Ambar Vishnoi

TExas Hold’Em Poker



Contents

[1.0 Analysis of the problem 2](#_Toc76114212)

[1.0.1 Summary of the project 2](#_Toc76114213)

[1.1 Problem Identification 2](#_Toc76114214)

[1.1.1 Features that make the problem solvable by computational methods 2](#_Toc76114215)

[1.1.2 Why the problem is amenable to a computational approach 3](#_Toc76114216)

[1.2 Stakeholders 3](#_Toc76114217)

[1.2.1 Identifying the end user 3](#_Toc76114218)

[1.2.2 Why the solution is appropriate for the end user 3](#_Toc76114219)

[1.3 Research 4](#_Toc76114220)

[1.3.1 Similar problems and solutions 4](#_Toc76114221)

[1.3.2 Essential features of the proposed solution 9](#_Toc76114222)

[1.3.3 Limitations of the proposed solution 10](#_Toc76114223)

[1.4 Specifying the proposed solution 10](#_Toc76114224)

[1.4.1 Solution’s requirements 10](#_Toc76114225)

[1.4.2 Success criteria 11](#_Toc76114226)

# Analysis of the problem

### Summary of the project

Poker (Texas Hold’em Poker) is one of the largest card games in the world. Many casinos feature multiple tables and there are regular large tournaments played with professional players and large prize pools. The game is famous for several reasons: it is one of the few games where players will play and bet against each other, rather than playing against the casino (such as Blackjack). It is also famous because the game is not purely based on luck, nor is it solely based on skill, instead, in order to be good at the game, one must have a good ability to turn their luck (or lack of it) into results.

The game consists of 5 cards which all players ‘share’ (known as ‘community cards’), with each player having 2 cards to themselves (known as ‘hole cards’). The players will then bet against each other as the community cards get revealed, to see who has the best hand, which should be a total of 5 cards with at least one card from your hole cards.

However, I always have thought that learning the game proves extremely difficult, and what is even harder, is getting good at it.

My aim is to create a Texas Hold’em Poker game that will allow inexperienced players to develop their skills and learn the game, while also allowing experienced players to have a challenging opponent. I will do this by creating AI players at various difficulty levels so that stakeholders can choose the experience they desire. I will also be implementing a multiplayer feature that will allow stakeholders to play on multiple machines against each other. Furthermore, a key part of the program will be the Poker Tutor – a feature which can be turned on or off and will guide novice players into making the right decisions by displaying odds and giving tips. Lastly, a tutorial feature will teach the basics of the game to any new player.

## 1.1 Problem Identification

### 1.1.1 Features that make the problem solvable by computational methods

One key feature of the program is the Poker Tutor, which must calculate the probabilities of winning the hand at each turn and display them in a user-friendly way. The Tutor must also use these probabilities to display suggested moves and tips that match any scenario.

In order to achieve this, the program must use all the available data from the current hand and carry out multiple complex calculations. These algorithms, therefore, require computational power since these methods would be time consuming and labour intensive.

Furthermore, the fact that players are able to play on different machines, means that a computational method is required - there is a need to communicate between remote devices. Players are able to play in the same game yet not be in the same place geographically which would be impossible to achieve without using a computational method.

Additionally, tasks in my solution can be decomposed and abstracted. For instance, when looking at the Poker Tutor. This task can be broken into further subtasks: calculating probabilities, suggesting moves, giving appropriate tips; which can then be broken further. Furthermore, the subtasks that have been decomposed can be abstracted to ensure that the subtasks will work in any scenario not just a specific one. Both of these procedures can be applied to any task, and so the problem requires computational thinking and is solvable by computational methods.

### 1.1.2 Why the problem is amenable to a computational approach

The game will require sequential processing, for instance, at the start of each hand, cards will be dealt, blinds will be placed, and then cards will be turned over in the centre. Iterative processing will also be required, during each betting stage players take turn deciding what to bet, the process repeats itself until everyone has the same bet. Moreover, during the card dealing, cards will be given to each player repetitively. Because of the nature of these processes (require iteration and sequential processing), they lend themselves to a computational approach.

