**Obraz zawierający tekst

Opis wygenerowany automatycznie**

PRACA DYPLOMOWA

Adam Marciniak

Planszowo-karciana wieloosobowa gra sieciowa

**Board and card multiplayer online game**

Promotor: dr inż. Zbigniew Szpunar

Kalisz 2023

**Temat pracy dyplomowej  
numer INF/99/2023**

| Student | **Adam Marciniak** |
| --- | --- |
| Numer albumu | **30692** |
| Studia  (stacjonarne, niestacjonarne) | **Stacjonarne** |
| Tytuł pracy (PL) | Planszowo-karciana wieloosobowa gra sieciowa |
| Tytuł pracy (EN) | Board and card multiplayer online game |
| Cel i zakres pracy (PL) | Celem projektu jest stworzenie gry planszowo-karcianej w formie aplikacji webowej, w której kilku graczy będzie mogło prowadzić wspólną rozgrywkę. Pracę będzie obejmować  analiza dziedziny gier planszowo-karcianych, potrzeb graczy, projektowanie aplikacji, tworzenie prototypów, implementacja projektu i systemów potrzebnych do pracy oraz testowe wprowadzenie na serwerze lokalnym |
| Promotor | **dr inż. Zbigniew Szpunar** |
| Termin oddania pracy | **29 lutego 2024 r**. |

Podpis promotora: …......................................

Zobowiązuję się samodzielnie wykonać pracę dyplomową na powyższy temat i nie zlecać innym osobom jej opracowania – w całości, ani w części. Wszystkie elementy – między innymi rysunki, tabele, programy komputerowe, urządzenia – które zostaną wykorzystane w pracy, a nie będą mojego autorstwa, zostaną w odpowiedni sposób oznaczone z podaniem źródła ich pochodzenia. Treść zawarta w elektronicznej wersji pracy będzie taka sama, jak w jej wersji papierowej. Przyjmuję do wiadomości, iż w przypadku niespełnienia powyższych warunków, Uczelnia może nie dopuścić mojej pracy dyplomowej do obrony lub wydać negatywną decyzję w sprawie nadania mi tytułu zawodowego.

|  |  |
| --- | --- |
| Kalisz, 30.09.2023 r. | Podpis studenta: ……….................................... |

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Oświadczenie autora pracy dyplomowej  
numer INF/99/2023**

Ja, niżej podpisany/a ……………………………………………….., Student/ka Wydziału Politechnicznego Akademii Kaliskiej im. Prezydenta Stanisława Wojciechowskiego oświadczam, że przedkładana praca dyplomowa pod tytułem

Planszowo-karciana wieloosobowa gra sieciowa

1. została przygotowana przeze mnie samodzielnie\*, wspólnie z ….....................................................;
2. nie narusza praw autorskich w rozumieniu ustawy z 04 lutego 1994 roku o prawie autorskim i prawach pokrewnych (tekst jedn. Dz.U. z 2019 poz. 1231) oraz dóbr osobistych chronionych prawem;
3. nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony;
4. nie była podstawą nadania dyplomu uczelni wyższej lub tytułu zawodowego ani mnie, ani innej osobie.

Ponadto oświadczam, że treść pracy przedstawionej przeze mnie do obrony, zawarta na przekazanym nośniku elektronicznym, jest identyczna z jej wersją drukowaną.

Miejsce, data i podpis dyplomanta: …..................................................................... .

\*) Uwzględniając merytoryczny wkład promotora (w ramach prowadzonego seminarium dyplomowego).

Wariant „wspólnie z ...” dotyczy prac dyplomowych realizowanych przez więcej niż jednego studenta.

