



Introduction to love2d

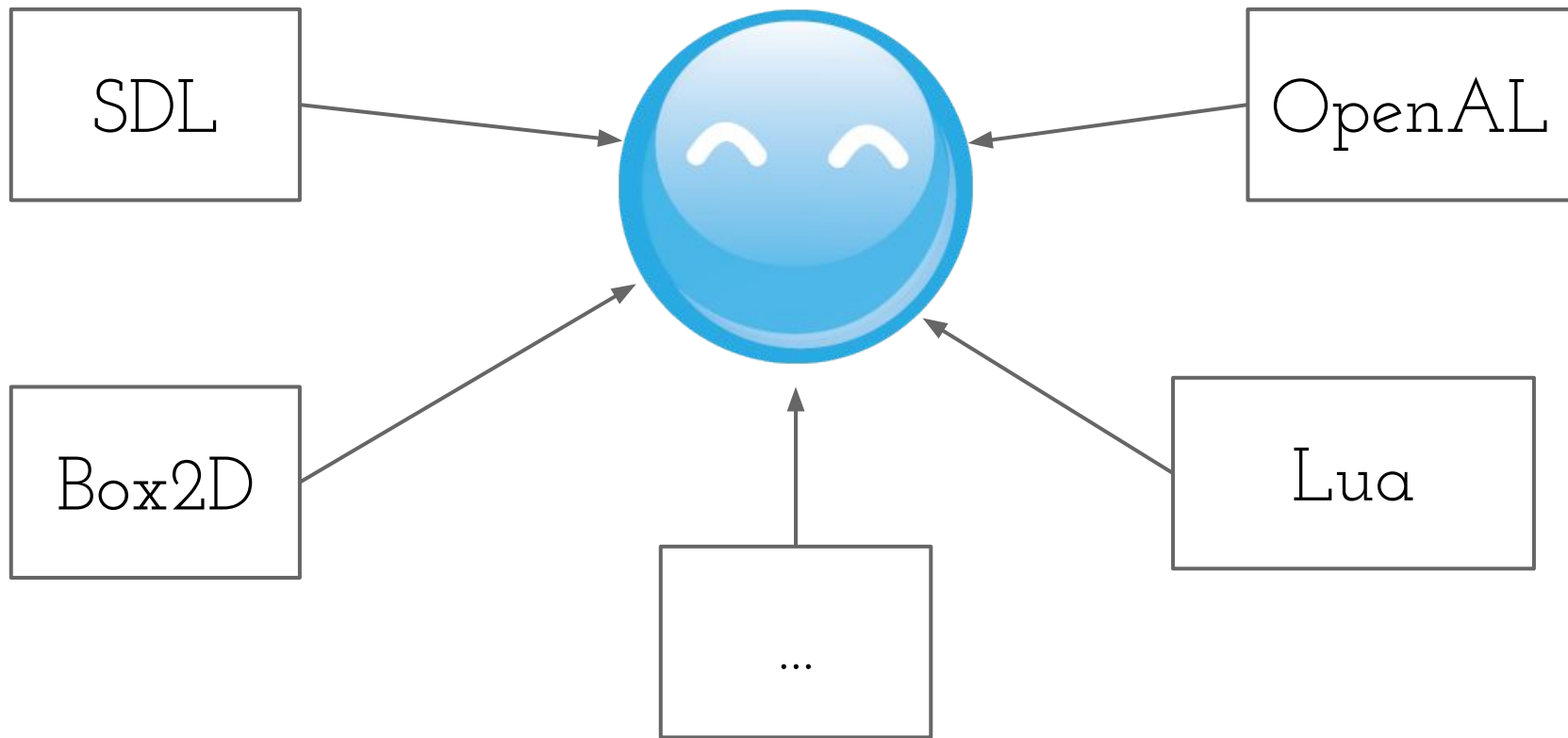
What is love?

A framework to
develop 2d games

What love is not

1. Programming language
2. Drag and drop game engine
3. Best game engine

What is love made of?



Getting started

1. Grab love from <http://love2d.org>
2. Install it
3. Create your game folder
4. Create "main.lua" inside that folder
5. Start coding!

Love has a very organized
module structure:

love.audio Provides an interface to output
sound to the user's speakers.

love.event Manages events, like keypresses.

love.filesystem Provides an interface to the
user's filesystem.

love.font Allows you to work with fonts.

love.graphics Drawing of shapes and images,
management of screen geometry.

love.image Provides an interface to decode
encoded image data.

love.math Provides system-independent
mathematical functions.

love.joystick Provides an interface to connected joysticks.

love.keyboard Provides an interface to the user's keyboard.

love.mouse Provides an interface to the user's
mouse.

love.physics Can simulate 2D rigid body
physics in a realistic manner.

love.sound This module is responsible for
decoding sound files.

love.system Provides access to information about
the user's system.

love.thread Allows you to work with
threads.

love.timer Provides an interface to your
system's clock.

love.window Provides an interface for the
program's window.

