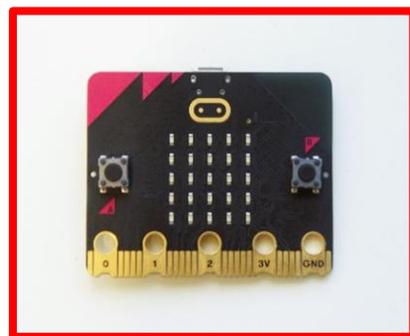
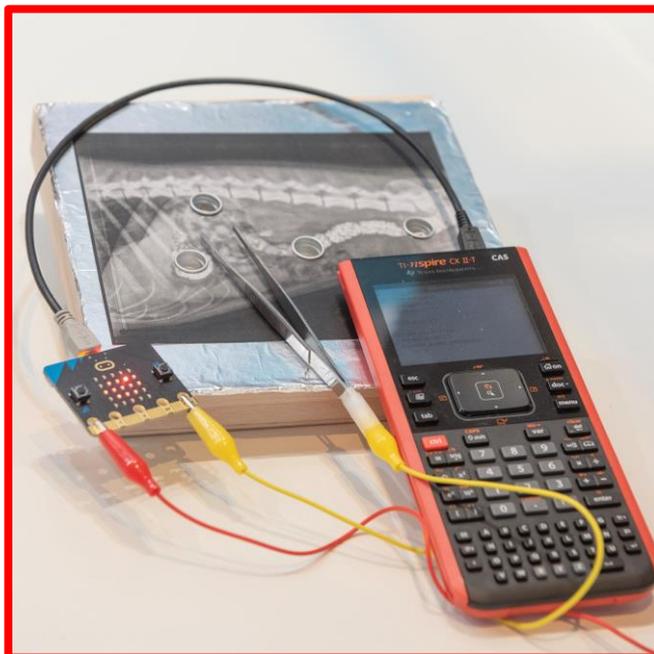




# Operationsspiel mit dem TI-Nspire und BBC micro:bit

Lehrerbündel

*Ann-Kathrin Coenen  
& Natalie Dirckx*

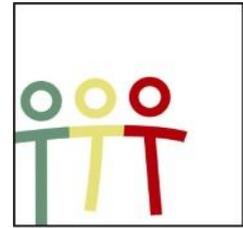


## Inhaltsangabe

<b>Inhaltsangabe</b> .....	<b>2</b>
<b>T<sup>3</sup>-Flandern en T<sup>3</sup>-Niederlande</b> .....	<b>3</b>
<b>Einführung</b> .....	<b>3</b>
<b>Einführung in den BBC micro:bit</b> .....	<b>4</b>
<b>Programmieren mit dem TI-Nspire CX II</b> .....	<b>5</b>
<b>Zubehör</b> .....	<b>6</b>
<b>Methode - das Spiel aufbauen</b> .....	<b>7</b>
<i>Verbindung zwischen TI-Nspire und micro:bit</i> .....	<i>8</i>
<b>Methode - Programmierung des BBC micro:bit</b> .....	<b>9</b>
<i>Spielplan</i> .....	<i>9</i>
<i>Code-Erklärung</i> .....	<i>9</i>
Starten der Python-Umgebung.....	9
Spiel starten.....	10
Variablen Spiel .....	11
Definition des Spiels .....	11
Ende des Spiels .....	12

## T<sup>3</sup>-Flandern en T<sup>3</sup>-Niederlande

Ann-Kathrin Coenen und Natalie Dirckx sind Lehrerinnen für Naturwissenschaften am Agnetencollege Peer. Sie gehören zu dem Lehrernetzwerk von T<sup>3</sup> Flandern, das eng mit den Niederlanden zusammenarbeitet. T<sup>3</sup> 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.



