Lappeenranta
University of Technology

CT10A9700 Summer School on Communications Engineering

24 hour Cross-platform HTML5 + JavaScript Code Camp 2013

Match Pairs

Poorang Vosough 0392601

Alireza Adli 0406160

Lappeenranta, autumn 2013

Table of Contents

1 Technologies we used

In this 24-hour code camp which was from 15.10.2013 till 16.10.2013 (Tuesday to Wednesday), we were supposed to work with HTML5 and JavaScript to develop a mobile application which can be deployed both on web pages and different mobile devices [1]. In this case, the technologies that we used in this code camp are as below:

- HTML5
- JavaScript
- CSS

Also, we used Microsoft Visual Studio 2012 IDE on Windows 8 operating system to develop our application. Visual Studio 2012 integrates HTML5, CSS and JavaScript technologies for developing mobile device applications and also web apps.

2 development process

Our application development process was done in two different parts. First, 24-hour code camp that started from Tuesday (15.10) till Wednesday (16.10). Second, one week time that we had after the code camp to complete our development process.

During 24-hour code camp we did the activities below:

- **Selecting the topic**

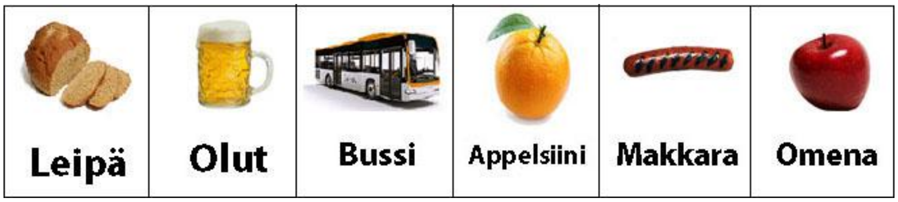
  We decided to develop a "match cards" game, which cards with the same images should be found by the player. Then, we decided to add an educational feature to this idea to make it more special and useful for the users. In fact, instead of plain image, we decided to mention the name of each image in Finnish language. Using Finnish basic words with their pictures can hit multiple objects. Tourists, students and travelers to Finland can learn basic Finnish words using this game. Also it can be helpful for Finnish kids who are starting to learn how to read and write in Finnish language.
- **Studying and Analyzing**

We had some background in HTML and CSS but our knowledge in JavaScript were quite poor. So at first, we quickly studied about HTML5 and JavaScript to become more familiar with these technologies and their features.

- **Select the best practice logic for developing Match pairs**

  Reviewing open source games on the web [2, 3, 4] which were similar to the idea we wanted to develop, helped us to select best practice logics for implementing Match Pairs.
- **Selecting the images**

  We decided to use six pairs of images with their Finnish names (twelve cards on the screen). The six images we selected are: bread, beer, bus, sausage, orange and apple; as figure 1 presents, the images include the name of the image in Finnish language.



Picture 1 the images of the cards

- **Coding**

  After studying the technologies and indentifying the best practices we started coding. We continued coding till Wednesday morning at 4:00am until Match pair run successfully.
- **Testing**

  We were testing the application during the implementing process, but we did the final testing of the application after the application successfully run.
- **Presenting**

  We presented our application as the first group on Wednesday (16.10) at 10 a.m.

Then, in one week time we had to complete our development process, the activities in below were done:

- **Adding game over page**
- **Putting comments on the code**

- **Testing and improving application logic**
- **Writing report**
- **Making our game poster**

3 Roles of the members

We divided tasks between group members in a way that both members are involved in coding and other activities. Table 1 presents how tasks are divided between group members.

Members	Roles
Poorang Vosough	• Developing matchpair.js & matchpair.css • Testing and debugging • Improving the code logic • Preparing slides & poster
Alireza Adli	• Developing default.html • Testing • Group wiki page • Preparing final report

Table 1 roles of the member

5 The code logic and different parts

There are three major layers (files) in Match pair project, Default.html, Matchpair.css and Matchpair.js. In this section the content and logic of these layers are briefly discussed.

- **Default.html:** In this layer the presentation layer of the game is formed. There are three different environments which are defined in presentation layer. First, starting page of the game which includes a "play" button to start the game; Second, game page which is the main page of the game; cards are located in game page when "play" button is clicked by the player. Finally, end page of the game (game over page), which is shown when all pairs are found by the player and the game is over.

- **Matchpair.css:** In this layer size of the pages, cards, images are initialized, back ground images of different game environments are set and six images are defined as pairs which are supposed to be found by the players.
- **Matchpair.js:** The logic of the game is defined in this layer. Init() and StartGame() functions define the mode that the game starts in. When the games starts, six pairs are randomly located on the screen. Users are supposed to find pair out of twelve cards; if two cards are selected by the player are a pair, they will be removed from the screen, if not they will be flipped back and remain on the screen. This process will continue until all pairs are removed from the screen. These logics are defined in below functions:

    - SelectCard()
    - CheckPattern()
    - IsMatchPattern()
    - RemoveTookCards()

6 Future works

Because of time limit, we couldn't finish all of our ideas about this game, though we consider ideas below to make this game as we want it to be in the first place.

Difficulty level: The most important option that we want to add to this game is to have different difficulty levels. So player can choose between easy, medium and hard modes which have 12, 18 and 24 cards respectively. Hence, game includes more words to teach or review.

Language selection: This application can be used as a part of language learning package or independently with possibility to choose between Finnish, French and Spanish languages. In this case, Match pair will be useful for wider range of users.

Reference

[1] Codecamp.fi. 2013. 24h Cross-Platform JavaScript Code Camp [web page]. [Accessed 23 October 2013]. Available at codecamp.fi

[2] Frebsite.nl. 2013. [online game]. [Accessed 23 October 2013]. Available at http://coolcarousels.frebsite.nl/c/36/

[3] My-html-codes.com. 2013. Dino pairs game| Using CSS3 Animation [online document]. [Accessed 23 October 2013]. Available at http://my-html-codes.com/HTML5_tutorials/Dino_pairs2/index.html

[4] joebilman.com. 2013. Card Match [online game]. [Accessed 23 October 2013]. Available at http://www.joebillman.com/projects/card-match-js/