

2.1. Duszek w labiryncie

DOWIESZ SIĘ, JAK

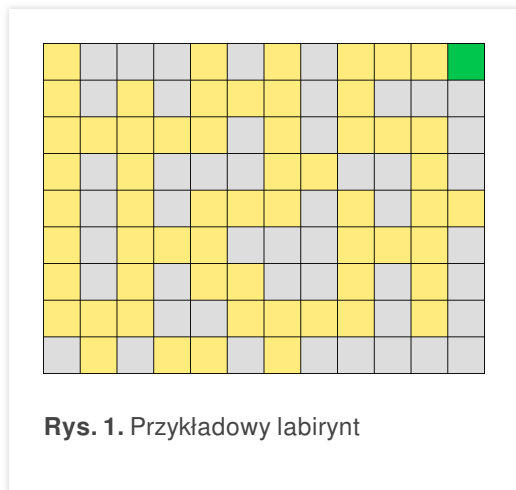
- sterować duszkiem,
- stosować pętlę **zawsze**,
- wykorzystywać blok warunkowy **jeżeli**.

Sterowanie żółwiem, duszkiem lub innym obiektem znasz ze szkoły podstawowej. Twój pierwszy projekt w środowisku Scratch będzie polegał na przeprowadzeniu duszka przez labirynt za pomocą klawiszy ze strzałkami.

PROJEKTOWANIE SCENY

Na początku trzeba przygotować **tło sceny**, czyli labirynt (rys. 1) możliwy do przejścia z miejsca startu do mety. Najlepiej to zrobić w programie graficznym (plik PNG), przyjmując następujące założenia:


- wymiary sceny 480 × 360,
- kratki wielkości 40,
- dwa dowolne kolory krutek, końcowa kratka – zielona.



STEROWANIE DUSZKIEM

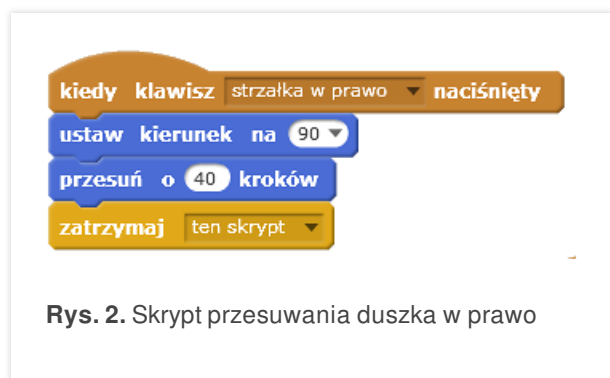
- Otwórz Scratcha i wybierz opcję **Nowe tło** → **Plik**, aby wstawić plik tła przygotowany w edytorze grafiki.

Twój duszek będzie chodził po polach labiryntu, dlatego musisz zmniejszyć jego postać.

- Kliknij przycisk ze strzałkami skierowanymi do środka  (kształt kursora się zmieni) i tak długo klikaj postać duszka, aż jego wielkość będzie odpowiednia.

Teraz zaprogramujesz sterowanie duszkiem. Utworzysz w tym celu odpowiednie skrypty. Załóż, że będziesz sterować duszkiem za pomocą klawiszy ze strzałkami – np. po naciśnięciu klawisza ze strzałką skierowaną w prawo duszek powinien pójść w kierunku prawego brzegu ekranu.

- Ułóż skrypt dla duszka, czyli polecenia do wykonania po naciśnięciu klawisza ze strzałką w prawo (rys. 2). Potrzebne bloki znajdziesz w grupach o odpowiednich kolorach.



Jeśli ten skrypt działa prawidłowo, wstaw kolejne skrypty do przesuwania się duszka w pozostałych kierunkach.


- Skopiuj skrypt – naciśnij prawy przycisk myszy i z menu podręcznego wybierz opcję **duplikuj**. Zmień rodzaj naciskanego klawisza (strzałka w lewo, strzałka w dół, strzałka w górę) i wybierz odpowiedni do tego kierunek ruchu duszka.
- Sprawdź, czy duszek porusza się w odpowiedni sposób na ekranie. Czy czasami obraca się i maszeruje do góry nogami? Aby temu zapobiec, trzeba zmienić styl jego obrotów.

URUCHAMIANIE PROJEKTU

- Najpierw ustal warunki początkowe, tworząc skrypt uruchamiany po kliknięciu zielonej flagi. Potem kolejno: ustaw duszka w położeniu początkowym i ustal odpowiedni styl obrotów (rys. 3).



Rys. 3. Skrypt zielonej flagi

 – uruchamia projekt,

 – zatrzymuje wszystkie skrypty,

 – maksymalizuje scenę.

Następnym problemem do rozwiązania jest trasa labiryntu – duszek powinien chodzić tylko po żółtych polach. Szare kwadraty są ścianami labiryntu.

- Ułóż z klocków kolejny skrypt **kiedy kliknięto zieloną flagę** (rys. 4).

Pętla **zawsze** działa *non stop* i wykonuje polecenia zawarte w jej klamrach. Blok warunkowy **jeżeli** sprawdza, czy warunek jest spełniony (czy duszek dotyka wybranego koloru); jeśli tak, to wykonuje się odpowiednie polecenie – duszek cofa się na poprzednie pole.



Rys. 4. Kolejny skrypt zielonej flagi

Aby wstawić odpowiedni kolor do warunku bloku **jeżeli**, należy go kliknąć, a potem kliknąć szary kwadrat na planszy labiryntu. Twój projekt działa – sterujesz duszkiem po polach labiryntu.

- Zrób jeszcze coś na zakończenie, gdy duszek dojdzie do mety – zielonego pola labiryntu. Przeanalizuj poniższe dwie możliwości (rys. 5).




Rys. 5. Skrypty na zakończenie

ZADANIA

1. Do projektu zrealizowanego podczas lekcji wprowadź zliczanie błędnych ruchów duszka.

– Na karcie **Dane** w skryptach duszka utwórz zmienną **suma**.

 Zmienna to rodzaj „szufladki”, w której przechowuje się jakąś wartość.

– Podczas ustalania warunków początkowych ustaw wartość tej zmiennej na 0.



– Jeśli przesuwasz duszka o –40 kroków, to zwiększ wartość zmiennej **suma** o 1.



– Po dojściu duszka do mety wypisz liczbę popełnionych przez niego błędów.



2. Uzupełnij projekt wykonany w zadaniu 1. o zliczanie drogi przebytej przez duszka i sprawdzanie, czy była to najkrótsza droga.



- Programowanie sterowania duszkiem
- Procedury bezparametrowe i z parametrem
- Tworzenie własnych bloków (procedur)
- Realizacja gry *Papier, nożyce, kamień*
- Animacja postaci duszka
- Wykorzystanie dźwięku w projektach
- Znajdowanie minimum i maksimum ciągu (serii) liczb
- Sprawdzanie parzystości liczby, szukanie liczb pierwszych
- Tworzenie skryptów rekurencyjnych
- Rozwiązywanie problemu wież Hanoi
- Algorytmiczne obliczanie największego wspólnego dzielnika (NWD) – algorytm Euklidesa
- Algorytm obliczania liczb Fibonacciego
- Algorytm porządkowania przez scalanie