

Technische Universität

Physical Computing Sommersemester 2020

Beleg zum Seminar „Physical Computing“

Beleg erstellt durch: Max Junghänel

Betreuender Dozent: Dr. Thiemo Leonhardt

Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0)

Inhalt

| | |
|--|----|
| Physical Computing | 3 |
| 1. Calliope Mini..... | 4 |
| Übergeordnete Aufgabe - x-seitiger Würfel..... | 4 |
| Stufe 1 | 5 |
| Stufe 2 | 5 |
| Stufe 3 | 5 |
| Stufe 4 | 6 |
| Stufe 5 | 7 |
| Stufe 6 | 7 |
| 2. ESP32 | 9 |
| Stufe 1 | 9 |
| Stufe 2 | 9 |
| Stufe 3 | 10 |
| Stufe 4 | 10 |
| Stufe 5 | 11 |
| Stufe 6 | 13 |
| 3. Appcreator..... | 15 |
| Stufe 1 | 15 |
| Stufe 2 | 16 |
| Stufe 3 | 17 |
| Stufe 4 | 18 |
| Stufe 5 | 20 |
| Stufe 6 | 21 |
| Literaturverzeichnis..... | 22 |

Physical Computing

Bloßes Programmieren am Computer wirkt für den Menschen oft abstrakt und wenig anschaulich. Die Wirkung, die die geschriebenen Programmzeilen entfalten, ist meist auf die Ausgabe eines Programmes am Bildschirm begrenzt. Physical Computing dagegen beschreibt eine anschaulichere Computer-Mensch-Interaktion. Dabei werden Software und Hardware eng miteinander verknüpft, sodass sich die Wirkung von Programmen realer wahrnehmen lässt und Einflüsse aus der realen Welt auch durch ein Computersystem erfassen und darstellen lassen. Man nutzt hierfür häufig Microcontroller mit verschiedenen Schnittstellen und Sensoren. Dieser Ansatz eignet sich besonders für Informatikunterricht in der Schule, weil es den Umgang mit Computersystemen erfahrbarer macht und vor allem jungen Schülern den Zugang zur Informatik erleichtert.

In dieser Seminararbeit werden drei Tools für das Physical Computing kurz vorgestellt: Calliope Mini, der ESP32-Microcontroller und der MIT AppInventor. Zu jedem Tool werden sechs Aufgaben formuliert, die jeweils das Spektrum der Lernzieltaxonomien nach Anderson und Krathwohl (2001) abdecken sollen. Diese Lernzielstufen beinhalten folgende Kriterien^[1]:

- Stufe 1: aufzählen, benennen, beschreiben, reproduzieren, wiederholen, kennen
- Stufe 2: erklären, demonstrieren, skizzieren, veranschaulichen, erläutern, subsumieren
- Stufe 3: aufzeigen, anwenden, definieren, klassifizieren, untersuchen, vervollständigen
- Stufe 4: kontrastieren, untersuchen, unterscheiden, vergleichen, differenzieren
- Stufe 5: begründen, bewerten, beurteilen, kritisieren, einschätzen, auswerten, überprüfen, entscheiden
- Stufe 6: entwerfen, gestalten, planen, generieren, ausdenken, produzieren

1. Calliope Mini

Der Calliope Mini ist ein Mikrocontroller, der für die Nutzung im Schulunterricht ausgelegt ist. Er erfüllt die Ansprüche an ein Informatiksystem, hat einige integrierte Sensoren, Antennen, Tasten und Elektronikanschlüsse. Er besitzt einen Flash-Speicher, der mit bis zu 25 unterschiedlichen Programmen bespielt werden kann. Als Ausgabeschnittstellen dienen Lautsprecher und ein einfaches RGB-LED-Display. Dieser Mikrocontroller kann recht leicht sowohl über eine graphische Programmierumgebung als auch über Programmcode programmiert werden und eignet sich daher für die Nutzung durch ein breites Altersspektrum. Die Nutzung des Tools gestaltet sich sehr unkompliziert. Per USB lässt sich Calliope mit einem Computer verbinden. Die Programme werden in Code-Editoren wie <https://makecode.calliope.cc/> gestaltet. Alle in diesem Kapitel erstellten Abbildungen sind Screenshots von Code, der vom Verfasser mit <https://makecode.calliope.cc/> erstellt wurde.

Übergeordnete Aufgabe - x-seitiger Würfel

Zum Spielen von Brettspielen benötigt man häufig Würfel. Entwirf ein Calliope-Programm, mit dem du die Augenzahl von 1-6 ausgeben kannst. Ein Wurf wird durch Schütteln von Calliope simuliert.

Die Blöcke in der Programmierumgebung sind wie folgend gegeben:



Stufe 1

Aufgabenstellung

Der Calliope Mini ist ein Informatiksystem. Nenne fünf Eingabeschnittstellen.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Aufgabe fordert von den SchülerInnen ein Erinnern und Wiedergeben von bloßen Fakten.

Erwartungshorizont und Bewertung

- Bewegungssensor
- Tasten A/B
- Mikrofon
- Funk/Bluetooth
- Touchpins
- USB-Port
- Thermometer ...

Bei dieser Aufgabenstellung kann man bis zu 5 Punkte vergeben. Ebenso wäre es möglich, auf die Nennung mehrerer Antworten eine Bewertungseinheit zu vergeben.

Stufe 2

Aufgabenstellung

Erkläre anhand der Programmblöcke, was eine Zufallsvariable ist.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Bei der zweiten Taxonomiestufe müssen die SchülerInnen nachweisen, dass sie einen Zusammenhang verstehen. Hier zeigen sie ihr Verständnis für das Konzept der Variable und einer zufälligen Zuweisung und die Umsetzung im Programm.

