

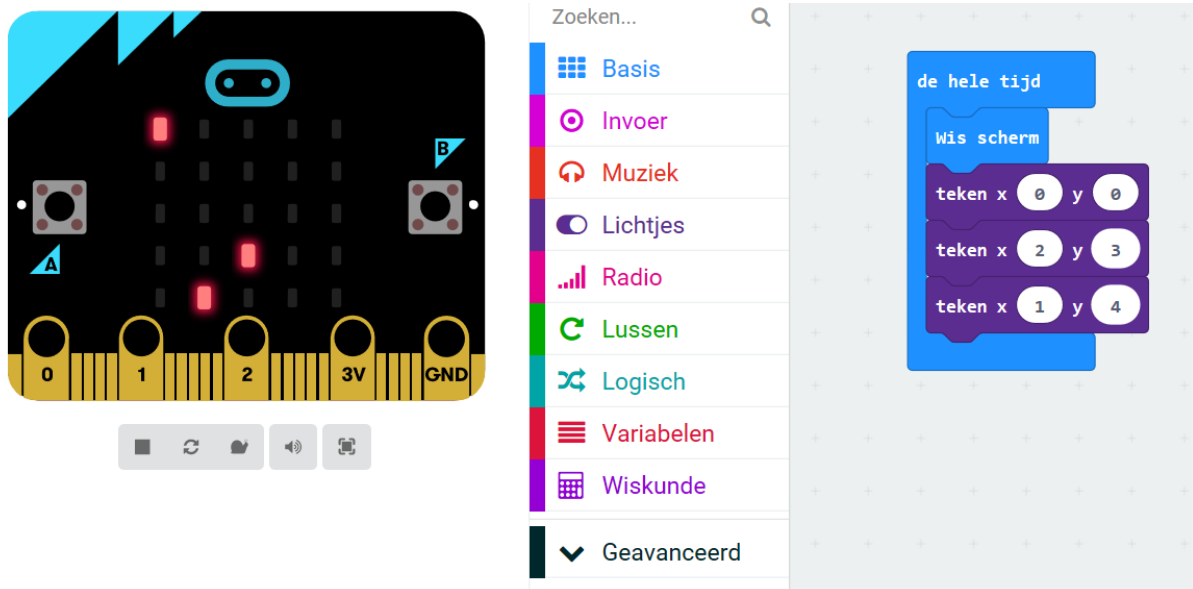
Space Invaders op de micro:bit

Op het eerste gezicht denk je misschien dat je met een scherm van vijf bij vijf ledjes niet zoveel kunt doen, maar dat valt best nog wel mee.

Om dat aan te tonen gaan we in deze handleiding Space Invaders programmeren op de micro:bit programma.

Begin met het aanmaken van een nieuw project en noem dat "Space Invaders".

Laten we beginnen met het aanzetten van een paar leds op het scherm. Voer daarvoor onderstaande code in. In de simulator links zie je dan drie leds oplichten.



Begrijp je het patroon van de brandende ledjes. Het ledje op het punt (0,0) bevindt zich helemaal links bovenin (eerste kolom en eerste rij). Het ledje op het punt (1,4) bevindt zich in de tweede kolom en de vijfde rij en is in het plaatje hierboven de onderste brandende led.

0,0	1,0	2,0	3,0	4,0
0,1	1,1	2,1	3,1	4,1
0,2	1,2	2,2	3,2	4,2
0,3	1,3	2,3	3,3	4,3
0,4	1,4	2,4	3,4	4,4

Het zou misschien duidelijker zijn geweest als de posities van (1,1) tot (5,5) zouden lopen maar jammer genoeg beginnen de meeste programmeertalen met het cijfer 0 voor een index (een index is een soort aanwijzer die aangeeft welke kolom of rij er bedoeld wordt).

Nu we weten hoe we ledjes kunnen laten branden kunnen we echt beginnen aan het spel. Voor de schutter gebruiken we een led op de onderste rij. De led moet zich horizontaal kunnen verplaatsen en verticaal een schot kunnen uitvoeren. De y-waarde van de schutter moet dus altijd 4 zijn (onderste rij), de x-waarde kan variëren van 0 tot 4 (eerste tot vijfde kolom).

Omdat de x-waarde kan variëren kunnen we daar het beste een variabele voor gebruiken.

Maak een nieuwe variabele en noem die “schutterX”.

