Computer Animation & Gaming
Final Project

Topic

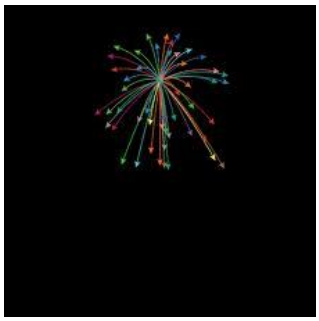
Three suggested topics are listed below. You are free to choose any one of them. All of them are introduced in the later chapters of the textbook, for which we may or may not have time to cover them in class. If you are not forming a team with another member, these topics have difficulties suitable for you. If you are forming a team of two members to work on the project, then the expected difficulty and effort should be proportionally increased. It is also okay to come up with your own topic, as long as it is related to computer animation. Since you will spend a significant amount of time working on this project, you should choose something both interesting and challenging to you.
This is meant to be open-ended, and allow you to pursue almost any related topic.

- *Integration with Research:* You are strongly encouraged to pick a topic that ties in closely with your research work. This project might give you a chance, for example, to initially explore a potential thesis topic. Or, you might use it as a chance to explore some idea you came up with while performing research but haven't been able to work on.
- *Originality:* Your project does not have to be a completely new idea. An example would be implementing a method or algorithm described in the textbook or an existing paper. More original work would include merging topics found in several papers, systems, games, or exploring a completely new topic.
- *Possible Topics:* Attached are three examples of possible topics, with difficulty suitable for single member. These are meant to give you ideas, not to be a comprehensive list of topics. There are also a bunch of connections between computer animation with computer games, modeling, and computer graphics. If you are uncertain whether your topic would be appropriate, please check with the instructor.

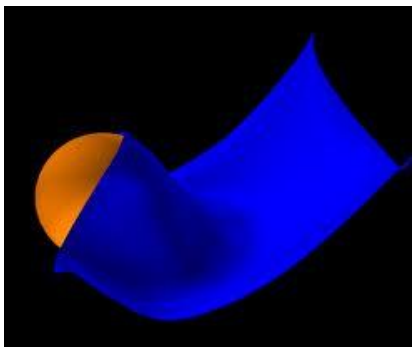
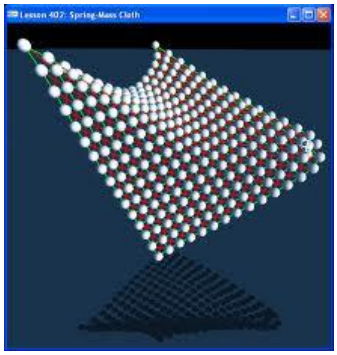
Three Example Topics:

1. **Particle System:** Implement a particle system with about 1000 particles. Use constrained randomization to set the source position, the direction, the initial velocity, terminal age, and the color. Use simple physics (i.e. gravity, wind, viscosity, friction) to control the motion of the particles. Add at least one interesting attribute to the basic particle system:

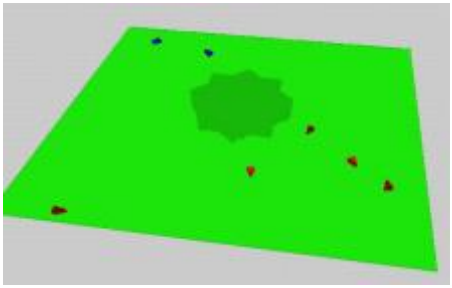
    - When particles hit the ground-plane (i.e. y <= 0) use kinematic collision response to bounce them off the ground with some damping.
    - Use the color and shape of the particles to create an effect of fire.
    - create some kind of environment such as a flame thrower, volcano, or water fountain.



2. **Spring-Mass-Damper Cloth:** Implement a 10x10 spring-mass-damper mesh to simulate a cloth: the mesh should be in the x-y plane. Fix one or two corner vertices in space. Implement gravity and a pseudo-random wind field as global forces. Getting a spring-damper system to behave the way you want it will require trial-and-error to set the appropriate values for time-delta, mass, spring constant, damper constant, and wind strength. Some suggestions:
   - code it so you can easily change the resolution of the 'cloth' (helps to use low-res cloth when debugging)
   - each vertex has the following attributes:
     - mass
     - current position
     - current velocity
     - accumulated force
     - 'fixed' flag (set if this vertex is not allowed to move)
   - each spring has the following attributes:
     - pointer to the two masses it attaches to
     - spring constant
     - damper constant
     - rest length (length in original configuration, for example)

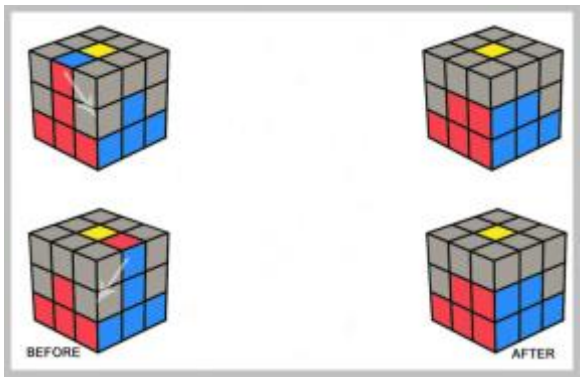
3. **Prey-Predator Behavioral Model:** Implement a prey-predator model in a simple 3D world where the creatures move on a 2D ground plane. Create two sets of creatures: prey and predator. Whenever a predator gets within epsilon of a prey, the prey is eaten (disappears). Give the two groups different capabilities (vision, acceleration, turning radius, etc.) to make the resulting animation interesting. Include spatial limits on the environment and at least one simple obstacle. A bounding sphere can be used in the avoidance calculation of the obstacle. Some suggestions:
   - Use forces to control the motion of the creatures:
     - The wall and obstacle will have repulsive forces associated with them.
     - for a prey (predator), a predator (prey) will have a repulsive (attractive) force associated with it - if it's within view.
   - For each creature type, keep the following information:
     - idle velocity
     - chase/escape velocity
     - vision angle / depth
   - For each creature, keep the following information:
     - type of creature (prey or predator or already-eaten)
     - current position
     - current velocity
     - current orientation (angle or vector)
   - Use some controlled randomness to individualize the creatures - at least in their initial values.



Some Example Projects from Past Years:

Marionette Control on iOS Devices

Golf-Bowling Game with Kinect



Solving Rubik's Cube

Motion-Marching Dancing Game with Kinect


Puppet Control with Hand Gesture using LeapMotion

Project Proposal

Your project proposal should consist of the equivalent of no more than two printed pages, and should be submitted on eLearning before the due time. Your proposal should include the following items:

- *Title* for project
- Your *Name*
- *Summary* – You should summarize the following items:
  - *Description* – What is it you are trying to solve/address?
  - *Importance* – Why is this problem important/interesting?
  - *Your Proposal* – What is it you plan to do?
  - *Relationship to Computer Animation (optional)* – If it is not obvious how your topic relates to computer animation, you should explain it here.
- *Goals* – You should give a list of final goals, specifying what you hope to accomplish by the end of the semester. Your goals should be as specific as possible.
- *Work Breakdown* – You should write a detailed list of the activities that needs to be performed to complete this project, estimate the time required for each, and

assign people (if team project with two members) to each of these tasks. One of the challenges in student team projects is ensuring that all team members have a shared understanding of the tasks required to complete the project, as well as knowing which tasks have been assigned to which people. If the work breakdown is accurate, and you successfully complete the tasks by the specified times, it prevents the typical end-of-the-semester crunch.