## Analiza

### Mocne strony

|  |  |
| --- | --- |
| ***Mocne strony*** | ***Rozwiązania*** |
| * Rozwijanie indywidualnych jednostek   + zdobywanie poziomów doświadczenia   + zmiana wyglądu   + nowe umiejętności | * Używanie umiejętności aktywnych  = zwiększanie ich poziomu,   + każdy poziom ewolucji jednostki określa max. poziom, do którego dana umiejętność może być rozwinięta   + rozwój umiejętności jest skończony  i zależy od ostatniej jednostki  w drzewku, * Co poziom wybieramy tylko stałe bonusy do określonych statystyk, które będą je zwiększać o poziom,   + np. +5% do HP = każdy poziom to dodatkowe 5% zwiększenia HP * Zdobywanie poziomów w czasie bitwy   + może wówczas wykonać dodatkowy ruch/atak |
| * Drzewko rozwoju jednostek | * Dodanie dodatkowego budynku (poziomu rozwoju) dla każdej ścieżki rozwoju * \* Możliwość posiadania każdego z budynków |
| * Małe umiejętności w czasie bitwy |  |
| * System przejmowania   + terenu i zasobów,   + budynków,   + plądrowania, |  |
| * Klimat   + dojrzały,   + surowy,   + bez upiększeń, |  |
| * Myślenie strategiczne, perspektywiczne |  |

### Słabe strony

|  |  |
| --- | --- |
| ***Słabe strony*** | ***Rozwiązania*** |
| * Jedna lub dwie mocne drużyny, reszta to mięso armatnie | * jednostki w miastach mogą się szkolić za kasę i pod warunkiem wybudowania odpowiedniego budynku w stolicy, * odradzanie się coraz silniejszych potworów, |
| * Brak możliwości poruszania się po polu bitwy |  |
| * Nuda na mapach | * Zdarzenia losowe |
| * Rozwój bohaterów |  |
| * Przejmowanie zasobów jest zbyt proste | * System strażników |
| * Zdobywanie zamków/miast jest zbyt mało przemyślane |  |

### Pomysły

|  |  |
| --- | --- |
| ***WAŻNE*** | ***dodatki*** |
| * Miejsce na polu bitwy gdzie kilka jednostek na raz będzie mogło odpocząć, zregenerować siły, uzupełnić manę, naprawić pancerz, * Bonusy do walki dla jednostek z określonymi rodzajami jednostek * Obóz szkoleniowy - budynek * Miasta generują mały przychód,  dają dodatkowe bonusy do obrażeń, pancerza i życia * Każdej akcji na polu bitwy towarzyszy odgłos np. awansowi, zabicia, umierania, * Sabotowanie pracy w kopalni | * Im większa twoja przewaga tym więcej ziemi na polu bitwy sie, zmienia, * Animacja wchodzenia oddziału na pole bitwy, dowódcy się spotykają, * Animowane pole bitwy  z randomowymi efektami, * Stolice z animowanymi dzielnicami każda od innego rodzaju jednostek  i efektami: animacjami dźwiękami, |

Wladca - obnizenie max zycia o 5% atak  
Zamet - 50% obrazen najsilniejszej jednostce, pojedynczy atak  
Elitarny zabójca - dekapitacja jezeli hp mniej od 15% = 100% szans trafienia  
Kamuflaz unika 1 ataku orezem  
-15% szans na trafienie, ochrona umysl, trucizny?  
Mag - Zmniejszenie obszaru ataku  
Zabójca magów  
Zmiana źródła dmg  
Tarcza - dodaje pancerza do poki sie nie zepsuje  
Kościana zbroja - dodaje pancerz wybranej jednostce

Umiejetnosc - mechanizm - zwrotny

Magowie - medytacja jako tryb ruchu, regeneracja many,   
Wojownicy maja to samo, ale regeneruje sie im życie  
  
Druzyny na,polu bitwy, kazda ma kapitana, przemieszczanie sie kazda,z druzyn, szyki bojowe, im mniej osob w druzynie tym mniejsze bonusy

Magowie – słabi, ich głównym zadaniem jest rzucanie czarów, ataki mało obszarowe zależne (jak i zaklęcia) od many

**WERSJE**

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

**Wytyczne:**

1. Nazewnictwo klas: AbstractX, XImpl, X (interfejs)
2. Nazwy pakietów:
   1. units
      1. api,
      2. impl,
         1. models,
         2. base
3. Baza danych
   1. pliki mapowania xml -> resources
   2. klasy pojo: XModel

