PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC1103- INTRODUCCIÓN A LA PROGRAMACIÓN

Tarea 2
Póquer Texas Hold'em

Entrega: 28 de octubre, 2015

1. Objetivo

En esta tarea practicarás los contenidos del curso en un desafiante juego de cartas, conocido como póquer Texas Hold'em. El objetivo es que programes el juego completo, por lo que deberás repartir cartas, llevar la cuenta de las apuestas de los jugadores y declarar el ganador. En esta tarea podrás practicar funciones, programación orientada a objetos, strings y listas, además de control de flujo. La tarea es relativamente larga y desafiante, por lo que es importante que la prepares con tiempo, para que aproveches al máximo la tarea como oportunidad de estudio y práctica.

Todas tus consultas sobre la tarea debes realizarlas a través del foro de tareas del curso, disponible en `http://bit.do/foro_tareas_IIC1103`. Este será el único medio para consultas sobre las tareas del curso, por lo que **no se aceptarán preguntas a través de otros medios**. Otras consultas debes enviarlas al e-mail de los coordinadores del curso `coordinacioniic1103@gmail.com`. La tarea es **individual**.

2. Instrucciones del juego

Tu tarea será programar una versión del juego Póquer Texas Hold'em. Tu versión del juego tendrá las siguientes características:

- Se juega sólo entre dos jugadores: un usuario del programa (jugador humano), y un robot muy bueno para el póquer (jugador robot). Cada jugador comienza con la misma cantidad de dinero (por ejemplo, $1000), que es un número que debes definir dentro de tu programa, no se le consulta al usuario.
- El juego usa baraja inglesa, que consta de cuatro *palos* o *pintas*, y cada uno consta de 13 cartas: nueve numerales (2 al 10), y cuatro literales (J, Q, K y A). Cada vez que comienza un juego se usa una baraja nueva. Para empezar el juego, se reparten dos cartas a cada jugador.
- El jugador humano comienza las apuestas, las que se acumulan en un pozo común (*bote*). Hay cuatro rondas de apuestas: la inicial, con cada jugador con dos cartas, y ninguna en la mesa; luego, se reparten tres cartas en la mesa, y se efectúa la segunda ronda de apuestas; después, se reparte la cuarta carta en

la mesa, y se efectúa la tercera ronda de apuestas: y finalmente, se reparte la quinta carta en la mesa, y se efectúa la ronda final de apuestas, para finalmente buscar al ganador según las cartas que tengan.

- Una *mano* corresponde a un juego desde que se reparten cartas hasta que se entrega el dinero apostado al ganador.
- En cada ronda de apuestas, el jugador humano comienza apostando. Al efectuarse una apuesta, el siguiente jugador tiene siempre tres opciones: retirarse, igualar la apuesta, o aumentar la apuesta. Si se retira, todo el dinero del *bote* se transpasa al otro jugador, y termina la mano actual. Si aumenta la apuesta, entonces corresponde al siguiente jugador las mismas opciones (retirarse, igualar o aumentar). Este proceso continúa hasta que un jugador se retira, o se iguala la apuesta.
- Al momento que un jugador iguala la apuesta, se reparten las cartas que corresponda en la mesa; si corresponde a la apuesta final, se comparan las jugadas de ambos jugadores, y al ganador se transfiere todo el *bote*. En caso de empate, se transfiere el *bote* en partes iguales a ambos jugadores.
- El juego se evalúa usando una combinación de cinco cartas (llamada *jugada*) entre las siete cartas disponibles (2 del jugador y 5 de la mesa). El orden de jugadasy mayor información puedes verlo en `http://www.poquer.com.es/ranking.html`. Puede pasar que la mejor jugada se haga con las dos cartas del jugador y tres de la mesa, con una de sus cartas y cuatro de la mesa, o incluso sin las cartas del jugador, usando las cinco de la mesa, Si ambos jugadores usan las cinco de la mesa, automáticamente es empate.
- Una vez entregado el dinero del bote, se debe preguntar al jugador humano si desea jugar otra mano, si es que su saldo es mayor que $10, o si desea abandonar el juego. Si su saldo es menor que $10, se acaba el juego inmediatamente.
- (**Opcional**): Puedes implementar tu tarea con *all-in* efectuado por el usuario, es decir, si el usuario no tiene el dinero suficiente para igualar la apuesta, o si desea apostar todo su dinero, automáticamente se muestran todas las cartas (es decir, haciendo visibles las del jugador robot), y el jugador robot puede igualar o retirarse, pero no superar la apuesta). Luego, deben repartirse las cartas restantes en la mesa hasta completar 5 cartas, y se debe declarar al ganador. Si el usuario pierde todo su dinero, se debe acabar el juego.

