Assignment02: Arm Inverse Kinematics

Guided by Professor Rachel McDonell

Implemented by: Yuzhou Shao

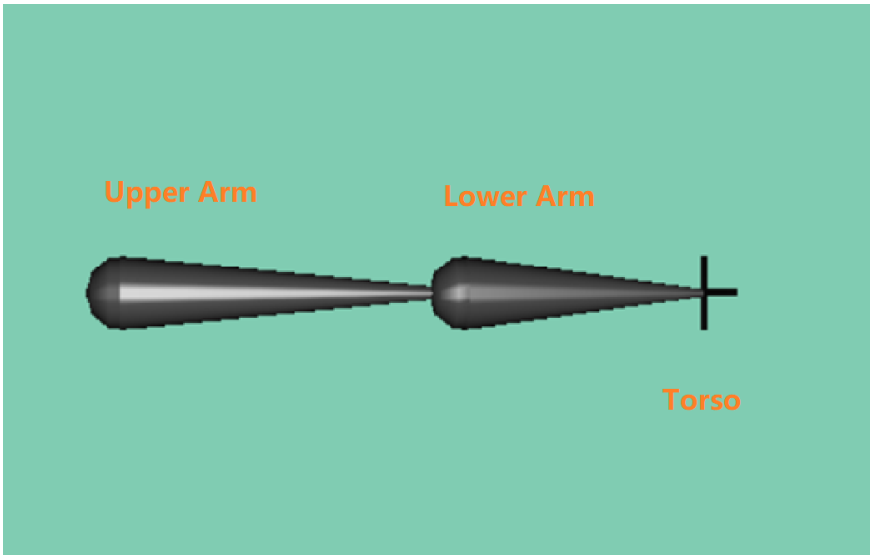
Student id: 19322035

Youtube Demo: https://youtu.be/qUMkEFlkiPc

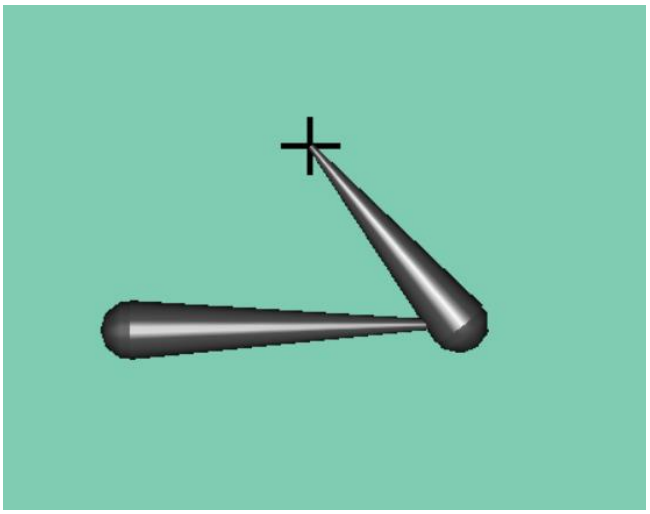
Mainly thank Professor Rachel McDonell who provides us this opportunity to study and guided us patiently. Also thank my peer, Mr. Shijun Zhang, who selflessly helped me debugging the code related to Complex inverse kinematics algorithms.

I know I submitted late, but I have never given up the struggling of fighting to the bugs. I have tried my best and daily working for 15 hours to study and implement each required functions and specifications which I could. I appreciate your patiently reviewing in advance.

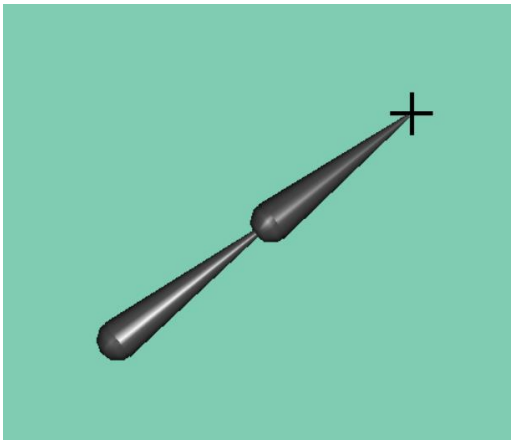
Create an arm and torso, where the arm should be constructed from a lower and upper arm, and attached to the torso.

