CS1230 Syllabus

Introduction to Computer Graphics, Fall 2017
http://cs.brown.edu/courses/cs123
cs1230tas@lists.brown.edu

1 Welcome

Welcome to CS1230, the longest running computer graphics course in the known universe! This short document will get you started with the course. For specifics, please also read through the CS1230 Student Guide (https://github.com/cs123tas/docs/blob/master/student_guide.pdf).

2 Staff / Info

- Professor: Andries van Dam (avd@cs.brown.edu)
- Head TA: Ruiqi Mao (rm28@cs.brown.edu)
- UTAs: Sacha Servan-Schreiber (aservans), Mason Bartle (mbartle), Emma Catlin (ecatlin), Loudon Cohen (lcohen2), Jordan White (jaw5), Shannon Ward (srw), Katya Schwiegershausen (kschwieg), Liam Callanan (lcallana)
- Lectures: Tuesday/Thursday 10:30-11:50 AM, CIT 368
- Labs: TBD in the SunLab (CIT 143)

3 Prerequisites

The official prerequisites for CS123 are CS15/16, CS17/18, or CS19. CS33, because it uses C, is very helpful but not strictly required. CS32, for its software design, is additionally helpful but not required. Some familiarity with C++ will be helpful, but help sessions and extra support will be offered to students who don't have any prior C++ experience. Some knowledge of basic linear algebra (i.e., vector and matrix multiplication, dot and cross products) may also be helpful, but none is required or assumed.

4 Topics

The full list of lecture topics can be found on the lectures page of the website (http://cs.brown.edu/courses/cs123/lectures.html). These topics include:

- **OpenGL**: Industry standard graphics library used to produce real-time 2D and 3D graphics.
- **3D geometry**: Different ways 3D shapes can be represented in graphics applications.
- **Image processing**: How to process images by applying filters and transformations while minimizing artifacts like aliasing (a.k.a. “jaggies”).
- **Viewing 3D scenes**: How to create a virtual camera that converts a 3D scene to a 2D image using linear algebra.

- **Illumination**: Mathematical models that can be used to calculate the amount of light reflected from an object in a scene.
- **Ray tracing**: A rendering method that is used to produce pseudo-realistic images with reflections, refraction, and other effects.
- **Color theory**: Various ways to think about and represent colors.
- **Stochastic rendering methods**: Various strategies for rendering 3D scenes that use probabilistic models to simulate the physics of light transport.
- **User interfaces**: Theory behind creating user interfaces for different form factors.
- **Virtual and Augmented Reality:** Creating fully immersive computer-generated experiences.

5 Course-Related Work Expectations

Over 14 weeks, students will spend 3 hours per week in class (42 hours total), and approximately 2 hours per week in labs (20 hours total). Homeworks, which consists of 1-2 week projects and the accompaning algorithm assignments, usually take 10-15 hours per week, though they can also require somewhat more or less time depending on how much extra credit a student chooses to implement. In any case, students will spend a minimum of 8.5 hours per week and 120 hours total on homework by the end of the semester.

6 Assignment Structure

Your CS1230 grade consists of 7 rigorously graded projects, 11 labs that are given completion grades (1 out of the 4 last labs is optional), and a final project that is graded according to your presentation to the class during finals period. There are no exams or quizzes, and final grades are not curved - if the work meets specification, it deserves an A and that is the most common grade.

- **Brush** (Due 9/17, weight 6%)
- **Shapes** (Due 10/1, weight 9%)
- **Filter** (Due 10/15, weight 13%)
- **Sceneview** (Due 10/29, weight 8%)
- **Intersect** (Due 11/12, weight 13%)
- **Ray** (Due 11/21, weight 11%)
- **Final** (Presentation date TBA , weight 20%)
- **10 out of 11 Labs** (Due Sunday before next lab, total weight 10%)

Each of the 7 programming projects has an accompanying “algo” assignment which is due the Thursday after the assignment is released. These are worth 10% of the total grade.

7 Materials

The primary text for this course is the most recent edition of *Computer Graphics: Principles and Practice, by John Hughes, Andy van Dam, Morgan McGuire, David Sklar, Jim Foley, Steve Feiner, and Kurt Akeley*. The book is currently available at Amazon.com.

The book is not required, as the lecture slides for CS1230 are nearly comprehensive. These slides do not replace the textbook, but they are the best source of information that is directly relevant to the assignments.

The course website has a wealth of information that will be useful to you throughout the semester:

- Handouts that will get you started on each project
- (http://cs.brown.edu/courses/cs123/projects.html)
- Handouts that will get you started on each lab
- (http://cs.brown.edu/courses/cs123/labs.html)
- Links to PDF and PowerPoint versions of each lecture as well as accompanying videos and/or demos
- (http://cs.brown.edu/courses/cs123/lectures.html)
- Documentation for the support code, help session slides, and links to external sources for help with OpenGL (http://cs.brown.edu/courses/cs123/docs.html)
- TA hours schedule and contact information (http://cs.brown.edu/courses/cs123/staff.html)

8 Collaboration Policy

CS1230 has a strictly enforced collaboration policy (https://github.com/cs123tas/docs/blob/master/collaboration.pdf) in line with Brown's Academic Code which emphasizes that all written work must be the student's own except for certain limited forms of collaboration that are explicitly permitted. Our collaboration policy in particular prohibits collaboration on the design and implementation of programs.

Please read the collaboration policy carefully, sign it, and give it to a TA after lecture as soon as possible.

9 Half-Credit Course

Students can also register for CSCI-1234 and complete specific additonal assignments for each project to earn an extra half-credit. This course can be used to get grad credit, and it can also count as a capstone course. More details can be found in the student half-credit guide (https://github.com/cs123tas/docs/blob/master/student_half-credit_missive.pdf) found in the docs section of the course website.

10 Late Policy

A late algo or lab checkoff will receive no credit. A project turned in up to 3 days late will be penalized by 15%, with an additional 10% penalty for every day late after 3. Every student has two late passes that can be used to waive the 15% deduction on a project. These late passes will automatically be applied optimally (to maximize your grade) at the end of the semester.

In exceptional circumstances, a doctor's or dean's note may allow for an additional extension or late pass. These will be handled on a case by case basis.

Lastly, all projects must be handed in with at least minimal functionality by the end of the semester in order to receive your grade.