

Software Development Verslag

Table of Contents

Plan van Aanpak.....	3
Aanleiding.....	3
Opdrachtgever.....	3
Probleem.....	3
Doelstelling.....	4
Randvoorwaarden.....	4
Aanpak.....	4
Planning.....	5
UML Analyse.....	6
Klassen Diagram Simulator.....	6
Klassen Diagram Controller.....	7
Eindgebruikertest.....	8
Simulator.....	8
Test 1.....	8
Test 1.....	8
Test 2.....	8

Plan van Aanpak

Aanleiding

Voor de minor Software Development moet er in groepjes van twee een controller en een simulator worden gemaakt voor stoplichten bij het kruispunt (Stationsweg, Schrans, Zuiderplein en Oostergoweg). Maar elke controller moet met elk andere simulator kunnen werken en elke simulator moet met elk andere controller werken. Dit word allemaal besproken tijdens de les waar de opdrachtgevers bij aanwezig zijn.

Opdrachtgever

De opdrachtgevers van deze opdracht zijn Jos Foppele en David Schweizer. Het contact loopt voornamelijk via een Discord groep en Whatsapp groep. Discord en Whatsapp zijn software waarmee je kunt chatten. Ook kunnen ze worden benaderd via e-mail.

Jos Foppele

Contact: j.foppele@nhl.nl

David Schweizer

Contact: David.Schweizer@nhl.nl

Probleem

Hoe kan een controller een simulator besturen, hoe kan er worden verzorgd dat elke controller met elk simulator kunnen samenwerken en hoe moet het kruispunt eruit zien? Dit zijn vragen die tijdens dit project voor komen en moeten worden opgelost.

Doelstelling

Om in een groepje van twee een controller en een simulator te bouwen dat kan samen werken met andere gebouwde simulatoren en controllers. De simulator en controller moeten een bepaald kruispunt simuleren op het stationsweg (Zie afbeelding 1.)



Afbeelding 1.(google maps)

De simulatoren moeten de auto's, bussen, treinen, voetgangers en fietsers simuleren. Deze objecten moeten reageren op stoplichten en de stoplichten worden gestuurd door een controller.

Randvoorwaarden

De opdrachtgevers hebben al een locatie voor de discussies, test sessies geregeld. Verder kan ieder zelf bepalen hoe ze de simulator of controller gaan bouwen en welke tools hiervoor nodig zijn.

Aanpak

Zo veel mogelijk aanwezig zijn bij de geplande lessen, meedoen aan de discussies. Vervolgens werken aan een prototype en die uit gaan breiden en aanpassen aan de hand van de discussies die zijn gevoerd.

Planning

De lessen Software Development zijn standaard ingeplant. Tijdens deze lessen word er gesproken over hoe de simulator en controller moeten werken. Tijdens deze lessen word er feedback geleverd waarna de groep bij de volgende les iets doet aan de feedback.

