

Generic Rules-based Engine for Indian Card Games

Piyush Dargopatil¹, Ketaki Kadlaskar², Pallavi Chapole³, Pooja Khaire⁴

Computer Department, Sinhgad College of Engineering, Pune, India^{1,2,3,4}

Abstract: Card games have become popular on the computer and have been extended to a multiplayer network format in recent years. There is a paucity of single platforms which would enable the game creators to define the rules for particular game, specifically for Indian Card Games. The Rule Engine developed for filling this lacuna aims to facilitate the creators to create any custom game belonging to the family of standard and well-known trick-taking games like Seven-Eight, Five-Three-Two, Judgment, etc. Creators are provided with a desktop app which allows them to choose rules from a generic set of trick-taking game rules resulting into custom game database. This rule-set is then used to build the application package for that particular game. The game created would be in the form of a hybrid application. AI Engine for a couple of popular trick-taking games is also provided.

Keywords: multiplayer, trick-taking, hybrid application, rule-set, AI Engine.

I. INTRODUCTION

Card games can be classified into various categories liketrick-taking card games, shedding card games, partnership card games, etc., of which trick-taking games are the most popular ones. The underlying principle of all the trick-taking card games is the same- players need to maximize the number of tricks made by playing better cards than those of their opponent [1]. There have been numerous applications and multiplayer card games, developed in recent years, more for the card games belonging to the Western culture than the traditional Indian card games. In India card games are played with many variations.

The Rules-based approach to develop multiple card games, customized games as well as well-known games belonging to the family of trick-taking card games out of a single general rule-set is the scintillating feature which gives genericness to the game engine. The generic rules-based engine eases the job of creating multiple games and reduces the redundancy of writing the same rules, which are applicable to trick-taking category of card games.

The games are created in two stages. The creators are provided with a desktop application which allows them to choose their own desired rules from the rule-set, the backbone of the game engine, thus enabling the creators to create customized games belonging to the tick-taking category of card games. These games are then deployed as individual hybrid applications which allow the end users to play the multiplayer card games over the internet.

These hybrid applications are single page applications which provide highly responsive and dynamic user interfaces to the end users. An artificially intelligent player is also provided for playing a couple of well-known card games. A desktop application which aims to allow human players to play against the artificially intelligent player is created using an AI engine. The above modules provide different functionalities but work together to engender a

game engine which acts as the seedbed for many trick-taking card games.

II. SYSTEMMODULES

The game engine is comprised of three main modules- namely the Rule-Set module, AI module and Gameplay module. All these modules work in tandem to facilitate the creators and end users to create and play games respectively.

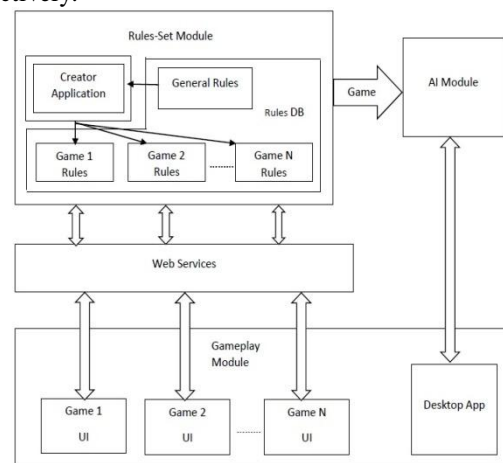


Fig. 1. Architectural diagram- the system modules

The Rule-Set is the database of general rules belonging to the family of trick-taking card games as well as rule sets of specific games created. An important sub-module, which provides the creators with a desktop application which allows the game creators to use the generic set of rules to create specific games, has been identified and realized in the Rule-Set module. The AI module is a desktop application which enables the end user to play couple of trick-taking card games against the system. Gameplay module provides the end users with a user interface. The Gameplay module also has a local storage. The Gameplay module validates the game playing with the help of the local storage.

