**CS4621 Final Project Report**

**David Landy (drl224), Isabella Wang (isw8)**

**Goal**

Animated hallway scene with projectile shooting features.

**What did we achieve**

We created the scene with original textures, shaders, and models. We have a modified FPS camera that can look around and shoot projectiles in the direction of view. Projectiles emit light and affect the shading of the environment around you.

**What we got off the web**

Fire texture, carpet texture. All other textures and models we created ourselves. Code we implemented ourselves.

**What we implemented and have working**

* Scene
* FPS Camera
* Camera Movement Path
* Projectiles
* Light Sources Mapped to Projectiles
* Projectiles Collide with Floor

1. **Scene**

Over 40 meshes and 5 textures, created in Blender. 6 different shaders of Fire, Phong, Phong with textures, and materials, and 4 light sources, added by hand into the scene xml.

1. **FPS Camera**

X rotation was unchanged from fly mode rotation, Z rotation was thrown out. Y rotation required using a separate axis – the world space Y axis worked well.

1. **Camera Movement Path**

Camera follows a fixed path of transformations around the scene – see next section for extra details.

1. **Projectiles**

Created Projectile object class and implemented multiple “spawning” to the scene by having a pool 5 that just stay hidden and in place whenever not in use. Projectiles have an age and max life span. Each projectile contains method to control movement and visibility. Included two types: shoot/shot-forward, toss. Shot projectiles move linearly in the direction of camera view, and can be “yo-yo’ed”: recalled and thrown again. Tossed projectiles are given a given an initial velocity and follow a simplified model of gravity.

1. **Light Sources Mapped to Projectiles**

Projectiles are linked to an object mesh and a light source to implement the effect of moving balls of light that affect the shading of the environment. In particular, 5 projectiles were connected to different colored Disco Bunnies which can be thrown independently.

1. **Projectiles Collide with Floor**

Checked whether any of the meshes mapped to a projectile fell to the floor (low y-coord).

**What we tried implementing and did not use**

1. **Mouse Camera Control**

Modified viewing mechanism to follow the mouse. Camera rotates based on mouse cursor displacement, but mouse remains snapped to center of the camera.

Was more complicated than we anticipated to accomplish smoothly, so we decided it wasn’t worth it.

1. **Spline Governed Camera Movement**

Tried experimenting with the existing framework of tessellated Bezier splines, but behavior was very cryptic and we decided to move onto other endeavours.

**What we did not implement**

1. **Move Shader, Models, Texture:** Scene could have been more impressive being larger and containing more objects, but we ran out of time.
2. **Full Collision Detection:** Began implementing this with floor detection, but did not have time to extend detection to all other geometry.

**What we learned**

* **Best thing learned:** Blender is super powerful! Had to deal with working with quads and converting everything to meshes. But learned to unwrap textures and a lot more about mapping UV coordinates. Learned that you can bake translations into objects and position in the scene (but didn’t do that). And learned how to actually model and deal with normals.
* **Gotcha’s:** Modifying the camera to do what we wanted was a lot more finicky than we expected. Keeping the mouse snapped to the center of the screen often resulted in either no more mouse reactions, or arbitrary rotations all over the place.

**Effort**

Early stages (first week) focused on planning the scene: importing models, textures, exploring the limitations (learning blender). Next week spent actually composing the scene: making models, mapping textures, developing the xml. Intermittently spent a few days figuring out the FPS camera. In the last week completed the scene and attempted to implement mouse control viewing and spline governed camera paths, as well as adding projectiles.