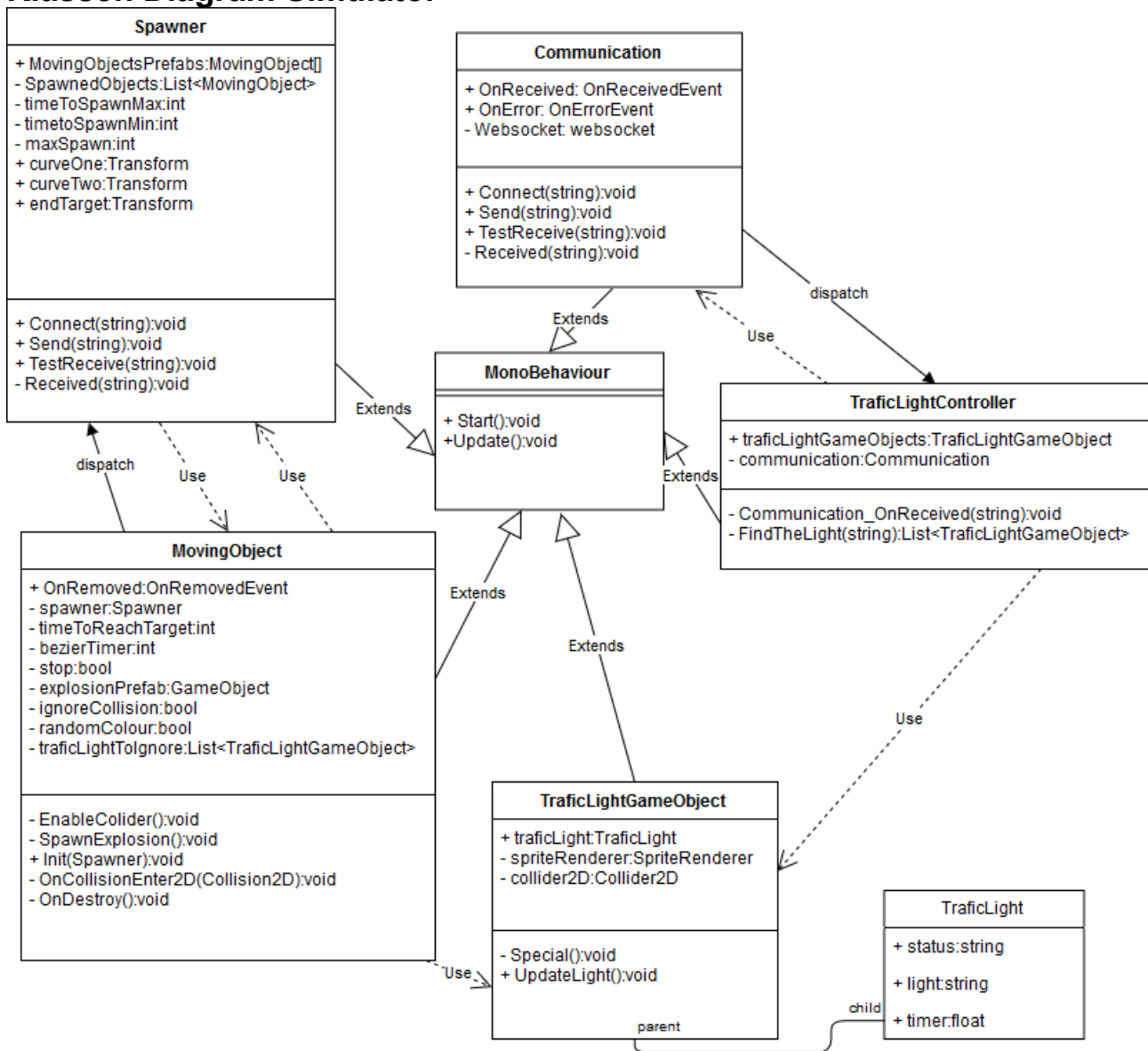
Hieronder een globale planning, achteraf is er nog informatie toegevoegd en aangepast.

Datum	Doel
04-09-2018	Als grote groep beslissingen maken
12-09-2018	
18-09-2018	
25-09-2018	
03-10-2018	Grote groep gesplitst in kleine groepen om bepaalde beslissingen te maken.
08-10-2018	
15-10-2018	
02-11-2018	Eerste geplande test sessie, connectie moet werken. Er kan data tussen de simulator en controller worden verstuurd. Meer structuur in het testen! Alles moet worden kunnen aangestuurd.
14-11-2018	
21-11-2018	
26-11-2018	
05-12-2018	DEAD LINE 1
11-12-2018	
19-12-2018	
08-01-2019	Informatie over deadline 2
16-01-2019	DEAD LINE 2

