User Interface

Design Plan IV (Final)

Software Engineering Capstone Project Team 2

10/26/2013

Contents

1.0 Introduction: The Final Design

This report is a brief overview of the UI Design process. It includes a brief look at the different design approaches and reasons why the chosen design approach was chosen. Furthermore, it looks at the how the UI changed over the course of the project and gives a brief overview of the final UI design. It also contains a discussion of the challenges faced and how they were overcome. Lastly, it discusses some aspects of the process that could have been executed better and provides some suggestions on how the overall process can be improved.

2.0 The Approach

There are various approaches to designing user interfaces. Two more recent and relevant approaches are skeuomorphism and flat design. They are the main topics of this year's debate in the design community, subjected by the release of Apple's new iOs.

2.1 Skeuomorphism

Up until recent years, skeuomorphism has been very popular in designing interfaces – earlier versions of Apple iOs and Microsoft Windows prior to Windows 8.

Skeuomorphism brings the real world into its designs and plays towards natural human intuition. It creates visual effects to give a look and feel of real world objects, textures and materials.

An example of this would be older versions of the Apple iOs calculator. Seen below, this design incorporates graphics that give a look and feel of a real life calculator. When a user looks at this design, they instinctively press the relevant buttons. The power of skeuomorphic design is that it creates designs that are familiar, which makes it easier to connect with the user.

Figure 1 Apple's iPhone calculator vs. real calculator

Another example, shown below, is a navigation bar element on Trade Me's website. Here, a depth is given to the button to give the illusion of pressing a physical button.

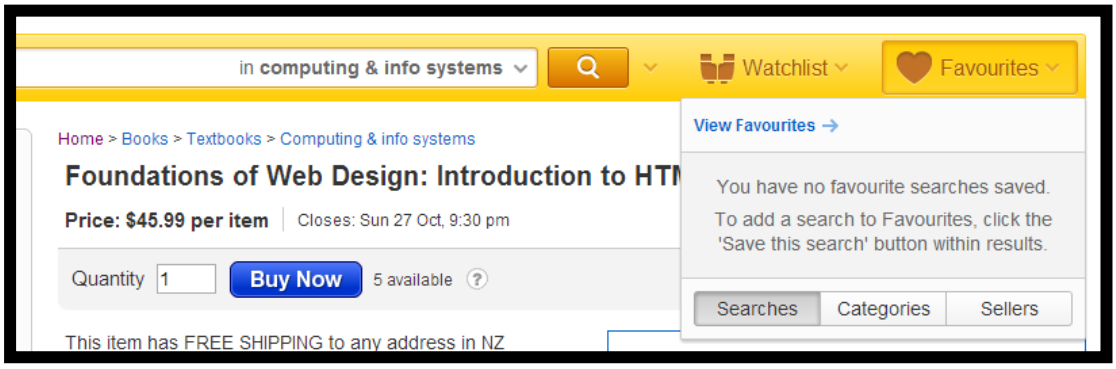

Figure 2 Trade Me: Navigation Bar

2.2 Flat Design

On the other hand, Flat Design strips down elements to their simplest form; there are no shadows, gloss, gradients or embedded elements. Instead, it plays with blocks of color and its different hues and also makes use of typography. Flat design focuses more on functionality rather than how it looks.

An example of flat design is Microsoft's Windows 8 Metro user interface, seen below.

Figure 3 Microsoft: Windows 8 Metro UI

2.3 Our Approach

Skeuomorphism creates designs which are closer to the real world, which means that it connects easily with the user because it is familiar. However, at times, the designs may seem "busy" and crowded. As a result, users may sometimes feel overwhelmed or disoriented.

The aim of this project was to create a user interface simple clean lines so that the focus was in playing the game. This is why a flat design approach was chosen as the general outlook of the user interface.

3.0 Evolution

Inevitably, the UI has changed dramatically over the course of the project. From the basic monochrome HTML to the final design, various ideas have shaped the design one way or the other.

A strong factor affecting this was the eventual addition of features as the project progressed. Initially, the main work of the project was laying the framework of the game logic; there was nothing to display. As the project progressed, more and more features were added and the UI began to take form.

Throughout the course of the project, the UI advanced through many different versions. One of these involved Bootstrap – the Twitter framework for web development. However, the elements provided by the Bootstrap library were rather skeuomorphic, as seen below, so that the idea was discarded. However, the final design still uses the Bootstrap styling for the select forms in the game lobby list, public lobby list and played games list.

Despite the many changes in the UI, the initial idea to have a simple and minimal interface still persevered throughout the course of the project.

