Snake_game.py ×

Snake_game.py > ...

```python
from turtle import Screen
import cv2
from cv2 import resize
import pygame
import mediapipe as mp
import random

pygame.init()
Red = (255,0,0)
Green = (0,255,0)
Blue = (0,0,255)
hybrid = (23,234,223)

#window
Snake_Window = pygame.display.set_mode([600,600])
pygame.display.set_caption("Snake Game")

#update
pygame.display.update()

#Clock
Clock = pygame.time.Clock()

#Score font
score_font = pygame.font.SysFont(None, 30)
```

Snake_game.py ×

Snake_game.py > ...

```python
#Write Score function
def write_score(text,color,x,y):
    screen_text = score_font.render(text,True,color)
    Snake_Window.blit(screen_text,[x,y])

#create body function
def create_body(Snake_Window,color,Snake_list,Snake_Size):
    for x,y in Snake_list:
        pygame.draw.rect(Snake_Window,color,[x,y,Snake_Size,Snake_Size])

#Game loop function
def Game_loop():
    game_exit = False
    game_over = False
    Snake_head_x = 100
    Snake_head_y = 100
    Score = 0
    Food_x = random.randint(2,19)*30
    Food_y = random.randint(2,19)*30
    Speed_x = 0
    Speed_y = 0
    Snake_Size = 30
    Snake_list = []
    Snake_length = 1
    fps = 60
```

```
52      cam = cv2.VideoCapture(0)
53      mphand = mp.solutions.hands
54      mpHands = mphand.Hands()
55      mpDraw = mp.solutions.drawing_utils
56      while not game_exit:
57          if game_over==True:
58              Snake_Window.fill(Red)
59              write_score("Game Over, Press Space to play again Your Score:" + str(Score),Green,10,100)
60              pygame.display.update()
61              for event in pygame.event.get():
62                  if event.type == pygame.QUIT:
63                      game_exit == True
64                  if event.type == pygame.KEYDOWN:
65                      if event.key == pygame.K_SPACE:
66                          Game_loop()
67          else:
68              success,img = cam.read()
69              image = cv2.resize(img,(600,600))
70              output = mpHands.process(image)
71              if output.multi_hand_landmarks:
72                  for landmarks in output.multi_hand_landmarks:
73                      for id , pos in enumerate(landmarks.landmark):
74                          height,width,z = image.shape
75                          pos_x,pos_y = int(width*(pos.x)), int(height*(pos.y))
76                          if id == 8:
77                              cv2.circle(image, (300,300), 500, Blue, cv2.FILLED)
```

```
77                  cv2.circle(image, (300,300), 500, Blue, cv2.FILLED)
78                  cv2.circle(image, (300,300), 30, Red, cv2.FILLED)
79                  cv2.circle(image, (pos_x,pos_y), 25, hybrid, cv2.FILLED)
80                  if pos_x > 350 and pos_y < 350 and pos_y > 250:
81                      Speed_x = -3
82                      Speed_y = 0
83                  if pos_x < 250 and pos_y < 350 and pos_y > 250:
84                      Speed_x = 3
85                      Speed_y = 0
86                  if pos_y < 250 and pos_x < 350 and pos_x > 250:
87                      Speed_y = -3
88                      Speed_x = 0
89                  if pos_y > 350 and pos_x < 350 and pos_x > 250:
90                      Speed_y = 3
91                      Speed_x = 0
92              for event in pygame.event.get():
93                  if event.type == pygame.QUIT:
94                      game_exit == True
95              if abs(Snake_head_x-Food_x) < 30 and abs(Snake_head_y-Food_y) < 30:
96                  Food_x = random.randint(2,19)*30
97                  Food_y = random.randint(2,19)*30
98                  Score += 5
99                  Snake_length += 5
100             Snake_head_x += Speed_x
101             Snake_head_y += Speed_y
102             Snake Window.fill(Blue)
```

```
102         Snake_Window.fill(Blue)
103         write_score('Score: ' + str(Score),Red, 10, 10)
104         body = []
105         body.append(Snake_head_x)
106         body.append(Snake_head_y)
107         Snake_list.append(body)
108         if Snake_head_x < 0 or Snake_head_x > 600 or Snake_head_y < 50 or Snake_head_y > 600:
109             game_over = True
110         if Snake_length < len(Snake_list):
111             del Snake_list[0]
112         if body in Snake_list[:-1]:
113             game_over = True
114         create_body(Snake_Window, Red, Snake_list, Snake_Size)
115         pygame.draw.rect(Snake_Window, hybrid, [0, 0, 600, 50])
116         pygame.draw.rect(Snake_Window, Green, [Food_x, Food_y, Snake_Size, Snake_Size])
117         pygame.draw.rect(Snake_Window, Red, [Snake_head_x, Snake_head_y, Snake_Size, Snake_Size])
118         pygame.display.update()
119         cv2.imshow("Snake_game", image)
120         cv2.waitKey(1)
121         Clock.tick(fps)
122 Game_loop()
123
124
125
```

ile Edit Selection View Go Run Terminal Help Snake_game.py - Gesture Control Snake Game - Visual Studio Code

EXPLORER ··· Snake_game.py ×

GESTU

Snake_game

Snake Game

(Scor

e_hea

_list

ed, S
ow, h
ow, G

nReadS
v-pyth
sync R
v-pyth
nReadS
v-pyth
sync R
v-pyth

e'::SourceReaderCB::OnReadSample videoio(MSMF): OnReadSample() is called with error status: -1072873821
[WARN:4@194.459] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (476) `anonymous-namespac
e'::SourceReaderCB::OnReadSample videoio(MSMF): async ReadSample() call is failed with error status: -1072873821

OUTLINE

TIMELINE