Lecture 1

Course Overview

Welcome to CS/INFO 5/4152

- Course is **completely full**
  - Had 95 applications for 72 (8x9) spots
  - But thankfully less than the 120 last year.
  - Increased groups to 9 meet demand
- Still a few possibilities if you are waiting
  - Not everyone has responded to an invite
  - Need to shore up skills in some teams
- If not in the class, talk to me afterwards

CS/INFO 4152: Advanced Topics

- Sequel to CS/INFO 3152
  - Prereq **unless** a non-Cornell grad (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 8-9 people
- Also design documents

CS/INFO 5152: Master's Version

- **Game Labs**: Similar to introductory course
  - Done outside of class for first three weeks
  - Special labs for programming or design
  - Complete according to your project role
  - Only INFO has a choice; CS is programming only
- **Mastery Topic**: Part of your individual grade
  - Must contribute something significant to project
  - Grade based on evaluation of your peers
  - Must be proposed during first beta release

CS/INFO 5152: Master’s Version

- **Game Labs**: Similar to introductory course
  - Done outside of class for first t eeks
  - Special l n
  - Complete ject role
  - Only INFO has a choice; CS is programming only

4152 students do also; but work is not graded

- **Mastery Topic**: Part of your individual grade
  - Must contribute som to project
  - Grade ba rs
  - Must be p ng first beta release

4152 students not required to do this

Game Development

- Uses the familiar **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/cs5152

cs4152 redirects

- Games demonstrated at **Showcase**
  - Once again, will open it up to the public
  - Public reaction is part of your grade
  - Submissions posted on the GDIAC website

Course Structure

- Most things happen during the “lecture” section
  - Meets three days a week (M,W,F) 9:05-9:55
  - 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

- Most things happen during the “lecture” section
  - Meets three days a week (M,W,F) 9:05-9:55
  - 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 topi … course (this is n …
  - **Design F** … testing
  - **Technica** … mobile platforms, memory management

There are **NO** C++ lectures.
Learn online and in the labs.

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 1-6 meet Wednesday 12:20-1:10 in **Rhodes 253**
  - Groups 7-9 meet Wednesday 2:30-3:20 in **Phillips 213**
- **Undergrads**: You must enroll in ENGRC 4152
  - Extra credit hour for work you are already doing
  - This is ***required***; it is not optional

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 1-6 meet Wednesday 12:20-1:10 in **Rhodes 253**
  - Groups 7-9 meet Wednesday 2:30-3:20 in **Phillips 213**

- **Undergrads**: You must en 4152
  - Extra cre ng
  - This is ***re*** ar

ENGRC does not apply
to grad students in 5152

Game Requirements

- Should be **mobile game** on iOS or Android
  - Develop cross-platform, but graded only on one
  - But **an exception** coming on the next page…
- 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 an App Store

What is the Exception?

What is the Exception?

Do **not** just want a PC game.
Must leverage its unique features.

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 years
- We do **not** provide any hardware
  - Mobile devices are about $200; used are cheaper
  - Cheapest Steam Deck is about $400
  - Just need one device for your whole group
- Either 2D or 3D is acceptable

Choosing a Platform

- You **must** develop iOS apps on a Macintosh
  - Only XCode can load the app on to a device
  - Do not need Apple Developer membership
  - But need membership ($100) to put on store
- 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

But Conversely

- You **must** develop Steam Deck on x86 Linux
  - Means a computer with an Intel or AMD chip
  - Possible if you have a really old Mac (unlikely)
  - More likely a partition on a Windows Machine
- **Any distribution** is acceptable
  - We have tested it on **Ubuntu**
  - Steam suggests **Manjaro** (closest to Steam Deck)
  - Need GCC, CMake, and Flatpak installed

