SOEN 6441 (Advance Programming Practices)

Project: Lanterns: The Harvest Festival
Build 3

GitHub repository - https://github.com/gugagreen/6441-summer-2015

Group A
Team members :-

Gustavo Gallindo.
Maxime Lamothe.
Parth Pareshkumar Patel.
Armaan Gill.
Ruixiang Tan

Submitted to: DR. Steven Winikoff

Instructions for running the application can be found on the README.txt found at the following link: https://github.com/gugagreen/6441-summer-2015/tree/master/lanterns

Table of Contents

1. Abstract

This document explains the software architecture of the project Lanterns**:** The Harvest Festival. It is the computer-based version of the board game. The document contains the overview of the basic classes involved and the relationships between them and software artifacts used for the incremental development of the game application.

2. Introduction

The project Lanterns the Harvest Festival is basically a game played with different combination of playing cards and favor tokens. The players of this game act here as artisans decorating the palace lake with floating lanterns. At the end of the festival which is the end of the game the player who earns the most honor wins the game.

3. Architecture Style

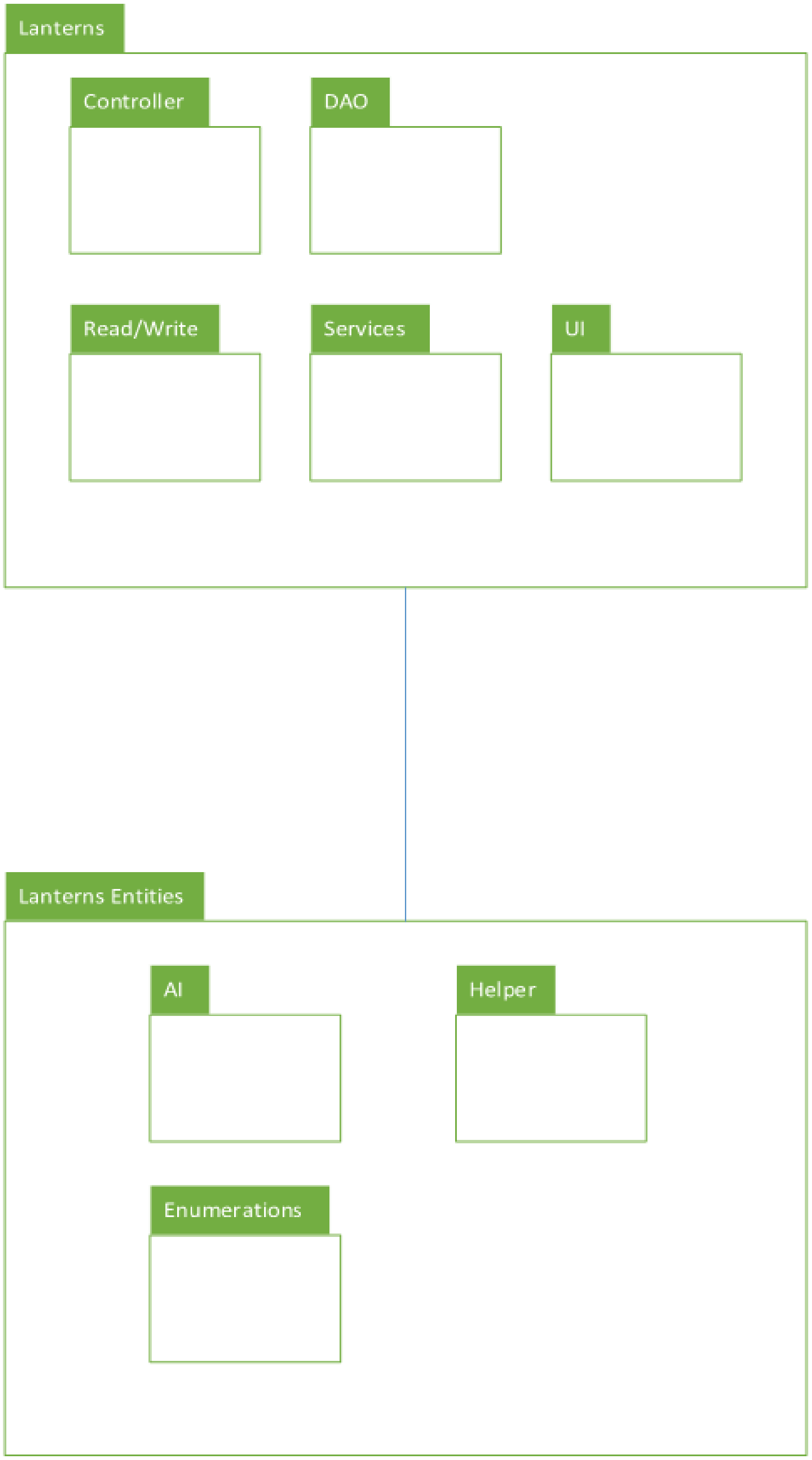
Our project Lanterns the Harvest Festival is based on commonly used **Model-View-Controller (MVC)** architecture. This pattern is used to implement user interfaces. This pattern separates the application into three interconnected components to separate internal representations of information from the user input. Our central components, which are Dao and Services packages, are part of the model and represent the behavior of the application, which is independent from the user interface. The view part represents the output of the information where multiple views of the same information are possible. In our project it could be multiple combinations of different lake tiles and player cards etc. The third part, Game controller, accepts input from the user and converts it for use in the model or view. The model notifies its associated views and controllers of state changes so that could help view to update interfaces and for the controller to modify the available commands.
We have implemented a **Strategy design pattern** for the end of game scenarios and player AI types. The strategy pattern (also known as the policy pattern) is a software design pattern that enables an algorithm's behavior to be selected at runtime. We implemented 3 end of game strategies and 4 AI players (Greedy , Random , Unfriendly and Unpredictable) as per requirement of the course project. The human player is also represented as part of the strategy pattern used for the AI. This allows the user to select any end game scenario as well as player intelligence levels for each new game, or newly loaded save game.

