

CASINO NIGHT

Henrique Nobre de Figueiredo



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2016

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Sistemas e Aplicações Distribuídas em Telecomunicações, do 1º ano, do Mestrado em Engenharia Electrotécnica e de Computadores - Telecomunicações



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

19 de junho de 2016

Índice

ÍNDICE	III
ÍNDICE DE FIGURAS	V
ÍNDICE DE TABELAS	VII
1. INTRODUÇÃO	1
1.1. PROBLEMA/JOGO	1
1.2. OBJECTIVOS.....	1
1.3. CONHECIMENTOS NECESSARIOS	2
1.4. ORGANIZAÇÃO DO RELATÓRIO	3
2. ORGANIZAÇÃO LÓGICA	5
2.1. DIAGRAMA SEQUENCIAL	5
2.2. FLUXOGRAMAS DE FUNCIONAMENTO.....	6
3. OBJETOS E EVENTOS	11
3.1. SERVESORTES.....	11
3.2. PROTOCOLO	11
3.3. APLETCASINO	12
3.4. CASINO	12
3.5. JOGOIMG.....	13
3.6. BUTTON.....	13
3.7. BOARD.....	14
3.8. ROLETA	14
3.9. USER.....	15
3.10. CARD.....	16
3.11. DECK	16
3.12. BLACKJACK	16
3.13. SLOTCHOICE.....	17
3.14. SLOTLINE	18
3.15. SLOTMACHINE	18
3.16. SOUND.....	19
3.17. PLAYERSONG	19
3.18. HEADER.....	19
4. CONCLUSÕES.....	21
REFERÊNCIAS DOCUMENTAIS	23

Índice de Figuras

Figura 1	– Diagrama Sequencial do Casino <i>Night</i>	5
Figura 2	– Fluxograma da Página Principal.....	6
Figura 3	– Fluxograma da Roleta.	7
Figura 4	– Fluxograma da <i>Slot Machine</i>	8
Figura 5	– Fluxograma do <i>Blackjack</i>	9
Figura 6	– Janela do <i>Applet</i>	12
Figura 7	– Output do Menu do Casino.	13
Figura 8	– Botões de aposta da roleta, com o número 14 selecionado.	14
Figura 9	– Classe Board com Timer activo.	14
Figura 10	– “Roleta Russa” janela de jogo.	15
Figura 11	- Exemplo de imagem criada com a classe “Card”	16
Figura 12	- Janela do jogo <i>Blackjack</i> com o resultado de uma mão.	17
Figura 13	– Exemplo de símbolo (Lima) na <i>Slot Machine</i>	17
Figura 14	– Janela do jogo <i>Slot Machine</i>	18
Figura 15	– Classe “PlayerSong” com botões de controlo da música.	19
Figura 16	– Classe Header integrada num dos jogos.....	19

Índice de Tabelas

Tabela 1	Lista de probabilidades utilizadas para as apostas na roleta.....	2
----------	---	---

1. INTRODUÇÃO

Este trabalho foi realizado no âmbito da unidade curricular de Sistemas e Aplicações Distribuídas em Telecomunicações tendo como principal objectivo a aplicação dos conhecimentos obtidos ao longo do semestre sobre a linguagem de programação JAVA.

1.1. PROBLEMA/JOGO

De modo a poder maximizar a aprendizagem e desenvolvimento de diferentes funções de JAVA foi proposto o desenvolvimento da série de jogos do trabalho anterior “Casino Night” de forma distribuída. Pretende-se que sejam desenvolvidos os seguintes jogos: *Roleta*, *Slot Machine* e *Blackjack*.

1.2. OBJECTIVOS

Os principais objectivos deste trabalho são:

- ➔ Desenvolvimento de uma aplicação em JAVA;
- ➔ Desenvolvimento da aplicação em versão *Stand-alone* e versão *Applet*;
- ➔ Uso, aprendizagem e desenvolvimento com Eventos.

- ➔ Uso, aprendizagem e desenvolvimento de diferentes tipos de conexão Cliente-Servidor.

1.3. CONHECIMENTOS NECESSARIOS

Pela especificidade do trabalho, certos requisitos são necessários para compreender e implementar este trabalho.

- ➔ Conhecimentos do modelo de gestão de eventos 1.1;
- ➔ Capacidades de desenvolvimento com as bibliotecas *Swing*;
- ➔ Conhecimento de *Timers* da biblioteca *Swing*;
- ➔ Conhecimento de *Threads* da biblioteca *Swing*.
- ➔ Capacidade com uso de *Sockets* assim como *ObjectInputStream* e *ObjectOutputStream*.
- ➔ Conhecimento das probabilidades da roleta listadas na tabela 1 [1]:

Tabela 1 Lista de probabilidades utilizadas para as apostas na roleta.

