Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2233 — Programación Avanzada 2019-1

16 de Mayo de 2019
Actividad Formativa

Actividad 07

Interfaces Gráficas

Introducción

Después de comenzar a conocer las bellezas de las interfaces gráficas, has decidido renunciar a realizar otro programa por consola y ahora solo programas con **PyQt5**. Tus amigos se enteraron de eso y solicitaron a los profesores de Programación Avanzada que les hagas un juego llamado Programadito. Este juego consta de dos jugadores, cada uno comienza con un mazo de 52 cartas en orden aleatorio (el mazo inglés: cuatro pintas con trece cartas, sin incluir cartas *joker*) y se juega por turnos. La mecánica de este juego es similar a la del conocido juego de cartas **El Nervioso**, aunque con ligeros cambios para ser original. En este **enlace** puedes revisar un video hogareño donde se explica el juego original.

Modo de juego

El objetivo de este juego consiste en que dos jugadores: el Jugador 1 y el Jugador 2. Cada uno con un mazo de cartas completo, independiente del otro y en desorden. Los jugadores compiten en quien agota primero sus 52 cartas. La mecánica de este juego es la siguiente:

1. El juego comienza con un pozo con únicamente la carta *joker* sobre la mesa, esta es la carta inicial y no le pertenece a ninguno de los jugadores.

Figura 1: Carta *joker*

2. El Jugador 1 saca la primera carta de su mazo y empieza a contar. Primero, diría "*A*". Luego, el Jugador 2 sacará la primera carta de su mazo y continua contando, ahora diría "2". Después el Jugador 1 sacará su siguiente carta y dirá "3"... Así, sucesivamente los jugadores se alternan e irán acumulando cartas en el pozo, cada una con la exclamación del conteo actual.

3. Cuando el número de la carta sacada **coincida** con la que exclama el jugador, entonces cada uno debe apretar una tecla específica. Pero debe hacerlo rápido, ya que el último que la apriete se lleva todas las cartas acumuladas en el pozo y las incorpora al final de su mazo.

4. El contador de las cartas (la carta que cada jugador exclama cuando saca una carta) siempre respeta el orden natural de las cartas: "A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K". Una vez que se llegue a la "K", se vuelve a empezar a contar desde "A".

5. El juego concluye cuando a alguno de los dos jugadores se le acaban las cartas, quien será declarado el ganador.

En esta actividad deberás crear una interfaz para el juego explicado anteriormente. **Se entregarán todas las imágenes necesarias para lograrlo**. A continuación, se detallan las distintas ventanas y el flujo de como se interactúa con el programa.

1. **Menú inicial**

   a) Debes crear un menú inicial en donde los jugadores puedan ingresar sus nombres. Este debe mostrarse antes de iniciar el juego y desaparecer al momento de comenzar a jugar.

   b) El juego no puede comenzar si el nombre de algún jugador está vacío o contiene caracteres no alfanuméricos[1].

   c) El juego no debe aceptar que ambos usuarios posean el mismo nombre.

   Puedes ver un ejemplo[2] del menú inicial en la figura 2.

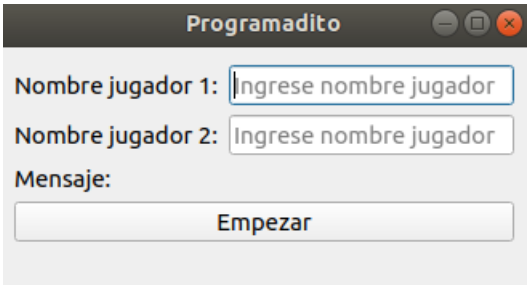

Figura 2: Ejemplo de menú inicial

2. **Inicio del juego**

   a) El programa debe poseer un botón de **"Empezar"** y otro de **"Reiniciar"**.

   - El botón "Empezar" da inicio al juego, no se puede jugar ninguna carta si este no ha sido presionado.
   - El botón "Reiniciar" permite reiniciar el juego. Reparte 52 cartas, ordenadas aleatoriamente, a cada jugador, vacía el pozo y queda esperando a que se presione "Empezar".

   b) Mediante el uso de las imágenes entregadas, se debe visualizar el tablero de juego con las cartas jugadas. En este se debe mostrar en su centro la última carta jugada en el pozo y debe detallar quién es el Jugador 1 y quién es el Jugador 2 (en la **sección 3** se explica con más detalle este punto).

[1] Podría serte útil el método `isalnum()`
[2] No es necesario que se vea exactamente así, pero si que contenga lo necesario para lograr lo pedido

Figura 3: Ejemplo de menú del juego al inicio

c) Al comienzo del juego (al apretar "Empezar"), solo debe mostrarse la carta *joker* como el inicio del pozo. Debe mostrarse un contador con el total de cartas que tiene el mazo de cada jugador, inicialmente en 52.

3. **Turnos**

a) En cada turno, la carta jugada sobre el pozo se debe mostrar en el tablero, como se ve en la figura 4.

