# CS 112 Project Assignment: Sliders

Instructor: Dan Fleck

## Overview

In this project you will use Python to implement the number sliders game. This game can be seen at:
http://www.classbrain.com/artgames/publish/sliders_game.shtml

You will only be implementing the 8-square game.

The objective of the game is to get all the numbers in order and have the blank tile at the bottom right. To do this, first the tiles are scrambled into a random location. Then, the user must choose a tile to move into the blank space. The process repeats until all tiles are in the correct order.

## Assignment

For CS112 you will need to implement the following parts:

**Main Menu**
The main menu should be displayed when the program starts:

```
Main Menu
1. Show Instructions
2. Play New Game
3. Resume Previous Game
4. Show instant replay of the last game

Enter selection:
```

Option 1 should display instructions for the game. That text is given in the instructions.txt file. (You can copy the file into your program.)

Option 2 should begin a new game by starting with a solved board (all tiles in order) and then randomly generating 1000 moves to effectively “scramble” the board.

Option 3 should resume a paused game if there is one. If no game was paused, the option should tell the user “No paused game present” and return to the main menu.

Option 4 should watch an instant replay of the most recent game played during this session. It can be either a game that was quit or a game that was paused, whichever was most recent.

**Game Menu**
When the user is playing a game, the Game board and menu should be displayed. You should display the board as follows:

```
| 1 | 3 | 5 |
| 2 | 7 | 4 |
| _ | 8 | 6 |

Select a tile to move or (p)ause, (q)uit, (s)cramble :
```

In the above game board, only numbers 2 and 8 are valid, all others should print an error message and then the Game Menu again. If the user chooses ‘p’ or ‘q’ the Main Menu should be displayed. If the user chooses ‘s’ the board should be scrambles from it’s current state. Essentially, you must generate 1000 random moves and then display the board again.

After each user move, you must determine if the puzzle is solved. If it is, display a message to the user that the game is completed, and return to the main menu.

Challenges:

- You will need to determine how to store the state of the game. How will you represent the location of each number?

- You will need to store each move the user makes so you can replay the game

- You will need to generate random moves using a random number generator. Check out the Python modules to find a random number generator.

- While you scramble the numbers, make sure you aren’t checking for a solution. You don’t want to scramble and at move 350 tell the user the puzzle has been solved!

- You will need to clear the screen and redisplay the game board after every move so it looks like the board changed, not that it was reprinted.

- When replaying a game, you will need to add in some waiting time (delays) so people can watch the game progress. (500 milliseconds between moves is probably adequate). Hint: Try the time module.

# What to turn in:

1. Source code
2. Answers to the following questions:
    1. What did you like/dislike about this project?
    2. What was the most difficult part of this project?
    3. How many hours of work did you do on this project?
    4. Did you work with anyone else on the high level issues?
        If so, who? (This is okay and will help you learn a lot)