Welcome to my Achievements System Plugin !

This is a simple but cool achievements system :)

- What’s inside ?

→ In this achievements system you will find some editor windows to help you setup your achievements, some scriptable objects you can rely on to check your data, and an easy way to use the achievements system in your scripts !

Check how it works below !

DISCLAIMER :

Please don't use the editor windows of this plugin while you have compilation errors, because it may not behave as expected and can cause some mess around generated assemblies and scriptable objects. (Nothing to be afraid of, just mess)

1 – Generate your achievements data glossary

You have a first editor window in here -> "Tools/DiZ Tools/Achievements System/" :

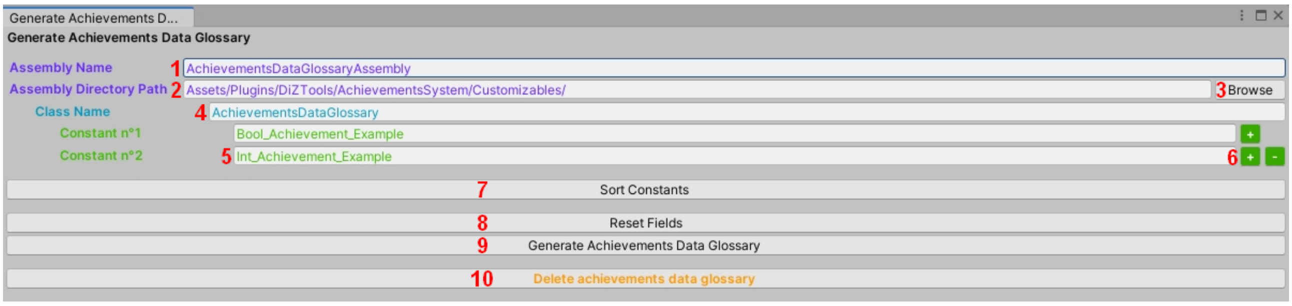
- "Generate Achievements Data Glossary Window"

In here, you will be able to setup and generate your achievements data glossary. Its means you will have a class filled with "string constants" representing the achievements data you want to rely on.

For instance, you could have a "Number of steps", a "Number of jumps" and whatever "int data" you want to keep for your achievements, and also a "level 1 succeeded", a "level 2 succeeded" and whatever "bool data" you want to keep for your achievements.

Note : I know "bool data" can look the same as an actual achievement, but it is not. Those are just data used by some achievement which would want to know if it should trigger or not.

After generating these data, you will be able to bind them to some achievements, and use them in your scripts ! (See details later below)



Generate Achievements Data Glossary

Assembly Name: AchievementsDataGlossaryAssembly

Assembly Directory Path: Assets/Plugins/DiZTools/AchievementsSystem/Customizables/

Class Name: AchievementsDataGlossary

Constant n°1: Bool_Achievement_Example

Constant n°2: Int_Achievement_Example

Generate Achievements D...
1
2
3
Browse
4
5
6
+
+
-
7
Sort Constants
8
Reset Fields
9
Generate Achievements Data Glossary
10
Delete achievements data glossary

In order to generate your achievements data glossary, you have, according to the window :

1. Assembly Name : This is how your assembly (.dll file) will be or is named.

2. Assembly Directory Path : The path to the assembly.

3. Browse Button : It will help you browse a path for your assembly.

4. Class Name : This is how your class name containing the string constants will be or is named. It is that name you use in your scripts to get access to its values.

5. Constants : This is all the "string constants" you want corresponding to the achievements data you will have.

6. + / - Buttons : Add a constant below, or remove the constant from the class.

7. Sort Constants Button : It sorts the constants by alphabetical order.

8. Reset Fields Button : It reset the fields above, as if you just opened the window with your current assembly info (or not if you didn't already generate one).

9. Generate Achievements Data Glossary Button : It can't be more explicit. No need to delete the old one, all is handled by the generation.

