Andrew Murray

Registration number 100214063

2021

INVESTIGATION AND SIMULATION OF CARD COUNTING METHODS

Supervised by Prof Elena Kulinskaya

University of East Anglia
Faculty of Science
School of Computing Sciences

Abstract

This document shows performance of card counting methods against casino rules.This involves analysing the methods against parameters that are designed to replicate a casino environment. This document will explain my approach and execution to achieve this. The project holds the main of identify the best method for a variety of situation by analysing its yield and betting correlation and consistency.

Acknowledgements

I would like to thank my supervisor Prof Elena Kulinskaya for the support given to me through the project. I would like to thank him for his constructive criticisms and discussions about inner workings and designs of my peer sourcing implementation

Contents

A. Introduction **6**

B. Context and related literature **6**
B.1. Motivation 9

C. Notebook **9**
C.1. Milestones 10
C.2. Meetings 11
C.3. Time allocation 11

D. Design **14**
D.1. Design of the project 14
D.1.1. Blackjack rules: 14
D.1.2. Blackjack Strategy: 16
D.1.3. What it hopes to accomplish: 17
D.1.4. Number of players in play: 17
D.1.5. Number of rounds: 17
D.1.6. Starting chips: 17
D.2. Minimum betting value: 18
D.2.1. Shoe size: 18
D.2.2. Reshuffle point: 18
D.2.3. Penetration: 19
D.2.4. Method selection: 19
D.2.5. Betting: 20
D.2.6. Calculating performance: 21
D.2.7. Requirement list: 21
D.2.8. Managing my project: 22
D.3. Design of the implementation 22
D.4. Simulation Plan: 25
D.5. Commentary on Issues: 25

E. Analysis and results **27**
E.1. How I analysed my results? 28

E.2. Overall performance of methods: 28
E.3. Number of players: 31
E.4. Shoe size: 31
E.5. Minimum betting value: 31
E.6. Penetration: 32
E.7. Shuffle rate: 32

F. Testing **32**
F.1. Unit testing: 32
F.2. Requirement testing: 32
F.3. Visual inspection: 33

G. Discussion, evaluation, and conclusion **33**
G.1. lessons learnt: 33
G.2. Evaluation: 34

H. Conclusion **36**

References **37**

I. Appendix **38**
I.1. Basic Strategy 38
I.2. UML 39
I.3. Simulated round 39
I.4. Simulation game 40
I.5. UML Diagram 40
I.6. UML Diagram zoomed 41
I.7. Updated Gantt Chart 41

List of Figures

1. Project Gantt chart 42

A. Introduction

Succeeding a progress report on 'Investigation and simulation of card counting methods, the project has been completed to compare the performance of the methods across various parameters employed by casinos. Thereby, analysing and evaluating their respected data to attempt to shift the 'house advantage'. Furthermore, the project portfolio will effectively distinguish the cases where one method may be superior to another across different circumstances using the quantitative data from the simulations. Despite primary focusing on the yield of a method, consideration is taken for its complexity to reflect more real-world scenarios. Moreover, comparisons to existing studies are made to evaluate my simulations. Lastly, this portfolio will steer the project aim to a conclusion.

B. Context and related literature

During my prior investigation into the project, various studies were found involving card counting methods in blackjack. Specifically, focusing my research on the yield, betting correlation and player efficiency of various methods. When developing the scope of my project, the focus was on yield and betting correlation. Thorp, E. (1969), uses the word yield in the context of the return the method grants you over a period and Vancura, O. and Fuchs, K. (1998) defines betting correlation (BC) as to how well a card counting system predicts good betting situations.

Haigh, J. (2010) published a mathematical equation to measure betting correlation, as well as publishing the respective BC of the 4 of the 7 selected methods chosen which provide additional data to evaluate and analyse against my own. Those methods being, Optimum, Hi-Lo and Halves 2, KO count methods. However, to conduct the mathematical equation, requires parameters outside the scope of my equation. Therefore, I will be using the optimal method which has a betting correlation of 1 as a baseline to determine whether a method has missed an opportunity of a good betting situation. Thereby, still fitting Vancura, O. and Fuchs, K. (1998) definition of betting correlation and utilizing Haigh, J. (2010) optimal method to do so, by establishing what a good betting scenario is as it accurately measures the effect of removing a card.

During this project, key distinctions have been made to understand the factors the makes one method different from another. The importance of these distinctions was enforced when evaluating the data to analyse why the results differ from another. According to Haigh, J. (2010), the differences in methods consisted of three main aspects: the ‘Type’, the ‘Level’ and parameters. The ‘Type’ refers to whether a method is balanced or unbalanced, which is determined by whether adding all the assigned values of the cards reaches a net 0 (balanced if it does). As previously stated in my literature review, the ‘Level’, refers to the assigned values of the cards, such that assigned values between -1 and 1 would be ’Level-1’ and values between -2 and 2 would be ’Level-2’. Lastly, the parameters refer to the number of parameters a method entails, for example, the K-O system involves a single parameter which is the running count whilst other methods may involve also tracking the number of aces played.

A balanced method, as previously mentioned, needs adjusting to account for the size of the shoe. This is done, according to Blackwood, K., (2006), by taking the running count and dividing it by the number of decks remaining. Unbalanced methods have a different starting running count to account for this.

In terms of the implementation of rules to simulate an accurate version of blackjack with casino rules. The rules were set using Haigh, J. (2010). This decision was made as to the academic material specifically focused on the science of casino blackjack and provided insight to specific rules employed to run the game as well as shift the favour towards the house (e.g., penetration and reshuffles).

In addition, the ‘Basic Strategy’ was also inspired by the same paper which can be seen in FIGURE. For the simulations to act according to the methods, it must have a strategy to deal with its hand and act accordingly. This ‘Basic Strategy’ varies depending on the shoe size, however, provides the template for whether the simulation should hit or hold its hand. This strategy includes exceptions to decrease the edge of the dealer in terms of whether they have a ‘soft’ hand or not and depending on what the dealer has down. Doing so, optimizes the winning probability for different hand combinations, improving the win rate. This is referred to as ‘composition-dependent by Thorp, E. O. (1966). However, due to the nature of the project attempting to reflect the real game, where players may struggle to adopt a method alongside additional mental calculations to their basic game, this was not included in the strategy. Furthermore, doubling was also not considered in the simulations to avoid complications where casinos will impose stricter rules on a player if they suspect card counting, to which doubling with favourable odds would raise red flags.

Despite being included in the ‘Basic strategy’, Vancura, O. and Fuchs, K. (1998) and Haigh, J. (2010) established that Double after splitting (DAS), surrender and re-splits was not an option in most casinos which led to the decision to leave it absent in the project.

Thorp, E. (1961), explained that if betting rates were constant then a player would be at a severe disadvantage, meaning a players bet was variable then they would be at a greater advantage than the house. However, this arose a problem of various methods having variable betting, where a method with a bigger betting spread could have a greater advantage if its bets more aggressively according to its ‘true count’ or ‘running count’. Nonetheless, equally if it losses and is inaccurate then it has a greater chance of going bust which is why the methods in the project have their betting strategies unchanged. Although, Vancura, O. and Fuchs, K. (1998) note that a higher betting spread may cause the casino to impose stricter rules if they suspect a player is card counting. An obvious difference in the spread is the 1-5 betting spread for Hi-Lo compared to 1-32 for KO count. Therefore, the project evaluates ‘more obvious’ methods ability to deal with stricter rules which were also stated by Werthamer, N.R.,(2008).

According to Werthamer, N.R., (2008), Casino’s will also penetrate the shoe (collection of decks), which entails getting rid of part of the shoe at the start of the game to discourage card counting. James, C. (2019) states that could potentially be around 33 per cent of the shoe, making it more difficult to card count as your true count or running count will reset quicker as the next reshuffle point would be closer. However, this rate will vary between casino’s and individuals working there, which makes it difficult to predict accurate penetration levels. Furthermore, the human aspect of penetrating the shoe means it is unlikely even the dealer knows what per cent of the deck was cut. Moreover, the shoes are reshuffled before all the cards run out which also leads to varying reshuffle points as well.

