Expose for Bachelor Thesis

Programming a cross-platform distributed system using client-server architecture with HTML5 techniques

1. Motivation

HTML5 has been the major and recommended version of HTML since 2014, which improves the ability of handling multimedia and graphical elements and running web application on mobile devices. In this situation, increasing framework based on HTML5 can be run on both mobile and desktop devices. Meanwhile, distributed system still plays an important role in web application deployment. It is often used to decrease the cost of the network or tolerate more individual failures in a system. The combination of the HTML5 techniques and distributed system can allow mobile devices, which normally have weaker computing power, to collaborate with stringer devices, such as personal computers.

1. Objects and Limitation
   1. Object

The thesis aims to construct a cross-platform distributed system in the form of a HTML application which is a cross-platform turn-based multiplayer soccer game. The web application will use ‘Node.js’ and ‘Socket.io’ to set up the server, and run ‘Phaser 3’, a HTML5 game framework, on the clients to implement the soccer game. The user can enter the application through the web browser. The application allows up to four users to play the game at the same time under the same LAN network and the AI will make up the place when the number of users is less than four. Each player controls the character to hit the ball into the goal to win a score.

* 1. Limitation

1. The application built for the thesis mainly focuses on the LAN network, whose network quality of the server is normally better than the ones on the Internet. In this case, the example application may still need to be refined before running on the Internet.
2. In actual deployment, different architectures of the distrusted system are often used together to build a more reliable and faster server. The thesis may not have enough time to test the combinations of the architectures, which can make the conclusion less practical.
3. Meaning

The thesis focusses on finding the solution for synchronizing the data among the server and clients and adjusting the GUI on different devices automatically. On the one hand, comparing to the “Peer to Peer” architecture, the “Client-Server” architecture may enable users to experience a more stable performance due to the synchronized communication. On the other hand, automatic GUI adjustment makes it possible for the web applications to run on different devices.

1. Thesis Structure

* Task description
* Declaration
* Abstract
* Table of Contents
* Introduction
* Main content
  + Background
  + Game introduction
  + Server introduction
  + Experient and collected data
  + Analysis
* Conclusion and outlook
  + Conclusion
  + Outlook
* Acknowledgement
* References
* Appendices

1. Challenge
2. Fitting the different display sizes on the different devices.
3. Avoiding and eliminating the intentional or unintentional fault made by client.
4. Narrowing the possible gap of latency caused by different computing speed and network situation.