
Speech Bubble
Manager

Table of Contents

Introduction

Speech Bubble Manager is a system that creates and manages speech bubbles through one line of code. It works in 2D and 3D. It will adapt the size of the speech bubble to fit the text it needs to contain. It uses object pooling to keep down garbage generation. It comes with a set of four types of speech bubbles and more can be created without too much hassle.

TL;DR; :

- Speech bubbles through one line of code
- 2D and 3D
- Adapts size to fit text
- Object pooling
- 4 types of bubbles

The system is useful not only for talking characters but also as a debugging tool; instead of printing to the debug console, let the character tell you its debugging message in person! This is especially useful for debugging AI as the character can actually talk you through his or her thinking. There is a Behaviour Designer integration package included so that speech bubbles can be easily created through an action.

As mentioned there are four types of speech bubbles included. Two of them are using 9-sliced sprites (*normal* and *serious*) while two of them are scaled to fit their content (*angry* and *thought)*. The 9-sliced work for all sizes of content, scaled images work best if the height and width are somewhat similar as they can look a bit strangely stretched otherwise. Test the demo to see this in action. To keep the ratio as close as possible introduce a line break here and there for the very best visuals when using *angry* and *thought* bubbles.

In unity 5.5 beta there are a new type of sprite, tiled. These will be included in this package in the future and will make it possible to create new, better looking, bubbles to replace the "scaled to fit"-type.

For questions, suggestions or feedback, don't hesitate to contact me at:

mike@vikingcrew.net

Instructions

This section describes how to use the Speech Bubble Manager package.

Adding Speech Bubble Manager to your game

In order to add Speech Bubble Manager to your game, drag the prefab *SpeechBubbleManager* to your gameplay scene. From one of your scripts on one of your game objects add this line of code:

```
VikingCrewTools.UI.SpeechBubbleManager.Instance.AddSpeechBubble(transform, "Hello world!");
```

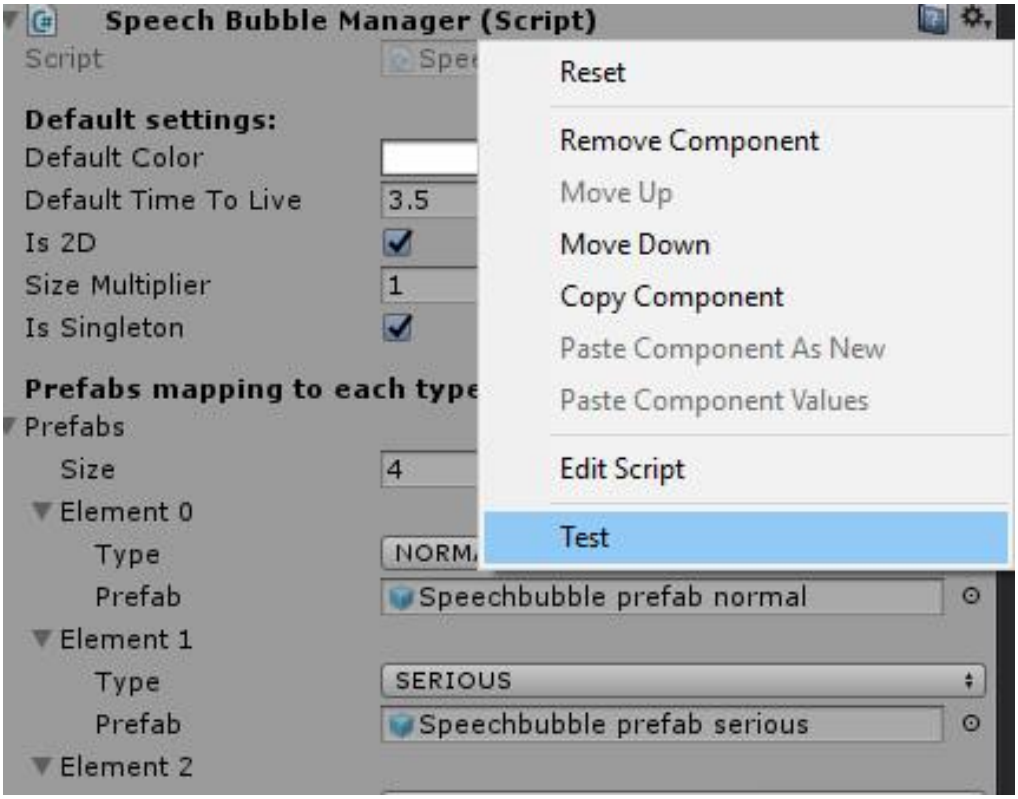
If your game is 2D then tick the ”Is 2D” box. If it is 3D untick it. This is for performance reasons, read more in *Settings*.

