

# justification

## **1. How can a player interact with the game? What are the possible actions?**

A player could have a series of actions to interact with the game: set, move, build. At the beginning of the game, each player needs to set their workers on the available space on the board. At every turn, the player is supposed to move one of workers and then build the towers, which could be block or the dome( but according to the rules, dome only can be built on the third floor).

In summary, there are 3 possible actions during this game: set, move and build.

## **2. What state does the game need to store? Where is it stored? Include the necessary parts of an object model to support your answer.**

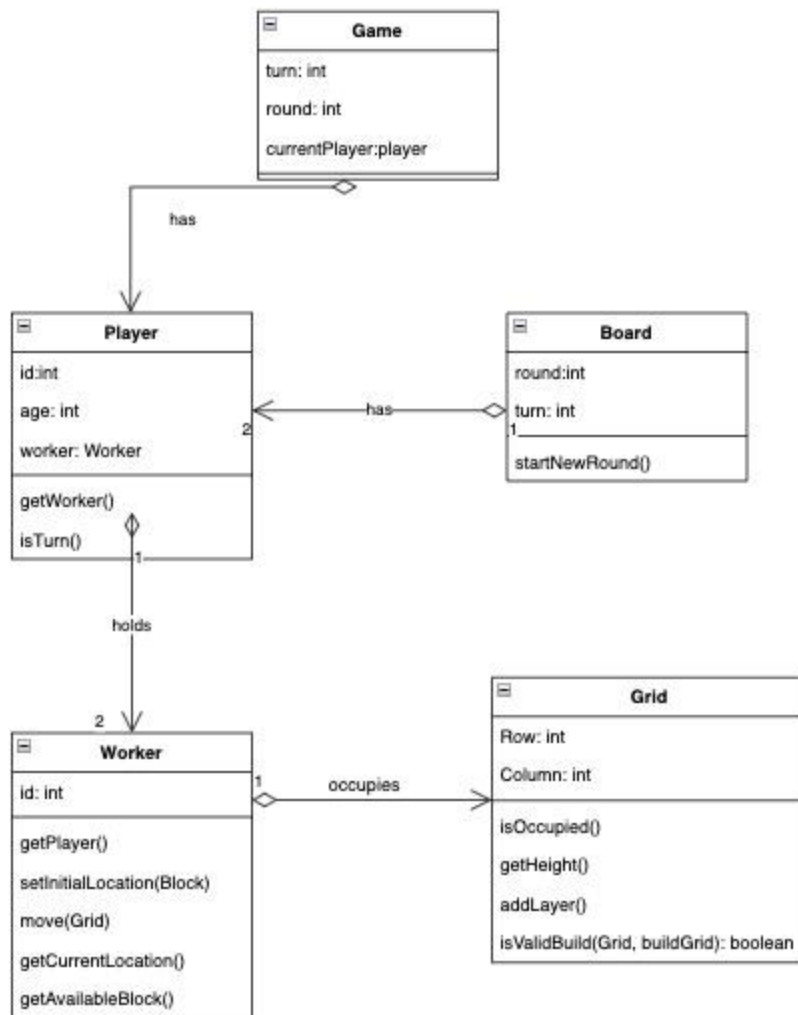
The game needs to store the game, the turn, players, workers and every grids situation.

Game state: record whether the game is still on progress, which player won. The number of the round. Which player is playing now. Does this turn end?

Player state: the player id, 2 worker id associated with him. Did he move in this turn? Did he build in this turn?

Worker state: the grid the worker is located. The available grid for the worker to move.

The grid state: the position and the height of the building.



Other alternatives:

- 1) Store the different towel in the towel state. Records where it is set on the board , whether it is a dome and whether there is a worker on it. But these information could be included in the grids and it is unnecessary to check if it is a dome. When the height is already 3, this grid would be no longer available. The advantage to add a state of towel is that the diagram as well the code could be more related to the rule, but would be less clear when developing the program.

**3. How does the game determine what is a valid build (either a normal block or a dome) and how does the game perform the build? Include the necessary parts of an object-level interaction diagram (using planned method names and calls) to support your answer.**

Valid building blocks: `getValidBuild()`: determine if the block location is a valid location( which means the (row of the block - row of the moved worker)  $\leq 1$  or  $\geq -1$  and (column of the block - column of the moved worker)  $\leq 1$  or  $\geq -1$ . What's more, the layer should be less than 3. And thus, this is a valid building action.

Since the interactive part is aiming at providing the process-flow, this is related to converting information between objects.

When comes to build the block, the system firstly needs to provide the possible building options, and the the player build the block and transfer the information back to the system, then the system determine if this is a valid action, finally action done!

