<AlphaXiu Gobang>

Software Requirements Specification

For <Subsystem or Feature>

Version <0.1>

Author: 软件1班陈涵 软件1班周灿苗 软件2班曹云舒 软件2班张天乐

catalogue

[1.Introduction 1](#_Toc75015759)

[1.1Purpose 1](#_Toc75015760)

[1.2Scope 1](#_Toc75015761)

[1.3Definitions, acronyms, abbreviations 1](#_Toc75015762)

[1.4References 1](#_Toc75015763)

[2.Architectural Representation 1](#_Toc75015764)

[3.Architectural Goals and Constraints 1](#_Toc75015765)

[3.1Safety and Security 1](#_Toc75015766)

[3.2Portability 1](#_Toc75015767)

[3.3Design and implementation strategy 1](#_Toc75015768)

[3.4Development tools 1](#_Toc75015769)

[4.Use-Case View 1](#_Toc75015770)

[4.1Use-Case Realizations 1](#_Toc75015771)

[5.Logical View 2](#_Toc75015772)

[5.1Overview 2](#_Toc75015773)

[5.2Architecturally Significant Design Packages 3](#_Toc75015774)

[6.Process View 4](#_Toc75015775)

[7.Deployment View 4](#_Toc75015776)

[8.Implementation View 4](#_Toc75015777)

[8.1Overview 4](#_Toc75015778)

[8.2Layers 4](#_Toc75015779)

[9.Data View (optional) 4](#_Toc75015780)

[10.Size and Performance 4](#_Toc75015781)

[11.Quality 5](#_Toc75015782)

< AlphaXiu Gobang >

# 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 AlphaXiu Gobang game which will realize by python. It provides players with many functions of playing AlphaXiu Gobang game.

## Definitions, acronyms, abbreviations

|  |  |
| --- | --- |
| Acronyms and Abbreviations | Definition |
| System | AlphaXiu Gobang 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 AlphaXiu Gobang Game development is based on JavaScript, so it’s suitable for most devices.

## Design and implementation strategy

Use object-oriented design and JavaScript to develop background. UI is developed by JavaScript.

## Development tools

Hardware: PC with Windows 10 and Mac OS Big Sur.

Software: vscode, coderunner.

