Guess Who?
A smartphone version of a kids game

Bonotto Sara
Laurea Magistrale in Informatica
Università di Padova
sara.bonotto@studenti.unipd.it

Ciman Matteo
Laurea Magistrale in Informatica
Università di Padova
matteo.ciman@studenti.unipd.it

***Abstract*—In the last decades mobile phones have become not only an instrument for make calls or send messages, but even a great instruments for games, relations and other every-day stuff. Mobile applications, especially games, are nowadays part of the life of every smartphone user, and are slowly substituting old table games with a mobile version that gives the possibility to play not only with people in the same room, but event at the other side of the world. To start moving in this direction, we propose a smartphone version of the box game "Guess Who?" by Hasbro Games.**

I. INTRODUCTION

For every person, the childhood is remembered especially for what we can call "box games", games played on a table with multiple players, usually stored in a box with the playground, cards and so on. In the last decade, what have seen an increment of the number of smartphone distributed in the world, event to the younger generations that always first has the first contact with a mobile device. The result of this process is that young guys start early to stop playing box games, preferring smartphones, computer and other devices to games with other guys. This could be a big problem for the next generation, causing problems even in interpersonal skills or loss of the reality.

What we do is to insert inside the unstoppable process of smartphone distribution even to young guys, an old style game that can be defined as "in step with the times", working on a new different and more electronic environment.

A. Game Description

The game of "'Guess Who?" Is very simple and is mostly based on the spirit of observation of the players. In fact, it has as main purpose the exclusion of the characters in the game based on questions about physical characteristics, accessories, etc.. the opponent. The game is played with a number of players equal to two. With the original game, the game begins when the two players have chosen a random card from the deck among those of the characters.

Then the two players are beginning to ask questions in order to exclude most people possible with the minimum number of questions in order to identify as quickly as possible to the character portrayed in the paper that the opponent has drawn from the deck. At each turn the player who asked the question cannot make another one until it receives the answer from the opponent, and answered to the question made by the other player.

The first player to guess is the winner. In the case in which a player gives the wrong answer to the game, he automatically loses the game.

The game was made to try to comply as closely as possible to the original game. Of course, some minor changes were inevitable.

The main change that was necessary is to establish a connection between two devices in order to allow it to carry the game as it is no longer to have a real poker table, but virtualized. In this way, players can also be found a few meters away. The connection choose is Bluetooth.

In this case the first operation that is performed when the two players want to start the game is the creation of the connection. By clicking on both "New Game", the application tries to activate the Bluetooth and then gives the possibility to be discoverable or to search for a player.

Once the connection has started, the game continues in the classic manner provided by the game itself except for the settlement of claims. To ensure that the basic rules of the game are respected, that is to be asked only questions that can be answered with "yes" or "no" answers, questions are being semi-composed, as explained later.

In this way the player will simply choose a physical feature, accessory, etc.. and the question is formulated and sent automatically to the device opponent.

B. Method of game and players

The game does not provide other conditions than the classic board game.

The number of players expected for the game are two.

II. USER INTERFACE AND USER INTERACTION

A. User Interface

What has to be clear when you transport a table game to a smartphone game is that it is impossible to reproduce all the interaction that can be between the players of the game. For

example, the simple questions “Have you complete?” or “Can I make my play?” is something that, in a mobile environment that gives the possibility to unknown players to play eth each other, is not reproducible.

For this reason, what we decided to implement is a not invasive GUI, that show messages only with an explicit request by the user. For example, the first player chooses the question to made, and the second has to answer to it. When the first player completes his task and the application sends the question to the other player, what we want to eliminate is the event that a dialog comes up as a pop-up and stops what the user is doing (for example, eliminating some faces). So, what happens is that when the application receives a message, depending on the state of it (Waiting for a question, waiting for an answer and so on), it activates the correct button on the GUI, and nothing more. The player will decide alone and with his timing when to show the message.

The only exception of this rule is when a player tries to guess the right face. Since this is the last action of the game (if he guess it will win, otherwise he will lose), the message received by the other player has a fixed prefix that indicates that the other one is trying to guess. In this situation a dialog will be shown with the result of the game.

B. Question Composition

Another problem that we had to resolve was the questions that a player can made to the other one. In the real game, the possible questions that can be done are theoretically infinite, or better, can cover a big number of different combination.

With this basis, we had two different possibilities to implement:

- Give the possibility to the player to make his question by hand, writing it on a text box and send it to the other one
- Give the possibility to make only a predefined set of question inserted at application level.

The first option could be the easiest and the fastest one, but has a big drawback: give the possibility to a player to write what he want can give the possibility to “not correct” player to write something that is not inherent to the game, causing the decline of the funny of the game.

For this reason we adopt a predefined set of question. The possible questions are divided into some different macro-categories: hairs, beard, eyes, moustache, sex and particular signs. After choosing this macro category, different options, depending of the first category chose gives the possibility to the payer to complete the question (for example, he could choose Eyes – Orange to ask if the character has orange eyes).

