

### **Jogo: *Bit Math***

O objetivo deste jogo é acertar no resultado da operação entre dois números codificados em formato binário. É executada uma de três operações binárias AND, OR ou XOR, que é indicada a partir de uma cor do led RGB. O resultado da operação é indicado pelo número de vezes que o botão é pressionado, havendo um tempo limite para se indicar o número.

O jogo tem como finalidade promover o conhecimento de operadores binários e da conversão entre sistemas de numeração, sendo inspirado em iniciativas idênticas da [Cisco](#).

#### **Devem ser implementadas as seguintes funcionalidades:**

##### **1. Uso do botão “Guess Button”:**

- 1.1. Inicialmente serve para iniciar o jogo (depois da mensagem “Press Button to Start!”). Esta pressão inicial não deve ser contabilizada e deve ativar o temporizador associado ao tempo necessário para indicar o resultado correto (definido para 7 segundos).
- 1.2. O jogador indica o número resultante da operação através de pressões/click(s) no botão. Deve ser usado o debounce e contabilizado de forma correta o número de pressões/click(s).
- 1.3. A pressão prolongada do botão (superior a 1 segundo) ativa a funcionalidade recomeçar/reset (ver ponto 3).

##### **2. Números e operação:**

- 2.1. Os números devem ser gerados de forma aleatória quando o programa inicia. Só devem ser gerados números consoante o número de leds disponíveis para cada número.
- 2.2. Os números devem ser apresentados em formato binário, em que os leds a vermelho correspondem ao número 2 e os leds de cor verde correspondem ao número 1.
- 2.3. A operação binária a realizar (AND, OR ou XOR) deve ser escolhida de forma aleatória quando o programa inicia ou quando se recomeça o jogo.
- 2.4. A operação AND corresponde à cor vermelha do led RGB, a operação OR corresponde à cor azul e a operação XOR corresponde à cor verde.

##### **3. Recomeçar/reset:**

- 3.1. Deve escolher novos números e escolher a operação, para iniciar novamente o jogo.
- 3.2. Deve acender e apagar os LEDS sequencialmente. O efeito de varrimento no número 2 deve ser da esquerda para a direita, e no número 1 deve ser da direita para a esquerda.
- 3.3. O led RGB deve piscar durante a operação de recomeçar.

##### **4. Jogo**

- 4.1. Deve ser apresentada uma só vez a mensagem que decorreu 50% do tempo.
- 4.2. Deve ser apresentada uma só vez a mensagem que decorreu 75% do tempo.
- 4.3. Se acertar no número dentro do tempo definido (7segundos) deve iniciar o modo vitória.
- 4.4. Se não acertar no número deve ser iniciado o modo de derrota.

##### **5. Modo Vitória**

- 5.1. Deve fazer piscar os leds dos números e o led RGB durante 5 segundos.
- 5.2. Após o piscar deve fazer o efeito de varrimento 3 vezes, consoante o descrito na funcionalidade 3.2. Terminado o varrimento deve reiniciar o jogo.

##### **6. Modo Derrota**

- 6.1. Deve apagar os leds dos números e só deve piscar o led RGB durante 5 segundos.
- 6.2. Após o piscar do led RGB deve reiniciar o jogo.

**Regras de montagem:**

O circuito associado ao jogo deverá ser montado de acordo com o seguinte:

- Os leds do número 2 devem ser ligados aos pinos de 10 a 13.
  - Os leds do número 1 devem ser ligados aos pinos de 6 a 9.
  - O botão de pressão deve ser ligado pino 2 e deve ser ligado em modo *INPUT\_PULLUP*.
  - A cor vermelha do led RGB deve ser ligada ao pino 5, a cor azul ao pino 4 e a cor verde ao pino 3.
- A não observância de uma ou mais regras de montagem implica um desconto de 20%.

**Demonstração de funcionamento em:** <https://youtu.be/llbvhd7cUfQ>

(vídeo ilustra uma ronda de jogo completa com vitória, seguida da demonstração de um *Reset* e termina com a demonstração de uma derrota)

O trabalho deverá ser realizado **individualmente** e submetido no inforestudante até dia 14/11, pelas 23:59.

**A submissão deverá ser feita num ficheiro .ZIP e deve incluir:**

- o código completo que foi desenvolvido para o jogo, formato ficheiro de texto *.TXT*;
- um diagrama/imagem do circuito, feito em tinkercad e em formato *.JPEG*;
- um relatório breve (máx. 2 páginas) documentando a implementação das funcionalidades, formato *.PDF*.

**NÃO UTILIZE A FUNÇÃO DE PARTILHA DO TINKERCAD NEM COLOQUE O SEU CIRCUITO COMO "PUBLIC".**

**NÃO ENTREGUE O TRABALHO COM LINKS PARA O PROJETO NO TINKERCAD, SOB PENA DE O SEU TRABALHO NÃO SER AVALIADO.**

**Sobre a avaliação:** O código desenvolvido, assim como o relatório serão validados por um sistema de deteção de plágio. Quaisquer situações irregulares serão tratadas em consonância com os regulamentos em vigor na UC. Os alunos/as poderão ser chamados a responder a questões sobre o trabalho (defesa), contando estas para a avaliação desta componente.

**Notas importantes:** este exercício é para ser resolvido com base na matéria ensinada até à semana de 02/11 (inclusive). O uso de *arrays* ou qualquer outro recurso que não tenha ainda sido abordado implica uma penalização de 20%. Não se permite o uso das funções *bitSet*, *bitRead*, *bitWrite*, *bitClear*, implicando uma penalização de 25%. Uso de funções, comentários e indentações corretas serão considerados para efeitos de avaliação.