Capstone Project Proposal

Team Name : Team NGX (New Game eXperience)

Prepared by: Ikwhan, Chang

March 9, 2014

EXECUTIVE SUMMARY

Team Name

Team NGX

Team Members

Ikwhan Chang
Sangtae Lee
Dongkyoung Jo

Project Name

Facebook Remote Volley Ball

Introduction

Two weeks ago, there is a huge mobile conference in Barcelona, Spain which is called “MWC”. Most of promising and massive mobile companies joined that event, showing their most advanced mobile technologies. Samsung, a massive world’s well-known mobile device manufacturing company, was also joined MWC, promoting the newest mobile device such as Galaxy S5, Galaxy Gear Fit. Especially, Galaxy Gear Fit is focused on health-care society, collecting the owner’s information: Heartbeat or Walk Steps and of course the owner can check their mobile information like SMS, Phone, Calendar even though they cannot open their host device(in this case, Galaxy S5). Moreover, many companies will join that competitive markets kind of wearable smart device.

In the era of high technology, most of IT companies focus on new technologies such as Cloud Computing or N-Screen, which means most of people has two or more smart devices and they do not want to make their data repeatedly for sharing in their devices so they want to OSMU(one-source multi-use) their personal data.

Our Facebook Remote Valley Ball project will be based on these trends, making a simple valley ball game in Facebook in-app platform. This game will be remote controlled using user’s own smart devices such as Galaxy or iPhone and they can see their game play view through Facebook in-app pages in their laptop or PC.

Goals

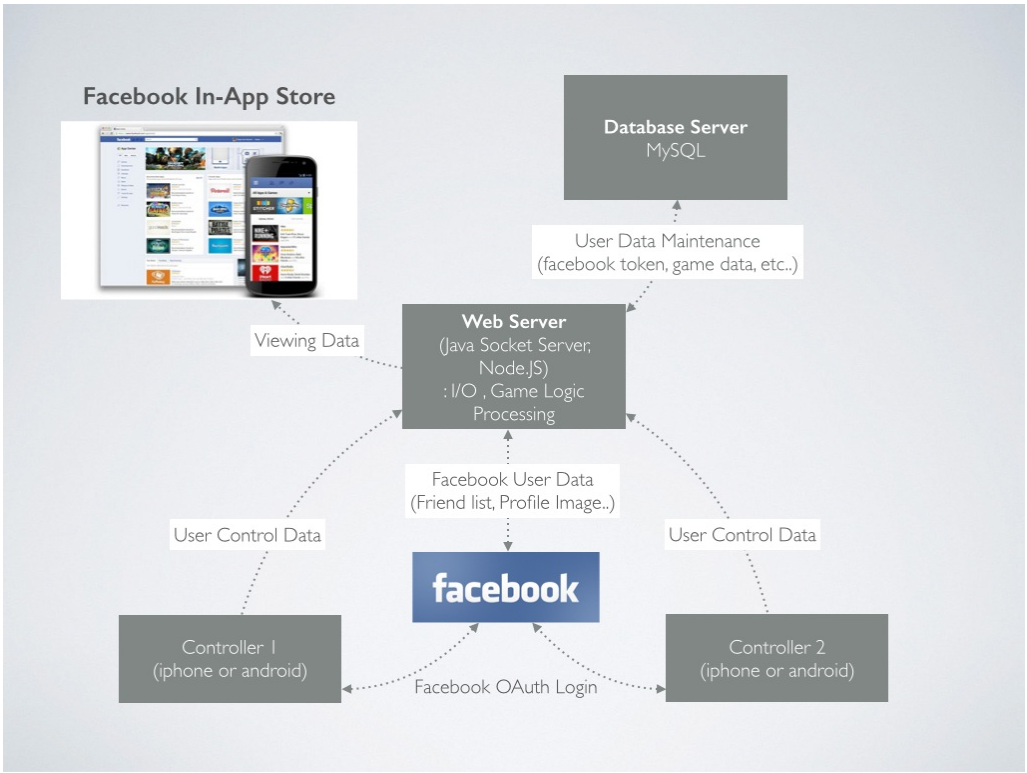
As we said, we will make just simple valley ball game. The most important thing, however, is user can REMOTE control their game using their smart devices for playing game that will be played in Facebook’s in-app pages. Our core skill is JVM Server for I/O between server and client, Web Server using Node.js for game logics, sharing data throughout MySQL Database server. Also we will make both Facebook in-app games using HTML5 or Adobe Flex(not yet decide) and smartphone application using Adobe Cordova for making hybrid native application. (If we have no time, then making android-based application will be our first thing to do)

Project Outline

Specification is as below.

