Tworzenie gry logicznej - labirynt

Materiał "Tworzenie gry logicznej - labirynt" składa się z sekcji: "Scenariusz gry", "Zasady gry i przepis dla komputera", "Tworzenie skryptów duszków", "Dodanie licznika żyć myszki", "Zadania uzupełniające".

Materiał zawiera zrzuty ekranu, filmy oraz ćwiczenia.

Tworzenie gry logicznej - labirynt

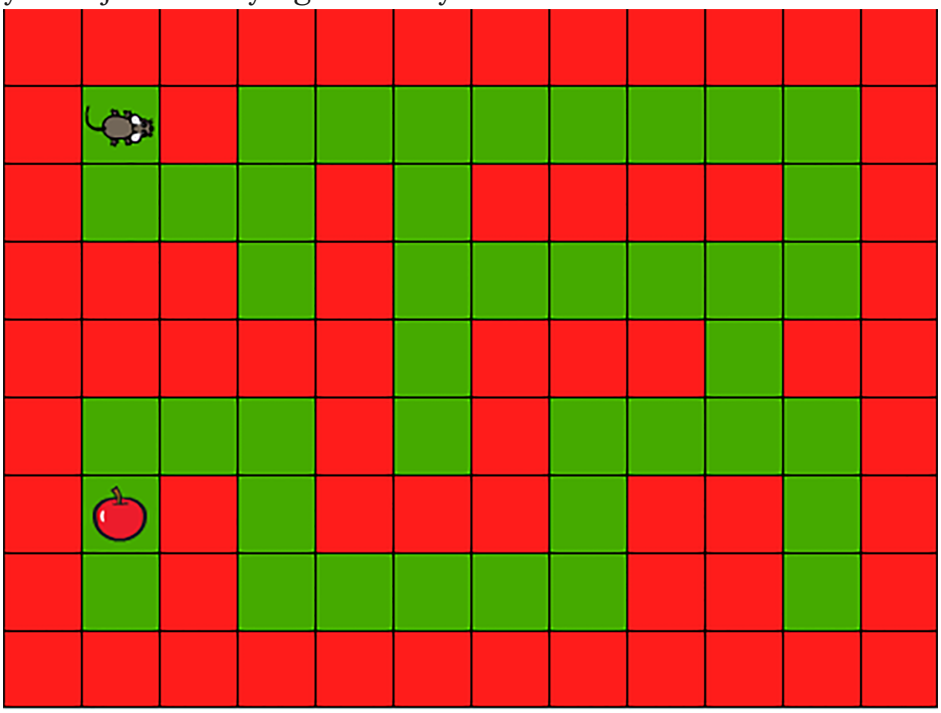
Scenariusz gry

Pewnie nie raz bawiliście się grą, w której bohater musiał przejść przez jakiś labirynt, wykonując określone zadania – np. zebrać skarby, uniknąć spotkania z groźnymi przeciwnikami i dotrzeć do określonego celu. Twoje zadanie będzie polegało na stworzeniu podobnej gry w programie Scratch.

Jeżeli potrzebujesz przypomnieć sobie podstawy środowiska Scratch, skorzystaj z tego materiału.

Najpierw zastanów się nad scenariuszem gry, bohaterami, którzy w niej wystąpią, nad tym, co będzie ich zadaniem i jakie przeszkody napotkają. Musisz także pomyśleć, jak będzie wyglądała plansza gry, a następnie opisać wszystkie czynności, które powinien wykonywać komputer.

Zacznij od projektu, w którym występuje mysz (steruje nią gracz). Jej zadanie polega na pokonaniu labiryntu i zdobyciu jabłka. Sceneria gry będzie przypominać szachownicę. Gracz będzie mógł przesunąć mysz za każdym razem o jedno pole oraz zmienić kierunek, w którym mysz będzie się poruszała. Na planszy wystąpią pola w dwóch kolorach – zielonym i czerwonym. Zielony będzie oznaczał korytarz, czyli pola, po których mysz będzie mogła się poruszać, a czerwony – ściany, czyli pola, niedostępne dla myszy. Gracz wygrywa, gdy doprowadzi mysz do jabłka ukrytego w labiryncie.



Podgląd gry labirynt.

Wykorzystaj poniższy dzienniczek na zapisanie swoich notatek i przemyśleń.

Źródło: GroMar, licencja: CC BY 3.0.

