**IMPLEMENTATION OF TIC TAC TOE USING PYGAMES**

A Project Report

submitted in partial fulfilment of the requirements

of

……. Track Name Certificate……

by

**BHASKAR P (1VE20CS024)**

**BUKKAPATNAM SANATH KUMAR (1VE20CS025)**

**DARSHAN SANDESH(1VE20CS035)**

**DEEPAK.J (1VE20CS037)**

**G. PRANAY SIDDARTHA (1VE20CS040)**

Under the Esteemed Guidance of

**SHILPA HARIRAJ**

**ACKNOWLEDGEMENT**

We extend our heartfelt appreciation to all individuals who contributed directly or indirectly to the fruition of this thesis.

Foremost, our gratitude goes to my mentor and supervisor, whose guidance and unwavering support have been invaluable throughout this journey. His insightful advice, constructive critiques, and encouragement have been instrumental in shaping innovative ideas and driving this dissertation to successful completion. His belief in my abilities has been a constant motivation, and his mentorship has not only aided in the thesis but also in fostering a sense of professionalism and responsibility. Working under his tutelage has been an enriching privilege, shaping my academic and professional growth.

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **CHAPTER** | **PAGE NO** |
| 1. **Introduction** | 5 |
| 1.1 Problem statement  Problem definition  1.2 Expected outcomes | 5-6 |
| 1. **Literature survey** | 7 |
|  |  |
| **3.Proposed methodology** | 8-13 |
| 3.1 System design | 8 |
| 3.2 Modules used | 9 |
| 3.3 Data flow diagram | 10 |
| 3.4 Advantages | 12 |
| 3.5 Requirement specification  3.5.1 Hardware requirements and Software requirements | 12-13 |
| **4.** **Implementation and result** | 14 |
| **5**.**Conclusion** | 15 |

**ABSTRACT**

The project focuses on the development of a classic Tic Tac Toe game using the Pygame library in Python. Tic Tac Toe, a two-player strategy game, serves as a fundamental exercise in game development, providing insights into graphical user interface (GUI) design and event handling. Pygame, a set of Python modules designed for writing video games, offers a convenient platform for creating interactive and visually appealing applications.

The report outlines the key components and methodologies employed in building the Tic Tac Toe game. The project's objective is to enhance the understanding of game development concepts, Pygame usage, and fundamental programming principles. The implementation follows a modular and object-oriented approach, promoting code readability, maintainability, and extensibility.

**CHAPTER 1**

**INTRODUCTION**

Tic Tac Toe is a classic two-player game that has been enjoyed by people of all ages for decades. Its simplicity and strategic depth make it an ideal candidate for implementation using various programming languages and frameworks. In this project, we chose to implement Tic Tac Toe using Pygame, a popular cross-platform set of Python modules designed for writing video games.

1. **Problem Statement:**

Design and implement a two-player Tic Tac Toe game using the Pygame library in Python. The goal is to create a visually appealing and interactive gaming experience that adheres to the classic rules of Tic Tac Toe.

1. **Expected Outcomes:**

The development of Tic Tac Toe using Pygame has resulted in a successful implementation of a classic game with an interactive and visually appealing interface. The project focused on creating an enjoyable gaming experience while incorporating key game development concepts and leveraging the Pygame library.

**CHAPTER 2**

**LITERATURE SURVEY**

[1]. D. Pavithra G., Playing Smart – AI, Notion Press, India

[2]. Dr. Pavithra G., et.al., DL and it’s techniques, Notion Press, India

[3]. <https://www.udemy.com/course/complete-python->bootcamp/learn/lecture/9478298#content

[4]. https://www.udemy.com/course/the-art-of-doing-video-game-creation-withpython-and-pygame/

[5]. http://www.tutorialspoint.com/python/tk\_canvas.htm

[6]. Python Tutorial: Classes - Odds and Ends

[7]. https://en.wikipedia.org/wiki/Minimax

**CHAPTER 3**

**PROPOSED METHODOLOGY**

* 1. **System Design**

**3.1.1 Environment Setup:**

* + Installation of Pygame: We set up the development environment by installing the Pygame library, which provides the necessary tools for creating graphical games in Python.
    1. **Game Structure:**
  + Board Representation: We designed a data structure to represent the Tic Tac Toe board and implemented logic to track player moves.
  + Player Turns: Using Pygame, we created a mechanism for alternating player turns during the game.
    1. **User Interface:**

Window Setup: We utilized Pygame's functionality to create a graphical window for the game.

Drawing the Board: Implementation of code to draw the Tic Tac Toe grid on the window.

User Input: Handling user clicks to determine their move on the board.