3. Clases y funciones disponibles para tu programa

Para realizar la tarea, dispondrás de un módulo Python llamado `poker`, el que dispone de clases y funciones muy útiles para tu tarea. Debes copiar el archivo `poker.py` en la misma carpeta del archivo de tu programa. El archivo lo puedes obtener desde `https://puc.classter.net`, en la pestaña `Evaluaciones`, bajo el link `Tarea2`.

Las clases que dispones en el módulo son `Tablero`, `Carta` y `Baraja`, y las funciones `obtenerApuestaRobot` y `compararJugadas`, los que se explican a continuación en detalle. Para usar las clases y funciones que se describe, debes usar las siguientes instrucciones:

```
from poker import Tablero, Carta, Baraja
from poker import obtenerApuestaJugador, compararJugadas
```

3.1. Clase `Tablero`

La clase `Tablero` corresponde, como su nombre lo indica, al tablero o mesa donde se jugará Póquer, el cual es de tamaño 80 x 20, medido en espacios para caracteres. Lo usarás para desplegar la gráfica del juego. Una posición (x, y) en el tablero se establece en relación al origen en la esquina superior izquierda del tablero, donde x es la distancia horizontal al origen (de izquierda a derecha) e y la distancia vertical (de arriba a abajo). La clase `Tablero` dispone de los siguientes métodos:

- `__str__()`: sobrecarga de la función `str()`, que permite generar un string para imprimir en pantalla el tablero completo. Puedes usar, por ejemplo, directamente la función `print(table)`, asumiendo que `table` es el nombre de tu variable de clase `Tablero`; en este caso, la función `print` llamará a la función `str()`, y desplegará el tablero en pantalla.
- `dibujarCarta(carta, x, y)`: recibe una variable `carta` de clase `Carta`, y dibuja la carta con origen en `(x,y)`, donde `x` e `y` son números enteros.
- `dibujarBorde(x, y, ancho, alto)`: dibuja un borde con origen en `(x,y)`, de tamaño `ancho` × `alto`. Todos los argumentos de la función deben ser números enteros.
- `escribirMensaje(mensaje,x,y)`: recibe un string en la variable `mensaje`, y lo escribe en el tablero partiendo en la coordenada `(x,y)`, donde `x` e `y` son números enteros.
- `borrarZona(x,y,ancho,alto)`: borra un rectángulo de `ancho` × `alto`, con origen en `(x,y)`. Todos los argumentos de la función deben ser números enteros.
- `borrarTablero()`: borra todo el tablero.

Para mostrar correctamente el tablero, la ventana donde muestren su juego debe ser suficientemente grande.

3.2. Clase `Carta`

Representa a una carta dentro de la baraja. Los valores de las cartas pueden ser un número entre `2` y `10`, y las letras `J`, `Q`, `K` y `A`, en este mismo orden de menor a mayor puntaje; mientras que el palo puede ser Corazones (`C`), Picas (`P`), Tréboles (`T`) o Diamantes (`D`). Los métodos de la clase `Carta` son:

- `obtenerValor()`: retorna un string con el valor de la carta. Los números son también representados como un string.
- `obtenerPalo()`: retorna un string de valor `'C'`, `'P'`, `'T'` o `'D'`, según corresponda al palo de la carta.
- `obtenerDimensiones()`: retorna dos números enteros `w,h` con el tamaño en pantalla de la carta.
- `definirEstado(estado)`: define el estado de la carta, donde el argumento `estado` debe ser el string `'oculto'`, si no se quiere mostrar la carta, o `'visible'` en caso contrario.
- `obtenerEstado()`: obtiene el estado de la carta, el valor de retorno es `'oculto'`, o `'visible'`.