Erwartungshorizont und Bewertung

Eine Variable steht stellvertretend für einen gewissen Zahlenwert. Bei einer Zufallsvariable wird dieser Wert in einer angegebenen Reichweite zufällig zugewiesen. Hier wird der Variablen „Zufallszahl“ beim Schütteln ein Wert zwischen 1-6 (*Zahl1*) zugewiesen.

Für diese Aufgabe können bis zu drei Punkte vergeben werden. Der erste für die Erklärung vom Begriff Variable, der zweite für die Zuweisung der Zufallsfunktion und der dritte für den Bezug zum Beispiel.

Stufe 3

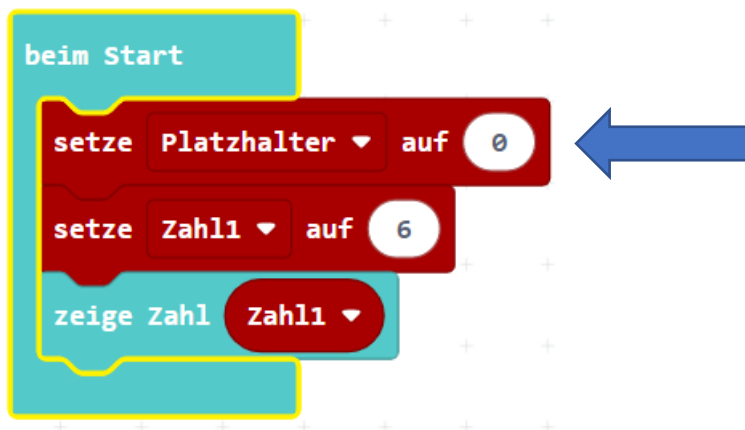
Aufgabenstellung

Beim Start soll eine weitere Variable mit dem Wert 0 und dem vorläufigen Namen *Platzhalter* initialisiert werden. Setze dies in der Programmierumgebung um.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die dritte Taxonomiestufe beschreibt ein Anwenden von verstandenen Zusammenhängen und Fähigkeiten. Für diese Aufgabe müssen die SchülerInnen zeigen, dass sie die Initialisierung von Variablen verstanden haben und dieses Wissen selbst durch zusammenklicken der Blöcke anwenden.

Erwartungshorizont und Bewertung

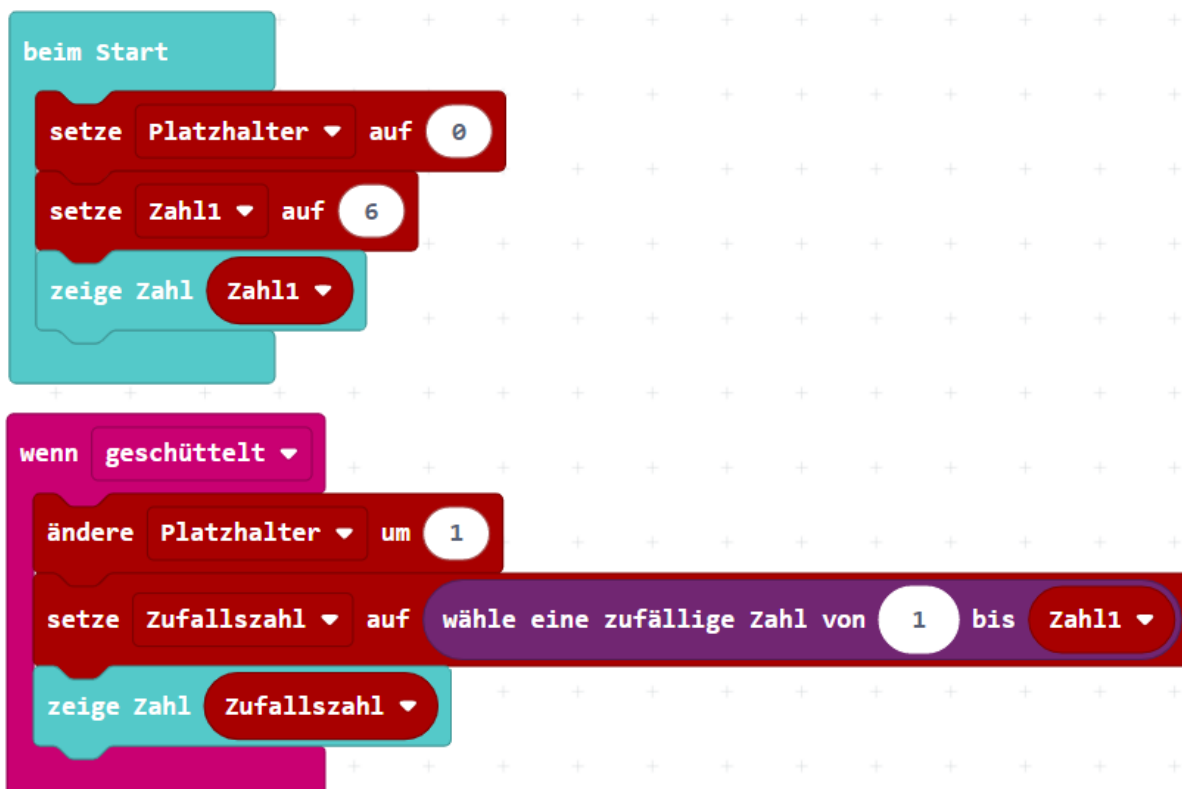


Hier sind bis zu zwei Punkte zu vergeben: Der erste für den richtigen Platzhalterblock, der zweite für das Einpflegen in den Startblock.

