



Programação 2015/2016

Mestrado em Engenharia Electrotécnica e de
Computadores (MEEC)

Blackjack



Projeto de Programação – Entrega Intermédia

1 Introdução

Com este projeto pretende-se que os alunos desenvolvam um jogo de cartas intitulado *blackjack*, frequentemente jogado em casinos. O objetivo do *blackjack* é ter mais pontos que o adversário, mas sem ultrapassar os 21 pontos. Este jogo usa um ou mais baralhos de 52 cartas e para se obter uma vitória é necessário vencer a casa (adversário) de uma das seguintes formas:

- Obter 21 pontos (Ás e uma carta de 10 pontos) com as duas primeiras cartas que um jogador recebe.
- Obter uma pontuação mais elevada que a casa, sem exceder 21 pontos.
- Deixar a casa obter uma ou mais cartas adicionais até que a sua mão exceda 21 pontos.

2 Pontuação das Cartas

A mão mais elevada no *blackjack* é chamada justamente de blackjack e corresponde a 21 pontos. A pontuação de cada carta está definida na Tabela 1.






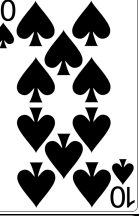
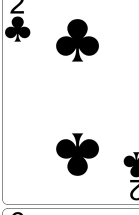
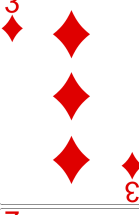
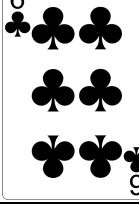
Cartas				Nome	Valor
				Ás	1 ou 11 pontos
				10, Valete, Rainha e Rei	10 pontos
				De 2 a 9	O valor indicado na carta
					

Tabela 1: Pontuação das cartas num jogo de *blackjack*.

3 Regras do Jogo Blackjack

Nesta implementação, quatro jogadores vão jogar *blackjack*. A casa (adversário) joga sempre contra os jogadores (humanos) e trata de todos os aspetos do jogo, tais como baralhar as cartas, distribuir cartas aos jogadores e gerir todas as apostas. O funcionamento da casa é controlado pelo computador.

Inicialmente, todos os jogadores devem realizar uma aposta, que tem um valor fixo (ver Secção 5). Não é permitido aos jogadores definir o montante que apostam. Depois, a casa atribui uma carta a cada jogador e uma carta para si, todas com a face voltada para cima. De seguida, noutra ronda, é distribuída uma carta a cada jogador (face para cima) e uma carta à casa com a face voltada para baixo. Desta forma, cada jogador tem duas cartas com a face voltada para cima e a casa tem uma carta com face voltada para cima e outra com a face voltada para baixo.

Depois de receberem as duas cartas iniciais, os jogadores podem escolher uma das seguintes opções:

- **Hit:** O jogador deseja mais uma carta.
- **Stand:** O jogador não quer pedir mais cartas.

Um jogador pode realizar vários pedidos de **Hit** até que realize um pedido de **Stand**. Sempre que haja um **Hit** a casa vai atribuir uma nova carta ao jogador. Sempre que as cartas de um jogador excederem 21, o jogador perde o valor apostado, isto é chamado de **Bust**. Note-se que um Ás pode valer 1 ou 11 pontos e um jogador pode tentar obter uma pontuação mais elevada, pedindo mais cartas e caso exceda 21 com um Ás (a valer 11 pontos) o Ás passa a valer 1 ponto.

No final de todos os jogadores fazerem **Stand** ou terem sofrido **Bust**, a casa vai virar a sua carta que está de face voltada para baixo. Se o total for 17 ou mais, a casa também faz um **Stand**. Se o total for 16 ou menos, deve pedir uma carta e continuar a pedir cartas até que some 17 ou mais pontos. No caso especial da casa possuir na sua mão, um Ás e um conjunto de cartas que totalize 6 pontos, deve também pedir mais uma carta.

As seguintes regras aplicam-se às apostas:

- Se um jogador tiver um *blackjack* nas duas primeiras cartas recebidas e a casa não tiver *blackjack*, a casa atribui ao jogador 1.5 vezes o valor apostado.
- Se a casa tiver um *blackjack*, recolhe as apostas de todos os jogadores que não tenham tido *blackjack*.
- Sempre que um jogador exceda o valor de 21 (**Bust**), perde *imediatamente* o valor apostado. Isto acontece mesmo que a casa faça **Bust**.
- Sempre que a casa exceder o valor de 21, atribui a cada jogador o valor apostado sem recolher a aposta, isto é, o jogador fica com duas vezes o valor da aposta.
- Sempre que a casa fizer **Stand** com um valor de 21 ou menos, atribui a cada jogador (que não tenha feito **Bust**) com uma pontuação mais alta o valor da aposta e recolhe as apostas dos jogadores com pontuações mais baixas.
- Em caso de empate, a casa não recolhe o valor apostado mas também não atribui mais dinheiro ao jogador.

