

SIMULACIÓN DE BLACKJACK:



Autores:

Carlos Gómez

Gabriela Gutiérrez

Alejandro Jarabo

Alejandro Ramos

Grupo: 23.1

Asignatura: Señales Aleatorias

Índice:

- 1. Introducción.*
- 2. Reglas del juego.*
- 3. Programa.*
- 4. Estudio de probabilidades.*
- 5. Conclusiones*

1. Introducción:

Nuestro trabajo se basa en una simulación de un juego de cartas basada en el blackjack o veintiuno.

De forma general debemos explicar que el blackjack es un juego habitualmente jugado en casinos, en el que juegan de cuatro a siete jugadores con una o más barajas inglesas, de 52 cartas cada una, omitiendo los comodines. Cada participante juega únicamente contra el crupier, con el fin de conseguir una mejor jugada que este. Por otro lado el crupier debe jugar según unas reglas fijas implantadas por el casino, por lo que no tiene capacidad de decisión.

De esta forma nos centraremos en las probabilidades de nosotros, como uno de los jugadores, de obtener 21 puntos sin pasarnos de esta cantidad en cada una de las rondas, conociendo los puntos del crupier y el resto de cartas que quedan en la baraja, para poder ganar el juego y aumentar nuestro capital inicial en cada jugada.

Luego indirectamente nuestro objetivo es conseguir saber la probabilidad de incrementar el capital del que partimos como jugador o por el contrario la probabilidad de reducir este capital basándonos en las cartas de las que disponemos, como consecuencia de perder la partida.

2. Reglas del juego:

En nuestra simulación utilizaremos una baraja inglesa de 52 cartas, sin comodines, donde los valores numéricos tienen el mismo valor en puntos, las figuras (J, Q, K) suman cada una diez puntos y por último los As, suman uno u once puntos según más convenga la jugador.

Para este juego que se basa en el blackjack aunque con ciertos rasgos distintos, el objetivo principal de cada jugador es llegar a 21 puntos sin pasarse, o al menos acercarse lo máximo posible a esa cantidad para ganar al crupier.

El juego comienza con el crupier repartiendo dos cartas a cada jugador y una a sí mismo. En base a la suma de sus cartas, cada jugador puede decidir si plantarse con esos puntos o por el contrario pedir otra carta, se pueden llegar a pedir hasta tres cartas más, teniendo un máximo de cinco cartas por rondas.

Partiendo de esta base debemos definir las reglas básicas que debe seguir el crupier por norma. Este en el juego siempre debe pedir otra carta si sus puntos suman 16 o menos, y plantarse con 17 puntos o más, el resto de jugadores juegan en conocimiento de las reglas que debe seguir. Así el crupier puede: ganar la partida con 21 puntos o una suma cercana a 21 más próxima que la de sus oponentes, perder la partida con una suma más alejada de 21 que los otros jugadores o pasándose de dicha cantidad, por último puede quedar en empate si suman la misma cantidad de puntos.

Sabiendo esto, hay cuatro formas posibles de que acabe el juego para uno de los jugadores:

- Pasándose de 21 puntos, si esto ocurre el jugador queda inmediatamente eliminado y pierde su apuesta inicial.
- Ganar con 21 puntos, o una cantidad más próxima a 21 que el crupier, recuperando su apuesta y ganando la misma cantidad apostada.
- Obtener igual suma que el crupier o que tanto el crupier como el jugador superen los 21 puntos, quedando así en empate y recuperando únicamente la mitad de su apuesta inicial.
- Obtener una suma más alejada de 21 que el crupier, perdiendo la partida y su apuesta inicial.

3. Programa:

Nuestro programa es una simulación completa del juego basándonos en las reglas anteriormente descritas. Creamos un juego en el lenguaje Java, de forma interactiva dado a que el participante no sólo obtiene datos del juego sacados por pantalla, sino que también puede introducir su apuesta y tomar decisiones sobre como seguirá su turno según las cartas que le correspondan, una vez esto ocurre el programa le dice si gana, pierde o empata con el crupier.

Hemos desglosado el programa en tres clases principales: Cartas, Jugador y BlackJack, cada una contiene una serie de atributos y métodos que componen el programa.

