Group 14

Author: Wang Yi

Software Requirements

Proj.2 Huarong Path Game

Table of Contents

[Project Objective 2](#_Toc10411767)

[Domain Analysis 2](#_Toc10411768)

[System Architecture 3](#_Toc10411769)

[Use Cases 3](#_Toc10411770)

[Software Requirements 4](#_Toc10411771)

[R1: ServerUI 4](#_Toc10411772)

[R2: ChefUI 4](#_Toc10411773)

[R3: OrderDB 4](#_Toc10411774)

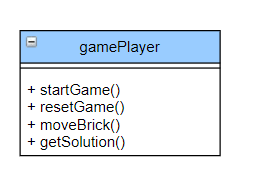
[R4: OrderProcessor 4](#_Toc10411775)

## Project Objective

In this project, we will develop a game of Huarong Path. By providing the visualized current state of the Huarong Path and operable interface, player could fully enjoy the game. Also, with the random initial state and the solution program, the interest of game will be improved and the player could challenge themselves.

## Domain Analysis

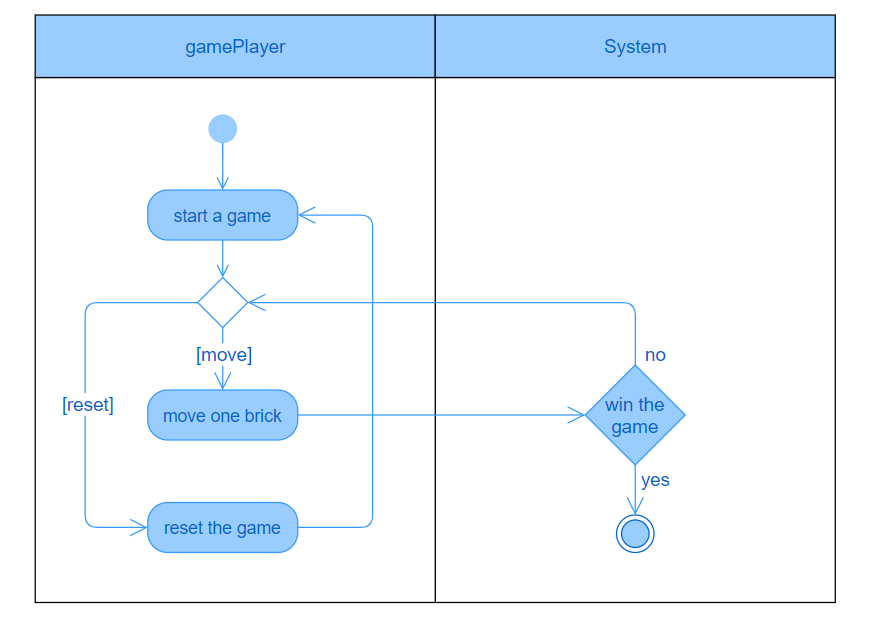
The participant of activities in a Huarong Path game is the gamePlayer.



The relationship of the participant is shown as follows:



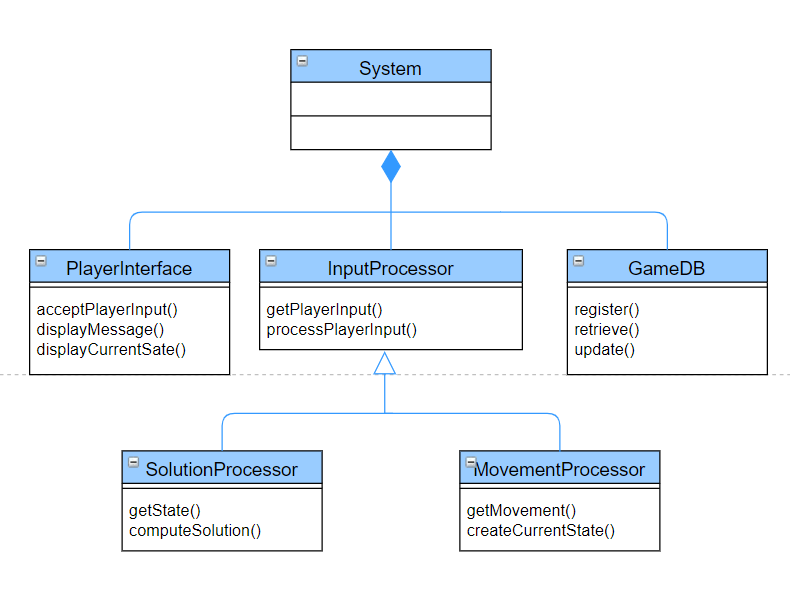
Here is the sequence of events for playing Huarong Path game:



## System Architecture

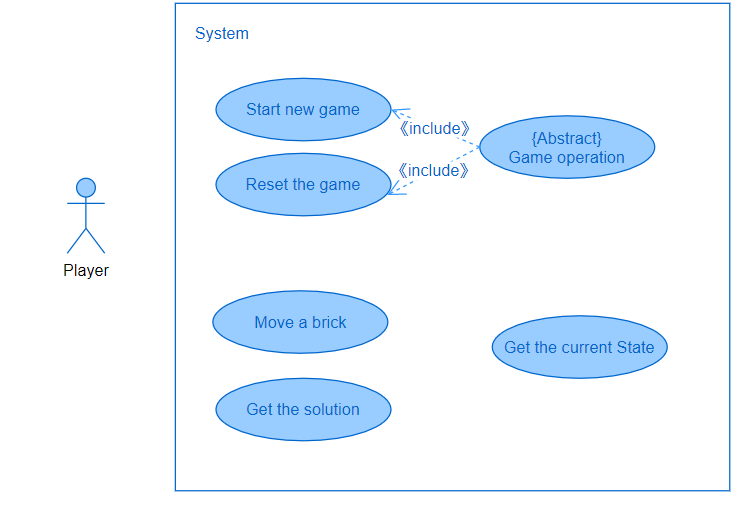
From the information above, we will design a software system that allows the player to have a visualized current state of the game and could move the bricks to win the game.

The system architecture is shown below:



## Use Cases

The system can achieve the following use cases from the player’s perspective:



## Software Requirements

### R1: PlayerInterface

* R1.1: The player should be able to move a block on the interface
  + R1.1.1: The player should be able to make a movement to the block that could be moved.
  + R1.1.2: The player can not move blocks which can not be moved.
  + R1.1.3: When player select one block, other blocks should be deselected.
* R1.2: The player should be able to get the current state of the game after each operation
* R1.3: The player should be able to get information from display area after each operation.
  + R1.3.1: The player should be able to receive warning from the display area when make illegal operation.
  + R1.3.2: The player should be able to receive correct message feedback matching with operation.
* R1.4: UI can update and display the number of operations.
* R1.5: UI can display the name of level.
* R1.6: Player should select level or random level.
  + R1.6.1: The player should play game ‘横刀立马’ without selecting level or random.
  + R1.6.2: The player should select 6 different level.
  + R1.6.3: The player should play random model in one of these six levels.
* R1.7: Player should be able to get hint from reach level.

### R2: Processor

* R2.1 The processor should be able to get the input from the UI.
  + R2.1.1 The processor should be able to handle the illegal input
  + R2.1.2 The processor should be able to execute the corresponding operation of the legal input
* R2.2 The processor should be able to execute the operation and give feedback to UI.

### R3: GameDB

* R3.1: The GameDB should be able to give the basic information of the game
  + R1.1.1: The GameDB should be able to give the size of blocks
  + R1.1.2: The GameDB should be able to give the positions of the blocks
  + R1.1.3: The GameDB should be able to give the map of the game level
* R3.2: The GameDB should be able to update the information after each operation from Processor

### R3: MapDB

* R3.1: The MapDB should save game level, name and solution.