



Rush Hour: pad-planning met een robot-arm

24 juni 2016

*Student:*Nils Hulzebosch
10749411Sjors Witteveen
10808493*Practicumgroep:*

F

Docent:

Arnoud Visser

1 Introductie

Voor het vak Zoeken, Sturen en Bewegen heeft elk groepje studenten een project gekozen in het kader van Robotica en zoeken, sturen en bewegen. In dit project is er onderzocht of en hoe het mogelijk is om een Rush Hour spel op te lossen met een robot arm.

1.1 Probleemschets

Rush Hour is een spel waarbij men voertuigen (auto's of vrachtwagens) over hun as (horizontaal of verticaal) moet verschuiven om de rode auto naar de uitgang te bewegen. Voertuigen mogen niet draaien, opgetild worden of tegen elkaar botsen.

Het probleem leent zich perfect voor een zoeken, sturen en bewegen project. Eerst moet met behulp van een zoekalgoritme een oplossing voor de puzzel worden gevonden. Deze oplossing is in de vorm van een pad (een sequentie van states) vanaf de begin state tot de goal state. De begin state is de gegeven (start) configuratie van voertuigen en de goal state een configuratie van voertuigen waarbij de rode auto voor de uitgang, de goal position staat.

Vervolgens moet dit pad worden omgezet naar een set van grijper-posities en daarna joint values van de robotarm. De grijper-posities bestaan uit x-, y- en z-coördinaten, de hoek van de grijper en de mate waarin hij open of gesloten is. Deze posities worden omgezet naar joint values, de hoeken van de scharnieren van de robotarm om zo'n positie te bereiken. Dit is het motion-planning gedeelte. De joint values moeten valide zijn, sommige standen zullen niet mogelijk zijn. Als laatste stap moeten de bewegingen daadwerkelijk worden uitgevoerd. Hierbij wordt getest of het systeem ook echt werkt in de praktijk.

1.2 Relevantie

Wat betreft maatschappelijke relevantie van dit project kan worden gekeken naar motion-planning. Dit wordt gebruikt bij het bewegen van robots in fabrieken (Fernau et al., 2003). Het onderzoeken hiervan kan tot nieuwe inzichten voor een sneller en effectiever productieproces leiden.

1.3 Vraagstelling

In dit onderzoek wordt de volgende vraag getracht te beantwoorden: welke stappen moeten worden doorlopen om een Rush Hour puzzel met een UMI RTX robotarm op te lossen? De verwachting is dat de implementatie van het speelbord erg bepalend is voor de mate van correcte uitvoering van de robotarm. Ook de precisie van de robotarm zelf zal invloed hebben op de uitvoering.

2 Methode

Deze paragraaf is opgedeeld in alle tussenstappen die tijdens dit project zijn gedaan om een Rush Hour puzzel op te kunnen lossen met de UMI RTX robotarm.

2.1 Verkrijgen van het pad

Voor het oplossen van een Rush Hour puzzel is er geprogrammeerd in Java. Er is gekozen voor het A* algoritme. Hierdoor wordt altijd de beste oplossing gevonden: de oplossing met het minst aantal stappen. Bovendien is er gebruikt gemaakt van een HashSet om dubbele states te voorkomen. Als er een state wordt gegeneerd die al in de HashSet zit, wordt deze niet aan de PriorityQueue met states (Grids) toegevoegd.

Het A* algoritme maakt gebruik van een score functie. Deze bestaat uit de som van de huidige kosten vanaf de begin state en de geschatte kosten naar de goal state. Hiervoor zijn drie admissible heuristieken gebruikt, namelijk de afstand tot de goal position, het aantal direct blokkerende voertuigen en het aantal indirect blokkerende voertuigen.

Het heuristiek 'afstand tot goal position' kijkt naar hoeveel stappen de rode auto moet doen om de goal position te bereiken. Hierbij wordt er niet naar obstakels tussen deze posities gekeken.

