

# Trabalho I<sup>1</sup>

O mundo de Pandora vive momentos de tensão entre os povos Ya'nki e Chi'ni. O povo Chi'ni enviou secretamente balões para espionar o território dos Ya'nki. Estes descobriram objetos voadores e enviam caças para derrubá-los via bombas de detonação por afastamento. Entretanto, os balões tiram fotos e enviam-nas para sua base de observação.

As fotos podem retratar desde instalações militares ou logísticas de grande importância até regiões vazias ou pouco relevantes. Por isso, a base avalia cada foto e atribui a ela um número que denota seu grau de importância: quanto maior o número, maior a importância.

O sensor fotográfico do balão possui uma certa capacidade de observação: o raio de abrangência e a faixa de foco. Observe Ilustração 1. O texto `v_O_v` representa um balão. A sua câmera fotográfica está posicionada na metade da parte inferior (você pode supor que está sob a letra O). O raio de abrangência está representado pela segmento rotulado com a letra  $r$ . A profundidade (representado pela letra  $p$ ) é a distância abaixo do balão em que o foco da câmera começa a ficar nítido. A profundidade  $h$  representa a faixa em que o foco continua nítido. Desta forma, a câmera capturará apenas os elementos que estiverem dentro da região pontilhada em vermelho.<sup>2</sup> No exemplo abaixo, os elementos capturados estão coloridos e os não capturados estão em cinza.

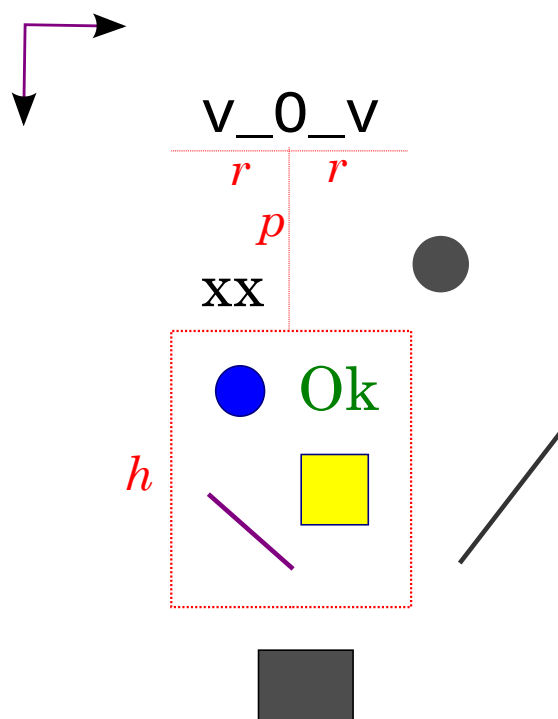
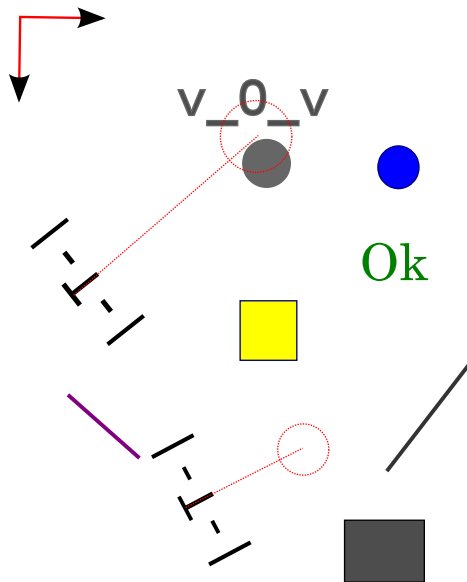


Ilustração 1: Balão fotografando

<sup>1</sup> Veja também a descrição geral para todos os trabalhos.

<sup>2</sup> A foto inclui todos elementos pelo menos parcialmente dentro da região, mas não precisa cortar a parte

Um caça tenta derrubar um balão disparando uma bomba de detonação por afastamento. Existem bombas de diferentes capacidades: A, B e C. Antes do lançamento o piloto define a capacidade da bomba e a distância que a bomba deve percorrer antes de explodir.. Quando a bomba atinge tal distância, ela detona-se e atinge qualquer elemento que estiver no raio de **10 unidades** (para bombas do tipo A), **15 unidades** (tipo B) ou **30 unidades** (tipo C) do ponto da explosão, destruindo-o.



*Ilustração 2: Disparo*

A Ilustração 2 mostra dois caças (representados por **-T-**) disparando bombas de capacidades diferentes e a distâncias diferentes. Note que uma das bombas atinge um balão e um círculo. A outra não acerta nenhum alvo.

A fim de direcionar o tiro, o caça pode ser rotacionado. Note a Ilustração 3. O caça da esquerda não está rotacionado (ou seja, giro de  $0^0$ ), enquanto que o caça da direita está rotacionado de um certo ângulo  $\alpha$ .



