

PECL2 – Fundamentos de la Programación

Las 4 en raya es un juego de estrategia para dos jugadores que se desarrolla en un tablero vertical dividido en 6 filas y 7 columnas, donde cada jugador introduce fichas de un color distinto con el objetivo de alinear cuatro de ellas de forma consecutiva. La partida se desarrolla por turnos, y cada ficha colocada desciende automáticamente hasta ocupar la posición libre más baja de la columna seleccionada, lo que obliga a ambos jugadores a anticipar movimientos y bloquear posibles combinaciones del oponente.

Se pide desarrollar un programa que permita ejecutar este juego. Al iniciar el juego, el programa deberá por pantalla ofrecer dos opciones diferentes: **modo de juego** y **estadísticas**. Para seleccionar el usuario introducirá por teclado la opción correspondiente.

```
=== Juego de Cuatro en Raya ===
Seleccione una opción:
1. Iniciar nueva partida
2. Estadísticas
3. Salir
```

Modo de juego

En el modo de juego lo primero que se mostrará es un menú en el que el usuario podrá elegir entre disputar la partida contra otro jugador o enfrentarse a la máquina. Una vez seleccionado el modo de juego se pedirá el nombre de los jugadores.

```

Seleccione el modo de juego:
1. Jugador vs Jugador
2. Jugador vs IA

```

Una vez comenzada la partida se mostrará el estado del tablero, el nombre del jugador que tenga el turno de juego, el cual deberá indicar por teclado el número de la columna en la que desea insertar la ficha. En el caso de jugar contra la máquina esta elegirá una columna que le permita luchar por ganar el juego. En el caso de no tener ninguna opción válida elegirá una columna aleatoriamente donde incorporar su ficha.

[illegible]

Tras cada jugada se comprobará si hay un ganador y se terminará el juego y se guardará la puntuación obtenida para mostrarla en la sección de estadísticas.

```

  0   1   2   3   4   5   6
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   |   |   |   |
+---+---+---+---+---+---+
| O | O | O |   |   |   |
+---+---+---+---+---+---+
| X | X | X | X |   |   |
+---+---+---+---+---+---+
¡Felicidades Jose, has ganado la partida!

```

En cada partida, se deberán respetar las siguientes reglas:

- Una ficha solo puede estar en la fila inferior del tablero de juego o encima de otra ficha, de manera que, al introducir la jugada, solo se puede elegir la columna.
- Se puede realizar 4 en raya en vertical, horizontal o en diagonal.

The figure consists of three 7x7 grids, each with columns labeled 0 to 6. The grids illustrate the steps of a 2D convolution operation.

- Grid 1 (Left):** A 3x3 kernel (red box) is located at the bottom left, covering cells (4,0), (4,1), (4,2), (5,0), (5,1), (5,2), (6,0), (6,1), and (6,2). A 3x3 dashed box (receptive field) is located at the top right, covering cells (0,4), (0,5), (0,6), (1,4), (1,5), (1,6), (2,4), (2,5), and (2,6).
- Grid 2 (Middle):** The 3x3 kernel (red box) is moved one step up and right, now covering cells (3,0), (3,1), (3,2), (4,0), (4,1), (4,2), (5,0), (5,1), and (5,2). The 3x3 dashed box (receptive field) is also shifted one step up and right, now covering cells (0,3), (0,4), (0,5), (1,3), (1,4), (1,5), (2,3), (2,4), and (2,5).
- Grid 3 (Right):** The 3x3 kernel (red box) is moved another step up and right, now covering cells (2,0), (2,1), (2,2), (3,0), (3,1), (3,2), (4,0), (4,1), and (4,2). The 3x3 dashed box (receptive field) is shifted again, now covering cells (0,2), (0,3), (0,4), (1,2), (1,3), (1,4), (2,2), (2,3), and (2,4). The output value 10 is shown in the bottom right cell of the grid, at (6,6).

- El final de partida podrá darse si no quedan huecos en el tablero, en cuyo caso no hay ganador, o que se termine la partida porque alguno de los jugadores ha obtenido 4 en raya. Tanto si se gana, como si se pierde o se empata, esto se reflejará en las estadísticas de los jugadores y dichos resultados se guardarán para tener un histórico de las partidas jugadas.
- Se deberá mostrar gráficamente en todo momento por terminal el tablero de juego, mostrando además cada acción de los distintos jugadores y se deberá indicar en cada momento el número de la ronda en la que se encuentran.
- Tras cada partida, cada el jugador obtendrá una puntuación que afectará de manera diferente en sus estadísticas.
 - Si el jugador ha ganado:
 - Aumentará en 1 su contador de victorias.