Het tweede heuristiek, 'het aantal direct blokkerende voertuigen', kijkt naar hoeveel blokkerende voertuigen er tussen de rode auto en de goal position staan. Voor elk voertuig wordt er een extra stap bij de schatting opgeteld, omdat dat voertuig eerst verschoven moet worden voordat de rode auto erlangs kan.

Ten slotte zorgt elke van de hierboven genoemde blokkerende voertuigen die aan beide kanten geblokkeerd zijn voor een extra stap in de schatting, omdat minimaal een indirect blokkerend voertuig verschoven moet worden om het direct blokkerende voertuig te kunnen verplaatsen.

Deze drie heuristieken vormen een (onder)schatting van het nog te bewandelen pad. Samen met de huidige kosten wordt de score bepaald. Aan de hand hiervan wordt een state (Grid) op een bepaald plek toegevoegd aan de PriorityQueue. Grids met een lage score komen vooraan en worden eerder geëxploreerd.

Uiteindelijk wordt het kortste pad gegeven vanaf de start state naar de goal state. Dit pad bestaat uit de sequentie van states. In elke state is één voertuig één plek verschoven.

2.2 Het pad vertalen naar moves

Omdat het pad uit een aantal states (Grids) bestaat, moeten deze states eerst worden vertaald naar moves (Move). De robotarm moet namelijk wel weten wat voor move er is gedaan, dit is niet te zien aan één enkele state. Om dit voor elkaar te krijgen zijn alle (vorige en volgende) states met elkaar vergeleken om hieruit de verandering van een auto te krijgen. Een Move bestaat dus uit twee punten, een punt (Point) met de oude x- en y-coördinaten en een punt met de nieuwe x- en y- coördinaten. Deze coördinaten bestaan uit het kolom- en rijnummer van het voertuig.

Voor de efficiëntie van het bewegen van de robotarm zijn meerdere verschuivingen van hetzelfde voertuig achter elkaar samengevormd tot één verschuiving van meerdere velden. Hierdoor neemt het aantal bewegingen die de robotarm moet uitvoeren af.

2.3 Moves vertalen naar posities van de grijper

Een positie van de grijper (GripperPosition) bestaat uit x-, y- en z-coördinaten (in millimeters), de hoek van de grijper (in graden) en de mate waarin de grijper open of dicht is (ook een hoek in graden).

Omdat de coördinaten van de punten bestaan uit kolom- en rijnummer, worden deze eerst omgezet naar Cartesische coördinaten in millimeters. Deze kunnen worden gebruikt om de positie van de grijper te bepalen.

Het verplaatsen van een auto van de oude naar de nieuwe positie kan worden opgedeeld in zes stappen. Bij alle stappen blijft de hoek van de grijper 0 graden.

1. De grijper beweegt naar de oude x- en y-coördinaten met een veilige hoogte. De z-coördinaat bestaat uit de hoogte van het bord plus de aangegeven veilige hoogte. De veilige hoogte is de hoogte waarop de grijper kan bewegen zonder een (toren van een) voertuig aan te raken. De grijper is open.

2. De grijper blijft op dezelfde x- en y-coördinaten, maar gaat van veilige hoogte naar grijp-hoogte. Dit is de hoogte waar hij een voertuig kan vastpakken aan zijn toren. De grijper blijft open.

3. Vervolgens blijft de positie van de grijper exact gelijk, maar wordt de grijper gesloten.

4. De grijper beweegt naar de nieuwe x- en y-coördinaten en houdt zijn hoogte gelijk (hoogte van het bord plus grijp-hoogte). De grijper blijft gesloten, waardoor het voertuig mee wordt genomen naar de nieuwe positie.

5. Eenmaal op de nieuwe positie aangekomen, en met dezelfde hoogte als de vorige stap, gaat de grijper open. Het voertuig staat nu op de juiste plek.

6. Vervolgens gaat de grijper op diezelfde positie weer naar de veilige hoogte (plus de hoogte van het bord) en blijft de grijper open.

Deze zes stappen worden voor alle zetten gedaan en in een ArrayList gezet. Als de oplossing van een puzzel bijvoorbeeld 30 zetten bedraagt, worden er dus 180 GripperPositions gemaakt.

