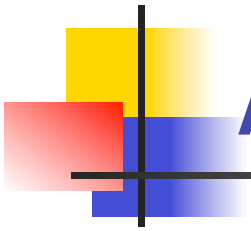
So you want to animate a headless stick figure walking ~~badly~~ well

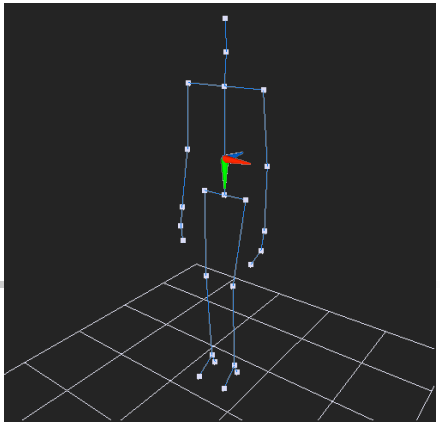
With the help of moCap data



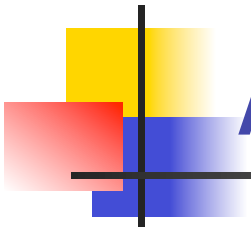
Assignment #4

- Which is something that you may wish to do since it is assignment #4.

Assignment #4



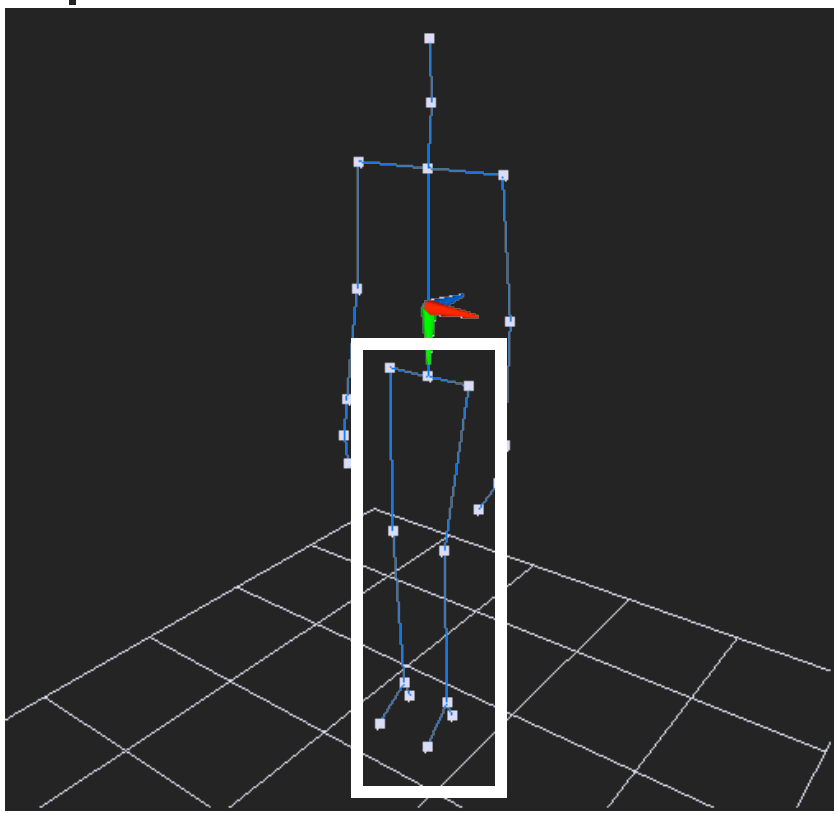
- Goal:
  - Simulate walking of a very simple two legged skeleton using keyframing
  - Read, interpret and apply motion capture data.
- In short:
  - You will be writing a BVH motion viewer.

