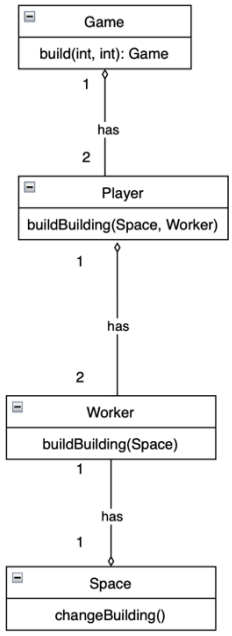
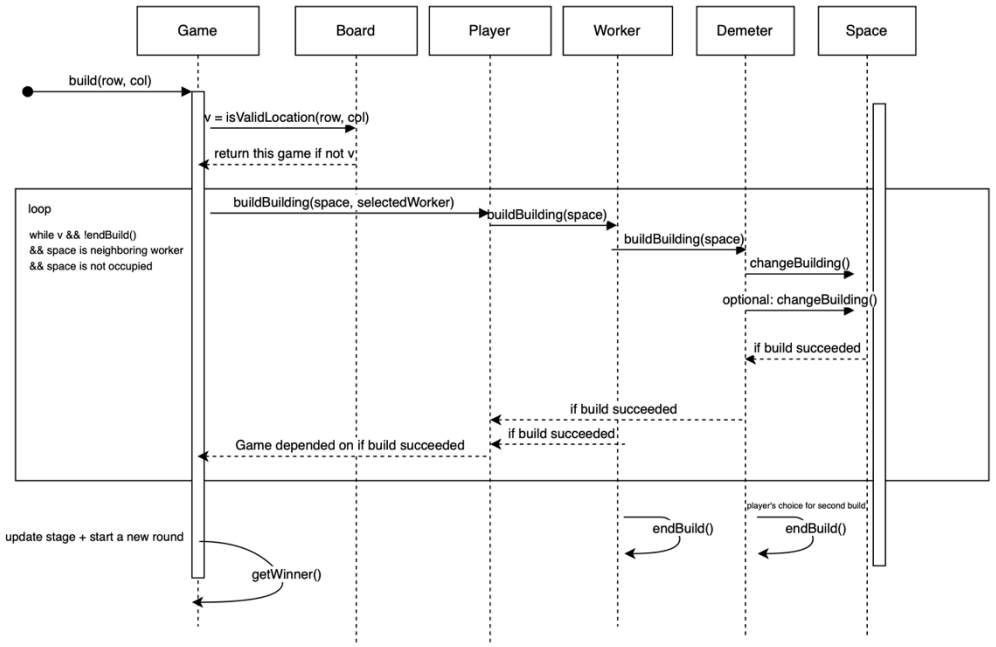
Re-answer the third question of Task 3 from HW3 (How does the game determine what is a valid build (either a normal block or a dome) and how does the game perform the build? Please include necessary parts of an object-level Interaction Diagram and Object Model to explain.)