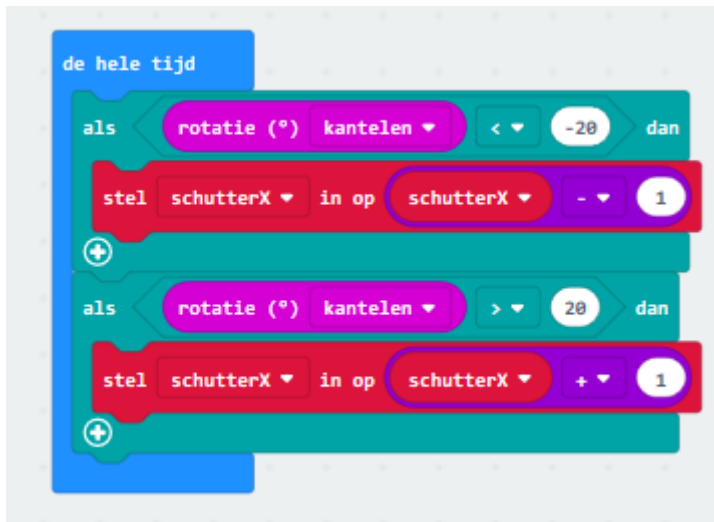
Gebruik daarna deze variabele om de schutter op het scherm te zetten.



De schutter doet nu nog helemaal niets maar laten we hem eens laten bewegen. We gaan de schutter laten bewegen door het laten kantelen van de micro:bit.

We moeten in onze code continu gaan checken wat de hellingshoek van de micro:bit is. Komt deze onder een bepaalde waarde dan moet het led een plaatsje naar links schuiven, komt hij boven een bepaalde waarde dan moet de led naar rechts schuiven.

Een eerste opzet ziet er als volgt uit:



Dit lijkt in eerste instantie de juiste code. Als we de onderkant van de micro:bit een beetje optillen veranderen we de variabele “schutterX” met -1 en als we de micro:bit wat naar onderen laten zakken doen we dat met +1.

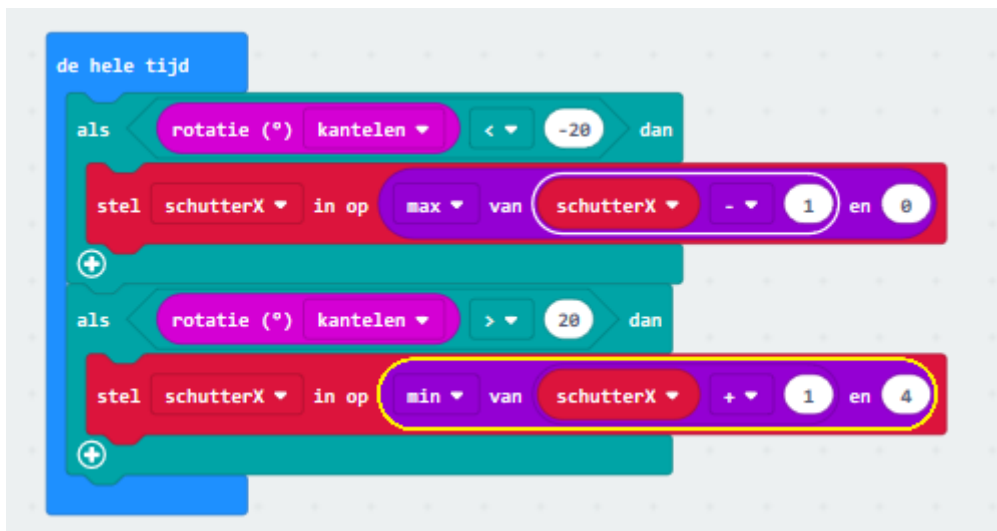
Als je dit in de simulator uitprobeert zul je zien dat het heel snel mis gaat. De led loopt eerst wel een stukje naar links of naar rechts maar daarna zien we ‘m niet meer. Dat komt omdat er eigenlijk geen stop op zit. De variabele “schutterX” begint bijvoorbeeld op 2 en als we de micro:bit dan wat naar onderen laten zakken wordt deze waarde eerst 3, dan 4, dan 5, dan 6, dan 7, etc. Het idee is goed maar de led mag nooit voorbij de waarde 4 komen, dat is immers de meest rechtse kolom.

