

Sprint2 – XAudio2 Voice

Student Information

Integrity Policy: All university integrity and class syllabus policies have been followed. I have neither given, nor received, nor have I tolerated others' use of unauthorized aid.

I understand and followed these policies: Yes No

Name:

Date:

Submission Details

Final **Changelist** number:

Verified build: Yes No

Required Configurations:

YouTubeLink:

Discussion (What did you learn):

YouTube Process

- Record the YouTube demo
 - You need to record in stereo with commentary
 - 2 channel with both computer (desktop) and microphone recording
 - Suggestion: **OBS** screen capture
- Record the desktop (enough to show your directory and the visual studio and output)
 - Show your directory in recording
 - Launch the visual studio (double click solution)
 - Show off relevant parts of the code with commentary
 - Launch and demo the Sprint
 - Play the demo and add your commentary in real-time
 - Watch your video
 - Verify that video clear and can you hear the commentary with audio in stereo?
- Note: Weekly Sprints cannot be longer than 5:00 mins
 - If you go over... do it again
- Publish your YouTube recording
 - Make sure it is accessible without any login or permission to play
 - It can be private but not restrictive to play by anyone with the link
 - If unplayable as-is... Grade 0
- Submit your code to perform to the appropriate Sprint directory
 - Verify it

Pdf form (this document)

- *Submit this PDF to perform*
 - *Fill in form*
 - *Name, changelist, etc...*
 - *Submit back to perform*
 - *Check it out*
 - *Submit it back to perform to the same location*

Verify Builds

- Follow the Piazza procedure on submission
 - Verify your submission compiles and works at the changelist number.

- Verify that only MINIMUM files are submitted
 - No – Generated files
 - *.pdb, *.suo, *.sdf, *.user, *.obj, *.exe, *.log, *.pdb, *.db, *.user
 - Anything that is generated by the compiler should not be included
 - No – Generated directories
 - /Debug, /Release, /Log, /ipch, /.vs
- Typical files project files that are required
 - *.sln, *.cpp, *.h
 - *.vcxproj, *.vcxproj.filters, CleanMe.bat

Standard Rules

Submit multiple times to Perforce

- Submit your work as you go to perforce several times (at least 5)
 - As soon as you get something working, submit to perforce
 - Have reasonable check-in comments
 - Points will be deducted if minimum is not reached

Write all programs in cross-platform C++

- Optimize for execution speed and robustness
- Working code doesn't mean full credit

Submission Report

- Fill out the submission Report
 - No report, no grade

Code and project needs to compile and run

- Make sure that your program compiles and runs
 - Warning level ALL ...
 - NO Warnings or ERRORS
 - Your code should be squeaky clean.
 - Code needs to work "as-is".
 - No modifications to files or deleting files necessary to compile or run.
 - All your code must compile from perforce with no modifications.
 - Otherwise it's a 0, no exceptions

Project needs to run to completion

- If it crashes for any reason...
 - It will not be graded and you get a 0

No Containers

- NO STL allowed {Vector, Lists, Sets, etc...}
 - No automatic containers or arrays
 - You need to do this the old fashion way - **YOU EARNED IT**

Leave Project Settings

- Do NOT change the project or warning level
 - Any changing of level or suppression of warnings is an integrity issue

Simple C++

- No modern C++
 - No Lambdas, Autos, templates, etc...
 - No Boost
- NO Streams
 - Used fopen, fread, fwrite...
- No code in MACROS
 - Code needs to be in cpp files to see and debug it easy
- **Exception:**
 - implicit problem needs templates

Leaking Memory

- If the program leaks memory
 - There is a deduction of 20% of grade
- If a class creates an object using new/malloc
 - It is responsible for its deletion
- Any **MEMORY** dynamically allocated that isn't freed up is **LEAKING**
 - Leaking is **HORRIBLE**, so you lose points

No Debug code or files disabled

- Make sure the program is returned to the original state
 - If you added debug code, please return to original state
- If you disabled file, you need to re-enable the files
 - All files must be active to get credit.
 - Better to lose points for unit tests than to disable and lose all points

~~No Adding files to this project~~

- ~~• This project will work "as is" do not add files...~~
- ~~• Grading system will overwrite project settings and will ignore any student's added files and will returned program to the original state~~

UnitTestFixture file (if provided) needs to be set by user

- Grading will be on the UnitTestFixture settings
 - Please explicitly set which tests you want graded... no regrading if set incorrectly

Due Dates

- See Piazza for due date and time
- Submit program performance in your student directory assignment supplied.
- Fill out your this **Submission Report** and **Sprint** to performance
 - **ONLY** use Adobe Reader to fill out form, all others will be rejected.
 - Fill out the form and discussion for full credit.

Goals

- Learn
 - XAudio2 API research
 - Learn how to use voices
 - Start learning and developing for this API

Assignments

1. XAudio2 voices

- Create a demo that explores Voices
 - NOTE
 - i. Only for 2 speaker STEREO
 - ii. Demo must be implemented inside the GameEngine project
 - Play voices (samples)
 - i. Loading
 - ii. Reading from a file
 - iii. Associating buffers
 - Repeated playback of a single voice
 - i. Need to play one voice several times with a small delay between samples
 - Example:
 - a. Gun sound effect lasting 2 sec
 - b. Firing off 5 sounds every 0.5 seconds so you hear the layering
 - Be able to change the voices attributes
 - i. Volume, pan, pitch
 - Types of playback
 - i. Looping
 - Basic controls
 - i. Start, Stop, Pause
 - ii. Expand and explore more
 - Research the API

- **Try and experiment with threads**
 - See if you can launch the audio system on a separate thread
 - Many things to explore and experiment with threads
- Discuss on YouTube Demo
 - Build and explain the code
 - Run the demo
 - Make sure you record in stereo

2. Read the first two chapters on Threads

- Currency in Action by Anthony Williams
- It's on DePaul Ebooks
- <https://proquestcombo-safaribooksonline-com.ezproxy.depaul.edu/book/programming/cplusplus/9781617294693>

3. Create Audio samples for this Sprint

- Use Audacity or something similar sample 48 kHz, wav format
- You can use the ones from class or find your own audio samples

4. Deliverables

- Stand-alone C++ demo
 - Create a demo to show off the **ALL** of the above features
 - i. Prove that you understand inside and out the features of XAudio2 Voices.
 - Use audio samples that allow you to demonstrate the above features easily
- Visual Studio 2019 Enterprise Edition
 - C++ WARNING LEVELL ALL
 - Minimum code, no temporaries or generated items
 - Needs to be able to compile and run "as-is" without checking out from perforce or changing the attributes of the files
- For some people – the demo is hardest part of this exercise

Validation

Simple checklist to make sure that everything is submitted correctly

- Submitted project to perforce correctly
 - Is the project compiling and running without any errors or warnings?
 - Is the submission report filled in and submitted to perforce?
 - Follow the verification process for perforce
 - Is all the code there and compiles "as-is"?
 - No extra files
 - Is the project leaking memory?
- Submitted the YouTube link to perforce?

Hints

Most assignments will have hints in a section like this.

- Dig into the material read the online blogs...
 - Lots and lots of information
- You can discuss the tools and drivers on Piazza
 - Share
- Use the Piazza FORUMs
 - Read, explore, ask questions