Tambola

Project report submitted in fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Satyam Aggarwal (161330)

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology
Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that this submission is my own work carried out at Octro Inc. from 3rd Feb 2020 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Satyam

Satyam Aggarwal, 161330

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Muneesh Garg

Mr. Muneesh Garg
Director of Engineering,
Octro, Inc.

Dated: 04-06-2020

Acknowledgment

It is my pleasure to express with deep sense of gratitude to Mr. Muneesh Garg and other members of technical staff for their constant guidance, continual encouragement, and understanding; more than all, they taught me patience in my endeavor. It is a great opportunity on my part of work with such intellectuals and experts in the field

In jubilant mood, I express ingeniously my whole-hearted thanks to my team members for their constant support, and various other Octro members working as limbs of our organization for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my university and computer science department to provide me with such a big opportunity.

Table of Content

List of Abbreviations

Abbreviation	**Explanation**
PHP	Hypertext Preprocessor
OTP	Open Telecom Platform
DBMS	Database Management System
SQL	Structured Query Language
XMPP	eXtensible Messaging and Presence Protocol
XML	eXtensible Markup Language
API	Application Programming Interface
DOM	Document Object Model
I/O	Input/ Output

List of Figures

Abstract

Gaming has been an integral part of human life. Playing electronic games comes with both advantages and certain disadvantages but if these games are played with certain limits their disadvantages can be overcome. It was proved in a study that playing electronic games helps to improve one's visual memory and the ability to make decisions.

In this project I have built an online tambola game where players can play the game with a bunch of other players which are too looking for some refreshment in their busy lives. Tambola game is one of the ancient games that is believed to be originated in Italy in early 1500's and since then it is being played as an integral part of many occasions like kitty parties, birthday parties etc.

My team at Octro Inc. and I decided to build the game on web using the modern JavaScript libraries and frameworks so that it can be cross-platform independent. There came many challenges in the completion of the game but at last it's the final result that matters, which came out to be very efficient and responsive.

The following report contains the detailed explanations about the project and how we were able to successfully complete the project, the stack of technologies used and many other things.

Chapter – 1

INTRODUCTION

1.1 Introduction

Tambola, which is also popularly called Tombola, Housie or Indian Bingo is a popular game which is believed to have developed in early 1500s in Italy. The Tambola game is easy to learn and fun to play. All of us must have played this game at some social gatherings, office get-togethers,society events and kitty parties.

Depending on the competency level Tambola can be played in many different ways. ‘Bingo’ is the American version of the game and it is a slightly different from the traditional Tambola. Each player have to buy at least one and at max four tickets in order to be able to enter a traditional Tambola game.

Tambola is played with a total call of 90 numbers from 1-90 being called out only once at a time by the dealer which the players strikes out on their respective tickets. A valid Tambola ticket always have twenty seven blank spaces, arranged in a random fashion in nine columns and three rows, where each row comprises of five numbers and others are kept just blank spaces, each column amy contain a maximum of three numbers, with some variation which depends on various Bingo or Tambola companies and where the game is played.

In a ticket the first column contains numbers from 1 to 9, and the second column contains numbers from 10 to 20, the third, 20 to 30 and so on until the last column, that contains numbers from 81 to 90.

The Tambola game begins with drawing out a ball marked with numbers. As the game progresses, the board is placed with the respective ball that is drawn. The objective of the game being marking all the numbers present in the ticket as called by the dealer. The player who is able to mark all the numbers first in a winning pattern and calls a claim is declared as the WINNER of that particular pattern, after it the dealer checks his ticket and verify if the numbers in the ticket are concurrent with numbers drawn.

If a claimed winning pattern comes out to be wrong, it is called BOGUS and one cannot continue that game with that same ticket. The game ends when all the 90 numbers are drawn by the dealer, or when a winner is declared for all the patterns of the game, whichever comes first. To win exciting prizes in a Tambola game, one need to perfectly match the following winning combinations.

Some most popular winning combinations tried in a Tambola game are as follows:
Early Five: Fastest first five numbers marked in a ticket.
Top Line: All the numbers in the topmost line are marked as they are called.
Middle Line: All the numbers in the middle line are marked as they are called.
Bottom Line: All the numbers in the bottom most line are marked as they are called.
Four Corners: The ticket with all four corners marked first i.e. 1st and last numbers of top and bottom rows.
Full House: The ticket with all the 15 numbers marked first.

1.2 Problem Statement

The Tambola game developed by the company is already active on the Play Store and have a large number of downloads as well as a good rating. On the client side, scripting is done on PHP. Each table are all isolated from each other is done using the architecture practices of OTP of Erlang. We also have use of very highly distributed in memory DBMS like Mnesia and Redis for storing some temporary useful data as cache.

We are also using MySQL for handling some permanent data like user profile and money. For communication between the server and the clients, we are taking the help of Ejabberd XMPP servers.

Now, the company is targeting to make the application compatible to web, i.e. shifting the code to JavaScript which was originally built using C++. Where back-end functionality remained the same.

For client-server communication we used XMPP technologies like Core for streaming XML etc. We also used Strophe.js which is made as an XMPP management library for JavaScript. Its purpose is to enable the real-time, web-based XMPP applications and messaging protocols that runs in a browser.

Strophe is a combination of libraries to speak the XMPP protocol. While most of the other XMPP implementations and libraries are only focused on some chatting-based applications but, Strophe takes a wider consideration. It is widely used for implementing notification systems, real-time games, search engines, as well as other traditional messaging apps.

The various implementations of Strophe library are ready for production, easy to use, well documented and easy to extend. Strophe.js surely makes creating any real-time web based applications very easy.

1.3 Objectives

i. To correctly modify the code originally written in C++ to JavaScript (Cocos2d-JS).

ii. To send request to the server whenever the client makes any request from login to playing on a public/private table.

iii. To successfully listen to the responses sent by the server in the form of XML.

iv. To successfully register namespaces of different requests with the XMPP, so as to automatically call the listeners of the respective response being sent by the server and carry out respectful operations with the data received in the query.

Fig. 1.1 Splash Screen for the game

1.4 Methodology

In this section, we will discuss the way I carried out the project.

I. Basic Idea:

i. First step was to learn JavaScript, as I have never done it before.

ii. Learning the XMPP protocol and Strophe library.

iii. Getting familiar with the Cocos-JS functionalities and different implementations for the front-end programming.

II. Tools Used:

i. Cocos2d-JS

ii. Google Chrome’s V8 engine – For Debugging Purposes

iii. JavaScript’s Strophe library

III. Language Used:

JavaScript

Chapter – 2

LITERATURE SURVEY

2.1 Introduction

Tambola which is additionally called Housie or Indian Bingo. Tambola by Octro is a LIVE ONLINE gathering game that you can play with REAL PLAYERS from around the globe or one can also host your own Private Get-together. You can play whenever and anyplace. Guest/Dealer calls the haphazardly produced number/signal each in turn. Give appropriate consideration to the numbers that are being called. In the event that a number called by Caller shows up on your ticket, tap on the number to stamp it. Cautiously monitor the considered numbers that are on your ticket. Correspondingly, all different players separate the numbers on their tickets as the numbers are called by the Caller. The champ being the main individual to separate every one of their quantities of a triumphant blend. A player can have more than one winning mixes on a solitary ticket. On the off chance that a specific winning blend has been effectively asserted, it can't be guaranteed once more.