B.1. Motivation

To further provide context for my project, elaborations have made on the purpose of the project. It focuses on the human aspects of these card counting methods, whilst some methods are not designed for use by human players at a casino, they are included to provide insight into the reason for the performance of other methods as well as itself. Furthermore, it would have been outside the scope of the project to attempt to weigh the complexity of a method against its performance accurately. Therefore, the complexity of the method is not measured but considered in the evaluation.

Secondly, a key aim of the project was the discovery of scenarios where a method type (Level, Type and Parameters) or method would have an advantage, created by varying casino game rules. Thereby, through analysis, an optimal method for a particular scenario can be concluded within the ability of a human player. Additionally, the results would reflect how a method can shift the odds from the house to the player.

Furthermore, the combination of simulations rules is designed to copy a real-world game an individual may face at a casino. Thereby, allowing the data to have real-world application. This is the reason the methods utilize ‘Basic strategy’ and not the optimal generic strategy.

C. Notebook

Whilst undertaking this project, a notebook has been updated on a weekly basis of the progress made. The updates include the time spent on the weekly tasks, important milestones, supervisor meetings and schedule changes as well as my notes if needed on that week. A summary is given to summarise aspects of the notebook.
In terms of my most important Milestones, despite having various, there are 5 key moments outlined below that defined my project.

C.1. Milestones

In terms of my most important Milestones, despite having various, there are 5 key moments outlined below that defined my project.

Milestones: Literature review: This was the precursor to all other milestones, and incredibly important as it established the direction of my project as well as allow me to begin to understand the necessary tools, I would need to complete the project. Furthermore, it marked the gathering of vital information from various studies needed.
Progress report- Another important milestone accomplished during my project, it provided me with the tools to realize the weaknesses in my design as well as demonstrate my plans and work towards the project. Furthermore, it help address many concerns surrounding my project through the preparation leading to this report.
Completed code- This was an incredibly important milestone as it formed the basis of my project. It involved overcoming multiple bugs as well as involved all the research I conducted to create simulations that lived up to my vision of the project. Furthermore, it incorporated many other aspects of my project such as the design, the established rules and methods as well as the ‘players’. Additionally, it provided all the raw data to be analysed and evaluated in the project portfolio.

Completed project portfolio- Arguably the most important milestone as it marks the end of the project and is what was being toward from the start of the year. This milestone is the collection of the effort that has gone into the project.

Testing: Unfortunately, my code had a few bugs which I missed on the initial phase of testing as the results did not reflect these bugs until certain scenarios were simulated. I was using an iterative approach and initially the values seemed fine, therefore, later on the project I had to go back to testing which was incredibly important as it meant I was behind schedule going back to my code.

C.2. Meetings

Meetings: In terms of the most important meetings with my supervisor, there are 5 key interactions that were vital to my project.

6/10/2020-12/10/2020 (first meeting)- This meeting was incredibly important as it is the first time I was introduced to Prof Elena Kulinskaya . Furthermore, I struggled to find a direction for my project to which the meeting was very insightful on informing me about key things to consider such as my parameters and what I wanted to measure.
27/10/2020 to 31/10/2020 (Literature review)- This meeting involved discussing my literature review to which feedback was given and highlighted key weaknesses in my research and my misunderstanding of what a literature review accomplishes. Overall was an incredibly important meeting to get me on the correct path.
14/11/2020 to 20/11/2020 (Scope issues): At this stage, the meeting was set up to discuss my design of the project to which a few draft UML and class diagrams were made. However, it uncovered issues in my scope of the investigation in terms of how many simulations I wanted to run.
28/11/2020/ to 4/12/2020 (Progress report): This meeting was prior to my progress report and helped me evaluate my own progress and clear up the aim of my project as it was still to some extent ambiguous.
05/02/2021 to 11/02/2021(Project still has scope issues): It was alerted that my proposed Monto Carlo simulation was unnecessary and did not make sense. I also was told I was doing too many simulations, to which certain parameters were lessened/removed. This meeting allowed me to combat those issues in time for the project portfolio.

C.3. Time allocation

Time spent on tasks
Semester 1
Project proposal Week 1- On the first week of the project, I began to complete the project proposal task. This involved hours of researching related academic material on card counting methods for blackjack. An understanding was needed so I could effectively come up with an approach to the project.
Week 2- Utilizing the research done on week 2, I wrote the project proposal with a basic

idea of what was needed to complete the project (research, design of the system, coding and data collection, and analysing results). Furthermore, proposed a primitive idea of my current project.

Literature review Week 3- This week marked the start of my literature review and consisted of mainly researching work surround card counting methods in blackjack as well as casino rules. This research led me to focus on a card counting method in a casino environment.
Week 4- I made the mistake of treating my literature review as utilizing the research which was discussed in meeting with my supervisor which meant I had to redo portions of it. However, demonstrated a good understanding of the areas of knowledge needed.
Week 5- Literature was redone and completed successfully and my meeting confirmed the improvements on addressing key issues and themes. As well as completing my selection of card counting methods to use that are diverse using my research. I was also inspired to investigate whether I could somehow quantify complexity to measure against a methods performance.
Week 6-8 consisted of further research to make a more informed decision on design choices and direction of my project to prepare for the project review.
Week 9-11- I was a little bit behind on designing how I wanted to process the data due to extenuating circumstances. However, had created functioning Deck and dealer class (coding). Furthermore, I had complete UML diagrams as well as a project plan on Trello to help me manage my project.

Project review
Week 11-13- I had spent my time progressing my program to run the simulations as well as completed my project review. Also had manage to successfully run my first simulation, however some parameters were missing.

Semester 2: Week 1-3: Due to other commitments with an exam and coursework's, work during this period was slow. However, managed to have 2 full working simulation methods working- Hi-Lo and KO-Count.
Week 4: Despite not having all the methods yet, at this point all parameters are coded for the simulation.
Week 5-7: All methods can successfully run in the simulations.

Week 8: Whilst testing my code, I had a bug that would let methods bet even if they had 0 betting units left. I originally did not notice as I test whilst I code and it would make the funds back after going into a negative in the earlier rounds, so this was fixed. Furthermore, I had originally attempted to control the betting strategies by not touching the betting boundaries but limiting the betting spread, it was clear from the results that to make it fairer for other methods I had put that method at a disadvantage as it may perform better using a more aggressive betting stance. Moreover, it also meant it has a higher likelihood of bankruptcy which I am considering so betting values were restored. Week 9-10- Had to recollect my data after a bug that the true count was accurate as shoe size did not change with the input I had entered the simulation. Therefore, most of the data was invalid and had to re-collect. This set me behind for my report and poster, as I had also been working on other work during this period.

D. Design

Following my motivation and context for the project, the design was split into two categories to achieve its purpose. The first being the design of the project, which is crucial to outline conditions needed to achieve the desired data that is being analysed and evaluated. The second being the design of the application (implementation) that would simulate the card counting methods playing Blackjack. This approach was adopted to attempt to mitigate errors and improving the chance of success by following an appropriate plan.

Firstly, the design of the Blackjack game was done to establish the environment needed for the project. This involved solidifying what the 'casino rules' are when it comes to Blackjack and the strategy needed to play the game, regardless of the card counting methods.

D.1. Design of the project

D.1.1. Blackjack rules:

A key aspect of the project is the 'casino' environment rules. Therefore, through my literature review, Haigh, J. (2010) was utilized to establish the Blackjack rules for casino's being, which was supported by relevant literature such as:

- In a casino, the cards are dealt by a casino employee, here called Dealer.
- The simulation will play as dealer and players but act as they would in that scenario.
- Players can vary from 1 to 7.
- Casinos can use from 1 to 8 shuffled decks, which is referred to as a shoe, depending on the table, each deck consisting of 52 cards.
- Dealers reshuffled when they wish which adds ambiguity to the simulations. James, C. (2019) suggestion that reshuffles point are typically close to 33 per cent of the cards played will be used as a suggestion for the simulations.

