PRZEWODNIK NAUCZYCIELA

28.08.2018

BeCREO TECHNOLOGIES

Spis treści

Organizacja pracy z grą Scottie Go! — 7

Uwagi metodyczne — 9

Moduł I - Europa 1 — 11

Wprowadzenie do modułu - instrukcje — 12

Zadanie 1 - sterowanie bohaterem, pierwsze instrukcje — 13

Zadanie 2 — 15

Zadanie 3 - cyfry, tworzenie liczb, dodawanie i odejmowanie — 16

Zadanie 4 - skręcanie — 18

Zadanie 5 — 20

Zadanie 6 — 21

Zadanie 7 — 23

Zadanie 8 - chodzenie do tyłu — 25

Zadanie 9 — 27

Zadanie 10 — 29

Moduł II - Europa 2 — 31

Wprowadzenie do modułu - podnoszenie obiektów na scenie — 32

Zadanie 1 — 33

Zadanie 2 — 35

Zadanie 3 — 37

Zadanie 4 — 39

Zadanie 5 - skakanie — 41

Zadanie 6 — 44

Zadanie 7 — 46

Zadanie 8 — 49

Zadanie 9 — 52

Zadanie 10 — 55

Spis treści

Moduł III - Ameryka Południowa — 57

Wprowadzenie do modułu - pętla POWTÓRZ — 58

Zadanie 1 - pierwsza pętla — 59

Zadanie 2 — 62

Zadanie 3 - praca w i poza pętlą — 64

Zadanie 4 - wprowadzanie pojęć matematycznych i fizycznych — 67

Zadanie 5 — 70

Zadanie 6 — 72

Zadanie 7 - pętle zagnieżdżone — 74

Zadanie 8 — 77

Moduł IV - Ameryka Północna — 79

Zadanie 1 - bóbr zostaje mostem — 80

Zadanie 2 — 82

Zadanie 3 — 84

Zadanie 4 - bóbr w ruchu — 86

Zadanie 5 — 89

Zadanie 6 — 92

Zadanie 7 - bóbr wykonuje obrót — 95

Zadanie 8 — 98

Zadanie 9 — 100

Zadanie 10 — 102

Moduł V - Antarktyda — 104

Wprowadzenie do modułu - interakcja z otoczeniem - klocki UŻYJ oraz UMIEŚĆ — 105

Zadanie 1 - klocek UŻYJ — 107

Zadanie 2 - klocek UMIEŚĆ — 110

...

Spis treści

Moduł V - Antarktyda c.d. — 104

...

Zadanie 3 — 113

Zadanie 4 — 115

Zadanie 5 — 117

Zadanie 6 — 120

Zadanie 7 - steruj bobrami i przesuwaj kry — 122

Zadanie 8 — 125

Zadanie 9 — 127

Zadanie 10 — 130

Moduł VI - Azja — 132

Wprowadzenie do modułu - pętla warunkowa - POWTÓRZ DOPÓKI — 133

Zadanie 1 - używamy klocka POWTÓRZ DOPÓKI — 135

Zadanie 2 — 138

Zadanie 3 — 141

Zadanie 4 - POWTÓRZ DOPÓKI z innym warunkiem — 144

Zadanie 5 - POWTÓRZ DOPÓKI w pętli — 147

Zadanie 6 — 150

Zadanie 7 — 153

Zadanie 8 — 155

Moduł VII - Australia 1 — 158

Wprowadzenie do modułu - nowe pojęcie: ZMIENNE — 159

Zadanie 1 - WPROWADZENIE - asfaltowa droga — 162

Zadanie 2 — 165

Zadanie 3 — 167

Zadanie 4 - X i Y — 170

...

Spis treści

Moduł VII - Australia 1 c.d. — 158

...

Zadanie 5 — 173

Zadanie 6 - zmienne i pętle — 175

Zadanie 7 — 178

Zadanie 8 - działania matematyczne na zmiennych — 180

Zadanie 9 — 181

Zadanie 10 — 185

Moduł VIII - Australia 2 — 187

Wprowadzenie do modułu - instrukcje warunkowe — 188

Zadanie 1 - Pole akcji — 192

Zadanie 2 — 195

Zadanie 3 — 197

Zadanie 4 — 199

Zadanie 5 — 201

Zadanie 6 - używanie wielu instrukci warunkowych — 204

Zadanie 7 — 207

Zadanie 8 - klocki WZNIESIENIE i TUTAJ — 209

Moduł IX - Ameryka Północna 2 — 212

Wprowadzenie do modułu (A) - W PRZECIWNYM RAZIE JEŻELI — 213

Wprowadzenie do modułu (B) - instrukcja W PRZECIWNYM RAZIE — 216

Wprowadzenie do modułu (C) - [opcjonalnie] - POWTÓRZ (CIĄGLE) — 219

Zadanie 1 - W PRZECIWNYM RAZIE — 222

Zadanie 2 - W PRZECIWNYM RAZIE JEŻELI i KOPERTY — 226

Zadanie 3 — 231

...

Spis treści

Moduł IX - Ameryka Północna 2 c.d. — 212

...

Zadanie 4 — 233

Zadanie 5 — 235

Zadanie 6 — 238

Zadanie 7 - taśmociągi — 240

Zadanie 8 - dźwignie — 243

Zadanie 9 — 246

Moduł X - Afryka — 248

...

Organizacja pracy z grą Scottie Go! 1/2

Dbaj o dobre oświetlenie stanowisk

Zabawa ze Scottie Go! wymaga wykonywania zdjęć napisanych przez uczniów programów. Niskie natężenie światła może uniemożliwić ich poprawne zeskanowanie.

Nie łącz klocków z różnych zestawów

Każdy klocek w zestawie ma unikalny numer identyfikacyjny i znak graficzny rozpoznawany przez kamerę Twojego urządzenia. Kiedy połączysz klocki z kilku zestawów w jeden programy mogą nie być prawidłowo rozpoznawane. Oznacz klocki z danego pudełek, np. jednym kolorem na pleckach klocka.

NIGDY NIE ŁĄCZ KLOCKÓW WERSJI DOMOWEJ Z WERSJĄ EDUKACYJNĄ

Optymalny model pracy to 2 osoby na jedno pudełko gry

Dbaj, aby uczniowie zestaw reguł związanych z pracą z zestawami Scottie Go!, np. spiszcie wspólnie kontrakt pracy i zabawy ze Scottiem. Jedną z reguł winno być, żeby w czasie pracy uczniowie nie pożyczali sobie klocków, a po skończonych zajęciach chowali je do pudełka wkładając je w odpowiednie gniazda. Jeśli pracujesz z młodszymi dziećmi warto przygotować odpowiednie klocki wcześniej.

Stwórz specjalne konto nauczyciela, które otwiera rozwiązania wszystkich modułów w grze

W każdej chwili możesz utworzyć konto nauczyciela z pełnym dostępem do wszystkich modułów i zadań. Konto - Nauczyciel - pozwala zapoznać się z wybranymi zadaniami bez konieczności wcześniejszego zrealizowania całego kursu Scottie Go! Dzięki takiej funkcjonalności możemy korzystać z gry na różnych przedmiotach i wykonywać konkretne, zaplanowane działania oraz koncepty programistyczne, w zależności od wieku i możliwości uczniów.

Aby utworzyć konto **Nauczyciel** wybierz opcję w zainstalowanej aplikacji **Scottie Go! Edu**

WYBIERZ GRACZA->NOWY GRACZ i wpisz specjalny kod:

Nauczyciel_13_cps#

NIE UDOSTĘPNIAJ TEGO KODU UCZNIOM!

Organizacja pracy z grą Scottie Go! 2/2

Konto nauczyciela daje możliwość pracy na jednym urządzeniu z uczniami. Na koncie nauczyciela wyświetlimy treść zadania i z jego pomocą przypiszemy postępy poszczególnym kontom uczniów.

Konto nauczyciela doskonale współpracuje z kartami postępu Scottie Go!
Wszystkie materiały dodatkowe możemy pobrać po zalogowaniu się na stronie gry:
www.scottiego.com/education

Uwagi metodyczne 1/2

Nie ma jednego prawidłowego rozwiązania

W poradniku podane zostały przykładowe rozwiązania załączonych zadań. Istnieje jednakże wiele sposobów realizacji tych samych problemów różnymi napisanymi programami – także krótszymi.

Gra zawiera 179 klocków, które są aktywne w każdym module i zadaniu

Możemy wielokrotnie powracać do każdego zadania realizując je ponownie z wykorzystaniem poznanych konceptów algorytmicznych z innych zadań i modułów oraz nowo poznanych klocków.

Wersja edukacyjna wykorzystuje formułę gry, jednak jest to pełnoprawne narzędzie edukacyjne

Z wykorzystaniem gry Scottie Go! edu, realizujemy warunki polskiej podstawy programowej oraz wykorzystujemy najnowsze zdobycze światowej metodyki nauki programowania poprzez wdrożenie myślenia komputacyjnego - umiejętności rozwiązywania problemów z różnych dziedzin z wykorzystaniem technologii informatycznej oraz świadomego korzystania z metod i narzędzi cyfrowych poprzez:

• rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

• programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

• rozwijanie kompetencji społecznych, takich jak: komunikacja i współpraca w grupie, w tym w środowiskach wirtualnych, udział w projektach zespołowych oraz organizacja i zarządzanie projektami.

PAMIĘTAJ, ŻE GRA TO POMOC, A NIE CEL SAM W SOBIE

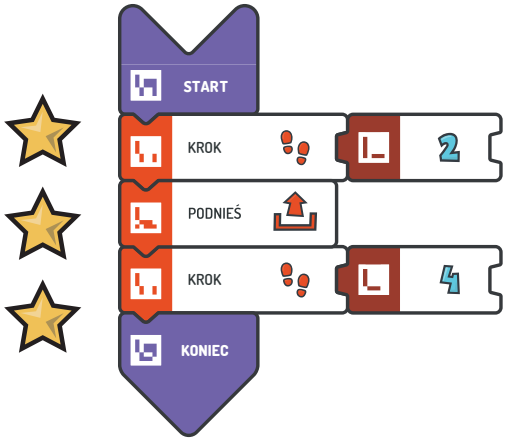
Wykorzystanie gry na lekcji powinno być zgodne z zasadami pracy metodami aktywnymi, celami lekcji, jej przebiegiem i ewaluacją. Układ gry nawiązuje do spiralnego modelu nauczania - w jednym zadaniu pojawia się nowa instrukcja, a w kolejnym (lub kolejnych) testujemy zdobytą wiedzę w praktyce. Stopniowo poziom trudności wzrasta. Zarówno nauczyciel, jak i uczeń rozwijają swoje kompetencje algorytmiczne. Wprowadzenie nowej instrukcji sygnalizuje wykrzyknik w górnym prawym rogu na grafice postaci jednego z bohaterów gry.

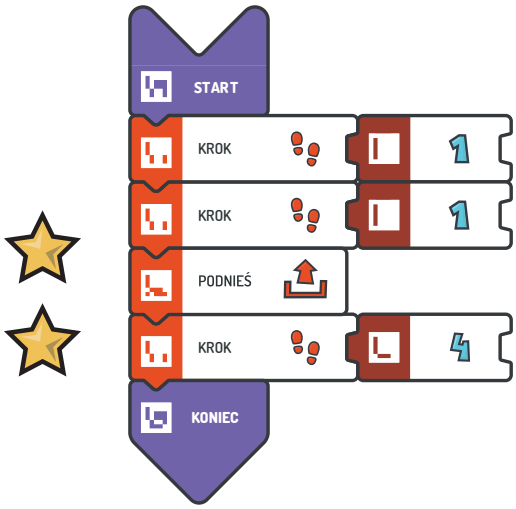
Uwagi metodyczne 2/2

Dobry programista i ambitny uczeń powinien zawsze szukać optymalnej drogi rozwiązania. W Scottie Go! przez optymalną drogę, rozumiemy takie skonstruowanie programu, które w jak najmniejszej ilości instrukcji pozwoli zrealizować określony cel.

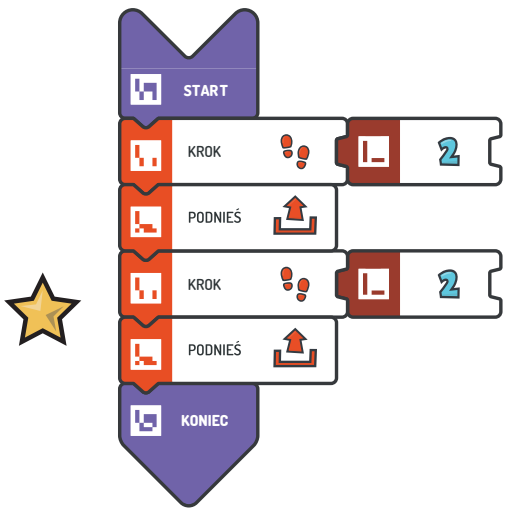
Nauczyciel w trakcie lekcji powinien budować atmosferę sprzyjającą uczeniu się. Aplikacja **ScottieGo! Edu**, realizuje elementy oceniania kształtującego w najbardziej atrakcyjnym dla ucznia środowisku gry. Uczniowie na koniec zadania otrzymują od 1 do 3 gwiazdek nagradzających jakość i optymalizację programu. To dobra, i pozbawiona rywalizacji zabawa.

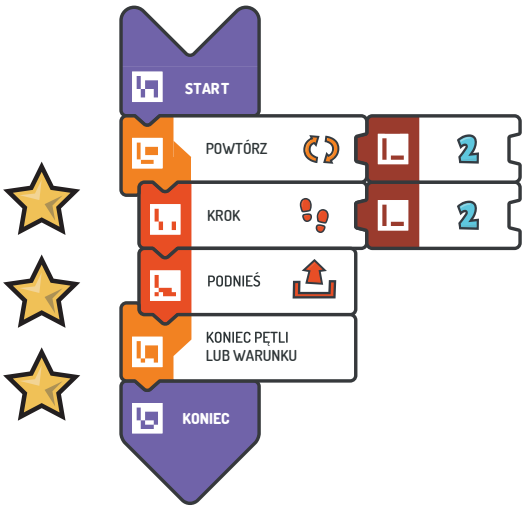
Po każdym wykonanym zadaniu gracz otrzymuje gwiazdki - trzy gwiazdki za najlepiej napisany program, który spełnia warunki zadania, dwie za nieco słabsze rozwiązanie. Jedna gwiazdka oznacza, że zadanie zostało wykonane najczęściej za pomocą zbyt długiego programu.





W niektórych zadaniach, aby otrzymać więcej niż jedną gwiazdkę, do rozwiązania zadania wymagane będzie wykorzystanie konkretnych klocków, na przykład klocka **POWTÓRZ**!





Moduł VI
Azja

Potrzebuję Twojej pomocy. Bambus to doskonale giętka, a zarazem wytrzymała roślina, która pozwoli zrobić rusztowanie w celu odbudowy statku. Zdobądź proszę dla mnie wszystkie łodygi bambusa.

WPROWADZENIE DO MODUŁU 1/2: Pętla warunkowa - POWTÓRZ DOPÓKI

Podobnie jak pętla **POWTÓRZ**, pętla **POWTÓRZ DOPÓKI** sprawia, że powtarzane zostają wielokrotnie wybrane instrukcje z tym, że będą one wykonywane przez cały czas, dopóki spełniony będzie pewien warunek. Tworzymy w ten sposób **pętle warunkowe**.

Przykład **pętli warunkowej** w codziennej sytuacji:

Wyobraźmy sobie jazdę samochodem do domu z pracy. Często trudno nam określić jakie konkretne kroki i ile razy trzeba wykonać, aby dotrzeć do domu. Najczęściej jednak w takiej sytuacji podświadomie stosujemy pętlę P**OWTÓRZ DOPÓKI**.

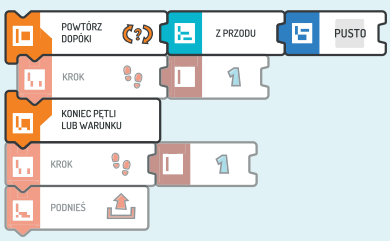
1. Sprawdzamy warunek – „czy przede mną jest prosta droga?”
2. Jeśli warunek jest spełniony – jedziemy prosto i wracamy znów do punktu (1) – sprawdzania warunku: „czy przede mną jest prosta droga?”.
3. Jeśli warunek nie jest spełniony, czyli przed nami jest zakręt – kończy się działanie pętli, warunek już nie jest sprawdzany, a wykonywane zostaje zadanie poza pętlą – czyli skręt zgodnie z kierunkiem zakrętu.

Wyobraźmy sobie jazdę samochodem do domu z pracy. Często trudno nam określić jakie konkretne kroki i ile razy trzeba wykonać, aby dotrzeć do domu. Najczęściej jednak w takiej sytuacji podświadomie stosujemy pętlę P**OWTÓRZ DOPÓKI**.

Pętle warunkowe w **Scottie Go!**

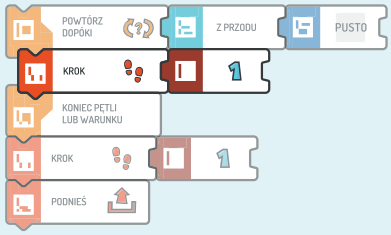
Przyjrzyjmy się zastosowaniu tej pętli na przykładzie zadania. W tym zadaniu Scottiemu będzie wolno używać tylko zestawienia klocków **KROK** i **cyfra 1**. (To ułatwi nam wytłumaczenie zastosowania pętli.)

1. Sprawdzamy warunek – „czy przede mną jest prosta droga?” **Warunek jest spełniony** (pole przed Scottiem jest puste).

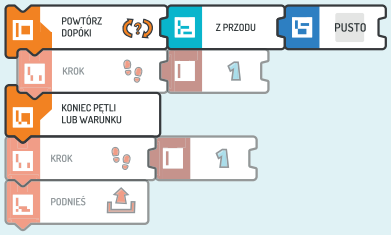


WPROWADZENIE DO MODUŁU 2/2: Pętla warunkowa - POWTÓRZ DOPÓKI

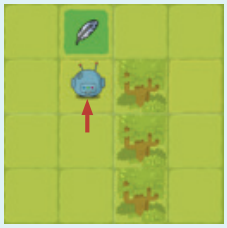
2. Wykonywane są instrukcje wewnątrz pętli. Scottie idzie jeden krok do przodu. Program wraca do sprawdzania warunku (punktu 1.).

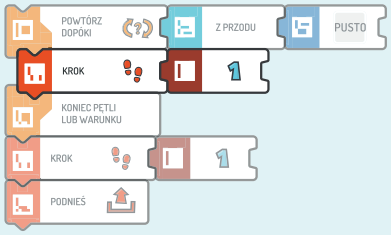


3. Następnie znów sprawdzanie warunku: „z przodu pusto?”. **Warunek jest spełniony.**

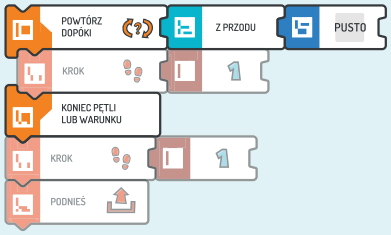


4. Wykonywane są instrukcje wewnątrz pętli – **KROK 1**, a następnie wraca do sprawdzania warunków (punkt 1).

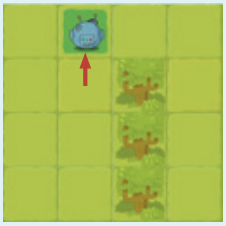


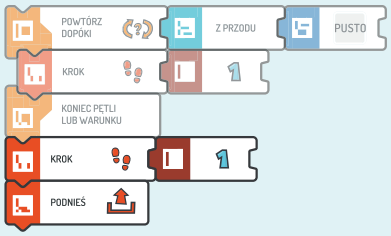


5. Znów sprawdzany jest warunek: „z przodu pusto?”. **Warunek <u>nie jest</u> spełniony.**



6. Pętla zostaje przerwana. Wykonywany jest program poza pętlą: **KROK (1)** i **PODNIEŚ.**

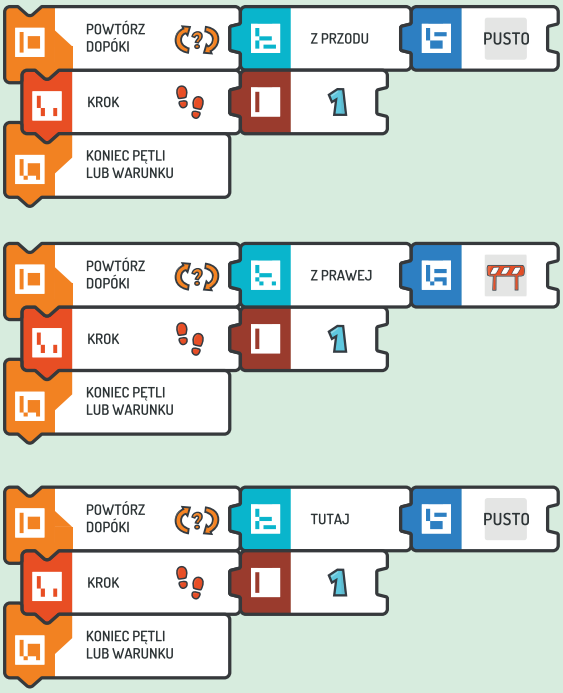




WPROWADZENIE: używamy klocka POWTÓRZ DOPÓKI

W kolejnych zadaniach używaj wyłącznie kombinacji klocków **KROK (1)** oraz pętli **POWTÓRZ DOPÓKI**

Czasami zamiast **POWTÓRZ DOPÓKI (Z PRZODU < PUSTO)** możesz zastosować połączenie **POWTÓRZ DOPÓKI (Z PRAWEJ < PRZESZKODA)** lub **POWTÓRZ DOPÓKI (TUTAJ < PUSTO)**.



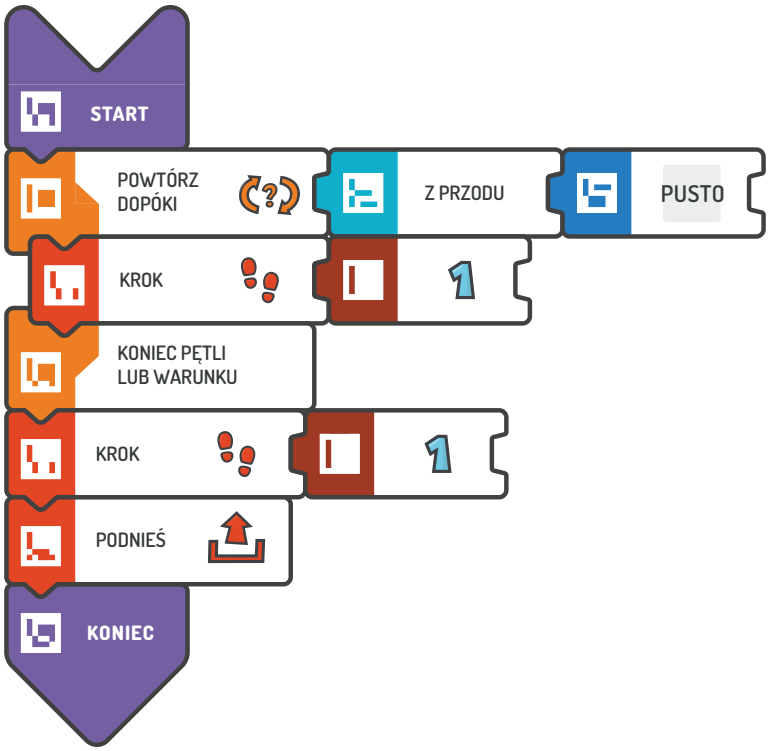
W ostatnim przypadku możesz potencjalnie skrócić cały program o jedną linię.

Przed wprowadzeniem pętli warunkowej POWTÓRZ DOPÓKI koniecznie ją omów wykorzystując przykład „z życia" podany we wprowadzeniu do modułu, lub stwórz własny przykład.

Konsekwentne użycie zestawienia **KROK (1)** z **POWTÓRZ DOPÓKI** na przestrzeni wielu następujących po sobie zadań sprawi, że zastosowanie pętli wyda się oczywiste. Wymagaj w tym module od uczniów używania wyłącznie **KROK (1)**.

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    KROK (1)
    PODNIEŚ
KONIEC
```

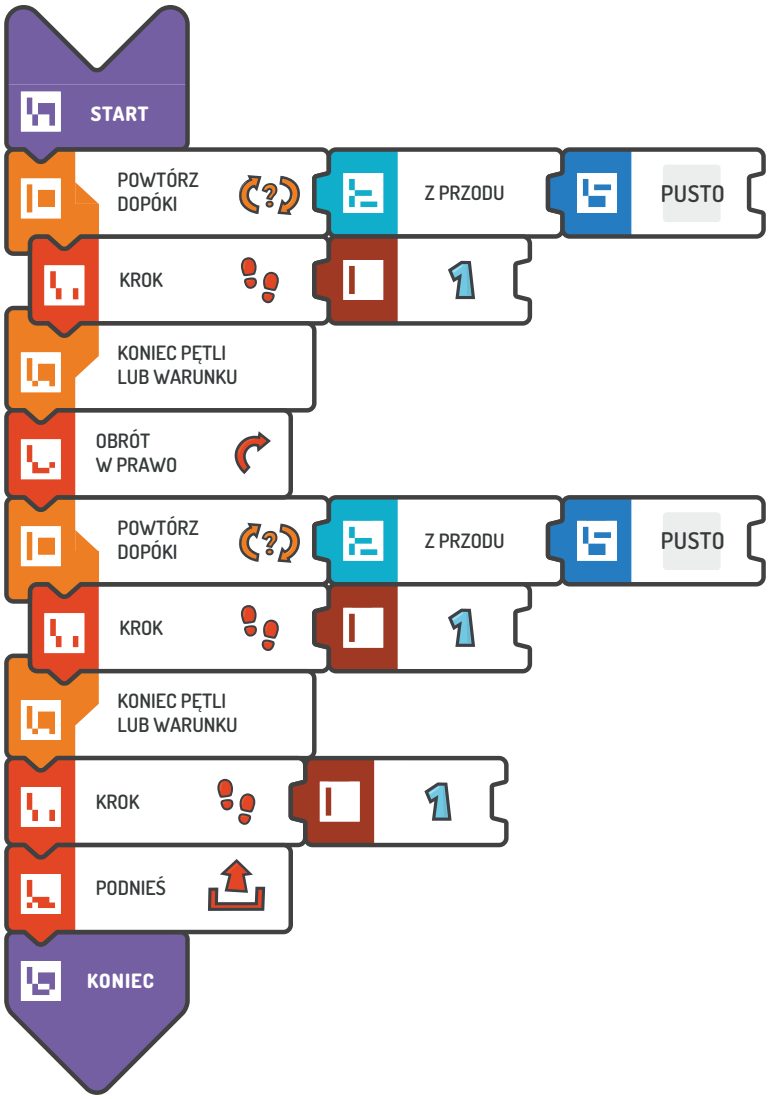
WSKAZÓWKA

W tym zadaniu wykorzystaj dwie pętle **POWTÓRZ DOPÓKI**.

Opcjonalnie, zamiast **POWTÓRZ DOPÓKI (Z PRZODU < PUSTO)** w tym zadaniu w drugiej pętli można wykorzystać połączenie **POWTÓRZ DOPÓKI (TUTAJ < PUSTO)**. Po odpowiednich zmianach program będzie krótszy o jedną linię.

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

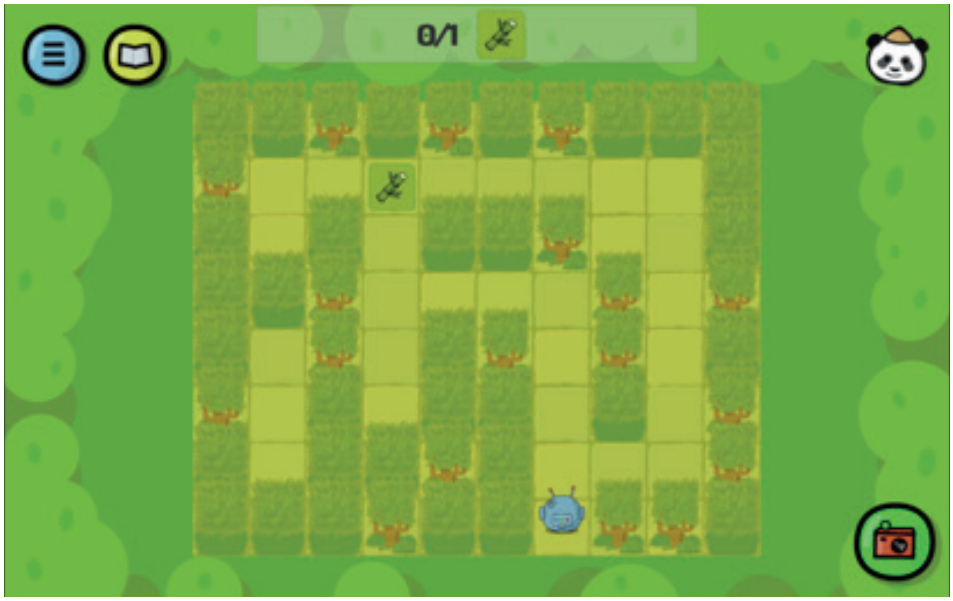
```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    OBRÓT W PRAWO
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    KROK (1)
    PODNIEŚ
KONIEC
```

WSKAZÓWKA

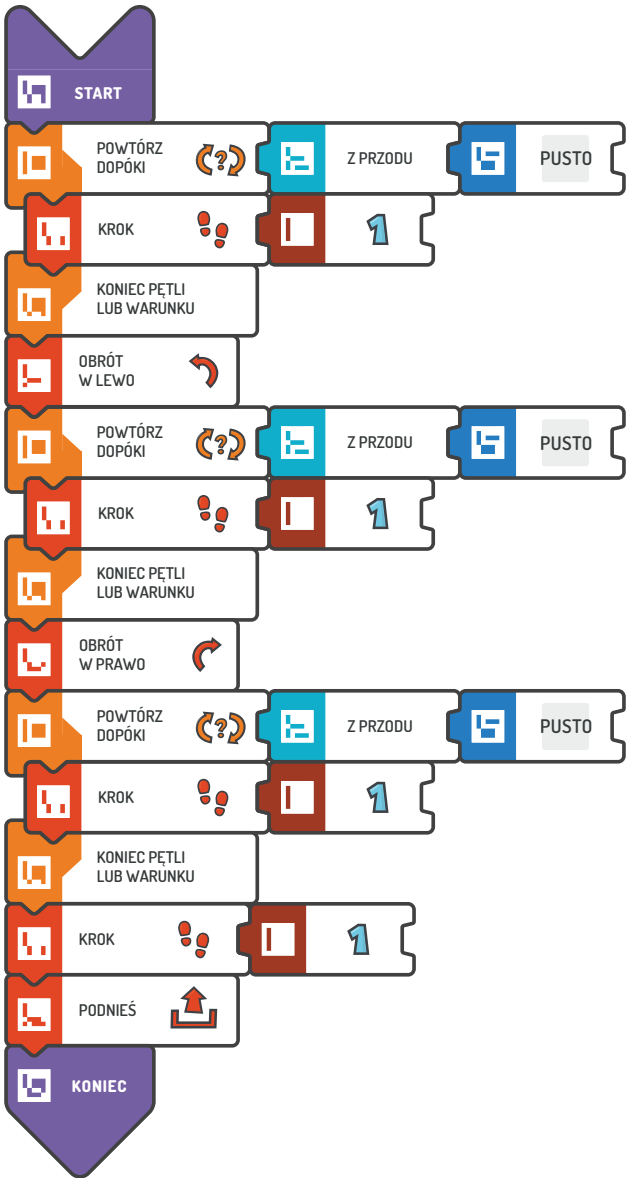
Jak widzieliśmy w poprzednich zadaniach **POWTÓRZ DOPÓKI** jest sprytniejszą wersją pętli **POWTÓRZ**, gdzie dopóki spełniony jest warunek, wykonywana jest lista poleceń, a gdy warunek nie jest spełniony, pętla kończy swoje działanie.

Wykorzystaj trzy pętle **POWTÓRZ DOPÓKI**.

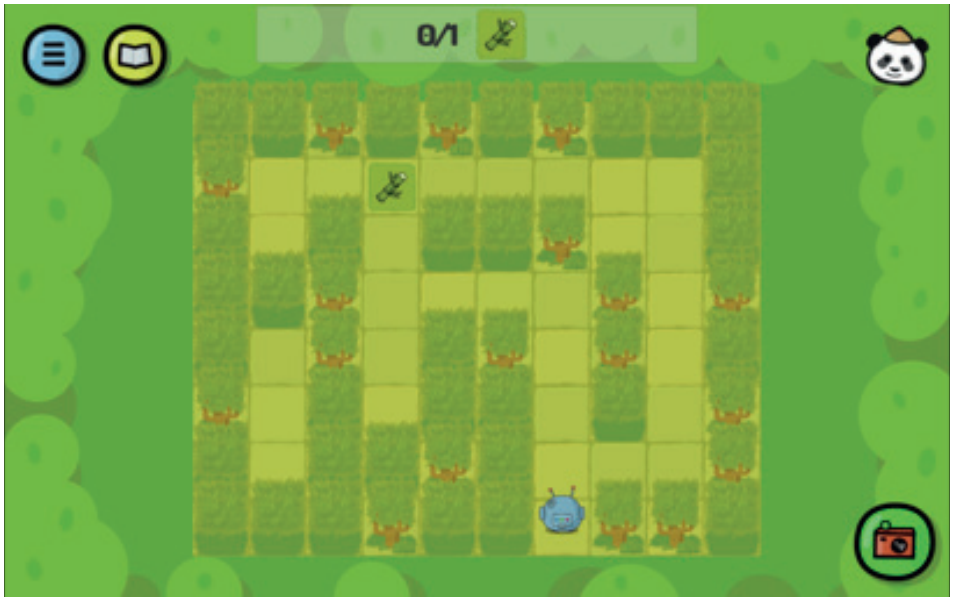
Wyzwanie:



Przykładowe rozwiązanie:



Wyzwanie:



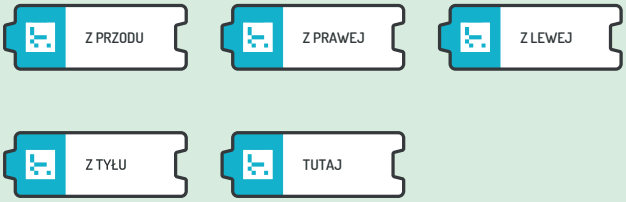
Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    OBRÓT W LEWO
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    OBRÓT W PRAWO
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
    }
    KROK (1)
    PODNIEŚ
KONIEC
```

