Proyecto 1:

Siete y medio

Introducción

El siete y medio es un juego de cartas que utiliza la baraja española de 40 cartas. El juego consiste en obtener siete puntos y medio, o acercarse a esta puntuación lo más posible. Las cartas tienen, indistintamente de su palo, el valor que indica su propio índice, excepto las figuras (sota, caballo y rey) que tienen una valor de medio punto cada una.

El objetivo es ganar los puntos apostados en cada tanda. En cada apuesta, cada jugador compite contra la banca, para ganar la apuesta el objetivo es intentar sumar siete y medio o el número más cercano sin pasarse de esta cantidad.

Resumen del juego

- El número de jugadores debe estar entre 2 y 8.
- Se jugará un máximo de 30 manos.
- Cada jugador inicia la partida con 20 puntos.
- El programa reparte una carta a cada jugador. A partir de aquí, cada jugador realiza dos acciones:
    - En primer lugar, escoge cuantos puntos apuesta en esta jugada.
    - En segundo lugar, decide si quiere recibir más cartas del mazo. Si no quiere más cartas, se planta. Puede pedir tantas cartas del mazo como crea conveniente y se puede plantar cuando quiera.
- Cuando todos han acabado de escoger cartas, el jugador que tiene siete y medio gana el doble de puntos de lo que había apostado. En caso de que ningún jugador tenga siete y medio, el jugador que más se ha acercado a esta cantidad sin pasarse, gana 1 punto. El resto de jugadores perderán tantos puntos como hayan apostado.
- El jugador que pierde todos sus puntos, queda eliminado de la partida.
- El jugador ganador es el que más puntos ha obtenido después de todas las partidas jugadas.

Desarrollo del juego

Habrá un máximo de 8 jugadores

Se podrán jugar 2 modalidades.

Todos los jugadores humanos.

1 jugador humano contra jugadores máquina.

Se reparten cartas del mazo a los jugadores para saber cómo quedan numerados, el número 1 el jugador con la carta más alta y así sucesivamente.

En caso de igualdad de números, la prioridad es oros copas espadas bastos.

Se numeran los jugadores.

El número 1 posee la banca.

Cada jugador inicia la partida con 20 puntos.

Habrá un máximo de 30 manos.

Por turnos, el jugador con número más bajo primero, cada jugador:

- Realiza una apuesta dentro de un rango prefijado de 20% puntos restantes redondeado hacia arriba.
- Seguidamente debe decidir si desea recibir más cartas del mazo. Si no lo desea debe indicarlo diciendo que se planta. Si por el contrario, desea cartas para intentar acercarse lo más posible a sumar siete y medio, podrá pedir todas las que quiera de una en una pudiéndose plantar cuando quiera.

Juega la banca

Una vez hayan hecho las apuestas todos los jugadores, le llega el turno a la banca. Si quedara algún jugador que no se hubiese pasado de siete y medio, y por lo tanto está todavía en condiciones de poder ganar su apuesta, la banca procederá a su vez a jugar.

La banca no hace apuestas, simplemente recibe las de los jugadores, y juega como los demás jugadores, plantándose (si cree que así gana a todos o algunos de los jugadores que quedan) o dándose cartas, de una en una, pero con todas sus cartas boca arriba hasta decidir plantarse.

Desarrollo

La banca juega contra todos y cada uno de los jugadores, y por lo tanto si ella se ha pasado, deberá pagar a todos aquellos jugadores que se hubieran plantado.

Si la banca se ha plantado comprueba con cada jugador su jugada para ver a quién vence y con quién pierde. En cada apuesta vence quien más se acerque a siete y medio.

En caso de empate gana la banca; por lo tanto, si la banca tiene siete y medio gana automáticamente a todos los jugadores.

La banca debe pagar la cantidad apostada, a cada jugador con el que pierda, y a la inversa, cada jugador que pierda con la banca debe pagarle a ésta lo apostado.

Si un jugador tiene siete y medio (y la banca no) cobra el doble de lo apostado y además toma la banca en la mano siguiente.

El jugador que pierde sus puntos queda eliminado de la partida.

