Description:

In the performance arts, choreography is a tricky process. In this project, the process of recording dance will be explored. In particular, the art of poi. I will be collaborating with a fire-spinner whose element of choice is the poi. The project will have a recording component that will attempt to capture the movement of the poi. There will be some feature to playback the recording.

The ultimate goal will be the collecting, reviewing and then piecing together, from these smaller recordings, an entire performance. By using these models, the artist can review the choreography and learn the moves without having to try and piece together something manually.

Design:

It is important that, when working with a project that manipulates data from some external source and renders it to the screen, you abstract the finer details in a responsible manner. The core of my progress in this project is in this design. The Kinect SDK, in this case OpenNI + NITE, is a complicated mess. Interfacing with OpenGL is also a tad complicated. Marrying the two is a tricky prospect.

The first component you need is one that manages the Window. This object will spawn two sub-components: one for input and one for output. The InputHandler object will handle keyboard and mouse interactions. The Presenter will handle drawing to the screen using an abstracted Canvas object.

With this in place, no component will have to deal with both Input and Output responsibilities. Nor will the drawing logic have to be dependent on the logic of the Kinect. This is extremely important to ensure that logic does not creep to the wrong places and should help our development.

The next component is the Kinect component, which is a fairly large object that will encompass all of the Kinect handling that I require.

Demo:

For the demo, I will show the hand tracking working along with the skeleton tracking and perhaps illustrate the least-common mechanism exposed by my code design. For the project completion, we will track the poi object and store the recordings of the movements.
Progress:

This project will be making use of a sample application to get started. The PointViewer application. This application can track your hand and trace out the path. Obviously, we want to save these points along with the skeleton. The skeleton finding is borrowed from the Player sample application.

We still have to store these points, but for the demo, it is sufficient to show the path tracing.

Along with hands being tracked, the difficult part remaining is to track the poi head. This is a very bright, colorful object. Therefore, we just need to coordinate the RGB data with the depth data, which we don't feel will be very difficult.

The hand tracing looks like this:

![Tracking hands](image1)

The skeleton tracking looks like this:

![Skeleton tracking](image2)