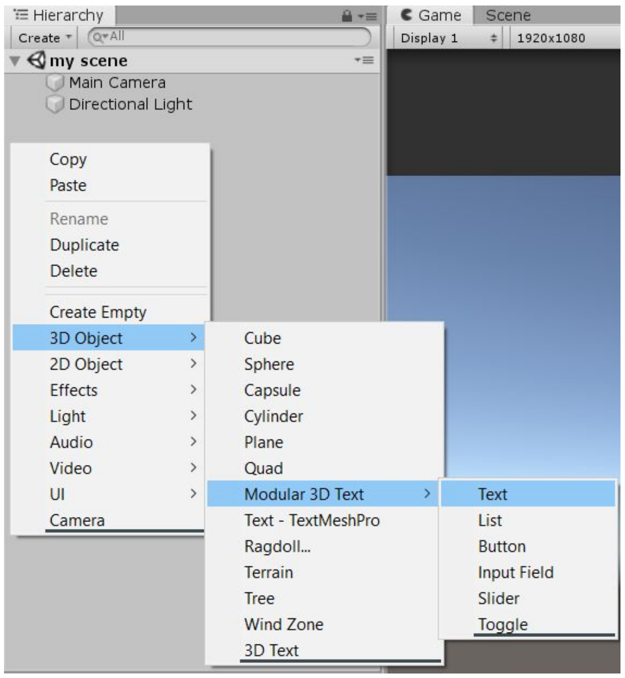
Modular 3D text

<u>Documentation</u>

(new documentation link with script references)

<u>How to use</u>:

Right click in scene hierarchy,
    3D Objects > Modular 3D Text > Anything



Important Note:

Remember to save mesh as an asset on prefabs if you want the mesh to persist between instances of that prefab and not just the settings. The option is moved to advance settings if it isn’t a prefab.

Performance Suggestions:

- ➔ Simply call *UpdateText(“your string”);* to update text. Auto create in *playmode* checks for changes in *Update()*. It is better for performance to avoid unnecessary stuff in *Update()*.
- ➔ Turn on pooling if a lot of text is changed in play mode. This option is under advanced settings.
- ➔ For best performance on texts that won’t be changed, combine the mesh, click “Optimize mesh” button found in advanced settings and remove the modular3DText script.

Asset Cleanup:

This paragraph is aimed at people who like as few files in their project as possible.

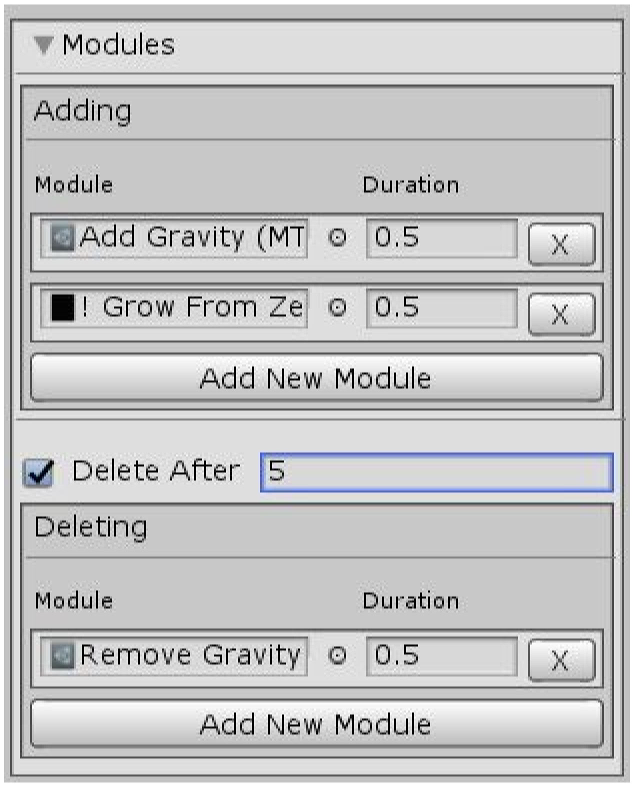
Some file structures might seem strange at first glance but they are that way just for easy cleanup purposes.

