

Assignment 3 Report:

Functionality:

A regular implementation of a 5x5 tic-tac-toe game is difficult to win even against a low-depth AI. This stems from the fact that setting up a winning position takes many turns and is easily foreseen and accordingly blocked by the opponent. Thus, in attempts to spark more anticipation and skill into the forever drawing game, this code implements player powerups that unexpectedly shift the dynamics of the game after just one turn. The game has four powerups incorporated and are only made available for human players as the heuristic of an AI that can consider the possibility of four power ups every move would be unrealistic given the time constraint of the assignment.

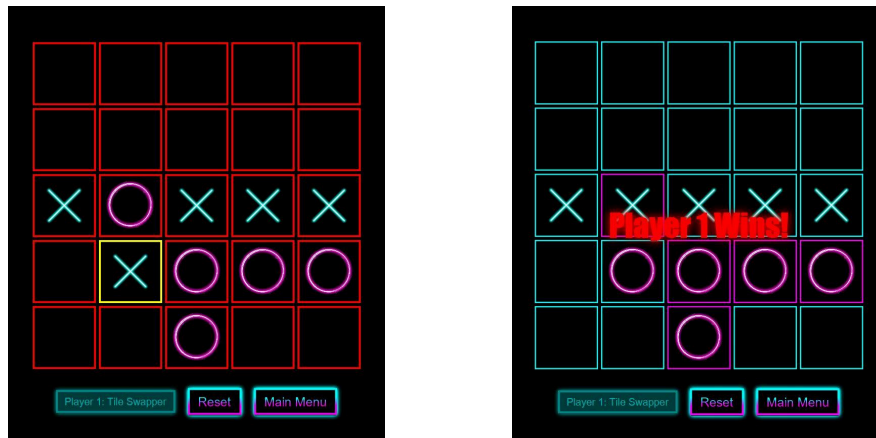


When the game is initialized a main menu is shown that allows the user to select the player roles (human or AI) as indicated by the icon on the left hand side, as well as the difficulty of the AI or the powerups of the human player. The difficulty accordingly scales the depth of the AI, foreseeing higher depths allows the AI to make more calculated moves by expanding its search tree. When the user hovers over the powerups, the names are provided as follows:

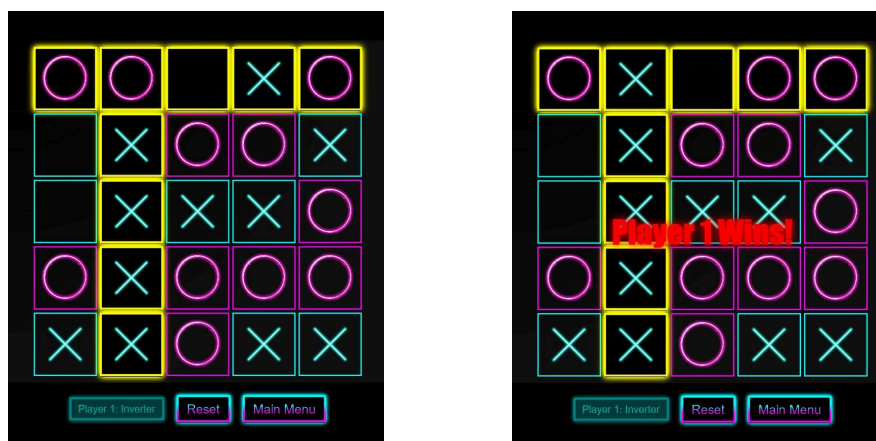
1. Tile Swapper → Swaps any two tiles, regardless of symbol or position
2. Tile Reserver → Stops any symbol from being placed in the reserver tile for four turns
3. Inverter → Inverts the symbol order of the row, column, or diagonal selected by the player
4. Tile Remover → Randomly remove two tiles from the opponent, adding a risk factor to the game

The inclusion of the powerups brings countless possibilities to a regular game of 5x5 tic-tac-toe, providing a new experience for every combination of powerups and AI difficulty. The inclusion of powerups required minute changes within the logic of a 5x5 game. For example, the 'Tile Swapper' and the 'Inverter' ability created scenarios in which the usage of one of these powerups results in both players winning simultaneously which is not possible in the vanilla 5x5 game. Such scenarios were handled by checking whether each player had won after the powerup was used and a draw was given if both players won from one move.

