**LAPORAN PRAKTIKUM**

**PRAKTIK GAME DEVELOPMENT**

**TUGAS 5**

Disusun oleh :

Andrean Ludvi Nur Aziz (V3920007)

**PROGRAM STUDI DIII TEKNIK INFORMATIKA PSDKU**

**SEKOLAH VOKASI**

**UNIVERSITAS SEBELAS MARET**

**2021**

D. TUGAS INDIVIDU

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 !

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

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

4. Pengumpulan Tugas Praktikum.

>Tempatkan kode program, beserta jawaban pada nomor 1, 2, dan 3 di akun github masing-masing. Kemudian kumpulkan link akun github di SPADA

**>Untuk kelas TI E** Tugas Praktikum di kumpulkan paling lambat tanggal **28 September 2021 jam 23.59**

**>Untuk kelas TI D** Tugas Praktikum di kumpulkan paling lambat tanggal **29 September 2021 jam 23.59**

JAWABAN :

1. Hasil Program

#Awal Part A  
import pygame, sys, random  
  
# Pada class Block dibawah 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))  
#Akhir part A  
  
#Awal Part E  
# Pada class Player dibawah terdapat fungsi init, update, screen constraint  
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()  
#Akhir part E  
  
#Awal Part C  
# Pada class Ball dibawah terdapat fungsi init, update, collisions, restart counter, reset ball  
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()  
#Akhir Part C  
  
#Awal 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  
#Akhir part G  
  
#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)  
#Akhir Part B  
  
#Awal 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)  
#Akhir part M  
  
#Awal part I  
# Pada class Opponent dibawah terdapat fungsi init, update, 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  
#Akhir part I  
  
#Awal Part H  
# Pada class Game Manager dibawah terdapat fungsi init, run game, reset ball, draw score  
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)  
  
 # mengupdate objek game  
 self.paddle\_group.update(self.ball\_group)  
 self.ball\_group.update()  
 self.reset\_ball()  
 self.draw\_score()  
#Akhir Part H  
  
#Awal 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)  
#Akhir part J  
  
#Awal part D  
# clock  
pygame.mixer.pre\_init(44100, -16, 2, 512)  
pygame.init()  
clock = pygame.time.Clock()  
  
# Mengatur tinggi dan panjang Layar  
screen\_width = 720  
screen\_height = 480  
screen = pygame.display.set\_mode((screen\_width, screen\_height))  
pygame.display.set\_caption('Pong')  
  
# Mengatur Tampilan dan suara 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)  
#Akhir part D  
  
#Awal 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)  
#Akhir part F  
  
#AAwal 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  
#Akhir part L  
  
#Awal part K  
 # background  
 screen.fill(bg\_color)  
 pygame.draw.rect(screen, accent\_color, middle\_strip)  
  
 # Untuk run game  
 game\_manager.run\_game()  
  
 # Render game  
 pygame.display.flip()  
 clock.tick(120)  
#Akhir part K

2. Bagian implementasinya terdapat pada class block,karena tanpa bagian ini AI tidak akan dapat bergerak

3. Bola berada di tengah window kemudian jika hitungan waktu mundur sudah dimulai maka bola akan bergerak secara acak di layar yang di sediakan, bagian sebelah kanan akan dapat digerakkan oleh player/pemain sedangkan pada bagian sebelah kiri akan mengikuti pergerakan bola.

Setelah itu bola akan bergerak sesuai dengan arah pantulan yang dihasilkan dan jika bola gagal dipantulkan maka pihak musuh akan mendapatkan poin.