Spis treści

[1. Cel i zakres pracy 5](#_Toc137725751)

[2. Wizja gry 6](#_Toc137725752)

[2.1 Słownik pojęciowy 7](#_Toc137725753)

[3. Model domenowy 9](#_Toc137725755)

[4. Model wymagań 11](#_Toc137725756)

[5. Model przypadków użycia 12](#_Toc137725757)

[6. Model architektury 14](#_Toc137725758)

[7. Projekt bazy danych 15](#_Toc137725759)

[8. Projekt interfejsu 16](#_Toc137725760)

[9. Implementacja 17](#_Toc137725761)

[9.1 System wyboru pola 17](#_Toc137725762)

# 1. **Cel i zakres pracy**

Celem projektu jest stworzenie gry planszowo-karcianej w formie aplikacji webowej. Ma umożliwiać graczom wspólną grę bez względu na:

* miejsce
* sytuacje
* ilość dostępnego miejsca
* dostępność czasu znajomych na wspólną grę poprzez możliwość grania z losowymi osobami lub zapisanie gry aby ją dokończyć później
* dostęp do własnego komputera

Gra ma być podobna do gier typu Talisman i Dungeon&Dragons aby zainteresować te grupy graczy. Ten projekt ma się wyróżniać spośród innych brakiem przymusu korzystania ze swojego komputera lub instalacji klienta gry, który zawsze dużo waży. Wszelakie dane wraz z klientem, maja być przechowywane na serwerze i dostarczane mu w razie potrzeby na dowolny komputer.

Stworzenie gry będzie obejmować etapy:

* zebranie danych na temat tego typu gier i światów fantasy
* stworzenie jednolitej wizji gry
* określenie głównych elementów gry
* stworzenie prototypów planszy i elementów graficznych
* wybranie technologii
* wstępna implementacja

# 2. **Wizja gry**

Gra będzie aplikacją webową z czterema podstronami.

* Strona główna:

Ta strona będzie prezentacją i krótkim opisem. Znajdą się na niej zdjęcia z rozgrywki wraz z opisem aby przybliżyć potencjalnemu graczowi na czym polega ta gra

* Strona gry:

W tym widoku będzie prowadzona rozgrywka. Klient gry na tej stronie będzie łączył się z serwerem gry aby umożliwić rozgrywkę z innymi graczami. Gra będzie składać się z planszy, na której będą umieszczone wybrane postacie graczy jako ich reprezentacje w grze. Rozgrywka będzie podzielona na tury. W czasie gry gracz będzie musiał walczyć z przeciwnikami, przeciwnościami losu rzuconymi przez grę i współgraczy. Celem gry będzie odblokowanie dostępu do ostatniego etapu gry i jak najszybsze dotarcie do mety oraz eliminacja innych graczy. Można grać z losowymi aktualnie dostępnymi graczami lub tworzyć pokoje z hasłem aby grupki przyjaciół mogły wspólnie grać bez ryzyka zajęcia miejsca przez obcą osobę. Możliwa będzie rozgrywka w ilości od dwóch do sześciu graczy. Gra będzie wyposażona w fabułę aby gracze lepiej mogli się wczuć w rozgrywkę. Gra będzie składa się z planszy, na której będą umieszczone wybrane postacie graczy. Każda postać posiada określone wartości cech oraz umiejętności. Rozgrywka będzie podzielona na tury, każdy gracz w swojej kolejce ma minutę na reakcję albo traci turę. Aktualnie wybrany gracz losuje wirtualnymi kośćmi o ile pól może się ruszyć. Po wejściu na pole gracz losuje ‘kartę przygody’. Karta przygody określa co lub kogo spotyka na swojej drodze postać.

Przykłady:

* + -gracz wylosował kartę wilka. Gracz wybiera walkę lub oswojenie (musi wylosować odpowiednią liczbę oczek)
  + -gracz wylosował kartę przedmiotu. Gracz może zabrać lub zostawić przedmiot(przedmioty dają dodatkowe umiejętności/ulepszenia )
  + -gracz wylosował spotkanie z wróżbitką. Gracz ponownie rzuca kośćmi. W zależności od wylosowanych oczek zostaje obarczony klątwą lub dostaje prezent

Na planszy będą też pola typu: miasto, kapliczka, gospoda gdzie gracz może np. odwiedzić medyka, targ, pomodlić się. Na polach typu przepaść lub pustynia rzuca kośćmi, w zależności od rzutu traci punkt życia lub kompana.

