CleverName: Alexa based Advisor for Monopoly[*]

Rahul Ashok Bhagat
Texas A&M University
College Station, Texas
rahul.bhagat@tamu.edu

Nitish Chandra
Texas A&M University
College Station, Texas
nitishchandra@tamu.edu

Purna Doddapaneni
Texas A&M University
College Station, Texas
doddapanenipc@tamu.edu

Dr.Frank M. Shipman
Texas A&M University
College Station, Texas
shipman@tamu.edu

ABSTRACT

Board game are a way of engaging multiple users to interact socially. However, with the increase in number of users, it becomes challenging to keep a balance between maintaining the state of the game and the social engagement. In this paper, we propose the idea of making board games more interactive and immersible with the help of intelligent personal assistant. In this paper, we focus on developing an Amazon Alexa based Assistant system to augment user involvement in"Monopoly". We also introduce an Android Application into the system to better facilitate player's ability to inform Alexa about the current state of the game and to mitigate some of the difficulties that arise when utilizing voice recognition software. The end system would be tested on 10 groups of Monopoly players with a survey at the end to establish enjoyability of having an Alexa-based personal assistant.

KEYWORDS

ACM proceedings, Intelligent Personal Assistance, Board Game Advisor, player engagement

1 INTRODUCTION

Monopoly is a popular board game enjoyed across the world for its complexity and its replay value. The game centers around buying, selling and developing properties utilizing play money in an attempt to "monopolize" all the different properties on the board. While its rules seem straightforward, Monopoly becomes increasingly complex when played by a large group of people. Keeping track of each player's properties, current bank balance and calculation of rent paid when a player lands on another playerâĂŹs property are just a few actions that need to be repeated throughout a game. While this adds to the allure of the game for many, it also makes the game more tedious.

Since 2016, there has been rise in the usage of digital assistants or conversation interfaces or chat-bots due to its great performance in conversing with machines in a dialogue nuance, using natural language. The Big Four: Apple's Siri, Microsoft's Cortana, Amazon's Alexa and Google's new Assistant are visible at the forefront of this technology [3]. With the development of intelligent assistants like Amazon Echo Alexa, a readily available voice recognition program exists to be used for program development. Particularly, we propose to develop an Alexa-based financial advisor to aid player‘s in Monopoly. This program would enhance the game of monopoly by automating some of the more tedious and redundant activities like math calculations. The financial advisor aspect of the program levels the playing field by advising players on basic investing strategies employed by players with significantly more experience. We believe that by automating these two aspects of the game we can further enhance the player‘s engagement in the game. Players are free to focus more on socialization and coming up with more advance strategies as they see fit. We believe automating some of the core aspects of the game will not detract anything from the game as players will still maintain the freedom to choose or ignore Alexa‘s proposed strategies.

2 RELATED WORK

Intelligent personal assistants like Alexa has been developed to aid a busy intelligent user in managing time commitments and completing tasks[6]. A typical knowledge worker juggles a broad range of tasks and responsibilities. While doing so, it must maintain awareness of deadlines and resources, as well as track current activities and new information that could affect its objectives and productivity. Much of its work will require coordination and collaboration with a broad range of people, both within and outside of its immediate organization.

Digital gaming is becoming increasingly present in modern life. It is constantly evolving, with modern developers looking at ways to increase the audience of those playing their games. A relatively recent development has been the introduction of online games and online gaming communities which allow gamers to play against, or in collaboration with, other gamers over the Internet[1]. While there have been releases of several board games in a digital format, not much work has been done to create a physical-digital hybrid system for these board games.

Educational games have also started to become more popular amongst instructors to teach students abstract concepts like Quantum Physics, Calculus and programming. In older times, games like chess were utilized to teach generals how to fight battles. In recent times, Chiarello used board games like Lab on a Chip to teach concepts about the immune system and nano-biotechnologies, Quantum Race to introduce concepts in Quantum Mechanics, and Time Race to elucidate the concepts of Time Dilation in physics[2]. Peter Drake's work utilizes coding of well known board games to emphasize various Computer Science and Programming principles[7].

[*] Produces the permission block, and copyright information

Games like Monopoly can easily be used to teach various concepts of finance like asset management and investing strategies.

