

Course Missive

Spring 2018

Introduction

Welcome to CS1950U a.k.a. Topics in 3D Game Engine Development!

You may also want to refer to the calendar, and the list of assignments, all of which are available at the course website (cs.brown.edu/courses/cs195u).

Announcements made during the semester will be sent to the class e-mail list, as well as posted on the home page of the course website.

In this course you will learn techniques needed to create 3D game engines, including component-based design, spatial subdivision, world representation, collision detection and response, game networking, and graphics. Near the end of the semester, you will work in a group to design and build a 3D game that may incorporate features from the engines you've developed.

In order to ensure that every student has a positive experience in this course and gets the attention and assistance they need from the course staff, enrollment has been capped at 25 students. If enrollment reaches this limit and additional students still wish to take the course, the staff will decide which students will be admitted, with preference given to seniors.

Course Staff

| Professor | Office | Email (@cs.brown.edu) |
|---------------|---------|-----------------------|
| Barbara Meier | CIT 401 | bjm |

| Head TA | Login |
|-----------|--------|
| Ben Attal | battal |

| TAs | Login |
|--------------|----------|
| Ben Sheeran | bsheeran |
| Jordan White | jaw5 |
| Jeff Hao | jmhao |

Prerequisites

There are two formal prerequisites for this course. The first is some form of software engineering. Any of CS123, CS1971, CS32, or CS33 will satisfy this requirement. The second prerequisite is graphics, which may be satisfied with CS123. If you have not satisfied one or both of these requirements, you are still be eligible to take the course, but you must contact the TAs or the instructor for an override code. Our goal is for anyone who wants to take this course can take the course.

Classes

The class will meet on Wednesdays from 3-5:20pm in CIT 506. If the time or location changes, we will make an announcement at the preceding class and send a message to the class e-mail list.

Approximately the first 30-90 minutes of this large time block will be used for a lecture on that week's game engine topics. On weeks besides the first checkpoint for each assignment, class will then move down to the Sunlab (after the lecture portion), where the remainder of the class period will be used to playtest the assignment that is due that week.

Playtesting is required in order for your project to be considered complete, so class attendance will effectively be mandatory, and missed playtesting must be made up. If you will be unable to attend class on a particular day, playtesting must be completed **before the next assignment is due**.

Projects and Grading

The course consists of four programming projects: a 2-week warm-up project, 2 regular projects, and a student-directed final project. Each project is divided into two or more weekly checkpoints. There are no homeworks or exams. Projects/checkpoints will be assigned each Wednesday after the lecture, and will be due the following Tuesday at midnight (11:59:59 pm). You will hand in assignments by running the `cs195u_handin` script from your project's root directory (typically where the Qt `.pro` file resides) on a CS department computer.

There are three sets of project-specific requirements for each checkpoint: playtesting, primary, and secondary requirements. Playtesting and secondary requirements are worth one point each, and primary requirements are worth two points. Thus, each checkpoint offers a total of four points. Additionally, there are a set of global requirements that must be fulfilled for every handin.

Before the final project begins, there are three mini-checkpoints (to hand in your idea, your group, and your design). Each mini-checkpoint is worth one point.

If a handin does not receive points for the primary or secondary requirements, you have one week to retry the assignment and submit another handin that meets the checkpoints requirements, **but only if you attended your design check!** If and when you retry, the handin will be regraded. If it meets the primary requirements, you will receive full credit for them. If it meets the secondary requirements, you will receive full credit for them. You cannot lose credit for using a retry. If you hand in a project late, you can still receive full credit for the primary and secondary requirements. Playtesting requirements and mini-checkpoints cannot be retried.

Furthermore, you have two extra retries, each of which allows you to retry a previous retry. This means if your retry handin still does not satisfy the primary or secondary requirements, you may use an extra retry to fix it and submit it again for grading. You may also hand in your project two weeks late and state that you want to use your standard and extra retry at the same time.

Note, however, that these projects are cumulative. The engine features that you implement in one week will generally also be necessary in order to complete the next weeks project. Since each project depends on the previous one, you will still need to complete them in order, and once you are late on one project you run the risk of staying behind schedule on every subsequent project. Do not let this happen! This grading system can be a little confusing at first, so don't hesitate to email the TAs if you have any questions.

All checkpoints have the same weight. Your letter grade in the course is determined by the number of points you have by the end of the semester.

| Points | Missing | Grade |
|--------|---------|-------|
| 46-54 | 0-8 | A |
| 35-45 | 9-19 | B |
| 34- | 20+ | C |

However, **regardless of the number of completes or no-credits, you must have handed in a working version of all checkpoints (minimally, the primary requirements for each) by the end of the semester to pass the class.**

Incompletes in the class (grades of INC) will not be given except in extenuating circumstances authorized by a dean or a note from Health Services. If you know in advance that you will be requesting an incomplete grade, please talk to the Head TA as soon as possible.

