|  |
| --- |
| **<<SNAKE GAME>>**  **21CSS101J – PROGRAMMING FOR PROBLEM SOLVING**  **Mini Project Report**  *Submitted by*  **AKSHAT MITTAL [RA2211003010790]**  **B.Tech. CSE - <<CORE>>**  **RITVIK RAJVANSHI [RA2211003010792]**  **B.Tech. CSE - <<CORE>>**  SRMIST-01.jpg  **SCHOOL OF COMPUTING**  **COLLEGE OF ENGINEERING AND TECHNOLOGY**  **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  **(Under Section 3 of UGC Act, 1956)**  S.R.M. NAGAR, KATTANKULATHUR – 603 203  KANCHEEPURAM DISTRICT  **December 2022** |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Problem Statement | 2 |
| 2 | Methodology / Procedure | 3-7 |
| 3 | Coding (C or Python) | 8-9 |
| 4 | Results | 9-10 |
| 5 | Conclusion | 11 |

# **INTRODUCTION: -**

The project is a game written in ‘PYTHON’ based on the game called ‘snake’ which has been around since the earliest days of home computing and has re-emerged in recent years on mobile phones.

Snake is the common name for a video game concept where the player maneuvers a line which grows in length, with the line itself being a primary obstacle. The originated in the 1976 ‘arcade game blockade’, and the ease of platform. After a variant was preloaded on “NOKIA MOBILE PHONE” IN 1998, there was a resurgence of interest in the snake concept as it found a large audience.

It Is not the world’s greatest game, but it does give you an idea of what you can achieve with a relatively simple swing program, and perhaps the basis by which to extend the principles and create more interesting games of your own.

PROBLEM STATEMENT: -

To build the snake game project we used the turtle module, random module, time module, and concept of python. Time module is an inbuilt module in python. It provides the functionality of time

1. Importing required module
2. Creating game screen
3. Creating snake and food
4. Keyboard binding screen.Listen() function listen when key will press.

Goals in creation of snake game:-

There were five primary goals in the creation of the python language.

1. It should be “simple object oriented.”
2. It should be “robust and secure.”
3. It should be “architecture neutral and portable.”
4. It should execute with “high performance”.
5. It should be “interpreted, threaded Software specifications: -

* OPERATING SYSTERM WINDOWS 10
* CODING LANGUAGE-PYTHON

## Methodology / Procedure: -

### **creating a Snake game in python using the following:**

* Turtle – It is a pre-installed library in python which is used for creating shapes, picture, and game.
* Time – It is used for counting the number of seconds elapsed since the epoch.
* Random– This module is used to generate random numbers in python using the random module.
* penup () – It stops drawing of the turtle pen.
* speed() – It is an integer value in the range 0 to 10. So, 0 is fastest, 10 is fast, 6 is normal, 3 is slow, and 1 is slowest. If no argument is given, returns the current speed.
* color() – It returns or set pen color and fill color.
* shape() – It set turtle shape to the shape of a given name.
* hideturtle() – It makes the turtle invisible.
* xcor() – Return the turtle’s x coordinate.
* ycor() – Return the turtle’s y coordinate.

## we will explain the easy way to code the snake game in python. It is recommended to go throw the below step.

Step 1:

* Firstly, we will import all the modules into the program, and we will give the default value for the game.

Step 2:

* Now, we will create the window screen for the game, and also we will create the head of the snake and food for the snake. The score will be displayed at the header of the game.
* The function turtle.Screen() is used to create a window. In this code, our window is “wn” for the game.
* We have to give the window a name with the function “wn.title(“Snake Game”)”.
* To set the background color for the window we have

used “wn.bgcolor(‘black’)”. Set the window height and width with the function “wn.setup(width=X, height=Y)”. Here, width=600 and height=600.

* The function window.tracer(0) turns off the screen updates. As, we do not need any screen updates other than the scoreboard, so it is set to 0.
* Now, we will create a snakehead, it is basically a turtle which will be a snake and it will move around.
* For creating a turtle we will use “turtle. Turtle()” and assign the name head. The head speed is set to 0 as we are just initializing and the head does not need to move.
* Let us initialize the head shape and color by

using “head.shape(“circle”)” and “head.color(“green”)”.

