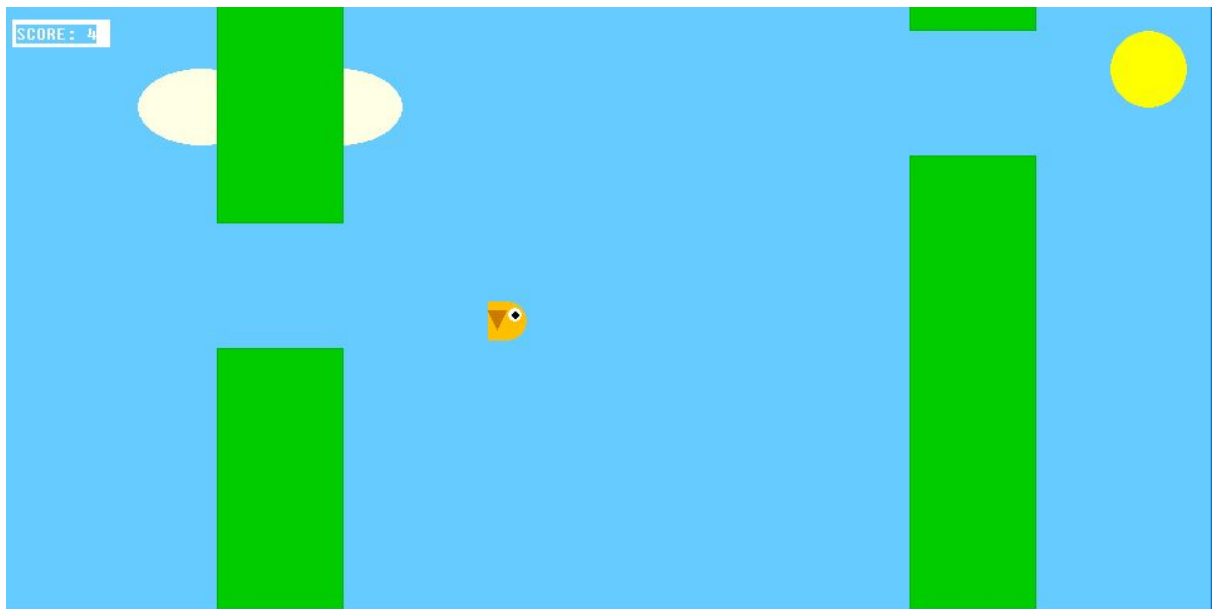


# FLAPI 68000



Bernat Pericàs Serra - 43212796M  
Alejandro Calle Ballester - 43460148S

## Introducció

Aquesta pràctica té com a objectiu implementar un joc involucrat amb l'E/S al simulador del 68k. Es va subministrar un codi de partida per seguir una mateixa estructura i a partir d'aquesta i els jocs d'exemple s'ha construït el joc **FLAPI 68000**.

Aquest joc és una versió del conegut (i polèmic) joc Flappy Bird però varia principalment en la forma de moure la figura. En el joc original, l'ocell es controla únicament amb una tecla i jugant amb la gravetat, en canvi en el **FLAPI 68000**, pots usar les quatre fletxes per controlar-lo i, la gravetat no accelera la caiguda, sinó que ho fa de manera constant.

## Estructura

Per estructurar el joc, hem pres com a punt de partida el codi que s'ha donat i l'exemple del joc que hi havia a l'aula digital.

D'aquesta manera, hem dividit el joc en diferents arxius .X68:

- **/RES**: Arxius d'àudio del joc.
- **AGENTLIST**: Crear, actualitzar, pintar i matar agents.
- **AUDIO**: Iniciar l'àudio i reproduir sons.
- **CANONADA**: Inicialitzar, actualitzar i pintar canonades.
- **CONST**: Constants del joc.
- **FONS**: Dibuixar el fons de pantalla del joc.
- **OCELL**: (PLAYER) Iniciar, actualitzar i pintar l'ocell.
- **SPAWNER**: Actualitzar l'awner.
- **STATES**: Gestió dels diferents estats del joc.
- **SYSCONST**: Constants del sistema pels TRAPS, teclat, pantalla i memòria.
- **SYSTEM**: Inicialitzar teclat i gestió de la memòria dinàmica.
- **SYSVAR**: Variables del sistema.
- **UTIL**: Conjunt de macros i subrutines útils a l'hora de fer un joc.
- **VAR**: Variables del joc.

## Dificultats trobades

A vegades no es detectava quan es passava per una canonada, és a dir, la puntuació del joc no s'actualitzava. Això passava quan la velocitat era elevada, o si el jugador anava cap endavant mentre passava per la canonada. Segurament, el problema era que, en anar tan de pressa, en actualitzar la posició i repintar, es botava la canonada. Es va solucionar augmentant el gruix de les canonades fins als 100px (`CANOWIDTH EQU 100`).

Dibuixar és una tasca molt costosa, pel que s'ha optat per dibuixos simples, tanmateix no era aquesta la finalitat de la pràctica.

Vam tenir bastants dificultats per detectar la col·lisió amb una canonada, fins que vam adonar-nos que podíem basar-nos en la detecció de col·lisió del *Star Fighter 68000*.