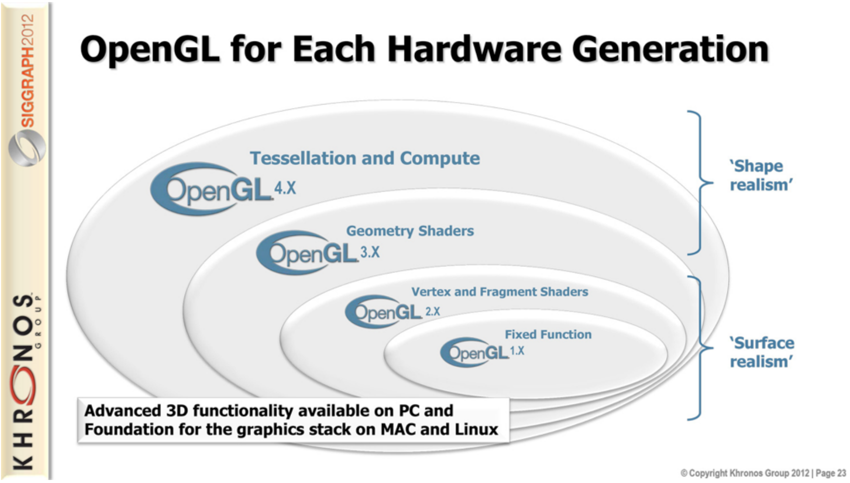
Working in C++

- Best option for cross-platform development
  - **iOS**: Obj-C and **C++**; **Android**: Java and **C++**
  - Important to understand if move to Unreal
  - See the online lectures to learn more
- 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 170 C++ classes (70k lines of code)
  - 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



KHRONOS GROUP

SIGGRAPH2012

OpenGL for Each Hardware Generation

Tessellation and Compute — OpenGL 4.X

Geometry Shaders — OpenGL 3.X

Vertex and Fragment Shaders — OpenGL 2.X

Fixed Function — OpenGL 1.X

'Shape realism'

'Surface realism'

Advanced 3D functionality available on PC and Foundation for the graphics stack on MAC and Linux

© Copyright Khronos Group 2012 | Page 23

- Supports modern(ish) C++
  - Full C++17 support
  - Heavy use of smart pointers
- Build is very light-weight
  - Engine has ~40 MB footprint
- Advanced input features
  - Built-in pinch and rotation
  - Orientation detection
  - Arbitrary text input
- Modern OpenGL support
  - OpenGLES 3.1 on mobile

CUGL Continues to Evolve

New Features for 2025

- Improved animation tools
  - No longer tied to scene graph
  - Can interpolate anything!
- Experimental 3D support
  - Support for OBJ, billboards
  - Also 3D/2D particle systems
- Improved Figma Support
  - Can design UIs with no code
  - But there are limitations

Did Not Quite Make It

- Still no Vulkan support
  - We were almost there!
  - Ran out of time before start
- Javascript support on hold
  - Intended to help UX designers
  - But PocketPy a better option
- SVG support still not there
  - Vulkan work delayed it
  - Negatively affects Figma

CUGL Continues to Evolve

New Features for 2025

- Improved animation tools
  - No longer tied
  - Can interpolate
- Experimental 3
  - Support for OE
  - Also 3D/2D particle systems
- Improved Figma Support
  - Can design UIs with no code
  - But there are limitations