When dealing, cards will be animated from the deck into each players’ hands. Likewise, player bets and folding will also need to be correctly displayed and animated. Furthermore, hole cards; chips; pot size and the community cards will all need to be stored throughout the game, appropriately. A need for animation, graphics and data storage makes the problem amenable to a computational approach, since these would not be achievable without using a computer.

## 1.2 Stakeholders

### 1.2.1 Identifying the end user

The end-user can be any person who has an interest in poker, wants to learn poker, or who is already experienced at poker. The end user can also be an experienced gamer or a casual gamer. End users who are new to poker should find that the solution is not too challenging, which might turn them away from the solution. Whereas users who have more knowledge on poker should find that the solution is not too basic, to ensure that the user does not get disinterested.

All end-users require the solution to be enjoyable, and the game should be relevant to their skill level. The solution should be simple to follow and efficient so that playing does not feel tedious, boring, or repetitive.

### 1.2.2 Why the solution is appropriate for the end user

The solution is able to satisfy end users of all levels of expertise in poker since there are various features that the end user can alter, so that the solution can be suited to them. For example, someone who is new to the game can learn the rules via the tutorial feature, whereas someone who is experienced is able to play against ‘expert’ level AI. There are also other options to make the solution easier or harder, such as choosing the amount of chips all players start with, or the blind sizes.

The end-user is able to play and compete against virtual AI which gives them the appropriate usage out of the solution since not all stakeholders will have other people to actively play against. Similarly, the solution provides end users with the ability to play poker even without the physical equipment required to play, such as the cards or betting chips. This appeals to the casual gamers, who might not want to invest in the physical equipment or want to spend time completing the lucrative task of distributing chips and shuffling cards, but instead can play poker in an instant, using my solution.

In addition, end-users can play against other end-users who are on different machines. This is important to the end users since stakeholders do not have to be physically together in order to play, instead can play at any place with other stakeholders in separate locations.

The user interface will also be accessible and clear so that end users are able to get the most pleasurable and satisfying experience, which is a required specification of the end user. Additionally, stakeholders will be able to play the game with just a few clicks, making the solution efficient, this ensures that the end-user does not find the solution exhausting and laborious to play.

## 1.3 Research

### 1.3.1 Similar problems and solutions

#### 1.3.1.1 Solution 1 – Zynga Poker



Zynga Poker is the biggest free online poker games. Each account has their own set of virtual chips that they can use to enter tournaments, win jackpots, and challenge other players. The game boasts the fact that they offer both casual Texas Hold’em as well as competitive tournaments.

In the figure above, a player has won with a flush, and the display has changed to show the player what hand has won. In addition the outline of the players profile has changed to indicate they have won. I would like to implement these features into my solution as this makes the solution more accessible to novice players since the display is clear and concise. Each player has their own profile picture, to make each player distinct from each other. This is a feature I would not like to implement since this is outside the scope of my solution, and it is much easier to give each player discrete names, and individual colours.

The game offers a VIP program, a feature which I will not be implementing since it is not within the scope of the solution. The VIP program offers in-game benefits such as ‘exclusive chip package offerings and special poker game modes’. I will not be implementing this feature into my solution since I will only be creating one poker game mode – Texas Hold’em - and players will not have their own accounts with their own chip count. Instead, each game will be treated separately, and any winnings or losses will be kept local to each game. Also, ‘in-game benefits’ is a feature I do not want in my solution since I want to keep the solution fair for all stakeholders and not a ‘pay-to-win’ system.

Furthermore, a key feature of the game is variety – you can play a faster 5 player table or a slower but larger 9 player table. The user can also play competitive tournament style poker with ‘high stakes’ and can compete against millions of other players in large scale leagues. Or the user can opt to have a more casual ‘classic experience’. Zynga Poker also promises ‘games for all experiences and skill levels’.

