Toelichting op UML voor KernDev4 Multiplayer
Ruben Bergshoeff, Yassine Minjon

Het multiplayer systeem bestaat bij de gratie van de Client, ClientConnection, GameManager en ServerConnection classes, dus dezen zal ik hier uitleggen. Andere classes zijn relevant voor de werking van de game maar staan los van netwerk avonturen.

ServerConnection *(overwegend Yassine)*
Deze class verzorgt de verbinding met de SQL database. Door gebruik te maken van Actions als parameter voor het opvragen van informatie hoeft de class zelf niet te weten wie of wat de informatie opvraagt. Zolang de geleverde method voldoet aan de eisen krijgt deze het antwoord terug. Bovendien is het een ScriptableObject based class, waardoor we makkelijk instanties die deze class nodig hadden konden verbinden aan dezelfde instance binnen de inspector zonder te goochelen met Singletons.

ClientConnection *(50/50)*
Deze class is de stabiele basis die de Unity clients verbindt met de serverside informatie die wij bijhouden. De kern-method hierin is de Get(…) method. Deze checkt of er al een ClientConnection instantie op de server bestaat gelinkt aan de ip van de Unity client. Als dit het geval is dan wordt deze hieraan gelinkt, zo niet dan creëren we een nieuwe.

GameManager *(Ruben)*
Om ons doel van losse game-setup files door middel van een SQL database te kunnen faciliteren moest ergens al deze data worden opgehaald. Dit gebeurt in de GameManager, die daarna ook het verloop van de verdere game aanstuurt. Hij verbindt zich dan ook aan de ClientConnection singleton om updates te krijgen over de staat van clients, en stuurt deze clients de nodige informatie om ze up to date te houden met de staat van het spel.
Een leuk component om naar te kijken is hier de HackManager, deze zet een sessie op die twee ClientConnection instances in een gezamenlijke sessie stopt. Deze wordt weer opgeheven als beide instances de game op dezelfde manier hebben opgelost.

Client *(overwegend Ruben)*
In deze class bevind zich de grootste hoeveelheid TCP grappen en data jongleren. Doordat we de games moeten kunnen communiceren van de server naar de client moet er flink wat data heen en weer. Om te voorkomen dat we te grote data pakketten kregen hebben we een systeem dat elke game en pakket om de beurt opvraagt en doorgeeft. Dit proces start in de LoadGames(…) method.
Verder is een belangrijk component de gelinkte ClientGameHandler, deze verwerkt de callbacks van de server op het moment dat de staat van de game verandert.