

Introdução à Computação Licenciatura em Videojogos

Projecto Final - Cillionaire

1º Semestre - 2019/2020

1 Introdução

Pretende desenvolver-se uma versão adaptada do jogo “Quem Quer Ser Milionário” em Linguagem de programação C.

Este trabalho tem como objectivo a aplicação dos conhecimentos adquiridos ao longo do semestre.

2 Descrição do problema

Cillionaire é um jogo do tipo trivia jogado no terminal Linux. O jogador começa com 0 Eur, 1 joker 50:50 e 1 joker 25:75. Em cada ronda é colocada uma questão com nível de dificuldade crescente e apresentadas as hipóteses de resposta. Por cada resposta certa o jogador sobe um patamar até atingir os 100 000 Euro, ver Tab. 2.1. Se errar uma questão, irá cair um patamar. O jogador perde o jogo se errar duas vezes seguidas. Nesta versão, não existe limite de perguntas.

O joker 50:50 pode ser jogado depois de colocada a questão e antes de bloquear a resposta. Este joker descarta duas das hipóteses erradas, ficando apenas com a hipótese certa e uma errada. O joker 25:75 descarta uma das hipóteses erradas, deixando apenas três hipóteses. Em cada pergunta, apenas um joker pode ser jogado.

Tabela 2.1: Patamares e nível de dificuldade das perguntas que são feitas quando o jogador está nesse patamar.

Nível Actual	Nível de dificuldade das questões
0	easy
500	easy
1000	easy
2000	medium
5000	medium
10000	hard
20000	hard
50000	hard
100000	

2.1 Invocação (parâmetros do main)

O jogo deverá ser invocado pelo terminal e deverá aceitar parâmetros passados pela linha de comando. Os parâmetros são opcionais e ordem pela qual são colocados deverá ser indiferente ao funcionamento.

O parâmetro `-s <int>` recebe um número inteiro que deverá ser utilizado como *seed* do gerador de números de aleatórios. Exemplo:

```
./cillionaire -s 73
```

Neste exemplo o gerador de números aleatórios deverá ser inicializado com o número 73. Caso este parâmetro não seja passado pela linha de comandos, o programa deverá utilizar a função `time()` para fazer essa inicialização.

O parâmetro `-f <filename>` recebe o *path* para o ficheiro com as perguntas que deverá ser lido para a memória. Exemplo:

```
./cillionaire -f perguntas.ini -s 27
```

Neste exemplo o ficheiro `perguntas.ini` deverá ser lido e o gerador de números aleatórios deverá ser inicializado com o número 27. O programa poderá iniciar sem que este parâmetro seja passado pela linha de comandos. O formato do ficheiro de perguntas encontra-se descrito na Secção ??.

2.2 Funcionamento

Quando iniciado, o programa deverá apresentar o seguinte menu:

```
*****
***          CILLIONAIRE          *
*****
*** n <name>      - new game      *
*** q            - quit          *
*** h            - show this menu *
*** r <filename>  - resume game   *
*** s <filename>  - save progress and quit *
*** j 50         - play 50:50 joker *
*** j 25         - play 25:75 joker *
*** c           - show credits   *
*****
>
```

2.2.1 opção n

Se o jogador escolher a opção `n <name>`, irá iniciar um novo jogo. O programa deverá registar o nome do jogador e verificar e dar início à primeira ronda. Exemplo

```
>n pedro
*** Hi pedro, let's get started!
*****
*** Name:  pedro          *
*** Level: 0              *
*** j50:   YES            *
*** j25:   YES            *
*****
*** Question: What is the code name for the mobile operating system Android 7.0?
*** A: Ice Cream Sandwich
*** B: Jelly Bean
*** C: Nougat
*** D: Marshmallow
```

```
>C
*** Hooray!
*****
*** Name:  pedro          *
*** Level: 500           *
*** j50:   YES            *
*** j25:   YES            *
*****
```

Caso o utilizador não especifique o seu nome o programa deverá assumir o nome default "newbie".

2.2.2 opção q

A qualquer momento o utilizador pode escolher seleccionar a opção q. Nesse caso o programa deverá simplesmente terminar. Exemplo:

```
>q
*** Sad to see you go...
```

2.2.3 opção h

A qualquer momento o utilizador pode escolher seleccionar a opção h que simplesmente repete o menu inicial.

