

Trabalho II¹

Batalhas navais estão prestes a eclodir. As formas em um arquivo .geo representam equipamentos navais espalhadas no oceano. Por exemplo, o texto <> e # representam, respectivamente, Rigid-Hulled Inflatable Boats (RHIB)² e minas flutuantes; círculos representam hovercrafts de assalto;³ linhas, submarinos; retângulos, os mais diversos tipos de embarcações. Os ataques a estes equipamentos são planejados em arquivos .qry.

Cada equipamento pode ser agredido de três formas. Na primeira forma, o barco é movimentado, atinge uma mina flutuante e é imediatamente destruído. A movimentação se dá em 2 fases: seleção das embarcações passíveis de movimentação e a efetiva movimentação de alguns dos barcos previamente selecionados.

Na segunda, o equipamento é atingido por um torpedo. Alguns equipamentos são imediatamente destruídos, outros precisam de 2 ou mais tiros para sua destruição. Algumas das naus selecionadas são designadas como naus-capitãs. Estas, ao serem atingidas por um torpedo tem reação específica.

Na terceira forma, uma irradiação afeta os equipamentos navais que estão inteiramente contidos numa área circular. Os equipamentos possuem um certo nível de proteção contra tal agressão. O nível de proteção degrada-se a cada agressão. O equipamento torna-se inativo quando o nível de proteção chega a zero. O grau desta redução depende da proporção da área do equipamento afetado e da área da região de irradiação: quanto mais “concentrada” a irradiação, maior a degradação:

$$Redução = \frac{Agressividade * Area(equipamento)}{Area(região)}$$

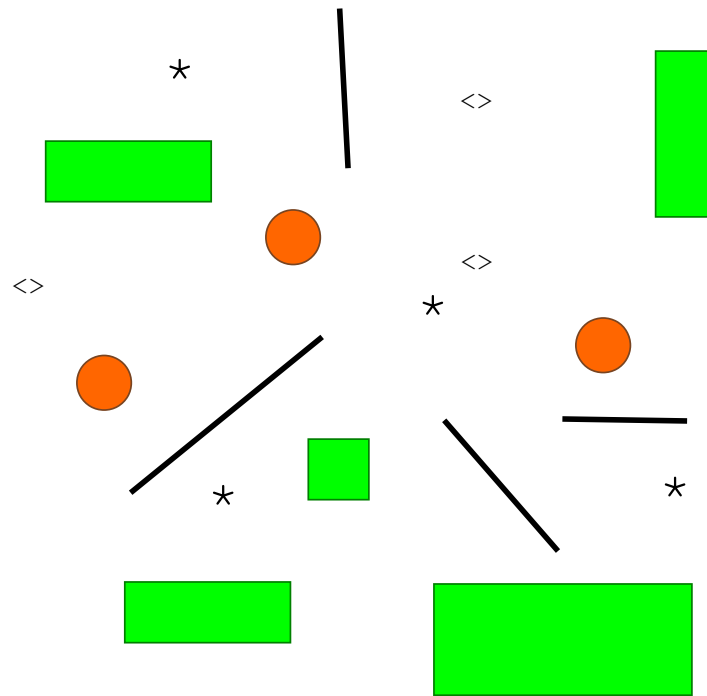
Para fins de cálculo:

- área de um texto: 0.1
- área de uma linha: 0.1*Comprimento(linha)
- Agressividade: uma constante informada por meio de um comando no arquivo .qry.

¹ Veja também a descrição geral para todos os trabalhos.

² <https://www.navy.gov.au/fleet/ships-boats-craft/other-vessels/rhib>

³ <https://images.app.goo.gl/MgAx6GZiv819qxyB9>



A tabela abaixo mostra os níveis iniciais de proteção de cada tipo de equipamento naval; a pontuação pela inativação de cada tipo de equipamento e a pontuação por sua destruição. Note que a pontuação pela destruição de círculos e retângulos depende de suas áreas (ou seja, quanto maior, menor a pontuação).

