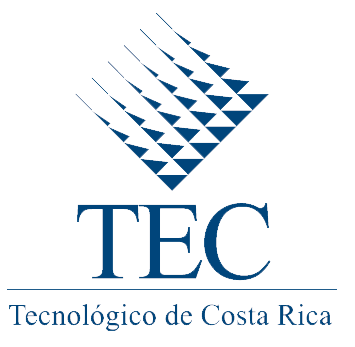
**Instituto Tecnológico de Costa Rica**

**Sede San Carlos**

****

**Escuela de Ingeniería en Computación**

**Curso: Taller de programación**

Damas Génesis

**Profesor:**

**Leo Víquez Acuña**

**Estudiantes:**

**José Daniel Alfaro Bolaños**

**Luis Alonso Vega Brenes**

**Junio de 2010**

## Introducción

En este documento se presenta un análisis del sistema *Damas Génesis*, un software elaborado para simular el juego de mesa conocido como **damas** o **tablero**. El juego en sí tiene una cantidad considerable de reglas, y además existen muchas variaciones en ellas, sin embargo, en este programa se consideraron las que se solicitan en la consigna del segundo proyecto, las cuales son reglas bastante estándar.

Entrando un poco en el sistema en sí, se puede tomar en cuenta que está diseñado en Scheme, una herramienta de desarrollo creada por PLT, orientada a la recursividad y que utiliza un lenguaje interpretado. Esta herramienta es muy utilizada a nivel universitario en la enseñanza de la programación. *Damas Génesis* fue desarrollado usando la versión 4.2.1 de DrScheme y fue probado en la plataforma de Windows 7.

El sistema fue creado usando los conceptos de árboles, objetos y matrices, los cuales constituyen la base de este proyecto: Las matrices representan el tablero con las fichas, los objetos contienen posibilidades de jugadas y los árboles cada posibilidad dado un estado de tablero. Por otra parte, la apariencia visual está desarrollada usando la interfaz gráfica de usuario.

La forma de manejar el programa es realmente básica y sencilla, el usuario inicia en diálogo donde selecciona el modo de juego y se ingresan los datos del usuario. Luego se ingresa al espacio de juego en sí. De ahí en adelante los jugadores pueden reproducir el juego de mesa de una forma instintiva, seleccionando las fichas y luego el lugar donde quieren moverlas.

## Análisis del problema

El objetivo del proyecto es generar un simulador del juego de mesa conocido como damas o tablero. El usuario podrá seleccionar un modo de juego, en este caso de jugador contra jugador o de jugador contra la máquina, y luego ingresará en la partida en sí, donde realizará las jugadas que desee y se realizará el desarrollo normal del juego hasta que alguno de los dos jugadores consiga acabar las fichas del oponente.

A continuación se lista una serie de funcionalidades requeridas para el sistema:

**Reglas del juego**

Movimientos

Las reglas son muy estándar, y algunas de ellas son: las fichas regulares solo podrán moverse un espacio hacia adelante, en el caso de tomar una ficha contraria, la ficha inicial quedará del otro lado de la del oponente. Las coronas podrán moverse libremente hacia cualquier dirección (en línea recta diagonal) y cuantos espacios quiera y el estilo de comida es el mismo que el de las fichas regulares.

Inicio del juego y coronas

Para el inicio del juego, ambos jugadores tienen doce fichas, las cuales estarán posicionadas a cada extremo del tablero, ocupando las 3 filas extremas de cada lado. Cuando una ficha regular alcanza el extremo contrario del tablero, se convierte en una reina o corona, la cual podrá moverse en la forma especificada arriba.

Cambios de turno

Cada vez que un jugador realiza una jugada (que no es de comida múltiple) se le da el turno al otro jugador. Se alterna de esta manera durante el transcurso de todo el juego.

**Tiempos de juego**

El juego temporizado contiene la cualidad de limitar el transcurso de tiempo que tiene un jugador para realizar o completar una jugada. Si el jugador no realiza su jugada en el tiempo establecido, se le da el turno al oponente.

**Salvar y cargar partidas**

Otro aspecto del sistema es la capacidad de guardar y cargar partidas, las cuales serán archivos con un nombre seleccionado por el usuario. También se debe dar la opción de generar un nuevo juego y abandonar otro.

## Solución del problema

De forma general, se utilizaron una serie de estructuras de datos para solventar las necesidades del juego. Entre ellas encontramos: matrices para definir tableros, objetos para contener los tableros y árboles con estados del tablero que tienen en sus hijos los posibles estados del tablero para cada movida del jugador.

Seguidamente se presentan las soluciones utilizadas para tratar con cada uno de los sub-problemas.

**Tablero**

El tablero de juego fue diseñado como una matriz de 8x8 espacios. De forma gráfica, se muestra como una serie de botones ilustrados con el contenido de la matriz de juego1. Las funciones de seleccionar y mover ficha se realizan al presionar estos botones, los cuales contienen la información de la posición que ocupan. A partir de esta información se realiza el cálculo de la movida que se realiza, ya sea de seleccionar una ficha, cancelar la selección, moverla, realizar una comida sencilla o una múltiple, si la ficha movida se corona o, finalmente, si el juego se ha terminado porque uno de los dos jugadores se ha quedado sin fichas.

