Tic-Tac-Toe Game

22.01.2024

—

Overview

The provided code is a Tic Tac Toe game implemented using Python's tkinter library. It offers two modes of play: single-player (against the computer) and multi-player (against another human player).

Goals

- Working Tic-Tac-Toe game.
- Project documentation
- Analysis of the code for bugs and other issues
- To prepare a final project review documentation before presentation.

DEADLINE FOR PROJECT COMPLETION: 24th January, 2024

Milestones

I. Defining Project Scope and Requirements

- Conduct a thorough examination of the project specifications.
- Identify the key components and qualities needed to play the game Tic Tac Toe ● Outline the restrictions and guidelines that apply to the game.

II. Creating the Algorithm Design

- Create an algorithm that determines the rules for player behaviour, win/loss scenarios, and other aspects of the game logic.
- Examine several approaches to controlling game state and user input.

III. Put into Practice

- Write Python code to implement the developed algorithm-based Tic-Tac-Toe game.
- Provide processes for player action, assessing the condition of the game, and selecting the winner.
- Make sure the code is organized, modular, and follows best practices for coding.
- Create an interface allowing users to engage with the game on a console.

IV. Overcoming Obstacles

- Determine and fix any issues that arise during the implementation stage.

- Optimise and debug the code to ensure a smooth gameplay experience.
- Test everything thoroughly to find and fix any unexpected problems.

V. Analysis and Review of Code

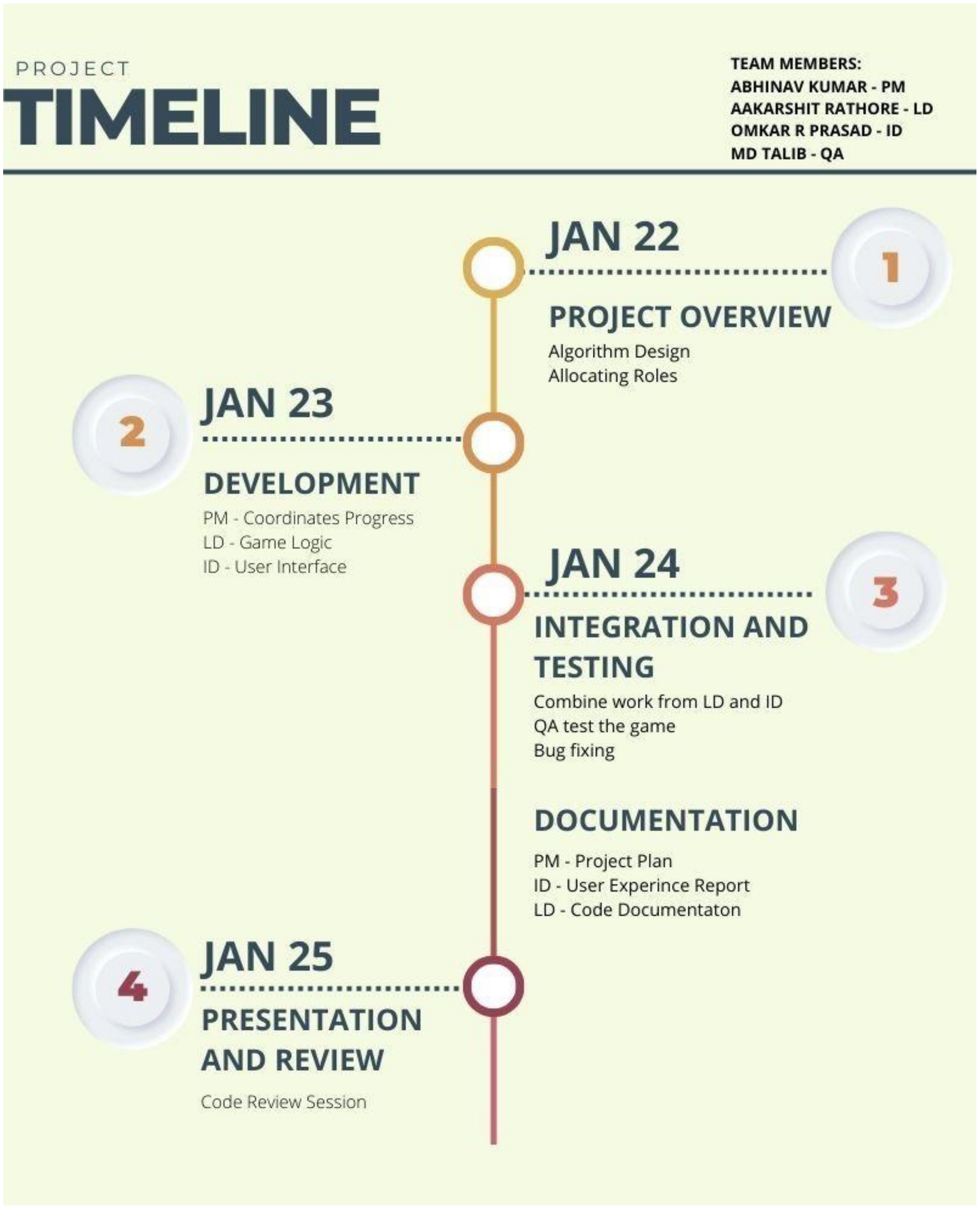
- To guarantee quality and respect to coding standards, perform a comprehensive code review.
- Get input from colleagues or team members to assess the functionality and organisation of the code.
- Analyse the code to find possible improvements or optimisations.

