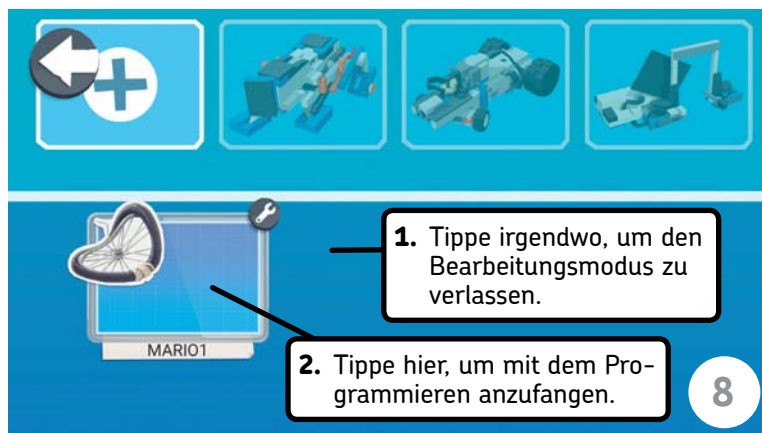


Schritt 7: Tippe auf die in Schritt 6 gezeigte Schaltfläche mit dem Stapel-symbol und wähle das *mittlere* Programmierniveau, also den *Fortgeschrittenenmodus* (das Symbol mit zwei Steinen übereinander). Tippe dann auf das Pinselsymbol mit der Bezeichnung **p**, wähle das kaputte Rad als Projektsymbol aus und tippe auf den Zurück-Pfeil oben links, um wieder auf den vorherigen Bildschirm zurückzugelangen.

Tippe auf das Bleistiftsymbol mit der Bezeichnung **n**, gib den Namen **MARIO1** ein und tippe auf die Eingabeschaltfläche.



Schritt 8: Um die Bearbeitung der Projekteigenschaften abzuschließen, tippst du irgendwo auf den Bildschirm.

Tippe anschließend auf das Projektsymbol, um mit dem Programmieren zu beginnen!

Das erste Programm für MARIO

HINWEIS:

In diesem Buch werden die Programmierblöcke beschrieben, die im Fortgeschrittenenmodus zur Verfügung stehen (zwei gestapelte Steine). Um auf anspruchsvollere Programmierblöcke zugreifen zu können, musst du den Expertenmodus wählen (drei gestapelte Steine).

Wenn die App mit dem Laden fertig ist, siehst du eine leere Leinwand. Ein Pop-up-Fenster fordert dich dazu auf, den Move Hub einzuschalten, indem du auf seine grüne Taste drückst. Folge den Anweisungen auf dem Bildschirm, um den Move Hub mit deinem Gerät zu verbinden. Sollten Probleme auftauchen, schaue auf den offiziellen Hilfeseiten <https://www.lego.com/de-de/themes/boost/support/> und <https://www.lego.com/de-de/service/help/produkte/elektronik-roboter/boost> nach.

Bevor du dein erstes Programm schreibst, sehen wir uns den Programmierbereich an (siehe Abb. 2-3).

