**Changes to the Game Engine**

**Bad**

1. The ActorLocations object used in the GameMap class contains the actors from all the maps in the World class instead of that particular game map. This makes it hard to keep track of the actors in that particular map. **CHANGES**: Each game map should be responsible of their own actors and have methods to access the ActorLocations.
2. ActorLocations is not accessible from outside of the class. This makes it hard to obtain all the Actors and their locations within the map. For example, targeting valid actors with the SniperRifle. No direct methods to loop over the whole GameMap. For example, targeting valid actors with the SniperRifle has to be done by obtaining the NumberRange and nested loops. **CHANGES**: GameMap class should have an accessor method as to retrieve the ActorLocations on the current GameMap. This would allow for less repeated code to obtain the ActorLocations. However, it will increase privacy leaks as ActorLocations cannot remain as a inaccessible private attribute.
3. Menu display is only capable as an Action.

**Good**

1. Classes in the engine that deal directly with the Player and gameplay such as the Actor, Ground and Item classes use interfaces. The use of interfaces allows us as developers to add useful methods to the these classes without editing the engine classes.
2. Classes that need to perceive changes over time (eg. Actors, Ground, Items) have a tick() method that is called once every turn to allow inherited classes to take advantage of the passage of time. Follows DRY principle.
3. Location and Exit classes in the engine are useful to obtain certain areas. Location instances can obtain Exits and Exits can obtain their Locations. This makes it easy to cover a large specific area using both these classes.
4. NumberRange class in the engine allows iteration over each coordinate in the GameMap.
5. Capabilities class in the engine allows for differentiation of Items or Actors without downcasting.