

## WZR-lab Komunikacja w architekturze multicast - instrukcja do zadania

Celem zadania jest poznanie możliwości komunikacji pomiędzy aplikacjami oraz możliwości interakcji pomiędzy obiektami w ramach wirtualnego świata z użyciem architektury multicast.

### Krótki opis programu bazowego:

Po uruchomieniu programu bazowego, użytkownik może sterować jednym obiektem ruchomym (MojObiekt lub MyVehicle lub my\_car). Opcje sterowania i ustawiania widoków są szczegółowo opisane w pomocy wywoływanej klawiszem F1. Obiekty ruchome innych użytkowników znajdują się w słowniku (obiekty\_ruchome lub other\_objects lub other\_cars), którego kluczem jest numer ID, zaś wartością wskaźnik do obiektu. Do odbierania komunikatów z innych aplikacji służy specjalny wątek wraz z funkcją obsługi (ReceiveThreadFun() lub WątekOdbioru()). Na wejście funkcji obsługi wątku przesyłany jest obiekt multi\_reciv służący do odbioru ramek. Po odbiorze ramki sprawdzany jest jej typ. W przypadku informacji o stanie obiektu jest on modyfikowany a obiekt ewentualnie jest do słownika obiektów ruchomych innych użytkowników.

Ustawienia początkowe aplikacji są realizowane w funkcji (InteractionInitialisation() lub PoczątekInterakcji()). Tworzone są między innymi obiekty sieciowe. W argumentach konstruktora podawany jest adres IP wirtualnej grupy roboczej - identyfikujący uczestników grupy.

Obsługa bieżącej symulacji i interakcji jest realizowana w funkcji (VirtualWorldCycle() lub Cykl\_Wirtualnego\_Swiata()), w której jest na bieżąco obliczany średni czas cyklu służący do synchronizacji zdarzeń w wirtualnym świecie. Wywołana jest również symulacja obiektu a informacja o nowym stanie obiektu przesyłana jest do innych użytkowników.

### Usuwanie blokady multicast w Windows 10:

Po uruchomieniu wielu aplikacji na jednym komputerze powinny się one komunikować pomiędzy sobą. W przypadku różnych komputerów, komunikacja jest możliwa po wyłączeniu zapory sieciowej (z opcji Panel sterowania/ Zapora sieciowa/ Włącz lub wyłącz zaporę należy wybrać „wyłącz zaporę sieciową”). Ponadto w przypadku systemu Windows 10 należy dodatkowo zainstalować protokół multimesji wykonując następujące czynności:

- Wejść w Panel Sterowania / Sieć i internet / Centrum sieci i udostępniania / Zmień ustawienia karty sieciowej
- Wybrać odpowiedni interfejs (połączenie WiFi lub inne)
- Wybrać właściwości
- W składnikach zaznaczyć “Protokół multipleksa karty sieciowej firmy Microsoft”
- Nacisnąć przycisk Zainstaluj...
- Wybrać Protokół
- Nacisnąć przycisk Dodaj...
- Upewnić się że zaznaczony jest protokół multimesji
- Zatwierdzić przyciskiem OK

### **Zadania:**

1. Wybór miejsca dla nowego pojazdu, tak aby zjawiał się w miejscu możliwie najmniej widocznym przez innych użytkowników (zakładając, że mają ustawiony widok z kokpitu) lub jak najdalszym od innego, najbliższego położonego pojazdu. Odległość można obliczyć licząc długość wektora pomiędzy środkami pojazdów. W przypadku terenu płaskiego należy pominąć składową y. Można losować położenie pojazdu np. 10 razy, a następnie wybrać położenie najbardziej odległe od innych użytkowników.

2. System logowania pozwalający na identyfikację użytkownika, tak aby stan obiektu własnego po ponownym uruchomieniu programu na tym samym lub innym komputerze był taki sam (należy więc tę informację przechowywać inaczej niż np. na dysku lokalnym). Identyfikacja użytkownika - uwierzytelnienie (system logowania) może być uproszczone do podawania ostatniej cyfry numeru ID.
3. System głosowania: przyjęcie każdego nowego uczestnika grupy roboczej wymaga uzyskania większości głosów pozostałych uczestników. Aplikacja nowego uczestnika nie może uczestniczyć w liczeniu głosów ani w podejmowaniu decyzji o przyjęciu.
4. Zamiana pojazdów pomiędzy dwoma użytkownikami na ich życzenie (potrzebna jest do tego celu ramka specjalnego typu). Wybór pojazdu innego użytkownika może odbywać się poprzez podanie jego numeru ID, znalezienie się w małej odległości lub w inny sposób zapewniający jednoznaczność wyboru. Niedopuszczalna jest możliwość wystąpienia sytuacji, gdy jeden z użytkowników się zamienił, a drugi nie.
5. Wykrywanie kolizji (np. poprzez porównywanie położenia pojazdów -  $vPos$ ) i poinformowanie każdego z uczestników. Pojazdy biorące udział w kolizji powinny się zatrzymać. Kontynuacja ruchu powinna być możliwa tylko w kierunku, w którym kolizja nie jest pogłębianą. Uwaga, należy sprawdzić, czy funkcja zadziała przy dużych prędkościach ruchu obiektów. Kształty obiektów można uprościć do sfer opisujących te obiekty, np. korzystając ze wzoru  $(vPos1 - vPos2).length() < radius1 + radius2$ . Moment kolizji powinien zostać uzgodniony przez obie aplikacje lub z wykorzystaniem trzeciej aplikacji. Niedopuszczalna jest sytuacja, gdy jeden z użytkowników uznał kolizję, a drugi nie.
6. Usuwanie nieaktywnych obiektów w dwóch przypadkach: gdy dostatecznie długo (np. przez 10 s.) nie ma łączności z powiązanymi z nimi aplikacjami (*timeout*) lub gdy użytkownik zamknie aplikację - można wówczas zaprogramować wysyłanie specjalnego komunikatu o zamknięciu.