

Callysto-Test zum rechnerischen Denken(CCTt) (Studentenversion) Posttest

angepasst von Christopher Bongert, 09.01.2023

Mit den Fragen in diesem Test sollen Ihre Fähigkeiten und Einstellungen zum rechnerischen Denken gemessen werden. Es kann sein, dass Sie die Antwort auf einige der Fragen nicht wissen. Wenn Sie die Antwort auf eine Frage nicht wissen, antworten Sie bitte mit Ihrer besten Vermutung.

Der Test dauert etwa 20 Minuten.

TEIL 1

Geben Sie an, wie gut jede der folgenden Aussagen Sie beschreibt:

	Stimmt gar nicht	Stimmt nicht	Stimmt	Stimmt voll und ganz
Ich benutze gerne Technik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich finde es einfach, neue Technologien zu nutzen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich bin zuversichtlich, dass ich meinen Computer selbst reparieren kann, wenn er nicht funktioniert	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Leute bitten mich um Hilfe mit ihrem Computer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bei der Beantwortung der folgenden Fragen kann es hilfreich sein, sich zu überlegen, wie Sie ein komplexes Problem lösen würden. Zum Beispiel:

- Planung einer Überraschungs-Geburtsstagsfeier für einen Freund
- Reparatur einer Reifenpanne am Fahrrad
- Entwerfen eines Fallschirms und eines Behälters, der ein Ei schützt, wenn es vom Dach der Schule herunterfällt

Geben Sie an, wie gut jede der folgenden Aussagen Sie beschreibt:

	Stimmt gar nicht	Stimmt nicht	Stimmt	Stimmt voll und ganz
Ich kann die Schritte zur Lösung eines komplexen Problems herausfinden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wenn ich ein komplexes Problem löse, versuche ich, es in kleinere oder einfachere Probleme aufzuteilen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wenn ich ein komplexes Problem löse, denke ich über andere Probleme nach, die ich bereits gelöst habe, um zu sehen, ob ich dieses Problem auf ähnliche Weise lösen kann.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich kann die Schritte erklären, die mich zur Lösung des Problems geführt haben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Geben Sie an, wie gut jede der folgenden Aussagen Sie beschreibt:

	Stimmt gar nicht	Stimmt nicht	Stimmt	Stimmt voll und ganz
Die Herausforderung des Programmierens reizt mich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich schreibe gerne Code, um Probleme zu lösen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich bin frustriert und möchte aufgeben, wenn ich einen Fehler in meinem Code habe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wenn mein Code einen Fehler hat, versuche ich, ihn selbst zu beheben, anstatt jemand anderen zu bitten, ihn zu beheben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Im folgenden Abschnitt beziehen wir uns auf "*rechnerisches Denken*" unter Verwendung der folgenden Definition:

Rechnerisches Denken ist ein Problemlösungsprozess, der Folgendes umfasst:

- Probleme so zu formulieren, dass sie mit Hilfe eines Computers gelöst werden können
- Daten logisch zu organisieren und zu analysieren
- Darstellung von Daten durch Abstraktionen, wie Modelle und Simulationen
- Automatisierung von Lösungen durch algorithmisches Denken (eine Reihe von geordneten Schritten)
- Identifizierung, Analyse und Umsetzung von Lösungen, um die effizienteste und effektivste Kombination von Schritten und Ressourcen zu erreichen
- Verallgemeinerung und Übertragung dieses Problemlösungsprozesses auf eine große Vielfalt von Problemen
- - Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.

Geben Sie an, wie gut jede der folgenden Aussagen Sie beschreibt:

	Stimmt gar nicht	Stimmt nicht	Stimmt	Stimmt voll und ganz
Es ist wichtig, rechnerisches Denken zu entwickeln	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich bin in der Lage, anderen rechnerisches Denken beibringen zu können	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich weiß, wie ich das Lernen über rechnerisches Denken interessant gestalten kann	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich bin begeistert, rechnerisches Denken zu lernen und anzuwenden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



TEIL 2


Bei den folgenden Fragen gibt es 4 Antwortmöglichkeiten (A, B, C oder D), von denen nur eine richtig ist. Wenn Sie die Antwort auf eine Frage nicht wissen, geben Sie Ihre beste Vermutung ein.

