

PPVR: Proprioceptive Painted Virtual Reality

Andrejs Ruzencevs
University of Toronto
Toronto, Canada
andrisole92@cs.toronto.edu

Sahil Narula
University of Toronto
Toronto, Canada
sahiln123@gmail.com

Sneha Desai
University of Toronto
Toronto, Canada
sneha.desai@mail.utoronto.ca

Sumeet Ranka
University of Toronto
Toronto, Canada
ranka47@cs.toronto.edu

ABSTRACT

Sketch canvases have grown digital with development. Where 2D space is ergonomic and precise, it is constrained. 3D space gives an immersive, boundless and life-sized environment. The main objective of this project is to build a proprioceptive hidden menu of drawing tools. This gives the user a view of their artwork without hindrance due to the menu. This application was tested during a theatrical performance of *The Tempest* written by William Shakespeare.

INTRODUCTION

This paper presents PPVR, a proprioceptive menu in conjunction with a painting system for a live theatrical performance of the play *The Tempest* written by William Shakespeare. This project allowed the system user (the cast member playing the character of Ariel) to select drawing objects from different zones around the body and create magical effects on an island.

BACKGROUND

The project takes into account the research work done in the areas of the kinesphere, proprioception, and the intensive development in the field of 3D drawing.

Regions around human body

Kinesphere is the abstraction of the volume surrounding a human body. The volume is defined by the points that can be reached by extending the limbs while standing at the same place on either foot. It is mainly categorized into two zones, *Super Zone* and *Mini Zone*. *Super Zone* refers to the region that requires off-balance reach and the *Mini Zone* are the areas which are easy to reach. The figure 1 depicts the kinesphere around the human body.

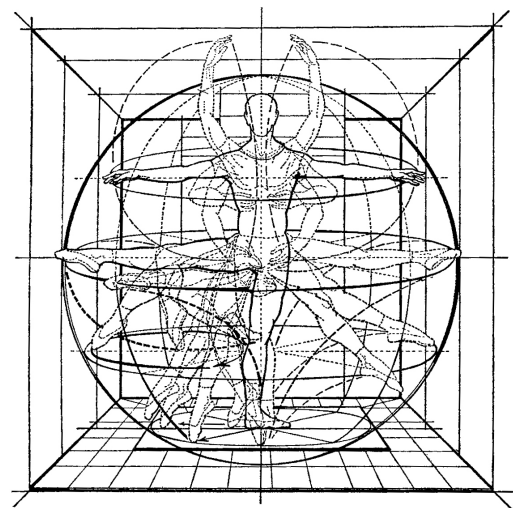


Figure 1. Kinesphere

Proprioception

Proprioception is defined as the sense of perceiving the position and movement of our body. Sometimes described as the "sixth sense", it can be also be defined as the sense of the relative position of one's own parts of the body and strength of effort being employed in movement[1].

KINESPHERE

As part of exploring the shape of the kinesphere, PaintLab, was used. The extensive painting tools were used to draw a skeleton depicting the regions around the human body. The experiment was conducted in two scenarios, a standing position and a kneeled position on one knee. The positions were chosen to explore the changes that happen in the kinesphere under different circumstances. Figures 2 and 3 depict the kinesphere under the two scenarios respectively.

The red lines denote the the skeleton of the kinesphere and the volume enclosed by the red lines include the reachable regions. The green lines represent the human body. The head, shoulders, chest, waist, and thighs have been represented for the most part. These regions of the human body are chosen