3 ISSUES

As we involve technology to resolve a problem in general, we do need to balance out the advantages of using it against the drawbacks that it can lead to. Especially with board games, there are several factors involved that constitute the essence of play . We have tried to minimize the influence of technology on these factors to make the game environment more engaging and entertaining.

3.1 Media Issues

The largest media issue present in our project is utilizing voice recognition. While Alexa can be programmed to pick up a specific voice command, having each player manually announce the command and the value of their dice roll each turn makes having the program more tedious. Alexa also needs to make quick decisions so that players are not left waiting for incoming advice. A workaround for this need to be found to ensure that the voice media does not detract from the cognitive issues present in the game. Specifically, advice given by Alexa could affect the “fun“ factor of the game. A social issue that is present in every board game are the interpersonal connections between the players. Losing to specific member due to poor luck easily incites strong feelings. Alexa can also be programmed to address this particular social issue by offering a simple “calm down“.

3.2 Social Issues

Games or Board Games in particular are key to social engagement. It helps to bring a sense of togetherness in a group of people.Immersion and clear flow of games and lucid rules improve the enjoyability of the game.These key factors should not be compromised during any service that hopes to add on to the board games. Alexa needs to act in a quick and prompt manner so as to allow seamless integration in the game. Furthermore, board games are generally enjoyed as a way to kill time. As such, we cannot make our system cut down too drastically on playing time.

3.3 Cognitive Issues

Cognition is the key-tool to engage users in any social engagement. To describe a game as an activity, it should have certain attributes - physical or mental skills and strengths, a specific set of rules to attain a goal. Cognitive tools assist in constructive thinking and aids in performing conceptual operations beyond the ability of a participant [5]. With *CleverName*, we take out the factor of immersion from the *Monopoly*. An intelligent user interface in the form of an additional element might cause the participants to detract from the game. Also, the advice from an autonomous system could take out the "fun" and "thrill" of the game because participants being advised under certain circumstances. These two factors, if strong enough, can result in the loss of cognitive characteristics of the board games. On the flip side of the coin, assessing the credibility and validity of Alexa’s advice can help develop financial reasoning skills.

4 DESIGN

4.1 Overview of Monopoly

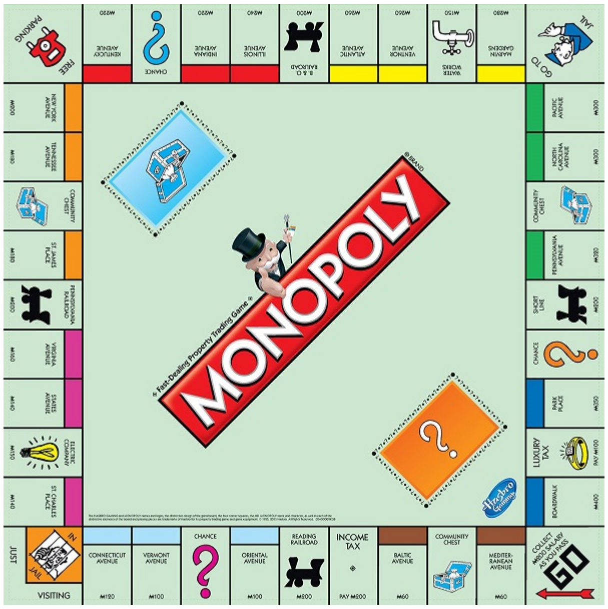

Figure 1: Monopoly Board [8]

Before we go ahead and describe about our system design, we want to give an overview of Monopoly as a board game. Monopoly can basically be played by two to eight players. The game consists of a board, two dices, player tokens, 32 houses, 12 hotels, Chance and Community Chest cards, 12 property cards and artificial playing currency (money). At the start of the game, each player is assigned a total of 1500 worth of corresponding currency - 2 * 500 notes, 2* 100 note, 2 * 50 note, 6 * 20 notes, 5 * 10 notes, 5 * 5 notes and 5 * 1 notes. All remaining money and properties reside with bank initially. Of all the players, one player role plays as Banker whose responsibility apart from playing the normal game includes conducting auctions, perform transactions of property and money. General rules of the game are as follows:

(1) Initially, all the player’s tokens are placed at ’Go’ square.
(2) A player needs to roll the dices on his/her turn and move his token to number of spaces corresponding to the count he/she obtains on the dices.
(3) If the player lands on an unowned property, the player can buy the property for the amount listed on the property card. If he refuses to buy the property, the bank conducts an auction and the highest bidder wins the property.
(4) If the player lands on an unmortgaged property owned by another player, then he/she pays rent amount equivalent to as specified in the property card, to that person. It is the responsibility of property owner to demand rent before the next player begins his/her turn.
(5) If the player lands on his/her own property or a mortgaged property owned by another player or on ’Free Parking’, nothing happens.

(6) If the player lands on Luxury Tax or Income Tax, he/she needs to pay the tax amount as specified in the game rules to the bank.
(7) If the player lands on a Chance or Community Chest, he/she picks up a card from the corresponding card deck and perform instructions as specified on the card.
(8) If the player lands on the Jail Space, he/she is just visiting and nothing happens. However, if the player lands on 'Go to Jail' square, the player's token is moved to jail and can get out of jail according to specific rules as described in the game book.
(9) If the player lands or crosses Go, he/she receives 200 money from the bank. Until next player's turn, the player can collect the money.
(10) Player can sell houses back to the Bank for half the purchase price or sell property to other players in the game.

The game starts with banker and other players following in the subsequent turns. Players gradually buy out properties from Bank and start building houses/hotels on it so as to collect more rent from other players. Player with not enough money either can mortgage his/her property with bank or sell the property to another player for any amount the owner can get. The player gets bankrupted if the amount of money he/she owns to the bank or other player is more than he/she can pay. A bankrupt player must immediately retire from the game. The last player left in the game wins.

4.2 Rationale

The goal of our system was to create an assistant, using Amazon's Alexa as our vehicle, to create a more interactive Monopoly assistant. The assistant would help players manage their properties, maintain bank accounts and advice players on decisions to make. Like a real-life assistant, our system should not hinder the player but accelerate the task at hand. As a result, we need to have voice commands which do not require repetition in the same time frame. We also require a way to correct and check if the assistant's current knowledge of the game is correct or incorrect. Our system addresses these issues with the following design.

4.3 System Design

We started with developing on Amazon Web Services for integrating Alexa and Android application with our back-end functionality.

4.3.1 Backend Database. In order to keep the database usage simple, we used DynamoDB as database for storing game state and player information. DynamoDB is a fast and flexible NoSQL cloud based database service that supports both document and key-value models for storing data. We created two tables to store the game and player information. The game information table included game unique identifier, user identifier, game time information, number of players in the game, identifier of the current player and identifier of the next player in the game sequence. This table serves the purpose of identifying a particular game version's information and retrieving it using the user and game version name information. This information was also essential to save game at any point of time and resume later. The player information table consists of all the attributes of a player - player identifier, amount of money and properties he own, current position on the Monopoly board, and player's position for rolling dice.

In order to maintain game state with real time game positioning, we developed a back-end API (Application Program Interface). The back-end API enables both Alexa and Android Application to communicate with the database. Currently, we provide 4 basic functionality for each of these two tables :

- **Read** : Allows the current game state and player information to be read from the database.
- **Create** : Allows adding new game and new players.
- **Update** : Allows banker or players to update game state or player information through either mobile application or Alexa.
- **Delete** : Configures the deletion of a particular game after certain days after game has ended.

4.3.2 Alexa Skill - Monopoly. Developing a skill set for Alexa requires building it on AWS Lambda which is event triggered serverless computing platform by Amazon. We needed a custom interaction model which includes following inputs :

(1) Intents - An action that fulfills a user's spoken request. An intent has slots to distinguish between different variables like game name, player name, count of dice etc.
(2) Sample Utterances - A list of likely spoken phrases mapped to the intents.
(3) Custom Slot Types - A representative list of possible values for a slot for an intent. Custom slot types are used when Amazon's built-in slot types does not satisfy the criterion.
(4) Dialog Model (Optional) - This feature allows for multi-turn conversation between the Alexa skill and the user. This helps in making the code for a particular action more simpler.

