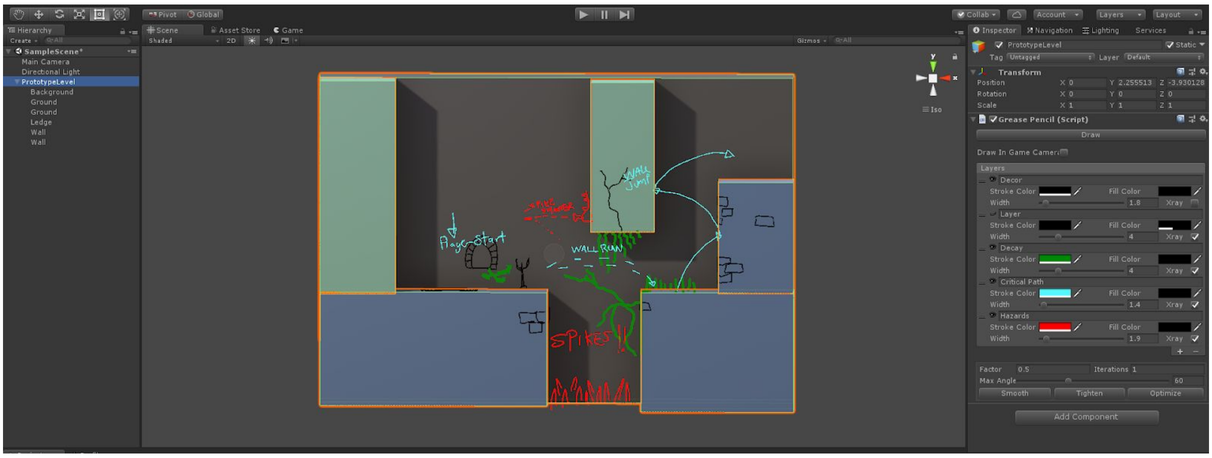
Wahooney's Toolbox

Grease Pencil 1.1.1

Grease Pencil allows you to add hand-written/-drawn annotations to objects in your scenes, including prefabs. Map out critical paths, notes for level dressing, play testing notes, and much much more!



SampleScene*
- Main Camera
- Directional Light
- PrototypeLevel
  - Background
  - Ground
  - Ground
  - Ledge
  - Wall
  - Wall

☑ PrototypeLevel ☑ Static

Tag Untagged Layer Default

Transform

Position	X 0	Y 2.255513	Z -3.930128
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ Grease Pencil (Script)

Draw

Draw In Game Camer□

Layers

Layer	Stroke Color	Fill Color	Width	Xray
Decor	■	■	1.8	□
Layer	■	■	4	☑
Decay	🟩	■	4	☑
Critical Path	🟦	■	1.4	☑
Hazards	🟥	■	1.9	☑

