

Documentatie Kernmodule 4

Sten Duindam

GDEV2

Studentnummer: 3019434

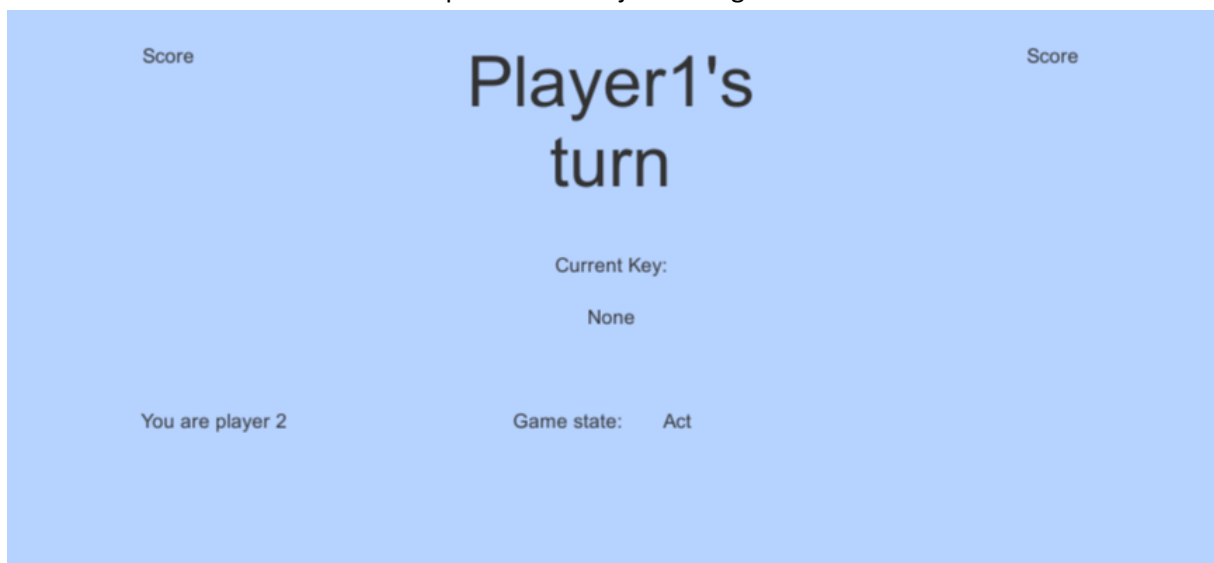
Rock Paper Scissors... Water?

Inleiding

Voor kernmodule 4 was het de bedoeling dat we een multiplayer game gingen maken. De randvoorwaarde waren dat het een turn-based game moet zijn en aan het einde van het spel moet er een score naar een online database verstuurd worden. In dit document beschrijf ik mijn spel, uitwerking en uitdagingen.

Game concept

Zoals de naam al doet vermoeden is het een turn based steen papier schaar variant. De twist in deze versie is dat het op reactie gebaseerd is. Beide spelers kunnen via multiplayer om de beurt een zet en een reactie doen. Hoe sneller beide spelers hierin zijn hoe hoger de scores kunnen worden.



De goede toetsen combinaties:

A verslaat S

S verslaat D

D verslaat W

W verslaat A

Wanneer de speler de juiste reactie toets invoert krijgt hij/zij een punt.

Controls:

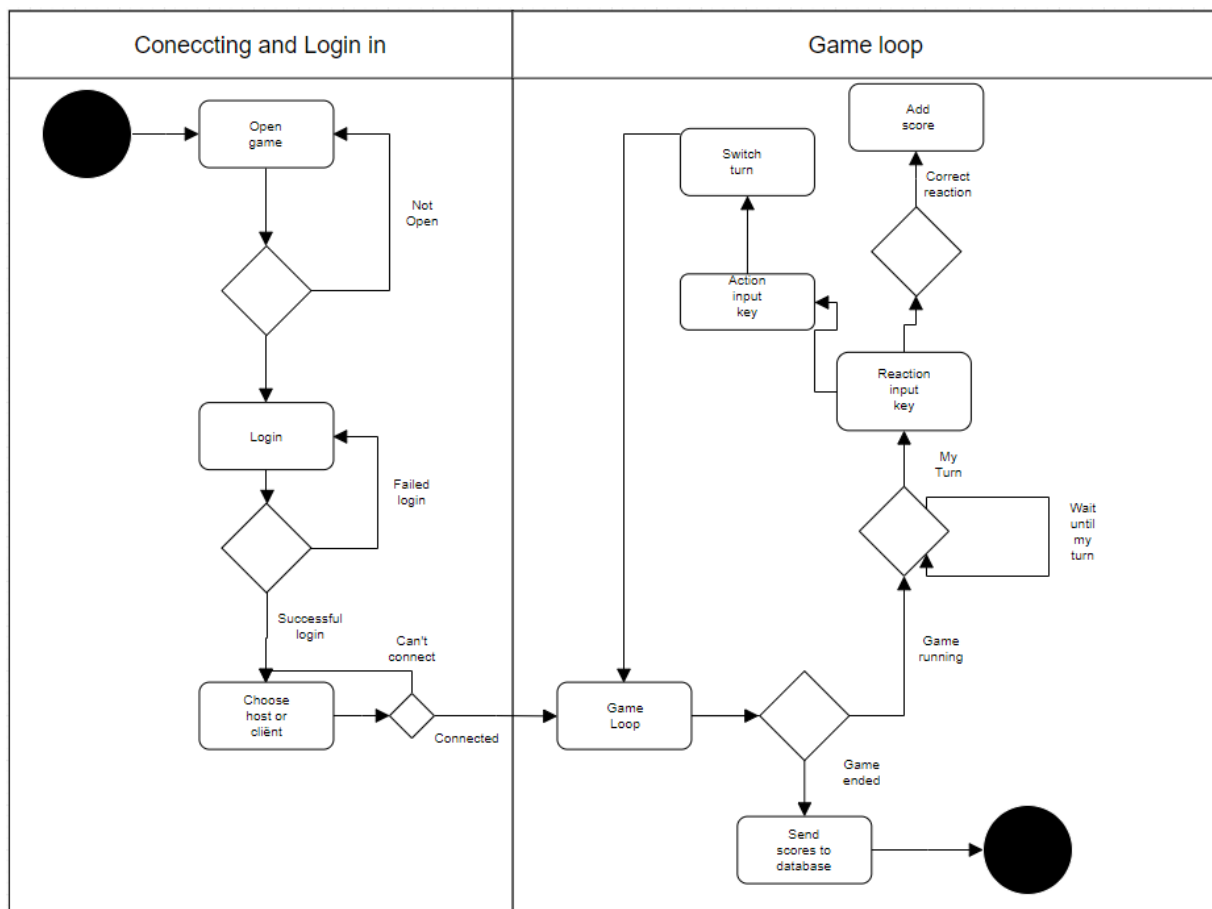
De speler geeft enkel input via de toetsen 'W' 'A' 'S' 'D'.

Multiplayer ontwerp

Middels drie scripts beheer en behandel ik de input, communicatie tussen server en cliënt en beurten. Deze drie scripts heten: TurnManager.cs, Manager.cs en PlayerTry.cs. De volgorde waarin deze communiceren is als volgt:

De speler geeft input aan de TurnManager. De TurnManager kijkt of het momenteel de beurt is van de speler. Als het de beurt is van de speler verwerkt hij de input en geeft een opdracht aan het speler object. Dit omdat alle communicatie via het speler object moet lopen in UNet. De speler geeft deze opdracht door aan de server die hem afhandeld en via een Rpc het resultaat naar alle cliënts stuurt.

De **activity diagram** hieronder laat de game flow zien.



TurnManager.cs

Om de beurten te regelen heb ik een static class gebruikt. Op deze manier hoeft er geen referentie te zijn tussen speler en de beurt manager. Wat uitstekend werkt aangezien de players pas geïnstantieerd worden op het moment dat de speler verbind met de server.

Deze TurnManager speelt alle communicatie weer terug naar het speler object dat contact zoekt met de TurnManager om te zorgen dat er een command gestuurd kan worden naar de server.

Manager.cs

Wanneer de speler met de server verbind wordt de CmdPlayerJoined functie aangeroepen op de server. Hier wordt de verbinding opgeslagen zodat wanneer de speler verbinding verbreekt en later weer terug komt wij een referentie hebben en het weer juist kunnen koppelen. Ook worden de player GameObjecten opgeslagen in een lijst. De eerste die verbind krijgt ook in deze functie de beurt en wordt als speler 1 gelabeld. Nu kunnen we vanuit de server eenvoudig bij de spelers komen en aangezien de spelers werken als enig communicatie middel tussen de server en cliënt kunnen we dus alles netjes behandelen. Denk aan UI en beurten.

