



Fundamentos de la Programación

Presentación de la Práctica 1 (Versión 3)

(Basado en la práctica de Mercedes Gómez, Luis Hernández, Ramón González y Federico Peinado)

Índice

1. Introducción
2. Funcionalidad de la Aplicación
 - 2.1. Versión 3
3. Detalles de Implementación
 - 3.1. Versión 3

1. Introducción

- ✓ En esta práctica seguimos trabajando con el juego **las siete y media**.
- ✓ La versión añade a la práctica la utilización de **estructuras y listas de longitud variable**.

2. Funcionalidad de la Aplicación (Versión 3)

- ✓ Cada versión avanza en objetivos y temario del curso.
- ✓ En esta tercera y última versión:
 - Se lleva la cuenta del **número de partidas** que se juegan en la ejecución correspondiente.
 - El menú permite seleccionar un cuarto modo de juego: el **modo D**.

✓ En este modo D:

- El **mazo de cartas** no está en un archivo sino que lo genera el programa para cada partida y se mantiene en memoria. Inicialmente este mazo cuenta con las 40 cartas dispuestas aleatoriamente y va reduciéndose a medida que las cartas son robadas por los jugadores.
- Se mantiene en memoria no sólo la puntuación que consigue cada jugador (actualizada a medida que roba cartas) sino también las cartas que roba. Así, cada vez que roba se muestran todas las cartas que tiene en su haber y no sólo la que acaba de robar.
- El comportamiento de los jugadores es igual al del modo C.
- En caso de que se produzca un empate no se recurre por defecto al azar para determinar el ganador. El ganador es aquel que ha conseguido la puntuación con un menor número de cartas. Si el número de cartas es igual se determina al azar.
- El ganador, la puntuación y las cartas robadas por cada jugador se almacena en un archivo cuyo nombre coincide con el número de la partida.

3. Detalles de Implementación (Versión 3)

- ✓ A continuación se dan algunas indicaciones para implementar la tercera versión.
- ✓ El **archivo de resultados** que se genera al jugar una partida en el modo D contendrá 4 líneas: la primera con el número de partida, la segunda con el ganador, la tercera con la puntuación y las cartas del humano, y la cuarta con la puntuación y las cartas de la máquina.
- ✓ Incluye un **tipo estructura** *tConjuntoCartas* que defina listas de hasta 40 cartas y cuyas variables servirán para representar tanto el mazo como las cartas robadas por el jugador humano y por la máquina.

- ✓ En el modo D, cada partida debe comenzar con un mazo en memoria que contiene 40 cartas dispuestas de forma aleatoria. Una forma de conseguirlo es rellenar el mazo inicialmente con las 40 cartas disponibles y después barajarlas.
- ✓ Incorpora, al menos, los siguientes subprogramas relacionados con el mazo (lista) de cartas:
 - `void inicializa(tConjuntoCartas &mazo) // Inicializa mazo a la lista vacía.`
 - `void crearMazo(tConjuntoCartas &mazo) // Rellena mazo con 40 cartas dispuestas de forma aleatoria.`
 - `void sacarCarta(tConjuntoCartas &mazo, int &carta) // Elimina una carta de un extremo de la lista mazo y la devuelve en carta.`
 - `void annadirCarta(tConjuntoCartas &mazo, int carta) // Añade la carta carta en un extremo de la lista mazo.`

✓ Para el modo D, implementa al menos los siguientes subprogramas:

- `float modoDhumano (tConjuntoCartas &mazo, tCartasPorAparecer cartas, tConjuntoCartas &cartasHumano) //` Permite realizar el turno del jugador humano en el modo D. Recibe el mazo y una variable `cartas` que indica cuántas cartas de cada tipo quedan en el mazo; al finalizar su ejecución deja `cartas` y `mazo` actualizados según las cartas que haya robado el humano en su turno y devuelve los puntos obtenidos por el humano así como las cartas que ha robado.
- `float modoDmaquina(tConjuntoCartas &mazo, tCartasPorAparecer cartas, tConjuntoCartas &cartasMaquina, float puntosHumano) //` Permite realizar el turno del jugador máquina en el modo D. Recibe el mazo, una variable `cartas` que indica cuántas cartas de cada tipo quedan en el mazo y la puntuación obtenida por el jugador humano; al finalizar su ejecución deja `cartas` y `mazo` actualizados según las cartas que haya robado la máquina en su turno y devuelve los puntos obtenidos por la máquina así como las cartas que ha robado.