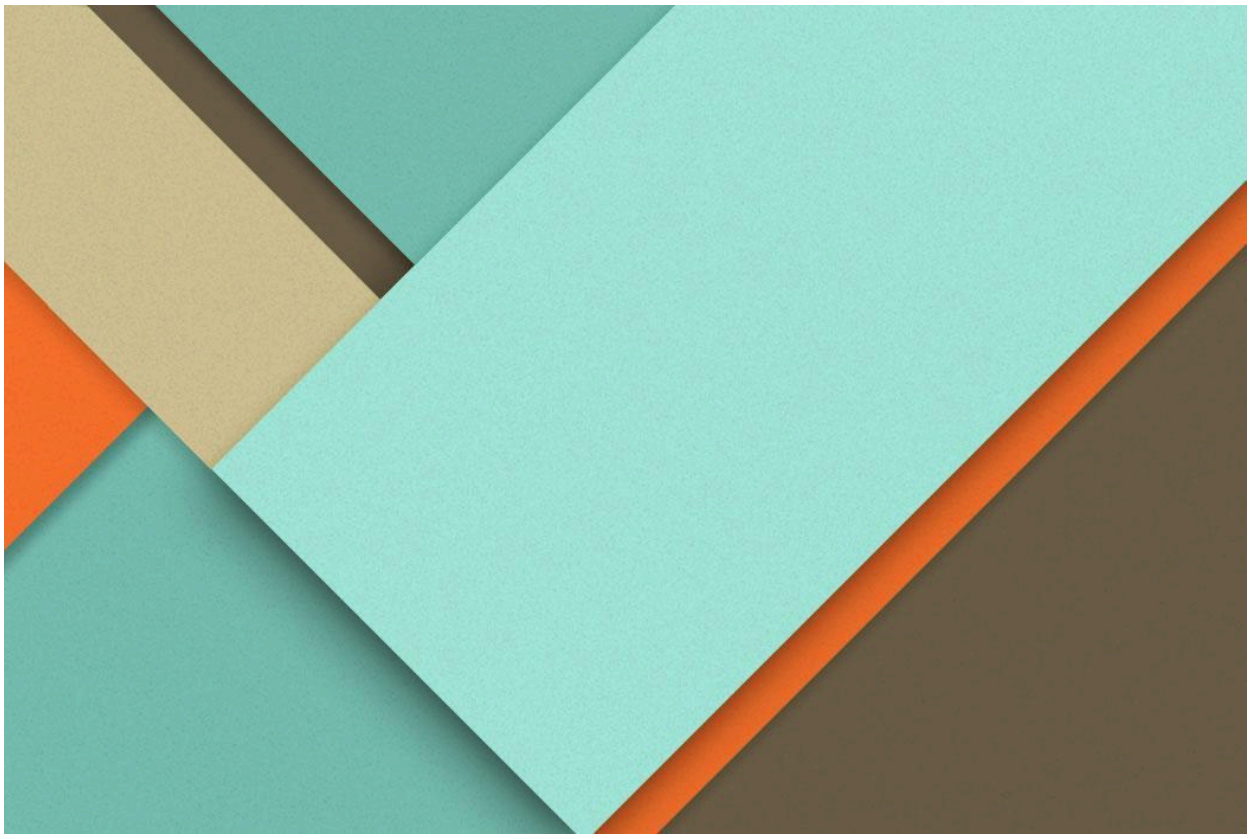

Tic Toc Toe

22.01.2024

Team Members	Assigned Roles
THALLU CHETHAN ADITYA REDDY(20BTRCL054)	Project Manager (PM)
ROHIT L(20BTRCL047)	Lead Developer (LD)
RAMAVIJJALA NAGA VISHNU ADITYA(20BTRCC008)	Interface Designer (ID)

RAMISETTY VEERA RAMA VIKHIL ROYAL(20BTRCL045)	Quality Assurance (QA)

Overview

1. **Game Rules:**

- Understand the rules of Tic-Tac-Toe.

- A 3x3 grid where two players take turns to mark their symbol (X or O) in an empty cell.

- The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins.

2. **User Interface:**

- Implement a user-friendly interface for the console or Jupyter Notebook.

- Display the Tic-Tac-Toe board after each move.

- Prompt players to input their moves.

3. **Board Representation:**

- Create a data structure to represent the Tic-Tac-Toe board.

- This could be a 2D list or another appropriate data structure.

4. **Player Turns:**

- Implement logic to alternate between players' turns.

- Ensure that players can only place their symbol in an empty cell.

5. **Winning Conditions:**

- Check for winning conditions after each move.

- Declare the winner or a tie when the game is over.

6. **Input Validation:**

- Validate user input to prevent errors or cheating.
- Handle invalid inputs gracefully and prompt the user for correct input.

7. **Game Loop:**

- Implement a game loop that continues until a player wins or the game ends in a tie.
- Ask players if they want to play again after the game concludes.

8. **Optional Features:**

- Allow players to choose their symbols (X or O) at the beginning of the game.
- Implement a simple AI for a single-player mode.
- Keep track of the overall score for multiple rounds.

9. **Documentation:**

- Provide clear instructions on how to play the game.
- Include comments in your code to explain complex sections.

10. **Testing:**

- Test the game thoroughly to ensure it works as expected.
- Consider edge cases and handle them appropriately.

11. **Code Structure:**

- Organize your code into functions or classes for better readability and maintainability.
- Follow good coding practices and naming conventions.

