

Flow Controller v1.4 for Unity

Flow control for game objects and particles.

(C) Kupio Ltd.

Email: support@kupio.com

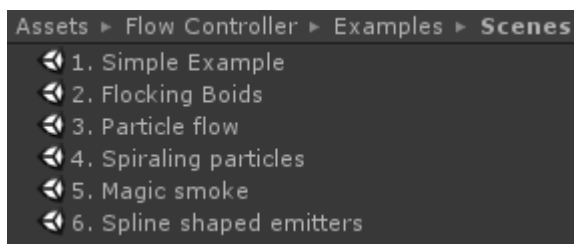
1. Introduction

Flow controller is a method of creating pre-baked vector fields, controlled by splines that can influence the movement of particles and game objects.

1.1 Asset files

The asset comes with a number of folders.

The Examples folder has some simple examples of the asset in use. This folder can be removed entirely and will not affect your own code. Look to the Scenes folder for examples.



In scripts, the two components you will most likely be interested in using are

Scripts/FlowControl/FlowControlRegion.cs

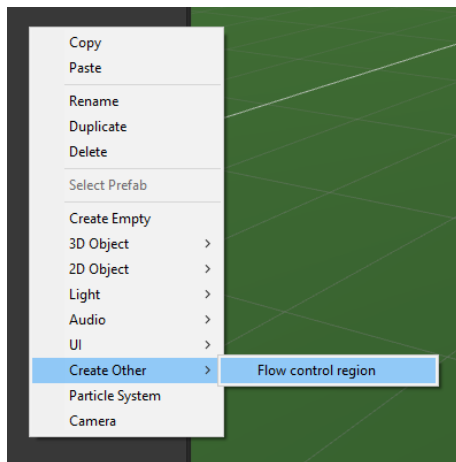
Scripts/FlowControl/ParticleFlowController.cs

These are explained in this document and illustrated in the example scenes.

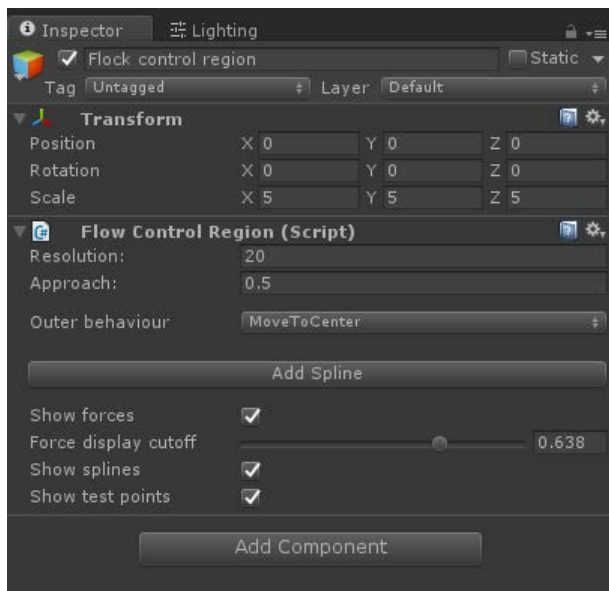
2. Documentation

2.1 Creating a region

From the Unity editor - GameObject -> Create other -> Flow control region



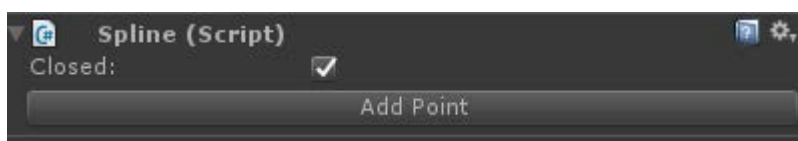
With the region selected, you can add splines to the region (Add spline in the inspector).



You can have as many splines as you like since the flow information is baked in the editor and therefore always has a fixed runtime cost.

Note that flow regions are axis aligned boxes, and are defined by their position and scale values. Rotation has no effect.