Gdy dwoje graczy spotka się na tym samym polu z innym graczem może wybrać walkę aby spróbować pozbawić drugą osobę punktu życia. Takim działaniem można przyczynić się do wyeliminowania przeciwnika z gry.

Gracze w czasie gry mogą wejść w posiadanie kart dzięki którym mogą utrudniać grę innym graczom, tymczasowo zmieniać działanie pól lub całej rozgrywki.

Przykłady:

* + -wylosowanie ‘karty przygody: zwój powodzi’, każda postać która wejdzie na to pole straci jedną turę
  + -karta: opętanie, gracz wybiera ofiarę (innego gracza) . Opętana postać przez 3 tury będzie się losowo teleportować po planszy
  + -gracz spotyka demona, ale nie udaje się mu go pokonać. Szansa spotkania demona na całej planszy podwaja się

Grę wygrywa gracz, który pierwszy dostanie się do mety. Wejście do ostatniego etapu będzie zablokowanie dopóki gracz nie pokona strażnika czyli musi być odpowiednio silny.

* Strona konta:

Ta strona ma posiadać możliwość:

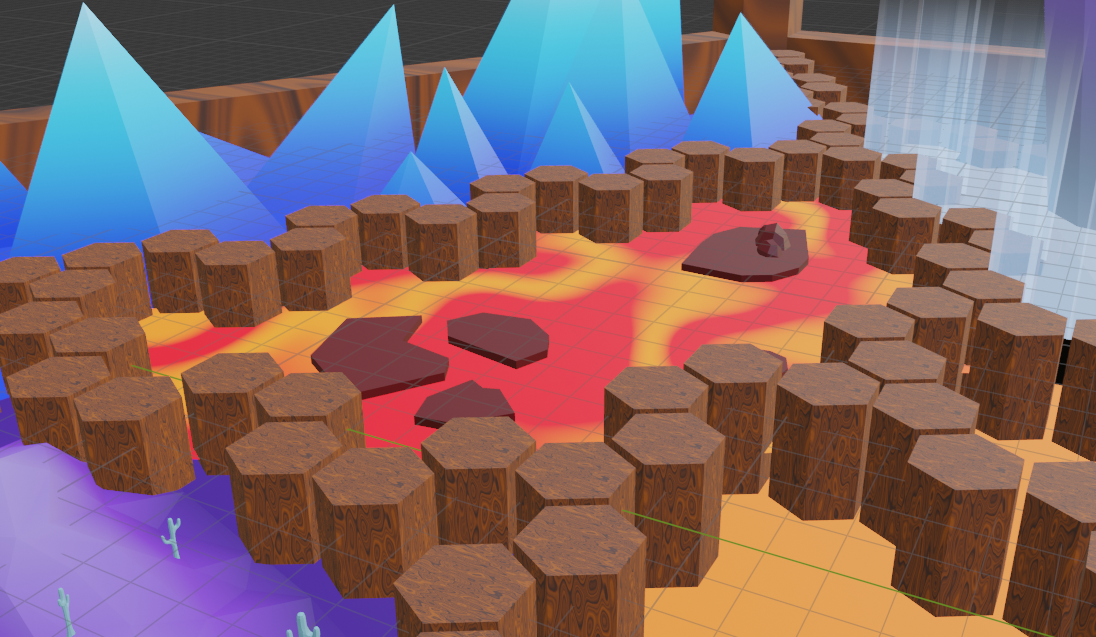
* + założenia konta graczom
  + konfigurowanie konta gracza
  + zmianę danych
  + przegląd wyników swoich ostatnich gier
  + ranking najczęściej wybieranej postaci
  + przegląd rankingu graczy

Ranking będzie oparty na punktacji, która jest sumą wypracowanych cech postaci oraz czasu i postępu w grze.

