<Monopoly Game>

Software Architecture Document

# Introduction

## Purpose

This document provides a comprehensive overview of the system from an architectural perspective, which will use a variety of different architectural views to describe various aspects of the system. It is used to record and express important decisions that have been made on the architectural aspects of the system.

## Scope

This document is for the developing monopoly game which will run on android platform. It provides players with many functions of playing monopoly game.

## Definitions, acronyms, abbreviations

|  |  |
| --- | --- |
| Acronyms and Abbreviations | Definition |
| System | Monopoly Game system |

## References

Use-case realization document.

# Architectural Representation

The system uses double levels architecture, which separates UI and LL. This document will represent the system software architecture through the following series of views.

# Architectural Goals and Constraints

## Safety and Security

User information and achieves are stored encrypted.

## Portability

This Monopoly Game development is based on android platform, so it’s suitable for most android devices.

## Design and implementation strategy

Use object-oriented design and java to develop background. UI is developed with android API.

## Development tools

Hardware: Android mobile phone, PC with Windows OS

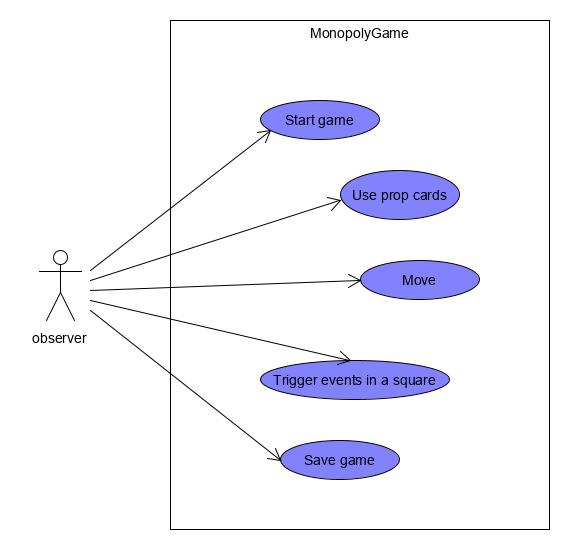
Software: JDK12, IDEA IntelliJ

# Use-Case View

## Use-Case Realizations

1. Start game
2. **Move**
3. Use propcards
4. Trigger events in square
5. Save game

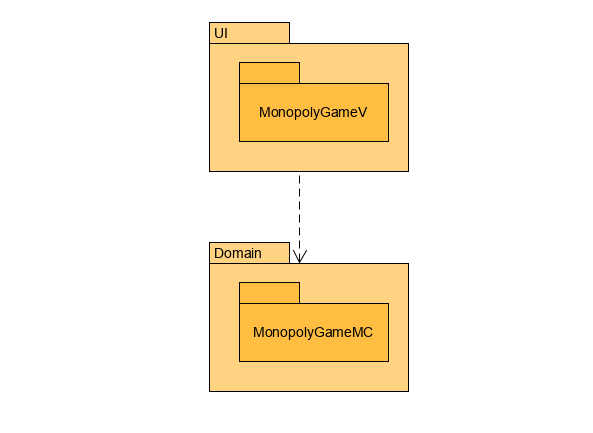
The use cases are listed above. Use case in **bold** is significant to the architecture.



