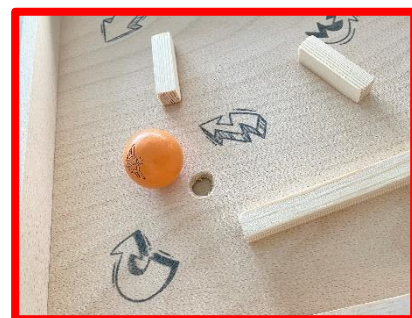
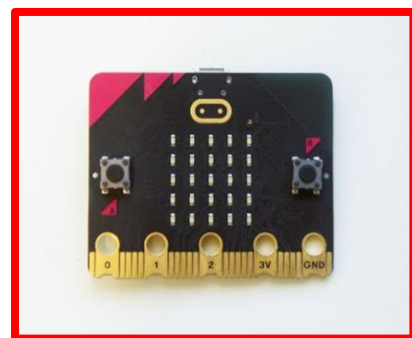
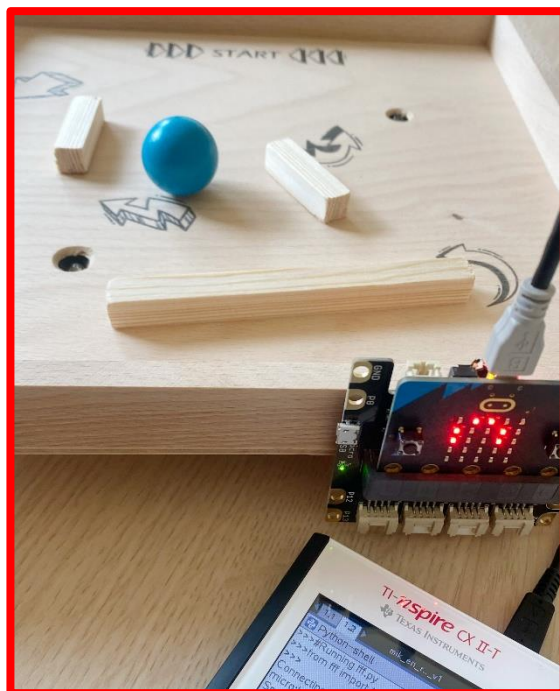




Spel 'Mik en rol' met de TI-Nspire en BBC micro:bit

Leerkrachtenbundel

*Ann-Kathrin Coenen
& Natalie Dirckx*

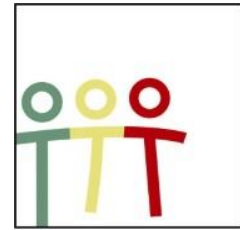


Inhoudsopgave

Inhoudsopgave	2
T³-Vlaanderen en T³-Nederland	3
Inleiding	3
Introductie van de BBC micro:bit	4
Programmeren met de TI-Nspire CX II	5
De benodigheden	6
Planning	7
Les 1: introductie en spelverloop ontwerpen	7
<i>Spelregels</i>	7
<i>Spelschema (Figuur 9)</i>	8
<i>Uitlezen lichtsensor</i>	8
Opstarten Python omgeving.....	8
Code schrijven voor één meting	9
Code schrijven meting per seconde	9
Les 2: experimenten en programmeren	10
<i>Experimenteren</i>	10
<i>Programmeren</i>	10
Tijdmodule invoegen en variabelen definiëren	11
Spelverloop coderen.....	11
Einde van het spel.....	12
Les 3: ontwerpen box en spelen	14
<i>Werkwijze houten box</i>	14
<i>Spelen</i>	14
Mogelijke uitbreidingen	15
<i>Meerdere lichtsensoren</i>	15
<i>Hindernissen toevoegen</i>	15
<i>Extra's toevoegen op display van micro:bit</i>	16
<i>Variëren in punten</i>	16
Spel	16

T³-Vlaanderen en T³-Nederland

Ann-Kathrin Coenen en Natalie Dirckx zijn als wetenschapsleerkrachten werkzaam aan het Agnetencollege Peer. Ze behoren ook tot het lerarennetwerk van T³ Vlaanderen dat nauw samenwerkt met Nederland. T³ staat voor Teachers Teaching with Technology. Het doel van deze organisatie is om de professionalisering van leerkrachten op het gebied van ICT en technologie in het onderwijs te bevorderen met gebruik van de technologie van Texas Instruments.



