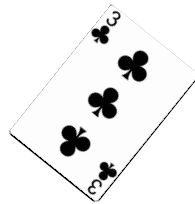

Herramientas Informáticas para Juegos de Azar

Práctica



The new PokerStove

Profesor de teoría: Manuel Núñez
Profesor de prácticas: Alberto Núñez

1.- Objetivo

Esta práctica consiste en desarrollar una versión mejorada de la aplicación **PokerStove**. El diseño de la GUI es libre y queda a responsabilidad del alumno. La única condición impuesta es que el diseño propuesto pueda reproducir las funcionalidades que da el programa original.

Existe un programa similar (y gratuito) llamado *equilab* que puede ser utilizado como inspiración para añadir funcionalidades que no cubre PokerStove.

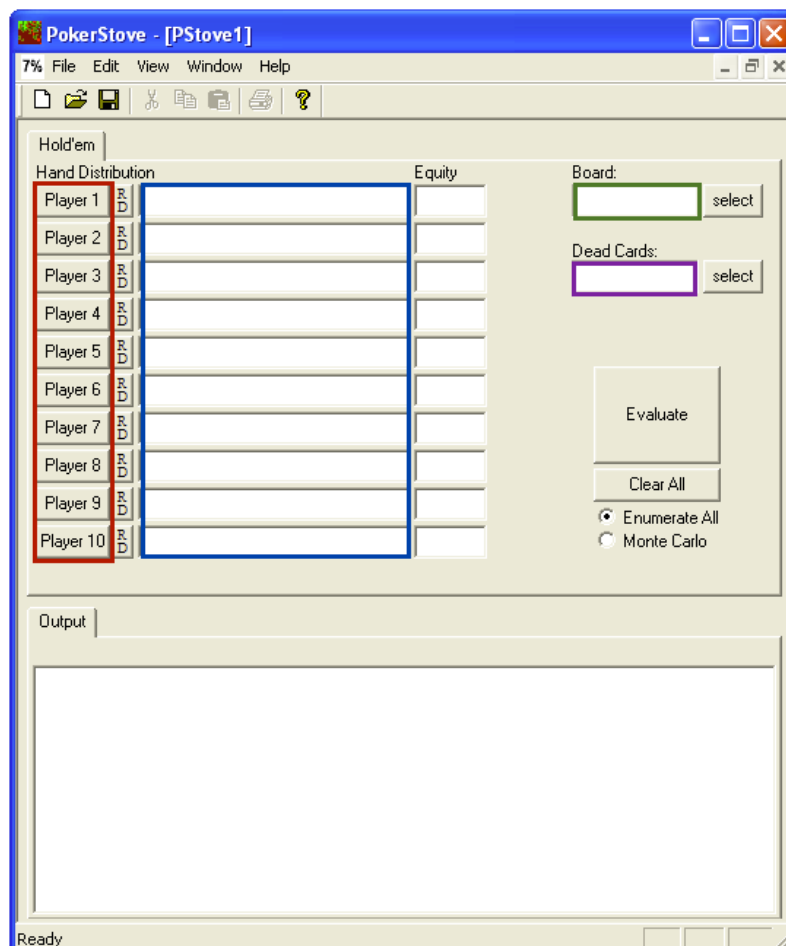
2.- Descripción

Esta práctica está formada por 1 apartado obligatorio y 1 opcional, de tal forma que el 70% de la calificación corresponde a la parte obligatoria y el 30% restante a la parte opcional. Para poder realizar la parte opcional **es necesario** haber realizado la parte obligatoria. No se tendrán en cuenta las partes opcionales realizadas cuando la parte obligatoria no esté completamente implementada.

La sintaxis para representar cada carta será la misma que se utilizó en la práctica 1.

