CS 1566 Final Project Report:

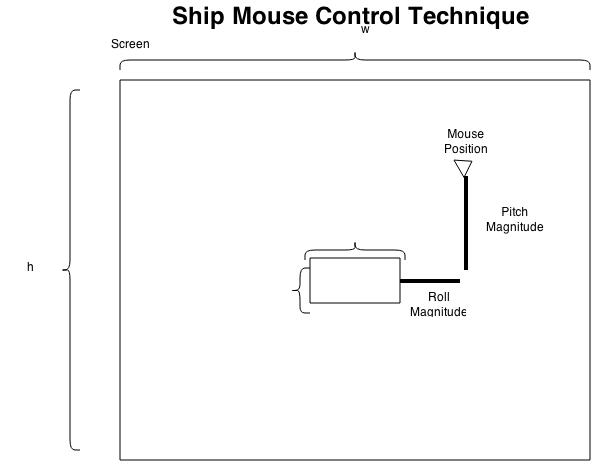
1. **Introduction**

Our project is a simulation of a procedurally generated planetary system. The user is allowed to fly through this camera using a freely moving camera that can be turned and accelerated like a spaceship.

Textures are generated in program using Brownian motion.

1. **User Interface**

Implementation Technique



Space – Stop Forward Motion

W – Accelerate Forward

S – Accelerate Backward

D – Yaw Right

A – Yaw Left

Mouse Move Up – Pitch Up

Mouse Move Down – Pitch Down

Mouse Move Left – Roll Left

Mouse Move Right – Roll Right

Mouse Motion Over the window is used for looking up/down and rolling

The mouse triggers movement as long as it is not located in a dead-zone in the center of the window.

1. Camera and Planetary Movements

Pitch was implemented by taking the cross product of the look direction and up vectors, and rotating about the resultant axis.

Roll was implemented by rotating around the look direction vector for the camera.

These simple operations, when combined with the mouse created a very intuitive user control scheme.

1. Planetary Movement

Planets in this simulation move in a manner that approximates Keplerian motion. Since our simulation does not incorporate mass and gravity, motion is along an ellipse following the equal time equal areas principle from Kepler’s second law. Precisely calculating the areas would require expensive integration, so I instead calculate areas using trigonometry. This introduces some error as I end up underestimating the area of each slice, but it results in motion that shows planets slowing as they move further from their orbital focus, and faster when they move closer.

1. Texturing and Planet Appearance