- Obtendrá una puntuación por partida inversamente proporcional al número de rondas que haya tardado en conseguir la victoria siendo la mínima puntuación de 500 puntos y la máxima de 3.500 reduciéndose en 500 puntos por cada ronda.
- Los puntos obtenidos en la partida se sumarán al histórico del jugador.
- Si el jugador ha empatado:
 - Aumentará su contador de partidas empatadas.
 - Se obtienen 250 puntos por la partida que se acumularán a la puntuación histórica del jugador.
- Si el jugador ha perdido:
 - Aumentará su contador de derrotas.
 - Obtiene 0 puntos por su participación.

Al finalizar cada partida, se mostrarán los siguientes resultados:

- El nombre de los jugadores.
- Ganador de la partida y puntuación obtenida de cada jugador.
- Número de rondas jugadas (cada 2 jugadas, una por cada jugador, es una nueva ronda).

Estos datos se guardarán para tener un histórico de las partidas y, en el caso de que un jugador haya jugado varias partidas antes, el programa deberá poder acumular los resultados obtenidos de las distintas partidas que ha jugado.

Modo estadísticas

El apartado de estadísticas registrará y mostrará información relevante sobre las partidas realizadas:

1. Permitirá registrar los resultados de forma manual de una partida.
2. Se contará con un ranking del Top 10 basado la puntuación de los jugadores.
3. Se podrán eliminar los datos registrados de un jugador. Esto afectará a los resultados disponibles en los rankings.
4. Se contará con un histórico de los resultados de las distintas partidas. Cada una se encontrará identificado por una marca de tiempo (fecha y hora), los nombres de los jugadores, ganador y puntuación obtenida por cada uno. En caso de las partidas con un solo jugador se indicará que se ha jugado contra el ordenador.
5. Se podrán consultar el histórico de datos de un jugador: número de victorias, número de derrotas, número de empates, número de partidas jugadas y puntuación total obtenida.
6. Se podrán importar los datos de las estadísticas de los jugadores ([ranking, histórico de partidas](#)) desde un fichero .txt.
7. Se podrán exportar los datos de las estadísticas de los jugadores ([ranking, histórico de partidas](#)) a un fichero .txt.

Condiciones de entrega

1. La práctica deberá ser entregada en de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos archivos: la memoria de la práctica en formato **PDF** y el código **ZIP** (no utilizar extensión .rar). No se aceptarán trabajos enviados pasada la fecha límite de entrega.
 - a. El ZIP con el contenido de la práctica deberá seguir la siguiente nomenclatura **PECL2_apellidosalumno1_apellidosalumno2.zip**
 - b. El PDF con la memoria de la práctica deberá seguir la siguiente nomenclatura **PECL2_apellidosalumno1_apellidosalumno2.pdf**
2. *No se aceptará ninguna práctica que no sea entregada por Blackboard*, no se aceptarán entregas por correo.
3. La práctica puede ser realizada individualmente o en pareja, siempre que los alumnos **pertenezcan al mismo grupo de laboratorio**. Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.
4. **La memoria deberá incluir, como anexo, el código fuente del programa.** *Si esto no fuera así, la práctica no podrá ser aprobada.*
5. La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.
6. Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo con las especificaciones indicadas en los enunciados.
7. Para aprobar, se debe desarrollar la solución haciendo uso de buenas prácticas de programación. Por ejemplo, el código debe estar debidamente comentado y contener comentarios aclarando lo que se presenta, nombre de variables y subprogramas significativos, evitar el uso de variables globales.
8. En la portada de la memoria deberán figurar los datos siguientes:

Grado en Matemáticas y Computación
Curso 20xx/20xx – Convocatoria Ordinaria
DNI – Apellidos, Nombre
9. La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel; 2) diseño general del sistema y de las estructuras de datos utilizadas; 3) subprogramas principales que intervienen con su documentación; 4) el código fuente, como anexo.
10. Dicha documentación, presentación, estructura, contenido, redacción, será un elemento básico en la evaluación de la práctica.
11. Si el profesor del laboratorio lo estimara necesario, deberá realizarse una defensa presencial de la práctica.