**game.py**

# Imports

import sys

from scenes import \*

# Main game class

class Game:

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

self.active\_scene = TitleScene()

self.clock = pygame.time.Clock()

def is\_quit\_event(self, event, pressed\_keys):

x\_out = event.type == pygame.QUIT

ctrl = pressed\_keys[pygame.K\_LCTRL] or pressed\_keys[pygame.K\_RCTRL]

q = pressed\_keys[pygame.K\_q]

return x\_out or (ctrl and q)

def run(self):

while self.active\_scene is not None:

# Get user input

pressed\_keys = pygame.key.get\_pressed()

filtered\_events = []

for event in pygame.event.get():

if self.is\_quit\_event(event, pressed\_keys):

self.active\_scene.terminate()

else:

filtered\_events.append(event)

# Manage scene

self.active\_scene.process\_input(filtered\_events, pressed\_keys)

self.active\_scene.update()

self.active\_scene.render()

self.active\_scene = self.active\_scene.next\_scene

# Update and tick

pygame.display.flip()

self.clock.tick(FPS)

self.end()

def end(self):

pygame.quit()

sys.exit()

# Let's do this!

if \_\_name\_\_ == "\_\_main\_\_":

g = Game()

g.run()

**config.py**

import pygame

# Window settings

SCREEN\_WIDTH = 800

SCREEN\_HEIGHT = 600

TITLE = "Name of Game"

FPS = 60

# Make window

pygame.init()

screen = pygame.display.set\_mode([SCREEN\_WIDTH, SCREEN\_HEIGHT])

pygame.display.set\_caption(TITLE)

# Load assets

''' Colors '''

BLACK = (0, 0, 0)

WHITE = (255, 255, 255)

''' Fonts '''

font\_sm = pygame.font.Font(None, 24)

font\_md = pygame.font.Font(None, 32)

font\_lg = pygame.font.Font(None, 64)

font\_xl = pygame.font.Font(None, 96)

''' Images '''

ship\_img = pygame.image.load('assets/images/playerShip1\_blue.png').convert\_alpha()

laser\_img = pygame.image.load('assets/images/laserBlue03.png').convert\_alpha()

mob\_img = pygame.image.load('assets/images/enemyRed1.png').convert\_alpha()

''' Music '''

''' Sounds '''

laser\_snd = pygame.mixer.Sound('assets/sounds/sfx\_laser1.ogg')

# Game settings

''' Controls '''

p1\_controls = {'left': pygame.K\_LEFT,

'right': pygame.K\_RIGHT,

'shoot': pygame.K\_SPACE}

**scenes.py**

# Imports

from config import \*

from sprites import \*

from tools import \*

# Scenes

class Scene:

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

self.next\_scene = self

def process\_input(self, events, pressed\_keys):

raise NotImplementedError

def update(self):

raise NotImplementedError

def render(self):

raise NotImplementedError

def terminate(self):

self.next\_scene = None

class TitleScene(Scene):

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

super().\_\_init\_\_()

def process\_input(self, events, pressed\_keys):

for event in events:

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_SPACE:

self.next\_scene = PlayScene()

def update(self):

pass

def render(self):

screen.fill(BLACK)

draw\_text(screen, TITLE, font\_xl, WHITE, [SCREEN\_WIDTH // 2, SCREEN\_HEIGHT // 2], 'center')

class PlayScene(Scene):

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

super().\_\_init\_\_()

self.player = pygame.sprite.Group()

self.lasers = pygame.sprite.Group()

self.mobs = pygame.sprite.Group()

self.ship = Ship(ship\_img, [SCREEN\_WIDTH // 2, SCREEN\_HEIGHT - 75])

self.player.add(self.ship)

self.level = 1

self.start\_level()

def start\_level(self):

if self.level == 1:

mob\_locs =[[100, 100], [250, 100], [400, 100]]

elif self.level == 2:

mob\_locs =[[100, 100], [250, 100], [400, 100]]

elif self.level == 3:

mob\_locs =[[100, 100], [250, 100], [400, 100]]

for loc in mob\_locs:

self.mobs.add( Mob(mob\_img, loc) )

def process\_input(self, events, pressed\_keys):

for event in events:

if event.type == pygame.KEYDOWN:

if event.key == p1\_controls['shoot']:

self.ship.shoot(self.lasers)

if pressed\_keys[p1\_controls['left']]:

self.ship.move\_left()

elif pressed\_keys[p1\_controls['right']]:

self.ship.move\_right()

def update(self):

self.lasers.update()

self.mobs.update(self.lasers)

if len(self.mobs) == 0:

if self.level < 3:

self.level += 1

self.start\_level()

else:

self.next\_scene = EndScene()

def render(self):

screen.fill(BLACK)

self.player.draw(screen)

self.lasers.draw(screen)

self.mobs.draw(screen)

class EndScene(Scene):

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

super().\_\_init\_\_()

def process\_input(self, events, pressed\_keys):

for event in events:

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_SPACE:

self.next\_scene = TitleScene()

def update(self):

pass

def render(self):

screen.fill(BLACK)

draw\_text(screen, "Game Over", font\_lg, WHITE,

[SCREEN\_WIDTH // 2, SCREEN\_HEIGHT // 2], 'center')

**sprites.py**

import pygame

import random

from config import \*

class Ship(pygame.sprite.Sprite):

def \_\_init\_\_(self, image, location):

super().\_\_init\_\_()

self.image = image

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

self.rect.center = location

def move\_left(self):

self.rect.x -= 5

if self.rect.left < 0:

self.rect.left = 0

def move\_right(self):

self.rect.x += 5

if self.rect.right > SCREEN\_WIDTH:

self.rect.right = SCREEN\_WIDTH

def shoot(self, lasers):

laser = Laser(laser\_img, [self.rect.centerx, self.rect.top])

lasers.add(laser)

laser\_snd.play()

print("Pew!")

def update(self):

pass

class Laser(pygame.sprite.Sprite):

def \_\_init\_\_(self, image, location):

super().\_\_init\_\_()

self.image = image

self.mask = pygame.mask.from\_surface(self.image)

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

self.rect.center = location

def update(self):

self.rect.y -= 8

if self.rect.bottom < 0:

self.kill()

class Mob(pygame.sprite.Sprite):

def \_\_init\_\_(self, image, location):

super().\_\_init\_\_()

self.image = image

self.mask = pygame.mask.from\_surface(self.image)

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

self.rect.center = location

self.health = 3

def drop\_bomb(self):

print("Bwap!")

def update(self, lasers):

hit\_list = pygame.sprite.spritecollide(self, lasers, True,

pygame.sprite.collide\_mask)

for hit in hit\_list:

self.health -= 1

if self.health <= 0:

self.kill()