Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI2691 – Laboratorio Algoritmos y Estructuras I
Enero-Marzo 2018

Proyecto: Cuatro en Línea

El objetivo de este proyecto es la implementación de un programa no trivial, haciendo uso del análisis descendente y de las reglas de estilo y documentación vistas a lo largo de este curso de Laboratorio de Algoritmos I, así como de la teoría de Algoritmos y Estructuras I.

1. Planteamiento del Problema

Se desea que usted diseñe e implemente un programa que permita jugar Cuatro en Línea entre una persona y el computador. En este juego se tienen dos contrincantes que deben colocar sus fichas sobre un tablero respetando ciertas restricciones. Una demostración del juego entre una persona y el computador se encuentra en esta dirección.

El programa debe proveer un tablero gráfico de 6 filas y 7 columnas.

A cada jugador se le asigna uno de dos colores posibles para las fichas (por ejemplo, rojo para la persona, y azul para la computadora). Cada jugador, por turnos, debe colocar una ficha sobre el tablero, respetando las siguientes restricciones:

- La ficha solo puede colocarse en una casilla no ocupada.
- La ficha puede colocarse en casillas de la fila extrema inferior, o en una casilla vacía que esté encima de una casilla ocupada por otra ficha.

El primer jugador que logre alinear consecutivamente cuatro fichas de su color, ya sea de manera vertical, horizontal o diagonal, gana la partida. Si el tablero se llena antes de haber un ganador, la partida se declara como empate.

2. Niveles del Juego

Cada vez que un jugador tiene el turno debe indicar la columna donde hará su jugada.

La estrategia que usará el computador depende del nivel de dificultad de la partida. Se tienen dos niveles:

1. ***Básico***: el computador escoge aleatoriamente la columna a jugar.
2. ***Medio*****:** el computador intenta construir su propia línea: horizontal, vertical o diagonal (a la izquierda o a la derecha), de acuerdo a las fichas ya colocadas. Si esto no es factible, escoge aleatoriamente la columna a jugar y el tipo de línea a construir. Para desarrollar esta estrategia es recomendable que se almacene la última columna y fila jugada por el computador, y el tipo de línea que se está construyendo: horizontal, vertical, diagonal-izquierda o diagonal-derecha. De esta manera el computador puede buscar una casilla vacía cercana a su última jugada, de acuerdo al tipo de línea que se está construyendo.

3. Funcionalidades del programa

Las funcionalidades que debe incluir el programa son las siguientes:

1. Al comenzar, se debe preguntar el nombre de la persona contrincante del computador.
2. El programa debe permitir jugar varias partidas. Para cada partida la persona puede escoger el nivel de dificultad: básico o medio.
3. Al finalizar una partida, la persona puede decidir si desea continuar jugando o no.
4. La primera partida la comienza cualquiera de los jugadores, usted decide la manera de seleccionarlo. El resto de las partidas la debe iniciar el ganador de la partida anterior. En caso de empate, puede comenzar cualquier jugador; igual que antes, usted decide la manera de seleccionarlo.
5. La persona puede abandonar la partida. En este caso se considera ganador a la computadora.
6. El tablero debe desplegarse a través de una interfaz gráfica y cada jugada deberá reflejarse en este tablero.
7. La entrada de datos será a través de la cónsola de comandos.
8. Cuando uno de los jugadores gana, hay que resaltar en el tablero las fichas alineadas.
9. Al finalizar el juego se debe imprimir en la cónsola de comando: el número de partidas realizadas, el número de partidas ganadas por la persona y el número de partidas ganadas por el computador, y el número de empates.

