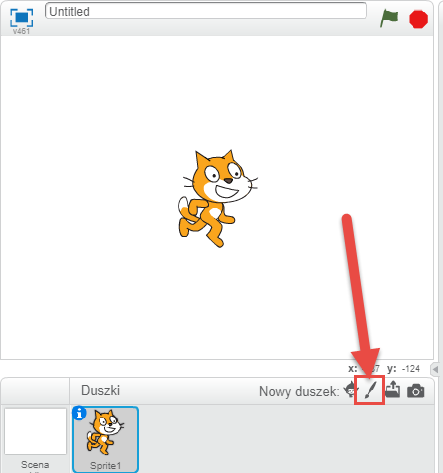
# **3. Ćwiczenia ze Scratch**

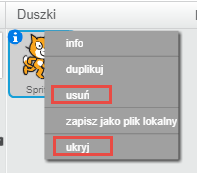
**Ćwiczenie 3.1**

## Projektowanie scen oraz duszków

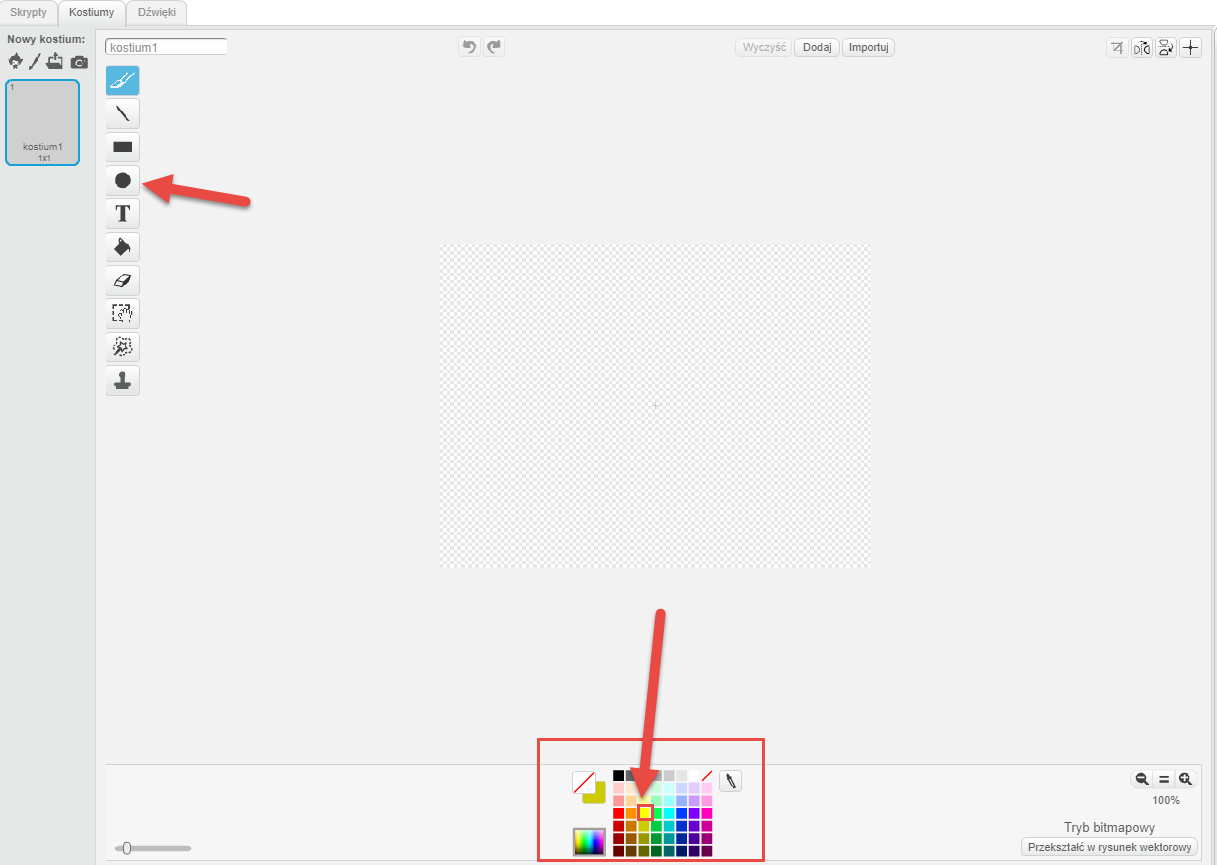
1. Uruchom przeglądarkę internetową i wejdź na stronę <https://scratch.mit.edu>
2. Przygotuj nowy projekt przez zaznaczenie opcji „stwórz”
3. Tworzymy nowego duszka. Na potrzebę następnych ćwiczeń stworzymy postać kultowego Pac-Mana (bardzo prosta koncepcja).
4. Wybieramy ikonkę pędzla pod aktywną sceną w panelu z Duszkami.