* Strona samouczka:

Będzie to miejsce, w którym wszyscy nowi gracze rozpoczynają swoją przygodę. Tutaj poznają wszystkie zasady, elementy i mechaniki gry. Poznają jak wybierać postaci i strategie najbardziej im odpowiadające. W tej zakładce będzie także możliwe przeglądanie wszystkich kart, pól i postaci wraz ze szczegółami ich działania.

## 2.1 Słownik pojęciowy

Biom – obszar/przestrzeń w którym występuję inny wygląd planszy przykład:[Rysunek 2.1.1]. Ma wpływ na to co można znaleźć i co może się wydarzyć w takim obszarze. Gracz przebywający w danym biomie jest narażony także na przeciwników w nim przebywających 

# *Rys. 2.1.1 Biom lawowy*

# 3. Model domenowy

Gra będzie podzielona na dwa główne elementy. Klienta uruchamianego w przeglądarce i serwer gry. Klient podczas ładowania będzie ściągał wszystkie dane potrzebne do samoistnej pracy i tylko komunikował się z serwerem [Rysunek 3.1]

Obraz zawierający diagram, tekst, zrzut ekranu, Plan

Opis wygenerowany automatycznie

*Rys. 3.1 Diagram domenowy (opracowanie własne)*

Client: element w którym zostaną zagregowane wszystkie elementy w jedną całość aby mogły się ze sobą komunikować

Game\_manager: zarządzanie całym systemem gry i sterowanie nią w zależności od danych przesłanych przez serwer lub decyzji użytkownika

Game data: przechowywanie aktualnego stanu gry i graczy

Main menu: tworzenie i obsługa menu głównego

Card fight system: system walki za pomocą kart i umiejętności postaci

Gameboard: ładowanie i przechowywanie danych modeli i grafik gry, z których korzysta Game\_manager. Wskład tego wchodzą takie elementy jak:

* Player: przechowywanie danych graczy, takich jak pozycja na mapie, posiadane karty, statystyki itp.
* Cards: grafika oraz dane o umiejętnościach tej karty
* Character: przechowywanie postaci, którą może wybrać gracz jako swoją postać w grze
* Enemy: przechowywanie danych o przeciwnikach, których może gracz napotkać w grze
* Place: przechowywanie modeli i opcji jakie może podjąć gracz podczas odwiedzin w specjalnych miejscach na mapie
* Biom: przechowywanie modeli elementów graficznych. Przebywanie w danym biomie ma wpływ na to jakiego rodzaju karty przygody gracz może wylosować

# 4. Model wymagań

Wymagania funkcjonalne (to co program ma robić):