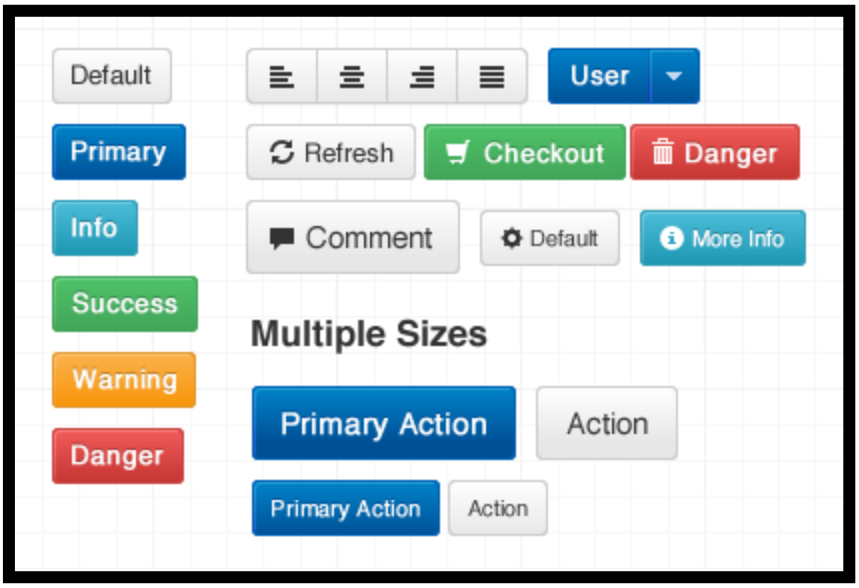

Figure 4 Bootstrap buttons

4.0 The Final Design

4.1 Basic Structure

The basic structure of the final design incorporates a simple layout consisting of a navigation bar with a menu set over the main page. The navigation is fixed to the top of each page so that the menu can be used to navigate to any other page from any page. The simplicity of this layout means that users are able to find their way around the website efficiently.

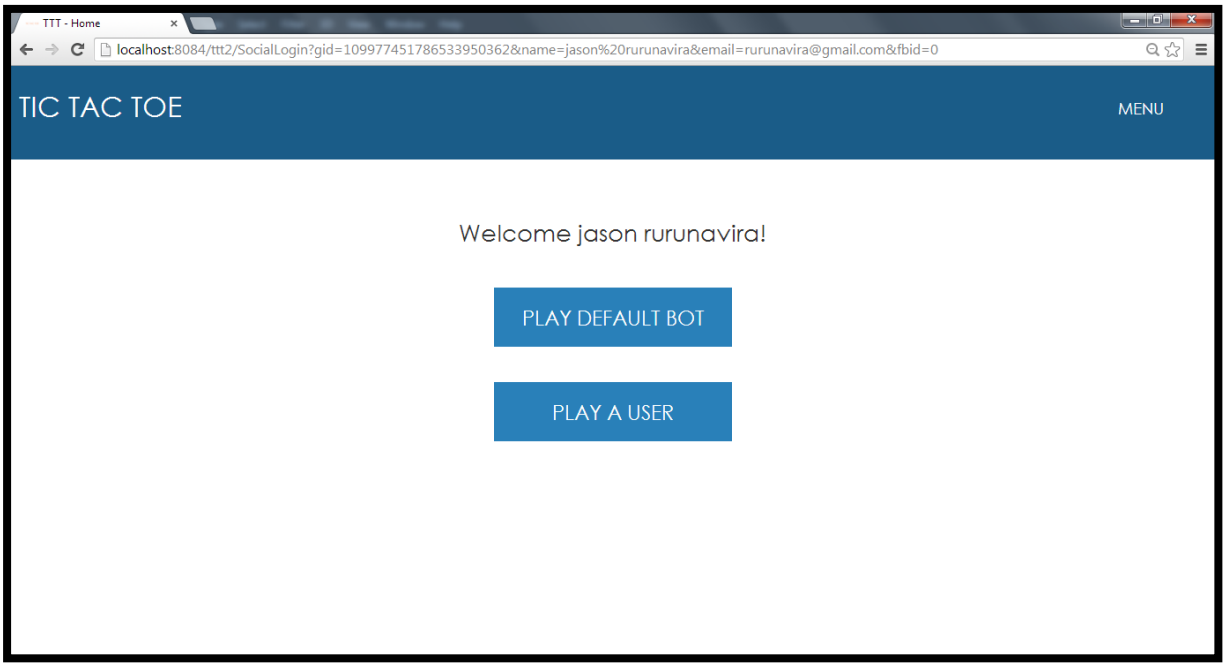

Figure 5 Tic Tac Toe: Homepage

4.2 The Features

The following are the available pages and features of the website. Refer to ***Appendix A*** for the screenshots.

1. **Login**
2. **Home**
3. **Play the default bot**: Play against the default bot.
4. **Play another user**: Play a game against another random user.
5. **Play a private game**: Play a game against a friend on Google+ or Facebook.
6. **Create a bot**: Create and submit a bot.
7. **Bot lobby**: View list of bots created by the user and their friends.
8. **Game lobby**: View list of public games.
9. **Played games**: View and analyze games played before.
10. **Logout**

4.3 The Elements

One of the goals in the design process was to try and build everything from scratch using basic HTML and styling them in CSS. The buttons, menu in the navigation bar and game grid are some of these custom elements.

The buttons, for example, are links with a styled background and a hover effect.

