import pygame  
import random  
import sys  
import os  
import time  
  
########################################################################################################################  
  
  
def file\_filler():  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for \_ in range(125):  
 for \_ in range(125):  
 print(f.write(random.choice('10')))  
 print(f.write('\n'))  
 f.close()  
  
  
def level\_smoother():  
 for \_ in range(5):  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 li1 = []  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 counter\_floar = 0  
 counter\_wall = 0  
 if x != 0 and y != 0 and x != len(li) - 1 and y != len(li) - 1:  
 if li[x - 1][y - 1] == '0':  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x - 1][y] == '0':  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x - 1][y + 1] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x][y - 1] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x][y] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x][y + 1] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x + 1][y - 1] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x + 1][y] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if li[x + 1][y + 1] == "0":  
 counter\_floar += 1  
 else:  
 counter\_wall += 1  
  
 if counter\_wall <= counter\_floar:  
 li1[-1].append(0)  
 else:  
 li1[-1].append(1)  
 else:  
 li1[-1].append(1)  
 print(li1)  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
  
  
def rocks\_creator():  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 li1 = []  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 if li[x][y] == "1":  
 li1[-1].append("1")  
 else:  
 li1[-1].append(random.choice('200000000000000000'))  
 print(li1)  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
 print(li1)  
  
  
def ors\_creator():  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 li1 = []  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 if li[x][y] == "1":  
 li1[-1].append("1")  
 elif li[x][y] == "2":  
 li1[-1].append("2")  
 else:  
 li1[-1].append(random.choice('3000000000'))  
 print(li1)  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
 print(li1)  
  
  
def player\_creator():  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 li1 = []  
 r = 0  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 if li[x][y] == "1":  
 li1[-1].append("1")  
 elif li[x][y] == "2":  
 li1[-1].append("2")  
 elif li[x][y] == "3":  
 li1[-1].append("3")  
 else:  
 if r != "4":  
 r = '4'  
 li1[-1].append(r)  
 else:  
 li1[-1].append('0')  
 print(li1)  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
 print(li1)  
  
  
def ex\_creator():  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 li.reverse()  
 for i in range(len(li)):  
 li[i] = li[i][::-1]  
 li1 = []  
 r = 0  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 if r != "5" and x == 1 and y == 1:  
 r = '5'  
 li1[-1].append(r)  
 else:  
 if li[x][y] == "1":  
 li1[-1].append("1")  
 elif li[x][y] == "2":  
 li1[-1].append("2")  
 elif li[x][y] == "3":  
 li1[-1].append("3")  
 elif li[x][y] == "4":  
 li1[-1].append("4")  
 else:  
 li1[-1].append('0')  
 print(li1)  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
 print(li1)  
  
  
def ways\_creator():  
 f = open("E:/pythonProject1/уровень.txt", 'r')  
 k = f.read()  
 f.close()  
 li = k.split()  
 player = []  
 ex = []  
 li1 = []  
 for x in range(0, len(li)):  
 for y in range(0, len(li)):  
 if li[x][y] == "4":  
 player = [x, y]  
 elif li[x][y] == "5":  
 ex = [x, y]  
 for x in range(0, len(li)):  
 li1.append([])  
 for y in range(0, len(li)):  
 if li[x][y] == "1":  
 li1[-1].append("1")  
 elif li[x][y] == "2":  
 li1[-1].append("2")  
 elif li[x][y] == "3":  
 li1[-1].append("3")  
 elif li[x][y] == "4":  
 li1[-1].append("4")  
 elif li[x][y] == "5":  
 li1[-1].append("5")  
 else:  
 li1[-1].append('0')  
 xes = int(player[0]) - int(ex[0])  
 yes = int(player[1]) - int(ex[1])  
 way\_maker = ""  
 for i in range(xes - 1):  
 way\_maker += "r"  
 for j in range(yes - 1):  
 way\_maker += "d"  
 way\_maker = random.sample(way\_maker, len(way\_maker))  
 print(player, ex, way\_maker)  
 for i in range(len(way\_maker)):  
 if way\_maker[i] == "d":  
 li1[int(ex[0])][int(ex[1]) + 1] = 0  
 ex[1] = int(ex[1]) + 1  
 else:  
 li1[int(ex[0]) + 1][int(ex[1])] = 0  
 ex[0] = int(ex[0]) + 1  
 f = open("E:/pythonProject1/уровень.txt", 'w')  
 for i in li1:  
 for j in i:  
 print(f.write(str(j)))  
 print(f.write('\n'))  
 f.close()  
 print(player, ex, way\_maker)  
  
  
file\_filler()  
level\_smoother()  
rocks\_creator()  
ors\_creator()  
player\_creator()  
ex\_creator()  
ways\_creator()  
  