2.4 Posities van de grijper vertalen naar joint values

De laatste stap is het vertalen van deze grijper-posities naar joint values. Dat zijn de waardes die de scharnieren van de robotarm moeten krijgen om de gewenste positie van de grijper te realiseren. De gebruikte robotarm heeft zes degrees of freedom, namelijk de hoogte van de arm (zed) en de hoeken van de shoulder, elbow, yaw, pitch, roll en gripper.

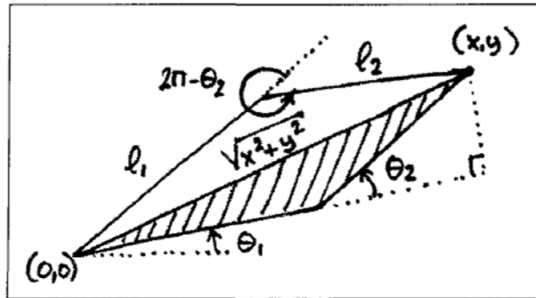
Als eerst worden de GripperPositions gecorrigeerd met de correctCartesian method uit de RobotJoints.class file. Vervolgens kunnen de roll en gripper angle direct overgenomen worden uit de GripperPositions. De zed wordt verkregen door het z coördinaat van de GripperPosition op te tellen bij de hoogte van de arm onderdelen van de UMI RTX robotarm. De shoulder en elbow angle worden met inverse kinematics berekend uit de x- en y-coördinaten van de GripperPosition (zie figuur 1).

Inverse kinematics

2D arm

$$\begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x = l_1 c_1 + l_2 c_{12}, \quad y = l_1 s_1 + l_2 s_{12}$$



$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2$$

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}$$

$$s_2 = \pm \sqrt{1 - c_2^2}$$

$$\theta_2 = \text{atan2}(s_2, c_2)$$

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(l_2 s_2, l_1 + l_2 c_2)$$

Figuur 1. Inverse kinematics 2D arm (Dorst, z.d.).

2.5 Implementatie van het spel

Er zijn verschillende spel-implementaties overwogen, maar uiteindelijk is er gekozen voor voertuigen van Lego die zich op een speelveld van lego bevinden. Doordat er voor Lego is gekozen konden de voertuigen zó worden gebouwd dat zij voor dit spel te gebruiken waren.

De voertuigen hebben allemaal een zo goed als gelijke breedte, lengte en hoogte. De vrachtwagens zijn anderhalf keer zo lang als de auto's. In het midden van elk voertuig is een toren gebouwd die door de grijper kan worden vastgepakt (op grijp-hoogte).

Er zijn zes auto's gebruikt. Deze staan op twee velden. De breedtes hiervan liggen tussen de 32 en 49 mm en de lengtes tussen de 82 en 94 mm. De hoogte is 70 mm. Hoewel de horizontale en verticale afmetingen een klein beetje kunnen verschillen, worden de auto's zo geplaatst dat hun toren op de y-as precies in het midden van een veld staat en op de x-as precies op de grens van twee velden. Hierdoor maken de verschillen in afmetingen niet uit.

