2020-05-28

The Year of the Cyber Athlete: Electric Boogaloo Edition

Matthew Duffy
Western Oregon University, mduffy16@mail.wou.edu

Western Oregon University

The Year of the Cyber Athlete: Electric Boogaloo Edition

M.P. Duffy

May 27, 2020

Abstract

The Year of the Cyber Athlete: Electric Boogaloo Edition is an aesthetic snapshot into the mind of a programmer who gets tasked with coding a project for a somewhat seedy individual. Our protagonist quickly turns the tables on this client and devises a plan so crazy.... it might just work. With a sordid, neo-noir vibe and a specially curated soundtrack, The Year of the Cyber Athlete: Electric Boogaloo Edition is the perfect way to ring in the new millennium! You can't afford to miss: The Year of the Cyber Athlete: Electric Boogaloo Edition coming May 28th, 2020 to an Academic Excellence Showcase near you!

Note: The Year of the Cyber Athlete: Electric Boogaloo Edition is best experienced at nighttime.

The Year of the Cyber Athlete: Electric Boogaloo Edition

You look at your watch. It's still 1999. The rain outside isn't falling as hard as you'd like it to, but that's okay. It's a quarter past one in the morning, and that's also okay. In fact, it's perfect. You just got some work. The client is another overweight, undereducated suit with a stiff neck. Also perfect. He wants three online blackjack casinos because those are apparently "where the money's going" these days. Whatever. He wants a standard model, a high-stakes model, and a "rapid fire" model. Cool. However, like any of these over-fraternized lackeys, they need to do it "different". As in, "do it my way". As in, "do it in a way that has received approximately eight seconds of thought without any testing or analysis but one in which my brain created and therefore is the objective best".

The client's grand idea is to use two types of perfect shuffles as the heart of the probability for these casinos. "Nobody has ever done it this way" was his rationale. I'm sure there are plenty of reasons why nobody has ever done it this way, a couple of which couldn't be any redder of flags if they tried, but once again, that's okay. The stupidity is part of the appeal of this line of work. That's because with stupidity, comes stupid money. The kind of money that can only be increased by having it and failing upwards. Your brain wonders how they acquire it to begin with. You remember that you aren't getting paid to unravel paradoxical socioeconomic philosophy, so you refocus on the task at hand.

Due to a truly higher than average level of stupidity from your client, he didn't make you sign or agree to much of anything, just a timeframe. So you'll whip up his casinos and since you haven't been preemptively banned from using any, milk them dry with your knowledge of perfect shuffles. You just need to do what this guy is truly mentally incapable of doing, and that's do a little number crunching first. Your specialty. You press play and get to work: **(Required listening during reading.)**

https://open.spotify.com/playlist/4qIq3wVqC1kQFNgGiKre7p?si=wue3A1M9SpGi7a1URIZyuw

Problem:

So the problem here is that it's going to take you a while to to code these casinos. Brainstorming this whole plan has already taken two hours and you'll need a solid 12 to code these things. The client gave you 24 hours which is way more than needed. However, after the 24 hours, they will all be available to the public and all activity logged. That means IP addresses, bets, amounts, all to and from whom and when, etc. Needless to say you can't bleed much of anything dry when everything will all be linked back to you. So you need to code the casinos, then do your thing in the leftover 10 hours, before the activity loggers are activated. The question here is, is it possible to reach the max earnings amount for each casino in the given time limit, while avoiding getting banned? If not, what is the maximum earnings you could make, given the fact that if you get banned from one, you get banned from all the others "below" it, but not "above" it? More on this in a bit.

Definitions:

Perfect Shuffles: A perfect shuffle is the act of splitting a deck of cards into two piles and interlacing one card at a time from each pile. There are two types. A perfect out-shuffle is when you start with the pile that has the first, or top-most, card from your deck before you split it. Consider a deck with the cards Ace, 2, 3, 4, 5, and 6. Let's denote it A23456. You'd split it into two piles so A23 and 456. With an out-shuffle, you'd start with the A23 pile. You'd flip over the A, then the 4, then back to the 2, then the 5, etc. You'd end up with A42536. With this method, the top and bottom cards of your deck always stay in the same position.

