Supplemental Assignment:

Deadline: 30th August 2021

There are 2 parts to the assignment – part 1 is developed in modern shader-based OpenGL and part 2 is developed in Unity/UE4. You will demo both projects on the same date, and should submit 2 reports on Blackboard.

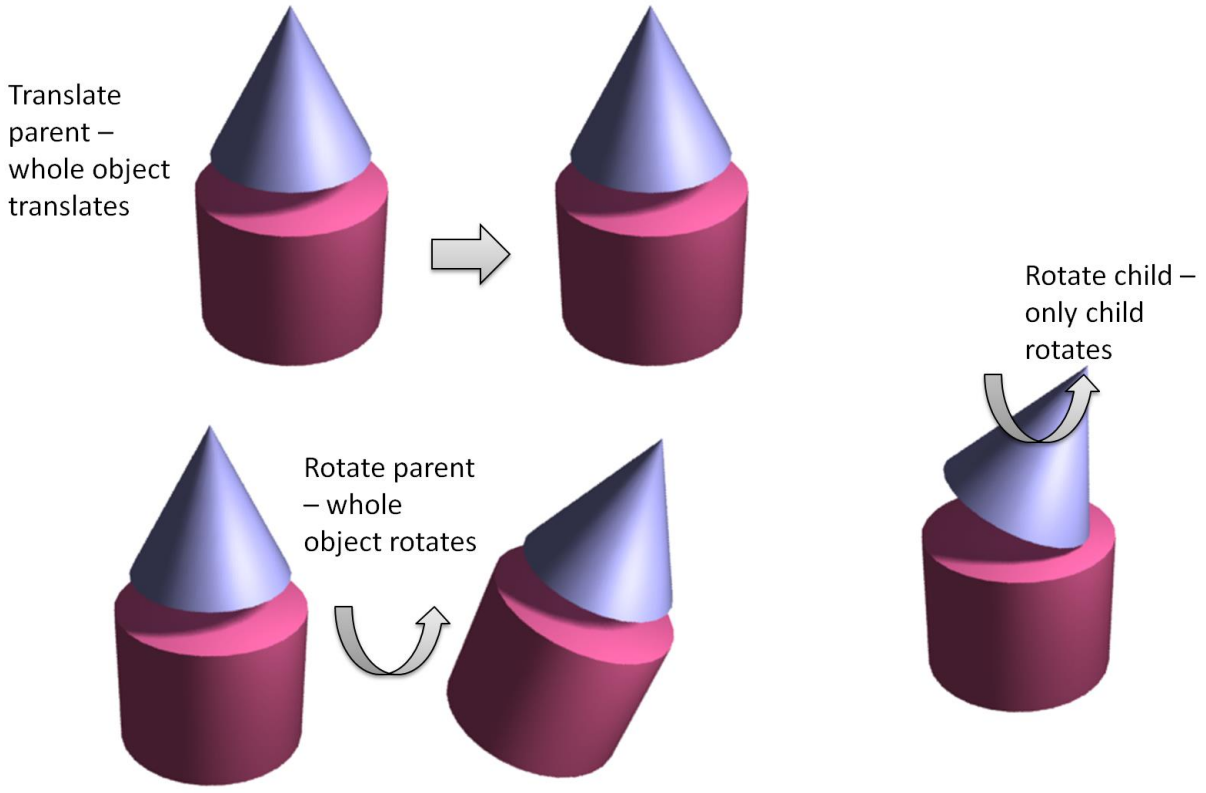
General Rules:

1. The deadline for the assignment is the 30th August 2021 for the demo and report. Late submissions will result in severe penalties.
2. This examination is strictly **individual** (no groupwork). Please note that soliciting the help of other individuals to help complete or improve your implementation is considered academic misconduct and may result in severe penalties. Also, be aware that demonstrating a project that was not created by you or not crediting out-of-the-box features and plugins is considered **cheating** and will be reported as such.. The demonstrator will check if you have an understanding of the code that you have written.
3. You will be required to **show your working** programs (part 1 and 2) to the demonstrator on the due date.
4. You will also be required to submit 2 pdf files with your **reports for part 1 and part 2** on Blackboard by the due date. Submissions must be on Blackboard as we will not be accepting submissions via email. Details of what is required for each part is detailed below.