La partida continúa hasta que un único jugador se hace con todos los puntos en juego, quedando los demás eliminados, o bien hasta que se disputa el máximo de

manos fijadas para la partida, que puede ser 15 ó 30, en cuyo caso el vencedor es quien acumula mayor cantidad de puntos al final de la partida.

La cantidad de puntos a apostar en cada mano se escoge entre 4 posibilidades, con valores que se van incrementando paulatinamente a medida que se desarrolla el juego.

En las primeras manos las apuestas posibles oscilan entre 2 y 5, y suben hasta un rango entre 6 y 12 puntos cuando el número de manos ya se acerca al máximo de 15 ó 30.

Cuando el jugador que posee la banca es eliminado, con el número más bajo.

Lógica de las apuestas

Cuando un jugador decide recibir una carta, lo hará en función de las cartas que hay repartidas, y lo hará calculando la probabilidad de pasarse si recibiese una nueva carta. Es decir, si un jugador tiene por ejemplo un cinco, calculará todas las posibles cartas que supongan pasarse de 7 y medio así como el número total de cartas que quedan por salir, la división entre el número de cartas que supongan pasarse de 7 y medio y el número de cartas que quedan por salir multiplicado por 100 nos da la probabilidad de pasarnos de 7 y medio. Si quedan en 10 cartas que supusiesen pasarnos de 7 y medio y un total de 20 cartas por salir, esta probabilidad sería (10/20)*100 = 50%.

Un jugador nunca arriesgará si esta probabilidad supera una probabilidad especificada más abajo.

Especificaciones funcionales

M02 Bases de datos

- La aplicación, tiene que ser capaz de mostrar distintos informes conectándose a la base de datos de Amazon Web Services.
- Estos informes han de ser mostrados por la pantalla del terminal y ser guardados en un fichero Xml con el formato correspondiente.
- Necesitamos sacar los siguientes informes:
    - Carta inicial más repetida por cada usuario.
    - Jugador que realiza la apuesta más alta por partida.
    - Jugador que realiza apuesta más baja por partida.

- Ratio por cada jugador en cada ronda en total, y además,la apuesta media . Ejemplo, el jugador en la ronda 2 gana una media de 70% de las manos, con una apuesta media de 2,5.
- Némesis de cada jugador. Ejemplo: Rafa pierde más partidas cuando juega con Leandro.
- Porcentaje de partidas ganadas Bots en general.
- Porcentaje de partidas ganadas según la carta inicial, tanto por jugador como en total de jugadores.
- Mostrar el porcentaje de partidas que ganan los jugadores en función del orden que tienen en la partida.
- Media de veces que pasa cada jugador por partida que ha jugado
- Mostrar los datos de los jugadores y el tiempo que han durado sus partidas ganadas cuya puntuación obtenida es mayor que la media puntos de las partidas ganadas totales.
- Cuantas partidas ganan los jugadores en partidas contra bot con una carta inicial de espadas o bastos.
- Cuántas rondas se gana cuando no se roba carta en ese turno.
- Mostrar el nombre del jugador que es capaz de ganar una partida robando cartas en la mitad o menos de las rondas.
- Cuantas rondas gana la banca por empate.
- Partida con la puntuación más alta de todos los jugadores, así como añadir una columna nueva en la que diga si ha ganado la partida o no.
- Calcular cuantas veces se pasa un jugador (humano) de 7 y medio por partida

M03 Programación

Se creará un lista de tuplas de cartas, a la que llamaremos mazo, en orden oros copas bastos espadas.

Cada tupla representará una carta y sus valores serán (valor real, prioridad, valor en el juego)

Se podría crear un diccionario, pero la lista facilita acceder al índice dado de la carta para repartirla, eliminarla si es repartida a algún jugador y añadirla a la lista de cartas del jugador.

mazo = [(valor real, prioridad, valor en el juego) , (valor real, prioridad, valor en el juego) ….]

Modo Juego Manual

Lista de jugadores:

Se irá pidiendo el nombre de los jugadores hasta un límite de 8 y se irán añadiendo a una lista inicial de jugadores.

Los nombres de los jugadores sólo contendrán letras y números, empezando siempre por una letra, y no contendrán espacios.

Se escogerá una carta aleatoriamente del mazo por cada jugador con la que se decidirá el orden de prioridad de los jugadores.

