Lecture 1

Course Overview
CS/INFO 4152: Advanced Topics

• Sequel to CS/INFO 3152
  • Prereq unless a non-Cornell Meng (or exempt)
  • Similar format and structure as Intro Game Design
  • Covers topics not touched in Intro Game Design

• Single semester long game project
  • At least 55% of your final grade
  • Interdisciplinary teams of 4-6 people

• Also design documents, but no labs
CS/INFO 4152: Advanced Topics

- Uses familiar the **milestone** schedule
  - Deliverables every two weeks (after week 3)
  - One extra prototype beyond 3152 schedule
  - Details on course website:
    http://www.cs.cornell.edu/courses/cs4152

- Games demonstrated at **Showcase**
  - Like BOOM, open to the public
  - Public reaction is part of your grade
  - Submissions posted on the GDIAC website
Course Structure

• Most of the course happens during lecture section
  • Meets three days a week (M,W,F) 11:15-12:05
  • Mixture of lectures, presentation, and discussions
  • Course is a bit more interactive than CS/INFO 3152

• Lectures: Common in first half of course
  • Advanced game development topics unique to course
    (this is not going to replace a graphics course)

• Design Focus: mechanics, user interfaces and testing

• Technical Focus: mobile platforms, memory management
Course Structure

• **Presentations**: Every two weeks
  • In-class critique of your game by your peers
  • Part of your participation grade comes from this
  • Because of class size, held over three sessions

• **Playtesting**: Follows every single deliverable
  • Handled just as in the introductory class
  • Will expect user-test scripts for alpha and onward

• **Critiques**: Ungraded, less formal presentations
  • **Example**: The pitch session next week
The Discussion Sections

- Discussion time was biggest request a few years ago
  - Like communication lab from CS/INFO 3152
  - Time to work on Assignments already assigned

- We have contacted your team about times
  - Groups 1-6 meet Monday 12:20-1:10 in Upson 202
  - Groups 7-9 meet Wednesday 2:30-3:20 in Hollister 312

- **Catch**: You must enroll in ENGRC 4152
  - Extra credit hour for work you are already doing
  - This is *required*; it is not optional
Game Requirements

- Should be **mobile game** on iOS or Android
  - Develop cross-platform, but graded only on one
  - Exceptions for 3D must have 5625 alums on team

- Some form of **innovative gameplay**
  - Interface innovation for mobile
  - 3D game should leverage camera control

- Target **public distribution**
  - Mobile apps should try to get on either App Store
Mobile Game Development

• Will use custom **C++ game engine: CUGL**
  • Built on top of SDL (Simple DirectMedia Layer)
  • Made to solve many problems from last year

• We do **not** provide any hardware
  • New devices are about $200; used are cheaper
  • Just need one device for your whole group

• Either 2D or 3D is acceptable
  • Will need **OpenGL ES** in either case
Choosing a Platform

- You **must** develop iOS apps on a Macintosh
  - Only XCode can load the app on to a device
  - No longer need Apple Developer membership
  - But need membership ($100) if want multiplayer

- You can develop Android on **either platform**
  - Android Studio finally works this year
  - But it is not good enough for your main IDE
  - You should target Mac/Windows for testing
Working in C++

• Best option for cross-platform development
  • iOS: Obj-C and C++; Android: Java and C++
  • Game developers should learn it anyway
  • Will have several lectures if it is new to you

• You should use a professional IDE
  • This means XCode or Visual Studio
  • Tools for analyzing memory performance
  • Android Studio is not a professional C++ IDE
Cornell University Game Library

- New game engine “written from scratch”
  - Core set of 90 C++ classes
  - Supports input, graphics, and audio

- Layered on top of some useful libraries
  - **SDL**: SimpleDirectMedia Layer
  - **Box2D**: The definitive 2D physics library

- Compatible with any C++ library out there
  - **Example**: Bullet for 3D physics
Working With CUGL: Good News

- Supports modern(ish) C++
  - Full C++11 support
  - Heavy use of smart pointers
- Build is very light-weight
  - Engine has 200 MB footprint
- Advanced input features
  - Built-in pinch and rotation
  - User-recordable gestures
  - Arbitrary text input
- Modern OpenGL support
  - OpenGLES 3.1 on mobile
Working With CUQL: The Bad News

- Engine is very spartan
  - Box2D is only 3d-party library
  - No support for external editors
  - No support for rigging

- Android sound is not great
  - Clipping if interrupt sounds
  - 100ms playback delay
  - MEng is working on it

- Windows is *debugging only*
  - No UWP development
  - This means no Surface support
**Intellectual Property**

- Your **group** retains all ownership
  - You can commercialize it later
  - You can make derivative works
  - Individual ownership is your responsibility

- But Cornell gets a non-exclusive license
  - Non-commercial use of final version submitted
  - We can post this version on our website
  - We claim no other rights to your game
Semester Schedule

