

El joc de la vida

Estructura de Computadors II - Pràctica

1 Introducció

En aquesta pràctica desenvolupareu una aplicació amb interfície gràfica sobre EASy68K. L'aplicació implementarà una versió per a dos jugadors del joc de la vida (*Game of Life*) creat per John Conway [1]. Per simplicitat, al llarg d'aquest document s'emprarà el terme Joc de la Vida en referència a la versió per a dos jugadors.

La pràctica consta de tres grans apartats on cada apartat requereix de la resolució de l'anterior. Aquests apartats, il·lustrats a la Figura 1, intenten reproduir tres possibles nivells en que es podria dividir una aplicació: el nivell de sistema, que accedeix al maquinari; el nivell d'interfície, que proporciona funcionalitats d'interfície gràfica; i el nivell d'aplicació, que en aquesta pràctica implementa el Joc de la Vida. Com podeu suposar, el nivell d'interfície empra funcionalitats del nivell de sistema i el nivell d'aplicació fa servir tant el nivell d'interfície com el de sistema.

2 El Joc de la Vida

El Joc de la Vida, també conegut com *Game of Life* o, simplement, *Life* és un autòmat cel·lular dissenyat per John Conway [1]. A grans trets, el Joc de la Vida considera una graella bidimensional infinita on cada cel·la pot estar viva o morta. A cada iteració (anomenada *generació*) s'avaluen totes les cel·les de la graella i es determina el seu nou estat. Així, algunes cel·les moriran (passaran de vives a mortes) i d'altres naixeran (passaran de mortes a vives).

Les regles d'avaluació de les cel·les són extremadament senzilles i només depenen del nombre de veïns vius de cada cel·la. Trobareu informació sobre aquestes regles a multitud de llocs web [2], així com a la descripció original del Joc de la Vida [1].

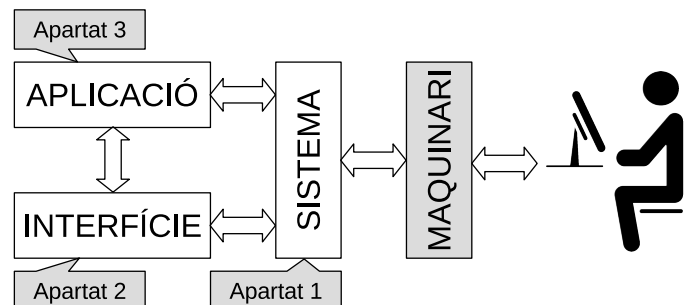


Figura 1: Apartats de la pràctica.

La versió del Joc de la Vida que s'ha d'implementar en aquesta pràctica es diferencia de l'original en dos aspectes:

Graella finita A la nostra implementació, la graella on es desenvolupa el joc tindrà unes mides predefinides. En particular tindrà GRDHEIGHT files i GRDWIDTH columnes. Ja que la graella és finita, s'han de considerar de manera especial els seus contorns. A la nostra implementació considerarem que els límits dret i esquerra són adjacents i els límits superior i inferior també. Així, per exemple, la cel·la (0,10), expressant les coordenades com a (fila,columna), és adjacent a les cel·les (0,9), (0,11), (1,10) i (GRDHEIGHT-1,10); i la cel·la (10,0) és adjacent a les cel·les (9,0), (11,0), (10,1) i (10,GRDWIDTH-1). A l'hora d'implementar-ho és molt recomanable que aprofiteu que GRDWIDTH i GRDHEIGHT són potències de dos. Tenir-ho en compte us facilitarà la feina i reduirà molt el temps d'execució.

Dos jugadors Implementarem una versió per a dos jugadors del Joc de la Vida. Existeixen moltes versions per a dos o més jugadors, però nosaltres ens centrarem en la que es descriu a [3]. Aquesta versió és molt semblant a l'original i, per tant, és necessari que entengueu tant el Joc de la Vida original com la versió per dos jugadors.

Tot i que no necessiteu entendre el Joc de la Vida per dos jugadors fins que arribeu a l'Apartat 3 de la pràctica, descrit a la Secció 4.3, és convenient que entengueu des d'un principi els conceptes bàsics del que serà el resultat final de la pràctica.

3 Material subministrat

Juntament amb aquest enunciat se us proporciona un conjunt d'arxius distribuïts en tres carpetes anomenades APARTAT1, APARTAT2 i APARTAT3 que contenen material relacionat amb cada un dels tres apartats de la pràctica. A cada carpeta trobareu el següent:

Arxius de treball En general es tracta d'arxius X68, però també pot haver-n'hi d'altres formats. Són els arxius que podeu modificar, amb les restriccions que s'indicaran més endavant, per a resoldre l'apartat.

MAINx.X68 A on x denota l'apartat. Aquest arxiu fa INCLUDE dels arxius propis de la pràctica i no el podeu modificar de cap manera. L'objectiu de l'apartat és aconseguir que l'execució de MAINx.X68 es correspongui amb el que es mostra al vídeo RESULTATx.mkv.

