

Trabalho de Programação 2 Processador CESAR16i

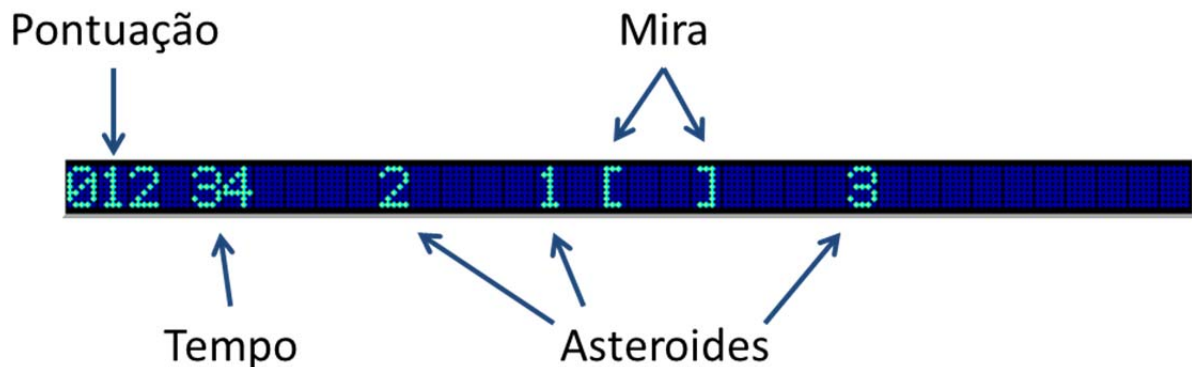
1. Descrição Geral

Sua última nave atravessou o campo de asteroides com louvor. Entretanto, a quantidade de combustível despendida no processo foi muito grande. Para tentar melhorar isso, você instalou em sua nave um canhão de LASER, para destruir os asteroides e não ter que desviar, poupando o combustível.

Mas, para que sua estratégia possa funcionar, você terá que exercitar sua habilidade de acertar os asteroides. Para isso, você deverá desenvolver para o computador CESAR um simulador de canhão LASER.

Esse simulador deverá “lançar” asteroides no espaço, nos quais você terá que “fazer mira” e atirar com o LASER. Além disso, o simulador deverá realizar a contagem dos asteroides destruídos durante um certo período de tempo. Isso será usado para verificar a melhoria de sua habilidade.

Na figura abaixo está reproduzido o visor do CESAR, quando operando como simulador de canhão LASER. Na figura, pode-se ver o número “012”, que representa a **pontuação** atingida pelo operador do canhão, e o número “34” que representa **tempo**, em segundos, já decorridos. Na figura também pode-se ver a “**mira**” do canhão, formada pelos caracteres “[” e “]”, e os dois espaços entre eles que é a área onde os asteroides serão atingidos. Finalmente, pode-se ver três **asteroides**, representados pelos números 2, 1 e 3.



Como se pode observar na figura acima, a pontuação deverá ser escrita nos três primeiros caracteres mais à esquerda do visor; então deve-se deixar um caractere em branco. Segue o tempo de simulação, em segundos, com dois caracteres. Finalmente, os 30 caracteres restantes serão usados para representar o campo de asteroides. Nessa área ocorrerá a animação de movimento dos asteroides e da mira. Nessa área é onde o operador do canhão poderá atingir os asteroides. Notar que essa área inicia (à esquerda) na sétima posição e termina na 36ª posição, as quais correspondem às coordenadas “0” até “29” do campo de asteroides, respectivamente. Essas coordenadas são usadas para indicar a localização da “mira” e dos “asteroides”.

Para atingir os asteroides o operador deverá posicionar a mira sobre o alvo e disparar o canhão, usando a tecla SPACE (caractere ASCII 32₁₀ ou 20₁₆). A mira está sobre um asteroide quando o asteroide estiver visível no espaço entre os caracteres “[” e “]”.

