

Final Project
CMP 161/L
Visualization and Computer Animation
Winter 2011
Hosh Tachmuradov
htachmur@ucsc.edu

Space Ship Animation



Overview

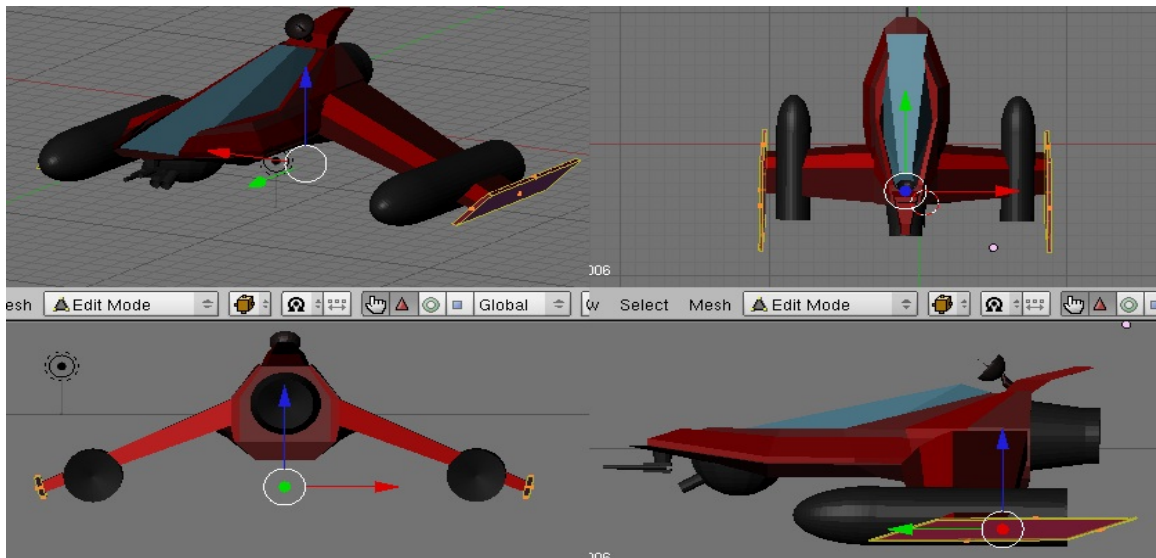
For my final project I made a short animation that will be added as a cut scene to the game I am currently working on. The animation is consisting of a space ship that is flying above terrain with a smoke coming from the left wing.

As one of the goals, I am planning to make movement of the ship to be as long as the user is using the application, i.e. regenerate the terrain below the ship. In addition, I will provide a user capability to switch between different view angles and change the speed of the ship movement. Finally, I am planning to animate vertex core behind the ship.

Technical Details

Loading 3D Model

First of all, I have uploaded wavefront OBJ file of a 3D model of a space ship. To load the model I used Nate Robins glm.h module that helped me to generate smooth normals with preservation of edges, weld redundant vertices & texture coordinate generation (spheremap and planar projections). Also the OBJ file loader was written in 1997, I found it very useful, easy to use and did not have any problems at that moment.



The ship model is centered at the origin.

Generating Terrain

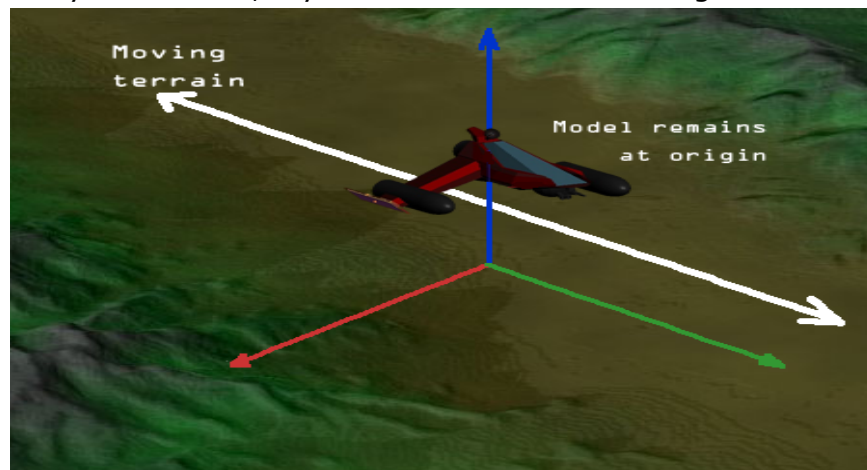
After I had the model loaded, I generated terrain. To do so, I used height field map and appropriate texture to procedurally generate terrain. I had a 2D image that I used to texture my terrain that I mapped over the square that is 512 pixels wide and 512 pixels tall. To determine the height of each vertex, I scanned and analyzed green channel pixel of a map texture. I have processed the map and generated the terrain in initialization and stored the result in gList object.

As one of the goals of this project I wanted to make this terrain unlimitedly long. To do so I have created 3 terrain plates that could be looped during the animation without notice of the user.

Since my height map generation process took place in initialization stage, I created one more gList object where I stored a "mirrored reflection" of the previous map. By "mirrored reflection", I meant that the end of the first terrain is the beginning of the second terrain.

In the process of animation, I drew 3 lists, where first and last list are identical and the middle one is reflection of them. During the animation I interchange two generated textures to produce continuous terrain.

As I previously mentioned, my model remains at the origin and terrain is



animated.

Lighting

In addition, I have used lighting from two different positions to make sure terrain was rendered nicely and is visually appealing.

This part was not hard to implement, but I have to admit that it was time consuming.

Camera angle

The user of the animation can change the camera's viewing.

- Key 1: view from top down (far clipping)
- Key 2: view from behind of the ship
- Key 3: view from the left in 45 degree angle
- Key 4: view from top down (close up)

Speed of the ships movement

The viewer of the animation also can control the speed of the ship.

- Key F: speed up the movement of the ship
- Key S: slow down the movement of the ship

Particle Engine

I have integrated a simple smoke particle animation to show that the space ship is damaged. The particles initial position is the left wing of the ship. Particles move with a random velocities in X and Y axes. Movement in Z-axis (behind the ship) is influenced by the speed of the ship (faster the ship faster the particles move to the back). Particles lose opacity as the life is increases



Vertex Core



In addition I have made an attempt to animate a vertex code behind the space ship. Vertex core appears only when the speed of the ship is slow.

Conclusion:

After viewing a resulting animation, I came to conclusion that I successfully combined different 3D animation techniques and result was satisfying. I have to admit that I am not very satisfied with vertex core animation and believe that it could be done better if I had little more time.

References:

During the development, I have used the following materials:

- glm.h (<http://www.cs.jhu.edu/~subodh/457/code/gltutor/glm.h>)
- Visualizing Vector Volume Data (http://www.kxcad.net/cae_MATLAB/techdoc/visualize/f5-3495.html)
- Flow and Particle Data Visualization (<http://vidi.cs.ucdavis.edu/research/flowvis>)
- A Novel Approach To Vortex Core Region Detection (http://www.simcenter.msstate.edu/publications/2002/Public/DSThompson_2002_vortex_detect.pdf)