Project 3
CSC 3150 - Game Development
Grading: 150 points
Due Date: April 3th, 2019.

Description: This project is an individual project designed to reinforce the theories and practices learned in class. You will be implementing the minimax algorithm in Unity3D in an adversarial Artificial Intelligence game. The game you will be playing is the game of Tic-Tac-Toe. The goal of the game is to get three X's in a row, while preventing your opponent from achieving three O's in a row. As a first step, you will create two prefabs, the X and the O prefabs. These can be simple polygon spheres and rectangles. You should also create a grid to establish the game board and playing space, see Figure 1.

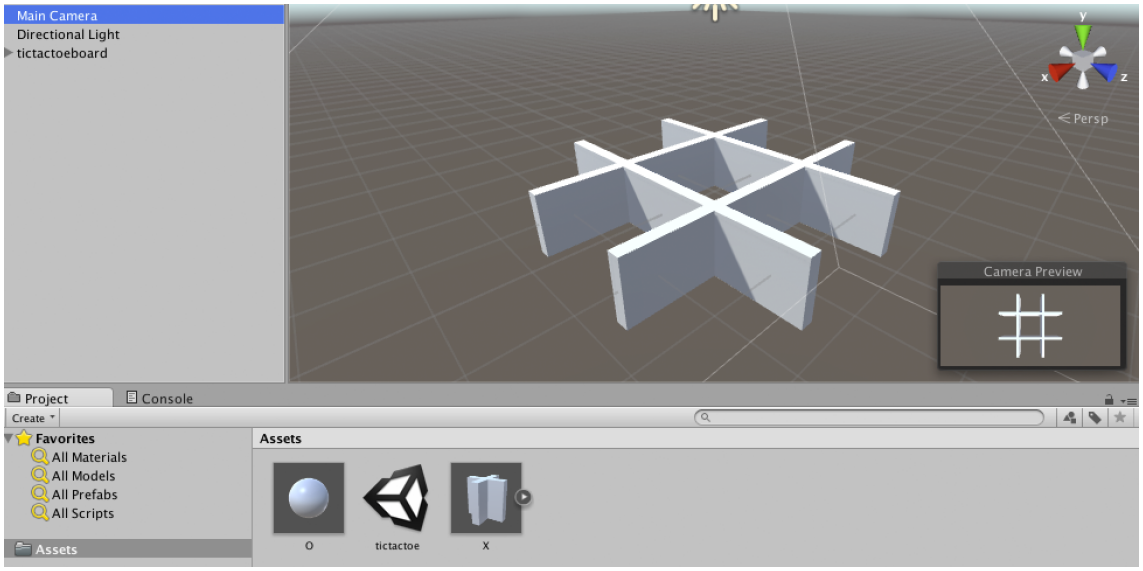

Figure 1: Visualization of the setup of the Unity scene given the game board and X and O prefabs.

Set the position of your camera to look down on your game board. Create a script that will accept the input of the user to place the X prefabs. For example, use the code

```
if(Input.GetKeyUp (''q''){
        Instantiate (xprefab, new Vector3(-2,1,2), Quaternion.identity);
}
```

to test whether or not the "q" key has been pressed to instantiate an X prefab in a certain Vector3 location that fits within your prebuilt grid. In this example, I am instantiating an X prefab in X=-2,Y=1,Z=2 position in the world with zero rotation (Quaternion.identity). You should use the following keys for placing the X's.

Top left = q, Top middle = w, Top right = e, Middle left = a, Middle middle = s, middle right = d, bottom left = z, bottom middle = x, bottom right = c.

After each placement, you should run the minimax algorithm to compute the optimal placement of the O. You should not need to perform any pruning or cut-off limitations as the maximum tree search space of tic-tac-toe is only 9 factorial. You should use C# to complete this assignment. Some useful methods include generic lists, and tic-tac-toe node definitions,

```
public class MNode : IComparable<MNode> { // example of generic class definition using
    IComparable interface in C#

List<MNode> result = new List<MNode> (); // example of a generic list using MNode in C#
```

Deliverables You are responsible for submitting,
1. A script file that accepts user input and then computes the minimax optimal move for the opponent. The script should instantiate X and O prefabs in the scene in the appropriate positions. You should use C# to complete this assignment.
2. Screen shots in JPG or PNG form that illustrate your method working at various game states. See Figure 2 for examples.

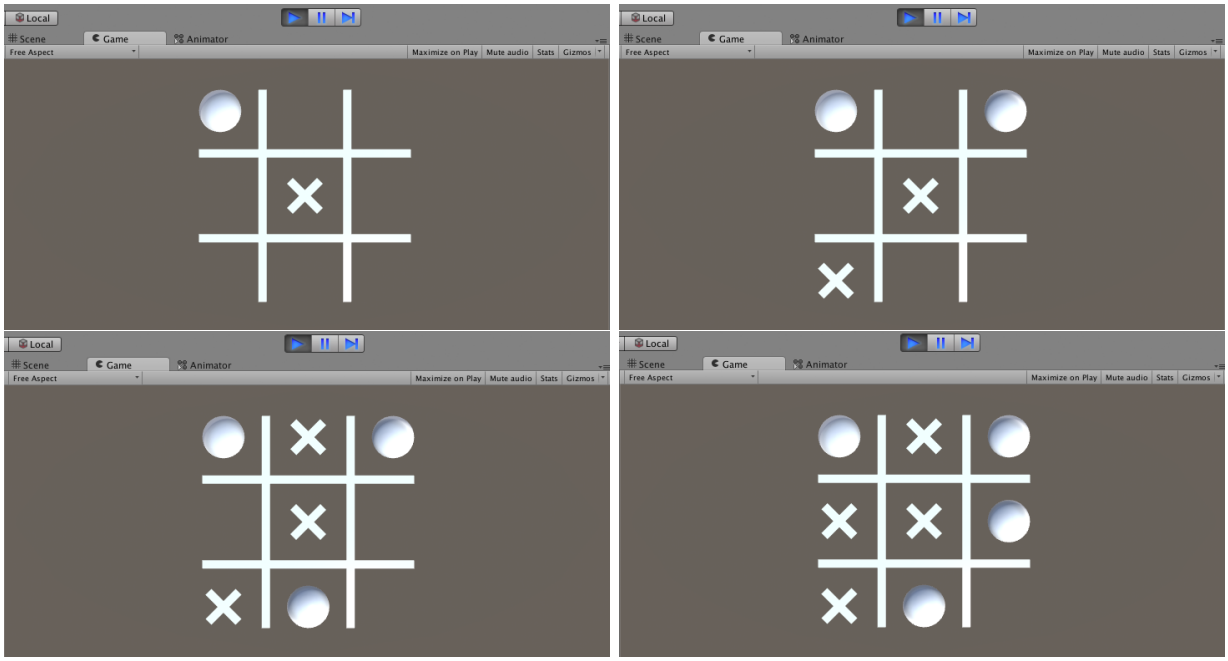

Figure 2: Visualization of the scene when running the adversarial code. The user is playing as Xs and the computer is playing as Os. X goes first (you). For these images, the main camera is set to Orthographic mode to remove perspective.

Submission Submit the files on Blackboard.