Notes

- Projects: start as soon as you can!
  - Try to get results **early** which you can continue to improve up until the deadline
  - Feel free to consult with me if you have problems

Graph traversal

- Be careful with coordinate systems
- Global position and orientation recorded with the data is irrelevant
  - When we transition into a new clip, need to align with where we're coming from
- As you go, enforce constraints (foot plants) in new coordinate system

Selecting walks in motion graph

- Depends on application
  - E.g. Snap Together Motion from assignment 3: direct correspondence between user controls and arcs to take
- More generally, allow user input to be higher level: make the character walk here somehow
  - Boils down to optimization over a graph
  - Exhaustive search too expensive: need heuristics
  - Example: branch and bound, other techniques from AI
  - More complex: try to organize the structure of the graph into a smaller summary graph, exploit that

Remaining issues

- We're still limited by data in the mocap sequence: what if we want to create a new jump?
- Next step is trying to parameterize motion: give user meaningful sliders
  - I want a jump this high and this far, even if I never recorded one quite like that
- Also: the unexpected
  - If the character's interaction with the environment isn't carefully controlled, we could be in trouble
  - This leads to kinematic or physical controllers

Spacetime constraints

- Before we get caught up in altering mocap data etc. look at one fundamental approach to motion:
  - Witkin, Kass, "Spacetime Constraints", SIGGRAPH'88
- Motivation: let computer calculate character motion automatically from higher level description

The set-up

- Define the physical properties of the character (e.g. masses, inertia tensors)
- Forward simulation is then straightforward (and results should look realistic)
- Problem: figuring out the right initial conditions, the right muscle forces to apply at the right time, etc.
  - Much harder than simple keyframing
- Approach here:
  - Specify constraints on motion (e.g. jump at time 1, land at time 2 at this position)
  - Treat forces in time as unknowns, solve for those that satisfy user constraints in the "best" way

Simple particle example

- Motion of a particle with additional controllable force
- Discretize time into N steps
- At each step, control force vector is unknown: 3N variables
- F=ma tells us how positions depend on control forces
- Constrain positions at start and end (for example): equations that must be true
- Specify objective to minimize
  - E.g. 2-norm of power
- Then run your favourite optimization routine

Character animation

- Motion is somewhat more complicated - e.g. Lagrangian dynamics
- Unknowns generally correspond to muscle torques
  - Also contacts (e.g. with ground) give rise to unknown contact forces
  - Can in fact implement articulated figure dynamics by solving for unknown joint constraint forces --- Lagrange Multipliers
- Bigger system, but same procedure

Objective

- Lots of continuing debate…
- Simple kinematic smoothness not so good
  - Human motion is far from optimally smooth
- Efficiency of motion is one possibility: use the minimum energy needed
  - But may lead to improbable strategies…
- Robustness to perturbations another plausible objective
  - Some evidence that this is true in some tasks
- Reality is more complicated
  - Learned motions, feedback control, …