4. Solución algorítmica sugerida

El diseño de la solución lo hacemos aplicando la técnica del análisis descendente, mediante el cual un problema es dividido en varios subproblemas de menor complejidad. El programa involucra dos componentes principales: uno con la lógica del juego y otro con el manejo de la interfaz gráfica.
En primer lugar se recomienda tener separados los dos componentes. La independencia de ambos componentes debe ser tal, que se pueda sustituir la parte gráfica por una salida por consola sin alterar la componente lógica del juego.
A continuación se muestra una propuesta de esquema general de solución algorítmica para el juego Podemos aplicar análisis descendente para obtener una primera versión:

```
[
   CONST
      … declaraciones de constantes ...
   VAR
      … declaraciones de variables ...

   {...precondiciones ...}

   do jugar otra partida ->
      Inicialización partida;
      do se desea seguir jugando y la partida no ha terminado ->
         ObtenerJugada;
         if jugada es válida ->
            ReflejarJugada;
            Cambiar turno
         [] jugada no válida ->
            Error
         fi
      od;
      Desplegar ganador de la partida;
      Finalizar partida
   od;
```

```
   Desplegar resultados de conjunto de partidas
]
```

A partir del análisis descendente, se pueden identificar los subproblemas más importantes: determinar si se desea jugar otra partida, inicializar la partida, determinar si se desea seguir jugando una partida, determinar si la partida no ha terminado, obtener la jugada, determinar si una jugada dada es válida, reflejar la jugada, cambiar el turno del jugador, mostrar resultado de la partida, y mostrar el resultado final.

4.1. Estructuras de datos sugerida

El tablero lógico se puede considerar como una matriz 6x7 de enteros. Cada casilla estará representada por un elemento de la matriz y podrá contener los siguientes valores:

- 0: para indicar que la casilla está vacía
- 1: para indicar que una casilla está ocupada por una ficha de la persona
- 2: para indicar que la casilla está ocupada por una ficha de la computadora

Una jugada puede representarse por una variable *jugada* que indica la columna del tablero. El turno se puede representar con una variable *turno* que toma valor 1 si el jugador es la persona, y 2, si el jugador es el computador. El nivel de dificultad de la partida se puede representar también con una variable *nivel* que toma valores 1 o 2.

4.2. Descripción de Subproblemas

A continuación se describe cada uno de los subproblemas presentes en la solución algorítmica sugerida:

1. **Determinar si desea jugar otra partida.** Debe verificar si el jugador desea jugar otra partida mediante un mensaje en pantalla.

2. **Inicializar partida.** Consiste en: pedir el nombre de la persona mediante un mensaje en pantalla y dibujar el tablero.

3. **Determinar si desea seguir jugando.** Debe verificar si la persona desea seguir jugando la partida mediante un mensaje en pantalla.

4. **Determinar si la partida no ha terminado.** Debe determinar si el tablero ya se llenó o si hay una línea ganadora. Hay una línea ganadora si la última persona en jugar logra 4 fichas consecutivas en linea de su propio color, ya sea vertical, horizontal o diagonalmente. Este subproblema se puede dividir a su vez en cuatro subproblemas: determinar alineación vertical, determinar alineación horizontal, determinar alineación diagonal-izquierda (abajo-derecha hacia arriba-izquierda) y determinar alineación diagonal-derecha (abajo-izquierda hacia arriba-derecha). También, otro subproblema es si hay una línea ganadora, ésta debe estar resaltada.

5. **Obtener jugada.** En el caso de que el turno corresponda a la persona, se debe pedir la jugada por la cónsola de comandos. En caso de que el turno corresponda al computador, se debe desarrollar la estrategia de acuerdo al nivel de dificultad de la partida.

6. **Determinar si la jugada es válida.** Según la descripción del problema deben establecerse las condiciones bajo las cuales una jugada es válida.

7. **Reflejar la jugada.** Este subproblema se subdivide a su vez en:
   - Determinar la fila que se está jugando de acuerdo a las fichas que se encuentran en el tablero.
   - Reflejar la jugada en el tablero lógico correspondiente.
   - Reflejar la jugada de forma gráfica en el tablero.

8. Cambiar de turno. Cambia el jugador que está de turno.

9. Finalizar partida. Muestra quién es el ganador de la partida, y reinicia el tablero en la interfaz gráfica, asícomo el tablero lógico.

