Love2d Tutorial

by josefnpat

Objective

The objective of this Tutorial is to:

- Learn how to use the LÖVE framework
- Make a game in LÖVE

Assumptions

- You have love 0.7.2 or 0.8.0 installed.
  - Visit love2d.org for more information.
- You are familiar with Lua 5.1.
- You are familiar with a code editor.
- You either have game assets, or a way to make them, such as:
  - GIMP - GNU Image Manipulation Program
  - Audacity - Free Audio Editor and Recorder
- You are running Linux.
  - This tutorial is written for Linux, but can be easily followed along with other operating systems with few substitutions.

Recommendations

For this tutorial, I suggest:

- Pausing to:
  - Take notes
  - Look up documentation in the wiki
- Reviewing slides to:
  - Ensure you understand the content

Formatting

Text will be formatted like so:

- Code that was already covered.

  `Text in monospace with no formatting`

- Reader references

  `Text in monospace with red and italics`

- Code that is new

  `Text in monospace with bold`

- CLI Interaction

  `Text in monospace in italics`

Starting love

Starting your program is rather simple.

- Navigate to your directory
- Run `love .` or `love <dirname>`

Overview of `assets.zip`

```
[1942gb]$ unzip assets.zip
Archive:  assets.zip
   creating: assets/
 extracting: assets/background.gif
 extracting: assets/bullet.gif
 extracting: assets/enemy.gif
  inflating: assets/font.ttf
  inflating: assets/music.ogg
 extracting: assets/player.gif
  inflating: assets/shoot.ogg
 extracting: assets/title.gif
```

