

Exercici L4

Memory

Fonaments de Programació II – Curs 2024-25

Objectiu

L'objectiu d'aquesta pràctica és que trebal·leu amb fitxers i també amb gestió dinàmica de la memòria. Per fer-ho implementareu una versió simplificada del joc *Memory* en una sola dimensió.

Descripció del joc

El *Memory* és un joc clàssic en què les cartes es col·loquen cap per avall i els jugadors les destapen per parelles intentant trobar dues imatges iguals. En aquesta versió, en comptes d'imatges, es fa servir un **vector de caràcters**, on cada lletra apareix **exactament dues vegades en posicions aleatòries**.

El jugador ha de trobar totes les parelles seguint aquestes regles:

- A cada torn, selecciona **dues posicions** dins del vector.
- Si les lletres seleccionades són **iguals**, es marquen amb el caràcter '-' per indicar que han estat descobertes.
- Si no són iguals, es demana a l'usuari que torni a introduir dues posicions vàlides.
- El joc **finalitza quan totes les parelles han estat trobades**.

El programa permet **seleccionar entre dos nivells de dificultat**:

- **Nivell fàcil**: es llegeixen les seqüències de lletres del fitxer `facil.txt`.
- **Nivell difícil**: es llegeixen seqüències del fitxer `dificil.txt`.

Comportament del programa

- Quan el programa s'executa, el primer que farà és preguntar a l'usuari, a través d'un menú de text, amb quin nivell vol jugar (fàcil o difícil). Depenent d'aquesta resposta, s'obrirà el fitxer `facil.txt` o bé el `dificil.txt`. Mentre l'usuari triï una opció inexistent, se li seguirà preguntant.

- Un cop obert el fitxer es carregarà el tauler. Dins del fitxer hi ha múltiples taulers possibles, se'n triarà un d'ells aleatòriament.
- Un cop el tauler estigui carregat, s'imprimirà per pantalla, representant cada casella amb una lletra si encara no ha estat descoberta o amb '-' si ja s'ha trobat la seva parella.
- A continuació es demanarà a l'usuari que introdueixi dues posicions enteres, en el rang [1, mida_tauler].
- Si l'usuari ha seleccionat dues lletres iguals, es marcaran amb un caràcter '-' i s'imprimirà un missatge d'èxit. Si no coincideixen, es mostrarà un missatge d'error i es demanarà una nova entrada. Si l'usuari introdueix una entrada no vàlida (una posició fora del rang, una casella ja descoberta, etc.), el programa li mostrarà un missatge i li demanarà que ho intenti de nou.
- A cada iteració es tornarà a mostrar el tauler.
- El joc continua fins que **totes les parelles han estat descobertes**. Quan això passi, es mostrarà un missatge indicant que el jugador ha completat la partida.

Requisits de la implementació

Lectura del vector des d'un fitxer de text.

Per permetre **diferents nivells de dificultat**, es disposarà de dos fitxers de text (`facil.txt` i `dificil.txt`). Aquests fitxers tenen el següent format:

- La primera línia indica el **nombre de parelles** amb el format:
`NUM_PARELLES = X`
on `X` és un nombre enter que indica quantes parelles hi ha.
- Les línies següents són **configuracions de tauler** vàlides, amb `X*2` caràcters separats per espais. Hi pot haver tantes línies com calgui.

Un possible exemple del fitxer `facil.txt` seria:

```
NUM_PARELLES = 4
A A B B C C D D
A B B A C D D C
C D A C D A B B
D D C C B B A A
A D C C A B D B
```

- Els fitxers de **categoria difícil** (`dificil.txt`) tindran **més parelles per seqüència**.

- Per conèixer la mida de la taula a reservar, **el programa ha de llegir aquest valor del fitxer** (no es pot pressuposar que són 4 o 8 parelles).
- Per seleccionar un tauler per a la partida, **es compten les línies disponibles i se'n tria una de manera aleatòria**. *Compte que la primera no compta perquè és la que conté la informació del nombre de parelles!*

Emmagatzematge usant gestió dinàmica de la memòria

Com que no sabem el nombre de parelles (i per tant, no sabem la mida del vector) fins que no llegim el fitxer (és a dir, en temps d'execució), haurem de fer servir gestió dinàmica de la memòria per emmagatzemar aquestes dades.

L'estructura haurà de ser una taula de caràcters de mida `X * 2`, amb la corresponent gestió de la memòria. És a dir, un cop sapiguem el nombre de parelles, reservarem el vector de la mida corresponent, farem el control que la memòria s'hagi reservat bé i el tractament de l'error si fos el cas, i l'omplirem de continguts. Al finalitzar el programa s'haurà d'alliberar la memòria.

Estructura modular del codi

Com que aquest problema té una certa complexitat i moltes tasques diferents, es farà servir un disseny modular per estructurar el codi. Encapsuleu aquelles funcionalitats similars en procediments.

Validació d'entrades de l'usuari

Per garantir que el joc funcioni correctament, el programa ha de comprovar:

- Que les posicions seleccionades són dins del rang `[1, mida_tauler]`.
- Que les dues posicions són diferents (no es pot seleccionar dues vegades la mateixa casella).
- Que les caselles seleccionades no hagin estat ja descobertes ('-').
- Que el fitxer s'ha obert correctament, i que el nombre de parelles es llegeix sense errors.

Gestió d'errors

- Si l'usuari introdueix una posició fora de rang, es mostra un missatge d'error i es demana una nova entrada.
- Si selecciona una posició ja descoberta ('-'), també es demana una nova entrada.

- Si el fitxer no es pot obrir, el programa atura l'execució amb un missatge d'error (`exit(EXIT_FAILURE)`) per evitar comportaments inesperats.

Si acabes aviat...

Fins ara, les lletres es mostraven directament per facilitar les proves durant el desenvolupament. Ara adapta el joc perquè es pugui jugar correctament: quan imprimeixis el tauler, totes les caselles han d'aparèixer **emmascarades** amb 'X', i només es descobriran quan el jugador encerti una parella (en comptes de marcar-les amb '-').

Per implementar aquest canvi, modifica la funció `imprimir_tauler` afegint-hi un **paràmetre booleà** que indiqui si el tauler ha d'estar emmascarat o no:

- **Si el tauler està emmascarat**, totes les caselles han de mostrar una 'X' en lloc del seu contingut.
- **Si no està emmascarat**, es mostren les lletres tal com estan emmagatzemades.

Si encara vols practicar més (bona idea!), fes que un cop la partida s'acabi, es demani a l'usuari si en vol jugar una altra. Això implica tornar-li a demanar la dificultat, alliberar la memòria prèvia, al·locar memòria de nou... etc.