Aposta	Probabilidade
Apostar num número	35:1
Apostar nos terços (1-12,13-24,25-36)	2:1
Apostar numa cor	1:1
Apostar par ou impar	1:1
Apostar num grupo de números (1-18,19-36)	1:1
Apostar na coluna do número	2:1

➔ Conhecimento das regras do *Blackjack* listadas abaixo [2]:

- Cada carta numérica corresponde ao seu valor, cada figura corresponde a 10 pontos e um Ás corresponde a 11 ou 1 ponto, dependendo da forma mais vantajosa de constituir a mão.
- O vencedor é aquele que tem a mão mais próxima ou igual a 21 pontos sem ultrapassar este valor.
- Caso o jogador ultrapasse este valor é considerado que rebenta (*bust*) e perde automaticamente a mão.
- O jogador pode pedir cartas até estar satisfeito ou até a sua mão totalizar um valor superior a 21.
- A primeira carta do *dealer* é visível para todos os jogadores.
- O *dealer* tem de virar sempre cartas até totalizar um valor igual ou superior a 17, mesmo que os valores das mãos na mesa sejam inferiores. O *dealer* pára de virar cartas quando a sua mão totaliza entre 17 e 21, ou quando rebenta (*bust*).
- Cada mão pode ser constituída no máximo por 5 cartas.
- Caso um jogador tenha um par de mão, poderá fazer um *Split* que consiste em dobrar a aposta, com o benefício de jogar duas mãos, cujas primeiras cartas serão cada um dos pares.
- Se o jogador tiver um Ás e uma figura, terá automaticamente *Blackjack* (21 pontos) e receberá o triplo da sua aposta.

1.4. ORGANIZAÇÃO DO RELATÓRIO

No capítulo 1 poderá ser encontrada uma estruturação do trabalho, os objectivos do mesmo e alguns requisitos para a compreensão e uso do mesmo.

No capítulo 2 é apresentada a lógica pretendida da aplicação a desenvolver, juntamente com fluxogramas e diagramas para demonstrar o seu funcionamento.

O capítulo 3 é dedicado à explicação da função dos objectos utilizados na aplicação assim como os métodos criados para os mesmos.

Finalmente uma breve conclusão sobre o trabalho é apresentada no capítulo 4.

2. ORGANIZAÇÃO LÓGICA

Neste capítulo estão presentes diagramas e fluxogramas de modo a apresentar a lógica da aplicação/jogo.

2.1. DIAGRAMA SEQUENCIAL

Na figura seguinte é possível observar o funcionamento geral da aplicação. Posteriormente serão apresentados fluxogramas das funcionalidades em particular de cada jogo.

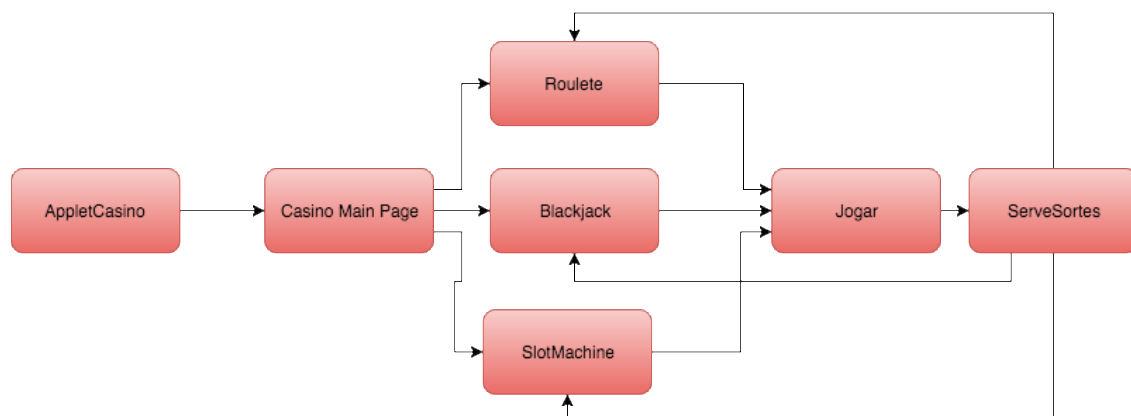


Figura 1 – Diagrama Sequencial do Casino Night

É possível correr este jogo através do “ApletCasino” (abrir como *Applet*) ou ainda directamente através do “Casino Menu” (como aplicação *stand-alone*), mas para o funcionamento desta aplicação o servidor “ServeSortes” terá sempre de correr em paralelo.

2.2. FLUXOGRAMAS DE FUNCIONAMENTO

De modo a perceber as funcionalidades implementadas em cada jogo breves fluxogramas de lógica vão ser apresentados nesta secção.

