

## Estrutura geral do projeto

O projeto é baseado num popular jogo para *smartphones*, Flappy Bird, que consiste no movimento de um pássaro na vertical. O objetivo do jogo é passar pelo maior número de obstáculos possível, de forma a obter a pontuação máxima. Para tal, o jogador não poderá chocar com nenhum dos limites do ecrã nem com nenhum dos obstáculos.

Em termos de implementação no P3, o código foi dividido em diversas sub-rotinas, que serão chamadas na rotina principal do jogo. Entre elas, as mais importantes são as seguintes:

- *SobePassaro* (rotina que faz o pássaro subir);
- *PassaroCai* (rotina que faz o pássaro cair)
- *Toca\_limite* (rotina que verifica se o pássaro toca em algum dos limites);
- Rotinas associadas à geração e movimentação dos obstáculos (*CriaObs*, *MoveObs*, *LimpaObs*, *PrintObs*);
- *Colisoes* (rotina que verifica se o pássaro colide com os obstáculos)

## Variáveis principais

PosPassaro	Posição inicial do pássaro
ContQueda	Intervalo de tempo que ocorre para a próxima queda
ContMoveObs	Intervalo de tempo para mover os obstáculos
ContCriaObs	Intervalo de tempo para criar os obstáculos
PonteiroObs	Indica onde os obstáculos serão colocados na tabela
EspacoObs	Espaço entre os limites dos obstáculos

## As rotinas *SobePassaro* e *PassaroCai*

Estas rotinas são as responsáveis pelo movimento na vertical do pássaro, algo essencial no jogo. Tanto para o pássaro subir, como para o pássaro descer, é utilizado o mesmo método de representação na janela de texto: incrementa-se ou decrementa-se a posição do pássaro e chama-se a rotina *Esc\_string*, responsável por escrever qualquer carácter (ou cadeias de caracteres) na janela de texto. Ao mesmo tempo é apagada a posição anterior do pássaro.

De acordo com o enunciado do projeto, para o pássaro descer utilizar-se-ia a força da gravidade. No entanto, devido a um erro de código, não foi possível implementar tal força no nosso código, e utilizámos, portanto, o regular temporizador. A implementação da gravidade basear-se-ia na fórmula do movimento uniformemente acelerado.

## A rotina *Toca\_limite*

Um dos objetivos principais do jogo é não tocar em nenhum dos limites, tanto superior ou inferior. Para verificarmos tal, a rotina *Toca\_limite* compara a posição atual do pássaro com ambos os limites – caso a posição do pássaro coincida com a dos limites, é chamada a rotina *FimJogo*, que apaga tudo o que está na janela de escrita e devolve a mensagem de fim de jogo.

## **Geração, movimentação e deteção de colisão de obstáculos (rotinas *CriaObs*, *MoveObs*, *LimpaObs*, *PrintObs* e *Colisoes*)**

Para gerar os obstáculos são utilizadas diversas rotinas. O método do nosso código baseia-se numa tabela que tem um número máximo de obstáculos a serem visualizados no ecrã, com um espaçamento definido entre estes. Assim, são gerados os obstáculos até preencher a janela de texto, que são depois desenhados na coluna da tabela seguinte e apagados na posição anterior, de modo a dar a sensação de movimento.

Uma das outras formas de fim de jogo é caso o pássaro colida com os obstáculos. Para tal, a rotina *Colisoes* verifica se isto ocorre, e caso aconteça, é invocada a função que apaga o que está escrito na janela de texto e escreve a mensagem de fim de jogo.

## **Conclusão**

Em suma, a implementação do projeto foi feita utilizando sub-rotinas que geram uma rotina principal. O nosso grupo encontrou algumas dificuldades tanto em gestão de tempo como em execução do código, o que levou à inconclusão do projecto. Faltou a implementação de algumas funcionalidades pedidas pelo enunciado, nomeadamente a implementação da rotina que permitia originar o espaçamento aleatório de obstáculos, a implementação da gravidade, a inicialização dos LEDs bem como o sistema de pontuações. Nos primeiros dois, apesar de várias tentativas, originou sempre um erro, pelo que decidimos avançar e implementar outras funcionalidades.

Apesar das dificuldades, o balanço da realização foi positivo, visto que o que está implementado no código aparenta funcionar corretamente, com comentários pertinentes e boa organização do mesmo.