Goals

1. **Implement Player vs. Player Mode:**

- Objective:Allow two human players to take turns playing the game.
- Key Components:
  - Develop a mechanism for alternating turns between Player 1 and Player 2.
  - Implement input validation to ensure that players can only make valid moves.
  - Display the updated board after each move and check for a winning condition or a tie.

2. **Create a Single-Player Mode with Basic AI:**

- **Objective:** Provide an option for a single player to play against a basic AI opponent.
- **Key Components:**
  - Develop a simple AI algorithm that can make valid moves on its turn.
  - Allow the player to choose whether to play as X or O.
  - Enhance the AI to make strategic moves and prevent the player from winning.

3. **Enhance User Experience and Interaction:**

- **Objective:** Improve the overall user experience by adding informative messages and enhancing the visual presentation.
- **Key Components:**
  - Display clear instructions on how to play the game at the beginning.

- Provide feedback after each move, indicating whose turn it is and the current state of the game.

- Add an option for players to restart the game or exit gracefully after a game concludes.

Achieving these goals will result in a well-rounded Tic-Tac-Toe game that supports both player vs. player and player vs. AI modes, while also ensuring a positive user experience.

Specifications

Team Member	Task Assigned
THALLU CHETHAN ADITYA REDDY(20BTRCL054)	•
ROHIT L(20BTRCL047)	•
RAMAVIJJALA NAGA VISHNU ADITYA(20BTRCC008)	•
RAMISETTY VEERA RAMA VIKHIL ROYAL(20BTRCL045)	

DEADLINE FOR PROJECT COMPLETION: 24th January, 2024

Milestones

I. Milestone 1

- Brainstorming session
- Assign roles

II. Milestone 1

- LD and ID develops the game logic.
- The PM coordinates the work.

III. Milestone 1

- QA tests the game.
- Bug Fixing

IV. Milestone 1

- Conduct a code review session.
- Reflect on learning and project experience

Timeline

1. Planning

Brainstorming session.

Finalize project scope.

Assign roles.

2.Development

LD develops the game logic.

3.Integration and Testing

•Combine code from LD and ID.

•QA tests the game.

•Bug fixing.

4.Presentation and Review

•Prepare a presentation of the project.

•Conduct a code review session.

•Reflect on learning and project experience

Challenges Faced:

- Ensuring the accurate implementation of the game rules and logic can be challenging. Handling different scenarios for player moves, checking for a win or a draw, and managing the game state requires careful consideration.
- Dealing with user input can be tricky. The program needs to validate and handle user moves effectively, ensuring that only valid moves are accepted and responding appropriately to invalid inputs.
- Implementing robust error handling is crucial. The program should handle unexpected situations, such as invalid input or system errors, without crashing.
- Designing an intuitive and visually appealing interface can be challenging. Ensuring responsiveness and user-friendly interactions adds complexity.
- Comprehensive testing is necessary to ensure the game behaves as expected. Writing effective unit tests and ensuring full coverage can be time-consuming, but it is crucial for identifying and fixing bugs.
- While Tic-Tac-Toe is not computationally intensive, optimizing the code for performance and efficiency is still important. Ensuring that the game runs smoothly and responds quickly to user input enhances the overall user

experience.

● Creating clear and comprehensive documentation that assists users in understanding how to play the game and how the code is structured can be a challenge, especially for those who may not be familiar with programming.

Learning Outcomes:

● **Algorithmic Thinking:**

○ Develop the ability to design algorithms to solve specific problems, in this **case, implementing the logic for a Tic-Tac-Toe game.**

● **Programming Proficiency:**

○ Enhance programming skills by implementing a complete project in Python, including managing data structures, handling user input, and developing **modular and organized code.**

● **Problem-Solving Skills:**

○ Gain experience in identifying and solving challenges that arise during software development, such as handling user input, implementing game logic, and addressing unexpected errors.

● **User Interface (UI) Design:**

○ If applicable, learn the basics of designing a simple user interface for console-based or Jupyter notebook environments, improving user experience

and interaction.

- **Error Handling and Debugging:**
  - Develop skills in detecting and fixing errors by implementing robust error handling mechanisms and debugging techniques.
- **Testing and Quality Assurance:**
  - Understand the importance of testing in software development. Develop and execute a testing plan, including unit tests and integration tests, to ensure the correctness and reliability of the game.
- **Documentation Skills:**
  - Improve documentation skills by creating clear and concise documentation for the Tic-Tac-Toe game, helping others understand the code and facilitating future maintenance.
- **Version Control:**
  - Learn to use version control systems (e.g., Git) to manage code changes, collaborate with others, and maintain a history of project development.
- **Cross-Platform Development:**
  - Gain insights into developing software that is compatible with different platforms, such as console and Jupyter notebook environments, expanding skills in cross-platform development.
- **User Feedback and Iterative Development:**
  - Understand the importance of user feedback and iterative development. Incorporate user suggestions to improve the game, demonstrating flexibility

and responsiveness to user needs.

- **Coding Best Practices:**

  ○ Apply coding best practices, including code modularity, readability, and adherence to coding standards. Learn the importance of code reviews for improving code quality.

- **Creativity and Innovation:**

  ○ Encourage creativity by exploring potential enhancements or variations to the Tic-Tac-Toe game. Experiment with new features and ideas to make the project more interesting.

Future Scope:

○ Regularly test and update the game to maintain compatibility with evolving Python versions and libraries

○ Encourage users to provide feedback on potential issues or improvements to facilitate continuous testing and refinement

○ Consider enhancing the visual feedback for successful moves and game outcomes to make the gameplay experience more engaging.

○ Enhance the computer's gameplay through the integration of reinforcement learning techniques.