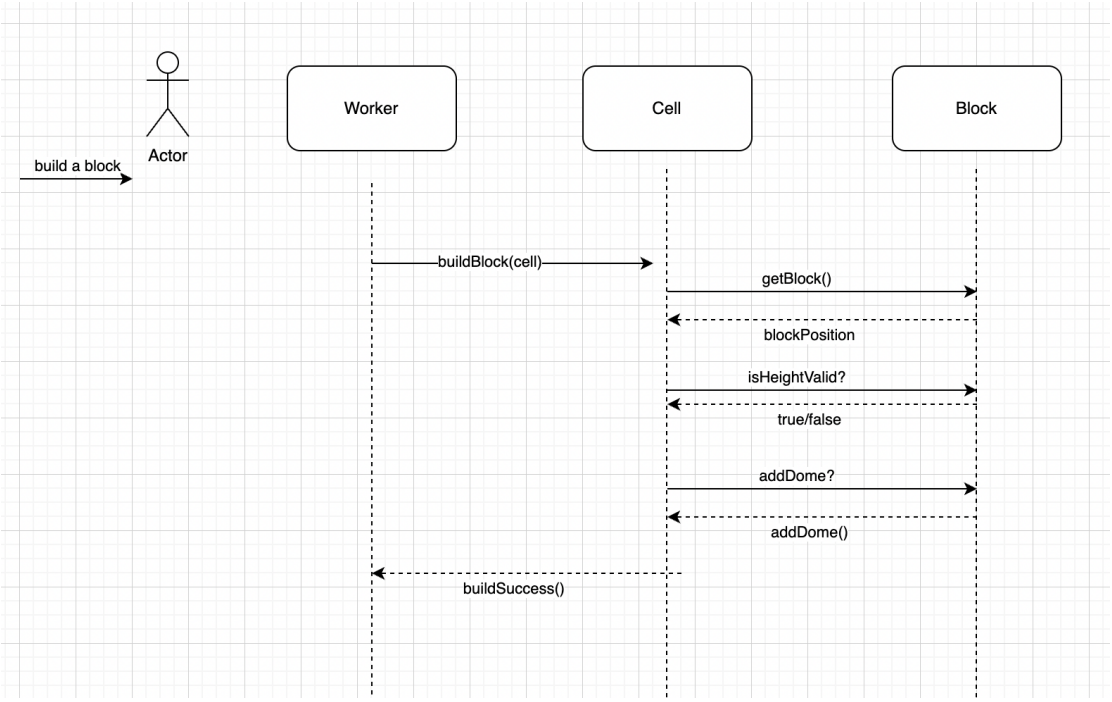
Valid Build Determination

The validity of a build can be determined based on the following criteria:

1. **Height Limit**: A block can only be added if the current height is less than 3.
2. **Dome Addition**: If the height= 3, a dome can be added instead of further blocks.



Responsible Objects and Methods

1. **Block Class**: Responsible for managing the state of individual blocks, including height and whether a dome is present.
    - buildBlock(): Increases the height or adds a dome based on current state.
    - addDome(cell Cell): Changes the state of the block to indicate a dome is present.
2. **Worker Class:** Coordinates the action of building for a specific player.
    - buildBlock(Cell targetCell): the worker performs the action to initiate the building process on a specific cell

Justification of Design Choices

1. **Encapsulation**: The Block class encapsulates all logic related to block states and modifications. This adheres to the principle of **Encapsulation**, keeping concerns separated and enhancing maintainability.
2. **Single Responsibility Principle (SRP)**: Each class has a specific responsibility:

    - The Block class manages its own state and update block height & hasDome or not.
    - The Worker class handles worker actions.
3. **Clarity and Simplicity**: The design is straightforward, allowing for easy understanding and modifications. Each method is focused on a single action, which enhances readability.

Alternatives Considered and Trade-offs

- **Validation Logic in Game Class**: One alternative was to place all validation logic within the Game class. However, this would lead to a more complex class and make it harder to manage the building logic. Keeping the validation tied to the Block class provides clarity and reduces the complexity of the Game class.