10. Mostrar resultado final. Se totaliza el número de juegos ganados por la persona, por el computador y los empatados, y se despliegan estos resultados en la cónsola de comandos.

NOTA IMPORTANTE:
Usted podrá desarrollar mejoras y extensiones al esqueleto general y los subproblemas indicados anteriormente siempre y cuando se describan y documenten con claridad.

5. Interfaz Gráfica

La descripción de los subproblemas anteriores permiten obtener el diseño de un programa que captura la parte lógica o de cálculo del problema dado. Sin embargo, también es necesario incluir instrucciones que manejen la parte de interfaz. Para facilitar la programación, se limitar las operaciones de entrada de datos por teclado al modo texto, mientras que para el despliegue de resultados se utilizarán dos modos de manera conjunta: modo texto (cónsola de comandos) y modo gráfico (interfaz gráfica).

Según los requerimientos del programa, inicialmente se debe solicitar el nombre del jugador. Esto se puede hacer usando una instrucción `input` del lenguaje y almacenarlo en la variable *`nombreJugador`* de tipo string o arreglo de caracteres. Luego, cada vez que se requiera, se puede desplegar un mensaje como:

"Por favor "+ `nombreJugador` + " , introduzca una jugada: "

El `nombreJugador` también se puede usar para dar mensajes en caso de una jugada no válida, para preguntar si el jugador desea volver a jugar, etc.

Para la salida gráfica del programa se utilizará una librería que permita dibujar líneas y círculos para representar el tablero y las jugadas realizadas.

Los subproblemas a resolver en el componente gráfico del programa son:

1. ***dibujarTablero.*** Dibuja un tablero inicial en la ventana o interfaz gráfica.

2. ***reflejarJugadaTablero*****.** Cada vez que se realiza una jugada modifica gráficamente el tablero. Para ello, debe conocer la columna donde se hace la jugada y el jugador a quien le corresponde el turno para colocar la ficha en el color correspondiente.

3. ***resaltarLinea.*** Cuando se detecta una línea ganadora la resalta. Para ello, debe conocer la fila y columna inicial y final de la línea, además el jugador a quien le corresponde el turno.

6. Entregas

El proyecto se desarrollará de forma incremental. Para cada una de las entregas debe especificar las precondiciones y las postcondiciones así como los invariantes representativos de los ciclos del esqueleto principal y de los subprogramas correspondientes.

6.1. Primera entrega

Debe escribir en GCL el algoritmo del programa, utilizando subprogramas (procedimientos y funciones) para cada uno de los subproblemas descritos en este enunciado. Proponga para cada uno las pre y postcondiciones, así como las acciones en GCL que tendría cada subalgoritmo. En el caso de los ciclos, no es obligatorio colocar los invariantes, pero si las funciones de cota.

Para los subprogramas relacionados a la entrada y salida de texto o a la interfaz gráfica, escriba en pseudolenguaje (similar al esqueleto que se presenta en la sección 4) las acciones que debe realizar. Puede suponer que hay instrucciones como

```
dibujarlinea(xinicial,yinicial,xfinal,yfinal,color) y
dibujarcirculo(xinicial,yinicial,xfinal,yfinal,color)
```

También debe entregarse en Python, el esqueleto del programa, el cual sólo tendría las declaraciones, el programa principal y los encabezados de los subprogramas. El cuerpo de estos subprogramas estaría vacío, con sólo una instrucción que retorne un valor arbitrario del tipo correcto o un mensaje indicando la operación que realizan, a fin de probar el flujo de control del programa principal, sin errores de sintaxis o ejecución. Opcional: que el programa dibuje el tablero inicial.

Luego se darán mas detalles de la primera entrega.

6.2. Segunda entrega

El programa en Pythom funcionando con todos los subprogramas planteados. Así como las aserciones correspondientes a precondiciones, postcondiciones, invariantes y cotas.

Posteriormente, se dará mas detalle de esta entrega.