

Voorkennis	Klonen, Lagen (voor-achtergrond)	Niveau	
Benodigheden	Stuurtje met Micro:Bit	Leerdoelen	

De auto besturen met stuurtje

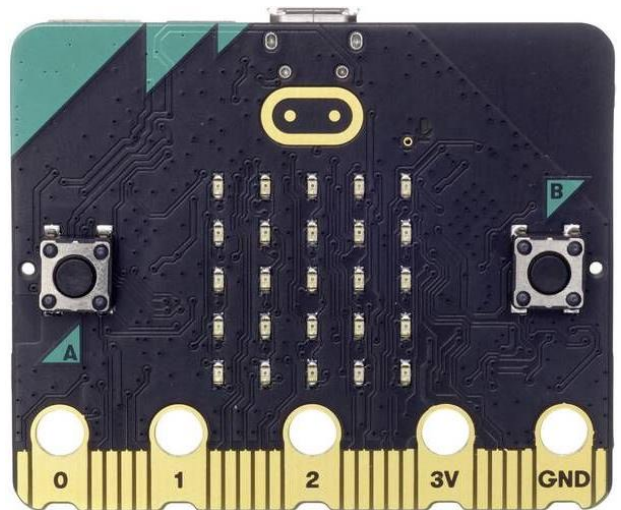
1 Intro

In dit spel moet de auto rondjes rond de school rijden. Iedere keer wanneer het zebra-pad wordt gepasseerd verdient de speler punten.

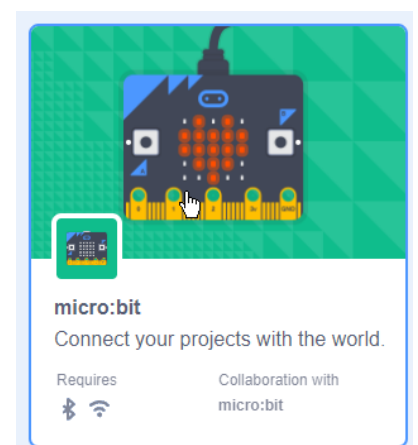
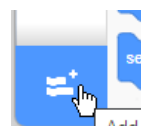
In deze oefening gebruiken we Scratch in combinatie met een Micro:Bit. De Micro:Bit heeft veel sensors waaronder een sensor die die de hoek waaronder de Micro:Bit bevindt kan meten. In het bijbehorende stuurtje zit een Micro:Bit. We gaan de hoeksensor hiervan gebruiken om te meten wat de stand van het stuur is. De kantelhoek die de Micro:Bit aangeeft is gelijk aan het aantal graden is dat het stuurwiel naar links of rechts is gedraaid. De rechterknop gaan we gebruiken om "gas" te geven en de linker knop om te remmen.

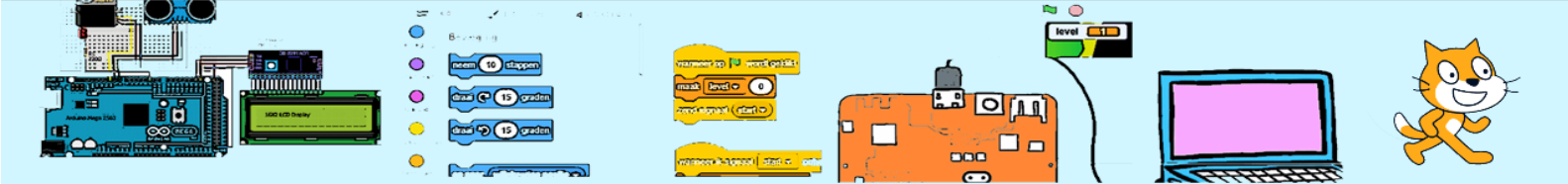
2 Verkennen van de Micro:Bit

De Micro:Bit heeft 2 knoppen die je kan uitlezen. Ook kan je uitlezen onder welke hoek de Micro:Bit worden gehouden. In deze oefening houden we de led's van de Micro:Bit naar ons toe en de USB-connector naar boven.



Om verder te gaan moeten we de Micro:Bit koppelen aan de laptop met Scratch. Vraag een mentor even te helpen. Ook moet de Micro:Bit-extensie worden toegevoegd.





Je kan nu de status van een knop uitvragen.

Maar je kan hem ook gebruiken als trigger.

