

Tanks - sprawozdanie

Stanisław Nieradko Bartłomiej Krawisz Jakub Bronowski

1. Wprowadzenie

Celem projektu było stworzenie gry w języku Python, która będzie symulacją bitwy czołgów obsługującą kilku graczy. Gracze sterują czołgami, które poruszają się po planszy i strzelają do siebie. Gra kończy się gdy wszyscy poza 1 graczem zostaną zniszczeni. Gracz, który pozostał na planszy wygrywa.

2. Model Komunikacji

Gra wykorzystuje protokół TCP do komunikacji między klientami a serwerem. Wybór ten został dokonany ze względu na wystarczającą wydajność względem UDP oraz prostotę implementacji.

Zdarzenie przesyłane z oraz do serwera są w formacie JSON, wg. poniższego schematu:

Nazwa	Typ	Opis
event_type	string	Typ zdarzenia (connect, refuse, ping, pong, setPlayerId, gameState, disconnect)
time	uint64	Czas zdarzenia [unix timestamp]
data	object	Dane zdarzenie (zależne od event_type)
	char	NULL kończący wiadomość

Przykładowe zdarzenia:

```
{"eventType":"connect","time":1711291958}

{"eventType":"refuse","time":1711291958}

{"eventType":"ping","time":1711293829}

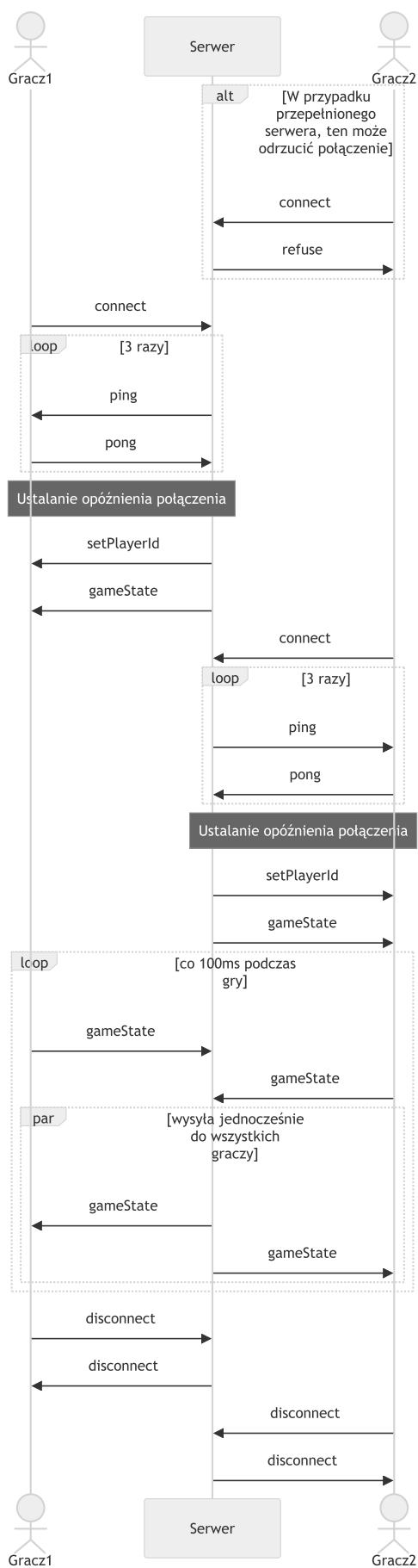
{"eventType":"pong","time":1711295934,"data":{"serverTime":1711293829 }}

{"eventType":"setPlayerId","time":1711301024,"data": 2}

{"eventType":"gameState","time":1711293829,"data": {
  "tanks": {
    "1": { "x": 115.2, "y": 254.32, "direction": 1, "speed": 100, "score": 50,
"alive": true },
    "2": { "x": 34.2, "y": 74.32, "direction": 0, "speed": 0, "score": 0,
"alive": false }
  },
  "bullets": [
    { "x": 115.2, "y": 254.32, "direction": 1, "speed": 100, "playerId": 2 },
    { "x": 34.2, "y": 74.32, "direction": 0, "speed": 0, "playerId": 1 }
  ],
  "isGameOver": false
}}

{"eventType":"disconnect","time":1711895043}
```

2.2 Diagram Sekwencji



Gracze dołączają do serwera poprzez wysłanie zdarzenia connect. Po 3 krotnej wymianie zdarzeń ping (ze strony serwera) oraz pong (ze strony użytkownika) w celu ustalenia opóźnienia (z ang. *latency*) połączenia. Po ustaleniu opóźnienia serwer odpowiada zdarzeniem setPlayerId z przypisanym identyfikatorem gracza oraz zdarzeniem gameState z aktualnym stanem gry.

W przypadku przepełnienia serwera, serwer wysyła zdarzenie refuse i zamyka połączenie.

Podczas gry co 100ms zarówno serwer wysyła każdemu graczowi zdarzenie gameState z aktualnym stanem gry, jak i każdy gracz stara się wysłać zdarzenie gameState z aktualnym stanem swojego czołgu.

Gdy gra kończy się (po zniszczeniu wszystkich graczy poza jednym), serwer wysyła zdarzenie gameState z isGameOver ustawionym na true, co informuje klientów iż gra się zakończyła. Po 3s serwer rozpoczyna nową grę.

W celu rozłączenia się z serwerem gracz wysyła zdarzenie disconnect (w wyniku którego otrzymuje on także zwrotne zdarzenie disconnect od serwera w celu potwierdzenia).

W przypadku zakończenia pracy serwera, zostaje nadaane zdarzenie disconnect dla każdego gracza.

2.3 Diagramy klas

