

Statki kosmiczne piu-piu

Michał Sadkowski 197776, Dawid Wesołowski 197943

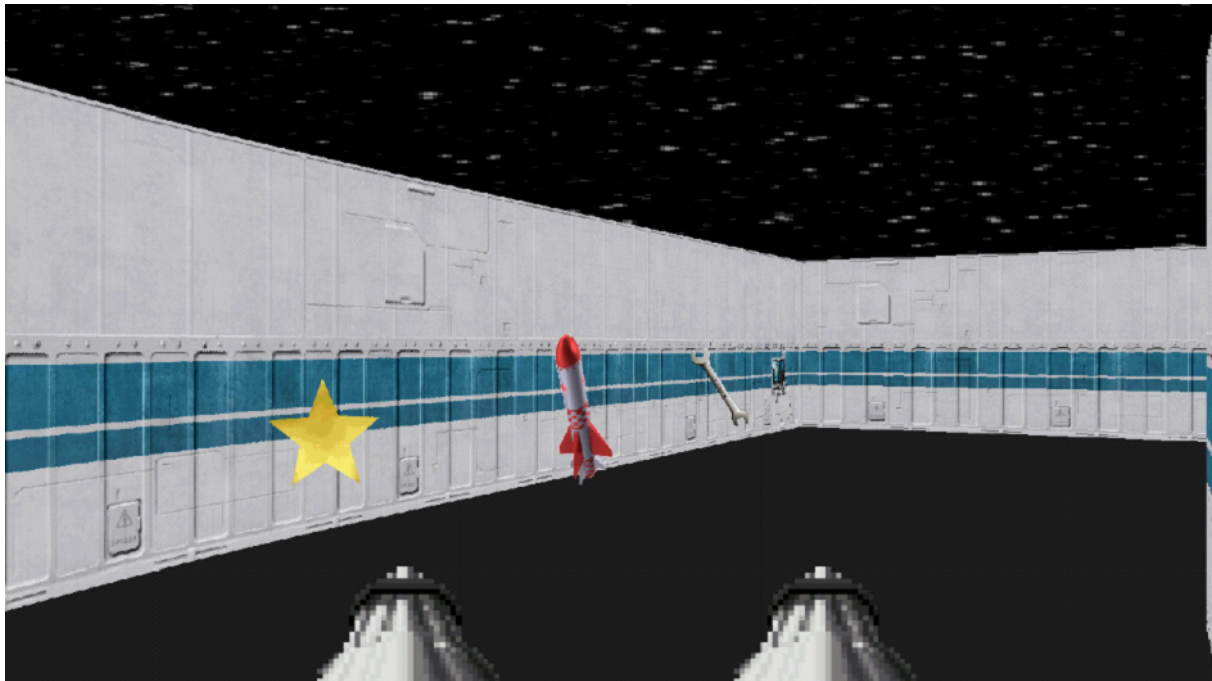
1. Opis gry

Statki kosmiczne piu-piu to gra typu FPS bazująca na systemie raycastowania znanym z legendarnego Wolfensteina 3D. Gracze mierzą się ze sobą na specjalnie przygotowanych arenach, a o zwycięstwie decyduje spryt, czas reakcji i taktyka.



1.1. Zasady

Gracze mierzą się na arenach 1v1. Zadaniem każdego z nich jest wyeliminowanie swojego przeciwnika. Na mapie rozłożone są zestawy naprawcze, mogące naprawić statek, gwiazdki dające przyspieszenie i rakietę zadającą niewyobrażalne obrażenia. Gra kończy się w momencie, gdy statek przeciwnika ulegnie destrukcji.



1.2. Sterowanie

- W – ruch do przodu
- A – ruch w lewo
- S – ruch w tył
- D – ruch w prawo
- Lewy przycisk myszy – strzał
- Prawy przycisk myszy – strzał rakietą
- ESC – wyjście z gry

2. Architektura sieciowa

Gra składa się z:

1. Serwera (server.c) - centralny punkt zarządzający stanem gry
2. Klientów (client.py) - instancje gry połączone z serwerem
3. Logiki gry (game.py, player.py, enemy.py, ...) - odpowiedzialnej za renderowanie i fizykę

2.1. Protokół komunikacji

- UDP jako podstawowy protokół dla minimalizacji opóźnień
- Stałe pakiety z polami player_id, position (x, y, kąt), actions, timestamp
- Ticki co 50 ms – serwer przetwarza akcje w batchach i rozsyła zaktualizowany stan
- Przykładowy pakiet od klienta:

```
{
  "player_id": 0,
  "pos": [12.5, 4.3],
  "angle": 45.0,
  "actions": [
    {"type": "shoot_normal", "direction": 30.0},
    {"type": "collect", "item_id": "repair_kit_42"}
  ],
  "timestamp": 2210000.543
}
```