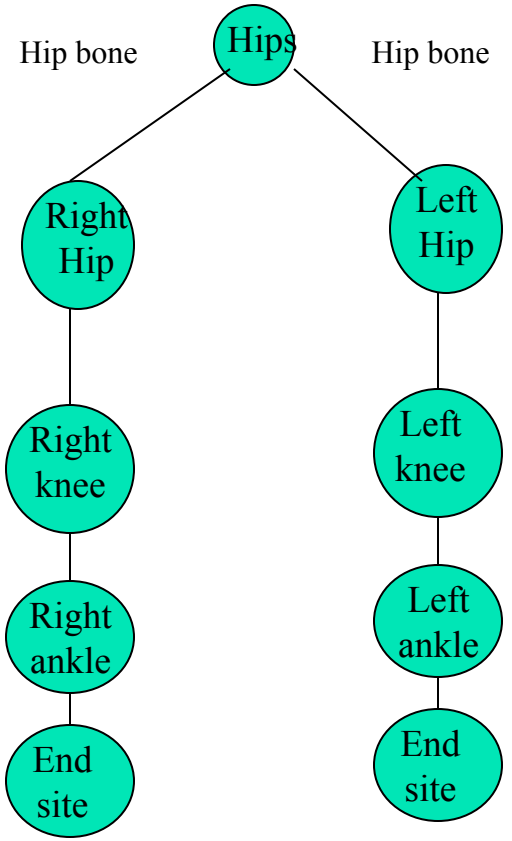


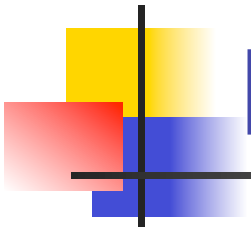
Assignment #4

- Program may be:
  - Interactive – show the animation in window on the screen
  - Batch – create an app that will produce a set of input files for a renderer.
- Usage:
  - `Motion <bvhfile>`

A look at the leg model

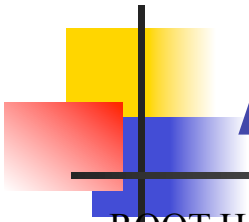






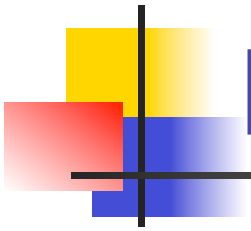
BVH data

- Info about the heirarchy will be found in the BVH file.
  - Sample bvh files on mycourses as well as ASSIGNMENTS section of course Web site.
  - More samples at bvhfiles.com (free registration required)



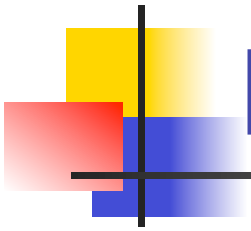
A look at the leg BVH

```
ROOT Hips
{
        OFFSET    0.000000    0.000000    0.000000
        CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
        JOINT LeftHip
        {
                OFFSET    4.740440    -3.195970   -0.322709
                CHANNELS 3 Zrotation Xrotation Yrotation
                JOINT LeftKnee
                {
                        OFFSET    -1.605680   -15.592300  1.005510
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        JOINT LeftAnkle
                        {
                                OFFSET    -0.632401   -16.421000  -1.413730
                                CHANNELS 3 Zrotation Xrotation Yrotation
                                End Site
                                {
                                        OFFSET    0.008423    -2.933000   -1.684880
                                }
                        }
                }
        }
```



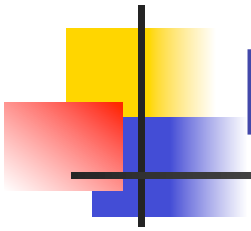
BVH structure data

- Sample BVH file also has data on the upper body / arms.
- Note: BVH file defines hierarchy and skeleton structure.



BVH motion data

- BVH file also defines sampling rate and motion data for each joint.
  - Playback should be at constant rate.
  - If at variable rate, use interpolation between samples.