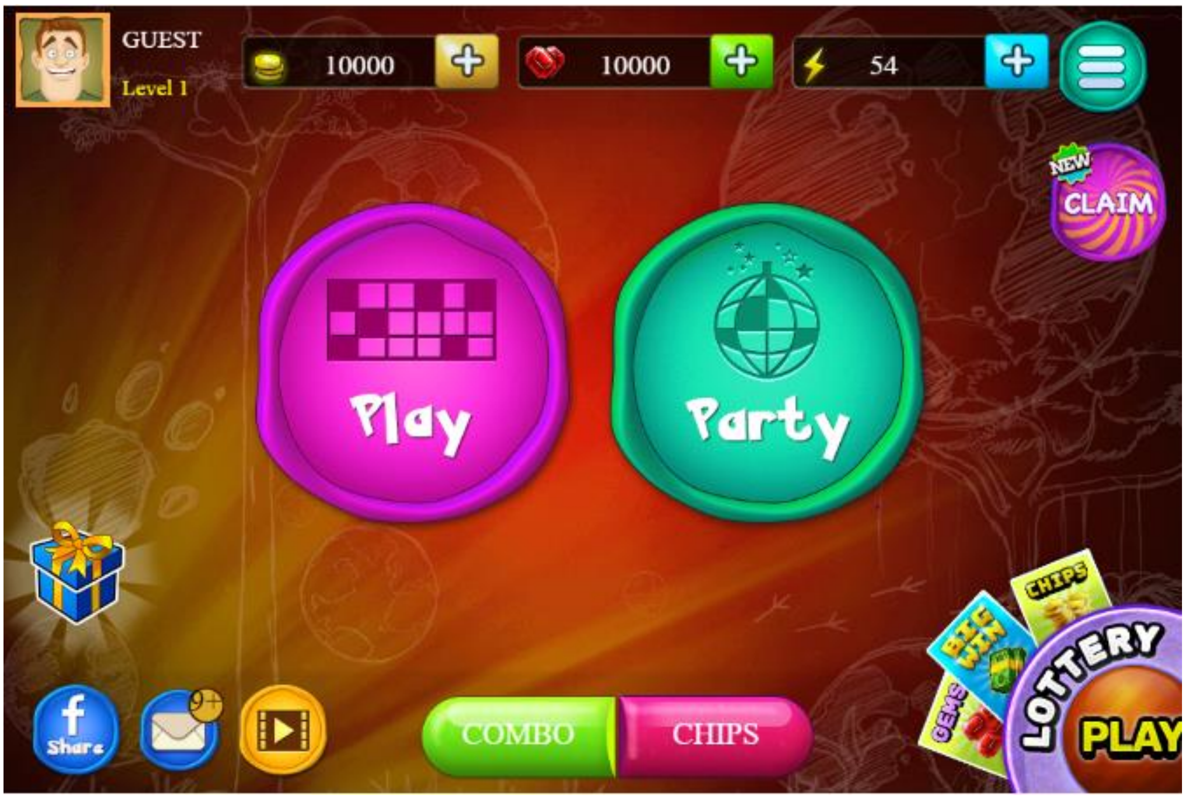


Fig. 2.1. Main Screen after logging in as Guest

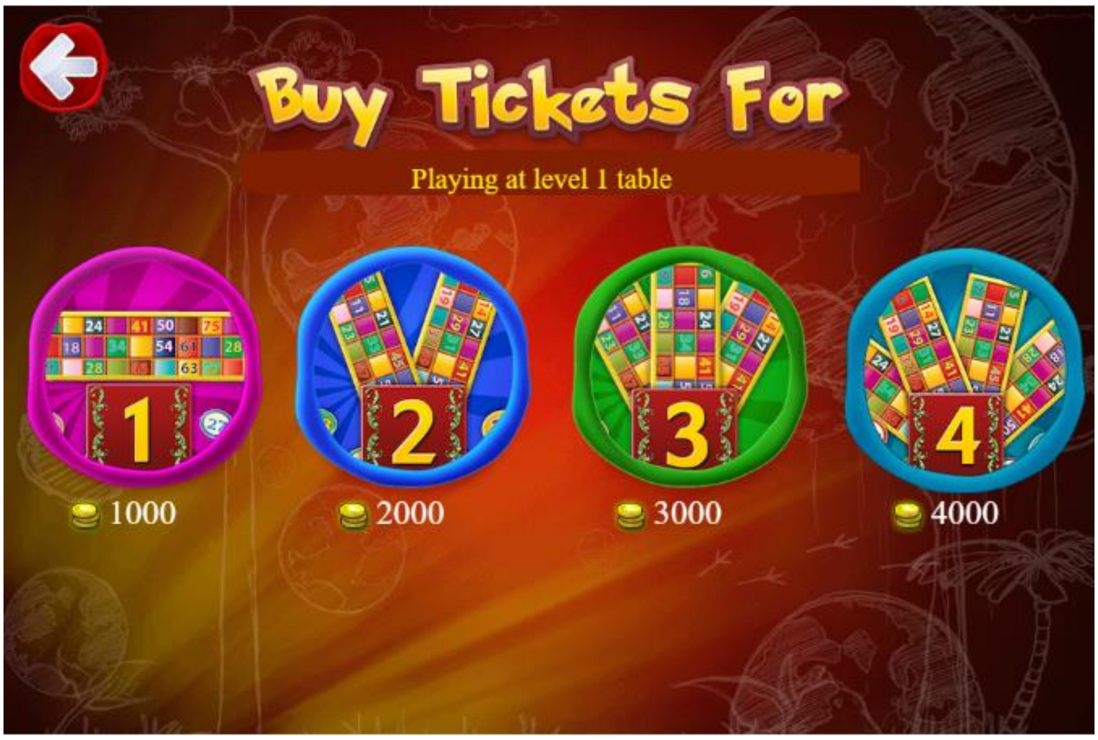


Fig. 2.2. Select ticket screen

The game Tambola here is actually played with virtual money initially assigned to the user. A newly joined player is a Level 1 player who gets initial 10,000 chips. Each ticket for a level 1 player costs around 1000 chips. The total amount of tickets being sold decides the winning amount for the game.

2.2 Table Types

Public Table:

This is the default game mode that Tambola offers. In this game mode, you are matched with random people who are also playing on Public Tables and compete to win. To join a public table, simply click on the "Play" button in the main, select how many tickets you want to play with and start playing.

Party Table:

A player is also able to create his own private Tambola party and then inviting friends by sharing unique code and password for the party to join the party. Just launch the Tambola app and select Party. Now one would see two options: Join and Create.

Creating a party:

Click on the Create button, a list of different options is then displayed, from them decide the price for a ticket and other relevant options and click on Proceed. One’s own Tambola party table is then created and now he can invite his Facebook friends. A Party ID and password are also displayed on the table. Passing this info to friends whom one allows to join the table.

Joining an existing party:

Clicking on the Join button, one will see three options: Enter Code, Friend’s Party and Nearby.

Friend’s Party:

In order to join a party created by one’s friend on Facebook, select the Friend’s Party button. Select party that one would want to join. Also, if a friend on Facebook invites one to join a party, one will get a message notification on the main Tambola screen. Click that particular message and one will be able to join the table.

Entering through Code:

If one has table ID and password for a friend’s party, click on Enter Code. Enter the relevant information. One will be able to join the party.

When someone creates a Tambola Party, the chips collection from selling the tickets will be distributed among the winners. Other than winning chips, one can also announce their own prizes for the winners. Different types of winners in a game are: early five, top line, four corners, bottom line, middle line and full house.

2.3 Winning Combinations

There are six combinations in a traditional Tambola Game:

Early Five: Fastest first five numbers marked in a ticket.

Top Line: All the numbers in the topmost line are marked as they are called.

Middle Line: All the numbers in the middle line are marked as they are called.

Bottom Line: All the numbers in the bottom most line is marked as they are called.

Four Corners: The ticket with all four corners marked first i.e. 1st and last numbers of top and bottom rows.

Full House: The ticket with all the 15 numbers marked first.

2.4 Claiming a Winning Combination:

When, one have selected all the numbers of a winning blend on his ticket, press the Claim button before the following number is called. On the new screen that shows up, select the winning mix or combinations that you are asserting. In the event that the triumphant blend

guaranteed by you is right, you will get the winning sum. In the event that the case is inaccurate, your ticket will be proclaimed Bogey.
In Tambola it is essential to guarantee the winning blend preceding the following number is called by the vendor. On the off chance that a player falters or defers the case and the following number is called, the case is viewed as invalid or intruder. When there is an intruder, the ticket is dropped.

