

NIGHTPEN, LLC UTILITIES – GAMEOBJECTS

Introduction

Thank you for purchasing the *NightPen, LLC Utilities—GameObjects Package* made for Unity 3D. This package will keep you from having to manually hook up any GameObjects that are defined in your scene to the game's source code. The Package will even work for GameObjects that are inactive in your scene!

Installation

The installation process is very easy. You simply need to import the package into your project. The only file that you need to keep is the *GameObjects.cs* file. This documentation, with example scenes, is for reference and can be safely deleted when you no longer need it. Likewise, you can move the *GameObjects.cs* script to any folder in your project. It is a static class and will auto-initialize once the scene starts running. To see a demo of how to use the *GameObjects.cs* script, look at the *DemoGameObjects* scene.

Using the GameObjects class in your game

Using GameObjects is very simple. The package consists of a single source file called *GameObjects.cs*. *GameObjects.cs* is implemented as a static class. As long as the file is placed into your Unity 3D Assets folder, the methods contained within *GameObjects.cs* will be available for use in your application or game. The GameObjects class is contained within the *NightPen.Utilities* namespace. To use it in your application, you need to import the *NightPen.Utilities* namespace into your source code. This is accomplished by adding the statement:

```
using NightPen.Utilities;
```

This statement needs to be added to each cs file that you want to use the GameObjects class in.

Note: You may also need to add the `using System.Collections.Generic;` statement in order to access the system `LIST<>` type.

You will now have access to the *GameObjects* class. This class provides you with access to 7, now 10 as of version 1.8, statically defined methods. These methods are:

1. `void Initialize()`

2. List<GameObject> GetByName(string name)
3. List<GameObject> GetByTag(string tag)
4. List<GameObject> GetByID(int id)
5. GameObject GetFirstOrDefaultByName(string name)
6. GameObject GetFirstOrDefaultByTag(string tag)
7. GameObject GetFirstOrDefaultByID(int id)
8. GetComponent<T>(string name)
9. Get()
 - a. Now has a feature in 1.8 that allows you to search for GameObject(s) using the full path name.

Now Listed in Version 1.8

10. GameObject.SetActive(string, bool)

Initialize() causes GameObjects to scan your active scene for GameObjects. If you want to load a different scene via **Application.Load("scene")**, you will need to call **GameObjects.Initialize()** in order to scan the new scene for GameObjects. The scan only needs to happen when the scene is first loaded.

GetByName, **GetByTag** and **GetByID** returns a list of GameObjects that match the parameters you specify. For example, to return a list of all GameObjects in the current scene named 'MyWall', you could type:

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using NightPen.Utilities;

public class DemoGameObjects : MonoBehaviour
{
    /// <summary>
    /// Show all GameObjects named MyWall in the current scene.
    /// </summary>
    void Start()
    {
        GameObjects.Initialize();

        List<GameObject> gos = GameObjects.GetByName("MyWall");

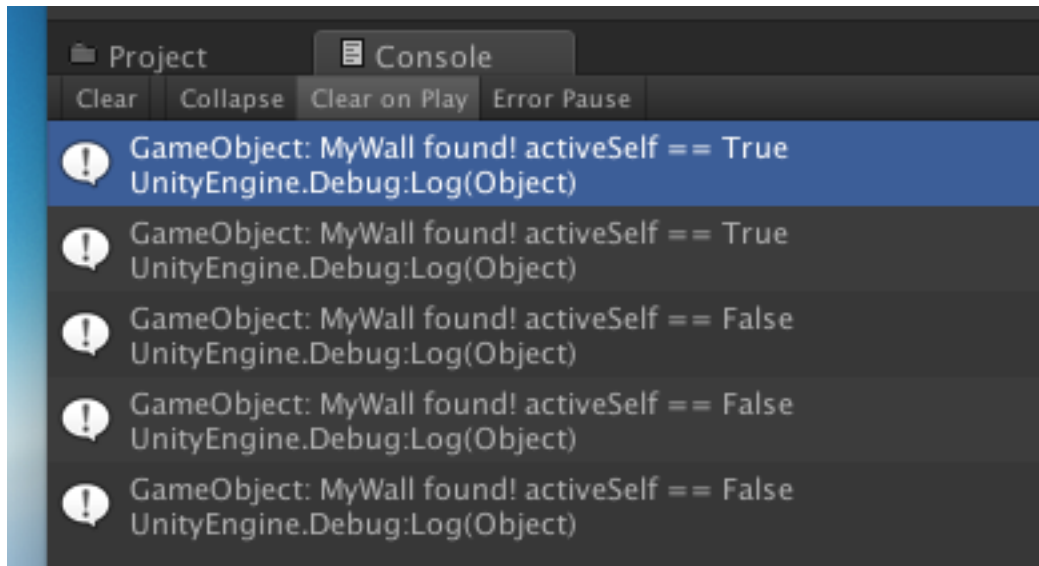
        foreach(GameObject go in gos)
        {
            Debug.Log("GameObject: " + go.name +
```

