



Práctica de curso 2024-25

El objetivo de la práctica es elaborar un programa Java que implemente el juego ¡La Conquista!

Descripción del Juego

¡La Conquista! es un juego para dos o más jugadores que se juega en una cuadrícula de puntos en turnos alternativos. En cada turno, el jugador correspondiente agrega un palo vertical u horizontal que conecte dos puntos (ver Figura 1). Si completa un cuadrado, coloca su inicial en el centro del cuadrado, y vuelve a colocar otro palo. El juego termina cuando no haya más palos que poner sobre la cuadrícula. Entonces, se contará el número de cuadrados completados por cada jugador. Gana el jugador que más cuadrados haya conseguido completar.

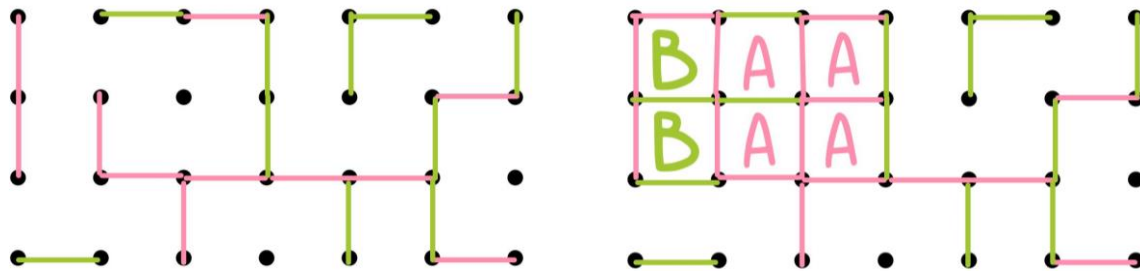


Figura 1. Ejemplo de situación de juego en una cuadrícula 3x6 antes (izquierda) y después de añadir algunos palos más y completar algunos cuadrados (derecha). Nota: el color de las líneas es irrelevante en el juego.

Requisitos de diseño del programa:

En esta práctica, se pide un programa que permita jugar a ¡La conquista! con dos jugadores. Para ello, inicialmente el programa dará la bienvenida al juego con el mensaje “Bienvenido a ¡La Conquista!” y posteriormente se mostrará un menú con diferentes opciones para los usuarios, tal y como se muestra en la Figura 2. Entonces, se le pedirá al usuario que elija una opción. Si el usuario introduce una opción incorrecta, se le mostrará el mensaje “Opción no válida”, se volverá a mostrar el menú, y se le pedirá de nuevo al usuario que elija una opción.

```
Bienvenido a ¡La Conquista!  
-----  
1. Nuevo Juego  
2. Cargar Partida  
3. Ver Resultados  
4. Salir  
-----  
Elige una opción:
```

Figura 2. Menú inicial del programa.

Si el usuario elige la **opción #4**, el programa terminará.

Si el usuario elige la **opción #3**, el programa mostrará los resultados de todas las partidas terminadas hasta el momento desde la más reciente a la más antigua, a razón de línea por partida. Cada línea contendrá la siguiente información (1) fecha de la partida; (2) tamaño del tablero jugado; (3) cuadrados del jugador 1; y (4) cuadrados del jugador 2 (ver Figura 3). Esta información se leerá de un fichero llamado “resultados.txt” situado en un directorio llamado “ficheros” dentro del directorio raíz del proyecto.

```
[Sat Nov 23 23:39:19 CET 2024] Tam: 3x3    Jugador 1: 6 vs. Jugador 2: 9
[Sat Nov 23 23:37:58 CET 2024] Tam: 2x2    Jugador 1: 0 vs. Jugador 2: 6
```

Figura 3. Ejemplo de posible representación de los resultados de las partidas terminadas.

Si el usuario elige la **opción #2**, el programa cargará la última partida guardada, y pudiendo continuarla en el mismo punto donde se dejó cuando se guardó, es decir, misma situación del tablero y mismo turno de jugador. La partida se cargará de un fichero llamado “partidaGuardada.txt” situado en el mismo directorio “ficheros” que en la opción anterior.

