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 50% of your final grade
  - Interdisciplinary teams of 6-7 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 organized you into sections
  - Groups 1-5 meet Wednesday 12:20-1:10 in Upson 206
  - Groups 6-9 meet Wednesday 2:30-3:20 in Upson 206

- **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 previous 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 is fully supported and stable
  - 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

- Custom game engine “written from scratch”
  - Core set of 100 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
This Engine is Good Enough

**Underhand**

- Strategic card game
- Inspired by *Reigns*
- 1 mill Android downloads

**Family Style**

- Multiplayer Coop
- Featured on App Store!
- Viral in Thailand
- 1.25 mill all platforms
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
  - Orientation detection
  - Arbitrary text input
- Modern OpenGL support
  - OpenGLES 3.1 on mobile
Working With CUCL: The Bad News

- Engine is very spartan
- Box2D is only 3d-party library
- No support for external editors
- No support for rigging
- Windows is *PC only*
- No UWP development
- So no Surface support
- No gesture recording
- Current Meng project
- Not read for use yet
- Audio still a work in progress
About Audio

- Completely new audio engine (no SDL_mixer)
  - Uses a modern DSP mixer graph
  - You have direct access to this graph
  - But can use legacy interface as well

- But all assets must be sampled at 48k!
  - Theoretically can request 44.1k from OS
  - But iOS will ignore you and say 48k
  - We have a resampler, but not fully tested
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>Team Workflow</th>
<th>1/25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 2</td>
<td>Initial Proposal</td>
<td>2/1</td>
</tr>
<tr>
<td>Week 3</td>
<td>Concept Document <em>(Project Kickoff)</em></td>
<td>2/8</td>
</tr>
<tr>
<td>Week 4</td>
<td>Nondigital Prototype Milestone Proposals</td>
<td>2/12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2/17</td>
</tr>
<tr>
<td>Week 5</td>
<td>Gameplay Specification</td>
<td>2/22</td>
</tr>
<tr>
<td><strong>February Break</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>Gameplay Prototype</td>
<td>2/26</td>
</tr>
<tr>
<td>Week 7</td>
<td>Architecture Specification</td>
<td>3/7</td>
</tr>
<tr>
<td>Week 8</td>
<td>Technical Prototype</td>
<td>3/9</td>
</tr>
<tr>
<td>Week 9</td>
<td>Document Revisions Level Design</td>
<td>3/21</td>
</tr>
</tbody>
</table>

---

*Pre-Production*

*Development*
# Semester Schedule

<table>
<thead>
<tr>
<th>Week 10</th>
<th>Alpha Release</th>
<th>3/23</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spring Break</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Week 10</strong></td>
<td>Code Walkthroughs Kickstarter Proposal</td>
<td>4/8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4/11</td>
</tr>
<tr>
<td><strong>Week 11</strong></td>
<td>Closed Beta Release (Feature Complete)</td>
<td>4/13</td>
</tr>
<tr>
<td><strong>Week 12</strong></td>
<td>App Store Page</td>
<td>4/25</td>
</tr>
<tr>
<td><strong>Week 13</strong></td>
<td>Open Beta Release (Open Playtesting)</td>
<td>4/27</td>
</tr>
<tr>
<td><strong>Week 14</strong></td>
<td>Postmortems Final Portfolio</td>
<td>5/4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5/8</td>
</tr>
<tr>
<td><strong>Week 15</strong></td>
<td>GDIAC Showcase</td>
<td>5/15</td>
</tr>
</tbody>
</table>
Kickstarter Video

- This is 2nd year for this “document”
  - Video pitching game to investors
  - Shows off the current build
  - Also shows off the development process

- Useful for applying to Boston FIG
  - This works as your submission trailer
  - Discarded took this approach last year

- But you must document all semester.
Group Management

• Every group has a **project leader**
  • Final say in all *group management decisions*
  • Coordinates designers and programmers

• Every group has a **lead programmer**
  • Responsible for the *code architecture*
  • Delegates coding tasks to others

• Every group has a **lead designer**
  • Responsible for the *visual style and interface*
  • Ensures other designers conform to style
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 (25%)
  - Participation and Reports (20%)
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

24
Using CATME for Reports

http://www.catme.org
This Week

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

• Lectures on *game mechanics*
  • Reviewing what you forgot from CS/INFO 3152
  • Augmented with advanced topics next week

• Set up your **CUGL** build environment
  • Download sample project and set it up for your IDE
  • Use Piazza if you are having problems
This Week

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

• Lectures on **game mechanics**
  • Reviewing what you forgot from CS/INFO 3152
  • Augmented with advanced topics next week

• Set up your **CUGL** build environment
  • Download sample project and set it up for your IDE
  • Use Piazza if you are having problems

When ready at the end of the week
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

- Turn pitch into an **initial write-up**
  - Respond to feedback from pitch session
  - Chance to get even more feedback on idea

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