# UNIVERSIDADE NOVA DE LISBOA

FCT – Departamento de Engenharia Electrotécnica

#### Secção de Robótica e Manufactura Integrada

|  |  |
| --- | --- |
| Mini-trabalho nº 3 (versão draft) | |
| Disciplina | **Modelação de dados em Engenharia** |
| Ano Letivo | 2014/2015 |
| Objetivo | Modelar informação num contexto de engenharia Eletrotécnica |
| Aulas | 3 aulas x 3 horas + 3 horas extra |
| Data de Entrega | 2015/06/07 |
| Objetivos concretos:   1. Modelar dados    1. Entender o conceito de modelação com *frames* e relevância num contexto de engenharia Eletrotécnica    2. Representação de modelos com a linguagem Golog    3. Modelar a informação do problema apresentado no anexo 1 2. Conceitos/tecnologias usadas    1. Frames    2. Slots    3. Relações e mecanismos de herança    4. Métodos    5. Demónios    6. Mecanismo modelação de frames GOLOG    7. Prolog 3. Ilustrar a utilização do modelo num contexto prático de aplicação 4. Responder à ficha de trabalho (Anexo 2), ao longo e até ao final do trabalho, tendo que ser entregue na data indicada. | |

# Anexo 1 – Descrição do trabalho a efetuar

**Objectivos:**

Este mini-trabalho destina-se a proporcionar aos alunos um contacto com os conceitos básicos de programação estruturada em Frames, tendo em vista a sua posterior utilização em atividades enquadradas no seio da Engenharia Eletrotécnica e de Computadores.

Concretamente, o estudo e a implementação do problema proposto neste trabalho servirão para fazer uma abordagem aos seguintes conceitos:

* Estudo do problema apresentado.
* Estruturação de Informação em Frames.
* Relações e mecanismos de herança.
* Utilização de Programação Lógica.
* Implementação de métodos e demónios de forma a implementar as funcionalidades pretendidas no problema.

**Parte 1 – Implementação do sistema “ar-condicionado”**

Recorrendo ao exercício relativo ao sistema de ar condicionado facultado na aula teórica, transponha o código em Prolog/Golog que implementa as respetivas *frames* para dentro dum ficheiro ar\_cond.pl. Para isso, organize as *queries* para a criação dessas *frames* dentro de predicados criar\_ac, criar\_termo, criar\_alarmes, etc.



Execute as seguintes experiencias:

1. Invocar os predicados referidos para a criação das *frames*. Executar o predicado show\_*frame* para visualização de cada uma das *frames*. Fazer scan das *frames* obtidas para dentro de slides PPT, conforme ilustrado no anexo.
2. Faça o teste do comportamento do sistema, mediante a chamada aos respetivos métodos *Golog*. Para isso, injete valores no *slot* *temp* (com o predicado *new\_value*) da *frame* termo. Observe as mudanças nos estados *das slots* das diversas *frames* e coloque-as dentro de slides.
3. Repetir o passo anterior para valores extremos de temperatura, de forma a poder-se observar as modificações de estado e criação automática de *frames* com alarmes específicos, dentro da *frames* alarme, alarm1, alarme2, … (veja o documento pdf das aulas teóricas). Fazer scans e colocar em slides.
4. Conforme reparou, o slot temp está repetido nas frames ac e termo. Apague o slot temp da frame ac. Em alternativa, crie uma nova relação com a especificação de herança inclusion([temp]) (ver aula sobre relations), que permita que a frame ac possa herdar esse slot a partir da frame termo. Considerar também a herança do slot clima de termo para ac.

Note-se que o comando delete\_frame não apaga nem os demónios nem as relations. Caso queira apagar frames, demónios e os métodos (tudo em simultâneo), utilize o predicado delete\_kb.

**Parte 2 – Exemplo sobre programação reactiva**

Esta parte vai focar-se particularmente na criação de demónios, cujo os efeitos podem ser percecionados visualmente, através de uma interface gráfica.

