

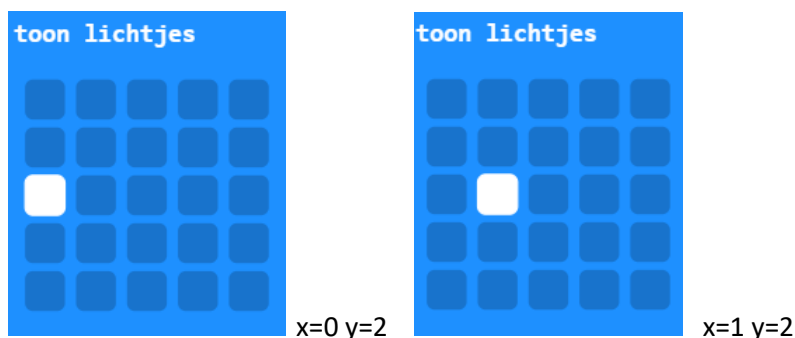
Voorkennis	Dobbelsteen	Niveau	
Benodigdheden	Micro:Bit	Leerdoelen	Parallele processen, variabelen

ReactieSpel

In dit spel “loopt” een led van links naar rechts en weer terug. De speler moet op de A-knop drukken precies wanneer de middelste led oplicht.

1. Het “lopende” ledje.

We beginnen met het laten bewegen van de led. Het ledje loopt van links naar rechts. We kiezen alleen de middelste rij te gebruiken. We maken eerst gebruik van de “toon lichtjes”-opdracht. Hieronder 2 opdrachten om de led te “verplaatsen” (De led verplaats zich niet maar één led gaat uit en een ander aan). Let op de x en y. Y blijft 2 maar de x verandert van 0 naar 4.



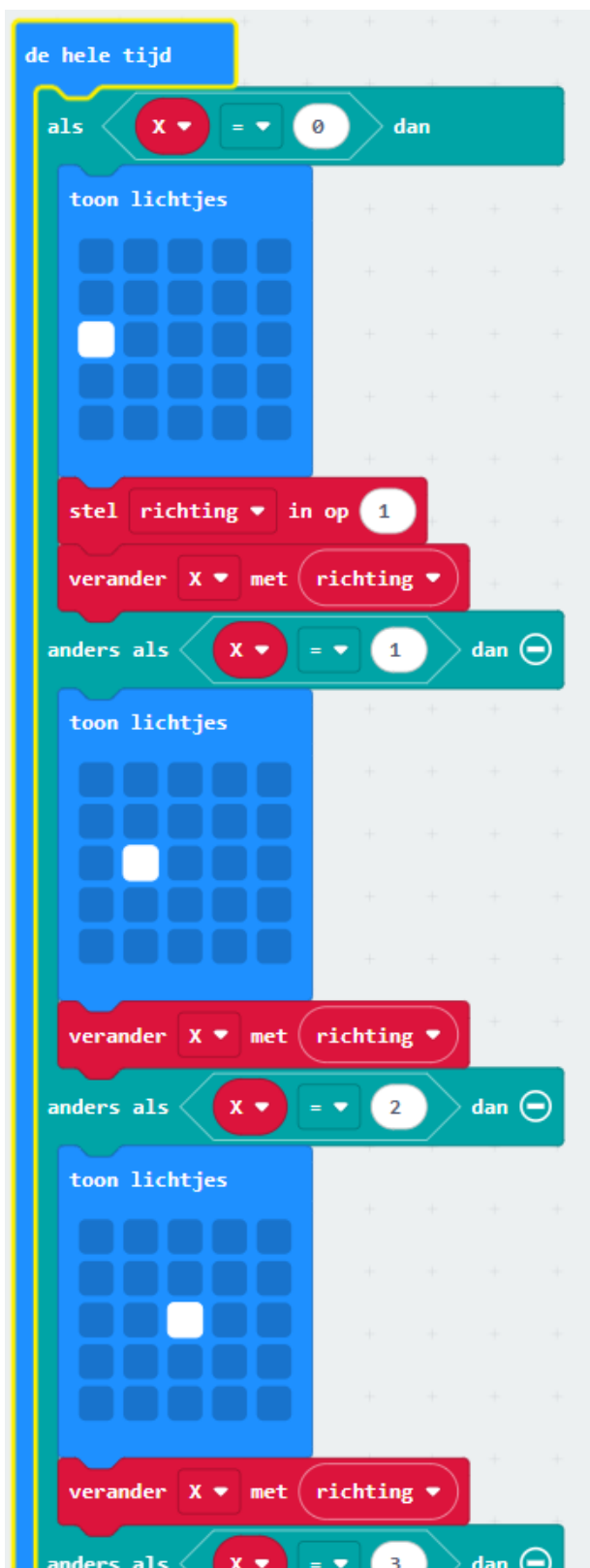
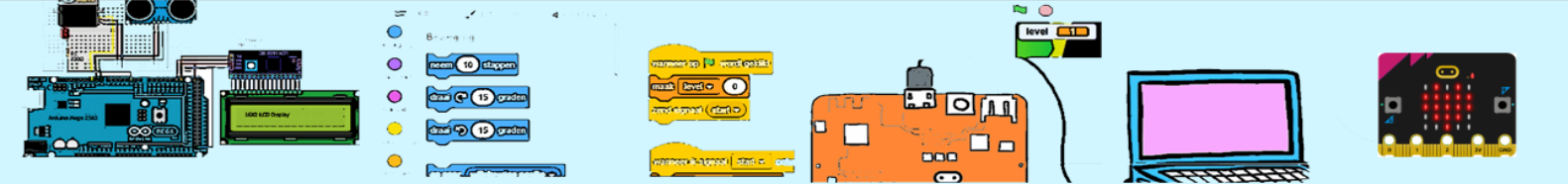
We maken eerst gebruik van de “toon lichtjes”-opdracht. (Later maken we een snellere variant.)

