TEST PROJECT WEB DESIGN

Client Side

WSC2017_TP17_CLIENT_SIDE_EN

Submitted by:

Marcelo Strehl - BR (Lead Expert)

Franz Stimpfl - AT

Volha Rabychyna - BY

Heinz Jany Guzman - CO

Wan Ahmad Jailani Wan Ngah - MY

Richard Sui Lun Wu - HK

Tsukasa Kamiyama - JP

Hassan IBNO EL FATA - MA

Kok Leong Tan - SG

Mark DuBois – US

Zabin Alharbi – SA

CONTENTS

This Test Project proposal consists of the following documentation/files:

1. WSC2017_TP17_CLIENT_SIDE_EN.PDF
2. WSC2017_TP17_CLIENT_SIDE_MEDIA.ZIP – Media Files

INTRODUCTION

In recent years, the internet has become an integral part of our daily lives, enabling the dissemination of information in an inexhaustible source of content and interaction. Every day the use of games has gained a prominent role in this universe, allowing millions of people to get access to fun and entertainment quickly and free.

Thinking about these concepts, you decided to develop a small game that works in the most common web browsers and that makes it possible to spread your talent in the skill of web design and development. The game will be called **Star Battle**.

You should design the game, develop the layout using HTML and CSS and develop client-side programming using JavaScript and its open source libraries. Some media files are available to you in a zip file. You can create more media and modify anything in the media if you want.

To be used in different resolutions your game needs to be developed in a tablet resolution with 960x600 pixels. But, if game is open in a big screen, the game must be in the center of the screen (horizontally and vertically).

DESCRIPTION OF PROJECT AND TASKS

This is a module of 5 hours. Your first 2.5 hours must be used to create the design of the game in three PNG images and the initial layout using HTML/CSS. Your layout should follow the design that you created. The final 2.5 hours you will create the functionality of game using JavaScript that allows the game to work correctly in different web browsers, following the requirements described below.

Star Battle game uses elements described below:

1. Main spaceship: Element that is controlled by the player.
2. Planets in background: Elements that move from right to left to give the impression of movement of the spaceship in the space.
3. Enemy spaceships: Elements that player needs to destroy to get points.
4. Asteroids: Elements that player needs to destroy to get points.
5. Friendly spaceships: Elements that player shouldn't destroy or player will lose points.
6. Fuel icon: Elements that player needs to collect to increment the fuel level.
7. Fuel Counter: Element that shows how much fuel is available. It must be a number and a graphical element and it should be animated when the fuel is decreasing and when the user gets more fuel.
8. Score Counter: Elements that shows how many points the user got destroying asteroids and enemy's spaceships.
9. Timer: Element that shows how much time the spaceship is flying.
10. Sound Button: Element that enable and disable the sounds in the game.
11. Font Size Buttons: Elements that increase and decrease the font size.
12. Pause/Continue button: Button to pause/continue the game.
13. Sensible areas to control the spaceship.
14. Logo: Add the provided logo in the game.
15. Shot: Shot by the ships.

FIRST 2.5 HOURS – DESIGN AND INITIAL LAYOUT:

1. **Deliver at least 3 PNG image files that present:**
    1.1. Game Instructions: The first screen of the game presents the instructions to the user and the "Start Game" button. The instructions for the game are included in the media files.
    1.2. Game board layout: This design must present all 15 elements described above in the game screen.
    1.3. Ranking Table presentation: This design must present the logo of game and ranking with the following columns: position, name, score and time in this order, with the table is presented the "Start Game" button.

2. Develop the initial markup (HTML + CSS) of your game application. When the address is accessed (http://competitorYY.wsad.local/XX_Client_Side) the game is presented to the user with the game instructions and the button "Start Game". The instructions must be presented in an animated way. XX is your country code. YY is your workstation

3. "Start Game" buttons must have active and hover effects. The background of the buttons in hover state must be: #f19e0d. The active state must follow the example called ripple which is provided in the media files.

