from pygamemadeeasy import \*

from random import randint

import time

game = PygameMadeEasy(288, 512)

player\_x = 50 # beginning x-value for player

jump\_from\_y = 250 # beginning y-value for player

pipe\_gap\_top = randint(50,320) # y-value for the top of the gap of the first pair of pipes

pipe\_gap\_height = randint(100,200) # y-value for the size of the gap of the first pair of pipes

pipe\_x = 288 # x-value for the first pair of pipes

pipe\_speed = 7 # x-value speed per frame the pipes will move

jump\_started\_at = 0 # time code (in seconds) that the most recent jump started

previous\_y = 0 # y-value the player was at in the most recently previous frame

score = 0 # player score

point\_claimed = False # has the point for the current pipe been claimed yet?

alive = True

def click(mousex, mousey):

global pipe\_gap\_height, pipe\_gap\_top

pipe\_gap\_height = randint(100,200)

pipe\_gap\_top = randint(50,350)

def preview(ui, window):

game.image(0, 0, "media/background.png") # background.png is 288 x 512

game.image(0, 450, "media/base.png") # base.png is 336 x 112

game.set\_stroke(colors.black)

game.text(20, 150, "Flappy Bird!", size=36)

game.text(20, 200, "Press [SPACE] to start", size=18)

game.image\_piskel\_animation(50, 250, "media/flappybird-animation.png", 34, 24, rotate=30) # 32x72 pixels, 3 frames

if ui.space:

return False

def calculate\_jump\_effect( jump\_started\_at ):

""" This calculation is done using a quadratic equation (you'll learn about it in math in a couple of years). I used it to make the jump and fall effect "smoother". Don't stress about the math """

x = float(time.time() - jump\_started\_at )

y = -350\*(x\*\*2) + 350\*x

y = int(y)

return y

def keydown(key):

global jump\_from\_y, jump\_started\_at, alive

if key == " " and alive:

jump\_from\_y = jump\_from\_y - calculate\_jump\_effect( jump\_started\_at )

jump\_started\_at = time.time()

game.sound("media/wing.wav")

def main(ui, window):

global player\_x, jump\_from\_y, previous\_y, score, alive

global pipe\_gap\_height, pipe\_gap\_top, pipe\_x, point\_claimed

# background

game.image(0,0,"media/background.png") # background.png is 288 x 512

game.image(0,450,"media/base.png") # base.png is 336 x 112

# pipes

pipe\_x = pipe\_x - pipe\_speed # move the pipe for this frame

box1 = game.image(pipe\_x,pipe\_gap\_top-320,"media/pipe.png", rotate=180) # top pipe

box2 = game.image(pipe\_x,pipe\_gap\_top+pipe\_gap\_height,"media/pipe.png") # bottom pipe

if pipe\_x < (player\_x - 52) and not point\_claimed: # player has cleared the pipe

score = score + 1

game.sound("media/point.wav")

point\_claimed = True

if pipe\_x < -50: # pipe has disappeared off screen, move it back to the otherside to reappear

pipe\_x = 288+randint(0,200)

pipe\_gap\_height = randint(100,200)

pipe\_gap\_top = randint(50,300)

point\_claimed = False # "new" pipe so new point to be claimed

# player

draw\_y = jump\_from\_y - calculate\_jump\_effect(jump\_started\_at)

if draw\_y < previous\_y:

angle = 30

else:

angle = -30

players\_box = game.image\_piskel\_animation(player\_x,draw\_y,"media/flappybird-animation.png",34,24, rotate=angle) # 32x72 pixels, 3 frames

previous\_y = draw\_y

# check for collision with pipe

if alive:

if game.is\_collision(players\_box, box1) or game.is\_collision(players\_box, box2):

alive = False

game.sound("media/hit.wav")

game.sound("media/die.wav")

# check if we fell to the ground

if draw\_y >= 450:

return False

def gameover(ui, window):

global score

game.image(0, 0, "media/background.png") # background.png is 288 x 512

game.image(0, 450, "media/base.png") # base.png is 336 x 112

game.text(20, 150, "Game over!", size=36)

game.text(20, 200, "Your score:", size=18)

game.text(20, 250, str(score), size=72)

game.text(20, 400, "Press [ESC] to quit", size=18)

game.play(preview)

jump\_started\_at = time.time()

game.play(main, keydown=keydown)

game.play(gameover)

|  |  |  |
| --- | --- | --- |
| ui.x = 0  ui.y = 0  ui.click = False  ui.up = False  ui.down = False  ui.left = False  ui.right = False  ui.space = False  ui.ctrl = False  ui.alt = False  ui.shift = False  ui.enter = False  ui.keyspressed = [] | colors.white = (0xFF, 0xFF, 0xFF)  colors.silver = (0xC0, 0xC0, 0xC0)  colors.gray = (0x80, 0x80, 0x80)  colors.black = (0x00, 0x00, 0x00)  colors.red = (0xFF, 0x00, 0x00)  colors.maroon = (0x80, 0x00, 0x00)  colors.yellow = (0xFF, 0xFF, 0x00)  colors.olive = (0x80, 0x80, 0x00)  colors.lime = (0x00, 0xFF, 0x00)  colors.green = (0x00, 0x80, 0x00)  colors.aqua = (0x00, 0xFF, 0xFF)  colors.teal = (0x00, 0x80, 0x80)  colors.blue = (0x00, 0x00, 0xFF)  colors.navy = (0x00, 0x00, 0x80)  colors.fuchsia = (0xFF, 0x00, 0xFF)  colors.purple = (0x80, 0x00, 0x80) | game = PygameMadeEasy(width, height)  game.set\_fill( color )  game.set\_stroke( color )  game.set\_background( color )  game.circle( x, y, radius )  game.line( x1, y1, x2, y2 )  game.rectangle( x, y, x2, y2 )  game.rectangle( x, y, w=, h= )  game.image( x, y, filename, rotate= )  game.image\_piskel\_animation( x, y, filename, cell\_width, cell\_height, rotate= )  game.text( x, y, message, size=, font= )  game.music( mp3\_filename )  game.sound( wav\_filename )  game.get\_pixel( x, y )  game.get\_distance\_between\_points( x1, y1, x2, y2 )  game.is\_collision( box1, box2 )  game.play( loop=, keydown=, keyup=, mousemotion=, mousebutton= ) |