III. RULE-SET MODULE

The Rule-Set module is the heart of game engine [2]. This module consists of a database and a sub-module which is a desktop application meant for the creators of the trick-taking card games.

A. Rules Database

The general rules belonging to the family of trick-taking card games are stored in the database. The creators chooses specific rules from this rule-set to create specific trick-taking games. The rules of the created games also reside in the same database. General trick-taking card game rules, right from the number of players that can participate in a game to the scoring scheme of a game have been identified and included in the rule-set. All the rules in the database are stored as documents. The rules do not follow a specific schema and hence storing rules as documents is favorable. MongoDB, which is a cross platform document-oriented database, is used to store the rules in form of documents in the collections. MongoDB has a flexible data model, which makes it easy to store data of any structure. MongoDB is highly scalable as it can be scaled up or scaled out horizontally, from a single server to thousands of nodes. The documents are made up of name-value pairs. The rules belonging to the same category are stored in a single collection. For example, the rule which specifies the number of players that can participate in a game, contains both the minimum and maximum limits on the number of players.

B. Creator-Application module

The creator application module allows the creator of the game to interact with the rules database. The creator chooses the specific rules from the general set of rules and individual games' collections are created. The creation of custom games is the highlight of the game engine and thus gives a sense of genericness to the Rule-Set module. This module eases the maintainability of the engine and it can be scaled whenever necessary. The redundancy of defining the same rules over and over again is avoided. Custom games belonging to trick-taking category can be easily developed and identified uniquely.

The creator application is a desktop application built using AngularJS and PHP as client side and server side scripts respectively. AngularJS aids the web developers to streamline their code efficiently and hence, it was befitting.

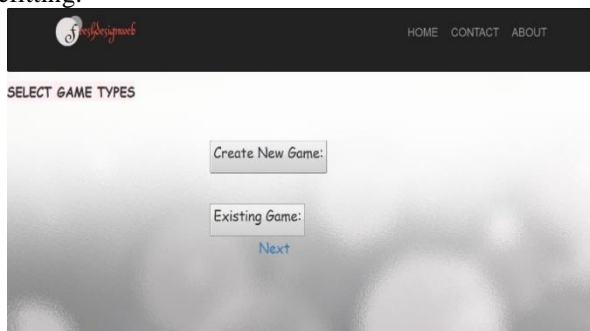


Fig. 2. Creator Application interface- Homepage

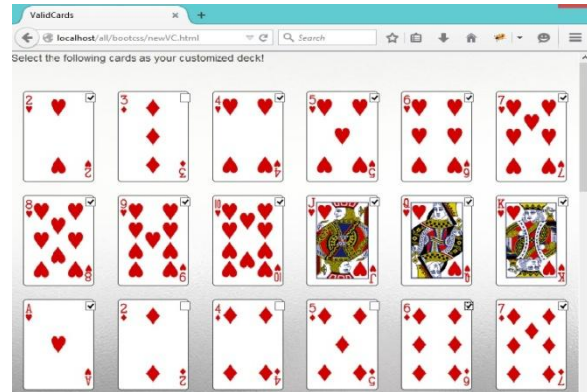


Fig. 3. Creator Application interface- Custom deck selection

AngularJS extends the HTML vocabulary and provides dynamic views in web applications. The resulting environment is expressive, readable and quick to develop. PHP provides database services to AngularJS. This module lets the creators define a game using their desired rules.

IV. AIMODULE

The Artificially Intelligent (AI) player is provided through a desktop application. The AI module is developed using C++ programming language [1]. The AI module performs two tasks. Firstly, it provides the AI player with the ability to choose the trump suit whenever AI player initiates the game and has to decide the trump suit. The choice of trump suit is an important aspect of trick-taking card games. Secondly, and most decisively, it provides the AI player with the ability to choose appropriate card in order to maximize its chances of winning the game.

The aim of players playing any trick-taking card game is to make as many tricks as they can by playing better cards than their opponents. The AI player's goal is no different. The AI player also needs to decide the best trump suit for the game, the trump suit that will increase its winning chances, whenever its AI player's turn to decide the trump [2].

