ASCII Camera

User Manual

Version 1.0.0

Contents

INTRODUCTION

ASCII Camera is a **Android and Windows** plugin that let you transform an input image or camera video frame into printable ASCII characters at real-time using a single decision tree. Real-time performance is achieved by using **pixel intensity comparison** inside internal nodes of the tree.

Possibilities of current version:

- Supported armeabi-v7a, x86, arm64-v8a-bit Android processors and x86, x86_64 bit Windows processors;
- Supported possibility to render image or camera video frame in ASCII characters;
- Supported possibility to render multiple effect instances;
- Supported in Unity Editor mode.

INSTALLATION GUIDELINE

Import the ASCIICamera package from the Asset Store. You should now have a folder with this name "ASCIICamera" in your project with the following structure:

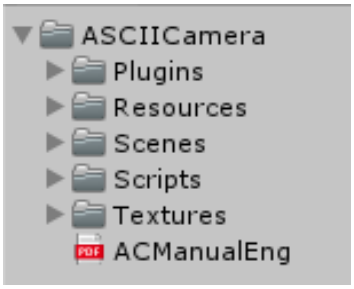

Figure 2.1: Structure of ASCIICamera asset

- **Plugins**: All the native dlls and shared libraries (in this folder you can directly enable/disable of using libraries for each Android/Windows processors);
- **Resources**: Include asset settings file;
- **Scenes**: Example scenes that's show all components in a ready setup and how work with some additional features of ASCIICamera asset;
- **Scripts**: C# classes that show how to work with ASCIICamera;
- **Textures**: Additional images that used in demo scenes.

USAGE GUIDELINE

You need at first choose one of the demo scene that available in assets that give you possibility to manage all possible functionality, so you can move it into your Unity scene. The main scenes is "CameraToAsciiGlyphsScene" that give you example how to use asset to convert web camera frames into ASCII characters texture.

When you select the "ARECameraHelper" component in your hierarchy window you can manage it with the component inspector:

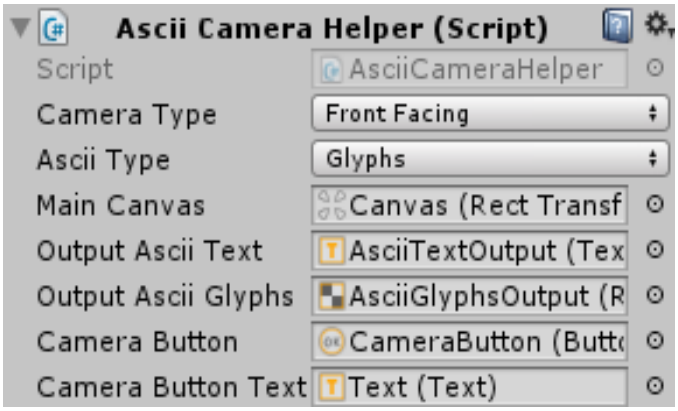

Figure 3.1: ARECameraHelper inspector view

- **Camera Type**: Choose the camera that will be used by default, like front or back camera if one of them is supported on device where application will be running. If chooses camera is not supported will be try to use default one for current device;
- **Ascii Type**: Choose what output data we what to get:
    - **Glyphs** - convert input data into ASCII texture;
    - **Text** - convert input data into ASCII text;
- **Main Canvas**: Default Unity "Canvas" component, that will be used for size calculation;
- **Output Ascii Text**: Default Unity "Text" component that will be used for ASCII text output;

- **Output Ascii Glyphs**: Default Unity "RawImage" component that will be used for ASCII texture output;
- **Camera Button(Text)**: Default Unity "Button" component that will be used for start converting process into ASCII data.

CLASSES/METHODS DESCRIPTION

Main Classes

- `AsciiEffect` – main class that has all ASCII converting functions;
- `AsciiBuffers` – classes that uses for store all buffers information.

Main Interfaces

- `IAsciiEffect`

```
{
    byte[] ConverToAsciiText(Color[] pixels, int width, int height);
    Color[] ConverToAsciiGlyphs(Color[] pixels, int width, int height);
    void Release();
}
```

AsciiEffect

Constructors

```
/// <summary>
/// Create new instance of ASCIICamera object for current supported platform
/// </summary>
public Asciiffect()
```

Properties

```
/// <summary>
/// Get web view object for current running platform
/// for get more additional possibilities that exists only for this platform.
/// </summary>
public object PlatformWebView
```

```
/// <summary>
/// Convert pixels array into ASCII text data
/// </summary>
/// <param name="pixels">Raw pixels of the input image.</param>
/// <param name="width">Width of the input image to be processed.</param>
/// <param name="height">Height of the input image to be processed.</param>
/// <returns></returns>
public byte[] ConverToAsciiText(Color[] pixels, int width, int height)
```

```
/// <summary>
/// Convert pixels array into ASCII texture data
/// </summary>
/// <param name="pixels">Raw pixels of the input image.</param>
/// <param name="width">Width of the input image to be processed.</param>
/// <param name="height">Height of the input image to be processed.</param>
/// <returns></returns>
public Color[] ConverToAsciiGlyphs(Color[] pixels, int width, int height)
```

```
/// <summary>
/// Release current instanse of ASCIICamera
/// </summary>
public void Release()
```