RESULTATx.mkv A on x denota l'apartat. És un vídeo (amb una breu explicació verbal, de manera que també l'haureu d'escollar) que mostra quin és el resultat esperat de l'execució de MAINx.X68 un cop completat l'apartat.

4 Treball a realitzar

L'objectiu de cada apartat és molt simple: afegir codi a alguns dels arxius de treball de l'apartat per tal que l'execució de MAINx.X68 sigui la mateixa que es mostra al vídeo RESULTATx.mkv.

Un cop acabeu un apartat i estigueu segurs de la seva correcció haureu de copiar tots els arxius de treball d'aquell apartat ja resolt (és a dir, tots els arxius excepte MAINx.X68 i RESULTATx.mkv) a la carpeta del següent apartat. És a dir, cada apartat depèn de la correcta resolució de l'anterior i, per tant, els heu de resoldre per ordre.

En aquest enunciat s'indicarà quines subrutines o fragments de codi heu de programar o completar a cada apartat. Dos requisits molt importants a complir són:

- Només podeu modificar els arxius de treball on s'hagin de programar o completar els fragments de codi o subrutines indicats. La resta d'arxius (incloent el MAINx.X68) no els podeu modificar de cap manera.
- Als arxius de treball on hageu de programar o completar fragments de codi o subrutines només hi podeu *afegir* elements. No podeu borrar o modificar codi preexistent.

Cal destacar que la informació que es proporciona en aquest enunciat sobre què fer a cada apartat és mínima. Trobareu informació més detallada als propis comentaris del codi subministrat. A més a més, és imprescindible que entengueu tot el codi per tal d'entendre tots els elements implicats. Ja que l'objectiu de cada apartat és aconseguir que MAINx.X68 es comporti tal i com mostra RESULTATx.mkv, aquest ha d'esser el fil conductor que us guïï si no teniu clar què fer a un apartat.

4.1 Apartat 1: Sistema (1 punt)

En aquest apartat només heu d'afegir codi a l'arxiu SYSTEM.X68, el qual conté codi de sistema. En particular, heu de programar les rutines MOUINIT, MOUREAD y SYSINIT. Dins SYSTEM.X68 en trobareu les etiquetes i comentaris descriptius.

Per tal de comprovar si l'apartat està resolt correctament heu de, a més d'executar MAIN1.X68 i comparar-lo amb el que mostra el vídeo RESULTAT1.mkv, verificar amb el debugger que MOUEDGE s'actualitza correctament.

4.2 Apartat 2: Interfície gràfica (2 punts)

En aquest apartat heu d'afegir codi als arxius BUTTON.X68 i BTNLIST.X68, els quals contenen codi relacionat amb la interfície gràfica. Més concretament, BUTTON.X68 implementa les tres funcionalitats bàsiques d'un botó (inicialització, actualització i dibuixat) i BTNLIST.X68 implementa funcionalitats que permeten gestionar un conjunt (llista) de botons (inicialitzar, afegir un botó a la llista, actualitzar tots els botons de la llista i dibuixar tots els botons de la llista).

En particular, dins BUTTON.X68 heu de programar les rutines BTNINIT, BTNUPD i BTNPLOT. Dins BTNLIST.X68 heu d'implementar BTLPLOT. Dins

aquests dos arxius trobareu les etiquetes i comentaris descriptius de les rutines a implementar.

Podreu comprovar si la vostra implementació va per bon camí executant MAIN2.X68 i comprovant si la seva execució coincideix amb el que es mostra a RESULTAT2.mkv.

4.3 Apartat 3: Aplicació (5 punts)

En aquest apartat programareu tant part de la gestió genèrica de l'aplicació (arxiu GOL.X68) com rutines més específiques de tractament de la graella i del Joc de la Vida (arxiu GRID.X68).

És recomanable que abans de començar a programar i, fins i tot, abans de continuar llegint l'enunciat, mireu el vídeo RESULTAT3.mkv per tal d'entendre què s'ha d'aconseguir en aquest apartat, que és el més complex de la pràctica.

També és molt recomanable que abans de començar a programar entengueu la forma en que es gestiona la graella de joc dins GRID.X68. S'ha de tenir en compte que a cada generació es decideix el nou valor d'una cel·la en funció dels valors de les cel·les veïnes. Per tal que aquest nou valor no influeixi en els càlculs de les cel·les fins a la següent generació són necessàries dues matrius. D'aquesta manera, a una generació s'avalua una matriu A i es posa el resultat dins la matriu B, a la següent generació s'avalua la matriu B i es posa el resultat a la matriu A i així successivament. A més a més, a l'hora de dibuixar la graella sempre s'empra la matriu on s'ha posat el resultat.

A nivell de programa això s'aconsegueix disposant de dues matrius (GRDMATR0 i GRDMATR1) i de dos punters a matriu (GRDSRC i GRDDST). Així, l'actualització de la graella comença intercanviant els dos punters i després consulta l'estat de la matriu apuntada per GRDSRC per tal de construir el resultat a la matriu apuntada per GRDDST. Amb aquesta lògica, el dibuixat sempre es basa en GRDDST i l'emplenat manual amb ratolí també opera sobre la matriu apuntada per GRDDST.