**TODO**

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

1. SPRINT: Baza danych i jej obsługa
   1. Podpięcie SPRINGA
   2. Stworzenie bazy danych z informacjami o jednostkach
   3. Stworzenie modeli
   4. uzupełnienie bazy danych

// Setup - tabele z domyślnymi wartościami

UNITS\_SETUP

Id, id\_description, id\_progress, id\_resistances, id\_attack, id\_movment

UNITS\_SETUP\_DESCRIPTION

name, note, race, type

UNITS\_SETUP\_PROGRESS

currentXp, nextLvlXp, level, id\_progress\_tree

UNITS\_SETUP\_RESISTANCES

hitPoints, meele, meele, projectile, elemental, mind

UNITS\_SETUP\_ATTACK

source, accuracy, damage, initiative, range, targets

UNITS\_SETUP\_MOVEMENT

stamina, specialPoints

UNITS\_SETUP\_PROGRESS\_TREE

id\_unit, id\_unit\_next\_stage

1. SPRINT: Atrybuty
   1. Refaktor – AbstractAttribute/public void update( - wywalic
   2. **Zapisywanie jednostek w bazie danych**
   3. Zapisywanie wartości atrybutów liczbowych jako zmiennoprzecinkowe
      1. zmniejszy to liczbę błędów w przyszłości,
   4. dodanie walidatorów
   5. odporności
      1. podpięcie generatorów zmian wartości atrybutów powiązanych do:
         1. procesu wczytywania danych
         2. Klas atrybutów prostych
      2. usunięcie wytrzymałosci pancerza >zmiany w spec, grafach,
   6. wartości
      1. sprawdzanie wartości przed ustawieniem
      2. dodanie zmniejszania poniżej 0% statystyk
      3. każdej jednostce zapisanej w bazie dać 200%HP
      4. stałe współczynniki przyrostu HP: wojownicy +50, magowie+30, strzelcy +40,
   7. Refaktor - rodzaje atrybutów:
      1. Opisowy,
      2. Liczbowy,
      3. Procentowy,
   8. wczytywanie domyślnych wartości współczynników zmiany
      1. **loadAdditionalData > zrobic kopiowanie atrybutu hitPoints dla maxHp**
      2. **~~dodanie pliku json opisującego wszystkie zależności między atrybutami~~**
2. SPRINT: Levelowanie jednostek
   1. Przy nowym poziomie powinny się zmniać wspołczynniki zmiany
3. SPRINT: Buffy / um. Pasywne
   1. Stworzenie języka do zapisu buffów
   2. Stworzenie dynamicznego mechanizmu do buffów
   3. usunąć podział na pozytyne negatyw’’ne
   4. stworzyć uniwersalny obiekt Action, który będzie opisywał oddziałowanie między jednotkami na polu bitwy,
   5. dodanie defaultowych buffów
      1. odporności wrodzone (np. dla nieumarłych buffem będzie odpornosc na zywioly, smierc – zawsze 75%) + sprawdzanie
4. SPRINT: Akcje – Atak
   1. Przebieg
      1. trafi / nie trafi
      2. sprawdź buffy chroniące
      3. zadaj obrażenia
      4. sprawdź życie jednostki
      5. nałóż buffy - kary
   2. Klasa akcji: Action
      1. Typ: attack,
      2. Mapa: nazwa trybutu – zmiana
5. SPRINT: Akcje - Ruch
   1. Przebieg
      1. sprawdź buffy „dobre”
      2. ... „złe”
      3. zaznacz jednostkę,
      4. wykonaj ruch,

* **Przemyśleć**
  + Po co dzielić jednostki na typy? Co daje taki podzial?
  + Celność - co z nią zrobić?
    - Zamienić na szansę na blok/unik/zasłonę,
  + ~~Ciosy: szybki (mniejsze szanse na zasłonę/użycie um. pasywnych) i ciężki (szansa na zadanie obrażeń krytycznych)~~
  + ~~3 tryby akcji na polu bitwy:~~
    - ~~Pass~~
    - ~~Ofensywny – szansa na zadanie obrażeń krytycznych / kontratak,~~
    - ~~Defensywny – większy % na uniknięcie obrażeń + mały % do pancerza~~
  + TURA: 60 sek na obmyślenie, po 10 sek na ruch
* **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