T³ VLAANDEREN

Figuur 1:
www.t3vlaanderen.be

Het is zeker de moeite waard om eens te snuisteren op de website, waar talloze kant-en-klare lesonderwerpen terug te vinden zijn.

Inleiding

Spelletjes ontwerpen is gewoon leuk voor de leerlingen. In de andere bundels zenuwspel en operatiespel wordt er enkel gebruik gemaakt van de pinnen van de micro:bit. In dit spel wordt de micro:bit uitgebreid met een Grove Shield voor micro:bit. Zo is het mogelijk om Grove sensoren te koppelen. Bij dit spel zullen de leerlingen leren om spelenderwijs een lichtsensor uit te lezen en de gemeten waarden te integreren in het spelverloop.

Dit project kan opgedeeld worden in een ontwerpend deel en een programmerend deel. Het bouwen van het labyrint voor de bal en het programmeren van het spel met Python op de TI-Nspire past ideaal binnen de lessen STEM, wetenschappen of techniek. Hierbij kan de link gelegd worden naar de leerstof over data capteren met sensoren in de 2^{de} graad. De leerlingen leren kennismaken met de lichtsensor. Verder zullen ze leren het spel in kleine onderdelen op te breken om de code te kunnen schrijven.

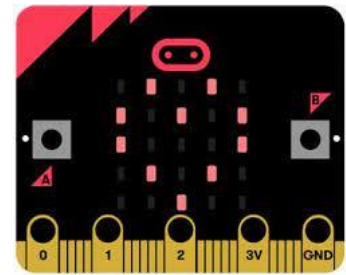
```
>>>
Connecting to micro:bit...
micro:bit is ready ✓
lichthoeveelheid = 0.6800000000000001
>>>|
```

Figuur 2: voorbeeld van een uitlezing van de lichtsensor met de handheld

Introductie van de BBC micro:bit

De BBC micro:bit is een populair, op zakformaat computertje of microcontroller. Het is een interface voor de samenwerking tussen software en hardware.

De micro:bit heeft een 5X5 LED lichtdisplay, drukknoppen A en B, aanraak invoerknoppen, ingebouwde microfoon en luidspreker. Daarnaast bevat deze microcontroller zelf nog vele sensoren onder andere voor temperatuur, licht, beweging en een kompas. Tot slot is interactie met andere toestellen of het internet ook mogelijk door middel van een bluetooth verbinding.



Figuur 3: BBC micro:bit

Ben je zelf niet bekend met de onderdelen van de micro:bit? Voor de leerlingen hebben we een BookWidget uitgewerkt. Via deze [Lerarenlink](#) kan je deze widget zelf gebruiken.

De micro:bit voert acties uit na het programmeren van instructies. Deze instructies worden in de programmeertaal Python geschreven op het grafische rekenmachine TI-84 Plus CE-T Python Edition of TI-Nspire CX II (Figuur 5). In deze bundel werd gekozen voor de Nspire om dit project uit te voeren.



Figuur 4: python programmeertaal

Python is een open source programmeertaal die eenvoudig en duidelijk is en toch breed toepasbaar in technologieën. Voor beginners wordt programmeren in python aangeraden en dat maakt deze programmeertaal zeer geschikt voor leerlingen.

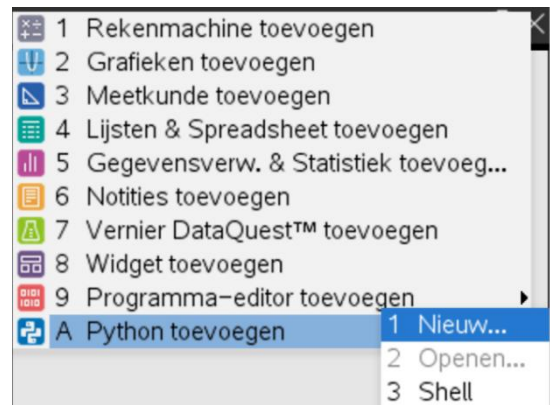


Figuur 5: TI-Nspire CX II

Programmeren met de TI-Nspire CX II

De TI-Nspire CX II is een grafische rekenmachine met praktische leermiddelen voor zowel de wiskunde- als de natuurwetenschappenlessen. Dit is zowel via de software als handheld te gebruiken.