A perfect in-shuffle is when you start with the pile that does not have the first card from your deck. Let's look at our A23456 deck again. You'd split it into two piles so A23 and 456. With an in-shuffle, you'd start with the 456 pile. You'd flip over the 4, then the A, then back to the 5, then the 2, etc. You'd end up with

4A5263. With this method, the top and bottom cards of your deck change every shuffle.

Blackjack: We're going to assume people know the basics on how blackjack works. You try to get your cards' face values to add to 21, or as close as possible, as is the dealer. You can hit and get another card or stand and not get another. If you go over 21 you bust. An ace can be 11 or 1, player's choice. Jack, Queen, and King are all 10. You're trying to get closer to 21 than the dealer. If the dealer busts and you haven't, you win that round.

Player: This is you. You're playing blackjack. You get dealt cards by the dealer like real life. You must either hit or stand. Yup. Oh yeah, you know exactly how the deck is arranged every time.

Bots: These are the other computer players playing at your table. You will be playing with at least one bot, going up to 4 bots. So a maximum of 5 seats at a table. You and between 1 and 4 bots. Each round you get a random number of bots.

Rounds (or hands): Each round the dealer will do between 1 and 7 perfect in shuffles or between 1 and 7 perfect out shuffles, and then deal the cards. You get a random amount of bots each hand, and can place a bet each hand.

Dealer: The dealer deals the cards. Shocker. They are also technically a bot that's playing and trying to get close or equal to 21. They'll deal a card to their fellow bots at the table first, player being last, then to themselves. They'll then deal another card to each bot, player being last, and another card to themselves but face down.

Dealer, Bot, and Player Behavior: The dealer and bots will behave the same in regards to hitting/standing. If the bot's total is 17 or more, it must stand. If the total is 16 or under, they must hit and take a card. The bot must continue to take cards until the total is 17 or more, at which point the bot must stand. If the bot has an ace, and counting it as 11 would bring the total to 17 or more (but not over 21), the bot must count the ace as 11 and stand. [3] After all the bots are served, the player can choose whether to hit or stand. Since we know all about perfect shuffles, we'll know the dealer's face down card, and the next few cards in the deck. This will allow us to choose the best option to win, if possible. A tie can occur when the player and dealer both have the same amount. In this case, the player loses no money.

Casinos: There are three types of casinos. The High Stakes Casino, or HSC, has both a high minimum and maximum bet amount, low average hands per hour, and only 3 or 4 bots. The Rapid Fire Casino, or RFC, has both a low minimum and maximum bet amount, high average hands per hour, and only 1 or 2 bots. Finally, the Standard Casino, or SC, has between 1 and 4 bots, and pretty average betting amount and hands per hour. More on all this later.

Banning: You can get banned from the casinos if you do certain things like hit the maximum earnings amount for that casino, as well as a few other things. The HSC is Rank 3, SC Rank 2, and RFC Rank 1. If you get banned from one, you get banned from all others that are ranked lower. However, none that are ranked higher. More on this later.

Examples:

Framework Example:

So let's just run a quick example that utilizes our formulas to show why everything works the way it does. We'll take a a standard, unopened, unshuffled deck of 52 cards numbered 1 thru 52 and do, say, 3 perfect in-shuffles. Let's figure out what the first, say, 13 cards are in this "new" deck. Let's label these "new" cards a thru m. To find what card a is equal to, we can use a previously proved theorem about perfect shuffles which states: Suppose we have an even deck of size n. After k in shuffles, a card in position x will end up in position $2^k x \bmod (n+1)$.[1] So let's apply that theorem to our little example here to find what card a is.

