

**Wydział Matematyki i Nauk Informacyjnych
Politechnika Warszawska**

**Od HTMLa do PostGISa
(projekt grupowy)**

Dokumentacja projektu

Autorzy:
Konrad Komisarczyk, Jakub Kosterna,
Mariusz Słapek, Patryk Wrona

Prowadzący:
Maciej Bednarz

Warszawa, 2020-2021

Spis treści

1. Wstęp	2
1.1. Ogólna charakterystyka projektu	2
1.2. Cel projektu	2
1.3. Opis biznesowy	2
2. Założenia projektowe	3
2.1. Użyte technologie	3
2.2. Uniwersum gry	3
2.2.1. Fabuła	3
2.2.2. Podstawowe elementy świata gry	3
2.3. Udostępniane endpointy	4
2.4. Tabele w bazie danych	5
2.5. Diagramy klas	6
2.6. Historyjki użytkownika	7
2.7. Architektura systemu	8
3. Wymagania funkcjonalne	9
4. Wymagania niefunkcjonalne	10
5. Tutorial	11

1. Wstęp

1.1. Ogólna charakterystyka projektu

Przedmiotem projektu jest stworzenie gry rozszerzonej rzeczywistości typu RPG. Gra ma być dostępna dla wielu użytkowników oraz wykorzystywać lokalizację użytkownika i specyfikę jego otoczenia.

1.2. Cel projektu

Celem projektu jest budowa serwera gry oraz interfejsu użytkownika. Na serwerze odbywać się będzie komunikacja z bazą danych użytkowników oraz na bieżąco będą generowane nowe obiekty, z którymi gracze będą mogli wchodzić w interakcje. Interfejs użytkownika dostępny będzie przez przeglądarkę internetową w dowolnym urządzeniu pozwalającym na udostępnienie lokalizacji.

1.3. Opis biznesowy

Kilka lat temu ogromny sukces komercyjny odniósła gra "Pokémon GO". Mniejszy sukces odniósły takie gry jak "Ingress" oraz "Harry Potter: Wizards Unite". Wszystkie te gry zostały stworzone przez Niantic Studios. Są to jedyne rozpoznawalne gry tego typu, które są sfabularyzowane oraz pozwalają na interakcję między użytkownikami. Zatem, rynek jest rozwijający się. Odmiennością naszej gry będzie to, że będzie dostępna ona przez przeglądarkę internetową, nie będzie wymagała ona instalacji aplikacji. Oferowała będzie także nową stylistykę oraz ciekawe rozwiązania z zakresu mechaniki gry.

2. Założenia projektowe

2.1. Użyte technologie

Serwer aplikacji zostanie napisany w języku Java, z wykorzystaniem framework'a Micronaut do obsługi requestów HTTP. Interfejs użytkownika stworzony zostanie z wykorzystaniem HTML, CSS i Javascript. Do wyświetlania mapy okolicy w interfejsie użytkownika oraz pobierania informacji o specyfice jego otoczenia wykorzystany zostanie projekt OpenStreetMap. Aplikacja komunikuje się z serwerem poprzez REST API, a dane składowane są w bazie danych.

2.2. Uniwersum gry

2.2.1. Fabuła

Rok 2021, naukowcy starają się stworzyć nowy rodzaj szczepionki, która zwiększyłaby odporność populacji w przypadku wybuchu nowej pandemii. Nowy eksperyment dotyczący sekwencji pewnego genu kierowany przez grupę badawczą Doktora Xenophiliusa Xhena zaczął wymykać się spod kontroli. Sam doktor zachorował na nieznaną chorobę, a ów gen DXX21 oznaczono jego iniacjalami. W obliczu tragedii, zaczęły powstawać nowe formy życia wskutek mutacji na poziomie genu DXX21. Lokalne władze są całkowicie bezradne i wprowadziły całkowity lockdown mieszkańców planety. Jedynie najwaleczniejsi z bohaterów zdecydowali się stanąć do walki z zagrożeniem, badając nowe istoty, zbierając o nich dane potrzebne naukowcom.

2.2.2. Podstawowe elementy świata gry

- Bohaterowie - każdy gracz wciela się w jednego bohatera. Bohater przemieszcza się po świecie zgodnie z lokalizacją urządzenia gracza. Bohater zbiera punkty doświadczenia, dzięki czemu może osiągać kolejne Poziomy i rozwijać Umiejętności Bojowe.
- Potworki - pojawiają się w losowych miejscach na świecie w pobliżu aktywnych Bohaterów. Bohaterowie mogą walczyć z nimi dzięki temu zbierając punkty doświadczenia, zdobywając Poziomy oraz Przedmioty. Istnieje wiele rodzajów Potworków, każdy pojawiający się w pobliżu innych rodzajów miejsc. Każdy rodzaj Potworka ma przypisane swoje Umiejętności Bojowe.
- Umiejętności Bojowe - wykorzystywane są w walkach z Potworkami i Pojedynkach. Gracze mogą rozwijać swoje postaci wykorzystując tzw. Punkty Doświadczenia, które zdobywają zwiększając umiejętności po zdobyciu kolejnego poziomu.
 - Health - liczba punktów życia, kiedy spadnie do 0, Bohater/Potworek ginie
 - Strength - zwiększa zadawane obrażenia
 - Defense - zmniejsza otrzymane obrażenia, w przypadku otrzymania ciosu
 - Speed - zwiększa szansę na uniknięcie ataku i zmniejsza szansę przeciwnika na uniknięcie ataku