There are several overloads of the *AddSpeechBubble*-method to cater for your specific needs, it can be made to follow a transform or be set to an absolute position, it can have assorted color tints, various times to live, different position offsets etc. It can also be set to spawn after a certain delay. That way a character could be set to say several things after one another for a monologue or several characters have a discussion.

For more details look through the public methods in *VikingCrew/Tools/UI/SpeechBubbleManager/SpeechBubbleManager.cs*

You might want to always have the speech bubbles on top of everything else. In the demo scenes this is handled by adding another camera that draws on top of everything else. This camera only clears depth and only draws the “UI” layer. For more info see *debeani’s* answer here: https://answers.unity.com/questions/878667/world-space-canvas-on-top-of-everything.html

Once you’re done, run the scene, select “Speech Bubble Manager” in the hierarchy, click the cog-wheel for the script settings and select “Test”. A speech bubble should now appear at position (0,0,0) in your world and be displayed for 5 seconds.



Settings

See *Figure 1 Speech Bubble Manager* settings.

Default Color – The default color to tint speech bubbles with unless a color is specified in parameters.

Default Time To Live – The default time a speech bubble will be shown before it disappears. When it has less than a second left it will begin to fade out.

Is 2D – Set accordingly. If the scene is in 2D then the speech bubble will be drawn by one single canvas which is good for performance. They will be internally sorted so that the newest one is drawn on top in case two bubbles intersect.

Size Multiplier (from version 1.0.1 and forward) - Depending on the scale, camera distance or other things in your scene you may want to edit the size speech bubbles in the scene. As this can be done on a scene per scene basis using this variable it is better to use this than to edit the scale of the speech bubble prefabs. If you're still on version 1.0 you can instead change the scale of the Speech Bubble Manager game object in the current scene to achieve a similar effect.

If the scene is 3D then one world space canvas is created for each speech bubble so that their draw order is dependent on their world space positions.

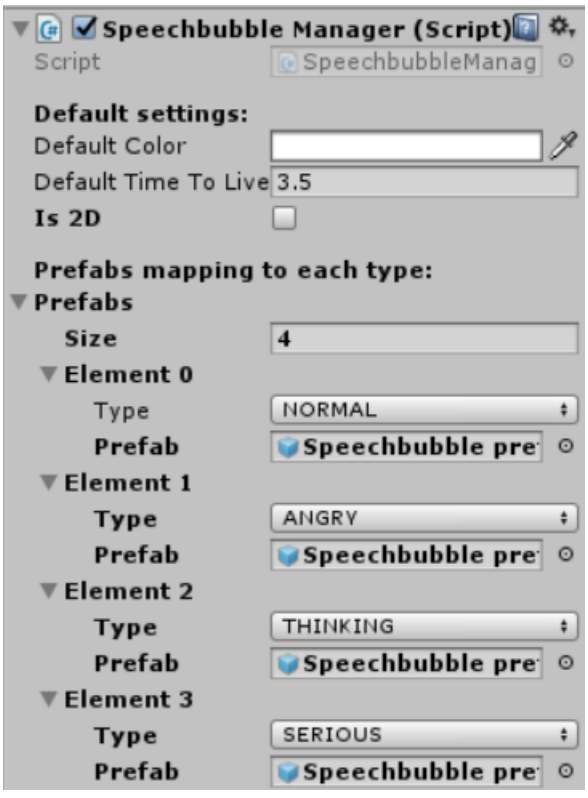


Figure 1 Speech Bubble Manager settings (Version 1.0, in 1.0.1 a new parameter was added to adjust size of speech bubbles in scene called *Size Multiplier*)

