

## Programujemy gry multimedialne w środowisku Scratch

1. Stosujemy powtarzanie poleceń – rysujemy piramidę
2. Stosujemy instrukcję warunkową
3. Tworzymy dwupoziomową grę dla dwóch graczy
  - 3.1. Umieszczamy na scenie więcej duszków
  - 3.2. Zliczamy punkty i określamy warunki zakończenia gry
  - 3.3. Dodajemy drugi poziom gry
4. Dodajemy do programu dźwięk

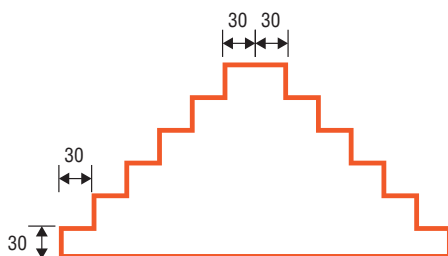


### Warto powtórzyć

1. W jaki sposób tworzymy programy w środowisku Scratch?
2. W jaki sposób programujemy powtarzające się polecenia?
3. Jak zmieniamy pozycję duszka na scenie?
4. W jaki sposób można zmienić tło sceny i postać duszka?

## 1. Stosujemy powtarzanie poleceń – rysujemy piramidę

Chcemy utworzyć program, w którym duszek narysuje piramidę pokazaną na rysunku 1. Zaczniemy od napisania programu rysującego jeden stopień piramidy. W jaki sposób można wykorzystać ten program do utworzenia całej piramidy, aby nie umieszczać wiele razy tych samych poleceń?



Rys. 1. Piramida – ćwiczenia 1-5



Rys. 2. Program rysujący jeden stopień piramidy – ćwiczenie 1.

**Uwaga**

Aby w programie Scratch 3.0 dodać do panelu poleceń polecenia z grupy **Pisak (Pen)**, należy w lewym dolnym rogu okna programu kliknąć przycisk **Dodaj**

**rozszerzenie**

i wybrać grupę **Pen (Pisak)**.



### Ćwiczenie 1. Rysujemy pierwszy stopień piramidy

1. W programie Scratch umieść w obszarze roboczym polecenia pokazane na rysunku 2.
2. Uruchom program.
3. Zapisz program w pliku pod nazwą *piramida*, korzystając z opcji **Plik/Zapisz na swoim komputerze (Zapisz jako)**.



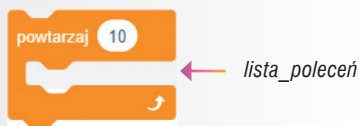
### Ćwiczenie 2. Dodajemy drugi stopień piramidy

1. Otwórz plik *piramida* zapisany w ćwiczeniu 1., korzystając z opcji **Plik/Wczytaj ze swojego komputera (Otwórz)**.
2. Dodaj polecenia rysujące drugi stopień w lewej części piramidy. Co zauważasz?
3. Zapisz plik pod tą samą nazwą.

Lewa część piramidy składa się z sześciu takich samych stopni, czyli te same polecenia trzeba by zapisać sześciokrotnie. Aby tego uniknąć, w językach programowania do zapisywania powtarzających się poleceń stosujemy **polecenia (instrukcje) powtarzania**.



**Aby w języku Scratch zapisać powtarzające się polecenia**, możemy użyć polecenia (instrukcji) powtarzania **powtarzaj (powtórz)** reprezentowanego przez element:



Polecenia, które mają być powtarzane, należy umieścić wewnątrz elementu **powtarzaj (powtórz)**. Liczbę powtórzeń określamy, zmieniając wartość w polu tekstowym polecenia (domyślnie jest w nim wpisana wartość 10).



### Ćwiczenie 3. Rysujemy lewą część piramidy – powtarzamy polecenia

1. Otwórz plik *piramida* zapisany w ćwiczeniu 2. Zmodyfikuj program, stosując polecenie **powtarzaj (powtórz)** do powtarzających się poleceń (rys. 3.).
2. Umieść dodatkowe polecenia przed poleceniem **pen down (przyłóż pisak)**: wyczyść scenę, podnieś pisak, przenieś duszka do pozycji określonej współrzędnymi (-180, -90), ustaw kierunek duszka na 90°, ustaw kolor pisaka na ciemnozielony, ustaw rozmiar pisaka na 4.
3. Zapisz plik pod tą samą nazwą. Uruchom program.

**Wskazówka:** Aby duszek nie rysował linii, gdy przesuwa się po scenie, stosujemy polecenie **pen up (podnieś pisak)**. Gdy duszek będzie we właściwej pozycji – przykładamy pisak (polecenie **pen down – przyłóż pisak**).

*Rys. 3. Przykład stosowania polecenia **powtarzaj (powtórz)** – ćwiczenie 3. (punkt 1.)*





### Ćwiczenie 4. Rysujemy prawą część piramidy

1. Otwórz plik *piramida* zapisany w ćwiczeniu 3.
2. Dodaj polecenia rysujące prawą część piramidy. Duszek ma kontynuować rysowanie od góry w prawo i w dół.
3. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Aby narysować stopień w prawej części piramidy, użyj tych samych czterech poleceń, które rysowały stopień w lewej części (tylko umieszczonych w innej kolejności) i powtórz je sześć razy.



### Ćwiczenie 5. Rysujemy podstawę piramidy

1. Otwórz plik *piramida* zapisany w ćwiczeniu 4. Uruchom program.
2. Dodaj polecenia rysujące podstawę piramidy. Długość podstawy (liczbę kroków, które musi przejść duszek) oblicz, korzystając z rysunku 1.
3. Na koniec ukryj duszka (polecenie **ukryj** z grupy **Wygląd**).
4. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Przed narysowaniem podstawy piramidy trzeba odpowiednio odwrócić duszka. Aby dobrać kierunek, sprawdź, w jakiej pozycji pozostał duszek (w jakim kierunku jest odwrócony) po wykonaniu programu zapisanego w ćwiczeniu 4.

