**Plan tworzenia**

1. stworzyć mechanizm do zarządzania atrybutami jednostek
2. stworzyć bazę danych z wszystkimi jednostkami
3. stworzyć pole bitwy

**Wytyczne:**

1. Nazewnictwo klas: XAbstract,XImpl, X
2. Nazwy pakietów: api, impl, units, models, base

**TODO**

XXX – nie zrobione, XXX – do opisania, XXX – done!

1. Ogólnie
   1. każda klasa = walidator
2. Sprint 1: Attrybuty
   1. AbstractUnit
      1. dodać defaultowe implementacje updateValue x 2 oraz te referenced jako prywatna
   2. Klasa akcji: Action
      1. Typ: attack,
      2. Mapa: nazwa trybutu - zmiana
   3. **dodanie walidatorów**
   4. dziedziczenie - dodać 2 rodzaje atrybutów: Opisowy, Liczbowy, Procentowy
   5. dodanie enuma określającego typ jednostki
   6. dodanie zmniejszania poniżej 0% statystyk
   7. Modyfikatory powiązanych atrybutów (enumy) - mają być oddzielnymi plikami
   8. AttributeName - zamienić na enum AttributeDescription: domain, name
   9. **pogrupowanie atrybutów w aspekty**
      1. **aspekt = zbiór atrybutów (np. aspekt = hp: obecne i max)**
   10. **klasa – mechanizm do zmiany wartości atrybutów**
3. Sprint 2: Buffy
   1. Stworzenie języka do zapisu buffów
   2. Stworzenie dynamicznego mechanizmu do buffów
   3. usunąć podział na pozytyne negatywne
   4. stworzyć uniwersalny obiekt Action, który będzie opisywał oddziałowanie między jednotkami na polu bitwy,
4. Stworzenie propertisów i ich kodów

* **Przemyśleć**
  + Celność - co z nią zrobić?
  + 3 tryby akcji na polu bitwy:
    - Pass
    - Ofensywny – szansa na zadanie obrażeń krytycznych,
    - Defensywny – większy % na uniknięcie obrażeń + mały % do pancerza
* **Do zrobienia**
  + Model *Akcji:*
    - AttributeChange
    - [] Buff
  + Pętla - Mechanizm atakowania
    - Zapisz id obecnie wybranej jednostki
    - Wybierz jednostkę do zaatakowania
    - Zapisz jej id
    - Wyślij atakującemu #ref do jednostki
    - Wyczyść stan gry