**T<sup>3</sup> VLAANDEREN**

Abbildung 1:

[www.t3vlaanderen.be](http://www.t3vlaanderen.be)

## Einführung

Ein Operationsspiel ist ein bekanntes Geschicklichkeitsspiel aus der Kindheit. Ein Patient liegt auf dem Operationstisch und hat verschiedene (Fremd-)Gegenstände verschluckt. In diesem Fall hat Ihr Hund Fremdkörper verschluckt. Sie sind der Tierarzt und werden Ihren Hund operieren. Mit einer Pinzette entfernen Sie die Fremdkörper vorsichtig. Aber seien Sie vorsichtig! Berühre nur den Gegenstand, sonst verlieren Sie Punkte.

Dieses Projekt kann in einen gestalterischen Teil und einen programmtechnischen Teil unterteilt werden. Der Bau eines Operationsspiels und die Programmierung des Spiels mit Python auf dem TI-Nspire passt ideal in den MINT-, Biologie-, Physik- oder Ingenieurunterricht.

Es kann eine Verbindung zum Unterricht über elektrische Systeme in der Physik hergestellt werden. Die Schüler lernen das Spiel in kleine Teile zu zerlegen um den Code zu schreiben. Und das Spiel kann verwendet werden um die Konzepte von Spannung, Stromstärke und Stromgefühl auf dieses Projekt anzuwenden. Darüber hinaus testet das Spiel auch die Auge-Hand-Koordination und die motorischen Fähigkeiten der Spieler.

## Einführung in den 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 Mikrophon 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.

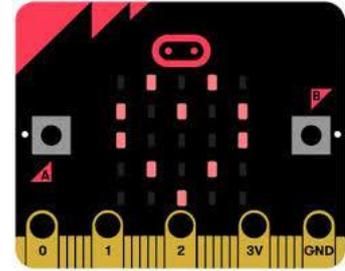


Abbildung 2: BBC micro:bit

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 4) geschrieben. In diesem Bündel wurde der Nspire für die Durchführung dieses Projekts gewählt.



Abbildung 3: Python Programmiersprache

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 4: TI-Nspire CX II

## Programmieren 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 5).

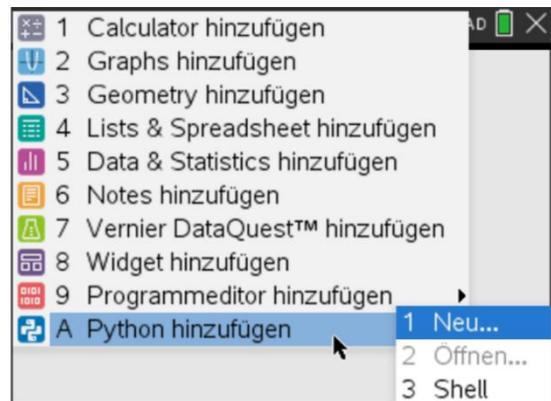


Abbildung 5: erstellen einer neuen Python-Datei auf dem TI-Nspire

Über die Menütaste können Sie teilweise vorgegebene Codestücke hinzufügen.

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 6) 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.

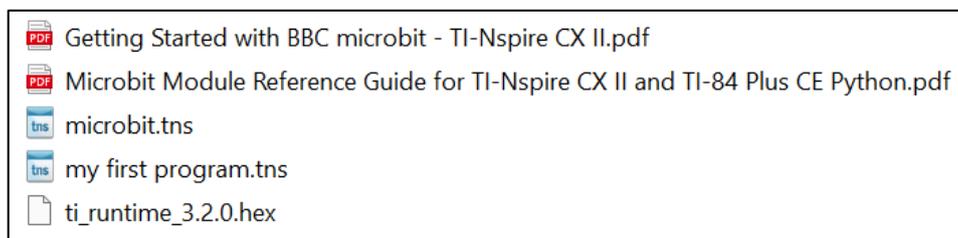


Abbildung 6: microbit.tns enthält das Modul für den Handheld, ti\_runtime.hex ist das Modul für den Microbit

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.