Your mine of gold

<http://www.love2d.org/wiki/>

```
function love.draw()  
    love.graphics.print("Hello World", 400, 300)  
end
```



Let's say you named your game
folder "hello_world"

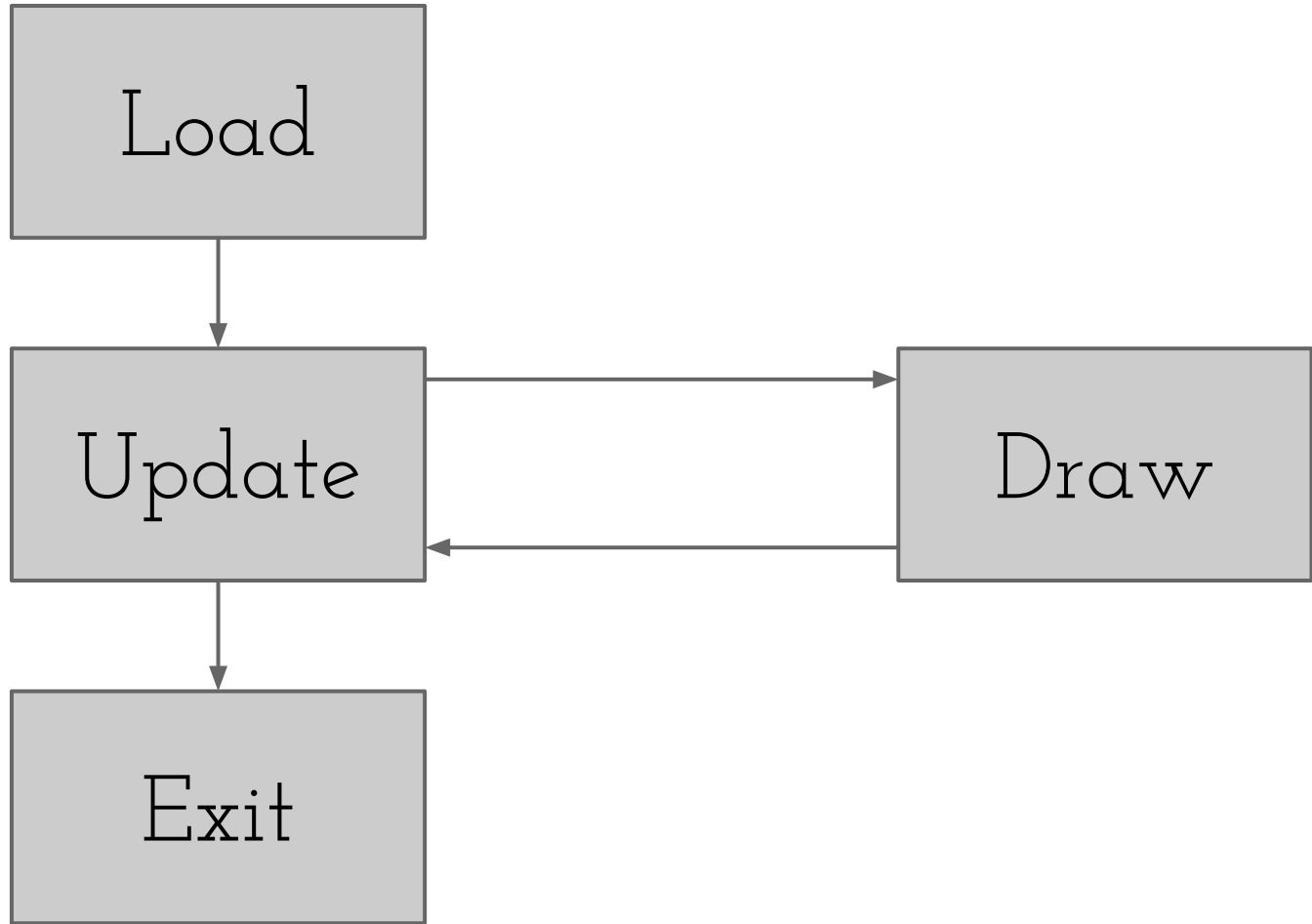
```
user@laptop ~$ love ./hello_world
```

