

SME0230 - Introdução à Programação de Computadores

Primeiro semestre de 2022

Professoras: Marina Andretta (andretta@icmc.usp.br) e
Franklina Toledo (fran@icmc.usp.br)

Monitores: Ana Alice Zulian (anaalicezulian@usp.br)
Bruno da Freiria Mischiato Borges (brunofborges@usp.br)

Trabalho: jogo *Chomp*

1 Grupos

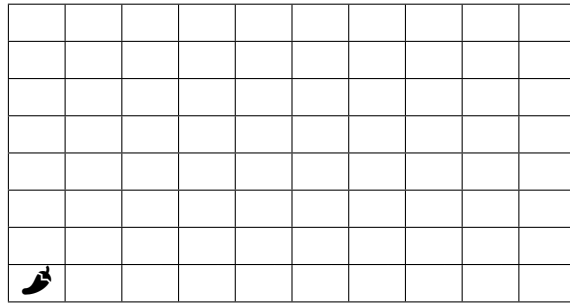
- O trabalho será dividido em duas etapas, que poderão ser feitas em grupos de até **3** pessoas (recomendamos fortemente que trabalhem em trios).
- As duas etapas do trabalho devem ser feitas pelo mesmo grupo.
- Caso alguém queira mudar de grupo, deverá entrar em contato previamente com uma das professoras.

2 Submissão

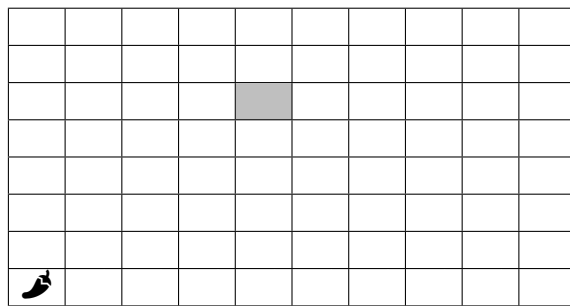
- As duas etapas do trabalho deverão ser submetidas no e-disciplinas.
- Apenas o arquivo fonte deve ser entregue, em um arquivo `.c`.
- No início do arquivo (`.c`) deverá haver um comentário com o nome e o número USP de todos os integrantes do grupo. Caso algum componente não esteja identificado, ficará com nota 0 (zero).

3 Como funciona o jogo

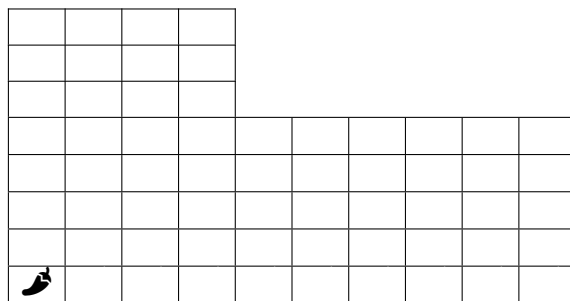
O *Chomp* possui um tabuleiro (representando uma barra de chocolate) com $n \times m$ quadrados (no nosso caso, o tabuleiro será 8×10), com o quadrado mais abaixo e à esquerda marcado (este é o pedaço do chocolate que tem pimenta!):



Dois jogadores (**Jogador 1** e **Jogador 2**) alternam suas jogadas. Em cada jogada, um jogador escolhe um quadrado e este quadrado é removido do tabuleiro, juntamente com os que estão acima e à direita dele. Por exemplo, se o **Jogador 1** escolher o quadrado na terceira linha e quinta coluna,



o tabuleiro fica assim:



Depois de atualizado o tabuleiro, a vez passa para o outro jogador. O jogador que pegar o quadrado marcado (que tem a pimenta) perde o jogo.

4 Objetivo

O objetivo deste trabalho é desenvolver um programa em linguagem C que implementa duas versões do jogo *Chomp*: uma em que dois jogadores usuários (humanos) jogam um contra o outro e uma em que um jogador humano joga contra o computador.

No início do programa, o usuário deverá escolher qual das versões será jogada. Deve ficar claro como os jogadores devem fazer para especificar suas jogadas. Na versão em que os dois jogadores (**Jogador 1** e **Jogador 2**) são usuários, o programa deverá mostrar na tela de quem é a vez de jogar.

O **Jogador 1** sempre começa o jogo. No caso usuário *versus* computador, **considera-se o computador como Jogador 1**. Após a primeira exibição do tabuleiro, o programa irá aguardar a inserção da instrução de movimento no terminal. A instrução pode ser válida ou não, e a verificação deverá ser feita após a inserção; caso não seja válida, o programa imprime uma mensagem de erro e aguarda uma nova inserção.

Em cada inserção de jogada, a representação do tabuleiro será naturalmente modificada conforme a jogada feita e isso deve ser mostrado ao jogador. Depois da jogada, o programa deverá verificar se o jogador venceu o jogo. Neste caso, uma mensagem é impressa na tela e o jogo termina. Caso contrário, passa-se a vez para o outro jogador.

5 Práticas obrigatórias e proibidas

É necessário realizar a indentação do código. É uma prática que torna o código legível e permite a manutenção com mais facilidade. Sugere-se que sejam utilizadas as duas formas comuns de indentação, como descrito a seguir por meio de exemplos arbitrários na Figura 1:

Figura 1: Exemplo de indentações para o código em linguagem C.

<pre>#include <stdio.h> void imprimir_valor(int n) { printf("%d\n", n); return; } int main() { int i = 0; while (i < 10) { imprimir_valor(i); i++; } return 0; }</pre>	<pre>#include <stdio.h> void imprimir_valor(int n) { printf("%d\n", n); return; } int main() { int i = 0; while (i < 10) { imprimir_valor(i); i++; } return 0; }</pre>
---	---

É necessário também comentar o código. Comentários que apenas descrevem em português, literalmente, o que o programa faz, geralmente não são muito úteis. São desejáveis comentários que explicam o que trechos do programa fazem, facilitando o entendimento do código por outras pessoas (além dos próprios autores).

Não é necessário dividir o código da primeira parte do trabalho em funções (se você não sabe

o que são funções, não se preocupe: isso será ensinado mais adiante na disciplina). Porém, caso o grupo opte por utilizar funções, todas deverão possuir uma instrução **return**, mesmo as funções declaradas como **void**.

Com o objetivo de promover as boas práticas de programação, **é expressamente proibida a utilização dos seguintes recursos de programação**: comando **goto** e variáveis globais. Se você não sabe o que são estes dois recursos, pode perguntar às professoras como eles funcionam. Mas só por curiosidade, já que eles **não** podem ser usados.

6 Datas de entrega

As duas etapas devem ser entregues no e-disciplinas, nas seguintes datas:

- Versão usuário x usuário: 29 de maio de 2022.
- Versão computador x usuário: 4 de julho de 2022.