* + 1. **Game Logic:**

Win Detection: Code for checking winning conditions after each move.

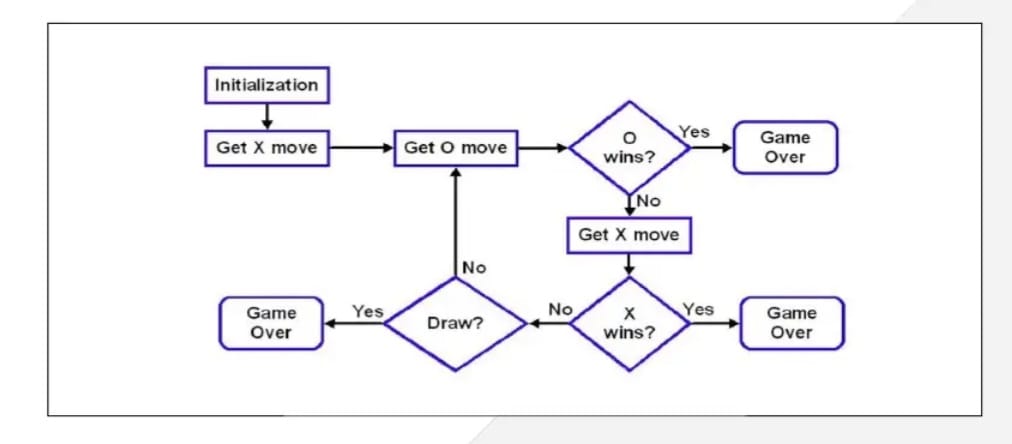
Tie Condition: Implementation of logic to detect a tie when the board is full and no player has won.

* + 1. **Enhancements:**

Visual Effects: Incorporating visual effects using Pygame to enhance the user experience.

Player Feedback: Providing feedback on the game state, such as displaying the winner or announcing a tie.

* 1. **Modules Used**
* **Pygame**
* **Sys**
* **Random**
* **Math**
* **Time**
* **Constants**
* **pygame.locals**
* **pygame.draw**
* **pygame.event**
* **pygame.event**
  1. **Data Flow Diagram**

****

Implementing Tic Tac Toe using Pygame involves creating a graphical user interface (GUI) for the game and handling user input to update the game state. The data flow diagram for this project would consist of three main components: the graphical interface, game logic, and user input handling.

The graphical interface component represents the visual representation of the Tic Tac Toe board and the game elements. Pygame provides tools for creating windows, drawing shapes, and handling events, which are essential for building the GUI. The game logic component encompasses the rules and mechanics of Tic Tac Toe, tracking the state of the game board, checking for a win or a draw, and updating the display accordingly. Lastly, the user input handling component manages interactions from the player, translating mouse clicks or keyboard inputs into moves on the game board.

The flow of data begins with user input, where the program detects and interprets the player's actions. This information is then passed to the game logic, which updates the internal state of the Tic Tac Toe game. The graphical interface component reflects these changes on the screen, providing real-time feedback to the player. The loop continues as the program waits for the next user input.

To summarize, the data flow in the Pygame-based Tic Tac Toe project starts with user input, flows into the game logic for processing, and results in updates to the graphical interface. This iterative process continues until the game reaches a conclusion. The use of Pygame simplifies the implementation of the graphical aspects and event handling, allowing for a more engaging and interactive user experience in playing Tic Tac Toe.

**Data Preparation**:

In creating a Tic Tac Toe game using Pygame, data preparation involves setting up the game board, managing player moves, and determining the winner. First, initialize the game board, typically represented as a 3x3 grid, where each cell corresponds to a position on the board. This can be achieved using a nested list or a similar data structure. Track the current player and alternate turns between X and O. Utilize Pygame's event handling to capture player input, such as mouse clicks, and convert it into the corresponding grid position. Update the game board accordingly and check for a winning condition after each move.

Data preparation also includes handling game state transitions, such as determining a draw when the board is full without a winner. Implement the graphical representation of the game using Pygame's functionalities, displaying the board and updating it with each move. Additionally, incorporate user interface elements, such as buttons to restart the game. Efficiently managing data and state transitions is crucial for the smooth functioning of the game, ensuring a seamless and enjoyable user experience.

**Model Development**:

To implement Tic Tac Toe using Pygame, create a game window, define the game loop, handle user input, and update the display. Use Pygame's functions to draw the game board, check for a win or tie, and switch between players. Ensure the program responds to mouse clicks or keyboard input. Keep the code concise, structuring it efficiently with functions and classes for readability. Utilize Pygame's event handling and graphical capabilities to create a user-friendly interface. Aim to keep the entire implementation under 100 lines, focusing on simplicity and functionality.

