ITCS 3153 Introduction to Artificial Intelligence

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 2014 Project from Moodle
2. Open the Scene named Menu inside of the Assets folder
3. Locate the *AI_Template.cs* file inside of the Assets / Resources / AI Scripts folder
    a. This is the script you will modify to create your AI algorithm
    b. Refer to the information below on available functions to call
    c. You may also use the *AI_Sample.cs* script to see an example implementation
    d. Change the name of the script to *AIScript_YourName.cs*
        i. You will also need to change the class name
4. **Do NOT make any modifications to any other scripts in the project**
    a. 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.
5. To test your script, simply run the Menu screen and select your script.

Coding Framework Information

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 other scripts. You can access those functions by using the *mainScript* variable that is already declared and initialized for you inside of the *AI_Template.cs* file. Below is a complete list of functions.

`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, the button is already engaged, or is on cooldown, nothing will happen.
`float getCharacterLocation()`	Returns the position of the character as a float
`float getOpponentLocation()`	Returns the position of the opposing character as a float
`float[] getButtonLocations()`	Returns an array of floats for representing the position of each button on your side
`float[] getButtonCooldowns()`	Returns an array of floats representing the time remaining before each button may be pressed again.
`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[] getBombDistances()`	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

Submission Information

Make sure that you've changed the name of the script to *AIScript_YourName*.cs and that you've also changed the class name to match the file name. Upload your completed AI script to Moodle.

A Few Points

- Whenever a button is pressed, that belt becomes disabled for 1.0 second. Neither player may press a button on that belt until the second has elapsed. Example: the blue player presses the button on belt 1. Both the blue player's button 1 and the red player's button 1 become disabled for 1.0 second.
- You can access the list of each bomb's current cooldown by calling `getButtonCooldowns()`.
- Calling `moveUp()` at the very top will have no effect. This is likewise true for `moveDown()` at the bottom.
- Make sure your script has elements of AI. Simply copying what the opponent is doing (i.e., following) or doing the same time every time regardless of what environmental percepts dictate is not AI.