A. Trump Decision:

The AI player decides the trump suit for the game based on cards in its hand. The trump suit is the suit which is ranked above the rest of the suits. This predominance of the trump suit makes it easy to for the AI player to decide the trump suit. The AI player counts the number of cards of each suit in its hand. The suit with the maximum number of cards is then picked as the trump suit. In case two or more suits have the same maximum number of cards, the AI player picks the suit with the highest face value card.

B. Playing the best possible card

The AI player needs to be very smart in choosing the best possible card to play in a round of the game. When AI player has to play the first card in the round, it will try to play a card of a non-trump suit that can win the round, all

the while preserving the trump cards for the later rounds of the game. When an opponent has played a card before the AI, the choice of card for AI for that round is restricted. The AI has to play the card of the same suit that the opponent has played. The AI player plays the card of that suit with the face value just greater than that of the opponent's. If AI player has no such card, looks for a trump card. When AI player has no chance of winning the round, it plays the least important card in its hand. Though the AI player tries to win each round of the game, it looks at the bigger picture of winning the game. Winning the game is more important for the AI than winning a round [3].

V. GAMEPLAYMODULE

The Gameplay module provides the end user with a user interface to play the created games. The gameplay module is deployed on the client side. This module facilitates the game playing by providing the validations of the gameplay. The Game play module contains a local storage which stores the rules, from the Rules Database, needed to carry out the client side validations. The Gameplay module is built using Angular JS, an open-source web application framework to address many of the challenges encountered in developing single-page applications.

A. User Interface

The user interface is built with Angular JS framework. The official Angular JS introduction describes Angular JS as a client-side technology, written entirely in JavaScript. It works with the long-established technologies of the web (HTML, CSS, and JavaScript) to make the development of web apps easier and faster than ever before [10]. It is a framework that is primarily used to build single-page web applications. AngularJS makes it easy to build interactive, modern web applications by increasing the level of abstraction between the developer and common web app development tasks. The AngularJS team describes it as a "structural framework for dynamic web apps." AngularJS, on the other hand, augments HTML to give it native Model-View-Controller (MVC) capabilities. This choice, as it turns out, makes building impressive and expressive client-side applications quick and enjoyable.

The rendering of playing cards is done using Scalable Vector Graphics (SVG) objects. Along with AngularJS, styling is provided using Bootstrap, which has ability to easily create responsive web designs. Responsive web design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

B. Local Storage and Validations

The main functionality of the Gameplay module, apart from providing user interface, is validating the moves of the ongoing game. The local storage contains a subset of the Rules Database. These rules are used to carry out aforementioned local validations. Local storage is needed to reduce the client-server interactions and provide data persistency. For example, in a multiplayer game, players have their own set of cards which are stored locally.

VI. ALGORITHMFOR SHUFFLING CARDS

The shuffling of cards is an important activity in any card game [4]. Distribution of cards to different players is to be done randomly [5]. A good shuffling algorithm needs to be used to reduce the chances of having a biased game. Pseudo-Random numbers are usually generated by linear congruential methods.

The shuffling algorithm uses the well-known by linear congruential method, originally given by D. H. Lehmer. The method is fast and easy to program [7].

A sequence of integers is started with a value y_0 and continued by

$$y_{i+1} \equiv ay_i + r \pmod{m}, \quad 0 \leq y_i < m \quad \text{for all } i. \quad (1)$$

The above linear congruential method generates nonnegative integers y_i which are smaller than the modulus m . If $y_n \equiv y_0 \pmod{m}$, then the whole sequence $\{y_i\}$ is repeated. The smallest integer n such that $y_n \equiv y_0 \pmod{m}$ is called the length of the period. For a good approximation to the continuous uniform distribution, the period should have maximum length for a given modulus m . The maximum periods of the linear congruential generators depend on relatively simple properties of a , r , and m .