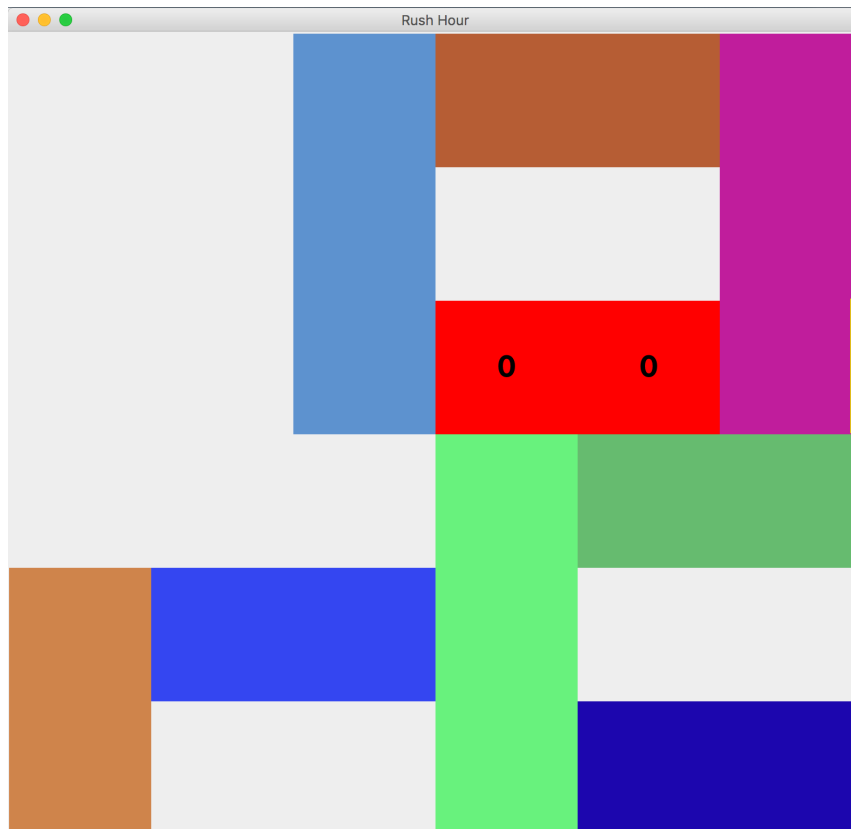
Daarnaast zijn er drie vrachtwagens gebruikt. Deze staan op drie velden. Ze hebben een breedte van 44 tot 49 mm, een lengte van 14,5 tot 15,9 mm en een hoogte van 70 mm. Ook hier geldt weer dat de toren in het midden staat. Door de lengte van de vrachtwagens verschuift de x-positie van de toren, deze komt nu in het midden van het middelste veld van de vrachtwagen te staan.

Het bord bestaat uit 6 bij 6 velden, die elk 64x64 mm zijn. De lengte van een zijde van het bord is dus 384 mm. Het bord is een veld van Lego en heeft een hoogte (dikte) van 8 mm.