We built 7 custom intents for implementing Monopoly on Alexa as shown in Table 1. The name of our Alexa skill is 'Monopoly'.

A lambda function was linked with this Alexa Skill Set. A singular function call is made every time a call to the corresponding skill set is spoken. There were several problems encountered as we developed Alexa skill :

(1) Short session - Alexa session times out in 8-10 seconds. This requires calling Alexa with 'Monopoly' skill name every time if the commands are spoken after the session time out duration.
(2) Accent Recognition - Alexa cannot figure out the different accents, so, it was hard for us to test different intents.
(3) Incorrect Slot value - The input request generated by Alexa skill sometimes incorrectly identifies slot values.

4.3.3 Android Application. Our system features an Android application which allows for game state updates independent of Alexa. As it stands now, the Android application is meant to be used by the banker in monopoly to keep track of the game states. This feature was implemented to address several issues which we believed detracted from the Monopoly experience when an Alexa is being used.

The first issue we addressed it that it became tedious to keep asking Alexa to update states at the later stages of Monopoly. It becomes

Intent	Command	Action	Players
startMonopoly	Alexa, ask Monopoly yo Start game with the name gameName for numPlayers players.	Starts a game of monopoly for numPlayers players	Any
introducePlayer	Player playerNum name is playerName	Insert names corresponding to each player	All (in order of their rolling dice sequence)
updateGameState	Alexa, ask Monopoly playerName rolled a rollCountOne and rollCountTwo	Updates the state of the game corresponding to roll counts and notifies player regarding his current position and property updates	Any
payMoney	Alexa, ask Monopoly FromPlayer gave money dollars to ToPlayer	Transfers money from one player to another	Any
buildHouse	Alexa, ask Monopoly playerName wants to build numHouses houses on propertyLoc	Allows player to build houses or hotel on the given property	Any
getFinancialAdvice	playerName wants financial advice	Provides financial advice to the given player	Any
showProperties	Alexa, ask Monopoly Which properties are owned by playerName	Speaks out the properties owned by the given player	Any

Table 1: Alexa 'Monopoly' skill intents, commands and corresponding actions

easier to use the Android applications to make changes. Furthermore, our system does not support the trading of multiple properties simultaneously; an occurrence that becomes exceedingly common towards the end of the game when rent on property is incredibly high. The application streamlines the game state update process and lets the group focus more on the game itself instead of waiting for Alexa to update its system. The second issue the application addresses is to ascertain correct changes are being made by Alexa. The voice recognition and language parsing software is not always accurate for custom Alexa skills and may incrementally misrepresent the state of the game as we use voice to make updates. The Android application allows us to rectify these mistakes as well.

The application features a simple UI that shows all the Monopoly property squares on the board that can be purchased. The properties are arranged vertically by their cost and horizontally by their property class. Each class is color coded like the original game and each properties name is displayed. When a property is clicked, various attributes associated with the property are displayed in a new activity. Currently, we display the owner, whether the property is mortgaged and the number of houses or hotel presents. The Amazon Cloud Dynamo DB contains all the current attributes and allows the banker to check the correctness of the current game state. Each click establishes a connection with the backend and sends the current game id and the name of the property of interest. The backend responds to the android application with the attributes associated with that property. Once the banker changes the attributes clicks the "Update" button, the changes are reflected in our online database before returning to the main UI.

4.3.4 Financial Advising. Financial Advising is an open problem. We have hedge funds and professional financial advisors whose sole job is to advise people on this front. In a game, such a monopoly the risk is not high. The worst thing that can happen is a player being evicted. The lowering of stakes can help us in avoiding the cold start problem. The cold start problem is identified as such: Suppose we have players starting the game. To avoid potential short-term loss of money nobody will be willing to spend any money. This will stall the game's progress. In essence, in a game such a monopoly we can tilt the risk versus reward ratio more favorably in terms of risk as compared to hedge funds. As such, we have to make efforts on making the game more enjoyable as well as providing sane financial advice.

We implement two strategies for financial planning. These two strategies are just a starting point as emphasized earlier, financial advising is an open problem and more complex use of statistics can be used to do financial planning.

