

Statki kosmiczne piu-piu

Michał Sadkowski 197776, Dawid Wesołowski 197943

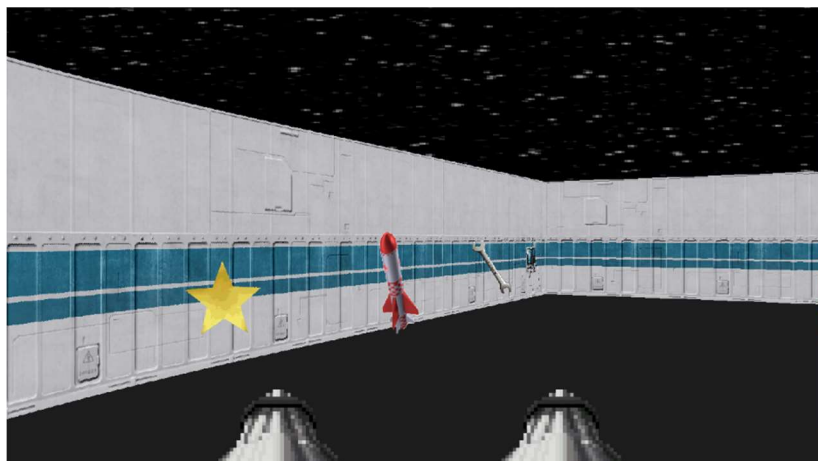
1. Opis gry

Statki kosmiczne piu-piu to gra typu FPS bazująca na systemie raycastowania znanym z legendarnego Wolfensteina 3D czy Doom. 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 rakiety zadające 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. Protokół komunikacji

- **UDP** jako podstawowy protokół dla minimalizacji opóźnień
- **Stale pakiety** z polami:
 - player_id
 - position (x, y, kąt)
 - actions (strzał/zbieranie)
 - timestamp (dokładny czas akcji)
- **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": 1630000000.123
}
```

3. Synchronizacja stanu gry

- **Autorytatywny serwer:** decyduje, które pociski trafiają, które przedmioty są zebrane, i aktualizuje zdrowie graczy
- **Interpolacja ruchu:** klienci wygładzają ruchy przeciwników między aktualizacjami serwera

- **Predykcja pocisków:** klienci renderują strzały lokalnie, ale ostateczny wynik trafienia jest weryfikowany przez serwer

4. Rozwiązywanie konfliktów

- **Zasada timestampów:** serwer priorytetyzuje akcje z **najwcześniejszym czasem wysłania**
- **Przykładowe scenariusze:**
 - **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. Mechanika sieciowa akcji

- Strzały:
 - **Zwykle:** niskie opóźnienie, częste, serwer weryfikuje kolizje z hitboxami statków
 - **Rakiety:** wymagają podniesienia przedmiotu, serwer blokuje nadmiarowe użycie i śledzi ich eksplozje
- Zbieranie przedmiotów:
 - Serwer zarządza **respawnem** apteczek i rakiet na mapie
 - Tylko **pierwszy gracz** w hitboxie przedmiotu może go zebrać

6. Architektura

Wykorzystano **architekturę klient-serwer** z centralnym serwerem koordynującym stan gry. Serwer odpowiada za:

- Śledzenie pozycji statków, pocisków (zwykłych i rakiet), oraz przedmiotów (apteczki, rakiety)
- **Rozwiązywanie konfliktów** (np. kolizje pocisków, jednoczesne zbieranie przedmiotów)
- **Weryfikację poprawności akcji klientów** (np. limit strzałów, czas odnowienia rakiet)
- Zarządzanie warunkami zwycięstwa (zdrowie graczy)

Klienci są odpowiedzialni za:

- Renderowanie stanu gry (statki, pociski, efekty wizualne)
- **Wysyłanie akcji gracza** (strzał zwykły, strzał rakieta, zbieranie przedmiotów)
- Lokalną predykcję ruchu pocisków i przeciwników.



7. Zarządzanie połączeniami

- **Reconnect:** gracz może dołączyć ponownie w ciągu 15s – jego statek jest "niezniszczalny" w tym czasie
- **Wykrywanie opóźnień:** gracze z pingiem >150 ms mają akcje odrzucane przez serwer (zapobiega teleportacji)
- **Kick za cheatowanie:** serwer wykrywa niemożliwe akcje (np. strzały 10x szybciej niż limit) i usuwa gracza

8. Bezpieczeństwo

- **Walidacja ruchu:** serwer sprawdza, czy prędkość statku nie przekracza maksymalnej wartości
- **Anti-cheat:** porównywanie stanu zdrowia klienta z obliczeniami serwera

9. Optymalizacje

- **Kompresja pozycji:** Współrzędne statków kodowane w 16-bitach zamiast floatów
- **Priorytetyzacja:** Pociski i eksplozje są wysyłane z wyższym priorytetem niż pozycje statków
- **Dead Reckoning:** Klienci przewidują ruch przeciwników na podstawie ostatnich danych

11. 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).