Forma	Nível Proteção	Pontos Inativação	Pontos Destruição	Tiros para Destruição
texto	5	30	500	1
linha	50	50	150	1
círculo	60	75	$75/(\text{area_forma}/5)$	2
retângulo	60	90	$90/(\text{area_forma}/5)$	3

A Entrada

A entrada do algoritmo será basicamente um conjunto de formas geométricas básicas (retângulos, círculos, etc) dispostos numa região do plano cartesiano .

Considere a Ilustração 1. Cada forma geométrica é definida por uma coordenada âncora (marcada, na figura, por um pequeno ponto vermelho) e por suas dimensões. A coordenada âncora do círculo é o seu centro e sua dimensão é definida por seu raio (r, na figura). A coordenada âncora do retângulo é seu canto inferior esquerdo⁴ e suas dimensões são sua largura (w) e sua altura (h). A coordenada âncora de um texto, normalmente, é o início do texto, porém, pode ser definida como o meio ou o fim do texto. Por fim, uma linha é determinada por duas âncoras em suas extremidades. As coordenadas que posicionam as formas geométricas são valores reais.

Cada forma geométrica é identificada por um número inteiro.

⁴ Note que o plano cartesiano está desenhado "de ponta-cabeça" em relação à representação usual.

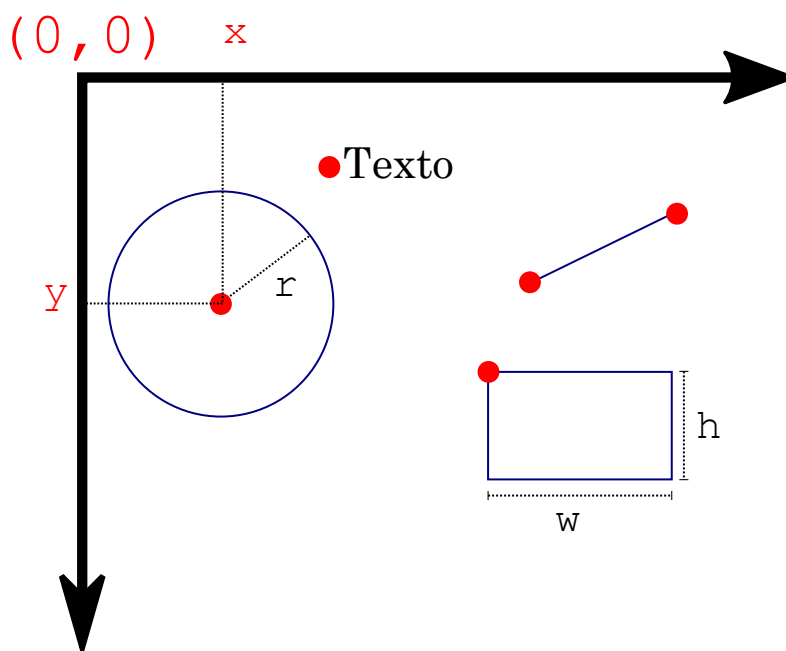


Ilustração 1: Formas no plano

As tabelas abaixo mostram os formatos dos arquivos de entrada (.geo e .qry). Cada comando tem um certo número de parâmetros. Os parâmetros mais comuns são:

- i, j, k : número inteiro, maior ou igual a 1. Identificador de uma forma geométrica.
- r : número real. Raio do círculo.
- x, y : números reais. Coordenada (x,y) .
- cor : string. Cor válida dentro do padrão SVG.⁵

comando	parâmetros	descrição
c	$i \ x \ y \ r \ corb \ corp$	desenhar círculo. $corb$ é a cor da borda e $corp$ é a cor do preenchimento
r	$i \ x \ y \ w \ h \ corb \ corp$	desenhar retângulo: w é a largura do retângulo e h , a altura. $corb$ é a cor da borda e $corp$ é a cor do preenchimento
l	$i \ x1 \ y1 \ x2 \ y2 \ cor$	Desenhar linha com extremidades nos pontos $(x1,y1)$ e $(x2,y2)$, com a cor especificada.
t	$i \ x \ y \ corb \ corp \ a \ txto$	desenha o texto $txto$ nas coordenadas (x, y) e com as cores indicadas. $corb$ é a cor da borda e $corp$ é a cor do preenchimento. O parâmetro a determina a posição da âncora do texto: i , no início; m , no meio, f , no fim. O texto $txto$ é o último parâmetro do comando. Pode incluir espaços em branco e se estende até o final da linha.
comandos .geo		

