

Micromouse Werkboekje

Saad el Morabit

Micromouse Project Informatie

Supervisie: Bart Groeneveld
Instituut: Universiteit van Amsterdam
Contact: morabitsaad@gmail.com
Github: github.com/SaelMo/MicroMouse-UvA

Inhoudsopgave

1	Wat is een Micromouse?	2
2	Robotica 101	3
2.1	Opstellen	3
2.2	Benen	4
2.3	Nek	4
2.4	Oren	5
2.5	Ogen	6
2.6	Zelf Evaluatie en Bonus	6
3	Muizenbrein	7
3.1	Leren van Fouten	7
3.2	Roteer de Muis	8
3.3	De Eerste Algoritme Is Gratis	8
3.4	Zelf Evaluatie en Bonus	9
4	Bonus Opdracht: Tom & Jerry of Minotaurus & Theseus	10
5	Appendix: Sensoren	11
5.1	Infrarood Sensor	11
5.2	Ultrasoon Sensor	11

1 Wat is een Micromouse?

Een Micromouse is een robot die een doolhof aflegt vanaf een beginpunt om een stuk "kaas" te vinden (het eindpunt). In dit project is de muis een Arduino, een Italiaanse microcomputer, die wordt voortbewogen met wielen. De gebruiker stuurt code naar de muis via een USB-kabel en de muis wordt dan losgelaten om het doolhof op te lossen. Dit is anders dan een RC-auto besturen omdat de gebruiker geen controle heeft over de muis wanneer het begint te rijden. Of de muis het doolhof kan oplossen of niet is afhankelijk van de code die het krijgt aan het begin.

Dit project bevat twee Arduinos. Op de twee Arduinos kan apart worden gewerkt aan het project, maar in de bonus opdracht is het ook mogelijk om een predator-prooi spel op te stellen voor de twee muizen. Het project bevat ook een set blokken waarmee de gebruiker het doolhof kan bouwen. De website www.mazegenerator.net kan worden gebruikt om een willekeurige doolhof te genereren, maar bij voorkeur is het doolhof gebouwd door de gebruiker zelf of een uitdager.



Theseus (1952) uit "Information technology: A digital genius at play". Rechts: Claude Shannon, één van de oprichters van informatietheorie, de theorie die gaat over hoe informatie gecommuniceerd, geproduceerd en getransformeerd kan worden. Theseus is één van de eerste voorbeelden van lerende kunstmatige intelligentie, een onderwerp die tot vandaag de dag nog maatschappelijk relevant is.

De hoofdstukken bouwen op elkaar, wat je leert in hoofdstuk 2 heb je nodig in hoofdstuk 3 en et cetera. De taken worden ook abstracter, de eerste taken bestaan veel uit directe instructies volgen, maar latere taken bestaan meer uit abstracte ideeën *zelf* naar code te vertalen. Hoofdstuk 2 legt de absolute basis uit van de Arduino en hoe je de motoren/sensoren kan gebruiken. In hoofdstuk 3 worden de opdrachten wat breder en ga je leren hoe je de motoren en sensoren kan gebruiken om de muis vlot te laten rijden en reageren op hun omgeving. Hoofdstuk 3 eindigt met een volledige algoritme implementatie zonder dat het werkboekje de gebruiker een enkele lijn code geeft.

Indien er voldoende tijd is, is er nog het kat & muis spel van hoofdstuk 4. Het spel eist van twee (groepen) gebruikers om de muizen niet alleen te programmeren voor een vastgestelde taak, maar ook om hun tegenstander te slim af te zijn. Hoofdstuk 5 legt de sensoren van de muis uit voor gebruikers die de onderdelen niet eerder hebben gezien, of een opfrisser kunnen gebruiken. Nieuwsgierigheid is aangemoedigd, zeker later in het werkboek. Als jij je eigen oplossingen voor de taken kan bedenken, of een eigen idee hebt voor een taak voor jouzelf, ga ervoor!

2 Robotica 101

2.1 Opstellen

Het doel van dit werkboekje is dat je niet alleen de muis kan besturen, maar dat je een algoritme kan bouwen waardoor je een hele familie van verschillende doelhoven kan oplossen. Dit is natuurlijk een onrealistische taak voor een eerstbeginnende. Maar dit kunnen wij nog steeds doen via één van de krachtigste ontdekkingen in de informatica: moeilijke problemen zijn een groep makkelijke problemen in een vermomming! Anders gezegd, het werkboekje snijdt de moeilijkere *taak* in kleinere *taken*. Er kan veel misgaan in de taken oplossen, maar onverwachte problemen oplossen is een vaardigheid geleerd met ervaring. Er bestaat geen "uniek objectief" correcte antwoord in de *meeste* gevallen, wees dus niet ontmoedigd door de drempels. Het is een kans om creatief te zijn in *jouw* oplossingen!

