

Love2d.js

The javascript love2d framework

By Magnus Oblerion



Link : <https://oblerion.itch.io/love2djs> ver b0.6.2 2024

1) Introduce

Description

[love2d](#) api for html5 game,
not using emscripten,
made with equivalent of lua functions written in javascript

Difference between love2d and love2d.js

- This framework work with javascript, not lua.
- `love.graphics.setFont(name)` work with name of font ("arial","ani","consola").
It can't load ttf file in this time.
- `love.window.setMode()` can't set fullscreen.
- `love.graphics.print(text,x,y,r,sx)` sx can set only font size
- some limitation due at your browsers can do and can't do

Features

- draw : text, rectangle, cercle , line
- load image, song ,video and use it
- input : keyboard, mouse and touch
- setting window and icon

!! At start of love2d.js, there is it !!

```
const LOVE2D_MOUSE=true;  
const LOVE2D_KEYBOARD=true;  
const LOVE2D_TOUCH=true;
```

true mean it load event at start, false not. So don't forget to change it.

2) API

love variable is initialised in love2d.js, it make with Love class.
it's object with all you need for use this framework.

2-1) Main function

love.load

** Initialize*

```
love.load = function()  
{  
  ... code  
}
```

** What is do*

when it is initialize , this function is call before all , once time. Perfect for load asset.

love.update

** Initialize*

```
love.update = function(dt)  
{  
  ... code  
}
```

** Parameter*

dt : delta time (time between two frame)

** What is do*

update loop function

love.draw

** Initialize*

```
love.draw = function()  
{  
  ... code  
}
```

** What is do*

draw loop function

love.keypressed

*** Initialize**

```
love.keypressed = function( key, code, repeat)
{
  ... code
}
```

*** Parameter**

- key : string key ("a","z","ArrowUp","Escape")
- code : ascii key code
- repeat : true if key is repeat, false if not

*** What is do**

this function is call when keyboard key is press

love.keyreleased

*** Initialize**

```
love.keyreleased = function(key,code)
{
  ... code
}
```

*** Parameter**

- key : string key ("a","z","ArrowUp","Escape")
- code : ascii key code

*** What is do**

this is call when key is release

2-2) Graphics

love.graphics_setColor

** Calling*

```
love.graphics_setColor(r,g,b,a);  
love.graphics_setColor(r,g,b);
```

** Parameter*

- r : red color (0 -> 255)
- b : blue color (0 -> 255)
- g : green color (0 -> 255)
- a : alpha color (0 -> 255)

** What is do*

set color for all drawing after

love.graphics_setAlpha

** Calling*

```
love.graphics_setAlpha(a);
```

** Parameter*

a : alpha color (0 → 255)

** What is do*

set alpha for all draw after

love.graphics_getColor

** Call*

```
let col = love.graphics_getColor();
```

** return*

code of currant color in hexadecimal

love.graphics_setFont

* *Call*

```
love.graphics_setFont(font);
```

* *Parameter*

- font : string name of font, not a path
-

love.graphics_setMode

* *Call*

```
love.graphics_setMode(w,h);
```

* *Parameter*

w : width of canvas

h : height of canvas

love.graphics_print

* *Call*

```
love.graphics_print(text,x,y,r,sx);
```

* *Parameter*

- text : string text
 - x : x position
 - y : y position
 - r : rotate in radiant
 - sx : size of text
-

love.graphics_rectangle

* *Call*

```
love.graphics_rectangle(mode,x,y,w,h);
```

* *parameter*

- mode : "fill" or "line" style of rectangle
- x : x position
- y : y position
- w : width
- h : height

love.graphics_circle

*** Call**

```
love.graphics_circle(mode,x,y,radius);
```

*** Parameter**

- mode : "fill" or "line" draw mode
 - x : x position of center
 - y : y position of center
 - radius : radius of circle
-

love.graphics_line

*** Call**

```
love.graphics_line(x,y,x2,y2);
```

*** Parameter**

- x : x position start
 - y : y position start
 - x2 : x position of end
 - y2 : y position of end
-

love.graphics_newImage

*** Call**

```
let img = love.graphics_newImage(filename);
```

*** Parameter**

- filename : string path of image. be aware it's can't verify path of file.

*** Return**

return at Image object.

Image object can draw with [love.graphics_draw\(\)](#)

Link : <https://oblerion.itch.io/love2djs> ver b0.6.2 2024

love.graphics_newVideo

* *Call*

```
let vid = love.graphics_newVideo(filename);
```

* *Parameter*

- filename : string path of file. be aware it can't verify path's file, use webm file .

