



the
gamedesigninitiative
at cornell university

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 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 Monday 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

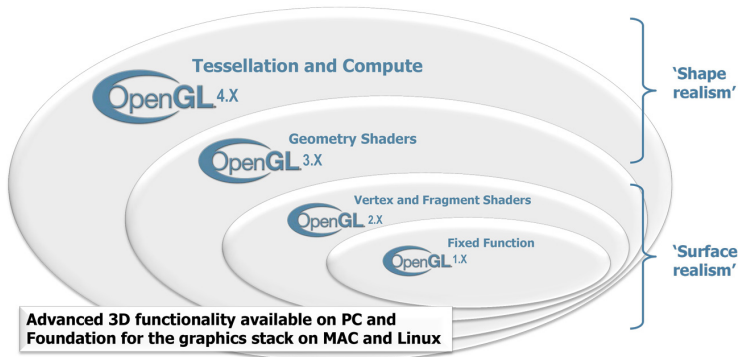
- Custom game engine “written from scratch”
 - Core set of 100 C++ classes
 - Supports input, graphics, and audio
- Layered on top of some useful libraries
 - **SDL**: SimpleDirectMedia Layer
 - **Box2D**: The definitive 2D physics library
- Compatible with any C++ library out there
 - **Example**: Bullet for 3D physics

Working With CUGL: Good News



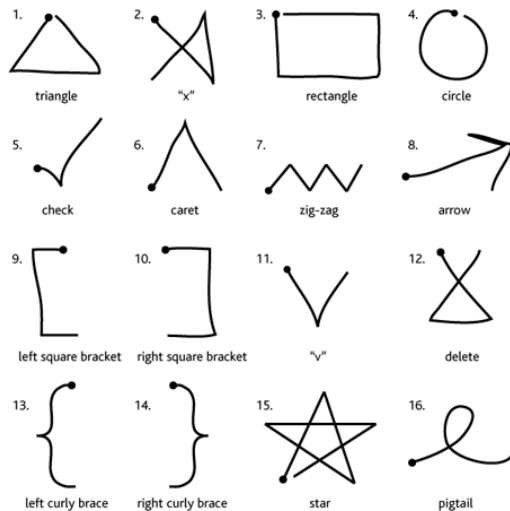
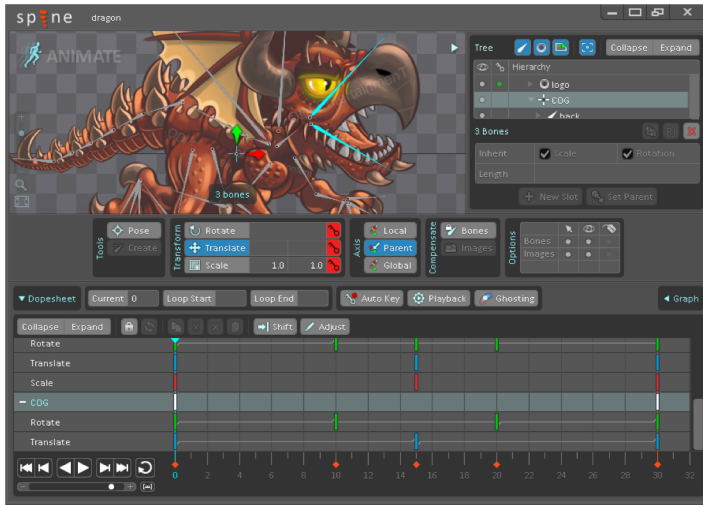
- Supports modern(ish) C++
 - Full C++11 support
 - Heavy use of smart pointers
- Build is very light-weight
 - Engine has 200 MB footprint
- Advanced input features
 - Built-in pinch and rotation
 - Orientation detection
 - Arbitrary text input
- Modern OpenGL support
 - OpenGL ES 3.1 on mobile

OpenGL for Each Hardware Generation



© Copyright Khronos Group 2012 | Page 23

Working With CUGL: 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
 - Was a feature last year
 - But never worked properly
- Audio is 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
 - Have not finished a continuous resampler

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

Week 1	Form Groups	1/26	Pre-Production
Week 2	Team Workflow	2/2	
Week 3	Concept Document (Project Kickoff)	2/9	
Week 4	Nondigital Prototype Milestone Proposals	2/13 2/16	
Week 5	Gameplay Specification	2/23	
<i>February Break</i>			Development
Week 6	Gameplay Prototype	2/27	
Week 7	Architecture Specification	3/9	
Week 8	Technical Prototype	3/11	
Week 9	Document Revisions	3/23	

Semester Schedule

Week 10	Alpha Release	3/25	Development
<i>Spring Break</i>			
Week 10	Code Walkthroughs Level Design	4/12 4/13	
Week 11	Closed Beta Release (Feature Complete)	4/15	
Week 12	Kickstarter Proposal	4/27	
Week 13	Open Beta Release (Open Playtesting)	4/29	
Week 14	Final Portfolio Postmortems	5/6	Release
Week 15	GDIAC Showcase	5/17	

Kickstarter Video

- This is a new “document” this year
 - 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 *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

Grade	Criteria
A	Bug-free, Fun-to-play
B	Complete and playable
C	Complete but unplayable
D/F	Serious delinquencies

- Individual grade represents contribution

Grade	Criteria
> Group	Visionary, group MVP
= Group	Good attitude, hard worker
< Group	Produce negative work
D/F	Abandon the group

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



Report

[View Comments](#)[View Raw Data](#)[Return to Main Page](#)

Class	Term	Format	Prof	School
am Review	ME 316	Fall 2015	Lecture	Leachman Washington State University

 Enable pop-up texts Show raw "Adjustment Factor"[Re-Display](#)Search:

Team ID	Contrib. to Team	Interact w/ Team	Keeping on Track	Expect Quality	Adj Factor (w/ Self)	Adj Factor (w/o Self)	Note
01	4.2	4.4	4.0	4.2	1.05	1.05	Under
01	3.6	4.2	4.0	3.4	1.00	1.00	
01	3.8	4.0	3.6	3.8	1.00	1.01	
01	3.0	4.2	3.6	3.4	0.91	0.87	
01	3.8	4.2	4.2	4.0	1.04	1.04	
02	3.8	4.2	3.8	4.0	1.00	1.00	
02	3.8	4.2	3.8	4.0	1.00	1.00	
02	4.5	4.2	3.8	4.2	1.04	1.02	
02	4.2	4.2	3.8	4.0	1.01	1.01	

<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