Fundusze Europejskie
Polska Cyfrowa

Zaprogramuj Przyszłość

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

WPROWADZENIE DO PROGRAMOWANIA Z ZESTAWEM „INTELIGENTNE MIASTO”

SEBASTIAN SZCZEPAŃSKI
FUNDACJA NETCAMP |2018

orange Fundacja

STOWARZYSZENIE MISTRZOWIE KODOWANIA

NETCAMP
Fundacja Rozwoju Branży Internetowej

WSTĘP

„Inteligentne Miasto" zostało stworzone przez Fundację Rozwoju Branży Internetowej w Szczecinie z jej inicjatywy, jako odpowiedź na coraz większe zainteresowanie programowaniem wśród dzieci oraz młodzieży. Początkowo było narzędziem używanym podczas warsztatów pozalekcyjnych. Po ponad trzech latach pracy, niezliczonej liczby młodych testerów i godzin spędzonych nad prototypowaniem i ulepszaniem modeli, w Państwa ręce trafiły modele stworzone dla edukacji.

Ideą powstania zestawu była możliwość połączenia świata wirtualnego (programów) ze światem rzeczywistym – w tym wypadku modeli infrastruktury drogowej i kolejowej. Wybór takich modeli był nieprzypadkowy – miał dotyczyć życia codziennego, nie odstraszać i być zrozumiały. Modele są programowalne tzn. że można nimi sterować za pomocą komputera. Dotyczy to oczywiście wszystkich kolorowych „świateł", ale także czujników, w które są wyposażone modele.

Rozwój technologii druku 3D pozwolił nam na podjęcie wyzwania, jakim było zaprojektowanie, wydrukowanie i złożenie autorskich modeli edukacyjnych. Dodatkowo wykorzystaliśmy rozwiązania Otwartych Rozwiązań tj. Arduino (platforma prototypowania elektroniki) oraz Snap4Arduino (program umożliwiający programowanie modeli z wykorzystaniem języka Scratch).

Staraliśmy się, aby uruchomienie i obsługa modeli była prosta i intuicyjna, a tym samym nie ograniczała dostępu do bardziej zaawansowanych opcji czy ustawień. Obsługa zestawu i radzenie sobie z ewentualnymi problemami to jedno z zagadnień tego przewodnika, ale zasadniczą jego częścią jest wprowadzenie merytoryczne do programowania.

Przewodnik zawiera wszystkie niezbędne pojęcia z podstaw programowania, dodatkowe wskazówki w ich nauczaniu oraz przykładowe zadania wraz z rozwiązaniami. Ważne jest, aby pamiętać, że głównym zadaniem uczniów jest rozwiązanie danego problemu np.: zaprogramowanie świateł drogowych według wskazanej kolejności. Rozwiązania uczniów mogą bardzo różnić się między sobą, nawet od rozwiązania sugerowanego w tym przewodniku. Nie ma znaczenia, w jaki sposób doszli do rozwiązania – najważniejsze jest, że je znaleźli. Możemy oczywiście mówić o optymalnym

rozwiązaniu zadania np. poprzez użycie określonej liczby bloczków, ale powinno to stanowić dodatkowe wyzwanie dla ucznia.

Życzymy dobrej zabawy!
Twórcy „Inteligentnego Miasta”

SPECYFIKACJA ZESTAWU:

W skład ZESTAWU „Inteligentne Miasto” wchodzi:

- 2 kontrolery z gniazdami IDC,
- 14 modeli urządzeń drogowych:
  - model sygnalizatora drogowego (4 sztuki),
  - model sygnalizatora kolejowego (2 sztuki),
  - model sygnalizatora pieszego (2 sztuki),
  - model szlabanu kolejowego (2 sztuki),
  - model lampy ulicznej (2 sztuki),
  - model czujnika przejazdu (2 sztuki),
- 12 podstawek magnetycznych do modeli,
- 2 kable USB A - USB B,
- 8 kabli IDC,
- 1 opakowanie zbiorcze.

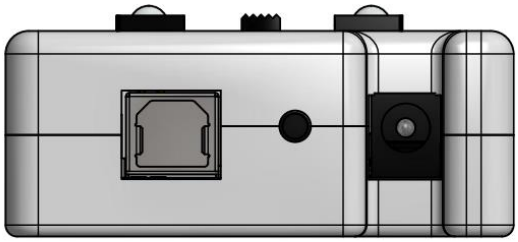
Zestaw umożliwia niezależnie pracę przy dwóch stanowiskach komputerowych. Poniżej został omówiony skład zestawu oraz zastosowanie poszczególnych elementów.

- Kontroler z gniazdami IDC (Rysunek 1) – urządzenie służące do komunikacji między komputerem a modelami. Jego zasadniczą częścią jest Arduino UNO oraz złącza IDC z możliwością podłączenia do 4 modeli. Kontroler wymaga zasilania z komputera lub w wersji przenośnej z powerbanka (w tej wersji działają tylko gotowe programy wgrane wcześniej do pamięci kontrolera). Urządzenie może działać w różnych trybach, które poznamy w dalszej części tego przewodnika.

