# Projekt MonsterTradingCardGame

## Erstellung der Battle Logic

### Problem Spezialfälle:

Anfangs wollte habe ich zu sehr die Vergleiche zwischen Kartentypen nicht übergreifend gestaltet und eine Methode für MonsterGegenMonster entwickelt die in der abstrakten Basisklasse war. Diese Methode musste dann aber in jeder einzelnen Monstertyp Klasse implementiert werden=> Code Duplication.

Lösung: eine Basismethode in der Basisklasse die alle Grundfälle abdeckt und eine weitere Funktion in dieser die in der Kindklasse die Spezialfälle genau festlegt.

Die Vererbungshierachie wurde so ausgelegt, dass diese leicht erweiterbar ist für mögliche weitere Spezialfälle. Zunächst war klar, dass eine abstrakte Klasse die mögliche gemeinsame Methoden aller Karten implementieren soll angelegt werden muss. Hierbei wurde bereits im Bezug auf die Vorgehensweise beim Vergleichen der Karten die Methode adaptDamage() erstellt. Dessen Aufgabe besteht darin, Sonderfälle und BonusDamage im Kampf je nach Gegner anzupassen. Dessen Wert ist die Grundlage zum Vergleichen der Karten mit berücksichtigten Sonderfällen. Hier wurde der Ansatz gewählt eine Grundvariante davon in der abstrakten Klasse zu erstellen, die auch die Bonuseffekte der Spells berücksichtigt. Am Ende dieser Funktion müssen dann noch Spezialfälle wie Kraken gegen Spells(kein Effekt) berücksichtigt werden. Dafür gibt es eine eigene Methode TestSpecialCase() => diese prüft ob es noch Spezialfälle gibt die nicht berücksichtigt wurden. Hier habe ich mich für eine virtual Funktion in der abstrakten Klasse entschieden um die Basisversionen = kein Sonderfall zu behandeln. Gibt es nun ein Objekt dass seinen Damage anpassen muss, so muss diese Methode in der Kindklasse angepasst werden. Man könnte diese Funktion auch in der abstrakten Klasse der Monster hinzufügen, dass wäre aber unübersichtlicher, da man so in der Kindklasse für jeden Monstertyp seine Schwächen sofort erkennt und leicht erweitern kann.