Bevor wir mit Teil 2 beginnen, stellen wir Ihnen ein Beispiel vor, damit Sie sich mit der Art der Fragen vertraut machen können, die Ihnen begegnen werden.

* Die Fragen in Teil 2 basieren auf dem von Marcos Román-González (2015) entwickelten Computational Thinking Test (CTt).

BEISPIEL

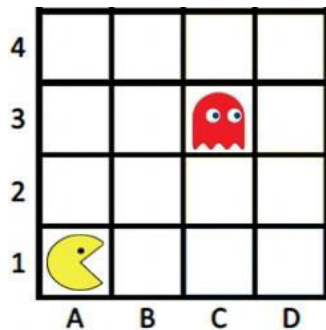
In diesem Beispiel werden Sie gefragt, welchen Anweisungen "Pacman" () folgen soll, um zum "Ghost" () zu gelangen.

Die Anweisung "Move Forward" () weist Pacman an, sich **einen Schritt** in die **Richtung** zu bewegen, in die er gerade blickt. Pacman kann nach links, rechts, oben oder unten blicken. "turn left" weist Pacman an, sich um 90° nach links zu drehen, "turn right" entsprechend 90° nach rechts.

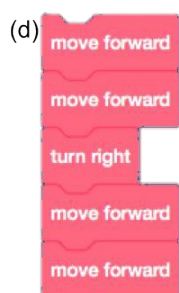
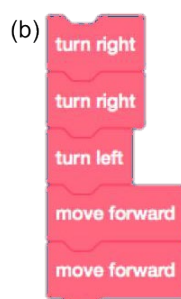
Derzeit startet Pacman an der Position **A1** auf dem Gitter.

Beispielfrage:

Welche Anweisungen führen Pacman zu dem Geist an Position **C3**?

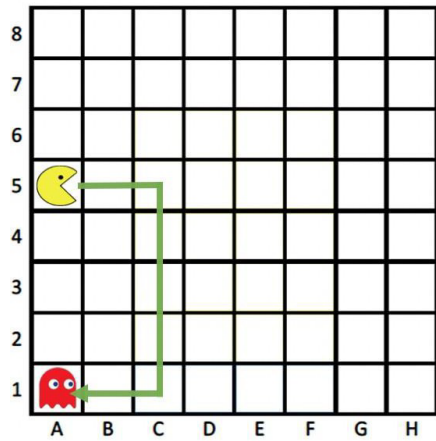


Wählen Sie die richtige Antwort aus (in diesem Beispiel ist die richtige Antwort **C**):



Nun werden Sie selbst einige Fragen beantworten.

Frage 1: Pacman startet auf Position **A5** in der Startaufstellung.



Welche Anweisungen führen Pacman zu dem Geist an der Position **A1**?

(a)

- turn right
- turn right
- move forward
- move forward
- move forward
- move forward
- turn left
- turn left

(b)

- turn right
- turn right
- move forward
- move forward
- move forward
- move forward
- turn right
- turn right

(c)

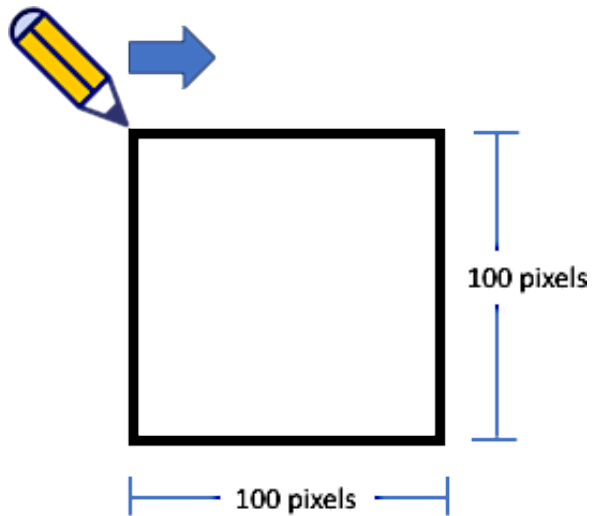
- move forward
- move forward
- turn right
- move forward
- move forward
- move forward
- move forward
- turn right
- move forward
- move forward

(d)

- move forward
- move forward
- turn right
- move forward
- move forward
- move forward
- move forward
- turn left
- move forward
- move forward

Frage 2:

Mit den folgenden Anweisungen sollte der Bleistift das Quadrat **einmal** zeichnen. Der Bleistift beginnt in der linken oberen Ecke und bewegt sich in Richtung des Pfeils.

