ITCS 4236/5236 Artificial Intelligence for Games  
Homework #4

**Instructions**Using the provided Bomber Belts Unity Project, write your own AI Script for the character to play the game. Your script will be used on a character while another classmate’s script will be used on the opposing character. Please refer to the information below on how and where to write the code.

1. Download the Bomber Belts Project from Moodle
2. Open the Scene named BomberBelts inside of the Assets folder
3. Locate the *AIScript.cs* file inside of the Scripts folder
   1. This is the script you will modify to create your AI algorithm
   2. Refer to the information below on available functions to call
   3. **Change the name of the script to *AIScript\_YourName*.*cs***
      1. **You will also need to change the class name**
4. **Do NOT make any modifications to any other scripts in the project**
5. Add your AIScript.cs file to one or both of the characters to test it.
   1. Select *BlueCharacter* or *RedCharacter* in the Hierarchy pane
   2. In the Inspector pane, scroll down to the bottom and click on “Add Component”
   3. Select “Scripts” -> *AIScript.cs*

**Any modifications to the speed of the bombs or players, or any adjustments to the game that grants an unfair advantage to a player will automatically result in a zero for this assignment.**

|  |  |
| --- | --- |
| void *MoveUp()* | Moves the player up. The player will continue to move up until otherwise instructed. |
| void *MoveDown()* | Moves the player down. The player will continue to move down until otherwise instructed. |
| void *Push()* | Attempts to push the closest button. If the character is too far from the button, or the button is already engaged, nothing will happen. |
| Vector3 *GetCharacterLocation()* | Returns the position of the character as a Vector3 (x, y, z) |
| Vector3 *GetOpponentLocation()* | Returns the position of the opposing character as a Vector3 (x, y, z) |
| Vector3[] *GetButtonLocations()* | Returns an array of Vector3s for representing the position of each button on your side |
| bool[] GetBeltDirection() | Returns an array of Boolean values that corresponds to whether or not the buttons on your side of the board have been engaged. True means the belt/button is engaged and the bomb is moving towards your opponent. |
| float[] GetBombLocation() | Returns an array of float values that represent the distance each bomb is from its corresponding button on your side |
| float GetPlayerSpeed() | Returns the speed at which the characters move |
| float GetBombSpeed() | Returns the speed at which the bombs move |

**Coding Framework Information**Inside of the *AIScript.cs* file you will find two functions: *Start()* and *Update()*. *Start* is run once when the game is loaded, and *Update* is called once per game step. All of the functionality for moving your character, sending bombs, and positional/state information of the belts and bombs have been written for you inside of the *GameScript.cs* file. You can access those functions by using the *gameScript* variable that is already declared and initialized for you inside of the *AIScript.cs* file. Below is a complete list of functions you can call.