

## TRABALHO PRÁTICO Nº 1

---

# Aplicação em Prolog para um Jogo de Tabuleiro

## Descrição

**Objetivo:** Implementar um jogo de tabuleiro para dois jogadores em linguagem Prolog,. Um jogo de tabuleiro caracteriza-se pelo tipo de tabuleiro e de peças, pelas regras de movimentação das peças (jogadas possíveis) e pelas condições de terminação do jogo com derrota, vitória ou empate. O jogo deve permitir três modos de utilização: Humano/Humano, Humano/Computador e Computador/Computador. Devem ser incluídos pelo menos dois níveis de jogo para o computador. Deve ser construída uma interface adequada com o utilizador, em modo de texto.

A aplicação terá um visualizador gráfico 3D, a realizar na Unidade Curricular de LAIG. Este visualizador 3D não será alvo de avaliação na Unidade Curricular de PLOG.

**Sistema de Desenvolvimento:** O sistema de desenvolvimento deve ser o SICStus Prolog 4.6, que inclui a possibilidade de criação de *sockets* para a comunicação com o módulo de visualização. O SICStus inclui ainda a biblioteca Jasper, que facilita a integração de um componente Prolog numa aplicação Java. A forma de integração do visualizador 3D com o jogo desenvolvido em Prolog deve ser discutida com os docentes da Unidade Curricular de LAIG.

## Condições de Realização

**Constituição dos Grupos:** Grupos de **2 estudantes**, inscritos na mesma turma teórico-prática, com paridade de número de estudante diferente. Excepcionalmente e apenas em caso de necessidade, podem aceitar-se trabalhos individuais. Estudantes dispensados da frequência das aulas teórico-práticas (e.g. trabalhadores-estudantes) deverão contactar o docente no sentido de marcar sessões de acompanhamento do progresso e avaliação do trabalho.

**Avaliação:** Dois momentos de avaliação: 1) **avaliação intercalar** com entrega do relatório até dia **1 de Novembro** de 2020; 2) **avaliação final** com entrega do relatório e código fonte até dia **29 de Novembro** de 2020, e com demonstrações realizadas na semana de 30 de Novembro a 4 de Dezembro de 2020.

**Pesos das Avaliações:** Ver ficha da Unidade Curricular no SIGARRA.

**Linguagens e Ferramentas:** O jogo deve ser desenvolvido em SICStus Prolog versão 4.6, e deve ser garantido o seu funcionamento em Windows e Linux. Caso seja necessária alguma configuração (para além da instalação padrão do software), ou seja usada uma fonte diferente da fonte por omissão, isso deverá estar expresso no ficheiro README que deverá ainda incluir os passos necessário para configurar e/ou instalar as componentes necessárias (em Windows e Linux). A impossibilidade de testar o código desenvolvido resultará em penalizações na avaliação. Deve ter o cuidado de nomear os predicados usados conforme pedido na descrição das avaliações intercalar e final. Todo o código deve ser devidamente comentado.

## TRABALHO PRÁTICO Nº 1

---

### Escolha do Trabalho

Os estudantes devem formar grupos de trabalho e selecionar o tema do mesmo no *Moodle*, no formulário a disponibilizar para o efeito a partir do dia **2 de Outubro** de 2020.

Haverá um limite do número de grupos que podem escolher o mesmo tema, de modo a procurar garantir que todos os temas são igualmente selecionados. No final deste enunciado encontra-se uma lista dos temas de trabalho possíveis.

### Avaliação Intercalar

Cada grupo deve entregar o código-fonte desenvolvido e um mini-relatório em formato README, que será alvo de discussão com o docente da turma respetiva. A submissão, em formato arquivo, deverá ser feita na plataforma *Moodle*, e o nome do ficheiro deverá ser:

PLOG\_TP1\_RI\_TX\_#GRUPO.ZIP

em que TX identifica a turma prática (ex. T6 para a turma 3MIEIC06), e #GRUPO é a designação do grupo. Ex: PLOG\_TP1\_RI\_T3\_Xadrez1.ZIP

O arquivo deve obrigatoriamente conter:

- Um ficheiro **README**, contendo:
  - Identificação do trabalho (jogo) e do grupo (designação do grupo, turma prática, assim como número e nome completo de cada um dos elementos);
  - Descrição do jogo: descrição sumária do jogo e suas regras (até 300 palavras); Devem incluir ainda ligações usadas na recolha de informação (página oficial do jogo, livro de regras, etc.);
  - Representação interna do estado do jogo: indicação de como representam o estado do jogo, incluindo tabuleiro (tipicamente usando uma lista de listas com diferentes átomos para as peças), jogador atual, e eventualmente peças capturadas ou ainda por jogar, ou outras informações que possam ser necessárias (dependendo do jogo). Deve incluir exemplos da representação, em Prolog, de estados de jogo inicial, intermédio e final (que deverão estar também no ficheiro de código - ver abaixo), e indicação do significado de cada átomo (ex., como representam as diferentes peças).
  - Visualização do estado de jogo: pequena descrição da implementação do predicado de visualização do estado de jogo (até 200 palavras).
- O código-fonte desenvolvido para a visualização do estado do jogo. O código deve estar devidamente comentado. O predicado principal deve ser *play/0*, e deve permitir visualizar um tabuleiro/estado de jogo quando invocado, através do predicado de visualização, que se deverá chamar *display\_game(+GameState, +Player)*, recebendo o estado de jogo atual, assim como o jogador que efetuará a próxima jogada. O predicado que devolve o estado inicial do jogo deverá denominar-se *initial(-GameState)*, mantendo-se para a entrega final.
- Uma ou mais **imagens** ilustrativas da execução do código, mostrando um estado de jogo inicial, e possíveis estados intermédio e final (estes estados de jogo devem ser codificados diretamente no ficheiro de código para esta demonstração, tal como pedido também no ficheiro README).