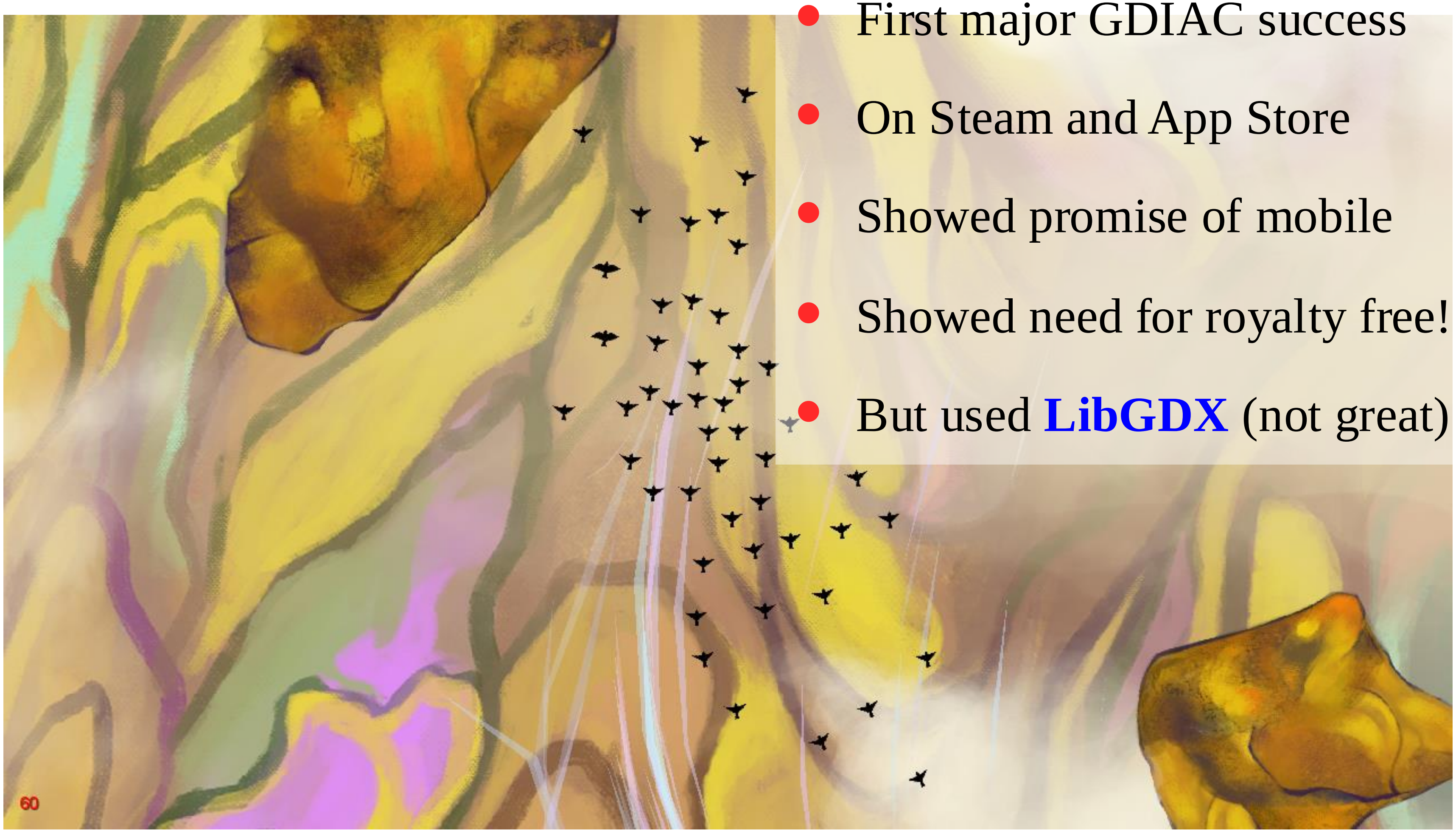
Did Not Quite Make It

- Still no Vulkan support
  - most there!
  - ime before start
- pport on hold
  - help UX designers
  - But PocketPy a better option
- SVG support still not there
  - Vulkan work delayed it
  - Negatively affects Figma

Is this good enough?

Why not use Unity?

2013: *Gathering Sky*



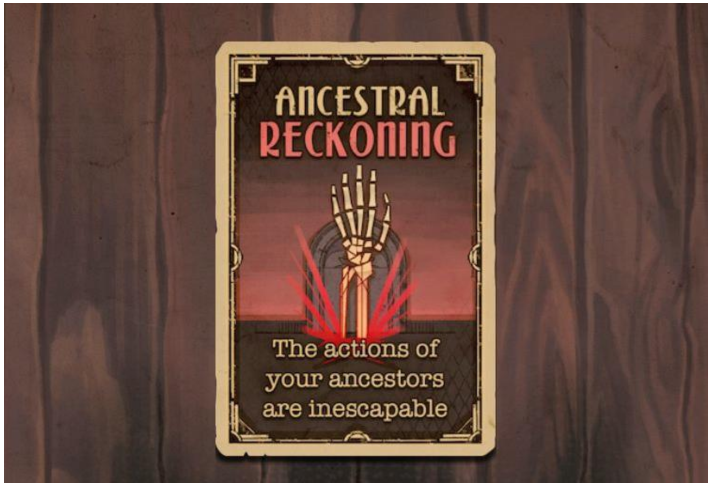
- First major GDIAC success
- On Steam and App Store
- Showed promise of mobile
- Showed need for royalty free!
- But used **LibGDX** (not great)

