Universidad Nacional de Costa Rica

Facultad de Ciencias Exactas y Naturales

Escuela de Informática

Curso: Estructuras de Datos

Prof. Santiago Caamaño Polini

MANUAL DE USUARIO

PROYECTO I - DAMAS II-2016

Se presenta el manual de usuario para el juego de damas inglesas en CLI y documentación relevante sobre el desarrollo de la aplicación. Incluidos diagramas de diseño UML, alcance, limitaciones y otros aspectos importantes.

Introducción

La aplicación es una representación en CLI de un juego de damas inglesas para 2 jugadores. En ella, el jugador es capaz de realizar todos los movimientos y jugadas normales de un juego de damas. Todo esto mediante una interfaz de usuario amigable e intuitiva, donde se utilizará una matriz de display para la visualización del estado del juego.

El objetivo de la realización de este proyecto es poner en práctica los principales conceptos estudiados en clase, como lo son la programación orientada a objetos, manejo de punteros, programación recursiva, listas simple y múltiplemente enlazadas, navegabilidad en listas, entre otros.

La aplicación será desarrollada en el lenguaje de programación C++, y como entorno de desarrollo (IDE) el Visual Studio Express 2012 para Escritorio.

Desarrollo de la aplicación

Para el desarrollo del juego se optó por utilizar una matriz hecha a base de listas cuádruplemente enlazadas como la principal estructura de datos. La idea de esto es representar de forma lógica lo que sucede en un tablero real de un juego de damas. En este caso se decidió que cada elemento de la lista (nodos) tuviera 4 enlaces debido a que los movimientos permitidos son únicamente en diagonal, nunca en sentido vertical u horizontal.

Las listas están compuestas por nodos interconectados que, por el contexto, llamaremos Casillas. Estas Casillas, a su vez, tienen punteros otras 4 Casillas, que se entenderán como sus "vecinos". Para facilitar la conceptualización, se optó por trabajar por un sistema de puntos cardinales, como se muestra en la Figura I. Y luego, cada Casilla contiene un objeto Ficha.

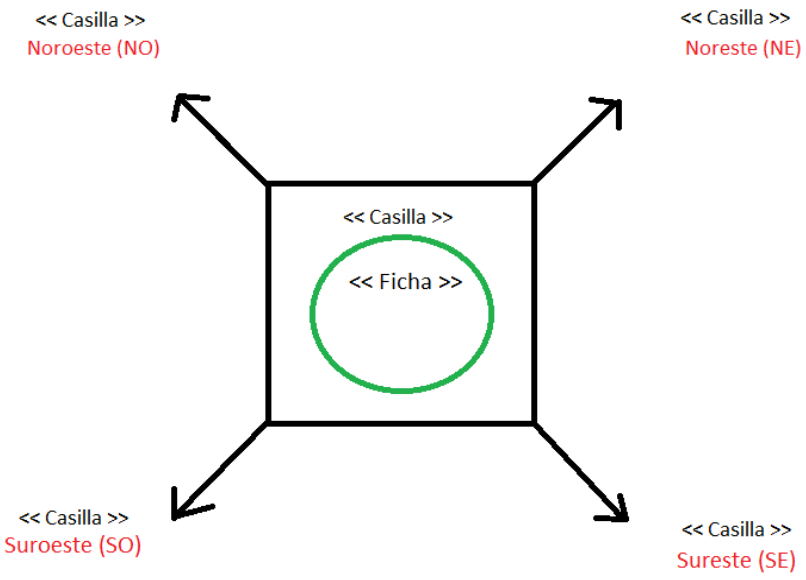


Figura I. Conceptualización de Nodo como Casilla.

Entendiendo las casillas de esta manera resulta muy conveniente, pues se facilita la navegabilidad a través del tablero. Resulta suficientemente claro que las casillas contienen un puntero a una Ficha. Por lo tanto, si una casilla en un momento dado se encuentra vacía, el valor de su puntero será NULL.

El paso siguiente fue desarrollar las listas que confirmarían la matriz y la interconexión de cada una de las casillas con sus casillas vecinas. El abordaje fue más o menos así:

1. Desarrollar una lista que formara la diagonal en sentido NO → SE.
2. Ir desplazándose por cada nodo de la diagonal recién creada e inicializar otra lista, con éste nodo como cabeza, e ir agregando nodos atrás y delante de esta sub-lista, pero esta vez en sentido SO → NE.
3. Por último, interconectar las casillas de las sub-listas con sus vecinos NO y SE.

Finalizado este proceso de interconexión de casillas se obtiene una matriz como se muestra en la Figura 2, donde toda casilla conoce quiénes son sus 4 vecinos.

