

## Computer Graphics Assignment 3: Bvh Viewer

Handed out: May 25, 2020

**Due: 23:59, June 14, 2020 (NO SCORE for late submissions!)**

- Only accept answers submitted via git push to this course project for you at <https://hconnect.hanyang.ac.kr> (<Year>\_<Course no.>\_<Class code>/<Year>\_<Course no.>\_<Student ID>.git).
- Place your files under the directory structure **<Assignment name>/<your files>** just like the following example.

```
+ 2020_ITE0000_2019000001
+ ClassAssignment1/
- main.py
- report.docx
```

- The submission time is determined not when the commit is made but when the git push is made.

1. Implement your own bvh file viewer.

- A. You have to implement all requirements in a single program. This assignment DOES NOT require each requirement to be a separate program
- B. The window size doesn't need to be (480, 480). Use the larger window that is enough to see the details of the viewer.

2. Requirements

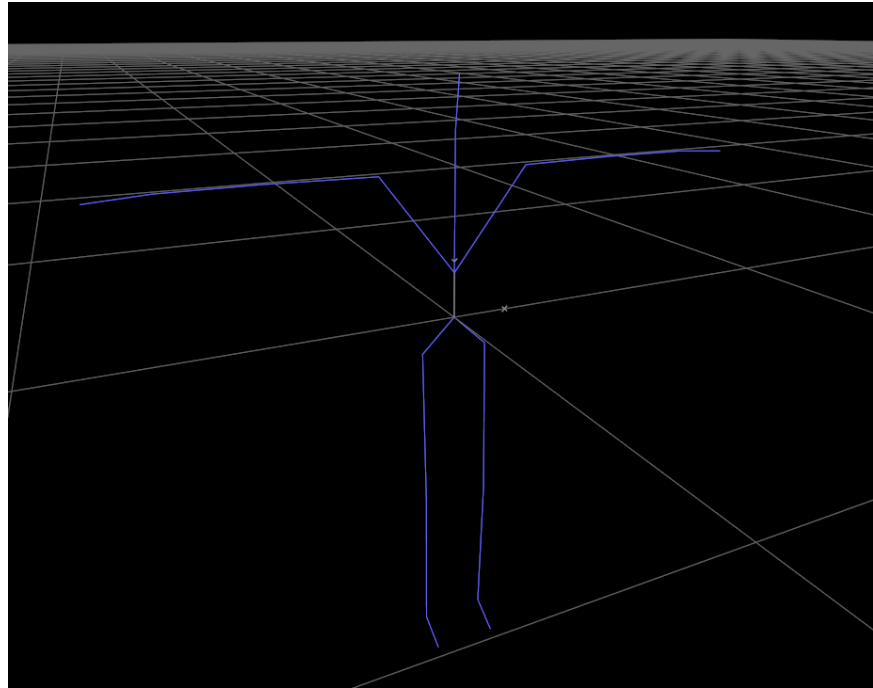
**A. Manipulate the camera in the same way as in ClassAssignment1 using your ClassAssignment1 code (10 pts).**

- i. Also draw the reference grid plane.

**B. Load a bvh file and render it (80 pts)**

- i. Open a bvh file by drag-and-drop to your bvh viewer window **(10 pts)**

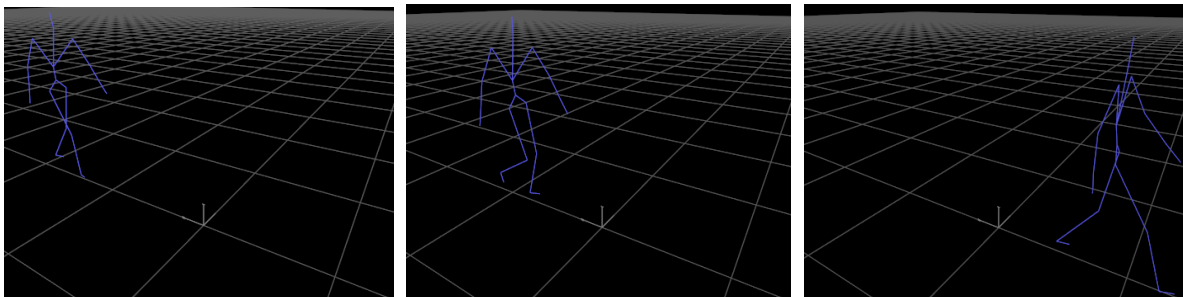
1. Google *glfwSetDropCallback* to see how to do it
  2. The viewer should render only one bvh file at a time. If a bvh file B is drag-and-dropped to the viewer while it is rendering another bvh file A, the viewer should only render the new bvh file B.
  3. **This feature is essential for scoring your assignment, so if not implemented, you won't get any score for "Load a bvh file and render it (80 pts)"**
- ii. Read the bvh file and **render the "skeleton" (t-pose) of the motion when you load the file by drag-and-drop (30 pts).**
1. **Do not automatically animate the character.** Just draw its skeleton (t-pose) when you open a file.
  2. Just draw joints with offsets between them only using the HIERARCHY section of the bvh file.
  3. In other words, draw a pose of the motion with zero translation (0,0,0) and no rotation (identity matrix) being applied to transitional joints and rotational joints, respectively.
  4. Draw the skeleton by line segments. Each line segment should connect each pair of parent-child joints.
  5. For end-effector joints such as foot, a line segmented should connect the end-effector joint and the "end site".
  6. An example screenshot for sample-walk.bvh



A.