Ook staan er Rpc's in dit script om alle UI en waarden gelijk te trekken onder iedere cliënt.

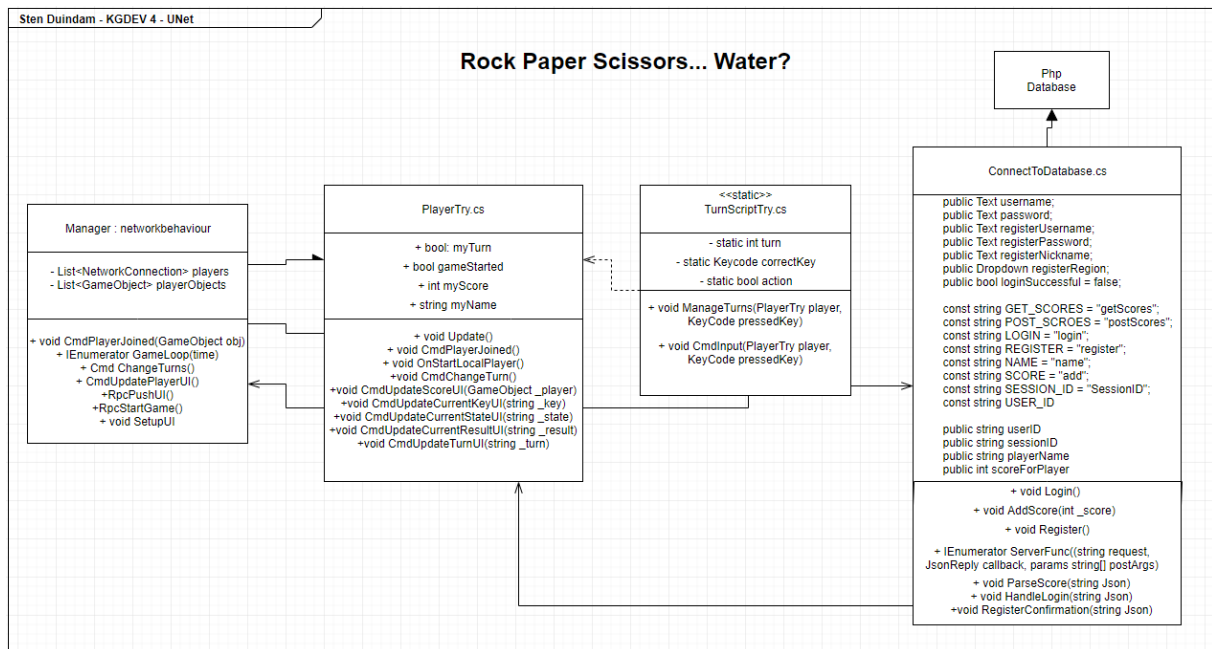
PlayerTry.cs

Dit script staat op het player object. Met local authority zorgt het ervoor dat er alleen input op het cliënt object ingevoerd kan worden en dus niet op alle andere player objecten.

Hier wordt de input van de speler afgehandeld en een verzoek ingediend bij de TurnManager om input te leveren op de server.

Dit script functioneert ook als communicatie punt met de server. Alle functies die een cliënt uit kan voeren staan hierop, dit zodat bijvoorbeeld de TurnManager.cs ook commands naar de server kan sturen.

UML



Communicatie met database ontwerp

Aan de hand van het gegeven ontwerp van Aaron heb ik een connectie gemaakt met de database middels WWWForm. Met een enkele Enumerator kan ik requests sturen naar de database die het vervolgens op een eigen php pagina ontvangt en verwerkt. Dit zorgt voor een flexibele werking en creëert ruimte om later nog functies eenvoudig toe te voegen. Dit alles wordt afgehandeld in ConnectToDatabase.cs.

Bekende bugs/problemen

Momenteel wanneer het spel afloopt kan de speler niets doen. Dit wil ik vervangen door de spelers terug te brengen naar een lobby waar ze opnieuw kunnen spelen.

Vanuit Unity wordt twee keer dezelfde score doorgestuurd naar de Database. Ik weet nog niet de oplossing om te zorgen dat de cliënts dit zelf aanroepen op cliënt niveau ipv op de server.

Wat had ik willen toevoegen?

Uiteraard had ik het graag beter vormgegeven. Ook wilde ik een timer bijhouden waardoor je ziet wie de snellere speler is, dit zorgt voor meer uitdaging en hopelijk ook meer spanning bij de spelers.