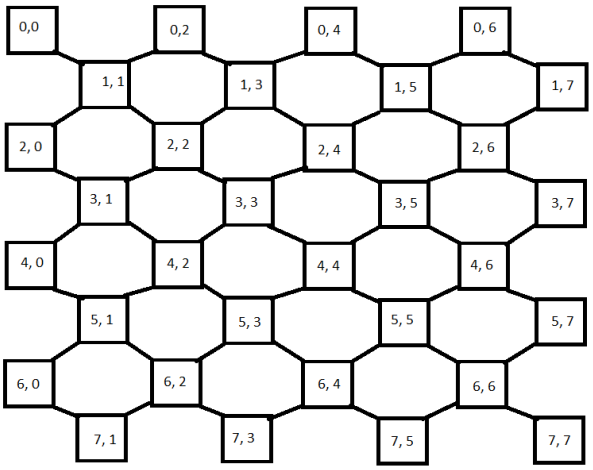


Figura 2. Matriz formada mediante lista de listas.

El otro gran problema por abordar era la colocación y movimiento de fichas. Para esto fue *fundamental (!)* implementar un método llamado *buscaUnaCasilla(int x, int y).* El cual recibe unas coordenadas por parámetro y recorre todo el tablero, buscando en cada lista y sub-lista, hasta encontrar la casilla solicitada. En caso de éxito devuelve un puntero hacia esa casilla, si no, un puntero NULL.

Teniendo todo el “setup” como planteado, queda definir, validar y permitir los movimientos de una ficha según las reglas del juego. Esto resultó relativamente sencillo una vez implementado el método antes mencionado. Movimientos especiales como los de una “reina”, de comer y de comer múltiple son validados mediante el conocimiento de “quién es mi vecino?” de cada ficha.

Las cuestiones relevantes al manejo del juego propiamente, como la gestión de los turno, condición de gane o finalización del juego, no serán explicadas en esta sección pues toman un papel secundario en el desarrollo. Sin embargo, el diagrama de clases UML da una idea suficientemente clara del funcionamiento general del juego. (Ver Figura 3.)

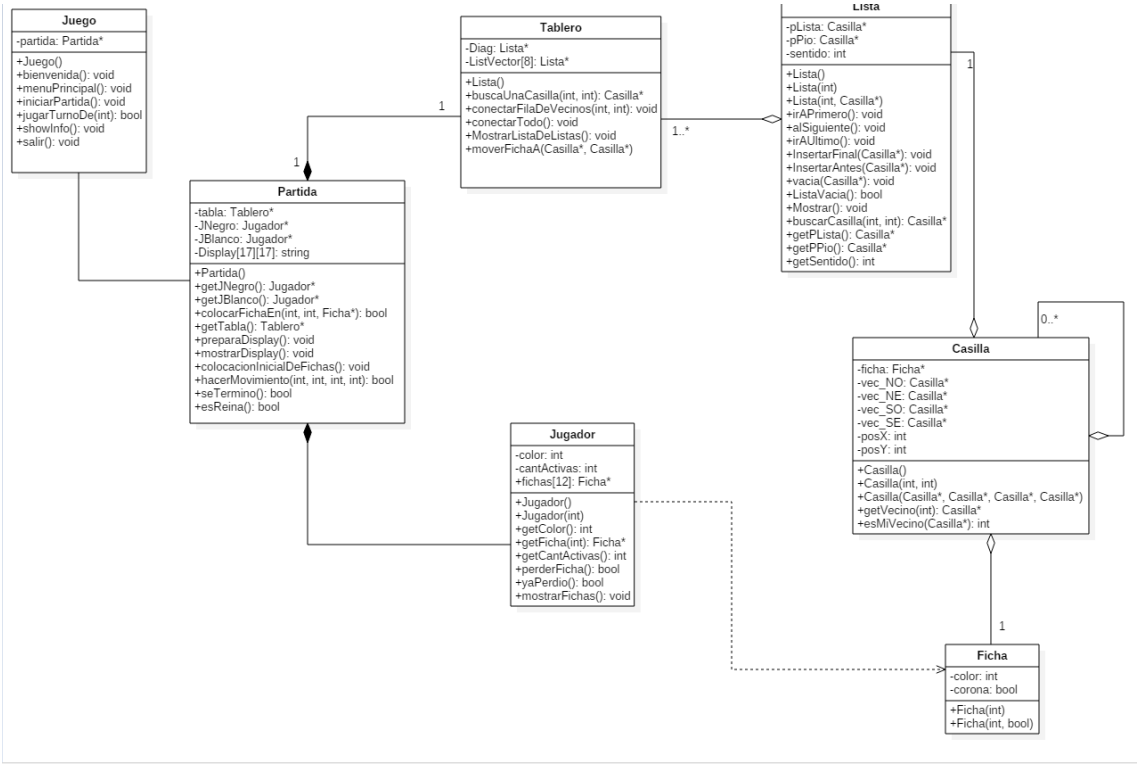
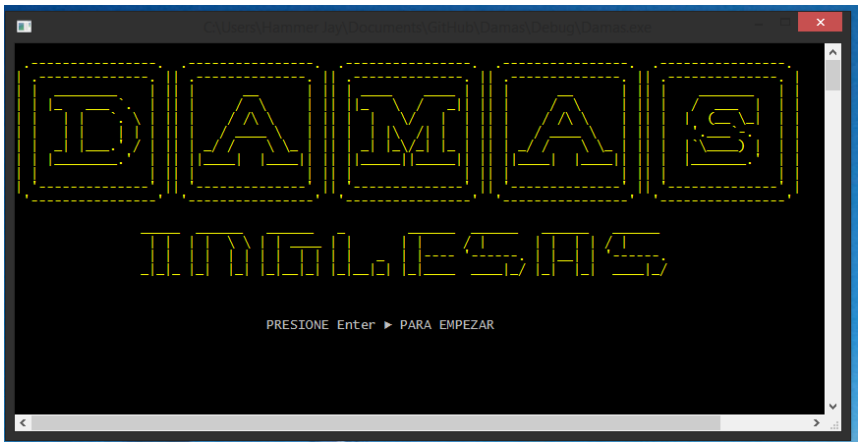


