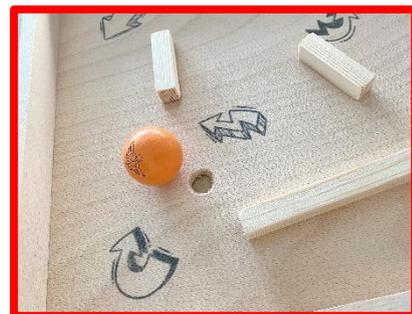
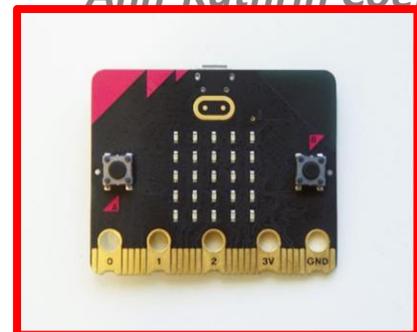




"Ziel- und Rollen" spiel mit dem TI-Nspire und BBC micro:bit

Lehrerbündel

Ann-Kathrin Coenen
rckx

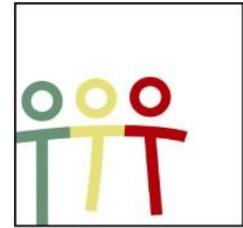


Inhaltsverzeichnis

T3-Flandern en T³-Niederlande	3
Einführung	3
Vorstellung des BBC micro:bit	4
Programmierung mit dem TI-Nspire CX II	5
Zubehör	6
Planung	7
Lektion 1: Einführung und Gameplay-Design	7
<i>Regeln</i>	7
<i>Spielplan (Abbildung 9)</i>	8
<i>Lichtsensordaten</i>	8
Start-Python-Umgebung.....	9
Schreiben von Code für eine einzelne Messung.....	9
Code-Schreibmessung pro Sekunde	9
Lektion 2: Experimentieren und Programmieren	10
<i>Experiment</i>	10
<i>Programm</i>	11
Zeitmodul einfügen und Variablen definieren	11
Ablauf des Programmierspiels.....	12
Ende des Spiels	13
Lektion 3: Entwerfen der Box und Spielen	14
<i>Kasten bauen</i>	14
<i>Spiel spielen</i>	14
Mögliche Erweiterungen	15
<i>Mehrere Lichtsensoren</i>	15
<i>Hinzufügen von Hindernissen</i>	15
<i>Hinzufügen von Extras auf dem micro:bit-Display</i>	16
<i>Unterschiedlich in Punkten</i>	16
Spiel	16

T3-Flandern en T³-Niederlande

Ann-Kathrin Coenen und Natalie Dirckx sind Lehrerinnen für Naturwissenschaften am Agnetencollege Peer. Sie gehören zu dem Lehrernetzwerk von T³ Flandern, das eng mit den Niederlanden zusammenarbeitet. T³ steht für Teachers Teaching with Technology (Lehrer unterrichten mit Technologie). Mit Unterstützung von der Technologie von Texas Instruments befördern sie die Professionalisierung von Lehrer im Bereich der Informatik und Technologie im Unterricht.



T³ VLAANDEREN

Abbildung 1:
www.t3vlaanderen.be

Einführung

Das Entwerfen von Spielen macht den Schülern einfach Spaß. In den anderen Bündeln von der Nervenspirale und das Operationsspiel werden nur die Pins des micro:bit verwendet. In diesem Spiel wird der micro:bit mit einem Grove Shield für micro:bit erweitert. So ist es beispielsweise möglich, Grove-Sensoren anzuschließen. In diesem Spiel lernen die Schüler, einen Lichtsensor spielerisch auszulesen und die Messwerte in den Spielverlauf zu integrieren.

Dieses Projekt kann in einen entwerfenden und einen programmierenden Teil aufgeteilt werden. Das Labyrinth für den Ball bauen und das Spiel mit Python auf dem TI-Nspire programmieren, passt perfekt in den MINT-, Naturwissenschafts- oder Ingenieurunterricht. Die Verbindung kann zum Thema Datenerfassung mit Sensoren hergestellt werden. Die Schüler lernen den Lichtsensor kennen. Außerdem lernen sie, das Spiel in kleine Teile zu zerlegen, um den Code zu schreiben.

```
>>>
Connecting to micro:bit...
micro:bit is ready ✓
Lichtmenge = 4.49
>>>|
```

Abbildung 2: Beispiel für die Datenmessung

Vorstellung des BBC micro:bit

Der BBC micro:bit ist ein beliebter Computer oder Mikrocontroller im Taschenformat. Er fungiert als eine Verbindung zwischen Software und Hardware.

