CSE 379
Lab #6
Spring 2021

Objective
In this lab, you will incorporate all of the concepts you learned in Labs #1 through #5 to design a game.

Description
Write an ARM assembly language program that implements the game *Flow Free*®, as described below.

The Motivation
This game is motivated by the game *Flow Free*®. If you are not familiar with the game, there is an online version that you may play to learn more about the game. You will find the link on the course website. This game will closely represent the gameplay and layout of this project.
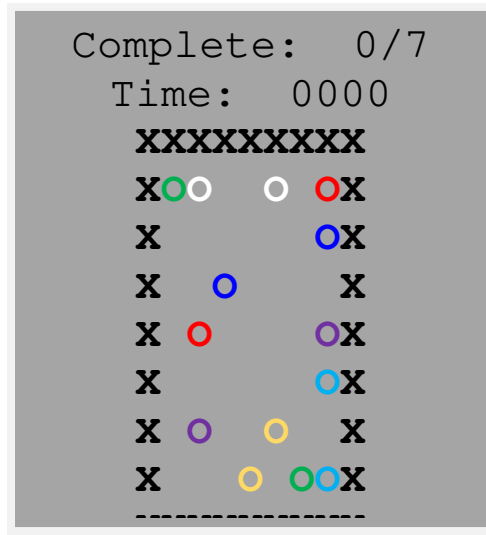
The Story
The objective of the game is to connect a pair of colored dots by drawing a line between them. The line will be of the same color of the dots being connected. What makes this challenging is that there are seven different colors that you will be working with, and lines of different colors may NOT cross one another. Furthermore, the objective is to fill the entire game board, so every space must either have a dot or a line! Don't delay! The running time of the game is kept (displayed at the top), and the quickest time to complete the board wins!
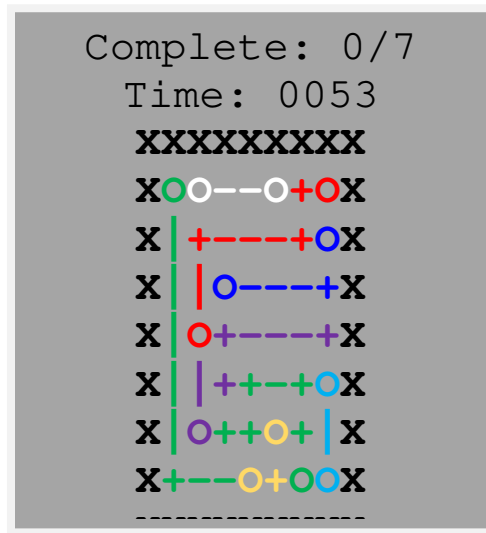
The Gameplay
When the game starts, the board should be displayed with seven staring/ending points, one pair for each of the seven colors that makeup the game. You have been provided with 16 game boards. When a game starts, a board must be chosen at random. The user will use the keys *w*, *a*, *s*, and *d* to move up, left, right, and down respectively. A color is selected by moving to a starting/ending point and pressing the space bar. The current position must be highlighted. The terminal's build in cursor can be used to indicate the current position.

The Board

Your board in *PuTTy* will look something like the board shown below. The first board is the initial board. The positions of your starting and ending points will vary.

```
        Complete:   0/7
          Time:   0000
          XXXXXXXXX
          XOO   O  OX
          X           OX
          X    O       X
          X  O        OX
          X           OX
          X  O    O    X
          X      O  OOX
          ------------------
```

This board shows the final board after the game is complete. Notice how every spot in the 7x7 grid is occupied by a start/stop point or a line.

```
        Complete: 0/7
          Time: 0053
          XXXXXXXXX
          XOO--O+OX
          X|+---+OX
          X||O---+X
          X|O+---+X
          X||++-+OX
          X|O++O+|X
          X+--O+OOX
          ------------------
```

Sixteen (16) game boards are provided on the course website. You may incorporate additional boards as well, but make sure that they have a solution before you implement them. The selection of which game board to display when the game starts must be random.

The Characters

The legend below shows the meaning of each character.

x	Game Board Boundary
–	Horizontal Line
|	Vertical Line
+	Corner
o	Start & End Points

The Colors

The chart below shows the colors for each pair of start/end points. ANSI escape sequences should be used in incorporate colors. The color of the background and walls does not matter as long as they are both uniquely distinguishable.

Green
Red
Blue
Yellow
Cyan
Magenta
White

Game Speed & Movement
The user should be able to draw lines at any rate. So, in other words, the user can navigate the board and draw lines as fast as they can press keys on the keyboard.

Hardware Utilized
- Serial Port (UART)
  - Used to accept the user's input and display the game.
- RGB LED
  - The RGB LED should be illuminated during game play to indicate the color of the line currently being drawn.
- Momentary Push Button (Switch 1 on the Tiva, labelled SW1)
  - The momentary push button (SW1) will be used as a pause button. When pressed, your game must stop, not allowing any movement and suspending the elapsed time shown at the stop of the board. To continue play, the switch (SW1) should be pressed again. When paused, the board should be hidden and the player should be notified that the game is paused. The user should be given the option to restart the current puzzle or start a different puzzle.
- Timer
  - The timer is used to show the elapsed time.

Game Requirements List

- At least 16 boards must be stored in memory.
- The board to be played must be selected at random.
- WASD will be used to move up, down, left, and right.
- The player's current position on the board must be indicated.
- If space bar is pressed when over on 'O' you will begin drawing a line in that color. Pressing space bar again will stop drawing.
- When drawing a line, the RGB LED will be the color of the line being drawn. While not drawing a line, the RGB LED must be off.
- SW1 will be used to pause and resume the game.
- When paused, the board is hidden and the player is notified that the game is paused.
- While pause the game board will be hidden and the player will be given the option to either restart a new puzzle, restart the current puzzle, or resume.
- The time since the game was started should be displayed at the top of the board.
- The number of lines currently connecting two end points should be displayed at the top of the board.
- While drawing a line, if the player crosses over an existing line, the existing line will be deleted to make room for the new line. Note the entire line that was crossed over is deleted.
- You are not allowed to cross over the color currently being drawn. If the player tries to cross over the line currently being draw, it should not allow them to move.
- If an 'O' that already has a line is selected, the original line will be erased.

Grading

The project will count twice, carrying the weight of two (2) labs.

Prelab Writeup

You must show your TA prelab write-up describing how you will solve the problem no later than Thursday, April 2. This will be part of your Lab #6 grade. Failure to submit this on time will result in a grade of zero (0) on this component of the lab.

Partners

You will work with a partner in this lab. Your partner *MUST* be the same partner you worked with on lab #5.

Documentation

Your program must be clearly commented, and documentation must also be provided. The documentation must follow the guidelines covered in lecture (found on the *Lectures* webpage of the course website). Your comments should describe what *each* section of your program does.

Demonstrations & Submissions

You will demonstrate your game to your TA via Zoom in one of your regularly scheduled lab sessions during the week May 4, 2021. Your source code and documentation must be submitted online using the submit command (*submit_cse379 filelist*) on *timberlake.cse.buffalo.edu* no later than 11:59 PM on Sunday, May 2, 2021. Late submissions will NOT be accepted.