Factor 0.5 Iterations 1

Max Angle 60

Smooth Tighten Optimize

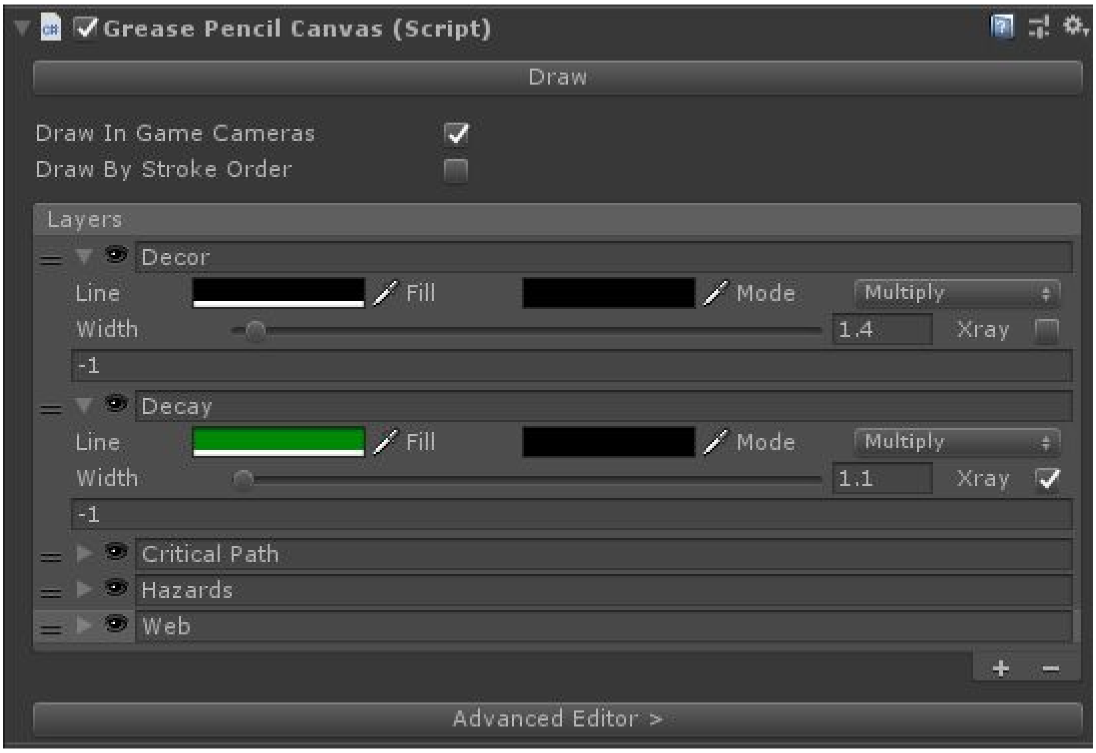
Add Component

Contents

Basic Usage

1. Add a Grease Pencil component to your selected object.
2. Pressing **Draw** will toggle drawing, you can draw in any scene view with **LMB**.
3. **Draw Mode** indicates the virtual canvas you will be drawing on:
    a. **Screen** will draw on a canvas that has its origin on the selected game object, with the canvas facing the camera.
    b. **Surface** will draw along the surface of any meshes in the scene
    c. **Planar** lets you define a plane using 3 points, Ctrl-Clicking will project a point of the triangle on the meshes in the scene in sequence.
4. Holding **Shift + LMB** will hard erase points, erasing the middle of a stroke will split it in two.
5. Holding **Shift + Ctrl + LMB** will soft erase points, erasing the middle of a stroke will keep it whole.
6. **Draw In Game Cameras** will draw strokes in your game views.

Layers



☑ Grease Pencil Canvas (Script)

Draw

Draw In Game Cameras ☑
Draw By Stroke Order □

Layers

Decor
Line ■ Fill ■ Mode Multiply
Width 1.4 Xray □
-1

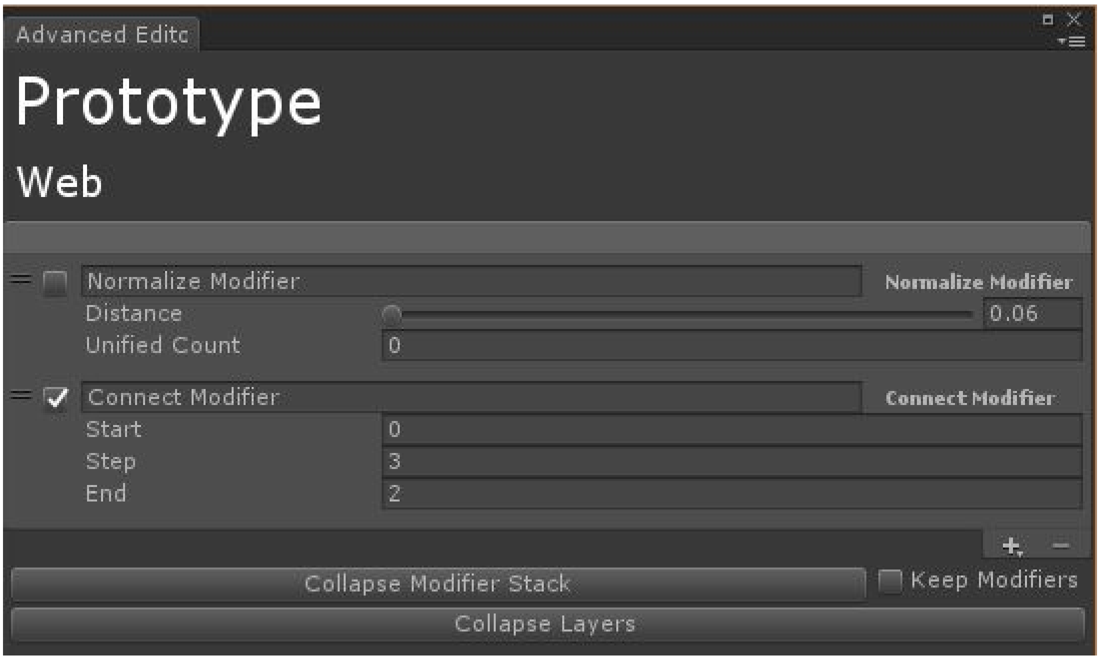
Decay
Line 🟩 Fill ■ Mode Multiply
Width 1.1 Xray ☑
-1

Critical Path
Hazards
Web

- You can Add and Remove layers by clicking the **+** and **-** buttons.
- Toggling the **Eye** icon will show and hide layers.
- **Line** and **Fill** colours will change the strokes' colours.
- **Mode** dictates the operation between the stroke colours and the layer colours.

