Campus 2059

A Vision of the Future
Birmingham University
Campus

Sam Black
524689

Librainians group
EE4A

Table of Contents

1 Abstract

This paper covers the work done to implement a monorail simulation; the construction, texturing and lighting of the train, stations and track using Blender, and adding animations linked to Blender's game engine logic blocks for interactive use.

The report also advises on how to present the simulation and allow access to users, and concludes that the work done could be improved by finishing the proposed agent activities and suggests extensions to the work.

2 Introduction

Campus 2059 is a project that imagines how the campus will have evolved 50 years in the future.

Fifty years of development would include improvements to teaching methods, leisure activities, transportation, robotics, communication, society and the environment.

To reflect these changes, an interactive 3D model of the campus is to be constructed, demonstrating how things will have changed in the future.

The Librainian's group envisaged a campus that was protected from a nuclear blast by a dome encompassing the central campus area; the rest of the campus was devastated by the blast and rendered uninhabitable. The dome includes an artificial sun, moon and starry sky to give the impression of being outdoors.

The teaching method has also altered; instead of going to the library to get a book, the library now stores brains of professors who had worked at the University previously, interacting with their students via a robotic carapace. The brains are looked after by robotic helpers that also act as security for the central brain, powered by geothermal energy from the spring directly below it in the library.

Transportation across the dome and to other domes is via monorail, and this is the subject of the rest of this report.

3 Model Design

The monorail system consists of 3 major parts; the train, station and track.

The train is a simple cylinder shape with a grove for the track, capped with a modified cone for the driving ends. The seats are a NURBS curve exported to mesh.

The carriage is two cylinder meshes inside one another (Figure 1), allowing the glass layer and doors to be sandwiched between for added realism. The separate layers also allows differing materials and textures to be used, with their normals set correctly to reflect light, cast shadows and react to collisions in the game engine.

The outer cylinder had the groove made for the track by moving the lower faces up and deleting the remaining curved section; this was replaced by extruding a flat edge. The inner cylinder's floor was also made by deleting the lower faces and extruding an edge to cover the gap.

Each cylinder was extruded along its length for each door, window or pillar (Figure 2); segements were then deleted to create the apertures for the doors or windows. At the base of the inner cylinder's door apertures, the floor was extruded and rotated to join with the outer cylinder, covering the gap between the cylinders.

The materials for the inner and outer carriage layers were default setting materials with differing colours; the glass layer is set to be transparent using the ZTransp option; raytracing is also enabled, but will not work in the game engine.

