Inverse Kinematics

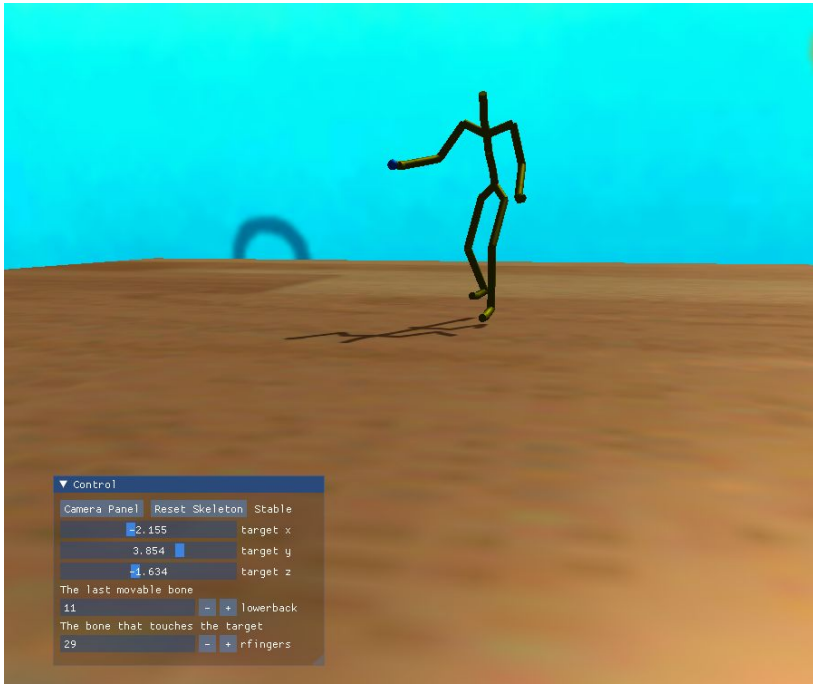
2023 Computer Animation and Special Effects

Outline

- Overview
- Objective
- Report
- Scoring
- Submission

Overview

- Use different bones to touch the ball
- Start bone
  - The last movable bone
- End bone
  - The bone that touches the ball



Demo link: https://youtu.be/52-auc1_QUk

Objective

- Everything you need to implement is in kinematics.cpp
- There are three functions you need to implement in this homework
  - void forwardSolver(...)
  - VectorXd pseudoInverseLinearSolver(...)
  - bool inverseJacobianIKSolver(...)
- Bonus
  - Return whether IK is stable so that the skeleton would not swing its hand in the air
  - Take rotation limit of bones into consideration in bool inverseJacobianIKSolver

Objective (cont.)

- void forwardSolver(...)
  - Goal
    - Implement forward kinematics, which is similiar to HW2

Objective (cont.)

- VectorXd pseudoInverseLinearSolver(...)
  - Goal
    - Find solution of linear least squares system, which will be needed for inverse kinematics
    - i.e. find x which min(| jacobian * x - target |)
  - Hint
    - You might use some pseudo-inverse methods such as SVD
    - There are some built-in functions in Eigen that you can use
    - Eigen::Matrixs4Xf means a matrix with 4 rows and unknown columns
      - Eigen::Matrix4Xf m(4, 10); // A matrix with 3 rows and 10 columns
    - Eigen::VectorXf means a vector with unknown size
      - Eigen::VectorXf v(10); // A vector with 10 elements

Objective (cont.)

- bool inverseJacobianIKSolver(...)
  - Goal
    - Implement inverse kinematics
    - We use inverse-Jacobain method in this homework
  - Hint
    - Review "kinematics.pptx" from p.20 - p.50
    - Review "acclaim_FK_IKnote.pdf" Inverse Kinematics part
    - Traverse from end bone to start bone
      - Make end bone touch the ball (target)
      - Start bone is the last movable bone, so you should stop at this bone
    - You can check struct Bone in bone.h

Report

- Suggested outline
  - Introduction/Motivation
  - Fundamentals
  - Implementation
  - Result and Discussion
    - How different step and epsilon affect the result
    - Touch the target or not
    - Least square solver
  - Bonus (Optional)
  - Conclusion

Scoring

- Forward kinematics - 0%
- Least square solver -20%
- Inverse kinematics - 60%
- Report - 20%
- Bonus - 10%

Submission

- Please upload only two files respectively
  - kinematics.cpp
  - report_< your student ID>.pdf
  - other necessary files (optional)
  - penalty of 5 points for incorrect submit format
- Late policies
  - Penalty of 10 points on each day after deadline
- Cheating policies
  - 0 points for any cheating on assignments
- Deadline
  - Sunday, 2023/05/14, 23:59