Creating new types of speech bubbles

As of Unity 5.4 there are two types of sprites that can be used for speech bubbles, 9-sliced and simple. 9-sliced works best for fitting text but it will stretch the edges of the sprite and if those edges are not uniform (e.g. jagged like the *angry* bubble) the stretching will look horrible. So, if the edges contain features you need to use simple sprites and have them be scaled to fit their content instead.

Creating 9-sliced speech bubbles

See *Figure 2 9-slice sprite setup.*

Start out by duplicating the *normal* speech bubble prefab. Edit your sprite sheet so that your bubble is exactly contained within the rectangle, blue color in the image. Next, set the *border values* (the green rectangle) to be as large as possible but still containing only those parts of the image that can be stretched without looking bad. E.g. the "arrow" of the bubble in the image would look bad if it was stretched and thus is in one of the corners and will have the same size regardless of how much content is in the bubble. Apply changes to the sprite sheet.

Next, assign the sprite to your new prefab's image child. Drag the prefab to your *Speech Bubble Manager* game object in the scene in order to make it a child. Write a message in the text component of your prefab. Adjust the *padding* settings in the *Horizontal Layout Group* to fit the text with proper borders.

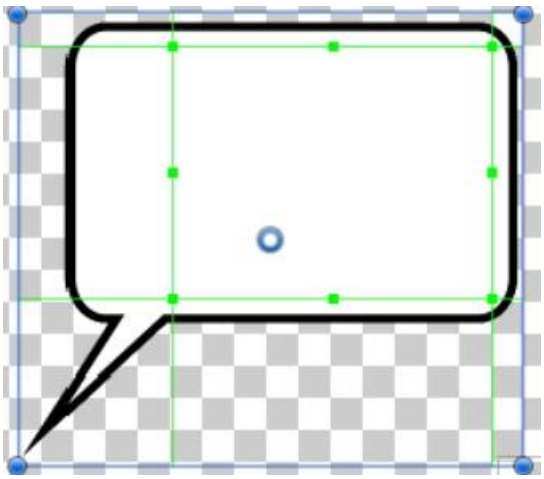


Figure 2 9-slice sprite setup

Creating scale-to-fit speech bubbles

See *Figure 3 Scale-to-fit sprite setup.*

Start out by duplicating the *angry* bubble prefab.

In this case we do not adjust the borders. Instead, consider where in the image text should fit. In the figure, this is marked with a red rectangle. The setup of this prefab will be similar to that of the 9-sliced one with one difference: Instead of a *Horizontal Layout Group* where you set the padding in pixels, this type of prefab has a *Ratio Layout Fitter* in which you set the ratio of the image to use as padding when scaling to fit the content. So, in our example with the *angry* bubble some trial and

error has led to using 23% padding on the left, 17% on the right, 20% on top and 25% on the bottom. See *Figure 4 Ratio Layout Fitter settings* for an example. You might need some trial and error to get this *just right*. Even so, you still might want to make sure text gets linebreaked so that you do not end up with too stretched out bubbles.

