Course Introduction

Background

- Competent in a language like C++, Java, or C#

- Have taken CS 2420, CS 3100 or equivalent

- Useful to have taken any of CS 5200, CS 5400, CS 5600, CS 5700

What Is The Course

- **My Purpose:** E*mphasize and integrate multiple Computer Science topics using game development as the platform*

  - We are writing games and learning about game development, but surprisingly, that is a secondary purpose

  - To that end, it isn't a course on using a game engine to create games

- Lot's of game topics, it **is** a game development course!

- Platform choices: Concepts apply across implementation platforms

  - HTML & JavaScript: lead platform

    - Many of you will write a lot of JavaScript professionally

  - C# & MonoGame: supported platform

    - C# Widely used in industry

    - MonoGame provides for deployment to mobile phones and consoles

- Large project and team collaboration

What This Course is NOT

- How to use Unreal, Unity, GameMaker, etc.

- Computer Graphics course

- Game Design

- Educational Games

- Gamification of <whatever>

Other Thoughts

- Expectation that students will have to find and learn material not necessarily presented in class

- Don't assume strong background with JavaScript; provide some instruction during the first three weeks

  - Will also provide some C# info

- Weekly quizzes during the first part of the semester; outside of class, using Canvas

  - Often including questions from game dev videos I'll have you watch

- Once the final project starts, will take the latter part of the weekly class to review project/team status

Is Game Development Computer Science?

- Software Engineering (2450)
- Languages/Scripting (multiple)
- Algorithms (2420)
- Data Structures (2420)
- Distributed/Parallel/Multi-Core (5030/5200/5500)
- Networking/Internet/Web (2610/5200)
- Graphics (5400)

- Math/Physics
- Artificial Intelligence (5600)
- Compilers (5300)
- Database (5800)
- Operating Systems (3100)
- Optimization
- GUI Development (2410/2610/3200)

Industry Recommendations

Additional Recommendations

ELECTRONIC ARTS™

Other Important Academic Experiences

- Technical
  - Open-ended Problem Solving Skills
  - Debugging
  - Full-cycle Projects (from prototyping to ship – with firm deadlines!)
  - Complex Application Development
    - Experience with larger code base

- Non-technical
  - Teams larger than 2 people
  - Collaboration with non-technical people
  - Communication Skills
  - Project Management (requirements, scheduling, customer expectations)
  - Force students to "show and tell" & accept feedback