

9310

# Implementação em Prolog do Jogo Oware

Relatório Intercalar



Universidade do Porto

---

Faculdade de Engenharia

**FEUP**

Mestrado Integrado em Engenharia Informática e  
Computação

Programação em Lógica

**Grupo 35:**

André Freitas - ei10036

Rui Gonçalves - ei10100

Faculdade de Engenharia da Universidade do Porto  
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

6 de Outubro de 2012

## Resumo

Este relatório tem o objetivo de descrever a fase de inicial de análise da implementação do Jogo Oware numa linguagem de programação lógica em matemática que é o Prolog. O jogo em questão é de tabuleiro, jogando-se com sementes, sendo muito popular na República do Gana. Apesar da simplicidade das regras tem um forte componente estratégico.

## 1 Introdução

Pretende-se explorar as capacidades do Prolog para representar um jogo e as suas regras através deste trabalho. Um jogo é uma excelente maneira de enriquecer o conhecimento na representação e estruturação dos dados bem como a aplicação de regras do jogo traduzidas nesta linguagem.

A implementação será em modo de texto, não sendo muito refinada no sentido da apresentação mas sim na componente de funcionalidades.

Neste documento pretende-se apresentar a descrição do problema, a representação dos estados do jogo em estruturas conhecidas em Prolog, a representação das jogadas, a visualização do tabuleiro e os aspetos de desenvolvimento do projeto.

## 2 Descrição do Problema

O Oware é dos jogos de tabuleiro mais antigos do mundo, tendo sido inventado há mais de 7 mil anos. É jogado por todo o Globo e não existem certezas relativamente à sua origem, porém, atribui-se a sua autoria tradicionalmente ao continente Africano. Atualmente é o jogo mais popular na República do Gana sendo um fenómeno nacional.



Figura 1: Pessoas jogando tradicionalmente o Oware

Como se pode constatar pela Figura 1, existe um tabuleiro com  $2 \times 6$  cavidades onde se colocam sementes ou feijões. Existem muitas interpretações das regras deste jogo, pelo que iremos adotar apenas a que é mais conhecida. Assim, o jogo começa com 4 sementes em cada buraco. Os jogadores jogam alternadamente, e em cada jogada tira-se as sementes de um buraco da nossa linha de jogo e vai-se distribuindo as sementes no sentido anti-horário.



Figura 2: Distribuição das sementes

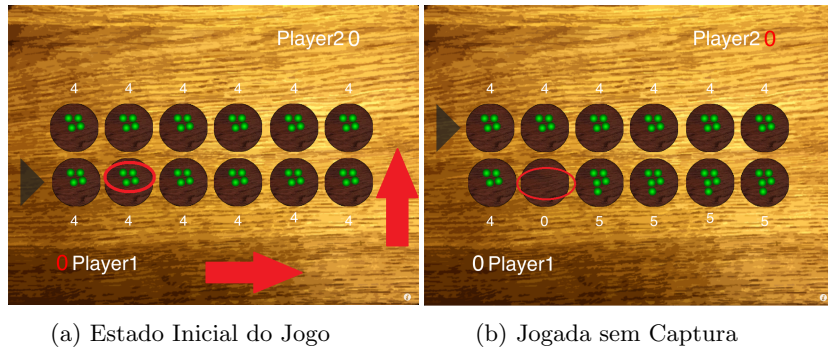
Quando distribuirmos a última semente e se colocarmos num buraco do adversário e esse sítio ficar com 2 ou 3 sementes, capturamos essas sementes. O jogo termina quando um jogador capturar 25 sementes ou ambos os dois

jogadores capturarem 24 sementes (empate).

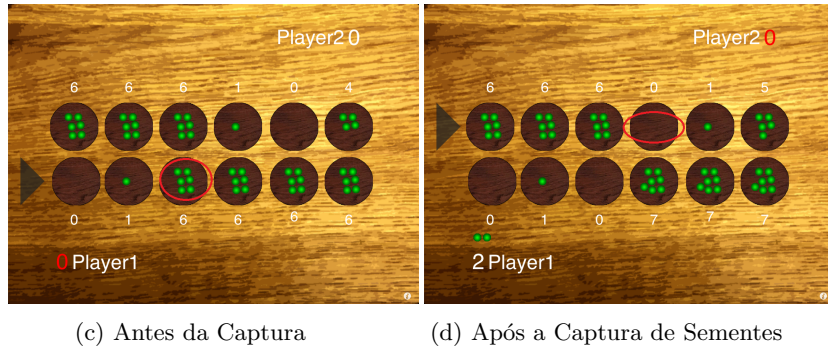
## 2.1 Ilustração de jogadas

Nas seguintes ilustrações serão descritas duas situações de jogos fundamentais. Pede-se especial atenção para as ilustrações, dado que é necessário estar atento às situações que descrevem para perceber o jogo.

Nesta primeira situação o jogo está a começar e o jogador vai distribuir a segunda cavidade de sementes. Assim, as sementes ficaram distribuídas e não houve captura. De notar novamente que o sentido da distribuição das sementes é anti-horário e que o jogador só pode retirar as sementes da sua linha.



A outra jogada importante é a da captura de sementes, ou seja, quando um jogador ao distribuir a última semente calha numa linha do seu adversário, ficam 2 ou 3 sementes nesse sítio, capturando-as. Atenção que se a meio da distribuição das sementes conseguir este número das mesmas nas cavidades, não as pode capturar, por isso é só na semente final. Destaca-se esta situação dado que existem várias interpretações relativas a esta regra.



Pode-se constatar assim a simplicidade do jogo, mas apesar disso requer um componente estratégico para que se possa ganhar, tentando prever-se uma panóplia de jogadas possíveis para vencer o adversário e maximizar as nossas capturas.

