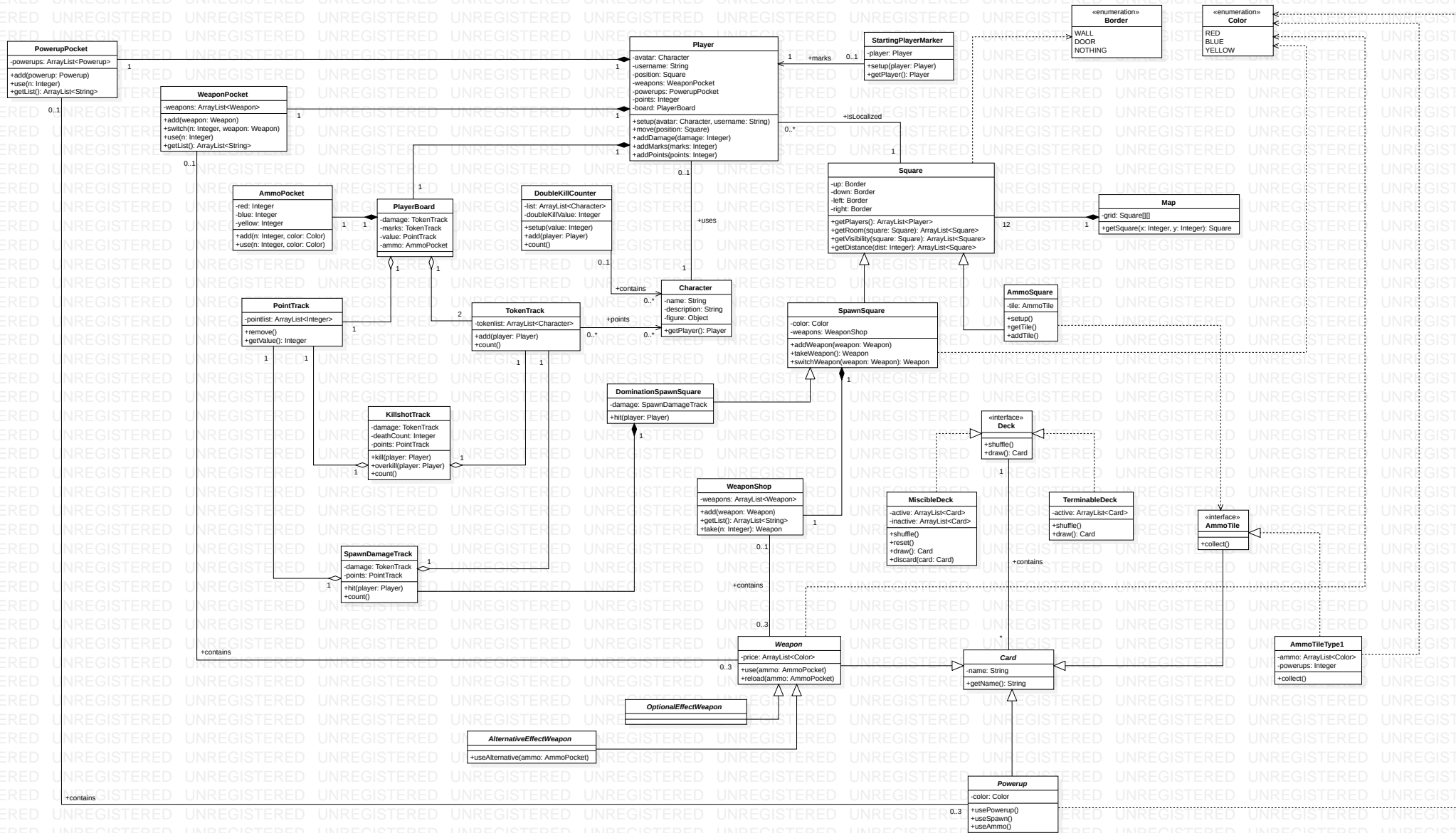


Model Map



NOTES

The Model (of MVC pattern) we created contains the data of the game and the methods that permit to use that data. Because of this decision some classes contains some method to use that object. The Controller will call this method when the player, using the View, will decide to use that object.

For example, the interface Weapon contains the method use(). When the player decides to shoot, the Controller will call the method use() of the Weapon, that is included in the Model.

We decided to exclude from the UML the 21 classes that represent the 21 different weapons to keep the UML schema simple. This classes will be named like the weapons (for example, VortexCannon, Thor...) and they will have the same methods and attributes described in OptionalEffectWeapon and AlternativeEffectWeapon (the weapons that have an alternative mode will extend AlternativeEffectWeapon, the weapons with optional effect will extend OptionalEffectWeapon).