Bursting the Mobile Bubble: Monster Jack

Stefano Martinallo
Chalmers University Of Technology
Campus Lindholmen
Lindholmen, Gothenburg
+031-764 70 00
stefano@chalmers.se

Chee Zhiquan
Chalmers University Of Technology
Campus Lindholmen
Lindholmen, Gothenburg
+031-764 70 00
zhiquan@chalmers.se

Tan Wei Ting
Chalmers University Of Technology
Campus Lindholmen
Lindholmen, Gothenburg
+031-764 70 00
weiting@chalmers.se

Li Xiameng
Chalmers University Of Technology
Campus Lindholmen
Lindholmen, Gothenburg
+031-764 70 00
xiameng@chalmers.se

ABSTRACT

With the introduction of mobile phones, people are able to stay connected remotely in a virtual reality. Yet, as such virtual connection improves, a real life interaction between people recedes. The idea of bursting mobile bubble is to make people step out of their comfort zones in the virtual reality to interact more in the human world. Bursting of the mobile bubble thus aim to allow communicate in real time within a given space, with the help of mobile devices, hence generating a collaborative and co-located experience that is shared among participants. In this paper, we will be presenting the concept of Monster Jack Game, the use of co-location to bind different phone users together, potential for face-to-face interaction and the app design process. Last but not least, we will be discussing some issues that will be part of future works due to time constraint on current project.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]:

1. Objective C - Main infrastructure of the application
2. C++ - Fetch, read and save data file
3. PHP - Receive hunter's party request and send query to alter entry in database
4. MySQL - Database to temporarily store data and location of hunter
5. JSON - JavaScript Object to port data from Web to iOS

General Terms

Documentation

Keywords

Collocation, Co- experience, interaction, mobile app, multi- user, mobile bubble, IOS, augmented reality.

1. INTRODUCTION

This is the generation that mobile devices are commonly owned by millions of people. They are connected in a virtual reality from different remote parts of the world with the help of the mobile technologies. As technology advances, we have undeniably improved social connection virtually, but at the cost of real life interactions among people. It is usual to see a group of people sitting together without physical interaction but looking at their own mobile phones instead. However, we have also realized that some phone users do interact with the people they are physically with, but in a virtual world, such as posting photos or status on Facebook and tagging the people involved (who are the ones at the same location). We see a potential to change the social behavior since they do interact with one another, just in different "places". Our main purpose of this project is to reconnect people in the real world by making the application run inline with reality in real-time to encourage face-to-face interactions. Our project can be thus seen as a 'service' to aid people who are in co-located space communicate physically as the use of the app will require some form of exploration in real spaces and social interactions to work.

This project will be in the form of a game app as we like to explore the idea of games that can be played anywhere and anytime as what Benford said "A new generation of entertainment technology takes computer games to the streets—and ultimately beyond" (Benford et al 2005, p. 1). Besides we would like to make a progress in game, that is beyond digital online games, which is a real-time game, that involves both the digital aspect of online digital games, but also the physciality of traditional games. With the aim of bursting mobile bubble, we came out with a mobile app named *Monster Jack*. Our app is designed to motivate people in the co-located space to be bound together in order to achieve a common goal. The app is designed in such way that some of the tasks can only be done when a group of people gather together and play at the same time.

2. Concept of Game

Monster Jack is an iOS game application game that allows players to hunt for monsters in an augmented reality, capture and store them in their libraries for future use. This hunting game can both be single player mode or multi-player mode, which the latter will be discussed further in the potential it has for face-to-face interactions. Like almost all application game, Monster Jack requires players to first create an account like name and a profile picture to start. When this is created, the profile will be basic, with no experience points and no monsters being caught yet. The player can then tap on the Monster Collection in the tab bar to view the whole collections of monsters he or she has to catch to proceed to the next level of the game. (Unfortunately, the game app is still a prototype in this phase, the creation of more levels to continue the game will be part of future work.) When player click on a specific monster name, it brings up the view of the details of the monsters, which includes a silhouette of the particular monster, the name, the experience points of hunter/player needed, the number of players needed to find the monster and the color zone to find the monster. If the player click on Hunting in the tab bar, the screen shows a camera view. Players will then have to point the device to a color which is the color zone that the particular monster resides in. This camera view will detect the color and the specific monster will appear and move around. Players will have to capture it to store in their monster collection. Once caught, the monster collection will no longer have a silhouette of the monster but the actual image of it. Besides playing on the single player mode, players can also choose to go into group hunting mode, which allow more players to play together and ability to catch different monster.

Use of co-location for multiplayer mode

When there are more players in the same location, they can create a party with 2 or more players using similar GPS coordinates. The application will allow them to individually send a request to server to create a party with a defined number of players, where they can then hunt for more powerful monsters as a group. The server will then check for duration of 3 minutes if the number of players that request for a party is as the same number of participants defined in the party. If 3 minutes is up and the number of collocated requests do not match, the party will not be formed, and vice versa. However, from the player's perspective, they do not see the request being sent to server only an animation of a bouncing circle over the player's profile picture that meant that the request is being processed. This timed request will make players want to find out the real player behind the mobile devices, and encourage them to join the party within the time limit.