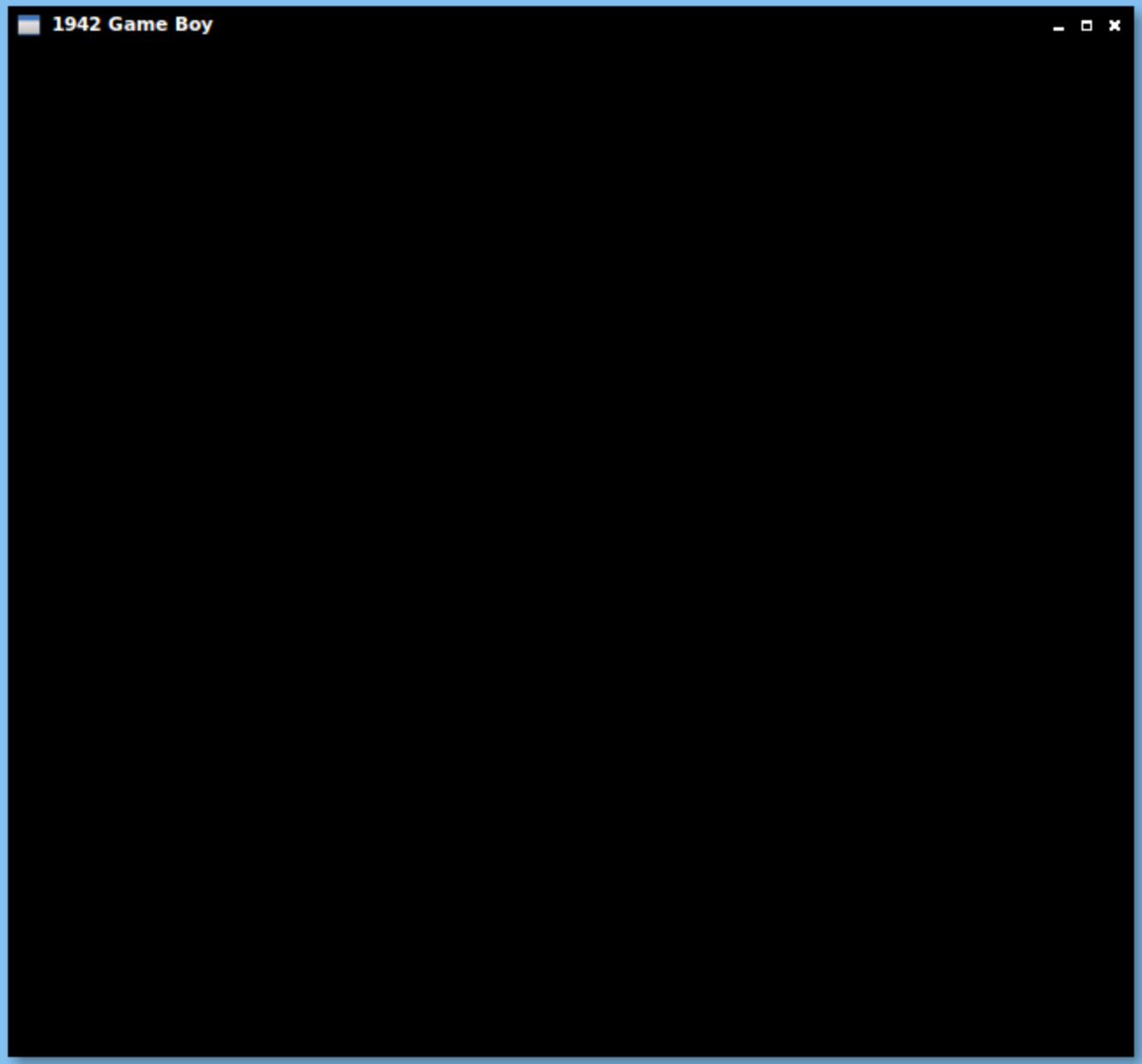
1 - config

conf.lua

```
-- Game Scale
scale = 4
function love.conf(t)
  t.title = "1942 Game Boy"
  t.screen.width = 160*scale
  t.screen.height = 144*scale
end
```

main.lua

```
debug = false

function love.load()

end

function love.draw()

end

function love.update(dt)

end

function love.keypressed(key)

end
```


1942 Game Boy


LÖVE®

main.lua:love.load()

```
function love.load()
  -- Load images (global assets)
  img_fn = {"bullet","enemy","player","title","background"}
  imgs = {}
  for _,v in ipairs(img_fn) do
    imgs[v]=love.graphics.newImage("assets/"..v..".gif")
  end

  -- Set filter to nearest
  for _,v in pairs(imgs) do
    v:setFilter("nearest","nearest")
  end

  -- Play music and loop it.
  music = love.audio.newSource( "assets/music.ogg" , "stream" )
  music:setLooping(true)
  love.audio.play(music)
  ...continued...
```

main.lua:love.load() cont.

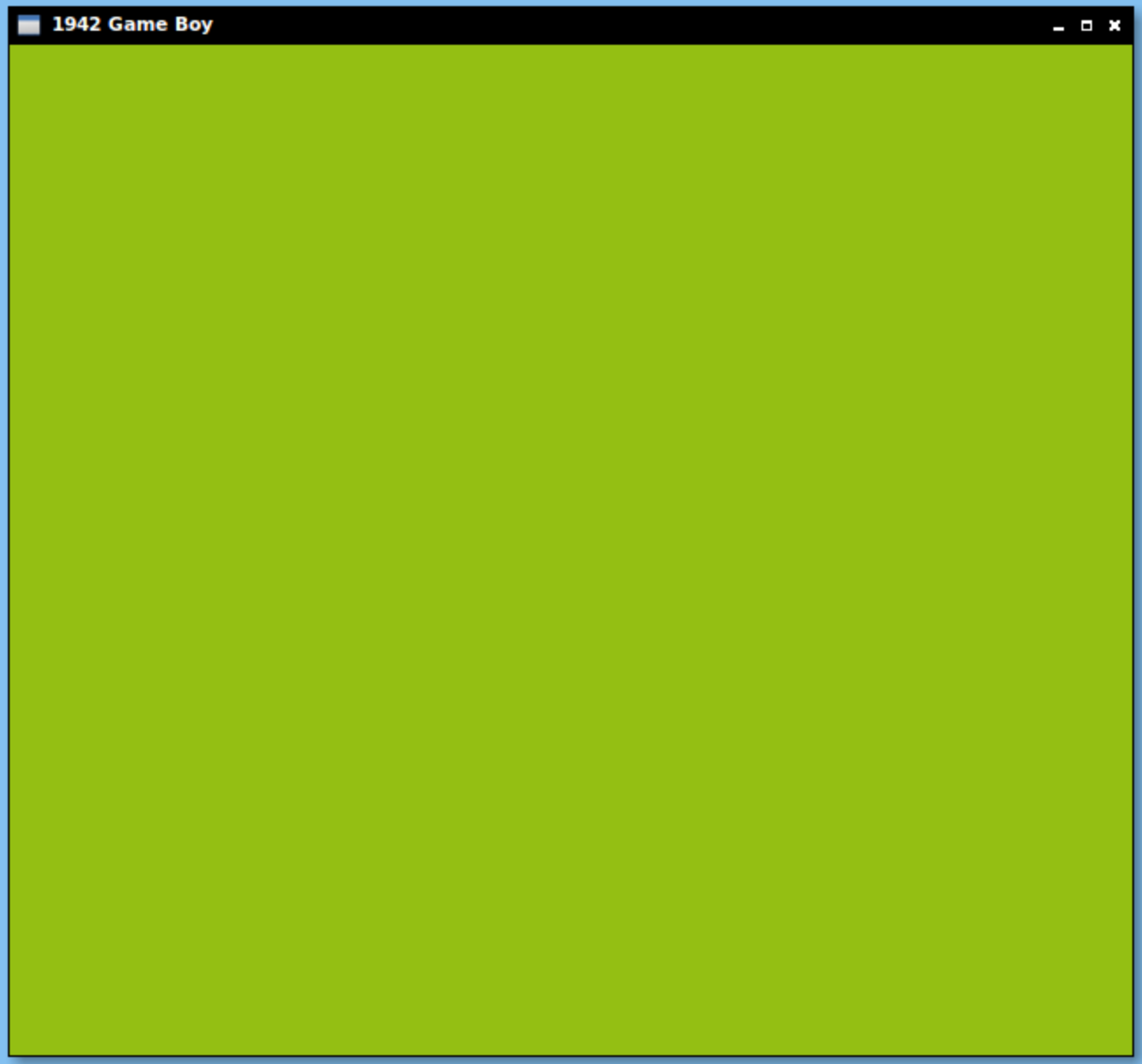
```
    -- load shoot sound
  shoot = love.audio.newSource( "assets/shoot.ogg" , "static" )

  -- Initialize font, and set it.
  font = love.graphics.newFont("assets/font.ttf",14*scale)
  love.graphics.setFont(font)

  -- define colors (global assets)
  bgcolor = {r=148,g=191,b=19}
  fontcolor = {r=46,g=115,b=46}

  -- initial state
  state = "splash"
end
```

