

Advanced Games

Game Code Architecture

Week 1 Game Review

What are you proud of?

What are you *ashamed* of?

What would you do differently?

How would your game scale?

What does a game need?

- **Content**

- 3d Models, Shaders, Textures, Text, Fonts, Music, Video, Saves, levels/gamestate.

- **Processing & io & Mechanics**

- Rendering, User input, Networking, Physics, AI, Gameplay rules.

- **Q:** When does a game need any of the above?

- **A:** Right Now (and without warning)

Abstraction

And so we build **Games Engines**

But do we need them?

We didn't always have them.

Q: How complex do you think a game needs to be before you think you need to write Engine Code?

When to start the engine

A:



A: Once your code gets abstract enough

A: From the start

A: Never, and write some crazy_[fast/bad] code

But in reality

3 people in a team, you will need some good code structure to support you.

And basing on the TRC, you will needs some pretty robust and abstract code

Let's restart

Advanced Games

How to Build a Game Engine

Step 1

Build the Wall

Game Code

User Input, File IO, job Scheduling

Engine Code

User Input, File IO, job Scheduling