* The function “head.penup()” makes sure that the path taken by the snake is not drawn.
* The “head.goto(0,0)” is used for snake position to be the center of the window and the direction to stop we will use head.direction = “stop”.
* And “pen.write()” function is used to write the text at the current turtle position.
* body every time it touches food. So, we used arrays for this. We create an array called segments, which is initialized empty

Step 3:

* Now, we need to define a function for each of these directions and set the head.direction to up, down, left, and right.
* After that, we will go ahead and make the snake move. So, we will define a function called move ().
* If the head goes up, the “y” coordinate is increased, if the head goes down, the “y” coordinate decreases.
* If the head moves right, the “x” coordinate increases and if the head moves

left, the “x” coordinate decreases.

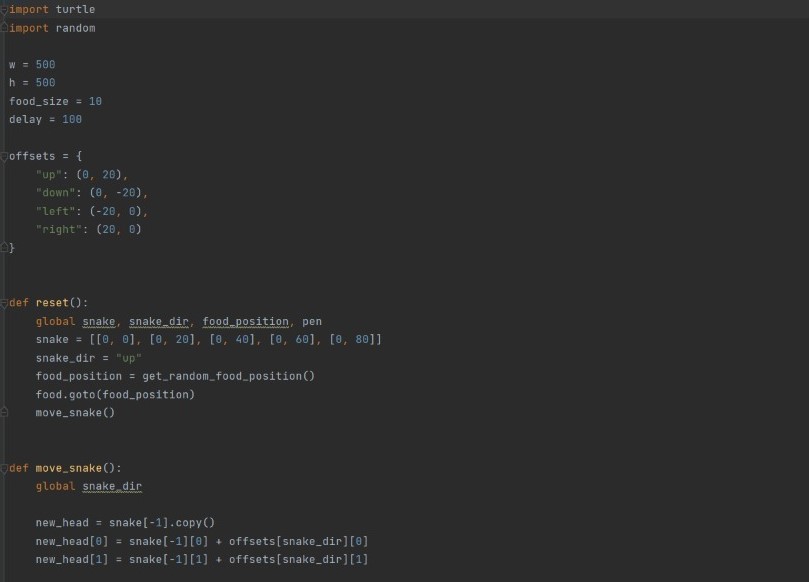
Step 4:

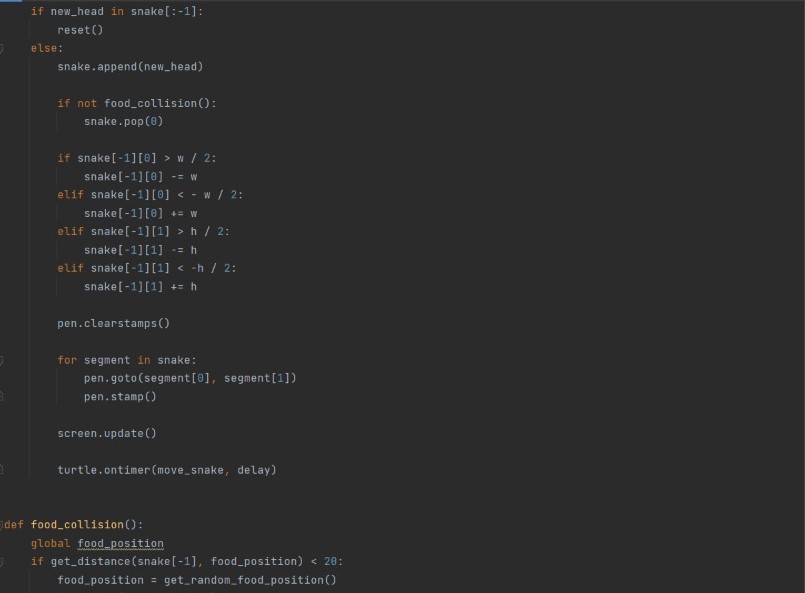
* We will assign a key to the snake movements. By clicking the keywords, we can move the snake up, down, left, and right direction.
* We need the system to listen to our control keypress, so we will add a function called wn.listen() that listens to the key pressed.
* Every keypress needs to be bound to a function that carries out an action. We will use the function ” wn.onkeypress(function, “key”) “ for all four. Here, I have used “y” for up, “h” for down, “g” for left, and “j” for right.
* Now, we can operate the movement of a snake on the screen.

