Syllabus of Record

Program: CET Siena
Course Code / Title: (SN/CS 4730) Computer Game Design
Total Hours: 45
Recommended Credits: 3
Primary Discipline / Suggested Cross Listings: Computer Science / Data Science
Language of Instruction: English
Prerequisites / Requirements: For UVA students: CS 2150 Program and Data Representation or CS 3140 Software Development Essentials. For students from another institution: prior coursework in Data Structures and Programming at same level required.

Description

This course introduces students to the concepts and tools used in the development of modern 2-D and 3-D real-time interactive computer video games. Topics covered include graphics, parallel processing, human-computer interaction, networking, artificial intelligence, and software engineering. The course focuses on both the design and technical aspects of creating a game, from concept inception and prototyping through coding and playtesting.

Objectives

Upon successful completion of this course, students will be able to:

- Understand the social and ethical context in which video games are developed, marketed, and played.
- Understand the technologies and platforms upon which modern interactive video games are developed.
- Understand the software engineering concepts necessary to develop video games (and other large systems) in a large development group.
- Comprehend the computational theory used in video games design, as well as, to a lesser extent, related fields (artificial intelligence, computer graphics, networks, etc.).
- Understand the theoretical topic of game theory, and how that applies to multi-player games (and, to a lesser extent, artificial intelligence).

Course Requirements

Active participation is essential in this course. Students are expected to attend each class and lab session, as outlined in the CET Attendance Policy. Students are expected to read all assigned materials and complete homework before each class session. Reading assignments are generally 20-30 pages per class session.

Graded assignments include:

- **Homework/Lab:** periodic assignments (usually every two weeks) designed to assess students' engagement with the topics covered throughout the course

- **Midterm Exam:** a written test taken in class (including open questions and coding exercises) to be done using the student's personal computer.
- **Final Exam:** a written test taken in class (including open questions and coding exercises) to be done using the student's personal computer.

Grading

The final grade is determined as follows:

- Participation: 10%
- Homework/Lab: 50%
- Midterm Exam: 20%
- Final Exam: 20%

Readings

Fullerton, Tracy. *Game Design Workshop: A playcentric approach to creating innovative games*. Boca Raton: CRC Press, 2024.

McGonigal, Jane. *Reality is broken: Why games make us better and how they can change the world*. London: Vintage, 2012.

Additional Resources

"Godot Docs – 4.3 branch." Godot Engine. https://docs.godotengine.org/en/stable/index.html.

StayAtHomeDev-Git. "Stayathomedev-Git/FPS-Godot-BASIC-Setup: A Basic FPS Controller Setup with Keyboard Movement, Mouse Look, Jumping, and Gravity." GitHub. https://github.com/StayAtHomeDev-Git/FPS-Godot-Basic-Setup.

StayAtHomeDev. "Make An FPS in Godot 4." YouTube video, 13:24. October 3, 2023. https://www.youtube.com/watch?v=N-jh8qc8tJs

Stopshut. "Every Game Engine Explained In 9 Minutes (Part 1)." YouTube video, 9:09. February 11, 2024. https://www.youtube.com/watch?v=WxaXmB-tPBE.

Outline of Course Content

Topic 1: Introduction to Game Development

- What are video games?
- Game classification by genre and platform
- Business models

Topic 2: Game Design Elements

- Mechanics
- Gameplay

- Level design
- Narrative
- Balancing

Topic 3: Game Development Tools

- What are game engines?
- 2D engines: MonoGame, Unity and Godot
- 3D engines: Unreal, Unity and Godot
- Asset development workflow

Topic 4: Game Physics

- Introduction to physics simulation
- Collision system in Python
- Collision system in Godot

Topic 5: Graphics & Sound

- Graphics: PBR or not PBR?
- Ray-tracing vs Rasterization
- Sound and immersiveness
- Voice actors and sound effects

Topic 6: Third Person Shooter with Tanks in Godot

- Game vision and gameplay design
- Assets implementation
- Single player game code implementation
- Design of the enemy AI

Topic 7: Multiplayer Functionalities in Godot

- Networking basics: IP, ports, TCP and UDP
- GDScript Remote Procedure Calls
- Peers and server interaction model
- Multiplayer game code implementation