Taak 0: Communicatie Tussen de Muis en de Computer

1. Installeer de Arduino IDE waarmee jouw computer kan communiceren met de Arduino. Het liefst haal je het online op www.arduino.cc/en/software, maar het is ook op de USB-stick
2. Verbindt de Arduino met de computer via de USB-kabel. Als de Arduino een batterij mist, gaat het nu opstarten. Dit is te zien via de gekleurde lichtjes die nu gloeien op de Micromouse

Je hebt nu de nodige software (IDE of Integrated Development Environment) en hardware (Arduino), het is handig om een voorbeeld code te openen om te testen of de Arduino deugt of niet.

Taak 1: Arduino IDE Navigatie en Voorbeelden

1. Open het IDE-programma die je hebt geïnstalleerd
2. Onder File, en dan onder Examples, en dan onder 01:Basics kies Blink
3. Onder tools (bovenaan de IDE-scherm), en dan onder ports kies de eerste COM. Dit geeft de informatie poort aan, volgende opgave zie je of dit de juiste was
4. Onder tools, en dan onder board, kies Arduino Uno. Dit vertelt de IDE welke Arduino jij gebruikt
5. Onder tools, en dan onder programmer, kies Arduino ISP. Dit vertelt IDE dat je je Arduino gaat gebruiken voor "In System Programming"

De Arduino is nu ingesteld en een stuk voorbeeld code is geopend.

Taak 2: Arduino Lampje Test

1. Kijk naar het rode lichtje op de muis. Het hoort constant aan te staan (*niet* knipperen)
2. Onder het balkje met File, Edit, Sketch, et cetera zie je twee cirkelvormige knoppen. Klik op het linkere knopje (verifiëren) om te zien of jouw code erkend wordt door de IDE
3. Klik op het knopje rechts van de verifieer knop. Het lichtje moet nu gaan knipperen
4. Maak een nieuwe, lege file (knop onder file in IDE). Dit geeft je een magere lap tekst met alleen void loop/setup() code. Negeer dit voor nu, en implementeer de magere code. Het lampje moet nu stoppen met knipperen omdat je de code *in* de Arduino hebt gewijzigd

Jij weet nu hoe je een code moet implementeren op de Arduino, en hoe je met een schone lei kan beginnen (lege file uploaden). Nu moeten wij de code alleen veranderen van "knipperen" naar "doolhof oplossen" en het hele project is klaar! Tijd voor de componenten.

2.2 Benen

Ten eerste willen wij de motoren activeren, hiervoor moeten wij eerst een pakket downloaden. Het pakket zorgt ervoor dat de Arduino kan communiceren met de motoren.^[1]

Taak 3: Pakketten Installatie

1. Onder Sketch, en dan onder include library, en dan onder manage libraries, zoek naar Motor Driver Library door Rupin versie 1.0.0
2. Installeer het pakket en open de examples weer, ga onder Motor Driver Library en open basic-Driver. Implementeer de code om te zien wat er gebeurt

Wij gaan bouwen boven op dit voorbeeld. Het kan zo zijn dat alleen 1 van de motoren aan staat. Het is tijd dat wij code gaan bestuderen: de code `m.motor(x, y, z)` heeft maar 3 waarden x, y, z die wij kunnen veranderen. Wij kunnen erachter komen wat ze doen door met de waarden te spelen.^[2]

Taak 4: DC-motoren

1. FORWARD lijkt het meest duidelijk, wat als we het veranderen naar BACKWARD?
2. Het getal 255 ziet er raar uit, wat gebeurt er als wij het veranderen naar een ander getal?
3. Als je de twee vorige opgaven hebt gedaan, dan weet je dat alleen de eerste waarde het probleem kan veroorzaken (deductie!). Kan jij het probleem nu oplossen? Hint: bekijk de Arduino board

2.3 Nek

Misschien heb je de servo motor al gezien, de ultrasoon sensor zit op een standje die kan draaien! De servo eist maar 1 waarde in plaats van 3, tijd om te kijken wat het doet.

