



TÉCNICO
LISBOA

RELATÓRIO DO PROJETO “PÁSSARO BAMBOLEANTE”

Realizado por:

Rodrigo Domingues Oliveira n° 83558

João Carlos Gomes Freitas n° 83478

Hugo Rafael Silva Guerreiro n°83475

1) Organização do programa:

O programa está dividido em 2 zonas:

- Zona1: Definição de constantes, variáveis e tabela de interrupções
- Zona2: Código

Dentro da Zona1 é de referir a importância do espaço reservado para as colunas. Estão reservados 17 espaços de memória para a manipulação das colunas. Cada um desses espaços irá conter uma posição correspondente à posição do início da abertura da coluna de cima da respetiva coluna no ecrã.

Dentro da Zona2, as funções de maior importância são:

- “Colunas” – função que move as colunas
- “Geracol” – Função que gera novas colunas
- “Colisao”- Função que verifica se houve colisão
- “Gravidade”- Função que modifica a taxa de atualização da função “Cair”
- “Cair” - Função que atualiza a posição do pássaro

2) Explicação do código geral

O funcionamento do jogo segue uma estrutura simples:

- 1) Inicia o jogo
- 2) Ciclo do jogo
- 3) Fim do jogo

No início do jogo inicializamos os registos mais importantes e o temporizador, escrevemos a tela inicial e preparamos o jogo para iniciar o ciclo. É no início de jogo que também geramos o valor aleatório usado pela função “Random”. Para gerar este valor, incrementamos uma variável no ciclo que aguarda a interação do jogador para inicializar o ciclo de jogo.

No ciclo de jogo, verificamos se várias flags foram ativas e se os contadores atingiram determinados valores. Quando algum valor é o esperado, é executado o código associado aquela flag ou contador. É neste ciclo que são realizadas as funções críticas ao funcionamento do jogo. Quanto às flags, podem ser alteradas de várias formas:

- Pelo temporizador
- Pelos botões de interrupção
- Por outras funções

No fim de jogo, escrevemos a tela de ecrã final e damos duas opções ao jogador, ou termina o jogo (apaga tudo na tela e na placa e entra num ciclo infinito, terminando qualquer possível interação), ou reinicia o jogo.

OBS.: Por motivos de eficiência, optamos por deixar a posição do corpo do pássaro guardada sempre no registo R7.

3) Explicação das funções mais importantes

Colunas: Esta função é responsável por mover cada coluna uma posição para a frente no ecrã. Internamente, a função avalia o número de obstáculos escritos no ecrã e conforme esse número vai a cada posição de memória que contém a posição da coluna e escreve essa coluna uma posição à frente no ecrã.

Geracol: Esta função é responsável por gerar uma nova coluna aleatória. Internamente, a função vai a cada posição de memória que contém as posições de cada coluna no ecrã e move o conteúdo da memória anterior para a memória seguinte, assim sucessivamente até ter libertado a memória inicial, de modo a poder escrever nela a nova coluna gerada.

Colisao: Esta função é responsável por verificar se houve colisão entre o pássaro e a coluna. Internamente, esta função recebe a coluna atual que o pássaro está a passar e vai comparar o valor da linha em que o pássaro está, com cada posição da abertura entre as colunas. Caso algum valor seja igual, significa que o pássaro está entre a abertura, caso contrário significa que o pássaro colidiu com a coluna. Esta função é chamada cada vez que as colunas são atualizadas, é feito um salto do pássaro ou o pássaro cai.

