

Nama : D. Ihsan Maulana

Nim : 20220040069

Kelas : TI 22 C

TUGAS KECERDASAN BUATAN

SESI 14

Catatan :

1. Buatlah suatu permainan sederhana Board Game : Pong Game, ikuti tahapan prosesnya pada kode-kode berikut ini. Kemudian sertakan link kodingan Pong Game dengan hasil visualisasinya.

Jawaban :

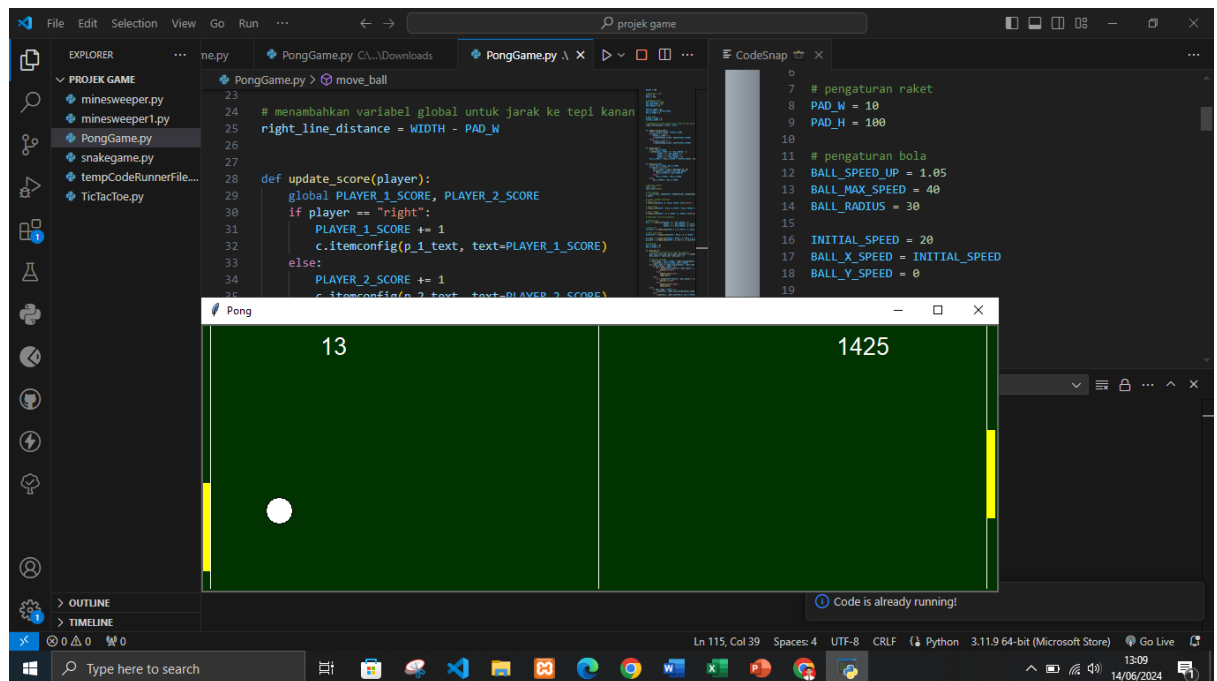
1. Berikut rangkaian Source Code dan hasil dari tahapan proses pada kode-kode dalam tugas :