Ćwiczenie 1

Określ, czym powinno być jabłko, które ma zdobyć mysz. Rysunkiem, który wykonasz na scenie, czy drugim duszkiem – a może masz inne pomysły?

Źródło: GroMar, licencja: CC BY 3.0.

Polecenie 1

Uruchom program Scratch i wczytaj do niego planszę z labiryntem jako nowe tło sceny. Pamiętaj o usunięciu domyślnego, białego tła.

Poniżej możesz pobrać plik graficzny, którego możesz użyć jako planszy z labiryntem.
W kolejnych ćwiczeniach będziemy korzystać z tej planszy, ale możesz również stworzyć własną grafikę.

Tło z labiryntem.
Źródło: GroMar, licencja: CC BY 3.0.
Plik o rozmiarze 15.68 KB w języku polskim

Ćwiczenie 2

Usuń domyślnego duszka – kotka. Dodaj z galerii duszki z postaciami myszy oraz jabłka. Zmień ich nazwy odpowiednio na: **Mysz** oraz **Jabłko**. Ustaw ich rozmiar w taki sposób, by w całości mieściły się na pojedynczym polu planszy.

Uwaga!

Pamiętaj, żeby zapisać swój projekt - np. pod nazwą **Labirynt 1**. Następne wersje projektu możesz zapisywać, zmieniając jedynie numer w nazwie na kolejny. Dzięki temu możliwe

będzie ewentualne wrócenie do dowolnej, wcześniejszej wersji.

Zasady gry i przepis dla komputera

Mając gotowy scenariusz, musisz uszczegółowić wszystkie czynności, które będzie wykonywał komputer. To znaczy, że należy określić, kiedy i w jaki sposób komputer ma reagować, aby gra działała dokładnie tak, jak chcesz.

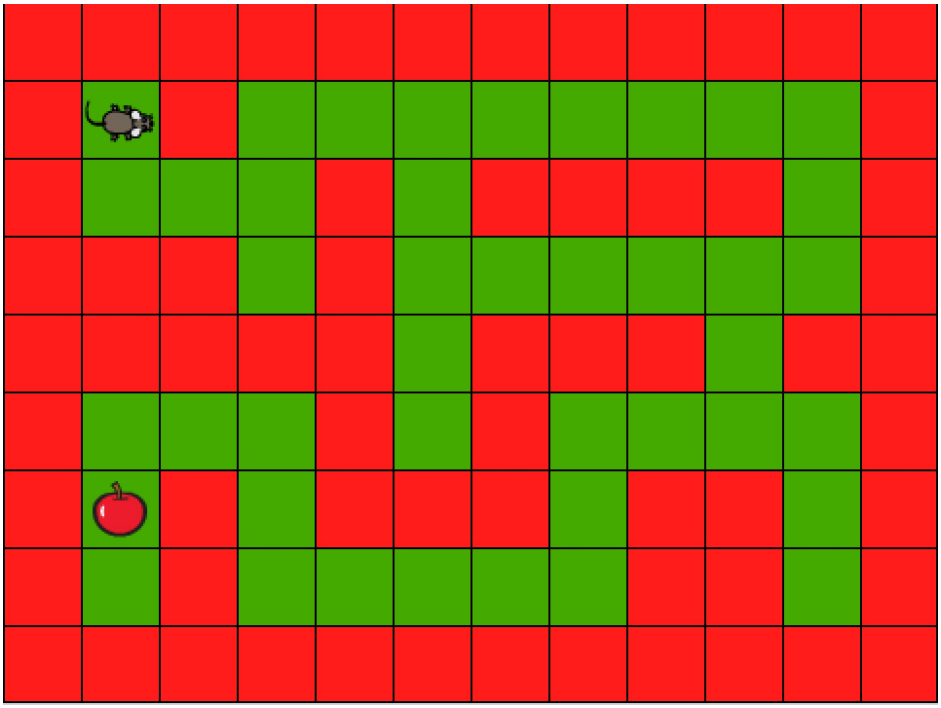
Ćwiczenie 3

Określ, jakie czynności powinien wykonywać komputer w tej grze, czyli co trzeba zaprogramować. Podziel je na trzy kategorie:

1. **czynności startowe** (wszystkie czynności wykonywane przed rozpoczęciem gry, np. po wciśnięciu zielonej flagi),
2. **sterowanie** (jak gracz będzie poruszał myszą),
3. **zdarzenia** w grze (co ma się stać, np. gdy mysz dotrze do jabłka).