- Model sygnalizatora drogowego – model posiada trzy diody możliwe do kontrolowania: czerwoną, żółtą i zieloną, jako jedyny może być podłączony w 2 sztukach sterowanych niezależnie
- Model sygnalizatora kolejowego – posiada dwie identyczne diody koloru czerwonego oraz krzyż świętego Andrzeja ostrzegający o jednotorowym przejeździe kolejowym.
- Model sygnalizatora pieszego – posiada dwie diody, jedną koloru czerwonego i jedną zielonego oraz przycisk (umieszczony dla wygody w górnej części modelu) naśladujący przycisk umieszczony w prawdziwych sygnalizatorach dla pieszych, wciskany przez nich z zamiarem przekroczenia przejścia.
- Model szlabanu kolejowego – kolejny model przejazdu kolejowego. W jego środku wbudowany jest mechanizm obracający jego ramię. Istnieje możliwość zamknięcia szlabanu w lewą lub prawą stronę bądź jego otwarcia. Ramię szlabanu jest przyczepiane w jednoznacznej pozycji za pomocą magnesów, dzięki czemu niemożliwe jest jego urwanie podczas pracy szlabanu. Dodatkowo w szlabanie wmontowany jest buzzer – mały głośniczek, który może być zaprogramowany do nadawania ostrzegawczego sygnału dźwiękowego.
- Model lampy ulicznej – zawiera jasne diody koloru białego. Model można uruchomić w tzw. „trybie energooszczędnym”, w którym świecić będzie się tylko połowa diod. Dodatkowo każda lampa wyposażona jest w czujnik światła umieszczony w górnej części obudowy – służy on do określania poziomu natężenia światła w pomieszczeniu. Jego zakrycie dłonią lub faktyczne wyłączenie źródeł światła w pomieszczeniu powoduje wykrycie zmniejszonej liczby natężenia światła. W przypadku bardzo oświetlonych luz zacienionych pomieszczeń czujnik ten może mieć problemy z poprawnym odczytem.
- Model czujnika przejazdu – wykrywa obiekty poruszające się przez niego za pomocą niewidocznej wiązki światła. Model wykrywa zarówno modele samochodów i kolejki, jak i przecięcie wiązki dłonią.
- Podstawki magnetyczne do modeli – podstawki zwiększające ciężar modelu i jego stabilność. Łączą się z modelami za pomocą wbudowanego magnesu. Podstawki są przygotowane dla wszystkich modeli z wyjątkiem czujnika przejazdu.
- Kabel USB A - USB B – kabel połączeniowy między komputerem a kontrolerem.

- Kabel IDC – przewód połączeniowy pomiędzy modelami a kontrolerem. Do każdego modelu wymagany jest jeden przewód podłączany do złączy umieszczonych w modelach.
- Opakowanie zbiorcze – zawiera naklejkę z wykazem wszystkich elementów i ich liczebności.

Rysunek 1: Kontroler zestawu:



TRYBY DZIAŁANIA KONTROLERA

Do zmieniania trybów służy przełącznik umieszczony w kontrolerze. Ten może działać w trzech różnych trybach, ale jednocześnie tylko w jednym.

- Tryb programowania z komputerem (kolor niebieski diody) – włączamy w przypadku podłączenia do komputera i chęci sterowania zestawem według własnego uznania.

- Tryb „demo” (kolor czerwony diody) – modele działają w zaprogramowany sposób niezależnie od oprogramowania komputerowego. W tym trybie zestaw może działać podłączony za pomocą kabla USB do ładowarki telefonicznej lub powerbanka (brak w zestawie). Tryb ten jest pomocny podczas prezentacji lub pokazów dla uczniów.
- Tryb „testowy” (kolor zielony diody) – dodatkowy tryb, w którym możemy przetestować poprawność działania modeli.

TRYB TESTOWY

Tryb testowy zawiera specjalny program sprawdzający wszystkie elementy modeli. Wystarczy podłączyć testowany model lub modele do kontrolera w tym trybie w dowolne złącze. Tryb testowy uruchamia modele według zasad:

- model sygnalizator drogowego – wszystkie diody powinny na zmianę: świecić jednocześnie oraz jednocześnie gasnąć,
- model sygnalizatora kolejowego – dwie diody czerwone na zmianę: świecą jednocześnie i gasną,
- model sygnalizatora pieszego – dioda zielona i czerwona na zmianę: świecą jednocześnie i gasną jednocześnie, zaś po wciśnięciu przycisku świecą się bez gaśnięcia,
- model szlabanu kolejowego – działa w schemacie: zamknij w lewo, otwórz szlaban, zamknij w prawo, otwórz szlaban, dodatkowo podczas otwierania się szlabanu powinien być słyszalny dźwięk z buzzera (głośniczka),
- model lampy ulicznej – diody świecą się i gasną na zmianę, jeśli zaś zasłonimy czujnik światła to lampa powinna świecić się światłem ciągłym,
- model czujnika przejazdu – można go sprawdzić podłączając go jednocześnie z modelem szlabanu kolejowego, w takim scenariuszu na czas, gdy umieścimy przeszkodę (np. dłoń) w czujniku, szlaban kolejowy powinien zatrzymać ruch ramieniem.

