# 24-780 Individual Mini-Project

Write a demo program. The demo must run minimum 60 seconds no longer than 150 seconds. Demo programs will be screened on November 8th. Since this year we have 85 students in total, all demos cannot be screened in one class. I am trying to get a room for screenings for evening time. Time and place will be announced soon.

A demo is a non-interactive presentation of audio and visual programming. Show off your programming skill and creativity.

Due to scheduling, the deadline for the individual project will be sometime into the time for the final group project, but it is highly recommended to finish it before the group project begins.

## Main Purpose

* Write more lines of code! There is no shortcut to learning programming. Only more lines of code make you a better programmer.
* Practice defining a solution to an open theme.
* Practice writing a program in longer term. You have one month to design and implement your demo.

## Important Dates

October 31st Individual Project Due

November 8th Screening (There will be a prize!)

## Requirements of your demo

1. Needs to be a demo (non-interactive demonstration of audio and visual). You need to use audio! (WAV or MML)
2. Must be copyright free or copyrighted to you. Do not use a material copyrighted to someone else. Create your own images, sound-effects, etc., or find copyright free resources.
3. Needs to be between 60-150 seconds.
4. Needs to be cross-platform. The source code must pass the compiler server.
5. Written in C/C++ and use FsSimpleWindow and YsSimpleSound libraries. If you have a reason to use a different framework/toolkit, discuss with the instructor.
6. Can include some static-image time, but must include 2D or 3D animation.
7. Doesn’t have to have a continuous background music, but must have sound effects.
8. Must NOT include a violent or inappropriate contents. (Use your common sense)
9. Must be contained in one source file (due to grading script), called demo.cpp.
10. Must not require more than 1024x768 resolution.
11. Flat directory structure. All the data files must be directly in the directory where the program can access by FsChangeToProgramDir() function. (No sub-directories in the Zip file. All files including demo.cpp will be copied to the resource directory, the EXE directory in Windows and Linux, or Resources directory in macos)
12. Must stop when the user presses ESC.
13. Code needs to be written fresh by yourself. Taking previous demos and making minor changes or only changing data is not accepted.

It is ok to use Problem Set 05 code as a basis for your demo. Or, you can start from scratch. If you want to deviate from some of the above restrictions, discuss with the instructor.

## NOT ACCEPTABLE!

**The demo code needs to be written by you. An existing code written by someone else with minimum modification is not explicitly acceptable even if you make major changes in the data. (Made clear since 2023)**

## What you need to submit

You need to submit a zip file that contains:

* demo.cpp <- Your main program.
* Data files.
* At least one screenshot of your demo in either .PNG or .JPG format.

The directory structure must be flat, which means no sub-directories. All files in the .zip file (including demo.cpp) will be copied to the resources directory (see lecture slides). Also do not access your file with full path name. Use FsChangeToProgramDir(); upfront, and then access the file without specifying directory. (Do not open a file with something like “/Users/myname/music.wav”)

## What’s demo and what’s not demo?

A demo is non-interactive. Any program that demonstrates audio and visual programing with no user interaction after the program starts can be considered a demo.

A computer game is NOT a demo because it requires a user interaction. However, if you set up your game program so that computers play against each other without user intervention, it is considered a demo.

## Where is this demo thing coming from?

See <https://en.wikipedia.org/wiki/Demoscene>

Demo is coming from a computer subculture started by the people who were cracking copy protection of computer games. They often added their own opening demo to claim credit. Demos later became its own purpose, and there started the demo culture. It is a very interesting part of the history of computer programming.

Carnegie Mellon Computer Club (<http://www.cmucc.org/>) hosts a demo party called Demosplash (<http://www.demosplash.org/> ) It is a very nice learning opportunity. There are a lot of things you can learn from demos and historic computers. I highly encourage you to visit the event. What’s better? Registration is free for CMU affiliates.

## Eigen Vectrex

Written by the CMU Computer Club demo team in 2019 for an 8-bit computer called Vectrex which was released in 1982.

<https://www.youtube.com/watch?v=1SokuBvyNWo>