

WEEKENDSCHOOL – PROGRAMMEREN –

LES 2B – BEGELEIDERS INSTRUCTIE

1. INTRODUCTIE

Dit is de beschrijving van de extra opgave (2C) van de tweede les programmeren. Het geeft achtergrond informatie, maar ook de uitgewerkte stappen.

2. LES 2C: DE UITWERKINGEN VAN DE OPDRACHTEN

2.1 JITTER VAN DE JOYSTICK

Bij de joystick moeten we analoge waarden meten. Het blijkt dat in sommige opstellingen de servo's, als ze in de uiterste positie staan mechanisch geblokkeerd worden terwijl het motortje probeert verder te draaien. Dat hoor je ook omdat de servo dan wat zoemt.

De betekenis is dat de servo stroom neemt en kennelijk een variabele stroom, want je ziet de gemeten waarden aan de uitgang van de joystick dan wat op en neer dansen. We noemen dat *jitter*. Dat komt omdat door de variabele stroom die de servo neemt de spanning wordt beïnvloedt. Weliswaar zit er een stabilisatie op het rode bordje (als de 9V voeding is aangesloten) of op de Arduino (voor het geval de voeding van de USB wordt gebruikt), maar kennelijk is er toch nog een effect.

Dat betekent dat de joystick ook andere waarden zal geven als je hem niet aanraakt.

Ze kunnen die jitter verminderen (maar niet helemaal kwijtraken) door de servo's in een andere positie te sturen (b.v. 90 graden). Ze zouden ook de servo's kunnen afkoppelen, maar ik wil eigenlijk niet stimuleren dat ze aan draden gaan trekken.

2.2 JOYSTICK BEPAALT POSITIE OF BEWEGING?

Je kunt de positie van de joystick direct de positie laten bepalen of de snelheid van de beweging. We proberen beide uit.

2.3 MIDDENSTAND VAN DE JOYSTICK IS PER BORDJE VERSCHILLEND

In de joystick zitten twee potentiometers. Die worden door de knop bewogen. De joystick is mechanisch simpel en daardoor kan de midden positie gemakkelijk anders zijn bij elke joystick. Maar je kunt ook niet teveel verwachten van een joystick van minder dan 1 euro. We laten daarom de leerlingen steeds eerst de positie in rust uitlezen (aannemende dat ze dan nog niet de joystick aanraken).

Over tijd kan door b.v. warmte de waarde veranderen. Dan gaat de eend kruipen langs één of beide assen. Omdat de neutrale waarde steeds opnieuw wordt uitgelezen als je opnieuw start zal het

probleem wel meevallen in de praktijk. Om het te voorkomen kun je testen of de waarde minstens 2 van de gemeten middenpositie afwijkt en niets uitvoeren als dat niet zo is. Maar de testen die ik gedaan heb laten zien dat het redelijk acceptabel lijkt zonder die test.

2.4 SCHOKKERIGE BEWEGINGEN

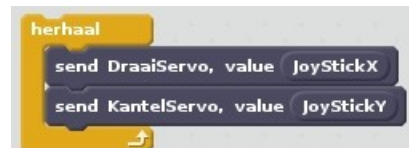
De bewegingen zijn een beetje schokkerig. De eend beweegt in stapjes van één graad en dat zie je. Als de eend heel erg schokkerig beweegt dan laat je mogelijk veranderende variabelen zien in het speelveld. Mijn ervaring is dat de tijd per stap die genomen wordt 15 maal zo groot is als er een variabele wordt vertoont die van waarde verandert.

3. UITWERKING VAN DE OPDRACHTEN

3.1 JOYSTICK BEPAALT POSITIE

OPDRACHT 4: Doe hetzelfde voor JoyStickY en KantelServo.

Omdat de joystick een bereik heeft van 0 tot 1024 en de servo slechts van 0 tot 180 werkt heeft de joystick slechts op in een beperkt gebied invloed.

