

# Laboratórios de Informática II

## LEI (1º ano)

# Projecto Integrado

Ano Lectivo de 2006/07

## 1 Objectivos

Com este projecto integrado pretende-se sedimentar os conhecimentos introduzidos nas aulas teóricas de Programação Imperativa, Sistemas de Computação, Física, Lógica e Cálculo II, relativos a:

- métodos rigorosos de análise de problemas (requisitos) e desenvolvimento de software;
- métodos de programação imperativa suportados em algoritmos e estruturas de dados lineares e sua implementação em C.
- técnicas de debugging, análise de programas e geração de código óptimo.

## 2 Resultados da aprendizagem

No final do semestre, os alunos devem ter adquirido um conjunto de competências genéricas e específicas detalhadas na página oficial da disciplina, em [www.di.uminho.pt/~prh/curli206.html](http://www.di.uminho.pt/~prh/curli206.html).

## 3 Organização e Funcionamento

O projecto será desenvolvido em grupos de 3 alunos dentro e fora das aulas da disciplina (1 sessão semanal de 2horas).

Em cada aula está presente um docente da equipa que irá esclarecendo questões específicas levantadas pelos elementos do grupo, quer ao nível da concepção e desenho da solução, quer ao nível da sua implementação e teste. Nos pontos de controlo (ver calendário abaixo) e no fim do semestre, cada grupo apresentará à equipe docente o trabalho realizado e os resultados obtidos, devendo entregar um relatório técnico de desenvolvimento devidamente estruturado e fundamentado, escrito em L<sup>A</sup>T<sub>E</sub>X (ver modelo disponível na página WWW da disciplina).

Em cada um desses pontos de controlo, serão escolhidos aleatoriamente alunos de alguns grupos para apresentar oralmente à turma (com recurso a um projector) o trabalho do grupo, de modo a promover uma discussão alargada das soluções encontradas.

### 3.1 Calendarização

O projecto deve ser executado ao longo de todo o semestre (o 2<sup>o</sup> do 1<sup>a</sup> ano), com 13 semanas, estando a **entrega** agendada para a primeira semana após a conclusão das aulas (**11 a 16 Jun**).

Para controlo da situação e avaliação intermédia, haverá 3 apresentações intercalares do projecto, nas semanas de: (**12 a 17 Mar**), (**16 a 21 Abr**) e (**21 a 26 Mai**).

## 4 Enunciado

Tal como a *arca de jogos* no sótão da nossa infância, o AJ é um programa final (uma *aplicação informática*) que nos permite jogar no computador vários jogos tradicionais.

O programa base é muito simples. Limita-se a pedir a identificação (ou registo, se for a primeira vez) do jogador e a colocar à sua escolha o conjunto de desafios lógicos que contém.

Seleccionado o jogo, entra-se no ambiente próprio dessa opção, na qual o jogador se manterá até explicitar que quer voltar ao início para escolher de novo, ou terminar a sessão. No fim de cada jogo, o AJ acumula na ficha do jogador os pontos obtidos. Terminada a sessão, o AJ mostra ao jogador, no monitor, a sua ficha, indicando os pontos acumulados em cada jogo, e grava a ficha para estar disponível para a próxima *visita ao sótão*.

Não sendo exaustiva, mostra-se abaixo a lista mínima de jogos que devem ser incluídos na *arca*. A ordem de apresentação corresponde à ordem de desenvolvimento.

