Task 12

Simulate Gaming Concepts Using Pygame

Aim: To simulate gaming concepts using Pygame

Algorithm:

1. import pygame package and initialize it
2. Define the window size and title
3. Create a snake class which initialize the snake position. color and movement
4. Create a fruit class
5. Create a function to check if the snakes collides
6. Create a function to check if the snake collides with window

Program:

```
# importing libraries
    import pygame
    import time
    import random
    snake - speed = 15

    window = x = 720
    window = y = 480

     black = pygame.color(t00 0)
     white = pygame.color(225, 255
red = Pygame.Color [225, 0, 0]
green = Pygame.Colour [0, 255, 0)
blue = Pygame.Colour (0, 0, 255)
Pygame.int()
   Pygame.display.set.caption ('greeks for greeks snakes')
   game_window = Pygame.display set_mode((window_x, window_y)
    FPS = Pygame.time.clock()
    Snake_Position = [100, 50]
    Snake_body = [100, 50]
                  [90, 50],
                  [80, 50]
                  [70, 50]
fruit - Position = [random. and range (1, window_x//10)) *10
                    random.rand range (1, (window_x//10))*10]
fruit - spawn = True
Direction = 'Right'
change_to = direction
 Score = 0
 def show - score(choice, Colour, Font, size);
 Score-font = Pygame.font. sys font(font, size)
  Score_Surface = Score -font.render ('score' + str(score)
  Score - rect = Score -Surface .get -rect()
  game -window .bilt (score _Surface, Score - rect)
   def .game - over();
```

```
my_font = Pygame.font. SysFont('times new roman'. 50
 game-over-surface = my-font render(
        'your score is :' + str(score), True, red)
  game-over-rect = game-over-surface get-rect()
  game-over-rect = game-over-surface get. rect()
  game-over-rect.midtop = (window-X/2, window-Y/4)
   game-window.blit(game-over-surface, game-over-rect)
 time.sleep(2)
  Pygame.quit()
  while True:
   for event it Pygame.event.get():
        If event.type == Pygame.Key down:
   If event.Key == Pygame.K-up:
            change-to = 'up'
       If event Key == Pygame.K-Down
             change-to = 'down'
         If event Key == Pygame.K-left:
              change to = 'left'
           If event Key == Pygame.K-Right:
               change-to = 'Right'
 If change-to == 'up' and direction != 'down':
        direction = 'up'
 If change-to == 'down' and Directions != 'up':
         direction = 'left'
```

```
change =to ==' Right' and Direction! = 'left'.
              direction = 'Right'
If direction == 'up'
      Snake - Position [1]-=10
   If direction == 'down'
         Snake - Position [1] +=10
      If direction==: 'left'!
            Snake - Position [0] -=10
       If direction == 'Right'
               Snake -Position [0] +=10
Snake -body . insert (0, list (snake - Position))
        Score +=10
        Fruit - - Spawn = False
else
    Snake -body . PoP()
    If not fruit - spawn;
        fruit - Position = [random. randrange (1, (window - x)) 10)
  Fruit - Spawn = True
  game - window . fill (black)
    for Pos in snake - body;
  Pygame - draw. rect (game - window, green, Pygame Rect (Post(0), Pos(1))
  Py game . draw. rect (game - window, white, Pygame . Rect
          fruit - Position [0], fruit - Position [1], 10,10]
  show - score (), white, 'times new roman', 20)
```

