# Forward Kinematics

2021.4.13

# Outline

- Demo
- Project overview
- Scoring criteria
- Objective and explanation
- Submission detail
- Hint and reminder

# Demo

- Forward Kinematics

# Demo (cont.)

- Time Warping（针对原本的动作做内插，使它能做到别的动作）

Before time warping

After time warping

# Project overview

- Solution layout
  - bin
  - assets
    - Shader and Texture
    - Acclaim - skeleton (asf) and motion (amc) files
    - Fonts - icon fonts
  - src (source code)
  - include (header files for src)
  - extern (project dependencies)
  - ForwardKinematics (Visual Studio project and main)

# Project overview (cont.)

- Environment
  - IDE: Visual studio 2017 / 2019
  - Platform: Windows

  - Graphics API: OpenGL
  - OpenGL Loading Library: glad2
  - OpenGL Toolkit: glfw
  - UI Library: dear imgui

  - Math Library: Eigen

# Project overview (cont.)

- src
  - acclaim (code for parsing acclaim files) 處理骨架檔和動作檔
  - grahpics (code for rendering geometries)
  - simulation (code for running simulation)
  - util (utilities)

- Everything you need to implement is in the simulation folder

# Scoring Criteria

- Forward kinematics - 50%
- Time warping - 30%
- Report - 20%
- Bonus - up to 15%

# Objective and explanation

- forwardSolver(posture, bone) → 把骨架檔、動作檔讀進來，做座標轉換
  - Convert motion data from joint space to the Cartesian space
- timeWarper(vector of posture, old keyframe, new keyframe)
  - Modify the given motion sequences with arbitrary profile

↳ 對 motion 做 time warping，使它的手到正確

# Objective and explanation (cont.)

- Report (below is a suggested outline)
  - Introduction/Motivation
  - Fundamentals
    - Describe local and global coordinates in your words
  - Implementation
  - Result and Discussion
  - Conclusion

解釋 local、global是什麼，如何轉換，
acclaim存的是什麼

# Submission detail

- Compress required files into a .zip file
  - Naming rule: CA2_StudentID.zip
    - e.g., CA2_309553010.zip

- Your zip file should contain following components
  - simulation/kinematics.cpp
  - Report in **PDF** format, no more than 10 pages

# Submission detail (cont.)

- Upload all your materials to new E3
  - No limit to the number of times of upload
  - The latest version is your final submission

# Submission detail (cont.)

- Late policies
  - Penalty of 10 points on each day after deadline
- Cheating policies
  - 0 points for any cheating on assignments
  - Allowing another student to examine your code is also considered as cheating
- Deadline
  - Monday, 2021/05/03, 23:55

# Hint and Reminder

- Blank template:

# Hint and Reminder (cont.)

- Eigen Library - Quick reference
  - Eigen::Vector4d (SIMD friendly)
    - We use 4D vector to represent 3D vector, so keep last dimension (3) = 0
    - You should use cross3() instead of cross()
    - 4d means 4D vector with double as internal type
  - Eigen::Affine3d - Affine transform
    - How to use?
  - Eigen::Quaterniond - Quaternion
    - Has an useful slerp() member function

作外积用这个

旋转用这个

# Hint and Reminder (cont.)

- namespace util
  - rotate{Degree|Radian}{XYZ|ZYX}
  - e.g. rotateDegreeXYZ(x, y, z) means: 一転 z→y→x
    - x, y, z are present in degree
    - rotate z degrees along z-axis first, then y degrees along y-axis, then x degrees along x-axis

