TEST PROJECT

Server Side A

Submitted by:

Manuel Schaffner CH

Competition Time:

2 hours

Assessment Browser:

Google Chrome

INTRODUCTION

A friend of yours would like to have a simple online game on his website. He has already prepared the front end template, but needs some help with the server-side implementation of the interaction logic and integration into his template.

The game he has chosen is called Tic-Tac-Toe.
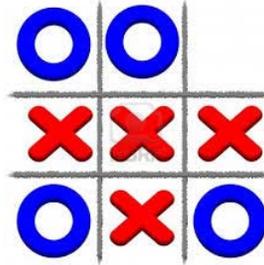
DESCRIPTION OF PROJECT AND TASKS

Tic-Tac-Toe is a common single user game. It should suit the following user stories / test cases:

1	- As an option he can upload a small photo on the initial page (no game session running or started before / first time loading the game in that browser session). It is resized to the size of 60x60px and stored on the server's file system - The player can read the game rules, see the high-scores and start a new game session by pressing the "Start new game"-button. The button is visible/functional at any time, also during a running game. A new / initialized game board will then appear on the same page. - The photo will be used instead of a "default-user"-photo in all games within that browser session i.e. the photo can only be uploaded once in a browser session – on the initial page – and will then be used for all game sessions (pressing "Start new game") instead of the default-user photo. - The photo will also be shown as the user's photo in the high-scores (if the user chooses to save a high score). - Accessing the page without an ongoing game session, no game board but the option to upload a photo will be displayed. - After a page refresh (F5), the board is still displayed in its last state without re-starting the game or changing any values (except the game time). - After the user has won (not lost), a text field to enter a nick name and submit button are shown. They disappear after pressing the submit-button. The nick name, date (dd/mm/yy), game time (in seconds; difference of timestamps from pressing "start new game" until now/winning) and number of moves by user and computer "O:4 - X:3" are saved as high score on the server and shown in the high score list. - In a tie-situation there will be no winner and the user can just start a new game if he likes (no special actions). - The high scores are displayed all the time, together with the user's photo (if he has uploaded one) and they are ordered by date/time (sequence of game sessions, latest/newest on top/first).
2	The game board consists of a square with nine fields. They are numbered as follows in the HTML-code provided (field order/numbering not to be changed):

	 • In a new game, all fields are empty. The player has the first move. The chosen field will be marked with the "default-user"-photo (or with the uploaded user-photo if available). • The computer has the next move. He will randomly choose one of the remaining fields and mark it with an X. The computer has a think time of two seconds (time until game board shows computer's move). • If there happen to be three O (default-user-photo resp. uploaded user-photo) or three X in a vertical, horizontal or diagonal row, the player/computer that did the last move has won and the game is finished - further interactions with the game board are ignored.
3	• The number of current moves and time running that game session (in seconds; difference of timestamps from pressing "start new game" until now/winning; game time updated after each interaction) are shown above the game board. • A message is displayed to inform the player if he has won or lost. • There should be no PHP-Errors appearing to the user while playing the game
4	• Moves are sent to the server by an AJAX-GET request. The response contains JSON information to update the current game board. • Data (photo, nick name, state of game,..) is sent to the server by a POST-request and a high-score/win can not be faked

Your friend would like to publish your code (with your agreement). He might also want to add other games later, reusing common code (eg. input handling from HTTP GET/POST parameters / AJAX-data, user-session and game-session handling, photo upload, saving/displaying the high scores,..). Therefore it should be nicely/modular organized, object oriented using interfaces and classes. Only methods accessed from the outside should be public.

OTHER

In the following screenshots you can see some of the different states while playing the game.