2.5 Multiple Claims

The game needs to allow everybody to win something in the game. Remembering that, we offer numerous cases in our game. At the end of the day, in games with an enormous number of players, there can be different cases made on a triumph condition. For instance, on the off chance that somebody asserted a top line, another person can likewise make a top line guarantee.

Chapter – 3

SYSTEM DEVELOPMENT

3.1 Introduction to Cocos2d-JS

Cocos2d is an open source programming system. It contains numerous branches that are Cocos2d-x, Cocos2d-objc, Cocos2d-XNA and Cocos2d-JS.
Cocos2d-JS is an adaptation, based on JavaScript, of Cocos2d-x engine which comprises of Cocos2d-x's and Cocos2d-html5's various JavaScript Bindings. Cocos furnishes any game with multi-platform and cross-browser capacities, joined by Cocos2d-x, it includes amicable and improved API's of JavaScript. Cocos2d-JS rehashed work processes in every stage in v3.0, it gives a steady advancement to any platform one need to convey to, regardless of web-based or local. "Running all over and over but Coding Once" appears to be unimaginably simple and common after using the software. With a single codebase, one is able to run a game on any internet browsers or local stages including OS-MAC, iOS, Android and Windows. It will get one's game extraordinary open doors practically all channels of appropriation.
Then again, on the off chance that anyone is just inspired by easygoing apps on the web-platform, one can implant straightforwardly in his/her site page the popular Cocos2d-JS-Lite version that is very simple to utilize and as light as a plume. Besides, JavaScript's amicable APIs make any game's advancement experience a breeze - simple to code, convey and test.
Meanwhile, Cocos2d-JS v3.0 is very ground-breaking alongside these are also some of the new cool highlights: Assets Manager, Editors Support, Object Pool, and so forth.

3.1.1 What is a Director in Cocos2d-JS

Cocos2d-x utilizes the idea of a Director, much the same as in a film! The Director object controls the progression of activities and guides the fundamental beneficiary. Consider yourself the Executive Producer and you instruct the Director! One normal Director task is to control Scene substitutions and advances. The Director is a common singleton (adequately, there's just each occurrence of the class in turn) object that you can call from anyplace in your code.
To interact with the Director, you need to call on it, e.g.

```
var director = cc.director.runScene(Scene)
```

3.1.2 Event Dispatch Mechanism

Event Dispatching popularly, is a technique to respond to user events.

The basics:

- Event listeners comprises of the functionality you want to run after an event is dispatched.
- Event dispatcher alerts the listeners of various events to perform their functionalities.
- Event objects are the main events that contain the main event information.

There are five different types of inbuilt Event Listeners:

- EventListenerTouch: Touch-events are handled by it.
- EventListenerKeyboard: Keyboard-events are handled.
- EventListenerAcceleration: Accelerometer-events are taken care.
- EventListenerMouse: Mouse-events are handled.
- EventListenerCustom: Customly developed events are handled.

Below is an example of how to create a custom event listener and register the listener to a particular event.

```
var listener = cc.EventListener.create({        //Defining the listener
        event: cc.EventListener.Custom,
        eventName: “displayHelloWorld”,
        callback: (event)=>{
                console.log(“Hello World”);
        }
});
cc.eventManager.addListener(listener,1);//To register the listener with
eventManager
cc.eventManager.dispatchEvent(new cc.EventCustom(“displayHelloWorld”));
//Manually dispatching a custom event
```

3.2 Introduction to JavaScript

JS or **JavaScript**, is a popular programming language which complies to the ECMAscript’s specification. It is a multi-paradigm, high level, and often just in time compiled. JavaScript has the normal curly-bracketed syntactical structure, prototypal object-orientation, dynamic typing and other first class functions.

Like many other, it is also one among the main technologies of the World Wide Web. It enables responsive and interactive pages and it is considered to be important piece of web-based applications. There are a number of websites which uses JS for client side pages , and nearly every majorly used web browsers comes with a dedicated specialized engine in order to execute JS.

It is also multi-paradigm i.e. JS supports crucial programming, event-driven and various functional styles. JavaScript comes embedded with APIs for dealing with contents, dates, standard data structures, DOM, and regular expressions. However, it itself doesn't include any methods for input/output (I/O), networking, graph facilities or storage as the environment hosting the language provides these APIs.

Firstly, engines of JavaScript were being embedded in browsers only, but now they come built-in in servers, mostly through Node.js usually. These are used in a varied quantity of applications which are developed with frameworks like Cordova and Electron.

Despite having some sort of similarities between Java and JavaScript, which includes syntax, name and various resp. standardly used libraries, both of the languages being largely different and differs in design with each other.

3.2.1 The Rise of JavaScript

Netscape handed over JavaScript to ECMA International in November 1996, as the beginning of a standardly stated specifications with which all the browser vendors have to comply. It prompted the official arrival of the 1st ECMAscript language particular in 1997'June.

The measures procedure proceeded for couple of years, after the arrival of ECMAscript 2 in 1998'June and ECMAscript 3 in 1999'December. Work on ECMAscript 4 started in the 2000's.

In the interim, Microsoft increased an undeniably prevailing situation in the program showcase. By the mid-2000s, Internet Explorer's piece of the overall industry arrived at 95%. This implied JScript turned into the true standard for customer-side scripting on the Web.

Microsoft at first took an interest in the guidelines procedure and executed a few propositions in its JScript language, however in the long run it quit teaming up on ECMA work. Accordingly, ECMAScript 4 was retired.

3.2.2 Standardisation and Growth

During era of I-Explorer strength in the mid-2000’s, scripting on the customer-side was stale. It began to update in 2004, the replacement of Netscape, company Mozilla, discharged the famous Firefox program. It was generally welcomed by everyone, which took a huge piece of the overall industry from the I-Explorer.

In 2005, ECMA International was joined by Mozilla, and they began working for XMLstandards. This prompted Mozilla to work mutually together with the Macromedia, they were actualizing E4X’s current build in the ActionScript 3, that depended on an ECMAscript’s 4th draft. Objective became the normalization of the Actionscript as the latest ECMAScript build. By then, Adobe discharged Tamarin execution as a venture and was open-sourced. Be that as it may, Tamarin and ActionScript 3 were excessively not quite the same as setting the customer-side scripting, also without collaboration from Microsoft, ECMAscript 4 never arrived at fulfilment.

Google appeared its Chrome program in 2008, with the V8 JavaScript motor that was the first to utilize without a moment to spare accumulation, fundamentally improving execution times. Other program merchants expected to update their motors to contend.

In July 2008, these divergent gatherings met up for a meeting in Oslo. This prompted the inevitable understanding in mid-2009 to consolidate all important work and drive the language forward. The outcome was the ECMAScript 5 norm, discharged in December 2009.

3.2.3 Website Client-Side Usage

JavaScript is one of the most dominating client-side scripting language for the Web, almost 95% of the websites actively using the same for this. Various scripts are merged in or taken from HTML docs and interaction takes place with the DOM. All of the major web browsers nowadays, have a built-in engine that can execute JavaScript code on the device of a user itself.

Here are some examples stating what is scripted behavior:

- Updating the content of a page without reloading the whole page. For example, websites related to social media use Ajax so that the users are able to post new messages without leaving the page.
- Animating page elements, which includes fading them in and out, moving, and resizing them.
- Games and videos.
- Validating values of a form in order to make sure that those values are acceptable before submitting them to the server.
- Transmitting info regarding user's behavior for analytics, personalization and ad tracking.

3.2.4 Features of JavaScript

The following are the features that are common to all the ECMAScript implementations of it, unless mentioned explicitly.

- **Imperative and Structured:**

  JavaScript bolsters a significant part of the organized programming sentence structure from C (e.g., if explanations, while loops, switch conditions, do while loops, and so on). One fractional special case is perusing: JavaScript initially had just capacity checking with var. ECMAScript 2015 included catchphrases let and const for square checking, which means JavaScript now has both capacity and square perusing. Like C, JavaScript makes a differentiation among articulations and explanations. One syntactic distinction from C is programmed semicolon addition, which permits the semicolons that would typically end explanations to be overlooked.

