorpoRigido.velocity.y;

    //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
    Vector2 andar = new Vector2(Vx, Vy);

    // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
    CorpoRigido.velocity = andar;
}

void MovimentoVertical()
{
    //DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando
    parado Recebe 0.

    DirecaoVertical = Input.GetAxis("Vertical");

```

```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
}
```

```
// pulo triplo
```

```
void OnCollisionEnter2D(Collision2D objetoTocado)
```

```
{
```

```
string tagTocada = objetoTocado.gameObject.tag;
```

```
if (tagTocada == "chao")
```

```
{
```

```
ContadorPulos = 0;
```

```
}
```

```
}
```

```
void PuloTriplo()
```

```
{
```

```
//verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
```

```
bool apertou = Input.GetButtonDown("Jump");
```

```
// esse bloco só é executado se for apertado o botão de Jump
```

```
// e se o personagem ContadorPulos < 2
```

```
if (apertou == true && ContadorPulos < 3)
```

```
{
```

```
ContadorPulos = ContadorPulos + 1;
```

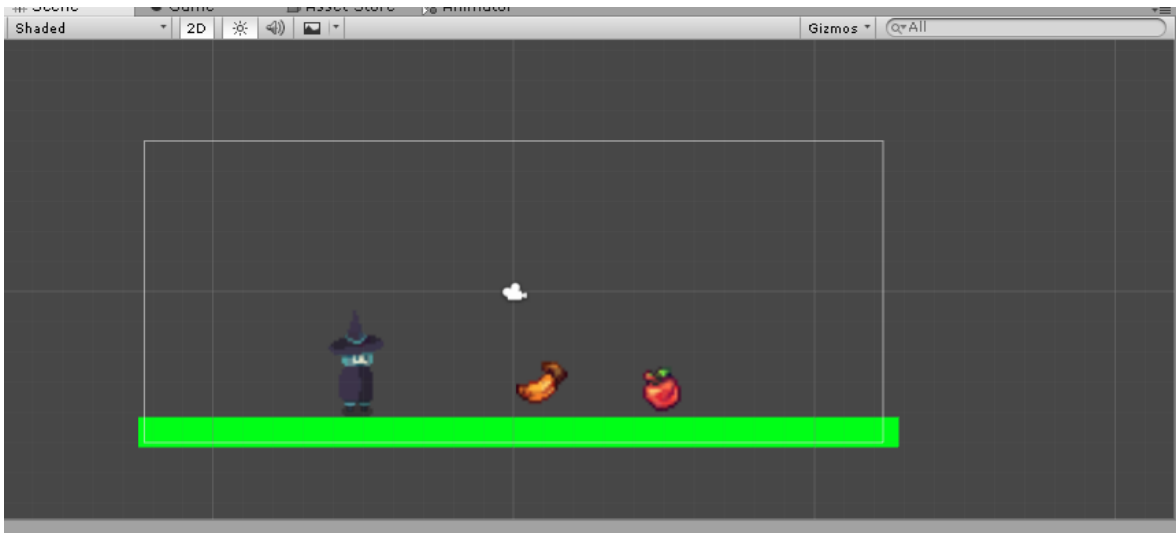
```
Vx = CorpoRigido.velocity.x;
```

```

Vy = VelocidadePular;
Vector2 pulo = new Vector2(Vx, Vy);
CorpoRigido.velocity = pulo;
}
}
}

```

5. Faça que o personagem ao tocar em uma banana permita pulo duplo e ao tocar em uma maçã permita pulo triplo.



```

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

public class ex5 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;
    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado

```

```

float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado

Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.

PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

float ContadorPulos;

float Pulo;


//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
void Start()
{
    ContadorPulos = 0;

    Vx = 0; //inicia a velocidade em x
    Vy = 0; //inicia a velocidade em y

    DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")
    DirecaoHorizontal = 0; // inicia a direção horizontal(andar)
    VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
    VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)
    VelocidadePular = 7; //inicia a velocidade de pular

    CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento
    que possui o script(corpo rígido do personagem)

    Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
    possui o script(colisor do personagem)

    CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

    //inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
    = 1 deve ser igual 0

    CorpoRigido.gravityScale = 1;

}


//"void Update ()" Executado uma vez por frame

```

```

//"void Update ()" Executado uma vez por frame.
void Update()
{
    MovimentoHorizontal();
    Pular();
    //MovimentoVertical ();

}

void MovimentoHorizontal()
{
    //DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.
    Quando parado Recebe 0.

    DirecaoHorizontal = Input.GetAxis("Horizontal");

    // Gera uma nova velocidade em x
    Vx = VelocidadeHorizontal * DirecaoHorizontal;

    //Recupera a velocidade em y que o personagem já possui
    Vy = CorpoRigido.velocity.y;

    //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
    Vector2 andar = new Vector2(Vx, Vy);

    // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
    CorpoRigido.velocity = andar;
}

void MovimentoVertical()
{
    //DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando
    parado Recebe 0.

