



Agnetencollege  
Peer

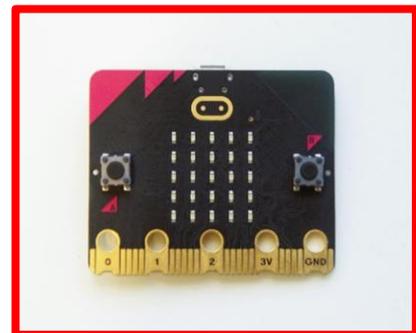
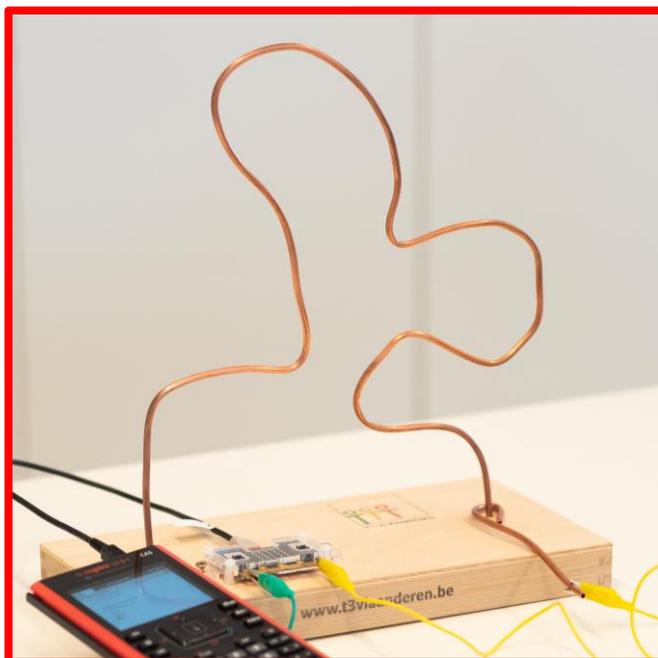
MINT-Projekt



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

Lehrerbündel

*Ann-Kathrin Coenen  
& Natalie Dirckx*

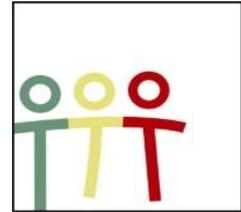


# Inhaltsangabe

<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>Erweiterung um einen zusätzlichen Pin</i> .....	9
<i>Verbindung zwischen TI-Nspire und micro:bit</i> .....	9
<b>Methode - Programmierung des BBC micro:bit</b> .....	<b>10</b>
<i>Spielplan</i> .....	10
<i>Code-Erklärung</i> .....	10
Starten der Python-Umgebung.....	10
Spiel starten.....	11
Variablen Spiel .....	12
Definition des Spiels .....	12
Ende des Spiels .....	13
<b>Erweiterung zusätzliche Kontaktstelle</b> .....	<b>15</b>

## 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

Eine Nervenspirale oder Kupferspirale ist ein bekanntes Geschicklichkeitsspiel aus der Kindheit. Ein Stock mit einem Ring wird entlang einem Kupferdraht von der einen Seite zur anderen bewegt, ohne den Kupferdraht zu berühren. Wenn der Draht dennoch berührt wird, werden Punkte von der Gesamtpunktzahl abgezogen.

Dieses Projekt kann in einen entwerfenden und einen programmierenden Teil aufgeteilt werden. Der Bau einer Nervenspirale und die Programmierung des Spiels mit Python auf dem TI-Nspire passen ideal in den MINT-, Physik- oder Technikunterricht. Hier kann die Verbindung zum Thema elektrische Systeme in der Physik hergestellt werden. Die Schüler lernen das Spiel in kleine Teile zu zerlegen und so den Code zu schreiben. Außerdem kann das Spiel genutzt werden um die Konzepte von Spannung, Stromstärke und Stromsinn anzuwenden.

## 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 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.

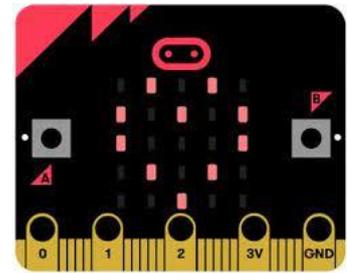


Abbildung 2: BBC Micro:bit

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). Über die Menütaste können Sie teilweise vorgegebene Codestücke hinzufügen.

