PLEASE NOTE:

* This is a team assignment. At most 2 students may form a team (same section) and work together.
* One submission per team is sufficient.
* You must demonstrate your project during class time.
* Both Members must be present to be graded during demo.

Assignment due (Softcopy):

Upload your project to eConestoga by 11:59pm Sunday 29-Oct-2017.

Demo schedule:

First available class after softcopy is due.

Before you start:

If you are not familier with 15-puzzle game, please check the following links:

1. <http://en.wikipedia.org/wiki/15_puzzle>
2. <http://migo.sixbit.org/puzzles/fifteen/>

Self Study:

1. Saving a text file and loading from a text file
2. Using OpenFileDialog and SaveFileDialog controls

Check FileApplication.zip application available in eConestoga.

Problem Specification: (15-Puzzle)

1. Create an N-Puzzle game in C#. TextBoxes will be used to specify the number of rows and columns.
2. Game will start WITHOUT the puzzle. After specifying row and column values and clicking Generate button, the puzzle will be created initialized randomly.
3. You are recommended to create a Tile class inheriting from Button class.
4. The game will show game-end message after solving the puzzle and all the tiles will be removed, ready to create a new puzzle combination.
5. The game will have a save game and a load game option (you may use menu or buttons).
6. Save game option will save the current game state to a text file.
7. Load game option will remove all the tiles of the current puzzle and let the user create a new puzzle by loading an existing game file.
8. Any time clicking Generate button will remove all tiles and create a new puzzle based on the row and column values.
9. You must use OpenFileDialog and SaveFileDialog controls to load and save. Use C:\ as the default path for save file and load file.

**Remember to incorporate the followings for all assignments (if applicable)**

* ~~Add hot keys.~~
* ~~Set tab order.~~
* ~~Add tool tip for controls. You may choose your own message for ToolTip text.~~
* ~~Set Accept and Cancel Button for Form.~~
* Add Title comment.
* Add Documentation comment.
* Add Implementation comment where (you think) necessary.
* All other standards.
* Organize the controls to have a better look.

Note: Use your own imagination for designing the user interface. The snapshot of the Form is given only to clarify the requirement. You don’t have to make it exactly the same.

Hints:

* You must understand that 50% of the randomly generated combinations are not solvable. That’s why, while testing your program you should use a valid initial combination (you may use the combination shown in the screen shot, or generate one by your own). Please add 1 game files (valid solvable combination file for 15-puzzle) in the zip file that you upload.
* When you are initializing your game with random combination, don’t worry about the solvability. Whether your game can identify end game condition will only be tested with the valid combinations (1 save game) you provide.
* While saving the combination you may save the number of rows and number of columns as well as the position of empty slot, in the file, so that you can recreate the maze.
* Test your game whether it can identify game end state.
* If you always want to generate solvable combination, there are algorithms available to test whether a random combination is solvable. You might also start with a game end combination (1 to n) and then scramble it by randomly moving the empty slot, say 50 times. (5 marks). Random combination generation is not required if you follow this approach.
* If your puzzle cannot always produce solvable combination you can get at most 95 in this assignment. You may get at most 100 if it can produce solvable combination all the time.

**Hardcopy Submission Requirements (follow the sequence as stated)**

1. Combined Coverpage and Marking sheet.
2. Assignment 3 Marking sheet. (last page of this document)
3. Run your program, and then take a snapshot (printscreen your form).
4. Printout of your code. Print using Visual Studio 2017 IDE.
5. Make sure all Programming standards are followed.
6. Make sure all Assignment standards are followed.

Note: Repeated violation of the same standard is counted.

Softcopy Submission Requirements

1. Make your solution folder a zip file (don’t forget to add 1 game files) according to the following example:

If the students’ names are Jason Bourne, for

Assignment 3, the name of the project will be JBourneAssignment3 and the zip

file will be JBourneAssignment3.zip

1. Log in to your eConestoga account.
2. Select the course PROG2370 and locate the Assignment’s dropbox
3. Upload the zip file.

