

PROGRAMACIÓ ORIENTADA A OBJECTES

TRIMESTRE 2 – CURS 2021/2022

PRÀCTICA 1. SESSIÓ 1

Objectiu: Treball amb taules de dues dimensions. Les matrius
Anàlisi descendent. Descomposició funcional

Durada: Dues sessions

Lliurament: Llistat imprès dels fonts i penjar el projecte al Moodle

Data Lliurament: Abans de la Pràctica 2. **3 de Febrer a les 23:50h**

Continuareu treballant amb el joc de la vida, amb uns petits canvis (indicats de color vermell en l'enunciat). Tornem a posar l'enunciat pels estudiants que no van cursar l'assignatura de Fonaments el trimestre passat. El canvis estaran remarcats en color vermell.

Joc de la Vida

El **joc de la vida** és un interessant joc desenvolupat pel matemàtic John Horton Conway de la Universitat de Cambridge. Està pensat com un joc i també com un model de cicle de vida dels organismes vius. Aquest joc es juga amb un tauler de dimensions molt grosses (infinites), encara que el nostre programa les limitarà. Cada element del tauler representa el lloc on viu un dels organismes vius. **Cada casella té un màxim de vuit caselles adjacents** que poden estar ocupades o buides. Cada casella té tants **veïns com caselles adjacents ocupades**.

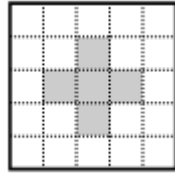
Aplicant tres regles a partir d'un estat inicial es van formant les següents generacions. Aquestes són:

- a) **Naixement:** Es produeix un naixement quan una **casella buida té tres veïns** (3 caselles adjacents amb organismes vius). En qualsevol altre cas la casella queda buida.
- b) **Mort:** Es produeix una mort quan una **casella ocupada té 0 o 1 veïns** (mort per aïllament i solitud) **o be 4 o més veïns** (mort per sobre població i asfíxia).
- c) **Sobre vivència:** Es produeix quan una **casella ocupada té 2 o 3 veïns**.

Es tracta de fer un programa que amb un tauler de dimensions determinades i donat un estat inicial vagi aplicant les regles anteriors. Cada vegada que s'apliquen aquestes regles a totes les caselles es produeix una nova **generació**. El número de generacions a produir també ho indicarà l'usuari.

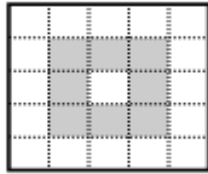
Exemple:

Suposeu un tauler de 5*5 amb el següent **estat inicial**, organismes vius a les caselles: (2,3) (3,2), (3,3) (3,4) (4,3)

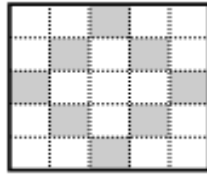


La casella central té 8 caselles adjacents però només 4 veïnes

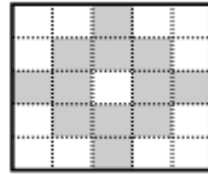
L'evolució de l'estat inicial fins a la **quarta generació** és:



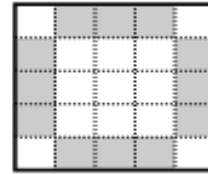
Generació 1



Generació 2



Generació 3



Generació 4

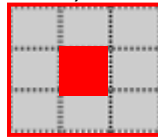
Especificacions del joc

- El taulell el representareu mitjançant una **matriu de booleans**.
- Les dimensions del taulell seran demanades a l'usuari i han de tenir un valor dins d'un interval [5, 15] i **no necessàriament serà quadrat**. Cal insistir fins a tenir dimensions correctes.
- La quantitat d'organismes vius inicialment també serà demanat a l'usuari. S'ha d'insistir mentre el valor no sigui dins de l'interval [1, nombre de caselles del taulell]. Les caselles que contindran aquests organismes vius seran generades **aleatòriament**. Controleu les repeticions de caselles. Per generar un número aleatori useu la funció Math.random() o la classe Random.
- L'usuari abans d'iniciar-se el joc ha d'indicar el número **de generacions** que vol assolir. L'aplicació a l'assolir-les **ha de parar** i mostrar resultats finals. Si abans d'assolir el número màxim de generació el número d'organismes vius que hi ha al taulell és zero l'aplicació **també ha de parar** (a partir d'aquesta ja no es produiran canvis). **També ha de parar si entre dues generacions consecutives no hi ha cap canvi** (veure exemple 2)
- **Després de cada generació a l'igual que un cop creat s'ha de mostrar el taulell de joc**. Useu una representació gràfica del taulell on es pugui veure clarament quines caselles contenen vida i quines no.
- **Quan finalitza el joc, mostreu sempre un resum de l'evolució del joc:**
 - 1.- **després de cada generació cal saber quantes caselles tenen organismes, però la visualització d'aquestes dades s'ha de fer un cop acabat el joc**. Com s'ha de mostrar al final, cal anar emmagatzemant les dades per poder-les mostrar posteriorment. Useu un **vector** de tantes posicions com el número de generacions a crear, i en cada posició un cop creada la generació emmagatzemeu-hi el número de caselles amb organismes vius.
 - 2.- indiqueu la generació que **té més vida**. és a dir, la que té menys caselles buides. Si dos o més generacions generen la mateixa quantitat de vida, **cal que** el missatge ho especifiqui i enumeri una d'elles, és irrellevant quina.
Imprescindible que amb un únic recorregut es determinin les dues dades demanades.

