# cs196-2 Innovating Game Development: Course Missive

May 14, 2006

## 1 MOTD

Group Websites CASE STUDY FORM!!!!1!

Dojo manual: Manual Read this: Controllers Ryan Houlette's lecture slides Amy Greenwald's lecture slides Al Reed's lecture slides Al Reed's Technical Design Document WebCT : here Wiki : here Advances in Computer Entertainment website Graphics Meets Games website

We're giving you webspace! go to /course/cs196-2/www/groups/ and make a group directory. Then you can put your webstuff there. Indie Games! Mind Rover Cosmic Blobs Yanif's Home Page Yanif's Lecture Slides Misha's Lecture Slides Brad Myers' Lecture Slides Harmonix' Lecture Slides

## 2 Assignments

Assignment 0 Assignment 1 Assignment 2

## 3 Helpsession Info

GtkRadiant Dojo

## 4 Taglines

Research for gaming. Innovating gaming without emulating games.

### 5 General

All TA hours will now be by appointment. newsgroup for updates.

Meetings	MWF 11-12		
	Lubrano Conference Room	CIT 4th Floor	
Course Staff		Login	Office Hours
Instructor	Chad Jenkins	$_{ m cjenkins}$	10-11 MW
Head TA	Pawel Wrotek	pwrotek	by appointment
UTAs	Alex Rice	acrice	by appointment
	Graham Rosser	grosser	by appointment
Contributors	Morgan McGuire	morgan	
	Joe LaViola	jjl	by appointment
Staff email:			
HTA	cs196-2hta (at) cs (dot) brown (dot) edu		
TAs	cs196-2tas (at) cs (dot) brown (dot) edu		

### 6 Course overview

Innovating Game Development is a **games research** course being offered in the Spring 2006 semester. We will be discussing current trends in game development, focusing mainly on new and innovative technologies that will likely shape the game industry in the coming years. Students will work alone or in groups to design and develop a fully functional video game based around some novel technological paradigm of their choice. First, students will have to create a project proposal and "sell" their idea to the course staff. Once their idea is approved, student teams will combine their inspiration and creativity, technical knowledge, and software engineering to prepare their project for "release" (demonstrations during Arcade Day). The course will satisfy the design requirement for the CS Sc.B. concentration.

Course Missive

#### 6.1 Games research: We will not be making game engines!

Game engines (and related engines for rendering, physics, AI, etc.) are huge engineering efforts that have yielded hundreds of thousands lines of code and consumed years of effort by experienced developers. It would be unwise to attempt such software engineering effort to reinvent such engines in the space of one semester.

Instead, we focus on games research. Specifically, improving the technology underlying interactive entertainment systems. We will emphasize focused innovation of a single aspect or component of game technology. 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.

#### 6.2 What is games research?

It is scientific research applied to games, silly. 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.
- Game development: the process by which a specific game is produced.
- Games research : the process of developing methologies and techniques applicable to classes of current and future games.

Our definition of games research is based on the following definition of research<sup>1</sup>:

<sup>&</sup>lt;sup>1</sup>This definition is based on terminology from human subjects research

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. A good games research project will yield responses like "You can do that in a game?!".

### 7 Prerequites: Is this class right for you?

More exactly, is games research right for you? Research is about the cutting-edge and advancing the stateof-the-art, which can be both exhilarating and frustrating. Similar to game development, games research prerequires focused direction, technical competency, and sincere motiviation. To gauge whether your direction is suited for this course, please answer these three simple questions (latex template). You may also want to ask yourself if the games industry is right for you (refer to Pausch's Academic's Guide to Electronic Arts).

For technical competency, there are no explicit prerequisites for cs196-2. You are expected to utilize your own technical background to develop and implemented a compelling project. Although there is no specific skill set, we assume a solid background in CS (or engineering) or a really good idea is absolutely critical for successful completion of the class.

Our introductory assignments requires competency in C++ programming. C++ will not be taught as part of this class. We highly recommend that you have or gain competency in C++ programming, but such skills are NOT required for this class. Courses in graphics and AI, such as cs123 and cs141, are also NOT required, but are certainly a plus.

