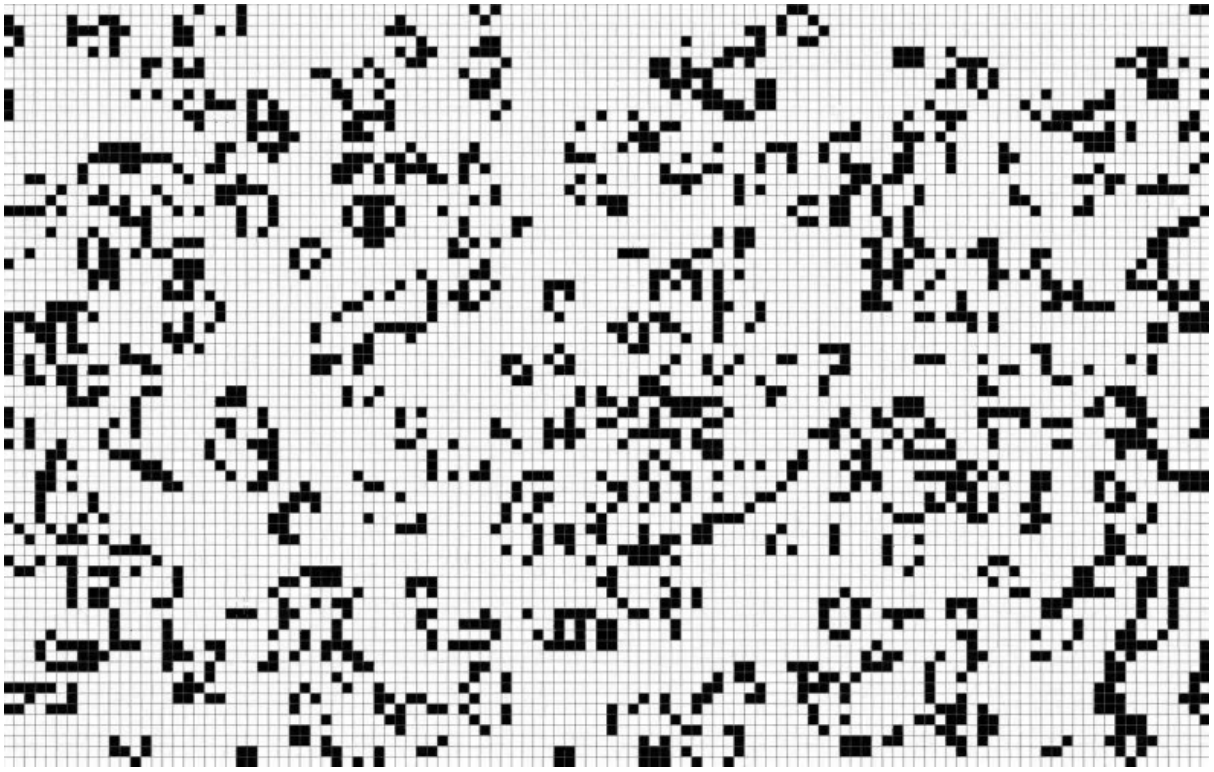
El Joc de la Vida



Assignatura: Estructura de Computados II,Part Pràctica.
Docent de l'Assignatura: Dr. Antonio Burguera Burguera.
Autors: Pere Joan Vives Morey, Carlos Lozano Alemañy.

INDEX

SINTESI DEL PROBLEMA

El treball consistirà en crear una aplicació a l'interfície de 'EASY68K' basat en el 'Joc de la vida'.
El 'Joc de la vida' va ser inventat pel matemàtic britànic "John Horton Conway", que consistia en veure com evolucionava un patró format per cel.les dins una graella en funció de l'estat en que es trobàs, fora cap interacció alguna de l'usuari, poguent així, arribar a veure patrons molt complexes a partir de regles molt senzilles.
En el nostre cas, l'escenari on es mostrarà aquesta evolució serà dins una graella finita,on hi "participaran" dos jugadors.
Cada jugador vindrà representat per un color, i ambdós tindran una evolució diferent a partir dels seus estats inicials, aquest estat inicial vinrà donat per l'usuari, que "pintarà" cel.les amb el ratol.lí.
Cada patró està format per cel.les,les quals poden estar o bé "vives" o bé "Mortes", cada cel.la tindrà 8 cel.les veïnes possibles, les quals influiran en l'evolució dels patrons en funció de les normes facilitades pel profesor.
Doncs atesos a les conidicions inicials, a cada "torn" que es fagi, l'estat de la graella haurà canviat, i podrem veure així l'evolució a partir de l'avaluació de cada estat, alhora de fer la pràctica podrem fer fins 250 "torns", ja que sinó l'implementació podria ser pràcticament infinita.
Per implementar aquesta aplicació, necessitarem complir tres aspectes:
·Apartat 1: Sistema
·Apartat 2: Interfície gràfica
·Apartat 3: Aplicació
A cada apartat haurem d'afegir codi ales subrutines en que s'ens demana, que, juntament amb les subrutines implementades pel professor, i amb el material subministrat, ens servirà per poder dur a terme l'aplicació.
cada apartat superior contindrà el codi implementat a l'apartat inferior, fins arribar així, a un apartat 3 complet, que contindrà tots els continguts.
Es per això, que alhora d'explicar les subrutines, explicarem les de l'apartat 3, que ve a continuació.

