WEEKENDSCHOOL – PROGRAMMEREN – LES 2A – BEGELEIDERSINSTRUCTIE

1. INTRODUCTIE

Dit is de beschrijving van programmeren les 2A. Het geeft achtergrondinformatie, maar ook de uitgewerkte stappen.

2. VOORBEREIDING VOOR DE LES

Het is belangrijk om voor de dag zelf jezelf voor te bereiden.

2.1 ONBEKEND MET SCRATCH?

Als je nog nooit met Scratch gewerkt hebt, dan moet je voor de les even Scratch leren. Als je al ervaring hebt met programmeren in een andere taal dan zul je in een half uurtje wel klaar zijn. Als programmeren nieuw is dan zal het zeker een paar uur kosten.

2.2 SCRATCH LEREN

Begin hier: https://scratch.mit.edu/ en kies *Probeer het* of *Bekijk voorbeeld*. Dan programmeer je in Scratch 2.0. In de les programmeren we ook in Scratch 2.0 maar dan op een Raspberry Pi. Daar zijn wat fouten waar we omheen werken, maar verder is het hetzelfde.

Introducties over Scratch vind je hier: https://scratch.mit.edu/help/

Je kunt niet precies dezelfde geluiden toevoegen etc., maar dat zou geen belemmering moeten zijn. Als je dat wel wilt, vraag dan even om de geluidsbestanden.

In de les gebruiken we een Raspberry Pi als computer. Die werkt bijna net zoals een Mac of Windows PC. Daar hoef je je niet op voor te bereiden.

Mocht je zelf een Raspberry Pi 2B of 3B hebben dan kan ik je ook een image voor een micro SD kaart sturen. Voor deze les 2A heb je ook minstens een Arduino Nano nodig. Die kan ik je sturen.

Als je wilt kun je ook een complete setup met Raspberry Pi, monitor, toetsenbord, plankje met eend etc. lenen.

2.3 HULP VOOR BEGELEIDERS

Als je ergens niet uitkomt, schroom dan niet om voor de les te bellen met Hans de Jong, 06 204 234 84. Kan in het algemeen elke dag tot 23 uur. Op de dag zelf kun je nog wel wat vragen, maar er is heel weinig tijd en daarom is het verstandig om alle vragen tevoren beantwoord te krijgen.

3. VOOR DE LES

3.1 OPBOUW VAN DE WERKPLEKKEN

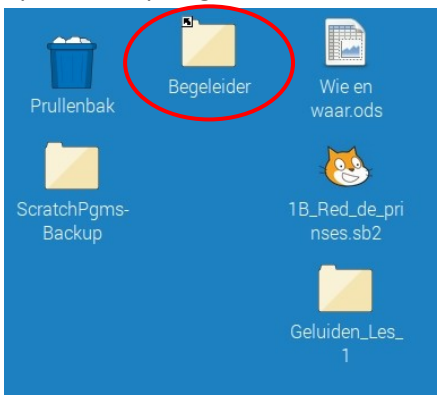
We proberen het zo te plannen dat er per begeleider maximaal 8 leerlingen zijn. Die zitten op 4 werkplekken. We proberen meer begeleiders te hebben. Elke begeleider bouwt dus maximaal 4 werkplekken op. Er is een aparte handleiding hoe dat moet.

3.2 KLAAR ZETTEN VAN DE RASPBERRY PI

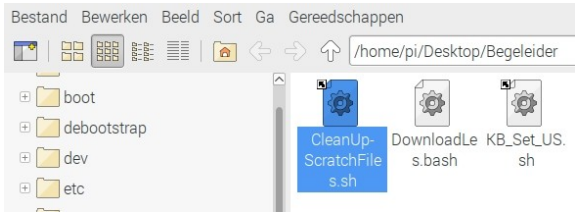
Als de werkplek klaar is, start dan de Raspberry Pi op. Daarna moeten er nog wat handelingen gedaan worden.

