



Ana Clara Keiko Ribeiro Hagiwara Da Silva

Gabriel Vazquez Mamede Diniz

Luiz Miguel de Toledo

Aplicabilidade da disciplina de Cálculo I 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 horizontal

Para iniciarmos, começaremos pelo script que anda horizontalmente. Determinado por:

```
Vector3 move = transform.right * moveX + transform.forward * moveZ;  
controller.Move(move * currentSpeed * Time.deltaTime);
```

Equação:

$$D = v \cdot t$$

- *move* é a direção.
- *currentSpeed* é a velocidade escalar (ou *runSpeed* se correndo).
- *Time.deltaTime* é o tempo entre frames (*t*).

Então o deslocamento aplicado é:

$$D = (\text{moveX} \cdot r + \text{moveZ} \cdot f) \cdot v \cdot \Delta t$$

Onde:

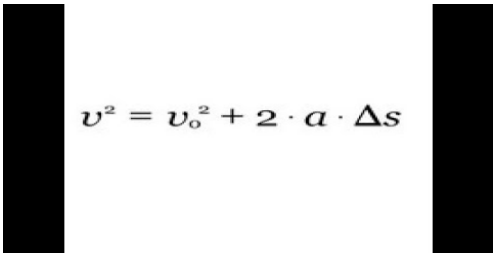
- *r* = vetor para a direita (*transform.right*)
- *f* = vetor para frente (*transform.forward*)
- *v* = *currentSpeed*

2. Pulo

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

Aqui usamos a equação da física para calcular a **velocidade inicial do pulo**:

Equação de Torricelli:


$$v^2 = v_o^2 + 2 \cdot a \cdot \Delta s$$

Exemplo: se jumpHeight = 1.5f e gravity = -9.81f, então o VO será 5,43m/s

3. Gravidade

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

Aqui, a força da gravidade altera a **velocidade vertical** ao longo do tempo:

Equação:

$$V = v_0 + a \cdot t$$

Onde:

- $a = \text{gravity}$
- $t = \Delta t$

Depois, o deslocamento vertical é aplicado:

$$\Delta y = v \cdot \Delta t$$

4. Corrida com toque duplo

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

Aqui temos uma comparação de tempos:

Condição:

$$V = \text{Tagora} - \text{Túltimotouque} < \text{doubleTapTime}$$

Se a diferença de tempo entre dois toques na tecla W for menor que doubleTapTime, o personagem começa a correr (velocidade dobra).

5. Rotação com o 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);
```

Horizontal:

$$\theta_{\text{horizontal}} += \Delta x_{\text{mouse}} \cdot \text{sensibilidade}$$

Vertical:

$$\theta_{\text{vertical}} -= \Delta y_{\text{mouse}} \cdot \text{sensibilidade} \text{ (limitado entre } -90^\circ \text{ e } 90^\circ)$$

Essa rotação vertical usa Quaternion.Euler, uma representação matemática de rotações em 3D com ângulos de Euler.