SCT Doc

Import

If you bought it from the Unity Asset store, simple go to the SCT Asset Store page and click download and import.

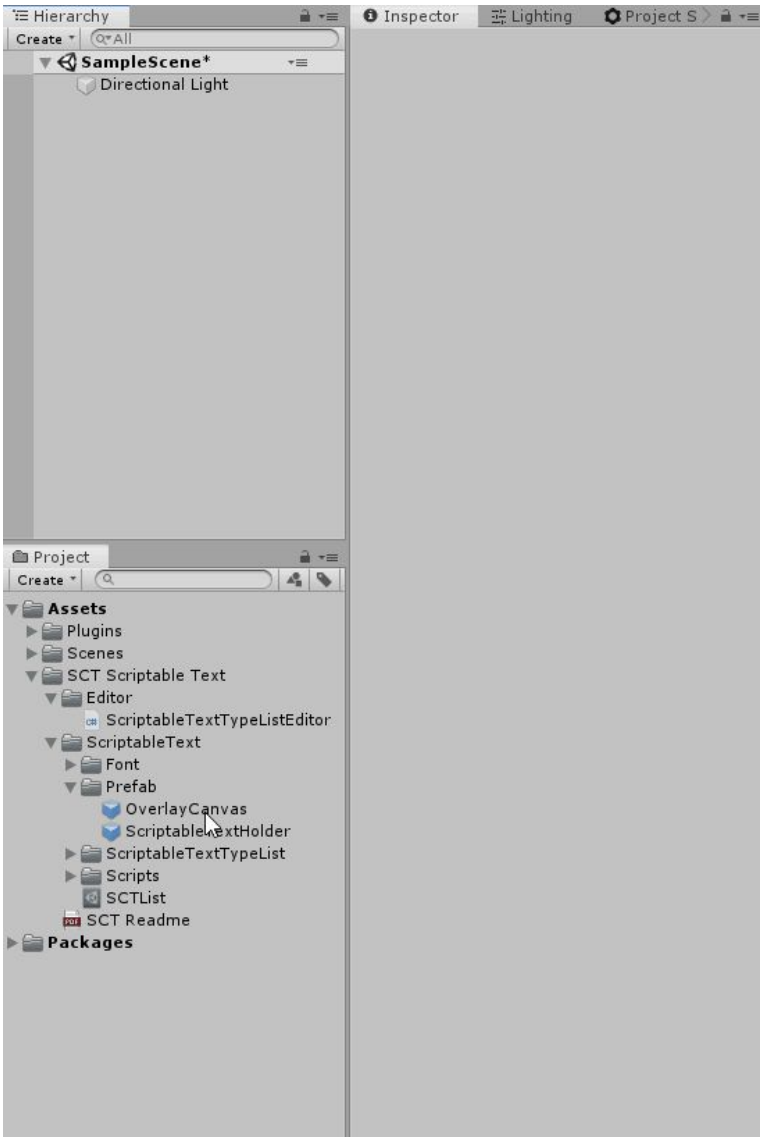
It is **good practice** to not directly import any Asset in you Project. Open a new Project and check if the Asset match your project Structure and delete unnecessary files.

Check out the Demo, you will get an Idea on how to work with SCT.

Setup

Lets setup a Test Scene, you can easily port this later to the place you want it.

- open new Scene
- drag n drop the prefab **OverlayCanvas** into the scene Hierarchy
- create a Camera or take the existing one
- assign the Camera to the **Target Camera** slot
- for testing purposes you can set the Pool Size to 15
- check out Scripting for implementation



Scripting

Implementation of the SCT is pretty easy.