Der micro:bit verfügt über eine 5X5-LED-Lichtdisplay, die Drucktasten A und B, Berührungstasten, ein eingebautes Mikrofon und einen Lautsprecher. Darüber hinaus enthält dieser Mikrocontroller selbst viele Sensoren, unter anderem für Temperatur, Licht, Bewegung und einen Kompass. Schließlich ist auch die Interaktion mit anderen Geräten oder dem Internet über eine Bluetooth-Verbindung möglich.

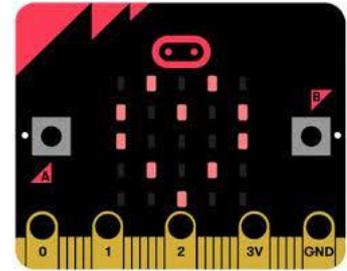


Abbildung 3: BBC micro:bit

Der micro:bit führt nach Programmieranweisungen Aktionen aus. Diese Anweisungen werden in der Programmiersprache Python auf dem grafischen Taschenrechner TI-84 Plus CE-T Python Edition oder TI-Nspire CX II (Abbildung 5) geschrieben. In diesem Bündel wurde der Nspire für die Durchführung dieses Projekts gewählt.



Abbildung 4: Programmiersprache Python

Python ist eine Open-Source-Programmiersprache, die einfach und eindeutig ist und dennoch in vielen Bereichen der Technik eingesetzt werden kann. Die Programmierung in Python wird für Anfänger empfohlen, das macht diese Programmiersprache sehr geeignet für Studenten.



Abbildung 5: TI-Nspire CX II

Programmierung mit dem TI-Nspire CX II

Der TI-Nspire CX II ist ein grafischer Taschenrechner mit praktischen Unterrichtswerkzeugen für den Mathematik und Naturwissenschaften Unterricht. Er kann sowohl mit der Software sowie als Handheld verwendet werden.

Das Python-Modul wird für dieses Projekt verwendet. Der Code kann sowohl auf dem Laptop sowie auf dem Handheld programmiert werden. Sie schreiben neuen Python-Code, indem Sie im Home-Menü ein neues Dokument erstellen und dann "Python hinzufügen" wählen (Abbildung 6). Über die Menütaste können Sie teilweise vorgegebene Codestücke hinzufügen.

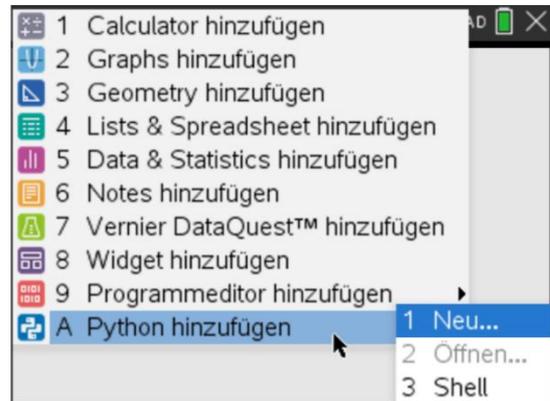


Abbildung 6: Erstellen einer neuen Python-Datei auf dem TI-Nspire

Sowohl der micro:bit als auch der Handheld müssen mit einem Modul ausgestattet sein, bevor eine Programmierung für den micro:bit möglich ist.

Sie können die erforderlichen Dateien (Abbildung 7) als Zip-Datei von der [TI Education Website](#) herunterladen. In diesem Ordner finden Sie alle notwendigen Dateien sowie einen Schritt-für-Schritt-Installationsplan. Nach der Installation von microbit.tns ist es möglich bestimmte Funktionen von dem micro:bit auf der Python-Seite des Handheld zu nutzen.

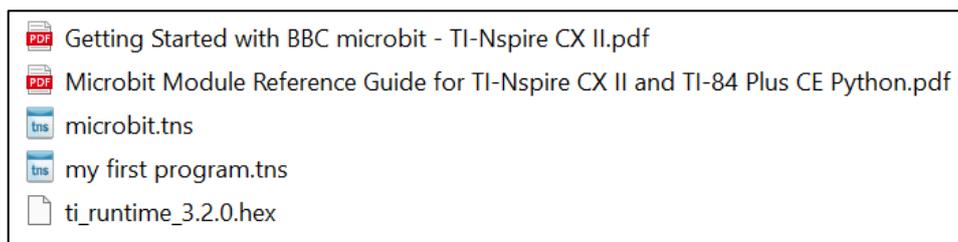


Abbildung 7: microbit.tns enthält das Modul für das Handheld, ti_runtime.hex ist das Modul für das Microbit

Eine Hex-Datei muss auf dem micro:bit installiert werden. Wenn der Code erfolgreich auf dem micro:bit platziert wurde, erscheint das Logo von Texas Instruments. Der micro:bit kann mit dem USB-Mini-zu-Mikro-Kabel an den TI-Nspire angeschlossen werden.