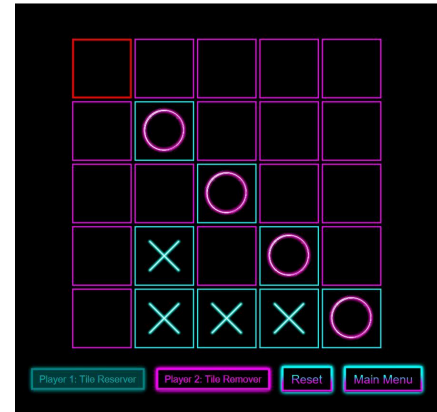
The below screenshots are an example of a 'Tile Swapper' being used by Player 1. When the powerup button is pressed to activate the ability, the grid changes colors to indicate to the player to select two tiles to swap, the first tile selected is indicated by a yellow border. By the second click, the swap is immediately made and the game state is checked for terminality. The 'Tile Swapper' ability also allows players to swap empty tiles, enabling them to displace their opponents if their own tiles are already well-positioned.



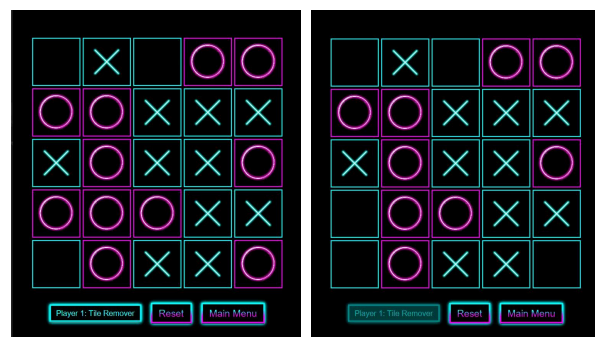
The 'Inverter' powerup creates the most interesting dynamic to the game, as now players need to consider the symmetry of the board, prioritizing tiles that are third in the row, column, or diagonal, as the inversion would not affect them. However, if an opponent blocks a winning position using a non centered tile, the player can react by placing a symbol in the opposite side of the row, column, or diagonal and use the inverter to win the game. When the powerup for the inverter is played, the user is indicated to select any tile on the board. Once the user presses any tile, the rows, columns, and diagonals (if possible) corresponding to the selected tile are highlighted by yellow borders. In the example below, the circle in the second column on the top row is selected by the inverter as it is blocking the player from winning. After the feasible rows, columns, or diagonals are highlighted, the user is required to press on any of the lines of the feasible highlighted lines to be inverted. In the case below, any tile pressed alongside the top row will invert the row, resulting in a win for Player X.



The screenshot on the right-hand side showcases the utility of the ‘Tile Reserver’ which can potentially allow the player to block their opponent without needing to place their own tile, providing them an opportunity to stop or at least delay their winning position. Surprisingly, this was by far the most troublesome ability to implement. Intuitively, the functionality of the powerup was to allow the player who reserved a tile to be able to place their own symbol in the tile but after hours of debugging and researching, I was never able to properly implement this feature, thus the functionality of the ‘Tile Reserver’ now blocks all players from placing a symbol in a reserved tile.



The most chaotic powerup is the ‘Tile Remover’ that introduces a level of luck and risk in the game, removing two of the opponents tiles at random. This powerup can potentially lead to an easy win for the user, freeing any winning line that is blocked by only one line, or could potentially lead to no influential move, as seen by the example of the right hand side in which two circles in uninteresting positions were deleted.



Extensions:

There are numerous avenues that can be extended given more time. One such idea can be the implementing a heuristic for the AI that can account for the powerup selected by the player to allow the AI to use the powerups as well. However, the game is already fun enough as it is, thus further extensions such as animations for the powerups, and sound design would greatly improve the user experience. Sound effects for the mouse hovering over buttons, clicking buttons, background music, and sound design for the powerups can greatly elevate the game if given more time to explore the features of the JavaFX library. Animations particularly can be interesting, allowing limitless possibilities for accentuating powerups in the game.