VText
Virtence GmbH
Altenburger Str. 13
04275 Leipzig

VText
Unity Package

info@virtence.com

April 29, 2015

1 Quickstart

VText is Unity package to generate dynamic 3D–Text geometry directly from truetype fonts(`.ttf` or `.otf`). Main features are

- Different Materials for
  - Frontface
  - Bevel
  - Sides
- Kerning
- Texture layout suitable for Lightmapping
- Available build targets
  - Android (Arm 32–bit)
  - Linux (32/64–bit)
  - Windows (32/64–bit)
  - OSX (32/64–bit)
  - iOS (Arm 32/64–bit)
- Layout along AnimationCurve

1.1 Install

After import of the VText package, select the menu `Windows/Virtence/Setup VText...`
This will show up a Dialog like figure 1.1. After a click on the `Check`-button, the setup will create all necessary toplevel folders, depending on your actual Unity version.

Figure 1.1: Project

After the setup quit Unity and load your project again!

1.1.1 Unity 4

You are able to (un–)install your required plugins via toggle checkmarks like shown in figure 1.2

Figure 1.2: Project

Required Folders

- `Plugins`
- `StreamingAssets`
- `Virtence`

1.1.2 Unity 5

Since the plugin managment changed in Unity 5, the plugins stay in the `Virtence/VText/Plugins` folder. Setup `Check` does some adjustments to the plugins Platform Settings.

Figure 1.3: Project

The inspector should show the plugin settings like figure1.4, 1.5 and 1.6

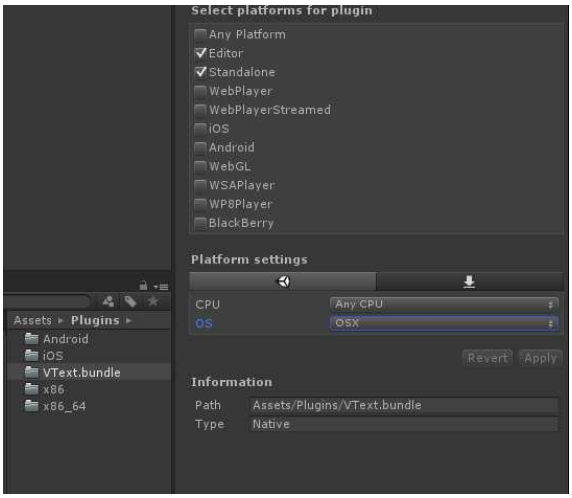

Figure 1.4: Unity5 Platform settings

Upgrading existant project

If you want to upgrade an existant VText project on Unity 5, you might get duplicate plugin Errors. Just delete the whole `Assets/Plugins` folder. On Unity 5 the plugins reside now in `Assets/Virtence/VText/Plugins`.