Ćwiczenie 4

Do ustawienia duszków myszy oraz jabłka na pozycjach startowych możesz wykorzystać blok **idź do x: ... y: ...** . Oblicz wartości x oraz y, które pozwolą umieścić postaci na środku tych samych pól, co na przykładowej planszy poniżej. Wpisz i sprawdź swoją odpowiedź.



Plansza gry - labirynt.

Mysz powinna zostać umieszczona na pozycji x [ ] y [ ].

Jabłko powinno zostać umieszczone na pozycji x [ ] y [ ].

Ćwiczenie 5

Określ, o ile kroków powinna przesunąć się mysz, aby znaleźć się dokładnie na środku następnego pola? Wpisz i sprawdź swoją odpowiedź.

Mysz powinna przesunąć się o [ ] kroków, aby znaleźć się na środku sąsiedniego pola.

Ćwiczenie 6

Podaj o wielokrotność ilu stopni powinien obracać się duszek (mysz), by mógł poruszać się tylko pionowo (w górę lub w dół) i poziomo (w prawo lub w lewo).

Duszek powinien obracać się o wielokrotność _____ stopni.

Ćwiczenie 7

Określ, co ma się stać, gdy mysz wejdzie na czerwone pole.

Ćwiczenie 8

Określ, co ma się stać, gdy mysz dotrze do jabłka.

Ćwiczenie 9

Określ, jakie klawisze wykorzystasz do sterowania duszkiem - myszą.

Tworzenie skryptów duszków

W trakcie poprzednich zadań, ustalono następujące reguły dotyczące gry:

1. **Czynności startowe**
    - Ustawienie myszy na polu startowym (np. $x: -180, y: 120$),
    - Ustawienie jabłka na odpowiednim polu (np. $x: -180, y: -80$),
2. **Sterowanie**
    - Strzałka w prawo: obróć o 90 stopni w prawo,
    - Strzałka w lewo: obróć o 90 stopni w lewo,
    - Spacja: przesuń o jedno pole (czyli 40 kroków),
3. **Zdarzenia**
    - Jeśli mysz zostanie wprowadzona na czerwone pole:
        - wyda dźwięk **pop** (dostępny w galerii dźwięków),
        - cofnie się na poprzednie pole.
    - Jeśli mysz dotrze do jabłka:
        - wyświetli w dymku komiksowym *„Hura! Wygrałam”*,
        - zakończy działanie wszystkich skryptów.

Należy jeszcze ustalić, w którym miejscu skryptów powinna nastąpić kontrola zdarzeń, czyli sprawdzenie, czy mysz nie weszła na czerwone pole lub nie znalazła jabłka. Zacznij jednak od utworzenia skryptów, które zrealizują czynności startowe oraz sterowanie. Później dodasz obsługę zdarzeń.

> **Ćwiczenie 10**
>
> Stwórz skrypty dla myszy oraz jabłka, które po wciśnięciu zielonej flagi ustawią duszki na pozycjach startowych. Przetestuj działanie skryptów i sprawdź, czy duszki znalazły się na pozycjach startowych po wciśnięciu zielonej flagi.

> **Ćwiczenie 11**
>
> Stwórz skrypty sterujące duszkiem - myszą zgodnie z ustaleniami z poprzednich zadań.
>
> - **Strzałka w prawo**: obróć o 90 stopni w prawo,
> - **Strzałka w lewo**: obróć o 90 stopni w lewo,
> - **Spacja**: przesuń o jedno pole (czyli 40 kroków).
>
> Przetestuj działanie skryptów. Sprawdź, czy możesz obracać duszka (mysz) oraz przesuwać na planszy.

Zwróć uwagę, że duszek - mysz może w tej chwili poruszać się nie tylko po zielonych polach, ale także po czerwonych.

Ćwiczenie 12

Wskaż bloki, które można byłoby wykorzystać do zbadania, czy mysz weszła na pole oznaczone kolorem czerwonym. Który z nich będzie najlepiej pasował?

Ćwiczenie 13

Określ zdarzenia, w których powinniśmy sprawdzić, czy duszek - mysz znajduje się na czerwonym polu?

