Many of one kind

It's common to have many sprites of one kind. As an example, dots moving down.

First, we define a `dot` "template" that we use to create the "real" dots: while this is not techinically needed, it is a good way to document what is a `dot`.

We initialize right away the the empty `dots` table that will contain the dots.

In `update()`, each time the space bar is pressed we create a `newDot` and add it to the `dots` table. Then we go through each `dot` and move it down a bit.

In `draw()` we print each dot at its current position.

```
local dot = { x = 0, y = 0, speed = 250, size =
'5', character = 'o' }
local dots = {}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
  math.randomseed(os.time())
end
```

Many of one kind

It's common to have many sprites of one kind. As an example, dots moving down.

First, we define a `dot` "template" that we use to create the "real" dots: while this is not techinically needed, it is a good way to document what is a `dot`.

We initialize right away the the empty `dots` table that will contain the dots.

In `update()`, each time the space bar is pressed we create a `newDot` and add it to the `dots` table. Then we go through each `dot` and move it down a bit.

In `draw()` we print each dot at its current position.

```
local dot = { x = 0, y = 0, speed = 250, size =
'5', character = 'o' }
local dots = {}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
  math.randomseed(os.time())
end
```

Many of one kind

It's common to have many sprites of one kind. As an example, dots moving down.

First, we define a `dot` "template" that we use to create the "real" dots: while this is not techinically needed, it is a good way to document what is a `dot`.

We initialize right away the the empty `dots` table that will contain the dots.

In `update()`, each time the space bar is pressed we create a `newDot` and add it to the `dots` table. Then we go through each `dot` and move it down a bit.

In `draw()` we print each dot at its current position.

```
local dot = { x = 0, y = 0, speed = 250, size =
'5', character = 'o' }
local dots = {}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
  math.randomseed(os.time())
end
```

Many of one kind

It's common to have many sprites of one kind. As an example, dots moving down.

First, we define a `dot` "template" that we use to create the "real" dots: while this is not techinically needed, it is a good way to document what is a `dot`.

We initialize right away the the empty `dots` table that will contain the dots.

In `update()`, each time the space bar is pressed we create a `newDot` and add it to the `dots` table. Then we go through each `dot` and move it down a bit.

In `draw()` we print each dot at its current position.

```
local dot = { x = 0, y = 0, speed = 250, size =
'5', character = 'o' }
local dots = {}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))
  math.randomseed(os.time())
end
```

```
function love.update(dt)
  if love.keyboard.isDown('space', ' ') then
    local newDot = {
      x = math.random(0, love.graphics.getWidth() -
dot.size),
      y = 0,
      speed = dot.speed,
      character = dot.character
    }
    table.insert(dots, newDot)
  end

  for i, d in ipairs(dots) do
    d.y = d.y + (d.speed * dt)

    if dot.y > love.graphics.getHeight() then
      table.remove(dots, i)
    end
  end
end

function love.draw()
  for i, d in ipairs(dots) do
    love.graphics.print(d.character, d.x, d.y)
  end
end
```


```
function love.update(dt)
  if love.keyboard.isDown('space', ' ') then
    local newDot = {
      x = math.random(0, love.graphics.getWidth() -
dot.size),
      y = 0,
      speed = dot.speed,
      character = dot.character
    }
    table.insert(dots, newDot)
  end

  for i, d in ipairs(dots) do
    d.y = d.y + (d.speed * dt)

    if dot.y > love.graphics.getHeight() then
      table.remove(dots, i)
    end
  end
end

function love.draw()
  for i, d in ipairs(dots) do
    love.graphics.print(d.character, d.x, d.y)
  end
end
```

```
function love.update(dt)
  if love.keyboard.isDown('space', ' ') then
    local newDot = {
      x = math.random(0, love.graphics.getWidth() -
dot.size),
      y = 0,
      speed = dot.speed,
      character = dot.character
    }
    table.insert(dots, newDot)
  end

  for i, d in ipairs(dots) do
    d.y = d.y + (d.speed * dt)

    if dot.y > love.graphics.getHeight() then
      table.remove(dots, i)
    end
  end
end

function love.draw()
  for i, d in ipairs(dots) do
    love.graphics.print(d.character, d.x, d.y)
  end
end
```

```
function love.update(dt)
  if love.keyboard.isDown('space', ' ') then
    local newDot = {
      x = math.random(0, love.graphics.getWidth() -
dot.size),
      y = 0,
      speed = dot.speed,
      character = dot.character
    }
    table.insert(dots, newDot)
  end

  for i, d in ipairs(dots) do
    d.y = d.y + (d.speed * dt)

    if dot.y > love.graphics.getHeight() then
      table.remove(dots, i)
    end
  end
end

function love.draw()
  for i, d in ipairs(dots) do
    love.graphics.print(d.character, d.x, d.y)
  end
end
```