    DirecaoVertical = Input.GetAxis("Vertical");

```



```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
}
```

```
// pulo triplo
```

```
void OnCollisionEnter2D(Collision2D objetoTocado)
```

```
{
```

```
string tagTocada = objetoTocado.gameObject.tag;
```

```
if (tagTocada == "chao")
```

```
{
```

```
ContadorPulos = 0;
```

```
}
```

```
if (tagTocada == "apple") {
```

```
Destroy(objetoTocado.gameObject);
```

```
Pulo = 3;
```

```
}
```

```
if (tagTocada == "Banana")
```

```
{
```

```
Destroy(objetoTocado.gameObject);
```

```
Pulo = 2;
```

```
}
```

```
}
```

```
//
```

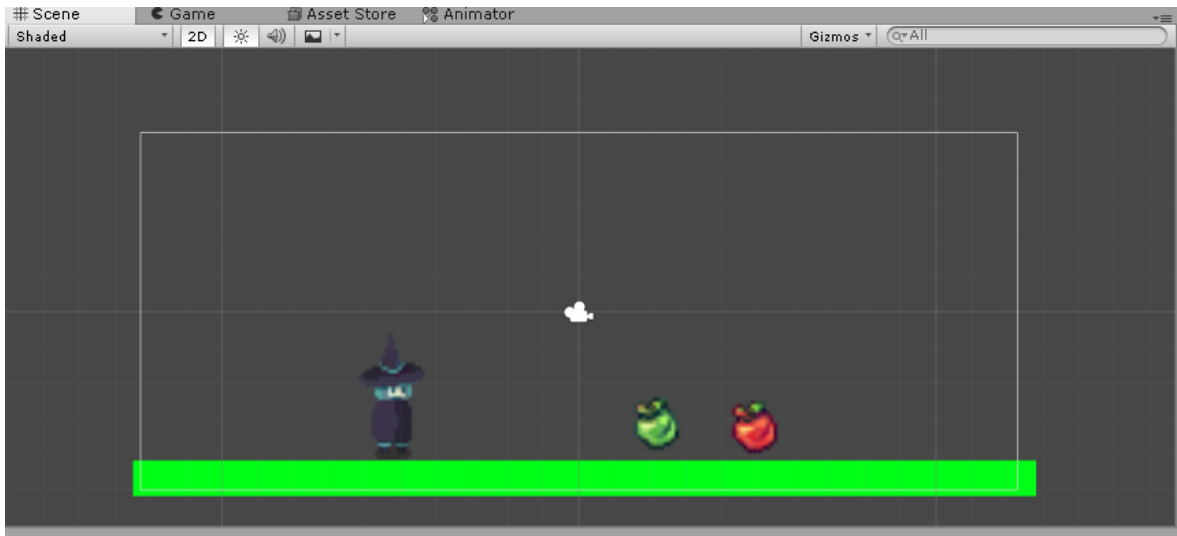
```

void Pular()
{
    bool apertou = Input.GetButtonDown("Jump");
    if (Pulo == 2)
    {
        if (apertou == true && ContadorPulos < 2)
        {
            ContadorPulos = ContadorPulos + 1;
            Vx = CorpoRigido.velocity.x;
            Vy = VelocidadePular;
            Vector2 pulo = new Vector2(Vx, Vy);
            CorpoRigido.velocity = pulo;
        }
    }

    if (Pulo == 3)
    {
        if (apertou == true && ContadorPulos < 3)
        {
            ContadorPulos = ContadorPulos + 1;
            Vx = CorpoRigido.velocity.x;
            Vy = VelocidadePular;
            Vector2 pulo = new Vector2(Vx, Vy);
            CorpoRigido.velocity = pulo;
        }
    }
}

```

6. Faça com que o jogador possa coletar maçãs verdes e vermelhas. As vermelhas aumentam a velocidade de movimento. As verdes diminuem.



```

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

public class ex6 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;

    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
    float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
    Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
    PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    void Start()
    {
        Vx = 0; //inicia a velocidade em x
    }

```

```

Vy = 0; //inicia a velocidade em y

DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")

DirecaoHorizontal = 0; // inicia a direção horizontal(andar)

VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)

VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)

VelocidadePular = 7; //inicia a velocidade de pular

CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento
que possui o script(corpo rígido do personagem)

Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
possui o script(colisor do personagem)

CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
= 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame

//"void Update ()" Executado uma vez por frame.

void Update()
{
    MovimentoHorizontal();

    //MovimentoVertical ();

}

void MovimentoHorizontal()
{

```

//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda. Quando parado Recebe 0.

DirecaoHorizontal = Input.GetAxis("Horizontal");

// Gera uma nova velocidade em x

Vx = VelocidadeHorizontal \* DirecaoHorizontal;

//Recupera a velocidade em y que o personagem já possui

Vy = CorpoRigido.velocity.y;

//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)

Vector2 andar = new Vector2(Vx, Vy);

// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem

CorpoRigido.velocity = andar;

}

/\* void MovimentoVertical()

{

//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando parado Recebe 0.

DirecaoVertical = Input.GetAxis("Vertical");

Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido

// Gera uma nova velocidade em y

Vy = VelocidadeVertical \* DirecaoVertical;

Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)

CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem

\*/

void OnCollisionEnter2D(Collision2D objetoTocado)

{

string tagTocada = objetoTocado.gameObject.tag;