De led verplaats zich afwisselend naar rechts en naar links. We hebben een variabele “richting” nodig om bij te houden welke kant we op bewegen. Om met de richting te kunnen rekenen maken de richting naar rechts gelijk aan +1 en de richting naar links gelijk aan -1. We hebben dus 2 variabelen nodig te weten “X” (locatie van het ledje) en “richting”.

Deze variabelen moeten we initialiseren bij de start.

Neem de code hier rechts over.





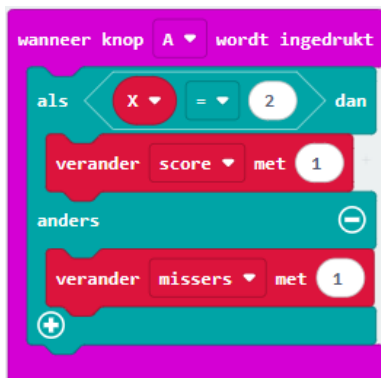
Hiernaast zien een groot gedeelte van het proces dat het ledje continu heen en weer laat bewegen. We zien dat de “toon lichtjes” -opdracht wordt uitgevoerd de hoort bij de X. D aarna wordt x aangepast voor de volgende “ronde”. De aanpassing is afhankelijk van de richting. Als x = 0 en als x = 4 wordt de richting om gedraaid.

Neem de code hiernaast over en vul de code aan voor x = 3 en 4. Bij x = 4 moet de richting weer veranderen zoals bij x=0 (maar de andere kant op).

Vergeet de initialisatie niet en test het resultaat. Als het goed is gaat het ledje “lopen” van links naar rechts en keert weerom.



2. Het Spel



We gaan een score bijhouden. Iedere keer wanneer de gebruiker op de A-knop drukt juist wanneer het middelste ledje aan krijgt hij een punt. Ook houden we bij hoe vaak de gebruiker mis drukt. Bij 3 misers is het spel afgelopen.

