# Flappy Bee

Alba Aguilera Cabellos, 45612336R

Aila Romaguera Mezquida, 45189101N

## Introducció: explicació de com jugar i llistat de característiques escollides

Aquesta pràctica tracta d'una implementació del joc popular del Flappy Bird amb alguns afegits. En el nostra cas particular es tracta d'una abella, per això "*Flappy Bee*", que ha d'aconseguir no xocar-se amb els arbres i evitar els enemics, a mesura que va intentant aconseguir bolles de mel que apareixen flotant per aconseguir punts. El joc s'acaba quan l'abella mor o quan aconsegueix arribar al nivell màxim, en el nostre cas el nivell 10.

**Controls de joc:**

L'abella es mourà fent ús de la tecla espai al ser pressionada per pujar, i alliberant aquesta per baixar. D'altra banda, per aconseguir les bolles de mel s'haurà de clicar a sobre fent ús del ratolí.

**Taula de punts/vides:**

| | POINTS | LIFES |
|---|---|---|
| LIFES START | | 3 |
| SPIDER DAMAGE | -200 | -1 |
| COLUMN DAMAGE | DEATH | DEATH |
| COLUMN SCORE | +25 | |
| HONEY BONUS | +100 | |

A l'inici de la partida es comença amb 0 punts i 3 vides, per tant, partint de les dades de l'esquema anterior, podem veure que si no s'han acumulat al menys 200 punts, la col·lisió amb una aranya serà mortal.

Les característiques escollides son la *visualització d'imatges* i la *implementació del ratolí*.

## Estructura del codi

MAIN; Classe d'execució on es declaren la resta de classes a fer servir per el programa, a més de realitzar les crides pertinents per a activar la jugabilitat.

GAME; Classe que engloba l'estructura del joc, amb la inicialització, el update, i el plot. A la inicialització s'afegeixen els elements del joc.

SYSTEM; Classe que inicialitza els Traps de sistema en mode supervisor. També s'inicialitzen els sons de la partida, i els perifèrics: la finestra de visualització amb els fotogrames per segon, el ratolí i el teclat.

BEE; Classe de gestió de l'abella, que es el jugador, amb la inicialització, actualització i dibuix.

AGENTS; S'han implementat algunes classes com **agents** per facilitar la programació i el rendiment del joc. Totes les classes d'aquest tipus disposen de una inicialització, actualització i dibuix.

- SPIDER; Classe de gestió de l'aranya que es el principal enemic del joc i apareixen cada 3 columnes.
- COLUMN; Classe de gestió de la columna. Les columnes van apareguent contínuament per la dreta de la pantalla i desapareixen per l'esquerra i suposen els obstacles per l'abella.
- HONEY; Classe de gestió de les bolles de mel, que apareixen continuament per aportar bonus al jugador.
    - XPLOSION; Classe que implementa les bolles resultants d'una explosió de mel.

SCORE; Classe de gestió del rectangle que apareix a la part superior esquerra on el jugador pot visualitzar el nivell, els punts i les vides.

SKY; Classe de dibuix del fons del joc que simula un cel.

IMGDATA; Classe que emmagatzema les dades per crear els polígons que conformen les lletres del logotip del joc a l'inici de la partida.

POLYIMG; Classe de tipus llibreria, dibuixa imatges a partir de coordenades que defineixen polígons.

STATES; Classe de gestió dels estats del joc.

- INTRO; Classe que gestiona l'estat de "introducció".
- WIN; Classe que gestiona l'estat de "victòria".
- GOVER; Classe que gestiona l'estat de "derrota".
- INSTRUCTIONS; Classe que gestiona l'estat que explica les instruccions-

VARS; Llistat de variables utilitzades al joc.

SYSVARS; Llistat de variables de sistema.

CONST; Llistat de constants del joc.

SYSCONST; Llistat de constants del joc.

## Implementació característiques escollides

- **Visualització d'imatges**; S'implementa a *BITMAP.X68,* i l'utilitzem a diversos llocs, com a la pantalla d'inici, al joc per implementar un núvol, i a les pantalles de "*WIN*" i "*GAME OVER*".
  Primerament, vàrem desenvolupar a Netbeans un programa que transforma imatges al format requerit per l'Easy68K en files de 8 Long, de la següent manera:
  DC.L $000000,$000000,$000000,$000000,$000000,$000000,$000000,$000000
  Aquests Longs conformen cada color de cada bit.

