THE CALLBACKS

```
love.load(args)     Run once when the game starts.
love.update(dt)     Used to update the game's state.
love.draw()         Used to draw to the screen.
love.quit()         Run when the game is quit.
                    Return true to cancel quitting.
love.focus(focused)
```
Run when the window gains/loses focus.

THE KEYBOARD

Callbacks
```
love.keypressed(key, unicode)
love.keyreleased(key, unicode)
```

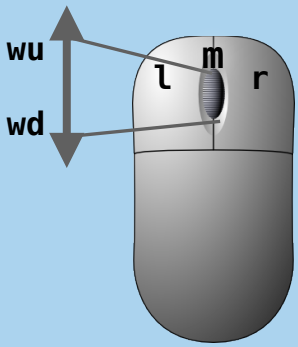
Module love.keyboard
```
isDown(key1, ...)
delay, interval = getKeyRepeat()
setKeyRepeat(delay, interval)
```

THE MOUSE

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

Module love.mouse
```
x, y = getPosition()
getX() getY()
isDown(button1, ...)
isGrabbed()
isVisible()
setGrab(grab)
setPosition(x, y)
setVisible(visible)
```

Enum MouseConstant



AUDIO

Module love.audio
```
pause()                          pause(source)
                                 play(source)
resume()                         resume(source)
rewind()                         rewind(source)
stop()                           stop(source)
getNumSources()
newSource(file, type)
newSource(data)
newSource(decoder, type)
```

Getters/Setters (prefix names with "get" or "set")
```
Orientation(fx, fy, fz, ux, uy, uz)
Position(x, y, z)
Velocity(x, y, z)
Volume(volume)
```

MODULE LOVE.GRAPHICS

Drawing
```
draw(drawable, x, y, r, sx, sy, ox, oy)
drawq(img, quad, x, y, r, sx, sy,ox,oy)
```

New Objects (prefix names with "new")
```
Font(filename, size) Font(size)
Framebuffer(width, height)
Image(filename) Image(file) Image(data)
ImageFont(imageOrFilename, glyphs)
ParticleSystem(image, buffer)
Quad(x, y, width, height, sw, sh)
Screenshot() -- returns ImageData
SpriteBatch(image, size)
```

Text/Shapes
```
print(text, x, y, r, sx, sy)
printf(text, x, y, limit, align)
circle(mode, x, y, radius, segments)
line(x1, y1, x2, y2, ...)
point(x, y)
polygon(mode, ...)
quad(mode, x1, y2, x2, y2, x3,y3,x4,y4)
rectangle(mode, x, y, width, height)
triangle(mode, x1, y1, x2, y2, x3, y3)
```

Transformations
```
push() pop()
translate(dx, dy)
rotate(angle)
scale(sx, sy)
```

Getters/Setters (prefix names with "get" or "set")
```
BackgroundColor(r, g, b)
Color(r, g, b, a)
Font(font) Font(file, size) Font(size)
Icon(drawable)
Mode(w, h, fullscreen, vsync, fsaa)
```

MODULE LOVE.TIMER

```
getDelta()              getFPS()
getMicroTime()          getTime()
sleep(milliseconds)
step()
```

Class Source
```
play() resume() pause() stop() rewind()
isLooping() isStatic() isStopped()
```

Getters/Setters (prefix names with "get" or "set")
```
Looping(loop)
Pitch(pitch)
Position(x, y, z)
Velocity(x, y, z)
Volume(volume)
```

Enum SourceType

`static` Decode the entire sound at once.

`stream` Decode the sound gradually.

FILESYSTEM

Module love.filesystem

```
enumerate(dir)
exists(filename)
isDirectory(filename)
isFile(filename)
iterator       = lines(filename)
chunk          = load(filename)
ok             = mkdir(name)
contents, size = read(filename, size)
ok             = remove(filename)
ok             = write(filename, data, size)
```

New Objects

```
newFile(filename)
newFileData(contents, name, decoder)
```

Getters/Setters

```
getAppdataDirectory()
modtime, errormsg = getLastModified(name)
getSaveDirectory()
getUserDirectory()
getWorkingDirectory()
setIdentity(name)
```

Class File

```
ok         = close()
eofReached = eof()
iterator   = lines()
ok         = open(mode)
contents   = read(bytes)
ok         = seek(pos)
pos        = tell()
ok         = write(data)
getSize()
```

Enum FileMode

`r` Allows you to (only) read from a file.
`w` Allows you to (only) write to a file.
`a` Same as `w` but data is appended to the end of file.