- **Width** adjusts the width of the strokes' outlines in pixels.
- **Xray** will allow your layers to be visible through other geometry.
- Layers can be reordered by dragging the tab on the left.
- **Advanced Editor** opens the dialog that allows you to assign modifiers to your layers.

Advanced Editor



□ Normalize Modifier

Distance 0.06

Unified Count 0

☑ Connect Modifier

Start 0

Step 3

End 2

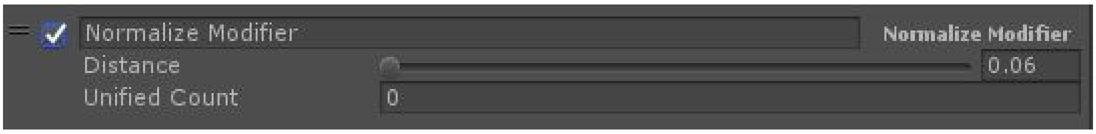
□ Keep Modifiers

- **Modifier list** is reorderable and is executed from the top to the bottom.
- **Collapse Modifier Stack** applies all the modifiers in the stack to the current layer of the **Grease Pencil Canvas**. When **Keep Modifiers** is off the stack is cleared.
- **Collapse Layers** collapses the modifiers and flattens all the layers into one.

Modifiers

Modifiers allow you to change your strokes without destroying your original data. It's easy to write your own modifiers in C#.

Normalize Modifier



☑ Normalize Modifier

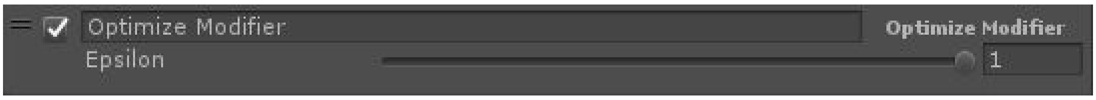
Distance 0.06

Unified Count 0

Normalize makes all the segments of all strokes the same length, while keeping the same basic shape.

- **Distance** the length of each segment.
- **Unified Count**, if non-zero, it subdivides all strokes to that number of points.

Optimize Modifier



☑ Optimize Modifier

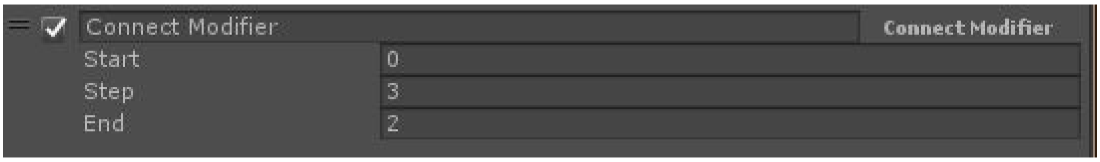
Optimize Modifier

Epsilon 1

Reduces the number of points based on curvature. Uses the same optimization method as the Line Renderer.

- **Epsilon** is the minimum deviation from a point must have from it's neighbours before being optimized away.

Connect Modifier



☑ Connect Modifier

Connect Modifier

Start 0

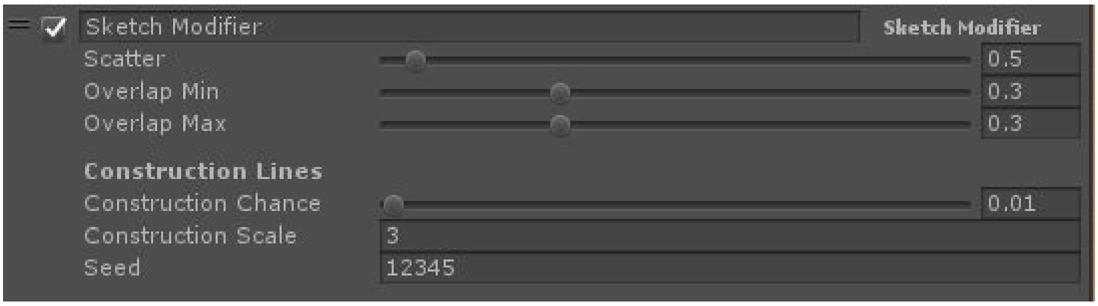
Step 3

End 2

Connect draws lines between multiple strokes. Useful to draw things like webs, train tracks, fences, etc.

- **Start** is the number of points to skip before starting to connect the strokes.
- **Step** is the number of points to skip between each connecting stroke.
- **End** is the number of points from the end before the connecting strokes end.