Example usage like the Typewriter effect, countdown is separated to the Examples folder. This includes the scripts, prefabs etc. So, if you don’t need them, you can safely delete the Examples folder without anything else breaking.

You can delete the entire Sample Scene folder. This includes Tutorial Animations and the limited controllers(VR cam) for sample scenes that you won’t need anywhere else.

The script naming convention(MText_ScriptName) is to clearly separate asset’s scripts to your owns at first glance. I personally felt a bit annoyed when I searched for something like “Character” script in a project and got 7 results from assets downloaded from Asset Store.

Modules/Effects:


Modules

Adding

Module	Duration
Add Gravity (MT	0.5
! Grow From Ze	0.5

Add New Module

✓ Delete After 5

Deleting

Module	Duration
Remove Gravity	0.5

Add New Module

Press the **Add New Module** button to add modules and **'x'** button besides a module to delete it.

- ➔ Effects will only apply in play mode & if the text object is active/enabled.
- ➔ Combined mesh can't have effects. If you have a combined mesh in editor, in play mode after the first call to create text it will recreate the old texts as separate objects along with effect. Select don't combine in the editor to avoid this.
- ➔ When pooling is turned on If you use "Add Gravity"/"Add rigidbody" in typing effects, use "Remove Gravity"/"Remove rigidbody" on deleting effects. This makes sure objects don't have pre-added components when reusing them through the pool. When using custom modules that add components, use a module to remove that component to be suitable with pooling.
- ➔ There are more settings in the Modules itself. Check them out. Example: You can add a bouncy physics material in the rigidbody module, a curve to choose how the scale change will happen.

<u>Module Creation</u>:

Modules are scriptable objects. Create a module script like the example, create an item for that module and assign the module in text to use.

You can create multiple modules with different parameters for a single script.

Modules can't be run from disabled text object

<u>Example module script:</u>

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

namespace MText
{
      [CreateAssetMenu(menuName = "Modular 3d Text/Modules/Your effect name goes here")]
      public class YourFileNameGoesHere : MText_Effect
      {
          public override IEnumerator ModuleRoutine(GameObject obj, float duration)
          {
            //enter your code here