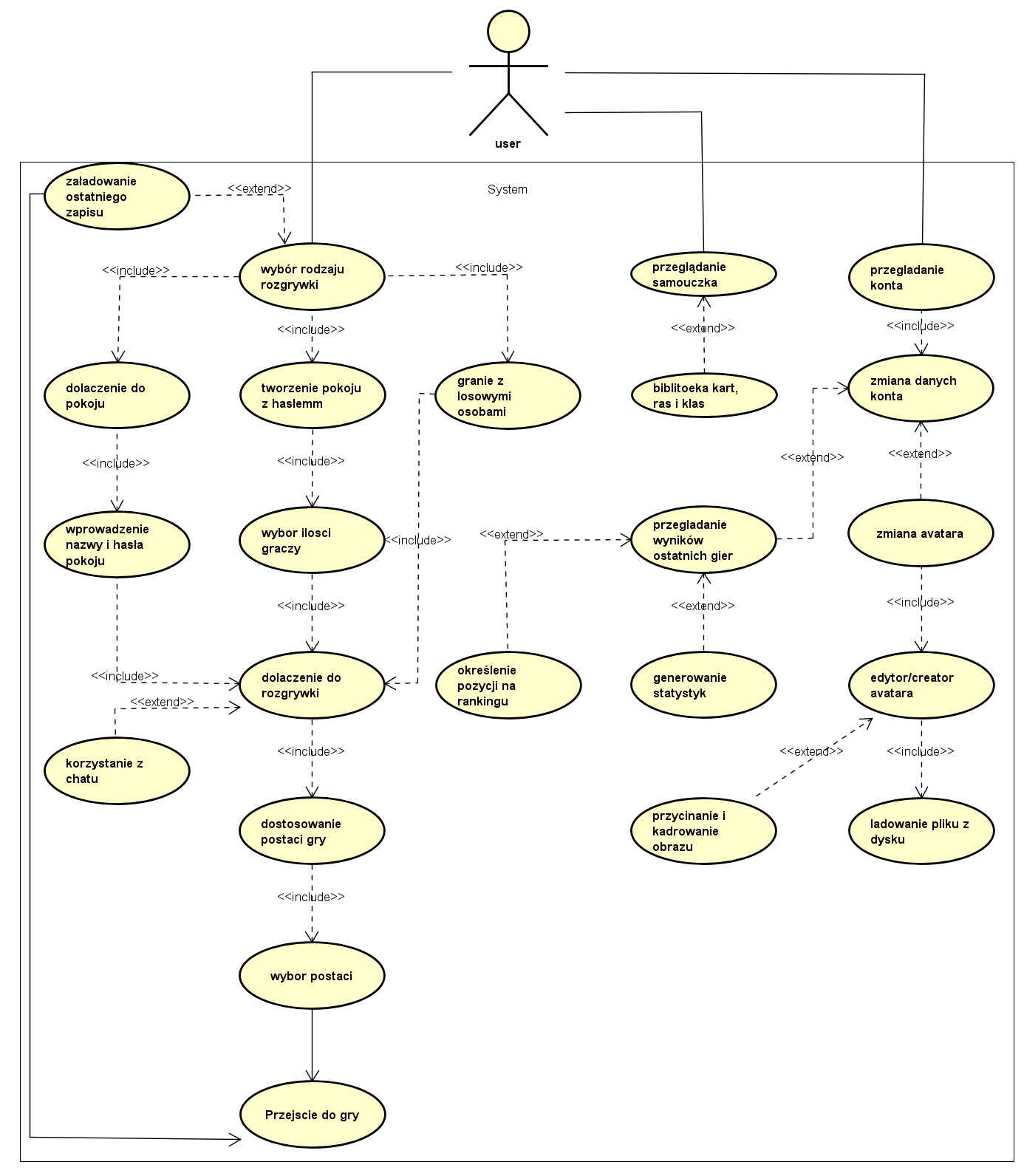
* umożliwienie graczom gry poprzez sieć
* umożliwienie graczom dostosowanie postaci
* umożliwienie graczom naukę i zasad gry
* umożliwienie graczom zmiany wyglądu ich awatara
* umożliwienie graczom zmiany danych konta
* umożliwienie graczom przeglądanie wyników swoich ostatnich gier
* generowanie statystyk dla gracza na podstawie jego gier
* utworzenie rankingu graczy
* utworzenie spisu podstawowych strategii w grze
* umożliwienie graczom rozmowy poprzez wbudowany chat

Wymagania niefunkcjonalne (jakie cechy ma mieć dana funkcjonalność np. pokój do gier ma mieć 6 osoby max):

* umożliwienie graczom grę w określonym składzie poprzez tworzenie pokoi z hasłem i ograniczać maksymalną ilość graczy do 6 w jednej sesji
* umożliwienie graczom grę z losowymi osobami
* umożliwienie graczom wybieranie rasy i klasy postaci
* program powinien posiadać samouczek, w którym w sposób opisowy z przykładowymi ekranami będą przedstawione zasady, elementy i cel gry
* program w widoku samouczka powinien umożliwiać przeglądanie kart, ras i klas
* zmiana awatara powinno umożliwiać wgranie własnego pliku lub wybranie jednej z gotowych ikon
* gracze powinni móc zmienić nazwę użytkownika, adres e-mail oraz hasło z zabezpieczeniami przed słabymi hasłami i wulgaryzmami
* wyniki gier powinny być podsumowaniem czasu, zebranych kart, cech i złota oraz czasu gry
* ranking powinien być obliczany na podstawie cech, kart, pokonanych wrogów i zdobytego złota

# 5. Model przypadków użycia

Poniższy diagram [Rysunek 5.1] obrazuje funkcjonalności, które będzie posiadać użytkownik w grze i na stronie gry.