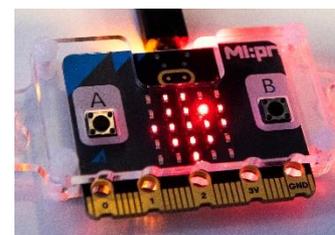
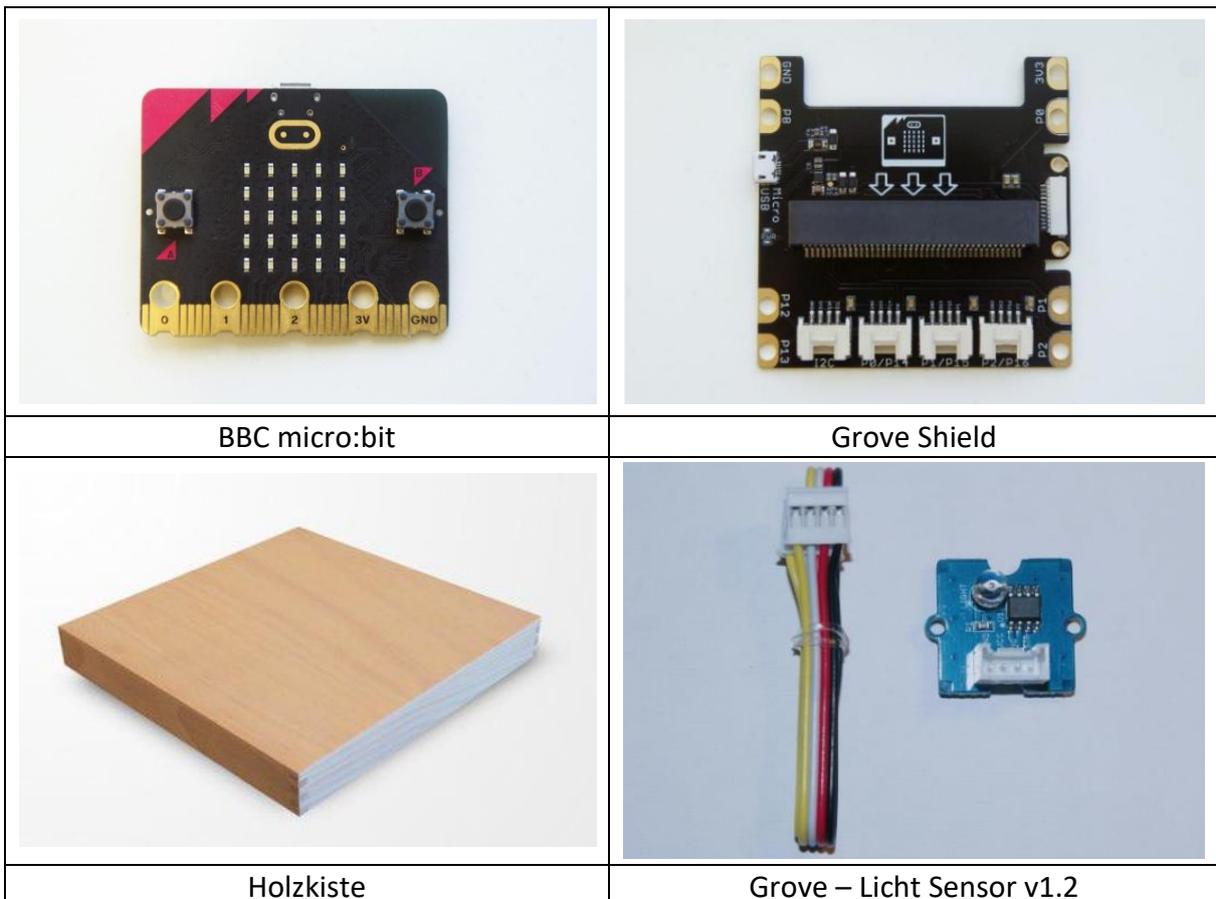


