

# CPSC 4050/6050 Graphical Systems Syllabus

**Instructor:** Dr. Andrew Duchowski  
**Office:** McAdams 309, 656-7677, *duchowski [at] clemson.edu*  
**Office hours:** TBD (or by appointment)  
**Email:** Communication with the instructor via email is encouraged. It is unlikely that a talk request will be answered, however.

**Required texts:** 1. None.

**Recommended texts:**

1. Angel, Edward, *Interactive Computer Graphics: A Top-Down Approach with OpenGL, 3<sup>rd</sup> Ed.*, Pearson Education, 2003. ISBN: 0-201-77343-0
2. *OpenGL Architecture Review Board; Dave Shreiner; Mason Woo, Jackie Neider; Tom Davis, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2, 5<sup>th</sup> Ed.*, Addison-Wesley, 2006. ISBN: 0-321-33573-2
3. *Randi J. Rost, OpenGL Shading Language, Addison-Wesley, 2004. ISBN: 0-321-19789-5*
4. *Foley, James D., van Dam, Andries, Feiner, Steven K., and Hughes, John F., Computer Graphics: Principles and Practice, 2nd ed.*, Addison-Wesley, 1990. ISBN: 0-201-84840-6

**Supplemental texts:** None.

**Grading scheme:** Based on programming assignments, final project, 1 or 2 midterms, and the final exam.  
 600-level students will be required to analyze and report on technical papers in the area.

**Grade distribution:**

		%	Grade
Projects	60%	90-100	A
Midterm(s)	20%	80-89	B
Final Exam	20%	70-79	C
		60-69	D
		< 60	F

**Programming assignments:** Problem specification and due date will be given in class.

**Assignment grading:** Source code and demonstrations will be required.

**Assignment format:** Electronic assignment must include the following:

- Makefile.
- Documentation of the program (e.g., user documentation in a README file).

**Assignment late policy:** Late assignments will be accepted but points will be deducted according to the formula  $(3n)^3$  where  $n$  is the number of days late. Example: assuming assignments are due on Tuesday, the point deduction is as follows:

--	--	--

Max points possible	Day received	Days late
100	Tuesday	0 (due date)
73	Wednesday	1 day late
0	Thursday	2 days late

Late assignments will receive lowest priority for grading and returning.

**Attendance:** Roll will be taken for the first one or two weeks while the class roll fluctuates. However, attendance is not required. Absence, excused or not, does not change the responsibility for assigned work. Tests missed due to *excused* absences will normally result in the test not being counted in the average grade (i.e., there will normally be no makeup tests). An *unexcused* absence from a test will normally result in a grade of zero for that test. Students are expected to give at least one week advance notice for excused absences.

**Independent work:** Unless otherwise stated explicitly (e.g., in the case of the final project), each student must do his or her work independently.

**Academic dishonesty:** The University policies on academic dishonesty apply. Publicly-available code or other material may be freely used if appropriately attributed. Each student is responsible for protecting his or her files from access by others. Work that is essentially the same and submitted without proper attribution is considered to be a violation of academic dishonesty policy by all those submitting the work, regardless of who actually did the work.

**Class cancelation:** Students are expected to wait for 15 minutes after the class beginning time before leaving if the instructor is late.

**Course description:** The course will cover computer graphics methods, data structures, analysis of algorithms, and selected implementation examples, generally coinciding with the main programming assignments assigned throughout the term:

- Project I: Image viewer
  - Objectives:
    - GUI
    - image file description (PPM)
    - displaying images
      - via `glDrawPixels`
      - via texture map
      - via GLSL
    - color conversions: RGB, grey, CIE XYZ
- Project II: Object viewer
  - Objectives:
    - GUI
    - polygonal object description files (Alias|Wavefront)
    - displaying objects
    - 3D transformations (rotations, translations, etc.) incl. camera pan/tilt/roll/truck, object rotations (quaternions)
    - face/vertex normal visualization
- Project III: Object renderer
  - Objectives:
    - OpenGL lighting & shading (flat (faceted)/smooth)
    - selection
    - hidden surface removal (based on normal and viewpoint)
    - Phong lighting/shading

- Project IV: Object shadower
  - Objectives:
    - texture mapping (image file formats)
    - shadow mapping
    - shadow volumes
  
- Project V: Animation (particle systems; time permitting)
  - Objectives:
    - particle systems
    - scene/animation description files