PaintSpace:

Exploring the Concept of Painting on a 3D Canvas Using Virtual Reality and 3D Input

Benjamin Madany

Advisor: Robert Signorile

Boston College

Abstract

3D technology has seen a wide range of innovations, from 3D graphics and modeling to 3D printing. Among the most recent of these innovations are immersive virtual reality and 3D input. These have allowed for the creation of unique, 3D experiences, and they also present the opportunity for a wide variety of applications whose purposes range from entertainment to educational or medical use. One possibility is an extension of 3D modeling that utilizes these recent technologies to present a 3D canvas to an artist. Applications such as Google's Tilt Brush explore this concept of drawing in 3D space. As the ability to draw in such space is novel, development of such a tool presents several challenges. This thesis explores the process of building a 3D painting application. I first present the key challenges encountered during development. Then, I detail various solutions and options related to these challenges. Next, I examine the capabilities and state of my application, and finally, I compare it to other available applications.

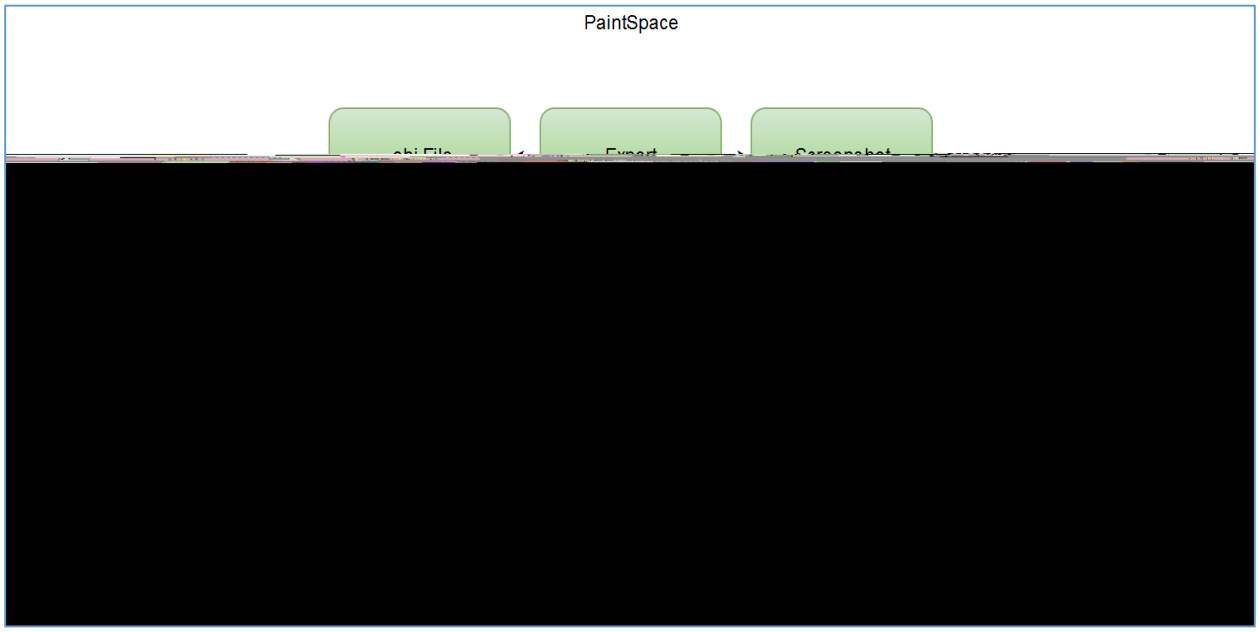
while keeping in mind the relevant frame-rate requirements for virtual reality? Also part of this challenge was the question of what the end goal of “drawing” was supposed to be. Should the resultant 3D artifact be a static mesh object intended for easy export and use in other applications? Or was the purpose to create artistic representations on a canvas, emphasizing aesthetic over practical use? Perhaps pursuing both end goals was also a viable option that would give the application greater depth. This question of the creation of 3D artifacts ties into the third major challenge, what sort of user interface and options are a result of and support the previous challenges?

