

CS 134 Innovating Game Development: Course Missive

January 22, 2008

1 General

Meetings

Time	MWF 11-12 (D hour)
Location	Lubrano Conference Room CIT 4th Floor

Course Staff

		Login	Office Hours
Instructor	Chad Jenkins	cjenkins	10-11 MW
Head TA	Korhan Bircan	kbircan	by appointment
UTAs	Scott Daniel	sadaniel	by appointment
	Neehar Reddy	neehar	by appointment

Staff email

HTA	cs134hta@cs.brown.edu
TAs	cs134tas.cs.brown.edu

Course email

Mailing list	cs134@list.cs.brown.edu
WWW	http://www.cs.brown.edu

2 Course overview

What technologies will shape the next generation of video games? Innovating Game Development is a project-oriented **games technology** course focusing on computational innovations for game development. Students examine innovative game technology through case studies of existing games and talks by industrial and academic game professionals. In teams, students propose and implement a project demonstrating a novel technology for gaming. Recommended: strong computational or engineering background.

2.1 Games technology: We will not be designing games or making game engines!

Game engines are huge engineering efforts that have yielded hundreds of thousands lines of code and consumed years of effort by experienced developers. However, many of the foundational ideas and technologies in games (e.g., 3D rendering, physical simulation, pathplanning, network protocols, databases) come from more serious research efforts from decades past. Instead of “reinventing in the wheel”, this course is about future directions for games by integrating cutting-edge topics in computing with modern game development. The central product of this course is the innovation of a single component or technology in an existing or created game. Projects will be expected to heavily utilize and improve existing third-party libraries and solutions. By leveraging external resources, effort and interest is directed to enhancing existing and/or creating new components applicable to game development as a whole.

2.2 What is games technology?

Simply, it is scientific research applied to games. We categorize efforts towards making games into roughly three categories:

- Game design: the process of designing the content, background and rules of a game or class of games (i.e., using an existing engine to create game content).
- Game development: the process by which a specific game or class of games is produced (i.e., creating specific engines for games given current technologies) .
- Games technology: the process of developing methodologies and techniques applicable to classes of current and future games (i.e., foundational computational, engineering, and scientific concepts for making games and other types of applications).

The future of games technology depends research to explore new possibilities and is defined as follows¹:

Research is a systematic investigation, including research development, testing and evaluation, designed to develop or contribute to **generalizable knowledge**.

The key aspect is the creation of generalizable knowledge applicable beyond a specific project. This aspect separates research from development in that your ideas should be applicable to entire classes of games and serve as a foundation for new types of games and innovations. The impact of good research is often inspiring others to adopt your ideas, use them for new purposes, and make them better.

3 Prerequisites: Is this class right for you?

CS 134 is a “capstone course” for upper-level undergrads (particularly in computer science) near the completion of the concentration and seeking experience in the area of interactive systems. Graduate students can take CS 134 to apply their research or find possible research topics in interactive entertainment. Upper-level engineering or digital media students seeking interdisciplinary projects are encouraged to explore the course to find common interests.

Although CS 32 or CS 36 are the explicit prerequisites, we assume every student has a solid background in a relevant area of expertise. If you do not meet the prerequisites, please consult the instructor before enrolling in CS 134.

¹This definition is based on terminology from human subjects research

4 Course Structure and Grading

Students enrolled in CS 134 are expected to:

- present and compose 2 **case studies** of innovative games or games research projects
- implement 2 **introductory assignments** for *level design* and *incorporating external libraries*
- complete 6 “**onepager**” brainstorming exercises for game ideas
- **propose and prototype** a course project for innovative games technology
- **publically present and report on** the completed project
- **class participation** through proactive discussion and questions

Your overall grade will be based on grading of each assignment divided roughly proportional to the following weights:

10	Case studies (2 case studies, written and presented in class)
10	Onepager brainstorming exercises (6 onepagers, 2 due on Feb 1, 8, and 15)
10	Assignment 1: Level Design (due Feb 8)
5	Assignment 2: External Libraries (due Feb 15)
15	Course Project: Proposal and Prototype (first proposal due March 1) (approved proposal due March 21)
15	Course Project: Intermediate Prototype and Game Trailer (due April 18)
15	Course Project: Final Implementation (due May 6) (Arcade Day demos TBD, likely May 9)
10	Course Project: Postmortem Report (due May 12)
10	Class Participation

We will assess a 10% penalty per day for late submissions.

