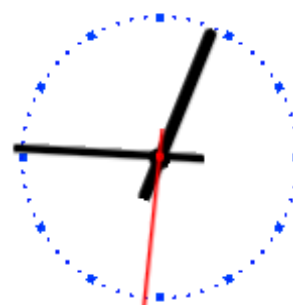


Voorkennis		Niveau	
Benodigdheden	Scratch	Leerdoelen	• Systeemtijd opvragen • Werken met variabele • Tussen verschillende sprites signalen sturen • Computer is snel, maar code regels kosten computertijd

Een klok met wijzers



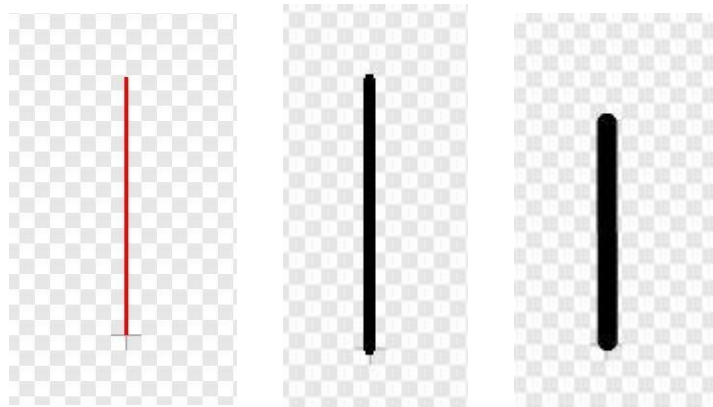
We gaan een klok maken die er ongeveer uit ziet zoals hier rechts.

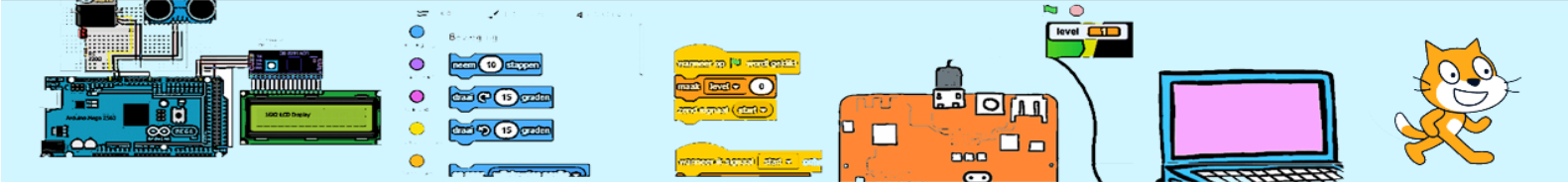
1. De wijzers

Er zijn vele manieren om de klok te maken. Hier laten we 1 manier zien. We gaan straks 3 sprites maken met de klokwijzers en laten deze sprites draaien. Om de wijzers van de klok netjes om het midden van de klok te laten draaien, moeten we de wijzers ook in het midden van de sprite laten beginnen. We beginnen met de seconde wijzer. Teken een nieuwe sprite en geef die de naam: "seconde". Rechtsonder het tekenveld zit een +vergrootglas. Hiermee kan je inzoomen en zie je het kruisje in het midden van de sprite.

Teken nu de secondewijzer in het rood recht omhoog vanaf het kruisje.

Maak een nieuwe sprite genaamd "minuut" en maak de zwarte minutenwijzer iets dikker en even lang is als de secondewijzer. Als laatste maak je een nieuwe sprite genaamd "uur" en teken je een nog dikkere en iets kortere urenwijzer.