The application’s user interface was primarily determined based on the chosen input device. This further influenced what sort of options would be available and how to design them. Another major influence on this challenge was the question of what sort of tools and options the application needed to support. In an application whose purpose is akin to painting on a canvas, it made sense to include brush color and size options, and other related configurations. Based on the type of artifacts being created, options such as export and import were also necessary to consider.

PaintSpace required solutions to a variety of challenges, and in the next chapter of this thesis, the design choices made to answer the major challenges of developing this application will be detailed.

2. Design Choices

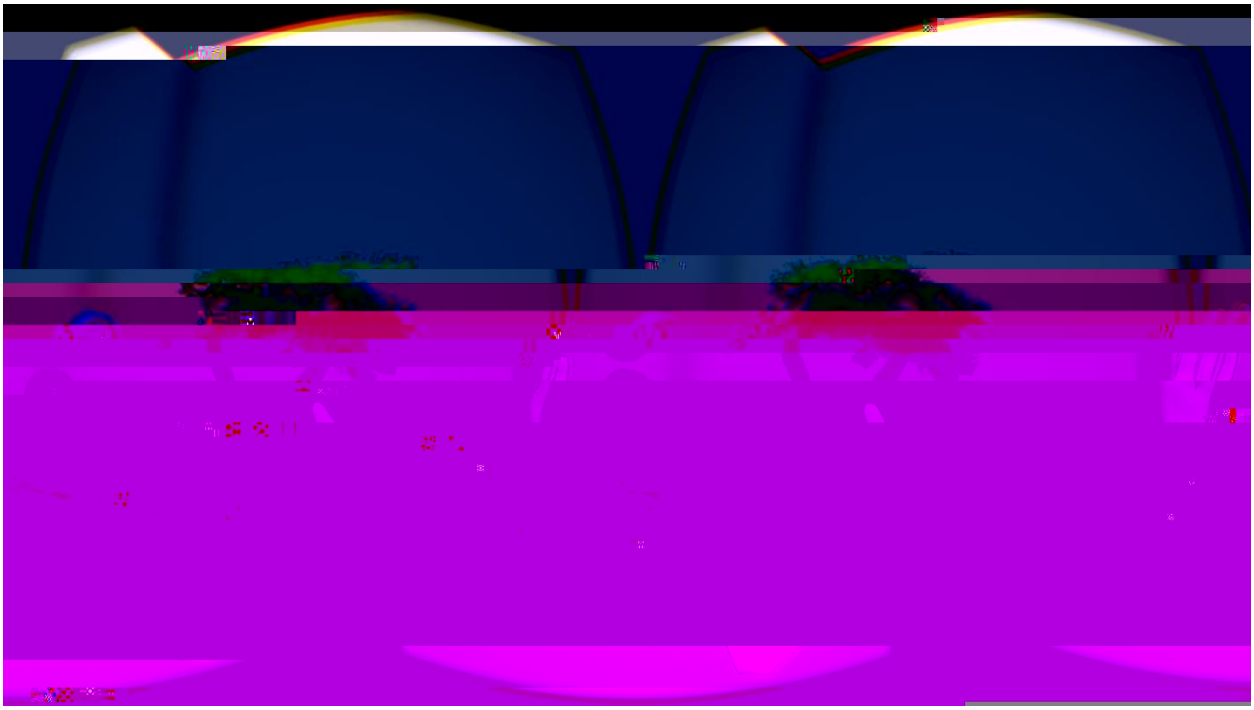
The following section will detail the methods and solutions used to solve the various challenges encountered during development, beginning with an overview of the application structure and design. The following diagram details at a high level the environment in which PaintSpace functions.



PaintSpace is built on top of Unreal Engine 4, and makes extensive use of two plugins for the engine, the Leap Motion plugin and the Oculus plugin. Each of these are primarily involved in enabling the use of specific hardware within Unreal. The Leap Motion plugin is integral to receiving and handling three dimensional input, which it enables through the use of Frame objects, which contain information such as hand and finger positions, types, and other qualities. These Frame objects enable the use of the user's hand motions as the primary form of input in PaintSpace, and they provide the basis for interpreting gestures made by the user within the

Each of the lightest colored blocks represents a C++ class or pair of classes. As described earlier, the 3rd

Below is an image example of both the TouchMenuComponent and PaintBrushComponent in action. While the PaintBrushComponent itself is invisible, the results of its functions can be seen in the drawing of a tree below.