2013: *Gathering Sky*

- First major GDIAC success
- On Steam and App Store
- Showed promise of mobile
- Showed need for royalty free!
- But used **LibGDX** (not great)

Three years later…

2016: CUGL 1.0 Released

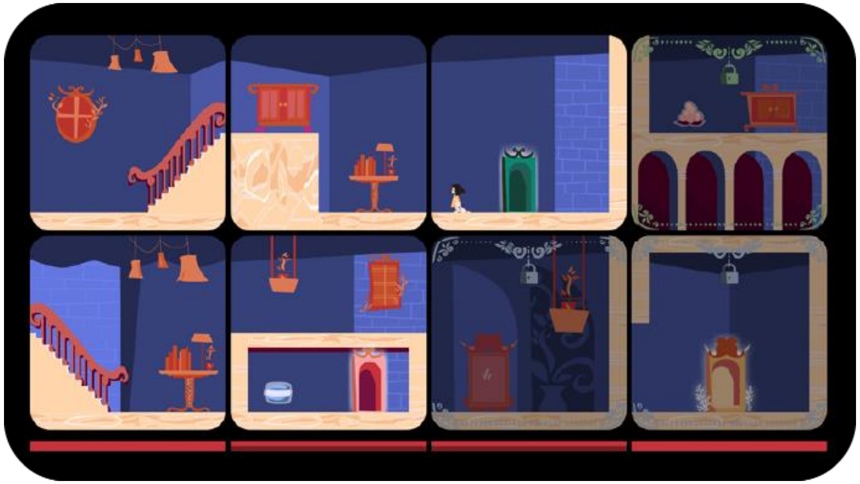


Underhand

- Strategic card game
- Inspired by *Reigns*
- Went viral on Reddit
- **1 mill Android downloads**

Manic Moving Mansion

- Real time puzzler
- Reorder rooms to guide player
- **Best Student Game** at BFIG
- Beats MIT Media Lab!



2016: CUGL 1.0 Released

Underhand

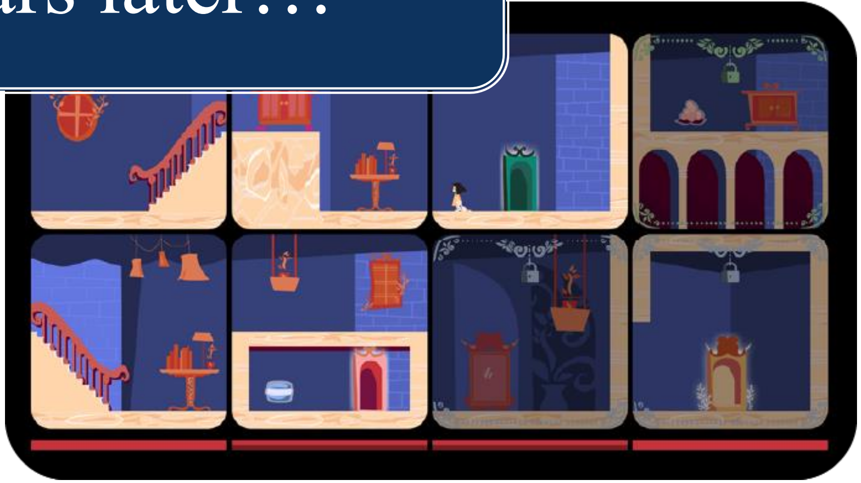
- Strategic card game
- Inspired by *Reigns*
- Went viral on Reddit

ownloads

Three years later…

Manic Mov

- Real time puzzler
- Reorder rooms to guide player
- **Best Student Game** at BFIG
- Beats MIT Media Lab!