La carta más alta determinará el jugador con mayor prioridad, la segunda más alta el segundo jugador con mayor prioridad y así sucesivamente.

En caso de cartas con valor real igual, gana la carta con mayor prioridad.

El jugador con mayor prioridad pasará a ser la banca.

Se creará la lista “jugadores” donde se insertarán los jugadores por orden de prioridad.

Para cada jugador

Un diccionario con clave su nombre y

valor = lista de elementos:

1. Una lista de tuplas con las cartas que tiene el jugador, donde cada tupla representa una carta.
   El elemento 0 será la primera carta.
   Todas las cartas se imprimirán en cada tirada para que podamos ver las cartas que han salido a los distintos jugadores.
2. El estado del jugador en la mano actual, por defecto al principio de cada mano, será “jugando”.
3. Estado del jugador en la partida, “jugando” si tiene puntos, “eliminado” si no tiene puntos
4. Prioridad del jugador, 0 la banca, 1 primer jugador, ….
5. Puntos acumulados en la mano, lo que suman sus cartas, si su estado en la mano no es eliminado será menor o igual a 7,5, en caso contrario, su estado en la mano será “eliminado”
6. Puntos apostados en la mano actual.
7. Puntos que le restan al jugador, nunca podrán ser negativos, si es cero y pierde dicha mano, su “estado partida” pasará a ser “eliminado”.
8. Contador mano actual, empieza valiendo 1

{ "nombre Jugador" : [[(valor real, prioridad, valor en el juego) , (valor real, prioridad, valor en el juego) .....] ,

estado mano->jugando, plantado, eliminado

estado partida --> jugando, eliminado

prioridad del jugador -> 0 banca, 1 primer jugador, 2 segundo jugador ....

puntos mano --> lo que suman sus cartas, si no esta eliminado, será menor o igual a 7,5,

puntos apostados --> de los puntos iniciales, cuántos puntos ha apostado el jugador,

puntos restantes --> de los puntos iniciales, cuántos le quedan al jugador

mano → contador de mano en la partida

}

Bucle nueva mano(0).

Empieza el juego.

Se reparten a cada uno de los jugadores 1 carta de la lista de cartas, se elimina dicha carta del mazo y se añade la carta (tupla) al primer elemento de la lista de tuplas del diccionario con clave el nombre del jugador.

Para cada jugador se actualizan los "puntos mano" con los puntos que representa en el juego dicha carta.

Empieza el jugador 1 (el cero es la banca y por tanto lo dejamos para el último turno)

Bucle mano por jugador(1)

Si el "estado partida" es jugando y su estado en mano también:

Se actualiza el contador mano.

Se imprimen todas las cartas que han salido.

También se imprime los puntos que tiene cada jugador, incluida la banca así como los puntos apostados.

El último jugador siempre tiene más información respecto a las apuestas de los demás jugadores.

Realiza una apuesta y por tanto se actualiza en el diccionario del jugador los puntos apostados así como los puntos restantes, no pudiendo ser los puntos restantes negativos.

Si el estado del jugador en la partida o el estado en la mano es “eliminado” , se pasa al siguiente jugador y volvemos al punto *Bucle mano por jugador(1)*

Sub_Bucle mano por jugador(2).

Si decide plantarse, se actualiza el "estado mano" de jugando a plantado y se pasará al siguiente jugador por orden de prioridad y volveremos al punto *Bucle mano por jugador(1)*.

Si decide seguir jugando, se le entrega otra carta aleatoria del mazo, se actualiza la lista de cartas del jugador con dicha carta, se elimina la carta del mazo,

Se actualizan los "puntos jugada" del jugador.

Si los "puntos jugada" superan el 7,5 y no está jugando la banca automáticamente se actualiza el "estado mano" a “eliminado” actualizamos los puntos restantes de la banca con los puntos apostados por dicho jugador. En caso que dicho jugador ya no disponga de puntos, actualizamos también su estado en la partida a “eliminado”. Y pasamos al siguiente jugador, volviendo al punto *“Bucle mano por jugador(1)”*.

Hay que considerar que en el caso que un jugador esté eliminado en la mano, cuando se imprima las cartas que tiene, se mostrarán todas sus cartas incluida la primera.