⁵ <http://www.december.com/html/spec/colorsvg.html>.
<https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>

O arquivo .qry pode conter as seguintes consultas. Note que algumas delas mostrarão, de alguma forma, o estado interno da árvore usada para armazenar as naus e as minas, entre elas:

1. Sempre que uma **nau for destruída** (ou seja, removida do “banco de dados”), desenhar no svg, o círculo envolvente armazenado no nó que foi removido e de seus filhos esquerdo e direito. O círculo deve ter borda vermelha tracejada, preenchimento vermelho e opacidade de 40%.
2. Sempre que uma **nau for inserida**, reportar no txt o círculo envolvente do nó onde foi armazenada a nau inserida e de seus ancestrais (depois da inserção)
3. Ao final do processamento do arquivo de consultas, deve ser produzido outro arquivo svg: fazer um percurso em largura na árvore final, desenhando o círculo envolvente de cada nó. O círculo deve ter cor magenta e opacidade de 10%

comando	parâmetros	descrição
na	v	<i>Atribui o nível de agressividade v (real) às regiões de ataque</i>
tp	x y	<p><i>Torpedo no ponto (x,y). Determinar as formas atingidas pelo torpedo, ou seja, todas as formas para as quais o ponto (x,y) é interno. Caso uma nau-capitã seja atingida ela também se comporta como uma bomba de irradiação (be) cujos parâmetros são iguais ao círculo envolvente do nó em que a nau estava armazenada.</i></p> <p><i>TXT: reportar todos os dados das formas atingidas (incluindo níveis de energia e o número de torpedos que já a atingiram) ou AGUA se o tiro errou o alvo. Caso a forma tenha sido destruída, removê-la e reportar destruição</i></p> <p><i>Se atingiu nau-capitã, reportar também o círculo envolvente.</i></p> <p><i>SVG: marcar o ponto (x,y) com um asterisco vermelho (se acertou alguma forma) ou cinza (caso contrário). Caso tenha atingido mais de uma forma, colocar o número de formas atingidas ao lado do asterisco.</i></p> <p><i>Se atingiu nau-capitã, o asterico deve ser maior.</i></p> <p><i>(Obviamente, a forma destruída não deve aparecer no svg final)</i></p> <p>Atender ao item 1.</p>

comando	parâmetros	descrição
tr	x y r dx dy id corb	<p>Torpedo replicante na área circular de centro em (x,y) e raio r. Cria um clone para cada uma das formas atingidas e os posiciona transladados de (dx,dy) de seu original. O nível de proteção atual e o número de torpedos recebidos é o mesmo do original. O identificador dos clones devem começar no valor id.</p> <p>A cor da borda da forma original deve ser mudada para corb.</p> <p>TXT: reportar todos os dados das formas atingidas (clonadas); reportar todos os dados dos clones</p> <p>Atender ao item 2.</p> <p>SVG: Colocar um @ no ponto (x,y). (Obviamente, como os clones são inseridos no “banco de dados”, provavelmente aparecerão no svg final)</p> <p>Obs. O torpedo replicante é um bônus que pode aumentar a possibilidade de pontuação.</p>
be	x y r	<p>Bomba de irradiação atingiu região circular especificada nos parâmetros.</p> <p>Formas inteiramente contidas na região devem ter seu nível de proteção diminuído proporcionalmente (como explicado acima). Formas cujo nível de proteção atingir 0 devem ser removidas.</p> <p>TXT: reportar todos os dados das formas atingidas, acrescentar a palavra REMOVIDA para aquelas removidas.</p> <p>SVG: Desenhar a região com contorno vermelho pontilhado e sem preenchimento. Colocar um ponto vermelho na posição das âncoras das formas atingidas.</p> <p>Atender ao item 1.</p>
se	id	<p>Insere a nau id no final da lista de naus selecionadas.</p> <p>TXT: reportar os dados da nau selecionada</p>
sec	id	<p>Similar ao comando se. Também designa a nau como capitã.</p>
org		<p>Ordena as naus selecionadas por sua área</p> <p>TXT: Reporta o identificador e a área das naus (na ordem produzida)</p>