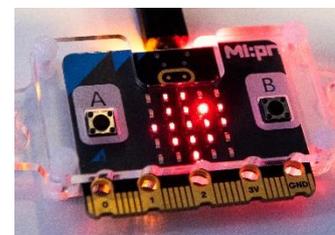
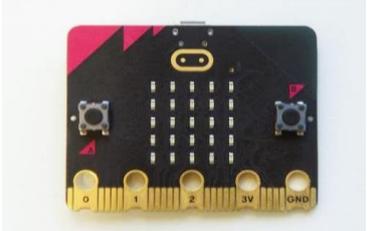
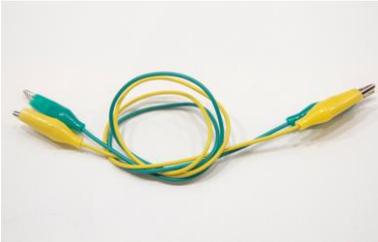
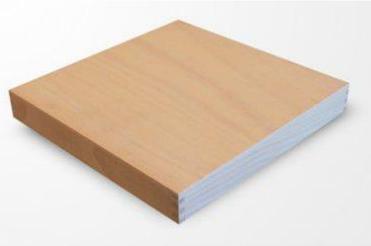


Abbildung 7: Texas Instruments Logo auf dem Display micro:bit

## Zubehör

Die folgenden Materialien werden für ein Operationsspiel benötigt:

- TI-Nspire CX II
- Micro:bit v2
- Mini-USB nach Micro-USB Kabel
- Alufolie
- Flaschenverschlüsse aus Aluminium/Weißblech
- Pappe (ca. 2 mm dick) mit den Maßen 15 cm X 15 cm
- Pinzette
- 2 Krokodilklemmen
- Kleine (nichtleitende) Gegenstände aus Kunststoff (sie müssen in die Flaschenverschlüsse passen, z. B. Anhänger für Armbänder)
- Ein Bild des Patienten
- Holzkiste mit einer Mindestgröße von 15 cm X 15 cm 'Art Panel' ([www.hobbyshop-online.nl](http://www.hobbyshop-online.nl))

		
BBC micro:bit	Krokodilklemmen	Kleine Gegenstände
		
Holzkiste (Art Panel)	Flaschenverschlüsse aus Aluminium/Weißblech	Pappe und Alufolie

## Methode - das Spiel aufbauen

Der Aufbau der Nervenspirale erfolgt wie folgt:

1. Mache ein Foto von einem Patienten und bestimme, wo die Fremdkörper platziert werden sollen. Zeichne an diesen Stellen einen Kreis in der Größe der Aluminiumflaschenverschlüsse und schneide sie aus.
2. Nehmen Sie die Pappe. Falls noch nicht geschehen, schneiden Sie ein Quadrat von 15 x 15 cm aus. Legen Sie das Bild auf die Pappe und zeichnen Sie die gleichen Kreise auf die Tafel. Nehmen Sie das Bild weg und schneiden Sie die Kreise aus (Abbildung 8).
3. Bedecke die Pappe auf der Vorder- und Rückseite vollständig mit Alufolie und klebe ihn fest. Schneide dann die Kreise ebenfalls aus der Folie aus (Abbildung 9).
4. Klebe das Bild so über die Alufolie auf der Pappe, dass die Löcher übereinstimmen.
5. Stecken Sie einen Flaschenverschluss in jedes Loch (Abbildung 10).
6. Legen Sie in jeden Flaschendeckel einen kleinen Gegenstand als Fremdkörper (Abbildung 12).
7. Befestige eine Krokodilklemme an der Pinzette und verbinde sie mit Pin 0 des BBC micro:bit. Diese Klemme kann sich von der Pinzette lösen, daher ist es am besten, sie mit Klebeband zu befestigen.
8. Befestigen Sie eine weitere Krokodilklemme auf der aluminiumbeschichtete Kartonplatte und verbinden Sie sie mit dem Erdungsstift (GND) des BBC micro:bit (Abbildung 11).
9. Nimm die Holzbox und drehe sie mit der hohlen Seite nach oben. Legen Sie das Operationsspiel darauf.



