LAPORAN PRAKTIKUM

PRAKTIK GAME DEVELOPMENT

TUGAS 5

Disusun oleh :

Arin Dwi Padmasari (V3920010)

PROGRAM STUDI DIII TEKNIK INFORMATIKA PSDKU

SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

2021

D. TUGAS INDIVIDU

a. 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 !

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

# Class Block terdapat fungsi init
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 ini ada fungsi init, screen constraint, update
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 ada fungsi init, update, restart counter, reset ball dan collisions
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

        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

#Awal 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 M
    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)

# Part I
# Class Opponent ada fungsi init, update, dan constraint
class Opponent(Block):
    def __init__(self, path, x_pos, y_pos, speed):
        super().__init__(path, x_pos, y_pos)
        self.speed = speed

    def update(self, ball_group):
        if self.rect.top < ball_group.sprite.rect.y:
            self.rect.y += self.speed
        if self.rect.bottom > ball_group.sprite.rect.y:
            self.rect.y -= self.speed
        self.constrain()

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

# Part H
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):
        # menggambar objek game
        self.paddle_group.draw(screen)
        self.ball_group.draw(screen)

        # Update objek game
        self.paddle_group.update(self.ball_group)
        self.ball_group.update()
        self.reset_ball()
        self.draw_score()

# Part J
    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
# untuk pengaturan waktu
pygame.mixer.pre_init(44100, -16, 2, 512)
pygame.init()
clock = pygame.time.Clock()

# Pengaturan tinggi dan panjang Layar
screen_width = 720
screen_height = 480
screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption('Pong')
```

```
# untuk pengaturan tampilan dan suara di Game
bg_color = pygame.Color('#2F373F')
accent_color = (27, 35, 43)
basic_font = pygame.font.Font('freesansbold.ttf', 32)
plob_sound = pygame.mixer.Sound("pong.ogg")
score_sound = pygame.mixer.Sound("score.ogg")
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)
opponent = Opponent('Paddle.png', 20, screen_width / 2, 5)
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 L
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 K
    # background
    screen.fill(bg_color)
    pygame.draw.rect(screen, accent_color, middle_strip)

    # Untuk running game
    game_manager.run_game()

    # Render game
    pygame.display.flip()
    clock.tick(120)
```

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

Implementasi nya terdapt pada block , Jika tidak ada bagian tersebut maka AI tidak bisa berpindah tempat.

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

Pada progra di atas menjelaskan mengenai permainan pong. Dengan menampilkan bola yang berada di tengah gambar dan pada samping kanan kiri terdapat semacam bentengan untuk memantulkan bol pada saat bergerak . Bola tersebut akan bergerak secara acak ke kanan atau kiri. Pada bentengan bagian kanan akan dapat dikendalikan oleh user dan pada bagian kiri akan mengikuti arah bola menuju. Jika bola melewati beteng pantulan maka score akan bertambah dipihak lawan.