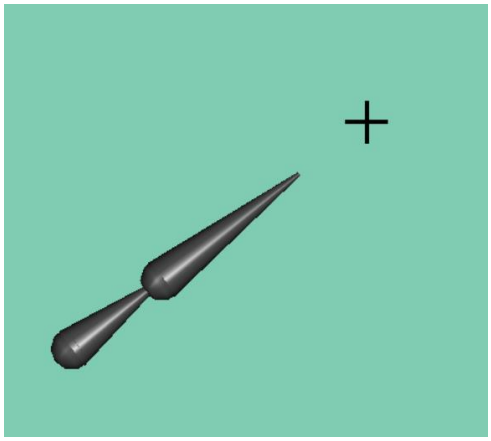


I have work out appropriate orientations for the lower and upper arm, given the desired position of the end effector.

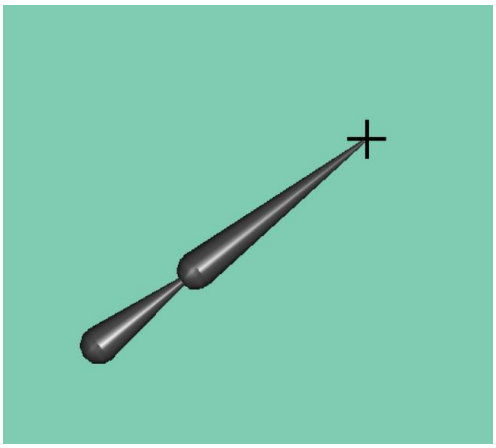


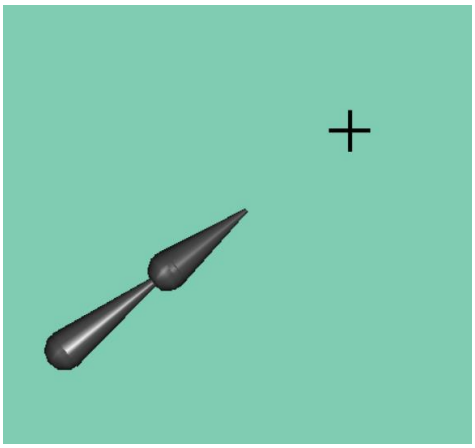
Reaching the desired position by adjusting the length of the first arm.



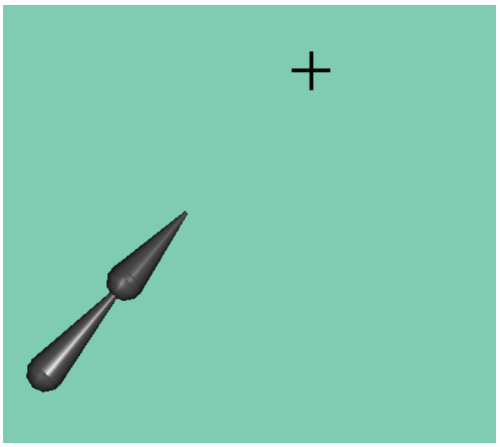


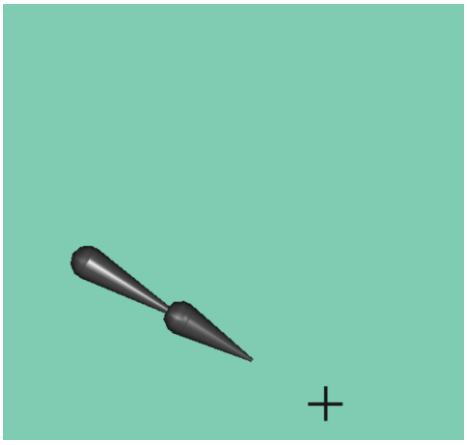
Reaching the desired position by adjusting the length of the second arm.