WPROWADZENIE: POWTÓRZ DOPÓKI z innym warunkiem

Możesz budować pętle **POWTÓRZ DOPÓKI** połączone z innymi warunkami koniecznymi do zrealizowania.

Gdzie (z punktu widzenia Scottiego)?



Co?

puste pole

przedmiot do zebrania

przeszkoda

wzniesienie (wpóźniejszych modułach)

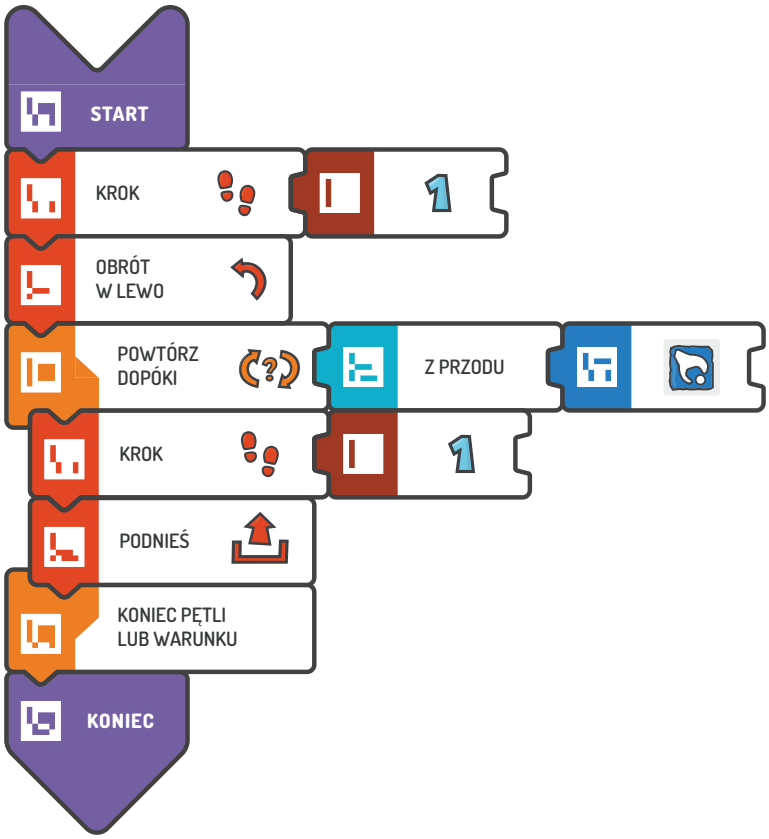
koperta: kwadrat (w późniejszych modułach)

koperta: krzyżyk (w późniejszych modułach)

pole akcji (w późniejszych modułach)

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

```
START
    KROK (1)
    OBRÓT W LEWO
    POWTÓRZ DOPÓKI (Z PRZODU < PRZEDMIOT DO ZEBRANIA) {
        KROK (1)
        PODNIEŚ
    }
KONIEC
```

WPROWADZENIE: POWTÓRZ DOPÓKI w pętli

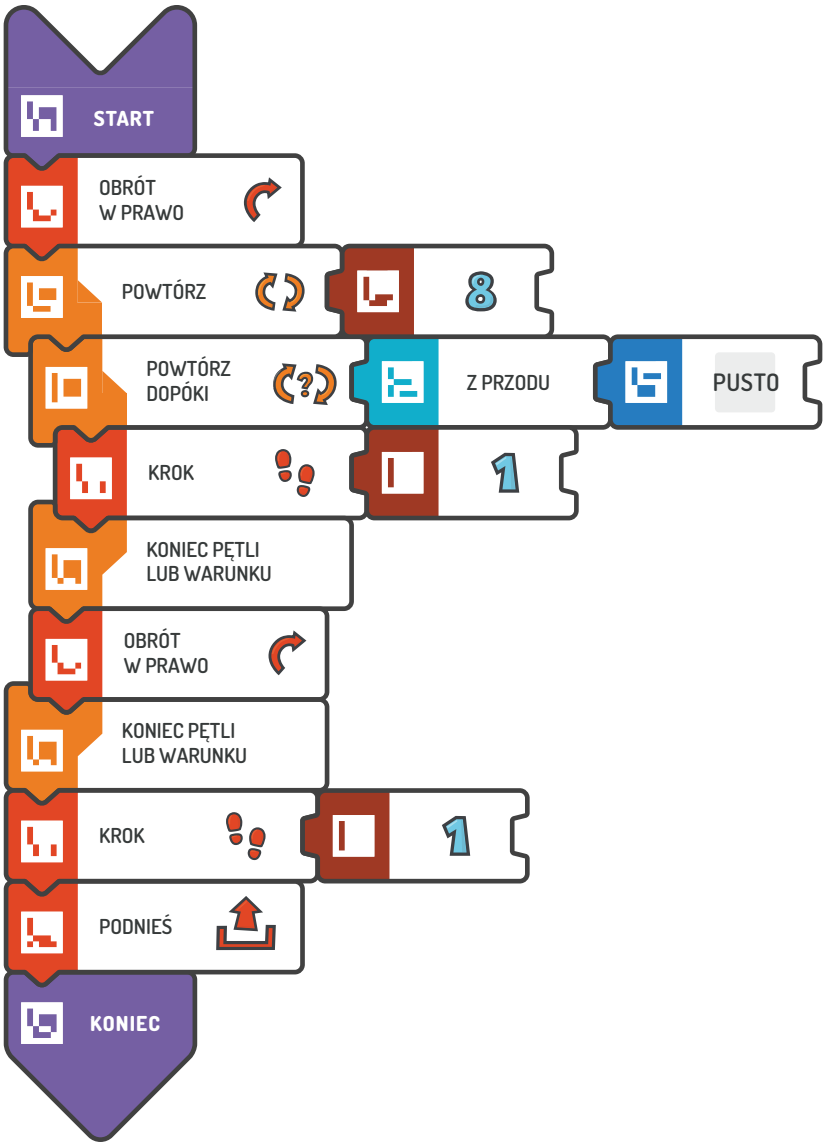
Jeśli trzeba, możesz wielokrotnie wywoływać pętle **POWTÓRZ DOPÓKI**. Wykorzystaj pętlę **POWTÓRZ** i w niej umieść kolejną pętlę **POWTÓRZ DOPÓKI** (Pętla w pętli - spróbuj, to nie takie trudne!).

Uczniowie bardziej zaawansowani mogą próbować rozwiązać to zadanie wykorzystując instrukcje warunkowe typu:

```
JEŻELI (Z PRAWEJ < PUSTO) {
        SKRĘĆ W PRAWO
}
```

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

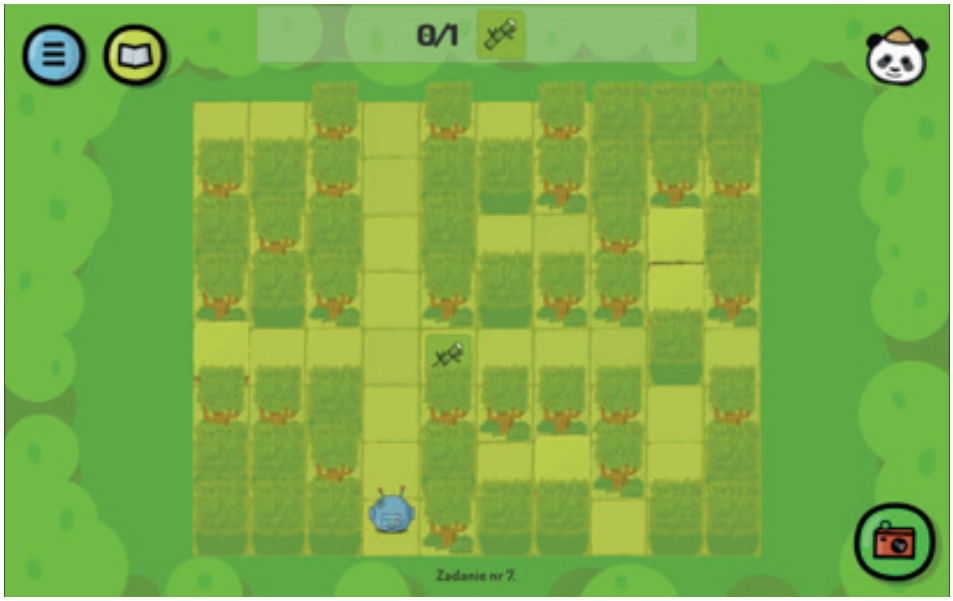
Przykładowe rozwiązanie (format tekstowy):

```
START
    OBRÓT W PRAWO
    POWTÓRZ (8) {
        POWTÓRZ DOPÓKI (Z PRZODU < PUSTO)  {
            KROK (1)
        }
        OBRÓT W PRAWO
    }
    KROK (1)
    PODNIEŚ
KONIEC
```

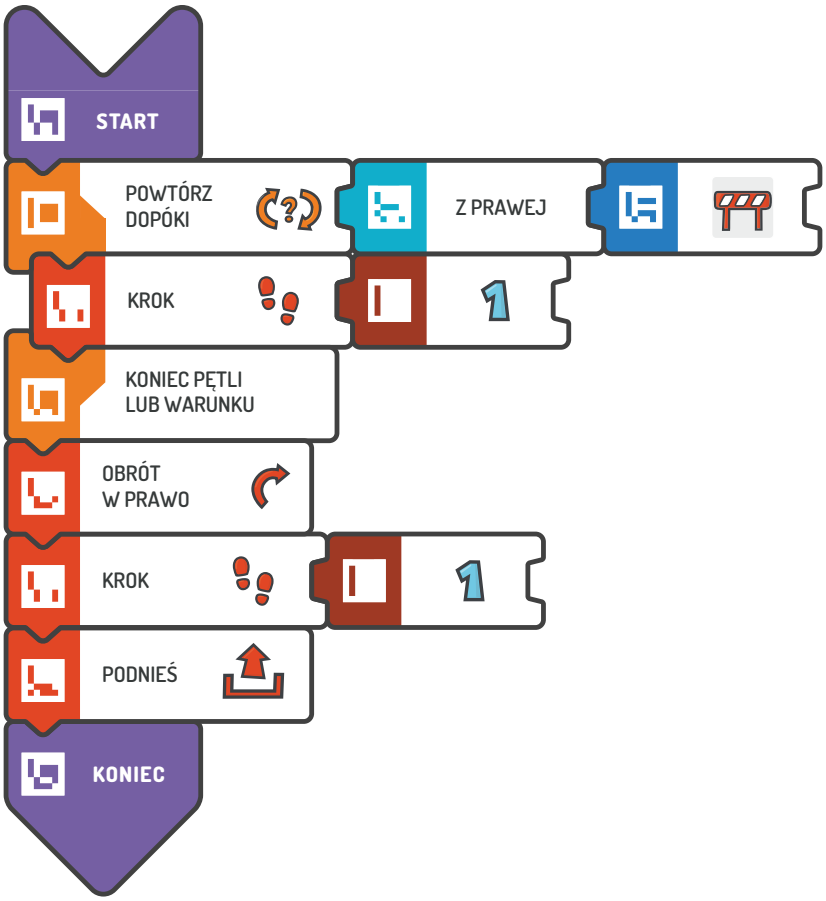
UWAGA

Wykorzystaj pętlę **POWTÓRZ DOPÓKI** tylko raz.

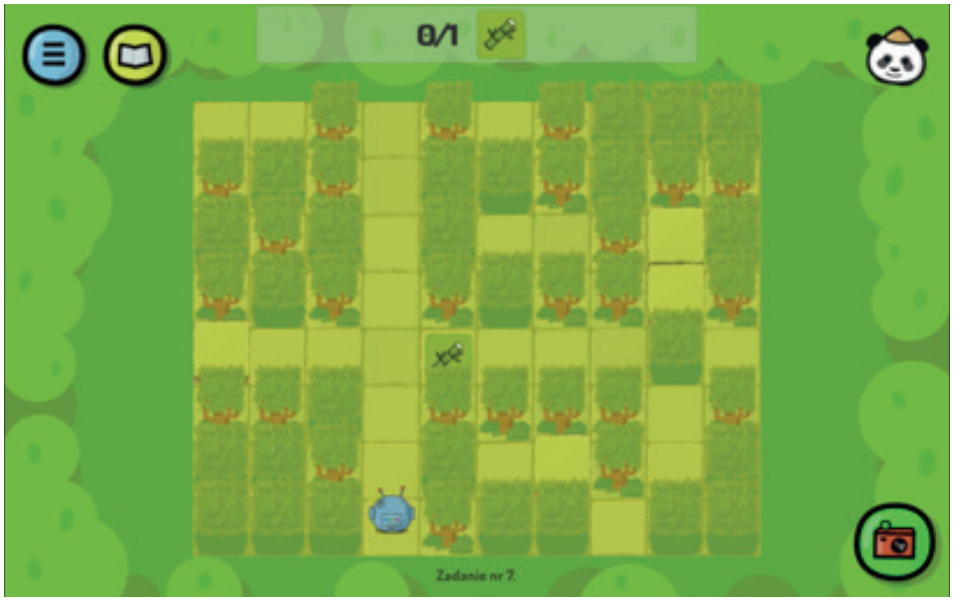
Wyzwanie:



Przykładowe rozwiązanie:



Wyzwanie:

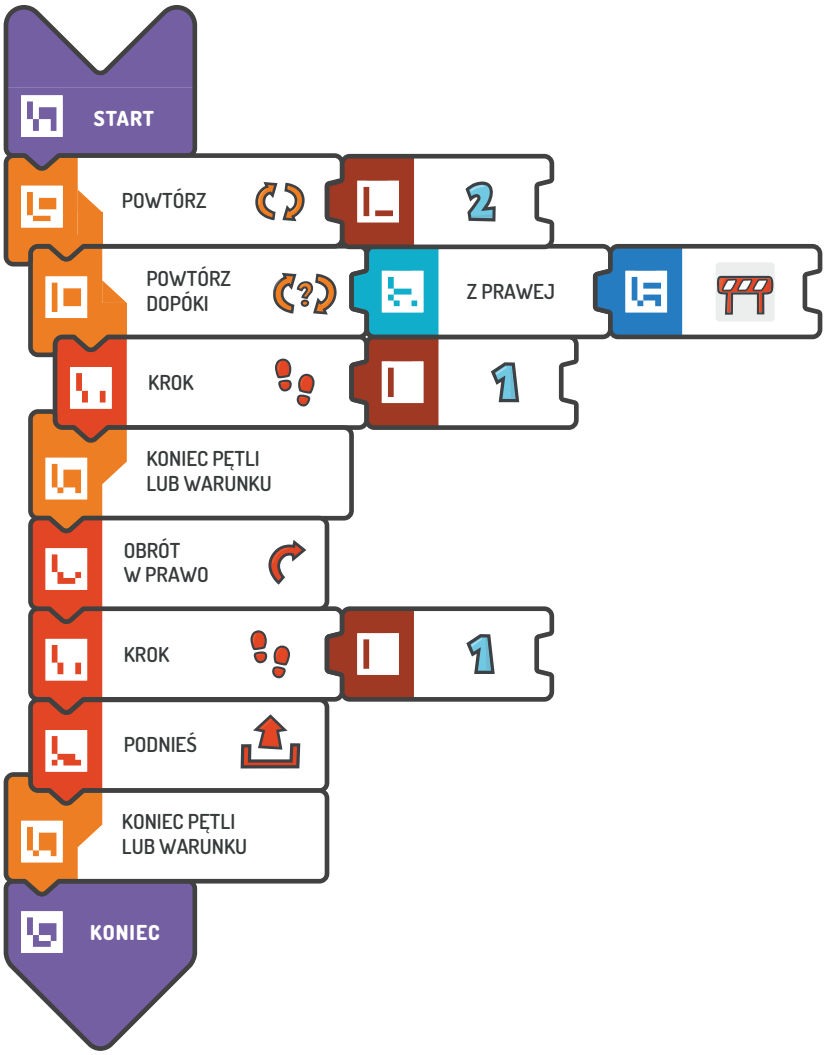


Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRAWEJ < PRZESZKODA) {
        KROK (1)
    }
    OBRÓT W PRAWO
    KROK (1)
    PODNIEŚ
KONIEC
```

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

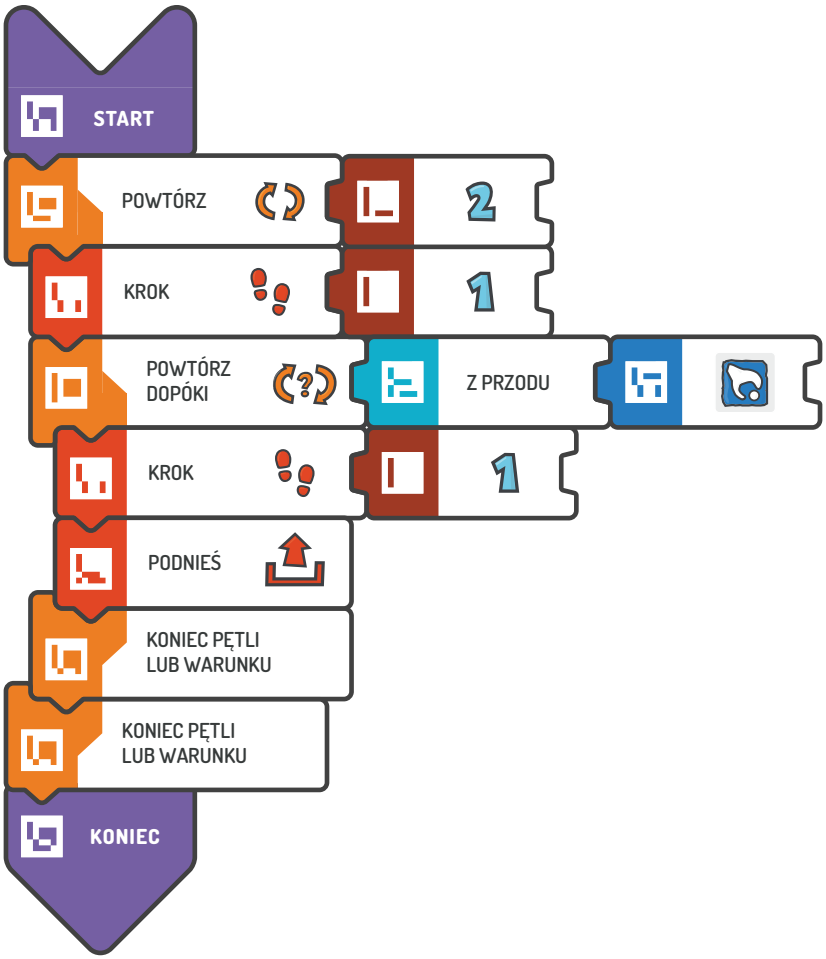
```
START
    POWTÓRZ (2) {
        POWTÓRZ DOPÓKI (Z PRAWEJ < PRZESZKODA) {
            KROK (1)
        }
        OBRÓT W PRAWO
        KROK (1)
        PODNIEŚ
    }
KONIEC
```

UWAGA

W tym zadaniu wykorzystaj pętlę **POWTÓRZ DOPÓKI** tylko jeden raz i pętlę **POWTÓRZ** również jeden raz.

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (2) {
        KROK (1)
        POWTÓRZ DOPÓKI (Z PRZODU < PRZEDMIOT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
        }
    }
KONIEC
```

Moduł VII
Australia 1

W tym module używamy pojęcia ZMIENNEJ.
Zbuduj dla Scottiego kamienne drogi, tak aby w trakcie prac szybciej mógł poruszać się po wiosce i zbierać części swojego statku.

WPROWADZENIE DO MODUŁU 1/3: Nowe pojęcie: ZMIENNE

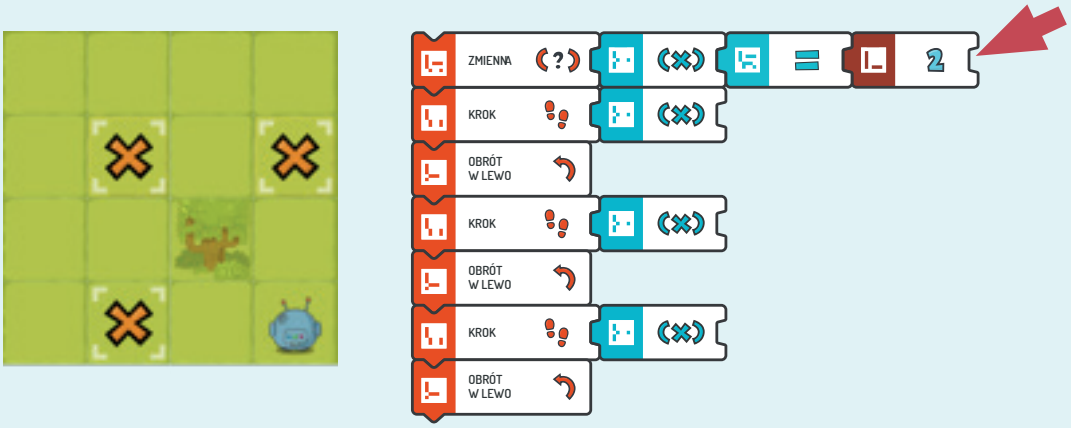
Zmienne można sobie wyobrazić jako pojemniki. Do takiego pojemnika można wrzucić różne dane np. liczby i je w nich przechowywać w celu ich późniejszego wykorzystania. Program komputerowy podczas swojego działania może zaglądać do tych pojemników sprawdzając i odczytując ich aktualną zawartość, a czasem zmieniając ich zawartość. Żeby program wiedział gdzie zajrzeć (można przecież wyobrazić, że w programie wystąpi więcej zmiennych), zmienne muszą mieć swoje nazwy.

Zmiennej możesz użyć np. przy określaniu ilości kroków Scottiego. Wyobraź sobie, że w jednym zadaniu Scottie porusza się dwukrotnie - dwa kroki do przodu i **OBRÓT W LEWO**.

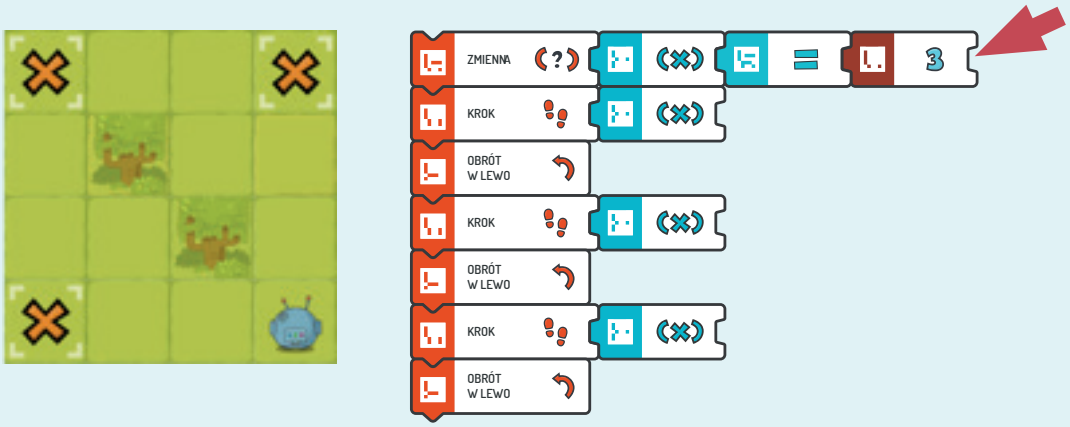
W kolejnym zadaniu (patrz rysunek poniżej) jego ruchy są prawie identyczne, jednak zmienia się tylko ilość kroków (Trzy kroki i **OBRÓT W LEWO**). W sumie obserwujemy, że musimy zmienić program w aż 3 miejscach, aby poprzedni program zadziałał do tej sytuacji.

WPROWADZENIE DO MODUŁU 2/3: Nowe pojęcie: ZMIENNE

Wróćmy do pierwszego zadania (patrz rysunek poniżej) i tym razem zróbmy to prościej. W tym zadaniu wykorzystamy zmienne. Przy każdym kroku użyjemy jego zmiennej zamiast ilości kroków. W pierwszej części programu zdefiniujemy tę zmienną znakiem (równa się) **= X** i przypiszemy jej wartość 2 czyli ilość kroków.



Aby rozwiązać kolejne zadanie wystarczy, że dokonasz zmiany w programie tylko w pierwszej linii kodu tam, gdzie przypisana jest wartość do zmiennej **X**.



Jak widać użycie zmiennych często znacząco skraca nam czas na poprawienie programu. Podsumowując:

Zmienne sprawiają, że program jest czytelny i łatwiej go zmienić.

