***JAYPEE INSTITUTE OF INFORMATION***

***TECHNOLOGY, NOIDA***



System and Networking Programming Lab

Minor Project II

End-Semester Project Evaluation

Project Title- **UNO** (Multiplayer Game)



Submitted to- Submitted by-

Gagandeep Kaur Chetna Gupta (11103462)

K. Rajalakshmi Palak Anmol (11103522)

Shreya Goyal (11103572)

Batch- B3

**Content**

1. Introduction
2. Literature Survey
3. Work Distribution
4. Software and Hardware Requirements
5. Problem Definition
6. Project Objective
7. Algorithm Used
8. Implementation
9. Test Cases
10. Risk Analysis
11. Conclusion
12. References

**Introduction**

The key objective of our project is to develop a multiplayer game using latest graphics for best architecture and user friendly look.

UNO multiplayer networking game is an android app developed to simulate the real time experience of playing the interesting card game anywhere and anytime with live opponents.

UNO is a card game based on the objective of being the first to rid yourself of all the cards in your hand and triumphantly shouting, "UNO".

**Literature Survey**

Initially games were developed by computer scientists primarily to entertain themselves and their colleagues. In late 1970s, with immense increase in computing power, graphical capabilities and mass production of personal computers, games development became an industry. Competition between developers and rapidly increasing expectations of players forced developers to make games as computationally complex and as graphically appealing as was allowed by the average hardware of the end user.

Development of a computer game requires collaboration of developers in many areas including programming, 2D and 3D art, music and sound effects, game design and testing. Programming tasks while developing a game are divided into a few areas including AI & game play, 3D graphics, networking, tools and physics.

The common challenges we will be facing are:

* Games must provide feedback to the player immediately, without any noticeable delay.
* Making it possible that the game can work on any of the OS like Windows 7, Windows 8, Fedora, and MAC etc.
* The need to minimize traffic since many users may pay for their Internet connection based on the amount of data they download or upload, it is vital that the amount of data sent over the network by the game is minimal.
* During a network game any client can unexpectedly loose Internet connectivity. If this client was also acting as the game server (host), all players will get disconnected and the game will be lost.

**Work Distribution**

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Work Assigned** | **Hours spent per week** |
| **Chetna Gupta**  **(11103462)** | |  | | --- | | * Implementation of graphics and User Interface. * Study of hardware requirements * Partial Implementation of networking * Testing | | **10** |
| **Palak Anmol**  **(11103522)** | * Study of software requirements * Implementation of game coding * Implementation of networking * Apply algorithim | **10** |
| **Shreya Goyal**  **(11103572)** | * Implementation of animation in the user interface * Partial implementation of game coding * Partial implementation of networking * Testing | **10** |

**Software Requirements**

* **Socket Programming (Networking) in Java** - Java is a [computer programming language](http://en.wikipedia.org/wiki/Computer_programming_language) that is [concurrent](http://en.wikipedia.org/wiki/Concurrent_computing), [class-based](http://en.wikipedia.org/wiki/Class-based), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](http://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

The endpoint in an interprocess communication is called a *socket*, or a [network socket](http://en.wikipedia.org/wiki/Network_socket) for disambiguation. Since most communication between computers is based on the Internet Protocol, an almost equivalent term is *Internet socket*. The data transmission between two sockets is organised by [communications protocols](http://en.wikipedia.org/wiki/Communications_protocol), usually implemented in the operating system of the participating computers. Application programs write to and read from these sockets. Therefore, network programming is essentially socket programming.

* **Android SDK** - A [software](http://www.webopedia.com/TERM/S/software.html) development kit that enables developers to create [applications](http://www.webopedia.com/TERM/A/application.html) for the [Android platform](http://www.webopedia.com/TERM/A/Android_platform.html). The Android [SDK](http://www.webopedia.com/TERM/S/SDK.html) includes sample projects with [source code](http://www.webopedia.com/TERM/S/source_code.html), development tools, an [emulator](http://www.webopedia.com/TERM/E/emulator.html), and required libraries to build Android applications. Applications are written using the [Java](http://www.webopedia.com/TERM/J/Java.html) programming language and run on [Dalvik](http://www.webopedia.com/TERM/D/Dalvik.html), a custom [virtual machine](http://www.webopedia.com/TERM/V/virtual_machine.html)
* **Eclipse** – Eclipse, free and open source software, is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications in other languages like C, C++, PHP, JavaScript, Python, Ruby etc.

**Hardware Requirements**

* Processor: Intel(R) Core i5 CPU
* RAM: 4.00 GB
* System Type: 32-bit Operating System, x64-based processor
* Intel® HD Graphics with following adapter information
* Chip Type: Intel(R) HD Graphics (Core i5)
* DAC Type: Internal
* Samsung Galaxy Tab 3 (Android version 4.4 )
* Samsung Grand (Android Version 4.2)

**Problem Definition**

The game will start when server identifies more than 2 connected clients.

In the card game Uno, the objective is to be the first player to get rid of all of the playing cards. After a player plays all their cards, the other players count the number of points pertaining to the values of the cards in their hands.