4. Package Diagram

The package diagram on the following page depicts the dependencies between the packages that make up our model. Lanterns module consist of main 2 sub modules such as Lantern Entities and Lantern Server.
Further breakdown of sub packages are:

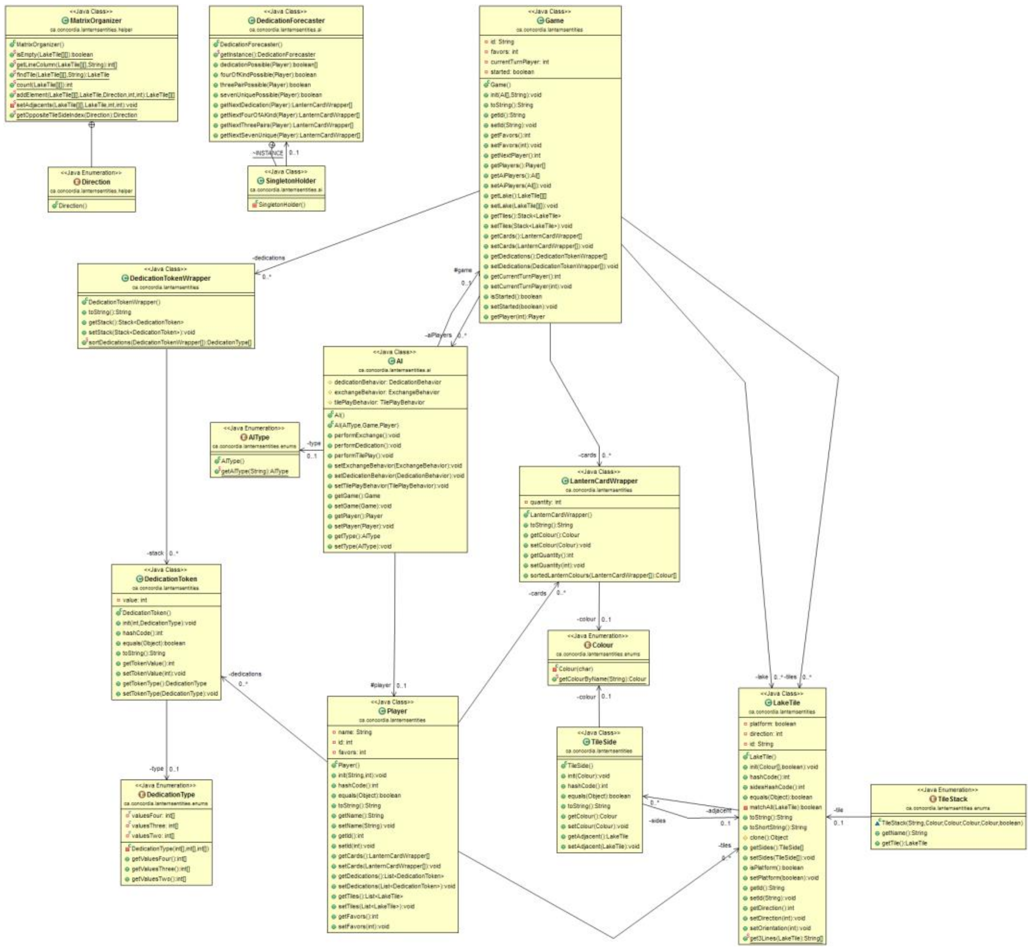
Lantern Entities contains Enumerations and Entities, such as colours and players.

