

Notes on UML

L. Laneve, A. Lopardo, R. Mencucci

March 2019

1 Effect

An **Effect** represents the atomic element of a **Weapon**'s behavior. Effects implement a *strategy* pattern to represent the gameplay interaction associated with them.

Multiple atomic effects may be combined to produce more complex effects. Such composition takes place using the opportune collections **EffectChoiche**, **EffectSequence** and **EffectSet**.

Some weapons have peculiar functionality and interactions never used in any other part of gameplay. We have found inconvenient introducing many different atomic effects only to use them once. Our choiche is to introduce one peculiar effect for each one of these weapons and have it implement all the related functionality, regardless of the conceptual non-atomicity of it.

With these choiches, simple weapons like Heatseeker are implemented with atomic effects alone, more complex weapons like Plasma Gun are implemented with a composition of atomic effects and only a few problematic weapons like Vortex are implemented with their own peculiar effect.

2 Action

An **Action** expresses the interaction with a player that eventually produces one or more **GameChanges**. It can be thought of as the main engine for interacting with human players.

Actions are implemented using an *observer* pattern. They are part of the model, as they express fundamental mechanics strictly connected with the applicative domain of the application.

Actions use synchronus requests to interact with the player. This greatly simplifies the implementation of them and allows for a clearer representation of the real-game behaviour they express.