Figure 6 The buttons

```css
/*buttons*/
.buttons {
        background: #2980b9;
        color: white;
        padding: 10px;
        margin-left: auto;
        margin-right: auto;
        white-space: nowrap;
        text-decoration: none;
        cursor: pointer;
        font-weight: 500;
        text-transform: uppercase;
}
.buttons:hover {
        opacity: 0.8;
        text-decoration: none;
        cursor: pointer;
        background: #3498db;
        color: white;
}
.buttons a {
        text-decoration: none;
        color: white;
}
```

Figure 7 The buttons: CSS

```
><div class="buttons buttons2 padBottom" id="playButton" ><a href="#" onclick="playGame();">Next Move</a></div><br>
 <div class="buttons buttons2 padBottom" id="backButton" ><a href="#" onclick="backGame();">Back Move</a></div>
```

Figure 8 The Buttons: HTML

Figure 6, above, demonstrates the hover effect on the buttons. The *NEXT MOVE* button is being hovered upon; therefore, it is a lighter shade. The HTML layout of the buttons is show in Figure 8. Each link is given a *buttons* class; this class contains the styling in CSS, shown in Figure 7.

4.4 The Game Grid

The most important element of the website, however, is the game grid. This is a table that is dynamically filled with subtables, in which each subgame is created.

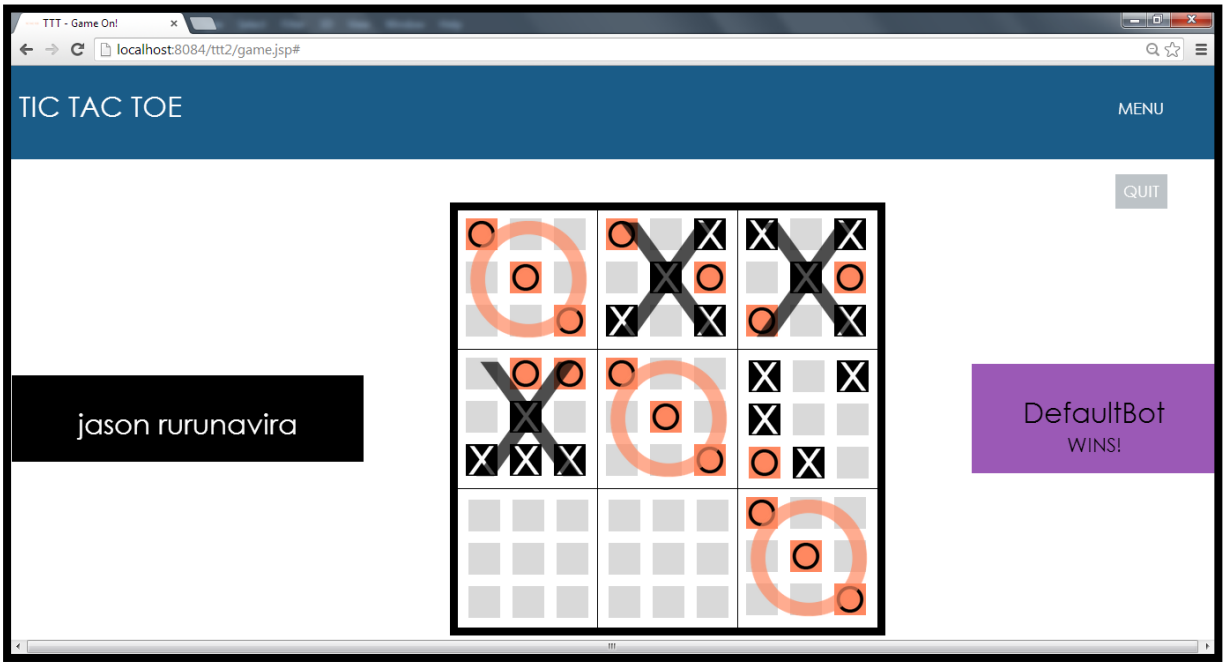

Figure 9 Tic Tac Toe: Game page

The game grid, as shown above in Figure 9, is a 3x3 table of which each cell is a 3x3 subtable.

Each cell of this subtable represents a position on the tic tac toe board. The cells consist of clickable images which trigger the function to make a new move. Upon making a new move, the cell is updated to an image of the corresponding token, X or O. When a subtable has been won, an overlay image of the subgame winning token is displayed over the completed subgame.

During the game, the labels on either side of the game grid, alert the players whose turn it is, the winner of the game (shown above) and if the game is a draw. The UI uses alternating colors and element opacity to achieve this.

