**Projekt „Gra Karciana”**

Karol Komorowski, Sławek Zadrożny, Daniel Ciuchta, Rafał Marchewka, Oskar Papież

18 maja 2018

**Spis treści**

1. **Wprowadzenie**
   1. Cel
   2. Zakres
2. **Omówienie projektu**
   1. Cel i zakres projektu
   2. Założenia i zależności
3. **Organizacja projektu**
   1. Struktura organizacyjna
   2. Role i zadania
4. **Zarządzanie projektem**
   1. Opis bazy danych
   2. Opis wykorzystywanych skryptów
   3. Opis wykorzystywanych kart
5. **Metody, narzędzia i stosowane technologie**
6. **Historia zmian**
7. **Wprowadzenie**

**1.1 Cel**

Niniejszy dokument opisuje organizację i plan pracy nad projektem “Gra Karciana”. Dokument definiuje czynności i procedury związane z realizacją projektu.

**1.2 Zakres**

Dokument zawiera informacje dotyczące sposobu realizacji projektu „Gra Karciana”, obejmuje następujące zagadnienia:

* Omówienie projektu
* Organizację projektu
* Opis projektu
* Metody, narzędzia i stosowane technologie

1. **Omówienie projektu**

**2.1 Cel i zakres projektu**

Celem projektu było stworzenie gry karcianej opartej na modelu free-to-play. Gra ma umożliwiać użytkownikowi potyczkę karcianą z innym użytkownikiem. Odbiorcami projektu miały zostać osoby z kręgu studentów semestru V kierunku informatyki.

**2.2 Założenia i zależności**

* Projekt został zrealizowany przez 4-osobową grupę studentów V semestru informatyki PWSIP
* Projekt został zrealizowany w ramach przedmiotu „Inżynieria Oprogramowania”, w czasie zajęć z tego przedmiotu i dodatkowych spotkań grupy
* Aplikacja obsługuje system rejestracji za pomocą strony internetowej, ponadto dostępne są loginy testowe w formacie :

Login : testxx – gdzie x to liczby od 01 do 20

Hasło : test

* Do projektu zaliczamy – aplikację desktopową, aplikację webową, aplikacje mobilną (Android.APK)

1. **Organizacja projektu**

**3.2 Struktura organizacyjna**

* Programowanie – Karol Komorowski, Rafał Marchewka, Sławek Zadrożny
* Bazy danych – Karol Komorowski, Daniel Ciuchta
* Grafika – Sławek Zadrożny, Rafał Marchewka

**3.3 Role i zadania**

* Programowanie – implementacja funkcji, skryptów PHP oraz ich opis
* Bazy danych – wprowadzenie tabel do bazy oraz edycja skryptów PHP wykorzystujących połączenie z bazą MySQL, opis tabel bazy danych
* Grafika – utworzenie oraz opis grafiki wykorzystywanej w aplikacji

1. **Zarządzanie projektem**
   1. **Opis bazy danych**

**Opis wykorzystywanych tabel:**

**Attack** – zawiera rodzaje ataków (nazwę, obrażenia, żywioł)

**Attack\_Ids** – przechowuje ataki, tabela powiązana z tabelą Card

**Card** – tabela zawiera rodzaje jednostek (karty atakujące, karty żywiołów)

**CardObject** – tabela zawierająca właściwości danej jednostki(ID, aktualne zdrowie, maksymalne zdrowie, wartość żywiołu)

**Deck\_card** – zawiera listę kart znajdujących się w talii(ID talii, ID karty)

**Deck\_drawn** – zawiera rozlosowaną talię na potrzeby rozgrywki(ID jednostki, miejsce karty w talii)

**Element** – zawiera rodzaje żywiołów (brak, ogień, woda, ziemia, powietrze)

**Element\_Count** – zawiera liczbę kart energii którymi wzmocniono daną jednostkę

**Element\_Sum** – zawiera wymagania dotyczące ataków, ile potrzeba energii danego rodzaju, aby wykonać atak

**Game\_Table** – tabela zawierają właściwości rozgrywanych partii (nazwę obu graczy, rodzaj talii, karty na ręce, ilość zagranych kart)

**Hand\_drawn** – tabela zawiera informacje o jednostkach na ręce gracza(rodzaj jednostki, pozycja karty w talii)

**MM\_Quick** – zawiera listę graczy, którzy zapisali się do dobierania(nazwa gracza)

**Pair\_Matched** – zawiera informacje odnośnie połączenia ze sobą dwóch graczy(nazwę połączonych graczy, potwierdzenie rozgrywki)

**Player** – zawiera informacje odnośnie gracza (nazwę, hasło, ilość kart, ilość talii, liczbę rozgrywek, ilość wygranych)

**Player\_hand** – zawiera informacje odnośnie kart, które gracz posiada na ręce(właściwości karty, np. życie posiadanej jednostki.)

