Audio Engine Development

Winter 2017

Gam 598

instructor:	Ed Keenan
email:	ekeenan2@depaul.edu
office hours:	Thursday 3-5pm Office, 9-10 Classroom or by email appointment
office:	CDM 830
phone:	(312)362-6747
website:	piazza.com/depaul/winter2017/gam598398 (Preferred communication)
lecture:	Lewis 1107, Wednesday, 5:45-9:00pm
Desired to Learn (D2L):	d2l.depaul.edu (Grades, Viewing lectures, Announcements)
Version Control:	perforce: ***140.192.39.61:1666***

Description:

This course discusses the design and principles audio engine development in a game environment. Students will be design, architect and implement a full functioning audio game engine. Exploration of comparative commercial Audio engines and the unique game engine requirements will aid in the understanding and design of the custom engine API. This class will focus on the runtime audio engine as well as the resource conversion and runtime management of the audio assets in the game. Topics include streaming, looping, playing resident sounds, sequencing together sound events, mixing, audio DSP effects, triggering sound from game events and more

Prerequisites:

- Required:
    - CSC 361/461: Optimized C++
        - Data Structures II and Proficient C++ programming is an absolute requirement
        - Introduction to Object Oriented concepts

Learning Goals:

- Students will be able to understand the audio engine requirements in a game environment.
- Students will be able design, development and implement a custom real-time sound engine.
- Students will be able to develop Audio features such as streaming, looping, mixing, event cuing, wet effects and scripting.
- Students will be able to convert, load and manage common audio formats.
- Students will learn how to create real-time Data Driven audio environment allowing the run-time object data to control the behavior and flow of an application.
- Students will reinforce their Software Engineering practices such as Iterative development, Testing, continuous integration and documentation.

Grading

Weekly

20% - Weekly Code/Video Submissions (approx. 8 assignments)

Week 6

15% - Milestone 1: Prototype: Basics Audio Engine

Week 11

65% - Milestone 2: Finished Audio Engine, Demo, and Documentation

Textbooks and printed resources

Course material will be many supplied through class notes, handouts or online links.

- Will be provided by the instructor
- Lectures, links, SDKs and other corresponding material

Software

- ***Microsoft Visual Studio 2015 Enterprise Edition***
  - Visual Studio Enterprise 2015 with update 3 32/64-bit
  - C# install and C++ (future classes)
  - Microsoft Visual Studio 2015 is not used in this class.
- ***Perforce - Visual Client (p4v)***
  - www.perforce.com
- Download and configuration instructions will be provided in class
  - Server address: ***140.192.39.61:1666***

Topics will include:

Milestones

Phase 1: Basics Audio system Driver

- Audio Low Level Driver Driver:
  - Research and determine base on our requirements
- Understanding the API of the system
  - Load, stream, play sound sounds
  - Manage sound wav files
- Comparison of existing solutions
  - Features and ideas
- Understand the restrictions / constraints of Driver
  - Focusing on understanding the restrictions and constraints
- Tools needed for development
  - Audio conversion
  - Audio preview

Phase 2: Sound System Components

- Asset converter – convert and pack sounds into a resource file (Bank)
- Audio Samples – sound wave data (PCM or ADPCM)
- Playlist – Script describes how play a sound event (wave data on hardware tracks)
- Sound Calls – Handle to control the playlist, volume, pan, pitch, priority
- Streaming – dynamically loading audio files
- Pipeline - queue of sounds
- Banks – collection of sound calls and associated
- Bus - virtual group or sounds associated together to adjust their attributes collectively
- Mixing class – enumerated types
- Tracks – hardware voice management
- Effects - Dry
- Effects – Wet (DSP)
- Resource Management
- Hardware controls – output/input

Phase 3: Design / Implementation / Documentation an Audio Engine

- Design the sound engine system
- UML diagrams
- Design Patterns
- Sound Components
- Interactive demo

Milestone 1: Prototype Basics Audio Systems (15%)

Students will design and build many discrete audio systems. The goal is to understand each component how it interacts with the low-level API. Shake out many mini-demos to prove that the completed audio engine is achievable. Engine in C++, with video and summary of each component.