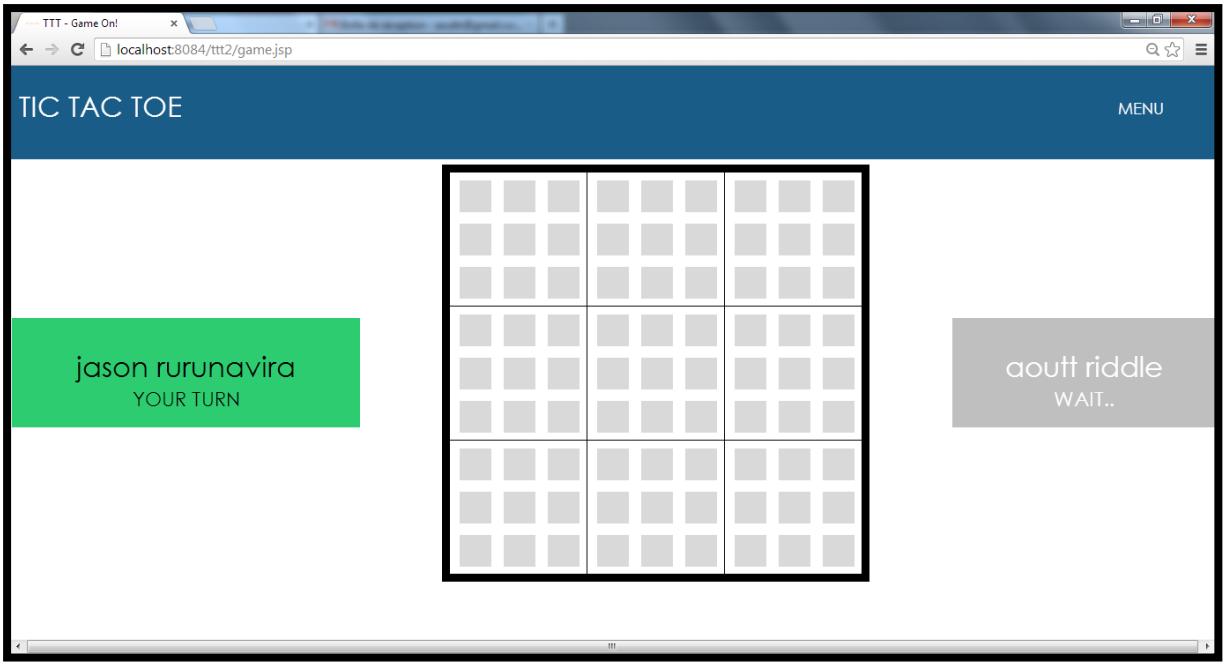

Figure 10 Game page: Alternating colors alert user whose turn it is.

4.5 The Game Grid: Dynamic Populating and Styling

Each cell of the main game table is a parent element to the subgames. When it is created, it is given an ID and a class name of *subTable*. The ID is used as a reference when updating a new move to a specific subgame.

Within this *subTable*, a table, with class name *subGame,* and an empty span element are created. The span element is given a matching ID as that of the parent *subTable*. When a subgame is won, the span tag has an image of the winning token created within it. The span class name is also changed to *subGameWin*.

These classes are used in styling the subgames and displaying the winning token for each subgames.

```
▼<td class="subTable" id="0-2">
  ▼<table>
    ▶<tbody>…</tbody>
    </table>
    <span id="0-2span"></span>
  </td>
```

Figure 11 Subgame Not Won: HTML

```
▼<td class="subTable" id="1-2">
  ▶<table class="subGame">…</table>
  ▼<span id="1-2span" class="subGameWin">
     <img src="images/ex_win.png">
   </span>
  </td>
```

Figure 12 Subgame Won: HTML

A similar technique is carried out when a player makes a new move. In each cell, an image of a blank token is placed. When a player makes a move, this image is updated to the correct token. The classes of the cells are updated from the style of default button to the player token. The figure below shows the HTML structure of the three different types of the buttons. The default button uses *blank.png*; the X token uses *ex.png* and the O token uses *oh.png*.

```
▼<td>
  ▼<a href="#" id="0-2-0-1">
     <img class="fadeGameButton gameButton" src="images/blank.png">
   </a>
 </td>
▼<td>
  ▼<a href="#" id="0-2-0-2">
     <img class="gameButton" src="images/ex.png">
   </a>
 </td>
</tr>
▶<tr>…</tr>
▼<tr>
 ▼<td>
  ▼<a href="#" id="0-2-2-0">
     <img class="gameButton" src="images/oh.png">
   </a>
 </td>
```

Figure 13 HTML structure of the game buttons.

5.0 Challenges and Improvements

5.1 Logic vs. Style

The main challenge of the designing the UI was to keep the styling separate from the logic framework.

The underlying game codes changed constantly over the course of the project. It was important to keep the styling separate from the logic framework to avoid modifying important code that was not relevant to the styling. Furthermore, isolating the styling would make it more efficient to progress as the focus would purely be on, CSS, for example.

To achieve this, elements were grouped into classes and all the styling was completed in CSS, as discussed in the previous sections. This approach worked well in keeping the styling from the logic framework. Furthermore, it made it easy to alternate between making general modifications and also make specific changes to a single element.

5.2 End User Testing

A drawback with Flat Design is that, since it focuses more on functionality over aesthetics, it creates interfaces that may seem too simple and not intuitive. Because of this, users may feel detached and disoriented. Microsoft's Windows 8 Metro UI, for example, had a huge learning curve for users moving on from the skeuomorphic designs of previous Windows versions.

Extensive end user testing will be valuable in making the website more engaging and user friendly by increasing its usability. Throughout this project, there hasn't been significant end user testing and this is evident in the mixed responses to the final product.

A strong recommendation for the future would be to have extensive end user testing.

6.0 Conclusion

All in all, the idea to have a simple minimal design persevered throughout the course of the project. Despite the various changes, the UI evolved in the direction set within the goals of the project UI design plan. In retrospect, more research into the design approaches and the chosen approach would have given the vision more depth. With future end user testing and research, there is great potential for improvements.

