

LFA - GEOMETRICS

Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e
Informática

André Clérigo, Bruno Acioli, João Silva, Pedro
Coutinho, Pedro Rocha



Conteúdo

1	Documentação	1
1.1	Geometrics Main	1
1.1.1	Syntax	1
1.1.2	point	2
1.1.3	Primitivas geométricas	2
1.1.4	Figuras geométricas	3
1.1.5	Propriedades das figuras	3
1.1.6	Adição/Remoção	4
1.1.7	Animations	4
1.1.8	Informações adicionais	5
1.2	Geometrics Aux	5
1.2.1	Syntax	6
2	Contactos dos autores	7

Capítulo 1

Documentação

Tal como foi pedido a linguagem foi separada em duas gramáticas, Geometrics Main e Geometrics Aux (Geometrics Database). O Geometrics Aux apenas permite a definição de figuras, não podendo definir animações; expressões nem statements de add e remove, estas figuras são exclusivamente criadas para serem mais tarde importadas por um ficheiro de Geometrics Main, sendo que um ficheiro de Geometrics Aux só por si não define um programa em Geometrics.

Nota: Para evitar conflitos entre nomes de objetos, é recomendado que os nomes de figuras ou primitivas geométricas sejam definidas como 'fileName_variableName'.

1.1 Geometrics Main

1.1.1 Syntax

Convenciona-se que todos os ficheiros em Geometrics tenham a extensão '.geom' sendo que esta só é sintaticamente obrigatória quando se faz o import de um ficheiro Geometrics Aux dentro de um ficheiro Geometrics Main.

Convencionou-se também que todos os ficheiros Geometrics Main devem começar com a linha 'geometrics:main', da mesma forma ficheiros de Geometrics Aux devem começar com a linha 'geometrics:database'.

Os imports de bancos de figuras, devem ser feitos com o comando 'import "filename.geom"'.

Comentários

Para o programador escrever um comentário, este deve inicia-lo com '/*' e termina-lo com '*/', sendo que estes podem incorporar multiplas linhas.

1.1.2 point

O 'point' (ponto) é o elemento mais básico da linguagem, todas as primitivas utilizam o ponto. Para criar um ponto o programador deve declara-lo da seguinte forma:

```
point 'nome do ponto' ('expr', 'expr')
e.g.
point p1 (5, 1*2)
point p2 (1, 2)
point p3 (-2, 1)
point p4 (x/2, x+x), em que x é o ID de uma variável que representa uma
expr.
```

Também é possível não criar nenhum ponto previamente à criação da primitiva e utilizar apenas um par de coordenadas.

Nota: Daqui para a frente, sempre que há referências a 'ponto' é referido às coordenadas literais de um 'point' constituídas por expressões (e.g. (5*2, 1)) ou é referido também ao ID da variável que representa o ponto.

1.1.3 Primitivas geométricas

A linguagem desenvolvida possui cinco primitivas geométricas: 'line'; 'circle'; 'rectangle'; 'square' e 'ellipse'. Utilizando estas primitivas é possível criar figuras.

line

A 'line' (linha) é a primitiva geométrica mais básica, deve ser declarada da seguinte forma:

```
line 'nome da linha' 'ponto de origem (point)' 'ponto de destino (point)'
e.g. line l1 p1 (5,1)
```

circle

O 'circle' (círculo) deve ser declarado da seguinte forma:

```
circle 'nome do círculo' 'ponto central (point)' 'raio (expr)'
e.g. circle c1 (-2, 0) 3
```

rectangle

O 'rectangle' (retângulo) deve ser declarado da seguinte forma:

```
rectangle 'nome do retângulo' 'ponto central (point)' 'altura (expr)' 'largura
(expr)'
e.g. rectangle r1 p1 5 10
```

square

O 'square' (quadrado) deve ser declarado da seguinte forma:

square 'nome do quadrado' 'ponto central (point)' 'lado (expr)'
e.g. square s1 p1 5

ellipse

A 'ellipse' (elipse) deve ser declarada da seguinte forma:

ellipse 'nome da elipse' 'ponto central (point)' 'eixo maior (expr)' 'eixo menor (expr)'
e.g. ellipse e1 p1 8 1

1.1.4 Figuras geométricas

A linguagem geometrics permite a criação de figuras utilizando as primitivas geométricas; outras figuras e/ou qualquer outro statement'. As proprias primitivas podem ser encaradas como figuras geométricas primitivas.

