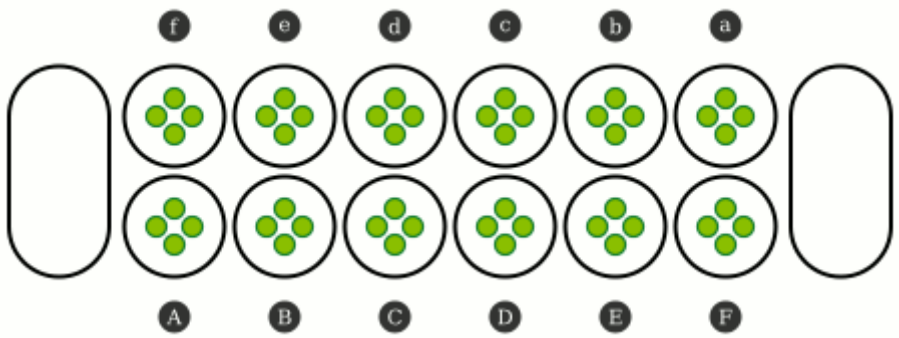
Oware Game

WORK OVERVIEW

The objective of this practical work is to develop a program to play the Oware game [1]-[3]. Oware is a strategy game for two players. It belongs to the wider family of mancala games, also known as count-and-capture games, consisting in the distribution of game pieces around the game board and its removal from the board when certain conditions are met. The playing rules can be found in the references [1]-[3]. As there are many variants of the rules, it is suggested that you follow those of the *abapa* version [1]. The program must allow two players to play the game, detect the end of the game and announce the winner or declare a draw. The program must execute in console mode, possibly with a colored characters interface.



Oware board in its starting position. Each of the playing holes (dishes or houses) contains 4 seeds; at either end there is a large score hole, empty at start.
source: http://www.joansala.com/auale/rules/en/

LEARNING OBJETIVES

The development of these programs will give students the opportunity to practice their skills of C/C++ programming, namely:

- use of program control structures (selection and repetition);
- use of several types of data structures (*strings*, *arrays/vectors, structs*);
- use of functions;
- development of simple program interfaces;
- management of invalid keyboard inputs;
- formatting outputs.

PROGRAM SPECIFICATION

Basic version

The program must do the following:

- Display the board in its starting position, as well as during and after each move.
- Allow each player, on his turn, to do his move, by collecting his seeds, sowing them and capturing the opponent seeds, if possible.
- After each move, detect if the game ended and announce the winner or declare a draw.

The program must execute in console mode. You are free to choose the interface of the program. The seeds in each hole can be indicated by numbers, instead of other symbols. Colored numbers may be used to distinguish the seeds of each player.

Enhanced version

If you feel that the basic version is too easy, you can implement an enhanced version that allows a player to play against the computer. It is suggested that you start by implementing a "random player version" that simply looks at the available choices and picks one of them, randomly; alternatively you can calculate the score from every possible move and pick the one with the highest immediate payoff.

PROGRAM DEVELOPMENT

This first practical work must be done individually.

When writing the program code you should take into account the suggestions given in class, specially the ones concerning the following issues:

- Adequate choice of identifiers of types, variables and functions.
- Adequate choice of the data structures to represent the data manipulated by the program.
- Modular structure of the code.
- Separation, as much as possible, of data processing from program input/output.
- Code commenting.
- Code robustness. Precautions should be taken in order to prevent the program to stop working due to incorrect input by the user, specially values outside the allowed ranges, and so on.
- Code efficiency; try to write efficient code.

WORK SUBMISSION

- Important note: Although the first practical work will not be scored, it is compulsory to do it in order to get approval in the practical part of the course. It is necessary to develop the program, to deliver and present it (if/when requested).
- Create a folder named **Tx_yyyyyyyyy**, in which **x** represents the class number (Portuguese "turma") and **yyyyyyyyy** represents the student's number (without the letters **up** or **ei**), for example, **T5_201700007**, for the student number 201700007 from class ("turma") 5. Copy the source code to the folder (only the files with extension **.cpp** and **.h**, if existing) of the program. Include also a file, **ReadMe.txt** (in simple text format), indicating the development state of the program, that is, if all the objectives were accomplished or, otherwise, which ones were not achieved, and also which improvements were made, if any.
- Compress the content of the folder **Tx_yyyyyyyyy** into a file named **Tx_yyyyyyyyy.zip** and upload that file in the FEUP Moodle's page of the course. Alternative ways of delivering the work will not be accepted.
- Deadline for submitting the work: **3rd/April/2020 (at 23:55h)**.

REFERENCES

1. Oware, https://en.wikipedia.org/wiki/Oware
2. Aualé, http://www.joansala.com/auale/about/en/
3. Jogo do Uril, http://viajar.sapo.cv/descubra-o-pais/lazer-e-cultura/jogo-do-uril (*in portuguese*)