Figura 4: Ejemplo de primera jugada

b) El **Jugador 1** mediante la tecla `S` debe ser capaz de mostrar la **primera** carta de su mazo, mientras que el **Jugador 2** lo hace mediante la tecla `K`. No se debe aceptar ninguna otra tecla (excepto las del siguiente punto) ni mostrar la carta de un jugador sin que sea su turno, o ocurra lo del siguiente punto.

c) Una ronda termina cuando la carta en la mesa sea igual a la carta enunciada. El **Jugador 1** deberá apretar `W` y el **Jugador 2** la tecla `I`. El primero en oprimir la tecla gana esa ronda. Deberá mostrarse un mensaje de la forma que quieras que indique al jugador ganador de la ronda y se deben de agregar las cartas en el pozo a la baraja del perdedor.

d) Después de cada ronda el juego comenzará con el jugador perdedor de la ronda anterior. En caso de la primera ronda, comienza siempre el Jugador 1.

e) En cada turno, deberá mostrarse arriba de la carta entregada el valor correspondiente al turno: primero el "A", después "2", luego "3", etc. En caso de llegar a la carta "K", la cuenta deberá partir desde el "A" nuevamente.

f) En caso de que un jugador presione la tecla `W` o `I`, según corresponda, cuando no es el momento indicado (es decir, cuando la carta mostrada difiere de la carta que el turno pide), se deberá mostrar un mensaje en la interfaz que indique el jugador que se equivocó. Además, este jugador deberá llevarse todas las cartas del pozo. La cuenta del turno sigue de forma normal.

4. **Fin del juego**

   Ganará el jugador que agote las cartas de su mazo primero. Deberás indicar quién es el ganador de la forma que quieras.

Observaciones

- **No debes subir** las imágenes que se te entregaron, para evitar subirlas solo debes copiar el archivo `.gitignore`, que se subió junto al enunciado, en la carpeta de la actividad de tu repositorio en tu computador. Si de todas formas las subes, **no obtendrás** la bonificiación en décimas de esta actividad.
- **Todo mensaje o acción se debe realizar mediante la interfaz**, el usuario solo interactúa con la interfaz, no con la consola.
- Las ventanas que construyas **no** deben reproducir exactamente los ejemplos mostrados en este enunciado. Mientras contenga los elementos pedidos, se considera correcto.

Notas y *tips*

- Los *labels* proveen una forma de cargar imágenes en PyQt. Tienen un método llamado `setPixmap` que recibe de argumento un objeto de la clase `QPixmap` (del módulo `PyQt5.QtGui`).
- Las instancias de `QPixmap` pueden recibir como argumento la ruta de un archivo de imagen para mostrar esa imagen en la interfaz.
- Los `QPixmap` tienen un método llamado `scaled` que recibe dos argumentos: `x` e `y`. El método retorna un nuevo objeto `QPixmap` con dimensiones `x` e `y`.
- Para simplificar el crear múltiples ventanas, considera que puedes usar una única ventana que muestre y esconda sus distintas componentes (*labels*, botones e *inputs*) dependiendo del caso.
- Los métodos `hide()` y `show()` esconden y muestran un *widget*, respectivamente.
- El método `setEnabled` de un *widget* recibe un booleano (`True` o `False`) y dependiendo de su valor habilita o deshabilita el *widget*. Por ejemplo, `boton.setEnabled(False)` deshabilita a `boton`, lo cual no permite que pueda ser apretado por el usuario.
- Considera usar la función `listdir` de `os` para obtener todos los nombres de cartas posibles desde la carpeta `cartas`.
- La función `cycle` de `itertools` recibe una lista y retorna un objeto iterable. El aplicar la función `next` sobre su resultado, cada vez retorna un elemento distinto de la lista inicial, siguiendo el orden de la lista. La gracia, es que después de retornar el último elemento y volver a ejecutar `next`, se devuelve y retorna el primer elemento. Esto permite obtener elementos que provengan de una lista "ciclica".

Requerimientos

- Menú inicial (2.00pts)
  - Los jugadores pueden ingresar su nombre (0.5pts)
  - Se verifica restricciones en los nombres de los jugadores. (0.5pts)
  - Es la primera pantalla en aparecer y luego de ingresar desaparece de inmediato. (1.00pts)
- Tablero de juego (2.00pts)
  - Botones "Empieza" y "Reiniciar" funcionan correctamente (0.50pts)
  - Las imágenes de las cartas se muestran al centro y el número de la carta pedida encima de ella (0.50pts)
  - Muestra donde se encuentra el Jugador 1 y 2 (0.50pts)
  - Los jugadores pueden saber en todo momento cuantas cartas le quedan a sus mazos (0.50pts)
- Funcionamiento del juego (2.00pts)
  - Las teclas definidas, al presionarlas, hacen lo indicado. (0.50pts)
  - Se cambia la imagen de la carta mostrada en el tablero como corresponde (a medida que se presionan las teclas) (0.60pts)
  - Se cambia el número de la carta enunciada arriba de la imagen de la carta mostrada siguiendo el orden correcto de la secuencia (0.20pts)
  - Los Jugadores 1 y 2 se van alternando correctamente conforme van pasando las rondas (0.20pts)
  - Se reconoce correctamente quién es el ganador de cada ronda y quién es el ganador final, y lo hace en los momentos correctos del juego (0.50pts)

Entrega

- **Lugar:** En su repositorio privado de GitHub, en la **carpeta** `Actividades/AC07/`
- **Hora del *push*:** 16:30

Auto-evaluación

Como esta corresponde a una actividad formativa, te extendemos la instancia de responder, después de terminada la actividad, una auto-evaluación de tu desempeño. Esta se habilitará a las **16:50 de jueves 16 de mayo** y tendrás plazo para responderla hasta las **23:59 del día siguiente, viernes 17 de mayo**. Puedes acceder al formulario mediante el siguiente enlace:

`https://forms.gle/LDP57wkUcvVR3FdJA`

El asistir, realizar la actividad y responder la auto-evaluación otorgará como bonificación al alumno 2 décimas para sumar en su peor actividad sumativa del semestre.