

Student-ID: 170024591

CS5003 - Masters Programming Projects

Due: 28 February 2018

Practical 1 - Report

1. Description of page content and basic structure

My solution for the quiz game is based on one main page that I have worked around and iteratively enhanced as I added more features. All elements that appear on the page are created in the html file to be easily able to address them and so to keep track of them more easily.

The page consists of eleven main container: header, initial page, pop-up, game over pop-up, first row, start button, question container, continue request, progress bar, answers container and the home menu.

When the page is first loaded, everything is hidden, except for the “initial page“ which asks the player to enter his or her name to join the game (see figure 1.1).

The name is then stored to address the player, the earnings (see figure 1.2, 1.3) and also the leaderboard, which will be described in section 2.4.

The “header“ with the home button, a “new game“ and a “end game“ button appear when entering the main game. The home button can only be clicked outside of ongoing games. The other two buttons are the other way around and allow the player to end the game or restart a whole new one at any time during an ongoing quiz (see figure 1.2). Just below is placed the secured amount of money (the current total win) of the player (also see figure 1.2).

Under the header comes the “first row“ which features five separate containers that show following features (see figure 4):

1. Questions played (counts up the answered questions for each quiz session)
2. Three hearts symbolising the amount of “lives“ left for a game
3. The fifty-fifty joker
4. The time-refresh joker
5. The amount of the (possible) win of the current quiz session

When a game is activate, the space beneath the first row will show the container with the question, then the animated bar with the decreasing time a player has to answer the current question and below the bar the four or two containers with the corresponding answers, depending on the type of the question - multiple or true/false (see figures 1.5 and 1.6).



Figure 1.1. Initial page, appears when first loaded.



Figure 1.2. Shows the player's name with the current total win he/she will go home with. Also shows the two buttons to either stop the game or start a new one.



Figure 1.3. The player is addressed by name, when a game has ended with earnings.



Figure 1.4. Upper part of the quiz page showing the so called “first row“.



Figure 1.5. Whole page, including the time decreasing bar, the question container and the containers with the answers.



Figure 1.6. Whole page, figuring a true/false question.

The space of the time bar and the answers is after each answered question replaced with the box asking the player if they want to continue or cash out. It is either green for when the answer was correct (see figure 1.7) or red, when the answer was incorrect (see figure 1.8).

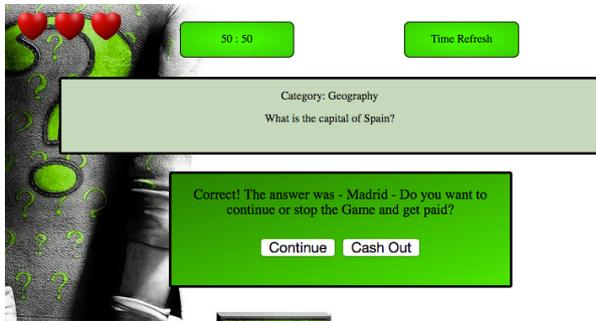


Figure 1.7. The player is asked to continue or to cash out, when the answer was correct.

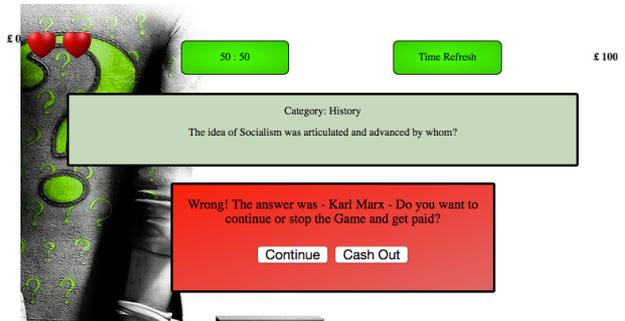


Figure 1.8. The player is asked to continue or to cash out, when the answer was incorrect.

Next, the end screen hides everything except for the headline and the total win and shows a container that adjusts accordingly, depending on how the game was ended and if the player won something or not. If the game is ended due to a game over, an extra button asks if the player wants to continue (see figures 1.9 and 1.10), which leads to the restart screen. If the player deliberately ended the game or has won it, the screen directly shows the restart screen (figures 1.11 and 1.12).



Figure 1.9. Screen that appears after a player is game over and the player didn't win any money.



Figure 1.10. Game over screen when player at least won the threshold money.



Figure 1.11. The game is ended and the player has won something.



Figure 1.12. The game has ended without the player having won anything.

