WEEKENDSCHOOL – PROGRAMMEREN – LES 2D – IS HET SPEL EERLIJK? BEGELEIDERSINSTRUCTIE

1. INTRODUCTIE

Dit is de beschrijving van programmeren les 2D. Kijk voor de algemene informatie het document *Weekendschool Programmeren - Begeleidersinstructie – Algemeen.* En omdat het een vervolg is op les 2A en 2B en daar verder op voortbouwt is het verstandig die leerlingen- en begeleidersinstructie ook te bekijken.

2. OPZET

De opzet is identiek aan les 2A. Aan het eind van de opdracht gebruiken ze ook nog een draad tussen twee pennen.

3. WAT ZE GAAN MAKEN

- Ze moeten bedenken of het spel eerlijk is.
- Dan moeten ze het spel eerlijk(er) maken.
- En tot slot kijken hoe ze kunnen onderzoeken of het spel nu wel eerlijk is.

4. LESDOEL VAN LES 2D

- Ze laten nadenken over wat ze maken en niet klakkeloos laten aannemen dat iets goed is.
- Ze laten nadenken over wat eigenlijk eerlijk is en hoeveel “oneerlijkheid” acceptabel is.
- Ze leren werken met status waarden (LED aan / LED uit / LED afgehandeld)
- Ze leren hoe ze een transitie in een toestand kunnen detecteren i.p.v. alleen te handelen op de toestand zelf.
- Ze leren dat sommige dingen het beste parallel kunnen (knopjes bekijken) en andere dingen beter serieel (beweging van de eend).

5. DE UITWERKINGEN VAN DE OPDRACHTEN

OPDRACHT 1: **Bespreek met een begeleider** hoe het spel allemaal oneerlijk kan zijn.

Het spel kan natuurlijk op veel manieren bewust oneerlijk gemaakt worden.

a. 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.
b. Bijvoorbeeld, als een speler op het knopje drukt, dan luistert de speelveld sprite pas weer als het signaal naar de eend is verwerkt. En dat duurt even. Drukt de andere speler in die tijd op de knop, dan wordt die niet gezien. Om dat te voorkomen is er een heel andere opzet nodig.

En dit is niet alles.

Speelveld

OPDRACHT 3: Maak van deze drie blokken een stapel en verzin welke opdracht er als vierde bij moet komen om te kunnen tellen hoe vaak de linker knop wordt ingedrukt.

OPDRACHT 4: **Bespreek met een begeleider** wat er gebeurt als je dat vierde blok vergeet.

Dan wordt er net zo lang geteld als de knop is ingedrukt. De snelheid hangt af van hoe snel de RPi is en hoe efficiënt Scratch is. Om op de flank van het signaal te reageren moet je zeker weten dat je van de oude in de nieuwe toestand gaat. En er niet al bent.

OPDRACHT 5: Zet in plaats van het blok om de punten links te tellen de hele stapel blokken die je al eerder gemaakt hebt om de punten links of rechts te tellen afhankelijk van de rode LED.

Speelveld

OPDRACHT 6: Verhuis dit blok van de oude stapel naar de nieuwe.

OPDRACHT 7: En het op nul zetten van de punten kunnen we dan beter doen direct na het zetten van de klok. Verhuis die dus ook.

Speelveld

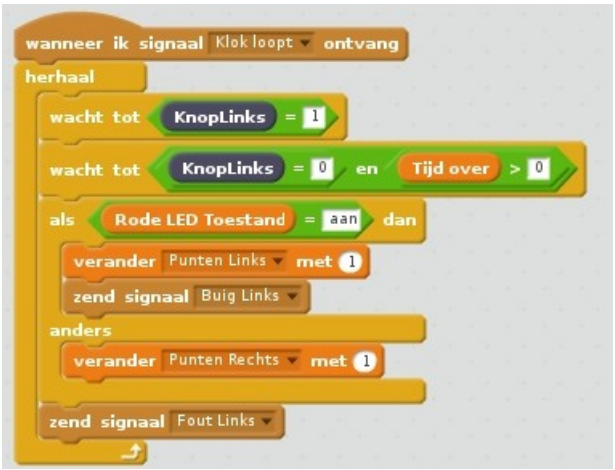
OPDRACHT 8: Probeer uit of de linker knop goed wordt afgehandeld. Test ook wat er gebeurt als je op de knop drukt nadat de tijd over is. Oei, dat gaat dus niet goed. Dat kunnen we simpel oplossen als we bij het kijken naar de knop ook kijken of de klok nog loopt. Zet dit blok op de juiste plek.

