COMPLETE Paint

DOCUMENTATION 0.3.2

A REALTIME PAINTING & DRAWING SOLUTION

HELLO@DIGITALSALMON.CO.UK

COMPONENTS
GETTING SET UP

INTRODUCTION

Welcome to Complete Paint. There are several example components included in the package which will get you ready to creating a painting application in no time.

Below is an outline of the components that are included in the package.

Some components are required for the system to work, others are optional. The majority of the components included in the package have their source exposed and can be modified or extended to fit the needs of your application.

CANVAS

In order for a surface to recieve paint, it must have a *PaintCanvas* component. There are two types of Paint Canvas included in the package;

MeshPaintCanvas | This should be added to an object with a MeshCollider component.

RawImagePaintCanvas | This should be added to an object with a RawImage component.

In both Canvas types, there are a few properties which can be edited:

Canvas Size | The resolution of the texture you are painting to.

Default Fill | The default texture to fill the canvas. Usually white/clear.

Brush Size Multiplier | Used to allow painting between multiple canvases with different sizes. Usually 1.

Uv Channel | The UV Channel you are painting to. Usually Uv0.

Max Undo Steps | The full painted canvas texture is stored in an undo buffer. Large buffers will allow more undo steps but increase memory consumption.

In the *MeshPaintCanvas*, there are some additional options.

Target Material | The material you are painting to. If left blank, this will be the first material on the Renderer component.

Target Texture Property | The texture property of the target material which you are drawing to. Usually '_MainTex', but could be used to draw to Normals/Roughness/Metallic, or a custom shader property.

PAINT BRUSH

In order for paint to be applied anywhere, there must be a *PaintBrush* component to apply it.

There are two types of PaintBrush included in the package.

MouseBrush | This brush applies paint by projecting from a camera, using the mouse position.

WorldSpaceBrush | This brush projects paint from a world transform. This is the best brush for VR.

Brushes are projected into the world, and therefore have a few properties.

Pass Through | Should only the first surface hit receive paint, or should all surfaces have paint applied?

Max Per Frame Threshold | The maximum uv distance a paint stroke will move in a single frame. Reduce this value to avoid drawing lines between uv islands. Increase this value to reduce 'skipping' on fast strokes.

Max Distace | What is the maximum projection distance for paint to be applied. A VR paint brush must be 0.1, whereas a laser pointer brush would be closer to 100.

Paint On Mesh/UI | Should this brush apply paint to this surface type.

A Mouse brush also has some additional properties.

Projection Camera | Which camera should be used to calculate projections. Usually this is your main game camera.

Emulate Distance | Since brushes are configured in 3D space, a mouse brush 'pretends' to project from a fixed distance, as set here.

Mouse Button | Which mouse button is used for input.

BRUSH CONFIGURATION

A PaintBrush component will decide how the brush projects, but the **BrushConfiguration** component dictates the manner in which the paint is applied.

BrushConfigurations are assigned in code, or using the included **BrushConfigurationSwitch** component.

Brush Size | The size of the brush, in UV space.

Blend Mode| Should alpha contribution from this paint increase or decrease canvas alpha. Set to subtract to erase.

Brush Texture Style | The way a brush texture is selected from the **BrushTextures** array.

There are also two subclasses of BrushConfiguration, which override some of the brush behaviour.

SprayCanConfiguration demonstrates how to modify the alpha or size of your paint based on the projection distance.

EraserBrushConfiguration demonstrates how to cause your brush to be subtractive/erase.

PAINT

Paint must be assigned to a PaintBrush in order to paint on a surface. There are two types of paint included in the project.

ColorPaint | A Fixed colour of paint

RainbowPaint | A dynamic rainbow paint colour.

Paint can be extended by inheriting from **Paint**, and overriding the **GetPaintColor** method. For an example of how to do this, open up the **RainbowPaint** script.

HELLO@DIGITALSALMON.CO.UK