Este projeto consistiu na elaboração de uma versão do jogo **Mastermind**, onde números são utilizados em vez de cores, em *Assembly*, para correr na placa do processador P3. A sua interface é constituida por duas partes: o *input*, que é feito através dos botões IA, I1, I2, I3, I4, I5 e I6, e do *output*, que é feito através da janela de texto, do *display* de cristal líquido, do *display* de 7 segmentos e dos LEDs.

Inicialmente, a janela de texto apresenta uma janela de boas vindas, indicando que o jogador deve premir IA para começar o jogo. Durante o jogo, o utilizador deve premir os botões I1 até I6 de forma a inserir a sua jogada, que corresponde a uma sucessão de 4 dígitos. Cada jogada deve ser efetuada em menos de 8 segundos, cujo tempo remanescente é apresentado através dos LEDs. Caso o jogador não consiga efetuar a sua jogada no tempo indicado, perde o jogo e terá que recomeçar clicando, novamente, em IA. No *display* de 7 segmentos é apresentado o número de tentativas efetuadas em cada jogo, sendo o máximo 12. Se o jogador atingir esse máximo, perde o jogo. No LCD é apresentada a melhor pontuação até ao momento, que corresponde ao jogo ganho com um menor número de tentativas.

A **primeira parte do nosso código** constitui-se pela definição de inúmeras constantes, seguida da tabela de interrupções. Depois, alocámos em memória as variáveis necessárias ao decorrer do jogo. Finalmente, começamos por definir todas as funções relacionadas com o decorrer do jogo sendo que a lógica principal do jogo se encontra no fim do ficheiro.

Alguns dos aspetos mais relevantes da implementação são:

* O número aleatório inicial é gerado de acordo com o número de repetições do ciclo *Infinity* antes do jogo começar. Este valor serve, inicialmente, como a “jogada anterior” necessária ao algoritmo de geração de números pseudo-aleatórios especificado no enunciado da primeira parte do projeto.
* A comparação entre a sequência do jogador e a sequência secreta é feita da seguinte forma: primeiro compara-se os digitos certos na posição certa e coloca-os a zero para que não causem interferência durante a restante comparação. De seguida, verifica-se os que existem mas que estão na posição errada. Finalmente, executamos a subtração entre 4 e a soma dos dígitos certos nas posições erradas e certas para obter o número de dígitos que estão na jogada do jogador, mas que não estão na sequência secreta.
* A comparação da jogada do jogador com a sequência secreta é armazenada em 12 bits: os 4 bits menos significativos armazenam o número de dígitos certos na posição certa, os segundos 4 bit o número de dígitos certos na posição errada e os terceiros 4 bits armazenam o número de digitos não encontrados. O nosso grupo tem a noção de que são necessários apenas 8 bits para armazenar esta informação visto que o número de dígitos não encontrados poderia ser suprimido visto ser o resultado da subtração entre 4 e a soma dos dígitos certos. No entanto, deicidimos codificar em 12 bits para simplificar a impressão: em variáveis de memória armazenamos os valores ASCII dos caracteres ‘x’, ‘o’ e ‘-‘. De seguida, na função *PrintTip*, recorrendo a rotações e a incrementos de endereços de memória, associamos cada conjunto de 4 bits a seu caracter.

Relativamente às tarefas pedidas no protocolo do projeto, o nosso grupo pensa que realizou as mesmas com sucesso e de forma eficiente, tentando sempre, depois de realizadas, olhar para as mesmas de forma a reduzi-las e melhorar a sua eficiência no aspeto global do projeto.

É importante notar que existiam alguns aspetos não especificados no protocolo e, por isso, o nosso grupo tomou em consideração as hipóteses, escolhendo sempre a que melhorava a experiência de utilizador: **(1)** decidimos que os LEDs se deveriam apagar da direita para a esquerda; **(2)** o botão IA, para além de dar início ao jogo no início e após o final de um jogo, permite também fazer *restart* ao jogo, com uma nova sequência, possibilitando assim que o jogador reinicie o jogo a qualquer momento e que não seja obrigado a concluir o que está em progresso; **(3)** o grupo decidiu mostrar um *splash screen* no início do jogo para permitir ao utilizador entender qual é o jogo de que se trata e possibilitar uma experiência mais imersiva. Além disso, **(4)** o LCD indica a melhor pontuação após a mensagem “Melhor Pont:”.

Em suma, como balanço final do nosso projeto, o nosso grupo pensa que o projeto foi realizado com sucesso, tanto o processo de desenvolvimento, trabalho em equipa e divisão de tarefas, como o resultado final. Focando-nos nos pontos mais importantes e numa fase posterior no seu aperfeiçoamento, permitindo utilizar todos os conhecimentos que nos tinham sido lecionados até ao momento.