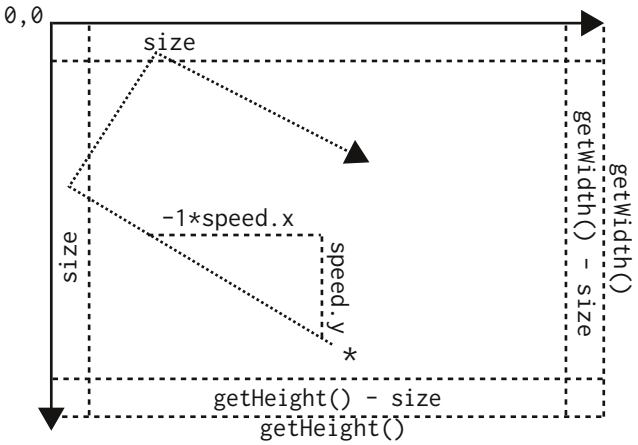
Moving randomly

The asset moves around in a random direction and bounces back at the borders.

On top of the usual fields, a star that is bouncing around has a speed and a direction for both the `x` and `y` axis.
The horizontal and vertical speeds get a random value between 200 and 300; the directions are `+1` or `-1`, making the assset go up, down, left or right.

In the `update(dt)` function:
– first, we calculate the next `x` and `y` position.
– if the next position is too close to the border of the window we invert the `x` and / or `y` direction.
– otherwise, we use the new values as the current coordinates.

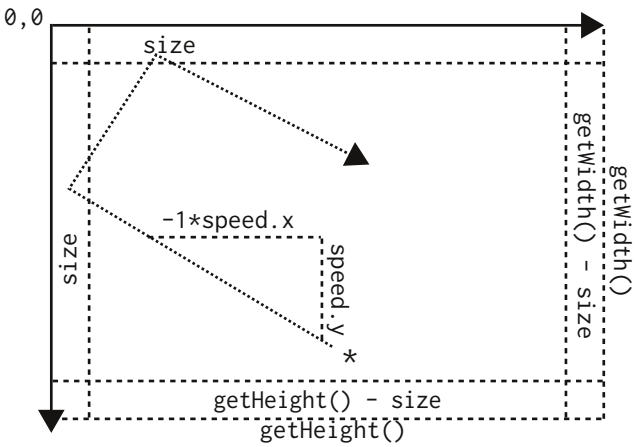


Moving randomly

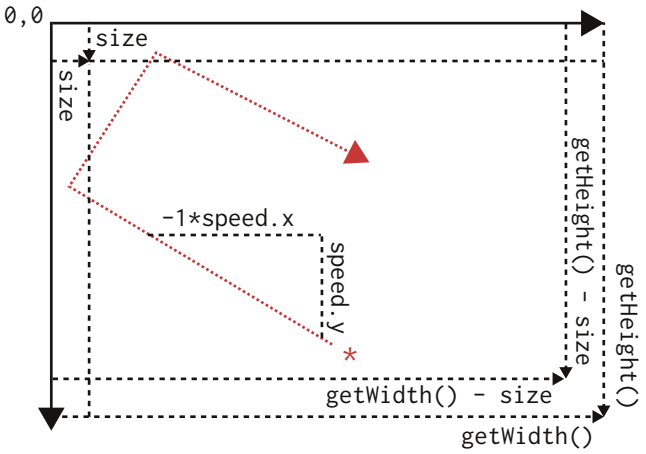
The asset moves around in a random direction and bounces back at the borders.

On top of the usual fields, a star that is bouncing around has a speed and a direction for both the `x` and `y` axis.
The horizontal and vertical speeds get a random value between 200 and 300; the directions are `+1` or `-1`, making the assset go up, down, left or right.

In the `update(dt)` function:
– first, we calculate the next `x` and `y` position.
– if the next position is too close to the border of the window we invert the `x` and / or `y` direction.
– otherwise, we use the new values as the current coordinates.

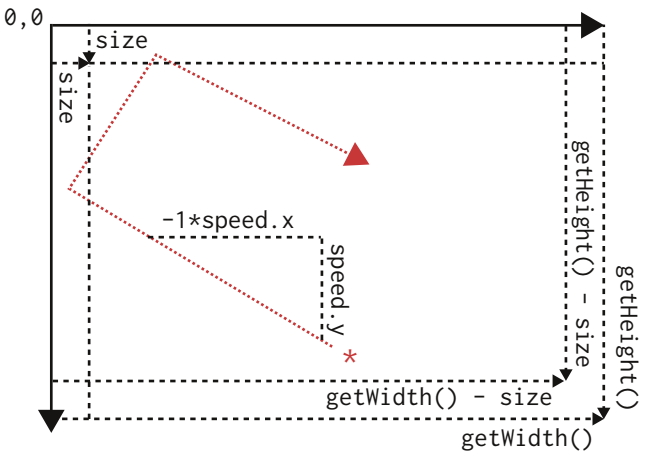


```
star = {x = 175, y = 200,
  speed = {x = 0, y = 0},
  direction =  {x = 1, y = 1},
  char = '*', size = 5}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))

  math.randomseed(os.time())
  star.speed.x = love.math.random(200, 300)
  star.speed.y = love.math.random(200, 300)
end
```




```
function love.update(dt)
  newX = star.x + (star.speed.x * dt *
star.direction.x)
  newY = star.y + (star.speed.y * dt *
star.direction.y)

  if (newX > (love.graphics.getWidth() -
star.size)) or (newX < star.size) then
    star.direction.x = star.direction.x * -1
  else
    star.x = newX
  end

  if (newY > (love.graphics.getHeight() -
star.size)) or (newY < star.size) then
    star.direction.y = star.direction.y * -1
  else
    star.y = newY
  end
end

function love.draw()
  love.graphics.print(star.char, star.x, star.y)
end
```


```
star = {x = 175, y = 200,
  speed = {x = 0, y = 0},
  direction =  {x = 1, y = 1},
  char = '*', size = 5}

function love.load(arg)
  love.graphics.setFont(love.graphics.newFont(36))

  math.randomseed(os.time())
  star.speed.x = love.math.random(200, 300)
  star.speed.y = love.math.random(200, 300)
end
```




```
function love.update(dt)
  newX = star.x + (star.speed.x * dt *
star.direction.x)
  newY = star.y + (star.speed.y * dt *
star.direction.y)

  if (newX > (love.graphics.getWidth() -
star.size)) or (newX < star.size) then
    star.direction.x = star.direction.x * -1
  else
    star.x = newX
  end

  if (newY > (love.graphics.getHeight() -
star.size)) or (newY < star.size) then
    star.direction.y = star.direction.y * -1
  else
    star.y = newY
  end
end

function love.draw()
  love.graphics.print(star.char, star.x, star.y)
end
```