2.3. Udostępniane endpointy

Na serwerze dostępne są odpowiednie endpointy obsługujący requesty HTTP użytkownika od strony aplikacji:

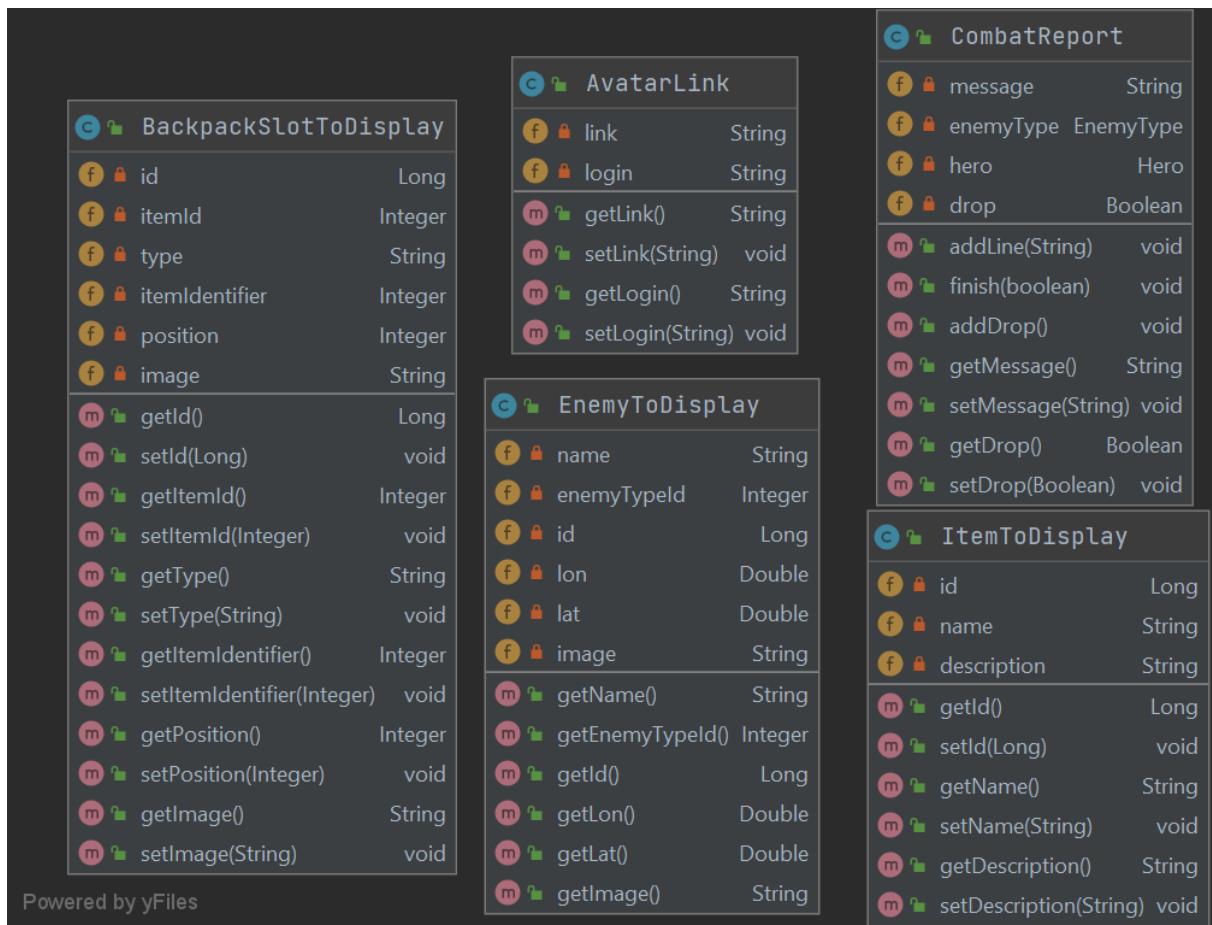
- **/generate** - Przyjmuje parametry login, lon oraz lat. Służy do generowania Potworków za pośrednictwem bazy danych Open Street Map. Wykorzystuje położenie użytkownika jako parametry i dodaje do bazy przeciwników o typach w zależności od poziomu gracza (na poziomie I jest ich 4, po zdobyciu doświadczenia i pokonaniu określonej liczby szkodników gracza na poziomie II może spotkać już 7 rodzajów, na III - 9, itd.). Wszystkie są tworzone w promieniu 1-2 km od geolokalizacji klienta, gdzie owa wartość jest zależna od konkretnego położenia na globie - np. bliżej równika będzie ona nieco inna niż na Zwrotniku Raka. Algorytm losujący zazwyczaj generuje 0-2 Potworki na każdy typ, ale matematycznie są możliwe także i większe ilości. Wszyscy przeciwnicy pojawiają się według określonych kryteriów Open Street Map, po znalezieniu punktów odpowiadających za dane obiekty. Oznacza to, że w parkach naturalnie przedzej będzie można spotkać szkodnika spawującego się w okolicach drzew, zaś na wsiach trudno będzie napotkać rzadkie Potworki wysokiego poziomu generujące się wokół instytucji państwowych.
- **/engage** - Przyjmuje parametry login, lon, lat oraz enemyId. Służy do inicjalizacji walki gracza o danym lon i lat z Potworkiem o zadanym enemyId. Użytkownik musi być w promieniu kilkunastu metrów (wartość ruchoma w zależności od szerokości i długości geofradycznej), aby móc stoczyć walkę z przeciwnikiem. Zwraca obiekt klasy CombatReport jako JSON.
- **/reload** - Przyjmuje parametry login, lon i lat. Służy do ładowania pobliskich Potworków z bazy danych, znajdujących się w pobliżu lon i lat należących do gracza. Użytkownik może fizycznie zobaczyć na mapie przeciwników w promieniu około 100-200 metrów - wartość ta waha się w zależności od lokalizacji użytkownika. Zwraca listę obiektów klasy EnemyToDisplay jako JSON.
- **/backpack** - Przyjmuje parametr login i zwraca Bohatera(Hero) o danym loginie jako obiekt JSON.
- **/backpack/itemDetails** - Przyjmuje parametry position i login. Zwraca przedmiot o danej pozycji w ekwipunku gracza powiązanego z danym loginem - jest to obiekt klasy ItemToDisplay zwracany jako JSON.
- **/backpack/consumeItem** - Przyjmuje parametry pos i login. Konsumuje przedmiot konsumpcyjny na danej pozycji w ekwipunku Bohatera o zadanym loginie.
- **/backpack/moveItem** - Przyjmuje parametry login, posBefore i posAfter. Zmienia pozycję przedmiotu z pozycji posBefore na pozycję posAfter. Całość wykonuje się w obrębie ekwipunku bohatera o zadanym loginie.
- **/backpack/deleteItem** - Przyjmuje parametry pos i login. Usuwa przedmiot na danej pozycji w ekwipunku Bohatera o zadanym loginie.
- **/player** - Przyjmuje parametr login, aby zwrócić Bohatera o zadanym loginie jako obiekt JSON.
- **/player/spendTalentPoint** - Przyjmuje parametry login i which. Zwiększa umiejętność Bohatera użytkownika o loginie login - jest to umiejętność wksazywana przez parametr which. Which może przyjmować wartości będącymi Umiejętnościami Bojowymi, czyli "strength", "defense", "health", "speed". Zwraca zmodyfikowanego Bohatera jako JSON.
- **/player/setAvatar** - Przyjmuje parametr JSON interpretowany jako obiekt klasy AvatarLink - zawiera on link do nowego avatara oraz login użytkownika, którego avatar chcemy zmienić.