4.1 Electronic Submission

Submission of your work will occur electronically. Written assignments (case studies, onepagers, proposals, postmortem) will be handed in using Brown’s MyCourses system and must be in PDF format. Implementations will be submitted using space provided by the course staff in the CS department file system. All implementation submissions must be immediately executable by departmental systems for Linux or Windows or include a simple installer. Special cases will only be made if explicit permission is requested in a timely manner and granted by the instructor or HTA.

4.2 Collaboration Policy

Students are highly encouraged to collaborate together and utilize external resources during this course. Any code or information provided by an external source or another student must be **explicitly cited**. Uncited usage of information that is not your own will be considered plagiarism. For each introductory assignment, case study, and onepager, **no sharing of code or text** is allowed between students, but usage of publically available resources is allowed. For course projects and case studies, you are encouraged to use any information or resources.

5 Case Studies

A case study is an in-depth analysis of a game or research project that emphasizes an innovation in gaming. The case study assignments are to be completed individually and consist of providing a written analysis (through completing the case study document) and a 10-minute in class presentation. A potential format for case studies can be found in the course directory.

The topic of the case study is open to any topic related to game innovation, but must be approved by the instructor. It is the student's individual responsibility to obtain this approval and arrange for slots in the class schedule to present their case study analysis.

Potential Case study topics:

- Network protocols (first person shooter)
- Network protocols (MMORPG)
- Distributed data management
- Network security
- Scripting and programming languages
- AI and search
- Group behavior
- Decision processes
- Character animation and motion capture
- Physical simulation and character control
- Graphics processing units
- Embodied gaming (incl. robotic and sensor network systems)
- Geometric modeling and deformation
- Computer vision and sensor processing
- Emerging sensor technologies
- Game controllers
- Console and platform hardware architectures
- Artisitic and asset development tools
- Affective computing
- Machine learning for gaming and game development
- Social networks
- Large scale software development
- Nonphoto realistic rendering
- Immersive displays
- Immersive sound
- High-resolution displays
- Location-based entertainment
- Web-based applications
- Mobile gaming
- User input paradigms (such as one button games)
- Randomness
- Balanced gameplay
- Educational games and edutainment
- Sound and mixing
- Asset management
- Camera control
- Level of detail
- Level mods

6 Onepager exercises

A onepager is a quick exploration of a potential game idea. One pagers are to be completed individually and outline the high-level aspects of a potential game. A possible format for onepagers can be found in the

course directory.

7 Introductory Projects

To make sure you have sufficient familiarity with game development tools, you must complete two introductory assignments. The first assignment, **Level Design**, requires the implementation of a single level for a game that you design. You can implement this level using the engine of your choice. However, the course staff will only support Irrlicht, Ogre, and XNA-based tools. The second assignment, **External Libraries**, requires you to build on your initial level by incorporating *at least* two additional external libraries to add new functionality for networking, physics, AI, sound, etc.

These assignments will be implemented with the Dojo/G3D game engine, for which the staff will provide support code.