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Form Groups</th>
<th>1/27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 2</td>
<td>Team Workflow</td>
<td>2/3</td>
</tr>
<tr>
<td>Week 3</td>
<td>Concept Document (Project Kickoff)</td>
<td>2/10</td>
</tr>
<tr>
<td>Week 4</td>
<td>Nondigital Prototype Milestone Proposals</td>
<td>2/14 2/17</td>
</tr>
</tbody>
</table>

February Break

<table>
<thead>
<tr>
<th>Week 5</th>
<th>Gameplay Specification</th>
<th>2/24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 6</td>
<td>Gameplay Prototype</td>
<td>2/26</td>
</tr>
<tr>
<td>Week 7</td>
<td>Architecture Specification</td>
<td>3/10</td>
</tr>
<tr>
<td>Week 8</td>
<td>Technical Prototype</td>
<td>3/12</td>
</tr>
<tr>
<td>Week 9</td>
<td>Document Revisions</td>
<td>3/24</td>
</tr>
</tbody>
</table>

Pre-Production

Development
## Semester Schedule

<table>
<thead>
<tr>
<th>Week 10</th>
<th>Alpha Release</th>
<th>3/26</th>
</tr>
</thead>
</table>

**Spring Break**

| Week 10 | Code Walkthroughs Level Design | 4/9  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>4/14</td>
</tr>
<tr>
<td>Week 11</td>
<td><strong>Closed Beta Release</strong> (Feature Complete)</td>
<td>4/16</td>
</tr>
<tr>
<td>Week 12</td>
<td>App Store Proposal</td>
<td>4/28</td>
</tr>
<tr>
<td>Week 13</td>
<td><strong>Open Beta Release</strong> (Open Playtesting)</td>
<td>4/30</td>
</tr>
<tr>
<td>Week 14</td>
<td>Final Portfolio Postmortems</td>
<td>5/7</td>
</tr>
<tr>
<td>Week 15</td>
<td>GDIAC Showcase</td>
<td>5/18</td>
</tr>
</tbody>
</table>
Group Management

• Every group has a **project leader**
  • Final say in all *design decisions*
  • Coordinates designers and programmers
  • Responsible for milestone reports

• Every group has a **lead programmer**
  • Responsible for the *code architecture*
  • Responsible for maintaining code base
  • Delegates coding tasks to others
Help Outside of Class

• Must meet as a group for 1/hour a week
  • Pick a regular time and place
  • Submit as part of your team workflow

• Will serve as a form of “office hours”
  • The instructor will come if invited
  • Use for “just-in-time” instruction
  • Algorithms/techniques unique to your group

• Will also be using Piazza this semester
Grading Policy

- Mixture of group and individual grades

- Group grades are same for all group members
  - Group Game Grade (25%)
  - Course Documents (25%)
  - Class presentations (5%)

- Individual grades distinguish group members
  - Individual Game Grade (30%)
  - Participation and Reports (15%)
Game Grade

• Group grade reflects the game quality

<table>
<thead>
<tr>
<th>Grade</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Bug-free, Fun-to-play</td>
</tr>
<tr>
<td>B</td>
<td>Complete and playable</td>
</tr>
<tr>
<td>C</td>
<td>Complete but unplayable</td>
</tr>
<tr>
<td>D/F</td>
<td>Serious delinquencies</td>
</tr>
</tbody>
</table>

• Individual grade represents contribution

<table>
<thead>
<tr>
<th>Grade</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; Group</td>
<td>Visionary, group MVP</td>
</tr>
<tr>
<td>= Group</td>
<td>Good attitude, hard worker</td>
</tr>
<tr>
<td>&lt; Group</td>
<td>Produce negative work</td>
</tr>
<tr>
<td>D/F</td>
<td>Abandon the group</td>
</tr>
</tbody>
</table>
ENGRC Grading

- ENGRC section also has a grade
  - No extra work; just time for testing/documents
  - New requirement by school of engineering

- All grades except the game grade
  - Charter & Group Reports (15%)
  - Course Documents (75%)
  - Attendance & Presentations (10%)

- Typically higher than course grade
Using CATME for Reports

http://www.catme.org
This Week

- Set up your **CUGL** build environment
  - Download sample project and set it up for your IDE
  - Download other demos from the course website
  - Use Piazza if you are having problems

- Lectures on **game mechanics**
  - Reviewing what you forgot from CS/INFO 3152
  - Augmented with advanced topics next week
  - Getting you ready for the **Concept Document**
Next Week

• **Pitch Session** next Wednesday, Friday
  • 5-10 minute “elevator pitch” for your game
  • Practice with short, concise description
  • Provide some feedback for Concept Document

• **Team Workflow** due at end of the week
  • Want rules of how you interact with each other

• **Concept Document** due in two weeks
  • Slightly different format from Intro course