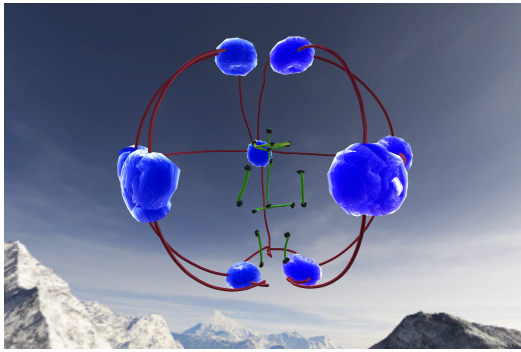


Figure 2. Kinesphere in a standing position

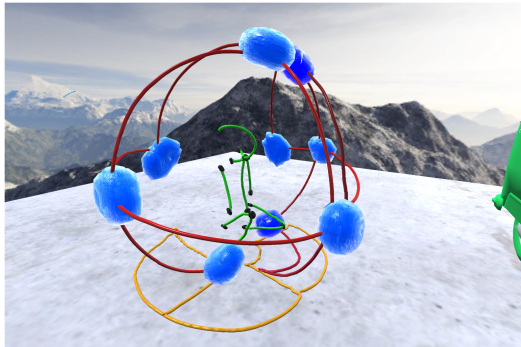


Figure 3. Kinesphere in a kneeled position

based on reachability and the awkwardness to utilize them for different functions in various scenarios. The small blue volumes describe different relative directions: left, right, top, bottom, front and back. There are two blue volumes each for front, back, top and bottom. They represent the extension in the corresponding direction using each of the hands. For example, the leftmost blue volume in the figure 2 represents the maximum extension of the left hand towards the back of the human body.

Comparing the two kinespheres, it can be noted that the kinesphere created in a kneeled position collapses on a 2D plane. In other words, it is similar to a projection of a part of a 3D sphere on a 2D plane. The regions obtained can again be used to define different zones around the human body.

PROPRIOCEPTION

In order to leverage the space around the body for the selection of different painting tools, two directions are proposed. One is a zone-centric approach, in which zones are predefined around the user, and the other is an object-centric approach, in which the zones are defined by the user. Both of these experiments are setup in the same fashion. A user would see an avatar in front of them with a small sphere placed somewhere in the avatar's kinesphere. The user would then attempt to find the zone around their body that corresponds to the sphere placed around the avatar by extending their arms (and asserting their understanding using the trigger button on the HTC Vive Controller).

The experiments are implemented in a game-like scenario where the user would have to meet certain requirements in order to move to the next level of the game. To qualify for the next level, a user must attain an accuracy of 90%, reach a threshold of at least 10 correct attempts within a level, and be within a user-defined radius threshold of the sphere that is placed around the avatar, all within 60 seconds. The user-defined threshold is set at the beginning of the game, allowing the user to choose their desired difficulty level. The reason for implementing the last metric is primarily for larger zones (lower levels), so that users would not be marked as having a correct attempt when selecting an area around their body that is very close to the boundary of two zones. The following sections describe the implementation of the two experiments in greater detail.

Zone-Centric Experiment

In this experiment, zones around the user are predefined. The number of zones around the body increases as the user moves to the next level. These zones are defined as sections of a sphere encapsulating the avatar (which are mimicked on the user as well). A correct attempt in this game is pressing the trigger on the controller within a specific range of ϕ and θ from the spherical coordinates of the specified zone around the avatar. For example, in a level with four zones, suppose a sphere appears in the top left region of the avatar then the desired range of ϕ will be $[270, 360]$ and the desired range of θ will be $[0-90]$.

Object-Centric Experiment

In this experiment, zones around the user are defined by the user. The number of zones defined by the user increases as the levels are cleared. A correct selection in this game is defined as the pressing of the trigger on the HTC Vive controller at a point whereby the Euclidean distance between the clicked point and the center of the specified zone around the avatar is smaller than the Euclidean distance between the clicked point and the center of any of the other zones. Note: The correct selection does not imply a successful attempt. The distance from the center of the zone has to lie inside the user-defined radius threshold.

PAINTING

The proprioceptive environment developed as part of the proprioception experiment is used in the 3D painting to create a menu that will not hinder the field-of-view of the artist. The following sections explain the different painting components of the application developed. Broadly they are categorized in three features: SkyDraw, FloorPaint and Tornado Tool.