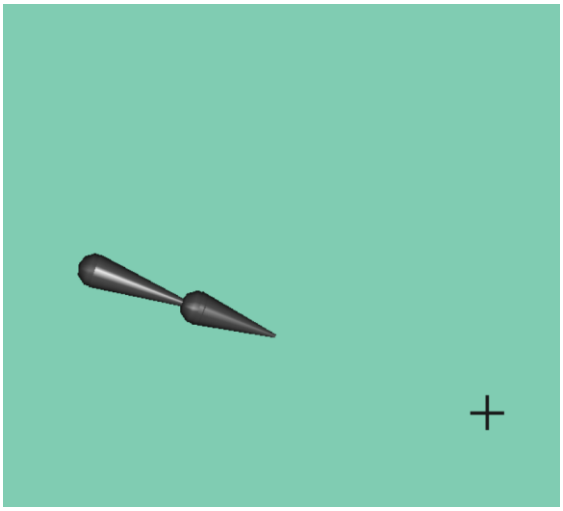


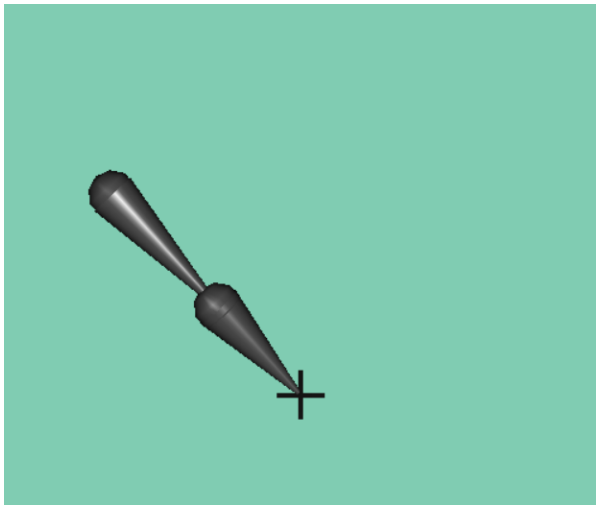
Reaching the desired position by rotating up and down



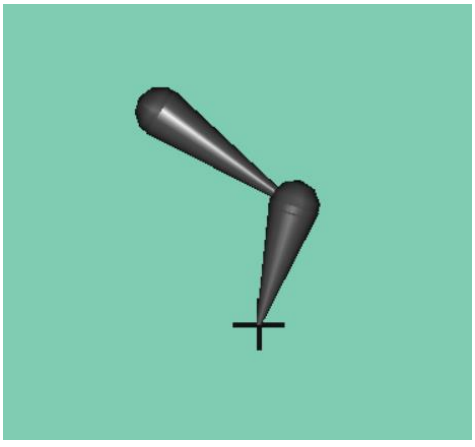


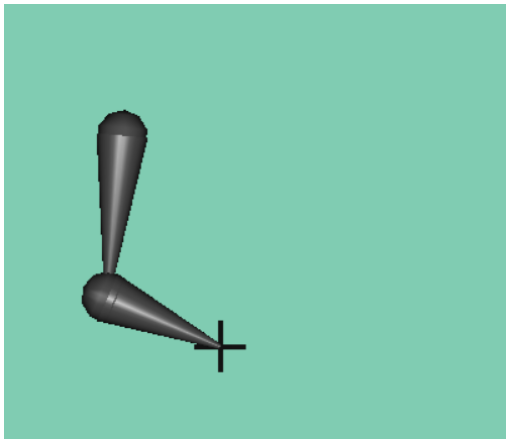
Reaching the desired position by moving left and right.



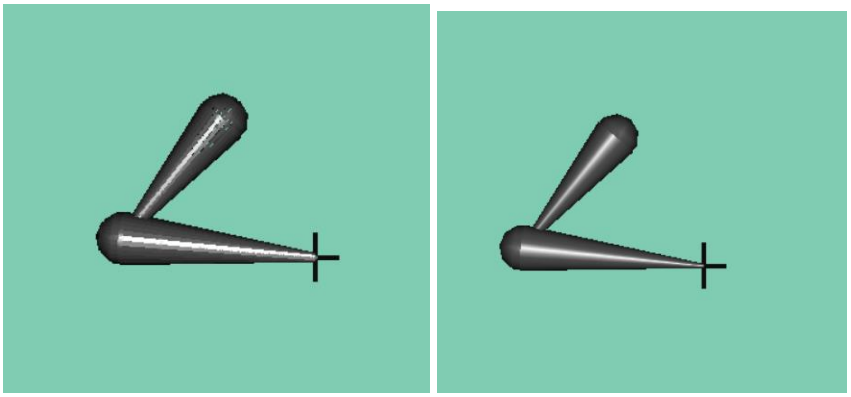


Show the alternative Inverse Kinematics approach.