Taak 5: Servo Motor

1. Onder `#include<MotorDriver.h>` type `#include<Servo.h>`. Het is een ingebouwd pakket, installatie is niet nodig omdat het al in de IDE zit
2. Onder `MotorDriver m;` type `Servo servo1;`
3. Tussen de haakjes van `void setup()`, type `servo1.attach(9);` (of 10)
4. Tussen loop type `servo1.write(30);` om de servo te roteren op dezelfde manier dat je de wielen hebt laten draaien. Speel met de waarde van 30 om een gevoel voor de servo te krijgen

Als je moeite hebt met errors raad ik aan om te kijken naar werkende code (internet/examples), rond te spelen met werkende code om te zien "wanneer het breekt" of iemand met ervaring te vragen. Voordat je afspraken kan maken met jouw Micromouse, moet je zijn taal leren.

^[1]Het pakket is ook op de USB, zet het in je documenten folder specifiek C: \Documents\Arduino\libraries. Jouw document folder kan ergens anders liggen gebaseerd op welke bestuursysteem (Windows, macOS, Linux) je gebruikt

^[2]Als je wilt weten wat een pakket doet in detail, open het met een tekstverwerker zoals Notepad en lees de code voor jezelf. Voor nu zijn de pakketten te geavanceerd en gebruiken wij de pakketten als een "blackbox". Een blackbox is een systeem waar je iets in kan doen, en iets terug kan krijgen, maar waarbij je niet echt weet wat het intern doet

2.4 Oren

Wij willen nu de ultrasoon sensor gebruiken, dit is moeilijker dan de vorige taken gezien wij niet alleen data moeten sturen naar de Arduino, maar ook ontvangen.^[3] Er is een "trigger" die een ultrasonisch golf creëert, de golf wordt terug gereflecteerd naar de Arduino als een "echo", de tijd tussen de trigger en de echo wordt gebruikt om de afstand te berekenen.

Taak 6: Ultrasonische Sensoren

1. Installeer het pakket NewPing op dezelfde manier dat je MotorDriver.h hebt geïnstalleerd. Type bovenaan `#include<NewPing.h>` om het pakket te importeren voor jouw gebruik
2. Wij willen nu de Arduino vertellen welke poorten wij willen gebruiken. Type `#define ECHO_PIN A4`, `#define TRIG_PIN A5` en `#define MAX_DISTANCE x` net onder de `#include` code. Kies zelf een getal voor x

3. Wij willen nu de sonar als object zien in het programma, type onder de `servo1` lijn

```
NewPing sonar (TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
```

4. Schrijf

```
float Dist = 0;
int iterations = 5;
float echoTime = 0;
```

onder NewPing, maar boven `void setup()`. Dit zijn "initiële waarden". Float betekent dat het een decimaal getal (0.13, 4/13, 3.0) is. Int betekent integer oftewel een geheel getal (0, -1, 3)

5. Tussen de void loop gaan wij nu de echotijd aanvragen van de sensor, en ermee de afstand vinden. Voor de echotijd gebruiken wij (vraagje: wat is de mediaan?)

```
echoTime = sonar.ping_median(iterations);
```

Om de afstand te berekenen gebruik

```
Dist = sonar.convert_cm(echoTime);
```

6. De sensor zit constant te meten, om ervoor te zorgen dat wij alleen informatie krijgen in vaste tijd stappen vertellen we de Arduino om voor x hoeveelheid tijd een pauze te nemen. Type `delay(x)`; tussen de void loop, speel met de delay waarde

Er is een probleem, wij kunnen afstanden meten en gebruiken, maar niet zelf aflezen/visualiseren. Je vraagt je misschien af "Boeien, wij hebben de data en kunnen ermee werken toch?" Dit is correct, de data komt vóór de visualisatie, maar het is ook belangrijk om data voor jezelf te visualiseren om mogelijke problemen op te sporen/diagnosticeren. Je kan "correcte code" schrijven die de IDE en de Arduino accepteren, maar het kan nog steeds incorrect zijn als het niet doet wat je wilt dat het doet. Je kan een grammaticaal correcte zin bouwen die niet logisch is bijvoorbeeld: "Ik ben aan het liegen." Laten wij dus de data gaan "printen", een vakterm voor "laat data op het scherm zien".

