ETGG3802

Lecture1: Welcome & First topic!

Compared to 3801

- 3801
  - Learning about commercial game engines
  - Use C#/Blueprints/C++ for *scripting* of game object logic.
  - 4/10 on the JDS (Jason Difficulty Scale)
  - Mostly individual, group work towards the end
- 3802
  - Making our own game engine
  - Make an Engine in C++ *and* use (Python) for *scripting* of game object logic.
  - 10/10 on the JDS
  - Mostly individual, group work towards the end

Why?

- C++ is hard…and that's why we should use it.
- Cover interesting Software Engineering problems
- Gives you a taste of creating a custom engine (good and bad)
- A standout portfolio piece…if you take it seriously

Class Flow

- 100 points per week of class
- Phase I (8 – 10 weeks)
  - Individual work only
  - Lecture on Tuesday.
  - Lab due before class following Tuesday.
  - Thursday lab day
  - Everyone doing the same thing – core engine features
- Phase II (5 – 7 weeks)
  - Divide up labor
  - Version Control
  - 1 – 2-week sprints
  - Advanced features:
    - Physics
    - Networking
    - …
- Quizzes / Exams if…

How to do poorly in this class

- Normal stuff
  - Missing class
  - Being <95% engaged during lecture.
  - Not thinking ahead
  - Not starting labs immediately
- (Sort of) unique to this class
  - Assuming you'll be able to find everything on Stack Overflow
    - You won't☺
  - NOT ASKING QUESTIONS
    - Best = Tuesday
    - Next Best = Thursday
    - Next Worst = Office Hour
    - Worst = Email

Topics / Labs Planned

1. Setting up the project, linking
2. LogManager
3. Singletons, Basic Python Scripting
4. Entity Component System (Unity GameObject + Components)
5. InputManager
6. More Advanced Python scripting (callbacks)
7. SceneManagement
8. Space Invaders, ptI (entirely in Python)
9. Space Invaders, ptII
10. …(group work)

Lab1 Review Topics

- Visual Studio 2019 / C++ refresher
  - create project
- linking
- OOP