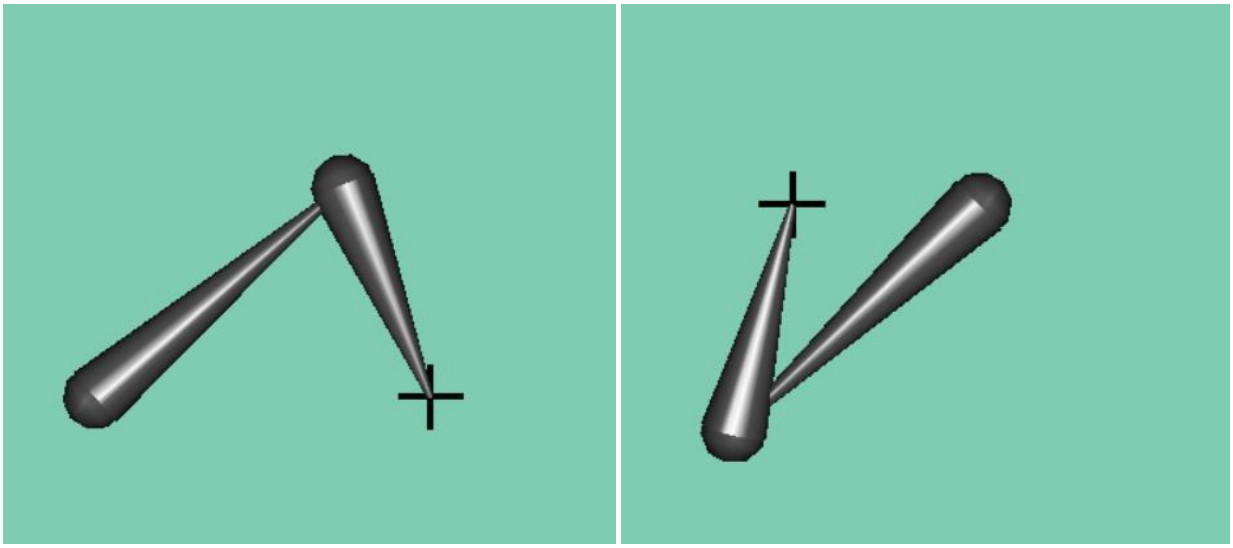




Shifting between polygon and wireframe appearances.



Added Constraints for reality movement.



I have also extended to more complex chains, other approaches like CCD.

```
103
104 void CCD(Node* node, vec3 TargetPos, int iterations = 100) {
105     while (iterations--) {
106         Node* currentNode = node;
107         while (currentNode->parent->parent != NULL) {
108             vec4 endPosition = currentNode->getEndPosition();
109             vec4 startPosition = currentNode->parent->getEndPosition();
110             vec3 toTarget = normalize(vec3(TargetPos.x - startPosition.x, TargetPos.y - startPosition.y, TargetPos.z - startPosition.z));
111             vec3 toEnd = normalize(vec3(endPosition.x - startPosition.x, endPosition.y - startPosition.y, endPosition.z - startPosition.z));
112             float cosine = dot(toEnd, toTarget);//Target with the current angle 目标与当前位置夹角
113             if (cosine < 0.99) {
114                 vec3 crossResult = cross(toEnd, toTarget);//get the rotation aixs 得到旋转轴
115                 float angle = glm::angle(toTarget, toEnd);//Include angle 夹角
116                 quat rotation = normalize(angleAxis(angle, crossResult));//get the quaternion for selecting 得到用于选择的四元数
117                 glm::vec3 euler = glm::eulerAngles(rotation);//rotating 转
118                 currentNode->setRotateAngle(euler.x, euler.y, euler.z);
119             }
120             currentNode = currentNode->parent;
121         }
122     }
123 }
```

References:

IK Demo by Dr. Ludovic Hoyet

Inverse Kinematics with OpenGL/Eigen3 : unstable Jacobian pseudoinverse
https://stackoverflow.com/questions/10115354/inverse-kinematics-with-opengl-eigen3-unstable-jacobian-pseudoinverse

Game Creation with XNA/Mathematics Physics/Inverse Kinematics
https://en.wikibooks.org/wiki/Game_Creation_with_XNA/Mathematics_Physics/Inverse_Kinematics#Jacobian_transpose_method

Cyclic Coordinate Descent in 2D https://www.ryanjuckett.com/cyclic-coordinate-descent-in-2d/

Mathematics for Inverse Kinematics http://graphics.cs.cmu.edu/nsp/course/15-464/Fall09/handouts/IK.pdf

傅老师游戏大师课(Mr. Fu's OpenGL tutorial, this is really helpful for Chinese Speakers.)
https://www.youtube.com/channel/UCPpDrebzulufI3G1TTeNu-g

OpenGL + C++: Modern Graphics for Groundbreaking Games | Udemy
https://www.udemy.com/course/graphics-with-modern-opengl/