- Quan finalitza una partida l'aplicació ha d'oferir a l'usuari la possibilitat de tornar a jugar. En cas afirmatiu, es mantenen les dimensions del taulell i el número d'organismes vius que hi ha inicialment. S'ha de treballar amb el mateix taulell, buidar-lo (no usar new) i ubicar novament aleatòriament els organismes. L'usuari especificarà el nombre de generacions que vol fer en cada partida.
- Quan l'usuari ja no vol jugar més, el programa s'acomiada i indica el nombre de partides que ha fet.

Pautes de Programació

IMPORTANT: quan tingueu que accedir a les caselles adjacents d'una casella **no utilitzeu 8 estructures condicionals**, useu una **estructura iterativa for** per recórrer la submatriu.



Accés a les caselles adjacents a la casella vermella del gràfic anterior, ubicada a la posició (x,y)

```
for (fil=x-1; fil<=x+1;fil++)
    for(col=y-1; col<=y+1;col++)
        Accés a la casella [fil][col] si
        existeix
```

Serà necessari que treballem amb dos taulells:

- ➔ El taulell que té la generació a partir de la que es crearà la següent, en direm taulell **joc**
- ➔ El taulell que recull els resultats de la nova generació en creació, en direm **aux**

Un cop finalitzada la nova generació s'haurà de copiar el contingut del taulell **aux** sobre el taulell **joc** per continuar generant la següent. **Adoneu-vos de què** així com es fa el recorregut del taulell de la generació actual es consulten caselles adjacents, que podrien tenir informació de la generació en creació que fos diferent de la que té la generació punt de partida si s'emmagatzema en un mateix taulell. **Calen dos taulells.**

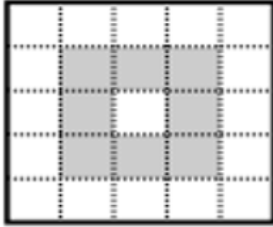
Per tant, per **crear una generació** calen dues passes:

➔ **Pas 1.-** recorregut del taulell **joc** que té la generació actual emmagatzemant els resultat en el taulell auxiliar **aux**, i

➔ **Pas 2.-** Copiar el contingut del taulell auxiliar **aux** sobre el taulell de **joc**, **per això és necessari que feu una còpia de referències.**

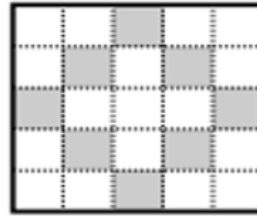
Pas 1

Taulell **joc**



Sobre aquest taulell es farà el recorregut per aplicar les 3 regles

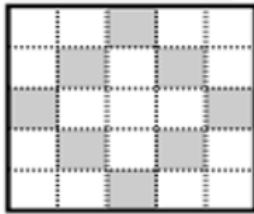
Taulell **aux**



Aquest taulell recollirà els canvis que produeix el recorregut

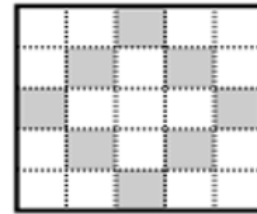
Pas 2

Taulell **aux**



Copiar el contingut del taulell **aux** sobre el **joc**, fent una **copia** de **referències**

Taulell **joc**



Exemples d'execució

Exemple 1

Indica de quines dimensions vols el taulell dins de l'interval [5,15] per cada dimensió, files i columnes

Indica el nombre de files

5

Indica el nombre de columnes

5

Indica quants organismes vius hi vols posar [1,25]

Indica el nombre de organismes

5

Organisme a la posició: 1,2

Organisme a la posició: 2,1

Organisme a la posició: 2,2

Organisme a la posició: 2,3

Organisme a la posició: 3,2

} Posicions generades aleatòriament.
No poden estar repetides!!!

Inicialment

□ □ □ □ □

□ □ ■ □ □

□ ■ ■ ■ □

□ □ ■ □ □

□ □ □ □ □

Quantes generacions vols [1, 10]?

