

Handreichung PhC

1. Kurzvorstellung

Destruction Derby mit Calliope mini und Calli:bot2

2. Einordnung in die Lehrpläne

Oberschule (alter Lehrplan 2019):

Klassenstufe 8 Lernbereich 2 – benötigt wahrscheinlich didaktische Reduktion

Oberschule (neuer Lehrplan 2022):

Klassenstufe 8 Lernbereich 1

Gymnasium (alter Lehrplan 2019):

Klassenstufe 8 Lernbereich 2 – benötigt wahrscheinlich didaktische Reduktion

Klassenstufe 9/10 Lernbereich 4

Gymnasium (neuer Lehrplan 2022):

Klassenstufe 8 Lernbereich 1

3. Lernziele

Kognitiv:

Die SuS wenden ihre Kenntnisse über Algorithmen und damit verbundenen Begriffen in der didaktisch reduzierten Lernumgebung Microsoft Makecode anhand des Beispiels des Destruction Derbys an.

Psychomotorisch:

Affektiv:

4. Voraussetzungen

Fachlich:

- Den SuS kennen den Algorithmus Begriff sowie die Begriffe Sequenz, Verzweigung und Wiederholung, Variable

Technisch:

- Die SuS benötigen ein mobiles Endgerät, welches über Bluetooth oder einen USB A verfügt. Vorzugsweise handelt es sich dabei um Desktop PCs oder Tabletts
- Geräte, welche nur über Bluetooth kommunizieren können, benötigen die App Calliope mini

Materiell:

- Jedes Team benötigt einen Calliope mini und Calli:bot2 (Bei einer Umsetzung mit Mobilien Endgeräten im USB Hub wird jeweils ein Micro USB Kabel pro Team benötigt)



- Auf dem Boden wird ein abgegrenzter Bereich zwischen 1m² bis 2m² benötigt
- Die Abgrenzung des Bodenbereiches sollte mindestens 5 cm hoch sein und nicht durch Calli:bots verschoben werden können (Mögliche Materialien: Kartons, Leisten oder auf die Seite gestellte Tische)
- Zettel und Stift

5. Kurzdarstellung

Zeit: 2 UE

Ziel: Alle SuS programmieren mit einem Partner einen Calli:bot2 als autonom fahrendes Auto. Dieses zeigt die Teamnummer und über die LED-Beleuchtung, wie viele Leben das Auto noch hat. Ein Auto verliert ein Leben, wenn der Calli:bot2 mit seinem Drucksensor gegen etwas fährt. Am Ende des Projektes werden alle Autos in die vorbereitete Arena gesetzt und gleichzeitig gestartet. Es gewinnt das Auto, welches als letztes noch mindestens ein Leben hat

Vorbereitung:

- Calliope und Calli:bot2 zusammenbauen
- Auf mobilen Endgeräten mit Bluetooth *Calliope mini* App herunterladen
<https://calliope.cc/programmieren/mobil>
- Material für Arena bereitlegen
- Je ein Zettel mit einer einzigartigen Teamnummer für alle Teams

Anforderung an Programmierung des Calliope:

- Das Programm lässt sich über einen Knopfdruck starten
- Calliope zeigt auf LED Feld solange die eigene Teamnummer an, bis die Leben des Calli:bot2 aufgebraucht sind. Danach soll das LED-Feld nur noch ein Kreuz zeigen
- Solange der Calliope/Calli:bot2 noch Leben hat, soll dieser zufällig nach vorne fahren
- Wenn der Calliope/Calli:bot2 kein Leben mehr hat soll dieser sich nicht weiter bewegen
- Wird der Drucksensor auf einer oder beiden Seiten des Calli:bot2 betätigt, so verliert der Calliope/Calli:bot2 ein Leben und fährt kurz zurück
- Die LED-Beleuchtung des Calli:bot2 gibt die Anzahl der übrigen Leben an
 - o Grün bedeutet noch 3 Leben
 - o Gelb bedeutet noch 2 Leben
 - o Violett bedeutet noch 1 Leben
 - o Rot bedeutet kein Leben mehr

Zusätzliche Anforderungen zur Differenzierung:

- Der Calliope/Calli:bot2 fährt nicht zufällig nach vorne, sondern versucht über Entfernungsmesser einen möglichst guten Weg zu finden

Achtung: Die zusätzlichen Anforderungen sollen in einer extra Projektdatei gelöst werden, sodass die SuS

auch auf diese am Ende zurückgreifen können, falls die zusätzlichen Anforderungen nicht umgesetzt werden konnten.

