



Introdução à Engenharia

ENG1000

Aula 10 – Estruturas Condicionais
e Interação em Love2D
2016.1



Prof. Augusto Baffa
<abaffa@inf.puc-rio.br>



De Volta ao “Hello World”

- Na última implementação do “Hello World” fizemos o texto se mover na tela
 - **Problema:** quando a posição do texto passava do limite da tela, o texto sumia (continuava se movendo para longe da tela)
- Com uma estrutura condicional podemos fazer o texto voltar para o início da tela.
- Como podemos fazer isso?

De Volta ao “Hello World”

```
local px      -- posição x do texto

function love.load()
    love.graphics.setColor(0, 0, 0)
    love.graphics.setBackgroundColor(255, 255, 255)
    px = 0
end

function love.update(dt)
    px = px + (100 * dt)
end

function love.draw()
    love.graphics.print("Hello World", px, 300)
end
```

```
local px                -- posição x do texto

function love.load()
    love.graphics.setColor(0, 0, 0)
    love.graphics.setBackgroundColor(255, 255, 255)
    px = 0
end

function love.update(dt)
    px = px + (100 * dt)

    if px > 800 then    -- a largura da janela é 800
        px = 0
    end
end

function love.draw()
    love.graphics.print("Hello World", px, 300)
end
```

```
local px                -- posição x do texto

function love.load()
    love.graphics.setColor(0, 0, 0)
    love.graphics.setBackgroundColor(255, 255, 255)
    px = 0
end

function love.update(dt)
    px = px + (100 * dt)

    if px > love.window.getWidth() then
        px = 0
    end

end

function love.draw()
    love.graphics.print("Hello World", px, 300)
end
```

Uma forma mais geral de
acessar a largura da janela.
Available since LÖVE [0.9.0](#)
and removed in LÖVE [0.10.0](#)

```
local px                -- posição x do texto

function love.load()
    love.graphics.setColor(0, 0, 0)
    love.graphics.setBackgroundColor(255, 255, 255)
    px = 0
end

function love.update(dt)
    px = px + (100 * dt)

    if px > love.graphics.getWidth() then
        px = 0
    end

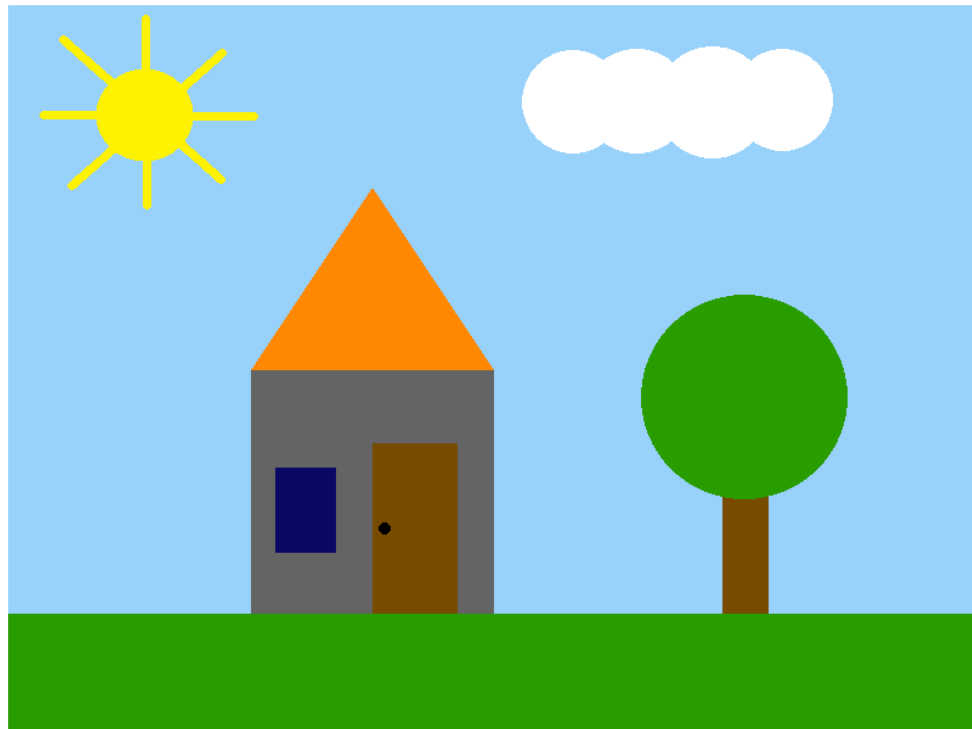
end

function love.draw()
    love.graphics.print("Hello World", px, 300)
end
```

