

PRELIMINARY REPORT OF PLOG ON OOLONG

Gonçalo Vasconcelos Cunha Miranda Moreno João Francisco Barreiros de Almeida

201503871

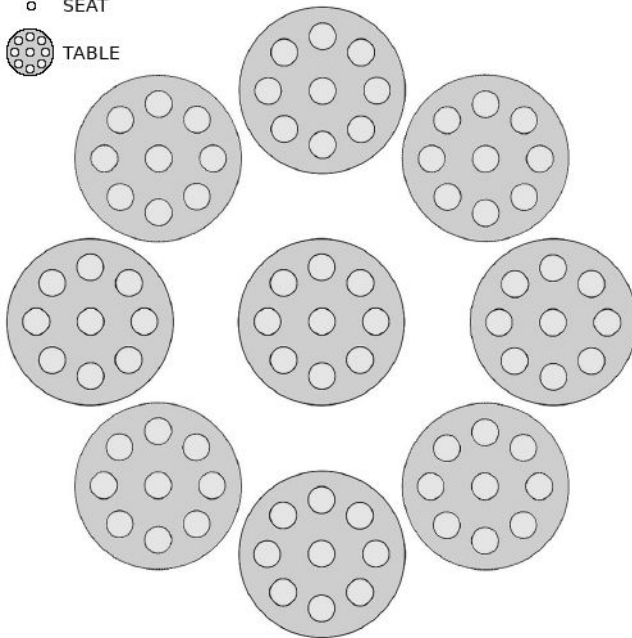
201505866

Oolong is an area-majority strategy game set in a Japanese tea house where the traditional [Oolong Tea](#) is being served. Each player represents a tea manufacturer trying to serve the most of its brand, thus maximizing profit. Once a player has served 5 people at a table they have won the favor of that table. Then, after a player has satisfied 5 tables they have won the favor of the house and therefore the game.

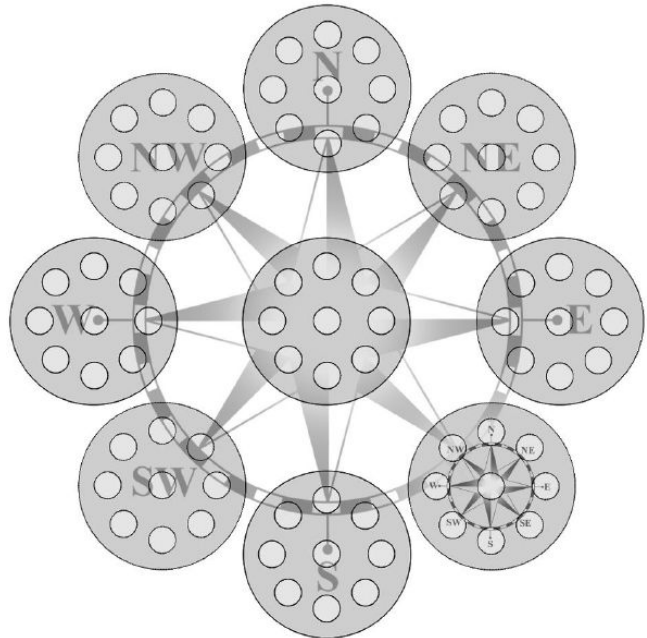
Setting the Game

The board is comprised of 9 tables, each has 9 seats where the game tokens are played. The arrangement of the tables should be similar to the layout of the seats of the tables. This way each seat is like a map for the overall playing area, with each table corresponding to a seat location. The easiest setup is the one below on the left, and for easier picturing the arrangement of the board a compass is regularly used.

○ SEAT
● TABLE



Pic 1 - Board arrangement



Pic 2 - Board arrangement visualization

Each player is then handed 40 tokens of different color, these tokens represent the tea the player is serving to a seat. The waiter pawn is placed in the center seat of the center table at the beginning of the game. Additionally each 8 perimeter tables will be randomly assigned a Special Marker that under certain conditions triggers an action.

Playing the Game

The player with the black tokens goes first, and since the waiter begins on the center table the player must serve tea on the center table. Depending on the seat the player decided to serve the waiter shall be moved accordingly. Below is depicted the actions that occur in a single turn.

1. Serve tea by placing a token.
2. Move the waiter accordingly.
3. Possibly trigger a Special Action.

Serving Tea

Every tea serving is directed as follows: The space on which tea is served indicates the table where the next player shall serve their next tea. For example, if John serves the first tea of the game on the N seat of the center table, Sarah must serve her tea on an empty space of the N table. However Sarah may not redirect John to play again on the same tile he was on, and as such may not serve tea on the corresponding (center) seat.

The Waiter

To more easily track where the next player must play, the Waiter pawn is moved immediately after a tea is served. Use the following guidelines to correctly place the Waiter.

1. Place the waiter on the tile which you have directed the next player to serve.
2. Place the waiter on the seat corresponding to the table you just played. For example, if you served tea on the center table, place the Waiter on the center seat, covering an already served tea if needed. Tea may not be served on the seat the waiter is on!

Special Actions

A Special Marker was randomly assigned to each perimeter table during setup of the game. These markers describe a special action that may be taken immediately once the needed number of matching tea has been served on that table. Once a marker has been used it cannot be used again for the rest of the game. If an action would trigger another action on a different tile that action also occurs, but multiple special actions are resolved in the order they are triggered. Below is a list of the markers and their effects.