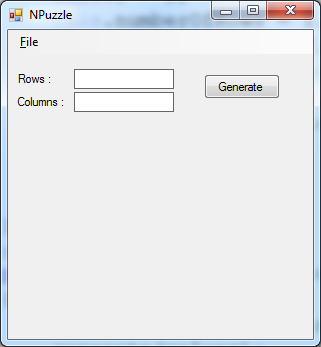


Figure 1: Game starts without a puzzle.

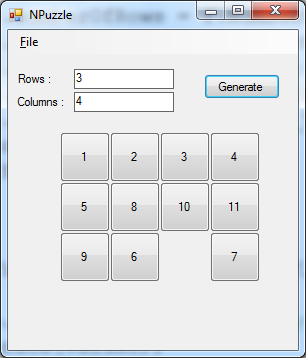


Figure 2: Puzzle is created after providing row and column values and clicking Generate button.

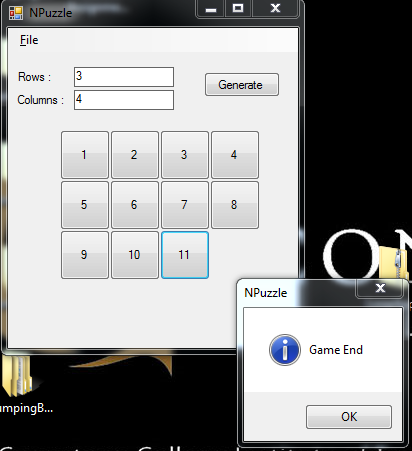


Figure 3: Game ends with a message. After clicking OK, it removes all tiles and goes to the initial state as shown in Figure 1.

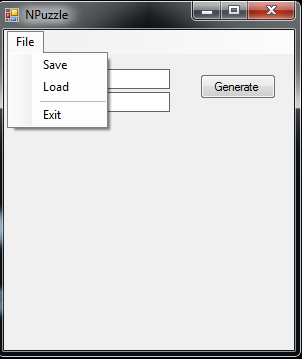


Figure 4: Game can be saved and loaded from menu (Menu can be added from Toolbox-> Menus & Toolbars-> MenuStrip) (or you may add buttons for save and load).

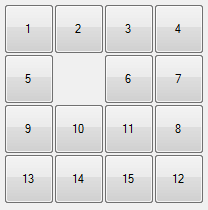


Figure 5: Solvable combination that you may use to load. It is up to you how you save it in a file. I will load this combination and play to find if your game find game end condition by showing a message.

Assignment 3 Marking Sheet

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(please fill your name)**

|  |  |
| --- | --- |
| **Spec** | **Marks** |
| (Attractive/Colorful) User Interface Design | /05 |
| Game is randomly initialized(it could be solvable or not solvable) | /10 |
| Game can be loaded from file | /15 |
| Game can be saved | /15 |
| Game shows Game End message at game end. | /15 |
| Clicking Generate any time will remove all tiles and create a new puzzle. | /05 |
| Game is customizable to an n-puzzle (r x c rows and columns combination) through TextBoxes. You will not get marks for this part if you submit a 15-puzzle (4x4 tiles). | /20 |
| Using OpenFileDialog and SaveFileDialog controls | /05 |
| 1 solvable combinations as text files (15 puzzle only) provided in the zip file. (see Figure 5). | /05 |
| Game is always solvable | /05 |
| **Total** | **\_\_\_\_\_\_\_\_\_\_\_\_/100** |

**Deduction:**

|  |  |
| --- | --- |
| Runtime errors | 15 x \_\_\_\_\_\_\_\_\_\_\_\_ = |
| Standard | 5 x \_\_\_\_\_\_\_\_\_\_\_\_ =\_\_\_\_\_\_\_\_/20 |
| Programming Standard | 1 x \_\_\_\_\_\_\_\_\_\_\_\_ =\_\_\_\_\_\_\_\_/20 |
| Late Submission | 20 x\_\_\_\_\_\_\_\_\_\_\_\_\_= |
| No Demo | 40 |
| Bugs | 3-15 (based on the nature and severity of the bug ) |
| **Total Deduction** |  |

|  |  |
| --- | --- |
| **Total Marks** |  |