2.2.1. PÁGINA PRINCIPAL

Na figura abaixo podemos ver um esquema ilustrativo do funcionamento da página principal. Este será uma janela simples que permitirá a escolha do jogo.

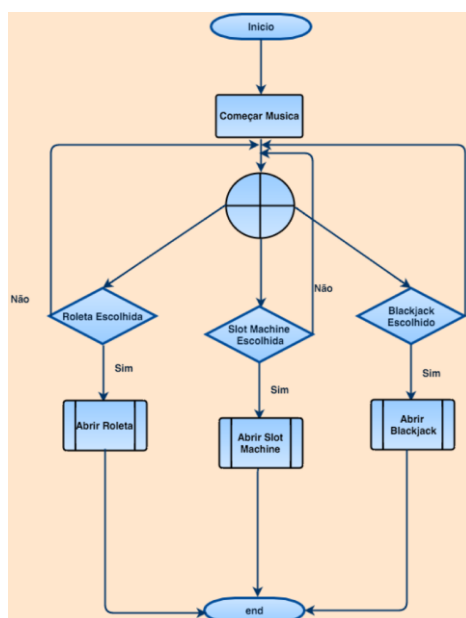


Figura 2 – Fluxograma da Página Principal.

A página principal é constituída por vários botões que dão acesso aos diversos jogos, assim como um gestor da música de fundo.

Esta página ainda não trata da conexão com o servidor, essa será feita em cada um dos jogos, sendo que a primeira mensagem enviada ao Servidor é um seletor com o tipo de jogo escolhido.

2.2.2. ROLETA

O jogo da roleta vai ser constituído por três áreas: Cabeçalho (que incluirá o saldo, o valor da aposta e o gestor de música), a Roleta (que incluirá a roleta, o número que saiu e os ganhos da aposta) e ainda Área de Aposta (que incluirá todos os botões de possíveis apostas, assim como as suas probabilidades). Será também pedido o nome aos jogadores assim que este jogo seja selecionado.

Como foi referido anteriormente é em cada jogo que a conexão é estabelecida com o servidor. Neste caso o servidor tratará da geração do número *random* para a roleta, assim como a gestão dos saldos dos jogadores.

A figura abaixo é um fluxograma que ilustra a sequência do pretendido neste jogo, representando também a troca de mensagens com o servidor.

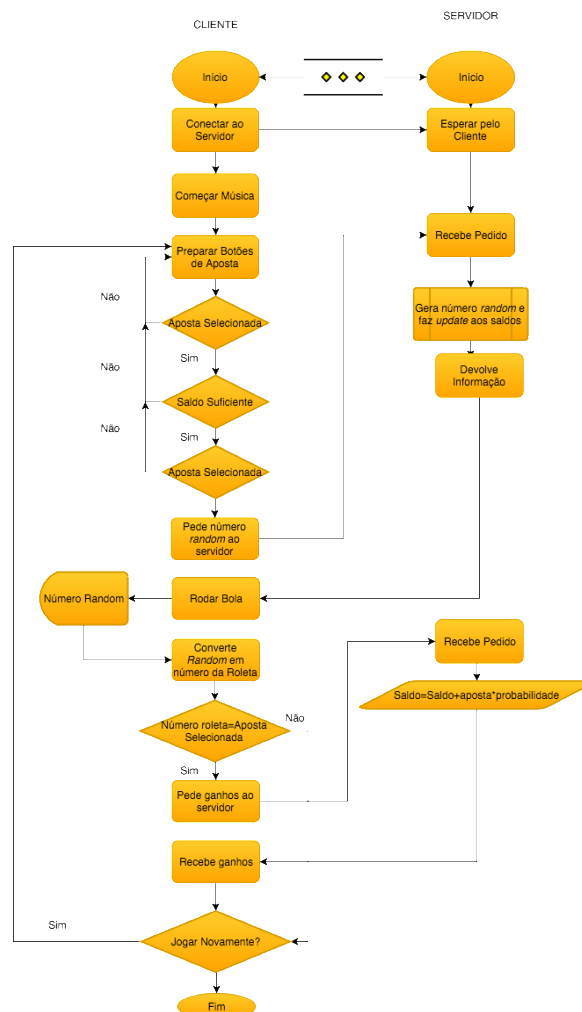


Figura 3 – Fluxograma da Roleta.

2.2.3. *SLOT MACHINE*

A *Slot Machine* será constituída por painéis com os diferentes tipos de imagens possíveis, assim como uma imagem com o valor dos prémios e os seus multiplicadores. Os painéis deveram fixar numa imagem sequencialmente. Será também pedido o nome aos jogadores assim que este jogo seja seleccionado.

Como foi referido anteriormente é em cada jogo que a conexão é estabelecida com o servidor. Neste caso o servidor tratará da geração tempo de rotação dos painéis, assim como a gestão dos saldos dos jogadores.