```

if (tagTocada == "apple")
{
    VelocidadeHorizontal = VelocidadeHorizontal + 5;
    Destroy(objetoTocado.gameObject);
}

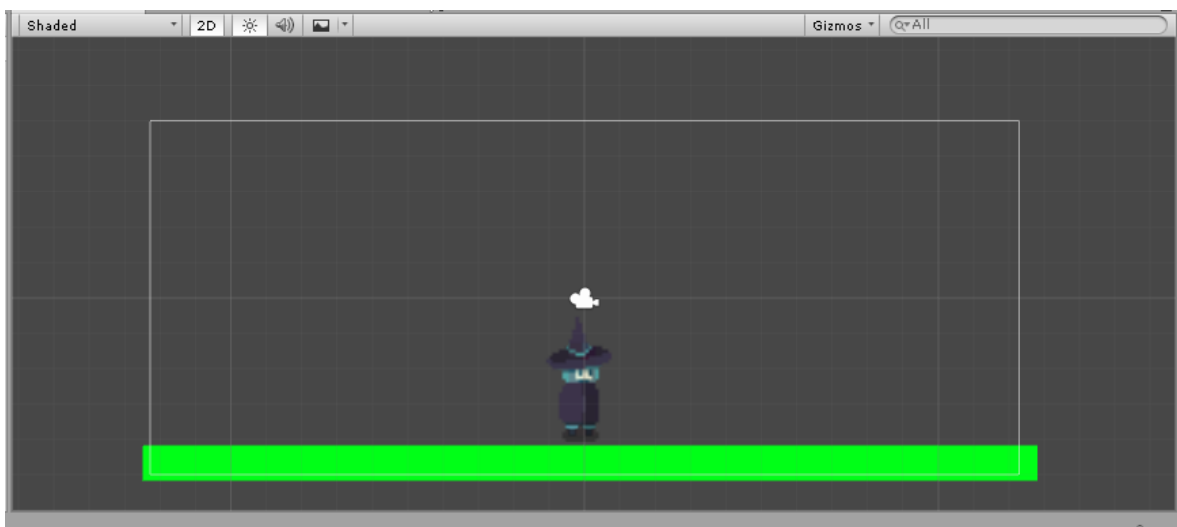
if (tagTocada == "Greenapple")
{
    VelocidadeHorizontal = VelocidadeHorizontal - 6;
    Destroy(objetoTocado.gameObject);
}

}

}

```

7. Faça um pulo triplo onde a intensidade do pulo diminui do pulo primário para o duplo e do duplo para o triplo.



```

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

public class ex7 : MonoBehaviour
{
    float Vx; //velocidade em x

```

```

float Vy; //velocidade em y

float VelocidadeHorizontal;

float VelocidadeVertical;

float VelocidadePular;

float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado

float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado

Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.

PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

float ContadorPulos;


//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
void Start()
{
    ContadorPulos = 0;

    Vx = 0; //inicia a velocidade em x

    Vy = 0; //inicia a velocidade em y

    DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")

    DirecaoHorizontal = 0; // inicia a direção horizontal(andar)

    VelocidadeHorizontal = 5; //incia a velocidade de andar na horizontal(direita e esquerda)

    VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)

    VelocidadePular = 7; //inicia a velocidade de pular

    CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rigido inserido no elemento
    que possui o script(corpo rígido do personagem)

    Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
    possui o script(colisor do personagem)

    CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

    //inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
    = 1 deve ser igual 0

    CorpoRigido.gravityScale = 1;

```

```
}
```

```
//"void Update ()" Executado uma vez por frame
```

```
//"void Update ()" Executado uma vez por frame.
```

```
void Update()
```

```
{
```

```
MovimentoHorizontal();
```

```
PuloTriplo();
```

```
//MovimentoVertical ();
```

```
}
```

```
void MovimentoHorizontal()
```

```
{
```

```
//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.  
Quando parado Recebe 0.
```

```
DirecaoHorizontal = Input.GetAxis("Horizontal");
```

```
// Gera uma nova velocidade em x
```

```
Vx = VelocidadeHorizontal * DirecaoHorizontal;
```

```
//Recupera a velocidade em y que o personagem já possui
```

```
Vy = CorpoRigido.velocity.y;
```

```
//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
Vector2 andar = new Vector2(Vx, Vy);
```

```
// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
CorpoRigido.velocity = andar;
```

```
}
```

```
void MovimentoVertical()
```



```

{
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando
parado Recebe 0.