* 1. **Opis wykorzystywanych skryptów**

**Opis skryptów PHP**

**Vars.php**

Skrypt zawiera informacje niezbędne do połączenia się z serwerem bazodanowym MySQL.

**Check\_matched.php**

Algorytm sprawdza czy gracz ma dobraną parę w systemie dobierania meczy. Algorytm otrzymuje nick gracza i sprawdzane jest czy znajduje się wpis w tabeli Pair\_Matched w której znajduje się ten nick. Jeżeli tak, wysyłane jest potwierdzenie otrzymania komunikatu przez klienta a do klienta wysyłana jest wiadomość o tym.

**Check\_queue.php**

Algorytm sprawdza status kolejki (liczbę graczy) a następnie wysyła informacje do klienta w formie liczby.

**Check\_session.php**

Algorytm sprawdza czy połączenie między klientem a serwerem jest nawiązane. Jako dane wejściowe otrzymuje nazwę użytkownika. Algorytm oczekuje wywołania przez klienta, po czym wysyła odpowiednią informację która zależy od pewnych danych. Jeżeli wszystko jest w porządku to wysyła informację o kodzie 0, jeżeli minął czas na dozwoloną odpowiedź wysyła informację 1, jeżeli baza nie wie, że użytkownik jest zalogowany wysyła informację 2, jeżeli użytkownik o podanych danych nie został odnaleziony wysyła informację 3.

**Create\_table.php**

Algorytm tworzy stół według danych graczy otrzymanych odpowiednich tabel – Pair\_Matched, Player, Deck\_card. Pobierana jest lista zestawionych par graczy, następnie pobierane są informacje o talii każdego z graczy by te mogły zostać potasowane. Po potasowaniu talii wybierane są karty z góry talii by trafić do „ręki” każdego z graczy a następnie te informacje są zapisywane w tableach Deck\_drawn oraz Player\_hand. Następnie dodawany jest wpis o utworzonym stole z odnośnikami do wcześniej przygotowanych danych.

**Do\_attack\_turn.php**

Algorytm odpowiada za obliczenie akcji ataku. Jeżeli obaj gracze wybiorą ataki to gra oblicza punkty życia kart i gracza w zależności od wyboru graczy, a następnie ustawia odpowiedni status – 2 jeżeli gra się nie zakończyła (gracz 1 zaczyna turę), 5 jeżeli wygrał gracz numer 1, 6 jeżeli wygrał gracz numer 2 bądź 7 jeżeli wystąpił remis (zdrowie obu graczy jest równe lub niższe 0).

**Draw\_card.php**

Algorytm jako dane wejściowe otrzymuje login użytkownika. Jego zadaniem jest wziąć kartę o najniższym ID powiązaną ze stołem i loginem podanego użytkownika z tabeli Deck\_drawn a następnie przenieść ją do tabeli Player\_hand.

**End\_turn.php**

Algorytm przyjmuje login użytkownika, następnie szuka odpowiadającego mu stołu by zakończyć turę. Jest to oparte na statusie rekordu Game\_table – status 2 odpowiada turze gracza o ID1, status 3 odpowiada turze gracza o ID2. Status o numerze 2 może być zmieniony przez gracza o ID1 zaś status o numerze 3 może być zmieniony przez gracza o ID2. Status 2 przechodzi w 3 zaś 3 w 2.

**get\_card\_data.php**

Algorytm pobiera dane o kartach i atakach z tabel Card, Attack\_Ids, Attack, Element i Element\_Sum. Dane te są następnie przekazywane do klienta w formie gdzie tabele oddzielone są znakiem &, rekordy znakiem $ a kolumny znakiem @.

**get\_table\_data.php**

Algorytm pobiera nick gracza. Jeżeli znajdzie stół odpowiadający nazwie gracza wysyła informacje z tabel CardObject, Player\_hand i Game\_table. Dane te odpowiadają kartom zagranym przez gracza i przeciwnika, kartom w ręce gracza, ilości kart w ręce przeciwnika i podstawowym informacjom o stole takim jak kolejność tur.

**Login.php**

Algorytm pobiera nick i hasło gracza. Jeżeli znajdzie w bazie dane odpowiadające pobranym danym i użytkownik nie będzie aktualnie zalogowany następuje zalogowanie użytkownika – zmieniane są wartości daty ostatniego logowania i potwierdzenia zalogowania użytkownika. Klient może otrzymać 4 rodzaje odpowiedzi – 0 w momencie gdy logowanie przebiegło pomyślnie, 1 – gdy login jest niepoprawny, 2 gdy hasło jest niepoprawne i 3 gdy użytkownik jest już zalogowany.

