CA HW2 report

1. Fundamentals

   a、Global coordinates: the x, y, and z axis.

   b、Local coordinate: the point of view from one particle.

2. Implementation

   a、Forward Kinematics:

      i. Calculate local coordinate of current bone from parent local coordinate and rot_parent_current and bone rotation.

      ii. Calculate bone end point from local coordinate, bone direction, and bone length.

      iii. Use recursive DFS to traverse all the bones.

```
24      if (bone->idx == 0)
25          bone->start_position = posture.bone_translations[0], bone->end_position = posture.bone_translations[0],
26          bone->rotation = util::rotateDegreeZYX(posture.bone_rotations[0]).toRotationMatrix();
27
28      auto chi = bone->child;
29      while (chi != nullptr) {
30          chi->start_position = bone->end_position;
31          chi->rotation = bone->rotation * chi->rot_parent_current *
32                          util::rotateDegreeZYX(posture.bone_rotations[chi->idx]).toRotationMatrix();
33          chi->end_position =
34              chi->start_position + (chi->rotation * (chi->length * chi->dir) + posture.bone_translations[chi->idx]);
35          forwardSolver(posture, chi);
36          chi = chi->sibling;
37      }
38  }
```

   b、Time Warper

i. Scale new frame id to fit old frame count.

ii. Do linear interpolation for the bone translations.

iii. Do spherical linear interpolation for bone rotations.

```
float ratio = (float)allframe_old / (float)allframe_new;
int idx_a = std::floor(ratio * (float)i), idx_b = std::ceil(ratio * (float)i);
float t = ratio * (float)i - idx_a;

Eigen::Quaterniond rot_a(postures[idx_a].bone_rotations[j]);
Eigen::Quaterniond rot_b(postures[idx_b].bone_rotations[j]);
Eigen::Quaterniond rot = rot_a.slerp(t, rot_b);

Eigen::Vector4d new_rot(rot.x(), rot.y(), rot.z(), 0);

new_poseture.bone_rotations[j] = new_rot;
new_poseture.bone_translations[j] =
    (1 - t) * postures[idx_a].bone_translations[j] + t * postures[idx_b].bone_translations[j];
```

c、Result and discussion

i. Rotation order really matters: rotating in ZYX order produces a normal pose, while in XYZ order produces a really abnormal pose.

ii. I tried converting the rotations to quaternion then slerp, but the result seems weird. After then, I changed to slerp between euler angles, and the result is normal.

d、Conclusion

i. This homework needs more knowledge to complete than HW1. I think this is a challenging project, but I have fun finishing it.