

EIF204 – Programación 2

Proyecto de programación #2

Prof. Georges E. Alfaro Salazar
Prof. Santiago Caamaño Polini (coordinador del curso)
Prof. Jennifer Fuentes Bustos
Prof. Karol Leitón Arrieta

OBJETIVOS DEL PROYECTO

El objetivo del proyecto es aplicar los conceptos teóricos, principios y técnicas estudiadas en clase. Es importante que al desarrollar el proyecto se escriba código correctamente estructurado, debidamente encapsulado y fácilmente reutilizable (sin considerar plagio del código).

Además, se considera importante la aplicación correcta de los patrones de diseño estudiados.

DESCRIPCIÓN DEL PROYECTO

El programa a desarrollar es un juego clásico de damas españolas (tablero), utilizando una interfaz de consola. La aplicación permitirá que una persona juegue contra la computadora. No se requiere que el programa guarde un registro de jugadas o lleve ningún tipo de bitácora. Se busca que el programa use alguna forma adecuada de representación del juego y el estado de la partida, y que implemente un conjunto de algoritmos utilizando un patrón de estrategia.

Se utilizará un tablero de 64 casillas (8 × 8), donde cada jugador cuenta con 12 piezas o peones.

8	O		O		O		O	
7		O		O		O		O
6	O		O		O		O	
5								
4								
3		X		X		X		X
2	X		X		X		X	
1		X		X		X		X
	1	2	3	4	5	6	7	8

Las filas y las columnas estarán numeradas de 1 a 8, comenzando de la esquina inferior izquierda del tablero. Al iniciar el juego, el programa debe consultar cuál jugador (la computadora o la persona) comenzará el juego. Los peones de la persona que juega siempre estarán colocados en la parte inferior del tablero.

Las piezas blancas (del jugador) se mostrarán usando una **X**, y las piezas negras de la computadora utilizarán una **O**. Cuando los peones logran coronarse y convertirse en damas, se emplearán una **B** y una **N**, respectivamente.

Para mover una pieza, el jugador indicará la fila y la columna donde se encuentra dicha pieza y la posición final.

Por ejemplo:

36 47

Indica que la pieza en la fila 3 y la columna 6 se moverá a la casilla en la fila 4 y columna 7.

8	O		O		O		O	
7		O		O		O		O
6	O		O		O		O	
5								
4							X	
3		X		X				X
2	X		X		X		X	
1		X		X		X		X
	1	2	3	4	5	6	7	8

En la siguiente figura, el jugador mueve su pieza desde la casilla 43 hasta la posición 87. El jugador ha capturado dos piezas de la computadora y se promociona a una dama.

43 87

8	O		O		O			
7		O		O		O		O
6	O		O				O	
5				O		O		
4	X		X					
3		X				X		X
2	X		X		X		X	
1				X		X		X
	1	2	3	4	5	6	7	8

8	O		O		O		B	
7		O		O				O
6	O		O				O	
5						O		
4	X							
3		X				X		X
2	X		X		X		X	
1				X		X		X
	1	2	3	4	5	6	7	8

El jugador podría también haber digitado una secuencia de casillas, indicada cada una por las que ha de pasar, por ejemplo:

43 56 87

Pero en la mayoría de los casos es innecesario, ya que la jugada realizada es la única posible. Tenga en cuenta que, dado que las damas pueden moverse libremente sobre cualquier diagonal sobre varias casillas, es posible que el usuario deba indicar todas las posiciones necesarias para evitar movimientos ambiguos.

Cada vez que se realice una jugada, se actualizará el despliegue del tablero.

CONSIDERACIONES DE IMPLEMENTACIÓN

El código del proyecto debe estar estructurado adecuadamente, respetando los principios y patrones de diseño estudiados en clase.

El programa debe contemplar la posibilidad de utilizar varias estrategias de juego. Las dos estrategias básicas a implementar son:

- Que el programa juegue de manera completamente aleatoria
- Una estrategia que garantice que el programa gane el juego

La estrategia para ganar el juego es compleja y se implementará usando un conjunto de clases y algoritmos debidamente separados del resto del programa. Aunque no se solicita ningún tipo de persistencia, es importante que se pueda verificar el funcionamiento de los algoritmos, mostrando algún tipo de información de rastreo. De la misma manera, el programa permitirá el uso de un modo de prueba, donde se pueda cargar una configuración para el tablero, con posiciones definidas de antemano.

Se solicita también que se implemente por lo menos una estrategia adicional, diferente de las dos básicas planteadas.

Por ejemplo, podría considerar estrategias tales como:

- hacer que el juego sea totalmente ofensivo (dirigido al ataque)
- tener un juego completamente defensivo.

