CSC – CPE 476: Real-time 3D Computer Graphics Software Systems
Syllabus

Professor: Zoë Wood
Office: Building 14, room 209
Phone: 756-5540
office hours: TTh 12-1:30pm & W by apt,
email: zwood@calpoly.edu
Schedule: Lecture TTh 1:40-3:00pm (20-129) Lab 3:10-4:30pm (20-127)

General: Welcome to 3D gaming. This course will teach you *some* of the important *computer graphics principals* of 3D games. This course is primarily focused on the graphics components of interactive 3D games/worlds. We will cover advanced real-time graphics techniques mostly in the context of games. **This course requires substantial math and programming skills.** Experience with C or C++ will be essential and experience with linear algebra will be very helpful. We will be using OpenGL and GLSL for our graphics APIs, along with C++ to create computer graphics games throughout the quarter (You are required to use 'modern graphics programming' for all assignments*). You are welcome to develop your programs under varying operating systems as long as the final programs can be demonstrated and run on multiple machines for the final game play demo.

Course Objectives, by the end of the quarter students will:

- Understand and master the graphics pipeline and the basic implementation of the pipeline in modern hardware (and graphics libraries)
- Understand programmatic choices related to geometry in real-time games, including understanding basic vocabulary, general computation and understanding of trade-offs of:
  - Various spatial data-structures
  - Various culling algorithms
  - Various geometric representations (polygonal, volume, parametric, etc.)
- Understand programmatic choices related to lighting and shading in real-time games, including understanding basic vocabulary, general computation and understanding of trade-offs of:
  - Various global illumination algorithms (i.e. shadow algorithms, ambient occlusion, etc.)
  - Various BRDFs and deferred shading
  - Related technologies, including texture mapping and framebuffers
- Understand programmatic choices related to animation in real-time games, including understanding basic vocabulary, general computation and understanding of trade-offs of:
  - Introductory physically-based modeling
  - Character animation (specifically skinned meshes)

- Be able to program a basic game with multiple moving components and interaction, shadow mapping, view frustum culling and two advanced graphics technologies from the provided list
- Write a large C++ real-time computer graphics application either as an individual or on a team and experience the joy of software engineering while working on a larger quarter long project.

Assignments:

- 1 pair program (5% of final grade)
  - OpenGL & C++ application
- 2 follow-on individual programs (vfc and shadow mapping – 5% each)
- One large team or individual final programming project (45% of final grade)
  - of your team's choice (again using OpenGL and C++)
  - project must be approved by the instructor (see final project proposal & rubric)
  - teams will be 1-6 people – see instructor for exceptions
  - all teams must meet in quarter deadlines (see syllabus for tentative deadlines)
  - Note that individual assessment (grades) will be made via input from team members for each milestone.
- 2 individual programming technologies (14% of final grade, 7% each)
  - OpenGL & C++ technologies integrated into group project
  - Code review required
- 2 mid-term exams (10% of final grade each)
- Final game play assessment (2% of final grade)
- Participation (4% of final grade)
  - attend class/ talk in class or office hours interaction

Please see the program description for deadline details. There is a strict late policy for all assignments – **no late programs/project demos will be accepted.**

Text: "Real-time rendering" Tomas Akenine-Moller and Eric Hanes (highly recommended)

Recommended: Any good modern graphics OpenGL reference, (e.g. "Foundations of 3D Computer Graphics" by S. Gortler) "Making Comics" and "Understanding Comics" by Scott McCloud

Participation: Throughout the course, there will be many opportunities to participate interactively in and out of the classroom. I expect you to participate by asking or answering questions, either in class or online on piazza. To encourage in-class participation, laptops, phones, and other smart devices are not to be used during class, unless specified for workshop format classes.

Honesty: Although I encourage you to have lively discussions with one another, <u>all work you hand in must be your own work</u>. 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 use glm or eigen for a matrix library) and all shading will be computed using GLSL shaders.

The following schedule for the lectures and assignments may change and is provided to give you a rough outline of the topics we will cover and the timings of your final project reviews. Check reading chapters with the topic in case your book edition varies.

Week 1	1/10	Introduction	Lab 1
	1/12	Graphics pipeline review – game loop/time based movement	
	Read	*Chpt. 2 from RTR (& Chpt. 3 if necessary)*	**Marketplace to for teams**
Week 2	1/17	Geometry in games – characters, terrain & acceleration	**Project Proposal D**
		Compare: OO vs. entity systems (links on polylearn)	Team meetings unle specified
	1/19	Animation basic (character vs. physically based)	
		Game design exercise	**Program 1 due**
Week 3	1/24	Performance – spatial data structures	
	Read	*Chpt. 14 from RTR*	
	1/26	Performance – spatial data structures	**25% Final project**
	Read	*Chpt. 14 from RTR*	
Week 4	1/31	Lighting review & texture	
	Read	*Chpt. 6 & 9*	
	2/2	Global illumination – shadows & precomputation	
	Read	*Chpt. 14 –* ***shadow map workshop***	**Program 2 due**
Week 5	2/7	Global illumination – AO, Physically based BRDFs, etc.	
	Read	*Chpt. 7 & 8 & 18*	
	2/9	Global illumination – alternatives	
	Read	*Chpt. 7 & 8 & 10*	**Individual Code R**
Week 6	2/14	**Midterm**	
	Read	*Your class notes*	**Individual Code R**

	2/16	Geometry in games: Polygon, voxels, parametric	
		Chpt. 12 & 13 in RTR	**50% Final project**
Week 7	2/21	**NO CLASS – secret MONDAY**	
	2/23	Polygon, voxels, parametric continued	
		Chpt. 12 & 13 in RTR	**Individual Code R**
Week 8	2/28	Geometry: culling	
	Read	*Chpt. 14 in RTR –* ***view frustum culling workshop***	**Program 3 due**
	3/2	Animation again – skinned meshes	
	Read	*Link (Bridson)*	**Individual Code R**
Week 9	3/7	Effects in games – particle systems, billboarding , transform feedback	
	Read	*Link (Baraff and Witkin, etc.)*	**75% Final project**
	3/9	Geometry shaders and subdivision surfaces	
	Read	*Chpt. 14 from RTR and link*	**Individual Code R**
Week 10	3/14	Visual story telling	
		McCloud handout	**Individual Code R**
	3/16	**Midterm**	
		90% Final project	**play testing sessio**
Final	3/23 (thur)	**1:10-4pm**	**100% Final projec**