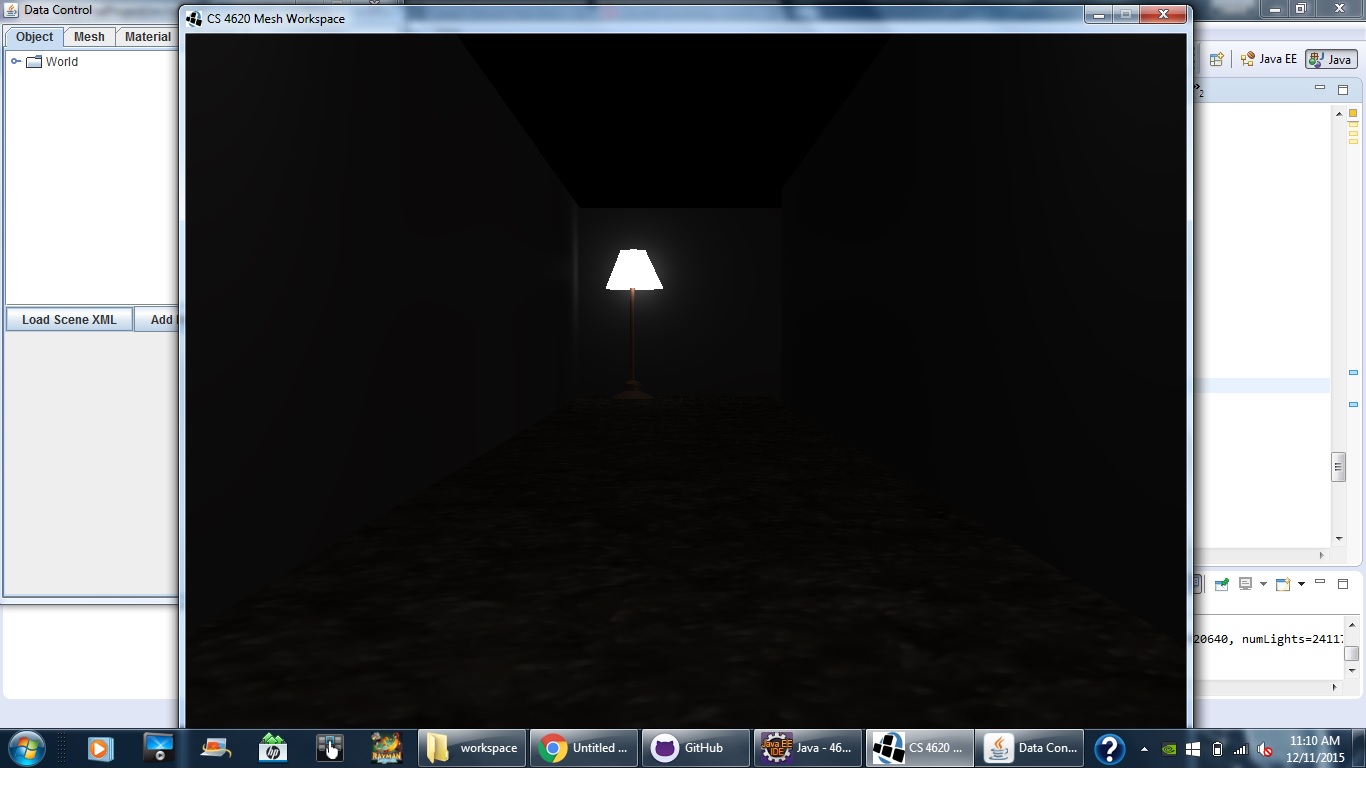


Figure 1: Startup Scene



Figure 2: Disco bunnies on the floor

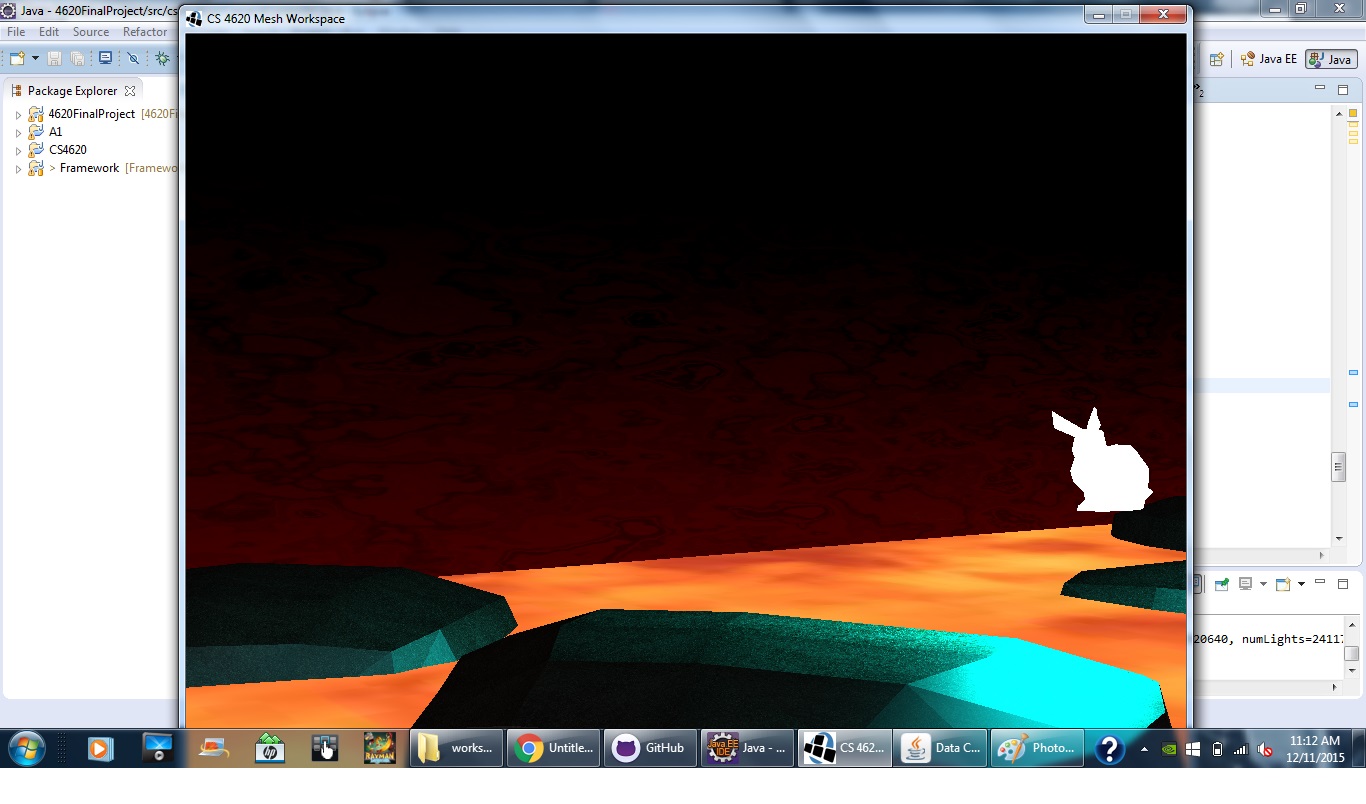