3.2.1 START CLEANUP-SCRATCHFILES.SH

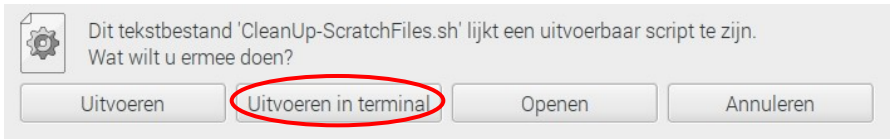
1. Open de map Begeleider



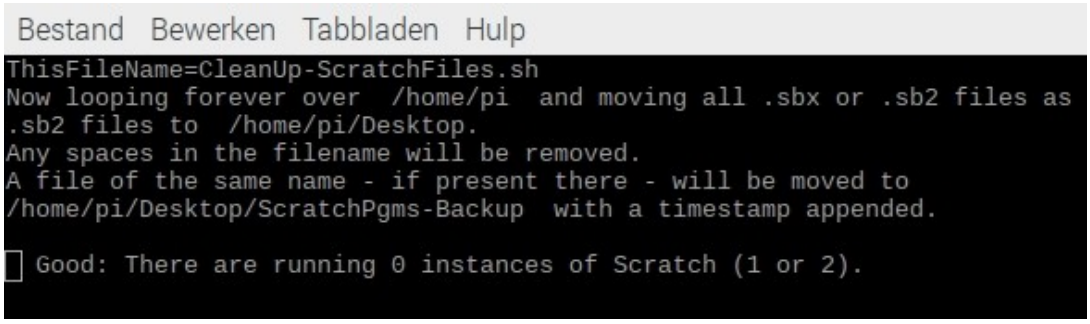
2. Dubbelklik op CleanUp-ScratchFiles.sh



3. Kies *Uitvoeren in terminal*



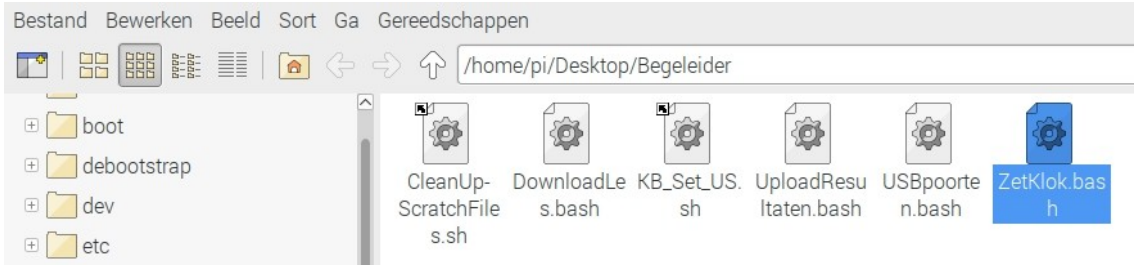
4. Je ziet nu dit venster, maar het kan ook zijn dat het venster vanzelf geminimaliseerd wordt. Als het zichtbaar is, minimaliseer het dan. **Het moet de hele les blijven draaien**.



Dit programma zorgt dat als je een .sb2 of .sbx file opslaat in /home/pi deze weer als – altijd – een .sb2 file op het bureaublad wordt gezet, met de eventuele spaties verwijderd uit de filenaam. Dit is nodig wegens fouten in Scratch 2 op Raspberry Pi.