The decision to use Unreal Engine 4 was made because it provided significant convenience in terms of implementing the creation of 3D artifacts. It also helped with the implementation of input and control, and so generally made implementing various parts of PaintSpace simple. The major drawback to this approach was the time cost of gaining significant familiarity with the use of the engine and the tools it provided. A significant portion of the first month or so spent working on PaintSpace involved becoming familiar with the inner workings of the engine, especially regarding static mesh rendering data and where to obtain it. Additionally, the use of Unreal Engine necessitated developing some familiarity with C++, though the use of Visual Studio 2015 made this relatively simple.

As is evident in the above discussion of the application overview and details, the question of input was answered using a specific piece of hardware, the Leap Motion Controller. This tool provided an intuitive and simple way to interface in 3D. The user is able to "see" their own hands while in the virtual reality, and is able to make movements outside the application that are replicated accurately within it. This is certainly a "natural" way for humans to interface with many objects, and while it is fairly novel in the realm of software, it is intuitive and simple to pick up the use of one's own hands in a virtual realm. Additionally, gesture based movement was implemented to make up for the sitting experience provided by the Oculus hardware. Because the Oculus is not intended for movement around a room, it was difficult to examine a 3D artifact from all sides. To answer this, the Leap Motion Controller was used to provide specific gestures that would manipulate the character position in the application space, and allow for rotation so that the user could view artifacts they created from all sides. The Leap Motion Controller was also able to provide a significant degree of accuracy and responsiveness that would only have been beaten by specialized hardware built for use in virtual reality.

Also shown above is the end result of the challenge that involved what sorts of artifacts to create. It was decided that, in order to provide a variety of styles and options, two primary methods of creating artifacts would be used. The first is the creation of instanced static meshes, these allow for the creation of thousands of meshes whose data can be grouped together to reduce the number of draw calls made to the GPU. This is a massive increase in efficiency over creating thousands of individual static meshes, and allowed the frame-rate to maintain the necessary level for comfortable use in virtual reality. The second method was to use a GPU particle emitter to create up to millions of particles that gave off a 3D spray-paint sort of feel. This allowed for the creation of a wider variety of "drawings" such as the tree seen above.

The use of these two approaches to “painting” stems from the intuitive nature of sculpting, the most analogous real life form of art.

3. PaintSpace

4. In Comparison and the Future

It is worth noting that there exists a somewhat similar set of applications to PaintSpace already. Some exist in a state similar to that of PaintSpace, while others are much more fully featured applications. All of them are relatively recent innovations. The most notable of these is Google's Tilt Brush. This is a tool designed for use with another virtual reality device, the HTC Vive. This piece of hardware incorporates a pair of motion controllers that allow for powerful tracking and interface design. The Vive also provides a room-scale experience, allowing the user to walk around their creation rather than sit in front of it and move their character around as is the case in PaintSpace. While Tilt Brush is certainly a much more complete application with more depth than PaintSpace, the significance of its mention here lies in the implications that exist regarding the future of this realm of applications. The development of applications such as Tilt Brush, the various others that exist, and even PaintSpace is indicative of an early exploration of one specific aspect of software that the recent improvements to virtual reality have opened up. As the hardware, in every area of these sorts of applications, continues to see innovation, new styles of creation in 3D space will crop up, and old ones will see improvement. Oculus plans to release a set of 3D input devices called Oculus Touch. Leap Motion plans an improved set of hardware to be released. There is plenty of room for exploration in this field, and applications such as PaintSpace or Tilt Brush only explore a few aspects. The future of this area, as with many others related to virtual reality, is bright, and holds many unknowns.

5. References

The following references include various devices, plugins, and other applications mentioned.

Additionally, a link to the demo from the presentation of PaintSpace and the source are included.

- Demo, https://www.youtube.com/watch?v=rARAndXZMKY
- Source, https://github.com/benmadany/PaintSpace
-