

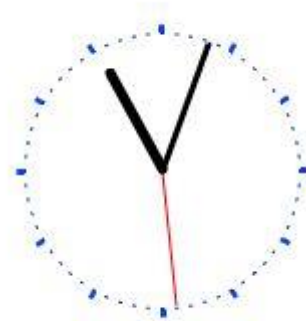
Een klok met wijzers

Voorkennis: Rekenen en graden

Leerdoel:

- Systemen tijd opvragen
- Werken met variabele
- Tussen verschillende sprites signalen sturen
- computer is snel, maar code regels kosten computertijd

wijzers

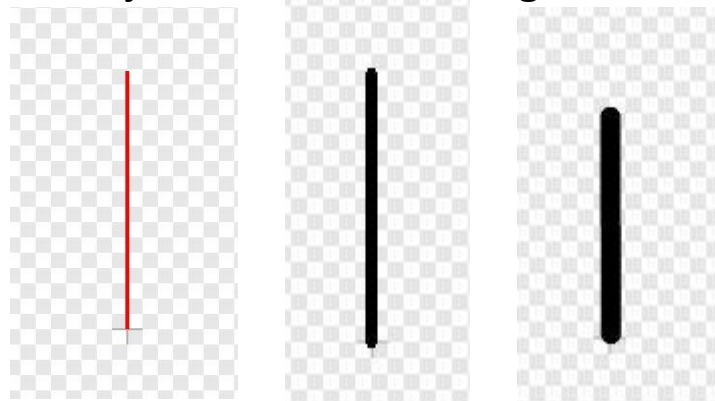


We gaan een klok maken die er ongeveer zo uit ziet

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.

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 spite genaamd "uur" en teken je een nog dikkere en iets kortere urenwijzer.



Wijzerplaat

We kunnen op veel manieren de wijzerplaat maken. Hier kiezen we er voor om de wijzerplaat door de computer te laten tekenen. We gaan een sprite genaamd "Wpminuten" maken met daarin 1 kleine blauw punt. De blauwe punt moet recht boven het middelpunt van de sprite "Wpminuten" 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 "Wpminuten" heeft maar 1 punt. We hebben er dus nog 59 extra nodig. We drukken de sprite WPminuten af en draaien de sprite WPminuten daarna naar de volgende positie. Een hele cirkel is 360 graden. Dus de volgende positie is $360 / 60 = 6$ graden verder.

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

We willen niet alleen minuten streepjes, maar willen ook dikkere uren streepjes. We kopiëren de sprite "Wpminuten" en noem de nieuwe sprite "Wpuren" 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.

Wijzerplaat codering

Codering voor WP minuut punten



Codering voor WP uur punten



Is de wijzerplaat goed getekend?

Wat gaat er mis?

Wijzerplaat Codering 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.

Codering voor de Sprite “WP minuut”



Codering voor de sprite “WP uren”



Secondenwijzer

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 geraaid.

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.

Secondewijzer codering



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.

Minutenwijzer codering



The image shows a Scratch script for a minute hand. It starts with a 'when green flag is clicked' event block. The first block is 'go to x: 0 y: 0'. The second block is 'set position of minute hand to 90', with a yellow callout box explaining: 'de beginstand is naar bovenwijzen en dat is 90 graden'. The third block is 'change position of minute hand by (current minute * 6)', with a yellow callout box explaining: 'voor iedere minuut komt er 360/60 graden bij'. The fourth block is 'change position of minute hand by (current second / 10)', with a yellow callout box explaining: 'in 60 seconden draait de minutenwijzer 6 graden -> dus voor iedere 10 seconden komt er 1 graad bij'. The fifth block is 'point to position of minute hand degrees'. The sixth block is a 'repeat' loop containing a 'wait 10 sec.' block and a 'rotate 1 degrees' block.

```
when green flag is clicked
  go to x: 0 y: 0
  make position of minute hand 90
  verander position of minute hand met huidige minuut * 6
  verander position of minute hand met huidige seconde / 10
  richt naar position of minute hand graden
  herhaal
    wacht 10 sec.
    draai 1 graden
```

de beginstand is naar bovenwijzen en dat is 90 graden

voor iedere minuut komt er 360/60 graden bij

in 60 seconden draait de minutenwijzer 6 graden -> dus voor iedere 10 seconden komt er 1 graad bij

Urenwijzer codering

Voor de urenwijzer volgen we het zelfde 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 uurwijzer steeds met 1 graad draaien.



