***Trippples***

**Relatório Final**



Mestrado Integrado em Engenharia Informática e Computação

Programação em Lógica

**Grupo 4:**

José João Pereira Oliveira – up201406208

Manuel António Gonçalves Gomes – up201402679

Faculdade de Engenharia da Universidade do Porto

Rua Roberto Frias, sn, 4200-465 Porto, Portugal

Porto, 13 de Novembro de 2016

Resumo

No âmbito da unidade curricular de Programação em Lógica desenvolveu-se uma solução de *software* capaz de simular o jogo de tabuleiro *Trippples*, recorrendo-se à linguagem *Prolog*.

A linguagem Prolog enquadra-se nos paradigmas de linguagens Lógicas e Declarativas, paradigmas estes que divergem de forma substancial do paradigma de Orientação a Objetos. Uma vez que este esteve presente na maioria dos projetos por nós desenvolvidos até ao momento, este projeto permitiu assim um contato com um diferente tipo de linguagem e de procedimentos de implementação de código.

O jogo *Trippples* consiste no conceito de cada jogador tentar movimentar-se entre a sua peça inicial e final do tabuleiro, vencendo o jogador que mais rapidamente atinja o objetivo. No entanto, cada jogador apenas se pode movimentar nas direções da peça sobre a qual se encontra o marcador do adversário. Por outro lado, uma situação de empate também é possível, quando ambos os jogadores ficam sem jogadas válidas possíveis.

Em relação à implementação da solução de *software* pretendida, desenvolveu-se um programa com interface em modo de texto com o auxílio do software SICTus Prolog.

Consideramos que todos os objetivos para este projeto foram atingidos com sucesso, tendo sido uma mais valia para o nosso desenvolvimento de pensamento lógico e capacidades técnicas.

**Palavras-chave**: *Prolog*, *Trippples*, Lógica.

**Índice**

[**1. Introdução** 4](#_Toc466850869)

[**2. O Jogo Trippples** 5](#_Toc466850870)

[2.1 História do Jogo 5](#_Toc466850871)

[2.1 Regras 5](#_Toc466850872)

[2.1.1 Objetivo do Jogo 5](#_Toc466850873)

[2.1.2 Preparação do Tabuleiro 6](#_Toc466850874)

[2.1.3 Movimentação no Tabuleiro 6](#_Toc466850875)

[2.1.3 Outras regras do Jogo 7](#_Toc466850876)

[**3. Lógica do Jogo** 8](#_Toc466850877)

[3.1 Representação do Estado do Jogo 8](#_Toc466850878)

[3.2 Visualização do Tabuleiro 10](#_Toc466850879)

[3.3 Lista de Jogadas Válidas 11](#_Toc466850880)

[3.4 Execução de Jogadas 11](#_Toc466850881)

[3.5 Avaliação do Tabuleiro e Jogadores 12](#_Toc466850882)

[3.6 Final do Jogo 12](#_Toc466850883)

[3.7 Jogada do Computador 12](#_Toc466850884)

[**4. Interface com o Utilizador** 14](#_Toc466850885)

[**5. Conclusões** 18](#_Toc466850886)

[**Bibliografia** 19](#_Toc466850887)

[**Anexos** 19](#_Toc466850888)

# **1. Introdução**

No âmbito da unidade curricular de Programação em Lógica pretende-se desenvolver um programa capaz de simular o jogo de tabuleiro *Trippples*, recorrendo-se à linguagem Prolog.

O jogo *Trippples* consiste no conceito de cada jogador tentar movimentar-se entre a sua peça inicial e final, vencendo o jogador que mais rapidamente atinja o objetivo. No entanto, cada jogador apenas se pode movimentar nas direções da peça sobre a qual se encontra o marcador do adversário. Este é aliás o aspeto que mais acrescenta complexidade ao jogo, pelo que se pretende-se implementar todas as regras associadas ao jogo, assim como as condições de vitória e de empate, salvaguardando-se no entanto que a interface do jogo seja simples e intuitiva para o utilizador, para uma melhor experiência de jogo.