            yield return null;
          }
      }
}
```

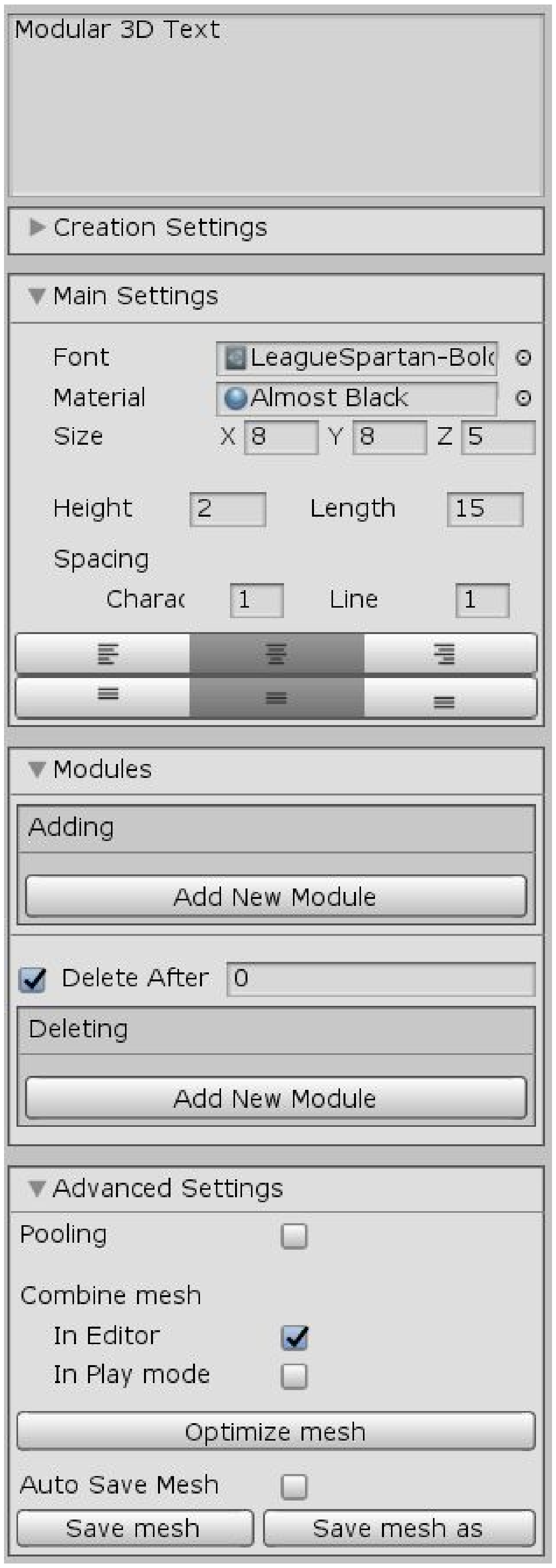
Here, *obj* is the Gameobject of the character being typed and duration is passed from the individual Text.

After creating the script, right click in the project window, click "Create> Modular 3d Text/Modules/Your effect name goes here".

3D UI System:

One object with **MText_UI_RaycastSelector** script attached needs to be in the scene to select items with mouse/touch. You can disable it when not using a 3D UI. This is automatically added when you create a 3d UI item (except plain text).

Text - (Modular3DText.cs)



Modular 3D Text

▶ Creation Settings

▼ Main Settings

Font	LeagueSpartan-Bolc
Material	Almost Black
Size	X 8 Y 8 Z 5
Height	2
Length	15

Spacing

Charac	1
Line	1

▼ Modules

Adding

Add New Module

[✓] Delete After 0

Deleting

Add New Module

▼ Advanced Settings

Pooling [ ]

Combine mesh

In Editor [✓]

In Play mode [ ]

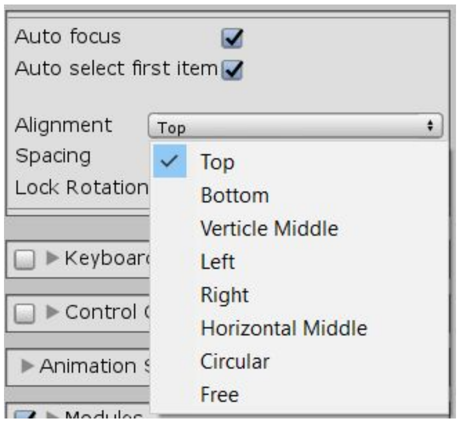
Optimize mesh

Auto Save Mesh [ ]

Save mesh | Save mesh as

List - (MText_UI_List.cs):

- ➔ The helper component updates the list automatically in Editor. In playmode/build you need use **UpdateList**() method to do it manually. This is to improve performance.



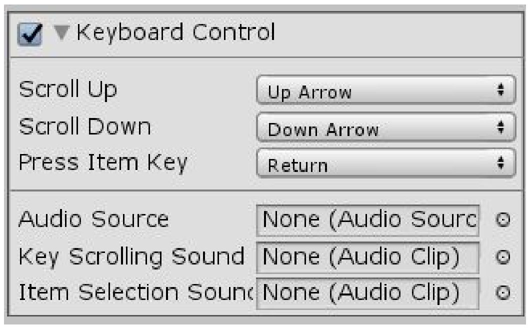
Auto focus ✓
Auto select first item ✓
Alignment: Top

Dropdown options:
✓ Top
Bottom
Verticle Middle
Left
Right
Horizontal Middle
Circular
Free

Spacing
Lock Rotation

- ➔ All 8 types of lists are included in one component. Just pick one and you are ready to go.

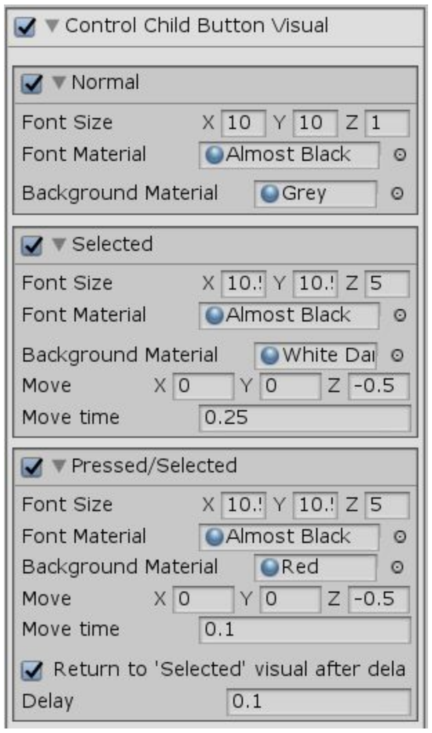
- ➔ Lock rotation freezes list item's rotations to zero.



✓ Keyboard Control

Scroll Up	Up Arrow
Scroll Down	Down Arrow
Press Item Key	Return
