

# Deathmatch

Krzysztof Nasuta 193328 Aleksander Iwicki 193354 Filip Dawidowski 193433

## 1. Wstęp

Celem projektu było stworzenie gry deathmatch, w której gracze mogą walczyć ze sobą na arenie. Do stworzenia gry użyto języka C++ oraz biblioteki Boost. Gra działa w trybie tekstowym, sterowanie odbywa się za pomocą klawiatury. Gracze mogą poruszać się po arenie, strzelać do siebie oraz zbierać ulepszenia (np. szybsze strzelanie). Grę wygrywa gracz, który zdobędzie wcześniej określoną liczbę punktów, które zdobywa się poprzez eliminacje przeciwników.

## 2. Działanie z punktu widzenia użytkownika

Na początku gracz wpisuje IP serwera, na którym chce grać. Następnie gracz wybiera nick, którym będzie się posługiwał w grze. Po wybraniu nicku gracz dołącza do lobby gdzie widzi wszystkich innych graczy i może zaznaczyć gotowość. Po zaznaczeniu gotowości przez wszystkich użytkowników rozpoczyna się rozgrywka. Gracz może poruszać się po arenie za pomocą klawiszy W, A, S, D i strzelać za pomocą strzałek. Po wejściu na pole z ulepszeniem gracz je automatycznie podnosi. Za każdą eliminację przeciwnika gracz zdobywa punkt. Grę wygrywa gracz, który zdobędzie wcześniej określoną liczbę punktów.

## 3. Działanie serwera

Na początku serwer oczekuje na połączenie od klientów. Po połączeniu serwer tworzy nowego gracza i dodaje go do listy graczy. Serwer oczekuje na gotowość graczy. Po zaznaczeniu gotowości przez wszystkich graczy serwer rozpoczyna grę. Przez całą grę serwer odbiera od klientów informacje o ich ruchach i strzałach i od razu przesyła je do innych graczy, jednocześnie symulując cały stan gry u siebie. Co określoną ilość czasu, np. raz na 10 ticków gry serwer rozsyła do graczy swój stan gry. Po zakończeniu gry serwer wysyła do wszystkich klientów informację o zakończeniu gry.

co się stanie jak gracz wejdzie w nieśmiertelność i zostanie trafiony

prorytetyzacja synchronizacji stanu gry zależnie od odległości

co jeśli udp nie przyjdzie w kolejności

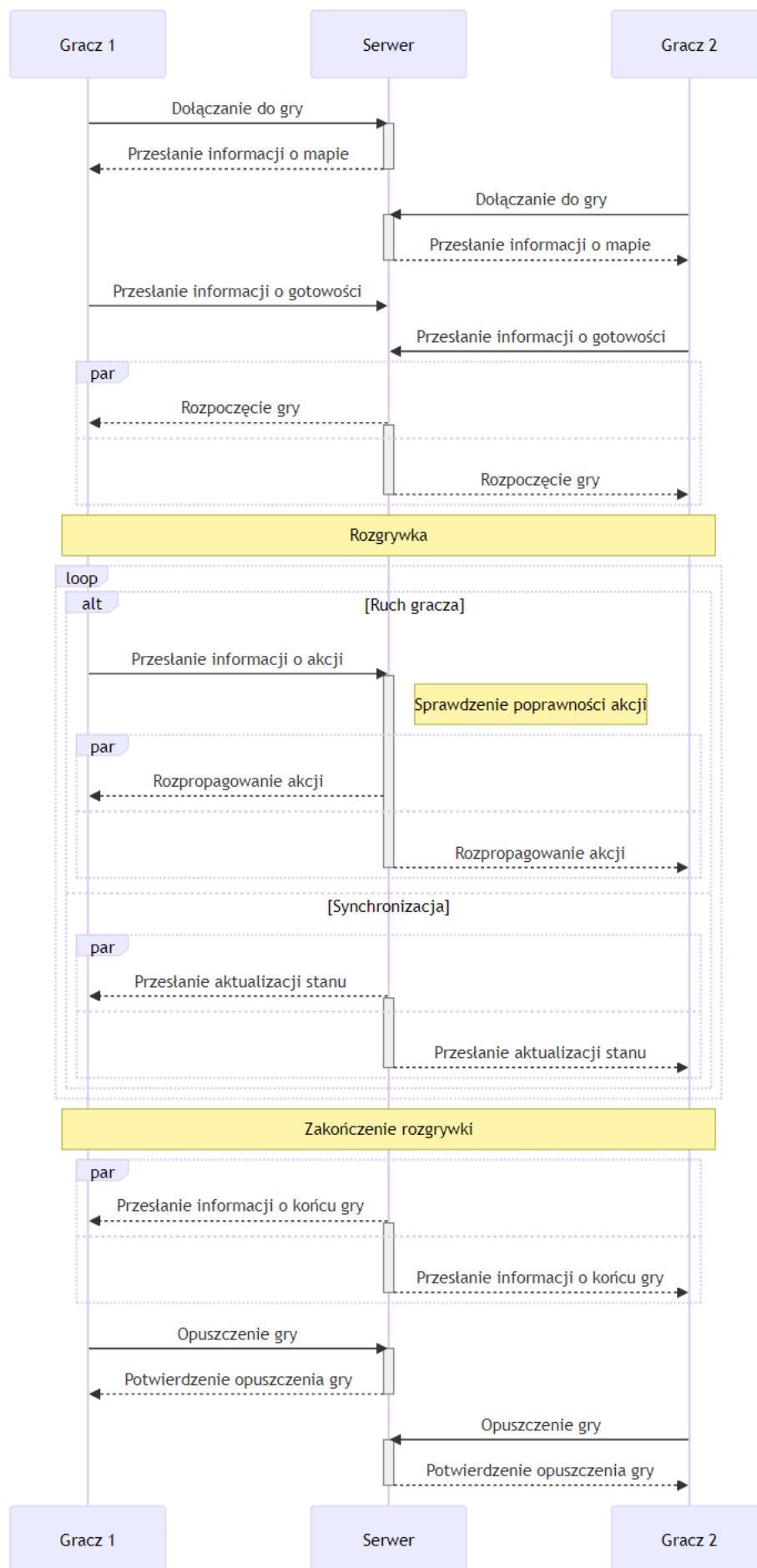
dlaczego chaotycznie nie będzie działać

### 3.1. Struktura komunikatów

Dołączenie do gry		
Nazwa pola	Typ danych	Opis
nick_length	int32	Długość nicku gracza (maksymalnie: 20)
nick	char[]	Nick gracza o długości nick_length

Przesłanie informacji o mapie		
Nazwa pola	Typ danych	Opis
width	int32	Szerokość mapy
height	int32	Długość mapy
tiles	MapTile[]	Tablica kafelków mapy o rozmiarach width*height, każdy kafelek ma rozmiar 4 bajtów

## 4. Diagram sekwencji



## 5. Diagram klas

