

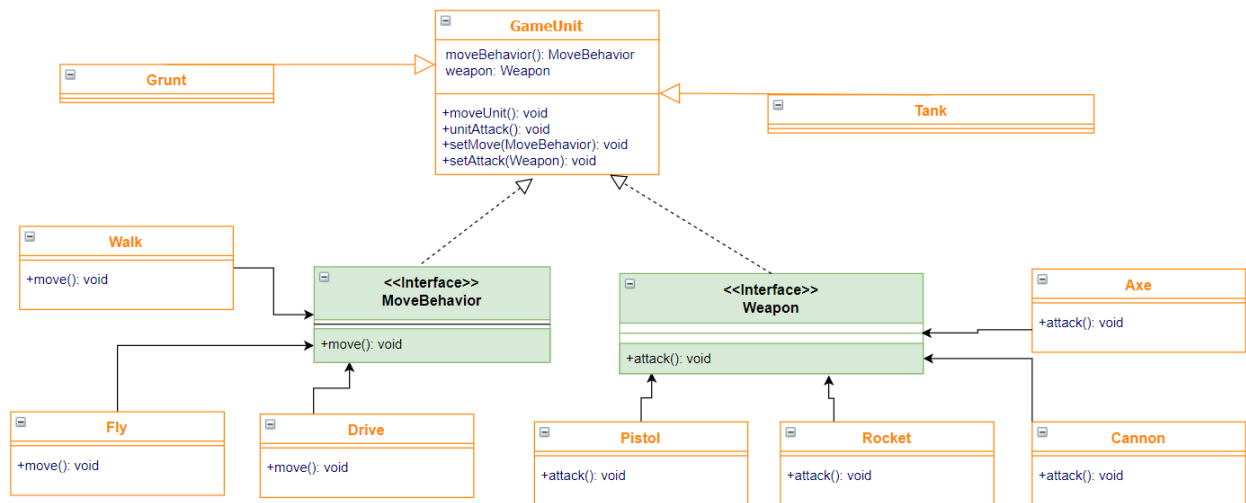
Exercise 2

Assignment 1: Implementation is directly used in the classes. There is no interface, thus limiting this class design to a very specialized scenario of RTS. Could be more abstract to be more flexible.

Assignment 2: Implementation of moving and attacking are in constant flux, can be replaced with an interface to make it easier for code reuse.

Assignment 3: see the code: behavior change via using an interface ("interface" keyword) and overwriting interface methods via the "implements" keyword

Assignment 4:



Assignment 5:

Inheritance – The classes grunt and tank are classes that inherit from GameUnit.

Encapsulation – The interface MoveBehavior, and all of its child/derived class (walk, fly, drive) encapsulated the method move().

Polymorphism – using an object of class GameUnit, we reference and utilize methods of the derived classes; grunt and tank.

Assignment 6:

This interface strategy doesn't have the capability to test/check if a move behavior is "invalid" for its unit type. Same for the grunts, and attack behavior. Thus a tank type of unit can be set to "walk", and a grunt type unit can be set to "fly or drive" despite limitations existing.