Appendix A: Screenshots

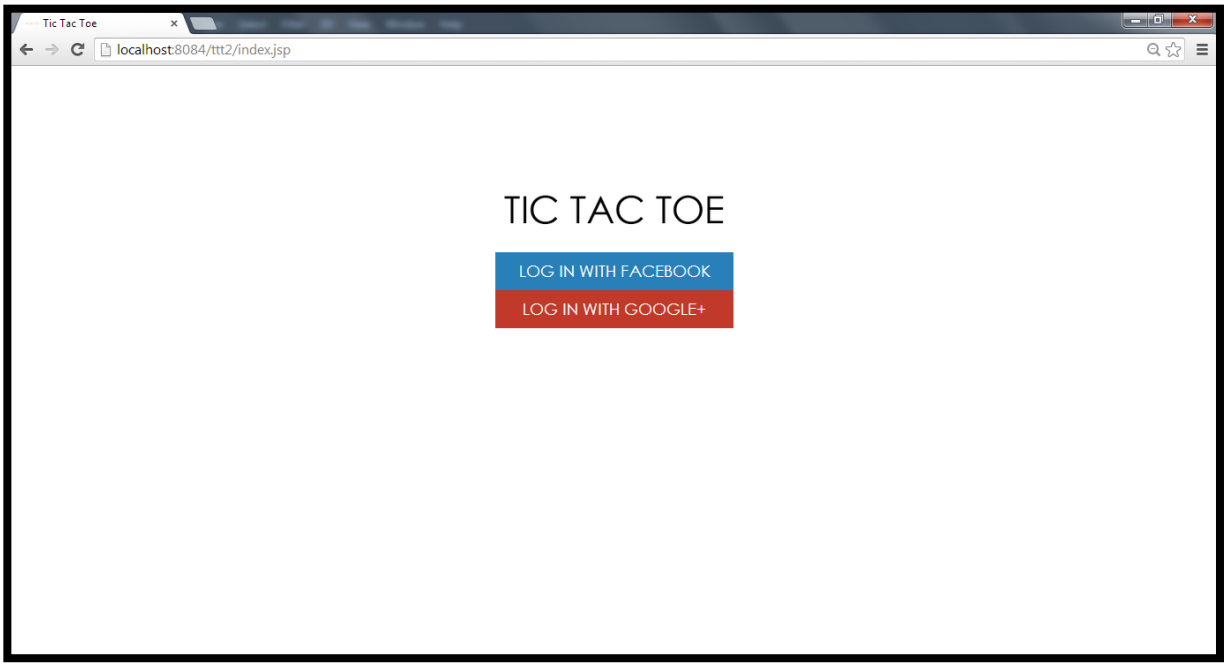

Figure 1.1: Login Page

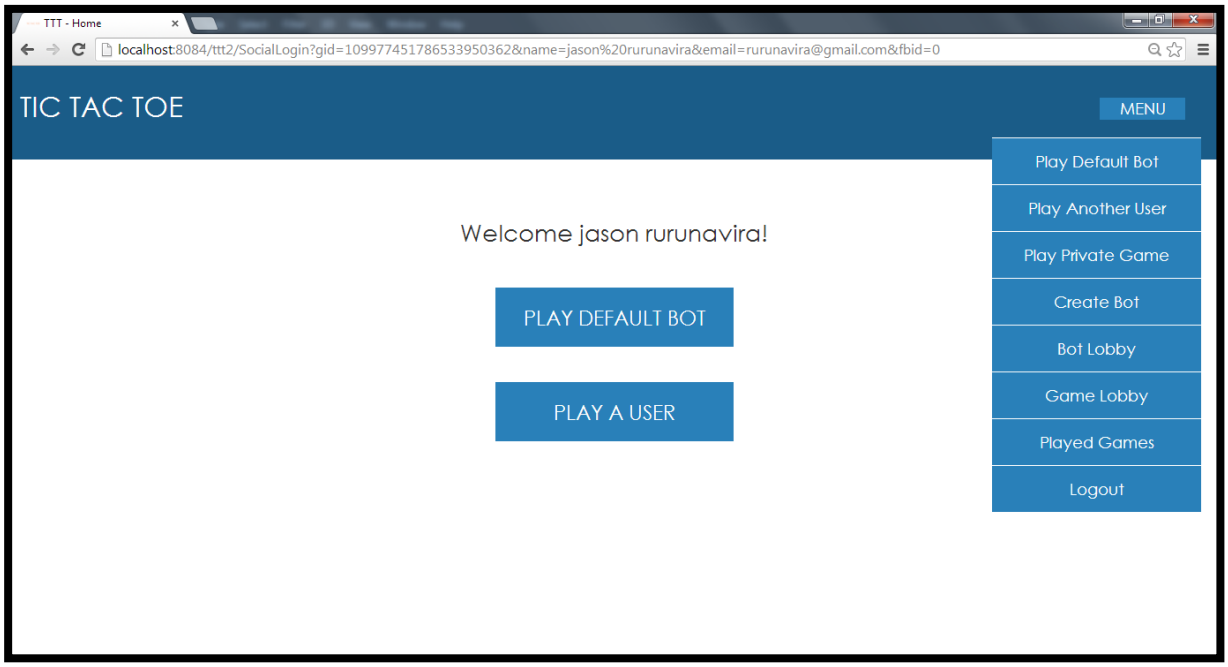