- **Weakly Typed:**

  JavaScript is considered to be a weakly typed language, meaning that certain types are implicitly casted which depends on the operation being used.

  - The '+' operator implicitly casts both operands to strings while both the operands being numbers. It's because the '+' operator doubles a concatenation operator.
  - The '-' operator casts both operands as a number

- Both of these unary operators (+, -) casts the operand to number

Values are casted as strings similar to the following:

- Strings are considered as it is
- Numbers are always converted to their string representation
- Array elements are casted to strings after which they are mrged by commas (,)
- Other objects are casted to the string [object Object] where Object is the name of the constructor of the object

Values in JavaScript casts to number values by casting them to strings and in turn then casts those strings to numbers. This process can be updated by defining the inbuilt toString() and the valueOf() functions on the prototype number and string casting respectively.

JavaScript largely for some time have received extreme criticism for its style of implementing the conversions as the complex rules can be easily mistaken for inconsistency e.g., if we add a number to string, that number will first be casted to a string before any concatenation is performed, but in case of subtraction of a number from a string, the string is first casted to a number before subtraction is performed.

- **Dynamic**

  JS is a dynamic-typed language. Like the other scripting languages, type is generally associated with a value inspite of associating with any expression. e.g., variable which was initially number is reallocated as a string. It has a varied ways for testing object's type, and also includes duck-typing.

  JavaScript also includes a function called 'eval' that could run statement literals as string objects during running the code.

- **Object Oriented**

  Inheritance(prototypal) in JS explained as follows: One makes objects of prototype, and then makes new instance. In JavaScript objects are considered as mutable, so that one is able to augment any newly formed instance, releasing those fresh methods and fields. These objects in turn acts as prototype for the freshly created ones. one doesn't require classes for making a lot same objects, they self-inherits from objects.

In JavaScript, an article is an affiliated exhibit, enlarged with a model (see beneath); each string key gives the name to an item property, and there are two linguistic approaches to determine such a name: spot documentation (obj.x = 10) and section documentation (obj['x'] = 10) A property might be included, bounce back, or erased at run-time. Most properties of an item (and any property that has a place with an article's model legacy chain) can be identified utilizing a for..in loop.

JavaScript has a number of built-in objects, including Function and Date.

- Prototypes:

  The JavaScript language takes in using prototypes whereas most other object-oriented programming languages uses classes for the purpose of inheritance. It’s possible to use a lot class-based features using prototype feature in JavaScript.

- Functions in form of constructors of object:

  Capacities twofold as item constructors, alongside their common job. Prefixing a capacity call with new will make an occurrence of a model, acquiring properties and strategies from the constructor (counting properties from the Object model). ECMAScript 5 offers the Object.create strategy, permitting express formation of an example without consequently acquiring from the Object model (more seasoned conditions can relegate the model to invalid). The constructor's model property decides the item utilized for the new article's inward model. New techniques can be included by altering the model of the capacity utilized as a constructor. JavaScript's worked in constructors, for example, Array or Object, likewise have models that can be adjusted. While it is conceivable to change the Object model, it is commonly viewed as terrible practice on the grounds that most items in JavaScript will acquire techniques and properties from the Object model, and they may not anticipate that the model should be altered.

    - Functions being referred as to Methods

      Unlike most of the object-oriented programming languages, in JavaScript, there isn’t any distinct difference between the definition of a function and that of a method. Rather, the serious distinction occurs when function is called. Whenever a function calling occurs as a method of certain object, function's local ‘*this’* keyword is bounded to that object for that invoke period.

- **Functional:**

  A basic function always is top-notch; it is viewed as an object. All things considered, it can have strategies and properties, e.g., .bind() and .call(). Any nested function is defined as a characterized function inside any other function which is made every time the external one is conjured. What's more, each of the function which is nested shapes a lexical conclusion: The lexical extent of an external function (counting any consistent, nearby factor, or contention esteem) turns out to be a piece of the inward condition of every internal function’s object, significantly thereafter executing the external function is concluded. JavaScript likewise underpins mysterious capacities.

- **Delegated:**

  JavaScript also includes explicit and implicit delegations.

    - **Roles (Mixins and Traits)**

      JavaScript supports numerous function-oriented implementations of patterns like Mixins and Traits. These functions defines various additional behaviours having atleast a method bounded to the ‘this’ keyword in the body of the function. A ‘Role’ needed to delegate externally through calling and applying to various objects which needed to be featured additional behaviour which aren’t transmitted through prototyping chain.

    - **Inheritance and Object composition**

      While express functions based appointment covers organization in JavaScript, certain designation as of now happens each time the model chain is strolled so as to, e.g., discover a technique that may be identified with yet isn't legitimately possessed by an article. When the strategy is thought that it was gets called inside this present item's unique situation. Subsequently legacy in the JavaScript is secured by an assignment automatism which is bounded to the model property of constructor’s capacities.

- **Promises in JavaScript:**

  JavaScript additionally bolsters promises which is its method of taking care of non-concurrent activities. There's an implicit object of Promise that offers a great deal of functionalities to take care of promises, and also characterizes how it ought to be dealt with. It permits one to connect a handler to an offbeat activity's inevitable achievement worth or disappointment reason. This lets non-concurrent strategies return esteems like coordinated techniques: rather than promptly restoring the last worth, the offbeat strategy restores a guarantee to flexibly the incentive sooner or later. As of late, combinator strategies were presented in the JavaScript particular which permits designers to consolidate various JavaScript promises and do procedure based on various situations. The strategies presented are: Promise.all(), Promise.race(), Promise.any() and Promise.allSettled().

3.3 Getting to know XMPP

The eXtensible Messaging and Presence Protocol (XMPP) is, at its most fundamental level, a convention for moving little, organized bits of information between two spots. From this unassuming premise, it has been utilized to construct huge scope texting frameworks, Internet gaming stages, web indexes, coordinated effort spaces, and voice and video conferencing frameworks. Increasingly one of a kind applications show up each day, further exhibiting how flexible and incredible XMPP can be. XMPP is made of a couple of little structure squares, and on these natives numerous bigger developments have been made.

Inside XMPP are frameworks for building distribute buy in administrations, multi-client talk, structure recovery and preparing, administration disclosure, continuous information move, security control, and remote method calls. Regularly, XMPP software engineers make their own, one of a kind development that are fitted precisely for the current issue.

Most web based life develops that have pushed sites like Facebook, MySpace, and Twitter into the bleeding edge are additionally prepared into XMPP. Inside XMPP, you'll discover programs brimming with contacts that make a social diagram with coordinated or undirected edges. Nearness warnings are sent consequently when contacts come on the web and go disconnected, and private and open messages are the bread and butter use of XMPP frameworks. Designers will now and again pick XMPP as the hidden innovation layer essentially in light of the fact that it gives them numerous social highlights for nothing, leaving them to focus on the novel bits of their application.

3.3.1 What is XMPP?

XMPP, similar to all conventions, characterizes an organization for moving information between at least two imparting elements. For XMPP's situation, the elements are typically a customer and a server, in spite of the fact that it likewise takes into account distributed correspondence between two servers or two customers. Numerous XMPP servers exist on the Internet, open to all, and structure a united system of interconnected frameworks.

Information traded over XMPP is in XML, giving the correspondence a rich, extensible structure. Numerous cutting edge conventions do without the data transfer capacity investment funds of a twofold encoding for the more down to earth highlight of being intelligible and thusly handily repaired. XMPP's decision to piggyback on XML implies that it can exploit the enormous measure of information and supporting programming for managing XML.

One significant component XMPP gets by utilizing XML will be XML's extensibility. It is incredibly simple to add new highlights to the convention that are both in reverse and forward perfect. This extensibility is put to incredible use in the in excess of 200 convention expansions enlisted with the XMPP Standards Foundation and has furnished designers with a rich and basically boundless arrangement of apparatuses.