**logoff.php**

Algorytm automatycznie wylogowuje wszystkich użytkowników od których serwer nie uzyskał odpowiedzi utrzymania sesji w przeciągu 60 sekund.

**matching.php**

Algorytm pobiera dane z tabeli MM\_Quick i wykonuje parowanie graczy metodą szybką – każdą dwójkę graczy paruje ze sobą i tworzy odpowiedni wpis w tabeli Pair\_matched

**Mm\_join.php**

Algorytm pobiera z klienta nazwę użytkownika i dodaje ją do tabeli MM\_Quick.

**Mm\_leave.php**

Algorytm pobiera z klienta nazwę użytkownika i usuwa ją z tabeli MM\_Quick.

**Send\_attack.php**

Algorytm otrzymuje od klienta numer wybranego ataku i zapisuje go w bazie danych (jeżeli jest poprawny, tzn. karta klienta posiada atak o tym numerze).

**Play\_card.php**

Algorytm pobiera z klienta id karty którą gracz chce zagrać. Następnie sprawdza jaki jest to typ karty. Jeżeli typ jest równy 1 karta trafia na stół, jeżeli typ jest równy 2 sprawdzane jest czy jest jakaś karta zagrana. Jeżeli tak, karta wzmacnia zagraną kartę, w przeciwnym wypadku funkcja zwraca błąd.

**Server.sh**

Plik serwerowy napisany w powłoce linuxa uruchamiający co sekundę wykonywanie plików serwerowych odpowiedzialnych za przeliczenie statusu gry (dobierania graczy, tury ataku czy wylogowywania graczy przy danym czasie nieaktywności).

* 1. **Opis wykorzystywanych kart**

## Energia ziemi

Podstawowy  
Siła karty:1  
Typ: Ziemia  
Dodaje 1 energię ziemi walczącej jednostce

## Energia ognia

Podstawowy  
Siła karty: 1  
Typ: Ogień  
Dodaje 1 energię ognia walczącej jednostce

## Energia wody

Podstawowy  
Siła karty: 1  
Typ: Woda  
Dodaje 1 energię wody walczącej jednostce

## Energia powietrza

Podstawowy  
Siła karty:1  
Typ: Powietrze  
Dodaje 1 energię powietrza walczącej jednostce

## Żołdak

Podstawowa  
Siła karty: 1  
Typ: Brak  
Zdrowie: 40  
Ataki:  
Bezmyślne natarcie – 10 ataku, brak typu, brak energii, brak specjalnych umiejętności  
Cięcie – 30 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Niski żołd – nie może zaatakować przeciwnika jeżeli ma 10 punktów zdrowia i mniej.

## Chochlik

Podstawowa  
Siła karty: 1  
Typ: Ogień  
Zdrowie: 20  
Ataki:  
Drapanie – 10 ataku, brak typu, brak wymagań, brak specjalnych umiejętności  
Zionięcie ogniem – 40 ataku, jedna energia ognia, brak specjalnych umiejętności  
Opis: Bliski krewny gremlinów mieszkający w jaskiniach na terenie gór Teara znanych z aktywności wulkanicznej. Podobnie jak gremliny są bardzo słabe lecz bardzo liczne.

## Wilk

Podstawowa  
Siła karty: 2  
Typ: Brak  
Zdrowie: 50  
Ataki:  
Drapanie – 10 ataku, brak typu, brak wymagań, brak specjalnych umiejętności  
Ugryzienie – 30 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Opis: Jedna z wielu dzikich bestii zamieszkujących praktycznie wszystkie krainy. Znane z bezwzględności i odwagi z jaką polują na inne istoty

## Orzeł górski

Pospolita  
Siła karty: 2  
Typ: Powietrze  
Zdrowie: 40  
Ataki:  
Dziobanie – 20 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Pikowanie – 30 ataku, powietrze, jedna energia powietrza, 20 dodatkowych obrażeń przeciw jednostkom o innym typie niż powietrze  
Opis: Ten majestatyczny i dziki ptak wbrew nazwie nie żyje tylko i wyłącznie w górach. Dzięki swojemu rozmiarowi jest w stanie zabić ofiarę samą swoją masą w czasie pikowania.