Hetzelfde geldt ook voor de andere kant op. De waarde van “schutterX” mag wel kleiner worden maar nooit minder dan 0.

Dit probleem kunnen we oplossen door gebruik te maken van de functies “max” en “min”. Bij het naar links gaan kiezen we voor het grootste waarde van schutterX of nul. Is de waarde van schutterX groter dan nul, dan nemen we die waarde, is hij kleiner dan nul geworden, dan houden we nul aan (die is dan het grootste of het maximum van de twee).

Op dezelfde manier moeten we bij het naar rechts gaan kiezen voor het minimum van de waarde van schutterX en 4.

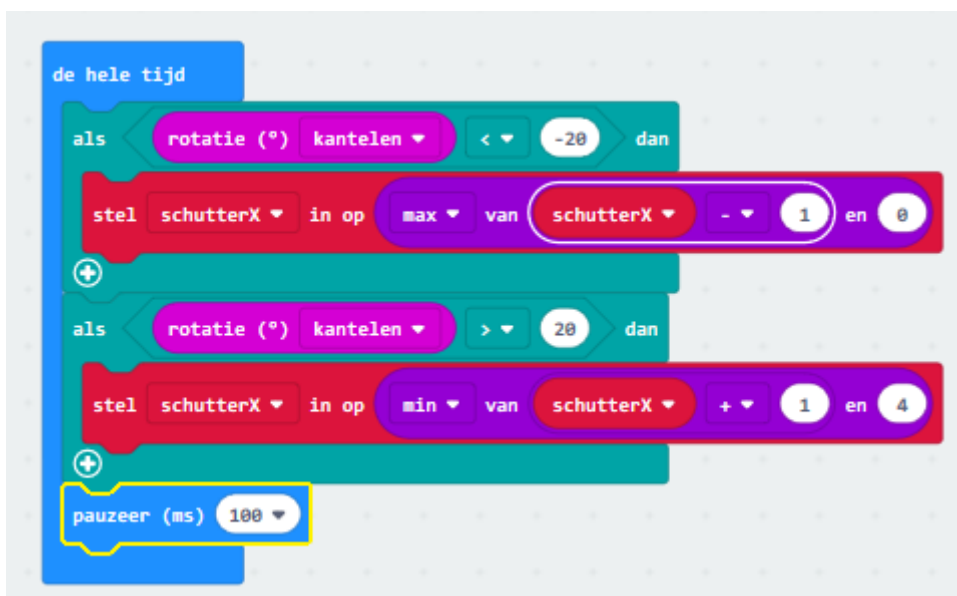
Pas de code aan:



Probeer het uit. Als het goed blijft kun je de led nu van links naar rechts en omgekeerd laten lopen en blijft deze ook altijd in beeld.

We kunnen de schutter nu aansturen maar eigenlijk beweegt hij veel te snel. We kunnen de boel wat vertragen door een pauze in te lassen.

Pas de code even aan:



Op dit moment start het spel zodra de micro:bit wordt aangezet en dat is misschien niet zo handig. We kunnen waarschijnlijk beter even wachten tot de speler er klaar voor is.

We laten de speler aangeven dat hij er klaar voor is door het gelijktijdig indrukken van de A- en B-toets. In dat geval zetten we een variabele "speelmodus" op waar. Gedurende het spel heeft de speler een aantal levens

ter beschikking. Elke keer dat een speler “dood” gaat gaat er één leven verloren. Als alle levens zijn opgemaakt zetten we de variabele “speelmodus” op niet-waar en is het spel afgelopen.

Maak een nieuwe variabele “speelmodus” en gebruik onderstaande code voor het indrukken van de twee knoppen.