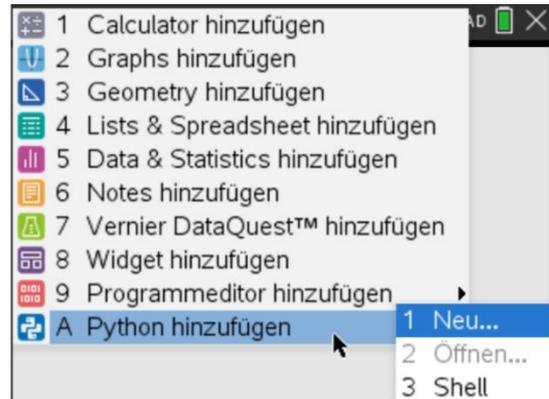


Abbildung 5: erstellen einer neuen Python-Datei auf dem TI- Nspire

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 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.

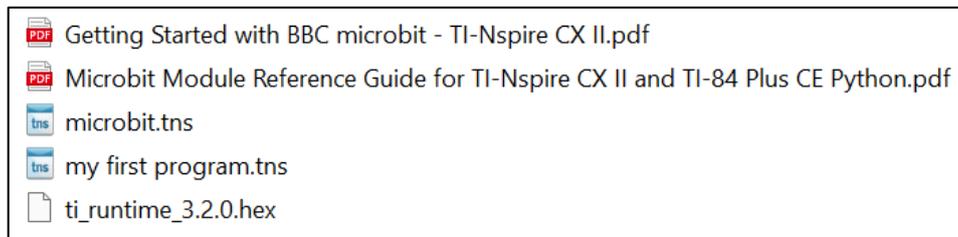


Abbildung 6: microbit.tns enthält das Modul für den Handheld, ti\_runtime.hex ist das Modul für den Microbit

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.

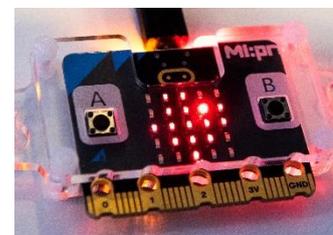
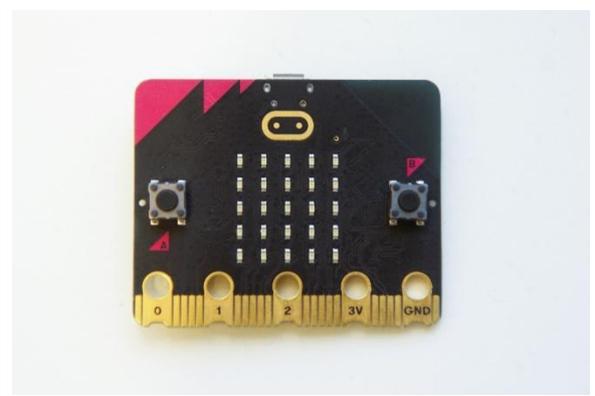
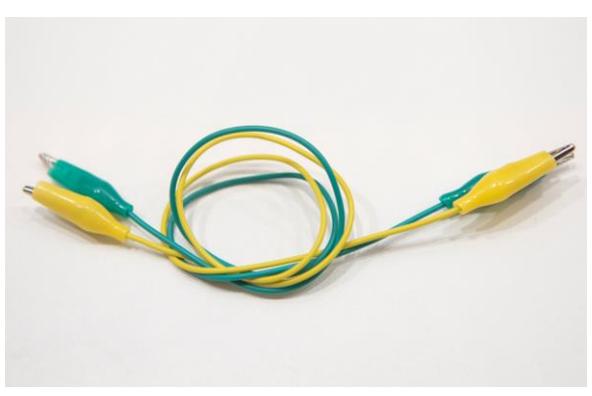


Abbildung 7: Texas Instruments Logo auf dem Display micro:bit

## Zubehör

Die folgenden Materialien werden für eine Nervenspirale benötigt:

- TI-Nspire CX II
- Micro:bit v2
- Mini-USB nach Micro-USB Kabel
- Flexibles Kupferdraht  $\varnothing$  4 mm 70 cm
- Holzkiste mit einer Mindestgröße von 15 cm X 15 cm 'Art Panel' ([www.hobbyshop-online.nl](http://www.hobbyshop-online.nl))
- 2 (3) Krokodilklemmen

