

TEST PROJECT

Client Side A

Submitted by:

Marcelo Strehl - BR

Competition Time:

3 hours

Assessment Browser:

Mozilla Firefox





INTRODUCTION

To show off your web design skills, you have decided to create a small web application called Custom Puzzle. In Custom Puzzle, a user can submit a picture to create a puzzle from, with the number of pieces determined by the level of difficulty chosen at the beginning.

The challenge is to rebuild the original image by rotating and repositioning the pieces in the shortest time possible. There is a separate rank for each difficulty level, which is presented when the user successfully rebuilds the puzzle.

You have asked a friend to build the layout and the database of the application to accommodate the different ranks. Your task is to develop the functionality of the application based on the requirements presented to you.

DESCRIPTION OF PROJECT AND TASKS

1. Start Modal: When the user opens the application, there must be pop-up through which the user can choose the level of difficulty, submit the image, and enter their name, with the following fields:
 - 1.1. **Name**;
 - 1.2. **Difficulty Level**, with the following options:
 - 1.2.1. **Easy**: Slice the image in 4 pieces;
 - 1.2.2. **Medium**: Slice the image in 9 pieces;
 - 1.2.3. **Hard**: Slice the image in 16 pieces.
 - 1.3. **Image Area**: This is the area to drop the image file. This is also the area where the image preview is displayed once a valid (JPG format only) image is dropped.
 - 1.4. **Start Button**: Clicking button starts the challenge.
2. All Start Modal fields are required in order to start the challenge. It is not possible to start the application without correctly entering the required information. In case there are errors, friendly messages must be displayed to the user. The image file cannot be uploaded to the server.
3. After the Start Button is clicked, the Start Modal must be closed and the challenge screen must be displayed with the following elements:
 - 3.1. **Name of the user**: According to data information provided on Start Modal;
 - 3.2. **Clock** to present the time elapsed;
 - 3.3. **Pause button** to pause the puzzle.

The game will content two areas: a starting area and a destination area.
Starting area will content the mixed images pieces without overlapping.
The mixed pieces are the result of randomizing slice positions and rotation of each slice image (by unit of 90 degrees).
Destination area will content the drag and drop pieces from the starting area - A preview image with degree of opacity will be visible in the destination area where the image provided is presented in the original form (i.e. this is the goal);
 - 3.4. **Restart Button**: to restart the puzzle. Clicking in this button at any time during the time running the application shows the Start Modal again, allowing the user to start a new puzzle.
4. The starting area and the destination area have a square proportion, so the image provided must be cropped to a square that covers the areas. The center of the image in the destination area must be kept (i.e. the crop must be centered).



5. To create each piece, the application must slice the image in square pieces, based on the level of difficulty chosen on the Start Modal.
6. The square pieces must be randomly shuffled across different positions in the starting area as well as randomly rotated.
7. The clock starts upon clicking the Start Button. The clock must show the time elapsed in the format (mm:ss).
8. Clicking on the Pause Button, the time is paused and the starting area and destination area are hidden to the user. The text of the PAUSE button must be changed to "RESUME". No actions are permitted during a pause time. Click again in the RESUME button, the starting and destination areas are presented again to the user, the label of the button return no "PAUSE" and the time continue.
9. The user can select one piece of the starting area by clicking on it. When a piece is selected, it can be rotated by using keyboard arrow: left or right and it can be dragged and dropped to the destination area. When selected the piece must be elevated over the rest of the other pieces.



10. The user can rotate the selected piece clockwise using the right arrow key (→) on the keyboard. Every key press rotates the piece 90 degrees clockwise.
11. The user can rotate the selected piece counterclockwise using the left arrow key (←) on the keyboard. Every key press rotates the piece 90 degrees counterclockwise.
12. When a piece is dropped to the destination area it will be accepted if the position and the rotation are good otherwise the piece will return to the starting area with the same starting rotation and position, no selected.
13. When a dropped piece is accepted in the destination area, it will snap in that position and the opacity at that position will disappear. The piece will reveal its real color with an animation.
14. The actions of elevate a selected piece, rotate a piece need to be animated in the application.



15. The actual state of the game needs to be saved in the local storage of the browser. If users reload the page it should show the puzzle which they were doing. Position of pieces, configuration variables and the time elapsed need to be presented again.
16. The puzzle is solved when the original image is completely rebuilt in the destination area. The clock must stop at this time.
17. The result must be saved to a MySQL database on the server, with the username, difficulty level and the time elapsed for each game.
18. After saving the result, a modal saying "Congratulations! You have solved the puzzle!" must be displayed with an animation, along with the rank table for the respective level (the level of the puzzle newly completed) and a button to restart the application.
19. The Rank Table presents the three best results ordered by elapsed time for the same level. The position and the results of the current user need to be presented too. If the current result is in the top 3 results, just the 3 best results are presented. The ranking table must present position, difficult level, name and time elapsed. If exists more than one register with the same elapsed time, these users receive the same position in the Rank Table.
20. Clicking the "Restart Application" button closes the results modal and shows the Start Modal, allowing the user to start a new puzzle.
21. You must use functions, methods and objects, object oriented programming to permit the reuse of your JavaScript code.
22. Your application need to run without JavaScript errors showed in console of the browser.

INSTRUCTIONS TO THE COMPETITOR

The folder "media" contains all the files provided by your friend to develop the application. To develop the task, you can use jQuery and jQuery UI if you find necessary. You can modify the provided files if necessary to guarantee the correct functionalities of your application but your application makes use of the general layout provided. The database is provided to you in one SQL file, you need to restore/create the database to register the results and retrieve the results to show the rank table.

Save the files of your application in directory on the server called "**XX_ClientSide_A**", where XX is your country code.

INTERNET ACCESS

- <http://api.jquery.com>
- <http://api.jqueryui.com>



MARKING SCHEME SUMMARY

SECTION	CRITERION	JUDGEMENT MARKS	OBJECTIVE MARKS	TOTAL
E1	Starting Application	2.00	3.25	5.25
E2	Using Application	0.50	4.00	4.50
E3	Results of Application	0.70	4.30	5.00
Total		3.20	11.55	14.75