

# Protokół sieciowy

## Opis koncepcyjny

Gra funkcjonuje w topologii klient-serwer. Przy konstrukcji protokołu zdecydowano się na strukturę korzystającą z obiektów komunikatów obsługiwanych przez strumienie bitów biblioteki RakNet. Protokół definiuje pewien zbiór komunikatów oraz stanów aplikacji, odpowiadających za prawidłowy przebieg konwersacji pomiędzy aplikacjami będącymi częścią systemu obsługującego toczącą się rozgrywkę.

Aplikacja serwer oczekuje na graczy w stanie zbierania graczy w lobby. Tylko na etapie zbierania graczy w lobby nowi gracze mogą zostać przyłączeni do rozgrywki. Na każdym etapie gracz może rozłączyć się z serwerem, zostać usunięty ze względu na timeout lub serwer może zostać wyłączony. W pierwszych dwóch przypadkach serwer usuwa graczy z tablicy alokacji, pozostali gracze otrzymują o tym informację, a program kliencki reaguje na zdarzenie poprzez usunięcie gracza oraz ewentualnie modelu jego pojazdu z rozgrywki. W drugim przypadku gracze otrzymają informację sieciową biblioteki RakNet o rozłączeniu z serwerem i nie będą w stanie kontynuować konwersacji sieciowej.

Po połączeniu się z serwerem gracz otrzymuje unikalne id i zostaje odpytany o alias – ciąg znaków o długości od 1 do 20 znaków. Jeżeli alias nie jest unikalny – inny gracz korzysta już z podanego aliasu - gracz otrzymuje odpowiedź o błędzie i nie zostaje włączony do rozgrywki. Wiadomość o przyłączeniu się nowego gracza jest rozsyłana do pozostałych graczy. W lobby pomyślnie zalogowani gracze mogą przysyłać sobie wiadomości tekstowe w postaci komunikatów rozsyłanych do pozostałych przez serwer. Każdy z graczy w lobby musi przed przystąpieniem do rozgrywki dać serwerowi sygnał gotowości do rozgrywki, który jest rozsyłany do innych graczy (w celu wyświetlenia o tym fakcie informacji na ekranie graczy). Kiedy gracz serwerowy uzna, że rozgrywka jest już możliwa oraz wszyscy gracze będą gotowi (o czym decyduje stan gotowości na serwerze), przesłany zostaje komunikat startujący grę. Serwer zbiera odpowiedzi na ten komunikat i jeżeli otrzymał odpowiedzi od wszystkich klientów, następuje przesłanie informacji o inicjalnych pozycjach graczy oraz rozpoczęciu odliczania. Następnie gracze otrzymują od serwera w regularnych odstępach trzy komunikaty zmiany świateł odliczających czas do rozpoczęcia wyścigu – odpowiada to zmianie świateł rozpoczynających wyścig na czerwone, żółte i zielone. Zmiana światła na zielone oznacza jednocześnie przejście do właściwej rozgrywki. Podczas właściwej rozgrywki każdy z graczy może wysłać powiadomienie o swojej akcji do serwera, które gracz serwerowy roześle dalej. Serwer rozsyła również okresowo wiadomość o wszystkich istotnych elementach sceny z jego punktu widzenia – jako decydującego. Kiedy po stronie serwera jeden z graczy zostanie wyeliminowany, gracze zostają o tym poinformowani. Kiedy po stronie serwera jeden z graczy dojedzie do mety, gracze zostają o tym poinformowani. W przypadku planowego rozłączenia serwera gracze otrzymują o takim fakcie powiadomienie biblioteki RakNet, jeżeli serwer rozłączy się w wyniku błędu lub zamknięcia go nie z poziomu aplikacji, gracze otrzymają powiadomienie o timeoucie dopiero po pewnym czasie.

## Komunikaty systemowe

Poniżej przedstawiono listę komunikatów aplikacji. Komunikaty mogą być powiadomieniami (ACK) lub funkcjonować na zasadzie żądania/odpowiedzi (REQUEST/REPLY). Pierwszy z typów oznacza, że komunikat jest przesyłany tylko w jedną stronę, drugi – że komunikat jest wysyłany z prośbą o odpowiedź, bez której nie nastąpi pewne zachowanie strony proszącej o odpowiedź.