An experience with a large variety will be a key feature of my solution – allowing stakeholders to fully customise their experience – for instance choosing how many players they would like to play against, the level of AI, the amount of starting chips, the blind sizes, or the ability to play against real players. However, I will not be implementing tournament style games or large-scale leagues as this is outside the scope of my solution.

Lastly, they offer fair play – the experience should feel like you are getting ‘the true Vegas-style game’. This is also a feature I would like to have in my solution as I feel that the user should have an experience as close to possible as a real table experience – so all probabilities must be the same as in real life.



The actual table is shown in the figure above. On each players turn; a timer appears above the players’ profile. If the player does not make a move within this timer, they are forced to fold or check. This is a feature I would like to implement in my game to ensure the game goes smoothly and quickly. Moreover it makes it evident to the end-user how much time they have to make a move, and does not keep other players waiting too long, which guarantees that the end user does not lose their interest.

In addition, there are icons next to each player, signalling if the player has checked, called, raised, or folded, along with their bet amount. These icons are an effective way to display a lot of information in a minimalist and straightforward way. Therefore, I will implement these icons that will have the same purpose, in order to satisfy the stakeholders need for the solution to be as simple as possible. Novice players may find these icons difficult, however, a tutorial feature will explain these icons and their meanings, to certify that the solution is appropriate for all end-users.

#### 1.3.1.2 Solution 2 – WSOP Poker



In the figure above, WSOP Poker’s main menu is shown. Although, I will not be including the assorted styles of tables as shown above, the overall aesthetic and layout of the menu is something I will be implementing into my main menu. This is because the menu is straightforward and obvious, so any new gamer will find it trivial navigating around the menus. The menu includes large buttons such as ‘PLAY NOW!’ and the buttons are inside larger blocks which include imagery of the type of table the user will be playing. My menu will also feature blocks of imagery with large buttons for the various different actions. WSOP Poker’s menu features a ‘Rules’ section whereas my menu will feature a tutorial section that will include all the basic rules of the game since I would like the solution to be accessible to players of all skill levels.



WSOP Poker’s table is shown in the figure above. The table shares a lot of features with Zynga Poker’s table: the timers, the placement of hole cards, community cards, dealer, and bet icons. However, there are some distinctive features such as the ‘Fold’ and ‘All-In’ buttons which are located to the left and right if the user’s hole cards. These buttons are large and easy for the user to click and the large space between them creates a clear distinction between the two actions. For these reasons, the buttons in my solution will also have a wide separation and have a generous size, however, my solution should include all actions, for example, check, call or raise.

Another feature of this table is the gift feature – where users can gift any other player a certain amount of chips. This is a feature that I will not be implementing into my solution, since this is not a requirement for the end user and I this is a pointless feature that does not add any realism – the end user should have the most realistic experience while playing.

WSOP Poker has a player ranking system, a competitive tournament style mode where players can ‘rise through the ranks’. In the figure above, players have an icon with a single letter next to their chip count – this is the player rank, a measure of how good a players’ poker skills are. Player ranks are outside the scope of my solution, so I will not be including them. Users will not have their own account and since each game is local (everything that happens within the game, stays within the game) ranks are a feature that would not fit with the overall solution.

In addition, when a player folds their profile picture becomes opaquer to signal that that player is no longer playing this hand. This feature is useful and explicitly shows who is currently playing the current hand. Accordingly, I will be implementing this feature into my solution as this fulfils the stakeholders' requirements of having the game be easy to follow.

#### 1.3.1.3 Solution 3 – Poker heat



Poker Heat’s table is shown above and shares lots of similarities with the aforementioned solutions. Poker Heat also features a chat system which can be activated using the button in the top right corner. Although this does add some realism (allows users to speak with one another), it is not a necessity to the end user and therefore I will not be implementing it. It is likely that friends who will play my solution will be in the same geographical area and so do not need a chat system.

Poker Heat features buttons in the bottom right for the various actions even when it is not your turn. These are ‘pre-select moves’ – players can select a move before their turn, and when it eventually comes to their turn, the action is done instantaneously. This is a feature I would like to have in the overall solution, as it speeds up the tempo of the game and reduces the amount of work the user must do in order to play the game. This gives the user an overall better experience since they are not waiting around for their turn to come.

