PG5500 - Embedded Systems

Assignment 3 - Eivind Vegsundvåg

Introduction

The solutions is a minimal(read: bug free) implementation of Lights Out game(https://en.wikipedia.org/wiki/Lights_Out_(game)).
A video introducing the playable game can be seen here:
https://www.youtube.com/watch?v=AtXkD9Ngn9s

Limitations

In the video, around the 1:10 mark, I note the following(paraphrased from Norwegian): “Which I did not make on my own”. This refers to the algorithm for solving the puzzle with the fewest possible steps. ALL code for the game itself is implemented blindly in a clean room. The algorithm referred to can be found at
http://www.ueda.info.waseda.ac.jp/~n-kato/lightsout/

Prerequisites

Adafruit GFX Library(https://github.com/adafruit/Adafruit-GFX-Library)
Adafruit ST7735 Library(https://github.com/adafruit/Adafruit-ST7735-Library)
Adafruit Fritzing Library(https://github.com/adafruit/Fritzing-Library)

Wiring

The wiring is available as a Fritzing sketch. Note that many components were not readily available as Fritzing components, and that the pin wirings therefore seem utterly wrong in the sketch due to similar, but different components being used.

Minimalism

This game is quite minimal. However, I still consider it fully complete. The minimalistic design has allowed me to leave quite a large available memory pool(33% available). Therefore, I expect this game to be fully stable.

Components

ST7735 TFT Display

The ST7735 display is controlled by the LightsOutGame and minimally by the LightsOutHandler classes. When a game starts, it initially draws the entire screen. For updates on text, it redraws the current text in the background colour, then draws the new text in the primary colour.
For the game board, it simply redraws the affected areas over the old states. Given the few pixel counts, this solution works very well.(Tests showed that when the Arduino was told to simply handle unlimited click events as fast as it could, it could handle about 20 redraws of five lights per second - barely noticeable for the human eye)

Analog Stick

The control stick is controlled by the InputHandler. This service is capable of reading the functions of an analog stick, mapping movement on the X and Y axes to movement on the grid. Clicks are also supported. The InputHandler resolves these actions to an InputType enum. These are passed from the LightsOutHandler to the LightsOutGame except during startup phases.

Simple Speaker

The speaker only plays sounds as directed by the LightsOutGame service. It can play the following three sound types:

- Click
- Move
- Game over/Victory

Unsolved issues

None. However, it needs some very minor adaptation to D-Pad and extra button, in addition to a smaller Arduino controller and volume knob before it can be embedded into a chassis. I fully intend to gift this setup to one or more people once these steps are completed.