########################################################################################################################  
  
  
def load\_image(name, color\_key=None):  
 fullname = os.path.join('E:/pythonProject1/', name)  
 try:  
 image = pygame.image.load(fullname)  
 except pygame.error as message:  
 print('Не удаётся загрузить:', name)  
 raise SystemExit(message)  
 image = image.convert\_alpha()  
 if color\_key is not None:  
 if color\_key == -1:  
 color\_key = image.get\_at((0, 0))  
 image.set\_colorkey(color\_key)  
 return image  
  
  
pygame.init()  
size = width, height = 800, 400  
screen = pygame.display.set\_mode(size)  
sprite\_group = pygame.sprite.Group()  
hero\_group = pygame.sprite.Group()  
  
tile\_images = {  
 'wall': load\_image('E:/pythonProject1/stone.png'),  
 'dirt': load\_image('E:/pythonProject1/dirt.png'),  
 'ore': load\_image('E:/pythonProject1/ore.png'),  
 'stone': load\_image('E:/pythonProject1/small\_stone.png'),  
 'hole': load\_image('E:/pythonProject1/hole.png')  
}  
player\_image = load\_image('E:/pythonProject1/mario.png')  
  
tile\_width = tile\_height = 50  
  
  
class ScreenFrame(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.rect = (0, 0, 800, 400)  
  
  
class SpriteGroup(pygame.sprite.Sprite):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
  
 def get\_event(self, event):  
 for inet in self:  
 inet.get\_event(event)  
  
  
class Sprite(pygame.sprite.Sprite):  
 def \_\_init\_\_(self, group):  
 super().\_\_init\_\_(group)  
 self.rect = None  
  
 def get\_event(self, event):  
 pass  
  
  
class Tile(Sprite):  
 def \_\_init\_\_(self, tile\_type, pos\_x, pos\_y):  
 super().\_\_init\_\_(sprite\_group)  
 self.image = tile\_images[tile\_type]  
 self.rect = self.image.get\_rect().move(tile\_width \* pos\_x, tile\_height \* pos\_y)  
  
  
class Player(Sprite):  
 def \_\_init\_\_(self, pos\_x, pos\_y):  
 super().\_\_init\_\_(hero\_group)  
 self.image = player\_image  
 self.rect = self.image.get\_rect().move(tile\_width \* pos\_x + 15, tile\_height \* pos\_y + 5)  
 self.pos = (pos\_x, pos\_y)  
  
 def move(self, x, y):  
 self.pos = (x, y)  
 self.rect = self.image.get\_rect().move(tile\_width \* self.pos[0] + 15,  
 tile\_height \* self.pos[1] + 5)  
  
  
class Camera:  
 def \_\_init\_\_(self):  
 x, y = hero.pos  
 self.dx = x  
 self.dy = y  
  
 def apply(self, obj):  
 obj.rect.x += self.dx \* 50  
 obj.rect.y += self.dy \* 50  
  
 def update(self, target):  
 self.dx = -(target.rect.x + target.rect.w // 2 - width // 2)  
 self.dy = -(target.rect.y + target.rect.h // 2 - height // 2)  
  
 def update\_up(self, target):  
 self.dx = 0  
 self.dy = 1  
  
 def update\_down(self, target):  
 self.dx = 0  
 self.dy = -1  
  
 def update\_right(self, target):  
 self.dx = -1  
 self.dy = 0  
  
 def update\_left(self, target):  
 self.dx = 1  
 self.dy = 0  
  
  
def terminate():  
 pygame.quit()  
 sys.exit()  
  
  
def start\_screen():  
 intro\_text = ["Cave Game", '',  
 "Цель - дойти до дыры"]  
 fon = pygame.transform.scale(load\_image('E:/pythonProject1/dirt.png'), size)  
 screen.blit((fon), (0, 0))  
 font = pygame.font.Font(None, 30)  
 text\_coord = 50  
 for line in intro\_text:  
 string\_rendered = font.render(line, 1, pygame.Color('white'))  
 intro\_rect = string\_rendered.get\_rect()  
 text\_coord += 10  
 intro\_rect.top = text\_coord  
 intro\_rect.x = 10  
 text\_coord += intro\_rect.height  
 screen.blit(string\_rendered, intro\_rect)  
  
 while True:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 terminate()  
 elif event.type == pygame.KEYDOWN or event.type == pygame.MOUSEBUTTONDOWN:  
 return  
 pygame.display.flip()  
  
  
def end\_screen():  
 intro\_text = ["Cave Game", '',  
 f"Время - {time.time() - seconds} секунды"]  
 fon = pygame.transform.scale(load\_image('E:/pythonProject1/dirt.png'), size)  
 screen.blit((fon), (0, 0))  
 font = pygame.font.Font(None, 30)  
 text\_coord = 50  
 for line in intro\_text:  
 string\_rendered = font.render(line, 1, pygame.Color('white'))  
 intro\_rect = string\_rendered.get\_rect()  
 text\_coord += 10  
 intro\_rect.top = text\_coord  
 intro\_rect.x = 10  
 text\_coord += intro\_rect.height  
 screen.blit(string\_rendered, intro\_rect)  
  