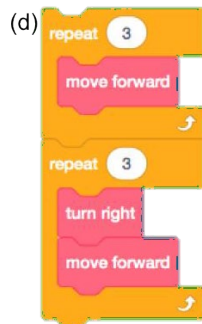
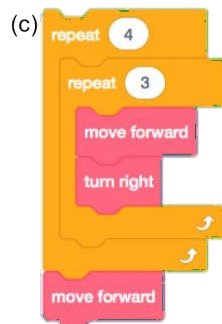
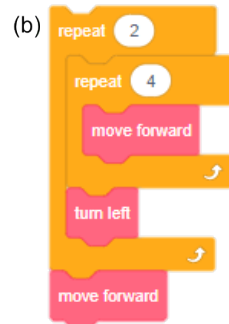
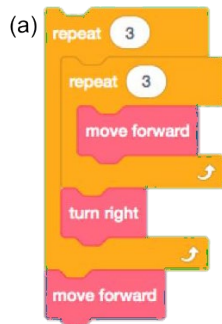
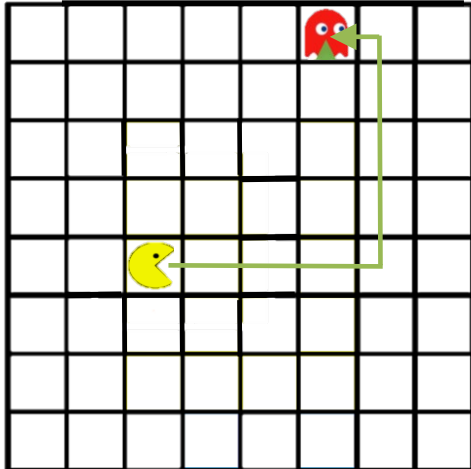


Worin besteht der **Fehler** in der Anleitung?

- (a) Die "repeat"-Schleife sollte 2 Mal wiederholt werden.
- (b) Der "move"-Block sollte außerhalb der Schleife liegen.
- (c) Der Block "turn left" muss durch "turn right" ersetzt werden.
- (d) Der Block "turn left" sollte sich außerhalb der Schleife befinden.

Frage 3:

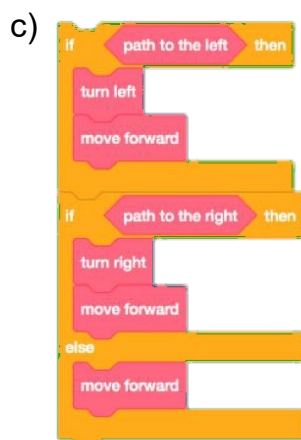
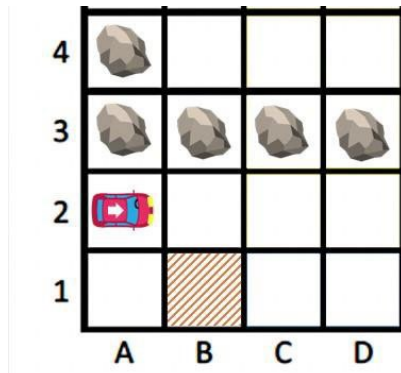
Welche Anweisungen führen den "Pacman" (**B6**) auf dem eingezeichneten Weg zum "Ghost" (**B4**)?



Frage 4:

Ein Auto startet an der Position **A2** und fährt in die *richtige* Richtung, wie der weiße Pfeil auf dem Autodach zeigt. Das Auto kann niemals durch einen Felsen fahren oder die gleiche Position wie ein Felsen einnehmen.

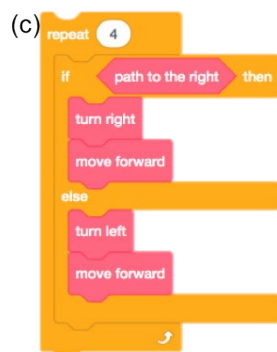
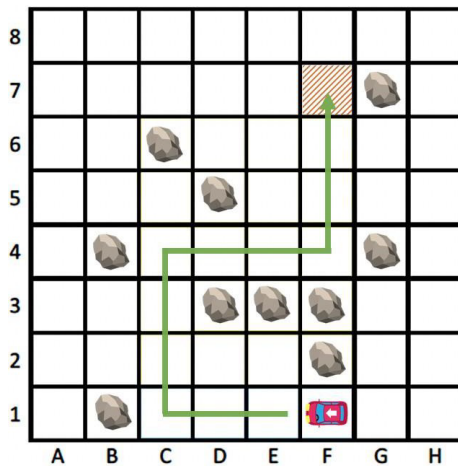
Welche Anweisung führt das Auto richtig zu der orange gemusterten Fliese (**B1**)?