How does it work

1. Love will search for main.lua and conf.lua
2. It will configure the game
3. Run it

Callback functions

1. Handled by the game engine
2. Called continuously
3. Need to be overloaded



love.load ()

- ❑ First function to be executed
- ❑ Initializes your game environment
- ❑ Will never be executed again (unless if you call it)

love.update(dt)

- ❑ Called each frame
- ❑ dt is the timing difference between previous and current frame
- ❑ Most of your math will be here

love.draw ()

- ❑ This is where you will draw all of your stuff
- ❑ Next function will be called right after is love.update(dt)

love.exit ()

- ❑ Called with the user request to exit the game

Drawing

Drawing basics

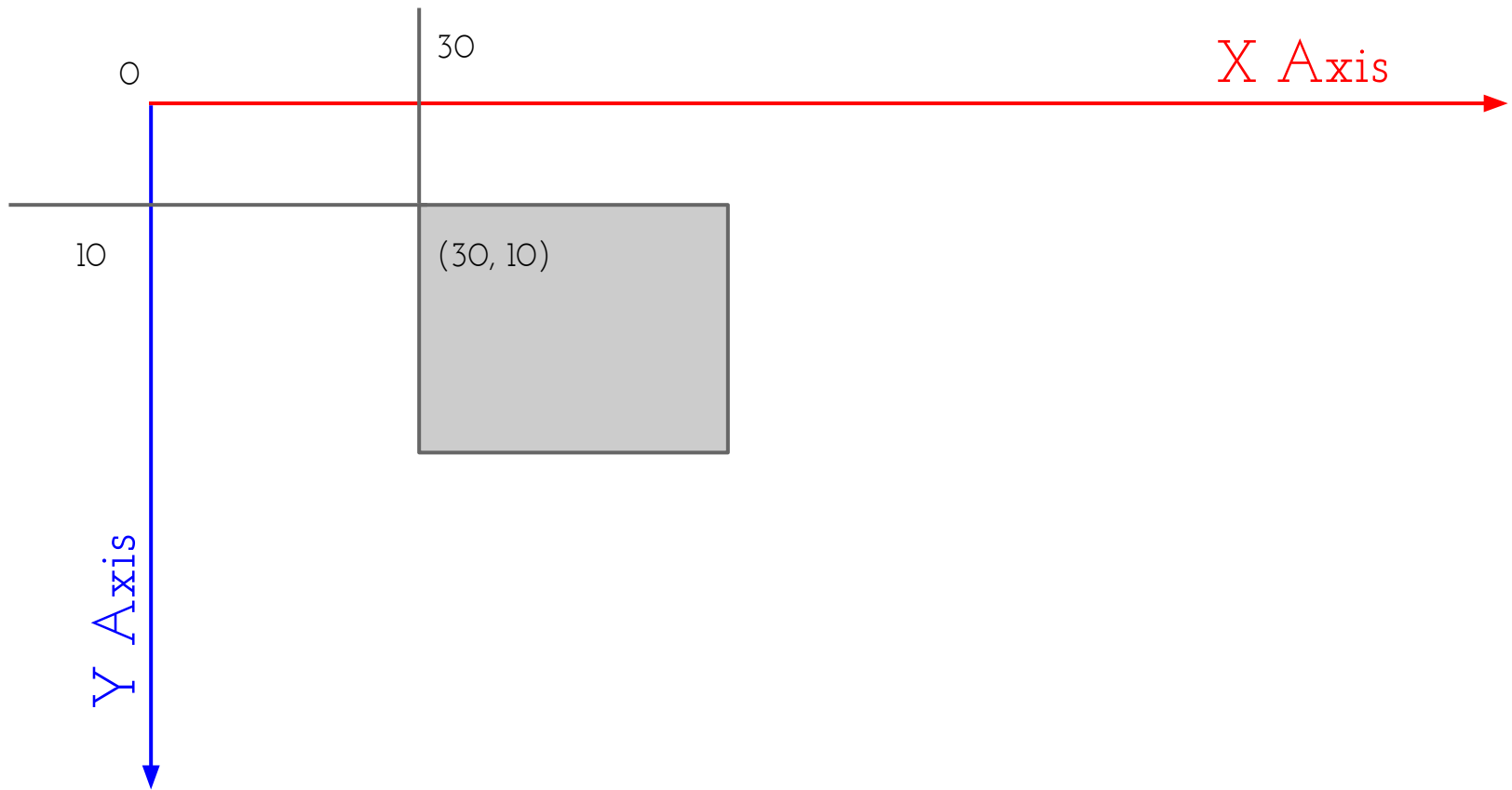


Image object

```
image = love.graphics.newImage( filename )
```

ARGUMENTS

string filename

The filepath to the image file.