2.4. Tabele w bazie danych

Hero	EnemyType	Item
f id Long f userId Long f health Integer f maxHealth Integer f strength Integer f defense Integer f speed Integer f avatarLink String f exp Integer f level Integer f fieldOfView Integer f talentPoints Integer f lastLon Double f lastLat Double	f id Long f name String f minUserLvl Integer f exp Integer f details String f health Integer f strength Integer f defense Integer f speed Integer f spawnKey String f spawnValue String f image String f spawnIntensity Double	f id Long f name String f type String f itemIdentifier Integer f damage Integer f protection Integer f healing Integer f description String f image String
		User
		f id Long f login String f password String
DropProbability	Enemy	BackpackSlot
f id Long f probability Double f enemyTypeId Integer f itemId Integer	f id Long f enemyTypeId Integer f lng Double f lat Double	f id Long f heroId Integer f itemId Integer f position Integer

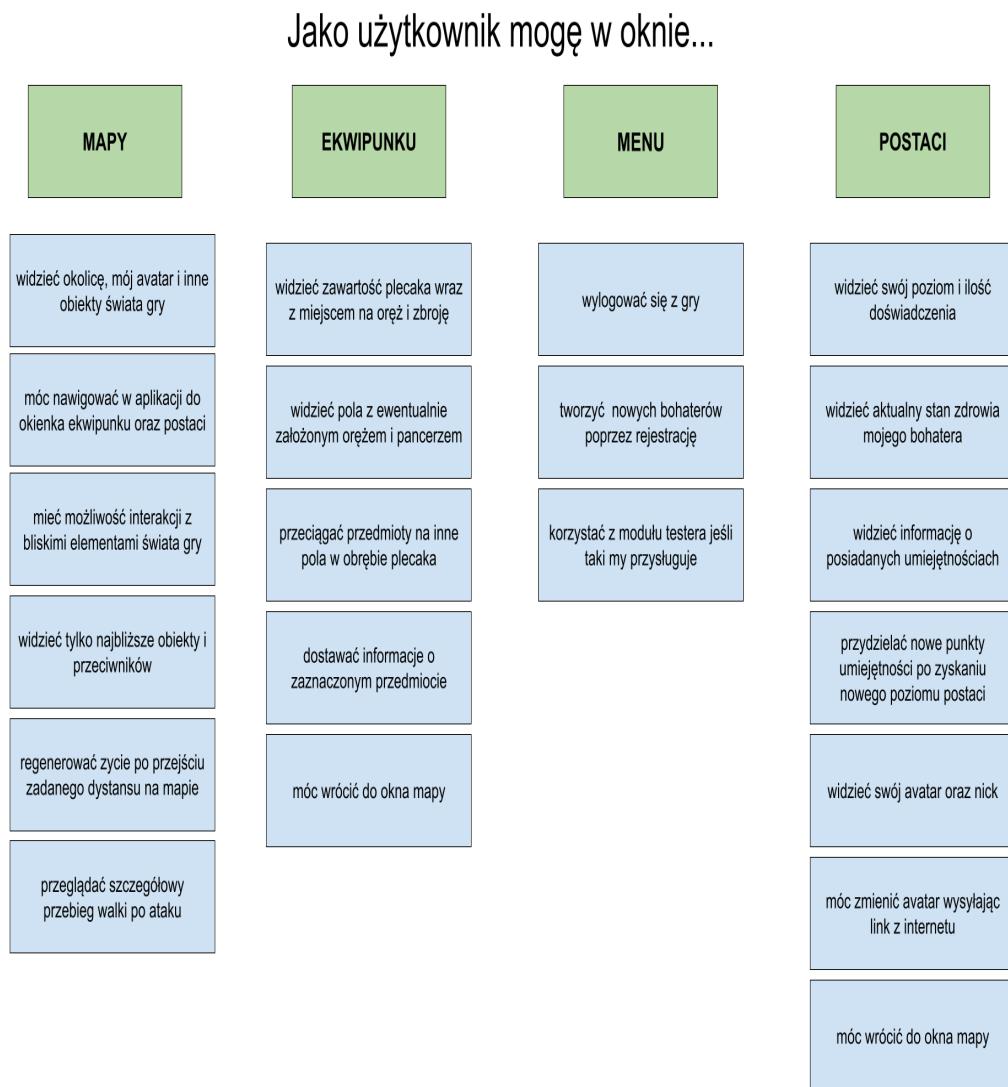
Rys. 2.1. Tabele w bazie danych

2.5. Diagramy klas



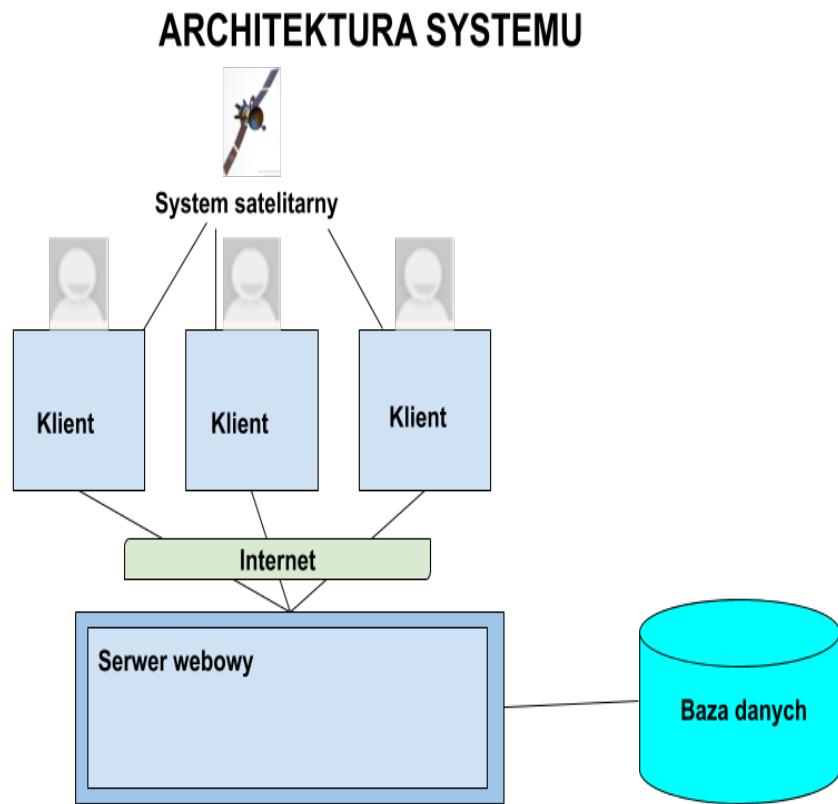
Rys. 2.2. Diagramy klas

2.6. Historyjki użytkownika



Rys. 2.3. User Stories

2.7. Architektura systemu



Rys. 2.4. Architektura systemu

3. Wymagania funkcjonalne

Program umożliwia następujące funkcje:

1. Widok mapy

- widoczna mapa okolicy i znajdujące się na niej obiekty
