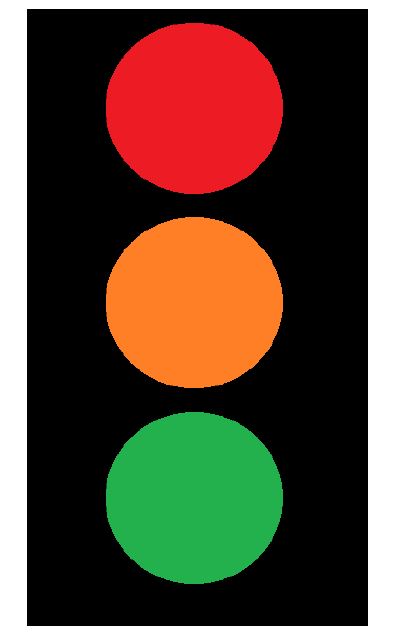
Software development



Teake Otter 09-06-2015

# Plan van aanpak

In de eerste 4 weken van het vak Software Development zijn er een aantal vergaderingen geweest waar besproken is wat de eisen en wensen zijn van het eind product, Er zijn groepen gevormd en we hebben de opzet van de software en het protocol met elkaar afgesteld. Met deze info heeft onze groep (Teake en Wim) een opzet gemaakt. Wim heeft de controller gemaakt en Teake de simulator.

Door ons goed aan het protocol te houden en het netwerk gedeelte snel af te hebben met functies om alle mogelijke pakketjes binnen het protocol te simuleren hebben we het netwerkgedeelte snel en robuust kunnen maken.

Om de software gelijk te houden met de andere groepen met betrekking tot de functionaliteit hebben we elke keer naar de testsessies toegewerkt. Zodat we niet in functionaliteit voorop liepen aangezien we deze dan toch niet zouden kunnen testen met de groep en alleen voor onvoorziene problemen zou lijden.

Dit betekend ook gelijk dat de planning ook gelijk liep met de rest van de software development groep.

# UML analysC:\Users\Teake\Desktop\Simulator.png

# Netwerk protocol

Versie: 20150303

IP protocol: TCP

Poort: 10000

Communcatie door het oversturen van raw bytes

**Controller functioneert als server**

**Simulator is de client**

**Soort:**

Voertuig(auto,bus,fiets,voetganger)

**Beschrijving:**

Deze data wordt vanuit de controller naar de simulator gestuurd om voertuigen te 'spawnen', dit voertuig krijgt een begin en een eind positie. De simulator kan hierop de route berekenen.

**Parameters:**

* Begin positie(Noord, oost, zuid, west en ventweg)
  + Noord: 0x00
  + Oost: 0x01
  + Zuid: 0x02
  + West: 0x03
  + Ventweg: 0x04
* Eind positie(Noord, oost, zuid, west en ventweg)
  + Noord: 0x00
  + Oost: 0x01
  + Zuid: 0x02
  + West: 0x03
  + Ventweg: 0x04
* Type(Auto, fiets, bus, voetganger)
  + Auto: 0x00
  + Fiets: 0x01
  + Bus: 0x02
  + Voetganger: 0x03

**Packet identifier:**

0x01

**Code voorbeeld:**

**C**

unsigned char vehiclePacket[] = {0x01, 0x03, 0x01, 0x02};

//identifier, Begin positie(West), Eind positie(Oost), Type( Bus)

**Soort:**

Stoplicht

**Beschrijving:**

Deze data wordt vanuit de controller naar de simulator gestuurd om stoplichten aan of uit te zetten.

**Parameters:**

* Stoplicht ID
  + 0x00 tot 0xFF (0 tot 255)
* State
  + Rood: 0x00
  + Oranje: 0x01
  + Groen: 0x02

**Packet identifier:**

0x02

**Code voorbeeld:**

**C**

unsigned char trafficLightPacket[] = {0x02, 0x2D, 0x00};

//identifier, Stoplicht ID(0x2D → id: 45), State(rood), nul

**Soort:**

Voertuig aanmelding

**Beschrijving:**

Deze data wordt vanuit de simulator naar de controller verstuurd om voertuigen aan te melden bij een stoplicht, hiermee wordt verteld aan de controller dat er een nieuw voertuig voor een stoplicht komt te staan of al staat.

**Parameters:**

* Stoplicht ID
  + 0x00 tot 0xFF (0 tot 255)
* Afstand
  + Onder stoplicht: 0x00 ← *als controller dit pakketje krijgt, verwijder voertuig uit stoplicht queue, dit pakketje wordt dus pas verstuurd als het voertuig er onder door rijdt(kan alleen als stoplicht op groen staat)*
  + Voor stoplicht: 0x01 (bepaalde afstand, ongeveer 50m).