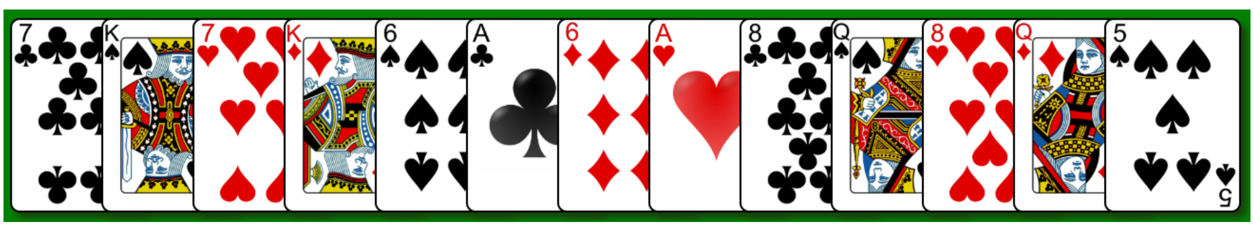
What we want to do is find the value of x where $2^3 x \bmod (52+1) = 1$ since a is in position 1 in our "new" deck. (If you wanted to find the card in position 2 or 3 in your new deck you would set it equal to 2 or 3.) What x here is, is the card in position x of a brand new deck. Simplifying this a bit, we want to find $1 \leq x \leq 52$ where $8x \bmod 53 = 1$. The value of x that works here is 20 since $8 * 20 = 160 \bmod 53 = 1$. That means the 20th card in an unshuffled, brand new deck is what our a card in our new deck is, and that is the 7 of Clubs.

This method works but is a bit tedious to find a thru m. However, we can use some programming to help:

https://playcode.io/609645

This program essentially does exactly what we just did, just multiple times. It looks at each card from a brand new 52 card deck in order, starting at the Ace of Hearts, and calculates it's new position in your deck after however many perfect in-shuffles you tell it to do. Once it finds the card that is in the first position in your new deck, it outputs its name. It then repeats this process for the second, third, fourth, etc cards, or however many you tell it to do.

So according to the program, our first 13 cards should be as follows: 7 of Clubs, King of Spades, 7 of Hearts, King of Diamonds, 6 of Spades, Ace of Clubs, 6 of Diamonds, Ace of Hearts, 8 of Clubs, Queen of Spades, 8 of Hearts, Queen of Diamonds, and then the 5 of Spades. Using the Faro Shuffle Simulator [4] and doing 3 in-shuffles, we see we are correct and our program functions as intended:



The program would be virtually identical for out-shuffles, just with the out shuffle formula plugged in instead of the in-shuffle one.

We can now use this to determine the exact order of any deck, no matter how many shuffles or what type. We will use this knowledge to setup the following outcome examples and explain them:

Easy Outcome Example 1:

The dealer will be doing 6 perfect out-shuffles. The player is sitting at a table with 3 bots. Cards are dealt and by our theorems and established behaviors we are given a King. The dealer gives himself a 4. More cards are dealt, and we are given a 7. The dealer plays his face down card and by our theorems we know that this is a 3. So the player has 17 while the dealer has 7.

The next phase is that each bot will either hit or stand based on the behaviors described above. This means: Bot 1 has an Ace and an 8, by the rules previously described Bot 1 stands with 19. Bot 2 has a 5 and a Queen, so he hits and gets a Queen. This makes 25 so he busts. Bot 3 has a 9 and a Jack, so he stands. So all 3 bots have been served and we are up.

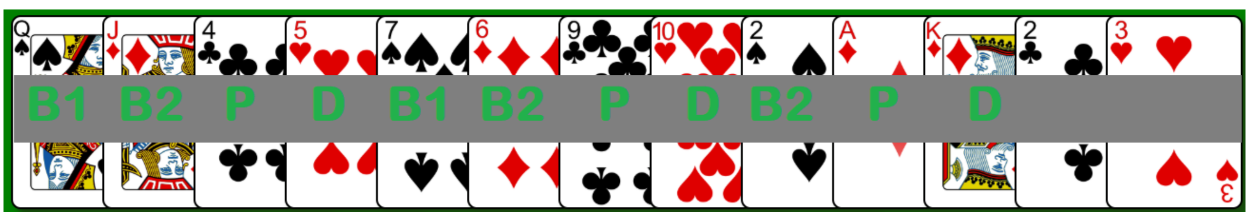
We have 17, but aren't required to hit/stand. The next card on the deck, after taking into account everyone's hits/stands, is an 8. This will bust us if we take it so we stand. The dealer takes the 8 giving themselves 15, and then by the rules they draw the 4 giving themselves 19 and then stands. We lose. Below is a picture detailing the deck and who gets what.

