Nama : Alfida Sofiya Mufti

NIM : V3920005

Kelas : TI D

PRAKTIK GAME DEVELOPMENT

1. Cobalah program pada poin C. Kode program pada poin C terdiri dari beberapa Part. Susun bagian-bagian kode tersebut sehingga dapat menjadi satu kesatuan program utuh !
   Jawab :

```
#Part A
import pygame, sys, random

class Block(pygame.sprite.Sprite):
def __init__(self,path,x_pos,y_pos):
super().__init__()
self.image = pygame.image.load(path)
self.rect = self.image.get_rect(center = (x_pos,y_pos))
#Part E
class Player(Block):
def __init__(self,path,x_pos,y_pos,speed):
super().__init__(path,x_pos,y_pos)
self.speed = speed
self.movement = 0

def screen_constrain(self):
if self.rect.top <= 0:
self.rect.top = 0
if self.rect.bottom >= screen_height:
self.rect.bottom = screen_height

def update(self,ball_group):
self.rect.y += self.movement
self.screen_constrain()
#Part C
class Ball(Block):
def __init__(self,path,x_pos,y_pos,speed_x,speed_y,paddles):
super().__init__(path,x_pos,y_pos)
self.speed_x = speed_x * random.choice((-1,1))
self.speed_y = speed_y * random.choice((-1,1))
self.paddles = paddles
self.active = False
self.score_time = 0

def update(self):
if self.active:
self.rect.x += self.speed_x
self.rect.y += self.speed_y
self.collisions()
else:
self.restart_counter()
#Part G
def collisions(self):
if self.rect.top <= 0 or self.rect.bottom >= screen_height:
pygame.mixer.Sound.play(plob_sound)
self.speed_y *= -1

#untuk mengatur Pandle di ball nya saat memantul
if pygame.sprite.spritecollide(self,self.paddles,False):
pygame.mixer.Sound.play(plob_sound)
collision_paddle = pygame.sprite.spritecollide(self,self.paddles,False)[0].rect
if abs(self.rect.right - collision_paddle.left) < 10 and self.speed_x > 0:
self.speed_x *= -1
if abs(self.rect.left - collision_paddle.right) < 10 and self.speed_x < 0:
self.speed_x *= -1
if abs(self.rect.top - collision_paddle.bottom) < 10 and self.speed_y < 0:
self.rect.top = collision_paddle.bottom
self.speed_y *= -1
if abs(self.rect.bottom - collision_paddle.top) < 10 and self.speed_y > 0:
self.rect.bottom = collision_paddle.top
self.speed_y *= -1
#Part B
def reset_ball(self):
self.active = False
self.speed_x *= random.choice((-1,1))
self.speed_y *= random.choice((-1,1))
self.score_time = pygame.time.get_ticks()
self.rect.center = (screen_width/2,screen_height/2)
pygame.mixer.Sound.play(score_sound)
#Part N
def restart_counter(self):
current_time = pygame.time.get_ticks()
countdown_number = 3

if current_time - self.score_time <= 700:
countdown_number = 3
if 700 < current_time - self.score_time <= 1400:
countdown_number = 2
```

```
if 1400 < current_time - self.score_time <= 2100:
countdown_number = 1
if current_time - self.score_time >= 2100:
self.active = True

time_counter = basic_font.render(str(countdown_number),True,accent_color)
time_counter_rect = time_counter.get_rect(center = (screen_width/2,screen_height/2 + 50))
pygame.draw.rect(screen,bg_color,time_counter_rect)
screen.blit(time_counter,time_counter_rect)
#Part J
class Opponent(Block): #ini adalah bagian opponent AI/Artificial Intelegence
def __init__(self,path,x_pos,y_pos,speed): #membuat fungsi init yang mengambil posisi dari self,path,x_pos,y_pos,speed
super().__init__(path,x_pos,y_pos)
self.speed = speed #membuat properti speed

def update(self,ball_group):
if self.rect.top < ball_group.sprite.rect.y: #jika paddle yang berada di posisi atas maka posisi bola akan memantulkan lawan arah
self.rect.y += self.speed #akan bertambah kecepatannya
if self.rect.bottom > ball_group.sprite.rect.y: #jika paddle yang berada di posisi bawah maka posisi bola akan memantul ke atas atau lawan arah
self.rect.y -= self.speed
self.constrain()

def constrain(self): #sebagai fungsi pembatas permainan
if self.rect.top <= 0: self.rect.top = 0 #ketika musuh berada disisi atas
if self.rect.bottom >= screen_height: self.rect.bottom = screen_height #ketika musuh berada di sisi bawah dan tidak lebih sama dengan ukuran layar atau self.rect.bottom = screen_height