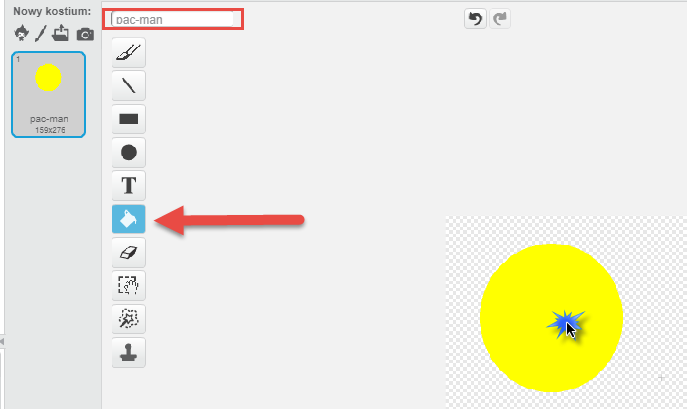


Domyślnego duszka, aby nie przeszkadzał możemy ukryć, bądź usunąć klikając na niego prawym przyciskiem myszy.

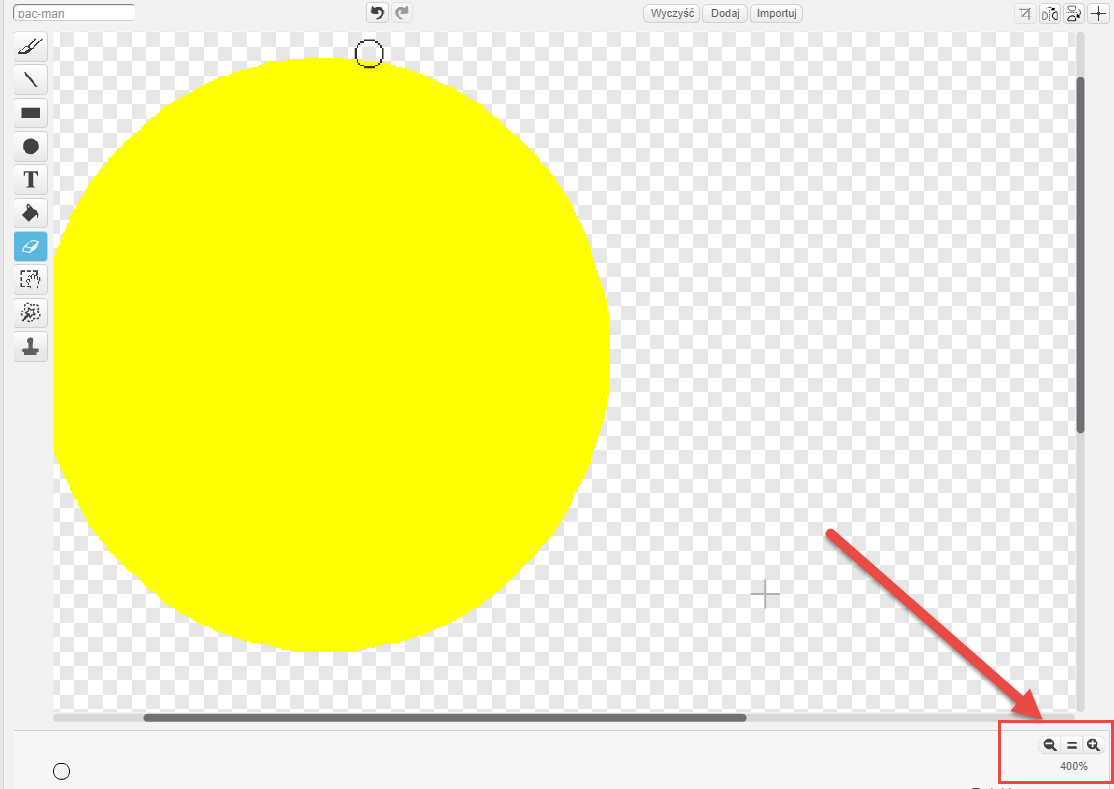


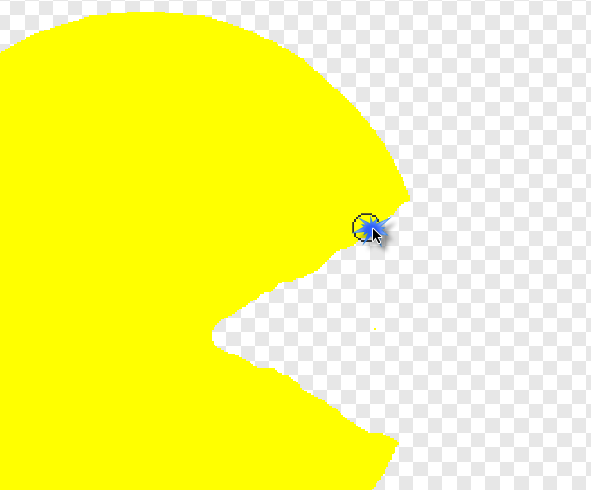
1. Tworzenie Pac-Mana zaczynamy od wybrania żółtego koloru, a następnie narysowania okręgu, który zostanie zapełniony żółtym kolorem.



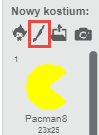


1. Kolejnym zabiegiem jest wycięcie „ust” Pac-Mana, w tym przypadku zastosujemy gumki, ponieważ chcemy wymazać niepotrzebną teksturę. Dla ułatwienia zadania możemy zwiększyć przybliżenie obrazu.



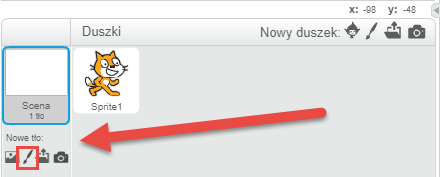


1. Dodajemy kolejne kostiumy dla Pac-Mana przedstawiające różne momenty podczas otwierania i zamykania ust (im więcej kostiumów, tym bardziej płynna będzie animacja)