1.1. Sobre a implementação

A implementação deverá utilizar o mecanismo de interrupção de tempo disponível no CESAR16i para realizar a animação dos asteroides e o desenho da “mira”. Também deverá ser utilizada a interrupção de teclado para receber os comandos do operador: movimento da mira para a esquerda e direita e disparo do LASER.

Sua implementação deverá ser dividida em duas partes: **Tratador de Interrupção (de tempo e teclado)** e **Programa Principal**.

1.2. Primeira parte – Tratadores de Interrupção

A função do **tratador de interrupção de tempo** é incrementar o temporizador de tempo de simulação, fazer a animação dos asteroides e desenhar a mira no visor.

A função do **tratador de interrupção de teclado** é filtrar as teclas digitadas e repassar apenas aquelas que correspondem aos comandos para o simulador.

1.3. Segunda parte – Programa Principal

A função do **programa principal** é iniciar uma simulação e apresentar no visor a pontuação e o tempo de simulação. Também é responsável por lançar os asteroides, escolher o seu tipo e verificar se foram atingidos pelos tiros de LASER.

1.4. Sobre a animação

A animação dos asteroides assim como a mira é feito de forma semelhante ao programa ANIMADO visto em aula. Para dar a “ilusão” de movimento, o programa deve apagar o(s) último(s) caractere(s) desenhado(s) no visor e redesenhá-lo(s) na nova posição.

Além da realização dessa operação, deve-se observar a periodicidade com que ela é realizada. Quanto menor for o tempo entre duas dessas operações (apagar e redesenhar), mais rápido será a sensação de movimento dos asteroides no visor, tornando mais difícil (ou até impossível) acertá-los. Esse é o caso de se apagar e redesenhar os asteroides a cada entrada de interrupção de tempo (100ms). Assim, um bom tempo para a animação é de 200ms ou mais (duas ou mais interrupções). Na sua implementação você poderá escolher a periodicidade que desejar para a animação. Mas lembre-se: você tem que testar o programa! E isso só será possível se você conseguir “acertar” os asteroides.

1.5. Arquivo base (LASER.CED) e entrega do trabalho

Para a realização do trabalho você receberá um arquivo chamado “LASER.CED”, que você deverá utilizar como base para o seu desenvolvimento. Nesse arquivo estão declaradas todas as áreas de memória, os endereços de acesso aos periféricos, uma rotina de geração de números aleatórios (chamada “**random**”) e as variáveis usadas na comunicação entre os tratadores de interrupção e o programa principal.

A entrega do trabalho será realizada em DUAS PARTES, cada uma em uma data diferente. Na primeira parte deverão ser entregues **apenas** os tratadores de interrupção de tempo e de teclado. Na segunda parte deverá ser entregue **apenas** o programa principal.

2. Divisão do espaço de endereçamento (alocação de memória)

Para realizar a implementação do trabalho o espaço de endereçamento do CESAR deverá ser dividido da seguinte forma:

Nome	Faixa de Endereços (em hexadecimal)	Descrição
APP	0000 – 7FFF	Área para colocar o PROGRAMA PRINCIPAL (PP). Aqui devem ser colocadas as rotinas, variáveis e a pilha do programa principal.
AVR	8000 – 80FF	Área para colocar as VARIÁVEIS DE COMUNICAÇÃO entre o programa principal e os tratadores de interrupção.
ARI	8100 – 81FF	Área para colocar a rotina de inicialização de variáveis das interrupções. Essa rotina deve iniciar no endereço 8100 ₁₆ . Essa rotina deve ser chamada no programa principal, ANTES DE HABILITAR AS INTERRUPÇÕES.
ATI	8200 – FF7F	Área para colocar os TRATADORES DE INTERRUPÇÃO de tempo e de teclado. Aqui devem ser colocadas as rotinas e variáveis usadas pelos tratadores de interrupção.
APR	FF80 – FFFF	Área reservada para os endereços dos periféricos.

A definição dessas áreas é realizada através da diretiva ORG. Essa definição está pronta no arquivo “LASER.CED” fornecido.