^[3]De pin labels (A4, A5, et cetera) kunnen anders zijn dan in de opgave zijn gegeven. Indien dit het geval is, volg de Arduino kabels. Bijvoorbeeld, als je geen info krijgt van de rechter infrarood sensor, kijk naar de A/D label op de sensors en kijk naar waar de kabel gaat van sensor naar de Arduino. Zie sectie 2.5 voor een korte uitleg

Taak 7: Data Weergeven

1. Tussen `setup()` type `Serial.begin(9600);` . Het beginstukje (`Serial`) is een manier om aan de Arduino te vertellen dat je data wilt importeren. Feitje: de S in USB staat voor Serial
2. Onder `update()` type `Serial.println(Dist);` . Dit print de numerieke waarde van `Dist`
3. Implementeer de code en ga onder Tools, dan onder Serial Monitor krijg je een tekst doos die je laat zien wat de waarde is van `Dist`. Check of alles goed werkt door fysiek met de sonar te spelen (hand of blokken voor de sensor)

2.5 Ogen

De infrarode (IR) sensor is vergelijkbaar in werking met de ultrasone sensor. Er is wel een praktisch verschil in de Micromouse: er zijn twee kabels voor IR data. Wij hebben VCC voor voeding, GND (Ground) voor de aarde en een analoge pin voor alle sensoren. De IR sensoren hebben nog een digitale binding. Besluit voor jezelf wanneer je liever met analoge of digitale informatie wilt werken.

Taak 8: Infrarood sensoren

1. Je hebt de belangrijkste aspecten van sensoren al geleerd in de vorige opgave. Om je op weg te helpen met initialisatie, hier is voorbeeld code

```
const int pinIRd1 = A1;
const int pinIRa1 = A0;
int IRvalueA1 = 0;
int IRvalueD1 = 0;
```

Voor 1 sensor initialiseer je nu digitale en analoge data. Gebruik dit om voor beide sensoren de digitale *en* analoge data te initialiseren. Met de voorbeeld heb je dan 8 lijnen aan code

2. Dit keer gaan wij niet leunen op een bestaand pakket, laten wij dus meteen data uit de pins krijgen. Om dit te doen probeer voor alle pins dit lijntje code te doen tussen `setup`

```
pinMode(pinIRd1, INPUT);
```

3. Tussen void loop lezen wij alle waarden af via de code voorbeeld

```
IRvalueA1 = analogRead(pinIRa1);
IRvalueD1 = digitalRead(pinIRd1);
```

Gebruik dit om voor alle 4 pins (2 digitale en 2 analoge) de data af te lezen

4. Rond de opdracht af door alle sensor data te printen. Combineer dit met wat je hebt gedaan in de vorige opgave om te verzorgen dat je makkelijk Serial Monitor kan gebruiken om alle sensor data af te lezen. Dit kan waardevol zijn in toekomstige taken voor jouzelf

2.6 Zelf Evaluatie en Bonus

Jij kan nu de Arduino aandrijven, data weergeven en de sensoren gebruiken. Hier zijn wat vragen die jij aan jezelf kan stellen om erachter te komen of je alles hebt geleerd wat je nodig zal hebben in de volgende hoofdstukken.

1. Ik kan een willekeurige example file vinden in de IDE, het implementeren op de Arduino en het dan "de-implementeren" met een lege file
2. Ik kan de servo motor laten draaien naar links of rechts

- Ik kan niet alleen data aflezen van de ultrasone sensor, maar via de blokken kan ik ook een gevoel krijgen of een object 5 centimeter van de muis is of 10 centimeter
- Ik heb een gevoel voor hoe de hoogte van de muren een invloed hebben op de sensoren
- Ik kan de motoren met de klok draaien, tegen de klok laten draaien en helemaal stoppen

Voor de bonus opdrachten, kies de opdracht die jou leuk lijkt om te doen

- Bouw een paar tetrominos (Tetris vormpjes) met de blokken. Gebruik beide infrarood sensoren als een "tetromino detector", waarbij jij de waarde kan aflezen via de IDE en alleen met die waarden weet welke tetromino het is. Bonuspunten als iemand een tetromino voor jou bouwt, het meet en jij dan met de data weet welke tetromino het is zonder de tetromino zelf te zien
- Bouw een muur voor de muis. Rij langzaam naar de muur en gebruik de sensor waarden en delay om de snelheid te berekenen van de Micromouse ($v = \frac{\Delta Afstand}{\Delta Tijd}$). Vergelijk de snelheid met de snelheid die je berekend door de muis te timen (bijv. met een telefoon) terwijl het een vaste en bekende afstand aflegt, bijvoorbeeld een 30 cm liniaal of 1.70 m persoon die ligt
- Door twee kolommen te bouwen bereken de rotatiesnelheid $\omega = \frac{\Delta Hoek}{\Delta Tijd}$ waarmee de servo draait
- Het is passé, maar probeer eens "Hello World!" uit te printen

3 Muizenbrein

You did it! Je hebt een robot aan de praat gekregen. Dit is in zichzelf een prestatie, maar nu willen wij nog verder. De onderdelen werken, wij weten hou wij de motoren moeten activeren en we weten hoe wij data kunnen krijgen uit de sensoren, maar hoe gebruiken wij al dat data? Om een stap verder te komen is dit hoofdstuk gesplitst in drie doelen: verzorg dat de muis niet de muur raakt, roteer de muis en implementeer een algoritme van mensen taal naar programmeertaal.