2019: *Family Style*

A CO-OP KITCHEN CLAMBER!
FOR 3 TO 8 PLAYERS
PASS INGREDIENTS BY SWIPING TO YOUR NEIGHBORS!
FINISH PLATES TO SCORE POINTS
OVER 120 RECIPES TO DISCOVER

- Multiplayer Coop game
- Front page of the App Store!
- Went viral in Thailand
- 15k actively daily users
- **2 million downloads**

2019: *Family Style*

A CO-OP KITCHEN CLAMBER!
FOR 3 TO 8 PLAYERS
PASS INGREDIENTS BY SWIPING TO YOUR NEIGHBORS!
FINISH PLATES TO SCORE POINTS
OVER 120 RECIPES TO DISCOVER

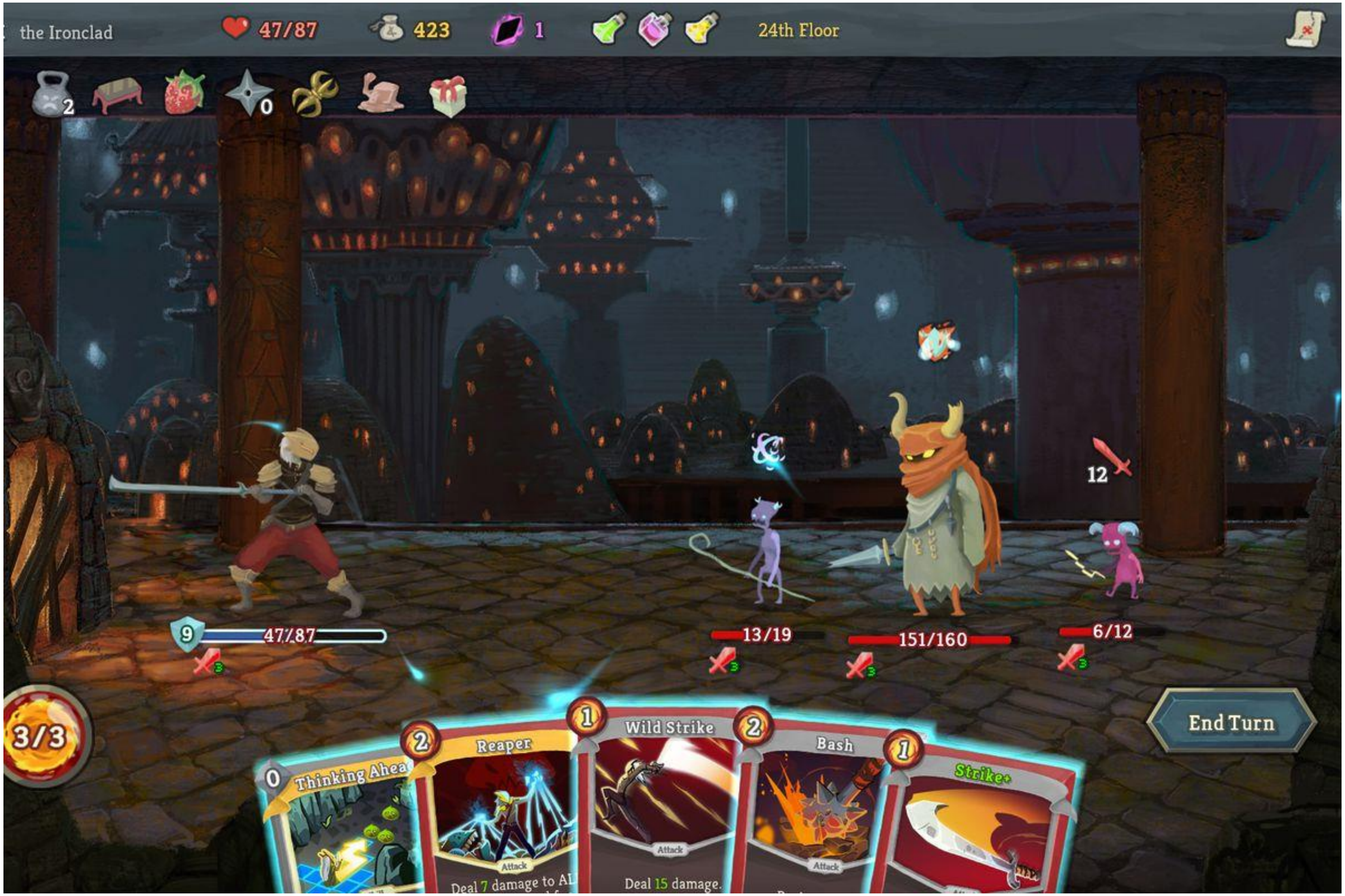
- Multiplayer Coop game
- Front page of the App Store!
- Went viral in Thailand
- 15k actively daily users
- **2 million downloads**