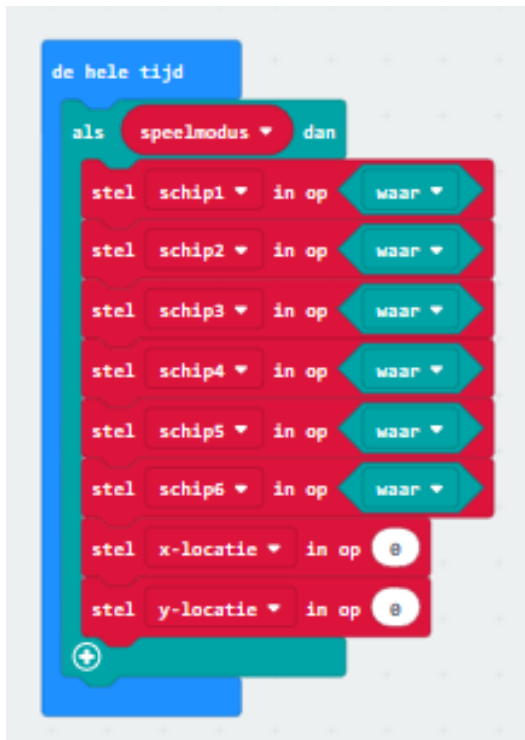
Nu willen we de schutter alleen zien als de variabele speelmodus op waar staat. Dat kunnen we voor elkaar krijgen door het plaatsen van een als-dan-blok om de code voor het tekenen van de schutter.



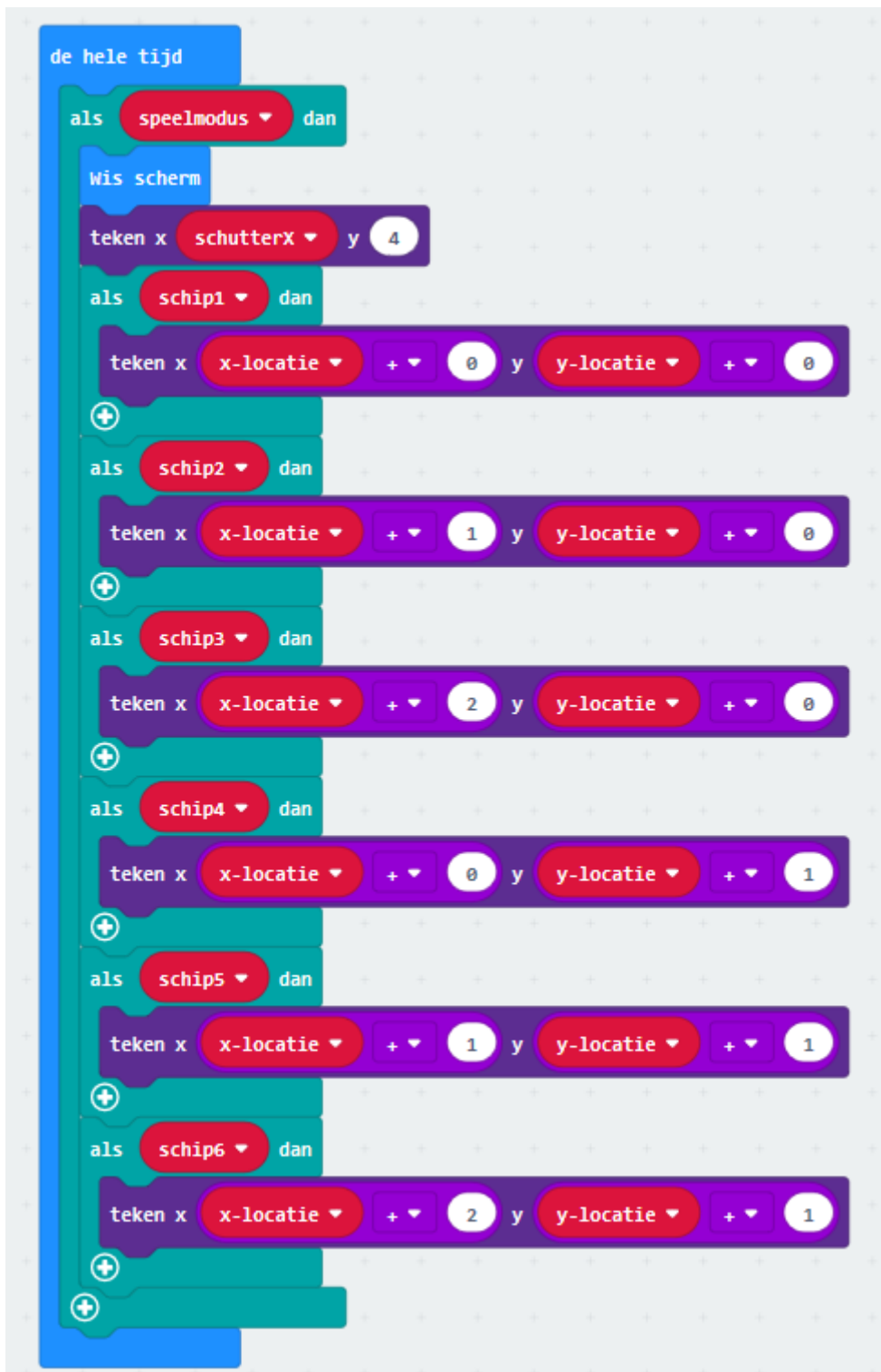
Ok, dat werkt. Laten we de ruimteschepen maar eens toevoegen. In totaal gaan we gebruik maken van zes ruimteschepen die als een geheel van links naar rechts over het scherm lopen. Van elk ruimte schip moeten we bijhouden of het “dood” of “levend” is. Daarvoor gebruiken we zes variabelen “schip1” tot en met “schip6”. Staat deze variabele op “waar”, dan is het ruimteschip levend, in het andere geval is het dood.