## 2.

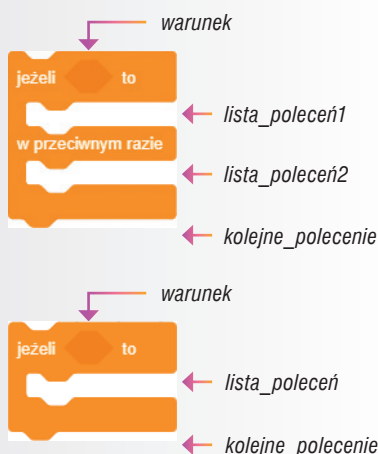
## Stosujemy instrukcję warunkową

{ Chcemy utworzyć program, w którym duszek będzie rysował różne figury zależnie od naciśnięcia wybranego klawisza, np. strzałki w lewo. Jak to zrobić w języku Scratch? }

Jeśli wynik działania programu zależy od spełnienia warunku (np. naciśnięcia klawisza strzałki w lewo), mamy do czynienia z sytuacją warunkową.



**Aby w języku Scratch zrealizować sytuację warunkową,** możemy użyć polecenia warunkowego (instrukcji warunkowej) **jeżeli** reprezentowanego przez elementy:

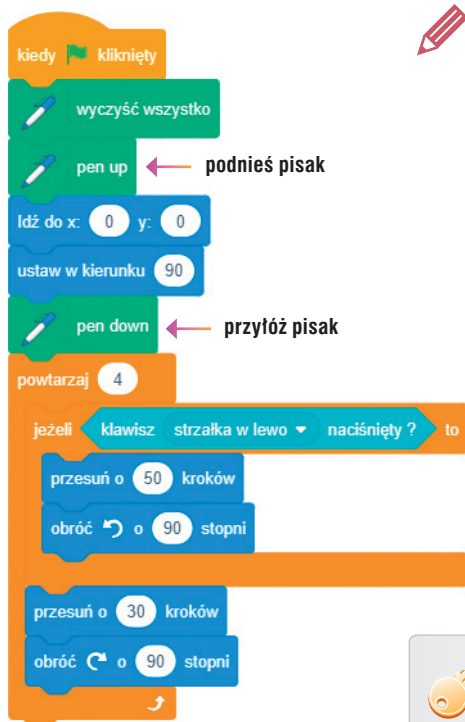


Jeśli *warunek* jest spełniony, realizowana jest *lista\_polecen1* programu umieszczona po słowie **to**.

Jeśli *warunek* nie jest spełniony, realizowana jest *lista\_polecen2* programu umieszczona po słowach **w przeciwnym razie**. Następnie realizowane jest *kolejne\_polecenie* programu.

W wersji uproszczonej – jeśli *warunek* jest spełniony, realizowana jest *lista\_polecen* programu umieszczona po słowie **to**. Następnie (niezależnie od spełnienia warunku) realizowane jest *kolejne\_polecenie* programu.

W programie pokazanym na rysunku 4. zastosowano instrukcję warunkową w wersji uproszczonej. Jeśli klawisz *strzałka w lewo* jest naciśnięty, czyli *warunek* jest spełniony, realizowane są polecenia **przesuń o 50 kroków** i **obróć (w lewo) o 90 stopni**, a następnie – polecenia **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni**. Wszystkie polecenia powtarzane są cztery razy. Jeśli klawisz *strzałka w lewo* nie jest naciśnięty, realizowane są tylko polecenia **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni**. Polecenia te powtarzane są cztery razy.



**Rys. 4.** Stosowanie instrukcji warunkowej w wersji uproszczonej – ćwiczenie 6.



**Ćwiczenie 6.** Stosujemy polecenie warunkowe w wersji uproszczonej i powtarzamy polecenia

1. W programie Scratch umieść w obszarze roboczym polecenia pokazane na rysunku 4.
2. Zapisz program w pliku pod nazwą *warunek 1*.
3. Uruchom program. Odpowiedz na pytanie: *Jaką figurę narysował duszek?*
4. Naciśnij i przytrzymaj klawisz strzałki w lewo i uruchom ponownie program. Odpowiedz na pytanie: *Jaką figurę teraz narysował duszek?*



**Aby wybrane polecenia były wykonywane nieskończenie wiele razy**, można zastosować polecenie **zawsze** z grupy **Kontrola**.

**Aby zakończyć działanie programu**,

należy kliknąć przycisk .



**Ćwiczenie 7.** Stosujemy polecenie warunkowe i wielokrotnie powtarzamy polecenia

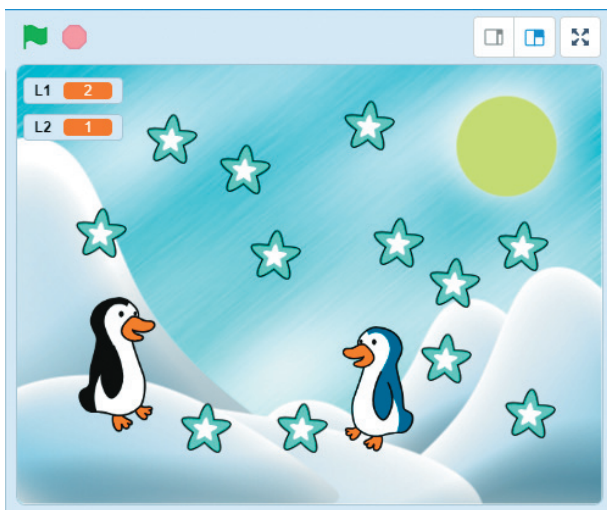
1. Zmodyfikuj program zapisany w ćwiczeniu 6., aby wykonanie poleceń **przesuń o 30 kroków** i **obróć (w prawo) o 90 stopni** zależało od naciśnięcia klawisza strzałki w prawo.
2. Zastąp polecenie **powtarzaj (powtórz)** poleceniem **zawsze**.
3. Zapisz program w pliku pod nazwą *warunek 2*.
4. Uruchom program. Naciskaj i puszczaj (np. naprzemiennie) klawisze strzałek w prawo i w lewo. Obserwuj efekty działania programu. Odpowiedz na pytanie: *Czym różni się rysowanie figur w programach warunek 1 i warunek 2?*

### 3. Tworzymy dwupoziomową grę dla dwóch graczy

Chcemy utworzyć grę, w której każdy z dwóch graczy, używając klawiszy, będzie sterował swoim duszkiem-pingwinem. Zadaniem duszków-pingwinów jest „łapanie” duszków-gwiazdek, które losowo poruszają się po scenie. Gra kończy się, gdy znikną wszystkie „złapane” gwiazdki, a wygrywa ten gracz, który złapie ich więcej. W jaki sposób utworzyć taką grę w języku Scratch?

