**ULTIMATE TIC TAC TOE**

CSC 102 – PROJECT PROPOSAL

Group Members: Derrick Asiedu Aboagye – g22A3680

Luyanda Thomas Biyela – g22B9077

Linda Langa – g22L4786

Daleroy Chinomona – g22C5309

Akhona Mdluli – g22M1142

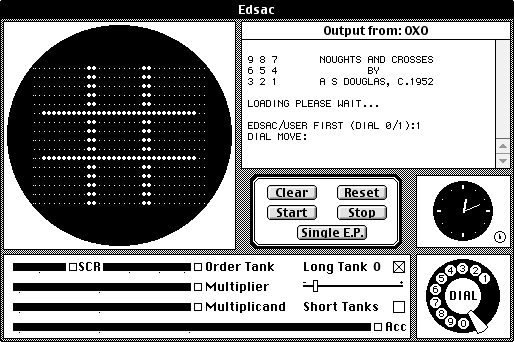
BACKGROUND

History of Tic-Tac-Toe and Motivation for implementing the “Ultimate” version of it:

According to Marcus (2021), the game Tic-tac-toe or noughts and crosses, as most of us are familiar with, had its version in ancient Rome and it was known as “terni lapilli”, which translates to “three pebbles at a time”. In this style of the game, players each had three pebbles/stones on their board and moved them on empty squares to continue playing. Proof of these games have been found in chalk in Rome and this shows how even people over a thousand years ago created different kinds of games to avoid boredom.

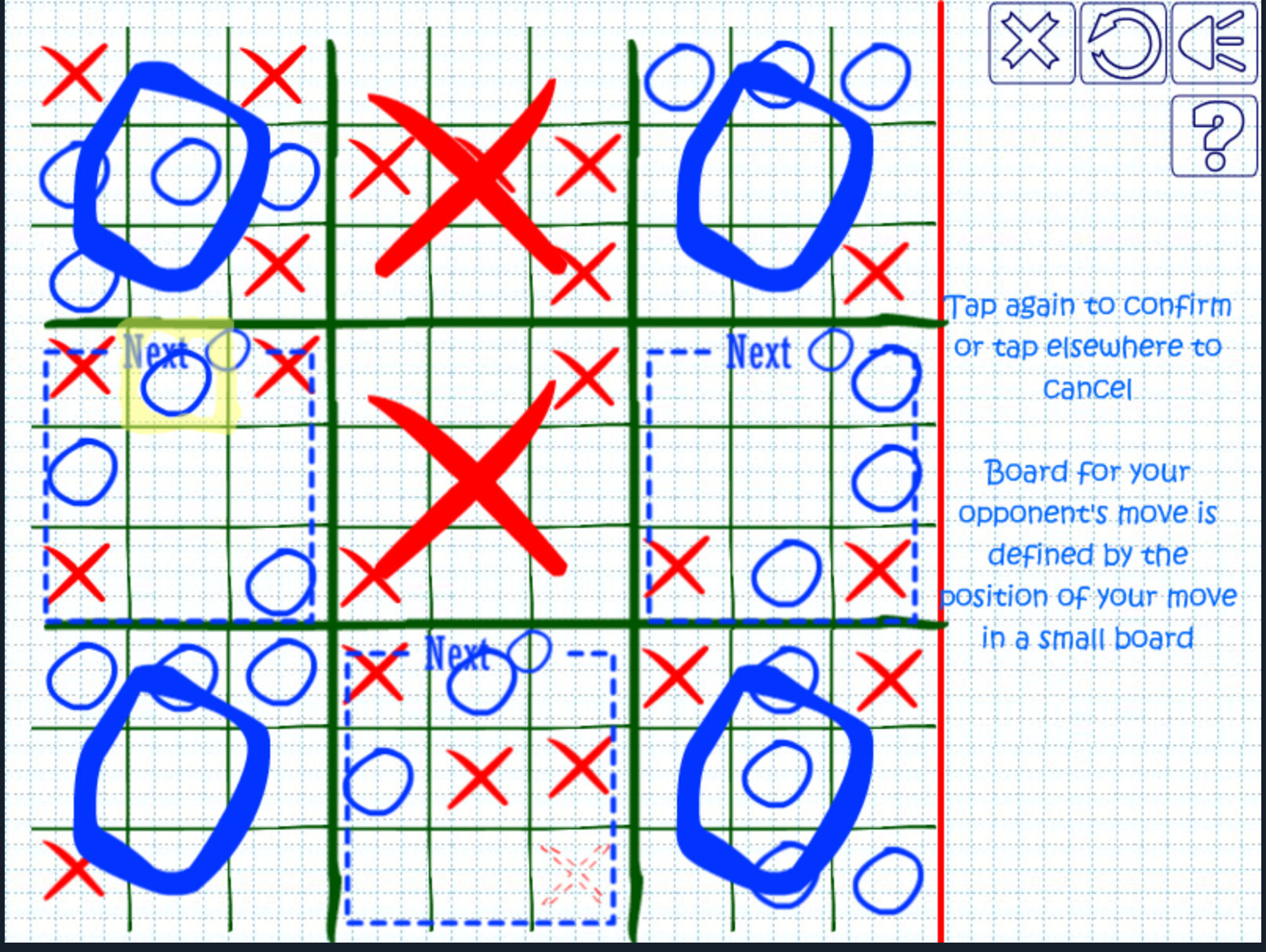
Besides ancient Rome, games resembling Tic-Tac-Toe have also have roots in ancient Egypt, the Middle East and Native America (Marcus, 2021). However, the modern-day Tic-Tac-Toe most of us have come to know started gaining momentum in the middle of the 19-th century in Britain. A famous tabloid in England referred to the game as “Noughts and Crosses”, where noughts are the O’s and crosses are the X’s used to play the game.