Stufe 4

Aufgabenstellung

Analysiere diese Modifikation des Ausgangsprogramms. Erkläre die Funktion des Zusatzes und benenne die Variable *Platzhalter* sinnvoll.



Anmerkung: Wenn Aufgabe 3 vorher durchgeführt wird, muss der obere Block weggelassen werden.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die vierte Taxonomiestufe beschreibt die Analysefähigkeit der SchülerInnen. Bei dieser Aufgabe müssen sie verstehen und wiedergeben können, welche Funktion die Blöcke erfüllen. Dies soll auch in einer sinnvollen Benennung der Variable zum Ausdruck gebracht werden.

Erwartungshorizont und Bewertung

Die Blöcke initialisieren beim Start des Programmes eine Variable *Platzhalter* mit dem Wert 0. Bei jedem Schütteln wird diese Variable inkrementiert. Sie ist also eine Möglichkeit, um die durchgeführten Würfe zu zählen.

Eine sinnvolle Benennung wäre z.B. *AnzWuerfe*.

Stufe 5

Aufgabenstellung

Bewerte die Einführung der Variable *Zahl1* zu Beginn des Programmes.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Bei dieser Aufgabe muss der Einsatz einer Variablen evaluiert werden. Die SchülerInnen müssen hier abwägen, ob der Einsatz nötig und sinnvoll ist und dies mit Argumenten stützen.

Erwartungshorizont und Bewertung

Die Variable *Zahl1* muss an dieser Stelle nicht eingeführt werden. Man könnte bei der Ermittlung der Zufallszahl einfach die Reichweite 1-6 angeben. Möchte man die Reichweite zu einem späteren Zeitpunkt allerdings verändern, ist die Initialisierung der Variable sinnvoll.

Hier kann man 2-3 Punkte vergeben. Für die Einschätzung, dass die Variable nicht unbedingt eingeführt werden muss (und so das Programm komplizierter macht als nötig) und für die Begründung jeweils einen. Für die Benennung der Möglichkeit einer weiteren Verwendung der Variable ist ebenfalls ein Punkt möglich.

Stufe 6

Aufgabenstellung

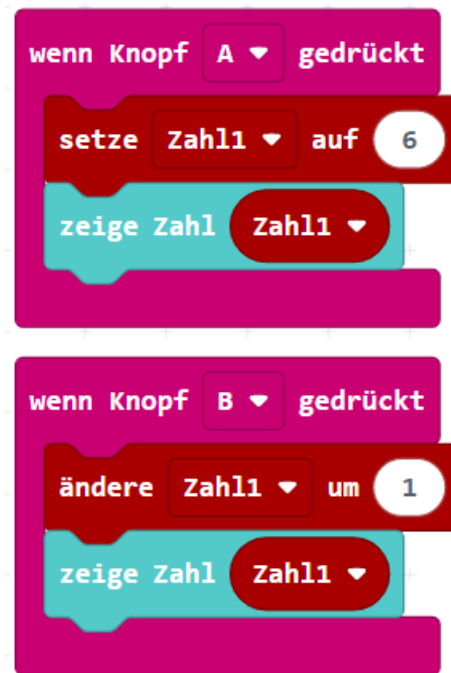
Gerade bei komplexeren Brettspielen oder Pen-And-Paper-Spielen nutzt man gerne auch Würfel mit mehr als sechs Seiten.

Entwickle und implementiere eine Funktion, mit der man die maximale Augenzahl einstellen kann. Die aktuelle Augenzahl wird dabei jeweils angezeigt. Ein Zurücksetzen auf die Standard-Augenzahl 6 soll auch möglich sein.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die sechste Taxonomiestufe beschreibt das Planen und Kreieren von Funktionen, Programmen etc.. Bei dieser Aufgabe müssen die SchülerInnen die Funktionalität des Blockes selbst planen und durch die Implementierung zeigen, dass sie bisher erworbenes Wissen über das Programmieren bei Eintreten von Bedingungen und Veränderung von Variablen anwenden können.

Erwartungshorizont und Bewertung



Möglicher Algorithmus zur Lösung:

Für diese Aufgabe sind ca. vier Punkte zu vergeben. 1 Punkt für die Bedingungsblöcke. 1 Punkt für den Variable-Setzen-Block. 1 Punkt für den Variable-Ändern-Block und 1 Punkt für die Ausgabe der Zahl.

2. ESP32

Der ESP32 ist ein Mikrocontroller mit Steckplatine. Mikrocontroller sind kleine Einplatinenrechner, die häufig als Bauteile in Elektronikgeräten verwendet werden. Der ESP32 kann über Steck-Pins und einen USB-Anschluss mit Peripheriegeräten oder einem Computer verbunden werden. Auf dem ROM-Speicher des ESP32 kann somit ein Programm aufgespielt werden. Schülerinnen und Schüler können die entsprechenden Programme vorher in einer bekannten Programmierumgebung erstellen. Im Unterricht kann ein solcher Mikrocontroller sehr vielfältig eingesetzt werden, da durch die verschiedenen Pins viele Möglichkeiten zum Anschließen von Peripheriegeräten bestehen. Sowohl Input-Pins, die mit Thermometer, Lichtsensoren etc. verbunden werden können als auch Output-Pins, die den Anschluss von LEDs, Lautsprechern etc. ermöglichen stehen zur Verfügung. So lassen sich auch einfache Stromkreise auf Leitersteckplatten realisieren und durch den Mikrocontroller ansteuern.

Stufe 1

Aufgabenstellung

Nenne drei Gegenstände aus deinem Alltag, in dem Microcontroller verbaut sind.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die unterste Taxonomiestufe beschreibt die Aufzählung und Benennung von Fakten wie in der Aufgabe gefordert.