Na figura seguinte podemos ver um fluxograma que ilustra a sequência a ser utilizada neste jogo.

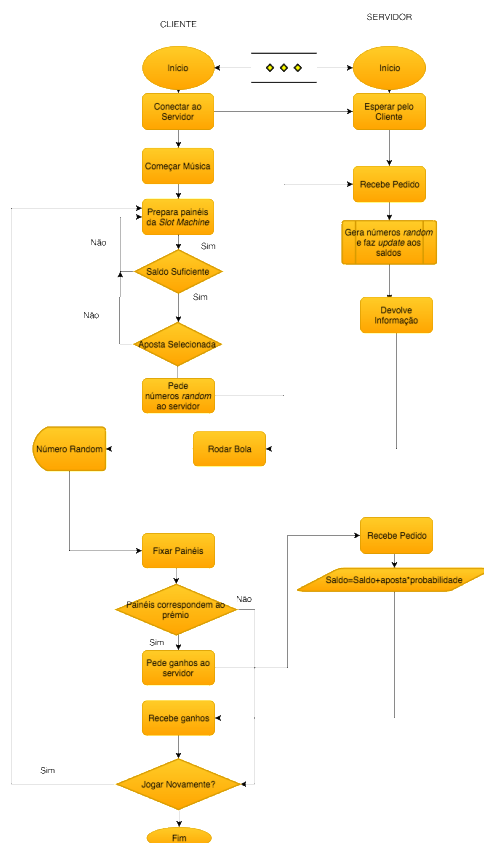


Figura 4 – Fluxograma da *Slot Machine*.

2.2.4. BLACKJACK

Este jogo será constituído por três partes: O cabeçalho, as cartas do *dealer* e respectivo *score* e as cartas do jogador. Deverá ser preparado um baralho que será utilizado para cada jogo e deverá ser apresentada a primeira carta do *dealer* de acordo com as regras. Será também pedido o nome aos jogadores assim que este jogo seja selecionado.

Como foi referido anteriormente é em cada jogo que a conexão é estabelecida com o servidor. Neste caso o servidor tratará dos *scores* do jogador e da casa, assim como a gestão dos saldos dos jogadores.

Na figura seguinte podemos ver um fluxograma que ilustra a sequência a ser utilizada neste jogo. Pela sua extensão, a opção “Split” está representada como um processo pré-definido e deverá jogar duas mãos seguidas caso o utilizador tenha dinheiro e caso as suas duas primeiras cartas sejam iguais.

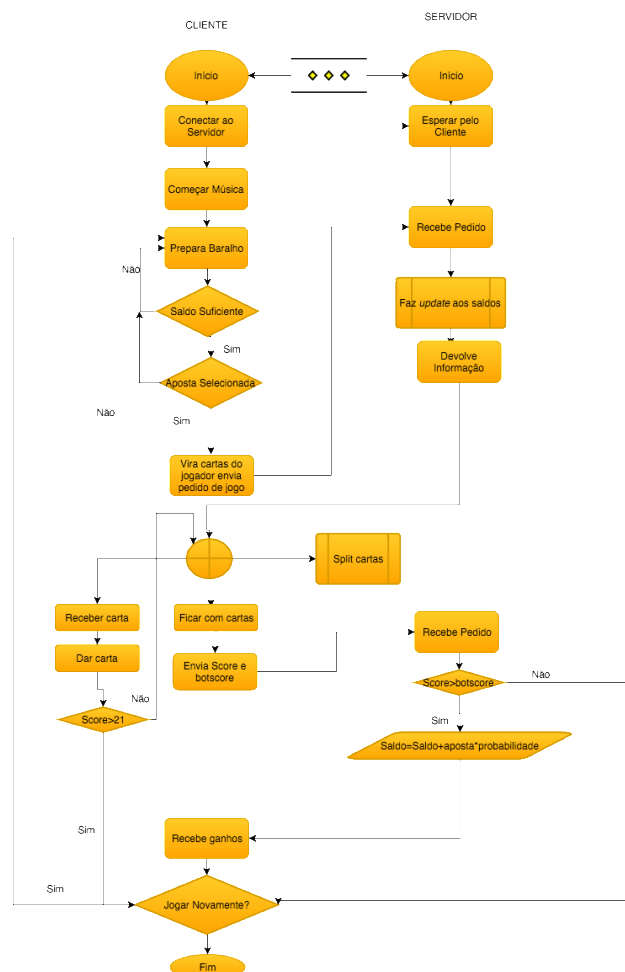


Figura 5 – Fluxograma do *Blackjack*.