Lantern Server contains Services, Read Write, Data Access classes, User Interface(UI) , Artificial Intelligence(AI). Moreover , UI package will acts as client to retrieve information
from the Lantern Server. In our project we are creating UI with Servlets and JSPs.

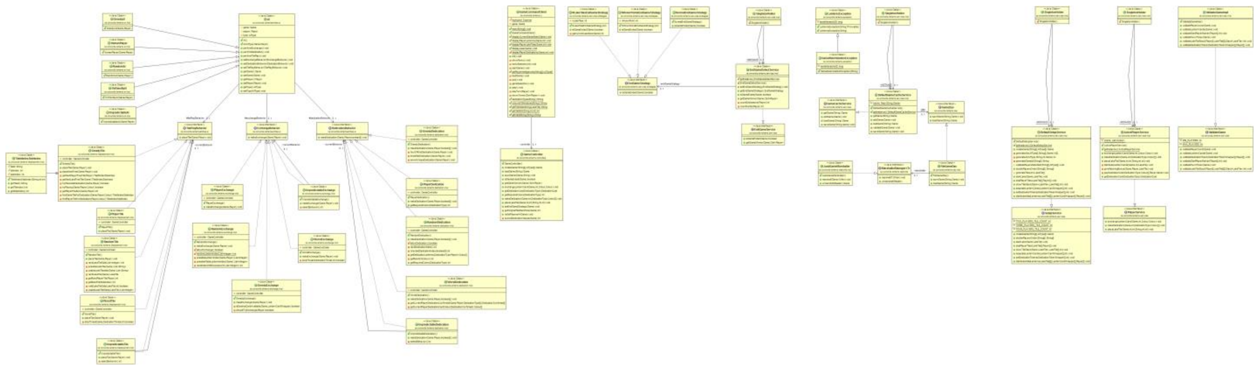

Lanterns
Controller
DAO
Read/Write
Services
UI
Lanterns Entities
AI
Helper
Enumerations

5. Class Diagrams

Lanterns Entities class diagram (zoom-in or open separate .png file for proper view).



Lanterns Server class diagram (zoom-in or preferably open the separate .png file for proper view).



6. Software Development Tools

The following are the tools used for the development of the application:-
Eclipse, GitHub, JUnit, Apache Maven, JAXB, Servlet and JSP

7. References

www.oracle.java.docs
www.javaranch.com
https://en.wikipedia.org/wiki/Strategy_pattern
Lecture Notes by Dr. Steven Winikoff , SOEN 6441.