

MANUAL TÉCNICO

En el siguiente documento se presenta una descripción de la estructura interna que posee el juego con la cual opera, además de unas recomendaciones que ayudaran al correcto funcionamiento del mismo.

El juego fue desarrollado con tecnología HTML5, CSS y JavaScript. El lenguaje JavaScript presenta una “similitud” con el lenguaje Java, puesto que mediante unas modificaciones podemos simular programación orientada a objetos (POO).

A continuación, se presentan las “clases” con sus “atributos” y “métodos” más relevantes:

Clase Banco: En esta clase es en donde implementamos el algoritmo de devuelta de monedas utilizando la técnica Backtracking conociendo la cantidad de monedas del banco, en este caso, del casino, que es en donde se deposita o se retira el dinero de los jugadores.

Clase Ganador: esta clase nos es útil para generar el número aleatorio que definirá quien ganó o perdió su apuesta. Además, esta clase dota las características de los números en este juego de la ruleta, es decir, identifica qué números son pares, impares, rojo, negro, 1º Docena, “18 a 36” ... asignando en un vector valores que posiciona a un número en los conjuntos correspondientes a los cuales pertenece; información con la que procedemos a identificar **ganadores** de una manera dinámica.

Clase Nodo: El corazón de las listas ligadas, puesto que este tipo de lista es la conexión lógica de varios de estos nodos. Al ser un nodo doblemente enlazado, contamos con los atributos dato, siguiente y anterior, siendo los dos últimos, referencias de conexión lógica a los demás nodos en la lista creando una estructura lógica muy dinámica a la hora de la práctica.

Durante el juego, a veces tenemos prioridades sólo en datos específicos, es por esto que manejamos las clases `NodoApuesta` y `NodoTablero`.

Clase `NodoTablero`: a la hora de crear el tablero, nos interesará la información con la que se trabajará de manera en general, sin importar los cambios que se presenten en la ejecución del juego. Esta información la representamos a manera de atributos, utilizando las variables:

`numero`: variable que almacena una de todas las posibilidades a la que el usuario puede apostar.

`valorAMultiplicar`: puesto que, al tener diferentes opciones de ganancia para los usuarios, aquí es en donde representamos la ganancia que obtendrá el jugador para un número apostado en específico. Las distintas cantidades de ganancia están expresadas de manera detallada en el manual de usuario.

`tipo`: el usuario puede apostar a un solo número en específico, como también a un conjunto de números. Almacenamos 0 si es el primer caso, y 1 para el otro caso.

Clase `NodoApuesta`: clase implementada a la hora de crear las listas de apuestas de los usuarios, pues aquí es en donde almacenamos los valores de las variables fundamentales para que el juego tenga el funcionamiento adecuado. Dichas variables son:

Las descritas en la clase `NodoTablero`.

`valorApostado`: cantidad ingresada por el usuario para saber cuánto hay que pagarle a cada jugador en caso de ser un ganador. Ese valor a pagar se calcula:

`valorAMultiplicar * valorApostado`

Clase `ListaDoblementeLigadaCircular`: Es con esta clase que representamos el tablero de apuestas y los jugadores, puesto que un usuario al apostar, está creando una lista de

posibilidades con las que el juego puede determinar la situación o el estado del jugador. En esta clase, además de contar con los métodos básicos de agregar, buscar, eliminar, recorrer y tamaño, contamos con el método `buscaGanador`, que recibe un dato como parámetro y lo que hará es recorrer la lista en busca del nodo que contiene dicho dato para acceder a sus atributos de valor y valor a multiplicar, y así obtener el valor que se le pagará al usuario en caso de que contenga el dato ingresado como parámetro.

La implementación de estas clases la encontraremos en el archivo `'ruleta.js'` que es el encargado de ejecutar las funciones que le dan vida, funcionalidad, dinamismo, al juego, junto con las instrucciones que crean los objetos de dichas clases y el llamado de sus respectivos métodos.