What Should You Make?

- This course is not just CS 3152 Part 2
  - Want your games to be different in some way
  - Mobile enforces this to some degree
- We have removed (almost all) restrictions
  - Can make a narrative-heavy game
  - Not limited to single-player games
- But it still must be **feasible**!
  - 3152 alums have the experience here

Rogue-Lites are Okay!

Deck-Building Games Are NOT



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	Team Workflow	1/25	Pre-Production
Week 2	Initial Proposal	2/1	
Week 3	Concept Document (**Project Kickoff**)	2/8	
Week 4	**Nondigital Prototype** Milestone Proposals	2/12 2/15	
	February Break		
Week 5	Gameplay Specification	2/22	Development
Week 6	**Gameplay Prototype**	2/26	
Week 7	Detailed Specifications	3/8	
Week 8	**Technical Prototype**	3/10	
Week 9	Document Revisions	3/22	

Semester Schedule

Week 10	**Alpha Release**	3/24
	Spring Break	
Week 10	Code Walkthroughs App Store Page	4/7 4/12
Week 11	**Closed Beta Release (Feature Complete)**	4/14
Week 12	Document Revisions	4/26
Week 13	**Open Beta Release (Open Playtesting)**	4/28
Week 14	Postmortems Final Portfolio	5/5 5/7
Week 15	GDIAC Showcase	5/17

Development

Release

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

Group Management

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

> - **Optional**: **lead user specialist**
>   - Get the game in the hands of players
>   - Record and *analyze all playtesting results*

- Every group has a **lead designer**
  - Responsible for the *visual style and interface*
  - Ensures other designers conform to style

Grading: 4152 vs 5152

Group Grades	4152	5152
Group Game Grade	25%	25%
Course Documents	25%	15%
Presentations	5%	0%*

Individual Grades	4152	5152
Game Grade	25%	20%
Participation	20%	10%
Activities/Labs	0%*	15%
Mastery Topic	0%*	15%

Grading: 4152 vs 5152

Group Grades	4152	5152
Group Game Grade	25%	25%
Course Documents	25%	15%
Presentations	5%	0%*

Mastery

Individual Grades	4152	5152
Game Grade	25%	
Participation	20%	
Activities/Labs	0%*	15%
Mastery Topic	0%*	

