import pygame  
import math  
import time  
  
import pygame.event  
  
  
# FUNCTION FOR SCALING IMAGES  
def scale(img, factor):  
 size = round(img.get\_width() \* factor), round(img.get\_height() \* factor)  
 return pygame.transform.scale(img, size)  
  
  
# IMPORTING IMAGES NEEDED  
F\_MAZE = pygame.image.load('Assets/PACMAN MAZE 1.png')  
F\_MAZE\_MASK = pygame.mask.from\_surface(F\_MAZE)  
PACMAN = pygame.image.load('Assets/Pacman circle.png')  
RED\_GHOST = scale(pygame.image.load('Assets/red ghost.png'), 0.035)  
BLUE\_GHOST = scale(pygame.image.load('Assets/blue ghost.png'), 0.035)  
YELLOW\_GHOST = scale(pygame.image.load('Assets/Yellow ghost.png'), 0.04)  
# TILE = pygame.image.load('Assets/Square2.png')  
# TILE = pygame.transform.scale(TILE, (20, 20))  
BLANK = pygame.image.load('Assets/black.png')  
COIN = pygame.image.load('Assets/coin.png')  
COIN = pygame.transform.scale(COIN, (5, 5))  
PACMAN = pygame.transform.scale(PACMAN, (18, 18))  
# SETTING UP MAIN WINDOW  
  
WIDTH, HEIGHT = F\_MAZE.get\_width(), F\_MAZE.get\_height()  
  
WIN = pygame.display.set\_mode((WIDTH, HEIGHT))  
pygame.display.set\_caption('PACMAN - MUNEEBA SE-071')  
icon = pygame.image.load('Assets/PACMAN ICON.png')  
pygame.display.set\_icon(icon)  
score = 0  
# FUNCTION FOR DRAWING  
  
def draw(win, images):  
 WIN.blit(F\_MAZE, (0, 0))  
 WIN.blit(PACMAN, (start\_x, start\_y))  
  
 for img, pos in images:  
 win.blit(img, pos)  
  
  
def collide\_maze(mask, x=0, y=0):  
 global start\_x, start\_y  
 Pacman\_mask = pygame.mask.from\_surface(PACMAN)  
 offset = (int(start\_x - x), int(start\_y - y))  
 poi = mask.overlap(Pacman\_mask, offset)  
 return poi  
  
  
########################### MAIN PROGRAM ##########################  
run = True  
FPS = 60  
clock = pygame.time.Clock()  
start\_x, start\_y = 20, 20  
  
images = [(RED\_GHOST, (20, 450)), (BLUE\_GHOST, (140, 60)), (YELLOW\_GHOST, (520, 150))]  
GRAY = (107, 107, 107)  
CELL\_WIDTH = WIDTH // 28  
print(CELL\_WIDTH)  
CELL\_HEIGHT = HEIGHT // 30  
print(CELL\_HEIGHT)  
  
vel = 1  
# SETTING MAZE  
MAZE = [  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [5, 5, 5, 5, 5, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 5, 5, 5, 5, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0],  
 [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],  
 [0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0],  
 [0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0],  
 [0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0],  
 [0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0],  
 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],  
 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],  
 [0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
]  
TILESIZE = 17  
# INFINITE EVENT LOOP  
while run:  
 clock.tick(FPS)  
  
 # DRAW CHARACTERS ON BOARD  
 WIN.fill((0, 0, 0))  
 draw(WIN, images)  
 pygame.display.update()  
  
 # draw coins  
 coin\_rect = []  
 for row\_index, row in enumerate(MAZE):  
 for col\_index, col in enumerate(row):  
 x = col\_index \* TILESIZE  
 y = row\_index \* TILESIZE  
 if col == 1:  
 WIN.blit(COIN, (x, y))  
 coin\_rect.append(pygame.Rect(x, y, 5, 5))  
 pygame.display.update()  
  
 # check collision of maze  
 if collide\_maze(F\_MAZE\_MASK) is not None:  
 print("collide")  
 if direction == 'LEFT':  
 start\_x += 1  
  
 if direction == 'RIGHT':  
 start\_x -= 1  
  
 if direction == 'UP':  
 start\_y += 1  
  
 if direction == 'DOWN':  
 start\_y -= 1  
  
 # check coin collection  
 Pacman\_rect = pygame.Rect(start\_x, start\_y, 18, 18)  
 for c in coin\_rect:  
 if c.colliderect(Pacman\_rect):  
 coin\_x = c.left  
 coin\_y = c.top  
 index\_column = coin\_x // TILESIZE  
 index\_row = coin\_y // TILESIZE  
 MAZE[index\_row][index\_column] = 0  
 coin\_rect.remove(c)  
 score += 1  
 print(score)  
  
 # PACMAN movement  
 keys = pygame.key.get\_pressed()  
 if keys[pygame.K\_LEFT]:  
 start\_x -= vel  
  
 direction = 'LEFT'  
  
 elif keys[pygame.K\_RIGHT]:  
 start\_x += vel  
 direction = 'RIGHT'  
  
 elif keys[pygame.K\_UP]:  
 start\_y -= vel  
 direction = 'UP'  
  
 elif keys[pygame.K\_DOWN]:  
 start\_y += vel  
 direction = 'DOWN'  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 run = False  
 break  
  
pygame.quit()