

# Relatório de Tecnologia da Informática

Jogo: Super Bit Smasher



UNIVERSIDADE D  
COIMBRA

André Filipe Costa Colaço

UC - 2020220301

## Introdução

O objetivo deste jogo consiste em chegar de um inteiro de 8 bits até a outro inteiro de 8 bits, gerados aleatoriamente, através de 3 operadores bitwise, o AND, o OR ou o XOR. A disponibilidade destas operações, depende de outro número, também gerado aleatoriamente. Também existem 4 led's que se vão desligando à medida que o tempo passa. O jogo acaba se o tempo expirar ou se o jogador conseguir transformar o valor inicial no valor target.

## Início do jogo

Primeiro, os 4 led's são ligados e o tempo é resetado (para o caso do jogo estiver a recomçar). Depois, criei uma função chamada *“gerar\_numeros”*. Esta função gera 3 números, dos quais 2 deles (o valor inicial e o valor target) vão de 1 a 256, exclusive, e o outro número (que vai determinar quais operações estão disponíveis) vai de 1 a 4, exclusive. Posto isto, são impressos os valores inicial e target, em binário, e também quais as operações disponíveis para este *round*.

## Leitura de dados no Serial Monitor

Para ler o que é escrito no Serial Monitor, criei uma função chamada *“ler”*. Esta função começa por imprimir uma mensagem para avisar o jogador de que pode escrever o número no Serial Monitor. De seguida, quando o jogador escrever o número, este é lido e passado para inteiro pois quando o número é escrito, está no tipo *string*. Por fim, o número é apresentado ao jogador, em binário.

## Botões de AND, OR e XOR

Para os três botões, criei outra função chamada *“buttons”*. Esta função começa por imprimir uma mensagem para avisar o jogador de que pode premir um botão. Posto isto, se o jogador premir um botão e se este tiver disponível, ou seja, se o operador referente a esse botão estiver disponível, é feita a operação pertencente a esse botão ao valor inicial e o número

escrito pelo jogador. Por fim, é apresentado ao jogador o resultado desta operação.

## Led's

Ora, primeiro defini o tempo do jogo para 60 segundos. Depois, criei uma última função chamada "leds". Como cada led simboliza  $\frac{1}{4}$  do tempo limite, então quando passa  $\frac{1}{4}$  do tempo, o primeiro led é apagado. Depois, quando passa metade do tempo, o segundo led é apagado e, posto isto, quando passa  $\frac{3}{4}$  do tempo, o terceiro led é também apagado. Por fim, quando o tempo limite é excedido, o último led apagasse, é impressa uma mensagem a informar que o jogo acabou e o este recomeça.

## Conclusão

Para concluir, achei este projeto mais fácil que o primeiro. Não tive dificuldades em nenhuma parte do código, visto que, já tinha tirado todas as dúvidas no primeiro projeto, facilitando este. No entanto, o jogo em si é a parte mais complicada para mim. Ainda não consigo fazer as conversões e as operações com a máscara, de cabeça, tão rápido como devia. Mas é algo que consigo treinar com este jogo.