

## Sprawozdanie sk2

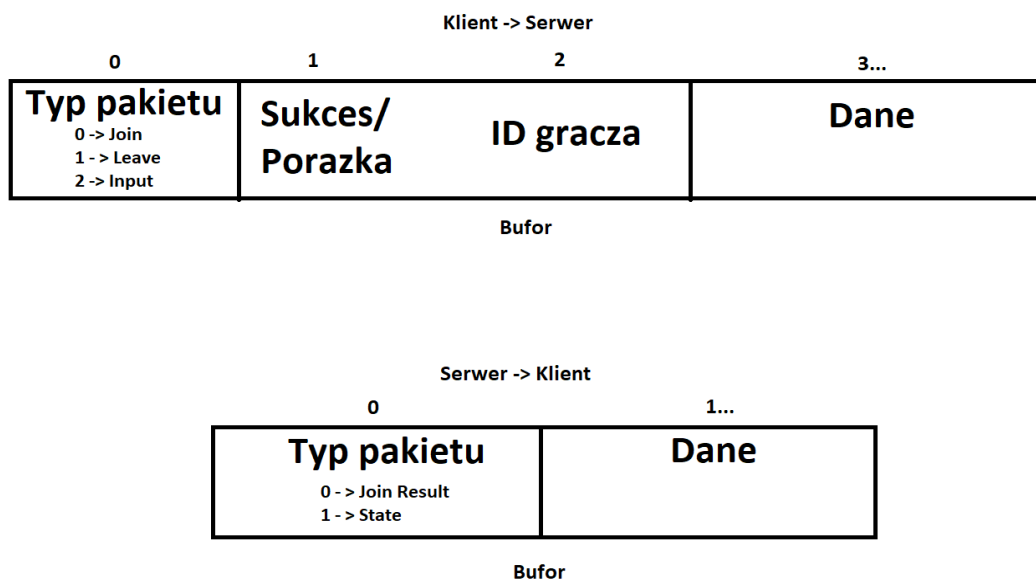
### Wieloosobowa gra czasu rzeczywistego

#### 1. Opis projektu

- Projekt to wieloosobowa gra platformowa 2D rozgrywana w czasie rzeczywistym. Celem graczy jest współpraca w celu otwarcia portalu i ukończenia poziomu. Każdy z graczy kontroluje awatara o unikalnym kolorze i musi dotrzeć do przycisku odpowiadającego jego barwie. Na przykład gracz z awatarem w kolorze niebieskim może aktywować jedynie niebieski przycisk, podczas gdy przycisk czerwony pozostanie dla niego nieaktywny. Portal otwiera się dopiero wtedy, gdy trzech gracze dotrą do swoich przycisków i wciśnięte zostaną one jednocześnie. Choć gra obecnie zawiera tylko jeden poziom, ma ogromny potencjał do rozszerzania poprzez dodanie nowych przeszkód i kolejnych poziomów.
- Użyta technologia:
  - Po stronie serwera: C++ dla systemu operacyjnego linux przy użyciu API `bsd-sockets`
  - Po stronie klienta: C++ dla systemu operacyjnego linux przy użyciu biblioteki obsługującej GUI, SFML (Simple and Fast Multimedia Library)

2. Komunikacja między klientem a serwerem opiera się na wymianie prostych wiadomości (pakietów) w ramach protokołu UDP. Serwer pełni funkcję centralnego zarządcy stanu gry, obliczając fizykę i interakcje, a następnie synchronizując dane z klientami. Dzięki zastosowaniu UDP komunikacja jest szybka, ale wymaga dodatkowej obsługi niezawodności (np. timeoutów)