3. Variáveis de Comunicação

A área de memória **AVR** é reservada para as variáveis que serão usadas na troca de informações entre o Programa Principal e os Tratadores de Interrupção. As variáveis e estruturas definidas para essa comunicação estão declaradas no arquivo “LASER.CED” fornecido, e estão descritas a seguir.

TIMER – uma palavra de 16 bits a partir do endereço 8000₁₆. O tratador de interrupção de tempo deve incrementar essa variável a cada segundo.

TECLA – uma palavra de 16 bits a partir do endereço 8002₁₆. O tratador de interrupção de teclado deve receber as teclas digitadas, descartar aquelas que não correspondem a teclas válidas e colocar nessa variável apenas as válidas (os 8 bits mais significativos desta palavra ficarão com 0s e os menos significativos terão o código ASCII do caractere digitado). O programa principal recebe os comandos do usuário através dessa variável.

MIRA – três palavras de 16 bits a partir do endereço 8004₁₆. Representa a mira do canhão LASER. O tratador de interrupções de tempo é responsável por colocar a “mira” no visor, enquanto que o programa principal é responsável por informar as coordenadas do visor onde colocá-la.

ASTEROIDES – 15 palavras de 16 bits a partir do endereço 800A₁₆. Representam 3 asteroides, cada um descrito com 5 palavras. O tratador de interrupções de tempo é responsável pela animação dos asteroides, apagando-os e redesenhando-os no visor. O programa principal é responsável por criar os asteroides e verificar se o asteroide foi destruído.

4. Uso da variável “TIMER”

A variável TIMER tem por função fornecer ao programa principal a informação de tempo, em segundos.

O **tratador de interrupção** de tempo tem por tarefa incrementar essa variável a cada segundo. Não é necessário fazer qualquer forma de verificação de consistência do valor de TIMER.

O **programa principal** deverá:

- No início da simulação, zerar o conteúdo da variável TIMER;
- Sempre que essa variável for alterada, seu valor deverá ser atualizado no visor;
- Quando seu valor atingir 60 segundos, a simulação deverá ser encerrada.

5. Uso da variável “TECLA”

A variável TECLA tem por função permitir a passagem das teclas digitadas do tratador de interrupção de teclado para o programa principal. Sempre que essa tecla tiver um valor negativo, significa que ela não tem nenhuma tecla válida; sempre que esse valor for positivo, então essa tecla possui uma tecla válida.

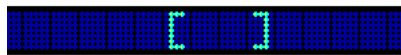
O **tratador de interrupção** de teclado é acionado sempre que o usuário digitar algo. Então, a tecla deve ser validada. As teclas válidas são SPACE (20₁₆), ENTER (0D₁₆) e as letras “I” (ou “i”), “A” (ou “a”), “D” (ou “d”), “P” (ou “p”), sendo que as letras minúsculas devem ser convertidas para maiúsculas, antes de serem passadas para o programa principal.

As teclas, depois de lidas, validadas e convertidas pelo tratador de interrupção de teclado, devem ser escritas na variável TECLA. Entretanto, isso só deve ser feito se o valor da variável TECLA for um número negativo. Essa restrição garante que o programa principal tenha tempo de processar a última tecla digitada, antes de receber uma nova tecla.

O **programa principal** saberá que nada foi digitado enquanto a variável TECLA tiver um valor negativo. Quando o valor dessa tecla for positivo, algo foi digitado e deve ser processado. Após ter lido o valor da variável TECLA, o programa principal deve escrever um número negativo na variável, avisando ao tratador de interrupção que a tecla já foi lida.

6. Uso da estrutura “MIRA”

A mira do canhão de LASER é desenhada no visor com as teclas “[” e “]”, separadas por dois espaços, conforme a figura abaixo:



A estrutura MIRA é formada por três palavras, que têm o seguinte significado:

Palavra 0 – ESTADO: Localizada no menor endereço de memória, controla como o tratador de interrupção de tempo deve desenhar a mira no visor. Os valores que podem aparecer nessa palavra são os números entre 0 e 3. Cada valor dessa variável indica como apagar e/ou desenhar a mira no visor.

