NAMA : KRESHNA PUTRA ADI WICAKSANA

NIM / KELAS : V3920032 / TI-D

MATAKULIAH : PRAKTIK GAME DEVELOPMENT

```python
#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
    # digunakan untuk membuat sebuah fungsi otomatis
    def collisions(self):
        #  percabangan jika nilai top kurang dari 0 dan bottom lebih dari ukuran window
        if self.rect.top <= 0 or self.rect.bottom >= screen_height:
            # digunakan untuk memutar sound atau musik
            pygame.mixer.Sound.play(plob_sound)
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
            self.speed_y *= -1

        # percabangan jika nilai salah akan menjalankan program yang ada
        if pygame.sprite.spritecollide(self,self.paddles,False):
            # digunakan untuk memutar sound
            pygame.mixer.Sound.play(plob_sound)
            # pendeklarasian nilai collision paddle otomatis
            collision_paddle = pygame.sprite.spritecollide(self,self.paddles,False)[0].rect
            # percabangan nilai jika paddle kiri kurang dari 10 pada posisinya dan kecepatan x lebih dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.right - collision_paddle.left) < 10 and self.speed_x > 0:
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                self.speed_x *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan x kurang dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.left - collision_paddle.right) < 10 and self.speed_x < 0:
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                    self.speed_x *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y kurang dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.top - collision_paddle.bottom) < 10 and self.speed_y < 0:
                # deklrasi persamaan nilai rect top dan paddle bottom
                self.rect.top = collision_paddle.bottom
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                self.speed_y *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y lebih dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.bottom - collision_paddle.top) < 10 and self.speed_y > 0:
                # deklrasi persamaan nilai rect bottom dan paddle top
                self.rect.bottom = collision_paddle.top
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                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 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
```

```
            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 I
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):
        # Drawing the game objects
        self.paddle_group.draw(screen)
        self.ball_group.draw(screen)

        # Updating the game objects
        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
pygame.mixer.pre_init(44100,-16,2,512)
pygame.init()
clock = pygame.time.Clock()

# Main Window
screen_width = 720
screen_height = 480
screen = pygame.display.set_mode((screen_width,screen_height))
pygame.display.set_caption('Pong')

# Global Variables
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
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 Stuff
    screen.fill(bg_color)
    pygame.draw.rect(screen,accent_color,middle_strip)

    # Run the game
    game_manager.run_game()

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

POTONGAN AI

```
#PART G
    # digunakan untuk membuat sebuah fungsi otomatis
    def collisions(self):
        #  percabangan jika nilai top kurang dari 0 dan bottom lebih dari ukuran window
        if self.rect.top <= 0 or self.rect.bottom >= screen_height:
            # digunakan untuk memutar sound atau musik
            pygame.mixer.Sound.play(plob_sound)
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
            self.speed_y *= -1

        # percabangan jika nilai salah akan menjalankan program yang ada
        if pygame.sprite.spritecollide(self,self.paddles,False):
            # digunakan untuk memutar sound
            pygame.mixer.Sound.play(plob_sound)
            # pendeklarasian nilai collision paddle otomatis
            collision_paddle = pygame.sprite.spritecollide(self,self.paddles,False)[0].rect
            # percabangan nilai jika paddle kiri kurang dari 10 pada posisinya dan kecepatan x lebih dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.right - collision_paddle.left) < 10 and self.speed_x > 0:
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                self.speed_x *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan x kurang dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.left - collision_paddle.right) < 10 and self.speed_x < 0:
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                    self.speed_x *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y kurang dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.top - collision_paddle.bottom) < 10 and self.speed_y < 0:
                # deklrasi persamaan nilai rect top dan paddle bottom
                self.rect.top = collision_paddle.bottom
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                self.speed_y *= -1
            # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y lebih dari 0 maka akan mengkalikan program didalamnya
            if abs(self.rect.bottom - collision_paddle.top) < 10 and self.speed_y > 0:
                # deklrasi persamaan nilai rect bottom dan paddle top
                self.rect.bottom = collision_paddle.top
            # digunakan untuk pendeklarasian nilai speed y dikalikan -1
                self.speed_y *= -1
```

ALUR AI BERJALAN

AI ini berjalan dengan mengikuti pergerakan dan memperhatikan batasan gerakan yang ada yaitu batas atas dan bawah pada jendela yang kemudian hal ini akan membuatnya berbalik.

Kemudian AI ini akan melakukan perulangan paddle keatas dan kebawah dengan kecepatan yang sudah diatur dan disesuaikan dengan kecepatan bola yang datang ke bagian dari UI