

2.7. Zakręt za zakrętem

DOWIESZ SIĘ

- w jaki sposób tworzy się skrypty rekurencyjne,
- jak rozwiązać problem wież Hanoi.

RYSOWANIE GWIAZD

Rysowaliście już wielokąt foremny. Wykorzystywaliście blok **powtórz ... razy**, który umożliwia wielokrotne powtarzanie tych samych poleceń. Jednak nie zawsze łatwo jest określić, ile razy powtórzyć polecenia.

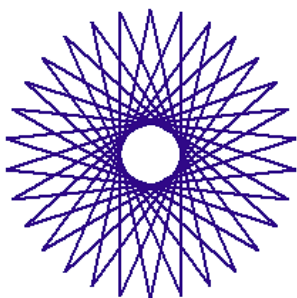
Gwiazda przedstawiona na rys. 1 powstała wskutek wielokrotnego wykonania przez duszka bloków **przesuń o 100 kroków i obróć (w prawo) o 156 stopni**. Ile razy? Dostatecznie wiele. Można pracowicie policzyć, ile potrzeba skrętów, aby uzyskać pełną gwiazdę, ale wygodniej jest zostawić tę część pracy komputerowi. Czy potrafisz zaplanować takie kąty skrętu, dla których powstaną ładne gwiazdy?

Najprostszy plan rysowania gwiazdy będzie wyglądał następująco:

- przejście naprzód o długość boku,
- skręt w prawo o wartość kąta,
- powtórzenie tych samych czynności dostatecznie wiele razy.

Zapiszesz ten plan, korzystając z możliwości tworzenia własnych bloków w Scratchu.

- Otwórz grupę **Więcej bloków**, kliknij **Stwórz blok**, nazwij go **wielo**, rozwiń **Opcje** i dodaj dwa parametry liczbowe – **bok i kąt** (rys. 2).



Rys. 1. Gwiazda o kącie skrętu 156°



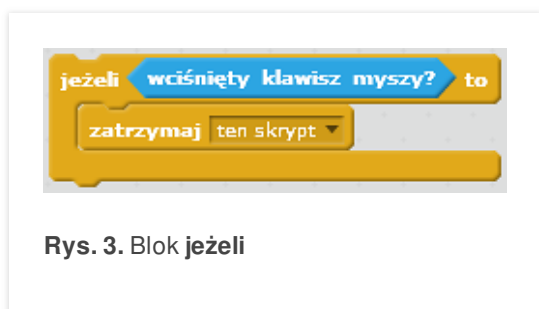
Rys. 2. Definicja bloku **wielo**

CO TO JEST REKURENCJA?

Jak to możliwe? Nie skończyliśmy definiowania procedury, a już z niej korzystamy. Spróbuj wytłumaczyć komuś, jak wykonywać jakąś czynność, wymagającą wielu powtórzeń, np. jak tańczyć tango: dwa długie, wolne kroki, dwa krótkie, szybkie kroki i tańczyć tango dalej. Taki sposób opisywania powtarzanych czynności jest możliwy również w Scratchu i nazywa się **rekurencją**. Istnieje pewne ryzyko stosowania rekurencji: jeśli nie zostanie określony warunek zakończenia działania, to będzie ona działać w nieskończoność. W przypadku tańca warunek jest prosty: należy skończyć, gdy przestanie grać muzyka. Blok **wielo** nie ma określonego warunku zatrzymania i byłoby bardzo niedobrze, gdyby nie przycisk **Zatrzymaj wszystko** – czerwona kropka na pasku nad sceną. Należy jednak unikać takich konstrukcji.

JAK ZATRZYMAĆ REKURENCJĘ?

Trzeba wstawić warunek zatrzymania. Niech duszek kończy rysowanie po naciśnięciu przycisku myszy. Do skryptu trzeba dopisać blok **jeżeli**. Jego sens jest następujący: jeśli zostanie naciśnięty lewy przycisk myszy, to należy zatrzymać wykonywanie skryptu. Gdzie umieścić ten blok? Na pewno przed wywołaniem bloku **wielo**, najlepiej na samym początku skryptu, tuż po nagłówku bloku (rys. 3).