3.2.2 ZET DATUM EN TIJD

1. Dubbelklik ZetKlokEnSysteem

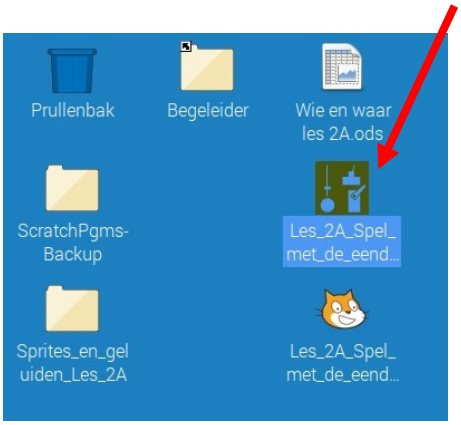


2. Kies *Uitvoeren in terminal*

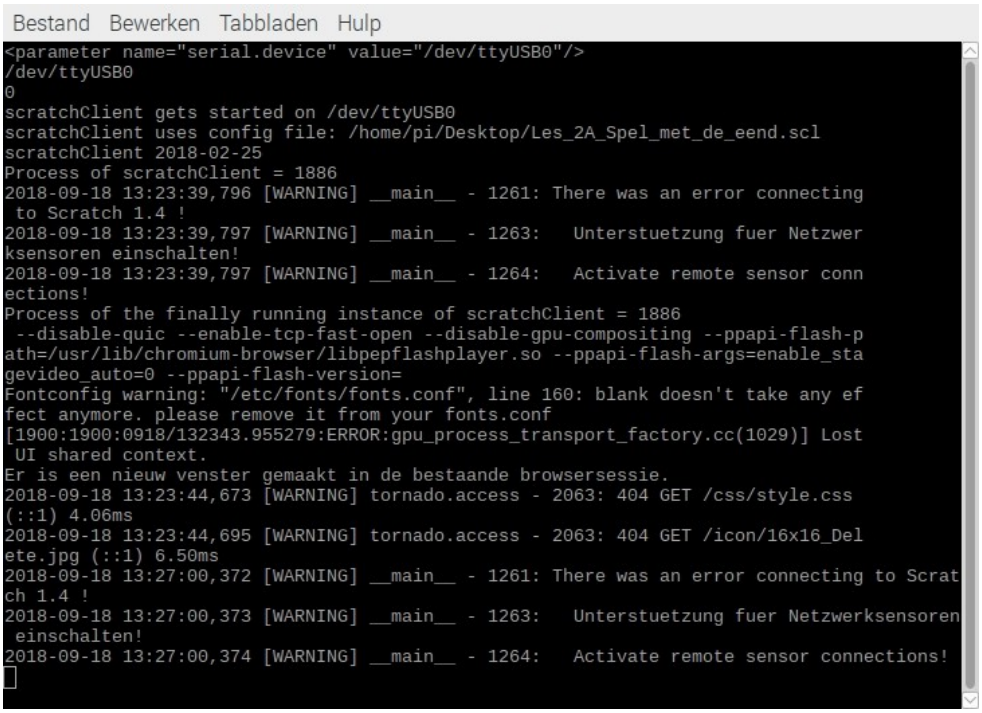
3.2.3 ZORG DAT SCRATCHCLIENT IS OPGESTART

Let op: deze stappen doen we pas als we met les 2A beginnen. Dus in de ochtend na het opzetten of in de lunchpauze als we les 2A pas in de middag doen.

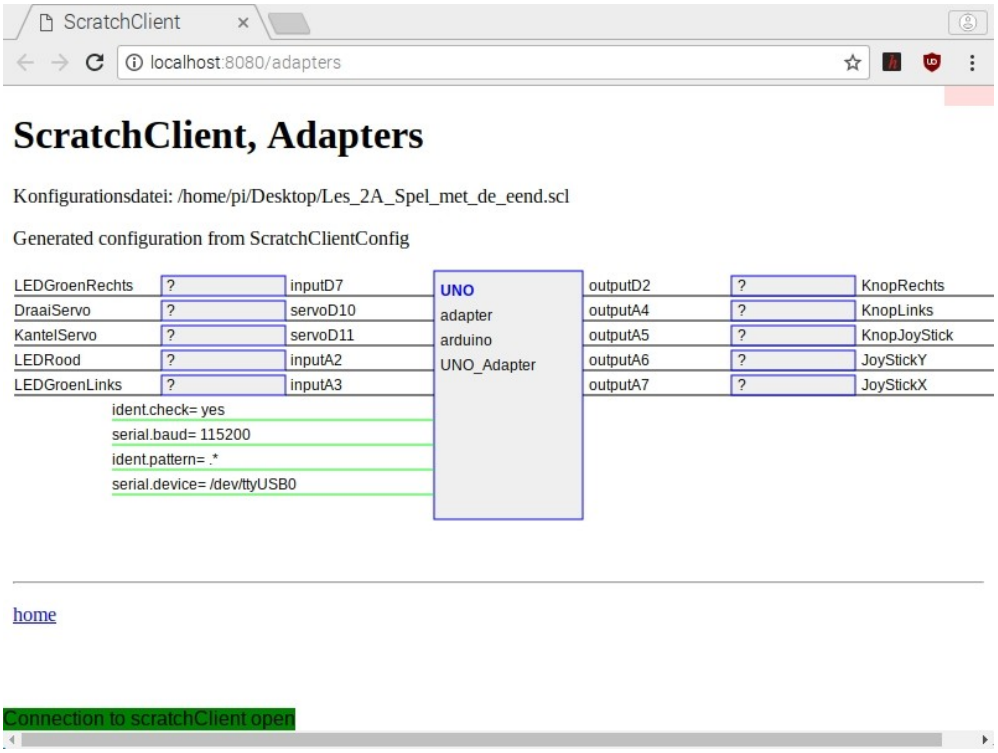
ScratchClient maakt de verbinding tussen Scratch en de Arduino op het plankje met de eend. Dubbelklik daarvoor het scratchClient icoon.