Ważne jest zwrócenie uwagi na poprawne nazewnictwo modeli. Dla przykładu model sygnalizatora przejazdu kolejowego często przez uczniów nazywany jest błędnie semaforem – który oczywiście jest innym urządzeniem używanym w ruchu kolejowym, a nie drogowym. Podczas pierwszego kontaktu uczniów

z modelami edukacyjnymi warto zrobić krótkie ich „przedstawienie” i opisać ich podstawowe funkcje.

Uruchomienie trybu testowego zalecamy w przypadku uszkodzenia, bądź podejrzenia wadliwego działania elektroniki w modelach. Dla przykładu można sprawdzić czy któraś z diod nie przepaliła się. Tu warto dodać, że żywotność zamontowanych diod to średnio 25 lat.

OPROGRAMOWANIE „CYFROWY ODKRYWCA”

Elementem niezbędnym do obsługi zestawu „Inteligentne Miasto” jest oprogramowanie pod nazwą „Cyfrowy Odkrywca”. Jest to program działający na zasadzie popularnego Scratcha zawierający dodatkowo specjalne bloki do sterowania modelami. Oprogramowanie jest bezpłatne i dostępne pod adresem http://cyfrowyodkrywca.pl/InteligentneMiasto/, w zakładce Oprogramowanie.

Należy pobrać plik instalatora z linku wymienionego wcześniej, a następnie go uruchomić i zainstalować w wybranym miejscu na komputerze (standardowa instalacja, jak w przypadku innych popularnych programów komputerowych). Dodatkowo podczas instalacji automatycznie wgrywają się wszystkie niezbędne sterowniki.

Dostęp do Internetu jest wymagany tylko podczas pobierania programu, w trakcie korzystania z niego takie połączenie jest zbędne. Oprogramowanie Cyfrowy Odkrywca dostępne jest na systemy Windows, Linux oraz MacOS.

Oprogramowania zostało stworzone na bazie projektu Snap4Ardunino i nie wymaga szczególnie wysokich paramentów wyposażenia komputera.

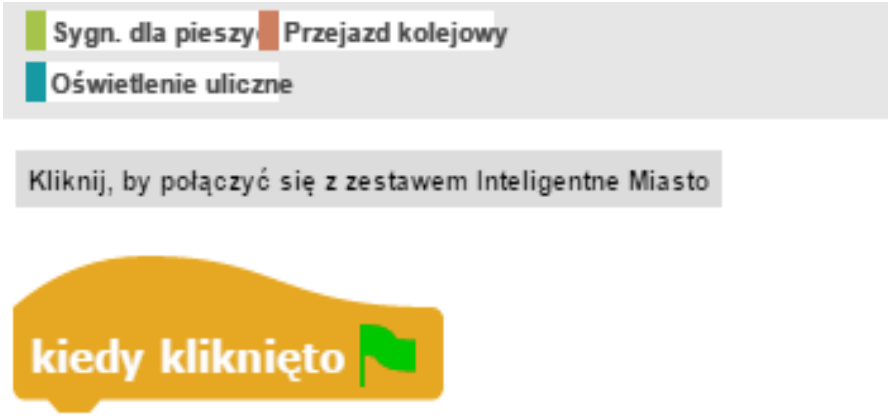
URUCHOMIENIE ZESTAWU

Aby uruchomić zestaw należy:

1. uruchomić oprogramowanie „Cyfrowy Odkrywca”,
2. podłączyć kontroler kablem USB do komputera,
3. ustawić tryb kontrolera na „programowanie z pomocą komputera” (Rysunek 1) – tak, aby prawa dioda świeciła się kolorem niebieskim,

4. kliknąć na przycisk „Kliknij, by połączyć się z zestawem Inteligentne Miasto” dostępny w kategorii „Start” (Rysunek 2),
5. wybrać port oznaczony nazwą „Inteligentne Miasto” (Rysunek 3),
6. podłączyć wybrane modele i sterować własnym miastem w oprogramowaniu „Cyfrowy Odkrywca”.

Rysunek 2: Przycisk połączenia z zestawem:



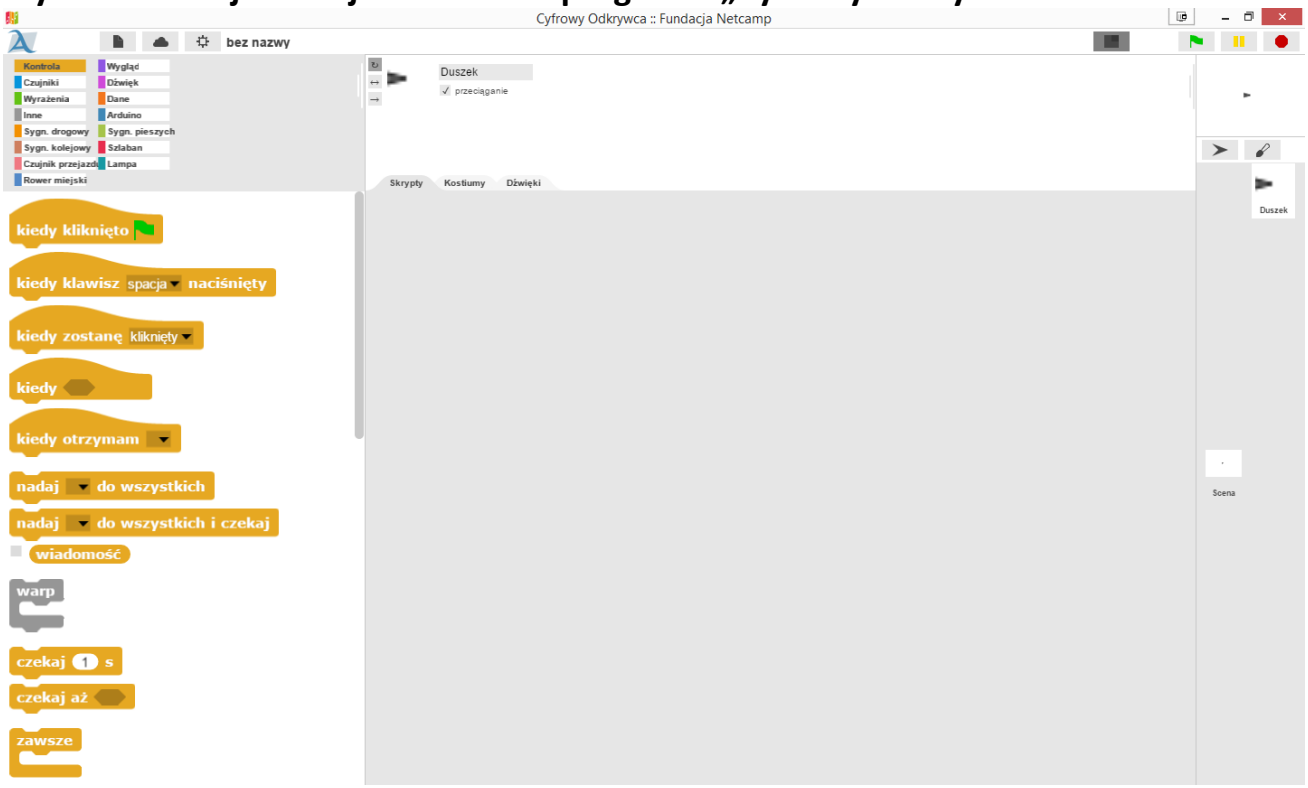
Rysunek 3: Przycisk połączenia z zestawem:

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: BLOCZKI

Oprogramowanie „Cyfrowy Odkrywca”, obsługując modele „Inteligentnego Miasta”, zostało stworzone na bazie języka Scratch. Jest to rodzina wizualnych języków programowania przeznaczonych głównie dla dzieci i młodzieży. Głównymi wymogami obsługi programu są: umiejętność liczenia, umiejętność czytania ze zrozumieniem oraz obsługa komputera na podstawowym poziomie.

Zasadniczą częścią budowy programu są tzw. „bloczki” – graficzne elementy, którym przypisane są pewne instrukcje komputera. Bloczek może np. sterować sygnalizatorem, wykonywać obliczenia lub sterować działaniem programu (Rysunek 4).

Rysunek 4: Najważniejsze składniki programu „Cyfrowy Odkrywca”:



W programie „Cyfrowy Odkrywca” bloczki podzielono na 13 różnych kategorii (Rysunek 4). Bloczki każdej z kategorii mają swój wyznaczony kolor. 9 kategorii jest standardowymi występującymi w językach typy Scrach:

- kategoria „Start” – zawiera bloczki rozpoczynające i kończące programy,
- kategoria „Kontrola” – w tej kategorii znajdują się bloczki sterujące programami,
- kategoria „Wygląd” – jednoczy bloczki sterujący duszkami Scratcha,
- kategoria „Czujniki” – zawiera bloczki wczytujące dane z tzw. „czujników” (np. klawiatury),
- kategoria „Dźwięk” – bloczki do obsługi dźwięków,
- kategoria „Wyrażenia” – głównie bloczki do operacji matematycznych, logicznych i operacji na tekstach,
- kategoria „Dane” – kategoria do tworzenia zmiennych i list (przydaje się w nauce programowania w kolejnych etapach edukacji),
- kategoria „Inne” – dodatkowa kategoria, w której możemy umieścić własne bloczki,

- kategoria „Arduino” – zaawansowana kategoria służąca do obsługi bezpośredniej mikrokontrolera Arduino Uno.

Dodatkowo 4 kategorie zawierają bloczki do sterowania modelami „Inteligentnego Miasta”:

- kategoria „Sygnalizator drogowy” – zawiera bloczki sterujące sygnalizatorami drogowymi; do jednego mikrokontrolera można podłączyć jednocześnie dwa sygnalizatory drogowe,
- kategoria „Sygnalizator dla pieszych” – kategoria bloczków obsługująca sygnalizator drogowy,
- kategoria „Przejazd kolejowy” – skupia wszystkie modele przejazdu kolejowego: szlaban kolejowy, sygnalizator przejazdu kolejowego oraz czujnik przejazdu,
- kategoria „Oświetlenie uliczne” – zawiera bloczki obsługujące model lampy ulicznej.