Audio Source	None (Audio Sourc
Key Scrolling Sound	None (Audio Clip)
Item Selection Sounc	None (Audio Clip)

- ➔ Don't forget to turn on the keyboard control.



Control Child Button Visual

Normal

Font Size	X	10	Y	10	Z	1
Font Material	Almost Black					
Background Material	Grey					

Selected

Font Size	X	10.!	Y	10.!	Z	5
Font Material	Almost Black					
Background Material	White Dai					
Move	X	0	Y	0	Z	-0.5
Move time	0.25					

Pressed/Selected

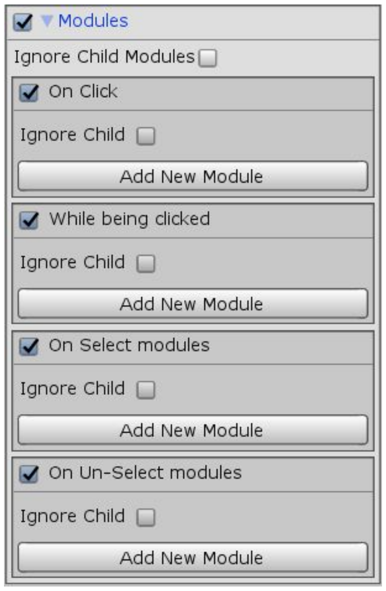
Font Size	X	10.!	Y	10.!	Z	5
Font Material	Almost Black					
Background Material	Red					
Move	X	0	Y	0	Z	-0.5
Move time	0.1					

✓ Return to 'Selected' visual after dela

Delay 0.1

➔ Visual Settings settings only apply to 3D Buttons(**MText_UI_Button**)

➔ You can turn on/off each individual component



Modules

Ignore Child Modules ☐

✓ On Click

Ignore Child ☐

✓ While being clicked

Ignore Child ☐

✓ On Select modules

Ignore Child ☐

✓ On Un-Select modules

Ignore Child ☐

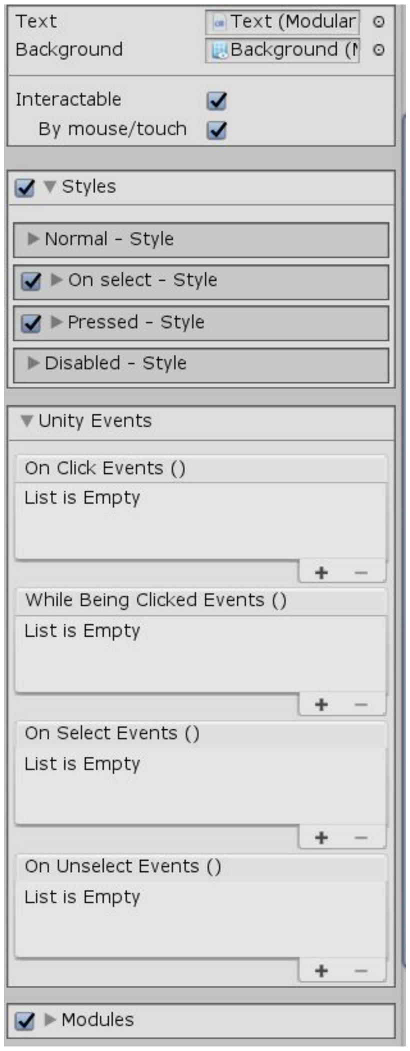
➔ You can turn on/off each individual component

➔ Ignore child toggle turns on/off child button modules to be used or not

➔ While being clicked - calls the module over and over again while mouse is clicked

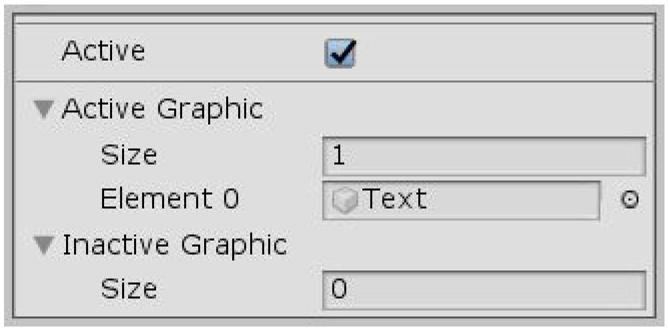
Button - (MText_UI_List.cs):

- While Being Clicked Events/Modules continuously calls those events while mouse click is held down on it. This is called in Update(), which is every frame.
- **onSelectEvent** is mouse hover or selected in a list. **onUnselectEvent** is the opposite
- To disable button via code, set **interactable = false;** and call method **DisabledButtonVisualUpdate();**
- Remember, the size of the Box-collider on the component itself determines if you are hovering/clicking on it, not the background size.


