

## PROGRAMACIÓN I - GRUPO B

### PRÁCTICA Nº4

30 de noviembre 2023 18:30-21:30

#### INSTRUCCIONES

---

1. Se debe acceder a la **SALA DE GRUPO** del campus virtual de vuestro equipo de trabajo.
2. Durante el desarrollo de la sesión práctica se debe seguir el procedimiento de trabajo en la sala asignada para realizar las actividades evaluables. Recordad que es **imprescindible GRABAR toda la sesión y COMPARTIR la pantalla completa**.
3. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de **Prácticas de Programación I antes de la hora de finalización de la sesión de prácticas**.
4. Se debe entregar **un único fichero en formato .txt o .cpp, sin comprimir**, con el nombre de **P4Ejercicio**.
5. El fichero entregado debe **incluir el nombre de los integrantes del equipo**.
6. Aunque las prácticas se realizan en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en su calificación**.
7. **El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su descalificación**.

Al final del enunciado de la práctica tenéis la puntuación de los diferentes hitos que debéis cumplir durante la sesión. Se recomienda que realicéis de forma previa al desarrollo una lectura comprensiva del enunciado, así como un debate en cada equipo con el que podáis planificar la estrategia a seguir durante la práctica.

Dicha estrategia deberá quedar plasmada en un flujograma que deberá ser entregado en papel al finalizar la práctica, indicando en el los nombres de los integrantes del equipo.

Recordad que podéis/debéis utilizar el chuleterio durante la sesión, y que debéis tener en cuenta la rúbrica general de evaluación. Por ser una práctica podéis consultarla durante la sesión sin problema.

**EJERCICIO PRÁCTICO [ 70% puntuación ] – 160 minutos**

Diseñar y escribir un programa en C++11, **P4Ejercicio**, que permita comenzar a jugar a una versión reducida del juego Candy Crush. El juego consistirá en rellenar de forma aleatoria una matriz con caracteres (cada uno de ellos representará un caramelo diferente). Dichos caramelos podrán ser movidos en cada turno del jugador siguiendo sus indicaciones.

Por ejemplo, si el tablero en un momento determinado del juego tuviera la siguiente composición

A	A	A	A
B	A	A	X
A	X	A	A
J	J	J	A

Y el jugador decidiera mover el caramelo de la posición (0,4) hacia abajo el tablero quedaría de la siguiente forma

A	A	A	X
B	A	A	A
A	X	A	A
J	J	J	A

La dirección de movimiento de las fichas será indicado mediante la utilización de los caracteres:

- 'a': movimiento hacia la izquierda
- 'd': movimiento hacia la derecha
- 's': movimiento hacia abajo
- 'w': movimiento hacia arriba

El programa que debéis implementar durante esta práctica tiene que cumplir los siguientes requisitos:

- Vuestro desarrollo debe incorporar el mecanismo que considereis oportuno para poder **representar variables de tipo Jugador** en la memoria del computador. Para este ejercicio un jugador consistirá en:
  - Un nombre completo.
  - Una variable puntuación del tipo que consideres oportuno que sea capaz de almacenar las puntuaciones obtenidas en las diferentes partidas que haya jugado durante la sesión de juego.
  - Una variable de tipo contenedor del tipo que consideres oportuno que pueda almacenar los caracteres que representarán los diferentes caramelos en el tablero que tendrá la etiqueta caramelos (mínimo 3, máximo 6 caramelos).
- En la **función principal del programa** debes definir una **variable contenedor** del tipo que consideres oportuno **que pueda representar el tablero de juego, dicho tablero tendrá un tamaño fijo de 6 filas por 5 columnas.**
- En la **función principal del programa** debes incorporar una **variable de tipo Jugador** de etiqueta **playerOne** donde poder almacenar la información de la persona que vaya a jugar a nuestra propia versión del juego mencionado.
- **Dinámica del juego en cada ronda:**
  - Al inicio del juego **el programa solicitará** al jugador la información necesaria para su identificación (nombre completo), así como los diferentes caracteres que quiera utilizar para la representación de los caramelos, almacenándolo en los campos correspondiente de la variable tipo **Jugador** declarada en la función principal del programa.
    - El jugador tendrá que seleccionar como mínimo 3 caracteres diferentes y como máximo 6.
    - No podrá repetir caracteres.
  - Posteriormente, el programa inicializará la variable de etiqueta tablero con el carácter '-'