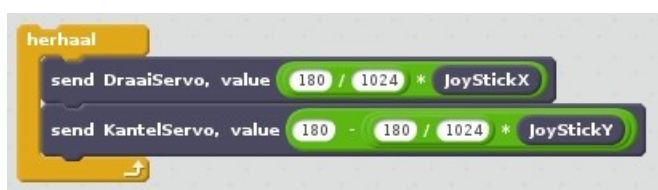


OPDRACHT 5: Pas het programma aan op deze manier. Doe het voor de DraaiServo en voor de KantelServo. Kijk wat het resultaat is en **laat het zien aan je begeleider**.



De ervaring leert dat je beter eerst kunt delen en dan pas vermenigvuldigen om overflow te voorkomen.

OPDRACHT 6: Pas het programma aan zodat de richting van de bewegingen logisch zijn.



3.2 JOYSTICK BEPAALT BEWEGING

Alternatief is dat ze de joystickwaarden gebruiken om bij de X en Y coördinaten op te tellen. Dan moeten ze bij de variabelen voor de X en de Y positie een waarde optellen die afgeleid is van de joystickwaarden. Als midden 512 is (theoretisch zou dat moeten, maar door de mechanische uitvoering van de joystick zou het ook ergens anders kunnen liggen), dan zou dit bijvoorbeeld zo kunnen

$$\text{DraaiServo} = \text{DraaiServo} + 0,01 * (\text{JoystickX} - 512)$$

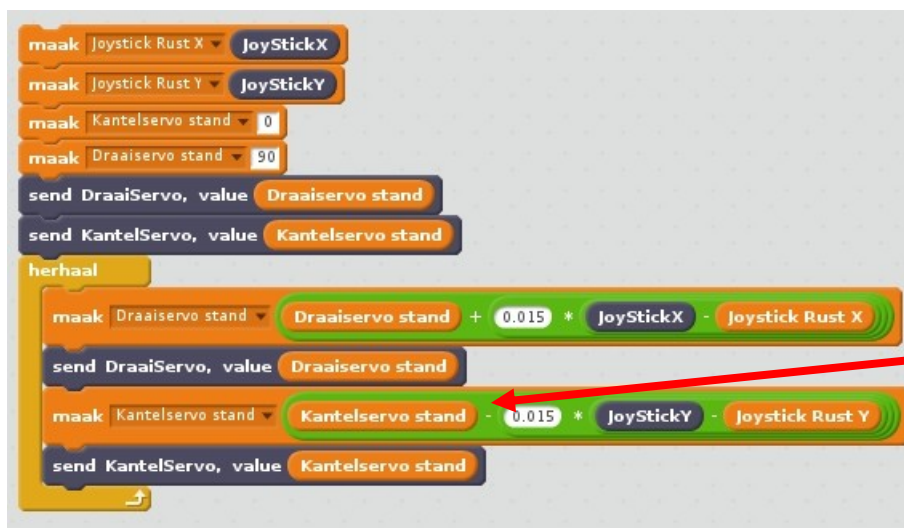
OPDRACHT 8: Maak een nieuw blok met de formule hierboven. Let op: je zult de waarde van de draaiservo in een variabele moeten bewaren, want die heb je de volgende keer weer nodig. En zet de servo's eerst in het midden rechtop, dus draaiservo op 90 en kantelservo op 0.

Als je het klaar hebt, probeer het dan uit. **Laat het zien aan een begeleider.**



OPDRACHT 9: Doe hetzelfde nu voor de kantelservo.