Imagine um cenário em que se pretende simular o comportamento de vida animal dentro dum ecossistema composto por presas e predadores. Para o caso concreto, concentre o seu esforço na modelação de gatos e de ratos. Dependendo do tipo de animal, este pode caçar, alimentar-se, movimentar-se, fugir e procriar. Os ratos alimentam-se com os queijos que forem encontrando no ecossistema. Conforme os animais se vão deslocando, caso dois deles fiquem próximos e se forem da mesma espécie, podem gerar novos animais da mesma espécie (ou seguir cada um o seu caminho). Se um deles for predador e o outro a presa, o predador alimenta-se desta, caso não consiga fugir. A figura 1 ilustra o ecossistema composto pelos referidos animais.

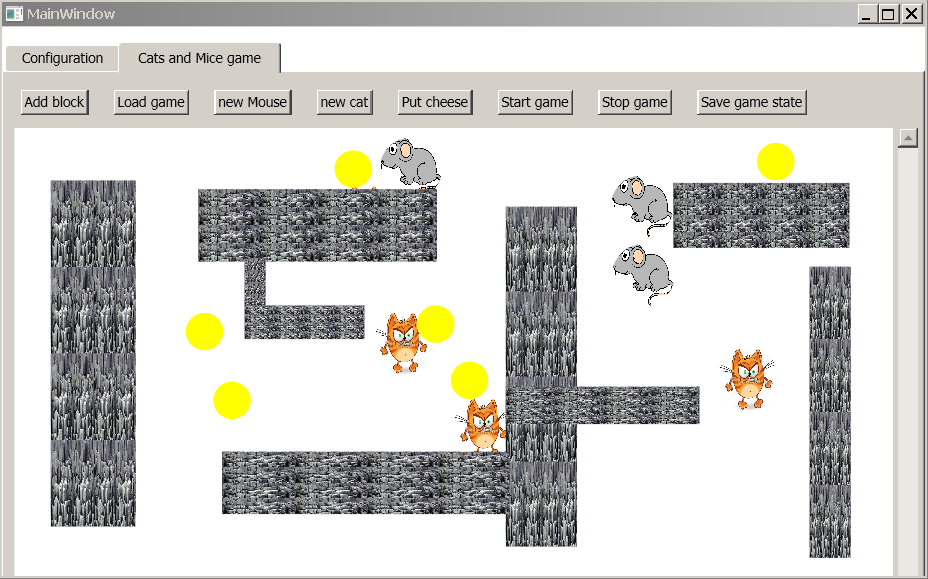


Figura 1 – Ecossistema constituído por predadores e presas.

A resposta a este problema implica a satisfação dos seguintes requisitos:

|  |  |
| --- | --- |
| Req.  Funcionais | Descrição |
| R1 | Parte 1, experiencia 1. |
| R2 | Parte 1, experiencia 2. |
| R3 | Parte 1, experiencia 3. |
| R4 | Parte 1, experiencia 4. |
|  |  |
| R5 | Parte 2, modelar o comportamento: ratos comem queijo |
| R6 | Parte 2, modelar o comportamento: predatório |
| R7 | Parte 2, modelar o comportamento de presas (sobrevivência/fugir) |
| R8 | Parte 2, modelar o comportamento de procriação de predadores e presas |
| R9 | Parte 2, modelar o comportamento de procriação de presas |
| R10 | Parte 2, evitar obstáculos |
|  |  |

Seja criativo! Por exemplo, se um rato come uma determinada quantidade de queijos, transforma-se em super-rato, conseguindo fugir mais depressa, atravessar as paredes do labirinto ou ser mais forte que os gatos.

## Elementos de Reflexão

Esta parte consiste em considerações mais reflexivas que poderão servir como “inspiração” para a escrita do relatório para este trabalho. No final do trabalho, as seguintes questões deverão poder ser respondidas:

* Quais as vantagens/desvantagens da modelação por Frames, em relação à modelação efectuadas nos precedentes trabalhos práticos.
* A utilidade da modelação em Golog no contexto da Engenharia.
* Como funcionam os demónios?

Este trabalho segue o seguinte plano:

1ª aula:

Apresentação do trabalho e das ferramentas a utilizar. Revisão dos conceitos relacionados com a programação em Golog.

Alunos: Modelação do problema em Frames. Implementação das Frames com os respectivos atributos e métodos.