DirecaoVertical = Input.GetAxis("Vertical");

Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu
corpo rígido

// Gera uma nova velocidade em y

Vy = VelocidadeVertical * DirecaoVertical;


Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da
velocidade em x e da velocidade em y (vx e vy)

CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade
do corpo rígido do personagem

}

```

```

// pulo triplo

void OnCollisionEnter2D(Collision2D objetoTocado)

```

```

{
string tagTocada = objetoTocado.gameObject.tag;

if (tagTocada == "chao")
{
ContadorPulos = 0;
}
}

```

```

void PuloTriplo()
{
//verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false

bool apertou = Input.GetButtonDown("Jump");

// esse bloco só é executado se for apertado o botão de Jump

// e se o personagem ContadorPulos < 2

```

```

if (apertou == true && ContadorPulos < 3)
{
    ContadorPulos = ContadorPulos + 1;
    Vx = CorpoRigido.velocity.x;
    Vy = VelocidadePular;
    Vector2 pulo = new Vector2(Vx, Vy);
    CorpoRigido.velocity = pulo;
}

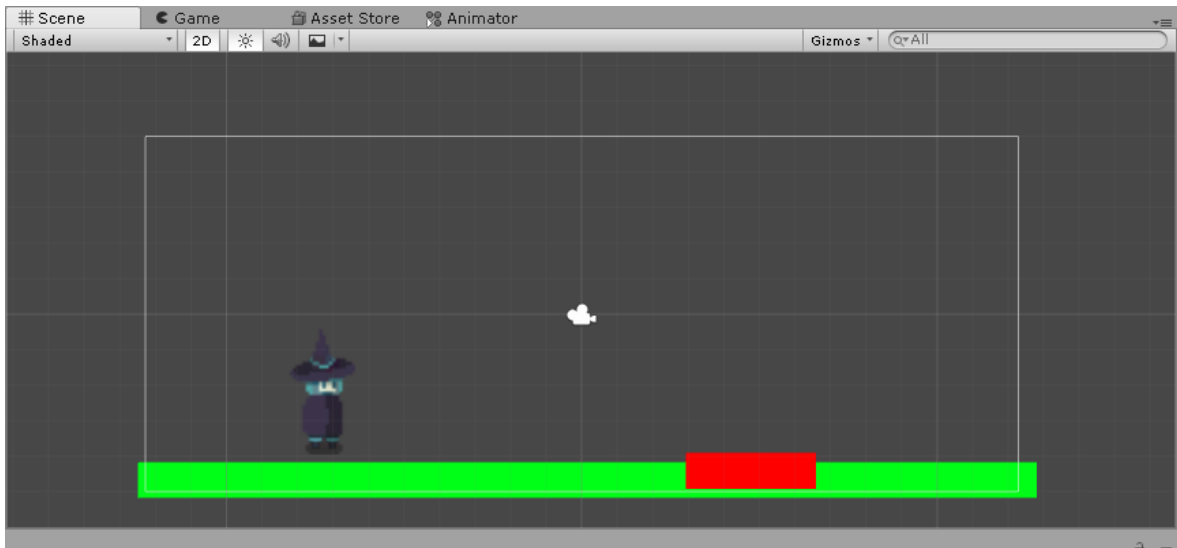
```

```

if ((apertou == true)&&(ContadorPulos == 1))
{
    VelocidadePular = 7;
}
if ((apertou == true) && (ContadorPulos == 2))
{
    VelocidadePular = 5;
}
if ((apertou == true) && (ContadorPulos == 3))
{
    VelocidadePular = 3;
}
}
}
}

```

8. Crie um pequeno jogo onde o personagem caminha por uma superfície. Em um determinado momento o personagem deve ser arremessado para o alto. (Utilize tags para marcar parte do chão, quando o jogador permanecer sobre uma determinada tag ele recebe um vetor de movimento com componente em y).



```

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

public class ex8 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;

    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
    float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado

    Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
    PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

    float ContadorPulos;

    GameObject meuObjeto = new GameObject(); // cria um novo objeto vazio

    //void Start () Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    //void Start () Executado na inicialização do jogo utilizado para iniciar variáveis e parametros

```

```

void Start()
{
    ContadorPulos = 0;

    Vx = 0;    //inicia a velocidade em x
    Vy = 0;    //inicia a velocidade em y

    DirecaoVertical = 0;    // inicia a direção vertical("andar para cima e para baixo")
    DirecaoHorizontal = 0;    // inicia a direção horizontal(andar)

    VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
    VelocidadeVertical = 5;    //inicia a velocidade de andar na vertical    (para cima e para baixo)
    VelocidadePular = 7;    //inicia a velocidade de pular

    CorpoRigido = GetComponent<Rigidbody2D>();    //recupera o corpo rígido inserido no elemento
    que possui o script(corpo rígido do personagem)

    Colisor = GetComponent<PolygonCollider2D>();    //recupera o colisor inserido no elemento que
    possui o script(colisor do personagem)

    CorpoRigido.freezeRotation = true;    //impede que rotacione no eixo.