Medium Outcome Example:

The dealer will be doing 7 perfect in-shuffles. The player is sitting at a table with 2 bots. Cards are dealt and we are given a 4. The dealer gives himself a 5. More cards are dealt, and we are given a 9. The dealer plays his face down card and we know that this is a 10. So the player has 13 while the dealer has 15.

Now for the bots to go. Bot 1 has a Queen and a 7, so Bot 1 stands with 17. Bot 2 has a Jack and a 6, so he hits and gets a 2. This makes 18 so he stands. So both bots have been served and we are up.

We have 13 and the next card on the deck is an Ace. This will bust us if we take it as 11, so we take it as 1 giving us 14. We stand knowing the dealer will take the King since he has 15 and must hit, busting himself. We win.

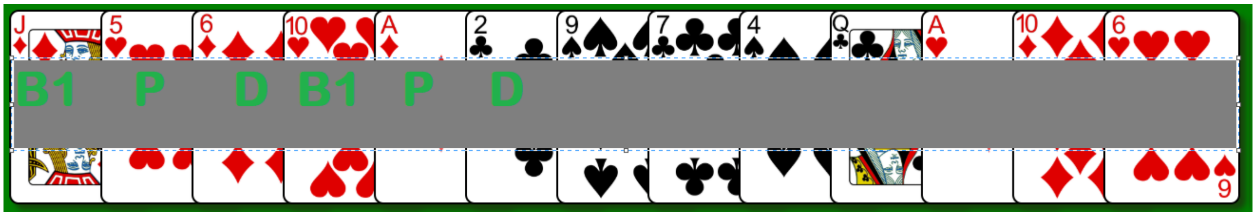


Complicated Outcome Example 1:

The dealer will be doing 6 perfect in-shuffles. The player is sitting at a table with 1 bot. Cards are dealt and we are given a 5. The dealer gives himself a 6. More cards are dealt, and we are given an Ace. The dealer plays his face down card and we know that this is a 2. So the player has 6 or 16 while the dealer has 8.

Now for the bot. Bot 1 has a Jack and a 10 and stands with 20. So the bot has been served and we are up.

We have 6 or 16 and the next card on the deck is a 9. With 16, the 9 would bust us. However, letting the dealer pick it up will give him 17, thus him standing and beating us. With 6 we could pick up the 9 giving us 15, but the dealer would pick up the 7 giving him 15, and then the 4 giving him 19. With 6 we couldn't pick up the 9 and 7 since we would get 22, so the dealer wins in any case here. We lose.