Features included:

* Players can interact with each other using In-game text messages facility, thus enhancing the social aspect.
* Leader board is maintained and shared with each player so as to display their scores and rankings.

**Rules:**

The deck consists of 88 cards, of which there are 22 of each colour (red, green, blue, and yellow), each colour having two of each rank. The ranks in each colour are 0 to 9, "Draw Two" (the last of which is classified as "action cards"). In addition, the deck contains four each of "Wild" and "Wild Draw Four" cards.

To start a hand, seven cards are dealt out to each player, and the top card of the deck is flipped over and set aside to begin the discard pile. The player to the dealer's left plays first, unless the first card on the discard pile is an action or Wild card. On a player's turn, he/she must do one of the following:

* play a card matching the discard in colour, number or symbol
* draw the top card of the deck

If a player chooses to draw the top card of the deck, and that card is playable (it matches the discard, or is a playable wild card), then the player may (but need not) immediately play that card.

Play proceeds clockwise around the table.

If it happens that you are only left with one card in your pile, you are required to yell “Uno”. If you fail to do that and you get caught, you will have to pick two more cards from the deck. But if you are successful at yelling “Uno” at the first instance without getting caught, you will be safe and you will not need to draw additional cards.

When a player sheds last card, the game is over and that player wins the play.

**Project Objective**

* The key objective of our project is to develop a multiplayer game using latest graphics for best architecture and user friendly look.
* Uno offers you both enjoyment and challenge and therefore you will be guaranteed of extreme fun. It fun-filled game that is suitable for people of all ages.
* The objective of [Uno online](http://unogame.co.uk/) card game is to get rid of all your cards as fast as possible.
* The game making experience will enable us to be familiar with various new game developing engines, networking engines and network programming challenges and also learn and understand various concepts of Java.

**Algorithms Used**

* **Multithreading:** Multithreading is the ability of a program or an operating system process to manage its use by more than one user at a time and to even manage multiple requests by the same user without having to have multiple copies of the program running on the computer.
* **RPC Details:** Remote Procedure Call or RPC is a way of calling a function on a remote machine. This may be a client calling a function on the server, or the server calling a function on some or all clients.
* **Master Server:** The master server is a dedicated server responsible for collecting the connection information (IP address and port) of all available game servers and providing that to the users.
* **Combinational Game Theory** – It is a branch of applied mathematics and theoretical computer science that studies sequential games , i.e two player games which have a position in which player take turns changing in defined ways to achieve a defined winning condition.

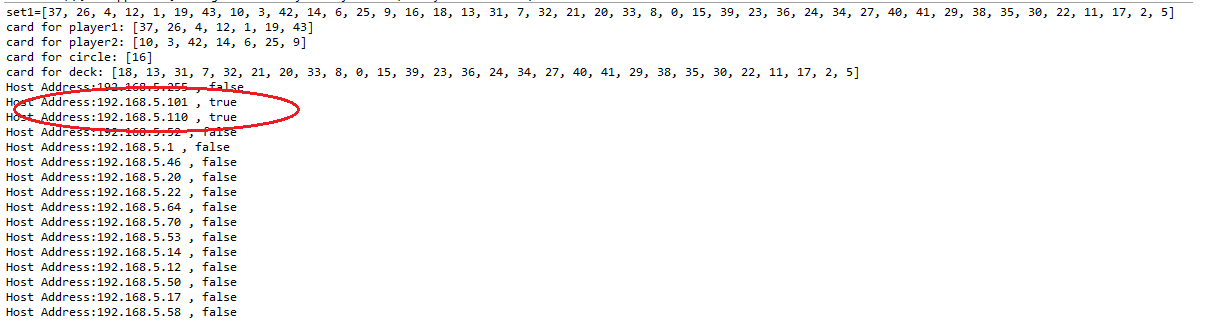
In our game a card is placed at the center according to which each player decides its move . if the card clicked by the player matches the given conditions i.e either colour of the top card or the face value of the top card matches,then the move is possible else player has to move another card. If the player is short of cards i.e he does not have a card to move he picks a card from the deck increasing the no.of cards by 1 in his hand. A player wins if he is the first to loose all his cards.

* **Transmission Control Protocol:** TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end.

**Implementation**

In our project client side code is written in Android and server side code is done in Java.