8 Course Project

The goal of cs196-2 is for you to produce an innovation for interactive games. The course project will be the primary medium for this innovation. The course project involves proposing, developing, and implementing a novel technology for gaming. Being on the cutting edge can lead to many unforeseen difficulties. To facilitate progress, the course staff is expecting the following procedure to be executed during the completing of your project:

- **Project proposal** is a thorough specification and rough prototype of your project that builds upon the ideas in a project worksheet. The proposal must be approved (or “greenlit”) by the course staff by Spring Break. Described in the proposal specification (in the course directory), your proposal must contain a detailed software engineering plan, a timeline for achieving milestones, a rubric for evaluating your progress, implemented prototype with storyboard, and a statement of support from a project mentor. The proposal prototype must be in a distributable package, such as a wizard-based installer or self-contained executable.
- **Project webpage** (updated weekly): once greenlit, every project group is expected to maintain a webpage of relevant material about their project. This webpage should contain all documents related to the project, progress reports, playable demos, screenshots, and other materials.
- **Project weeklies**: starting Feb 22nd, everyone is expected to present “weeklies” every Friday. A weekly is short (2-5 minute) presentation highlighting relevant progress made during the week through demos, videos, screenshots, and/or oral communication. If your proposal has not been submitted, you are expected to present a game idea from one of your onepagers. A related note: each group is encouraged expected to maintain a CVS (Concurrent Versions System) repository of your source and assets. At any point in time during the project, the contents of the CVS repository should yield a working demo when checked out.
- **Intermediate Prototype and Game Trailer**: to demonstrate progress after Spring Break. This graded milestone will expect the unfinalized components of your game to be in place to form a working game. This prototype must be used to produce a 90-180 second video trailer for your game.
- **Final Implementation**: is the final product of the project. This will be submitted before Arcade Day demonstrations to (optionally) create DVDs, cover art/packaging, posters, and final webpages for the game projects.
- **Arcade Day demonstrations**: all course projects will be demonstrated over a 2-hour period at the end of the semester. You are expected to have a poster describing your game and its innovations. The course staff will visit each group’s demonstration area to hear the presentation of your poster and ask questions. The Arcade Day will be open to the public. You are expected to attend for the entire demonstration period.

- **Project postmortem:** the postmortem is a review of a completed game by the developer. Refer to Game Developer Magazine’s Postmortem format.

Project feedback from the course staff is incredibly important, especially during the iterations for crafting your worksheets and proposals. Once you submit a case study, onepager, or proposal, the course staff will make our best attempts provide feedback within 36 hours. Brown’s MyCourses system will be used to facilitate submission and feedback.

9 Textbooks and Readings

Required:

- **M. McGuire, O.C. Jenkins, “Creating Games: Mechanics, Content, and Technology”, working draft, AK Peters Publishing**

Recommended: Interfaces

- Bowman, D., Kruijff, E., LaViola, J., and Poupyrev, I. “3D User Interfaces: Theory and Practice”, Addison-Wesley, ISBN 0-201-75867-9, July 2004.

Recommended: Programming

- **E. Lengyel, Mathematics for 3D Game Programming and Computer Graphics**
- Haines and Moller, Real-Time Rendering
- Bourg, Physics for Game Developers
- Eberly, 3D Game Engine Design
- Stroustrup, The C++ Programming Language Read, How to be a Programmer (<http://samizdat.mines.edu/howto/How>)

Recommended: Art

- **Walker and Walker, Game Modeling Using Low Polygon Techniques**
- Steed, Animating Real-Time Game Characters
- Steed, Modeling a Character in 3DS Max
- Derakhshani, Introducing Maya 6: 3D for Beginners

Recommended: Audio

- Brandon, Audio for Games
- Marks, The Complete Guide to Game Audio

Recommended: Management

- **Game Developer Magazine Post-Mortem series (available on Gamasutra.com)**
- Brooks, The Mythical Man-Month
- McConnell, Rapid Development
- Maguire, Debugging the Development Process

- Spolsky, Joel on Software

Recommended: Game Design

- **Koster, A Theory of Fun for Game Design**
- Rollings and Morris, Game Architecture and Game Design
- Salen and Zimmerman, Rules of Play: Game Design Fundamentals
- Fullerton, Swain, and Hoffman: Game Design Workshop: Designing, Prototyping, and Playtesting Games