main.lua:love.draw()

```
function love.draw()
  -- Set color
  love.graphics.setColor(bgcolor.r,bgcolor.g,bgcolor.b)
  -- Draw rectangle for background
  love.graphics.rectangle("fill",
    0,0,love.graphics.getWidth(),love.graphics.getHeight())
  -- Return the color back to normal.
  love.graphics.setColor(255,255,255)
end
```

main.lua:love.keypressed(key)

```
function love.keypressed(key)
  if key == "`" then
    debug = not debug
  end
end
```


1942 Game Boy


LÖVE®

splash.lua

```
splash = {}

function splash.load()

end

function splash.draw()

end

function splash.update(dt)

end

function splash.keypressed(key)

end
```

splash.lua:splash.load()

```
function splash.load()
  splash.dt_temp = 0
end
```

splash.lua:splash.draw()

```
function splash.draw()
  love.graphics.draw(imgs["title"],0,(splash.dt_temp-1)*32*scale,0,scale,scale)
  love.graphics.setColor(fontcolor.r,fontcolor.g,fontcolor.b)
  -- Show after 2.5 seconds
  if splash.dt_temp == 2.5 then
    love.graphics.printf("Press Start",
      0,80*scale,love.graphics.getWidth(),"center")
  end
  -- Reset the color
  love.graphics.setColor(255,255,255)
end
```

splash.lua:splash.update(dt)

```
function splash.update(dt)
  -- Update dt_temp
  splash.dt_temp = splash.dt_temp + dt
  -- Wait 2.5 seconds, then stop in place.
  if splash.dt_temp > 2.5 then
    splash.dt_temp = 2.5
  end
end
```

splash.lua:splash.keypressed(key)

```
function splash.keypressed(key)
  -- Change to game state, and init game.
  state = "game"
end
```

main.lua

```
debug = false
require('splash')
...code snip...
```

main.lua:love.load()

```
function love.load()
  ...code snip...
  --load the splash
  splash.load()
end
```

main.lua:love.draw()

```
function love.draw()
  ...code snip...
  -- Call the state's draw function
  if state == "splash" then
    splash.draw()
  end
end
```

main.lua:love.update(dt)

```
function love.update(dt)
  -- Call the state's update function
  if state == "splash" then
    splash.update(dt)
  end
end
```

main.lua:love.keypressed(key)

```
function love.keypressed(key)
  -- Call the state's keypressed function
  if state == "splash" then
    splash.keypressed(key)
  end
  if key == "`" then
    debug = not debug
  end
end
```


