Exploring Methods to Generate an Accurate Night Sky for VR

Anna Sikkink

Advisor: Jason Leigh, Ph.D.

MS Plan B, Spring 2023

Introduction

Kilo Hōkū VR, a virtual reality simulation of sailing on the double-hulled sailing canoe Hōkūleʻa, began as a class project in 2016, developed by Patrick Karjala, Kari Noe, Dean Lodes, and me. Our initial goal was to discover how a virtual reality (VR) environment could aid in the learning and teaching of Modern Hawaiian wayfinding. Users experience being onboard the Hōkūleʻa on the open ocean, where they can view stars and highlight constellations, and see the Hawaiian star compass in context. With these tools, users can apply Modern Hawaiian wayfinding techniques to navigate and sail the Hōkūleʻa in VR. Kilo Hōkū VR has since undergone continual development as my team and I work to build the simulation into a more effective teaching tool, including the development of a port to the Quest 2 VR headset.

A user learning with Kilo Hōkū VR under instructor guidance.

Motivation

A significant update needed for Kilo Hōkū VR was to improve the original implementation of the sky to allow for more precise interactions and improved accuracy of the sky visualization.

The initial implementation of the sky in Kilo Hōkū VR consisted of a sphere object with a starfield texture displayed on the inside of the sphere. Colliders were manually placed over each constellation to allow for interactivity. This method was a simple way to create a mostly accurate sky and include the interactions we wanted, but some visual defects are apparent with this method, and without positional data for the stars, this sky visualization could not be easily improved upon in its current state.

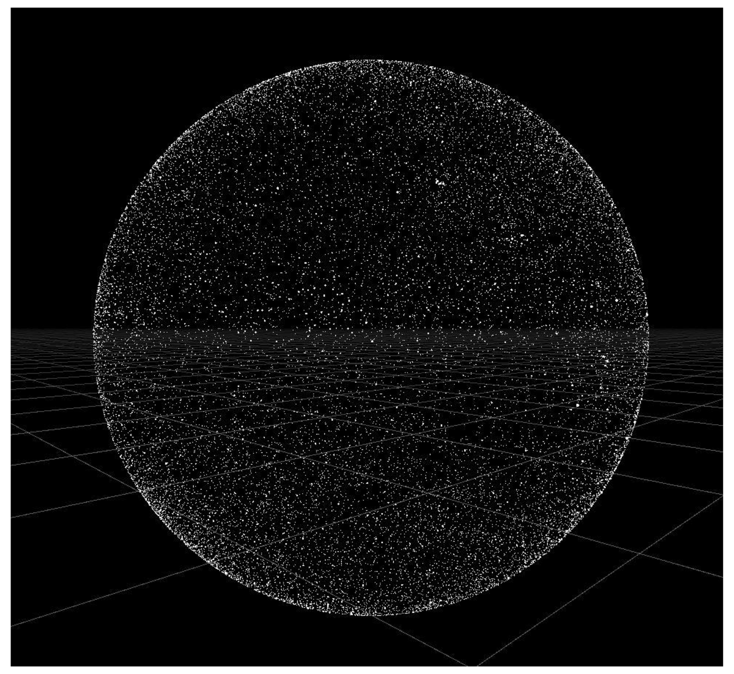
Methodology

My goal with this project was to create a new virtual celestial sphere which could be utilized in Kilo Hōkū VR, one in which each star would be generated based on positional and observational data from a star catalog. With the star position data in the simulation, the stars themselves will appear clearer and accurately placed. The Hawaiian starlines, constellation lines, and colliders can also be generated from the positional data.

The celestial sphere in Kilo Hōkū VR is intended to be a tool for learning the Hawaiian starlines and constellations to build the visual recognition skills needed to navigate using Hawaiian wayfinding techniques. Therefore, it is most important to display the stars which are visible to the naked eye, which includes those of +6.5 magnitude or brighter. I decided to use the Hipparcos (HIP) star catalog, which contains 117,955 stars of up to magnitude +12.4.