Abbildung 8: Texas Instruments Logo auf dem Display micro:bit

Zubehör

Die folgenden Materialien werden für dieses Spiel benötigt:

- TI-Nspire CX II
- Micro:bit v2
- Grove Shield für micro:bit v2.0
- Grove - Lichtsensor v1.2
- Mini-USB nach Micro-USB Kabel
- Holzlatten 1 cm dick
- Holzkiste mit einer Mindestgröße von 15 cm x 15 cm 'Art Panel' (www.hobbyshop-online.nl)
- 2x Holzkugel Durchmesser 3 cm
- 2x Milchkappen aus Kunststoff
- Bohrer 10 mm und Senkkopfbohrer 16 mm
- Hobelleisten 12x12 mm
- Klebeband
- Holzleim



Planung

Das Spiel kann nachfolgendem Zeitplan aufgebaut werden.

Lektion 1: Einführung und Gameplay-Design

- Einführung mit micro:bit und Grove Shield
- Gameplay entwerfen
- Lernen den Lichtsensor zu lesen

Lektion 2: Experimentieren und Programmieren

- Lichtintensität beim Spielen bestimmen
- Programm

Lektion 3: Entwerfen der Box und Spielen

- Kasten bauen
- Spiel spielen

Lektion 1: Einführung und Gameplay-Design

Regeln

1. Nehmen Sie den Ball und lassen Sie ihn am Startpunkt in den Startraum los.
2. Legen Sie das Spiel auf den Tisch und setzen Sie sich.
3. Starten Sie das Spiel.
4. Nehmen Sie die Kanten der Holzkiste mit den Händen fest.
5. Versuchen Sie 45 Sekunden lang, den Ball durch Bewegen des Kastens in ein Loch zu zielen.
6. Erhalte einen Punkt, wenn der Ball im Loch gelandet ist.
7. Nimm den Ball direkt aus dem Loch, wenn der Ton ertönt.
8. Lege den Ball zurück auf den Startplatz und setze das Spiel fort.
9. Wer nach 25 Sekunden die meisten Punkte gesammelt hat, ist der Gewinner des Spiels.

Spielplan (Abbildung 9)

Das Erstellen eines Spielplans basierend auf den Spielregeln hilft den Schülern, eine Struktur im Spiel zu erkennen. Um programmieren zu können, ist es wichtig zu wissen, was im Code enthalten sein muss. Indem die Schüler ihren eigenen Spielplan erstellen und über die kleinen Schritte nachdenken können, wird das Programmieren einfacher.

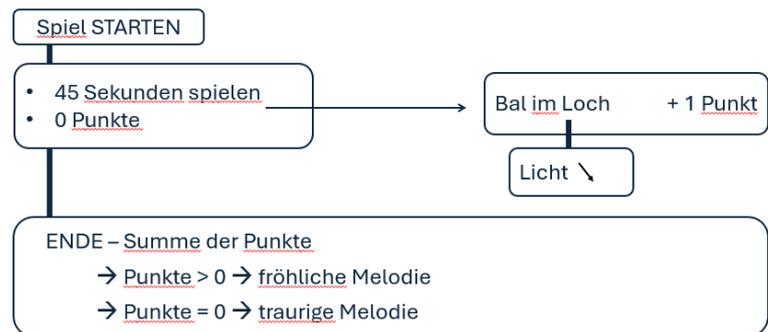


Abbildung 9: Beispiel für einen möglichen Spielplan

Lichtsensor-Auslesung

Um zu erkennen, wenn sich ein Ball im Loch befindet, wird ein Lichtsensor verwendet. Wenn die Kugel das Loch schließt, in dem sich der Sensor befindet, nimmt die Lichtintensität ab.

Schieben Sie den BBC micro:bit in den Grove Shield. Verbinden Sie den Sensor mit dem entsprechenden Kabel mit dem Pin 0 des Grove Shield (Abbildung 10). Verbinden Sie den micro:bit mit dem entsprechenden Kabel (Abbildung 11).

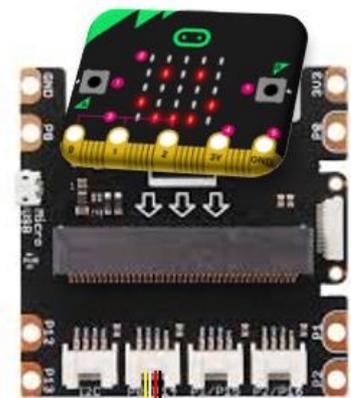


Abbildung 10: Anschlussplan



Abbildung 11: Mini-zu-Micro-USB-Kabel

Schreiben Sie den folgenden Code, um einen Messwert vom Lichtsensor zu erhalten.