*Rys 5.1 pierwsza wersja bazy danych*

Dostępne będzie przeglądanie danych i ustawień konta w tym:

* przeglądanie ostatnio rozegranych gier
* przeglądanie statystyki z gier
* zmian i edycji awatara (ustawienie który kawałek załadowanego obrazu ma być używany jako awatar)
* zmian danych konta (nick, e-mail)
* przeglądanie rankingu graczy

W podstronie samouczek będzie dostępna biblioteka:

* kart
* postaci
* poradników uczących o grze.

Podstrona gry pozwala na:

* wybranie rodzaju rozgrywki:
  + dołączenie do losowego pokoju
  + stworzenie własnego
  + dołączenie do pokoju o znanej nazwie i haśle
  + załadowanie zapisanej gry aby ją dokończyć.
* po dołączeniu do pokoju i w czasie gry gracze mogą korzystać z chatu gry aby rozmawiać między sobą
* podczas gry gracze najpierw wybierają postać. Dopiero po wybraniu i zatwierdzeniu gotowości do gry następuje przejście na plansze

# 6. Model architektury

Architektura systemu [Rysunek 6.1] składa się z dwóch warstw. Serwera i klientów gry. W warstwie serwera będą dwa serwery. Jeden będzie udostępniał całą stronę wraz z danymi z bazy danych. Drugi natomiast będzie obsługiwał klientów podczas gry i przesyłał dane pomiędzy nimi.

Obraz zawierający zrzut ekranu, Prostokąt, linia, diagram

Opis wygenerowany automatycznie

*Rys 6.1 architektura systemu*

Server roboczo będzie oparty na serwerze xampp i node.js. Xampp zapewni bazę danych oraz hosting aplikacji webowej. Na nim będzie pracować aplikacja oparta na laravelu oraz babylon.js

Server node.js będzie serwerem gry, który będzie odpowiedzialny za kwestię łączenia graczy, dołączania do jednego pokoju gry, utrzymania rozgrywki, prawidłowego jej przebiegu, komunikacji pomiędzy graczami i klientami (przeglądarkami i aplikacjami na nich działającymi).

# 7. Projekt bazy danych

Poniższy rysunek [Rysunek 7.1] pokazuje planowaną bazę danych. Baza będzie miała prostą strukturę do przechowywania danych użytkowników, rozegranych gier i danych potrzebnych systemowi.

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, oprogramowanie

Opis wygenerowany automatycznie

*Rys. 7.1 pierwsza wersja bazy danych*

Tabela users będzie typową tabelą do przechowywania danych użytkowników. User\_games przechowuje dane gracza z danej gry czyli ilość punktów, zebranego złota i kart w danej grze. Posiada dwa pola liczbowe user\_id które oznacza, którego użytkownika są to dane i game\_id która służy do indentyfikacji gry. Tabela games zapisuje rozegrane gry oraz id gracza który wygrał oraz czas gry. Dane te są jednakowego dla kilku graczy więc nie ma po co jest duplikować w tabeli User\_games. Tabele heroes, cards i tutorials służą do przechowywania danych postaci dostępnych grze, kart i kart poradników wraz ze ścieżkami do plików i opisami. Tabela user\_friends zapisuje id przyjaciół danego użytkownika

# 8. Projekt interfejsu

Rysunek 8.1 obrazuje prototyp widoku menu głównego gry, które składa się czterech dużych przycisków. Każdy przycisk posiada animacje pulsacji wskazujący, który jest aktualnie wybierany. Przyciski będą przejściem do innych funkcji.