Text: Text (Modular
Background: Background (
Interactable ✓
By mouse/touch ✓
✓ Styles
Normal - Style
✓ On select - Style
✓ Pressed - Style
Disabled - Style
Unity Events
On Click Events ()
List is Empty
While Being Clicked Events ()
List is Empty
On Select Events ()
List is Empty
On Unselect Events ()
List is Empty
✓ Modules

Toggle (MText_UI_Toggle.cs):

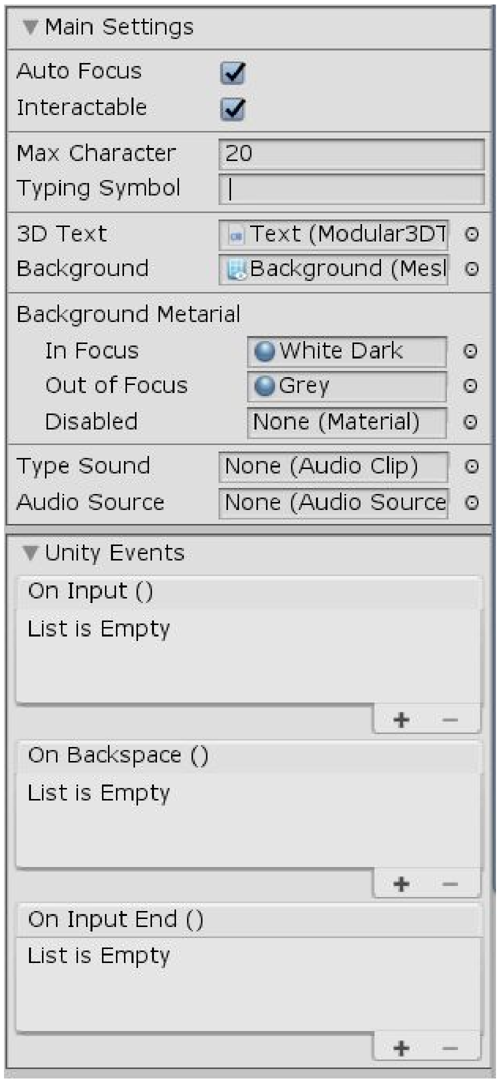
- ➔ Toggle relies on Button to call **Toggle()** which is automatically added when creating toggle
- ➔ Use the **Set(bool)** method to directly set the toggle instead of switching using **Toggle()**



Active ✓
Active Graphic
Size 1
Element 0 Text
Inactive Graphic
Size 0

<u>3D Input Field (Mtext_UI_InputString.cs)</u>:

- ➔ Auto focus focuses the input on game start. Focused = you can type in
- ➔ Interactable includes touch/mouse both.
- ➔ Difference between Select() and Focus() is select checks if the input field is in a list.



Main Settings

Auto Focus	✓
Interactable	✓
Max Character	20
Typing Symbol	|
3D Text	Text (Modular3DT
Background	Background (Mesl
Background Metarial	
In Focus	White Dark
Out of Focus	Grey
Disabled	None (Material)
Type Sound	None (Audio Clip)
Audio Source	None (Audio Source

Unity Events

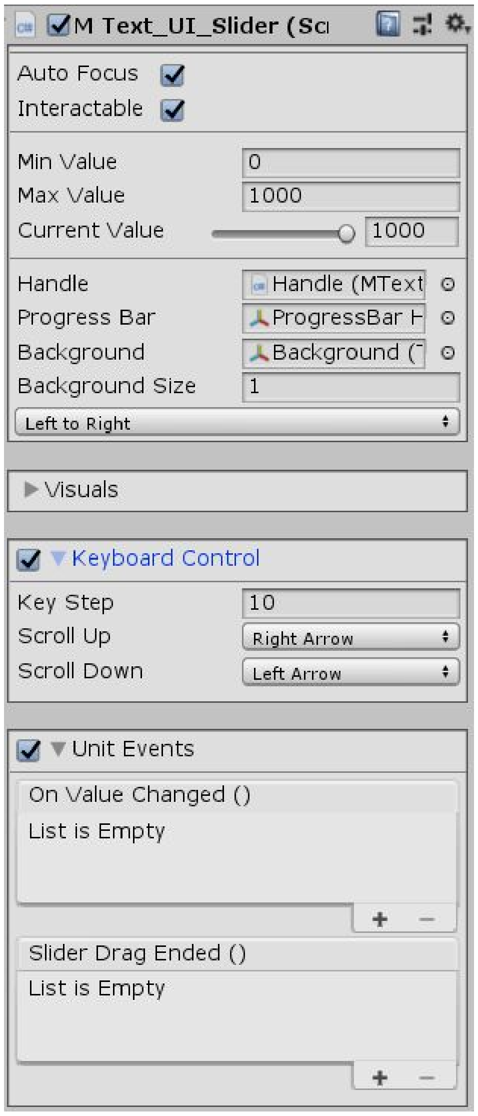
On Input ()
List is Empty

On Backspace ()
List is Empty

On Input End ()
List is Empty

3D Slider (MText_UI_Slider.cs):

➔ `UpdateValue(int newValue)/ UpdateValue(float newValue)` use these to update slider via code



M Text_UI_Slider (Sc

Auto Focus ✓
Interactable ✓

Min Value: 0
Max Value: 1000
Current Value: 1000

Handle: Handle (MText
Progress Bar: ProgressBar H
Background: Background (
Background Size: 1
Left to Right