Lastly, a click on the home button leads to the home menu, showing a summary of the number of games played, questions answered correctly and games won (see figure 1.13). It features the possibility to change the player, which resets everything to null to let another player try his or her luck (see figure 1.14 and 1.15). It also features a flag-button which leads to the leaderboard, where all the games from all players are listed in descending order of their wins (see figure 1.16).



Figure 1.13. Home Menu showing the information of current player.



Figure 1.14. Entering a new name - status before having clicked the “change name” button.



Figure 1.15. After having submitted the name change and having reopened the home menu as it closes after submitting.

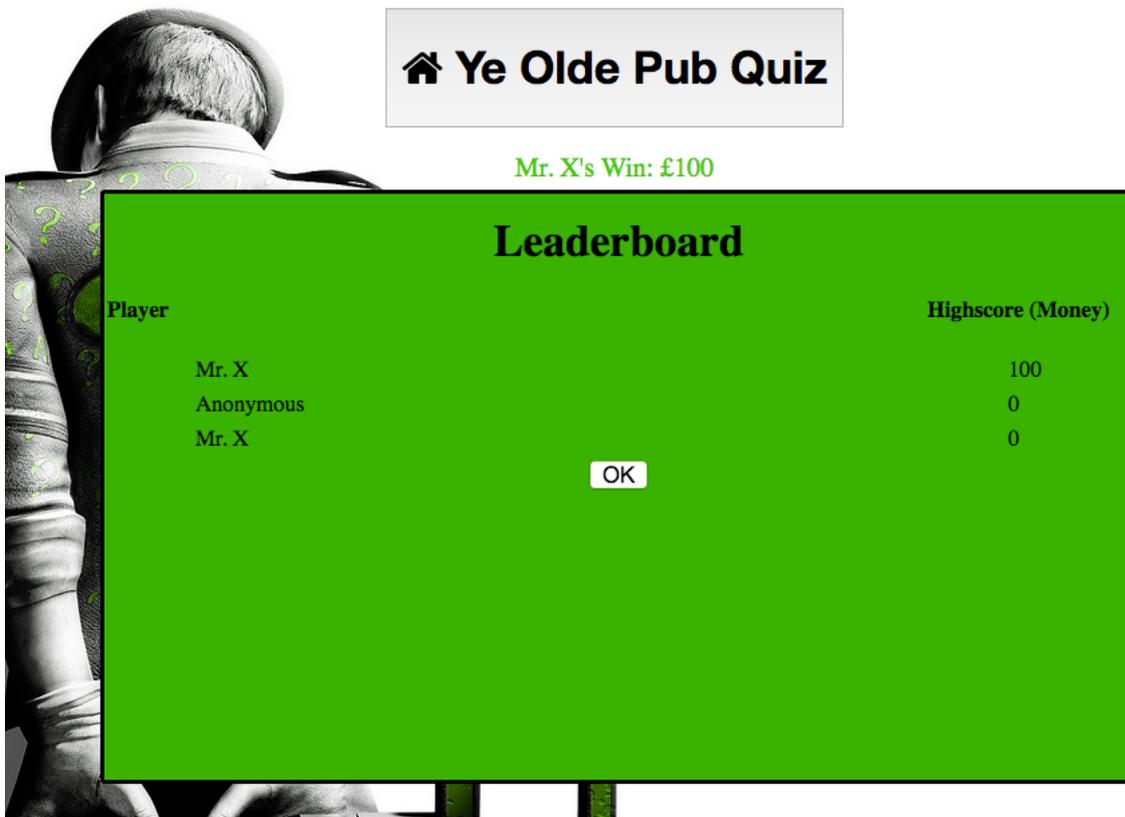


Figure 1.16. Leaderboard with the amounts won by all players in descending order and an “ok” button to return to the home menu.

2. Code design and functionality

I chose jQuery for fetching the JSON objects and for addressing and manipulating the DOM elements as it is more compact and handles errors through callback functions.

2.1 The main functions

When the page is loaded, all queries that will occur multiple times when the game is played are called once and then stored in according variables for efficiency when the page is fully loaded. Then the token for the session is fetched (“getToken()”) and if the player has submitted the name, the game starts by calling the function “restart()”. This function manipulates the screen accordingly and calls the function “newQuestion ()” which is always called when a new question is needed and fetches the JSON object.