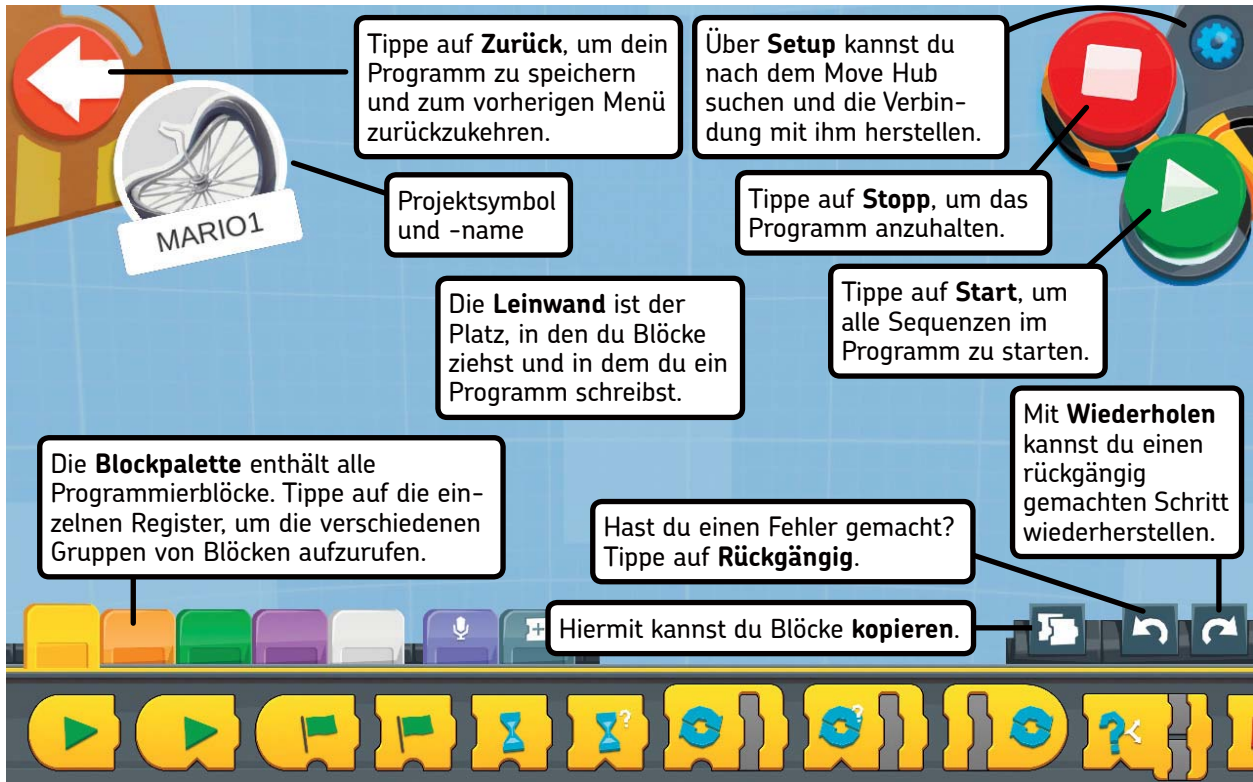


ABBILDUNG 2-3: Die als Kreative Leinwand bezeichnete Programmierumgebung

Um die offizielle Bezeichnung eines Programmierblocks und eine kurze Beschreibung seiner Funktion zu lesen, hältst du in der Palette oder auf der Leinwand den Finger auf ihn.

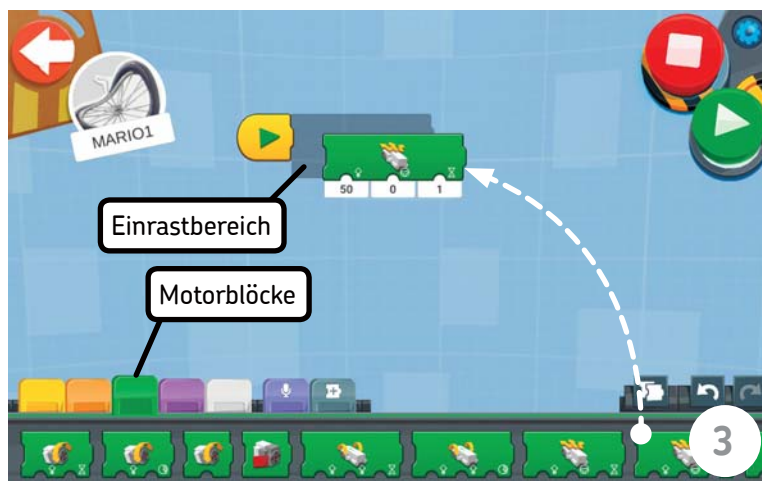
Wir wollen nun ein kurzes Programm schreiben. Befolge die angegebenen Schritte. Die Bilder helfen dir, die richtigen Blöcke zu finden.



Schritt 1: Die erste Palette von links enthält **Programmablaufblöcke**. Tippe darauf, um Blöcke daraus verwenden zu können.



Schritt 2: Ziehe den **Start**-Block auf die Leinwand und lasse ihn los. Mit ihm beginnt die Codesequenz.



Schritt 3: Tippe auf das grüne Register, um zur Palette mit den Motorblöcken zu wechseln, und ziehe den Block **Steuerradlenkung der Antriebsbasis für eine Dauer** auf die Leinwand. Er zeigt am unteren Rand ein Lenkrad und eine Sanduhr. Es gibt vier sehr ähnliche Blöcke, weshalb du genau hinschauen musst, um den richtigen auszuwählen. Ziehe den Block an den zuvor platzierten Block (den **Start**-Block) heran, bis eine graue Fläche erscheint. Jetzt kannst du ihn loslassen, und er wird am vorherigen Block einrasten.