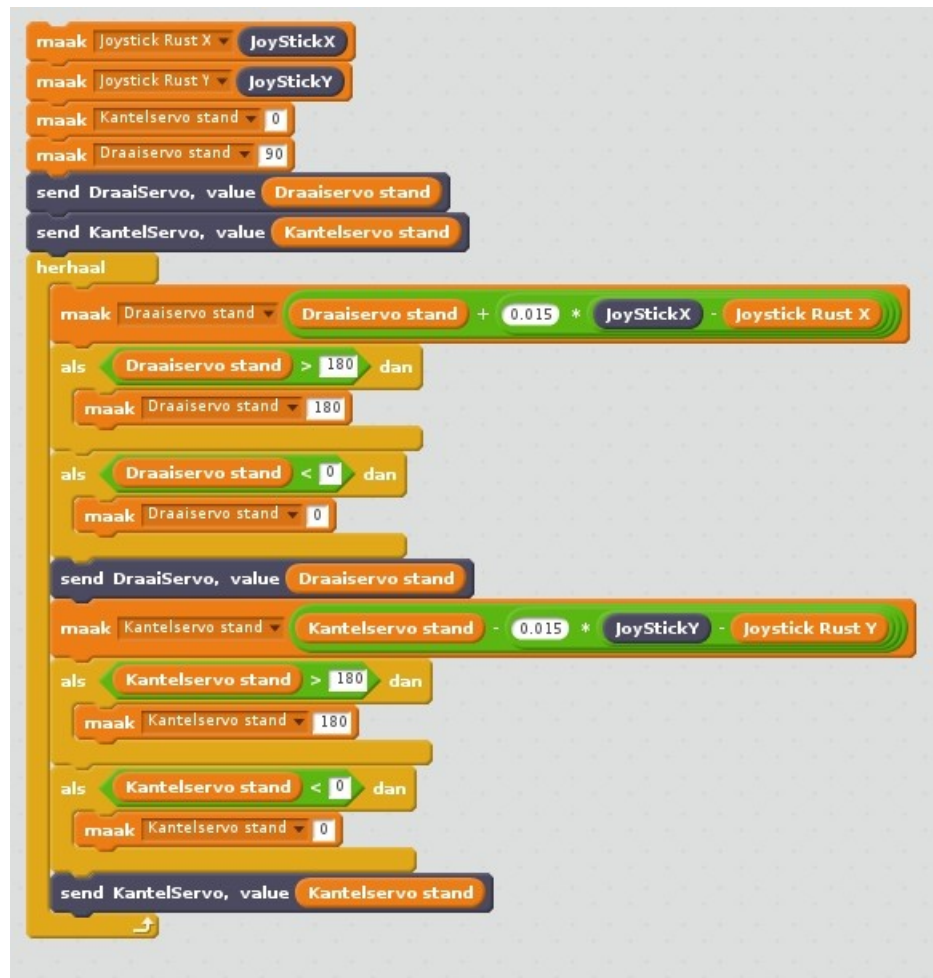
OPDRACHT 10: Pas het programma aan zodat het logisch werkt. Daar zijn verschillende mogelijkheden voor. Het simpelste is een enkel minteken op de juiste plaats ...



Door hier een minteken te gebruiken buigt de eend naar je toe als je de joystick knop naar je toe beweegt.

OPDRACHT 11: Pas het programma zo aan dat de variabelen niet buiten de 0 tot 180 kunnen komen. Het zou slim zijn om nu een eigen blok te maken hiervoor, want straks moeten we nog meer naar de joystick kijken. Dus dat mag je ook doen. Kijk welke variabelen er zijn en definieer een eigen blok zoals je dat ook in les 2B gedaan hebt. Heb je les 2B niet gedaan? Vraag dan een begeleider.

Dit is als ze geen eigen blok maken.