**ID\_PLAYER\_ALIAS** – (REQUEST/REPLY) – żądanie jest prośbą o alias użytkownika, zawiera numer porządkowy gracza przypisany przez serwer. Jest wysyłane tuż po połączeniu się nowego klienta z serwerem. Odpowiedź zawiera alias użytkownika oraz numer użytkownika, który zwraca swój alias. Powinna być przesyłana tuż po otrzymaniu żądania aliasu.

**PLAYER\_ALIAS\_VERIFICATION** – (ACK) – jest to powiadomienie o tym, czy podany uprzednio alias nie jest jeszcze wykorzystywany w rozgrywce (status komunikatu false oznacza brak możliwości gry pod danym aliasem ze względu na jego zajętość). Jeżeli podano prawidłowy alias wiadomość zawiera również informacje o ilości graczy, nazwach graczy, ich numerach porządkowych oraz stanie ich gotowości do rozgrywki. Wiadomość weryfikacji jest przesyłana tuż po otrzymaniu odpowiedzi na ID\_PLAYER\_ALIAS.

**LOBBY\_MSG** – (ACK) – wiadomość przesyłana w lobby wraz z numerem gracza, który przesyła wiadomość. Jest przesyłana tylko w stanie *lobby*, po wysłaniu przez gracza nowej wiadomości lobby.

**ANOTHER\_PLAYER\_ACCEPTED** – (ACK) – powiadomienie o przyłączeniu się nowego gracza do rozgrywki. Zawiera numer porządkowy gracza oraz jego nazwę. Jest przesyłana przez serwer tuż po pozytywnej weryfikacji nazwy nowego użytkownika.

**ANOTHER\_PLAYER\_QUIT** – (ACK) – powiadomienie o timeoucie bądź opuszczeniu rozgrywki przez innego klienta. Zawiera numer gracza opuszczającego rozgrywkę. Może być wysłana/otrzymana w każdym stanie i skutkuje usunięciem gracza i ewentualnie modelu jego pojazdu z rozgrywki.

**GET\_READY** – (ACK) – powiadomienie o zmianie stanu gotowości gracza. Zawiera numer porządkowy gracza zmieniającego stan gotowości oraz stan gotowości w postaci wartości logicznej – false oznacza brak gotowości, true oznacza gotowość do rozgrywki. Przesyłana w przypadku zmiany przez gracza stanu gotowości – tylko w stanie *lobby*.

**GAME\_START** – (REQUEST/REPLY) – żądanie oznacza chęć uruchomienia rozgrywki przez gracza serwerowego – przesyłany tylko do klientów, z których każdy powinien odesłać odpowiedź, co oznacza, że serwer może kontynuować rozpoczęcie rozgrywki. Odpowiedź zawiera numer gracza, który potwierdza gotowość do rozgrywki. Jeżeli nie wszyscy klienci odpowiedzą na żądanie, gra nie może zostać uruchomiona i wracamy do lobby, gdzie prawdopodobnie nastąpi timeout jednego z graczy.

**GAME\_START\_VERIFIED** – (ACK) – wiadomość przesyłana przez serwer oznaczająca, że wszyscy klienci potwierdzili gotowość do rozgrywki (odpowiedzieli na GAME\_START). Po otrzymaniu/wysłaniu komunikatu następuje przejście aplikacji serwera/klienta do stanu *PREGAME*.

**PREGAME\_LIGHT** – (ACK) – wiadomość dla klientów o zmianie światła odliczającego czas do rozpoczęcia wyścigu. Zawiera informację o świetle, gdzie 0 to światło czerwone, 1 to światło żółte, 2 to światło zielone. Wartość 2 oznacza jednocześnie zlecenie przejścia do stanu *GAME*. Może być otrzymana/wysłana jedynie w stanie *PREGAME*.

