

# Slides de apoio à aula prática 6

Integrar Jogo do Galo na *framework* de interação com o  
utilizador (po-uilib)

# 0. Pré-requisitos

- Descarregar *po-uilib.jar* ([link](#))
- Descarregar *jogogalo.jar* ([link](#))
- Ler documento de explicação da framework da cadeira ([link](#))
- Descompactar *jogogalo.jar* (`jar -xf jogogalo.jar`)

# 1. Ponto de entrada na aplicação

## jogogalo/textui/Main.java

```
2
3 import jogogalo.core.AplicacaoJogoGalo;
4
5 import jogogalo.textui.main.MainMenu;
6
7 import static pt.utl.ist.po.ui.UserInteraction.IO;
8
9
10 public class Main {
11
12     public static void main(String[] args) throws Exception {
13
14         AplicacaoJogoGalo app = new AplicacaoJogoGalo();
15         MainMenu menu = new MainMenu(app);
16
17         menu.open();
18         IO.close();
19     }
```

Criada instância (do tipo `AplicacaoJogoGalo`) sobre a qual os comandos vão executar operações

Primeiro menu (main) é criado e instância criada anteriormente é passada no construtor

- main menu será aberto
- Opção *Sair* é colocada automaticamente
- Que outras opções estarão disponíveis neste menu? (próximo slide)

## 2. Main Menu

jogogalo/textui/main/MainMenu.java

```
1 package jogogalo.textui.main;
2
3 import jogogalo.core.AplicacaoJogoGalo;
4
5 import pt.utl.ist.po.ui.Command;
6 import pt.utl.ist.po.ui.Menu;
7
8 /**
9  * Represents the main menu of this application. This is the first menu
10  * shown to the users.
11  */
12 public class MainMenu extends Menu {
13     /**
14      * Constructor
15      */
16     public MainMenu(AplicacaoJogoGalo app) {
17         super("Aplicação Jogo do Galo", new Command<?>[] {
18             new PlayGame(app),
19         });
20     }
21 }
```

Um menu deve herdar da classe Menu, já implementada na framework.

- Um Menu é criado recorrendo a um *array* de comandos (instâncias de sub-classes de Command).
- Para cada opção disponível no Menu deve ser criado um objecto de uma sub-classe de Command.

Neste caso, o menu é criado apenas com 1 comando, pelo que o menu terá duas opções (o *Sair* é colocado automaticamente).

Qual a funcionalidade da opção deste menu?  
(próximo slide)

# 3.1 Comando PlayGame

## jogogalo/textui/main/PlayGame.java

```
/* of this application.
 */
public class PlayGame extends Command<AplicacaoJogoGalo> {

    /**
     * Constructor.
     *
     * @param app the target entity.
     */
    public PlayGame(AplicacaoJogoGalo app) {
        super("Novo Jogo", app);
    }

    /**
     * Execute the command.
     */
    @Override
    @SuppressWarnings("nls")
    public final void execute() {
        Form f = new Form();
        InputInteger inI = new InputInteger(f, "Indique dimensão do jogo: ");
        f.parse();

        JogoGalo jogo = new JogoGalo(inI.value());

        Command<?>[] commands = {
            new HumanPlayer(jogo, true),
            new HumanPlayer(jogo, false),
        };

        Menu jogoMenu = new Menu("Jogar", commands);
        jogoMenu.entry(1).invisible();
        jogoMenu.open();

        if (jogo.getEstado() != EstadoJogo.NAO_FINALIZADO) {
            entity().addJogo(jogo);
        }
    }
}
```

Os comandos vão executar operações sobre entidades, como tal, é necessário especificar o tipo dessa entidade.