Speelveld

OPDRACHT 10: Bedenk hoe dat komt en wat je er aan kunt doen. **Bespreek het met een begeleider** en maak dan de verandering.

Dit ligt aan het blok *zend signaal … **en wacht***. Omdat er gewacht wordt tot de eend klaar is, wordt de volgende druk op de drukknop gemist. Dat wordt opgelost als je alleen een signaal stuurt en *niet* wacht.

Speelveld



OPDRACHT 11: Dan hebben we nog een probleempje. Als de rode LED brandt en je drukt twee keer op de knop, hoeveel punten krijg je dan? En wat als we straks de stapel kopiëren en aanpassen voor de rechter knop, wat zou er dan gebeuren als de beide spelers op de knop drukken? Wie krijgt er dan een punt? Hoe zouden we dat kunnen oplossen? Hint: je moet niet alleen kijken of de rode LED aan of uit is, maar ook … **Bespreek het met een begeleider.** Maak dan de aanpassing.

Speelveld

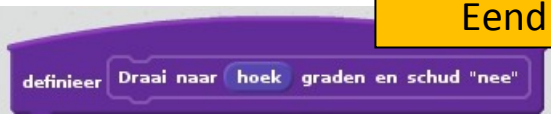
Je moet niet alleen kijken of de rode LED aan is, maar ook of die al afgehandeld is. Als door deze of straks door de rechter knop al gebruikt gemaakt is van het aan zijn van de knop, dan wil je niet nogmaals een druk op de knop behandelen alsof het de eerste is. Daarvoor zetten we zo snel mogelijk de waarde op iets anders dan *aan* of *uit*. Als de LED weer uit gaat dan wordt de status vanzelf weer *uit*. Hierbij houden we even geen rekening met race conditities waardoor net aan het eind dat de LED rood is, de status eerst op *uit* wordt gezet terwijl in de bovenstaande herhaal lus net getest is dat de waarde geen *aan* of *uit* is. Mocht een leerling zo slim zijn om dat op te merken dan nemen we hem of haar onmiddellijk aan als software ontwikkelaar ☺. We komen hiermee op het gebied van semaforen en kritieke secties en gaat veel te ver.

OPDRACHT 12: Kopieer nu de stapel voor de rechter knop en maak de nodige aanpassingen.

OPDRACHT 13: Test wat er gebeurt als beide spelers de knoppen indrukken als de rode LED uit is. Hoe gaat het met de punten? Hoe gaat het met de eend. Snapt die nog wat die moet doen of is die de kluts kwijt? **Bespreek het met een begeleider.**

Speelveld

Door de twee stapels die onafhankelijk van elkaar lopen zal de eend een signaal Fout Rechts en een signaal Fout Links krijgen. Beiden roepen het eigen blok aan (zie hiernaast). Dat blok geeft opdrachten aan de servo's en alles loopt door elkaar heen.



Eend

OPDRACHT 14: **Bespreek samen met een begeleider** hoe je dit probleem zou kunnen aanpakken. En ga het dan maken.

