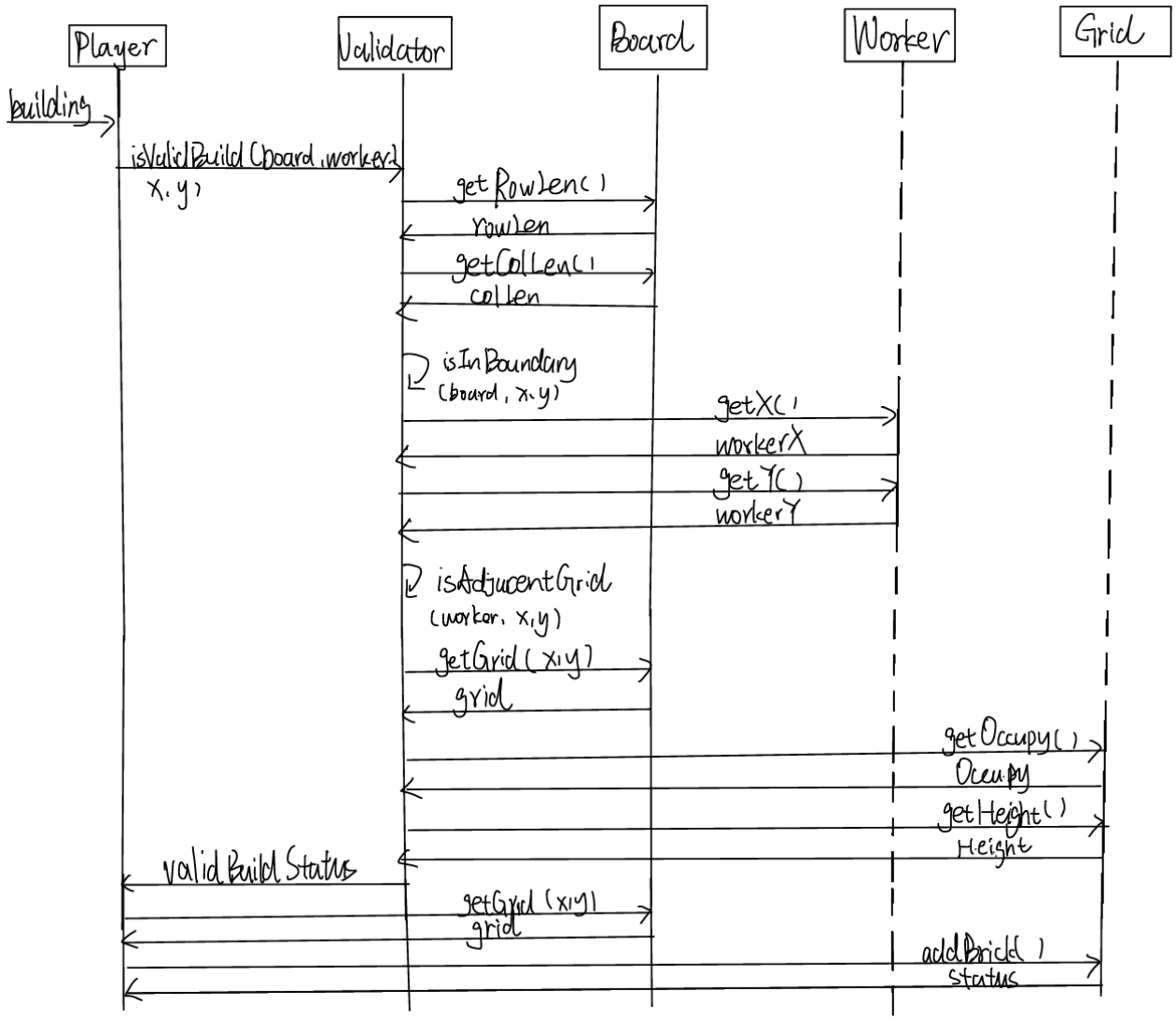
Valid Build Check

- Responsibility: Validator
- Actions & Methos:
  - Validator.isValidBuils(Board, int, int): check all the pre-requirements before building
  - Validator.isInBoundary(Board, int, int): check whether the selected position is in the boundary of the board
  - Validator.isAdjacentGrid(Worker, int, int): check whether the selected position is one of the eight grids that are adjacent to the worker
  - Grid.getOccupied(): check whether the selected position is not occupied by another worker
  - Grid.getHeight(): check whether the height of the grid is less than 4
- Justification:
  - Encapsulation: The Validator class focuses on validation rules, ensuring that each class is responsible for only one functional domain. This simplifies code maintenance and understanding.
  - Low Coupling: The Validator class maintains a low level of coupling with the Board and Worker classes. It does not directly alter the states of these objects but instead makes judgments based on their current states.
- Alternatives: Moving certain validation logic into the Worker class. This can reduce the responsibilities of the Validator class, but it may increase the complexity of the Worker class. What's more, using a Validator class for checks provides better high cohesion.

P.S: I don't check the block that player chooses. I only check the height of a grid. If the height of a grid is 3, worker.build() means using a dome; if the height is less than 3, it means using a block; otherwise worker.builds() is not valid.


Player
Validator
Board
Worker
Grid
building
isValidBuild (board, worker, x, y)
getRowLen()
rowLen
getColLen()
colLen
isInBoundary (board, x, y)
getX()
workerX
getY()
workerY
isAdjucentGrid (worker, x, y)
getGrid(x,y)
grid
getOccupy()
Occupy
getHeight()
Height
validBuildStatus
getGrid(x,y)
grid
addBrick()
Status