- Each player plays against the dealer ('the house')
- Each card has an associated value, to which the value on the card is the number on the card. However, the exception is 'King', 'Queen' and 'Jack' which are valued are 10, and 'Ace' can be valued at either 1 or 11 depending on the player's choice. The suit of any card does not matter.
- Each card has an associated value, to which the value on the card is the number on the card. However, the exception is 'King', 'Queen' and 'Jack' which are valued are 10, and 'Ace' can be valued at either 1 or 11 depending on the player's choice. The suit of any card does not matter.
- At the start of every round, a player put at least the minimum bet size to play. However, some casino's also have a maximum bet size. The amount bet can vary from round to round depending on the player's choice/chips remaining.
- Once all players have placed their bets, cards are dealt one by one to each player starting from the player on the left (this will be player 1 in the simulations) then one to the dealer which are all made face-up. This process is repeated but each player is given a face-down card to which they can decide whether they want another card 'hit' or stay with their number 'hold'. The aim being attempting to get closer to hand value of 21 than the dealer without exceeding that value using the knowledge of the Dealer's up card.
- Then all cards are made face-up and compared to dealer. If a player gets Blackjack and the dealer does not (an ace card and another card of value 10), then it is paid out at 3:2.
- The dealer must reach a value of 17 or more they must stand, whereas if it is 16 or less, she must draw cards until the value exceeds 16.
- If the dealer has busted (value of 21) then all players are paid 1:1 if they have not busted.
- If a player has the same value as the dealer, then it is a tie, and the bet is recovered but nothing is won.

- During the card comparisons, if a player has a value greater than the dealer's but less than or equal to 21 then it is paid out at 1:1. However, if the value is greater than 21, regardless of the dealer. Lastly if the value at the end of the round is less than the dealers (assuming dealer has not busted) then the bet is lost.
- Players can only join the game at a reshuffle point.
- No insurance and surrender or doubling after splitting (DAS) as some casinos don't accept it

The rules were also supported by relevant literature such as Wen-Jen Hsu Bin-Tzong Chie (2021) and Tulcea, C. I. (1983).

D.1.2. Blackjack Strategy:

The basic strategy used in this project is inspired by Appendix 1 (Haigh, J., 2010). A strategy is essential as it simulates the player choices are given around and circumstance. Despite DAS, surrender and insurance being included in its strategy, Vancura, O. and Fuchs, K. (1998) confirm the absence of this in some casino games which led to the design decision to not include it in the simulation design. Furthermore, whilst being basic, it does account for different hand combinations and is, therefore 'composition-dependant' as stated by Thorp, E. O. (1966). Furthermore, splits were not considered to reduce the complexity of the simulation. The strategy employed is as stated below:

- If the sum<21 then a simulation is bust
- If the sum>=17 then the simulation must stay
- If sum <=16 and sum >=13 hit, unless dealer has a 7-value card face up
- If sum= 12 hit, unless dealer has 4,5 or 6
- If sum <12 then hit

This template will be used by all card counting methods to ensure that data recorded is influenced by only the betting method itself.

D.1.3. What it hopes to accomplish:

To replicate a card counting method performance against casino rules, I had to design accurate simulations for it to represent a 'real-life' scenario a player may face. This entails quantifying the values of casino tactics such as 'shuffle-rate', 'penetration', 'shoe-size', 'minimum bet'. Furthermore, other parameters such as 'number of players' and 'number of rounds' are considered to replicate a likely scenario a player may face. Each of these parameters was appropriately evaluated to accurately fit this scenario, thereby establishing the design of my simulation.

D.1.4. Number of players in play:

As stated by Vancura, O. and Fuchs, K. (1998), the players in a game of Blackjack can vary from 1-7. Therefore, to attempt to account for the various possibilities of players in the game, 2 values were selected to provide data on a performance given players at a table whilst maintaining a suitable scope for the project (reduced from the originally planned 3 values). The selected were 2 and 4 players. This change was also reflected in the updated 'MoScoW' analysis.

D.1.5. Number of rounds:

According to Andrew Gould, a supposed 10-year worker at the casino, the average casino aims for 50 rounds an hour and the average blackjack spends 2-3 hours playing Blackjack in a blog on Quora. Whilst the reliability of this source is questionable, it provided realistic numbers that led to my decision to choose 100 rounds for the simulations to play (simulating a player playing for 2 hours).

D.1.6. Starting chips:

For simplicity, this number was selected at 200, the focus was on the ratio of minimum betting value to starting chips.

D.2. Minimum betting value:

The minimum betting value (MBV) was set with consideration for starting chips and the number of rounds played. This is due to the ratio of these factors significantly affecting a methods chance of ruin, as well as its ability to perform. As the number of rounds was set to 100, it did not seem unrealistic for a player to have enough funds to pay the 1/200 of his budget for the minimum bet for that game. Therefore, a decision was made for the simulations to consider using 1 and 3 betting units (1/200 and 3/200 of their original budget) for the game. This will provide data on a method's performance given a simulation betting to budget ratio.

D.2.1. Shoe size:

Shoe size refers to the combined multiple shuffled decks that will be used for the Blackjack game. To capture the range of possibilities that Haigh, J. (2010) stated, which is 1-8 decks, 3 shoe sizes were chosen. In addition, these sizes were chosen with equal spacing across the spectrum to provide accurate data on the effect of increasing or decreasing shoe size on a card-counting method's performance. Leading to the decision of 2,4 and 6 decks in the shoe.

D.2.2. Reshuffle point:

The literature provided a lot of ambiguity when it to reshuffling the shoe due to the nature of it being incredibly dependant on the dealer. However, James, C. (2019) suggested a typical shuffle point is close to 33 per cent of cards played, which provided a figure to design my simulation. Therefore, it was elected to use 20 and 40 per cent reshuffle points as to not be too aggressive to undermine the other parameters due to the nature of this collaborating with penetration to increase the shuffle rate even more (cards removed before play meaning they cannot be used, and reshuffle comes earlier).

D.2.3. Penetration:

Penetration refers to the number of cards set aside to not be used in play with relation to the cards left. According to Vancura, O. and Fuchs, K. (1998), players must guess how many decks are in the shoe after penetration which can add a 10 per cent inaccuracy to the actual true count. Therefore, the simulations will account for slight inaccuracy. Richard N. Werthamer (2005) utilizes penetration to 80 per cent which influenced the decision to use both 90 per cent and 80 per cent to capture the effect of penetration on the card counting methods. To attempt to counter this accuracy, the methods will round up or down the number of decks remaining rather than calculate precisely.

D.2.4. Method selection:

As one of the aims of the project was to understand situations where a method may be favourable to another, it was important to have each method provide a unique approach to the factors that make it a card counting method (level, whether balanced or unbalanced and parameters). Therefore, 7 methods were included in the design, each one providing insight into the potential benefits of picking a certain method type.

As discussed in my literature review, the 'Level-1' methods consist of Hi-Lo and KO count which represents balanced and unbalanced respectively, as well as both, being a single parameter. The design choice to include the Optimum method provides my project with the ability to measure betting correlation as Haigh, J. (2010) determined it had a betting correlation of 1. Furthermore, this means an ideal betting scenario can be identified using this method and compared to other methods, to meet Vancura, O. and Fuchs, K. (1998) definition of betting correlation. The comparison will check a methods ability to determine a good betting situation vs optimal. From the 'Level-2' methods, Zen 2 was selected to represent an unbalanced and single-parameter method, whilst Hi-Opt 2 was selected to represent a balanced and multi-parameter method. The 'Level-3' selection involved the same process of attempting to pick methods that were diverse from each other whilst also being comparable to isolate the factors that make a method more favourable given a particular scenario. This involved selecting the balanced Uston Advanced Point Count method with two parameters and Wong Halves which is also balanced but one single parameter. The Values of these methods can be seen in Table 1 as described by Haigh, J. (2010), Vancura, O. and Fuchs, K., (1998) Thorp, E. O. (1966).