Enum FileDecoder

`file` The data is unencoded.
`base64` The data is base64-encoded.

EVENTS

Enum Event

`jp` When a joystick button is pressed.
`jr` When a joystick button is released.
`mp` When a mouse button is pressed.
`mr` When a mouse button is released.
`q` Makes the game quit.
`f` When focus to the window is gained or lost.

Module love.event

```
iterator = poll()
pump() -- low level function
push(e, a, b, c)
e, a, b, c = wait()
```

GRAPHICS CLASSES AND ENUMS

Class Font

```
getHeight()
getWidth(text)
getLineHeight()
setLineHeight(height)
```

Class Framebuffer

```
getImageData()
renderTo(function())
```

Class Image

```
min, mag = getFilter()
getHeight()
getWidth()
horiz, vert = getWrap()
setFilter(min, mag)
setWrap(horiz, vert)
```

Class Quad

```
flip(flipX, flipY)
x, y, w, h = getViewport()
setViewport(x, y, w, h)
```

Class SpriteBatch

```
add(x, y, r, sx, sy, ox, oy)
addq(quad, x, y, r, sx, sy, ox, oy)
clear()
```

Enum ColorMode

`modulate` Images (etc) are affected by the current color.
`replace` Opposite of `modulate`.

Enum DrawMode

`fill` Draw filled shape.
`line` Draw outlined shape.

Enum FilterMode

`linear` Scale image with linear interpolation.
`nearest` Nearest neighbor interpolation.

Enum LineStyle/PointStyle

`rough` Draw rough lines/points.
`smooth` Draw smooth lines/points.

Enum WrapMode

`clamp` The image appears only once.
`repeat` The image repeats itself.

Enum AlignMode

```
center
left
right
```

Enum BlendMode

```
additive
alpha
subtractive
multiplicative
```

CORE CLASSES

Class Object

```
type()
typeOf(name)
```

Class Data

```
ptr, size = getPointer()
getSize()
```

JOYSTICKS

Callbacks

```
love.joystickpressed(id, button)
love.joystickreleased(id, button)
```

Module love.joystick

```
close(id)
isDown(id, button)
isOpen(id)
open(id)
axisDir1, axisDir2, ... = getAxes(id)
direction               = getAxis(id, axis)
dx, dy                  = getBall(id, ball)
direction               = getHat(id, hat)
getName(id)
getNumAxes(id)
getNumBalls(id)
getNumButtons(id)
getNumHats(id)
getNumJoysticks()
```

Enum JoystickConstant

c	Centered.
d	Down.
l	Left.
ld	Left+Down.
lu	Left+Up.
r	Right.
rd	Right+Down.
ru	Right+Up.
u	Up.

THREADS

Module love.thread

```
getThread(name)
getThreads()
newThread(name, filenameOrFileOrData)
```

Class Thread

```
demand(msgName)
getName()
kill()
peek(msgName)
receive(msgName)
send(msgName, value)
start()
wait()
```

DATA MANIPULATION

Module love.font

```
newFontData(rasterizer)
newGlyphData(rasterizer, glyph)
newRasterizer(imageData, glyphs)
```

Module love.image

```
newEncodedImageData(imageData, format)
newImageData(width, height)
newImageData(filename)
newImageData(data)
```

Class ImageData

```
encode(format)
mapPixel(function(x, y, r, g, b, a))
paste(imageData, x, y, sx, sy, sw, sh)
getHeight()
r, g, b, a = getPixel(x, y)
getString()
getWidth()
setPixel(x, y, r, g, b, a)
```

Enum ImageFormat

bmp	BMP image format.
tga	Targa image format.

Module love.sound

```
newDecoder(fileOrFilename, buffer)
newSoundData(decoder)
newSoundData(filename)
newSoundData(samples, rate, bits,
             channels)
```

Class Decoder

```
bitSize     = getBits()
channelType = getChannels()
getSampleRate()
```

Class SoundData

```
bitSize     = getBits()
channelType = getChannels()
getSample(index)
getSampleRate()
setSample(index, sample)
```

Based on LÖVE 0.7.1, with some of the 0.7.2 API included.

Designed by Michael Ebens, 2011.
http://nova-fusion.com

Make sure...

1. Your images and framebuffers are Po2 compliant.
2. You're not creating resources or using `love.graphics.setFont(size)` every frame (i.e. in `love.update` or `love.draw`).
3. To have fun!

Key

optionalParameter

`function(...) -- pass in a function with the given arguments.`