```

        " found! activeSelf == " + go.activeSelf);
    }
}
}

```

If you attach this script to a GameObject, as has been done in the [DemoGameObjects](#) scene, a list of GameObjects with the name 'MyWall' will be returned when the scene is played. The output from a sample run of this scene is shown below:



Note: Even if the GameObject is inactive, it is still returned.

The methods in the *GameObjects* class will even work for the uGUI elements, both active and inactive. This lets you setup your uGUI elements before you show it to the player. This is illustrated in the [GameObjectsUI](#) scene. The code for that scene is shown here for reference:

```

using UnityEngine.UI;
using System.Collections;
using NightPen.Utilities;

public class DemoChangeUIText : MonoBehaviour
{
    void Start()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();

        //Get the uGUI text control named MyText.
        GameObject go = GameObjects.GetFirstOrDefaultByName("MyText");
    }
}

```

```

        //Change the text
        ((Text)go.GetComponent<Text>()).text = "The Text is Changed!";

        //Show the text to the player.
        go.SetActive(true);
    }
}

```

The **GetByName**, **GetByTag** and **GetByID** functions return a List. The list will never be null, but can have a count of 0 if no GameObjects with the specified name, tag, or instance ID are found.

The **GetFirstOrDefaultByName**, **GetFirstOrDefaultByTag**, and **GetFirstOrDefaultByID** functions will return a single GameObject with the specified name, tag, or ID. If none of these options are found, then these functions return a null GameObject.

In version 1.6 of the “NightPen, LLC Utilities—GameObjects”, two new functions are available. **GetComponent<T>** allows for a specified component, T, to be found from within a specified GameObject. If the component, or GameObject, does not exist, then an empty list will be returned. The other component added is **Get()**. This component is a shorthand form of **GetFirstOrDefaultByName**.

List of Functions and Use

Initialize()

Initializes the GameObjects class by scanning the current scene. This method should normally be called in one of your class’ **Awake()** methods.

void Initialize()

Parameters

None

Return Value

None

Example

```

using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoInitialize : MonoBehaviour {
    void Start ()

```

```

    {
        //This will force the GameObjects to be scanned.

        void Initialize ();
    }
}

```

GetByName

Gets a list of GameObjects in the current scene with the specified name.

List<GameObject> GetByName(string name)

Parameters

String name Name of the GameObjects to be found and returned.

Return Value

Returns a list of GameObjects that match the specified name. If no GameObjects with the specified name exist in the scene, an empty list is returned.

Example

```

using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoGetByName : MonoBehaviour {
    void Start ()
    {
        //This will force the GameObjects to be scanned.

        GameObjects.Initialize ();

        //This will get the uGUI text with the name GameObjectName.
        List<GameObject> gos = GameObjects.GetByName ("GameObjectName");
        Debug.Log ("You have " + gos.Count +
            " GameObjects in the scene with GameObjectName.");
    }
}

```

GetByTag

Gets a list of GameObjects in the current scene with the specified tag.

List<GameObject> GetByTag(string tag)

Parameters

string tag The name of the tag to be found and returned.

Return Value

Returns a list of GameObjects that match the specified tag. If no GameObjects with the specified tag exist in the scene, an empty list is returned.

Example

```
using UnityEngine.UI;
using System.Collections;
using NightPen.Utilities;

public class DemoGetByTag : MonoBehaviour
{
    void Awake()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();
    }

    void Start()
    {
        //Get the uGUI text control named MyText.
        List<GameObject> gos = GameObjects.GetByTag("MyTag");
        Debug.Log("There are" + gos.Count +
            "GameObjects in the scene with the tag MyTag.");
    }
}
```

GetByID

Gets a list of GameObjects in the current scene with the specified ID.

List<GameObject> GetByID(int id)

Parameters

Int id The ID of the GameObjects to be found and returned.

Return Value

Returns a list of GameObjects that match the specified ID. If no GameObjects with the specified ID exist in the scene, an empty list is returned.

Example

```
using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoGetById : MonoBehaviour {
```

```

void Awake()
{
    //Force the GameObjects to be scanned.
    GameObjects.Initialize();
}
void Start () {

    List<GameObject> gos; = GameObject.GetByID (101);
    Debug.Log("There are " + gos.Count + " GameObjects with the matching ID
.");
}
}

```

GetFirstOrDefaultByName

Gets a GameObject in the current scene that matches the specified name.

Note: This function has a shorthand version of this method called [Get\(\)](#).

GameObject GetFirstOrDefaultByName(string name)

Parameters

String name The name of the specific GameObject to be found and returned.

Return Value

Returns the first GameObject with the specified name. If no GameObject with the specified name is found in the scene, a null will be returned.

Example

```

using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoGetFirstOrDefaultByName : MonoBehaviour
{
    void Awake()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();
    }

    void Start () {

        //This will search for the GameObject with the name
        GameObjectName
        GameObject gos = GetFirstOrDefaultByName("GameObjectName")
        Debug.Log("Found the GameObject" + gos +);
    }
}

```