Si el usuario elige la **opción #1**, el programa solicitará introducir el tamaño del tablero en el formato AxB donde A será el número de filas de cuadrados, y B será el número de columnas, ambos con un valor máximo de 5. Una vez introducido, se mostrará por pantalla un tablero vacío en el que sólo habrá una cuadrícula de puntos con las dimensiones especificadas. En esta versión del juego, la interacción para añadir nuevos palitos será a través de la asociación de cada posible palito de la cuadrícula con un carácter comprendido entre A-Z, a-z y 0-9. Por ejemplo, la Figura 4 muestra el tablero inicial para el tamaño 2x5.

•	A	•	B	•	C	•	D	•	E	•
F		G		H		I		J		K
•	L	•	M	•	N	•	O	•	P	•
Q		R		S		T		U		V
•	W	•	X	•	Y	•	Z	•	a	•

[Jugador1] Próximo palito (** para guardar y salir):

Figura 4. Ejemplo de representación para un juego nuevo de tamaño 2x5.

Al inicio del juego, empezará el jugador #1 o el jugador #2 con un 50% de posibilidades cada uno. A continuación, se solicitará al jugador correspondiente que introduzca un carácter existente en el tablero, para así poner un palito en su ubicación. En caso de que se complete un cuadrado, el cuadrado se marcará con un carácter representativo del jugador correspondiente, y el mismo jugador volverá a colocar un nuevo palito. En caso contrario, el turno pasará al otro jugador. Para guardar un juego, se deberá escribir la secuencia “***”. De esta manera, el juego se guardará en el fichero indicado en la opción #2, y se mostrará el menú inicial. Las entradas deberán validarse. Una vez completados todos los palitos (ver Figura 5), el programa mostrará el mensaje “¡Juego Terminado!”, se mostrará el número de cuadrados obtenidos por cada jugador, y se almacenará la información detallada en la descripción de la opción #3, en el fichero correspondiente. A continuación, se le preguntará al usuario si quiere jugar otra partida. En caso afirmativo, se le volverá a preguntar por el tamaño del tablero de la nueva partida, y empezará un nuevo juego. En caso negativo, el programa volverá al menú inicial.

•	---	•	---	•	---	•
	*		*		*	
•	---	•	---	•	---	•
	#		*		*	
•	---	•	---	•	---	•
	*		#		#	
•	---	•	---	•	---	•

Juego Terminado!!
 Jugador 1: 3 cuadrados
 Jugador 2: 6 cuadrados

Figura 5. Situación del tablero e información a mostrar una vez terminada la partida.

Guías para la programación y refinamiento:

Para cada nueva partida (o partida cargada), es recomendable que sólo exista una matriz que contenga:

- Los puntos (•) en las casillas correspondientes
- Los caracteres mencionados anteriormente para cada posible palito a colocar. A medida que se van colocando dichos palitos, se van reemplazando por un carácter asociado que no se utilice para las posiciones de los palitos (p. ej., |, -).
- Un espacio en blanco para los centros de los cuadrados, que, a medida que el juego avance, se irán completando con el carácter asociado a cada jugador (p. ej., *, #).

Es recomendable que el programa tenga un método para representar dicha matriz. Con el objetivo de que la representación visual esté optimizada (ver Figura 5), se recomienda que las líneas horizontales se representen como tres guiones (-), y se deje el correspondiente espacio dentro de los cuadrados.

Se recomienda también que, para hacer las pruebas de los diferentes métodos que se irán programando, no se empiece el programa desde el principio, sino que directamente se prueben los métodos (o la parte del código correspondiente) para unos valores dados (independientemente del resto del programa).

Como la mayoría del software, antes de hacer la entrega, se recomienda la realización de pruebas con potenciales usuarios del juego. De esta manera, los programadores podremos detectar si los usuarios entienden las instrucciones del programa y así mejorar su usabilidad. En próximas asignaturas veréis cómo hacerlo de una forma más metodológica, pero podéis probar el programa desarrollado con amigos y familiares para intentar refinarlo antes del envío final, y documentarlo en la memoria.

