Requirements Gathering for Multiplayer Tic-Tac-Toe

By Argjend, Leif, Hazal, John, Zack, Joseph, Gary, & Dave

Introduction

- **Project Overview:** Our project is a Tic-tac-toe game featuring customization options designed for Windows 10 & 11. Users will be presented with a classic 3x3 game board where they can play locally with another user on the same machine, or against a computer bot with adjustable difficulties. The game will also feature sound effects, game theme music, instance win/loss tracking, and the ability for users to customize their icons and backgrounds with uploaded images. The game will also implement match replayability and persistently store saved match replays & users customized themes.
- **Scope:** This version of the game will support both single-player vs a computer and two-player modes. The system allows users to:
  - Play the classic game of Tic-Tac-Toe within the standard 3x3 grid space.
  - Customize the look and feel of the game by uploading custom backgrounds and icons.
  - Track wins/losses for easy accounting in continuous sessions.
- **Purpose of the Requirements Document:** To outline and detail the functional and nonfunctional requirements for the Tic-Tac-Toe game. By specifying the expected behavior and constraints, the document aims to guide the development, testing, and validation process *ensuring the final product aligns with the project goals.*

Requirements Gathering

Customers)

1. **Casual Tic Tac Toe Players:** These players want to sink a little time into playing the game, and are likely doing it on a device without the resources to play higher spec games. Like students slacking off in class on web browser games.

2. **Advanced Tic Tac Toe Players:** These are people who have played a lot and perhaps studied strategy or the logic of Tic-Tac-Toe. They would like an application that supports an AI that is at their skill level, and would like to have a customized set up to play on with their real opponents. As they will naturally be spending more time playing, robust options and features will serve them.

System Requirements)

- **Functional**
  - System will have GUI that players will use to interact with game logic
  - If the end user's device runs the acceptable operating systems, the game will also run. ??
  - Application will be provided in compressed format

Does the application actually do anything?
The answer will lead to the functional requirements.

- Customization file formats will be consistent across operating systems
- GUI appearance doesn't differ from applications based on operating system
- There will only be one light weight release of the application
- Network Independence, the game will be fully functional offline. No online features, such as multiplayer over the internet, are included.

- **Non Functional**
  - Launch time will be within 5 seconds
  - Exit time will be within 2 seconds
  - Application can be distributed through any methods of file sharing
  - Weight of application should be at max 2 gigs [Non]
  - Platforms the application will run on are Windows 10 or 11

what is an AI opponent.

Application Requirements)

- **Functional**
  - There will be instructions on how to use the application embedded within
  - AI opponent has 3 difficulty levels
    - The system must provide three AI difficulty settings—easy, medium, and hard—that adjust the AI's decision-making logic accordingly.
  - There are two game modes; solo vs AI and multiplayer for two players
    - The game must allow two players to play on the same device, alternating turns for each move during the game session.
  - The Tic-Tac-Toe grid will be 3 x 3
    - The system must implement a standard 3x3 grid for all matches, ensuring each player can place their symbols in an empty cell on their turn by clicking on the selected area.
  - Instance-Based Win/Loss Tracking:
    - The game must track and display the win/loss record for each player at session end
  - Matches can be replayed by a users request or at a later time if saved by user
    - Recent replays file ~ used to house instance based-sessions
    - Save replay file ~ used to house persistent sessions added by user
      - Saved replay elements
        - Memo set by user
        - Which theme was being used
      - Ability to watch saved replay
  - Customization of Icons, Background & Music
    - Theme elements are as follows)
      - Background color/Image
      - Symbols (board pieces)
      - Symbol placement sound
      - Background music
      - End game sound
      - Grid lines

why are there Icons?