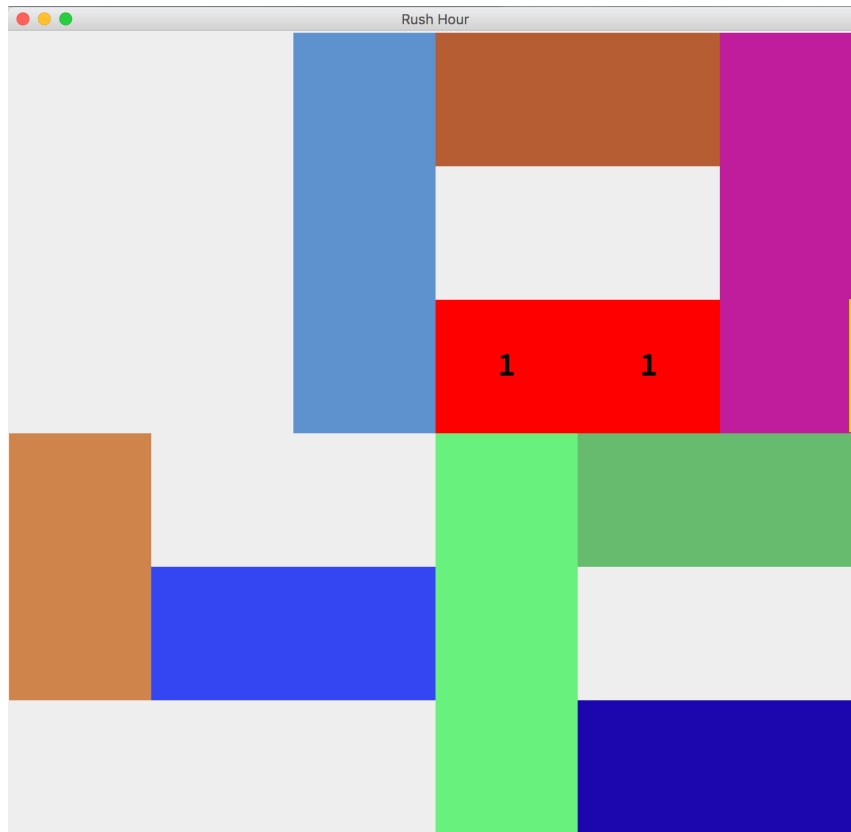
2.6 Keuze van de puzzel

Er zijn enorm veel verschillende puzzels te bedenken. Dit hangt af van de grootte van het bord, het aantal voertuigen en de lengtes, richtingen en posities van de voertuigen. Elke puzzel heeft een andere complexiteit.

Er is gekozen om met één puzzel te werken. Hierdoor blijven alle parameters gelijk (hetzelfde aantal voertuigen met dezelfde afmetingen en een even groot bord). Deze puzzel bestaat uit 6 bij 6 velden en bevat zes auto's en drie vrachtwagen. In de afbeelding hieronder is te zien hoe de startconfiguratie van deze puzzel eruit ziet. De afbeelding daaronder geeft de configuratie van de puzzel na het doen van één zet volgens A*. Deze verschuiving zal bij de Resultaten worden toegelicht. In beide afbeeldingen is de uitgang (rechts) weergegeven met een gele lijn.



Figuur 2. Startconfiguratie van de gebruikte puzzel.



Figuur 3. Configuratie van de puzzel na één verschuiving.

3 Resultaten

De resultaten zijn opgedeeld in twee categorieën: theorie en praktijk. In het theorie-gedeelte komen de oplossing van de puzzel en de waarden van de GripperPositions en joint values aan bod. In het praktijk-gedeelte worden deze waarden bestuurd in een simulatie, omdat de robotarm zelf een bug had. Hierdoor was het onmogelijk om de robotarm te testen.

3.1 Theorie

Het oplossen van de puzzel kost 83 zetten als elke verschuiving als één zet wordt gezien. Als meerdere verschuivingen van hetzelfde voertuig als één zet wordt gezien, kost het oplossen 50 zetten. Het oplossen duurt gemiddeld 95 ms. Het omzetten naar moves, GripperPositions en ten slotte joint values duurt gemiddeld 5 ms. Uitgaande van 50 zetten worden er 300 (50 maal 6) GripperPositions en evenveel joint values aangemaakt.

In de onderstaande tabellen zijn de (zes) Gripper-Positions en joint values van de eerste zet (zie afbeeldingen in de vorige paragraaf) van de puzzel te zien. De Gripper-Positions zijn nagegaan en kloppen met de werkelijkheid. De joint values komen overeen met het test-programma dat is aangeleverd voor Task 2 en 3 van dit vak.

	x-coord (mm)	y-coord (mm)	z-coord (mm)	roll angle (°)	gripper angle (°)
1	-14.0	408.0	98.0	0.0	30.0
2	-14.0	408.0	63.0	0.0	30.0
3	-14.0	408.0	63.0	0.0	0.0
4	-14.0	216.0	63.0	0.0	0.0
5	-14.0	216.0	63.0	0.0	30.0
6	-14.0	216.0	98.0	0.0	30.0

