JOC DEL MEMORY

Autor: Nadal Llabrés Belmar

Vídeo explicatiu

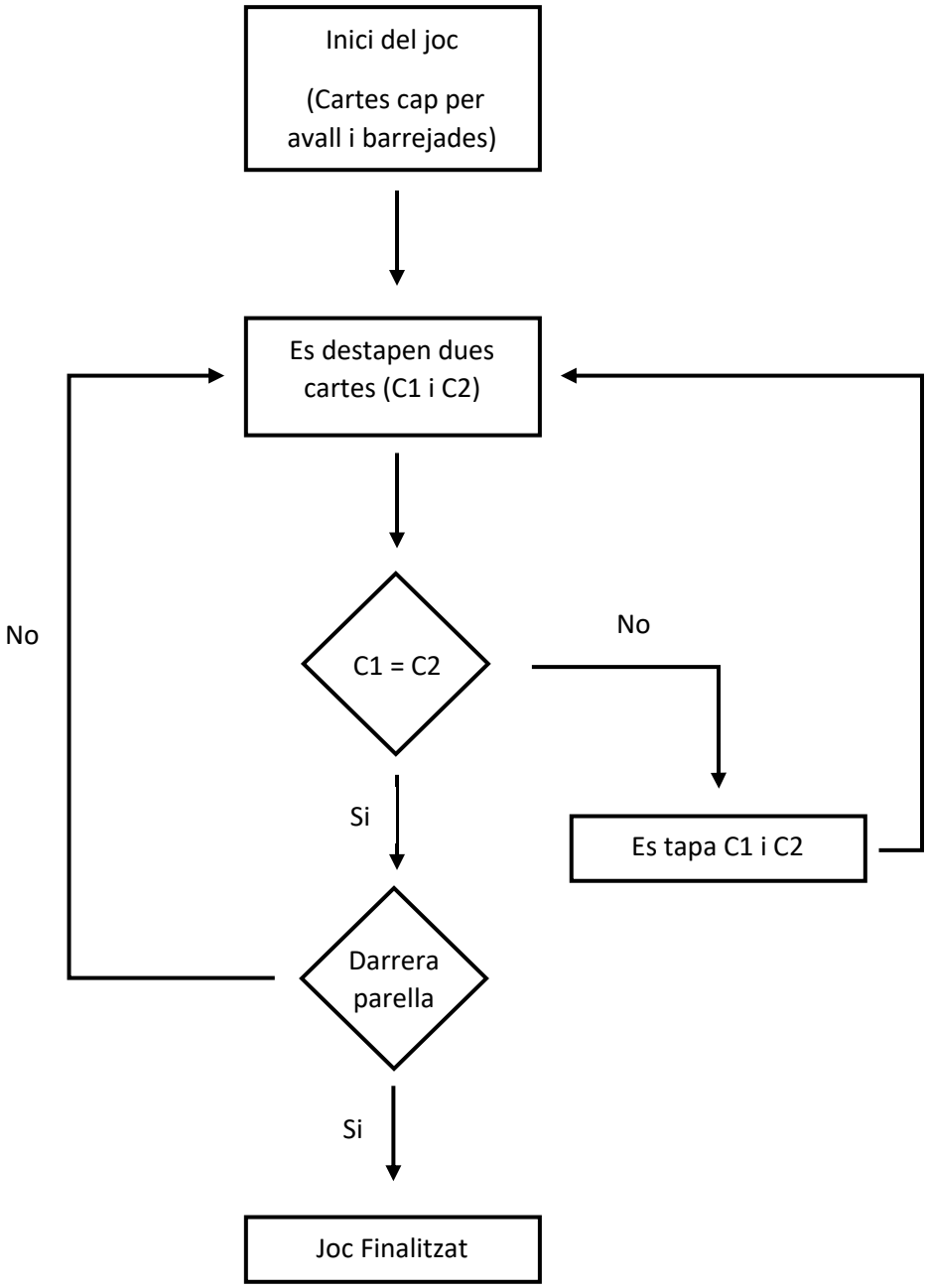
Assignatura: Programació II

Professor: Miquel Mascaró Oliver

Índex

1. Introducció

El joc del *Memory* consisteix amb tenir un nombre determinat de cartes cap per avall damunt un tauler. Cada carta té una parella i l'objectiu es trobar-la. És un joc d'un únic jugador, una vegada s'hagin trobat totes les parelles el joc haurà finalitzat. El procés que es segueix és el següent:



El *Memory* que es programarà a aquesta pràctica tindrà un tauler de 4 x 4 cartes i s'aniran escollint cada una d'elles amb el ratolí. El joc disposarà d'un menú a la part superior de la finestra on estaran implementades les següents opcions:

Inicialitzar

Col·loca totes les cartes cap per avall ordenades de manera que si A1 i A2 són parella, A1 estarà a la posició (0, 0) i A2 a la posició (0, 1) i així amb totes les parelles.