2.1 Diseño y funcionalidad básica (obligatorio)

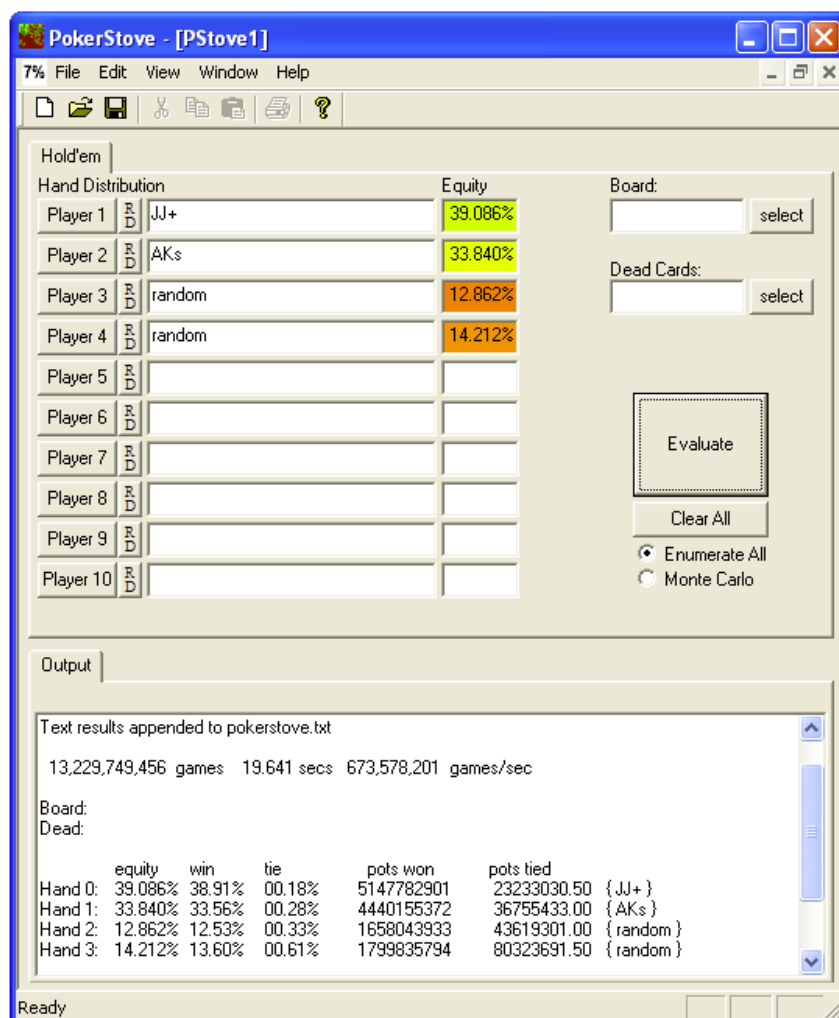
La herramienta PokerStove permite calcular (entre otras cosas) la probabilidad de ganar una mano hasta con un máximo de 10 jugadores. La siguiente figura muestra la GUI de PokerStove. En ella se puede apreciar cómo se representa cada jugador (cuadrado rojo). En cada mano se puede indicar qué cartas, o rango, posee o se estima que tenga cada jugador (cuadrado azul). Además, es posible indicar las cartas que existen en el *board*, de forma que no sólo se pueden simular situaciones pre-flop, sino *flop*, *turn* y *river* (cuadrado verde). Finalmente, se pueden descartar cartas que no entrarán en el cálculo (cuadrado violeta).



La siguiente imagen muestra un ejemplo de ejecución donde intervienen 4 jugadores. El jugador 1 juega con el rango **JJ+**, el jugador 2 con **AKs** y los jugadores 3 y 4 con **manos aleatorias**. Tras ejecutar la aplicación durante aproximadamente 20 segundos, detenemos la ejecución para observar el resultado.

El número de manos simuladas es de 13.229.749.456. Es decir, que se han simulado más de 600 millones de manos por segundo. Los resultados muestran que el jugador con más probabilidad de ganar (*equity*) es el jugador 1 con un 39.086%, seguido del jugador 2 con un 33.84%.

Es importante remarcar que entre los resultados (ver cuadro de texto inferior) también se muestran datos como los valores totales y porcentajes de manos empatadas y ganadas.



El usuario podrá definir los rangos, tanto de *hero* como de villano(s), textualmente, gráficamente, o dando un % de las mejores manos de un cierto ranking de manos. Por defecto, la herramienta incluirá al menos un ranking de manos iniciales que se tomará como base para calcular rangos que incluyan el *n*% de mejores manos iniciales, por ejemplo Sklansky-Chubukov, pero se debe permitir la inclusión de otros rankings.

Como novedad, la herramienta debe permitir definir fácilmente *rangos polarizados*. La tupla (w,x,y,z) , con $0 \leq w,x,y,z \leq 100$ y $w+x+y+z \leq 100$, denota que el rango lo conforman las manos situadas en los percentiles $(w,w+x]$ y $(w+x+y, w+x+y+z]$ del ranking de manos elegido.

Para cada par de combinaciones se van a ejecutar $2 \cdot 10^6$ manos simuladas. Por ejemplo, si se quiere evaluar a dos jugadores, tal que el jugador 1 esté jugando A6s o ATo y el jugador 2 esté jugando 77, entonces se ejecutarán en total $(4 + 12) \cdot 6 \cdot (2 \cdot 10^6)$ manos, dado que hay 4 combos para A6s, 12 combos para ATo y 6 combos para 77.