Start-Python-Umgebung

Gehen Sie zum Startbildschirm und wählen Sie 'Neu'. Hier können Sie 'Python hinzufügen' wählen. Wählen Sie dann 'Neu'. Wählen Sie einen Namen für Ihr Programm, ohne ein Leerzeichen zu verwenden. Über die Menütaste können Sie die erforderlichen Funktionen hinzufügen. Es ist auch jederzeit möglich, den Code selbst einzugeben.

Schreiben von Code für eine einzelne Messung

```
from microbit import*  
print("Lichtmenge =", grove.read_lightlevel(pin0))
```

Dadurch ist es möglich, den Bibliothekscode für micro:bit zu verwenden.

Wählen Sie im Menü "Weitere Module", dort können Sie "BBC micro:bit" auswählen. Die erste Option ist "from microbit import*".

Auf dem Bildschirm des Handheld möchten wir die Lichtmenge anzeigen.

Wählen Sie im Menü "Eingebaute Funktionen" und gehen Sie zu "I/O". Die erste Option ist print(). Sie müssen den Text " Lichtmenge = " selbst mit den Buchstaben unten eingeben. Die Zeichen " zeigen an, dass es als Text gedruckt wird. Geben Sie danach ein "," ein und wählen Sie die Grove-Geräte für den micro:bit aus. Wählen Sie bei der Eingabe den Lichtstärkesensor (lightlevel) und geben Sie dann "pin0" ein.

Sobald der Code geschrieben ist, kann er ausgeführt werden. Wählen Sie dazu "Starten" im Menü. Sie können auch die Tasten 'Ctrl'+ 'R' drücken. Eine Shell wird auf einer neuen Seite geöffnet. Mit 'Ctrl' + Pfeil nach links oder rechts können Sie zwischen den Seiten wechseln. Wenn Sie also etwas an Ihrem Code ändern wollen, müssen Sie die Änderung auf der vorherigen Seite vornehmen und das Programm erneut in der Shell ausführen.

Code-Schreibmessung pro Sekunde

Die Schüler werden sehen, dass nur ein Wert auf dem Handheld gedruckt ist. Das Programm muss erneut ausgeführt werden, um eine neue Messung zu erhalten.

Um kontinuierlich neue Werte zu drucken, kann eine 'while'-Schleife verwendet werden, die die Messung und den Druck so lange anfordert, bis 'esc' gedrückt wird.

```

from microbit import*

while get_key() != "esc":
    print("Lichtmenge =", grove.read_lightlevel(pin0))
    sleep(1000)

```

Wählen Sie im Menü "Weitere Module" die Option "TI System". Und dort finden Sie als Option "while get_key()="esc" ". Danach sehen Sie, dass der Code eingerückt ist. Alles auf dieser Registerkarte gehört also zu dieser while-Schleife.

Diese Schleife wird so lange ausgeführt, bis 'esc' gedrückt wird. Wir möchten, dass jede Sekunde eine Messung auf dem Handheld gedruckt wird. Bei der Eingabe einer Zeit für den micro:bit muss die Einheit in Millisekunden berücksichtigt werden.

Wählen Sie im Menü "Eingebaute Funktionen" und gehen Sie zu "I/O". Die erste Option ist print(). Sie müssen den Text " Lichtmenge =" selbst eingeben, indem Sie die Buchstaben am unteren Rand verwenden. Im der microbit-module wählen Sie "Grove Devices", dann "Input" und dann "var=read_lightlevel(pin)". Für den Pin wählen Sie "pin0".

Diesen finden Sie im BBC microbit-Modul unter "Commands". Hier wählen Sie "sleep()".

Die Schüler können nun die Werte des Lichtsensors ablesen und ausdrucken lassen. Dies sollte für das Experiment verwendet werden, das die Schüler im Unterricht durchführen werden.

Lektion 2: Experimentieren und Programmieren

Experiment



Abbildung 13: Sensor und Milchkappe

Der Sensor muss im Loch befestigt werden, aber das Kabel ragt noch heraus. Am einfachsten ist es, den Sensor an einem Verschluss einer Plastikmilchflasche zu kleben (Abbildung 13). Diesen stecken wir später unter das Loch, wo der Ball liegen kann (Abbildung 17). Die Schüler sollten sich einen Moment Zeit nehmen und ausprobieren, wie sich der Lichtwert verändert, wenn man die Milchkappe abdeckt (Abbildung 12). Denn eine dunkle Kappe lässt weniger Licht durch als eine helle Kappe.