Card value	1	2	3	4	5	6	7	8	9	10
Halves	-1	0.5	1	1	1.5	1	0.5	0	-0.5	-1
Hi- Opt2	0	1	1	2	2	1	1	0	0	-2
Hi Lo	-1	1	1	1	1	1	0	0	0	-1
KO count	-1	1	1	1	1	1	1	0	0	-1
Optimal	-1.28	0.82	0.94	1.21	1.52	0.98	0.57	-0.06	-0.42	-1.07
Zen 2	-1	1	2	2	2	2	1	0	0	-2
Uston Advanced Point- -Count	0	1	2	2	3	2	2	1	-1	-3

Table 1- counting vectors

D.2.5. Betting:

The original design was to interfere with the betting spread of the methods as it posed a problem where it was unrealistic for casino's to allow a player to play conservatively for multi rounds then bet substantially when the odds are favourable. This was made evident by Vancura, O. and Fuchs, K. (1998), who explained KO-counts' 1-32 betting spread, meaning a max bet of 32 when odds are very favourable, whilst Hi-Lo has a betting spread 1-5 betting spread.
However, by interfering it would compromise the accuracy of the representation of the simulation of that method, therefore, it was re-designed to accommodate original betting strategies. The design compromise was evaluating them against their ability to cope with stricter Blackjack parameters as Casino's would likely force the player to bet a maximum or manipulate other factors to make it harder to accurately card counting, as the simulation would not reflect this possibility. Furthermore, different methods would have different aims, by manipulating the betting strategy, it is inevitably causing a bias in the data.

D.2.6. Calculating performance:

In terms of measuring the performance of a method, three key areas will be the deciding factors: Yield, bust rate and betting correlation. Further evaluations will be made on the deviation of the results, however, will not be included in its raw performance. Yield refers to the number of betting units gained or lost, and bust rate refers to the number of times a simulation is left with 0 divided by the number of times it played. Lastly, betting correlation will be measured using the Haigh, J. (2010) optimal method as it identifies the ideal betting scenarios, therefore other methods will be measured whether they have also identified that as an ideal betting scenario.

D.2.7. Requirement list:

To manage the requirements described in my design choices, a MoSCoW analysis was conducted, to not only create a hierarchy of what needs to be achieved in the project but direction into what iterations need to come first in my approach. Furthermore, this allows me to control the scope of my project when adding additional functionality/parameters by ensuring the most important aspects of the project are completed first. Moreover, providing the incentive to add additional features, however, prioritising the ideas that are fundamental to the project. Moreover, the requirements inevitable changed slightly after discussions with my advisor and further research which led to the final condensed version, seen below, which displays the ‘Must-Have’, ‘Should-Have’ and ’Could-Have’ labels to indicate importance.

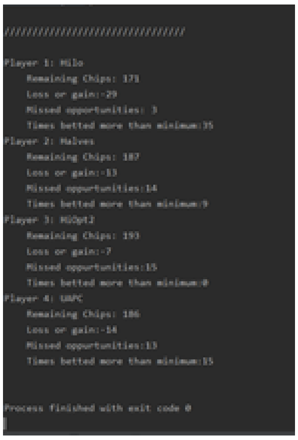


figure 1- updated MosCoW analysis

D.2.8. Managing my project:

Due to the dynamic nature when progressing through the project, it was necessary to track what has been completed, what needs to be done or reviewed, and issues. The solution used was a Trello board which is a collaboration tool to organise my project into sections, shown by figure 2, which consisted of a To-Do, To-Do code, review and completed list. This significantly aided my ability to meet deadlines as it promoted splitting tasks into smaller sections which supported my iterative approach. Furthermore, clear planning meant I could accomplish tasks more effectively and to a more successful degree due to the clarity, reducing periods where I may be unsure of what to do next or in what order. In addition, this was a tool used throughout the process to provide the direction needed to accomplish certain tasks, this is evident in the difference between the 'scrum' board at the early stages of the process compared to late respectively.

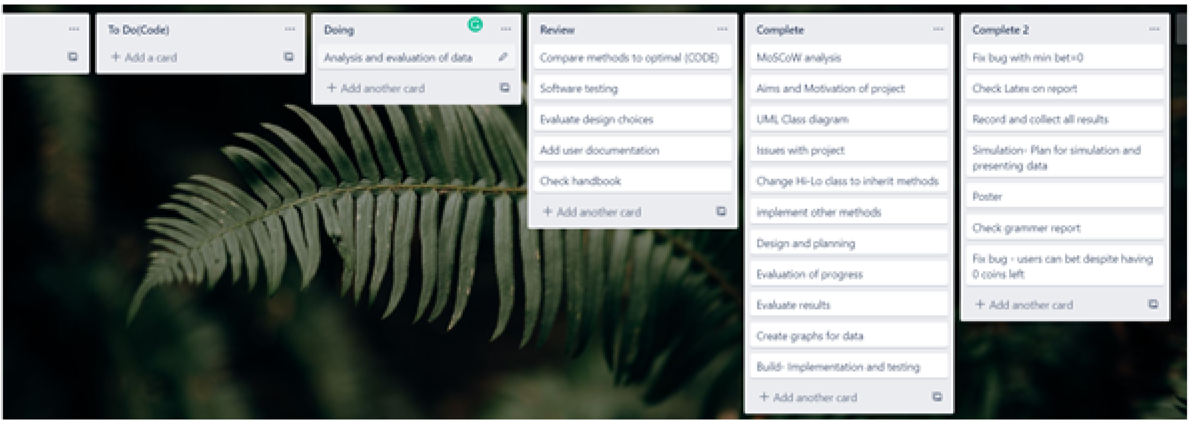


figure 2- new Trello

D.3. Design of the implementation

Following the design of the project, a visual representation of the requirements was needed in greater detail than the MoSCoW analysis. Therefore, a UML class diagram was created, shown by Appendix 2, which modelled the system and its requirements. This allowed me to gain an understanding of the architecture of my program in terms of the necessary classes and the respected methods needed. Furthermore, the design aided the breakdown of tasks, an example of this was implementing the Player interface on week 7 without having to deal with another aspect of the project. Thereby, splitting tasks into more manageable sections to produce a more comprehensive project.

Moreover, this worked well with Trello that was also used to manage the wider project. However, a separate section was created for the tasks relating to code that I needed to do, titled ‘To-Do (code)’, seen in figure 2. In tandem, is how I conducted my iterative approach.

In addition, to program the simulations I had to use the IntelliJ IDE as I opted to code in Java and is an environment that I am familiar with. In terms of collecting data, all the data was saved to a text file that contained the descriptions of what was happening during every game, but an addition excel spreadsheet was used to keep the numbers of every simulation. Lastly, to prevent the loss of my program or errors arising from changes I was unable to find, I implemented version control.

For the design choices of my UML diagram, it was clear there was 3 main aspects of the program: the player/methods (simulations), dealer and a class to handle the game. Therefore, it was clear a Blackjack game class was needed to simulate the game, the dealer class needed to act as a casino dealer, players to represent the methods and a deck class to manage the shoe. To improve the architecture, an interface was created to outline the behaviours the methods should have (players). In addition, a simulation class was created to easily enter input into the blackjack game class which made sense given the number of simulation variations I was conducting. The UML diagram was created by outlining the tasks each class had to complete after identifying the classes needed. Below is a rough plan made used to create the UML diagram.

Basic UML PLAN:
Each method had to:

- Choose its bet
- Implement basic strategy.
- Print its actions on console.
- Add its hand values together.
- Choose whether to hit or stay given first card.

- Calculate its chips remaining, after wins and losses.
- Update its running/ true count every round also reset after shuffle.
- Add card to first card.

Deck must:

- Create a deck, and create a shoe given input of deck size.
- Handle reshuffle and penetration'
- Allow dealing from deck.
- Print what is happening..

Dealer:

- Print what is happening
- Abide by dealer rules when choosing to hit or stay given cards.
- Show dealers first card.
- Add card to first card.

Blackjack Game must:

- Bring together, dealer, deck, and players
- Assign player indexes.
- Check cards
- Add card to first card.
- Print rounds.

Simulation must:

- Take input for number of players (and what methods), starting chips, min bet and deck number.
- Easily allow more parameters to be added.es.

D.4. Simulation Plan:

The order of the simulations followed a template, attached to the project, to ensure all combinations of parameters were met, to produce the results for every method. Due to the nature of having multiple parameters in the simulations, it was calculated that I had to simulated 288 (144x2 as ran with the same method alongside) different potential casino environments following my requirements. However, due to the desire to provide more accurate results, each method will be run alongside itself (being the second method in the simulation) which provides double with the data for each combination of parameters. In total, it equated to 28800 rounds of Blackjack. The full results would be held in an excel spreadsheet to allow for the easier analysis of that data.

Furthermore, the corresponding output of that simulation would be saved in a separate folder depending on its simulation batch and number. The simulation batch refers to its parameter variations as described in the template, to which there are 48 (not including methods). All methods are in every batch, and the variation can be confirmed by the template, results sheet or text file which is in the folder. The simulation number to the simulated game it was in, as multiple games were done to complete one batch.