*Server Implementation-*

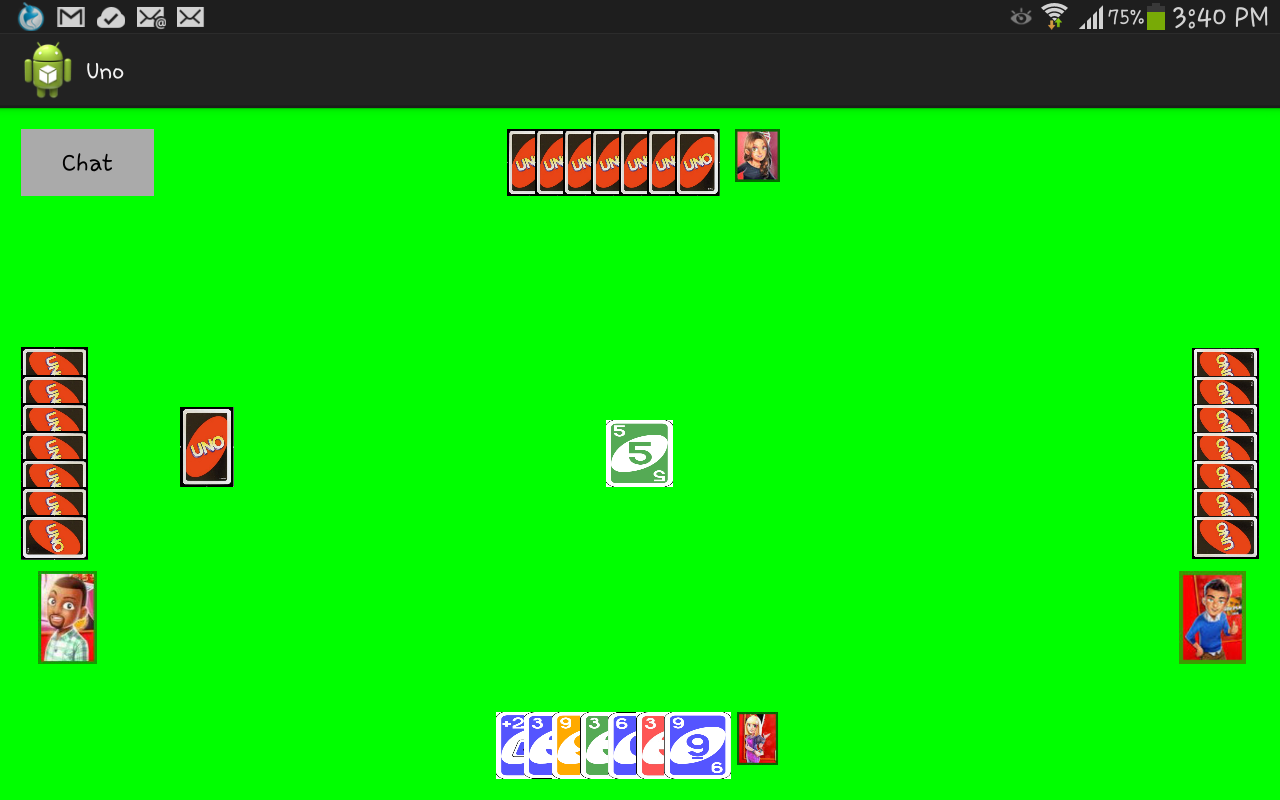
****

*Client Implementation-*

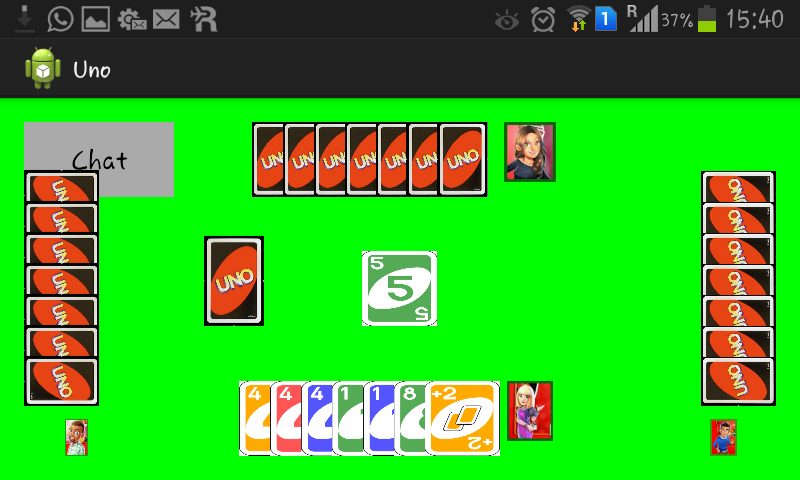
**Initial page**

****

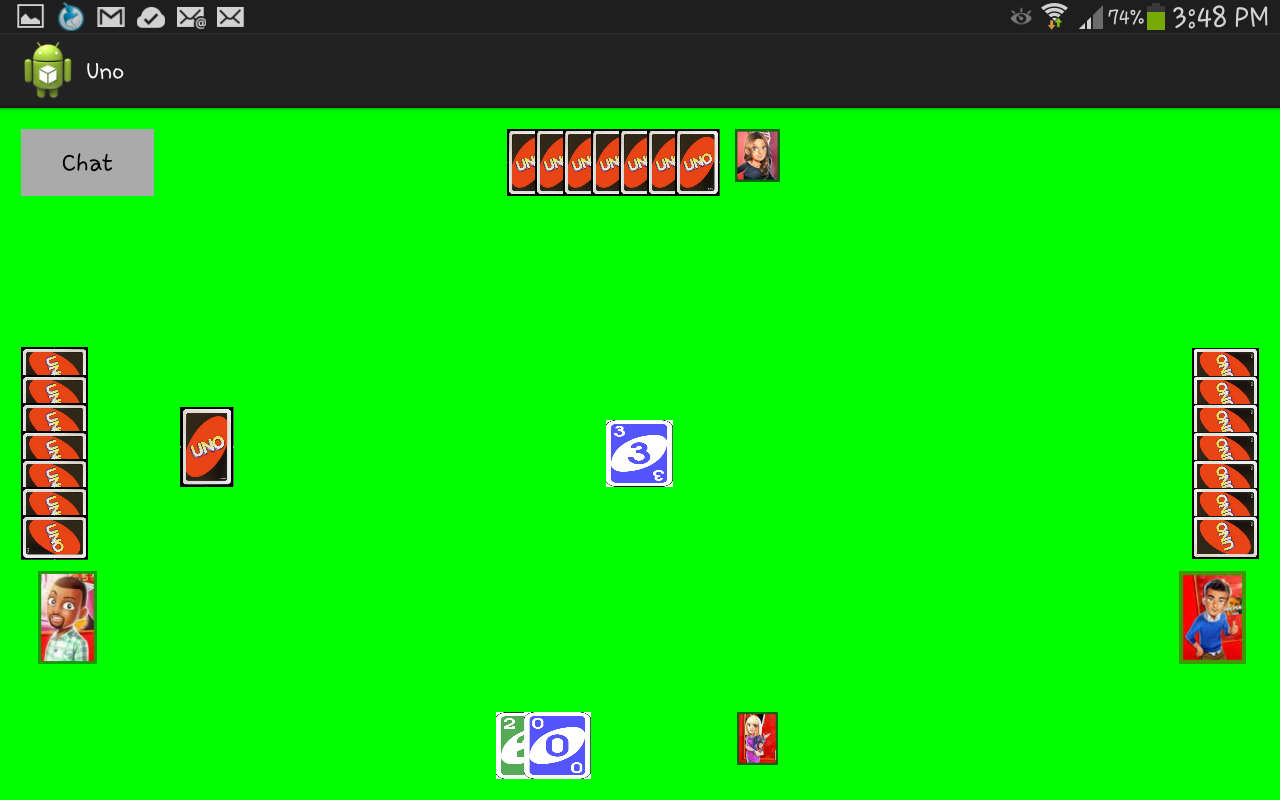