WPROWADZENIE DO MODUŁU 3/3:
Nowe pojęcie: ZMIENNE

Aby zobaczyć jak działają zmienne musisz:

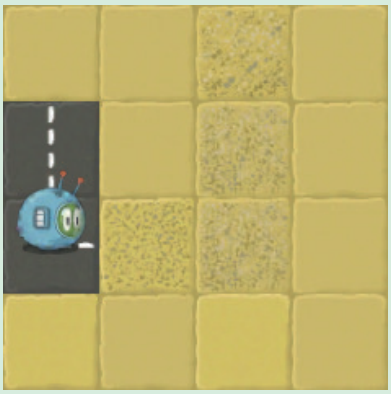
1. „utworzyć” zmienną nadając jej nazwę i określając jaką wartość będzie przechowywać,

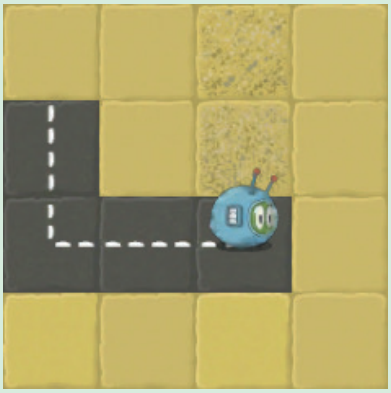
2. a następnie w programie użyć tej zmiennej wywołując tylko jej nazwę, zamiast konkretnej wartości.

WPROWADZENIE - asfaltowa droga

Scottie idąc po drodze z piasku w jej miejsce zbuduje drogę asfaltową.

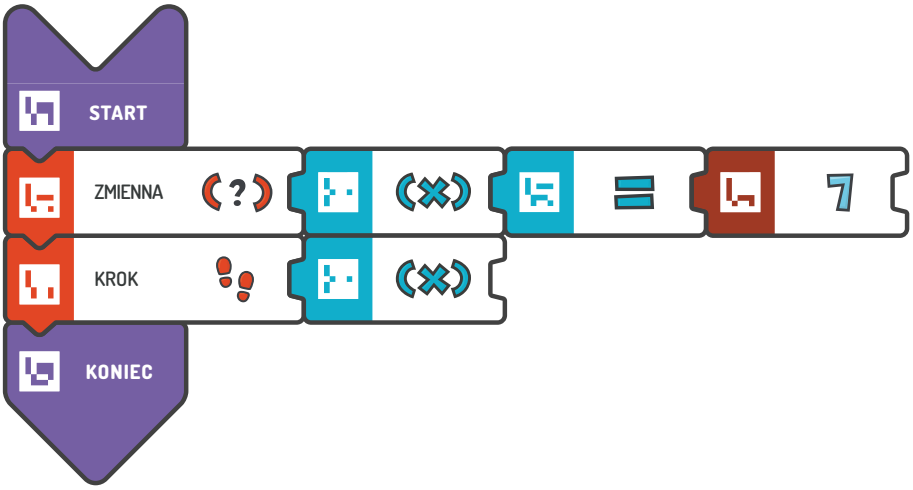
Pamiętaj, aby w zadaniu stworzyć zmienną **X =** ilość kroków, a następnie ją wywołać w programie.





Wyzwanie:

Przykładowe rozwiązanie:

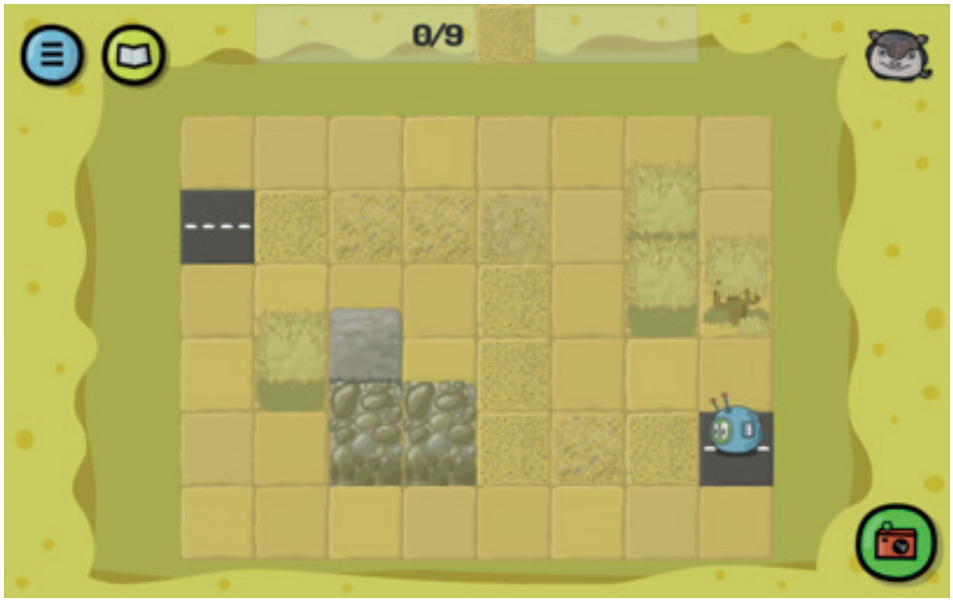


Wyzwanie:

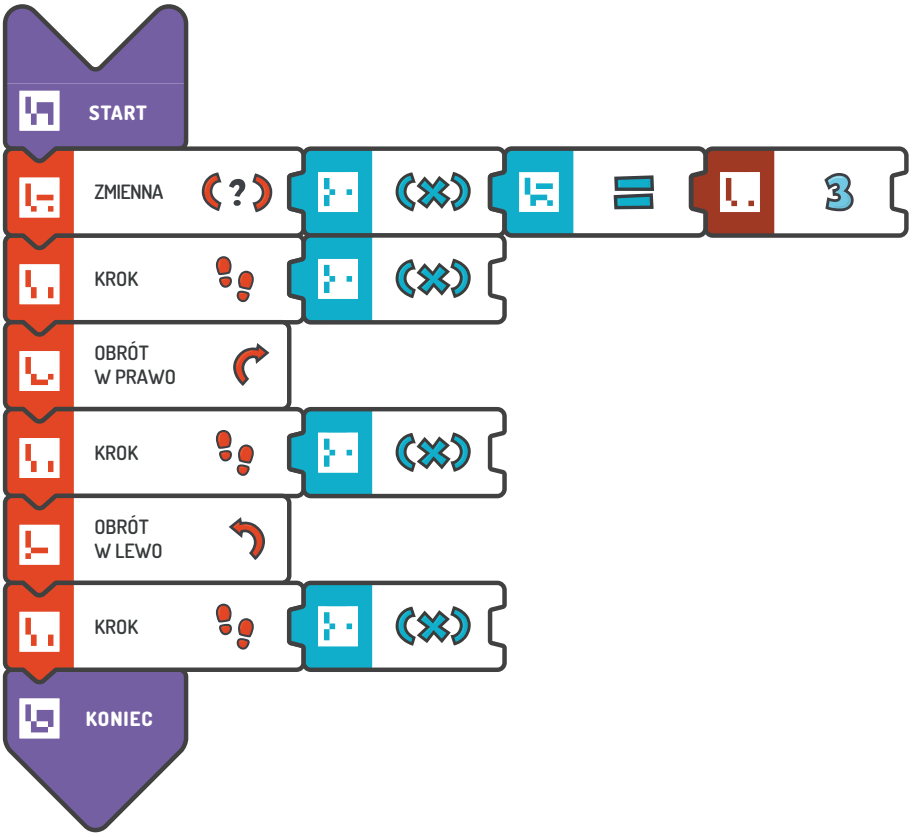
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=7
    KROK (X)
KONIEC
```

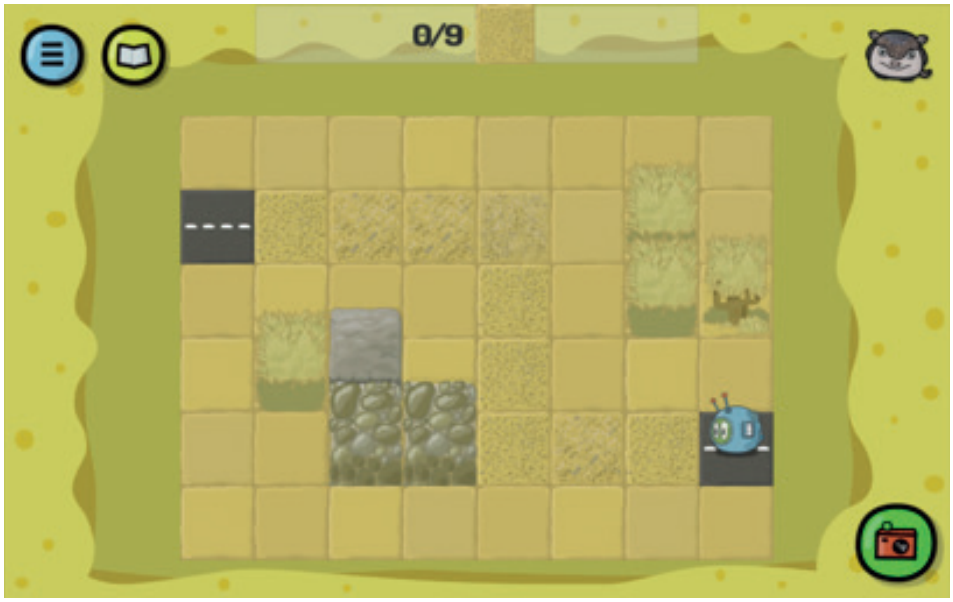
Wyzwanie:



Przykładowe rozwiązanie:



Wyzwanie:



Przykładowe rozwiązanie (format tekstowy):

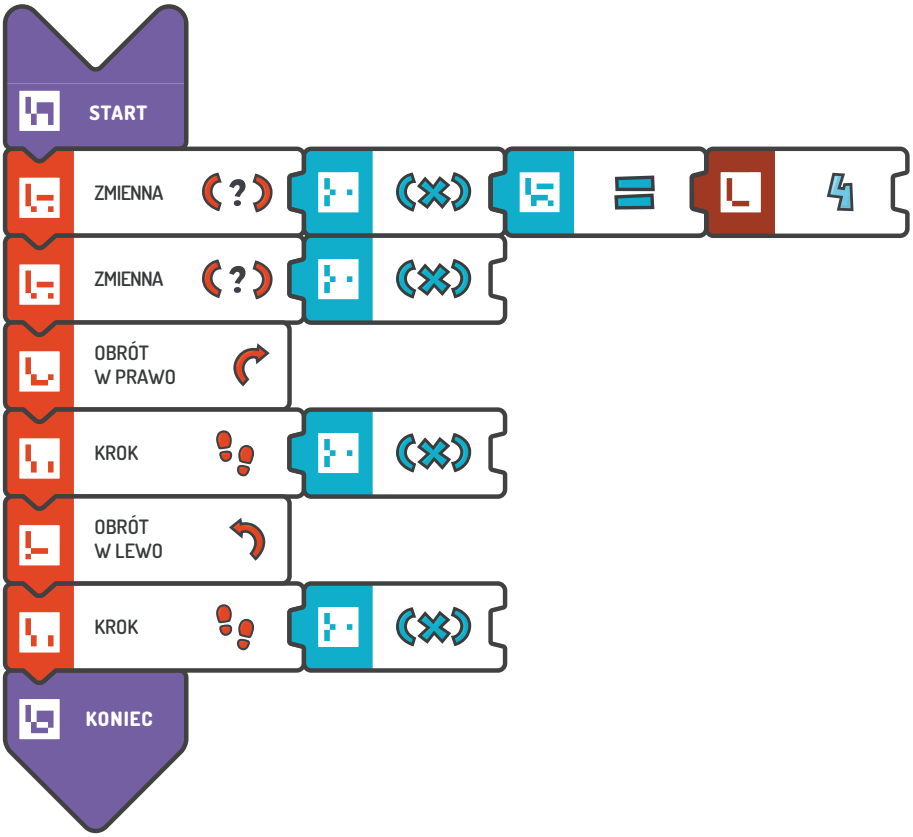
```
START
    ZMIENNA X=3
    KROK (X)
    OBRÓT W PRAWO
    KROK (X)
    OBRÓT W LEWO
    KROK (X)
KONIEC
```

UWAGA

Twój poprzedni program potrafi skutecznie rozwiązać także to zadanie. **Wystarczy zmienić wartość zmiennej X.**

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

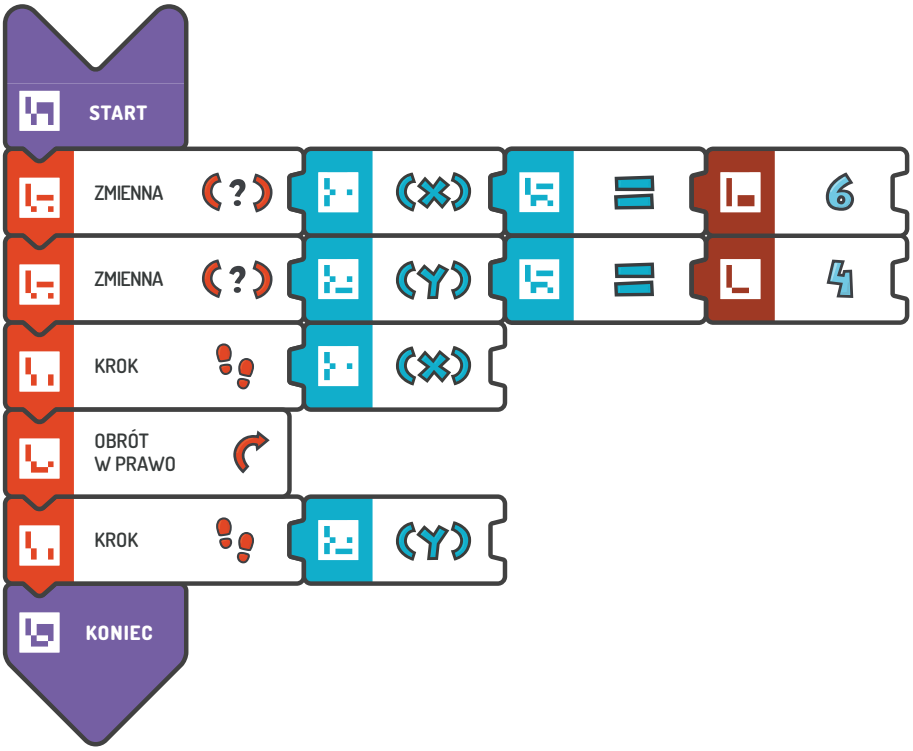
```
START
    ZMIENNA X=4
    KROK (X)
    OBRÓT W PRAWO
    KROK (X)
    OBRÓT W LEWO
    KROK (X)
KONIEC
```

WPROWADZENIE: X i Y

W grze Scottie GO! możemy stworzyć i wykorzystać jednocześnie dwie zmienne: **X** i **Y**.

Wyzwanie:

Przykładowe rozwiązanie:



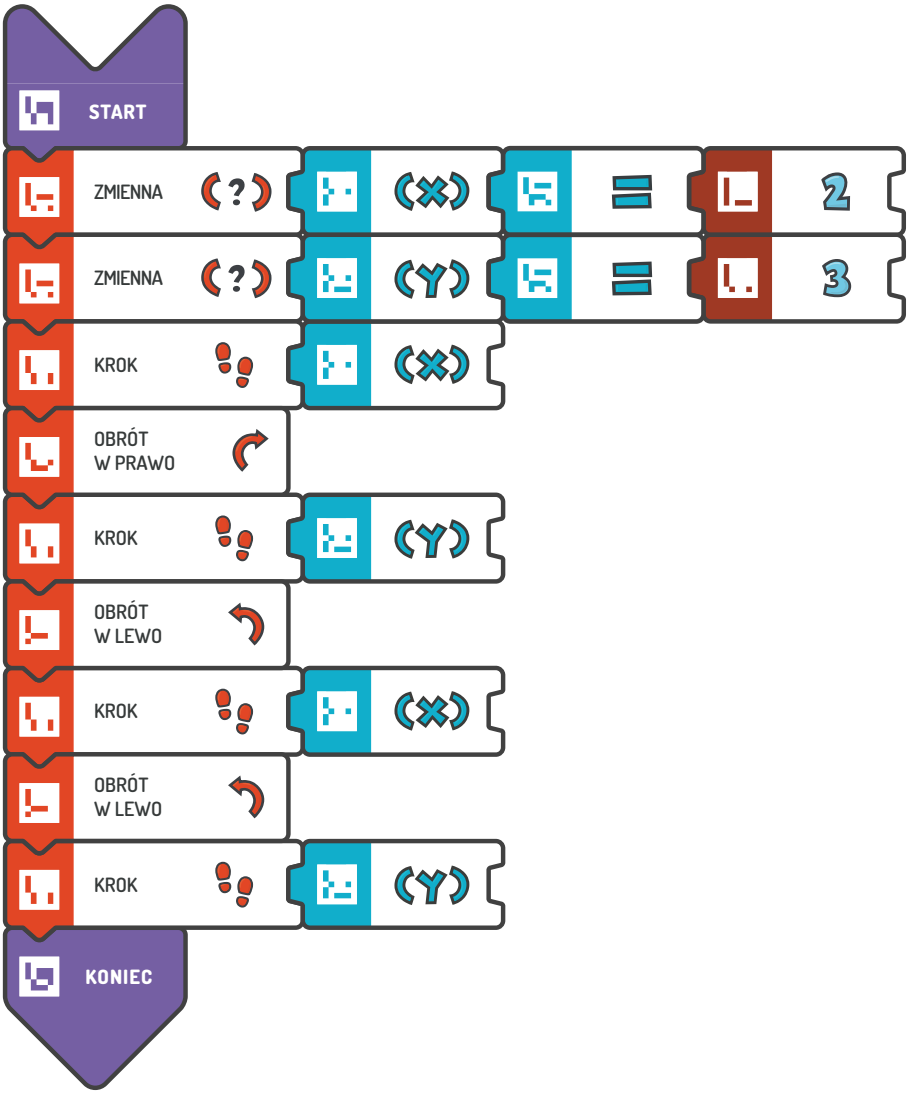
Wyzwanie:

Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=6
    ZMIENNA Y=4
    KROK (X)
    OBRÓT W PRAWO
    KROK (Y)
KONIEC
```

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

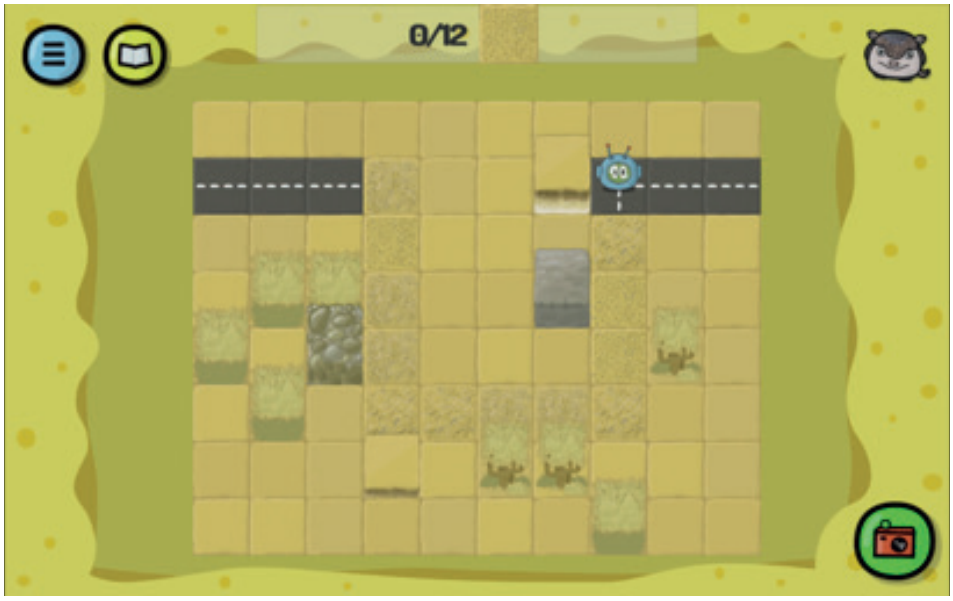
Przykładowe rozwiązanie (format tekstowy):

START
ZMIENNA X=2
ZMIENNA Y=3
KROK (X)
OBRÓT W PRAWO
KROK (Y)
OBRÓT W LEWO
KROK (X)
OBRÓT W LEWO
KROK (Y)
KONIEC

WPROWADZENIE:
Zmienne i pętle

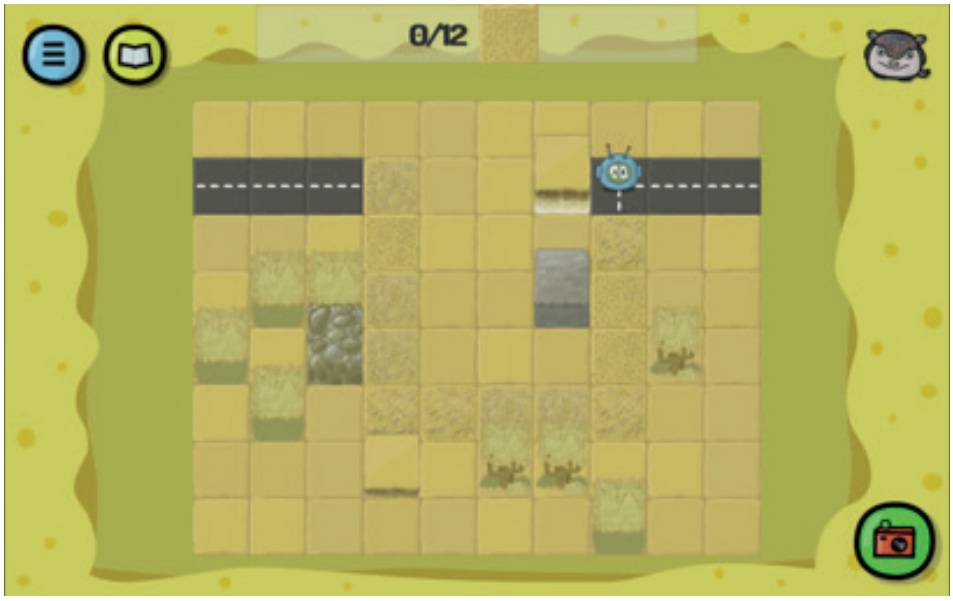
Wykorzystaj w zadaniu dwie zmienne **X** i **Y** wraz z pętlą **POWTÓRZ**. Użyj zmiennej **X**, aby przechowywać ilość powtórzeń pętli oraz zmienną **Y**, aby przechowywać ilość kroków.

Wyzwanie:



Przykładowe rozwiązanie:

Wyzwanie:



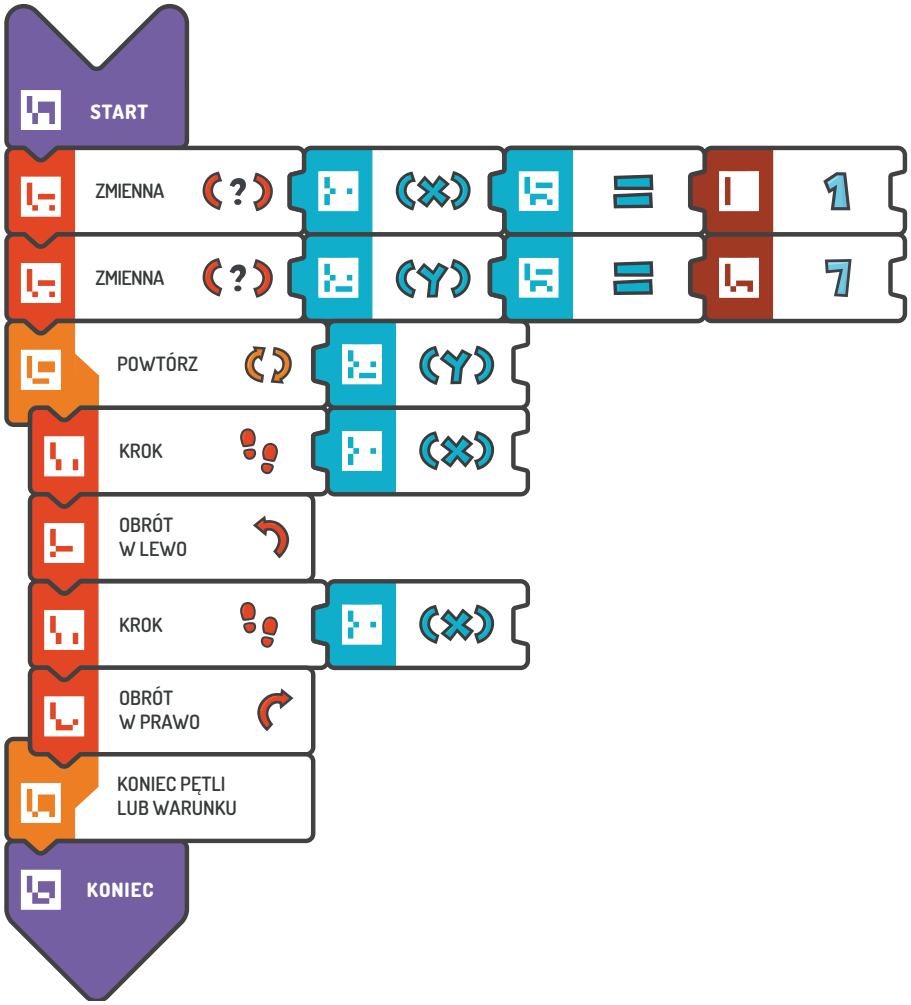
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=3
    ZMIENNA Y=4
    POWTÓRZ (X) {
        KROK (Y)
        OBRÓT W PRAWO
    }
KONIEC
```

Wyzwanie

(**Wykorzystaj w zadaniu dwie zmienne. Jedna z nich będzie miała tym razem za zdanie przechowywać ilość powtórzeń pętli**):

Przykładowe rozwiązanie:



Wyzwanie

(**Wykorzystaj w zadaniu dwie zmienne. Jedna z nich będzie miała tym razem za zdanie przechowywać ilość powtórzeń pętli)**:

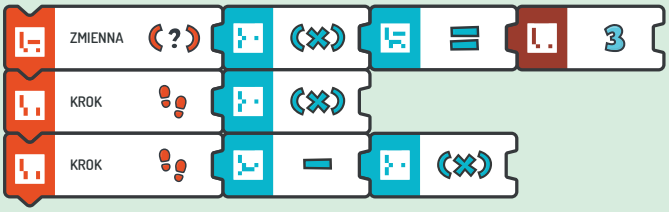
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=1
    ZMIENNA Y=7
    POWTÓRZ (Y) {
        KROK (X)
        OBRÓT W LEWO
        KROK (X)
        OBRÓT W PRAWO
    }
KONIEC
```

WPROWADZENIE:
Działania matematyczne na zmiennych

Na zmiennych możemy również wykonywać działania matematyczne.

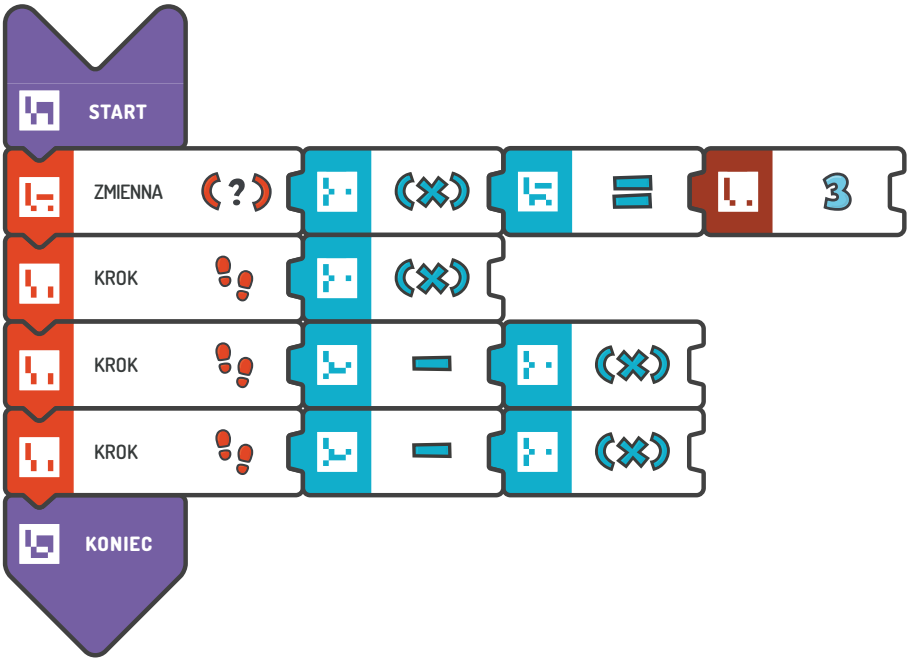
W poniższym fragmencie programua zmienimy znak na ilości kroków, które wykonać ma Scottie poprzez uieszczenie klocka **MINUS** przed zmienną **X**.



W tym zadaniu wykorzystaj koniecznie klocek z cyfrą 3 jeden raz.