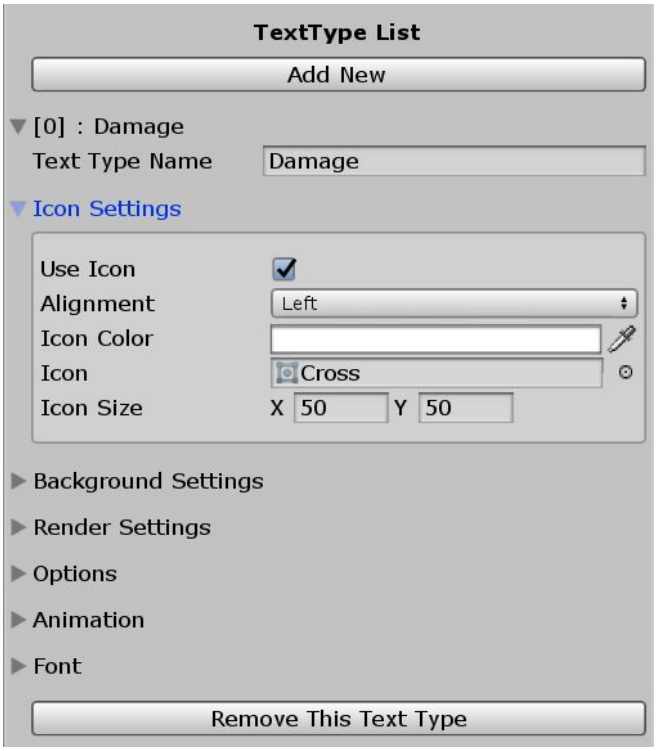
```
using UnityEngine;
using SCT;

Scripting component
public class Demo : MonoBehaviour
{
    Event function
    private void OnCollisionEnter(Collision other)
    {
        ScriptableTextDisplay.Instance.InitializeScriptableText(0,transform.position,"Ouch!");
    }
}
```

- add namespace SCT
- call `ScriptableTextDisplay.Instance.InitializeScriptableText(0,transform.position,"Ouch!");`
- check out the **overload methods**, with one you could set a icon at runtime
- InitializeScriptableText needs 3 arguments
    1. first is the Text Type position from the List
    2. position, where it should be
    3. Text to display, you can use int,float aswell, just use `.ToString()`

Example

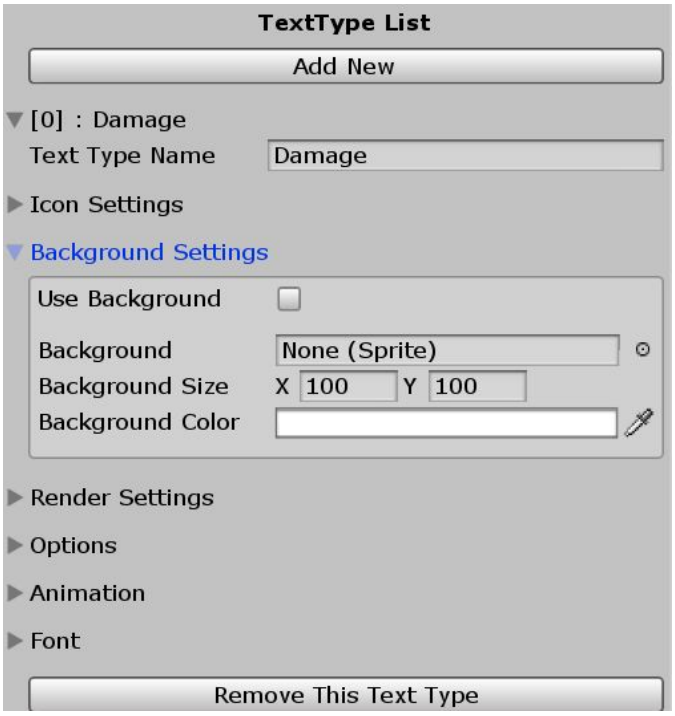
Creating a new Text Type



Lets create a simple Damage Text with Icon.

- enable “use icon”
- set the alignment to left , the icon will always be left
- choose your icon and set the size you want it to be