Tic-Tac-Toe’s history changed a lot when the game became digitalised in 1952 by a British computer scientist known as Sandy Douglas (Marcus, 2021). The game was “OXO”, and it is one of the first computer games that enabled humans to play against game-playing computers. Below is an image of “OXO”:



MOTIVATION

Tic-tac-toe can be played on almost any surface as long as players can distinguish whether they are an “X” or an “O”. The version of Tic-Tac-Toe we will be implementing is also known as Ultimate Tic-Tac-Toe or Strategic Tic-Tac-Toe. The motivation behind this is to ensure that the game lasts longer than it would be if played on a common 3x3 grid; this will ensure that players think more about their decisions and come up with different strategies so that they win at least three smaller 3x3 grid games in order to come out victories in the whole game. Below is an image of our game, Ultimate Tic-Tac-Toe, should look like:



**PROBLEM STATEMENT**

The problem we are trying to solve includes the short time-span of the normal 3 x 3 tic-tac-toe game. The “why” of this problem is to ensure maximum satisfaction is reached when a player overcomes the hurdle of having to play a game knowing each move, they make should have a meaning; which is not only to win a 3 x 3 game three times, but to win while aligning them three times to win overall. The aim of this version of ultimate tic tac toe is to have the players play an X or an O at a particular square. This should happen until one party loses or wins a game or remains at stalemate. Each game result will be treated as a move played by a player in a bigger grid. This game has a high complexity. As to features of the game, it will only be limited to a one vs one game. Given that the complexity of implementing the game and playing it is roughly “difficult”.

**APPROACH**

The aim of the project is to implement a tic tac toe game in java. We have decided to lay out a few goals that will allow us to do this task. The First is to code a 3 x 3 data structure for the main playing grid. Each of the sub- games can then use their own 3 x 3 data structure. The progress of each game will be stored in the bigger grid, The smaller grids may perhaps be treated as objects and will work almost identically to the bigger grid. Given that this game only allows for two human players, progress can be stored in a single text file if needs be. The GUI for the game will be relatively simple and we will be taking a very minimalist approach in our graphical presentation, as this is a game that only relies on the recognition of common symbols like "X" and "O". Our work will then be presented for assessment in the form of a GitHub code and Project Document. We will be sure to include the extensions of this proposal, which consist of the steps through the SDLC and a systematically structured visual guide that shows the game being implemented. We will do this in the form of a story board. We intend to show the game of beginning, middle and end with the relevant explanations. We also intend to include detailed instances of any special cases that we may have.

