Euchre

Chris Schlenker

10/9/17

Hanover College, Computer Science

**Introduction**

Where I am from, euchre is a very popular card game. It is a card game that has been played for hundreds of years and is based on strategy. Personally, I have been playing euchre since I was a junior in high school and I would consider myself somewhat of an expert on the game. Having said this, I still sometimes wonder which card gives me the best chance of winning a given hand. For my project, I plan to build a euchre mobile application where you can play against 3 virtual players. I will be using game theory and reinforcement learning integrated with neural networks to help me code these virtual players.

Euchre is a competitive card game played between two pairs of teams. Only 24 cards are needed from the deck, and they are as follows: Aces, Kings, Queens, Jacks, 10s, and 9s in every suit. The dealer begins by dealing to his left, and continues to deal counterclockwise. Each player receives 5 cards. The dealer can either give the cards in rounds of 3, then 2, or vice versa, but he must keep the pattern uniform. The dealer will end up with 4 undealt cards known as the “kitty” and they will then turn the top card of the kitty over. Then each player will have their turn to call “trump” based on the top card of the kitty, making the called suit the highest-ranking cards for that hand, and attempt to win the hand. The player to the left of the dealer will lead and the players will follow the lead playing clockwise. The player who wins that trick will lead the next trick.

The goal of the game is to reach 10 points. To score 4 points, you must call “alone” (play without your teammate) and win all 5 tricks, but you can still earn one point by winning either 3 or 4 tricks. Calling “alone” means that you are calling up trump and you are playing without your partner. The gameplay will then commence normally for that hand. To score 2 points, you and your teammate must either win all 5 tricks after calling suit, or win 3 out of 5 tricks after the other team has called suit (this is called a euchre). To score just 1 point you must simply call suit and win 3 out of 5 tricks. Another important aspect of the game are the ranks of the cards. For example, let us say that a player has determined that hearts will be the trump. Then the rank of the cards will be the Jack of hearts (the right Bauer), the Jack of Diamonds (left Bauer), Ace, King, Queen, 10, and 9, respectively. All other suits will be ranked normally with Ace being high.

The virtual players will each have individual names and will initially be making decisions based on heuristics that I program into the agents. These heuristics will allow them to make decisions that are within the rules of euchre. From there, I will learn the game theory behind euchre and how to program AI using game theory using *Artificial Intelligence for Games* (Millington & Funge) before eventually writing learning algorithms for the agents. The agents will possess the ability to pass or call trump, lead when it is their turn, and follow based on which cards have been laid.

I intend to learn the game theory of euchre while developing the AI (artificial intelligence) that the user will be playing against. The other component of coding my agents will be using reinforcement learning with neural networks. The idea is that when you are playing against the computers, they will always play the card which maximizes their chances of winning a given hand. Before I can develop AI capable of doing this, I will first have to learn how to wrap my head around programming artificial intelligence. I believe this will be a challenging and fun endeavor.

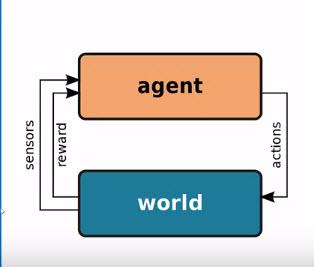
In summation, I am interested in this project for three reasons. First, I get to program an application based on a game I enjoy playing. Second, I will learn how to program AI for the first time in my college career. This is something that I have always been interested in, and I want to make some headway with this. Third, I will learn the game theory behind euchre. This will allow me to apply my knowledge of math to my senior project. I will have to learn how to integrate game theory with AI to build this application. Since these are two topics I am very interested in, it is something that I will work hard to achieve.

**Background/Related Work**

An online developer, Mike Lynch, is in the process of developing his own application on Euchre. The basic overview of my project is similar, but I plan to use game theory to help my agents decide which card to play, while he is using minimax which is a decision rule used in decision theory to help minimize the loss of any worse-case scenario. In this small but important detail, my project is different. There is a useful, online blog where a person played through 100 hands and found the chances that every single card had of winning. For instance, the right bower (the highest card) won 100% of the time, while the Ace of next suit (same color, different suit) only won 44% of the time (euchreuniverse). Knowing these statistics will help me initially start coding my logic for the virtual players before I start building the automated intelligence. This blog has done what I plan to do as part of my application, which is keep a database of all hands played to know the percent chance a certain card has to win. Over time as many games are played, these statistics may change slightly or significantly, but this will give me a good jumping off point.