1942 Game Boy
1932: Game Boy
PRESS START


LÖVE®

game.lua

```
game = {}

function game.load()

end

function game.draw()

end

function game.update(dt)

end

function game.keypressed(key)

end
```

game.lua:game.load()

```
function game.load()
  -- background init
  game.clock = 0
end
```

game.lua:game.draw()

```
function game.draw()
  -- Draw moving background
  for i = 0,4 do
    for j = -1,4 do
      love.graphics.draw(imgs["background"],
                         i*32*scale,
                         (j+game.clock%1)*32*scale,
                         0,scale,scale)
    end
  end
end
```

game.lua:game.update(dt)

```
function game.update(dt)
  -- clock for background
  game.clock = game.clock + dt
end
```

splash.lua:splash.keypressed(key)

```
function splash.keypressed(key)
  -- Change to game state, and init game.
  state = "game"
  game.load()
end
```

main.lua

```
debug = false
require('splash')
require('game')
...code snip...
```

main.lua:love.load()

```
function love.load()
  ...code snip...
  splash.load()
  game.load()
end
```

main.lua:love.draw()

```
function love.draw()
  ...code snip...
  -- Call the state's draw function
  if state == "splash" then
    splash.draw()
  elseif state == "game" then
    game.draw()
  end
end
```

main.lua:love.update(dt)

```
function love.update(dt)
  -- Call the state's update function
  if state == "splash" then
    splash.update(dt)
  elseif state == "game" then
    game.update(dt)
  end
end
```

main.lua:love.keypressed(key)

```
function love.keypressed(key)
  -- Call the state's keypressed function
  if state == "splash" then
    splash.keypressed(key)
  elseif state == "game" then
    game.keypressed(key)
  end
  if key == "`" then
    debug = not debug
  end
end
```


1942 Game Boy


LÖVE®

game.lua:game.load()

```
function game.load()
  ...code snip...
  -- enemy init
  game.enemy_size = imgs["enemy"]:getWidth()
  game.enemies = {}
  game.enemy_dt = 0
  game.enemy_rate = 2
end
```

game.lua:game.draw()