Participation

Game Grade

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

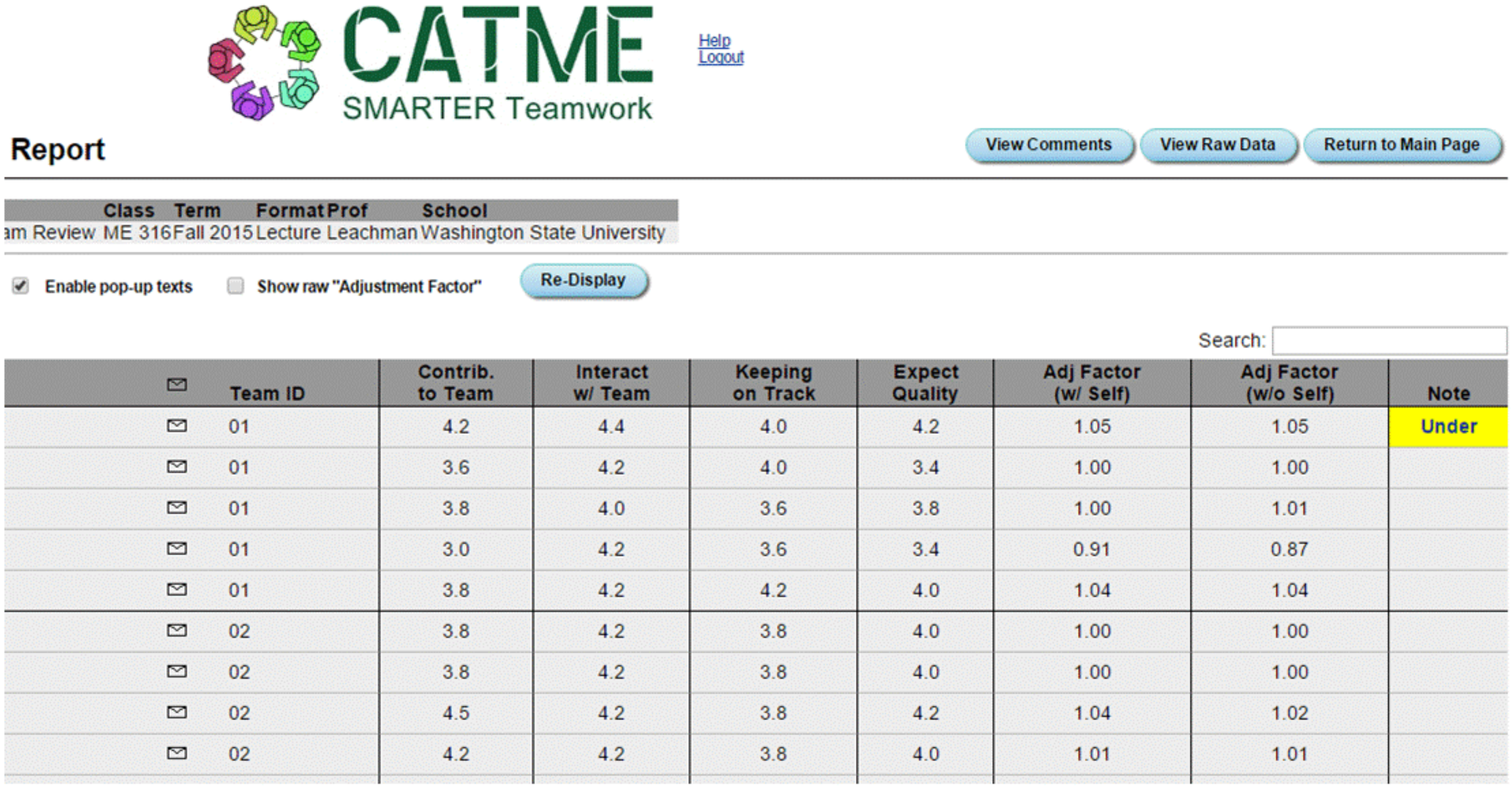
Policy on Generative AI

- We **DO** allow generative AI in the course
  - We conform to Steam’s policy on such context
  - All data sets must be clean (properly licensed)
- However, you might not find it helpful
  - Last year it was often a source of negative work
  - It also caused significant friction between artists
- Come up with a **policy for your team**
  - Violation will constitute a “creative conflict”
  - Such conflicts are addressed in your team workflow

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
  - Workflow & Group Reports (13%)
  - Course Documents (77%)
  - Attendance & Presentations (10%)
- Typically higher than course grade

Using CATME for Reports



Report

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

☑ Enable pop-up texts ☐ Show raw "Adjustment Factor"

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

- **Team Workflow** due at end of the week
  - Want rules of how you interact with each other
- Lectures on **mobile game design**
  - Video to review material from CS/INFO 3152
  - Lecture to delve deeper into mobile mechanics
- Set up your **CUGL** build environment
  - Download sample project and set it up
  - Programmers start the first game lab

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