Player 1 (Initially)



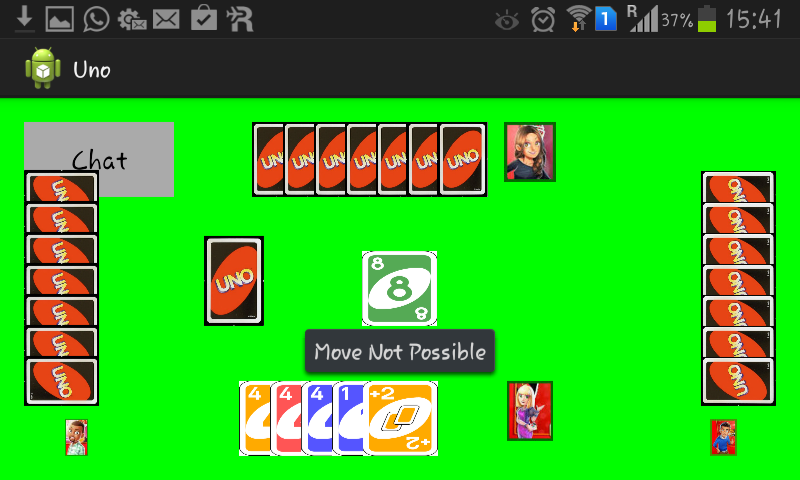
Player 2(Initially)



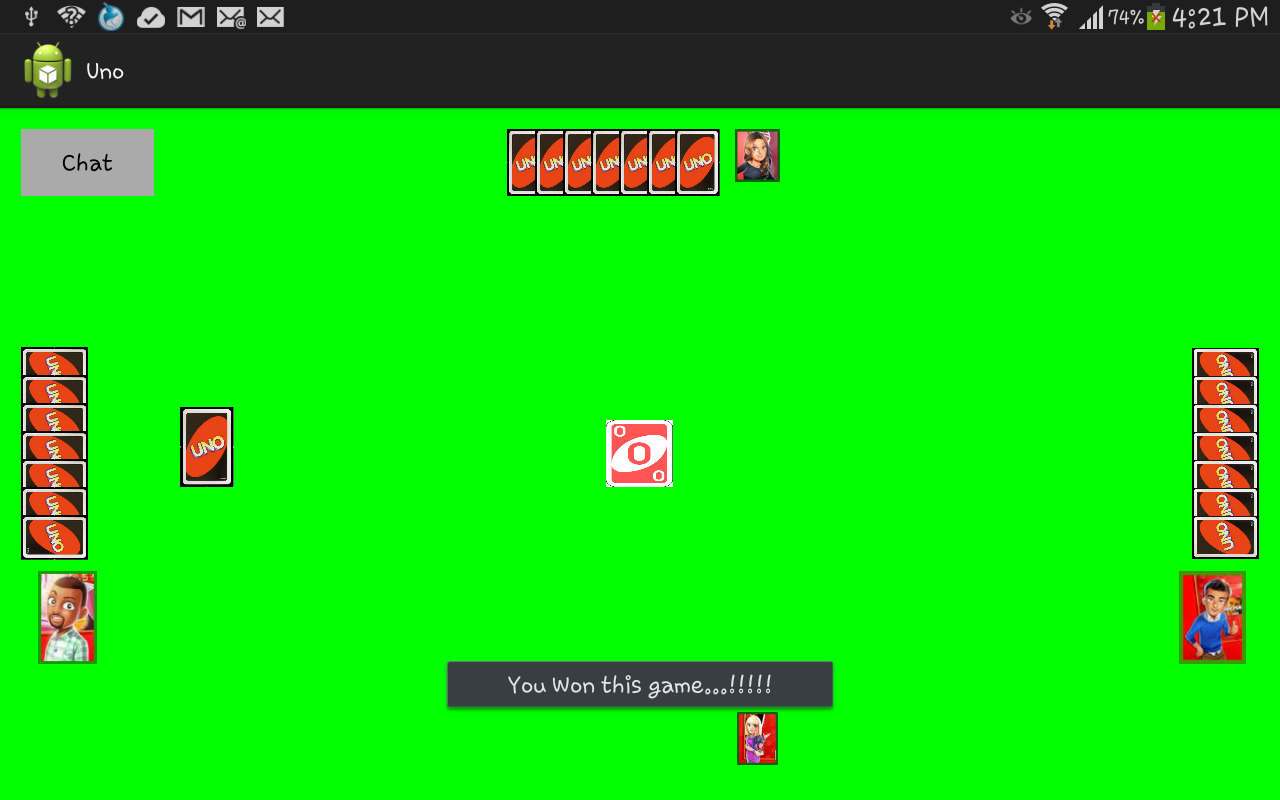
Player1(After removal of cards)