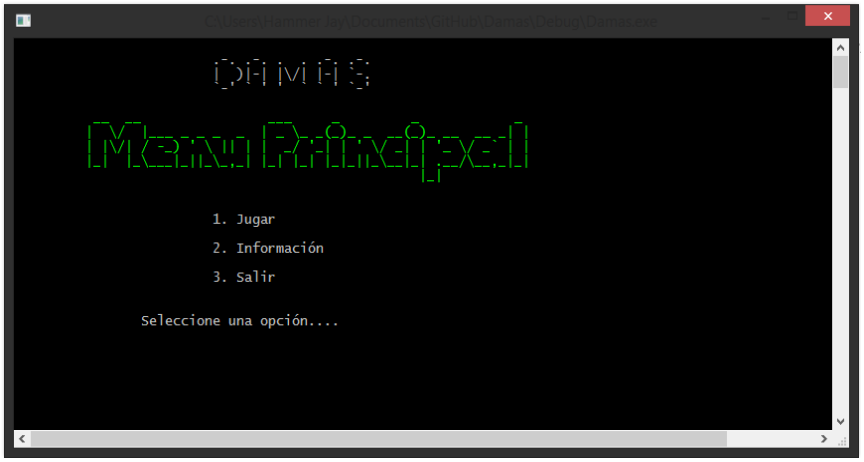
Figura 3. Diagrama de clases UML del sistema.

MANUAL DE USUARIO

1. Al iniciar el juego, el usuario es recibido por una pantalla de bienvenida al juego, como en la imagen. Presionar “Enter” para continuar.

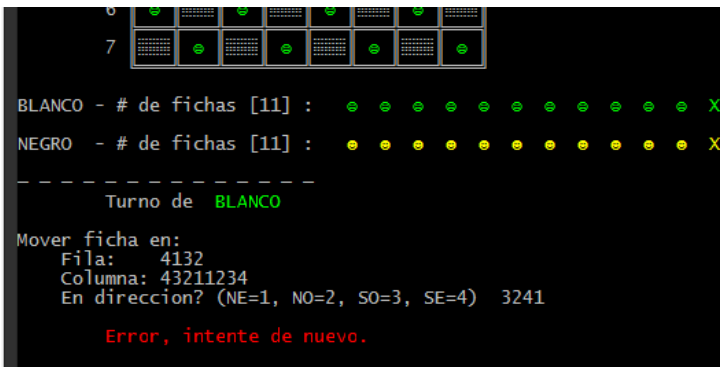


2. Luego, se presenta el menú principal del juego. El usuario puede seleccionar: iniciar una partida de damas, ver información sobre el juego o salir de la aplicación. La selección se hace ingresando el número de tal.



3. Si selecciona jugar, se mostrará un tablero que muestra la posición de las fichas de cada jugador. Así como la cantidad de fichas en juego y las fichas comidas de cada uno.

4. El sistema se encarga de manejar los turnos por jugador. Se le solicita al jugador que indique las coordenadas de la ficha que desea mover y en qué dirección desea hacerlo. Si el movimiento es inválido, o los datos ingresados no corresponden, se mostrará un mensaje de error y la posibilidad de hacer el reingreso.



5. Los usuarios alternarán turnos hasta que el juego acabe. El juego finaliza cuando algún jugador se quede sin fichas o un contrincante se retire.

6. Terminado el juego, se devuelve al usuario al menú principal.