Our first strategy just advises people to buy a property if they can that has the requisite capital and if the property is free. This is done until the capital of the player becomes half of its initial value. After that, the advice for buying becomes more complicated. This idea for such a strategy comes from TCP(Transmission Control Protocol) which does congestion control thorough exponential backoff. As the network becomes congested the clients reduce their requests until the network congestion reduces below a certain threshold. The second strategy is more complicated and relies on the assets and liabilities of a particular player. We calculate the number of turns in which a certain player becomes bankrupt assuming he buys the property. if this number of turns is greater than 10 we fee that as a distant future and advise the player to buy the property. Now an interesting extrapolation of the problem is the number we consider as distant future. We picked 10 just as an experimental number. The beauty of the idea is the number can be adjusted according to the players and time. These lower number would mean the game could reach its conclusion in fewer steps. A more complicated algorithm would be calculating the probabilities of landing on each of the blocks of monopoly and using these probabilities to calculate the overall number of assets or liabilities. Of course, we still have many variables like the chance card which can be considered an asset or a liability. we can implement another strategy which is just a summation of all the good practices to be followed such as always buy railroads never buy utilities and other things such as For every other property type, only buy them to complete a monopoly or to prevent opponents from completing one. Often this may mean buying as many properties as you can early in the game. At the beginning of the game, focus on acquiring a complete C-G [Color Group: all 2 or 3 properties of the same color] in Sides 1+2, even if it means trading away properties on Sides 2+3. After acquiring one of these C-Gs, build 3 houses as quickly as possible: no more houses, no less! Once your first C-G starts to generate some cash, focus on completing a C-G and building 3 houses in Sides 3+4. Single properties are the least good investment if you donâĂŹt build on them.[4]

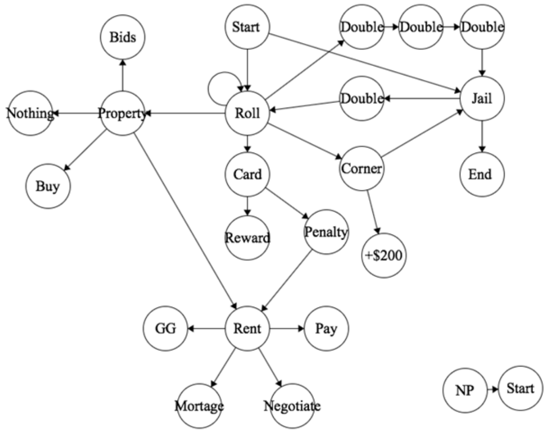

Figure 2: Block Diagram of approach.

5 IMPLEMENTATION

We started with developing on Amazon Web Services for integrating Alexa with our back-end functionality. Developing a skill set for Alexa requires building it on AWS Lambda which is event triggered server-less computing platform by Amazon. In order to keep the database usage simple, we used DynamoDB as database for storing game state and player information. DynamoDB is a fast and flexible NoSQL cloud based database service that supports both document and key-value models for storing data. We created a master table/collection to store the state of each game and associated player information. In player information, we have currently included name, current account balance, current position, and the properties owned or mortgaged. We have used a game identifier for each game to distinguish multiple concurrent game state maintenance.