Voor het bijhouden van de locaties maken we gebruik van een variabele “x-locatie” en een “y-locatie”. De x-locatie geeft aan in welke kolom het meest linkse ruimteschip zit. De y-locatie geeft aan in welke rij het meeste rechtse ruimteschip zit.

Gebruik onderstaande code voor het instellen van deze variabelen.



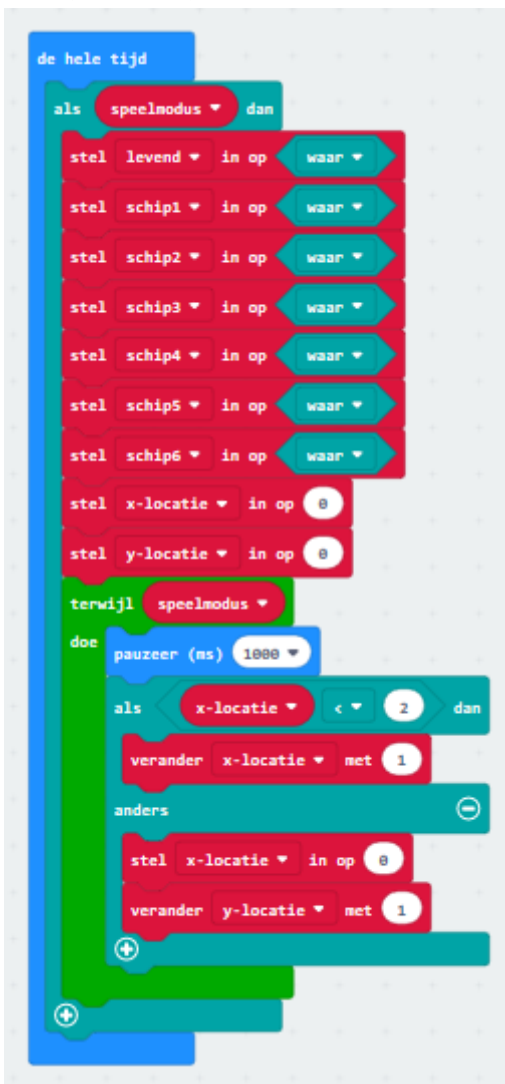
Een ruimteschip moet op het led-scherm getekend worden als het ruimteschip nog “levend” is. Voor de locatie gebruiken we de eerder genoemde variabelen x-locatie en y-locatie.



Zoals je ziet worden de ruimteschepen verdeeld over de eerste twee rijen en de eerste drie kolommen.

Drukken we nu tegelijkertijd op de A-knop en B-knop dan zien we de ruimteschepen in de linkerbovenhoek verschijnen. Ze staan nog stil maar dat gaan we nu aanpakken.

Voeg de volgende code toe (een uitbreiding op eerdere code).