Potential for face-to-face interaction

This function of group hunting allows players whether or not related, but situated at the same location to play the game together since they can achieve common goal and hence facilitate real-life interactions. When a party is created, players have to be close and stay connected to play the game. They will then communicate in real life to strategize on their moving patterns to stay connected in the party. Hence, this will require them to know who are the different players in the parties, hence encourages interaction. They also have to communicate through the game to discuss where to move to. New parties can also be created, with new participants when the GPS coordinates of users change. In the next phase of the prototype, we want to be able be enhance the gameplay by having more game levels so that the objective is strong and can function like a real game.

3. Design process

The initial phase of the project was the brainstorming. All members presented several ideas about the project and eventually after taking into considerations of feasibility aspects and degree of interest, the final idea was selected. Based on the raw idea of catching monsters in an augmented reality, a few interesting ideas such as creating a party to hunt for monsters, was added to make the app more related to connecting people and bursting the mobile bubble. After the idea generation process, flowcharts were created to ensure the dynamic flow of the app, both for the users and the structural flow for the programmers. Several sketches of the user interface were done to consolidate the main ideas. A tab view was the best solution to display the various details to the players, such as their profile with their contents and monsters they have or have not caught, the hunting page, and the page to create a party and hunt with other people. The most challenging aspects were creating a network so that users can connect together and hunt as a group. The initial idea was to be able to view players near you, and then select intended participants to invite them to join a group and hunt. Various ways such as the WiTap, iBeacons and Bluetooth were tried but did not work. Eventually the solution was to make use of GPS coordinates and send this data to a server to check this result, and create the party. However this would mean that the players are no longer able to see the players in the nearby areas, instead of choosing who to create a party with, it is now assign by the server. This server also stores the data of the all the monsters available to catch. In the hunting page, the camera will detect the RGB colour of the area that the camera is pointed at and takes an average color of the image, and then show the monster that reflects the particular average color. As the camera takes an average color, it can only show one monster each time on the screen. Interface design and debugging process are parallel conducted. Initial idea to enhance connections between players was to allow the clicking on other players' profile and subsequently adding them as friends. However, this would mean that the server will have to store all these datas, and in this current phase, the server used is not able to hold so much information, thus this aspect will be left for future work.

3.1 Frameworks used

1. UIKIT - User interface and animation of monster in hunting mode
2. GPUImage by Brad Larsson - Fetching average color (to determine which monster to be shown) from live camera input
3. Core Location - Retrieval of GPS Co-ordinates

3.2 Technical features

2. Hunter’s Data, Monster’s Data and Collection

    2.1 Data is stored into .dat file inside iOS. Open and read using C++ codes. The data contains details about the hunter such as name, experience point and the monsters he/she has.

    2.2 Monster’s Data and Collection is a data model to specify details for the monsters

3. Hunting Mode

    3.1 GPUImage is used to activate live camera input and also to calculate the average color of the image in a frame, returning RGBa values.

3.2 If RGBa values show red, monster that is associated with red appears

3.3 User can tap on the screen to show a focusing square and the monster must be inside the focusing square to be able to capture it using the capture button.

4. Group Hunt

4.1 Core Location is used to gather current GPS coordinates of the user.

4.2 The user sends the GPS coordinates alongside with his/her profile details such as name and experience to the web server.

4.3 The web server receives the data and helps form a party connecting the nearby users.

4.4 The web server utilizes PHP and MySQL and transfer data to IOS using JSON

4. Future Development

The future plans for Monster Jack would include adding more monsters for the players to catch. The other major game feature added would be the handling of monster. If the monster is not handle well, the monster will automatically leave the collection. Handling of monster includes taking care of it, feeding it, playing with it, fighting other monsters with it. The monster can have levels to show how powerful it is and also a specific set of skills. The food to feed the monster can be obtained by the same method as capturing the monster (focusing on a screen to reveal the object) Players can use their monsters to fight with other players with their monsters.

The hunting mode will also be updated, possibly the application can be able to detect two different colors and thus showing more than one monster. The player can also design their own monster and have a chance of featuring it in the game. Another possible update would be the trading of monsters, which include game currency in the game.

5. Conclusion

The application was initially more focused towards the backend side, which is the coding part of the application, and then later on the design was focused and implemented. The coding part requires a lot of research especially the networking section. This game also require certain framework to work, the framework that was used for the live camera input is a third party framework that is not official from the Apple Developer, however it is necessary for the feature to work.

6. REFERENCES

[1] Benford,S., Magerkurth C. and Ljungstrand, O.(2005) Bridging the PHYSICAL AND DIGITAL in Pervasive Gaming. Communications of The ACM, March 2005/Vol.48, No.3,pp.54-57