Erwartungshorizont und Bewertung

Fernseher, Auto, Uhren, ...

Abhängig von der Situation können pro Nennung ein Punkt oder insgesamt eine Bewertungseinheit vergeben werden.

Stufe 2

Aufgabenstellung

Erkläre den Unterschied zwischen digitalen und analogen (INPUT/OUTPUT) Signalen.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Schülerinnen und Schüler müssen bei dieser Aufgabe einen Zusammenhang erklären. Dazu kann auch Recherche nötig sein. Das entspricht der zweiten Stufe der Lernzieltaxonomie

Erwartungshorizont und Bewertung

Ein analoges Signal ist eine physikalische Größe, die im Verlauf der Größe (Amplitude) als auch im zeitlichen Verlauf kontinuierliche Werte annehmen kann. Ein digitales Signal (digitus: Finger, lat.) ist eine physikalische Größe, die nur bestimmte diskrete Werte annehmen kann. Die Werte entsprechen der Anzahl der vereinbarten Zustände. Werden zwei Zustände vereinbart, dann handelt es sich um binäre (digitale) Signale.

Analoge Signale bestehen aus einem kontinuierlichen Signal- bzw. Datenstrom. Den gibt es bei digitalen Signalen nicht. Wenn digitale Daten anfallen, dann entstehen sie meist explosionsartig. Das Datenaufkommen überschreitet dann häufig ein Maß, das nicht mehr verarbeitet werden kann. Die Daten müssen dann zwischengespeichert werden, damit sie nicht verloren gehen.^[2]

Punkte gibt es für die Erklärung des kontinuierlichen (analog) und diskreten (digital) Wertebereiches.

Stufe 3

Aufgabenstellung

(In einer Anleitungsphase wie z.B. auf dem AB MicroPython: Einstieg | Blinkende Lichter, wird das Anschließen einer roten LED in den PIN 23 eingeführt. ^[3])

Aufgabe: Schließe eine blaue LED an, die über den digitalen PIN 32 angesteuert wird. Programmiere die LED, sodass sie dauerhaft leuchtet.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Anwendung von bekanntem Wissen – hier das Anschließen und Programmieren einer neuen LED nach bekanntem Muster – entspricht der dritten Stufe der Lernzieltaxonomie nach Krathwohl.

Erwartungshorizont und Bewertung

```
blau = Pin(27, Pin.OUT)
blau.value(True)
```

Für jede Zeile korrekten Code gibt es einen Punkt, sodass 2 Punkte für die Aufgabe zu vergeben sind.

Stufe 4

Aufgabenstellung

Vergleiche den ESP32-Mikrocontroller und einen Laptop in der Tabelle.

| Kriterium | Laptop | Mikrocontroller |
|-------------------------------|--------|-----------------|
| Informatiksystem | | |
| Eingabe | | |
| Verarbeitung | | |
| Ausgabe | | |
| Schnittstellenarten | | |
| verbaute Speicher | | |
| Leistung | | |
| Prozessorkerne | | |
| Betriebssystem | | |
| Stromversorgung und Verbrauch | | |

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die vierte Lernzieltaxonomiestufe umfasst auch die Fähigkeit, Vergleiche zu ziehen. Bei dieser Aufgabe ist dies sowohl selbstständig, als auch unter Nutzung von Recherchematerialien möglich.

Erwartungshorizont und Bewertung

Je nach angedachtem Schwierigkeitsgrad und zur Verfügung gestellten Hilfsmitteln sind hier unterschiedliche Punkteverteilungen möglich. In dieser Tabelle würde sich beispielsweise ein Punkt pro Vergleichskriterium anbieten. Es wäre auch möglich, die Schüler eigene Vergleichskriterien finden zu lassen.

| Kriterium | Laptop | Mikrocontroller |
|-------------------------------|--|----------------------------|
| Informatiksystem | ja | ja |
| Eingabe | über Peripheriegeräte, Tastatur, Touchpad, DVD-Laufwerk, ... | über Pins, USB |
| Verarbeitung | Arbeitsspeicher, Prozessor, Grafikkarte | Arbeitsspeicher, Prozessor |
| Ausgabe | Display, Lautsprecher, Schnittstellen | über Pins, USB |
| Schnittstellenarten | HDMI, USB, VGA, ... | Pins, USB |
| verbaute Speicher | RAM, ROM, interne Festplatte | ROM, RAM |
| Leistung | | niedrige Taktrate |
| Prozessorkerne | z.B. Quadcore | 1 |
| Betriebssystem | ja | nein |
| Stromversorgung und Verbrauch | Netzteil, 60 W | USB, kaum Stromverbrauch |

Stufe 5

Aufgabenstellung

Der folgende Code erschafft ein Disco-Licht, bei dem eine RGB-Leuchte nacheinander verschiedene Farben anzeigt. Der Code ist allerdings fehlerhaft. Markiere die Syntaxfehler, die auch der Interpreter finden würde.

```
from machine import *

grün = Pin(27: Pin.OUT)
rot = Pin(32: Pin.OUT)
blau = Pin(23: Pin.OUT)

while:
#rot
blau.value(False)
rot.value(True)
gruen.value(False)

#grün
blau.value(False)
rot.value(False)
gruen.value(True)

#blau
blau.value(True)
rot.value(False)
gruen.value(False)
```

```
#gelb
blau.value(False)
rot.value(True)
gruen.value(False)

#pink
blau.value(True)
rot.value(True)
gruen.value(False)

#türkis
blau.value(True)
rot.value(False)
gruen.value(True)

#weiß
blau.value(True)
rot.value(True)
gruen.value(True)
```

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Bei dieser Aufgabe wird die Analysefähigkeit der Schülerinnen und Schüler überprüft. Sie müssen einen Code hinsichtlich der Syntax überprüfen und Fehler herausfinden. Das entspricht der fünften Stufe der Lernzieltaxonomie.