XML is referred to essentially as an archive position, however in XMPP, XML information is composed as a couple of streams, one stream for every bearing of correspondence. Each XML stream comprises of an initial component, trailed by XMPP refrains and other top-level components, and afterward an end component.

Each XMPP verse is a first-level kid component of the stream with all its descendent components and properties. Toward the finish of a XMPP association, the two streams structure a couple of substantial XML records.

XMPP refrains make up the center piece of the convention, and XMPP applications are worried about sending and reacting to different sorts of verses. Refrains may contain data about other elements' accessibility on the system, individual messages like email, or organized correspondence planned for PC preparing. A model stanza is appeared here:

```
<message to=’eli@lo.com’
from=’dary@pem.com/dance’
type=’chat’>
<body>How Are You? </body>
</message>
```

In a commonplace customer server XMPP meeting, a refrain, for example, this one from Eli to Mr. Dary will make a trip from Eli's customer to her server. Her server will see that it is routed to an element on a remote server and will build up a XMPP association with the remote server and forward the message there. This correspondence between servers takes after the email organize, yet not at all like email servers, XMPP servers consistently discuss legitimately with one another and not through transitional servers.
This immediate correspondence wipes out some normal vectors for spam and unapproved messages. This is only one of the numerous manners by which XMPP is intended for security. It additionally underpins scrambled correspondences between endpoints through utilization of Transport Layer Security (TLS) and solid confirmation instruments by means of Simple Authentication and Security Layers (SASL).
XMPP is intended for the trading of little bits of data, not enormous masses of twofold information. XMPP can, in any case, be utilized to arrange and set up out-of-band or in-band transports, which can move huge squares from point to point. For these sorts of moves, XMPP works as a flagging layer.

3.4 Strophe.js

A JavaScript’s powerful library used for implementing XMPP via Web Sockets and Bidirectional-stream Over Synchronous HTTP(BOSH).

Its purpose is primarily to enable various real-time XMPP applications that may run on any platform/browser.

Chapter –4

Results

4.1 Main Screen After Logging-In

Once any user logs into the game as a guest or log in with Facebook, he is welcomed with a welcome message on the screen and the main screen as shown in fig 4.1 is displayed on the screen. The screen contains a number of buttons to perform different operations. The player can buy chips and gems by pressing the respective buttons on the screen.

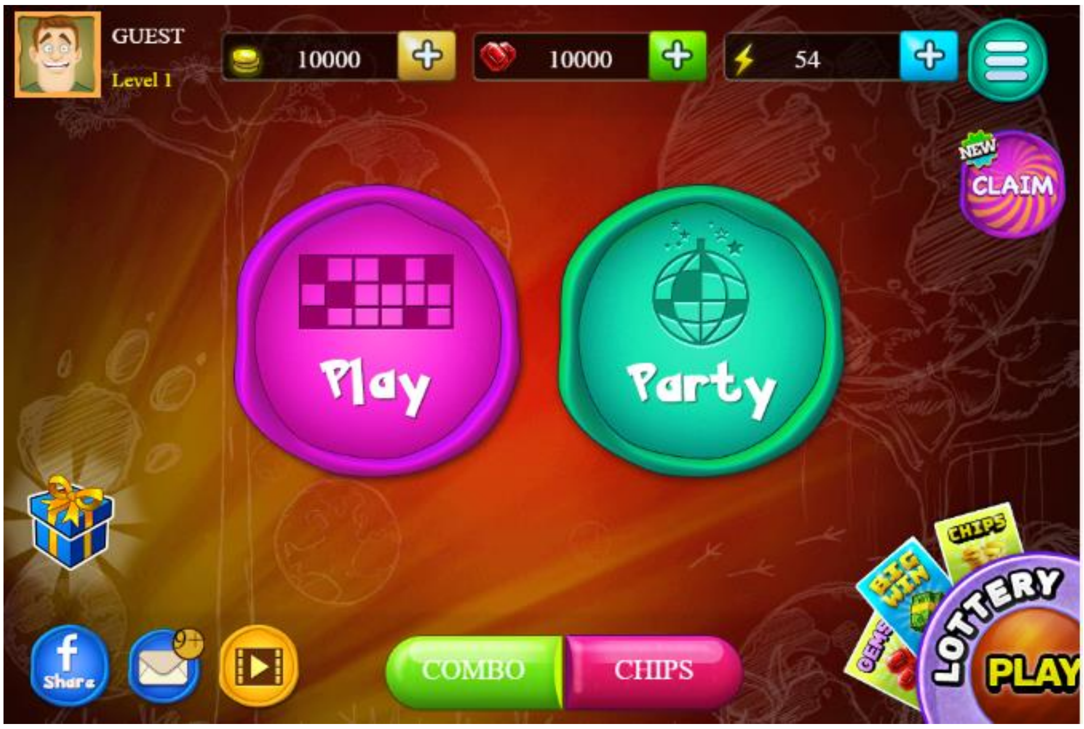


Fig. 4.1. Main Screen

A player can also control a bunch of options and functionalities in the settings menu. There are two types of gameplays in the game i.e. a player can join a public table in which there will be random people playing on the same table matched randomly and other being a private table or also called a party in which a player can either create a party or join someone else’s party just by entering the unique game id and the password randomly assigned by the server. Once any player joins a private party and all the players are ready to play, the admin can start the party and the number will start to be called.

4.2 Select the number of tickets

Any player who wishes to be counted-in in a public or private party has to buy ticket for the game. The game allows the players to buy 1 to 4 tickets. Playing with more tickets surely increases the chances to win maximum prizes in the game. At level 1 table the cost of each ticket is 1000 which gradually increases as the level increases. Once the player successfully buys the ticket he is taken directly to the table in case of a public game and to the party lobby in the case of a private game where only the admin will be able to start the game.

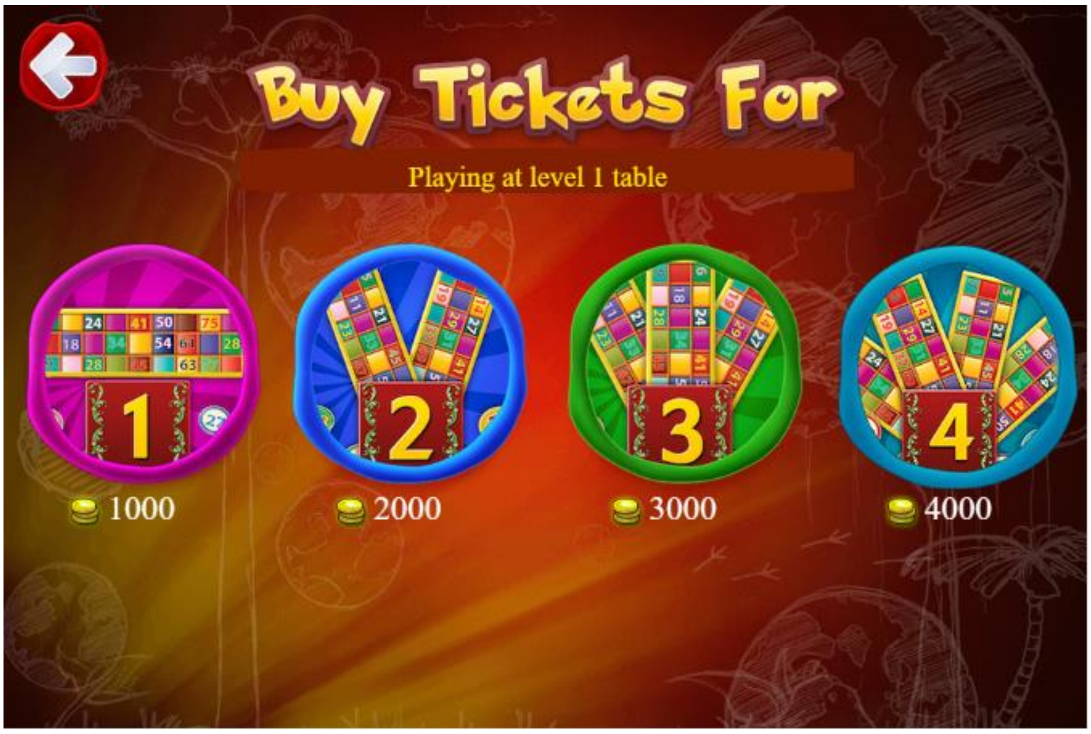


Fig. 4.2. Buy Ticket Screen

4.3 Waiting for the players to join

Once a player buys a ticket he is directed to the waiting screen where more players are being added to the game. The players to be added are decided by the server, players in the waiting queue are added first and then other requests are entertained. Once all the players are assigned a particular table the server sends a XMPP stanza containing all the information regarding the game table, this stanza gets processed by a particular listener where the iq of the stanza was registered and respective operations are performed.