I will be using game theory to build my AI. Game theory is the branch of mathematics concerned with the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice of action depends critically on the actions of other participants. Also, “Game theory classifies games according to the number of players, the kinds of goal those players have, and the information each player has about the game” (Millington & Funge 668). In Euchre, there are 4 players split into two pairs. The goal of Euchre is to try to help your team win and make your opponents lose. If you score 1 point for winning, then your opponent scores -1 for losing; this is known as a zero-sum game. The information in Euchre is known as “Imperfect information”. In a game like chess the players have “perfect information” because they know all the possible moves the opponent can make and the effects those moves would have. In Euchre, all we know for sure are the 5 cards that we hold in our hand, and possibly one more card depending on whether the top card in the “kitty” is turned over. This aspect will make it more difficult to use game theory.

A major part of my project will be coding the agents, and to do this I plan to use game theory to help them make the smartest decisions, but I want to have my agents to be somewhat adaptive as well. Ron Livingston has written a dissertation which covers reinforcement learning algorithms and how to apply those algorithms to Euchre. This dissertation will guide me in coding the agents and having them adapt as more games are played. One aspect of coding agents is having “rewards”. These are what will be given to the virtual player after they have decided. “A positive reward indicates a resulting state that is ‘good’ and a negative reward indicates one that is ‘bad’, with the magnitude of the reward relative to how good or bad the state is “ (Livingston 4). These rewards will also be relative, in other words some ‘good’ decisions may be given a greater reward than other ‘good’ decisions and will have a greater influence in future decisions made by the agent. The following figure illustrates a feed-back loop that agent will go through to learn.



**Overview of Euchre Mobile Application**

Two of the major components for this project will be creating the graphics for a user friendly and appealing interface, and coding the AI that the user will play against (thanks to advice from a recent Hanover graduate, I am thinking that I will be able to find a tutorial that will help me build the graphics for a basic card game). This will at least help me get started so I can start digging into the aspect of this project that I am most interested in, which is coding my own virtual players. For this part, I plan on hand coding the game theory to create the virtual players. After accomplishing this, I want to work on having some adaptive learning for the virtual players. To do this, I will simply pull from my database of games to see if playing a certain card has given them a higher percentage of winning in past games. If it has, then they will play that card. The idea is that the virtual players will get better as more games are played through reinforcement learning. I will also be developing a database for storing and recording games. I will be able to pull from this database to find the chances of winning for each card. This should help in my initial coding for the agents. For this project, I will be using Android Studios and right now, I am planning to have 4 different activities, MainActivity, GameActivity, RulebookActivity, and StatisticsActivity. I think that each of these activities enriches the project in their own way and will give the user a fuller gameplay experience. Once these 4 activities are completed, with Main and Game Activities being the most important, I will be able to possibly expand the project or work on any necessary tweaks.

## To help me code the application, I have a few different sources that will help bridge any gaps in my knowledge on how to play euchre or how to use android studios. I will be using bicyclecards.com to refer to when I am coding the rules into my application. I will also be using Head First: Android Development to help me build my project. I’ll also be using online tutorials to develop my basic card game. I’m building a basic card game because this will yield the smallest working project and will give me a good base to build on. After I build the basic card game, I will be able to move on to more important parts of the project, such as developing a database, implementing euchre rules, and creating my agents.

**Project Details**

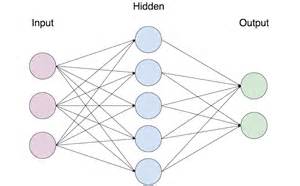
There are four important components that will require the most time to build my application. The first component will be generating graphics and animations for playing the card game, and the second will be building all the logic for the agents using game theory. The third component will consist of writing learning algorithms to help the agents reduce error in their decisions. The fourth component will involve building the database and integrating it with the application, with the end goal of allowing my agents to pull from this database to help in their learning process. I am planning to have 4 different activities for the user to navigate through and they are listed and explained in greater detail below.

**MainActivity:** From here the user will be able to navigate to one of the 3 following activities. This will just be a basic activity with no built-in logic. I will simply use intents for navigation. The most important part of this activity will be building an appealing user interface.

**GameActivity:** This will be where the card game is played. This is also where I plan to have the coding of the agents. All the game theory I research will be implemented here. A major part of creating the game activity will be initially building the basic card game and coding euchre rules into the game later. Once these basic tasks are complete, I will then move onto coding my agents.

The first part of coding the agents will involve learning the game theory behind euchre. I have started doing research to learn more about the area of game theory and more specifically how it applies to creating AI. I plan to do more research into this area and find out exactly how I can write algorithms using game theory into my application.

I will need to do more research to create reinforcement learning algorithms and building those algorithms from neural networks. Artificial neural networks have been used before in computer programming, and they are based on the “concept of a network of neurons, modelled on how physical brains works” (Livingston 9).