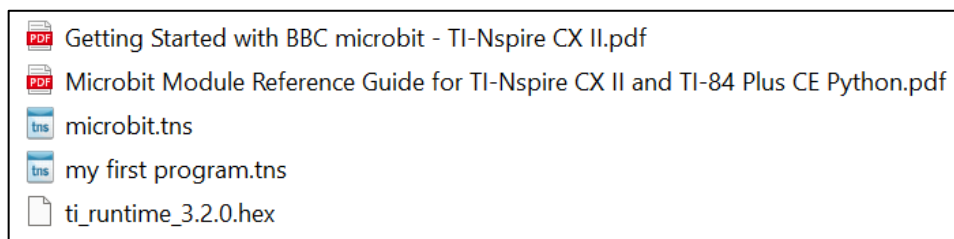
Voor dit project zal er gebruikgemaakt worden van de python module. De code kan zowel op de laptop als de handheld geprogrammeerd worden. Een nieuwe pythoncode schrijf je door in het home-menu een nieuw document te maken en vervolgens 'Python toevoegen' te selecteren (Figuur 6). Met behulp van de menu-knop is het mogelijk om deels voorgeschreven stukken code toe te voegen.



Figuur 6: een nieuw python-bestand maken op de TI-Nspire

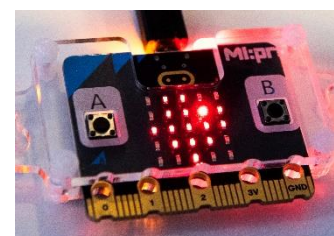
Zowel de micro:bit als de handheld moeten van een module voorzien worden vooraleer het mogelijk is om te kunnen programmeren voor de micro:bit.

Via de website van [TI education](https://www.ti.com/education) kan je de nodige files (Figuur 7) downloaden als zip. Alle nodige bestanden vind je in deze map inclusief een stappenplan voor installatie. Na installatie van microbit.tns is het mogelijk om bepaalde functies van de micro:bit te kunnen gebruiken in de Python pagina van de handheld.



Figuur 7: microbit.tns bevat de module voor de handheld, ti_runtime.hex is de module voor de microbit

Op de micro:bit dient een hex-file geïnstalleerd te worden. Wanneer de code succesvol op de micro:bit wordt geplaatst, zal het Texas Instruments logo verschijnen. De micro:bit kan via de USB mini naar micro kabel verbonden worden met de TI-Nspire.

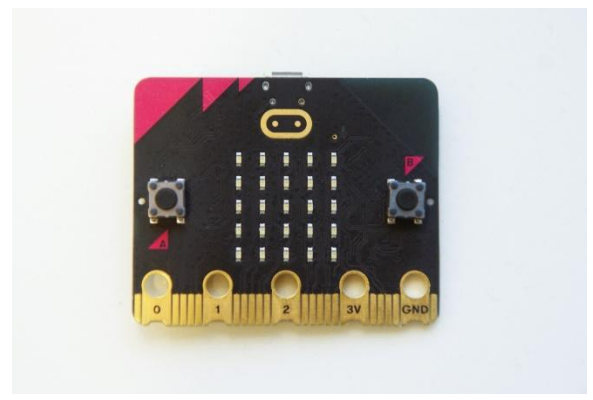
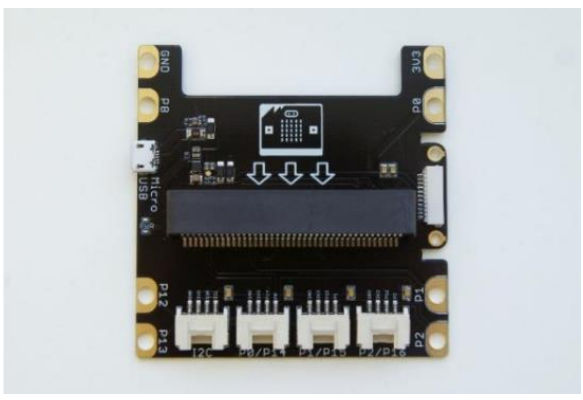

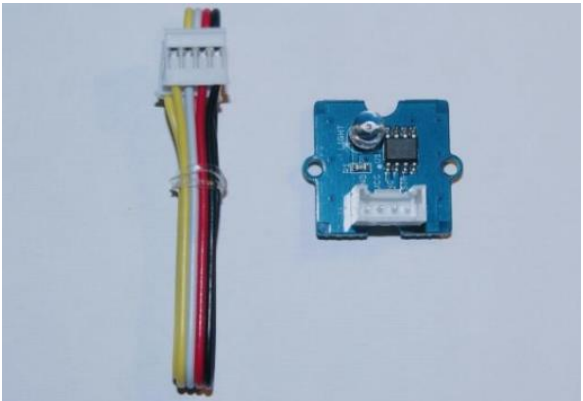


Figuur 8: Texas Instruments logo op display micro:bit

De benodigheden

Voor dit spel zijn de volgende benodigheden vereist:

- TI-Nspire CX II
- Micro:bit v2
- Grove Shield voor micro:bit v2.0
- Grove - Light sensor v1.2
- Mini-USB naar micro-USB kabel
- Houten latjes 1 cm dik
- Houten box van minimale afmeting 15 cm X 15 cm 'Art Panel' (www.hobbyshop-online.nl)
- 2x houten bal diameter 3 cm (but pétanque)
- 2x plastic melkdopjes
- Boor 10 mm en soeverinboor 16 mm
- Schaaflatje 12x12 mm
- Tape
- Houtlijm

	
BBC micro:bit	Grove Shield
	
Houten box (Art Panel)	Grove - Light sensor v1.2

Planning

Het spel kan opgebouwd worden volgens volgende planning.

Les 1: introductie en spelverloop ontwerpen

- introductie met micro:bit en Grove Shield
- Spelverloop ontwerpen
- Leren uitlezen lichtsensor

Les 2: experimenten en programmeren

- bepalen lichtsterkte voor spel
- programmeren

Les 3: ontwerpen box en spelen

- Box knutselen
- Spel spelen

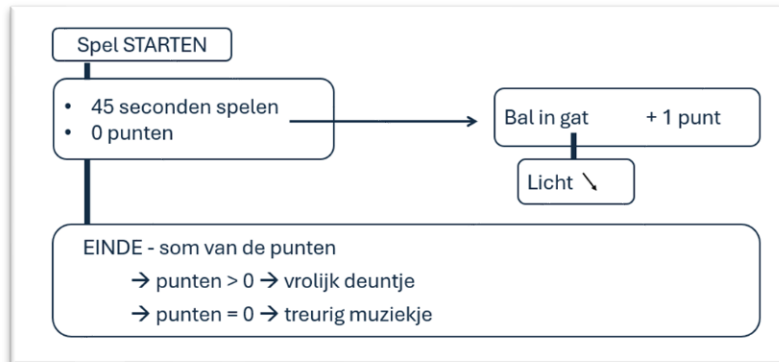
Les 1: introductie en spelverloop ontwerpen

Spelregels

1. Neem de bal en laat deze los in de box op de startplaats.
2. Leg het spel op de tafel en ga zitten.
3. Start het spel.
4. Neem met je handen de randen van de houten box vast.
5. Probeer gedurende 45 seconden door de box te bewegen de bal te mikken in een gat.
6. Verdien een punt wanneer de bal in het gat is beland.
7. Neem de bal direct uit het gat als het geluid gaat.
8. Plaats de bal opnieuw op de startplaats en speel het spel verder.
9. Wie na 25 seconden het meeste punten heeft verdiend, is de winnaar van het spel.

Spelschema (Figuur 9)

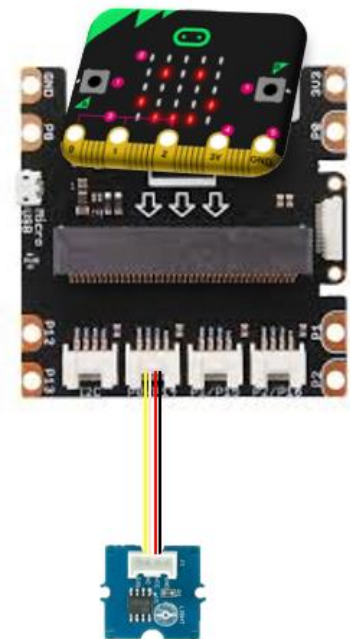
Het maken van een spelschema op basis van de spelregels helpt de leerlingen om een structuur te zien in het spel. Om te kunnen coderen is het belangrijk om te weten wat allemaal in de code opgenomen dient te worden. Door de leerlingen zelf een spelschema te laten maken en te laten nadenken over de kleine stapjes, zal het programmeren makkelijker verlopen.