D.5. Commentary on Issues:

Several issues arose during the project, the most impactful being the bug I found incredibly late (week 8 of semester 2) which meant I had to recollect all the data for the simulations again which was a long endeavour as I failed to realize that the deck number did not update with input. This set me back heavily which is reflected in the updated Gantt chart. At this point, I had already created some graphs and analysed them but became redundant as the data was inaccurate. The bug involved the true count not updating with the input entered across the simulations.

Furthermore, there was an issue in my design of the implementation as shown in my Appendix 2, I had previously counted the optimal method as a method that will run through the simulations alongside the other methods. However, this design would not allow for the necessary comparison to calculate the betting correlation (BC) as it is evaluating the cards of its own and not the cards of the other methods which is what it was intended for. Therefore, I had to remove this and instead, code this method into the others to ensure the comparison of the same situations.

Thirdly, for a significant portion of the project, I remained unsure of the effect of changing the betting spread which led to uncertainty when programming, despite already deciding a direction and receiving wise advice from my project supervisor. I was able to remedy this by doing further research into the reason for certain betting strategies.

Another issue that arose, was the proposal of making it a Monto Carlo simulation which did not make sense, so that idea was scrapped and ignored after submitting my project progress report.
There was another issue with a bug, however, luckily found during testing, which allowed methods to bet with less than the minimum bet limit. This meant some methods had the potential to enter negative chips and come back out to positive potentially unnoticed but was fixed with a few statements to ensure players have the funds to bet.

In addition, I attempted to account for the human element in calculating the True count. However, it is unclear whether forcing the method to round on the decimals is an accurate substitute for this inaccuracy without further testing.

In addition, the literature was ambiguous when referring how to a player would weigh the addition ace count in collaboration with the running or true count. The description of 'bet more' can be conducted in multiple ways which led to the design decision to have set value increases to the true count/running count further if a shoe is rich in aces.

Lastly, an issue I had was that when the deck is reshuffled, the player will use the true count from the previous round to decide on its betting strategy, whilst the first cards would make this accurate, the shuffle was in between rounds which meant the second card everyone receives is from the shuffled shoe which throws off the betting for that round. However, was resolved before data collection.

E. Analysis and results

After the completion of the program came the data gathering of all the combination of the parameters described in the 'MoSCoW' analysis. This involved entering the input into the Simulation class, which would then pass it to BlackjackGame class that performs the game on the methods and parameters entered. However, the shuffle rate and penetration were handled by the Deck class which had to be altered in line, which decreased the convenience of the program

Once the desired parameters were entered alongside the participating methods (players), the program would run the 100 rounds and print the important corresponding information needed to understand the results such as the cards being dealt, betting numbers, counts, remaining chips, when a shoe is reshuffled, round calculations and dealer's position. A typical view of a simulated round can be seen by Appendix 3. Furthermore, a separate print displayed the overall performance of a method, shown by Appendix 4, which displayed the simulations remaining chips, its net loss or gain, missed opportunities to bet (used to calculate betting correlation) and times it betted more than the minimum (evaluate how conservative a method is). These steps were done for every variation of the proposed parameters in the design.

To improve the accuracy of the result analysis, each variation ran with its method in the same game. However, the missed opportunities would also indicate whether a game was not in the favour of the simulation.
Whilst the betting values of each method can be seen in Table 1, the betting strategies varied incredibly between balanced and unbalanced methods. Vancura, O. and Fuchs, K. (1998) suggest unbalanced methods should bet more 'aggressively' when the higher the running count is, whilst Thorp, E. (1961) states how balanced methods should bet according to its true count and minimum bet. The running count (RC) calculation does not account for shoe size besides the reduced running count at the start of a game and during reshuffles. This meant as it gets greater, it is much more likely to encourage larger bets as there is no deck size divisor. Furthermore, the betting spread boundaries

are significantly different, Vancura, O. and Fuchs, K. (1998) suggests a 1-32 betting spread for the KO count method whilst James, C. (2019) advocates for a 1-5 betting spread in terms of betting units.

E.1. How I analysed my results?

After collecting the raw data seen in the attached Results sheet, the data was sorted to isolate key themes/interests in terms of what I wanted to measure. These interests include:

- the overall performance of the methods
- effect of varying shoe size/players in game/penetrating levels/reshuffle points/minimum betting values
- Analysing multi-level parameter methods versus single
- Influence of method 'level' on results

E.2. Overall performance of methods:

Each method has 96 recorded performance for the 46 variations of the other factors, to analyse the data, it was first sorted by the method name. To which the corresponding averages for each method's betting correlation and yield were worked out for those 96 simulated games.

Figure 3 shows the average betting correlation across all simulations, with a betting correlation of 0.9, Hi-Lo has the highest accuracy across the multiple parameters a casino uses to discourage card counting. This means it was able to identify 90 per cent of the ideal betting scenarios outlined by the optimal method that resides within each method. However, Figure 4, shows that this does not translate to higher returns on investment as it ranked 4th behind Halves, Ko count and Hi-Opt2 respectively for average yield. This indicates that despite accurately judging good betting scenarios, it may bet more conservatively than the other named methods in those situations. Furthermore, the 52 spread for the standard deviation in its yield indicates it is more affected by changes to the parameter than other methods with similar yield results.

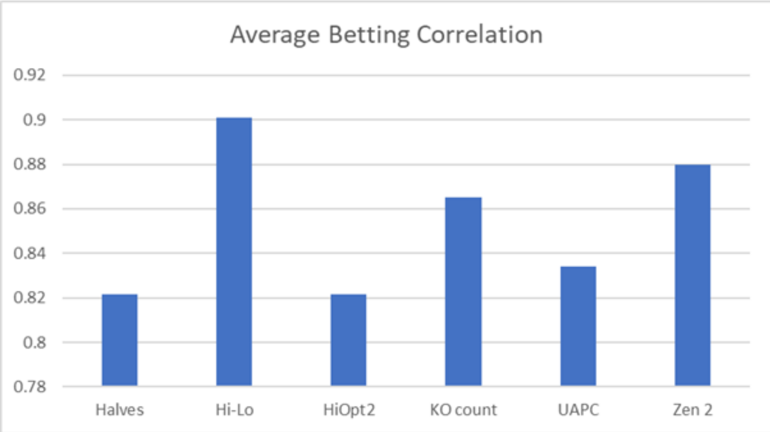


Figure 3- Average betting Correlation

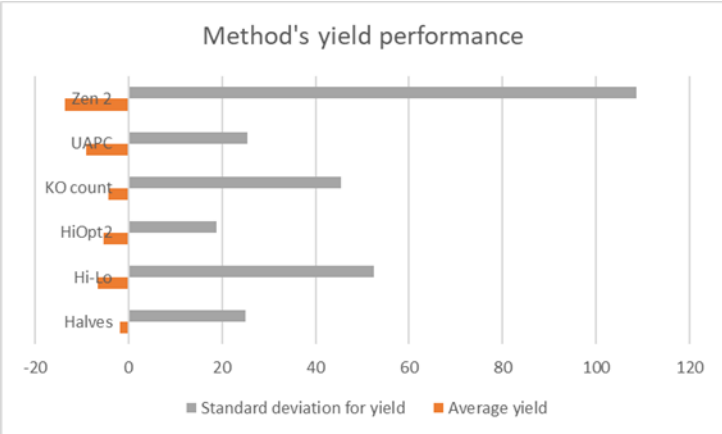


Figure 4-Average betting Correlation

Despite having the joint worst Average Betting Correlation (AVG) with HiOpt2, Halves shows the best performance in terms of its return to the player at -1.9. This demonstrates that it has the best ability to cope with the constantly changing parameters during a casino Blackjack game whilst having the lowest deviation in performance across those scenarios. The data indicates that Halves manages to take advantage of betting scenarios, to a greater degree than the other methods, when it does identify them, so be it at a lesser rate in comparison. This is supported by the fact that it maintains the highest return of betting units whilst providing similar results in those different scenarios.

