

NAMA : KRESHNA PUTRA ADI WICAKSANA

NIM / KELAS : V3920032 / TI-D

MATAKULIAH : PRAKTIK GAME DEVELOPMENT

```
1 #PART A
2 import pygame, sys, random
3
4 class Block(pygame.sprite.Sprite):
5     def __init__(self, path, x_pos, y_pos):
6         super().__init__()
7         self.image = pygame.image.load(path)
8         self.rect = self.image.get_rect(center = (x_pos, y_pos))
9
10 #PART E
11 class Player(Block):
12     def __init__(self, path, x_pos, y_pos, speed):
13         super().__init__(path, x_pos, y_pos)
14         self.speed = speed
15         self.movement = 0
16
17     def screen_constrain(self):
18         if self.rect.top <= 0:
19             self.rect.top = 0
20         if self.rect.bottom >= screen_height:
21             self.rect.bottom = screen_height
22
23     def update(self, ball_group):
24         self.rect.y += self.movement
25         self.screen_constrain()
26
27 #PART C
28 class Ball(Block):
29     def __init__(self, path, x_pos, y_pos, speed_x, speed_y, paddles):
30         super().__init__(path, x_pos, y_pos)
31         self.speed_x = speed_x * random.choice((-1,1))
32         self.speed_y = speed_y * random.choice((-1,1))
33         self.paddles = paddles
34         self.active = False
35         self.score_time = 0
36
37     def update(self):
38         if self.active:
39             self.rect.x += self.speed_x
40             self.rect.y += self.speed_y
41             self.collisions()
42         else:
43             self.restart_counter()
44
45 #PART G
46 # digunakan untuk membuat sebuah fungsi otomatis
47 def collisions(self):
48     # percabangan jika nilai top kurang dari 0 dan bottom lebih dari ukuran window
49     if self.rect.top <= 0 or self.rect.bottom >= screen_height:
50         # digunakan untuk memutar sound atau musik
51         pygame.mixer.Sound.play(plob_sound)
52         # digunakan untuk pendeklarasian nilai speed y dikalikan -1
53         self.speed_y *= -1
54
55     # percabangan jika nilai salah akan menjalankan program yang ada
56     if pygame.sprite.spritecollide(self, self.paddles, False):
57         # digunakan untuk memutar sound
58         pygame.mixer.Sound.play(plob_sound)
59         # pendeklarasian nilai collision paddle otomatis
60         collision_paddle = pygame.sprite.spritecollide(self, self.paddles, False)[0].rect
61         # percabangan nilai jika paddle kiri kurang dari 10 pada posisinya dan kecepatan x lebih dari 0 maka akan mengkalikan program didalamnya
62         if abs(self.rect.right - collision_paddle.left) < 10 and self.speed_x > 0:
63             # digunakan untuk pendeklarasian nilai speed y dikalikan -1
64             self.speed_x *= -1
65         # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan x kurang dari 0 maka akan mengkalikan program didalamnya
66         if abs(self.rect.left - collision_paddle.right) < 10 and self.speed_x < 0:
67             # digunakan untuk pendeklarasian nilai speed y dikalikan -1
68             self.speed_x *= -1
69         # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y kurang dari 0 maka akan mengkalikan program didalamnya
70         if abs(self.rect.top - collision_paddle.bottom) < 10 and self.speed_y < 0:
71             # deklrasi persamaan nilai rect top dan paddle bottom
72             self.rect.top = collision_paddle.bottom
73             # digunakan untuk pendeklarasian nilai speed y dikalikan -1
74             self.speed_y *= -1
75         # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y lebih dari 0 maka akan mengkalikan program didalamnya
76         if abs(self.rect.bottom - collision_paddle.top) < 10 and self.speed_y > 0:
77             # deklrasi persamaan nilai rect bottom dan paddle top
78             self.rect.bottom = collision_paddle.top
79             # digunakan untuk pendeklarasian nilai speed y dikalikan -1
80             self.speed_y *= -1
81
82 #PART B
83 def reset_ball(self):
84     self.active = False
85     self.speed_x = random.choice((-1,1))
86     self.speed_y = random.choice((-1,1))
87     self.score_time = pygame.time.get_ticks()
88     self.rect.center = (screen_width/2, screen_height/2)
89     pygame.mixer.Sound.play(score_sound)
90
91 # PART M
92 def restart_counter(self):
93     current_time = pygame.time.get_ticks()
94     countdown_number = 3
95
96     if current_time - self.score_time <= 700:
97         countdown_number = 3
98     if 700 < current_time - self.score_time <= 1400:
99         countdown_number = 2
```

```

94         countdown_number = 2
95         if 1400 < current_time - self.score_time ≤ 2100:
96             countdown_number = 1
97         if current_time - self.score_time ≥ 2100:
98             self.active = True
99
100         time_counter = basic_font.render(str(countdown_number),True,accent_color)
101         time_counter_rect = time_counter.get_rect(center = (screen_width/2,screen_height/2 + 50))
102         pygame.draw.rect(screen,bg_color,time_counter_rect)
103         screen.blit(time_counter,time_counter_rect)
104
105 #PART I
106 class Opponent(Block):
107     def __init__(self,path,x_pos,y_pos,speed):
108         super().__init__(path,x_pos,y_pos)
109         self.speed = speed
110
111     def update(self,ball_group):
112         if self.rect.top < ball_group.sprite.rect.y:
113             self.rect.y += self.speed
114         if self.rect.bottom > ball_group.sprite.rect.y:
115             self.rect.y -= self.speed
116         self.constrain()
117
118     def constrain(self):
119         if self.rect.top ≤ 0: self.rect.top = 0
120         if self.rect.bottom ≥ screen_height: self.rect.bottom = screen_height
121
122 #PART H
123 class GameManager:
124     def __init__(self,ball_group,paddle_group):
125         self.player_score = 0
126         self.opponent_score = 0
127         self.ball_group = ball_group
128         self.paddle_group = paddle_group
129
130     def run_game(self):
131         # Drawing the game objects
132         self.paddle_group.draw(screen)
133         self.ball_group.draw(screen)
134
135         # Updating the game objects
136         self.paddle_group.update(self.ball_group)
137         self.ball_group.update()
138         self.reset_ball()
139         self.draw_score()
140
141 # PART J
142     def reset_ball(self):
143         if self.ball_group.sprite.rect.right ≥ screen_width:
144             self.opponent_score += 1
145             self.ball_group.sprite.reset_ball()
146         if self.ball_group.sprite.rect.left ≤ 0:
147             self.player_score += 1
148             self.ball_group.sprite.reset_ball()
149
150     def draw_score(self):
151         player_score = basic_font.render(str(self.player_score),True,accent_color)
152         opponent_score = basic_font.render(str(self.opponent_score),True,accent_color)
153
154         player_score_rect = player_score.get_rect(midleft = (screen_width / 2 + 40,screen_height/2))
155         opponent_score_rect = opponent_score.get_rect(midright = (screen_width / 2 - 40,screen_height/2))
156
157         screen.blit(player_score,player_score_rect)
158         screen.blit(opponent_score,opponent_score_rect)
159
160 # PART D
161 pygame.mixer.pre_init(44100,-16,2,512)
162 pygame.init()
163 clock = pygame.time.Clock()
164
165 # Main Window
166 screen_width = 720
167 screen_height = 480
168 screen = pygame.display.set_mode((screen_width,screen_height))
169 pygame.display.set_caption('Pong')
170
171 # Global Variables
172 bg_color = pygame.Color('#2F373F')
173 accent_color = (27,35,43)
174 basic_font = pygame.font.Font('freesansbold.ttf', 32)
175 plob_sound = pygame.mixer.Sound("pong.ogg")
176 score_sound = pygame.mixer.Sound("score.ogg")
177 middle_strip = pygame.Rect(screen_width/2 - 2,0,4,screen_height)
178
179 # PART F
180 player = Player('Paddle.png',screen_width - 20,screen_height/2,5)
181 opponent = Opponent('Paddle.png',20,screen_width/2,5)
182 paddle_group = pygame.sprite.Group()
183 paddle_group.add(player)
184 paddle_group.add(opponent)
185
186 ball = Ball('Ball.png',screen_width/2,screen_height/2,4,4,paddle_group)
187 ball_sprite = pygame.sprite.GroupSingle()
188 ball_sprite.add(ball)
189
190 game_manager = GameManager(ball_sprite,paddle_group)
191
192 # PART L
193 while True:
194     for event in pygame.event.get():
195         if event.type == pygame.QUIT:
196             pygame.quit()
197             sys.exit()
198         if event.type == pygame.KEYDOWN:
199             if event.key == pygame.K_UP:
200                 player.movement -= player.speed
201             if event.key == pygame.K_DOWN:
202                 player.movement += player.speed
203             if event.type == pygame.KEYUP:
204                 if event.key == pygame.K_UP:
205                     player.movement += player.speed
206                 if event.key == pygame.K_DOWN:
207                     player.movement -= player.speed