PART 1: Hand hierarchy in OpenGL

Worth 20% of supplemental grade

The purpose of this assignment is to test your knowledge of transformations and hierarchies

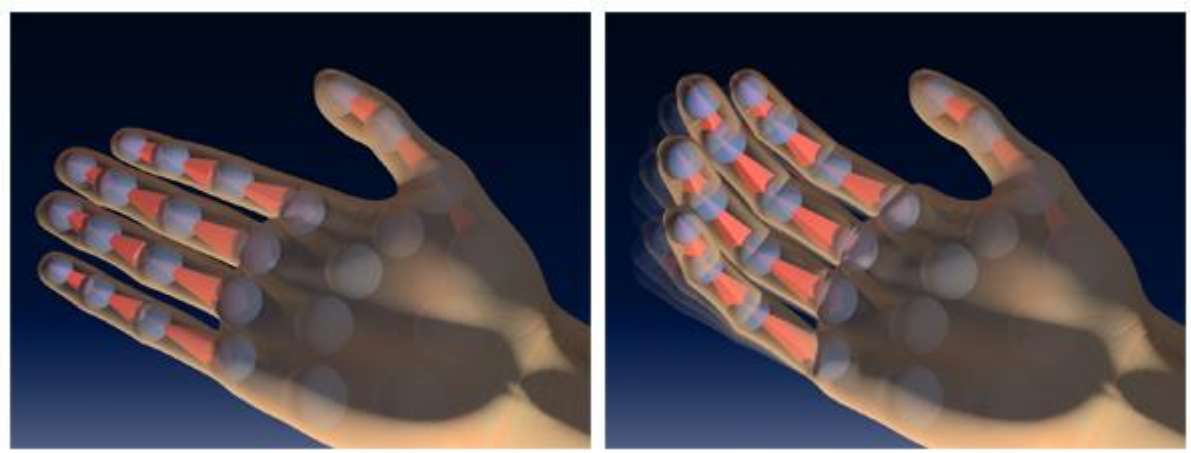


Many objects we wish to model form a natural hierarchy of connected components. "Movement" of one component causes another set of "attached" components to move in like fashion. This hierarchy can be represented by a tree (more generally a graph) where each child node inherits some set of transformations (which define the movement) specified by the parent. For example, consider the hierarchy above. A rotation of about the base cylinder causes the entire object to rotate, but a rotation about the cone only causes the cone to move. The cone inherits both rotations, whereas the base inherits only rotations of the base.

---

1. You are required to create a hierarchical model of a hand. The hand should be constructed from a base, 4 fingers (with 3 finger segments each) and a thumb (with 3 segments). Your program must demonstrate the parent-child relationships.

2. You should use glm maths library (or other), or create your own. Remember that OpenGL uses column-order matrices so if you are using a math library that assumes row-order, you will have to get the transpose of the resulting matrix. Remember to check the structure of the translation matrix to help figure this out. You can also download the basic cylinder object class from Blackboard, if you would like to use a cylinder to represent the plane.



Requirements:

Draw a scene graph to represent your object. Include details of the initial position and orientation of each part, and the relative offsets from parent to child.

Bone class: Here you define a bone, which should have an ID or NAME. The bone should also store a reference to its parent and to a list of its children. The bone should also have an initial orientation, local transformation, and the ability to update its orientation with respect to its parent (to create animation). You may want a function here that can calculate the global transform of the bone, based on the global transform of the parent.
Skeleton class: Here you define a skeleton by the number of bones that it has. You will want to handle the root bone differently to the other bones. You will also need to have a function to calculate the global transformation of all of the bones in the hierarchy. This function should be recursive and should

traverse the hierarchy, working out the global transformation of each bone (using its local orientation and the global transform of its parent).

Rendering: Finally, you will need to render the hierarchy in your display function. You should set up your initial object positions, update their local transformations to produce animation, and then calculate their global transformation which will be passed to the vertex shader as a uniform matrix. The use of shaders is mandatory.