On the other hand, the worst performance was Zen 2. When a player attempts to count cards, they select the method based on its complexity, yield performance and consistency. Zen 2 falls short on every aspect, despite having a high betting correlation of 0.88, its performance is incredibly vulnerable to changes in parameters meaning inconsistent results as well as the lowest average yield return at -13.6. Considering the betting correlation, it is more than likely that the betting strategy this method utilizes caused the results as KO-count, another unbalanced single parameter, shares a similar betting correlation whilst significantly outperforming Zen 2 in terms of its resilience to different scenarios and average yield return. This point is further supported by the fact that Zen 2 on average only bets 5 times more above the minimum bet more than KO-count yet performed much worse. Furthermore, Zen 2 was the only method to have the simulation have no remaining chips, which was done on two occasions.

By analysing the number of times a method bets above the average, it was made apparent that the reason for the lack of deviation for HiOpt2 was due to incredibly conservative betting. By using this betting strategy, it essentially works as playing basic strategy, the method bets in extremely favourable scenarios but because of the parameters in place they are far and few between which mean the method often bets the minimum shown by the 7 average bets above minimum bets per 100 rounds, which is significant compared to KO's 23. Therefore, the results will be consistent as no real risk are taken to capitalise on favourable scenarios to fluctuate a players finance. Despite this playstyle, it managed to perform better than Uston Advanced Point Count (UAPC) in terms of its yield, which indicates that across the parameters in my simulation, it may be more favourable to play using basic strategy then adopt UAPC as a strategy as a player would lose less.

KO count performed the second-best behind Halves in terms of its return to a player, however, is only a 'level 1' method, an unbalanced method, with a single parameter, which makes it the least complex method that achieves the best in comparison to the other methods. This method has a respectable BC of 0.86 whilst avoiding the need to convert to true count. However, this method can produce diverse results which a player must be aware of, as well as account for when calculating how many chips to bet concerning their starting chips as the rounds consist of more dispersion. Nonetheless, remains the most realistic decision according to my data in terms of the trade-off for performance and complexity.

Moreover, the data discourages the idea of utilizing additional parameters in a method, with both of those methods finishing 3rd and 4th, despite the additional complexity, in terms of yield. Furthermore, provides shows no trends that indicate whether a particular 'level' or method type is superior in the conditions of the simulation.

The simulations suggest that despite the diversity in card counting methods, a casino is still able to maintain its edge using various combinations of the stated parameters. Not a single method could manage the parameters to the extent where a player could realistically return a profit over multiple sessions.

E.3. Number of players:

Similar to the process done above, the average yield for both variations in the simulations was calculated. It was found that in the simulations with only 2 players, the average yield was -12.5, 12x less than the average yield with 4 players. This is due to the effect the number of players has on the running/true count, the more players, than the more cards distributed per round which leads to a more informed betting decision for the next round. This is means reshuffles have a far greater effect on a Blackjack game with less players.

E.4. Shoe size:

Seen in figure 5 , it is clear the impact increasing the deck number has on methods ability to perform. The increase in shoe size significantly increases the 'houses' edge on a player, although the reason why for the significant difference from two decks to four in the shoe is unclear. Despite this, a decreasing average in the methods yield is expected as the shoe size increases.

Shoe size	AVG net	AVG BC
2	3.078125	0.804375
4	-16.6458	0.870625
6	-6.875	0.886667

E.5. Minimum betting value:

As the methods utilize the minimum bet to base their strategies, the two are intertwined. The data suggests that using a larger ratio of minimum bet size to your starting chips will decrease your yield. This is the case as the 3 minimum bet loses 3 more betting units on average per simulation more than the 1 minimum. However, only two variations minimum bets were implemented, and at least one more variation is needed to make a more appropriate conclusion about its influence.

E.6. Penetration:

The data heavily suggests the penetration plays a huge role in limiting a method's ability to identify good betting opportunities. By increasing penetration from 10 to 20 per cent (from 0.9 to 0.8), the average yield dropped 8.5 betting units. Thereby, by removing an additional 10 per cent from the shoe, reshuffles are forced to come earlier which resets the counts and also make it difficult to gauge the effect of card removal when there a player doesn't know what cards were removed.

E.7. Shuffle rate:

For the shuffle rate, the data analysed for the two shuffle rate variables suggest that a higher shuffle rate will promote higher yields. However, this data is likely skewed by the results of the other parameters as it does not make logical sense.

F. Testing

F.1. Unit testing:

Throughout the implementation of the application, unit testing was utilized when a class was added. This was done to ensure the class works as designed. This approach also fits in with my iterative development cycle. An example of this was done after creating the Deck class, to which a deck was initialised, and it was tested to see if it could remove cards, shuffle cards, and penetrate a deck whilst proving it works.

F.2. Requirement testing:

Whilst conducting my investigation into card counting methods, the priority of each requirement changed, especially so after key meetings. Therefore, it was necessary to keep up to date with the requirements proposed from the original MoSCoW analysis. The solution was to utilize a traffic light system, to which the requirements that have been completed are marked green, those under review marked yellow and those not completed marked red, shown in Figure 5.

MoScoW analysis:

Should-Have:

- 6 Unique card counting methods, with diverse types, levels, and parameters.
- Accurate simulation of each of those methods using Basic Strategy, with Blackjack rules at Casinos.
- Manipulation simulation games using:
  - 2 different shoe penetration points,
  - 3 different shoe sizes,
  - 2 different number of players in play,
  - 2 different bet size,
  - 3 different shoe sizes,
  - 2 reshuffle points.
- 100 simulations of each combination of parameters.
- Betting according to true and running count.
- Easy display of results at the end of the simulation in terms of its net gain or loss and betting correlation

Should-Have:

- Easily scalable to add another parameter to test the methods.
- Display the result of the rounds of the simulations.

Could-Have:

- Manipulation of betting strategies
- Manipulate number of starting chips as well
- New strategy created from results.
- Player interface

Wont-Have:

- Graphical user interface to interact with simulations and present results
- Database to record results

Figure 5

F.3. Visual inspection:

A basic player was created alongside the 6 methods as a comparison of a no-method approach, which allowed me to conceptualise what the results should look like depending on that base and the effect of different betting strategies. This allowed me to find a few bugs such as a faulty for loop that forced a few methods to only bet 5. Furthermore, each stage of the Blackjack process was printed to visually present what is happening to identify possible mistakes. Through this process, I was able to identify and remedy the bug shown by, which meant true count was not updated the round after a shuffle.

G. Discussion, evaluation, and conclusion

G.1. lessons learnt:

This project has taught me how to handle large amounts of data and organise it to provide the necessary analysis and raw data. Furthermore, it allowed me to gain experience following planning and implementation to ultimately reach an end goal.

G.2. Evaluation:

Through testing, it was found that the bet method some of the methods were being called twice. Whilst it did not affect the betting amounts, there was a separate counter on those methods to show how many times a method has bet that would have been affected. This bug was attempted to be fixed and passed testing; however, it has affected my confidence in using it as a tool to analyse. Therefore, I should have designed it better so it is not called twice as I could not understand why it was.

Despite the extreme care taken to ensure all parameter are correct with the simulation ran for it, as penetration and shuffle rate were handled by the deck class, it allows an opportunity for error. Ideally, this would be solved by passing it through the Simulation class with other parameters, as it significantly reduces the risk of error. However, due to other bugs, I did not have the time to make this change.

It may have been more beneficial to reduce the number of parameters, to run more simulations for fewer variations to increase the accuracy of the representation of the simulations. This is due to some unexpected results when it came to multi-parameter methods and higher-level methods. By using this suggestion, it would further prove/disprove the analysis made on the data.

Furthermore, it was assumed in the design that, whilst I am utilizing casino rules for normal games, that variations would not change between shuffles. In a casino environment, even if a dealer tried, it is incredibly unlikely to have the exact parameters from one re-shuffle to the next. This does not include casinos flagging suspicious behaviour as would be demonstrated by KO-count where it would bet very high relative to other methods when the odds were highly favourable but normal otherwise. In this case, for my project to suggest it as a method for players to use in casino's then further research would have to be done on how dealers suspect card counting and their appropriate response.

The design also contains methods that are unrealistic for a player to use in casino's, such as UAPC, despite it being included for data on method performance, which reduces the potential of my study as a less complex method could have taken its place to potentially be better suited to the purpose of the study.