#Part I
class GameManager:
def __init__(self,ball_group,paddle_group):
self.player_score = 0
self.opponent_score = 0
self.ball_group = ball_group
self.paddle_group = paddle_group

def run_game(self):
# Memulai
self.paddle_group.draw(screen)
self.ball_group.draw(screen)

# Mengupdate Ball
self.paddle_group.update(self.ball_group)
self.ball_group.update()
self.reset_ball()
self.draw_score()
#Part K
def reset_ball(self):
if self.ball_group.sprite.rect.right >= screen_width:
self.opponent_score += 1
self.ball_group.sprite.reset_ball()
if self.ball_group.sprite.rect.left <= 0:
self.player_score += 1
self.ball_group.sprite.reset_ball()

def draw_score(self):
player_score = basic_font.render(str(self.player_score),True,accent_color)
opponent_score = basic_font.render(str(self.opponent_score),True,accent_color)

player_score_rect = player_score.get_rect(midleft = (screen_width / 2 +
40,screen_height/2))
opponent_score_rect = opponent_score.get_rect(midright = (screen_width / 2 -
40,screen_height/2))

screen.blit(player_score,player_score_rect)
screen.blit(opponent_score,opponent_score_rect)
#Part D
# pengaturan umum
pygame.mixer.pre_init(44100,-16,2,512)
pygame.init()
clock = pygame.time.Clock()

# Property ada widhtm height yang di satukan menjadi screen
screen_width = 720
screen_height = 480
screen = pygame.display.set_mode((screen_width,screen_height))
pygame.display.set_caption('Pong')

# Variabel Global
bg_color = pygame.Color('#2F373F')
accent_color = (27,35,43)
basic_font = pygame.font.Font('freesansbold.ttf', 32) #untuk merubah Jenis font
plob_sound = pygame.mixer.Sound("pong.ogg") #untuk menambahkan sound ketika bola
memantul
score_sound = pygame.mixer.Sound("score.ogg") # untuk ketika permainan berhenti
atau bola loss
middle_strip = pygame.Rect(screen_width/2 - 2,0,4,screen_height)
#Part F
# Objek Game
```

```
player = Player('Paddle.png',screen_width - 20,screen_height/2,5) #menambahkan object paddle png
opponent = Opponent('Paddle.png',20,screen_width/2,5)#menambahkan object paddle png
paddle_group = pygame.sprite.Group()
paddle_group.add(player)
paddle_group.add(opponent)

ball = Ball('Ball.png',screen_width/2,screen_height/2,4,4,paddle_group)
ball_sprite = pygame.sprite.GroupSingle()
ball_sprite.add(ball)

game_manager = GameManager(ball_sprite,paddle_group)
#Part M
#part ini untuk mengatur keyboard agar berfungsi untuk melakukan gerakan pada game menggunakan UP, DOWN, UP, DOWN
while True:
for event in pygame.event.get():
if event.type == pygame.QUIT:
pygame.quit()
sys.exit()
if event.type == pygame.KEYDOWN:
if event.key == pygame.K_UP:
player.movement -= player.speed
if event.key == pygame.K_DOWN:
player.movement += player.speed
if event.type == pygame.KEYUP:
if event.key == pygame.K_UP:
player.movement += player.speed
if event.key == pygame.K_DOWN:
player.movement -= player.speed

#Part L
screen.fill(bg_color) #Membuat Latar Belakang
pygame.draw.rect(screen,accent_color,middle_strip)#Menggambar Rect dengan Screen Width dan height dan Midle String

game_manager.run_game() # Untuk menjalankan game

pygame.display.flip()  # Rendering
clock.tick(120)
```

2. Langkah selanjutnya adalah, identifikasi pada bagian manakah implementasi AI pada program game tersebut. Jelaskan !

   Jawab :

Pada program diatas implementasi AI terdapat di part J. pada source code part J tersebut berfungsi untuk memprediksi gerakan player dengan gerakan yaitu memantulkan, memukul, bertahan, berjalan mundur, berjalan maju. Mengoptimalkan pergerakan NPC agar bisa memprediksi arah pukulan bola yang dipukul oleh player sebelum memprediksi NPC (Non Player Character) hanya akan menebak arah pukulan secara acak.

3. Jelaskan bagaimana alur AI yang digunakan pada program tersebut !

Jawab :

Alur AI berfungsi untuk mengatur arah gerak bola dan paddle pada program game pingpong tersebut.