## 8 Course Structure and Grading

Students enrolled in cs196-2 are expected to:

- present and compose 2 case studies of innovative games or games research projects
- design and implement an innovative **course project** to improve an existing game
- situation pending, implement 4 introductory projects
- class participation through proactive discussion and questions

Each of these tasks are described in the following subsections.

#### 8.1 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 unforseen difficulties. To facilitate progress, the course staff is expecting the following procedure to be executed during the completing of your project:

Thus, we ask for a proposal that covers many of the software engineering aspects of the project while discussing how the technology will lead to compelling games.

• **Project worksheets** (due every Friday, pdf, latex): you should complete a 2 page "idea" worksheet for every potentially interesting game concept. At least one new worksheet must be handed in every week until your project proposal is approved. Multiple worksheets can be handed in simultaneously to received staff feedback; however, you should be careful to put thought into your submissions. Once a worksheet is approved, you can submit a project proposal based on the approved idea.

- **Project proposal** (due 3/24, pdf, latex): the project proposal is thorough specification of your project that builds upon the ideas in a project worksheet. The proposal is where you "sell" your idea to the course staff for "greenlight" approval. Your proposal should contain storyboard and conceptual materials for describing the project, a detailed software engineering plan, a timeline for achieving milestones, a rubric for evaluating your progress, and a statement of support from a project mentor. The final proposal is due by 3/24.
- **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** (presented weekly): once greenlit, a project group will 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. A related note: each group will be 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.
- Arcade Day (TBD, likely Friday 5/12): all course projects will be demonstrated over a 3-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** (TBD, likely duedate 5/15): 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 worksheet or proposal, the course staff will provide feedback within 36 hours. We are currently looking at submission and feedback procedures to facilitate this process.

Please be thorough in crafting the project proposal. The course staff will be very strict in evaluating the proposals and achievement of stated milestones.

Note: Once your project proposal has a greenlight, you will not be obligated to complete any introductory assignments. Once approved, the project team will focus on the final project and meeting milestones set forth in the project proposal. It is in everyone's best interest to have project ideas approved as soon as possible.

#### 8.2 Introductory Projects

To help students towards the course project, there will be four introductory programming assignments. The introductory assignments involve the implementation of selected components for a broad set of video games:

- **Project 0: Level design.** You will gain familarity with a level editor (such as GtkRadiant) to create a level for Dojo. Because Dojo incorporates physics (via the Open Dynamics Engine), we envision a marble navigation game, such as Marble Blast by Garage Games. You will load your level into Dojo and deal with issues such as character control, basic game interaction, triggers, etc...
- **Project 1: Path planning.** This assignment involves creation of a search mechanism for nonplayer character enemies. To accomplish this you will write a 3D pathfinding algorithm for non-player characters and integrate it with your game.
- **Project 2: Character animation.** You will incorporate a fully animated, motion-captured humanoid character into your game. You will implement the motion graph data structure to store the character's motions, and come up with a scheme for controlling the character in-game.
- **Project 3: User Input.** This project involves the integration of a creative user input sensor modality, hardware technology, or interface paradigm.

