**CS560 – Animation**

**Inverse Kinematics - CCD**

1. **Instructions to Build and Run the application**
2. Unzip the submission file and save it in your preferred folder.
3. It is strongly recommended to use a path name with small length; issues may appear with paths longer than 200 characters.
4. Under the graphics folder, double click the Graphics.sln solution file.
5. Go to Build->Rebuild Solution. It is recommended to build the application in release mode, because in debug mode in slower machines framerate could be affected. However, the test results in DigiPen’s PCs showed the application running at 60FPS in average, in both debug and release modes.
6. Once the solution is built, under the graphics folder, execute **copy\_files\_debug.bat** and/or **copy\_files\_release.bat**.
7. Now execute the application whether within visual studio or using the .exe files generated in the **Graphics\Debug** or **Graphics\Release** folders.

**NOTE**: The documentation below will refer to the location of the classes assuming that the reader will search for them in the Solution Explorer of Visual Studio.

1. **Files of interest for project 3.**

* FollowMeComponent.cpp

1. **Object Modeling**

***Manipulator:***

A snake-like manipulator was implemented with 8 links.

***Target Object:***

* The target object is a light blue ball. You can move it with the num-keyboard:
  + 8 – Y+
  + 0 – Y-
  + 5 – Z+
  + 2 – Z-
  + 1 – X-
  + 3 – X+

***Move along a path:***

* Code from previous project was reused. This time, the path is just a line.

1. **Inverse Kinematics**

***Inverse kinematics algorithm implementation***

* The algorithm implemented as CCD.

***Smooth motion***

Angle Clamps and constraints were used to simulate smoothness. As a matter of fact, I tried to used interpolation, but the results were not so good, so that is why I applied this alternative and smoothness is present now.

***Constraints***

Some constraints were applied to make the “snake” get entangled in some moments. Take a look at the comments in the FollwMeComponent.cpp class and the comments written there.

1. **Visualization**

***Integration with project 1 and 2***

* Done.

***Interactive control of target object***

* You can move the light blue target ball with the keys described in section *III-Target Object.*

***Rendering***

Done.

***Rendering speed***

Always at 60FPS aprox, in Release mode.

1. **Documentation**

Required code is appropriately commented. Instructions about where to find features implemented and their comments are in the previous sections of this document.

1. **Extra credits**

The grading guidelines mention other objects such as textures, shading, other objects, etc… Are objects in my project like the loading screen (using threads), UI in the scene, or lights (BRDF) valid for extra-credit?