- pojawianie się okna informującego o możliwej interakcji, gdy bohater zbiży się do przeciwnika i na niego kliknie (przy zachowaniu odpowiednio małego dystansu)
- istoty pojawiają się gdy gracz jest blisko nich, po oddaleniu od miejsca istot, nie są one widoczne
- potwory pojawiają się w charakterystycznych dla nich miejscach
- wraz z przejściem dystansu kilkudziesięciu metrów graczowi jest regenerowany jeden punkt zdrowia (nie jest to wykonywane w wypadku zbyt szybkiego ruchu, tzw. oszuikiwania)
- przyciski przejścia do ekwipunku, widoku postaci oraz menu głównego

2. Widok menu

- przycisk do wylogowania
- panel z logiem gry
- miejsce do wpisywania kodów umożliwiających moduł testera
- tworzenie nowego bohatera

3. Ekwipunek

- miejsce na założoną broń i pancerz
- wizualizacja przedmiotów jako elementów w plecaku
- możliwość przeciągania przedmiotów w inne pola - całkowita dostępność nieużywanego ekwipunku: 20
- pole usuwania niepotrzebnego przedmiotu
- dodawanie się łupu po zakończeniu walki
- możliwość użycia przedmiotów typu "consumable" - regenerujących punkty życia
- przycisk powrotu do mapy

4. Widok postaci

- informacja o aktualnym poziomie i ilości doświadczenia
- komunikat o aktualnym życiu
- informacja o umiejętnościach
- przydzielanie punktów doświadczenia
- avatar postaci i jego zmiana
- przycisk powrotu do mapy