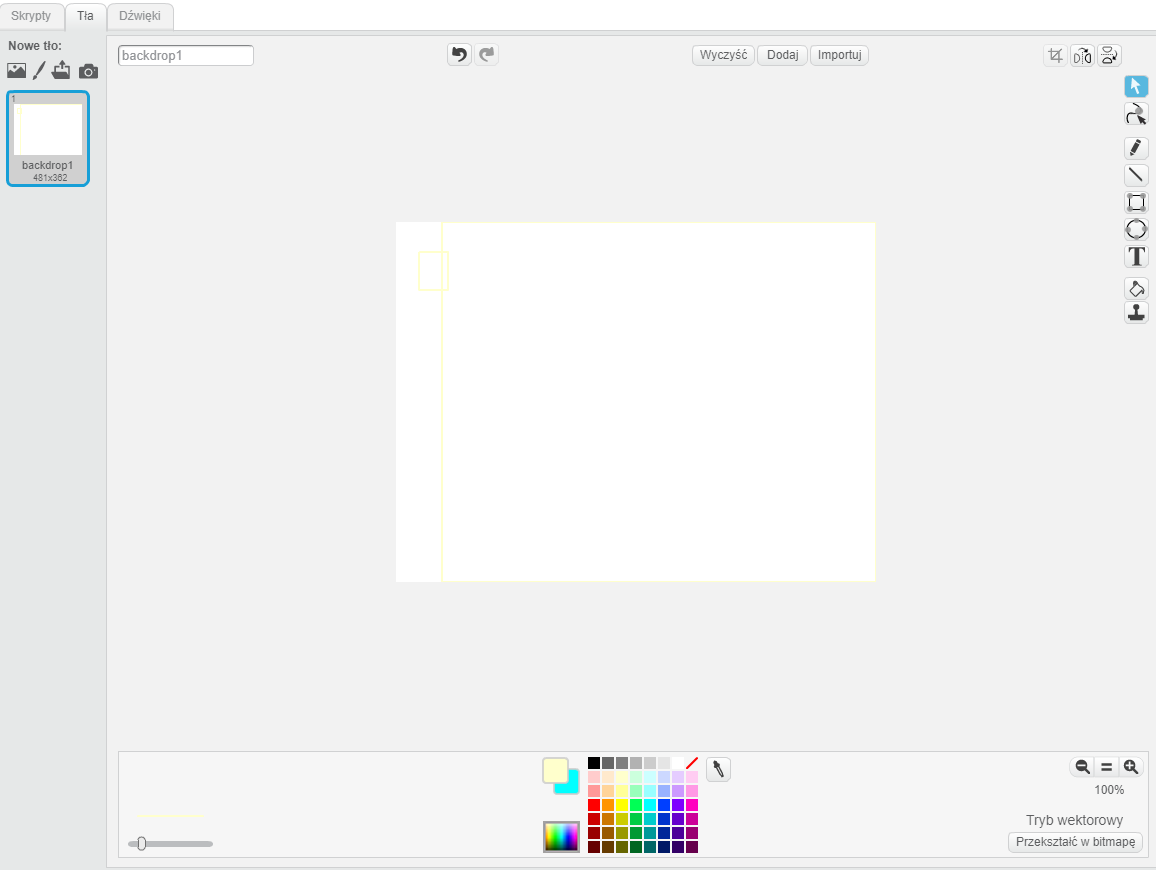
 

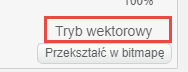
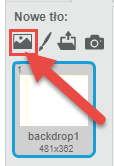
Tak przygotowanego duszka z kostiumami zapisujemy i zachowujemy do późniejszego ćwiczenia.

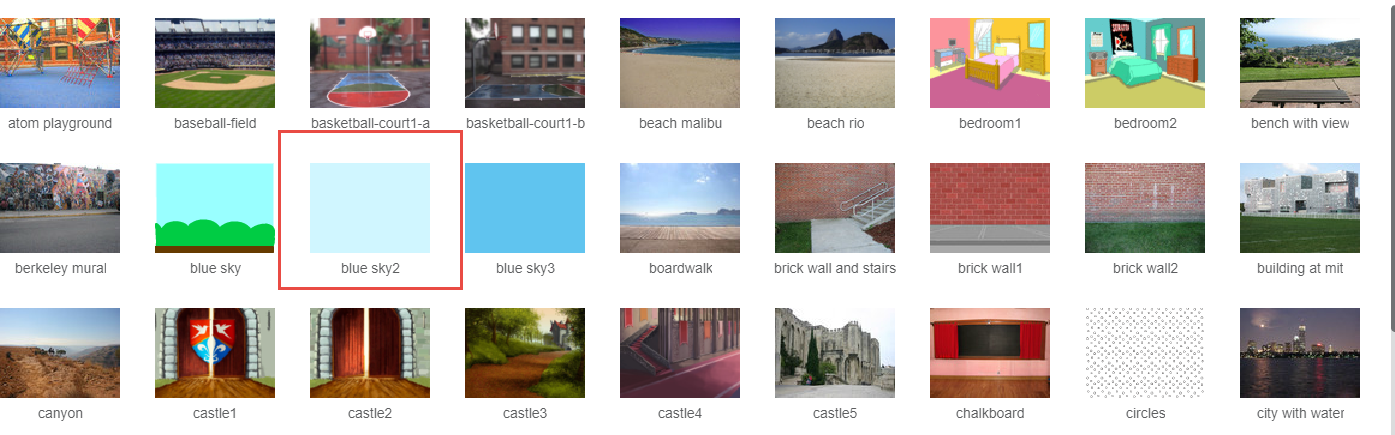
1. Tworzenie nowej sceny. Do stworzenia tła wykorzystamy prostokąty oraz cieniowanie w trybie wektorowym.
2. Na ekranie głównym klikamy w ikonke pędzla pod aktywną sceną



