

# Ingeniería software

Javier Andrés Reyes González  
Felipe Sebastián Canales Saavedra  
Felipe Aracena Carvacho  
Departamento de ingeniería informática.  
Universidad tecnológica metropolitana de Chile  
Profesor: Sebastián Salazar Molina  
Sección: 30411

## Paper

### JUEGO DAMAS

#### Introducción

Las damas es un juego que se realiza por dos contrincantes que consiste en mover las piezas en diagonal a través de cuadros negros o blanco en un tablero que varia pero que generalmente son de 100 cuadros con la intención de capturar las piezas del contrario saltando por encima de ellas. La partida finaliza en cualquier de estas 3 situaciones.

- Pierde el que se queda sin piezas.
- Cuando el jugador tiene muy pocas piezas puede retirarse.
- Cuando en un turno no puede moverse.

En la programación el juego damas puede conducir en un problema complejo, porque al igual que en el ajedrez. Necesita programar una gran cantidad de movimientos establecidos en diferentes posiciones y/o situaciones que puede contraer conflictos simples o complejos dependiendo la magnitud del movimiento que realizo el contri cario.

## Objetivos

Dejar sin movimiento al rival, por medio de:

- Inmovilizar al contrincante, sin que pueda mover ninguna de sus piezas.
- Capturar todas las piezas enemigas

## Problema

Al comenzar a solo resolver el problema de las damas, lo primero que tuvimos inquietud fue de como se podría realizar que la CPU pudiera mover fichas por sí misma, ya que contrae una gran complejidad para programar por su gran cantidad de movimiento que se puede llegar a realizar con las damas.

## Implementación

Lo primero fue seleccionar el tipo de programa por el cual se hizo la implementación para Linux "gcc" y para windows "code::Block".

Al empezar a programar lo que se realizo primero fue crear el procedimiento de rellenar la tabla con los valores de "a" que son fichas negras y fichas blancas representada con la letra "b" que nos permitió visualizar. adicionalmente empezar a crear los movimiento que se pueda realizar el usuario.

El procedimiento siguiente creados fue MoverUsuario que permite hacer los distintos movimientos con que el jugador da una coordenada horizontal y una vertical para que se mueva. Sus principales características es que al tratar de mover en un espacio imposible que se pueda dar(Por ejemplo que salga del tablero y/o este ocupada la posición) da un mensaje de error. además nos invita nuevamente a realizar una nueva coordenada.

Si la coordenada está bien (que se encuentre vacio el espacio) se puede mover y/o comer si se encuentra una ficha negra ("a" o "A").

Luego al tener la experiencia de crear los movimientos del usuario. Se consiguió lograr crear los movimientos de la CPU que lo denomino "cpu". Que en un comienzo verifica si ocupa ficha o dama(dependiendo si ya pudo coronarse en el juego) para luego revisar en su alrededor si está ocupado los espacios y en algunos casos cuando quiera comer ver si no tiene una ficha que obstaculice su movimiento. Si no se llega a poder mover, se elige una nueva y hace le mismo procedimiento.

Los movimientos que realice la cpu nos obliga a comer una ficha o dama (dependiendo el caso) como regla. En ese caso se hizo el procedimiento llamado "Obligación". Este procedimiento permite adquirir la ficha obligatoriamente sin poder hacer otro movimiento.

Como por regla debemos adquirir o comer la ficha contrincante, al crear el procedimiento "obligación" que el usuario adquiere la ficha contrincante se realizó un procedimiento que es "ComeCPU" que hace lo mismo que el procedimiento obligación.

Luego realizamos el procedimiento "IdentificadamaUsuario" y "IdentificadamaCPU" que realiza que al llegar a los bordes (superior o inferior dependiendo si es usuario y/o CPU) se transforma automáticamente en dama que permite trasladarse de forma hacia arriba o abajo como estime conveniente.

Por último para que el juego realice todas las reglas necesarias como fundamental que si pueda comer o adquirir una ficha el procedimiento "Obligacion" es el primero en ejecutarse cuando aparte, si no ocupa el procedimiento "MoverUsuario". Es lo mismo que se realiza para el turno de la CPU.

Como muestra en la imagen de a continuación:

```
do
{
    Obligacion();    //Procedimiento que te obliga a comer ficha o dama de la CPU
    if(turno==1)
    {
        MoverUsuario();    //Movimiento del Usuario;
        IdentificarDamaUsuario(); //Identifica ficha para transformar a dama
        Limpiar();    //limpia la pantalla y ejecuta nuevamente llenado
        if (fichao==0) //Revisa si ganaste partida
        {
            cout<<"GANASTE " <<endl;
            terminar=2;
        }
        break;
    }
    ComeCPU();    //procedimiento en que CPU puede comer tu ficha
    if(turno==0)    //Movimiento de la CPU
    {
        cpu();
        IdentificarDamaCPU(); //Identifica ficha para transformarla en Dama CPU
        Limpiar();    //limpia la pantalla y ejecuta nuevamente llenado
        if(fichax==0) //Revisa si persiste la partida
        {
            cout<<"YOU LOST... LOSEER! " <<endl;
            terminar=2;
        }
        break;
    }
}
```

## **Conclusión**

Al realizar el programa fue una buena forma para reforzar los conocimientos, ya que permite comprender de una manera rápida y sencilla las técnicas que competen a la investigación de la complejidad de movimientos. Al contener una gran cantidad de movimientos permite resolver los problemas de forma instantánea y de que al poder jugar la CPU pueda elegir al azar la ficha con la que desea hacer el movimiento. De lo cual nos trajo un gran problema para poder ganar en el juego.