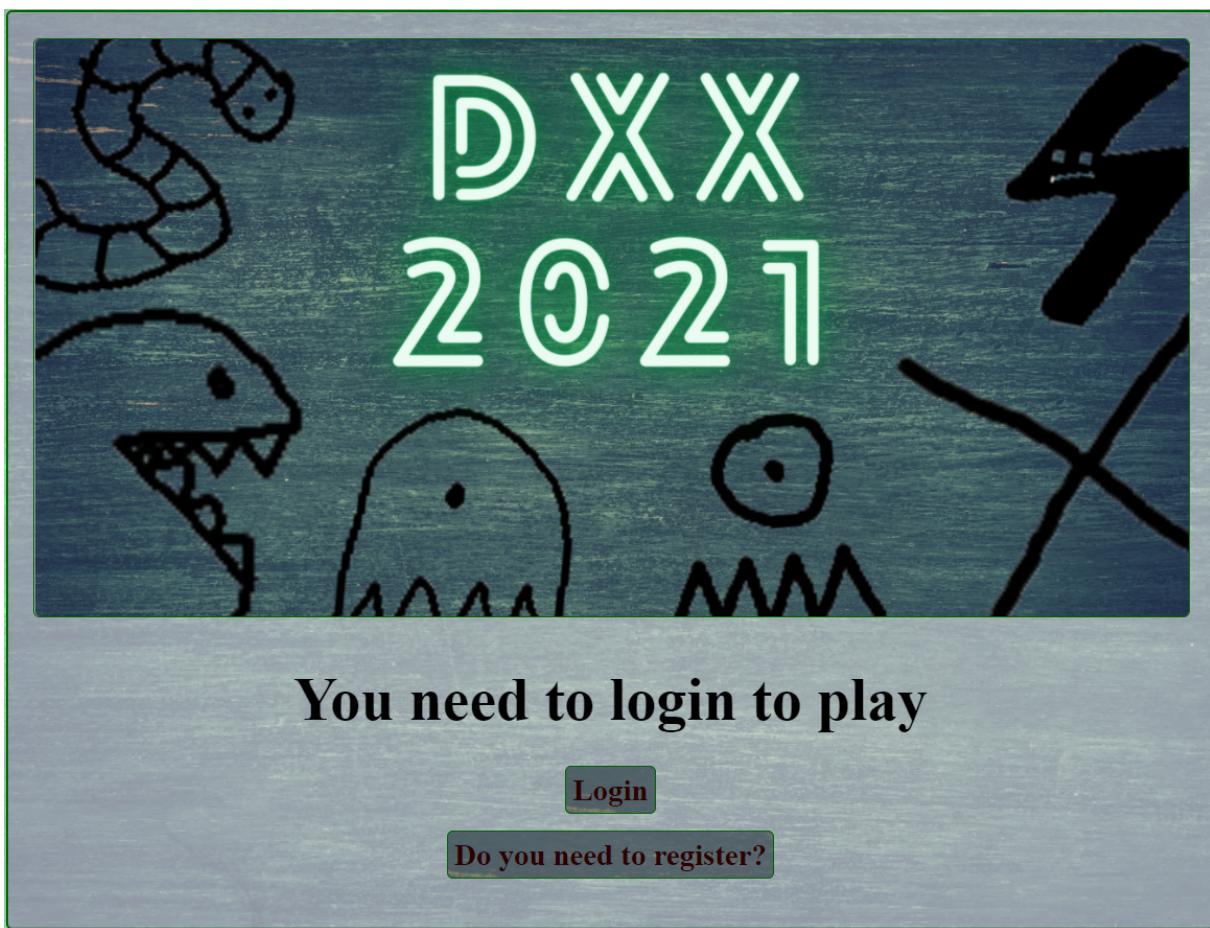
4. Wymagania niefunkcjonalne

1. aplikacja wymaga do działania *Java SE Runtime Environment 8*
2. program działa w przeglądarce internetowej obsługującej Javascript
3. program potrzebuje do działania dostęp do internetu
4. program łatwo rozszerzyć o dodatkowe funkcjonalności, np. rozbudowane systemy walki i rozwoju bohaterów
5. interfejs graficzny wygląda estetycznie na różnego rodzaju urządzeniach(komputer, tablet, smartfon)
6. aplikacja intuicyjna dla użytkownika
7. dostępność 24h na dobę
8. program odpowiedni dla dzieci
9. używanie aplikacji darmowe, możliwe płatne dodatkowe funkcjonalności
10. bezpieczeństwo danych użytkownika, hasła haszowane

5. Tutorial

Aplikacja jest bardzo instyntowna i prosta w użyciu nawet dla najmłodszych. Jej użytkowanie ogranicza się do przełączania między kilkoma ekranami gry.

1. Ekran początkowy



Tuż po otwarzeniu gry użytkownikowi wyświetla się widok powitalny. Widnieje na nim obrazek początkowy programu, a także informacja o konieczności zalogowania się do systemu. Nowy gracz bądź osoba chcąca zacząć od nowa może także założyć nowe konto.