Ćwiczenie 14

Zmodyfikuj skrypt wywoływany klawiszem **spacja** w taki sposób, aby po przesunięciu duszka badał, czy nie znalazł się na czerwonym polu. W tym celu wykorzystaj klocek **dotyka koloru ... ?** oraz klocek **jeżeli ... to** . Połącz je odpowiednio ze sobą. Jeżeli duszek dotyka koloru czerwonego, to zgodnie z planem powinien:

- wydać dźwięk **pop**,
- cofnąć się na poprzednie pole.

Teraz mysz nie może już przebywać na czerwonym polu. Pozostało jeszcze zaprogramować, co ma się stać, gdy dotrze do jabłka.

Ćwiczenie 15

Zmodyfikuj skrypt reagujący na naciśnięcie klawisza **spacja**, w taki sposób, aby sprawdzał po przesunięciu duszka - myszy, czy nie dotyka on duszka - jabłka.

Ważne!

> Zapisz swój projekt pod kolejną nazwą, np. **Labirynt 2**.

Dodanie licznika żyć myszy

W celu uatrakcyjnienia gry zmień nieco jej scenariusz. Dodaj licznik żyć dla myszy. Przy starcie wartość licznika ustal na przykład na 3 życia. Każde wejście na ścianę będzie odejmowało jedno życie. Jeżeli licznik osiągnie wartość zero, wówczas gra skończy się przegraną.

Realizacja powyższego scenariusza wymaga przechowania informacji o liczbie żyć, czyli danej. Dane mogą być przechowywane przez program w zmiennych.

W uproszczeniu zmienna to skrytka (szufladka), w której przechowuje się na przykład liczbę. Program może do niej zajrzeć i sprawdzić, jaka jest wartość zmiennej (przechowywana liczba). Wolno także zmieniać tę wartość – na przykład dodać do niej inną liczbę.

> **Ćwiczenie 16**
>
> Stwórz zmienną i nazwij ją **Licznik żyć**. Zmienna powinna być dostępna dla wszystkich duszków.

> **Ćwiczenie 17**
>
> Zmodyfikuj skrypty w taki sposób, aby uwzględniony został licznik żyć myszy, który na początku przyjmie wartość, np. 3. Za każdym razem, gdy mysz wejdzie na ścianę, jego wartość zmniejsz o 1. Gdy wartość osiągnie 0, wyświetl napis *„Koniec gry”* przez 2 sekundy, następnie zatrzymaj wszystkie skrypty.

Sprawdź podczas testowania zmian, że nawet gdy mysz straci wszystkie życia, można nią dalej poruszać, a gdy dojdzie do jabłka - wyświetli napis *„Hurra! Wygrałam!”*. Dalsze poruszanie myszy, wynika z działania środowiska Scratch. Jednak możesz oprogramować grę w taki sposób, aby uniemożliwić wygraną, jeśli mysz nie ma więcej żyć, na przykład poprzez ukrycie jabłka lub zatrzymanie skryptu programowo dodając w skrypcie warunek, który nigdy nie będzie spełniony, np. **liczba żyć > 0**.

Ćwiczenie 18

Jeżeli gra zakończy się, ponieważ mysz wykorzystała wszystkie życia, ukryj jabłko. Wykorzystaj w tym celu zmienną **Licznik żyć** i zmodyfikuj odpowiednio skrypty duszka - jabłka.

> **Uwaga!**
>
> Nie zapomnij pokazać duszka - jabłka za każdym razem, gdy zostanie naciśnięta zielona flaga!

Ważne!

Zapisz swój projekt pod kolejną nazwą, np. **Labirynt 3**.
Ciekawostka

Tworząc zmienne, możesz zdecydować, czy chcesz, aby dostęp do danej zmiennej miały tylko skrypty wybranego duszka, czy też wszystkie skrypty, niezależnie od tego, do którego z duszków zostały przypisane. Jeśli w twoim projekcie dwie postaci poruszałyby się po labiryncie, należy stworzyć dwie zmienne na przykład o nazwie **Liczba żyć**. Jednak każda z tych zmiennych powinna być przypisana w tym przypadku do konkretnego duszka. Dzięki temu obie zmienne nazywałyby się tak samo, ale przechowywały wartość dostępną tylko dla wybranego duszka.
Jeżeli chcesz, aby duszki mogły sprawdzić, ile żyć pozostało temu drugiemu, należy utworzyć zmienne o różnych nazwach i udostępnić je wszystkim duszkom.
Pamiętaj, żeby nadawać zmiennym takie nazwy, które mówią, co w danej zmiennej jest przechowywane. Unikaj nazw typu **Zmienna1, Zmienna2; A, B** – zazwyczaj nic one nie znaczą i po pewnym czasie, gdy zajrzysz do skryptów, nie będzie wiadomo do czego były wykorzystane.