EXPLICACIO DE LES SUBRUTINES

Començarem explicant les subrutines de l'arxiu SYSTEM.X68:

SYSINIT

En aquesta subrutina, el primer que feim es deshabilitar les interrupcions, per fer-ho, posam un 7 o un 6 en els tres bits de interrupcions del registre SR, a partir d'aquí feim una crida a la subrutina SCRINIT, que s'encarrega de inicalitzar la finestra, i a la subrutina MOUINIT. Per acabar, passarem a mode usuari posant el valor 1 a la posició 13 del registre de estat SR.

MOUINIT

En primer lloc instalam la subrutina de interrupció MOUREAD a la posició ($80+MOUTRAP*4) d'on MOUTRAP = 1, per tant a la posició $84. Seguidament posam a 0 les variables MOUVAL i MOUEDGE, les quals les emplearem per emmagatzemar l'estat del ratolí. MOUVAL conté en els tres bits menys significatius (bit 2 = botó central, bit 1 = botó esquerre, bit 0 = botó dret) informació sobre si hem pitjat alguns d'aquests botons, de manera si els bits tenen un 0, s'entendrà que aquest botó no s'ha pitjat, i un 1 si s'han pitjat.
De la mateixa manera, MOUEDGE conté informació sobre aquests tres botons, però a diferència de MOUVAL ens dirà si aquests botons han estat clicats, de manera que si algun d'aquests 3 bits té un 1, entendrem que a aquest s'hi ha fet un click.

MOUREAD

L'objectiu d'aquesta subrutina de interrupció és obtenir i emmagatzemar la posició i estat del ratolí. Per a fer-ho en primer lloc guardarem l'actual MOUVAL a un registre de dades D2, això ens servirà més tard per si s'ha produit un clic. A continuació feim us de la tasca 61, la qual després de fer la crida TRAP #15 ens posarà dins D0 l'estat d'alguns botons, entre ells els del ratolí, i dins D1.L, als 16 bits alts la pocisió Y, i als 16 bits baixos la posició X. A continuació ficam la posició del ratolí dins les variables MOUY I MOUX i dins MOUVAL l'estat. Com podem veure, ara tendrem l'estat actual del ratolí dins MOUVAL i l'anterior dins D2. Si a l'estat anterior contingut dins D2 algun dels botons esteia pitjat i a l'actual no ho està, llavors entenem que s'ha clicat. Aquí hem suposat que només es pitjarà un botó a la vegada ja que el que feim es que si es dona el cas de un clic guardarem l'estat anterior MOUVAL contingut dins D2 dins MOUEDGE, i si no guardarem 0's dins MOUEDGE.

Anirem ara a explicar les subrutines de l'arxiu BUTTON.X68

BTNINIT

S'utilitzarà per inicialitzar un botó, com a dades d'entrada tendrem dins A0 el punter al VDB, el qual conté un punter al SDB i un byte d'estat IPC (Inside, Pressed, Clicked) referent a l'estat del ratolí respecte d'un botó, i dins A1 el punter al SDB, per tant el que feim es ficar aquest SDB dins el VDB i ficam 0's als 4 bytes d'estat.

BTNUPD

Actualitzam el VDB del botó i cridam el callback si es necessari. Tendrem el VDB dins A0, a partir d'aquest treim el SDB, el qual ficarem dins A1 i el ICP que ficarem dins A2. Amb això

podrem treure les següents dades: posició i dimensions del botó, a aquestes dades les ficarem dins la pila i calcularem la posició de l'extrem dret inferior del botó de manera que ara tendrem dues posicions amb les que podrem determinar la posició del ratolí respecte el botó.A més, també guardarem el callback que correspon a aquest botó dins A3. A continuació obtenim els valors de MOUX, MOUY, MOUVAL Y MOUEDGE. Amb els valors MOUX i MOUY podrem mirar si el ratolí es troba dins o fora del botó, això definirà el valor I del ICP, si esteim a dedins, ficarem un 1 a la I de ICP i si esteim a fora, un 0. En el cas de que estigui a dins passarem a mirar si l'hem pitjat o clicat, amb el que definirem els valors PC del ICP. Si es dóna el cas de que esteim a dedins i hem clicat el ratolí, farem una crida al callback, el qual haurem guardat dins A3.