Im Code für das Spiel müssen sie ihren experimentellen Wert eingeben, den der Sensor beim Abdecken der Kappe anzeigt. Das Limit kann beispielsweise 15 betragen. Wird eine Lichtstärke von weniger als 15 erreicht, wird ein Punkt vergeben.



Abbildung 12: verdeckter Sensor

Programm

Laut Spielplan (Abbildung 9) kann der Code nun erstellt werden. Die nächsten Schritte müssen programmiert werden.

1. Die Zeit muss gestartet werden, da das Spiel nur 45 Sekunden dauern kann.
2. Wenn die Lichtintensität unter einen bestimmten Wert fällt, wird ein Punkt hinzugefügt.
3. Nach 45 Sekunden wird geprüft, ob die Punkte größer als Null sind oder nicht, wobei eine fröhliche oder traurige Musik erklingt.

Zeitmodul einfügen und Variablen definieren

```
from microbit import*  
from time import *  
  
Punkte=0  
t0=clock()  
Zeit=0
```

Auch hier wird das Microbit-Modul wie in Lektion 1 erwähnt importiert.

Durch Einfügen von "from time import*" ist es möglich, die Zeit im geschriebenen Programm zu verwenden.

Wählen Sie im Menü "Weitere Module" die Option "Time". Die erste Option ist "from time import".*

Zuerst müssen wir definieren, dass es Punkte im Spiel gibt und dass das Spiel mit 0 Punkten beginnt. Außerdem beginnt beim Start die Zeit zu laufen.

Geben Sie "Punkte = 0", "t0 = clock" und "Zeit = 0" ein. Clock() finden Sie auch im Menü "Weitere Module", wo Sie "Time" auswählen können und auch clock() sehen.

Dadurch können wir Variablen erstellen, die wir später im Code aufrufen können.

```
print("Spiel an")
```

Wir möchten auch den Text "Spiel an" auf den Handheld drucken.

Wählen Sie im Menü "Eingebaute Funktionen" und gehen Sie zu "I/O". Die erste Option ist print(). Sie müssen den Text "Spiel an" selbst eingeben, indem Sie die Buchstaben am unteren Rand verwenden.

Ablauf des Programmierspiels

```
while Zeit <25:  
  display.show(Punkte),  
  Zeit=clock()-t0
```

Wir erstellen eine neue while-Schleife, die so lange ausgeführt wird, wie die Zeit weniger als 25 Sekunden beträgt. Wir definieren die Zeit als die aktuelle Zeit abzüglich der Startzeit. Wenn sie länger als 25 Sekunden dauert, wird diese Schleife beendet. Während der Spielzeit wollen wir die Anzahl der Punkte auf dem Bildschirm des micro:bit sehen.

Wählen Sie "Eingebaute Funktionen" aus dem Menü und gehen Sie zur Option "Steuerung" und wählen Sie "While". Wenn Sie die var-Taste drücken, erscheint eine Dropdown-Liste und Sie wählen "Zeit". Sie geben selbst "<25" ein.

Um die Punkte auf dem Display des micro:bit anzuzeigen, gehen Sie zu "Display" im BBC microbit-Modul und wählen Sie "show(text)". Zwischen denn "(" wählen Sie mit der var-Taste wieder die variablen Punkte. Die Regel "Zeit =..." können Sie analog mit der VAR-Taste bauen.

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

Wenn die Lichtintensität unter den Wert von 15 fällt (bestimmt durch das durchgeführte Experiment), muss ein Punkt hinzugefügt werden. Gleichzeitig wollen wir eine schöne Musik hören, damit der Spieler auch eine auditive Rückmeldung über den erreichten Punkt erhält. Dazu wird eine if-Funktion verwendet.

If ist in den "Eingebauten Funktionen" bei "Steuerung" zu finden. Danach können Sie den Sensor wieder hinzufügen. Im der microbit-module wählen Sie "Grove Devices", dann "Input" und dan "var=read_lightlevel(pin)". Für den Pin wählen Sie "pin0". Geben Sie danach selbst <5 ein.

Da die folgenden Zeilen noch zu dieser if-Funktion gehören, werden zwei zusätzliche Blöcke eingerückt. Wenn diese nicht automatisch hinzugefügt werden, können Sie dies auch mit der Tabulatortaste tun.