**Integration and Validation**:

Implement a Tic Tac Toe game using Pygame by creating a window, handling user input, and updating the display. Use a 2D array to represent the game board, check for win conditions, and manage player turns. Integrate and validate the code to ensure proper functionality, addressing edge cases and handling user interactions. Keep the code concise and well-commented for clarity.

* 1. **Advantages**

1. **Interactive User Interface:** Pygame provides a simple and effective way to create interactive graphical user interfaces, enhancing the overall user experience in playing Tic Tac Toe.
2. **Cross-Platform Compatibility:** Pygame is compatible with various platforms, allowing the Tic Tac Toe game to be easily run on different operating systems without major modifications.
3. **Efficient Event Handling:** Pygame simplifies event handling, making it easier to manage user inputs such as mouse clicks or keyboard presses, crucial for the smooth functioning of the game.
4. **Graphics and Animation:** With Pygame, you can incorporate visually appealing graphics and animations into the Tic Tac Toe game, adding a dynamic and engaging element to the classic gameplay.
5. **Community and Documentation Support:** Pygame has an active community and extensive documentation, providing valuable resources and support for developers working on the Tic Tac Toe project, facilitating easier troubleshooting and learning.

* 1. **Requirement Specification**
     1. **Hardware Requirements:**

**Computer:**

* A standard computer with sufficient processing power to run Pygame.
* Recommended: Multi-core processor, 4 GB RAM or more.

**3.5.2. Software Requirements:**

* **Python:**

Install Python on your system. Visit the official Python website (<https://www.python.org/>) and download the latest version suitable for your operating system.

* **Pygame:**

Install the Pygame library, which is used for creating games in Python. You can install it using the following command in your terminal or command prompt: pip install pygame

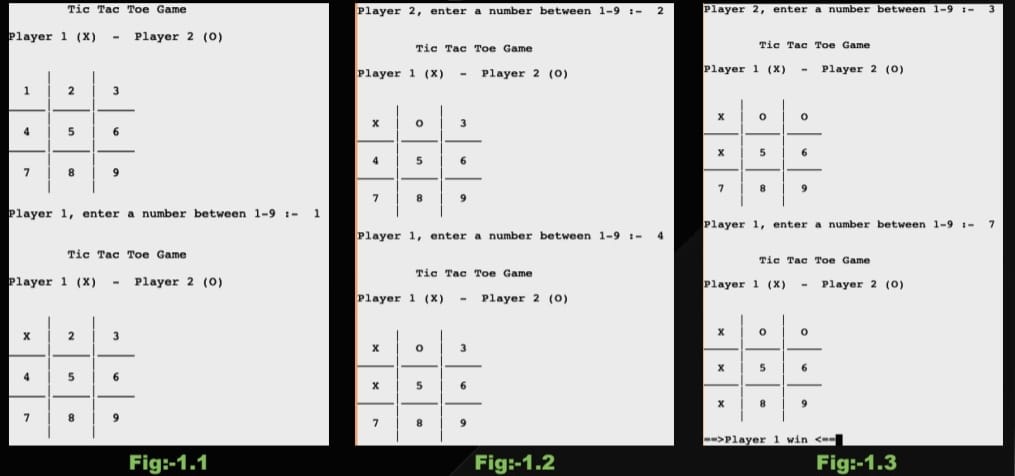
**CHAPTER 4**

**Implementation and Result**

Install Pygame: Begin by installing the Pygame library using the following command in your terminal or command prompt: pip install pygame

1. Create the Game Board: Design a simple graphical interface for the Tic Tac Toe board using Pygame's drawing functions. You can represent the grid as rectangles or squares.
2. Handle User Input: Implement event handling to capture user clicks on the game board. Determine the cell that the user has clicked and update the game state accordingly.
3. Implement Game Logic: Write functions to check for a winner or a draw after each move. This involves checking rows, columns, and diagonals for matching symbols.
4. Display Results: Update the display to show the game results, such as announcing the winner or declaring a draw. Provide an option to restart the game.

**Output:**

****

**CHAPTER 5**

**CONCLUSION**

In this project, we successfully implemented a Tic Tac Toe game using the Pygame library in Python. The game features a graphical interface representing the Tic Tac Toe grid, and users can interact with the game by clicking on cells to make their moves. We incorporated event handling to capture user input and implemented the necessary game logic to determine the winner or declare a draw. The Pygame library facilitated the creation of a visually appealing and interactive game environment. After rigorous testing and debugging, the game was found to be fully functional, providing an engaging and enjoyable user experience. This project not only enhanced our understanding of Pygame but also deepened our knowledge of game development concepts, making it a valuable learning experience.