## Atrybuty

### Typy atrybutów

* proste – są zwykłym „pojemnikiem” na dane, nie potrzebują żadnych dodatkowych funkcjonalności związanych z ich danymi,
  + przykład: *nazwa jednostki*
* złożone – posiadają atrybuty powiązane, na które wpływają podczas zmiany wartości, są mikro – serwisami do zarządzania swoimi danymi,
  + przykład: *wytrzymałość pancerza,*

### Przechowywane dane

* wartość: liczbową lub tekstową
* identyfikator (id):
  + nazwa,
  + domena – nazwę grupy atrybutów do której należy,

### Opis atrybutów

#### Odporność na broń białą

#### Odporność na pociski

#### Odporność na żywioły

* ogólna odporność na każdy z żywiołów,
* każda jednostka posiada wrodzoną odporność n wybrany żywioł, której nie można zmienić
  + jej wartość jest stała = 75%
* odporności na pozostałe żywioły są rozwiązane jako umiejętności pasywne / buffy,
* odporność może być dodania lub ujemna (wrażliwość na określony rodzaj magii)

#### Odporność umysłowa

* im większy jest jej współczynnik, tym:
  + mniejsza szansa na rzucenie klątwy lub negatywnego buffa na wybraną jednostkę,
  + negatywne zaklęcia mają mniejszą siłę oddziaływania,
  + trwają krócej,

#### Obrażenia

* mają swoje źródło, których może być kilka,
* im bardziej doświadczona jednostka, tym zakres obrażeń jest mniejszy, np.
  + lvl 1: <25;50>
  + lvl2: <65;75>
* wyjątkiem są magowie, których obrażenia obszarowe sprawiają, że zakres maleje wolniej niż w przypadku innych jednostek,

#### Rasa jednostki

* I – Imperium, E – Elfy, U – Nieumarli, L – Legiony, C - klany

#### Rodzaj jednostki

* W – wojownik, R – strzelec, S – wsparcie, M – Mag,

#### Wytrzymałość pancerza

* Jego procentowa wytrzymałość wpływa na wszystkie odporności

#### Źródło obrażeń

* Fire, Water, Air, Death, Poison, Mind, Meele, Projectile, Life, Earth

### Dodatkowe informacje

* **Wczytywanie**: Cały proces odbywa się w klasie *UnitAttrbutesCreator*

|  |  |  |
| --- | --- | --- |
| **Atrybut** | **Klasa** | **Kod - properties** |
| Nazwa | SimpleTextAttribute | description.name |
| Rasa | SimpleTextAttribute | description.race |
| Notka | SimpleTextAttribute | description.note |
| Rodzaj | SimpleTextAttribute | description.type |
| Liczba punktów doświadczenia |  | progress.currentXp |
| Punkty doświadczenia  do następnego poziomu |  | progress.nextLvlXp |
| Poziom |  | progress.level |
| Punkty życia | UnitHitPoints | resistance.hitPoints |
| ~~Max punkty życia~~ | ~~UnitMaxHitPoints~~ | ~~hitPoints.max~~ |
| Pancerz (wytrzymałość) | UnitArmorDurability | resistance.armorDurability |
| Odporność na broń białą | UnitMeeleResistance | resistance.meele |
| Odporność na pociski | UnitProjectileResistance | resistance.projectile |
| Odporność na żywioły | UnitElementalResistance | resistance.elemental |
| Odporność umysłowa | UnitMindResistance | resistance.mind |
| Atak |  | attack.sourceX |
| Celność |  | attack.accuracy |
| Obrażenia | UnitDamage | attack.damage |
| Inicjatywa |  | attack.initiative |
| Zasięg |  | attack.range |
| Cele |  | attack.targets |
| Ruch |  | action.stamina |
| „Gniew” |  | action.specialPoints |

## Modyfikatory wartości atrybutów

* implementacja: *AttrbuteValueChangeFactorImpl*
* jako dane wejściowe są opisywane przez liczby
  + **całkowite** np. 120 = dodać 120, -50 = odjąć 50
  + **zmiennoprzecinkowe**: 50.0 = +50%, zaś -33.0 = -33%
    - w programie są przechowywane jako liczby: pomiędzy (0;1)
    - w plikach z danymi są zapisywane w postaci X.0 np. 50.0 = 50%
* w jednym z konstruktorów tej klasy możliwe jest obliczenie współczynnika na podstawie dawnej i nowej wartości atrybutu
  + wykorzystywane w *Atrybutach powiązanych*