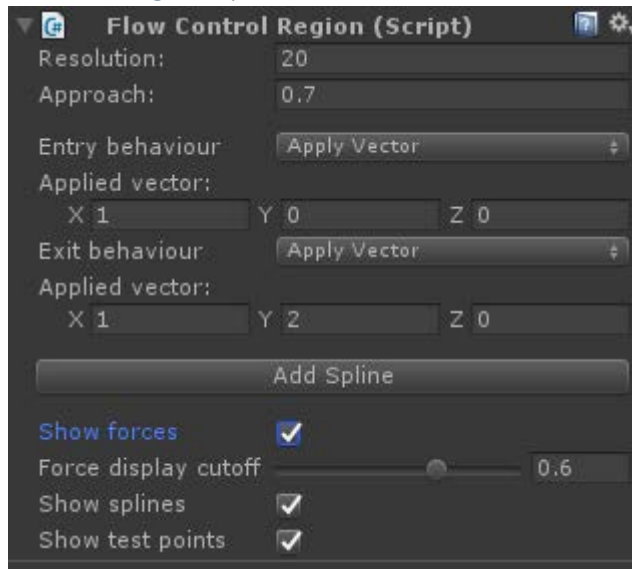
Splines are edited as game objects in the hierarchy, with an object for each point. Points can be added by selecting the spline and clicking 'Add point' in the inspector.



The same panel has a checkbox to mark if the spline is a closed loop or not.

Spline point handles are marked as yellow circles in the editor and can be dragged into position to edit the curve.

2.2 Flow region options



Resolution

The flow region is a baked 3D grid which influences particles or game objects within it. The higher the resolution, the better the quality. Setting a lower resolution will help when editing splines as it speeds up the generation of the grid and reduces editor lag.

The asset uses multi-threaded code to make this process as quick as possible.

Approach

The approach value controls how sharply objects that are far from a curve will approach that curve. A high value (1) will approach gently and objects will align with the curve before approaching. A low value will result in a more perpendicular approach to the nearest curve.

Entry / Exit behaviour

When sampling the grid, this defines the behaviour when sampling a point outside the grid. It can either apply some fixed vector (applied vector), or move away from/towards the centre of the grid. There are two options – one for particles that have yet to enter the control region (entry behaviour) and one for particles that have left the control area (exit behaviour).

Set the exit behaviour to 'maintain direction' to stop influencing the particle after it has left the region.

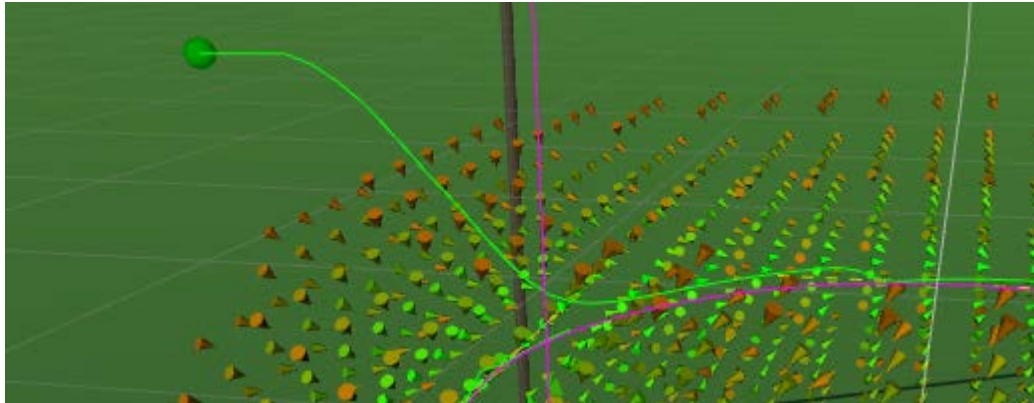
Set it to 'Move to Transform' to move towards some GameObject in the scene.

2.3 Testing points

Test points can be added in the editor to test the behaviour of the flow.

Create a game object as a child of the flow control region, and add the flow control testing point script.

A green dot will appear in the editor with a long leading tail that will project the movement of an object in the flow region.



By moving the green dot around, you can test the movement of a particle for different starting positions.

The behaviour of the projection can be controlled by the test point's inspector properties.