3. OBJETOS E EVENTOS

Neste capítulo poderão ser encontrados as descrições dos objectos criados para os efeitos do trabalho, assim como os métodos e eventos correspondentes.

3.1. SERVESORTES

Esta classe funciona como servidor de todos os jogos. É criado um *Socket* de servidor que vai atender cada um dos clientes criando uma *thread* para cada um. “ServeSortes” cria também um ficheiro *log* quando os jogadores se desconectam, assim como um ficheiro com os *HighScores*.

Caso fosse utilizado um domínio publico para correr a aplicação, o servidor deveria estar continuamente a correr.

3.2. PROTOCOLO

De modo a criar uma forma universal de transmitir mensagens entre o cliente e o servidor foi criada esta classe que contém quatro objetos (Quantidade máxima de informação diferente a ser enviada nos jogos). De modo a utilizar esta classe, os objetos são preenchidos com a informação relevante e enviados/recebidos através de métodos desta própria classe.

Adicionalmente esta classe tem de implementar *Serializable* de modo a enviar e receber objetos.

3.3. APLETCASINO

Esta classe cria a ponte entre a aplicação em modo *Applet* e em modo *Stand-alone*. Na figura abaixo podemos ver um output quando a aplicação é corrida como *Applet*.

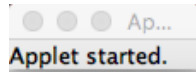


Figura 6 – Janela do *Applet*.

Assim sendo, esta classe pode ser considerada a classe principal e é aquela que contém o método:

```
public static void main(String[] args)
```

Assim como o método:

```
public void init()
```

De modo a poder utilizar estas funcionalidades a classe “apletcasino” tem de conter as características de um *Applet*.

3.4. CASINO

Esta classe é uma *JFrame* (Janela) que contém o acesso a cada um dos jogos através de botões.

É de referir também que todas as classes que são janelas contêm a classe *PlayerSong* (Reprodutor de áudio) que será explicada posteriormente.

Na imagem abaixo podemos ver o output deste menu.

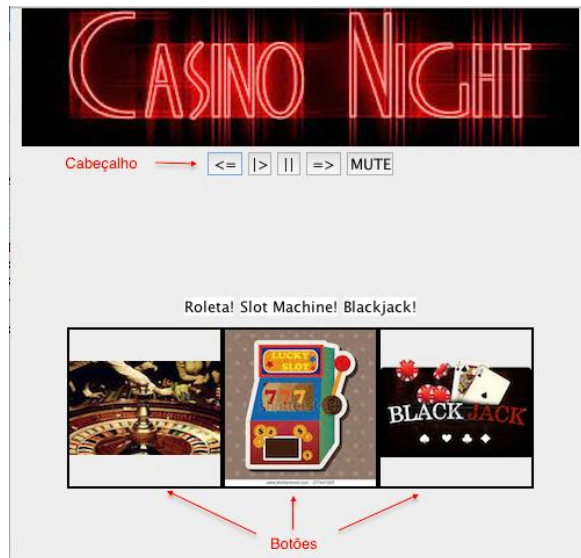


Figura 7 – Output do Menu do Casino.

3.5. JOGOIMG

Esta é uma classe auxiliar criada para poder adicionar imagens em coordenadas específicas da janela. Um exemplo do uso desta classe pode ser visto na Figura 7, na imagem que tem escrito “Casino Night”. As imagens desta aplicação geralmente são geradas através de *JLabel*'s mas de modo a poder utilizar a sua posição absoluta esta classe teve de ser utilizada pontualmente. Abaixo podemos ver o construtor desta classe.

```
public Jogoimg(String url,int cordx,int cordy)
```

3.6. BUTTON

A classe “Button” corresponde aos botões de aposta da roleta. Esta classe, têm a chamada de todas as imagens de botões de aposta, assim como um método que converte o botão selecionado na probabilidade de o mesmo sair, isto é, cria um multiplicador que juntamente com o valor da aposta, determina o valor total dos ganhos.

Na figura seguinte podemos ver um “ArrayList” de objectos desta classe.

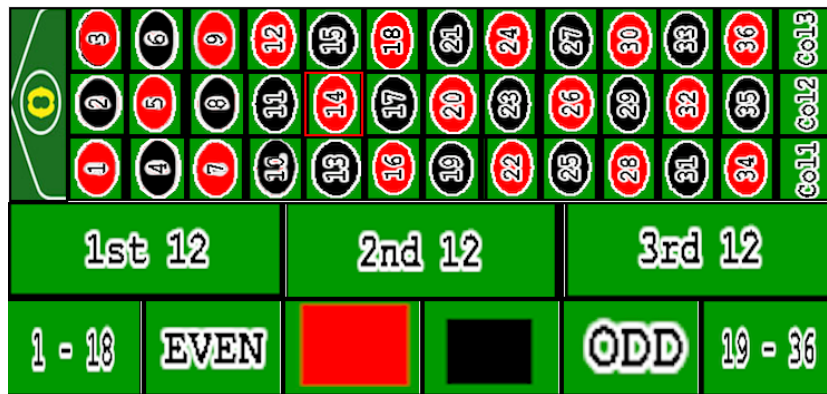


Figura 8 – Botões de aposta da roleta, com o número 14 selecionado.

