

# CS 4500 Project

## Phase 5: Graphical Observers, Referee-Player Integration

**Due:** Thursday, November 14, 11:59pm

### **Submission:**

For the *Programming Task*, place

- `observer.PP` (or `Observer.PP`) in the `Admin/` directory within `Tsuro`

For the *Testing Task*, place the following in a new directory `5/` within `Tsuro`:

- `xref` and `ref-tests`
- `xobs` and `obs-tests`
- `README-tests.md`, describing how to run the test harnesses.

## **Programming Task**

Implement a graphical game observer.

The purpose of the game observer is to enable human observers to watch a game of `Tsuro` as it is played out by some AI players. The human observer should be able to see the current board state, which player's turn it is, and what is their requested placement (and out of which tiles).

As a minimum, the observer should be able to provide a rendering based on your tile rendering style (also see Testing Task below).

## **Testing Task**

### **Referee and Player Integration**

This task is an integration exercise for the referee and player components. As in previous testing tasks, the components might need some modifications to accommodate the test specification.

Develop a test harness for the referee named xref using Dumb as the player strategy.

The input is an array of three to five JSON strings, which represent the names of the players. The Players are arranged in descending order of age. The test harness should set up a game with the specified players and let the referee supervise it to determine the outcome.

The output is a JSON object with two fields:

```
{ "winners" : [ SORTED-NAME-ARRAY, ... ],  
  "losers"  : SORTED-NAME-ARRAY }
```

The sequence of SORTED-NAME-ARRAYs in the "winners" field drops any trailing empty array.

A SORTED-NAME-ARRAY is:

```
[String, ...]
```

where the Strings are sorted in ascending, lexicographic order.

Each SORTED-NAME-ARRAY in the "winners" field represents one rank of the final outcome. The first one collects all first-place finishers, the second one the second-place finishers, and so on. The array in the "losers" field collects all the players that requested illegal moves and were terminated as a result.

Example input:

```
["jack","alice","bob"]
```

Output:

```
{"winners": [["jack"]], "losers": ["alice", "bob"]}
```

Create two tests and place them in 5/ref-tests.

## Graphical Observer

Develop a testing harness for the observer component, named xobs.

**Input:** The input is a JSON game state specification, optionally followed by one turn specification. The game state specification is the same as in Phase 3. The turn specification gives the turn requested by the current player, and has two forms:

Initial move:

```
[ [color, tile-index, rotation, x, y], tile-index, tile-index, tile-index ]
```

Intermediate move:

```
[ [color, tile-index, rotation, x, y], tile-index, tile-index ]
```

You can assume that the input specifies a valid state specification. If no turn specification is present, the harness should simply render the current state, without the current turn.

**Output:** A rendering of the game state and the player's placement request in a file called `tsuro.EXT` where EXT is `png`, `jpg`, or `txt`, depending on what your chosen rendering style is.

Provide two example inputs and renderings in `5/obs-tests`.