Erwartungshorizont und Bewertung

Fehler:

grün = : ü wird vom Interpreter nicht als Zeichen akzeptiert.

Die Doppelpunkte bei der Festlegung der Variablen: Dies müssten Kommata sein.

Die while-Schleife muss mit while True eingeleitet werden.

Der Code in der while-Schleife muss eingerückt sein.

Für jeden erkannten Fehler wird ein Punkt vergeben.

Stufe 6

Aufgabenstellung

Die aus Ausgabe 5 programmierte Disco-LED zeigt keinen sichtbaren Farbwechsel. Das liegt daran, dass das menschliche Auge die Farbwechsel aufgrund der Geschwindigkeit nicht wahrnehmen kann. Verändere den Code soweit, dass der Farbwechsel langsamer abläuft und er optisch wahrgenommen werden kann.

Mögliche Hilfe:

Importiere aus dem Package *time* die *sleep* – Funktion und implementiere nach jedem Farbwechsel im korrigierten Code von Aufgabe 5 eine kurze Pause. Die *sleep*-Funktion wird durch *sleep(Dauer in s)* aufgerufen.

Zusatz 1: Schreibe den Code so, dass sich die *sleep*-Dauer möglichst leicht ändern lässt.

Zusatz 2: Programmiere einen unterschiedlich schnellen Wechsel der Farben nach jedem Schleifendurchlauf.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Schüler müssen selbstständig eine Idee für einen langsameren Farbwechsel entwickeln und diese als Code umsetzen. Diese Planung und Produktion des Codes entspricht der sechsten Stufe der Lernzieltaxonomie.

Erwartungshorizont und Bewertung

```
from machine import *
from time import sleep

gruen = Pin(27, Pin.OUT)
rot = Pin(32, Pin.OUT)
blau = Pin(23, Pin.OUT)

a = 0.5

while True:
    #rot
    blau.value(False)
    rot.value(True)
    gruen.value(False)
    sleep(a)

    #grün
    blau.value(False)
    rot.value(False)
    gruen.value(True)
    sleep(a)

    #blau
    blau.value(True)
    rot.value(False)
    gruen.value(False)
    sleep(a)

    #gelb
    blau.value(False)
    rot.value(True)
    gruen.value(False)
    sleep(a)
```

```
#pink
blau.value(True)
rot.value(True)
gruen.value(False)
sleep(a)

#türkis
blau.value(True)
rot.value(False)
gruen.value(True)
sleep(a)

#weiß
blau.value(True)
rot.value(True)
gruen.value(True)
sleep(a)
```

Punkte sind zu vergeben für den Import der *sleep*-Funktion sowie für die korrekte Umsetzung des Codes, der ein *sleep(time)* zwischen jedem Farbwechsel verlangt. Dafür würde ich vier Punkte vergeben. Wenn die Anforderung von Zusatzaufgabe 1 umgesetzt wird, kann es einen weiteren Punkt geben. Hierfür bietet sich die Einführung einer Variablen an (in der Beispiellösung *a*), sodass der Wert schnell für den gesamten Code zu verändern ist.

Weiterhin könnte man zu Erfüllung von Zusatz 2 den *sleep*-Timer für jeden Schleifendurchlauf neu bestimmen lassen. Vorstellbar ist die Nutzung der *random*-Funktion, die aus dem Random-Package importiert werden muss oder die Veränderung der Variablen um einen gewissen Wert.

3. Appcreator

Das Massachusetts Institute of Technology bietet eine kostenlose Plattform an, mit der man sehr simpel Apps für mobile Geräte auf Basis des Android-Betriebssystems erstellen kann. Unter dem Link <http://ai2.appinventor.mit.edu/> kann man sich mit einem GoogleMail-Account einloggen. Alternativ kann die Lehrkraft auch temporäre Zugänge für die Schüler anlegen, welche Anonymität gewährleisten. Die für den Anfang recht komplexe Oberfläche besteht aus einem Designbereich und einem Programmierbereich auf Blockbasis. Hat man sich einen Überblick über die vielfältigen Möglichkeiten im Designbereich verschafft, ist schnell eine einfache App zusammengeclickt. Eigenschaften der Objekte wie z.B. Größe, Farbe, Schriftarten etc. lassen sich ebenfalls einstellen. Die auf Blöcken basierende Programmierumgebung ermöglicht sowohl einfache als auch komplexe Codes.


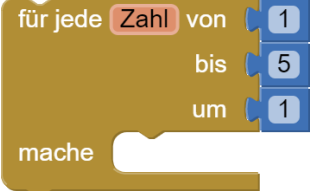

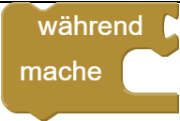


Die App lässt sich mithilfe der AI2 Companion App auf einem Android-Gerät öffnen, indem man in der appinventor-Umgebung die „Verbinden“-Funktion wählt und den generierten QR-Code scannt. Möglich ist auch die Ausführung auf Windows mithilfe eines Emulators. Dieser funktioniert allerdings bedeutend schlechter, weshalb an dieser Stelle von der Nutzung des Emulators abgeraten wird.

Die in diesem Kapitel verwendeten Blockcode-Grafiken stehen unter (CC BY-SA 3.0)-Lizenz von Massachusetts Institute of Technology. Die Screenshots wurden von Autor selbstständig angefertigt unter Nutzung der Oberfläche von <http://appinventor.mit.edu/>.

