Name   Just van der linde

Class   GDV2

Date   01-16-2018

Tool   Vertex Painter

Doc   Technical document

THE TOOL

Intro.

For years artists and designers have been struggling with texturing,
UV mapping and shaders, but no more.
This amazing tool takes care of all these problems.
Artists and even designers can now texture and design 3D models in the
editor window of Unity.

Why this tool?

I chose this tool while wandering through the asset store, browsing
tools for inspiration.
I wandered upon various mesh and shader painters, which really looked
cool and inspired me to make something in this direction.
Because we are going to treat shaders later on in the course, and
because I'm scared of it I decided to make an easier version of it.

What does the tool do?

The tool takes the vertices from the mesh selected, and add colors to
them. If a vertex already contains a color, a mathematical formula is
applied adding a lerp between colors.

The buddy system

For the school assignment we were told to output data from your tool,
and import data from someone else's output.
I tried creating output for my tool, but it didn't work out.
The first method I researched was binary formatting.
Later on I learned this method was not suited for colors and vector3.
I also tried exporting it to XML, but didn't figure it out in time.
The array containing the selected vertices and the array containing
the colors count up very quickly, so XML also didn't sound suitable to
me. If this part causes me to fail the assignment, I'll research other
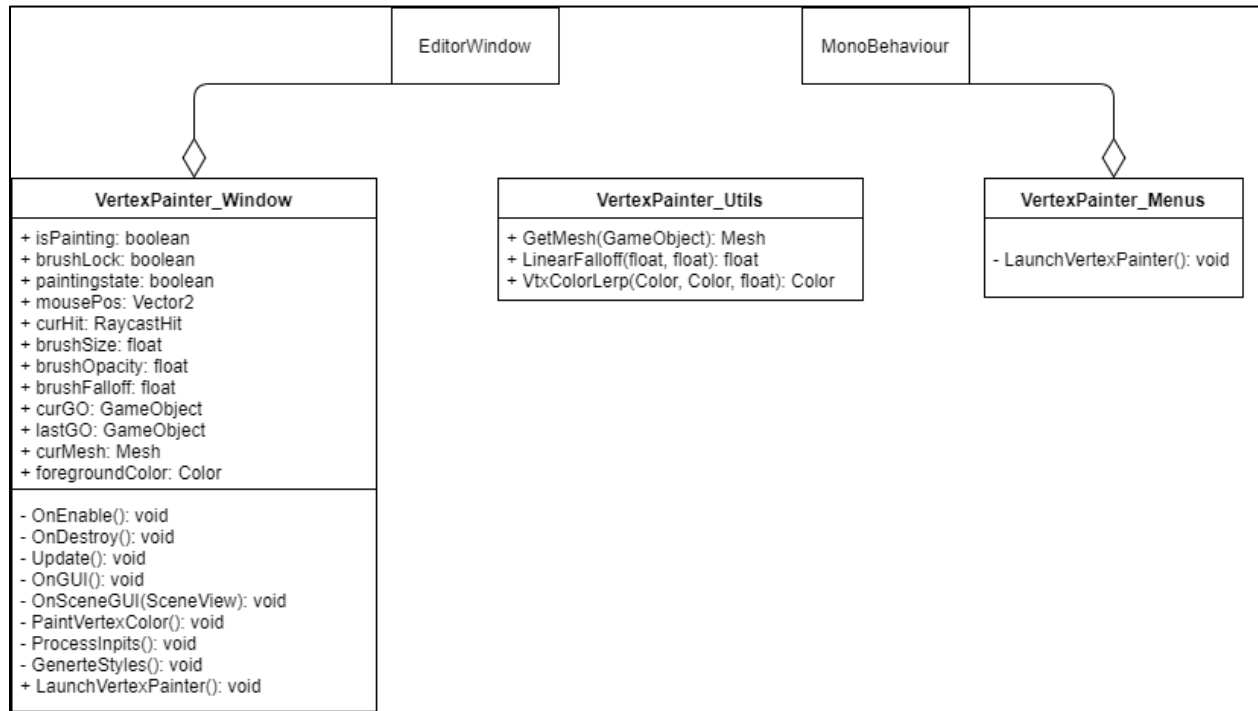methods and get it done.