This study aimed to replicate key themes involved with a method performing at a casino. However, human inaccuracies cannot be measured accurately. Even the shuffle of the cards will not be entirely accurate to that of a casino (simulation cannot copy different dealer's shuffle technique). Every dealer will have different muscle memory of where to penetrate the deck, different points they believe they should reshuffle, and many other factors. Although, simulating it indicates how a method may do.

Despite the simulation of over 28800 rounds, due to the number of variations, each method only has 2 simulations on each variation which is not enough data to establish the best method for that variation. Furthermore, most of the literature mentioned in the report suggested their betting strategies. However, as the betting values are to measure the effect of card removal, it should be possible to vary the betting strategy, to get the highest possible yield.

The intention of including unrealistic methods for a human to use was to provide insight into the more advanced and theoretically more accurate ways of accounting for card removal. However, they generally performed less which suggests that either my data is incorrect, due to a bug or incorrect portrayal of the method, or the more basic models are more accurate.

To account for the inaccuracy of guessing deck size remaining, the number was rounded to add a slight inaccuracy in the count. However, this would have affected only balanced methods and the degree of inaccuracy is unclear compared to the 10 per cent described by Vancura, O. and Fuchs, K. (1998). A more effective solution could have been to add a random number generator to assign an inaccuracy from 0 to 10 per cent The collection of data involved input from myself which inherently comes with a risk of not entering the parameter variation correctly, leading to an inaccurate dataset. Though care was taken to ensure to actively attempt to decrease this chance.

On the other hand, the application was successful in producing data that was used to compare performance markers to demonstrate the strengths and weaknesses of various methods. The application also is very easy to add extra parameters and variations.

From the methods taken from Haigh, J. (2010), comparing the betting correlations from SEEN BELOW, all my versions were less than the betting correlations shown in Appendix 1. However, mine includes the additional parameters such as penetration and reshuffle rate to which that study does not account for in those BC's.

Method	Average Betting Correlation	Average yield	Standard deviation for yield
Halves	0.821354167	-1.916666667	24.94480017
Hi-Lo	0.9009375	-6.59375	52.48146064
HiOpt2	0.821666667	-5.302083333	18.8287899
KO count	0.865104167	-4.385416667	45.36366245
UAPC	0.834270833	-9.041666667	25.36849944
Zen 2	0.88	-13.64583333	108.7125394

Figure 5

H. Conclusion

The project managed to meet its aims and objectives which was a successful implementation of card counting methods in a simulated casino Blackjack environment. However, I believe my selection of the parameters were too strict which led to the methods unable to successfully take the edge from the house. Despite this, I was able to provide a ranking of the resilient methods across multiple potential scenarios a player may face at a casino which was one of the key drivers of this project. Moreover, an application was created that can easily accommodate additional features and parameters. In additional, the modular application has the required documentation and notes for it to be replicated.

References

Haigh, J. (2010). Risk and reward: The science of casino blackjack, by n. richard werthamer. pp. 131.£ 44-99 (hbk). 2009. isbn: 978-1-4419-0252-8 (springer). The Mathematical Gazette, 94(531):569–570.

Thorp, E. O. (1966). Beat the Dealer: a winning strategy for the game of twenty one, volume 310. Vintage.

Zimran, A., Klis, A., Fuster, A., and Rivelli, C. (2009). The game of blackjack and analysis of counting cards.

Vancura, O. and Fuchs, K. (1998). Knock-Out Blackjack: The Easiest CardCounting System Ever Devised. Huntington Press Inc.

Tulcea, C. I. (1983). A Book on Casino Blackjack. Pocket. James, C. (2019). How to count cards.
https://www.blackjackapprenticeship.com/how-to-countcards/: :text=TocalculateourTrueCount,wouldbear

Thorp, E. (1961). A favorable strategy for twenty-one. Proceedings of the National Academy of Sciences of the United States of America. Retrieved (2009, December 1) from http://www.jstor.org/stable/70615.

Wen-Jen Hsu Bin-Tzong Chie (2021) How do advantage players exploit casinos' premium offers? Simulations and solutions, International Gambling Studies, 21:1, 74-91, DOI: 10.1080/14459795.2020.1801794

I. Appendix

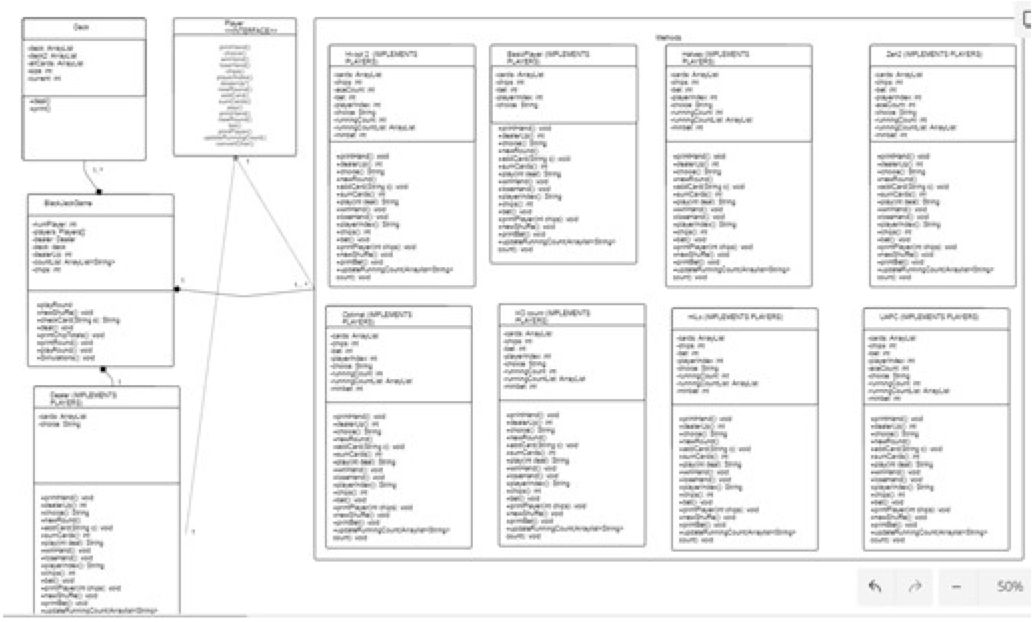
I.1. Basic Strategy

Table 2.1 Generic Strategy (Optimal Basic Strategy for three to six decks)

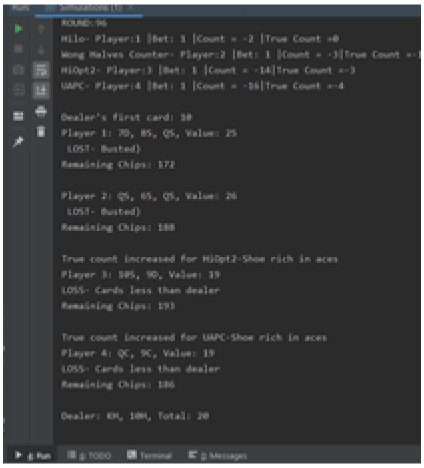
Action	Hand value	When Dealer shows	
Double	11	2 through 10	
	10	2 through 9	
	9, soft 17 and 18	3 through 6	
	Soft 15 and 16	4 through 6	
	Soft 13 and 14	5 or 6	
Split		DAS allowed	DAS not allowed
	Aces and 8 + 8	Any	Any
	2 + 2 and 3 + 3	2 through 7	4 through 7
	4 + 4	5 or 6	–
	6 + 6	2 through 6	3 through 6
	7 + 7	2 through 7	2 through 7
	9 + 9	2 through 9, but not 7	2 through 9, but not 7
Stand on	13 and higher	2 or 3	
	12 and higher	4 through 6	
	17 and higehr should be higher	Ace, or 7 through 10	
	Soft 18 and higher	2,7, or 8	
	Soft 18, 3 or more cards	3 through 6	
	Soft 19 and higher	9,10, or ace	
Surrender, when available	15	10	
	16, but split 8 + 8	9, 10, or ace	

