3D Text Effects - Manual

Quick Folder Overview

This is quick overview of the folders of 3D Text Effects:

Animation Presets Folder

this folder contains various "base animations" that can be used with the functions of 3D Text Effects. It has 3 subfolders:

1. Empty animations - are animations that are used as placeholders when you don't want to animate a glyph
2. Enter Exit Animations - are animations of the glyph appearing or disappearing. You can take any "appearing" animation and turn it into a " disappearing " animation by setting animation mode to "Reverse".
3. In Place Animations - are standard animations (anything that does not make the glyph appear or disappear)

Example Scenes Folder

This folder contains some example scenes.

1. Design ideas : in the scenes Gold . Metal ,Sci-Fi,Candy and Blood. It also contains a scene called
2. Showing all text prefabs: in the scene "Prefabs Scenes"
3. Example of all animation sequncers : in the sub folder Sequence Animations

4.

Font folder

This folder contains ready made 3D fonts create from a curation of open SIL licence fonts. The original font files are in the sub folder "Base Fonts"

Text Animation Prefabs

This folder contains 7 ready text prefabs that can be placed in your games instantly.

Tutorials Folder

contains 4 tutorial scenes and scripts which you can learn from.

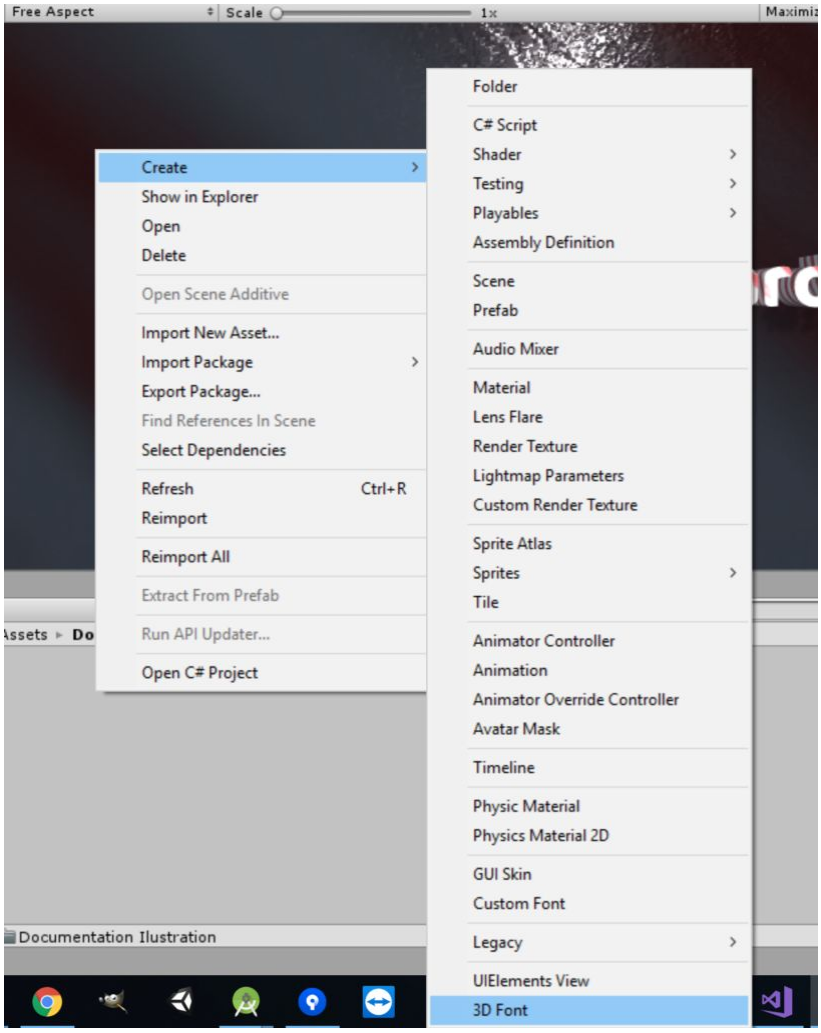
Material and Textures Folder

Contains a curation of CC0 PBR materials that can be used with your 3D texts.

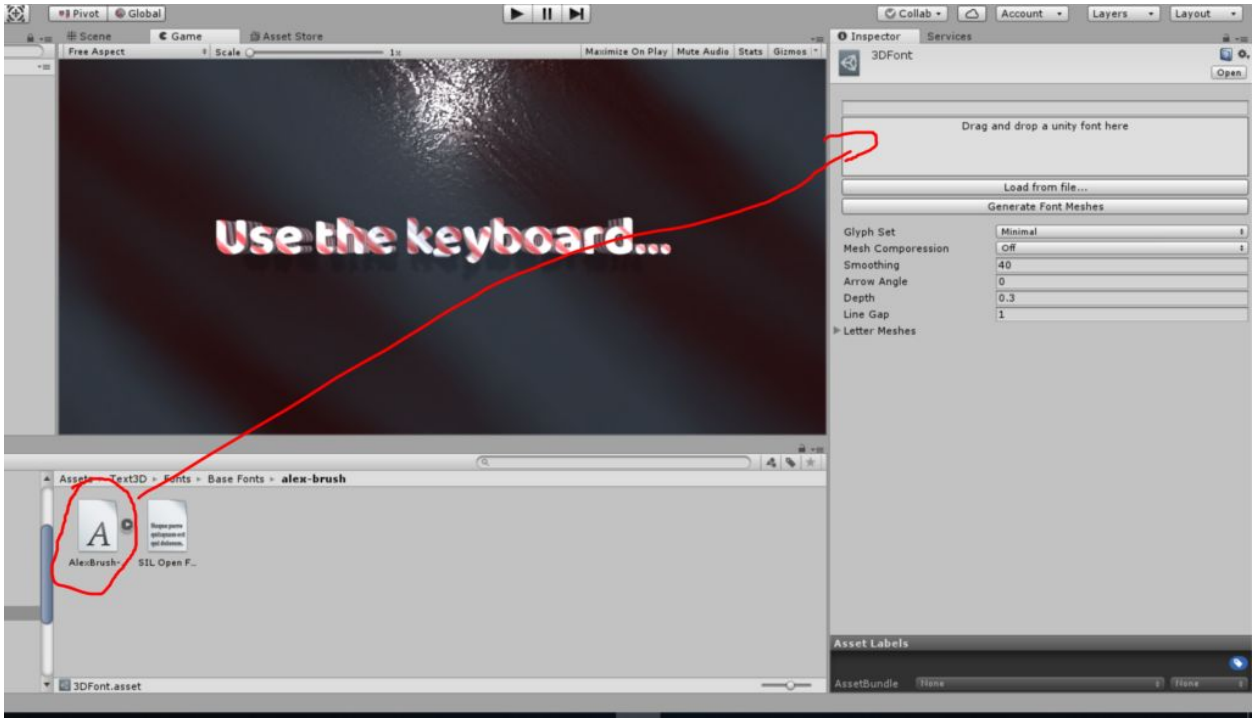
How To Use 3D Text Effects

How to create a 3D Font

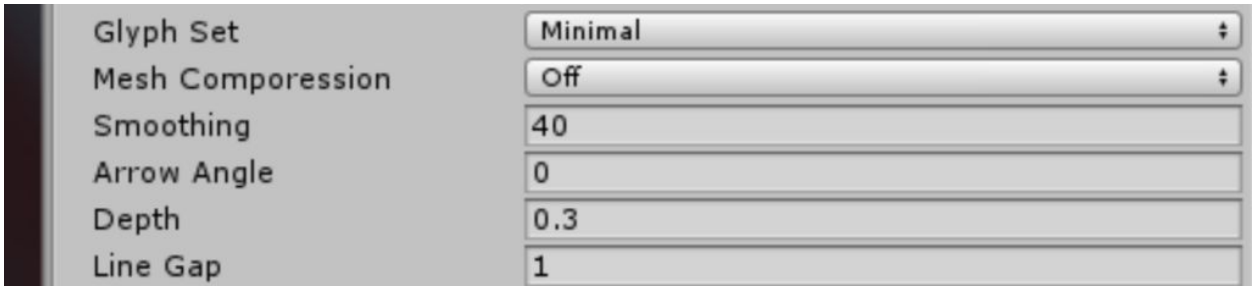
1. Right click the project view and select Create / 3D Font:



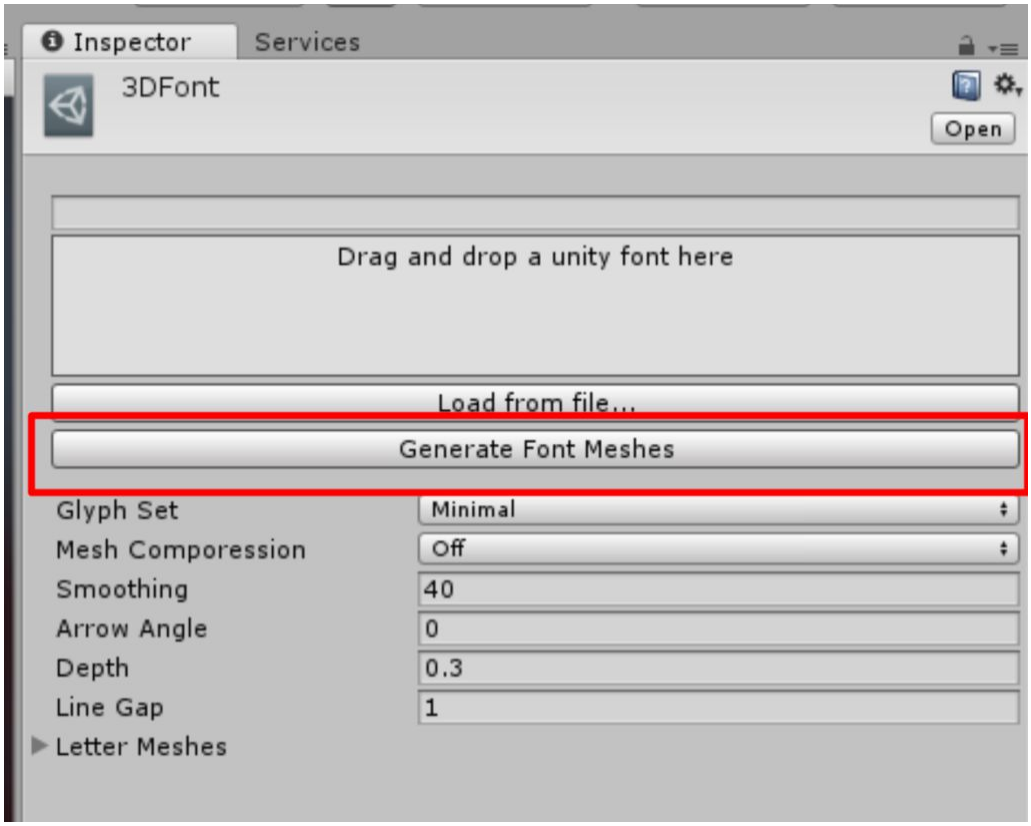
2. Drag a unity font object into the 3D font inspector:



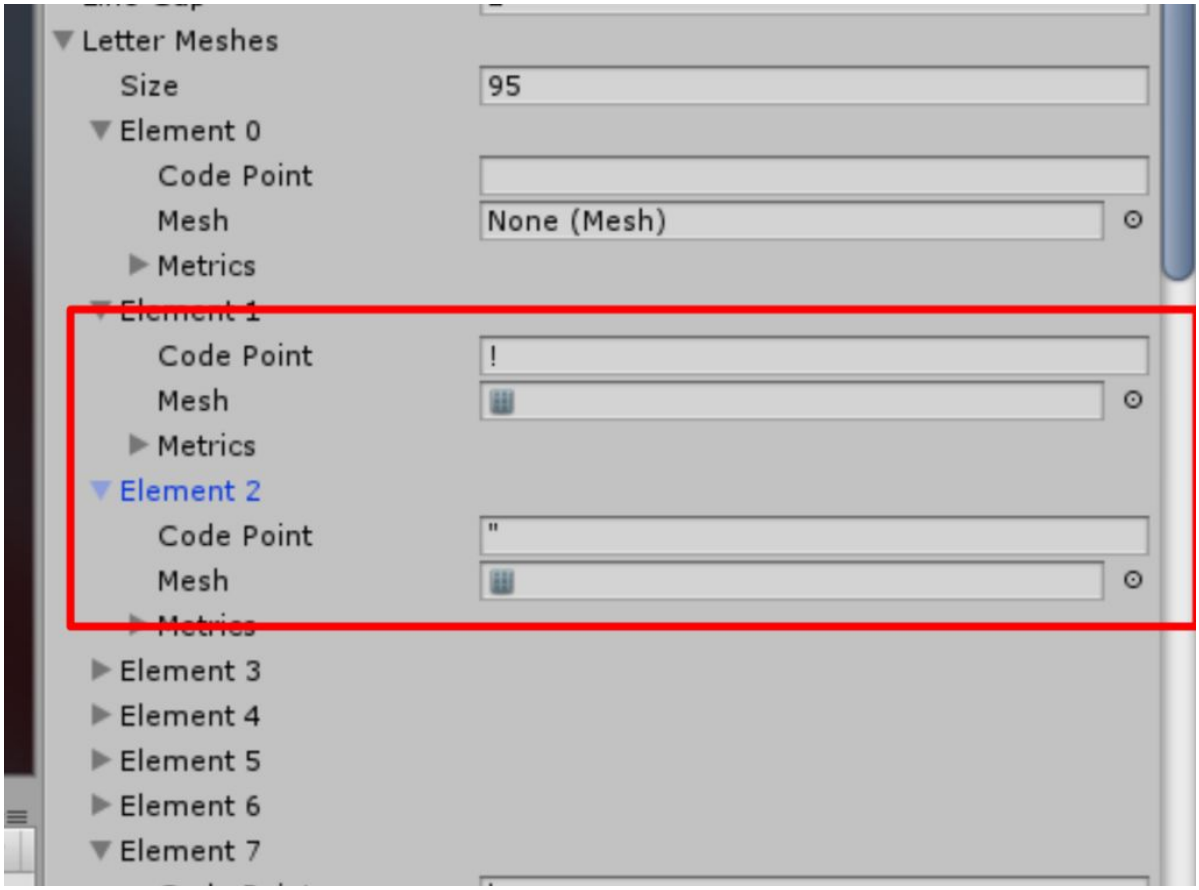
3.Select a glyph set and parameters (Learn more):



4.Click the "Generate Font Meshes" button:



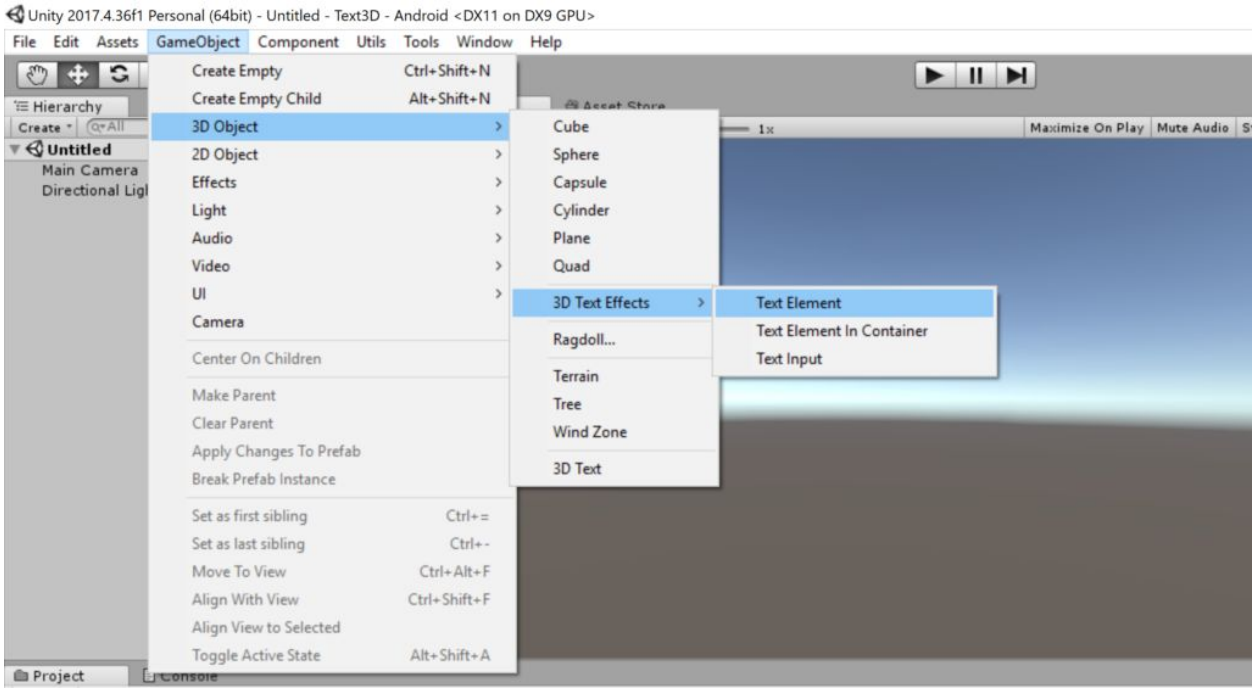
5. You should be able to see results in the "Letter Meshes" array:



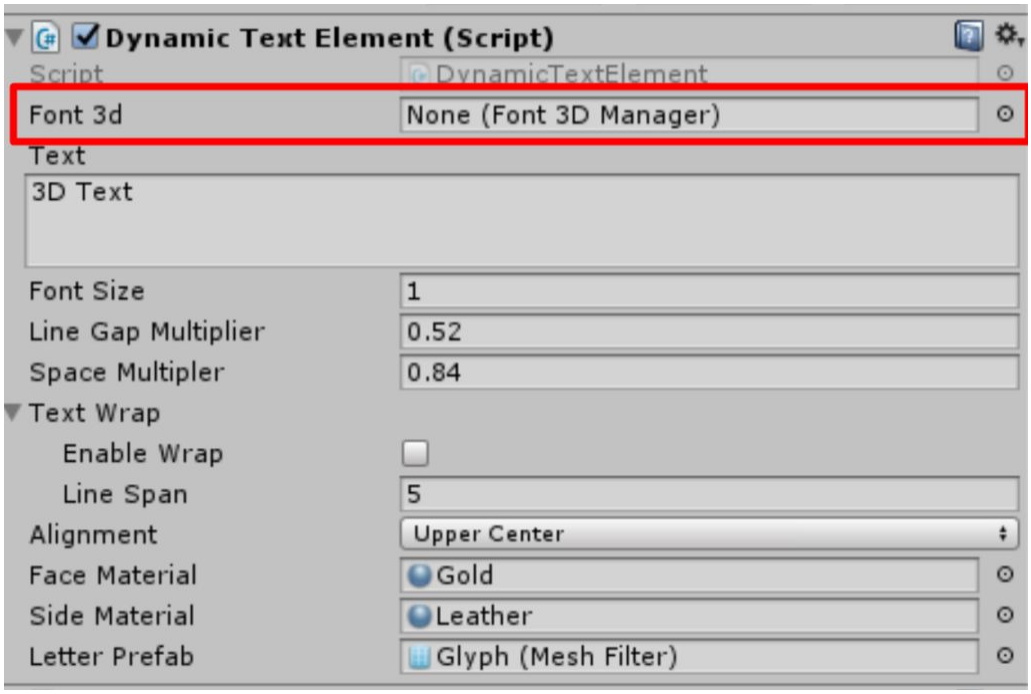
The font is now ready to use!

How to add a 3d Text element

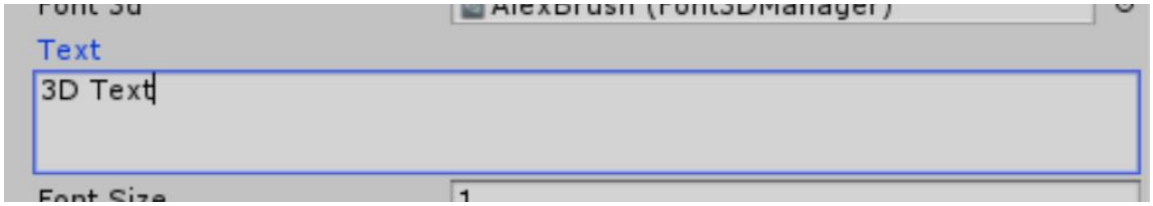
1. Go to GameObject/3D Object/3D Text Effects/ Text Element



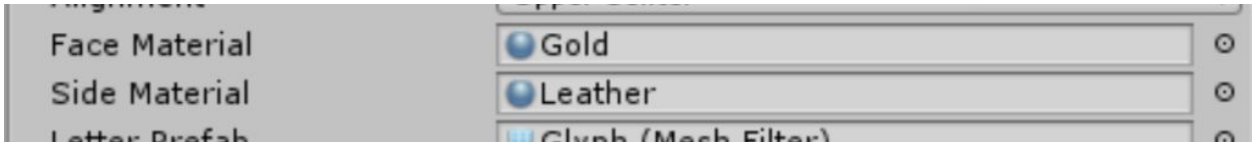
2. Drag and drop a 3D font to the text element:



3. Type in some text :



4. Assign the font materials :



Using the text object

The text object is just like any unity game object. You can place it in your scene or in prefabs. You can also modify the text from script

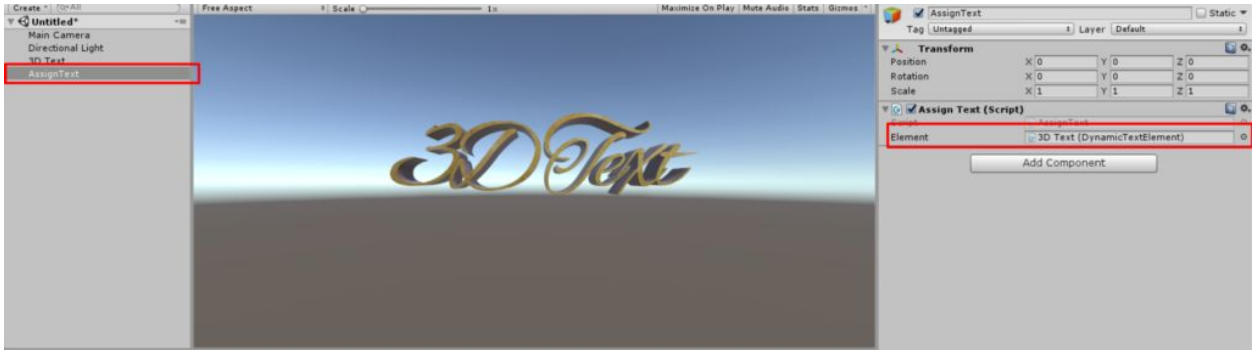
How to assign text from script

1. Create a new monobehviour:

```
public class AssignText : MonoBehaviour {

    public DynamicTextElement Element;

        // Use this for initialization
        void Start () {
      Element.Text = "New Text";
        }

        // Update is called once per frame
        void Update () {

        }
}
```

as you can see , the text can be changed by setting Element.Text

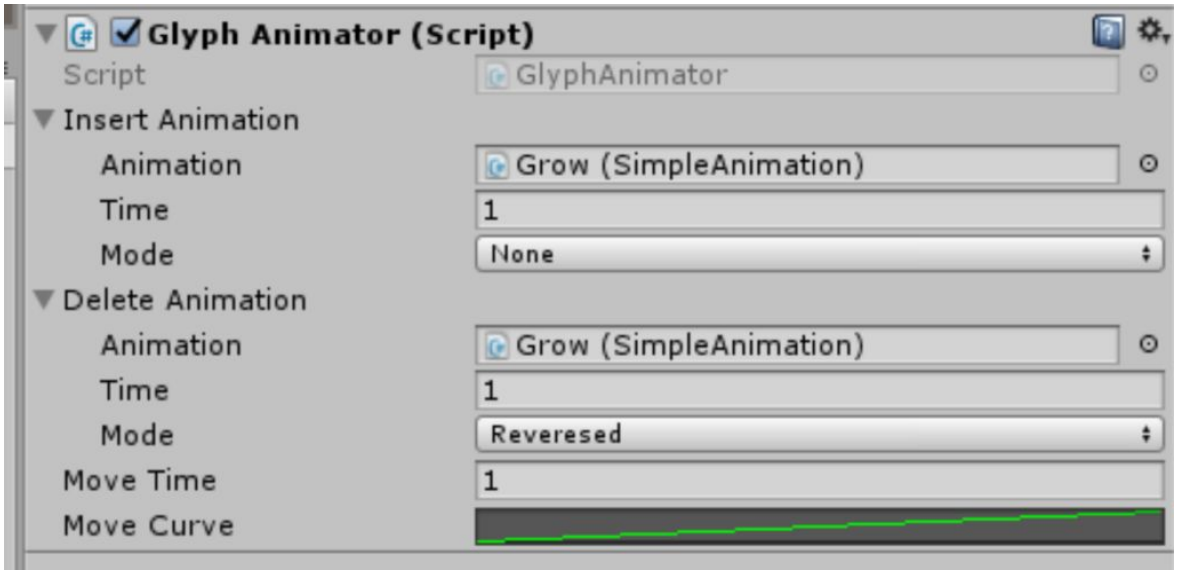
2. Create a new game object and add the AssignText component. Drag the text element into the object:



How to set the default animations of the text element

Each text element can have default enter and exit animations assigned to it. The default animations are show when the text is created or when it is animated from script.

In order to set up default animations add the GlyphAnimator component to your 3D Text :



You can now drag and drop any SimpleAnimation object as an Insert/Delete animation. You can find a handful of ready made SimpleAnimation objects in the Animation Presets folder.

You can select the animation time and mode.

Now the text can be easily animated from script.

How to use a 3d text animation prefab in a scene

The following article is based on the scene in Text3D/Tutorials/Quick Text Creation.

We would often want to show text effects in response to a game event. This can be done easily by saving the text effect into a prefab and then instantiating it using script. 3D Text Effects has a method that does this for you , here is how it works :

1. Create a new mono behavior. This behavior will trigger the text effect when you would like to show. Add the following property to the script :

```
public GameObject PrefabObject;
```

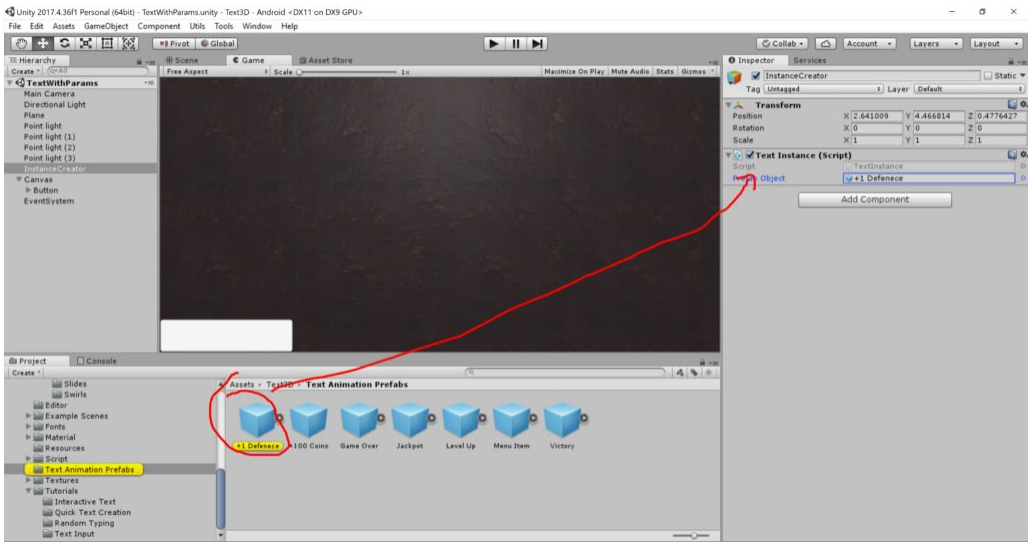
2. Now we can show the prefab object from script using the following call :

```
DynamicTextElement.ShowTextEffect(PrefabObject, showText);
```

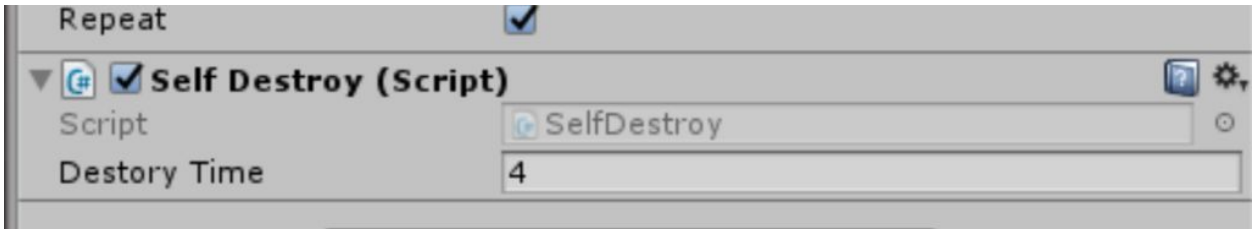
this method accpets a prefab object and text string. The text string can be formatted in any way you would like. In the this tutorial , it is set to a random number of defense points:

```
string TextFormat = "+{0} Defense";
 int RandomNumber = Random.Range(1, 100); // select a random number. In a real life use case,  this number can be something relevant to your game instead of a random.
 string showText = string.Format(TextFormat, RandomNumber); // format the text
```

3. Drag and drop the text prefab into the script object:



4. Make sure to destroy the text effect game object after you are done with it. This can be done by adding a SelfDestory comopnent to the prefab :

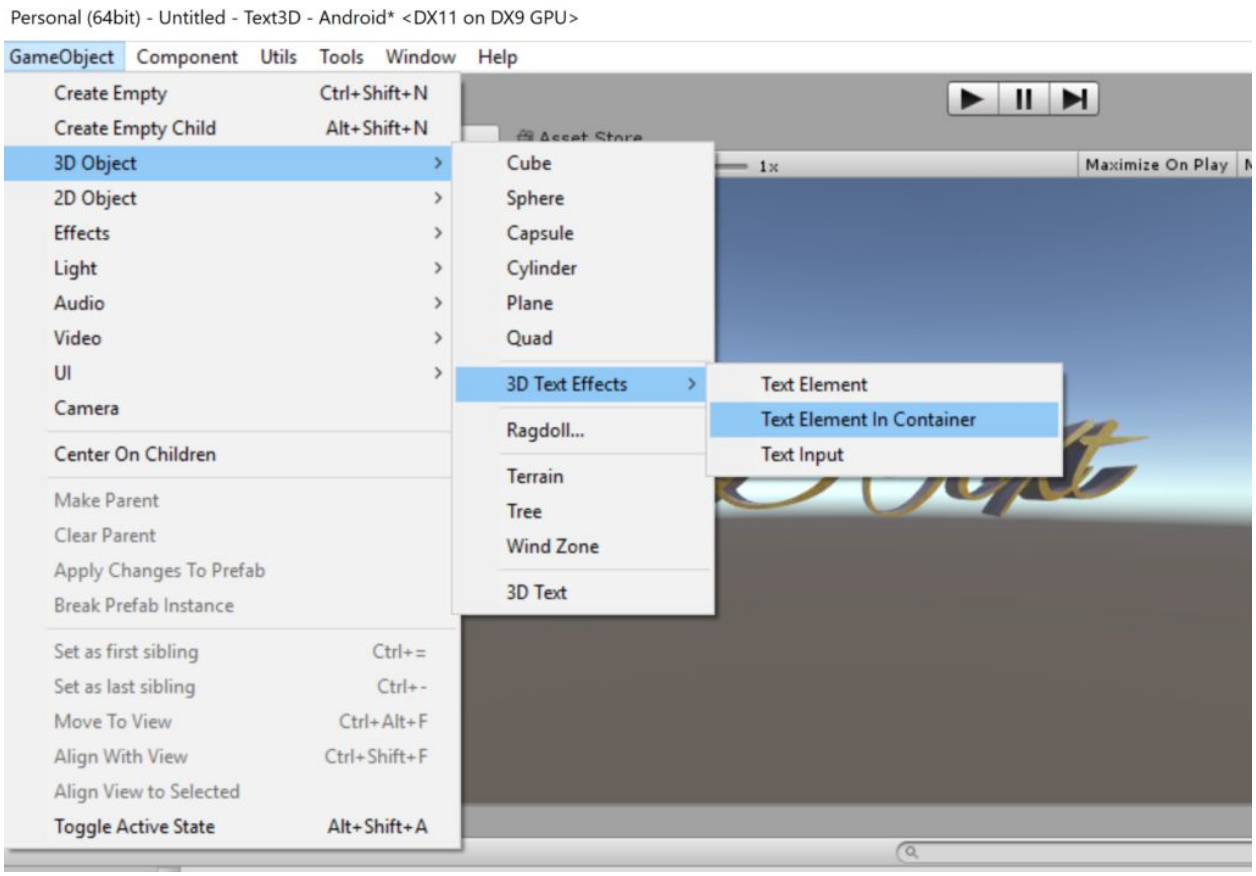


Run the scene. You should see the 3D text effect going.