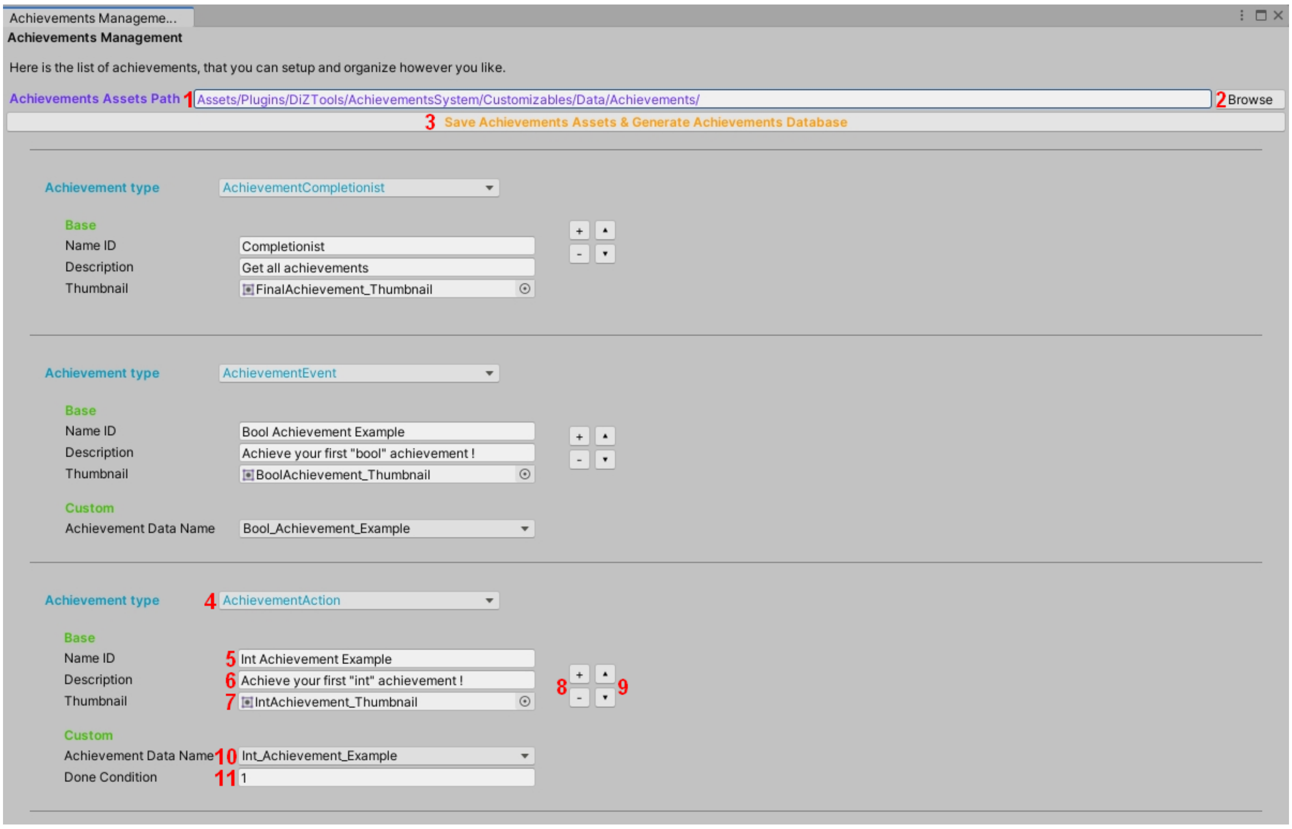
10. Delete achievements data glossary Button : This button will find and delete your current achievements data glossary assembly.

2 – Achievements Management Window

A second editor window will help you setup your actual achievements :
(still in here -> "Tools/DiZ Tools/Achievements System/")

- "Achievements Management Window"

In here, you will be able to create and setup your achievements, their location, their types, their info, their order in the list, their achievement data they are linked with, etc.



Achievements Manageme...

Achievements Management

Here is the list of achievements, that you can setup and organize however you like.