Schritt 4: Der Block **Steuerradlenkung der Antriebsbasis für eine Dauer** nimmt drei Werte entgegen, mit denen du das Verhalten des Roboters ändern kannst. Die Zahl links regelt die Geschwindigkeit des Roboters. Der Vorgabewert beträgt 50.

Um den Roboter zu verlangsamen, ändere die Zahl von 50 in 20. Wenn du auf die Zahl tippst, erscheint ein Zahlenfeld oder ein Schieberegler (siehe Abb. 2-4). Gib **20** ein und tippe dann außerhalb des Zahlenfelds, um es wieder zu schließen.



Schritt 5: Um das Programm zu starten, tippe auf den **Start**-Block im Code.

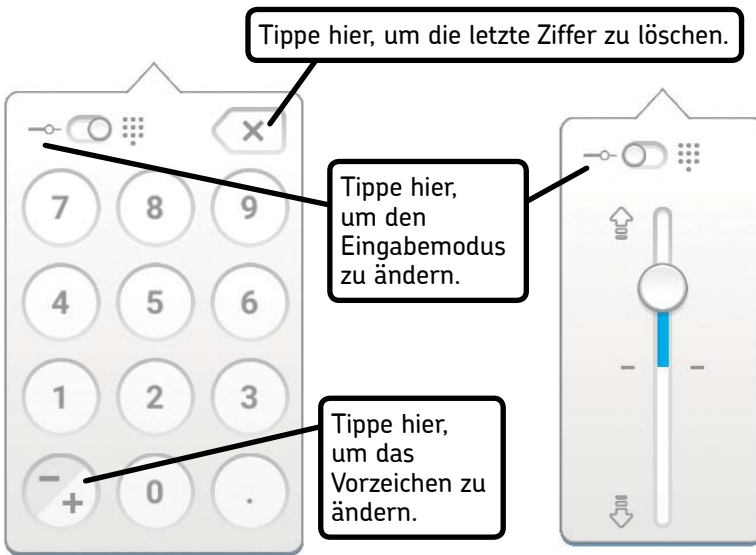


ABBILDUNG 2-4: Um den Geschwindigkeitswert zu ändern, kannst du das Zahlenfeld oder den vertikalen Schieberegler verwenden. Um zwischen den beiden Darstellungsarten umzuschalten, tippst du auf die Schaltfläche oben links in dem Feld.

Abb. 2-5 zeigt das fertige Programm.

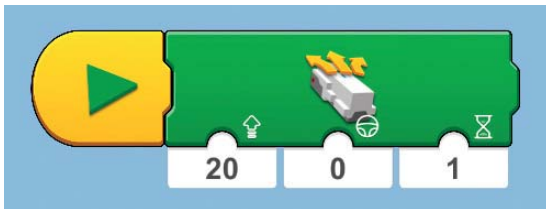


ABBILDUNG 2-5: Das erste Programm für den Roboter MARIO

Wenn du das Programm startest, bewegt sich MARIO eine Sekunde lang mit der Geschwindigkeit 20 geradeaus.

Funktioniert das Programm? Herzlichen Glückwunsch! Damit hast du deinen ersten Schritt in der faszinierenden Welt der Roboterprogrammierung unternommen!

Sollte das Programm nicht funktionieren, so vergewissere dich, dass der Move Hub korrekt angeschlossen ist und dass seine blaue LED leuchtet. Um Verbindungsprobleme zu lösen, schlage auf <https://www.lego.com/de-de/themes/boost/support/> und <https://www.lego.com/de-de/service/help/produkte/elektronik-roboter/boost> nach. Fährt der Roboter nicht wie erwartet geradeaus, kann das an schwachen Batterien des Move Hubs liegen.

Was du gelernt hast

In diesem Kapitel hast du MARIO gebaut, einen einfachen Roboter mit Rädern, mit dessen Hilfe du deine ersten Schritte in der Welt der Programmierung unternimmst. Du hast dabei gelernt, wie du die Kreative Leinwand aufrufst und wie du Projekte erstellst und ihre Eigenschaften änderst, etwa ihren Namen und das Projektsymbol. Außerdem hast du mit dem **Start**-Block und dem Block **Steuerradlenkung der Antriebsbasis für eine Dauer** dein erstes Programm geschrieben.