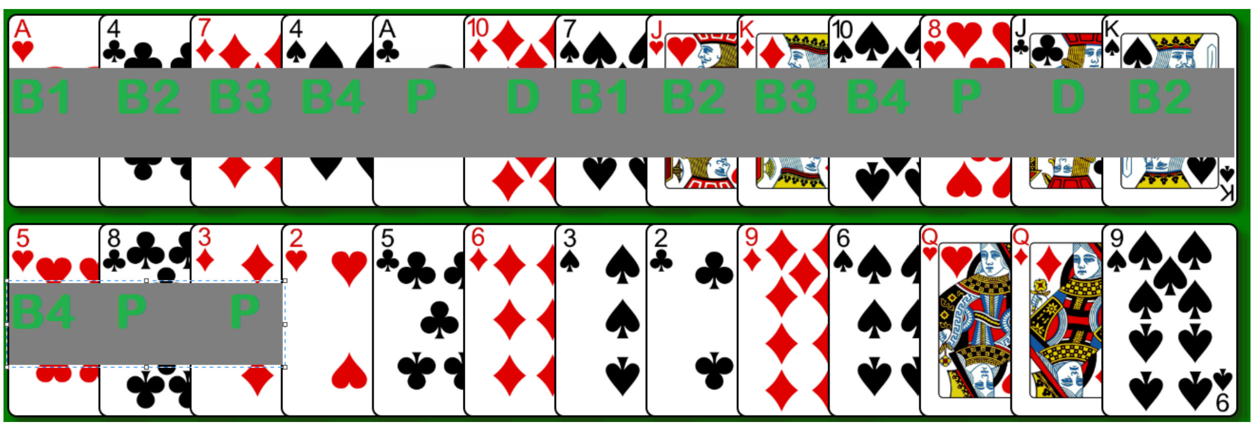


Complicated Outcome Example 2:

The dealer will be doing 4 perfect out-shuffles. The player is sitting at a table with 4 bots. Cards are dealt and we are given an Ace. The dealer gives himself a 10. More cards are dealt, and we are given an 8. The dealer plays his face down card and we know that this is a Jack. So the player has 9 or 19 while the dealer has 20.

Now for the bots. Bot 1 has an Ace and a 7 and stands with 18. Bot 2 has a 4 and a Jack, so with 14 Bot 2 hits getting the King and busting. Bot 3 has a 7 and a King, so with 17 Bot 3 stands. Finally, Bot 4 has a 4 and a 10, so with 14 Bot 4 hits getting the 5 and standing with 19. So the bots have been served and we are up.

We have 9 or 19 and the next card on the deck is an 8. With 19, the 8 would bust us. The dealer has 20 and won't be picking up anything. With 9 we could pick up the 8 giving us 17, then pick up the 3 giving us 20. The 2 will bust us so we stand with 20. The best case scenario here is to tie.



Outcomes:

I'm not going to go through every possible scenario step by step like I did in the examples. Just know that I'm using the same format, rules, and behaviors to determine if the player can win or not.

Perfect In-shuffles:

Win/Lose?	1 Shuffle	2 Shuffles	3 Shuffles	4 Shuffles	5 Shuffles	6 Shuffles	7 Shuffles
1 Bot	Win	Lose	Lose	Lose	Win	Lose	Lose
2 Bots	Tie	Win	Lose	Lose	Win	Lose	Win
3 Bots	Win	Win	Win	Win	Win	Lose	Lose
4 Bots	Win	Win	Tie	Tie	Win	Win	Lose

Perfect Out-shuffles:

Win/Lose?	1 Shuffle	2 Shuffles	3 Shuffles	4 Shuffles	5 Shuffles	6 Shuffles	7 Shuffles
1 Bot	Win	Win	Lose	Lose	Tie	Win	Win
2 Bots	Win	Tie	Tie	Win	Win	Lose	Win
3 Bots	Lose	Win	Win	Lose	Win	Lose	Win
4 Bots	Tie	Win	Win	Tie	Win	Win	Lose

We have that for the Standard Casino, you can win 30/56 scenarios, lose 18/56, and neither win nor lose anything 8/56.

For the High Stakes Casino, you are only playing with 3 or 4 bots, so you can win 17/28 scenarios, lose 7/28 and neither win nor lose anything 4/28.

For the Rapid Fire Casino, you are only playing with 1 or 2 bots, so you can win 13/28 scenarios, lose 11/28 and neither win nor lose anything 4/28.

Hands Per Hour:

So according to Jim Kilby, the Boyd Professor of Gaming at the University of Nevada, Las Vegas, his book[2] states that casinos typically shoot for a certain number of blackjack hands per hour, depending on the amount of people playing at the table. Below is a table with the typical hands per hour from his book.

Players	Hands Per Hour
1	209
2	139
3	105
4	84
5	70

Since we will always be playing with at least one other player, we can ignore the top row. It's worth remembering that the number of bots the player will have will be random, depending on what casino they're in.

