Workshop 1: Particle Cannon

Force and Motion

Remember from last week's lecture:

Basic 2-3 Force: sin and cos

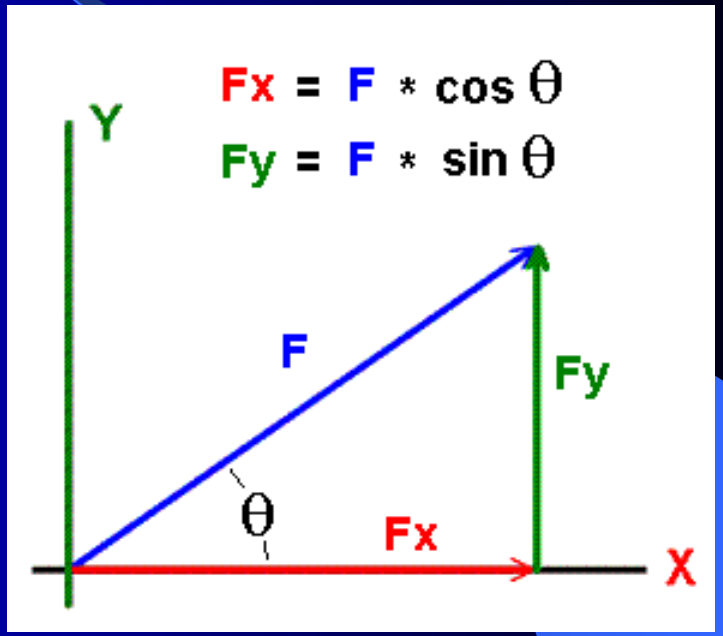
***As mentioned in the previous class, you need to understand basic forces, sin and cos fluently. Ask the teaching assistant for help if it is not clear.

Recall the way to do force calculations:

- Split each force into its x,y,z components

- If you have 1 or more sources of force on an object, the "resultant" force is simply the summation of the x,y,z components from each source.

- If you had sources of force from wind, gravity, and friction you can split the force from each into x,y,z components and then sum all of them to predict the object's movement.



Double Buffering 1

In the current basic code, the objects are rendered immediately using the flush() command.

This can cause artifacts (rips and tears and flashing polygons) due to

(1) the rendering process and also from

(2) the interaction with the graphics card refresh rate.

Double Buffering 2

So, in serious applications, a technique is used called

Double Buffering

In Double Buffering, we simply

(1) write our 3D world to an unseen buffer and then

(2) swap the display buffer with the unseen buffer.

Double Buffering 3

To do this in OpenGL/GLUT, we add one line to main:

glutInitDisplayMode(GLUT_DOUBLE);

which tells GLUT to use double buffering

and instead of the glFlush() command, we call

glutSwapBuffers();

and now everything should render nicely.

Gravity, Acceleration and Forces 1

In many computer graphics situations, using gravity or in general forces are either necessary or simply aesthetically nice.

Simple Usage:

- kicking a football
- dropping an item
- piloting a spaceship

Sophisticated Usages:

- falling leaves (gravity+wind)
- walking Terminator robot
- driving a racecar
- swimming
- collisions between two objects

Gravity, Acceleration and Forces 2 (basics)

v = velocity = change in location (x) of an object per time

v = velocity = $\Delta x/\Delta t$ or in shorthand: $x = v*t$

a = acceleration = change in velocity of an object per time

a = acceleration = $\Delta v/\Delta t$ or in shorthand: $v = a*t$

average velocity is $(v_2-v_1)/(t_2-t_1)$

average acceleration is $(a_2-a_1)/(t_2-t_1)$

Force is directly tied to acceleration

$f = m*a$ or $a = (f/m)$

Gravity, Acceleration and Forces 3

Simulating forces can result in nice effects and movement. Recall

$f = G * (m_1 * m_2)/(r*r)$

where
G = *Gravitational Constant* (see Wikipedia)
m_1 and m_2 are the masses of two objects and
r is the distance between them

If Y is pointing up, then gravity can be simulated by a change in velocity
Very rough estimate (ideally it should be done differentially):
$\Delta v/\Delta t = (v_{new} - v_{old})/(t_{new} - t_{old}) = a$
$v_{new} = v_{old} + a * (t_{new} - t_{old})$

NewVelocity = OldVelocity + a * t (where a = accel of gravity and t = time)
Since gravity is pushing downwards:

NewVelocity = OldVelocity - a * t

Workshops and Assignments

- Each workshop is only a small part of the overall grade – they are meant to provide motivation as learning tools and also for exploring CG. So, don't worry about finishing all workshops completely – just do your best and *study/explore the preworkshops carefully*.

- Ask the teaching assistants for the size of teams allowed. For workshops and all assignments, note that each and every student must make their own submission on the LML Course Manager.

  If teams are allowed and John and Mary do the workshop as a team then John must make a submission and also Mary must make a submission.

- The submissions are part of the agreement that the work is yours and that you take credit (or blame in the case of plagiarism). **Every student must always make their own submission.**

- Example: If John assumes that Mary's submission includes his; and does not make his own submission, his grade will be zero.

Submissions

The deadline is at the end of class, but if necessary, they may be submitted by 11:59pm of the day of the Workshop. Place in a ZIP file the following and submit on the LML Course Manager:

The top level of the zip file should contain a directory called ***firstname.lastname.project*** as described below:

(1) a file named "AnswerJournal.txt" which should list
- The full names of the students you worked with.
  - *John Doe and Mary Smith*
- The name of the machine you had it working on.
  - *0009747*
- Mention which of the problems you solved.
  - *Solved problems 1 and 2. Partially working prob. 3*

(2) The source code, Makefile and

(3) Working executable of your solution