3.1 Cartas:

Esta clase se compone de un atributo principal llamado barajas que es de tipo array de números enteros, el cual tiene una longitud de 52 posiciones y dentro de él hemos almacenado cada valor de una baraja inglesa, los números con su valor numérico y las figuras con un valor de 10 puntos. Además tiene un atributo aleatorio (r) de la clase Random de java.

Cartas está formada varios métodos, los principales los describiremos a continuación:

Su primer método getCarta, devuelve un valor de tipo int (numero entero) sacado del array de baraja. Dicho valor es una correspondencia a una carta del juego y se obtiene de forma aleatoria con el método nextInt (52).

El siguiente método posibles, devuelve un número entero que corresponde a los casos posibles de que la suma de las cartas de un jugador más cada número dentro de array (cartas restantes en la baraja) no supere los 21 puntos. En consecuencia con este método, el siguiente llamado pPedirCartaSinPasarse, calcula la probabilidad de que con las cartas que ya tenemos podamos pedir una carta más sin pasarnos de los 21 puntos, dividiendo el valor del resultado del método anterior entre las posiciones distintas de cero del array barajas.

El próximo método de esta clase se denomina pConseguir21, sirve para calcular la probabilidad que tiene el jugador de que la suma de sus cartas más una siguiente aleatoria resulte exactamente 21. Este valor será cero si el jugador ya se ha pasado de 21, tiene esta cantidad, o de otra forma si no quedan cartas en la baraja con las que completar su suma de 21.

Finalmente añadimos un último método, Bernoulli, que lo utilizamos para calcular la probabilidad de obtener Blackjack en un determinado número de rondas, partiendo de otros métodos subyacentes a este para conseguir dicho calculo. Cabe mencionar que este método interactúa con el jugador ya que este último decide el número de rondas y de veces que se desea obtener 21 puntos.

3.2 Jugador:

La clase jugador solo tiene atributo principal llamado cartas, compuesto por un array de números cuya máxima longitud es de cinco posiciones, cada puesto del array se llena con el valor de la carta que se le entrega al jugador.

A partir de aquí contiene dos métodos getters: getCarta y getValor, que devuelven el primero el valor de la carta dada al jugador y el segundo analiza si dicho valor "As" o "Figura". Además tiene un método set: setCarta que coloca dentro del array del jugador (su mano, en el juego real) el valor de la carta que se obtiene con los dos métodos anteriores.

Continuamos describiendo un método llamado comparaJugada, con el que comparamos las sumas de las cartas de dos jugadores, en nuestro caso siempre se comparará nuestra jugada con la del crupier. Este método es el que permite saber si se gana, se pierde o se empata ya que evalúa dichas sumas respecto de 21 puntos máximos y entre los propios jugadores. Para poder ejecutar este método a continuación creamos suma, un método que devuelve la suma total de las cartas de un jugador determinado.

El siguiente método, desborda, devuelve un tipo boolean, siendo true si la suma de puntos en las cartas de un jugador excede de 21 puntos y false en caso contrario. Así creamos el método convertidor que en caso de obtener un true anteriormente, cambia el As de valor 11 a valor 1, para que el jugador no pierda por desbordarse con uno o varios ases.

3.3 BlackJack:

La clase BlackJack contiene el método principal main que ejecutara todo el programa y varias frases que saldrán por pantallas que informaran al jugador de sus cartas y de la carta del crupier, además de si ha ganado, perdido o empatado, y de la probabilidad que tiene según sus cartas de obtener 21 o de no pasarse de dicha puntuación. Le hemos incluido varios scanner que permiten decidir al jugador si desea continuar el juego y cuanto desea apostar introduciendo los comandos necesarios.

Inicia creado cuatro jugadores que participaran en el juego, la persona que interactúa con el programa será el jugador 2 y el crupier es siempre el jugador 0, sigue repartiendo las cartas a cada jugador en el orden un a cada jugador, luego el crupier y otra segunda carta a cada jugador, esto lo realiza llamando a métodos getters y setter antes descritos.

A partir de este punto el jugador 2 (interactuando con el juego) conoce sus cartas, la carta del crupier y las probabilidades de obtener 21 puntos exactos o de no pasarse de ese máximo en caso de pedir otra carta. Según estos datos es el momento en que dicho jugador debe decidir cuánto apuesta y si desea una carta más o terminar la ronda con las cartas que ya tiene en la mano, esto se repetirá hasta un máximo de 5 veces según las decisiones del jugador de querer o no una carta más, dentro de una misma ronda.