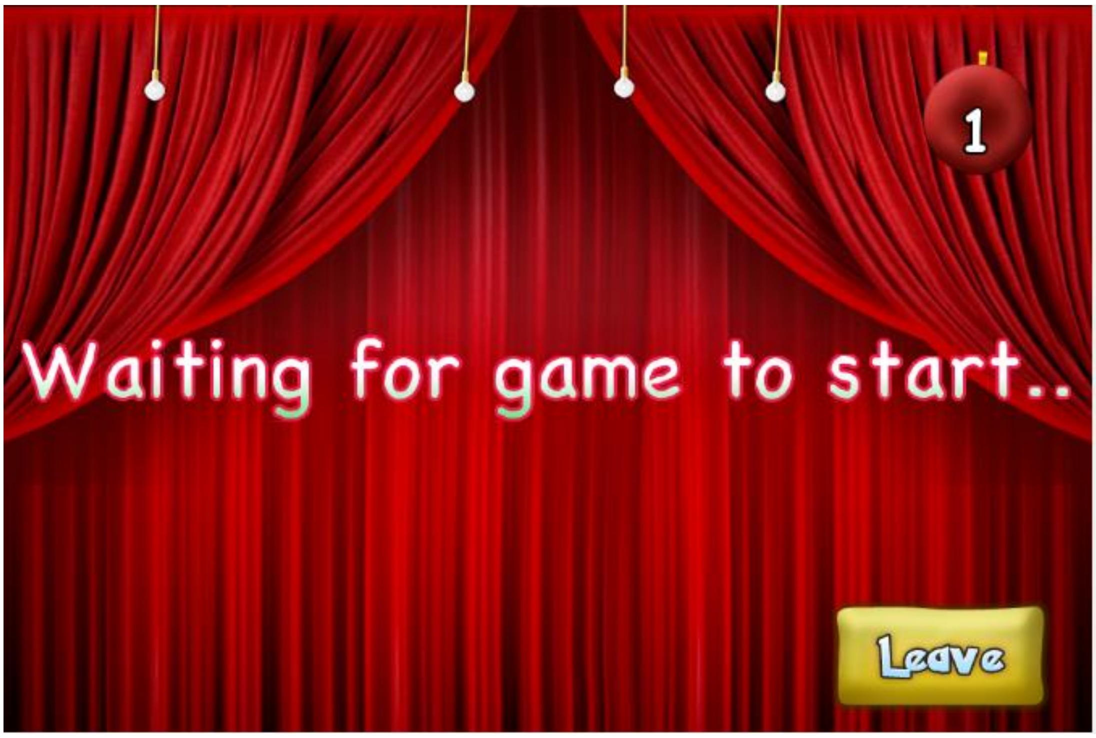


Fig. 4.3. Waiting Screen

4.4 Playing on public table with 1 ticket

Once the table information is processed by the listener, a table containing ticket/s is displayed on the screen based on the number of tickets bought by the player.

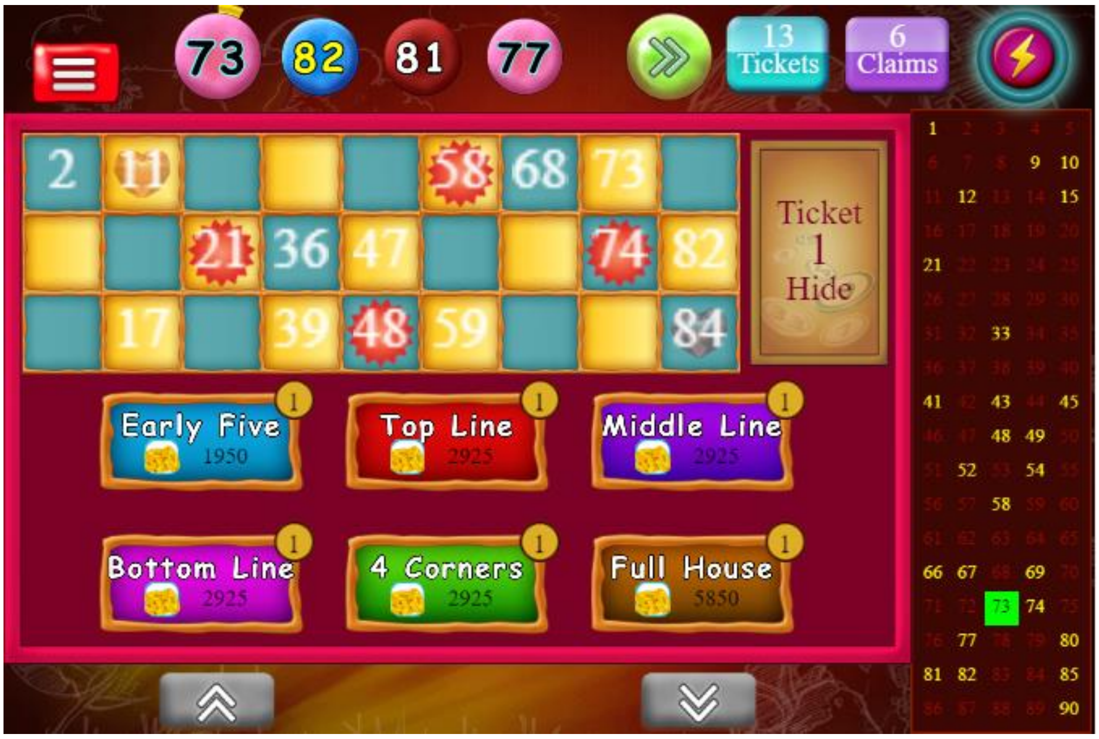


Fig. 4.4. Game Play Table

Each ticket consists of 3 rows and 8 columns with numbers in random 15 cells from 1 to 90. The dealer calls these numbers one by one in a random order and the player is supposed to mark the numbers announced by the dealer.

4.5 Claiming a winning pattern

Once the player observes a correct and verified winning pattern, he can claim that pattern by pressing that particular button. A claim request is then sent to the server with information about the state of the table and the numbers involved in the winning pattern. The server checks and verifies all the numbers with the announces numbers and send a success response if all the numbers are called already or a failure response if any number was not called. In this scenario the ticket is termed as a Bogus ticket and that ticket is no longer valid in the game.