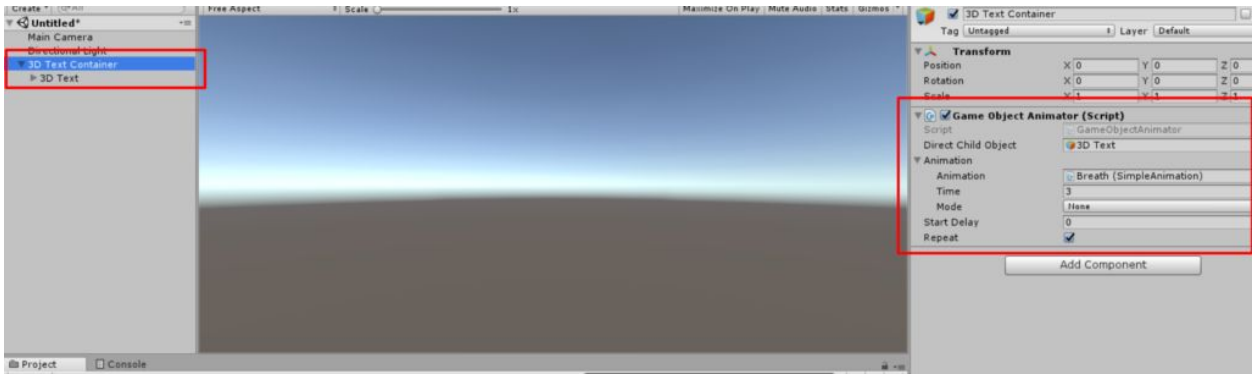
How to animate the parent object

Before animating individual glyphs , you may want to animate the entire text object. For example make it spin or grow. This can be done easily by adding the text object to a container object. Luckily , you can do this using the Game object menu.

1. Add a new object in a container by clicking GameObject/3D Object/3D Text Effects/Text Element in Container:



2. Select the container object , it has an ObjectAnimator component. Drag and drop a [SimpleAnimation]object into the Animation property.



3. You can select if the animation is Repeating , it's time span and it's start delay

[Learn more about GameObjectAnimator]

How to create a simple sequenced animation

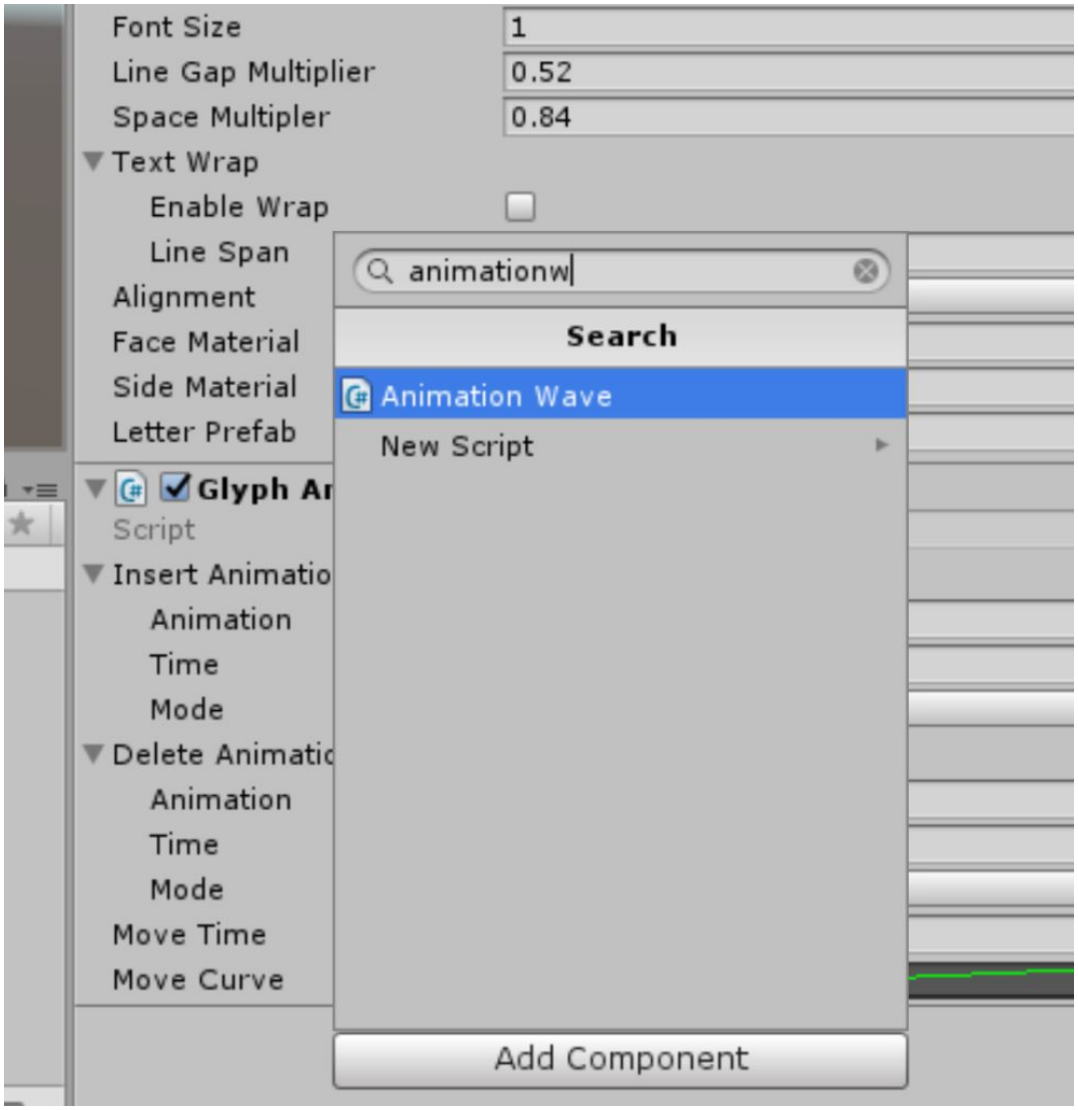
The key to any good 3D text effect is motion. On many occasions we would want to animate each glyph of the text and to create a lively animation. When we want to do this we can either write an animation script , or use the configurable animation scripts that come with 3D Text Effects.

For this tutorial we will create a simple wave animation:

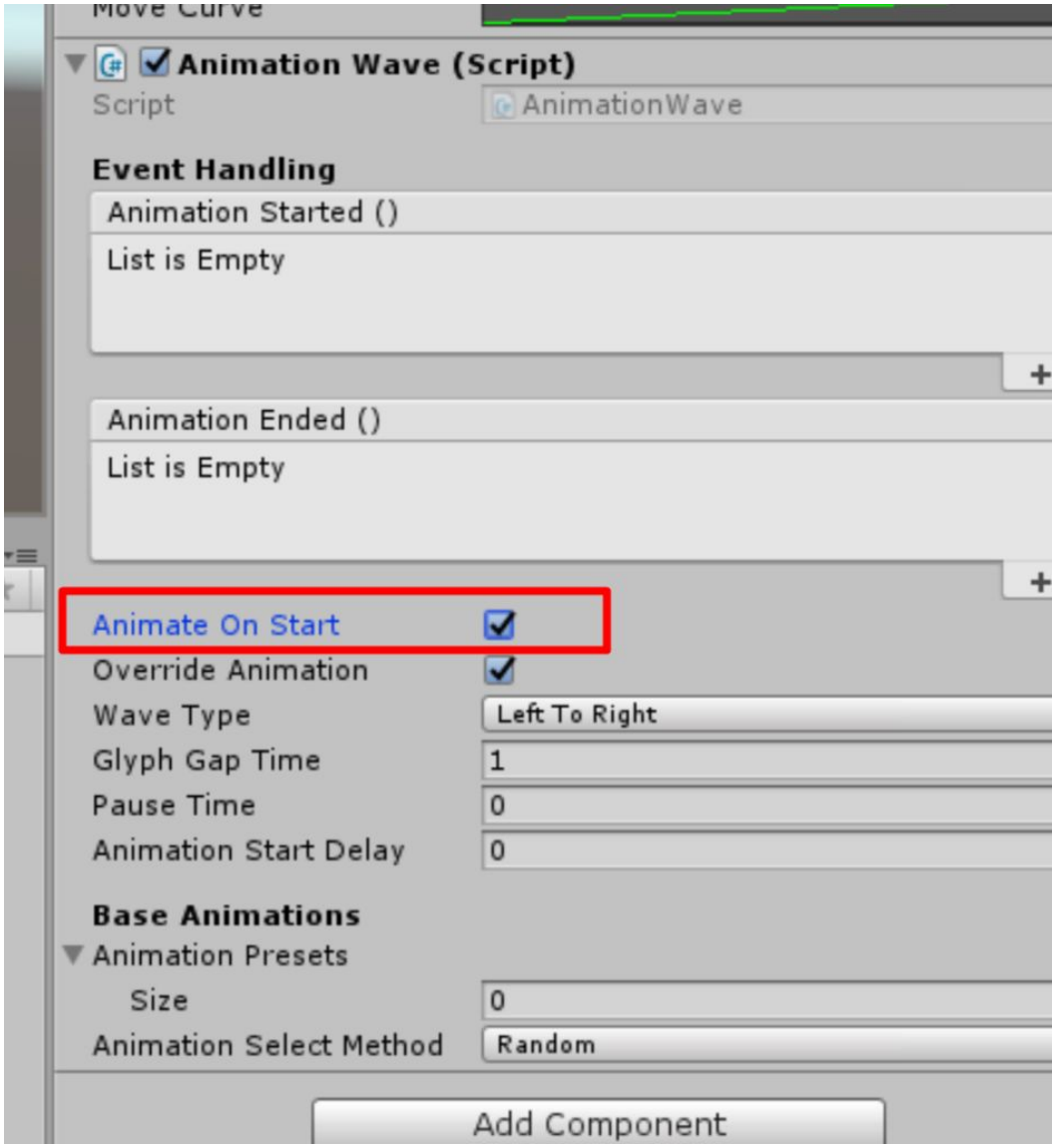
1. Add a 3D text object to your scene.

2. Add a glyph animator object to your 3D Text . **If you want the wave animation to start with no text on screen, assign the Invisible animation preset as the default insert animation.** this is useful to create text enter animations.

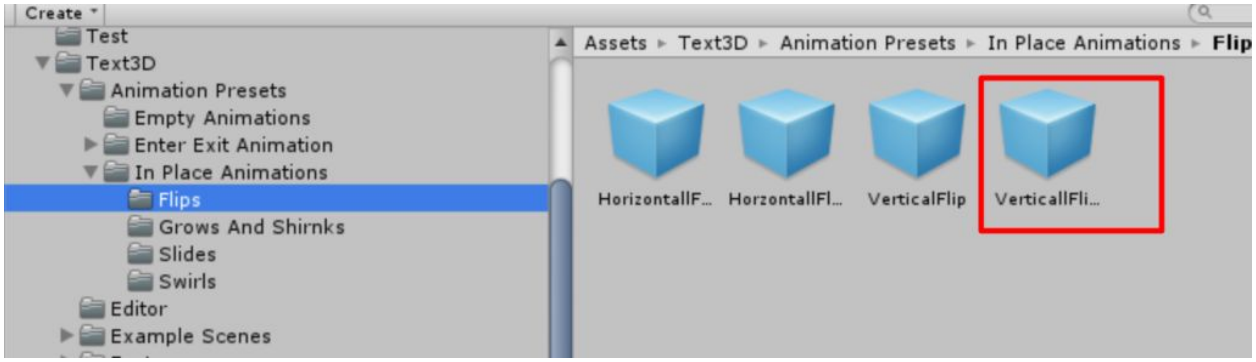
3.Add an animation wave component to your text object

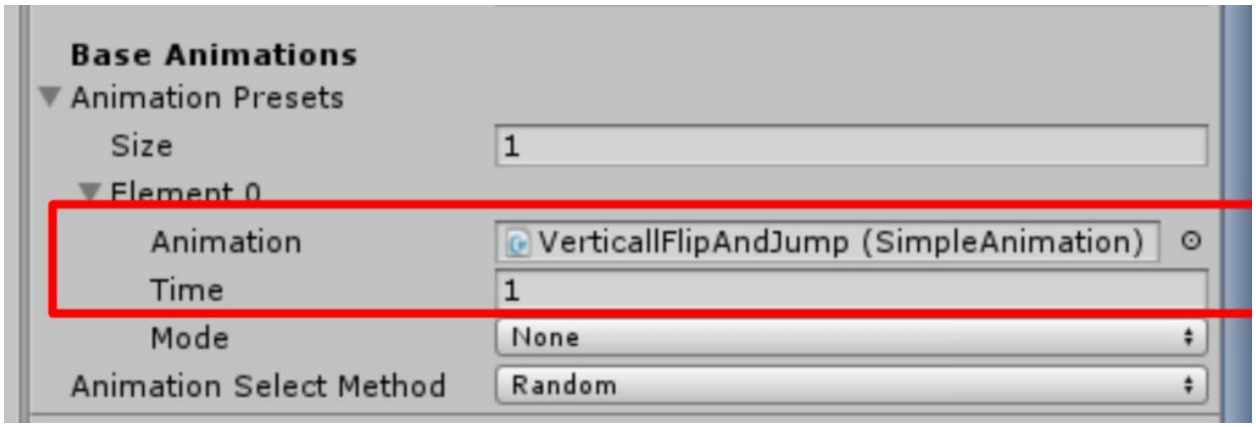


4. Toggle the "Animate on start" option. This will make the wave animation run when the OnStart method of the object is called:



5. Set the base animation to VerticalFlipAndJump, by dragging it into animation presets:





Make sure to configure the animation time as well.

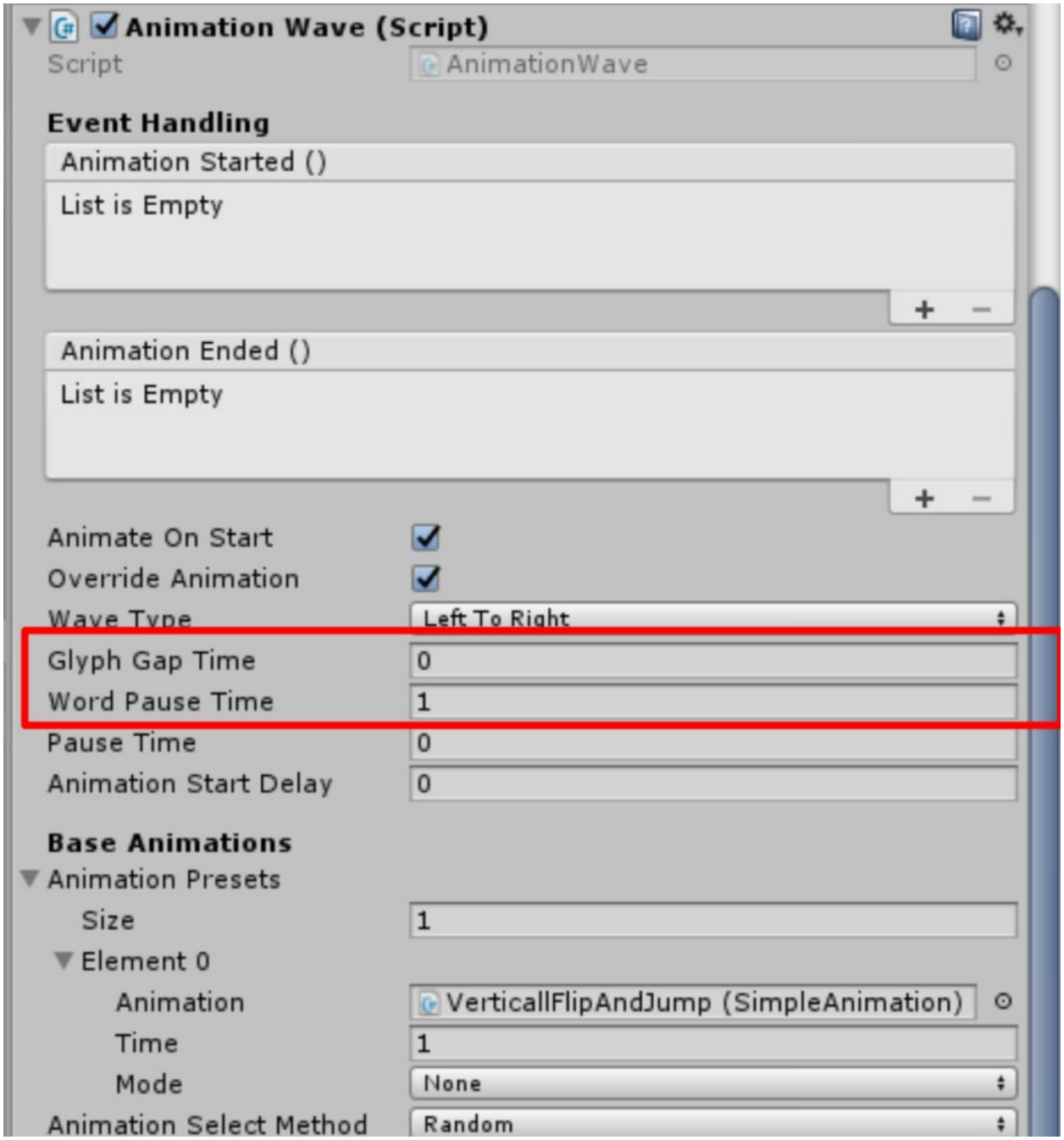
6. Run the game. You should see each glyph perform a flip and jump

Learn more about the AnimationWave component

How to make a word based animation

The set up to this tutorial is similar to “How to create a simple sequenced animation”. This time however , instead of animating each glyph at a time , we will animate each word at a time.