Frage 5:

Ein Auto startet an der Position **F1** und fährt in die *linke* Richtung, wie der weiße Pfeil auf dem Dach des Autos zeigt. Das Auto kann niemals durch einen Felsen fahren oder die gleiche Position wie ein Felsen einnehmen.

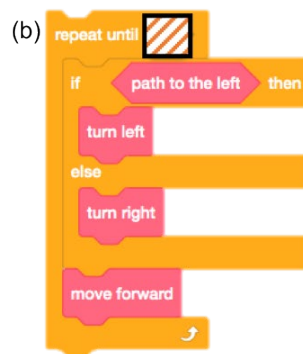
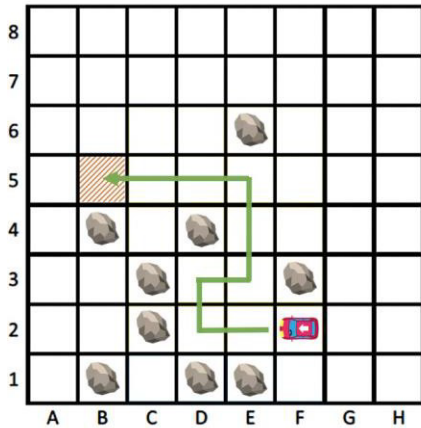
Welche Anweisungen führen das Auto auf dem grün markierten Weg korrekt zum orange gemusterten Feld (**F7**)?



Frage 6:

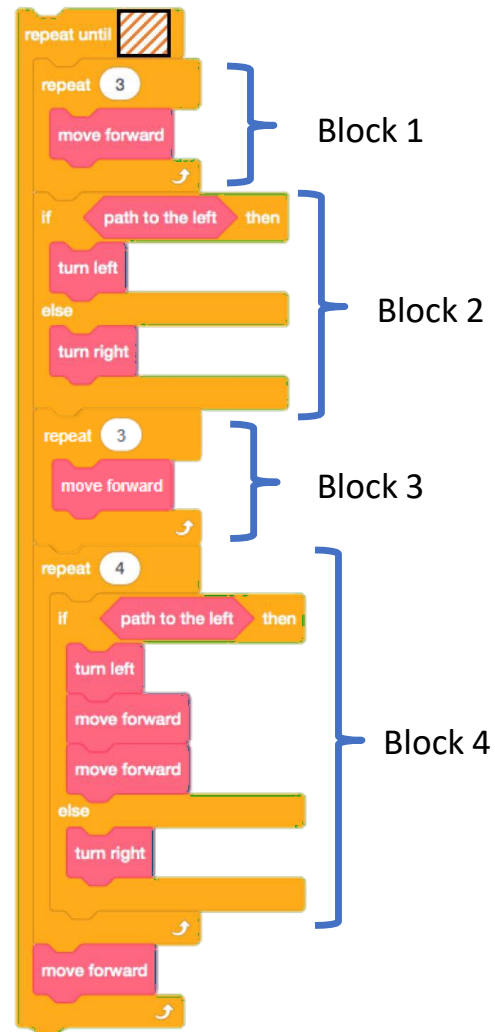
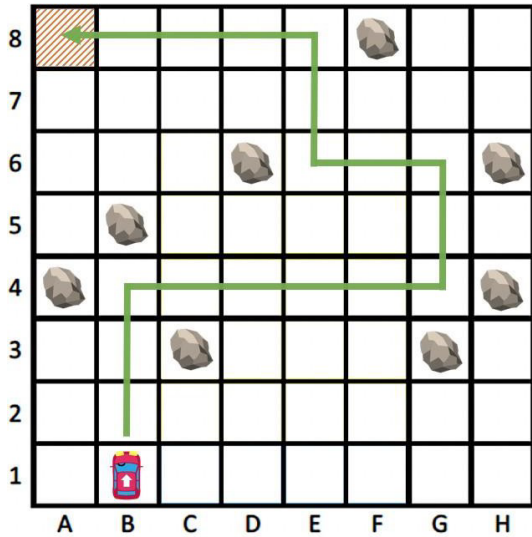
Ein Auto startet an der Position **F2** und fährt in die *linke* Richtung, wie der weiße Pfeil auf dem Auto zeigt. Das Auto kann niemals durch einen Felsen fahren oder die gleiche Position wie ein Felsen einnehmen.