<https://www.aboutespanol.com/estrategias-para-ganar-el-juego-de-damas-2077678>

<https://es.wikihow.com/ganar-en-damas>

<https://www.fmjd.org/downloads/Course/pt/S6.%20Liquidando%20el%20juego.pdf>

En cualquiera de las estrategias utilizadas, el programa siempre realizará jugadas legales dentro de las reglas del juego, y verificará también que el jugador no realice ningún movimiento incorrecto.

El usuario debe poder detener la partida en cualquier momento e iniciar una nueva. De la misma manera, el jugador podrá guardar una partida para reanudarla posteriormente.

En la entrega, se deberán incluir los diagramas de clase y los archivos de proyecto correspondientes. Cada profesor indicará cual es el ambiente de desarrollo (IDE) a utilizar (*CodeBlocks*, *NetBeans*, *Visual Studio*, etc.). Los archivos de proyecto tienen que entregarse en el formato adecuado para su revisión. No se recibirán proyectos en un formato diferente al indicado. Los diagramas de clase podrían elaborarse usando UMLet (<http://www.umlet.com/>) u alguna otra aplicación apropiada y se guardarán usando el formato propio de la aplicación y en PDF.

En particular:

- Las clases deben separar correctamente la declaración de la interfaz y su implementación, usando archivos de cabecera (archivos .h) y código fuente (archivos .cpp) por aparte. Los archivos de cabecera deberán siempre usar guardas (se puede especificar la directiva `#pragma once` cuando se utilice Visual Studio).
- Las clases de entidad no deberán contener código de entrada/salida (como salida a la consola usando `cout`, por ejemplo)
- El manejo de la interfaz con el usuario debe hacerse por medio de una o varias clases diseñadas para tal efecto.

Es importante que el diseño general del programa considere la reutilización de código. En todos los casos, los contenedores y otras estructuras deben implementarse de la forma más general posible.

ENTREGA Y EVALUACIÓN

El proyecto debe entregarse **por medio del aula virtual, en el espacio asignado para ello, o de la manera solicitada por el profesor del curso**. La fecha de entrega es el día **viernes 7 de junio de 2019**. No se aceptará ningún proyecto después de esa fecha, ni se admitirá la entrega del proyecto por correo electrónico. **La realización de proyecto se hará individualmente o en pareja. En caso de que el proyecto se resuelva en pareja, se indicará previamente cuál será la conformación del grupo de trabajo, enviando un correo con un documento de texto o de la manera en que el profesor lo indique. No se aceptarán trabajos si no se recibe oportunamente dicha información.**

Incluya comentarios en el código de los programas y describa cada una de las clases y métodos utilizados cuando sea útil y conveniente.

El código del programa debe compilar correctamente. Si el código no puede compilarse, el trabajo será calificado con una nota de 0 (cero). Para ser evaluado, tiene que cumplirse al menos un 50% de la funcionalidad solicitada. De lo contrario, el proyecto será considerado como no entregado y se penalizará la nota correspondiente.

Durante la revisión del proyecto, es muy importante defender adecuadamente la solución propuesta.

Se evaluarán los siguientes aspectos:

Criterio de evaluación	Valor
1. El programa despliega correctamente el tablero (incluyendo las piezas del juego)	10 pts.
2. La interfaz del programa permite indicar cuál es la pieza a mover y su nueva posición	5 pts.
3. El programa verifica que las jugadas realizadas sean legales	10 pts.
4. El programa tiene una opción para permitir cambiar la estrategia a utilizar	10 pts.
5. El programa verifica correctamente el fin de la partida	5 pts.
6. El programa permite terminar en cualquier momento (finalizando la partida)	5 pts.
7. El programa permite reiniciar una partida en cualquier momento	5 pts.
8. El programa guarda el estado de la partida	5 pts.
9. La estrategia aleatoria mueve las piezas correctamente	10 pts.
10. La opción para cargar una configuración previa del tablero funciona correctamente	10 pts.
11. La estrategia para ganar la partida funciona correctamente	15 pts.
12. La estrategia adicional seleccionada está debidamente implementada (esto incluye la implementación según la documentación correspondiente)	10 pts.

Observaciones generales:

- Los proyectos deben entregarse con toda la documentación, diagramas, código fuente y cualquier otro material solicitado.
- Los trabajos no se copiarán de ninguna llave USB u otro dispositivo en el momento, sino que se deben entregar en el formato adecuado.
- **Cualquier trabajo práctico, que no sea de elaboración original del estudiante y haya sido copiado o adaptado de otro origen (plagio), de manera parcial o total, se calificará con nota 0 (cero) y se procederá como lo indiquen los reglamentos vigentes de la universidad.**