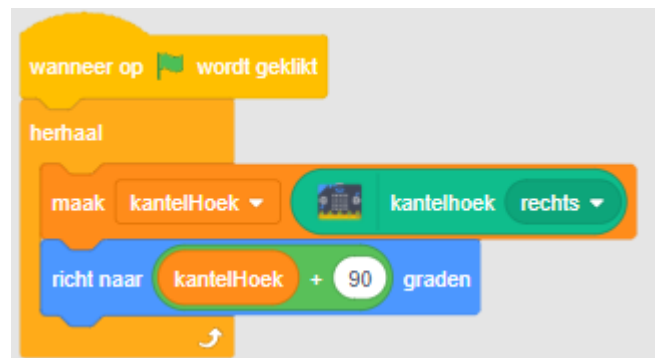
Je kan de kantelhoek uitlezen en je kan hem als trigger gebruiken.



We gaan eerst wat ervaring opdoen met de kantelsensor van de Micro:bit.

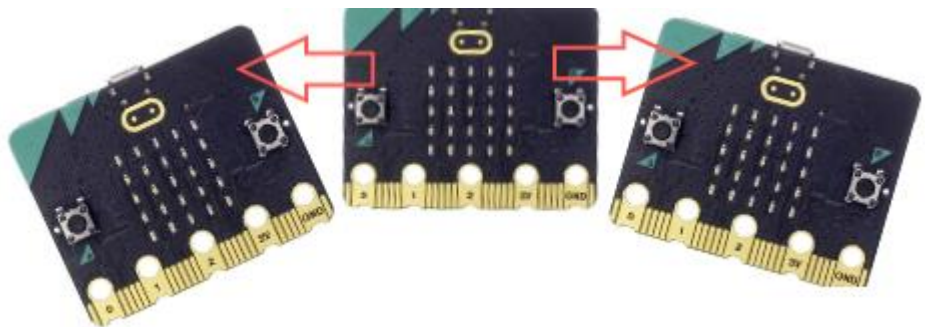
We willen de waarde van de kantelhoek gaan gebruiken. Maak daarom eerste een variabele aan met de naam "kantelHoek". Zorg dat het vinkje ervoor actief is zodat we deze waarde op het scherm zien.

Voeg de code hier rechts toe aan de kat.



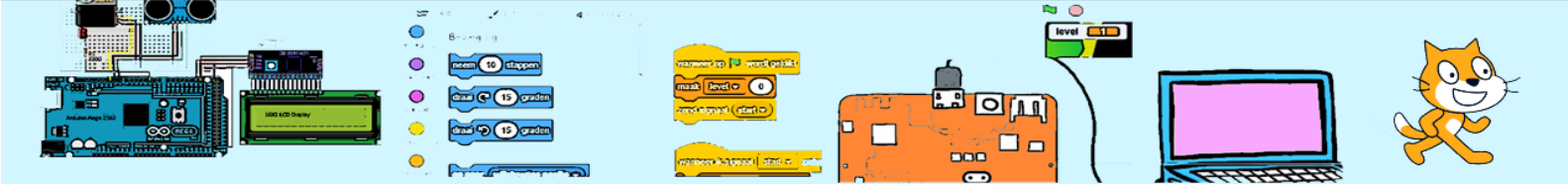
Draai het stuurkje (de Micro:Bit) nu langzaam naar links en naar rechts.

Kijk gelijktijdig naar de variabele en de kat.



Kantel de Micro:Bit nu ook langzaam voor en achterover.

We zien dat wanneer de Micro:Bit voorover wordt gehouden dit de kantelhoek beïnvloed. We moeten het stuurkje dus niet vooroverhouden.



3 Het startbestand

Download het startbestand ScratchAutoBegin.sb3.

<https://coderclub.nl/dl/ScratchAutoBegin.sb3>

Hierin bevinden zich alle benodigde uiterlijkheden van de gebruikte sprites.



Van enkele sprites is ook gedeeltelijk de code al toegevoegd. De sprites met code zijn:

Huis1, School, autoStoplichtOost, autoStoplichtWest, voetgangersstoplichtNoord, VoetgangersStoplichtZuid, en voetgangersDetectie, grasveld1, grasveld2 en "GameOver".

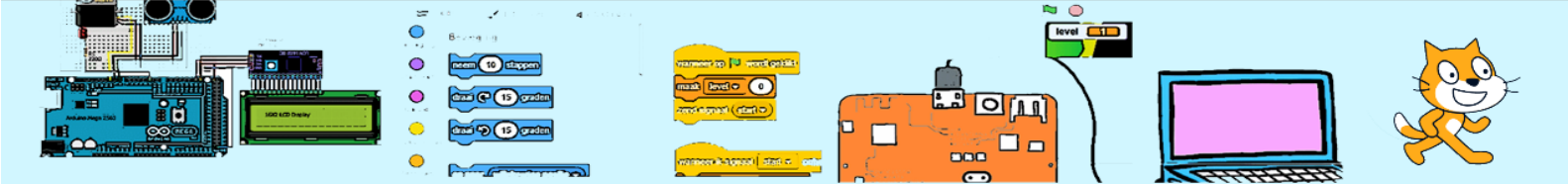
De eerste 6 sprites versturen het bericht "Aanrijding" wanneer de auto hen raakt. Het spel moet dan afgelopen zijn.

