William Scott (wrs35)
SN: 11876177
COSC363 Computer Graphics - Assignment 1: A Robot’s World

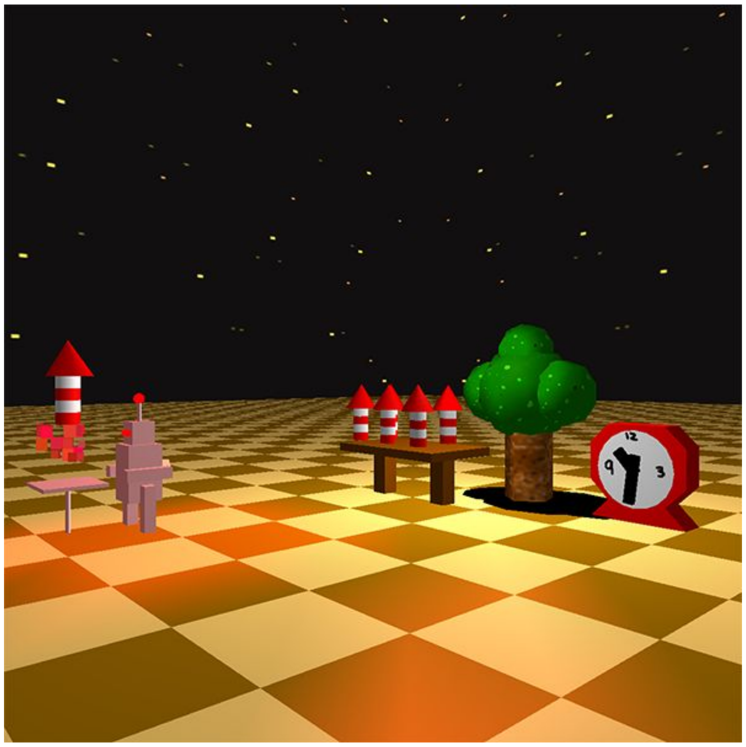
Description

Compiling the Project - From Source Folder

```
g++ -o Cosc363Assignment1 COSC363-Assignment1.cpp Camera.cpp Clock.cpp Ground.cpp
Launchpad.cpp Object.cpp Robot.cpp Skybox.cpp Tree.cpp Workbench.cpp -lm -lGL -lGLU -lglut
```

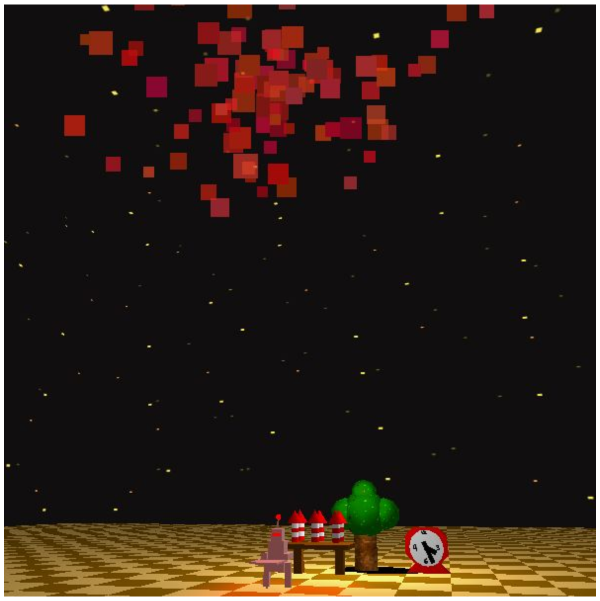
The scene I’ve created features a robot with a red antenna light who works in a loop moving fireworks from a table to a launchpad, and setting them off beneath a starry sky. On their way up the fireworks eject flames, and when they reach their peak the fireworks explode into a beautiful rose-pink display. Near the resupply table for the fireworks, a shady tree can be seen swaying in the wind, and next to the tree a large clock with moving hands can be found.