Palavra 1 – POSICAO ATUAL: Contém a coordenada do visor onde a mira está desenhada. Essa palavra só tem significado quando ESTADO valer 2 ou 3. Essa coordenada representa a posição do caractere “[” da mira. Essa coordenada pode ser um valor entre “0” (posição mais à esquerda do campo de asteroides) e “26” (posição mais à direita);

Palavra 2 – NOVA POSIÇÃO: Contém a coordenada do visor onde a mira deve ser redesenhada. Essa palavra só tem significado quando ESTADO valer 1 ou 2. Essa coordenada representa a posição do caractere “[” da mira. Essa coordenada pode ser um valor entre “0” (posição mais à esquerda do campo de asteroides) e “26” (posição mais à direita).

Na tabela abaixo está descrito o comportamento de cada parte do programa, em função do valor de ESTADO.

Valor do ESTADO	Comportamento do tratador de interrupção e do programa principal
0	O tratador de interrupção não tem nada a fazer. O programa principal pode pedir que a mira seja desenhada no visor, escrevendo “1” em ESTADO. Notar que, antes de fazer isso, o programa principal deve escrever em NOVA POSIÇÃO a coordenada do visor onde se pretende que a mira seja desenhada.
1	O tratador de interrupção deve desenhar a mira, pela primeira vez, na coordenada NOVA POSIÇÃO e copiar esse valor para a POSICAO ATUAL. Portanto, não há nada a ser apagado. Além disso, deve escrever “2” em ESTADO. O programa principal não tem nada a fazer
2	O tratador de interrupção deve animar o desenho da mira. Para isso, deve-se apagar a mira da coordenada POSIÇÃO ATUAL e redesenhar a mira na coordenada NOVA POSIÇÃO. Depois disso, o valor de NOVA POSIÇÃO deve ser copiado para POSIÇÃO ATUAL. O programa principal pode pedir que a mira seja removida do visor, escrevendo “3” em ESTADO.
3	O tratador de interrupção deve apagar a mira que está desenhada na coordenada POSIÇÃO ATUAL. A mira não deve ser redesenhada. Além disso, deve escrever “0” em ESTADO. O programa principal não tem nada a fazer.

Notar que o **programa principal** tem por funções ativar a mira (escrevendo “1” em ESTADO), desativar a mira (escrevendo “3” em ESTADO) e alterar as coordenadas onde redesenhar a mira (escrevendo a nova coordenada da mira em NOVA POSIÇÃO). Também é responsável por garantir que as coordenadas de desenho da mira estejam, sempre, dentro da área reservada para isso.

7. Uso do vetor de estruturas “ASTEROIDES”

Esse vetor representa as informações de três asteroides, onde cada asteroide é representado através de uma estrutura formada por 5 palavras. A estrutura que representa o primeiro asteroide (asteroide “0”) ocupa as 5 primeiras palavras; a estrutura do asteroide “1” ocupa as próximas 5 palavras e o asteroide “2” ocupa as 5 palavras dessa área.

A estrutura que representa cada asteroide é composta, então, pelos seguintes elementos:

Palavra 0 - ESTADO: Localizada no menor endereço de memória, controla como o tratador de interrupção de tempo deve animar os asteroides no visor. Os valores que podem aparecer nessa palavra são os números entre 0 e 3. Cada valor dessa variável indica como apagar e/ou desenhar cada asteroide no visor.

Palavra 1 – POSIÇÃO ATUAL: Contém a coordenada do visor onde o asteroide está desenhado. Essa palavra só tem significado quando ESTADO valer 2 ou 3. Essa coordenada pode ser um valor entre “0” (posição mais a esquerda do campo de asteroides) e “29” (posição mais a direita).