Je krijgt dan dit venster:



Ook wordt een browser gestart. Daarin kun je de variabelen zien zodra die een waarde hebben gekregen.



4. GEZOND COMPUTEREN

Afhankelijk van of we het vorige week al gedaan hebben geven we een korte introductie over hoe je gezond moet computeren. Als we dat wel gedaan hebben, dan vragen we ze om de 5 regels uit het filmpje.

- We beginnen met een introductie over hoe ze moeten computeren zonder gezondheidsklachten te krijgen. Dit gebeurt door de gastdocent. Er is een aparte instructie. Begeleiders hebben geen functie in dit onderdeel.
- Daarna spelen we stukjes video af
  http://www.hetklokhuis.nl/tv-uitzending/1142/Gezond%20computeren
  Van 4:50 tot 8:35 en daarna van 11:07 tot 12:23.

5. WAT ZE GAAN MAKEN

- We laten een filmpje zien over wat ze deze les gaan maken. Dit is de Scratch 1.4 versie. De Scratch 2.0 versie moet nog opgenomen worden.
  https://www.youtube.com/watch?v=Qo1gnXNzhqE
- We leggen we uit dat programmeren de één na laatste stap is. De laatste stap is testen. Maar het begint met het analyseren van het probleem en bedenken hoe je het aanpakt. Dat gaan we daarom eerst doen. Daarvoor gaan ze in groepjes (een aantal tweetallen) met een begeleider zitten en praten over welke poppetjes er zijn en wat die doen.
- De begeleider zit hierbij samen met een aantal tweetallen. Als je klaar bent met de discussie dan ga je naar de computers.
- Dan beginnen we met de opgave *Les_2A_Spel_met_de_eend*

6. WELK LEERPROCES?

De Lifelong Kindergarten group van het MIT in Boston, waar Scratch ontwikkeld is, propageert het Creative Learning: leren door te doen. Zij leggen de nadruk op het door de leerlingen zelf laten ontdekken en spelen en zo te leren. Ik geloof daar ook zeer in, echter we hebben maar drie zondagen en willen ze toch wat meer meegeven. Daarom is gekozen om ze opdrachten te laten maken. Voor spelen en eigen inbreng is gelegenheid aan het eind. Maar als ze eerder zelf dingen willen uitproberen en een wat andere richting in gaan dan is dat prima, maar kijk wel dat ze niet in een onderwerp blijven hangen omdat ze de rest (zogenaamd) niet snappen.