El funcionament emprant aquesta doble matriu pot resultar una mica confús, de manera que és molt recomanable analitzar bé el codi proporcionat i no començar a treballar fins haver-lo entès completament.

Un cop tingueu clars tots aquests conceptes i hageu analitzat el codi proporcionat, heu d'implementar el següent. Dins l'arxiu GRID.X68 heu d'implementar GRDMUPD, GRDRUPD, GRDLLEFT, GRDLRGT i GRDPLOT.

Dins l'arxiu GOL.X68 no heu d'implementar cap subrutina, però sí que heu de posar els *callbacks* pertinents als *Static Data Blocks* (SDB) dels botons, definits al final de l'arxiu.

Com de costum, dins aquests dos arxius trobareu les etiquetes i comentaris descriptius de les rutines a implementar.

Podeu verificar la correcció de la vostra implementació executant MAIN3.X68 i comprovant si la seva execució concorda amb el mostrat al vídeo RESULTAT3.mkv.

5 Entrega

L'entrega consistirà en un únic arxiu ZIP anomenat PRACTICA.ZIP que contendrà:

Carpeta CODI Ha de contenir tot el codi i arxius addicionals de l'apartat 3 (tant el que heu fet vosaltres com el que ja se us ha subministrat) llest per a ser executat. És a dir, s'ha de poder obrir i executar l'arxiu MAIN3.X68 i ha de donar com a resultat el que es veu a RESULTAT3.mkv. No heu d'entregar cap altre carpeta, ja que tota la feina està continguda a l'apartat 3.

Arxiu DOCUMENT.PDF És la documentació de la pràctica, amb un màxim de 5 pàgines. Es penalitzarà amb 1 punt cada pàgina que superi les 5. El document ha d'incloure una portada amb, com a mínim, el nom dels autors, i un índex. La portada i l'índex no entren en el còmput de pàgines. A més a més, de manera opcional, podeu posar un últim apartat a la documentació d'opinió personal sobre la pràctica. Aquest apartat no entra en el còmput de pàgines i no té cap pes en la nota de la pràctica.

6 Criteris d'avaluació

Les puntuacions mostrades a cada un dels apartats es refereixen al pes que tenen les tasques de cada apartat en la nota de la pràctica. Recordau que només heu d'entregar l'apartat 3 però que aquest inclou els altres dos apartats de manera implícita. Els dos punts que falten per arribar a 10 provenen de la documentació.

Per tal d'avaluar la pràctica es considerarà:

Funcionament Si l'execució de MAIN3.X68 no es correspon amb el que s'indica a RESULTAT3.mkv, o bé provoca errors en temps d'assemblatge o d'execució, la pràctica s'avaluarà amb 0 punts.

Especificacions Es comprovarà que les subrutines programades compleixin les especificacions indicades (entrada, sortida, registres que modifiquen, ...). En particular, es provarà el vostre codi substituint alguns arxius per d'altres implementats pel professor. Per exemple, emprar el vostre GRID.X68 amb un BUTTON.X68 implementat pel professor, o emprar el vostre BTNLIST.X68 amb un GOL.X68 implementat pel professor, ... Els errors detectats penalitzaran des de 1 punt (errors lleus, com ara registres no restaurats però que no afecten a l'execució de la pràctica) fins a 10 punts (errors molt greus, com ara que la pràctica no funciona quan s'intercanvia algun dels arxius amb algun del professor).

Qualitat del codi Es valorarà l'ús adequat del llenguatge assemblador del MC68000.

Optimització del codi Aquest apartat està molt relacionat amb l'anterior, però fent èmfasi en els recursos emprats. En particular, es penalitzaran fragments de codi que emprin un excés de registres o de memòria i també fragments de codi excessivament lents. Com a part de l'avaluació d'aquest punt es podrien prendre mesures de nombre de cicles de rellotge consumits.

Claredat del codi Es valorarà la claredat del codi. En aquest sentit és vital emprar adequadament les constants proporcionades i comentar el codi de manera correcta (es penalitzarà tant un excés de comentaris com una manca de comentaris).

Documentació Dos punts de la nota de la pràctica provenen de la documentació. De la documentació es valoraran els continguts i l'estructura, però també la redacció i la forma. Així, una documentació ben estructurada i amb els continguts adients pot valorar-se amb 0 punts si hi han molts errors ortogràfics. A més a més, una documentació excessivament dolenta en tots els apartats pot provocar un suspens de tota la pràctica amb 4.0 punts amb independència de la resta dels apartats.

Referències

- [1] M. Gardner, "Mathematical games. the fantastic combinations of john conway's new solitaire game "life" by martin gardner," *Scientific American*, vol. 223, pp. 120–123, 1970. [Online]. Available: <https://www.ibiblio.org/lifepatterns/october1970.html>
- [2] Wikipedia, "Conway's game of life," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- [3] M. Levene and G. Roussos, "A two-player game of life," *International Journal of Modern Physics C*, vol. 14, no. 02, p. 195–201, Feb 2003. [Online]. Available: <https://arxiv.org/abs/cond-mat/0207679>