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

#### 8.3 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 30-minute in class presentation. 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
```

## 9 Grading

The grading structure for the course will vary individually among the class based on the respective progress of each student. Each grading criteria has been given a weight. Your overall grade will be based on grading of each assignment divided by the sum of their overall weight. The various aspects of the class will be weighted as follows:

- 5 Project 0
- 10 Projects 1-3
- 5 a case study
- 50 course project
- 5 class participation

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

### 9.1 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 should be **explicitly cited**. Uncited usage of information that is not your own will be considered plagerism. For each introductory assignment, **no** sharing of code 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.

## 10 Textbooks and Readings

Required: 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

## 11 Calendar/Syllabus

The course staff has been fortunate to assemble a good combination of speakers from industry, researchers, and student presentations. We will begin with some basic coverage of game components, 3D graphics, physics, and level design to gain common familiarity. After these fundamentals, the course will roughly follow a pattern of lectures on Monday, guest speakers on Wednesdays, and demo presentations on Fridays.

The format of this class is open-ended, so the topics may change due to flux in student interest or spawning of interesting threads.

Calendar key – (L: lecture, G: guest lecture, O: project out, I: project due)

Week of	Mon	Wed	Fri
1/23	no class	L: Introduction	L: Physics
	L: Camera interfaces		Half-Life 2
	Antigrav		Dismount
			On The Run
1/30	L: What is a Player (AI)?	L: 3D Graphics and NPR	Input/Output Modalities (jjl)
	Sims' Evolving Creatures (paper)	Doom 3	I: Project worksheet
	Virtual Fighter 4 (paper)	Jet Set Radio Future	
	O: Level Design	Sly Cooper	
2/6	G: Ryan Houlette	G: Amy Greenwald	Student weeklies
	Simbionic		I: Level design
			I: Project worksheet
2/13	G: Yanif Ahmad	G: Harmonix	Student weeklies
	Distributed data management		I: Project worksheet
	O: Path planning		
2/20	Open	G: Demiurge	Student weeklies
	Bioneurofeedback gaming		I: Path planning
			I: Project worksheet
2/27	Mijail Serruya: Bioneurofeedback gaming	Kent Quirk	Student weeklies
	O: Character animation		I: Project worksheet
3/6	Interaction Paradigms (jjl)	G: Brad Myers	Student weeklies
		Galactic Village	I: Character animation
			1: project worksheet
3/13	Max McGuire	G: Mad Doc	Student weeklies
	Iron Lore		I: project worksheet
2 / 2 2	O: User interfaces		
3/20	Big Huge Games	Olga Karpenko	Student weeklies
			I: User interfaces
2/07			I: Final project proposal
3/27	spring break	spring break	spring break
4/3	open	open	Student weeklies
4/10	open	Brandon Curiel	Student weeklies
4/17	open	open	Student weeklies
4/24	open	Kob Lindeman	Student weeklies
5/1	open	open	Student weeklies
5/8	open	open	Arcade/Final Demonstrations
5/15	1: Project postmortem	no class	no class

## 12 Potential Projects

A continually growing list of project ideas:

- Robotic First Person Shooters and Real-time Strategy games
- Hopper Football: physics-based soccer with biped hoppers (Raibert and Hodgins 1991)
- Game development curriculum: developing tools to promote education through game development at the high-school level
- Control of physics-based humanoids
- Exercise-based video games: game interfaces that encourage physical activity
- Grand theft cycling (kipp): bicycle interfaces for video games
- Behavior cloning/capture: learning decision processes from human demonstration Laird's Human Level AI slides
- Nintendo DS Hacking: utilizing the wireless functionality to creating distributed mobile gaming. (do the google search yourself :)
- Artificial life: methods for evolving game characters through genetic algorithms and machine learning
- CIT Doom (jjl)
- •

## 13 Available Technologies and Game Engines

Dojo is the only game engine that will be officially supported by the course staff. Dojo is currently in development and utilizes G3D for rendering and ODE for physics:

- G3D
- ODE

Other game and specialized engines that could be used include:

- Ogre
- Torque
- Blender
- Genesis3D
- Delta3D
- SimBionic
- fmod
- Unreal Engine
- Maya
- MotionBuilder