After the confirmation by the user, the *QuestionComposer* class takes the two options and composes the question with the Italian grammar. This is done providing at the application all the string necessary to construct the question, and the task of this class is to compose them correctly. For example, if the user chooses Eyes and Black, the *QuestionComposer* class knows that for Eyes the first word of the question is “Ha”, after it will attach “gli occhi” (that is the corresponding string of eyes in the question), and after “neri” (that is the corresponding string for Black. What we have at practically is a correspondence between an option and the string to insert in the question when we compose it.

III. DESIGN AND IMPLEMENTATION

A. General Architecture

The application has been developed using Android 2.2. This choice has been made because that version is the most supported version by now, and because our target wants to be wide as much as possible.

The general architecture of the application is composed by:

- *UI Manager*. It manages user interaction on the screen, providing method for selection of faces, question composer etc.
- *Control Manager*. It manages the different turns of the game and the timing of it.
- *Network Manager*. It manages connection with other phones. The basic idea is that it has to be network independent, and provide the same feature with every type of connection. At this time only Bluetooth is supported, but it could be extended with WiFi or UMTS/GPRS support. It takes care of the connection set-up, message exchanges and eventually connection-lost.

Thread synchronization and other general stuff are made by the main thread that is the central point of the game that controls all other part of the application.

What we have decide for the application, is that it is not the application that gives the timing of the game, but are the two players that manage the timing. This means, for example, that when a user makes a question to the other one, the one that receives the question will not receive immediately a message popping-up that shows the question, but only a button that becomes enabled, saying that there is a new question to answer. This avoid annoyance to the user, for example we avoid the situation where one user is eliminating some faces, and a pop-up comes out for the question of the opponent, probably causing to the player the loss of the last question he has made and the answer received.

B. Bluetooth Connection

In this section we are going to focus on the Bluetooth connection management, first of all describing the protocol in general, and after giving some implementation details.

First of all, our choice for Bluetooth has been made thinking about the contest where our application has been developed. In Italy, free WiFi connection is not very diffused, and it can be very expensive. For this reason, Bluetooth is the only choice to give to the user a free way to play with friends.

The communication protocol developed can be divided into two different phases:

1. A handshake phase

2. A communication phase

In the handshake phase, we have two different devices that want to connect with each other. What is necessary is that Bluetooth is activated for both devices, otherwise, obviously the game will not start. To start a communication, one device has to become the server, and start to wait for an incoming connection, and the other one has to become a client, looking for a server. When the two devices discover each other, they try to complete a connection and get an InputStream and OutputStream with which get or send packages.

For our purpose, every device send, in every situation, only strings. This simplifies too much the management of the different phases of the game, but on the other side we cannot provide multiple language support, and gives the possibility to use at the same time different languages. At this time, only Italian is implemented.

During the different phases of the game, the BluetoothConnectionManager is responsible to send or receive messages. Due to the phase-organization of the game, when a device receives a message from the other one, is able to understand what that message means and activate the correct button in the GUI or to show the right dialog.

It is possible that during games some problems happens and the connection can be lost. In this case, our BluetoothConnectionManager is able to understand if something happened, and stop the game avoiding any crash of the application. In future work, we explain an idea to better manage this situations.

IV. FUTURE WORKS

The game that was made is in basic form in the sense that the minimum features have been implemented, but still allow the smooth running of the game. Future extensions can be implemented in order to embellish and complement the game and its options.

A. Different connections modes

At this time, only Bluetooth is supported as way of connection. But with our organization, is possible to add new different connection ways, such as WiFi, to provide more new user experience and to connect even people too far from each other.

B. Not random character choice

At the beginning of the game, each player gets his character that the other one has to guess. To improve our game, it could be given the possibility to choose alone the character, and not randomly

C. Multilanguage support

At this time, only Italian in supported as language of the game. What we could do is to add support for other languages, for example English, and provide the possibility to play with different languages for the two players. The biggest problem in this situation is that the question is made at asker-side and not at receiver side, and because grammatical rules of Italian are different from English grammatical rules. The biggest problem is in composing the right question in the correct manner.

D. Automatic save of the game

At this time, when there are connection problems, or a player wants to stop the game, the game is interrupted and nothing more. A great improvement should automatically save the state of the game (characters excluded, turn of the game) and restart the leaved game next time that the two payers meet themselves and start a new game.

E. Difficulty levels

To make the game more challenge, it could be inserted different difficulty levels, starting a game with different number of characters depending of the difficulty chose (Easy level less characters than Hard level).

F. Statistics

Insert a sort of statistics or history of the player, providing information about victories, losses, number of games and so on.

G. Grafical improvements

At this time the GUI is very easy and not sophisticates. To give a better user experience, there can be added different graphical improvements in order to give a better game experience to the players.

V. CONCLUSIONS

What have produced is a close reproduction of the table game “Guess Who?” with all its rules and options. What our application has in more respect to the table game, is the possibility to help children with some mobility problems or deaf without speech children to play with it. In fact, eliminating all vocal interaction between players, this game can be played even by deaf without speech children with children without this problem, eliminating the necessity of the presence of another person that made as a translator for the two players.

REFERENCES

[1] Android Developers Resources: http://developer.android.com/index.html

[2] Android consortium official website: http://www.android.com

[3] Beginning Android 3, Mark Murphy, Apress, 2011

[4] Hello, Android – Introducing Google’s Mobile Development Platform, Third Edition, Ed Burnette, The Pragmatic Bookshelf, 2010