

VIVEKANANDHA COLLEGE OF ENGINEERING FOR WOMEN [AUTONOMOUS]



Tiruchengode-637 205 Namakkal(DT)

SNAKE GAME MINI PROJECT

Project Report was Submitted in Partial fulfillment of the
Requirement for the

Department Of Computer Science Engineering

UNDER THE GUIDANCE OF

T.KAVITHA ,AP(CSE)

SUBMITTED BY

NAME	REGISTER NUMBER
<i>Salma Aaysha.M</i>	<i>612922101100</i>
<i>Varthini.A</i>	<i>612922101126</i>
<i>Rakshana firthouse.A</i>	<i>612922101093</i>
<i>Yashini.R</i>	<i>612922101126</i>
<i>Suba .M</i>	<i>612922101115</i>
<i>Pradeepa .M</i>	<i>612922101081</i>

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Mrs.T.KAVITHA/AP/CSE who gave me the golden opportunity to do this wonderful project of snake game who game me guidelines for this project. I am really thankful to them.

INDEX

Table of Contents

ABSTRACT :	4
INTRODUCTION :	5
SOFTWARE :	7
PROJECT REQUIREMENTS:-	7
1. Importing required module.....	7
2. Creating game screen	8
3. Creating snake and food	8
4. Keyboard binding	8
5. Snake and fruit collision	9
6. Snake and border collision	9
7. When snake touch itself	9
PROJECT CODING :	10
PROJECT OUTPUT:.....	16
CONCLUSION :	17
FUTURE IMPLEMENTATIONS :.....	18
REFERENCE:	19

SNAKE GAME

ABSTRACT :

The python snake game is retro style implemented using python programming language. It aims to recreate the nostalgic gaming experience of popular snake game which gained popularity in late 1970s and early 1980. In this game players control a snake moves across a rectangular grid aiming to eat food items that appears randomly. As the snake consumes food, it grows. Larger making it increasingly challenging to navigate with colliding with the walls or its own tail. The objective is to achieve the highest possible score by eating much food. The python snake game utilizes core programming concepts such as game loops, event handling collision detecting and graphical rendering it leverages pygame modules, to provide a user friendly interface and handle graphics, sound and keyboard input.

INTRODUCTION :

– The objective of this python project is to build a snake game project. In this python project, the player has to move a snake so it touches the fruit. The player controls a long, thin creature, resembling a snake, which roams around on a blast bordered plane, picking up food (or some other item), trying to avoid hitting its own tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, making the game increasingly. If the snake touches itself or the border of the game then the game will over.

ABOUT SNAKE GAME:

Snake game was invented by Taneli Armanto in the year 1997 at that time he was a design engineer working in NOKIA . The concept of this game was taken from the arcade game called 'Blockade'. This game was first launched in NOKIA 6110 mobile phone. More about the Technical aspects of the game will be talked further in this report.

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. After a variant was preloaded on Nokia mobile phones in 1998, there was a resurgence of interest in the snake concept as it found a larger audience.

In 1996, *Next Generation* ranked it number 41 on their "Top 100 Games of All Time", citing the need for both quick reactions and

forethought. In lieu of a title they listed it as "***Snake game***" in quotes.

On November 29, 2012, the Museum of Modern Art in New York City announced that the Nokia port of Snake was one of 40 games that the curators wished to add to the museum's collection in the future.

SOFTWARE :

PROJECT REQUIREMENTS:-

To build the snake game project we used the turtle module, random module, time module, and concept of python.

Turtle module gives us a feature to draw on a drawing board

Random module will be used to generate random numbers

To install python modules we use the pip install command in the command line:

```
pip install turtle
```

```
pip install random
```

We will be creating a **Python-based-game** using the following modules:

- **Turtle:** It is a pre-installed python library that enables users to create shapes and pictures by providing them with a virtual canvas.
- **Random:** This function is used to generate random numbers in Python by using *random* module.

1. Importing required module

```
import turtle  
import random
```

We require turtle, random module to import.