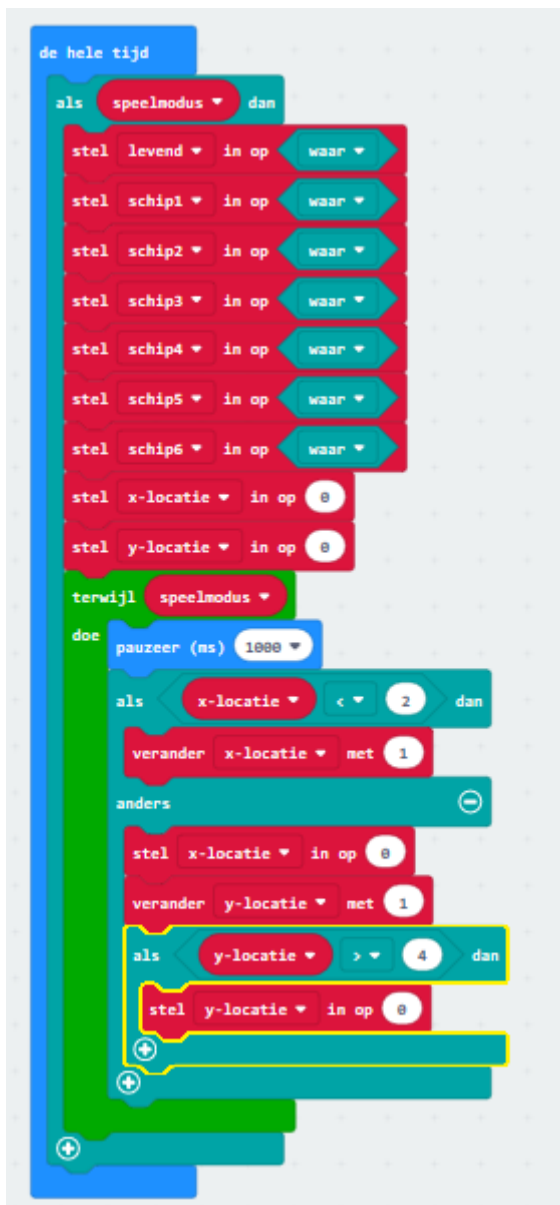
We laten de ruimteschepen langzaam van links naar rechts over het scherm schuiven waarbij het schip steeds één rij naar beneden zakt.

Als de x-locatie van de schepen 0 of 1 is kunnen de schepen één positie naar rechts schuiven. In het andere geval moet er weer vanaf links begonnen worden maar wel een rij lager.

Voor het tekenen van de ruimteschepen wordt steeds een pauze van 1 seconde ingelast, om het spel wat spannender te maken zou je die tijd natuurlijk kleiner kunnen maken.

Het probleem is nog wel dat de ruimteschepen op een gegeven moment onder aan het scherm verdwijnen en niet meer terugkomen. We zouden willen dat ze na beneden te zijn geweest weer bovenaan beginnen.

Dat betekent dat de y-locatie van de ruimteschepen bij het groter worden dan 4 teruggezet moet worden op 0.



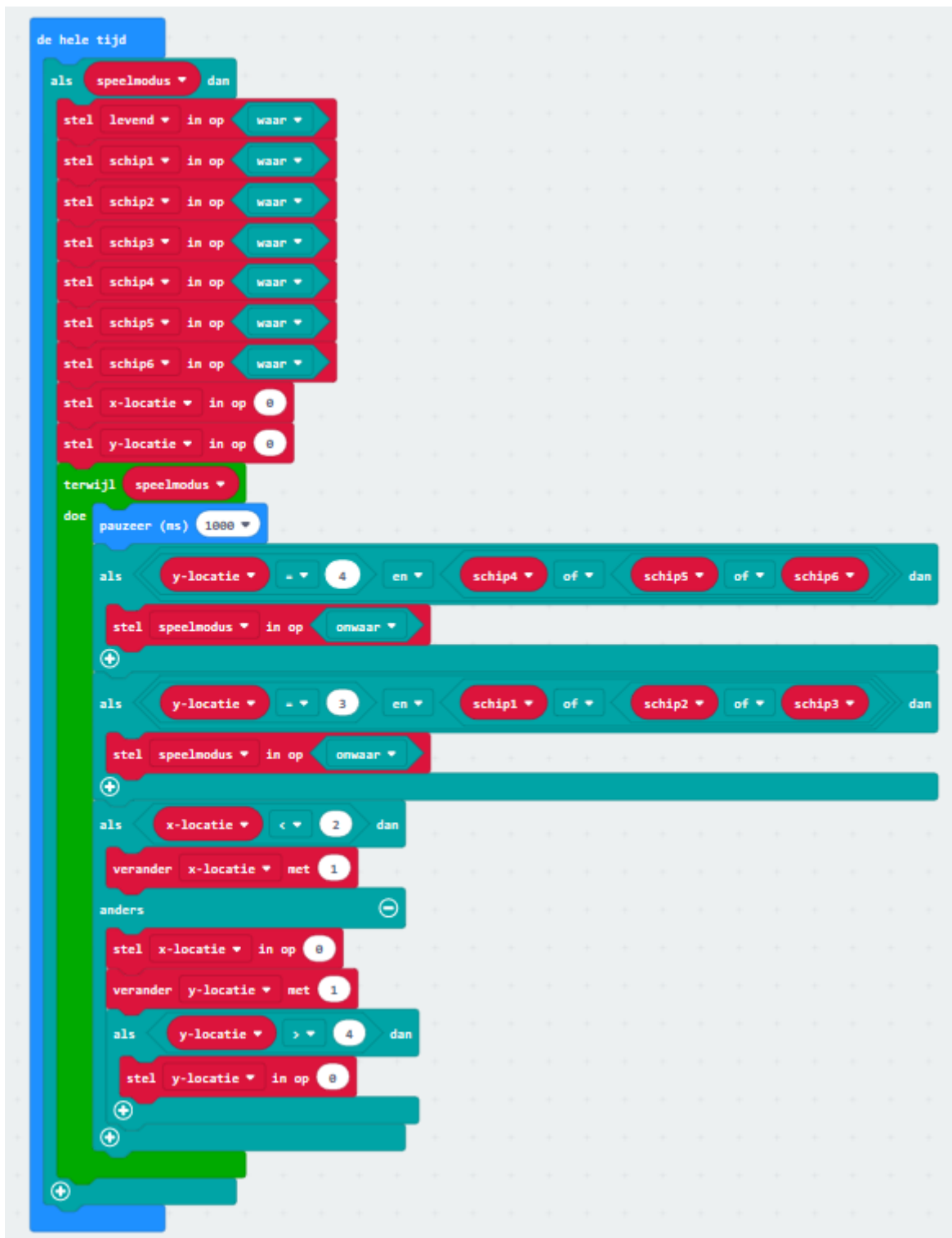
Probeer het maar, je zal zien dat de ruimteschepen weer van voor af aan beginnen.