**Movimientos**

Para revisar los movimientos que realizan los jugadores humanos, se revisan las posibles jugadas que puede tener una ficha cuando está seleccionada, y en el siguiente paso se comprueba que la casilla a la que se quiere mover esté dentro de las posibilidades. Si esta condición se cumple, el sistema devuelve el estado del tablero que genera realizar la jugada mencionada anteriormente. Este estado incluye los cambios de ficha regular a corona y las comidas de fichas enemigas. Además, si el movimiento se completa con éxito, se le da el turno al otro jugador.

**Creación de fichas reina**

Para la transformación de una ficha regular a una corona, se verifica la posición que tiene luego de moverse, y se cambia por una corona si se da el caso de que haya llegado a un extremo del tablero.

**Temporizadores**

El sistema tiene un proceso aparte que lleva un contador de tiempo. Este se encarga de cambiar el turno del jugador cuando este último se haya tomado más tiempo del que le corresponde, según las reglas establecidas anteriormente en relación con los tiempos de juego para cada movida.

**Inteligencia artificial**

Las jugadas del computador están generadas en función de las posibles jugadas que tiene en el momento de jugar y las posibles reacciones que tiene el jugador rival. Estas posibilidades son manejadas de una forma jerárquica en una cantidad definida de niveles, tomando el estado del tablero y los resultados o consecuencias de mover cada ficha en cada posible posición. La máquina elige la movida que le garantice una relación más favorable en la cantidad de fichas de ambos jugadores.

**Partidas salvadas**

Al guardar una partida, se guarda la información relevante del juego, incluyendo el tablero, el turno y los datos de los jugadores. Luego, al cargar una partida guardada, se coloca esa información en el sistema, y se muestra el estado de la partida en el espacio de juego. El usuario puede salvar y cargar el juego utilizando un diálogo sencillo.2

## Análisis de resultados

El siguiente cuadro presenta la lista de secciones del proyecto, las cuales representan al mismo tiempo las soluciones a los requerimientos propuestos para la correcta implementación del sistema en general.

|  |  |
| --- | --- |
| Apartado o sección del programa | Resultados |
| Movimientos de fichas regulares y coronas | Realizado con éxito |
| Coronar fichas | Realizado con éxito |
| Condiciones de gane de juego | Realizado con éxito |
| Salvar y cargar partidas | Realizado con éxito |
| Juego temporizado | Realizado con éxito |
| Inteligencia artificial | Realizado con éxitoa |
| Extras (varios niveles de AI) | Realizado con éxito |

Notas:

1. La inteligencia artificial calcula un “árbol de juego” con las posibles movidas de cada jugador y calcula la jugada más favorable, sin embargo, no toma en cuenta que un jugador realice comidas múltiples, ni tampoco indica cuando la máquina puede realizarlas.

## Conclusión

El sistema de Damas Génesis, de forma general, implementa las funcionalidades requeridas. Algunas de las observaciones o conclusiones que se pueden tomar en cuenta son las siguientes:

* La implementación de las reglas del juego es más sencilla al subdividir cada apartado en funciones independientes que pueden ser útiles para otros apartados.
* La utilización de matrices: En resumen, se puede afirmar que es una forma muy sencilla de manejar tablas o situaciones en las que se necesita información organizada en dos dimensiones. Para utilizarlas generalmente es más sencillo manejar paralelamente vectores, y utilizar iteraciones para recorrerlos.
* El uso de objetos: Estos son útiles para crear varias copias de variables que son similares y que pasarán a llamarse instancias. Los objetos contienen información interna que puede ser obtenida de forma indirecta, utilizando funciones intermedias. En el lenguaje de programación de Scheme, el manejo de objetos es un tanto complicado y requiere más esfuerzo para utilizarse, pero sigue teniendo la misma facilidad que permite la interacción y encapsulación de objetos.
* Los árboles como estructuras de datos son realmente útiles en muchos casos, y en este proyecto fueron de gran utilidad para la generación de posibilidades estados del tablero y de todas las sucesiones que puede computar la inteligencia artificial. Además, permitieron realizar cálculos recursivos en las posibles jugadas consecuencia de un tablero dado.
* Los archivos fueron una vez más utilizados para guardar estados de juego que trascienden la ejecución del programa, y que permiten continuar una partida luego. Para implementarlos, siempre es útil organizar la información relevante del juego y guardarla, luego cuando se carga, se lee la información y cada sección se sobrepone en el elemento correspondiente.

## Citas bibliográficas

1. *Controles de interfaz de usuario generales*. PLT Scheme. Consultado en ayuda de Scheme: /PLT/doc/gui/Windowing\_Classes.html
2. *Controles de interfaz de usuario gráfica: Diálogos.* PLT Scheme. Consultado en ayuda de Scheme: / PLT/doc/gui/Windowing\_Functions.html#(part.\_.Dialogs)