W dalszej części przewodnika omówimy najważniejsze i najczęściej używane bloczki wybranych kategorii. Warto zaznaczyć, że istnieją 3 różne kategorie bloczków różniących się kształtem (Rysunek 5):

Rysunek 5: Rodzaje bloczków:

Wyróżnione rodzaje bloczków różnią się nie tylko kształtem, ale przede wszystkim funkcjami:

- bloczki komend – wywołują daną komendę np.: włącz światło czerwone dla pieszych w modelu sygnalizatora dla pieszych,
- bloczki funkcji – wykonują pewne obliczenia lub operacje, które zwracają określony wynik, np.: funkcja zaokrąglij przyjmuje daną liczbę i zwraca jej zaokrąglenie; w celu zwrócenia wyniku klikamy na bloczek funkcji,

- bloczki predykatu – zwracają „prawdę” lub „fałsz”; dla przykładu bloczek „Jedzie pociąg” pokazuje wartość „prawda” w przypadku wykrycia ruchu w modelu czujnika przejazdu.

Bloczki można w łatwy sposób powielać lub usuwać. Powielenie bloczka może nastąpić poprzez ponowne wybranie bloczka z danej kategorii lub kliknięcie menu kontekstowego bloczka (prawy przycisk myszy) i wybranie opcji „powiel”; zaś usunięcie bloczka możemy wykonać poprzez odłożenie go w dowolną kategorię bloczków albo wybranie opcji „usuń” menu kontekstowego bloczku.

W razie potrzeby wielkość bloczków można powiększyć używając opcji „Powiększ bloki” znajdującej się w menu ustawień (ikona zębatki). Standardowy rozmiar bloczków przy uruchomieniu programu to 1,5. Bloczki dodatkowo można kopiować oraz usuwać.

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: SKRYPTY

Bloczki można łączyć ze sobą w większe moduły zwane skryptami. Warto podkreślić, że nie wszystkie bloczki da się ze sobą połączyć. Bloczki łączymy na zasadzie łączenia puzzli w największym obszarze programu (Rysunek 4). Skrypt złożony jest z kilku bloczków i jest stworzony do wykonywania jednego lub więcej zadań jednocześnie. Przykładowy skrypt może sterować jednym modelem, ale możne też sterować działaniem kilku modeli na raz (Rysunek 6).

Rysunek 6: Przykładowe skrypty

Często pojęcie skryptu jest używane ze słowem „program”. Należy rozróżnić pojedynczy skrypt od programu. Program to zbiór większej ilości skryptów

całościowo określających działania komputera w sposób przez nas wcześniej zaplanowany i stworzony – zaprogramowany. W naszym przypadku pojedynczy skrypt może zamykać szlaban albo sterować światłami sygnalizatora kolejowego (Rysunek 6), zaś program może zawierać kilka skryptów określających działanie wszystkich modeli przejazdu kolejowego w różnych warunkach.

W przypadku tworzenia większej ilości skryptów lub skryptów zawierających dużą liczbę bloczków, obszar roboczy automatycznie powiększy się do pożądanych rozmiarów. Dodatkowo mogą pojawić się paski przewijania w dół lub w prawo.

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: ZASADY BUDOWANIA SKRYPTÓW

Zasadniczo każdy skrypt powinien posiadać początek i koniec. Do rozpoczęcia i zakończenia skryptu najczęściej używa się bloczków kategorii „Start”:

Klocek	Opis
	Bloczek uruchamiający skrypt po jego kliknięciu prawym przyciskiem myszki. Każde kliknięcie powoduje jego ponowne uruchomienie. Jest to najprostszy i najczęściej wykorzystywany bloczek do startowania skryptu.
	Ten bloczek uruchamia skrypt po naciśnięciu określonego klawisza klawiatury. Standardowo jest to klawisz „spacja”, może on być oczywiście dowolnie zmieniony.
	Uruchamia skrypt w przypadku spełnienie pewnego warunku – np. w przypadku przejazdu zabawkowej kolejki przez czujnik przejazdu (Rysunek 6).

Używanie bloczków kończących skrypty jest dobrowolne i zazwyczaj pomijane. Warto zaznaczyć iż pojedyncze bloczki możemy uruchamiać bezpośrednio przez kliknięcie na dany bloczek bez potrzeby używania bloczków startowych. Jest to szczególnie przydatne, gdy chcemy sprawdzić działanie modelu.

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: STEROWANIE SKRYPTAMI

Działaniem skryptów można sterować używając bloczków kategorii kontrola. Dodatkowo można rozróżnić skrypty, które wykonają się raz, kilka razy albo będą działać do momentu ich przerwania. Skrypt widoczny po lewej stronie (Rysunek 7) po uruchomieniu wykona się tylko raz - włączy lewe światło sygnalizatora kolejowego na jedną sekundę i następnie włączy prawe światło sygnalizatora kolejowego na taki sam czas, zaś skrypt po prawej stronie (Rysunek 7) będzie po jego uruchomieniu sterować światła sygnalizatora kolejowego do czasu wyłączenia skryptu.

Rysunek 7: Wykonywalność skryptów:

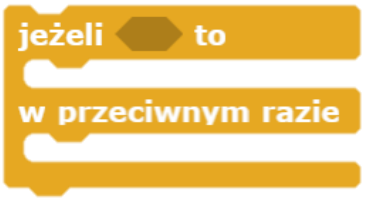
Jak zatem wyłączyć skrypt i rozróżnić skrypt aktywny od nieaktywnego? Różnice widać na kolejnym rysunku (Rysunek 8).