Uma figura é criada da seguinte forma:

figure 'nome da figura' { 'primitivas; outras figuras e/ou statements' }.

e.g.

```
figure f1 {  
    add f3 to f5  
    remove f4  
}
```

1.1.5 Propriedades das figuras

Qualquer tipo de primitiva geométrica pode ter propriedades (features) indicadas pelo programador. Essas propriedades são 'color'; 'thickness'; 'filled' e 'hidden'.

color

A propriedade 'color' é utilizada para definir a cor da primitiva indicada, deve ser declarada ao escrever 'color' seguida de um código HTML (e.g. #0259bf); um código RGB (e.g. 0, 144, 255) ou o nome em inglês de uma cor pré definida (e.g. green). Se o programador escolher utilizar uma cor pré definida ele pode também indicar a opacidade desejada com um número de 0 a 100 (e.g grey 80, ou grey 80%).

thickness

A propriedade 'thickness' é utilizada para definir a grossura em pixels da(s) linha(s) da primitiva indicada, deve ser declarada ao escrever 'thickness' seguida do número de pixels desejados.

e.g. line l1 p1 p2 thickness 10

filled

A propriedade 'filled' é utilizada para preencher o interior da primitiva indicada, deve ser declarada ao escrever 'filled' seguida de 'yes' para preencher a figura ou de 'no' para a esvaziar.

e.g.
square1 filled yes
square1 filled no

hidden

A propriedade 'hidden' é utilizada para remover ou reaver a visibilidade da primitiva indicada, deve ser declarada ao escrever 'hidden' seguida de 'yes' para esconder a figura ou de 'no' para a revelar .

e.g.
line1 hidden yes
line1 hidden no

1.1.6 Adição/Remoção

É permitido ao programador invocar statements de adição e remoção. Os comandos para invocar os mesmos são 'add'; 'remove'.

add

O statement 'add' é utilizado para adicionar qualquer ponto, primitiva, figura, animação, propriedades e/ou números a qualquer figura.

e.g.
add circle c1 p2 2 to figure1
add figure2 to figure1
add animation1 to figure2
add -5 to figure3

remove

O statement 'remove' é utilizado para remover um objeto ou uma expressão, para utilizar este comando o programador deve escrever 'remove' 'nome do objeto/expressão'.

e.g.
remove p1
remove f1
remove x

1.1.7 Animations

O programador pode utilizar uma 'animation' para fazer uma figura realizar um movimento específico. Cada animação deve indicar o tempo total (em ms) que a

mesma vai demorar a realizar todos os movimentos, para o fazer o programador deve escrever 'tick is' 'milisegundos'. Além do tempo total, o programador pode especificar quantas vezes deseja repetir a animação, para o fazer o programador deve escrever 'loop' 'número de vezes' 'times'.

Os movimentos disponiveis são 'rotate', 'translate' e 'waitms'.

rotate

O statement 'rotate' é utilizado para rodar uma primitiva ou figura um certo número de graus sobre um ponto ou outra primitiva ou figura, para utilizar este comando o programador deve escrever 'rotate' 'nome da primitiva/figura' que vai rodar' 'número de graus' 'over' 'nome do/a ponto/primitiva/figura'.

e.g.

```
rotate f1 10 over p1
```

```
rotate l2 720 over c1
```

```
rotate r1 700 over f1
```

translate

O statement 'translate' é utilizado para realizar uma translação de uma primitiva ou figura a um certo ponto ou outra figura ou primitiva, para utilizar este comando o programador deve escrever 'translate' 'nome da primitiva/figura' 'to' 'nome do/a ponto/primitiva/figura'.

e.g.

```
translate f1 to p1
```

```
translate l2 to c1
```

```
translate r1 to f1
```

waitms

O statement 'waitms' é utilizado para fazer a figura parar um certo tempo ue deve ser especificado em milisegundos, para utilizar este comando o programador deve escrever 'wait' 'milisegundos'.

e.g.

```
wait 1000
```

1.1.8 Informações adicionais

Apesar do \n ser um statement linguagem, este não é tratado, fazendo com que seja sempre semânticamente válido.

1.2 Geometrics Aux

A Geometrics Aux (Geometrics Database) é utilizada como um banco de dados ou seja apenas podemos definir figuras e estas podem conter outras figuras e/ou primitiva com as respectivas propriedades.

1.2.1 Sintax

'nome da figura' 'is' 'figura/primitivas' e a sintaxe dessas primitivas é igual ao Geometrics Main. Como já foi mencionado, ficheiros de Geometrics Aux devem começar com a linha 'geometrics:database'.

Capítulo 2

Contactos dos autores

98485	André Clérigo	andreclerigo@ua.pt
101077	Bruno Acioli	baal@ua.pt
97512	João Silva	joaocastanheira@ua.pt
93278	Pedro Coutinho	pmacoutinho@ua.pt
98256	Pedro Rocha	pedro.lebrerocha@ua.pt