3. Particles

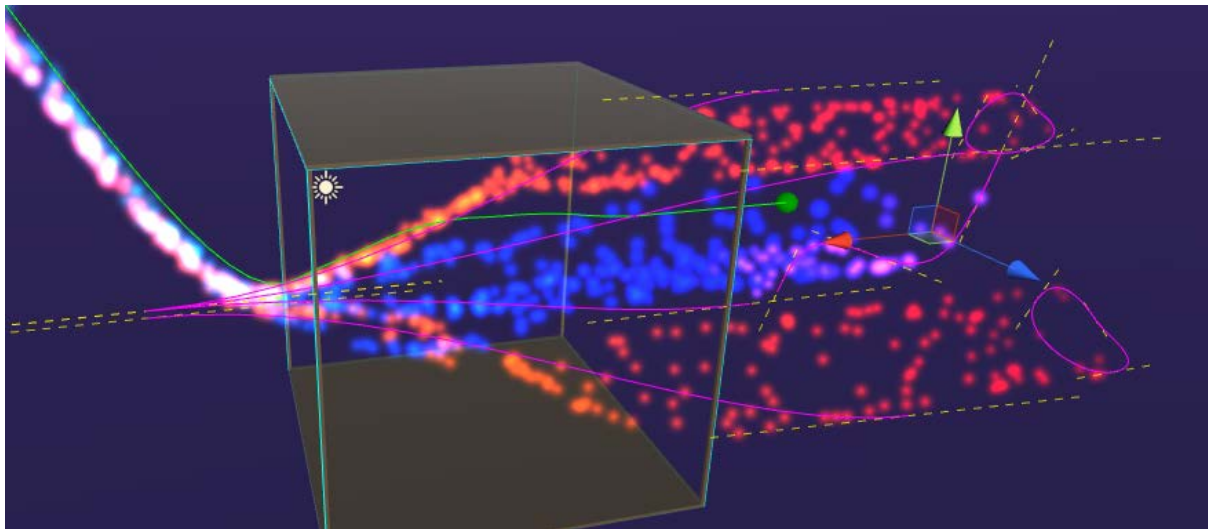
Create a particle system and add the Particle Flow Controller component. Any particles that flow through the flow control region will be influenced by the flow.



In the inspector for the component, you need to set the flow region. You can also add inertia for the particles and a particle speed. The higher the inertia, the less the flow control will have an effect on particles' directions.

4. Using splines as emitters

It's possible to use spline curves to define a particle emitter shape. This is demonstrated in example scene 6.



To do this, simply add a SplineEmitter component to a particle system



The component will show an add spline button. You can add as many splines as you wish, and they will act together as one emitter shape.

5. Scripting your own effects

The interface to the flow control region is very simple. In your code, get a reference to the region using a property, or a GetComponent call if necessary:

```
FlowControlRegion fcr = GetComponent<FlowControlRegion>();
```

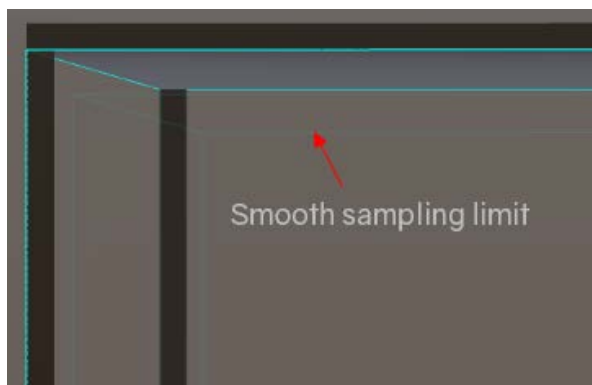
Once you have the region, you can simply sample any world position using a call to `SampleWorldPos`, e.g.

```
Vector3 force = fcr.SampleWorldCoord(position);
```

The returned value is a normalized vector that can be used in any way you wish, for example as the steering component in boids simulations. Note that this sample is quantized to the resolution of the vector field, so it may be desirable to smooth the movement of your particles with inertia.

To avoid this, use `SampleWorldCoordSmooth` instead. This will interpolate the samples and give a much higher quality result, but requires a little more CPU time.

One other side-effect of smooth sampling, is that part of the region around the edge will become out of bounds. This is dependent on the resolution of your field, but it will be indicated by a faint box in the editor.



6. Support

If you have any questions, please ask support@kupio.com