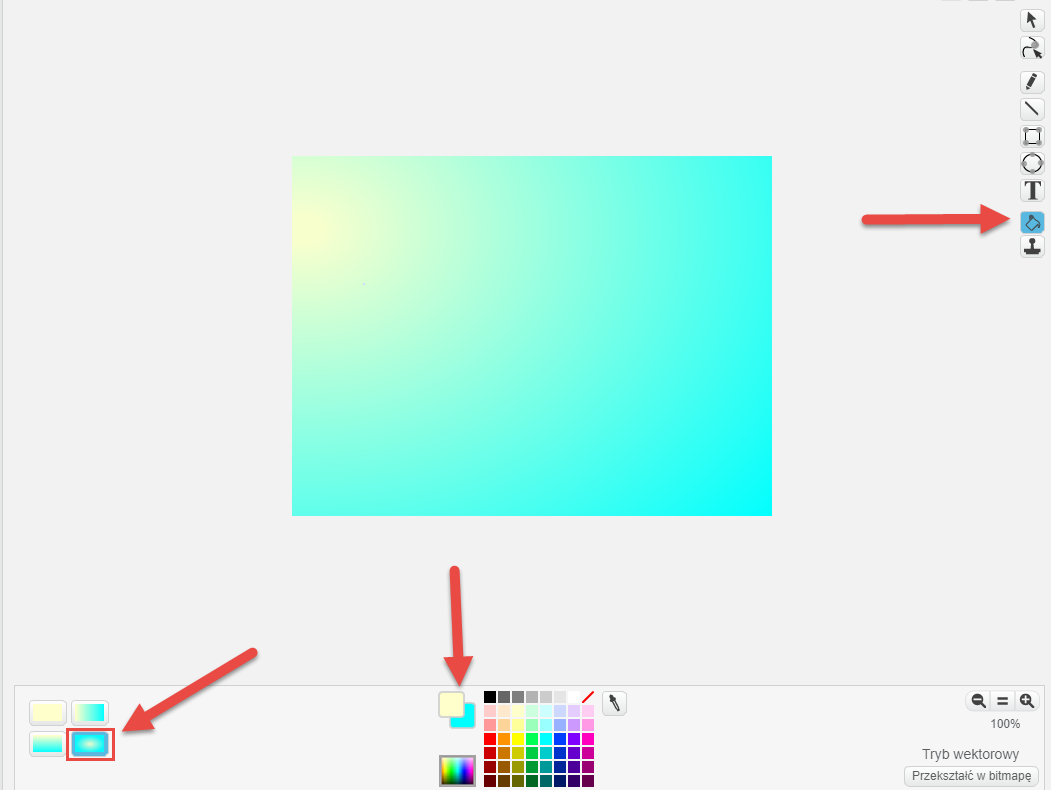
1. Pojawi się okno edycji sceny, zmieniamy tryb na wektorowy i wczytujemy jednolite tło (kolor nie ma znaczenia w tym przypadku), na które później naniesiemy zmiany.