3.3. Clase `Baraja`

Representa la baraja que usarás en cada mano. Tiene sólo dos métodos:

- `barajar()`: mezcla las cartas. Debes llamar este método antes de comenzar cada mano.
- `obtenerCarta()`: retorna una instancia de la clase `Carta`, con estado `'visible'`. Puede ser llamada repetidas veces y siempre entregará una carta distinta, hasta que se acabe la baraja (52 cartas). Si no hay más cartas en la baraja, retorna `None`.

3.4. Función `obtenerApuestaRobot`

Para obtener las decisiones del jugador robot, debes ejecutar la función `obtenerApuestaRobot(cartasRobot, cartasMesa, dinero_robot, apuesta_robot, apuesta_a_superar)`, donde `cartasRobot` es una lista de dos instancias de clase `Carta`, correspondientes a las cartas del jugador robot, y `cartasMesa` es una lista de instancias de clase `Carta`, que corresponde a las cartas en la mesa (puede ser una lista vacía si aún no hay cartas en la mesa). Las variable real `dinero_robot` es el monto de dinero restante del jugador robot, la variable real `apuesta_robot` es la apuesta acumulada en la mano actual del jugador robot, y la variable real `apuesta_a_superar` es el valor de la apuesta a superar. La función retorna un número real, correspondiente al monto de la apuesta que desea hacer el robot; si el valor de retorno es 0, significa que el robot se retira; si es igual a `apuesta_a_superar`, el robot iguala la apuesta; mientras que si el valor de retorno es mayor que `apuesta_a_superar`, el robot aumenta la apuesta.

3.5. Función `compararJugadas`

La función para comparar las jugadas de ambos jugadores está implementada en la función `compararJugadas(cartasHumano, cartasRobot, cartasMesa)`, donde `cartasHumano` y `cartasRobot` son listas de dos instancias de clase `Carta`, correspondientes a las cartas del Jugador Humano y Robot, respectivamente, y `cartasMesa` es una lista de cinco instancias de clase `Carta`, correspondiente a las cartas de la mesa. Retorna el string `'empate'` si hay empate, `'gana_humano'` si gana el jugador humano (usuario), y `'gana_robot'` si gana el jugador robot.

4. Lo que debes programar

Tu programa debe ejecutar las siguientes tareas:

- Dar un mensaje de bienvenida, con un breve instructivo del juego. Debe mostrar un mensaje `"Presionar <ENTER>para continuar"` al final del instructivo.
- Repartir cartas a los jugadores.
- Mostrar en pantalla las cartas de los jugadores en el tablero. Las cartas del usuario deben estar visibles, mientras que las cartas del Jugador robot deben estar ocultas.

- Llevar el control de las apuestas en cada ronda de apuestas. En particular, no hay límite en el monto de las apuestas, pero el jugador no puede apostar más de su saldo disponible. Si un jugador no tiene dinero para pagar una apuesta, debe retirarse. Entre cada ronda de apuestas, tu programa debe ir agregando las cartas que corresponda a la mesa.
- Declarar al ganador y entregarles su dinero, o repartir el dinero en caso de empate.
- Debes usar los elementos gráficos de la clase Tablero para mostrar el juego en pantalla.
- Imprimir en pantalla, en forma clara, lo siguiente:
  - Comienzo del juego
  - Las cartas de la mesa y jugadores
  - El dinero de cada jugador
  - Las acciones del jugador robot
  - El ganador y el dinero ganado
- Las cartas del jugador robot sólo deben ser mostradas al final de cada *mano*.

