

# Proof of concept

---

MMORPG ARCHITECTUUR – 2D TOPDOWN MEDIEVAL

Finn Alberts, Aaron Dang, Laurent Dassen en  
Peter Derks  
ZUYD HOGESCHOOL | HBO ICT



## Inhoud

1. Inleiding.....	3
2. Doel .....	3
3. Resultaat .....	3
4. Discussie.....	3
5. Unity Instructies .....	4
Foutmelding tijdens het openen.....	4
Parrelsync.....	4
Build .....	5

## 1. Inleiding

Binnen dit project wordt een architectuur ontworpen voor een Massive Multiplayer Online Role-Playing Game (MMORPG). Deze architectuur wordt ontworpen middels Kruchten 4 + 1, maar dient ook middels een proof of concept getest te worden, om te zien of datgene wat ontworpen is ook in de praktijk werkt.

## 2. Doel

Binnen het proof of concept wordt de communicatie tussen de client en game-server aangetoond. Daarnaast wordt ook het wisselen tussen locaties (die los van elkaar staan) gedemonstreerd in het proof of concept. Hierbij is het voornaamste doel om te zien of de uitgewerkte architectuur hiervoor in de praktijk werkt. Verder zal het proof of concept aantonen hoe de server altijd 'de waarheid' bevat en dat de input van de client(s) niet als betrouwbaar wordt aangenomen. Daarom heeft de server naast het synchroniseren van data tussen clients ook een controlerende/corrigerende rol. Deze rol zal ook in het proof of concept worden aangetoond.

## 3. Resultaat

Het proof of concept is ontworpen met de Unity game engine en maakt gebruik van Mirror Networking (open source netwerkoplossing voor Unity). Binnen het prototype wordt het event-driven systeem van de architectuur aangetoond. Hierbij worden de acties van de gebruikers als events aangeroepen en worden deze op de server berekend en eventueel gecorrigeerd. Verder worden de willekeurige acties (zoals het toepassen van schade binnen een bepaalde range of het afhandelen van critical hits) op de server afgehandeld. Ook bevat het prototype een component (load and distribute manager) voor het bijhouden van de hoeveelheid spelers die online zijn en op welke locatie deze zich bevinden. Met dit proof of concept wordt aangetoond dat de ontworpen client-server communicatie geschikt is voor een MMORPG. Daarnaast wordt binnen het proof of concept een basis gelegd voor toekomstige schaalbaarheid van de MMORPG, door het bijhouden van huidige spelers en bijbehorende locaties kunnen er acties ondernomen worden voor het op- of afschalen van de resources wanneer bepaalde spelerlimieten worden overschreden.

Het proof of concept bevat op dit moment een ingebouwde load and distribute manager, in de praktijk zal dit een losse API moeten worden die bijhoudt welke servers er online staan met de corresponderende server adressen. Zo kan de client wanneer deze een connectie wil opzetten met een server een request naar de API sturen die vervolgens een server adres terugstuurt. De client zal vervolgens met dit server adres een connectie maken.