Durchführung:

- Die SuS werden in Teams zu jeweils 2 SuS aufgeteilt
 - Jedes Team bekommt einen Calliope mini, Calli:bot2 und eine Teamnummer
 - Jedes Team steht ein mobiles Endgerät zu Verfügung
 - o Mit USB-A: Arbeit auf der Website <https://makecode.calliope.cc/>
 - o Mit Bluetooth: Arbeit mit der App <https://calliope.cc/programmieren/mobil>
 - Bei Arbeit mit Bluetooth muss eine Bluetoothverbindung zum Calliope hergestellt werden <https://calliope.cc/programmieren/mobil>
 - Die SuS erstellen die Projektdatei
 - In der Projektdatei muss die Erweiterung für den Calli:bot2 hinzugefügt werden **ACHTUNG:** Nicht die Erweiterung *callibot* nutzen
 - o Oben rechts auf Zahnradsymbol klicken, dann auf Erweiterungen und dann in der Suchleiste folgende Paket-URL eingeben: <https://github.com/knotechgmbh/Callibot2>
 - o Gefunden Erweiterung *callibot2* hinzufügen
 - Die SuS programmieren Calliope entsprechend der Anforderungen
 - Lehrperson hilft bei Fragen
 - Lehrperson baut nebenbei die Arena für das Ende des Projektes auf
 - Mindestens 15 min vor Ende des Projektes bringen alle Teams ihr Calli:bot2 zur Arena und lassen diese alle gleichzeitig auf das Signal der Lehrperson losfahren
- Anmerkung:** Bei genügend Zeit könnte es sich lohnen die Fahrzeuge der SuS mit der Programmierung der zusätzlichen Anforderungen erst einmal nur mit dem Programm der normalen Anforderungen fahren zu lassen. In einer zweiten Runde können die SuS dann die Autos mit der Programmierung der zusätzlichen Anforderungen fahren lassen.
- Je nach späterer Nutzung der Calliope können die SuS am Ende des Projektes den Calliope mit einem leeren Programm flashen

Mögliche Lösung allgemeine Anforderungen:



Bei dem Start des Calliope wird die LED-Anzeige auf die Teamnummer gesetzt. In diesem Fall die 1.

Die Variable Leben wird auf 3 gesetzt, da jedes Auto zum Start 3 Leben hat

```

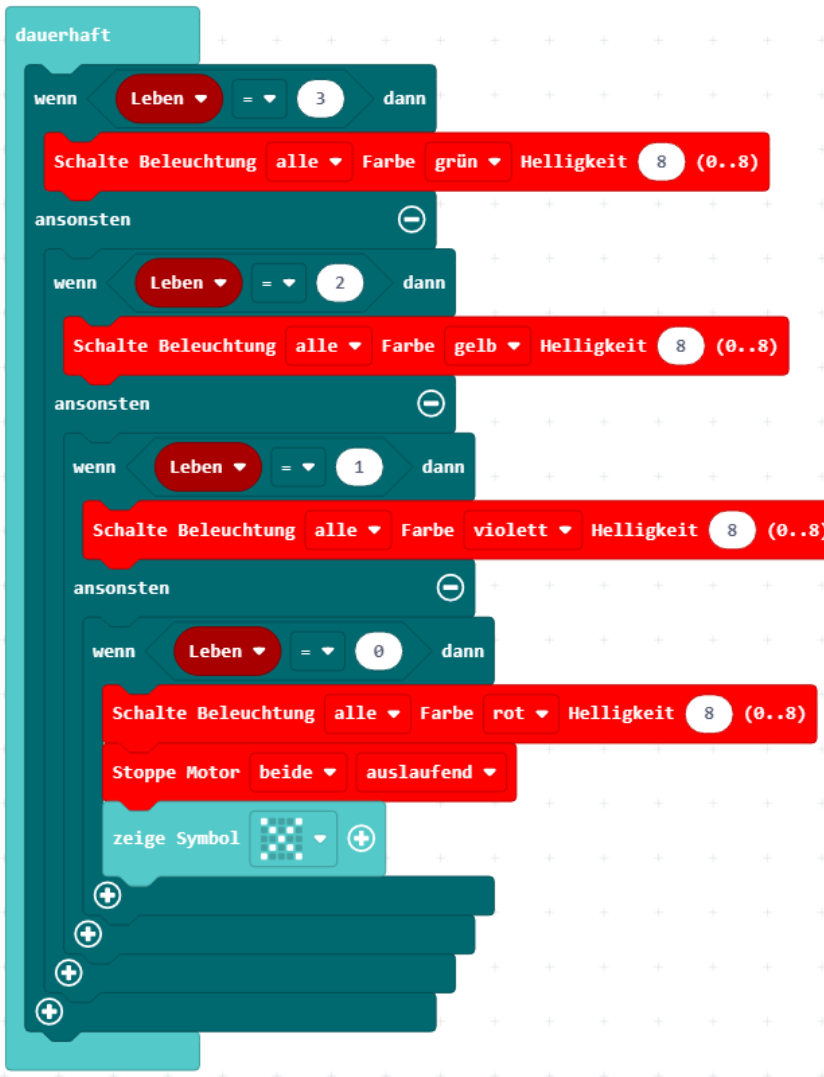
wenn Knopf A geklickt
  während Leben ≠ 0
    mache
      wenn Stoßstange links an oder Stoßstange rechts an dann
        Schalte Motor beide rückwärts mit 50
        pausiere (ms) 500
        Stoppe Motor beide bremsend
        setze Leben auf Leben - 1
      ansonsten
        setze Zufallsbewegung auf wähle eine zufällige Zahl von 0 bis 2
        wenn Zufallsbewegung = 0 dann
          Schalte Motor beide vorwärts mit 50
        ansonsten
          wenn Zufallsbewegung = 1 dann
            Schalte Motor links vorwärts mit 50
            Schalte Motor rechts vorwärts mit 5
          ansonsten
            wenn Zufallsbewegung = 2 dann
              Schalte Motor rechts vorwärts mit 50
              Schalte Motor links vorwärts mit 5
          pausiere (ms) 200
  