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, projekt graficzny, design

Opis wygenerowany automatycznie

*Rys. 8.1 projekt prototypu interfejsu*

Przycisk z kostką do gry wyszuka graczowi losowy pokój gry i go do niego doda. Kolejny pozwoli na stworzenie własnej gry. Następny umożliwi dołączenie do pokoju o określonej nazwie i haśle. Ostatni przycisk będzie funkcją do załadowania zapisanej gry.

**Obraz zawierający tort urodzinowy, urodziny, deser, Dekorowanie ciasta

Opis wygenerowany automatycznie**

*Rys. 8.2 prototyp planszy*

# 9. Implementacja

Program w podstawowym zamyśle miał być aplikacją internetową lub aplikacją instalowaną w systemie gracza o grafice 2D. Do próby stworzenia tego wybrano biblioteki: Phaser, Pygame i Arcade, które odrzucono z powodu zbytniej problematyczności lub prostoty biblioteki. Następnym wyborem był silnik Panda3D, a plansze z rozgrywką i widokiem 2D zamieniono na wersję 3D. Panda3D także okazał się zbyt niszowym produktem.

Ostatecznie powrócono do koncepcji aplikacji webowej, ale z trójwymiarową mapą aby nie tracić czasu. Wybór padł na silnik Babylon.js wmieszanego w framework laravel wraz z pakietem Brezze. Grafika w tej grze została wykonana w formie 2D i 3D w stylizacji low-poly (minimalna złożoność modeli, które tylko mają być symbolem tego co mają przedstawiać) aby zminimalizować ilość czasu poświęconego na tworzenie grafiki, nie obciążać zużycia procesora gracza, zmniejszyć ilość danych do przechowywania na serwerze, skrócić czas ładowania plików oraz gry. Plansza, postaci graczy i elementy na niej są modelami trójwymiarowymi, karty zostały w 2D.

## 9.1 System wyboru pola

Gdy gracz wylosuje ilość oczek na kostce ten system podświetli graczowi pola na, które może przemieścić swoją postać [Rysunek 9.1.3]. Jest to obliczane poprzez funkcje rekurencyjną [Rysunek 9.1.1], która spisuje wszystkie możliwe ścieżki przeskakując po wszystkich sąsiadach pola za pomocą przeszukiwania w głąb. Wybrano przeszukiwanie w głąb gdyż to skraca szukanie ścieżek dzięki funkcji *is\_on\_the\_list*[Rysunek 9.1.2]. Sprawdzającej co iteracje czy aby nowo wybrane pole nie należy już do jakieś ścieżki co odrzuca duplikaty ścieżek na wczesnym etapie ich tworzenia. Na koniec podświetla ostatnie pole każdej możliwej ścieżki poprzez przestawienie widoczności modelu znacznika pola na widoczny[Rysunek 9.1.5 i Rysunek 9.1.6]. Po wybraniu któregoś z podświetlonych pól, następuje wygenerowanie animacji ruchu postaci gracza poprzez użycie funkcji Babylon do jednorazowych animacji, polegającej w tym przypadku na podaniu pozycji punktów z których i do których ma się dany model poruszać i czasu na wykonanie tego.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

*Rys.9.1.1 Kod funkcji rekurencyjnej, która służy do wyszukiwania ścieżekObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie*

*Rys.9.1.2 Kod funkcji odrzucajcej duplikaty ścieżek*

**Obraz zawierający zrzut ekranu, kreskówka, Animacja

Opis wygenerowany automatycznie** **Obraz zawierający wzór, szkarłupnie, gwiazda

Opis wygenerowany automatycznie**

*Rys. 9.1.3 Widok pracy systemu pola w grze Rys. 9.1.4 model górnych powierzchni pól*