Figure 1.2: Home page

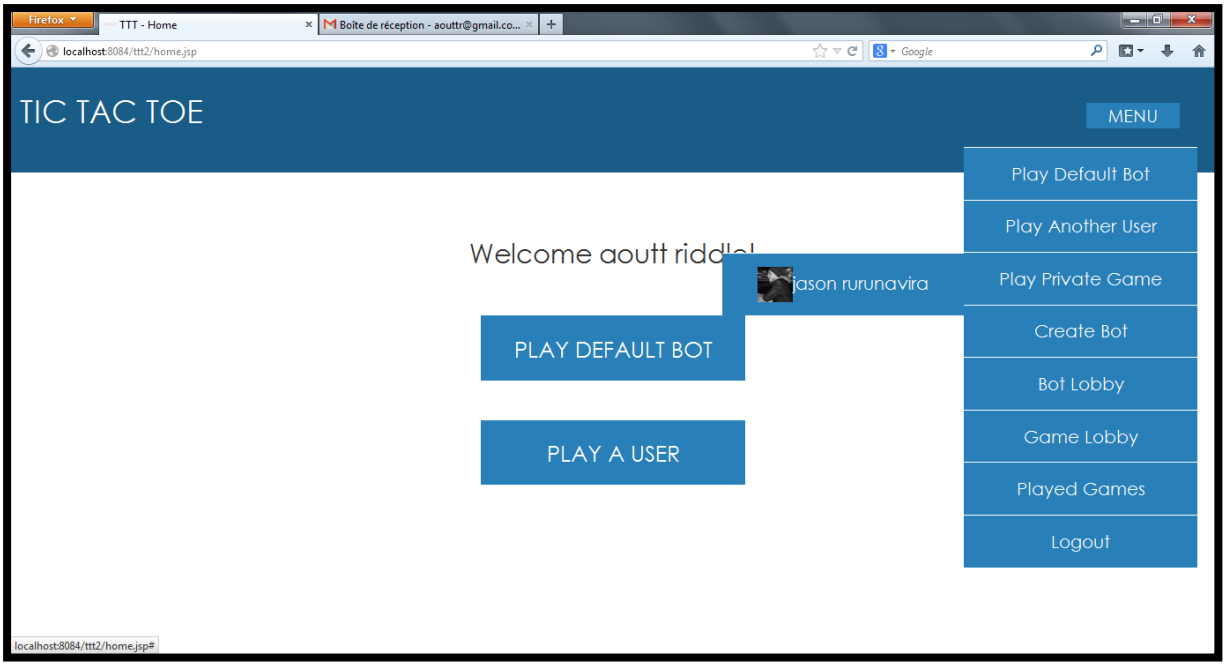

Figure 1.3: Menu showing online friends

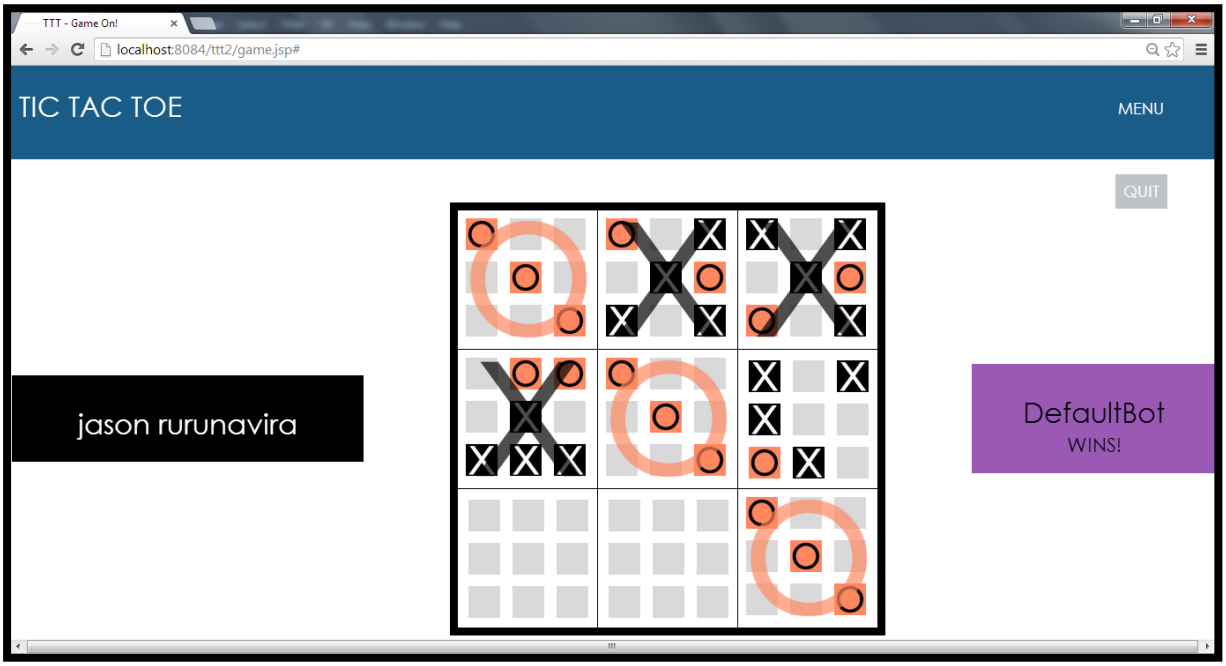