2.2.4 opção c

A qualquer momento o utilizador pode escolher seleccionar a opção c que deverá mostrar a informação sobre os criadores do jogo. A formatação com a qual esta informação se apresenta é livre.

2.3 opção j

Em cada ronda, o jogador pode utilizar (se ainda tiver) um joker antes de responder à questão. Ao utilizar o joker as hipóteses de resposta deverão ser novamente apresentadas retirando uma ou duas hipóteses consoante o tipo de joker que foi jogado. Exemplo:

```
*** Question: Which Beatles album does NOT feature any of the band members on it' s cover?
*** A: Rubber Soul
*** B: The Beatles (White Album)
*** C: Abbey Road
*** D: Magical Mystery Tour
>j 50
*** Question: Which Beatles album does NOT feature any of the band members on it' s cover?
*** B: The Beatles (White Album)
*** D: Magical Mystery Tour
>
```

O jogador tem apenas um joker 50:50 e um joker 25:75. Se tentar jogar um joker que já tenha sido jogado, ou se tentar jogar um joker o segundo joker na mesma ronda, o programa deverá dar a mensagem de erro "Illegal move" e continuar à espera que o utilizador bloqueie a resposta. Exemplo:

```
>j 50
*** Question: Which Beatles album does NOT feature any of the band members on it' s cover?
*** B: The Beatles (White Album)
*** D: Magical Mystery Tour
>j 25
*** Illegal move
>
```

2.3.1 opção s

A qualquer momento o utilizador pode escolher seleccionar a opção s <filename> que deverá gravar no ficheiro com nome passado como argumento toda a informação necessária para que o programa possa terminar e mais tarde retomar a execução no mesmo ponto em que estava. Exemplo:

```
> s jogo1.txt
*** Ok, your progress has been saved. See you later!
```

A formatação do ficheiro, assim como a informação a gravar é livre e deverá ser escolhida pelos alunos. Esta selecção deverá ser descrita e justificada no relatório.

2.3.2 opção r

O utilizador pode escolher a opção r <filename> apenas no início do programa. Depois do jogo começar, o jogador já não poderá seleccionar esta opção. Se o fizer o programa deverá mostra a mensagem "Illegal move". Exemplo:

```
*****
***          CILLIONAIRE          *
*****
*** n <name>      - new game      *
*** q            - quit          *
*** h            - show this menu *
*** r <filename> - resume game    *
*** s <filename> - save progress and quit *
*** j 50         - play 50:50 joker *
*** j 25         - play 25:75 joker *
*** c            - show credits  *
*****
>r game1.tmp
*** Ok pedro, where were we? Oh there you go:
*****
*** Name:  pedro          *
*** Level: 500           *
*** j50:   YES           *
*** j25:   YES           *
*****
*** Question: Which Beatles album does NOT feature any of the band members on it' s cover?
*** B: The Beatles (White Album)
*** D: Magical Mystery Tour
>
```

3 Ganhar ou perder o jogo

Se o utilizador atingir o último patamar ganha o jogo. Nesse caso o programa deverá terminar depois de apresentar a seguinte mensagem:

```
>B
*** Hooray!
*** This is incredible! You have won!
*** Congratulations pedro!
```

O utilizador é eliminado se errar duas questões seguidas. Por exemplo:

```

*** Question: Carcassonne is based on which French town?
*** A: Paris
*** B: Carcassonne
*** C: Marseille
*** D: Clermont-Ferrand
>A
*** Woops... That's not correct.
*** The correct answer was B: Carcassonne
*****
*** Name: pedro *
*** Level: 0 *
*** j50: YES *
*** j25: YES *
*****
*** Question: In a standard game of Monopoly, what colour are the two cheapest properties?
*** A: Green
*** B: Yellow
*** C: Brown
*** D: Blue
>A
*** Woops... That's not correct.
*** The correct answer was C: Brown
*** Sorry, you have lost the game. Bye!

```

4 Ficheiro de questões

É obrigatório fazer a leitura completa do ficheiro para a memória no início do programa. Depois de lido, o ficheiro deverá ser fechado e não deverá voltar a ser lido.

As perguntas deverão ser colocadas na memória sequencialmente e deverão ser lidas segundo uma política “first in first out” (FIFO). Quando uma pergunta é feita, deverá ser eliminada da memória, para que não volte a ser feita a mesma pergunta.

