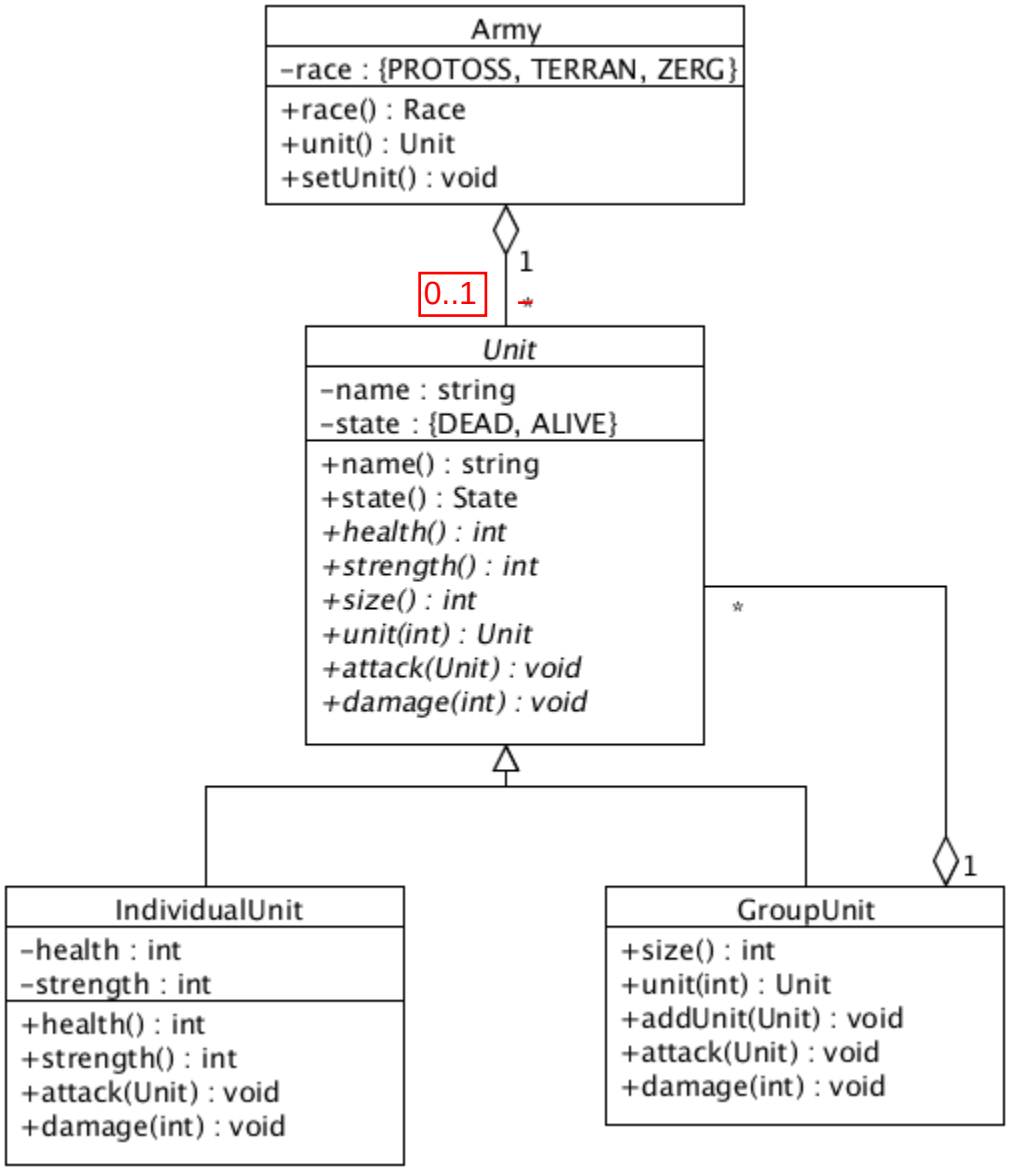
Q3a) Design Patterns

With the recent release of StarCraft 2, work on StarCraft 3 has already started at Blizzard. A new hierarchical system for the organization of unit has been proposed as a new way to manage armies. Units would be grouped together and these groups could be grouped together or with other units to create a hierarchy with, at the top, a common root owned by the army. This would be done using the Composite Pattern. These units should be able to transparently attack and be damaged, wether they represent a group unit or an individual unit. It should also be possible to access the units attributes (name, state, health and strenght). Finally, It should be possible to naviguate through the units by accessing their child units, if they have any. The resulting UML model is the following:



