## MINI PROJECT– II

## SYNOPSIS



Department of Computer Science & Application

## Institute of Engineering & Technology

SUBMITTED TO: - SUBMITTED BY:

Mr. Manoj Varshney 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, which thus 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

The study described in this research report focused on variables which were posited to capture students’ experiences of the online tutoring service, e-Learning, and relationships with the students’ perceptions of their academic capabilities and academic performance. A theoretical model incorporating variables from the Technology Acceptance Model, the Theory of Planned Behaviour, and Social Cognitive Theory was developed and tested. A total of 506 undergraduate students from a university located in Sydney, Australia, completed an online survey. Data were analysed using confirmatory factor analysis (CFA) and structural equation modelling (SEM). The results suggested that the perceived usefulness of E-Learning had a direct positive relationship with academic self-efficacy, and an indirect positive association with the students’ academic grades through academic self-efficacy. There was a direct positive relationship between academic self-efficacy and students’ academic grades. The implications of these results and directions for future research are discussed in this report.

# **Contents**

1. Introduction
2. Software Requirement
3. Project Description
4. Working
5. Implementation
6. References

# INTRODUCTION

The emergence of modern technologies has had profound impacts on the education landscape, with online learning now an integral part of the learning process. The main advantages of online learning are flexibility and accessibility (Wu, Tennyson, & Hsia, 2010). Student access to educators to assist them is no longer restricted to the hours of operation of schools and universities, but can be provided anytime and anywhere. Face-to-face tutoring is a well-established, and effective, instructional method. However, there is a need for more empirical research to be directed toward investigating users’ experiences with online tutoring services, their impact on academic confidence (self-efficacy), and achievement scores. The project is about  a classic arcade game where the player controls a growing snake on a screen, collecting food and avoiding obstacles. The game typically starts with a single dot representing the snake's head, which moves around the screen in a particular direction. The player can change the direction of the snake's movement by using arrow keys or other input devices. The objective of the game is to eat food items that appear randomly on the screen, which causes the snake to grow longer. Python libraries, such as Pygame, make it easy to create 2D games, including Snake games, without needing to have extensive knowledge of graphics programming.

## 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.

 Bottom of Form

## 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:

Ms Madhu (Technical Trainer in GLA University)

## GitHub Repository link:

## <https://github.com/purookulsh13/DBInserter>