ECE 4273 Spring 2024 Final Project: Retro Video Gaming

Let us journey back to the early 1980's when 280x192 was considered "high resolution" and most microprocessors ran at only 1 MHz or less. With less to look at, video games required a bit more imagination and much better gameplay.

I want to give you some flexibility in this project, but I also want to try to keep everyone's project at a similar level of difficulty. From the following tables, select features that total up to at least 4 points (5 points for a 3 person teams).

Game type (pick one)

Points	Implementation feature
0.5	Non-animated turn-based game (everything stops when waiting on the player)
1	Animated real-time game (objects continuously in motion)

Display (pick one)

Points	Implementation feature
0	Use the MCUXpresso console window for output
0	Use LEDs directly attached to the microcontroller module
0.5	Use a UART to connect with the PC for output
0.5	Use LEDs scanned in a multiplexed fashion (at least 3 rows and columns)
0.5	Use a character (HD44780 style, non-graphical) LCD for output
1	Use a character (HD44780 style, non-graphical) LCD for output, with custom characters to implement pseudo graphics or sprites
2	Use a graphical LCD for output
2	Use the oscilloscope in X-Y mode as a vector graphics display
3	Use a TV or computer monitor for output (connected directly to your circuit, no PC)

Input (pick one or more)

Points	Implementation feature
0	Use the MCUXpresso console window for input
0	Use switches directly attached to the microcontroller module
0.5	Use a UART to connect with the PC for input
0.5	Use a keypad for input (must use at least 5 keys)
0.5	Use a joystick with an analog interface
0.5	Use a game controller with a serial interface
0.5	Build your own game controller with at least 5 switches that uses a serial interface
0.5	Build your own game controller with a joystick that uses a serial interface

Sound (pick one or more)

Points	Implementation feature
0.5	Use a linear-feedback shift-register (LFSR) to generate a noise sound effect
0.5	Use PWM or timers to generate square wave beep sound effect
1	Use D-to-A to generate a sine wave based sound effect
1	Play a song with a least 8 notes
1	Play the song in the background during play (i.e. using interrupts)

Notes:

- The game must be non-trivial in the instructor's opinion
- You can use the equipment at your lab station, parts from previous lab assignments, and/or other components that you acquire (subject to the next note).
- Remember, this project should show your mastery of digital design; using a pre-built module that substantially implements a high level function will not earn you any credit.
- The lab does have some graphical LCDs, speakers, audio amplifier chips, and piezo transducers that you are welcome to borrow .
- Please be considerate of everyone in the lab by limiting the duration and volume levels of any generated audio. In many cases you can test the generated waveforms using the oscilloscope instead of a speaker/transducer.

Example projects (just so you can see how the points total up):

- Tic-tac-toe: Turn based game with the board drawn using ASCII characters on PC video UART. User inputs a number selecting a particular square on PC and transferred via UART. Plays a victory song if the player wins. Total: 0.5+0.5+0.5+1 = 2.5 (this is less than 4, so one would receive proportionally less credit)
- Freeway crossing: Player attempts to cross a freeway with multiple lanes of continuously moving traffic. Using the non-graphical LCD with custom car/truck characters for output and original Nintendo controller for input. Generate noise effect for crashes an play a song for each successful crossing. Total: 1+1+0.5+0.5+1 = 4
- Space rocks: Player attempts to pulverize large flying space rocks into smaller ones without crashing their spaceship. Using oscilloscope vector display and switches connected directly to microcontroller. Ship's laser generates "pew-pew" sound effect and destroyed rocks make an explosion sound. Total: 1+2+0+0.5+0.5 = 4
- Optimally packed falling shapes: Player attempts to optimally fill a space with continuously falling shapes. Using a graphical LCD for output and a homemade controller with a serial interface for input. Continuously plays music for maximum player enjoyment/annoyance. Total: 1+2+0.5+0.5+1+1 = 6 (any excess points do not earn extra credit)

Grading

- 10% Preliminary hardware/software design report (due April 12)
  - A general description of the project
  - A complete schematic, with enough detail so that someone else could build the exact circuit you want without ambiguity. Since this is a preliminary report, the schematic does not necessarily need to work, but you should have a reasonable expectation that it would.
  - For discrete components like resistors and capacitors, show how their values were chosen.
  - A high level description of how the software will implement the project functions
- 40% Final report (due April 26, with a penalty-free extension to May 3 available)
  - A general description of the project (updated, if needed)
  - Directions for use (think of it as the user manual for your project)
  - Any design analysis of component values (updated, if needed)
  - The final complete schematic
  - The final complete software
  - Any other supporting documentation to show that the project meets the objectives/specifications (for example, oscilloscope snapshots)
- 10% Project poster (due by May 10)
  - Names of the team members and the project title
  - Summary of the project, its features/specifications
  - A simple overview of how the project words (such as a block diagram of the hardware)
  - Posters should be approximately 2 by 3 feet in size on posterboard or foamboard
- 40% Demonstration (April 30 – May 3 during regular lab time, or by appointment)
  - Each team member must participate in the project demonstration and discuss some digital aspect of the project that they contributed to for full credit
  - The 4 (or 5 for large teams) points of the implementation features contributes to this portion of the grade