Initial call of the page (before playing a game, option to upload a photo) – URL remains the same while playing game sessions:
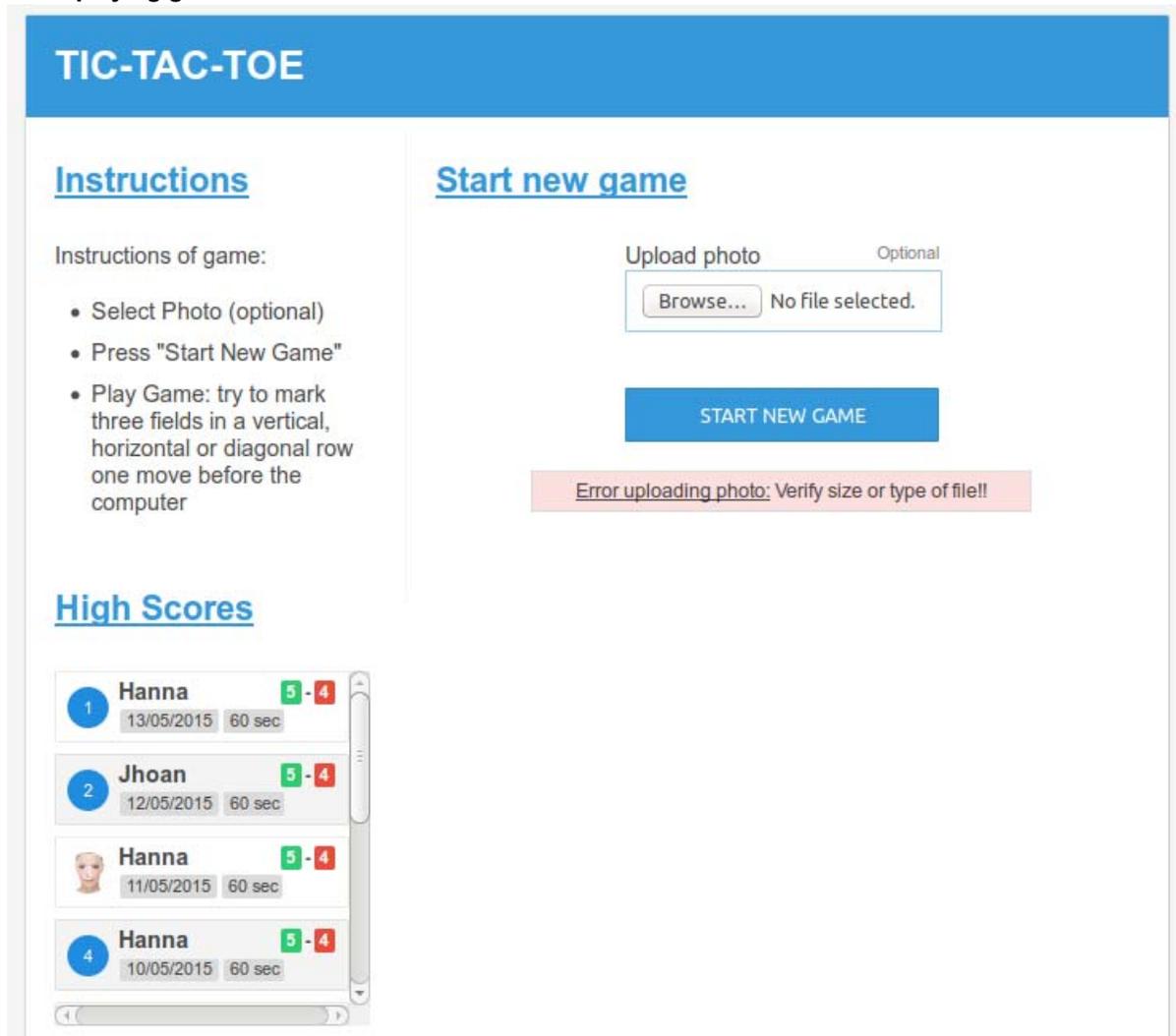
TIC-TAC-TOE

Instructions

Instructions of game:

- Select Photo (optional)
- Press "Start New Game"
- Play Game: try to mark three fields in a vertical, horizontal or diagonal row one move before the computer

Start new game

Upload photo   Optional

Browse...   No file selected.

START NEW GAME

Error uploading photo: Verify size or type of file!!

High Scores

1	Hanna	5 - 4
	13/05/2015 60 sec	
2	Jhoan	5 - 4
	12/05/2015 60 sec	
	Hanna	5 - 4
	11/05/2015 60 sec	
4	Hanna	5 - 4
	10/05/2015 60 sec	

While playing (player with photo):

TIC-TAC-TOE

Instructions

Instructions of game:

- Select Photo (optional)
- Press "Start New Game"
- Play Game: try to mark three fields in a vertical, horizontal or diagonal row one move before the computer

High Scores

1	Hanna 13/05/2015 60 sec	5 - 4
2	Jhoan 12/05/2015 60 sec	5 - 4
	Hanna 11/05/2015 60 sec	5 - 4
4	Hanna 10/05/2015 60 sec	5 - 4

Playing game...

Time:	12 sec

Player: 2   Computer: 2

START NEW GAME

After winning (player without photo, option to enter nick name):

The template contains all these three main screens of the game in one file to show how the HTML-code of the content section will change.

INSTRUCTIONS TO THE COMPETITOR

Save your files in your working directory on the server called "**XX_ServerSide_A**", where XX is your country code. Name the file to start the application "index.php" and put it directly into the directory mentioned.

Save all your files to be assessed into a suitable directory structure and naming inside the directory mentioned above. Save all your work files (those that will/should not be assessed – if any) into a subdirectory called "work".

For this module you are *not* allowed to use any frameworks or implement the logic on the client side - you need to code from scratch using PHP and MySQL. Applications developed using any framework will not be considered. Also implementing the game logic on the client side will not be considered – except for the AJAX request/response-handling.

The template for the frontend is provided and should be modified to integrate the functionality. This should happen dynamically (e.g. response from server). The application should look and behave (css..) as the provided template.

Assessment will be done on the files and the data in your database on the central server.

Create a SQL dump of your database and put the *.sql file in a folder "/dbdump" inside the directory mentioned above (XX_ServerSide_A).

Save the images uploaded by the user in a directory called "/pictures" inside the directory mentioned above (XX_ServerSide_A).

FILES PROVIDED

ITEM	DESCRIPTION
WSC2015_TP17_ServerSide_A_media_template.zip	Template files to be used

INTERNET ACCESS

- no internet access

MARKING SCHEME SUMMARY

SECTION	CRITERION	JUDGEMENT MARKS	OBJECTIVE MARKS	TOTAL
G1	Functionality: Gameplay	0	3	3
G2	Functionality: Game setup & session	0	3.25	3.25
G3	Functionality: Interaction & Structure	0.5	1.25	1.75
G4	Functionality: Interaction client-side	0.5	1.5	2
Total		**1**	**9**	**10**