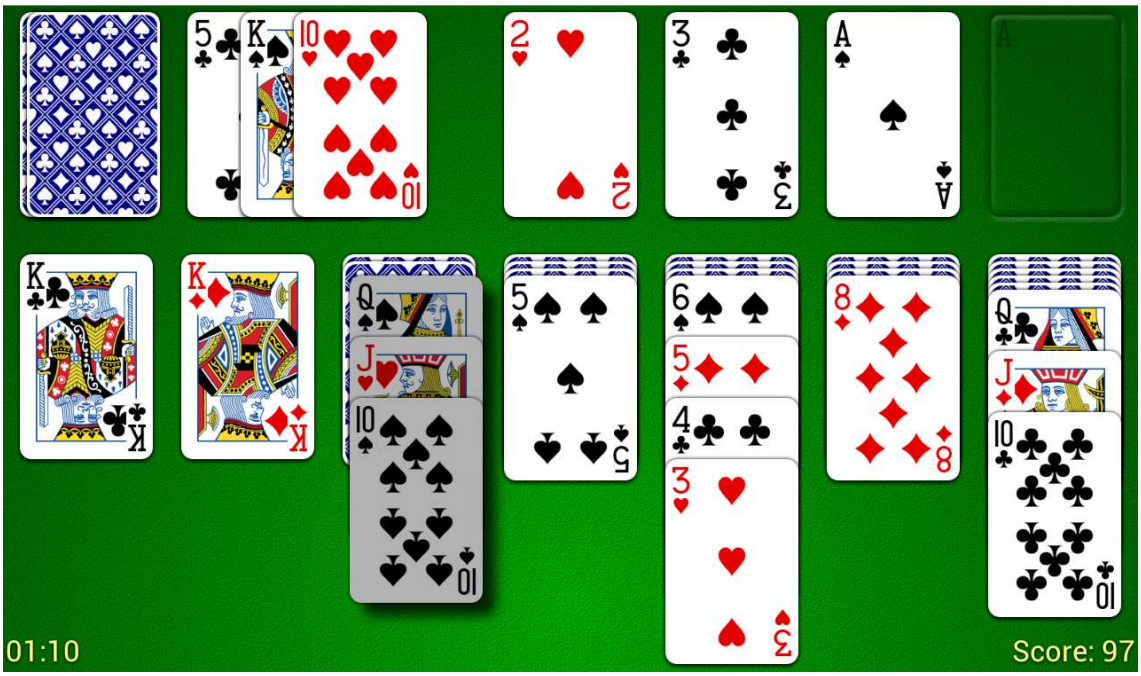
Pràctica Final MTP2: Solitari



Marc Cané Salamià

1939666

Grup lab. 4

Prof. lab. Miquel Feixas

Curs 2015/16

Descripció disseny:

El joc del solitari m'ha servit molt per veure un ús real de les classes, perquè servèixen, com usarles, en quin tipus de problemes es poden aplicar...

Quan tens diferents tipus de dades que s'allunyen de les predeterminades de C++ va bé tenir maneres d'agruparles, pero si a més aquestes dades es comporten de manera molt diferent a les predeterminades també va bé poguer definir les operacions que aquestes poden fer.

L'orientació a objectes obre un nou món de possibilitats als programes fets en llenguatges d'alt nivell, facilitant la programació i la comprensió del codi.

Descripció classes:

La classe baralla l'hem usat bàsicament per barrejar les cartes i repartirles per el tauler i la mà.

La classe joc és la que conté el bucle principal i totes les dades del joc, les piles i el tauler. Es on hi ha la majoria de codi que agafa informació de les altres classes i fa les operacions necessàries perquè el joc es vagi actualitzant.

La classe tauler és una matriu dinàmica de dades que conté les cartes que tenim a la taula, i ens ajuda a abstreuren's de si les files comencen a 0 o a 1.

La classe carta és la que guarda quin valor té cada carta i de quin pal és, juntament amb si està girada o no.

Finalment la classe pila ens ajuda a representar les piles de cartes com la ma i els pals.