When $r \neq 0 \pmod{m}$ the method is called Mixed Congruential Method. The result is stated as

Theorem: A complete period of the sequence $\{y_i\}$ contains all residues mod m , if and only if

- (i) r and m are relatively prime;
- (ii) $a \equiv 1 \pmod{p}$ for all prime factors p of m ;
- (iii) $a \equiv 1 \pmod{4}$ if 4 is a factor of m .

A proof of this theorem can be found in Hull/Dobell, Jansson, and Knuth.

VII. DEPLOYMENT

The Generic Rules-based Engine creates a game database. The game database is then utilized for developing a hybrid application. Hybrid applications combine the advantages of both types of the development of applications (web and native) and the best choice is to create cross-platform applications [6].

The application is developed using Phone Gap, mobile application development framework, based upon the open source Apache Cordova project. It allows you to write an app once with HTML, CSS and JavaScript, and then deploy it to a wide range of mobile devices without losing the features of a native application [8]. It extends the features of HTML and JavaScript to work with the device. Phonegap currently supports development for the operating systems Apple iOS, BlackBerry, Android, and Microsoft Windows.

Different application packages for different games can be developed using corresponding subsets of the Rules Database specified by the creator. The genericness of the engine is proved when the creator is facilitated to create

more than one games, and develop separate application packages for the same.

VIII. FUTUREWORK

A. Extending to other families of card games

The rule-based approach to create multiple card games can be extended to all families of card game such as shedding cards, partnership games, comparison games, accumulation games.

B. Social Integration

To make game play experience more human, social components to the applications developed could be added. Addition of video chat to ongoing games, so that players can see each other; however, this would require a significant upgrade to the web hosting services used currently.

C. AI Player

Implementation of an advanced AI for all games created can be achieved. This feature can be advantageous, where in, requirement of minimum number of players for a game can be fulfilled by AI player.

IX. CONCLUSION

The generic game engine based on rules makes it possible to create many trick-taking card games using the same general rule-set. Creators are able to create the trick-taking card games they wish to create thus enabling different popular and standard games to be played with many variations. Many never-played-before custom games which can be very interesting can also be created. The games created are delivered in the form of hybrid cross-platform applications. The user interface provided to end users is ultra-responsive with a high degree of dynamism. Overall, the game engine excels in creating a wide range of popular as well as custom trick taking card games and furnish highly attractive trick-taking card game applications.

ACKNOWLEDGMENT

We would like to thank our project guides, **Mrs. D. R. Pawar** and **Ms. Pradnya Baviskar**, for guiding us throughout the course of the project and for providing us with helpful feedback on our progress. This work is sponsored by Soft-Corner Pune.

REFERENCES

- [1]. "Developing an artificial intelligence for whist", Jason Fong, University of California, 2011
- [2]. "A Lightweight Rule-Based AI Engine for mobile games", Lynne Hall, Russell James and Lynne Newall, University of California.
- [3]. Markus Enzenberger, Martin Muller, Segal R, "Fuego-An Open Source Framework for Board Games and Go Engine based on Monte Carlo Tree Search" ,Computational Intelligence and AI in Games, IEEE Transaction, Dec 2010
- [4]. Dealing cards in poker games, Information Technology :Coding and computing, 2005.ITCC 2005,Vol 1, Publisher: IEEE
- [5]. AI for General Strategy game playing, Publisher: Wiley-IEEE Press,2014
- [6]. Development of web based card game engine", University of Gothenburg, Goteborg, Sweden, September 2010.
- [7]. Pseudo-Random Numbers: The Exact Distribution of Pairs, U. Dieter, Mathematics of Computation, Volume 25, Number 116, October, 1971.

- [8]. Croos Platform App- A Comparitive Study, Paulo R. M. de Andrade, Adriano B. Albuquerque, IJCSIT, 2014
- [9]. Steve Francia, "MongoDB and PHP- Document-oriented data for Web Developers", O'Reilly, 2012
- [10]. Ari Lerner, "ng-book- The Complete Book on AngularJS", 2013