Kleur	Doel
	Discussiëren
	Discussiëren(Kleine groepjes)
	Testen
	Deadline

UML Analyse

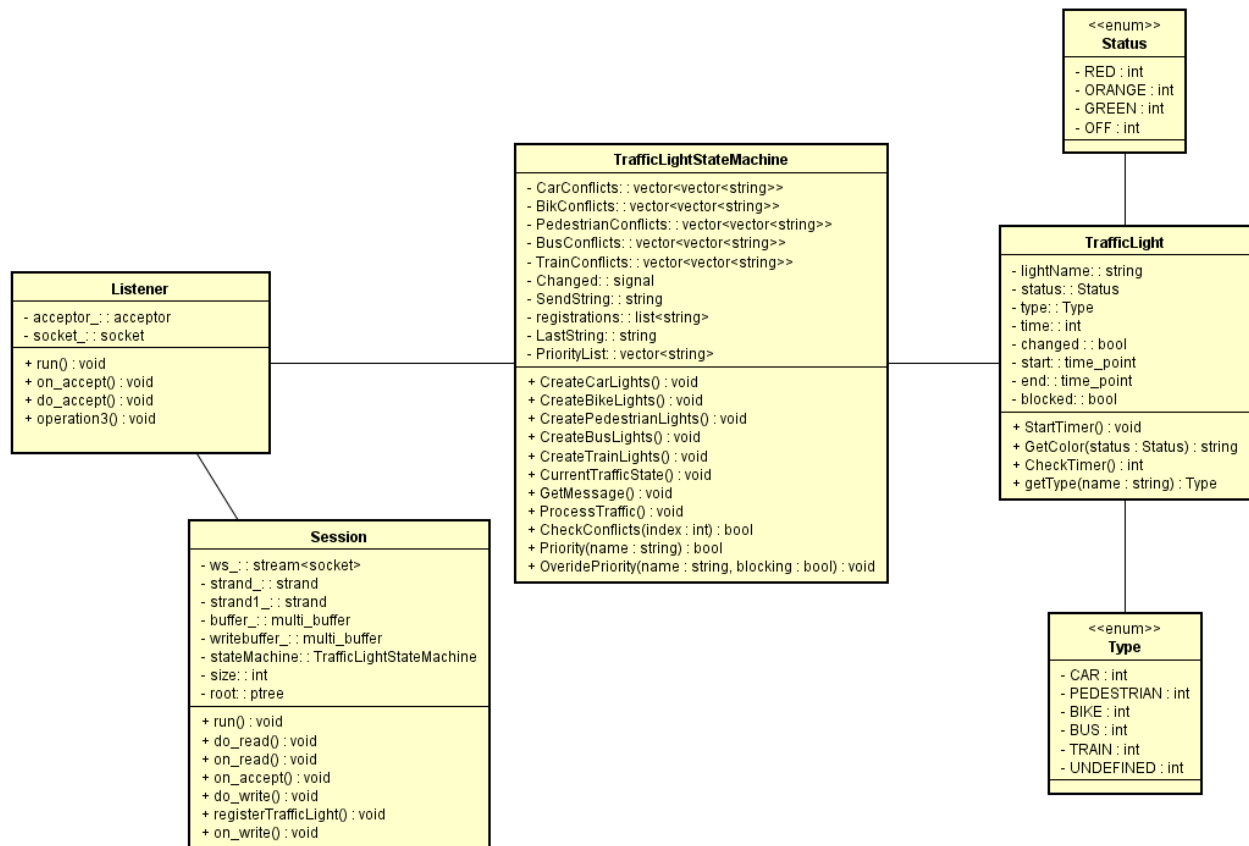
Klassen Diagram Simulator



Het programma is gemaakt in Unity, het programmeren in Unity is voornamelijk component based, elk MonoBehaviour object kan vast worden gemaakt aan een GameObject.

Tijdens de development van de simulator zijn er aanpassingen gedaan om bugs te fixen. Waardoor MovingObject bijvoorbeeld de Spawner nodig heeft.

Klassen Diagram Controller



De controller is gemaakt in C++, omdat er geen nut was voor een frontend van de controller is er gekozen voor een simpele ConsoleApplication. Tijdens het ontwikkelen van de applicatie is er meerdere keren een nieuwe keuze gemaakt met betrekking tot het prioriteren van de verschillende verkeersdeelnemers. Er is uiteindelijk gekozen voor een constructie die zorgt dat alle andere stoplichten geblokkeerd worden om weer op groen te gaan.

Eindgebruikertest

Simulator

Test 1

Tijdens het testen was er tevoren gekomen dat alle stoplichten goed worden bestuurd op 1 controller na. Bij een controller was er alleen een verbinding maar het versturen en ontvangen van data ging verkeerd. Verder zijn er wat ongelukken gebeurd wat leuke explosies heeft opgeleverd. Ook zijn er opmerkingen geweest over de “Zombie modus” In deze modus neemt de gebruiker controller over een zombie en kan vervolgens stoplichten veranderen van status(groen en rood).

De simulator was bij deze test als voldoende beschouwt.

Zie Bijlage(“Simulator Test 1 Results.xlsx”) voor alle resultaten tijdens de eerste test.

Test 2

Het resultaat is hetzelfde als bij test 1 hier en daar wat controllers die niet werkte en een paar ongelukken met explosies.

Controller

Test 1

Spijtig genoeg was bij de eerste test de controller niet aanwezig om te testen.

Test 2

Uit de test bleek dat de controller alle stoplichten aan kon sturen en over het algemeen ook een verbinding op kon zetten. Bij twee simulatoren was het niet mogelijk om de functionaliteit van de stoplichten te testen door het feit dat er een leeg bericht gestuurd werd vanuit de controller. De verbinding werd wel opgezet maar de simulatoren waren niet geprogrammeerd om met lege of corrupte berichten om te gaan. Daarnaast was er nog een simulator waarbij de verbinding helemaal niet opgezet kon worden met de foutmelding dat de verbinding niet open was. Na de verbinding met een websocket client op een iphone getest te hebben bleek dat de er niks mis was met de verbinding naar de controller.

Bij de test kwam wel naar voren dat er in sommige omstandigheden het mogelijk is dat er een paar stoplichten wat langer op rood staan dan gewenst. De reden voor deze onwenselijkheid is de poging het verkeer met een zo groot mogelijke efficiënte te laten rijden. Wanneer er een stoplicht is wat geen andere stoplichten belemmert dan zal deze meteen op groen gaan als deze de kans krijgt. Een mogelijke oplossing hiervoor zou kunnen zijn om een maximale “rood tijd” toe te voegen.