import math, random

from livewires import games

games.init(screen\_width = 640, screen\_height = 480, fps = 50)

class Player(games.Sprite):

""" The player's player. """

waitTime = 50

waitCounter = 0

canShoot = True

def update(self):

""" Rotate based on left and right arrow keys. """

if games.keyboard.is\_pressed(games.K\_RIGHT):

self.angle += 1

if games.keyboard.is\_pressed(games.K\_LEFT):

self.angle -= 1

""" Move based on up and down arrow keys. """

if games.keyboard.is\_pressed(games.K\_UP) and self.y > 50:

self.y -= 3

if games.keyboard.is\_pressed(games.K\_DOWN) and self.y < 430:

self.y += 3

""" Fire missile if spacebar pressed and missile wait is over. """

if games.keyboard.is\_pressed(games.K\_SPACE) and Player.canShoot == True:

newBullet = Bullet(self.x, self.y, self.angle + 90)

games.screen.add(newBullet)

Player.canShoot = False

Player.waitCounter = 0

if Player.canShoot == False:

Player.waitCounter += 1.5

if Player.waitCounter >= Player.waitTime:

Player.canShoot = True

class Bullet(games.Sprite):

""" A bullet shot from the player's player. """

image = games.load\_image("bullet.png")

VELOCITY\_FACTOR = 7

""" Fire bullet from certain position based on the angle of Player. """

def \_\_init\_\_(self, playerX, playerY, playerAngle):

angle = playerAngle \* math.pi / 180

if playerAngle == 90:

y = playerY + 35

x = playerX + 70

elif 450 >= playerAngle > 430:

y = playerY + 25

x = playerX + 70

elif 430 >= playerAngle > 410:

y = playerY + 10

x = playerX + 70

elif 410 >= playerAngle > 390:

y = playerY - 5

x = playerX + 70

elif 390 >= playerAngle > 360:

y = playerY - 20

x = playerX + 40

if 90 < playerAngle <= 110:

y = playerY + 35

x = playerX + 65

elif 110 < playerAngle <= 130:

y = playerY + 45

x = playerX + 60

elif 130 < playerAngle <= 150:

y = playerY + 55

x = playerX + 40

elif 150 < playerAngle <= 180:

y = playerY + 50

x = playerX + 15

if 360 >= playerAngle > 180:

y = playerY + 35

x = playerX

dx = Bullet.VELOCITY\_FACTOR \* math.sin(angle)

dy = Bullet.VELOCITY\_FACTOR \* -math.cos(angle)

super(Bullet, self).\_\_init\_\_(image = Bullet.image, x = x, y = y, dx = dx, dy = dy)

def update(self):

if self.x > games.screen.width + 100:

self.destroy()

""" Destroy piranha and bullet when they collide. """

if self.overlapping\_sprites:

for sprite in self.overlapping\_sprites:

sprite.destroy()

""" Blood explosion. """

newExplosion = Explosion(x = self.x + 70, y = self.y)

games.screen.add(newExplosion)

self.destroy()

class Piranha(games.Sprite):

""" Initializes piranha sprite. """

def \_\_init\_\_(self, image, x, y, dx, dy):

super(Piranha, self).\_\_init\_\_(image = image, x = x, y = y, dx = dx, dy = dy)

def update(self):

self.x -= 1

if self.x <= 100:

self.x = 100

class Explosion(games.Animation):

""" Explosion animation. """

bloodExplosionFiles = ["bloodexplosion1.jpg",

"bloodexplosion2.jpg",

"bloodexplosion3.jpg",

"bloodexplosion4.jpg",]

def \_\_init\_\_(self, x, y):

super(Explosion, self).\_\_init\_\_(images = Explosion.bloodExplosionFiles,

x = x, y = y,

repeat\_interval = 6, n\_repeats = 1,

is\_collideable = False)

def main():

""" Establish background. """

wallImage = games.load\_image("waterbackground.jpg", transparent = False)

games.screen.background = wallImage

""" Create dock. """

dockImage = games.load\_image("dock.gif")

dock = games.Sprite(image = dockImage, x = 0, y = 240, is\_collideable = False)

games.screen.add(dock)

""" Create the player. """

playerImage = games.load\_image("playerImage.jpg")

player = Player(image = playerImage, x = 45, y = 240, is\_collideable = False)

games.screen.add(player)

""" Create 20 piranhas. """

for i in range(25):

x = random.randint(700, 3000)

y = random.randint(20, 460)

piranhaImage = games.load\_image("piranha.jpg")

piranha = Piranha(image = piranhaImage, x = x, y = y, dx = -.25, dy = 0)

games.screen.add(piranha)

games.screen.mainloop()

""" Kick it off! """

main()