

CCPROG1 Machine Project

Silly Tic-tac-toe

Due on Jan. 25, 2021, 11:59pm

Make a 2-player 5x5 (not 3x3) grid tic-tac-toe game, where each player can choose from a menu of special abilities at the beginning of the game, and use those abilities in lieu of their regular turn during the game.

At the start of the game, allow player 1 to choose from a list of abilities, and then allow player 2 to choose the same number of abilities player 1 chose, from the same list.

List of abilities (you can change their names to something more creative)

1. Take 2 turns – Player gets two turns after using their normal turn to use this ability (so just a net gain of only one extra turn).
2. Flip a space – Player chooses a space occupied by an opponent, and then replaces the opponent's symbol with their own.
3. Blank four corners and center – Removes all symbols from the four corners and center of the grid.
4. Free row – Player chooses 2 spaces they occupy that are within the same row, and then the computer removes all the symbols of the opponent within the same row.
5. Opponent chooses 3 for you – Lets the opponent choose three unoccupied spaces for the player (instead of the normal choice of only one space).
6. Even it out – Computer counts the number of spaces each player occupies, and if the opponent occupies more spaces than the player, the player can select spaces occupied by the opponent and those spaces become unoccupied and the opponent's symbols removed from them, until they have equal number of spaces.
7. Delete ability – Player chooses and removes an unused ability from the opponent's abilities so the opponent can no longer use it.
8. Counter – when an opponent plays an ability, the player with the counter ability will be asked by the computer if they would like to counter it, and if they agree, the opponent will not be able to use the ability, can no longer use that ability, and will instead take a normal turn. This ability can only be used during the opponent's turn.

The game proceeds similarly to normal tic-tac-toe, where each player takes turns selecting an unoccupied space, except there are 25 spaces, and a player needs to occupy 5 spaces in a horizontal, vertical or diagonal row to win. Also, instead of taking their usual turn, they may use one of their chosen abilities instead. Once a player uses an ability, that player cannot use that ability again.

At the beginning of the game (before abilities are chosen) ask the players if they would like the abilities to be hidden, wherein if they will be hidden, during the game, reference the abilities by a generic label instead, such as ability 1. (In this case, the players should look away while a player is choosing their abilities, and they should remember the generic names given by the computer to each ability they choose.)

Each space in the grid can have a label, either 1 to 25 or A to Y, and players can select a space by typing the label of the space they choose. If the chosen label is invalid (for example, it does not exist or it is already occupied) the player should choose another space until the player chooses a valid space.

The game ends when a player occupies all five spaces in a horizontal, vertical or diagonal row, or all 25 spaces have been occupied.

TECHNICAL REQUIREMENTS:

- Each ability must be defined in its own function.
- There must be sufficient internal documentation.
- The code must be efficient.
- Must follow CCPROG1 Coding Standards available here:
https://dlsu.instructure.com/courses/46150/discussion_topics/260438

OTHER REQUIREMENTS:

- Screens must be user-friendly
- Create a Test Case document, and save it as PDF. For each function, identify appropriate test cases/scenarios. Test and validate your functions. Example:

Function: function declaration/signature

Test Description	Input value/parameters	Expected output/result	Actual output/result	Pass/Fail
Case 1 desc				P or F
Case 2 desc				
:				