Palavra 2 – NOVA POSIÇÃO: Contém a coordenada do visor onde o asteroide deve ser redesenhado. Essa palavra só tem significado quando ESTADO valer 1 ou 2. Essa coordenada pode ser um valor entre “0” (posição mais a esquerda do campo de asteroides) e “29” (posição mais a direita).

Palavra 3 – TIPO: Indica o tipo do asteroide, que é usado para determinar a forma do asteroide no visor, seu tamanho e o número de pontos ganhos quando for destruído. Existem 4 tipos de asteroides, numerados de 0 até 3. Portanto, deve-se considerar apenas os dois bits menos significativos dessa palavra; os bits restantes devem ser ignorados.

Palavra 4 – TAMANHO: Indica o tamanho atual do asteroide, ou seja, quantos “tiros” são necessários para destruí-lo. Sempre que o asteroide for atingido por um tiro, deve-se decrementá-lo, indicando a redução de seu tamanho. Quando TAMANHO chegar a zero, o asteroide foi destruído e deve ser removido da campo de asteroides.

Na tabela abaixo está descrito o comportamento de cada parte do programa, em função do valor de ESTADO.

Valor do ESTADO	Comportamento do tratador de interrupção e do programa principal
0	O tratador de interrupção não tem nada a fazer. O programa principal pode pedir que o asteroide seja desenhado no visor, escrevendo “1” em ESTADO. Notar que, antes de fazer isso, o programa principal deve escrever em NOVA POSIÇÃO a coordenada do visor onde se pretende que o asteroide seja desenhado.
1	O tratador de interrupção deve desenhar o asteroide, pela primeira vez, na coordenada

	NOVA POSIÇÃO e copiar esse valor para a POSICAO ATUAL. Portanto, não há nada a ser apagado. Além disso, deve escrever “2” em ESTADO. O programa principal não tem nada a fazer.
2	O tratador de interrupção deve animar o asteroide. Para isso, deve-se apagar o asteroide da coordenada POSIÇÃO ATUAL e redesenhá-lo na coordenada NOVA POSIÇÃO. Depois disso, o valor de NOVA POSIÇÃO deve ser copiado para POSIÇÃO ATUAL. O programa principal pode pedir que o asteroide seja removida do visor, escrevendo “3” em ESTADO.
3	O tratador de interrupção deve apagar o asteroide que está desenhado na coordenada POSIÇÃO ATUAL. Ele não deve ser redesenhado. Além disso, deve escrever “0” em ESTADO. O programa principal não tem nada a fazer.

7.1. Controle dos asteroides pelo tratador de interrupção de tempo

O tratador de interrupção de tempo tem por tarefa animar o movimento dos asteroides. Para isso, deve-se apagar a última coordenada onde ele foi desenhado e desenhá-lo novamente na nova posição. Os asteroides entram no campo de asteroides pelo lado direito do visor e devem ser animados para se movimentar para a esquerda, até saírem do campo de asteroides na posição mais a esquerda. Quando o asteroide sair do campo de asteroides, o programa principal deve pedir que seja removido. Deve-se tomar o cuidado para não permitir que os asteroides sejam desenhados fora da área reservada para isso.

Para determinar o caractere a ser usado para apresentar no visor cada asteroide, deve-se usar o TIPO do asteroide. O tipo do asteroide também determina o seu tamanho (quantos tiros deve receber para que seja destruído). Esse relacionamento está apresentado na tabela abaixo.

Tipo	Caractere	Tamanho
0	31 ₁₆ (49 ₁₀)	1
1	32 ₁₆ (50 ₁₀)	2
2	33 ₁₆ (51 ₁₀)	3
3	34 ₁₆ (52 ₁₀)	4

7.2. Controle dos asteroides pelo programa principal

O programa principal tem por tarefa criar novos asteroides, controlar se foram atingidos e atualizar a pontuação total do operador do canhão.

