

Projecto de Laboratórios de Informática I

(1ª fase)

LightBot em *Haskell*

2014/2015 — LEI

1 Introdução

Neste enunciado apresentam-se as tarefas referentes à primeira fase do projecto da unidade curricular de Laboratórios de Informática I. O projecto será desenvolvido por grupos de 2 elementos, e consiste em pequenas aplicações *Haskell* que deverão responder a diferentes tarefas (apresentadas adiante).

O projecto baseia-se no puzzle *LightBot* (<http://lightbot.com>) onde se controla um *robot* num tabuleiro de blocos por intermédio de comandos muito simples com o objectivo de acender todas as lâmpadas disponíveis. Convidam-se os alunos a jogar a versão do jogo *online* (<http://lightbot.com/hocflash.html>) para se familiarizarem com as regras do jogo.

2 Entrada/Saída de Dados

Nas diferentes tarefas computacionais realizadas nesta fase do projecto, o formato para *entrada* e *saída* de dados é sempre textual. Os programas realizados irão ler os dados do `stdin`, e escrever os resultados em `stdout`.

2.1 Formato de Entrada

O formato de entrada é comum a todas as tarefas. Irá representar o tabuleiro onde o *robot* se move; a posição e orientação inicial do *robot*; e o programa para controlar o *robot*.

- Um tabuleiro de dimensão $m \times n$ (com $m, n > 0$) será representado por n linhas contendo cada uma delas uma sequência de m caracteres alfabéticos. Esses caracteres estão associados a diferentes níveis: o

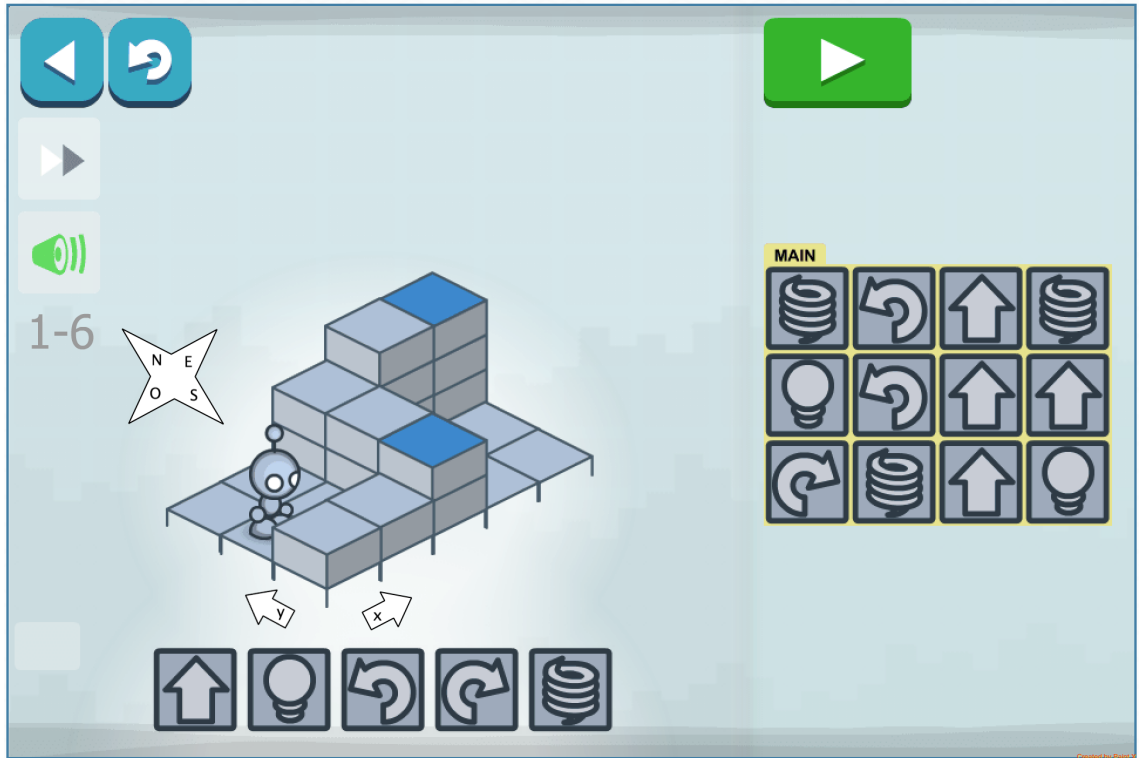


Figura 1: Tabuleiro do jogo *LighBot*

carácter *a* ou *A* ao nível 0; o *b* ou *B* ao 1; e assim sucessivamente. A utilização da letra maiúscula ou minúscula sinaliza se a posição tem ou não uma “lâmpada”. A posição com coordenadas (0,0) corresponde ao primeiro carácter da *n*-ésima linha.

- O estado inicial do *robot* é representada por uma única linha contendo

$$\langle x_pos \rangle \langle y_pos \rangle \langle orient \rangle$$

onde $0 \leq x_pos < m$ e $0 \leq y_pos < n$ (respectivamente as coordenadas *x* e *y*) especificam a posição inicial do *robot*, e $\langle orient \rangle$ é um dos caracteres N, E, S ou O que denota a orientação inicial do *robot*.

