Tic Tac Toe

22.01.2024

—

Team Members	Assigned Roles
Manish Ghoshal	Lead Developer /QA & Interface Designer
Dukul Bhardwaj	Project Manager

Overview

The scope of the project is to develop a Tic-Tac-Toe game using Python that can be played in the console or a Jupyter notebook.

Goals

1. To understand array manipulations which can be useful for the
   game board.
2. To ensure that the deadlines are met by each member.
3. Analysis of the code for bugs and other issues.
4. To prepare a final project review documentation before presentation.

Specifications

Team Member	Task Assigned
Dukul Bhardwaj	- To conduct regular team meetings. - To ensure the deadlines are met. - To prepare a progress report and timeline. - Test and evaluate the code efficiency.
Manish Ghoshal	- Design user interface for the game. - Implement input/output in console or Jupyter notebook. - Test and evaluate the code efficiency. - To prepare User Experience Report
Manish Ghoshal	- Write core game logic. - Ensure code quality and efficiency. - Test and evaluate the code efficiency. - To prepare Code Documentation

DEADLINE FOR PROJECT COMPLETION: 24th January, 2024

Milestones

I. Define Project Scope and Requirements

- Conduct a detailed analysis of the project requirements.
- Identify key features and functionalities needed for the Tic-Tac-Toe game.
- Specify the rules and constraints for the game.

II. Algorithm Design

- Develop an algorithm for the game logic, outlining the rules for player moves and win/lose conditions.
- Consider different approaches for handling user input and game state

III. Implementation

- Write Python code to implement the Tic-Tac-Toe game based on the designed algorithm.
- Develop functions for player moves, checking the game state, and determining the winner.
- Ensure that the code is modular, well-organized, and adheres to best coding practices.
- Create a console-based user interface for playing the game.

IV. Overcoming Roadblocks

- Identify and address any challenges or roadblocks encountered during the implementation phase.
- Debug and optimize the code to ensure smooth gameplay.
- Test the game thoroughly to catch and fix any unexpected issues.

V. Code review and analysis

- Conduct a comprehensive code review to ensure code quality and adherence to coding standards.
- Collaborate with team members or peers to gather feedback on the code structure and functionality.
- Perform code analysis to identify potential improvements or optimizations

.

VI.   Documentation

- Create clear and concise documentation for the Tic-Tac-Toe game.
- Include details on how to play the game, rules, and any additional features.
- Document the codebase with inline comments for better understanding.
- Provide a README file with instructions for running the game in both console and Jupyter notebook environments.

VII.   Testing

- Develop and execute a comprehensive testing plan to ensure the game functions as intended.
- Include unit tests for individual functions as well as integration tests for the entire game.
- Address any bugs or issues identified during testing.

VIII.   Deployment

- Prepare the game for deployment in both console and Jupyter notebook environments.
- Create executable files or packages for easy distribution.
- Ensure that the game can be easily installed and run on different systems

IX.   User Feedback and Iteration

- Gather feedback from users who play the game.
- Iterate on the game based on user feedback, addressing any usability or functionality concerns.
- Consider adding enhancements or additional features based on user suggestions.

Timeline

PROJECT
TIMELINE

JAN 22
PROJECT OVERVIEW AND DISCUSSION

Algorithm Design
Allocating Roles

JAN 23
DEVELOPMENT

LD – Game logic
ID – User Interface
PM – Coordinates Progress

JAN 24
INTEGRATION AND TESTING

Combine work from LD and ID
QA tests the game
Bug Fixing

DOCUMENTATION

PM – Project Plan
ID – User Experience Report
LD – Code Documentation

JAN 25
PRESENTATION AND REVIEW

Code Review Session

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.