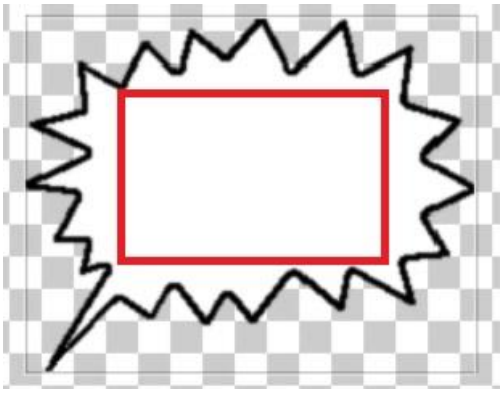


Figure 3 Scale-to-fit sprite setup

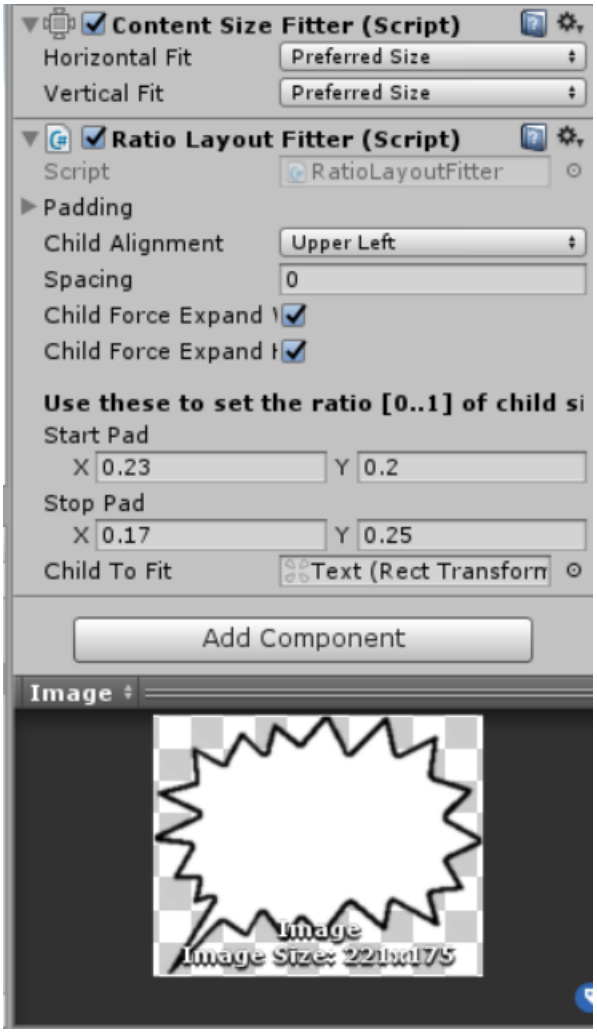


Figure 4 Ratio Layout Fitter settings

Structure of package

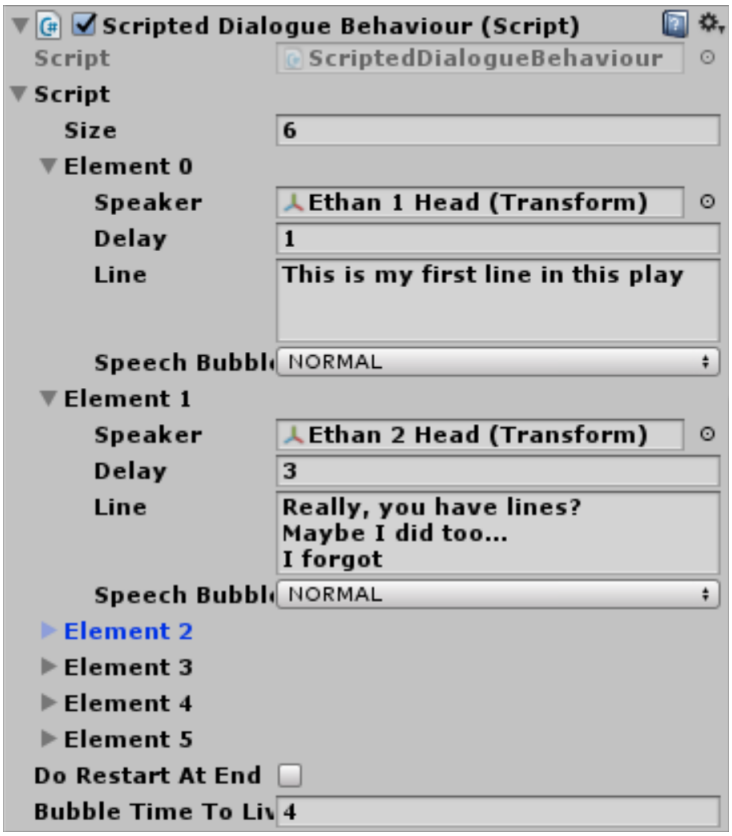
The folder *VikingCrewTools/UI/SpeechBubbleManager* contains everything you need. There is also a *Demo* folder in there that contains showcases both 2D and 3D demo scenes. This is not at all needed to be kept in your project once you feel confident in using this system. The demo scenes use some general scripts in the *VikingCrewTools* folder that you can also delete once you don't need the demo scene any more.

