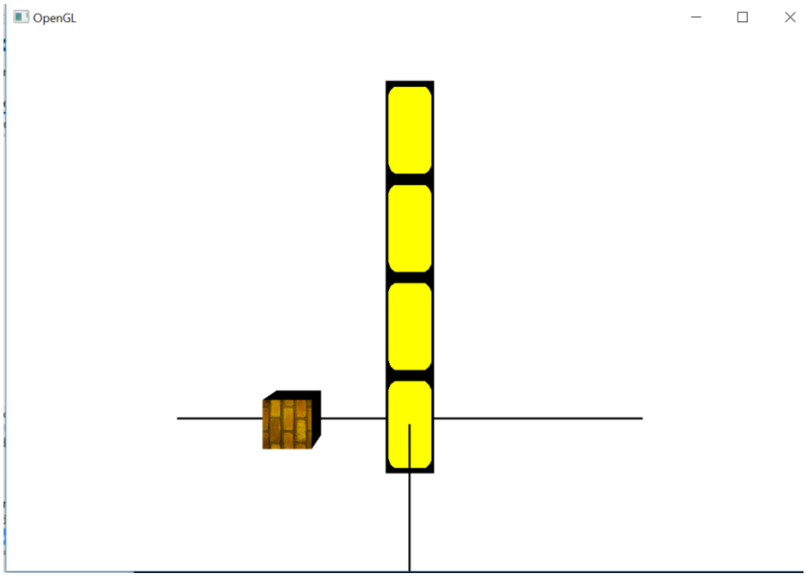
Assignment 3 - Kinematic Chain and Inverse Kinematics

Part 1 – System initialization

- Go to course site to useful links and download forClass3D.zip
- Try to compile and run it on your computer (The project was built on VS2015). Try to clean and rebuild it.
- Use the cube mesh to render a cube in (5,0,0) and chain of boxes scale is z direction by scaleFactor = 2.
- Rotate the scene around x axis by 90 degrees.
- Draw axis in the center of the scene before you draw the rest.



Part 2 – data structures (recommendation)

- Build data structure of at least 2 matrices (one for translation and one for rotation). This data structure will save the transformations you did for each object.
- Build data structure to whole scene which will hold the camera properties, boxes transformation, tip position (end of chain), destination position and Shaders.
- Use additional simple Shader for picking (Shader which draw each cube in a different color);

Part 3 – Requirements

1. **Add axis system for each joint** according to the previous link axis system
2. **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 nothing is picked rotate the whole scene.
- 'up and down arrows' – rotates picked link around the current X axis (use Euler angles). When nothing is picked rotate the whole scene.

3. **Picking implementation using mouse callbacks:**

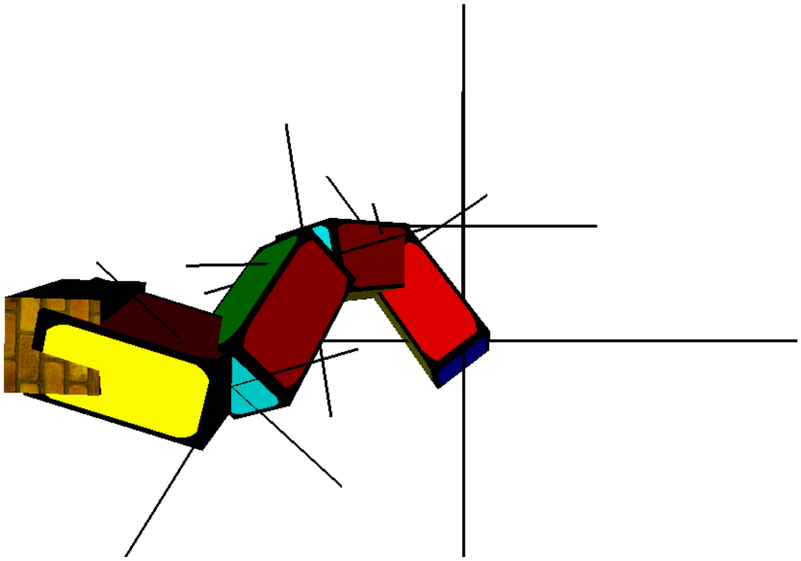
- Use color picking to decide which object is behind the pixel
- Object remain picked until another object is picked (user presses on it) or until the user press on the screen where there is no object.
- When left mouse button is pressed rotate picked link around the previous link Z axis (the first link will rotate around the scene z axis) when the cursor slides left or right and rotate picked link around the current X axis when the cursor slides up or down. When nothing is picked rotate the whole scene.
- When right mouse button is pressed translate destination cube or whole chain (according to picking) under the mouse cursor (always moves in the plane parallel to the screen plane.
- When scrolling up translate destination cube or whole chain (according to picking) away into the screen. When scrolling down picked object get closer.
- Don’t swap buffers after you render using the picking Shader.
- Don't forget to clean buffers before drawing.

4. **IK implementation:**

- Each joint has 2 or 3 degrees of freedom (for your choice).
- Implement one method for IK solver (Cyclic Coordinates Decent, recommended, or pseudo inverse based solver using Eigen library, see attached code).
- The chain tip reaches the destination when the distance between them is smaller than delta = 0.1. print the distance at when destination is reached.
- The solver will not start when the destination is too far. In this case it will print "cannot reach".
- Use small steps when calculating angles.
- Recommended not use Euler angles in the IK solver. We will check the Euler angle rotations only before the first use of the IK solver.
- When you use glm::acos(A) make sure A value is between -1 and 1.

5. **Bonuses:**

- 10 points bonus for implementing two method of IK solver (CCD and pseudo inverse based solver).
- 10 points bonus for implementing ray picking (in additional to color picking).



Part 4 – Submission

- If you did one of the bonuses or both add readme file which explain what you did and how to activate it.
- Zip your project and rename it to <id1>_<id2>.zip
- Submit to the submission system before deadline.