Achievements Assets Path: Assets/Plugins/DiZTools/AchievementsSystem/Customizables/Data/Achievements/

Browse

Save Achievements Assets & Generate Achievements Database

Achievement type: AchievementCompletionist

Base
Name ID: Completionist
Description: Get all achievements
Thumbnail: FinalAchievement_Thumbnail

Achievement type: AchievementEvent

Base
Name ID: Bool Achievement Example
Description: Achieve your first "bool" achievement !
Thumbnail: BoolAchievement_Thumbnail

Custom
Achievement Data Name: Bool_Achievement_Example

Achievement type: AchievementAction

Base
Name ID: Int Achievement Example
Description: Achieve your first "int" achievement !
Thumbnail: IntAchievement_Thumbnail

Custom
Achievement Data Name: Int_Achievement_Example
Done Condition: 1

Here is the description of some important fields in the window :

1. Achievements Assets Path : This is where the achievements scriptable objects are stored.

2. Browse Button : It will help you browse a path for your achievements assets.

3. Save and Generate Button : This will save the achievements in the achievements list, AND will generate the achievements data base according to all the achievement data bound.

4. Achievement type : Select the type of the achievement. When you change this type, the old achievement scriptable object is deleted, and a new one is created with the new type.

5. Name ID : The name of your achievement. It must be unique through all of your achievements.

6. Description : The description of your achievement.

7. Thumbnail : The thumbnail for your achievement when it will appear on play.

8. + / - Buttons : Add an achievement below, or remove the achievement from the achievements list.

9. Up / Down Buttons : Switch the achievement place in the list with the achievement above or below this one.

10. Achievement Data Name : The achievement data this achievement is bind to.

11. Done Condition : The condition for the achievement to be done, relative to its type.
    - For an Achievement Action : The achievement data must be superior or equal to the done condition for the achievement to be trigger.

Notes :

- There is no Custom section for the “Achievement Completionist” type, because it is trigger when all of the others are done (no order required).

- You can have only ONE “Achievement Completionist” in your achievements list.

- There is no “Done Condition” for the “Achievement Event” because this kind of achievement triggers when the achievement data bind to it is true.

- Keep your achievements scriptable objects at one unique place (specifying by the achievements assets path), because the plugin won’t be able to get the achievements otherwise.

