Name: ____

Midterm Exam

15-462 / 15-662 Computer Graphics, Spring 2021

105 points

Due 11:59pm Thursday, March 4th (Pittsburgh time)

Submit your results to Gradescope in any format (typeset, handwritten, etc.). 5 points of your score is assigned for neatness, so please make sure your answers are easy to understand and read! The first 24 hours of the exam you may post clarification questions to piazza. For more details, see https://piazza.com/class/kjsthko53ls2hy?cid=193. Good luck and have fun!

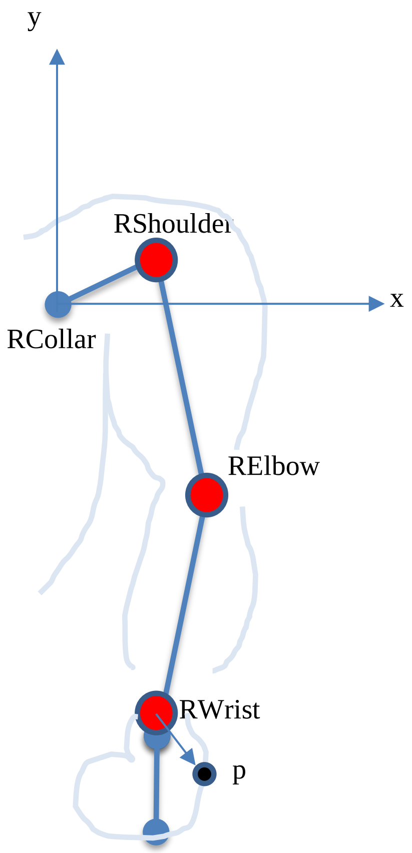
0. (5 points) These points are for neatness in your handin ☺

1. **(60 points) Processing Motion Capture data**

This problem will be focused on interpretation of a motion file format for character animation. This example is based on a real file format used in games, movies, animation research, etc. The format consists of information about the skeleton and information about the motion.

Here is the motion format we will use. It begins with a skeleton, which is drawn to the right in its home pose. Note how the "OFFSET" associated with each joint describes its translation with respect to its parent in the home pose. Take a little time to trace out the skeleton from the definition to make sure you understand the skeleton representation.

```
HIERARCHY
ROOT RCollar
{
   OFFSET        0.0    0.0   0.0
   JOINT RShoulder
   {
          OFFSET          4.0    2.0   0.0
          CHANNELS      Zrotation   Xrotation   Yrotation
          JOINT  RElbow
          {
                    OFFSET          2.0       -10.0     0.0
                    CHANNELS      Zrotation  Xrotation       Yrotation
                    JOINT   RWrist
                    {
                              OFFSET          -2.0    -10.0    0.0
                              CHANNELS      Zrotation  Xrotation  Yrotation
                              End Site
                              {
                                        OFFSET     0.0  -4.0    0.0
                              }
                    }
          }
   }
}
MOTION
Frames:  2
90    0    0     -90 0 0    0 0 0
 0  -90  0      0   0 0    0 0 0
```



Name: ____

The CHANNELS keyword for each joint gives a representation for rotational motion at that joint. Each of the red colored joints is a three degree of freedom joint (i.e., a ball joint), with motion described in Euler angles. The rotations occur in a specific order as expressed in the file format. For this example, all joints have rotations expressed in "ZXY" order. What this means is that we would form the rotation matrix for one joint as follows, where the subscripted matrices represent rotations about a single axis.

$$R = R_z(\theta_z)R_x(\theta_x)R_y(\theta_y)$$

Two frames of motion are shown. Each row contains Euler angles for all three joints in the order they appear in the skeleton definition, i.e.:

$$\theta_{sz} \quad \theta_{sx} \quad \theta_{sy} \quad \theta_{ez} \quad \theta_{ex} \quad \theta_{ey} \quad \theta_{wz} \quad \theta_{wx} \quad \theta_{wy}$$

Where subscript *s* indicates the RShoulder joint, *e* is the RElbow, and *w* is the RWrist joint. All angles are expressed in degrees. For example, we find that the rotation of the RShoulder joint is 90 degrees about the Z-axis in frame 1 and -90 degrees about the X-axis in frame 2. When a joint moves, it carries all of its children with it (i.e., rotating the RShoulder joint will rotate the upper arm, lower arm, and hand).

We use this motion information to construct linear maps that can then be used to transform the triangles that define our character's shape. For example, to express rotation of a point p about the RWrist joint, we could write:

$$p' = R_w p = R_z(\theta_{wz})R_x(\theta_{wx})R_y(\theta_{wy})p$$

a) (5 points) Draw a sketch of the arm pose in the first frame of the motion. Be careful about the order of rotations expressed in the motion information! Use any point of view that is convenient to show the pose. Your sketch does not have to be perfect, but it should tell us that you interpreted the rotations correctly.

b) (5 points) Draw a sketch of the arm pose in the second frame of the motion. You may use a point of view different from that of part a). Just include your coordinate axes in the drawing.

c) (5 points) Write the 4x4 transformation matrix indicated by $R_z(\theta_z)$.

d) (5 points) Let's use the shorthand $T(t_x,t_y,t_z)$ to express the 4x4 transformation matrix that translates a point by (tx,ty,tz). Write the 4x4 transformation matrix indicated by $T(t_x,t_y,t_z)$.

e) (10 points) In our skeleton, the RCollar joint remains fixed at the origin. Point p is defined in the local coordinate frame attached to the end site, and will move in response to rotations at the RWrist, RElbow, and RShoulder joints. What is the correct sequence of transformations required to express point p in the world coordinate frame (i.e., relative to the RCollar joint)? You may express your answer as a sequence of primitive transformations, but please use the notation shown in parts c) and d) for the transformation matrices. We are looking for you to have these primitives in the correct order with the correct parameters.