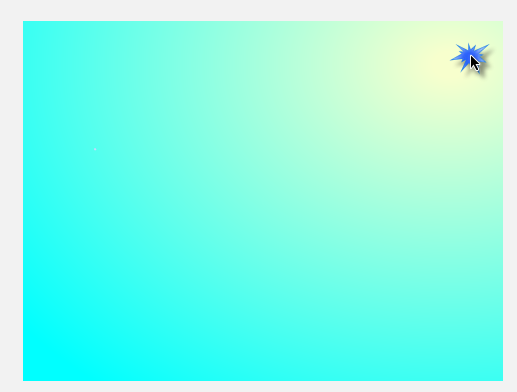


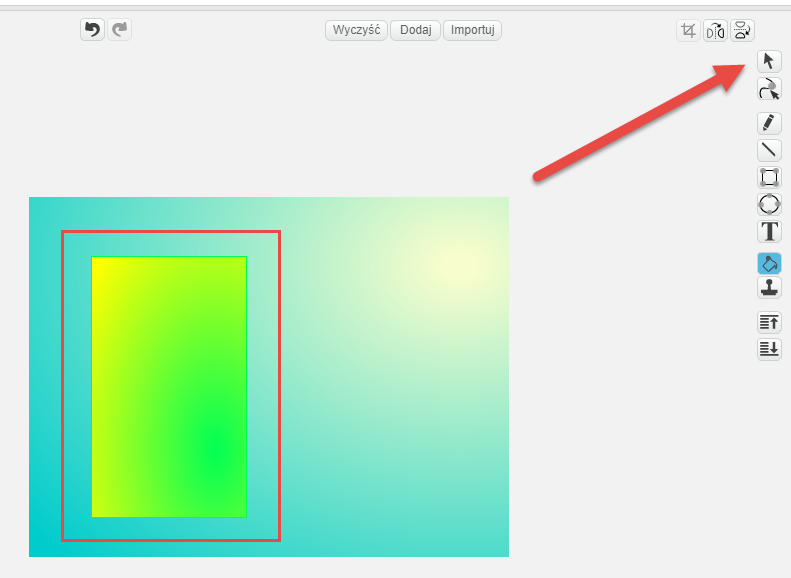


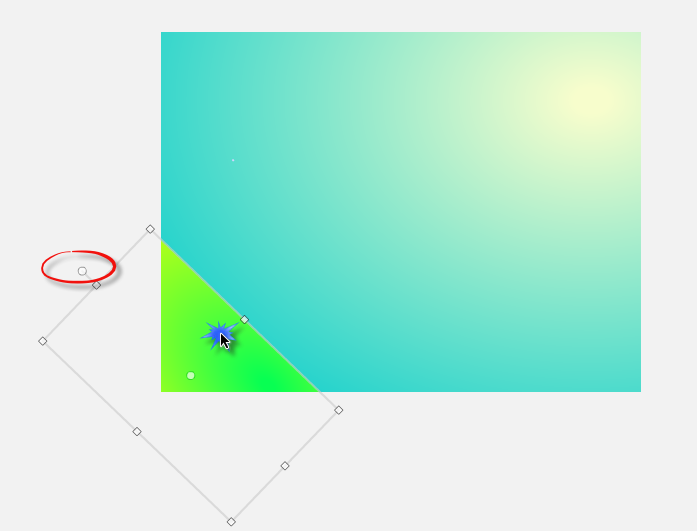
1. Klikamy na narzędzie wypełniania, po czym wybieramy odpowiedni kolor i nanosimy zmiany na nasze tło tworząc efekt nieba.



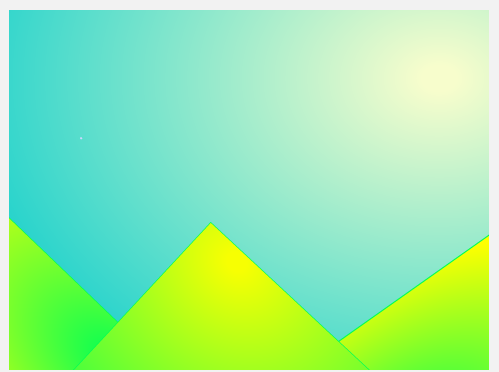


1. Kolejnym krokiem będzie stworzenie dodatkowych elementów tła poprzez dodanie trzech prostokątów i nałożenie zielono – żółtego cieniowania. Prostokąty przenosimy przytrzymując lewy przycisk myszy częściowo poza ekran aktywnej sceny, aby wydobyć pożądany efekt.





