COSC422 Advanced Computer Graphics

Assignment 3

Skeleton Makeover

Due: 11:55pm, Wednesday, 20 October 2021

Maximum Marks: 20

I. Introduction:

The Open Asset Import Library (Assimp) provides a set of classes and methods useful for loading and animating character models. Assimp is also useful for visualising the animation sequences stored in motion capture data files. When a mocap file in BVH format is loaded, Assimp generates a mesh model of a skeleton using the values of parameters defined in the joint hierarchy.

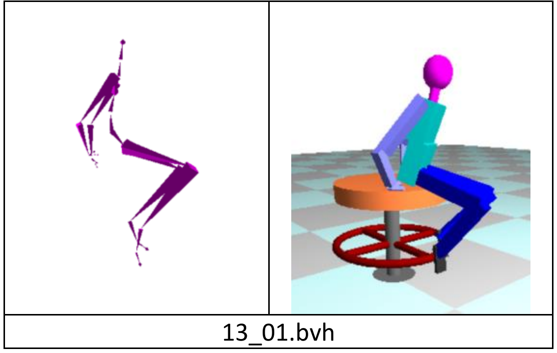
Important note: Recent versions of Assimp (version 4.1.0 and above) contain an error in the BVH loader module that causes BVH animations to be displayed incorrectly. The sequence of Euler angle rotations was wrongly calculated in the implementation file. The correct version of BVHLoader.cpp is provided in the programming exercises section on Learn ("13. Setting Up Assimp.."). Please go through the installation notes given in the section (updated 3rd Oct) before installing Assimp.

II. Task Description:

Programming exercise 15 discusses the sequence of steps required for animating a skeleton model. You could use this program (`SkeletonAnimation.cpp`) as the base code for your assignment task.

As part of this assignment, you will enhance the animated display generated by `SkeletonAnimation.cpp` by designing a surrounding environment comprising of simple graphics objects for the model, and also replace the skeletal model generated by Assimp with a more "humanoid-like" or "robot-like" model. The graphics environment allows the model to interact with a few objects in the scene during an animation sequence.

As an example, the output of 13_01.bvh from the CMU dataset and a modified/enhanced version of the character model are shown in the figure below.



13_01.bvh

The main tasks for this assignment and the marks for each task are detailed below.

2.1 Model Enhancement (max 7 Marks): The render() function in `SkeletonAnimation.cpp` may be modified to replace the default mesh generated for each link with another geometrical shape such as a GLUT object. The process of replacing the mesh structure with a set of GLUT objects is outlined in Assign3.pdf. It is not necessary to create a new shape for each link of the skeleton mesh (a set of links, where appropriate, could be replaced with a single object)

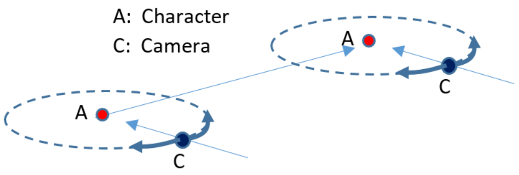
2.2 Character's environment (max 6 Marks): A simple environment that complements the animated display of the model could be generated using a set of geometrical shapes such as GLUT objects. The scene should contain a floor plane. It is not necessary to texture map any objects.

2.3 Extra Features (max 4 Marks): You may implement any of the following additional features to gain up to a maximum of 5 marks.

- Planar shadows (1 Mark): Use only a basic shadow generation method (COSC363)

- Camera motion with the character model (1 Mark): Several animation sequences will involve translational motion of the skeleton (eg. walk, run sequences). For such animation sequences, the camera should follow the skeleton.

- Camera view orientation (1 Mark): Using the arrow keys, the user should be able to move the camera towards or away from the model, or change the view angle around the object towards left or right, keeping the model at the centre, as shown in Fig. 4.



- Hand/foot position tracking (1 Mark): An animation display may require the global position of the end point of a joint chain to determine the world coordinates of a hand or foot of the model. This information is useful for interaction of the model with objects in the scene (eg. Lifting an object)

- Physics based motion (1 Mark): Examples are throwing/dribbling a ball, skipping.

- Animation looping (1-2 Marks): An animation sequence may be looped for generating a continuous motion sequence.

III. Report (Max. 4 pages; Max. marks: 3):

Please prepare a report describing your work, and include the following sections:

- A brief outline of the implemented methods, including any problems/challenges faced and how you attempted to solve them. Please describe the methods used for positioning the model within the view frame and tracking it during motion.
- A few screenshots of the animated model in its environment.
- The complete list of keyboard/mouse functions defined for user interaction.
- Reference to the source of the BVH file, if different from that provided in the course.

IV. Program Development:

Please develop your programs using the fixed function (legacy) version of OpenGL. You may use programs and other supplementary materials provided in the course (eg. BVH files, SkeletalAnimation.cpp).

Demo programs found on the Internet and other OpenGL resources should not be submitted as part of the assignment.

V. Assignment Submission

Submit your files using the assignment link on Learn (learn.canterbury.ac.nz) before 11:55pm on 20 October 2021. Your submission must contain:

1. The source codes and all supplementary files (bvh files, mesh files) needed to run your programs. Please do not include freeglut, opengl, glew, glm or assimp library files.

2. Your report in Word or PDF format.

Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications. You may submit up to one week late for a 15% penalty.

2. This is not a group project. Your assignment must represent your own individual work. In particular, students are not permitted to share program source code in any way. However, you may discuss ideas, implementation issues etc using the class forum on Learn.

3. Standard departmental regulations regarding dishonest practices and late submissions apply.