Vam tenir dubtes a l'hora d'integrar el so, a vegades no es sentia i d'altres no aturava de reproduir-se. Finalment varem solventar el problema de manera que el so va quedar coordinat amb el joc i els seus moviments. Encara així, no va suposar grans dificultats, i no era obligatori, per tant sempre haguéssim pogut no implementar les funcions de so.

## Principals afegits

El primer afegit que se'ns va ocórrer fou el d'augmentar la velocitat progressivament, pel que la velocitat de les canonades ara depèn del nombre de canonades mostrades (*CANOCOUNT*).

Un altre afegit fou el dels sons al travessar una canonada de manera exitosa i el renou al xocar contra una canonada.

També tenim un arxiu anomenat *FONS.X68* que correspon al fons de pantalla del joc a l'estat *STAPLAY*.

El forat de la canonada es crea en una posició Y aleatòria usant la subrutina *UTLRAND* proporcionada.

Per simular l'aleteig de l'ocell s'usa la variable *ALAONO* i la subrutina *ALTERNAR* (dins *UTIL.X68*), aquesta alterna de valor de *ALAONO* entre 1 i 0, cada *#ALATEIG* cicles (per defecte 12). Al pintar l'ocell, depenent del valor de *ALAONO*, es pinta l'ala baixa o l'ala alta.

Un possible afegit seria el de recollir monedes, que serien agents i es detectaria que s'han recollit de manera semblant a com es detecta la col·lisió a les canonades, un cop recollides es mataria l'agent i s'incrementaria algun comptador de monedes, o directament s'afegiria la puntuació de cada moneda a la puntuació global del joc.

Es va intentar afegir alguna mena de registre de puntuacions en forma de lectura i escriptura a un fitxer però no vam trobar cap manera senzilla de fer-ho, i, com que a cap exemple proporcionat estava fet vam suposar que no és una tasca que haguéssim de realitzar.

La mida de la pantalla per defecte és de 960x480, però el joc funciona igual amb altres resolucions.

## Com jugar

Un cop iniciat el joc, primer visualitzarem la pantalla d'inici, on es mostra el nom del joc i els autors.

Per començar la partida, haurem d'espitjar el botó d'**espai** i d'aquesta manera canviarem de pantalla a l'habitual on veurem com l'ocell avança cap a les canonades (en realitat són aquestes les que avancen cap a l'ocell).

Per moure l'ocell i passar entre els forats de les canonades haurem de fer ús de les **quatre fletxes**, d'aquesta manera podrem travessar les canonades i contrarestar la gravetat.

A cada canonada superada, la velocitat augmenta linealment, pel que la dificultat augmenta fins que arriba un punt on el jugador no pot travessar la canonada i perd la partida, es mostra la puntuació obtinguda i es demana si vol tornar a intentar-ho.

El fet que la velocitat de les canonades augmenti però la del moviment de l'ocell no, és un fet intencionat, d'aquesta manera el joc sempre acabarà i serà impossible de *rompre* o d'arribar a un estat no controlat. Nosaltres, amb la configuració per defecte, no hem estat capaços de superar els 45 punts.

## Conclusions

S'ha realitzat un joc totalment funcional que ens ha permès comprendre com funciona l'entrada i sortida en un simulador del 68K. A més, hem consolidat les funcions gràfiques i hem pogut aprofundir en el tema de les interrupcions i comprendre de primera ma com s'implementen en un cas real.

Hem pogut comprovar que anteriorment quan es feien els programes amb aquest llenguatge o similars resultava molt més costós que en l'actualitat amb els llenguatges de programació d'alt nivell. És per aquest motiu que hem acabat realitzant un joc bastant simple, la dificultat de l'ús d'un llenguatge d'aquest nivell ha estat un important coll de botella a l'hora de realitzar la pràctica, fins i tot tenint tres jocs d'exemple i una estructura perfectament definida. Els jocs disponibles a la pròpia web de l'*Easy 68K* no han aportat pràcticament cap ajuda.

## Opinió

En primer lloc, no és una pràctica curta, però si tens clar el que has de fer i les expectatives baixes, es pot aconseguir. Definitivament ha estat una pràctica complicada, sobretot per les mateixes limitacions del llenguatge, però amb paciència i amb els jocs d'exemple (sobretot els jocs d'exemple) hem aconseguit produir un joc modest.

Som conscients de que no és el joc més complicat que podríem haver creat, però amb el temps que tenim i, donat que és el primer que hem fet, estem satisfets.

Ens ha donat una visió aproximada de la complexitat associada a realitzar projectes d'aquest estil, com per exemple els jocs *YAR* i *YAP*.

Un cop acabat, ens hem adonat que si ara haguéssim de realitzar un altre joc, tardaríem molt manco, ja tenim l'estructura realitzada, moltes parts del codi reutilitzables i moltes lliçons apreses.

És possible que per solventar la manca de temps tota la part pràctica de l'assignatura estés enfocada directament a la pràctica final, és a dir, a cada tema pràctic, realitzar la feina però que els resultats d'aquesta serveixin directament a la pràctica final, en forma de mòduls, subrutines o el que calgui. D'aquesta manera els alumnes tendrien feina avançada alhora de realitzar la pràctica final o, fins i tot, es podria començar la pràctica final des del principi de curs.