2. Wijzerplaat

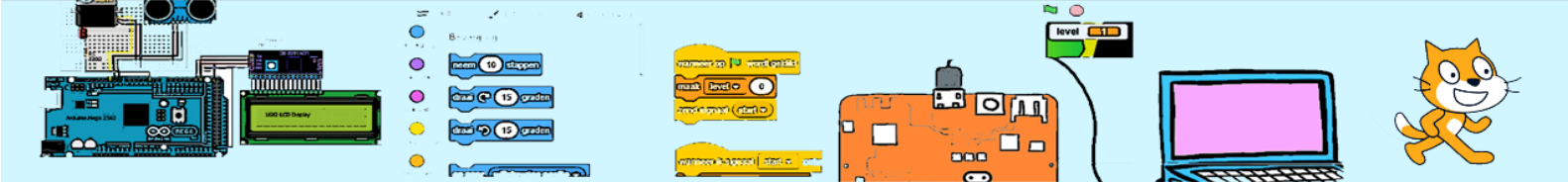
We kunnen op veel manieren de wijzerplaat maken. Hier kiezen we ervoor om de wijzerplaat door de computer te laten tekenen. We gaan een sprite genaamd “WP-minuten” maken met daarin 1 kleine blauw punt. De blauwe punt moet rechtsboven het middelpunt van de sprite “WP-minuten” komen en de afstand tot het middelpunt ongeveer als de lengte van de secondewijzer.

We gaan nu programmeren dat de computer de wijzerplaat gaat tekenen.

Een uur heeft 60 minuten, dus moeten we 60 punten hebben. De sprite “WP-minuten” heeft maar 1 punt. We hebben er dus nog 59 extra nodig. We drukken de sprite WP-minuten af en draaien de sprite WP-minuten daarna naar de volgende positie. Een hele cirkel is 360 graden. Dus de volgende positie is $360 \text{ graden} / 60 \text{ minuten} = 6 \text{ graden/ minuut}$ verder.



Vraag: wat gebeurt er als we eerst draaien en daarna pas afdrucken?



We willen niet alleen minuten-streepjes, maar willen ook iets dikkere uren-streepjes. We kopiëren de sprite “WP-minuten” en noem de nieuwe sprite “WP-uren” en maak de blauwe punt dikker. Bereken nu zelf hoeveel dikke punten ontbreken en hoeveel graden er gedraaid moet worden naar de volgende uur positie. Eerst zelf proberen alvorens naar de code hiernaast te kijken.



Is de wijzerplaat goed getekend? Wat gaat er mis?

Wijzerplaat Coding verbetering

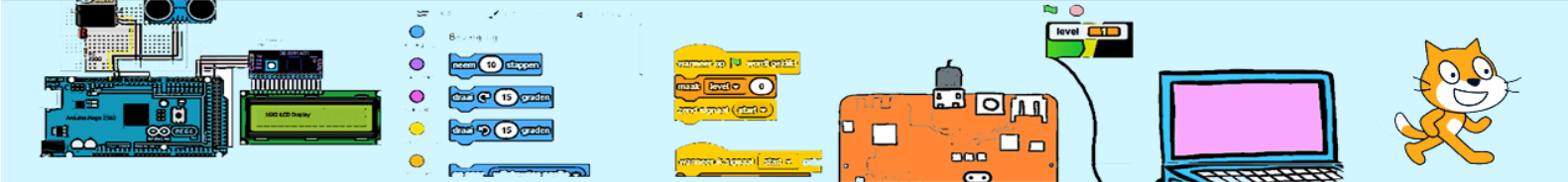
Heb je gezien dat de code “wis alles” de eerst getekende uur markering heeft weggehaald? Om de code regels van verschillende sprites op elkaar af te stemmen laten we de code behorende bij de sprite “WP-uren” pas beginnen als de opdracht “wis alles” bij de sprite “WP minuten” klaar is. Aan het einde van de code “WP minuten” gaan we een signaal zenden. En we laten de code behorende bij “WP-uren” wachten op dat signaal.

Coding voor de Sprite “WP-minuut”



Coding voor de sprite “WP-uren”





3. Secondewijzer

We kunnen aan de computer vragen hoe laat het is. We kunnen vragen wat de huidige seconde, huidige minuut of het huidige uur is.

Om de secondewijzer in de juiste beginstand te draaien, moeten we eerst weten welke seconde het nu is. De secondewijzer is in 60 seconden rondgedraaid en dat is 360 graden. Dus per seconde draait de secondewijzer $360 / 60 = 6$ graden. De beginstand is dus "huidige seconde" * 6 graden gedraaid.