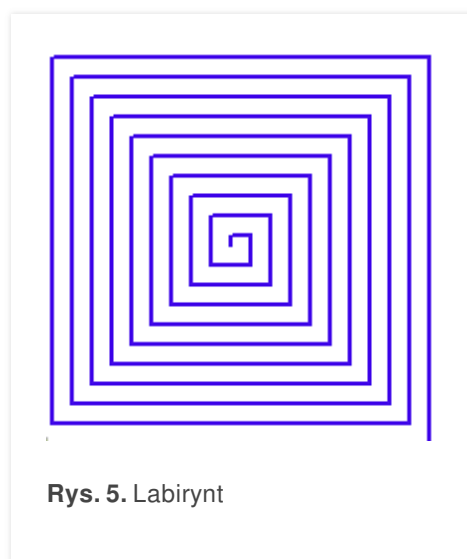
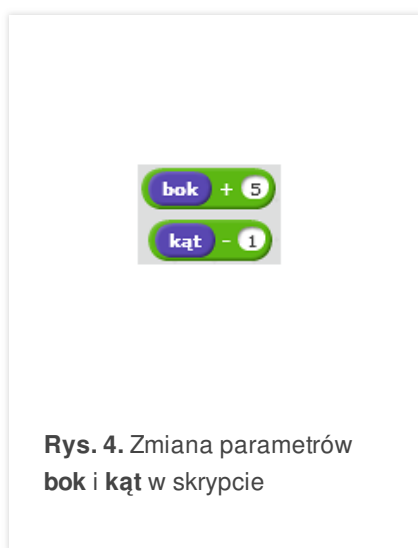
Z rekurencji będziesz korzystać wielokrotnie. Umożliwia ona rozwiązywanie trudnych zadań. Dlatego przyjrzyj się uważnie działaniu pierwszego skryptu rekurencyjnego. Pamiętaj, że trzeba zatrzymać wykonywanie skryptu, naciskając przycisk myszy.

- Narysuj kilka gwiazd, wypróbuj różne kąty skrętu. Podaj również kąt 90 stopni. Zostanie wówczas narysowany kwadrat. Zatem blok **wielo** może rysować również wielokąty.

JAK ZMIENIĆ REKURENCJĘ?

- Zmień wartość parametru w wywołaniu rekurencyjnym bloku **wielo**, np. niech **bok** zwiększa się o 5 lub **kąt** maleje o 1° (rys. 4).

Co się dzieje z rysunkiem? Jak można zatrzymać działanie rekurencji w zmodyfikowanym bloku **wielo**?



Skoro za pomocą bloku **wielo** można narysować kwadrat, to może uda się utworzyć również figurę przedstawioną na rys. 5. Ponieważ w tym wypadku długość rysowanego boku się zmienia, trzeba będzie wprowadzić pewne zmiany w bloku. Utworzysz nowy blok **wielospi** z parametrami **bok**, **kąt**. Blok ten powinien zaczynać rysowanie od najkrótszego boku, np. o długości 5. Następny bok, rysowany po skręcie, będzie o 5 dłuższy od poprzedniego i tak dalej, aż do narysowania najdłuższego boku, który będzie dotykał krawędzi. Są tu trzy elementy charakterystyczne dla budowy skryptu rekurencyjnego:

- **warunek zakończenia:** „aż do zetknięcia z krawędzią”,
- **kroki do wykonania:** narysowanie boku i wykonanie skrętu o podany kąt,
- **powtarzanie rekurencyjne:** „i tak dalej”.

Skąd jednak skrypt będzie wiedział, że ma narysować kolejny bok dłuższy od poprzedniego? To proste – blok rysuje bok, którego wartość podajesz w jego wywołaniu. Jeśli przy każdym kolejnym wywołaniu bok ma być dłuższy o 5 od poprzedniego, to należy zwiększać długość boku w wywołaniu rekurencyjnym.

- Zbuduj rekurencyjny blok **wielospi** z parametrami **bok**, **kąt**. Wybierz najpierw parametry wywołania: **bok 5**, **kąt 90°** (rys. 6).
- Jeśli chcesz się przyjrzeć kolejnym ruchom duszka, musisz zwolnić jego działanie. W tym celu dodaj blok **czekaj 0.5 s**. To polecenie zatrzymuje duszka na pół sekundy.