Sky Draw

Sky Draw is a submodule that allows drawing of Unity prefabs anywhere in the 3D space. When a scene starts, a prefab is initiated 10 meters away in the front of the controller. The trackpad then can be used to move the object forward, backward, and to change the scale of the object (left and right on the trackpad). The object is created and placed on the indicated position of the landscape on the press of the trigger button. The Sky Draw component has several parameters:



Figure 4. PPVR Sky Draw in action

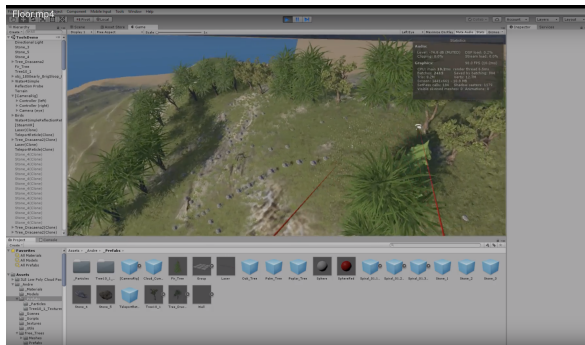


Figure 5. Floor Paint in action

1. **Step** - Checks the distance between the last drawn object and the newly drawn object. E.g. If the distance is 1, then the next object is be drawn 1 meter away from the previous one. Otherwise, nothing will happen on trigger press.
2. **Prefab Array** - An array of prefabs to draw. The tool randomly selects the next object from the prefab array. Prefabs can be anything from a simple transform to a complex particle system.
3. **Single Mode** (*boolean*) - Draw as many objects as possible on trigger hold, or draw object one-by-one on separate trigger clicks.
4. **Cooldown** - If a single mode is a false, cooldown to draw the next object. (if we don't want too many objects)

Figure 4 shows the Sky Draw functionality being used and a video of it can be found here: <https://drive.google.com/open?id=1f47VdKXFGU19c3LRa6R9G5RfsAI4mVkw>.

Floor Painting

This features is similar to Sky Draw, but creates objects on the ground. The tool works in the similar manner as the Sky Draw, and shares the majority of parameters such as Step, Prefabs, Single Mode, Cooldown. Dotween Pro has been used to create the effect of a tree growing up from the ground. Figure 5 shows the Floor Painting tool being used and a video of it can be found here: https://drive.google.com/open?id=1Nb3QsY8ZqOtT_49Nk72rfeASkdBagNa6.

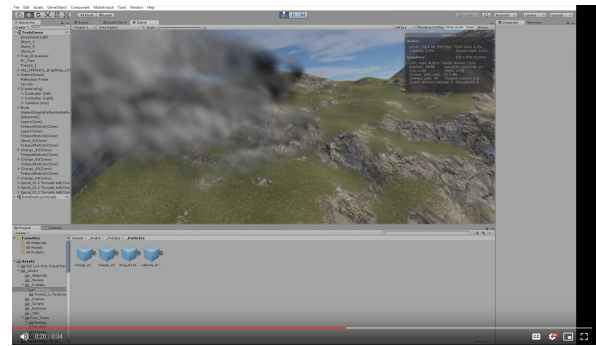


Figure 6. Tornado Tool in action

Tornado Tool

This is a simple tool that allows the user to create an array of points on the ground and launch a tornado on the press of the grip button. The tornado will traverse along the selected points and stop on the last one. Figure 6 shows the Tornado Tool being used and a video of it can be found here: https://drive.google.com/open?id=1Qrp4QBenXSbiDgRs6vkoXLLaccB8_Prr.

THE TEMPEST

PPVR was tested during a theatrical performance of the *The Tempest*. In this play, Ariel, a magician, casts magical spells on an island. That magic is enacted by the actor playing Ariel using PPVR to create effects in 3D. The resulting image of what Ariel sees in the headset is projected on a concave 180 degree backdrop at the back of the stage.