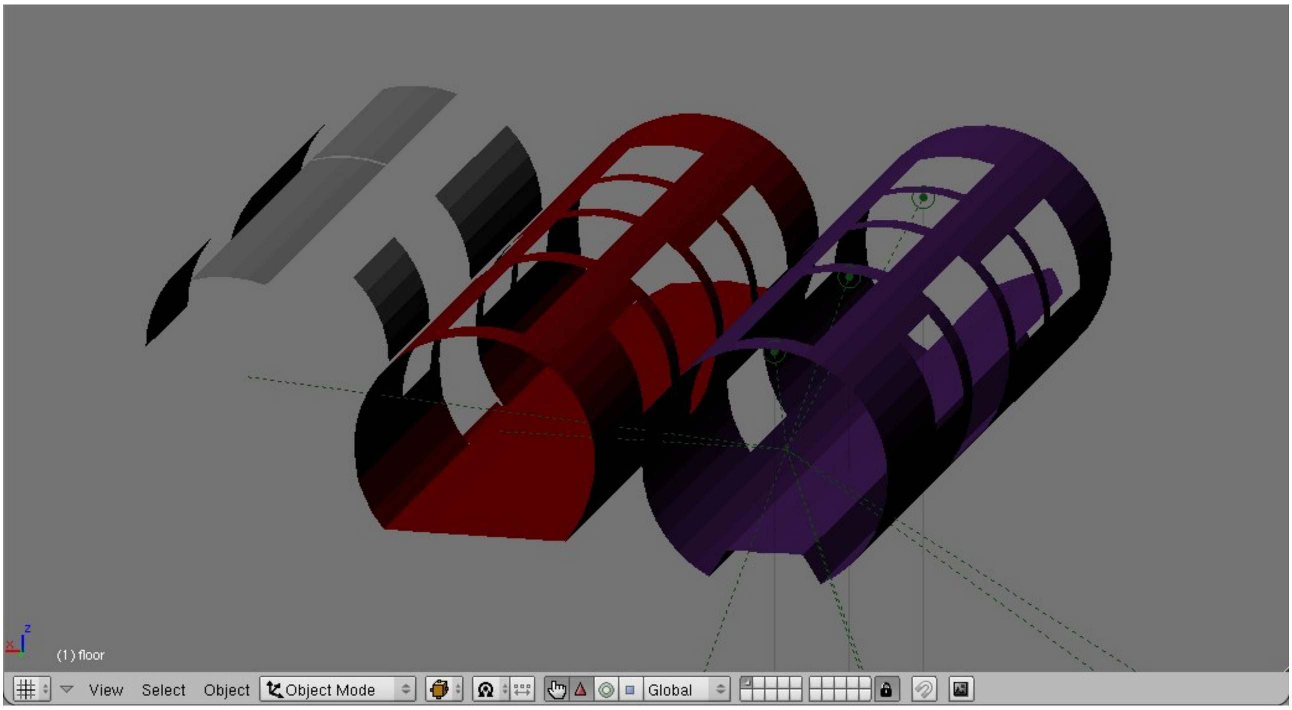


Figure 1: Carriage Layers: left to right: glass, inner and outer

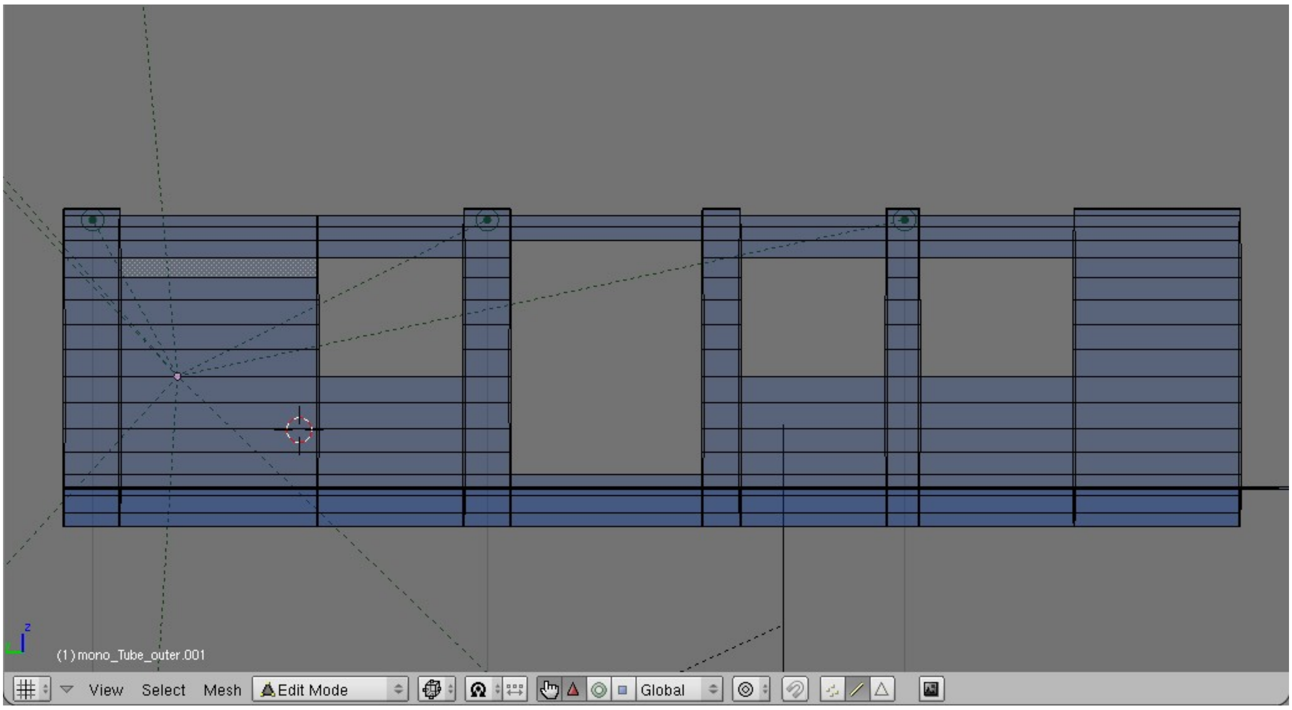


Figure 2: Cylinder extrusions for door, windows and pillars

The nose cone is two cone meshes, inner and outer like the carriage, that have been modified to include the groove at the front and aperture for the forward window (Figure 3). The inner and outer materials are the same as for the carriage body.