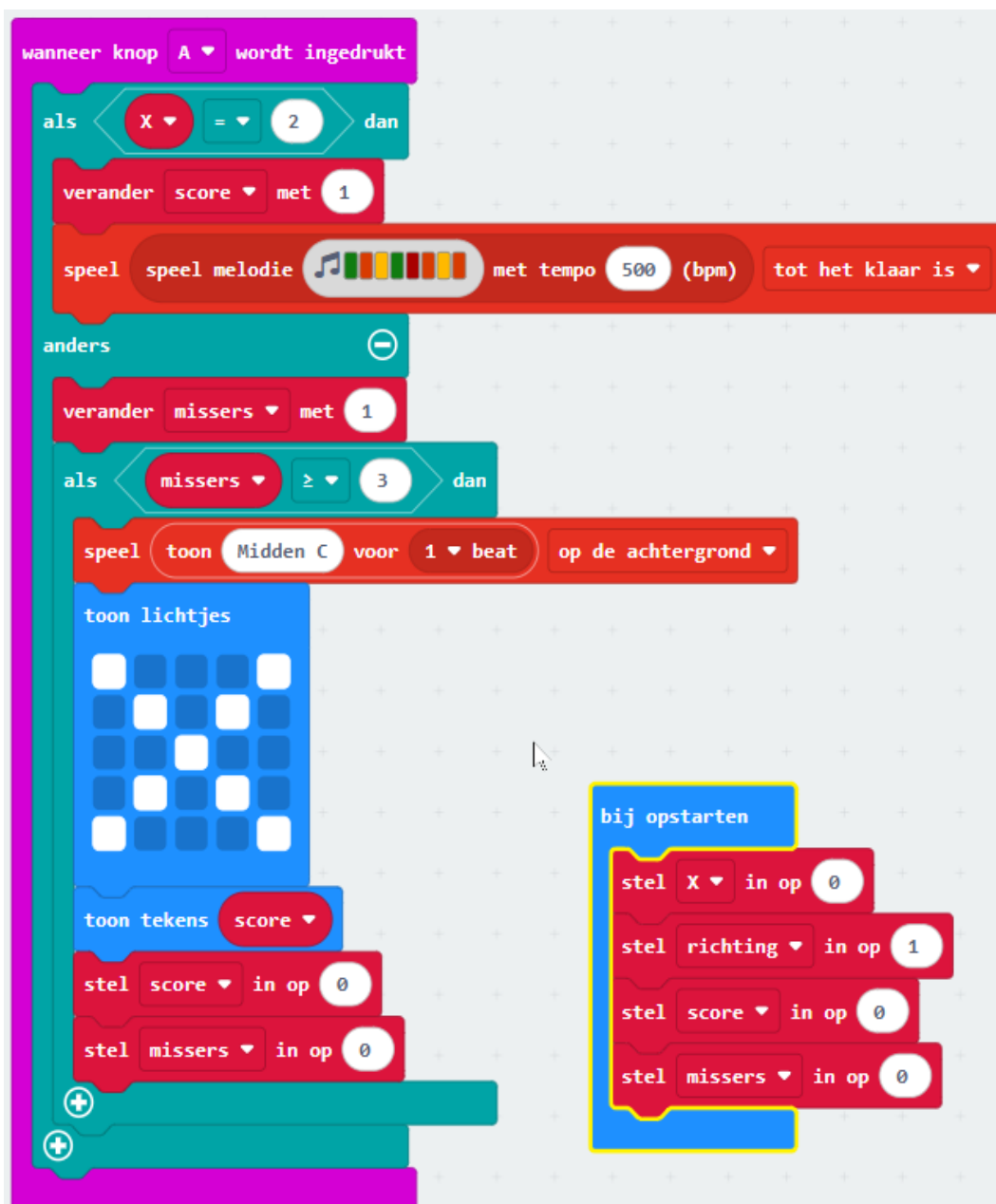
Hoe weten we of de gebruiker op het juiste moment klikt? Wanneer de A-knop wordt ingedrukt controleren we of de variabele X gelijk is aan 2. Zo ja dan verhogen we de score en zo dan verhogen we de missers. We moeten deze variabelen bij start initialiseren op 0.

