

CS 3258 Spring 2021

COMPUTER GRAPHICS SYLLABUS & POLICY HANDOUT

Bobby Bodenheimer
bobby.bodenheimer@vuse.vanderbilt.edu

January 25, 2021

1 Course Description

This course is about *computer graphics*. In this course, computer graphics means the procedural or programmatic use of a computer to create or manipulate images. We'll first talk about what images are, then we'll learn how to manipulate them, but the majority of the course will be spent on generating images.

The course requirements consist of five programming projects. There are no tests. The programming projects are quite intensive, and so a working knowledge of C++ (or C) is required.

1.1 Prerequisites

CS 3251; Math 2400, 2410, 2501, or 2600 (linear algebra).

1.2 Class Time and Attendance

This class will be taught online, synchronously on T-Th from 11:10-12:25. Students are expected to be in virtual attendance during these times.

1.3 CS Course Status

This course is a depth course for purposes of the CS major.

2 Textbook

The textbook for this course is *Fundamentals of Computer Graphics*, 4th ed., Steve Marschner and Peter Shirley, CRC Press, ISBN 9781482229394. This is an excellent book on the modern fundamentals of graphics. However, the material we cover in this course is taught from a more signal-processing approach to graphics than the book. As a result, we jump around in the book quite a bit, and often material is covered in class that is somewhat different from the book. Thus, often there won't be material in the book discussing the particulars of the class lecture or the projects. I will point out what sections of the book are relevant when they occur.

3 What You Will Learn

3.1 Computer Graphics Topics

The major topics of this course:

Image procession	Anti-aliasing	Modeling
Meshes	2D and 3D transformations	Shading
Clipping	Texture-Mapping	Ray-Tracing
Color	Visual Perception	Graphics Hardware (GPUs)
Image Forensics		

3.2 Computer Science Topics & Outcomes

From the general perspective of computer science these are the topics we will cover in this course:

- (1) Techniques, skills, and tools necessary for computing practice.
- (2) Principles and practices for secure computing.

In terms of computer science learning outcomes, what you should get out of this course are:

- (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

4 Grading

Grades are computed on a point basis. Each project is given a maximum number of points. The score you receive on the project is recorded and the total summed over the quarter. Typically, each assignment will be worth approximately 50 points. Grades will either be evaluated as the best of a strict percentage basis (90+% → A, 80-90% → B, etc.) or by determining the modes of the class grade distribution and assigning letter grades to each mode.

Assignments will be handed out in class, and are also available on Oak.

5 Assignments Due

The due dates for assignments are strictly enforced. The assignments are due at midnight on the deadline day, usually two or three weeks (depending on the assignment) after the assignment is given.

5.1 Late Policy

The late policy works as follows: You have five (5) “floating” days that you can use to turn in assignments late. Weekends, holidays, power outages, and other natural disasters count in these floating days. So, for example, let's say that a project is due by midnight on Thursday, February 11, 2021. If you turn it in at 12:01 a.m. on Friday you've used up one of your floating days. If you turn it in at 11:59 p.m. on the same Friday, you've still only used up one of your floating days. If you turn it in at 12:01 a.m. on Saturday, you've used up two floating days.

Projects can be turned in late as long as you have an appropriate number of floating days remaining for the time amount indicated. The exception is the last project, which must be turned in by noon on December 12th whether you have multiple floating days remaining or not. If you use up all your floating days, then projects must be turned in on time. If they're turned in late, they will have 50% automatically deducted from them, OR they won't be counted at all, completely at the discretion of myself and the TA.

USE YOUR FLOATING DAYS WISELY! It's usually not a good idea to use floating days because you didn't allocate enough time for the project, i.e., you started the project Sunday night and it was due on Tuesday. The projects take time to code – I'm giving you plenty of time to do them. Floating days are best used when you have 2 assignments due and a test on the same day, and you need an extra day to do some final checks and modifications.

If truly terrible unforeseen circumstances arise you might be able to arrange an extension with me. Giving an extension is completely dependent on what my workload is like at the time — don't expect one and don't take it for granted.

6 Assignments

We will be *running* your assignments in order to test them. You will be turning in files for each assignment comprising your source code and your test data. Details on how to turn in the assignments will be provided later.