Escapçar

Barreja totes les cartes i les deixa col·locades cap per avall. (Es farà ús de l'algoritme *Fisher-Yates*).

Destapar tot

Destapa totes les cartes del tauler sense alterar l'ordre.

Tapar tot

Tapa totes les cartes del tauler sense alterar l'ordre.

Sortir

Acaba el joc.

2. Disseny

La pràctica està composada per una sèrie de classes i mètodes que complementen tot el disseny descendent i l'estructura de dades. Les classes utilitzades són les següents:

- Memory (Classe principal).
- Tauler.
- Carta.
- Imatge.

A continuació es farà una explicació de cada una de les classes i els mètodes que les componen.

2.1. Memory (Classe principal).

La classe principal és una finestra que implementa *MouseListener* i conté tota la configuració relacionada amb les característiques d'aquest *JFrame*. Els atributs són:

Atributs de menú:

- private JMenuBar jmBar;
- private JMenu jmJoc;
- private JMenuItem jmItemInicialitzar;
- private JMenuItem jmItemEscapçar;
- private JMenuItem jmItemTaparTot;
- private JMenuItem jmItemDestaparTot;
- private JMenuItem jmItemSortir;
- private JToolBar jtBar;

Altres atributs:

- private Tauler tauler;
- private int numeroCartesGirades = 0;
- private int[] coordenades = new int[2];

Té definits els següents mètodes:

Constructor

Crea un nou tauler i ajusta la configuració de la finestra. També inicialitza l'array "cartesAparellades".

private void initComponents()

Conté totes les instruccions relacionades amb les components del menú superior i la finestra. També hi ha implementats els escoltadors de cada ítem de la barra de menús.

Mètode *main*

Crea una nova finestra i la fa visible, també s'afegeix una instrucció anomenada *setLookAndFeel()* que el que fa es detectar el sistema operatiu on s'executa el programa i adaptar la barra de menú a l'estil del SO.

Mètodes del *MouseListener*

- *mouseClicked*
- *mousePressed*
- *mouseReleased*
- *mouseExited*

Aquests mètodes els fa implementar l'extensió de *MouseListener* de forma obligatòria, però en aquest cas només es farà ús del *mouseReleased.*

public void mouseReleased(MouseEvent e)

Implementa tot el procés encarregat de detectar on s'ha fet el clic i executar les accions corresponents depenent de si s'ha trobat la parella o no, o si el joc ja ha acabat.

public void resetParelles()

Inicialitza totes les components de l'array "cartesAparellades" a *false.* La cridada a aquest mètode es fa just abans de començar a jugar o després de fer clic a qualssevol opció del menú.

public boolean jocAcabat()

Retorna un booleà. Si totes les components de l'array "cartesAparellades" estan *true* retorna *true*, en cas contrari retorna *false*. S'utilitza per saber si el joc ja ha acabat o no.

2.2. Classe Tauler

És un panell que conté els mètodes encarregats de construir l'estructura matricial del tauler.

Els seus atributs són:

- **DIMENSIO (int):** Constant pública que determina el nombre de cartes per costat que tindrà el panell. És pública perquè es necessita la dimensió per el programa principal abans de crear l'objecte tauler.
- **MAXIM (int):** Constant privada que indica el tamany del costat del panell amb píxels.
- **COSTAT (int):** Constant privada que representa el tamany amb píxels del costat de la carta.
- **Carta (Carta):** Array bidimensional privat que s'utlitza per referir-se a les posicions del tauler (cartes).
- **img (Imatge):** Objecte Imatge que s'empra per assignar imatges a les caselles (cartes).

Mètodes que conté:

Constructor

Fa una cridada al mètode "inicialitzar()" per començar amb les cartes ordenades.

public void paintComponent(Graphics g)

Realitza un procés iteratiu que recorre tot el tauler i crida al mètode del paintComponent() de cada carta per assignar la imatge pertinent.

public Dimension getPreferredSize()

Retorna un objecte *Dimension* que conté les mides "x" i "y" del tauler.

public void destaparTot()

Mètode iteratiu que recorre tot el tauler posició per posició i va destapant cada carta. Una vegada totes destapades, es fa la cridada al mètode *repaint()* perquè es faci un repintat de tot el panell.

public void taparTot()

Recorre de forma iterativa totes les posicions del tauler i va tapant totes les cartes. Una vegada acabat el procés iteratiu realitza un repintat del panell sencer cridant al mètode *repaint()*.

public boolean dinsCasella(int i, int j, int x, int y)

Retorna un booleà. Si les coordenades "x" i "y" es troben dins la casella del tauler "i" i "j" retornarà *true*, sinó, *false.*

public boolean cartaDestapada(int i, int j)