2.2 “theFunction()”

The program is structured around the main function “theFunction()” which is called as the callback function when the JSON objects are successfully fetched and makes the changes in the webpage by hiding or showing elements to then get the main screen where the quiz game can be played. The function takes the returned JSON object, checks the type of the question and puts the correct answer randomly in one of the answer boxes. This is managed by creating a random number between 0 and 3 and putting the answer in the box with that number. For this I created an array with the four spaces (constant called “answers”). For the incorrect answers the function goes through this array and checks if the spot it is looking at right now is the one with the correct answer and if not, puts the first incorrect answer in it, then the second one and so on. For the true/false questions, this is done with a random number between 0 and 1. Furthermore, the function manages the click listeners for each of the answer boxes. When the correct answer is clicked, the function “rightAnswer()” is called, otherwise the function “wrongAnswer()” is called.

2.3 “rightAnswer()”, “wrongAnswer()”, “endGame()”

Both functions modify the screen to show the container as explained above. Additionally, wrongAnswer() checks how many lives the player has left and either sets the “Game over” screen (see above) or the one that asks the player if they want to continue or cash out (see above). If the player is game over, then it also checks if the threshold of the correctly answered questions was passed as to give the player the fixed amount he falls back to (1000 for 10 correct answers or 10000). This is set in “getDifficulty()”.

When a game ends or is ended, “endGame()” is called, where the screen is reset accordingly. Only at this point the home button becomes active and calls the function “homeScreenOn()”.

2.4 Features

The game stores the players name until the page is reloaded, allows a full game of 30 questions and counts up the earnings for every correctly answered question. The player can continue or cash out (and keep the money) after every question and loses the game after three incorrect answers.

There is a time bar that restricts the time a player has to give an answer. If the time runs out, it is handled like an incorrect answer.

During an ongoing game, the player can restart a new game session which resets everything to 0 (money, questions, etc.) and end the game.

For every game (15 questions) the player has one 50-50 joker and one time-refresh joker which are handled in the functions “ffJoker()” and “timeJoker()” respectively. After having been used, they are greyed out (see figures 2.1 and 2.2).

The leaderboard is managed by the functions “insertIntoLeaderboard()” where the current array of objects of the players and their scores (money) is sorted, “inserRows()” that creates the table with this sorted array and “showLeaderboard()” that handles the click events and showing of the screen. The scores are created in the function “getScore()”.



Figure 2.1. Both jokers are still available.

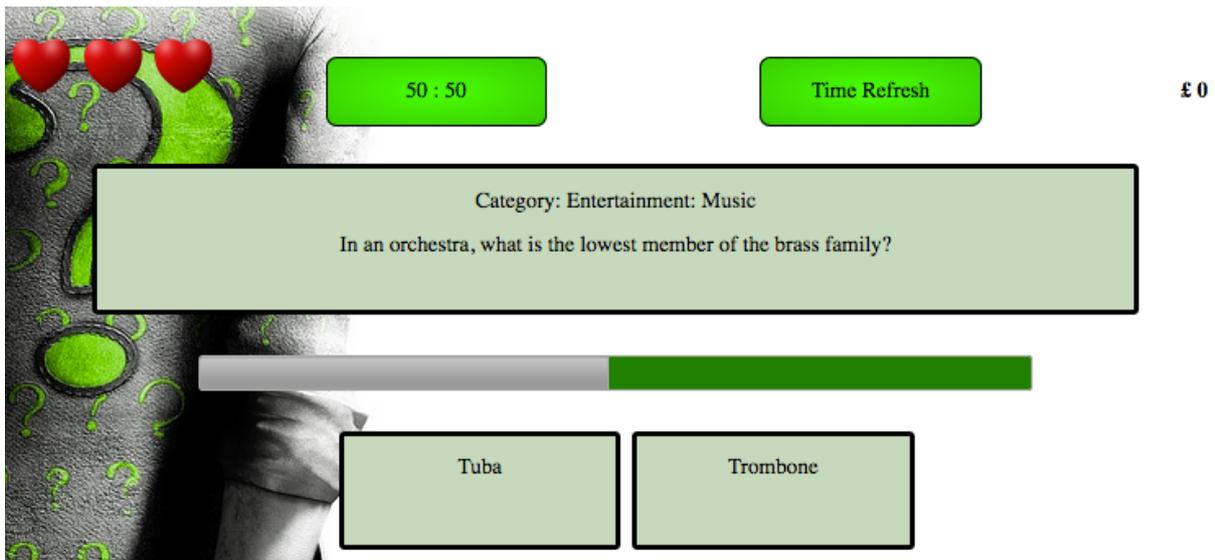


Figure 2.2. Both jokers have been used up.

Word Count: 1,229

3. References

- Background: <https://wall.alphacoders.com/big.php?i=156261>
- Hearts: <http://pngimg.com/imgs/objects/heart/>
- Icons: https://www.w3schools.com/css/css_icons.asp