

# Object Oriented - Project 2: Shadow Quest

James Stone - 761353

October 2016

## 1 Reflection

### 1.1 Changes made

Rather than follow my design for the assignment from Part A, I decided to model my solution off the sample solution. I followed this design closely with no major changes. I did however make some minor changes to a few private fields were removed such as:

`attacked` from the `Unit` class

`talking` from the `NPC` class

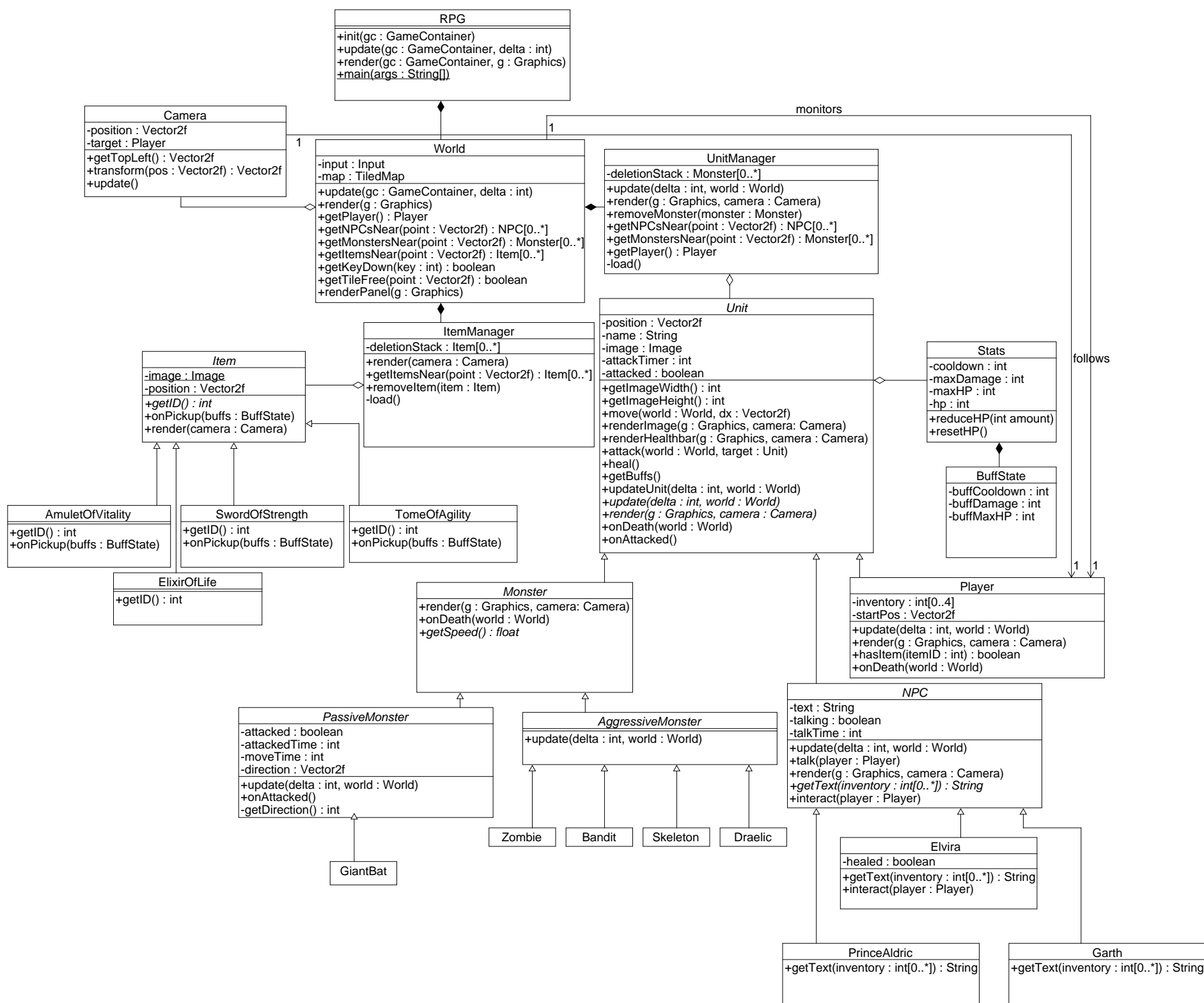
These were instead replaced by a private method called `underAttack` and `talking`. The advantage of this was it was one less field I had to keep updated, and they both could be represented through a simple calculation from other fields `attackTime <= 0` and `talkingtime <= 0`

#### 1.1.1 Class diagram

Please see the next page for the Class diagram.

And then the following page for the rest of the reflection.

Note: the class Diagram was modified from the Sample Solution.



## 1.2 Knowledge gained

I really enjoyed this assignment, it's really been an eye opener for me into how beneficial the Object oriented paradigm can be. It really maximises the amount of code re-usability, and all the advantages that come with that:

- less development time
- less bugs
- easier to test as more modular and abstract

This was really highlighted to me most when I had a bug in my **move** method of the **Unit** class. Once i fixed this bug suddenly all of my Monsters and the player could move again.

## 1.3 What I would do differently

If I was to do this again I would consider making the **statusPanel** its own class. To me it seems rather odd having it attached to a **world**, when the status panel relates to a **player** and there is one **statusPanel** per **RPG**. I would also consider making the **inventory** its own class. Alternatively I'd consider using an **ItemManager** as a **player's** inventory