Figuur 9: voorbeeld van een mogelijk spelschema

Uitlezen lichtsensor

Om te detecteren wanneer er een bal in het gat zit, wordt gebruik gemaakt van een lichtsensor. Als de bal het gat afsluit waarin de sensor zit, zal de lichtintensiteit afnemen. Schuif de BBC micro:bit in de Grove Shield. Sluit de sensor met behulp van de bijhorende kabel aan op de pin 0-poort van de Grove Shield (Figuur 10). Koppel de micro:bit met behulp van de bijhorende kabel aan de handheld (Figuur 11).



Figuur 10: aansluitschema



Figuur 11: mini naar micro usb-kabel

Schrijf volgende code om een meting van de lichtsensor te verkrijgen.

Opstarten Python omgeving

Ga naar het home scherm en kies 'nieuw'. Hier kan je 'Python toevoegen' kiezen. Daarna kies je 'Nieuw'. Kies een naam voor je programma zonder een spatie te gebruiken. Via de

menu- knop kan je de nodige functies toevoegen. Het is ook altijd mogelijk om de code zelf in te typen.

Code schrijven voor één meting

```
sensor_test.py
from microbit import*
print("lichthoeveelheid =",grove.read_lightlevel(pin0))
```

De code 'from microbit import*' maakt het mogelijk om de library code voor micro:bit te gebruiken.

Bij menu 'Meer modules' kan je 'BBC micro:bit' kiezen. De eerst optie is 'from microbit import'.*

Op het scherm van de handheld willen we de lichthoeveelheid weergeven.

Kies bij menu 'Ingebouwde functies' en ga naar 'I/O'. De eerste optie is print(). De tekst "lichthoeveelheid = " dien je zelf te typen met de letters onderaan. De "-tekens duiden dat het geprint wordt als tekst. Hierna typ je een ',' en kies je bij de micro:bit voor de Grove Devices. Bij input kies je voor de lightlevel sensor en je vult dan nog "pin0" in.

Als de code geschreven is, kan deze gerund worden. Dit doe je door in 'menu', 'uitvoeren' te kiezen. Dit kan ook door 'Ctrl'+ 'R' te drukken. In een nieuwe pagina zal een Shell openen. Met 'Ctrl' + pijltje naar links of rechts kan je tussen de pagina's wisselen. Wil je iets in je code veranderen, zal je dus op de vorige pagina je aanpassing moeten maken en het programma opnieuw runnen in de shell.

Code schrijven meting per seconde

De leerlingen zullen zien dat er maar één waarde geprint wordt op de handheld. Het programma moet opnieuw gerund worden om een nieuwe meetwaarde te bekomen. Om voortdurend nieuwe waarden te laten printen, kan er gebruik gemaakt worden van een 'while' lus die de meting blijft opvragen en printen tot er op 'esc' gedrukt wordt.

```
from microbit import*
while get_key() != "esc":
    print("lichthoeveelheid =",grove.read_lightlevel(pin0))
    sleep(1000)
```

Voeg in de regel boven de print code de 'while' lus in.

Kies bij het menu 'meer modules' voor 'TI systems'. Daar kan je bovenstaande 'while' lus kiezen.

Voor de print code dien je een tab te gebruiken. Als er geen blokjes voor print code staan, behoort deze code niet tot de lus en krijg je weer maar één meting.

Deze lus gaat blijven lopen tot er op 'esc' gedrukt wordt. We willen wel dat er telkens om de 1 seconde een meting geprint wordt op de handheld. Bij een tijdsingave voor de micro:bit dient er rekening gehouden te worden met de eenheid in milliseconden.

Kies bij het menu 'meer modules' voor 'Time'. Daar kan je bovenstaande 'Sleep' kiezen en vul binnen de haakjes 1000 in.

De leerlingen kunnen nu de waarden van de lichtsensor uitlezen en laten printen. Dit dient gebruikt te worden voor het experiment dat de leerlingen zullen uitvoeren in de les.

Les 2: experimenten en programmeren

Experimenteren



Figuur 13: sensor in dopje

De sensor moet in het gat bevestigd worden, maar het kabeltje steek nog uit. Het makkelijkst kan dit door de sensor in een dopje van een plastic melkfles te bevestigen (Figuur 13). Dit plakken we later dan onder het gat waar het balletje op kan liggen (Figuur 17). De leerlingen moeten even experimenteren hoe de lichtwaarde zal veranderen bij het afdekken van het melkdopje (Figuur 12). Want een donkergekleurd dopje laat minder licht door dan een lichtgekleurd dopje.