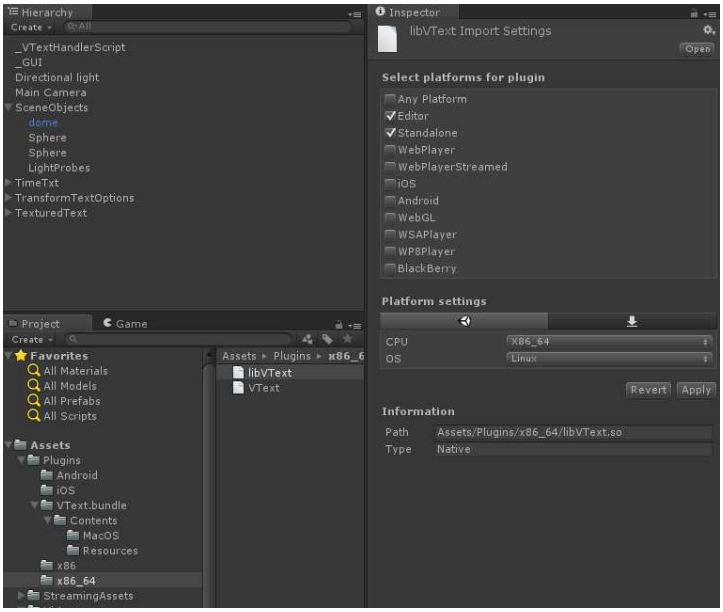

Figure 1.5: Unity5 Platform settings

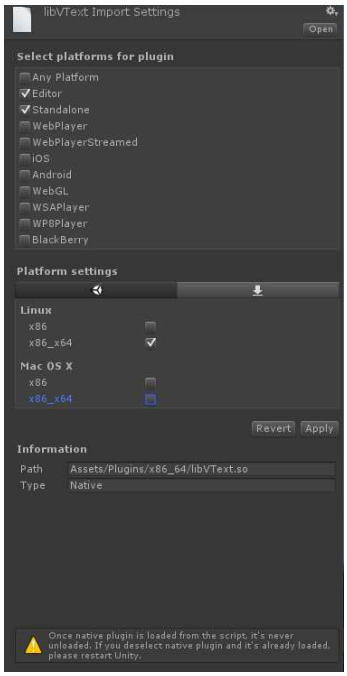

Figure 1.6: Unity5 Platform settings

1.2 Demo scenes

Important! Before opening the test scenes, ensure you setup the folders and plug-ins like in section 1.1. Also install all Fonts from `Assets/Virtence/VText/Fonts` to `Assets/StreamingAssets/Fonts`!

The VText package ships with the two demo scenes: `StartScene` (section 1.2.1) and `TextureLayoutScene` (section 1.2.2).

You'll find them in the folder `Assets/Virtence/VText/_DemoScene`

1.2.1 StartScene

StartScene is a simple animated VText scene with a graphical user interface. In play mode Unity will show something like figure 1.7. It's intention is to show some basic starting points for you to integrate VText into your projects.

Figure 1.7: StartScene

1.2.2 TextureLayoutScene

Figure 1.8 shows the Unity views for TextureLayoutScene. It demonstrates the generated UV–Layout for the `DroidSans.ttf` font.

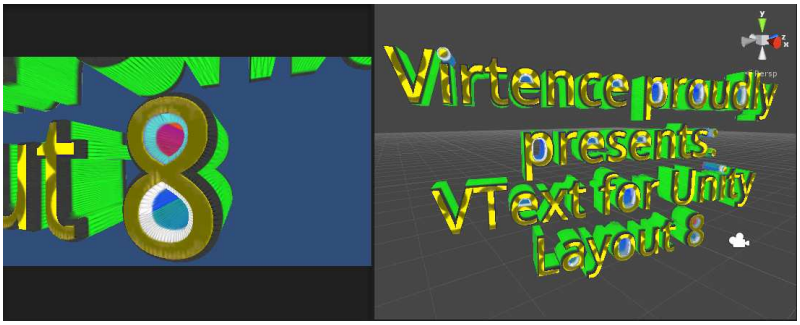

Figure 1.8: TextureLayoutScene

See section 2.2 for detailed information about texture layout.

1.2.3 Creating VText

Clicking on the `components/Virtence/VText` menu item will add an empty child to your hierarchy, which contains the VTextInterface component.

The Inspector editor pane shown in Figure 1.9 will allow you to adjust the settings.

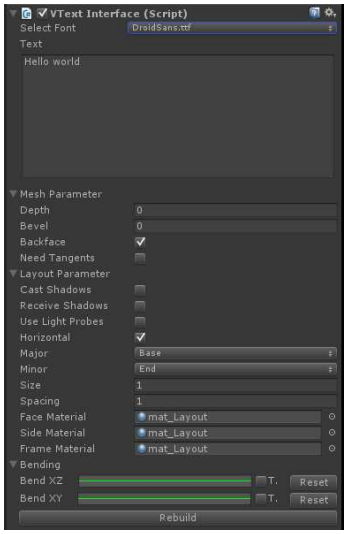

Figure 1.9: Editor

Table 2.1 and 2.2 will explain the parameters in detail.

