



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
SEGUNDO SEMESTRE DE 2013

IIC 1103 ★ Introducción a la Programación

Proyecto 1

Objetivo General

- Estudiar y aplicar el uso de variables y expresiones aritméticas y booleanas.
- Estudiar y aplicar control de flujo.
- Aprender a utilizar funciones.

Introducción

El Video Poker es un juego de casino, en el cual se utiliza una máquina electrónica que permite apostar, muy similar al poker, pero de manera individual. La Figura 1 muestra un ejemplo de cómo está implementado el Video Poker en algunas máquinas de apuesta.

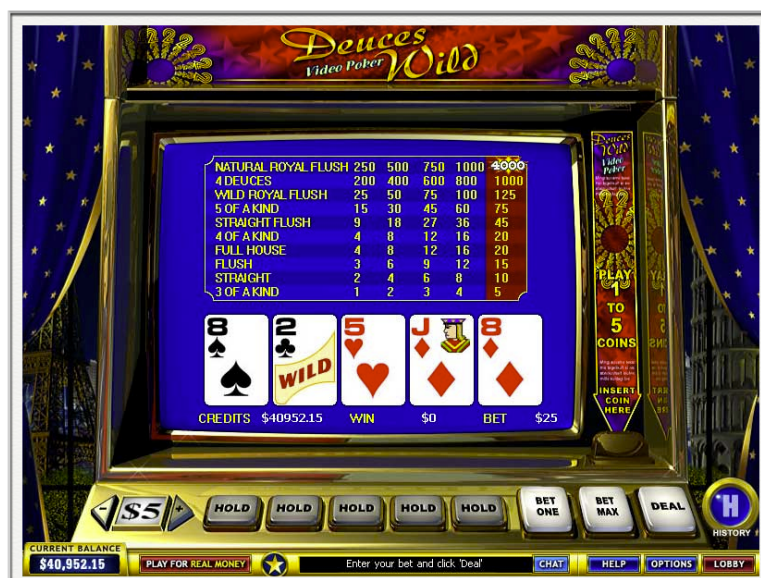


Figura 1: Implementación del juego Video Poker. Para jugar una variación de Video Poker llamada Jacks or Better visite <http://www.videopokercity.com/jacksorbetter.php>

Enunciado

El juego funciona por rondas. El jugador puede jugar tantas rondas como quiera mientras tenga crédito. El juego se acaba una vez que el jugador decide retirarse o se queda sin crédito. Cada ronda funciona de la siguiente forma:

1. El jugador elige cuántas fichas apostar (entre 1 y 5).
2. El computador baraja un mazo completo y entrega 5 cartas al jugador. El jugador puede elegir cuántas y cuáles cartas quiere cambiar. Para ello, debe seleccionar las cartas a cambiar y el computador las reemplaza por cartas nuevas del mismo mazo (no pueden salir las mismas de antes).
3. Luego de que el jugador tiene su mano final, el computador automáticamente revisa si hay alguna combinación de cartas ganadora (Tabla 1). Si no la hay, el jugador pierde las fichas apostadas. De lo contrario, el computador ofrece dos opciones:
 - a) Cobrar el premio correspondiente (Tabla 2)
 - b) La posibilidad de duplicar el premio correspondiente. La duplicación funciona de la siguiente manera:
 - El computador muestra 5 cartas nuevas, de las cuales la que está más a la izquierda está descubierta y las otras están cubiertas.
 - El usuario elige una de las cartas cubiertas. Si es mayor a la descubierta, duplica su premio. Si es igual, el premio queda igual. Si es menor, pierde lo ganado.
 - Si el jugador gana o empata, puede cobrar su premio o volver a duplicar cuantas veces quiera (mientras no pierda).

Combinaciones ganadoras

Existen 8 combinaciones de cartas que dan premios (Tabla 1). Note que los premios van de mayor a menor importancia, y que una mano de 5 cartas sólo tendrá un premio (Por ejemplo, si un usuario tiene un trío tendrá al mismo tiempo un par, pero el premio entregado será exclusivamente el correspondiente al trío, pues es el más alto).