Visuals

✓ Keyboard Control
Key Step: 10
Scroll Up: Right Arrow
Scroll Down: Left Arrow

✓ Unit Events
On Value Changed ()
List is Empty

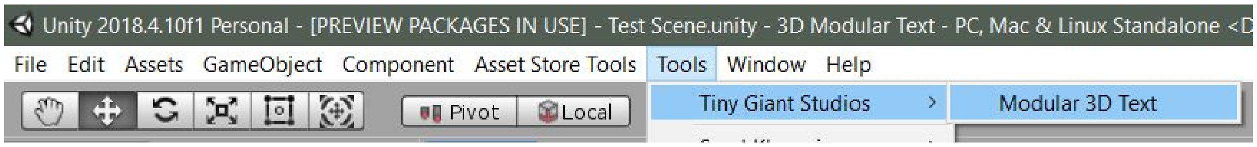
Slider Drag Ended ()
List is Empty

➔

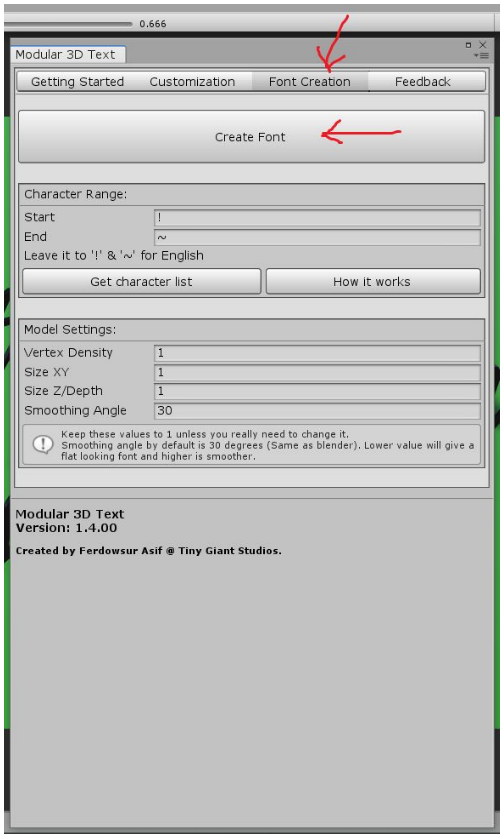
Creating fonts:

In Editor (Video Tutorial: https://youtu.be/m9JwBc-0DUA):
[Supports only TTF file formats. Most fonts found online are in TTF format] [Added in 1.4.0]

1. Go to Tools>Tiny Giant Studios>Modular 3D Text


Unity 2018.4.10f1 Personal - [PREVIEW PACKAGES IN USE] - Test Scene.unity - 3D Modular Text - PC, Mac & Linux Standalone <D
File
Edit
Assets
GameObject
Component
Asset Store Tools
Tools
Window
Help
Pivot
Local
Tiny Giant Studios
Modular 3D Text

2. Select Font Creation Tab and click Create Font



Character Range:

Start: !
End: ~
Leave it to '!' & '~' for English

Get character list | How it works

Model Settings:

Vertex Density: 1
Size XY: 1
Size Z/Depth: 1
Smoothing Angle: 30

Keep these values to 1 unless you really need to change it.
Smoothing angle by default is 30 degrees (Same as blender). Lower value will give a flat looking font and higher is smoother.

Modular 3D Text
Version: 1.4.00

Created by Ferdowsur Asif @ Tiny Giant Studios.

3. Select your font file from anywhere in the computer

4. Give a few seconds to process and then select location inside your project where your mesh/font will be saved.

Your font is ready to be used.

For different Languages: (Video tutorial: https://youtu.be/JN_DSmdiRSI)

Before creating the font, Click the Get Character List Button which will take you to https://unicode-table.com/en/#control-character In character range, for your language, from the link , enter the first character and the last character. This will make the created font include every character inbetween them. Then, Click create font.