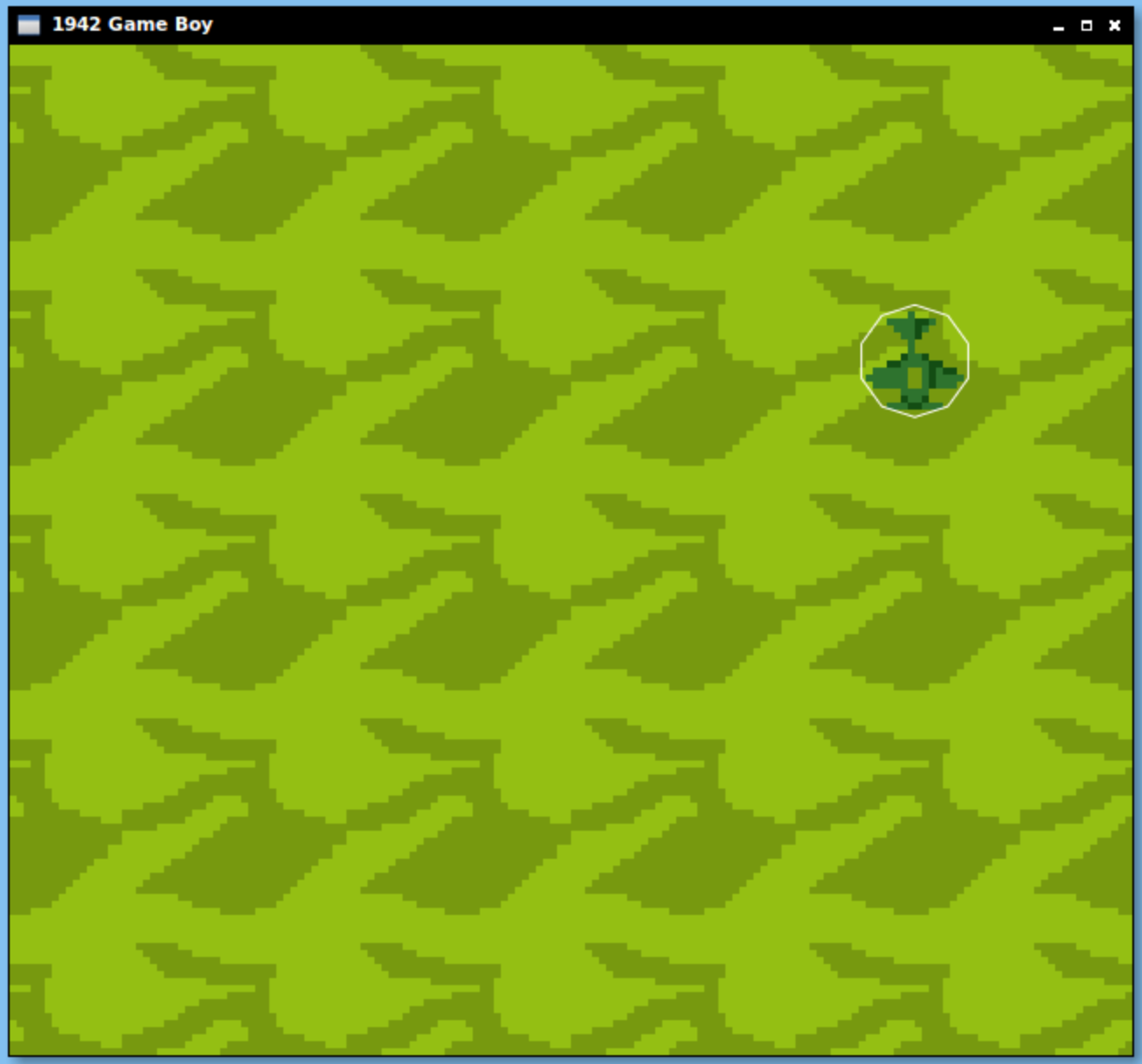
```
function game.draw()
  ...code snip...
  -- Draw enemies
  for _,v in ipairs(game.enemies) do
    love.graphics.draw(imgs["enemy"],
                       v.x,v.y,
                       0,scale,scale,
                       game.enemy_size/2,game.enemy_size/2)
    if debug then love.graphics.circle("line",v.x,v.y,game.enemy_size/2*scale) end
  end
end
```

game.lua:game.update()

```
function game.update()
  ...code snip...
  -- Update game.enemies
  game.enemy_dt = game.enemy_dt + dt

  -- Enemy spawn
  if game.enemy_dt > game.enemy_rate then
    game.enemy_dt = game.enemy_dt - game.enemy_rate
    game.enemy_rate = game.enemy_rate - 0.01 * game.enemy_rate
    local enemy = {}
    enemy.x = math.random((8)*scale,(160-8)*scale)
    enemy.y = -game.enemy_size
    table.insert(game.enemies,enemy)
  end
  ...continued...
```

game.lua:game.update() cont.

```
  -- Update enemy
  for ei,ev in ipairs(game.enemies) do
    ev.y = ev.y + 70*dt*scale
    if ev.y > 144*scale then
      table.remove(game.enemies,ei)
    end
  end
end
```


1942 Game Boy


LÖVE®

game.lua:game.load()

```
function game.load()
  ...code snip...
  -- player init
  game.player_size = imgs["player"]:getWidth()
  game.playerx = (160/2)*scale
  game.playery = (144-12)*scale
end
```

game.lua:game.draw()

```
function game.draw()
  ...code snip...
  -- Draw player
  love.graphics.draw(imgs["player"],
                     game.playerx,game.playery,
                     0,scale,scale,
                     game.player_size/2,game.player_size/2)
  if debug then
    love.graphics.circle("line",game.playerx,game.playery,game.player_size/2*scale)
  end
end
```

game.lua:game.dist(x1,y1,x2,y2)

```
-- Distance formula.
function game.dist(x1,y1,x2,y2)
  return math.sqrt( (x1 - x2)^2 + (y1 - y2)^2 )
end
```

game.lua:game.update()

```
function game.update()
  ...code snip...
  -- Update enemy
  for ei,ev in ipairs(game.enemies) do
    ev.y = ev.y + 70*dt*scale
    if ev.y > 144*scale then
      table.remove(game.enemies,ei)
    end
    -- If a player gets too close to enemy
    if game.dist(game.playerx,game.playery,ev.x,ev.y) < (12+8)*scale then
      splash.load()
      state = "splash"
    end
  end
  ...continued...
```

game.lua:game.update() cont.

```
  -- Update player movement
  if love.keyboard.isDown("right") then
    game.playerx = game.playerx + 100*dt*scale
  end
  if love.keyboard.isDown("left") then
    game.playerx = game.playerx - 100*dt*scale
  end
  -- Keep the player on the map
  if game.playerx > 160*scale then
    game.playerx = 160*scale
  end
  if game.playerx < 0 then
    game.playerx = 0
  end
end
```