Die Punkte werden mit der var-Taste neu aufgebaut. Im BBC microbit-Modul können Sie unter "Music" mit "play(song)" eine Melodie auswählen.

Ende des Spiels

```
if Punkte>0:
    display.show(Punkte)
    music.play(music.ENTERTAINER)
else:
    music.play(music.DADADADUM)
print("Spiel beendet ->Punktzahl =",Punkte)
display.clear()
```

Wenn die 25-Sekunden-Schleife vorbei ist, möchten wir, dass die Punkte sichtbar werden. Wenn die Punkte größer als Null sind, wollen wir, dass eine fröhliche Musik gespielt wird, ansonsten wollen wir eine traurige Musik. Dies kann mit einem "if... else-Funktion. Da die 'while'-Schleife nun beendet ist, beginnt das "if... else-Funktion ohne die Blöcke.

"If... else" finden Sie in "Eingebaute Funktionen" bei "Steuerung". Wählen Sie nach "if" mit der var-taste wieder "Punkte" und geben danach selbst ">0" ein. Im BBC microbit-Modul können Sie "show(text)" wieder hinzufügen und in Klammern "Punkte" mit der var-Taste auswählen. Mit "play(song)" im micro:bit-Modul können Sie auch eine Melodie hinzufügen.

Nach "else" fügen Sie eine traurige Melodie ein, weil dann null Punkte erreicht sind. Mit "Print" lassen wir die Punkte auf dem Handheld erscheinen. Außerdem wollen wir das Display des micro:bit löschen. Dieser letzte Schritt ist für den Betrieb des Spiels nicht notwendig.

Wenn Sie für sich selbst einen Kommentar, zu dem nicht ausgeführten Code eingeben möchten, können Sie diesen nach einem #-Zeichen notieren.



```
1.1 1.2 mik_en_rol_v1 RAD 11/26
*fff.py
from microbit import*
from time import *

Punkte=0
t0=clock()
Zeit=0

print("Spiel an")

while Zeit <25:
    display.show(Punkte)
    Zeit=clock()-t0
    if grove.read_lightlevel(pin0)<15:
        Punkte = Punkte+1
        music.play(music.JUMP_UP)
    if Punkte>0:
        display.show(Punkte)
        music.play(music.ENTERTAINER)
    else:
        music.play(music.DADADADUM)
print("Spiel beendet ->Punktzahl =",Punkte)
display.clear()
```

Lektion 3: Entwerfen der Box und Spielen

Nicht alle Schüler werden ihren Code nach Lektion 2 fertig programmiert haben. Je nachdem, wie viel Zeit für den Bau der Box aufgewendet wurde, kann Zeit für die weitere Programmierung vorgesehen werden.

Kasten bauen

Dieses Spiel wird wie folgt aufgebaut für eine Holzkugel mit einem Durchmesser von 30 mm:

1. Nehmen Sie die Holzkiste und bohren Sie an einer beliebigen Stelle ein 10 mm Loch (Abbildung 14).
2. Bohren Sie mit einem 14-mm-Senkkopfbohrer (Abbildung 16).
3. Markieren Sie den Startpunkt im Holztablett (Abbildung 15).
4. Kleben Sie die Kappe mit dem Lichtsensor mit Klebeband unter das Loch (Abbildung 17).
5. Schließen Sie den Sensor an Pin0 des Grove Shield an.
6. Schieben Sie den micro:bit in den Grove Shield.
7. Befestigen Sie den micro:bit am Handheld.

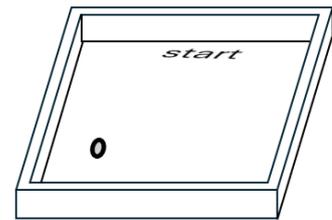


Abbildung 14: Holztablett mit Bohrloch und Startplatz



Abbildung 15: Start-Anzeige



Abbildung 16: Senkkopfbohrloch



Abbildung 17: Sensor steckt unter dem Loch fest

Spiel spielen

Das Spiel ist jetzt spielbereit. Führen Sie den Code aus und spielen Sie.

Mögliche Erweiterungen

Mehrere Lichtsensoren.

Anstatt nur ein Loch zu bohren, ist es auch möglich, mehrere Löcher zu bohren. Sie können bis zu drei Lichtsensoren an das Grove Shield anschließen.

In der 'while'-Schleife kann eine 'if'-Funktion hinzugefügt werden (Abbildung 18).

