Daniel Edmond
301027138

# ENSC 894 - Programming Assignment

## Assignment 1 and 2

### Program logic and implementation

I have implemented my 3D floorplan of an elderly home using simple shapes in openGL. The space is built-up relative to the center of the floorplan. The user is able to pivot the camera from this centerpoint using the left mouse button, allowing for easy viewing of the entire space. An overhead view of the floorplan is shown in Figure 1.

The elderly person is depicted performing the following tasks:

- Lying down in bed
- Sitting in the living room
- Cooking at the stove
- Walking around the home
- Sitting at the shower

The limbs of the humanoid figure can be easily transformed to depict activities performed by the elderly person using this space. These humanoid figures performing various tasks are shown in Figure 2.

Where possible, I have built drawing objects that can easily be translated, rotated, and scaled to produce furniture, walls, etc.

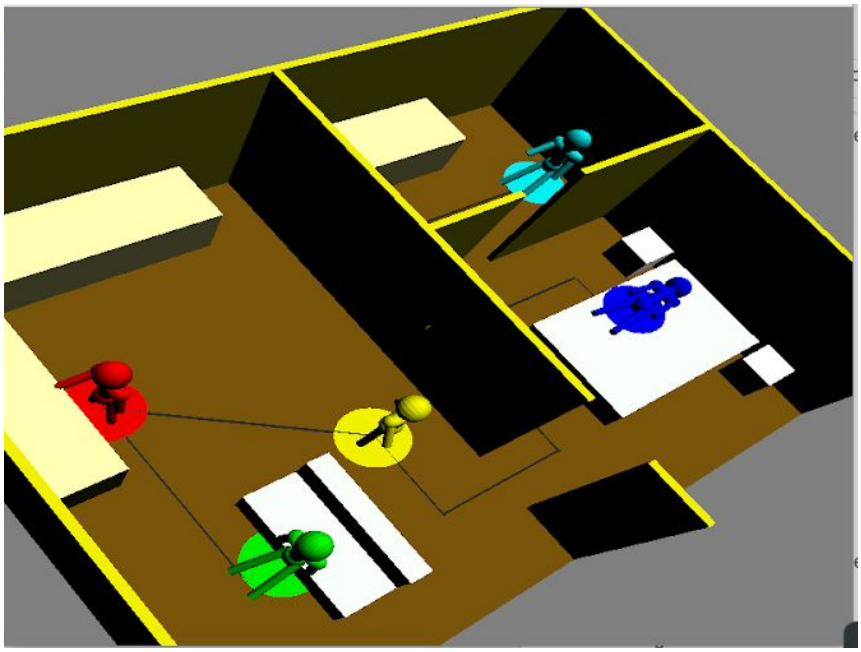


Figure 1 : Overhead view of floorplan

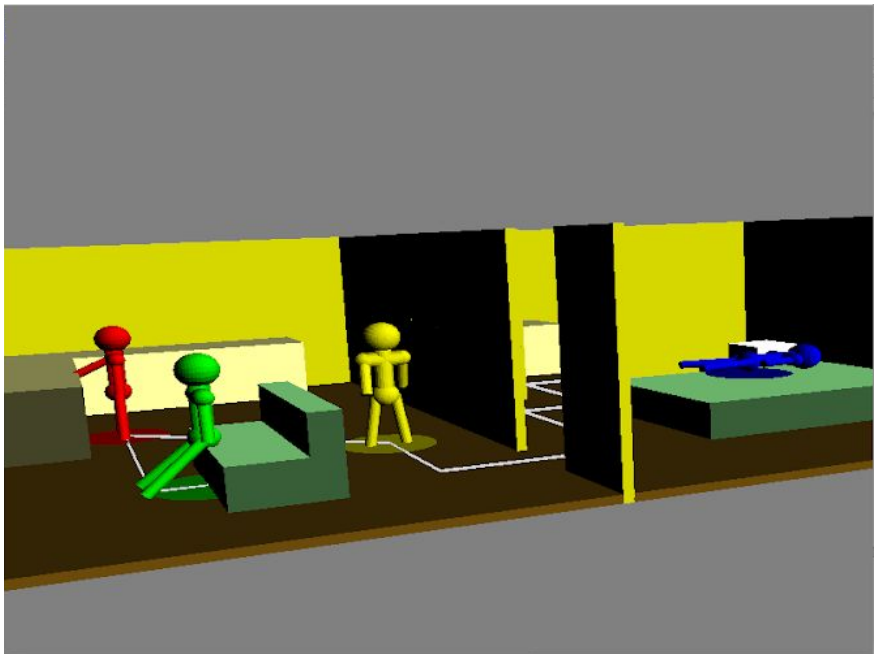


Figure 2 : Elderly figures shown performing various tasks

## Bug Report

There are no known bugs in this program at this time, although there are occasionally some rendering artifacts due to overlapping quads. Fixing this would involve reworking how I build-up the drawing objects in my design, a consideration that I would make when designing my drawing objects in future projects.

## Future Work

I would like to better organize and abstract my data structures in a future revision of this program. This would permit easier manipulation of the floor plan, and better abstraction of the humanoid figure and its position/orientation.