Stufe 1

Aufgabenstellung

Ordne die Blöcke den algorithmischen Grundstrukturen Sequenz (1), Verzweigung (2) und Schleife (3) zu und kreuze an.

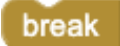


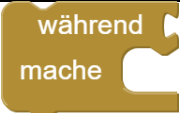


| Block | (1) | (2) | (3) |
|--|-----|-----|-----|
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Im Falle, dass die SchülerInnen die algorithmischen Grundstrukturen kennen, reproduzieren sie in dieser Aufgabe ihr Wissen und ordnen die Blöcke durch Erkennen der if... else-Konstruktion bzw. der while-Schleife und der for-Schleife zu. Die Verarbeitungsebene geht nicht über eine Reproduktion und ein Erkennen hinaus, deshalb betrifft es die Lernzieltaxonomiestufe 1.

Erwartungshorizont und Bewertung

Sequenz(1), Verzweigung (2) und Schleife (3)

| Block | (1) | (2) | (3) |
|--|-----|-----|-----|
|  | | | X |
|  | | | X |
|  | | | X |
|  | | | X |
|  | X | | |
|  | | X | |

Für die richtige Zuordnung jedes Blockes kann ein Punkt vergeben werden. Diskutabel ist die Zuordnung des größeren Blocks 5. Zwar beginnt der Block mit dem Schlüsselwort „wenn“, dieses leitet hier allerdings keine Verzweigung, sondern den Aufruf einer Funktion ein.

Stufe 2

Aufgabenstellung

- Ziehe zuerst einen **Button** (Taste) aus der Kategorie *Benutzerschnittstelle (User Interface)*, eine **Zeichenfläche (Canvas)** aus *Zeichnen und Animation*.
- Benenne anschließend den Button in „Aufnahme“ und das Canvas in „Leinwand“ um, indem ihr die jeweilige Komponente auswählt und auf **Umbenennen** klickt.

Beschreibe am Beispiel der **Taste** den Zusammenhang Klasse-Objekt-Attribut-Methode.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Schüler kennen den Zusammenhang Klasse-Objekt-Attribut-Methode. Diesen sollen sie an einem einfachen Beispiel beschreiben.

Erwartungshorizont und Bewertung

Durch das Ziehen der Taste auf den Screen wird ein neues Objekt „Taste1“ erzeugt. Objekte gehören Klassen an, wie hier Objekt Taste1 der Klasse „Taste“ angehört. Es hat verschiedene Attributwerte wie Name, Schriftart, Größe usw.. Auf Objekten sind für die Klasse spezifische Methoden ausführbar, wodurch Attributwerte verändert werden können. Z.B.: Taste1.SchriftGrößeÄndern(13).

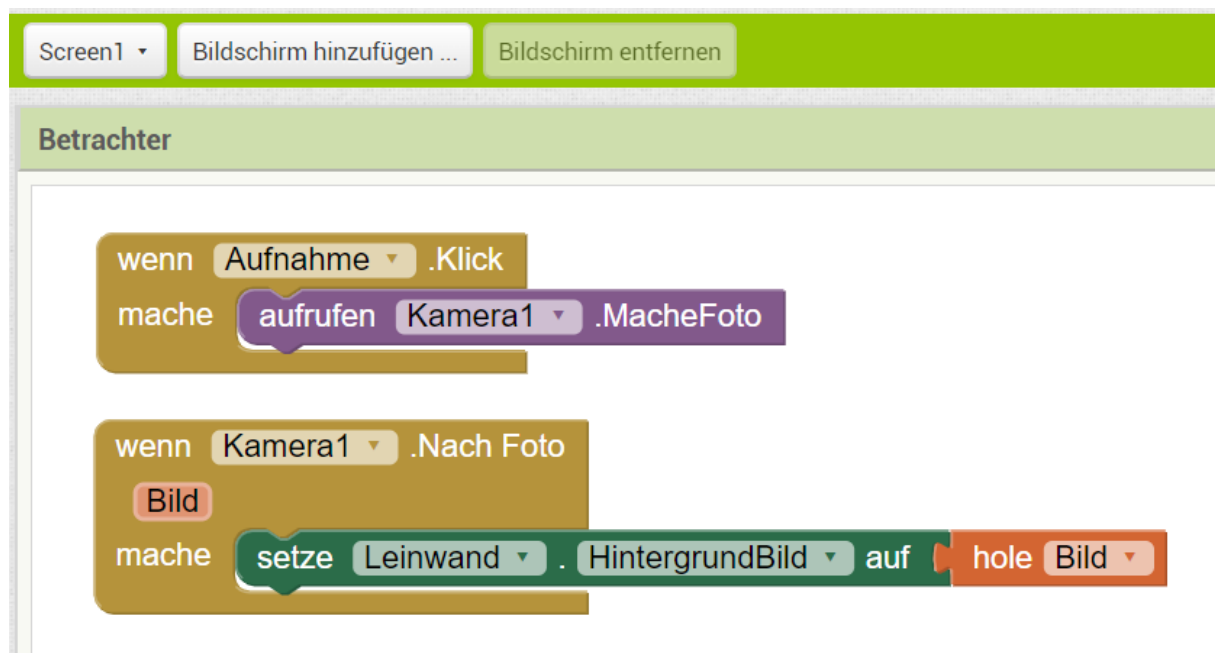
Für jedes Element des KOAM-Zusammenhangs kann ein Punkt vergeben werden, sodass insgesamt 4 Punkte zu vergeben sind.