*Ilustração 3: Rotação de um caça*

## A Entrada

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

Considere a Ilustração 4. 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 esquerdo<sup>3</sup> e suas dimensões são sua largura ( $w$ ) e sua altura ( $h$ ). A

<sup>3</sup> Note que o plano cartesiano está desenhado "de ponta-cabeça" em relação à representação usual.

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.

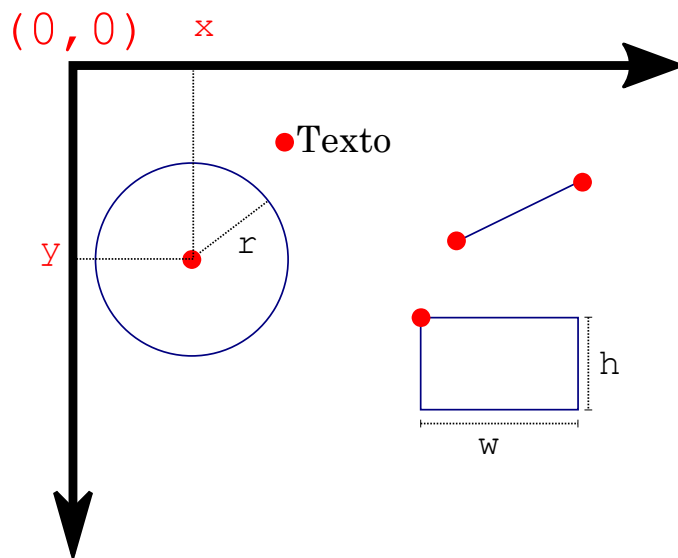


Ilustração 4: Formas no plano

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.<sup>4</sup>

comando	parâmetros	descrição
<b>c</b>	$i \ x \ y \ r \ corb \ corp$	<i>desenhar círculo. corb é a cor da borda e corp é a cor do preenchimento</i>
<b>r</b>	$i \ x \ y \ w \ h \ corb \ corp$	<i>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</i>
<b>l</b>	$i \ x1 \ y1 \ x2 \ y2 \ cor$	<i>Desenhar linha com extremidades nos pontos <math>(x1,y1)</math> e <math>(x2,y2)</math>, com a cor especificada.</i>
<b>ts</b>	$fFamily \ fWeight \ fSize$	<i>Muda o estilo dos textos (comando t) subsequentes. font family: sans (sans-serif), serif, cursive; font weight ( n: normal, b: bold, b+: bolder, l:   lighter)</i>

<sup>4</sup> <http://www.december.com/html/spec/colorsvg.html>.  
<https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>

<b>t</b>	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		

O arquivo .qry pode conter as seguintes consultas:

comando	parâmetros	descrição
<b>mv</b>	i dx dy	Move figura/texto de identificador i de dx (no eixo x) e dy (no eixo y). dx e dy podem ser negativos TXT: reportar os atributos da figura i, a posição original e a posição final. SVG: naturalmente figura aparecerá na posição final
<b>g</b>	i grs	Rotaciona figura texto de identificador i em grs graus. grs pode ser negativo. <sup>5</sup> TXT: reportar atributos da figura i; reportar a inclinação (graus) antes e depois da operação.
<b>ff</b>	i r p h	Define os parâmetros de foco da foto (raio, profundidade, altura) do balão i.
<b>tf</b>	i l	Balão i tira foto e coloca no final da lista l (0..9). TXT: reportar atributos a figura i, atributos da câmera (comando ff), identificadores dos elementos fotografados com a posição relativa do elemento à borda da foto (dx, dy) SVG: desenhar o retângulo da região capturada pela foto, bordas pontilhadas em vermelho.
<b>df</b>	i l sfx	Envia fotos da lista l do balão i para a base. Base categoriza cada foto. TXT: para cada foto enviada, reportar identificador e atributos da foto e pontuação. <sup>6</sup> SVG: cria novo svg com mesmo nome do svg + sufixo sfx (ex. a1-q1-sfx.svg) mostrando as fotos baixadas em sequência

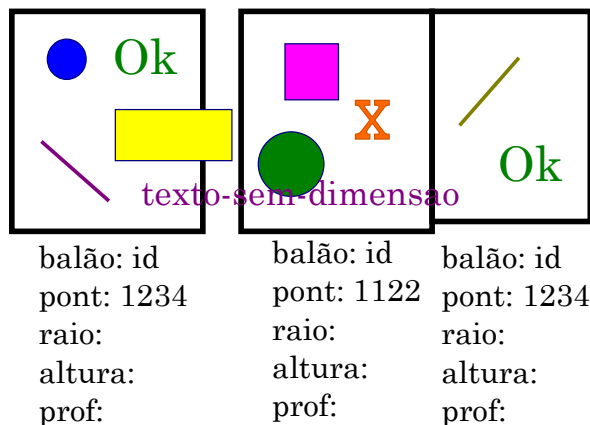
<sup>5</sup> <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/rotate>  
[https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Basic\\_Transformations](https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Basic_Transformations)