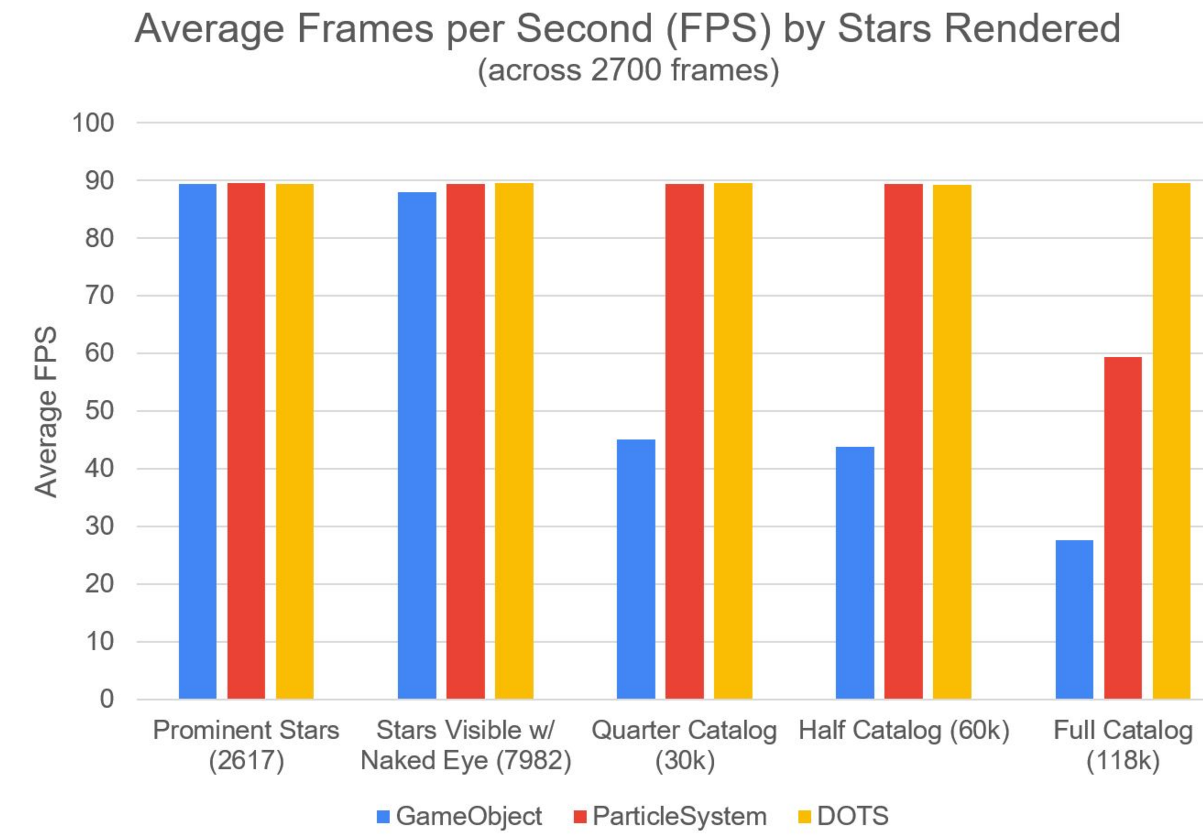
I investigated three methods to generate an accurate representation of the night sky and determined their suitability for use in Virtual Reality (VR). Developed in Unity3D, these implementations were created using GameObjects, ParticleSystem, and Unity's Data Oriented Technology Stack (DOTS). DOTS is a combination of technologies and packages that provide a data-oriented design approach to building games and simulations in Unity, and is still in pre-release at the time this project was completed.

To compare the efficiency of each method, a series of performance tests were conducted, generating stars from magnitude +5.5 or higher, to the full catalog.



Full HIP catalog generated with DOTS, as seen externally.