Name: ____

f) (5 points) Consider the joint RShoulder. Suppose we know that $\theta_{sz} = 0$. Give an example of values for θ_{sx} and θ_{sy} that put the shoulder joint into gimbal lock. You may want to choose the simplest example possible. (It will help you on the next parts!)

g) (5 points) What is the 3x3 rotation matrix for your example?

h) (5 points) If the shoulder joint is in gimbal lock, there will be some rotation axis about which local rotation at the joint is impossible. What direction is it for your example? Please use the world coordinate axes.

i) (5 points) Create a single 3x3 rotation matrix that represents your rotation into the gimbal lock position (from part g)), followed by a very tiny rotation (of epsilon) about the axis you found in part h). This 3x3 matrix represents where we would like to go by moving slightly in the direction which is "impossible."

j) (5 points) The rotation matrix created in part i) represents where we would like the arm to go, but seem to be prevented from doing so by gimbal lock. The following article gives pseudocode for computing Euler angles for any rotation matrix, including ours: http://www.close-range.com/docs/Computing_Euler_angles_from_a_rotation_matrix.pdf. (You don't have to read it, just know it exists.) Our rotation order is different, but we could derive similar pseudocode. This result indicates that there should be some set of Euler angles that pose the arm to produce the rotation in part i). Knowing that incremental motion directly towards the goal appears to be impossible, what do you suppose will happen if we attempt to move the shoulder joint by linearly interpolating Euler angle parameters corresponding to the rotations in parts g) and i)?

k) (5 points) Given the results above, what practical problem(s) might the Euler angle representation pose for animators?

l) (5 points EXTRA CREDIT) Prove or demonstrate that your intuition from part j) is correct.

2. **(40 points) Shadow Maps.**

We have not yet talked about how to represent shadows in the standard graphics pipeline. At first glance, this appears to be a difficult thing to do, because determining whether some triangle A casts a shadow on triangle B would appear to require checking all triangles against all others, which violates the spirit of the graphics pipeline, where we want to pass as many triangles as possible through the pipeline in any order that seems convenient.

There are, however, some tricks that make rendering shadows more efficient. We will work through one of them – the shadow map. The concept of the shadow map is to begin by rendering objects from the point of view of each light source (i.e., treating each light source as a camera). Geometry closest to a light source should be illuminated by that light. Geometry behind other objects from the point of view of the light source should be in shadow.

Here is the setup for our problem. You have an overhead light source located at point (0,20,0). Your scene is built on top of a simple 10x10 horizontal ground plane centered at the origin. You may assume that all of the geometry you have is contained in the pyramid formed from the corners of the ground plane and the light location.

a. (5 points) Draw a sketch of the scene.

b. (10 points) We will begin by rendering the scene from the point of view of the light source. What 4x4 matrix will project points in the scene onto a horizontal "image plane" that is located distance 1 below the light source? Create a simple matrix that projects all points onto this plane.

c. (5 points EXTRA CREDIT) We want to preserve some depth information. Create a matrix that preserves depth. Your matrix should map points in the image plane to z=1 and points in the ground plane to z=20. For full credit, create the matrix, test depth at y=0, 9.5, and 19, and use the results to discuss how depth resolution varies for nearer vs. farther distances from the light.

Assume we have a correct matrix M which transforms points in the world onto a plane distance 1 from the light source, while preserving some depth information. Using matrix M, we can render the scene from the point of view of the light source with the depth buffer turned on and store the resulting depth values in a texture. This texture is called the **shadow map**.

Let's now think about how to use the shadow map to render shadows into our scene.

d. (5 points) The shadow map can be stored as a texture. Let u and v be parameters that allow indexing into this texture (i.e., into the shadow map). What is the intuitive meaning of these parameters? Give specific examples in your answer (e.g., what does u=0, v=0 correspond to).

e. (5 points) You are rendering a fragment from the point of view of the actual camera. Suppose that fragment is located at point q in 3D space. How do we find parameters u and v given q? Describe the necessary process in words.

f. (5 points) Give the matrix to transform q to obtain coordinates u, v, and w, where u and v are texture coordinates into the shadow map and w is a depth estimate to compare to the shadow map value.

g. (5 points) Under what conditions do we render the object in shadow?

h. (5 points) Does all of this processing (described in parts f) through g)) have to be done repeatedly at the fragment level, or can some / all of it be done at the vertex level. Describe in detail what calculations can be done per-vertex and which must be done per-fragment.