2. Creating game screen

- **title()** will set the desired title of the screen
- **setup()** used to set the height and width of the screen
- **tracer(0)** will turn off the screen update
- **bgcolor()** will set the background color
- **forward()** will use to move the turtle in a forwarding direction for a specified amount.
- **right()** used to turn the turtle clockwise
- **left()** used to turn the turtle anticlockwise
- **penup()** will not draw while its move

3. Creating snake and food

- **turtle()** will be used to create a new turtle object
- **hideturtle()** will use to hide the turtle
- **goto()** used to move the turtle at x and y coordinates

4. Keyboard binding

screen.listen() function listen when key will press.

If the Up key will press then the snake will move in up direction.

If the Down key is pressed then the snake will move in the down direction.

If Left key will press then the snake will move in left direction.

If the Right key will press then the snake will move in the right direction

5. Snake and fruit collision

If the snake touches the fruit then the fruit will go at any random position and score will increase and the size of the snake will also increase.

6. Snake and border collision

If the snake touches the border of the game then the game will over.

screen.clear() will delete all the drawing of the turtle on the screen.

7. When snake touch itself

When the snake touch itself, the game will be over. It starts again a new game with small snake. The game plays in vice versa.

PROJECT CODING :

```
import turtle
```

```
import random
```

```
w = 800
```

```
h= 500
```

```
food_size = 15
```

```
delay = 100
```

```
offsets = {
```

```
    "up": (0, 20),
```

```
    "down": (0, -20),
```

```
    "left": (-20, 0),
```

```
    "right": (20, 0)
```

```
}
```

```
def reset():
```

```
    global snake, snake_dir, food_position, pen
```

```
    snake = [[0, 0], [0, 20], [0, 40], [0, 60], [0, 80]]
```

```
snake_dir = "up"  
food_position = get_random_food_position()  
food.goto(food_position)  
move_snake()
```

```
def move_snake():  
    global snake_dir  
  
    new_head = snake[-1].copy()  
    new_head[0] = snake[-1][0] + offsets[snake_dir][0]  
    new_head[1] = snake[-1][1] + offsets[snake_dir][1]  
  
    if new_head in snake[:-1]:  
        reset()  
    else:  
        snake.append(new_head)  
  
    if not food_collision():  
        snake.pop(0)
```

```
if snake[-1][0] > w / 2:  
    snake[-1][0] -= w  
elif snake[-1][0] < - w / 2:  
    snake[-1][0] += w  
elif snake[-1][1] > h / 2:  
    snake[-1][1] -= h  
elif snake[-1][1] < -h / 2:  
    snake[-1][1] += h
```

```
pen.clearstamps()
```

```
for segment in snake:  
    pen.goto(segment[0], segment[1])  
    pen.stamp()
```

```
screen.update()
```

```
turtle.ontimer(move_snake, delay)
```

```

def food_collision():
    global food_position
    if get_distance(snake[-1], food_position) < 20:
        food_position = get_random_food_position()
        food.goto(food_position)
        return True
    return False

def get_random_food_position():
    X = random.randint(- w / 2 + food_size, w / 2 - food_size)
    Y = random.randint(- h / 2 + food_size, h / 2 - food_size)
    return (x, y)

def get_distance(pos1, pos2):
    X1, y1 = pos1
    X2, y2 = pos2
    distance = ((y2 - y1) ** 2 + (x2 - x1) ** 2) ** 0.5
    return distance

def go_up():
    global snake_dir
    if snake_dir != "down":

```

```
snake_dir = "up"
```

```
def go_right():  
    global snake_dir  
    if snake_dir != "left":  
        snake_dir = "right"
```

```
def go_down():  
    global snake_dir  
    if snake_dir != "up":  
        snake_dir = "down"
```

```
def go_left():  
    global snake_dir  
    if snake_dir != "right":  
        snake_dir = "left"
```

```
screen = turtle.Screen()  
screen.setup(w, h)  
screen.title("SNAKE GAME")  
screen.bgcolor("violet")
```

```
screen.setup(500, 500)
```

```
screen.tracer(0)
```

```
pen = turtle.Turtle("square")
```

```
pen.penup()
```

```
food = turtle.Turtle()
```

```
food.shape("circle")
```

```
food.color("black")
```

```
food.shapesize(food_size / 20)
```

```
food.penup()
```

```
screen.listen()
```

```
screen.onkey(go_up, "Up")
```

```
screen.onkey(go_right, "Right")
```