1. Zachowujemy scenę na potrzeby następnego ćwiczenia

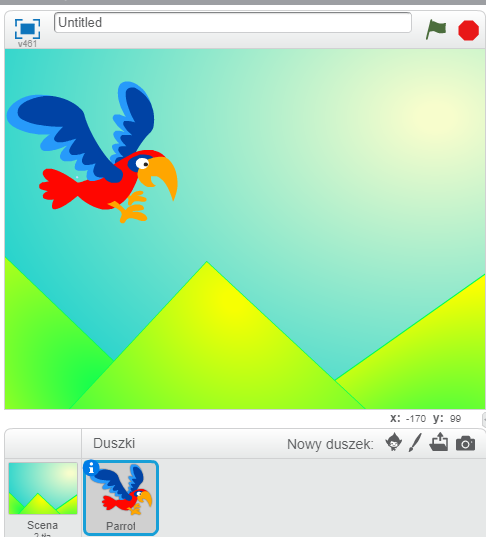


**Ćwiczenie 3.2**

# **Zastosowanie różnych skryptów w celu rozwiązania tego samego problemu.**

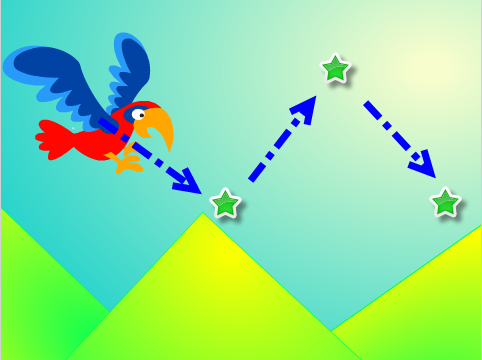
*Do wykonania ćwiczenia wykorzystamy scenę z poprzedniego ćwiczenia.*

1. Uruchamiamy projekt z przygotowaną sceną, bądź wczytujemy scenę do nowego projektu, oraz tworzymy nowego duszka – Papugę.