Indica el nombre de generacions

4

Evolució : 1

□ □ □ □ □

□ ■ ■ ■ □

□ ■ □ ■ □

□ ■ ■ ■ □

□ □ □ □ □

Evolució : 2

□ □ ■ □ □

□ ■ □ ■ □

■ □ □ □ ■

□ ■ □ ■ □

□ □ ■ □ □

Evolució : 3

□ □ ■ □ □

□ ■ ■ ■ □

■ ■ □ ■ ■

□ ■ ■ ■ □

□ □ ■ □ □

Evolució : 4

□ ■ ■ ■ □

■ □ □ □ ■

■ □ □ □ ■

■ □ □ □ ■

□ ■ ■ ■ □

Evolució : 1 --> hi ha 8 organismes
 Evolució : 2 --> hi ha 8 organismes
 Evolució : 3 --> hi ha 12 organismes
 Evolució : 4 --> hi ha 12 organismes
 Hi ha més d'una evolució amb la mateixa vida. Una d'elles és la 3
 Vols tornar a jugar (s/n)?

n

Adéu!! Has fet 1 partides.

Per extreure aquestes dades **NOMÉS**
 s'ha de fer un **únic** recorregut del vector

És irrellevant la generació
 indicada quan hi ha
 coincidència

Exemple 2

Indica de quines dimensions vols el taulell dins de l'interval [5,15] per cada dimensió, files i columnes

Indica el nombre de files

6

Indica el nombre de columnes

7

Indica quants organismes vius hi vols posar [1,42]

Indica el nombre de organismes

33

Organisme a la posició: 4,2
 Organisme a la posició: 3,1
 Organisme a la posició: 2,1
 Organisme a la posició: 1,1
 Organisme a la posició: 1,3
 Organisme a la posició: 3,4
 Organisme a la posició: 3,3
 Organisme a la posició: 0,3
 Organisme a la posició: 3,6
 Organisme a la posició: 1,4
 Organisme a la posició: 2,5
 Organisme a la posició: 4,0
 Organisme a la posició: 5,1
 Organisme a la posició: 4,6
 Organisme a la posició: 2,3
 Organisme a la posició: 0,1
 Organisme a la posició: 5,4
 Organisme a la posició: 1,0
 Organisme a la posició: 5,3
 Organisme a la posició: 2,2
 Organisme a la posició: 4,1
 Organisme a la posició: 4,3
 Organisme a la posició: 0,2
 Organisme a la posició: 4,5
 Organisme a la posició: 2,4
 Organisme a la posició: 1,5
 Organisme a la posició: 5,2
 Organisme a la posició: 4,4
 Organisme a la posició: 1,2
 Organisme a la posició: 3,0
 Organisme a la posició: 2,6
 Organisme a la posició: 2,0
 Organisme a la posició: 5,6

Posicions generades aleatòriament.
 No poden estar repetides!!!

Inicialment

```

■ ■ ■ ■ □ □
■ ■ ■ ■ ■ □
■ ■ ■ ■ ■ ■
■ ■ □ ■ ■ ■
■ ■ ■ ■ ■ ■
□ ■ ■ ■ □ ■
  
```

Quantes generacions vols [1, 10]?

Indica el nombre de generacions

3

Evolució : 1

```

■ □ □ □ □ □
□ □ □ □ □ ■
□ □ □ □ □ ■
□ □ □ □ □ □
□ □ □ □ □ ■
■ □ □ □ □ ■

```

Evolució : 2

```

□ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □

```

**Generació sense vida. No cal continuar però SI
cal mostrar els resultats finals**

Evolució : 1 --> hi ha 6 organismes
Evolució : 2 --> hi ha 0 organismes
Evolució : 3 --> hi ha 0 organismes
L'evolució amb més vida ha estat la: 1

Vols tornar a jugar (s/n)?

s

```

Organisme a la posició: 3,0
Organisme a la posició: 5,0
Organisme a la posició: 5,4
Organisme a la posició: 4,0
Organisme a la posició: 4,5
Organisme a la posició: 0,6
Organisme a la posició: 0,2
Organisme a la posició: 0,5
Organisme a la posició: 4,3
Organisme a la posició: 4,1
Organisme a la posició: 2,2
Organisme a la posició: 2,3
Organisme a la posició: 3,1
Organisme a la posició: 1,3
Organisme a la posició: 0,3
Organisme a la posició: 5,3
Organisme a la posició: 0,4
Organisme a la posició: 2,0
Organisme a la posició: 4,2
Organisme a la posició: 1,6
Organisme a la posició: 2,6
Organisme a la posició: 5,6
Organisme a la posició: 3,6
Organisme a la posició: 2,5
Organisme a la posició: 5,5
Organisme a la posició: 2,1
Organisme a la posició: 0,1
Organisme a la posició: 1,0
Organisme a la posició: 3,4
Organisme a la posició: 3,2
Organisme a la posició: 2,4
Organisme a la posició: 4,6
Organisme a la posició: 1,5

```

NOVA PARTIDA. Es mantenen les dimensions del taulell i el nombre d'éssers vius inicials. **El taulell NO S'HA de crear novament, només buidar-lo.**

Novament posicions generades aleatòriament. No poden estar repetides!!!