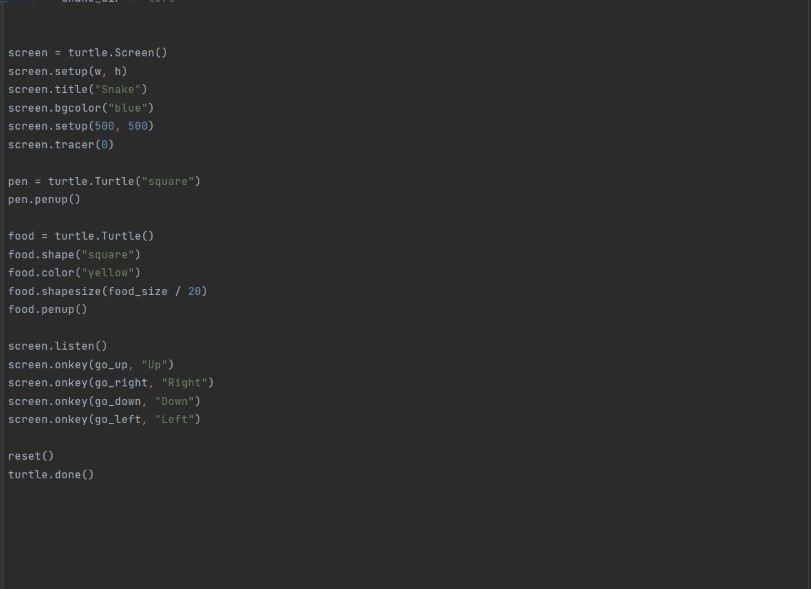
Step5:

* So, now the function does not hing until it’s called. We need to call the function every time we update the screen or the window.
* We have to be sure that the snake die when it collides with the border. We already have the coordinate of the border. So, we just need to reset the snake head position when it touches the coordinates.
* Also, the snake needs to stop moving and hence change the direction to stop.
* To slow down the snake movement we need to use the time module otherwise the default behavior for the move function is very fast.
* So, we will use the function time.sleep() to reduce turtle speed.
* The segment needs to disappear when the snake dies.
* So, now we need to set the position of these segment outside the window coordinates. The game restarts and hence clear the segment list.
* We need to adda segment to the snake body every time it touches the food. So, we have
* The condition that checks for the head ’s collision with food.
* Create a new segment, define its speed, shape, and color and appendit to segments array.
* Now, adding the segment to the snake head is not enough. These segments need to move when the snakehead moves.
* To move the last segment which is in the position x to x-1 and x-1 to x-2 and soon.
* The snake need to die if it touches itself. So, we are going to check if the distance between the segment and head is less than 20. If it is, reset the head position and head direction.
* At last, we need to see the situation when the score increases. The first one is when the head collides with the food. Increase the score and update the high\_score.
* We used pen.write() function to write the score on the screen.
* We need to reset the score when the snakehead collides with the border and with its own tail
* And then call the function time.sleep(delay) to reduce turtle speed.

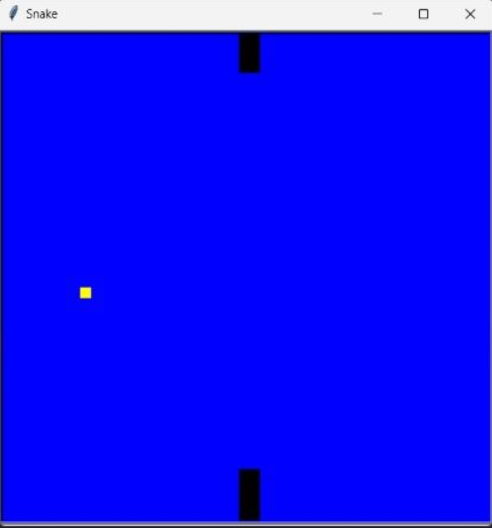
## Complete code for snake game in python using turtle: -