Uma forma mais geral de
acessar a largura da janela.
Available since LÖVE [0.10.0](#)

Exercício 1

1. Estenda o programa do exercício da aula anterior criando uma animação que faça com que a nuvem se mova de um lado para o outro da tela. Se a nuvem estiver tampando o sol, então a cor de fundo do cenário deve ficar mais escura.



Módulo love.keyboard

- O módulo love.keyboard contem funções dedicadas a interação pelo teclado.
- É possível verificar se um determinada tecla foi pressionada usando o comando love.keyboard.isDown

```
love.keyboard.isDown(key)
```

- A função retorna verdadeiro se a tecla passada como parâmetro estiver pressionada.

Módulo love.keyboard

- É necessário utilizar uma **estrutura condicional** para verificar se a tecla foi pressionada.
- **Exemplo:**

```
if love.keyboard.isDown("right") then
    px = px + (100 * dt)
end
```

- Lista de teclas: <http://www.love2d.org/wiki/KeyConstant>

De Volta ao “Hello World”

```
local px      -- posição x do texto

function love.load()
    love.graphics.setColor(0, 0, 0)
    love.graphics.setBackgroundColor(255, 255, 255)
    px = 0
end

function love.update(dt)

    if love.keyboard.isDown("right") then
        px = px + (100 * dt)
    end

end

function love.draw()
    love.graphics.print("Hello World", px, 300)
end
```

Função `love.keypressed`

- A função `love.keypressed` é um callback chamado sempre que o usuário pressiona uma tecla no teclado. Permite fazer a leitura do teclado de forma independente.

```
love.keypressed( key, scancode, isrepeat )  
Available since LÖVE 0.10.0
```

- Key – tecla pressionada
- Scancode – código da tecla pressionada
- Isrepeat – indica se a tecla é uma repetição

```
love.keypressed( key, isrepeat )  
Available since LÖVE 0.9.0 and removed in LÖVE 0.10.0
```

Função love.keyreleased

- A função love.keyreleased é um callback chamado sempre que o usuário solta uma tecla no teclado. Permite fazer a leitura do teclado de forma independente.

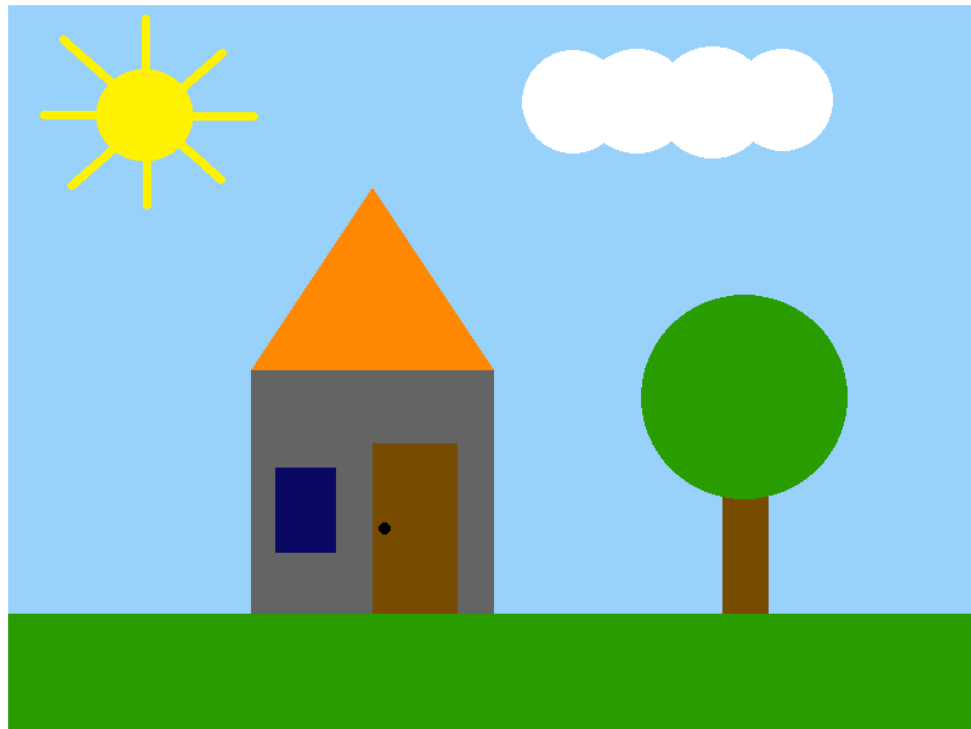
```
love.keyreleased( key, scancode )  
Available since LÖVE 0.10.0
```

