HD Development: Achievement Asset

Welcome to my achievement asset pack.

This asset pack can be used to create achievements for your game very quickly. It includes in game banners that appear when an achievement has been completed, four different achievement check types (ints, floats, distance, strings) and customizable achievement layout settings. This asset pack can also be used to create a game checkpoint system.

For this asset pack I used the Brecht Lecluyse's custom hide property for hiding variables.

This was retrieved from :http://www.brechtos.com/hiding-or-disabling-inspector-properties-using-propertydrawers-within-unity-5/

Table Of Contents:

Setting Up the Main Menu Scene:

1. If no canvas exists: Create a canvas
2. As a child, create an empty object and call it “Achievement List”
3. Add the “Achievements List” script to the child. (Object: “Achievement List”)
4. Add the scroll rect as a component as well (Object: “Achievement List”)
5. Create a child for the “Achievement List” object and call it “Scroll Rect”

   5.1. This object should have a pivot of 0, 0
6. In the “Achievement List” object assign all the variables

   6.1. For an explanation of the variables themselves take a look at the spread sheet.
7. To include the different sorting methods for the achievements create a new child in the canvas called “Achievement Sorting Methods”
8. Create a button for “Alphabetical Sorting”, “Completed Sorting”, “Normal Sorting” and “Type Sorting”. They all need to call the function “Sort Achievements” on gameObject “Achievement List”

   8.1. Alphabetical sorting sorts the achievement in alphabetical order and needs a value “0” in the OnClick event

   8.2. Completed Sorting sorts the achievements based on which achievements have been completed. This sorting method has a value of “1”.

   8.3. Type sorting sorts the achievements based on the achievements type (float, int, position, string). Requires a value of “2”.

   8.4. Normal Sorting sorts the list based on the order in which they appear in the “Achievements List”achievement order. Requires a value of “-1”.
9. Create a button for “Reverse List” and “Unreversed list”. They both need to call the “Reverse List Order” function on object “Achievements List”.

   9.1. Reverse List will show the sorting methods in the opposite order. Needs the bool to be true. Example: the completed sorting order displays the completed achievements at the top and with this enabled those will be displayed at the bottom of the list;

   9.2. Unreversed List will set the sorting methods back to their normal order. For this the bool needs to be set to false.

Creating the Achievement Item prefab:

1. In a placeholder canvas create an image object (UI>Image)
2. Call that object Achievement Background
3. As a child, create another image object which will be called Achievement Icon
4. As a child of the Achievement Icon create two text objects (UI>Text)
    4.1. Call the first one Title
    4.2. Call the second one Description
5. Create an empty object as a child of Achievement Background
6. Call that object Progress Display Parent
7. As a child of this object create a slider
8. Call that slider Progress bar
9. As a second child of the Progress Display Parent create a text object
10. Call the second child Progress Text
11. Disable the Progress Display Parent and both of its children
12. Drag the Achievement Background object into your prefab folder
13. The objects in the scene that were created for this can be deleted

Setting up the Game Scene:

1. Create an empty object
2. Call that object AchievementManager
3. Add the script “Game Scene Achievements” to the object
4. To create the banner first create a canvas (if one doesn’t exist)
    - 4.1.Create a child with the Image component in the canvas and call that Banner
    - 4.2.As a component of the Banner object add the “Banner Creator” script
    - 4.3.Add an image child to the Banner object and call that Flashing Effect
        1. This image should be larger than the parent image
    - 4.4.Include another image child to the banner object and call that Banner Icon
    - 4.5.Add two text object to the banner object
        1. The first one is the Header of the banner, so “Achievement Completed” would be a good header
        2. The second one is the Body of the banner
    - 4.6.Assign the variables in the Banner Creator script
5. Assign the variables for the AchievementManager

Useful Functions

1. **Game Scene Achievements**: *FindAchievementIndex(string nameOfAchievement)*;

   1.1. This function looks for the achievement with a certain name and returns the index of the achievement.

   1.2.This is important for the other functions in this script

2. **Game Scene Achievements**: *UpdateAchievementString(int achievementIndex, string value)*;

   2.1. returns true if the string sent is the same as the value in the main menu list, so when the achievement is completed

3. **Game Scene Achievements**: *UpdateAchievementPosition(int achievementIndex, Vector3 position)*;

   3.1. returns true when the achievement is completed based on the settings

4. **Game Scene Achievements**: *UpdateAchievementFloat (int achievementIndex, float value)*;

5. **Game Scene Achievements**: *UpdateAchievementInt(int achievementIndex, int value)*;

6. **Achievements List**: *StartSettingUpAchievementList()*;

   6.1.Creates the list of the achievements as long as "shouldDisplayAchievements" is true

   6.2.When this function should not be called at start then remove it from the Start() function and call it when needed

7. **Achievements List**: *ReverseListOrder(bool shouldListBeReversed);*

   7.1.When this function is called it sets the "reverseListOrder" equal to the "shouldListBeReversed" value

   7.2.When true the achievements will be displayed in the opposite order as usual

8. **Achievements List**: *SortAchievements(int sortOption);*

   8.1.Sets the sort option index to the desired value

   8.2."-1" = lists the achievements in the order as they were input into the script

   8.3."0" = Alphabetical order

   8.4."1" = Completed and not completed

   8.5."2" = Sorts by the type of achievement it is (int, float, position, string)

9. **Banner Creator**: *CreateNewBanner(string title, Sprite achievementIcon);*

   9.1.Creates a new banner when an achievement is completed

   9.2. the icon can be a null value and it will use the image that was used when the banner was set up

   9.3.when a banner is already in effect then it adds this banner information to the queue

   9.4.The latest banner information will be retrieved from the queue.