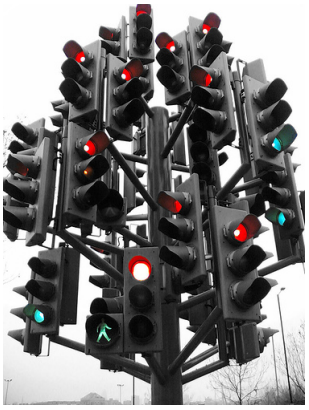
Gelijk oversteken
Een Roblox profielwerkstuk

1 Inleiding



Eerst naar links kijken, dan naar rechts kijken en dan voor de zekerheid nog een keer naar links kijken. Alleen als de weg dan nog vrij is, kun je veilig oversteken.

Als we dit bij elk kruispunt waar we met de auto, fiets of op de voet langskomen zouden moeten doen, dan zouden we de hele dag in de file staan. Om te helpen bij het oversteken hebben we daarom op veel kruispunten stoplichten gezet. Wil je weten hoe stoplichten veilig gemaakt worden, waarbij we ook nog zo kort mogelijk voor een rood licht staan? Kijk dan verder in deze opdracht.

2 Wat ga je doen?

Voor dit profielwerkstuk ga je kijken naar een computerprogramma dat een kruispunt met stoplichten simuleert. Met dit programma ga je onderzoeken wat de beste manier is om de stoplichten te gebruiken. In Hoofdstuk 3 kun je lezen wat simulatie is en waarom we het gebruiken. Ook zal het programma dat voor deze opdracht gebruikt wordt, kort worden uitgelegd. In Hoofdstuk 4 wordt vervolgens uitgelegd hoe je jouw onderzoek zou kunnen aanpakken. Tot slot staat in Hoofdstuk 5 uitgelegd hoe je een simpele simulatie kunt uitvoeren.

Als je iets in de opdracht niet snapt, kijk dan eerst in de woordenlijst aan het eind (Hoofdstuk 6). Hierin worden de belangrijkste begrippen uit de opdracht uitgelegd. Ook kun je op de websites kijken die in Hoofdstuk 7 staan genoemd. Als het dan nog steeds niet lukt, kun je ook een mailtje sturen naar de contactpersoon voor deze opdracht op de Universiteit Twente: Marieke Huisman (Marieke.Huisman@ewi.utwente.nl).

3 Simuleren

Zodra we een computerprogramma hebben gemaakt, willen we weten of het goed werkt. Normaal gesproken controleren we dit door het programma te testen; we starten het programma en kijken simpelweg of het doet wat we er van

verwachten. Maar als het programma ingewikkelder wordt, of wordt gebruikt in een ingewikkelde of gevaarlijke situatie, dan is simpel testen niet goed genoeg meer.

Een andere techniek die we kunnen gebruiken om te controleren of een computerprogramma goed werkt is simulatie. Bij simulatie wordt de omgeving waarin het computerprogramma gebruikt gaat worden nagebootst op de computer, en in die virtuele omgeving wordt getest of het programma doet wat het moet doen. Bij deze manier van controleren letten we dus niet zozeer op hoe het programma werkt, maar vooral op hoe de omgeving reageert op het programma.

Voor deze opdracht gaan we een aantal stoplichten op een kruispunt simuleren. De virtuele omgeving die we gaan gebruiken is een kruispunt met stoplichten. Het computerprogramma dat we willen testen is het programma dat bepaalt wanneer de lichten op rood of groen staan. We willen graag dat er geen ongelukken kunnen ontstaan bij het kruispunt, en ook dat er geen files ontstaan. Om dit uit te zoeken moeten we te weten komen hoe de omgeving op de stoplichten gaat reageren. Maar omdat we niet willen riskeren dat er ongelukken gebeuren, kunnen we de stoplichten niet zomaar ergens neer zetten en kijken of ze werken. Daarom is dit een typisch voorbeeld waarin we simulatie willen gebruiken.

3.1 Roblox

Om de simulatie simpel te houden hebben we alvast een virtuele omgeving gemaakt. Voor deze omgeving hebben we gebruikt gemaakt van Roblox, zie `http://www.roblox.com`. Dit is een programma dat bedoeld is om internet spelletjes mee te maken (en te spelen), maar dat ook gebruikt kan worden voor een simpele simulatie. Het voordeel van Roblox is dat je straks zelf makkelijk aanpassingen aan de omgeving kunt maken. In Figuur 3.1 kun je zien hoe de virtuele omgeving er uit ziet.

