proScreenShot Documentation

Table of Contents

1. How to start
2. Component settings
3. Scripting API

1. How to start

To start using proScreenShot just create empty GameObject and attach proScreenShot Component to it.

Default settings should work for most cases but if you need more specific settings check nr 2 and 3.

You can make screenshot by clicking “MAKE SCREENSHOT” in the Inspector Component. If you want to make screenshot at Runtime you can create new C# script and attach it to the same GameObject and use this simple function to get start with:

```
private void OnGUI()
{
   if(GUI.Button(new Rect(10,10,200,50),”Make Screenshot”))
   {
      this.gameObject.GetComponent<proScreenShot>().MakeScreenShot();
   }
}
```

And that’s it. It will make screenshot with the settings that are in the inspector of proScreenShot Component.

I Suggest looking at these example scenes:

- SimpleScreenshot_example01
- Album_example02

2. Explaining settings

Directory - by default (if empty) it will save screenshot appropriately depending on the platform (eg. if iOS it will save to iOS app Documents, if Standalone it will save to StreamingAssets). It will also additionally save screenshot in the Assets/MyScreenShots/ in the editor to preview screenshot (only if using editor).

File Primary Name - The file base name.

File Extension - file extension name. You can give any extension name for the file. Note: it will not change the format of the file. If you want to change format, you need to use Encode Format field.

Add Date To File Name - if enabled it will add timestamp on the file name (eg. fileprimaryname_09-07-2015_15:31:33.jpg).

Add Random Number To File Name - if enabled it will add a random number to the file name (just for randomization), (eg. ScreenShot_[758].jpg).

Use Screen Resolution - if enabled, screenshot size will always be equal to the screen resolution.

Screenshot Sound - you can add a sound that will be played when calling MakeScreenShot() function.

Shot Resolution - you can set custom resolution for the screenshot (make sure you disable "use screen resolution" when setting custom resolution.

Blend Textures - you can blend custom textures to each screenshot (eg. Logo, watermarks, frames, etc).

Blend Textures Position - set the position for each blend texture. Make sure that Blend Texture Position count is equal to Blend Texture count.

Ignore Object List - when making screenshot you can disable some of the GameObject that you don't to be rendered when making a screenshot.

Screenshot Format - Format for the image file (ARGB32, RGB24, RGB16).

Encode Format - Format for the image file (JPG, PNG).

Blend Type - When blending textures you can choose either a Simple (it will just put texture on top) or Transparent (it will blend textures together (good when blend texture has alpha)).

MAKE SCREENSHOT - you can make a screenshot in the Editor (offline mode).

See **BlendLogoAndIgnore_example03** for more information.

3. Scripting API

proScreenShot is fully customizable at Runtime. You can use this API for making custom settings at Runtime:

Firstly you get the Component, eg:

```
proScreenShot _proScreenshot = this.gameObject.GetComponent<proScreenShot>();
```

API:

```
// Set Custom Name
_proScreenshot.SetName("CustomName");

// Force custom path for screenshots
_proScreenshot.ForceFullCustomPath(“custom_path”);

// Set Extension
_proScreenshot.SetExtension(".png");

// Load Texture From Resources and Blend it with screenshot
_proScreenshot.AddBlendTexture(new
Vector2(0,-250),(Texture2D)Resources.Load("watermark"));

// Ignore Red Cube in the scene
_proScreenshot.AddIgnoreObject(GameObject.Find("RedCube"));

// Set Custom resolution for screenshot and disable to use screen resolution
_proScreenshot._useScreenResolution = false;
_proScreenshot.SetShotResolution(800,600);

// Set blend format
_proScreenshot.SetBlendFormat("Transparent");

// Set Encode format
_proScreenshot.SetEncodeFormat("JPG");

// Set Texture format
_proScreenshot.SetTextureFormat("ARGB32");

// Set Sound
_proScreenshot._screenshotSound = (AudioClip) Resources.Load("sound");
```

```
// Add date to screenshot name
_proScreenshot._addDateFileName = true;


// Add Random Number to screenshot name
_proScreenshot._addRandomNumberToName = true;

// Clean Blend textures if you want to change them between screenshots
_proScreenshot.CleanBlendTextures();

// Clean Ignore Object List if you want to change them between screenshots
_proScreenshot.CleanIgnoreObjectList();

// proScreenShot will auto save all the screenshot list, but if you want to clear use this
_proScreenshot.ClearSavedScreenShotList();

// Get all the paths of made screenshots
string[] list = _proScreenshot.GetPathListOfSavedScreenShots();

// Get texture from the given path
Texture2D scr = _proScreenshot.GetTextureFromPath(“path”);

// Get path of the last made screenshot
string path = _proScreenshot.GetPathOfLastScreenShot();

// Play Screenshot sound manually
_proScreenshot.PlaySound();

// Delete screenshot at path
_proScreenshot.DeleteScreenShot(string path);

// Delete last screenshot
_proScreenshot.DeleteLastScreenShot();

// Delete all screenshots (that are saved)
_proScreenshot.DeleteAllScreenShots();

// Make Screenshot
_proScreenshot.MakeScreenShot();
```

See **RuntimeAPI_example04** example.

Fore more information contact:

pro.andrius@gmail.com
http://prounitytools.com