It would be advisable to start with a basic hierarchy such as that shown above with a simple cone and cylinder.

Requirements and Examination

Your program should have the following features:

- Correct structure (~30%)
  - Bone Class
  - Skeleton Class
  - Recursive function for tree traversal
- Basic model showing a parent/child relationship (~10%)
- Hand model (as described above) showing animation of the parent/child relationship (~40%)
- Extra Feature (~20%)
  - Good visual appearance of the hand (e.g., Using 3D model rather than Cylinders). Character model provided on Blackboard, if needed.
  - Forward kinematics to create an interesting/plausible hand animation
  - Other?

Note: The [approximate] marking scheme provided shows the maximum marks that can be obtained for each section if completed perfectly. Merely attempting a section does not imply the full score indicated.

PART 2: Interactive Forrest-Scene Movie with Animated Characters

Worth 80% of supplemental grade

The purpose of Part 2 is for you to demonstrate the real-time animation concepts and methods you have learned in an interactive movie, set in a forest with characters singing, dancing, playing music or talking around a fireplace.

1. You must create an Interactive movie using the animation concepts that you have learned. See the end for specifications.

2. You must use a higher-level graphics engine such as Unity or Unreal Engine 4 OR continue in modern shader-based OpenGL (version 3.0 and upwards). Assets such as characters and environments can be imported, but the movie should be constructed, and logic should be implemented by you (e.g., you can import a skinned character model, but not one that has full game logic for navigation).

3. The demo will be a 15-min presentation to the demonstrator, consisting of a description of your work along with a demo running to show the features of your game. Each required feature must be demonstrated with a clear explanation of the underlying mechanisms/principles of how you achieved it.

4. A written report, documenting your research, and implementation with screenshots, technical features, citations etc. is also required. It should include clear details and links of where the assets came from (character, scene, etc.), which parts were coded by yourself and which out-of-the-box features were used. It should also include a link to an online repository containing your code. It should be submitted via Blackboard. Please also capture a movie showcasing your game and include a youtube link, with "CS7GV5" in the title.

Requirements and Examination

The essential components are:

- **Must have** 3-dimensional objects and views
- **Must be** a camp-fire themed movie
- **Must be** set to a music track related to the camp-fire theme, and the animation and music should be clearly related (using for example animation timing, staging, related visualizations, lipsync to the lyrics, etc).
- **Must be** 30 seconds or longer

The required components are:

- **Must have at least** four different reasonably realistically moving articulated animated characters, as part of the forest camp fire story-line and visible during the camera path (approx. 10%)
    a. There must be multiple joints and children (e.g., arm links and fingers)
    b. The articulated characters will be assessed to ensure they have correct hierarchy and can move correctly according to its structure.
    c. It can use forward or inverse kinematics
    d. It does not have to be organic/human (i.e., can be a robot or object come to life)
- **Must have** an interactive element involving the fire (e.g., flame turns green when character approaches, etc.) (approx. 10%)
- **Must clearly demonstrate** the following features as part of the scene (approx. 50%)
    a. Motion capture of dancing/playing an instrument/singing
    b. Particle system for the fire
    c. IK for characters holding an instrument or a pot over the fire
    d. Facial animation showing singing or talking
    e. Motion state machine as part of the scene (the more complex, the more marks)

The essential components must be present for the assignment to be graded. The required components are worth approximately 70% of the assignment mark.

Additional Features:

The final 30% (approx) will be given for research and execution of additional animation features

- These can include some of the following, or indeed others that you think of:

1. Stylized motion
2. Crowd Simulation
3. Gesture or personality modelling for the character
4. Complex/scripted camera motion
5. Interesting character behaviours/AI
6. Character interactions
7. Particularly imaginative narrative
8. Cloth Simulation
9. Physically-based animation
10. ??? your own imagination is the only limit

Note: The [approximate] marking scheme provided shows the maximum marks that can be obtained for each section if completed perfectly. Merely attempting a section does not imply the full score.