So for the Standard Casino, there will be between 1 and 4 bots plus the player so between 2 and 5 players each hand. Bots will come and go randomly each round. So we can just average the hands per hour for rows

2 through 5, giving us 99.5 hands per hour, or we'll just say 100.

For the High Stakes Casino, there will only be 3 or 4 bots, so total players will be 4 or 5. Averaging those values we get 77 hands per hour on average.

Finally, for the Rapid Fire Casino, there are only 1 or 2 bots, making the amount of players either 2 or 3 and thus the average hands per round to be 122. Here's a table for reference. I included hands per minute and seconds per hand to help illustrate how fast things are going. These are approximations:

Casino	Avg. Hands Per Hour	Avg. Hands Per Minute	Avg. Seconds per Hand
HSC	77	1.28	47
SC	100	1.67	36
RFC	122	2.03	30

Wins & Losses per Hour:

So looking at the HSC, we will average 77 hands of blackjack in a given hour. From our previous observations we can win 17/28 scenarios, lose 7/28, and neither win nor lose anything 4/28, since we're only playing with 3 or 4 bots. So in a given hour after playing 77 hands we should win approximately ((17/28) * 77) = 47 hands, lose ((7/28) * 77) = 19 hands, and tie ((4/28) * 77) = 11 hands.

Now looking at the SC, we will average 100 hands of blackjack in a given hour. We can win 30/56 scenarios, lose 18/56, and neither win nor lose anything 8/56. So in a given hour after playing 100 hands we should win approximately ((30/56) * 100) = 54 hands, lose ((18/56) * 100) = 32 hands, and tie ((8/56) * 100) = 14 hands.

Finally looking at the RFC, we will average 122 hands of blackjack in a given hour. We can win 13/28 scenarios, lose 11/28 and neither win nor lose anything 4/28, since we're only playing with 1 or 2 bots. So in a given hour after playing 122 hands we should win approximately ((13/28) * 122) = 57 hands, lose ((11/28) * 122) = 48 hands, and tie ((4/28) * 122) = 17 hands. Here's another, slightly more complicated, table. The percentages are approximate:

Casino	Hands Per Hour	Wins per Hour	Win Percentage	Losses per Hour	Loss Percentage	Ties per Hour	Tie Percentage
HSC	77	47	61.04	19	24.66	11	14.29
SC	100	54	54.00	32	32.00	14	14.00
RFC	122	57	46.72	48	39.34	17	13.93

It's clear that the HSC has the highest chance to win and lowest chance to lose. The RFC is the opposite with the lowest chance to win and highest chance to lose. Is it even possible to consistently make money in the RFC since the win percentage is less than fifty percent? We'll soon find out but due to the HSC's higher max bet amount and high win rate, this will definitely be where we make the most money. However, if we get banned from it we get banned from the other two casinos as well, so we'll want to save this casino for last and milk the other two casinos first, for as long as possible, that is.

Betting:

Okay time for the brass tacks. Betting. How does betting in Blackjack work? The standard payout for blackjack is pretty simple. If you bet $10 and win, you get $10. If you lose, you lose your $10. If you tie nothing happens. You place a bet at the start of each hand. Typically there are special things you can do like splitting and insurance but we're going to only focus on basic betting for the scope of this application. Our three casinos each have slightly different rules about minimum/maximum betting/winning amounts. Like most casinos, each has a minimum and maximum bet amount, and a maximum winnings amount before you get banned. Here's another table:

Casino	Min. Bet Amount	Max. Bet Amount	Max. Earnings Amount
HSC	100	10,000	1,000,000
SC	10	1,000	100,000
RFC	1	100	10,000

So let's walk through some wagers at each casino. If you win you win you bet amount, if you tie you gain/lose nothing, and if you lose you lose your full bet.

We'll start with the RFC. Let's do a $20 dollar wager each hand for an hour. So we have 122 hands. A win with a $20 bet will return $20. We'll win 57 times with 20 * 57 = $1,140 in winnings. We'll lose 48 times and lose 20 * 48 = $960. We'll tie 17 times and have nothing happen. So in an hour we can expect to make $1,140 - $960 = $180.

