HuaRongDao requirement

System general description

In the project, we are going to design a game software HuaRongDao to display the UI interface and design game system that enable the user to play the game. The software will enable the user to see current board and operate on it. User can also see the time and step he/she has taken and the best time and step record he/she has reached before. Fundamental operation will be provided such as clicking a piece, moving the piece in a direction, undoing, resetting, choosing current game level. Hint function will also be included in our project for better experience for the user.

Basic settings of the requirement

- There are 4 kinds of pieces, 9 pieces in total, classified by sizes and display. here are their sizes, name and display in the board (a x b, a for width and b for height):

Kind	Piece	Size
1	Caocao	2x2
2	Guanyu	2x1
3	Zhangfei, Machao, Huangzhong,Zhaoyun	1x2
4	Zu1, Zu2, Zu3, Zu4	1x1

- The board is 4x5, and will always have two cells that have no piece on them.
- The victory state is that Caocao is at the bottom and medium of the board.
- there are four direction: up, down, left, right for a piece to move.
- A piece can move if and only if after the move, the piece will not encounter other pieces and will be on the board.

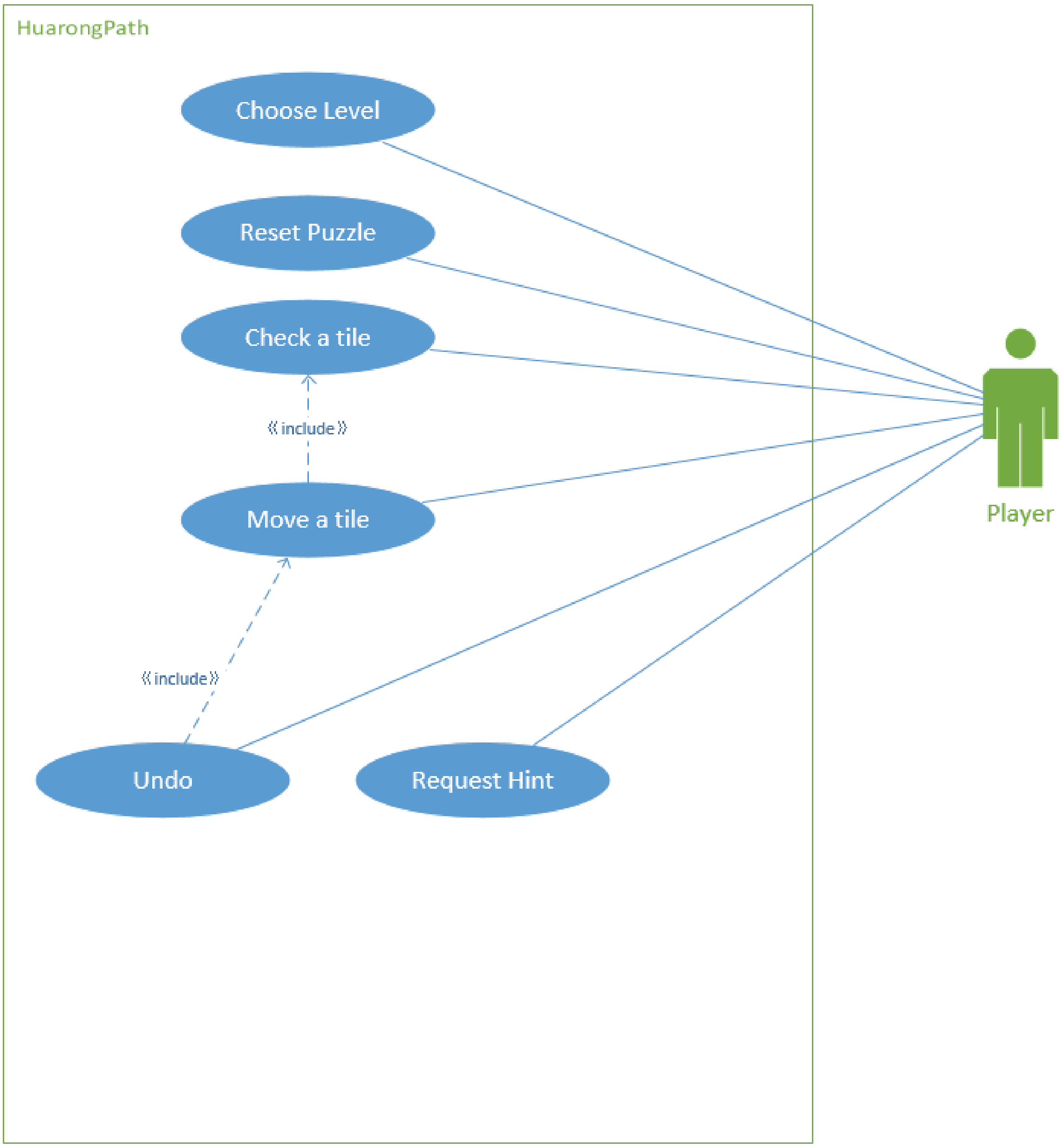
Description of usage for user

In the HuaRongDao, the system should enable the player do such things:

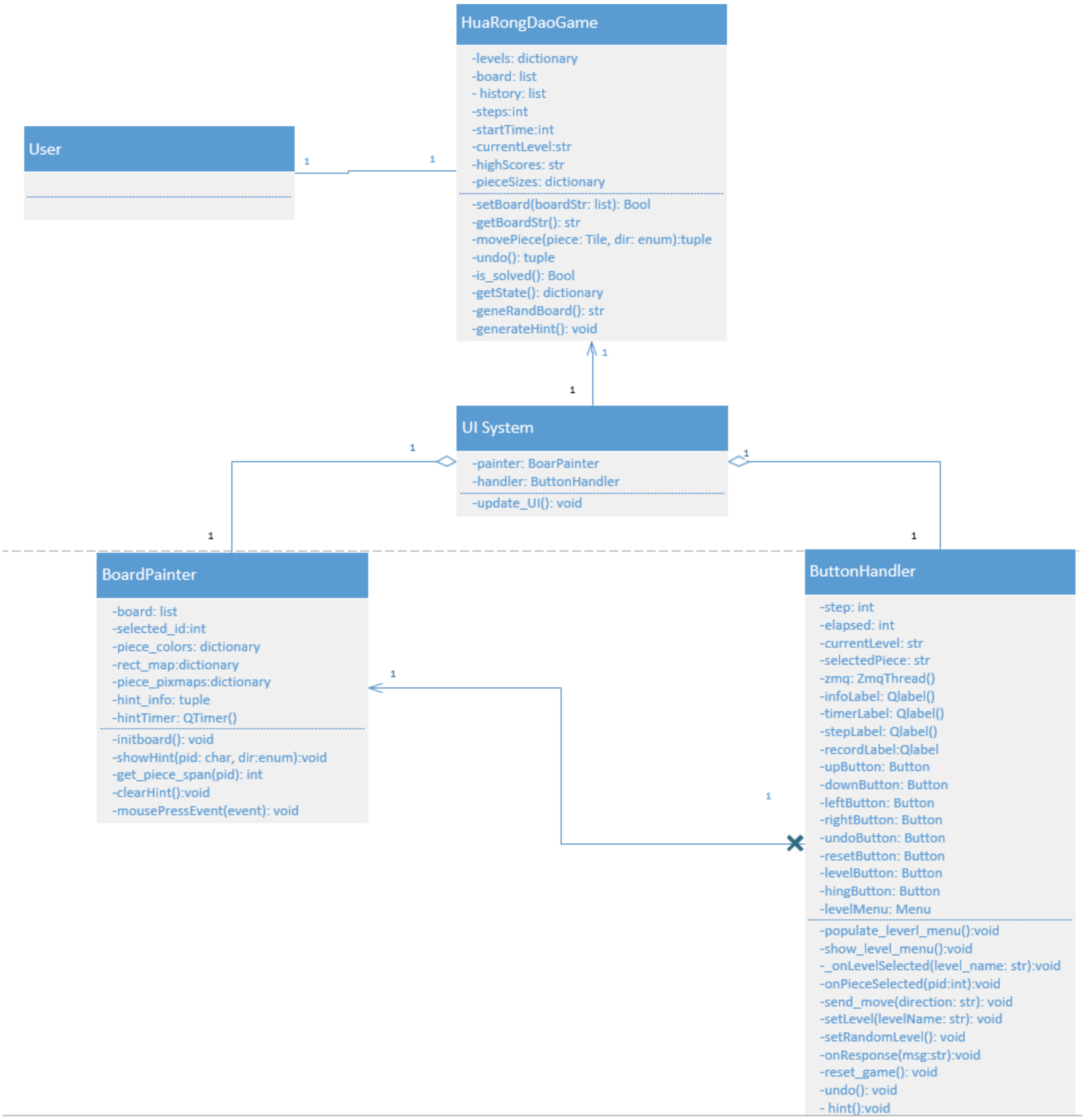
1. Check a tile: click and select a tile
2. Move a tile: after checking a tile, choose a direction and move it.
3. Undo: return back to last step's board.
4. Select level: choose the level of the game.
5. Request hint: ask the system for a hint a step.
6. Reset puzzle: reset the game.