Daarna doen we de regel: (we wachten 1 seconde) en dan de regel 2: (we draaien de sprite "seconde" 6 graden) en dan de regel 3: (wachten we 1 seconde) en dan de regel 4: (draaien de sprite "seconde" 6 graden) en dan de regel 5: enzovoorts.

Het is veel makkelijker om tegen de computer te zeggen dat hij de regels 1 en 2 eindeloos moet herhalen.

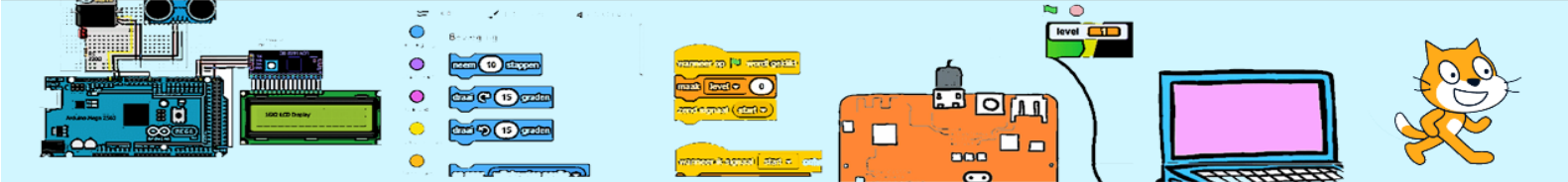


4. Minutenwijzer

Net als bij de secondewijzer moeten we ook bij de minutenwijzer de beginstand bepalen. We willen de minutenwijzer in kleine stappen rond laten draaien. We kiezen nu dat de minutenwijzer iedere 10 seconde een klein stapje draait. Daarom is de beginstand van de minutenwijzer niet alleen afhankelijk van de "huidige minuut" maar ook van de "huidige seconde". We gaan een nieuwe variabele maken met de naam "positie minuut". Deze variabele gaat ons helpen in de berekening van de positie van de minutenwijzer.

Advies. Het computerprogramma wordt makkelijker leesbaar (zeker later als je nog eens iets wil aanpassen) als er per regel maar 1 berekening wordt uitgevoerd. Verder is het verstandig commentaar toe te voegen waarin je aangeeft waarom de regel nodig is.





5. Urenwijzer

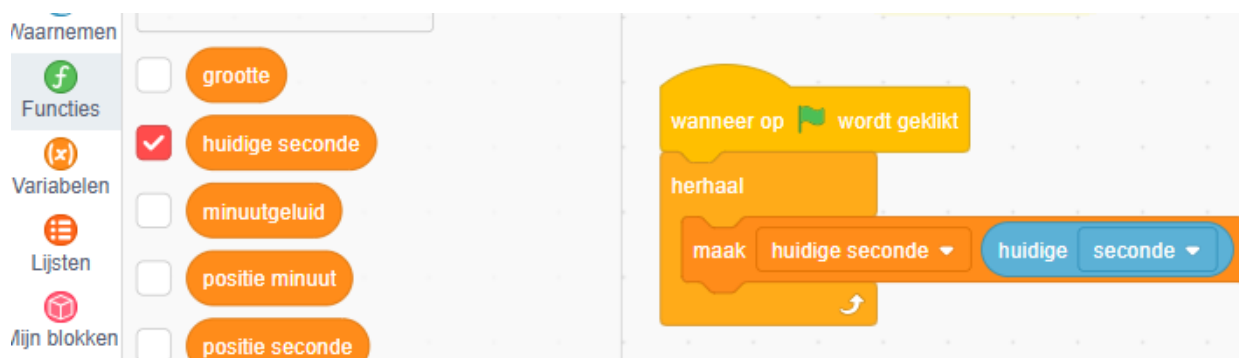
Voor de urenwijzer volgen we hetzelfde manier als met de minuten. We kopiëren de code van de sprite “minuut” naar de sprite “uur”. Moeten we nog wel een nieuwe variabele maken genaamd “positie uur” en de gekopieerde code een beetje aanpassen. We laten de urenwijzer iedere 2 minuten steeds 1 graad draaien.