### 14 Influential Games List

In any field, professionals work within a mainstream culture that references important previous work. These form the critical jargon and jumping off point for new ideas.

Research is important in any field. It is how we build on the successes of the past and avoid their failures. You would not try to write a book or create a car without first learning about the ones that preceded yours. When creating a game, you should research previous games. This list summarizes some of the most important games. It is intended as a jumping off point for further research if a game sounds like one you Rd like to make. Read through it to familiarize yourself with the previous work.

For brevity, we focus on a limited number of critically aclaimed (or derided) and popular games are listed. In many cases, a previous game introduced a concept but (e.g. Crystal Caverns predated Wolfenstein but with less impact). These are the games that designers often list as their major influences.

More information can be found at MobyGames and The Video Game Museum. Categorized List:

- Physics and Simulation: Half-Life 2, Gish, Grand Turismo 4, Trespasser
- Freedom of action: Deus Ex, GTAIII, Shenmu
- AI: Wasteland (1987)
- Audio: Loom (1982), Medal of Honor, Jade Empire, God of War
- Animation: Prince of Persia (1990), God of War (2005)
- Network/Co-operation: Ultima Online, Halo, Doom, Battlefield 1942
- Level Design Innovation: Shadow of the Collossus, Gish, Katamari Damacy, Guitar Hero, Paper Mario, Feel the Magic: XY/XX (2005), Full Spectrum Warrior (2004), Perimeter (2004)
- Gameplay (Interaction of ludemes): Counter-Strike, Civilization, Age of Empires, MULE, Starcraft, Advance Wars, Grand Theft Auto III, Diablo (Diablo II), SSX Tricky, Dragon Quest VIII, WoW
- Gameplay (Clever): Incredible Machine, Defenders of the Ancients (DotA), Mario Kart
- Gameplay (Well honed ludemes): Tetris, Pac-Man, Burnout Revenge
- Graphics: Ico (Shadow of the Collossus), JetGrind Radio (Cel Damage), Myst, Jade Empire, Gears of War, Far Cry, Project Gotham Racing 3, Curious George, Doom III, Thievius Racoonus
- User Input: Guitar Hero, Karaoke Revolution, AntiGrav, DDR, SuperMario 64, Doom/Quake, Marble Madness, Nintendogs, Grand Theft Auto III, Atari Football (1978), Nintendo Power Glove, Duck Hunt (1985), Nintendo NES Robot, Dance Dance Revolution, Mocap Boxing
- Story/Scripting: Wing Commander III (1994), Half-Life, Alone in the Dark (1992), Out of This World, System Shock (1994), Metal Gear Solid (series), Indigo Prophecy

#### 14.1 Morgan's List of Most Influential Games

- Doom: Possibly the most significant video game ever made. Introduced or popularized several ideas core to modern gaming: first person perspective for a non-vehicle game, over the gun view, reload animations, shareware distribution/playable demo, mouse-look, cooperative network multiplayer, horroraction, dynamic lighting, mod-able and reusable engine, and multiple bonuses for replay value.
- Tetris
- Stuntz Broderbund: Not only one of the first true-3d (polygonal) race car sims, but one that incoroporated physics for leaving the ground and a complete editor for making your own tracks out of pre-defined roller-coaster like pieces.

- Grand Theft Auto 3 (GTA3) Halo: Return of co-op multiplayer (absent since doom). Use of co-op in vehicles.
- Battlefield 1942 (and sequels)
- Counter-Strike
- $\bullet\,$  Half-Life
- Ultima Online: Everquest, Asheron's Call, Dark Age of Camelot, City of Heros, World of Warcraft
- There, Second Life
- Poker
- Asteroids
- Mario 64
- Chess
- Checkers
- Pente
- Go
- $\bullet~{\rm Connect}{\text{-}4}$
- Advance Wars
- Pac-Man
- Myst (Myst II, III, Riven sequels, RealMyst)
- Age of Empires
- Civilization
- Zelda (and sequels)

### 14.2 Chad's Additions

- EA Madden 92
- EA NHL 94
- Space Invaders (1978)
- Pong
- Super Mario Brothers
- Street Fighter
- Daytona USA
- Virtua Fighter
- Donkey Kong

## 15 Related External Links

- Brown Games Technology Research Group
- Craig Reynolds' Game Research and Technology Page
- Indie Game Jam
- Experimental Gameplay Workshop
- Game Developer Magazine
- Game Developer's Conference
- Gamasutra
- I3D Symposium
- AAAI Symposia on AI and Interactive Entertainment
- AI and Interactive Digital Entertainment Conference
- Journal of Game Development
- Boston Postmortem
- Framesticks Artificial Life
- Game AI
- Academic's Guide to Electronic Arts
- Experimental Gameplay Project
- G4TV Icons Series
- Kile: in integrated environment for LaTeX (but really vi is good enough)