Fast way using blender: (Video tutorial : https://youtu.be/lbRe4-wlf74)

1. **Install blender** if you don't have it already.
2. **Open the font extractor folder** inside the modular 3d text folder. (Unzip it)
3. **Open the blender file.**
4. Click **run script.**
5. **Select extract font** from the left toolbar
6. **Select your font.**
7. Do any changes you want in this step, like retopology, optimization (Check for optimization tips below) according to your needs. Or just skip this step if you want.
8. File>Export>**Export as FBX** to your project. Default export settings are fine.
9. In unity, right click in the project window, **Create > Modular 3d Text>New font.**
10. **Assign exported FBX file to the font. Click the create/recreate button.**
11. **Font is ready to use.**

Optimization:

(To people who are knowledgeable about optimization, do your own thing. This is just a fast and simple way to do things, not perfect.)

The font created in blender is not optimized. Didn't include auto-optimization in blender script because some fonts have very different geometry than expected from an average font which would result in losing details or not enough optimization.

To reduce vertices. A fast but not the best way to optimize it would be,

- Select all objects (Press A in scene view).
- Press Tab to enter edit mode. (If you can't enter edit, select and deselect an object by left clicking one and then, an empty area, Select everything again (Press A) then Press Tab to enter edit mode)
- Go to the Mesh menu (on the top menus on view window)
- Cleanup > Limited Dissolve.
- Cleanup > Remove doubles.