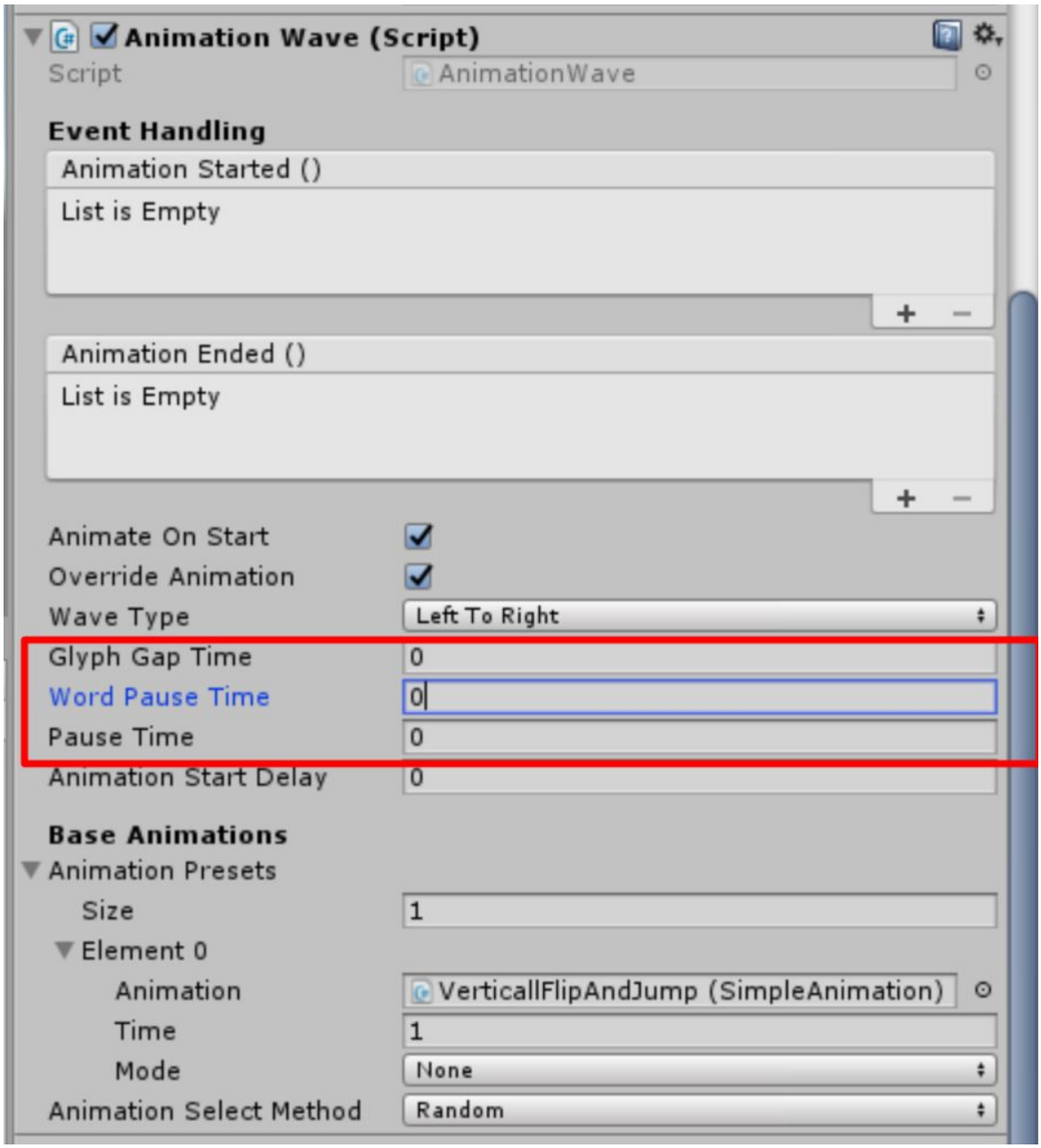
This can be done by changing the pause times of the Animation Wave component. If we set "Glyph Gap Time" to 0 and "Word Gap Time" to 1. It means that the skip between each glyph is imidiate , and the skip between each word is 1 second. This will make the glyphs animate for each word at a time.



How to instantly animate all the text letters

The set up to this tutorial is similar to “How to create a simple sequenced animation”. This time however , instead of animating each glyph at a time , we will animate all the glyphs togather.

This can be done by setting all the pause times to 0. This means there is no time gap between the glyph animations:

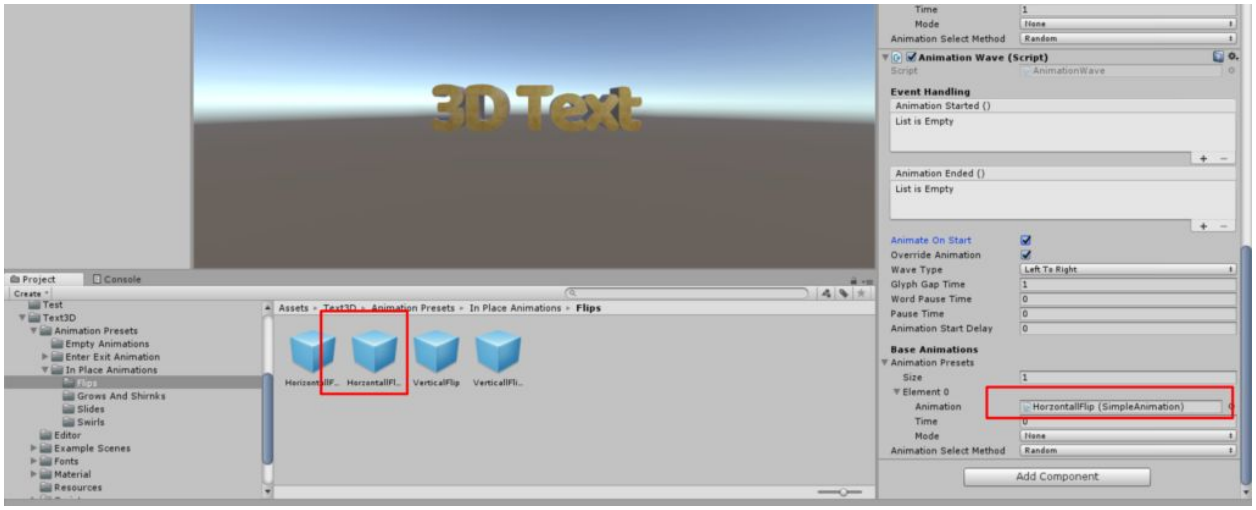


Run the scene. All the glyph should flip instantly.

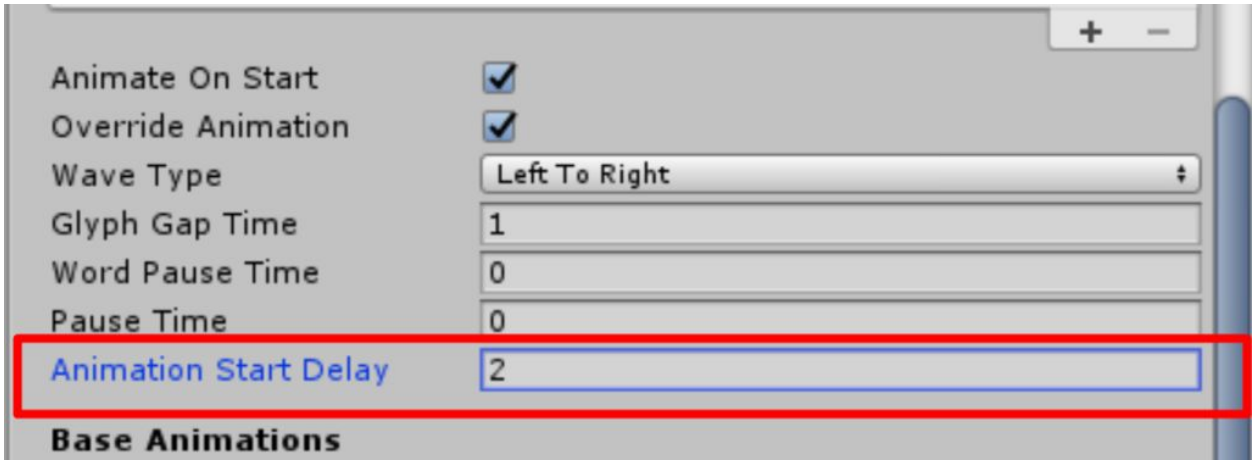
How to create a multiple step animation

The set up to this tutorial is similar to “How to create a simple sequenced animation”. This time however , We will add another step to the animation. The Animation wave component can be added multiple times to a gameobject. You can use this to combine wave sequences and create a more complex animation. It can be done in the following way:

1. Add Yet another Wave Animation component to the text object, This time select the horizontal flip animation:



2. Select a time delay for the second wave animation :



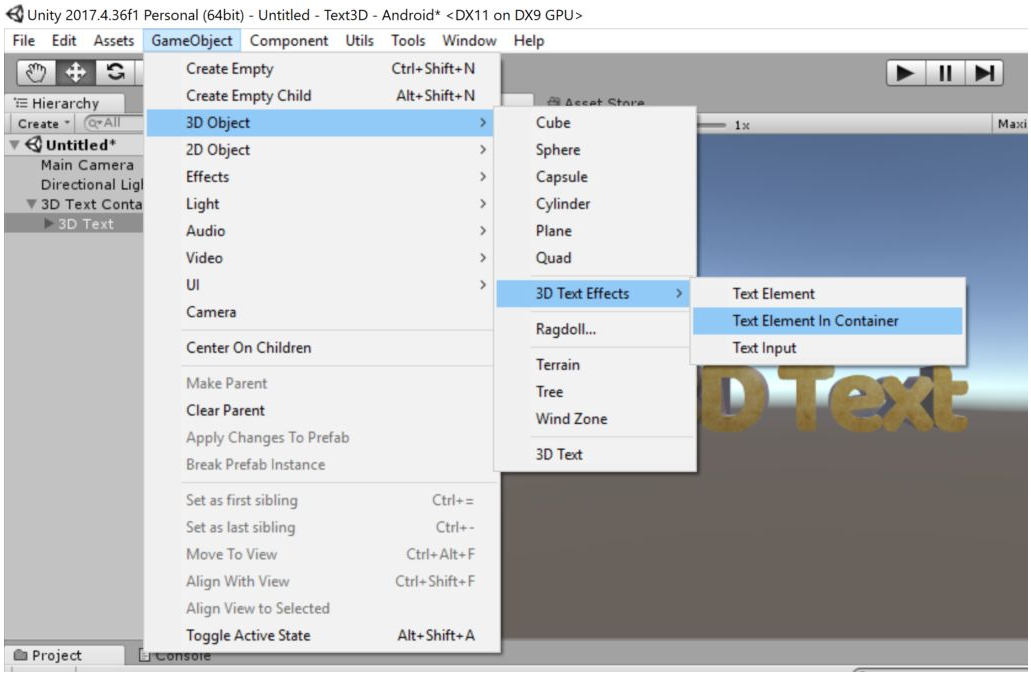
Setting the time delay to "2" will make the animation start 2 seconds after the OnStart method is called

Run the scene. You should see the second wave start after 2 seconds.

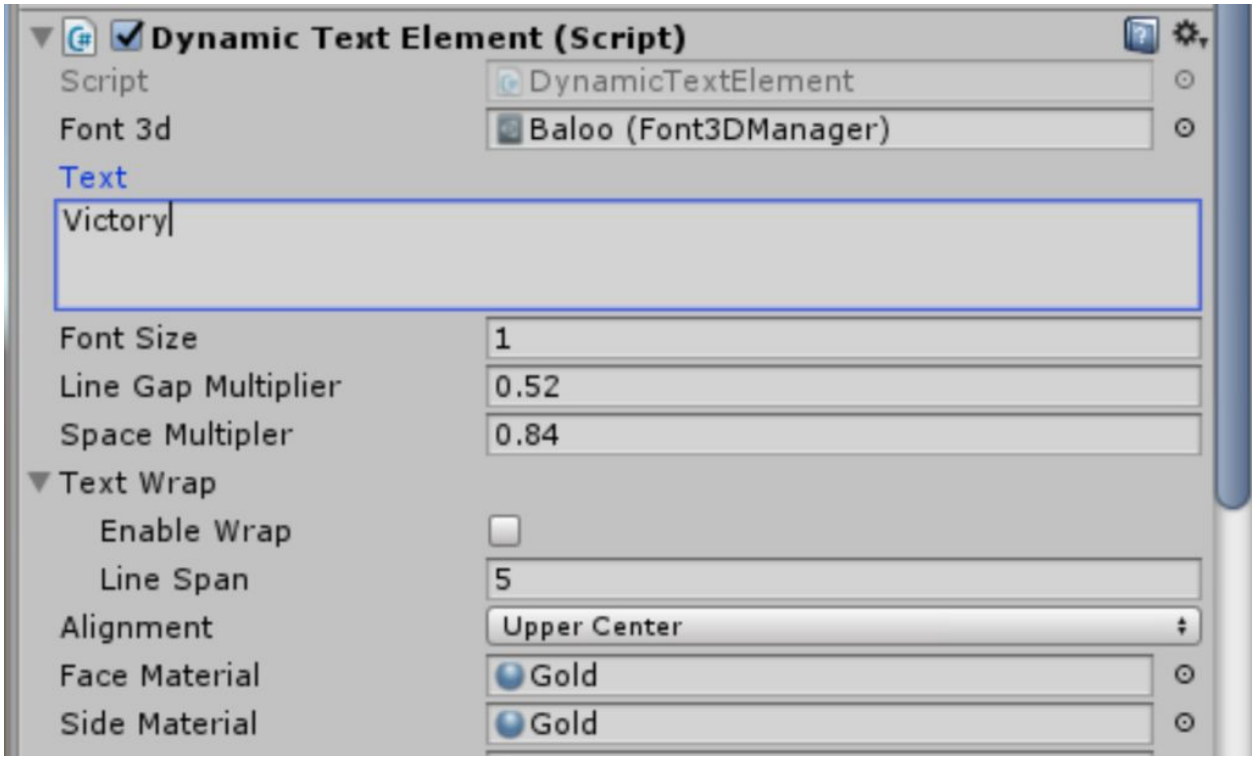
How to recreate the victory animation

This tutorial will recreate the "Victory" animation that can be located in Text3D/Text Animation Prefabs/Victory . The victory animation has 3 stages. Here is how they can be recreated:

Create a new 3D text object in a container



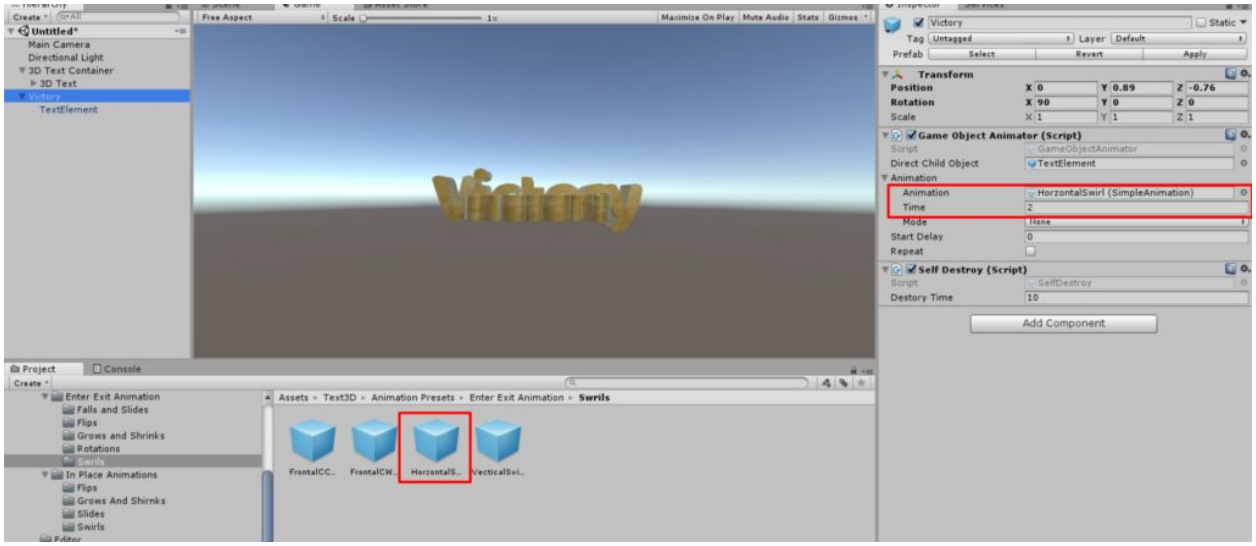
Assign both the font and side materials to "Gold" and set the text to "Victory"



Step one - Spinning the entire Text

The first step is to create a swirl of the entire victory text in a similar way to [this tutorial](#).