Figuur 12: afgedekte sensor

In de code voor het spel zullen ze dus hun experimentele waarde moeten ingeven die de sensor zal aangeven bij het afdekken van het dopje. Zo kan bijvoorbeeld de grens op 15 liggen. Wordt er dan een lichtintensiteit van minder dan 15 behaald, dan wordt er een punt toegekend.

Programmeren

Volgens het spelschema (Figuur 9) kan nu de code opgebouwd worden. Volgende stappen dienen geprogrammeerd te worden.

1. De tijd moet gestart worden want het spel mag maar 45 seconden duren.
2. Wanneer de lichtintensiteit onder een bepaalde waarde zakt, wordt er een punt bijgeteld.
3. Na 45 seconden wordt er gekeken of de punten groter zijn dan nul of niet met een bijhorend blij of droevig muziekje.

Tijdmodule invoegen en variabelen definiëren

```
* fff.py
from microbit import*
from time import *

punten=0
t0=clock()
tijd=0
```

Ook nu wordt de microbit module geïmporteerd zoals bij les 1 reeds is vermeld.

Door het invoegen van 'from time import*' is het mogelijk om tijd te gebruiken in het geschreven programma.

Kies bij menu 'Meer modules', daar kan je 'Time' kiezen. De eerst optie is 'from time import'.*

Eerst moeten we nog definiëren dat er punten zijn in het spel en dat het spel start met 0 punten. Verder moet bij de start de klok begint te lopen.

Typ zelf 'punten = 0', 't0 = clock' en 'tijd = 0'. Clock() kan je ook terugvinden bij menu 'Meer modules', daar kan je 'Time' kiezen en zie je ook clock() staan.

Hierdoor maken we variabelen aan die we later in de code kunnen oproepen.

```
print("Spel bezig")
```

Op de handheld willen we graag ook de tekst "Spel bezig" printen.

Kies bij menu 'Ingebouwde functies' en ga naar 'I/O'. De eerste optie is print(). De tekst "Spel bezig" dien je zelf te typen met de letters onderaan.

Spelverloop coderen

```
while tijd <25:
    display.show(punten)
    tijd=clock()-t0
```

We maken een nieuwe while lus die blijft lopen zolang de tijd minder dan 25 seconden bedraagt. We definiëren de tijd als zijnde de huidige tijd min de starttijd. Wanneer deze groter is dan 25 seconden, stopt deze lus. Gedurende de speltijd willen we op het scherm van de micro:bit het aantal punten zien.

Kies bij menu 'Ingebouwde functies' en ga naar 'de optie 'besturing' en kies 'while'. Wanneer je op de var-toets drukt verschijnt er een keuze lijstje en kies je 'tijd'. '<25' typ je zelf.

Om de punten op het display van de micro:bit te tonen, ga je weer naar 'Display' in de microbit module.

De regel van de 'tijd =...' kan je analoog opbouwen met de var toets.

```
if grove.read_lightlevel(pin0)<15:  
    punten = punten+1  
    music.play(music.JUMP_UP)
```

Als de lichtintensiteit onder de waarde van 15 (wordt bepaald het uitgevoerde experiment) zakt, dan moet er een punt worden bijgeteld. Tegelijk willen we een leuk muzikje horen zodat de speler ook auditief feedback krijgt van het behaalde punt.

Hiervoor wordt een if functie gebruikt.

'If' vind je bij de ingebouwde functies 'Besturing'. Hierna voeg je de sensor weer toe door bij de micro:bit voor de Grove Devices te kiezen. Bij input kies je voor de lightlevel sensor en je vult dan nog "pin0" in. Hierachter typ je zelf <5.

Omdat de volgende regels nog steeds horen bij deze if functie zullen er weer twee blokjes extra inspringen. Als deze niet automatisch worden toegevoegd, kan dat ook met de tab toets.

De punten worden weer opgebouwd met de var toets. In de microbit module kan je bij 'Music' een deuntje kiezen.

Einde van het spel

```
if punten>0:  
    display.show(punten)  
    music.play(music.ENTERTAINER)  
else:  
    music.play(music.DADADADUM)  
print("Spel einde ->Puntenaantal =",punten)  
display.clear()
```

Wanneer de lus van 25 seconden afgelopen is, willen we dat de punten zichtbaar worden. Als de punten meer dan nul bedragen, willen we dat er een blij muzikje speelt en anders willen we een droevig muzikje. Dit kan met een 'if ... else' functie. Aangezien de 'while' lus nu afgelopen is, start de 'if...else' functie zonder de blokjes.