VI. Record-keeping

- Provide a succinct and lucid description of the Tic-Tac-Toe game, including rules, gaming guidelines, and extra features.
- For better understanding, include inline comments throughout the codebase.

.

Timeline



PROJECT

TIMELINE

TEAM MEMBERS:
ABHINAV KUMAR - PM
AAKARSHIT RATHORE - LD
OMKAR R PRASAD - ID
MD TALIB - QA

1 **JAN 22**

PROJECT OVERVIEW

Algorithm Design
Allocating Roles

2 **JAN 23**

DEVELOPMENT

PM - Coordinates Progress
LD - Game Logic
ID - User Interface

3 **JAN 24**

INTEGRATION AND TESTING

Combine work from LD and ID
QA test the game
Bug fixing

DOCUMENTATION

PM - Project Plan
ID - User Experince Report
LD - Code Documentaton

4 **JAN 25**

PRESENTATION AND REVIEW

Code Review Session

Challenges Faced:

- It can be difficult to ensure that the logic and game rules are applied correctly. Careful thought must go into handling various player move circumstances, determining if a win or a draw occurs, and controlling the game's state.
- Handling input from users can be challenging. To ensure that only legitimate moves are allowed and that incorrect inputs are handled appropriately, the program must properly validate and manage user moves.
- It is imperative to implement effective error handling. Unexpected events like incorrect input or system failures should be handled by the program without crashing.
- Even though Tic Tac Toe doesn't require a lot of processing power, it's still crucial to optimize the code for effectiveness and speed. Ensuring that the game operates seamlessly and reacts promptly to user input improves the user experience in general.

Learning Outcomes:

- **Algorithmic Ideas:**

○ Gain the capacity to create algorithms that address certain issues; in this example, this is putting the reasoning for a Tic-Tac-Toe game into practice. ● **Programming Expertise:**

○ Develop expertise in recognising and resolving issues that crop up in software development, like managing user input, putting game logic into practise, and dealing with unforeseen problems.

● **Error Handling and Debugging:**

○ By putting strong error handling systems and debugging approaches into place, you may improve your skills in identifying and repairing mistakes.

● **Testing and Quality Assurance:**

○ Recognize the significance of testing in the process of developing software. Create and carry out a testing strategy that includes both integration and unit tests to guarantee the accuracy and dependability of the game.

● **Documentation abilities:**

○ Make the Tic-Tac-Toe game's documentation clear and simple, assist others in understanding the code, and make future maintenance easier.

Future Scope:

- **GUI Improvements:** The graphical user interface (GUI) could be improved by using more modern and visually appealing design elements. Additionally, the GUI could be made more responsive and user-friendly by incorporating features such as undo/redo and the ability to reset the game board.
- **BOT Model improvements:** The (PvC) Person vs computer system currently runs on a randomizer function where the computer selects a random solution out of the available solution of the computer. We will try to implement (RL) reinforcement learning into the system for a responsive solution to every case in the grid.
- **Choice of the grid:** The current model consists of a singular grid format i.e. 3x3. We will try and implement a multiple grid option for the user e.g. 4x4, 5x5, etc. This is proposed to improve the UI/UX experience.