Sketch Modifier



☑ Sketch Modifier

Sketch Modifier

Scatter 0.5

Overlap Min 0.3

Overlap Max 0.3

Construction Lines

Construction Chance 0.01

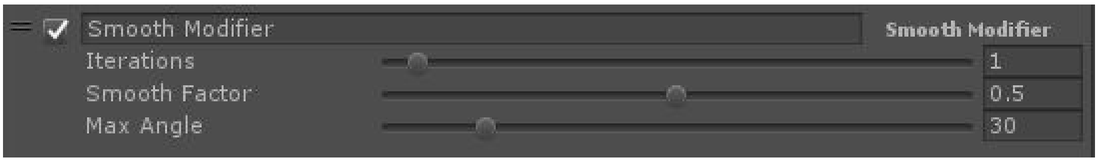
Construction Scale 3

Seed 12345

Sketch gives the effect of having the strokes sketched with short overlapping strokes.

- **Scatter** is the distance from the original stroke the new strokes are pushed out to.
- **Overlap Min/Max** the distance over the previous/next points.
- **Construction Lines** are oversized/exaggerated lines that are drawn based on probability.
- **Construction Chance** is how likely a sketch stroke will be oversized. 0 - None, 1 - All.
- **Construction Scale** is how oversized the lines are.
- **Seed** changes the randomness of the strokes to allow for variations.

Smooth Modifier



☑ Smooth Modifier — Smooth Modifier

Iterations 1

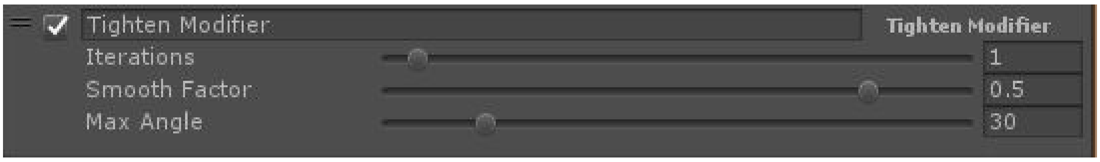
Smooth Factor 0.5

Max Angle 30

Smooth modifier averages out the points of the strokes, resulting in smoother strokes.

- **Iterations** - the number of times the smooth process is run over the strokes.
- **Smooth Factor** - how much the points are averaged out. 0 - none, 1 - Colinear.
- **Max Angle** - Any point with an angle higher than this will not be smoothed out.

Tighten Modifier



☑ Tighten Modifier — Tighten Modifier

Iterations 1

Smooth Factor 0.5

Max Angle 30

The Tighten modifier is very similar to the Smooth Modifier, except it attempts to better maintain the original shape.

- **Iterations** - the number of times the smooth process is run over the strokes.
- **Smooth Factor** - how much the points are averaged out. 0 - none, 1 - Colinear.
- **Max Angle** - Any point with an angle higher than this will not be smoothed out.

Writing new modifiers is easy, see any of the above modifiers’ source code.

Run-time Support (Experimental)

Grease Pencil 1.1 allows for the drawing and serialization of strokes at run-time. There isn’t a 1-to-1 feature parity with editor drawing.

Quick start Guide

1. Create a canvas renderer (ie. UI Image), add the **Grease Pencil Runtime** script. This is your “tablet” on the screen, this captures mouse input for drawing.
2. Next create a new Game Object with **Grease Pencil Canvas**, and assign it the **Grease Pencil Runtime** > **Active Grease Pencil**. This connects your tablet to your grease pencil object.
3. The next step is where we have a largest departure from the Editor-Time Grease Pencil, we’re going to need surfaces to draw on. For that we have **Grease Pencil Collider Surface** and **Grease Pencil Planar Surface**.
    a. **Grease Pencil Collider Surface** lets you draw on the surface of any attached colliders.
    b. **Grease Pencil Planar Surface** creates a flat infinite plane to draw on.

    Surfaces are also extendible, so you can add your own surfaces as you need them through the Surface API.

Serialization

Grease Pencils can be serialized (including modifiers) by invoking:

- **GreasePencilCanvas.Serialize()** which returns a json string.
- **GreasePencilCanvas.Deserialize(string json)** which fills the current Grease Pencil Canvas with the strokes defined in the json string.

Grease Pencil Listeners (Experimental)

Grease Pencil Listeners react to drawings and strokes. The example included in this package, **Build 2D Edges**, creates 2D Edge Colliders based on strokes.

History

1.1.1

- Updated for better Unity 2019.1 support
- Added LWRP and HDRP support (Grease Pencil Canvases may need to be enabled and disabled if they don't render, but will work correctly after that)

1.1.0

- Added modifier stack
- Added per stroke fill and line colours, with a palette of customizable colours.
- Renamed Grease Pencil to Grease Pencil Canvas for clarity.
- Added experimental run-time drawing and serialization

1.0.1

- Fixed bug in run-time drawing

1.0

- Initial Release

Contact the developer at: wahooney@gmail.com