Dropping down

In this example, up to 4 small asterisks (the stars) are falling down at various speed.

A `star` has a `speed` between 200 and 400, and will appear every half to one and a half seconds.

In the `load()` function we define the size of the stars (of course, you can also load an image here!) and initialize the random function (nothing is really random in a computer, did you know it?)

The hardwork is done in update(dt).
If there are less than 4 stars, and it's time to create a new star, with a random horizontal position and a random speed (between the allowed boundaries).
And we set a random delay for the next star.
Finally, we move down all the stars in the list and remove the ones that felt out of the screen.

In `draw()`, we loop through all the stars and *draw* each of them at its current position.

```
stars = {}
star = {speed = {min = 200, max = 400}, nextDelay = 0,
delay = {min = 0.5, max = 1.5}, height = 5, char = '*'}

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


Dropping down

In this example, up to 4 small asterisks (the stars) are falling down at various speed.

A `star` has a `speed` between 200 and 400, and will appear every half to one and a half seconds.

In the `load()` function we define the size of the stars (of course, you can also load an image here!) and initialize the random function (nothing is really random in a computer, did you know it?)

The hardwork is done in update(dt).
If there are less than 4 stars, and it's time to create a new star, with a random horizontal position and a random speed (between the allowed boundaries).
And we set a random delay for the next star.
Finally, we move down all the stars in the list and remove the ones that felt out of the screen.

In `draw()`, we loop through all the stars and *draw* each of them at its current position.

```
stars = {}
star = {speed = {min = 200, max = 400}, nextDelay = 0,
delay = {min = 0.5, max = 1.5}, height = 5, char = '*'}

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


Dropping down

In this example, up to 4 small asterisks (the stars) are falling down at various speed.

A `star` has a `speed` between 200 and 400, and will appear every half to one and a half seconds.

In the `load()` function we define the size of the stars (of course, you can also load an image here!) and initialize the random function (nothing is really random in a computer, did you know it?)

The hardwork is done in update(dt).
If there are less than 4 stars, and it's time to create a new star, with a random horizontal position and a random speed (between the allowed boundaries).
And we set a random delay for the next star.
Finally, we move down all the stars in the list and remove the ones that felt out of the screen.

In `draw()`, we loop through all the stars and *draw* each of them at its current position.

```
stars = {}
star = {speed = {min = 200, max = 400}, nextDelay = 0,
delay = {min = 0.5, max = 1.5}, height = 5, char = '*'}

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


Dropping down

In this example, up to 4 small asterisks (the stars) are falling down at various speed.

A `star` has a `speed` between 200 and 400, and will appear every half to one and a half seconds.

In the `load()` function we define the size of the stars (of course, you can also load an image here!) and initialize the random function (nothing is really random in a computer, did you know it?)

The hardwork is done in update(dt).
If there are less than 4 stars, and it's time to create a new star, with a random horizontal position and a random speed (between the allowed boundaries).
And we set a random delay for the next star.
Finally, we move down all the stars in the list and remove the ones that felt out of the screen.

In `draw()`, we loop through all the stars and *draw* each of them at its current position.

```
stars = {}
star = {speed = {min = 200, max = 400}, nextDelay = 0,
delay = {min = 0.5, max = 1.5}, height = 5, char = '*'}

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

```
function love.update(dt)
  if #stars < 4 then
    if star.nextDelay > 0 then
      star.nextDelay = star.nextDelay - dt
    else
      newStar = {
        x = math.random(0, love.graphics.getWidth() -
                                            star.height),
        y = 0 - star.height,
       speed = math.random(star.speed.min, star.speed.max),
                char = star.char
            }
            table.insert(stars, newStar)
            star.nextDelay = math.random(star.delay.min,
                                          star.delay.max)
        end
    end
    for i, star in ipairs(stars) do
        star.y = star.y + (star.speed * dt)
        if star.y > love.graphics.getHeight() then
            table.remove(stars, i)
        end
    end
end

function love.draw()
    for i, star in ipairs(stars) do
        love.graphics.print(star.char, star.x, star.y)
    end
end
```


```
function love.update(dt)
  if #stars < 4 then
    if star.nextDelay > 0 then
      star.nextDelay = star.nextDelay - dt
    else
      newStar = {
        x = math.random(0, love.graphics.getWidth() -
                                            star.height),
        y = 0 - star.height,
       speed = math.random(star.speed.min, star.speed.max),
                char = star.char
            }
            table.insert(stars, newStar)
            star.nextDelay = math.random(star.delay.min,
                                          star.delay.max)
        end
    end
    for i, star in ipairs(stars) do
        star.y = star.y + (star.speed * dt)
        if star.y > love.graphics.getHeight() then
            table.remove(stars, i)
        end
    end
end

function love.draw()
    for i, star in ipairs(stars) do
        love.graphics.print(star.char, star.x, star.y)
    end
end
```



```
function love.update(dt)
  if #stars < 4 then
    if star.nextDelay > 0 then
      star.nextDelay = star.nextDelay - dt
    else
      newStar = {
        x = math.random(0, love.graphics.getWidth() -
                                            star.height),
        y = 0 - star.height,
       speed = math.random(star.speed.min, star.speed.max),
                char = star.char
            }
            table.insert(stars, newStar)
            star.nextDelay = math.random(star.delay.min,
                                          star.delay.max)
        end
    end
    for i, star in ipairs(stars) do
        star.y = star.y + (star.speed * dt)
        if star.y > love.graphics.getHeight() then
            table.remove(stars, i)
        end
    end
end

function love.draw()
    for i, star in ipairs(stars) do
        love.graphics.print(star.char, star.x, star.y)
    end
end
```



```
function love.update(dt)
  if #stars < 4 then
    if star.nextDelay > 0 then
      star.nextDelay = star.nextDelay - dt
    else
      newStar = {
        x = math.random(0, love.graphics.getWidth() -
                                            star.height),
        y = 0 - star.height,
       speed = math.random(star.speed.min, star.speed.max),
                char = star.char
            }
            table.insert(stars, newStar)
            star.nextDelay = math.random(star.delay.min,
                                          star.delay.max)
        end
    end
    for i, star in ipairs(stars) do
        star.y = star.y + (star.speed * dt)
        if star.y > love.graphics.getHeight() then
            table.remove(stars, i)
        end
    end
end

function love.draw()
    for i, star in ipairs(stars) do
        love.graphics.print(star.char, star.x, star.y)
    end
end
```