Práctica 2 FP1 2019
Dominó V2

Fecha de entrega: 24 de Noviembre de 2019.

1. Descripción de la funcionalidad de Dominó V2.

En la segunda versión vamos a utilizar dos arrays de strings para almacenar las fichas del jugador y las fichas que se pueden robar de la mesa (llamado pozo, inicialmente con las 28 fichas posibles).

Inicio del juego:

Se debe generar el pozo con las 28 fichas posibles desordenadas.

Se roba al azar una ficha del pozo y se coloca en el tablero.

El jugador roba al azar 7 fichas del pozo.

Las fichas recogidas del pozo se eliminan del pozo.

Desarrollo del juego:

Antes de cada jugada se muestran las fichas colocadas en la mesa (tablero), el número de fichas colocadas por el jugador, el número de fichas robadas por el jugador y las fichas del jugador.

Las opciones que puede elegir el jugador en su turno son:

1) Colocar una de sus fichas a la **izquierda** del tablero. En este caso se pregunta cuál de las fichas del jugador se desea colocar a la izquierda del tablero.

Sólo se puede colocar la ficha si uno de sus valores coincide con el valor del extremo izquierdo del tablero.

Si se coloca la ficha debe representarse en el tablero en la posición correcta.

La ficha colocada se elimina de la lista de fichas del jugador.

2) Colocar una de sus fichas a la **derecha** del tablero. En este caso se pregunta cuál de las fichas del jugador se desea colocar a la derecha del tablero.

Sólo se puede colocar la ficha si uno de sus valores coincide con el valor del extremo derecho del tablero.

Si se coloca la ficha debe representarse en el tablero en la posición correcta.

La ficha colocada se elimina de la lista de fichas del jugador.

3) Robar una ficha del pozo aleatoriamente que se elimina del pozo y se añade a la lista de fichas del jugador.

No se permite robar si es posible colocar alguna ficha del jugador en el tablero.

4) Salir del juego. En este caso se muestran los puntos del jugador (la suma de los dos valores de todas sus fichas) y se termina el programa.

La Figura 1 muestra un ejemplo de ejecución del programa.

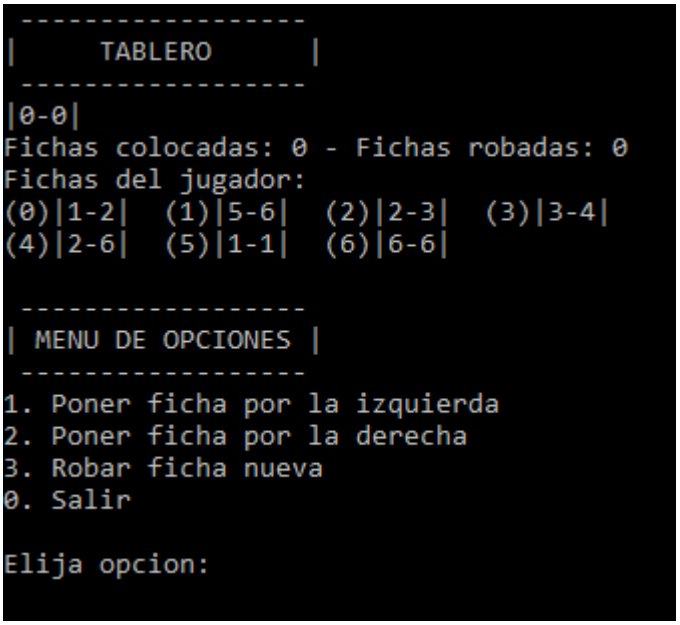

Figura 1: Ejemplo de ejecución

Final del juego:

Una ronda de dominó termina en uno de los siguientes casos:

1) Si el jugador coloca todas sus fichas.

2) Si no quedan fichas para robar en el pozo y el jugador no puede poner ninguna ficha. En este caso se deben mostrar los puntos del jugador antes de salir.

3) Si se selecciona la opción de Salir

Funcionalidades opcionales:

- Salvar la partida en un archivo, para cargarla más adelante y continuar con ella.
- Permitir que el programa se pueda configurar para jugar con otras variantes del juego: entre 6 y 9 como máximo de puntos de las fichas.

2. Detalles de implementación de la versión 2

En la versión 2 el pozo y las fichas del jugador se almacenan en dos arrays de strings (`pozo` y `fichasJug`). Asimismo, dispondremos de dos contadores para saber cuántas fichas tiene el jugador (`fichasCont`) y cuántas fichas hay en el pozo (`pozoCont`). El tablero se sigue almacenando en un string.

```
const int NUM_FICHAS = 28;
typedef string tArray[NUM_FICHAS];
```

Además, deberás incorporar los subprogramas necesarios decidiendo el tipo de retorno, el tipo de cada argumento y si cada argumento se pasa por valor o por referencia. Algunas de dichas funciones podrían ser las siguientes:

```
//Genera en pozo todas las piezas posibles del dominó para
fichas con valores de 0 a maxPuntos:
void generaPozo(tArray pozo, int maxPuntos);

//Desordena el pozo:
void desordenaPozo(tArray pozo);

//Escribe en pantalla toda la información de la partida,
incluyendo las fichas del jugador:
void mostrarTablero(tArray fichasJug, short int fichasCont,
string tablero, short int numColocadas, short int
numRobadas);

//Quita la última ficha del pozo y la devuelve:
string robarFicha(tArray pozo, short int pozoCont);

//Quita de las fichas del jugador su ficha fichaNum-ésima
desplazando el resto de posiciones a la izquierda para
rellenar el hueco dejado por la ficha quitada:
void eliminaFicha (tArray fichasJug, short int fichasCont,
short int fichaNum);

//Suma los puntos acumulados en todas las fichas del jugador:
short int sumaPuntos(tArray fichasJug, short int fichasCont);
```

El resto de funciones de la versión 1 deben adaptarse para el correcto funcionamiento del programa usando arrays.

Ayuda: Para desordenar el pozo podemos utilizar el algoritmo de Fisher-Yates, que desordena cualquier array de manera no sesgada en un recorrido:

```
void desordenaPozo(tArray pozo) {
     int idx1, idx2;
     string tmp;

     for (int i = 0; i < 1000; i++) {
          idx1 = rand() % NUM_FICHAS;
          idx2 = rand() % NUM_FICHAS;
          if (idx1 != idx2) {
               tmp = pozo[idx1];
               pozo[idx1] = pozo[idx2];
               pozo[idx2] = tmp;
          }
     }
}
```