# pygame template - skeleton for new games

import pygame, sys

import random

import os

game = False

menu = True

ships = False

shipSpeed = 1

game\_folder = os.path.dirname(\_\_file\_\_)

img\_folder = os.path.join(game\_folder, 'img')

font\_folder = os.path.join(game\_folder, 'font')

snd\_folder = os.path.join(game\_folder, 'snd')

WIDTH = 800

HEIGHT = 800

#pygame stuff

pygame.init()

pygame.mixer.init()

SCREEN = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("SHMUP")

clock = pygame.time.Clock()

backy = WIDTH/2

FPS = 30

font = pygame.font.Font(os.path.join(font\_folder,'font.ttf'), 30)

background = pygame.image.load(os.path.join(img\_folder,'assets/Background/space.png'))

background\_rect = background.get\_rect()

background\_rect.center = (WIDTH/2,backy)

background1 = pygame.image.load(os.path.join(img\_folder,'assets/Background/space.png'))

background\_rect1 = background1.get\_rect()

SHIP = pygame.image.load(os.path.join(img\_folder,'body.png'))

WHITE = (255,255,255)

BLACK = (0,0,0)

RED = (255,0,0)

GREEN = (0,255,0)

BLUE = (0,0,255)

class Player(pygame.sprite.Sprite):

def \_\_init\_\_(self):

pygame.sprite.Sprite.\_\_init\_\_(self)

self.image = SHIP

self.rect = self.image.get\_rect()

self.rect.centerx = WIDTH / 2

self.rect.bottom = HEIGHT - 10

self.speedx = 0

self.speedy = 0

def update(self):

self.speedx = 0

keystate = pygame.key.get\_pressed()

if keystate[pygame.K\_LEFT]:

self.rect.x += -shipSpeed

if keystate[pygame.K\_RIGHT]:

self.rect.x += shipSpeed

if self.rect.right > WIDTH:

self.rect.right = WIDTH

if self.rect.left <0:

self.rect.left = 0

all\_sprites = pygame.sprite.Group()

player = Player()

all\_sprites.add(player)

gameStart = font.render('Play', True, WHITE)

gameButton = gameStart.get\_rect()

gameButton.center = (WIDTH/2,440)

shipSel = font.render('Ship',True, WHITE)

shipButton = shipSel.get\_rect()

shipButton.center = (WIDTH/2,500)

back = font.render('Back', True, WHITE)

backButton = back.get\_rect()

backButton.center = (55,HEIGHT - 30)

running = True #Game loop

while running:

clock.tick(FPS)

while menu:

SCREEN.fill(BLACK)

background\_rect.center = (WIDTH/2,backy)

background\_rect1.center = (WIDTH/2,backy - 500)

SCREEN.blit(background, background\_rect)

SCREEN.blit(background1, background\_rect1)

SCREEN.blit(gameStart, gameButton)

SCREEN.blit(shipSel, shipButton)

backy += 1

if backy > HEIGHT:

backy = WIDTH/2

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

print("ThisWorks")

if gameButton.collidepoint(pygame.mouse.get\_pos()):

game = True

menu = False

ship = False

if shipButton.collidepoint(pygame.mouse.get\_pos()):

print("Works")

ship = True

menu = False

game = False

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

running = False

all\_sprites.update()

pygame.display.flip()

while ship:

SCREEN.fill(BLACK)

SCREEN.blit(back, backButton)

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

if backButton.collidepoint(pygame.mouse.get\_pos()):

menu = True

game = False

ship = False

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

running = False

all\_sprites.update()

pygame.display.flip()

while game:

SCREEN.fill(BLACK)

all\_sprites.draw(SCREEN)

all\_sprites.update()

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

running = False

all\_sprites.update()

pygame.display.flip()