

SE701 Assignment 3 Report

SOFTENG 701: Advanced Software Engineering Development Methods

Dhruv Phadnis

dpha900

The University of Auckland

I. CHANGEABILITY JUSTIFICATIONS

- 1) **MVC Pattern** - The base pattern for the implementation is the Model-View-Controller pattern. This instantly provides a separation between these three components. Unless functionality is missing, changes should not cross those boundaries.
- 2) **Modularity** - Extending from MVC, all other game concepts like rules and the UI have been implemented off of interfaces. This way, a different RuleSet or UI can easily be added by implementing the interface and swapping it in. This was inspired by a template-like pattern.
- 3) **Parameterisation** - For minor changes such as changing the number of players, houses or starting seeds. The model has been written in a generic way such that all of this can be changed through parameters.

A compatibility check function is required by the primary interfaces to ensure the two techniques in 2) & 3) work together.

II. FUTURE CHANGES

This is not required but I wish I got around to these:

- 1) Implemented a Player interface, then you could have a Local, AI, and Networked players.
- 2) The RuleSet interface functions are too broad, could've been decomposed into further interfaces (e.g. Sower, CaptureDetector, NextPlayerDecider, Scorer, BoardTraverser). So now you'd write smaller components, but can combine them into many more combinations.
- 3) Redo with a publisher-subscriber pattern at the core.