    //inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
    = 1 deve ser igual 0

    CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame

//"void Update ()" Executado uma vez por frame.
void Update()
{
    MovimentoHorizontal();

    PuloTriplo();

    //MovimentoVertical ();

```

```
}
```

```
void MovimentoHorizontal()
```

```
{
```

```
//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.  
Quando parado Recebe 0.
```

```
DirecaoHorizontal = Input.GetAxis("Horizontal");
```

```
// Gera uma nova velocidade em x
```

```
Vx = VelocidadeHorizontal * DirecaoHorizontal;
```

```
//Recupera a velocidade em y que o personagem já possui
```

```
Vy = CorpoRigido.velocity.y;
```

```
//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
Vector2 andar = new Vector2(Vx, Vy);
```

```
// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
CorpoRigido.velocity = andar;
```

```
}
```

```
void MovimentoVertical()
```

```
{
```

```
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando  
parado Recebe 0.
```

```
DirecaoVertical = Input.GetAxis("Vertical");
```

```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu  
corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da  
velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade  
do corpo rígido do personagem
```

```
}
```

```

// pulo triplo
void OnCollisionEnter2D(Collision2D objetoTocado)
{
    string tagTocada = objetoTocado.gameObject.tag;
    if (tagTocada == "chao")
    {
        ContadorPulos = 0;
    }
    if (tagTocada == "Lanca")
    {
        // Define a nova posição do objeto com Y igual a 10
        Vector3 novaPosicao = new Vector3(transform.position.x, 10f, transform.position.z);

        // Atribui a nova posição ao componente Transform do objeto
        transform.position = novaPosicao;
    }
}

void PuloTriplo()
{
    //verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
    bool apertou = Input.GetButtonDown("Jump");
    // esse bloco só é executado se for apertado o botão de Jump
    // e se o personagem ContadorPulos < 2
    if (apertou == true && ContadorPulos < 3)
    {
        ContadorPulos = ContadorPulos + 1;
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular;
    }
}

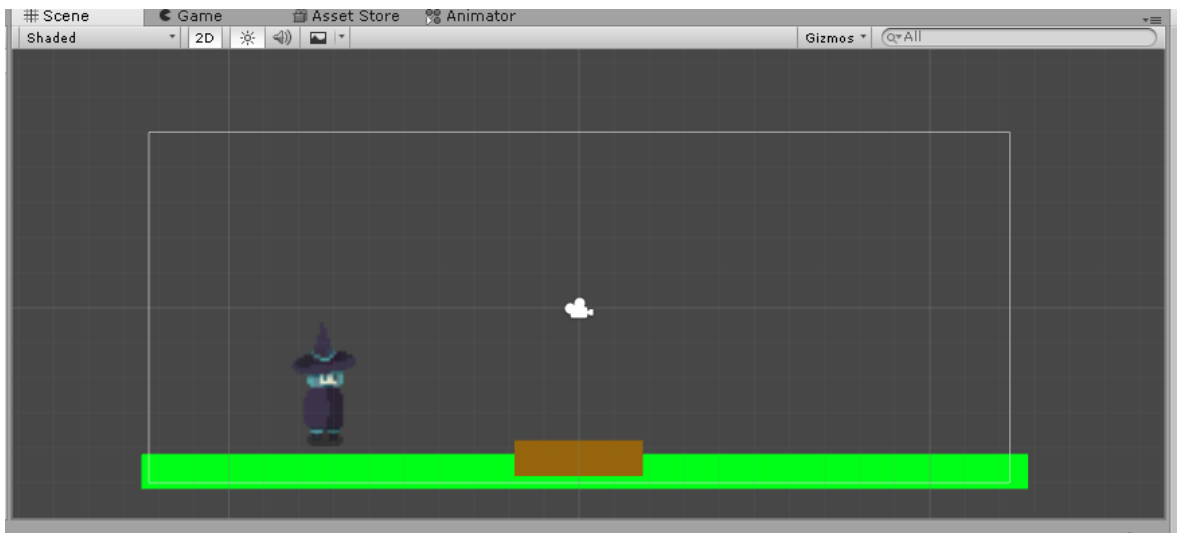
```

```

Vector2 pulo = new Vector2(Vx, Vy);
CorpoRigido.velocity = pulo;
}
}
}

```

9. Crie um pequeno jogo onde o personagem caminha por uma superfície. Em um determinado momento o personagem deve ser arremessado para o alto em um ângulo de 45°.