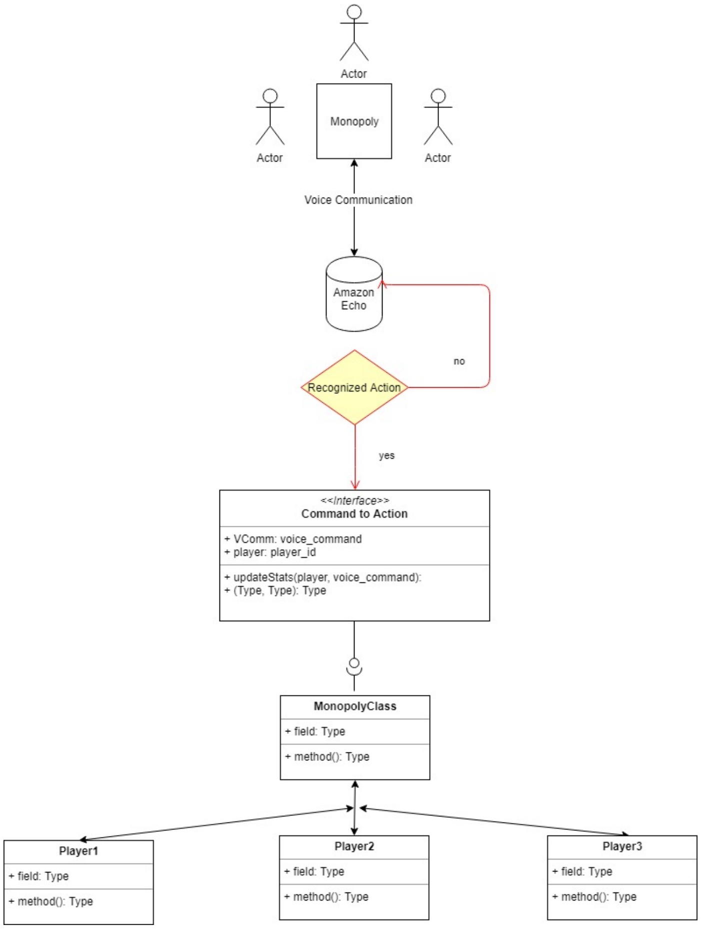

Figure 3: Block Diagram of approach.

In order to maintain game state with real time game positioning, we developed a back-end API (Application Program Interface). The back-end API enables both Alexa and Android Application to communicate with the database. Currently, we provide 4 basic functionality :

- **Read** : Allows the current game state and player information to be read from the database.
- **Create** : Allows adding new game and new players.
- **Update** : Allows banker or players to update game state or player information through either mobile application or Alexa.

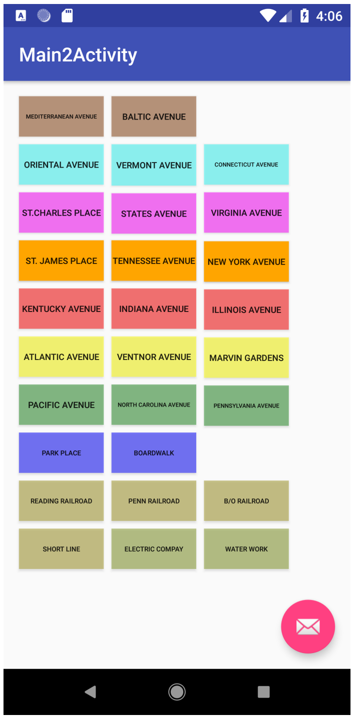

Figure 4: Main UI Screen with Property Buttons

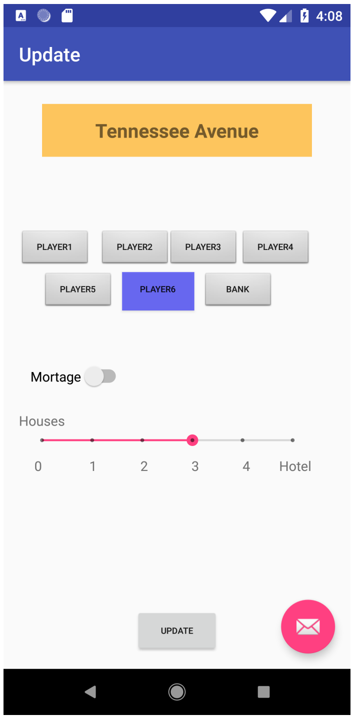

Figure 5: Property Information for Tennessee Avenue

- **Delete** : Configures the deletion of a particular game after certain days after game has ended.

Concurrently, we also started building our android based mobile application. We initiated our design with displaying a grid layout of the monopoly board as shown in figure 7. The layout consists of 40 rectangular blocks of the Monopoly board. Each block represents a different property or utility or significance as in the game itself. To simulate this, these blocks are click-able buttons which navigates to a different screen showing its significance value in the game. The block screen as shown in figure 4, contains property or utility information along with the associate owner and its monetary value. We are planning to enhance this board by making it visually more attractive and appealing for better user interaction. Particularly, we wish to display the number of houses, the owners of each property and other such information in a visual manner to better assist player based decision making.