## Walidatory

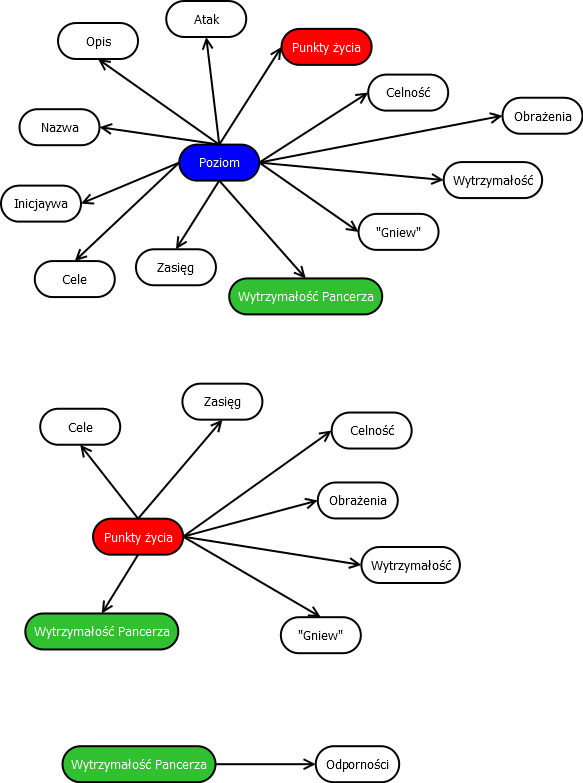
* każdy obiekt przechowujący dane powinien implementować interfejs *Validatable*

## Powiązane atrybuty

### Opis

* interfejs:*LinkedAttributes*
* istnieje tylko jedna implementacja *LinkedAttributesImpl, która* przechowuje mapę:
  + atrybut –>zamiana w postaci współczynnika *double*
  + współczynnik zmienno przecinkowy określa o ile procent zmiany wartości *atrybutu powiązanego* przypada na 1% zmiany *atrybutu głównego*
    - np. w atrybucie punktów życia 0.1 dla Pancerza oznacza, że np. przy obniżeniu o 1% Punktów życia należy obniżyć 0.1% wartości pancerza,
* wczytywane z pliku z danymi jednostki
  + zapisywane są jako obiekt o stałej nazwie *„attribute.links”*
  + obiekt ten jest złożony z pod-obiektów, których nazwy odpowiadają kodom atrybutów
  + każdy z tych pod-obiektów posiada listę atrybutów powiązanych wraz z współczynnikami zmiany,
* przy zmniejszaniu wartości atrybutów powiązanych wymagana jest znajomość zmiany *atrybutu głównego*, tak aby można było odpowiednio przeskalować zmianę na *atrybut powiązany*
* **proces**
  + zmiany odbywa się w metodach [*atrybut*]*.updateReferencedAttributes*
  + przebieg:
    - obliczany jest *współczynnik zmiany* wartości atrybutu powiązanego w zależności od:
      * jego współczynnika zmiany
      * zmiany wartości atrybutu głównego
    - pobierana jest obecna wartość atrybutu powiązanego
    - jego wartość jest aktualizowana przed *współczynnik zmiany* (zawsze procentowy)

### Powiązania



**(2)**

**(1)**

**(2)**

**(1)**

#### Objaśnienia:

* (1) – tylko dla jednostek zadających obrażenia obszarowe
* (2) – tylko dla jednostek dystansowych

## Zmiana wartości atrybutów – proces

* Dane wejściowe:
  + HP = 200, zmiana = -100
* obliczanie współczynnika *zmiany atrybutu głównego*
  + 0.5 (-50%)
* pobranie nazwy atrybutu powiązanego i jego współczynnika zmiany:
  + atrybut = DAMAGE, zmiana = 0.75
* obliczenie procentowej zmiany dla *atrybutu powiązanego*
  + 0.5 \* 0.75 = 0,375
* zamiana wartości *atrybutu powiązanego* w zależności od współczynnika zmiany
  + wartość atrybutu DAMAGE = 75
  + zmiana: 1.0 - 0.375 = 0.625
  + nowa wartość: 75 \* 0.625 = 46,87
* Wynik działania
  + **nowa wartość:** 46,87