Als ze wel een eigen blok maken, laat ze dan eerst kijken welke variabelen er in het spel zijn:

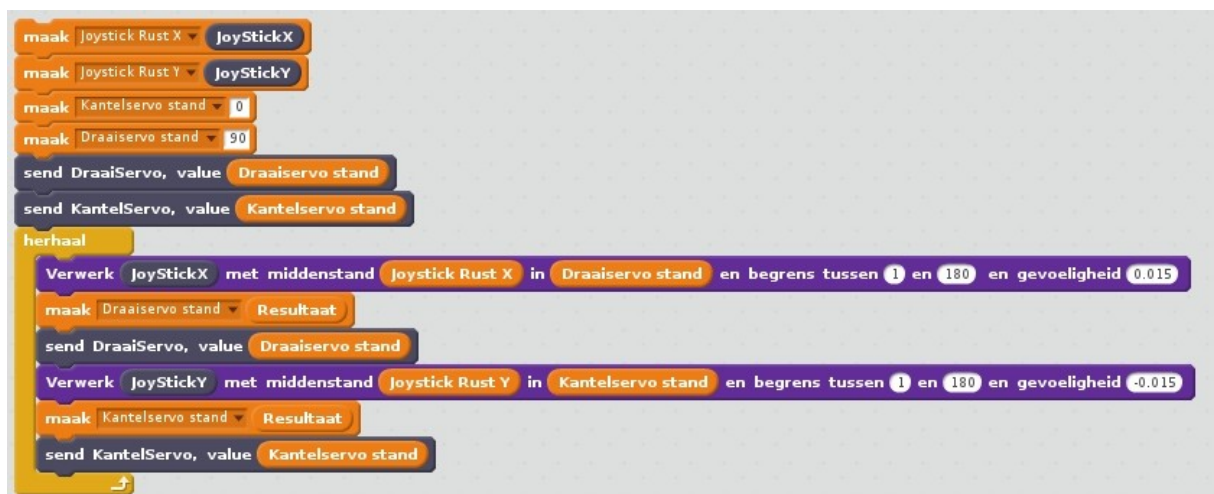
- *Joystick waarde*
- *Joystick ruststand*
- Stand van de servo. Maar we gaan straks het ook voor iets anders gebruiken, dus *waarde* is misschien beter.
- Ene uiterste van het bereik. Noem dat *laag*. Ze denken misschien dat ze dat vast op 0 kunnen zetten, maar als we het straks voor iets anders gebruiken dan hebben we een andere waarde nodig.
- Het andere uiterste van het bereik. Noem dat *hoog*.
- Het is handig om ook de *gevoeligheid* te kunnen besturen, al was het maar om het minteken voor de richting kwijt te kunnen.

Helaas kan een eigen blok in Scratch geen waarde teruggeven. De enige manier is om een variabele te maken waar het resultaat in teruggegeven wordt. Noem die variabele *Resultaat*.

Dit is de definitie van het eigen blok.



En hier het programma met tweemaal een aanroep.



4. GEBRUIK VAN DE JOYSTICK OM TE TEKENEN

OPDRACHT 16: Zet deze blokken op de juiste manier samen

Je moet de flanken van het signaal detecteren. In dit geval een neergaande flank, van 1 naar 0. Daartoe wacht je tot de knop losgelaten is. Dan wacht je tot hij ingedrukt is. Daarmee heb je de overgang te pakken en kun je actie ondernemen: de pen omlaag laten gaan. Dan wacht je weer op de volgende neergaande flank als de knop weer ingedrukt wordt. Dan kun je de pen weer omhoog halen.