Fundamentação das rotinas *Gravidade* e *Cair*: A gravidade implementada tem como base a definição física de aceleração e velocidade, ou seja, a aceleração é a variação da velocidade e a velocidade é a variação da posição. Através da definição de Movimento Rectilíneo Uniformemente Acelerado, é possível concluir que o movimento do corpo é caracterizado pela variação linear da velocidade. Assim, o movimento do pássaro neste projeto foi feito de forma a fazer variar o tempo necessário para o pássaro descer uma linha, em que esta variação é feita de forma constante. Inicialmente, o pássaro precisa de 0.5s* para descer uma linha, e de 0.5s* para atualizar o tempo necessário para chamar a rotina *Cair* (a rotina *Gravidade* faz a atualização do tempo necessário para descer uma linha), ou seja, a cada 0.5s o pássaro irá precisar de -0.2s* relativamente ao tempo anterior para descer uma linha. Uma vez que a taxa de atualização da posição do pássaro atinge o valor mínimo de 0.1s, devido a limitações do P3, foi preciso implementar a seguinte funcionalidade: a cada 0.5s* a taxa de atualização do pássaro deixará de ser -0.2s* relativamente a anterior (uma vez que o mínimo é de 0.1s) e será incrementado o valor de quedas múltiplas do pássaro a cada 0.1s, ou seja, ao invés de descer apenas 1 linha, passará a descer 2 linhas e assim por diante. Para melhor demonstração, segue um algoritmo em Python que descreve o número de linhas y que o pássaro desceu após um certo tempo.

```
tempo = 20
t = 0
conta = 0
contc = 0
atualizaa = 5
atualizac = 5
fatora = 2
cairn = 0
y=0
for i in range(tempo+1):
    if(conta == atualizaa):
        if(atualizac>2):
            atualizac = atualizac - fatora
        else:
            cairn = cairn + 1
            conta = 0
    if(contc >= atualizac):
        for i in range(cairn+1):
            y = y + 1
            contc = 0
        print("y:", y, "t:", t, "ContA:", conta, "ContC:", contc)
        conta = conta + 1
        contc = contc + 1
        t = t+1
```

Assim, conclui-se que o movimento do pássaro neste projeto é uma boa aproximação do gráfico que representa o movimento do pássaro de acordo com as leis da física.

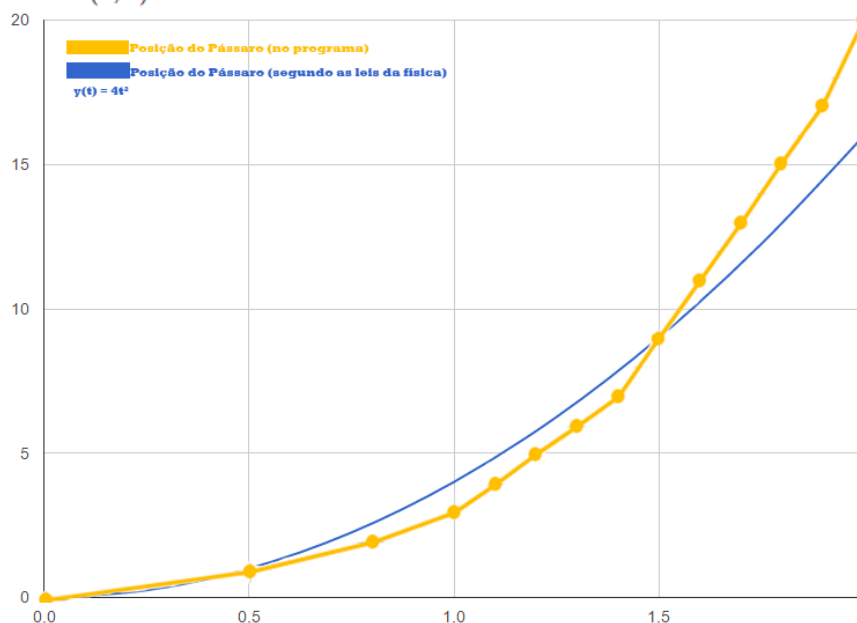
* - Valores escolhidos para uma melhor experiência do jogador.

O algoritmo foi usado para criar uma comparação das posições do pássaro neste projeto com o gráfico que representa o movimento do pássaro de acordo com as leis da física.

Uma vez que a cada 0.5s diminuimos a taxa de atualização em 0.2s, é possível concluir que a aceleração do pássaro é de 8 linhas / segundo²:

Assim: $y(t) = 4t^2$

$$a = \frac{2}{(0,5)^2} = 8 \text{ linhas/segundo}^2$$



4) Funcionalidades extra

A única funcionalidade extra que considerámos pertinente implementar foi a de reinicializar o jogo.

Decidimos ter esta funcionalidade uma vez que cada vez que o jogador perdia, teria de reinicializar manualmente o programa. Deste modo, esta função permite ao jogador reinicializar o jogo, diretamente no programa.