1. **Jogo da Forca** (JF), em que o programa AJ propõe o enigma—palavra ou frase<sup>1</sup>—e o jogador tenta adivinhar letra-a-letra ou o termo completo de uma só vez;
2. **Mine-Sweeper** (MSw), em que se imita o famoso jogo de detecção de bombas num campo quadrado (de dimensão N) minado aleatoriamente. Nesta versão, o jogador pode pedir ajuda (até A vezes), sendo-lhe indicado pelo computador uma célula favorável para a próxima posição a destapar (aquela cuja vizinhança está mais livre de bombas);
3. **Spite and Malice** (SM), nesta paciência para 2 jogadores, o objectivo de ambos é ser o primeiro a livrar-se das cartas do seu monte de descarte jogando-as para as pilhas do centro numa sequência ascendente que começa com um Ás e acaba com uma rainha.
4. **Bem e Mal** (BM), ou o **Jogo da Verdade e da Mentira** em que, dada uma fórmula proposicional (proposta pelo jogador humano, ou gerada pelo computador) os dois jogadores têm objectivos contraditórios, um quer tornar a fórmula verdadeira e o outro quer torná-la falsa; uma jogada consiste em fixar o valor verdadeiro ou falso para uma variável proposicional da dita fórmula cujo valor ainda esteja em aberto; os jogadores vão jogando alternadamente, até que todas as variáveis tenham valor atribuído. Calculado o valor da fórmula (de acordo com os valores escolhidos para as variáveis e de acordo com as tabelas de verdade dos vários conectivos), ganha o jogador que atingiu o seu objectivo.
5. **Galo Cervejeiro** (GC), semelhante ao tradicional Jogo do Galo, jogador e computador defrontam-se sobre um tabuleiro quadrado (3x3) podendo em cada jogada: colocar numa casa vazia uma caneca de cerveja *cheia*, *meada*, *vazia*; ou beber um golo de cerveja de uma caneca não-vazia colocada numa casa do tabuleiro (cada golo é sempre exactamente igual a meia caneca). Ganha o primeiro que conseguir por em linha (horizontal, vertical, ou diagonal) três canecas com o mesmo nível (*cheia*, *meada*, *vazia*).

Como extra deixam-se abaixo mais alguns desafios:

1. **Master-Mind** (MM), em que se oferecem as duas hipóteses: ora o computador propõe o enigma (sequência de 4 cores escolhidas num conjunto de 6) e o jogador procura descobrir; ora o jogador pensa e é o computador que tenta adivinhar.

---

<sup>1</sup>É muito divertido e educativo considerar uma lista de Provérbios.

2. **Quatro em Linha (4L)**, é uma outra variante, bem conhecida, do Jogo do Galo no qual o objectivo é ir colocando peças numa das 7 colunas de um tabuleiro com 6 linhas, até se alinhar (na horizontal, vertical ou diagonal) 4 peças da mesma cor (do jogador ou do computador); como o tabuleiro está na vertical, o efeito da gravidade faz que cada peça colocada num coluna desça até à primeira casa livre a contar da base.
3. **Bubble Frenzy (BF)**, em que as bolas são partículas elementares que se combinam ou aniquilam segundo as regras da cromodinâmica quântica.

## 4.1 Tarefas a Desenvolver

Para isso é necessário realizar as seguintes tarefas, algumas das quais devem prosseguir em paralelo:

1. Fazer a análise exaustiva do problema no seu todo e de cada jogo em particular, de modo a conceber a arquitectura do sistema e representar essa arquitectura através de um diagrama de blocos.
2. Criar um algoritmo que modele o sistema principal e cada jogo em particular, tendo neste caso especial cuidado com as estruturas de dados a usar tendo em vista a eficiência final a atingir e a simplicidade/clareza dos referidos algoritmos.
3. Desenvolver um programa em C que implemente o algoritmo do sistema principal e funções para cada jogo específico; esse programa deve ter uma interface simples (alfa-numérica tradicional) e deve trabalhar com os dados em memória—armazenadas em *arrays* e *listas-ligadas*, conforme for mais apropriado—gravando, num ficheiro binário, no fim da execução toda a informação necessária para se poder recuperar numa execução seguinte.
4. Numa fase final, deve melhorar-se o programa usando uma interface gráfica (via janelas).

Note-se porém que esta lista não pretende definir ao detalhe os objectivos de cada fase do trabalho, as metas a cumprir para avaliação, ou as tarefas a fazer em cada semana. Para esse efeito irá sendo publicado e divulgado um Guião para apoio a cada fase do trabalho laboratorial, bem como se irão publicando os enunciados refinados de cada jogo.

## 5 Material necessário

Qt.	Descrição	Obs.
1	PC / Linux / gcc / L <sup>A</sup> T <sub>E</sub> X	

## 6 Elementos de estudo

Sugere-se a consulta à página oficial da disciplina, em

[www.di.uminho.pt/~prh/curli206.html](http://www.di.uminho.pt/~prh/curli206.html)

para ver a lista de bibliografia recomendada.