1. Select the container object and set it's animation to horizontal swirl:
2.

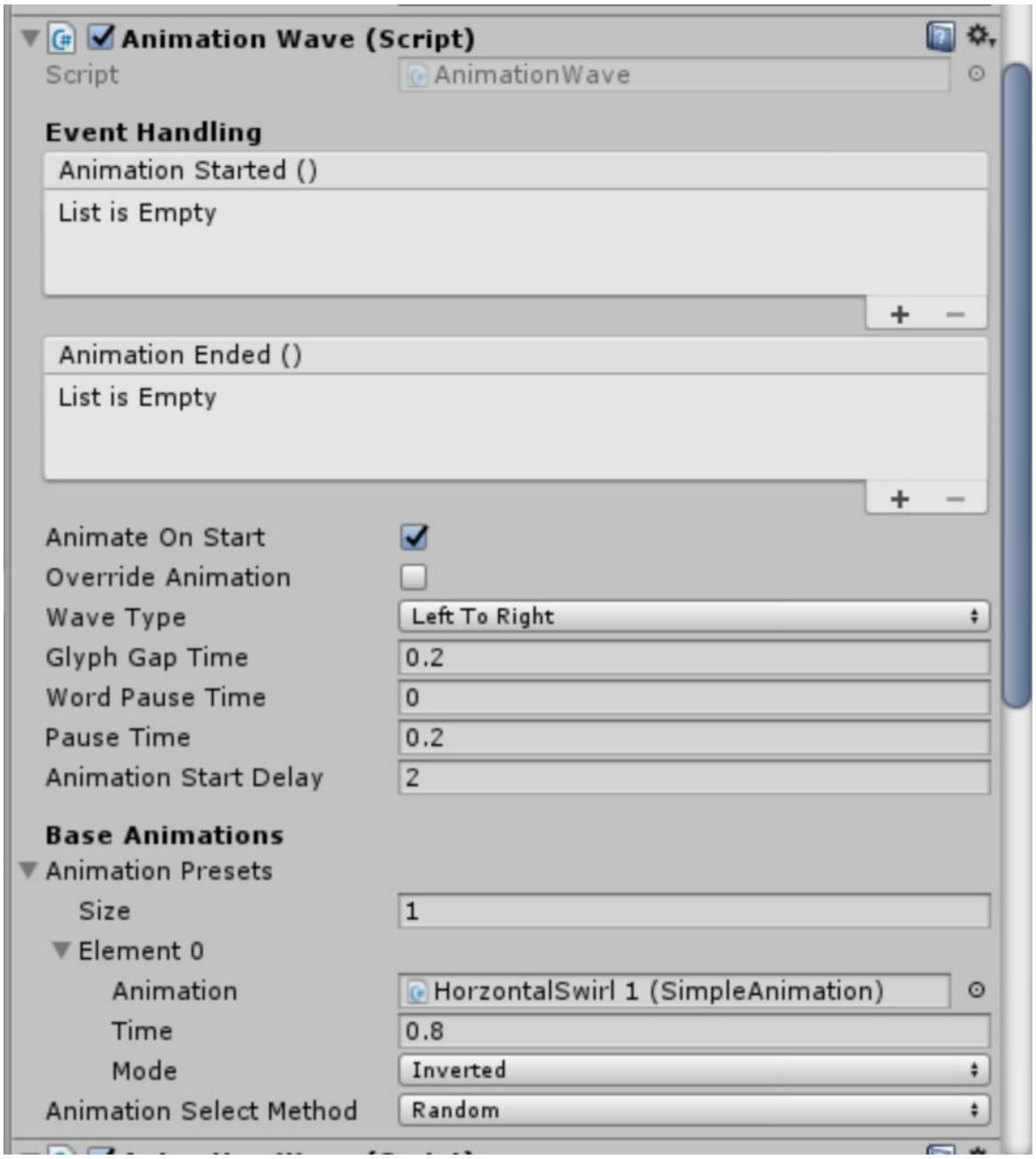


Set the time duration to 2 seconds , and Repeat to false.

step two - sPINNING EACH gLYPH

The second step is to spin each glyph , this can be done by adding a wave animation in a similar way to [this tutorial]

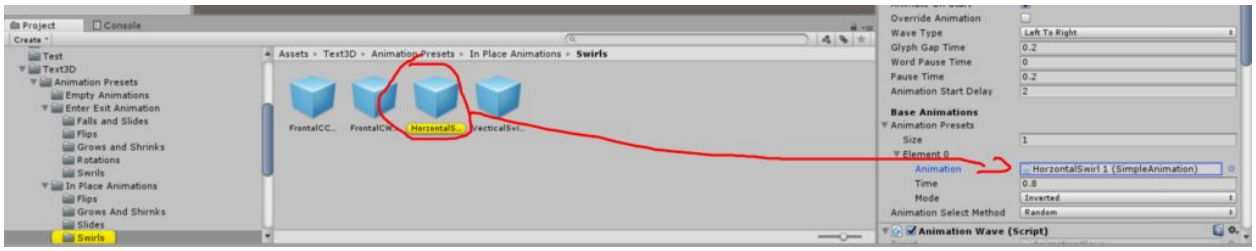
1. Select the text element and add an animation wave component to it :
2.



Toogle "Animate on start" Set the "Glyph Gap Time" to 2.

We would like this step to start after step one is completed. Step one is set to 2 seconds , so we can make the animation wave start a 2 second mark by setting the "Animation Start Delay" to 2.

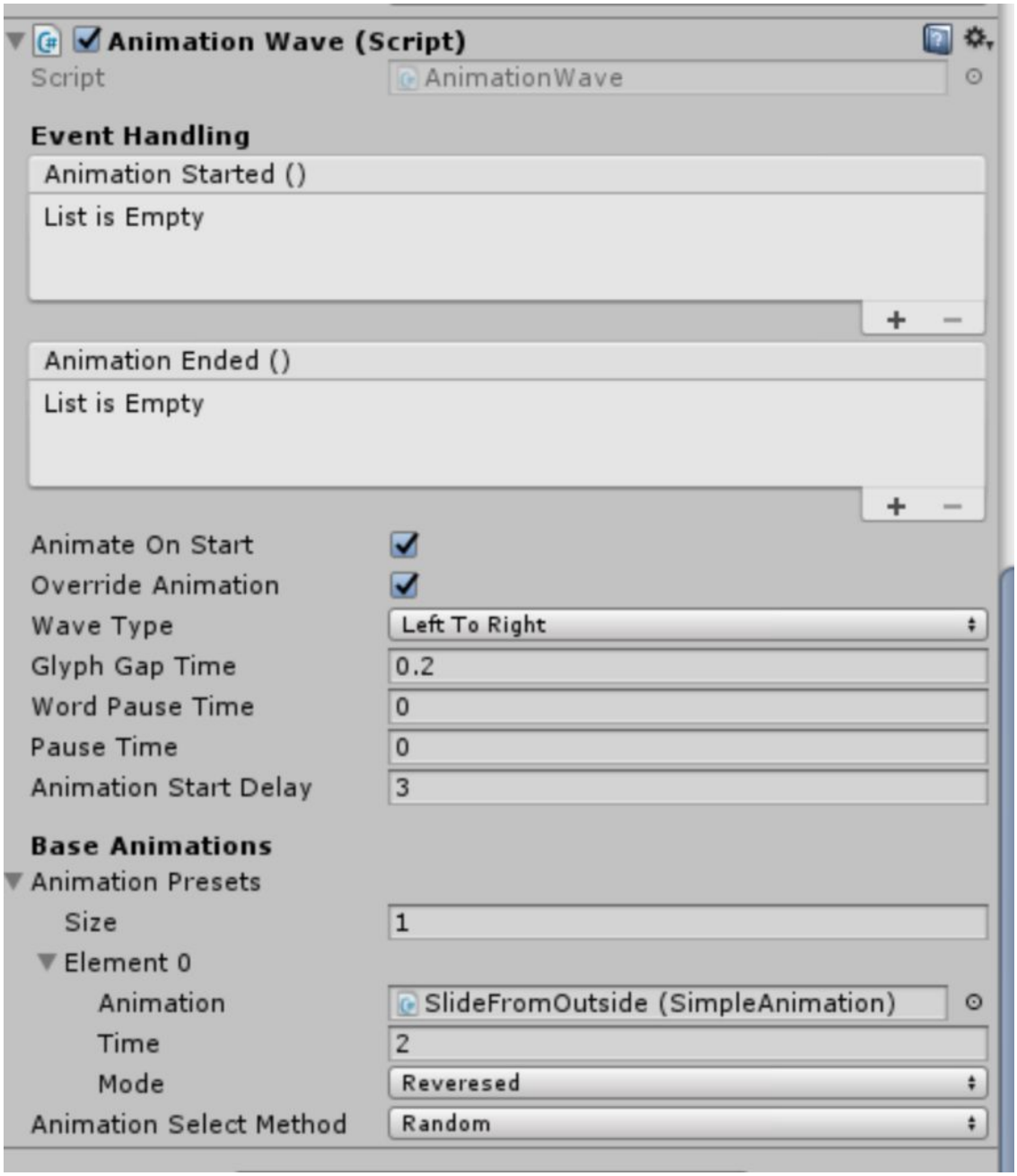
2.Drag "HorizontalSwirl 1" into the animation presets array and set it's "Time" duration to 0.8:



STEP Tree - Exit each glyph

At the end of the victory animation , each glyph slides to the outsides of the screen. this can be done by adding another wave animation in a similar way to [this tutorial]

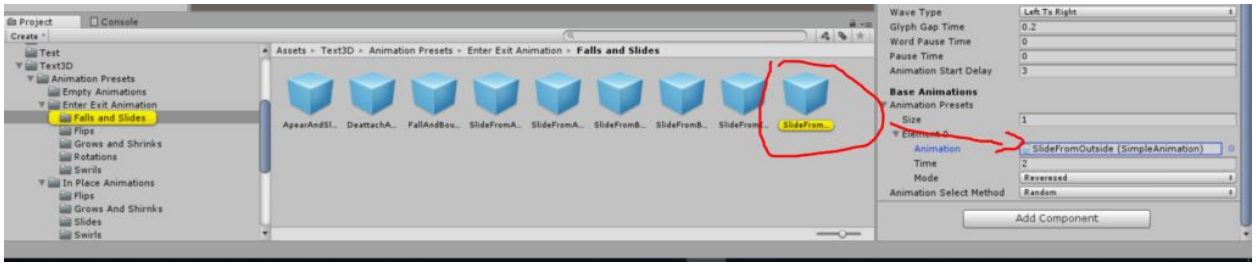
1. Add another wave animation component to the Text Element:



Enable "Animate On Start" and set "Glyph Gap Time" to 0.2.

The exit animation should start at some time margin after the first animation wave. In this example we set "animation start delay" to 3. In the previous step it was set to 2. This means the last wave starts 1 second after the first wave.

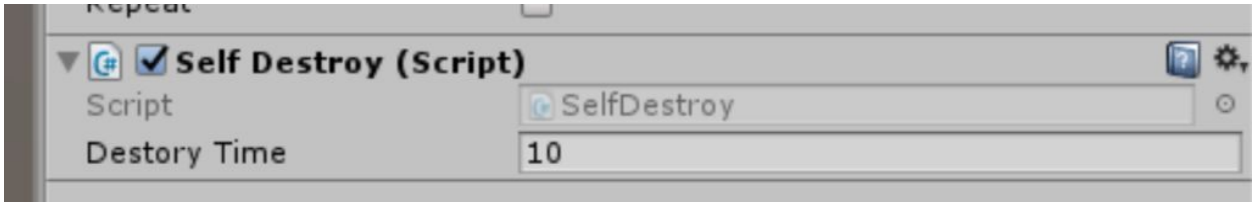
2. Drag the "SlideFromOutside" animation to the "Animation Presets" array:



The "SlideFromOutside" animation slides the glyph from the outside of the screen to the inside. This is the opposite from what we want. We can reverse the animation by setting "Mode" to reversed. This essentially turns the animation into a "SlideToOutside" animation.

Destory the animation automatically when done

An important concept in unity is not to keep unused game objects in the scene. Therefore it is important that the object self destroys after the animation is done. You can do that automatically by adding a SelfDestory Component to the container game object:



This one destroys the game object after 10 seconds , which is enough time for the animation to complete.