* *Return*

return video object. can draw with [love.graphics_draw\(\)](#).

object.play() -> play video.

object.pause() -> stop video.

love.graphics_newQuad

* *Call*

```
let q = love.graphics_newQuad(x, y, w, h);
```

* *Parameter*

- x : position x of quad on image
- y : position y of quad on image
- w : width
- h : height

* *Return*

quad object

* *What is do*

object quad can use with [love.graphics_draw\(\)](#) for draw part of image (titlesheet)

love.graphics_draw

* *Call*

```
love.graphics_draw(object ,x ,y ,r);  
love.graphics_draw(object ,x ,y ,r ,sx ,sy);  
love.graphics_draw(object ,quad ,x ,y ,r ,sx ,sy);
```

* *Parameter*

- object : Image or Video object
- quad : quad object create by [love.graphics_newQuad](#)
- x : x position
- y : y position
- r : rotate in radiant (0 if not rotate)
- sx : width size on screen
- sy : height size on screen

* *What is do*

draw image object at screen or part of image object (quad)

love.graphics_scale

* *Call*

```
love.graphics_scale(sw,sh);
```

* *Parameter*

- sw : Scales the width of the current drawing (1=100%, 0.5=50%, 2=200%, etc.)
- sh : Scales the height of the current drawing (1=100%, 0.5=50%, 2=200%, etc.)

2-3) Audio

love.audio_newSource

*** Call**

```
let song = love.audio_newSource(filename,type);
```

*** Parameter**

- filename : string file path. be aware it can't verify path of file. file is (mp3 ogg wav)
- type : "static" or "streaming" , once play or loop play

*** Return**

return at Song object,

play with [love.audio_play\(object\)](#)

pause with [love.audio_pause\(object\)](#)

stop with [love.audio_stop\(object\)](#)

set volume with [love.audio_setVolume\(object\)](#)

get volume with [love.audio_getVolume\(object\)](#)

love.audio_play

*** Call**

```
love.audio_play(object);
```

*** Parameter**

- object : Song object

*** What is do**

play sound or music

love.audio_pause

*** Call**

```
love.audio_pause(object);
```

*** Parameter**

- object : Song object

*** What is do**

pause Song object source

Link : <https://oblerion.itch.io/love2djs> ver b0.6.2 2024

love.audio_stop

* *Call*

```
love.audio_stop(object);
```

* *Parameter*

- object : Song object

* *What is do*

stop Song object source

love.audio_setVolume

* *Call*

```
love.audio_setVolume(volume);
```

* *Parameter*

- volume : 0.0 (minimum) - > 1.0 (maximum)

* *What is do*

set volume of all song play after

love.audio_getVolume

* *Call*

```
let v = love.audio_getVolume();
```

* *Return*

number of currant volume

2-4) Keyboard

love.keyboard_isDown

* *Call*

```
if(love.keyboard_isDown(key)){...}
```

* *Parameter*

- key : string of key

* *Return*

true if down, false if not

* *What is do*

testing if key is down

2-5) Mouse

love.mouse_getX

* *Call*

```
let mx = love.mouse_getX();
```

* *Return*

mouse x position

love.mouse_getY

* *Call*

```
let my = love.mouse_getY();
```

* *Return*

mouse y position

Link : <https://oblerion.itch.io/love2djs> ver b0.6.2 2024

love.mouse_isDown

*** Call**

```
If( love.mouse_isDown(id) ) {...}
```

*** Parameter**

Id : 1 = left btn, 2 = right btn, 3 = middle btn

2-6) Touch

love.touch_getPosition

not working on all touch screen, use mouse input

*** Call**

```
let t = love.touch_getPosition(id);
```

*** Parameter**

id : number >= 0 , id of touche

*** Return**

t : {x,y} -> pos of touch

love.touch_getTouches

not working on all touch screen, use mouse input

*** Call**

```
let t = love.touch_getTouches();
```

*** Return**

t : array , list of id touch

2-7) Window

love.window_getWidth

** Call*

```
let w = love.window_getWidth();
```

** Return*

Return number of canvas width

love.window_getHeight

** Call*

```
let h = love.window_getHeight();
```

** Return*

Return number of canvas height

love.window_setIcon

** Call*

```
love.window_setIcon(object);
```

** Parameter*

- object : Image object create with [love.graphics.newImage\(\)](#)
-

love.window_setTitle

** Call*

```
love.window_setTitle(name) ;
```

** Parameter*

name = string title name

love.window_getTitle

** Call*

```
let title = love.window_getTitle() ;
```

2-8) System

love.system_writeSave

*** Call**

```
love.system_writeSave(name,val) ; // save val in to local browser stockage
```

*** Parameter**

name = string key

val = object need to save

love.system_readSave

*** Call**

```
let obj = love.system_readSave(name) ;
```

*** Parameter**

name = string key

*** Return**

object saved

love.system_getOs

*** Call**

```
let os = love.system_getOs() ;
```

*** Return**

« Windows », « OS X », « IOS », « UNIX », « Android », « Chrome OS »

love.system_openURL

*** Call**

```
love.system_openURL(url) ;
```

*** Parameter**

url = string url of page to open

2-9) Event

love.event_quit

** Call*

```
love.event_quit();
```

** What is do*

break main loop and draw black screen "game is stopped"