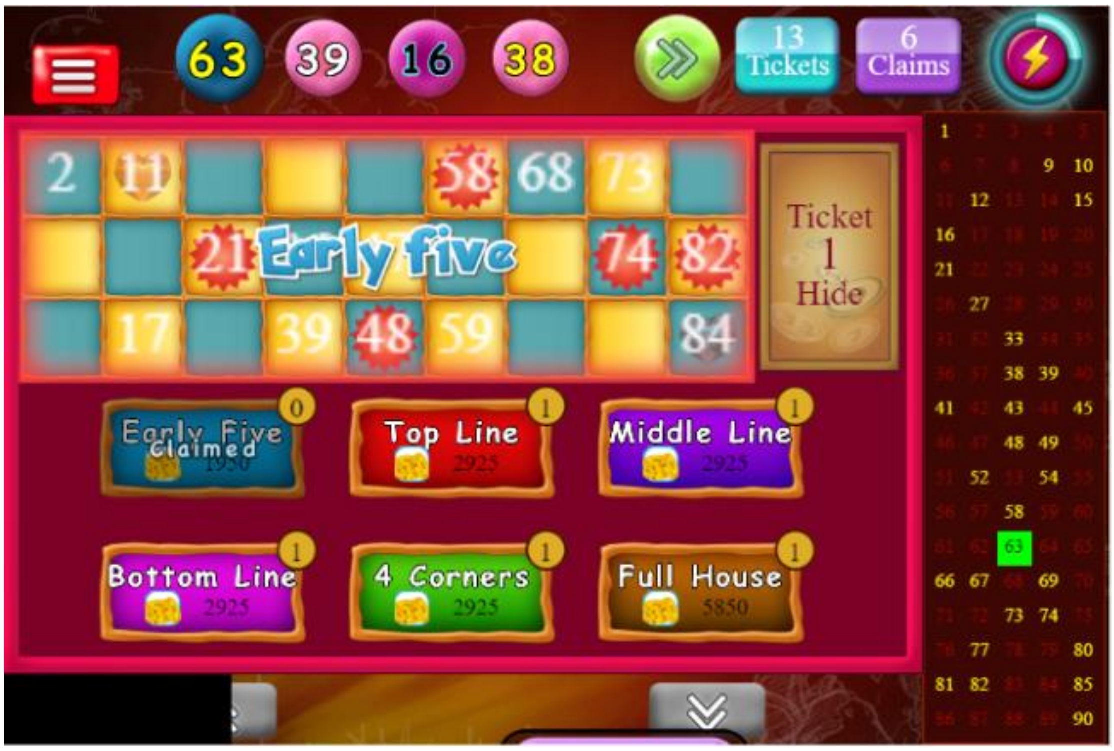
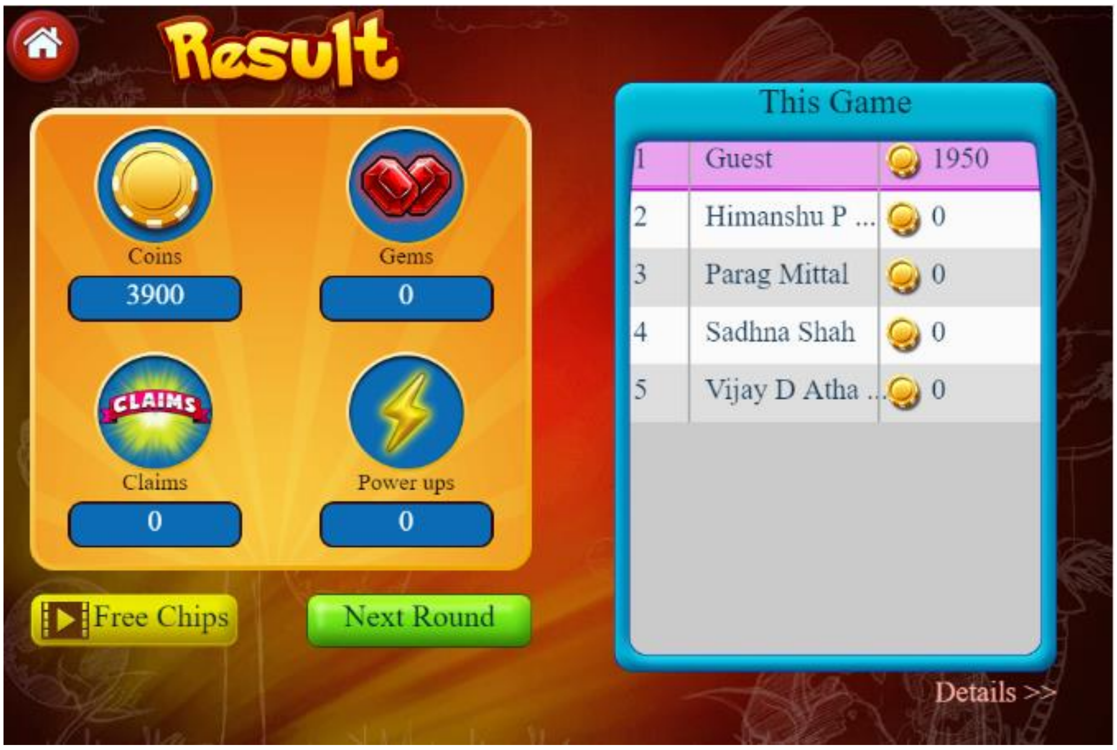


Fig. 4.5. Claiming winning patterns

4.6 Result Screen

Once the game is completed a result page is shown with the winnings of the player and all other players in the game. Player can redirect directly from here to another game by the NEXT ROUND button.



Coins	3900	Gems	0
Claims	0	Power ups	0

1	Guest	1950
2	Himanshu P ...	0
3	Parag Mittal	0
4	Sadhna Shah	0
5	Vijay D Atha ...	0

Fig. 4.6. Result Screen

4.7 Creating a party

Tambola parties are often played at many gatherings like kitty parties, birthday parties etc. where family and friends can play tambola together and enjoy. Our game also has a feature of tambola parties where any one member can create a party

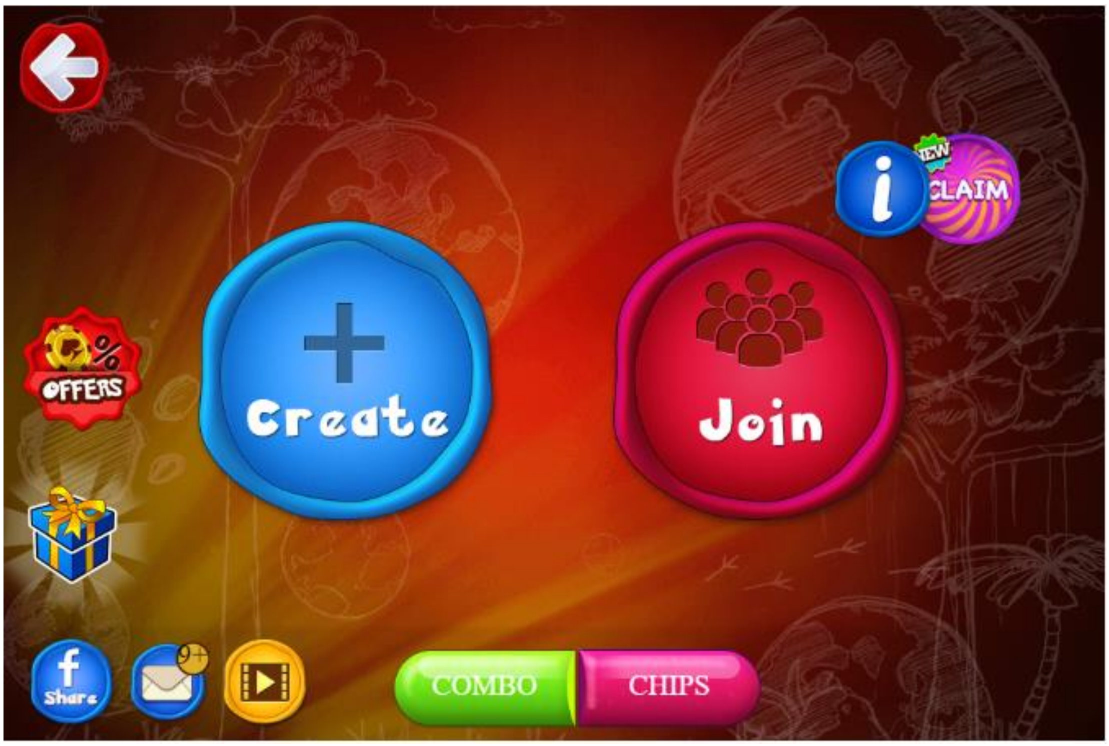
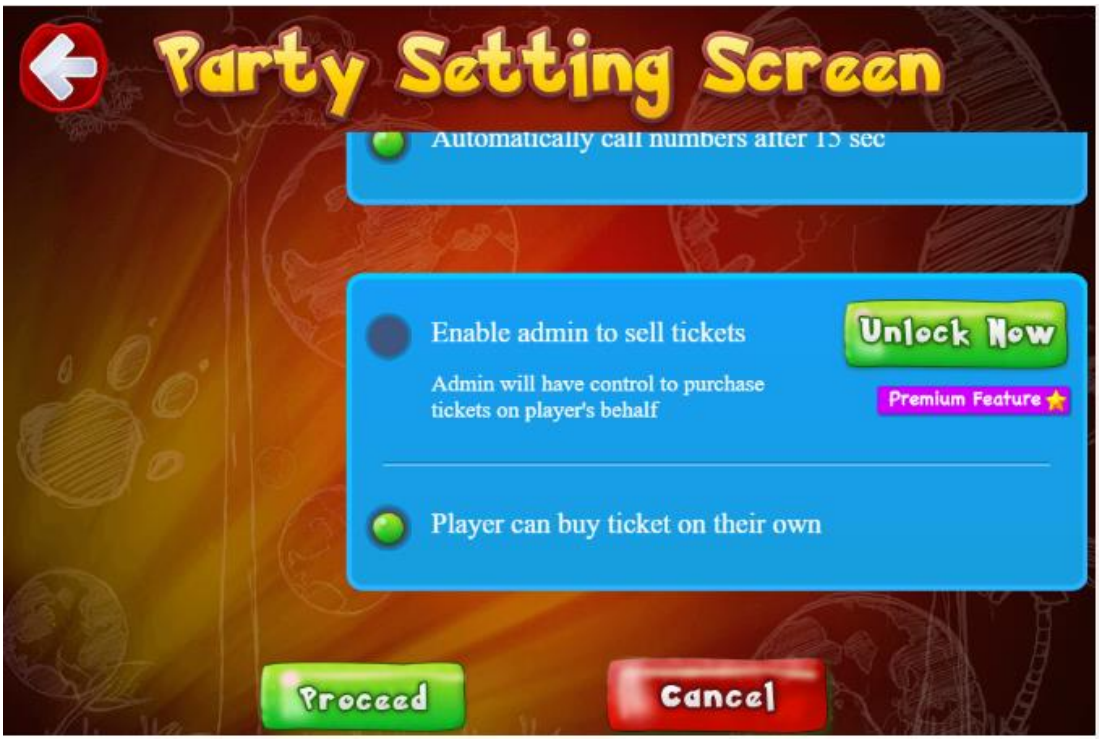