**PLAYER\_ACTION** – (ACK) – informacja o akcji gracza spośród dostępnych – skrętu w lewo, prawo, przyspieszenie, hamowania oraz strzału. Zawiera numer gracza wykonującego akcję oraz identyfikator akcji (określony w odpowiednim typie wyliczeniowym). Komunikat jest propagowany

do pozostałych graczy. Może zostać wysłany/otrzymany tylko w stanie *GAME*.

**PLAYER\_RACE\_FINISHED** – (ACK) – informacja o ukończeniu przez pewnego gracza wyścigu. Zawiera identyfikator tego gracza oraz pozycję, na której ukończył wyścig. Może zostać wysłana jedynie przez serwer na podstawie jego informacji o scenie – oraz jedynie w stanie *GAME*.

**PLAYER\_ELIMINATED** – (ACK) – informacja o wyeliminowaniu gracza z wyścigu poprzez zniszczenie jego pojazdu. Zawiera identyfikator tego gracza. Może zostać wysłana jedynie przez serwer na podstawie jego informacji o scenie – oraz jedynie w stanie *GAME*.

**GAME\_SYNCHRO** – (ACK) – całościowa informacja o istotnych elementach sceny (rozmieszczeniu pojazdów, punktach życia poszczególnych graczy) przesyłana okresowo przez serwer w stanie *GAME*.

## Stany systemowe

**INIT** – stan inicjalizacyjny, występuje tylko po stronie klienta. W tym stanie klient porozumiewa się z serwerem w celu zostania zakwalifikowanym do rozgrywki. Ustalany jest alias klienta. Jeżeli ustalony alias jest wolny (nie jest wykorzystywany po stronie serwera przez innego gracza), klient zmienia stan na *LOBBY*.

**LOBBY** – występuje po stronie klienta i serwera. Stan początkowy serwera. Jest to stan, w którym klienci są gromadzeni w lobby, gdzie mogą wysyłać i odbierać komunikaty tekstowe od innych graczy. Kończy się po stronie serwera wraz z wysłaniem przez serwer prośby o potwierdzenie gotowości do rozpoczęcia rozgrywki *GAME\_START*, po stronie klienta wraz z otrzymaniem komunikatu *GAME\_START\_VERIFIED*.

**SERVER\_GAME\_INIT** – występuje jedynie po stronie serwera. Jest to stan, w którym serwer zbiera odpowiedzi klientów o gotowości do rozgrywki (na komunikat *GAME\_START*). Jeżeli nie zjawią się po określonym czasie, serwer wraca do stanu *LOBBY*. Jeżeli zebrano wszystkie odpowiedzi, następuje przejście do stanu *PREGAME*.

**PREGAME** – występuje po stronie serwera i klienta. Stan odliczania czasu do wyścigu za pomocą sygnalizacji świetlnej. Po wysłaniu informacji o wystąpieniu zielonego światła automatycznie zmieniany na stan *GAME*.

**GAME** – występuje po stronie serwera i klienta. Stan rozgrywki, podczas którego gracze uczestniczą we właściwym wyścigu.

## Aplikacja testowa protokołu

W dystrybucji pakietu opisującego protokół sieciowy zawarto testową aplikację ukazującą przebieg przykładowych konwersacji między serwerem a klientami. W celu zaobserwowania przykładowej konwersacji należy uruchomić jedną instancję serwera oraz co najmniej jedną instancję klienta aplikacji prezentującej protokół. Aplikacja serwera nasłuchuje na lokalnym gnieździe, a klienci łączą się z localhost, przykładową konwersację prowadzi się więc lokalnie. Aplikacja wypisuje na standardowe wyjście wszelkie wystąpienia komunikatów opisanych powyżej. Ze względu na fakt, że aplikacja prezentująca protokół nie jest docelową implementacją protokołu należy liczyć się z licznymi uproszczeniami, które nie zaburzają komunikacji jednocześnie umożliwiając obserwację

przykładowej konwersacji. W docelowej implementacji jednak uproszczenia przykładowej aplikacji są nie do przyjęcia.