BTNPLOT

Dibuixarem un botó a partir de un punter a la VDB del botó. Guardarem el SDB dins A1 i el CPI dins D7, seguidament definirem el tamany del pinzell amb la tasca 93, el qual li hem posat 5, i el color de pintat, el qual ve donat per l'etiqueta BTNPENCL, amb la tasca 80. Ara ens ocuparem de mirar si el ratolí es a dins o a fora del botó, això ho podem mirar a partir de la informació continguda dins D7 (ICP), ens podrem trobar tres casos: que el ratolí estigui a fora, en aquest cas el color de fill serà CLRBLACK i ja no ens fixarem en els altres dos valors CP. Que el ratolí estigui a dins però el valor P (pressed) sigui 0, en aquest cas el color de fill serà BTNSELCL, i que el ratolí estigui a dins i el valor P sigui 1, en aquest cas el color serà BTNPRSCL. Ara que ja tenim els colors definits, el que haurem de fer es dibuixar el rectangle o botó, per fer-ho, utilitzarem la tasca 87 i farem crida TRAP #15, però antes, haurem de treure les coordenades del botó, ja que es el que aquesta tasca ens demana. Un cop hem pintat el botó, ara haurem de pintar les lletres, dins A1 tendrem l'String amb les lletres, però per a que les lletres quedin cèntriques necessitam saber el tamany d'aquest string, per això, feim us de la subrutina UTLSTRLN la qual ens retornarà aquest tamany dins D0. Seguidament calcularem la posició cèntrica a la que haurà de estar l'String i emplearem la tasca 95 per al dibuixat d'aquest.

Subrutines de l'arxiu BTNLIST.X68

BTLPLOT

Així com a la subrutina BTLUPD feim una crida a la subrutina BTNUPD per a cada botó, a aquesta subrutina farem una crida a la subrutina BTNPLOT per a cada botó, per tant la subrutina BTLUPD s'encarregarà de que cada botó s'actualitzi (el seu CPI i callback) i la subrutina BTLUPD s'encarregarà de que cada botó es pinti amb els seus corresponents colors.

Subrutines de l'arxiu GRID.X68

GRDMUPD

En aquesta subrutina modificarem la graella depenent de l'estat del ratolí. Per començar dins A0 guardarem el punter a a la matriu GRDDST, la qual es la que s'ensenya per pantalla. També guardarem els valors MOUVAL, MOUEDGE, MOUY, MOUX que corresponen a la posició i estat del ratolí, i les coordenades de l'inici de la graella. A

continuació, per poder determinar si el ratolí es troba a l'interior necessitam conèixer un parell de coordenades més, la X del centre i del final de la graella i la Y del final de la graella. Amb la X inicial i final i la Y inicial i final podrem determinar si el ratolí es troba a dins o fora, i amb la X del centre podrem saber si el ratolí es troba a la part esquerre o dreta de la graella. Feim les verificacions necessàries i un cop sabem que el ratolí es a dins i a l'esquerre o la dreta i que aquest ha pitjat el botó esquerre o dret , calcularem la cel·la exacte on es troba. Quan sabem la cel·la on es troba, haurem de colocar el punter A0 a aquesta cel·la . Ficarem un 1 si esteim a la part esquerre i hem pitjat el botó esquerre i un 2 si esteim a la part dreta. En el cas de que el ratolí estigui a fora o que estigui a dins i tengui el botó dret pitjat, ficarem un 0.

GRDRUPD

El primer que feim es canviar els punters GRDSRC I GRDDST a més de sumar 1 al nombre de generacions. Com que a aquesta subrutina faig us dels 8 registres de dades D0-D7 he creat una variable anomenada ETIQUETA la qual té un tamany inical de 2048 que correspon al nombre de cel·les de la matriu, aquesta em serà útil per al bucle LOOP. Per a mirar el nombre de veinats que tenim a cada posició de la matriu, farem ús de 3 registres d'adreces: A1 s'encarregarà de mirar la part superior, es a dir, dels 3 veinats que tenim a dalt de la posició actual, A2 s'encarregarà de la part inferior i A3 de la posició actual i els veinats esquerre i dret, per fer-nos una idea, durant una iteració del LOOP aquests registres hauràn mirat el següent:

A1-A1-A1
A3-A3-A3
A2-A2-A2

Per comptar el nombre de veinats també farem ús de dues variables que he creat, anomenades PLAYER1 i PLAYER2 les quals al principi de cada iteració es posaràn a 0, ja que quan canviam de cel·la haurem de tornar els veinats. L'ordre que seguim per mirar els veinats es el següent:

2-1-3
7-9-8
5-4-6

Ara ens trobam amb un problema, i es que es podria donar el cas de que els veinats de dalt o a baix es trobin a fora de la matriu, com passa per exemple quan la posició que esteim mirant es troba a la primera o darrera fila. Per a evitar agafar un valor que es trobi a fora de la matriu hem creat una subrutina anomenada COMPROVA que mira si la direcció de A1, A2 i A3 es troba a dins de la matriu. Si es compleix que tots els veinats estan dins la matriu es guardaràn els valors d'aquesta manera:

D0-D1-D2
D3- X- D4 X:valor actual que mirarem després
D5-D6-D7

Un cop tenim a aquests registres els valors, amb la subrutina VEINATS mirarem el valor de cada registre i sumarem a les variables PLAYER1 i PLAYER2.
Ara que ja tenim el nombre de veinats comptats, miram el valor actual X i feim crida de la subrutina ACTUALITZACIO la qual a partir del nombre de veinats i el valor actual aplica les normes del joc de la vida i fica a la pila el nou valor. Un cop tenim aquest nou valor, el ficam a la matriu GRDDST que tenim apuntada per A4 i anam a la següent iteració.

GRDPLOT

A aquesta subrutina tendrem la matriu GRDDST apuntada per A0, posarem a 0 el nombre de cel·les de jugadors 1 i 2 a l'inici (GRDNPLR1 i GRDNPLR2). El que farem per fer el dibuixat és dos bucles. El primer bucle farà 32 iteracions que corresponen al nombre de files i el segon bucle que estarà dins el primer farà 64 iteracions, que corresponen al nombre de columnes. Al començament del primer bucle reiniciarem les iteracions del segon bucle i calcularem les coordenades de la cel·la (D1 = X1, D2 = Y1, D3 = X2, D4 = Y2). Quan entram al segon bucle agafam el valor de la cel·la actual i el ficam dins la pila ja que a continuació farem una crida a la subrutina de bibilioteca PINTA, la qual a partir d'aquestes coordenades i el valor de la cel·la definirà el color amb el que pintarem la cel·la. Un cop hem sortir de PINTA empleam la tasca 87 i pintam la cel·la i passam a la següent.

Una vegada hem pintat totes les cel·les, ens encarregarem de pintar el marcador. Primer pintarem els rectangles on hi haurà els nombres i en acabar, farem una crida a la subrutina MARCADOR, la qual pinta l'asterisc que ens diu quins dels dos jugadors guanya.

GRDLLEFT i GRDLRIGHT

Les dues son molt semblants, un cop tenim a A0 el punter a la matriu GRDDST i a A1 el punter al contingut del fitxer, en el cas de GRDLLEFT farem 32 iteracions per cada fila i en acabar canviarem de fila, de manera que només mirarem la part esquerre. En el cas de GRDLRIGHT a cada iteració haurem de sumar als punters A1 i A0 la mitat de la graella, de manera que a l'inic de cada iteració ens trobam a la primera cel·la després de la meitat de la graella.

subrutines de l'arxiu GOL.X68

El que hem fet es colocar a cada botó que apareix a la pantalla un callback, aquest callback correspon a l'acció que s'hauria de fer una vegada hem apretat el botó.

OPINIO PERSONAL

Aquest treball ha estat complicat, a la vegada que entretengut.
Personalment, no teniem consciència sobre l'existència del joc, i sa veritat que és un joc bastant curiós.
El més interessant es veure com hem estat capaços de crear una aplicació, (amb ajuda el professorat això si, lo qual ens ob els ulls de cara a idees o projectes futurs que volguem fer. Tampoc pensàvem que amb 'easy68k' poguèssim fer coses com implementar una aplicació interactiva i la veritat que quan ho varem veure varem flipar, però ara ja veim que si que es pot fer, i perfectament.
Per altra banda, el principal inconvenient que hem tingut ha estat la dificultat per trobar els erros, i l'organització, ja que no pensàvem que seria tan llar, aposta ho esteim entregant a l'avaluació extraordinària. No ens tornarà a passar, això de segur.
També com a darrer punt, trobam que els videos proporcionats pel professor, i les contestacions a dubtes que ens han anat sorgint han estat de 10,cosa que agraïm :)
I això es tot!

FI