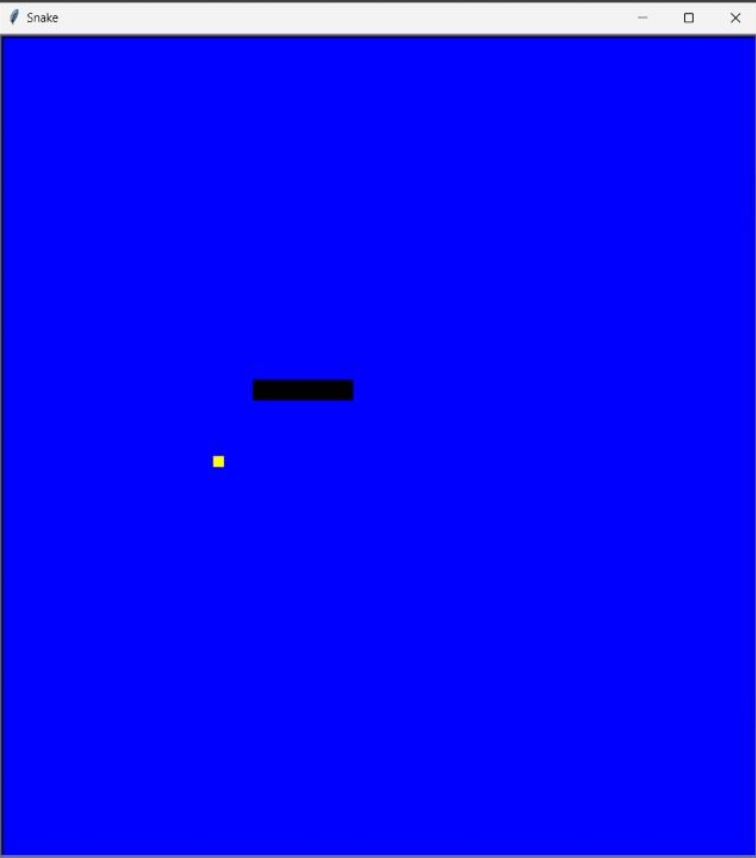


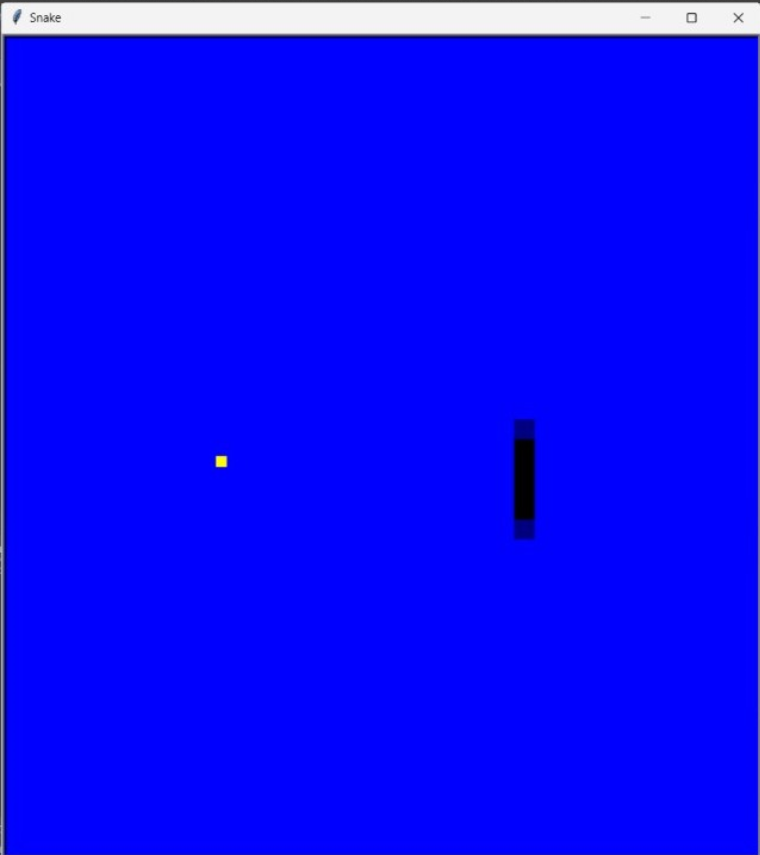




RESULT:-







CONCLUSION: -

In this game, player is to catch the maximum number of fruits without hitting wall or itself. Creating a snake game has given us a lot of challenges. The coding of snake was extremely difficult with many errors arising. Many systems had to be written numerous ways before a final working solution was found.

THANK YOU