Wyzwanie:

Przykładowe rozwiązanie:



Wyzwanie:

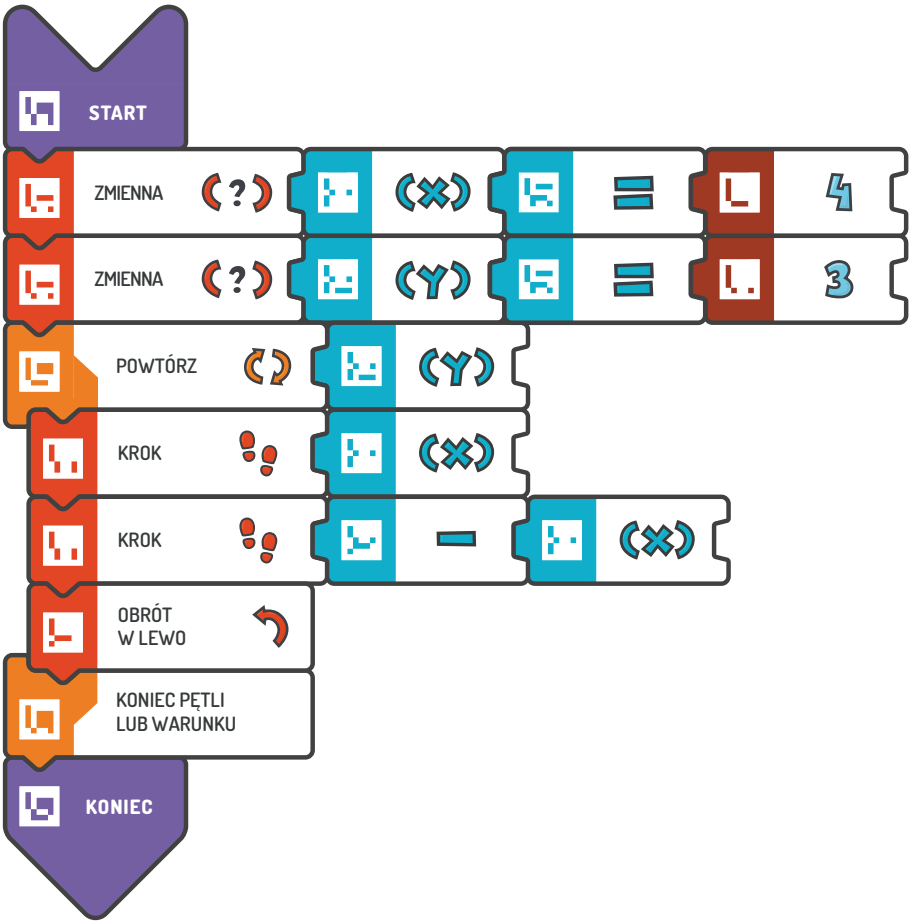
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=3
    KROK (X)
    KROK (-X)
    KROK (-X)
KONIEC
```

Wyzwanie

(**W tym zadaniu użyj dwóch zmiennych: jednej dla liczby kroków i drugiej dla ilości powtórzeń pętli)**:

Przykładowe rozwiązanie:



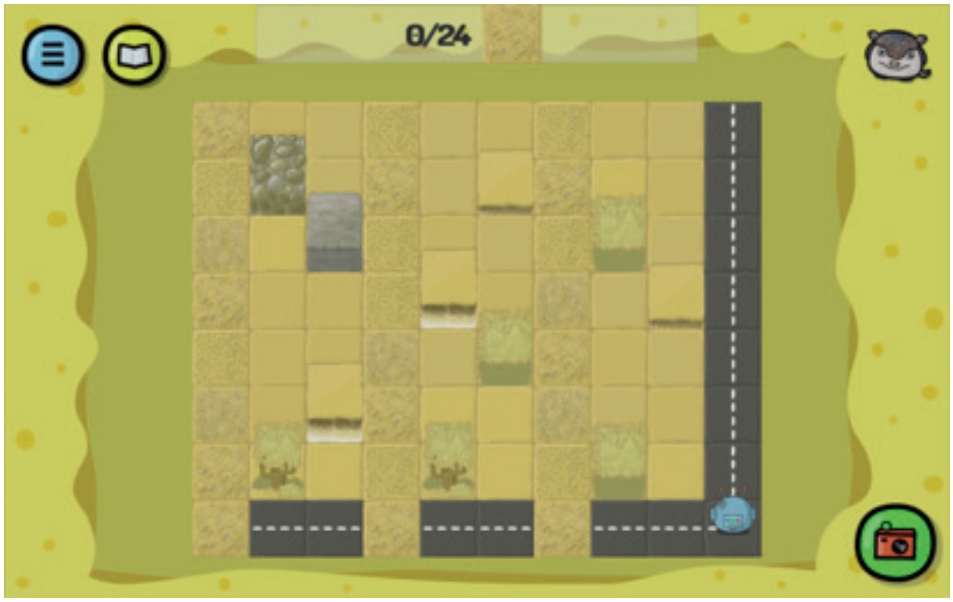
Wyzwanie:

(**W tym zadaniu użyj dwóch zmiennych: jednej dla liczby kroków i drugiej dla ilości powtórzeń pętli)**:

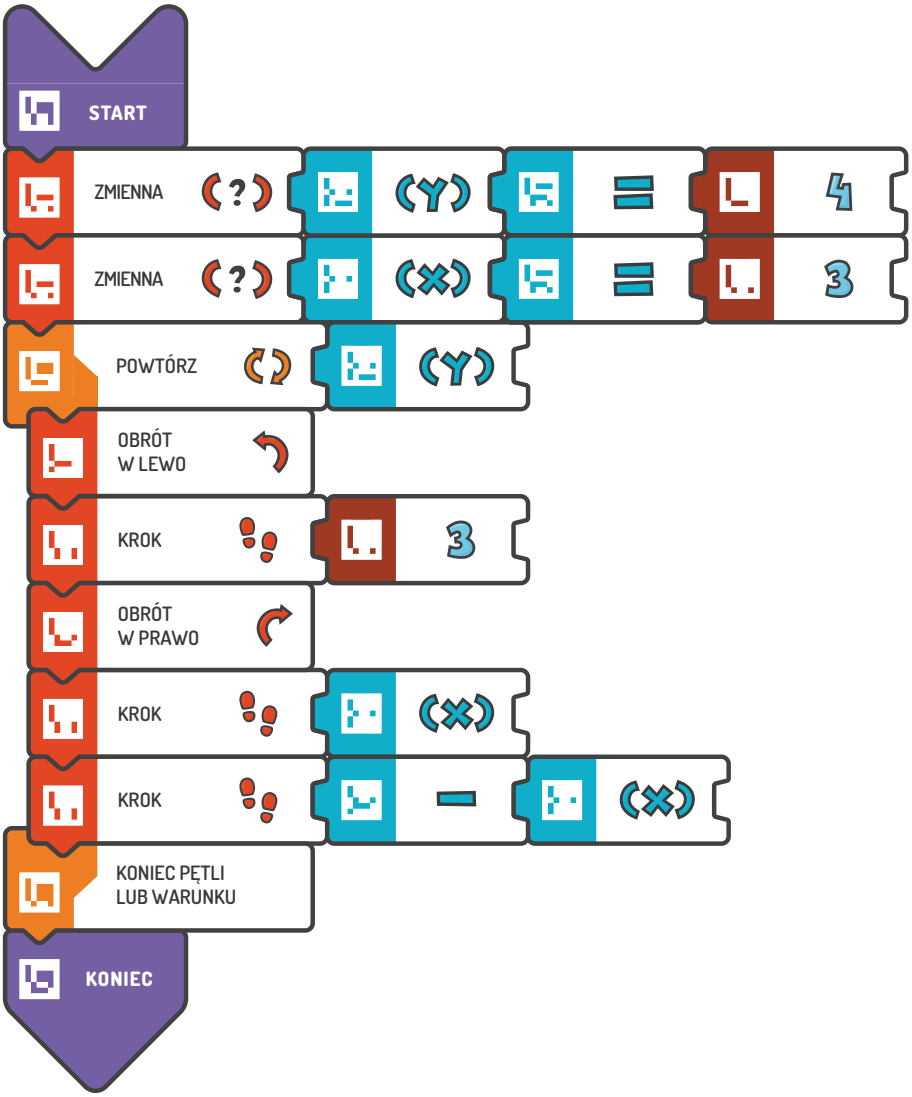
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=4
    ZMIENNA Y=3
    POWTÓRZ (Y) {
        KROK (X)
        KROK (-X)
        OBRÓT W LEWO
    }
KONIEC
```

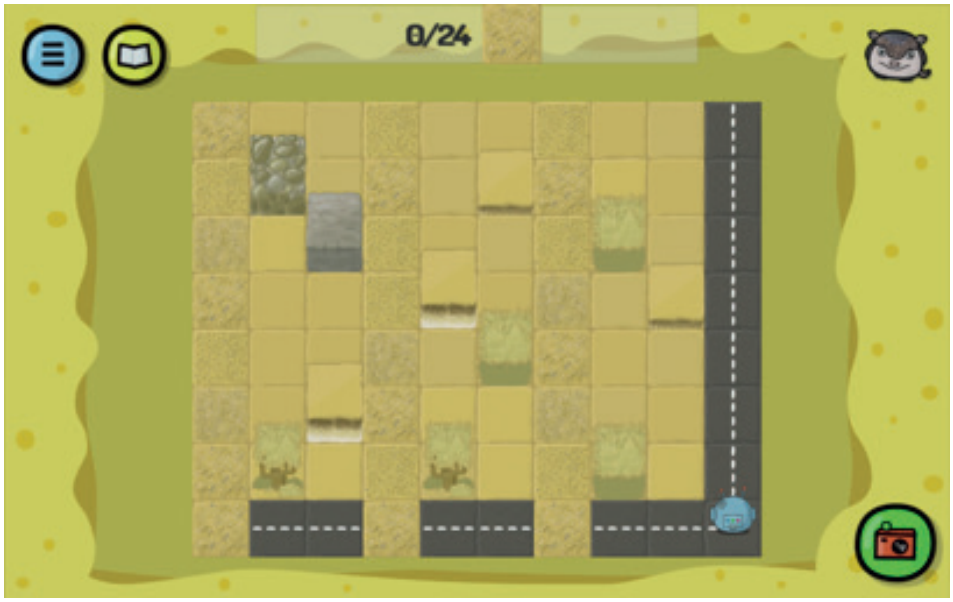
Wyzwanie:



Przykładowe rozwiązanie:



Wyzwanie:



Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA Y=3
    ZMIENNA X=7
    POWTÓRZ (Y) {
        OBRÓT W LEWO
        KROK (3)
        OBRÓT W PRAWO
        KROK (X)
        KROK (-X)
    }
KONIEC
```

Moduł VIII
Australia 2

Instrukcje warunkowe - JEŻELI

Zasadź brakujące drzewka w odpowiednich polach.

W tym module musisz użyć instrukcji warunkowej.

WPROWADZENIE DO MODUŁU 1/4: Instrukcje warunkowe

Codziennie każdy z nas podejmuje jakieś decyzje. Czy założyć kurtkę do pracy lub szkoły? Ile zjem kanapek na śniadanie? Czy do szkoły lub pracy pojechać samochodem, metrem lub autobusem? Programy komputerowe wciąż coś sprawdzają, przeliczają i podejmują decyzje. Pisząc program i wprowadzając takie instrukcje musisz wszystko dokładnie komputerowi wytłumaczyć, aby wiedział co w jakiej sytuacji zrobić.

Instrukcja JEŻELI, W PRZECIWNYM RAZIE

To podstawowa instrukcja warunkowa związana z podejmowaniem decyzji. Instrukcja warunkowa jest potężnym narzędziem i w związku z powyższym występuje w prawdopodobnie większości programów.

Kluczowe elementy instrukcji warunkowej to:

1. Warunek, który program sprawdza,

WPROWADZENIE DO MODUŁU 2/4: Instrukcje warunkowe

2. Instrukcje, które zostaną wykonane, gdy spełniony zostanie ten warunek,

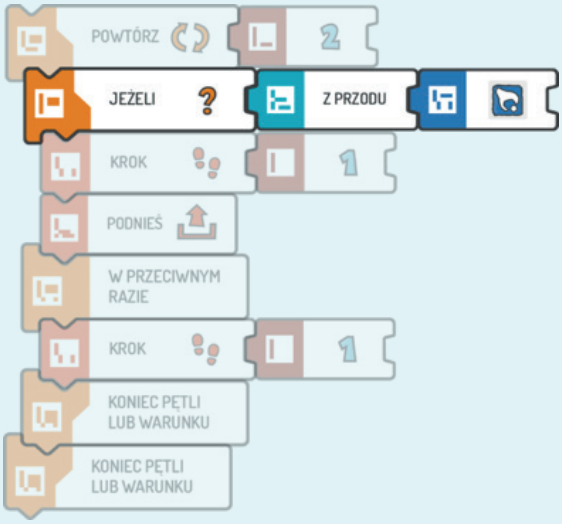
3. Instrukcje, które zostaną wykonane, gdy warunek nie będzie spełniony.

Zobaczymy to na przykładzie. Dla lepszego zrozumienia tego zagadnienia instrukcję warunkową wywołamy dwukrotnie **poprzez umieszczenie instrukcji warunkowej wewnątrz pętli POWTÓRZ (2)**

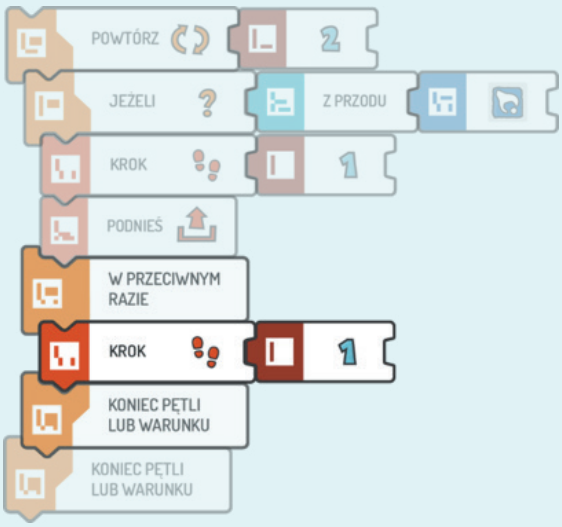
WPROWADZENIE DO MODUŁU 3/4: Instrukcje warunkowe

Przykład zastosowania instrukcji warunkowej.

1. Scottie sprawdza pierwszy raz warunek: **„Czy z przodu jest OBIEKT DO ZEBRANIA?”** Warunek nie jest spełniony.



2. Gdy warunek nie jest spełniony wywoływana jest instrukcja **W PRZECIWNYM RAZIE**. Scottie porusza się do przodu o jedno pole - **KROK (1)**.



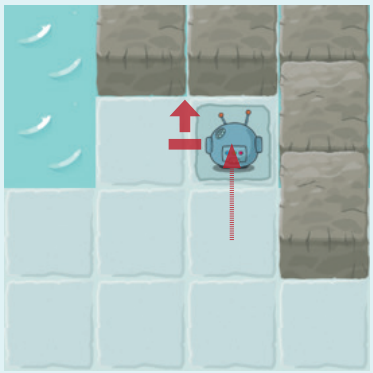
...

WPROWADZENIE DO MODUŁU 4/4: Instrukcje warunkowe

Przykład zastosowania instrukcji warunkowej c.d.

3. Następnie drugi raz - zastosowaliśmy pętlę **POWTÓRZ (2)** - sprawdzany jest warunek: **„Czy z przodu OBIEKT DO ZEBRANIA?”**. Warunek jest spełniony.

4. Gdy warunek jest spełniony wywoływana jest instrukcja warunkowa **JEŻELI**. Scottie wykonuje instrukcje **KROK (1)** i **PODNIEŚ**.

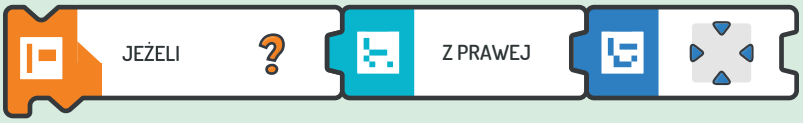


Gdy w instrukcji warunkowej jeden z warunków zostanie spełniony, drugi warunek nie jest już sprawdzany, ani wykonywany.

WPROWADZENIE:
Pole akcji

Jeden z naszych zwierzęcych bohaterów ma wspaniały ogród. W zamian za pomoc Scottie zobowiązał się zasadzić w nim kilka roślin.

Zasadź brakujące drzewka w odpowiednich polach wykorzystując do tego nowe klocki, w szczególności pętlę **JEŻELI** w połączeniu z klockiem **POLE AKCJI.** Położenie pola akcji w zadaniu względem Scottiego będzie warunkiem, który będzie sprawdzany - np:

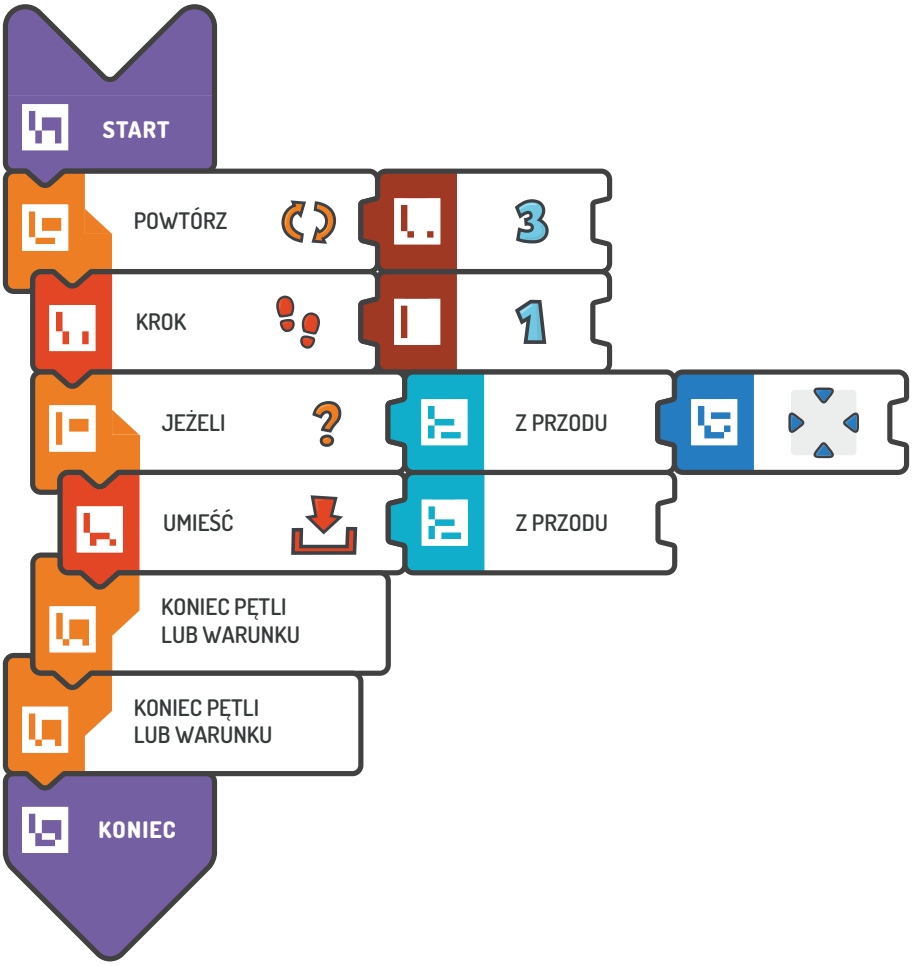


PAMIĘTAJ!

Na końcu warunku umieszczamy klocek **KONIEC PĘTLI LUB WARUNKU**.

Wyzwanie **(Pamiętaj! W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



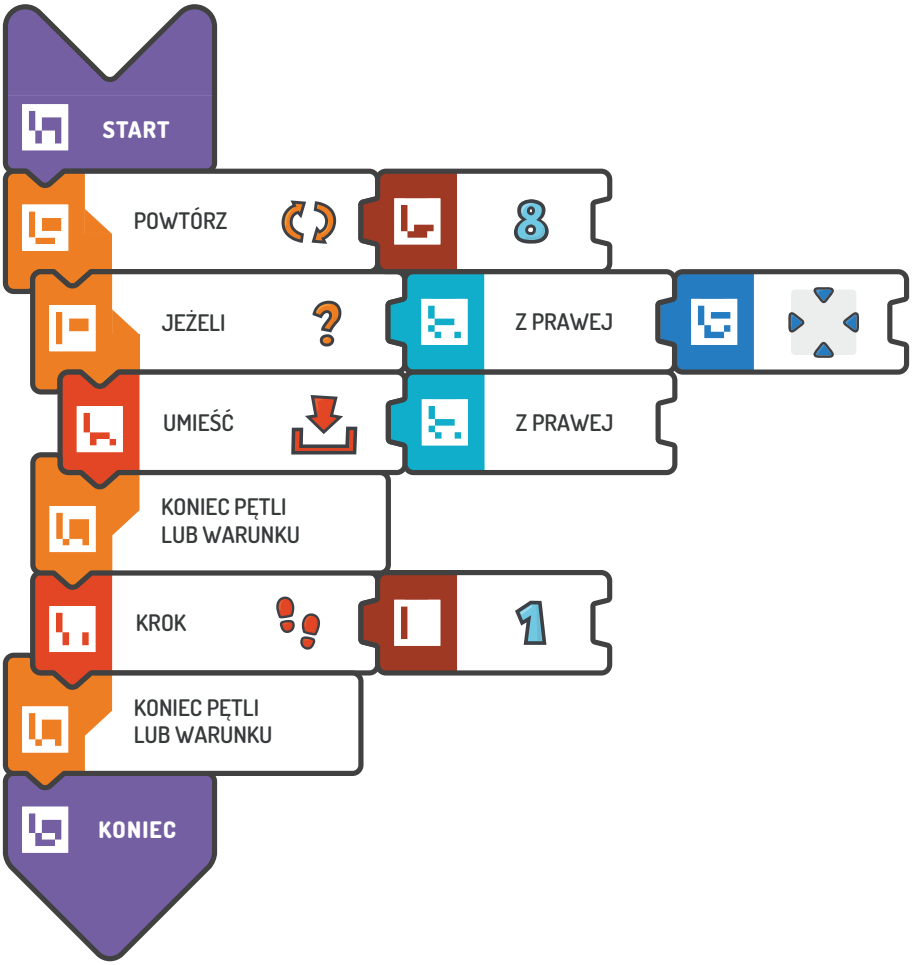
Wyzwanie **(Pamiętaj! W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (3) {
        KROK (1)
        JEŻELI (Z PRZODU < POLE AKCJI) {
            UMIEŚĆ (Z PRZODU)
        }
    }
KONIEC
```

Wyzwanie **(Pamiętaj! W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



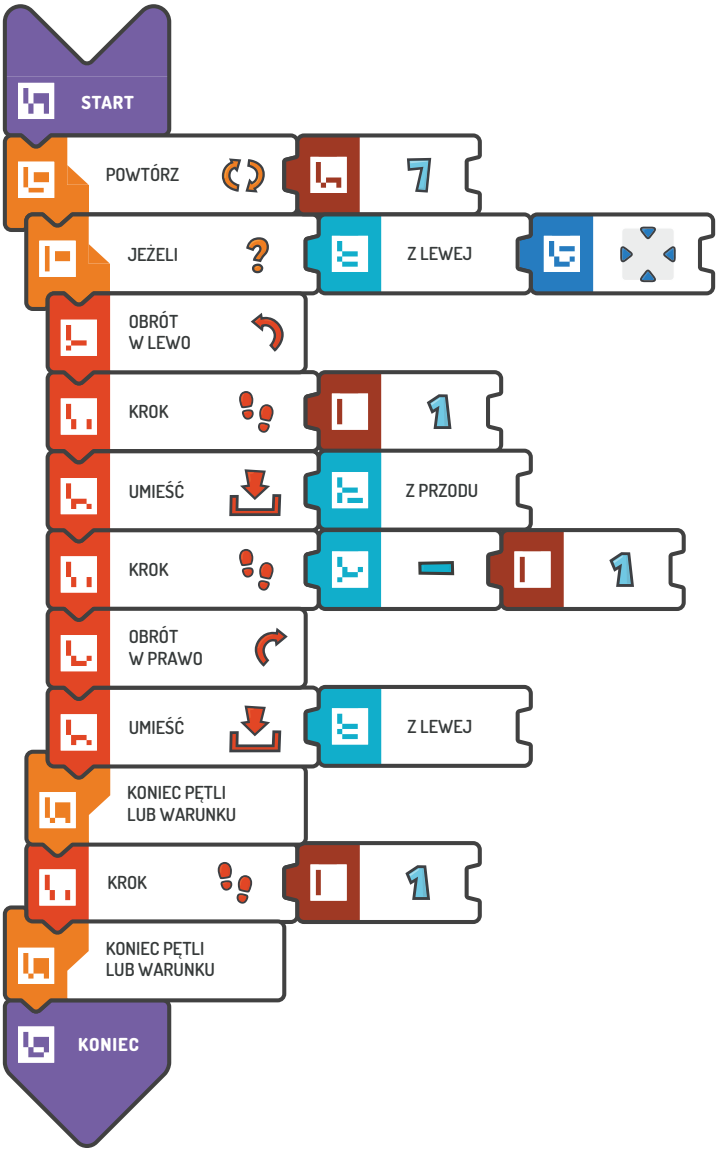
Wyzwanie **(Pamiętaj! W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (8) {
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
        KROK (1)
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



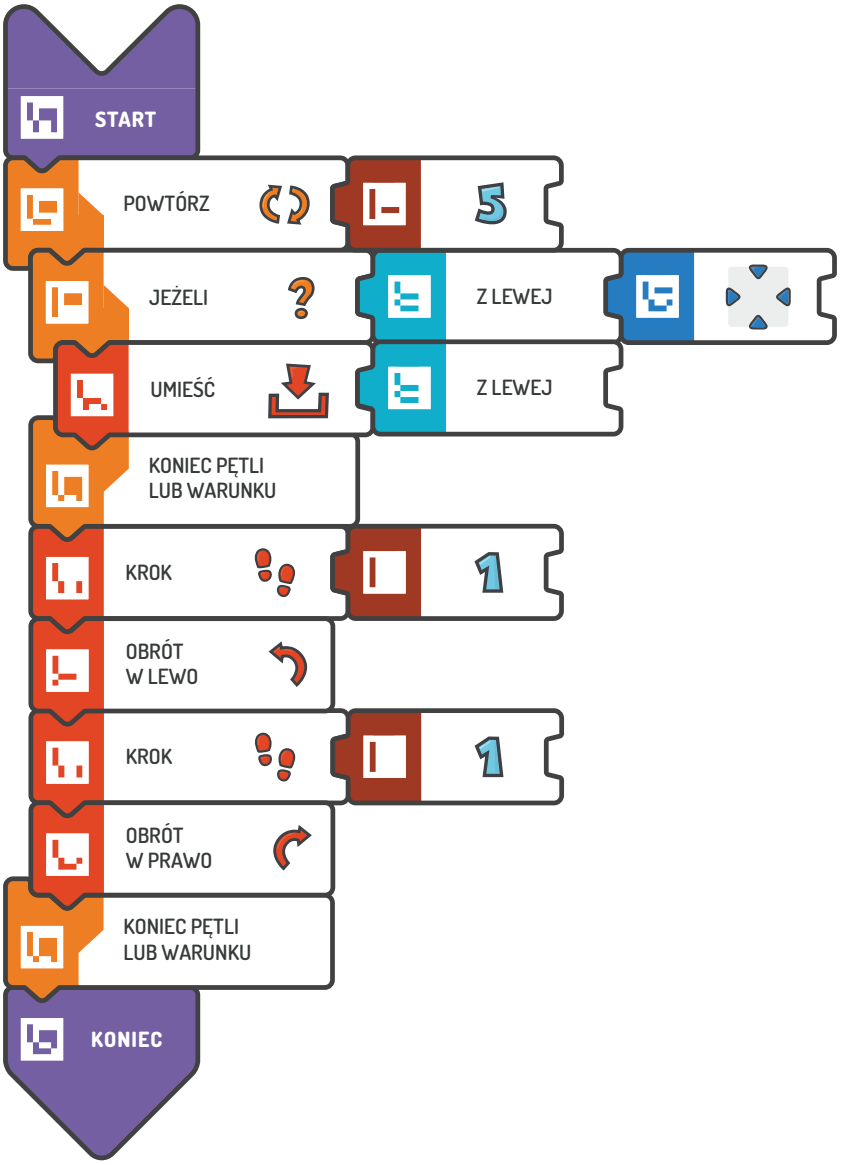
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (7) {
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            OBRÓT W LEWO
            KROK (1)
            UMIEŚĆ (Z PRZODU)
            KROK (-1)
            OBRÓT W PRAWO
            UMIEŚĆ (Z LEWEJ)
        }
        KROK (1)
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (5) {
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            UMIEŚĆ (Z LEWEJ)
        }
        KROK (1)
        OBRÓT W LEWO
        KROK (1)
        OBRÓT W PRAWO
    }
KONIEC
```

WSKAZÓWKA

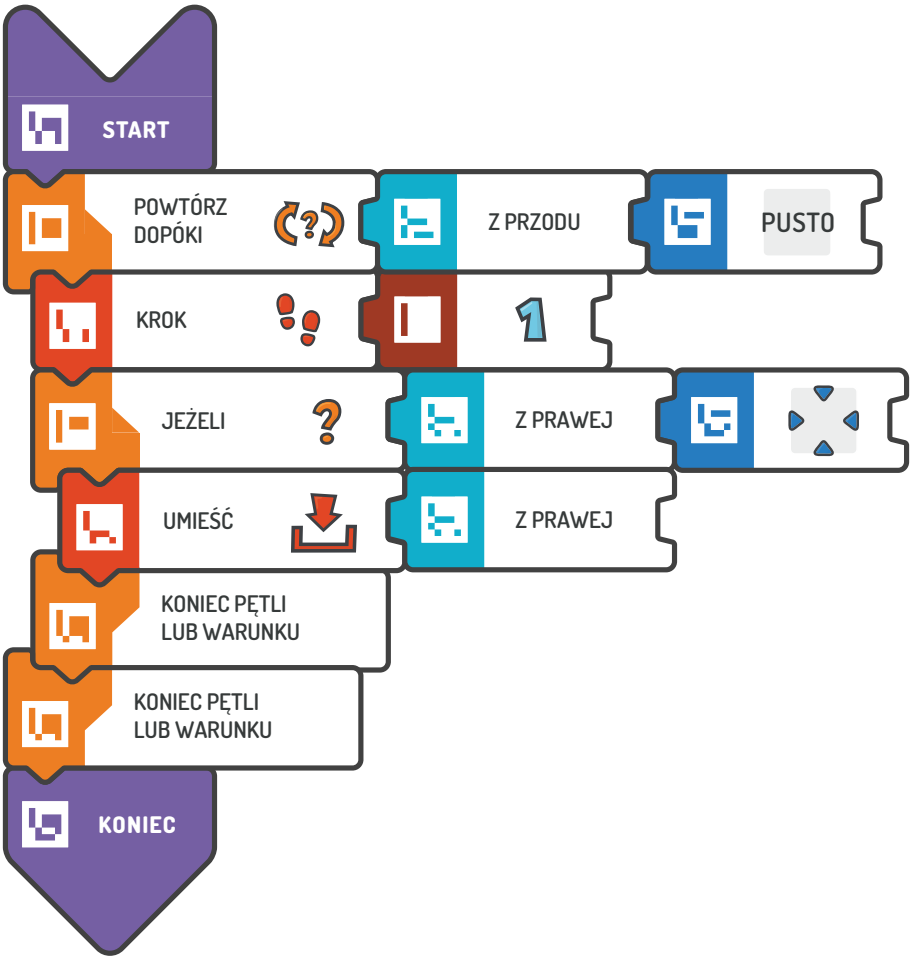
W tym zadaniu połączysz pętlę warunkową z rozszerzeniem **JEŻELI**.

Na horyzoncie pojawiła się przeszkoda, więc możesz skorzystać z klocka **POWTÓRZ DOPÓKI** zamiast zwykłej pętli **POWTÓRZ**.

Wyzwanie

(**Zobacz, przeszkoda na horyzoncie. Skorzystaj z < POWTÓRZ DOPÓKI> zamiast zwykłej pętli < POWTÓRZ>)**:

Przykładowe rozwiązanie:



Wyzwanie

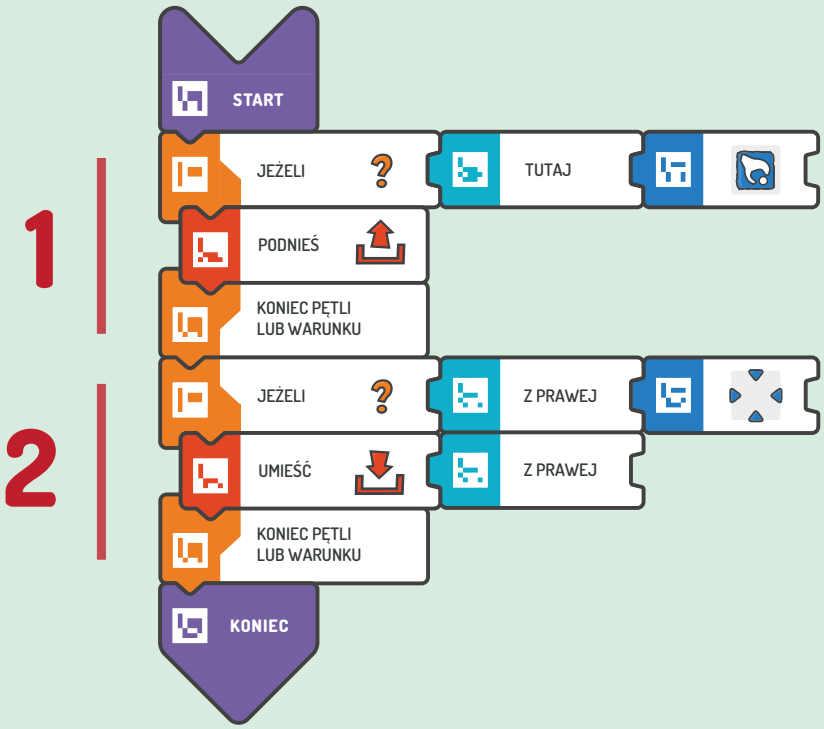
(**Zobacz, przeszkoda na horyzoncie. Skorzystaj z < POWTÓRZ DOPÓKI> zamiast zwykłej pętli < POWTÓRZ>)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
    }
KONIEC
```