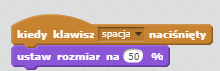


1. Celem naszego ćwiczenia będzie przemieszczenia papugi w wybrane miejsca, a następnie powrót w drugą stronę. Papuga powinna przemieszczać się w taki sposób bez przerwy.

Patrząc się na dostępne skrypty (przede wszystkim ruchu) zastanówmy się w jaki sposób możemy przemieścić papugę w konkretne, wybrane miejsca? Rozwiązań może być wiele!



Ponieważ domyślnie papuga zajmuje dość dużą część ekranu, dobrym pomysłem było by zmniejszenie jej wielkości. Można to zrobić edytując duszka, bądź wykorzystując skrypt pomniejszenia:



Do rozwiązania problemu możemy wykorzystać między innymi:

1. Skrypty obrotu oraz wykonania wybranej ilości kroków



Powyższe rozwiązanie jest jednym z najprostszych, ale czy na pewno jest najbardziej efektywne? Proszę zwrócić uwagę ilość zastosowanych skryptów.

*Jeśli chcemy zresetować pozycję papugi, możemy nacisnąć na nią lewym przyciskiem myszy i przeciągnąć na wybraną pozycję!*

1. Skrypty wykorzystujące położenie względem osi XY

Szukamy odpowiedniego punktu poprzez przemieszczanie kursora myszy po ekranie śledząc zmianę współrzędnych.

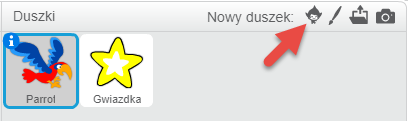


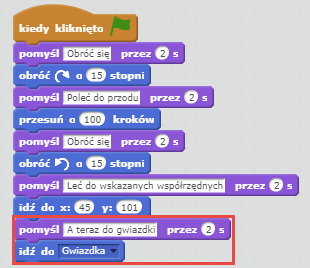
Wprowadzamy zapisane współrzędne do odpowiedniego skryptu.



1. Ruch do innego obiektu (duszka)

Do realizacji tego rozwiązania potrzebujemy dodatkowego duszka, do którego będziemy mogli odnieść się w skrypcie. Wyszukujemy go z blibioteki, a następnie umieszczamy na scenie.





Które z rozwiązań najbardziej Ci odpowiada i dlaczego?

1. Kolejnym problemem będzie stworzenie drogi powrotnej. Możemy zastosować te same skrypty, lecz musimy nanieść na nie odpowiednie zmiany. Dodatkowo, należy odpowiedzieć na następujące pytania:
2. Jak obrócić papugę?

Obrócenie duszka najprościej wykonać poprzez dodanie dodatkowego kostiumu, obróconego wokół osi Y.

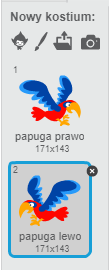
Duplikujemy podstawowy kostium.



Na ekranie duplikatu wybieramy obracanie.



