Syllabus and Class Overview
CSCE 4210/5250
Ian Parberry

Class Structure
• There will be 9 lectures (2 hours) & 7 labs (1 hour).
• The rest of the scheduled lecture and lab times will be open development time. Use it wisely.
• Lectures will consist of PowerPoint presentations. Notes will be online. Attendance is optional.
• Labs will consist of a tutorial to be completed during the session. Attendance is mandatory.

About Me
• My name is Ian Parberry.
  • If you wish to be formal, call me “Dr. Parberry”.
  • Contact: ian@unt.edu.
  • For more information: http://larc.unt.edu/ian.

Office Hours

My Credentials
Why you should listen to what I have to say.

Who Am I?
• A professor (since 1984).
• Pioneer of undergraduate game programming education since 1993.
• Credits on 4 commercial games.
• Author of 4 books on game development.
• Author of 36 academic articles on game development research.
1997: SubHunt (Spectrum Pacific)

Programmer.

1997: G-Nome (7th Level)

“Special thanks to...”

2001: TrapWords (MindGames)

Programmer.

2013: TUG (Nerd Kingdom)

Director of Computer Sciences.

Game Programming Books


Research
Alumni of This Class

- 67 alumni in the game industry since 1994
- 3 alumni authors of game development books
- 3 alumni started game companies
- 3 alumni professors of game development
- 245 credits on 143 commercial games
- 180 million copies sold
- $9B in revenue

Distinguished Alumnus Jason West

Jason West, formerly Infinity Ward (Activision), Respawn Entertainment (Electronic Arts), 1996.

Distinguished Alumnus Art Griffith


Distinguished Alumnus Cesar Stastny

Cesar Stastny, Director of Technology, Treyarch (Activision), 2005.

Distinguished Alumnus Gloria Kennickel

Gloria Kennickel (2006), id Software. Renderer programmer for Rage, voted Best Overall Game and Best Shooter of E3 2010.

Do You Belong In This Class?

This is a tough class. Are you up to it?
Learning Outcomes

1. Be familiar with Windows programming.
2. Be able to use Visual C++ 2013.
3. Be able to use the Microsoft Direct3D 11.2 SDK.
4. Be able to program a 3D billboard game.
5. Be able to work in a team with other programmers using Subversion.
6. Be able to code one or more aspects of a game, including graphics, sound, and gameplay.

Prerequisites

- The listed prerequisite for this class is CSCE 2100: Computing Foundations I.
- You will be forcibly dropped from this class if you do not have the official prerequisite.
- You should be familiar with the basic data structures and formalisms used in computer science.

Expectations

You need to be or become:
- a strong C++ programmer.
- familiar with object-oriented design.
- comfortable with the following C++ concepts:
  - Derived classes
  - Virtual functions
  - Function and operator overloading
  - Memory allocation with new and delete
  - Default parameters
  - The keywords const, static, public, private, and protected
  - The C++ standard library

I expect you to:
- be excited and energized by the opportunity to learn new things.
- take responsibility for your education.
- be competitive.
- exceed expectations.

Requirements

These are things you must do.
Musts

- You must work in a group of 3 ± 1 programmers.
- You must use the class revision control system.
- Your game must run on the computers in the lab.
- You must pass all of the checkpoints.

What Resources Can You Use?

- Almost anything goes.
- Don’t recycle a complete game engine, but you may borrow and integrate code from any legal source.
- You will be graded on your contribution to the code.
- Be honest – attribute your work.
- Using code without acknowledging it to the instructor is cheating, and will be dealt with in accordance to the departmental cheating policy.

The Revision Control System

- You must use the Subversion server provided (more details later).
- I will be monitoring your progress during the semester.
- You must observe Best Practices when using the RCS (more later).
- I will be showing group progress in class towards the end of the semester.

Checkpoints

- The labs (required).
- The pitch (required).
- The milestone (required).
- The game contest (optional).
- The final presentation (required).