## TRABALHO PRÁTICO Nº 1

---

### Avaliação Final

Cada grupo deve entregar um relatório em formato README e o código-fonte desenvolvido, bem como realizar uma demonstração da aplicação. A submissão deverá ser em formato ZIP na plataforma *Moodle*, e o nome do ficheiro deverá ser:

PLOG\_TP1\_FINAL\_TX\_#GRUPO.ZIP

em que TX indica a turma prática (ex. T6 para a turma 3MIEIC06), e #GRUPO é a designação do grupo. Exemplo: PLOG\_TP1\_FINAL\_T3\_Xadrez1.ZIP

O ficheiro ZIP deverá conter um ficheiro README e o código-fonte PROLOG, o qual deverá ser devidamente comentado.

O ficheiro README deve ter a seguinte estrutura:

- **Identificação do trabalho e do grupo:** como no relatório intercalar
- **Instalação e Execução:** incluir todos os passos necessários para correta execução do jogo em ambientes Linux e Windows (para além da instalação do SICStus Prolog 4.6).
- **O Jogo XXX:** baseado no relatório intercalar.
- **Lógica do Jogo:** Descrever (não basta copiar o código fonte) o projeto e implementação da lógica do jogo em Prolog. O predicado de início de jogo deve ser *play/0*. Esta secção deve ter informação sobre os seguintes tópicos (até 2000 palavras no total):
  - o **Representação do Estado do Jogo:** baseado no relatório intercalar.
  - o **Visualização do Estado do Jogo:** baseado no relatório intercalar. Pode incluir informação sobre o sistema de menus entretanto criado, assim como interação com o utilizador, incluindo formas de controlo de entrada.
  - o **Lista de Jogadas Válidas:** Obtenção de lista com jogadas possíveis. O predicado deve chamar-se *valid\_moves(+GameState, +Player, -ListOfMoves)*.
  - o **Execução de Jogadas:** Validação e execução de uma jogada, obtendo o novo estado do jogo. O predicado deve chamar-se *move(+GameState, +Move, -NewGameState)*.
  - o **Final do Jogo:** Verificação do fim do jogo, com identificação do vencedor. O predicado deve chamar-se *game\_over(+GameState, -Winner)*.
  - o **Avaliação do Tabuleiro:** Forma(s) de avaliação do estado do jogo. O predicado deve chamar-se *value(+GameState, +Player, -Value)*.
  - o **Jogada do Computador:** Escolha da jogada a efetuar pelo computador, dependendo do nível de dificuldade. O predicado deve chamar-se *choose\_move(+GameState, +Player, +Level, -Move)*.
- **Conclusões:** Conclusões do trabalho, incluindo limitações do trabalho desenvolvido (*known issues*), assim como possíveis melhorias identificadas (*roadmap*). (até 250 palavras)
- **Bibliografia:** Listagem de livros, artigos, páginas Web e outros recursos usados durante o desenvolvimento do trabalho

## TRABALHO PRÁTICO Nº 1

---

### Problemas (Jogos) Propostos

Os jogos a implementar são jogos de tabuleiro para dois jogadores em que não existe a influência do fator sorte no decorrer do jogo. Os jogos não incluem dados nem sorteios de qualquer tipo ou informação inicialmente escondida.

Jogos propostos:

1. Alliances - [https://nestorgames.com/#alliances\\_detail](https://nestorgames.com/#alliances_detail)
2. Ampel - [https://nestorgames.com/#ampel\\_detail](https://nestorgames.com/#ampel_detail)
3. Bide - [https://nestorgames.com/#bide\\_detail](https://nestorgames.com/#bide_detail)
4. Emulsion - <https://boardgamegeek.com/boardgame/311851/emulsion>
5. Gauss - [https://nestorgames.com/#gauss\\_detail](https://nestorgames.com/#gauss_detail)
6. Gekitai - <https://boardgamegeek.com/boardgame/295449/gekitai>
7. Glaisher - [https://nestorgames.com/#glaisher\\_detail](https://nestorgames.com/#glaisher_detail)
8. Green Skull - [https://nestorgames.com/#greenskull\\_detail](https://nestorgames.com/#greenskull_detail)
9. Greener - [https://nestorgames.com/#green\\_detail](https://nestorgames.com/#green_detail)
10. Jin Li - [https://nestorgames.com/#jinli\\_detail](https://nestorgames.com/#jinli_detail)
11. Mapello - [https://nestorgames.com/#mapello\\_detail](https://nestorgames.com/#mapello_detail)
12. Mitsudomoe - [https://nestorgames.com/#mitsudomoe\\_detail](https://nestorgames.com/#mitsudomoe_detail)
13. Nava - <https://boardgamegeek.com/boardgame/250491/nava>
14. SQU - [https://nestorgames.com/#squ\\_detail](https://nestorgames.com/#squ_detail)
15. Swack - <https://boardgamegeek.com/boardgame/314462/swack>
16. Taiji - [https://nestorgames.com/#taiji\\_detail](https://nestorgames.com/#taiji_detail)
17. Talpa - [https://nestorgames.com/#talpa\\_detail](https://nestorgames.com/#talpa_detail)
18. Three Dragons - <https://boardgamegeek.com/boardgame/306972/three-dragons>
19. Tumbleweed - <https://boardgamegeek.com/boardgame/318702/tumbleweed>