2-10) Timer

love.timer_getDelta

** Call*

```
let dt = love.timer_getDelta();
```

** Return*

dt : time between two frame

2-11) Math

math_random

** Call*

```
let rdm1 = math_random(min,max); // min → max  
let rdm2 = math_random(max); // 0 → max
```

** Parameter*

- min : minimum number
- max : maximum number

Table index

1) Introduce.....	2
Description.....	2
Difference between love2d and love2d.js.....	2
Features.....	2
2) API.....	3
2-1) Main function.....	3
love.load.....	3
* Initialize.....	3
* What is do.....	3
love.update.....	3
* Initialize.....	3
* Parameter.....	3
* What is do.....	3
love.draw.....	3
* Initialize.....	3
* What is do.....	3
love.keypressed.....	4
* Initialize.....	4
* Parameter.....	4
* What is do.....	4
love.keyreleased.....	4
* Initialize.....	4
* Parameter.....	4
* What is do.....	4
2-2) Graphics.....	5
love.graphics_setColor.....	5
* Calling.....	5
* Parameter.....	5
* What is do.....	5
love.graphics_setAlpha.....	5
* Calling.....	5
* Parameter.....	5
* What is do.....	5
love.graphics_getColor.....	5
* Call.....	5
* return.....	5
love.graphics_setFont.....	6
* Call.....	6
* Parameter.....	6
love.graphics_setMode.....	6
* Call.....	6
* Parameter.....	6
love.graphics_print.....	6
* Call.....	6
* Parameter.....	6
love.graphics_rectangle.....	6

* Call.....	6
* parameter.....	6
love.graphics_circle.....	7
* Call.....	7
* Parameter.....	7
love.graphics_line.....	7
* Call.....	7
* Parameter.....	7
love.graphics_newImage.....	7
* Call.....	7
* Parameter.....	7
* Return.....	7
love.graphics_newVideo.....	8
* Call.....	8
* Parameter.....	8
* Return.....	8
love.graphics_newQuad.....	8
* Call.....	8
* Parameter.....	8
* Return.....	8
* What is do.....	8
love.graphics_draw.....	9
* Call.....	9
* Parameter.....	9
* What is do.....	9
love.graphics_scale.....	9
* Call.....	9
* Parameter.....	9
2-3) Audio.....	10
love.audio_newSource.....	10
* Call.....	10
* Parameter.....	10
* Return.....	10
love.audio_play.....	10
* Call.....	10
* Parameter.....	10
* What is do.....	10
love.audio_pause.....	10
* Call.....	10
* Parameter.....	10
* What is do.....	10
love.audio_stop.....	11
* Call.....	11
* Parameter.....	11
* What is do.....	11
love.audio_setVolume.....	11
* Call.....	11
* Parameter.....	11

* What is do.....	11
love.audio_getVolume.....	11
* Call.....	11
* Return.....	11
2-4) Keyboard.....	12
love.keyboard_isDown.....	12
* Call.....	12
* Parameter.....	12
* Return.....	12
* What is do.....	12
2-5) Mouse.....	12
love.mouse_getX.....	12
* Call.....	12
* Return.....	12
love.mouse_getY.....	12
* Call.....	12
* Return.....	12
love.mouse_isDown.....	13
* Call.....	13
* Parameter.....	13
2-6) Touch.....	13
love.touch_getPosition.....	13
* Call.....	13
* Parameter.....	13
* Return.....	13
love.touch_getTouches.....	13
* Call.....	13
* Return.....	13
2-7) Window.....	14
love.window_getWidth.....	14
* Call.....	14
* Return.....	14
love.window_getHeight.....	14
* Call.....	14
* Return.....	14
love.window_setIcon.....	14
* Call.....	14
* Parameter.....	14
love.window_setTitle.....	14
* Call.....	14
* Parameter.....	14
love.window_getTitle.....	15
* Call.....	15
* Return.....	15
2-8) System.....	15
love.system_writeSave.....	15
* Call.....	15
* Parameter.....	15

love.system_readSave.....	15
* Call.....	15
* Parameter.....	15
* Return.....	15
love.system_getOs.....	15
* Call.....	15
* Return.....	15
love.system_openURL.....	16
* Call.....	16
* Parameter.....	16
2-9) Event.....	16
love.event_quit.....	16
* Call.....	16
* What is do.....	16
2-10) Timer.....	16
love.timer_getDelta.....	16
* Call.....	16
* Return.....	16
2-11) Math.....	16
math_random.....	16
* Call.....	16
* Parameter.....	16
Table index.....	17