6 EVALUATION

In order to assess the effectiveness of our system, we need to create an evaluation framework focused around player testing. Our proposed evaluation would take 10 groups of players and have them play the game. The groups will be split into two 5-group sections, Group A and Group B. Group A will proceed to play the game without our financial advisor. The players will be asked out to fill out a generic survey about how they felt about the game and would rate the overall âĂIJfun factorâĂİ on a scale of 1-10. The same group of people would return a week later and be asked to play the game but with our system this time. The players would then fill out the following survey:

(1) The application made the game more enjoyable
(2) The application was easy to use
(3) The financial advice offered was relevant and helpful
(4) I want to follow the advice given by Alexa
(5) I would use this application again when playing Monopoly
(6) I

The final question would be the overall fun-factor question on a scale of 1-10. The rating of the question above are based on a subjective scale of Highly Disagree, Disagree, Neutral, Agree and Highly Agree. Group B will follow the same process but will play with our system first and then without the system.

The goal of this evaluation framework is to eliminate order effects. Order effects are generally hard to control but if in both conditions, players indicate that they enjoyed a game on Monopoly more with our system, then we were successful in implementing an effective Monopoly Assistant and Financial Advisor. If a group of players enjoy it more than the other group, then liking our Assistant system is based on the usage of utilizing it before or after the non-Assistant trial. As with all new computer applications, there is a learning curve which needs to be taken account. Players may not actually like the application until they become comfortable with using it. Our evaluation schema can be easily extended to have players play Monopoly multiple times with the application and can be tested for in future iterations of this system.

7 CONCLUSIONS

Our system successfully implements an Alexa-based assistant for Monopoly. It utilizes an Amazon AWS back-end to compute various parameters and offer feedback and advice on the current state of the game. The back-end connects to a DynamoDB database which can be updated with the Alexa voice commands and an Android Application. An android application was ultimately created to mitigate some usability issues presented by the voice-recognition software in Alexa and allows for an Alexa independent mechanism to update the game state in the database. Finally, financial advising was implemented to give players feedback on when to make a purchase. The system works well as long as there is a stable Internet connection present for all devices. An evaluation framework was presented to test the efficacy of our assistant system. The evaluation framework attempts to control for order effects and proposes expected data to indicate a effective versus ineffective system.

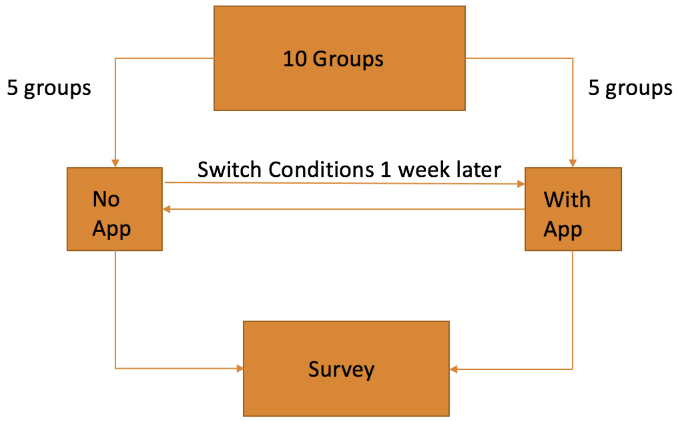

Figure 6: Proposed Evaluation Framework

Some challenges that were faced during this work that would drastically affect our design choices in future implementations. The biggest problem faced was our inability to test the Alexa system due to the nature of the group members accent. Only one member's speaking mannerisms could be picked up by Alexa every-time. As such, much of our testing was done manually by feeding in data in a text format. Another challenge we faced was interfacing all the different parts of the system. While plenty of documentation exists to connect Alexa to a backend and database system, very little consideration is given to interfacing Alexa directly to a smart-phone without publishing our Alexa skill-set. As such, we had to create roundabout mechanisms in the back end to facilitate communication. We would look at competitor products like Google's Assistant software, Cortana or Siri instead if we were to redo this work. With regards to the application, the overhead for the UI was significantly higher than expected due to the large number of buttons that need to be generated when switching between the UI screen and the property screen. One way to rectify this is to utilize a static canvas template instead of the button and find other ways to start a new activity. This would create a less resource hungry UI and preserve battery life for the system as a whole.