Zostawiamy obrócony kostium, do którego będziemy odnosić się w skrypcie wyglądu

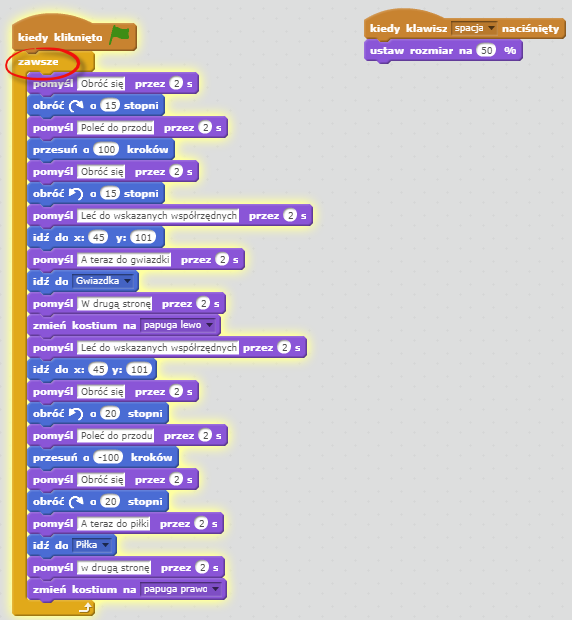


1. Jakie zmiany wprowadzić w skopiowanych skryptach, aby papuga poruszała się w dobrą stronę?

* Współrzędne mogą zostać bez zmian
* Podczas poruszania się o wybraną ilość kroków np. 100, należy dodać wartość ujemną. Dlaczego? Poruszanie się o 100 kroków przesuwa postać o 100 kroków w **prawo**, wartością odwrotną, czyli **lewo** będzie -100 kroków
* Dodajemy kolejnego duszka, poza gwiazdką aby ustawić pozycję startową



1. Ostatnią czynnością jest zapętlenie wykonywanej czynności aby papuga podróżowała bez przerwy w lewą, a następnie prawą stronę. W tym celu wykorzystamy blok ***zawsze***



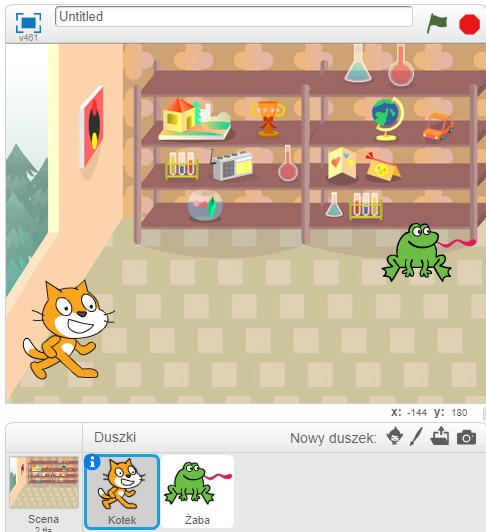


**Ćwiczenie 3.3**

## **Zabawa w ściganego**

W tym ćwiczeniu wykorzystamy poruszanie duszkiem za pomocą myszki oraz interakcję między obiektami w programie.

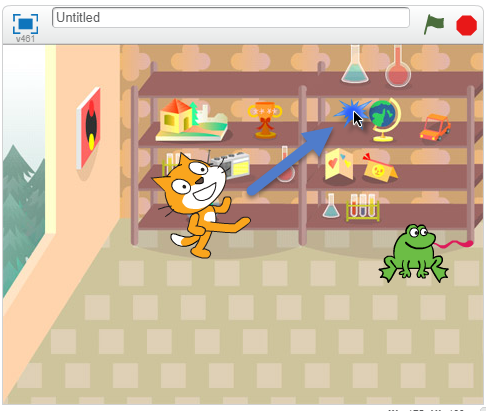
1. Przygotowujemy dwa dowolne duszki oraz wybraną przez siebie scenę.



1. Programujemy ruch za myszką (w tym przypadku dla kotka)



Powyższy skrypt sprawi, że nasz duszek cały czas przesuwać się będzie w stronę pozycji kursora myszki.



1. Programujemy zachowanie drugiego duszka (żaby)
2. Poniższy skrypt również poruszy żabę w stronę naszego kursora oraz nałoży warunek, który w przypadku dotknięcia przez kota wyświetli komunikat o złapaniu. Zatrzymanie skryptu jest w tym przypadku bardzo ważne, aby ta sama instrukcja nie wykonywała się wiele razy w kółko (kiedy żaba zostanie już złapana).





1. Niewielkim usprawnieniem będzie ustawienie żabie konkretnej pozycji startowej podczas rozpoczęcia programu po to, aby nie została przypadkowo od razu złapana przez kotka. Ustawienie kierunku na -90 nakieruje żabę na kota.



1. Programujemy zachowanie pierwszego duszka (goniącego)

Skrypt nie będzie znacznie różnił się od tego w punkcie 3. Pozycję startową duszka ustawiamy przeciwną do drugiej postaci, po to aby duszki pojawiły się po dwóch różnych stronach sceny.





W jaki sposób można usprawnić powyższą grę, bądź dodać jej dodatkowe, ciekawsze elementy?