En caso que sus puntos no superen 7,5 volvemos al punto “*Sub_Bucle mano por jugador(2)*.

Una vez el jugador se planta o es eliminado de la mano, se pasa al punto *Bucle mano por jugador(1)* con el siguiente jugador por orden de prioridad mientras no sea la banca.

Fin “Bucle mano por jugador(1)

Fin “Sub_Bucle mano por jugador(2)

Si queda algún jugador en estado “plantado” le toca el turno a la banca. Como la banca no realiza apuestas, actualizamos su contador mano y pasamos directamente al punto *“Sub_Bucle mano por jugador(2)* con la banca.

Una vez ha jugado la banca, se realiza el reparto de puntos.

Si la banca se ha pasado:

Deberá pagar las cantidades correspondientes a cada uno de los jugadores según lo apostado, actualizando los puntos apostados de los jugadores a 0 y los puntos restantes = puntos restantes + 2 *puntos apostados(lo que habían apostado + lo que paga la banca).

Si no tiene puntos suficientes para pagar a todos los jugadores, pagará por orden de prioridad de los jugadores, a cada uno de ellos y el último en cobrar se quedará con la cantidad sobrante de puntos. Es decir, si el último jugador al cual puede pagar la banca había apostado 3 puntos y a la banca sólo le quedan 2, cobrará estos 2.

Si la banca se ha plantado:

Si tiene 7,5 gana a todos los jugadores, suma a sus puntos restantes la cantidad apostada por cada uno de los jugadores, actualiza los puntos apostados de éstos a 0.

Si no tiene 7,5, se compara los puntos acumulados en la mano con cada uno de los jugadores por orden de prioridad, en caso de empate gana la banca.

Si gana la banca, se actualizan los puntos restantes de ésta añadiendo los puntos apostados del jugador en proceso y actualizando los puntos apostados del jugador a 0. En caso que el jugador se quede con puntos restantes a 0, se actualiza su estado en la partida a “eliminado”

Si pierde la banca, se restan los puntos apostados por el jugador a la banca, se actualizan los puntos restantes del jugador a “puntos restantes” = . “puntos restantes” + 2*”puntos apostados por el jugador”.

Si gana el jugador y además éste tiene 7,5, se restan 2*”puntos apostados” a la banca y se actualizan los puntos restantes del jugador a “puntos restantes” = . “puntos restantes” + 3*”puntos apostados por el jugador” , éste pasa a ser la banca (actualizamos su “prioridad “ a cero) y la de la banca a la prioridad más alta, es decir, si en ese momento la prioridad más alta es 5, la banca pasa a tener prioridad 6.

En caso que la banca se quede a cero o no tenga suficiente para pagar la apuesta del jugador en proceso, pagará a éste con sus puntos restantes y el resto de jugadores se quedará sin cobrar, se actualiza el estado de la banca a “eliminado”, su prioridad a la prioridad más alta y el jugador con mayor prioridad que no está en estado del juego = “eliminado” pasa a ser la banca, es decir, actualizamos su prioridad a 0.

actualizaremos la lista “jugadores” de manera que la nueva banca pase al primer lugar de la lista, y el jugador que tenía la banca, ahora con la prioridad más alta, pasará al final de la lista.

Iniciaremos una nueva mano.

En este punto, todos los jugadores con estado en la partida = “eliminado” han de tener sus puntos restantes y puntos apostados actualizados a cero.

Si el primer jugador de la lista “jugadores” tiene una prioridad diferente de cero, éste se moverá al final de la lista, y el jugador con prioridad cero pasará al primer lugar de la lista.

Se volverá a crear la lista “mazo” con todas las cartas que había al inicio, se actualizará el estado de la mano de todos los jugadores cuyo estado en la partida no sea “eliminado” a “jugando”. Se reseteará la lista de cartas de cada jugador, puntos acumulados en la mano y puntos apostados.