'If...else' vind je bij de ingebouwde functies 'Besturing'. Na 'if' typ je >0 zelf. In de micro:bit module kan je de display weer toevoegen en tussen haakjes kan je met de var-toets punten kiezen. Ook kan je in de micro:bit module weer een muzikje toevoegen.

Na 'else' voeg je een droevig muzikje in want dan heb je nul punten behaald.

Via print laten we de punten op de handheld verschijnen.

We willen het display van de micro:bit ook weer leegmaken. Deze laatste stap is niet noodzakelijk voor de werking van het spel.

Als je voor jezelf een opmerking bij de code wilt typen die niet wordt uitgevoerd, dan kan je deze noteren na een # teken. Deze tekst wordt lichtgrijs weergegeven in de code.

```

from microbit import *
from time import *

punten=0
t0=clock()
tijd=0

print("Spel bezig")

while tijd <25:
    ♦♦ display.show(punten)
    ♦♦ tijd=clock()-t0
    ♦♦ if grove.read_lightlevel(pin0)<15:
        ♦♦♦♦ punten = punten+1
        ♦♦♦♦ music.play(music.JUMP_UP)
    if punten>0:
        ♦♦ display.show(punten)
        ♦♦ music.play(music.ENTERTAINER)
    else:
        ♦♦ music.play(music.DADADADUM)
print("Spel einde ->Puntenaantal =",punten)
display.clear()

```

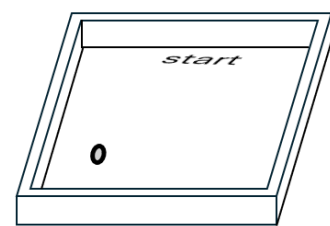
Les 3: ontwerpen box en spelen

Niet alle leerlingen zullen al klaar zijn na les 2 met het coderen van hun code. Afhankelijk van de tijd die gespendeerd wordt aan het opbouwen van de box, kan er dus nog tijd voorzien worden om verder te programmeren.

Werkwijze houten box

Het bouwen van dit spel gebeurt als volgt voor een houten balletje met een doorsnede van 30 mm:

1. Neem het houten plankje en boor op een willekeurige plaats een gat van 10 mm.
2. Boor in het gat nog even na met een soevereinboor van 14 mm (Figuur 18).
3. Markeer de startplaats in het houten bakje (Figuur 15).
4. Plak het dopje met de lichtsensor onder het gat met tape (Figuur 17).
5. Sluit de sensor aan op pin0 van de Grove Shield.
6. Schuif de micro:bit in de Grove Shield.
7. Kopel de micro:bit aan de handheld.



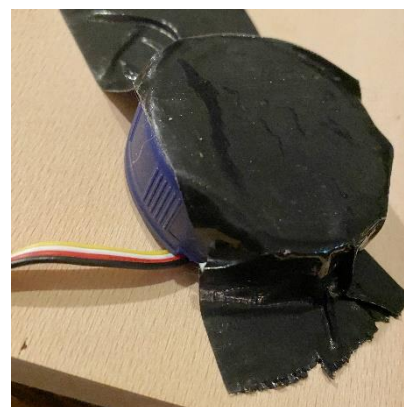
Figuur 15: houten bakje met geboord gat en startplaats



Figuur 16: startaanduiding



Figuur 18: boorgat na soevereinboor



Figuur 17: sensor onder het gat geplakt

Spelen

Het spel is nu klaar om te spelen. Run de code en speel.

Mogelijke uitbreidingen

Meerdere lichtsensoren.

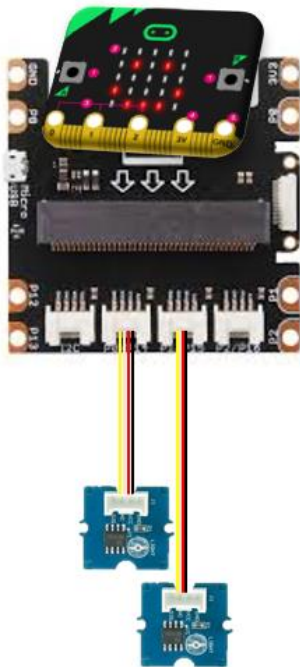
In plaats van maar één gat te maken, is het ook mogelijk om meerdere gaten te maken. Je kan tot drie lichtsensoren aansluiten aan de Grove Shield.