4. Estudio de la Probabilidad:

En esta simulación estudiaremos en una primera parte probabilidades calculadas a través del programa en java.

Primeramente el programa calcula la probabilidad de una vez el jugador conoce sus dos primeras cartas, y el programa internamente conoce el valor de las cartas del resto de jugadores más las cartas de crupier, este calcula la probabilidad de que el jugador que interactúa con el juego (jugador 2), obtenga una suma total en sus cartas igual o menor que 21 puntos en caso de que pidiese una carta adicional a las que tiene.

Esto se calcula mediante el Caso de Laplace, el programa en este caso suma el valor total de las dos cartas con cada posición de las cartas del array barajas y cuenta las veces que esta suma es menor de 21, lo que representa los casos favorables. Luego cuenta la cantidad de posiciones de dicho array que son distintas de cero, que representaría a los casos posibles. Por último y siguiendo el Caso de Laplace dividimos los casos probables entre los posibles.

La siguiente probabilidad calculada por el programa representa la posibilidad de obtener exactamente 21 puntos una vez sabiendo el valor de nuestras dos primeras cartas. Esto se calcula de la misma forma que el anterior, solo que los casos favorables se reducen a las veces en que la suma total de las dos cartas iniciales del jugador más una tercera (no conocida) sea de 21 puntos.

Además de estos dos casos de Laplace, el programa desarrolla y calcula otro tipo de probabilidad basándose en el Ensayo de Bernoulli, para esto se crearon posteriormente métodos añadidos a la clase cartas: pSacarBlackJack, mediaBlackJack, factorial, combinatoria y bernoulli.

Primeramente explicamos en que el Ensayo de Bernoulli en este caso consiste en un experimento compuesto por n ensayos (número de rondas, introducido por el jugador) en que solo hay dos sucesos posibles: obtener Blackjack (llegar a 21 puntos exactos con las dos primeras cartas), A y su caso opuesto de no conseguir blackjack.

Queremos calcular P (B), es decir, la probabilidad de que A ocurra un numero k de veces, este número también debe introducirlo el jugador. Para ello nos basamos en la fórmula:

$$P(B) = \binom{n}{k} [P(A)]^k * [1 - P(A)]^{n-k}$$

Así el jugador conoce su posibilidad de obtener blackjack en un número de rondas y veces determinado por él.

Por supuesto se puede comprobar usando el programa que mientras más alto sea el número de veces, menor es la probabilidad de obtener blackjack de forma seguida.

Por ultimo añadimos una probabilidad calculada al inicio de cada ronda que nos dice la probabilidad en esa ronda de obtener Blackjack, para ello tiene en cuenta las cartas que

han sido sacadas de la baraja, y según aquellas que no han sido descartadas aun, calcula los casos favorables según la cantidad de veces que la suma de dos de las cartas restantes en la baraja sumen entre ellas 21 puntos y de que puedan salir repartidas al jugador en cuestión, en este caso el jugador 2. En resumen, calcula la probabilidad de obtener seguidamente dos cartas distintas cuya suma de valores sea el de 21.

5. Conclusiones:

Para concluir nos gustaría resumidamente explicar la función del programa.

Este programa, Blackjack, es un juego como ya hemos explicado que su objetivo es apostar y ganar dinero consiguiendo ganar al crupier.

Nuestro programa tiene un bote inicial de 100€, del cual el jugador dispone y puede realizar apuestas en cada ronda según su decisión personal y las cartas suyas, del crupier y del resto de jugadores.

A diferencia del juego real cabe destacar que ofrecemos información vital al jugador calculando las probabilidades antes descritas, ya que a pesar de que el juego se base en las variables aleatorias, las cartas desconocidas que no sabemos qué valor siguiente puede tocarnos a nosotros o al resto de participantes, el jugador dispone de la probabilidad que tiene de sacar 21 o no pasarlos, incluso de obtener 21 puntos desde el inicio o de llegar a ese valor varias veces seguidas. Esta información extra sirve para guiar al jugador y que este en base a ello pueda apostar más o menos dinero según las probabilidades.

