EASY
FONT

1. Changelog

v1.0 - Initial version

2. What is this

First of all: **Thanks for buying this package!**

This package is a really easy to use custom dynamic font renderer. It does the same as the Unity 3DTextMesh and **more:**

1. Shadows: Add projected text shadows
2. Outline: You can add outline if you wish
3. Gradients: Add color gradients to your text
4. Detail texture: Add detail to the text, like zebra texture!
5. Easily tweak text resolution without moving the text

As any Unity dynamic fonts it will create the font texture “on the fly”, so you just have to include your desired ttf, dfont, otf an you are ready. Even if you don’t include a font in the project the dynamic fonts will work with the system installed ones. This is PERFECT for localization (forget to create a custom bitmap font for those huge japanese characters :))

A **HUGE** amount of effort have been made to make this package feel exactly the same as a unity component (auto refresh, prefab highlight...). If you have ever made a custom inspector you already know how difficult is to make this right.

3. What’s inside

In this package you will find:

1. "EastFontTextMesh.cs" : This is the script for the text
2. “EasyFontCustomEditor.cs” : The custom inspector that makes everything feel like a Unity component
3. “EasyFontDetail.shader”: A font detail texture shader
4. A sample scene

4. How it works

Two simple three steps for making a text:

1. Add a Empty GameObject
2. Add the “CustomTextMesh.cs” to it
3. Tweak!

5. Params

Here there is an explanation for all the configurable font params:

1. **Don’t override Materials**: By default, “Easy Font” auto changes the font renderer materials as needed. Normally you want this but for those special cases where you need special custom materials enable this option.

WARNING: If you enable "**Don't override materials**" you will have to feed the correct materials with the correct font textures

2. **Text:** The text to show. Unity alt+enter works for line break and also \n
3. **Font:** The font type
4. **Custom fill material:** If you want to have detail textures on the text assign here a material with the TextShaderDetail.shader
5. **Font size:** The font resolution. This will generate the font texture based on this value. Higher values will generate bigger textures. You have a helper text in the lower part of the inspector with the current texture size
6. **Character size:** The proportional size of each character quad. Use this to change the font size instead of the transform, although you can use the transform if you wish
7. **Text anchor:** The pivot of the whole text
8. **Text alignment:** The alignment of the text. Only works when there are more than one lines
9. **Line spacing:** The space between paragraph lines
10. **Top color:** The color for the upper part of the characters
11. **Bottom color:** The color for the bottom part of the characters
12. **Enable shadow:** Enable or disable the font shadow. This increases the vertex count by two
13. **Shadow color:** The color of the shadow
14. **Shadow distance:** The shadow offset form the main text. Normally you don't want to change the z value
15. **Enable Outline:** Enable or disable the font outline. This multiplies the vertex count of the text by 4
16. **Outline color**: Sets the outline color
17. **Outline with:** Sets the outline width. Too high values will look weird.

6. FAQs

Why the text goes a bit up and down when writing?
Because we center the text vertically based on the previous characters height. Thats why the pivot is centered to the text.

How EASY FONT works with japanese, chinese....
It just works, like the Unity 3D text does.

Some character doesn't show correctly. What I have to do?
Add a Unity 3D Text mesh to the scene and configure it with the same text and same font type as the EASY FONT. If the character doesn't show correctly is Unity's fault ;) ... Contact me.

Does this works on mobile?
Sure, I'm an mobile developer so everything works 100% on mobile.

How the detail texture works?
A new set of uv are created on the channel 2. This is used for the detail texture.

Does the text mesh recreate when animating colors?
You can animate colors in two ways. Changing the font color material or the top, bottom colors. Neither of both recreates the mesh. Also there is a helper method that changes the Top and bottom color at the same time called SetColor(_color)

Sometimes when I open a scene the text is screwed up!
Don't worry you haven't lost any work. Just select a text and press "Refresh all" or hit play. This is not even necessary, when playing the text mesh are recalculated