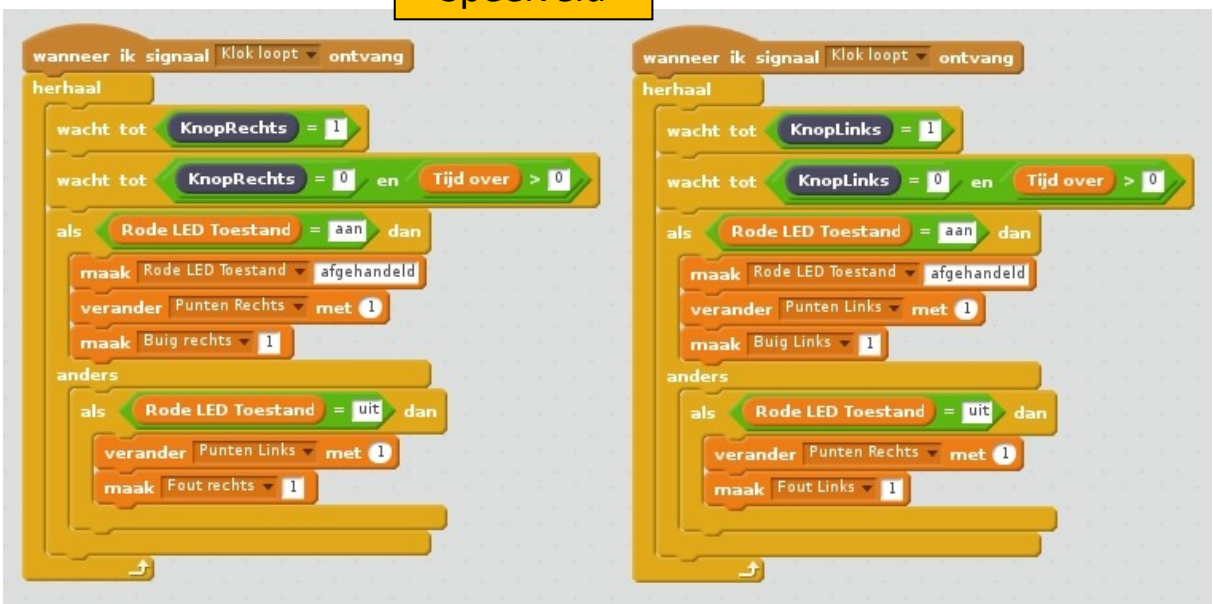
Wat we hier gaan doen is de bewegingen van de eend ontkoppelen van het drukken op de knoppen. Dat laatste willen we asynchroon afhandelen, dus zonder enige afhankelijkheid zodat de punten goed geteld worden. Maar de eend kan effectief slechts één opdracht tegelijk uitvoeren, dus serieel.