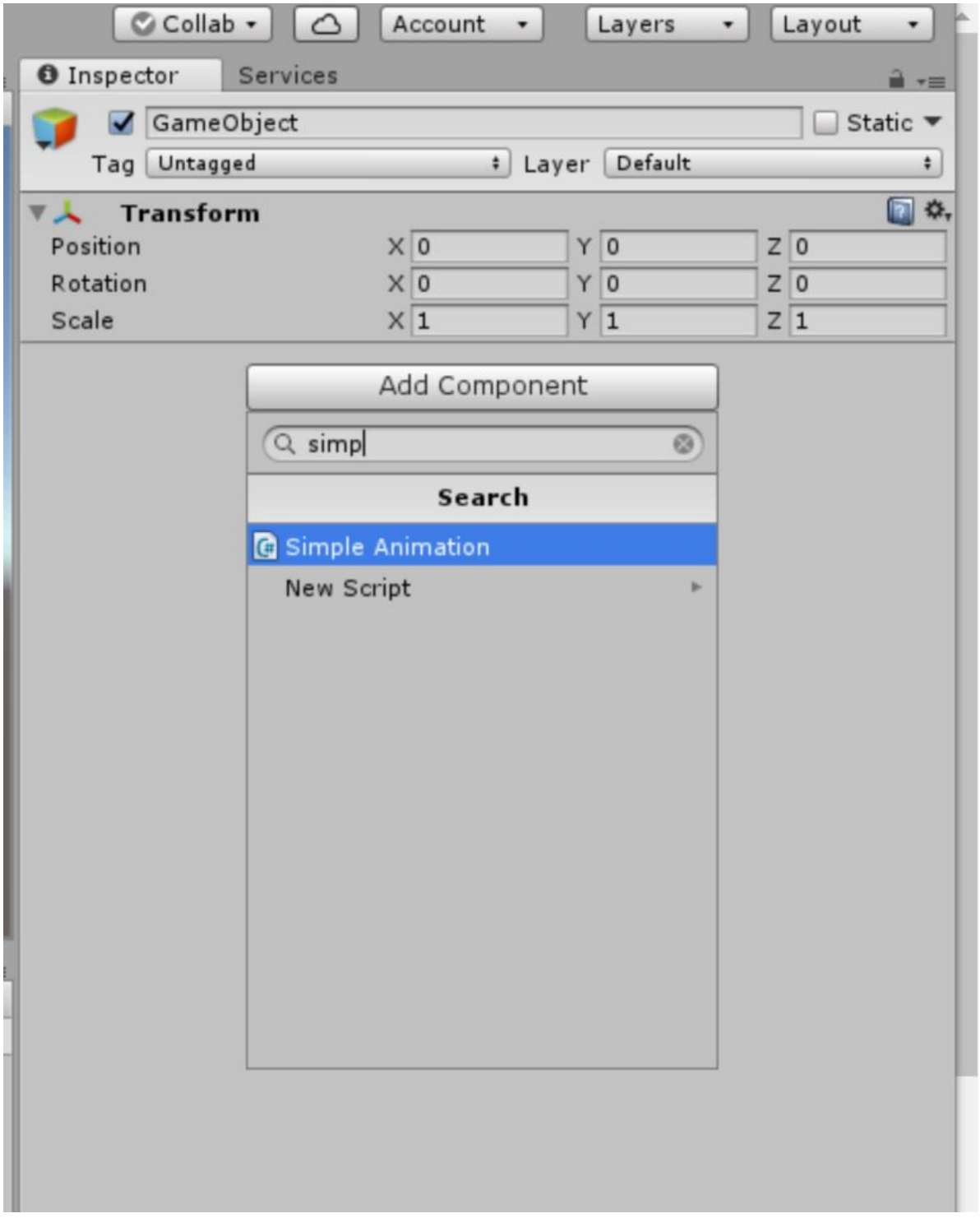
Save the animation to a prefab

The animation is now ready to use. Save it to a prefab and use it in your game

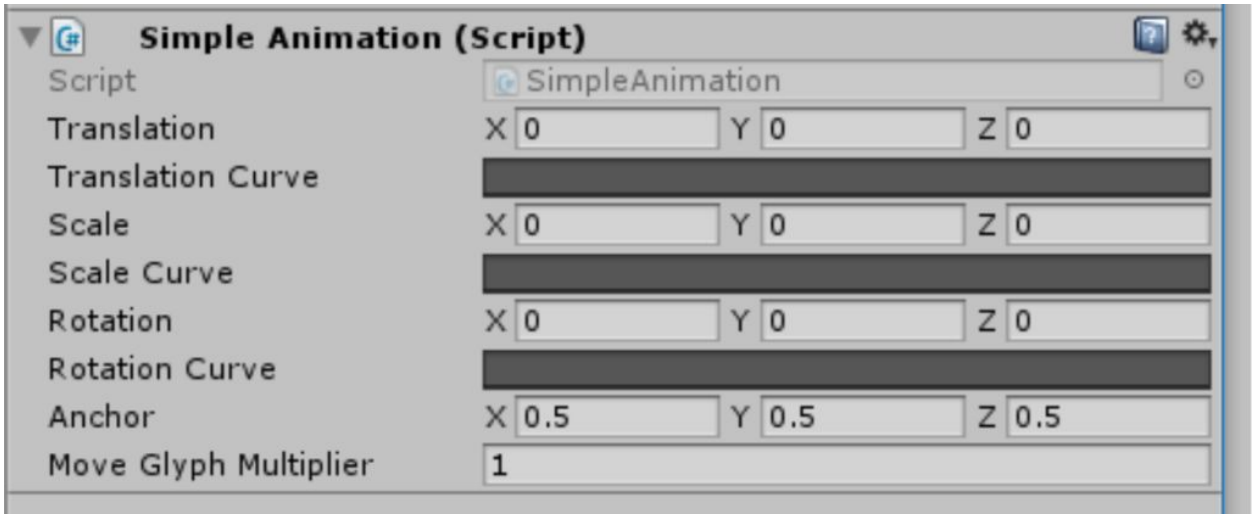
How to create your own base animation

This tutorial shows you how to create your own base animation. For the sake of this tutorial , we will recreate the Flip and Jump animation that can be found in the animation presets folder

1. Create a new prefab game object and attach a SimpleAnimation component to it:
2.

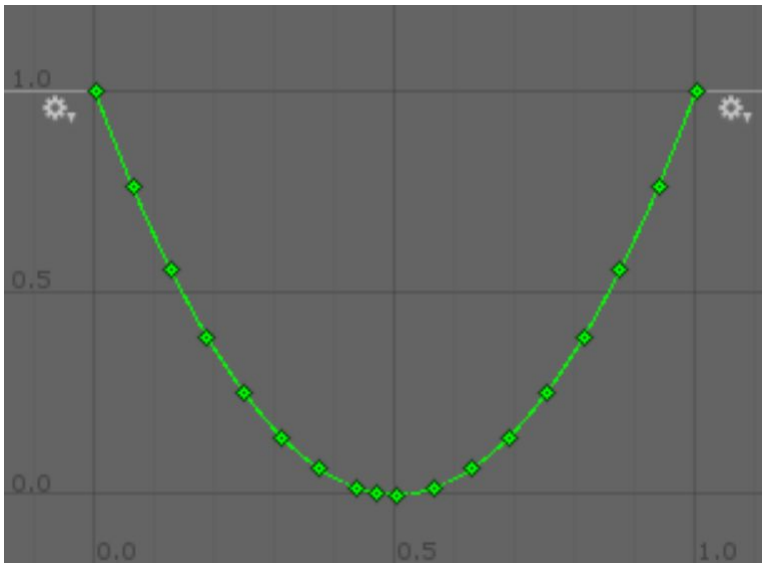


2. The Simple animation component has 3 Time lines: Translation , Scale and rotation. Each has it's own animation curve and setup:



Translation

First we will do the translation time line. For the flip animation the translation starts at (0,0,0) and peaks at (0,-0.5,0) and then goes back to (0,0,0). The timeline is between 0 and 1. when it's value is 1 it will be the original glyph position , when it's value is 0 it will be on the "Translation" property. Lets set up the translation so the glyph jumps:

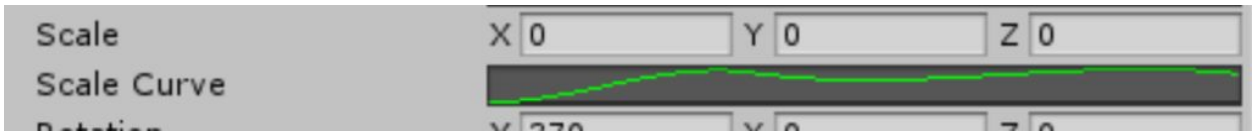


As you can see on the time line. It starts with 1 then declines to 0 and then goes back to 1. This means

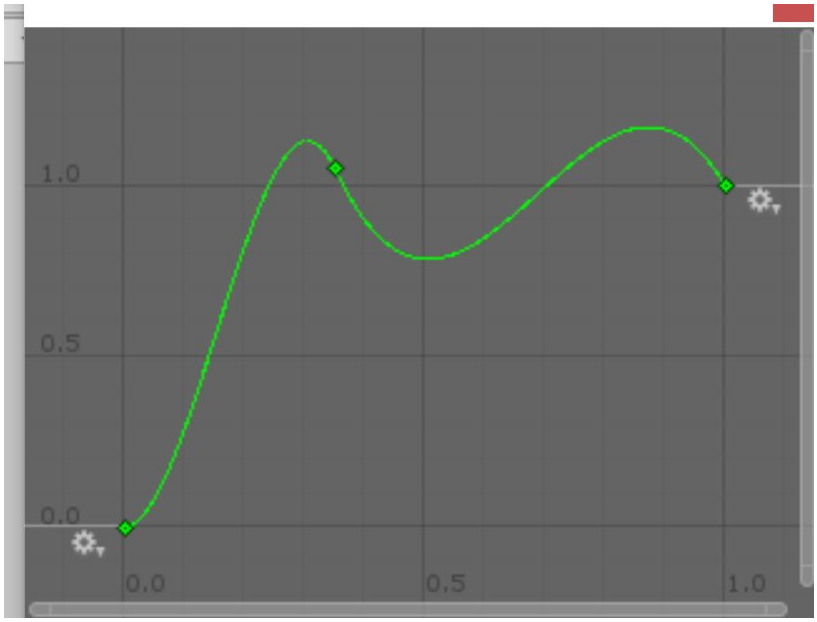
the translation starts at (0,0,0) and peaks at (0,-0.5,0) and then goes back to (0,0,0).

Scale

The scale works in a similar way to the translation. when the timeline is 1 the original scale is applied to the glyph, when the timeline is 0 the specified scale is applied.



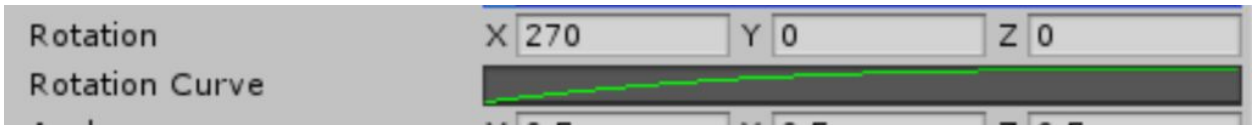
Setting the scale to (0,0,0) means the glyph starts out invisible , and then grows to (1,1,1). In this example we are setting an easing curve with an anticipation effect:



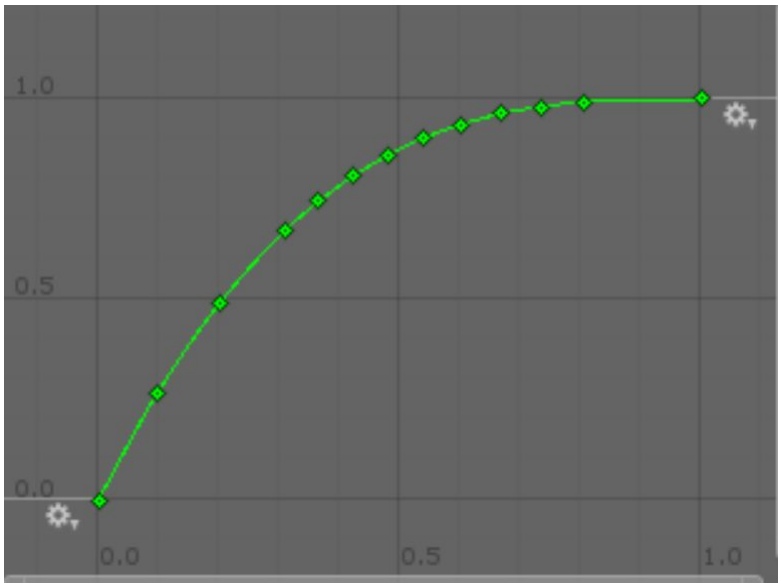
As you can see , this timeline goes beyond "1". This will scale the text more then it's original scale. this concept can be used with translation and rotation as well

Rotation

The rotation works in a similar way to scale and translation. You can set each component with a degree value:

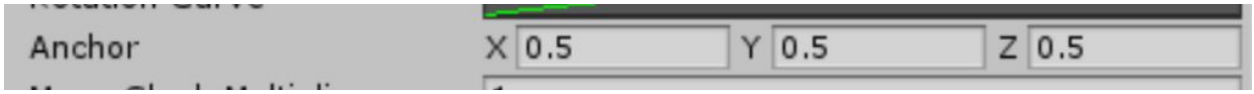


Here we are setting a 270 degree rotation on the x axis as the 0 value of the timeline. We can then set the timeline itself :



Anchor

Sometime we would like the scale and rotation to operate on anchor instead of the center of the glyph:



The anchor determines the center of scale and rotation. For the sake of this animation it is set to (0.5,0.5,0.5) , which is the center of the glyph. You can set it to (0.5,0,0.5) which is the bottom center of the glyph , or to (0,0.5,0.5) which is the left center of the glyph. each component blends along an axis of the glyph.

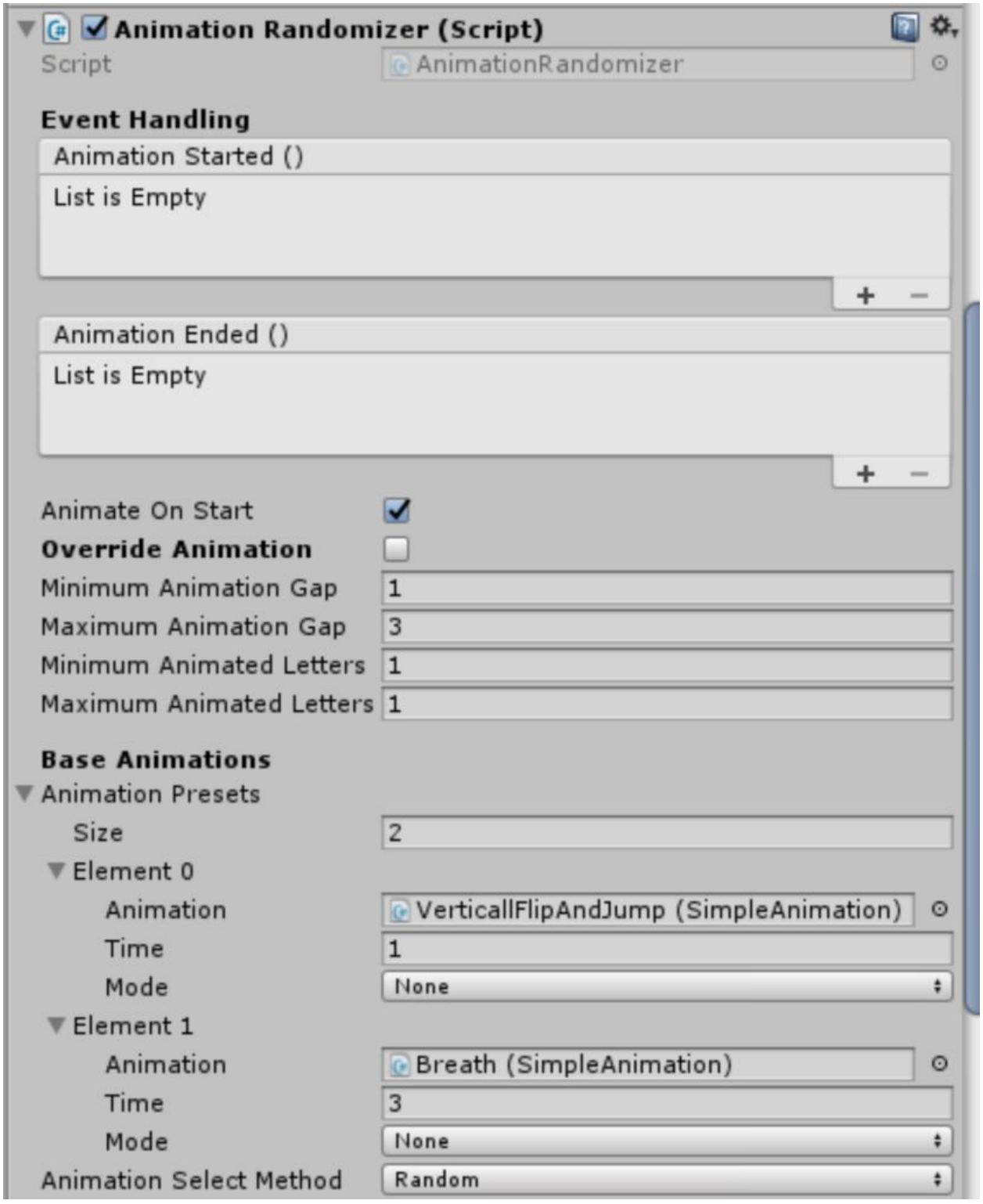
Move glyph Multiplier

some custom animation move other glyphs, You can set this multiplier to determine how fast other glyphs move. Setting it to 0 means the glyphs move immediately. For most cases you would like to set it 1 , which means the glyph move when the animation is done.

How to add random animations to text letters

A nice way to juice up a static 3D Text is to add some random animations to it. An example for that can be seen in the scene in Text3D/Tutorials/Interactive Text . If you run the scene and wait a little , you can see that glyph will jump and spin randomly from time to time. This can be done by adding an animation randomizer to your text:

1. Add an AnimationRandomizer Component to your text:



Enalbe "Animate on Start and set up your parameters similar to the image above.

