Czy można zbudować trójkąt? - gra matematyczna w Scratch

Materiał opisuje tworzenie gry opartej o twierdzenie o nazwie "Nierówność trójkąta" i składa się z sekcji: "Scenariusz gry", "Przygotowujemy pierwszą aplikację", "Projektujemy drugą aplikację", "Sortujemy odcinki według długości", "Znajdujemy dwa najkrótsze odcinki i najdłuższy", "Podsumowanie – szacujemy złożoność obliczeniową", "Zadania uzupełniające".

Materiał zawiera zrzuty ekranu, filmy oraz ćwiczenia.

Czy można zbudować trójkąt? - gra matematyczna w Scratch

Gra będzie się składać z dwóch aplikacji. Zadaniem użytkownika pierwszej z nich jest odpowiedzenie na proste pytanie, czy z trzech odcinków o podanych długościach (wylosowanych przez duszka) można zbudować trójkąt. Duszek powinien sprawdzić odpowiedź i wyświetlić odpowiedni komunikat.

W drugiej aplikacji duszek postawi przed użytkownikiem trudniejsze zadanie. Odcinków będzie więcej, trzeba odpowiedzieć na pytanie, czy z każdych trzech odcinków można zbudować trójkąt. Twoim zadaniem będzie oprogramować działania duszka, który sprawdza odpowiedź użytkownika. Działanie przykładowej aplikacji możesz obejrzeć na poniższym filmie.

Film dostępny pod adresem /preview/resource/RS4TBPErlNR45

Film przedstawiający poprawne działanie skryptu

Źródło: Janusz Wierzbicki, Maciej Borowiecki, licencja: CC BY 3.0.

Przedstawienie działania przykładowej aplikacji. Pojawiają się długości odcinków, a obok nich pytanie w dymku „Czy z każdych trzech odcinków o podanych długościach można zbudować trójkąt?". Na górze znajdują się przyciski „Tak" i „Nie".

> Aby wykonać to ćwiczenie, ważna jest znajomość podstaw obsługi środowiska Scratch oraz wiadomości z matematyki. Jeśli nie masz doświadczenia z tym językiem, zachęcamy do zapoznania się z materiałem Wprowadzenie do programu Scratch.

> Przydatna może się okazać również wiedza z materiałów Od problemu do algorytmu oraz Od algorytmu do programu.

W obu aplikacjach wykorzystywane jest jedno proste tło składające się z dwóch części (w różnych kolorach). W górnej części znajdują się duszki – przyciski, w dolnej duszek, który sprawdza i informuje o poprawności odpowiedzi.

Skorzystaj z poniższego notatnika do zapisania swoich uwag.

> **Ćwiczenie 1**
>
> Przygotuj tło do aplikacji. Może być bardziej rozbudowane graficznie, według twojego pomysłu. Następnie załaduj je do nowego projektu w środowisku Scratch.

> **Ćwiczenie 2**
>
> Przygotuj dwa duszki: przycisk **Tak** oraz przycisk **Nie**. Następnie załaduj je do wcześniej utworzonego projektu. Domyślnie niech przyciski będą ukryte. Utwórz także trzy zmienne do pamiętania długości odcinków.

Przygotowujemy pierwszą aplikację

> **Ćwiczenie 3**
>
> Określ, jakie działania powinny być wykonane po uruchomieniu aplikacji, czyli kliknięciu w zieloną flagę, do momentu kiedy wymagana będzie akcja użytkownika. Przygotuj właściwy skrypt.

Należy zadbać o komunikację pomiędzy duszkami. Duszek odpowiedzialny za sprawdzenie odpowiedzi musi wiedzieć, w który przycisk kliknął użytkownik. W Scratchu do komunikacji pomiędzy duszkami można wykorzystać komunikaty. Więcej informacji o nadawaniu i odbieraniu komunikatów znajdziesz po kliknięciu w poniższą zakładkę.

Nadawanie i odbieranie komunikatów

> **Ćwiczenie 4**
>
> Przygotuj skrypty dla przycisków, które będą nadawały odpowiednie komunikaty.

> **Ćwiczenie 5**
>
> Przygotuj skrypty dla duszka sprawdzającego odpowiedź, reagujące na otrzymanie komunikatów **tak** i **nie**. Zawrzyj w nich komunikat dla duszków - przycisków, informujący o zakończeniu sprawdzania (przycisk należy ukryć).

> **Ćwiczenie 6**
>
> Przygotuj skrypty pokazujące przyciski na początku głównego programu oraz skrypty ukrywające przyciski po otrzymaniu komunikatu **koniec**.

Projektujemy drugą aplikację

Do stworzenia drugiej aplikacji powinieneś posiadać wiedzę z poniższych zagadnień:

1. Tworzenie i wykorzystywanie (np. zamienianie elementów miejscami) list w środowisku Scratch. Informacje na ten temat możesz znaleźć w rozwijanej liście, która znajduje się poniżej.

