Solaris VR – Short Summary

Jacob Centner with James Ward
Computer Science
University of Wyoming

Engineering, Honors *Gillette, Wyoming*

This summary is intended to provide an overview of the product produced for this Honors thesis.

Solaris is a senior design project by Jacob Claytor, Jeremy Reynolds, Mickale Bush, Trystan Bennett, and me. It was constructed for Computer Science Senior Design and for this Honors project. Solaris is a virtual reality solar system, built for the HTC Vive and soon to be ported to the C.A.V.E. (Cave Automated Virtual Environment) at the UW Shell Visualization Center. The project was designed and implemented in the Unity engine, with modeling performed in Blender.

Space exploration has always been the cutting edge of technology, and VR currently is the cutting edge; our team wanted to bring the two together. Besides that, humanity is at the dawn of a new space age, with SpaceX, Blue Origin, and other companies leading the way. VR opens up new possibilities to get people excited and involved in that.

At its core, Solaris is designed to be a sandbox. What happens when you remove the sun, or add another one? What happens when you double the mass of a planet? What if a black hole appears somewhere nearby? Interaction is a primary goal, and the HTC Vive was chosen to facilitate that since it has two programmable controllers and accessible programming libraries (Steam VR).

Another primary purpose of Solaris to be visually stunning. Especially in VR, where the immersion is enhanced tenfold, graphical fidelity is paramount. Visual effect was used as a mechanism for inspiring interest in space.

It is also designed to be educational. In Solaris, any object can be selected, and an information panel is displayed showing details about it. Also, Solaris operates using realistic Newtonian physics.

The implementation of Newtonian physics was a large part of my contribution to the project. Fifty times a second, gravitational forces are calculated between all bodies in the solar system, and forces are applied in real time. If the user decides to move the Sun, the planets follow as they would had it happened in the real solar system.

Another significant contribution of mine was the calculation and determination of scale. The solar system is vast, with Neptune at a distance of 30.1 AU from the Sun. Planets are extraordinarily small in comparison (Earth's radius is 4.26×10^{-5} AU). Creating Solaris with entirely realistic sizes would either render the planets too small to see, or if the scale was such that the planets could be seen, Unity would be unable to handle the immense distances. To solve this, the planets are scaled up approximately 21.5 times from what is real so that they are still visible. Even with this, they are small, so trails were implemented to show the path they have traveled in the past few seconds, allowing them to be found. Also, a mini-map system was implemented, allowing instant transportation to any of the planets, to further aid accessibility.

My final contribution to the project was adjustable time-scaling. The simulation can run at speeds of up to one year simulated per real-time second and still be accurate, down to freezing time completely. This simulation is implemented with no cost in CPU or GPU performance, and does not suffer meaningful mathematical error.

Some other features of Solaris include grabbing and moving objects, crashing planets together to destroy them, custom planetary ring systems, detailed textures, and a vibrant Sun. There is also an ambient audio track recorded by one of the team members, and a low frequency noise played when near large objects to add a psychological sense of immense mass.

The project was a success, although there is still a lot of work to do and many ideas the team wants to implement this summer. Those ideas are:

- Creating and throwing new objects of customizable mass and size
- Adjusting the mass of existing objects
- Allow direct deletion of objects
- Additional objects in the solar system, like comets, asteroids, satellites, and moons
- A custom model for the International Space Station

Solaris was demoed successfully on May 10 to computer science faculty, students, and the public. The next steps are porting the project to the C.A.V.E. when its version of Unity is updated this summer, implementing additional features and content, and polishing the existing content.

A demo version of the project, built for Windows, is available upon request, by email to jcentner@uwyo.edu.