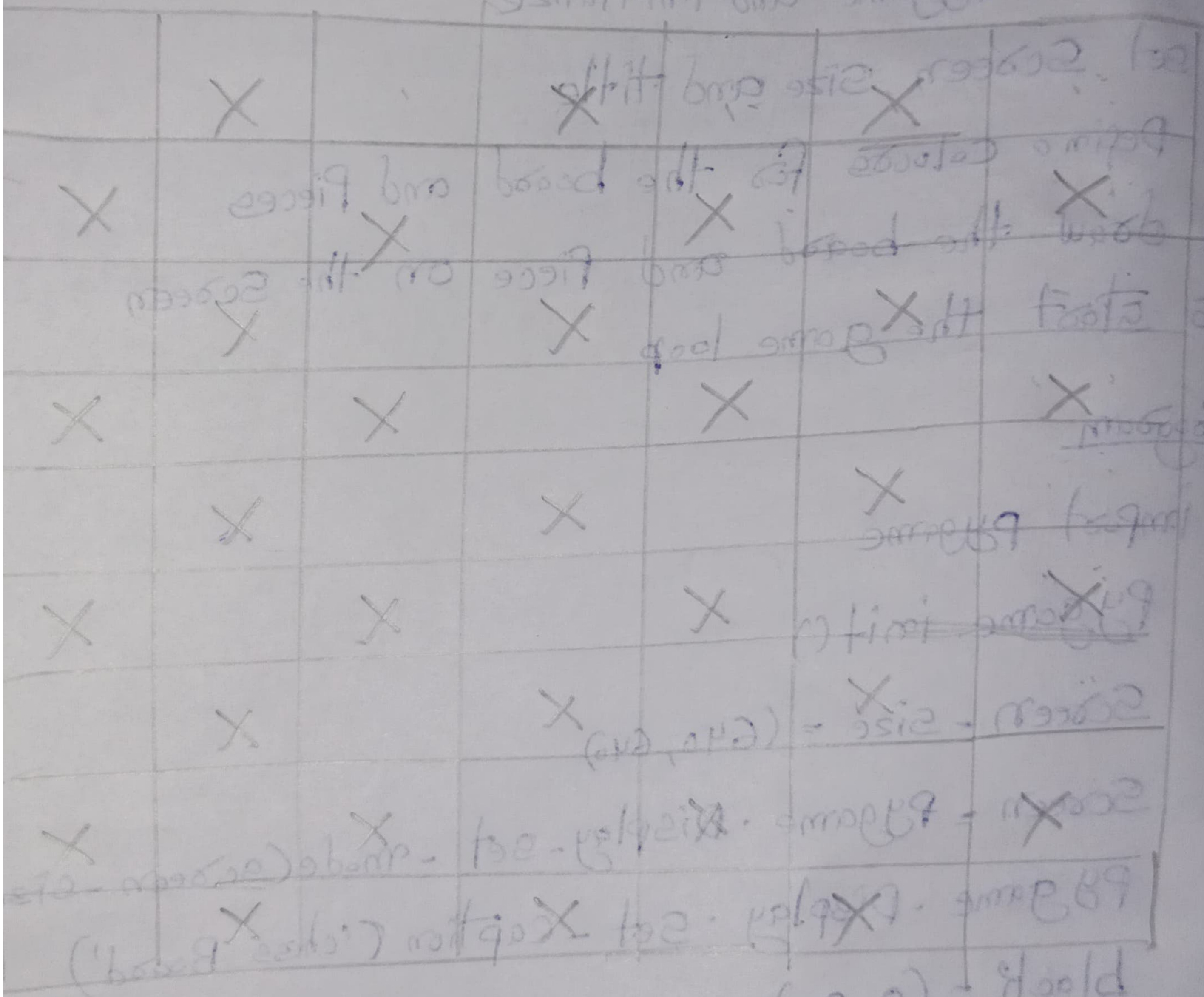
Problem 12.2 : write a Python to develop a chess board using Py game.

Algorithm

1. import Pygame and initialize
2 set Screen size and title
3. Define colours for the board and Pieces
4. draw the board and Piece on the Screen
5. Start the game loop

Program

```
import pygame
  Pygame. init()
  Screen - size = (640, 640)
  Screen = Pygame . Display - set - mode(screen - size)
  Py game . Display . set - caption ('chess Board')
  black = (0, 0, 0)
  white = (255, 255, 255)
  brown = (153, 760)
  Def draw - board()'
        for row in range(8):
            for col in range (8) :
```



X No.	
ERFORMANC	
SULT AND ANA	
VA VOCE (3)	
CORD (4)	
OTAL (15)	
GN WITH DATE	

Result: Hence writing a Python Program to develop a chess board using python Program is done succesfully.