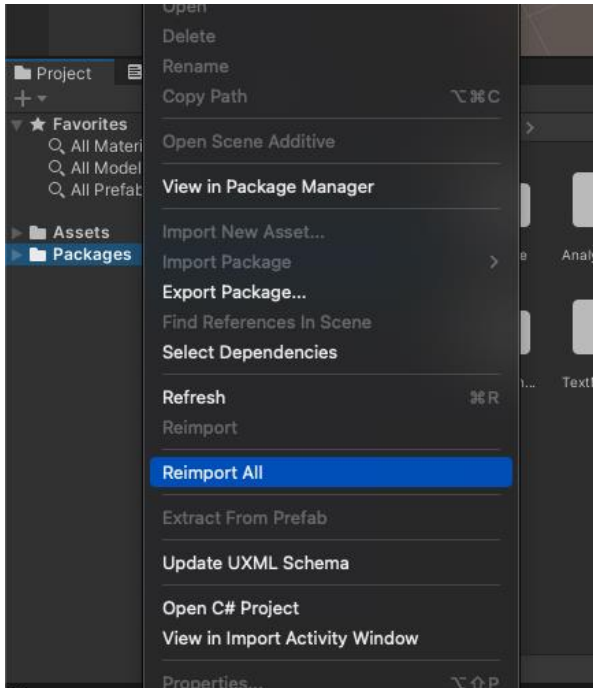
## 4. Discussie

Vanwege budgettaire limieten is ervoor gekozen om geen daadwerkelijke servers online te zetten en is alles binnen het prototype lokaal getest. Hierdoor is niet de invloed van verschillende latenties getest binnen het prototype. Ondanks dat het systeem op kleine schaal is getest (met 5 spelers getest), is het de verwachting dat het systeem ook op grotere schaal zal werken, vanwege de beperkte load op de server. Toch zal het systeem nog getest moeten worden met een grotere load op de server zodat dit ook daadwerkelijk bewezen kan worden. Verder zijn er ook imitaties aan Unity en Mirror verbonden, binnen Mirror is het niet mogelijk om verschillende gebruikers binnen verschillende scenes te laden. Hierdoor is het niet mogelijk om verschillende gebruikers op andere locaties op dezelfde server te laden (alle gebruikers moeten zich binnen dezelfde scene bevinden). Verder is Unity niet de meest efficiënte engine qua performance en zal er voor de performance die nodig is voor de schaal van een MMORPG gekeken kunnen worden naar andere game engines.

## 5. Unity Instructies

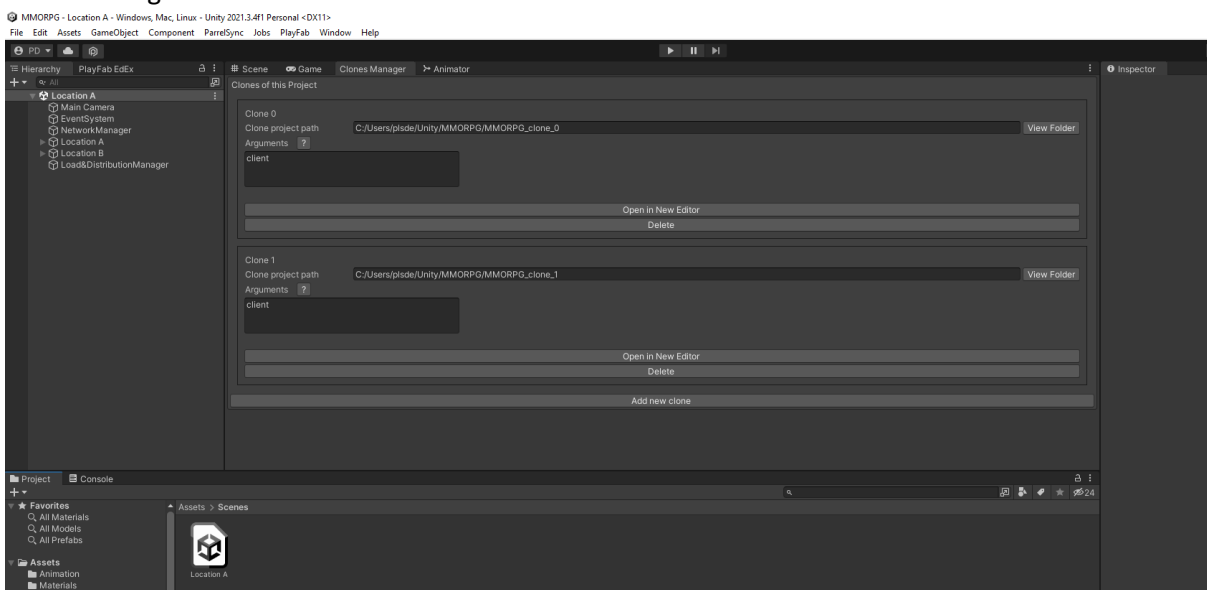
### 5.1 Foutmelding tijdens het openen

Tijdens het voor de eerste keer openen van het project met de Unity Editor kan het voorkomen dat Unity een foutmelding geeft en vraagt of het project in de “Safe mode” moet worden gestart. Kies hier voor “Ignore”. Er zullen nu veel foutmeldingen in de console staan. Om deze op te lossen, ga naar het “Project”-tab, klik met rechts op “Assets” en kies voor “Reimport all”.



### 5.2 Parrelsync

Parrelsync is een sdk voor het runnen van meerdere instanties van het unity voor het testen van multiplayer functionaliteiten. Per instantie kan het project gekloond worden. De clones manager kan binnen de unity editor geopend worden. Hierin kunnen de clones worden geopend of nieuwe clones worden aangemaakt.



### 5.3 Build

De controls van de game zijn als volgt:

- Pijltjes toesten voor het bewegen
- Z voor het nemen van damage op de speler
- X voor het aanvallen (schieten van een pijl)

De build van het prototype staat op de github repository onder releases.  
(<https://github.com/JellybeanKing/MMORPG>)