SDLC

**JOURNEY/TIMELINE/MILESTONES OF ULTIMATE TIC-TAC-TOE:**

**No.2**

**No.1**





Code 2D arrays -

21 September

Discussion on requirements, scoping, design etc. -

17 September



**No.3**

**No.4**

Create the grid from the 2D Array -

21-22 September

Create logic behind X’s and O’s, i.e., Winning, losing and Stalemate conditions/methods - 24 September



**No.5**



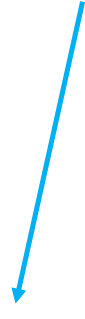
**No.6**

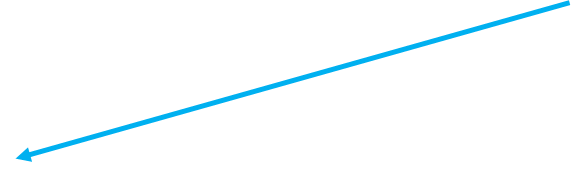
Work on the main framework of the game -

30 September

Start working on GUI using Java Swing -

27 September





Game Wrap-up and Completion -

10-11 October

Combine front-end with back-end - integrating GUI with game logic

8 October

**No.8**

**No.7**

1. The first Board: The grid of the game must be displayed on screen
2. Code 2D Arrays: The background 2D arrays for relating the on-screen grid to the progress of the players must be set up.
3. Let there be X’s and O’s: Just display an X or O on the correct cell clicked.
4. Click to array: The cell clicked on the screen should transfer input to be stored to our 2D arrays.
5. Winner, Winner, Winner!!!: Algorithm to switch players and determine the winner should be implemented.
6. U’s and I’s: Improve on the UI to be user friendly and look better.
7. The main menu: The main menu of the game should be created and works properly.
8. XO celebration: We are done with the project.

MODEL USED:

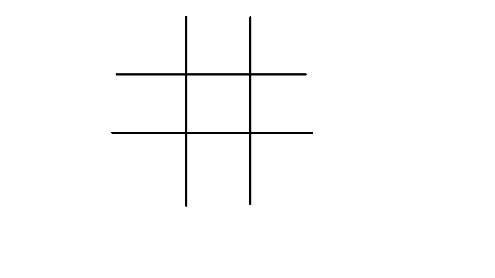
The incremental model was used in the development of this project. This means the requirements for Ultimate Tic-Tac-Toe were split into different parts; where what is required, designed, implemented and tested on each specific part was assessed. We used the incremental model because it helps prioritize the requirements and also because it makes it easy to recognize errors. Another main reason for using the incremental model is due to the project team not being very well skilled or trained in software development. Thus, the incremental model helped us in being more flexible and better at managing due to risks being managed during iteration. Finally, the incremental model made it very easy to test and debug the code at each stage.

VISUAL GUIDE:

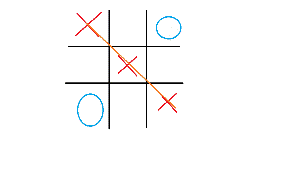
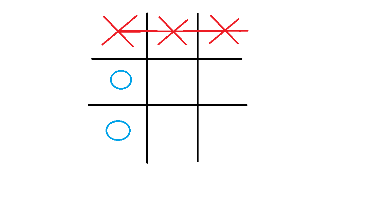
How the game works - The rules of ultimate Tic-Tac-Toe

**The board:**

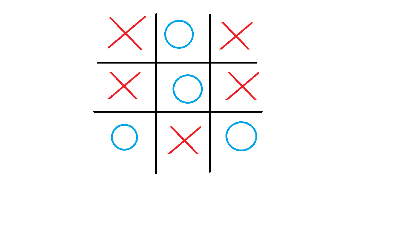
A standard tic tac toe board is composed of a 3 x 3 grid.



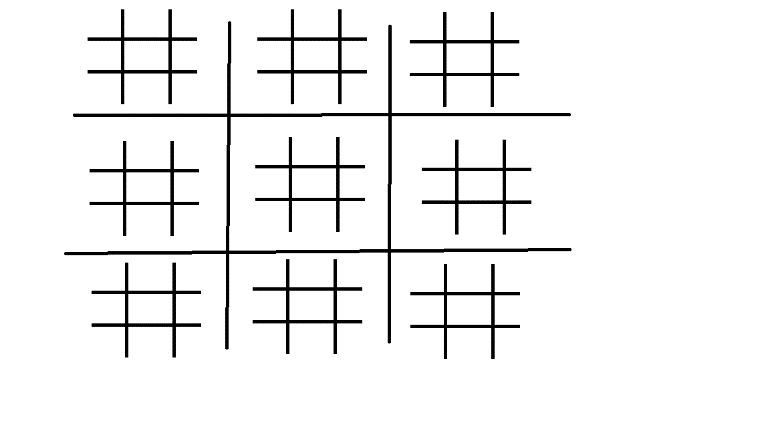
Whereby both players take a chance to play 1 usually X and play 2 usually O. A win is determined by either X or O in the same row. For example, X wins in these situations (O wins in similar situations).



