

Programação I

1º Semestre 2018/2019

3º Trabalho

Data de Entrega: 1 de Janeiro de 2019

Introdução

Neste trabalho, os alunos adquirem prática de utilização da matéria dada durante o semestre. Desenvolve-se uma abordagem estruturada (top-down) ao desenvolvimento de programas organizados em funções em vários ficheiros fonte (módulos).

Cada grupo entregará no site da sua turma um documento (com extensão docx, doc ou pdf) que descreva a solução encontrada, assim como os ficheiros fonte das respetivas soluções, devidamente comentados, comprimidos num ficheiro (com extensão zip, rar, gz, etc.).

É valorizada a simplicidade das soluções que cumpram o enunciado e a construção do código organizado em funções que minimizem a existência de código repetido.

Recomendações

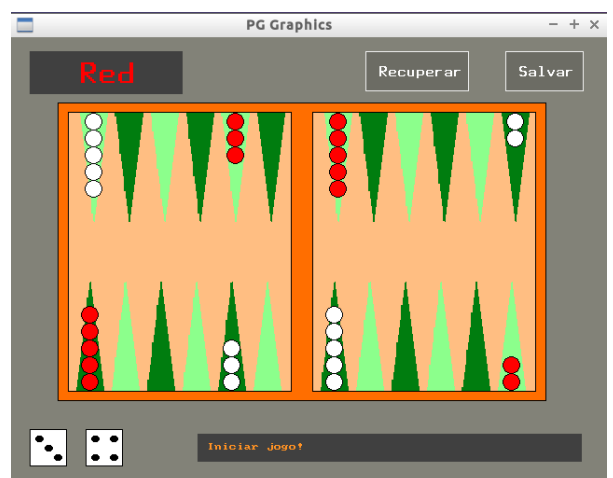
É valorizada a simplicidade das soluções que, no entanto, devem cumprir os requisitos do enunciado. A compilação dos programas não deve gerar *warnings*, os nomes das variáveis devem descrever o seu propósito e todos os valores lidos do teclado devem ser validados. A utilização de “números mágicos” no código devem ser evitados e recomenda-se a utilização de constantes através da definição de macros. O aluno deve citar os *sites* que consultou aquando da elaboração dos programas que implementou na realização do trabalho.

Objetivo

Pretende-se realizar uma aplicação em modo gráfico usando a biblioteca *libpg* que corra uma versão simplificada do jogo do gamão para dois jogadores humanos, com as regras especificadas no exercício 3 do 2º trabalho.

Cada jogador tem 15 peças (brancas ou vermelhas) distribuídas inicialmente num tabuleiro de 24 posições (casas). A Fig. 1 apresenta o tabuleiro tradicional de Gamão no seu estado inicial (antes de qualquer jogada).

Fig. 1 – Aspeto inicial do tabuleiro



Em anexo é fornecido o código fonte de uma versão incompleta do jogo do gamão em modo gráfico bem como o executável com uma versão completa do jogo (*gamon_ok*). O código está distribuído pelos seguintes ficheiros:

gamon.c – Implementação parcial do código de apresentação e de execução das jogadas. **A completar.**

model.c – Este módulo começará por idêntico ao desenvolvido por vós no 2º trabalho.

model.h – contém as assinaturas (protótipos) das funções presentes em **model.c** (completo e idêntico ao do 2º trabalho).

utils.c - funções utilitárias, por exemplo para a geração de valores aleatórios (completo).

utils.h – protótipos das funções definidas em **utils.c** (completo).

componentes.c – funções para suportar componentes gráficos como caixas de texto e botões (completo).

componentes.h – tipos e protótipos das funções definidas em **componentes.c** (completo)

dice.c – funções para suportar o componente com apresentação gráfica para representar um dado. **A completar.**

dice.h - tipos e protótipos das funções definidas em **dice.c**. Está completo.

O trabalho deverá ser desenvolvido nas seguintes fases:

1. Analise cuidadosamente o código fornecido em especial o presente no ficheiro **gamon.c** tentando perceber o papel dos tipos e funções nele presentes.
2. Completar o ficheiro **dice.c** de modo a que o componente dado funcione de forma idêntica à observada na aplicação **gamon_ok**.
3. Completar o ficheiro **gamon.c** (realizando o código necessário ao processamento de eventos de rato, possivelmente acrescentando mais funções), de modo a que a aplicação se comporte de forma idêntica ao executável fornecido em anexo (**gamon_ok**).
 - Os jogadores alternam entre si as jogadas.
 - Um turno de cada jogador consiste num máximo de duas jogadas correspondentes a cada um dos dados podendo terminar antes caso não possa efectuar nenhuma jogada com os dados disponíveis. Neste último caso aparece uma mensagem indicativa na caixa de texto ao fundo do tabuleiro.
 - Um jogador, usando clicks com o botão esquerdo do rato, começa por seleccionar o dado que pretende e de seguida a casa que pretende. Caso possível o movimento é realizado. Caso a jogada seja inválida é indicado o erro na caixa de texto ao fundo do tabuleiro
 - Um jogador com peças capturadas terá de as libertar seleccionando previamente um dado e depois a zona das peças capturadas

Nesta fase deverão ser usados, sem nenhuma alteração, os ficheiros **model.c** e **model.h** desenvolvidos no 2º trabalho.

4. Acrescentar o código necessário aos ficheiros **gamon.c** e (possivelmente) **model.c**, de modo a que o estado do jogo possa ser salvo e posteriormente recuperado utilizando botões (Salvar e Recuperar) da forma apresentada no executável fornecido em anexo.

Bom trabalho.

Diogo Cardoso,
Jorge Martins,
Manuel Carvalho