The seats were NURBS curves exported as a mesh; the materials were again the default settings with the colour changed.
The entire carriage can be seen in Figure 4.

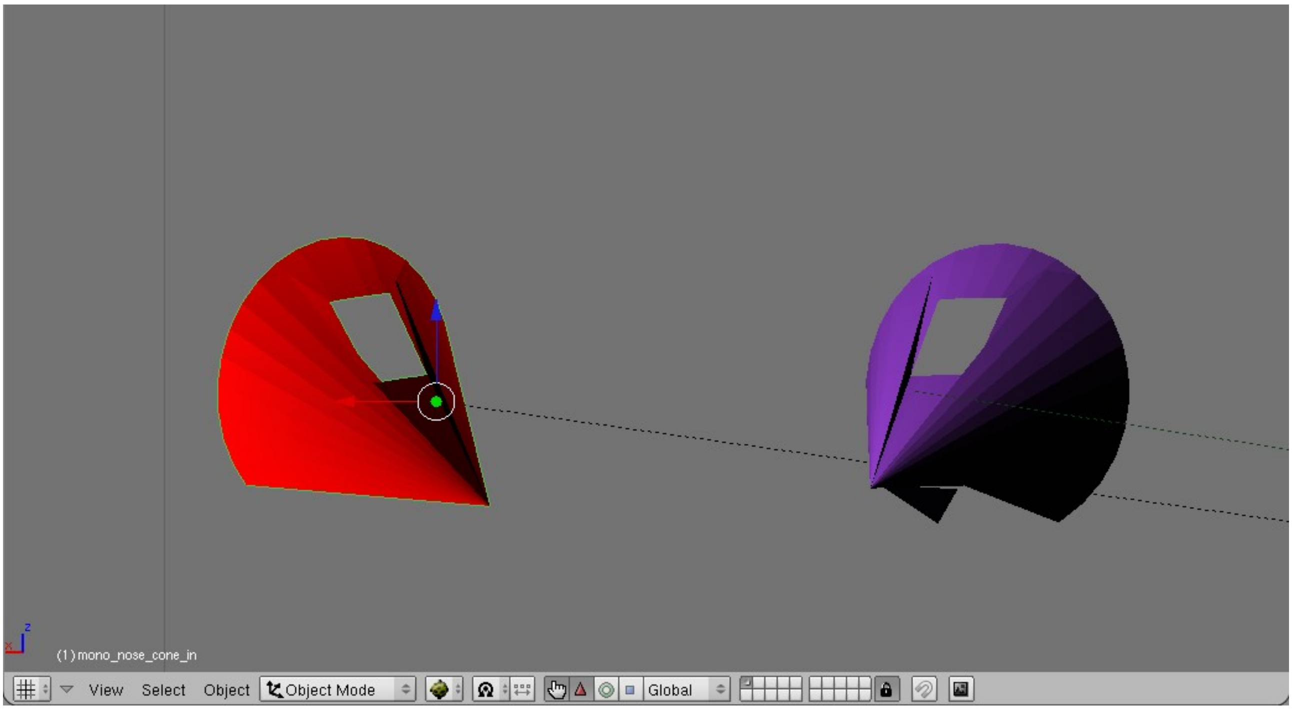


Figure 3: Nose Cones: inner (left) and outer

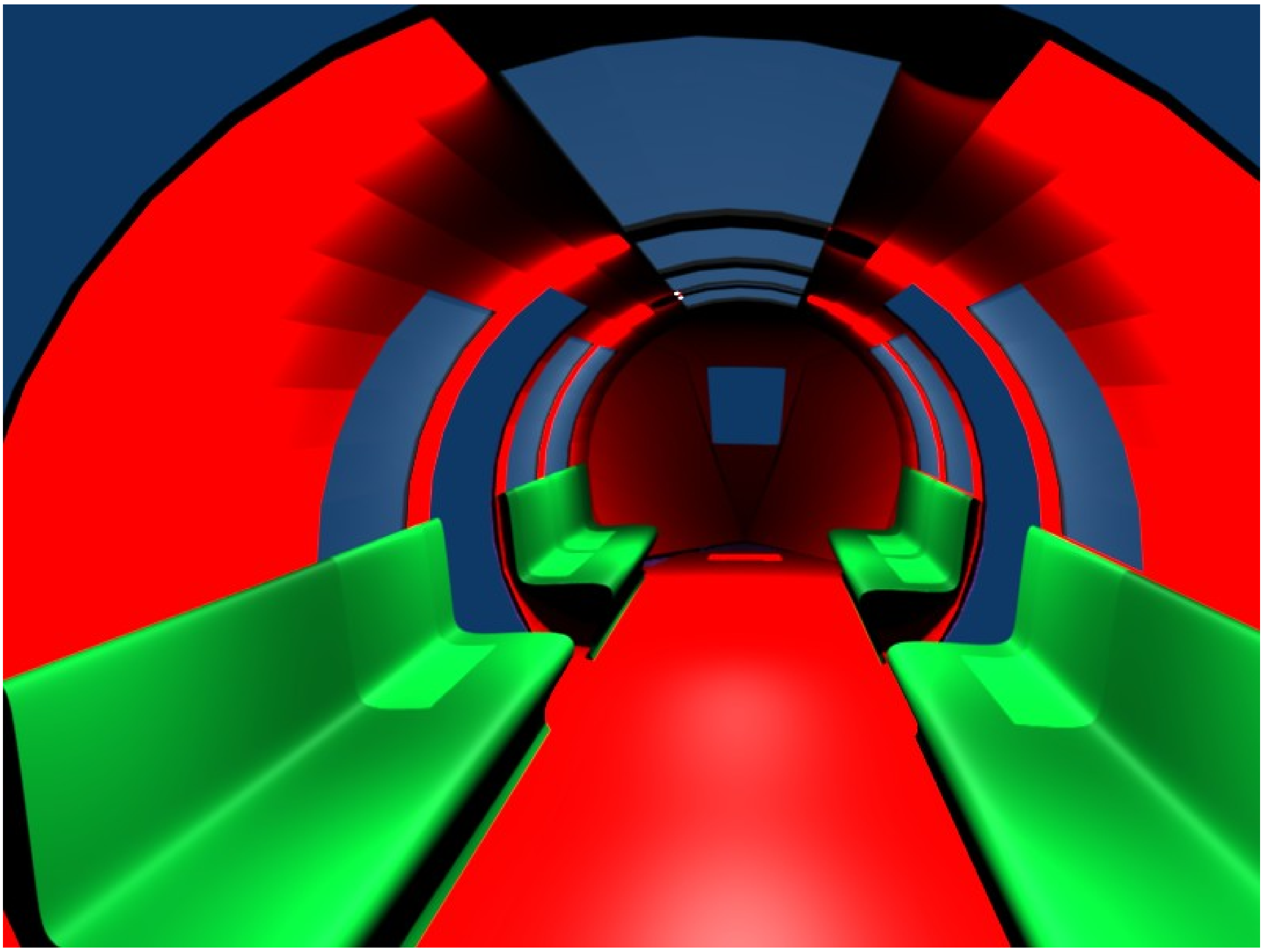
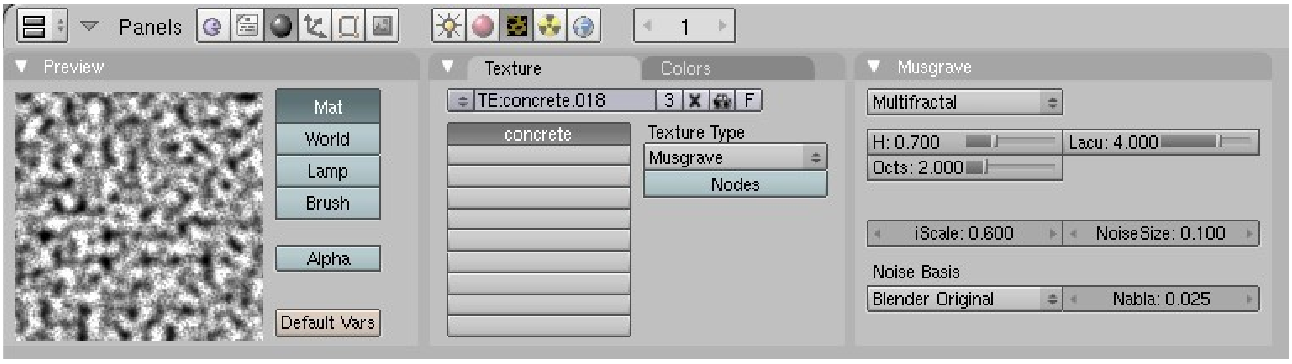