2.2. Sekcja krytyczna i synchronizacja

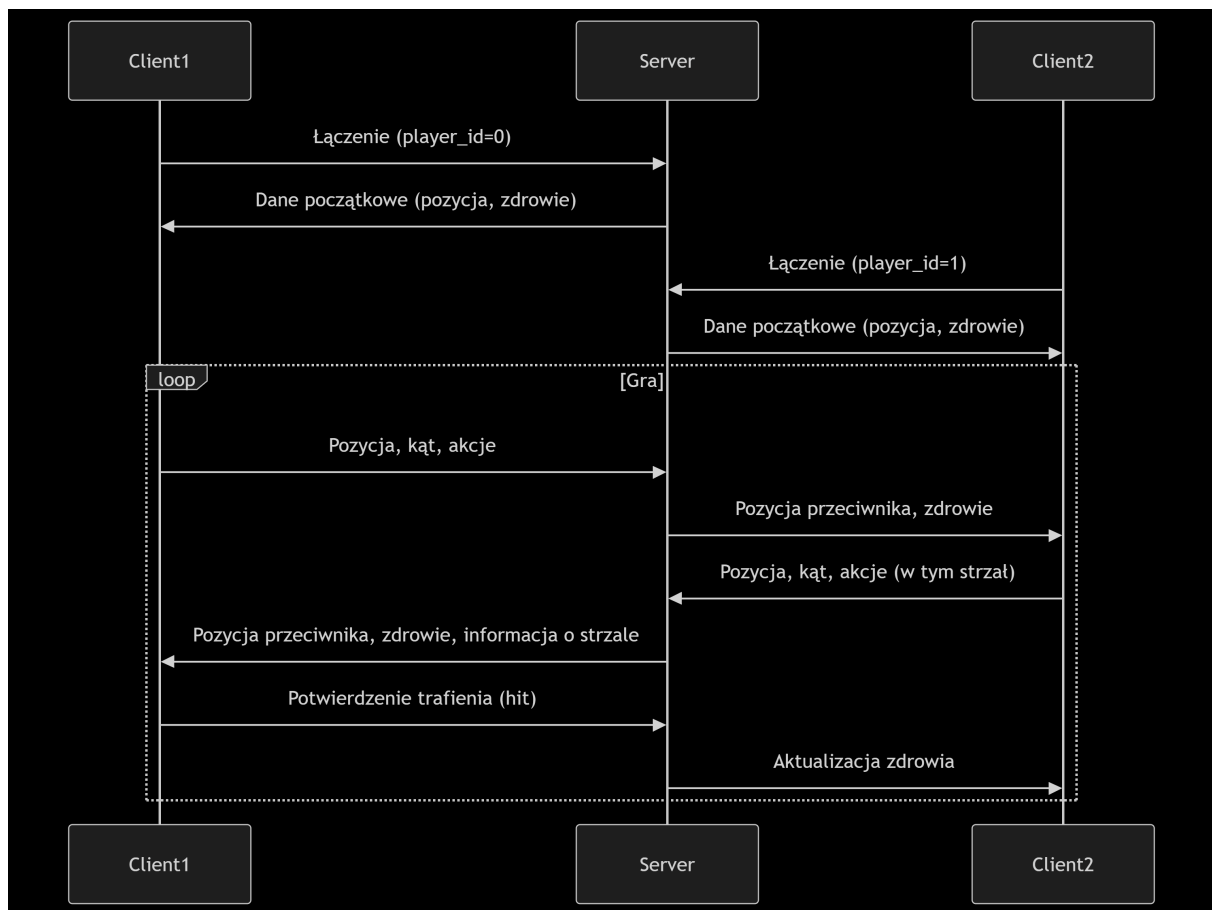
Główną sekcją krytyczną w systemie jest dostęp do współdzielonych danych o graczach na serwerze. Wątki obsługujące poszczególnych klientów mogą jednocześnie próbować modyfikować:

1. Pozycje graczy
2. Zdrowie graczy
3. Informacje o trafieniach
4. Akcje strzałów

Serwer używa mutexów do synchronizacji dostępu do tych danych. Blokada jest aktywowana przy każdej operacji modyfikującej stan gry. Bez tej blokady mogłoby dojść do:

- Wyścigów przy modyfikacji zdrowia graczy
- Niespójności w pozycjach graczy
- Utracie informacji o trafieniach

3. Diagram sekwencji



4. Komunikacja sieciowa

Protokół komunikacji

Dane przesyłane są w formacie JSON z następującymi polami:

Od klienta do serwera:

- pos: aktualna pozycja gracza
- angle: kierunek patrzenia gracza
- actions: lista akcji (np. strzałów)
- hit: flaga informująca o trafieniu

Od serwera do klienta:

- enemy_pos: pozycja przeciwnika
- enemy_angle: kierunek przeciwnika
- enemy_health: zdrowie przeciwnika
- your_health: zdrowie gracza
- enemy_shot: informacja o strzale przeciwnika
- hit: informacja o otrzymaniu trafienia

Konflikty

Serwer odpowiada za rozwiązywanie konfliktów na podstawie timestampów - serwer priorytetyzuje akcje z najwcześniejszym czasem wysłania, o ile nie wystąpiło zbyt duże opóźnienie, np.

- Jednoczesne zebranie apteczki: gracz z niższym pingiem otrzymuje przedmiot
- Kolizja pocisków: serwer decyduje, który strzał trafił pierwszy (i czy ewentualnie drugi użytkownik był jeszcze w stanie strzelać)
- Strzał rakieta a naprawa: jeśli serwer odbierze akcję naprawy przed trafieniem rakiety, gracz przeżyje



5. Podsumowanie

Sieciowa implementacja Statków kosmicznych piu-piu koncentruje się na precyzyjnym śledzeniu pocisków, zarządzaniu przedmiotami oraz sprawiedliwym rozstrzyganiu konfliktów w czasie rzeczywistym. Kluczowe elementy to niskie opóźnienia, odporność na oszustwa oraz płynna synchronizacja dynamicznych obiektów (rakiety, apteczki).