Poker Heat has a bar next to the user’s profile, which provides a quick way for the user to see the likelihood of winning. Located next to this bar, is the user’s best possible hand. These features aid unpractised players in making better decisions, and therefore will be a part of the Poker Tutor. This means experienced player can play without these assists, and novice players can choose to have them on in order to have a more pleasurable experience. The bar is a much simpler way of providing the complex probabilities to a new player, however the tutor should have an option to enable the user to look at the probabilities in detail.

As shown in the figure below, Poker Heat offers a collection of ‘rings’ which are rewards for playing the game. In order to earn these rings, the user must complete various missions and tasks given to them each day. Players can then choose to wear these rings on the table to show other players their achievements and poker skill. I have chosen not to implement a rewards system and daily missions since this is outside the scope of my solution.





Lastly, Poker Heat offers daily rewards and free chips that the player can use in their poker games by means of a daily free spin shown in the figure above. In my solution, the player does not have a chip count and each game they are given a fresh set of chips, therefore, the free chip system does not fit with my overall solution and so I have decided not to implement it. Furthermore, the chips you receive are scaled based on your ‘league’ and since I have decided also not to implement a ranking system, a daily rewards system does not suit my solution and is not a requirement for the stakeholders.

### 1.3.2 Essential features of the proposed solution

A key feature of the solution is the ability to customize the game experience: being able to activate the poker tutor, change AI difficulty, change number of players, starting chip amount and blind sizes. This variety of choices allows the end user to tailor the experience to each individual requirement that they might have – for example make the game more challenging or make the game shorter.

The solution should have an aesthetically pleasing menu that has all the appropriate actions such as create new game, settings, quit and tutorial. Each action should fulfil its purpose and be functional. The menu is the first thing the end user will see and so a good-looking menu will make the overall experience for the stakeholders better. A functional menu ensures the end user does not get frustrated trying to navigate the menu and fulfils the stakeholders’ requirement for my solution to be accessible and a satisfying experience.

The Poker Tutor should display probabilities in a novice friendly way, should suggest appropriate moves based on several variables: player chip amount, other players’ chip amounts, probability of winning, probabilities of getting each hand, and other players’ actions. The Poker Tutor should also give various tips throughout the game. The Poker Tutor is essential to providing a new gamer or a novice poker player a more pleasant encounter with my solution. Inexperienced players are able to learn Poker while also having the opportunity to win, which they might not be able to do, whilst learning poker against skilled opponents and no assists or tips.

Another essential feature of my program is the AI. Stakeholders require my solution to be played without other real players and AI are a perfect solution to this problem. Stakeholders can also customize the difficulty of the AI to suit their own skill level, which ensures that the solution provides the end user with a challenge whilst not being impossible to win.

Each player should have a time limit per turn to ensure that the game goes quickly and there is not a lot of downtime between turns. A timer should be displayed for each player in an obvious way so that the end user knows exactly how much time they have to make a move. Furthermore, being able to pre-select moves a key feature of the solution as this also minimises the amount of time between each turn where the user is doing nothing, and so therefore fulfils the stakeholder’s requirements of the solution being efficient.

Additionally, stakeholders require the ability to be able to play against other stakeholders. This allows for a more competitive style of poker while also creating a more realistic experience which the end user requires. It also allows stakeholders who may not have the physical requirements to play poker, to virtually play poker in a way that is as close to the physical experience as possible. Furthermore, stakeholders may want to play with less than the minimum number of players required to start a game, so the solution should allow the end user to fill empty spaces with AI of their chosen difficulty, as this will make the table appear to be full.

