

# Game Development

## Kernmodule 4 Networking

Mick Teunissen

Github: <https://github.com/MicksTape/GameDev4-Networking>

Inhoud:

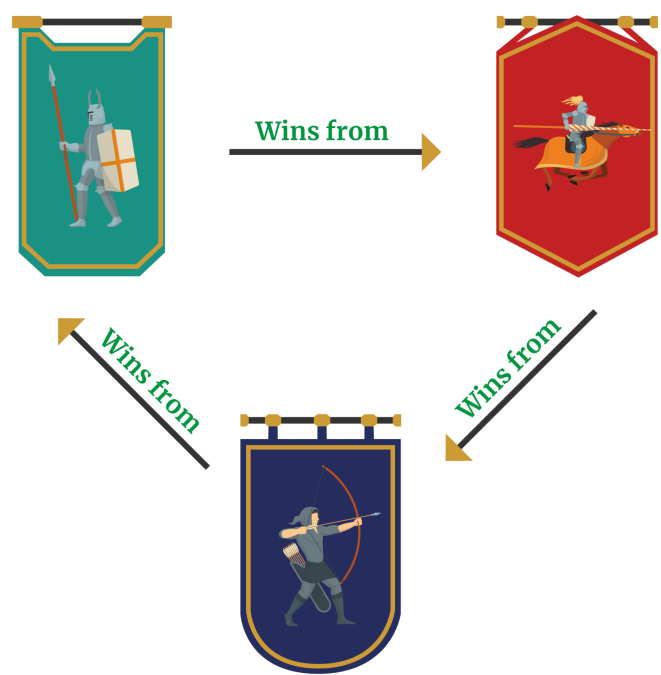
- Beschrijving Game Concept
- Beschrijving Networking en implementatie
- Technische uitwerking
  - Herkansing - Session
- Message Flow / Sequence diagram
- UML
- Bronnen



























### Beschrijving Game Concept

Fentearis is een 2D-game vergelijkbaar met "steen-papier-schaar", maar met een ander thema. De speler neemt de rol aan van een koning die moet beslissen welke troepen hij naar het slagveld stuurt om de andere koning(speler 2) te verslaan. Als er geen keuze wordt gemaakt, zal de generaal namens de koning beslissen. Er zijn drie soorten troepen in het spel, waarbij elke soort de ene kan verslaan en de andere niet, vergelijkbaar met "steen-papier-schaar". Het doel is om zo vaak mogelijk van je tegenstander te winnen om punten te behalen. Het ultieme doel is natuurlijk om de hoogste score te behalen en zo veel mogelijk potjes te spelen.

Win Conditioies



Unit 1	Unit 2	Result
 Infantry	 Infantry	Draw
 Infantry	 Archer	 Infantry Loses
 Infantry	 Cavalry	 Infantry Wins
 Archer	 Infantry	 Archer Wins
 Archer	 Archer	Draw
 Archer	 Cavalry	 Archer Loses
 Cavalry	 Infantry	 Cavalry Loses
 Cavalry	 Archer	 Cavalry wins
 Cavalry	 Cavalry	Draw

# Beschrijving networking implementatie

## Server Login

Hier logged de speler in op de server. En wordt er een sessie ID aangemaakt.

## Inloggen

Hier kan de speler inloggen met zijn wachtwoord en gebruikersnaam. Er wordt gecontroleerd of de username bestaat en het wachtwoord hiermee overeenkomt.

## Register

Hier kan de speler zichzelf registreren door zijn eigen gebruikersnaam en wachtwoord in te vullen. Hier wordt op meerdere checks uitgevoerd zodat niemand met mijn database probeert te klooien. Deze checks zijn onder andere:

- Of de username als bestaat
- Of de 2 wachtwoorden overeenkomen
- Of het aantal karakters tussen de 4 en 10 ligt
- Sanitize door middel van `mysqli_real_escape_string`

Als er iets verkeerd gaat wordt er een error message weergegeven met de bijbehorende error. Klopt alles en voldoet alles aan de regels worden de gegevens geüpdate in de database en kan de speler inloggen op de inlog pagina.

## Account Settings

Hier krijgt de speler de mogelijkheid om zijn gebruikersnaam en wachtwoord te veranderen. Als de speler in unity op de knop drukt zal de standaard webbrowser automatisch openen met de bijbehorende webpagina om deze gegevens aan te passen. Hierbij maak ik gebruik van een filter die de spaties weghalen bij de ingevulde gegevens. Wanneer ze dit aanpassen wordt het direct geüpdatet in phpmyadmin en wordt het ook weer direct in unity geüpdatet zodat je meteen je aangepaste gebruikersnaam/wachtwoord kan zien/gebruiken.

## Highscores

Spelers worden hier net als bij account settings doorgestuurd naar de webbrowser om de highscores te bekijken. Hier is een overzicht te zien van de top 5 spelers en een overzicht van de meest actueel gespeelde games en wie deze gewonnen heeft. Dit wordt na iedere gespeelde game geüpdate door middel van de leave button.

## Lobby Host/Client

Hier kan de speler kiezen of hij/zij de host of client wil zijn. De client kan joinen op basis van het IP adres van de host. Bij een lokale verbinding op bijvoorbeeld dezelfde computer/netwerk gebruik je "127.0.0.1". Zodra er verbinding is gemaakt kan de game beginnen.