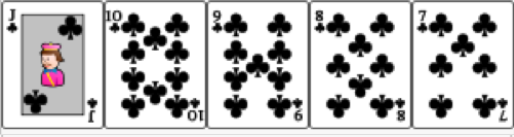
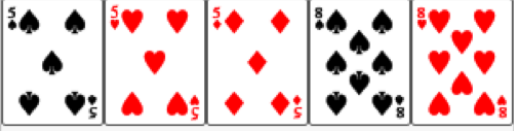
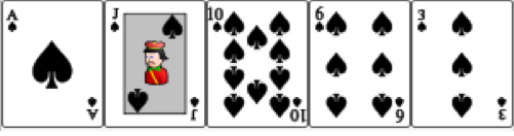

Nombre	Ejemplo	Descripción
Escala color		Cinco cartas en secuencia, de la misma pinta.
Poker		Cuatro cartas con el mismo número (la 5a puede ser cualquiera).
Full House		Tres cartas con el mismo número, y las otras dos cartas con el mismo número.
Color		Todas las cartas de la misma pinta, cualquier número.
Escala		Cinco cartas en secuencia, de cualquier pinta.
Trío		Tres cartas iguales, las otras dos son distintas.
Dos Pares		Dos pares de cartas iguales.
Un Par		Un par de cartas iguales.

Tabla 1: Combinaciones ganadoras en el Video Poker

# de fichas apostadas:	1	2	3	4	5
Escala Color	50	100	150	200	350
Poker	25	50	75	100	125
Full House	9	18	27	36	45
Color	6	12	18	24	30
Escala	4	8	12	16	20
Trío	3	6	9	12	15
Dos Pares	2	4	6	8	10
Un Par	1	2	3	4	5

Tabla 2: Premios según combinación de cartas y fichas apostadas

Librería

Junto con el enunciado de esta tarea se encuentran dos archivos llamados `pokerGUI.py` y `cartas.zip`.

Cartas.zip

Al descomprimir este archivo usted encontrará una carpeta llamada `cartas`, la cual contiene todas las imágenes que se necesitan para realizar esta tarea. Dentro de estas imágenes se encuentra `base.gif`, que es la contracara de una carta. Además, están las 52 cartas de un mazo. Los nombres de estas cartas son de la forma `[número][pinta].gif`, donde `[número]` es un número entre 1 y 13 que representa el valor de la carta (1 representa la carta as y 13 la carta rey), y `[pinta]` es la letra con que comienza la pinta de la carta. Por ejemplo la imagen que representa una reina de trébol tiene por nombre `12t.gif`. A continuación se muestran ejemplos de archivos de cartas con sus nombres.

`pokerGUI.py`

Este archivo le permitirá realizar su tarea utilizando componentes gráficas. Al iniciar una ejecución de su tarea se mostrará una interfaz que le permitirá interactuar con el usuario (Figura 2). Para esto, usted podrá utilizar ciertas instrucciones. A continuación se detalla cada una y cuál es su función.

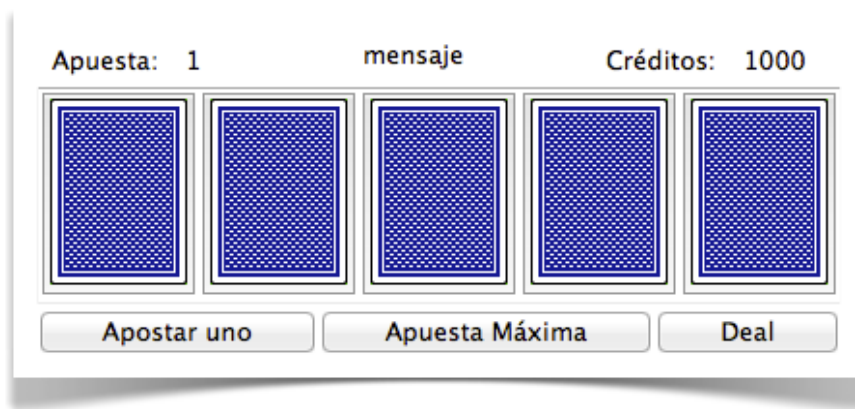


Figura 2: Interfaz para interactuar con el usuario

- `poker.preguntar(preg)`: muestra al usuario la pregunta *preg* y toma el valor `True` o `False` dependiendo de si el usuario responde si o no respectivamente.
- `poker.alerta(msg)`: muestra una ventana con el mensaje *msg*, y espera a que el usuario presione Ok para seguir ejecutando el programa.
- `poker.mensaje(msg)`: escribe el mensaje *msg* en la parte superior central de la interfaz.
- `poker.llenarMazo()`: devuelve al mazo todas las cartas que se han sacado.