Figure 1.4: Game Page

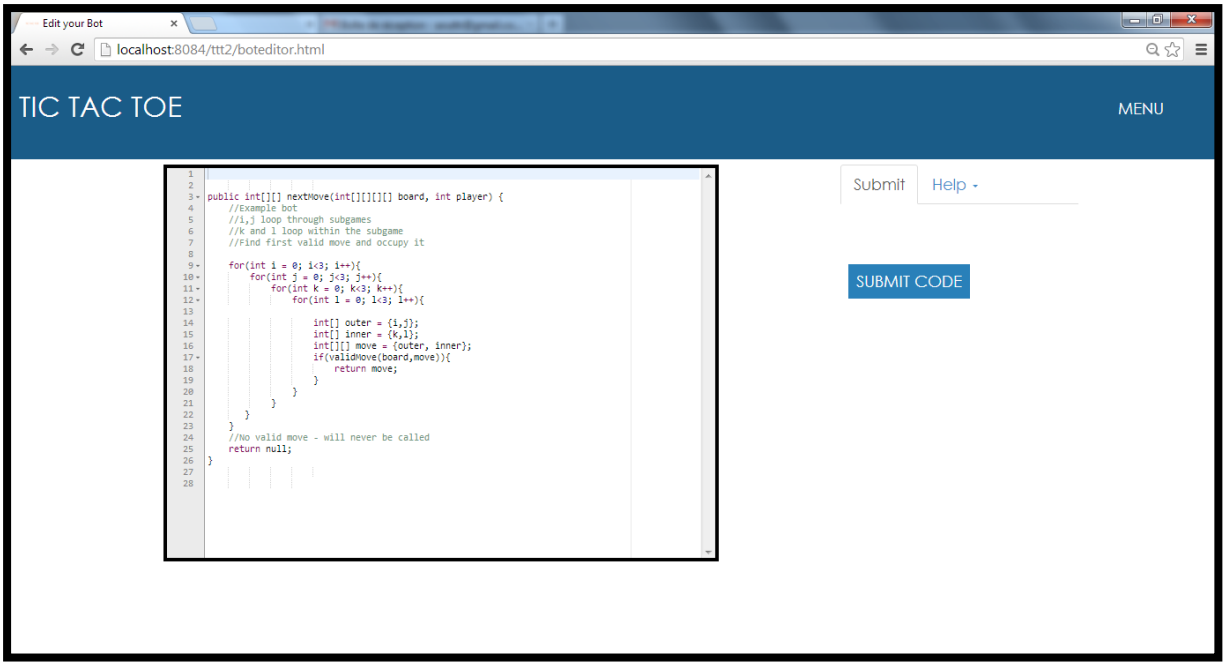

Figure 1.5: Create Bot page

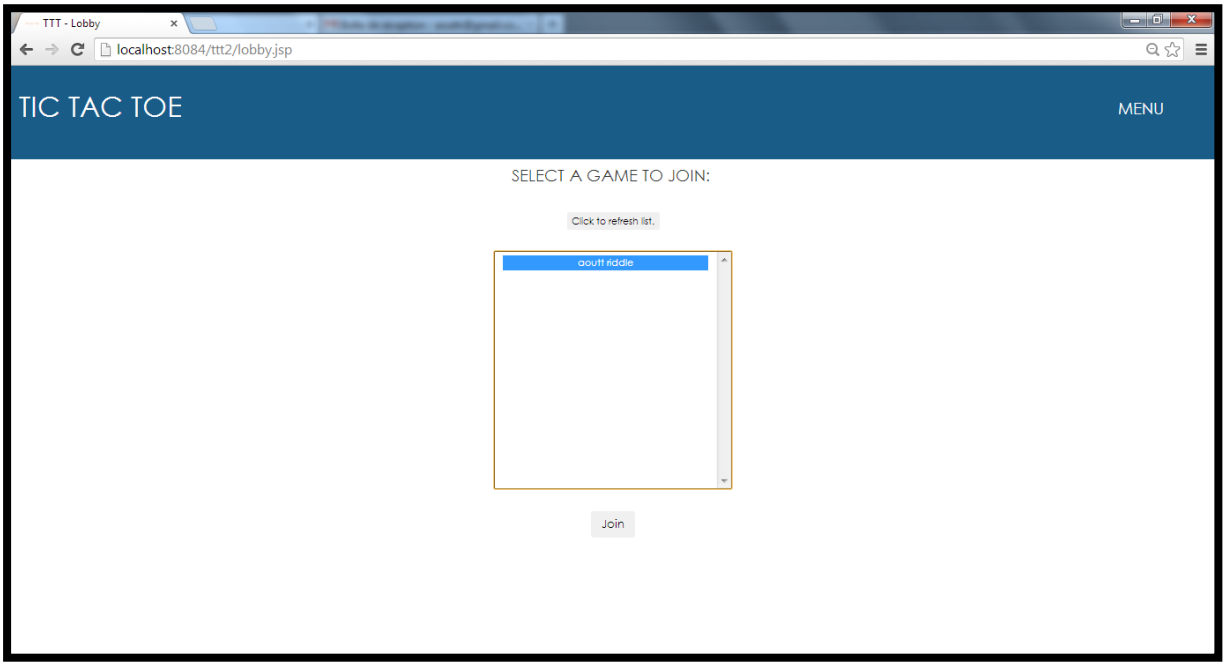