- Para a continuación rellenar de manera aleatoria el tablero 6x5 con los caracteres almacenados en el campo **caramelos** del Jugador.  
Teniendo en cuenta siempre que cuando haya alguna casilla sin caramelo contendrá el carácter '-'.
- A continuación, el juego deberá implementar un bucle de control en el que mientras el jugador quiera continuar con una nueva partida se realizarán las siguientes acciones:
  - Mostrar el estado del tablero.
  - Mostrar un menú contextual en el que las opciones sean:
    1. Mostrar tablero
    2. Seleccionar caramelo a mover
    3. Abandonar ronda actual y empezar otra
      - En el caso de seleccionar la opción Mostrar tablero el programa mostrará por terminal el estado actual del tablero.
      - En el caso de seleccionar la opción Seleccionar caramelo a mover el programa preguntará por las coordenadas del caramelo elegido y a continuación por el movimiento que desea realizar el jugador (arriba, abajo, izquierda o derecha).
        - En caso de poder realizarse el movimiento se intercambiarán las posiciones del caramelo indicado y con el contenido de su casilla destino.
        - En caso de no poder realizarse dicho intercambio, bien por haberse indicado una casilla fuera del tamaño del tablero, o bien por haberse seleccionado un caramelo ubicado en uno de los bordes del tablero junto a un movimiento que alcanzaría una posición fuera de su rango, se mostrará un aviso por terminal y la posición de los caramelos quedarán igual.

- En el caso de seleccionar abandonar ronda actual y comenzar otra el programa borrará el tablero actual y procederá a rellenarlo de nuevo con otra combinación aleatoria de caramelos.
- Al final de cada ronda el programa debe preguntar si se desea seguir jugando o no:
  1. En caso afirmativo debe seleccionar de nuevo una palabra de manera aleatoria y repetir la secuencia de juego indicada anteriormente.
  2. En caso negativo el programa finalizará, mostrando previamente el resultado de la sesión de juego, y todas las palabras secretas que el usuario había introducido inicialmente.

### **RÚBRICA DE CALIFICACIÓN**

- El 30% de la nota de la práctica, se obtendrá sobre trabajo realizado durante la sesión (resolución de preguntas, dominio de la materia, etc.). Esta nota será individual, y se considerarán para su valoración aspectos como:
  - **Defensa individual** a cuestiones que plantee el profesor durante la sesión a cada integrante del equipo.
  - **Trabajo/debate en equipo** sobre el desarrollo a realizar en la práctica. Quedando representado en un flujograma en el que se visualice el concepto del programa a realizar. **Deberá desarrollarse al inicio de la práctica y se entregará en papel a la finalización de la misma.**
- El 70% restante será sobre la puntuación del trabajo entregado antes de la finalización de cada práctica (misma nota para cada integrante del equipo), siguiendo la siguiente rúbrica general para cada apartado puntuable:

	%máx. (*)
El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegido son adecuados.	100%

(\*) El porcentaje (% máx.) representa el valor máximo sobre el que se evalúa el elemento indicado.

Puntuación por apartados. El 70% de la calificación de la práctica será obtenida sobre la corrección de la práctica entregada siguiendo el siguiente reparto de puntos para cada apartado:

Escritura correcta del código con las tabulaciones adecuadas, nombres significativos para las variables, comentarios, etc. (Limpieza del trabajo desarrollado)	0,5 puntos
Definición del tipo de datos <b>Jugador y declaración variables tablero y playerOne</b>	0,25 puntos
<u>Dinámica del juego:</u>	
Petición e inicialización de la información que es capaz de almacenar la variable <b>playerOne</b> de tipo <b>Jugador</b> : <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Caracteres para representar los caramelos</li> <li>• Puntuaciones</li> </ul>	0,25 puntos 1,5 puntos 0,25 puntos
Inicialización de la información almacenada en el tablero	0,5 puntos
Bucle de control de la sesión de juego	0,25 puntos
Rellenar el tablero de juego de forma aleatoria con los caramelos indicados por el jugador (estarán almacenados el campo correspondiente de la variable <b>playerOne</b> de tipo <b>Jugador</b> )	2 puntos
Mostrar el tablero al jugador	1 punto
Gestión del menú	1 punto
Gestión movimiento de caramelo	2 puntos
Gestión del almacenamiento de puntuaciones de las diferentes rondas en el campo correspondiente de la variable <b>playerOne</b> de tipo <b>Jugador</b>	0,5 puntos