Design Journal

for

Gin Rummy

Prepared by Devon X. Dalrymple

West Virginia University at Parkersburg

2022-03-07

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 What is this design journal for? 1

1.2 How are the sections organized? 1

1.3 Updating of sections 1

1.4 Source of Rules Used 1

2. Data Dictionary 2

3. The SOLID Principles 2

3.1 Why is this included 2

3.2 S – Single Responsibility 2

3.3 O – Open Closed 2

3.4 L – Liskov’s Substitution 2

3.5 I – Interface Segregation 2

3.6 D – Dependency Inversion 3

4. Use Cases 3

5. Functional Requirements 3

6. Design Story 3

7. Themes 3

8. CRC Cards 3

9. Design Patterns 3

# Introduction

## What is this design journal for?

This design journal is meant for a Web version of the card game, Gin Rummy. It is to be built so that games can be played against an AI opponent from any popular Web browser.

## How are the sections organized?

Most sections are ordered into the chronological order that they were done during the design process. The first section is reserved for use with the introduction and the last section is reserved for the data dictionary so that they may be found in their expected locations.

## Updating of sections

As new sections of the design journal bring to light faults or previous ignorance in the design process, older sections will be updated to try to keep the whole documentation from contradicting itself. If it is found that contradictory information still exists in this document, then the newer (later) sections should be used for determining the correct information.

## Source of Rules Used

### Link

<https://cardgames.io/ginrummy/>

### Reason

This provides a playable version of the game along with rules that are well written, easy to read, and to understand.

# Data Dictionary

No terms as of now…

# The SOLID Principles

## Why is this included

The SOLID principles are included in this design journal as to be a part of the design process from the get-go. Even though these principles would likely be still be applied to the design process, having them listed here can help to show areas of improvement related to them a lot quicker.

## Source?

No source was used to list these besides my own memory of these principles.

## S – Single Responsibility

The single responsibility principle pushes for classes to do one specific type of task and to do that task well. Instead of a MouseAndKeyboard class, a better following of this principle would be something like having a Mouse class and a Keyboard class.

## O – Open Closed

The open closed principle pushes for classes/objects to be open for extension but closed for modification. To add new features or new relationships, existing methods should not have to be changed. Whereas it is fine to add new methods that help the class carry out its responsibilities.

## L – Liskov Substitution

Liskov’s substitution indicates that a subclass should be able to be used in the place of a superclass without causing functionality to work in unexpected ways for the caller. A tabby cat should be able to be used in the place of a cat to call upon to meow, hiss, and such without performing erratically.

## I – Interface Segregation

The interface segregation principle states that an object/class should not have to fulfill contracts for behavior that does not relate to them. Having IRunningActor, ISwimmingActor, and IFlyingActor as interfaces is better than having a IMovingActor interface that requires the implementation of it to have details for running, swimming, and flying even if the actor cannot do one of these things.

## D – Dependency Inversion

The dependency inversion principle states that abstractions be relied upon rather than concrete implementations. By allowing the most abstract but relevant type to be used, reuse of code is improved, and as new features are added to systems, code can connect to the new but similar types.

# Use Cases

## Brief Overview

1. Depending the end point for points

# Functional Requirements

# Design Story

# Themes

# CRC Cards

# Design Patterns