2. Formularz rejestracji



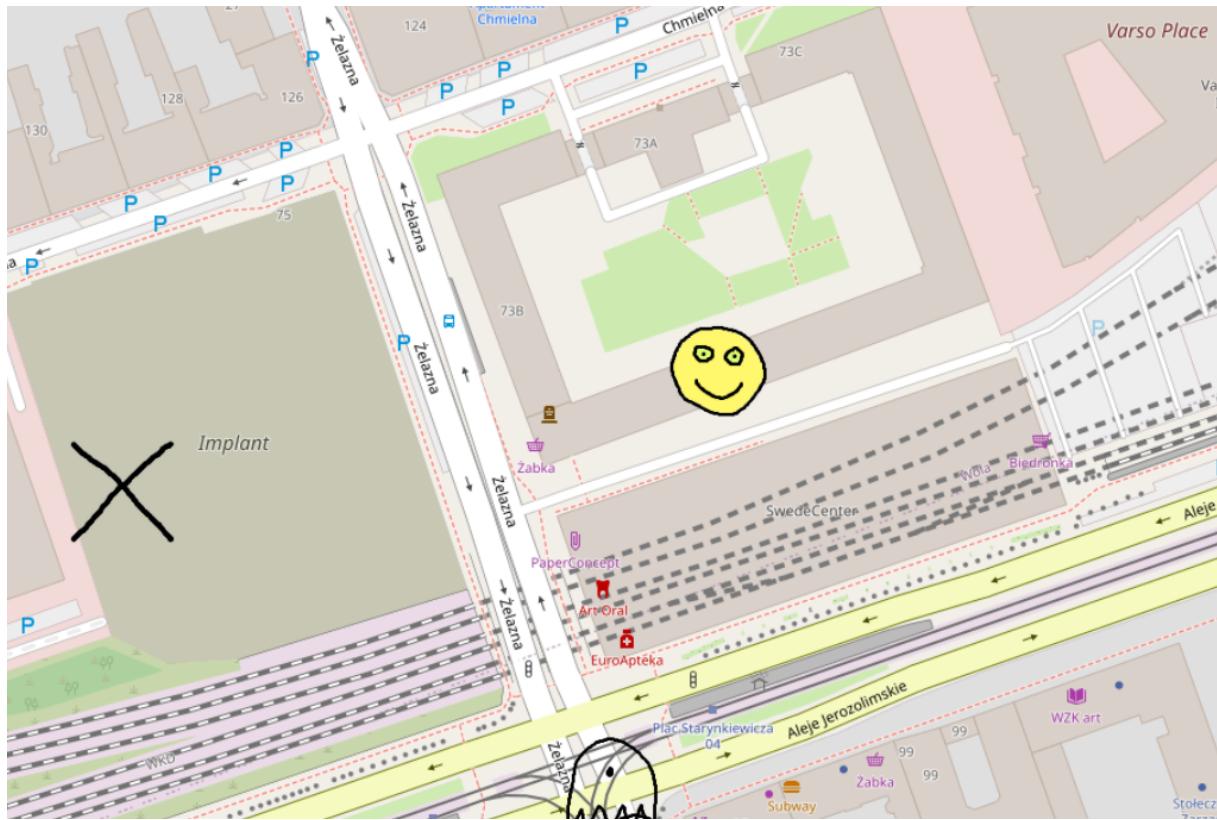
W przypadku wybrania opcji "zarejestruj", klientowi wyświetlają się dwa pola tekstowe - odpowiadające kolejno za nową nazwę użytkownika oraz hasło. Login musi być unikalny, zaś hasło jest przechowywane w postaci inteligentnego hashcodu. Po ukończeniu akcji gracz może kliknąć opcję "Register", która zapisze wpisane dane do systemu, bądź opcjonalnie - wrócić do głównego ekranu bez zakładania nowego konta.

3. Okno logowania



Użytkownik posiadający już konto loguje się do niego po kliknięciu opcji "Login" w ekranie głównym, a w efekcie dostaje formularz jak wyżej. Oprócz unikalnej nazwy użytkownika, musi on także wpisać hasło. W przypadku podania nieprawidłowego ciągu znaków wyświetla się stosowny komunikat i klient może podejmować kolejne próby bez ustanku.

4. Ekran główny gry



Po pomyślnym zalogowaniu, graczowi wyświetla się właściwa gra. W jej centrum zawsze znajduje się jego avatar - domyślnie uśmiechnięta buzia. Klient zlokalizowany jest w fizycznym miejscu urządzenia, a dana lokacja jest aktualizowana co sekundę. Z czasem pojawiają się także przeciwnicy - potworki. Gracz może na nie kliknąć, jeżeli zastanawia się czy podjąć z nimi walkę. Program jest przemyślany pod kątem ekranów o różnej rozdzielczości - niezależnie

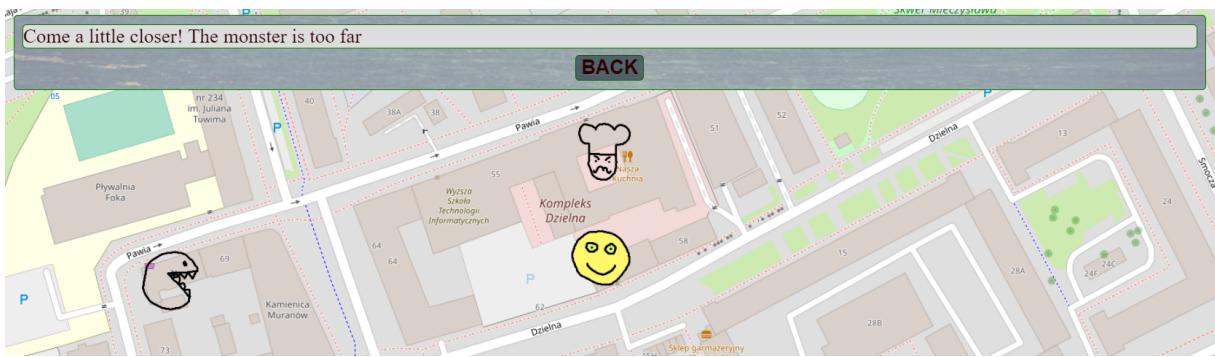
od urządzenia, promień generowanych i widocznych przeciwników jest stały.