## Kappa

Podstawowa  
Siła karty: 2  
Typ: Woda  
Zdrowie: 60  
Ataki:  
Drapanie – 10 ataku, brak typu, brak wymagań, brak specjalnych umiejętności  
Nurkowanie – 40 ataku, dwie energie wody, atak sekwencyjny; w turze wybrania ataku jednostka ani gracz nie przyjmuje żadnych obrażeń, w kolejnej turze zdaje obrażenia lecz nie może wybrać w niej ataku.  
Opis: Duch opiekun rzek i jezior. Według legend każdy, nawet najmniejszy akwen wodny posiada takiego opiekuna. Zaciekle broni swojego terytorium przed zakusami innych istot.



## Wędrujący rycerz

Podstawowa  
Siła karty: 4  
Typ: Brak  
Zdrowie: 90  
Ataki:  
Cięcie – 30 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Bohaterska szarża – 0(min.40) ataku, brak typu, dwie dowolne energie, zadaje dodatkowe 20 obrażeń za każdą posiadaną energię.  
Poświęcenie – pasywna – w momencie śmierci przyjmuje na siebie wszystkie obrażenia (nadwyżka nie jest zadawana graczowi).

## 

## Jaszczur jaskiniowy

Podstawowa  
Siła karty: 6  
Typ: Brak  
Zdrowie: 120  
Ataki:  
Jaszczurze szpony – 30 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Jaszczurze ugryzienie – 50 ataku, brak typu, dwie dowolne energie, brak specjalnych umiejętności  
Siarkowy oddech – 20 ataku, brak typu, jedna dowolna energia, zadaje dodatkowe 20 ataku za każdą posiadaną energię ognia.

## 

## Gremlin

Podstawowa  
Siła karty: 1  
Typ: Brak  
Zdrowie: 30  
Ataki:   
Drapanie – 10 ataku, brak typu, brak wymagań, brak specjalnych umiejętności  
Desperacka szarża – 10 ataku + brakujące zdrowie, brak typu, brak wymagań, jednostka ginie po użyciu.  
Opis: Niewielkie stworzenia zamieszkujące polany i góry północnego Alaru. Bardzo słabe, często padają ofiarą każdego innego stworzenia, jednakże każdego utraconego osobnika mogą bardzo szybko zastąpić swoim olbrzymim przyrostem naturalnym.



## Gremlinołówka

Podstawowa  
Siła karty: 2  
Typ: Ziemia  
Zdrowie: 50  
Ataki:  
Ugryzienie – 30 ataku, brak typu, jedna dowolna energia, brak specjalnych umiejętności  
Konsumpcja – 30 ataku, brak typu, jedna energia ziemi i jedna dowolna energia, leczy za połowę zadanych obrażeń.  
Opis: Znacznie większa siostra muchołówki, wzięła nazwę od głównego posiłku – gremlinów jednakże zdarzają się przypadki zjedzenia zamroczonego alkoholem człowieka.

1. **Metody, narzędzia i stosowane technologie**

* System operacyjny Microsoft Windows
* Serwer WWW Apache
* Język C#
* Unity 3D
* PHP
* Baza danych MySQL
* StarUML
* Microsoft Office

1. **Historia zmian**

* 14/11/2017 – obrys gry
* 19/11/2017 – skrypty sql tworzące bazę danych
* 28/11/2017 – stworzenie menu i dodanie grafik
* 2/12/2017 – logowanie + sprawdzenie połączenia
* 13/12/2017 – stworzenie kart oraz animacji
* 19/12/2017 – skrypty php od sprawdzenia połączenia i dobierania graczy
* 10/01/2018 – poprawione skrypty php, wersja testowa serwera wywołującego automatycznie zapytania php
* 14/01/2018 – pobieranie danych stołu, poprawki graficzne
* 16/01/2018 – finalna wersja projekt zespołowy 1
* 24/03/2018 – interfejs ataków i przygotowanie skryptów służących do pobierania danych o nich
* 28/03/2018 – poprawienie skryptów serwera php, dodanie skryptu vars.php zawierającego dane logowania do bazy MySQL
* 3/04/2018 – dodano menu układania talii oraz wprowadzono grafiki awatarów
* 7/04/2018 – dokończono migrację serwera – usunięto wszystkie odwołania do starego serwera z kodu
* 14/04/2018 – dodano skrypty odpowiedzialne za obliczanie tury ataku
* 16/04/2018 – przygotowanie podstaw strony internetowej z działającym modułem rejestracji
* 18/04/2018 – podpięto bootstrap pod stronę internetową
* 23/04/2018 – poprawiono błędy z bootstrapem
* 9/05/2018 – podpięto skrypty ataku pod plik wykonywalny serwera
* 11/05/2018 – dodano obsługę statusów po stronie klienta, klient widzi informację o zakończeniu rozgrywki.