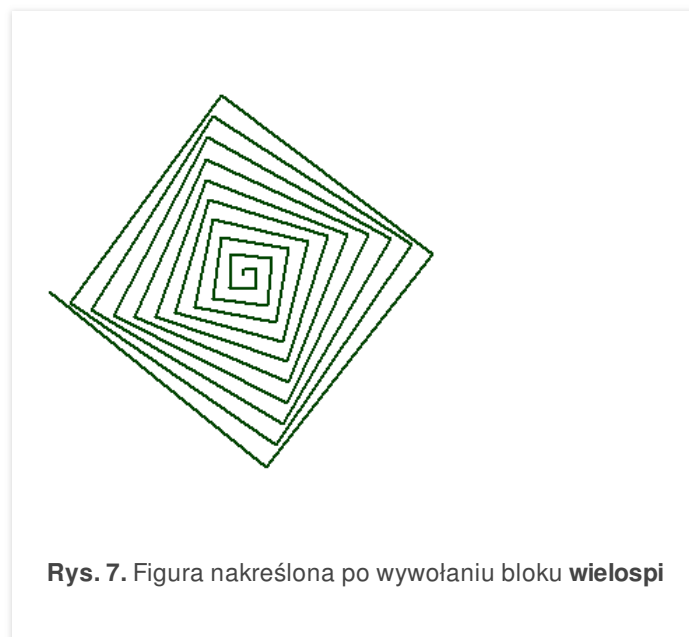
Co się stanie, jeśli blok **czekaj** przestawisz na koniec bloku za wywołanie rekurencyjne? Tym razem duszek znów rysuje bardzo szybko, ale ma kłopoty z zakończeniem działania. Czy to możliwe, że blok dodany po wywołaniu rekurencyjnym zostanie zrealizowany?

- Na końcu, za blokiem **czekaj**, dodaj blok **zmień rozmiar o 5**.

Jak teraz jest wykonywany skrypt? Jak powiększył się duszek?

- Wywołuj blok **wielospi** dla różnych kątów i oglądaj otrzymane rysunki. Często są bardzo ładne (rys. 7).

W projekcie **wielospi.sb2**, znajdującym się pod adresem scratch.mit.edu/projects/19090724, możesz obejrzeć wiele różnych figur tworzonych przez blok **wielospi**.



WIEŻE HANOI

Koronnym przykładem przydatności rekurencji jest problem znany jako wieże Hanoi. W 1883 roku francuski matematyk Édouard Lucas (czytaj: eduard luka) wymyślił i rozpowszechnił zabawkę zwaną *wieże Hanoi* (rys. 8). Na podstawie znajdują się trzy słupki, na jednym z nich umieszcza się osiem krążków o coraz mniejszych średnicach. Gra polega na przeniesieniu wszystkich krążków na inny słupek z zachowaniem następujących reguł:

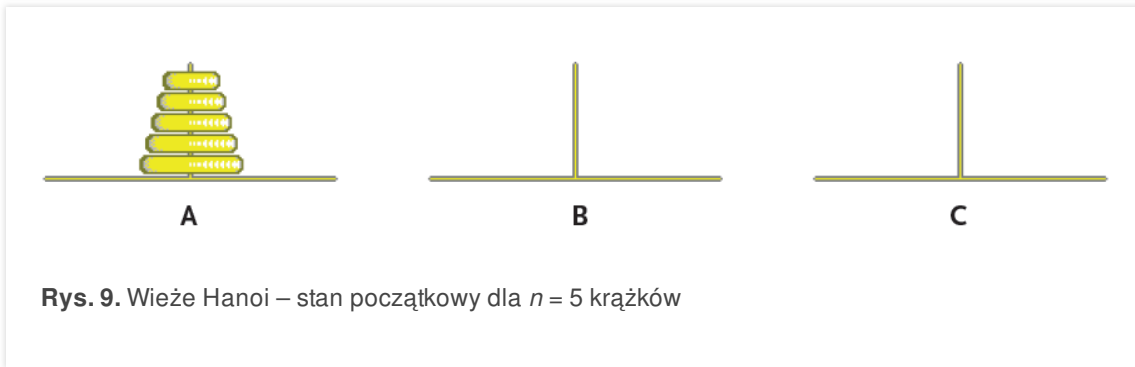
- wolno przenosić po jednym krążku,
- nie wolno położyć większego krążka na mniejszym,
- można wykorzystywać trzeci słupek jako pomocniczy.