• Code

```
1 from tkinter import *
2 import random
3
4 WIDTH = 600
5 HEIGHT = 400
6
7 # parameter rules
8 PADD_M = 10
9 PADD_W = 100
10
11 # parameter ball
12 BALL_SPEED_UP = 1.05
13 BALL_MAX_SPEED = 40
14 BALL_RADIUS = 10
15
16 INITIAL_SPEED = 20
17 BALL_X_SPEED = INITIAL_SPEED
18 BALL_Y_SPEED = 0
19
20 # skor awal
21 PLAYER_1_SCORE = 0
22 PLAYER_2_SCORE = 0
23
24 # parameter variabel global untuk skor ke dua pemain permainan
25 right_line_distance = WIDTH - PADD_W
26
27
28 def update_score(player):
29     global PLAYER_1_SCORE, PLAYER_2_SCORE
30     if player == "right":
31         PLAYER_1_SCORE += 1
32         c.delete(canvas_1_text, text=PLAYER_1_SCORE)
33     else:
34         PLAYER_2_SCORE += 1
35         c.delete(canvas_2_text, text=PLAYER_2_SCORE)
36
37
38 def spawn_ball():
39     global BALL_X_SPEED
40     c.coords(BALL, WIDTH / 2 - BALL_RADIUS / 2,
41             HEIGHT / 2 - BALL_RADIUS / 2,
42             WIDTH / 2 + BALL_RADIUS / 2,
43             HEIGHT / 2 + BALL_RADIUS / 2)
44     BALL_X_SPEED = -(BALL_X_SPEED * -INITIAL_SPEED) / abs(BALL_X_SPEED)
45
46
47 def bounce(ball):
48     global BALL_X_SPEED, BALL_Y_SPEED
49     if ball == "vertical":
50         BALL_X_SPEED = random.randrange(10, 10)
51         if abs(BALL_X_SPEED) < BALL_MAX_SPEED:
52             BALL_X_SPEED = -BALL_X_SPEED
53         else:
54             BALL_Y_SPEED = -BALL_Y_SPEED
55
56     elif:
57         BALL_Y_SPEED = -BALL_Y_SPEED
58
59 # membuat canvas
60 root = Tk()
61 root.title("Pong")
62
63 # area initial
64 c = Canvas(root, width=WIDTH, height=HEIGHT, background="black")
65 c.pack()
66
67 # membuat lapangan permainan
68 # garis kiri
69 c.create_line(PADD_W, 0, PADD_W, HEIGHT, fill="white")
70
71 # garis kanan
72 c.create_line(WIDTH - PADD_W, 0, WIDTH - PADD_W, HEIGHT, fill="white")
73
74 # garis tengah
75 c.create_line(WIDTH / 2, 0, WIDTH / 2, HEIGHT, fill="white")
76
77 # parameter lapangan permainan
78
79 # membuat bola
80 BALL = c.create_oval(WIDTH / 2 - BALL_RADIUS / 2,
81                     HEIGHT / 2 - BALL_RADIUS / 2, WIDTH / 2 + BALL_RADIUS / 2,
82                     HEIGHT / 2 + BALL_RADIUS / 2, fill="white")
83
84 # racket kiri
85 left_PADD = c.create_line(PADD_W / 2, 0, PADD_W / 2, PADD_W, width=PADD_W, fill="yellow")
86
87 # racket yang kanan
88 right_PADD = c.create_line(WIDTH - PADD_W, 0, WIDTH - PADD_W, PADD_W, width=PADD_W, fill="yellow")
89
90 p_1_text = c.create_text(WIDTH / 6, PADD_W / 4, text=PLAYER_1_SCORE, font="arial 20", fill="white")
91 p_2_text = c.create_text(WIDTH / 6, PADD_W / 4, text=PLAYER_2_SCORE, font="arial 20", fill="white")
```

```
1 # parameter
2 BALL_X_SPEED = 20
3 BALL_Y_SPEED = 0
4
5 def move_ball():
6     # menentukan koordinat titik bola dan pantainya
7     ball_left, ball_top, ball_right, ball_bot = c.coords(BALL)
8     ball_center = (ball_top + ball_bot) / 2
9
10     # pantainya vertikal
11     if ball_right < BALL_X_SPEED + right_line_distance and ball_left > BALL_X_SPEED - PADD_W:
12         c.move(ball, BALL_X_SPEED, BALL_Y_SPEED)
13     elif ball_right > right_line_distance or ball_left < -PADD_W:
14         if ball_right < WIDTH / 2:
15             c.coords(HEIGHT_PADD[1]) < ball_center + c.coords(HEIGHT_PADD[1])
16             bounce("vertical")
17         else:
18             update_score("left")
19             spawn_ball()
20     elif:
21         if c.coords(HEIGHT_PADD[1]) < ball_center + c.coords(HEIGHT_PADD[1]):
22             bounce("vertical")
23         else:
24             update_score("right")
25             spawn_ball()
26
27     if ball_right > WIDTH / 2:
28         c.move(ball, right_line_distance - ball_right, BALL_Y_SPEED)
29     else:
30         c.move(ball, -ball_left + PADD_W, BALL_Y_SPEED)
31
32     # pantainya horizontal
33     if ball_top < BALL_Y_SPEED < 0 or ball_bot > BALL_Y_SPEED > HEIGHT:
34         bounce("horizontal")
35
36 # membuat variabel global untuk kecepatan racket
37 left_PADD_SPEED = 0
38 right_PADD_SPEED = 0
39
40 # parameter fungsi untuk menggerakan racket
41
42 def move_padd():
43     global left_PADD_SPEED, right_PADD_SPEED
44     left_PADD_TOP = c.coords(left_PADD)[1]
45     left_PADD_BOTTOM = c.coords(left_PADD)[1]
46     right_PADD_TOP = c.coords(right_PADD)[1]
47     right_PADD_BOTTOM = c.coords(right_PADD)[1]
48
49     if left_PADD_TOP < left_PADD_SPEED < 0 and left_PADD_BOTTOM < left_PADD_SPEED < HEIGHT:
50         c.move(left_PADD, 0, left_PADD_SPEED)
51
52     if right_PADD_TOP < right_PADD_SPEED < 0 and right_PADD_BOTTOM < right_PADD_SPEED < HEIGHT:
53         c.move(right_PADD, 0, right_PADD_SPEED)
54
55 def movement_handler(event):
56     global left_PADD_SPEED, right_PADD_SPEED
57     if event.keysym == "w":
58         left_PADD_SPEED = -PADD_SPEED
59     elif event.keysym == "s":
60         left_PADD_SPEED = PADD_SPEED
61     elif event.keysym == "W":
62         right_PADD_SPEED = -PADD_SPEED
63     elif event.keysym == "S":
64         right_PADD_SPEED = PADD_SPEED
65
66 def stop_paddle(event):
67     global left_PADD_SPEED, right_PADD_SPEED
68     if event.keysym in ["w", "s", "W", "S"]:
69         left_PADD_SPEED = 0
70         right_PADD_SPEED = 0
71
72 def quit_game(event):
73     root.destroy()
74
75 def main():
76     move_ball()
77     move_padd()
78     root.after(50, main)
79
80 # memanggil fungsi utama untuk memulai permainan
81 main()
82
83 # membuat fungsi-fungsi lain ke canvas
84 c.bind("keypress", movement_handler)
85 c.bind("keyrelease", stop_padd)
86 c.bind("keypress", quit_game)
87
88 # menjalankan jendela
89 c.focus_set()
90 root.mainloop()
```

- Hasil



- Link Github

https://github.com/ihsnmilna/TugasKecerdasanBuatan_Sesi14/tree/main