De modo a melhor se expor o âmbito assim como a implementação prática do tema de projeto em epígrafe, estruturou-se o relatório da seguinte forma:

* História e regras do jogo;
* Lógica do jogo;
* Interface;
* Conclusões.

# **2. O Jogo Trippples**

## 2.1 História do Jogo

O jogo *Trippples* consiste num jogo de tabuleiro criado na década de 1970 idealizado para 2 jogadores, havendo no entanto uma versão para jogadores. O tabuleiro é composto por 64 peças removíveis, sendo 56 peças direcionais (3 direções), 2 peças de chegada e 2 de partida, 4 peças de zona neutra (brancas), assim como marcadores transparentes que mostram a posição de cada jogador no tabuleiro. As peças formam a superfície do tabuleiro podendo ser rearranjadas de inúmeras combinações possíveis.



Figura 1:Tabuleiro do jogo Trippples1

O objetivo do jogo passa por cada jogador se tentar mover (jogadas efetuadas por turnos), através de um marcador, desde a sua peça de partida até à peça de chegada correspondente.

## 2.1 Regras

### 2.1.1 Objetivo do Jogo

O objetivo do jogo passa por cada um dos jogadores movimentar o seu marcador de posição desde a peça de partida até à sua peça de chegada respetiva (movimentação diagonal ao longo do tabuleiro). Deve fazê-lo movimentando o marcador na mesma direção de uma das 3 setas da peça sobre a qual se encontra o marcador do adversário. Uma das principais características do jogo é que para se conseguir movimentar para uma qualquer peça em particular, deve-se tentar levar o adversário a ir para uma determinada peça que lhe permita mover-se para a peça pretendida.

### 2.1.2 Preparação do Tabuleiro

Para se preparar um tabuleiro para se iniciar um jogo começam-se por se colocar no tabuleiro as 2 peças de partida (quadrado e círculo preenchidos) e as 2 peças de chegada (quadrado e círculo não preenchidos). As 2 peças de partida são colocadas em cantos opostos, mas no mesmo lado do tabuleiro. Por sua vez, as 2 peças de chegada são colocadas no lado oposto do tabuleiro diagonalmente opostas à peça de partida com a mesma forma geométrica.

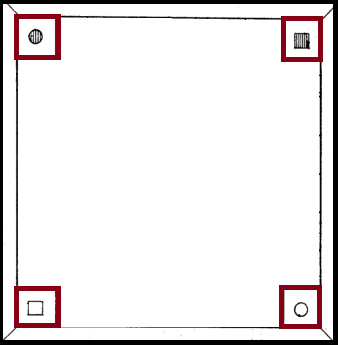
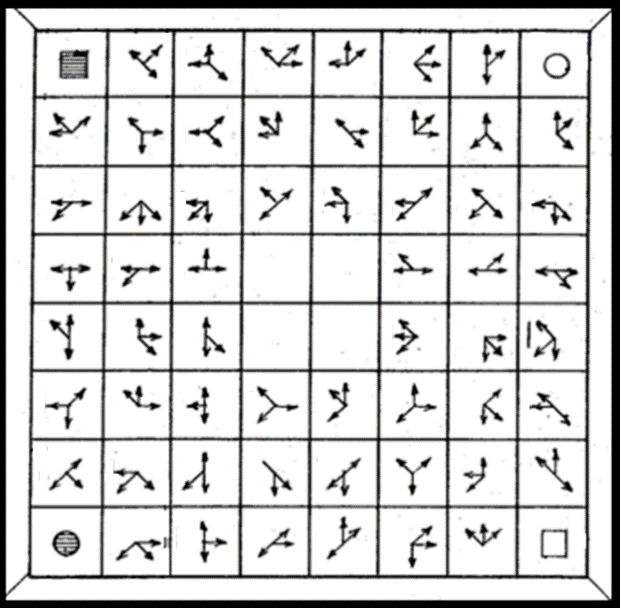
 De seguida, e de forma arbitrária, um jogador é escolhido para selecionar uma peça direcional e coloca-la no tabuleiro. Seguidamente, o jogador oponente também escolhe uma peça direcional e coloca-a no tabuleiro, e assim sucessivamente até que todas as peças de jogo estejam posicionadas no tabuleiro. As peças podem ser colocadas de forma aleatória, ou estrategicamente, à medida que o nível de jogo aumenta.