- Os comandos são representados por uma sequência não vazia construída com os caracteres A (avancar); S (saltar); E (esquerda); D (direita); e L (luz).

A título de exemplo, os dados de entrada para a configuração apresentada na Figura 1 correspondem às linhas de texto:

```
aacdD
aaca
bbCaa
0 1 S
SEASLEAADSAL
```

2.2 Formato de Saída

O formato de saída será específico para cada uma das tarefas propostas.

3 Tarefas

Nesta fase do projecto serão consideradas três tarefas computacionais. Estas tarefas correspondem aos problemas disponibilizados na plataforma **mooshak** (<http://mooshak.di.uminho.pt>), onde serão submetidas as respectivas soluções.

3.1 Validação do *Input* (tarefa de ambientação)

Pretende-se nesta tarefa realizar um programa que permita validar se o *input* fornecido cumpre os requisitos impostos pela descrição apresentada na Secção 2.1. O programa desenvolvido deve imprimir uma única linha de resultado contendo:

- OK — se o formato do *input* estiver de acordo com a descrição apresentada na Secção 2.1; ou
- <num> — se a linha <num> é a primeira onde os dados divergem do formato prescrito.

O objectivo desta primeira tarefa é fundamentalmente o de ambientar os grupos com o sistema de submissão/avaliação da plataforma **mooshak**. Será por isso dedicada uma das aulas da UC à sua realização, possibilitando dessa forma aos grupos de trabalho interagirem com o sistema e interpretarem o *feedback* que lhes é dado.

3.2 Cálculo da próxima posição

Pretende-se implementar um programa que determine qual a posição do *robot* após a execução do primeiro comando fornecido. Deve atender aos seguintes aspectos:

- assume-se que os dados de entrada estão de acordo com a especificação apresentada na Secção 2.1;
- se o comando não for aplicável (e.g. comando L sobre uma posição do tabuleiro sem *lâmpada*), o programa deve retornar a mensagem “ERRO”.
- no caso do comando ser aplicável, a posição seguinte do *robot* deverá ser apresentada de forma análoga à posição inicial do *robot*, i.e. numa única linha contendo

$\langle x_pos \rangle \langle y_pos \rangle \langle orient \rangle$

3.3 Execução de programas

Nesta última tarefa pretende-se implementar um programa que faça o *robot* executar a sequência de comandos apresentados até acender todas as lâmpadas disponíveis. Deve atender aos seguintes aspectos:

- tal como na tarefa 2, assume-se que os dados de entrada estão de acordo com a especificação apresentada na Secção 2.1;
- os comandos são executados em sequência;
- comandos que não sejam aplicáveis deverão deixar o estado do *robot* inalterado (tal como na versão do jogo online);
- sempre que um comando L for executado com sucesso deverá ser impressa uma linha contendo as coordenadas x e y da posição onde o *robot* se encontra (separadas por um único espaço);
- quando todas as *lâmpadas* do tabuleiro se encontrarem ligadas, o programa imprime uma linha com a mensagem “FIM $\langle tick_count \rangle$ ” (em que $\langle tick_count \rangle$ é o número de comandos válidos executados). O programa deve depois terminar independentemente de ainda existirem comandos não processados;
- se a sequência de comandos terminar sem que todas as lâmpadas se encontrem ligadas deve imprimir a mensagem “INCOMPLETO”.

4 Entrega e Avaliação

A data limite para entrega de todas as componentes desta primeira fase do projecto é **30 de Novembro de 2014**, e a respectiva avaliação terá

um peso de 50% na nota final da UC. As tarefas computacionais deverão ser submetidas na plataforma **mooshak**, sendo que estas serão desde logo objecto de uma avaliação automática por parte da plataforma (com um peso discriminado abaixo). Cada grupo é responsável por submeter na plataforma **mooshak** unicamente programas da sua autoria¹.

Para além dos programas submetidos na plataforma **mooshak**, será considerada parte integrante do projecto todo o material de suporte à sua realização depositado no repositório SVN do respectivo grupo (código, documentação, ficheiros de teste, etc.). A utilização das diferentes ferramentas abordadas no curso (como **haddock**; **SVN**; **Make**; **L^AT_EX**; etc.) deve seguir as recomendações enunciadas nas respectivas sessões laboratoriais. A avaliação desta fase do projecto terá em linha de conta todo esse material, atribuindo-lhe os seguintes pesos relativos:

Avaliação automática das tarefas computacionais	30%
Avaliação qualitativa das tarefas computacionais e do processo de desenvolvimento	30%
Utilização do SVN e estrutura do repositório	10%
Quantidade e qualidade dos testes	10%
Documentação do código	10%
Definição da Makefile	5%
Esqueleto do relatório em L^AT_EX	5%

A nota final é atribuída independentemente a cada membro do grupo em função da respectiva prestação.

¹Os programas submetidos irão ser processados por ferramentas de detecção de plágio e, na eventualidade serem detectadas cópias, estas serão consideradas fraude dando-se-lhes tratamento consequente.