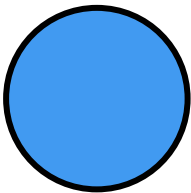


# Animação

Fundamentos de Computação Gráfica (INF01047)

**Prof. Eduardo Gastal**

<https://inf.ufrgs.br/~eslgastal>



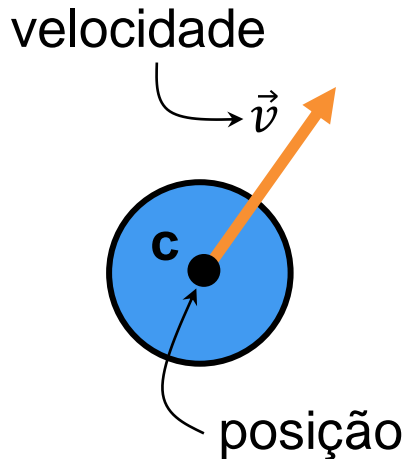
```
while (true)
    // movimenta bola
    c = c +  $\vec{v}$ ;
    render();
```

? X

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \int_0^t \vec{v}(t') dt'$$

Errado pois velocidade da animação irá depender da velocidade de processamento.

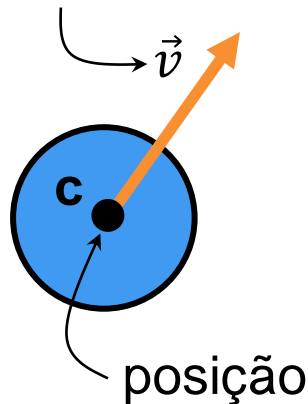


```
while (true)
    // movimenta bola
     $\mathbf{c} = \mathbf{c} + \vec{v}$ ; ?
    render();
```

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \int_0^t \vec{v} dt'$$

velocidade



```
while (true)
    // movimenta bola
```

```
    c = c +  $\vec{v}$ ;
```

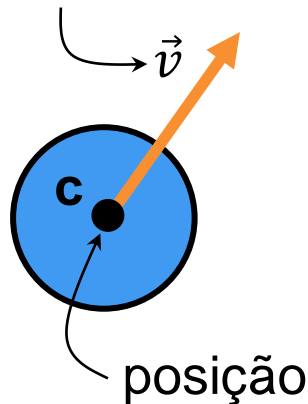
?

```
    render();
```

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} \boxed{t}$$

velocidade



```
while (true)
```

```
    // movimenta bola
```

```
    c = c +  $\vec{v}$  t;
```

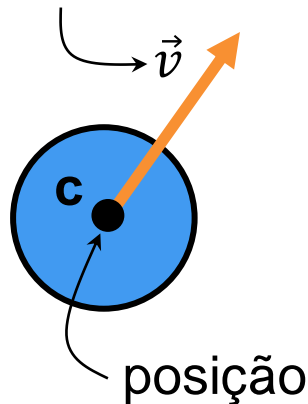
? X

```
    render();
```

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

velocidade

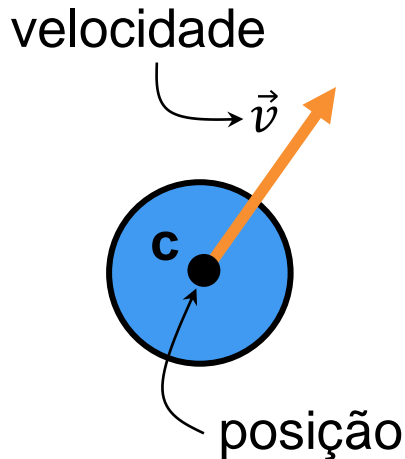


```
while (true)
    // movimenta bola
     $\mathbf{c} = \mathbf{c} + \vec{v}$ ; ?
    render();
```

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

$$\mathbf{c}(t + \Delta t) = \mathbf{c}(0) + \vec{v}(t + \Delta t)$$



```
while (true)
    // movimenta bola
     $\mathbf{c} = \mathbf{c} + \vec{v}$ ;
    render();
```

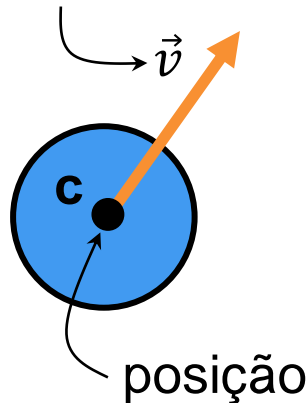
?