Rysunek 8: Wykonywalność skryptów:

Pamiętamy, że skrypt uruchamiamy poprzez pojedyncze kliknięcie myszą na blok początkowy. Ponowne kliknięcie na blok startowy spowoduje przerwanie działania skryptu. Dla rozróżnienia skryptu aktywnego (prawy skrypt, Rysunek 8) od nieaktywnego (lewy skrypt, Rysunek 8) służy jasnozielona ramka otaczająca skrypt. Tylko skrypt aktualnie działający posiada taką ramkę. W jednym czasie może działać jednocześnie kilka skryptów.

Ilość powtórzeń działania skryptu, przerwy w jego działaniu czy wykonywanie tylko jego określonych części możemy determinować za pomocą bloczków kategorii „Kontrola”:

Klocek	Opis
czekaj 1 s	Ten bloczek wstrzymuje działanie skryptu na 1 sekundę. Czas ten może być zmieniony przez użytkownika również na liczby ułamkowe z zastosowaniem znaku ”.” oddzielającym części całkowite od ułamkowych.
czekaj aż	Bloczek wstrzymuje działanie skryptu do czasu zaistnienia pewnego warunku (predykat zwróci wartość „prawda”). Dla przykładu skrypt może czekać aż w pomieszczeniu będzie ciemno – używając bloczku „Jest ciemno”.
powtórz 10 razy	Bloczek powtarza wszystkie komendy znajdujące się w jego wnętrzu określoną liczbę razy. Najczęściej popełnianym błędem przez uczniów jest nieumieszczanie komend w jego wnętrzu (Rysunek 9).
zawsze	Bloczek powtarza działanie wszystkich komend w jego wnętrzu do czasu przerwania działania skryptu.
jeżeli to	Bloczek uruchamia komendy w środku bloku jeśli zostanie spełniony konkretny warunek. Jest to tzw. „instrukcja warunkowa”.

Klocek	Opis
	Rodzaj rozbudowanej instrukcji warunkowej wykonującej ten lub inny zestaw komend w zależności od spełnienia albo nie spełnienia określonego warunku.

Rysunek poniżej (Rysunek 9) przedstawia dwa podobne skrypty z czego ten widoczny po stronie lewej posiad kilka błędów, zaś po stronie prawej jest wykonany prawidłowo. Skrypt miał za zadanie 10 razy zmienić kolory świateł na sygnalizatorze dla pieszych na czas 1 sekundy.

Rysunek 9: Poprawność tworzenia skryptów:

W pierwszym skrypcie popełniono następujące błędy:

1. bloczki „kiedy kliknięto” i „powtórz 10 razy” nie są ze sobą połączone, co w tym wypadku przerwie działanie skryptu na samym początku,
2. bloczki sterujące modelem światłami sygnalizatora pieszego nie są umieszczone w środku bloczka „powtórz 10 razy”, co spowoduje ich wykonanie tylko jeden raz,
3. światło czerwone sygnalizatora nie zostaje wyłączone, co spowoduje, że w jednym czasie mogą świecić się dwa kolory świateł jednocześnie, co jest niezgodne z treścią zadania. Warto zaznaczyć że podobnie jak w życiu codziennym – jeśli pozostawimy włączone światło przed wyjściem, będzie ono zapalone do czasu dopóki ktoś go nie wyłączy. Podobnie wszystkie diody modeli będą się świecić do czasu kiedy nie wykonamy komendy „wyłącz” określone światło.

4. Działanie światła zielonego nie zostało określone czasowo, dlatego skrypt niezwłocznie przejdzie do wykonywania kolejnej komendy. W naszym wypadku jest to koniec działania skryptu – nasz skrypt pozostawi włączone światło zielone wraz z czerwonym.

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: WŁASNE BLOCZKI

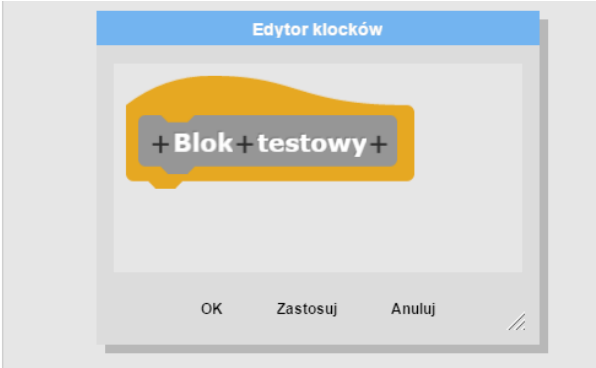
W programie „Cyfrowy Odkrywca” możemy tworzyć własne bloczki, które jeśli nie określimy innej kategorii, znajdą się domyślnie w kategorii „Inne”.

