**Plan tworzenia**

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

**Wytyczne:**

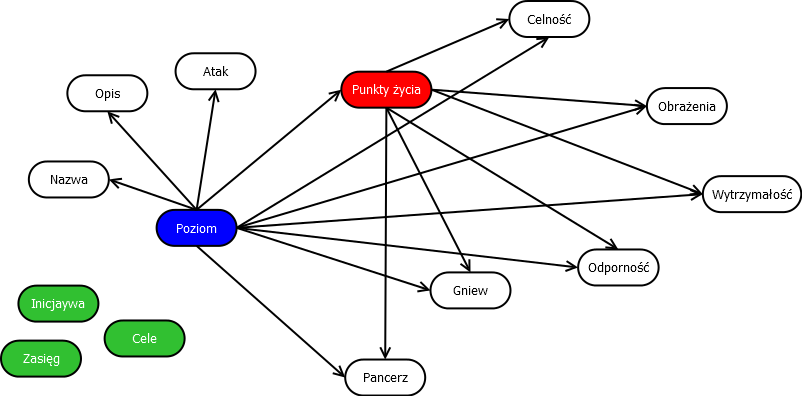
1. Nazewnictwo klas: XAbstract,XImpl, X
2. Nazwy pakietów: api, impl, units, models, base

**TODO**

1. Ogólnie
   1. każda klasa = walidator
2. Sprint 1: Attrybuty
   1. Klasa akcji: Action
      1. Typ: attack,
      2. Mapa: nazwa trybutu - zmiana
   2. Dodać metody do klas atrybutów, które przeliczają zmianę wartości atrybutu na procenty
      1. Np. zmiana hp = 100 > 50 = zmiana -50%, obrażenia spadają o 0.75 na jednostę, więc spadek obrażeń = 50% \* 0,75
   3. dziedziczenie - dodać 2 rodzaje atrybutów: Opisowy, Liczbowy, Procentowy
   4. dodanie enuma określającego typ jednostki
   5. dodanie zmniejszania poniżej 0% statystyk
   6. Modyfikatory powiązanych atrybutów (enumy) - mają być oddzielnymi plikami
   7. AttributeName - zamienić na enum AttributeDescription: domain, name
   8. **pogrupowanie atrybutów w aspekty**
      1. **aspekt = zbiór atrybutów (np. aspekt = hp: obecne i max)**
   9. **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

### Atrybuty

|  |  |  |
| --- | --- | --- |
| **Atrybut** | **Klasa** | **Kod** |
| **Nazwa** | UnitName | name |
| **Opis** | UnitDescrption | description |
| **Doświadczenie** |  | currentXp nextLvlXp |
| **Poziom** |  | level |
| **Rodzaj** | UnitType | type  W – wojownik, R – strzelec, S – wsparcie, M – Mag,  **uwaga**: nie jest wczytywany z properties,  a wyznaczany na pods. *zasięgu* oraz *liczby celi* |
| **Punkty życia** | UnitHitPoints | hitPoints |
| **Max punkty zycia** | UnitMaxHitPoints | maxHitPoints |
| **Odporność pancerza** | UnitArmorDurability | armorDurability określa liczbę punktów obrażeń, które moż przyjąć pancerz po których jego ochrona obniży się o 1% |
| **Odporność** | UnirResistance**X** | resistance**X**  **X** = Fire, Water, Air, Death, Poison, Mind, Life, Earth, |
| **Atak** |  | source Fire, Water, Air, Death, Poison, Mind, Meele, Projectile, Life, Earth  **uwaga**: MEELE = PROJECTILE, rozróżnienie istnieje tylko dla sprawdzania *pancerza* |
| **Celność** |  | accuracy |
| **Obrażenia** | UnitDamage | damage |
| **Inicjatywa** |  | initiative |
| **Zasięg** |  | range |
| **Cele** |  | targets |
| **Ruch** |  | stamina |
| **„Gniew”** |  | specialPoints |



### Modyfikatory powiązanych atrybutów

* implementują interfejs LinkedAttributesChange
* każdy z nich przechowuje mapę: atrybut – zamiana w postaci wpółczynnika double

### Modyfikatory wartości atrybutów:

* całkowity: integer
* procentowy: double
* **Przemyśleć**
  + Pancerz:
    - ~~Naturalny i nabyty?~~
    - Czy zadawanie obrażeń = zmiana w pancerzu i odpornościach (woda, ziemia, powietrze, ogień + pominięcie odporności z danej rasy)?
    - Nisczenie pancerza? Nieodwracalne?
  + 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