Welche Anweisung führt das Auto auf dem eingezeichneten Weg korrekt zur orange gemusterten Fliese (**B5**)?



Frage 7:

Die folgenden Anweisungen sollen das Auto von **B1** auf das orange gemusterte Plättchen (**A8**) bringen auf dem vorgezeichneten Weg.

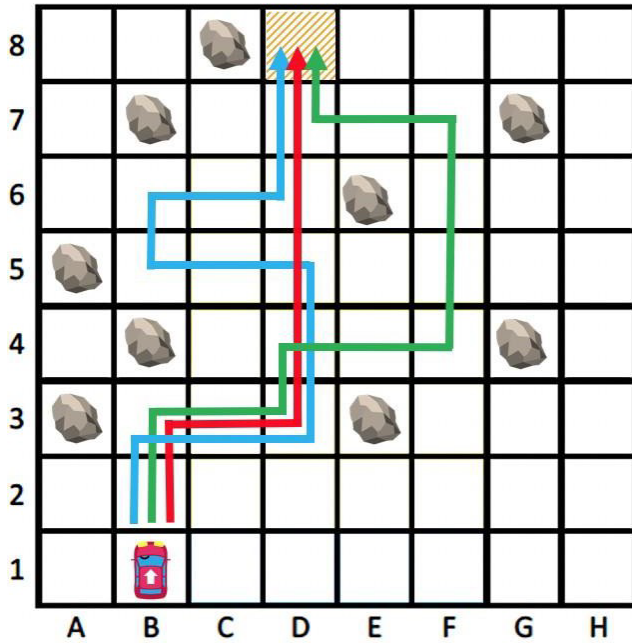


Welcher Block enthält einen **Fehler**?

- (a) Block 1 & 2
- (b) Block 2 & 3
- (c) Block 3 & 4
- (d) Nur Block 4

Frage 8:

Die folgenden Anweisungen bewegen das Auto **(B1)** auf das orange gemusterte Plättchen **(D8)**, indem sie einen der markierten Wege benutzen.

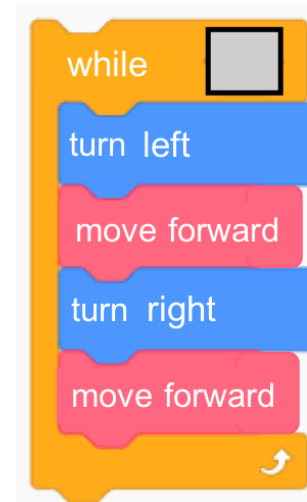
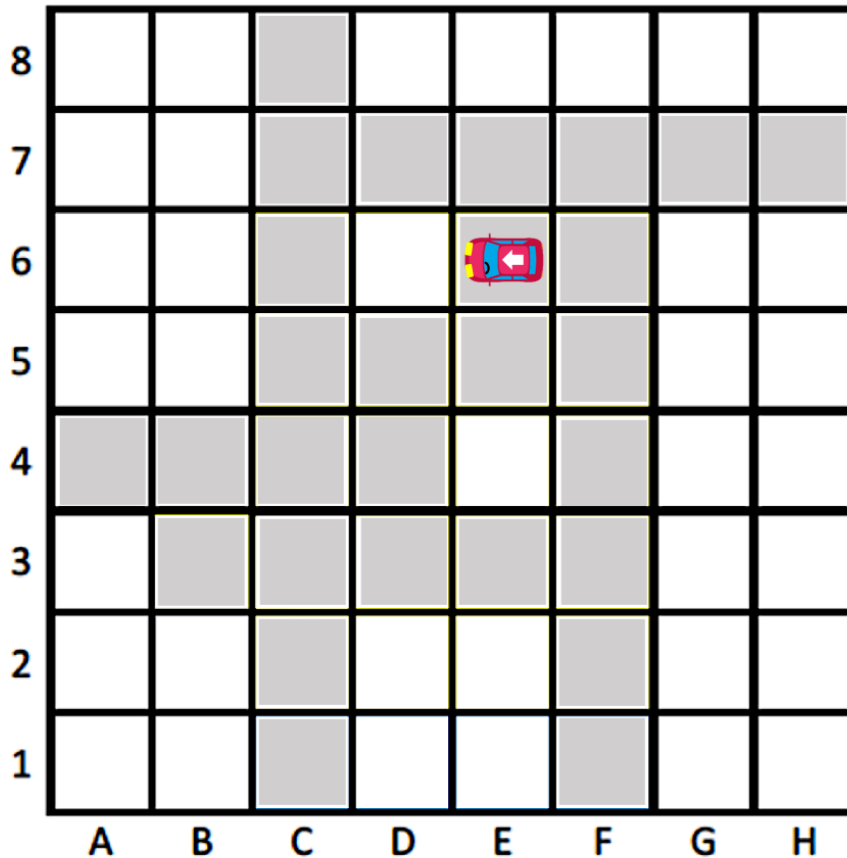