Intentad mantener un registro y copia de las versiones que vais desarrollando. Más adelante veréis lo que son las plataformas de control de versiones, especialmente útiles para proyectos colaborativos en la que varias personas trabajan sobre el mismo software. Podéis investigar un poco más sobre ello, o mantener manualmente un buen registro de vuestras versiones.

En esta práctica se permite el uso de herramientas de inteligencia artificial generativa sin repercusión en la nota de la práctica. Sin embargo, será obligatorio indicar en la memoria del proyecto con la correspondiente referencia a la herramienta, detallando sus diferentes usos.

Versiones de la Práctica:

Versión básica (máximo 7 puntos de 10):

Las opciones #1 y #4 antes descritas (sin lectura ni escritura de ficheros).

Versión completa (máximo 10 puntos de 10):

El juego tal y como se ha descrito. Además, cualquier otra mejora o ampliación del juego (p. ej., deshacer jugadas, elegir el número de jugadores, guardar múltiples partidas, cargar cuadrículas con diferentes formas, tableros con cuadrados que valen doble, uso de colores, permitir al usuario elegir sus propios caracteres) deberá ser comentada con el profesor de laboratorio, que podrá aumentar hasta en 1 punto la nota de la práctica.

Entrega de la Práctica:

La práctica debe realizarse por parejas, idealmente del mismo grupo de laboratorio o de grupos que comparten un mismo profesor. Cualquier otra situación deberá ser acordada con los profesores de prácticas.

Será calificada con arreglo a los criterios de calidad explicados en la asignatura (p. ej., robustez, eficiencia, mantenibilidad), el documento de “Malas prácticas en programación”, y considerando el acto de defensa.

Solo un alumno de cada pareja debe entregar la práctica. En todos los documentos entregados deben aparecer los nombres de los alumnos que la han realizado, junto con sus grupos de teoría y laboratorio. Ambos alumnos se hacen responsables de la entrega.

Se debe entregar, en el campus virtual:

- El **código** del programa en un único archivo .java de nombre elegido por los alumnos.
 - Debe ser compilable desde la línea de órdenes con la orden “`javac nombre.java`” y poder ser ejecutado con la orden “`java nombre`”.
- La **documentación** del programa en un archivo pdf, de no más de 6 páginas, en la que debe describirse:
 - Una pequeña introducción sobre el programa que se ha desarrollado (p.ej., características del juego, posibles dificultades, qué versión se ha realizado, la estrategia general de resolución).
 - La cabecera de los métodos desarrollados y su funcionalidad, incluyendo la especificación de cada uno de ellos (objetivo, precondiciones, entradas, salidas, etc.).
 - las estructuras de datos relevantes utilizadas en el programa, con una explicación de su contenido y significado en el programa.

Esta documentación no debe contener el código del programa a no ser que se trate de ejemplos para mejorar la comprensión de su información.

La entrega debe realizarse antes de la última clase de laboratorio de diciembre. Es decir, la fecha límite será:

- Domingo 15 de diciembre a las 23:59h para X4 y X5
- Lunes 16 de diciembre a las 23:59h para X3 y X6
- Martes 17 de diciembre a las 23:59h para X1, X2, X7 y X10
- Miércoles 18 de diciembre a las 23:59h para X8 y X9

Defensa de la Práctica:

El profesor de prácticas convocará a los alumnos un acto de “defensa de la práctica” que podrá ser durante el último laboratorio de prácticas de la asignatura (última semana lectiva de diciembre), en horario de tutorías, o en un horario acordado con el profesor de laboratorio. En ella, el profesor hará preguntas a los alumnos, relacionadas con la práctica entregada y su elaboración, y pedirá la realización de modificaciones en directo, para que demostréis que sois conocedores de cómo funciona vuestro programa y de su código. No saber describir el software desarrollado o no saber hacer los cambios solicitados podrá suponer la suspensión de la práctica.