Next let's look at the SC. Let's do a $100 dollar wager each hand for an hour. So we have 100 hands. We'll win 54 times with 100 * 54 = $5,400 in winnings. We'll lose 32 times and lose 100 * 32 = $3,200. We'll tie 14 times and have nothing happen. So in an hour we can expect to make $5,400 - $3,200 = $2,200.

Finally let's look at the HSC. Let's do a $2,000 dollar wager each hand for an hour. So we have 77 hands. We'll win 47 times with 2,000 * 47 = $94,000 in winnings. We'll lose 19 times and lose 2,000 * 19 = $38,000. We'll tie 11 times and have nothing happen. So in an hour we can expect to make $94,000 - $38,000 = $56,000.

The important thing here is that betting increases proportionally. You can see this by betting $1, $2, and $3 at each casino. Yes, this violates the minimum bet amount but for the sake of explaining this we'll examine it. The following table is earnings per hour betting $1, $2, and $3. Earnings per hour is just winnings per hour minus losses per hour.

Casino	$1 Bet	$2 Bet	$3 Bet	Difference
HSC	28	56	84	+28
SC	22	44	66	+22
RFC	9	18	27	+9

What this means is, looking at RFC for example, you'll make $9n per hour where n is your bet amount. $22n per hour for the SC and $28n per hour for the HSC, once again where n is your bet amount. This makes sense since 9*20=180 as desired from our above RFC example.

You're probably wondering why not just bet the maximum over and over? Not only do the casinos kick you out after you earn a certain amount, but you can also get kicked out for suspicious betting. Betting ten grand over and over is the definition of suspicious. So we'll need to vary our bet amounts. I'm going to use some programming for this. Looking at the RFC for example, you can bet inclusively between 1 and 100. I'm going to use a little programming to show how you can calculate your expected earnings even with random bets.

https://playcode.io/609640

The program I made takes the casino information as inputs, so Difference and then Average Hands per Hour, and then the bet range. For the RFC example we can bet randomly between 10 and 90 inclusive. It then asks for the amount of hours you intend to play for. We'll put in 6 to show your expected earnings after six hours of playing non-stop. What this program does is it simulates hands, in this case each with a random bet amount between 10 and 90. It uses the fact we discovered that you'll make 9n dollars per hour in the RFC where n is your bet amount, so it multiplies your bet by 9, then divides by 122 to get our average winnings for that particular hand. It then repeats this process 122 times to simulate a full hour of hands. It then repeats that entire process 6 times, for 6 hours of play.

So for this example each execution of the program is doing 732 hands of blackjack. The whole time it is logging your total earnings over the entire 732 hands. It then finds your total earnings, average earnings per hand, and thus your average bet amount per hand. In this case our average earnings amount is consistently around 450. Sometimes a little under, or sometimes a little over, but that's okay, since we are using random numbers. On average it's right around 450 like we want. Our "margin of error" isn't really important since we're just trying to get an idea of how long this whole process will take. It just so happens that 450=9*50. So what we proved is that on average, betting randomly from a range is the same as betting (min+max)/2 over and over where min and max are the endpoints of your betting range. What this means is that with this program we can make "random" bets yet still be able to calculate our average expected earnings. This will come in handy in the next section.

Banning:

Okay so each casino has to be programmed with a built-in banning system to catch cheaters. Yes we're programming our own police essentially but that's okay. There are a number of ways to get banned. First, hitting the max earnings amount. Earnings is your winnings minus your losses. Another way is consistently betting the max. It's really there as a way to catch blatant showboating idiots and kick them out, so it should be avoided if at all possible. Once again, the client is not smart and thus we can simply bet one under the max.

Finally, the client is using some anti-card counting proprietary software that is so poorly coded it makes you laugh. You anticipate a lot of people getting unjustly banned, but whatever. Basically it sucks so much that you're forced to do random bets with a range of, say, at least ten percent of the max just to be safe, or else risk getting banned for some bizarre consistent betting thing where it thinks you're cheating or something. Good thing we made a program that bets randomly within a specified range already! So the most efficient way to follow these rules is to simply bet within a range of ten percent of the max, with the max-1 being your upper bound.