Figure 4: Mono rail train complete

The station is composed of the platforms, lifts and ramps.

All of the station structure is textured with a concrete material; the texture is a Musgrace multifractal (Figure 5) on a grey material, giving the texture seen in Figure 6.



TE:concrete.018 3 F

concrete

Texture Type: Musgrave

Multifractal

H: 0.700

Lacu: 4.000

Octs: 2.000

iScale: 0.600

NoiseSize: 0.100

Noise Basis: Blender Original

Nabla: 0.025

Figure 5: Concrete settings

Figure 6: Concrete texture preview

The station was constructed out of cube meshes merged into a single object (Figure 7); this removed a bug in the game engine where the fences (orange in Figure 7) would move to be above the platform level.

The lift is assembled using a cylinder as the outer skin, a cube mesh for the inner shaft, a simple circle mesh for the roof and a cube mesh for the lift car. The outer, inner and car mesh faces were subdivided and some faces deleted to allow entry and exit from the lift.

The ramps are a cube that was extruded to make one half of the ramp, which was duplicated, mirrored and had it's normals fixed to be outside correctly, before both halves were merged to be one object (Figure 8).

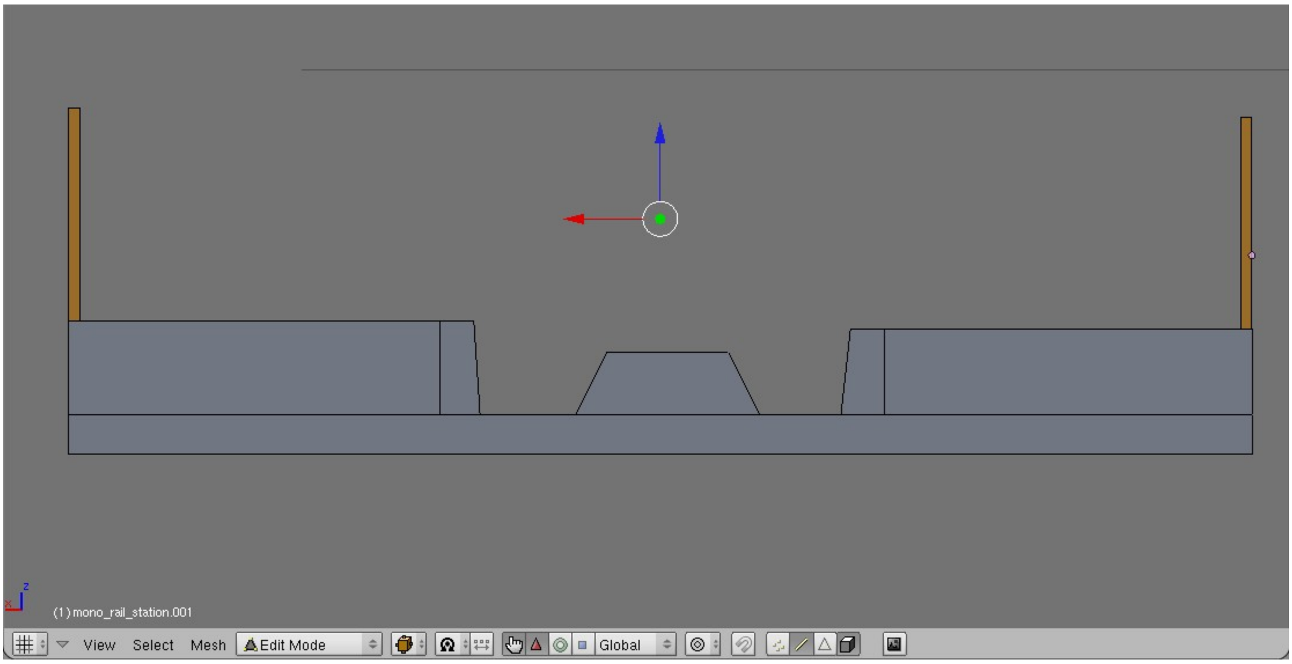


Figure 7: Station platform

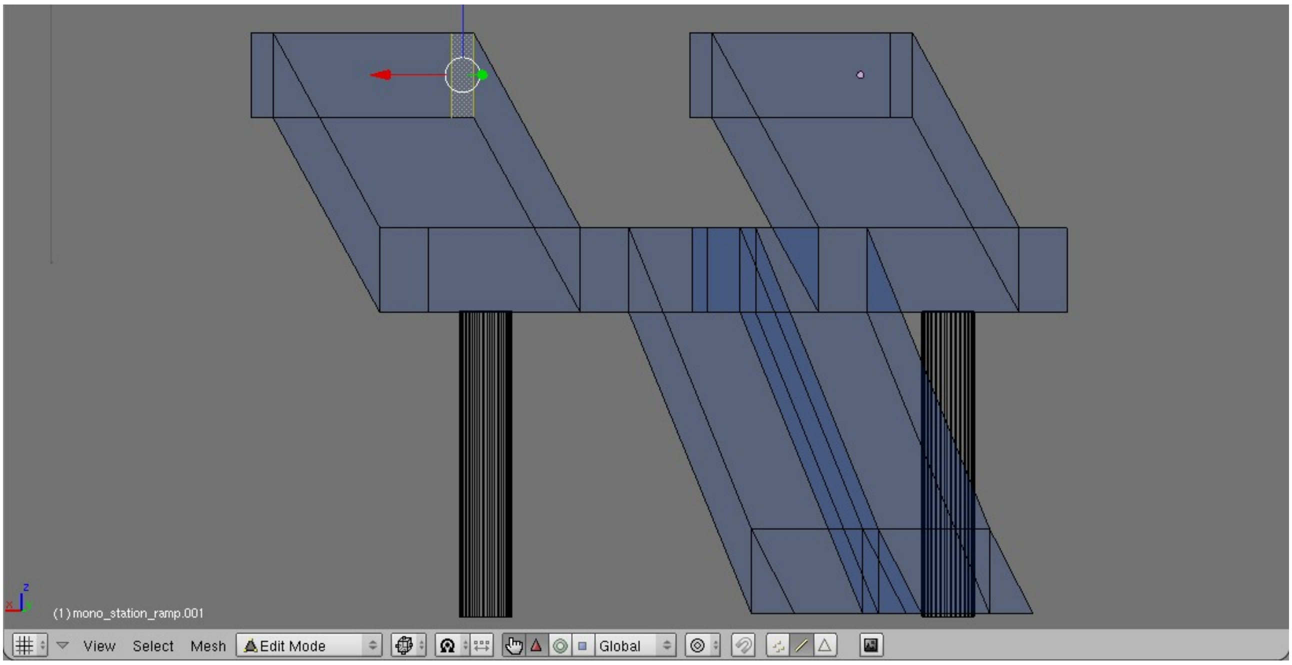


Figure 8: Station platform ramps

The track is a simple cube mesh that has been extruded to fill the profile of the track in the station platform.

The pillars for the station, ramps and tracks are simple cylinder meshes.

4 Intelligent Agent

The intelligent agents for the project were to be the train and the station lifts. However, due to an unknown bug in the game engine where using the lift causes the lift car model to vanish, the lift agent was not completed.

The train agent was to include 3 parts; its travel between stations, the door motion and announcing the trains arrival at stations.

The door motion is a simple reflex agent that awaits a key press to activate an IPO animation of the door opening and closing with a synchronous sound being played.