Stufe 3

Aufgabenstellung

(Im Vorfeld wurden im Designer eine Taste, eine Kamera und eine Zeichenfläche hinzugefügt und entsprechend benannt.)

Der folgende Blockcode soll umgesetzt und ausgeführt werden:



(Verwendetes Material vom InfoSphere - Schülerlabor Informatik der RWTH Aachen, Lizenz: CC BY-SA 4.0 ^[4])

Aufgaben im Anschluss: Füge eine weitere Taste auf deinem Bildschirm hinzu. Sie soll die Funktion haben, das aktuelle Bild zu löschen und dementsprechend benannt sein.

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die SchülerInnen wenden die vorher angeleitete Erstellung eines Buttons, die Änderung des Attributs Name und die Belegung mit einer Funktion an. Das Anwenden fällt in die dritte Stufe der Lernzieltaxonomie.

Erwartungshorizont und Bewertung

```
wenn Löschen .Klick
mache aufrufen Leinwand .Lösche
```

Die Taste mit dem Attribut Name="Löschen" muss erstellt werden, dafür gibt es einen Punkt. Einen weiteren Punkt gibt es für die in den Blöcken programmierte Funktionalität.

Stufe 4

Aufgabenstellung

Es sind zwei Programme gegeben, mit denen eine Zufallszahl erraten werden soll.

Benenne funktionale Unterschiede in der Umsetzung der Programme.

Programm 1:

```
global initialisieren zufallszahl auf 0
global initialisieren anzahl_versuche auf 6

wenn Screen1 .Initialisiere
mache setze global zufallszahl auf zufällige Zahl zwischen 1 bis 100

wenn zahl_überprüfen .Klick
mache setze global anzahl_versuche auf hole global anzahl_versuche - 1
      aufrufen feedback_anzahl_versuche
      aufrufen feedback_tipp

zu feedback_anzahl_versuche
mache wenn hole global anzahl_versuche > 1
      dann setze Label_Anzahl_Versuche .Text auf hole global anzahl_versuche
      sonst setze Label_Anzahl_Versuche .Text auf "Keine Versuche übrig!"
           setze Label_Ergebnis .Text auf verbinde "Du hast die Zahl leider nicht erraten. Die Zufal..."
           hole global zufallszahl

zu feedback_tipp
mache wenn Textbox_Tipp .Text = hole global zufallszahl
      dann setze Label_Ergebnis .Text auf verbinde "Richtig! Die Zufallszahl ist die "
           hole global zufallszahl
      sonst wenn Textbox_Tipp .Text < hole global zufallszahl
      dann setze Label_Ergebnis .Text auf "Deine getippte Zahl war zu klein."
      sonst setze Label_Ergebnis .Text auf "Deine getippte Zahl war zu groß."
```

Programm 2: (Umsetzung in der Programmiersprache Python. Screenshot von Thonny 3.2.7 in Windows 10)

```

1 from random import randint
2
3 print("Erraten Sie eine Zahl zwischen 1 und 10 in möglichst wenig Versuchen")
4 tipp=0
5 geheimnis = randint(1, 10)
6 schritte = 0
7
8 while tipp != geheimnis:
9     eingabe = input("Bitte geben Sie Ihren Tipp ein: ")
10    tipp = int(eingabe)
11
12    if tipp == geheimnis:
13        print("Gewonnen")
14
15    elif tipp>geheimnis and tipp<=10:
16        print("Zu groß")
17    elif tipp<geheimnis and tipp>=1:
18        print("Zu klein")
19    else:
20        print("außerhalb des Wertebereiches!")
21
22    schritte=schritte+1
23
24 print("Du hast gewonnen und ",schritte," Schritte gebraucht!")
25

```

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Bei dieser Aufgabe sollen zwei ähnliche Programme verglichen werden, die in unterschiedlichen Programmierumgebungen erstellt wurden. Dabei werden die Unterschiede im Programmablauf benannt. Solche Vergleichs- und Differenzierungsleistungen fallen in die vierte Stufe der Lernzieltaxonomie.

Erwartungshorizont und Bewertung

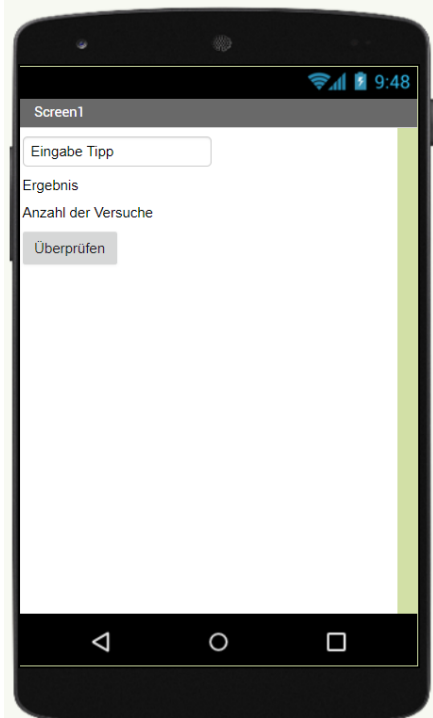
| Kriterium | Blockcode | Pythoncode |
|------------------------------|---|---|
| Zufallszahl | von 1-100 | von 1-10 |
| maximale Anzahl der Versuche | 6 | unbegrenzt |
| Falsche Eingabe abfangen | - | alle Zahlen außerhalb des Wertebereichs 1-10 |
| Funktionen | Funktionale Programmierung, Aktivierung der Funktionen erfolgt durch Buttons | keine Funktionen definiert |
| Schleife | Hintergrundschleife, die bei der Aktivierung eines Buttons (zahl_überprüfen.Klick) die Funktionen aufruft | while-Schleife, die erst durch die Eingabe der richtigen Zahl beendet wird. |

Die Vorgabe der Unterscheidungskriterien ist bei der Aufgabenstellung denkbar. Für das korrekte Erkennen jedes Kriteriums ist ein Punkt zu vergeben, sodass insgesamt 5 Punkte erhalten werden können.