De grasvelden verlagen de score met 1 en het spel gaat door.

De stoplichten reageren al op berichten om van kleur te veranderen.

De voetgangerDetectie-sensor verhoogt de variabele "aantalVoetgangers" wanneer Tatiana deze sprite raakt.

Bekijk de uiterlijkheden en codes van alle genoemde sprites om de opdracht beter te begrijpen.



4 De auto

We gaan de snelheid van de auto bijhouden in een variabele **snelheid**. In een herhaalloop laten we de auto steeds de inhoud van **snelheid** stappen nemen. De snelheid van de auto op het scherm is dus evenredig met de inhoud van **snelheid** en hoe vaak de herhaalloop doorlopen wordt in een seconden.

De draaisnelheid van de auto is evenredig met de kantelhoek van het stuur. Als een auto stil staat dan verandert hij niet van richting. Hoe sneller een auto rijdt hoe sneller zijn richting kan veranderen. In de code hiernaast wordt dit benaderd met de formule draaiing = **kantelhoek** * wortel van de **snelheid** * 0.02. Die vermenigvuldiging met 0.02 is nodig omdat de kantelhoek snel tientallen hoog is en de herhaalloop ruim 10 maal per seconden wordt doorlopen. Je kan zelf experimenteren met een verandering van deze formule.

```
wanneer op vlag wordt geklikt
zet klok op 0
verander uiterlijk naar heel
ga naar laag voorgrond
maak snelheid 0
maak score 0
verander uiterlijk naar heel
richt naar 0 graden
ga naar x: -186 y: -90
verschijn
herhaal
draai kantelhoek rechts * wortel van snelheid * 0.02 graden
als knop A ingedrukt? dan
  Versnel
als knop B ingedrukt? dan
  Rem
neem snelheid stappen
keer om aan de rand
wacht 0.08 sec.
```

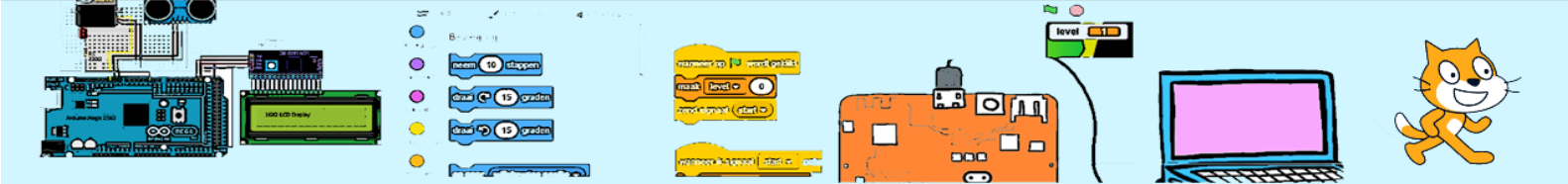
Je moet bij de auto-sprite zelf 4 blokken code maken. Twee daarvan zijn de “eigen blokken” **versne** en **rem**.

In het blok **versnel** moet de snelheid worden verhoogd met 0.5. Maar de snelheid mag niet hoger worden dan 5.

Het blok **rem** is ingewikkelder. Als de snelheid gelijk aan 0 is dan is de auto gestopt en dan moet de auto 1 stap achterwaarts nemen. En anders als de $0 < \text{snelheid} < 0.4$ is dan moet de snelheid 0 worden gemaakt en als de $\text{snelheid} > 0.4$ dan moet de snelheid met 0.4 worden verminderd. Je mag later na het spelen van het spel andere waarden kiezen.

Als de B-button 2 maal binnen 0.2 seconden ingedrukt (paniek stop) blijft dan moet de snelheid 0 worden gemaakt. Als op dat moment de snelheid hoger is dan 1 dan moet het “skid”-geluid (piepende banden) hoorbaar zijn. Tip: gebruik **klok**

Het **derde blok** moet starten na drukken op de groene vlag. Dit blok moet eenmaal per seconden de snelheid van de auto verminderen. Dit simuleert dat wanneer je geen gas geeft in een auto de snelheid langzaam afneemt. Hiervoor



gebruiken we de formule $\text{snelheid} = \text{snelheid} * 0.95$. Als de snelheid al 0 is dan hoeft er niets gebeuren. Als de snelheid < 0.5 dan moet de snelheid op 0 worden gezet.

Het **vierde blok** moet wanneer het bericht **Aanrijding** wordt ontvangen de auto stoppen en het bericht **GameOver** versturen.