# Assignment 3

## Program logic and Implementation

I have built a forward kinematic model of the Phantom haptic device by very simply building up the links of the manipulator via reference frame rotational and translation in openGL. Once the reference frame for each link was established with rotations and translations, applying the desired link rotations (around the axises of rotation) to the chain of manipulator reference frames was trivial.

A GUI input box allows the user to manipulate each of the joint angles visually in real-time. Examples images of the manipulator with various input angles are shown in Figures 3 and 4.

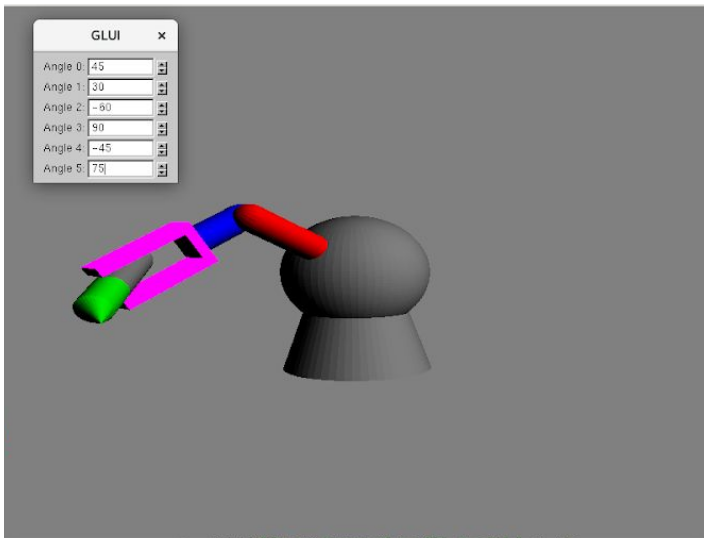



Figure 3 : Phantom haptic manipulator forward kinematics simulation.

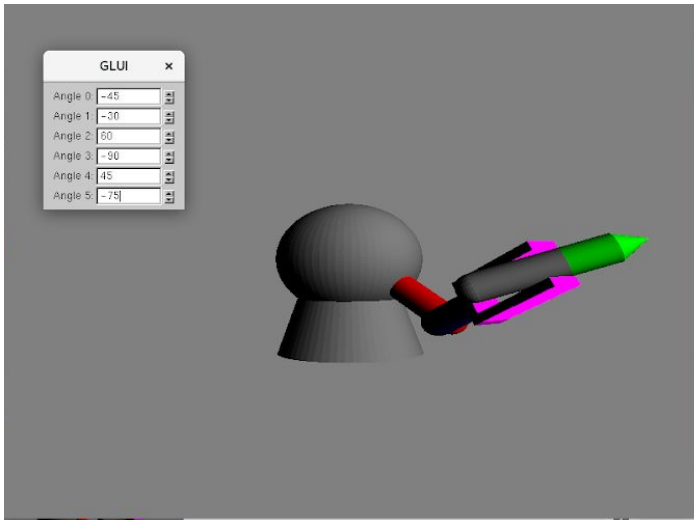


Figure 4 : Phantom haptic manipulator forward kinematics simulation.

### Bug Report

See future work for additional considerations.

### Future Work

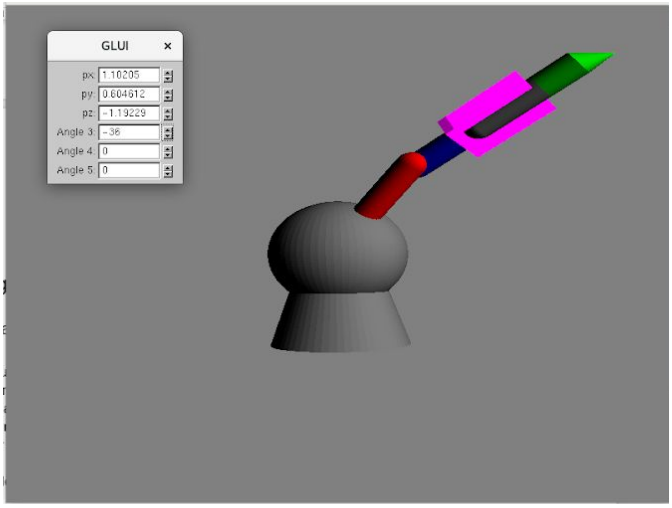
My program currently doesn't consider the valid workspace of the manipulator. The manipulator links are permitted to intersect other parts of the device, which would clearly not be possible. I would like to implement the manipulator workspace into future revisions of this program so that only valid configurations can be input by the user.

## Assignment 4

### Program logic and implementation

I have used assignment 3 described above as a starting point for drawing and rendering the Phantom haptic device. The GUI input box is modified to allow the user to input a cartesian coordinate for the end effector and its orientation. Closed-form inverse kinematic solutions from the course text have been implemented to calculate the first 3 joint angles from the desired end effector coordinate.

An example image of the manipulator with an input end effector position is shown in Figures 5.




## Bug Report

At this time, the inverse kinematic solution calculated by my program is not correct. I have not yet been able to debug the issue with my formulae, or how they have been applied to my specific model.

## Future Work

I will be debugging the issues with my inverse kinematics solution. Similar to assignment 3, it would also be valuable to capture the device workspace so that only valid configurations can be input by the user.