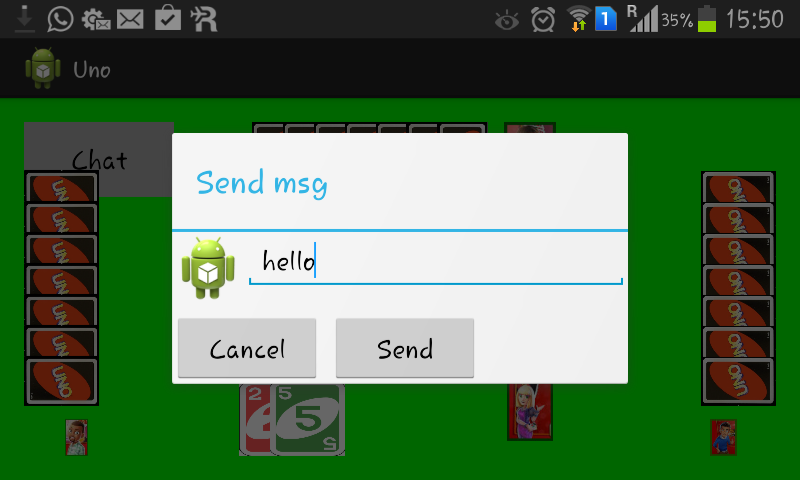
If move is not possible



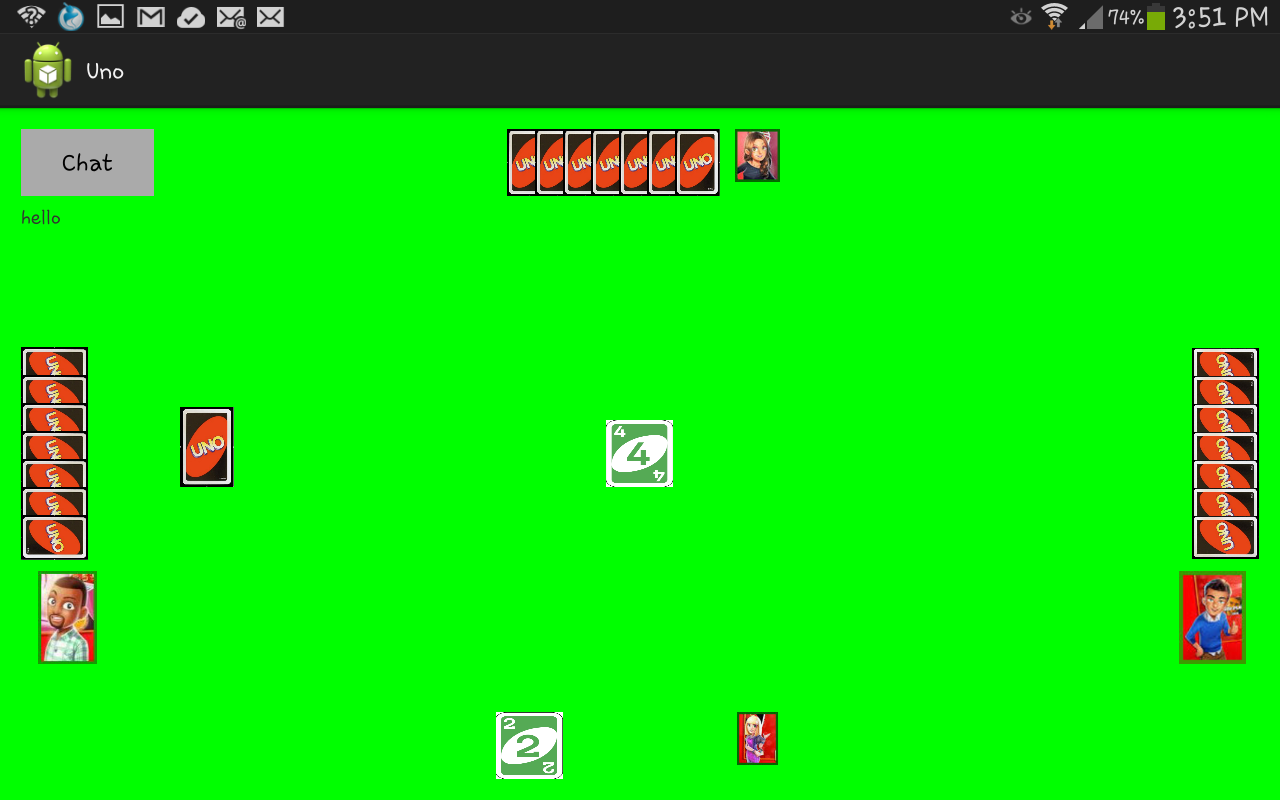
You Won



Chat (sender)



Chat (Reciever)



The following files are use for graphics, animation, game coding and networking.

**Server-Side Code**

* **StartServer.java** - This file searches for all the connected IP addresses and act as a main control for transfer of data among various clients and maintains there present information.
* **CardDistribution.java** - This file generates a random set of cards that needs to be distributed to each player.
* **Player.java** - It keeps the record of each player cards.
* **SendData.java –** It is a thread running on the server that sends the game data to each client.
* **ClsSocket.java –** It is a thread running on the server that receives the game data from each client.
* **ChatSendData-** It is a thread running on the server that sends text messages to the other client.
* **ChatClsSocket.**java – It is a thread running on the server that receives the text messages from one client and send it to another.
* **CheckSockets.java –** It is a thread running on the server that checks for all the available player(clients) in the beginning of the game.

**Client-Side Code**

**For GUI & Animation**

* **Splash.xml** - Start page graphics from where the player will login.
* **Activity\_main.xml** - It is used to provide graphics to the main file where player plays the game.
* **Sticker\_fall.xml** - it gives animation to the cards when game starts.