Usecase Diagram

The usecase diagram :


HuarongPath
Choose Level
Reset Puzzle
Check a tile
《include》
Move a tile
《include》
Undo
Request Hint
Player

Class Diagram



Common Workflows

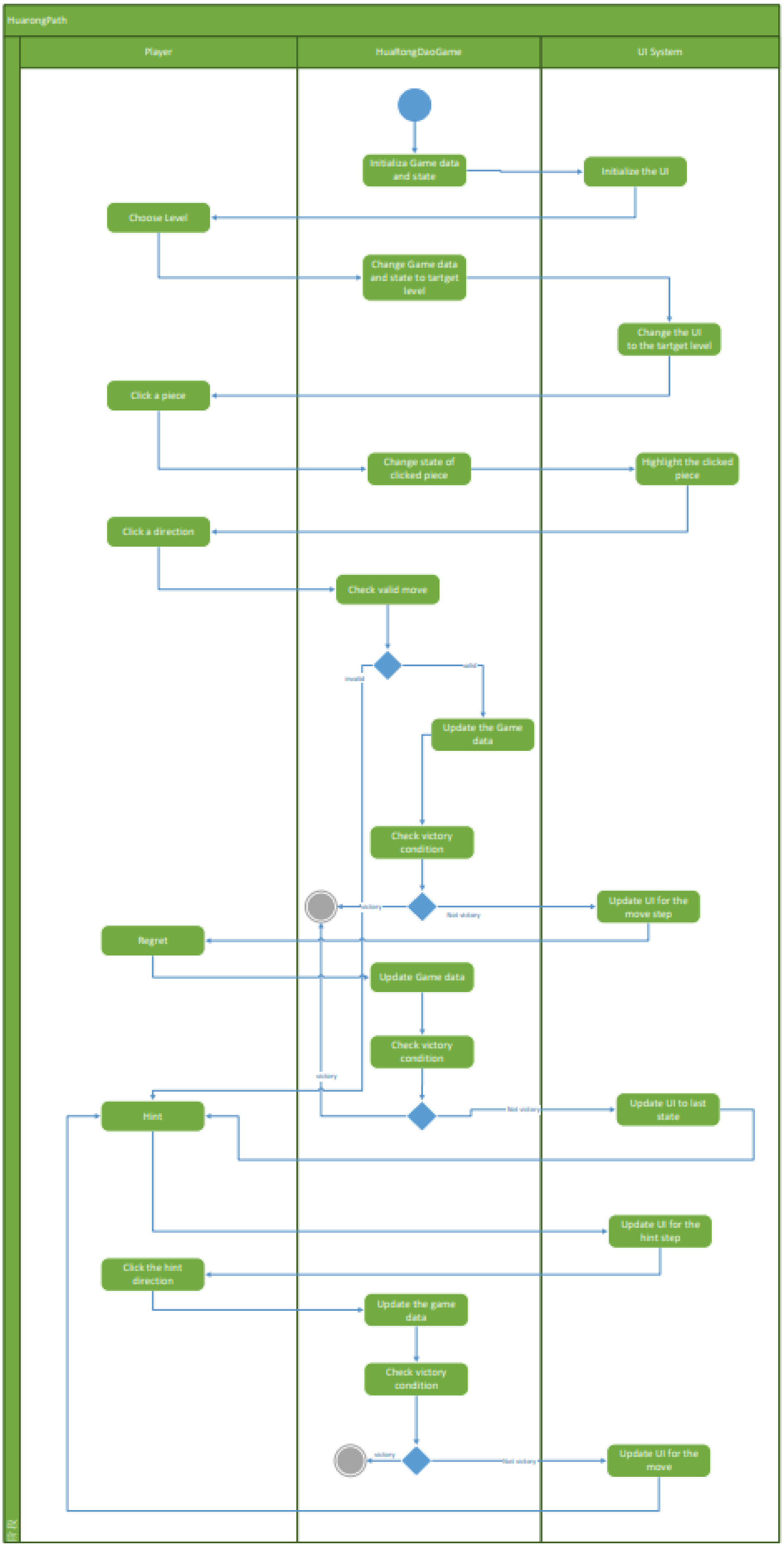
Here we use some user operations to express the workflows that will frequently happen when using the system.