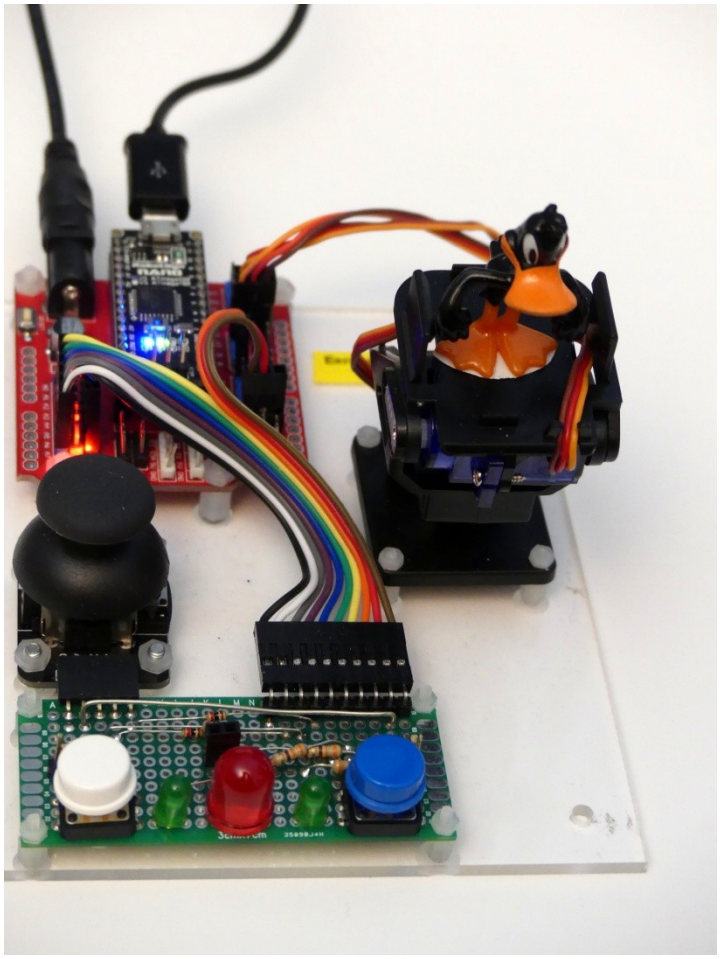
7. LESDOEL VAN LESDAG 2

- Leren hoe ze zonder gezondheidsklachten met beeldscherm, muis en toetsenbord kunnen werken.
- Laten kennismaken met het analyseren van een probleem.
- Verdere kennismaking met programmeren in Scratch.
- Laten zien dat je met heel weinig moeite een animatie kunt maken.
- Ze kennis laten maken met de bouwstenen van Scratch
  - Herhalen

    - Testen op gebeurtenissen (toetsaanslagen, het raken van een object)
    - Beweging
    - Variabelen
    - Geluid
    - Random getallen
    - Zenden en ontvangen van berichten tussen de sprites
- Ze leren dat programmeren niet alleen gaat over het veranderen van pixels op een scherm, maar dat je ook LEDjes, motortjes, schakelaars etc. kunt aansturen.

8. DE OPSTELLING

We gebruiken een Raspberry Pi, monitor, toetsenbord en twee muizen, plus een bordje met knopjes en lampjes en een eend op een draai/kantel plateau. Hieronder staat een foto van een prototype.



9. VERBINDING MET SCRATCHCLIENT MAKEN

Volg nu de leerlingenhandleiding om Scratch te starten (dubbelklik) en de verbinding met scratchClient te leggen (druk op de groene stip). De LED op de Arduino moet nu na een paar seconden langzaam gaan knipperen. Dan is de verbinding gelegd.

9.1 WAT KAN ER FOUT GAAN?

9.1.1 OUTPUT VAN SCRATCH KOMT NIET AAN OF SENSOREN ZIJN NIET ZICHTBAAR

Als de sensoren niet zichtbaar zijn in Scratch terwijl ScratchClient wel loopt en contact lijkt te hebben (als je op een knop op het bord drukt dan zie je iets in het venster veranderen), dan kan het zijn dat er twee Scratch programma's lopen. Stop degene die je niet nodig hebt.

Je zult in dit geval ook zien dat het programma CleanUp-ScratchFiles.sh in de statusbalk knippert.

Tot hier is het bijgewerkt. De rest volgt later.

10. DE UITWERKINGEN VAN DE OPDRACHTEN

10.1 DE LINKER GROENE LED LATEN BRANDEN ZOLANG ER WORDT AFGETELD.

10.2 DE RECHTER GROENE LED LATEN BRANDEN GEDURENDE DE LAATSTE 5 SECONDEN VAN HET AFTELLEN

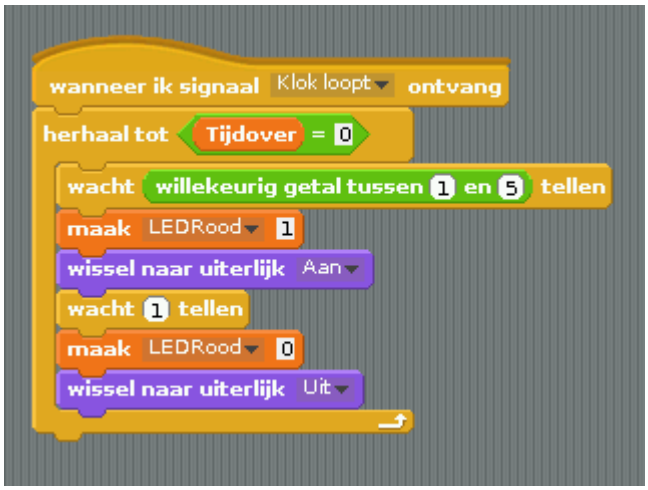
10.3 KLOK LATEN STARTEN ALS JE OP DE JOYSTICKKNOP DRUKT