Figura 3: Tabuleiro num estado totalmente preenchido com as peças de jogo

Figura 2: Tabuleiro no estado inicial com as peças de chegada e de partida posicionados no tabuleiro (figura adaptada)

### 2.1.3 Movimentação no Tabuleiro

Quando o tabuleiro já se encontra totalmente preenchido é efetuada a escolha de forma arbitrária da peça de partida para cada jogador (quadrado ou círculo preenchidos). De seguida cada jogador coloca o seu marcador transparente de posição sobre a sua peça de partida respetiva.

Por razões históricas do jogo, o jogador de peça de partida quadrado preenchido (jogador A) é quem faz a primeira jogada do jogo, fazendo mover o seu marcador horizontalmente, verticalmente ou diagonalmente para uma qualquer peça adjacente à sua peça de partida. De seguida, o seu oponente (jogador B) movimenta o seu marcador na mesma direção de uma das 3 setas da peça sobre a qual se encontra o marcador do jogador A. A partir daqui, cada jogador no seu turno, faz movimentar o seu marcador consoante as opções dadas peças peça direcional em que se encontra o seu adversário até chegar à sua peça de chegada.

### 2.1.3 Outras regras do Jogo

Para além da regra principal de jogo, existem também as seguintes regras:

* As peças de jogo não podem ser movimentadas para uma peça ocupada ou de zona neutra; no entanto quando uma jogada legal é possível, o jogador é obrigado a fazê-la;
* Se um jogador deixa o seu adversário sem movimentação possível, o adversário perde o seu turno.
* Uma partida termina num empate quando ambos os jogadores estão com os seus marcadores paralisados.

# **3. Lógica do Jogo**

## 3.1 Representação do Estado do Jogo

O tabuleiro de jogo é representado por uma lista de listas 8x8 na seguinte forma:

**board([[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

**[T, T, T, T, T, T, T, T],**

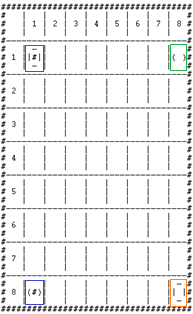
**[T, T, T, T, T, T, T, T]]).**

Cada célula de cada linha corresponde a uma posição no tabuleiro, sendo ***T*** o número da peça do tabuleiro naquela posição.

Para além do tabuleiro são também utilizadas 2 estruturas auxiliares, uma para cada jogador, nas quais são guardadas as posições atual e final de cada jogador:

* ***player(+Jogador, -Linha, -Coluna, +Linha Posição Final, -Coluna PosiçãoFinal).***

Tal como já foi referido, o jogo é composto por 64 peças diferentes, sendo que no estado inicial apenas se encontram posicionadas no tabuleiro as 2 peças de partida e as 2 peças de chegada.

 O tabuleiro inicial é declarado da seguinte forma:

**startingBoard([2, 0, 0, 0, 0, 0, 0, 3],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[0, 0, 0, 0, 0, 0, 0, 0],**

**[4, 0, 0, 0, 0, 0, 0, 1]]).**

Figura 2: Estado inicial do tabuleiro

Cada um dos valores de ***P*** tem o seguinte significado:

* 0 se espaço vazio;
* *1* se quadrado não preenchido (peça de chegada);
* *2* se quadrado preenchido (peça de partida);
* *3* se círculo não preenchido (peça de chegada);
* *4* se círculo preenchido (peça de partida).

O tabuleiro nesta fase encontra-se por isso ainda sem peças colocadas. É então efetuado o preenchimento do tabuleiro de forma aleatória com recurso ao predicado ***generateRandomBoard(+Modo de Jogo, +Tabuleiro Vazio, +Peça a colocar, +Tabuleiro preenchido)***.