So for the RFC the max is 100 and ten percent of that is 10, and the max minus one is 99, and 99-10 is 89 so we will be randomly betting between 89 and 99. Going off our previous work we know that randomly betting between 89 and 99 is the same as betting 94 over and over. 94*9=846 so we should expect to make that every hour.

For the SC the max is 1000 and ten percent of that is 100, and the max minus one is 999, and 999-100 is 899 so we will be randomly betting between 899 and 999. We know that randomly betting between 899 and 999 is the same as betting 949 over and over. 949*22=21,868 so we should expect to make that every hour.

Finally for the HSC, we have that the max is 10,000 and ten percent of that is 1000, and the max minus one is 9,999, and 9,999-1000 is 8,999 so we will be randomly betting between 8,999 and 9,999. We know that randomly betting between 8,999 and 9,999 is the same as betting 9,499 over and over. 9,499*28=265,972 so we should expect to make that every hour.

By our above calculations we should expect to hit the max for the HSC in about 4 hours, the max for the SC in about 5 hours and the max for the RFC in about 12 hours. We can confirm these by plugging in the corresponding casino and bet information into our random betting program. Here's a table:

Casino	Hours to Max Earn.	Average Bet	Avg Earnings per Hr
HSC	3.75	9,499	265,972
SC	4.8	949	20,878
RFC	11.8	94	846

So time to solve our problem. Our goal here is to simply maximize earnings. This can easily be done with the HFC. 1 million in under 4 hours no problem. Well, besides the one where we get instantly banned from the other two casinos. We're going to want to work our way up. We know we have 10 hours. 3.75 will be spent in the HFC, 4.8 in the SC, both of which will be completely maxed out, so that leaves 1.45 hours left in the RFC. So we have 1.45*846= approximately 1,226. So we'll spend 1.45 hours in the RFC, then 4.8 in the SC and get banned, then move up to the HSC and max that out and get banned again. So here's our breakdown of maximum earnings:

Casino	Hours Spent	Earnings
RFC	1.45	1,226
SC	4.8	100,000
HSC	3.75	1,000,000
Totals:	**10**	**1,101,226**

Conclusion:

You look at your watch. It's 2000. You watch the credits flow into you account in real time. It used to be the biggest rush there was, now you do it more out of tradition than anything else. You decide to try to relax a bit and gaze outside, to your lovely next-door neighbor Apartment Building #725. You just made two months rent in a literal days work. You feel like you're cheating the system, and you are. And once again, that's okay.

You transfer your credits to an out-of-network account for safety and block the client on everything another human being can be blocked on. That's quite a lot of things these days so it takes you a few hours more. Finally, you throw yourself on the bed, exhausted, letting gravity take over, albeit just for a second, and try to relax a bit more. That's a tall order for you lately, so you flip the tape and hit play one last time: **(Required listening during relaxing.)**

https://open.spotify.com/playlist/4OwAvyzODXzqmcfCOnehte?si=1bwxIewjQ7m6qrVTMUYxww

As you doze off, that feeling comes back. No, not that one. The other one. The good one. The feeling that you've already made it, and that everything else is just one big victory lap. You like that one.

M.P. Duffy, 2020.

References

[1] Benham, H., Carlson, M., Duffy, M., Lamont, R., Maddox, Z., and Osborne, K. *card stuff, or An Investigative Analytical Deep Dive into the Chaotic yet Orderly Phenomenon Known as the Perfect Shuffle*, Western Oregon University, Mathematics Department, March, 2020.

[2] Kilby, J., *Casino Operations Management*, 2nd Edition, John Wiley & Sons, 2004.

[3] Bicycle Playing Cards, "How to Play: Blackjack", https://bicyclecards.com/how-to-play/blackjack/

[4] Wallace, R., Faro Shuffle Simulator, http://robertjwallace.com/faro/