Inicialment

```

□ ■ ■ ■ ■ ■
■ □ □ ■ □ ■
■ ■ ■ ■ ■ ■
■ ■ ■ □ ■ □
■ ■ ■ ■ □ ■
■ □ □ ■ ■ ■

```

Quantes generacions vols [1, 10]?

Indica el nombre de generacions

3

Evolució : 1

```

□ ■ ■ ■ □ ■
■ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □
□ □ □ □ □ □
■ □ □ ■ □ ■

```

Evolució : 2

```

■ ■ ■ ■ ■
■ ■ ■ ■ ■
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □

```

Si dues generacions consecutives són iguals, no hi ha cap canvi, no s'han de fer més generacions

Evolució : 3

```

■ ■ ■ ■ ■
■ ■ ■ ■ ■
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □

```

```

Evolució : 1 --> hi ha 8 organismes
Evolució : 2 --> hi ha 4 organismes
Evolució : 3 --> hi ha 4 organismes
L'evolució amb més vida ha estat la: 1
Vols tornar a jugar (s/n)?
n

```

Adéu!! Has fet 2 partides.

Important

Escriviu procediments per aquelles parts que per sí soles tenen una clara funcionalitat i també per aquelles que s'han d'executar varies vegades amb l'objectiu de reaprofitar codi. Recordeu de crear un paquet dins del projecte corresponent a la pràctica 1 de l'assignatura.

Organització. Totes les pràctiques s'han de fer en grups de 2 estudiants

Seguireu la mateixa estructura que la usada en les pràctiques de l'assignatura de Fonaments de Programació. Per cada pràctica cal crear un nou projecte amb nom **PràcticaXCognom1Nom1&&Cognom2Nom2**. La X representa el número de la pràctica. Cada projecte ha de tenir paquets per organitzar les classes, un paquet per cadascun dels exercicis que se us demana. En aquesta sessió només dissenyareu una classe que serà ubicada dins del paquet corresponent a la sessió 1.

```

▼ 📁 Pràctica1LinaJuan
  ▼ 📁 src
    ▼ 📁 Pràctica1Sessió1
      > 📄 JocVida.java
      > 📁 Pràctica1Sessió2
    > 📁 JRE System Library [JavaSE-1.8]

```


Què se us subministra?

L'enunciat amb les especificacions i línies a seguir per el desenvolupament del joc.

Què s'ha de lliurar i com?

S'ha de lliurar la carpeta que conté el projecte Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar un **llistat en paper** del codi desenvolupat (no la classe Prova). El format de lliurament d'aquest codi ha de seguir el patró indicat en la presentació de l'assignatura: amb portada, índex, número de pàgina, tabulació ... En **aquest llistat** cal que indiqueu:

- **la distribució de la feina entre els dos estudiants. És a dir, el grau de participació de cada membre del grup en la realització d'aquesta activitat.**
- **Si el programa no funciona cal que indiqueu quina/es parts no funcionen explicant que és el que passa.**

On s'ha de lliurar?

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

Quan s'ha de lliurar?

El lliurament es podrà fer fins el **dia indicat a sota**. Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.

Lliurament Moodle Grups 101 i 102 → 3 de Febrer a les 23:50h

Pauta de correcció:

Aquesta pràctica de dues sessions resoldreu un únic exercici per sessió.

Cadascun té un pes de 5 punts però la correcció es farà sobre 10 i després es ponderarà.

En la correcció de l'exercici d'aquesta sessió es valorarà:

- 1.- **Descomposició funcional** aplicada, tant per un reaprofitament del codi com per la claredat de la solució mostrada. És imprescindible que cada tasca amb una funcionalitat pròpia tingui el seu propi procediment.
- 2.- L'aplicació dels **esquemes** adients (recorregut i cerca)
- 3.- La **robustesa**, és a dir el control de què les dades demanades a l'usuari siguin correctes.
- 4.- **Eficiència** en quan a no fer càlculs repetits amb unes mateixes dades.
- 5.- Ús de les estructures de dades indicades en l'enunciat. Taules per emmagatzemar dades estadístiques.
- 6.- Pautes de programació per l'accés a les caselles adjacents.

La qualificació dependrà dels ítems indicats prèviament. **Obligatòriament heu de seguir les pautes de programació** donades en l'escriptura del programa. La vostra solució s'ha d'ajustar a l'enunciat de l'exercici.

Si el programa no funciona la qualificació màxima que es pot assolir és un 3.