Ruch duszków-pingwinów ma zależeć od naciśniętego klawisza, dlatego w programie zastosujemy instrukcje warunkowe sprawdzające, jaki klawisz nacisnął gracz.

Sytuacja warunkowa występuje również wtedy, gdy duszek-pingwin i duszek-gwiazdka dotkną się. W tym przypadku również użyjemy instrukcji warunkowej. Gdy warunek będzie spełniony, gwiazdkę ukryjemy i przyznamy punkty temu duszkowi-pingwinowi, który „złapał” duszka-gwiazdkę. Zatem do dzieła!



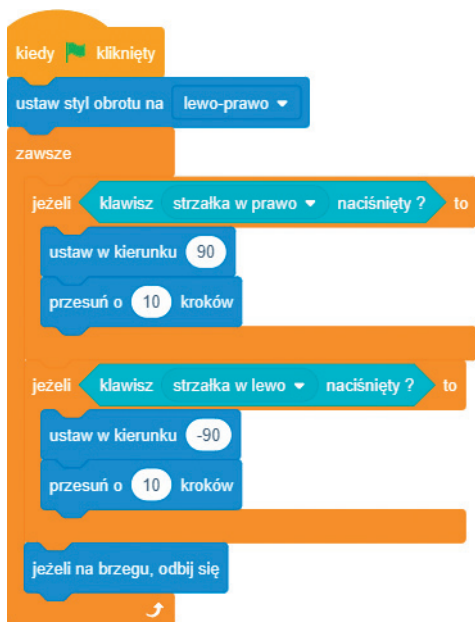
Rys. 5. Przykładowa scena gry – ćwiczenia 8-13

#### 3.1. Umieszczamy duszki na scenie



**Ćwiczenie 8.** Umieszczamy jednego duszka i sterujemy nim w czterech kierunkach

1. Utwórz nowy program. Zmień wygląd sceny na krajobraz zimowy. Tłó wybierz z Biblioteki teł.
2. Usuń duszka-kota. Z Biblioteki duszków wybierz duszka-pingwina. Zmień nazwę duszka na *Niebieski*. Usuń niepotrzebne kostiumy.
3. Utwórz skrypt dla duszka-pingwina, aby poruszał się po scenie w prawo, w lewo, w górę i w dół, zależnie od naciśniętych klawiszy strzałek. Wszystkie polecenia **jeżeli** powtarzaj nieskończenie wiele razy (zastosuj polecenie **zawsze**). Na rysunku 6. pokazano początkową część skryptu.
4. Zadbaj, aby duszek nie chodził do góry nogami.
5. Zapisz program w pliku pod nazwą *pingwiny*. Uruchom program i sprawdź jego działanie.



Rys. 6. Początkowa część skryptu dla duszka-pingwina – ćwiczenie 8.

### Wskazówki (ćw. 8.):

- Umieść najpierw jedno polecenie **jeżeli** z poleceniami, które mają się powtarzać. Z menu kontekstowego elementu **jeżeli** wybierz polecenie **duplikuj** i dodaj kolejne polecenia warunkowe umożliwiające sterowanie duszkiem w pozostałych kierunkach. W zduplikowanych poleceniach wystarczy zmienić odpowiednio warunki i kierunki.
- Po wszystkich poleceniach **jeżeli** dodaj polecenie **jeżeli na brzegu, odbij się**, aby duszek nie „uciekał” poza scenę.
- Aby duszek nie chodził do góry nogami, możesz na początku skryptu duszka dodać polecenie **ustal styl obrotu na lewo-prawo**.
- Nazwę duszka zmień we właściwościach duszka i w polu tekstowym widocznym po wybraniu karty **Kostiumy**.



**Aby zdublować duszka razem z ułożonym dla niego skryptem**, należy z menu kontekstowego duszka umieszczonego w obszarze Duszki (Duszki) wybrać polecenie **duplikuj**.



**Ćwiczenie 9.** Dodajemy drugiego duszka-pingwina, duplikując pierwszego duszka wraz z poleceniami

1. Otwórz plik *pingwiny* zapisany w ćwiczeniu 8. Dodaj drugiego duszka-pingwina, duplikując pierwszego duszka wraz ze skryptem.
2. Zmień kolor drugiego duszka-pingwina na czarny. Skorzystaj z edytora graficznego programu Scratch (zakładka **Kostiumy**). Nazwij duszka **Czarny**.
3. Ustal inne klawisze do sterowania drugim duszkiem, np. litery: *a* (w lewo), *d* (w prawo), *w* (w górę), *s* (w dół).
4. Zapisz plik pod tą samą nazwą.

W programie Scratch współrzędna  $x$  przyjmuje wartości od -240 do 240, a  $y$  od -180 do 180. Środek sceny ma współrzędne  $x = 0$  i  $y = 0$ , co zapisujemy (0,0).



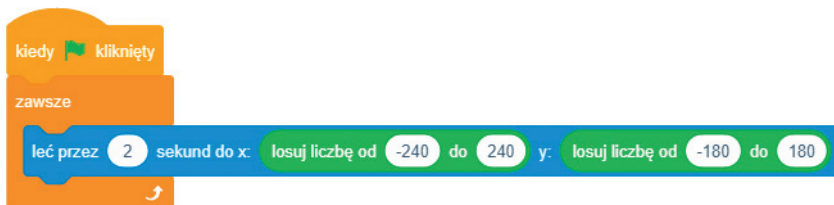
**Aby duszek poruszał się losowo po całej scenie**, należy w polach tekstowych polecenia **leć** wstawić element **losuj liczbę (losuj)** z grupy **Wyrażenia** i wpisać odpowiednio wartości dla  $x$  od -240 do 240, a dla  $y$  od -180 do 180.



### Ćwiczenie 10. Dodajemy duszka-gwiazdkę poruszającego się losowo po scenie

1. Otwórz plik *pingwiny* zapisany w ćwiczeniu 9. Umieść na scenie duszka-gwiazdkę, który ma się losowo przemieszczać po całej scenie. Skrypt odpowiadający za ruch duszka-gwiazdki pokazano na rysunku 7.
2. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Wybierz gwiazdkę z Biblioteki duszków i zmień kolor z żółtego na niebieski.



