MINI PROJECT– II

SYNOPSIS

Department of Computer Science & Application

Institute of Engineering & Technology

SUBMITTED TO: -

Mr. Manoj Varshney

SUBMITTED BY:

Puroo Kulshrestha

(201500535)

Acknowledgement

It gives us a great sense of pleasure to present the synopsis of the B.TECH mini project undertaken during B.TECH III Year. This project is going to be an acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals. We owe special debt of gratitude to MS Madhu, Technical Trainer , for providing us with an encouraging platform to develop this project, whichthus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work.

His sincerity, thoroughness and perseverance has been a constant source of inspiration for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

Puroo Kulshrestha (201500535

ABSTRACT

Snake game is a classic video game that has been enjoyed by people of all ages since its inception in the late 1970s. It is a simple game that involves controlling a snake that moves around the screen and eats food. As the snake eats food, it grows longer, and the player's objective is to prevent the snake from colliding with the walls or its own body. In this abstract, we will discuss the development of a snake game using Python programming language.

The snake game was developed using the Pygame library in Python. Pygame is a set of Python modules designed for writing video games. It provides functionality for handling graphics, sound, input, and other game-related tasks. The game was developed by creating a Pygame window and adding a snake and food sprite. The snake was made up of a series of rectangles, and the food was a single rectangle. The game starts by initializing the Pygame library and setting up the window. The snake and food sprites are then created and placed on the screen. The game loop is then started, which is responsible for updating the game state and drawing the screen. The game loop works by continually updating the position of the snake and checking for collisions with the walls and food. If the snake collides with the food, the snake's length is increased, and a new food sprite is randomly placed on the screen. If the snake collides with the wall or its own body, the game ends. The game also includes a score counter that increases as the snake eats food. The score is displayed on the screen along with the game over message when the game ends.

The snake game in Python was developed using object-oriented programming (OOP) principles. The game was broken down into several classes, including the Game class, Snake class, and Food class. Each class had its own set of methods and attributes that were used to implement the game's functionality. The Game class was responsible for managing the game loop and updating the game state. It also handled user input and displayed the score and game over message. The Snake class represented the snake sprite and was responsible for updating the snake's position and checking for collisions. It also handled the snake's movement and growth. The Food class represented the food sprite and was responsible for randomly placing new food on the screen.

Overall, the snake game in Python was a fun and challenging project that helped to reinforce fundamental programming concepts such as OOP, game logic, and Pygame functionality. The completed game was a great example of how Python can be used to develop simple games and applications. With some modifications, the game could be expanded to include additional features such as different game modes, power-ups, and obstacles.

Contents

INTRODUCTION

The snake game is a classic video game that has been popular since the late 1970s. The game involves controlling a snake that moves around a playing area, with the objective of eating food to grow longer while avoiding obstacles such as walls and the snake's own tail. In this article, we will introduce how to build a basic snake game in Python.

To start building the snake game, we will need to use the Pygame library, which is a set of Python modules designed for writing video games. Pygame provides a simple way to create and handle graphics and user inputs. First, we will need to set up a game window using Pygame. This can be done by importing the Pygame module, initializing it, and creating a screen object. We will also need to set the dimensions of the game window. Next, we will define the snake and its movement. The snake will be represented by a series of rectangles, and its movement will be controlled using arrow keys. We will also define the position of the food, which the snake will need to eat to grow. Finally, we will add a score system that keeps track of how many pieces of food the snake has eaten. The game will end when the snake collides with an obstacle or its own tail, and the player's score will be displayed.

In conclusion, building a snake game in Python using Pygame is a fun and educational project for anyone interested in learning about game development. By following these basic steps, you can create a simple yet engaging game that will provide hours of entertainment.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements :

- Python programming language (version 3.x)
- Pygame library (version 1.9.x or higher)

Hardware Requirements :

- A computer with a modern CPU and GPU
- A monitor with a minimum resolution of 800x600 pixels
- A keyboard or other input device for controlling the game

PROJECT DESCRIPTION

The Snake game is a classic arcade game that has been popular for decades. The game consists of a player-controlled snake that must eat food and grow in length without hitting any walls or its own tail. In this project, we will create a Snake game using Python programming language.

Gameplay: The game begins with a snake consisting of a single square on the screen. The snake will move continuously in one direction, and the player must change its direction to make it move towards the food. The food will appear randomly on the screen, and the snake must eat it to grow in length. As the snake grows longer, it becomes more difficult to avoid hitting walls or its own tail. If the snake hits any of these, the game is over.