Welchen farbigen Weg wird das Auto laut Anleitung nehmen?

- (a) Den roten Weg
- (b) Den blauen Weg
- (c) Den grünen Weg
- (d) Keinen der oben genannten Wege

Frage 9:

Ein Auto startet an der Position E6, mit Blick nach links.



Wenn das Auto den obigen Anweisungen folgt, wo wird es dann landen?

- (a) A2
- (b) B3
- (c) D4
- (d) E4

Frage 10:

In welche Richtung wird das Auto am Ende der Anleitung schauen?

- (a) Nach oben
- (b) nach rechts
- (c) nach unten
- (d) nach links

Frage 11:

Du blätterst durch ein Kartenspiel und folgst dabei dieser einfachen Regel: Drehe eine Karte um. Wenn die Karte ein Kreuz ist, legst du (ohne sie anzusehen) so viele Karten ab, wie du bereits aufgedeckt hast. Diesen Schritt wiederholst du so lange, bis du keine Karten mehr hast.

Welcher der folgenden Algorithmen setzt die Regel um?

(a)

$n = 0$

solange sich Karten im Stapel befinden:

 Karte umdrehen

$n = n + 1$

 wenn die Karte ein Kreuz ist:

 n-mal wiederholen:

 Karte ablegen

(b)

solange sich Karten im Stapel befinden:

 Karte umdrehen

 wenn die Karte ein Kreuz ist:

 Karte ablegen

(c)

$n = 0$

solange sich Karten im Stapel befinden:

 Karte umdrehen

 n-mal wiederholen:

 Karte ablegen

 wenn die Karte ein Kreuz ist:

$n = n + 1$

(d)

$n = 0$

solange sich Karten im Stapel befinden:

 Karte umdrehen

 solange die aktuelle Karte ein Kreuz ist:

$n = n + 1$

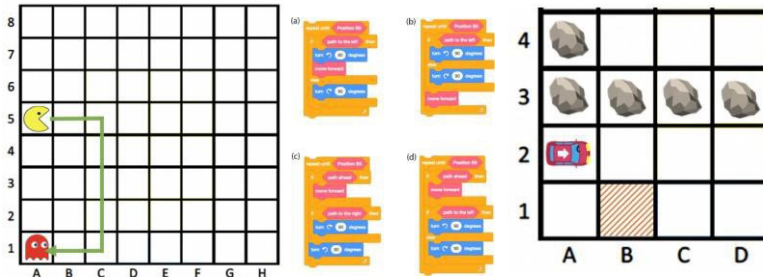
 Karte ablegen

 Karte umdrehen

FEEDBACK

Auf einer Skala von 0 bis 10, wie haben Sie Ihrer Meinung nach die Fragen in Teil 2 beantwortet?

Hinweis: Teil 2 enthält alle Fragen, in denen Sie aufgefordert wurden, Codierungsprobleme wie dieses zu lösen:



0 1 2 3 4 5 6 7 8 9 10

Sehr schlecht

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Sehr gut

Cutumisu, M., Adams, C., Yuen, C., Hackman, L., Lu, C., & Samuel, M. (2019). Callysto Computational Thinking Test (CCTt) Teacher Version [Measurement instrument]. Available: <https://callysto.ca>