

AIML(Program #) for Unity 3D – tutorial for PC

How the AIML works?

You can find the answers here:

<http://www.alicebot.org/aiml.html>

<http://www.alicebot.org/documentation/aiml-reference.html>

How the C # interprets the AIML files?

You can find the answers here:

<http://aimlbot.sourceforge.net/>

How the Program # - AIMLbot.dll was implemented in Unity3D?

You must be wondering why you are not finding the "AIMLbot.dll" file. Well, that's because if you want to change something in a class, you need not open the original project and re-build of the DLL.

Now, answering the question, first class has changed "**Bot.cs**" by adding the following:

Two variables were created:

- *private string **MyPath***

Defines the path where the AIML folders and config will stay. These folders are responsible for contain all the files responsible for the operation of your bot.

```
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

    /// <summary>
    /// Loads settings based upon the default location of the Settings.xml file
    /// </summary>
    private string MyPath = Application.streamingAssetsPath;
    1 reference
    public string ChangeMyPath
    {
        get
        {
            return MyPath;
        }
        set
        {
            MyPath = value;
        }
    }
    1 reference
    public void loadSettings()
    {
        // try a safe default setting for the settings xml file
        string path = Path.Combine(MyPath, Path.Combine("config", "Settings.xml"));
        this.loadSettings(path);
    }

    /// <summary>
```

The "**MyPath**" variable was inserted elsewhere in class "**Bot.cs**". See the image below:

```
263 | 1 reference
    | public string PathToAIML
264 | {
265 |     get
266 |     {
267 |         return Path.Combine(MyPath, this.GlobalSettings.grabSetting("aimldirectory"));
268 |     }
269 | }
270 |
271 | /// <summary> The directory to look in for the various XML configuration files
    | 6 references
274 | public string PathToConfigFiles
275 | {
276 |     get
277 |     {
278 |         return Path.Combine(MyPath, this.GlobalSettings.grabSetting("configdirectory"));
279 |     }
280 | }
281 |
282 | /// <summary> The directory into which the various log files will be written
    | 2 references
285 | public string PathToLogs
286 | {
287 |     get
288 |     {
289 |         return Path.Combine(MyPath, this.GlobalSettings.grabSetting("logdirectory"));
290 |     }
    | }
```

This allows other necessary files are found.

```
527 | // Load the dictionaries for this Bot from the various configuration files
528 | this.Person2Substitutions.loadSettings(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("person2substitut
529 | this.PersonSubstitutions.loadSettings(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("personsubstitutio
530 | this.GenderSubstitutions.loadSettings(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("gendersubstitutio
531 | this.DefaultPredicates.loadSettings(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("defaultpredicates"
532 | this.Substitutions.loadSettings(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("substitutionsfile")));
533 |
534 | // Grab the splitters for this bot
535 | this.loadSplitters(Path.Combine(this.PathToConfigFiles, this.GlobalSettings.grabSetting("splittersfile")));
536 |
537 |
538 | <summary>
539 | Loads the splitters for this bot from the supplied config file (or sets up some safe defaults)
540 | </summary>
541 | <param name="pathToSplitters">Path to the config file</param>
```

And the variable:

- `public string ChangeMyPath`

When you instantiate the "**Bot.cs**" class, the variable "**ChangeMyPath**" is the one that will get the folder path or change it.

Here is an example of how it would look in another class:

You can find this example in class "ChatbotPC.cs".

```
38 // Loads all the AIML files in the \AIML folder
39 1 reference
40 public void Initialize()
41 {
42     AimplBot.ChangeMyPath = Application.streamingAssetsPath;
43     AimplBot.loadSettings();
44     AimplBot.isAcceptingUserInput = false;
45     AimplBot.loadAIMLFromFiles();
46     AimplBot.isAcceptingUserInput = true;
47 }
```

IMPORTANT: All related to Bot (AIML and the config folder) files must be in [StreamingAssets](#) folder.

