

*Syllabus*  
**CSC – CPE 474**  
**Computer Animation**



**Professor:** Christian Eckhardt  
**Office:** Building 14, room 221  
**Phone:** 756-2416  
**office hours:** see PolyLearn top section  
**email:** ceckhard@calpoly.edu  
**Schedule:** see PolyLearn top section

**General:**

Welcome to computer animation! This course will build on top of CSC/CPE 471 (Introduction to Computer Graphics), and will teach you the basics of animation principles. This course requires substantial math and programming skills. Experience with C or C++ will be essential, and experience with linear algebra and calculus will be required.

**Development Environment:**

We will be using

- *C++11* as the main language
- *Modern OpenGL/GLSL* as the graphics API
- *GLEW* for cross-platform initialization of OpenGL
- *GLFW* for windowing
- *Eigen* for linear algebra
- *Ceres* for nonlinear least squares
- *Visual Studio & CMake* for cross-platform compilation and development
- *Perception Neuron* full body mocap suit

**Final Grade**

- 16%: 10-12 labs
- 24%: 4 Programming assignments
- 30%: 2 Midterms
- 30%: Final project (2 x 7.5 for milestones, 15 for the final presentation)

**Objectives:**

- Understand spline curves: continuity, tangents, curvature, arc-length parameterization, curve fitting
- Better understanding of 3D rotations including interpolation of orientations
- Understand bilinear interpolation and 2D mesh deformation
- Understand vertex skinning on the CPU and the GPU
- Apply nonlinear least squares to solve inverse kinematics
- Understand particle dynamics and basic integration schemes for ordinary differential equations
- Simulate simple cloth system with collisions
- Learn and use some C++11 features
- Learn the basics of CMake

**Labs:**

Each lab exercise is due within two lab sessions. For example, if a lab is assigned on Wednesday, it must be completed by the end of Friday's lab. Labs must be checked off by me personally during the lab unless you have my prior permission.

**Assignments:**

There will be four substantial programming assignments. If your program is late you will lose 20% within first 24 hours after deadline, 40% within 48 hours, 100% after 48 hours. However, you get 2 \*free\* days for the entire quarter which can be applied to the four programming assignments only. You do not need to explain why you are using the days—these two late days will be automatically applied to any late assignments. After your two late days have been used up, the late penalties apply.

**Midterms:**

There will be 2 written midterms. You may bring a double-sided sheet of hand-written notes.

**Project:**

You will get roughly 5 weeks at the second half of the quarter to work on the final project. You may use any code base for the project, including mobile and web-based.

## Participation

I expect you to participate in class and engage with the class material (studies suggest that taking longhand notes is one of the better ways to guarantee your engagement with the material in class) )<sup>1</sup> I also expect you to form a community of scholars for the duration of the quarter (and hopefully longer). My teaching style is very interactive – if you want to know more about why see (<sup>2</sup>). **Laptops** have been shown to be distracting in lecture<sup>3</sup> and **are not allowed unless specified (or a specific exception is negotiated) -- same for cell phones.** There also might be random pop quizzes during the class as a part of participation.

## Honesty

Although I encourage you to have lively discussions with one another, all work you hand in must be your own work. If your program or parts of your program are plagiarized from another student or unapproved sources including tutorials, you will fail the course and a letter will be put in your file with Cal Poly Judicial Affairs. Note some old tutorials do not use modern graphics – if you use them, this can result in problems. You can talk to one another about your solutions and you may look at another student's code that has a bug (I encourage you to help each other with de-bugging), but you cannot look at someone else's working code.

Note that I expect your OpenGL code to conform to at least OpenGL 3.0 standards (sometimes referred to as “modern graphics”) some specifics include no use of immediate mode for rendering and no OpenGL matrix stack calls (instead we will be using glm for a matrix library) and all shading will be computed using GLSL shaders. Your code must compile and run on the machines in the CSL.

## Disturbing Behavior

Students with disturbing or threatening behavior will fail the class in order to restore a reasonable learning environment for all the other students.

Only emails beginning with a proper greeting and a corresponding ending will be answered.

---

<sup>1</sup> <http://www.theatlantic.com/technology/archive/2014/05/to-remember-a-lecture-better-take-notes-by-hand/361478/>

<sup>2</sup> Applying the Seven Principles for Good Practice in Undergraduate Education" (1991) Chickering and Gamson

<sup>3</sup> <http://www.yorku.ca/ncepeda/laptopFAQ.html>