The agent will use this network to adjust weights on certain inputs to tell them what the output should have been. Essentially, the main goal is to adjust weights on certain inputs to reduce error, making the agent ‘smarter’. In this way, my agents will be able to learn using neural networks.

**RulebookActivity:** From here, the user will be able to look at a scroll view of a complete list of rules to help them learn how to play Euchre. This is a very basic component and there won’t be much logic here. This activity is included to help any novice euchre players better understand euchre. Again, the main part of this activity will be creating a friendly user interface.

**StatisticsActivity:** The user will be able to look at the records and results of previous games using this activity. This activity will pull from the SQL DB to update the statistics after a game is played.

**SQL DB:** I am including a database to record all games that are played. I plan on the agents being able to use this database to create a certain degree of adaptivity. The main intention of this database is to hold percentages of winning for all cards.

**Timeline**

10 /9/17 – 10/15/17: Finish written proposal of project

10/16/17 – 10/20/17: Give oral presentation of project proposal

10/25/17 – 10/29/17: Start building basic card game app (euchre rules not implemented)

10/30/17 – 11/3/17: Implement euchre rules into gameplay

11/6/17 – 11/10/17: Build SQL Database to record games

11/13/17 – 11/24/17: Begin Coding Game Theory into card game

11/27/17 – 12/1/17: Begin coding AI using learning algorithms (additional research if needed)

12/4/17 – 12/8/17: Finish touching up all User Interface

12/11/17 – 12/15/17: Finish AI implementation using Game theory

**Christmas Break:** Make final additions onto euchre application. Also, improve on any component where there is room. Continue coding AI using learning algorithms.

**Post Break:** Finish coding learning algorithms into application.

Rubric

|  |  |
| --- | --- |
|  | **Description** |
| **A** | The basic card game has been completed 1) with complete euchre rules. 2) The database has been built and has been used to update StatisticsActivity. 3) The user interface is friendly and appealing. 4) Game theory has been implemented to help agents make decisions. A.I. are now able to learn from mistakes, and adapt their playing style to maximize their chances of winning. |
| **B** | The basic card game has been completed 1) with complete euchre rules. 2) The database has been built and has been used to update StatisticsActivity. 3) The User Interface is friendly and appealing. 4) Game theory has been implemented to help agents make decisions, but A.I. is still incomplete. |
| **C** | The basic card game has been completed 1) with some integrated euchre rules. 2) The database has been built but has not been used to update records. 3) User interface is mostly friendly, but still somewhat unappealing. 4) There is no A.I. |
| **D** | Only a basic card game has been built. 1) There is no euchre functionality in the game. 2) Database has not been built. 3) User interface is unfriendly. 4) There is no A.I. |
| **F** | No work has been completed. |

**References**

1. Asbridge, Thomas, and Anthony J.F. Griffiths. *Head First Android Development A Brain-Friendly Guide*. Oreilly & Associates Inc, 2017.

This book is a user-friendly guide on how to develop android applications. Specifically, I will be using the chapters on how to develop SQL databases, and how to use intents for application navigation.

1. Lynch, Mike. “Minimax In Action!” *Mike Lynch*, Mike Lynch, 3 Feb. 2017, mikelynchgames.com/game-development/minimax-in-action/.

Mike Lynch is developing his own euchre application and is using minimax, which is a rule used in decision theory to minimize the possible loss for a worst-case scenario. This application may be used as a reference to see how a euchre application is set up and represents a possible way to code my agents.

1. “Euchre – Card Game Rules.” *Bicycle Playing Cards*, [www.bicyclecards.com/how-to-play/euchre/](http://www.bicyclecards.com/how-to-play/euchre/).

After developing my basic card game, I will then implement my euchre functionality. I will refer to this website to assure my agents follow all the rules of euchre.

1. <http://eprints.utas.edu.au/246/1/jrl_thesis.pdf>

This is a dissertation on cut-throat euchre, which has slightly different rules from Euchre, and Sergeant-Major, which are similar in their rules and strategies to play the game. Ron Livingston uses reinforcement learning to code his agents along with applying neural networks to the learning process. Livingston provides several citations to learning algorithms that he uses in his dissertation. I intend to use these algorithms to help my agents learn how to play Euchre

1. <http://euchreuniverse.blogspot.com/2006/05/euchre-probabilities-what-are-chances.html>

This cite contains the percentages that every single card has to win a game. I will use these statistics to develop simple heuristics to initially help my agents make decisions before any game theory and reinforcement learning are applied.

1. Millington, Ian and Funge, John. *Artificial Intelligence for Games*. CRC Press, 2014.

I intend to use this book to help me develop my AI. This book contains learning algorithms and useful information on game theory.