Don't reference UI components

- Application must allow users to select a set of 3 default themes
  - Users can't edit default themes
- Application must allow users to save their own custom theme
  - Users can re-use default theme aspects while creating their custom theme
  - Users custom theme will be housed in persistent data storage
    - Save theme file ~ used to house persistent custom themes created by user
- Home screen is loaded on app launch

- **Non Functional**
  - AI opponent will make moves within 1-2 seconds
- Our game must have the following views
  - **Home:** Used to display ability for users to create matches, the first user must specify match type and difficulty if a single player is selected. Also will have buttons to go to replay and customization views.
  - **Game:** Will show the 3 x 3 game board, a return button, music on-off toggle and if the game is over it will load a component that allows the user to see the winner, and ability to save/view the replay of that match.
  - **Replay:** Will show a list of memos and upon selecting a memo it'll load the replay by showing the 3 x 3 playing grid like usual, except now you can to go through each move using a forward/backward arrow.
  - **Customization:** Allows the user to select themes from a dropdown menu and allows user to customize their own select theme and save.

Use Cases

Use Case 1: Single Player vs AI Gameplay

Actor: Player

Description: The player opens the application, selects their preferred AI difficulty on the home screen, then plays a game to completion. Upon finishing the game, they will decide if they want to save the replay of the completed game or not. Then, they will decide if they want to play another game.

Preconditions: The game is installed and opens correctly without error.

Main Flow:

1: The player launches the application

2: The player selects their preferred difficulty from the options offered after indicating they are playing single player.

3: The player is presented with the main game screen with the 3 x 3 grid where they will play a complete game of Tic-Tac-Toe.

4: The application will prompt the player on whether or not they would like to save the replay of the finished game with a memo.

4a: If the player elects to save the replay, the user will have to supply a memo prior to saving. Replay will be held in persistent storage.

5: Once the player has made their selection, they will also be prompted if they would like to play again. If they hit yes, go to 3. If no, return to the title screen.

6: On the home screen, the player selects to close the game.

This is a lot of use cases!! separate them

Use Case 2: Multiplayer Gameplay

Actors: Player 1 and Player 2

Description: Player 1 opens the game, selects multiplayer gameplay on the home screen, and then plays a match locally by sharing the device with player 2. When the game ends, they will choose to play multiple games because of how much fun they're having.

Preconditions: Have two players who intend to play Tic-Tac-Toe

Main Flow:

1: The owner of the device which the application is installed on (referred to as Player 1 though it could be Player 2) will open the application to the home screen.

2: Player 1 selects multiplayer

3: Player 1 and 2 play a complete game of Tic-Tac-Toe.

4: Upon completing the game, the players will be prompted if they want to save the replay or not.

5: Player 1 and 2 decide they want to play again, and hit replay. The main game screen will display the current score between players.

6: After several games, Player 1 decides to hit no on the replay prompt and goes back to the home screen.

7: On the home screen the player closes the game.

Use Case 3: Customizing Theme

Actor: Player 1

Description: A player wants to change the current customization options.

Preconditions: Game must be open and player must be on home page

Main Flow:

1: While on the home screen the player selects the customization menu

2: The player selects one of the three default themes

2A: The player chooses the custom theme

2B: The player customizes various theme elements

3: Player Saves Selected Theme

4: Game updates to reflect new theme

Postconditions: The game displays the selected theme .

Use Case 4: Replaying a Previous Game

Actor: Player 1

Description: the player starts the application. On the title screen the player selects the replay option. Once on the replay screen the player can choose from a previously played and saved game based on the name the player gives at the end of a game.

Precondition: there should be games played and saved.

Main flow:

1: Player starts the application and chooses the replay option on the home screen.

2: Player then will be directed to the replay screen with the list of previously saved games.

3: Player will select a game and it will be loaded in the game screen with the replay toggles

4: Player can then move forward/backward as needed to see each move in the game

4a: If the player elects to close replay, they can click return that'll go back to step 1.
Postcondition: The selected memo game should be replayed.

Use Case 5: Immediate Replay after Match
Actor: Player 1
Description: After finishing a game the player can immediately replay that game by clicking on the replay button.
Precondition: Complete a new game
Main flow:
1: The player starts the application and chooses the new game option.
2: The player completes the game and clicks the replay option shown upon finishing a game.
3: The player will be shown a replay of the game that was just completed.|
4: Player can then move forward/backward as needed to see each move in the game
4a: If the player elects to close replay, they can click return that'll go back to step 1.
Post condition: The game should be replayed.

Sequence Diagrams?

Glossary?

In these use cases, you are merely describing a UI design you have already completed