3.1 Leren van Fouten

Om keuzen te maken met de sensor data moeten wij abstracte ideeën overbrengen naar code voor de IDE. Om de muis te stoppen willen wij tegen het programma zeggen: "Hey, wanneer de sensoren zeggen dat er iets net voor de muis is trap de rem!" Om dit te doen hebben wij if, else if, en else statements nodig. Ik geef een voorbeeld die het probleem "technisch oplost", maar verbetering eist.

Taak 9: Remmen

- Onder update() type (met de waarde voor x, y, v, w ingevuld)

```
if(z){
    m.motor(x, v, 0);
    m.motor(y, w, 0);}
else{
    m.motor(x, v, MotorRightSpeed);
    m.motor(y, w, MotorLeftSpeed);}
```

en verwijder de oude m.motor() code buiten de if statement

- Bonus vraag: v, w is normaal FORWARD, BACKWARD maar het accepteert ook getallen, kijk naar hoe de motoren reageren op verschillende getallen
- In de vorige opdracht heb je gezegd "als z waar is, niet rijden, anders doorrijden". Wij willen dat de z waarde een conditional is. Probeer bijvoorbeeld $Dist < 5$ in te typen voor z, kijk wat er gebeurt en of een ander getal fijner is
- Test jouw code uit, de taak is voltooid wanneer je code de muis remt voordat het een muur raakt. Verbeter de code met and/or statements (Google het) om infrarood sensoren ook te gebruiken. Des te beter je muis is met remmen voor obstakels, des te beter

Dit taak is een stuk moeilijker als dit de eerste keer is dat je werkt met if/elif/else statements, als dit het geval is voor jou raad ik www.arduinogetstarted.com aan. Uitleg is kort en bondig plus er zijn veel voorbeelden die het duidelijk maken wat er aan de hand is.

3.2 Roteer de Muis

Om de muis te laten draaien willen wij één van de wielen de andere kant laten roteren. Het doel is om de muis een vaste hoek te laten draaien. Je kan de Arduino timer library gebruiken, als je wilt.

Taak 10: Rotatie

1. Probeer de muis te laten rijden met één wiel die de andere kant draait dan het andere. Het volgt zijn eigen staart als een opgewonden bulterriër
2. Gebruik de delay code om ervoor te zorgen dat het alleen roteert voor een vaste tijd. Probeer erachter te komen hoe lang de delay moet zijn om te verzorgen dat de Arduino een kwartslag maakt, een halve draai, en een volle draai. Onthou de delay waarden voor toekomstige taken
3. Onder de if statement van de vorige opdracht, probeer de code die je hebt bedacht voor het roteren van de muis erin te zetten. Verzorg ervoor dat wanneer de muis een muur ziet dat het draait naar een richting zonder muur en dan doorrijdt
4. Laat de Arduino los in een doolhof (of kamer) en krijg een gevoel voor hoe het reageert op verschillende obstakels. Als het vast komt in een doos, kan het ontsnappen? Kan het reageren op muren van verschillende hoogten? Hoe weet de Arduino de beste hoek om te draaien wanneer het een muur raakt?

3.3 De Eerste Algoritme Is Gratis

Wij zijn ver gekomen, jij hebt de muis aan de praat gekregen en ervoor zorgt dat het niet alleen kan rijden maar ook kan reageren op obstakels. Nu is het tijd voor de volgende stap, waar wij van simpelweg programmeren/robotica overstappen naar algoritmen. Wij gebruiken een simpele algoritme genaamd Wall Follower^[4] en het doel is dat van de beschrijving van het algoritme en alles dat je al hebt gedaan je een code onafhankelijk kan schrijven. Dit is de piek van het project, informatici doen niet alleen aan code in systemen zetten, maar ook aan "ideeën" veranderen in algoritmen en dan code. Dit is niet alleen een uitdaging in de correcte code op te schrijven, maar het vraagt aan jou of je een abstracte idee in jouw hoofd kan veranderen naar computertaal? Veel geluk!

