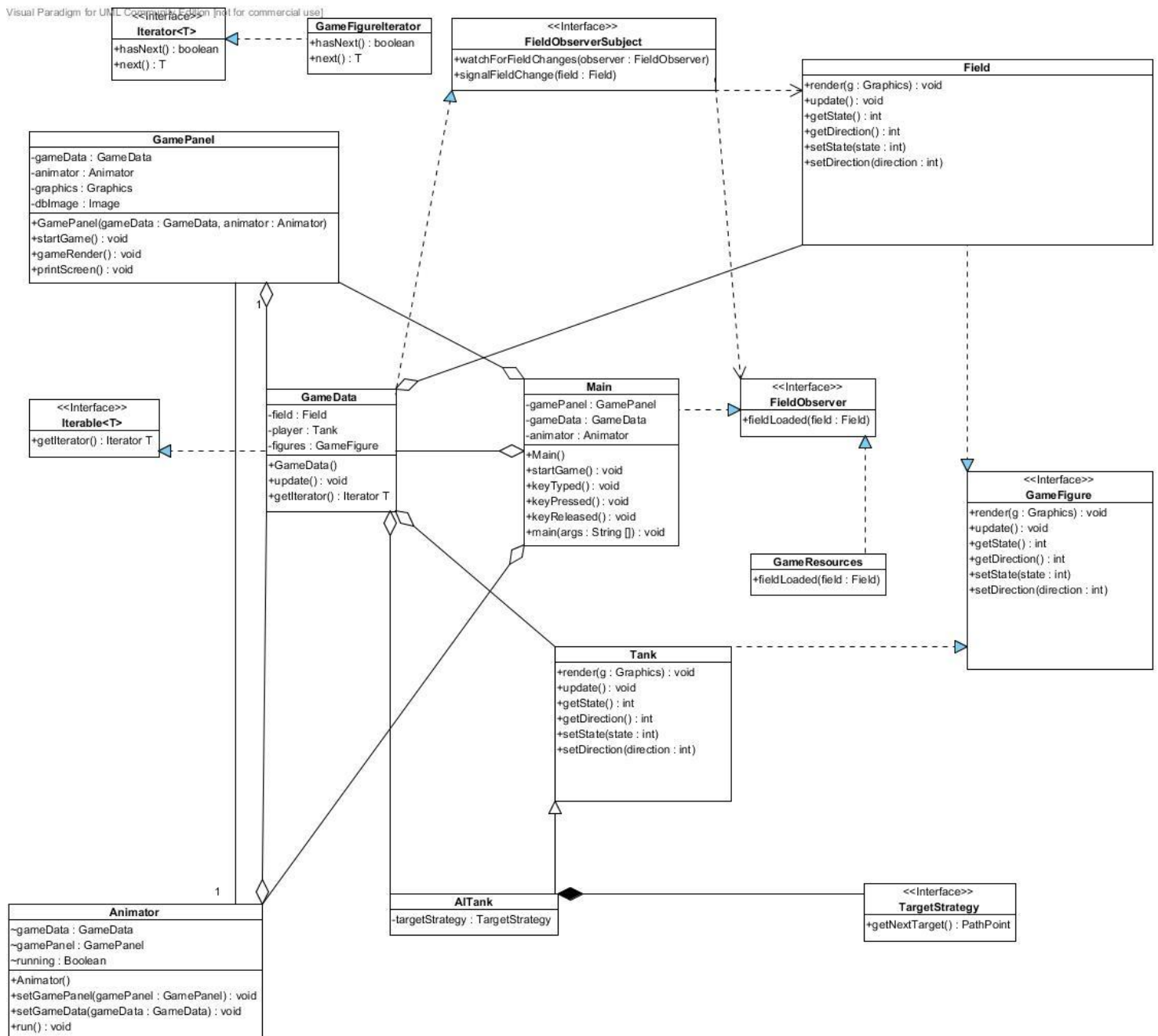


# Maze Wars Design Documentation

## UML Diagram



## Polymorphism Discussion

Maze Wars will implement the following design patterns: Iterator, and Observer. Beginning with Iterator, the `Iterator<T>` interface is the Iterator participant and the `Iterable<T>` interface is the Aggregate participant. The `GameFigureIterator` is the concrete Iterator, while the `GameData` class is the concrete iteration of the Aggregate participant.

To implement the Observer design pattern, the `FieldObserver` interface is the Observer participant, and the `FieldObserverSubject` is the Subject participant. The following classes are the concrete Observers: `Main`, and `GameResources`. The concrete implementation of the Subject is the `GameData` class.

The concrete implementation of the interfaces used the these design patterns is one example of how polymorphism is used. Another is the case of the `AITank` units that oppose the player in the game. The `AITank` is a child of the players `Tank` object. The `AITank` overrides the `Update` and `GetNextCorner`, also another example of polymorphism.