If you have a problem with the grade you received for an assignment, you should first talk to the TA who graded that assignment. If you are still unhappy, you can contact the Head TA; if the Head TA is unable to resolve the problem, contact the professor.

Final Project

The last project in this course will be an open-ended final project, for which you are allowed and encouraged to work in a group with other students. Unlike the other programming projects, you will determine the requirements for this assignment. You will be able to pick from a list of engine features to implement, and you will write your own game requirements based on the kind of game you want to create.

You should start coming up with ideas for your final project, at the latest, by the time you are halfway through Platformer (the last predefined project). An idea for your project will be due at the same time as the final handin for Platformer, and a more detailed design proposal, including your list of requirements, will be due after that. If you want to work in a group, this will also be the time to form one; some class time will be dedicated to helping people finalize project groups the week before the design proposal is due. Although it is possible to do the final project on your own, everyone is encouraged to form at least a two-person group so that you will have the resources to make a more interesting game.

Design Checks

There will be a mandatory design check for every checkpoint, held on the Thursdays, Fridays and Saturdays preceding its due date in CIT 207. Each design check will last 20 minutes. You will sign up for a design check using `cs195u.signup <project>` in the terminal.

Design checks do not represent a percentage of your project grade. Instead, they must be completed to receive a standard retry for the checkpoint. If you do not complete the design check for a specific checkpoint, you will have to use an extra retry to retry that checkpoint.

The design checks in this course are informal. To receive credit for a design check, you need to answer a few questions regarding the assignments content. It is important that you have thought

about the project, understand the concepts that it depends on, and have a plan for solving the major problems that it involves. Questions for each design check will be posted with the assignments. The TAs will expect you to answer these questions at a conceptual level. While it is not necessary to have prepared a design diagram or have any code written, feel free to bring either to your design checks to help answer the questions.

Playtesting

Playtesting is an important part of this class. In addition to creating a game engine, you are also creating playable video games, and you will want to make sure that other people can play them successfully. Even in the checkpoint weeks when you are not required to have a finished game, your project should still be a functional demonstration.

Furthermore, since projects do not receive numeric grades, playtesting will be an important opportunity to gauge how well your project works and detect non-obvious bugs before they become a problem. At playtesting you are encouraged to try to “break” the games you playtest, and report any bugs or odd behavior you find.

As mentioned previously, playtesting will occur during class time after lecture on final weeks of projects. Students will be randomly split into roughly equal-sized groups, and each student will playtest every other student’s game in the group. You will need to fill out a playtesting form for each game you playtest, so that the person you’re playtesting has a written record of your feedback. The feedback form is anonymous.

Collaboration Policy

In order to make sure that each student in CS1950U is graded as fairly and individually as possible, the course staff have written a collaboration policy by which we expect all students to abide. **Please read this policy carefully**, as it may differ from collaboration policies in other CS classes you have taken. The policy isn’t too long, and we have tried to make it easy to read.

CS1950U involves some challenging software design problems for which there is often no single “right” solution. Thus, it is helpful and encouraged to discuss the projects with your peers and help each other find more creative solutions to these problems. However, the work you hand in should be entirely your own and represent your own understanding of the concepts taught in this class. For these reasons, the collaboration policy generally allows you to talk about the projects and your code’s high-level design with other CS1950U students, but not to share code or help other students with debugging.

As with other CS classes that prohibit sharing code between students, you are responsible for ensuring that the permissions on your source code directories do not allow other students to view them. Ask a consultant or a TA for help with permissions if you are unsure of how to use them.

TA Hours

TA hours will be held most days of the week in CIT 207. Once the TAs work out their own class schedules, the exact hours will be posted on the course web page. You can go to TA hours to ask questions about the concepts and algorithms presented in class, get advice on the design of your engine, and ask for help in solving particularly difficult bugs.

TAs are here to help you, but remember, TAs are students too. Please don't ask them questions outside of official TA hours. This includes talking to them in person or electronically while they are at home or in the lab.

If you need to contact the TAs outside of TA hours, e-mail the alias `cs1950utas@lists.cs.brown.edu`. You should generally use this alias instead of sending e-mail to TAs individually, as most questions can be answered by any TA and you are more likely to get a timely response by e-mailing the alias.

If TA hours are rescheduled or canceled for any reason, there will be an announcement to the class e-mail list. If you feel you can't possibly make the scheduled TA hours (especially after a reschedule), get in touch with the head TA.