 while True:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 terminate()  
 elif event.type == pygame.KEYDOWN or event.type == pygame.MOUSEBUTTONDOWN:  
 file\_filler()  
 level\_smoother()  
 rocks\_creator()  
 ors\_creator()  
 player\_creator()  
 ex\_creator()  
 ways\_creator()  
 global level\_map  
 level\_map = load\_level('уровень.txt')  
 hero, max\_x, max\_y = generate\_level(level\_map)  
 for sprite in sprite\_group:  
 print(sprite)  
 print(sprite\_group)  
 camera = Camera()  
 print(hero.pos)  
 if hero in sprite\_group:  
 print(111)  
 camera.update(hero)  
 return  
 pygame.display.flip()  
  
  
def load\_level(filename):  
 filename = 'E:/pythonProject1/' + filename  
 with open(filename, 'r') as mapFile:  
 level\_map = [line.strip() for line in mapFile]  
 max\_width = max(map(len, level\_map))  
 return list(map(lambda x: list(x.ljust(max\_width, '1')), level\_map))  
  
  
def generate\_level(level):  
 new\_player = 0  
 x = 0  
 y = 0  
 for y in range(len(level)):  
 for x in range(len(level[y])):  
 if level[y][x] == '0':  
 Tile('dirt', x, y)  
 elif level[y][x] == '1':  
 Tile('wall', x, y)  
 elif level[y][x] == '2':  
 Tile('stone', x, y)  
 elif level[y][x] == '3':  
 Tile('ore', x, y)  
 elif level[y][x] == '4':  
 Tile('hole', x, y)  
 elif level[y][x] == '5':  
 Tile('dirt', x, y)  
 new\_player = Player(x, y)  
 return new\_player, x, y  
  
  
def move(hero, movement):  
 x, y = hero.pos  
 if movement == 'up':  
 if y > 0 and (level\_map[y - 1][x] == '0' or level\_map[y - 1][x] == '4' or level\_map[y - 1][x] == '5'):  
 print(2)  
 camera.update\_up(hero)  
 for sprite in sprite\_group:  
 camera.apply(sprite)  
 hero.pos = x, y - 1  
 # hero.move(x, y - 1)  
 elif movement == 'down':  
 if y < max\_y - 1 and (level\_map[y + 1][x] == '0' or level\_map[y + 1][x] == '4' or level\_map[y + 1][x] == '5'):  
 print(2)  
 camera.update\_down(hero)  
 for sprite in sprite\_group:  
 camera.apply(sprite)  
 hero.pos = x, y + 1  
 # hero.move(x, y + 1)  
 elif movement == 'left':  
 if x > 0 and (level\_map[y][x - 1] == '0' or level\_map[y][x - 1] == '4' or level\_map[y][x - 1] == '5'):  
 camera.update\_left(hero)  
 for sprite in sprite\_group:  
 camera.apply(sprite)  
 hero.pos = x - 1, y  
 # hero.move(x - 1, y)  
 elif movement == 'right':  
 if x < max\_x - 1 and (level\_map[y][x + 1] == '0' or level\_map[y][x + 1] == '4' or level\_map[y][x + 1] == '5'):  
 camera.update\_right(hero)  
 for sprite in sprite\_group:  
 camera.apply(sprite)  
 hero.pos = x + 1, y  
 # hero.move(x + 1, y)  
 if level\_map[y][x] == '4':  
 end\_screen()  
 hero.pos = (1, 1)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 running = True  
 start\_screen()  
 level\_map = load\_level('уровень.txt')  
 hero, max\_x, max\_y = generate\_level(level\_map)  
 for sprite in sprite\_group:  
 print(sprite)  
 print(sprite\_group)  
 camera = Camera()  
 print(hero.pos)  
 if hero in sprite\_group:  
 print(111)  
 camera.update(hero)  
 seconds = time.time()  
 while running:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 running = False  
 elif event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_UP:  
 print(1)  
 move(hero, 'up')  
 if event.key == pygame.K\_DOWN:  
 print(1)  
 move(hero, 'down')  
 if event.key == pygame.K\_RIGHT:  
 move(hero, 'right')  
 if event.key == pygame.K\_LEFT:  
 move(hero, 'left')  
  
 screen.fill(pygame.Color('black'))  
 print(hero.pos)  
 sprite\_group.draw(screen)  
 hero\_group.draw(screen)  
 pygame.display.flip()  
 pygame.quit()