De ruimteschepen bewegen nu wel mooi over het scherm maar als een van de schepen de schutter raakt is deze af (dood) en dat hebben we nog niet geprogrammeerd.

Maar hoe weten we nu wanneer de schutter geraakt wordt? De schutter bevindt zich altijd op de onderste rij en als er een ruimteschip op de onderste rij komt gaat de schutter zeker dood. Ruimteschepen boven hem kan de schutter wel neerhalen maar ruimteschepen naast hem niet meer.

Schip4, schip5 en schip6 bevinden zich op de onderste rij als de y-locatie gelijk

is aan 4. De schutter wordt echter alleen geraakt als een van die drie ruimteschepen nog leeft en dat moeten we dus gaan testen.



Nu we dat in orde hebben rest ons niets meer dan het neerschieten van de ruimteschepen te programmeren. Hiervoor gaan we de A-knop gebruiken, zodra we daar op drukken moet het onderste ruimteschip in dezelfde kolom

verdwijnen. Bij een volgende druk op de knop zou dan het bovenste ruimteschip kunnen verdwijnen.

```
wanneer knop A wordt ingedrukt
als schutterX = x-locatie dan
  als schip4 dan
    stel schip4 in op onwaar
  anders
    als schip1 dan
      stel schip1 in op onwaar
als schutterX = x-locatie + 1 dan
  als schip5 dan
    stel schip5 in op onwaar
  anders
    als schip2 dan
      stel schip2 in op onwaar
als schutterX = x-locatie + 2 dan
  als schip6 dan
    stel schip6 in op onwaar
  anders
    als schip3 dan
      stel schip3 in op onwaar
```

Door de opzet van het spel weten we dat ruimteschip1 en ruimteschip4 boven de schutter hangen als schutterX gelijk is aan x-locatie. Is dit het geval dan

moeten we eersten testen of ruimteschip5 al geraakt is. Is dat het niet geval dan moet het ruimteschip verdwijnen. Is dat wel het geval, dan kunnen we ruimteschip1 laten verdwijnen.

Als schutterX gelijk is aan x-locatie+1, dan hangen de ruimteschepen met nummer 2 en 5 boven de schutter en kunnen we die eventueel laten verdwijnen.

Eenzelfde strategie kunnen we toepassen op ruimteschip3 en ruimteschip6.

Als het goed is werkt het spel nu naar behoren, probeer het maar.