



Ana Clara Keiko Ribeiro Hagiwara Da Silva

Gabriel Vazquez Mamede Diniz

Luiz Miguel de Toledo

Implementação da disciplina de Algoritmos e Lógica de  
Programação no jogo Beyond Worlds

**SÃO PAULO**

**2025**

Aplicabilidade da disciplina de  
Cálculo I produzido à Fundação Escola  
de Comércio Álvares Penteado, São Paulo,  
durante o 1º semestre do Bacharelado  
em Ciência da Computação.

Orientador: Prof. Renata Muniz do Nascimento

**SÃO PAULO**

**2025**

## 1. Movimentação do personagem

```
public float moveSpeed = 5f;
public float jumpHeight = 1.5f;
public float gravity = -9.81f;
public float mouseSensitivity = 2f;
public float runSpeed = 10f;
private bool isRunning = false;
private float lastWPressTime = -1f;
public float doubleTapTime = 0.3f;

private Vector3 velocity;
private bool isGrounded;
private CharacterController controller;

private Transform cam;

private float xRotation = 0f;
```

Para o início do código, é importante declarar as variáveis, tais como a da velocidade padrão do jogador, a velocidade quando está correndo, o tamanho do pulo, a sensibilidade do mouse entre outros.

```
void Start()
{
    controller = GetComponent<CharacterController>();
    cam = transform.Find("Main Camera");
    Cursor.lockState = CursorLockMode.Locked;
}
```

Logo depois, no void start, é essencial a mudança de posição da câmera principal, assim como desativar o cursor do mouse.

```
// checagem de solo
isGrounded = controller.isGrounded;
if (isGrounded && velocity.y < 0) {
    velocity.y = -2f; // mantém o personagem preso ao chão
}
```

Já no void update, é fundamental checar onde está o solo.

```
// entrada de movimento
float moveX = Input.GetAxis("Horizontal");
float moveZ = Input.GetAxis("Vertical");
```

Aqui é recebido a entrada de movimento, que já deixa o nosso personagem se mover

```
// Detectar duplo toque no W
if (Input.GetKeyDown(KeyCode.W))
{
    if (Time.time - lastWPressTime < doubleTapTime)
    {
        isRunning = true;
    }
    lastWPressTime = Time.time;
}
```

Para finalizar, criamos um sistema de velocidade, que, quando clicado duas vezes na tecla “W”, será mudado a velocidade do personagem, para assim criar velocidade.

## 2. Pulo

```
// pulo
if (Input.GetButtonDown("Jump") && isGrounded)
{
    velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);
}
```

Agora é inserido a funcionalidade de pulo

```
// gravidade
velocity.y += gravity * Time.deltaTime;
controller.Move(velocity * Time.deltaTime);
```

Que vem acompanhado da gravidade, para que assim que o jogador pule, acabe voltando ao chão novamente.

## 3. Rotação do mouse

```
// rotação com o mouse
float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity;
transform.Rotate(Vector3.up * mouseX);

float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity;
xRotation -= mouseY;
xRotation = Mathf.Clamp(xRotation, -90f, 90f);
cam.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
```

Esse código possibilita a rotação do mouse tanto para o eixo X, quanto o eixo Y.