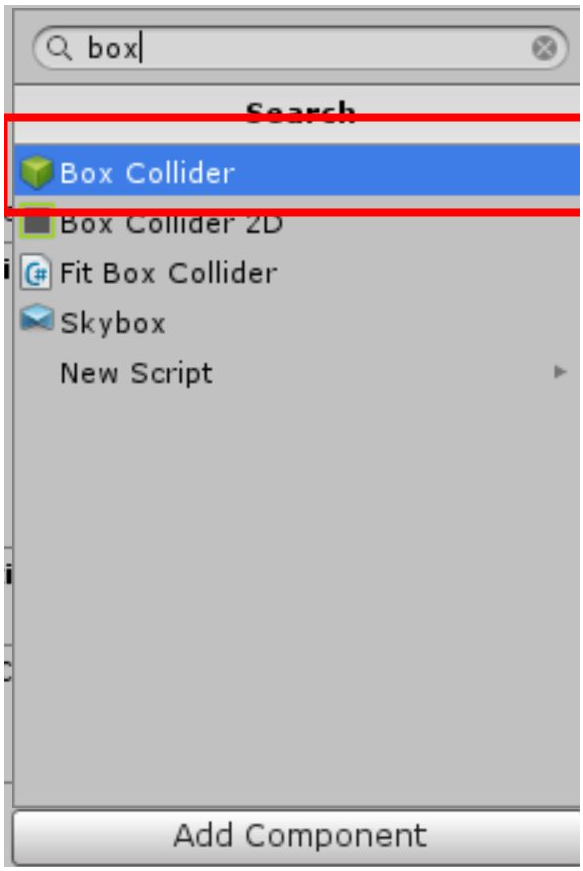
This will make the glyph jump and spin from time to time.

you can learn about the different parameters of the animation randomizer in here

How to handle user touch/click

Sometimes we would like to create an interactive 3D text object , such that can react to user clicks. We can do that by adding a box collider to the Text element and handling click events just like how we do with all unity colliders. However , text objects may change in size and it is important that we are able to match the collider to the text. You can do that by adding a FitBoxCollider component to your 3D text element:

1. Add a BoxCollider to the 3D Text Element:



2.Add a FitBoxCollider to the 3D Text Element:

Set the z size of the box collider manually.

3. now you box collider will change automatically with the text element.

4. You can handle clicks from a custom script like this :

```
  void OnMouseDown()
  {
// handle the click event
  }
```

5. Take a look at "Text3D/Tutorials/Interactive" Text to see a real use case of interactive text.

How to make your own animation scripts

This tutorial is based on the scene found in Text3D/Tutorials/Random Typing.

On many occasion we would like to create a completely custom text sequence from script. 3D Text Effect has access methods that allow you to animate text changes. This can be done in the following way:

1. Add a 3D text element to your scene

2. Create a new script and add a DynamicTextElement property to it:

```
  public DynamicTextElement TextElement; // the text element we will be operating on
// you can also add a custom animation:
AnimationEntry animation;
```

3. set up default animation for the text element. These are the animations that are used when no animation is specified in the script call.

4. You can modify the text element using the following calls:

```
 // append text:
```

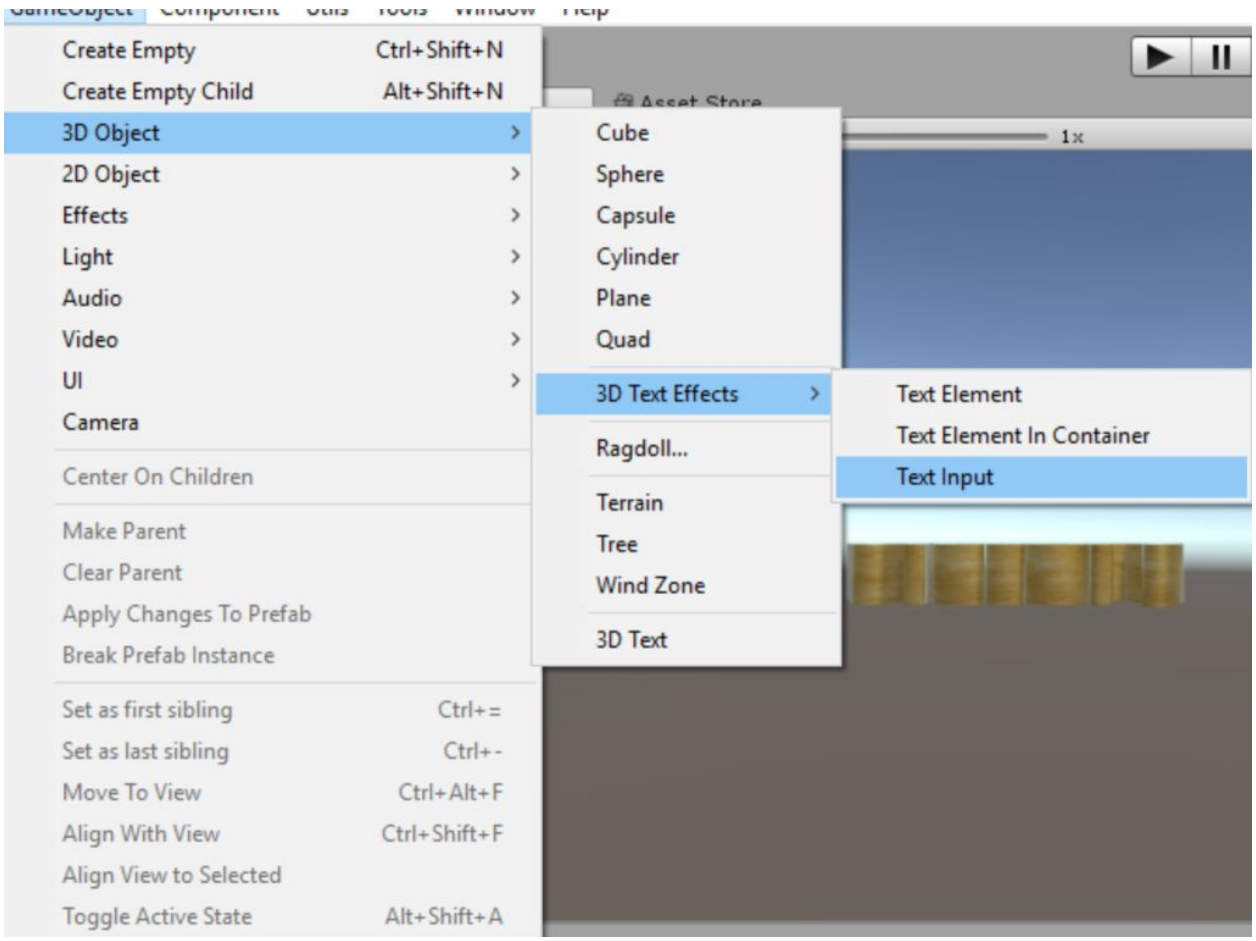
```
  TextElement.AppendText("New text"); // appends "New text" at the end of text element
  TextElement.AppendText("New text",animation); // appends a text with a custom animation
// insert text:
  TextElement.InsertText(2, "T"); // inserts the letter T in index 2
  TextElement.InsertText(2, "T",animation); // use a custom animation
// remove text
  TextElement.RemoveAt(2); // remove one char in index 2
  TextElement.RemoveAt(2,animation); // use a custom animation
  TextElement.RemoveLast(); // removes the last char
  TextElement.RemoveLast(animation); // use a custom animation
  TextElement.RemoveText(2,3); // removes 3 chars from index 2
  TextElement.RemoveText(2,3,animation); // use a custom animation
// animating glyph without changing text
TextElement.AnimateGlyph(2,animation) ;// animates the glyph at index 2 with a custom animation
// change all the text
TextElement.AnimateTextChange("New Text"); // replace all the text with "New Text"
```

5. In the Random typing tutorial(Text3D/Tutorials/Random Typing) , we use two of these methods to create random text changes at a constant time. Make sure to review it if you wish to learn more.

How to handle user input

Handling keyboard input is extremely simple:

Go to GameObject/3D Object/3D Text Effects/Text Input :



2. Place the input in your scene. It is ready for use.

3. If you wish to enable or disable input , use the AcceptInput property:

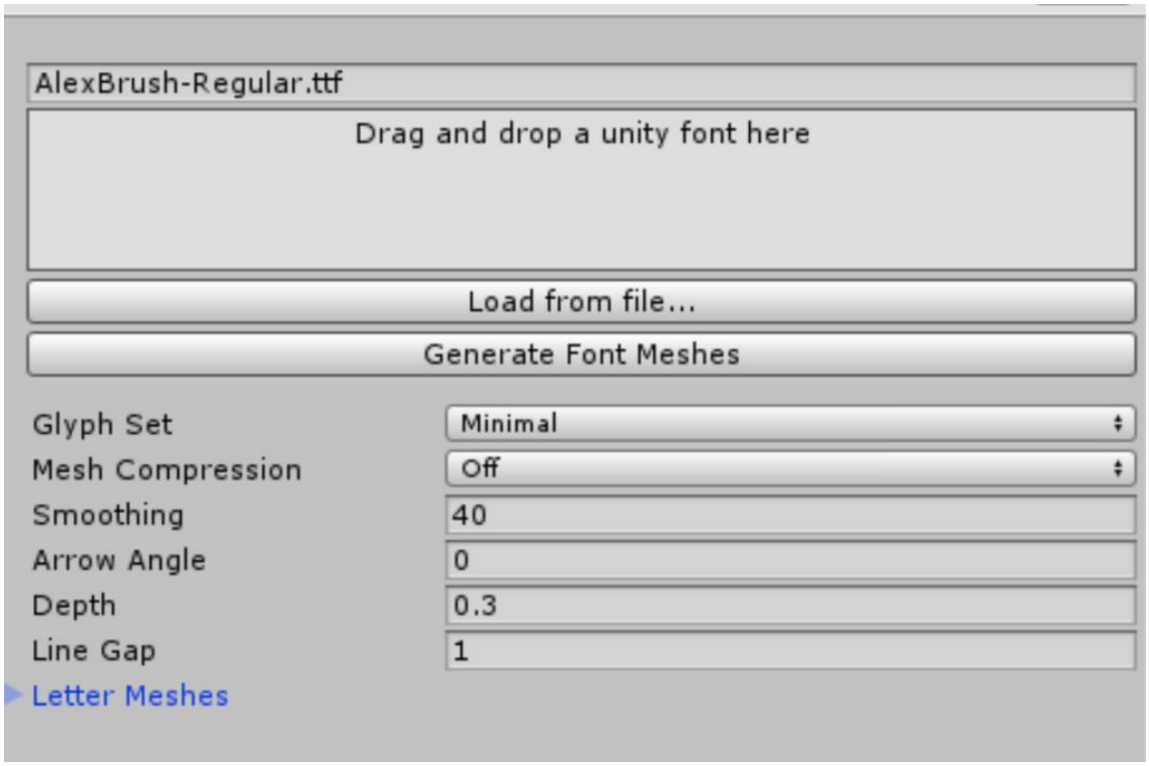
GetComponent<TextInputTutorial>.AccpetInput = true/false;

Component Reference

3d font asset reference

This is an overview of the 3D font asset settings. To learn how to add a 3D font to your project click here

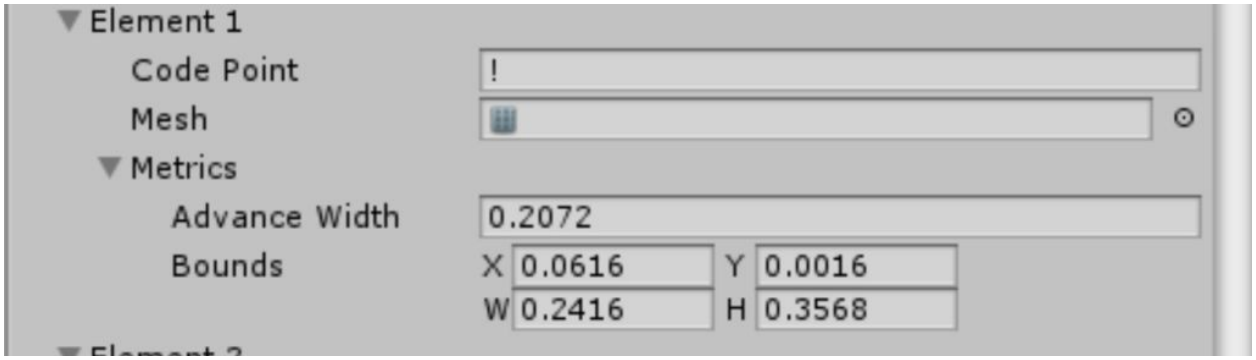
Basic settings



- **Glyph set** - Is the glyph set used when click "Generate Font Meshes". **Minimal**- contains only basic letters numbers and symbols. **ASCII -** contains all ASCII charectars. **Dynamic -** Will load all the glyphs available in the font file , use this option if you are using a non english font.
- **Mesh Compression -** enabled unity's built in mesh compression
- **Smoothing -** defines the amount of vertices per unit when creating the glyph meshes. The higher this values is , the smoother the glyphs are. The lower this value is , the lower the vertex count.
- **Arrow Angle** - the tilt angle of the a glyph faces. Use this option to make the top or bottom of the font larger.
- **Depth -** the depth of the font mesh. setting this to 0 will create a completely 2D font.
- **Line Gap-** the line gap for this font. This will usually be loaded automatically when clicking "Create Font Meshes"

The Letter meshes array

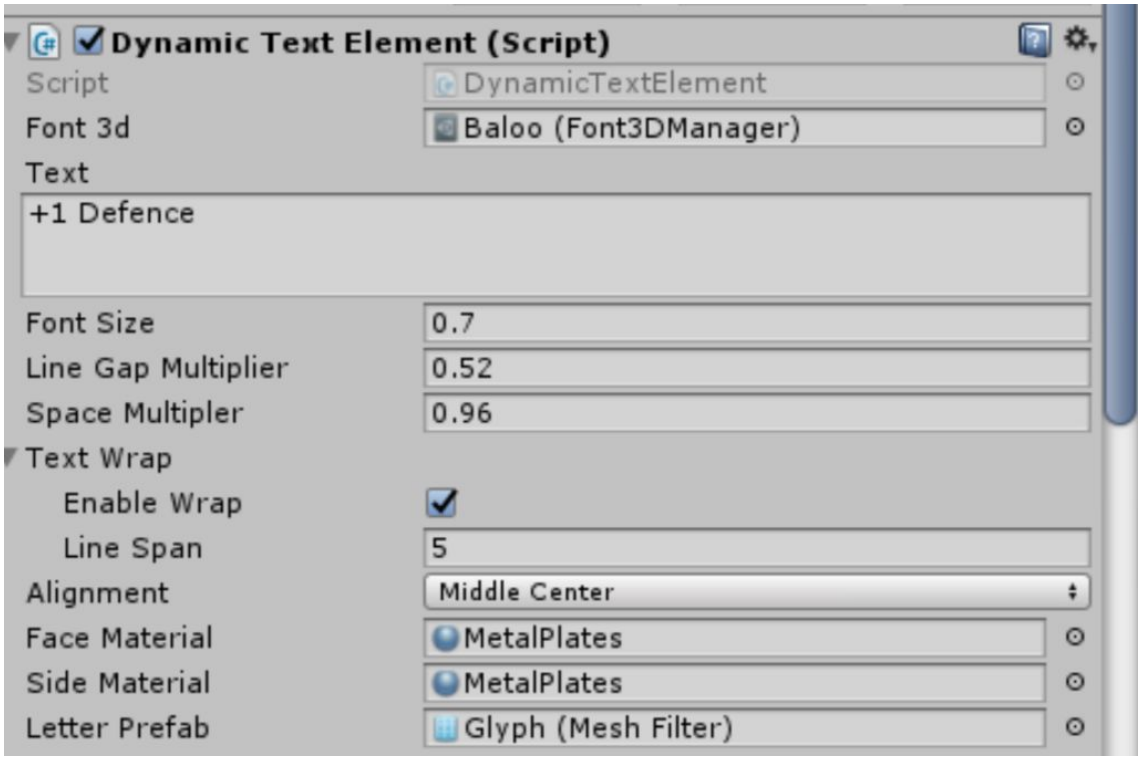
The letter meshes array contains information about each glyph in the font. In normal use it will be loaded automatically and should not be edited in most use cases.