- Key – tecla pressionada
- Scancode – código da tecla pressionada

```
love.keyreleased( key )
```

Exercício 2

2. Estenda o programa do exercício anterior para permitir que o usuário possa movimentar o sol no eixo X e Y usando as setas direcionais do teclado.



Módulo love.mouse

- O módulo love.mouse também permite acessar a posição do mouse através dos comandos love.mouse.getX e love.mouse.getY

```
love.mouse.getX()
```

```
love.mouse.getY()
```

- As funções retornam a posição do mouse relativa a janela do programa nos eixos X e Y.

```
mousex = love.mouse.getX()  
mousey = love.mouse.getY()  
...  
mousex, mouseY = love.mouse.getPosition()
```

Módulo love.mouse

- O módulo love.mouse contem funções dedicadas a interação pelo mouse.
- É possível verificar se um determinado botão do mouse foi pressionada usando o comando love.mouse.isDown

```
love.mouse.isDown(button)
```

- A função retorna verdadeiro se o botão passada como parâmetro estiver pressionado.

Módulo love.mouse

- É necessário utilizar uma **estrutura condicional** para verificar se o botão foi pressionado.
- **Exemplo:**

```
if love.mouse.isDown("l") then  
    texto = "Left! :)"
```

End

Removed in LÖVE [0.10.0](#)

```
if love.mouse.isDown(1) then  
    texto = "Left! :)"
```

End

Available since LÖVE [0.10.0](#)

- Lista de teclas: <http://www.love2d.org/wiki/MouseConstant>

De Volta ao “Hello World”

- Utilizando as funções de interação pelo mouse, podemos modificar o “Hello World” para permitir que:
 - O usuário possa movimentar o texto com o mouse;
 - O texto seja modificado quando o usuário clicar com o mouse:
 - Botão da esquerda: “Left (l)” até o [0.10.0](#) e 1 atualmente
 - Botão da direita: “Right (r)” até o [0.10.0](#) e 2 atualmente
 - Botão do meio: “middle (m)” até o [0.10.0](#) e 3 atualmente
 - Nenhum botão: “Hello World!”
- Como podemos fazer isso?

De Volta ao “Hello World”

```
local px      -- posição x do texto
local py      -- posição y do texto
local texto = "Hello World!"

function love.update(dt)
    if love.mouse.isDown("l") then
        texto = "Left! :)"
    elseif love.mouse.isDown("r") then
        texto = "Right! :)"
    else
        texto = "Hello World!"
    end
    px = love.mouse.getX()
    py = love.mouse.getY()
end

function love.draw()
    love.graphics.print(texto, px, py)
end
```

Função love.mousepressed

- A função love.mousepressed é um callback chamado sempre que o usuário pressiona um botão do mouse. Permite fazer a leitura dos botões de forma independente.

```
love.mousepressed( x, y, button, istouch )  
Available since LÖVE 0.10.0
```

- x, y – posições do mouse no momento do clique
- button – botão pressionado
- Istouch – se a interface é touch

```
love.mousepressed( x, y, button)  
Available since LÖVE 0.9.0 and removed in LÖVE 0.10.0
```