Aby stworzyć własny bloczek klikamy przycisk „nowy blok” widoczny w kategorii „Inne” (Rysunek 10):

Rysunek 10: Tworzenie nowego bloku:

Możemy nadać dowolną nazwę nowego bloczku, wybrać jego rodzaj oraz przypisać go do wybranej kategorii. Nowy bloczek powinien pojawić się w wybranej kategorii albo w kategorii „Inne” (Rysunek 11):

Rysunek 11: Edytor klocków:



Po określeniu podstawowych parametrów nowego bloku, w tym przykładzie bloku o nazwie „Blok tekstowy”, powinno pojawić się okno Edytora klocków, w którym możemy określić, co będzie wykonywać nowostworzony bloczek. Robimy to poprzez dodawanie innych bloczków tak jakbyśmy tworzyli nowy skrypt. Od tej pory blok testowy będzie działał jak skrypt i wykonywał instrukcje w nim określone (Rysunek 12). W każdej chwili możemy wrócić do ponownej edycji utworzonego przez nas bloczka wybierając opcje „edytuj” menu kontekstowego.

Rysunek 12: Instrukcje bloku testowego:

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: INSTRUKCJA WARUNKOWA

Jedną z podstawowych instrukcji w językach programowania jest instrukcja „Jeżeli…”. Można tworzyć skrypty wykonujące komendy jedną po drugiej, ale w przypadku, gdy chcemy tworzyć skrypty, które zachowają się różnie zależnie od warunków, wykorzystanie instrukcji warunkowej jest prostsze i bardziej czytelne.

Dla przykładu gdybyśmy chcieli stworzyć prosty program włączający automatycznie model lampy ulicznej po zakryciu czujnika natężenia światła, moglibyśmy stworzyć dwa oddzielne skrypty, które wykonają to zadanie, ale gdyby dołożyć dodatkowe warunki ilość, skryptów mogłaby się mnożyć i przestać być czytelna. W przypadku, gdy mamy coś wykonać pod jakimś warunkiem, idealnie nadają się do tego bloczki „Jeżeli”. W zadaniu potrzebujemy minimum 6 bloczków (Rysunek 13).

Rysunek 13: Bloczki składowe skryptu:

- Komenda „kiedy kliknięto” - bloczek startowy rozpoczynający skrypt.
- Komenda „zawsze” – spowoduje, że nasz skrypt będzie działał ciągle i cały czas sprawdzał czy poziom natężenia się zmienia.
- Komendę „jeżeli … to, w przeciwnym wypadku …” – instrukcja mówiąca, który zestaw komend ma się w danym momencie wykonać w zależności od warunku.
- Komenda „Włącz lampę” i „Wyłącz lampę” – steruje modelem oświetlenia ulicznego.
- Predykat „Jest ciemno” – zwraca „prawdę” w przypadku niskiego natężenia światła albo „fałsz” w przypadku wysokiego natężenia światła w pomieszczeniu.

Cały skrypt złożony z tych bloczków wygląda następująco (Rysunek 14).

Rysunek 14: Przykładowy skrypt:

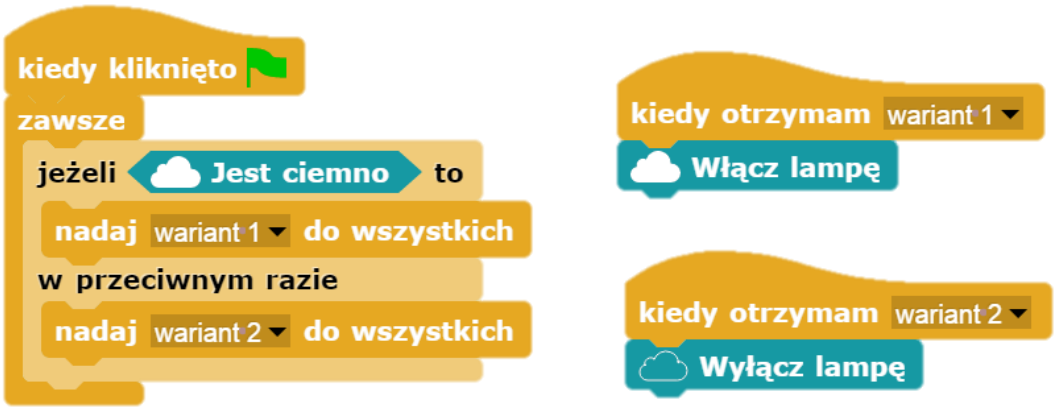
Warto zaznaczyć, że skrypt musi zostać aktywowany (musi posiadać jasnozieloną obwódkę). Dodatkowo działa bloczek „zawsze”, który sprawdza cały czas natężenie światła poprzez czujnik umieszczony w modelu lampy ulicznej i w zależności od tego wykona komendę „Włącz lampę” albo „Wyłącz lampę”.

PODSTAWY PROGRAMU CYFROWY ODKRYWCA: WSKAZÓWKI

Oprogramowanie „Cyfrowy Odkrywca” jak każdy program nie jest wolny od błędów, dlatego zalecamy zapisywanie co jakiś czas pracy z użyciem opcji „Zapisz” lub „Zapisz jako”. Całość zapisze się nam wewnątrz oprogramowania, jeśli chcemy zapisać cały projekt jako plik musimy użyć opcji ”Eksportuj projekt” dla jego zapisu, oraz opcji „Importuj” dla wczytania projektu.

