01 - Introduction

CS 3160 - Game Programming
Max Gilson

CS 3160 - Game Programming

- Instructor
  - Max Gilson
  - BSEE - Wright State University 2018
  - MSEE - Wright State University 2019
  - 8+ years experience in industry in hardware and software development
  - Published games as a solo developer and on a team
    - Started in 2006 with Scratch

Syllabus

- Lecture
- Projects
- Quizzes
- Software
- Pilot
- Schedule
- Grading
- Late Assignments

AI Generative Work and Copied Work

- Using generative AI, like ChatGPT, to complete your quizzes, labs, or exams will result in a 0 for the course.
- You must be capable of writing your own code for your assignments without copying and pasting from other sources
- Copying work from others, online resources, or generative AI is an academic integrity violation

How to Get Help for Assignments Steps

Once you need help on an assignment:

1. Ask me during lecture
2. Come to my office hours
3. Ask question in email to me
    a. I will not review your code over email

Note: If asking for help 1 or 2 days before the project is due don’t expect immediate help!

Projects

- Project 1 starts this week

What is “Game Programming”

- Game programming is only one part of game development
- Game development can involve many disciplines
  - Programming (writing the code)
  - Game design (creating the rules of the game)
  - Visual art design (creating 3D models or 2D sprites)
  - Writing (writing the story or dialogue)
  - Audio art design (creating sounds or music)
  - Quality assurance (finding bugs)

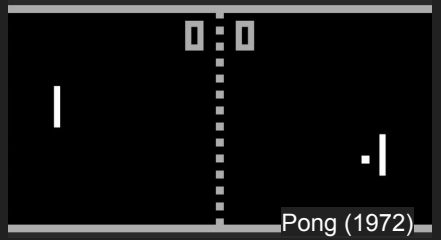
Game Programming

- This course will focus on game programming, not game design or complete game development
  - We'll touch on game design and development when we need to throughout the semester
- A game designer decides what must be in the game
  - "We need a double jump in this game"
  - "This game must be in first-person perspective"
- A game programmer figures out how to write the code to satisfy the requirements of the game designer and other developers

Some History

- In the beginning, game programming was almost entirely in assembly
- Computers were barely fast enough to run even simple games
- Most games were built by a single person in a few months
- As technology improved, games became larger and more sophisticated

Bertie the Brain (1950)


Pong (1972)

Super Mario Bros. (1985)


Ultima Underworld (1992)


World of Warcraft (2004)


Minecraft (2011)


Elden Ring (2022)

Game Programming Today

- Today, there exist many tools to help you develop games:
  - Game engines (Unity, Unreal, Godot)
  - Physics systems (PhysX, Euphoria)
  - Audio systems (FMOD, Wwise)
  - Debugging (RenderDoc, Visual Studio Debugger)
  - Networking (Photon, Socket.IO)
- These tools make it easier to create more sophisticated games
- Learning these tools can make it easier to get a job in game development
- Large companies will often have proprietary versions of these tools
- Or you can always write these from scratch (not recommended)

The Reality of Game Programming

- Creating games is hard
- Requires multidisciplinary focus if done solo
  - Programming, art, music, writing, game design
  - Hire/buy what you cannot do well
- Even “small” games take years to build
  - Undertale took 3 years to make with 1 developer
- Large games are made by 100’s of people
  - Skyrim took 6 years to make with 100+ developers
- Start by making small games (the smaller the better)
- Companies want to hire developers that have skills in common tools
  - Check job postings to know for sure