Rys. 7. Skrypt dla duszka-gwiazdki – ćwiczenie 10.

## 3.2. Zliczamy punkty i określamy warunki zakończenia gry

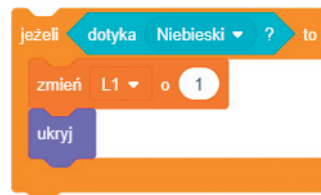
{ Chcemy zliczać punkty, gdy duszek-pingwin i duszek-gwiazdka dotkną się, a także skończyć grę po uzyskaniu przez gracza określonej liczby punktów. Jak zliczać punkty oddzielnie dla każdego duszka i określić warunki zakończenia gry? }

Aby zliczać punkty w grze, utworzymy dwie zmienne *L1* i *L2*, które będziemy nazywać licznikami. W zmiennych *L1* i *L2* będziemy zapamiętywać punkty zdobyte przez duszki-pingwiny.



### Ćwiczenie 11. Zliczamy punkty oddzielnie dla każdego duszka-pingwina

1. Otwórz plik *pingwiny* zapisany w ćwiczeniu 10. Utwórz dwie zmienne o nazwach *L1* i *L2*.
2. W skrypcie każdego duszka-pingwina dodaj przed poleceniem **zawsze** zerowanie jego licznika: *L1* dla niebieskiego pingwina, *L2* dla czarnego.
3. Do skryptu duszka-gwiazdki dodaj zliczanie punktów oddzielnie dla każdego duszka-pingwina. Na początku skryptu duszka-gwiazdki umieść polecenie **pokaż** z grupy **Wygląd**. Następnie w poleceniu **zawsze** po poleceniu **leć** umieść dwie instrukcje warunkowe. Jeśli duszek-pingwin i duszek-gwiazdka dotkną się – wartość odpowiedniej zmiennej ma zwiększyć się o 1 i gwiazdka ma zniknąć (zastosuj polecenie **ukryj** z grupy **Wygląd**). Na rysunku 8. pokazano instrukcję warunkową zliczającą punkty dla niebieskiego duszka-pingwina.
4. Zapisz plik pod tą samą nazwą. Uruchom program i sprawdź, czy gwiazdka znika po dotknięciu się z wybranym duszkiem-pingwinem i czy dolicza się jeden punkt.




Rys. 8. Zliczanie punktów dla niebieskiego duszka-pingwina – ćwiczenie 11.



### Wskazówki (ćw. 11.):

- Aby utworzyć nową zmienną, kliknij przycisk **Utwórz zmienną** w grupie **Zmienne (Dane)**.

- Aby wyzerować licznik, zastosuj polecenie **ustaw**: . Na liście wyboru wybierz odpowiednią zmienną.
- Aby zliczać punkty, gdy duszek-pingwin i duszek-gwiazdka dotkną się, do polecenia

**jeżeli** w polu warunku wstaw element  (z grupy **Czujniki**)

i wybierz danego duszka. Następnie dodaj zliczanie punktów (rys. 8.).

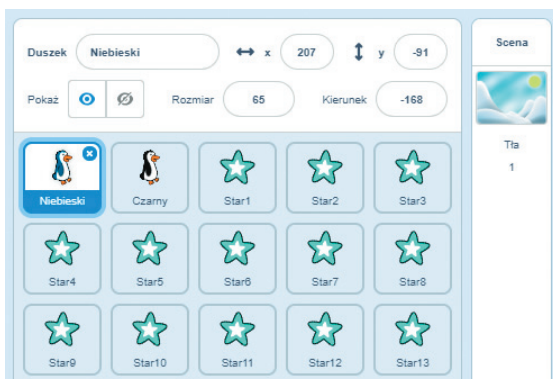


Jeśli skrypt dla duszka-gwiazdki działa poprawnie, możemy zduplikować tego duszka razem z jego skrypcem.





### Ćwiczenie 12. Zwiększamy liczbę duszków-gwiazdek

- Otwórz plik *pingwiny* zapisany w ćwiczeniu 11. Zduplikuj czternaście razy duszka-gwiazdkę wraz ze skrypcem (rys. 9.).
- Zapisz plik pod tą samą nazwą.





**Rys. 9.** Obszar *Duszek (Duszki)* – ćwiczenie 12.


Nasza gra zakończy się, gdy zostaną zebrane wszystkie duszki-gwiazdki. Aby zaprogramować zakończenie gry, zastosujemy polecenie warunkowe.

W instrukcji warunkowej języka Scratch warunkiem może być wyrażenie logiczne reprezentowane przez elementy z grupy **Wyrażenia**, np. , ,

, .

W polach tekstowych elementów można umieszczać m.in.: nazwy zmiennych, liczby, wyrażenia. Warunek może być prosty, np. , ,

 lub złożony, np. ,

.






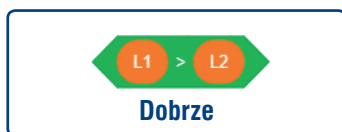
### Ćwiczenie 13. Dodajemy warunek zakończenia gry

1. Otwórz plik *pingwiny* zapisany w ćwiczeniu 12. Dodaj warunek zakończenia gry. Gra ma się zakończyć, gdy gracze „złapią” wszystkie gwiazdki, czyli w sumie uzyskają 15 punktów. Wygrywa ten, który złapie więcej gwiazdek.
2. W każdym skrypcie duszka-pingwina dodaj po wszystkich poleceniach warunkowych kolejne polecenie **jeżeli** z warunkiem złożonym ze spójnikiem „i”. Na rysunku 10. pokazano polecenie dla niebieskiego duszka-pingwina. Zmień odpowiednio warunek dla drugiego duszka-pingwina. Jeśli warunek jest spełniony, ma wyświetlić się odpowiedni komunikat i gra ma się zakończyć (zastosuj polecenie **zatrzymaj wszystko**).
3. Zapisz plik pod tą samą nazwą.