10.4 DE EEND DIE OP BERICHTEN REAGEERT

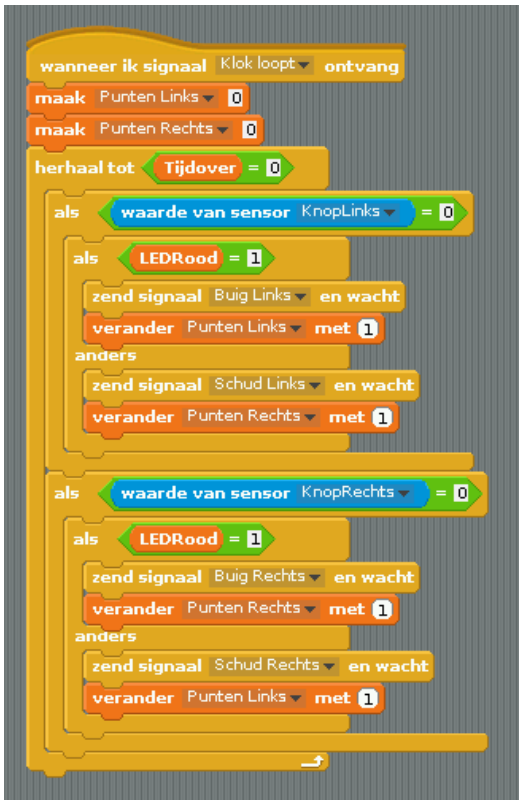
11. HET ECHTE WERK

11.1 STAP 1: ZEND EEN SIGNAAL ALS DE KLOK LOOPT

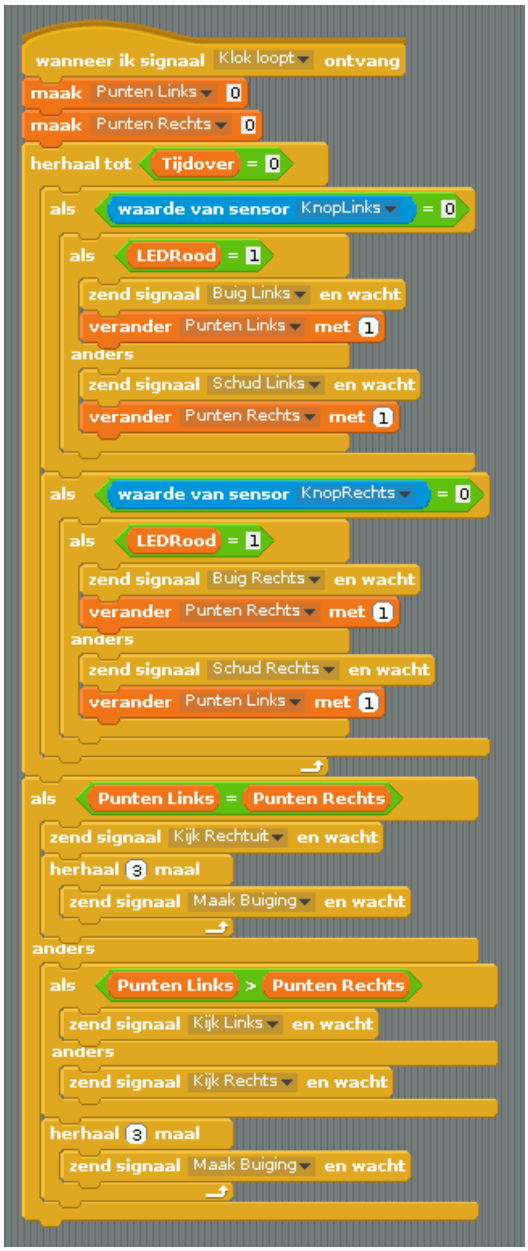
11.2 STAP 2: EEN NIEUWE SPRITE: DE RODE LED



11.3 DE KNOPPEN BEKIJKEN



11.4 WIE IS DE WINNAAR



11.5 GELUID

12. MAAK HET ZO DAT JE HET THUIS OOK KUNT SPELEN

Ze moeten een test toevoegen bij het testen of een knop is ingedrukt om ook te kijken of een toets op het toetsenbord is ingedrukt.