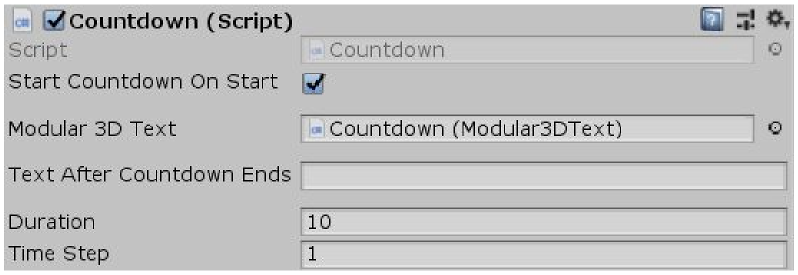
This can reduce a lot of unnecessary vertex but sometimes result in few characters having strange geometry. You can Undo & deselect
If the font looks good enough, export and use.

2. Slow way:

- In unity, right click in the project window, Create>Modular 3d Text>New Font.
- In the list you see after selecting the font, add a new item. Assign any object you want to that item and type a character for it besides it. Now when you type the character in, that object is used.

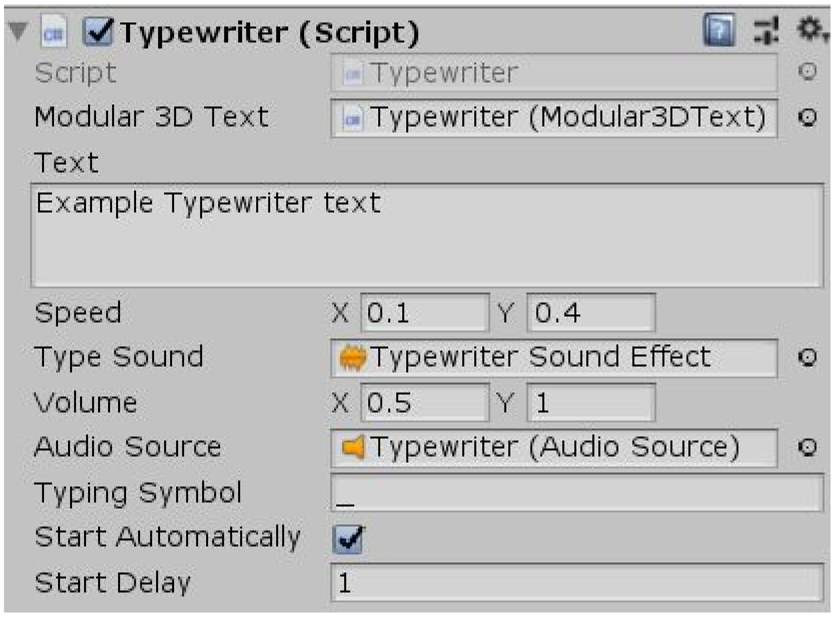
Fix positioning + Rotation + scaling for the font. You can fix it for all the characters at once by specifying it in the font file and + or you can assign a parent to the object, add “MText_Character.cs” component to the parent and fix transform for each character individually. That script shows the default expected scale and size.

About the included examples:



Countdown (Script)

Script	Countdown
Start Countdown On Start	✓
Modular 3D Text	Countdown (Modular3DText)
Text After Countdown Ends	
Duration	10
Time Step	1



Typewriter (Script)

Script	Typewriter
Modular 3D Text	Typewriter (Modular3DText)
Text	Example Typewriter text
Speed	X 0.1 Y 0.4
Type Sound	Typewriter Sound Effect
Volume	X 0.5 Y 1
Audio Source	Typewriter (Audio Source)
Typing Symbol	_
Start Automatically	✓
Start Delay	1

Speed and volume’s x and y value are the minimum and maximum value. Keep them the same if you don’t like them to fluctuate. A little randomization makes them feel natural.

Script references:

All scripts are in namespace MText. So, don’t forget to add using MText on top of the script when referencing scripts on this tool.

Script references are now maintained on online here: *(new documentation link)*