Stufe 5

Aufgabenstellung

Beurteile die Bedienbarkeit der Design-Oberfläche. Gib Hinweise, wie man die Bedienbarkeit der App verbessern kann.



Screenshot von <http://ai2.appinventor.mit.edu/>

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die kritische Auseinandersetzung mit einem Lerngegenstand und eine darauffolgende Beurteilung, die in dieser Aufgabe verlangt wird, entspricht der fünften Lernzieltaxonomiestufe nach Krathwohl.

Inhaltlich ist diese Aufgabe eher dem Gebiet der Medieninformatik zuzuordnen, die sich mit der Mensch-Computer-Interaktion beschäftigt.

Erwartungshorizont und Bewertung

Diese Aufgabenstellung ist offen formuliert. Daher ist es schwierig, dafür konkrete Punktverteilungen festzulegen. Man könnte nach einer festen Zahl an Verbesserungsvorschlägen fragen und jeden Vorschlag und dessen Begründung mit jeweils einem Punkt bewerten.

Beispielhaft sind hier einige Beurteilungsaspekte dargestellt:

- Das Design enthält alle relevanten Elemente.
- Das Design ist nicht anschaulich oder intuitiv zu bedienen.
- Die Anordnung der Elemente muss überarbeitet werden. Z.B.: der Überprüfen-Button mittig unten.
- allgemein die Schaltflächen und Textfelder größer darstellen
- bessere Visualisierung: z.B. ein Fortschrittsbalken für die Anzahl der Versuche implementieren, bei falschen Eingaben den Hintergrund oder die Textfarbe auf rot setzen.
- Zur Funktionalität: eine Restart-Funktion implementieren

Stufe 6

Aufgabenstellung

Verändere den Blockcode aus Stufe 4 insofern, dass die Eingabe der geratenen Zahl auf ihre Gültigkeit (Zahl 1-100) hin überprüft und etwaige Fehleingaben abgefangen werden.

```
global initialisieren zufallszahl auf 0
global initialisieren anzahl_versuche auf 6

wenn Screen1 .Initialisiere
mache
  setze global zufallszahl auf zufällige Zahl zwischen 1 bis 100

wenn zahl_überprüfen .Klick
mache
  setze global anzahl_versuche auf hole global anzahl_versuche - 1
  aufrufen feedback_anzahl_versuche
  aufrufen feedback_tipp
```

Einordnung laut der Lernzieltaxonomie nach Krathwohl

Die Erweiterung des Programmes muss von den Schülerinnen und Schülern selbstständig entwickelt und als Programmcode umgesetzt werden. Daher fällt diese Aufgabe in die sechste Stufe der Lernzieltaxonomie nach Krathwohl.

Erwartungshorizont und Bewertung

```
wenn zahl_überprüfen .Klick
mache
  wenn
    Textbox_Tipp . Text ≥ 1 und Textbox_Tipp . Text ≤ 100
  dann
    setze global anzahl_versuche auf hole global anzahl_versuche - 1
    aufrufen feedback_anzahl_versuche
    aufrufen feedback_tipp
  sonst
    setze Label_Ergebnis . Text auf "Bitte gib eine Zahl zwischen 1 und 100 ein."
```

Für das Einfügen der wenn-dann-Struktur mit der korrekten Bedingung und des Einfügens des sonst-Teiles gibt es jeweils einen Punkt zu erwerben.

Literaturverzeichnis


- [1] <https://dbs-lin.ruhr-uni-bochum.de/lehreladen/planung-durchfuehrung-kompetenzorientierter-lehre/lehr-und-lernziele/lernzieltaxonomien-im-vergleich/> , 28.12.20
- [2] <https://www.elektronik-kompodium.de/sites/kom/2405151.htm>, 15.10.20
- [3] Verwendetes Material aus dem Projekt TUD-Sylber² in der Didaktik der Informatik der TU Dresden.
AB MicroPython: Einstieg | Blinkende Lichter
- [4] <http://ai2.appinventor.mit.edu/>

Eidesstattliche Erklärung

Hiermit versichere ich, **Max Junghänel**, die Seminararbeit mit dem Titel „*Beleg zum Seminar Physical Computing*“ in der Veranstaltung „*Physical Computing*“ im Sommersemester 2020 bei Hr. Dr. Thiemo Leonhardt selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von mir angegebenen Quellen und Hilfsmittel angefertigt zu haben. Alle aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Die Arbeit wurde noch keiner Prüfungsbehörde in gleicher oder ähnlicher Form vorgelegt. Mir ist bekannt, dass ein Betrugsversuch mit der Note "nicht ausreichend" (5,0) geahndet wird und im Wiederholungsfall zum Ausschluss von der Erbringung weiterer Prüfungsleistungen führen kann.


Dresden, 28.12.20



Max Junghänel

Für alle von mir selbst erstellten Dokumente erteile ich der Professur für Didaktik der Informatik der Fakultät Informatik der TU Dresden für Zwecke der Lehre und Forschung ein zeitlich und sachlich unbeschränktes, nichtexklusives Nutzungsrecht. Ich gebe alle von mir erstellten Dokumente zur weiteren Nutzung in Lehrveranstaltungen frei und willige insb. der Veröffentlichung in OPAL-Lernräumen ein.

Dresden, 28.12.2020



Max Junghänel