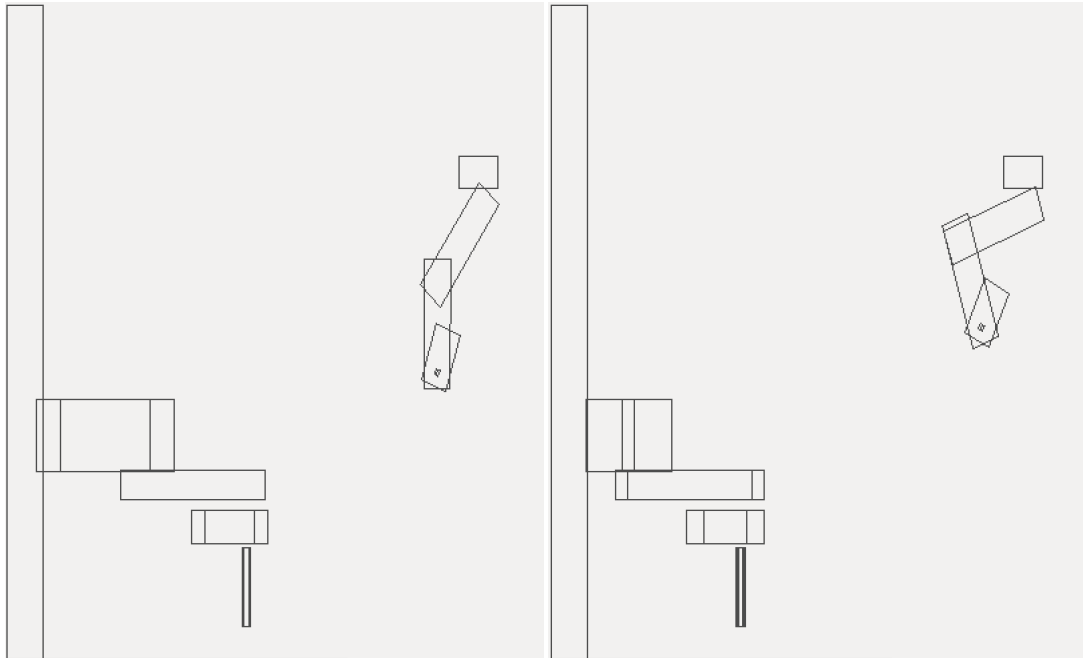
Tabel 1. Gripper-Positions van de eerste verschuiving.

	zed (mm)	shoulder (°)	elbow (°)	yaw (°)	pitch (°)	roll (°)	gripper (°)
1	305.0	-37.72517036931886	37.29803678776188	0.0	-90.0	0.0	30.0
2	270.0	-37.72517036931886	37.29803678776188	0.0	-90.0	0.0	30.0
3	270.0	-37.72517036931886	37.29803678776188	0.0	-90.0	0.0	0.0
4	270.0	-70.18013947601841	88.92980313820537	0.0	-90.0	0.0	0.0
5	270.0	-70.18013947601841	88.92980313820537	0.0	-90.0	0.0	30.0
6	305.0	-70.18013947601841	88.92980313820537	0.0	-90.0	0.0	30.0

Tabel 2. Joint values van de eerste verschuiving.

3.2 Praktijk

Omdat de robotarm niet getest kon worden, zijn de joint values bestudeerd in de simulatie van Task 3. Hierin zijn de voertuigen en het bord zelf niet te zien, maar het is duidelijk dat de robotarm de juiste bewegingen maakt om een auto vast te pakken en te verschuiven. De simulatie van de eerste verschuiving is te zien in de afbeelding hieronder. De auto die linksonder staat wordt één plek omhoog geschoven.



Figuur 4. Simulatie van de robotarm van stap 3 (links) naar stap 4 (rechts) tijdens de eerste verschuiving.

4 Discussie

De hoofdvraag was: welke stappen moeten worden doorlopen om een Rush Hour puzzel met een UMI RTX robotarm op te lossen? In dit onderzoek zijn de volgende stappen doorlopen om de joint values te verkrijgen.

Eerst is er met behulp van A* het kortste pad van de puzzel gezocht. Vervolgens is dit pad, bestaande uit states, omgezet naar moves. Deze moves bestaan uit oude en nieuwe coördinaten van het verschoven voertuig in de vorm van kolom- en rijnummer. Deze zijn omgezet naar Cartesische coördinaten, die gebruikt werden om de GripperPositions te bepalen. De GripperPositions zijn vervolgens omgezet naar joint values met behulp van inverse kinematics. Deze joint values kunnen direct door de robotarm worden gebruikt om de juiste posities aan te nemen, maar de mogelijkheid om dit te testen was helaas niet aanwezig, vanwege bugs in de software van de robotarm.