5. Wejście w interakcję z potworkiem



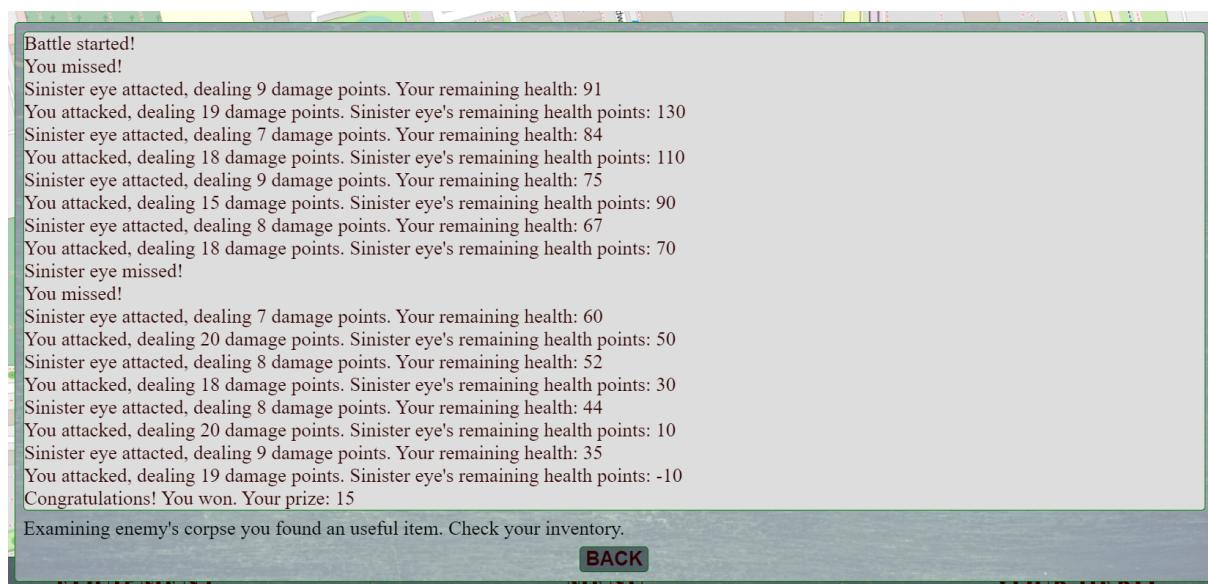
Kliknięcie na przeciwnika spowoduje pojawienie się okienka "Engage". Można je zignorować, co jest równoważne zrezygnowaniu z walki, albo też kliknąć na przycisk, który zaowocuje rozpoczętą walką... albo ekranem takim jak niżej.

6. Zbyt daleka odległość



Aby zaatakować potwora, odległość między nim musi wynosić maksymalnie kilkanaście metrów. Jeżeli gracz spróbuje zrobić to z większego dystansu - operacja się nie powiedzie i ujrzy ekran jak wyżej.

7. Raport walki



Po potwierdzeniu chęci interakcji z potworkiem przy jednoczesnym zachowaniu bliskiego dystansu, użytkownikowi wyświetla się raport walki z przeciwnikiem. Są w nim widoczne kolejne etapy walki, takie jak informacja czy atak był udany, zadane obrażenia czy zmiana punktów życia w poszczególnych etapach. Akcja może zakończyć się dwójako:

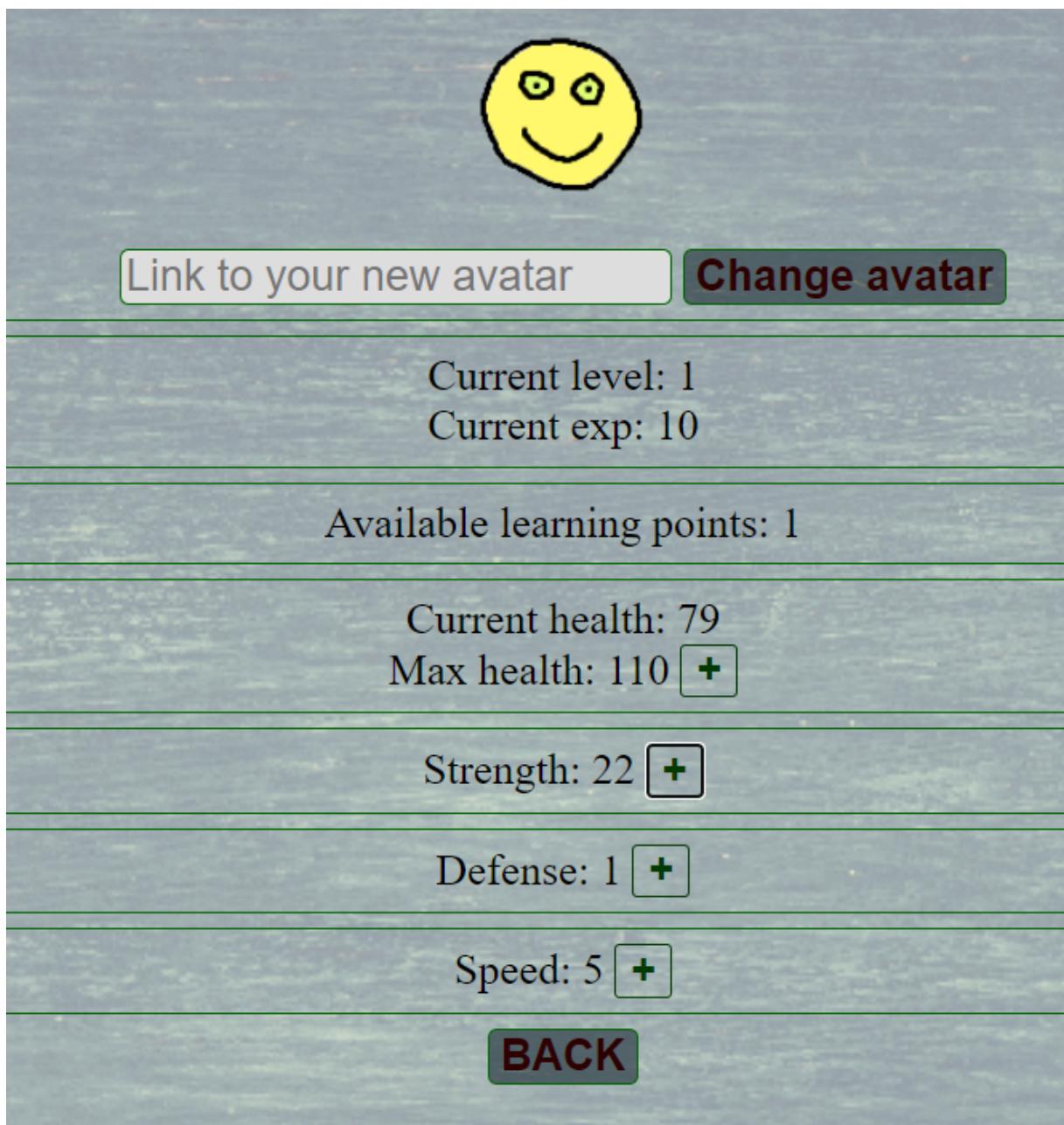
- Zwycięstwem bohatera (jak na obrazku) - wtedy ten otrzymuje punkty doświadczenia i ewentualne itemki. Owocuje też jednak straceniem pewnej liczby punktów zdrowia, które użytkownik może nadrobić poprzez poruszanie się na mapie (co promuje zdrowy tryb życia) bądź używając itemek typu "consumable". Zdobycie odpowiedniej ilości punktów zdrowia wraz z pokonaniem konkretnej ilości różnych typów bohaterów skutkuje zdobyciem kolejnego poziomu, co skutkuje pojawianiem się większej ilości przeciwników na mapie i otrzymaniem dodatkowych punktów doświadczenia do wymienienia na kolejne umiejętności.
- Bolesną porażką - skutkuje to wyzerowaniem punktów życia i obniżeniem punktów EXP do minimalnego na dany poziom.

8. Panel dolny



Podczas całej rozgrywki (po zalogowaniu się) na dole jest wyświetlony panel zawierający trzy opcje. Można je kliknąć w dowolnym momencie, a skutkują one scenariuszami, jak niżej.

9. Menu gracza



Wybranie środkowego prostokąta z wyżej wspomnianego dialogu owocuje wyświetleniem się ekranu jak wyżej. Klient może tu wybrać nowy avatar wpisując link URL bądź bezpośrednią ścieżkę do obrazka ze swojego dysku; podejrzeń aktualny level, punkty EXP dostępne punkty talentu i aktualne zdrowie; a także zobaczyć statystyki dotyczące możliwego maksymalnego poziomu zdrowia, siły, obrony i szybkości. Te cztery może zwiększyć wykorzystując punkty doświadczenia. Na koniec też oczywiście opcja "Back" do powrotu do głównego obrazu gry.

10. Ekwipunek



Po kliknięciu na trzecią spośród opcji z panelu dolnego graczowi wyświetla się ekwipunek. Jest to panel złożony z trzech części:

- "Equipped" - jest to prostokąt 1x2, w które klient może przesunąć broń ręczną zwiększąjącą siłę, a także przedmiot powodujący mniejsze obrażenia zadawane przez przeciwników. W obu tych kwadratach może być maksymalnie po przedmiocie i muszą to być itemki właśnie tego typu - odpowiednio typu "weapon" i "armor". Aktywne przedmioty można także przenieść do pozostałych dwóch prostokątów.
- "Trash" - przesunięty do tego pojedynczego kwadratu przedmiot jest bezpowrotnie usuwany.
- "Backpack" - prostokąt 4x5, w którym domyślnie pojawiają się zdobyte przedmioty dzięki walkom z potworkom (przy odpowiadającym szczęściu). Można je w nim dowolnie przestawiać. Do przepełnionego plecaka nie trafiają już kolejne itemki.

11. Menu



Środkowa akcja ukazuje klientowi interfejs jak wyżej. Znowu widać w nim grafikę główną gry, można się wylogować, wrócić do klasycznej rozgrywki... a także wpisać cheat - specjalny kod, który zaowocuje zmienieniem przebiegu aplikacji, w zależności od wpisania ścisłe tajnej komendy.

12. Przykładowy efekt kodu



Dla testerów został udostępniony na przykład specjalny cheat, który powoduje wyświetlenie zupełnie nowego pola. Zaznaczenie opcji "Do you want to enable tester mode?" skutkuje możliwością szybkiego poruszania się po mapie z pomocą strzałek na klawiaturze, a także program przestaje pobierać lokalizację użytkownika i potworki generują się o wiele częściej.