3.7. BOARD

Esta classe é das mais relevantes para o jogo da roleta, pois ela responsável pela animação da bola à volta da imagem da roleta. Este processo é parado de acordo com um número *random* que determina o ângulo em que a bola ficará.

É importante referir também, de modo a evitar que o período de rodar a bola seja pequeno demais, foram implementadas um mínimo de duas voltas à roleta antes de a bola parar.

Na figura abaixo podemos ver a classe “Board” representada.



Figura 9 – Classe Board com Timer activo.

3.8. ROLETA

A “Roleta” é classe responsável pela criação da janela de jogo, assim como o carregamento de cada um dos seus componentes e ligação ao servidor.

Quando um utilizador tenta jogar, esta classe gera uma mensagem com um pedido para o servidor. Este devolve um número *random* que é convertido para o correspondente ângulo em que a bola parou no multiplicador responsável pelos ganhos da aposta.

Caso a aposta corresponda ao número em que a bola parou, o servidor incrementa o saldo do utilizador.

Na figura seguinte podemos ver a janela “Roleta Russa” com todos os componentes anteriormente referidos.

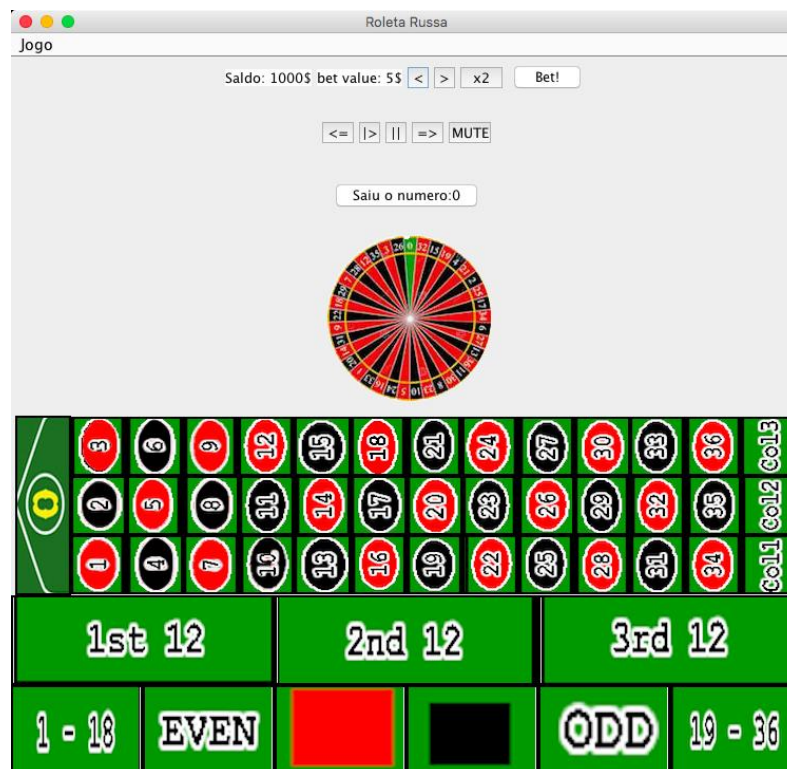


Figura 10 – “Roleta Russa” janela de jogo.

3.9. USER

Esta classe é uma classe auxiliar, constituída na sua maioria por métodos *Getters* e *Setters*. É que que a informação de um jogador é armazenada, como por exemplo, o saldo, o valor da aposta, entre outros. Apesar de conter métodos específicos para o jogo de *blackjack*, os restantes métodos são aplicáveis a aos restantes jogos.

Adicionalmente esta classe tem de implementar *Serializable* de modo a poder ser transmitida entre o servidor e o cliente.

3.10. CARD

A classe de criação dos diferentes naipes e número/figura de um baralho de cartas. Esta consiste simplesmente na construção de um *JPanel* com a imagem correspondente à carta especificada no construtor. Esta mantém também registo do naipe e número/figura.

Na figura seguinte podemos ver o exemplo de um *JPanel* construído através desta classe.



Figura 11 - Exemplo de imagem criada com a classe “Card”

3.11. DECK

De modo a organizar e manter o número de cartas de acordo com o regular, esta classe foi criada para preparar um baralho de cartas para o *Blackjack*. Utilizando a classe vista anteriormente “Card”, cria e posteriormente baralha 52 objectos do tipo “Card” e guarda-os num “ArrayList”. Nesta classe esta também definido o método de remover cartas do baralho que é utilizado para dar cartas aos jogadores.