Im nächsten Kapitel lernst du andere Motorblöcke kennen und erfährst, wie du ihre Parameter änderst, um MARIO herumfahren zu lassen!



Versuch 2-1

Du kannst den Geschwindigkeitswert ändern, um den Roboter schneller oder langsamer zu machen. Was aber passiert, wenn du einen *negativen* Wert angibst?

Die richtige Lösung erfährst du im Anhang auf S. 239.

Passende und unpassende Zahnradkombinationen

Es gibt zwei Arten von Zahnradkombinationen: passende und unpassende. Bei einer passenden Kombination ergibt die Summe der Radien der beiden Zahnräder in LEGO-Einheiten eine *ganze Zahl*. In einem solchen Fall kannst du die Zahnräder auf Achsen montieren, die du in einen Technic-Stein oder Liftarm einsetzt. Bei einer unpassenden Kombination ergibt die Summe der Radien eine Bruchzahl. Dabei ist es nicht möglich, die Zahnräder direkt auf Achsen in den Löchern von Steinen oder Liftarmen zu montieren.

Auch unpassende Kombinationen können nützlich sein, allerdings musst du dabei berücksichtigen, dass sie von den LEGO-Konstrukteuren nicht vorgesehen wurden. Das heißt, dass die Zähne eher locker ineinandergreifen und dass es schwierig sein kann, einen Rahmen zu bauen, der die Zahnräder zusammenhält. In den folgenden Abschnitten findest du auch einige Beispiele für solche unpassenden Kombinationen.

