

# Descrição do projeto (parte 2)

---

Projeto feito pelo grupo nº51: Francisco Abrunhosa nº95580 e Diogo Venâncio nº95555

Neste projeto foi realizado um programa em assembly para simular o movimento de um corpo num plano inclinado supondo que o tempo está discretizado em intervalos de tempo constantes e sendo representada no terminal do simulador do P4.

## *Inicialização de estruturas:*

Foram declaradas várias variáveis, constantes e outros dados para controlarem o ciclo e regirem as funções principais.

Foi escrito no terminal as paredes da simulação apenas uma vez para tornar o código mais eficiente.

Foram também inicializadas outras estruturas essenciais ao ciclo como o stack pointer, o cronómetro e as variáveis para serem usadas nos ciclos.

## *Verificação e controlo do ciclo:*

O cronómetro vai colocar uma variável a 1 (FLAG\_CONTROLO) de modo a fazer o controlo do ciclo aquando da sua verificação.

Se o valor da variável for 1, o ciclo é executado, se não, volta a fazer a verificação até passar 1 décima de segundo (intervalo do cronómetro).

No final da rotina VERIFICA\_ATLZ, é decrementado este valor para dar *reset* no ciclo.

## *Actualização da posição da bola:*

Antes de executar o ciclo, é eliminada a posição da bola do ciclo anterior. Nesta parte do código também é feita uma verificação da posição anterior de modo a não apagar as paredes.

Executa as funções de aceleração, velocidade e posição realizadas na primeira parte do projeto. Na função da aceleração é usado o *output* do acelerómetro como forma de *input* da aceleração que depois é multiplicado por  $g / 255$  para ajustar o seu valor. Na função da velocidade, é usada a aceleração calculada anteriormente de modo a obter a velocidade naquele instante e por fim, na função da posição, é usada a velocidade para actualizar a posição da bola.

## *Representação no terminal:*

Antes de a bola ser representada no terminal, é executada uma rotina de verificação da sua posição (VRF\_POS) para fazer com que a bola não ultrapasse as paredes. Nesta função a sua velocidade passa ao simétrico e é retirado o “excesso” da sua posição, adicionando este valor ao limite da parede (78) de

modo a proporcionar um efeito de “*bounce back*” ao utilizador. É também guardado o valor da parte inteira da posição numa variável auxiliar (POS\_ATLZD) para depois ser usada na função seguinte.

Na função PRINT, a bola é desenhada no terminal através da variável POS\_ATLZD e de um OR com o valor 0100h para a colocar na linha/coluna correspondente.

#### *Bugs conhecidos:*

- Se for utilizada uma aceleração muito próxima da aceleração máxima, a bola ou fica parada, ou fica muito rápida e por isso, a experiência para o utilizador passa a ser negativa.
- Devido aos arredondamentos e conversões a bola acaba por nem sempre ter a posição mais precisa