3.12. BLACKJACK

Esta classe constitui a janela principal do jogo *Blackjack*. São criados dois objectos “User”, o *dealer* e o jogador, atribuídas duas cartas a cada um e estabelecida a conexão com o servidor. Quando o jogador aposta a suas cartas são reveladas assim como a primeira do *dealer*. De acordo com as opções “Hit”, “Stand” ou “Split” o jogo é processado de acordo com o fluxograma da Figura 5.

Esta ainda contida nesta classe o método de conversão do valor das cartas em pontos de modo a calcular o vencedor da mão, posteriormente estes pontos são enviados para o servidor e comparados. Finalmente o servidor responde com o saldo atualizado do jogador, de acordo com o resultado da jogada. Na seguinte figuras poderá ser vista uma mão a ser jogada neste jogo.

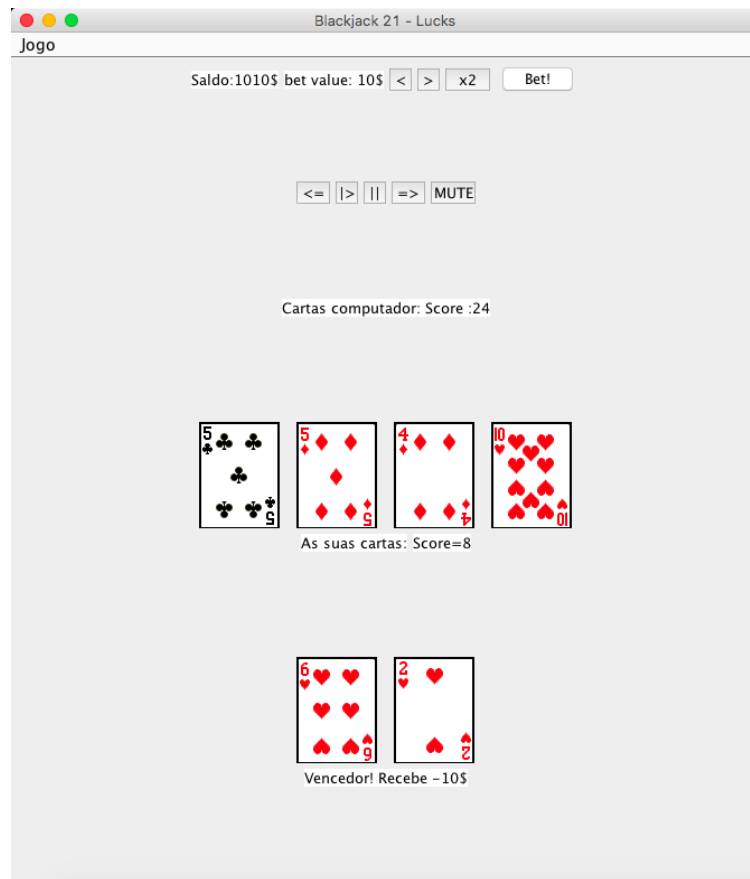


Figura 12 - Janela do jogo *Blackjack* com o resultado de uma mão.

3.13. SLOTCHOICE

Uma classe muito semelhante à classe “Card” mas para cada símbolo presente na *Slot Machine*. Analogamente o símbolo é criado num *JPanel* e é mantido um registo numérico do mesmo.

Na Figura 13 podemos ver um dos símbolos desta classe representado.



Figura 13 – Exemplo de símbolo (Lima) na *Slot Machine*.

3.14. SLOTLINE

De modo a associar uma probabilidade para cada símbolo foi criada esta classe que consiste num *JPanel* com um *CardLayout* de modo a conter todos os símbolos repetidos uma determinada quantidade de vezes. Este painel contém 316 símbolos que parece elevado, mas que se tornou necessário para poder diferenciar o valor dos prémios.

O algoritmo determina também que não podem existir símbolos seguidos. Este objecto é utilizado para posteriormente criar uma animação.

3.15. SLOTMACHINE

A janela do jogo *Slot Machine* é representada por este objecto e integra um cabeçalho, uma tabela de prémios, três “SlotLinha”, uma alavanca para iniciar o jogo e a ligação ao servidor.

Quando a alavanca é clicada, é enviado um pedido ao servidor e é ativo um *timer* que avança uma casa cada um dos “SlotLinha” por um determinado tempo aleatório (tempo aleatório retornado do servidor). É sempre assegurando um mínimo de tempo antes de cada uma das linhas parar, no final caso a combinação corresponda a alguma das combinações listadas, o jogador recebe a sua aposta vezes o multiplicador do prémio.