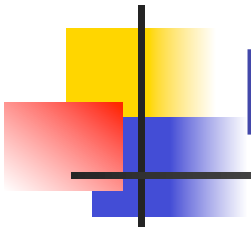
BVH Motion data

MOTION

Frames: 401

Frame Time: 0.016667

2.525769	37.627438	-5.008272
4.046218	-9.756648	-12.119053
-5.588056	5.988222	-0.216810
-0.573275	1.807009	4.602962
1.838464	0.356896	…



BVH Motion Data

- Each line is one sample of motion data.
- The numbers appear in the order of the channel specifications as the skeleton hierarchy was parsed.

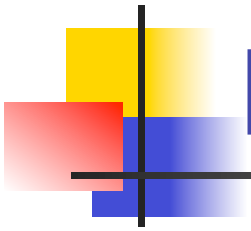
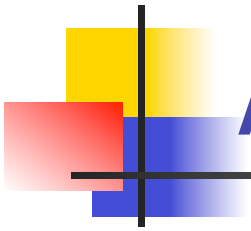


Figure Model

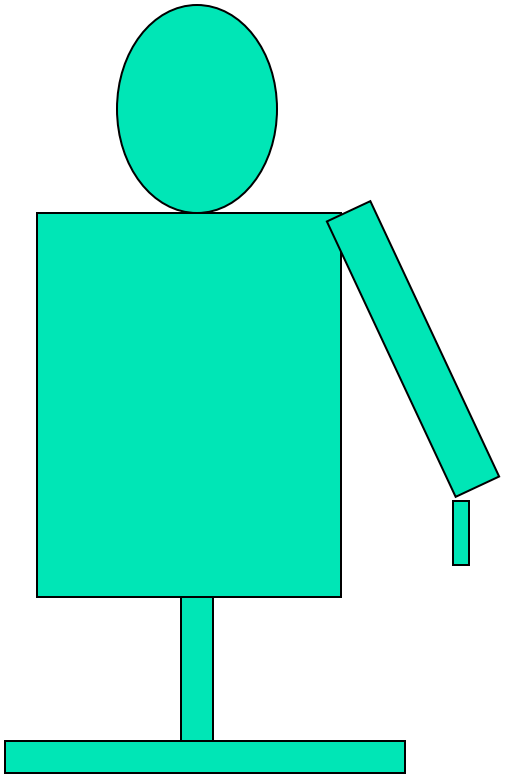
- Use simple geometry to model bones
  - Cylinders
  - Boxes
  - Lines
  - Spheres at joints (optional)
- Include a simple floor as a point of reference
  - Estimate placement by position of foot.

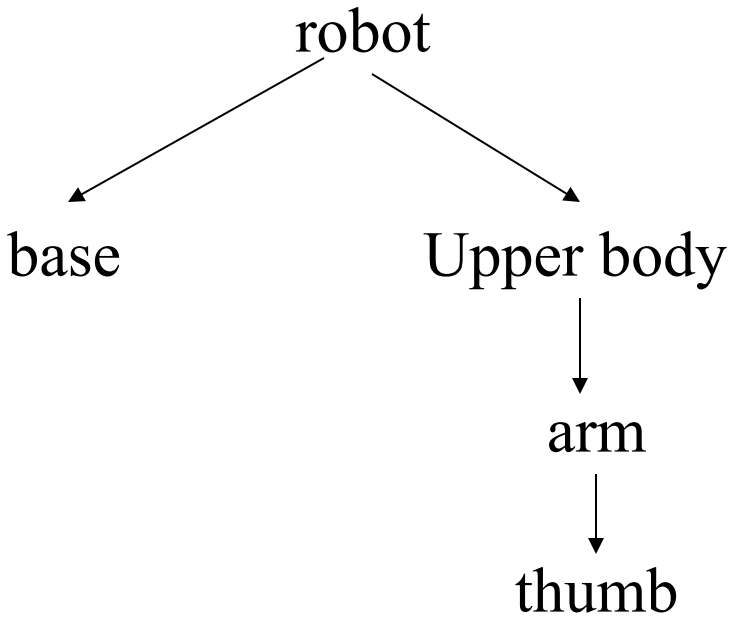


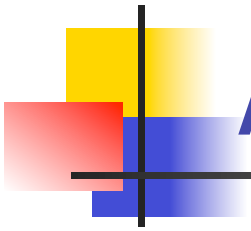
Articulated Figures

- Most rendering systems / API maintain a transformation matrix stack
  - Push when going into the hierarchy
  - Pop when leaving the hierarchy