Each assignment should contain everything needed to compile and test your program. **DO NOT** make links to files located in other places. Keep these directories around and the files in them until after your assignment has been graded. You don't need to copy over executables as your program will be remade before testing. Please provide either a Makefile or a shell script which will build your executable from scratch.

All of your projects require you to provide test data of some sort. Part of your project's grade will be based on how thorough your test data is. The test files should be named something completely obvious like `test1.dat`, `test2.dat`, etc. You should describe the test data and how to run your project in the README.md file of your github distribution. You will **lose** 5 points if you don't provide any test data.

Note that any program which crashes while being tested *automatically* loses 5 points. Thus, you should pay special attention in checking for strange input or handling internal errors before they propagate. Also, note that your program should pretty much work as advertised. If it doesn't, we won't be spending too much time trying to find out why when we're grading it.

7 Collaboration, Acceptable Practice, and the Honor System

I encourage collaboration as long as you're learning something. But, when it gets to the point that you are collaborating to save work at the expense of learning, stop.

This means that, in the early stages of thinking about an assignment, it's good to discuss the project with your fellow students and to learn about various library functions and algorithms together. You can discuss algorithms and data structures in a general way.

As work progresses on an assignment you should collaborate less and less. For example, you aren't allowed to COPY source code or specific data structures. You can certainly look at someone else's source code. But if you have it in front of you while you're typing into the computer, or if you just edit a copy of their file, then you're stepping over the line. It's easy to see when a program has been copied—the variable names may have been changed but they have exactly the same structure and do exactly the same things; the control structures all go exactly the same way, the program succeeds and fails in exactly the same way, etc.

To summarize, you can work together at a high level but I insist that each one of you write your own programs. In particular, note that you are bound by all provisions of the Honor Code, found in the Vanderbilt Student Handbook and available online at http://www.vanderbilt.edu/student_handbook/the-honor-system/. Other pertinent material may be found at the Honor Council's web site, <https://studentorg.vanderbilt.edu/honorcouncil/>.

If you have any doubts about whether you are stepping over the line, it is best to ask me for clarification — not another student or the T.A. Uncertainty about the Honor Code or its application does not excuse a violation.

8 Office Hours and Recitation

Prof. Bodenheimer is easily available to meet. Visit <https://bobbybodenheimer.youcanbook.me/> to schedule a meeting. If none of those times work for you, send email.

Additionally, there will be a recitation session I will usually conduct for this class at a time to be determined. If there's someone there and questions are asked, I'll answer them until there are no more questions (or the hour gets REALLY late). If there's no one there with 5 to 10 minutes, I'll leave. My intent with these sessions is to answer questions about the assignments in more detail than I may have time to go into them during class, clear up confusing points, and generally answer a question once to everyone rather than ten times to ten different people.

As mentioned earlier, this course requires a large amount of programming. Be prepared.

9 Teaching Assistant

The TA for this course is Haley Adams, haley.a.adams@vanderbilt.edu. Office hours TDB.

10 Brightspace/Slack/Github

We will use Brightspace to post the syllabus and other information. We will use the messaging application Slack for informal online discussions outside of the classroom. Invitations to join the channel will be sent after Add/Drop. Slack will NOT be used however for formal course information (assignments, etc), but rather Github and Brightspace.

11 Important Dates

Jan. 26	First Class, Assignment 1 out
Feb. 11	Assignment 1 DUE; Assignment 2 out.
Feb. 23	In-class reading day.
Mar. 4	Assignment 2 due; Assignment 3 out.
Mar. 18	Assignment 3 DUE.; Assignment 4 out
Apr. 6	Assignment 4 DUE; Assignment 5 out.
Apr. 8	In-class reading day.
Apr. 29	Last Class
May 4	Assignment 5 DUE

12 Excellence by All

It is my belief that everyone in this class can excel and that our collective learning experience is improved by including everyone fully. Toward that end, I commit to doing my best to use inclusive language and practices. If you observe actions in which I am not respectful or inclusive, please bring those to my attention.

13 Disabilities

Vanderbilt is committed to equal opportunity for students with disabilities, as am I. If you need course accommodations due to a disability, please contact VU Student Access Services (<https://www.vanderbilt.edu/student-access/disability/>) to initiate that process.