Om Roblox te installeren moet je het volgende doen:

1. Ga naar `http://www.roblox.com/Install/Download.aspx` en klik op download Roblox.

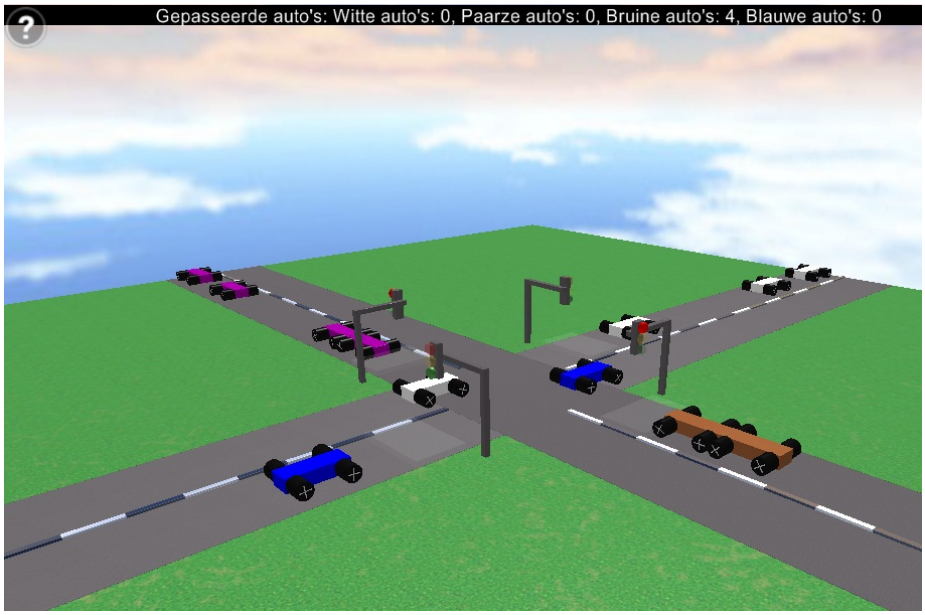
2. Als het bestand klaar is met downloaden, open het dan en volg de instructies om Roblox te installeren.

3. Om later programma's van de Roblox website te kunnen downloaden en te bewerken heb je een Roblox account nodig. Op `https://www.roblox.com/Login/Default.aspx` kun je zo'n account aanmaken (bereikbaar door in het hoofdmenu naar MyRoblox te gaan).

3.2 Hoe werkt het kruispunt?

Het kruispunt dat we gaan simuleren voor deze opdracht is al beschikbaar via Roblox. Je kunt dit kruispunt op de volgende manier downloaden:

1. Eerst moet je Roblox installeren en een account registreren. In Hoofdstuk 3.1 kun je vinden wat je hiervoor moet doen.



2. Open Roblox studio. Als je Roblox op de standaardlocatie hebt geïnstalleerd, dan kun je Roblox studio vinden in het Start menu bij: Programma's, Roblox en dan Roblox studio. Bij Roblox studio zie je bovenaan in het venster een aantal knoppen, terwijl de normale Roblox browser alleen een adresbalk heeft.
3. Log in met je Roblox account.
4. Ga naar de gebruikerspagina van de gebruiker 'superfmt': `http://www.roblox.com/User.aspx?ID=13432773` (of via het People menu zoeken op 'superfmt').
5. Rechts bij 'Active Places' staat het kruispunt, klik op de edit-knop. Let op: alleen als je bent ingelogd op Roblox, en met Roblox studio naar de website gaat zul je de edit-knop zien.
6. Het kruispunt is nu geopend in de editor, en je kunt het opslaan via het menu 'File' en dan 'Save As'.

Het kruispunt wat je net hebt gedownload bevat al een voorbeeldsimulatie. Je kunt deze simulatie op de volgende manier starten:

1. Start de simulatie door rechtsboven op de groene startknop te drukken (). Het kan even duren ($\pm$ 5 seconden) voordat de auto's beginnen met rijden.
2. Zodra je de simulatie hebt gestart, verschijnt er boven in beeld een zwarte balk waarin staat hoeveel auto's tot op dat moment de stoplichten hebben gepasseerd.
3. Je kunt de simulatie tijdelijk op pauze zetten met de gele pauzeknop ().

