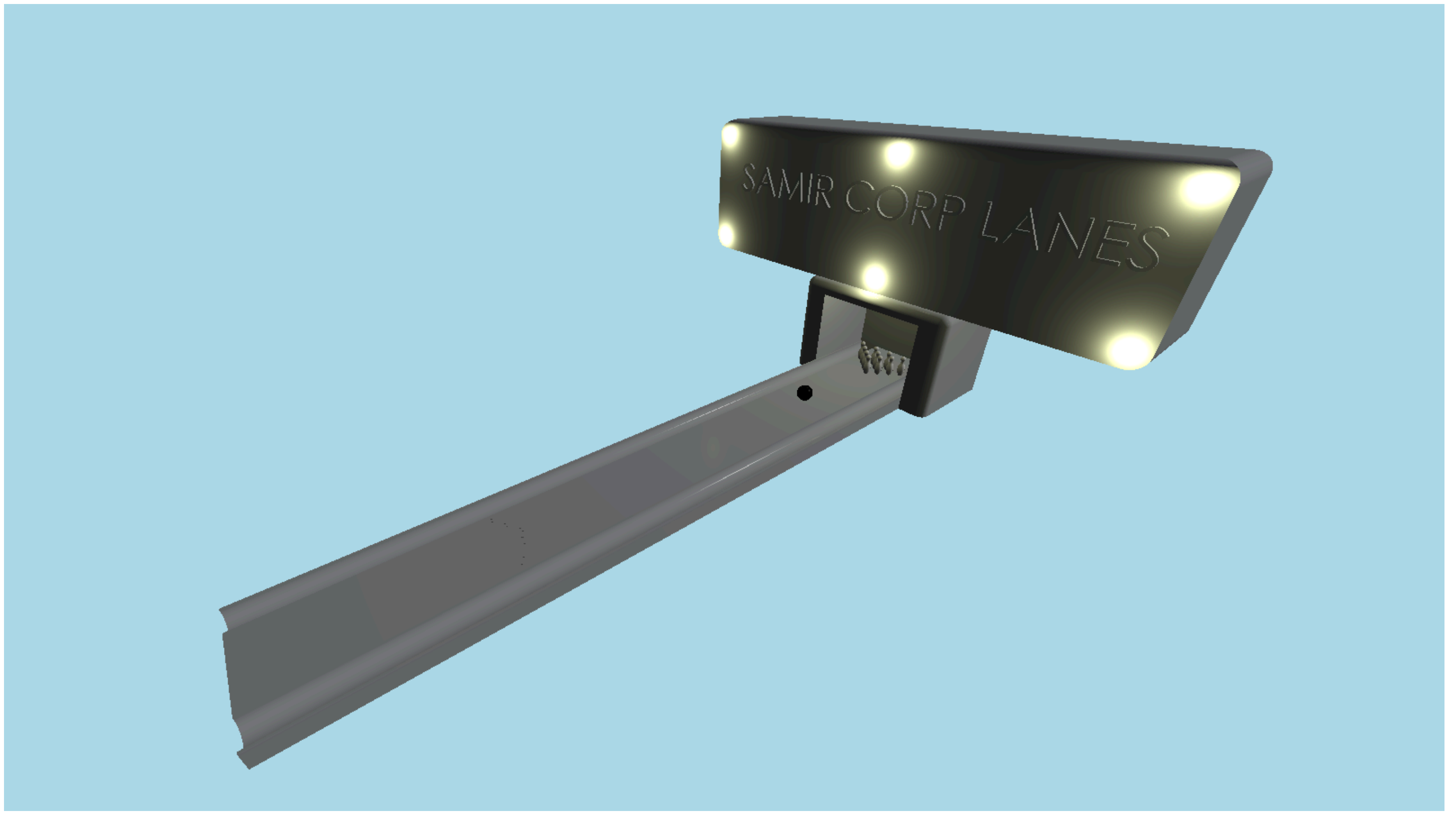
3DMODEL FINAL PROJECT DOCUMENTATION

CRUZ, Renz & YU, Cedric Apr 8, 2024



Paste screenshot(s) of your project on the space above

I. PROJECT DESCRIPTION

The project is a self playing bowling game made using three.js and ammo.js. Three.js was used for the handling of the visual elements of the project and its 3D models which were loaded from STL file types. Ammo.js is the physics engine for the underlying simulation where an impulse is induced to a bowling ball and upon impact with the pins, collisions are animated and the user will see “pin action” in bowling terms. The project rolls the ball at different angles into the pins to visualize which bowling approach is best to get strikes.

The project was chosen because it is a fun sport to mess around with and the interaction of the ball with the pins make for a satisfying experience. Concepts learned from the course are also highly applicable such as 3D modeling, camera positioning, lighting used on the lane sign, use of mesh materials and textures implemented to improve the visual quality of the project.

II. HOW TO RUN YOUR PROGRAM

The project needs to be run on a localhost environment following the default installation guide of three.js which is to have three.js installed through npm which is another requirement. Follow the steps below to run the project:

1. Open the directory of the project in the CMD
2. Run both these commands in CMD to install the dependencies needed. “npm install three” “npm install ammo.js” “npm install –save-dev vite”
3. Type “cd src/” in CMD to navigate to the src folder that contains the code
4. Now, in CMD type “npm start”
5. In CMD, there should be a localhost link that you can perform CTRL+Click on to open the URL in a browser.

III. CONCEPTS YOU APPLIED

Lighting

Directional and Ambient lighting were used in the project to simulate a basic bowling alley atmosphere. Ambient light was used to illuminate the scene to a preferred light level. Shadows were created by having a Directional light in the positive x, y, and z direction. The signboard was illuminated using Point lights that were adjusted to have a yellowish hue. Six of these Point lights were placed on the signboard.

Camera and Perspective

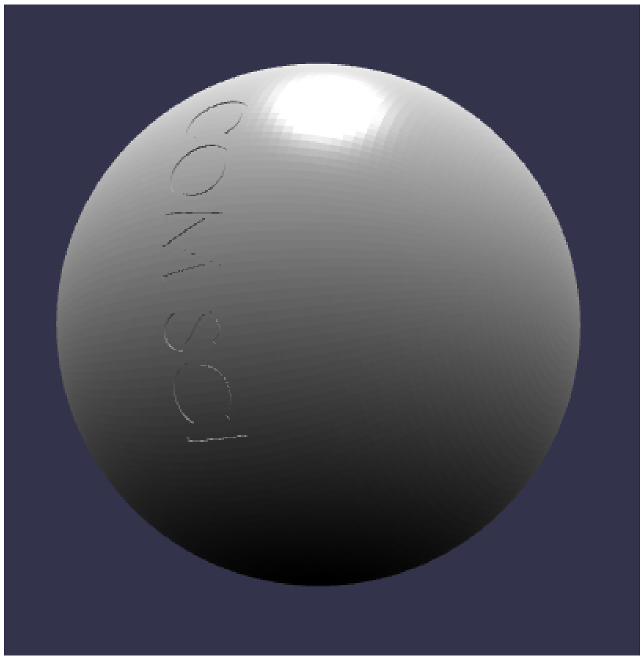
The camera is implemented in the program through the use of a Perspective Camera with the near plane and far plane set to 0.1 and 1000 respectively. The camera is positioned at (-35, 45, 0) with its lookAt value set to look at the point (100, 0, 0) which is sufficient to view the entire scene. The camera also has orbital controls that enable viewing at multiple angles.