Metodes publics:

Baralla: //Pre: Index>=0 i index<52 //Post: Retorna la carta que es troba en aquella posicio de la baralla

```
Carta Get_Carta(int index)const;
```

Joc: //Pre: -- //Post: Mostra l'estat del joc

```
void mostrar();

//Pre: -- //Post: Comença el joc

void Jugar();
```

Tauler:

//Pre: -- //Post: Mostra el tauler

void mostrar()const;

//Pre: 0<col<8 //Post: Retorna la ultima carta de la columna col o bé la "fila" carta de la columna col

Carta Get_Carta_Tauler(int col, int fila=-1)const;

//Pre: 0<col<8 //Post: Inserteix la carta inserir a la ultima posicio de la columna col

int Mida_Col(int col)const;

//MODIFICADORS

//Pre: 0<col<8 //Post: Treu una carta de la columna col

void Inserir_Carta_Tauler(Carta inserir, int col);

//Pre: 0<col<8 //Post: Retorna la mida de la columna col

void Treure_Carta(int col);

//Pre: 0<col<8 //Post: Obre la ultimam carta de la columna col

void Obrir_ultima(int col);

//Pre: Baralla amb 52 cartes //Post: Reparteix les primeres 28 cartes al tauler

void repartir(Baralla);

Carta:

//Pre: -- //Post: Mostra la carta si aquesta està oberta o bé mostra dos asteriscs si és tancada

void mostrar()const;

//Pre: -- //Post: Retorna cert si la carta es de color negre

bool esnegra()const;

//Pre: -- //Post: Retorna cert si la carta és oberta

bool esoberta()const;

//Pre: -- //Post:Retorna el pal de la carta

char Get_Pal()const;

//Pre: -- //Post: Retorna cert si la carta entrada casa amb la actual(gran)

bool Casen(Carta, bool espila=0)const;

//Pre: -- //Post:Retorna cert si la carta és rei

bool EsRei()const;

//Pre: -- //Post:Retorna cert si la carta té el valor A

bool EsA()const;

//MODIFICADORS

//Pre: -- //Post: La carta es modifica amb les dades entrades

void Modificar(char valor, char pal);

//Pre: -- //Post: Els elements intercanviats

void Intercanviar(Carta &b);

//Pre: -- //Post: Modifica la carta i passa a estar en l'estat que hem entrat

void Set_obrir(bool obrir);

Pilacartes: // Pre: -- ; Post: retorna cert si la pila es buida; fals en c.c.

bool buida() const;

// Pre: pila no buida; Post: retorna el valor del cim de la pila

Carta cim() const;

// MODIFICADORS ----------

// Pre: --; Post: ha afegit s a dalt de la pila

void empila(Carta s);

// Pre: pila no buida; Post: ha eliminat element de dalt de la pila

void desempila();

// Pre: -- ; Post: Les cartes de la pila girades

void Girar_Cartes(PilaCartes &b);

// Pre: -- ; Post: Retorna cert si la pila té el rei al cim

bool Pila_Completada();

Decisions de disseny:

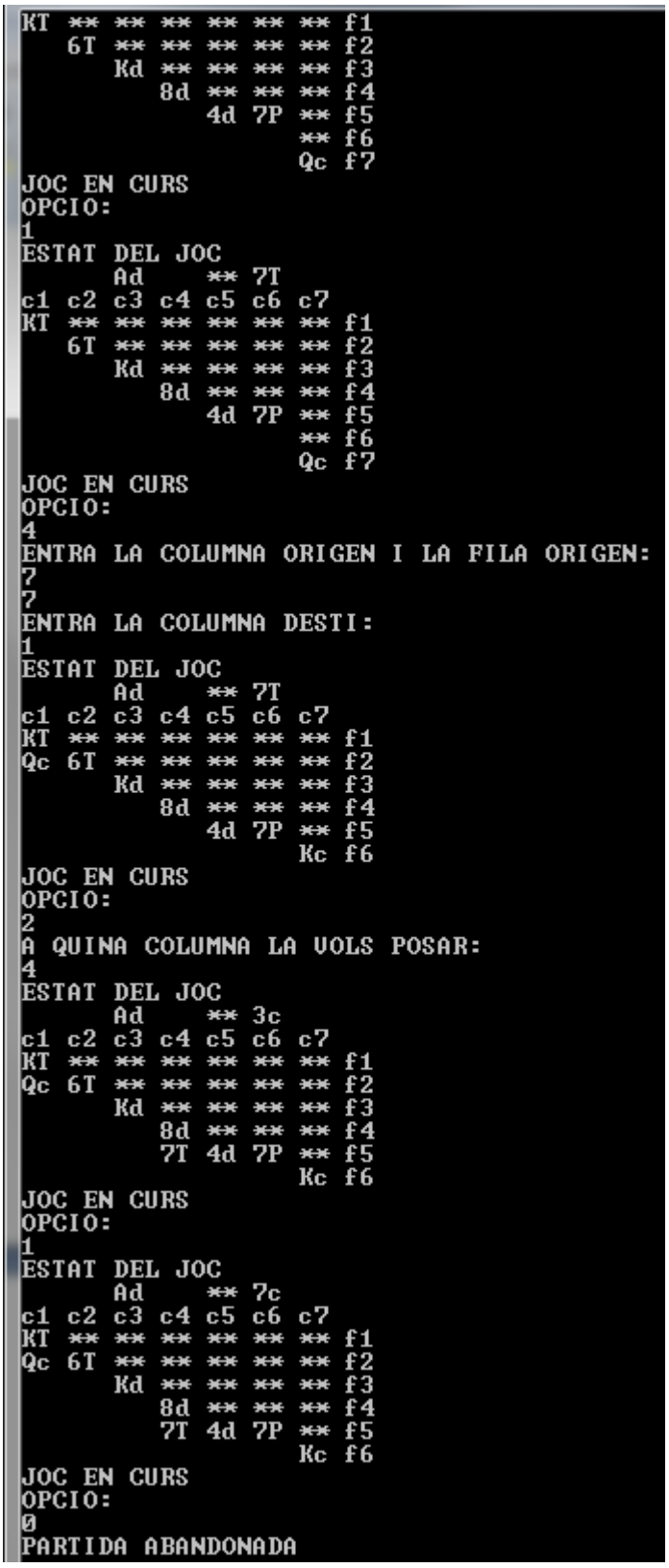
Vaig optar per definir el valor com un char per simplificar el metode mostrar() i no haver de comprovar cada cop si era un numero o no. Tot i que aixo em va dificultar fer els metodes com per exemple el "Casen" crec que es una opció correcta perquè estalvies algo de feina a la màquina, i tot i que amb els ordinadors d'avui dia no tenim problemes de potència, des del punt de vista de la programació és molt més correcte.

