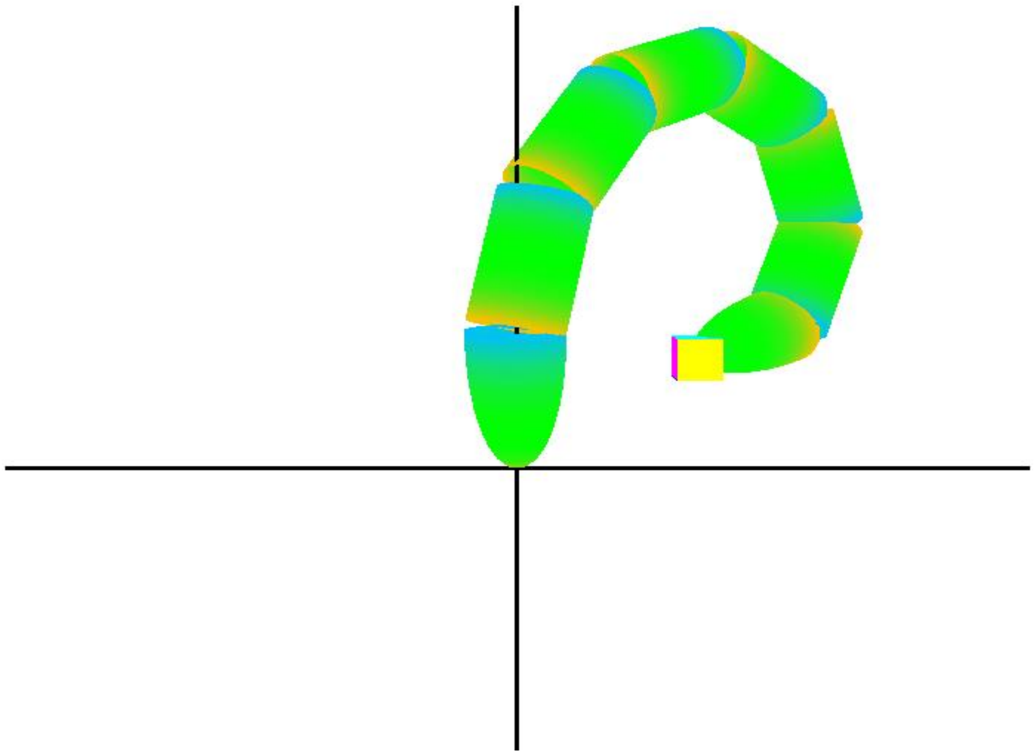
<u>Assignment 3 - Inverse Kinematics & skinning</u>

System initialization

- Download the code for the assignment
- Try to compile and run it on your computer. Try to clean and rebuild it.

Part A– IK



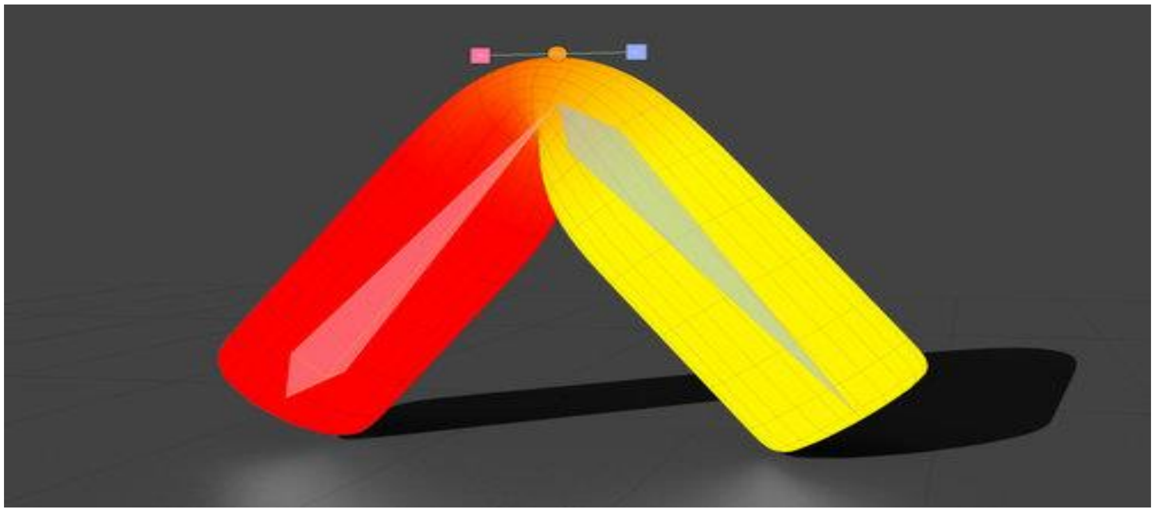
1. **Add to callback_key function:**
    - 'space' – starts and stops IK solver.
    - 'right and left arrows' – rotates picked link around the previous link Z axis (the first link will rotate around the scene z axis). When the box is picked move the box on the screen plane according to arrows. See also camera mode.
    - 'up and down arrows' – rotates picked link around the current X axis (use Euler angles). When the box is picked move the box on the screen plane according to arrows. See also camera mode.
    - 'b' pick the box if isn't picked else pick the first link. The first link is picked by default.

- 'n' changes the picked link to the next one. If the last link is picked picks the first one.
- 'p' changes the picked link to the previous one. If the first link is picked picks the last one.
- 'c' enters or exits camera mode. In camera mode the arrows will rotate the camera around the origin of the scene.

2. **IK implementation (Submission date 30/5/18)**
    - Each joint has 2 or 3 degrees of freedom (for your choice).
    - Implement FABRIK method for IK solver. In makeChange() and calculateStep(bool) Functions in IK class
    - Stop the solver when the distance between the chain tip and the destination is smaller than delta = 0.1. print the distance at when destination is reached.
    - Calculate Euler angles after each step or each time before rendering.
    - When calculating Euler angles from the joint positions (you get from FABRIK) recommended to calculate the rotation matrices from the initial position and not to search for the difference in each matrix. You can do it as follow:
        1. Start from the initial position when all matrices equal to the identity matrix.
        2. Calculate the vector V that perpendicular to Z axis and the vector

           Vp = Pi+1 – Pi (Pi will be (0,0,0))
        3. Find the angle between the X axis and the vector V and generate the first Z rotation matrix
        4. Find the angle between Z axis and Vp and generate the X rotation matrix
        5. You can choose how to change the last Z rotation matrix (It will be important in the next Part)
        6. Before calculating the next joint rotation matrix you must calculate Pi, Pi+1 in the previous coordinate system.
        7. Goto 2
    - When you use glm::acos(A) make sure A value is between -1 and 1.

Part B – Skinning (Submission date 6/6/18)



- You will implement skinning using Dual Quaternions Skinning algorithm on the Shader.
- Add attribute of (vec3) weights to the vertex shader. It will contain the weights of the previous joint (in the kinematic chain) and the two joints of the link.
- Add attributes of mat4 to the vertex shader for the transformation of the correlated joints.
- You may watch the shader language syntax video (link at useful link on the course site)

Submission

- Each part weight is 15% of the total course grade.
- Zip your project and rename it to <id1>_<id2>.zip
- Submit to the submission system before deadline.