Se o jogador está no patamar 500 Euro, o programa deverá percorrer a lista de perguntas começando no início e deverá seleccionar a primeira pergunta que encontrar que seja de nível “easy”. Por outro lado, se o jogador está no patamar 20000 Euro o algoritmo deverá percorrer a lista começando no início e deverá seleccionar a primeira pergunta que encontrar com o nível de dificuldade “medium”.

Cada vez que uma pergunta é seleccionada, o programa deverá gerar um número aleatório entre 0 e 3. Esse número gerado irá determinar a posição em que a resposta correcta irá aparecer. Por exemplo, considere a pergunta: “what is the name of the scientist who derived the relativity theory?” Cujas resposta correcta é “Albert Einstein” e as opções são “Isaac Newton”, “Edward Lorenz”, e “Blaise Pascal”. Se o número gerado for 0 a questão deverá ser colocada da seguinte forma:

```

*** Question: what is the name of the scientist who derived the relativity theory?
*** A: Albert Einstein
*** B: Isaac Newton
*** C: Edward Lorenz
*** D: Blaise Pascal
>

```

Por outro lado se o número gerado for 3 a pergunta deverá surgir da seguinte forma:

```

*** Question: what is the name of the scientist who derived the relativity theory?
*** A: Isaac Newton
*** B: Edward Lorenz
*** C: Blaise Pascal
*** D: Albert Einstein
>

```

Se por alguma razão, não existirem perguntas com nível de dificuldade requerido o programa deverá terminar com a mensagem:

```
*** This is embarrassing but we're out of questions.
```

As perguntas que se encontram no ficheiro de perguntas não têm nenhuma ordem específica. O ficheiro deverá ter um número generoso de perguntas de vários níveis de dificuldade.

4.1 Formato do ficheiro de perguntas

O ficheiro deverá ter a extensão .ini e deverá seguir o formato do seguinte exemplo

```
; Base de dados de perguntas
; linhas que começam com ; deverão ser ignoradas
QUESTION=What amount of bits commonly equals one byte?
OPTION0=8
OPTION1=1
OPTION2=2
OPTION3=64
CATEGORY=Science: Computers
DIFFICULTY=easy
QUESTION=How many zeptometres are inside one femtometre?
OPTION0=1,000,000
OPTION1=10
OPTION2=1,000,000,000
OPTION3=1000
CATEGORY=Science: Mathematics
DIFFICULTY=hard
```

O número máximo de perguntas no ficheiro não é conhecido e portanto deverá ser alocada memória dinâmica para guardar as perguntas na memória. Aconselha-se a utilização de uma estrutura de dados do tipo lista ligada para guardar a informação.

O ficheiro apenas poderá ser lido uma vez durante a execução do programa.

4.2 Estruturas de dados

Cada pergunta deverá ser lida do ficheiro e guardada numa estrutura de dados. Essa estrutura deverá conter, no mínimo, os seguintes campos:

- **question:** String com tamanho máximo de 128 bytes.
- **answers:** Vector com 4 strings de 64 bytes cada uma. A resposta correcta deverá ser sempre colocada no índice 0 do vector.
- **category:** Tipo enumerado que identifica a categoria da pergunta.
- **difficulty:** Tipo enumerado que classifica o nível de dificuldade da pergunta.

Caso o aluno considere necessário, a estrutura pode conter mais campos.

As categorias possíveis são:

- Entertainment: Books
- Entertainment: Film
- General Knowledge
- Entertainment: Music
- Entertainment: Musicals and Theatres

- Entertainment: Television
- Entertainment: Video Games
- Entertainment: Board Games
- Science and Nature
- Science: Computers
- Science: Mathematics
- Mythology
- Sports
- Geography
- History
- Politics
- Art
- Celebrities
- Animals
- Vehicles
- Entertainment: Comics
- Science: Gadgets
- Entertainment: Japanese Anime and Manga
- Entertainment: Cartoon and Animations

5 Memory leaks

O programa não deverá conter *memory leaks*. Para verificar se um programa tem memory leaks pode usar-se a ferramenta valgrind. A existência de *memory leaks* será testada na plataforma de testes.

6 Relatório e Qualidade do Código - 3 Valores

Cada grupo deverá incluir um breve relatório utilizando **Markdown** composto por:

- Título
- Nomes dos autores (primeiro e último) e respetivos números de alunos
- Nome do grupo de trabalho (de acordo com as plataforma pandora [1])
- Descrição da solução
- Estrutura do software
- Conclusões e matéria aprendida
- Referências, incluindo trocas de ideias com colegas, código aberto reutilizado e bibliotecas utilizadas

O código deverá estar correctamente indentado e comentado. É estritamente proibida a utilização da instrução goto. A utilização de variáveis globais (excepto constantes) é também proibida.