Inicialmente declaramos y definimos todas las variables, arreglos y objetos que serán de utilidad a lo largo del desarrollo del juego. Luego, estaremos usando las siguientes funciones que se encargarán del funcionamiento del juego:

recibeApuesta: esta función se encargará de identificar en cual opción de apuesta está interesado el jugador para que luego proceda a preguntar el valor que le desea apostar a dicha opción. Controla que los datos ingresados sean los adecuados según las reglas del juego.

dineroInicial: función que pedirá el ingreso de la cantidad de dinero inicial con la que cuenta el jugador al principio del juego. Controlando nuevamente que los datos ingresados sean válidos.

principal: esta es encargada de recolectar los datos de la partida para operar sobre ellos. Con los datos recolectados creará las listas de apuestas de los jugadores. Controlará si el jugador desea o no participar en la ronda actual. Controlará siempre que los datos sean los adecuados para la dinámica del juego. Los parámetros para esta función, identifica qué jugador está interactuando con la interfaz para la actualización acertada de dichos datos.

crearListaApuestas: es la encargada de conectar de manera lógica las distintas opciones de apuesta de cada jugador, para que su manipulación sea dinámica en la práctica. Además de hacerlo de manera lógica, también le informa la información que lleva hasta el momento a través del apartado de Apuestas Jugador.

confirmarApuesta: se encargará de que al confirmar el jugador sus apuestas para la ronda actual, no pueda modificar dichas apuestas y darles paso a las apuestas de los demás jugadores en el caso de que haya varios.

jugar: función que se encargará de solicitar que se genere el número aleatorio con el que se definirá qué jugadores obtienen ganancia y tengan la opción de reclamar su premio en caso de ser ganadores.

buscaValorADevolver: Se encargará de llamar el método que determinan la cantidad de dinero que se debe devolver y se muestra al jugador su ganancia, ya sea nula o positiva.

Confirma: simplemente preguntará al jugador si la apuesta que acabó de realizar la desea incluir en su juego, por si en algún momento se ha cometido un error.

muestraLLista: Con fines de modificación de la interfaz que ve el jugador (o jugadores) hace posible la visualización de las apuestas que se lleva hasta el momento de la ronda actual.

devolver: encargada de llamar el método que determina la cantidad de monedas que ha ganado el jugador con el valor de ganancia y mostrándola a su vez en pantalla para informar al jugador. Si por alguna razón el banco se encuentra en un estado de bancarrota y no puede pagar más apuestas, se controla simplemente acabando el juego.

eliminarApuesta: Si el jugador en un momento determinado desea cancelar una apuesta realizada, la función se encargará de eliminar dicha apuesta, tanto de manera lógica haciendo los correspondientes llamados de los métodos que permiten esta acción, como de manera gráfica en el apartado de Apuestas Jugador, mostrando a su vez un mensaje que informa que la eliminación se hizo de manera exitosa. Controla si la apuesta que va a eliminar no exista o que simplemente no haya apuestas para la ronda actual.

limpiarListas: con fines meramente gráficos, una vez sea empezada una nueva ronda, la información que se suministraba anteriormente sea eliminada para una mejor legibilidad de la información.

numeroJugadores: función ejecutada al principio del juego donde pide la cantidad de jugadores para la partida, para que los métodos e instrucciones ejecutadas durante el juego se realicen de manera óptima. Controla que el dato ingresado no se salga de los rangos especificados.

RECOMENDACIONES

Jugar preferiblemente en los navegadores



*[Google Chrome](#)



*[Mozilla Firefox](#)



*[Torch](#)



*[Microsoft Edge](#)

AVISO



**NO RECARGAR LA PAGINA MIENTRAS SE ESTÁ EN JUEGO, PUES LA
INFORMACIÓN DE LO TRANSCURRIDO DEL JUEGO SE PERDERÁ
LA INTERFAZ GRÁFICA SE PUEDE VER ALTERADA POR LA RELACIÓN DE
ASPECTO DE CADA PANTALLA**

REALIZADO POR:

Daniela Ocampo

Julián Agudelo

Estudiantes de ingeniería de sistemas

Universidad de Antioquia



2019