It includes most of the buttons and the functionality in the system.

1. game start -> the game system initialize game data -> the UI system paint the game UI.
2. The user select a level -> the game system change its data to the level -> the UI system change UI to the level
3. The user click a piece -> the game system should update the selected piece -> the UI should highlight the clicked piece using its func that deal with the click
4. The user click a direction -> the game system should check if the move is valid ; if it is not valid, nothing happens to user; if it is valid -> game system update the current state of the board -> UI system update the UI for the move action.
5. The user click regret -> game system undo -> UI system update the board
6. The user click hint -> the UI system give hint visually using relevant function

7. The user click the hinted direction -> game system update the game data and check if victory condition is reached, if reached, game end; if not-> UI system update UI for the move step.
8. repeat 6,7 until victory.

Activity Diagram according to the example



Requirements Collection

1. Requirement1: Game UI
    - R1.1: The user should be able to see current board, including each piece and their kind (Caocao, Zhangfei,...)
    - R1.2: The user should be able to see the step that has been used and time that has passed.
    - R1.3: The user should be able to see the piece in which is selected currently.
    - R1.4: The user should be able to see the board's best time and step record in history.
    - R1.5: The user should be able to select a piece from the board
    - R1.6: The user should be able to move the piece selected.
    - R1.7: The user should be able to undo a step.
    - R1.8: The user should be able to select level from the UI.
    - R1.9: The user should be able to request for hint from the UI.
    - R1.10: The user should be able to reset the game from the UI.
    - R1.11: The user should be able to set random level from the UI.
2. Requirement2: The Board game system
    - R2.1: The Board should set the game board according to user selection.
    - R2.2: The game system should update its board from a move operation.
        - R2.2.1: if the move operation is valid, update the board and check if the game end with victory
        - R2.2.2: if the move operation is invalid, the game should do nothing.
    - R2.3: The game system should be able to undo a step and update its board for the undo.
    - R2.4: the game system should be able to give a hint step according to current board.
    - R2.5: the game system should be able to check if the board is solved.
    - R2.6: the game system should show the record of current level.
    - R2.7: the game system should be able to generate a random level.