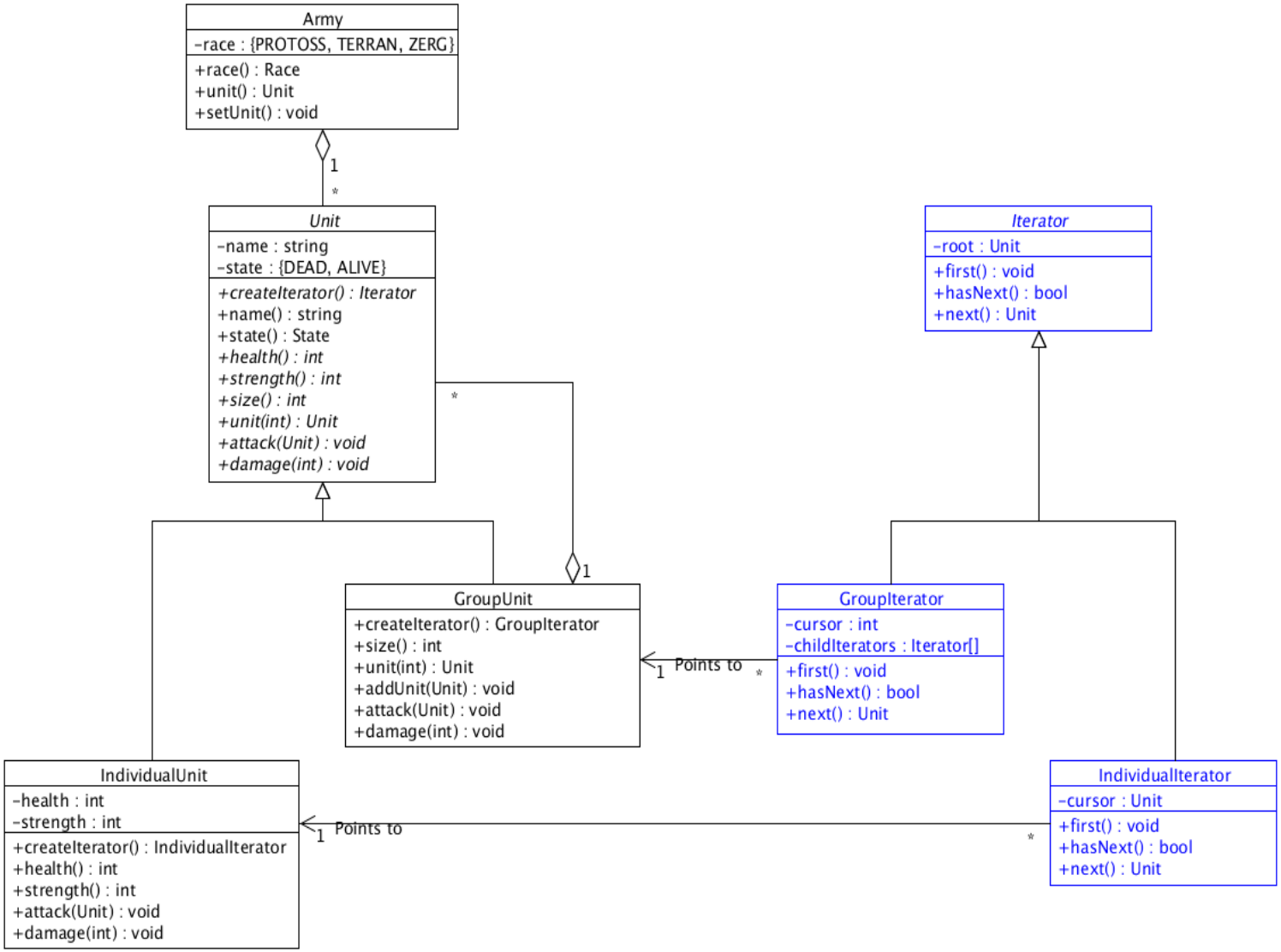
However, this design raises the problem of iterating through the individual units the same way it is possible to iterate through a simple list. This comes in useful when we want to display

information about all unit in the army to the player. We don't want to do a recursive traversing of the units everytime this is required. This is where the Iterator Pattern becomes handy. By implementing the Iterator Pattern, it will be possible to iterate through the units as if they were stored in a simple list.

Nice example of a nonrecursive implementation of the Iterator Pattern.
A nonrecursive implementation is easier when the iterator iterates only over leaf nodes and not all nodes. (In this case, the main program returns information about individual units only.)

Q3b) Design Patterns

The UML model* for the implementation of the Iterator Pattern is the following:



* Due to limitation of the software (UMLet), it was not possible to color the new associations (Points to) or the new operations (createIterator in all Unit classes).