Announcing the train at each station and controlling the train's speed agents use messages, states, memory, and reflex elements.

Along the track, there are empty objects that detect where the train is near (Figure 9), and send messages bases on their function; in the train announcer case, the empty object would send a message to the station, which would then check which direction the train is travelling to determine which sound file to play.

To control the train's direction and speed, the empty objects on the track would send messages to the train instructing it at what speed to travel; full speed, slow speed or to stop. The stop message from the station would also enable the doors to be opened, and the restart message (by hitting the spacebar) would signal the doors should be closed.

When the train reaches a terminal station, the station announces that passengers should disembark and broadcasts a message that sets the new direction of the train.

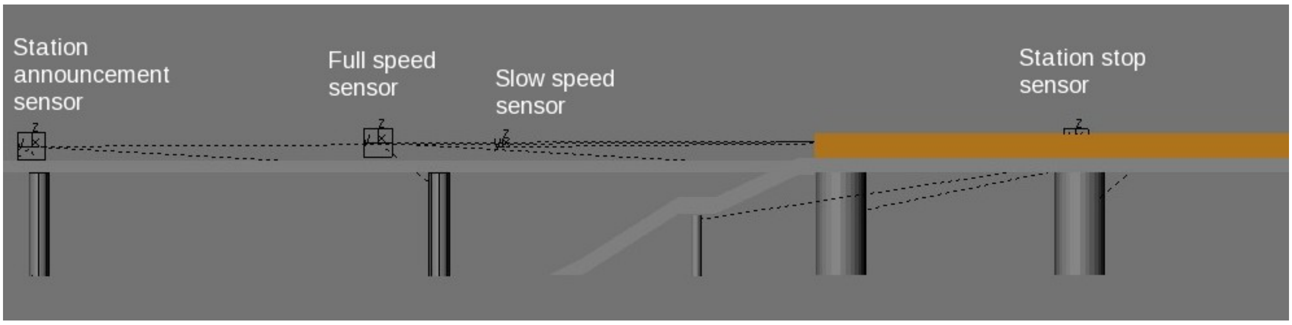


Figure 9: Track sensor positions and functions

5 Evaluation

The door agent, as a simple reflex agent, was tested by pressing the action key, waiting for the animation to finish (ensuring that the sound was correctly aligned with the movement), pressing the action key again afterwards, awaiting the full animation, then continuously pressing the action key to ensure that only one animation and sound were played. The door agent worked as expected and without error.

The train agent was similarly tested, but as it is mostly autonomous, the simulation was simply observed to ensure that the train was travelling at the correct speed, the announcements played at the correct times, the train stopped in the correct locations and was able to be restarted when the spacebar was pressed. One bug is that the sounds are played on starting up the game engine; this is possibly a bug in blender, or it could be a design decision in the sound actuators, according to the blender game engine API manual.

6 Recommendations for deployment

This application would probably be best suited to a fixed monitor, mouse and keyboard interface; the keyboard could either be a full QWERTY keyboard or a simplified keyboard with enough keys to fulfil the model's interaction triggers.
Using more esoteric interfaces, such as a head mounted display or head tracking system would limit the system's availability to all users, and since the simulation would be available to anyone who visits the open day, screening users so as to not cause illness with unusual interfaces or providing support continuously for a novel interface would outweigh any possibly benefits over the traditional fixed display, keyboard and mouse.

7 Conclusion

Blender as a tool is very capable at constructing interactive 3D agents for rapid prototyping a simulation or static scene; it does however suffer from quirks that can be infuriating to work around or to debug successfully; an example of this would be the lift car disappearing instead of moving as instructed.
The project could be extended by adding the doors to the train, which could be done in a few days saving any quirks or bugs that would be encountered. Implementing the extra agent ideas, such as the lift and driving the train, would enhance the simulation immensely.
The train could also benefit from moving along a path, with the carriages moving correctly around corners instead of the current straight line.

8 References

Network West Midlands sound files http://bve4.net/index.php
Blender Game Engine API documentation
http://www.blender.org/documentation/248PythonDoc/GE/

9 Code access

Code can be accessed via GitHub:

http://github.com/samwwwblack/personal-uni/tree/249d705a40d440e73f4b653ea0ff4024f7307bf6/ee4a