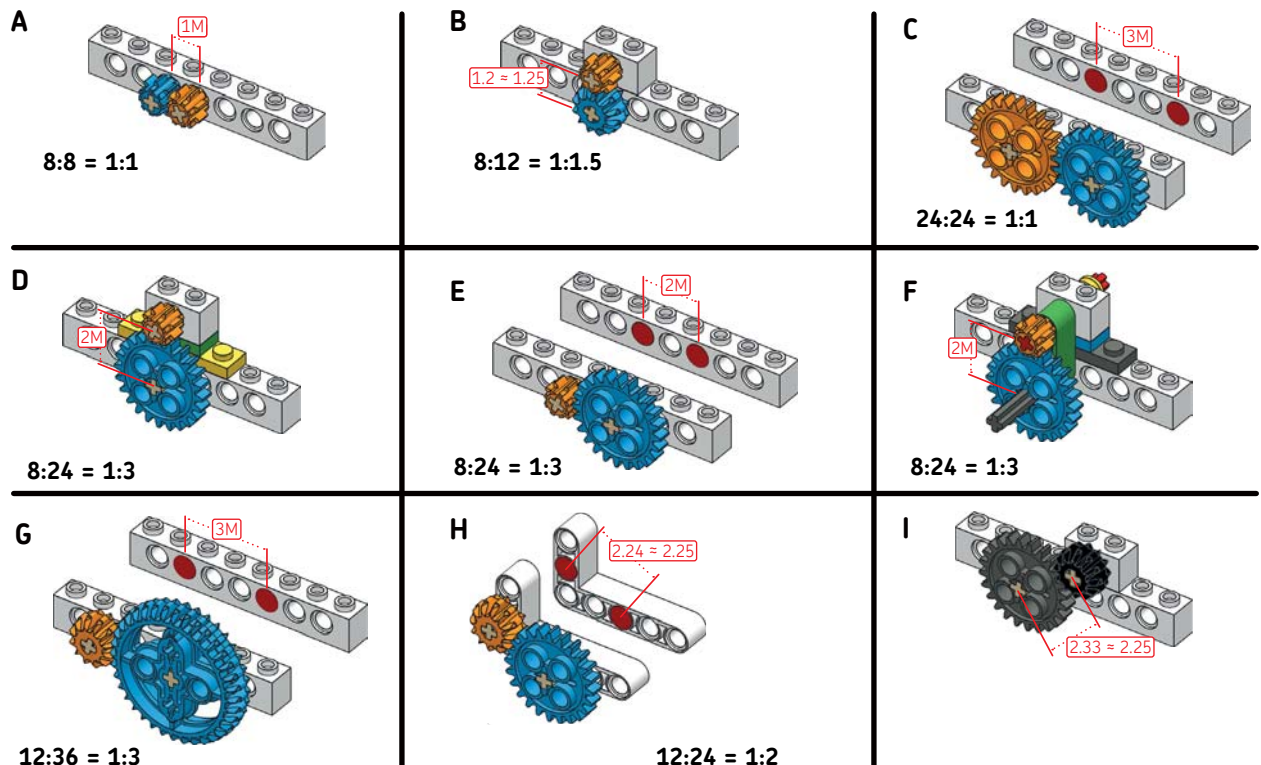
Zahnräder auf parallelen Achsen

In diesem und dem folgenden Abschnitt findest du verschiedene Beispiele für Getriebe, die du in deine Roboter einbauen kannst. Abb. 13-11 zeigt Möglichkeiten, um Zahnräder auf parallelen Achsen zu montieren.

Bei jedem der Beispiele sind das Übersetzungsverhältnis sowie in Rot der Abstand zwischen den Mittelpunkten der Zahnräder in LEGO-Einheiten angegeben. Beispielsweise leistet das Getriebe in **F** die gleiche Übersetzung wie das in **D**, ist allerdings durch den 3M-Liftarm, der die Achsen verbindet, dagegen gesichert, dass sich die Zahnräder voneinander lösen.

Die Getriebe in **B**, **H** und **I** bestehen aus Zahnrädern mit unpassendem Verhältnis, funktionieren aber trotzdem.

ABBILDUNG 13-11: Zahnräder auf parallelen Achsen. In Rot sind die Abstände zwischen den Achsen in LEGO-Einheiten angegeben.



Zahnräder im rechten Winkel

Abb. 13-12 zeigt einige Beispiele für die Montage von Zahnrädern im rechten Winkel zueinander. Bei einigen dieser Getriebe wurden dünne Stopper oder ein 1×2-Technic-Stein mit zwei Löchern eingebaut, um für einen zusätzlichen Abstand von einer halben Noppenbreite zu sorgen, sodass die Zahnräder ineinandergreifen können.