System zaczyna swoją pracę po załadowaniu danych z serwera. Po tym następuje przetworzenie ich w tym odczyt pozycji pól mapy ze specjalnie przygotowanego w tym celu modelu samych górnych powierzchni pól [Rysunek 9.1.4]. Na miejsce tych pól zostaje sklonowany model pojedynczego pola z znacznikiem [Rysunek 9.1.6]. Do każdego pola są przypisane w tablicy jego sąsiednie pola

Obraz zawierający zrzut ekranu, sztuka

Opis wygenerowany automatycznieObraz zawierający zrzut ekranu, Kubeł na śmieci, cylinder, kosz

Opis wygenerowany automatycznie

*Rys. 9.1.5 pojedyńcze pole z ukrytym znacznikiem Rysunek 9.1.6 pole z widocznym znacznikiem*

## 9.2 System komunikacji klienta z serwerem

Komunikacja oparta jest na bibliotece WebSocket, która pozwala na nawiązywanie stałego połączenia i obustronną komunikacje. Podczas wysyłania wiadomości dane są zapisywane do obiektów zawierających dane identyfikujące każdego gracza, numer pokoju, rodzaj wybranego pokoju, dane sceny lub jej aktualizacji [Rysunek 9.2.1]. Następnie obiekt jest parsowany do formatu json i wysyłany na serwer [Rysunek 9.2.2]. Na serwerze dane są odczytywane i przesyłane do reszty graczy w tym samym pokoju oraz aktualizowany jest stan sceny na serwerze na wypadek utraty połączenia przez jednego lub więcej graczy.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

*Rys. 9.2.1 kawałek kodu klasy Game\_sync odczytujący akceptacje serwera na dolączenie do pokoju i wysłanie danych gracza tego klienta do określonego pokoju*

*Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie*

*Rys. 9.2.2 funkcja klasy Connection\_manager parsujący podany obiekt do formatu json i wysyłajacy go na serwer*

Zestawienie prac dokumentacji

|  |  |
| --- | --- |
| Prace wykonane | Prace do wykonania |
| Wstęp (postęp:100%) | Drobne poprawki, które ujednolicą dokument z faktycznie powstały systemem |
| Model domenowy(postęp:80%) | Dopisanie klas, które będą wspomagać główne klasy |
| Model wymagań(postęp:95%) | Sprawdzenie czy wszystkie wymagania są tam ujęte |
| Model przypadków użycia (postęp:50%) | Zrobienie diagramu przypadków dla administratora |
| Projekt architektury(postęp:100%) |  |
| Projekt bazy danych(postęp:100%) |  |
| Projekt interfejsu(postęp:100%) |  |
| Implementacja(postęp:10% | Dalsze rozszerzanie pliku w miarę powstawania oprogramowania |

Zestawienie prac implementacyjnych

|  |  |
| --- | --- |
| Prace wykonane | Prace do wykonania |
| Postawienie projektu na laravelu | -Stworzenie podstron do przeglądania elementów gry i samouczków  -dodanie panelu do zarządzania stronę przez administratora |
| Stworzenie głównej petli gry i sceny 3D(postęp:100%) | -Dodanie skybox’a do sceny gry aby imitować otoczenie;  -System walki kart i system rzutu kostką  -Dodanie wyświetlania statystyk innych graczy podczas gry |
| Stworzenie modeli mapy, karty, postaci i innych grafik(postęp:90%) | -Stworzenie grafik do przedstawienia fabuły gry |
| Zaprogramowanie systemu wyboru pola(postęp:100%) |  |
| Stworzenie menu głównego(postęp:50 %) | -Stworzenie panelu ustawień;  -Obsługa wyboru innych opcji menu głównego;  -Dodanie lobby po zakończeniu gry gdzie będzie podsumowanie wyników;  -Dodanie czatu w grze |
| Łączenie się klienta z serwerem gry i system powiadomień(postęp:80%) |  |
| Prototyp serwera (postęp:70%) | -Dopracowanie serwera gry |