- `poker.sacarCarta()`: toma como valor el número correspondiente a una carta del mazo que no se ha sacado. Una carta es representada por un número entre 0 y 51. Usted debe encargarse de esta representación. Por ejemplo, las cartas del 0 al 12 podrían representar desde el as hasta el rey de corazón.
- `poker.cambiarApuesta(apuesta)`: cambia por *apuesta* la cantidad de fichas a apostar que se muestra en la esquina superior izquierda de la interfaz.
- `poker.cambiarCreditos(credit)`: cambia por *credit* la cantidad de créditos que se muestra en la esquina superior derecha de la interfaz.
- `poker.cambiarCarta(pos,carta)`: cambia por *carta* la carta en la posición *pos* de la interfaz ($0 \leq pos \leq 4$). El valor *carta* debe ser el nombre de la imagen correspondiente a la carta que se desea poner pero sin la extensión .gif. Por ejemplo para poner la carta 10 de diamantes en la tercera posición se debe llamar a la instrucción `poker.cambiarCarta(2, '10d')`.
- `poker.cambiarTextoEnCarta(pos,texto)`: pone el texto *texto* sobre la carta en la posición *pos* ($0 \leq pos \leq 4$). Esto sirve para informar al usuario qué cartas está reteniendo. A modo de ejemplo ver la Figura 4
- `poker.esperarClick()`: al llamar a esta instrucción el programa se detiene hasta que el usuario haga click en alguno de los botones de la interfaz. Luego, toma como valor el número del botón que se presionó. (Figura 3)
- `poker.terminar()`: cierra la interfaz de usuario. Esta instrucción se debe ejecutar una vez que el usuario haya perdido todo su crédito.

Importante

Si el usuario cierra la interfaz, la instrucción `esperarClick` tomará el valor `-1` independiente del botón que se presione. Si esto ocurre su programa debe terminar.

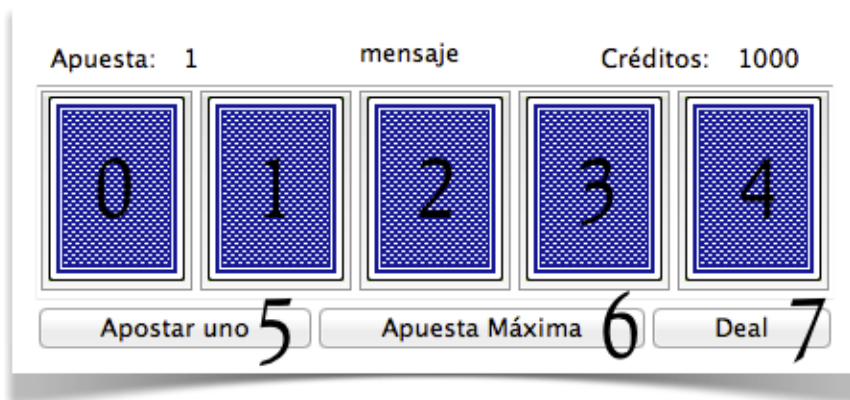


Figura 3: Número asociado a cada botón. La instrucción `poker.esperarClick()` tomará el valor del número asociado al botón que se presione.



Figura 4: A cada carta se le puede agregar un texto utilizando la instrucción `cambiarTextoEnCarta(pos,texto)`.

Configuración

Para que su tarea funcione correctamente, usted debe tener en una carpeta llamada `[número de alumno]_proyecto.1` que contenga lo siguiente:

1. Un archivo llamado `[número de alumno]_proyecto.1.py` que tenga su proyecto.
2. El archivo `pokerGUI.py`
3. Una carpeta llamada `cartas`, que contenga todas las imágenes de cartas que se encuentran en `cartas.zip`.

Además, el archivo `[número de alumno]_proyecto.1.py` debe tener la estructura que se muestra en la Figura 5

```
import pokerGUI

def tarea(poker):
    #INICIO TAREA

    #Código

    #FIN TAREA

app = pokerGUI.Application(None)
app.title('Video Poker')
app.loadProgram(tarea)
app.start()
```

Figura 5: Template para el archivo `[número de alumno]_proyecto.1.py`. En la página del curso podrán encontrar un archivo de ejemplo con este template.

Importante

Para que su proyecto funcione correctamente, en su computador debe estar instalado el paquete tkinter, que viene instalado por defecto con las distribuciones de Python. Si tienen problemas para ejecutar su proyecto pongan un aviso en el foro del curso.

Informe

Además de entregar el código de su archivo, usted deberá entregar un informe llamado `[numero_de_alumno]_informe_proyecto_1.pdf` en el cual explique en no más de una plana qué fue lo más difícil a la hora de desarrollar su proyecto y cómo lo resolvió. También podrá agregar a este informe una nueva plana con una sección que explique en qué aspectos usted hizo más de lo que se pide en este enunciado.

Entrega

Antes del día 17 de septiembre a las 23:59:00 usted debe entregar un archivo llamado `[numero_de_alumno]_proyecto_1.zip` en el sitio web del curso. Este archivo debe contener exclusivamente su archivo `[numero_de_alumno]_proyecto_1.py` y su informe en formato PDF. Las 10 tareas que funcionen perfectamente y tengan la menor cantidad de líneas de código (sin usar punto y coma, escribiendo cada instrucción en una línea) formarán parte del cuadro de honor del sitio del curso.