**Discworld Ankh Morpork**

*Software Architecture Documentation*

# 1 - Introduction

**1.1****Purpose**

This document provides a comprehensive architectural overview of the Discworld Ankh Morpork board game, using a number of different architectural views to depict different aspects of the game. It is intended to capture and convey the significant architectural decisions which have been made on the system.

**1.2****Scope**

This Software Architecture Document provides an architectural overview of the Discworld Ankh Morpork. The Discworld Ankh Morpork is being developed by Group 7 – Advance Programming Practices in Concordia University to play the game on computer by 2 to 4 people.

This Document has been generated directly from the Discworld Ankh Morpork Analysis & Design Model implemented in \_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**1.3****Definitions, Acronyms and Abbreviations**

Need to write.

**1.4****References**

Applicable references are:

Need to write.

# 2 - Architectural Representation

This document presents the architecture as a series of views; use case view, logical view, process view and deployment view. There is no separate implementation view described in this document. These are views on an underlying Unified Modeling Language (UML) model developed using -------------------------------.

# 3.  Goals and Features

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

* Ability to create and populate the data structures required to model the state of a game.
* Program must initialize all data structures appropriately for the start of a new game, following all of the instructions given on page 3 of the [rule book](http://www.fantasyflightgames.com/ffg_content/kingdoms/support/KN20_Kingdoms_Rulebook_sm2.pdf).
* Ability to save the current game state to a file, in a format of your choice (note that the user must be able to specify the name of the file to save to)
* Ability to load the game state from a file (note that the user must be able to specify the name of the file to load from)
* Ability to display the game state in text mode
* Ability to model an entire game, with 2-4 human players (using only the green-bordered Player cards for now; the brown-bordered cards are an optional feature for this build)
* Your code for this build must incorporate at least one design pattern other than Singleton (i.e. if you include Singleton, the requirement is to incorporate two patterns).