**Packet identifier:**

0x03

**Code voorbeeld:**

**C**

unsigned char vehicleSignup[] = {0x03, 0x2D, 0x01, 0x00};

//identifier, Stoplicht ID(0x2D → id: 45), Afstand(staat voor stoplicht), nul

# TestPlan 12 mei

## Logica

Graag de volgende logica inbouwen voor de test sessie op 12 mei:

* Controller stuurt een aantal voertuig packets (0x01) met de volgende gegevens:
  + Voertuig 1:
    - Begin positie: Zuid
    - Eind positie: West
    - Type voertuig: Auto
  + Voertuig 2:
    - Begin positie: West
    - Eind positie: Zuid
    - Type voertuig: Auto
  + Voertuig 3:
    - Begin positie: Oost
    - Eind positie: Zuid
    - Type voertuig: Auto
  + Voertuig 4:
    - Begin positie: Zuid
    - Eind positie: Oost
    - Type voertuig: Auto
* De auto’s zullen nu gespawned worden in de simulator. \*Let op\*: Als het goed is komen dus aan alle 3 kanten van het linker kruispunt auto’s te staan en gaan ze allemaal “de bocht om” (geen enkele auto zal “rechtdoor” gaan op het kruispunt).
* De auto’s naderen na een x aantal seconden een stoplicht, simulator stuurt voertuig aanmelding (packet ID 0x03):
  + Stoplichten:
    - ID: 4 (0x04) (Richting Zuid -> West)
    - ID: 5 (0x05) (Richting Zuid -> Oost)
    - ID: 1 (0x01) (Richting Oost -> Zuid)
    - ID: 2 (0x02) (Richting West -> Zuid)
  + Voor stoplicht (0x01)
* De controller zal daarna één of meerdere stoplicht(en) op een bepaalde kleur zetten (de logica hierachter moet je zelf uit programmeren). Packet stoplicht naar simulator (packet ID 0x02):
  + Stoplicht ID: ID van het stoplicht
  + State: Groen (0x02), Oranje (0x01) of Rood (0x00)
* De simulator zal bevestigen dat de auto er onderdoor rijdt zodat de queue van de controller weer op 0 zal gaan (voor dat stoplicht). Packet voertuig aanmelding (ID 0x03):
  + Stoplicht ID: ID van het stoplicht
  + Onder stoplicht (dus afmelding): 0x00

# TestPlan 21 April

## Logica

Graag de volgende logica inbouwen voor de test sessie op 21 april:

* Controller stuurt voertuig packet (0x01) met de volgende gegevens:
  + Begin positie: Zuid
  + Eind positie: West
  + Type voertuig: Auto
* De auto zal nu gespawned worden in de simulator. \*Let op\*: Auto gaat naar links dus links voorsorteren (in hoeverre dat al mogelijk is moet je zelf bepalen, als hij maar in de juiste baan komt).
* De auto zal na een x aantal seconden het eerste stoplicht naderen, simulator stuurt voertuig aanmelding (packet ID 0x03):
  + Stoplicht ID: 4 (0x04)
  + Voor stoplicht (0x01)
* De controller zal daarna het stoplicht op groen doen (de logica hierachter moet je zelf uit programmeren. Het kan dus zijn dat je meerdere packets stuurt van stoplichten die dan ook groen worden). Packet stoplicht naar simulator (packet ID 0x02):
  + Stoplicht ID: 4 (0x04)
  + State: Groen (0x02)
* De simulator zal bevestigen dat de auto er onderdoor rijdt zodat de queue van de controller weer op 0 zal gaan (voor dat stoplicht). Packet voertuig aanmelding (ID 0x03):
  + Stoplicht ID: 4 (0x04)
  + Onder stoplicht (dus afmelding): 0x00
* Na een x aantal seconden zal dit stoplicht (en misschien anderen) ook weer op rood gaan (controller stuurt packet ID 0x02).
  + Stoplicht ID: 4 (0x04)
  + State: Rood (0x00)

Graag de bovenstaande logica implementeren voor 21 april met de bijbehorende verbose output (zodat het makkelijk is te controleren). Naast de bovenstaande logica testen ook een beginnetje maken met de simulator (zodat we iets kunnen laten zien).

# RotatieSchema

Onderstaand schema om de testen in te delen, aangezien we met 5 groepen zijn komt elke groep 1 ronde niet aan beurt. Als we ons aan onderstaand schema houden zijn we binnen een uur klaar.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tijd** | **Groep** | **Met** |  | **Groep** | **Met** |
| 11:40 - 11:50 | 1 | 2 |  | 3 | 4 |
| 11:50 - 12:00 | 5 | 1 |  | 3 | 2 |
| 12:00 - 12:10 | 1 | 4 |  | 2 | 5 |
| 12:10 - 12:20 | 1 | 3 |  | 4 | 5 |
| 12:20 - 12:30 | 2 | 4 |  | 3 | 5 |

## Groepen

Groep 1

* Bauke Jansen
* Hendrik Wiersma

Groep 2

* Jordi Epema
* Ane-Jouke Schat

Groep 3

* Sybren Sijbrandij
* Bert Labordus

Groep 4

* Dave van Klink
* Eddie Sprietsma
* Wouter Brinksma

Groep 5

* Teake Otter
* Wim Dalof

# Test resultaat

# De simulator heeft de testsessies met weinig problemen doorstaan. Alleen bij de laatste sessie treden er enkele onvoorziene foutmeldingen op, Deze hadden vooral te maken met pakketjes van de andere partij die onze simulator niet begreep. In de allerlaatste testsessie van 3 juni treden er bij de controller als de simulator geen problemen meer op.