Proposed Game:

Tic-tac-toe

Game Title:

4Square

By

Nyiko Ashanti Ngobeni

G22N3906

Christopher McCune

G22M5298

Alisha Schuler-Nitzsche

G22S7874

Mihlali Xhala

G22X7152

Mthobisi Latha

G22L5863

Our proposed game is a board game tic tac toe. The board is supposed to be a 3 X 3 grid of squares. The symbols used are crosses and noughts each representing a single player, there can only be 2 players (Carrol, M.T. and Dougherty, S.T., 2004). To win either three of either "X" or "O" symbols should be in a row horizontally, vertically or diagonally. If either of the three winning conditions are not met by either player and the board is fully occupied then it is a draw.

This game dates back to 1558 when it was initially called Terni lapilli, meaning three pebbles at a time (Do, N., 2005.). The US renamed it Tic-tac-toe in the 20th century (Abu Dalffa, M., Abu-Nasser, B.S. and Abu-Naser, S.S., 2019.). The first version of Tic-tac-toe was played by the Roman empire so we can safely conclude that they discovered it. Unlike today Romans had to move around pebbles leaving spaces open, today once you play in a certain square it remains occupied.

The game we will create is much more Modern and interesting in the sense that a played square remains occupied till there's a winner or the players draw, and the board will be a 4 X 4 grid of squares rather than a traditional 3 X 3 grid. The larger grid will make the game more complex as it will require more thinking and cautious moves than the normal tic tac toe. We will increase the complexity by allowing the players to only place their first move on the outer squares.

The aim of our version of Tic-tac-toe is to increase the difficulty of the game, allowing it to become a game for a larger age group than the original game, as the original game is mostly challenging for only younger children. This variation allows for a more strategic gameplay of Tic-ccc-toe to get the symbols in a row, diagonally, vertically, or horizontally.

To create our game we will need to code in java. The code will be broken up into different functions, such as one that creates the playing board, one that keeps track of which players turn it is and what symbols each player is using, one that places the correct symbol into the correct place and another method that checks whether either player has won or if it ends in a draw. To allow for the continuation of the game we will use a loop.

To make the game look more appealing to users we will use GUI. While using GUI to enhance the game experience we must ensure that it doesn’t interfere with the game itself.

To end the project we will showcase our code on GitHub code and have a project document. In this document, there will be a proposal revision as well as an extension showing the SDLC we followed to complete our project and a storyboard of the game in action. The storyboard will showcase the opening of our game, the game being played and the ending of our game with each panel having captions that will further explain what is being shown.

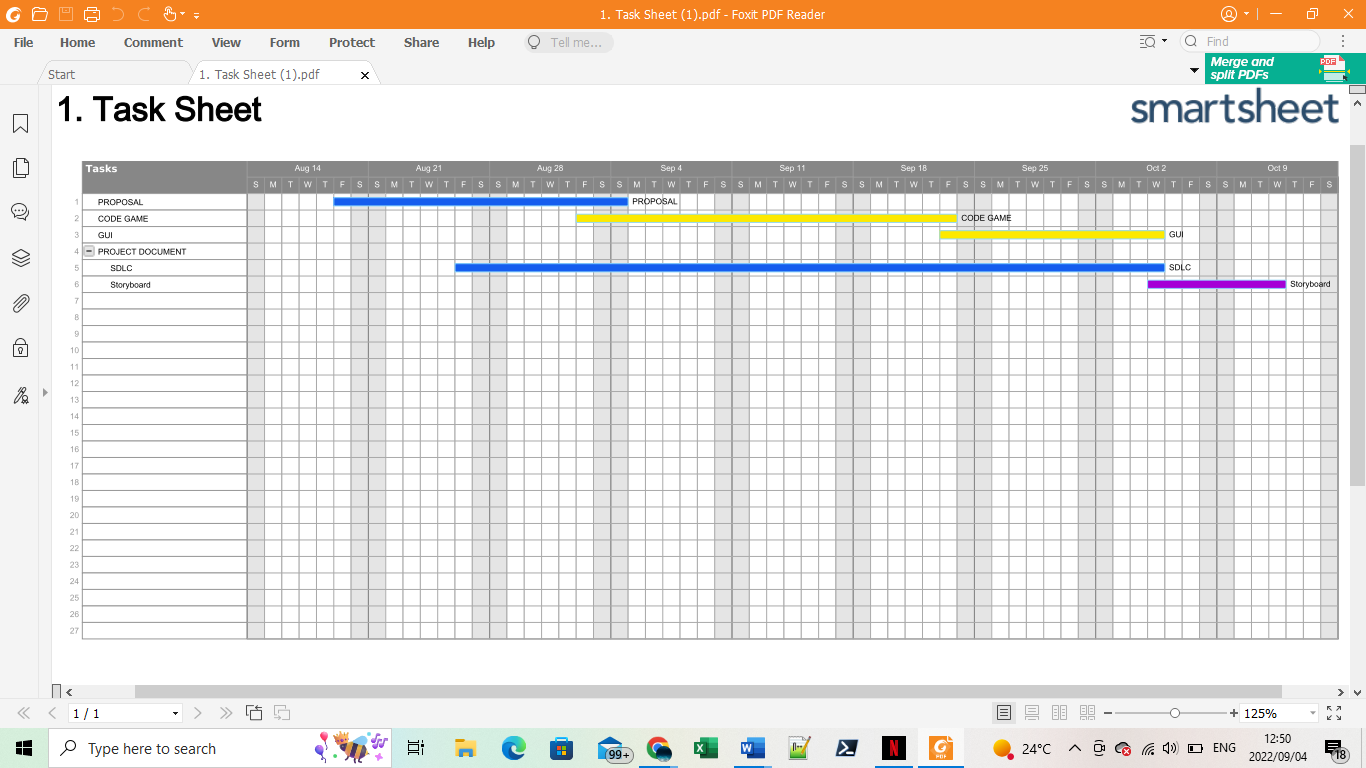


Figure 1: Our predicted timeline for the above milestones with dates.

Reference

Abu Dalffa, M., Abu-Nasser, B.S. and Abu-Naser, S.S., 2019. Tic-Tac-Toe Learning Using Artificial Neural Networks.

Carroll, M.T. and Dougherty, S.T., 2004. Tic-tac-toe on a finite plane. Mathematics Magazine, 77(4), pp.260-274.

Do, N., 2005. How to Win at TicTacToe. Gazette of the Australian Mathematical Society, 32(3), pp.151-161.

JUNI, 2021. Code a Tic Tac Toe Game Java | Beginner-Intermediate Tutorial. Available at: <https://junilearning.com/blog/coding-projects/java-beginner-tic-tac-toe-tutorial/> (Accessed: 29 August 2022).

Lavenda  Z,2022, 4 square project Gantt chart,Smartsheet, <https://app.smartsheet.com/b/publish?EQBCT=3674656f1a554bb886de884111f88af5> (Accessed: 30 August 2022).