La baralla havia pensat ferla com una pila pero ràpidament men vaig adonar els problemes que comportava al intentar barrejar les cartes i vaig decidir canviarla a un vector.

A l'inici també havia pensat tenir un caràcter per poder saber de quin tipus era cada pila, pero al final va resultar ser poc util i poc pràctic i el vaig acabar treient.

Dificultats: La manera de repartir les cartes. Les piles buides. Les estructures de dades de C que comencen per 0 mentre que les columnes i les files comencen per 1. Moure múltiples cartes. Fer visible una carta del cim de la pila (i no a la seva copia). La funció per determinar si casen o no les cartes (i fer que no repetís codi amb quan estàs mirant si casen les cartes d'una pila).

Exemple execució:



```
KT ** ** ** ** ** ** f1
   6T ** ** ** ** ** f2
      Kd ** ** ** ** f3
         8d ** ** ** f4
            4d 7P ** f5
                  ** f6
                  Qc f7
JOC EN CURS
OPCIO:
1
ESTAT DEL JOC
      Ad    ** 7T
c1 c2 c3 c4 c5 c6 c7
KT ** ** ** ** ** ** f1
   6T ** ** ** ** ** f2
      Kd ** ** ** ** f3
         8d ** ** ** f4
            4d 7P ** f5
                  ** f6
                  Qc f7
JOC EN CURS
OPCIO:
4
ENTRA LA COLUMNA ORIGEN I LA FILA ORIGEN:
7
7
ENTRA LA COLUMNA DESTI:
1
ESTAT DEL JOC
      Ad    ** 7T
c1 c2 c3 c4 c5 c6 c7
KT ** ** ** ** ** ** f1
Qc 6T ** ** ** ** ** f2
      Kd ** ** ** ** f3
         8d ** ** ** f4
            4d 7P ** f5
                  Kc f6
JOC EN CURS
OPCIO:
2
A QUINA COLUMNA LA VOLS POSAR:
4
ESTAT DEL JOC
      Ad    ** 3c
c1 c2 c3 c4 c5 c6 c7
KT ** ** ** ** ** ** f1
Qc 6T ** ** ** ** ** f2
      Kd ** ** ** ** f3
         8d ** ** ** f4
         7T 4d 7P ** f5
                  Kc f6
JOC EN CURS
OPCIO:
1
ESTAT DEL JOC
      Ad    ** 7c
c1 c2 c3 c4 c5 c6 c7
KT ** ** ** ** ** ** f1
Qc 6T ** ** ** ** ** f2
      Kd ** ** ** ** f3
         8d ** ** ** f4
         7T 4d 7P ** f5
                  Kc f6
JOC EN CURS
OPCIO:
0
PARTIDA ABANDONADA
```