	
BBC Micro:bit	Krokodilklemmen
	
Holzkiste (Art Panel)	Weiches Kupferdraht $\varnothing$ 4 mm

## Methode - das Spiel aufbauen

Der Aufbau der Nervenspirale erfolgt wie folgt:

1. Nehmen Sie ein Holzbrett und bohren Sie an beiden Enden ein 4 mm großes Loch (Abbildung 8).
2. Schneide 20 cm von dem Kupferdraht ab. Nimm das Stück Kupferdraht für den Spielstab und biege es an einem Ende zu einem Ring. Dies wird der Spielstab sein, mit dem sich der Spieler auf dem Draht bewegt.
3. Biegen Sie die restlichen 50 cm in eine lustige und aufregende Form für die Spirale.
4. Führen Sie beide Enden der Form durch die Löcher in das Holzbrett (Abbildung 9). Achte darauf, dass unten ein etwa 1 cm langes Stück Kupferdraht herausragt, um deine Krokodilklemmen zu befestigen (Abbildung 11).
5. Befestigen Sie die eine Krokodilklemme an einem der Enden der Form und schließen Sie sie die andere Seite an Pin 0 des BBC micro:bit an (Abbildung 12).
6. Befestigen Sie eine Krokodilklemme am Spielstab und verbinden Sie sie mit dem Erdungstift (GND) des BBC micro:bit (Abbildung 12). Diese Klemme kann sich vom Spielstab lösen, daher ist es am besten, sie mit Klebeband zu befestigen.

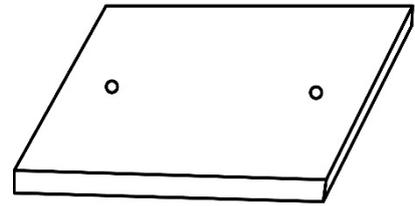


Abbildung 8: Holzbrett mit Löchern

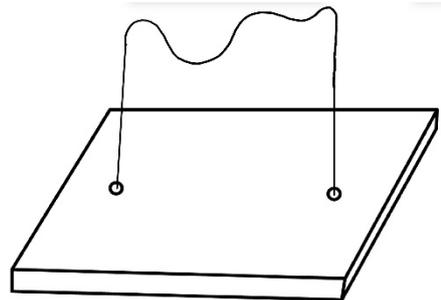


Abbildung 9: Anbringen der Nervenspirale das Regal

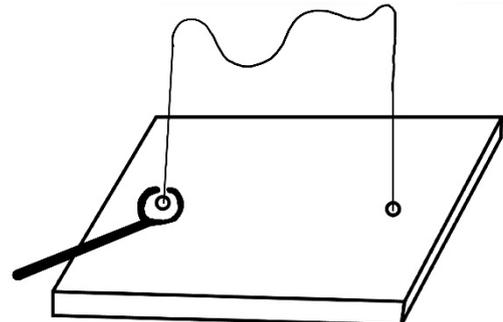


Abbildung 10: Nervenspirale mit Spielstab



Abbildung 11: Die Krokodilklemme kann an der Unterseite des durchbohrten Kupferrohrs befestigt werden.

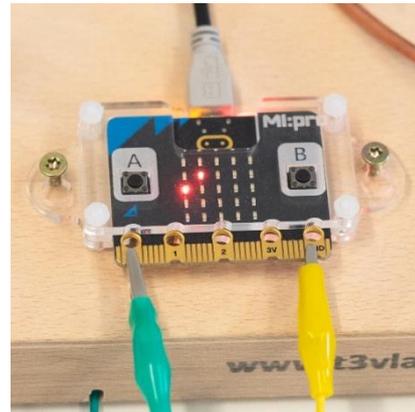


Abbildung 12: Die Krokodilklemme der Spule wird an Pin 0 und die Klemme des Spielstabs an GND angeschlossen.

## Erweiterung um einen zusätzlichen Pin

Das Spiel kann durch einen zusätzlichen Stift erweitert werden. Wird dieser Kontaktpunkt nach dem Abfahren der Spirale mit dem Spielstab berührt, bleibt die Zeit stehen. Für diese Kontaktstelle müssen Sie 0,5 cm von der Spirale entfernt ein zusätzliches 4 mm Loch bohren. Durch dieses Loch steckst du ein kleines Stück Kupferrohr (Abbildung 13), an dessen Unterseite du auch eine Krokodilklemme befestigen kannst (Abbildung 14). Auf der anderen Seite befestigen Sie diese Krokodilklemme an Pin 1 des micro:bit (Abbildung 15).