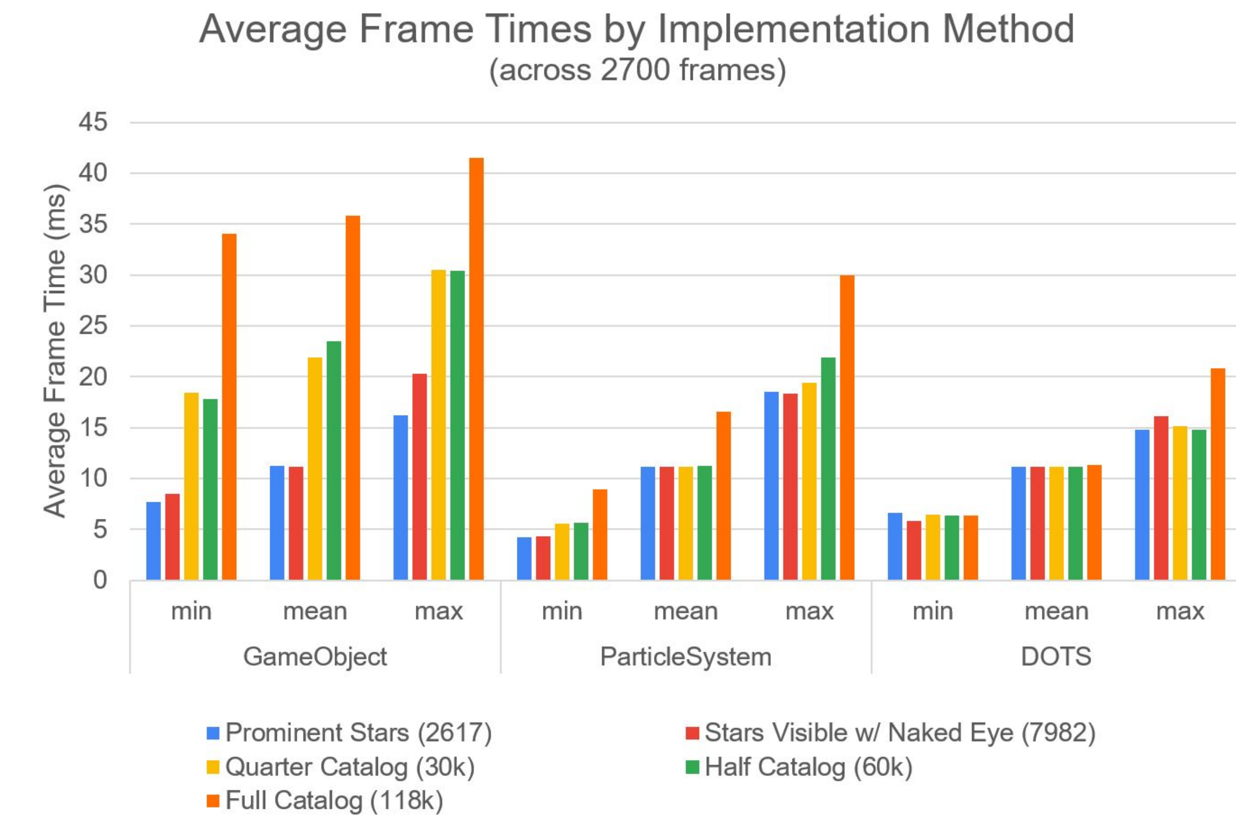
Results



A chart of average FPS results for each implementation method, by each of the five tests completed.

Unity's Profile Analyzer package was used to obtain data on the amount of time it takes to create each frame in milliseconds (ms). This data was recorded over 2700 frames, which is roughly 30 seconds of time at 90 frames per second (FPS), for each of the five tests and for each implementation method.

When considering the entirety of the data recorded for frame times, we can see the variations between each implementation, and can discern where each method starts struggling to maintain frame times within the target budget of 11.1 ms, which is required in VR applications to avoid VR sickness.



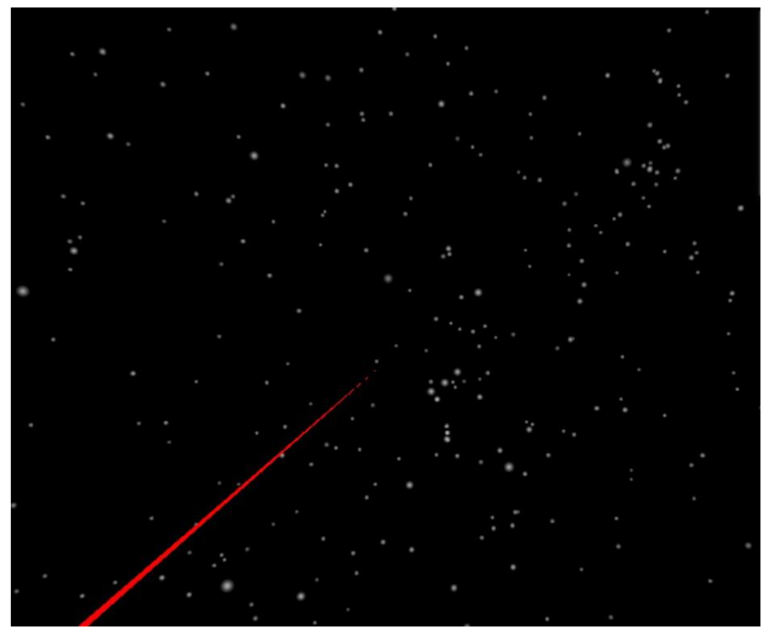
A chart of average frame time results for each implementation method, by each of the five tests completed.

Conclusion

Based on the test results, if we are generating only the stars visible to the naked eye, the mean frame times for all methods are within the target budget and run at nearly 90 FPS. Once rendering a quarter of the catalog (30,000 stars), the GameObject method drops off drastically, with an average frame time nearly double the target. The ParticleSystem method is able to keep the target frame times until the entire catalog of stars is rendered. However, the DOTS implementation average frame times remain around our target frame time for all tests, including the full star catalog.

The results show that using Unity's Data Oriented Technology Stack, and designing the implementation from a data-oriented approach provides the required performance benefits needed to keep the application running at 90 FPS, even when rendering nearly 118,000 star entities.

This project has shown that a functional version of a generated celestial sphere has been proven effective and is able to run in virtual reality. By utilizing Unity DOTS, this is feasible with a large-scale data set like the HIP star catalog. My Kilo Hōkū development team and I can use the findings of this project to enhance the next version of the Kilo Hōkū VR application by replacing the original virtual celestial sphere with a generated one using the DOTS implementation techniques. We will be able to use this DOTS version to display all stars visible to the naked eye in virtual reality on the Quest 2 headset, ultimately making the Kilo Hōkū VR application more accurate and effective as a learning tool for classrooms.



Generated stars in the constellation Orion, as seen from within VR.

Information and Computer Sciences Department, University of Hawaiʻi at Mānoa