RETURNS

Image image

An Image object which can be drawn on screen.

Drawing image object

```
function love.load()  
    image = love.graphics.newImage("img.png")  
end
```

```
function love.draw()  
    love.graphics.draw(image, 100, 100)  
end
```


Advanced drawing

```
love.graphics.draw( drawable, x, y, r, sx, sy, ox, oy)
```

ARGUMENTS

Drawable `drawable`

A drawable object.

number `x` (0)

The position to draw the object (x-axis).

number `y` (0)

The position to draw the object (y-axis).

number `r` (0)

Orientation (radians).

number `sx` (1)

Scale factor (x-axis).

number `sy` (`sx`)

Scale factor (y-axis).

number `ox` (0)

Origin offset (x-axis).

number `oy` (0)

Origin offset (y-axis).

RETURNS

Nothing.

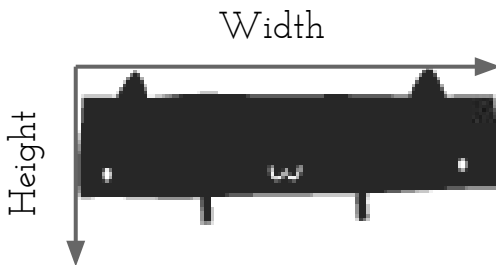
Canvas	Off-screen render target.	0.8.0	
Framebuffer	Off-screen render target.	0.7.0	0.8.0
Image	Drawable image type.		
Mesh	A 2D polygon mesh used for drawing arbitrary textured shapes.	0.9.0	
ParticleSystem	Used to create cool effects, like fire.		
SpriteBatch	Store image positions in a buffer, and draw it in one call.		
Texture	Superclass for drawable objects which represent a texture.	0.9.1	

Example 1

```
function love.draw()  
    love.graphics.draw(image, 300, 300, math.pi)  
end
```

Example 2

```
function love.draw()  
  love.graphics.draw(image, 300, 300, math.pi  
    , 1, 1  
    , image:getWidth() / 2  
    , image:getHeight() / 2 )  
end
```



Window size


```
width = love.window.getWidth( )
```

ARGUMENTS

None.

RETURNS

number width

The width of the window.

```
height = love.window.getHeight( )
```

ARGUMENTS

None.

RETURNS

number height

The width of the window.

Example 3

```
function love.draw()  
    love.graphics.draw(image,  
                        love.window:getWidth()/2,  
                        love.window:getHeight()/2,  
                        0, 1, 1  
                        , image:getWidth() / 2  
                        , image:getHeight() / 2 )  
  
end
```

Understanding love.update

A simple animation

local object = { img = nil, x = 30, y = 30, rot = 0 }

```
function love.load ( )  
    love.graphics.setBackgroundColor(255,  
                                     255,  
                                     255)  
  
    object.img = love.graphics.newImage(  
        "imageUrl.png")  
  
end
```



```
function love.update ( dt )  
    object.x = object.x + 5  
    object.y = object.y + 5  
end
```

```
function love.draw ( )  
    love.graphics.draw(object.img, object.x, object.y)  
end
```

Drawing Text

```
function love.draw ( )  
    love.graphics.print("Hello!", 100, 100)  
end
```

Changing color

```
function love.draw ( )  
    love.graphics.setColor(234, 38, 35)  
    love.graphics.print("Hello!", 100, 100)  
  
    love.graphics.setColor(58, 167, 75)  
    love.graphics.print("Awesome news!", 100, 300)  
end
```

Changing **Font**

```
local fText
```

```
function love.load( )  
    fText = love.graphics.newFont("url.otf")  
end
```



```
function love.draw ( )  
    love.graphics.setFont(fText)  
    love.graphics.print("Awesome news!", 100, 300)  
end
```

Drawing shapes

love.graphics.arc

love.graphics.circle

love.graphics.line

love.graphics.point

love.graphics.polygon

love.graphics.rectangle

Playing



Using TEsound library

How does it work

1. Get TEsound.lua file in your game folder
2. Add **require** "TEsound"
3. Add TEsound.cleanup() in love.update

Playing sounds

Loading sound



Always load sound in memory
then Play it

```
local snd = love.sound.newSoundData(  
    "SoundUrl.wav")
```

```
function love.load( )  
    TEsound.play(snd, "bg")  
end
```

```
function love.keyreleased( key )  
  if key == "a" then  
    TEsound.play("bg")  
  elseif key == "z" then  
    TEsound.pause("bg")  
  end  
end
```

Exercise

Hamster Ball