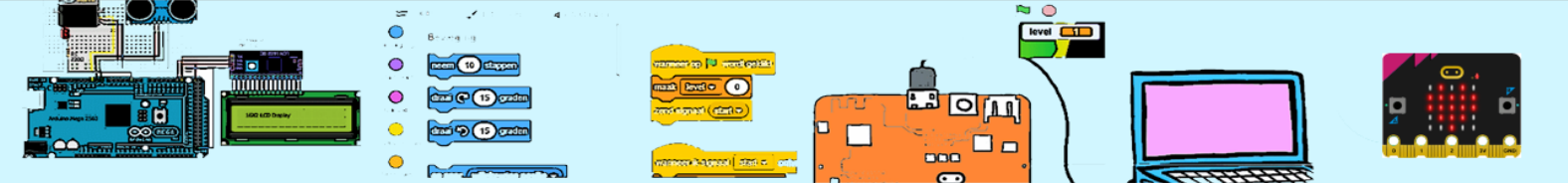
Dit gaat wel goed maar de gebruiker heeft geen idee wat zijn/haar score is. We moeten de code uitbreiden.

Laat een muziekje horen wanneer wordt gescoord.

En herstart het spel wanneer het aantal missers groter wordt dan 3.

We Laten dan ook nog even de score zien.