# Logical View

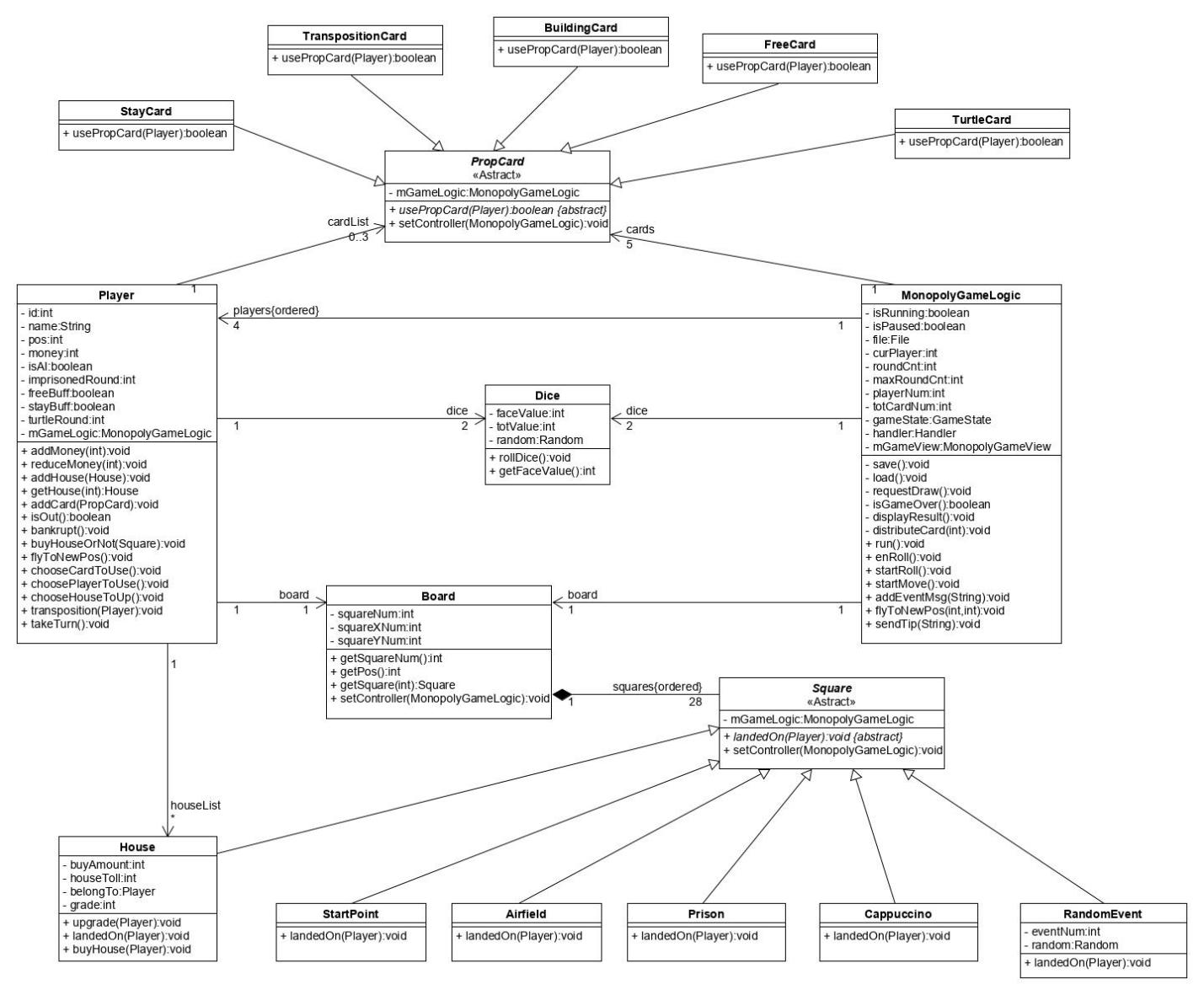
## Overview

The system applies typical MVC pattern -- model, view and control. This system is simply designed with just two layer -- UI layer which is view, and Domain layer which is model along with control.



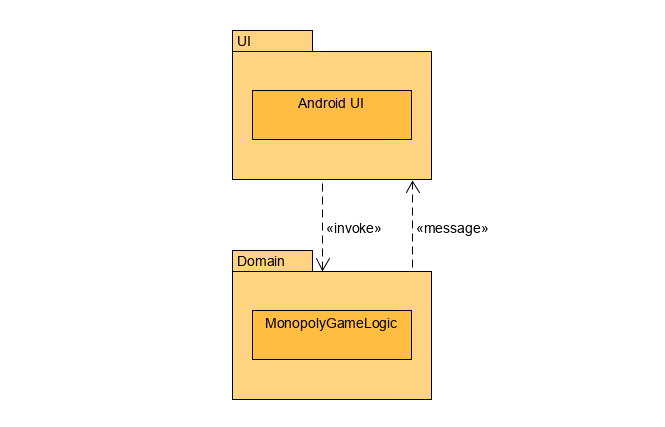
## Architecturally Significant Design Packages

MonopolyGameMC package:



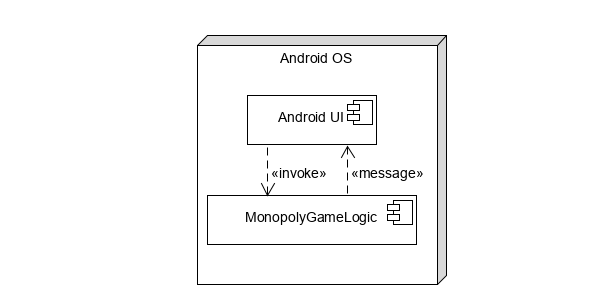
MonopolyGameLogic is a significant class that control the game logic and communicates with UI layer.

# Process View



The Monopoly Game has two thread, one is UI thread and the other is game logic thread. MonopolyGameLogic send message to UI layer for request of drawing, and UI layer invoke MonopolyGameLogic to gain attributes for presenting.

# Deployment View



This monopoly game is going to deploy and run on mobile devices which using android OS. Hardware configuration requirements: Android 9 or later version, 512MB RAM.

# Implementation View

## Overview

All the UI components are in UI layer and all the logic process in game are in Domain layer.

## Layers

UI layer and Domain layer have different threads. UI layer provides graphical and textual interfaces to interact with users and receives user’s operations.

Domain layer handles operations from user through the communication between threads and gives feedback to UI layer.

# Data View (optional)

None.

# Size and Performance

The software as designed will be run in Android platform. As it is a single player game, it has good performance in response time.

# Quality

Extensibility: To every object, enough interfaces are left, which provide possibility to develop more functions for the game.

Reliability: Before finishing development, enough tests and debug are needed. It will be tested on different mobile phones which have different version of android OS.

Portability: Use android API to develop this game, in order to make it can run on most android OS devices.