Zadania uzupełniające

Polecenie 2

Podstawowa wersja gry już działa, jednak sterowanie myszą jest dość trudne. Warto je uprościć, wykorzystując w tym celu na przykład klawisze strzałek w lewo, do góry, w prawo, na dół. Wciśnięcie każdego z tych klawiszy powinno przesunąć duszka w kierunku zgodnym z wciśniętą strzałką.
Zastanów się, jak zmodyfikować skrypt reagujący na naciśnięcie klawisza spacji w taki sposób, aby reagował na naciśnięcie klawisza strzałka w dół, a przed przesunięciem duszka ustawił go w kierunku do dołu.
Popraw skrypt i przetestuj jego działanie.
Następnie zbuduj skrypty, które reagując na klawisze pozostałych strzałek, skierują i przesuną duszka odpowiednio: w górę, w lewo lub w prawo. Pamiętaj, że każdy ze skryptów musi kontrolować, czy w wyniku przesunięcia mysz weszła na czerwone pole lub dotarła do jabłka.
Możesz zduplikować skrypt reagujący na klawisz strzałki w dół i przerobić go tak, by reagował na inny klawisz strzałki.

> **Uwaga!**
>
> Usuń skrypty obracające duszka. Nie będą dłużej potrzebne!

Wskazówka

Duplikowanie skryptów:
Możesz powielić dowolny skrypt i na jego podstawie stworzyć nowy, wprowadzając odpowiednie zmiany. W tym celu wykonaj poniższe czynności.

- Kliknij na pierwszym klocku skryptu od góry prawym klawiszem myszy.
- Z menu wybierz opcję **duplikuj**. Przesuń kursor myszy – będzie się przesuwał razem z powielonym skryptem.
- Kliknij lewym klawiszem myszy w miejscu, gdzie będzie znajdował się cały skopiowany skrypt.

Polecenie 3

Dodaj drugi kostium dla duszka - jabłka, który będzie wyglądał jak ugryzione jabłko. Następnie dodaj skrypt, który cały czas będzie badał, czy jabłko nie dotyka myszy. Jeżeli dotyka, to kostium zostanie zmieniony na ogryzek, a np. po 1 sekundzie duszek zostanie ukryty.

> **Uwaga!**
>
> Nie zapomnij ustawić kostiumu duszka - jabłka na całe jabłko oraz pokazać duszka za każdym razem, gdy zostanie wciśnięta zielona flaga!

Polecenie 4

Dodaj drugie tło z innym labiryntem. Gdy gracz przejdzie przez pierwszy labirynt, zmień tło na następne. Pamiętaj, aby w tym przypadku ponownie ustawić wszystkie duszki na pozycjach startowych. Pozycje te mogą być inne niż na pierwszej planszy.

Polecenie 5

Podobnie jak został dodany licznik żyć, można wprowadzić zdobywanie punktów. Punkty dodajemy na przykład za liczbę zebranych jabłek. W tym wypadku warto dodać kilka dodatkowych jabłek na planszy. Gra może kończyć się zwycięstwem, gdy gracz zbierze wszystkie jabłka.

Polecenie 6

Zrealizuj własne pomysły, co ma się stać, gdy mysz wejdzie na ścianę lub dotrze do jabłka. Zmodyfikuj projekt zgodnie z nimi.

Polecenie 7

Dodaj do gry nowe duszki, które będą przeszkadzały myszy dotrzeć do jabłka. Mogą one poruszać się w określonych miejscach labiryntu, przenikać przez ściany lub działać tak, jak to zaplanujesz. Zastanów się również, w jaki sposób mysz powinna reagować na spotkanie z nimi, czy na przykład zakończyć grę, czy wrócić do punktu startowego? A może tylko stracić punkty lub zareagować w inny sposób?

Polecenie 8

Dodaj plansze (nowe tła sceny), które pokazane zostaną, gdy gra się zakończy - zwycięstwem lub przegraną myszy.