

# WebRPG

## Przemysław Królik

Platforma ułatwiająca prowadzenie rozgrywek RPG poprzez WWW

Stworzona przy użyciu node.js i html5

### 1. Wymagania, konfiguracja i uruchomienie

- 1.1. Wymagania.
- 1.2. Użyte technologie/elementy/moduły/etc.
- 1.3. Przebieg konfiguracji
- 1.4. Pierwsze uruchomienie

### 2. Przebieg działania

- 2.1. Od czego zacząć?
- 2.2. Stworzony pokój
- 2.3. Co dalej?

### 3. Plany na przyszłość

- 3.1. Czego nie ma a co ma być
- 3.2. Co miało być a nie ma

## **1. Wymagania, konfiguracja i uruchomienie**

### **1.1. Wymagania**

Podstawowe wymagania odnośnie uruchomienia aplikacji:

1. Środowisko node.js w wersji 0.10
2. Menadżer npm
3. Nierelacyjna baza danych mongodb
4. Akceptacja "Ciasteczek"
5. Uruchamiania kodu javascript po stronie klienta
6. Przeglądarka internetowa obsługująca HTML5 oraz websocket

### **1.2. Użyte technologie/elementy/moduły/etc.**

node.js, socket.io, cookies, parse, mongoose, canvas, javascript Class

### **1.3. Przebieg konfiguracji**

1. Po ściągnięciu wszystkich plików należy zaktualizować zależności poprzez *npm update*.
2. Przechodzimy do katalogu model i kopiujemy plik modelsSchema.js nazywając kopie models.js lub zmieniamy nazwę plików na models.js
3. Edytujemy plik models.js podając ścieżkę połączenia się do bazy danych mongodb

### **1.3. Pierwsze uruchomienie**

1. W głównym katalogu wpisujemy node app.js i gotowe :)

## 2. Przebieg działania

### 2.1. Od czego zacząć?

1. Wpisujemy w oknie przeglądarki localhost:3000/game/create
2. Podajemy nick admina pokoju
3. Wpisujemy id pokoju
4. Po kliknięciu przycisku strona przenosi nas na odpowiedni pokój prosząc o podanie nicka który zostanie zapisany w cookies przeglądarki
5. Należy zwrócić uwagę na adres URL znajdujący się w przeglądarce, końcówką (GET) wskazuje na ?id=xxx gdzie xxx to nr pokoju.
6. Każdy stworzony pokój zostaje umieszczony automatycznie w bazie danych.
7. Tworzony pokój ma predefiniowany standard jednakże może być modyfikowany, aby to uczynić wystarczy wg schematu sample.json oraz models.js w folderze model wytworzyć dokument w kolekcji bazy danych lub zmodyfikować plik serwera.
8. Po załadowaniu widzimy planszę stworzoną z pliku.

### 2.2. Stworzony pokój.

1. Jeżeli użytkownik poda nick administratora zostanie mu automatycznie przydzielona rola administratora pokoju.
2. Administrator nie może brać czynnego udziału w rozgrywce, tj nie ma wpływu na postacie znajdujące się na planszy nie licząc faktu:
  - a. Po zalogowaniu się nowego użytkownika administrator musi przydzielić mu współrzędne na planszy na których to będzie umieszczony
  - b. Administrator decyduje który z graczy ma wykonać obecnie ruch wpisując id danego gracza i ilość przemieszczeń na mapie w danych rubrykach
  - c. Po wypełnieniu powyższego pkt administrator musi czekać na zakończenie ruchu gracza
3. Każdy z użytkowników ma możliwość rzutu kostką n-ścienną a wynik będzie zapisany w czacie
4. Jeżeli użytkownik otrzyma możliwość ruchu po kliknięciu na planszę będzie miał możliwość wykonania ruchu na dane pole

5. Każda z osób w pokoju może dowolnie przeglądać planszę poprzez klikanie przycisków-strzałek

### **2.3. Co dalej?**

Obecnie gra fanatazji i wyobraźni z przyszłościowymi pomysłami :)

### **3. Plany na przyszłość**

#### **3.1. Czego nie ma a co ma być**

1. Logowanie użytkowników
2. Większe ilości zabezpieczeń
3. Możliwość wyświetlenia użytkowników w pokoju
4. Otwarty świat dla wszystkich graczy
5. Rozróżnianie kont
6. Wiele, wiele innych

#### **3.2. Co miało być a nie ma**

1. Stworzenie potworów które przemieszczałyby się przy pomocy administratora (do wykonania na poziomie kodu ruchów użytkownika)
2. Stworzenie przedmiotów/opisów/itp na każdym osobnym polu (do wykonania na poziomie "zablokowanego" pola)
3. Zapis punktów życia każdego użytkownika (do wykonania na poziomie bazy danych i serwera)