  Després a l'Easy68k en la rutina *MAPPLOT* iteram per l'array de Longs mentres que cridam a les subrutines del TRAP #15 corresponents a posar un color i dibuixar un quadrat.
  Les imatges s'han processat com Bitmaps ja que les hem creades al paint i ens resultava més fàcil comprendre així com es transformava la imatge en la llista de Longs, però el programa fet a Netbeans anomenat *BitmapConverter2.java* fàcilment podría ser modificat per funciona amb altres formats d'imatges.

- **Implementació del ratolí**; La realitzem en *SYSTEM.X68* a les rutines de MSEINIT i MSEUPD. A MSEINIT inicialitzem el IRQ del ratolí, habilitant el nivell 1 pel moviment i el click del botó del ratolí. Guardem les coordenades a la variable MSECOORD i l'estat del ratolí (pitjat o alliberat) a MSEVAL. Després a MSEUPD anem comprovant si el ratolí s'ha mogut i actualitzem les coordenades.

## Principals dificultats:

Una de les majors dificultats amb la que ens hem trobat ha estat l'organització dels codis a desenvolupar en cada moment, doncs a partir de la idea d'un videojoc ja existent, contàvem

amb l'avantatge de saber com tindria que quedar finalment el nostre treball, però també amb la dificultat de com hauríem de plantejar el disseny de manera que no ens veiéssim sobrepassades per la gran quantitat de implementacions que teníem que aconseguir, tot això sense fer menció dels problemes esporàdics amb els que ens hem trobat a mesura que la programació anava avançant, la majoria dels quals no sabíem a que es devien i era necessari estudiar-los i solucionar-los però ser capaços de prosseguir.
D'aquesta manera, hem anant aprenent a mesura que hem anat programant, solucions que a l'inici pensàvem que eren més ideals més endavant han resultat enrevessades. Un exemple d'això és el tractament de les col·lisions dels agents. Primerament vàrem definir les columnes com un sol agent amb dues parts, la superior i la inferior, llavors treballem amb l'espai entre els dos extrems per determinar si hi ha col·lisió o no. En canvi per els altres agents mirem la dimensió de l'agent directament. Això pot parèixer que no es una programació orientada a objectes, ja que s'han implementat de distinta manera dos objectius iguals.

Una altra dificultat que vàrem tenir al principi va ser definir el moviment de l'abella, ja que volíem implementar una caiguda lliure i que els "bots" també siguin afectats per la gravetat. Dur condicions del món real i recrear-les al joc ha sigut un poc complex a nivell conceptual, però la implementació finalment a resultat senzilla, amb l'ús de punt fixe com a recomanació del professor.

La segona gran complicació va ser la implementació dels requisits necessaris que vàrem seleccionar, especialment ha sigut un repte la visualització d'imatges, ja que estem acostumades a que un programa directament processi la imatge i la mostri per pantalla, però en el cas de l'Easy68k hem hagut d'entendre com son formades les imatges i com "repintar-les" a l'ensamblador.

## Conclusió

Ens ha resultat divertit poder crear un joc des de zero, ja que s'ha tractat un procés molt creatiu, des de definir les restriccions, fins al disseny del joc i dels elements.
Creiem que una clara millora que es podria fer és fer el codi més compacte i no tenir tant de codi redundant. La naturalesa del llenguatge ensamblador dificulta aquest enfocament, ja que és més fàcil enfrontar-se als reptes de manera seqüencial, però el problema apareix quan el projecte es va fent més gran, i et trobes en la situació de que hi ha codi repetit. Pot ser també

hem perdut massa temps en qüestions estètiques, ja que aquests aspectes ens resulten personalment més atractius.

Així i tot el joc que hem implementat es senzill, hem aconseguit uns objectius realistes i un rendiment sense falles. A més, hem aconseguit implementar les dues implementacions extres necessàries de manera exitosa. Tot i que ha sigut una pràctica on s'ha hagut de treballar i estudiar molt, s'ha obtingut un resultat més satisfactori al esperat, més que pel joc en sí, pel fet d'haver sigut capaços de dur a terme la idea que es va plantejar.