- Server
  - Spring : web MVC framework
  - JSON : data format
  - Tomcat : JVM web server
  - MySQL 5.5 : Database
- Client
  - Apache Cordova : for Hybrid(multi-platform) native application
  - Adobe Edge(or Unity) : for Facebook application
- Cooperation System
  - Jenkins : Continuous Integration
  - Github : Repository
  - JIRA : Issue tracker, sharing TODO
- Tools
  - Eclipse with STS(Spring Tool Suite)
  - Eclipse with Android SDK
  - IntelliJ IDEA
  - Adobe Edge
  - Adobe Photoshop : for UI

Project Architecture(simple)



How to Play(Example)

1. Install facebook application on user's account(user should have their own Facebook account)
2. Add their primary controller(default : keyboard)
   - user can add or change their devices (e.g. iPhone, iPad, Android device)
3. Connect to our Facebook game through Facebook on PC
4. Make new room or Join room which has just one player
   - User also join room which is already started for just looking game(they can make a chat)
5. Ready and Play
   - If user use their own device(e.g. iPhone) for controlling their game, they can also use chatting, video, audio through their device
6. Game finish and writing post in their own Facebook timeline about results.

Game Rules

- It same as pikachu volleyball : http://games.softpedia.com/get/Freeware-Games/Pikaball-Volleyball.shtml
- Two players, One ball
- One character : seems like Pikachu
- One net on the centre of screen
- Player has each their own area which is split up an Net located in centre of screen
- Available action
  - Moving : left, right, jump
  - Shoot or Push
- If the ball touch on the ground during game, the opposite user can get a single point
- Time limitation : 3 min or 5 min(based on room owner's choice)

Why Facebook API?

At first, our application wants to connect between users. If they hardly find their friends, however, then user maybe boring. For finding their friends easily, we should makes conformably. Secondly, if user installed our Facebook application in their's own account, then they want to add their controller(e.g. iPhone, Android). For easy way to add their own device, they should download our iPhone or Android application via App Store, run the apps, and login their Facebook account. As a result, they can find their own devices through this processes.

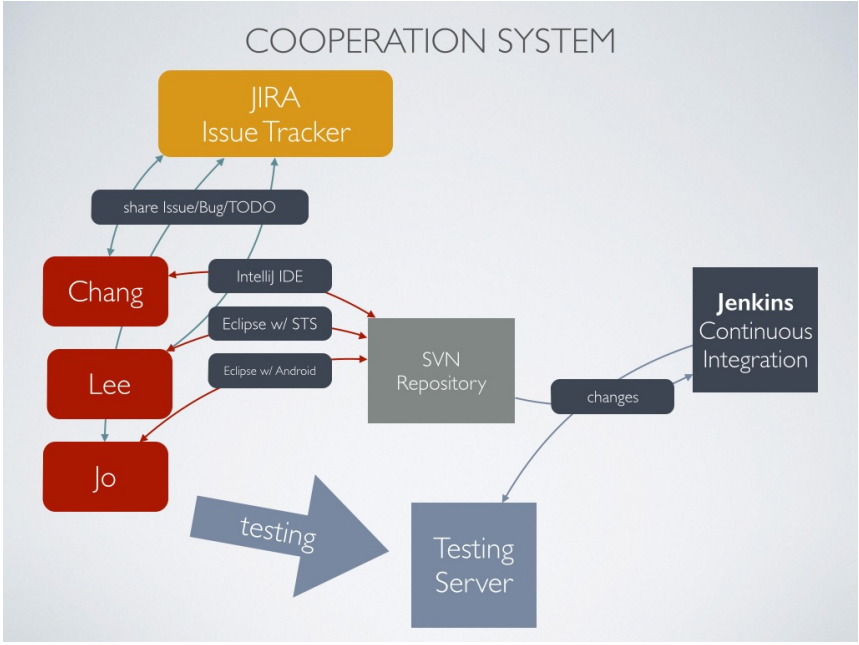
Here is what Facebook API we will use.

- Login
- Publishing : user's game activities
- Canvas
- Friends

Team Roles

Name	Part
Ikwhan Chang	Facebook In-App Game(using Adobe Edge or Unity Engine), iPhone Apps
Sangtae Lee	I/O Server, Game Server(Main), Database
Dongkyoung Jo	Hybrid Client, Game Server(Assistance), Facebook API

Team Cooperation System(based on Agile development)



Schedule

Week	To-Do
Week 1	Project goal setting,
Week 2	Propose project
Week 3	**Design**: User Interface, System Architectures, form of I/O JSON Data **Setting** : Subversion, CI(Continuous Integration) using Jenkins CI, IDE
Week 4	**Design**: I/O JSON Data(Transactions) **Setting** : WAS, I/O, DB server
Week 5	**Jo** : set a canvas **Lee** : developing default I/O server environment(thread, sockets..) **Chang** : Setting facebook app environments
Week 6	**Jo** : sync with I/O server **Lee** : sync with client **Chang** : game class design and implementation
Week 7	**Jo :** Facebook oauth **Lee :** sync with I/O server **Chang :** implementation to character's moving
Week 8	Mid-term Examination

Week	To-Do
Week 9	**Jo :** implementation to default client UI **Lee :** game connecting between user1 and user2 **Chang :** implementation to default game UI
Week 10	**Jo :** implementation to remote control **Lee :** game controlling(rule) **Chang :** implementation to friend's list
Week 11	**Jo :** implementation to remote control **Lee :** JSON transactions for remote control from client **Chang :** implementation to character's remote moving
Week 12	**Jo :** implementation to remote control **Lee :** JSON transactions for remote control from client **Chang :** implementation to character's remote moving
Week 13	First Demonstration and feedback
Week 14	First Revising our project
Week 15	Final Demonstration
Week 16	Final Examination.