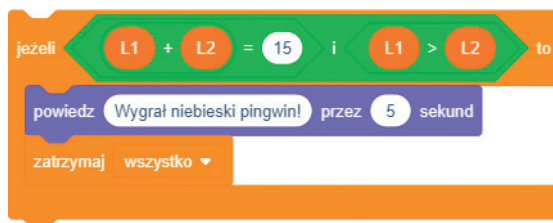
#### Wskazówki:

- Jeśli tworzysz warunek w poleceniu **jeżeli**, nie wpisuj nazwy zmiennej *L1* czy *L2* z klawiatury, tylko przeciągaj element np.  do odpowiedniego pola tekstowego

elementu .



- Polecenie **zatrzymaj wszystko** zatrzymuje wszystkie skrypty dla wszystkich duszków.



**Rys. 10.** Warunek zakończenia gry w skrypcie niebieskiego duszka-pingwina – ćwiczenie 13.

### 3.3. Dodajemy drugi poziom gry

{ Chcemy w naszej grze przechodzić na kolejne poziomy.  
W jaki sposób je utworzyć i określić warunki przejścia na wyższy poziom? }

Do kolejnych poziomów będziemy przechodzić, wykorzystując zmianę tła sceny. W języku Scratch możemy zmienić tło sceny podobnie jak kostiumy duszków. Aby kontrolować zmianę tła, możemy napisać skrypt dla sceny, w którym umieścimy instrukcję warunkową z warunkiem przejścia na wyższy poziom. Możemy też utworzyć warunek powrotu na niższy poziom.

Na wyższym poziomie można wprowadzić różne utrudnienia i ograniczenia, np. dodać duszki, które zwiększają lub zmniejszają liczbę punktów, zwiększyć szybkość poruszania się duszków, kontrolować czas gry itp.



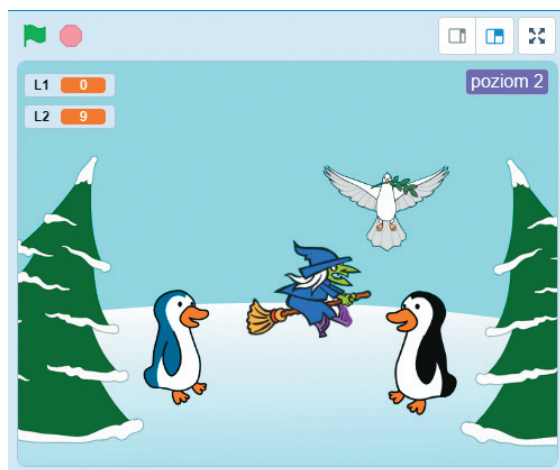
Aby dodać nową scenę, należy kliknąć przycisk **Wybierz tło (Nowe tło)** w obszarze scen.



Aby utworzyć skrypt dla sceny, należy:

- sprawdzić, czy scena jest aktywna – na ikonie sceny powinno być widoczne niebieskie obramowanie,
- przejść do karty **Skrypty** i w obszarze tworzenia programu ułożyć skrypt dla sceny.

Nie wszystkie polecenia są dostępne dla skryptu sceny. Na przykład nie ma możliwości zastosowania poleceń z grupy **Ruch**, a z grupy **Pisak** aktywne jest tylko polecenie **Wyczyść**.



**Rys. 11.** Przykładowy widok drugiego poziomu gry – ćwiczenie 14.

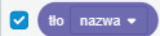
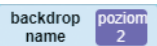


**Rys. 12.** Skrypt dla sceny – ćwiczenie 14.



### Ćwiczenie 14. Dodajemy drugi poziom gry

1. Otwórz plik *pingwiny* zapisany w ćwiczeniu 13. Dodaj do gry przejście na drugi poziom. Powinno ono nastąpić, gdy graczę zdobędą łącznie 15 punktów.
2. Dodaj nowe tło. Zmień nazwy tła na: *poziom 1* i *poziom 2* oraz usuń niepotrzebne tła w zakładce **Tła**. Dodaj wyświetlanie nazw tła (poziomów) na scenie (rys. 11.).
3. Napisz skrypt dla sceny. Umieść w nim warunki przejścia na kolejny poziom, zależnie od liczby punktów (rys. 12.).
4. Zapisz program w pliku pod nazwą *pingwiny dwa poziomy*.

Aby zmienić nazwę tła, należy wybrać zakładkę **Tła** i wpisać nową nazwę w polu tekstowym nad polem edytora graficznego. Aby nazwy tła (poziomów) były widoczne na scenie, należy kliknąć pole wyboru przy elemencie  w grupie **Wygląd**. Domyślnie wyświetlany jest **normalny widok (odczyt)** 

(normalny odczyt) . Aby zmienić go na większy, należy kliknąć prawym przyciskiem myszy element z nazwą tła w normalnym widoku (odczytanie) i wybrać polecenie **powiększony widok (duży odczyt)** poziom 2 .

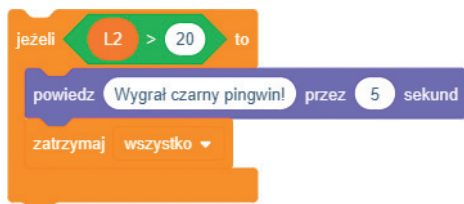


### Ćwiczenie 15. Określamy warunki przejścia na drugi poziom gry

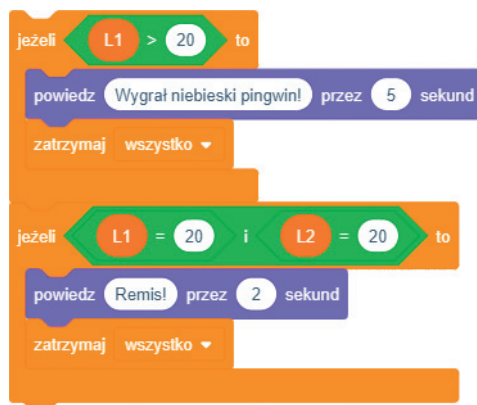
1. Otwórz plik *pingwiny dwa poziomy* zapisany w ćwiczeniu 14. Dodaj dwa nowe duszki, np. duszka-gołębia i duszka-czarownicę.
2. Napisz skrypty dla duszka-gołębia i duszka-czarownicy. Nowe duszki nie powinny być widoczne na *poziomie1*, tylko na *poziomie2*. Mają poruszać się losowo po całej scenie, tak jak duszki-gwiazdki. Gdy duszek-pingwin i duszek-gołąb dotkną się, licznik danego duszka-pingwina powinien zwiększyć się o 1, a gdy duszek-pingwin dotknie się z duszkiem-czarownicą – zmniejszyć o 1 (rys. 13.).
3. Gra kończy się, gdy jeden z duszków zdobędzie ponad 20 punktów lub obydwa zdobędą po 20 punktów (w przypadku remisu). Zmodyfikuj skrypty dla duszków pingwinów. W skrypcie dla czarnego duszka-pingwina zamień polecenie **jeżeli** z warunkiem zakończenia gry na pokazane na rysunku 14a. W skrypcie dla niebieskiego duszka-pingwina zamień polecenie **jeżeli** z warunkiem zakończenia gry na polecenia pokazane na rysunku 14b.
4. Zapisz plik pod tą samą nazwą.



