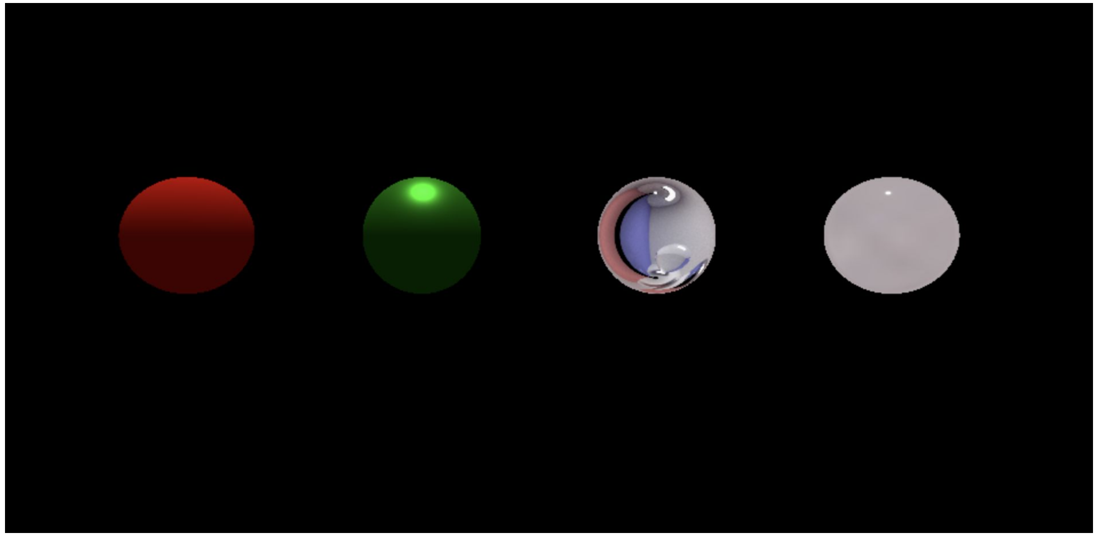
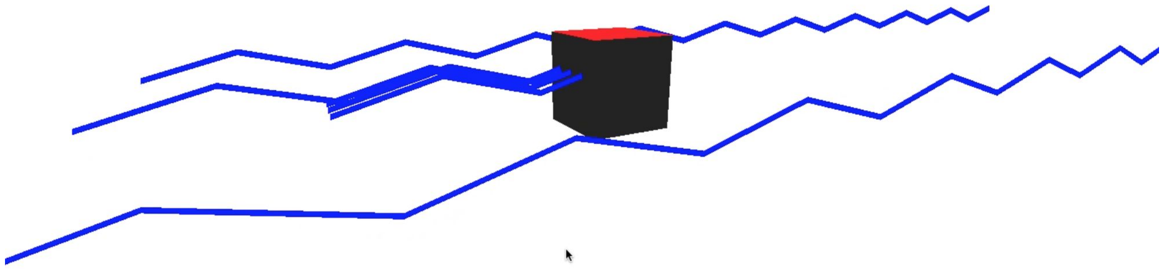
Implementing different materials



- THREE.js has abstractions for most materials you can think of, just need to adjust parameters
- Leftmost is "LambertMaterial", 3 right are all "PhysicalMaterials" with different parameters
  - Potential issue: there could be other materials + parameters that can achieve the same visual effects, so need to exercise caution with testing for material intersections down the line
- Next steps: Proper background texture, mipmap, more realistic reflection/refraction (THREE documentation has really good examples). And then, making the lasers work with these materials (THREE has methods for intersections and etc. Our challenge is the laser reflection/refraction mechanics)

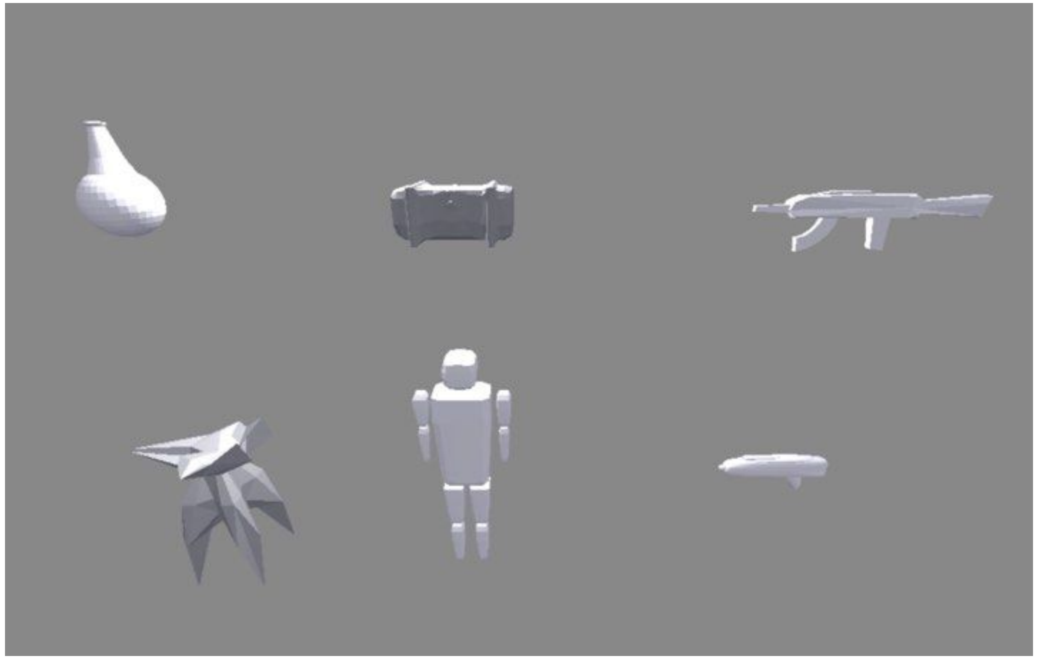
Raygun Logic



(Above) is a demonstration of raygun shot/enemy intersection

- Our core gameplay revolves heavily on the use of a raygun.
- The logic of this raygun's shots is dictated by the principles we learned in raytracing.
- For example each of the enemies in the game will have a different material layered over their model. When a ray hits an enemy it will be reflected according to that particular models respective materials brdf.
- We currently have implemented raygun shot/enemy intersection, and controls to fire the raygun
- NEXT STEPS: Upon merging our code together we will begin working on reflections according to BRDFs

Object Models



- 3D object models for the main interactable elements for the game scene (enemies, weapons, etc)
- Modeled in Blender and loaded into WebGL using THREE.js collada object loaders
- Next Steps: application of textures and materials, interaction with raytracing collisions and gameplay mechanics