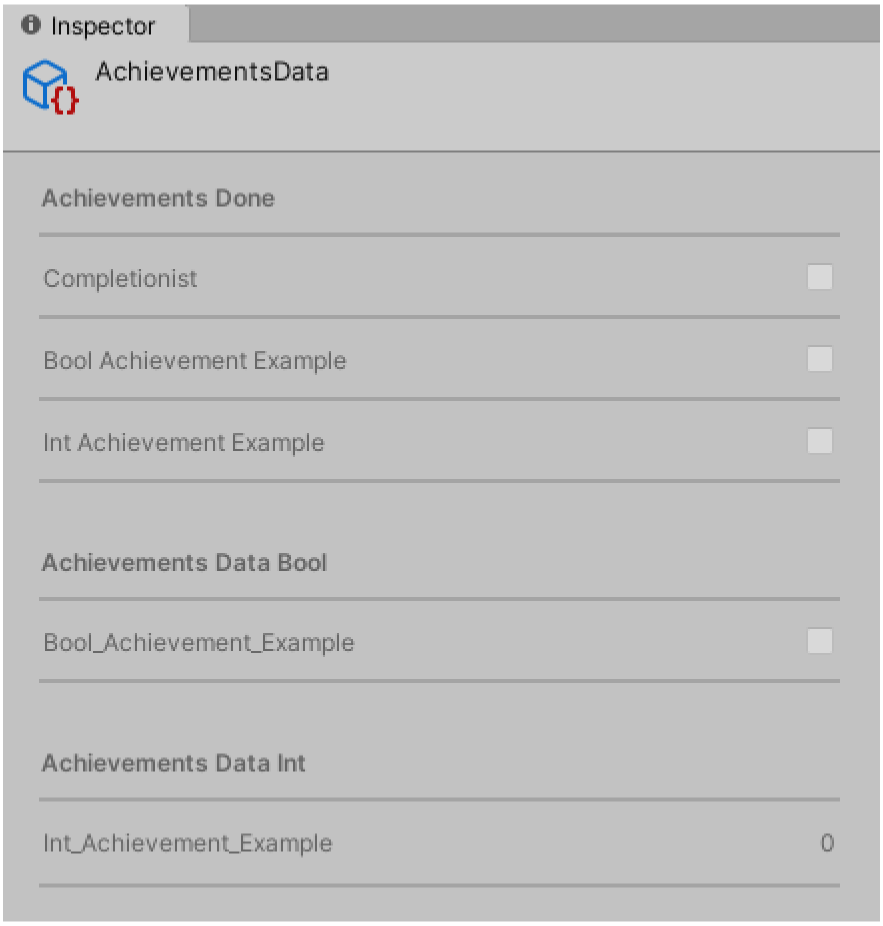
3 – Achievements Data Base

This is a scriptable object that is generated by the “Achievements Management Window”.

It shows you your current achievements data, and updates in real time on play.

IT IS NOT a scriptable you want to save with others values than the ones generated by the “Achievements Management Window”, because it will be the base of the achievements data the player will have in the beginning of the game.

Don’t worry, a real saving file for achievements data is stored in “Appdata” folder is created the first time you play the game (or launch the game in editor), and all progress is saved in there.



AchievementsData

Achievements Done

Completionist □

Bool Achievement Example □

Int Achievement Example □

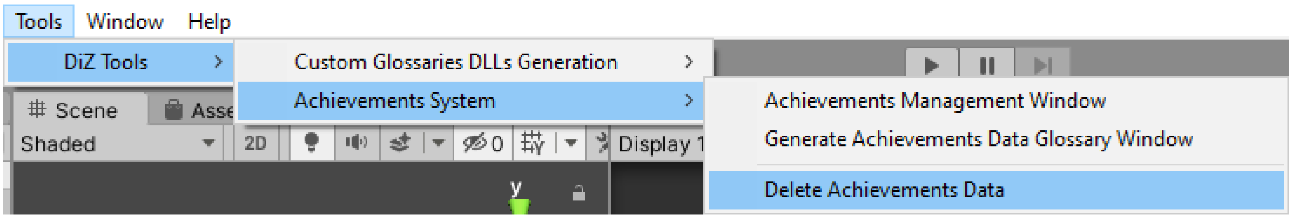
Achievements Data Bool

Bool_Achievement_Example □

Achievements Data Int

Int_Achievement_Example 0

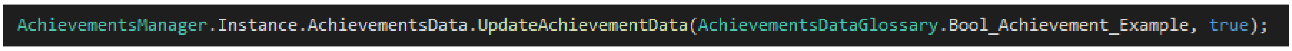
And you can always reset the saving file for achievements data in “Appdata” folder and delete it by clicking here :


Tools
Window
Help
DiZ Tools
Custom Glossaries DLLs Generation
Achievements System
Achievements Management Window
Generate Achievements Data Glossary Window
Delete Achievements Data
Scene
Asse
Shaded
2D
0
Display 1

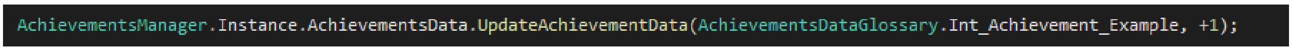
4 – Using the achievements system in scripts

The final thing you need to do in order for the achievements system to work, is to put the AchievementsManager.cs script in your scene.

And in your code, you just need to call the AchievementsManager, and tell it to update some achievements data like so :



```
AchievementsManager.Instance.AchievementsData.UpdateAchievementData(AchievementsDataGlossary.Bool_Achievement_Example, true);
```



```
AchievementsManager.Instance.AchievementsData.UpdateAchievementData(AchievementsDataGlossary.Int_Achievement_Example, +1);
```

And the system will take care of the rest :)

In the end, this plugin is not that hard to setup and realy easy to use.

Don’t hesitate to refer to this documentation if you are lost.

And you can take a look at the demo (DemoScene and DemoScript.cs) for a better understanding of how things work :)

DiZ