^[4]Buiten doolhoven wordt dit algoritme weinig toegepast. Maar er is een uitzondering: natuurkundigen hebben een materiaal haar weerstand berekend door te "wall followen" langs geleidende/isolerende nano-stokjes. Er zijn nog algoritmen zoals flood fill die doolhoven oplossen, maar ook de basis vormen voor de "bucket tool" in kunstprogramma's

Taak 11: De Topology is Naar de Muizen

1. Bouw een doolhof die beschreven kan worden als een "simply connected space". Dit is de standaard doolhof die je al ruwweg kent, maar met de restrictie dat er geen losse blokken zijn. Alle blokken hebben een connectie met de buitenste muur
2. Als je ooit zelf een echte doolhof hebt opgelost heb je misschien het gevoel gekregen dat als je aan de muur blijft kleven je het ooit zal oplossen, al kan het lang duren. Wij kunnen dit "gut feeling" veranderen naar code, maar bedenk een voorbeeld van wanneer dit duidelijk niet werkt en wat simply connected space doolhoven hiermee te maken hebben
3. Voordat dit in de IDE komt, laten we het op pen en papier oplossen. Maak een ruwe schets van het doolhof en probeer het op te lossen door één zijde te volgen. Als je een punt vindt waar je meerdere richtingen op kan gaan, neem de richting waardoor je niet jouw muur verliest. Dit kunnen wij doen omdat in twee dimensies muren alleen maar twee zijden hebben
4. Argumenteer voor jezelf waarom de video <https://youtu.be/IIBwiGrUgzc> (titel: Maze Transformed door gebruiker Maze Master) hier relevant is, en wat de connectie is met de eerste stap
5. Genoeg voorbereiding, type de code die de "muur kleeft" techniek nabootst. In principe kan je dit oplossen met alles wat je in dit hoofdstuk hebt geleerd, maar als je een betere methode kan bedenken voor jezelf ga ervoor!

3.4 Zelf Evaluatie en Bonus

1. Ik weet hoe ik de muis naar voren kan laten gaan, naar achter, een kwartslag links of recht en hoe ik het kan stoppen
2. Mijn Micromouse stopt wanneer het een muur ziet, draait weg en blijft door rijden
3. Het Wall Following algoritme is geïmplementeerd en mijn Micromouse kan alle simply connected doolhoven nu oplossen. Bonuspunten als een andere persoon een toets doolhof heeft gebouwd

Bonus, nu dat je algoritmen onder de knie hebt kun je verschillende taken geven aan jouw muis

- Carry bot: leg een kleine houtblokje op de robot en probeer ervoor te zorgen dat het een rechte lijn kan rijden zonder het blokje te laten vallen door met de snelheid te werken. Hint: de wetten van Newton zijn verbazend handig hier!
Dit wordt uitdagender wanneer je de robot ook laat draaien met het blokje, en wanneer je meerdere blokjes stapelt. Dit is een simulatie van hoe drager robots worden gebruikt om o.a. radioactief materiaal of warenhuis spullen te vervoeren. Als het object valt komen wij in de problemen, in het eerste voorbeeld is er gezondheid (HBO series Chernobyl heeft een voorbeeld). In de tweede voorbeeld moet het ook perfect zijn, een enkele fout leidt tot een ketting reactie die een warenhuis stil kan laten staan waardoor veel geld in de goot komt
- Security bot: leg de robot in een rustige plaats (sensoren detecteren niks). Verzorg er nu voor dat de nek (servo) van de muis iedere paar second rondkijkt. De file "example sweep" onder servo en "for-loops" kunnen jou hier helpen. Als de Arduino nu zit rond te kijken, en iets opmeet, verzorg ervoor dat de Arduino richting wat het heeft gevonden gaat rijden. De security bot bewaakt een plaats, en gaat achter iemand aan wanneer die hem ziet. Je kan dit leuker maken door blink aan te doen wanneer het iemand heeft gedetecteerd
- Jager bot: zorg ervoor dat de muis een bewegende object of persoon kan volgen. Bonus als je dit combineert met de bovenste opgave
- Lijn volger bot: gebruik de sensoren en een zwarte lijn (stift, stof, plakband, et cetera) om een lijn volger Arduino te maken. Youtube en Google hebben veel voorbeelden
- Random bot: gebruik random(min, max) om de robot een willekeurige richting te laten rijden. Laat het los in een doolhof en kijk hoelang het duurt voordat het doolhof opgelost is. Dit is een

voorbeeld van een toevalsbeweging (random walk) algoritme. Hier is het niet super efficiënt, maar toevalsbeweging algoritmen zijn gigantisch in de wetenschap en financiën. Bijvoorbeeld heeft Albert Einstein een pre-computer versie ervan gebruikt om het probleem van Browniaanse beweging (stuifmeelkorrels) op te lossen, een belangrijk bewijs stuk voor het bestaan van atomaire/moleculaire deeltjes

