# Introduction:

For my game I decided to make TicTacToe. This is because I didn’t have a group to work with it seemed my simplest option.

# Design Documentation:

## Problem Analysis:

### Scope of the problem:

There were many problems when making this game as it was completely new territory for me.

### Game Rules:

### Success Criteria:

My success criteria was that the game must be able to play played by 1 or 2 ai and human players

It should be able to be played on any machine regardless of machine it was built on

It should be Well structured and readable.

## Technical Design

### Program Structure:

My program has a split architecture system so that the Rendering and game logic are separated. This means that

My game also support grids of n sizing though the logic would need scaling up with such things.

### Programming Concepts:

### Game Structure:

The games structure isn’t that complicated each class has its own property’s which they all use and inherit. The only thing different is that I don’t use any console WriteLines outside of the renderer this means that the graphic rendering is completely separate from the logic so it can be changed out.

The methods that make move and render things etc are all abstract, this means that when/if the renderer is changed that they will simply adapt to this.

### Move Validation:

The logic validated moves by assigning players to them. So it checks the cell to see if it is occupied if not assign current player to it. But it automatically assigns a placeholder “Player.None” this is so that there are no null references and hence exceptions.

### Graphical User Interface:

It simply uses the console to print a game board to it and refreshes each time a move is made.

### AI:

The AI is simple but effective. It will block you if you’re about to win otherwise it will pursue its own win. Usually if its AI vs AI it ends in a stalemate.

### Implementation:

For implementing everything I used properties since these would be publicly accessible so that all the classes could use them to implement them/

## Extensibility:

The game logic could be extended to support bigger grids and different types of game.

I could make it so it support things such as powerup cells that would change the game in some way or blocking cells that couldn’t be occupied

I could implement different types of Ai along with this. So they play more tactically or aggressively.

The game grid could also be a different shape which could make the game more fun

Given time I could make the plane so that its 3D tic tac toe or even Graphically render in with something like OpenGL

Bibliography:

Since I made the game a while ago and only added some bit recently, The sources I used then were

<http://stackoverflow.com/>

<https://msdn.microsoft.com/en-us/library/618ayhy6.aspx>

and various google searches.