Na Figura seguinte podemos ver a janela “Slot Machine”.

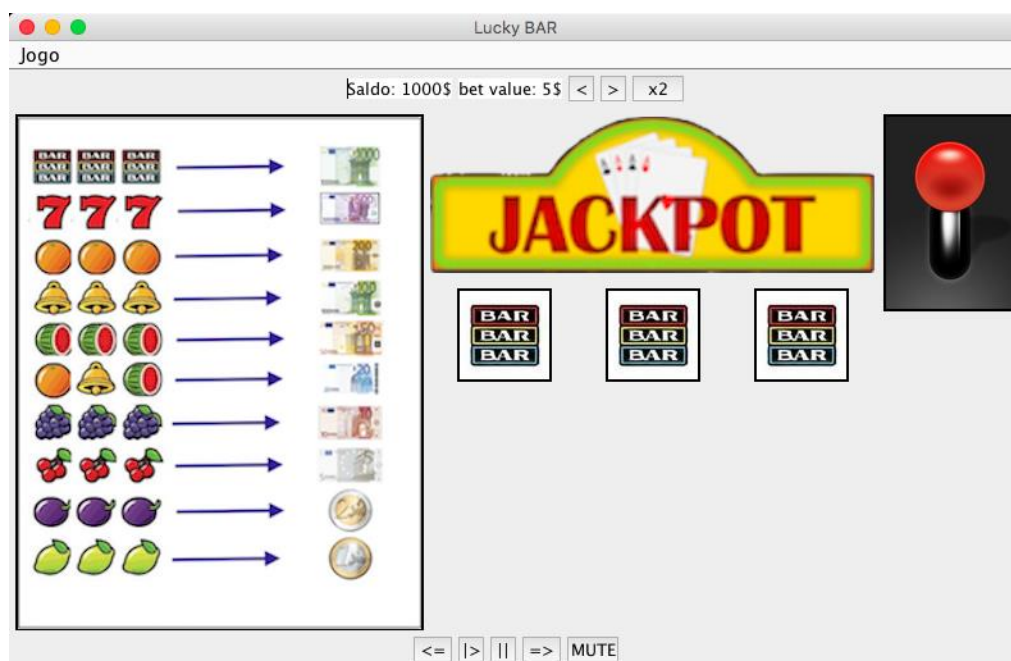


Figura 14 – Janela do jogo *Slot Machine*

3.16. SOUND

Uma pequena classe que permite abrir uma música guardada no armazenamento local.

Contém também os métodos para parar, começar e comutar o som.

Esta classe mantém uma música de fundo durante os jogos, mas também é responsável pelo som da bola da roleta.

3.17. PLAYERSONG

Esta classe é um painel que é constituído pela classe anterior “Sound” e uma série de botões que permitem o controlo da música, de modo a criar um navegador de músicas.

A figura seguinte ilustra esta classe, sendo que a mesma é reproduzida em todas as janelas desta aplicação.



Figura 15 – Classe “PlayerSong” com botões de controlo da música.

3.18. HEADER

Finalmente a classe “Header” integra a classe anterior e compõe o cabeçalho das paginas. Contém o saldo do jogador, o valor da aposta, botões para mudar o valor da aposta e o botão de apostar. Como o mesmo formato de cabeçalho foi utilizado para todos os jogos, apenas o botão de aposta difere de jogo para jogo, sendo que a resposta ao seu evento é definida em cada jogo.

Na Figura 16 podemos ver esta classe integrada num dos jogos.

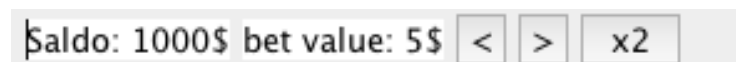


Figura 16 – Classe Header integrada num dos jogos.

4. CONCLUSÕES

Com base nos conhecimentos adquiridos ao longo da Disciplina de Sistemas de Aplicações Distribuídas em Telecomunicações foi desenvolvido este trabalho. Os três jogos desenvolvidos promoveram a aprendizagem e proficiência da linguagem JAVA.

Em conjunto com trabalho anterior, que serviu de base para este, foram desenvolvidas competências com uma linguagem orientada a objetos, entre outros destacam-se: Uso de eventos, uso de *Frames/Paineis/Botoes/TextAreas*, uso de *Threads* e o uso de *Sockets*.

Em suma, este trabalho permitiu-me desenvolver uma aplicação distribuída melhorando os meus conhecimentos nas tecnologias associadas.

Referências Documentais

- [1] “Apostas Roleta.” Accessed January 17, 2016. <http://forum.bonus-apostas.com/index.php?topic=368.0>.
- [2] “Regras Blackjack.” Accessed January 24, 2016. <http://www.metodoblackjack.pt/regras-do-blackjack/>.