- Wykonaj tę grę dla trzech krążków (możesz spróbować zrobić to w pamięci). Powinno ci się udać przenieść wszystkie krążki w siedmiu ruchach.

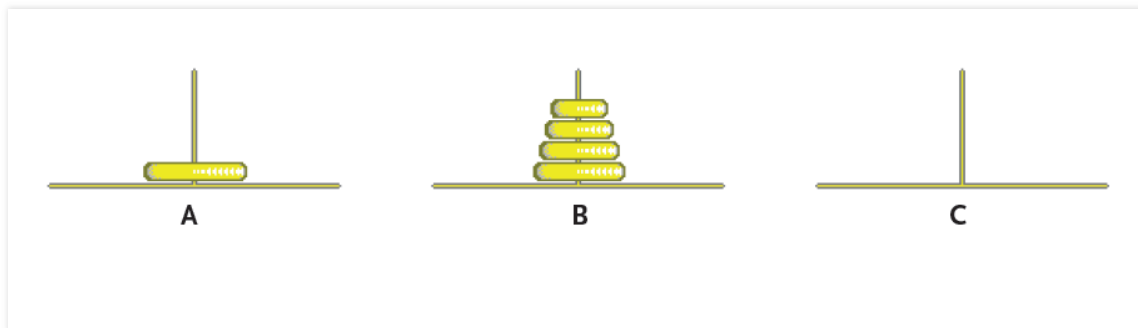
Według legendy (stworzonej prawdopodobnie przez samego Lucasa) na początku świata Brahma postawił w klasztorze w Hanoi wieżę z 64 złotych krążków. Nakazał mnichom przenosić je według podanych wyżej reguł. Gdy mnisi przeniosą wszystkie krążki, nastąpi koniec świata.

To zadanie ma dość proste rozwiązanie rekurencyjne. Jeśli założymy, że przekładanie $n - 1$ krążków jakoś się uda, to algorytm – rozwiązanie dla n krążków może wyglądać następująco.

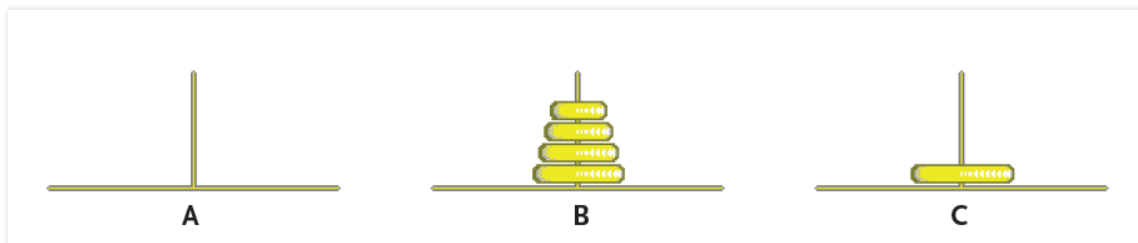


Rys. 9. Wieże Hanoi – stan początkowy dla $n = 5$ krążków

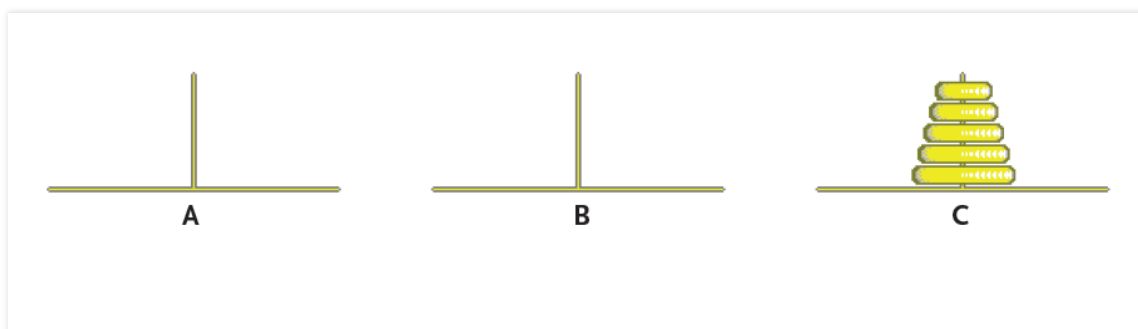
1. Jeśli nie ma już krążków do przenoszenia, to koniec gry.
2. Przenieś $n - 1$ krążków ze słupka A na B, posługując się słupkiem C.