* **Dialog**.xml – It is used to give a dialog box for sending chat messages.

**For Game Coding & Chat messages**

* **Splash.java –** This file is the first page of the game
* **MainActivity.java** - This file is used to perform tasks when the client (player) clicks the card.
* **CardInfoBean.java** - This file returns information like colour and face value of the card clicked.
* **CardMapper.java** -This file maps each card with an id that helps us to retrieve information in MainActivity.java as which card is clicked.

**For Networking**

* **SendData.java-** It is a thread running on each client which sends the information of the card clicked to the server.
* **ClsSocket.java**- It is a thread running each client to receive the changed card information .
* **ChatSendData.java-** It is a thread running on each client to send the text messages to the server which in turn sends it to the other client.
* **ChatClsSocket.java –** It is a thread runningon each client to receive text messages from the otherclient via server.

**Test Cases**

The Android lint tool is a static code analysis tool that checks Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.

In our project we have used the Lint tool to find out the bugs in the client side coding of the game and improve them. We used the command

*lint –html testing.html Uno*

to perform the testing and stored it in the HTML file testing.html.

No errors were found in the final code of our program.

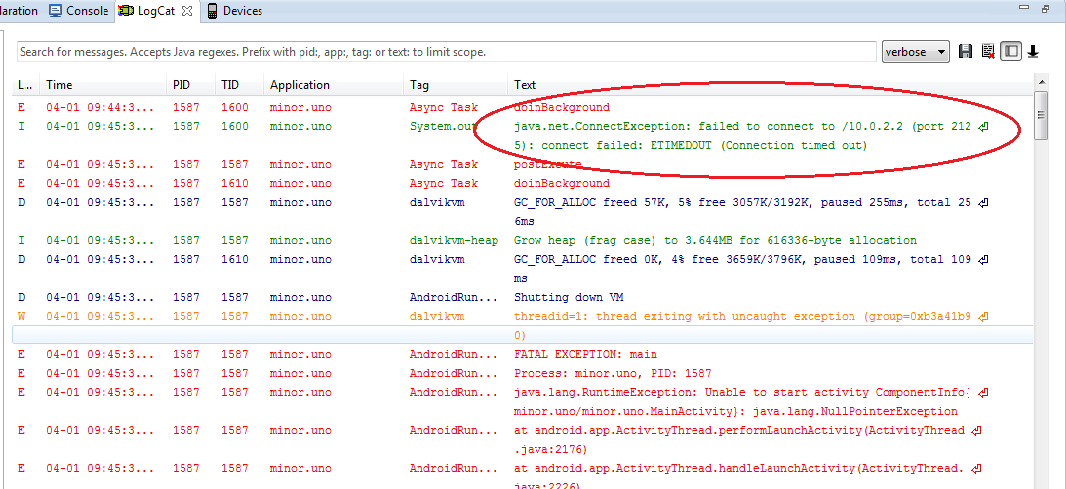
Following are the screenshots of the result-