Função love.mousepressed

- A função love.mousepressed é um callback chamado sempre que o usuário pressiona um botão do mouse. Permite fazer a leitura dos botões de forma independente.

```
love.mousepressed( x, y, button, istouch )  
Available since LÖVE 0.10.0
```

- x, y – posições do mouse no momento do clique
- button – botão pressionado
- Istouch – se a interface é touch

```
love.mousepressed( x, y, button )  
Available since LÖVE 0.9.0 and removed in LÖVE 0.10.0
```

Função love.mousereleased

- A função love.mousereleased é um callback chamado sempre que o usuário solta um botão do mouse. Permite fazer a leitura dos botões de forma independente.

```
love.mousereleased( x, y, button, istouch )
```

Available since LÖVE [0.10.0](#)

- x, y – posições do mouse no momento do clique
- button – botão pressionado
- Istouch – se a interface é touch

```
love.mousereleased( x, y, button)
```

Available since LÖVE [0.9.0](#) and removed in LÖVE [0.10.0](#)

Função love.mousemoved

- A função love.mousemoved é um callback chamado sempre que o usuário movimenta o mouse. Permite fazer a leitura dos movimentos de forma independente.

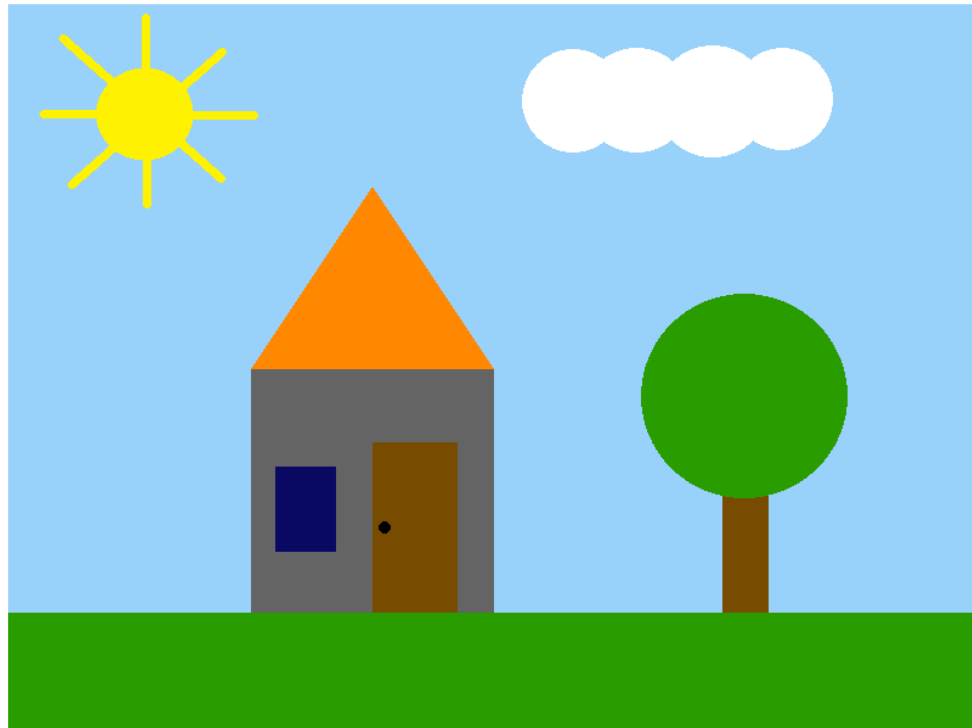
```
love.mousemoved ( x, y, dx, dy )
```

Available since LÖVE [0.9.2](#)

- x, y – posições do mouse no momento do clique
- dx, dy – quanto se movimentou desde a última posição

Exercício 3

3. Modifique o programa do exercício anterior para permitir que o usuário possa movimentar o sol utilizando o mouse.



Exercícios

Lista de Exercícios 06 – Interação Teclado e Mouse

<http://www.inf.puc-rio.br/~abaffa/eng1000/>