Articulated Figures

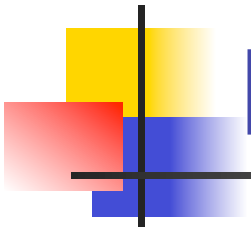






Articulated Figures

- We know how to transform of each component with respect to another component.
  - Use the matrix stack in order to calculate the local coordinates of each component.



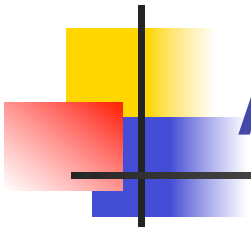
Drawing Articulated Figures

Define your camera orientation

Push Matrix
Concatenate transformation for robot as a whole

PushMatrix
Concatenate transformations for robot base wrt the center of the robot
Draw robot base
Pop Matrix

Push matrix
Concatenate transformations for robot body wrt the center of the robot
Draw robot body
…



Articulated Figures

Push Matrix

Concatenate Transformations of Arm wrt body

Draw arm

Push Matrix

Concatenate Transformation of Thumb wrt Arm

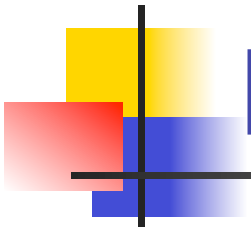
Draw thumb

Pop Matrix // Thumb

Pop Matrix // Arm

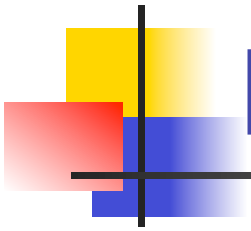
Pop Matrix // body

Pop Matrix // robot



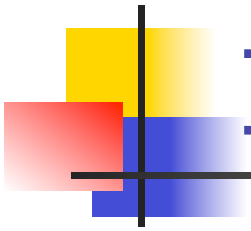
Bells and Whistles

- 20 points
  - Basic read / playback of data
- Extras
  - 5 points
    - Interactive control of “time” / playback
  - 5 points
    - Include ability to speed up / slow down motion
    - Change sampling rate
    - No filtering required



Notes

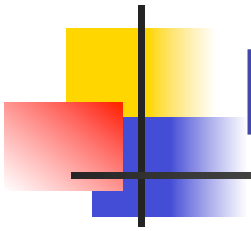
- Try several BVH files
  - Grading will be done only a variety of samples.
  - I.e. Don’t hardcode hierarchy



Important Web sites

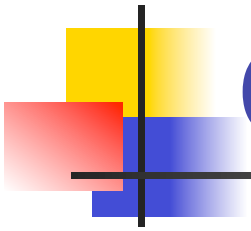
- All linked from course Web site (ASSIGNMENTS) and mycourses.

- Tips on reading and interpretting BVH files:
  - http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html
- BVH samples:
  - http://www.centralsource.com/blender/bvh/files.htm
  - http://www.cs.rit.edu/~jmg/animation/bvhdata



Due dates

- Submission
  - Due February 9th
  - Via myCourses
  - Please include documentation on
    - how to run your app
    - How to build your app
      - Makefile
      - Visual Studio (.dws, and .dsp files)
    - Renderer used if batch
    - Platform (sun vs. Windows vs. Mac)
  - Submit as a single zip or tar file



Questions?

- Next time:
  - No lecture Wedensday
- Next Monday:
  - Advanced algorithms for articulated figure motion
  - Questions?