8 IMPROVEMENTS AND FUTURE WORK

As the system currently stands, many improvements can be made. Many of these improvements fall into three distinct categories and possible implementation for each presented.

The first category involves improvements made to the Alexa interface. As it stands, Alexa has a hard time recognizing accents that are not traditionally English. We propose a machine learning model to train Alexa to recognize Monopoly specific commands which can be used to facilitate a more diverse group of Players. The model would require us to collect several sound-bytes of players saying various Monopoly property names, property values, such as mortgage, owner, buy, sell etc., and other actions like send to jail. This task is easily accomplished utilizing a neural net and is a subset of larger problems in Natural Language Processing. The second Alexa interface modification we propose is to utilize better structured commands. At the moment, Alexa commands have be to given in a very specific order with regards to variables. For example, the location of the owners name or the number of houses a player wishes to build need to be given in a very specific order to work.

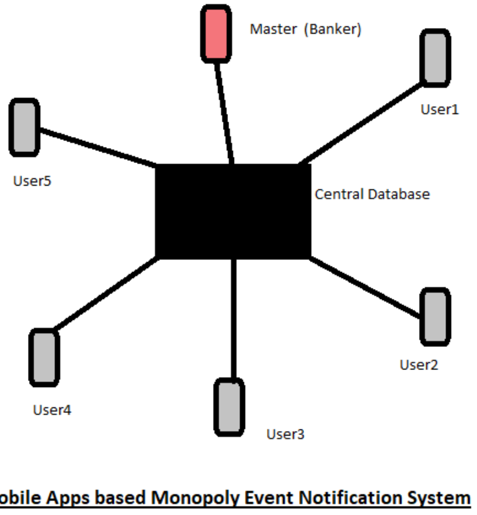

Figure 7: Block Diagram of approach.

The second category of improvement is meant the mobile application. We had several ideas to implement functionality that would better enhance the game in the long run. Ideally, we would create a system where all players utilize their smart-phones to play the game. By introducing a in-game chat functionality, we can allow players scheme with each other via back channel communication without other players in the game being aware of it occurring. This creates interesting dynamics which could potentially lead to better game play. Another nuance that can be introduced is to allow players to keep track of each other money manually on their application. Knowing how much money other players can become a vital negotiation tool. Players who take the extra effort to track their opponents cash flow will be in a better position to haggle and make better decisions.

The final category of improvement is to improve our financial advisory system. The advisory can become more complex and give players information to better base their decisions on. This would be a significant step up over a simple buy or sell property. The first idea we had was to create heat-maps for each players risk in their upcoming turn. These heat-maps can easily be accessed and personalized for each player utilizing the system present in figure 6. The advisory system can get as complicated as the person coding it chooses. Several financial investment strategies exist and can be easily extended for monopoly.

REFERENCES

[1] Paul Cairns, Anna L Cox, Matthew Day, Hayley Martin, and Thomas Perryman. 2013. Who but not where: The effect of social play on immersion in digital games. *International Journal of Human-Computer Studies* 71, 11 (2013), 1069–1077.

[2] Fabio Chiarello. 2016. Board Games and Board Game Design as Learning Tools for Complex Scientific Concepts: Some Experiences. *International Journal of Game-Based Learning* 6, 2 (2016), 1–14.

[3] Robert Dale. 2016. The return of the chatbots. *Natural Language Engineering* 22, 5 (2016), 811–817.

[4] Tim Darling. 2007. Monopoly Rules. (2007). http://www.amnesta.net/monopoly

[5] Jan G Hogle. 1996. Considering games as cognitive tools: In search of effective. (1996).

[6] Karen Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah L McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe. 2007. An intelligent personal assistant for task and time management. *AI Magazine* 28, 2 (2007), 47.

[7] Drake Peter. 2011. Teaching Introductory Programming With. *Proceedings of the 42nd ACM technical symposium on Computer science education* (2011).

[8] Hobby Warehouse. [n. d.]. Monopoly Classic Edition Board Game. ([n. d.]). https://www.hobbywarehouse.com.au/monopoly-classic-edition-board-game.html