WPROWADZENIE:
używanie wielu instrukcji warunkowych

W programie można umieszczać wiele instrukcji warunkowych. To znakomite ćwiczenie logiczne i możliwość utrwalenia składni typowej dla wszystkich języków programowania. Każda instrukcja warunkowa może sprawdzać inny warunek, np:



W powyższym programie Scottie:

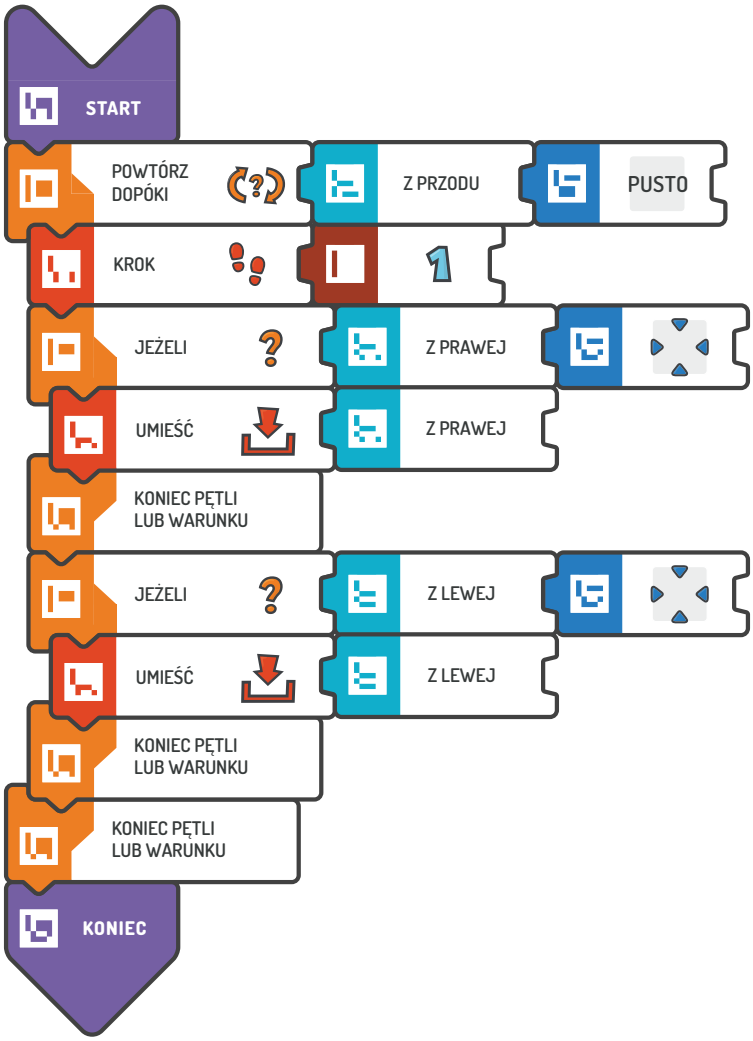
1. **Najpierw** sprawdzi czy na polu, na którym stoi, leży przedmiot do zebrania
   - **jeżeli tak**, Scottie podniesie przedmiot do zebrania (i przejdzie do następnego warunku)
   - **jeżeli nie**, Scottie nie wykona instrukcji wewnątrz tego warunku (i przejdzie do następnego warunku)

2. **Potem** sprawdzi czy na polu po jego prawej stronie jest pole akcji
   - **jeżeli tak**, Scottie umieści po prawej przedmiot w przyborniku (i przejdzie do następnego warunku)
   - **jeżeli nie**, Scottie nie wykona instrukcji wewnątrz tego warunku (i przejdzie do następnego warunku)

Wyzwanie

(W programie można umieszczać wiele instrukcji warunkowych. Każda z nich może sprawdzać inny warunek):

Przykładowe rozwiązanie:



Wyzwanie

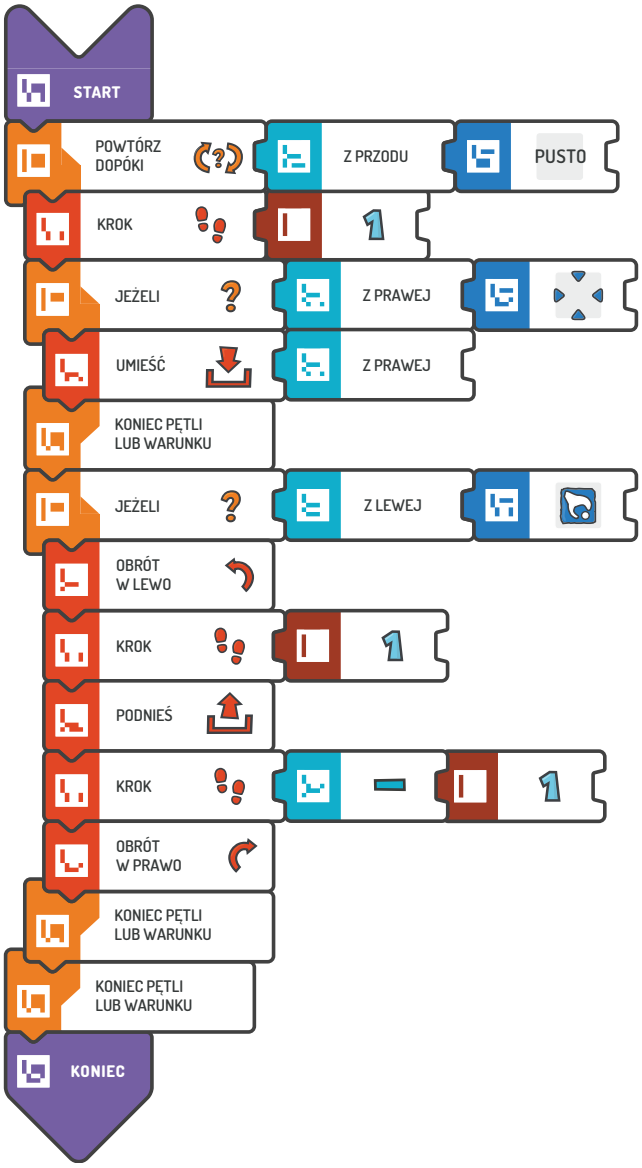
(W programie można umieszczać wiele instrukcji warunkowych. Każda z nich może sprawdzać inny warunek):

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            UMIEŚC (Z LEWEJ)
        }
    }
KONIEC
```

Wyzwanie **(Teraz nie masz już roślinek w zasobniku. Podnieś sadzonkę przed jej posadzeniem)**:

Przykładowe rozwiązanie:



Wyzwanie **(Teraz nie masz już roślinek w zasobniku. Podnieś sadzonkę przed jej posadzeniem)**:

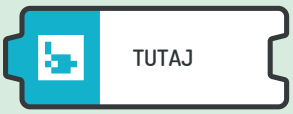
Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
        JEŻELI (Z LEWEJ < OBIEKT DO ZEBRANIA) {
            OBRÓT W LEWO
            KROK (1)
            PODNIEŚ
            KROK (-1)
            OBRÓT W PRAWO
        }
    }
KONIEC
```

WPROWADZENIE:
Klocki WZNIESIENIE i TUTAJ

Druga część ogrodu naszego zwierzęcego przyjaciela to skalisty obszar Australii. Pokaż Scottiemu jak omijać wzniesienia. Użyj klocków **WZNIESIENIE**

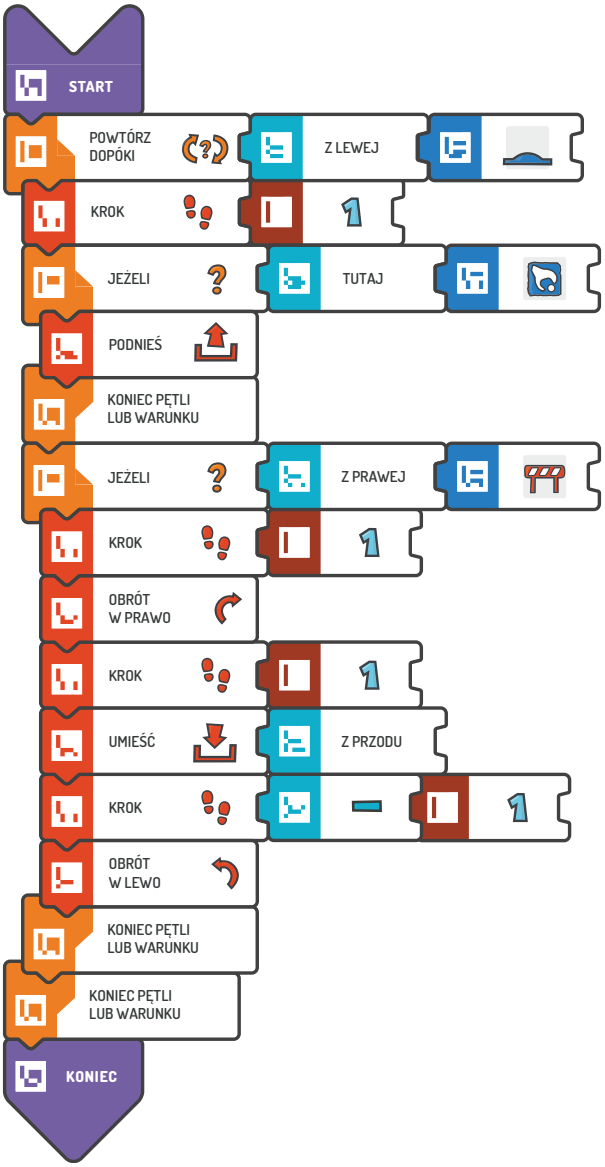
i **TUTAJ**.



