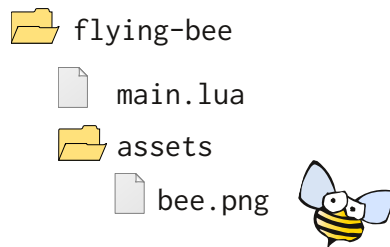


Adding assets



Assets are external files like images or sound. We will start by adding an image that will appear on screen.

Create or download a png image and put it in the (new) assets directory in your project directory.



With a text editor (like Atom) create a main.lua file that loads and displays the image:

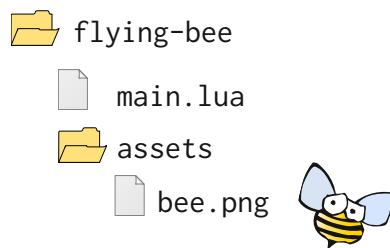
- Create a bee "table" (bee = { ... }) defining the x and y coordinates of your image and the img placeholder that will contain the image data.

Adding assets



Assets are external files like images or sound. We will start by adding an image that will appear on screen.

Create or download a png image and put it in the (new) assets directory in your project directory.



With a text editor (like Atom) create a main.lua file that loads and displays the image:

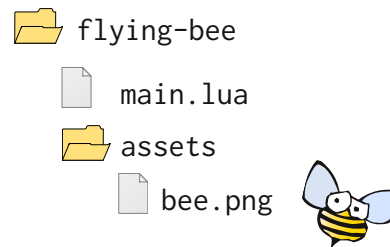
- Create a bee "table" (bee = { ... }) defining the x and y coordinates of your image and the img placeholder that will contain the image data.

Adding assets



Assets are external files like images or sound. We will start by adding an image that will appear on screen.

Create or download a png image and put it in the (new) assets directory in your project directory.



With a text editor (like Atom) create a main.lua file that loads and displays the image:

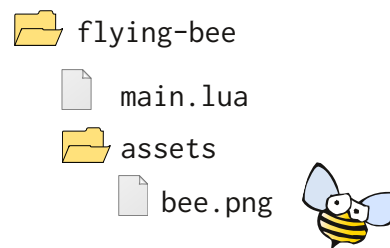
- Create a bee "table" (bee = { ... }) defining the x and y coordinates of your image and the img placeholder that will contain the image data.

Adding assets



Assets are external files like images or sound. We will start by adding an image that will appear on screen.

Create or download a png image and put it in the (new) assets directory in your project directory.



With a text editor (like Atom) create a main.lua file that loads and displays the image:

- Create a bee "table" (bee = { ... }) defining the x and y coordinates of your image and the img placeholder that will contain the image data.

- In `load()`, load the `bee.png` image (or whatever your image file is called) into `img`.
- In `draw()`, ask Löve to draw the image at the given `x` and `y` coordinates.

```
local bee = { x = 175, y = 200, img = nil}
```

```
function love.load(arg)
    bee.img = love.graphics.newImage(
        'assets/bee.png'
    )
end
```



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

- In `load()`, load the `bee.png` image (or whatever your image file is called) into `img`.
- In `draw()`, ask Löve to draw the image at the given `x` and `y` coordinates.

```
local bee = { x = 175, y = 200, img = nil}
```

```
function love.load(arg)
    bee.img = love.graphics.newImage(
        'assets/bee.png'
    )
end
```



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

- In `load()`, load the `bee.png` image (or whatever your image file is called) into `img`.
- In `draw()`, ask Löve to draw the image at the given `x` and `y` coordinates.

```
local bee = { x = 175, y = 200, img = nil}
```

```
function love.load(arg)
    bee.img = love.graphics.newImage(
        'assets/bee.png'
    )
end
```



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

- In `load()`, load the `bee.png` image (or whatever your image file is called) into `img`.
- In `draw()`, ask Löve to draw the image at the given `x` and `y` coordinates.

```
local bee = { x = 175, y = 200, img = nil}
```

```
function love.load(arg)
    bee.img = love.graphics.newImage(
        'assets/bee.png'
    )
end
```



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