### 1.3.3 Limitations of the proposed solution

Each game is kept local to each other. This means that any amount of winnings or losses in a game will not be transferred over to the next game. Players also do not have individual accounts where they would be able to have their own chip amount. Because of these reasons, having features such as daily bonuses, free chips, missions, and tasks for the user to complete are all impossible in my solution. However, the Poker Tutor could offer tasks within each game that would help newer players to make better choices, and they could be rewarded within each game. On the other hand, receiving daily free chips, cannot be implemented since each player does not have their own wallet that carries over through each game, and rewards such as rings, that require multiple games to earn are also unable to be implemented.

In addition, the solution will only include one type of poker. This means that the solution has a reduced variety, since stakeholders might want to play a different variation of poker, however the solution only includes Texas Hold’em. Because of this, stakeholders will only have the option to customize their Texas Hold’em experience and not have to chance to experiment with other forms of Poker. The solution will only include one type of Poker since multiple variations is outside the scope of my solution.

Moreover, it is impossible to implement a league or ranking system because no information from previous games will be stored, so the solution will not have any idea on how skilled the end user is. Since the solution will not store any data on each player, a global leader-board is not possible, and so the solution will not be able to compare the skill level of players with other players and so would not be able to fit players into leagues or ranks. The solution could give a rough skill rating at the end of each game; however, this would not carry over to later games.

## 1.4 Specifying the proposed solution

### 1.4.1 Solution’s requirements

The solution will require hardware inputs such as a mouse and keyboard. These inputs are necessary in order to play the game since the solution requires the user’s inputs, which will be done by clicking buttons using the mouse, or entering in username, or numbers (such as betting amount) using the keyboard.

Furthermore, the end user should have a monitor since the solution will display the poker game. In order to play Poker on my solution, the end user will have to see, the cards, bets, where to enter inputs and the menu screen, so the solution lends itself to some type of display. Furthermore, the solution will be in 1920x1080 as this is the resolution most monitors will be in and is the largest and clearest display that the majority of stakeholders will have.

The solution requires the operating system Windows 10, since this is the operating system, I will be developing my solution in and so will not work on other operating systems or previous versions. Moreover, Python 3.8.6 will be required as this is the version that I will be developing my solution in. The Python library Tkinter will also be required as this will be a key library that I will be using in the development process.

Python requires a 64-bit CPU, with 4 GB RAM and 5 GB of secondary storage, so these are all additional requirements for the end users’ desktop since they will be needing Python. Additionally, more secondary storage will be required for assets that my solution requires and for the solution itself, therefore the solution requires 8 GB secondary storage space in total, leaving 3 GB for assets and code.

### 1.4.2 Success criteria

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Comments** | **Justification** | **Criteria met Y/N** |
| Customization and variety | 5 or 9 players; starting amount of chips; blind sizes; AI difficulty; whether they want to play with AI; with other real people; or a mix of both. | Stakeholders require the solution to be suited towards them and allows users of different experiences to all interact appropriately with my solution. |  |
| Accessibility | Users should also be able to easily follow and navigate menus. Buttons, controls, icons and displays should all be straightforward and obvious. | Stakeholders necessitate that the solution should be professional and clean and new gamers should find the experience uncomplicated and trouble-free. |  |
| Realistic | The solution must be an accurate representation of poker and include all rules of poker in real life. The solution should be true-to-life, authentic and credible | This will improve the users’ experience – makes the solution enjoyable and pleasurable. Ensures the solution does not feel fake or virtual, which would make the user irritated. |  |
| Operational | The solution must be practical, functioning and well-performing, so that any inputs performs their intended purpose. | Confirms that the user is not irritated or unsatisfied while playing the solution. Eliminates any inconveniences for the user which might discourage them continuing their game. |  |
| Novice Friendly | The Poker Tutor should be able to suggest helpful moves and tips. Inexperienced gamers should be able to choose a low difficulty level AI | The solution should be appropriate for all end-users. Novice gamers or users with little experience in poker should find that my solution accommodates for their skill level. |  |
| Multiplayer | Stakeholders should be able to play against other stakeholders on different machines. | Stakeholders who do not have the physical means to play Poker can use my solution as a virtual way to play against others. A hot-seat style game, would be difficult since players have to keep their hole cards private. |  |
|  |  |  |  |