Der zweite Lichtsensor wird dann mit dem zweiten Pin (Abbildung 19) des Grove Shield verlinkt. Passen Sie dies auch im Code an.

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

Abbildung 18: Code mit zwei Lichtsensoren

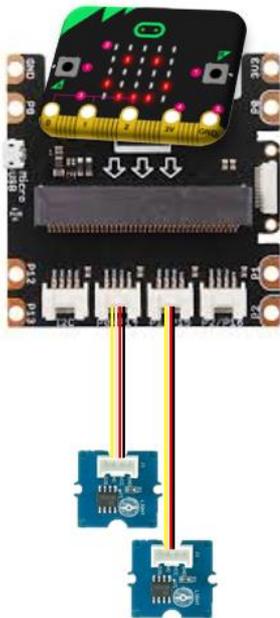


Abbildung 19: Anschluss eines zweiten Lichtsensors

Hinzufügen von Hindernissen

Um das Spiel attraktiver zu machen, können Hindernisse hinzugefügt werden. Es können Nägel gehämmert oder Hobelleisten (12x12 mm) auf dem Parcours geklebt werden.



Abbildung 20: Hindernisse mit Hobelleisten

Hinzufügen von Extras auf dem micro:bit-Display

Wenn ein Punkt verdient wird, wird ein Musikstück gespielt. Es ist auch möglich, gleichzeitig ein Bild auf dem micro:bit erscheinen zu lassen. Der Code dafür kann mit der Funktion 'if' hinzugefügt werden (Abbildung 21). Weil Sie direkt nach dem Punkt wissen möchten, wie hoch der Punktestand ist, müssen Sie auch das Display löschen, damit das Bild wieder verschwindet.

```
while Zeit < 25:  
  display.show(Punkte)  
  Zeit = clock() - t0  
  if grove.read_lightlevel(pin0) < 15:  
    Punkte = Punkte + 1  
    display.show(Image.YES)  
    music.play(music.JUMP_UP)  
    display.clear()  
  if grove.read_lightlevel(pin1) < 15:  
    Punkte = Punkte + 1  
    display.show(Image.YES)  
    music.play(music.JUMP_UP)  
    display.clear()
```

Abbildung 21: Bild auf micro:bit anzeigen

Unterschiedlich in Punkten

Wird eine Lücke mit Hindernissen erschwert, können zwei Punkte vergeben werden. Der Code lautet dann 'Punkte = Punkte + 2'. Die Spieler können wählen, ob sie den Ball innerhalb ihrer Spielzeit auf den Ein- oder Zwei-Punkte-Abstand zielen möchten.

Spiel

In Abbildung 22 wird ein fertiges Spiel mit Hobelleisten als Hindernisse und zwei Lichtsensoren gezeigt. Der micro:bit wird seitlich mit Klebeband an der Box befestigt. Am einfachsten lässt sich das Spiel im Sitzen spielen, weil der micro:bit und der Handheld noch mit Kabeln an der Unterseite verbunden sind.

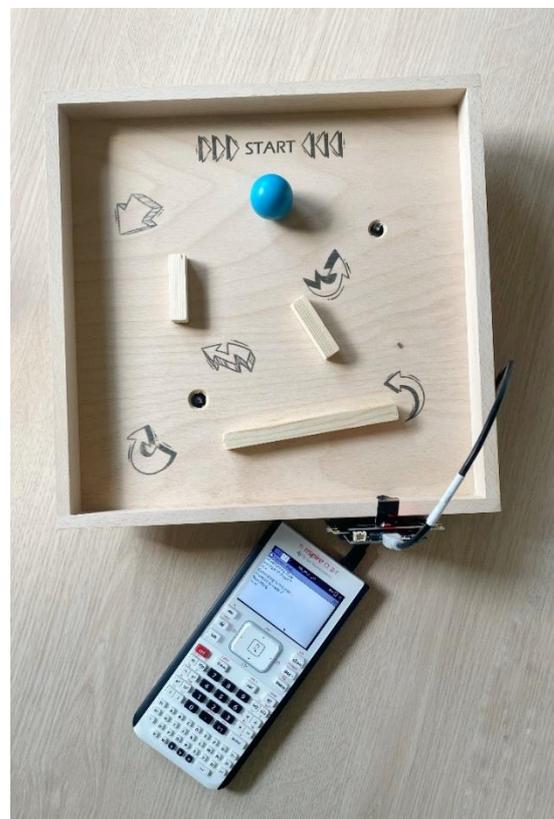


Abbildung 22: Das fertige Spiel