```

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

public class ex9 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;

    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
    float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado

```

Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.

PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

float ContadorPulos;

GameObject meuObjeto = new GameObject(); // cria um novo objeto vazio

//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros

//"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros

void Start()

{

ContadorPulos = 0;

Vx = 0; //inicia a velocidade em x

Vy = 0; //inicia a velocidade em y

DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")

DirecaoHorizontal = 0; // inicia a direção horizontal(andar)

VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)

VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)

VelocidadePular = 7; //inicia a velocidade de pular

CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento que possui o script(corpo rígido do personagem)

Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que possui o script(colisor do personagem)

CorpoRigido.freezeRotation = true; //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale = 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame



```
//"void Update ()" Executado uma vez por frame.
```

```
void Update()
```

```
{
```

```
MovimentoHorizontal();
```

```
PuloTriplo();
```

```
//MovimentoVertical ();
```

```
}
```

```
void MovimentoHorizontal()
```

```
{
```

```
//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.  
Quando parado Recebe 0.
```

```
DirecaoHorizontal = Input.GetAxis("Horizontal");
```

```
// Gera uma nova velocidade em x
```

```
Vx = VelocidadeHorizontal * DirecaoHorizontal;
```

```
//Recupera a velocidade em y que o personagem já possui
```

```
Vy = CorpoRigido.velocity.y;
```

```
//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
Vector2 andar = new Vector2(Vx, Vy);
```

```
// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
CorpoRigido.velocity = andar;
```

```
}
```

```
void MovimentoVertical()
```

```
{
```

```
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando  
parado Recebe 0.
```

```
DirecaoVertical = Input.GetAxis("Vertical");
```

```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
}
```

```
// pulo triplo
```

```
void OnCollisionEnter2D(Collision2D objetoTocado)
```

```
{
```

```
string tagTocada = objetoTocado.gameObject.tag;
```

```
if (tagTocada == "chao")
```

```
{
```

```
ContadorPulos = 0;
```

```
}
```

```
if (tagTocada == "Lanca")
```

```
{
```

```
// Define a nova posição do objeto com Y igual a 10
```

```
Vector3 novaPosicao = new Vector3(transform.position.x + 10f, transform.position.y + 10f, transform.position.z);
```

```
// Atribui a nova posição ao componente Transform do objeto
```

```
transform.position = novaPosicao;
```

```
}
```

```
}
```

```
void PuloTriplo()
```

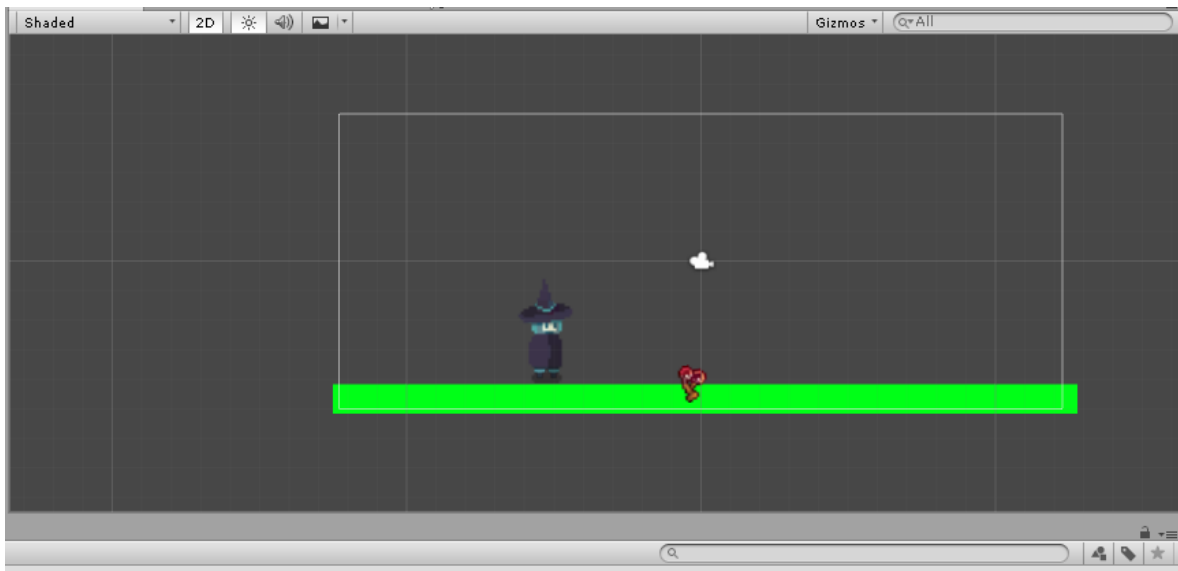
```
{
```

```
//verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
bool apertou = Input.GetButtonDown("Jump");

// esse bloco só é executado se for apertado o botão de Jump
// e se o personagem ContadorPulos < 2
if (apertou == true && ContadorPulos < 3)
{
    ContadorPulos = ContadorPulos + 1;

    Vx = CorpoRigido.velocity.x;
    Vy = VelocidadePular;
    Vector2 pulo = new Vector2(Vx, Vy);
    CorpoRigido.velocity = pulo;
}
}
}
```

10. Faça com que o jogador possa coletar cogumelos que permitam que o mesmo possa executar pulo triplo por 10 vezes.



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

```

public class ex10 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;

    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
    float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
    Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
    PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

    float ContadorPulos;
    float coguactive;
    float contadorcogu;