4 Funcionamento do Programa

O funcionamento do programa pode ser sumariado da seguinte forma:

- Leitura dos parâmetros de configuração.
- O nome de cada jogador deve ser identificado, pela string "Player" e por um número único, e.g. Player 1, Player 2, Player 3 e Player 4.
- As cartas devem ser sempre baralhadas antes de começarem a serem distribuídas pela casa.
- Sempre que forem distribuídas cartas, é necessário actualizar a interface gráfica para mostrar as cartas aos jogadores.
- Depois de serem distribuídas as duas cartas iniciais, é necessário permitir aos jogadores (utilizadores do jogo) escolher se pretendem fazer **Hit** ou **Stand**, através do teclado:

- Tecla h: **Hit**
- Tecla s: **Stand**
- Tecla n: **Novo jogo**
- Devem dar oportunidade a cada jogador de fazer a sua escolha de uma forma sequencial e assinalar na interface gráfica qual o jogador que está a interagir com a casa (fazendo **Hit** ou **Stand**).
- O jogador quando está a interagir com a casa pode fazer **Hit** várias vezes recebendo uma carta da casa. Se ocorrer **Bust** ou fazer **Stand**, o controlo passa para o próximo jogador.
- No final de cada jogo, é necessário actualizar o dinheiro que cada jogador tem.
- A tecla novo jogo apenas tem efeito quando todas as cartas estiverem distribuídas.
- Um jogador quando não tiver mais dinheiro para apostar não pode jogar mais.
- O valor da aposta de cada jogador é fixo e é um parâmetro de configuração.
- O utilizador do programa pode sair a qualquer momento carregando a tecla q. Nesta situação o ficheiro de estatísticas deve ser escrito.

Em relação ao código, é obrigatório obedecerem às seguintes regras:

- Uma carta deve ser identificada através de um valor inteiro de 0 a 51. Não podem ser usadas estruturas de dados.
- Devem usar um vector de inteiros para representar um conjunto de cartas.

5 Leitura dos Parâmetros de Jogo e Escrita de Ficheiros

O funcionamento do jogo deve ser ditado por um conjunto de parâmetros que devem ser introduzidos pelo utilizador (teclado) antes de começar qualquer jogo:

- Número de baralhos de 52 cartas que vai ser utilizado pela casa. Não pode haver mais do que 6 baralhos nem menos que 1.
- Valor inicial de dinheiro (em euros) que cada jogador possui. Não pode ser um valor inferior a 10.
- Valor de cada aposta que cada jogador pode efetuar. Não pode ser um valor inferior a 1 nem ser superior a 20% do valor inicial de dinheiro do jogador.

Quando o programa terminar é necessário escrever um ficheiro com o nome *stats.txt* que indique a seguinte informação para cada jogador do *blackjack*:

- Nome do jogador.
- Número de jogos que ganhou, empatou e perdeu.
- Valor monetário com que o jogador ficou no final.
- Valor monetário que a casa ganhou ou perdeu.

6 Aspecto Gráfico

A aplicação *blackjack* deverá ter um aspeto gráfico semelhante ao que está ilustrado na Figura 1. A aplicação deverá também mostrar o nome da aplicação (Blackjack) e o autor da aplicação. Para realizar a parte gráfica

devem utilizar o código de apoio ao projeto. No entanto, note-se que necessitam de modificar o código de forma a mostrar qual o jogador que se encontra a jogar, por exemplo desenhando a área de jogo com outra cor. Opcionalmente, podem mostrar os pontos que cada jogador tem e marcar os jogadores que fizeram **Bust** (ver Figura 2).



Figura 1: Aplicação Blackjack ilustrando o jogador 2 a efectuar uma jogada.



Figura 2: Aplicação Blackjack ilustrando o momento em que a casa distribui as cartas e o dinheiro aos jogadores.

7 Desenvolvimento do Jogo

Também é fundamental que os alunos cumpram as regras que se seguem no desenvolvimento do jogo.

7.1 Desenvolvimento Faseado

O desenvolvimento deste projeto deverá ser feito de uma forma faseada, devendo os alunos garantir que todas as funcionalidades codificadas até ao momento estão a funcionar corretamente.

É preferível um programa que implementa poucas funcionalidades, mas que funcionam corretamente, do que um programa totalmente desenvolvido mas que não faz nada.