<sup>6</sup> A pontuação é função das formas fotografadas e seus atributos.

comando	parâmetros	descrição
<b>d</b>	i capac dist j dx	<p><i>Caça de identificador i dispara bomba de capacidade capac (A, B ou C) a uma distância dist.</i></p> <p><i>Os elementos atingidos são destruídos, i.e., removidos do “banco de dados”.</i></p> <p><i>Caso o disparo atinja um balão, clonar os elementos contidos nas fotos que não foram enviadas para base. Os identificadores dos clones começam em j e estão transladados dx no eixo x. Os clones (quando aplicável) tem suas cores de borda e preenchimento intercambiadas em relação aos originais. Clonagens devem ser feitas antes da destruição.<sup>7</sup></i></p> <p><i>TXT: reportar id e atributos dos elementos atingidos; id e atributos dos clones; reportar os atributos dos clones.</i></p> <p><i>SVG: colocar um X vermelho (FF0000) na âncora das figuras atingidas.</i></p>
<b>b?</b>		<i>Reporta os dados de todos os balões existentes, incluindo seus atributos e o número de fotos por fila e a rotação corrente do balão.</i>
<b>c?</b>		<i>Reporta os dados de todos as caças existentes: seus atributos, a rotação corrente, quantos disparos já efetuou e os identificadores dos elementos que acertou até o momento.</i>
<b>Comandos .qry</b>		

Note que as fotos estão na sequência em que foram tiradas.

## A Pontuação da Fotos



<sup>7</sup> Caso elemento a ser clonado também tenha sido atingido pelo mesmo disparo.

Quando a base recebe uma sequência de fotos, ela pontua cada foto recebida. Esta pontuação depende do conteúdo capturado na foto:

TIPO	VALOR ESPECÍFICO	PONTUAÇÃO
<b>Texto</b>		número de caracteres
(caça)	- T -	100
<b>Linha</b>		comprimento
(energia alta potência)	cor: amarela (FFFF00)	X 3
(energia baixa potência)	cor: DDFF55	X 2
(dados)	cor: 0000FF	X 4
<b>Círculos</b>		area/2
(planta nuclear)	borda: FFFFFFFF preenchimento: FFFF00	X 8
(prédio público)	borda: D45500 preenchimento: FF7F2A	X 2
(hospital)	borda: AA0000 preenchimento: DE8787	X 4
(ignorar)	borda: FFFFFFFF preenchimento: B3B3B3	0
<b>Retângulos</b>		area/4
	borda: 800080 borda: AA0088	+ 10 + 15
	preenchimento: 008033 preenchimento: FFCC00	+ 20 + 30

## A Saída

Ao final do processamento do arquivo .qry é produzido o svg representando a situação final do “banco de dados”, com as anotações descritas nas especificações.

Caso o arquivo .qry contenha um ou mais comandos **df**, outros arquivos svg devem ser produzidos

## IMPLEMENTAÇÃO

O “banco de dados” **deve** ser armazenado numa lista (**deve** ser conforme .h disponibilizado no AVA). A implementação da lista **deve** ser dinâmica e duplamente encadeada. O iterador e as funções map, filter, fold **devem** ser **implementadas** e **utilizadas**.

Cada balão deve armazenar 10 filas de fotos. A implementação da fila deve ser circular, estática. Cada fila terá no máximo **15 elementos**.

**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”).

## 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.

A nota é proporcional ao número de testes corretos, aplicados os descontos abaixo:

Critério	Desconto
Escrever struct em arquivo .h	2.5
Modularização Pobre: .h mal projetado, mal documentado	até 1.5
Não implementado conforme especificado	tipicamente até 3.0, mas em casos graves, o desconto pode ser total
Procedimentos extensos e/ou complicados	até 1.0
Não usar o makefile provido	Em geral, nenhum desconto. Mas, se ocorrer problema de compilação ou execução que poderia ter sido evitado pelo uso do modelo do makefile provido, a consequência pode ser grave.
Erro de compilação	Nenhum teste será executado. Portanto, nota Zero, especialmente se for causado por negligência do aluno.

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

## 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] <sup>8</sup>

### ATENÇÃO:

\* os fontes devem ser compilados com a opção `-fstack-protector-all`.

\* adotamos o padrão C99. Usar a opção `-std=c99`.

## DÚVIDAS:

- para selecionar forma em uma foto, não considerar rotação
- texto: dimensão zero

---

<sup>8</sup> Podem ser produzidos os respectivos arquivos .svg e/ou .txt, dependendo da especificação do comando.