Zet de klok aan. Ziet het er goed uit?

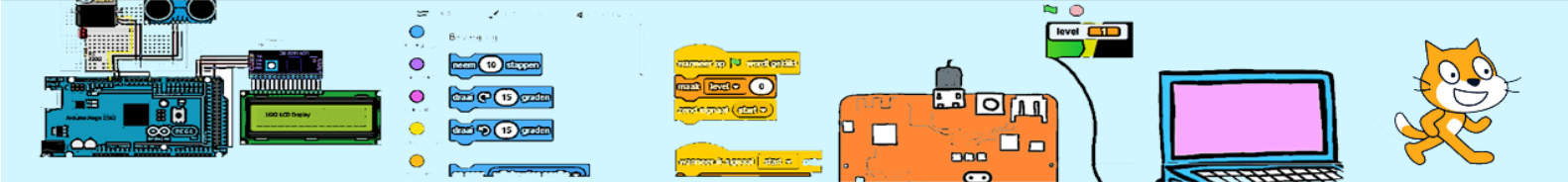
6. Controle of de klok goed loopt

We gaan controleren of de klok goed loopt. Dat doen we door een nieuwe variabele “huidige seconde” te maken en de volgende code regels toe te voegen bij de sprite “seconde”



We zien dat de seconde wijzer na verloop van tijd achterloopt dan de wat de variabele “huidige seconde” aangeeft.

Vraag: Hoe kan dat?



7. Verbeterde Klok 1

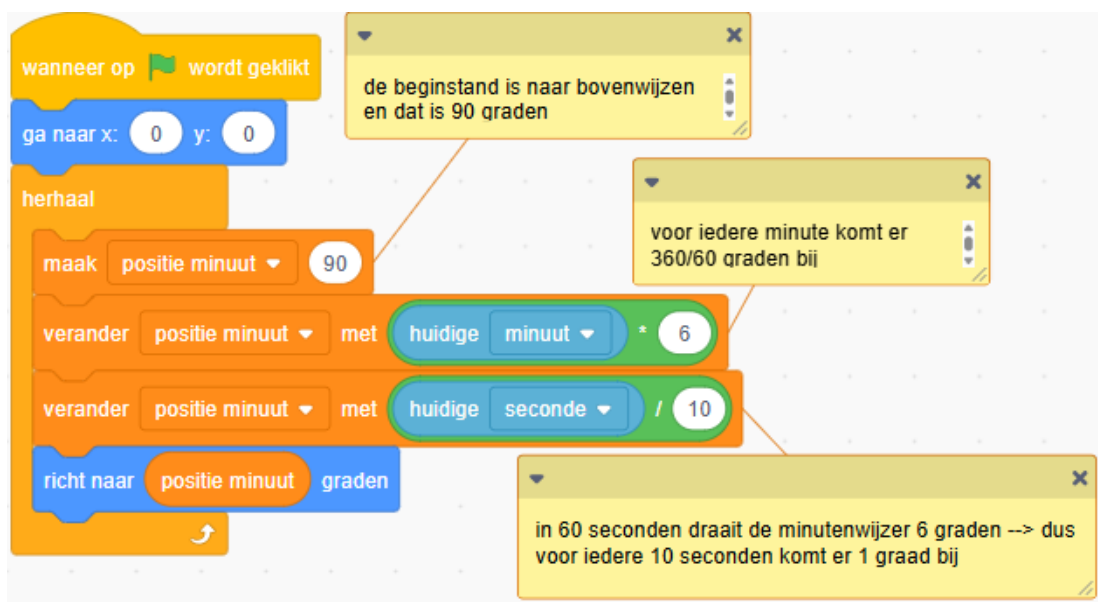
We gaan nu niet meer 1 seconde wachten en dan 6 graden draaien, maar we vragen continu de tijd aan de computer en berekenen iedere keer de nieuwe positie van de secondewijzer. En omdat we maar 1 berekening per regel willen uitvoeren, maken we een nieuwe variabele genaamd "positie seconde" en veranderen de regels als volgt:

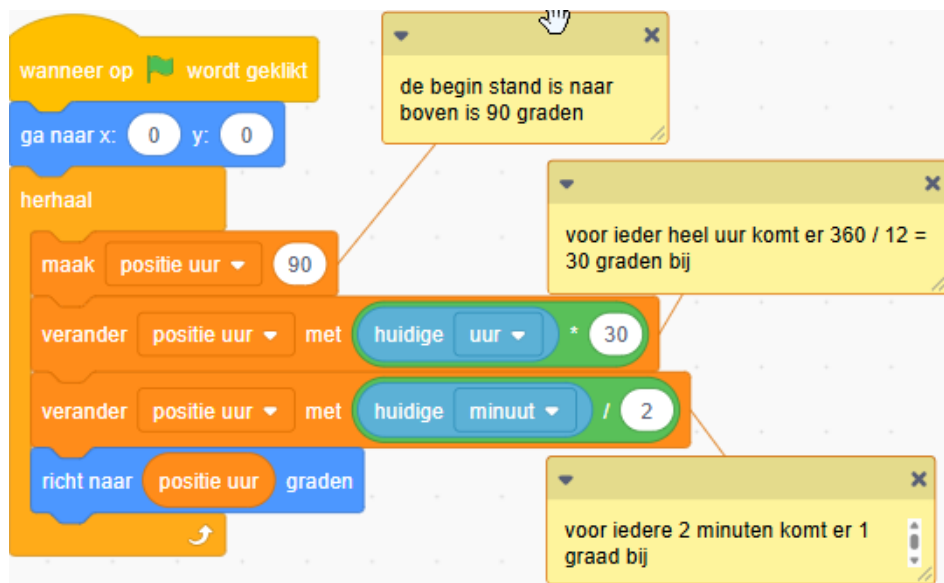
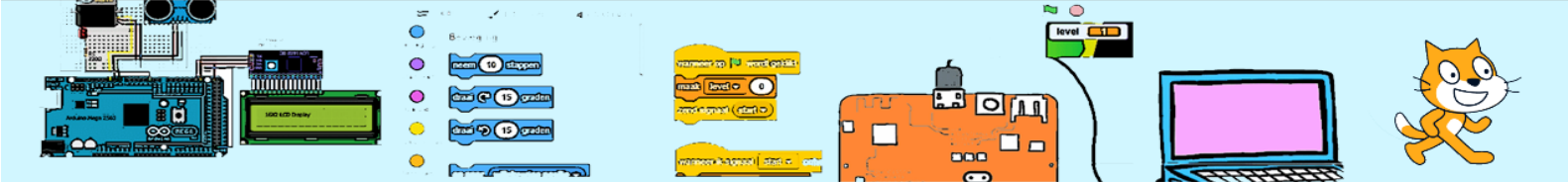


Vraag: loopt de secondewijzer nog steeds achter?

8. Verbeterde Klok 2

Heb je ook aan de minuten- en urenwijzer gedacht? Die zullen ook achter gaan lopen. Dat zal wel veel langer duren voordat je dat ziet. De aangepaste regels zien er als volgt uit:





Je hebt nu geleerd:

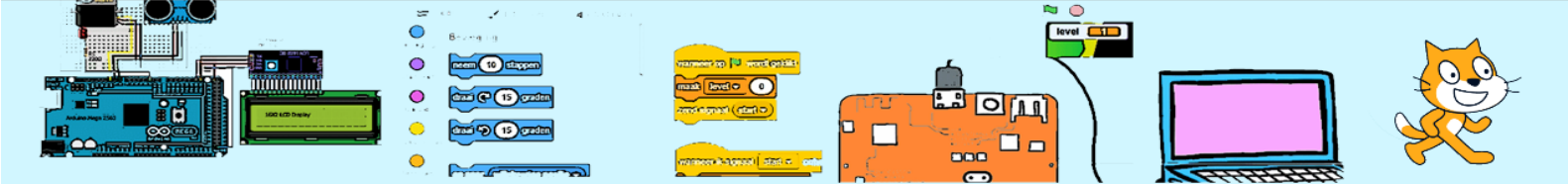
- dat de computer snel is maar het kost tijd om code regels uit te voeren,
- je variabele kan maken en
- je kan nu signalen sturen tussen verschillende sprites.