13. IS HET SPEL EERLIJK?

Het spel kan natuurlijk op veel manieren bewust oneerlijk gemaakt worden. Bijvoorbeeld door een tijdvertraging in te bouwen in de wachtlus die naar de knopjes kijkt, zal het knopje dat als eerste wordt getest na de tijdvertraging een grotere kans hebben om te worden gedetecteerd als winnaar.

Maar er kunnen ook onbewust elementen van onbalans in zo'n systeem zitten. En een vraag die ze zichzelf zouden moeten stellen is: wat gebeurt er als beide knopjes tegelijk worden ingedrukt? Wie wint er dan?

En hoe zou je dat kunnen testen? Hoe kun je de knopjes exact tegelijk indrukken? Dat doen we door beide knopjes aan elkaar te verbinden met een draadje. Als je dan op 1 knopje drukt, activeer je beide signalen.

Dus laat ze het draadje bevestigen. Dan is het ook handig om de random tijd voor het rode LEDje te vervangen door een constante tijd van b.v. 2 seconde, of minder als je ook stopt om de eend te laten bewegen.

Laat ze kijken wat er gebeurt. Het blijkt dat het programma niet eerlijk is. Ik heb nog niet gevonden waar het precies aan ligt.

Vraag ze om te bedenken hoe vaak ze het zouden moeten testen om betrouwbaar te zijn. En vertel ze dat er wiskunde bestaat (kansberekening) om dat goed te kunnen bepalen. Dat krijgen ze op de middelbare school.

En vraag ze ook: "Als nu steeds de ene wint, wil dat zeggen dat het spel echt oneerlijk is?".

Leg ze voor wat er zou gebeuren als het zo was dat als het linkerknopje 1 ms (test eerst even of ze weten dat milli staat voor 1/1000) later wordt ingedrukt de kans op winnen wel 50% is, terwijl als ze echt tegelijk worden ingedrukt het linkerknopje steeds wint. Het spel is dan theoretisch oneerlijk, maar is het in praktijk echt erg? In het verkeer is je reactietijd 1 seconde. In het geval dat je geconcentreerd bent op een LEDje en knopje is dat veel minder. Internet geeft 0,2 seconde aan. Bij hardlopen geldt een reactietijd van minder dan 0,1 seconde als valse start, omdat je dan geacht wordt al gestart te zijn voordat je op het startschot had kunnen reageren.

Dus zou 1 ms vertraging op een 200 ms reactietijd (die ook bij iedereen wat anders is) erg zijn?

Dus als het ene knopje een grotere kans heeft om te winnen, hoe zou je dan je test kunnen aanpassen om te kijken hoe (on)eerlijk het is? Antwoord: zet een vertraging tussen het ene en het andere knopje. Dat kunnen we hier niet doen (geen materiaal).

14. OPDRACHT 2B

14.1 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 betekent dat de servo stroom neemt en kennelijk een variabele stroom, want je ziet de gemeten waarden aan de uitgang 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 USB voeding wordt gebruikt), maar kennelijk is er toch nog een effect.

Dat betekent dat de joystick ook heen en weer gaat 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.

14.2 JOYSTICK BEPAALT POSITIE OF BEWEGING?

14.2.1 JOYSTICK BEPAALT BEWEGING

➔ Vanaf hier moet het nog aangepast worden

Alternatief is dat ze de joystick waarden gebruiken om bij de X en Y coördinaten op te tellen. Dan moeten ze een variabele voor X en een voor Y maken en daarbij een waarde optellen die afgeleid is van de joystick waarde. 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

Xpositie = Xpositie + 0,1 * (JoystickX – 512)

Wacht 0,05 seconde

Door de waarde 0,1 en 0,05 te variëren kunnen ze de gevoeligheid aanpassen. De wachttijd heeft invloed op de reactiesnelheid (hoe kleiner hoe beter) en de belasting van de computer. Bedenk dat er Arduino via ScratchClient ca. 10 berichten per seconde verstuurt, dus de 0,05 is waarschijnlijk al zo gevoelig als het kan.