ACTIVITY DIAGRAM
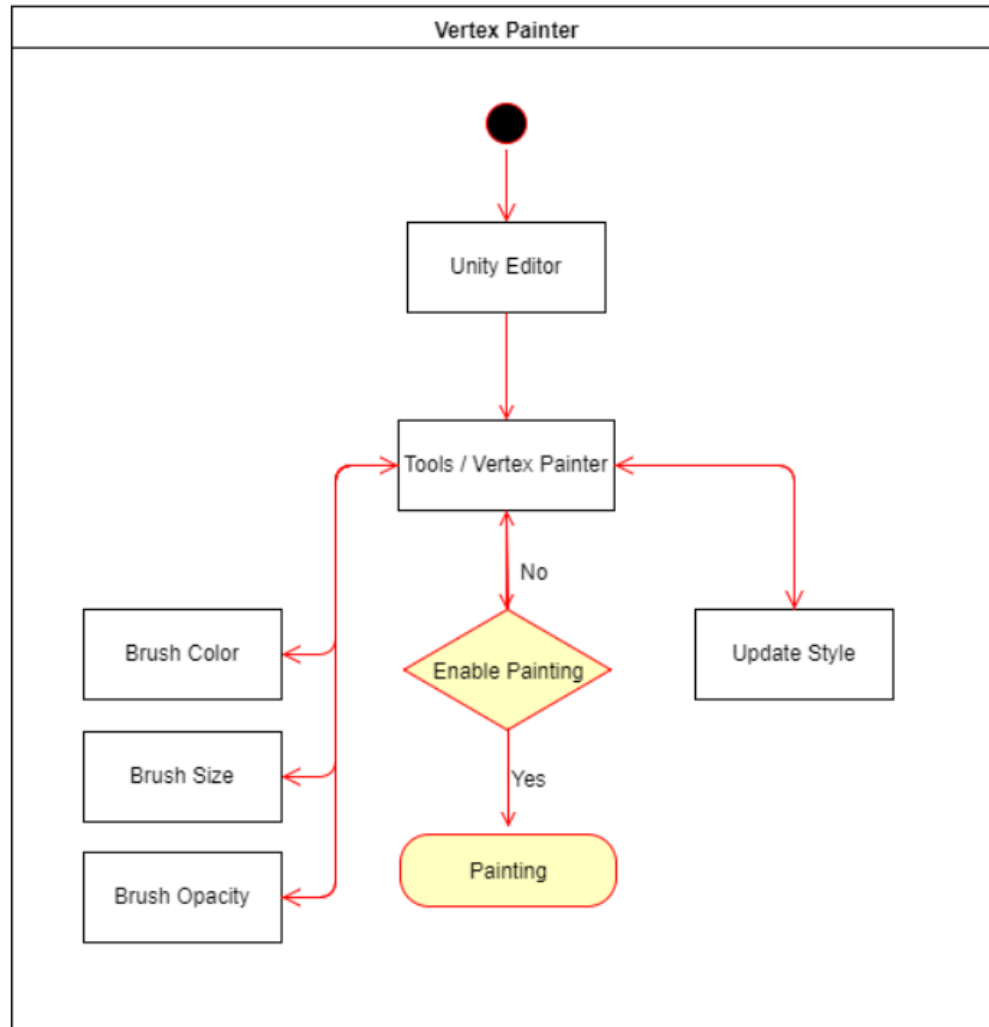
When entered the Unity Editor, the player can choose the Vertex Painter from the Tools tab.
When opened, the user cannot paint until he has pressed the "Enable Painting" button, or used the hotkey (P).

Update style is a more advanced feature, which let the user update the GUI style (which is not done automatically).

CLASS DIAGRAM

```
                    ┌──────────────┐              ┌──────────────┐
                    │ EditorWindow │              │ MonoBehaviour│
                    └──────────────┘              └──────────────┘
```

VertexPainter_Window
+ isPainting: boolean
+ brushLock: boolean
+ paintingstate: boolean
+ mousePos: Vector2
+ curHit: RaycastHit
+ brushSize: float
+ brushOpacity: float
+ brushFalloff: float
+ curGO: GameObject
+ lastGO: GameObject
+ curMesh: Mesh
+ foregroundColor: Color
- OnEnable(): void
- OnDestroy(): void
- Update(): void
- OnGUI(): void
- OnSceneGUI(SceneView): void
- PaintVertexColor(): void
- ProcessInpits(): void
- GenerteStyles(): void
+ LaunchVertexPainter(): void

VertexPainter_Utils
+ GetMesh(GameObject): Mesh
+ LinearFalloff(float, float): float
+ VtxColorLerp(Color, Color, float): Color

VertexPainter_Menus
- LaunchVertexPainter(): void

I divided the tool up in three scripts:
The Menus, which simply contains the "Tools" menu in the
Unity bar, followed with the "Vertex Painter" option.
Utils, which contains helpful tools and additions called in
the Vertex Painter.
Window, which is the main part of the tool, containing the
main mechanic and the GUI style.
The Window script is build up in 4 regions: Variables, Main
Method, GUI Methods, TempPainter Methods and Utility Methods
for better readability.

PLANNING

WEEK	GOAL	NOTES
1	Orientation	
2	Orientation	
3	Work on first concept	Automated AI assigner
4	Receive feedback on concept	Still the AI assigner
5	Work on first concept	Fail horribly
6	Think of new project	Vertex Painter
7	Vacation	
8	Vacation	
9	Work on new project	Vertex Painter
10	Finalize project	