Figure 1.6: Game Lobby

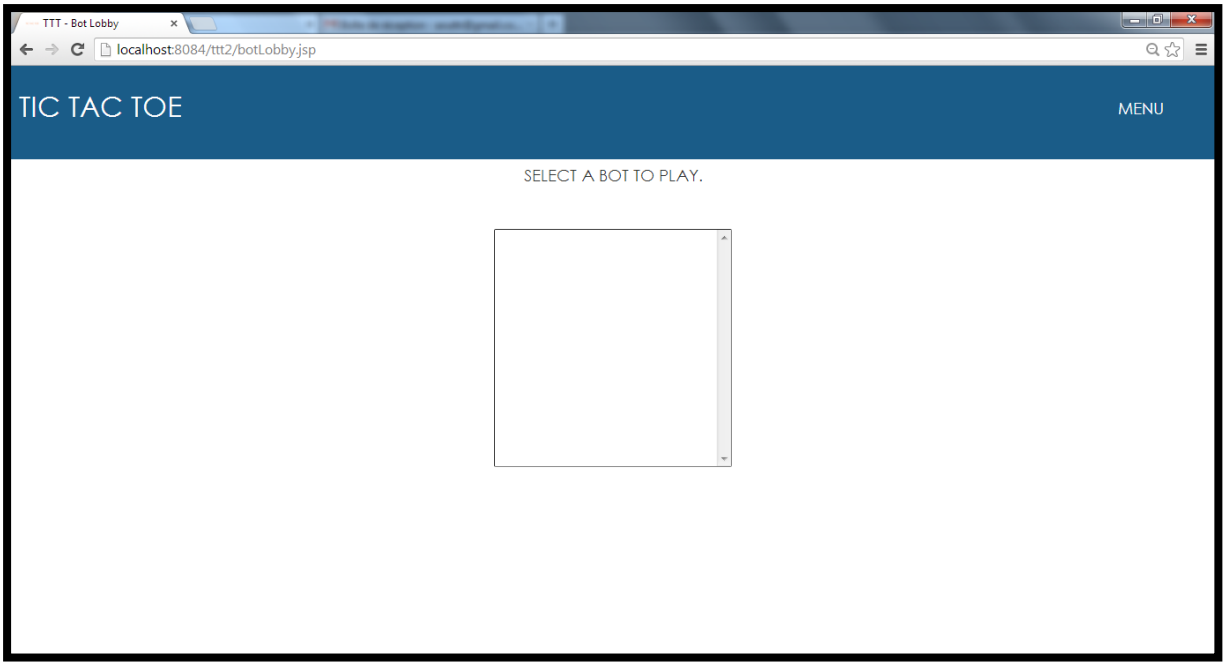

Figure 1.7: Bot Lobby

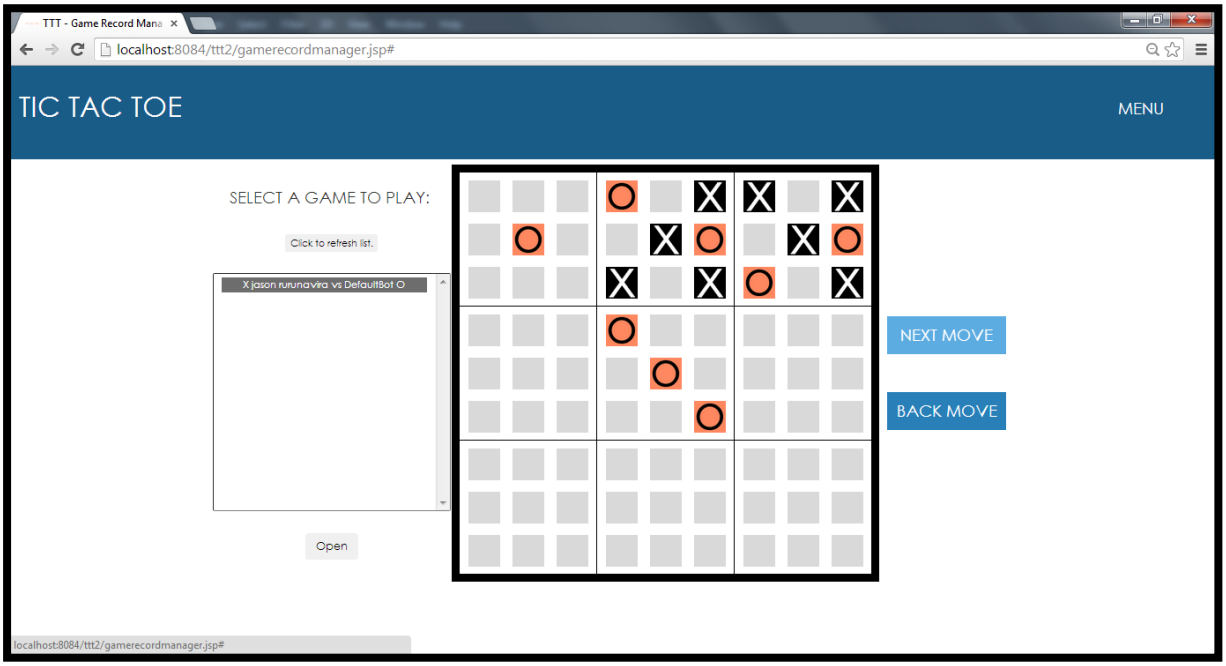

Figure 1.8: Played Games