6.1 Compilação

O ficheiro fonte deve ser escrito em linguagem C. O código deverá ser compilado em linux utilizando o gcc [2] com as seguintes opções:

```
|| -g -Wall -std=c99 -Wpedantic -Wextra
```

7 Material a entregar

Cada grupo deverá entregar no moodle um ficheiro zip contendo

- Ficheiro '.c' com código devidamente comentado e indentado.
- A pasta '.git' onde deverá ser possível visualizar *commits* feitos pelos vários elementos do grupo
- Um relatório em formato **markdown**.

O nome do ficheiro zip deverá ser o nome do grupo de acordo com a plataforma pandora[1]. **Apenas um elemento do grupo deve submeter o ficheiro no Moodle.**

8 Forma e data de entrega

Os alunos deverão formar grupos de 1, 2 ou 3 alunos. Os grupos devem estar claramente identificados no relatório com nome e número. **Todos os elementos do grupo têm que estar registados no Pandora e têm que ser membros aprovados da equipa.**

Os alunos deverão submeter os seus trabalhos na plataforma de testes Pandora [1]. A plataforma irá realizar vários testes. O número de testes bem sucedidos irá determinar a nota mínima e máxima do grupo. A plataforma apenas aceitará trabalhos até às 10:00 de 13 de Janeiro de 2020.

Adicionalmente os alunos deverão que entregar através da plataforma moodle o ficheiro zip com o material descrito na secção 7. O material poderá ser entregue no moodle até às 23:55 do dia 19 de Janeiro de 2020.

9 Peso na avaliação

Este trabalho será cotado de 0 a 20 valores e terá um peso de 40% da nota da componente prática da unidade curricular.

10 Honestidade académica

Nesta disciplina, espera-se que cada aluno siga os mais altos padrões de honestidade académica. Isto significa que cada ideia que não seja do aluno deve ser claramente indicada, com devida referência ao respectivo autor. O não cumprimento desta regra constitui plágio [3].

O plágio inclui a utilização de ideias, código ou conjuntos de soluções de outros alunos ou indivíduos, ou quaisquer outras fontes para além dos textos de apoio à disciplina, sem dar o respectivo crédito a essas fontes. Os alunos são encorajados a discutir os problemas com outros alunos e devem mencionar essa discussão quando submetem os Projectos. Essa menção **não** influenciará a nota. Os alunos não deverão, no entanto, copiar códigos, documentação e relatórios de outros alunos, ou dar os seus próprios códigos, documentação e relatórios a outros em qualquer circunstância. De facto, não devem sequer deixar códigos, documentação e relatórios em computadores de uso partilhado.

Nesta disciplina, a desonestidade académica é considerada fraude, com todas as consequências legais que daí advêm. Qualquer fraude terá como consequência imediata a anulação dos Projectos de todos os alunos envolvidos (incluindo os que possibilitaram a ocorrência). Qualquer suspeita de desonestidade académica será relatada aos órgãos superiores da escola para possível instauração de um processo disciplinar. Este poderá resultar em reprovação à disciplina, reprovação de ano ou mesmo suspensão temporária ou definitiva da ULHT.

11 Licenças

O enunciado e restante documentação são disponibilizados através da licença [CC BY-NC-SA 4.0].

Referências

- [1] Pandora. Universidade Lusofona de Humanidades e Tecnologias. [Online]. Available: <https://ikaika.ulusofona.pt>
- [2] GNU Project. GCC, the GNU Compiler Collection. Free Software Foundation. [Online]. Available: <http://www.gnu.org/software/gcc/gcc.html>
- [3] “Texto adaptado da disciplina de algoritmos e estruturas de dados,” Instituto Superior Técnico.
- [4] A. Pereira, *C e Algoritmos*, 2nd ed. Sílabo.
- [5] Moodle. Universidade Lusofona de Humanidades e Tecnologias. [Online]. Available: <https://secure.grupolusofona.pt/ulht/moodle/>
- [6] Github. GitHub. [Online]. Available: <https://github.com/>
- [7] Repositório do mini projecto de introdução à computação. Universidade Lusofona de Humanidades e Tecnologias. [Online]. Available: https://github.com/VideojogosLusofona/ic_2019_trabalho1