Abbildung 8: pappe mit ausgeschnittenen Kreisen



Abbildung 9: aluminiumbeschichtete Kartonplatte

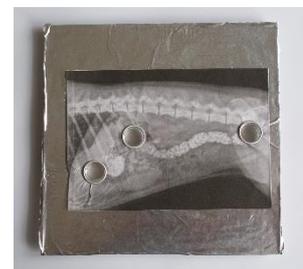


Abbildung 10: komplettes Spielbrett

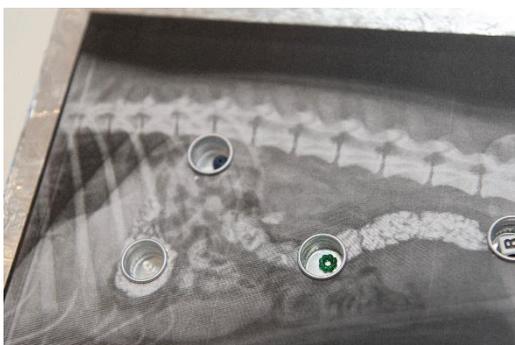


Abbildung 12: platzierung der Fremdkörper

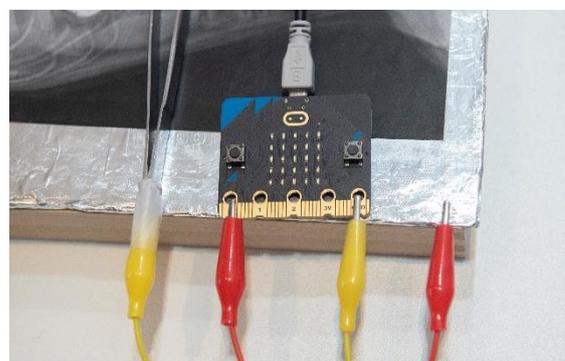


Abbildung 11: die Krokodilklemme des Spiels wird an Pin 0 und die Klemme der Pinzette an GND angeschlossen

## Verbindung zwischen TI-Nspire und micro:bit

Wenn die Form fertig ist, kann das micro:bit mit dem Mini-USB zu Micro-USB Kabel gemäß Abbildung 13 angeschlossen werden. Abbildung 14 zeigt eine vollständige Einrichtung.



Abbildung 13: Mini-zu-Micro-USB-Kabel

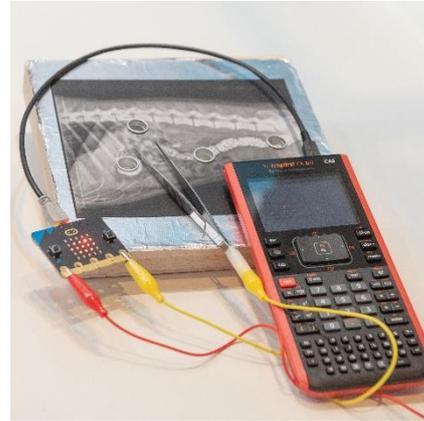


Abbildung 14: ein fertiges Operationsspiel

## Methode - Programmierung des BBC micro:bit

### Spielplan

Um Schüler, die noch nie programmiert haben, in den Lernprozess einzuführen, unterteilen wir das Spiel in kleine Schritte.

Das Spiel sollte ein Signal zum Starten erhalten. Mit dem Start sollte festgelegt werden wie viele Punkte ein Spieler von Anfang an bekommt. In diesem Fall gibt es zu Beginn 3 Punkte. Wenn man den Rand des Spielbretts oder den Flaschendeckel berührt, muss man einen Punkt von den verbleibenden Punkten abziehen.

Wenn man keine Punkte mehr hat, hat man verloren.

Der micro:bit hat ein Display, auf dem Sie Zahlen anzeigen können, und der micro:bit kann auch Töne abspielen. Das kann man mit dem Berühren des Randes oder des Flaschenverschlusses kombinieren.