And a draw is achieved when both players have nowhere to move, for example:

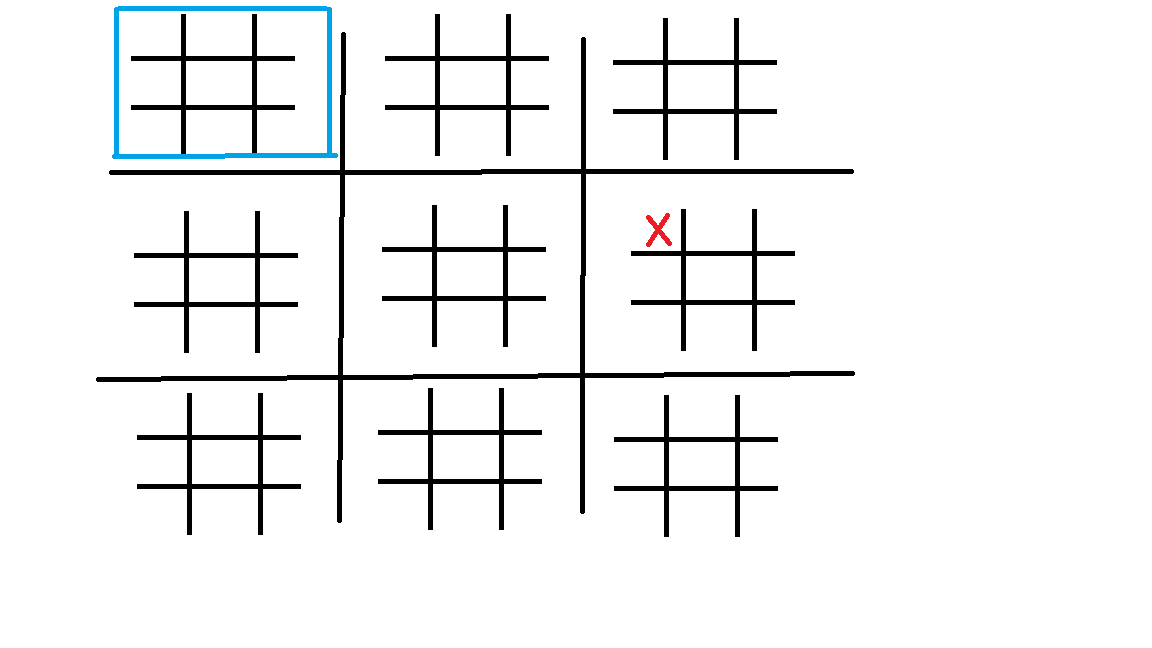


Ultimate Tic-Tac-Toe is made up of a big 3 x 3 grid, with each square consisting of a standard 3x3 grid.