4 Bonus Opdracht: Tom & Jerry of Minotaurus & Theseus

De katten speler heeft dezelfde opdrachten gedaan als de muis, hier zijn nog wat handige opdrachten om voor te bereiden op het spel

Taak 12: Kattenkwaad

1. De kat moet sneller dan de muis origineel zijn, check met de muis speler hoeveel sneller de kat is met een simpele race. Gebruik delay om de snelheid te veranderen
2. De kat en muis zijn idealiter beide aan de langzame kant, gezien de kat wint wanneer het de muis aanraakt. Voor de veiligheid probeer te verzorgen dat beide kat en muis langzaam genoeg aanbotsen dat het op zijn hoogst een stootje is wanneer zij elkaar raken

Spelregels van Kat & Muis

1. De kat, muis en kaas worden neergezet in een doolhof. Het liefst door een derde maar indien er geen derde is, laat de katten speler de muis en kaas neerzetten en de muizen speler de kat
2. De muis zoekt nu naar kaas en de kat zoekt naar de muis. Als de muis de kaas vindt voordat de kat de muis vindt, dan wint de muis. Als de kat de muis vindt voordat de muis de kaas vindt, dan wint de kat
3. Wanneer 1 speler wint, en beide spelers zin hebben in nog een potje, dan krijgt de verliezer een bonus om het interessant te houden. De verliezer mag de code updaten op zijn kat of muis, maar de winnaar niet. Een alternatief, de verliezer kan het doolhof veranderen. De verliezer mag niet beide doen in een enkele ronde
4. De twee spelers blijven spelen totdat de verliezer eindelijk wint, en nu mag de andere speler die verloren heeft zijn code updaten of doolhof veranderen

Dit is een asymmetrisch spel, 1 speler heeft altijd een voordeel over de andere. Maar omdat de verliezer het privilege krijgt om zijn code of het doolhof te veranderen, kan de asymmetrie vallen in hun voordeel. Dit is een waardevolle oefening voor beide partijen, want het eist van hun om niet alleen een werkende Micromouse te hebben maar een betere muis dan hun tegenstander. Omdat de balans constant zit te schommelen zijn de spelers nooit zeker of hun robot altijd zal winnen.

Hier zijn nog een paar ideeën

- De sensoren kunnen worden gebruikt als snelheid detectors, waarmee de robots op elkaar hun beweging kunnen reageren. Dit is moeilijk om te doen, een mogelijke manier om een werkende programma te maken is om de robot periodiek stil te laten staan, de detectoren een scan te laten doen voor beweging en als het iets ziet meteen die richting te bewegen
- Bedenk hoe je de servo kan roteren maar niet de infrarood sensoren. Je kan dit slim gebruiken door alleen met je ultrasoon sensor te kijken zonder jouw muis helemaal om te draaien. Dit is makkelijker gezegd dan gedaan, maar het geeft een groot concurrentievoordeel
- Een inzicht vraag, 1 van de Arduinos heeft zes batterijen en de andere vier. Heeft dit een effect op de motoren, of verzorgt dit alleen ervoor dat de ene Arduino langer meegaat dan de andere?

- Om minder voorspelbaar te zijn kun jij *random(min, max)* implementeren in jouw code om hier en daar jouw muis zijn strategie te veranderen. Dit is een tweezijdige zwaard, je kan hiermee per ongeluk in jouw tegenstander zijn voordeel spelen, of in zijn nadeel. Een situatie waar random handig zou zijn is als je niet simply connected doolhoven gebruikt en de wall follower algoritme daardoor in loops komt, als je er nu verzorgt dat het hier en daar willekeurige de andere richting gaat, verzorg je ervoor dat het uiteindelijk uit zijn loop komt door stomme geluk

5 Appendix: Sensoren

Hier zijn twee subsecties waar je een korte uitleg krijgt over hoe de sensoren werken. De opgaven zijn er om jou een groter inzicht te geven in de technologie zelf, in plaats van hoe jij het gebruikt.

5.1 Infrarood Sensor

Licht komt in verschillende golflengten, sommige kunnen wij met onze ogen zien (380-750 nanometer) en de andere zijn onzichtbaar. Infrarood (micrometer tot millimeter golflengte) is een voorbeeld van licht die wij niet kunnen zien maar wel meten met apparatuur. De Micromouse gebruikt twee "infrared reflection sensors TCRT5000". De sensor heeft een lampje dat licht uitzendt, het licht wordt gereflecteerd en er is een infrarood detector op de module die jou vertelt wanneer het denkt dat het het gereflecteerde signaal heeft teruggekregen. Gebaseerd op de kracht van het gereflecteerde signaal vergeleken met het uitgezonden signaal en hoeveel tijd is verstreken krijg je meteen een afstand.