In de 'while' lus kan er een 'if' functie toegevoegd worden (Figuur 19).

De tweede lichtsensor wordt dan aangesloten op de tweede pin (Figuur 20) van de Grove Shield. Pas dit ook aan in de code.

```
while tijd <25:  
  display.show(punten)  
  tijd=clock()-t0  
  if grove.read_lightlevel(pin0)<15:  
    punten = punten+1  
    music.play(music.JUMP_UP)  
  if grove.read_lightlevel(pin1)<15:  
    punten = punten+1  
    music.play(music.JUMP_UP)
```

Figuur 19: code met twee lichtsensoren



Figuur 20: aansluiting van tweede lichtsensor

Hindernissen toevoegen

Om het spel attractiever te maken kunnen er hindernissen worden toegevoegd. Er kunnen nageltjes geklopt worden in het parcours of er kunnen schaaflatjes (12x12 mm) op het parcours worden gelijmd.



Figuur 21: hindernissen met schaaflatjes

Extra's toevoegen op display van micro:bit

Wanneer een punt wordt verdiend, wordt er een muziekje afgespeeld. Het is ook mogelijk om tegelijk een afbeelding op de micro:bit te laten verschijnen. De code hiervoor kan in de 'if' functie worden bijgeschreven (Figuur 22). Omdat je na het punt direct weer wilt weten wat het puntensaldo is, dien je ook het display te wissen zodat de afbeelding weer verdwijnt.

```
while tijd < 45:  
  display.show(punten)  
  tijd=clock()-t0  
  if grove.read_lightlevel(pin0)<15:  
    punten = punten+1  
    display.show(Image.YES)  
    music.play(music.JUMP_UP)  
    display.clear()
```

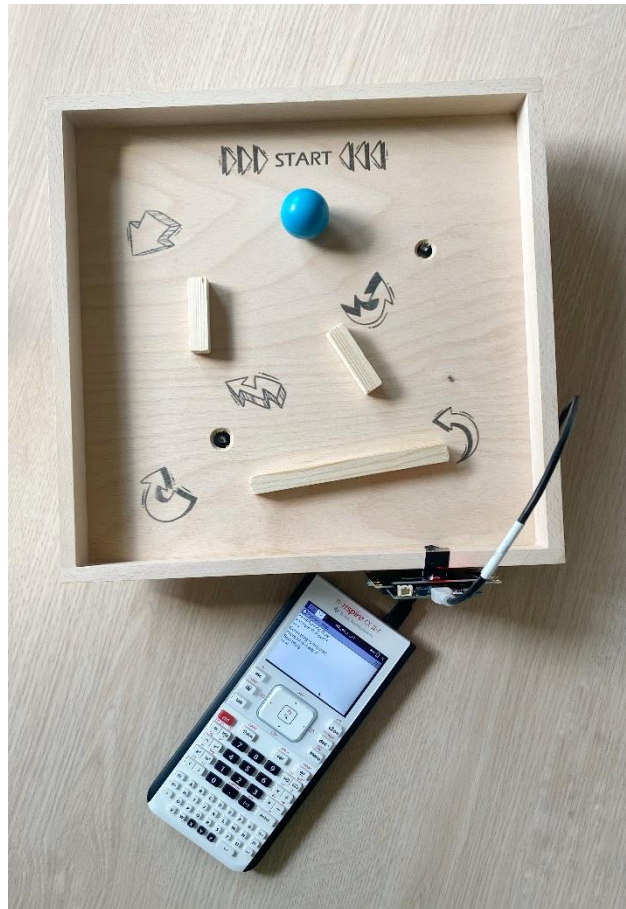
Figuur 22: afbeelding tonen op micro:bit

Variëren in punten

Als één gat met hindernissen bemoeilijkt wordt, kan daar twee punten aan toegekend worden. De code is dan 'punten=punten+2'. Spelers kunnen kiezen om het balletje te mikken op het gat van één of twee punten binnen hun speltijd.

Spel

In Figuur 23 wordt een afgewerkt spel getoond met schaaflatjes als hindernissen en twee lichtsensoren. De micro:bit is aan de zijkant van de box bevestigd met tape. Het makkelijkst is het spel al zittend te spelen omdat de micro:bit en de handheld nog met kabels verbonden zijn met de onderkant.



Figuur 23: het afgewerkt spel