```

```

284 # PART K
285 # Background Stuff
286 screen.fill(bg_color)
287 pygame.draw.rect(screen,accent_color,middle_strip)
288
289 # Run the game
290 game_manager.run_game()
291
292 # Rendering
293 pygame.display.flip()
294 clock.tick(120)

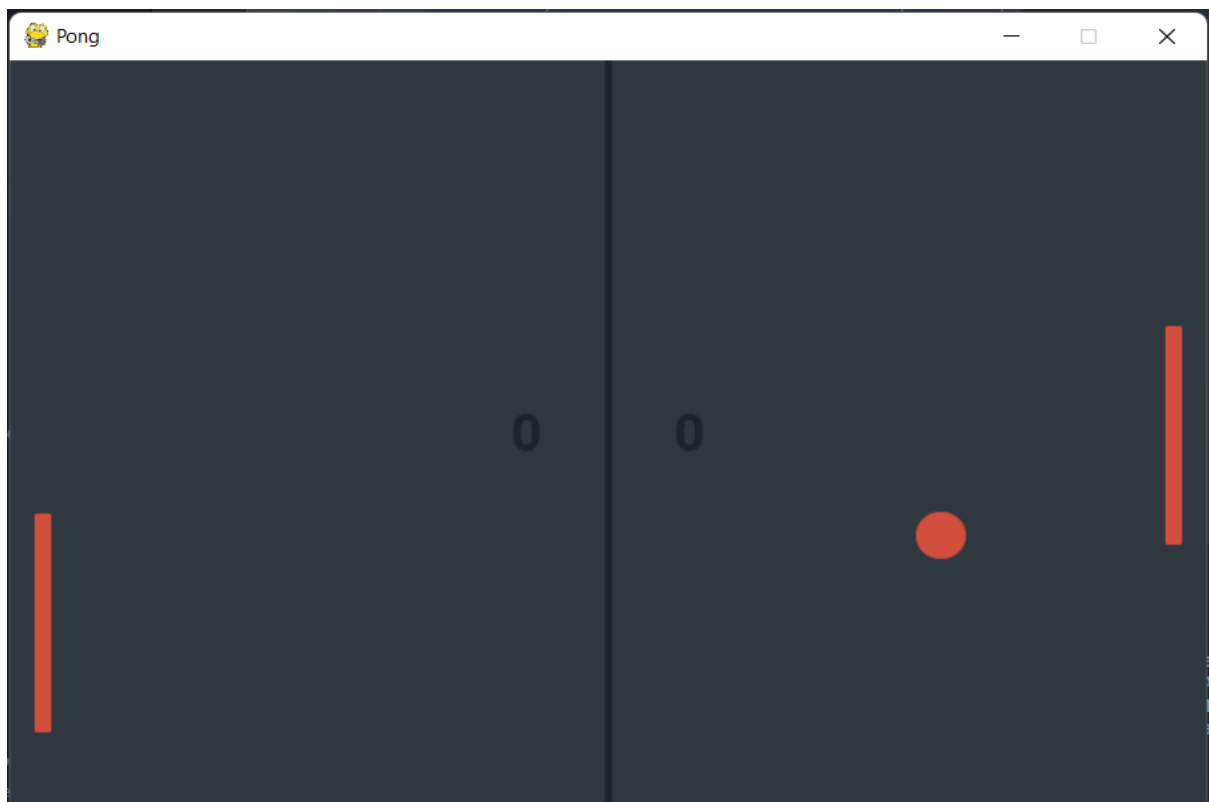
```