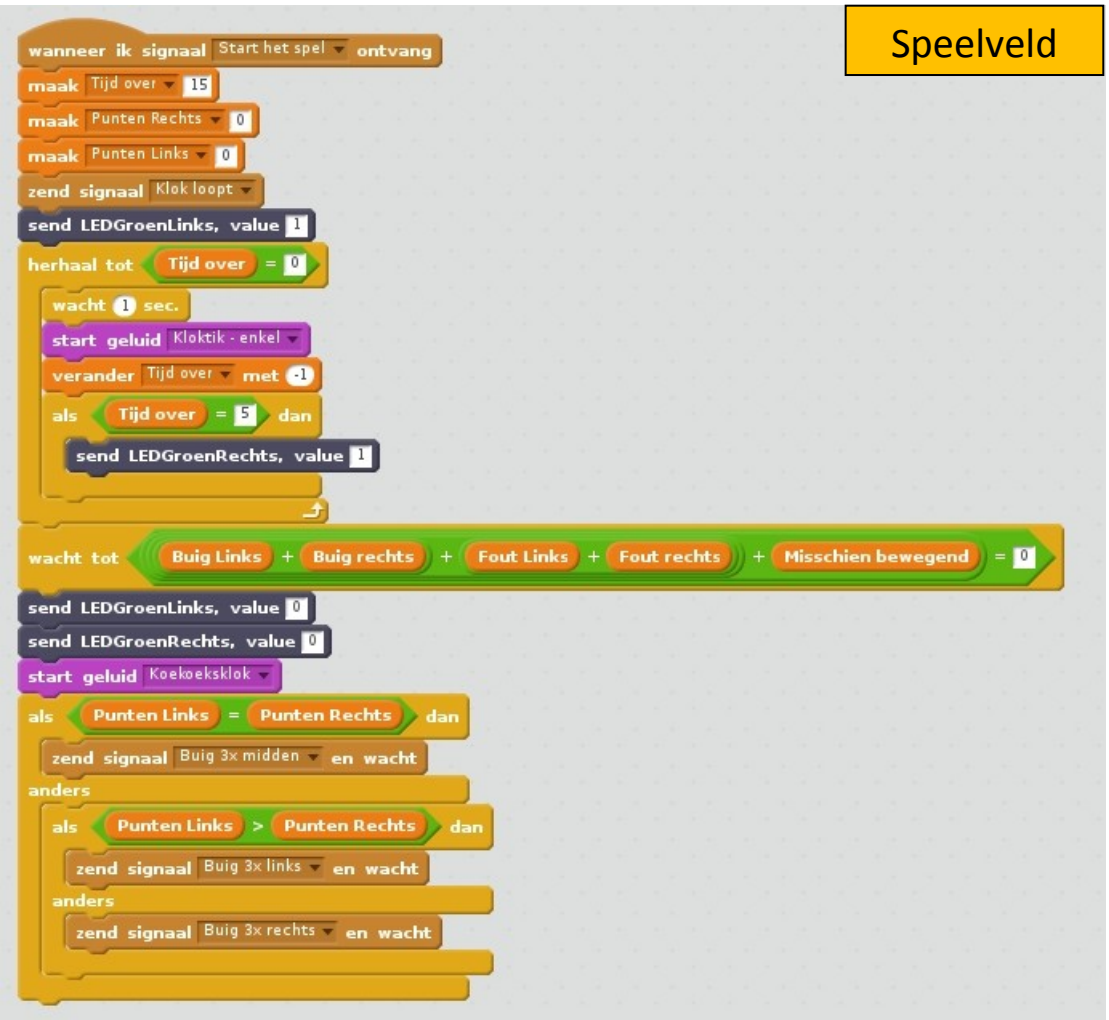
5.1 GEEF DE EEND WAT MEER RUST

Daartoe maken we variabelen zoals *Buig links* en *Fout rechts.* Die variabelen worden gezet waar we eerst het signaal naar de eend stuurden. Ze worden opgevangen door een grote herhaallus waarin we kijken of er iets moet worden uitgevoerd. Dat wordt dan gedaan met een zend signaal … en wacht. Door deze ontkoppeling wordt de puntentelling wel goed gedaan en missen we misschien alleen een beweging van de eend.

Speelveld

Speelveld





Tot slot moeten we zorgen dat de eend pas gaat bewegen voor de puntentelling als hij niet meer zal bewegen wegens drukken op de knoppen.

Daartoe dient het *wacht tot …* blok. Als Buig Links etc. niet allemaal op 0 staan, dan komt er nog een eendbeweging aan. Maar als ze allemaal wel op nul staan kan het bezig zijn dat er nog een beweging wordt afgehandeld. Daarom is er een variabele *Misschien bewegend* gemaakt. Die wordt pas 0 als er niets beweegt. Dus als niets beweegt en ook niet meer gaat bewegen dan kunnen we veilig voor de puntentelling bewegen.

Je kunt kiezen waar je de LEDs wilt laten uitgaan en wanneer de koekoeksklok moet slaan. Hier is gekozen om dat pas te doen als de eend tot rust gekomen is.

5.2 IS HET SPEL NU EERLIJK?

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 krijgt dan een punt?

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 dat je op de twee pennen zet. Als je dan op 1

knopje drukt, activeer je de signalen van beide knoppen tegelijk (afgezien van een paar nanoseconde looptijd in de bedrading).

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. 1 seconde, of minder als je ook stopt om de eend te laten bewegen.

Laat ze kijken wat er gebeurt.

Het is niet zo makkelijk om deze test te doen met de hand. Laat het ze wel één keer doen en zet dan de tijd op 60 i.p.v. 15 seconde.

Maar vraag ze dan of dit niet slimmer zou kunnen. Dat kan. We kunnen een relais aansluiten en die de knoppen laten besturen. Dan kunnen we de test automatiseren.

Sluit het relais aan en start een scratchClient op die voor het relais is voorbereid. Details staan op een blad in de les.

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 elektrische vertraging tussen het ene en het andere knopje. Dat kunnen we hier niet doen (geen materiaal).