Retorna un booleà. Si la que es troba a les coordenades passades per paràmetre retorna *true,* en cas contrari, retorna *false.*

public void destapa(int i, int j)

Destapa la carta que es troba a la posició especificada a la part paramètrica.

public void tapa(int i, int j)

Tapa la carta que es troba a la coordenada enviada per paràmetre.

public int getDimensio()

Retorna un valor (int) corresponent a l'atribut DIMENSIO.

public Rectangle getRectangle(int i, int j)

Retorna el Rectangle ubicat a la posició que s'indica a la part paramètrica del mètode.

public int getIDCarta(int i, int j)

Retorna l'identificador (int) corresponent a la carta que es troba a la posició especificada per paràmetre.

public void inicialitzar()

Col·loca totes les cartes amb ordre i cap per avall. (Opció "inicialitzar" del menú superior del joc).

public void escapçar()

Mitjançant l'algoritme *Fisher-Yates*, assigna les imatges a les cartes de manera que quedin desordenades i tapades. L'idea d'aquest algoritme es repartir les imatges d'una forma aleatòria.

public boolean taulerDestapat()

Retorna un booleà. Realitza un procés iteratiu comprovant si cada carta del tauler està destapada o no. En el cas de que totes estiguin destapades es retornarà *true*, en cas contrari, *false.*

2.3. Classe Carta

Gestiona totes les característiques que componen una carta. Els seus atributs són els següents:

- **Rectangle:** Objecte Rectangle2D.Float, privat.
- **Destapada:** Variable booleana privada que serveix per saber si la carta esta cap per amunt o cap per avall.
- **Imatge:** Objecte Imatge privat que serà la imatge que tindrà la carta assignada.
- **REVERS:** Objecte Imatge privat i constant que emmagatzema la part posterior de la carta.

Els mètodes que conformen la classe són:

Constructor (Rectangle2D.Float r)

Assigna l'estat *false* a l'atribut "destapada", el rectangle que es passa per paràmetre s'assigna a l'atribut rectangle i a l'atribut "REVERS" es posa la ruta de la imatge corresponent a la part posterior de la carta.

public void paintComponent(Graphics g)

Comprova si la carta està cap per amunt o cap per avall i després crida al paintComponent de la classe Imatge per dibuixar la imatge corresponent. Si la carta està cap per amunt dibuixarà la imatge que estigui emmagatzemada a l'atribut imatge, sinó dibuixarà la part posterior de la carta.

public void setImatge(Imatge imatge)

Assigna a l'atribut imatge l'objecte Imatge que s'ha enviat per la part paramètrica.

public void destaparCarta()

Canvia l'estat de l'atribut booleà "destapada" a *true.*

public void taparCarta()

Canvia l'estat de l'atribut booleà "destapada" a *false.*

public Rectangle2D.Float getRec()

Retorna el Rectangle que es correspon amb l'atribut rectangle.

public Imatge getImg()

Retorna l'atribut "imatge".

public boolean estaDestapada()

Retorna un booleà. Concretament l'estat de l'atribut "destapada".

2.4. Classe Imatge

Carrega les imatges provinents de memòria secundària i assigna un identificador a cada imatge per poder emparellar-les. Els atributs de la classe són:

- **bi:** Objecte BufferedImage privat on es carregarà la imatge de memòria secundària.
- **ID:** Identificador privat d'imatge (int).

Els mètodes que formen la classe són:

Constructor 1 (int nomImatge, int ID)

Quan es crea el nou objecte s'ha d'enviar per paràmetre un nom d'imatge corresponent a un valor enter i un altre enter que serà l'ID d'aquesta imatge.

Constructor 2 (String nomImatge)

Únicament s'ha d'enviar per paràmetre una cadena de caràcters que equival a la ruta d'on es troba l'arxiu d'imatge a memòria secundària.

public void paintComponent(Graphics g, float x, float y)

Fa el dibuixat de la imatge amb el mètode *drawImage* a la posició que s'ha especificat als paràmetres.

public int getID()

Retorna un valor enter corresponent a l'ID de la imatge.

3. Conclusions

Durant el desenvolupament de la pràctica s'ha anat veient la importància de crear una estructura de dades correcte.

Un dels grans problemes que varen sorgir durant el desenvolupament va ser degut a tenir una estructura de dades errònia. Degut a això, la creació de les accions del joc es varen començar a complicar molt fins a arribar a un punt d'estancament i haver de començar de nou per crear una estructura de dades correcta. Després d'haver començat de nou i tenir una estructura de dades adequada i un bon disseny descendent les coses sortien d'una manera molt més natural i senzilla. Sense dubte aquesta ha estat la dificultat més gran que ha tingut la pràctica.

Un altre punt que ha estat molt útil de la pràctica ha estat entendre el funcionament dels mètodes *repaint(), paint inmediatelly()* i *paintComponent()*, ja que són mètodes diferents als que estem acostumats a veure.