- Many mini demo prototypes:
    - Asset converter
    - Playlist
    - Sound Calls
    - Streaming
    - Looping
    - Pipeline
    - Banks
    - Bus
    - Tracks
    - Effects - Dry
    - Effects – Wet (DSP)
    - Resource Management
    - Hardware controls – output/input

- Complete Code base
    - 100% working game stored in perforce
    - Needs to compile and run
    - Not compiling or working (FAIL)
- Feature List
    - Self-grading feature checklist (supplied PDF)
    - Link to Video Demo
- Video Demo
    - YouTube Video Demo
    - Demo of the game
    - Identifying features in the game
    - 5-10 minutes with clear audio commentary

Milestone 2: Finished Audio Engine (65%)

Students will design and build a single runtime audio game engine library written in C++. In addition to the runtime engine, students will design an asset converter (stand-alone tool) will assemble the audio assets and metadata used in the real-time. A compelling demo to show off the newly constructed engine.

- Asset converter
    - Stand-alone tool
- Single integrated C++ library
    - Incorporates the components fully working from Milestone1
    - Static Library
- Compelling Demo
    - Show off your work
- Documentation
    - Description of the features
    - Architecture design document (TBD)
- Complete Code base
    - 100% working game stored in perforce
    - Needs to compile and run
    - Not compiling or working (FAIL)
- Feature List
    - Self-grading feature checklist (supplied PDF)
    - Link to Video Demo
- Video Demo
    - YouTube Video Demo
    - Demo of the game
    - Identifying features in the game
    - 5-10 minutes with clear audio commentary

Piazza Discussion forum

- Statistics show: students who participate more and help other students do better!
    - The correlation is ridiculous!
        - Poor understanding / poor participation.
        - Great understanding / Great participation
    - As you master the material, help others learn!
        - Want to be a Master programmer so master it!
- Everyone is ***expected*** and encouraged to participate on the Piazza discussion forum. All class-related discussion here this term.
    - At least one real question or response per week from every student.
- Everyone is expected to keep up with the material on Piazza and are responsible for its content. Critical class updates and directions will be presented there.
    - Not participating or reading the material on Piazza is not an excuse.
- All correspondence that is not personal in nature should be vectored through Piazza
    - Sensitive material, use Piazza private note, not email.
- The quicker you begin asking questions on Piazza (rather than via emails), the quicker you'll benefit from the collective knowledge of your classmates and instructors. I encourage you to ask questions when you're struggling to understand a concept.
- Keep the forum professional and positive, help each other out.
    - Karma really pays off here.
    - Help each other whenever you can.
        - There will be a section where you'll need help (trust me).

NOTE: Do ***NOT*** post until you have watched the entire lecture ***FIRST*** (in class or online)

This will prevent frustration on all sides (members asking or answering questions)

Perforce Submissions

- Everyone is expected to submit several submissions to perforce a week.
    - Minimum 5 significant (real) submissions on 3 separate days.
    - To promote incremental development and prevent last day rush.
    - Grade deduction will occur if not followed
- The biggest reason students get into trouble with software design:
    - Not starting the project early
    - Not working on the material frequently enough
    - Taking too large of a bite(byte) of the design
- Both are minimized with this *Perforce RULE*
- Even my simplest programs take 10-20 submissions.
    - For these project assignments my average is 40-400 submissions, so 5 will be no problem.
- Detailed perforce changelist comments are expected

Collaborating together on programming assignments

- You are encouraged to work together
    - Use the Piazza forums heavy
    - Even share your material with others in the common directory
        - Obviously not the answers
- Everyone is 100% responsible for the work they do.
    - If you get help with a section of code,
    - Please refactor the code the ***snot out of it***
        - Comment and understand that material
        - Transform the code to ***make it yours***
    - Be able to answer ***any*** question regarding the code you commit
- System for Detecting Software Plagiarism
    - We will be using MOSS - Measure of Software Similarity (Stanford University)
        - Indicates possible code infringements (plagiarism)
        - MOSS - will detect the similarity independent of naming convention, indentation style or formatting, it compares abstract syntax tree of your code.
    - I will pursue any plagiarism/integrity violations aggressively, arguing for full expulsion from the university for the offenders.
        - Don't put me or you in this scenario
- If you gain significant support / help from another student
    - Fully disclose the support / help you had in a Readme.txt file submitted with your assignments.
        - Disclosing the help, is ***not permission*** for copying the code.
        - Only there to clarify and acknowledge help you were given from a fellow student.