3. Przenieś n -ty krążek ze słupka A na C.

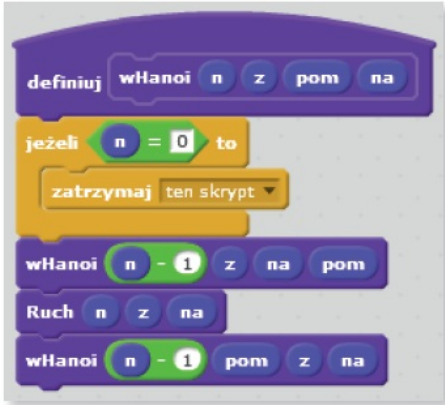


4. Przenieś $n - 1$ krążków ze słupka B na C, posługując się słupkiem A.



Skrypt bloku rekurencyjnego **wHanoi**, zbudowany według tego algorytmu, i wynik jego działania dla $n = 3$ możesz zobaczyć na rys. 10., a wypróbować po uruchomieniu pokazu **wHanoi.sb2** znajdującego się pod adresem scratch.mit.edu/projects/37051896.

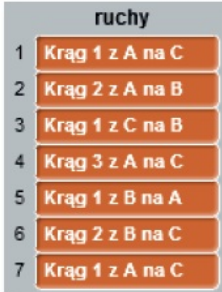
Program



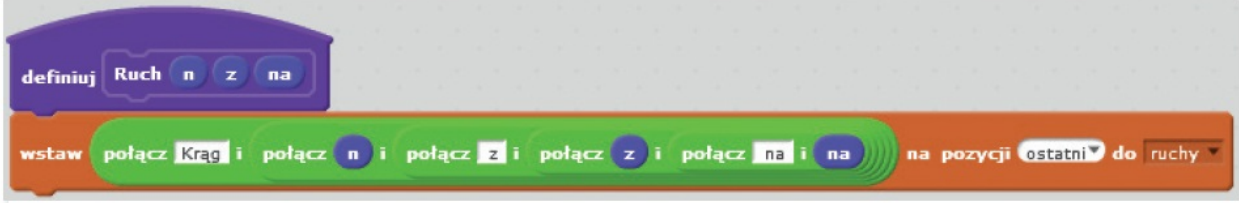
```

definiuj wHanoi n z pom na
jeżeli n = 0 to
  zatrzymaj ten skrypt
wHanoi n - 1 z na pom
Ruch n z na
wHanoi n - 1 pom z na
        
```

Wynik



ruchy	
1	Krąg 1 z A na C
2	Krąg 2 z A na B
3	Krąg 1 z C na B
4	Krąg 3 z A na C
5	Krąg 1 z B na A
6	Krąg 2 z B na C
7	Krąg 1 z A na C



```

definiuj Ruch n z na
wstaw połącz Krąg i połącz n i połącz z i połącz z i połącz na i na na pozycji ostatni do ruchy
        
```

Rys. 10. Skrypt **wHanoi** i jego rozwiązanie dla trzech krążków

Kręgi są ponumerowane od najwyższego. Ile ruchów trzeba wykonać? Dla jednego krążka to oczywiście 1 ruch, dla dwóch krążków – 3 ruchy, dla trzech – 7 (jak widać powyżej). Dla większej liczby krążków możesz to sprawdzić w projekcie. Ruchów jest dużo i nie da się już tego poprawić, optymalizując algorytm.

ZADANIA

1. Zmodyfikuj blok **wielospi**, tak aby długość boku była niezmienna, a zmieniał się kąt skrętu duszka. Bok nie powinien być zbyt długi. Dla jakich wartości kątów procedura powinna kończyć rysowanie?
2. Zmodyfikuj blok **wielospi**, tak aby jego parametrami były: **bok**, **kąt** i **skok**. Po wywołaniu rekurencyjnym **bok** powinien zwiększać się o wielkość **skoku**. Jakie ryzyko towarzyszy takiemu rozwiązaniu?
3. Policz lub odgadnij, ile razy trzeba przemieścić krążki wieży Hanoi, aby cała wieża złożona z n krążków została przeniesiona.