Integration

The project required a magical island giving the user ability to walk on the terrain, select across multiple zones around the user, draw/paint in the open world using strokes and cause environmental changes. Integration of each of the functions is described as following:

1. **Walking on the terrain** - 'Movement.cs' script is used for handling the controller touchpad events to move the first person in the scene. The script is generic enough to be attached to any controller. For our presentation, we integrated it to the left controller leaving the mission critical features to be integrated on the right (active) hand. On every touch event, a force is incorporated to the first person character in the direction of the touch. The rotation is taken care by the camera head angle itself.
2. **Select across multiple zones** - Proprioceptive zones are attached around the right part of the first person character giving access to four zones around the user with the right controller. 'ZoneFinder.cs' is also controller independent and could be incorporated to each/both of the controllers. The kinesphere around the body could be split into as many zones as required, but for our presentation we had split it into four zones around the user's chest. Two zones above the first person character are incorporated with environmental changes causing rain from one zone and tornado from the other. Zones below the user are preset for selecting array of different prefabs. For our presentation, we used

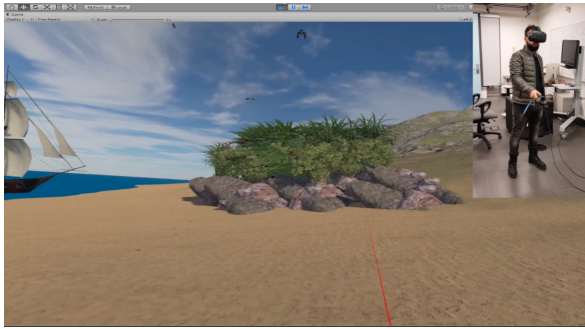


Figure 7. Painting/Drawing in the open world

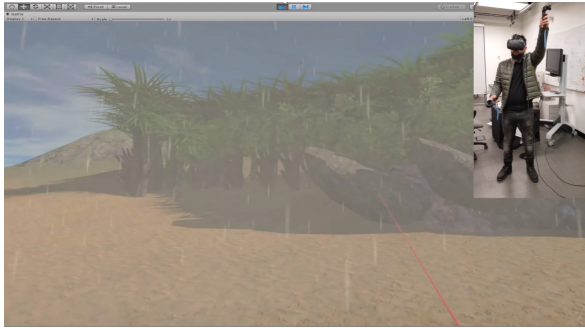


Figure 8. Initiation of rain

one zone for selecting stones and the other for trees. The zones are selected using controller's grip button.

3. **Painting/Drawing in the open world** - Allows the user to draw from the selected array of prefabs in the scene using the trigger button. The user can draw as many objects as possible using step mode or one object per trigger using single mode.
4. **Cause environmental changes** - Pressing grip button in either zone above the first person character causes the initiation of rain or tornado. Intensity of the rain could be controlled by the touchpad of the right controller. Tornado could also be executed by initializing various points through which the tornado should pass but due to lack of button ergonomics, it couldn't be integrated for the presentation.

Figure 10 shows the individual playing Ariel using PPVR to create magical effects on the island.



Figure 9. Initiation of tornado



Figure 10. PPVR in action

FUTURE WORK

For the proprioception side of the project, it will be important to perform the experiments and collect all relevant user data and feedback. For painting, the future work will be adding extra functionality to the tools, such as dynamically choosing objects from a side-menu, a terrain tool, and a teleportation mechanism to navigate through the 3D scene. A slider on the human body could be added to control developed features such as paint brush size or rain intensity for finer user control. In the end, the final integration of the study of the proprioception experiment together with the new features in the painting tool may give better results.

CONCLUSION

This project has led to the creation of two proprioceptive game-like experiments and three drawing tools. We would have liked to conduct our proprioception study using the experiments created, however, due to time constraints, we were unable to do so. The 3D environment allows for the opportunity to eliminate a menu in constant view and leverage the space enclosed by the kinesphere around the user. PPVR uses this idea and merges the efforts on both the proprioceptive and painting sides of the project to create an application for the performance of *The Tempest*.

REFERENCES

1. 1994. *Mosby's Medical, Nursing & Allied Health Dictionary*. Mosby-Year Book, Bloomington, IN, USA. 1285 pages.