The final element of Generic Strategy is to always refuse the insurance bet

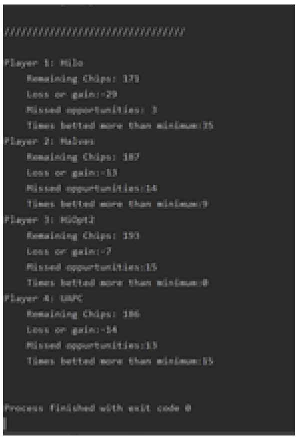
I.2. UML



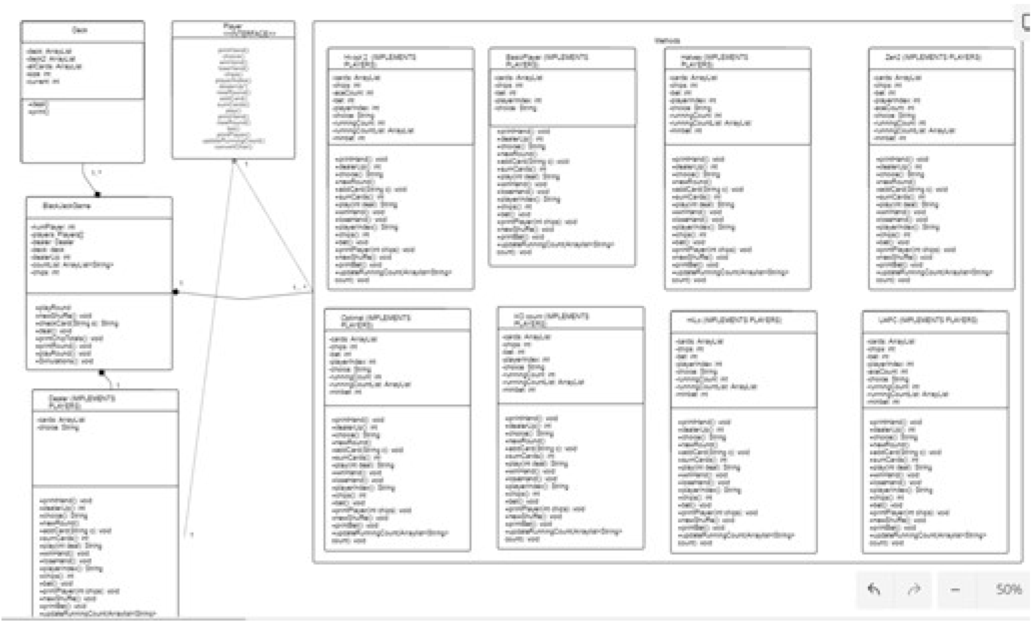
I.3. Simulated round



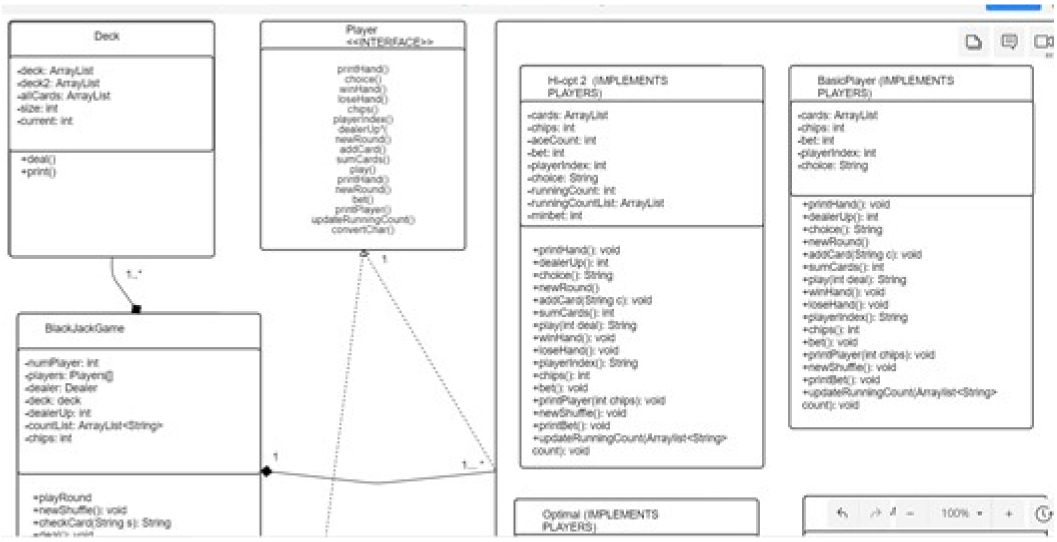
I.4. Simulation game



I.5. UML Diagram



I.6. UML Diagram zoomed



I.7. Updated Gantt Chart

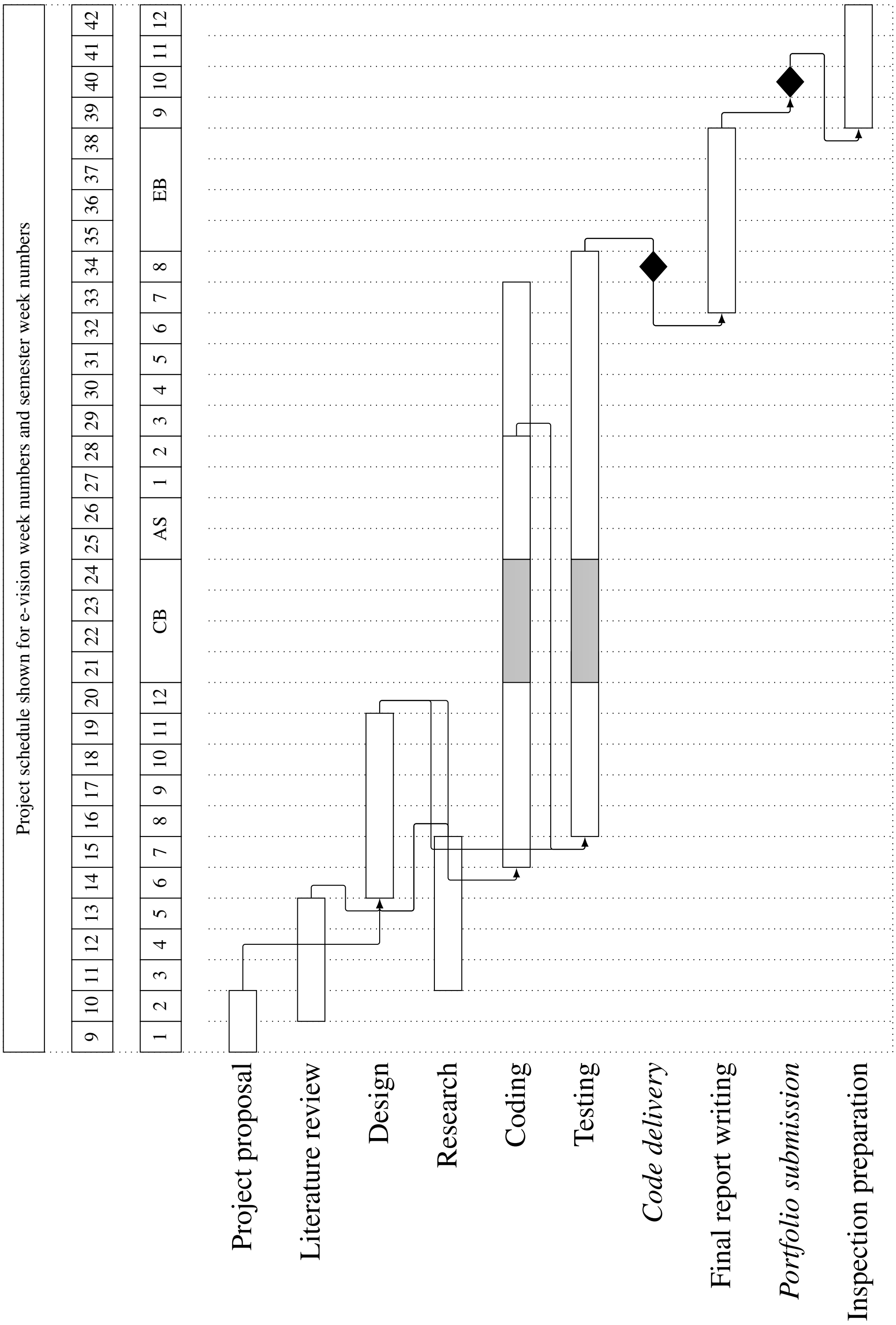


Figure 1: Project Gantt chart