Important Aspects



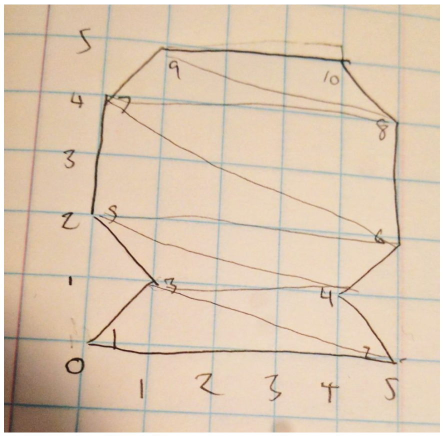
The firework launching with flame particles trailing behind, shady tree and large clock seen on the right hand side. The clock and tree feature continuous animation; and the tree casts a planar shadow.

William Scott (wrs35)
SN: 11876177
COSC363 Computer Graphics - Assignment 1: A Robot's World


The firework (Particle System) exploding overhead the scene.

Extra Features

- Planar Shadows are cast by the 'Tree' This can be found in Tree::draw(void).
- A red spot light is attached to the robot which moves around the seen, the movement can be seen on the ground below it. This light is enabled in setupLighting() and moves in Robot::drawHead(void).
    - The 'Clock' is a custom-built model using vertex coordinates and polygon definitions. The clock was designed on graph paper and stored within the '.off' file format.



Sketch of 'Clock': Used for determining how to create the model

- A skybox has been implemented to display a starry night sky. This is handled by the function Skybox::draw(void).
- Particle systems have been implemented for both the ascent, and explosion of the 'Firework'. Drawing of the particles is handled by Robot::drawFireworkExplosion(void) and Robot::drawFireworkFire(void).

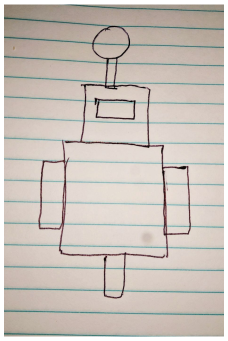
William Scott (wrs35)
SN: 11876177
COSC363 Computer Graphics - Assignment 1: A Robot's World

Models

All models and textures used in the scene were created by myself.

The Robot

The Robot is made out of GLUT/GLU objects. A major difference between the final Robot and the sketch shown below was the decision to change to legs and include a walking animation. The robot moves a red spotlight with it, and features a continuously animating wind-up key on its back.



Early sketch of The Robot

Table

The Table is made out of GLUT/GLU objects. This is where the Robot comes to resupply for another firework after a successful launch.

Firework

The Firework is made out of GLUT/GLU objects. This is what the Robot moves within the scene. It also has an ascent animation with fire particles, which transitions into a firework particle explosion.

Launch Pad

The Launch Pad is made out of GLUT/GLU objects. This is where the Robot moves the firework for launching.

Shady Tree

A single shady tree made out of GLUT/GLU objects. The tree is textured with Grass.tga and Wood.tga. The Shady tree features a continuous swaying animation, and casts a planar shadow.

Clock

The clock is made of a GLU Disk with the texture ClockFace.tga, and a body made from vertex coordinates loaded through the '.off' file format. The body is textured with ClockBody.tga. The clock also features a continuous animation with its quick moving "minute" and "hour" hands.

Special Challenges

Making the particle systems was a fun extra feature which required some new thought and experimenting within OpenGL. A problem I ran into was getting everything to animate smoothly, and be able to draw the particles as required. Ultimately, I solved this problem by moving all of the firework carrying and particle systems into Robot.cpp and piggybacking off of the function Robot::draw(). If I revisit this project in the future, I would work to split the firework and particle system logic out of Robot and into their own classes.

Controls

Arrow Left: Look Left
Arrow Right: Look Right
Arrow Up: Move Forward
Arrow Down: Move Backward
Page Up: Look Up
Page Down: Look Down

Resources & References

- loadTGA.h - R. Mukundan, Department of Computer Science and Software Engineering University of Canterbury, Christchurch, New Zealand.
- loadBMP.h - R. Mukundan, Department of Computer Science and Software Engineering University of Canterbury, Christchurch, New Zealand.
- void loadMeshFile(const char* fname) and void normal(int tindx) from Lab 2's Cannon.cpp
- Modified version of void floor() from Lab 3's Train.cpp
- COSC363 Computer Graphics 2017 Lecture Notes, and provided Lab Code
- OpenGL® 2.1, GLX, and GLU Reference Pages
  https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/
- Transparency, Translucency, and Blending
  https://www.opengl.org/archives/resources/faq/technical/transparency.htm
- Particles / Instancing
  http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/