Save and Load

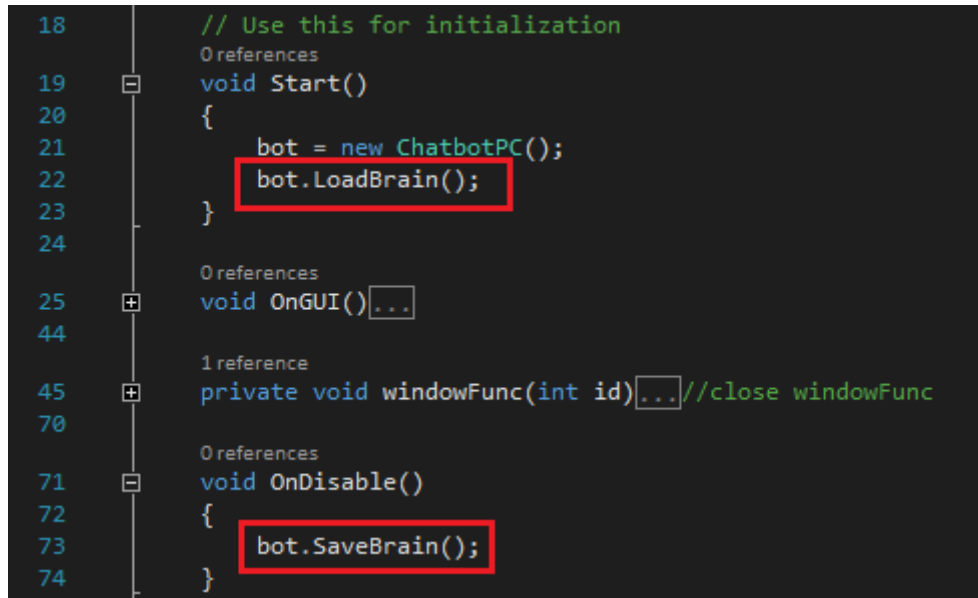
You can find this example in class "ChatbotPC.cs".

```
56 1 reference
57 public void SaveBrain()
58 {
59     try
60     {
61         myUser.Predicates.DictionaryAsXML.Save(pathToUserSettings);
62         Debug.Log("Brain saved");
63     }
64     catch (Exception e)
65     {
66         Debug.Log("Brain not saved");
67         Debug.Log(e);
68     }
69 }
70
71 1 reference
72 public void LoadBrain()
73 {
74     try
75     {
76         myUser.Predicates.loadSettings(pathToUserSettings);
77         Debug.Log("Brain loaded");
78     }
79     catch (Exception e)
80     {
81         Debug.Log("Brain not loaded");
82         Debug.Log(e);
83     }
84 }
```

An XML file is created for the bot remember all the information that the user entered during the conversation, such as name, age, country, etc ...

This file is loaded when you start the conversation with the bot and saved when the scene is closed.

You can find this example in class "**ChatWindowExamplePC.cs**".



```
18 // Use this for initialization
19 0 references
20 void Start()
21 {
22     bot = new ChatbotPC();
23     bot.LoadBrain();
24 }
25
26 0 references
27 void OnGUI()...
44
45 1 reference
46 private void windowFunc(int id) ... //close windowFunc
70
71 0 references
72 void OnDisable()
73 {
74     bot.SaveBrain();
75 }
```

This solution was based on this response:

<http://stackoverflow.com/questions/11184984/aimlbot-c-sharp-save-brain>

You can now open the **ChatbotPC.cs** class for more information. In class **ChatWindowExamplePC.cs** You can find an example of how to create a bot in your project.

If you are interested in developing mobile or web, you can go to the tutorial:

AIML(Program #) for Unity 3D – tutorial for mobile and web

Maybe you might like to see this:

<https://xinyustudio.wordpress.com/2014/03/08/get-started-with-aiml-c-programming-i-just-say-hello-to-the-robot/>

<https://mattg19.wordpress.com/2011/03/07/using-aiml-in-c-a-program-tutorial/>