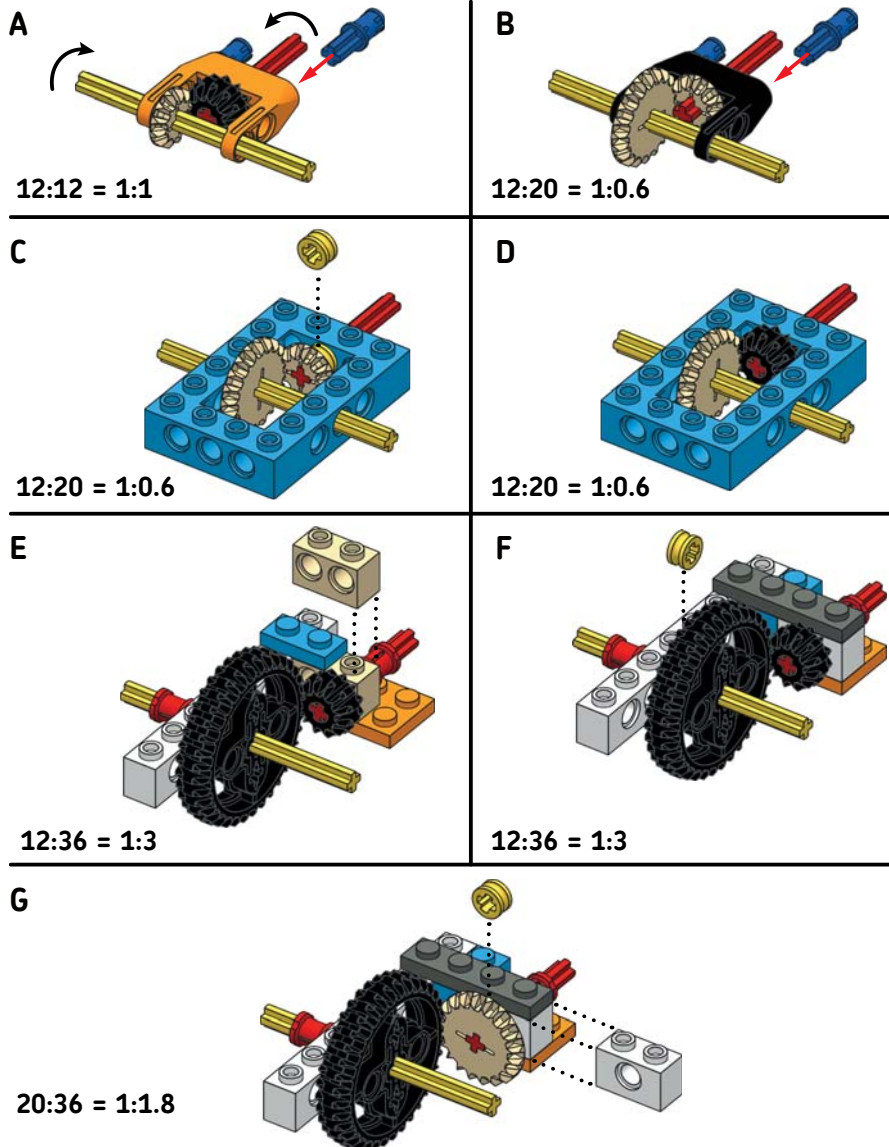


ABBILDUNG 13-12: Im rechten Winkel gekoppelte Zahnräder



14

BrickPecker: Ein Vogel sortiert Steine

In diesem Kapitel erhältst du eine ausführliche Bauanleitung für BrickPecker, den »Steinspecht« (siehe Abb. 14-1), der Steine in seinem Nest nach Farben sortieren kann. Du fütterst den BrickPecker über eine Rutsche mit bunten »Samenkörnern«, die der Roboter mit seinem Schnabel aufnimmt und dann sortiert in den verschiedenen Behältern ablegt.

In jedem Schritt der Bauanleitung ist eine Liste aller Teile angegeben, die du in dem jeweiligen Schritt benötigst.

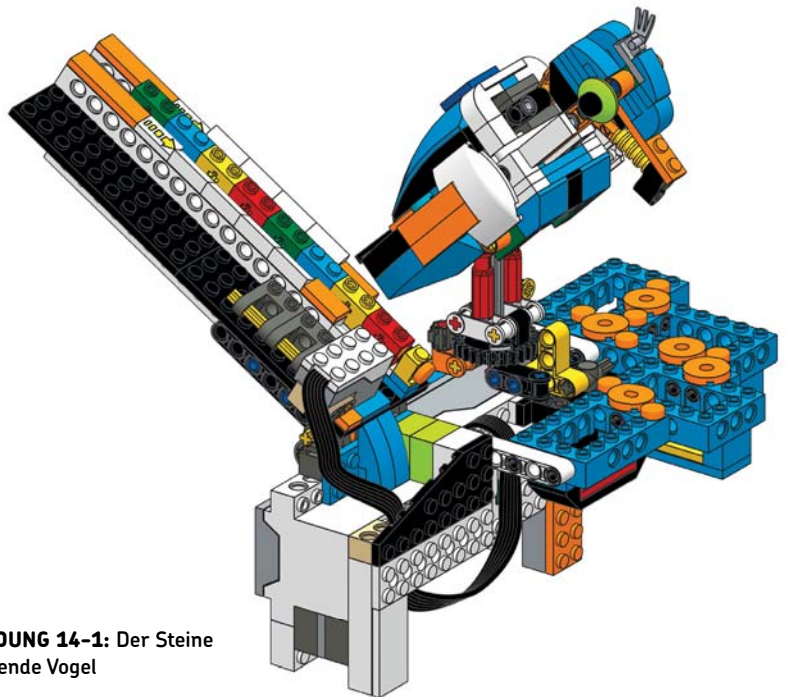
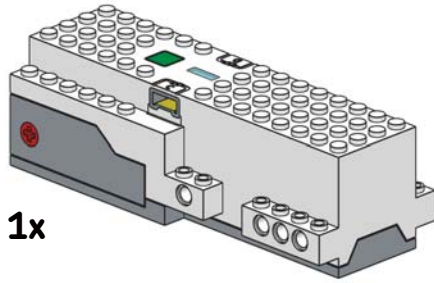
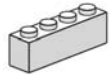