Abbildung 13: zusätzlicher Stift oben



Abbildung 15: Die Krokodilklemmen der zusätzlichen Kontaktstelle werden an Pin 1 des micro:bit angeschlossen.

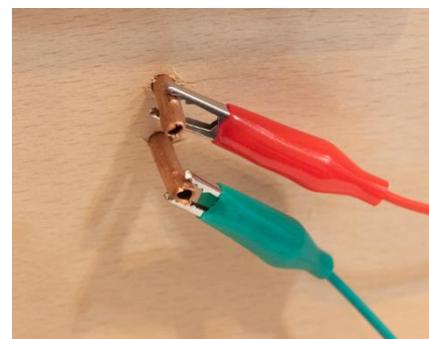


Abbildung 14: zusätzlicher Stift unten

## 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 16 angeschlossen werden. Abbildung 17 zeigt eine vollständige Einrichtung.



Abbildung 16: Mini-zu-Micro-USB-Kabel

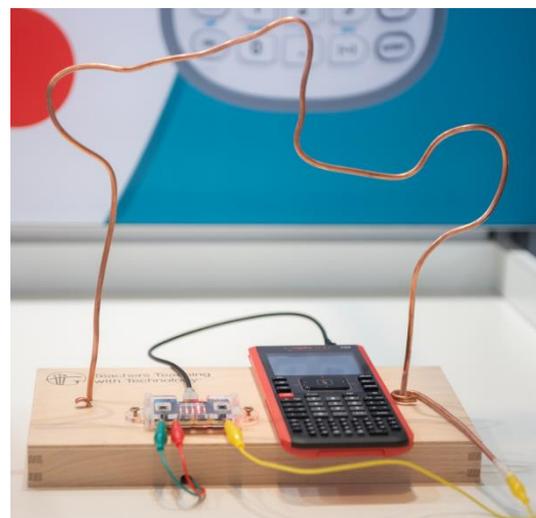


Abbildung 17: eine fertige Nervenspirale

## 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 auch die Zeit zu laufen beginnen. Außerdem sollte festgelegt werden, wie viele Punkte ein Spieler vom Start anbekommt. Beim Berühren der Spirale soll ein Punkt von den restlichen Punkten abgezogen werden.

Wenn Sie keine Punkte mehr haben, haben Sie verloren. Wenn Sie nach Ablauf der Zeit noch Punkte haben, haben Sie gewonnen.

Der micro:bit hat ein Display, auf dem du Zahlen anzeigen kannst, und der micro:bit kann auch Töne abspielen. Das kann man mit dem Gewinnen oder Berühren der Spirale kombinieren.

Wir erstellen ein Beispieldiagramm (Abbildung 18), um die Programmierung zu erleichtern.

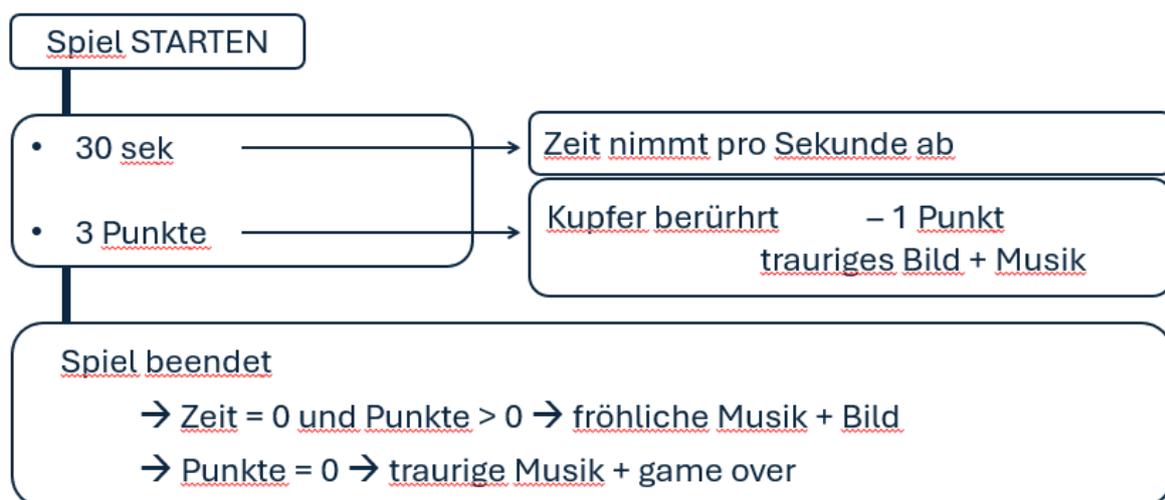


Abbildung 18: Beispiel für einen möglichen Spielplan

### 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.



## Variablen Spiel

```
print("Spiel an")
Punkte = 3
t0 = clock()
Zeit = 0
```

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 und die Uhr ab dem Zeitpunkt läuft, an dem der Knopf A gedrückt wurde.