4. You should draw the elements described below to be included in your design. Create the elements that represent the same visual style.
    4.1. Main spaceship (controlled by player)
    4.2. Timer Icon
    4.3. Sound button
    4.4. Font size buttons
    4.5. Pause/continue button
    4.6. Fuel counter
    4.7. Fuel icon

5. The HTML and CSS code must be valid in the W3C standards for HTML 5 and CSS 3 rules.

FINAL 2.5 HOURS – GAME FUNCTIONALITIES:

1. Pressing the "Start Game" button in the initial screen, starts the game. The planets should be animated and move from right to left to give the impression of movement of the spaceship in the space. The timer starts from zero and displays the time in seconds that the spaceship is in movement. The fuel counter starts to decrease, one point per second. When the game starts, there are 15 points of fuel (15 seconds). The max capacity of fuel is 30 points (30 seconds of flight). The score starts with zero points.

2. The spaceship fires when the user presses space bar, the user cannot keep pressing the space bar to fire many times sequentially, i.e. only one shoot for each space bar pressing.

3. The shot of the main spaceship can destroy just one target. The shot cannot pass through one target and hit other elements.

4. The user can move the spaceship in the screen using sensible areas that you include in the interface. As a videogame controller, these areas are activated in mouse hover and deactivate in mouse leave actions. The areas must be drawing in cross distribution. The up area moves the main spaceship up. The down area moves the main spaceship down. The left area moves the main spaceship to the left and the right area moves the main spaceship to the right. It is not possible fly the spaceship off the screen.

5. During flight, the spaceship needs to destroy the enemy spaceships and asteroids that are presented in space. If the spaceship collides with an asteroid element or a spaceship element, that element is destroyed and the fuel must decrease by 15 points.

6. The spaceships and asteroids needs to be presented in random position and in an animated way, flying from right to left.

7. The enemy spaceships must shoot (from right to left). The player must dodge these shots; otherwise, if the player is hit by a shot, the fuel counter should decrease by 15 points.

8. The enemy spaceships are destroyed by one shot. Each enemy spaceship destroyed increases the score counter by 5 points. If the user destroys a friendly spaceship the score counter is decreased by 10 points.

9. The game permits negative scores.

10. The asteroids must be destroyed with two shots. Destroying each asteroid, the score is increased by 10 points.

11. During the flight, the main spaceship needs to collect fuel icons by flying into them. Fuel icons should be dropped in an animated way from the top of the screen in random horizontal position. For each fuel icon collected, the fuel counter is increased by 15 points. One point gives 1 second of flight time.

12. During the flight, the user can pause the game clicking the pause button, or by pressing the letter "p" on the keyboard. When the game is paused, all interactions and sounds must be stopped. If the user clicks the pause button again, or presses the letter "p" again, the game continues from the moment that it was paused.

13. The planets in background need to present an animation to give the sensation of movement. The planets must move with different speed, the bigger planets must move faster than the smaller planets, creating a parallax effect. Include at least 5 different size planets in the background.

14. If the fuel counter reaches zero, the game is over.

15. When the game is over, the animations, sounds, interactions and the timer counter stops, the game will collect the name of the user in a form field "Name". The user fills the "Name" field and clicks the button "Continue". The "Continue" button should be disabled until the user fills in the "Name" field.

16. Use the file "ranking.sql" in the media folder "php" to create the table inside your database.

17. The user name, the time of flight and the score need to be registered in the server through an AJAX request. Send this information to the server address and using these variables:

    17.1.   http://competitorYY.wsad.local/XX_Client_Side /register.php
              XX is your country code. YY is your workstation number.
    17.2.   Use the register.php file available in media folder "php". Update the connection information in line 3 to use your username and password, update host and dbname to connect to your database.
    17.3.   Method: post
    17.4.   name: name of the user
    17.5.   time: number (integer) time in seconds
    17.6.   score: number (integer) of points earned during the flight

18. The register.php will register your information in one database and will return a JSON information with a ranking of users. An example of the JSON structure is presented below:

```
[
    {"id":"1","name":"Player 1", "time":"20", "score":"10"},
    {"id":"2","name":"Player 2","time":"14","score":"8"}
]
```

19. The game presents the user ranking to the user with the button "Start Game" that permits restarting the game. Clicking in the "Start Game" button the game instructions are presented to the user again.