Listy

2. Definiowanie i wykorzystywanie nowych bloków w środowisku Scratch.

Często bardziej złożony problem warto podzielić na podproblemy i zapisać je w postaci oddzielnych bloków. Więcej informacji na temat tworzenia bloków w Scratch znajdziesz po rozwinięciu poniższej zakładki.

Tworzenie bloków w Scratch

3. Znajdowanie najmniejszego i największego elementu (np. listy).

Więcej informacji na ten temat znajdziesz w ćwiczeniu dodatkowym po rozwinięciu listy.

Ćwiczenie dodatkowe

4. Zamiana elementów listy.

Więcej informacji na ten temat znajdziesz w ćwiczeniu dodatkowym po rozwinięciu listy.

Ćwiczenie dodatkowe

5. Sortowanie elementów (np. listy).

Sortowanie to jeden z podstawowych problemów informatycznych. Istnieje wiele algorytmów sortowania, jedne działają szybciej, inne wolniej. W tym podrozdziale poznasz jedną z nich i wykorzystasz w tworzonej aplikacji. W przyszłości, jeśli będziesz kontynuować swoją przygodę z algorytmami, poznasz także inne i porównasz ich efektywność.

Algorytm sortowania przez wybieranie

> **Ćwiczenie 7**
>
> Utwórz listę do przechowywania długości odcinków (np. o nazwie **Odcinki**) oraz zmienną (np. o nazwie **n**), w której będzie przechowywana liczba losowanych odcinków.

Skrypty dla przycisków będą analogiczne jak w poprzedniej aplikacji. Zmienią się dla duszka sprawdzającego odpowiedź, ponieważ inny będzie algorytm sprawdzenia odpowiedzi użytkownika oraz zastosowane struktury danych (zamiast trzech pojedynczych zmiennych lista).

> **Ćwiczenie 8**
>
> Przygotuj skrypt, uruchamiany po kliknięciu w zieloną flagę, losujący najpierw liczbę odcinków, a potem ich długości.

> **Ćwiczenie 9**
>
> Określ jak sprawdzić, czy z każdych trzech odcinków można zbudować trójkąt. Spróbuj znaleźć więcej niż jedną propozycję rozwiązania tego problemu.

> **Ćwiczenie 10**
>
> Utwórz skrypt (nowy blok) sortujący listę odcinków zgodnie z algorytmem przez wybieranie. Pamiętaj o utworzeniu wszystkich niezbędnych zmiennych.

> **Ćwiczenie 11**
>
> Zmodyfikuj skrypt początkowy oraz skrypty uruchamiane po otrzymaniu komunikatów **tak** i **nie.** Duszek zadający pytanie powinien najpierw posortować wylosowane odcinki.

Znajdujemy dwa najkrótsze odcinki i najdłuższy

Podstawowa wersja gry już działa. W poniższych ćwiczeniach będziemy ją poprawiać, czyli tworzyć inną wersję.

Żeby stwierdzić, czy z każdej trójki odcinków można zbudować trójkąt, nie trzeba porządkować odcinków według ich długości. Wystarczy znaleźć dwa najkrótsze odcinki oraz najdłuższy, czyli dwie liczby najmniejsze i największą. Możesz doprowadzić do sytuacji, że dwie najmniejsze liczby znajdą się na dwóch pierwszych miejscach, a największa na ostatniej. Możesz także znaleźć ich wartości wykorzystując trzy pomocnicze zmienne.

> **Ćwiczenie 12**
>
> Utwórz trzy pomocnicze zmienne, w których zapamiętasz poszukiwane liczby.

> **Ćwiczenie 13**
>
> Zastanów się, jakie nadać wartości początkowe zmiennym **min1**, **min2** i **max**. Przygotuj pomocniczy skrypt nadający im wartości początkowe.

Ćwiczenie 14

Zapisz w postaci listy kroków algorytm znajdowania dwóch wartości najmniejszych i największej.

Do zapisania listy kroków możesz wykorzystać poniższy notatnik.

Notatnik

Ćwiczenie 15

Utwórz nowy blok znajdujący dwie wartości najmniejsze i największą zgodnie z algorytmem z poprzedniego ćwiczenia.

Pamiętaj o dostosowaniu skryptów wywoływanych jako reakcja na nadanie komunikatów **tak** i **nie**, oraz skryptu początkowego.

Zadania uzupełniające

Ćwiczenie 16

Przygotuj aplikację, w której losowana jest lista liczb. Następnie niech duszek poprosi o podanie liczby i sprawdzi, czy ona występuje na liście.

Ćwiczenie 17

Przygotuj podobną aplikację, jak w poprzednim ćwiczeniu, ale po wylosowaniu posortuj listę. Zastanów się, czy wyszukując liczbę w posortowanej liście można zastosować inny algorytm, który wykona mniej operacji.