#### I. Struktura pakietu



II. Jest 5 rodzajów pakietów potrzebnych do skutecznej komunikacji:

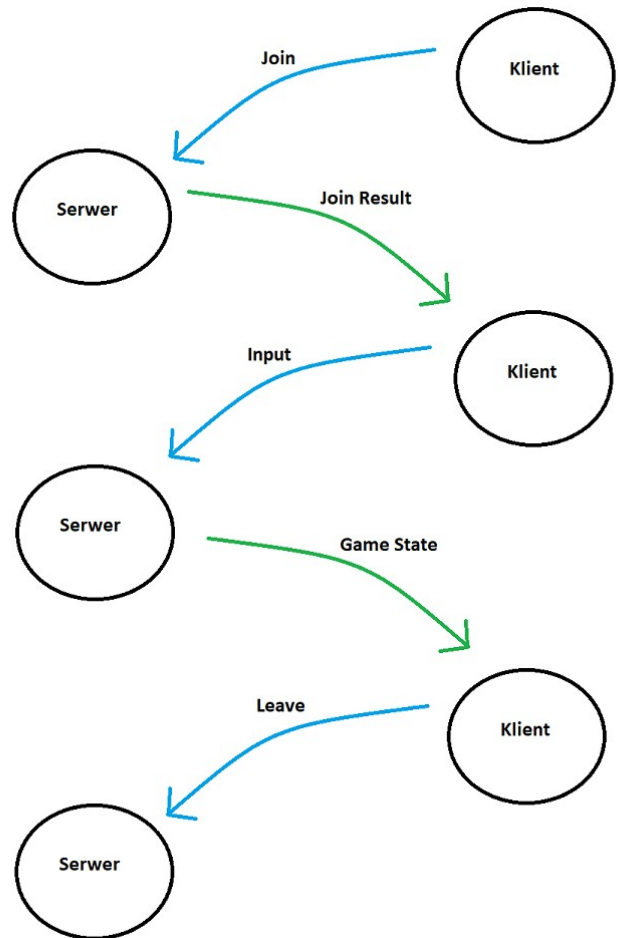
→Klient wysyła wiadomość, informując serwer o chęci dołączenia do gry [**Join**]

→Serwer nadaje i wysyła klientowi jego ID [**Join\_Result**]

→Klient wysyła swoje dane wejściowe, obsługa klawiatury [**Input**]

→Serwer przetwarza aktualny stan gry i wysyła go do wszystkich klientów [**Game State**]

→Klient informuje serwer o opuszczeniu gry [**Leave**]



### 3. Podsumowanie

- Najważniejsze informacje o implementacji
  - Serwer działa w sposób współbieżny, umożliwiając jednoczesne obsługiwane wielu gier. Oznacza to, że jeśli w danej instancji gry zostanie osiągnięta maksymalna liczba graczy, a nowy gracz spróbuje się połączyć, zostanie dla niego utworzona nowa instancja gry, która będzie działać w osobnym wątku
  - Klient używa SFML do renderowania obiektów gry: tło, ziemia, przyciski, gracze
  - Klient zbiera dane z klawiatury (góra, dół, lewo, prawo) i wysyła je do serwera
  - Klient aktualizuje stan graczy (pozycje, kierunki, stan przycisków) na podstawie informacji od serwera
  - Gra działa na stałej liczbie ticków na sekundę (120 ticków). Jeśli tick jest za szybki, klient czeka na resztę czasu.
  - Jeśli dany gracz nie będzie się poruszał przez określony czas zostanie automatycznie rozłączony przez serwer
  - Jeśli w grze przez 100s nie będzie ani jednego gracza, gra zostanie zamknięta
  - W przypadku gdy serwer odrzuci prośbę klienta o nadanie ID, klient będzie ponawiał prośbę w równych odstępach czasu aż do uzyskania ID

- Co sprawiło trudność
  - Przekształcenie serwera na współbieżny tak aby obsługiwał wiele gier jednocześnie
  - Projektowanie gry gdzie cała logika odbywa się na serwerze, a nie na kliencie
  - Błędy trudne do zdiagnozowania wynikające z niepoprawnego zrozumienia komunikacji sieciowej