comando	parâmetros	descrição
mvh	k dx	<p>Move horizontalmente as k primeiras (se $k > 0$; k últimas, se $K < 0$) embarcações selecionadas em dx unidades (para direita, se $dx > 0$ ou para esquerda, se $dx < 0$). Caso alguma embarcação atinja uma mina durante o percurso, a mina é detonada e a embarcação é destruída (também deve ser retirada da lista de selecionados. Quando uma mina é detonada, ela também é destruída.</p> <p>TXT: Reportar os dados finais das embarcações movidas. Caso uma embarcação seja atingida por uma mina, reportar a destruição da embarcação e da mina (incluindo id e posição).</p> <p>SVG: colocar no SVG final um símbolo & no local da mina detonada.</p> <p>Atender ao item 1.</p>
mvv	k dy	<p>Semelhante ao mvh. Move verticalmente as embarcações.</p>
Comandos .qry		

A Saída

Ao final do processamento das agressões (ou seja, execução do arquivo .qry), como de costume, é produzido o svg representando a situação final do “banco de dados”, com as anotações descritas nas especificações e apresentando a pontuação. A pontuação também deve ser apresentada no respectivo arquivo .txt.

A pontuação consiste de 3 indicadores: a pontuação total; a proporção em pontos obtidos e a pontuação máxima; proporção entre pontos obtidos e o número de agressões contidas no arquivo .qry.

IMPLEMENTAÇÃO

O “banco de dados” **deve** ser armazenado numa árvore (**deve** ser conforme .h disponibilizado no AVA).

Usar o modelo de makefile disponibilizado no AVA da disciplina.

É **terminantemente proibido** declarar structs nos arquivos de cabeçalho (.h).

O programa **deve** estar bem modularizado (arquivos .h e .c). Cada estrutura de dados **deve** estar em um módulo separado. O arquivo .h **deve** estar muito bem documentado (lembre-se que é um “contrato”).

Implementar algoritmo de ordenação **quicksort** para ordenar a lista de selecionados. **Embaralhar** a lista antes de fazer a ordenação.

AVALIAÇÃO

Espera-se uma atitude pró-ativa para a aquisição dos conhecimentos (i.e., estudo) para resolver o problema proposto.

A avaliação consistirá da execução dos testes e da inspeção de código.

O Que Entregar

Submeter no Classroom o arquivo .zip com os fontes , conforme descrito anteriormente.

RESUMO DOS PARÂMETROS DO PROGRAMA TED

Parâmetro / argumento	Opcional	Descrição
-e <i>path</i>	S	Diretório-base de entrada (BED)
-f <i>arq.geo</i>	N	Arquivo com a descrição da cidade. Este arquivo deve estar sob o diretório BED .
-o <i>path</i>	N	Diretório-base de saída (BSD)
-q <i>arqcons.qry</i>	S	Arquivo com consultas. Este arquivo deve estar sob o diretório BED .

RESUMO DOS ARQUIVOS PRODUZIDOS

-f	-q	comando com sufixo	arquivos
<i>arq.geo</i>			arq.svg
<i>arq.geo</i>	<i>arqcons.qry</i>		arq.svg arq-arqcons.svg arq-arqcons.txt
<i>arq.geo</i>	<i>arqcons.qry</i>	<i>sufx</i>	arq.svg arq-arqcons.svg arq-arqcons.txt arq-arqcons-sufx.[svg txt] ⁶

ATENÇÃO:

- * os fontes devem ser compilados com a opção `-fstack-protector-all`.
- * adotamos o padrão C99. Usar a opção `-std=c99`.

⁶ Podem ser produzidos os respectivos arquivos .svg e/ou .txt, dependendo da especificação do comando.