CS370 - Assignment 4

1. Using the example robot from lab 14 suppose we know the *position* where one of the upper arms is starting and have a specific final *location* we want the arm to have. Give a method to create the path the robot should follow to get from the starting point to the end point.

This is the subject of key framing in animation where the initial and final orientations of a model are given and the rendering algorithm must create intermediate frames to produce a smooth transition. It also occurs in robotics as inverse kinematics, where the end of the arm must go from one point to another to accomplish a task. If we let the various components have sizes and orientations given by:

- Base - height h_1, rotation θ
- Lower arm - height h_2, width w_2, rotation ϕ
- Upper arm - height h_3, width w_3, rotation ψ

Therefore, the final transformation for points on the upper arm is given by:

$$M = R_y(\theta)T(0, h_1, 0)R_z(\phi)T(\pm(w_2 + w_3)/2, h_2, 0)R_z(\psi)$$

One solution to the problem is to find joint angles (θ, ϕ, ψ) corresponding to the two desired points (note, however, that these sets of angles are often *not* unique). Then the angles can be *linearly* parameterized by

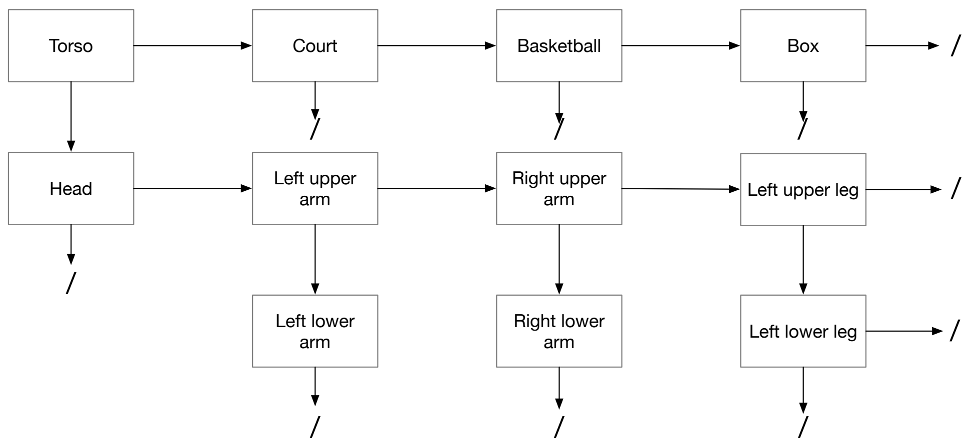
$$\begin{aligned}\theta &= (1-\alpha)\theta_i + \alpha\theta_f \\ \phi &= (1-\alpha)\phi_i + \alpha\phi_f \\ \psi &= (1-\alpha)\psi_i + \alpha\psi_f\end{aligned}$$

where (θ_i, ϕ_i, ψ_i) are the intial angles and (θ_f, ϕ_f, ψ_f) are the final angles. As the parameter varies from $0 \leq \alpha \leq 1$, the joint angles will cause the tip to go from the initial point to the final point.

Unfortunately, this simple approach can often result in problems such as gimble lock (joint angles that produce an impossible physical configuration) or large rates of change in angles which produces strange animations. A common way to avoid such problems is to specify a path the end must follow (rather than simply the endpoints.) Then using a more mathematically advanced technique involving *quaternions* (an extension of complex numbers) for the angular changes can produce much more acceptable results.

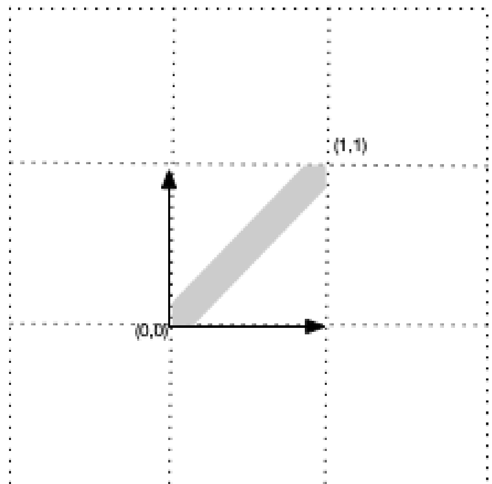
2. Using the example scene graph diagram from lab 14, sketch the scene graph for the programming portion of the assignment. **Note:** You only need to show the relationships between the nodes, i.e. it is not necessary to include transformations.