Wir erstellen ein Beispieldiagramm (Abbildung 15), um die Programmierung zu erleichtern.

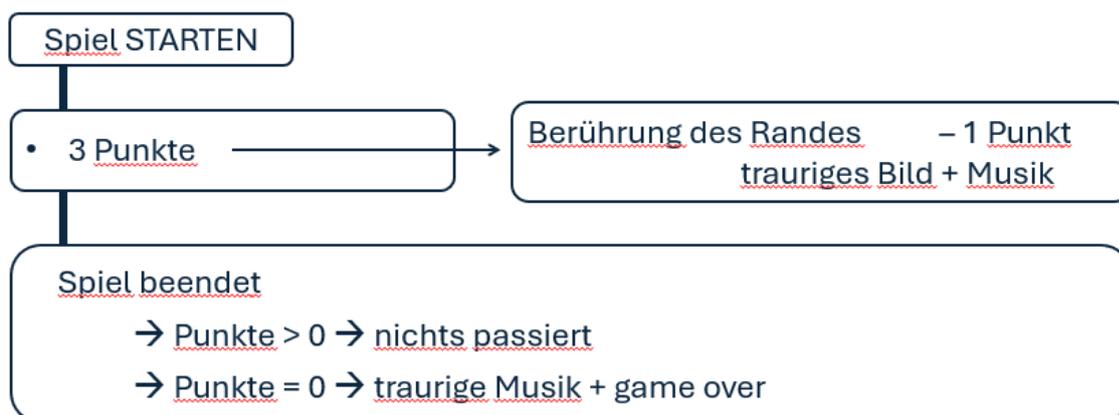


Abbildung 15: Beispiel für einen möglichen Spielplan

### Code-Erklärung

#### Starten der 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.

## Spiel starten

```
from microbit import *
```

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\*'.  
Wählen Sie im Menü 'Eingebaute Funktionen' und gehen Sie zu 'I/O'. Die erste Option ist print(). Sie müssen den Text "Drücken Sie A" selbst mit den Buchstaben am unteren Rand eingeben.

```
print("drücken Sie A")
```

Auf dem Bildschirm des Handheld soll "drücken Sie A" erscheinen.

Wählen Sie im Menü 'Eingebaute Funktionen' und gehen Sie zu 'I/O'. Die erste Option ist print(). Sie müssen den Text "Drücken Sie A" selbst mit den Buchstaben am unteren Rand eingeben.

```
while get_key() != "esc":  
    display.scroll("?")  
    if button_a.was_pressed():  
        break
```

Wir wollen, dass der Code so lange läuft, bis "esc" gedrückt wird. Hierfür verwenden wir eine while-Schleife. Auf dem Bildschirm des micro:bit soll ein "?"-Scroll angezeigt werden. Und wenn die Taste A auf dem micro:bit gedrückt wird, stoppt es und das Spiel beginnt. Der Code fährt mit dem Spiel fort. Dieses abgebildete Stück Code kann auch weggelassen werden, aber dann startet das Spiel sofort, wenn man das Programm auf dem Handheld laufen lässt. Also baut man jetzt ein zusätzliches Startsignal (Knopf A) auf dem micro:bit ein.

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.

Wählen Sie im Menü 'Weitere Module' die Option 'BBC micro:bit', dann 'Display' und dann 'scroll(text of number)'. Sie müssen noch das "?" zwischen den () eingeben. Drücken Sie auf 'ENTER'. Wählen Sie im Menü 'Eingebaute Funktionen' und gehen Sie zu 'Steuerung'. Die erste Option ist print(). Gehen Sie wieder zum BBC micro:bit-Menü und wählen Sie 'Buttons and Logo Touch', dann 'button a' und dann 'was\_pressed ()'. Drücken Sie auf 'ENTER' nach dem ':' und schreiben Sie "break".

## Variablen Spiel

```
print("Spiel an")  
Punkte = 3
```

Wenn auf dem micro:bit die Taste A gedrückt wird, soll das Spiel starten. Auf dem Handheld wird dann "Spiel an" angezeigt. Zunächst müssen wir noch festlegen, dass das Spiel mit 3 Punkte beginnt.