## Game

Hier kunnen beide spelers een unit kiezen binnen 10 seconden. Wanneer beide een unit gekozen hebben, wordt de score naar de server(host) gestuurd en wordt op het volgende scherm de winnaar bekendgemaakt en wordt 1 score toegevoegd aan de winnende speler van die ronde. Hierna herhaald het proces zich

## Technische uitwerking

De game maakt gebruik van het Unity Transport System. De server en de client kunnen gegevens uitwisselen door middel van game events(net messages). Zowel de server als de client hebben hun eigen specifieke net message, zoals bijvoorbeeld "NetWelcome". Deze net messages erven van de overkoepelende class "NetMessage".

Wanneer de client een game event wil verzenden, worden de bijbehorende gegevens geserialised en vervolgens naar de server verzonden. Daar worden de gegevens weer omgezet naar hun oorspronkelijke gedeserialiseerde vorm. Dit geldt ook vice versa van de server naar de client.

Gegevens zoals scores, gebruikersnamen, wachtwoorden, enzovoort, worden opgeslagen in de game manager die gebruikmaakt van een "singleton". Hierdoor kan de game manager efficiënt en consistent toegang krijgen tot de gegevens. Bovendien is het ook mogelijk om deze gegevens op te halen vanuit een JSON-array door middel van een webrequest naar de PHP-database.

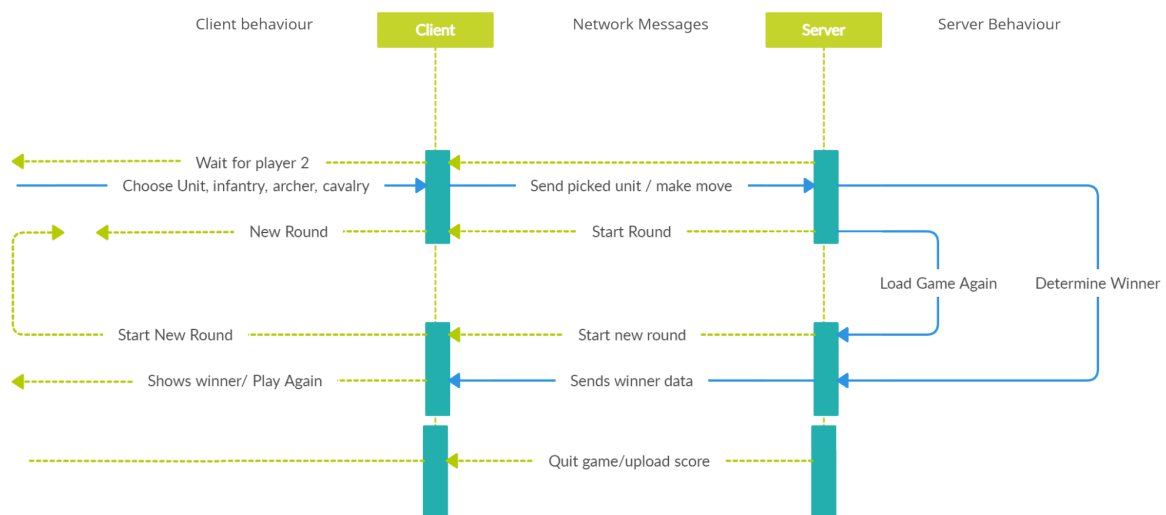
### Herkansing - Session

Aan de hand van de feedback heb ik een Session-ID geïmplementeerd in mijn game. Zodra een speler succesvol inlogt op de server, wordt er automatisch een unieke Session-ID gegenereerd en toegewezen. Deze Session-ID werkt als een sleutel waarmee de speler toegang krijgt tot verschillende functionaliteiten binnen de game. Deze functionaliteiten omvatten onder andere de EditAccountpagina, de Highscorepagina en het opslaan van scores. De Session-ID zorgt ervoor dat alleen geautoriseerde spelers met geldige inloggegevens toegang hebben tot deze game functies.

## Message Flow / Sequence Diagram

De flow van de game ziet er als volgt uit:

1. Start Game
2. Server Login speler krijgt Session Id
3. Register of login
4. Kies Host of Client
5. Speler Kiest een Unit
6. De speler wacht tot de timer is afgelopen of de andere speler heeft een unit heeft gekozen
7. De winnaar wordt bekend gemaakt en krijgt een punt, bij een draw krijg niemand punten
8. De nieuwe ronde begint weer opnieuw. Dit proces herhaalt zich tot een speler de game verlaat.



# UML



## Bronnen:

<https://docs.unity3d.com/Packages/com.unity.transport@1.0/manual/index.html>

<https://docs.unity3d.com/2020.3/Documentation/Manual/UNetMessages.html>

<https://docs.unity3d.com/Packages/com.unity.transport@0.3/api/Unity.Networking.Transport.DataStreamReader.html>

De 2d assets zijn een combinatie van mijn eigen gemaakte assets in Illustrator en assets van Freepik.

macrovector:

[https://www.freepik.com/free-vector/medieval-decorative-icons-set\\_3862010.htm#query=medieval%20knight&position=13&from\\_view=search&track=ais](https://www.freepik.com/free-vector/medieval-decorative-icons-set_3862010.htm#query=medieval%20knight&position=13&from_view=search&track=ais)

Omelapics:

[https://www.freepik.com/free-vector/castles-hills-medieval-landscapevector-cartoon-illustration\\_1053598.htm#query=medieval%20background%20vector&position=1&from\\_view=search&track=ais](https://www.freepik.com/free-vector/castles-hills-medieval-landscapevector-cartoon-illustration_1053598.htm#query=medieval%20background%20vector&position=1&from_view=search&track=ais)