Se comprueba que al menos quedan dos jugadores que no tengan su estado en la partida = “eliminado”

Si sólo queda un jugador con el estado de la partida = “eliminado” se acaba la partida y gana este jugador. En caso contrario volvemos al punto ***Bucle nueva mano(0).***

Repetiremos este proceso hasta llegar al tope de número de manos o bien hasta que sólo quede un jugador en la partida cuyo estado en la partida no sea “eliminado”

Modo Juego humano contra jugadores máquina, todo jugadores máquina.

La única variación en el modo en que algunos o todos los jugadores sean máquinas, será en la toma de decisiones a la hora de pedir carta y de apostar.

Jugadores boot piden carta

Si el jugador no es la banca:

Si tenemos menos puntos que la banca, siempre elegiremos carta.

Si tenemos más puntos que la banca, calcularemos la probabilidad de no pasarnos, calculando el total de cartas que hay en el mazo con las que no nos pasamos dividido entre (cartas del mazo):

Si la probabilidad de no pasarnos es mayor de un 65% pediremos carta con total seguridad.

Si está entre un 50 y un 65%, esta será la probabilidad con la que pediremos carta.

Si es menor de un 50%, dividiremos esta probabilidad entre 3, y como resultado nos dará la probabilidad con la que pediremos carta.

Si el jugador es la banca:

Pedirá carta mientras haya algún jugador que tenga más puntos que la banca.

Apostar:

Las apuestas serán más arriesgadas cuantos más puntos tengamos.

M04 Lenguajes de marcas

- Se les dará un fichero XML de configuración que se tendrá que leer una única vez al iniciar el programa

- Se les dará un fichero XML con la información de las cartas que se leerá la primera vez que se repartan cartas, a partir de ese momento se trabajará con las cartas en “memoria”
- Los informes (especificados en el apartado M02 Bases de datos) se presentarán en formato web siguiendo las siguientes especificaciones:
    - La página HTML no puede contener CSS incrustado
    - La página ha de tener un encabezado
    - La página ha de contener un índice en forma de anclas (enlaces dentro de la propia web) a los diferentes informes
    - Cada informe ha de presentarse con un título, un marco con bordes redondeados, una tabla con los datos, los títulos de la tabla han de tener un formato diferente a los datos y un pequeño texto explicativo sobre el informe. El marco no puede ocupar todo el ancho de la página y todo tiene que estar centrado excepto el título del informe
    - Cada informe ha de presentarse en un marco creado con Divs
    - Los informes han de tener un color de fondo (la página en sí NO puede tener color de fondo)
    - El posicionamiento de los elementos ha de realizarse con FlexBox

M05 Entorns de desenvolupament

- Control de versiones:
    - El proyecto debe estar gestionado desde el primer día en Github.
    - El proyecto tendrá un fichero “README.md” con la definición y las instrucciones para la instalación y utilización del proyecto. Así como también la información de contacto de sus autores (email, twitter, etc).
    - Cada alumno creará y trabajará en su propia rama de trabajo. La rama de trabajo llevará su nombre.
    - Cada alumno hará al menos 5 commits al proyecto de Github.
- Se realizará un diagrama de clases del proyecto.

Especificaciones no funcionales

M02 Bases de datos

- Especificar toma de requisitos del proyecto para saber qué datos son necesarios guardar en BD (para especificar Tablas y Atributos y cuáles no.
- Generar diagrama Chen de BD a partir de los requisitos tomados Draw.io.
- A partir del diagrama Chen, generar el modelo entidad-relación correspondiente con MYSQL-WORKBENCH.
- La Base de datos ha de contener todo lo necesario para su consistencia (PK , FK, tipo de los datos).