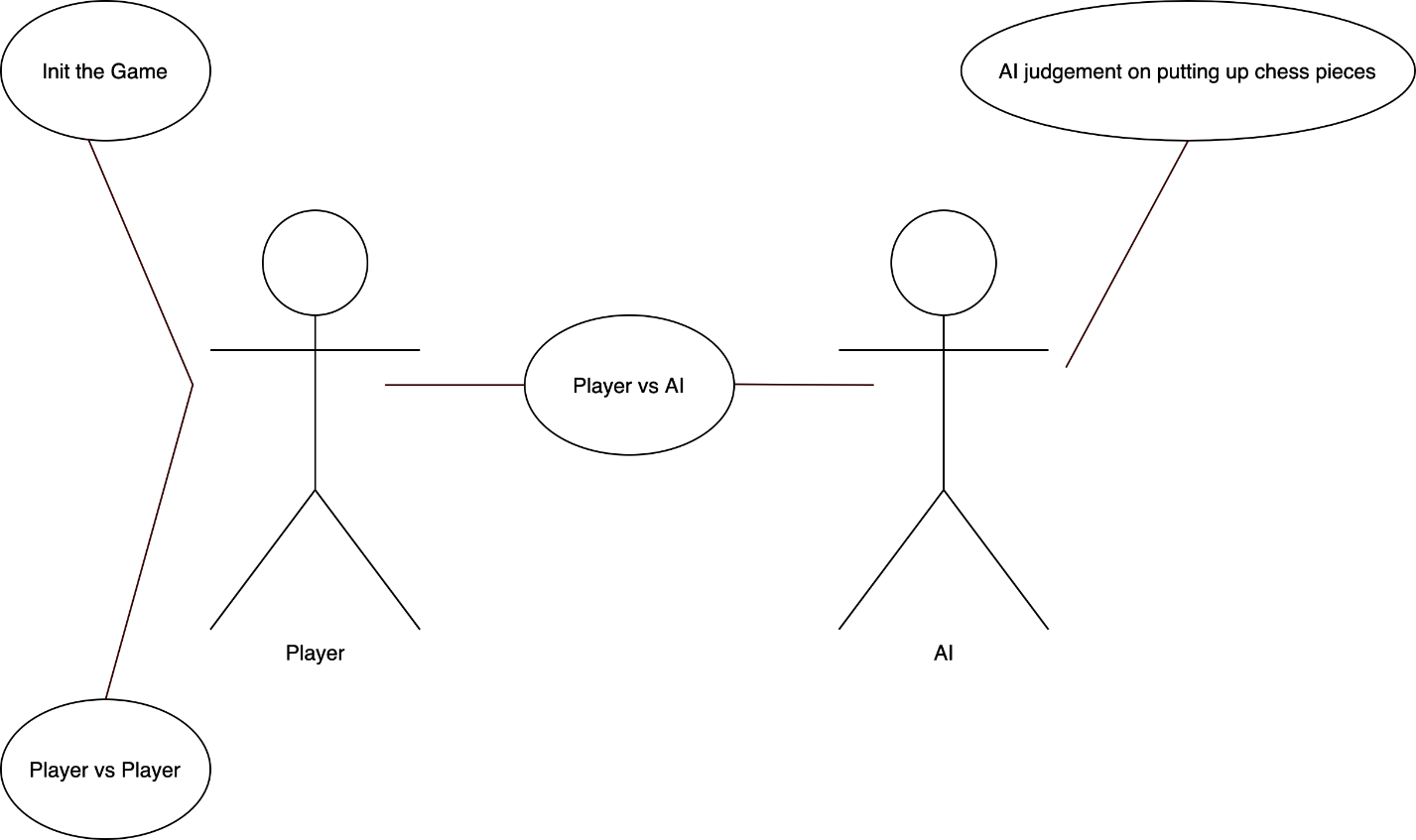
# Use-Case View

## Use-Case Realizations

1. Init the Game
2. Player vs player
3. Player vs AI
4. AI Judgemnent on putting up chess pices
5. Give a Weight for a Situation

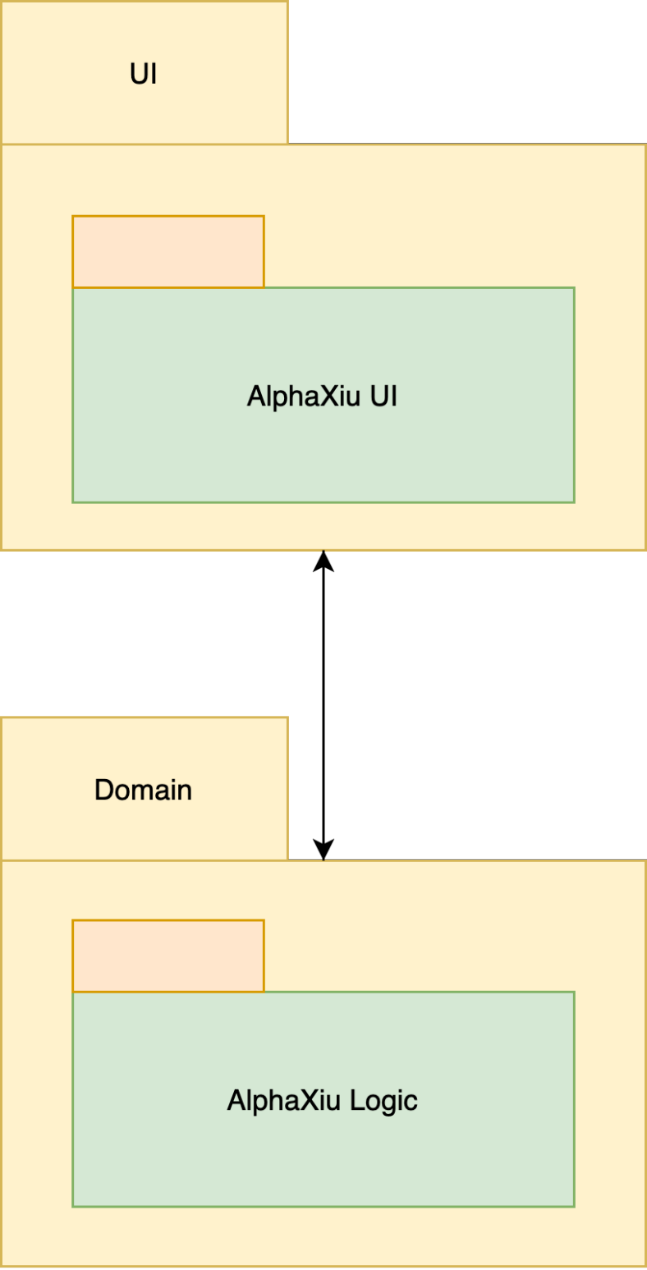
The use cases are listed above. Use case in **bold** is significant to the architecture.

# 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

AlphaXiu GobangGameMC package:

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

# Process View

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

# Deployment View

This AlphaXiu Gobang game is going to deploy and run on devices which has a browser with Chromium 80.0 or later. Hardware configuration requirements: x86 or ARM CPU, 2GiB 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 browser. As it is a single player/dual players 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 platforms which have different operation systems.

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