    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    void Start()
    {
        ContadorPulos = 0;
        Vx = 0; //inicia a velocidade em x
        Vy = 0; //inicia a velocidade em y
        DirecaoVertical = 0; // inicia a direção vertical("andar para cima e para baixo")
        DirecaoHorizontal = 0; // inicia a direção horizontal(andar)
        VelocidadeHorizontal = 5; //incia a velocidade de andar na horizontal(direita e esquerda)
        VelocidadeVertical = 5; //inicia a velocidade de andar na vertical (para cima e para baixo)
        VelocidadePular = 7; //inicia a velocidade de pular

        CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rigido inserido no elemento
        que possui o script(corpo rígido do personagem)

        Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
        possui o script(colisor do personagem)
    }
}

```

```

CorpoRigido.freezeRotation = true;          //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
= 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

//"void Update ()" Executado uma vez por frame

//"void Update ()" Executado uma vez por frame.
void Update()
{
    MovimentoHorizontal();
    PuloTriplo();
    //MovimentoVertical ();

}

void MovimentoHorizontal()
{
    //DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.
    Quando parado Recebe 0.

    DirecaoHorizontal = Input.GetAxis("Horizontal");

    // Gera uma nova velocidade em x
    Vx = VelocidadeHorizontal * DirecaoHorizontal;

    //Recupera a velocidade em y que o personagem já possui
    Vy = CorpoRigido.velocity.y;

    //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
    Vector2 andar = new Vector2(Vx, Vy);

    // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem

```

```
CorpoRigido.velocity = andar;
```

```
}
```

```
void MovimentoVertical()
```

```
{
```

```
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando parado Recebe 0.
```

```
DirecaoVertical = Input.GetAxis("Vertical");
```

```
Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu corpo rígido
```

```
// Gera uma nova velocidade em y
```

```
Vy = VelocidadeVertical * DirecaoVertical;
```

```
Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
```

```
CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
```

```
}
```

```
// pulo triplo
```

```
void OnCollisionEnter2D(Collision2D objetoTocado)
```

```
{
```

```
string tagTocada = objetoTocado.gameObject.tag;
```

```
if (tagTocada == "chao")
```

```
{
```

```
ContadorPulos = 0;
```

```
}
```

```
if (tagTocada == "cogu")
```

```
{
```

```
Destroy(objetoTocado.gameObject);
```

```
coguactive = 1;
```

```

}

}

void PuloTriplo()
{
//verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
bool apertou = Input.GetButtonDown("Jump");

// esse bloco só é executado se for apertado o botão de Jump
// e se o personagem ContadorPulos < 2
if ((apertou == true)&&(cogunactive == 1)&&(ContadorPulos < 3)&&(contadorcogu < 30))
{

ContadorPulos = ContadorPulos + 1;
contadorcogu = contadorcogu + 1;
Vx = CorpoRigido.velocity.x;
Vy = VelocidadePular;
Vector2 pulo = new Vector2(Vx, Vy);
CorpoRigido.velocity = pulo;

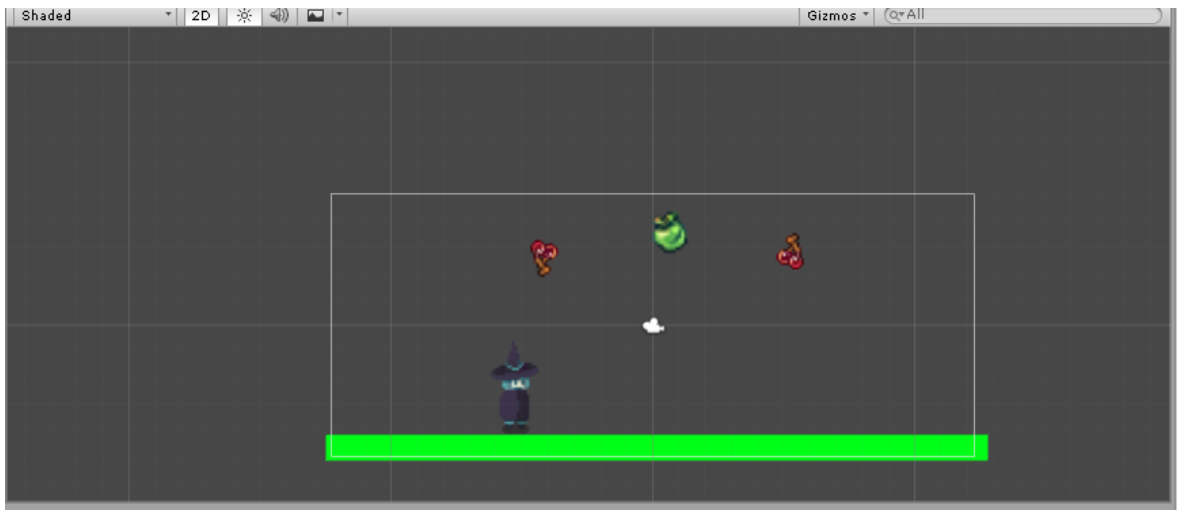
}

}

}

```

11. Faça um jogo que caiam coisas do céu, caso o jogador seja atingido por algo verde sua velocidade aumenta 2x. caso seja atingido por algo azul sua velocidade aumenta 3x. caso seja atingido por uma pedra sua velocidade diminui 4 vezes.