1942 Game Boy


LÖVE®

game.lua:game.load()

```
function game.load()
  ...code snip...
  -- bullet init
  game.ammo = 10
  game.recharge_dt = 0
  game.recharge_rate = 1
  game.bullet_size = imgs["bullet"]:getWidth()
  game.bullets = {}
end
```

game.lua:game.draw()

```
function game.draw()
  ...code snip...
  -- Draw game.bullets
  for _,v in ipairs(game.bullets) do
    love.graphics.draw(imgs["bullet"],
                       v.x,v.y,
                       0,scale,scale,
                       game.bullet_size/2,game.bullet_size/2)
    if debug then love.graphics.circle("line",v.x,v.y,game.bullet_size/2*scale) end
  end
end
```

game.lua:game.update()

```
function game.update()
  ...code snip...
  -- Update bullets
  for bi,bv in ipairs(game.bullets) do
    bv.y = bv.y - 100*dt*scale
    if bv.y < 0 then
      table.remove(game.bullets,bi)
    end
    -- Update bullets with game.enemies
    for ei,ev in ipairs(game.enemies) do
      if game.dist(bv.x,bv.y,ev.x,ev.y) < (2+8)*scale then
        table.remove(game.enemies,ei)
        table.remove(game.bullets,bi)
      end
    end
  end
  ...continued...
```

game.lua:game.update() cont.

```
  -- Update player ammunition
  game.recharge_dt = game.recharge_dt + dt
  if game.recharge_dt > game.recharge_rate then
    game.recharge_dt = game.recharge_dt - game.recharge_rate
    game.ammo = game.ammo + 1
    if game.ammo > 10 then
      game.ammo = 10
    end
  end
end
```

game.lua:game.keypressed(key)

```
function game.keypressed(key)
  -- Shoot a bullet
  if key == " " and game.ammo > 0 then
    love.audio.play(shoot)
    game.ammo = game.ammo - 1
    local bullet = {}
    bullet.x = game.playerx
    bullet.y = game.playery
    table.insert(game.bullets,bullet)
  end
end
```


1942 Game Boy


LÖVE®

game.lua:game.load()

```
function game.load()
  ...code snip...
  -- info init
  game.score = 0
end
```

game.lua:game.draw()

```
function game.draw()
  ...code snip...
  -- Draw game info
  love.graphics.setColor(fontcolor.r,fontcolor.g,fontcolor.b)
  love.graphics.printf(
    "score:"..game.score..
    " ammo:"..game.ammo,
  0,0,love.graphics.getWidth(),"center")

  if debug then love.graphics.print(
    "enemies: "..#game.enemies..
    "\nbullets:"..#game.bullets..
    "\nenemy_rate:"..game.enemy_rate..
    "\nFPS:"..love.timer.getFPS(),
  0,14*scale) end
  love.graphics.setColor(255,255,255)
end
```

game.lua:game.update()

```
function game.update()
  ...code snip...
    -- Update bullets with game.enemies
    for ei,ev in ipairs(game.enemies) do
      if game.dist(bv.x,bv.y,ev.x,ev.y) < (2+8)*scale then
        game.score = game.score + 1
        table.remove(game.enemies,ei)
        table.remove(game.bullets,bi)
      end
    end
  end
  ...code snip...
end
```

game.lua:splash.draw()

```
function splash.draw()
  ...code snip...
  -- If previous game
  if game.score ~= 0 then
    love.graphics.printf("Score:"..game.score,0,96*scale,160*scale,"center")
  end
  -- Reset the color
  love.graphics.setColor(255,255,255)
end
```


1942 Game Boy
SCORE:1 AMMO:4
ENEMIES: 1
BULLETS:5
ENEMY_RATE:1.90198009
FPS:61


1942 Game Boy
SCORE:132 AMMO:8