First, you should select a font for geometry creation. If the FontSelector shows only `(none)`, you may need to copy your fonts to the `StreamingAssets/Fonts` folder.

Note: All Fonts in that folder will be distributed when building for mobile devices! So keep only fonts which are really required in your project to reduce the memory footprint. Also be aware of copyrights!

2 Details

2.1 Parameters

2.1.1 Mesh parameter

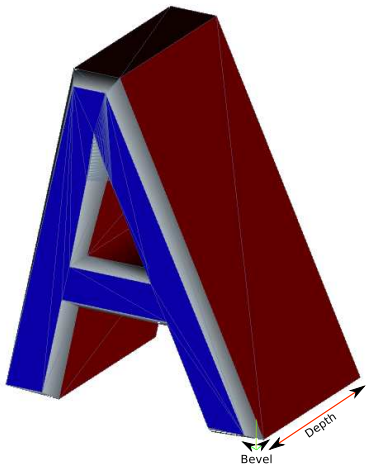

Figure 2.1: Glyph

Name	Description
Select Font	from `StreamingAssets/Fonts` folder.
Text	The actual text to generate
Depth	The relative depth of extruded glyph
Bevel	Outline width and depth–shift
Backface	VText will generate back side if enabled.
Need Tangents	VText will generate tangents on each glyph if enabled. Use only if really needed.

Table 2.1: Mesh parameter

The Mesh parameter are the Editor section of `VTextInterface.parameter`. See Section 2.3.2 for accessing them by Script.

2.1.2 Layout parameter

Name	Description
Cast Shadows	Generated glyphs will cast shadow
Receive Shadows	Generated glyphs will receive shadow
Use Light Probes	Text will use Light Probes if available
Horizontal	Major layout direction. Vertical layout if disabled.
Major	Major layout mode.
Minor	Minor layout mode.
Size	Local scaling factor for each glyph.
Spacing	Local scaling factor for **Minor** adjustment.
Face Material	Material for glyph's face.
Side Material	Material for glyph's extruded side.
Frame Material	Material for glyph's bevel frame.
Bend XZ	Adjust Z–Position along AnimationCurve(see 2.1.3).
Bend XY	Adjust Y–Position along AnimationCurve(see 2.1.3).

Table 2.2: Layout parameter

The Layout parameter are the Editor section of `VTextInterface.layout`. See Section 2.3.3 for accessing them by Script.

2.1.3 Bending

The bending editor contains three elements[1]:

Curve Editor

A click on the curve image will open Unity's Curve Editor. Insert keys here and adjust the bending to fit your needs.

Tangent Toggle

Turn the switch on if you want to align the glyphs along the tangents of your curve.

Reset

Pressing the Reset button will reset the AnimationCurve to linear zero.

[1] from left to right

2.2 Textures

2.2.1 Layout

The UV–Coordinates of each glyph are constructed in a light mapping conformal way.

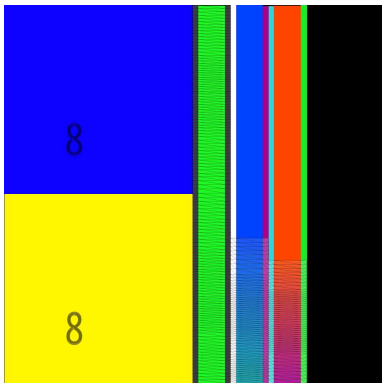

Figure 2.2: Texture Layout of DroidSans glyph 8

As shown in Figure 2.2, the front face consumes the yellow UV–range $(0, 0) \ldots (0.5, 0.5)$. The blue UV–range $(0, 0.5) \ldots (0.5, 1.0)$ covers the backface area. Each contour side covers an U–range of $(0.5, 0.5 + n * 0.1)$ where n is the index of active contour. Bevel range is dependant on its amount, and whether backface rendering is enabled. The V–range is normalized from the length of the longest contour. Figure 2.3 shows the resulting unlit material in Unity.