Após a execução de algumas jogadas efetuadas por cada um dos jogadores, um dos possíveis estados intermédios de jogo seria:

**board([[2, 5, 6, 7, 8, 9, 10, 3],**

**[11, 12, 13, 14, 15, 16, 17, 18],**

**[19, 20, 21, 22, 23, 24, 25, 26],**

**[27, 28, 29, 0, 30, 44, 31, 32],**

**[33, 34, 35, 0, 0, 36, 37, 38,],**

**[39, 40, 41, 42, 43, 0, 45, 46],**

**[47, 48, 49, 50, 51, 52, 53, 54],**

**[4, 55, 56, 57, 58, 59, 60, 1]).**

***player(1, 3, 3, 8, 8).***

***player(2, 7, 3, 1, 8).***

Verifica-se que neste estado de jogo, o jogador 1 se encontra na posição de linha 3 e coluna 2, ao passo que o jogador 2 se encontra na posição de linha 7 e coluna 2.

Por fim, um possível estado final de jogo seria representado pelas seguintes estruturas:

***player(1, 8, 8, 8, 8).***

***player(2, 4, 6, 1, 8).***

Tal como se verifica, enquanto como o jogador se encontra na linha 4 e coluna 6, o jogador 1 encontra-se na linha 8 e coluna 8, pelo que se encontra na sua posição de chegada, e logo venceu a partida de *Trippples*.

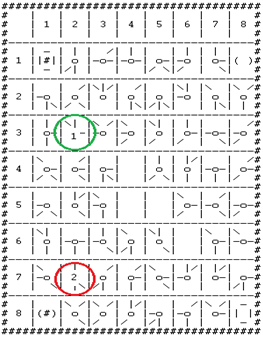
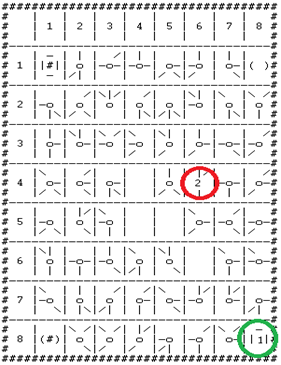


Figura 6: Estado final do jogo

Figura 3: Estado intermédio do jogo

## 3.2 Visualização do Tabuleiro

Para a visualização do tabuleiro foi utilizado o predicado ***displayBoard(+Tabuleiro, +Linha, +Linha do Tabuleiro, +Linha Jogador 1, +Coluna Jogador 1, +Linha jogador 2, +Coluna do Jogador 2)***. Este predicado recebe o tabuleiro a imprimir, assim como as coordenadas de cada jogador. Por outro lado, este predicado chama si próprio de forma recursiva, imprimindo uma linha do tabuleiro de cada vez.

Assim, em cada iteração deste predicado são executados os seguintes predicados:

* ***displayTop(+Célula do tabuleiro)***: imprime a parte de cima da peça do jogo;
* ***displayMid([+Célula do tabuleiro, +Linha, +Coluna, +Linha Jogador 1, +Coluna Jogador 1, +Linha jogador 2, +Coluna do Jogador 2)***: imprime a parte central da peça do tabuleiro;
* ***displayBottom(+Célula do tabuleiro)***: imprime a parte de baixo da peça do jogo;

Note-se que foi necessário usar 3 predicados diferentes para imprimir cada peça do tabuleiro, uma vez que cada peça corresponde a um bloco 3x3, sendo que o elemento central de cada peça pode ou não estar preenchido com o número de cada jogador, caso este esteja presente na mesma.

Para a representação de cada peça é usada o seguinte tipo de estrutura:

***tile(+Número da Peça, -Topo Esquerdo, -Topo Centro, -Topo Direito, -Meio Esquerdo, -Meio Centro, -Meio Direito, -Baixo Esquerdo, -Baixo Centro, -Baixo Direito).***

## 3.3 Lista de Jogadas Válidas

Para avaliação das jogadas válidas disponíveis, são utilizados os seguintes predicados:

* ***checkMove(+Tabuleiro, +Jogador1, +Jogador2, +Linha, +Coluna, +Modo de Jogo)***: este predicado verifica se as coordenadas Linha e Coluna escolhidas pelo jogador 1 estão dentro dos limites do tabuleiro, assim como não coincidem com as coordenadas de uma peça neutra ou de partida de um jogador; No caso da jogada ser inválida são pedidas ao utilizador novas coordenadas;
* ***validMove(+Tabuleiro, +Jogador1, +Jogador2, +Linha, +Coluna, +Modo de Jogo)***: este predicado é executado apenas em caso de sucesso do predicado ***checkMove(+Tabuleiro, +Jogador1, +Jogador2, +Linha, +Coluna, +Modo de Jogo)*** e verifica se as coordenadas escolhidas pelo jogador correspondem a uma direção da peça no qual o jogador 2 se encontra; em caso de sucesso, o movimento do jogador 1 é executado, caso contrário uma nova jogada é pedida ao jogador 1.

## 3.4 Execução de Jogadas

Para a execução de jogadas são usados os seguintes predicados na seguinte ordem:

* ***updateGame(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)***: atualiza o estado do jogo, dando começo a um novo pedido de jogada;
* ***movePlayer(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)***: este predicado irá executar um dos seguintes predicados consoante o modo de jogo escolhido:
  + ***playerInput(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)***: o qual é utilizado no caso do jogador 1 ser um humano e que pede as coordenadas da jogada pretendida;
  + ***generateCompPlay(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)***: o qual é gerado no caso do jogador 1 ser o computador, e que gera uma jogada consoante o modo de jogo escolhido
* No caso da jogada escolhida ser válida são executados os seguintes predicados:
  + ***updatePlayer(+Jogador, +Linha, +Coluna)***: o qual atualiza a posição do jogador, por meio dos predicados *builtin* *retract* e *assert*;
  + ***updateBoard(+Tabuleiro)***: predicado que imprime o novo estado do tabuleiro do jogo.

Note-se que em cada turno é também avaliado o final do jogo com predicados próprios que serão referidos no tópico ***3.6 Final do Jogo***.

## 3.5 Avaliação do Tabuleiro e Jogadores

Como já foi referido, a partir do momento que o tabuleiro de jogo é gerado, este não mais é alterado. De modo a se poder efetuar uma avaliação do jogo é utilizada a seguinte estrutura:

* ***player(+Jogador, -Linha, -Coluna, -Linha da Posição Final, -Coluna da posição Final)***: através desta estrutura é possível sempre saber-se em que posição se encontra cada jogador de modo simples e eficaz. Por outro lado, permite também avaliar se o jogador conseguiu ganhar o jogo, na seguinte situação:
  + ***Linha igual Linha da Posição Final e Coluna igual à Coluna da Posição Final.***

## 3.6 Final do Jogo

O final de uma partida de Tripples pode acontecer decorrente de uma situação de vitória de um dos jogadores, por algum deles ter chegado à sua peça de chegada, assim como de um empate. Pra avaliação do final do jogo são utilizados os seguinte predicados:

* ***isGameTied(+Tabuleiro, +Jogador1, +Jogador2)***: este predicado é verdadeiro quando nenhum dos jogadores possui qualquer jogada válida possível para o tabuleiro em jogo, estando assim perante uma situação de empate;
* ***gameOver(-Winner)***: este predicado, quando verdadeiro, retorna o vencedor do jogo (***Winner***), no caso do jogador Winner estar posicionado na sua peça de chegada.

## 3.7 Jogada do Computador

Para o modo de jogo com computador, foi desenvolvido o predicado ***generateCompPlay(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)***, o qual gera uma jogada por parte do Computador, baseada no nível de jogo do computador escolhido. Foram desenvolvidos 2 níveis de jogo do computador, *Begginer* e Advanced, procurando-se com isso implementar algum desafio para o utilizador Humano, quando o modo de jogo escolhido é o Humano vs Computador.

Como tal, foram desenvolvidos os seguinte predicados para cada nível de jogo:

* **Modo Begginer - *generateCompRandomPlay(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo):*** este predicado gera uma jogada aleatória para o coputador;
* **Modo Advanced *- generateCompAIPlay(+Tabuleiro, +Jogador1, +Jogador2, +Modo de Jogo)*:** este predicado gera uma nova jogada para o computador, sendo o modo como a mesma é obtida, baseada num algoritmo ganancioso. Assim, e perante a posição do computador no tabuleiro em relação à sua peça de chegada, o computador irá sempre escolher para cada turno, a direção possível dentro das jogadas possível, a que mais lhe permite aproximar-se da sua peça final. Como exemplo, caso o computador se encontra com coordena de linha igual à posição final, e como valor de coluna inferior à posição final, o computador irá optar, caso seja possível, movimentar-se para a direita.

