Sprint1 – Audio Research

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: Rachel Micolichek

Date: 04/09/22

Submission Details

Final ***Changelist*** number: 141523

Verified build: (•) Yes ( ) No

Required Configurations: x86 - DEBUG and Release

YouTubeLink: https://youtu.be/-0FQBW9B-TU

Discussion (What did you learn):

The reading for this assignment made me realize how important audio is in game play. A lot of games I play, I don't use audio. For example playing Pokemon on a DS, I have it muted. When playing on the PlayStation, I tend to use the subtitles more than listening to the audio. After doing the reading, I played a game on the Play Station and paid closer attention to what the audio was doing. For example, my character was climbing up a mountain and an NPC was talking to her while they climbed. Because of how the audio was configured, it sounded like we were climbing as opposed to just sitting next to each other talking. I also saw that when I walked through town if someone was calling out for a side quest, I could tell the direction based on where the sound was coming from. I could also tell if I was close or far from them based on the volume.

I used Audacity as my audio conversion tool. I use it for work, so I'm pretty familiar with it. I used it to check that the wav file I found online was actually stereo. Then, I used the conversion to make sure it was 32-bit and had a sample rate of 48K.

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 that 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 perforce to the appropriate Sprint directory
  - Verify it

Pdf form (this document)

- *Submit this PDF to perforce*
  - *Fill in form*
    - *Name, changlelist, etc…*
  - *Submit back to perforce*
    - *Check it out*
    - *Submit it back to perforce 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~~

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

- Grading will be on the UnitTestConfiguration 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 perforce in your student directory assignment supplied.
- Fill out your this ***Submission Report*** and ***Sprint*** to perforce
  - ***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 Research and Demo
    - Create a XAudio2 project
    - Play a sound or several sounds
    - Explore the API and reference material

Assignments

1. ***Research XAudio2***
   - Understanding the XAudio2 API
     - Create a program
       i. We need a low-level driver to build our engine on top of..
       ii. Documentation is online
         - Google "XAudio2"
         - Start here
           a. https://docs.microsoft.com/en-us/windows/desktop/xaudio2/xaudio2-apis-portal
       iii. Get the demo working from the online sample
     - Play Sound A
       i. Get a basic voice demo working
         - Add it to Sprint1 project
       ii. Use ***LRMonoPhase4.wav*** in the demo
       iii. Comment your CODE and learn every function
       iv. Start to refactor to cleaner code
     - Play Sound B - a stereo song you like
       i. It doesn't matter what it is…
       ii. Need 30-60 seconds length (trim it to length)
         - Stereo, 32-bit, Wav, sample rate 48K
         - Use your audio tool (Audacity – hint)

- YouTube Demo
  - Build and explain the code
  - Run the demo
    i. Play Sound A
    ii. Play Sound B
  - Make sure you record in stereo
    i. If its not in Stereo on YouTube you 0 points

2. ***Read and Study Audio basics***
   - Read **Game Engine Architecture 3rd Edition** Chapter 14 AUDIO
     - Link: DePaul ebooks safari
   - Understand the vocabulary and terms
     - Make sure you understand the basics
     - Bring questions to class

3. ***Find conversion tools***
   - We are going to need Audio preview, editing and conversion tools
   - Requirements
     - Can preview an audio sample (play)
     - Can recompress (make louder through dynamic range compression)
     - Cut and edit (reduce in length, volume, sometime pitch)
     - Convert to RAW formats (PCM, ADPCM, WAV) formats
       - Must match low-level driver format
     - Windows tool
     - Free
   - You must download and try the tool
     - Suggestion ***Audacity***
   - In the discussion area… mention which tool you are using for editing

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