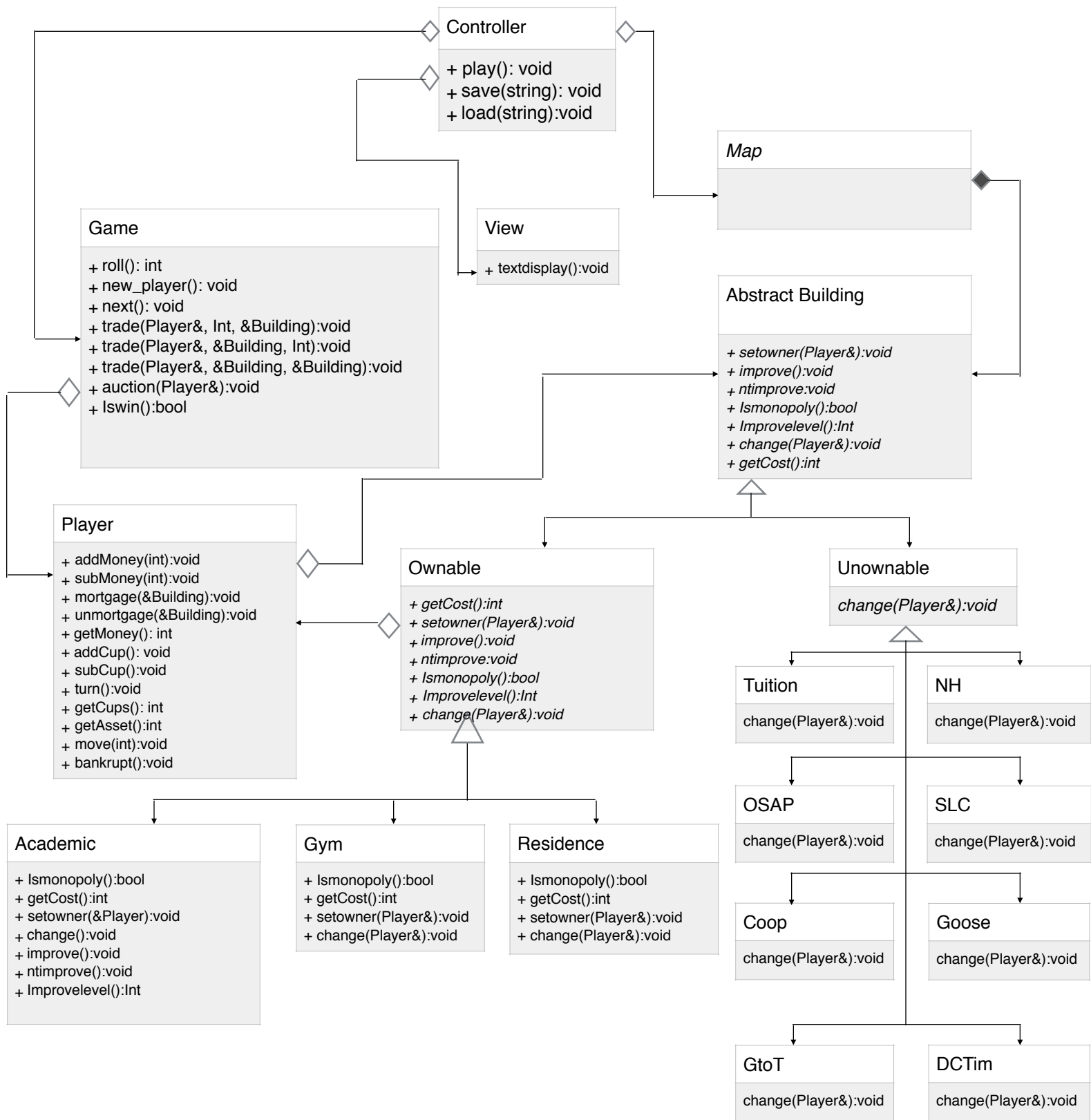


Project BB7K

Plan of Attack

Weixiang Deng
Weifeng Jiang



Breakdown

Task	Weixiang Deng	Weifeng Jiang
UML	July 13 - July 16	
Question	July 13 - July 16	
Game		July 17 - July 18
Player		July 19 - July 20
Building: ownable	July 17 - July 18	
Building: unownable	July 19 - July 20	
Map	July 21	
Controller		July 21 - July 22
Main function	July 22	
Test	July 23 - July 25	
Bonus	July 25 - July 28	

Q&A

Q1: After reading this subsection, would the Observer Pattern be a good pattern to use when implementing a gameboard? Why or why not?

Answer: Yes, the observer pattern works well. We have Building and Player classes, and the Ownable which is sub-class of Building would observe the Building class to see if any player here.

Q2: Suppose that we wanted to model SLC and Needles Hall more closely to Chance and Community Chest cards. Is there a suitable design pattern you could use? How would you use it?

Answer: Yes, the template pattern is useful. We can use it where NH and SLC would inherit from the common superclass(Unownable) with a change() virtual method. The method change() in SLC and NH can override it by themselves.

Q3: What could you do to ensure there are never more than 4 Roll Up the Rim cups?

Answer: We can make a static integer that count the total number of cups that all players own, and check as soon as each player comes to NH or SLC.

Q4: Is the Decorator Pattern a good pattern to use when implementing Improvements? Why or why not?

Answer: Yes, Academic buildings in the map is still buildings after implementing improvements, they do not change. So the decorator pattern is suitable for this situation. So we create a Building class with Ownable subclass with Academic subclass.