The Labs

- There will be 7 lab sessions, each of which will have you complete a tutorial.
- Completion of a tutorial during the lab session in which it is assigned will earn you 1 point.
- Completion of a tutorial during the following lab session will earn you 1/2 point.
- You will lose a letter grade for every 2 points that you miss (I will round in your favor).
- You will gain a letter grade if you get 7 points.

Where Will the Labs Be?

- NTRP F204
- Open during posted lab sessions plus more hours during the week (details will be posted).
- 24 Intel Core i7-3930K CPU @ 3.2GHz with 32GB of RAM (hex-core, hyperthreaded)
- Dual monitors, DirectX 11.2 capable video cards
- 64-bit Windows 8.1
- Visual Studio 2013
The Pitch

- Week 4, 5.
- Your group will give a PowerPoint presentation to the class trying to sell them on your game.
- This is called the pitch.
- It is Pass/Fail for the group.

The Milestone

- Week 9, 10
- Your group will demonstrate your game to the class.
- This is called the milestone.
- It is Pass/Fail for the group.

The Contest

- In Week 15 there will be a class game contest.
- Participation is optional.
- I will invite judges from the game industry.

The Final Presentation

- Finals Week: December 7-10, 2015.
- Each group will get 30 minutes to show their game and answer my questions.
- All programmers must attend.
- A schedule will be drawn up towards the end of the semester.

Grades

How will your grade be determined?

My Grading Philosophy

- I grade the executable, not the source code.
- How do I grade a game?
  A: it really knocks my socks off
  B: it’s a pretty cool game
  C: it’s an OK game
  D: it’s not there, but at least you tried
  F: you really blew it off, didn’t you?
- Yes, this is not well-defined. Welcome to the real world. The game industry doesn’t know how either.
Things I Look For in a Game

1. Completeness points: Is it finished?
2. Techno points: Does it contain things not covered in class?

- Undergraduate students enrolled in CSCE 4210: I expect either completeness or techno points.
- Graduate students enrolled in CSCE 5250: I expect both completeness and techno points.

Completeness points

- Does it run?
- Are there bugs?
- Is it actually a game?
- Is it fun?
- Does it engage the player?
- How far does it go beyond Ned’s Turkey Farm?
- Is there sound?
- Did you use Subversion properly?
- Were you on time for the milestone?

Techno Points

Awarded for things not covered in class. Examples:
- Better graphics (lighting, bump mapping).
- A level editor.
- A particle engine.
- Procedural content generation.

Your Grade

Your individual grade in this class will depend on:
1. The quality of your group’s game.
2. Your contribution to its executable code.
3. Your correct usage of Subversion.
4. Your group’s performance on the pitch and milestone.
5. Your answers to questions during the final presentation.

Instructional Methodology

What makes this class different.

Incremental Development

- This class will be taught by showing the development of a simple game incrementally, i.e. step by step.
- A simple side-scroller: Ned’s Turkey Farm
- The idea is not to teach you about a single game engine, but about game engine development using Ned’s Turkey Farm as an example.
- The labs will show you how to build onto this code.
Demos for Ned 2014a

- Demo 0: The Black Screen of Death
- Demo 1: Direct3D
- Demo 2: Scripting and Debugging
- Demo 3: Sprites
- Demo 4: Animated Sprites and Scrolling Background
- Demo 5: Artificial Intelligence
- Demo 6: Scripted Level Design
- Demo 7: Sound
- Demo 8: Playability (Sprite Sheets, Text, and HUD)
- Demo 9: Pixel Shaders
- Demo 10: Two-player networking
Why Teach This Way?

• Once you know how to program, the best way of learning is by reading and doing.
• I will not be teaching you everything there is to know about DirectX 11.2.
• You are capable of reading the documentation and the sample code online.
• It is, however, large and complex.
• Getting started is the hardest thing.
• I show you how to get started in small, easy steps.

Ned Design Philosophy

• KISS: Keep It Simple, Stupid.
• Ned is an educational tool, not an all-purpose game engine.
• Teaching code ≠ production code.
• When introducing a concept, I try to keep it as simple as possible, but no simpler.
• This means that Ned is pretty bare-bones in some places.