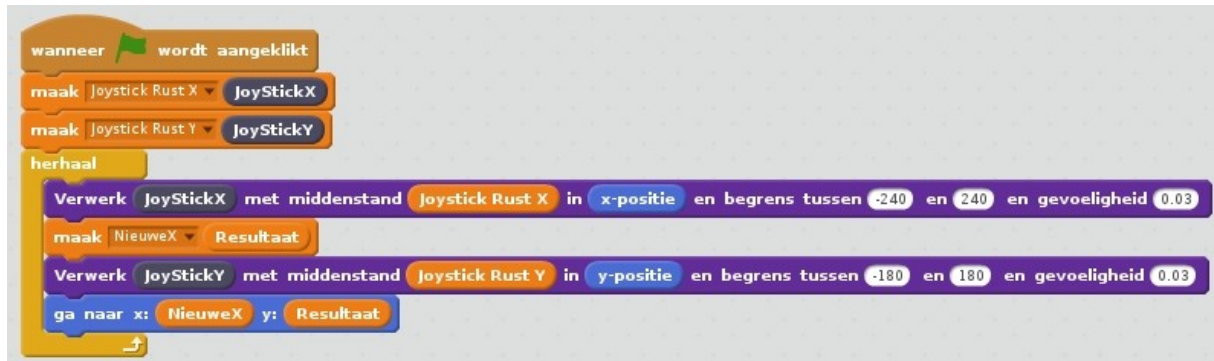


OPDRACHT 17: Nu komt een grote stap. Je moet het potlood laten bewegen door de joystick te bewegen. Veel lijkt erop. Probeer het uit. **Doe het eventueel samen met je begeleider.**

Het is hier uitgewerkt voor als je eerder een eigen blok hebt gemaakt.

Dat eigen blok werkt – heel hinderlijk – alleen in de sprite waar die is opgenomen. Om hem ook bij het potlood te gebruiken maken we een kopie naar het scriptsblad van de potlood sprite.

Daarna maken we een script in de potlood sprite die de sprite beweegt aan de hand van de joystick. Bedenk dat je de x en y eerst beiden moet berekenen en dan naar de nieuwe coördinaten kunt bewegen. Als je het voor x en y apart doet krijg je een kartellijn. Omdat bij het berekenen van y de variabele *Resultaten* weer verandert, zet je de nieuwe waarde van X in een variabele: *NieuweX*.



Dit verandert de penkleur. Zolang de rechter knop wordt ingedrukt zal elke 0.05 seconde de penkleur worden veranderd met stapjes van 1.



Deze blokken zorgen voor de lijndikte. Als je de knop indrukt wissel je af tussen de lijndikte vergroten en verkleinen. Door de knop ingedrukt te houden wordt de lijndikte elke 0.05 seconde verhoogd of verlaagd.



OPDRACHT 18: Programmeer dat de tekening wordt gewist door de joystickknop twee seconden ingedrukt te houden.

Eerst wachten tot de knop ingedrukt wordt.

Dan start je een tijdmeting en wacht tot de tijd 2 seconde is gepasseerd. Test nooit op gelijk, want het kan zijn dat dit niet opgemerkt wordt. De conditie kan worden geïnspecteerd net voor en net na de 2 seconde. Daarom testen op groter. Als de wachtconditie is vervuld door de tijd (want



de knop is nog steeds ingedrukt), wis dan alles en stop ook met tekenen (pen op). Je wilt immers niet door het indrukken van de joystickknop dat de pen omlaag gaat in dit geval.

OPDRACHT 19: Als dat allemaal gelukt is, dan kun je ook nog proberen om met de muis te tekenen. Als de muis ingedrukt wordt, dan begin je met tekenen op de plek waar die staat. Je beweegt nu de muis en het potlood volgt. Tot je de muisknop loslaat. Zorg dat het tekenen met de muis niet beïnvloed wordt door de joystick knop. En omgekeerd ook niet. Let op: dit is een hele lastige opdracht.

Eerst wachten we tot de muisknop niet ingedrukt is. Als we dat niet doen, dan gaat de pen direct neer als we op de groene vlag drukken met de muis. Die is dan immers ingedrukt.

Dan wachten we tot de muis ingedrukt is en daar beginnen we te tekenen. Dus eerst de pen omhoog, anders komt er al een streep. Dan naar de muisaanwijzer gaan en daar de pen neerlaten. Dan doorgaan tot de muis niet meer ingedrukt is en tot die tijd de hele tijd de muisaanwijzer volgen. Als de muis niet meer ingedrukt is dan de pen omhoog doen.

Door het zo te programmeren interfereert het niet met de joystick.

