EC 327

December 11, 2015

Project Documentation

Daniel Andronov

Emily Ubik

Alex Czaja

Eric Nurkatov

Evan Vincent

**Development Team**

Project Lead – Daniel Andronov

GUI Designer – Evan Vincent

Interface Designer – Eric Nurkatov

Processing Designer – Emily Ubik

Documentation Manager – Alex Czaja

**Overview of the App**

The app has three main parts, the GUI, the interface, and the back end. The GUI displays the board with the number tiles on it as well as the game time and the moves made. There is also a button that scrambles the board so the player can begin playing. The interface connects the GUI and the back-end together so that pressing on the tiles will move them, the scramble button works as it should, and information about the state of the board is passed along so it can be displayed.

The backend is all the code that makes the game run. The main cpp file uses two classes, one that describes the board and the other that contains it. The board class, called Grid, contains the solution to the puzzle, the current state of the board, and is able to check if the puzzle has been solved, scramble the puzzle, display the board, and move tiles around. The grid or board is contained in the Model class that contains an instance of the board and accepts input to manipulate it. The main cpp file creates an instance of the model and passes input into it.

**Description of the back-end**

The back-end is made up of two classes and a main cpp file.

**The Grid Class**

|  |
| --- |
| Grid |
| - size: int  - sol: int array  - contents: int array |
| + Grid( int )  + scramble(): void  + check(): boolean  + swap( int , int ): void  + display(): void |

The Grid class the game board. It contains the tiles and can move them. Upon construction, the size is initialized to four. The solution array, sol, and the contents array, contents, are created in a for loop and both filled to the solved state. The contents array is then scrambled using the scramble method.

The scramble method

The scramble method uses a for loop to iterate through the array and randomly assign new indices for each array element. It uses the C Standard Library’s rand function to generate the random indices.

The check method

The check method simply iterates through both the solution and contents array to check each element. If and corresponding elements are not equal, then the function will return an false. Otherwise, true.

The swap method

The swap method will move a tile around on the board. The board has tiles numbered zero through fifteen, the zero tile representing the empty space on a physical puzzle.

The function takes in two integer arguments, corresponding to the row and column index of the tile selected to move. The swap function scans the array to find the position of the the zero or “empty” tile and records it’s indices. Then the two array elements are swapped using a temporary holder variable.

The display method

The display method simply prints

|  |
| --- |
| Model |
| + time: int  + moves: int  + grid: grid |
| + input( int, int): bool  + Model( Grid\* )  + ~Model( ) |