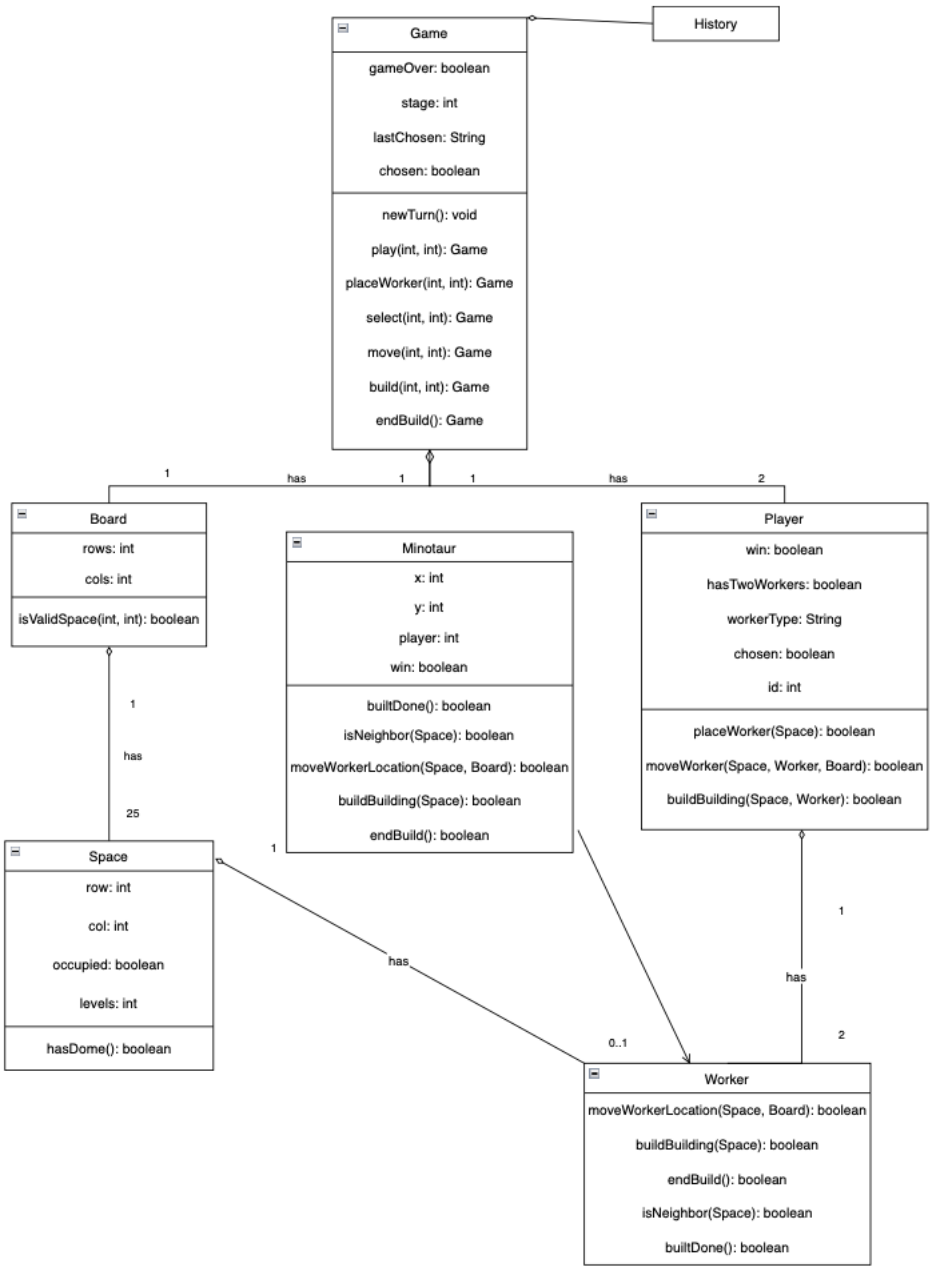


a. When a player wants to build, they specify the row and column index for the building.
b. Determine what is a valid build:
    i. No player wins so far.
    ii. The specified space has to be on the board.
    iii. The selected worker for this round should be at a location neighboring the space.
    iv. The space should not be occupied by another worker and should not have a dome.
    v. If the selected worker is not a demeter, then the round ends after the first build; if the selected worker is a demeter, the player has the option for a second build. If the player wants a second build, the space cannot be the same as the first built location in this same round.
c. Perform the build:
    i. If the building location currently has fewer than 3 levels, then 1 normal block should be added. Update the building by incrementing the number of levels by 1.
    ii. If the building at the location currently has 3 levels, then a dome should be added. Update the building so that the number of levels is 4. Update the space to being occupied.

iii. If the worker is a demeter: if it is the first build, update the lastBuild location to the location being built; if it is the second build, update the lastBuild location to be null.
iv. This player's turn is over. The game should be changed to the other player's turn.
v. Note that, a boolean that indicates whether the build is legal is being passed back to the game class. If the build is illegal, then the same game is being return; if the build is legal, then changes are made and the game has a new state.

d. Since the build operation just adds a building and does not move any worker, we do not need to check if the game is over. The requirements for a valid build are checked, and then the building is built.

How does your design help solve the extensibility issue of including god cards? Please write a paragraph (including considered alternatives and using the course's design vocabulary) and embed an updated object model (only Minotaur is sufficient) with the relevant objects to illustrate.



The use of polymorphism helps solve the extensibility issue of including god cards. There is a worker interface, and different god cards implement it with methods that may or may not be the same as each other. I started out thinking that the implementations for different god cards might be different, but gradually I realized that it might be better if I could use an abstract class to allow more reusability of code. Since most god cards do share most of their methods, and for

card-specific methods, we could override them. For example, minotaur is only different from a normal worker in the way that they perform move, so we would only need to write a move function for minotaur only. Such an alternative embraces design goals such as modularity and flexibility.

What design pattern(s) did you use in your application, and why did you use them? If you didn't use any design pattern, why not? Note that you need to update your answers to the questions of Task 3 from HW3.

I use the composite design pattern in the toString() function in the game state. It treats each cell uniformly and make up a string that represents the game state in JSON. I choose to use it because there are a lot of cells on the board, and I want the same features for each of the cell. It is also useful for hierarchical structures as I am processing the string in JSON format.

On the other hand, I have considered using the template method. Since we are clear with the rules supported by god cards, and the god cards usually share some methods with the normal/default worker, I have considered maintaining a set of relatively fixed worker methods and overriding specific methods for certain god cards. However, it might be possible that some god cards have very different methods than the normal worker class, or some might possess a combination of rules from different god cards. Since I am not sure if the few god cards for this homework are a good representation of all the god cards of the Santorini game, I did not use template method.