Para **criar novos asteroides** o programa deve chamar a função “**random()**” fornecida e usar o valor retornado para determinar **se deve criar** um novo asteroide e **qual deve ser o seu tipo**. Ao decidir por criar um novo asteroide, o programa deve procurar no vetor ASTEROIDES por um asteroide que tenha “ESTADO==0” e usá-lo na criação do novo asteroide. Se todos os elementos do vetor estiverem ativos, o programa não tem como criar um novo asteroide e deve abandonar a criação.

Ao criar um asteroide o programa deve realizar o seguinte procedimento:

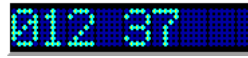
- Escrever o tipo do asteroide em TIPO;
- Escrever o tamanho do asteroide em TAMANHO;
- Escrever a primeira coordenada onde o asteroide deve aparecer em NOVA POSIÇÃO;
- Escrever “1” em ESTADO. **Essa deve ser a última ação no processo de ativação de um asteroide.**

Quando o operador disparar o canhão, o programa principal receberá uma tecla SPACE (ASCII 32₁₀ ou 20₁₆). Nesse momento, deve ser **verificado se um asteroide foi atingido**. Para isso, o programa principal deve verificar se as coordenadas que estão no espaço entre os caracteres “[” e “]” estão sobre um dos asteroides. Isso é feito comparando as coordenadas desse espaço com as coordenadas atuais de cada asteroide que tiver “ESTADO==2”. Se as coordenadas forem iguais, o asteroide foi atingido.

Nesse caso, o TAMANHO do asteroide atingido deverá ser decrementado. Se o resultado dessa operação for zero, o asteroide deverá ser removido e a pontuação correspondente ao tipo do asteroide deverá ser somada à pontuação total do operador. Para cada tipo corresponde uma pontuação. Na tabela abaixo REPETE-SE a tabela anterior e acrescenta-se a informação de pontuação fornecida por cada tipo de asteroide.

Tipo de asteroide	Caractere que representa cada tipo de asteroide	Tamanho de cada tipo de asteroide	Pontuação fornecida por cada tipo de asteroide
0	31 ₁₆ (49 ₁₀)	1	1
1	32 ₁₆ (50 ₁₀)	2	2
2	33 ₁₆ (51 ₁₀)	3	4
3	34 ₁₆ (52 ₁₀)	4	8

Quando um asteroide for destruído, o programa principal deve **atualizar a pontuação do operador**. Para isso, o programa deve colocar nos três caracteres mais a esquerda do visor a pontuação do usuário, conforme figura a seguir:



012 37

8. Qual sua tarefa na PRIMEIRA PARTE DO TRABALHO – implementação dos tratadores de interrupção

Nessa parte do trabalho você deve implementar os tratadores de interrupção de tempo e de teclado. Notar que os detalhes da implementação dos tratadores de interrupção estão descritos nos itens anteriores. A seguir está um resumo das funções que devem ser implementadas pelos dois tratadores de interrupção.

8.1. Tratador de interrupção de teclado

Esse tratador deve realizar as seguintes funções:

- Ler a tecla digitada pelo operador no teclado
- Realizar a filtragem da tecla lida e a conversão de minúsculas em maiúsculas
- Então, se "TECLA<0", escreve-se o valor resultante da validação e conversão na variável TECLA.

8.2. Tratador de interrupção de tempo

Esse tratador deve realizar as seguintes funções:

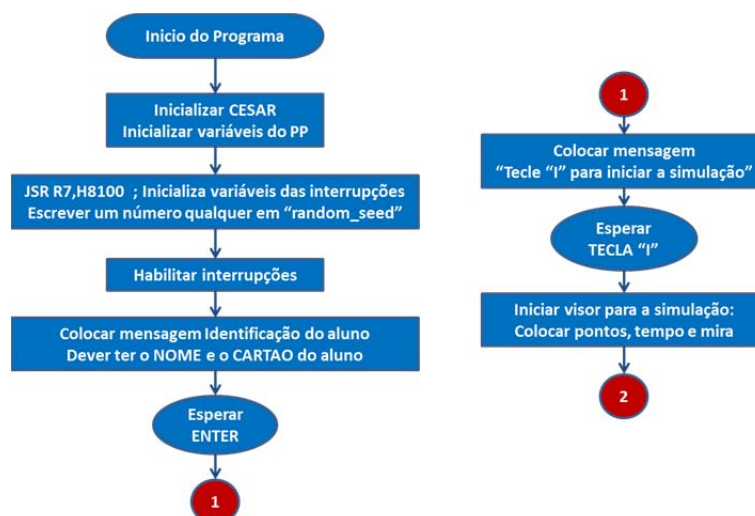
- Incrementar a variável TIMER uma vez por segundo;
- Analisar o vetor ASTEROIDES e realizar a animação dos asteroides, conforme determinado pelo valor de ESTADO;
- Controlar as coordenadas dos asteroides para removê-los ao saírem do campo de asteroides (os 30 caracteres mais a direita do visor);
- Analisar a estrutura MIRA e redesenhá-la, conforme determinado pelo valor de ESTADO.

Notar que a mira e os asteroides são desenhados na mesma área do visor. Então, para garantir a correta operação da animação, todos os asteroides e a mira devem ser apagados, então determina-se suas novas coordenadas, e finalmente desenha-se cada uma delas no visor. Caso as coordenadas dos caracteres "[" e "]" da mira devam ser desenhados nas mesmas coordenadas de um asteroide, o asteroide será sobreposto pela mira.

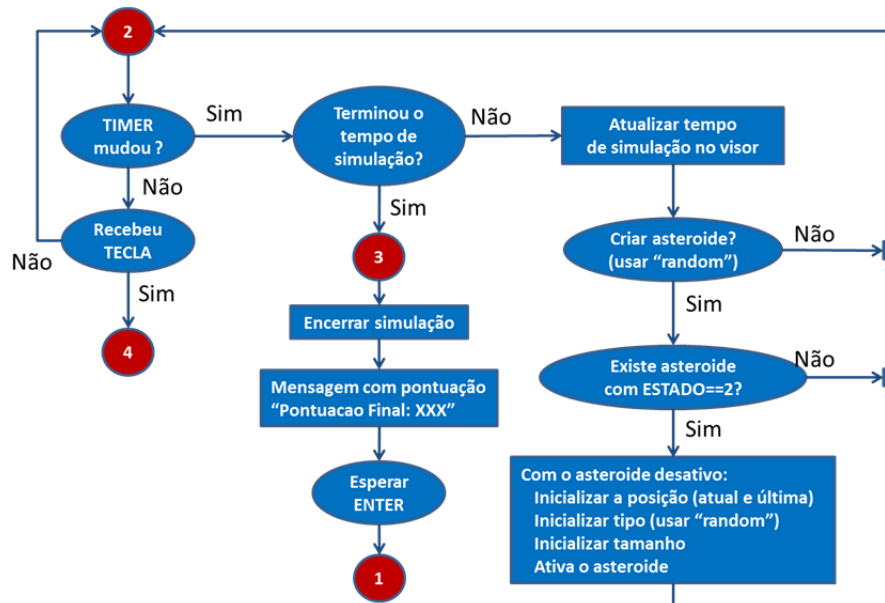
9. Qual sua tarefa na SEGUNDA PARTE DO TRABALHO – implementação do programa principal

Nessa parte do trabalho você deve implementar o programa principal. O algoritmo a ser implementado no programa principal está detalhado nas três figuras abaixo. Notar que nas figuras existem "rótulos" (números dentro de círculos em vermelho), que foram usados para indicar a conexão entre as três figuras.

Na primeira figura são apresentados os procedimentos de inicialização, a chamada da rotina de inicialização das variáveis dos tratadores de interrupção (localizada em 8100₁₆), a colocação da mensagem de identificação do aluno, a mensagem de pedido pelo início da simulação e o início da simulação propriamente dito.