*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, t0 = clock und Zeit = 0 ein. Clock() finden Sie auch im Menü 'Weitere Module', wo Sie 'Time' auswählen können und auch clock() sehen. Damit werden Variablen erstellt, die wir später im Code aufrufen können.*

## Definition des Spiels

```
while Zeit <30 and Punkte>0:
  ++Zeit = clock()-t0
  ++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 Zeit weniger als 30 Sekunden beträgt und die Punkte nicht Null sind.

Wir definieren Zeit als die aktuelle Zeit minus die Startzeit. Wenn die Zeit 30 Sekunden überschreitet, wird das Spiel angehalten. 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 'Zeit'. '<30' tippen Sie selbst ein, ebenso wie das 'und'. Bei der var-Taste wählen Sie nun 'Punkte' und geben dann '>0' ein.*



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

```
if Punkte > 0:  
  display.show(Image.HAPPY)  
  music.play(music.ENTERTAINER)  
else:  
  music.play(music.DADADADUM)
```

Wenn die Punkte nach Ablauf der Zeit immer noch größer als Null sind, erscheinen auf dem micro:bit ein fröhliches Gesicht und ein schöner Ton.

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

'If...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 19 sehen Sie den Code in seiner Gesamtheit.

```
1.1 1.2 Nervenspirale RAD 11/35
dd.py
from microbit import *
from time import *

print("drücken Sie A")

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

print("Spiel an")
Punkte = 3
t0 = clock()
Zeit = 0

while Zeit <30 and Punkte>0:
    Zeit = clock()-t0
    display.show(Punkte)
    if pin0.is_touched():
        Punkte = Punkte - 1
        display.show(Image.SAD)
        music.play(music.JUMP_DOWN )
        sleep(1)

if Punkte > 0:
    display.show(Image.HAPPY)
    music.play(music.ENTERTAINER)
else:
    music.play(music.DADADADUM)

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

display.clear()
```

Abbildung 19: 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.

## Erweiterung zusätzliche Kontaktstelle

Bei dem oben beschriebenen Code endet das Spiel nach 30 Sekunden. Wenn Sie die Spirale schneller abschließen, ertönt die Siegmusik erst, wenn die volle Zeit verstrichen ist.

Durch Hinzufügen eines zusätzlichen Kontaktpunktes können Sie das Spiel schneller beenden. Wenn Sie die Spirale beenden und den zusätzlichen Kontaktpunkt berühren, berühren Sie pin 1.

Im Code können Sie sehen, dass 'if pin1.is\_touched():' jetzt in der while-Schleife steht. Das ist jetzt möglich, weil man nicht mehr die ganze Schleife durchlaufen muss, um dann die Anzahl der Punkte zu sehen. Wenn Pin1 berührt wird und Ihre Punkte > 0 sind, haben Sie gewonnen. Sie können den Code mit Leerzeichen oder Tabulator einrücken.

Außerdem können die Schüler die Anzahl der Punkte und die Dauer anpassen, um den Schwierigkeitsgrad ihrer gebauten Spiralen zu erhöhen.

Viel Spaß!

```
while Zeit <30 and Punkte>0:
  ⚡ Zeit = clock()-t0
  ⚡ display.show(Punkte)
  ⚡ if pin0.is_touched():
    ⚡ Punkte = Punkte - 1
    ⚡ display.show(Image.SAD)
    ⚡ music.play(music.JUMP_DOWN )
    ⚡ sleep(1)
  ⚡ if pin1.is_touched():
    ⚡ display.show(Image.HAPPY)
    ⚡ music.play(music.ENTERTAINER)
else:
  ⚡ music.play(music.DADADADUM)
```



Abbildung 20: Spiel mit zusätzlichem Kontaktpunkt