Rys. 13. Skrypt dla duszka-czarownicy – ćwiczenie 15.



Rys. 14a. Zmodyfikowane polecenie warunkowe dla czarnego duszka-pingwina – ćwiczenie 15.



Rys. 14b. Zmodyfikowane polecenia warunkowe dla niebieskiego duszka-pingwina – ćwiczenie 15.

## 4. Dodajemy dźwięk do programu

W środowisku programowania Scratch, podobnie jak w Balcie, można dodać do programu dźwięki. Można nagrać własny dźwięk i zapisać go w pliku lub skorzystać z gotowych nagrań.



Aby dołączyć dźwięk do programu, możemy wstawić do wybranego skryptu

polecenie **zagraj dźwięk** z grupy **Dźwięk**, np.

zagraj dźwięk bird ▾



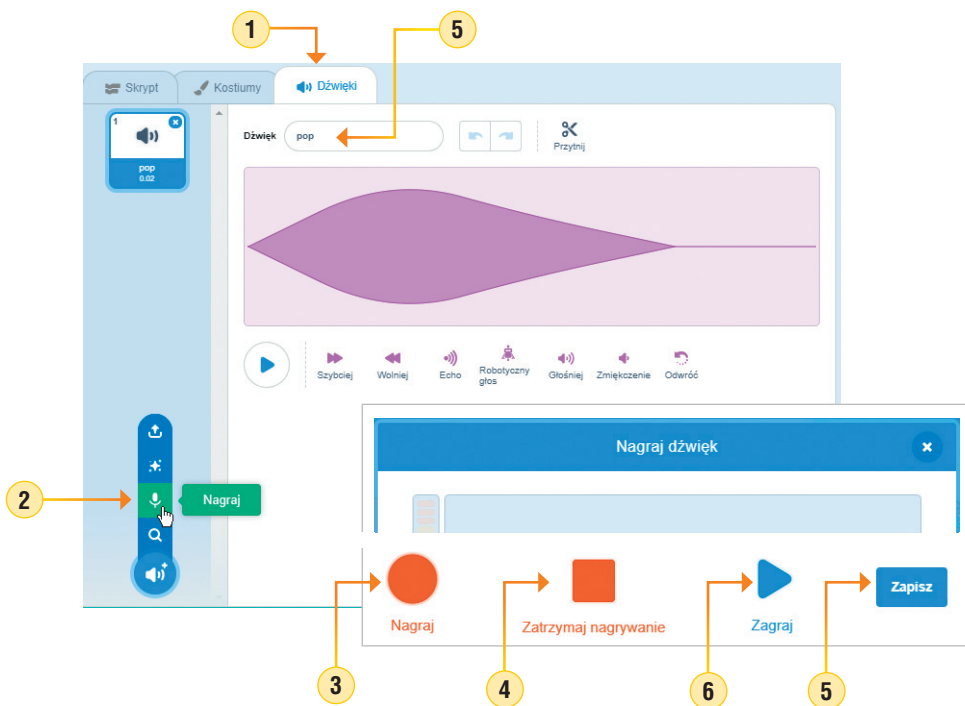
### Ćwiczenie 16. Dodajemy dźwięk do duszka-ptaka

1. Otwórz plik *pingwiny dwa poziomy* zapisany w ćwiczeniu 15.
2. Dodaj do skryptu duszka-ptaka dźwięk, który będzie słyszał, gdy duszek-ptak i duszek-pingwin dotkną się.
3. Zapisz plik pod tą samą nazwą.



### Przykład 1. Nagranie pliku dźwiękowego

1. Kliknij zakładkę **Dźwięki**.
2. Kliknij ikonę z mikrofonem.
3. Kliknij przycisk **Nagraj** i zacznij mówić do mikrofonu.
4. Kliknij przycisk **Zatrzymaj nagrywanie (Zatrzymaj)**.
5. Zapisz nagranie – możesz pozostawić domyślną nazwę nagrania lub ją zmienić.
6. Kliknij przycisk **Zagraj (Odtwórz)**, aby odtworzyć nagranie.





## Ćwiczenie 17. Dodajemy własne nagrania do programu

1. Otwórz plik *pingwiny dwa poziomy* zapisany w ćwiczeniu 16. Na początku skryptu niebieskiego duszka-pingwina dodaj własne nagranie, w którym opowiesz, na czym polega gra.
2. Zapisz plik pod tą samą nazwą.

**Wskazówka:** Utworzone nagranie wybierz z listy wyboru polecenia **zagraj dźwięk**.



## Warto zapamiętać

- Aby zapisać powtarzające się polecenia, w językach programowania stosujemy instrukcje powtarzania. W języku Scratch przykładem takiej instrukcji jest instrukcja **powtórz**.
- W języku Scratch można stosować zmienne. Zmienną należy najpierw utworzyć.
- W języku Scratch sytuację warunkową realizujemy, stosując instrukcję warunkową, która w wersji uproszczonej ma postać: **jeżeli warunek to lista\_poleceń**.
- *Warunkiem* w poleceniu **jeżeli** może być wyrażenie logiczne (np. *punkty >10*, *licznik = 20*) lub zdarzenie z grupy **Czujniki** (np. **klawisz ... naciśnięty?**, **dotyka ... ?**, **dotyka koloru ...?**).
- W języku Scratch możemy zaprogramować sterowanie duszkiem zależnie od naciśniętego klawisza.
- Polecenia warunkowe możemy zastosować na przykład do określania warunków zakończenia gry lub przechodzenia na wyższy poziom gry.
- Dla sceny w języku Scratch możemy napisać skrypt. Tło sceny możemy zmienić w podobny sposób, jak kostiumy duszków.