### 3 Representação do Estado do Jogo

O tabuleiro é representado por uma lista constituída por duas listas, sendo a primeira a linha do primeiro jogador e a segunda do segundo jogador. De destacar que o tabuleiro é de tamanho constante, apenas variando o valor das sementes em cada posição. O jogo começa com 4 sementes, logo a sua representação no início é:

`[[4,4,4,4,4,4],[4,4,4,4,4,4]]`

Se o jogador retirar as sementes da posição 0, ao distribuí-las no sentido anti-horário o tabuleiro fica da seguinte maneira:

`[[0,4,4,4,4,4],[5,5,5,5,4,4]]`

No caso de um dos jogadores atingir uma pontuação igual ou superior a 25 esse jogador ganha o jogo pois capturou mais de metade das sementes em jogo, Por exemplo:

`[[0,5,0,3,0,0],[0,0,0,4,8,4]]`

Num jogo de Oware também é possível, mas pouco provável, empatar pois se os dois jogadores atingirem os 24 pontos todas as sementes em jogo foram capturadas.

`[[0,0,0,0,0,0],[0,0,0,0,0,0]]`

## 4 Representação de um Movimento

Tal como constatado anteriormente, o jogo apenas tem uma jogada, que é a distribuição de sementes. Se dessa distribuição resultar na última posição 2 ou 3 sementes é feita uma captura. Assim sendo, existe um predicado para distribuir sementes, que por sua vez chama outro predicado que calcula a sequência de posição para distribuir, alterando depois o tabuleiro e verificando se existe possibilidade de captura através de um outro predicado. Após isto, se tiver existido uma captura, os pontos(sementes) do jogador é atualizado.

```
playSeeds(Board,PlayerNum,I,NewBoard)
computeSequence(PlayerNum,I,Seeds,Sequence)
removeSeeds(PlayerNum,[H|[Th|Tt]],I,Seeds,[Hnew|Tnew])
addSeeds(PlayerNum,[H|[Th|Tt]],I,Seeds,[Hnew|Tnew])
evaluateCapture(Board,WhoPlayed,I)
addScore(Score,Val,NewScore)
```

## 5 Visualização do Tabuleiro

Para a visualização do tabuleiro são usados três predicados: um principal para representar a totalidade do tabuleiro, um para apresentar o logótipo do Jogo e outro para imprimir uma linha de sementes.

```
printBoard([H|[Th|_]],P1Score,P2Score)
printLogo
printBoardLine([H|T])
```



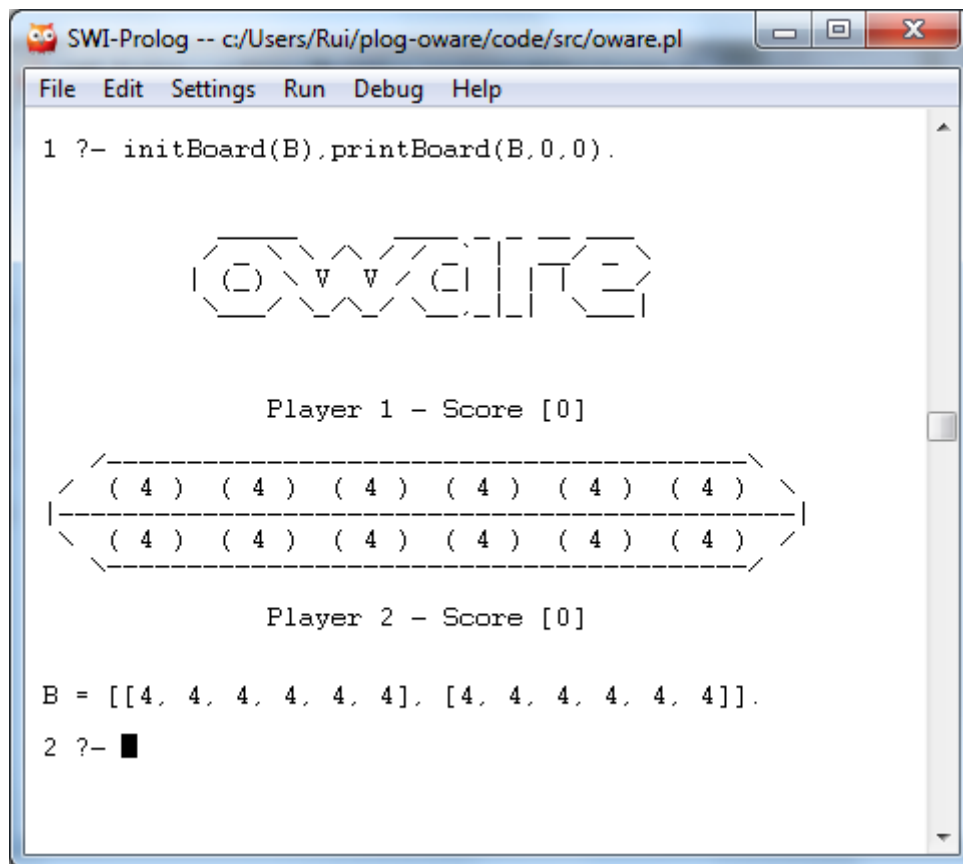


Figura 3: Começo do jogo

## 6 Conclusões e Perspetivas de Desenvolvimento

Que conclui da análise do jogo e da pesquisa bibliográfica realizada? Como vai ser desenvolvido o trabalho? Que parte (%) do trabalho estima que falta fazer?

## **A   Nome do Anexo A**

Código Prolog implementado (representação do estado, cabeçalhos dos predicados de jogada e predicado que permite a visualização simples, em modo de texto, do tabuleiro).