The image shows a Scratch script for the hour hand. It starts with a 'wanneer vlag wordt aangeklikt' (when green flag clicked) event. The code then moves the sprite to x: 0, y: 0. A 'maak' (make) block sets the 'positie uur' (hour position) variable to 90, with a comment: 'de begin stand is naar boven is 90 graden'. Two 'verander' (change) blocks follow: the first changes 'positie uur' by 'huidige uur' * 30, with a comment 'voor ieder heel uur komt er 360 / 12 = 30 graden bij'; the second changes 'positie uur' by 'huidige minuut' / 2, with a comment 'voor iedere 2 minuten komt er 1 graad bij'. A 'richt naar' (point to) block sets the direction to 'positie uur' graden. A 'herhaal' (repeat) loop contains a 'wacht' (wait) block for 120 seconds, with a comment 'om 1 graad te draaien moet er 3600/30 seconden gewacht worden', and a 'draai' (turn) block for 1 graden.

Zet de klok aan. Ziet het er goed uit?

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 achter loopt dan de wat de variabele “huidige seconde” aangeeft.

Vraag: Hoe kan dat?

Verbeterde Klok 1

We gaan nu niet meer 1 seconde wachten en dan 6 graden draaien, maar we 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:

Codering voor de sprite “seconde”



Vraag: loopt de secondewijzer nog steeds achter?

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:

The image displays two Scratch code snippets for a clock face. The top snippet is for the minute hand, and the bottom snippet is for the hour hand. Both start with a 'when green flag is clicked' event.

Minute Hand Logic:

- Event: **wanneer vlag wordt aangeklikt**
- Block: **ga naar x: 0 y: 0**
- Loop: **herhaal**
 - Block: **maak positie minuut 90** (comment: de beginstand is naar bovenwijzen en dat is 90 graden)
 - Block: **verander positie minuut met huidige minuut * 6** (comment: voor iedere minute komt er 360/60 graden bij)
 - Block: **verander positie minuut met huidige seconde / 10** (comment: in 60 seconden draait de minutenwijzer 6 graden --> dus voor iedere 10 seconden komt er 1 graad bij)
 - Block: **richt naar positie minuut graden**

Hour Hand Logic:

- Event: **wanneer vlag wordt aangeklikt**
- Block: **ga naar x: 0 y: 0**
- Loop: **herhaal**
 - Block: **maak positie uur 90** (comment: de begin stand is naar boven is 90 graden)
 - Block: **verander positie uur met huidige uur * 30** (comment: voor ieder heel uur komt er 360 / 12 = 30 graden bij)
 - Block: **verander positie uur met huidige minuut / 2** (comment: voor iedere 2 minuten komt er 1 graad bij)
 - Block: **richt naar positie uur graden**

Iets leuk toevoegen 1

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

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.



Iets leuk toevoegen 2

We willen de Tovenaar 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”

The image shows two columns of Scratch code blocks for the 'Tovenaar' sprite. The left column contains the following blocks: 'wanneer vlag wordt aangeklikt', 'verdwijn', 'maak grootte 1 %', 'zet video uit', 'wanneer spatiebalk wordt ingedrukt', 'verander uiterlijk naar wizard', 'verschijn', 'zet volume op 100 %', 'herhaal 100 keer' (containing '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', and 'speel slagwerk 4 0.25 tellen'. The right column contains: 'wanneer pijltje omhoog wordt ingedrukt', 'gebruik instrument 11', 'zeg Hallo mama ik ga de klok maken 2 sec.', 'herhaal 100 keer' (containing 'verander grootte met -2'), 'zend signaal weer terug', 'verdwijn', 'herhaal 4 keer' (containing 'speel noot 60 1 tellen' and 'verander volume met -20'), 'maak video helderheid 90', 'herhaal 17 keer' (containing 'maak videodoorzichtigheid video helderheid %', 'verander video helderheid met -4', and 'wacht 0.3 sec.').

Iets leuk toevoegen 3

Code toevoegingen voor de sprites

“seconde”



“minuut”



“uur”