# Hint and Reminder (cont.)

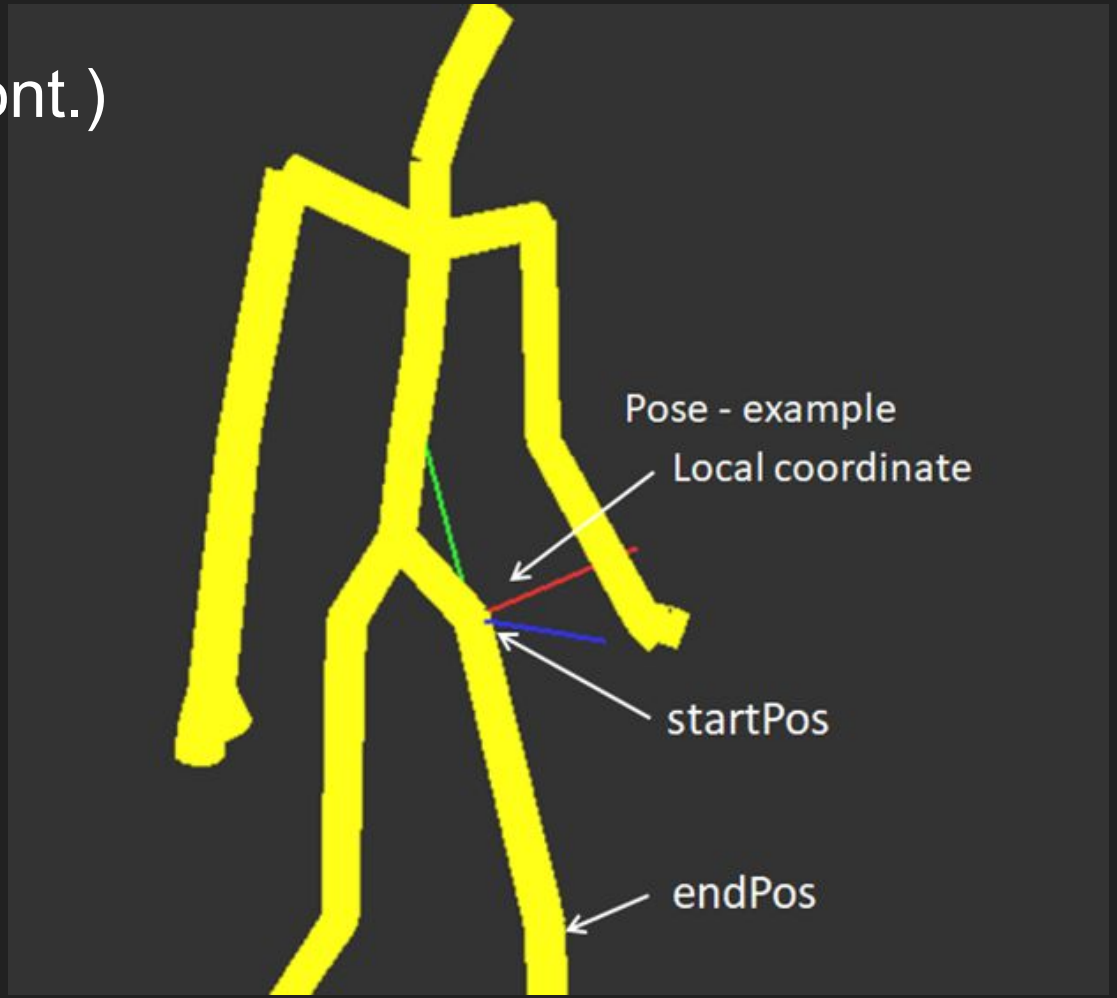
- Pose example
- Each bone has
  - local coordinate
  - start position
  - end position

由 root 一路連下去

只有 root 是絕對位置

其他都是相對 parent 的位置

(用 dfs, bfs)



# Hint and Reminder (cont.)

- forwardSolver:
  - You have to set each bone's global start and end position and rotation
    - Read local coordinate data from posture first
  - Hint: Maybe you can use DFS or BFS to traverse all bones

skeleton.cpp

constructor

讀檔

↓

rotate bone 2 local coord

(bone中的 dir 已經變成 local 的向量)

↓

compute rotation 2 parent coord

(bone中的 rot_parent_current 已經是 parent→child 的 rotation matrix)

motion.cpp

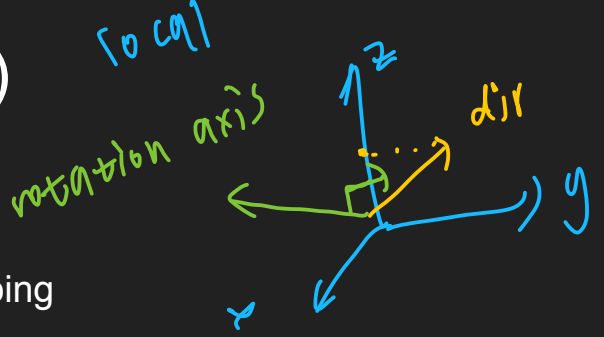
set bone transform

先call forwardsolver

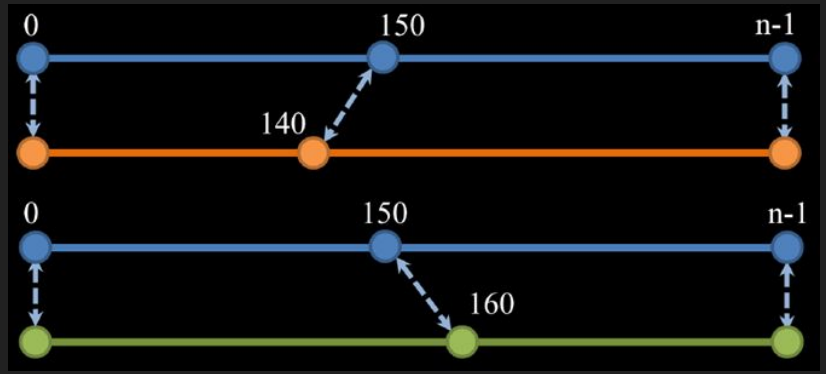
(會算出每個bone的global位置)

然後set model matrix

# Hint and Reminder (cont.)




- timeWarper:
  - You have to implement time warpping
    - If your implement is correct then the skeleton can catch the ball
  - Perform interpolation
    - Linear interpolation on translation
    - Spherical linear interpolation on rotation




平移可線性內插

旋轉不可線性內插

轉成 quaternion，用 slerp

# Hint and Reminder (cont.)

- How to properly report bonus?
  - Mention it in your report.
  - If your implementation violates with the original implementation, please make a toggle for switching.
  - If your bonus hides any original feature, you will get not get the score for the features that the TAs cannot test.

# Hint and Reminder (cont.)

- How to contact TA?
  - Please ask your question on new E3 forum.
    - or send email to **BOTH** TAs via new E3 if the question is personal.
  - If you need to ask question face-to-face, please send email for appointment.

  - IMPORTANT: please sort out and arrange your question, so we can help you without wasting time on trivial matters.