## Pytania

1. W jaki sposób w języku Scratch zapisujemy powtarzające się polecenia?
2. Wyjaśnij na przykładzie, w jaki sposób realizuje się sytuację warunkową w języku Scratch.
3. Podaj przykłady wyrażeń i zdarzeń, które mogą być warunkami w poleceniu **jeżeli**.
4. Jak utworzyć zmienną w języku Scratch? Omów na przykładzie.
5. W jaki sposób można określać warunki zakończenia gry? Podaj dwa przykłady.
6. W jaki sposób można utworzyć grę dla dwóch graczy? Omów na przykładzie.
7. Jak zmienić tło sceny?
8. Do czego można wykorzystać możliwość zmiany tła sceny?



## Zadania

1. Zmodyfikuj program *piramida* zapisany w ćwiczeniu 5., aby duszek narysował piramidę „do góry nogami” (czyli podstawą zwróconą ku górze). Zapisz program w pliku pod nazwą *piramida odwrotnie*.
2. Utwórz program, w którym duszek narysuje schodki prowadzące od lewego górnego rogu sceny do dolnego prawego rogu. Ustal szerokość stopnia na 25 kroków, a wysokość na 20. Zmień tło sceny. Ustal kolor pisaka na czerwony, a jego grubość na 2. Zapisz program w pliku pod nazwą *schody*.



Rys. 15. Skrypt – zadanie 3.

- Umieść program pokazany na rysunku 15. w obszarze tworzenia programu i uruchom go. Wyjaśnij, na czym polega działanie tego programu. Odpowiedz na pytanie: *Jakie polecenia są realizowane, gdy naciśnięty jest klawisz z literą d, a jakie – gdy naciśnięty jest klawisz z literą w?* Dodaj jeszcze dwa polecenia warunkowe umożliwiające ruch duszka w pozostałych dwóch kierunkach. Dobierz samodzielnie klawisze. Zapisz program w pliku pod nazwą *rysowanie duszkiem*.
- Utwórz grę, w której wezmą udział dwa duszki. Wybierz samodzielnie tło i postacie duszków (możesz również zmodyfikować istniejące postacie lub narysować nowe). Jeden z duszków ma poruszać się losowo po całej scenie. Drugim duszkiem będziesz sterować (w lewo, w prawo, w górę, w dół), naciskając wybrane klawisze. Jeśli duszki się dotkną, otrzymujesz 10 punktów. Gdy zdobędziesz 100 punktów, wyświetl komunikat „Brawo!” i zatrzymaj wszystkie skrypty – gra ma się zakończyć. Zapisz program w pliku pod nazwą określającą tematykę gry.

**Wskazówka:** Własne duszki lub tła możesz narysować w edytorze graficznym programu Scratch. Możesz je również utworzyć w edytorze graficznym Paint, zapisać w plikach z rozszerzeniem *png*, a następnie wczytać z plików do gry.

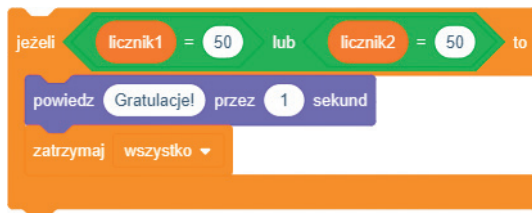
- Otwórz plik zapisany w zadaniu 4. Dodaj trzeciego duszka, który ma poruszać się losowo po całej scenie. Jeśli nowy duszek i duszek sterowany klawiszami dotkną się, otrzymujesz -10 punktów. W skrypcie dla nowego duszka wprowadź drugi warunek zakończenia gry. Gdy otrzymasz -50 punktów, wyświetl komunikat „Nie udało się” i zatrzymaj wszystkie skrypty – gra ma się również zakończyć. Zapisz plik pod tą samą nazwą.
- Utwórz grę dla dwóch graczy z udziałem trzech duszków. Wybierz samodzielnie tło i postacie duszków (możesz również zmodyfikować istniejące postacie lub narysować nowe). Jeden z duszków ma poruszać się losowo po całej scenie. Dwoma pozostałymi duszkami będą sterować gracze (w lewo, w prawo, w górę, w dół), naciskając wybrane klawisze. Jeśli któryś z duszków sterowanych klawiszami dotknie się z duszkiem poruszającym się losowo, gracz otrzymuje 2 punkty. Dla każdego z duszków utwórz oddzielny licznik, tworząc zmienne: *licznik1*, *licznik2*. Gdy jeden z duszków zdobędzie ponad 20 punktów lub obydwa zdobędą po 20 punktów, wyświetl odpowiednie komunikaty i zatrzymaj wszystkie skrypty – gra ma się zakończyć. Zapisz program w pliku pod nazwą określającą tematykę gry.

**Wskazówka:** Na początku skryptu dla duszka losowo przemieszczającego się po scenie wyzeruj obydwie zmienne. Umieść w tym skrypcie trzy polecenia **jeżeli**:

- dwa polecenia realizujące sytuację związaną z dotykiem się duszków, w jednym zmieniamy *licznik1* o 2 punkty, a w drugim – *licznik2* o 2 punkty,
- polecenie realizujące warunek zakończenia gry (rys. 16.), w którym jako warunku

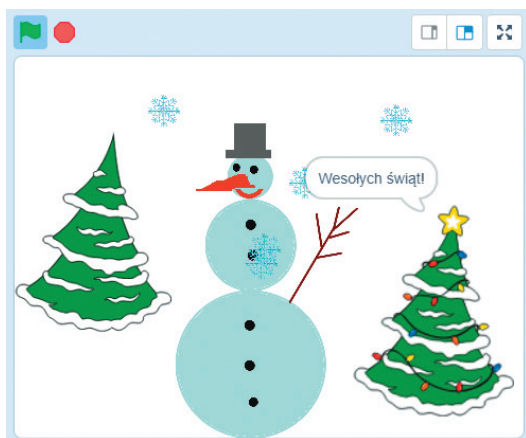
użyj elementu:





**Rys. 16.** Warunek zakończenia gry – zadanie 6.

7. Korzystając z możliwości zmiany tła, utwórz animowaną scenę pokazującą kolejne etapy lepienia bałwana. Dodaj pięć gwiazdek losowo poruszających się po scenie i duszki-choinki (rys. 17.). Dla jednego duszka-choinki utwórz skrypt wyświetlający świąteczne życzenia. Zapisz program w pliku pod nazwą *bałwanek*.