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Test** | **Will Test be formed?** | **Explanation** | **Software Components** |
| **1. Requirements Testing** | Yes | It is required because it will let us know the needs of the project, which will eventually help us in the better development of project. | It is applied before the project is started on the overall project. |
| **2. Unit Testing** | Yes | It is the most important of all testing levels.  It will help in finding the bugs during the progress of the project, which is much more economical to fix as compared to that when done after completion of the project. | Unit testing will be performed on individual blocks of code in the starting of the project. |
| **3. Integration**  **Testing** | Yes | Integration testing will be performed to verify functional, performance, and reliability requirements placed on major design items. | It is performed after the unit testing, taking as input the modules that have been unit tested. |
| **4. Performance Testing** | Yes | Performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. | It is performed when a particular workload is being executed whose responsiveness needs to be calculated. |
| **5. Security Testing** | No | Security covers aspects like confidentiality, integrity, authentication etc. Since our project doesn’t involve private data of user so we don’t need this test. |  |

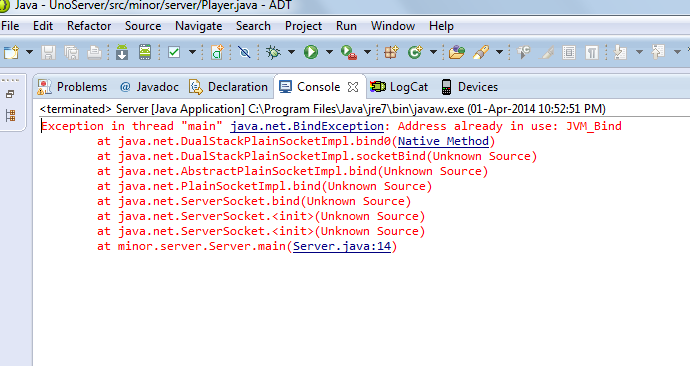
Errors encountered-

* Client Server Connection

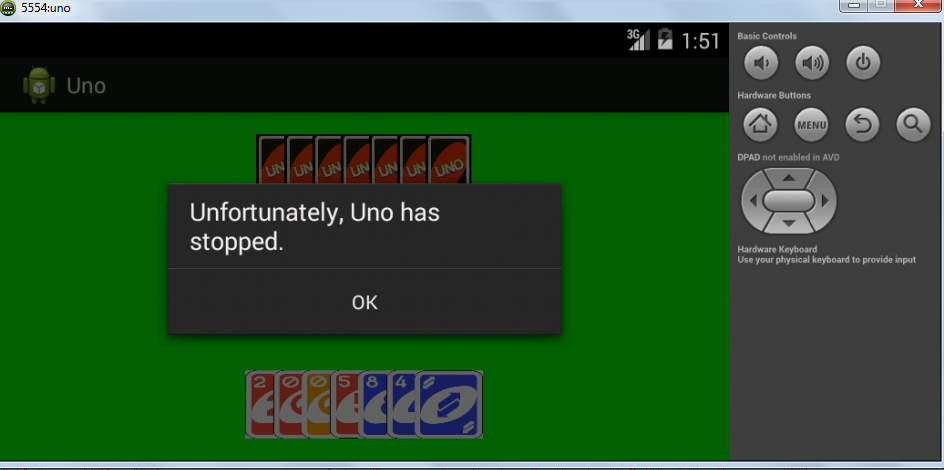
Exception was raised when connection was tried to establish using wrong IP address.

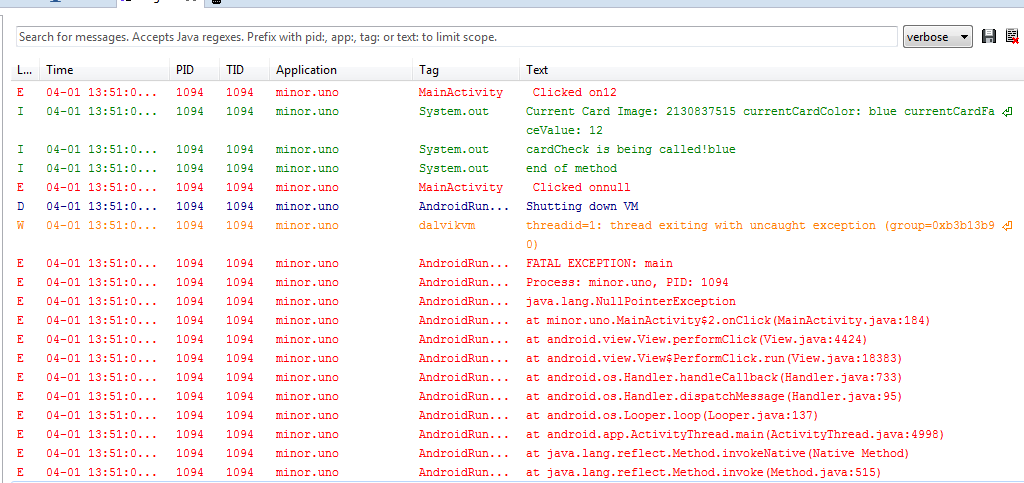


* If there are more than two clients for a connection, again an exception is raised.