Figure 2.3: Textured View

You may wonder why front- and back–face layout covers such a small area in the UV–range. It's because the maximum glyph size of `DroidSans.ttf` is that much larger than the common used glyphs. Other fonts may show different ranges.

2.3 Programming interface

You are able to change several parameter of your VText component at runtime via the `VTextInterface` [2] class.

2.3.1 VTextInterface

Property	Type	Description
RenderText	string	The Text to generate.
parameter	VTextParameter	Mesh construction parameter.
layout	VTextLayout	Text layout parameter.

Table 2.3: VTextInterface

First of all you may want to access the generated text via the `VTextInterface.RenderText` property.

2.3.2 VTextParameter

`VTextInterface.parameter` property is a VTextParameter–class.

Property	Type	Description
Depth	float	The relative depth of extruded glyph
Bevel	float	Outline width and depth–shift
Backface	bool	Will generate back side if `true`
GenerateTangents	bool	Will generate tangents on each glyph if `true`
Fontname	string	The font file name

Table 2.4: Mesh properties

Warning: Writing to any of those properties will rebuild glyphs. Expect performance dropdown if you animate one of them!

[2] C# Script located in `assets/Virtence/VText`

2.3.3 VTextLayout

`VTextInterface.layout` property is a VTextLayout–class.

Property	Type	Description
Horizontal	bool	Major layout direction. Vertical layout if `false`.
Major	align	Major layout mode (see Table2.6).
Minor	align	Minor layout mode (see Table2.6).
Size	float	Local scaling factor for each glyph.
Spacing	float	Local scaling factor for minor adjustment.
CurveXZ	AnimationCurve	Z–Position AnimationCurve.
OrientationXZ	bool	Text faces tangent in Z if `true`.
CurveXY	AnimationCurve	Y–Position AnimationCurve.
OrientationXY	bool	Text faces tangent in Y if `true`.
CastShadows	bool	Generated glyphs will cast shadow.
ReceiveShadows	bool	Generated glyphs will receive shadow.
UseLightProbes	bool	Text will use Light Probes if available.

Table 2.5: Layout properties

VTextLayout.align

`VTextLayout.align` is an enumerator for major and minor alignment. The Following values are available:

Name	Description
Base	Pivot is baseline of font
Start	Pivot is Top–Left of entire text
Center	Pivot is center of entire text
End	Pivot is Bottom–Right of entire text

Table 2.6: enum align

Base is baseline only for horizontal Major. The same as Start in any other case.

2.4 Performance

As long as you are using only 2D–Text[3] all VText objects referencing the same font use only one dictionary for glyph lookup. So Unity is able to batch most of your text, even if some use different sizes.

3D–Text requires a separate glyph dictionary for each VText object. So batching is only possible in VText itself.

It is generally no good idea to change parameters at runtime, which require a complete rebuild of the glyphs[4]! Especially text with a lot of different glyphs will need some time to rebuild. A change of layout parameter is not that costly, because no rebuild of glyphs is required.

Depending on the settings of **Need Tangents**, extra attributes are generated. This is only required if one of your materials contains a bumpmapping shader. For best performance turn it off if not required.

If the parameter **Backface** is turned on, additional faces are generated for each glyph. If you'll never see the backfaces in your scene, turn it off.

2.5 Build

If you change or set the name of the font via script, you should be aware of the **case sensitivity** of filenames on non Windows based operating systems! If your text will not show up in your build, first check the right spelling(including case) of the fontname.

[3] depth and bevel zero
[4] such as changing Depth or Bevel

2.6 Legal

2.6.1 3rd party

This package uses the following 3rd party products:

freetype2

```
Portions of this software are copyright © 2014 The FreeType
Project (www.freetype.org).  All rights reserved.
```

Fonts

This package contains the following fonts, which are licensed under the SIL Open Font License:

- CostaRica.otf
- Dotrice-Bold-Condensed.otf
- Lack.otf
- Lobster.otf
- Segment14.otf
- Xolonium-Bold.otf
- RacingSansOne-Regular.ttf