ABBILDUNG 14-1: Der Steine sortierende Vogel

1



1x



1x



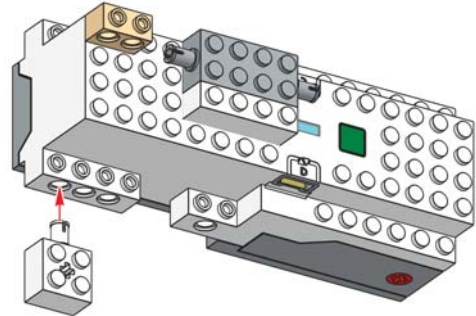
1x



1x



1x



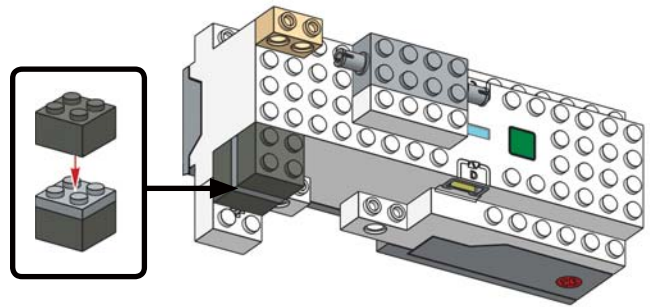
2



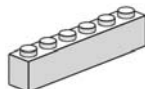
2x



1x



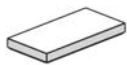
3



2x



1x



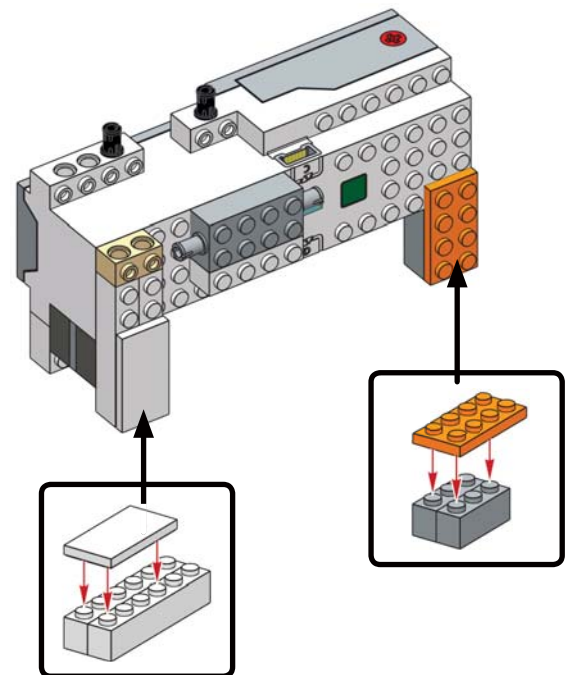
1x



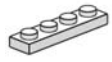
2x



2x



4



1x



1x



1x



2x



1x



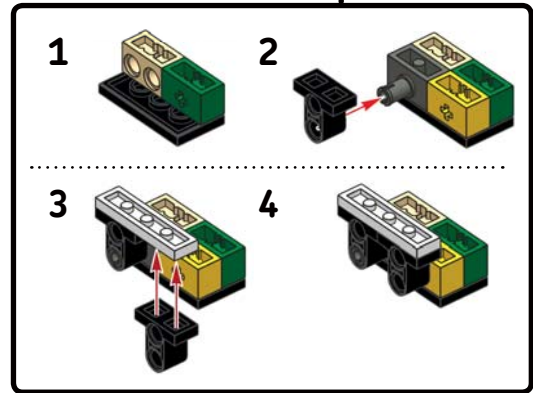
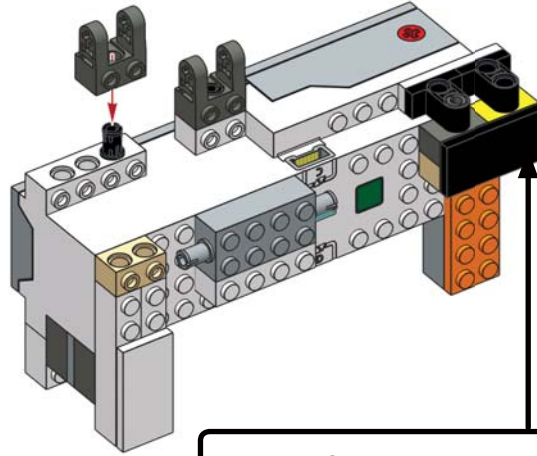
1x



2x



1x



5



1x



2x



1x



2x



1x