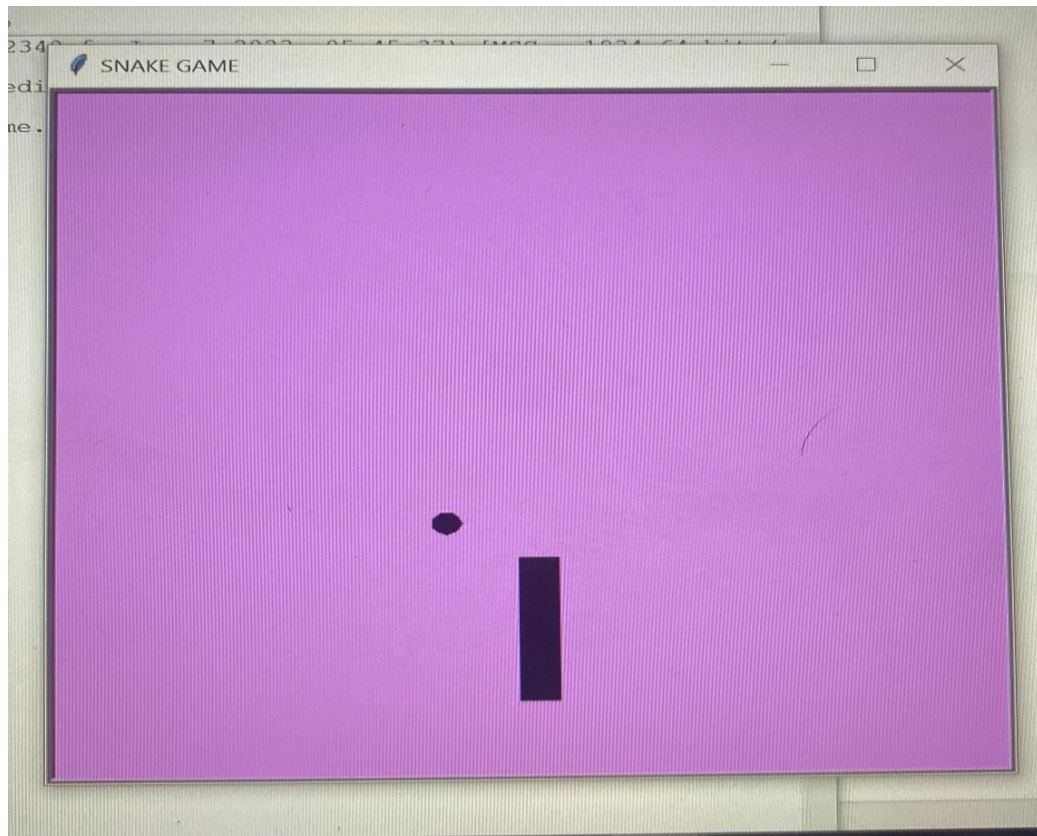
```
screen.onkey(go_down, "Down")
```

```
screen.onkey(go_left, "Left")
```

```
reset()
```

```
turtle.done()
```

PROJECT OUTPUT:



CONCLUSION :

We successfully developed Snake game project in python. We learn how to used turtle modules and draw on the screen using a turtle. We also learn about random module.

It is our teams hope that this document will be of huge help with understanding of our projects we have used an approach which has proved beneficial in terms of our learning this language.

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. it is recommended that anyone who wishes to recreate this game starts simply when writing the code. It is advisable that they first perfect the snake movement controls before messing with the food generation. By taking the code in small sections, it is easier to get individual features to work. Building off this, use functions to contain each aspect of the game. Using functions made it easier to determine where errors were occurring when debugging the code. It also kept the code more organized.

FUTURE IMPLEMENTATIONS :

Our project will be able to implement in future after making some changes and modification as make our project at a very low level. So the modification that can be done in our project are:-

- ✓ Add player profile
- ✓ Add multiplayer option
- ✓ Display the scores between two players
- ✓ Build the barriers with different levels

And so many modifications I would like to add up on this project. In future, I will modify this project with better gaming experience.

REFERENCES:

- Geeks for geeks-<https://www.google.com/url?sa=t&rctj&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiRr8u7gK70AhW0IbcAHWJIDEoQFnoECBkQAAQ&url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fcreate-a-snake-game-using-turtle-in-python%2F&usg=AOvVaw3xuwT-TDPpBFVAUzoc5Sap>
- <https://chat.openai.com/#>
- <https://www.studocu.com/in>