Este comando vai executar operações sobre objectos do tipo *AplicacaoJogoGalo*

Essa entidade deve ser passada ao Comando aquando a criação das suas instâncias

No construtor do comando deve especificar-se a descrição que irá figurar no menu respectivo

## 3.2 Comando PlayGame

jogogalo/textui/main/PlayGame.java

```

* of this application.
*/
public class PlayGame extends Command<AplicacaoJogoGalo> {

    /**
     * Constructor.
     *
     * @param app the target entity.
     */
    public PlayGame(AplicacaoJogoGalo app) {
        super("Novo Jogo", app);
    }

    /**
     * Execute the command.
     */
    @Override
    @SuppressWarnings("nls")
    public final void execute() {
        Form f = new Form();
        InputInteger inI = new InputInteger(f, "Indique dimensão do jogo: ");
        f.parse();

        JogoGalo jogo = new JogoGalo(inI.value());

        Command<?>[] commands = {
            new HumanPlayer(jogo, true),
            new HumanPlayer(jogo, false),
        };

        Menu jogoMenu = new Menu("Jogar", commands);
        jogoMenu.entry(1).invisible();
        jogoMenu.open();

        if (jogo.getEstado() != EstadoJogo.NAO_FINALIZADO) {
            entity().addJogo(jogo);
        }
    }
}

```

O *execute()* será o método executado quando o utilizador seleccionar a opção correspondente e este comando

Na execução do comando é possível pedir ao utilizador para introduzir dados através de instâncias de *Form* (ver documento de explicação da framework para mais detalhes sobre *Form*)

Neste comando específico não acontece, mas é também possível apresentar cadeias de caracteres ao utilizador (utilizado *Display*)

# 3.3 Comando PlayGame

## jogogalo/textui/main/PlayGame.java

```

* of this application.
*/
public class PlayGame extends Command<AplicacaoJogoGalo> {

    /**
     * Constructor.
     *
     * @param app the target entity.
     */
    public PlayGame(AplicacaoJogoGalo app) {
        super("Novo Jogo", app);
    }

    /**
     * Execute the command.
     */
    @Override
    @SuppressWarnings("nls")
    public final void execute() {
        Form f = new Form();
        InputInteger inI = new InputInteger(f, "Indique dimensão do jogo: ");
        f.parse();

        JogoGalo jogo = new JogoGalo(inI.value());

        Command<?>[] commands = {
            new HumanPlayer(jogo, true),
            new HumanPlayer(jogo, false),
        };

        Menu jogoMenu = new Menu("Jogar", commands);
        jogoMenu.entry(1).invisible();
        jogoMenu.open();

        if (jogo.getEstado() != EstadoJogo.NAO_FINALIZADO) {
            entity().addJogo(jogo);
        }
    }
}

```

Neste caso, o comando irá pedir ao utilizador para inserir uma dimensão do tabuleiro.

Em seguida irá ser criada uma instância do Jogo do Galo.

Um novo menu será criado e aberto. Desta feita, os comandos deste menu irão executar operações sobre o Jogo do Galo criado (*jogo*).

Notem que não é necessário criar uma sub-classe de menu para criar e abrir um novo menu (tal como foi feito em MainMenu.java)!

# 3.4 Comando PlayGame

## jogogalo/textui/main/PlayGame.java

```
/* of this application.
*/
public class PlayGame extends Command<AplicacaoJogoGalo> {

    /**
     * Constructor.
     *
     * @param app the target entity.
     */
    public PlayGame(AplicacaoJogoGalo app) {
        super("Novo Jogo", app);
    }

    /**
     * Execute the command.
     */
    @Override
    @SuppressWarnings("nls")
    public final void execute() {
        Form f = new Form();
        InputInteger inI = new InputInteger(f, "Indique dimensão do jogo: ");
        f.parse();

        JogoGalo jogo = new JogoGalo(inI.value());

        Command<?>[] commands = {
            new HumanPlayer(jogo, true),
            new HumanPlayer(jogo, false),
        };

        Menu jogoMenu = new Menu("Jogar", commands);
        jogoMenu.entry(1).invisible();
        jogoMenu.open();

        if (jogo.getEstado() != EstadoJogo.NAO_FINALIZADO) {
            entity().addJogo(jogo);
        }
    }
}
```

*entity()* será um dos métodos mais importantes que irão utilizar no contexto dos Comandos.

Tal método permite aceder à entidade sobre a qual os comandos estão a executar as operações.