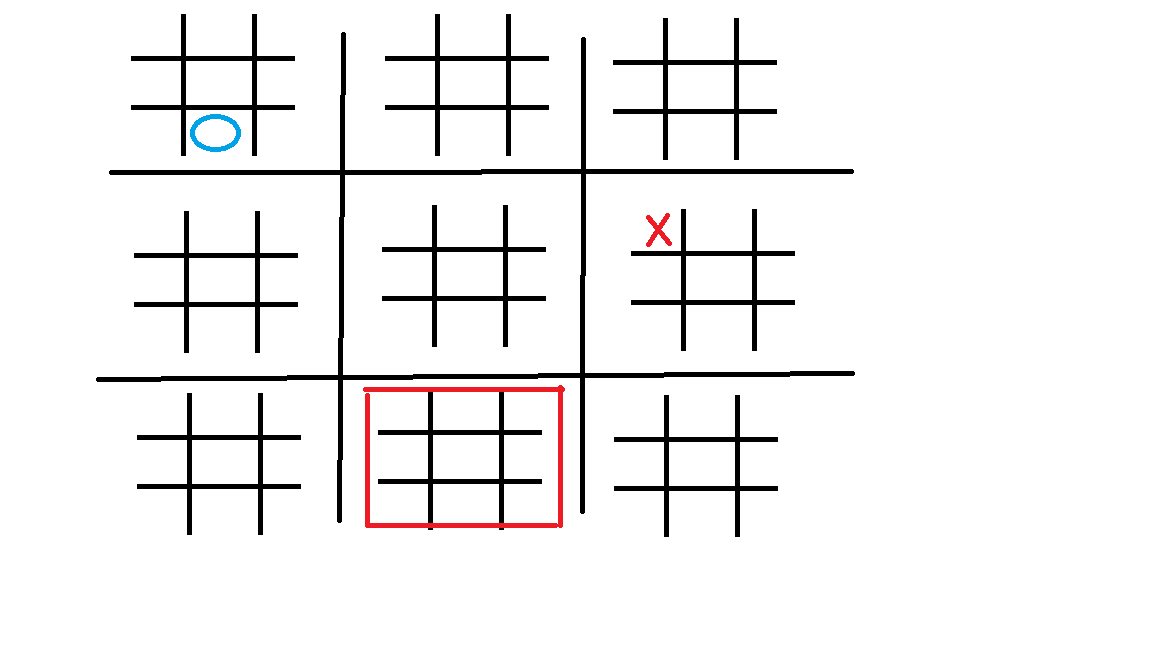
**Moving:**

The player (player 1) who goes first uses X gets and on the first move is allowed to play anywhere on the board. The second player’s move depends on either of the smaller boards which player 1 plays. For example, if player 1 chooses to play in the northwest corner of one of the smaller boards, player 2 is forced to play in the northwest corner of the larger board (the corner highlighted in blue).



From here onwards both players depend on where the other player plays.

For example, let's say O (player 2) decides to move in the southern cell of the northwestern board. X is then forced to move to the southern cell of the larger board.



**Draws in the smaller boards:**

If a draw is achieved in either one of the smaller boards then the players will not be able to play in that board, however, if one of the players chooses to play in the larger square where the draw is then the other player gets to choose any square to play in where there is no draw or win.

**Wins in the smaller boards:**

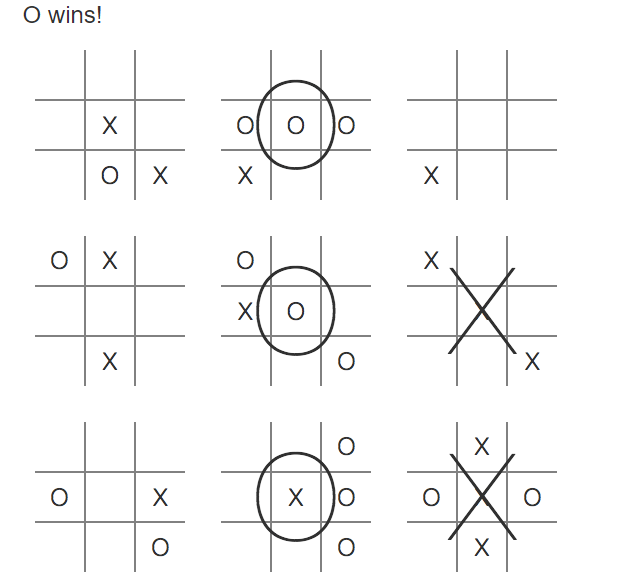
Wins in the smaller boards are obtained in a similar manner to the standard tic tac toe, however, wins in the smaller board do not mean that a player has won the game. Like draws in the smaller boards, if a player chooses to force the other player to play in the larger board where a draw is obtained then the other gets to choose any smaller board to play in where there is no draw or win.

**Draws in the bigger board:**

A draw on the bigger board means that the game has ended but there is no winner or loser between the two players.

**Wins in the bigger board:**

A win in the bigger board is obtained when one of the players manages to win in smaller boards that are either linear, diagonal, or vertical to one another. An example of win is like:



References

Marcus, M. (2021, March 4). *History of Tic Tac Toe - Learn the Lore Right Here*. Cool Math Games. Retrieved September 4, 2022, from <https://www.coolmathgames.com/blog/tic-tac-toe-history-three-in-a-row-thru-the-ages>

B, Zhang. (2016). Ultimate tic tac toe rules. Retrieved from UltimateTicTacToe:

[https://ultimate-t3.herokuapp.com](https://ultimate-t3.herokuapp.com/)