Schedule

Put this stuff on your calendar. Now.
Lectures

1. Syllabus, Group Formation
2. Game Design
3. Subversion, Visual Studio 2013, Demo 0
4. 3D Math, DirectX 11.2, Demo 1
5. Demo 2, Demo 3
6. Demo 4, Demo 5
7. Demo 6, Demo 7
8. Demo 8, Demo 9
9. Demo 10

Labs

1. Game Design
2. Demo 0: Windows programming
3. Demo 1: Top down games
4. Demo 3: Sprites
5. Demo 5: AI
6. Demo 7: Sound
7. Demo 9: Fullscreen pixel shaders

Section 003, 004

Lectures: Monday
Lab: Monday (003), Tuesday (004)

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug. 18</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Aug. 25</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Sept. 14</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Sept. 21</td>
<td>Pitches</td>
<td>Dev</td>
</tr>
<tr>
<td>Sept. 26</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Oct. 5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Oct. 12</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Oct. 10</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Oct. 15</td>
<td>Milestone</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 2</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Nov. 9</td>
<td>9</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 16</td>
<td>-</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 23</td>
<td>-</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 30</td>
<td>Contest</td>
<td>Dev</td>
</tr>
</tbody>
</table>

Section 001, 002

Lectures: Wednesday
Lab: Wednesday (001), Thursday (002)

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug. 20</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Sept. 2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Sept. 9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Sept. 16</td>
<td>Pitches</td>
<td>Dev</td>
</tr>
<tr>
<td>Sept. 23</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Sept. 30</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Oct. 7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Oct. 14</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Oct. 21</td>
<td>Milestone</td>
<td>Dev</td>
</tr>
<tr>
<td>Oct. 28</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Nov. 4</td>
<td>9</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 11</td>
<td>-</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 18</td>
<td>-</td>
<td>Dev</td>
</tr>
<tr>
<td>Nov. 25</td>
<td>-</td>
<td>Dev</td>
</tr>
<tr>
<td>Dec. 2</td>
<td>Contest</td>
<td>Dev</td>
</tr>
</tbody>
</table>

Cell Phone Policy

- Please mute your cellphones.
- However, having a cell phone ring in class is not the end of the world.
- Please put them away during my lectures.
- Similarly for computers, unless you are working on your game.
Cheating Policy

- The Department of Computer Science & Engineering cheating policy will be adhered to.
- Any student caught cheating will receive a grade of F for this course, and further disciplinary action will be taken.

---

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING CHEATING POLICY

Cheating:
- Turning in someone else's work as your own. This includes, but is not limited to, another classmate, another textbook, a student in another class, or a student in a prior semester.
- Allowing someone to turn in your work as their own.
- Several people writing one program and turning in multiple copies, all represented (implicitly or explicitly) as individual work.
- Stealing an examination or solution.
- Using unauthorized material during a test or quiz.
- Changing a test, program, or other student work after the work has been graded and requesting that the work be regraded.

Not Cheating:
- Turning in work alone or with the help of the course's staff.
- Submission of one assignment for a group of students if group work is explicitly permitted (or required).
- Getting or giving help on any university or department operating system.
- High-level discussion of course material for better understanding. Discussion of assignments for clarification.

---

DEPARTMENT OF COMPUTER SCIENCE and ENGINEERING CHEATING POLICY

- When cheating is deemed to have occurred, appropriate disciplinary action will be taken.
- A notice will be placed in the student's permanent computer science record outlining the behavior and the subsequent disciplinary action.
- The instructor may impose a penalty of failure in the course and may deny the student permission to drop the course.
- It is also possible that the student may be barred from subsequent registration in any computer science and engineering courses at the University of North Texas.
- The matter may be referred to the appropriate dean for further university action.

The student is directed to the University of North Texas Student Handbook for general university regulations on cheating under the section on categories of misconduct.
- All procedures concerning cheating are subject to the student's right to due process as outlined in the Student Handbook.
ADA

The University of North Texas complies with Section 504 of the 1973 Rehabilitation Act and with the Americans with Disabilities Act of 1990. The University of North Texas provides academic adjustments and auxiliary aids to individuals with disabilities, as defined under the law. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring accommodation, please see the instructor and/or contact the Office of Disability Accommodation at 940-565-4323 during the first week of class.