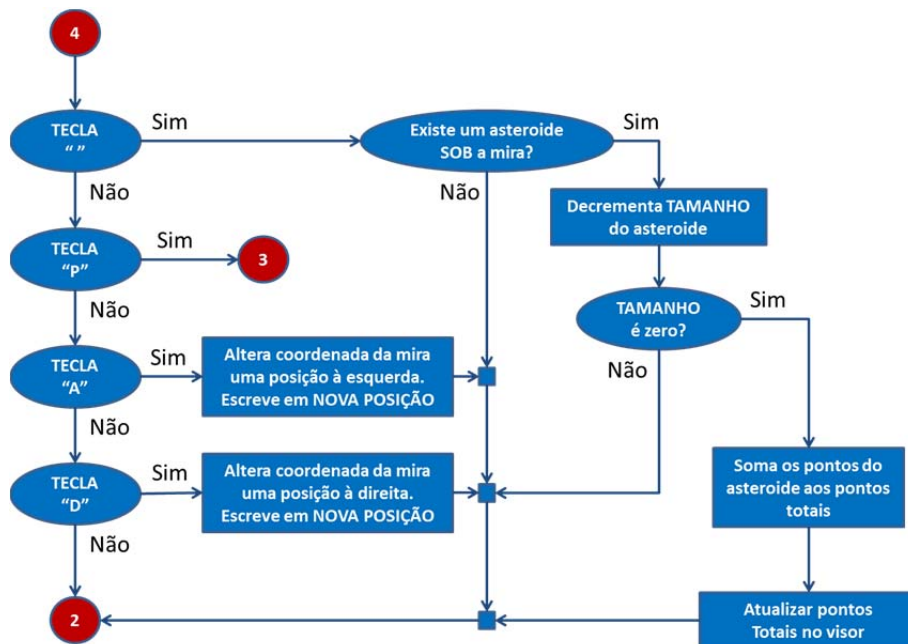


A segunda figura apresenta o *laço* de programa que espera a ocorrência de algum evento: a passagem do tempo ou o acionamento de alguma tecla. Nessa figura também estão representadas as ações a serem realizadas quando o contador de tempo mudar, quando o tempo de simulação encerrar (após 60 segundos de simulação) e como criar um novo asteroide.



A terceira figura apresenta o tratamento das teclas digitadas pelo operador do canhão. A tecla “ ” (SPACE) indica o disparo do canhão. Nesse caso, deve-se verificar se um asteroide foi atingido. Se foi, deve-se decrementar o tamanho do mesmo. Se o tamanho chegou a zero, o asteroide foi destruído. Nesse caso, deve-se somar os pontos totais e desativar o asteroide.

A tecla “P” indica que o operador desistiu da simulação. As teclas “A” e “D” são usadas para movimentar a mira do canhão para a esquerda e direita, respectivamente.



10. Correção e Entregáveis

A correção da primeira parte (tratadores de interrupção) será feita através de um programa principal especialmente construído para efetuar a correção automática da implementação do aluno. De forma semelhante, a correção da segunda parte (programa principal) será feita através de um tratador de interrupção especialmente construído para realizar a correção automática da implementação do aluno. Portanto, a leitura e escrita das variáveis de comunicação devem ser realizadas **exatamente** conforme especificado.

A nota final do trabalho será proporcional aos casos de teste para os quais as implementações forem bem sucedidas. A primeira parte do trabalho (tratadores de interrupção) terá **peso 4 sobre 10** enquanto que a segunda parte (programa principal) terá **peso 6 sobre 10**.

Cada parte do trabalho deverá ser entregue na entrada adequada do Moodle da turma. Deve ser entregue um arquivo fonte (arquivo .CED) com a solução correspondente, escrito em *assembly* do CESAR16i. Além disso, esse programa fonte deverá conter comentários descritivos da implementação. Sugere-se usar comandos da linguagem “C”.

Cada parte do trabalho deverá ser entregue até a data prevista. Não serão aceitos trabalhos entregues além do prazo estabelecido. Trabalhos não entregues até a data prevista receberão nota zero.

11. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.