There is nothing that assumes any specific location of any files in the package so feel free to move everything around as you feel fit your structure.

The code is namespaced (this should not need mentioning, but, alas, many packages out there seem to not bother about this) using *VikingCrew.Tools.UI* namespace for core functionality like the Speech Bubble Manager and *VikingCrewDevelopment* for demo-specific scripts.

3d speech bubbles scripted dialogue

As several people have asked for it I have added a simple scene where the characters follow a predetermined dialogue. The script is attached to the Speech Bubble Manager object in that scene. This can be set in the editor as follows:



Each element can be set to *Speaker*, the transform that the speech bubble should follow, *Delay*, the delay from the script was started until that particular line in the dialogue should be spoken, and *Line*, the text that should be spoken. Although this is a fairly trivial script it might have its uses in cutscenes or the like.

Integrations

If you think an integration should be added let me know.

TextMesh Pro

All that is needed on your part is to use the prefab for the manager; SpeechBubble Manager TMP instead of the ordinary one and all bubbles will use TMP instead of unity UI text.

TMP has been integrated in the system by inheriting the SpeechBubbleBase into SpeechBubbleTmp. Then the prefabs for each bubble type has been created with a TMP text component instead of a normal old Unity UI text component. A similar approach could be used if you need another asset for UI components.

Behaviour Designer

In order to add actions in BD for speaking, run the unity package located in *Assets\Viking Crew Tools\UI\SpeechBubbleManager\integrations\Behaviour designer integration.* This will add an action called Speak to BD. This is very good to use for debugging at runtime as well.

FAQ

Q: My scene becomes cluttered by bubbles; can I clear it?

A: Yes, you can instantly clear it with a call to `SpeechBubbleManager.Clear()` if needed. This might be a bit drastic though, as you might just want to make sure every character has at the most one bubble each.

What you can do in this case is keep track of the reference to the speech bubble returned by the call to `SpeechBubbleManager.AddSpeechBubble` in your calling script and store its Iteration value. When adding a new bubble, check if the iteration value is the same, if it is that means it is the same bubble being shown. In that case you can make a call to that specific bubble with the method:

```
SpeechBubbleBehaviour.UpdateText(string text, float newTimeToLive)
```

to just update that one instead of creating a new one. This makes the use a bit more cumbersome so only do this if you think it is a problem in the first place.

Q: I want my Orcs to have one set of speech bubbles and my Elves to have another. I also plan to add Dwarves, Gnomes, Hobgoblins etc etc. What to do?

A: By default, the Speech Bubble Manager acts as a singleton and is called through the static variable

```
SpeechBubbleManager.Instance
```

You can, however, choose to have as many managers as you want in a scene by unchecking the box "Is Singleton" in the Speech Bubble Manager object in the inspector. This will allow you to have several managers in the scene. These can be set up with different bubbles in their *Prefab mappings to each type*. So, in your scenario you would have one manager for the Orcs, one for the Elves and so on. Note that in that case each team would need to have a reference to their specific manager as you could obviously not use the static instance variable any more.