```

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

public class ex11 : MonoBehaviour
{
    float Vx; //velocidade em x
    float Vy; //velocidade em y
    float VelocidadeHorizontal;
    float VelocidadeVertical;
    float VelocidadePular;

    float DirecaoHorizontal; //variavel que armazena a direção: para esquerda, para a direita ou parado
    float DirecaoVertical; //variavel que armazena a direção: para cima, para a baixo ou parado
    Rigidbody2D CorpoRigido; //corpo rígido do elemento onde o script foi inserido.
    PolygonCollider2D Colisor; // colisor do elemento onde o script foi inserido.

    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    //"void Start ()" Executado na inicialização do jogo utilizado para iniciar variáveis e parametros
    void Start()
    {
        Vx = 0; //inicia a velocidade em x
        Vy = 0; //inicia a velocidade em y
    }
}

```



```

DirecaoVertical = 0;    // inicia a direção vertical("andar para cima e para baixo")
DirecaoHorizontal = 0;  // inicia a direção horizontal(andar)
VelocidadeHorizontal = 5; //inicia a velocidade de andar na horizontal(direita e esquerda)
VelocidadeVertical = 5; //inicia a velocidade de andar na vertical      (para cima e para baixo)
VelocidadePular = 7;    //inicia a velocidade de pular

CorpoRigido = GetComponent<Rigidbody2D>(); //recupera o corpo rígido inserido no elemento
que possui o script(corpo rígido do personagem)

Colisor = GetComponent<PolygonCollider2D>(); //recupera o colisor inserido no elemento que
possui o script(colisor do personagem)

CorpoRigido.freezeRotation = true;          //impede que rotacione no eixo.

//inicia a escala de gravidade com 1 - caso utilize MovimentoVertical() ao invés Pular() gravityScale
= 1 deve ser igual 0

CorpoRigido.gravityScale = 1;

}

```

```

//"void Update ()" Executado uma vez por frame

```

```

//"void Update ()" Executado uma vez por frame.

```

```

void Update()

```

```

{

```

```

MovimentoHorizontal();

```

```

//MovimentoVertical ();

```

```

}

```

```

void MovimentoHorizontal()

```

```

{

```

```

//DirecaoHorizontal Recebe até 1 quando pressionado o botão para direita e até -1 para esquerda.
Quando parado Recebe 0.

```

```

DirecaoHorizontal = Input.GetAxis("Horizontal");

// Gera uma nova velocidade em x
Vx = VelocidadeHorizontal * DirecaoHorizontal;

//Recupera a velocidade em y que o personagem já possui
Vy = CorpoRigido.velocity.y;

//Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
Vector2 andar = new Vector2(Vx, Vy);

// O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
CorpoRigido.velocity = andar;
}

/* void MovimentoVertical()
{
//DirecaoVertical Recebe até 1 quando pressionado o botão para cima e até -1 para baixo. Quando
parado Recebe 0.

DirecaoVertical = Input.GetAxis("Vertical");

Vx = CorpoRigido.velocity.x; //Recupera a velocidade em x que o personagem já possui em seu
corpo rígido

// Gera uma nova velocidade em y
Vy = VelocidadeVertical * DirecaoVertical;

Vector2 vetorMovimento = new Vector2(Vx, Vy); //Cria um vetor de velocidade com os valores da
velocidade em x e da velocidade em y (vx e vy)

CorpoRigido.velocity = vetorMovimento; // O vetor de velocidade é adicionado a velocidade
do corpo rígido do personagem
} */

void OnCollisionEnter2D(Collision2D objetoTocado)
{
string tagTocada = objetoTocado.gameObject.tag;

if (tagTocada == "Greenapple")

```

```

{
VelocidadeHorizontal = VelocidadeHorizontal * 2;

Destroy(objetoTocado.gameObject);
}

if (tagTocada == "Blueapple")
{
VelocidadeHorizontal = VelocidadeHorizontal * 3;

Destroy(objetoTocado.gameObject);
}

if (tagTocada == "Rock")
{
VelocidadeHorizontal = VelocidadeHorizontal / 4;

Destroy(objetoTocado.gameObject);
}

}

}

```

12. Faça o flappyBird tendo você, uma figura histórica ou o professor como personagem



using UnityEngine;

```
using System.Collections;

public class FlappyBird : MonoBehaviour {

    public float jumpForce = 200f;
    public GameObject gameOverText;

    private Rigidbody2D rb2d;
    private bool isDead = false;

    void Start () {
        rb2d = GetComponent<Rigidbody2D>();
    }

    void Update () {
        if (Input.GetButtonDown("Jump") && !isDead) {
            rb2d.velocity = Vector2.zero;
            rb2d.AddForce(new Vector2(0, jumpForce));
        }
    }

    void OnCollisionEnter2D(Collision2D other) {
        if (other.collider.CompareTag("Obstacle")) {
            isDead = true;
            gameOverText.SetActive(true);
        }
    }
}
```