**Podstawy chematu do multiplayera**

Script component:

Np. **ShipControllerInfo**

GameObject / Prefab: **Ship**

Pusty GameObject ze skryptem np. **LocalPlayerController** który jako parametr przyjmuje element **ShipControllerInfo**

Pusty GameObject ze skryptem np. **NetworkPlayerController** który jako parametr przyjmuje element **ShipControllerInfo**

Pusty GameObject ze skryptem np. **AIController** który jako parametr przyjmuje element **ShipControllerInfo**

Script component:

Np. **ShipController**

Statek będzie mógł być sterowany na 3 różne sposoby: przez gracza lokalnego (klawisze z klawiatury), przez gracza sieciowego (info o sterowaniu, położeniu itd. przekazywane przez serwer) oraz proste AI do sterowania lokalnymi statkami (te które nie są obecnie sterowane przez gracza lokalnego).

Zatem gracze **LocalPlayerController** i **AIController** będą zarządzane i ustawiane przez komputer gracza lokalnego więc informacje o ich ruchu muszą być wysyłane na serwer.   
Gracz **NetworkPlayerController** jest zarządzany przez serwer i on tylko zbiera informacje wysyłane przez serwer do gracza (z technicznego punktu widzenia w przypadku gracza sieciowego nie będzie miało znaczenia czy to jest **AI** czy **LocalPlayer** po drugiej stronie, czyli w przypadku walki 4v4 jeden gracz będzie wysyłał do drugiego informacje o 3x **AI** i 1x **LocalPlayer** a drugi odbierał to jako 4x **NetworkPlayer**)

Każdy z trzech graczy w praktyce steruje jednym statkiem i robi to pośrednio poprzez **ShipControllerInfo** który jest skryptowym komponentem całego GameObjectu statka.  
**ShipControllerInfo** powinien zawierać tylko i wyłącznie informacje o „kontrolkach” czyli np. informacje zero-jedynkowe o poziomie wciśniętych klawiszy (np. WASD) oraz zmienne/funkcje umożliwiające szybki dostęp do parametrów opisujących ruch i położenie statku (**velocity**, **angularVelocity**, **transform** etc) wraz z możliwością ich sforcowania (ustawienia).

Tak jest ponieważ **ShipControllerInfo** posłuży tak naprawdę jako zbiór parametrów które będą synchronizowane pomiędzy graczami, dlatego każdy z powyższych trzech graczy będzie tylko ustawiał wartości w **ShipControllerInfo** odnośnie kontrolek (np. wartość float od -1 do +1 odpowiadające skręcaniem (wciśniętym klawiszom A-D)), a wszelkie obliczenia fizyki, ruchu, obliczania dodatkowych parametrów etc. będą wykonywane w **ShipController** którego obliczenia będą zależały tylko i wyłącznie od wartości pobieranych z **ShipControllerInfo**.  
Dlatego jeśli **ShipController** będzie obliczał sobie jakieś dodatkowe informacje potrzebne do ruchu (np. siła skręcania która rośnie wraz z czasem trzymania przycisku czy coś) to powinna być możliwość synchronizowania (pobierania i ustawiania) tej wartości poprzez **ShipControllerInfo**.