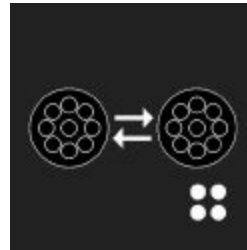
Black player may move one of their tea from any not served table to any other not served table.
Triggered with 3 matching



(2x) Player may rotate the targeted table to any orientation. The waiter also rotates with table.
Needs 4 matching teas.



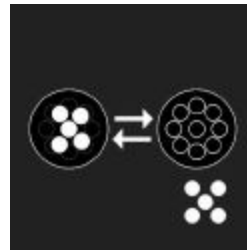
Green player may move one of their teas from any not served table to any other not served table.
Needs 3 matching teas.



Triggering player may swap the position of any two not claimed table (maintaining orientation). The waiter also moves.
Needs 4 matching teas.



Black player may move the Waiter from its current seat to the same seat on a different table, changing the next targeted table.
Needs 5 matching teas.



Triggering player may swap the position of any claimed table with an unclaimed one (maintaining orientation). The waiter also moves.
Needs 5 matching teas.



Green player may move the Waiter from its current seat to the same seat on a different table, changing the next targeted table.
Needs 5 matching tokens.

Claiming a table

Once a player has 5 served teas of their color on a table, they have claimed it. The table remains in play, though once all empty seats have been served it is considered complete. If any play (including special actions) should require a player to serve their tea on a completed table, they instead choose any seat on any other incomplete table to serve their tea (it is generally a poor strategic move to direct your opponent to a completed tile).

Winning

Once a player has claimed 5 tables they immediately win the game. It is not necessary to fill every space on the tiles to complete the game.

Game state representation

Oolong is a board game where positions are relevant, so we believe a list of lists is ideal to represent the game state internally, where the index of the element in the list represents the table and seat in which the token is to be placed. So, taking into account the picture below, the element at the array in the position (3,3) represents the North seat at the North table.

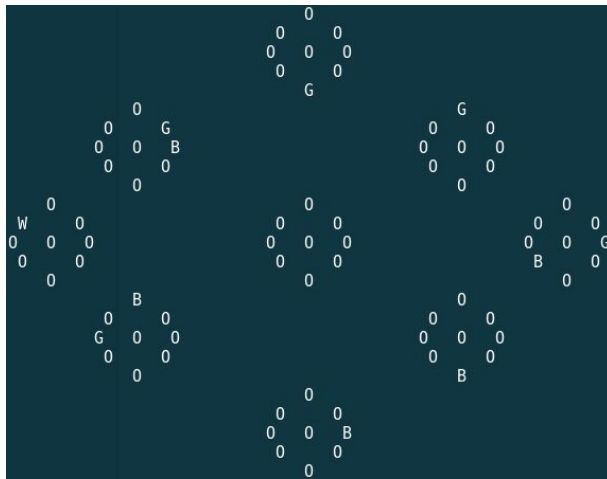


Pic. 3 - Index in the array of table in memory

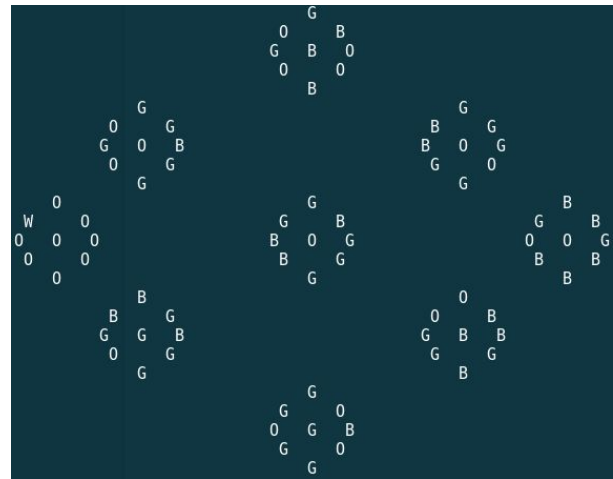
Game Visualization

At the moment we still have not defined how we are going to represent each special action and how to visually demonstrate to the user which special action is associated with each table, so the example we present of the visualization of the board is still not complete, but should not be very different.

We have associated the character 'O' with an empty space to resemble the physical board game. The black pieces have the 'B', the green pieces a 'G' and the waiter a 'W'.



Pic. 4 - Visualization of board mid-game



Pic. 5 - Visualization of board end-game (G wins)

When the user is asked to input the position of the tea to serve, for easier usability, they will be asked to insert first the table, according to the compass, so the top table is N. The seats of the table follow the same logic, so the leftmost seat on the leftmost table should be addressed as (W,W), meaning West table and West seat.

Player Movement

After the user has decided on the place to serve the tea, a predicate shall be called which will be executed which in turn will handle the board. The predicate shall receive the board (the user need not worry about this parameter), the table to serve and the respective seat. The last 2 parameters will be provided by the user.

```
serveTea(Board, T, S).
```

Pic. 6 - Header of function that receives user input

This shall be the only predicate that directly receives input from the user, however internally more predicates will be called to ensure that the serving is valid and that its repercussions are handled correctly. Below is the header of these predicates.

```
moveWaiter(Board, T, S).
```

Pic. 7 - Predicate to move Waiter

```
activateAction(Board, T, P).
```

Pic. 8 - Predicate to activate special actions

```
at(Elem, 0, [Elem|_]).  
at(Elem, Index, [_|Tail]) :-  
    Index1 is Index-1,  
    Index>0,  
    at(Elem, Index1, Tail).
```

Pic. 9 - Predicate to get element at given index

The predicate defined in Pic. 9 is used to retrieve an element at index “Index” in a given list. The program will then check if it is a valid seat, in which case “Elem” takes the value ‘O’, otherwise it is not a valid seat as it is already served.