Połącz je z poznanymi wcześniej warunkami.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ DOPÓKI (Z LEWEJ < WZNIESIENIE) {
        KROK (1)
        JEŻELI (TUTAJ < OBIEKT DO PODNIESIENIA) {
            PODNIEŚ
        }
        JEŻELI (Z PRAWEJ < PRZESZKODA) {
            KROK (1)
            OBRÓT W PRAWO
            KROK (1)
            UMIEŚĆ (Z PRZODU)
            KROK (-1)
            OBRÓT W LEWO
        }
    }
KONIEC
```

Moduł IX
Ameryka Północna 2

Pętla i warunki JEŻELI,
W PRZECIWNYM RAZIE JEŻELI, W PRZECIWNYM RAZIE

Remont statku Scottiego ma się ku końcowi!
Nasi pracownicy zostawili jednak spory bałagan
w warsztacie. Pomóż Scottiemu pozbierać
śruby i narzędzia z podłogi.

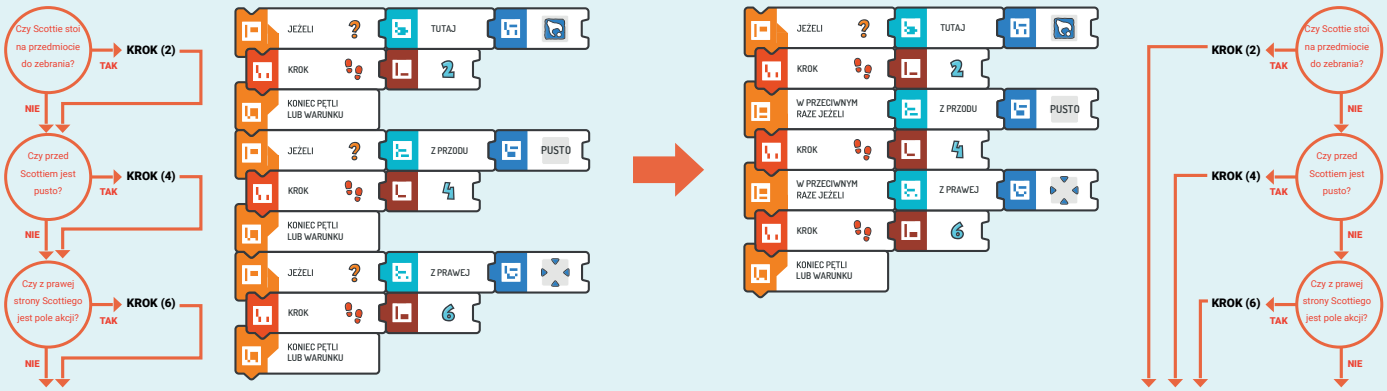
WPROWADZENIE DO MODUŁU (A): W PRZECIWNYM RAZIE JEŻELI 1/3

Widzieliśmy w **Module VIII**, że często będziemy musieli użyć wielu instrukcji warunkowych obok siebie. Pojawiają sie w związku z tym nowe wyzwania.

Jak sprawdzić wiele warunków w kolejności (dla przykładu użyjemy 3 warunków)?

W tym celu w programowaniu używa się instrukcji warunkowej **W PRZECIWNYM RAZIE JEŻELI** (w wielu językach programowania z angielskiego „**ELSE IF**").

Używając klocka **W PRZECIWNYM RAZIE JEŻELI** upraszczamy program, **ale też rozszerzamy wyrażenie warunkowe, miast dodawać nowego.**

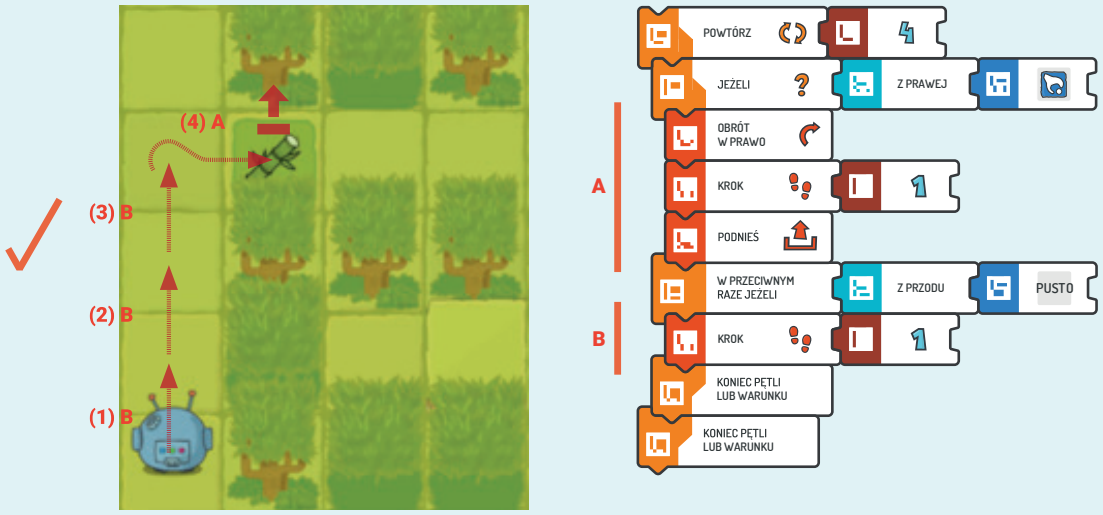


Kiedy używamy **W PRZECIWNYM RAZIE JEŻELI, kolejność sprawdzanych warunków jest kluczowa.** jeżeli spełniony będzie 1szy warunek, **2gi i 3ci nie będą spełnione.** Jeżeli spełniony będzie 2gi warunek**, 3ci już nie będzie spełniony**.

WPROWADZENIE DO MODUŁU (A): W PRZECIWNYM RAZIE JEŻELI 2/3

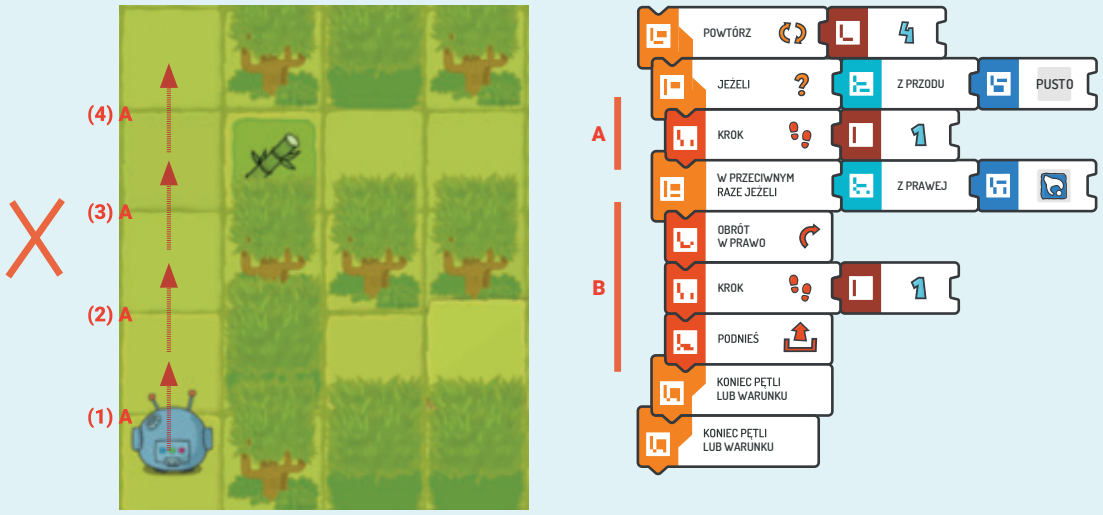
Przykład I (z użyciem instrukcji **W PRZECIWNYM RAZIE JEŻELI**): **Scottie ma podnieść bambus:**

Poprawne rozwiązanie.



Różnica pomiędzy poprawnym i złym rozwiązaniem **leży jedynie w kolejności sprawdzanych warunków** (i związanych z nimi instrukcjami dla Scottiego). Zwróćmy uwagę, że **każde powtórzenie pętli to realizacja instrukcji związanych z tylko jednym warunkiem**

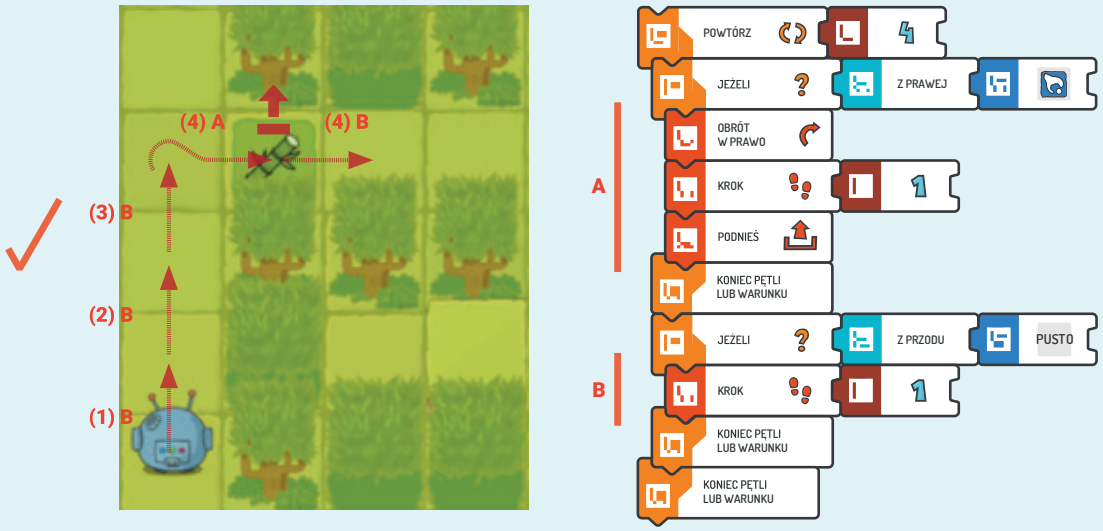
Złe rozwiązanie.



WPROWADZENIE DO MODUŁU (A): W PRZECIWNYM RAZIE JEŻELI 3/3

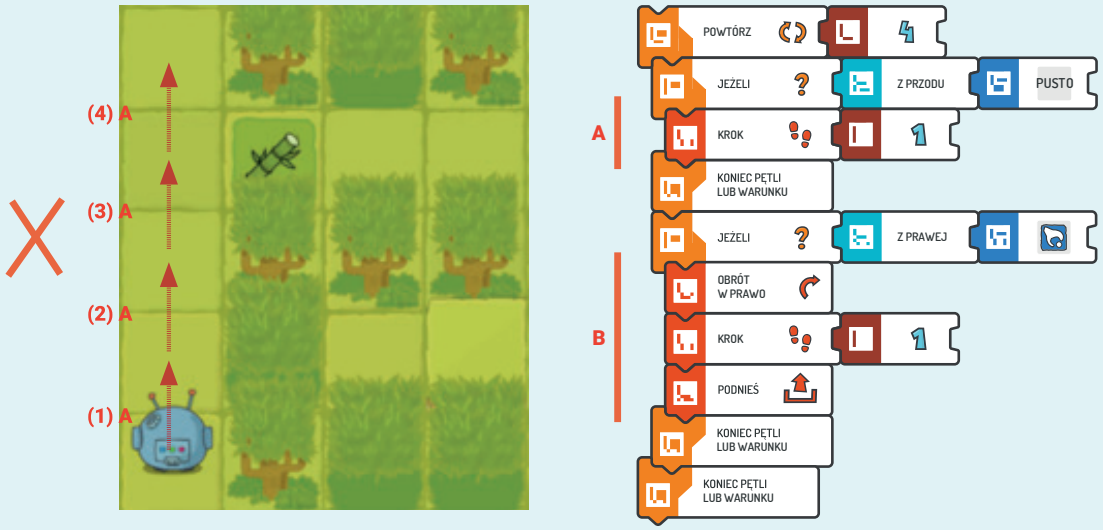
Przykład I (z użyciem 2ch pętli **JEŻELI**). **Scottie ma podnieść bambus**

Poprawne rozwiązanie.



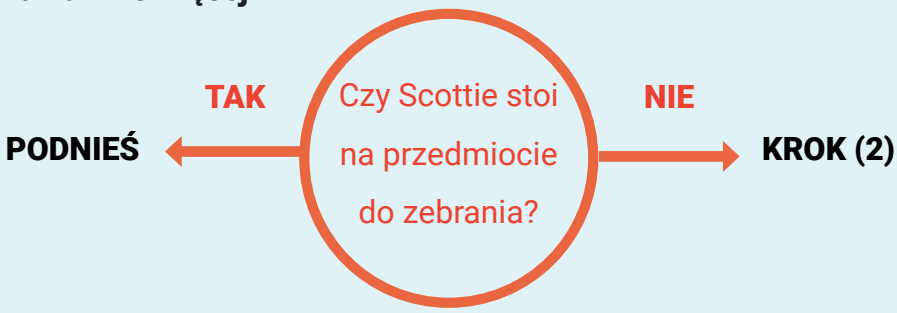
Poprawne rozwiązanie **bez użycia W PRZECIWYM RAZIE JEŻELI** jest o **1ną linijkę dłuższe . Zwróćmy uwagę, że przy czwartym powtórzeniu pętli wykonane spełnione są obydwa warunki.** Oczywyście klejność sprawdzanych warunków pozostaje kluczowa.

Złe rozwiązanie.



WPROWADZENIE DO MODUŁU (B): instrukcja W PRZECIWNYM RAZIE 1/3

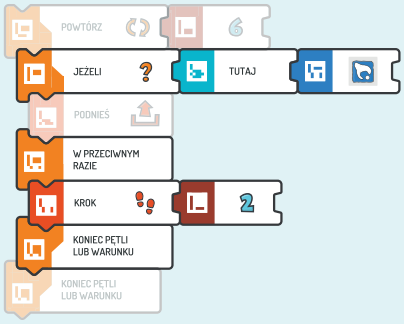
Oprócz konieczności pracy z wieloma warunkami czasami będziemy potrzebowali aby Scottie zrobił coś tylko **jeżeli poprzednie warunki nie będą spełnione** - na przykład **jak zaprogramować 2 alternatywne działania i nic więcej?**



Aby stworzyć tą konstrukcję w programowaniu używa się instrukcji warunkowej **W PRZECIWNYM RAZIE** (w większości języków programowania z angielskiego „**ELSE**").

To skuteczny sposób aby opisać „domyślne" instrukcje Scottiego - tzn. instrukcje które wykona zawsze, jak długo nie spełnimy warunku, który Jeżeli instrukcje nie są zawarte w instrukcji warunkowej, zostaną zawsze wykonane. Sprawdza się to szczególnie w pętli.

Z każdym powtórzeniem pętli Scottie ZAWSZE zrobi 2 kroki



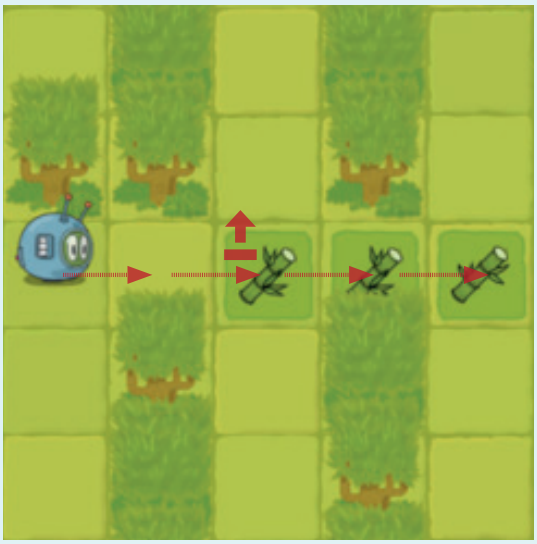
Z każdym powtórzeniem Scottie zrobi 2 kroki TYLKO jeżeli nie stoi na przedmiocie do zebrania

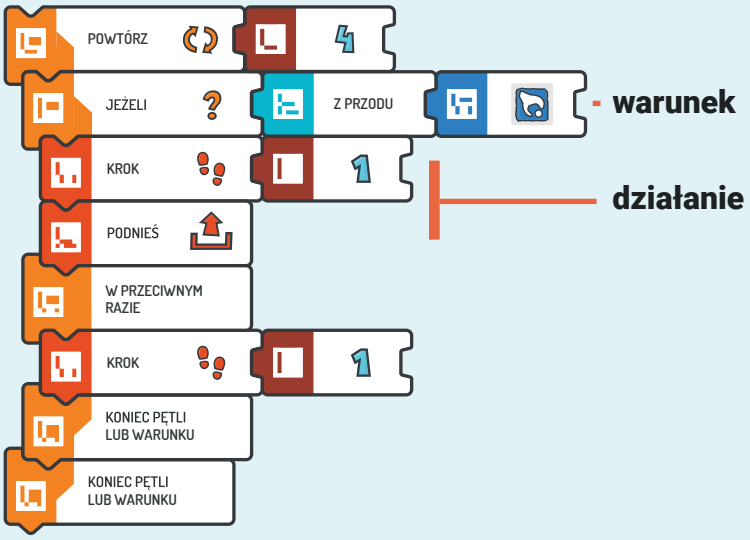
W typowym języku - **nie robimy tego w Scottie Go!** - programowania będziemy mogli użyć również słowa **NIE** przy tworzeniu podobnych instrukcji warunkowych.

WPROWADZENIE DO MODUŁU (B): instrukcja W PRZECIWNYM RAZIE 2/3

Przykłady użycia instukcji **W PRZECIWNYM RAZIE:**

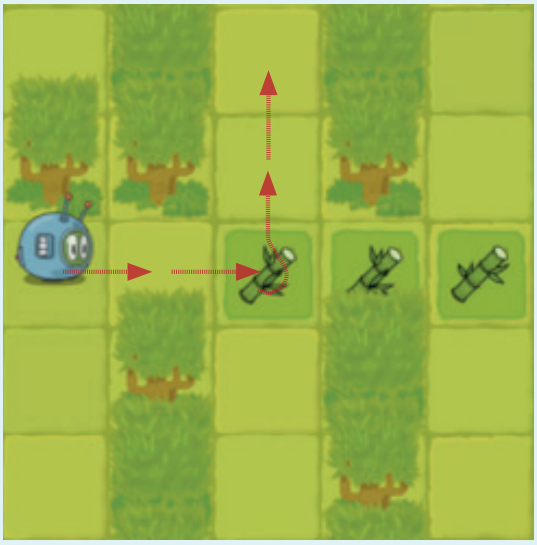
Chcemy napiszemy program w którym Scottie idzie przed siebie dopóki nie wydarzy się coś co wymaga od niego reakcji - np. **Scottie będzie sprawdzał czy nie ma przed nim przedmiotu do zebrania** lub **będzie chciał skręcić jeżeli z lewej będzie pusto .** Program będzie miał kostrukcję którą łatwo będzie zmodyfikować i dostosować do naszych (nowych) potrzeb.

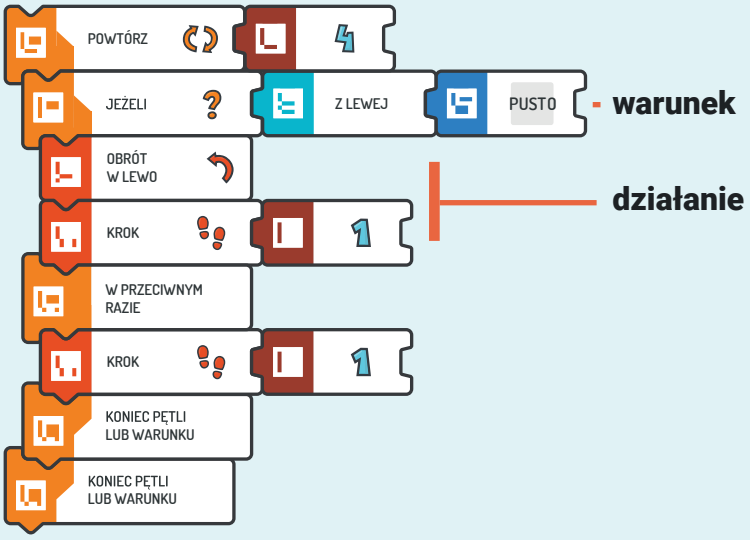




Nie musimy określać warunku koniecznego aby Scottie szedł przez siebie.

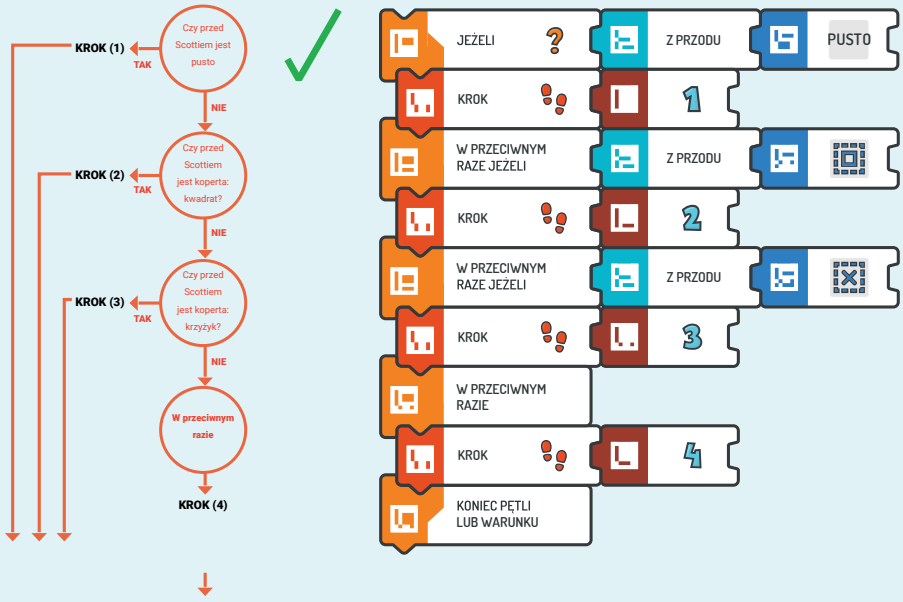
Wystarczy, że określimy jak i na jaką szczególną sytuację ma zareagować.



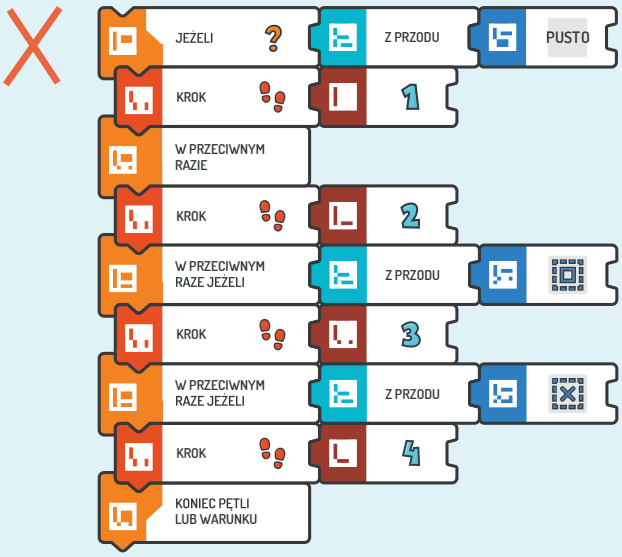


WPROWADZENIE DO MODUŁU (B): instrukcja W PRZECIWNYM RAZIE 3/3

Możemy używać instrukcji **W PRZECIWNYM RAZIE JEŻELI** oraz **W PRZECIWNYM RAZIE** w jednym programie,



ale **W PRZECIWNYM RAZIE MOŻE zostać tylko raz i zawsze na końcu programu.** Poniższe fragmenty programu będą odrzucone jako złożone niepoprawnie.

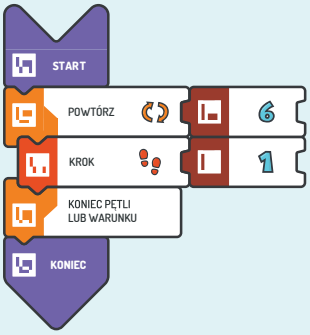


WPROWADZENIE DO MODUŁU (C): [opcjonalnie] - POWTÓRZ (CIĄGLE) 1/3

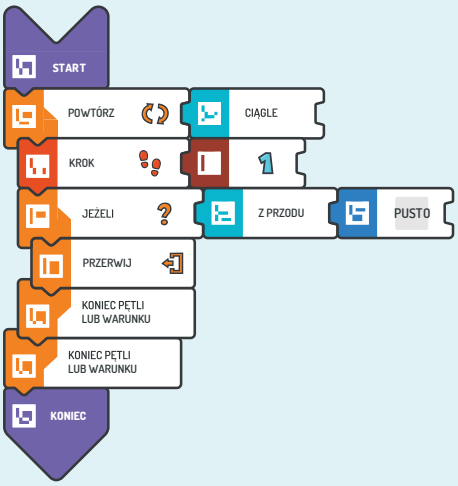
W programach z instrukcją warunkową bez użycia pętli **POWTÓRZ** instrukcja warunkowa sprawdzi warunek tylko raz i program zostanie zakończony. Jednak nam zależy, aby sprawdzanie warunku odbywało się wielokrotnie tak długo, aż Scottie znajdzie i zbierze wszystkie śruby i narzędzia. Temu zadaniu może służyć nowy klocek **CIĄGLE**.

W zadaniach w tym module istnieje konieczność użycia zatem pętli **POWTÓRZ** - w zależności od wieku i doświadczenia uczniów - na jeden z dwóch sposobów:

1. **POWTÓRZ określoną ilość razy** – konstrukcja dobrze znana uczniom z modułu 3. Jedyną trudnością jest określenie, ile razy program będzie musiał być powtarzany. **W poniższym przykładzie Scottie wykona 1 krok 6 razy niezależnie od okoliczności przyrody.**



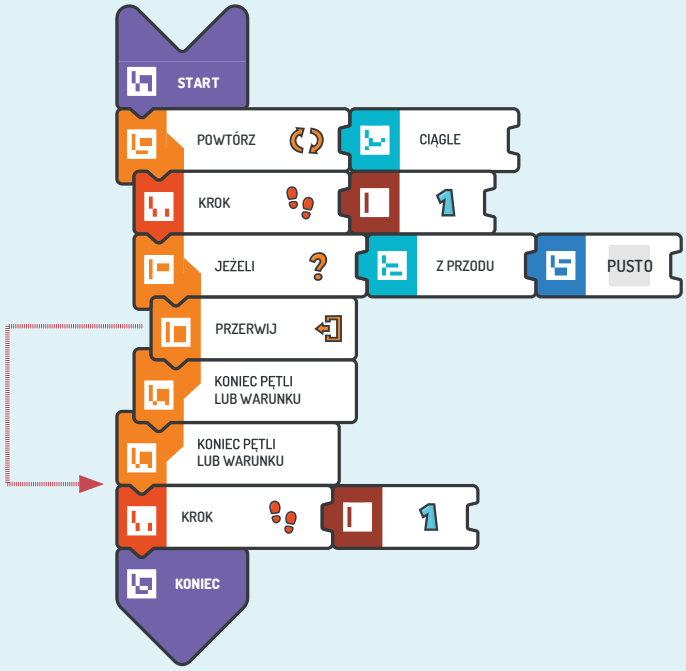
2. POWTÓRZ (CIĄGLE) wraz z instrukcją **PRZERWIJ**.



Konstrukcja ta nie jest wprowadzona w samouczku, jednak jest ciekawszym rozwiązaniem i najbardziej uniwersalnym. **Podobny program omówimy szczegółowo na następnej stronie.**

WPROWADZENIE DO MODUŁU (C): [opcjonalnie] - POWTÓRZ (CIĄGLE) 2/3

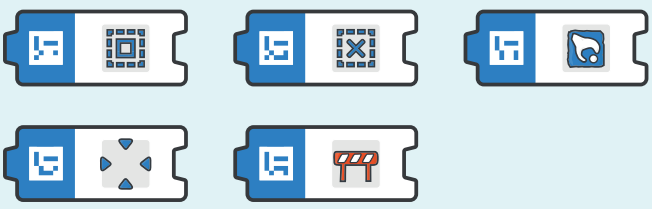
POWTÓRZ (CIĄGLE) wraz z instrukcją **PRZERWIJ**.



W powyższym programie instrukcja krok 1 wykonywana jest bez końca do czasu, aż program napotka instrukcję **PRZERWIJ (**którą z reguły umieszczamy w stosownej instrukcji warunkowej - np. odnoszącej się do odpowiedniego obiektu na scenie). **W naszym przykładzie Scottie przestanie wykonywać pętlę jeżeli przed nim będzie puste pole.**

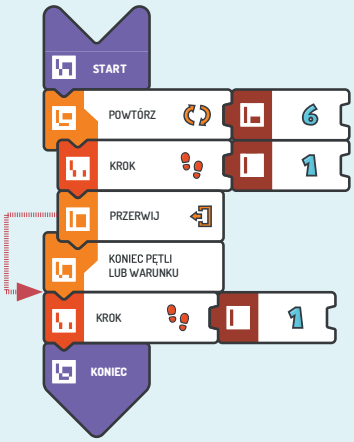
Warunek wywołujący instrukcję **PRZERWIJ** umieszczony w strategicznym miejscu, będzie kluczowy dla działania ułożonego przez nas kodu. **Instrukcje w pętli umieszczone po instrukcji PRZERWIJ nie zostaną wywołane.**

W tym module na podłodze fabryki znajdziemy **koperty KWADRAT** i **KRZYŻYK**, **przedmioty do zebrania**, **pola akcji, oraz przeszkody**. Wszytkie będą mogły posłużyć jako punkty odniesienia, w oparciu o które skonstruujemy odpowiednie **instrukcje warunkowe**.

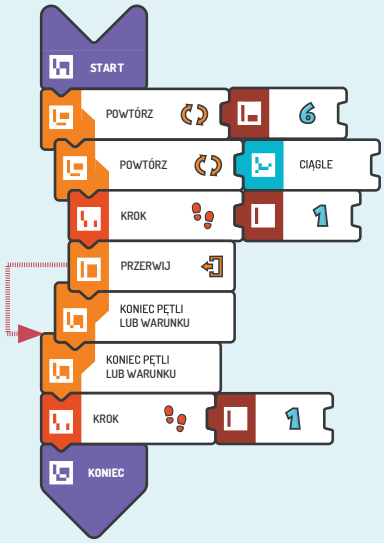


WPROWADZENIE DO MODUŁU (C): [opcjonalnie] - POWTÓRZ (CIĄGLE) 3/3

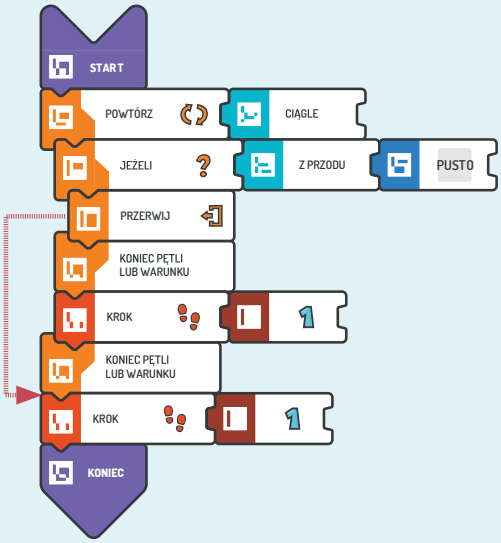
Instrukcja PRZERWIJ „wyrzuci” nas poza każdą pętlę **bezpośrednio ją zawierającą.**



Scottie wykona 2 kroki.
(1 w pętli;
1 poza pętlą)



Scottie wykona 7 kroków.
(1 w pętli dla
każdej pętli **CIĄGLE**
powtórzonej 6 razy;
1 poza pętlami)

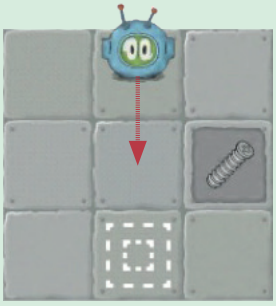


Scottie będzie szedł dopóki nie zatrzyma się na pustym polu.
(jeżeli przed
Scottiem nie jest
pusto, Scottie
wykonuje po 1dnym
kroku bez końca;
jeżeli przed Scottiem
jest puste pole
wychodzimy z pętli;
na zewnątrz pętli
robimy jeszcze 1
krok przed klockiem
KONIEC)

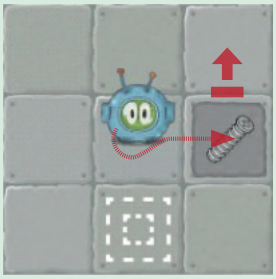
WPROWADZENIE: W PRZECIWNYM RAZIE (1/2)

W zadaniach w tym module istnieje konieczność użycia pętli **POWTÓRZ** w zestawieniu z **instukcjami warunkowymi**. W tym zadaniu użyj instrukcji **W PRZECIWNYM RAZIE.**

W PRZECIWNYM RAZIE pozwala opisać zestaw intrukcji które Scottie wykona **tylko i wyłącznie** jeżeli wcześniejsze warunki nie zostały spełnione. Na przykładzie gotowego programu, **w którym Scottie orientuje się w terenie w odniesieniu do kopert umieszczonych na planszy**:



1SZE POWTÓRZENIE PĘTLI:
Przed Scottiem nie ma koperty z kwadratem ->
Scottie realizuje instrukcje umieszczone pomiędzy klockami
W PRZECIWNYM RAZIE oraz
KONIEC PĘTLI LUB WARUNKU



2GIE POWTÓRZENIE PĘTLI:
Przed Scottiem jest teraz koperta z kwadratem ->
Scottie realizuje instrukcje umieszczone pomiędzy klockami
JEŻELI oraz
W PRZECIWNYM RAZIE

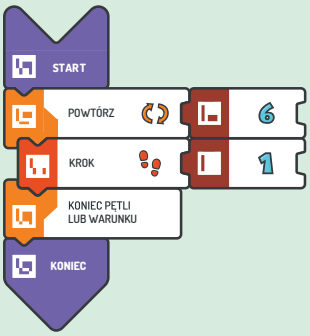
WPROWADZENIE: W PRZECIWNYM RAZIE (2/2)

UWAGA.

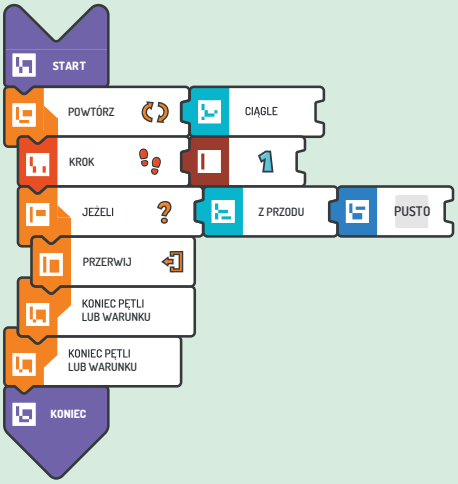
W tym i kolejnych zadaniach możesz użyć konstrukcji POWTÓRZ (CIĄGLE).

W zadaniach w tym module istnieje konieczność użycia pętli **POWTÓRZ** - w zależności od wieku i doświadczenia uczniów - na jeden z dwóch sposobów:

1. **POWTÓRZ określoną ilość razy** – konstrukcja dobrze znana uczniom z modułu 3. Jedyną trudnością jest określenie, ile razy program będzie musiał być powtarzany. **W poniższym przykładzie Scottie wykona 1 krok 6 razy niezależnie od okoliczności przyrody.**



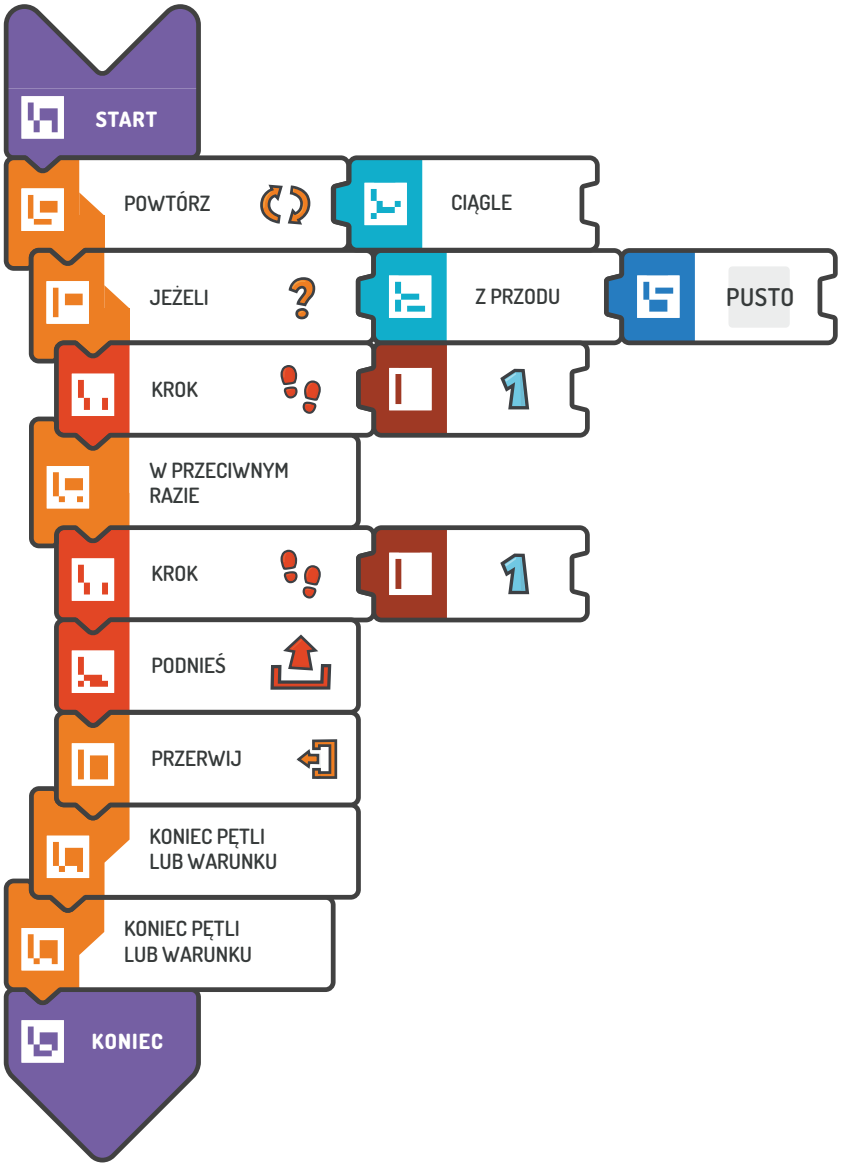
2. POWTÓRZ (CIĄGLE) wraz z instrukcją **PRZERWIJ**.



Konstrukcja ta nie jest wprowadzona w samouczku, jednak jest ciekawszym rozwiązaniem i najbardziej uniwersalnym. Możesz przeczytać o nim więcej we **Wpowadzeniu do modułu C.**

Wyzwanie **(W tym module użyj instrukcji warunkowej W PRZECIWNYM RAZIE)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej W PRZECIWNYM RAZIE)**:

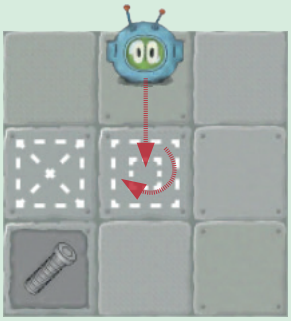
Przykładowe rozwiązanie (format tekstowy):

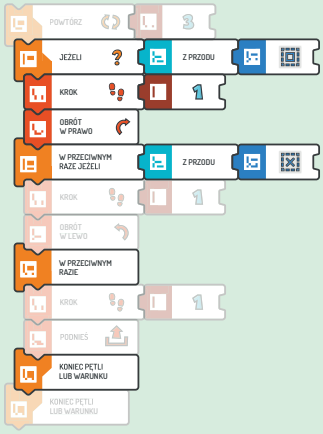
```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRZODU < PUSTO) {
            KROK (1)
        } W PRZECIWNYM RAZIE {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        }
    }
KONIEC
```

WPROWADZENIE:
W PRZECIWNYM RAZIE JEŻELI i KOPERTY (1/3)

W zadaniach w tym module musimy użyć pętli **POWTÓRZ** w zestawieniu z **instukcjami warunkowymi**. W tym zadaniu użyj nowej instrukcji **W PRZECIWNYM RAZIE JEŻELI.**

W PRZECIWNYM RAZIE JEŻELI pozwala opisać **dodatkowe warunki dla Scottiego do sprawdzenia** jeżeli pierwszy warunek nie będzie spełniony. Scottie będzie sprawdzał warunki po kolei, **dopóki dopóki któryś z nich nie zostanie spełniony.** Na przykład:





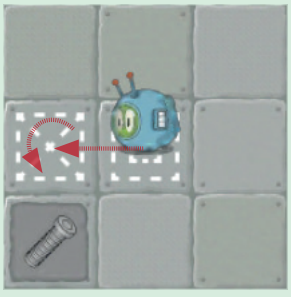
1SZE POWTÓRZENIE PĘTLI:

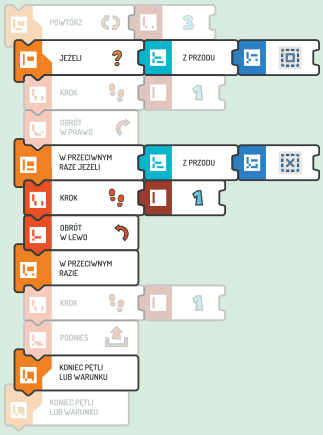
Przed Scottiem jest koperta z kwadratem ->

Scottie realizuje:

KROK (1)

OBRÓT W PRAWO





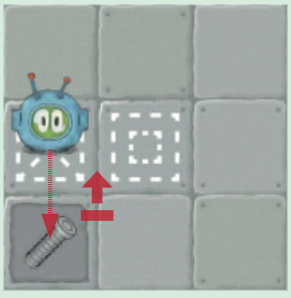
2GIE POWTÓRZENIE PĘTLI:

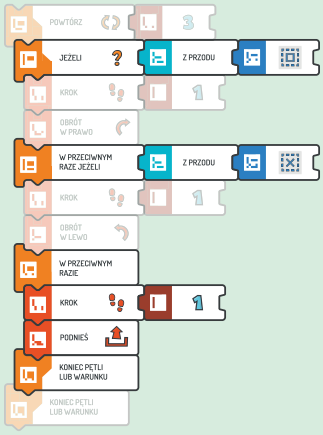
Przed Scottiem jest koperta z krzyżykiem ->

Scottie realizuje:

KROK (1)

OBRÓT W LEWO





3CIE POWTÓRZENIE PĘTLI:

Przed Scottiem nie ma koperty ->

Scottie realizuje:

KROK (1)

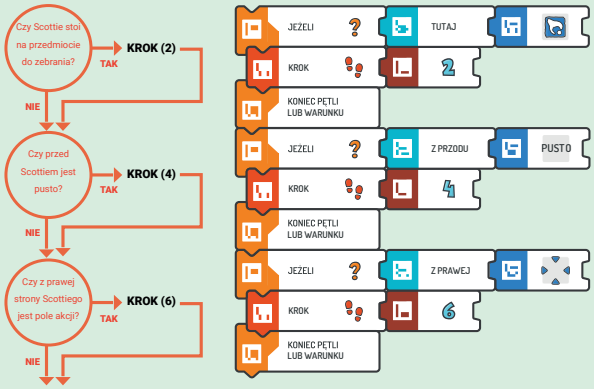
PODNIEŚ

WPROWADZENIE:
W PRZECIWNYM RAZIE JEŻELI i KOPERTY (2/3)

Instrukcja **W PRZECIWNYM RAZIE JEŻELI** pozwala nam opisać alternatywne zestawy instrukcji, **z których wykanamy tylko jeden.**

Bez użycia

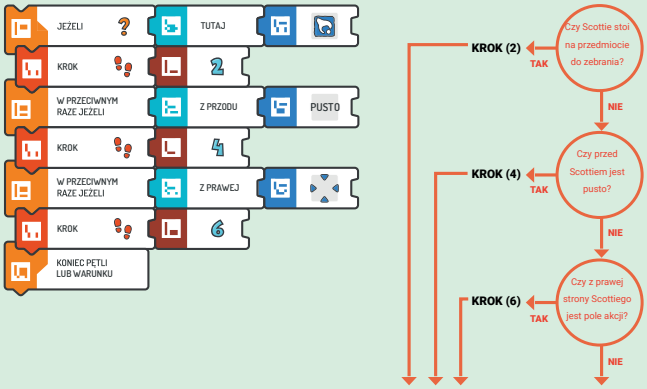
W PRZECIWNYM RAZIE JEŻELI



Scottie sprawdzi warunki po kolei. Możliwy jest scenariusz, w którym Scottie wykona w sumie 2+4+6 = 12 kroków!

Z użyciem

W PRZECIWNYM RAZIE JEŻELI



Scottie sprawdzi warunki po kolei. Spełnienie dowolnego z warunków wyklucza realizację pozostałych. Scottie wykona 2,4, lub 6 kroków.

PAMIĘTAJ.

W tym i kolejnych zadaniach możesz użyć konstrukcji POWTÓRZ (CIĄGLE).

Konstrukcja ta nie jest wprowadzona w samouczku, jednak jest ciekawszym rozwiązaniem i najbardziej uniwersalnym. Możesz przeczytać o nim więcej we **Wpowadzeniu do modułu C.**

WPROWADZENIE:
W PRZECIWNYM RAZIE JEŻELI i KOPERTY (3/3)

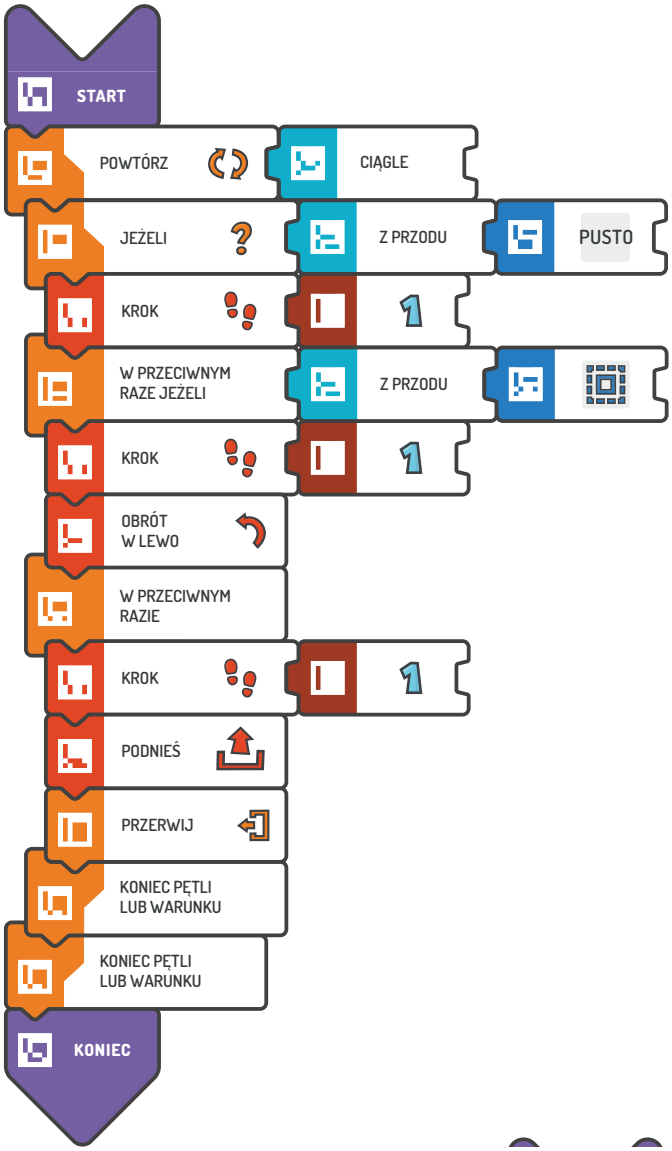
W tym module na podłodze warsztatu poza znanymi nam już **przedmiotami do zebrania**, **polami akcji, oraz przeszkodami**, znajdziemy **kopertę: KWADRAT** oraz **kopertę: KRZYŻYK**.

Pracownicy warsztatu używają ich jako puntków orientacyjnych, z pomocą których organizują dla siebie przestrzeń pracy. Scottiemu będą mogły posłużyć jako punkty odniesienia, w oparciu o które skonstruujemy odpowiednie **instrukcje warunkowe JEŻELI,** oraz **W PRZECIWNYM RAZIE JEŻELI**.

Użyj koperty aby skonstruować instrukcję warunkową w tym zadaniu.

Wyzwanie **(W tym module użyj instrukcji warunkowej W PRZECIWNYM RAZIE JEŻELI)**:

Przykładowe rozwiązanie:



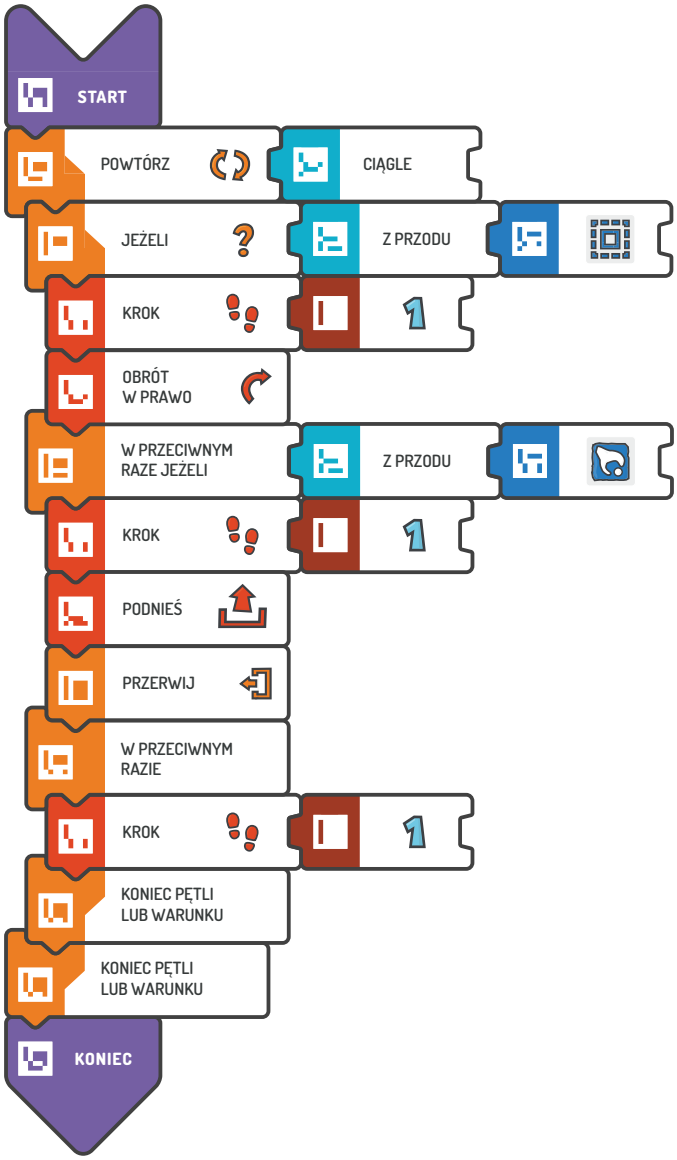
Wyzwanie **(W tym module użyj instrukcji warunkowej W PRZECIWNYM RAZIE JEŻELI)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRZODU < PUSTO) {
            KROK (1)
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < KOPERTA: KWADRAT) {
            KROK (1)
            OBRÓT W LEWO
        } W PRZECIWNYM RAZIE {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        }
    }
KONIEC
```