Por tanto, se debe tener en cuenta que NO todas las combinaciones tienen el mismo peso en un rango. Por ejemplo, para computar 7h7d vs KK+, AKs, haremos n simulaciones donde en un total de $\frac{6}{16} \cdot n$ ejecutaremos 7h7d vs AA, ya que la pareja de ases se puede formar de 6 formas distintas. De forma similar, en $\frac{6}{16} \cdot n$ simulaciones ejecutaremos 7h7d vs KK y en $\frac{4}{16} \cdot n$ ejecutaremos 7h7d vs AKs.

Se debe evitar repetir cartas. Por ejemplo, si nos piden enfrentar AKs con KK, lo más sencillo es fijar (aleatoriamente) el palo de AKs, fijar (aleatoriamente) el palo de los dos reyes (distintos del anterior) y quitar esas cuatro cartas de la baraja. Otra opción es considerar todas las formas de generar AKs y KK, evitando repeticiones de cartas, y enfrentarlas entre ellas.

Se debe tener especial consideración con el empleo de algoritmos y estructuras de datos que permitan obtener un rendimiento aceptable. Con esta práctica no se pretende superar en rendimiento a PokerStove, pero sí alcanzar un rendimiento similar. Por ejemplo, no sería aceptable obtener un tiempo de ejecución de minutos para la simulación mostrada en la segunda figura. Además, la GUI diseñada debe ser al menos tan *friendly* como la de PokerStove.

La documentación de la herramienta (incluida y estructurada en la ayuda de la propia herramienta) incluirá detalles sobre su uso, sobre la forma en la que se toman las manos de un rango (aleatoriamente pero teniendo en cuenta el peso de cada combo), sobre los rankings de manos incluidos por defecto, etc.

2.2 Introducir nuevos rangos y rankings (opcional)

Como parte opcional se propone la inclusión de nuevos rankings y rangos por parte del usuario final.

Para ello, cada ranking y rango podrá introducirse a través de la aplicación, quedando cada uno guardado en un fichero de texto en los directorios asignados para ello. De esta forma, al iniciar la aplicación, ésta leerá los directorios correspondientes, procesará los ficheros y finalmente cargará las estructuras de datos oportunas.

Para poder realizar esta funcionalidad, será necesario incluir también en la ayuda de la aplicación las instrucciones pertinentes. Entre otras cosas se deberá explicar detalladamente:

- Cómo introducir los rangos y rankings.
- Sintaxis utilizada.
- Un ejemplo.

Si se implementa este apartado, **no se podrán utilizar rangos o rankings introducidos en el código fuente** de la aplicación. Cada uno de ellos deberá ser procesado mediante un fichero externo.

2.3 Cálculo de equity en PLO (opcional)

Como parte opcional se propone la extensión de la herramienta para calcular la probabilidad que tiene de ganar una mano inicial de Omaha High (cuatro cartas) contra otra.

3.- Fecha de entrega

Los días **1 y 15 de Diciembre de 2015** la clase se impartirá en el **laboratorio** para resolver **dudas** para la realización de esta práctica. La práctica deberá entregarse antes del día **12 de Enero de 2016** antes de empezar la clase, impartida en el laboratorio, en la que será **evaluada**.

No se permitirá la entrega de prácticas fuera del plazo establecido.

4.- Modo de entrega

La práctica deberá entregarse a través del Campus Virtual, mediante el entregador habilitado para esta práctica.

No se tendrán en cuenta aquellas prácticas enviadas por otros medios que no sean el entregador habilitado, como por ejemplo, enviar la práctica como fichero adjunto en un e-mail.

Una vez finalizada la práctica, ésta se entrega mediante **un único fichero** comprimido con extensión “.zip”. Dicho fichero deberá contener:

- a) El fichero **alumnos.txt**. El **nombre y apellidos** de cada miembro del grupo que haya desarrollado esta práctica deberá estar presente en este fichero.
- b) Todos los ficheros que formen el proyecto (código fuente, directorios, ficheros de configuración, etc.) así como librerías externas o imágenes utilizadas.

NOTA: Se recomienda comprobar que, una vez generado el fichero comprimido “.zip”, el proyecto de la práctica puede abrirse correctamente. Esto se comprueba descomprimiendo el fichero y abriendo el proyecto correspondiente. En algunos casos, ficheros de configuración ocultos no se incluyen en el fichero comprimido, de forma que el proyecto no puede abrirse.