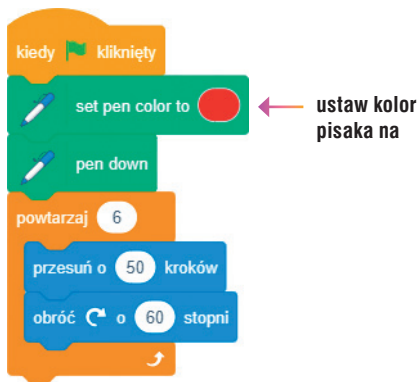


**Rys. 17.** Przykładowy efekt wykonania programu – zadanie 7.

### Dla zainteresowanych

8. Na rysunku 18a pokazano skrypt rysujący sześciokąt o boku 50 kroków. Utwórz program rysujący kompozycję z sześciokątów foremnych (rys. 18b). Zapisz program w pliku pod nazwą *kompozycja*.

**Wskazówka:** Figura foremna ma wszystkie boki i kąty równe.



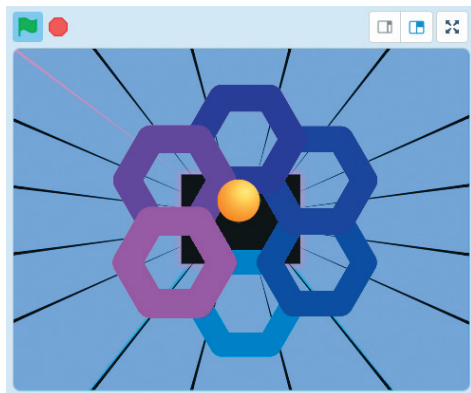
**Rys. 18a.** Skrypt rysujący sześciokąt foremny – zadanie 8.



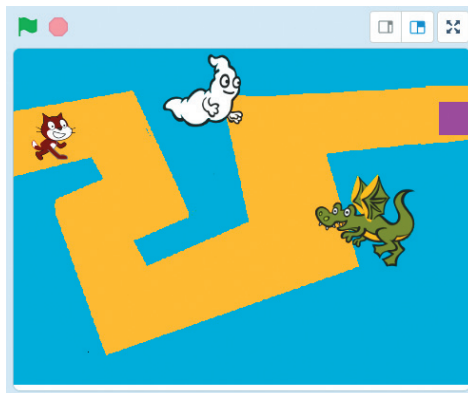
**Rys. 18b.** Efekt wykonania programu – zadanie 8.



9. Zmodyfikuj program zapisany w zadaniu 8., aby pisak wielokrotnie zmieniał kolor i grubość. Po scenie ma losowo poruszać się piłka, która po zetknięciu z sześciokątem o wybranym kolorze wyda dźwięk (kolor wybierz samodzielnie). Zmień tło na wybrane z Biblioteki tef. Zapisz program w pliku pod nazwą *animowana kompozycja*.



**Rys. 19.** Przykładowy efekt wykonania programu – zadanie 9.



**Rys. 20.** Przykładowy efekt wykonania programu – zadanie 10.

10. Narysuj ścieżkę dla duszka-kota, używając wybranych kolorów do pokolorowania ścieżki i tła (patrz wzór na rysunku 20.). Na końcu ścieżki umieść kwadrat w wybranym kolorze, symbolizujący prezent, do którego zmierza duszek-kot. Napisz grę, w której używając klawiszy strzałek, będziesz przesuwał duszka-kota w stronę prezentu. Duszek-kot ma przesuwać się o 5 kroków tylko po ścieżce – nie może wyjść poza nią. Jeśli duszek dotknie tła, powinien cofnąć się o 5 kroków (czyli przesunąć się o -5 kroków). Po scenie mają losowo przelatywać dwa duszki: duszek-duch i duszek-smok. Gdy duszek-kot i duszek-smok dotkną się, duszek-kot wraca na pozycję początkową i licznik zmniejsza się o jeden. Gdy duszek-kot i duszek-duch dotkną się, licznik zwiększa się o jeden. Gracz wygrywa, gdy duszek-kot dojdzie do prezentu (czyli dotknie koloru, na jaki pomalowano prezent). Należy wtedy wyświetlić komunikat „Gratulacje” i uzyskaną liczbę punktów. Gracz przegrywa, gdy uzyska -5 punktów – w takim wypadku ma wyświetlić się komunikat „Nie udało się!”. Zapisz program w pliku pod nazwą *kotek na ścieżce*.
11. Otwórz plik *pingwiny dwa poziomy* zapisany w ćwiczeniu 17. Zmodyfikuj grę, dodając kolejny poziom, np. zwiększając prędkość lotu duszka-ptaka i duszka-czarownicy na kolejnych poziomach. Zapisz plik pod nazwą *pingwiny trzy poziomy*.
12. Korzystając z możliwości rysowania własnego duszka i nowych kostiumów dla niego, utwórz animację rosnącego kwiatka. Narysuj najpierw kawałek lodygi, potem dalszą część, pączek, listki, kolejno rozwinięte płatki kwiatka. Aby animacja była płynna, przygotuj dziesięć kostiumów dla duszka. Kolejny kostium rysuj, duplikując poprzedni i zmieniaj lub uzupełniaj rysunek. Następnie zmień kostiumy, tak aby animacja „wróciła” do początku (kwiatek zwiędnie). Dodaj do animacji własne tło. Zapisz program w pliku pod nazwą *mój kwiatek*.
13. W języku Scratch utwórz multimedialną „Bajkę o rybaku i rybce” wzorowaną na tej, którą utworzyliśmy w środowisku programowania Báltie w temacie 6. (ćwiczenia 1-6). Zapisz program w pliku pod nazwą *rybak*.
14. Utwórz grę według własnego pomysłu. Korzystając z **Pomocy**, zapoznaj się z możliwością wykorzystania w programie **stopera**. Uwzględnij wyświetlanie czasu trwania utworzonej gry. Zapisz program w pliku pod nazwą *moja gra*.