```

Wenn Knopf A gedrückt starte
Solange das Auto nicht 0
Leben hat, kann es fahren

Falls die Stoßstange des Autos
getroffen ist, fährt das Auto
zurück und setzt die Anzahl
der Leben auf eins weniger

Wurde die Stoßstange nicht
getroffen, so wird die Variable
Zufallsbewegung mit einem
zufälligen Wert zwischen 0
und 2 belegt. Je nach Wert
fährt das Auto geraderaus,
nach links oder nach rechts

Diese Bewegung wird 200 ms
ausgeführt und danach beginnt
die Schleife von vorn



Dauerhaft wird geprüft, auf welchem Wert die Variable Leben steht. Hat diesen den Wert 3, so ist die Beleuchtung grün mit Helligkeitsstufe 8

Hat Leben der Wert 2 wird die Beleuchtung auf die Farbe Gelb gesetzt mit der Helligkeit 8

Hat Leben der Wert 1 wird die Beleuchtung auf die Farbe Violett gesetzt mit der Helligkeit 8

Hat Leben der Wert 0 wird die Beleuchtung auf die Farbe Rot gesetzt mit der Helligkeit 8. Außerdem wird der Motor beider Räder ausgestellt und das Symbol *Kreuz* wird auf der LED-Anzeige angezeigt

Mögliche Lösung zusätzlicher Anforderungen:

```

wenn Knopf B geklickt
  während Leben ≠ 0
    mache
      wenn Stoßstange links an oder Stoßstange rechts an dann
        Schalte Motor beide rückwärts mit 50
        pausiere (ms) 500
        Stoppe Motor beide bremsend
        setze Leben auf Leben - 1
      sonst wenn Entfernung cm < 10 dann
        setze Zufallsbewegung auf wähle eine zufällige Zahl von 0 bis 1
        wenn Zufallsbewegung = 0 dann
          Schalte Motor links vorwärts mit 50
          Schalte Motor rechts vorwärts mit 5
        ansonsten
          wenn Zufallsbewegung = 1 dann
            Schalte Motor rechts vorwärts mit 50
            Schalte Motor links vorwärts mit 5
        ansonsten
          Schalte Motor beide vorwärts mit 50
        pausiere (ms) 200
  
```

Wenn Knopf B gedrückt starte
Solange das Auto nicht 0
Leben hat, kann es fahren

Falls die Stoßstange des Autos
getroffen ist, fährt das Auto
zurück und setzt die Anzahl
der Leben auf eins weniger

Ansonsten Falls die
Entfernung zu einem Objekt
unter 10 cm beträgt, wird die
Variable Zufallsbewegung auf
0 oder 1 gesetzt.

Ist die Variable 0, so fährt das
Auto nach links und ist sie 1 so
fährt das Auto nach rechts

Wenn weder die Stoßstange
berührt wurde noch die
Entfernung zu einem Objekt
unter 10 cm beträgt, fährt das
Auto gerade aus

Nach 200ms beginnt die
Schleife von vorne