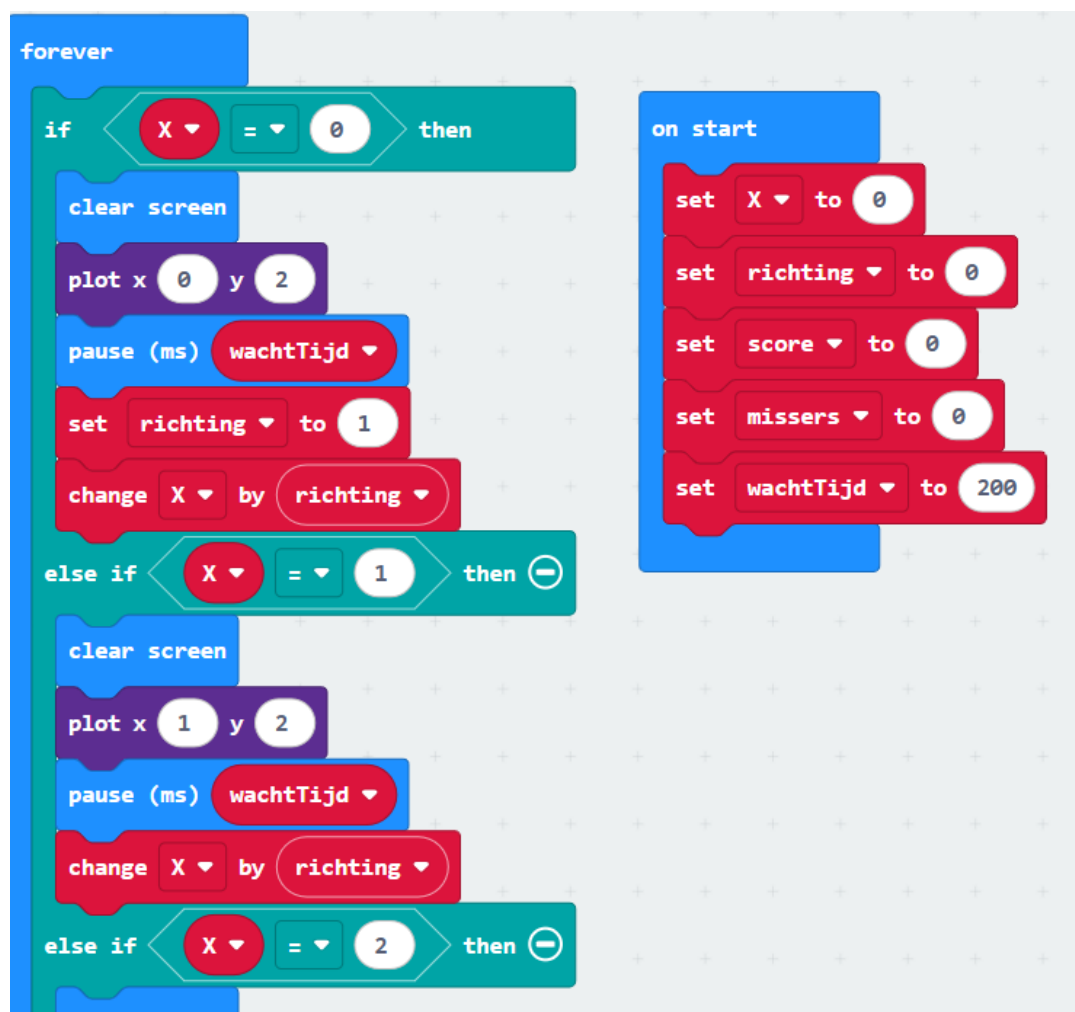


3. Verbetering

Het spel werkt maar al redelijk snel lukt het goed te scoren. Dit omdat het ledje redelijk traag beweegt. Dit terwijl we nog geen spelvertrager hebben ingebouwd. Hoe komt dit?

We hebben gebruik gemaakt van een opdracht die veel functionaliteit ondersteunt te weten “toon lichtjes”. Deze opdracht kan veel maar meestal neemt de snelheid van een opdracht af met de toename van functionaliteit.

Daarom gaan we een opdracht (plot x y) gebruiken die minder kan maar veel sneller is. We moeten wel steeds het scherm wissen tussen de verschillende X-waardes.



Daar deze opdracht snel is kunnen we een wachttijd gebruiken. Hoe hoger de wachttijd hoe lager de spelsnelheid. Hieronder de code. Neem de code over en vul aan voor alle X-en.

Dit werkt. Probeer de wachttijd wat korter te maken. Zie het resultaat?

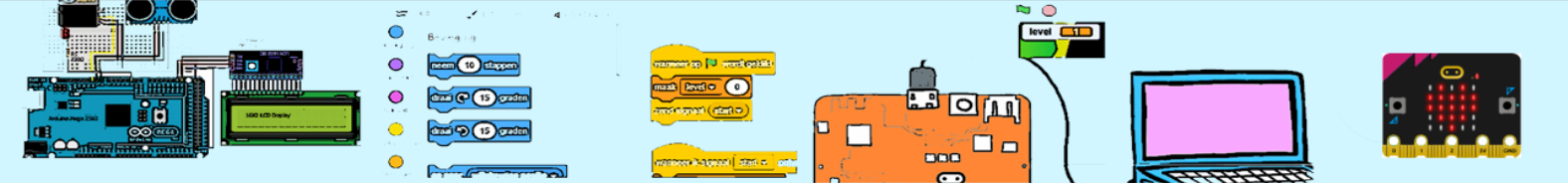
Maar wat gebeurt er op het moment dat de gebruik 3 missers heeft?

Het kruis en de score zijn heel even te zien en het ledje blijft “bewegen”. Hoe komt dit?

We hebben meerdere processen naast elkaar. Het proces dat het lopen van het ledje verzorgt blijft doorlopen onafhankelijk van het proces dat op de A-knop loopt. Hoe kunnen we dit oplossen?

We gaan een variabele gebruiken om de processen op elkaar af te stemmen. Hieronder de aanpassingen.

Het proces dat het ledje laat lopen is alleen actie wanneer de variabele “loop” gelijk is aan 1.



```

else
  change missers by 1
  if missers >= 3 then
    play tone Middle C for 1 beat in background
    show leds
    set loop to 0
    show string score
    pause (ms) 2000
    set loop to 1
    set score to 0
    set missers to 0
  +
  +

on start
  set X to 0
  set richting to 0
  set score to 0
  set missers to 0
  set wachtTijd to 200
  set loop to 1

de hele tijd
  als loop = 1 dan
    als X = 0 dan
      Wis scherm
      teken x 0 y 2
      pauzeer (ms) wachtTijd
      stel richting in op 1
      verander X met richting
    anders als X = 1 dan
      Wis scherm
  +

```

4. Uitdaging

Maak het zo dat het spel steeds sneller wordt naar mate de score hoger wordt.