Figure 3: Lava room with Disco Bunny mid-flight

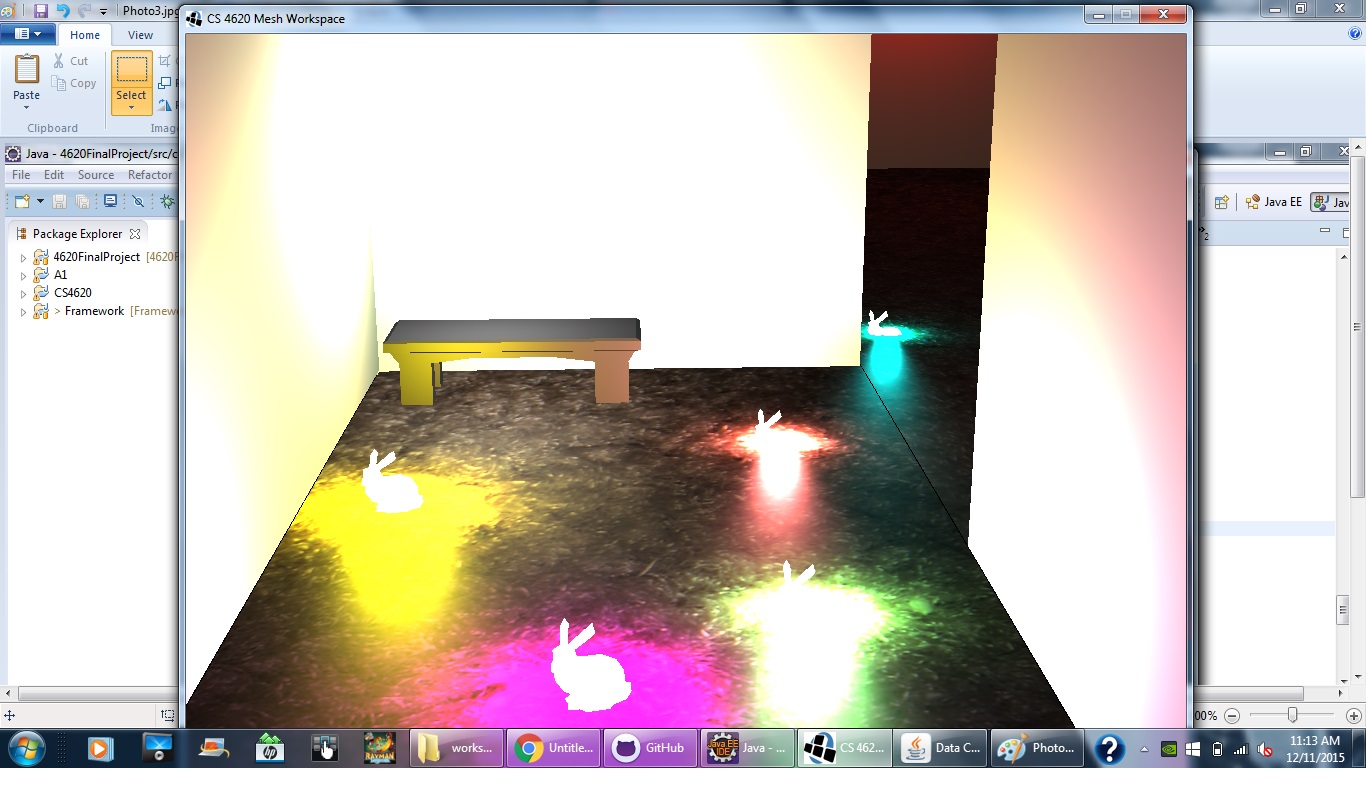


Figure 4: Final position with Disco Bunnies and Bench Mesh.