Fig. 4.7. Create Party

simply by adding the party name and can select a bunch of different options from the menu and can proceed by pressing he proceed button.



● Automatically call numbers after 15 sec

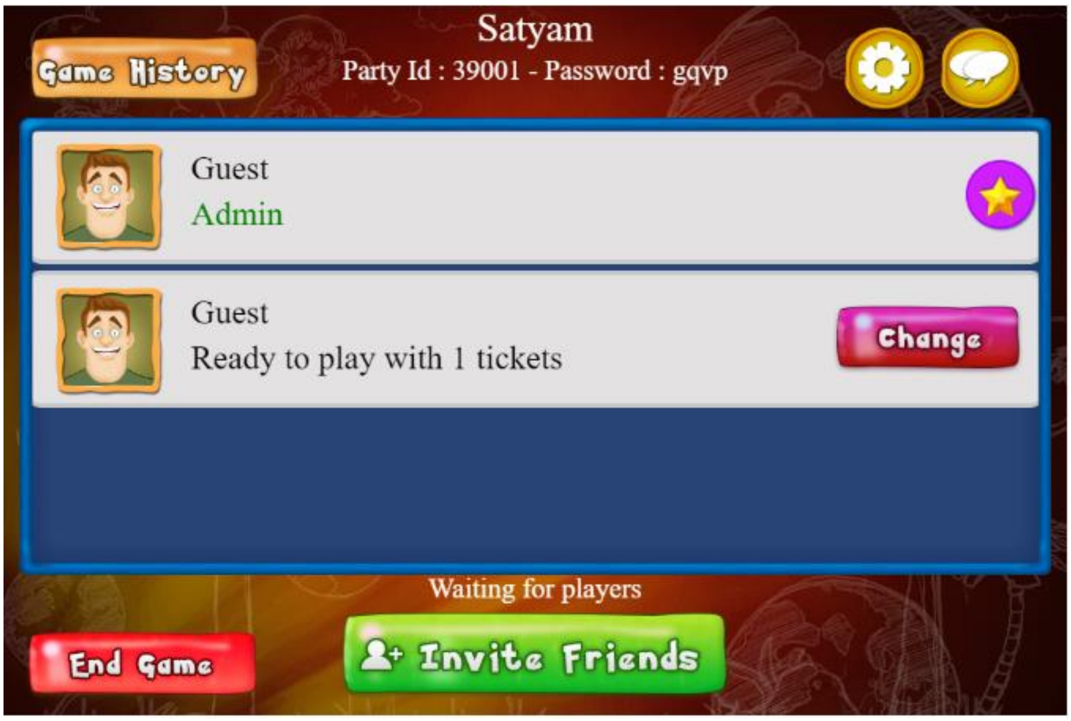
○ Enable admin to sell tickets
Admin will have control to purchase tickets on player's behalf

● Player can buy ticket on their own

Fig. 4.8. Party Settings Screen

4.8 Waiting for the players to join the party

The party will be created and the admin will be directed to the party lobby screen. From where the admin can either invite the players using Facebook, WhatsApp or a text message, or he can simply share the unique party id and password with the players.



Satyam

Party Id : 39001 - Password : gqvp

Guest

Admin

Guest

Ready to play with 1 tickets

Fig. 4.9. Party Lobby Screen

Once there are 2 or more players added in the game and all of them have successfully bought a ticket the start button be automatically added and the admin can then start the party.

Chapter –5

Conclusion

5.1 Conclusion

My team at Octro, Inc and me were successfully able to complete the whole functionality of the Tambola game as per the schedule and despite all the challenges that came in the way. Also the game is doing good as of now and hope it will do well in the future too.

5.2 Future Scope

As of now the Tambola game is up to date with all the functionalities that were initially taken into consideration. But we will surely look for upgrades in the game and add more good functionalities too.

References:

[1] Tambola.octro.com. 2020. *Octro Tambola - How To Play*. [online] Available at: <http://tambola.octro.com/HowToPlay/index.php> [Accessed 23 May 2020].

[2] Cocos2d.org. 2020. *Cocos2d Is A Family Of Open-Source Software Frameworks For Building Cross-Platform Games&Apps.*. [online] Available at: <http://cocos2d.org/> [Accessed 23 May 2020].

[3] GitHub. 2020. *Cocos2d/Cocos2d-Js*. [online] Available at: <https://github.com/cocos2d/cocos2d-js> [Accessed 23 May 2020].

[4] Docs.cocos2d-x.org. 2020. *Director · Gitbook*. [online] Available at: <https://docs.cocos2d-x.org/cocos2d-x/v4/en/basic_concepts/director.html> [Accessed 23 May 2020].

[5] MDN Web Docs. 2020. *Javascript*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Accessed 23 May 2020].

[6] En.wikipedia.org. 2020. *Javascript*. [online] Available at: <https://en.wikipedia.org/wiki/JavaScript> [Accessed 23 May 2020].

major_satym

ORIGINALITY REPORT

13%	12%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	en.wikipedia.org Internet Source	8%
2	www.jungleetambola.com Internet Source	2%
3	Submitted to St. Patrick's College Student Paper	1%
4	www.infosmush.com Internet Source	1%
5	crbtech.in Internet Source	1%
6	Submitted to Jaypee University of Information Technology Student Paper	<1%
7	www.internetbingoblog.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off