9. Iets leuk toevoegen

Nu gaan we nog iets leuk toevoegen, bijvoorbeeld: we laten de tovenaar (Wizard) de wijzers op de grond laten vallen.

Maak een nieuwe sprite Toverenaar (Wizard) aan en voeg een tweede uiterlijk toe met dezelfde Toverenaar. Van de tweede tovenaar moet je z'n arm met de toverstaf een beetje omhoog draaien. Selecteer eerst zijn arm, beetje draaien en een beetje omhoog zodat zijn arm weer op de juiste plaats zit.





We willen de Toveraar van heel klein langzaam groot laten worden en dan zijn arm bewegen. Om daarna de wijzers op de grond te laten vallen, sturen we een signaal naar de andere sprites. Verder kan jij je fantasie gebruiken om nog maar te laten gebeuren, bijvoorbeeld geluid toe te voegen. En code om de wijzers weer terug op hun plaats te brengen.

Code voor de sprite “Tovenaar”

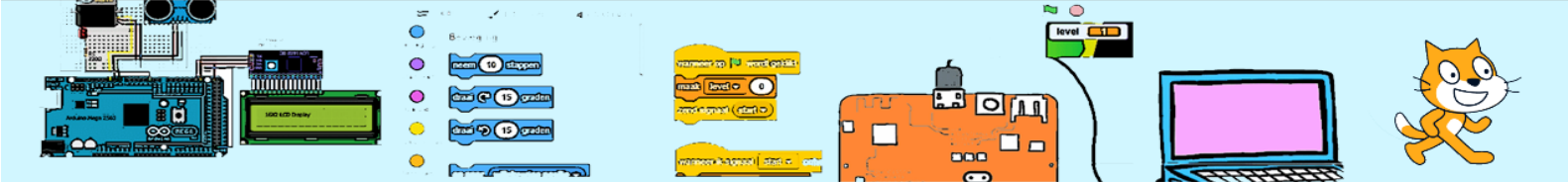
```

    wanneer op vlag wordt geklikt
    verdwijnt
    maak grootte 1 %
    zet video uit

    wanneer spatiebalk is ingedrukt
    verander uiterlijk naar wizard
    verschijnt
    zet volume op 100 %
    herhaal 100
    verander grootte met 2
    wacht 0.2 sec.
    verander uiterlijk naar wizard2
    wacht 0.2 sec.
    verander uiterlijk naar wizard
    zend signaal val op de grond
    speel drum (4) Crashbekken gedurende 0.25 maten

    wanneer pijltje omhoog is ingedrukt
    maak instrument (11) Saxofoon
    zeg Hallo mama ik ga de klok maken 2 sec.
    herhaal 100
    verander grootte met -2
    zend signaal weer terug
    verdwijnt
    herhaal 4
    speel noot 60 gedurende 1 maten
    verander volume met -20
    zet video aan
    maak video helderheid 90
    herhaal 17
    zet video transparantie op video helderheid
    verander video helderheid met -4
    wacht 0.3 sec.

    wanneer pijltje omlaag is ingedrukt
    zet video uit
  
```

Code toevoegingen voor de sprites

“seconde”

```

    wanneer ik signaal val op de grond ontvang
    verberg variabele huidige seconde
    schuif in 1 sec. naar x: 0 y: -170

    wanneer ik signaal weer terug ontvang
    verberg variabele huidige seconde
    schuif in 1 sec. naar x: 0 y: 0
  
```

“minuut”

```

    wanneer ik signaal val op de grond ontvang
    schuif in 1 sec. naar x: -22 y: -150

    wanneer ik signaal weer terug ontvang
    schuif in 2 sec. naar x: 0 y: 0
  
```

“uur”

```

    wanneer ik signaal val op de grond ontvang
    schuif in 1 sec. naar x: 40 y: -160

    wanneer ik signaal weer terug ontvang
    schuif in 4 sec. naar x: 0 y: 0
    zeg Let op!!! 2 sec.
  
```