**Week-2 (24-Sept to 30-Sept) Status Report**

**Observer Pattern**

* Understanding and implementing observer pattern by using Weather data example
* How it is providing encapsulation, loose coupling on implementation
* Implemented observer pattern using java build in java.util.Observable and java.util.Observer

**Reference** - Head First Design Pattern (O'Reilly): Chapter-2

XP Value: Simplicity

To ensure simplicity, I tried to keep the scope of the project (Game) more focused to basic functionality instead to adding complex feature. Additionally, I recommended creating small tasks that could be achieved in a week and we have MVP at end of each week.

**Week-1 (17-Sept to 23-Sept) Status Report**

**Abstract Factory and Factory Method**

*Factory method* defines an interface for creating an object, but let subclasses decide which class to instantiate. *Abstract factory* provides an interface for creating families of related or dependent objects without specifying there concrete object.

* Understanding implementation of factory pattern by using pizza shop example
* How it is providing encapsulation, loose coupling and less dependence on implementation

**Reference** - Head First Design Pattern (O'Reilly): Chapter-4

XP Value: Feedback

Initial discussion was around how to manage daily task and what feature could be added to the project(Game) so I try to provide my feedback as and when required such that our team is going in right direction.