Debes programar tu tarea usando programación orientada a objetos. Te sugerimos crear al menos dos clases, una que encapsule los atributos y métodos de un Jugador, y otra que controle el juego. Los atributos y métodos de cada clase que implementes son de libre creación, mientras cumplas con los objetivos de la tarea. A continuación se describen de manera superficial las dos clases sugeridas:

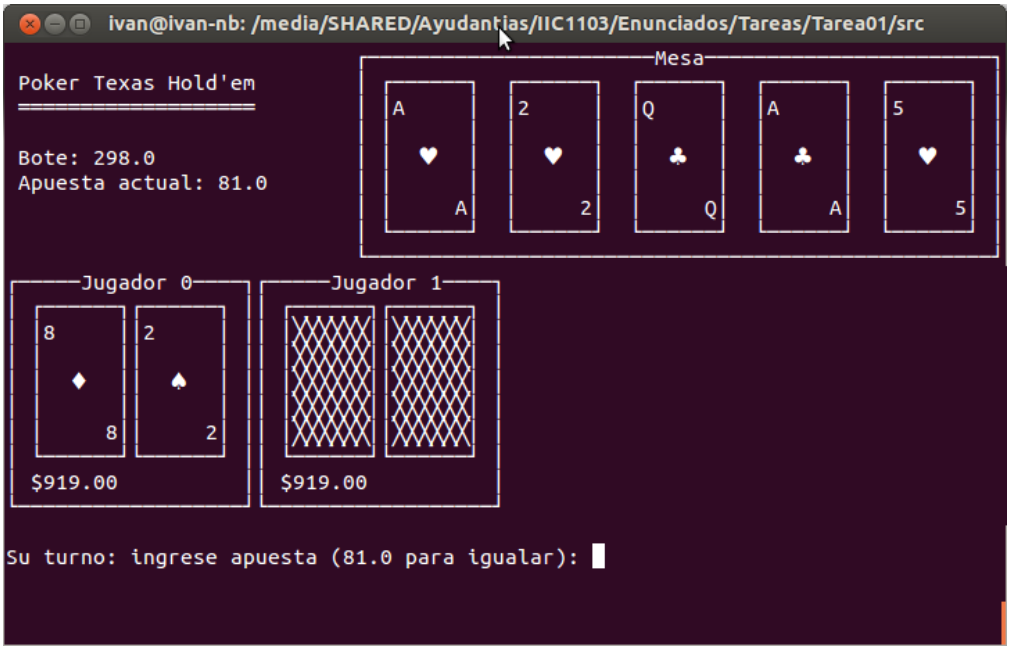
Clase `Jugador`

La clase `Jugador` que debes implementar debe ser capaz de almacenar las dos cartas para jugar, entregar y recibir dinero, saber si se trata de jugador humano o robot, debe ser capaz de entregar sus cartas como una lista de dos elementos.

Clase `ControlJuego`

Esta clase es la que debe controlar el juego, y mediante sus métodos se encarga de iniciar el juego, manejar los requerimientos de los jugadores, repartir las cartas, dibujar los elementos que corresponda en el tablero y mostrarlo, y llevar el control de las apuestas en cada mano.

La siguiente imagen muestra un ejemplo de tablero de juego, con cinco cartas repartidas en la mesa. En tu tarea, sólo se pide que sean dos jugadores. Nota que las cartas del Jugador 1 (robot) están ocultas.



5. Entrega

La entrega sebe realizarse mediante buzon electrónico, que estará disponible en la pagina del curso `https://puc.classter.net` hasta las 23:59 del 28 de octubre de 2015. Sólo debes incluir el archivo con tu tarea, sin `poker.py`.

5.1. Política de integridad académica

Tanto esta tarea, como el resto de las evaluaciones, te acoges a la *Política de Integridad Académica del Departamento de Ciencia de la Computación*, explícito en el Programa del curso. En particular, seremos muy enfáticos en buscar y sancionar los casos de copia, utilizando un software para detectar copias. Te recordamos que por ser alumno de la Escuela de Ingeniería, adhieres también al Código de Honor, publicado en `https://www.ing.puc.cl/nuestra-escuela/ingenieria-uc/codigo-de-honor/`.

5.2. Entornos de programación

El módulo `poker` puede ser usado sin problemas en IDLE de Python en Windows, OS X y Linux, en consola de Windows y terminal de OS X y Linux, y pycharm en Linux. En pycharm en Windows, deberás configurar la codificación de fuentes en UTF-8 y usar la fuente Courier New. En otros entornos, deberás probar si se visualizan correctamente las cartas, mensajes y marcos que definas.