Recommendations to the extensions in game engine

1) Actions class in engine should be merged into Item class. Not only the names are similar to Action class, it can cause easy confusion for those new to the engine codes.

What it would solve?: If the implementation is done in Item class instead, it is much easier to understand and use these methods in the Item class instead since Actions class is only ever used in Item class.

Design of the proposed extension: Refactor the whole Actions class such that the whole implementation is done in Item class instead. In the item class, ArrayList is used to represent the variable allowableActions with it's respective methods is done inside item class instead.

Advantages and/or disadvantages: The code is much more beginner friendly and avoids confusion between the multiple classes available in the engine. The disadvantage is experienced programmers may find it better for the implementation of Actions to be within their own class.

2) Ready made Game AI. This engine is lacking in AI where the engine is only restricted to a single player mode where the players can only play within themselves.

What would it solve? First it solves the hassle of developers having to develop an game AI for themselves. Second it gives the developer an opportunity to test the game engine and the actions of the game without any problems

Design of the proposed extension: Create a new class called AI. In this class there will be a few methods that make use of simple calculations to determine a player's action eg which location to move, what action to take etc. Then in the game folder itself, developers can extend this AI class and make changes to improve the AI.

Advantages or Disadvantages: A ready made game AI reduces work for developers to test their game. The game AI can also be improved upon as the developer sees fit. It saves time for new developers to develop an AI from scratch. Disadvantages are developers may prefer to build their own AI from scratch.

The new functionalities that developers can use is they can enable or disable the AI if they want to by choosing not to use that class or not.

3) GameMap extension. In the engine currently, the gamemap is only restricted to one map per display. This is an inconvenient feature as a player would be like to be able to have an overview of all the game maps he's in.

What would it solve? Player leaving the map unnesccarily.

Design of the proposed extensions:

In GameMap class, create another method called extendMap. In that method, just concatenate the current map and the new map object whilst keeping the state of the game intact.

In World class, whenever add gamemap method is called, we just call extendMap method of the current map so it will extend the current map and it will show on the display

Advantages: It solves the map display problem in UI, players do not have to 'leave' a map anymore. Maps can be extended dynamically while maintaining the game state.