Het is gelukt om de puzzel op te lossen en de juiste joint values te verkrijgen. Helaas was het niet mogelijk om de robotarm in de praktijk te testen. Dit was erg teleurstellend, omdat er in de praktijk vaak onverwachte aspecten een rol gaan spelen. Een voorbeeld hiervan kan de precisie van de robotarm zijn. Dit is een aspect waar in puur theorie geen rekening mee gehouden kan worden. Dit zou in een vervolgproject kunnen worden getest. Dan kan ook worden getest of de Lego-implementatie een juiste keuze is geweest.

De stappen die moeten worden doorlopen om een Rush Hour puzzel met de UMI RTX robotarm op te lossen staan uitgebreid in de methode beschreven.

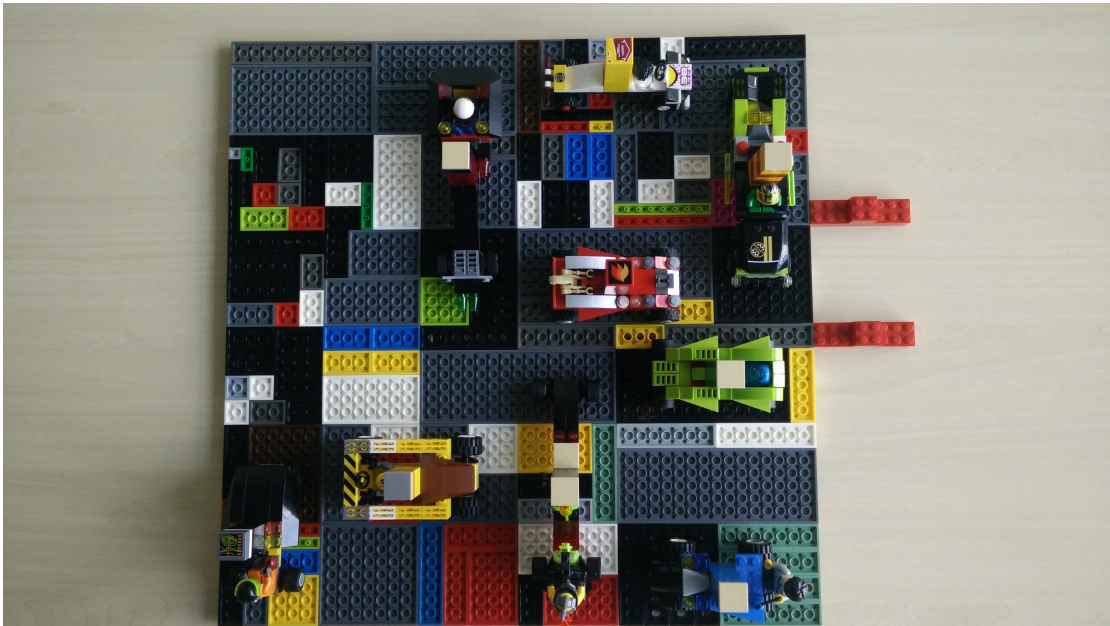
5 Referenties

Dorst, L. (z.d.). Path Planning [College-slides]. Geraadpleegd van https://blackboard.uva.nl/bbcswebdav/pid-5842885-dt-content-rid-6388954_1/courses/2315N001.5082ZOSB6Y.S2.1.2013/Path_Planning_Lecture.pdf

Dorst, L. (z.d.). Poses and Kinematics [College-slides]. Geraadpleegd van https://blackboard.uva.nl/bbcswebdav/pid-5842885-dt-content-rid-6388955_1/courses/2315N001.5082ZOSB6Y.S2.1.2013/Kinematics_Lecture.pdf

Fernau, H., Hagerup, T., Nishimura, N., Ragde, P. & Reinhardt, K. (2013). On the parameterized complexity of a generalized Rush Hour puzzle. CCCG 2003, Halifax, Nova Scotia, August 11–13, 2003.

6 Bijlage: Afbeeldingen van het speelveld



Figuur 5. Bovenaanzicht van het bord met de startconfiguratie.



Figuur 6. Zij-aanzicht van het bord met de startconfiguratie.