1. Er wordt aangenomen door de sensor dat licht altijd met licht snelheid reist. Kan jij tegenvoorbeelden bedenken? En zo ja, kunnen zij een probleem voor jouw Micromouse veroorzaken?
2. Er is een thermo-grafische plaatje van dieren te zien op de volgende pagina, weet jij waarom de kleuren variëren en wat de biologische oorzaak kan zijn (bron: www.worldofwarmth.com)?
3. Nu dat wij toch denken over straling en temperatuur, 1 van de belangrijkste stukken bewijs voor de oerknal is kosmische achtergrondstraling. Het bestaan van dit straling was voorspelt door de oudere versies van de oerknal theorie, en het was voorspelt dat het ongeveer 5 kelvin heet was. Gebruik $E = k_b T$ (k_b is de Boltzmann constante) om de energie te berekenen, dan de golflengte (met de wet van Planck en golfsnelheid). De ontdekking van dit straling was per ongeluk gedaan door Penzias en Wilson met hun telescoop. Wat voor soort telescoop (infrarood, gamma, radio) was dit denk je?
4. Maar als infrarood verbonden is met temperatuur, betekent dat niet dat een donker materiaal dat makkelijk licht kan absorberen er anders uit gaat zien dan een wittere materiaal? Test met jouw Micromouse hoeverre jouw sensoren kleuren kunnen onderscheiden en waarom
5. Sommige kikkers en slangen kunnen in het infrarood zien. Wat voor soort voordelen of nadelen zou dit hebben voor hun, evolutionair gesproken
6. Steve Kalman en andere hebben van Amsterdam en andere locaties infrarood foto's gemaakt. Zoek de foto's op en kom erachter waarom sommige objecten donkerder zijn dan andere

5.2 Ultrasoon Sensor

De ultrasone sensor is een sensor die geluid produceert en detecteert ver boven het menselijke gehoor. De Micromouse gebruikt de HC-SR04.

1. Wat zijn voordelen/nadelen van de ultrasoon sensor vergeleken met de infrarood sensor
2. Kapitein Nemo wil 20000 leuga onder de zee zitten en jij weet dat jij je nu 19999 leuga onder de zee bent. Je wilt natuurlijk niet de zeebodem raken, je zendt een ultrasonische puls naar beneden en over een tijdsverschil van $\Delta t = 2s$ reist het terug. Je googelt de snelheid van ultrasonische pulsen rond jouw diepte en schat hem op $3800 \frac{m}{s}$. Is het veilig, gebaseerd op jouw meting, om naar beneden te gaan?

3. Bonus: als je de vorige opgave hebt gedaan, laten wij nu aannemen dat de kapitein een betweter is. "Nou leuk jouw calculatie, maar golfsnelheid in water varieert gebaseerd op diepte, druk, en salinitet(zout). Hoe weten wij of jouw calculatie niet een dikke factor twee er vandaan zit?!"
4. Hoefeizerneuzen (vleermuizen) zijn bekend voor hun abnormale uiterlijk, dit komt doordat hun gezicht gebouwd is voor echolocatie net als sommige walvissen en dolfijnen. Dit eist een vraag: dieren met echolocatie variëren veel met hoe groot ze zijn, maar kunnen allemaal ongeveer dezelfde *golflengte* gebruiken.
Wanneer zou jij voor een detectie met sonar (of zelf met licht net als de infrarood sensor) kiezen voor een lange of korte golflengte? Hint: denk aan een stok in het water met golven die er langs vloeien, de stok beïnvloedt de golven, maar het is belangrijk hoe groot de stok is vergeleken met de golven, toch?
5. Bonus: de Hoefeizerneuzen, en andere vleermuizen, zijn geweldig met het kunnen inschatten van de snelheid van hun prooi, maar dit is vreselijk moeilijk met ultrasone sensoren en zulke snelle prooi. Dit gebeurt via een gerelateerde, maar andere fenomeen. Laten wij zeggen dat jij een vleermuis bent, jij zendt een ultrasone golf uit, en jij merkt op dat wanneer een object snel beweegt, de echo een klein beetje anders is in frequentie (dieper of hogere toon). Hoe kan jij dit verschil in frequentie gebruiken om erachter te komen hoe snel en in welke richting jouw prooi gaat? Hint: denk aan andere objecten die veranderen in toon wanneer ze langs jou bewegen. Waarschuwing, dit is niet iets dat vleermuizen actief aan nadenken, dit is allemaal automatisch en natuurlijk voor hun, net als hoe je hart klopt zonder dat je er specifiek over nadenkt