Assim sugerem-se os seguintes passos, pela ordem apresentada, para realização do projeto:

- Inicialização da biblioteca SDL e criação da janela gráfica (código fornecido pelo Prof.).
- Leitura dos parâmetros de funcionamento do programa.
- Baralhar as cartas da casa.
- Distribuir as cartas ao jogador.
- Atualização do ecrã.
- Interação com utilizador e reconhecimento das teclas s, h e n.
- Cálculo dos pontos de um conjunto de cartas.
- Verificar se ocorreu **Bust**.
- Rotina para cálculo do dinheiro das apostas.
- Escrita do ficheiro com os resultados dos jogos.

Os alunos deverão garantir a robustez da aplicação verificando todos os casos de erro (por exemplo quando um parâmetro de entrada não seja válido).

7.2 Documentação

O código produzido pelos alunos deverá ser comentado. Os comentários presentes no código deverão explicitar e explicar o funcionamento da aplicação assim como as decisões tomadas. As seguintes regras devem ser cumpridas:

- O código deve ser comentado sempre que realize alguma operação não óbvia.
- Os comentários devem ser claros, gramaticalmente corretos e usando frases simples.
- A declaração de todas as variáveis e constantes deve ser acompanhada de um comentário com uma breve descrição de para que servem.
- Cada bloco de código (seleção ou repetição) deve ser precedido de um breve comentário explicativo.
- Todos os programas devem ter um comentário inicial que identifique, no mínimo, o trabalho, o seu autor, o fim a que se destina e a data de realização.

7.3 Indentação

Um ponto fundamental na organização de escrita de código é a indentação, isto é, organização hierárquica das linhas de código, de acordo com âmbito onde elas se encontram. A indentação deixa o código fonte do programa mais organizado, mais legível, fácil de entender e de modificar, sendo uma parte essencial do trabalho.

7.4 Estrutura do Código

Todos os programas em C devem possuir a mesma estrutura genérica, composta pelas seguintes secções:

- Bloco de comentários.
- Diretivas #include.
- Constantes globais e variáveis globais (caso sejam necessárias).
- Declaração de funções.
- Função main().
- Definição de funções.

Como regra geral deve considerar que as funções devem caber num único ecrã, isto é, devem ter no máximo cerca de 30 linhas. Também deve cumprir as seguintes regras:

- Inicialize sempre as variáveis na sua declaração.
- Teste a validade dos parâmetros recebidos por uma função.
- Declare constantes e evite usar números no corpo das funções.
- Evite repetições de código, use funções, ciclos, etc.
- Evite o uso de variáveis globais.
- Não use **goto**.
- Escreva código simples e claro que um colega seu possa perceber !

7.5 Compilação

O compilador a usar na execução do projeto é o gcc em ambiente Linux. **Os projetos que não compilem, i.e. que tenham erros de sintaxe, não serão avaliados.** A existência de avisos durante a fase de compilação poderá ser indício da existência de problemas no código. Estes deverão ser eliminados corretamente ou ignorados com cuidado extremo.

7.6 Decisões do Projeto

Como em qualquer projeto de informática, o funcionamento do programa não está totalmente definido no enunciado, existindo algumas ambiguidades e omissões. Para resolver essas omissões os alunos deverão tomar algumas decisões aquando do desenvolvimento do projeto. Estas decisões devem ser fundamentadas, sem nunca ir contra o definido no enunciado.

7.7 Biblioteca SDL

Durante o desenvolvimento deste projeto devera ser usada a biblioteca SDL2. A aplicação deverá ser compilada usando as bibliotecas SDL2, SDL2_image, SDL2_ttf. Mais informação disponível aqui:

- <http://wiki.libsdl.org/APIByCategory>
- <https://wiki.libsdl.org/FrontPage>

8 Submissão

Os alunos deverão submeter o código desenvolvido através do sistema FENIX até 3 de Abril. Apenas será necessário entregar o código correspondente ao programa desenvolvido.

Só será aceite um ficheiro de texto com extensão `.c`. Não utilize um processador de texto (e.g. Word) para formatar o seu programa.

9 Plágio

Os trabalhos serão objeto de um sistema de deteção de plágio. Os alunos podem conversar entre si para discutir possíveis soluções para algum problema que tenham mas não podem partilhar código fonte. Nesta entrega intermédia, todo o código deve ser realizado individualmente! Se uma cópia for detetada os alunos envolvidos na cópia serão penalizados.

10 Avaliação da Entrega Intermédia

A avaliação do projeto terá em conta diversos parâmetros:

- Funcionalidades implementadas, tais como:
 - baralhar cartas,
 - distribuir cartas,
 - interface com o utilizador do programa,
 - escrita de ficheiros,
 - gestão dos dinheiro apostado e recebido, etc.
- Cumprimento das regras do jogo e das especificações do projeto.
- Estruturação da aplicação, nomeadamente o uso de funções.
- Tratamento de erros.
- Comentários e legibilidade do código.