- Todas las tablas han de contener atributos de control (fechacreación, usuariocreación, fechamodificación, usuariomodificación)
- Almacenar los DML … así como los distintos modelos lógicos en una carpeta dentro del proyecto java.
- Algunos detalles del modelo:
    - Se tienen que guardar los detalles del juego como:
        - número mínimo de jugadores
        - número máximo de jugadores
        - apuesta minima
        - apuesta maxima
        - nombre de juego
        - descripción de las reglas de victoria
        - descripción de las reglas de derrota
    - La base de datos ha de ser capaz de almacenar los datos de todos los jugadores humanos.
        - usuario
        - password
        - email
    - Los participantes de las partidas pueden ser humanos o bots.
    - Para la partida se tienen que guardar los siguientes datos.
        - apuesta inicial
        - hora inicio y fin de partida
        - condiciones para ganar la partida
        - resultado de la partida(quien ha ganado)
        - el número de jugadores.
    - Se deben guardar todas las cartas de la baraja, con sus valores reales y los valores del juego, así como el tipo(palo de la baraja) que son.
    - Se deben guardar los turnos de cada partida con los datos de la partida en ese momento.
        - Puntuación del usuario en ese momento inicialmente
        - Puntuación del usuario al final del turno
        - acción que va a realizar el usuario
        - carta que le han dado inicialmente en ese turno
        - orden del usuario en cuestión…
    - Las acciones pueden ser pasar o robar, y ha de guardarse en el caso de que sea robar la carta o cartas que se roban
    - Las cartas forman parte de una baraja, ya que existen varios tipos de barajas, aunque a este juego se juega con la española.
    - Se tienen que poder guardar las puntuaciones de las partidas de todos los jugadores

M03 Programación

- El proyecto se ha de realizar en python.
- El código fuente estará comentado.
- La entrega del proyecto se realizará en la correspondiente tarea del Moodle.

M04 Lenguajes de marcas

- El fichero XML de configuración ha de tener el siguiente formato:
    - Tag Raíz: Inital_Config
    - Tag Clave de configuración: “nombre de la clave”
- El fichero de configuración ha contener las siguientes claves:
    - Num_Min_Players
    - Num_Max_Players
    - Num_Max_Rounds
    - Num_Initial_Points
    - Is_Allowed_Auto_Mode
- El fichero de configuración se ha de llamar: Basic_Config_Game.xml
- El fichero XML de cartas ha de tener el siguiente formato:
    - Tag Raíz: Cards
    - Tag Carta: Card (Cada card contendrá la siguiente información)
        - Tag Identificador: Id
        - Tag Valor: Value
        - Tag tipo (palo): Suit
        - Tag Carta activa: Is_Active

M05 Entorns de desenvolupament

- La rama principal debe llamarse “main”.
- Existirá una rama llamada “pre-producción” en donde se harán los “merge” necesarios antes de subir cambios a la rama principal.
- La entrega final del proyecto, a parte de entregarse en los cursos del moodle de otros módulos, también se hará en GitHub a través de una “realese”. La llevará a cabo el propietario de la repo.
- Se entregará la URL de la “realese” y el Diagrama de clases en una tarea habilitada para tal fin en el curso del Moodle.

Definición de “acabado”

M2: El diseño de la Base de datos tiene que tener todos los estándares vistos en clase, como evitar la redundancia ... pks ... fk ... La base de datos ha de ser funcional en el caso de que quisiéramos utilizarla en nuestra aplicación. Las consultas tienen que funcionar.

M3: El programa funciona completamente en alguno de los modos propuestos.

Gestión y fechas del proyecto

30 Nov a 15 dic incluido (16 a 18 present.)

Normativa y criterios de evaluación/corrección

- **M2:**
  - Diseño de base de datos (3,5 diagrama de Chen cumpliendo todos los estándares + 1,5 Modelo relacional cumpliendo todos los estándares coherente con el diagrama anterior)
  - Consultas a realizar (5 si todas van bien, si fallan menos de la mitad, 1, si fallan la mitad o más: 0)
  - Todo lo anterior suma 10 puntos como máximo.
- **M3:**
  - Si el proyecto está totalmente completado y funciona correctamente, se obtendrá 10 puntos que conformarán el 30% de proyecto de la nota global.
  - Si se ha conseguido que el proyecto funcione completamente bien en alguno de los 2 modos (todo jugadores humanos, jugador humano vs jugadores máquina), se obtendrá 5 puntos.
  - Si no se cumple ninguno de los casos anteriores, se obtendrá 0 puntos.
  - La puntuación del proyecto será: 0 puntos, 5 puntos ó 10 puntos.
- **M5:**
  - La correcta utilización de la herramienta GitHub.
  - El diseño y aspecto general del Diagrama de clases.
  - La facilidad de lectura del Diagrama de clases.
  - La puntuación del proyecto estará compuesta por 30% el control de versiones y 70% el Diagrama de clases.