- Modifying any Unit Test to alter the outcome results is also an ***Academic Integrity Violation***
- If you are stuck and find yourself even tempted to plagiarize
    - Ask for help !!!!
        - Use on Piazza -> Visit during offices hours, make an appointment
        - ***Don't ever compromise your integrity!***
- Material was uniquely created for this Class.
    - You indirectly by the process of tuition, "paid" for the contents and material of this class.
        - Do not share this ***copyrighted*** material in any form
        - It is design for your personal use, while enrolled in the Class.
    - Do ***NOT*** post any content or revealing material to any external website or forum outside of this class.
        - The Class Piazza forum is provided for this service, ask questions there, not on the internet (i.e. StackOverflow and other software forums)
- After you leave this class
    - You are expressly ***FORBIDDEN*** to provide or share the content with others.
    - Academic Integrity Violations can still be applied to students who provide material support to other students even after completion of the class.
- Just follow the golden rule:
    - ***"I have neither given, nor received, nor have I tolerated others' use of unauthorized aid."***

Tentative Schedule:

This is a Topics class, the dynamics and tasks will vary based on the progress and obstacles in the class. I require that:

- Everyone participates every week on piazza
- Experiments and shares their discoveries (piazza and perforce common directory)
- Do the weekly assignments
- Need a function Milestone 1 completed on time

Weekly

20% - Weekly Code/Video Submissions (approx. 8 assignments)

Week 6

15% - Milestone 1: Prototype: Basics Audio Engine

Week 11

65% - Milestone 2: Finished Audio Engine, Demo, and Documentation

January 9, 2017	Last day to add classes to WQ2017 schedule
January 15, 2017	Last day to drop classes with no penalty
January 16, 2017	Grades of "W" assigned for WQ2017 classes dropped on or after this day
January 20, 2017	Last day to select auditor status
February 19, 2017	Last day to withdraw from WQ2017 classes

Course Policies

Changes to Syllabus

This syllabus is subject to change as necessary during the quarter. If a change occurs, it will be thoroughly addressed during class, posted under Announcements in D2L and sent via email.

Online Course Evaluations

Evaluations are a way for students to provide valuable feedback regarding their instructor and the course. Detailed feedback will enable the instructor to continuously tailor teaching methods and course content to meet the learning goals of the course and the academic needs of the students. They are a requirement of the course and are key to continue to provide you with the highest quality of teaching. The evaluations are anonymous; the instructor and administration do not track who entered what responses. A program is used to check if the student completed the evaluations, but the evaluation is completely separate from the student's identity. Since 100% participation is our goal, students are sent periodic reminders over three weeks. Students do not receive reminders once they complete the evaluation. Students complete the evaluation online in CampusConnect.

Academic Integrity and Plagiarism

This course will be subject to the university's academic integrity policy. More information can be found at http://academicintegrity.depaul.edu/. If you have any questions be sure to consult with your professor.

Academic Policies

All students are required to manage their class schedules each term in accordance with the deadlines for enrolling and withdrawing as indicated in the University Academic Calendar. Information on enrollment, withdrawal, grading and incompletes can be found at: cdm.depaul.edu/enrollment.

Students with Disabilities

Students who feel they may need an accommodation based on the impact of a disability should contact the instructor privately to discuss their specific needs. All discussions will remain confidential.
To ensure that you receive the most appropriate accommodation based on your needs, contact the instructor as early as possible in the quarter (preferably within the first week of class), and make sure that you have contacted the Center for Students with Disabilities (CSD) at: csd@depaul.edu.

Lewis Center 1420, 25 East Jackson Blvd.
Phone number: (312)362-8002
Fax: (312)362-6544
TTY: (773)325.7296

Retroactive withdrawal

This policy exists to assist students for whom extenuating circumstances prevented them from meeting the withdrawal deadline. During their college career students may be allowed one medical/personal administrative withdrawal and one college office administrative withdrawal, each for one or more courses in a single term. Repeated requests will not be considered. Submitting an appeal for retroactive withdrawal does not guarantee approval. Information on enrollment, withdrawal, grading and incompletes can be found at: http://www.cdm.depaul.edu/Enrollment-Policies.aspx