Ryan Lovrien
Phelps Williams
31-Mar-2008

CS569 Final Project: TRON

The final project will consist of a game modeled after the Disney movie TRON. The game will essentially be a variant of the game Snake. A ground plane will be displayed with one human and one AI character. The character will be displayed as a motorcycle type vehicle modeled from primitive shapes. The two vehicles will travel at a constant speed either continuing straight, or turning left or right. A trail will be left behind each vehicle which either user could crash into later. The objective of the game is to trap the other character and cause them to collide with a trail or the edge of the map.

Features:

A 3rd person perspective will be used to show off the vehicle model and provide a wider view of the environment.

Shader development will occur including a reflective shaded ground plane. Normal maps could be used to add interesting texture to the wall structure drawn behind each vehicle. Using a normal map for the ground plane might also provide an interesting visual effect.

A wall is drawn behind each character as they move at a constant speed. Collision detection will be used to determine when either the boundary of the environment (map) is reached or the character intersects a wall.

A particle system will be used to animate the game ending collision between a character and a vehicle trail, this would include explosions and possibly a smoke effect. The point of emission of the wall structure can be animated with a sparkling effect. The vehicle could also have a smoke exhaust effect.

Another pair of interesting ideas should time allow might be a map rendered to a quadrangle in view of the user with a texture provided from a FBO. The scene will be drawn from a static camera looking down on the map and this perspective will be drawn to the FBO. A second similar approach will be to use a camera looking in reverse providing the user with a backwards view. This camera would be rendered to a FBO and then to a quadrangle static in the users view space.

A few special features might be possible including the ability to make your walls turn to “glass” requiring a glass shader. Another possibility is reflective glass windows on character vehicles. If time allows, the vehicle will be animated with spinning wheels, vehicle movement jitter, and possibly other animations.

Architecture:

We are planning on extending the software framework used in the earlier projects. A control interface will need development taking keystrokes from the user and translating them to motion in the game environment.

A spatial data structure will be used to efficiently cull walls outside of the view frustum of a character. Because the vehicle trails are created as the game progresses an efficient approach to managing these elements will be necessary to prevent the game from slowing. The ground plane will be drawn from a plane primitive while vehicle trails will be drawn with scaled boxes. When the user turns a new box element will be created. Otherwise the box will continuously be scaled.

The Character type will contain a mesh hierarchy representing the vehicle, a list of walls, and a controller. The controller interface will be implemented by both the keyboard based user interface and the simple AI element.