POTONGAN AI

```

1 #PART G
2 # digunakan untuk membuat sebuah fungsi otomatis
3 def collisions(self):
4     # percabangan jika nilai top kurang dari 0 dan bottom lebih dari ukuran window
5     if self.rect.top <= 0 or self.rect.bottom >= screen_height:
6         # digunakan untuk memutar sound atau musik
7         pygame.mixer.Sound.play(plob_sound)
8         # digunakan untuk pendeklarasian nilai speed y dikalikan -1
9         self.speed_y *= -1
10
11 # percabangan jika nilai salah akan menjalankan program yang ada
12 if pygame.sprite.spritecollide(self,self.paddles,False):
13     # digunakan untuk memutar sound
14     pygame.mixer.Sound.play(plob_sound)
15     # pendeklarasian nilai collision_paddle otomatis
16     collision_paddle = pygame.sprite.spritecollide(self,self.paddles,False)[0].rect
17     # percabangan nilai jika paddle kiri kurang dari 10 pada posisinya dan kecepatan x lebih dari 0 maka akan mengkalikan program didalamnya
18     if abs(self.rect.right - collision_paddle.left) < 10 and self.speed_x > 0:
19         # digunakan untuk pendeklarasian nilai speed y dikalikan -1
20         self.speed_x *= -1
21     # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan x kurang dari 0 maka akan mengkalikan program didalamnya
22     if abs(self.rect.left - collision_paddle.right) < 10 and self.speed_x < 0:
23         # digunakan untuk pendeklarasian nilai speed y dikalikan -1
24         self.speed_x *= -1
25     # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y kurang dari 0 maka akan mengkalikan program didalamnya
26     if abs(self.rect.top - collision_paddle.bottom) < 10 and self.speed_y < 0:
27         # deklrasi persamaan nilai rect top dan paddle bottom
28         self.rect.top = collision_paddle.bottom
29     # digunakan untuk pendeklarasian nilai speed y dikalikan -1
30     self.speed_y *= -1
31     # percabangan nilai jika paddle kanan kurang dari 10 pada posisinya dan kecepatan y lebih dari 0 maka akan mengkalikan program didalamnya
32     if abs(self.rect.bottom - collision_paddle.top) < 10 and self.speed_y > 0:
33         # deklrasi persamaan nilai rect bottom dan paddle top
34         self.rect.bottom = collision_paddle.top
35     # digunakan untuk pendeklarasian nilai speed y dikalikan -1
36     self.speed_y *= -1
37

```



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