4. Als je klaar bent, kun je de simulatie resetten met de roze stopknop ().

4 Onderzoeksvraag

Voor deze opdracht is het de bedoeling dat je zelf verschillende schema's gaat bedenken voor de stoplichten. Met behulp van simulatie kun je daarna onderzoeken hoe goed de verschillende schema's werken.

Eerst zal je dus een aantal schema's moeten bedenken die je wilt gaan vergelijken. Je kunt daar het schema uit het voorbeeld voor gebruiken, maar bedenk tenminste één ander schema (bijvoorbeeld een schema waarbij alle rijbanen één voor één groen licht krijgen). Deze schema's moet je dan gaan programmeren in Roblox.

Zodra je de schema's hebt geprogrammeerd, moet je simulaties gaan gebruiken om te onderzoeken welke van de schema's het beste werkt. Daarvoor start je een simulatie en die laat je een paar minuten (bijvoorbeeld 5) lopen. Rechtsonderin, in de rand van het scherm, loopt de tijd mee (voorafgegaan door een 't'). Na afloop noteer je hoeveel auto's de stoplichten hebben gepasseerd, en of er ongelukken zijn gebeurd. Deze simulatie herhaal je voor elk schema dat je hebt bedacht.

Ook kun je het aantal auto's op de weg variëren. Door voor elk schema het aantal auto's op de weg steeds hoger te maken, kun je onderzoeken hoeveel auto's je schema maximaal aankan.

Als je al deze gegevens hebt verzameld, dan kun je ze samenvoegen in een tabel. In de tabel staat dan per schema en per drukte op de weg hoeveel auto's de stoplichten hebben gepasseerd. Aan de hand van deze tabel kun je dan bepalen welk schema in welke situatie het beste is.

Het aantal auto's dat een stoplicht kan passeren binnen een bepaalde tijd wordt wel de verwerkingscapaciteit (of *throughput*) genoemd. Het optimaliseren van de verwerkingscapaciteit is niet alleen bij kruispunten van belang, maar bijvoorbeeld ook bij communicatie via een computernetwerk.

4.1 Het kruispunt uitbreiden

Het kruispunt dat standaard in Roblox beschikbaar is, is een simpel kruispunt. De auto's rijden alleen maar rechtuit, en het gaat om een simpel éénbaanskruispunt. Als je een uitgebreidere versie van het kruispunt wilt simuleren kan dat ook, maar dan moet je zelf eerst de uitbreidingen maken.

Als je het kruispunt wilt gaan uitbreiden is het belangrijk dat je eerst begrijpt hoe het simpele kruispunt werkt. Hiervoor moet je de bestaande scripts bekijken en begrijpen wat ze doen (in Hoofdstuk 3.2 kun je lezen hoe je het kruispunt in de editor opent, zodat je de scripts kunt lezen). Om de scripts beter te kunnen begrijpen, is elke functie voorzien van commentaar dat beschrijft wat de functie doet.

Zodra je begrijpt hoe het kruispunt werkt, kun je beginnen met je eigen uitbreiding. Je zou bijvoorbeeld kunnen denken aan de volgende uitbreidingen:

- Zorg ervoor dat de auto's niet alleen maar rechtuit gaan, maar soms ook afslaan.
- Maak van het kruispunt een meerbaanskruispunt.

- Als je een meerbaanskruispunt hebt gemaakt, dan kun je die uitbreiden met voorsorteervakken.
- Voeg fietsers of voetgangers toe aan het kruispunt.
- Zorg ervoor dat nieuwe auto's na een willekeurige tijd verschijnen, in plaats van na een vaststaande constante tijd.

5 Voorbeeld

Als voorbeeld laten we zien hoe je een simpel schema maakt en kan instellen voor de simulatie.

Eerst gaan we een schema bedenken dat we willen testen. Voor dit voorbeeld staan de lichten 5 seconden op rood, 1 seconde op geel en 2 seconden op groen. Omdat de auto's niet afslaan, kunnen de auto's die in tegenovergestelde richting rijden tegelijkertijd groen licht hebben. De kleur in de naam van de stoplichten slaat op de kleur van de auto's. Dus `witLicht` is bijvoorbeeld de naam van het stoplicht voor de witte auto's.