My Advice To You

Act like a grownup. Really.

Advice

• Get real.
• Don’t blow it off. Start early, and finish it before the end of the semester.
• I am not your mother. Don’t expect me to nag you.
• You are responsible for getting it done on time.
• Bring group problems to me as soon as possible.
• Don’t make me fail you.
• If you can’t hack it, get out of the class.

More Advice

• Read your UNT email for messages from me.
• Stay in contact with your group.
  • Exchange email addresses and phone numbers.
  • Answer them.
  • Do not drop off the face of the Earth.
• Meet regularly outside of class times.
• Look at past games at http://larc.unt.edu/demos/

What Has Changed This Year?

In case you were wondering.

Code Updates

Ned 2014 has been updated to Ned 2014a.
• This includes code and lecture notes.
• But this happens quite often anyway.
We’re Growing

This class had more students sign up than can be accommodated in one room.

• There are legal limits on the number of students who can fit into a classroom.
• Large classrooms in Discovery Park are at a premium.
• Administration are “encouraging” us not to turn away students.
• We therefore opened a second section.

Active Learning

After experience with active learning in CSCE 4220 this spring, we decided to change to a 2 hour lecture and 1 hour lab per week in this class.

Confucious

I hear, and I forget;  
I see, and I remember;  
I do, and I understand.

More Better Computers

• The lab used to have 12 current-generation and 10 previous-generation computers.
• The Dean of the College of Engineering and the Chair of the Department of Computer Science & Engineering kindly provided funds for 12 new computers in the lab, bringing us up to a total of 24. (Upgrades to these computers will come from student fees.)
• That means we can have a maximum of 24 students in each lab session.
• To maximize your convenience, we have scheduled 4 lab sessions, MTWT 7:30-8:20pm.
Some Common Laments

I do listen to you.

Lecture Timing

“The material in the later lectures comes too late in the semester to be really useful.”

• Old solution: Make it clear that students are expected to read ahead in the lecture notes rather than wait for me to spoon feed them.
• New solution: Make sure the core material is delivered quickly in 7 lectures.

Code

“Reading code in class is a waste of my time.”

• The best way of learning code is to look over the shoulder of someone who can code better than you.
• If you are bored, you are probably not paying enough attention.
• This class has students at varying levels of ability.
• You may be beyond the need to read my code, but have some respect for those with lesser abilities than you.

Homework

“I want more homework to help me learn faster and motivate me to get started on the project sooner.”

“I want less homework because it distracts me from the main project.”

• Old solution: 1-3 homeworks.
• New solution: No homeworks, 7 required labs.

Windows and DirectX

“I want to use Unix and OpenGL”

• You can learn graphics using Unix, gcc, and OpenGL in CSCE 4230: Intro to Computer Graphics and CSCE 5220: Computer Graphics.
• This is where you learn programming for Windows using Visual C++ and DirectX.
• We will be using DirectX 11.2. Windows 8.1, Visual Studio 2013, and fairly new graphics hardware is required for DirectX 11.2.
• The software is free from MSDNAA.
• If you can’t afford the hardware, use the lab.

PowerPoints

“I hate PowerPoint Professors.”

• I wrote most of the code and all of the PowerPoints for this class – I didn’t just Google them.
• The PowerPoints have enough detail for them to serve as lecture notes, which is good because there really is no textbook that covers all of this material.
• I suffer from Multiple Sclerosis.
  ▪ PowerPoints are an accommodation that I need to be able to perform my job.
  ▪ I am legally entitled to this accommodation by the ADA.
In Conclusion

Summary

- This is a group project class using Subversion, Windows 8.1, Visual Studio 2013, and DirectX 11.2.
- There is a game development lab in NTRP F204.
- There will be tutorials, a milestone, a game contest, and final presentations.
- I’ll be teaching by dissecting a collection of game demos for Ned’s Turkey Farm.
- I have high expectations.

One does not simply walk into Mordor CSCE 4210/5250