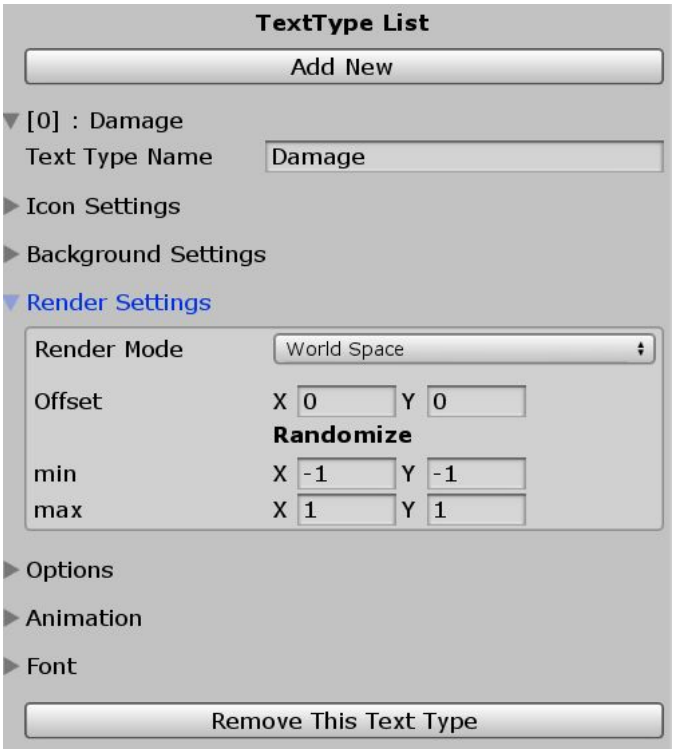
Background Settings



This time we will ignore the Background Settings.
But you could use it create a speech bubble like effect.

*Background and icons don't work well together

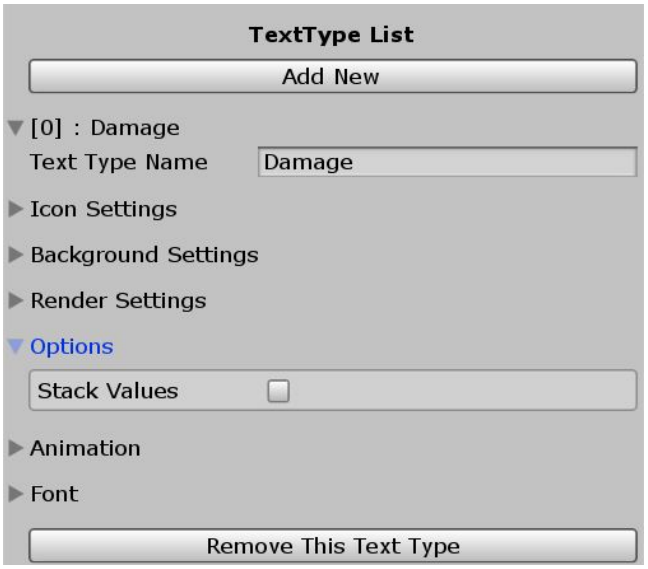
Render Settings



We will choose *World Space* for Render Mode.
This means the text will appear at the position you entered in your code.
While with Screen Space you have to define the position on Screen it appears(e.g y0.5 x0.5 would be screen center)

Offset is just an offset from start position.
Randomize will be a random offset.