```
}  
}
```

GetFirstOrDefaultByTag

Gets the first GameObject in the current scene that matches the specified tag.

GameObject GetFirstOrDefaultByTag(string tag)

Parameters

String tag The name of the specific tag that will be found and returned.

Return Value

Returns the first GameObject with the specified tag. If no GameObject with the specified tag is found in the scene, a null will be returned.

Example

```
using UnityEngine;  
using System.Collections;  
using NightPen.Utilities;  
using System.Collections.Generic;  
  
public class DemoGetFirstOrDefaultByTag : MonoBehaviour  
{  
    void Awake()  
    {  
        //Force the GameObjects to be scanned.  
        GameObjects.Initialize();  
    }  
  
    void Start () {  
  
        //This will force a search for a specific tag.  
        GameObject gos = GetFirstOrDefaultByTag("MyTag")  
        Debug.Log("Found the GameObject" + gos +);  
    }  
}
```

GetFirstOrDefaultByInstanceID

Gets the first GameObject in the current scene that matches the specified Instance ID.

GameObject GetFirstOrDefaultByInstanceID(int id)

Parameters

Int id This is the specific instance ID that will be found and returned.

Return Value

Returns the first GameObject with the specified Instance ID. If no GameObject with the specified Instance ID is found in the scene, a null will be returned.

Example

```
using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoGetFirstOrDefaultByInstanceID : MonoBehaviour
{
    void Awake()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();
    }

    void Start () {
        //This will force a search for a specific instance ID.
        GameObject gos = GetFirstOrDefaultByInstanceID(int 103)
        Debug.Log("Found the GameObject" + gos +);
    }
}
```

GetComponent<T>(string name)

Gets the component of T type from the specified GameObject. Works even if the GameObject is disabled.

GameObjects.GetComponent<T>(string name)

Parameters

T The type of component to be found and returned.
string name The name of the GameObject to be searched.

Return Value

Returns the component of T type if the GameObject has one attached. If no component of T type is attached to the GameObject, a null will be returned.

Example

```
using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoSetText : MonoBehaviour
```

```

{
    void Awake()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();
    }

    void Start () {

        //Change the text of the uGUI text component attached to the MyText GameO
bject.
        GameObjects.GetComponent<Text>("MyText").text = "The Text is Change
d!";
    }
}

```

GameObjects.Get(string name)

This is a shorthand version of [GetFirstOrDefaultByName](#).

Parameters

String name The name of the GameObject to be found and returned.

Return Value

Returns the first GameObject with the specified name. If no GameObject with the specified name is found in the scene, a null will be returned.

Example

```

using UnityEngine;
using System.Collections;
using NightPen.Utilities;
using System.Collections.Generic;

public class DemoGetFirstOrDefaultByInstanceID : MonoBehaviour
{
    void Awake()
    {
        //Force the GameObjects to be scanned.
        GameObjects.Initialize();
    }

    void Start () {

        //Show the text to the player.
        GameObjects.Get("MyText").SetActive(true);
    }
}

```

Note: As of version 1.8, *Get()* now has an added feature. This feature allows you to get the GameObject based on the full path name of a GameObject, even child objects.

Example

```
using UnityEngine;
using System.Collections;
```

```
using NightPen.Utilities;
```

```
public class ParentChildExample : MonoBehaviour
```

```
{
    void Awake()
    {
        //Force the game objects to be scanned.
        GameObjects.Initialize();
    }

    /// <summary>
    /// This example shows how to get a game object using the full path name
    /// of the game object.
    /// </summary>
    IEnumerator Start()
    {
        for(int row=1; row<4; ++row)
        {
            for(int cube=1; cube<5; ++cube)
            {
                GameObjects.Get("BunchOfCubes/Row "+row.ToString()+"/Cube "+cube.ToString()).SetActive(false);
                yield return new WaitForSeconds(.1f);
            }
        }
    }
}
```

GameObjects.SetActive(string name, bool)

This sets a GameObject to true or false.

Parameters

| | |
|--------------------|--|
| <i>String name</i> | The name of the GameObject to be found and returned. |
| <i>Bool</i> | The Boolean variable to set a GameObject to. |

Return Value

Sets the named GameObject to True or False.

Example

```
using UnityEngine;
using System.Collections;
using NightPen.Utilities;

public class SetActiveExample : MonoBehaviour
{
    void Awake()
    {
        GameObjects.Initialize();
    }

    IEnumerator Start()
    {
        //Set all squares inactive at once.
        GameObjects.SetActive("square", false);
        yield return new WaitForSeconds(1.0f);

        //Set the spheres inactive 1 at a time.
        GameObjects.SetActive("Sphere1", false);
        GameObjects.SetActive("Sphere2", false);
        yield return new WaitForSeconds(1.0f);

        //Set the first sphere active.
        GameObjects.SetActive("Sphere1", true);
        GameObjects.SetActive (

    }
}
```

Performance

GameObjects are made to be very fast. Internally, GameObjects are stored in a dictionary of lists. When your code asks for a list of GameObjects, the search time is almost instantaneous. What needs to be done is to look up the corresponding list of GameObjects and return them to your application.

For example, the screens that are shown below display the results of a running performance test scene. Even when asking for 1,134 GameObjects, every frame produces no discernable difference in performance.