Neste caso, *entity()* irá devolver uma instância de *AplicacaoJogoGalo* (a mesma que foi criada em *Main.java*, ver **slide 3**)



# 4.1 Adicionar um novo comando

## jogogalo/textui/main/MainMenu.java

```
MainMenu.java x
1 package jogogalo.textui.main;
2
3 import jogogalo.core.AplicacaoJogoGalo;
4
5 import pt.utl.ist.po.ui.Command;
6 import pt.utl.ist.po.ui.Menu;
7
8 /**
9  * Represents the main menu of this application. This is the first menu
10  * shown to the users.
11  */
12 public class MainMenu extends Menu {
13     /**
14      * Constructor
15      */
16     public MainMenu(AplicacaoJogoGalo app) {
17         super("Aplicação Jogo do Galo", new Command<?>[] {
18             new PlayGame(app),
19             new ViewData(app)
20         });
21     }
22 }
23
```

1. Para adicionar uma nova entrada num menu, terão que adicionar uma instância do comando respectivo, no momento da criação do menu.

O MainMenu.java do **slide 4** foi alterado nesse sentido.

## 4.2 Adicionar um novo comando

### jogogalo/textui/main/ViewData.java

```
ViewData.java
1  package jogogalo.textui.main;
2
3  import pt.utl.ist.po.ui.Menu;
4  import pt.utl.ist.po.ui.Command;
5
6  import jogogalo.core.AplicacaoJogoGalo;
7
8  import jogogalo.textui.historico.*;
9
10 /*
11  * Command for entering in the game archive. Opens a new Menu when
12  * are accessible.
13  */
14 public class ViewData extends Command<AplicacaoJogoGalo> {
15
16
17     public ViewData(AplicacaoJogoGalo app) {
18         super("Ver Histórico", app);
19     }
20
21     @Override
22     @SuppressWarnings("nls")
23     public final void execute() {
24
25         Command<?>[] commands = {
26             new MostararNumeroJogosTerminados(entity()),
27             new MostrarJogo(entity()),
28             new VitoriasJogador2(entity()),
29             new ResumoJogosTerminados(entity())
30         };
31
32         Menu historicoMenu = new Menu("Historico", commands);
33         historicoMenu.open();
34     }
35 }
```

2. Naturalmente, devem depois implementar o comando que criaram.

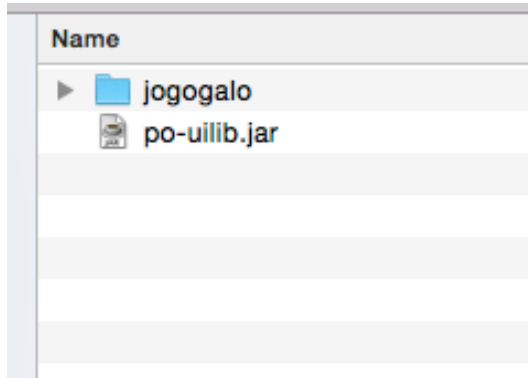
O comando (*ViewData*) irá executar operações, novamente, sobre uma instância de *AplicacaoJogoGalo*

Este comando irá ser a entrada 2 do menu (ver **slide** anterior) e irá ter o nome “Ver Histórico”

Em particular, o comando apenas cria e abre um novo menu

Os comandos desse menu vão operar sobre a mesma entidade passada ao comando *ViewData* (obtida através de *entity()*)

## 5.1 Utilização de *po-uilib.jar*



Devem colocar o .jar que contém a framework da cadeira na mesma diretoria de *jogogalo/*

A partir desta diretoria podem compilar e correr desta forma...

```
Fernandos-MacBook-Air:a fpedro$ javac -cp .:po-uilib.jar: `find jogogalo -name "*.java"`
Fernandos-MacBook-Air:a fpedro$ java -cp .:po-uilib.jar jogogalo.textui.Main
Aplicação Jogo do Galo
1 - Novo Jogo
2 - Ver Histórico
0 - Sair
Escolha uma opção: █
```

## 5.2 Utilização de *po-uilib.jar*

Pode dar-se o caso de terem uma versão do *java* diferente daquela em que os .class em po-uilib.jar foram criados.

```
warning: po-uilib.jar(pt/utl/ist/po/ui/Input.class): major version 52 is newer than 51, the highest major version supported by this compiler.  
It is recommended that the compiler be upgraded.
```

Nesse caso, podem criar um novo po-uilib.jar, desta feita com os .class compilados na vossa máquina

```
$ jar -xf po-uilib.jar  
$ javac -cp . `find pt -name "*.java"`  
$ jar -cf po-uilib.jar pt/  
$
```