- **Code Point -** the code point for this glyph. In the above image this is the exclamation mark.
- **Mesh -** the mesh representing the code point
- **Advance Width -** this is the distance between the current glyph and the next one in a text element.
- **Bounds -** these are the bounds of the glyph within the text line. 3D Text Effects uses this information to determine the text and line size

3D text element reference

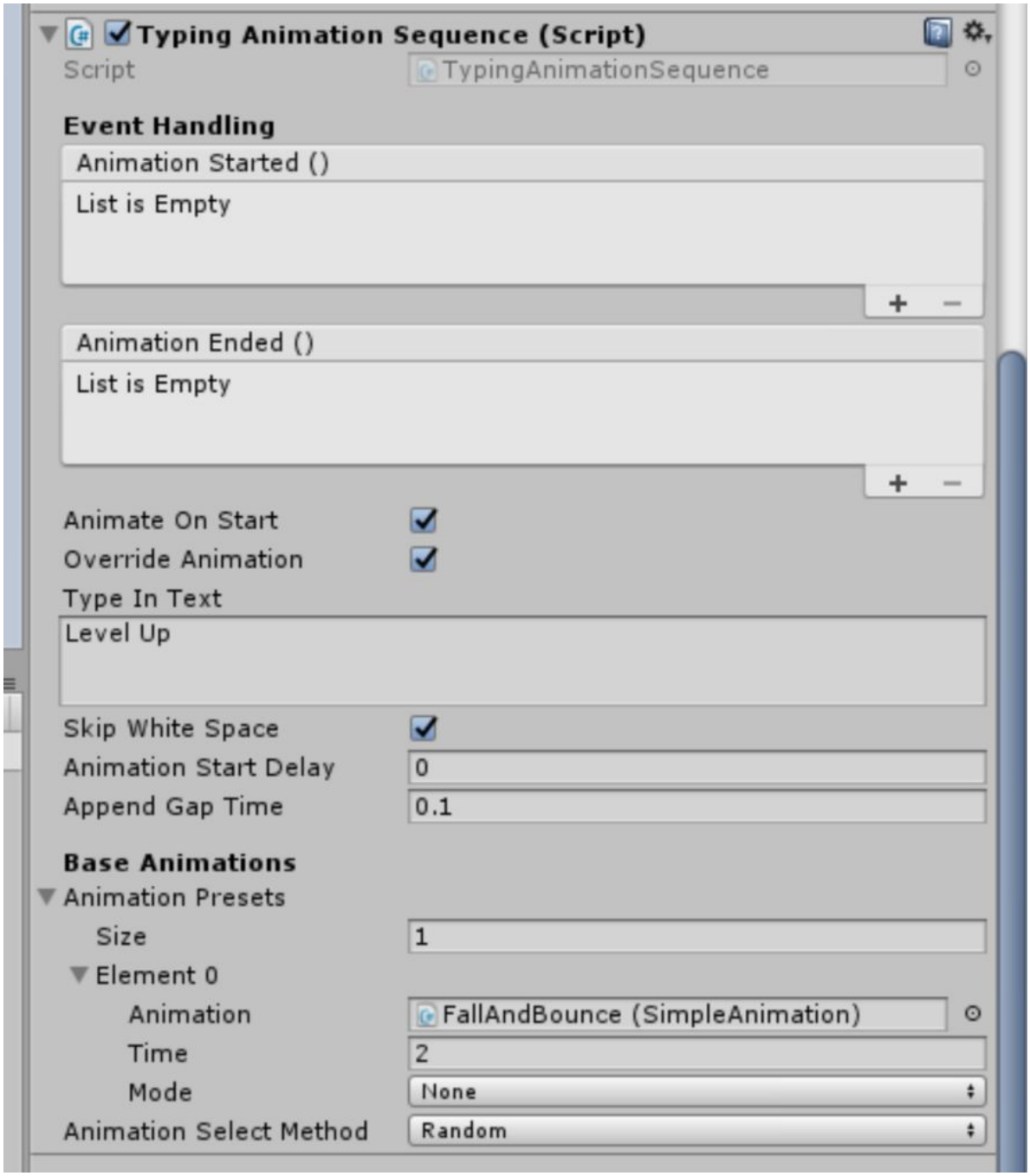
This is a reference to the properties of the 3D Text Element:



- **Font 3D -** A 3D font asset for the text element. See how to create a 3D font asset
- **Text** - the text for this text element.
- **Font Size -** the font scaling for this text element
- **Line Gap multiplier -** the gap between text rows
- **Space multiplier -** the space between text chars
- **Text Wrap -** enable "Text wrap" if you wish to bound the text to a specified "Line span".
- **Alignment - the horizontal and vertical aligment of the text element relative to the game object transform position**.
- **Face/Side Material -** these materials are assign to the face and side of the glyph objects
- **Letter Prefab -** this is the prefab object used for each glyph. It is best to use the one that comes with 3D Text Effects. You can add components to it if you wish.

TypingAnimationSequnce reference

This is a reference to the Typing Animation component:



- Animate On Start - will start the typing animation when the onStart method is called on the gameobject
- Override Animation - if the glyph is in the middle of animation , setting this to true will override the animation with a new one
- Type in Text - is the text that the animation will type in into the Text Element attached to it.
- Skip white spaces - if this is enabled , the typing will halt on white space charterers

- Animation start delay - The amount of seconds from the moment the animation is triggered until it
  actually starts.
- Append Gap Time - the gap time between the typing of each glyph (in seconds)
- Base Animations - is an array of base animations , if you have more then one base animation then set "Animation Select Method"
- Animation Select Method - **Random:** selects a random animation from the array for each glyph . **Sequential:** each glyph is matched with an array index. the first glyph to the first animation preset , the seconds glyph to the seconds animation preset and so on...
-

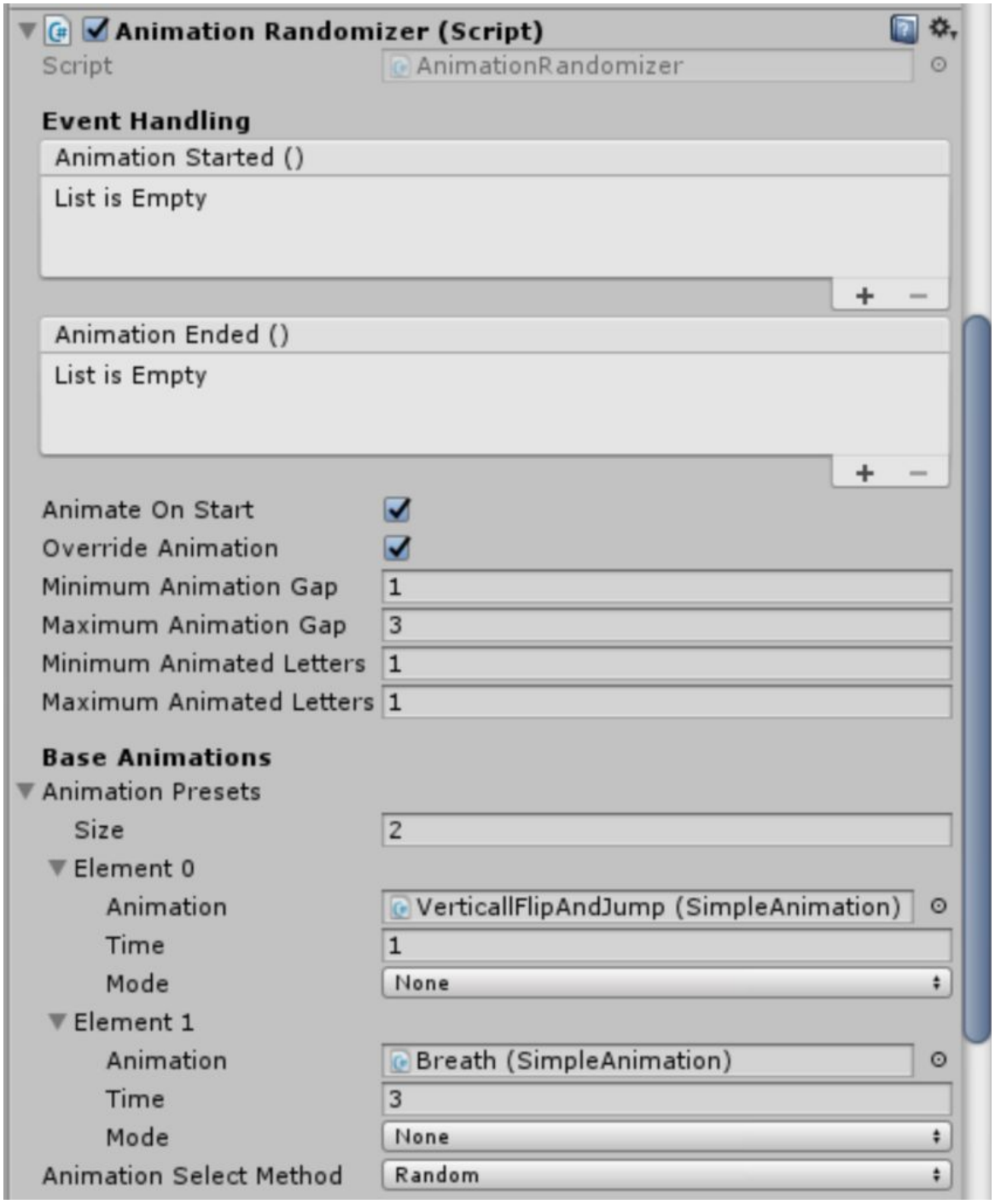
Triggering the typing animation from script

You can start and stop the typing animation using the following methods:

```
var typing = GetComponent<TypingAnimationSequence>();
typing.StartAnimation();
typing.StopAnimation();
```

AnimationRandomaizer reference

This is a reference to the AnimationRandomizer component:

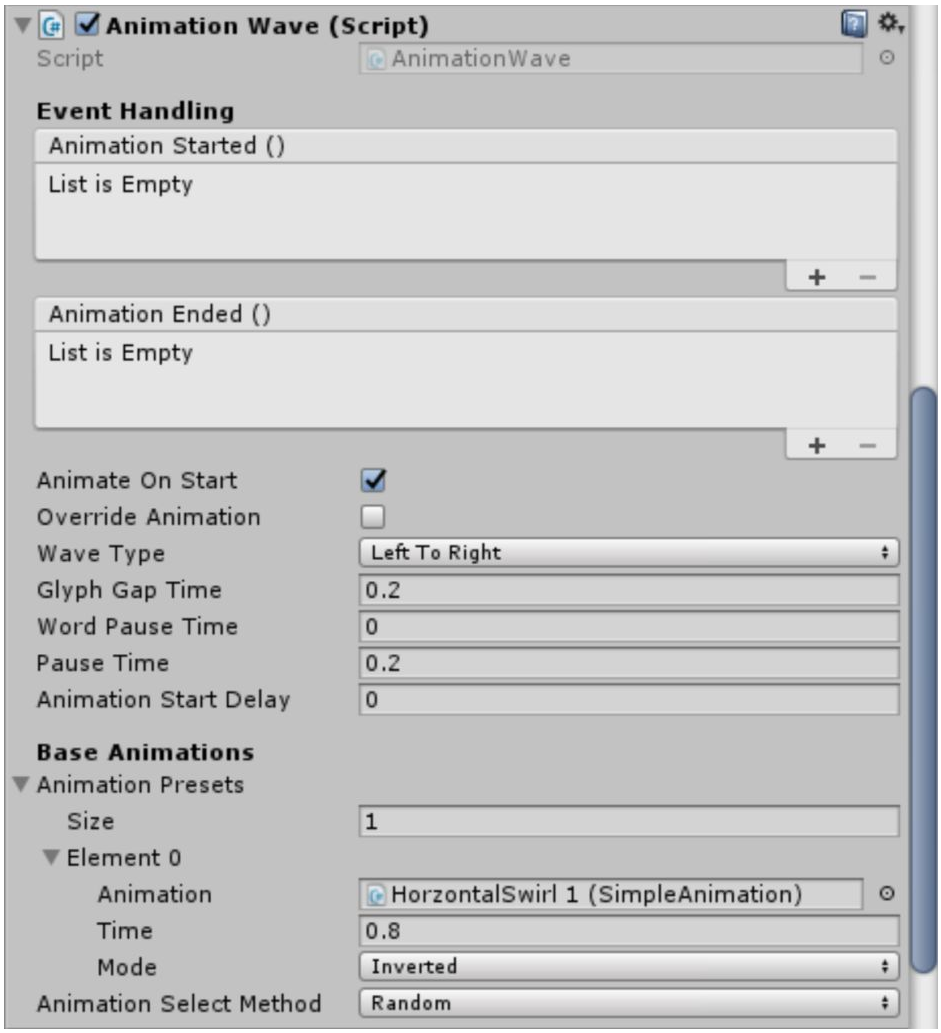


- Animate On Start - will start the typing animation when the onStart method is called on the gameobject
- Override Animation - if the glyph is in the middle of animation , setting this to true will override the animation with a new one
- Minimum/ Maximum Animation Gap - is a time range between animation bursts. The actual time is a value selected randomly between the minimum and maximum values.

- Minimum Maximum Animated Letters -is the amount of letters animated in each burst. the actual amount is selected randomly between the minimum and maximum.
- Base Animations - is an array of base animations , if you have more then one base animation then set "Animation Select Method"
- Animation Select Method - **Random:** selects a random animation from the array for each glyph . **Sequential:** each glyph is matched with an array index. the first glyph to the first animation preset , the seconds glyph to the seconds animation preset and so on...

AnimationWave reference

This is a reference to the animation wave component :



- Animate On Start - will start the typing animation when the onStart method is called on the gameobject

- Override Animation - if the glyph is in the middle of animation , setting this to true will override the animation with a new one
- Wave Type - have several presets for the wave animation: Left To Right , Right To Left, Left To Right And Back , Right To Left and Back, Repeating Left to right, Repeating right to left and Repeating back and forth
- Glyph gap time - is the time between each glyph animation in the wave
- Word pause time - is a pause delay for when the wave reaches the end of a word in the text.
- Pause Time - is a pause delay between each phase of the wave. For example if you are using
  the "Repeating Left to Right" Wave type , this is the pause between each wave repeat.
- Animation Start Delay - The amount of seconds from the moment the animation is triggered until it actually starts.
- Animation Select Method - **Random:** selects a random animation from the array for each glyph . **Sequential:** each glyph is matched with an array index. the first glyph to the first animation preset , the seconds glyph to the seconds animation preset and so on...

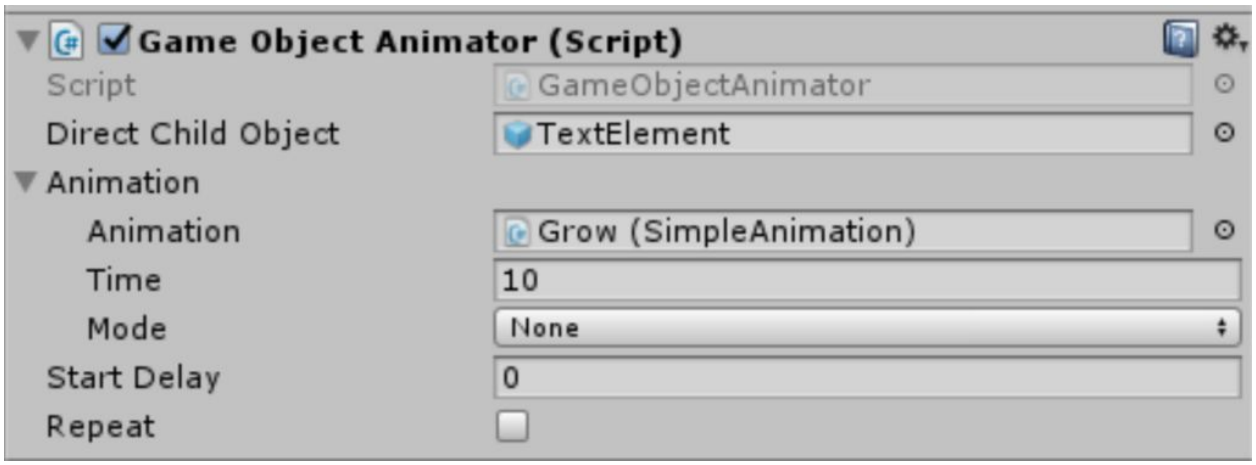
Triggering the Wave animation from script

You can start and stop the wave animation using the following methods:

```
var wave= GetComponent<AnimationWave>();
wave.StartAnimation();
wave.StopAnimation();
```

GameObjectAnimator reference

This is a reference to the GameObjectAnimator component:



- **Direct child object -** Is the game object that is being animated , it should be a direct child of the gameobject with the GameObjectAnimator component
- **Animation -** is the animation applied to the gameobject with a time and mode settings
- **Start Delay -** The amount of seconds from the moment the animation is triggered until it actually starts.
- **Reapeat -** is the animation repeating or played only once