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

$$\mathbf{c}(t + \Delta t) = \mathbf{c}(0) + \vec{v} t + \vec{v} \Delta t$$

velocidade





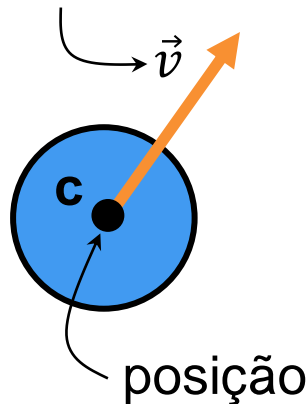
```
while (true)
    // movimenta bola
     $\mathbf{c} = \mathbf{c} + \vec{v};$ 
    render();
```

Integração numérica

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

$$\mathbf{c}(t + \Delta t) = \mathbf{c}(t) + \vec{v} \Delta t$$

velocidade



```
while (true)
```

```
    // movimenta bola
```

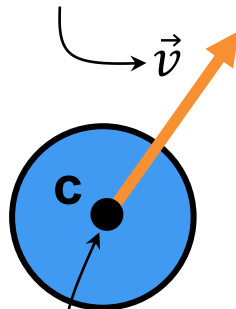
```
     $\Delta t = \dots$ 
```

```
     $\mathbf{c} = \mathbf{c} + \vec{v} \Delta t;$ 
```

```
    render();
```

Exato se  $\vec{v}$   
for constante.  
Aproximado  
se  $\vec{v}$  variar  
com o tempo.

velocidade



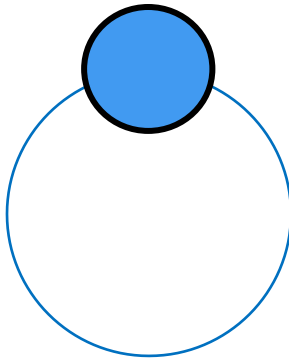
posição

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

**Euler method**

$\mathbf{c}(t + \Delta t) = \mathbf{c}(t) + \vec{v} \Delta t$  **E se  $\vec{v}(t)$  não for constante  $\forall t$ ?**

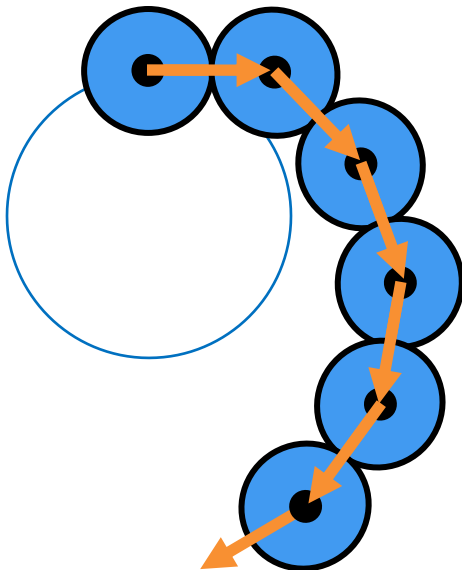
# Integration Error



# Integration Error

Para movimentação da câmera e personagens controlados com as teclas WASD, a animação através do método de Euler é adequada.

A função de velocidade  $\vec{v}(t)$  é definida pelo usuário a partir do “feedback” que este recebe do sistema. Logo, qualquer desvio da trajetória esperada será corrigido pelo próprio usuário.



- **Julia pluto notebook**

```
while (true)
```

```
    // movimenta bola
```

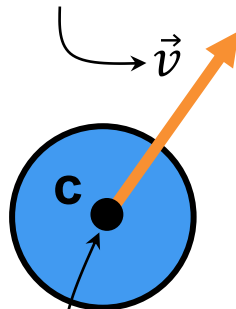
```
     $\Delta t = \dots$ 
```

```
     $\mathbf{c} = \mathbf{c} + \vec{v} \Delta t;$ 
```

```
    render();
```

Exato se  $\vec{v}$   
for constante.  
Aproximado  
se  $\vec{v}$  variar  
com o tempo.

velocidade



posição

$$\mathbf{c}(t) = \mathbf{c}(0) + \vec{v} t$$

$$\mathbf{c}(t + \Delta t) = \mathbf{c}(t) + \vec{v} \Delta t$$

```
 $t_{\text{prev}} = \text{glfwGetTime}();$  velocidade
```

```
while (true)
```

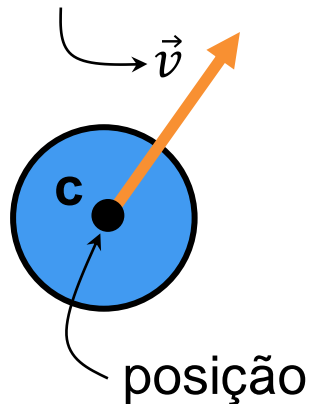
```
    // movimenta bola
```

```
     $t_{\text{now}} = \text{glfwGetTime}();$ 
```

```
     $\Delta t = t_{\text{now}} - t_{\text{prev}};$ 
```

```
     $\mathbf{c} = \mathbf{c} + \vec{v} \Delta t;$ 
```

```
    render();
```



```
 $t_{\text{prev}} = \text{glfwGetTime}();$  velocidade
```

```
while (true)
```

```
    // movimenta bola
```

```
     $t_{\text{now}} = \text{glfwGetTime}();$ 
```

```
     $\Delta t = t_{\text{now}} - t_{\text{prev}};$ 
```

```
     $t_{\text{prev}} = t_{\text{now}};$ 
```

```
     $\mathbf{c} = \mathbf{c} + \vec{v} \Delta t;$ 
```

```
    render();
```