Game Design: We can use Pygame library to create the Snake game. Pygame is a set of Python modules designed for writing video games. We will use the following steps to design the Snake game:

Step 1: Initialize Pygame and the game window We will use the Pygame library to create the game window and initialize the game environment. We will also set the title of the window and its size.

Step 2: Create the Snake We will create the Snake using a list of squares. The Snake will move by adding a new square to the front of the list and removing the last square. We will also set the initial direction of the Snake.

Step 3: Create the Food We will create the Food using a random coordinate generator. The Food will appear at a random location on the screen, and the Snake must move towards it to eat it.

Step 4: Check for Collision We will check for collisions between the Snake and the Food, as well as the Snake and the walls or its own tail. If the Snake hits any of these, the game is over.

Step 5: Update the Screen We will update the screen by drawing the Snake and the Food on the screen. We will also display the score and the game over message when the game ends.

Step 6: Handle User Input We will handle user input to change the direction of the Snake. We will use the arrow keys to change the direction of the Snake.

Conclusion: In conclusion, we have designed a simple Snake game using Python programming language and Pygame library. The game consists of a player-controlled Snake that must eat food and grow in length without hitting any walls or its own tail. The game is a good way to learn about game programming and can be extended to add more features and complexity.

WORKING

The classic snake game is a simple game that can be implemented using Python programming language. Here is a basic overview of how the game can be implemented:

1. **Import the necessary modules:**
   To implement the game, we need to import the pygame, random and time module which is used for building games in Python.

2. **Initialize the game:**
   To initialize the game, we need to set up the screen size, background color, and caption.

```
pygame.init()
white = (255, 255, 255)
yellow = (255, 255, 102)
black = (0, 0, 0)
red = (213, 50, 80)
green = (0, 255, 0)
blue = (50, 153, 213)
dis_width = 600
dis_height = 400
```

3. **Set up the Snake:**
   The snake is represented as a list of coordinates. We initialize the snake with a starting length and a starting position.

```
snake_block = 10
snake_speed = 15
def our_snake(snake_block, snake_list):
    for x in snake_list:
        pygame.draw.rect(dis, black, [x[0], x[1], snake_block, snake_block])
```

4. **Set up the Food:**
   The food is represented as a rectangle with a random position on the screen.

```
foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
```

5. **Game Loop:**
   The game loop is used to keep the game running. It handles user input, updates the snake position, and checks for collisions.

IMPLEMENTATION

The Snake game is a classic game where the player controls a snake on a grid, collecting food to increase its length while avoiding walls and its own tail. The game ends when the snake collides with a wall or its own tail.

In Python, the Pygame library can be used to create a graphical interface for the game. Pygame provides tools for displaying graphics, handling user input, and managing game loops.

To create the Snake game using Pygame, we first need to install Pygame on our system. This can be done using pip, the Python package installer. Once Pygame is installed, we can begin implementing the game. The first step is to initialize Pygame by importing the library and calling the init() function. This initializes Pygame and prepares it for use. We can then set up the game window by creating a new Pygame display with the desired size. We also assign the new display to the screen variable for convenience.

Now that we have set up the game window and background color, we can begin implementing the game logic. The first step is to create the snake. To create the snake, we need to define its size, color, and starting position on the screen. We can also define the snake's speed and initial length. The create_snake() function creates the snake by iterating over the desired length and adding new blocks to the snake_list at the appropriate positions. In this example, we start the snake in the center of the screen by setting the initial x and y positions to half the screen width and height, respectively. Once we have created the snake, we can create the food. Like the snake, we need to define the food size, color, and position on the screen. We also create a create_food() function that generates random x and y positions within the screen boundaries.

Once we have created the snake and food, we can begin implementing the game loop. The game loop is the core of the game logic and runs continuously while the game is in progress. The game_over variable to False at the beginning of the loop. This variable is used to determine when the game should end. We then use a for loop to iterate over the Pygame events and check for the QUIT event. If the user clicks the close button on the window, the game_over variable is set to True and the loop exits.

REFERENCES;

Books:

- Invent Your Own Computer Games with Python
- Python Game Programming
- Pygame Documentation

Websites:

- https://realpython.com/
- http://programarcadegames.com/
- https://www.python.org/
- https://www.youtube.com/watch?v=bfRwxS5d0SI

Faculty Guidelines:

Mr Manoj Varshney (Technical Trainer in GLA University)

GitHub Repository link:

https://github.com/purookulsh13/DBInserter