# **4. Interface com o Utilizador**

Esta solução de *software* foi desenhada a pensar numa fácil interação por parte do utilizador, pelo que foram desenvolvidos vários menus para uma melhor experiência de utilização.

Assim que se executa a aplicação com o predicado ***run.*** é apresentada o seguinte menu abaixo:

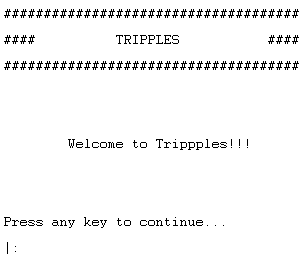


Figura 7: Ecrã inicial do Trippples

Premindo-se uma qualquer tecla, o jogo é iniciado e é apresentado o menu inicial:

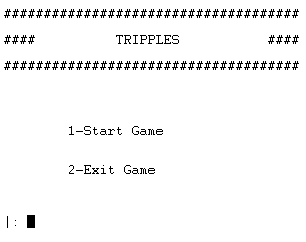


Figura 8: Menu inicial do jogo Trippples

Neste, caso o utilizador opte pela opção 1 irá inicial um novo jogo, enquanto pela opção 2 irá sair do jogo. Foram criados para além deste menu, vários outros submenus de forma a possibilitar ao utilizador a escolha do modo de jogo pretendido.

Assim, escolhendo-se a opção 1, é apresentado o seguinte menu:

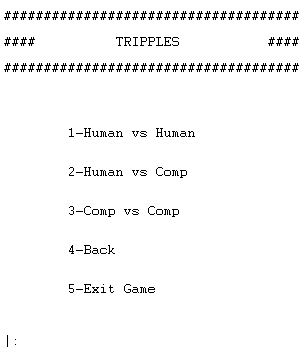


Figura 9: Menu de escolha do modo de jogo

A funcionalidade de cada opção é a seguinte:

1. Inicia um jogo humano vs humano;
2. Inicia um jogo humano vs computador;
3. Inicia um jogo computador vs computador;
4. Retorna ao menu anterior.

Note-se que no caso do modo humano vs computador é mostrado um novo submenu onde o utilizador pode escolher o nível do computador, *Begginer* ou *Advanced*.

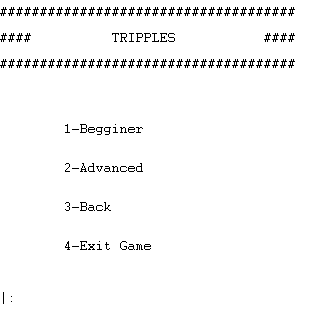


Figura 10: Menu de escolho do nível do computador

Assim que é iniciado um jogo de *Trippples* é mostrado em cada turno o estado do tabuleiro aos jogadores, sendo pedidas as coordenadas da jogada que o utilizador do turno em questão pretende executar. No caso de um jogador não ter qualquer jogada válida possível, é também mostrado uma mensagem de alerta, perdendo este jogador o seu turno.

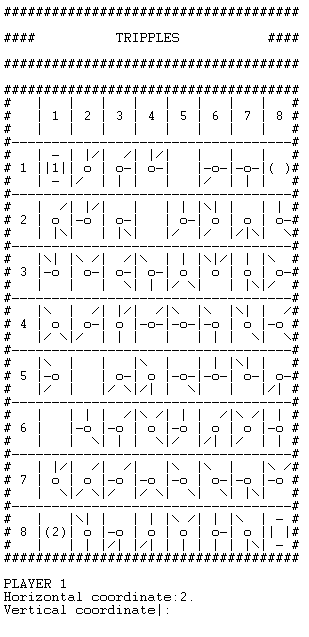


Figura 11: Turno do jogador 1

Como já foi referido, o jogo pode acabar com uma vitória de um dos jogadores, ou com um empate no caso de ambos os jogadores não terem qualquer jogada válida possível, sendo os seguintes ecrãs mostrados:

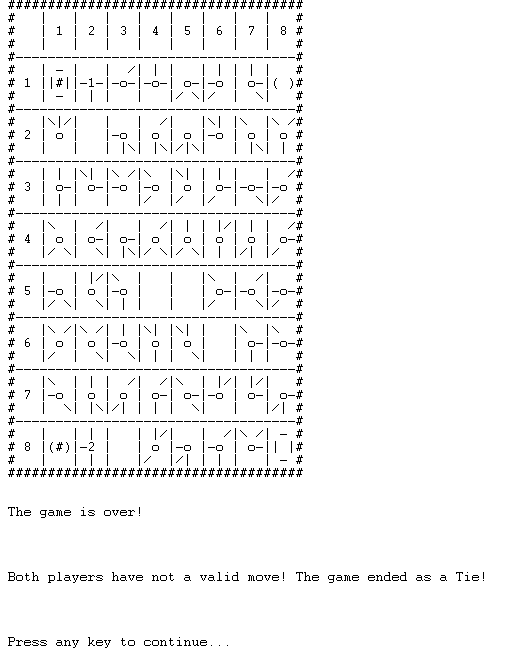
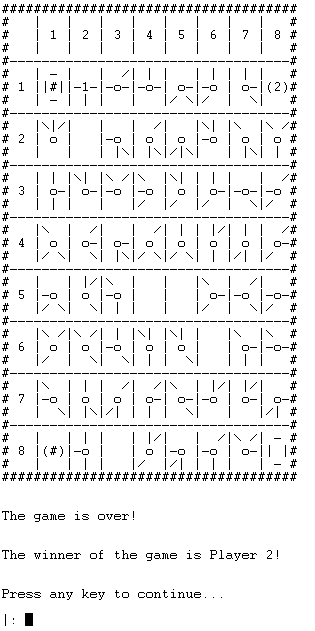


Figura 13: Ecrã do jogo com mensagem de vitória do jogador 2

Figura 12: Ecrã do jogo com mensagem de empate

Note-se que em ambas situações, e para que os utilizadores possam iniciar rapidamente uma nova partida de *Trippples*, assim que seja premida qualquer tecla, o ecrã inicial de jogo é novamente mostrado, tendo por isso o utilizador de a opção de sair do jogo ou de iniciar um novo jogo quer no modo em que jogou o anterior, quer num novo modo de jogo.

# **5. Conclusões**

A elaboração deste projeto em linguagem *Prolog* permitiu melhor compreender a metodologia de desenvolvimento de *software* baseada nos paradigmas de linguagens Lógicas e Declarativas. Uma vez que estes paradigmas divergem de certa forma do paradigma de Orientação a Objetos, paradigma este com o qual temos trabalhado na maioria dos projetos até à data, concluímos assim que este projeto permitiu expandir o nosso conhecimento, assim como enriquecer o nosso vocabulário de programação.

Por outro lado, consideramos que o projeto foi concluído com sucesso, uma vez que os objetivos propostos para o mesmo foram alcançados integralmente. No entanto, e uma vez que o *SICStus Prolog* apenas nos permitiu trabalhar a interface com o utilizador em modo de texto, não contribuindo por isso para uma experiência tão agradável para o utilizador, sentimos que este seria o aspeto a ser melhorado, projetando-se por isso como objetivo a implementação de um ambiente gráfico.

# **Bibliografia**

1. Leon Sterling, Ehud Shapiro; “The Art of Prolog 2ª Edition”; ISBN: 978-026193382;
2. Ralpharama, Trippples, <http://www.ralpharama.co.uk/item.php?itemid=571>, 3 de Novembro;
3. Board Game Geek, Trippples, <https://boardgamegeek.com/image/1966310/trippples>, 5 de Novembro.

# **Anexos**

Todos os anexos se encontram numa pasta ./Trippples, estando o código estruturado nos seguintes ficheiros:

* trippples.pl;
* Interface.pl;
* Tiles.pl;
* Utilities.pl.