2ª aula:

Docente: Resumo dos conceitos relacionados com a programação reactiva. Acompanhamento dos alunos.

Alunos: Completar o modelo em frames dos animais. Criação dos demónios para os comportamentos pretendidos.

3ª aula:

Docente: Modelação em UML. Acompanhar os alunos na finalização do trabalho.

Alunos: Finalização do trabalho. Representar as frames através de diagramas UML.

Entrega:

- todos os resultados relativos à solução do problema e implementação

- Impressão do anexo 2 preenchido (imprimir apenas este)

Docentes:

Teórica:

Luis Camarinha-Matos, [cam@uninova.pt](mailto:cam@uninova.pt)

Prática:

João Rosas, [jrosas@uninova.pt](mailto:jrosas@uninova.pt)

Tiago Cardoso, [tomfc@uninova.pt](mailto:tomfc@uninova.pt)

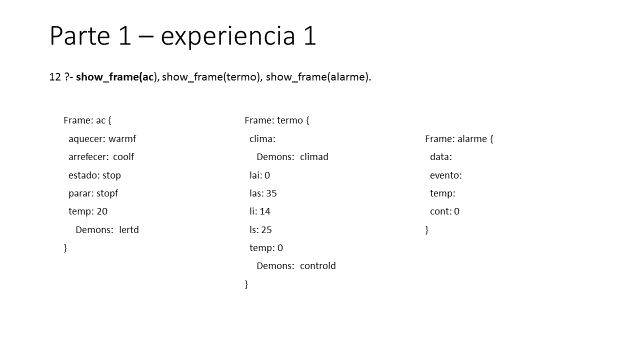
# Anexo 2 – Ficha de laboratório

|  |  |
| --- | --- |
| **Resposta à Ficha de trabalho STR nº 3** | |
| Disciplina | Modelação de Dados em Engenharia |
| Ano Lectivo | 2012/2013 |
| Aluno Nº\_\_\_\_\_  Aluno Nº\_\_\_\_\_  Aluno Nº\_\_\_\_\_ | Nome:  Nome:  Nome: |
| Data Entrega: |  |

Diagrama das Frames (aumentar área se necessário)

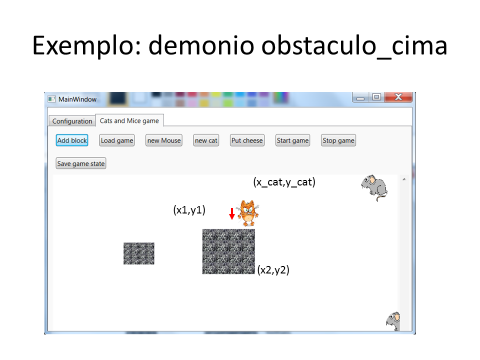
|  |
| --- |
|  |

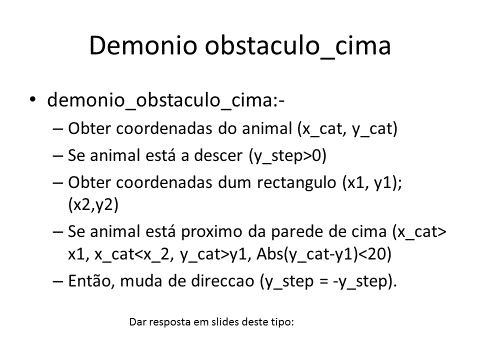
**Slides para parte 1**



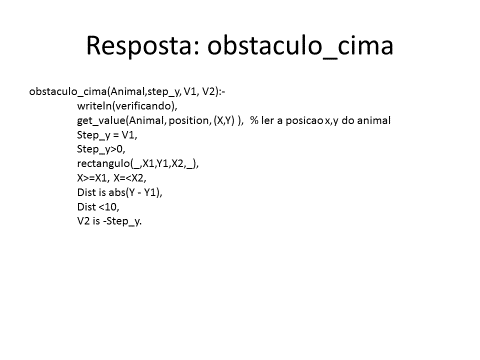
Colocar aqui os restantes slides para a parte 1

**Slides para parte 2**





(este slide vai ser substituído por uma versão melhorada: Jrosas)



Colocar aqui os restantes slides para a parte 2