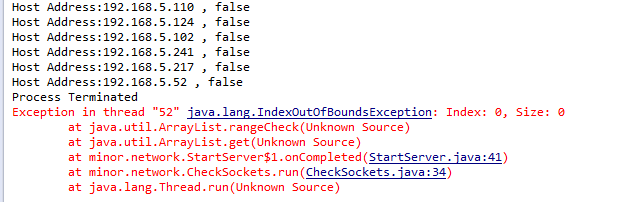


* NullPointerException is raised if a card, which doesn’t have id assigned to it, is clicked and then the application terminated.

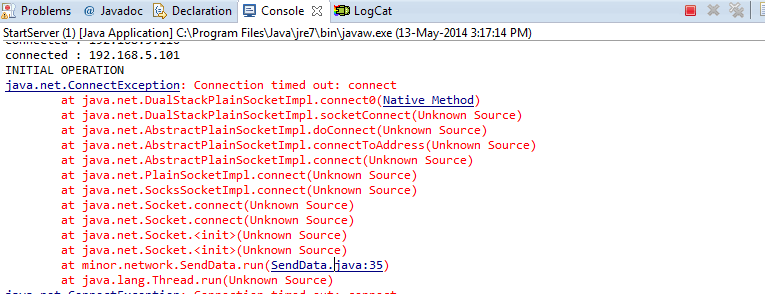




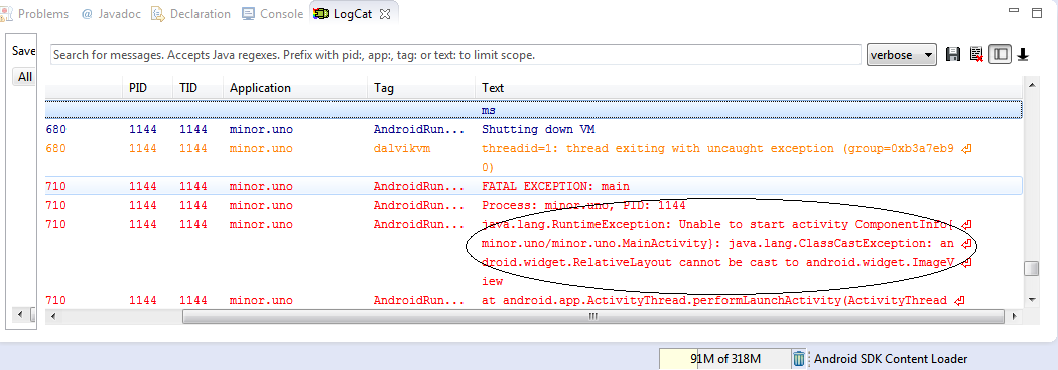
* If number of players available are less than two then the game will not begin and it will raise an exception

****

* If a player leaves the game in between then an exception is raised.

****

* An exception was raised when a conflict arose in the GUI part of the code.

****

**Risk Analysis**

Risk Analysis helps to mitigate the risks and highlight the potential problem areas of a project.

Following are the findings of the analysis for our project-

* If more than two players play the game at same time, the server will be busy and an exception will be raised java.net.BindException.
* The application does make the client system a bit slow and even unresponsive sometimes upon execution.
* There have been certain execution issues sometimes when the programmed application for client was invoked and made to execute first and after that server application executed. However, it always worked fine when executing server first and then client.
* Not meeting the optimum software requirements leads to a less responsive execution and sometimes some strange sudden termination.
* There might be some firewall issues, so before turning server on it must be disabled.
* The devices with API level less than 8 and more than 18 won’t support our application because the *minSdkVersion* and *maxSdkVersion* specified in the Manifest.xml file is set to 8 and 18 respectively.

**Conclusion**

In the end semester evaluation of our project, we demonstrated the final GUI of our project and a game for two players. Networking is used in the way that a client-server connection is established and server distributes the cards to the two players. On clicking a card from one’s own pile, player sends the card to the server where it is validated and if eligible for playing according to the rules it is played and displayed on both players’ screen by server.

Hence the game continues this way. The first player to loose all his cards is the winner of the game .

We also tried implementing a short version of In game chat facility which is used to send text messages to other player.

In the final phase we mainly focused on implementing the concepts of networking and socket programming besides making an effective GUI and integrating that with rules of the game and adding features like chatting.

Future Scope:

In addition to these features, voice chat and controlling movement of cards through voice commands can be added to this game to make it enhanced and interactive. Besides this game can be made platform and version independent so that it is available in devices that support various other OS like Windows, IOS.

**References**

**Web Links -**

* <http://www.tutorialspoint.com/android/>
* <http://developer.android.com/index.html>
* <http://www.tutorialspoint.com/java/java_networking.htm>
* <http://www.androidhive.info/2013/06/android-working-with-xml-animations/>
* <http://www.coreservlets.com/android-tutorial/>
* <http://docs.oracle.com/javase/tutorial/2d/index.html>

**Books-**

* Java by Deitel and Deitel