iii. **Animate the loaded motion if you press the <spacebar> key (30 pts).**

1. As time goes by, draw each pose of the motion using each line of frame data in the MOTION of the bvh file, from the start frame to the end frame.
2. Calling `glfw.swap_interval(1)` in your main function, then just drawing the pose of each frame for each iteration of the main loop (the while loop in the main function) would be enough for timing of rendering each pose.
3. After drawing the last frame, just **replay** the motion again (draw from the first frame to the last frame again).
4. Example screenshots for `sample-walk.bvh`



iv. When open a bvh file, print out the following information of the bvh file to stdout (console) **(10 pts)**

1. File name
2. Number of frames

3. FPS (which is  $1/\text{FrameTime}$ )
4. Number of joints (including root)
5. List of all joint names

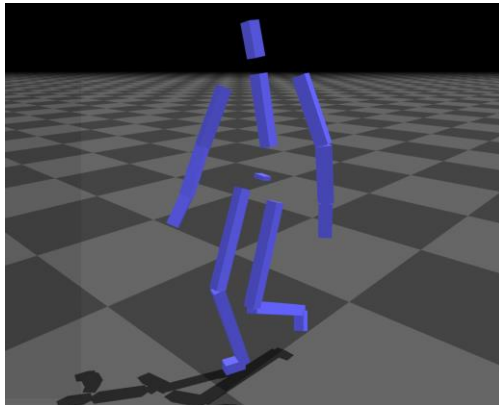
### 3. Report (10 pts)

- A. Submit a report of **at most 2 pages** in docx file format (MS Word). Do not exceed the limit.
- B. The report should include:
  - i. Which requirements you implemented (5 pts)
  - ii. A few screenshot images of your program with downloaded bvh files (5 pts)
    1. Download bvh files from the Internet and open them with your viewer. You may download bvh files from
      - A. <http://motion.hahasoha.net/>
      - B. <http://mocap.cs.sfu.ca/>
      - C. <http://dancedb.eu/main/performances>
    2. Choose some of the best looking screenshot images of your viewer.
    3. **Do not use the sample bvh files for this requirement.** You must use other bvh files downloaded from internet to get score for this requirement.
- C. You do not need to try to write a long report. Just write down the required information. Use either English or Korean.

### 4. Extra credits (Only one of two items A, B will be reflected in the score)

- A. Use a box to draw each body part instead of a line segment (**+10 pts**).
  - i. The boxes should be rendered with proper shading and lighting settings.
  - ii. If you render the body parts without lighting and shading, you cannot get the score for this item.
  - iii. You can implement this requirement for only one of the sample bvh files provided.

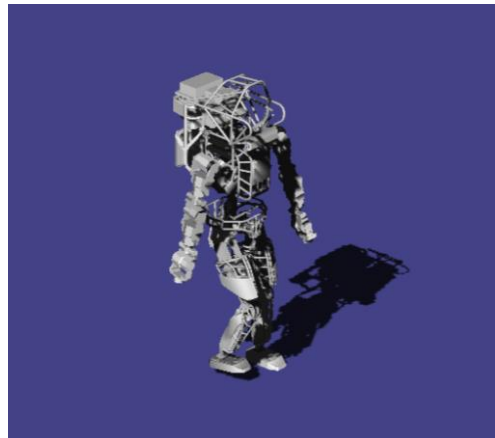
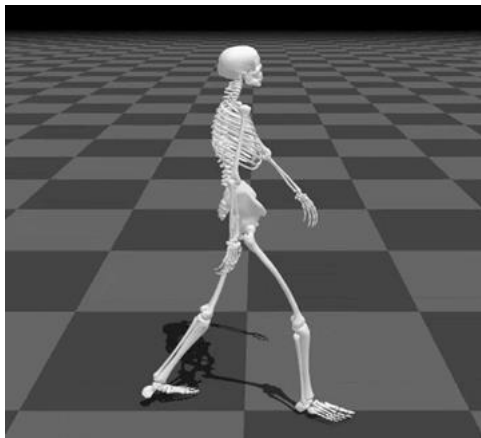
- iv. An example screenshot for sample-walk.bvh



- 1.
2. (You don't need to draw the shadow and shaded ground plane)

- B. Use different obj files to draw each body part instead of a line segment **(+20 pts)**.

- i. The meshes should be rendered with proper shading and lighting settings.
- ii. If you render the body parts without lighting and shading, you cannot get the score for this item.
- iii. You can implement this requirement for only one of the sample bvh files provided.
- iv. Example screenshots (not for sample-walk.bvh)



- 1.
2. (You don't need to draw the shadow and shaded ground plane)
3. Search and download cool obj files from the internet and use them.
  - A. If you use simple obj files like box shapes, you'll only get 10 pts.

## 5. Runtime Environment

- A. **Your program should be able to run on systems only with Python 3.7 or later, NumPy, PyOpenGL, glfw. Do not use any other additional python modules.**
- B. Only **glfw** is allowed for event processing and window & OpenGL context management. **Do not use glut functions for this purpose.**
- C. **If your program does not meet this requirement, it will not run on TA's computer so you will not get any score for this assignment (except report).**

**6. Test your viewer with sample bvh files.**

- A. sample-walk.bvh: A walking motion file
- B. sample-spin.bvh: A walking while spinning around a vertical axis

**7. What you have to submit:**

- A. **.py files**
  - i. You can use multiple .py files for this assignment. In this case, explain how to run the program in the report.
- B. **.docx report file**

**8. Additional information**

- A. *drop\_callback* in glfw python binding is slightly different from that of original glfw written in C. *drop\_callback* in python takes only two parameters, *window* and *paths*. *paths* is a list of dropped file paths.
- B. Python provides powerful string methods helpful for parsing a bvh file. Among them, `split()` will be most useful.