Wyzwanie **(W tym module użyj złożonych instrukcji warunkowych)**:

Przykładowe rozwiązanie:



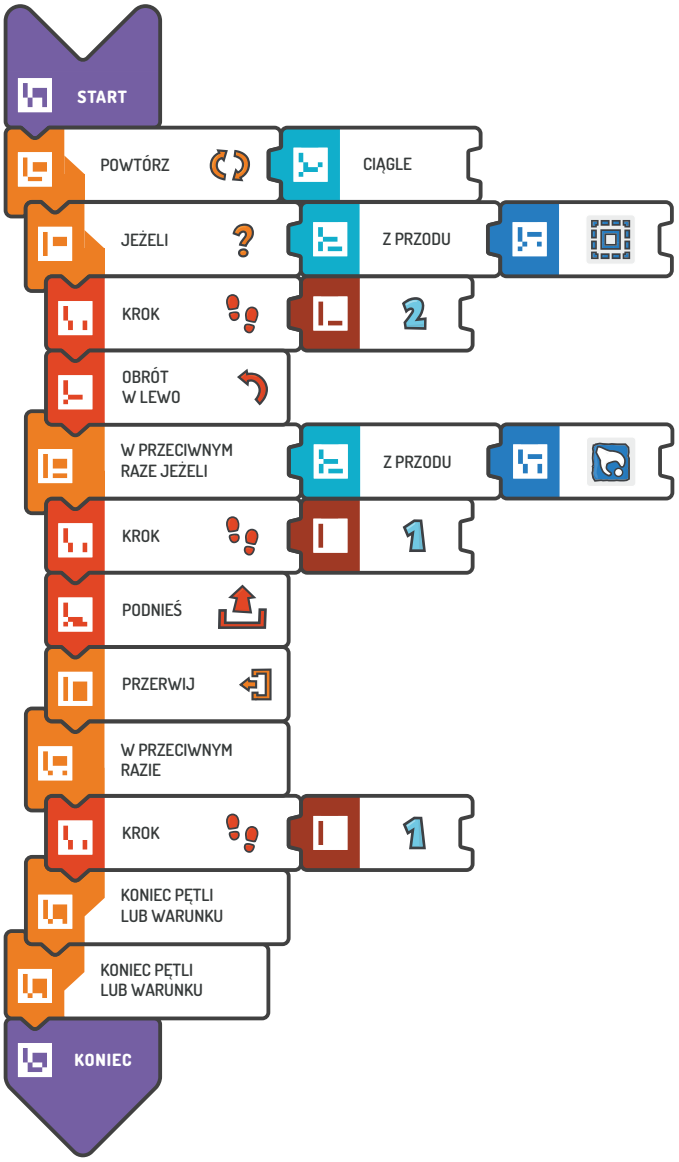
Wyzwanie **(W tym module użyj złożonych instrukcji warunkowych)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRZODU < KOPERTA: KWADRAT) {
            KROK (1)
            OBRÓT W PRAWO
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < OBIEKT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

Wyzwanie **(W tym module użyj złożonych instrukcji warunkowych)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj złożonych instrukcji warunkowych)**:

Przykładowe rozwiązanie (format tekstowy):

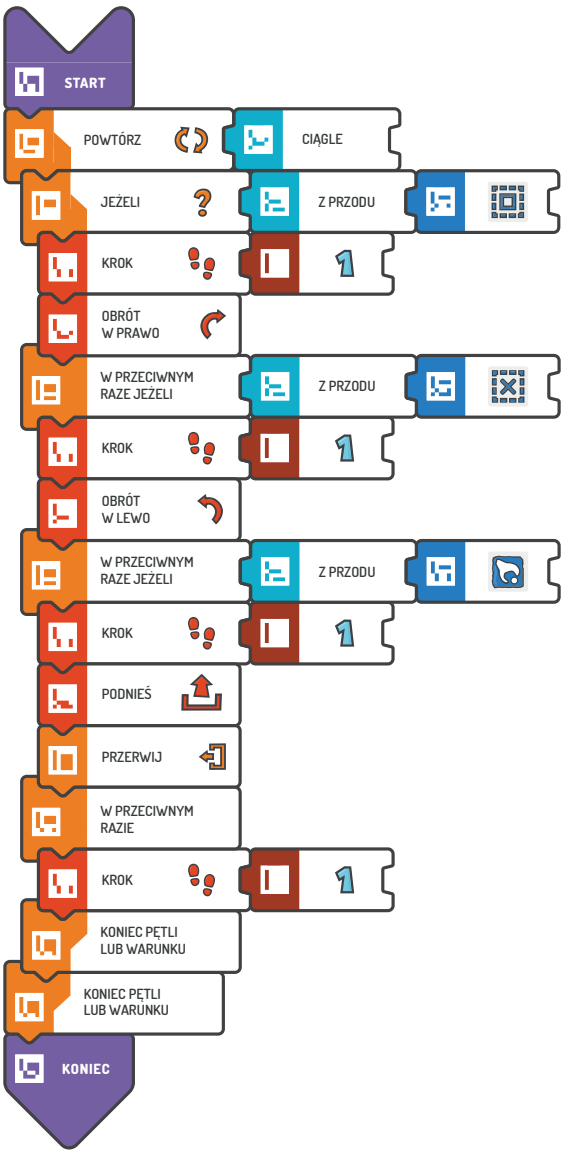
```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRZODU < KOPERTA: KWADRAT) {
            KROK (2)
            OBRÓT W LEWO
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < OBIEKT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

WSKAZÓWKA

Możemy używać wielu instrukcji **W PRZECIWNYM RAZIE JEŻELI** w jednym programie.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



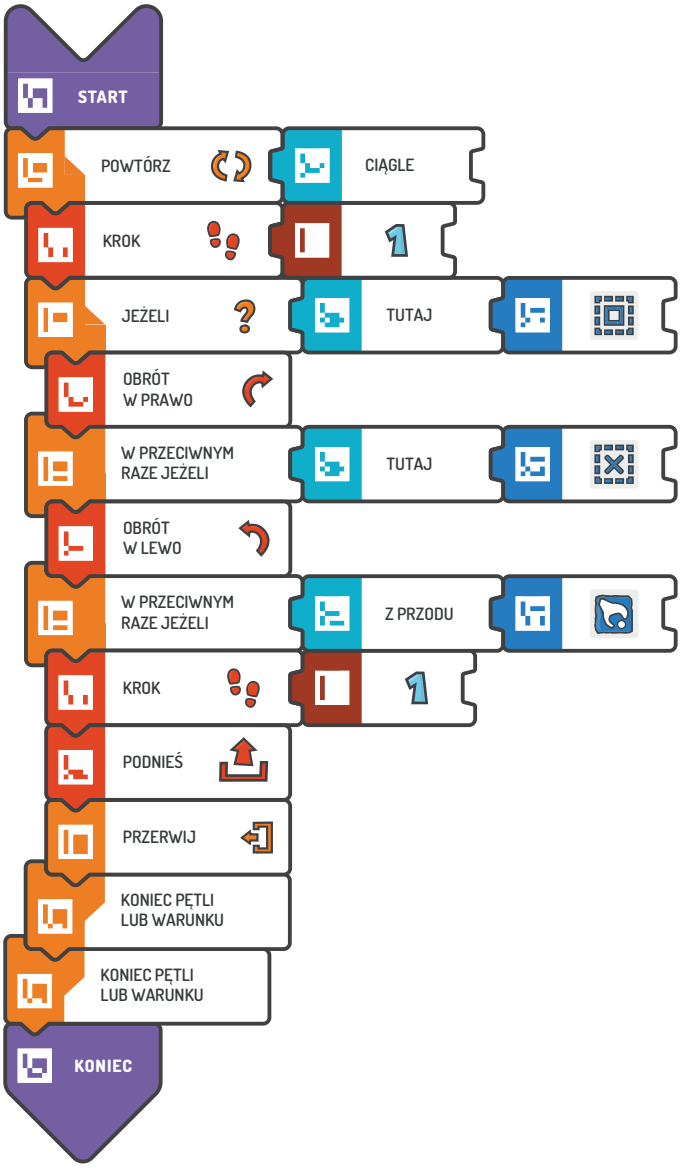
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRZODU < KOPERTA: KWADRAT) {
            KROK (1)
            OBRÓT W PRAWO
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < KOPERTA: KRZYŻYK) {
            KROK (1)
            OBRÓT W LEWO
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < OBIEKT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

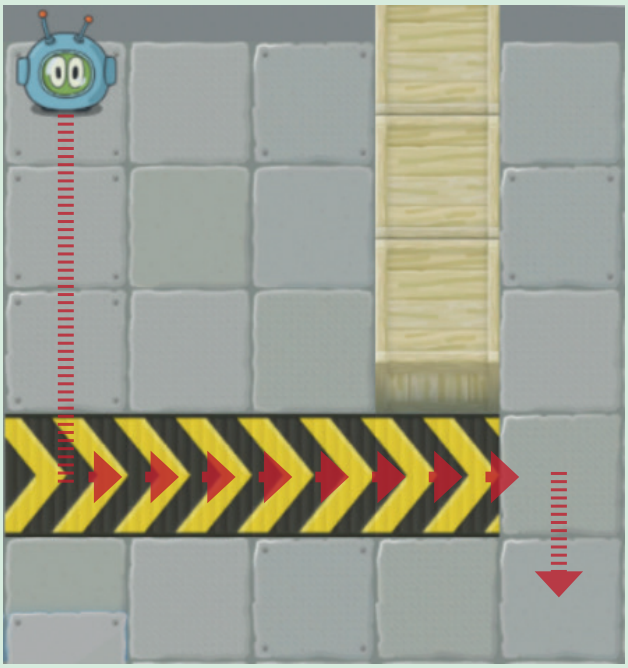
Przykładowe rozwiązanie (format tekstowy):

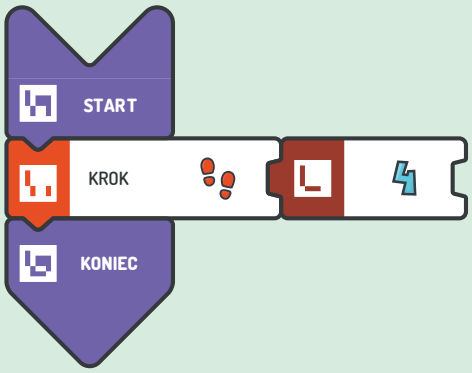
```
START
    POWTÓRZ (CIĄGLE) {
        KROK (1)
        JEŻELI (TUTAJ < KOPERTA: KWADRAT) {
            OBRÓT W PRAWO
        } W PRZECIWNYM RAZIE JEŻELI
        (TUTAJ < KOPERTA: KRZYŻYK) {
            OBRÓT W LEWO
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < PRZEDMIOT DO ZEBRANIA) {
            KROK(1)
            PODNIEŚ
            PRZERWIJ
        }
    }
KONIEC
```

WPROWADZENIE:
Taśmociągi

W warsztacie znajdziemy **pasy transmisyjne.** Scottie może dzięki nim dotrzeć podróżować po planszy bez konieczności budowania złożonych programów.

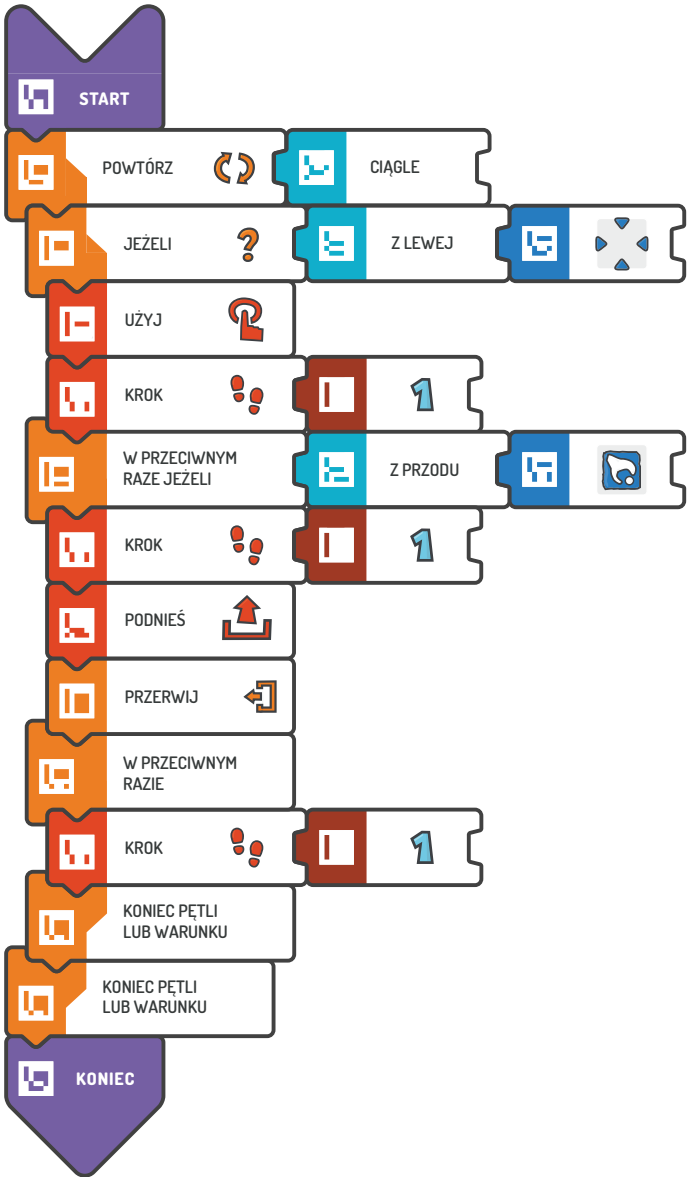
Wejście na pas transmisyjny **przerywa program, który Scottie dokończy po dojechaniu na miejce.**





Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            UŻYJ
            KROK (1)
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < PRZEDMIOT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

WPROWADZENIE:
Dźwignie

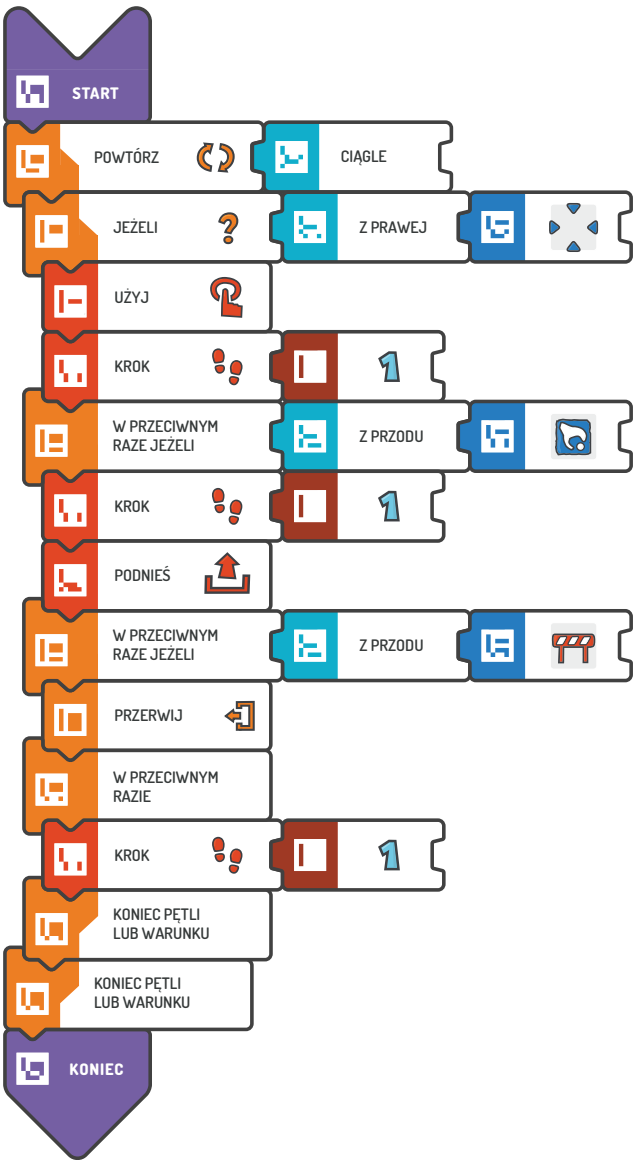
Niektóre taśmociągi w warsztacie są wyłączone. Aby uruchomić wyłączomy taśmociąg, znajdź i aktywuj na planszy dźwignię za pomocą instrukcji **UŻYJ**.

Pamiętaj!

Scottie nie może być zwrócony do dźwigni plecami, aby instrukcja UŻYJ mogła zadziałać.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



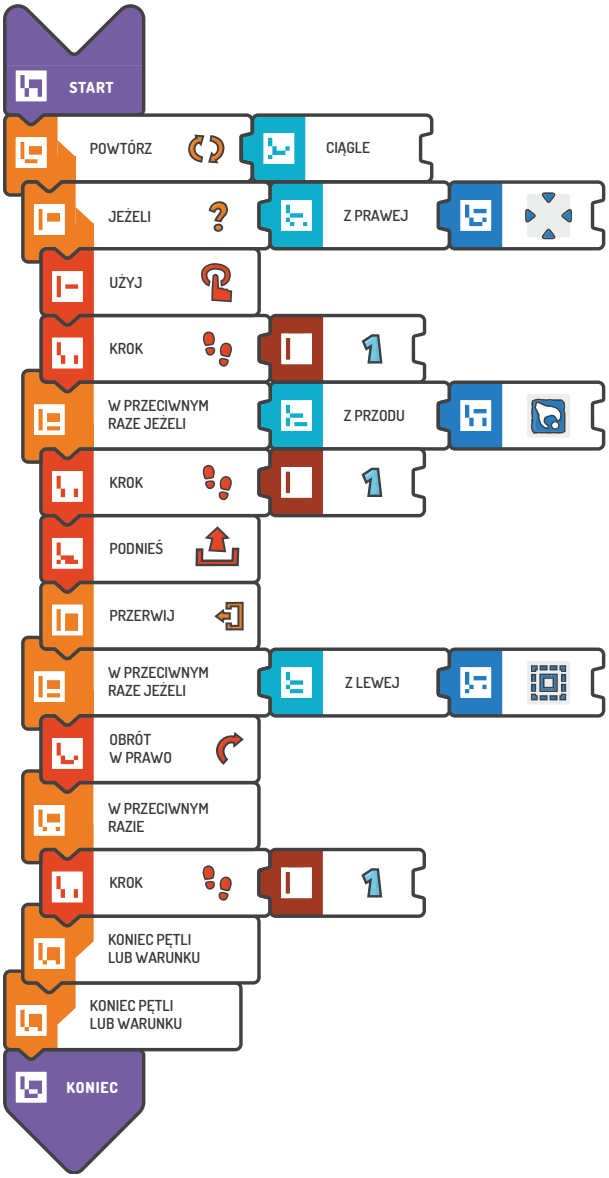
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UŻYJ
            KROK (1)
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < OBIEKT DO ZBIERANIA) {
            KROK (1)
            PODNIEŚ
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < PRZESZKODA) {
            PRZERWIJ
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (CIĄGLE) {
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UŻYJ
            KROK (1)
        } W PRZECIWNYM RAZIE JEŻELI
        (Z PRZODU < PRZEDMIOT DO ZEBRANIA) {
            KROK (1)
            PODNIEŚ
            PRZERWIJ
        } W PRZECIWNYM RAZIE JEŻELI
        (Z LEWEJ < KOPERTA: KWADRAT) {
            OBRÓT W PRAWO
        } W PRZECIWNYM RAZIE {
            KROK (1)
        }
    }
KONIEC
```

Moduł X
Afryka

Instrukcje warunkowe - JEŻELI

Zasadź brakujące drzewka w odpowiednich polach.

W tym module musisz użyć instrukcji warunkowej.

WPROWADZENIE DO MODUŁU 1/4: Instrukcje warunkowe

Codziennie każdy z nas podejmuje jakieś decyzje. Czy założyć kurtkę do pracy lub szkoły? Ile zjem kanapek na śniadanie? Czy do szkoły lub pracy pojechać samochodem, metrem lub autobusem? Programy komputerowe wciąż coś sprawdzają, przeliczają i podejmują decyzje. Pisząc program i wprowadzając takie instrukcje musisz wszystko dokładnie komputerowi wytłumaczyć, aby wiedział co w jakiej sytuacji zrobić.

Instrukcja JEŻELI, W PRZECIWNYM RAZIE

To podstawowa instrukcja warunkowa związana z podejmowaniem decyzji. Instrukcja warunkowa jest potężnym narzędziem i w związku z powyższym występuje w prawdopodobnie większości programów.

Kluczowe elementy instrukcji warunkowej to:

1. Warunek, który program sprawdza,

WPROWADZENIE DO MODUŁU 2/4: Instrukcje warunkowe

2. Instrukcje, które zostaną wykonane, gdy spełniony zostanie ten warunek,

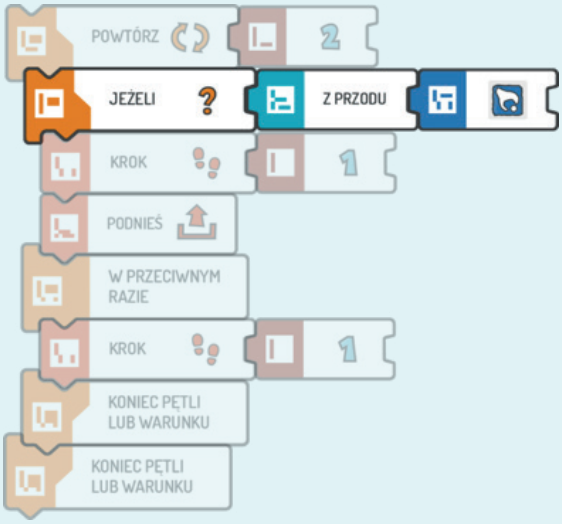
3. Instrukcje, które zostaną wykonane, gdy warunek nie będzie spełniony.

Zobaczymy to na przykładzie. Dla lepszego zrozumienia tego zagadnienia instrukcję warunkową wywołamy dwukrotnie **poprzez umieszczenie instrukcji warunkowej wewnątrz pętli POWTÓRZ (2)**

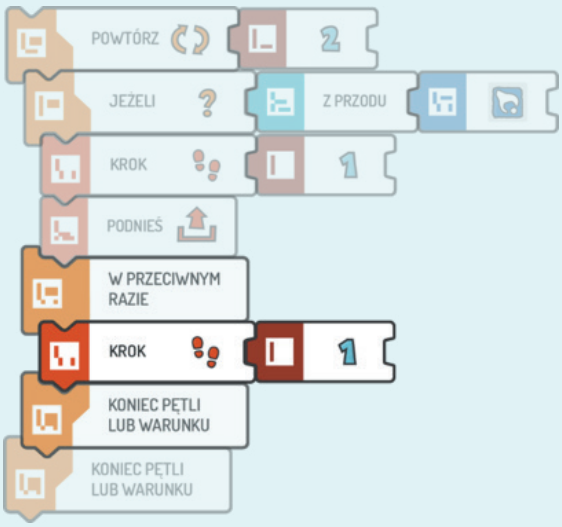
WPROWADZENIE DO MODUŁU 3/4: Instrukcje warunkowe

Przykład zastosowania instrukcji warunkowej.

1. Scottie sprawdza pierwszy raz warunek: **„Czy z przodu jest OBIEKT DO ZEBRANIA?”** Warunek nie jest spełniony.



2. Gdy warunek nie jest spełniony wywoływana jest instrukcja **W PRZECIWNYM RAZIE**. Scottie porusza się do przodu o jedno pole - **KROK (1)**.



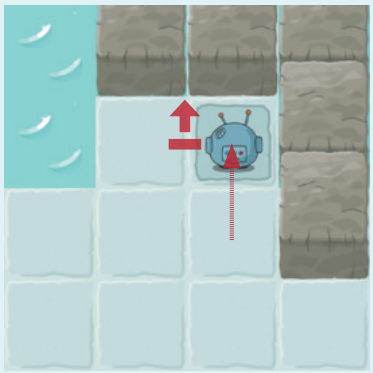
...

WPROWADZENIE DO MODUŁU 4/4: Instrukcje warunkowe

Przykład zastosowania instrukcji warunkowej c.d.

3. Następnie drugi raz - zastosowaliśmy pętlę **POWTÓRZ (2)** - sprawdzany jest warunek: **„Czy z przodu OBIEKT DO ZEBRANIA?”**. Warunek jest spełniony.