Selecting the `torso` as the root node, we can construct the scene graph as follows where horizontal links represent *siblings* and vertical links represent *children*



Note: Sibling nodes can be arranged in any order (since they are all independent). For example, we could have selected the `Court` as the root node and had the `Torso` be a sibling of `Court`.

3. Given the following texture map

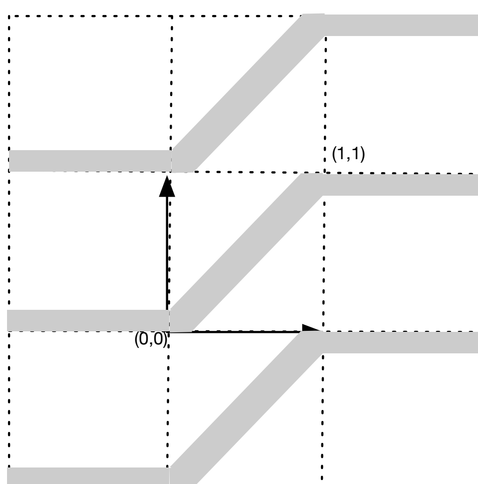


with wrapping modes

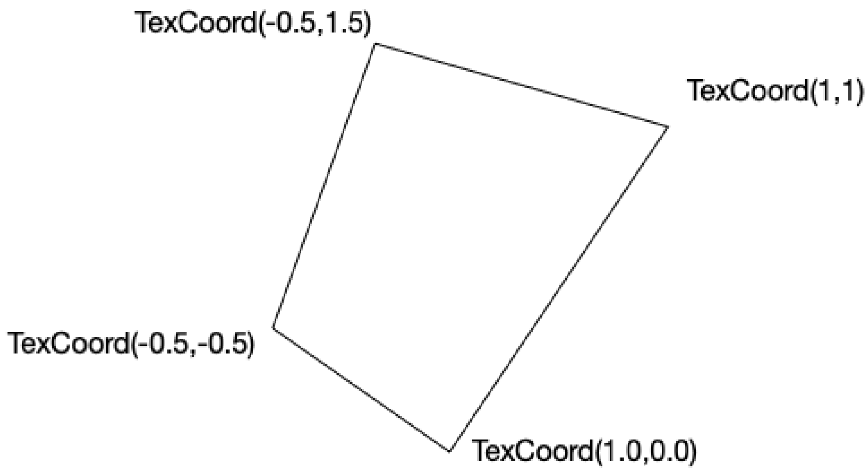
```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
```

sketch the surrounding parts of the texture plane.

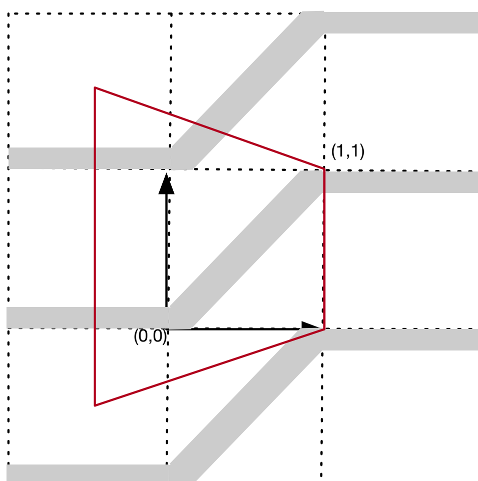
Since we are *clamping* in the horizontal (s) direction, we will *extend* the border pixels horizontally giving the bars at the top to the right and bottom to the left. Then since we are *repeating* in the vertical (t) direction, we will *copy* the entire middle row to the other two rows. This will give the extended texture plane as shown



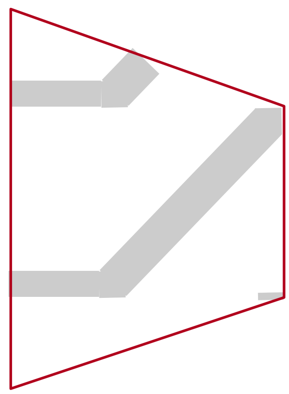
4. Using the texture plane from question 3, sketch the textured figure given below using the provided texture coordinates



If we mark the texture coordinates on the texture plane from question 3, we will be using the area marked in red below



which we then "cut" out giving



Finally we then "stretch" this portion of the texture plane onto the original object giving roughly