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.

Geben Sie selbst Punkte = 3 ein. Damit werden Variablen erstellt, die wir später im Code aufrufen können.

## Definition des Spiels

```
while Punkte>0:  
  display.show(Punkte)  
  if pin0.is_touched():  
    Punkte = Punkte - 1  
    display.show(Image.SAD)  
    music.play(music.JUMP_DOWN )  
    sleep(1)
```

Wir erstellen eine neue while-Schleife, die so lange läuft, wie die Punkte nicht Null sind. Auf dem Display des micro:bit wollen wir die Punkte sehen.

In der if-Funktion programmieren wir, dass die Punkte jedes Mal um 1 Wert abnehmen sollen, wenn der Pin 0 (Stromkreis geschlossen) berührt wird. Hier verlinken wir ein weiteres Bild und eine Musik auf dem micro:bit.

Danach wird 1 Sekunde Ruhe (Schlaf) eingebaut, bevor der Code weiterläuft.

Wählen Sie im Menü 'Integrierte Funktionen'. Unter der zweiten Option 'Steuerung' wählen Sie 'while'. Wenn Sie die var-Taste drücken, erscheint eine Auswahlliste und Sie wählen 'Punkte' und geben dann '>0' ein. Um die Punkte auf dem Display des micro:bit anzuzeigen, gehen Sie wieder zu 'Display' im Microbit-Modul und wählen Sie 'show(text)'. Zwischen dem '(' wählen Sie mit der var-Taste wieder die variable Punkte.

If ist in den eingebauten Funktionen bei 'Steuerung' zu finden. Der 'pin0.is\_touched' sollte nach diesem If platziert werden. Gehen Sie zum BBC Microbit-Modul und wählen Sie unter 'I/O Pins' den 'Digital'. Darunter sehen Sie 'pin.is touched()'. Wählen Sie dann pin0.

Die Punkte werden mit der var-Taste neu aufgebaut. Im BBC Microbit-Modul können Sie unter 'Display' mit 'show(image)' ein Bild und unter 'Music' mit 'play(song)' eine Melodie auswählen.



Danach bauen Sie ein 'sleep(1)' ein. Diesen finden Sie im BBC microbit-Modul unter 'Commands'.

### Ende des Spiels

```
else:  
    music.play(music.DADADADUM)
```

Wenn die Punkte gleich Null sind, wird eine traurige Musik gespielt.

'else' finden Sie bei den eingebauten Funktionen unter 'Steuerung'. Unter der var-Taste finden Sie die Punkte zurück.

Im BBC Microbit-Modul finden Sie wieder unter 'Display' und 'Music' die Bilder und Melodien.

```
print("Spiel beendet")  
display.scroll("Game over")  
  
display.clear()
```

Der Handheld zeigt die Meldung "Spiel beendet" und der micro:bit zeigt "Game over" an.

Wählen Sie im Menü 'Eingebaute Funktionen' und gehen Sie zu 'I/O'. Wählen Sie 'print()' und geben Sie den Text selbst ein. Der Code für das Display befindet sich im BBC Microbit-Modul unter 'Display' und dann 'scroll (text of number)'.

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

In Abbildung 16 sehen Sie den Code in seiner Gesamtheit.

```
1.1 *Operation...iel RAD 11/28
*Operationsspiel.py
from microbit import *

print("drücken Sie A")

while get_key() != "esc":
    display.scroll("?")
    if button_a.was_pressed():
        break

print("Spiel an")

Punkte = 3

while Punkte > 0:
    display.show(Punkte)
    if pin0.is_touched():
        Punkte = Punkte - 1
        display.show(Image.SAD)
        music.play(music.JUMP_DOWN)
        sleep(1)

else:
    music.play(music.DADADADUM)

print("Spiel beendet")
display.scroll("Game over")

display.clear()
```

Abbildung 16: Vollständiger Code

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.