4. Gdy warunek jest spełniony wywoływana jest instrukcja warunkowa **JEŻELI**. Scottie wykonuje instrukcje **KROK (1)** i **PODNIEŚ**.



Gdy w instrukcji warunkowej jeden z warunków zostanie spełniony, drugi warunek nie jest już sprawdzany, ani wykonywany.

WPROWADZENIE:
Pole akcji

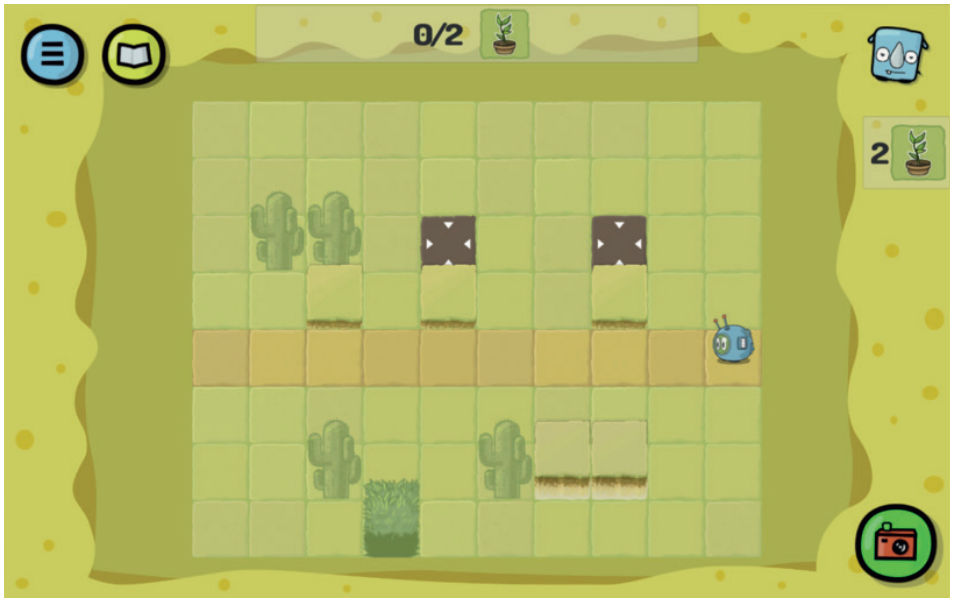
Jeden z naszych zwierzęcych bohaterów ma wspaniały ogród. W zamian za pomoc Scottie zobowiązał się zasadzić w nim kilka roślin.

Zasadź brakujące drzewka w odpowiednich polach wykorzystując do tego nowe klocki, w szczególności pętlę **JEŻELI** w połączeniu z klockiem **POLE AKCJI.** Położenie pola akcji w zadaniu względem Scottiego będzie warunkiem, który będzie sprawdzany.

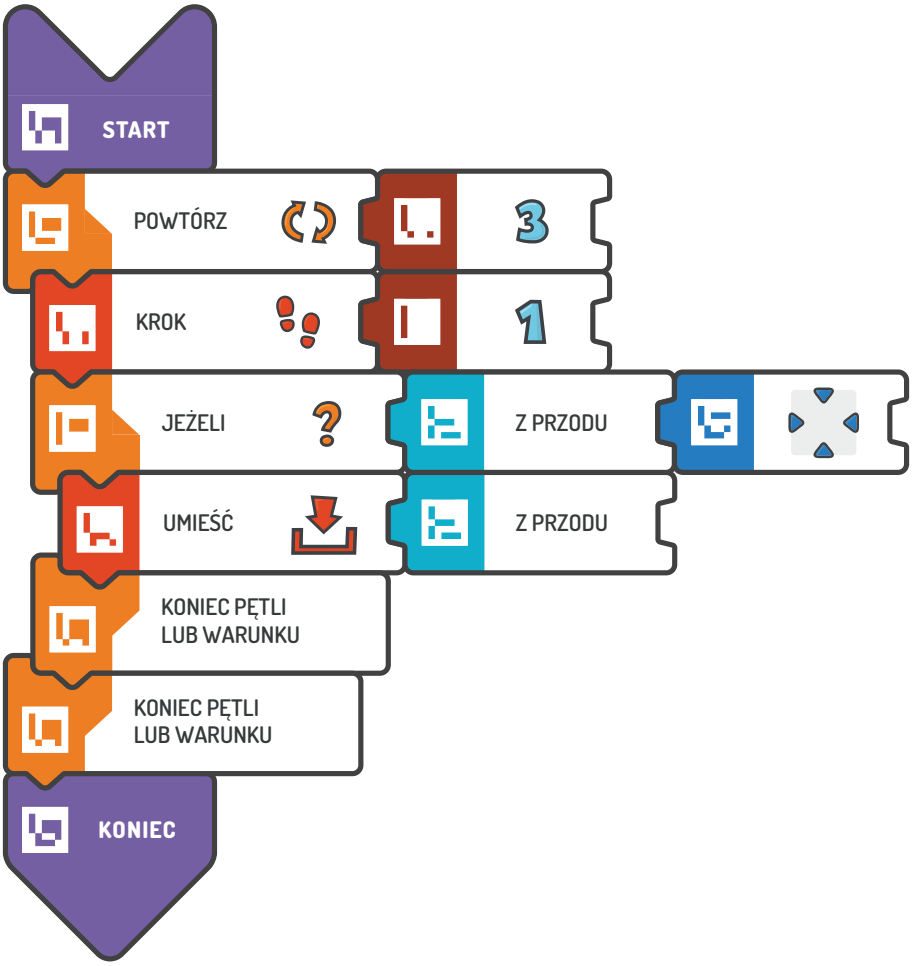
PAMIĘTAJ!

Na końcu warunku umieszczamy klocek **KONIEC PĘTLI LUB WARUNKU**.

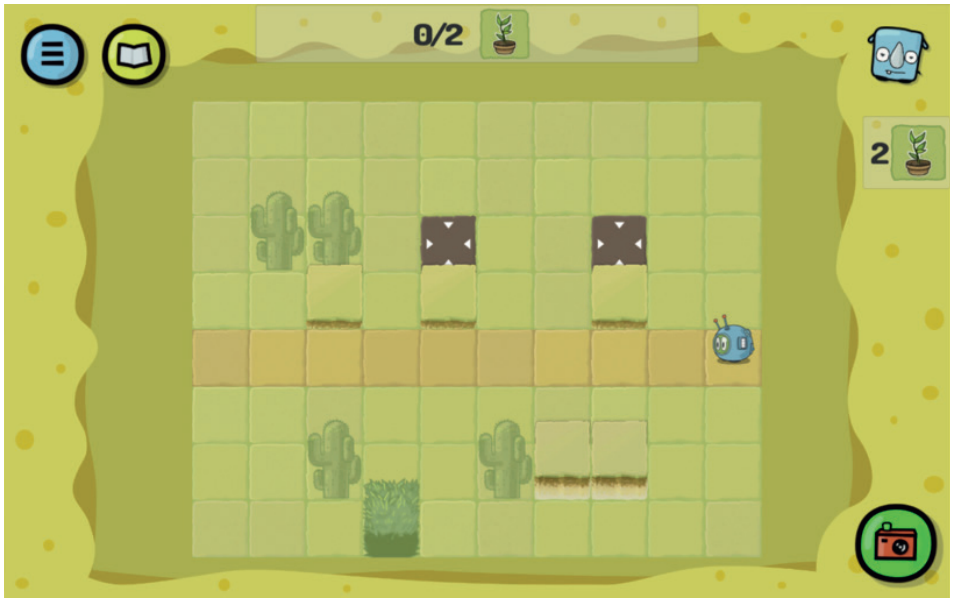
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:



Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

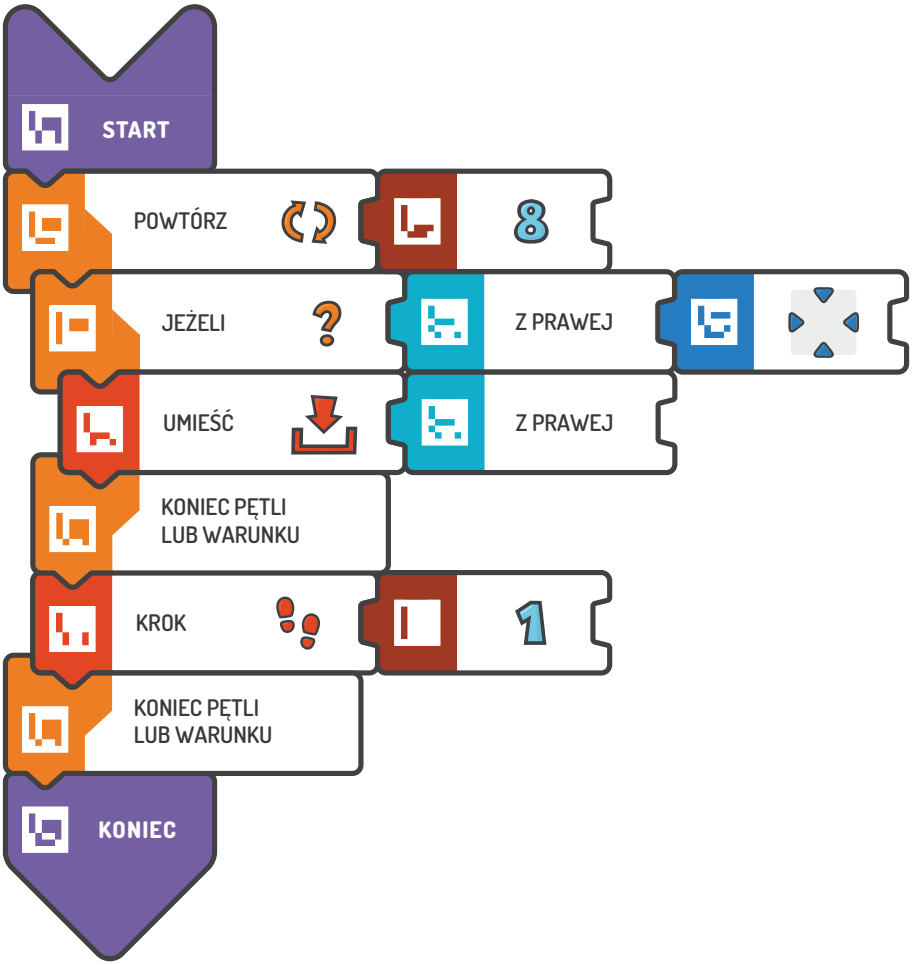


Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (3) {
        KROK (1)
        JEŻELI (Z PRZODU < POLE AKCJI) {
            UMIEŚĆ (Z PRZODU)
        }
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

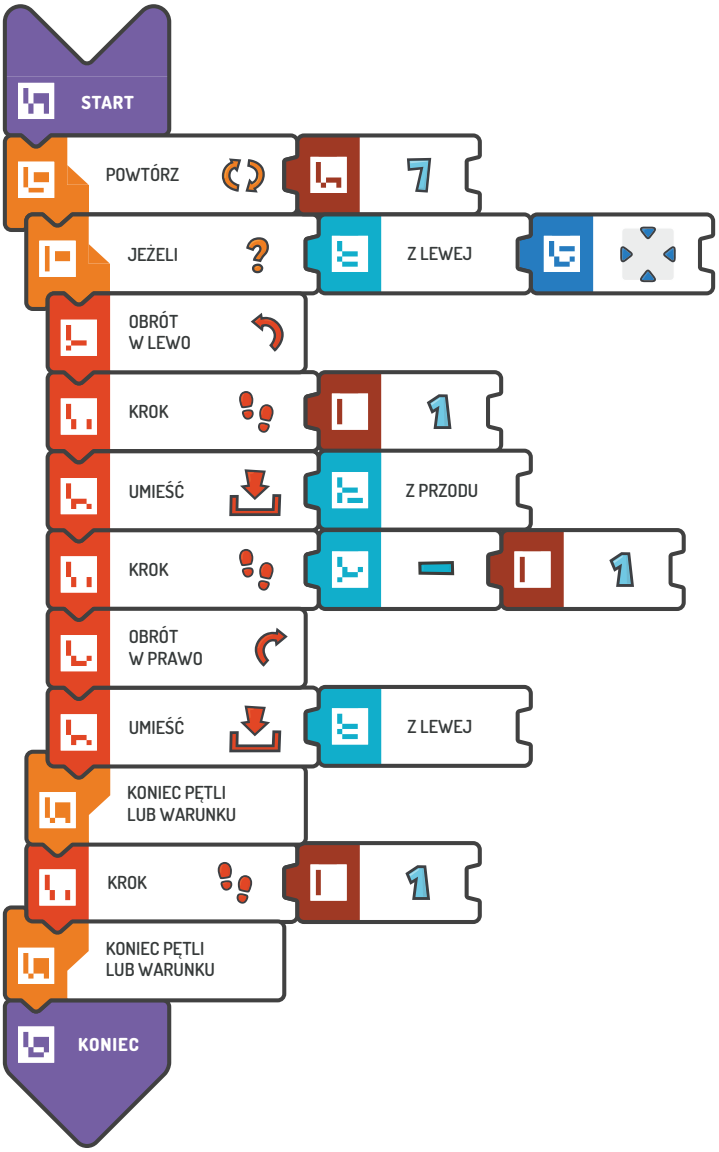
```
START
    POWTÓRZ (8) {
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
        KROK (1)
    }
KONIEC
```

UWAGA

Twój poprzedni program potrafi skutecznie rozwiązać także to zadanie. **Wystarczy zmienić wartość zmiennej X.**

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

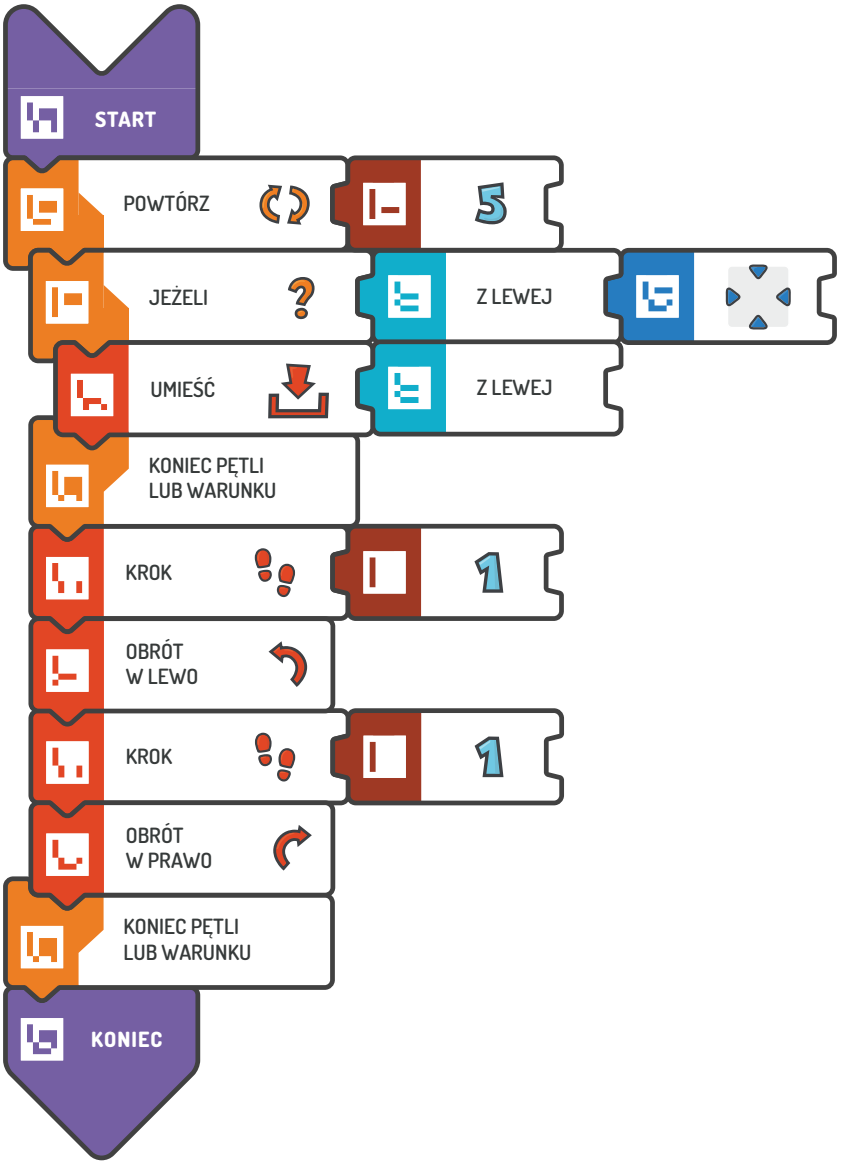
```
START
    POWTÓRZ (7) {
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            OBRÓT W LEWO
            KROK (1)
            UMIEŚĆ (Z PRZODU)
            KROK (-1)
            OBRÓT W PRAWO
            UMIEŚĆ (Z LEWEJ)
        }
        KROK (1)
    }
KONIEC
```

WPROWADZENIE: X i Y

W grze Scottie GO! możemy stworzyć i wykorzystać jednocześnie dwie zmienne: **X** i **Y**.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



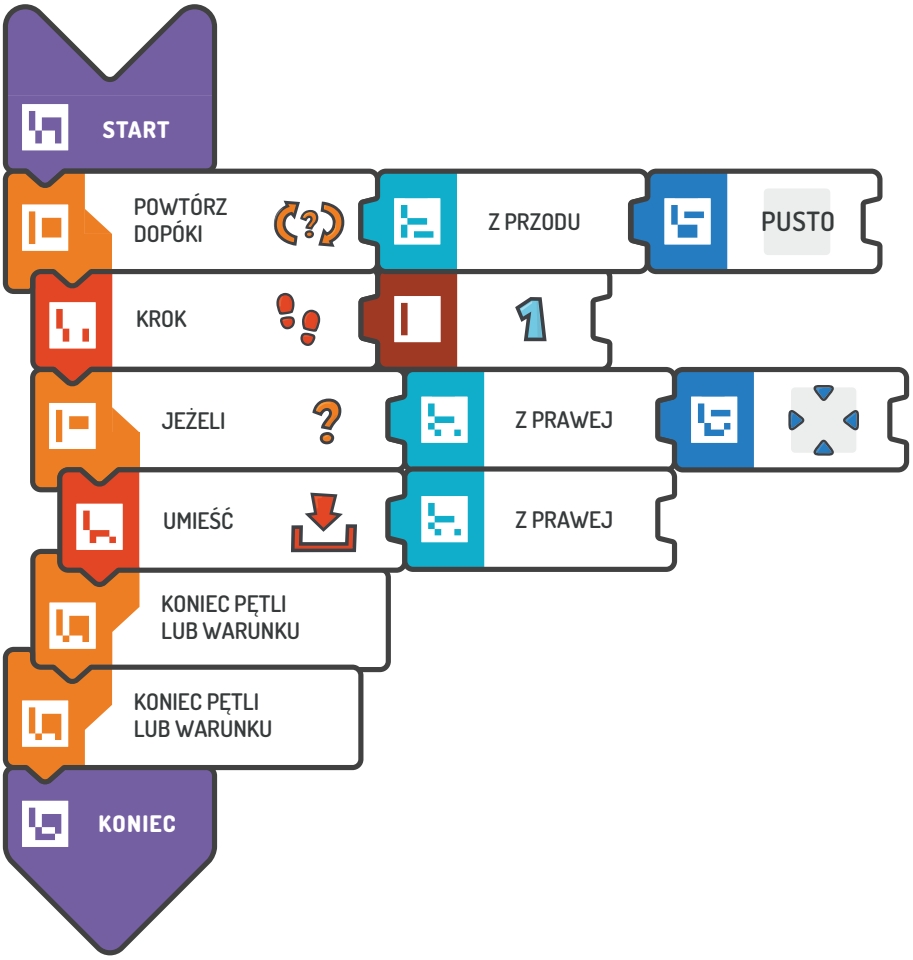
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    POWTÓRZ (5) {
        JEŻELI (Z LEWEJ < POLE AKCJI) {
            UMIEŚĆ (Z LEWEJ)
        }
        KROK (1)
        OBRÓT W LEWO
        KROK (1)
        OBRÓT W PRAWO
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

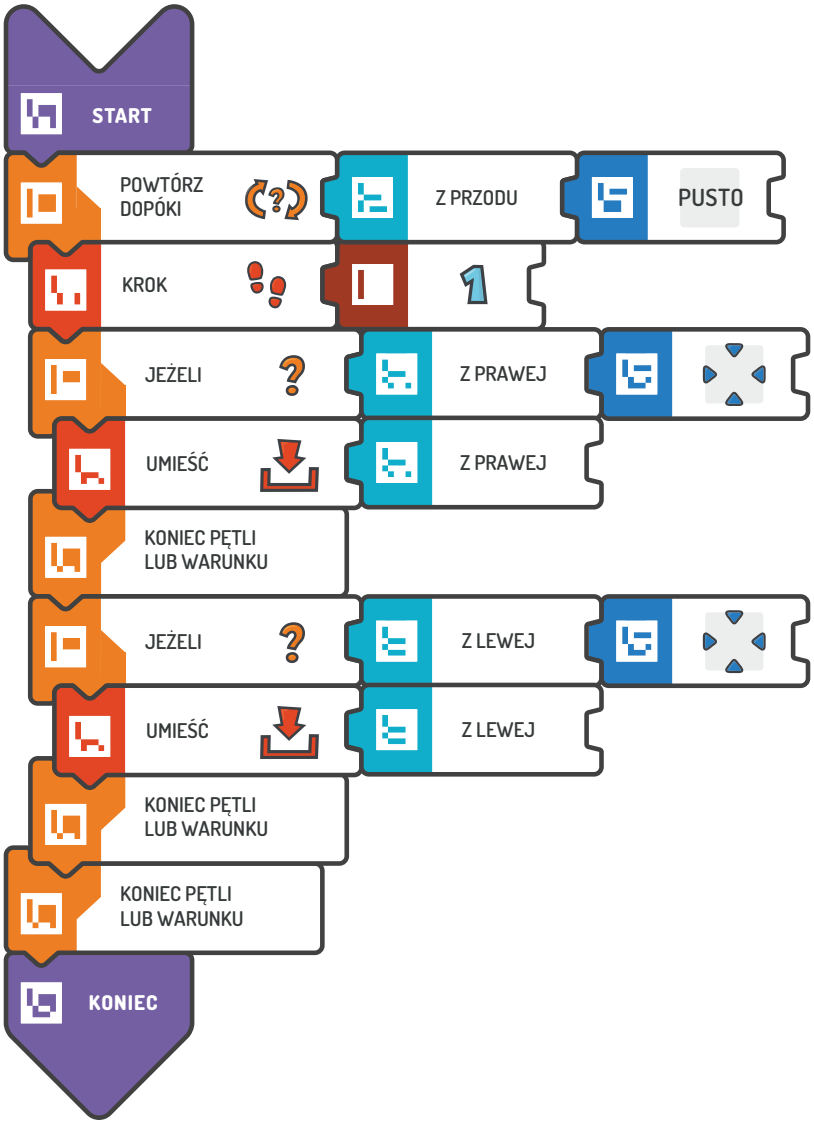
```
START
    POWTÓRZ DOPÓKI (Z PRZODU < PUSTO) {
        KROK (1)
        JEŻELI (Z PRAWEJ < POLE AKCJI) {
            UMIEŚĆ (Z PRAWEJ)
        }
    }
KONIEC
```

WPROWADZENIE:
Zmienne i pętle

Wykorzystaj w zadaniu dwie zmienne **X** i **Y** wraz z pętlą **POWTÓRZ**. Użyj zmiennej **X**, aby przechowywać ilość powtórzeń pętli oraz zmienną **Y**, aby przechowywać ilość kroków.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



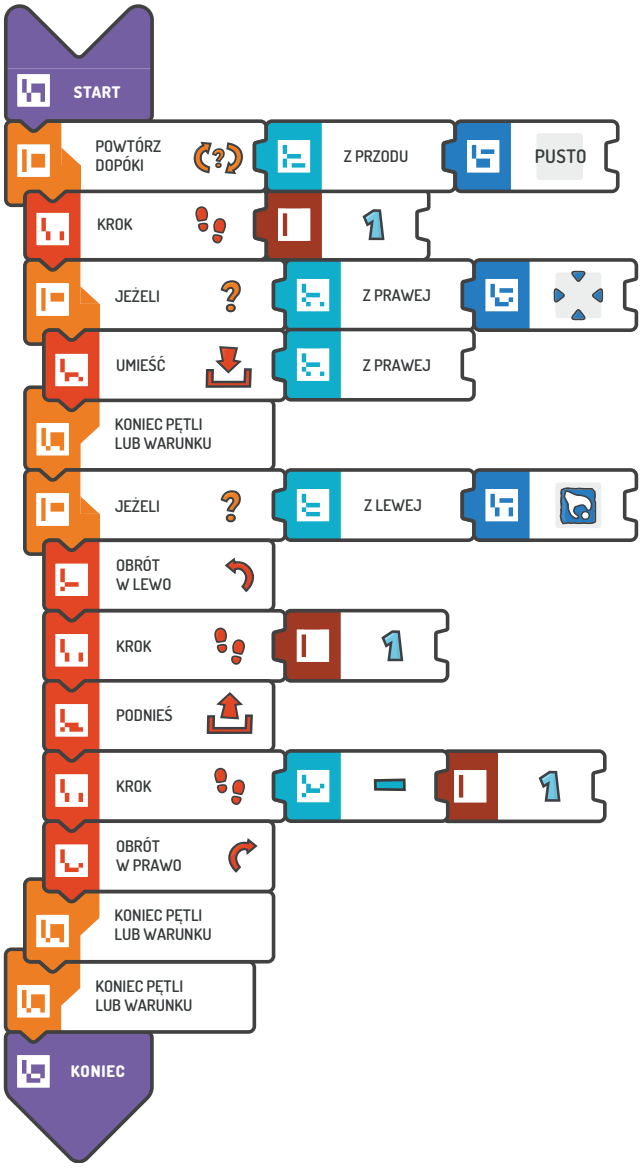
Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=3
    ZMIENNA Y=4
    POWTÓRZ (X) {
        KROK (Y)
        OBRÓT W PRAWO
    }
KONIEC
```

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

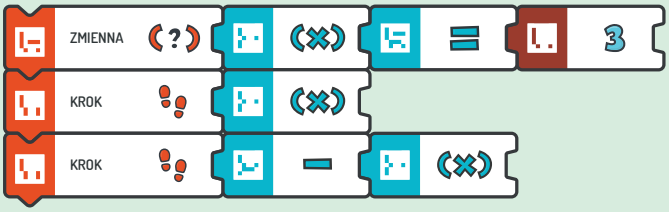
Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=1
    ZMIENNA Y=7
    POWTÓRZ (Y) {
        KROK (X)
        OBRÓT W LEWO
        KROK (X)
        OBRÓT W PRAWO
    }
KONIEC
```

WPROWADZENIE:
Działania matematyczne na zmiennych

Na zmiennych możemy również wykonywać działania matematyczne.

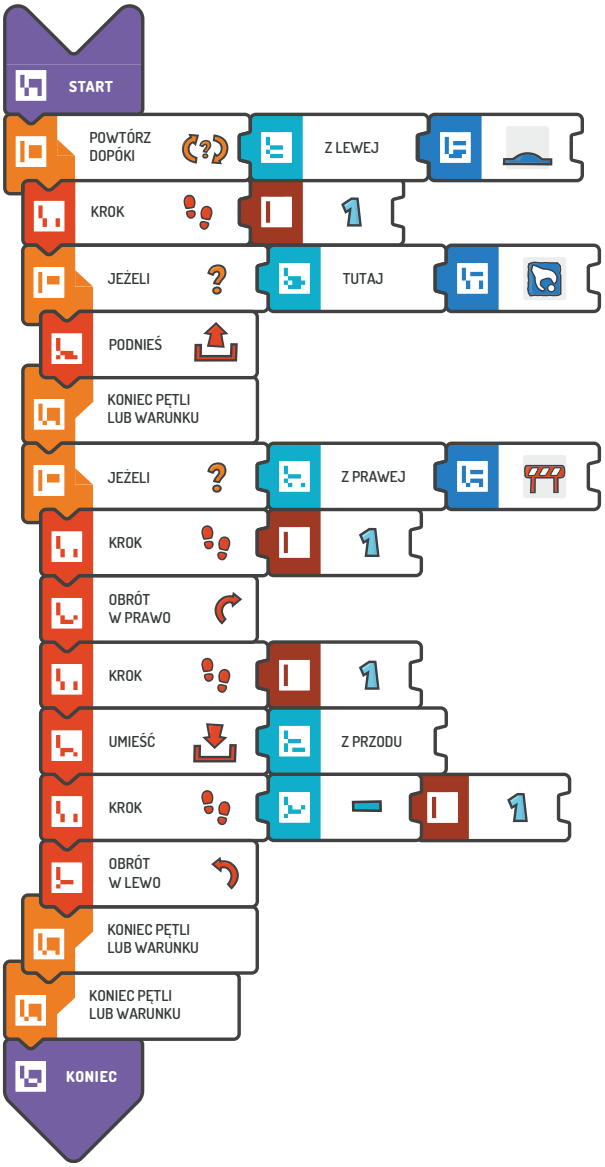
W poniższym fragmencie programua zmienimy znak na ilości kroków, które wykonać ma Scottie poprzez uieszczenie klocka **MINUS** przed zmienną **X**.



W tym zadaniu wykorzystaj koniecznie klocek z cyfrą 3 jeden raz.

Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie:



Wyzwanie **(W tym module użyj instrukcji warunkowej)**:

Przykładowe rozwiązanie (format tekstowy):

```
START
    ZMIENNA X=3
    KROK (X)
    KROK (-X)
    KROK (-X)
KONIEC
```