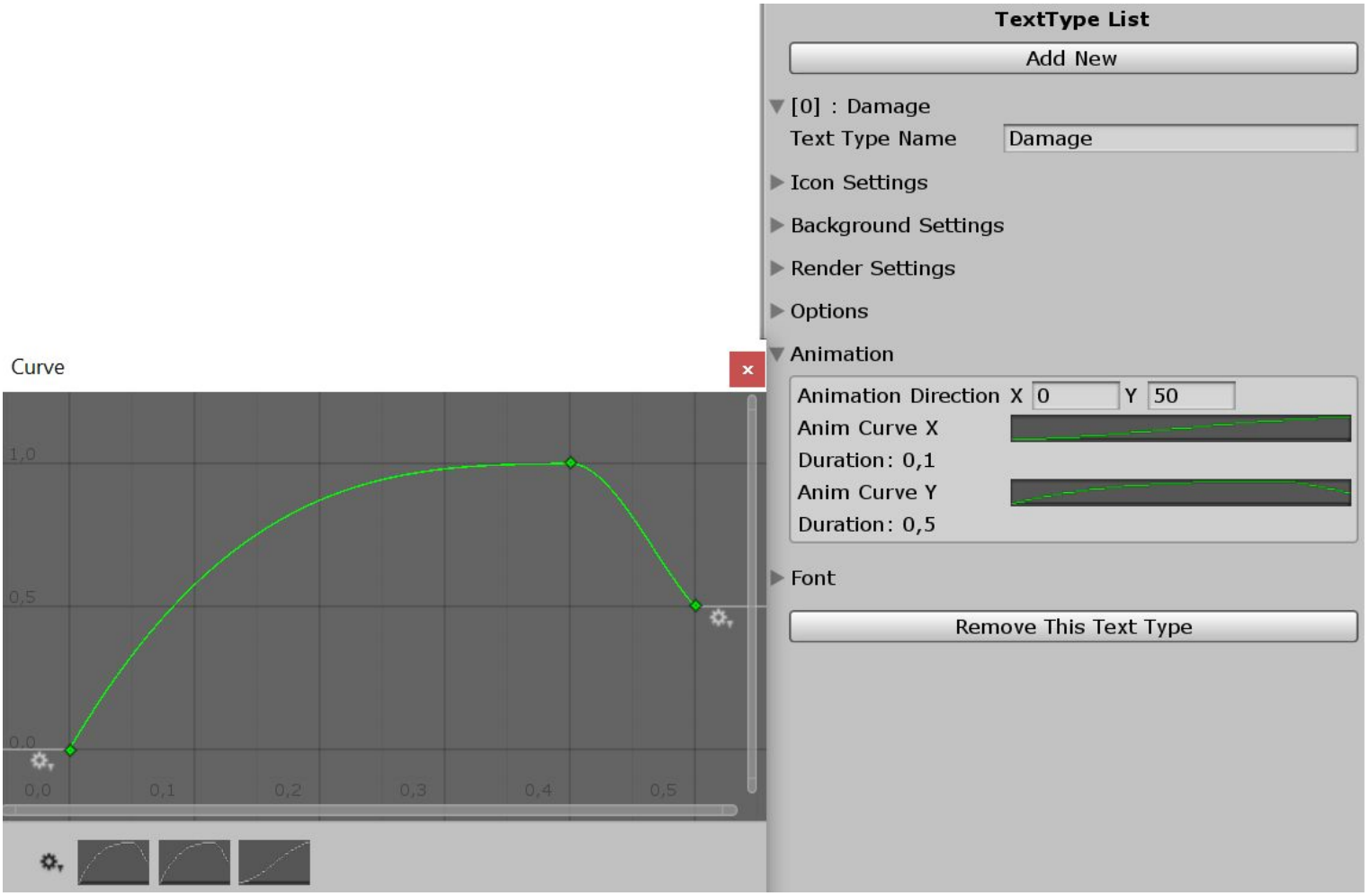
Options



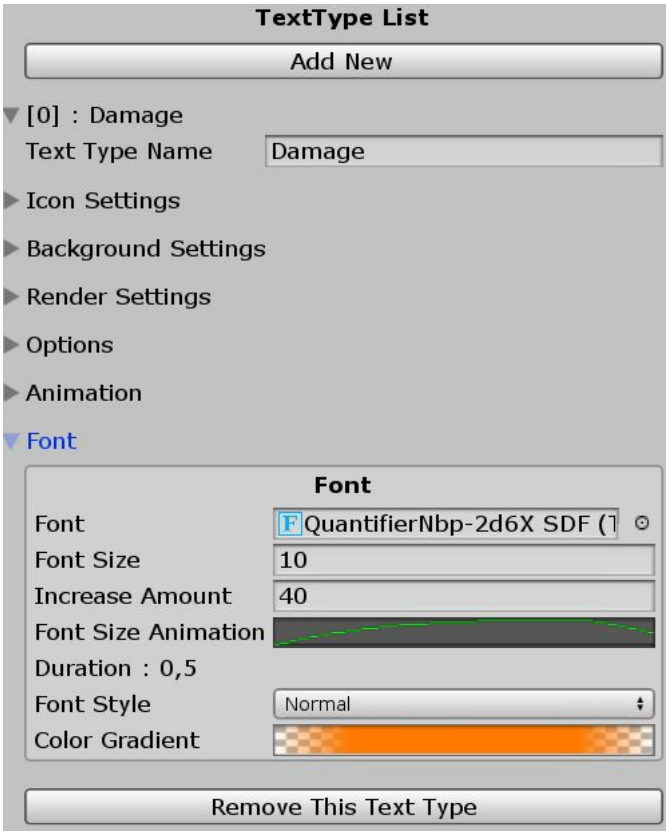
Enable Stack Values to create a Counter like Text.
Till the animation ends the numbers will be added together instead of creating a lot of single ones.

If you use Stack Values you have to call it with
Initialize Stacking Scriptable Text(text Index,position,value,key)
The key is an identifier to reuse the same Text Component again. Good choice is a ability name.

Animation



For this one i want that the text floats up and goes down a little bit. The animation should be short. You can set the Time and keys in the Animation Curve Editor.



SCT supports TextMeshPro. You can take default font for testing.
With a small Font size but a big increase amount it should look like a little Flash.
At this on the Animation Curve is the same like in Animation.

For some smoothness add alpha at the start and end to the Gradient.