20. The ranking needs to be ordered by the score and for time elapsed in decreasing order. If more than one user has the same score and the same time, they receive the same position in the ranking. The JSON data returned by the server is not ordered, it is your task to order the data correctly to present to the user on the client side.

21. To increase the game interaction, some sound effects need to be used:
    21.1.   background.mp3: used during the flight in loop;
    21.2.   destroyed.mp3: played when one spaceship (friendly or enemy) or asteroid is destroyed;
    21.3.   shoot.mp3: played when the main spaceship shoots.

22. Use your talent to increase the usability of the game as much as possible to permit a better experience for the user.

23. To improve the accessibility of the game you must have options to increase/decrease the font size in the screen for timer and score counter.

24. There must be an option to disable/enable game sounds. If the sound is disabled, none of sounds should be played. If sound is enabled all sounds must be played.

25. Your game should work without JavaScript errors or messages shown in the browser console.

26. Maintain your HTML/CSS and JavaScript code organized and clean to facilitate future maintenance. Use correct indentation and comments. Use meaningful variable names and document your code as much as possible so another developer would be able to modify your work in the future

27. The game needs to work correctly in two browsers, Google Chrome and Mozilla Firefox. The game requirements will be checked in Google Chrome and compatibility will be checked in Mozilla Firefox.

INSTRUCTIONS TO THE COMPETITOR

- The media files are available in the ZIP file. You can modify the supplied files and create new media files to ensure the correct functionality and improve the application. You can use any supplied JavaScript framework if you find it necessary.

- Save your design files in a folder call "**XX_Client_Side/XX_design**" where XX is your country code.

- File names:
    o   Instructions: XX_instructions.png
    o   Game board: XX_game_board.png
    o   Ranking: XX_ranking.png

- You should create additional images for each of the requested resolution to highlight hidden elements, animations, interactions, or any additional information that will assist in the presentation of the game design.

- Additional file names
  - Instructions: XX_instructions_2.png, XX_instructions_3.png…
  - Game board: XX_game_board_2.png, XX_game_board_3.png …
  - Ranking: XX_ranking.png, XX_ranking_2.png, XX_ranking_3.png …

- Save any image source files to a folder named "**XX_source**" inside the "**XX_Client_Side/XX_design**" folder. The source files are the files that contain the layers, development files, ie .psd, .ai, .svg, .jpg.

- Save the working game to the directory on the server named "**XX_Client_Side**". Be sure that your main file is called index.html.

- Be sure that the address is correct and the variables and formats are correct to permit a correct registration in the database. In case of error the JSON error message is returned as follows.

```
[
  {"error":"Error Message"}
]
```

- You are responsible for the time management in your development. If you finalize some tasks you can continue to other tasks. The initial 2.5 hours only define what will be evaluated first.

MARKING SCHEME SUMMARY

SECTION	CRITERION	JUDGEMENT MARKS	MEASUREMENT MARKS	TOTAL
E1	Client Side General	0.70	1.85	2.55
E2	Client Side Design	0.50	1.50	2.00
E3	Client Side Welcome Screen	0.50	1.65	2.15
E4	Client Side Animation	1.50	0.50	2.00
E5	Client Side Ranking	1.10	0.70	1.80
E6	Client Side Code Quality	0.70	1.00	1.70
E7	Client Side Elements	0.60	1.20	1.80
F1	Client Side Fuels	1.30	0.80	2.10
F2	Client Side Game Constraints	0.50	2.30	2.80
F3	Client Side Game Scene	0	3.00	3.00
F4	Client Side Enemies	0.60	2.20	2.80
F5	Client Side Final Design	0	2.40	2.40
F6	Client Side Game Over	0	2.40	2.40
F7	Client Side Game Quality	0.70	1.80	2.50
Total		**8.70**	**23.30**	**32.00**