Textures and Materials

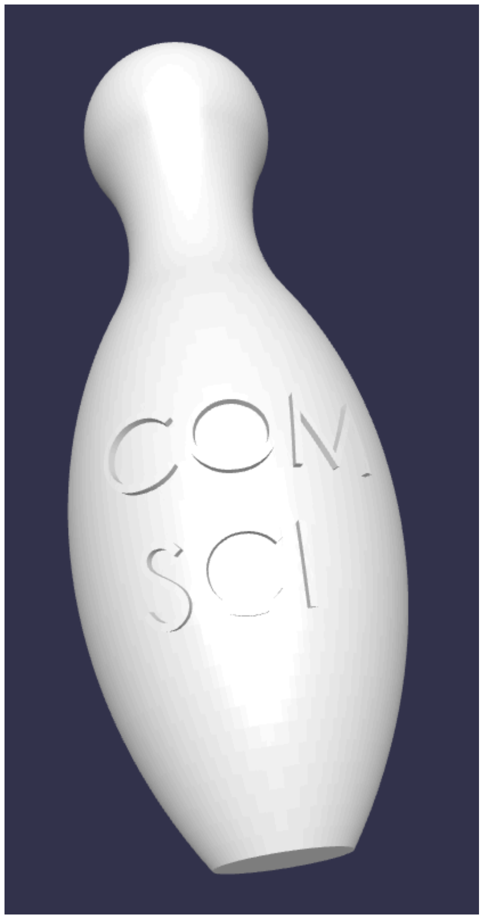
All of the textures have some form of specular reflection since the bowling ball, pins, and bowling lane are known to be polished surfaces. With those considerations in mind, Phong material meshes were used for all the models. These models were colored to preference together with their shininess values.

Modeling

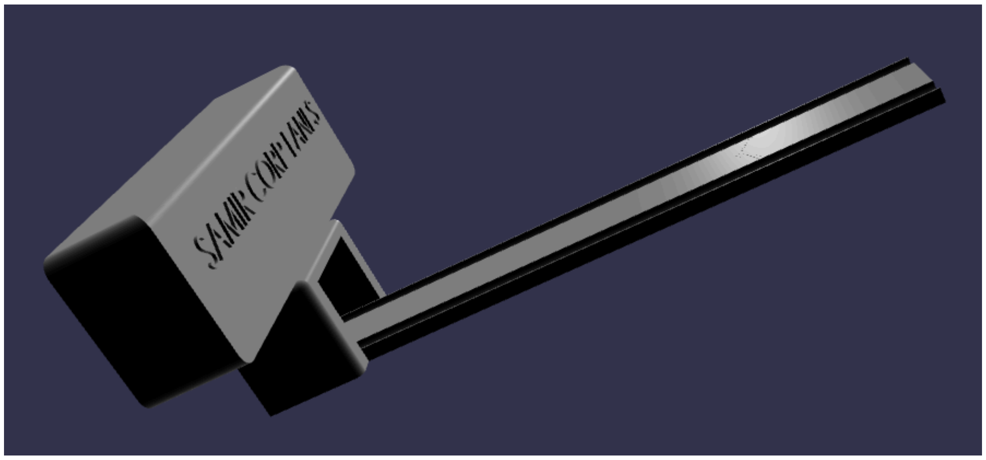
Assets used in the project were modelled using Solidworks and exported as STL filetype. The said files were then imported into three.js using the STLLoader. We were able to apply our knowledge in modeling transformation and made use of 3D primitives such as sphere, cylinder, plane, and cube. The images below are the models built from the modification of the 3D Primitives used in the program:



Bowling Ball



Bowling Pin



Bowling Lane

IV. RESOURCES THAT YOU BORROWED FROM THE PUBLIC DOMAIN

Ammo JS Library (n.d.). Github. Retrieved from https://github.com/kripken/ammo.js

Bowling by iliagrigorevdev (n.d). Github. Retrieved from https://github.com/iliagrigorevdev/bowling/tree/master/res

V. REFERENCE

Threejs Docs (n.d.). threejs. Retrieved from https://threejs.org/docs/

Ammo JS Github Repository (n.d.). Github. Retrieved from https://github.com/kripken/ammo.js