GetAssetPreview in UnityEngine

This asset provides a quickly way to get any object preview like AssetPreview.GetAssetPreview() Unity Editor method actually does.

> See full PDF at: https://dev.z3nth10n.net/dev/assets/getassetpreview/manuals/README.pdf

Setup

Single example

To setup this asset just create a new object in any class:

```
public class Example : MonoBehaviour
{
    private AssetPreviewer previewer;
}
```

You only need to create a new object on any MonoBehaviour methods call (Awake, Start...):

```
private void Start()
{
    previewer = new AssetPreviewer();
}
```

There are three constructors:

```
public AssetPreviewer() { }

public AssetPreviewer(int w, int h) { }

public AssetPreviewer(int w, int h, float rSpeed) { }
```

- Where 'w' is texture width, by default is 512.
- Where 'h' is texture height, by default is 512.
- Where 'rSpeed' is rotation speed, by default is 10. (This is how quickly object will rotate, check video)

Also, you can use 'AssetPreviewer.Instance' as a singleton.

Then, I recomend to call GetAssetPreview once and store its returning value:

```
// ... previewer from before
private Texture2D texture;

private void Start()
{
    texture = previewer.GetAssetPreviewer();
    // Or...
    texture = AssetPreviewer.Instance.GetAssetPreviewer();
}
```

Then, you can draw the texture OnGUI() like this:

```
private void OnGUI()
{
    GUI.DrawTexture(new Rect(0, 0, 128, 128), texture);
}
```

If you want to rotate on hover, just do this:

```
private void OnGUI()
{
    Rect rect = new Rect(0, 0, 128, 128);
    Event e = Event.current;
        mPos = e.mousePosition;

        if (!rect.Contains(mPos))
        {
            if (texture != null)
                GUI.DrawTexture(rect, texture);
        }
        else
    {
            previewer.OnHoverDisplay(e, rect, mPos);
        // Or...
        AssetPreviewer.Instance.OnHoverDisplay(e, rect, mPos);
    }
}
```

Multiple Example

If you need to know how you can get previews from multiple objects at once I recommend you to see the comments inside the `MultipleExample.cs` file under `UnitedTeamworkAssociated/GetAssetPreview/Scripts/Examples` folder.

If you need to process a `list of GameObjects` (instead of using a static path to a `Resources` folder) it could be easily by creating a `Coroutine` that yields all the `GameObjects` through a foreach loop.

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

// This uses a complete example from above

public class ExampleClass : MonoBehaviour
{
    private AssetPreviewer m_Previewer;
    private PreviewOptions m_Options;
    private List<Texture2D> m_Textures;
    public GameObject[] m_ObjectArray;

    public void Start()
    {
        m_Previewer = new AssetPreviewer();
        m_Textures = nw List<Texture2D>();

        // This is an example of options (read documentation to know more about them)
        m_Options = new PreviewOptions
        {
            DestroyObjectAtFinish = true,
            CloneObject = false
        };
    }

    public IEnumerator GetPreviews()
    {
        foreach(var obj in m_ObjectArray)
        {
            var texture = m_previewer.GetAssetPreview(obj, m_Options);
            yield return new WaitForEndOfFrame();

            m_Textures.Add(texture);
        }
    }

    // Then, you could implement an OnGUI method to display on a list as we did
    // on the 'MultipleExample.cs' OnGUI method.
}
```

Main idea && advanced implementation

The idea of this is very simple, but keep in mind that all the previews are obtained through coroutines.

Intercommunitation between scripts is essential, due to this fact, we implemented several callback methods where you can retrieve the progress while the Textures are being obtained (`PreviewUtility.CreatedTexture`), or when the process finishes (`PreviewUtility.Finished`).

For documentation, just check the Documentation API.

TODO

- By the moment, I recommend you to execute it on IMGUI or moveless scenes due to small lag spikes (this will be fixed on the next update, thanks to 'UnityEngine.Rendering' utils (available in 2018.3)).
- All the batched previews can be only processed by using the Resources folder (this feature will be changed on next updates, allowing GameObjects from outside)

Issues

Having issues? Just report in the issue section. **Thanks for the feedback!**

VERY IMPORTANT DO NOT POST ENTIRE FILES OF THIS PROJECT, JUST POST THE CODE THAT YOU MODIFIED OR STACKTRACES!

And that's all!

Thanks for buying this asset.

Best regards from United Teamwork Association.