You must demonstrate your program during class time. (It is an individual assignment)

# Assignment due (Submitting Softcopy to eConestoga):

|  |  |
| --- | --- |
| Section 5 | 30-Sep-2018(Sunday) 11:59PM |
| Section 6 | 30-Sep-2018(Sunday) 11:59PM |

# Introduction: Q-Puzzle game

1. Q-Puzzle game is puzzle game consisting of at most 4 colored doors and corresponding 4 colored boxes. The number of doors and boxes may vary, however for every colored box there must be a matching colored door.
2. The control pad will contain four directional buttons: Up, Down, Left, Right.
3. The player will select a box and use the control pad to move the box. The box will continue to move until it hits a wall or another box. If there is a door with the same color of the box in the direction of its movement, the box will be removed.
4. Your objective will be to remove all the boxes through their corresponding colored doors and the game ends.
5. See <https://www.gamesloon.com/free-puzzle-7/sokoban-games-269/q-puzzle-339.html> to get a better idea.

# Important notes:

1. The complete project is divided into 2 assignments:
   1. In Assignment 1 you will:
      1. Create a game level design layout
      2. Design a level using the layout
      3. Save the level as a text file
   2. In Assignment 2 you will:
      1. Load the previously saved level
      2. Play
      3. Details of Assignment 2 will be available in Assignment 2’s specification.
2. It is important to understand that you must complete Assignment 1 before starting working on Assignment 2.
3. The rest of the document will specify the requirements for Assignment 1 only.

# Problem Specification (Assignment 1):

1. Create a Control Panel Form which will be used to either Design a level or Play. In this assignment the Play button will not do anything.
2. Create a Maze Designer Form which can be invoked by the Design button.
3. The designer form will contain a toolbox at the left. Toolbox will contain Buttons/Pictureboxes for necessary tools (e.g., different types of boxes, different types of doors and wall).
4. The form will also contain necessary TextBoxes to input number of rows and the number of columns. It will also have Save feature through the menu.
5. The user will input number of rows and number of columns, then click the Generate button and a grid (of PictureBoxes) will be generated. Initially the grid will be empty. By clicking on a tool button, the user will choose a tool and click on a cell to assign that item to the cell.
6. Upon completing the level design the user will be able to save the maze to a text file.
7. After saving a MessageBox will be shown with a confirmation message.
8. For saving you must use Save dialog box to navigate to a location and to provide/choose a file name.
9. In case of invalid input(s), error message will be displayed.
10. You may check the demo video available under Assignments module of eConestoga.

**Hints:**

1. Note: these are just hints, you may achieve the save outcome in various ways.
2. Please see the attached video carefully. Your game must fulfill all the functionalities implemented in the video.
3. If you want to make your toolbox buttons with image and text, use an ImageList, add images to the list, and assign an image to each button. See the Buttons’ ImageList and ImageIndex properties.
4. The Grid is composed of PictureBoxes which are generated dynamically. Upon clicking on a cell previously selected tool’s image needs to be shown. You must load the image through resources. Using absolute/relative paths to the images will result in deduction of marks. Check the hands on program that addresses how to load an image to a PictureBox through resources.
5. For each tool you may assign an integer, say, 1 for Wall, 2 for Red Door, 3 for Blue Door etc. Assign that value to a PictureBox cell as well as the image once clicked. You may want to make a class inheriting from PictureBox class. Check the hands on programs and Powerpoint presentations addressing inheritance.
6. The save file format you will see in the demo considers the following pattern (Check file handling hands on program):
   1. Number of rows
   2. Number of columns
   3. For each cell
      1. Cell’s row
      2. Cell’s column
      3. Cell’s content (1 for Wall, 2 for Red Door etc.)

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

* Add Title/Header comment.
* Add Documentation comment by using /\*\* … \*/ for all methods.
* Add Implementation comment where (you think) necessary.

Note: The user interface should be as close as the demo provided. All functionalities must be implemented as well. You may use your own images for the tools.

# Softcopy Submission Requirements

1. Name your project according to the following example:

Example: if the student’s name is Jason Bourne, for

Assignment 1, the name of the project will be JBourneAssignment1)

1. Take a screenshot of a level: JBourneAssignment1Screenshot.jpg.
2. Save the level to a text file: JBourneAssignment1Level1.txt. You may use your own extension instead of .txt.
3. Make a zip file as JBourneAssignment1.zip. This zip file will contain:
   1. The complete project made in C#
   2. Screenshot of a level
   3. Saved level (txt file)
4. Log in to your eConestoga account.
5. Select the course PROG2370 and locate the Assignment’s dropbox
6. Upload the zip file.

# NO Hardcopy Submission accepted

Make sure all Programming standards are followed. Please read pages 12-40 of Standards.pdf file available under Resources module in eConestoga for detailed information. Make sure all Assignment standards are followed.

Note: Repeated violation of the same standard is counted.

Assignment 1 Marking Sheet

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

|  |  |
| --- | --- |
| **Spec** | **Marks** |
| User Interface Design | 10 |
| Can generate the grid | 10 |
| Clicked cells are updated based on the selected tool | 15 |
| By selecting a new tool and clicking on a cell which contains a preselected item will overwrite | 10 |
| “None” tool will empty a cell | 10 |
| Save dialog boxe used | 10 |
| Can save | 20 |
| Images are assigned through resources | 5 |
| Messagebox shown after saving | 5 |
| Messagebox for any type of error | 5 |
| Total | **\_\_\_\_\_\_\_\_\_\_\_\_/100** |

**Deduction:**

|  |  |
| --- | --- |
| Runtime errors | 15 x \_\_\_\_\_\_\_\_\_\_\_\_ = |
| Assignment Standard | 5 x \_\_\_\_\_\_\_\_\_\_\_\_ =\_\_\_\_\_\_\_\_/20 |
| Programming Standard | 1 x \_\_\_\_\_\_\_\_\_\_\_\_ =\_\_\_\_\_\_\_\_/20 |
| Late Submission | \_\_\_\_\_\_\_days = |
| **Total Deduction** |  |

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