5 De zebra

De zebra moet de score bijhouden. Iedere keer wanneer de auto passeert bij groen licht verdient de speler 5 punten. Als dit gebeurt bij rood licht dan wordt de score op 0 gezet. Daarnaast houdt de zebra bij of de zebra vrij is van de auto via de variabele **zebraVrij**. Zodra de auto de zebra raakt is deze niet meer vrij (onwaar) totdat de auto de zebra heeft verlaten (`richt tot` `richt` `raak &` `auto = ?`). Daarna is de zebra weer vrij (waar). De sprite Tatiana kan met deze variabele kijken of de weg vrij is om over te steken. Vergeet niet de variabelen **score** en **zebraVrij** te initialiseren.

6 De verkeerslicht regelenheid

Op een kruispunt met verkeerslichten staat gewoonlijk een licht grijs kastje dat de stoplichten aanstuurt. De verkeerslicht regelenheid is een geautomatiseerde verkeerslichtregelaar.



In deze opdracht hebben we ook een verkeerslicht regelenheid. De verkeerslichten worden bestuurd met events zoals **autoStoplichtNaarGroen**. De code voor de verkeerslichten is al gemaakt maar de code voor deze regelaar moet jij maken. Spiek bij de stopLichten-sprite voor welke events beschikbaar zijn om de verkeerslichten te sturen.

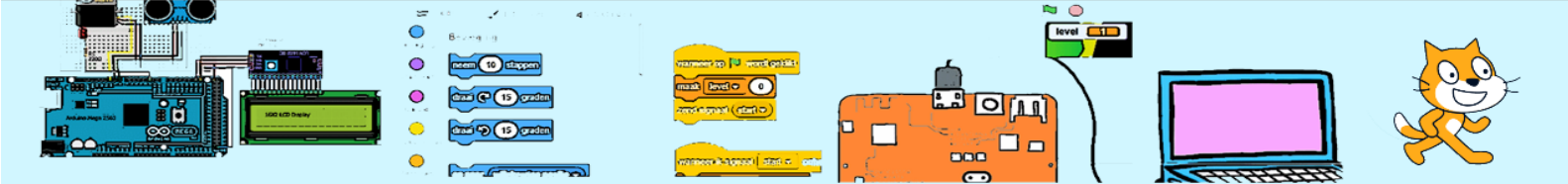
De verkeerslichtregelaar moet de variabele `aantalVoetgangers` uit lezen. Wanneer deze variabele hoger is dan 0 dan begint een cyclus. Eerst moeten de verkeerslichten voor de auto's op oranje springen. Na een korte periode (2 seconden) moeten deze rood worden en na weer een kleine periode kunnen de stoplichten voor de voetgangers op groen worden gezet.

Na ongeveer 6 seconden moet alles weer worden omgedraaid. Maar Tatiana moet genoeg tijd hebben om over te steken ook wanneer zij nog net groen licht heeft gehaald.

Wanneer de het voetgangersstop licht naar rood gaat moet het `aantalVoetgangers` op 0 worden gezet,

7 Tatiana

Tatiana is een voetganger die met onregelmatig tussenpozen via het zebra de weg oversteekt. Er kunnen meerdere Tatiana's zijn. Er moet dus een herhalloop komen die op onregelmatige tijden (tussen ~15 en ~25 seconden) een kloon van Tatiana maakt. Het is de kloon die gaat lopen.



Tatiana verlaat de school en loopt dan naar boven tot zij de voetgangerDetectie-sprite raakt. Daar wacht zij tot het **voetgangersStoplicht** groen is **en** het zebrapad vrij is. Daarna steekt zij over en loopt tot de rand van het scherm en verdwijnt.

De sprite Tatiana heeft 4 uiterlijkheden. Verzorg dat wanneer Tatiana loopt deze uiterlijkheden afwisselend worden gebruikt zodat het lijkt alsof zij loopt. Wanneer Tatiana stil staat dan moet zij het bijpassende uiterlijk hebben. Zorg er voor dat Tatiana loopt en niet rent. Een juist tempo is een herhalloop waarin Tatiana iedere 0.1 seconden 3 stappen zet en dan van uiterlijk verwisseld.

Wanneer Tatiana de rand raakt moet de kloon worden opgeruimd.

Wanneer de auto Tatiana raakt dan moet het bericht **Aanrijding** worden verstuurd.

Tatiana moet bijna in de voorgrond laag komen. Zij moet in een laag zitten die lager is dan de school maar boven alle andere sprites. Zij loopt bijvoorbeeld over het zebrapad maar verschijnt van onder het dak van de school. Wanneer iedere sprite direct na het klikken naar de voorgrond gaat zoals hiernaast gecodeerd. Dan weten we niet welke sprite op de voorgrond eindigt. Verzin een systeem zodat je wel zeker weet dat de school in de bovenste lagen komen en dat Tatiana in de 2^{de} laag vanaf boven komt.