Een stoplicht kan worden ingesteld met de functie

$$\_G.zetLichtManager(vertraging, roodtijd, geeltijd, groentijd, stoplicht).$$

De argumenten *roodtijd, geeltijd, groentijd* zijn de tijden (in seconden) die het stoplicht op rood, geel en groen moet staan. Bij het argument stoplicht, geef je het stoplicht mee waarvoor dit schema moet gelden. Een voorbeeldaanroep van de functie ziet er zo uit:

```
_G.zetLichtManager(0,5,1,2,witLicht)
```

Deze functie zorgt ervoor dat het stoplicht eerst *roodtijd* op rood staat, dan *groentijd* op groen en tot slot *geeltijd* op geel. Het eerste argument van de functie, *vertraging*, kunnen we gebruiken om de eerste roodtijd van het stoplicht te verlengen, waarmee we de schema's per stoplicht van elkaar kunnen laten verspringen. In ons schema hoort het stoplicht voor de witte en blauwe auto's direct op rood te springen, dus hier is geen extra vertraging nodig. Het licht voor de paarse en bruine auto's moet pas na 4 seconden op rood te springen (let op dat het licht aan het begin van het schema niet rood is omdat het dan op rood springt, maar omdat het dan nog steeds op de begintoestand rood staat). Daarom gebruiken we voor het stoplicht voor de paarse en bruine auto's een vertraging van 4 seconden.

De functies die we nodig hebben om dit schema in te stellen zien er dan zo uit.

```
_G.zetLichtManager(0,5,1,2,witLicht)
_G.zetLichtManager(0,5,1,2,blauwLicht)
_G.zetLichtManager(4,5,1,2,paarsLicht)
_G.zetLichtManager(4,5,1,2,bruinLicht)
```

Om dit schema in de simulatie te gebruiken moeten deze functies gekopieerd worden naar het goede script. Dit gaat als volgt:

1. Kopieer het bestand *VoorbeeldConfig* en geef het een nieuwe naam, bijvoorbeeld *config1*.

2. Zoek in de nieuwe configuratie naar functies die het stoplichtschema instellen, en vervang die door de bovenstaande functies.

3. Onderaan in het bestand zie je de regel *_G.voorbeeld_config = config* staan. Verander hier *voorbeeld_config* naar de naam van jouw configuratie, en laat de rest van de regel staan. Het ziet er dan bijvoorbeeld zo uit: *_G.config1 = config*.

4. Open tenslotte het script *Main* en vervang op de laatste regel *voorbeeld_config* door de naam van jouw configuratie, dus: *_G.config1()*

Je hebt nu je stoplichten ingesteld. Met de startknop kun je nu de simulatie starten en kijken hoe goed jouw configuratie werkt.

6 Woordenlijst

Configuratie Een configuratie is een verzameling instellingen. In deze opdracht bestaat de configuratie uit de instellingen van de stoplichten en de auto's.

Roblox Roblox is een programma waarmee op een makkelijke manier spelletjes kunnen worden gemaakt. Voor deze opdracht gebruiken we het om simulaties mee te doen.

Simulatie Bij een simulatie laten we een echte situatie nabootsen op een computer. Daarmee kunnen we dan met de computer onderzoeken wat er in het echt zou gebeuren in die situatie.

Virtuele omgeving De situatie die we met simulatie willen controleren, wordt ook wel de omgeving genoemd. De omgeving die we op de computer hebben nagemaakt heet dan de virtuele omgeving.

7 Handige links

Het kan zijn dat je tijdens het maken van deze opdracht ergens niet uit komt. In dat geval kun je een kijkje nemen op één van de onderstaande websites. Je kunt ook een mail sturen naar Marieke Huisman (Marieke.Huisman@ewi.utwente.nl) van de Universiteit Twente.

Roblox forum Het Roblox forum kun je vinden op `http://www.roblox.com/Forum`. Vooral in het Help Center zijn veel Roblox programmeurs actief. Is er dus iets waar je niet uit komt, dan kun je deze programmeurs om hulp vragen.

Roblox wiki De Roblox wiki kan je vinden op `http://wiki.roblox.com/index.php/Tutorials`. Op deze wiki zijn veel handleidingen en tutorials te vinden over hoe Roblox werkt. Dit is dus een goede plaats om te beginnen met zoeken als je iets wil veranderen aan de simulatie, maar niet goed weet hoe je dat moet doen.

Lua wiki De script taal die door Roblox wordt gebruikt heet Lua. Op `http://lua-users.org/wiki/TutorialDirectory` kun je een uitleg vinden van de meeste dingen die je met Lua kan doen.