W przypadku, gdy stworzony skrypt ze względu na swoją długość traci przejrzystość – tzn. korzystamy z dużej ilości bloczków i nie jesteśmy w stanie w łatwy sposób przewidzieć działania takiego skryptu, możemy skorzystać z jednej z podstawowych strategii programistycznej „dziel i zwyciężaj”. W tej strategii dzielimy złożone skrypty na możliwie mniejsze, obsługujące np. maksymalnie jeden model. Oczywiście w takim przypadku dochodzi potrzeba komunikacji pomiędzy skryptami. Do tego celu służą specjalne bloczki (Rysunek 15):

Rysunek 15: Komunikacja między skryptami:



Bloczki „nadaj … do wszystkich” oraz „kiedy otrzymam…” potrafią komunikować się miedzy sobą, przy czym warto podkreślić, że komunikacja taka jest jednostronna. Bloczek „nadaj” wysyła zdefiniowaną przez nas wiadomość, a wszystkie bloczki „kiedy otrzymam” przyjmują wiadomości. Oczywiście tylko bloczki, które „czekają” na daną wiadomość uruchomią się po jej otrzymaniu.

Mamy nadzieję że niniejszy Przewodnik wyczerpująco wprowadza w temat podstaw programowania i będzie pomocny podczas zabawy z Inteligentnym Miastem. Ponad to stworzyliśmy dodatek (Dodatek A) zawierający wybrane zadania wraz z rozwiązaniami. Zadania były tak dobierane aby przypominały sytuacje z życia codziennego.

DODATEK A

Zadanie 1: Szlaban sterowany ręcznie

Stwórz szlaban sterowany za pomocą klawiatury komputera, który na po naciśnięciu klawisza "strzałka w górę" otwiera ramię, klawisza "strzałka w prawo" zamyka ramię prawo, zaś "strzałka w lewo" zamyka ramię w lewo.

Zadanie 2: Światła startowe

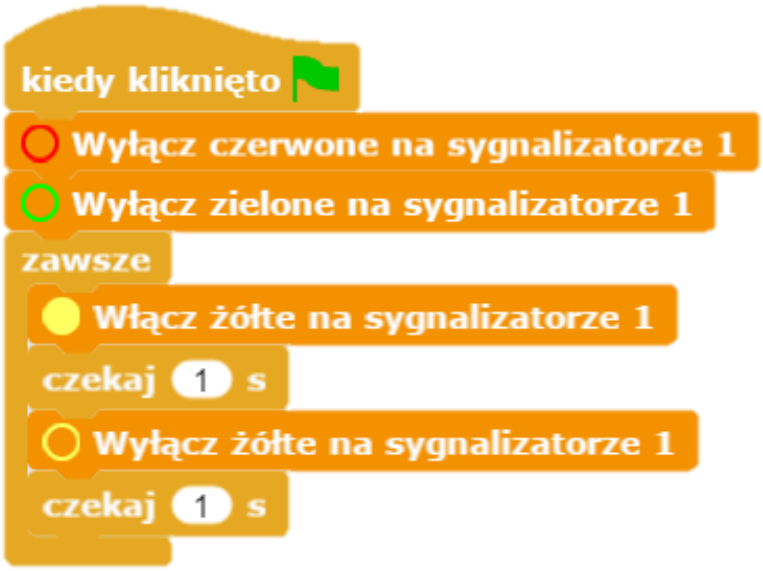
Czy wiesz jak wyglądają światła startowe na wyścigach? Wykorzystaj sygnalizator drogowy. Włącz światło zielone, po 3 sekundach włącz dodatkowo światło żółte, a po kolejnych 3 sekundach światło zielone. Zapalone wszystkie światła oznaczają sygnał startu.

Zadanie 3: Energooszczędne oświetlenie uliczne

Ustaw lampę tak aby włączała się lewy przycisk myszki, a po 6 sekundach bezczynności, przechodziła w tryb energooszczędny.

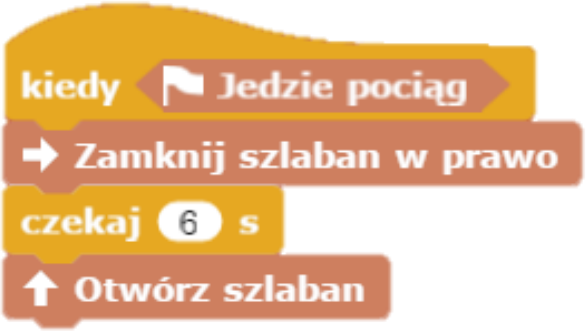
Zadanie 4: Awaria świateł

Do sygnalizowania awarii świateł ulicznych używane jest migające światło żółte . Stwórz sygnalizator drogowy z migającym światłem żółtym co 1 sekundę.



Zadanie 5: Automatyczny szlaban

Zaprogramuj szlaban tak, aby podczas przejazdu pociągu zamykał się na 6 sekund. Wykorzystaj model czujnika przejazdu.



Zadanie 6: Przejście dla pieszych

Stwórz przejście dla pieszych z sygnalizatorem, który będzie zmieniać światła co 6 sekund.

Zadanie 7: Sygnalizator przejazdu kolejowego

Zadaniem kolejnego skryptu będzie włączenie świateł sygnalizatora przejazdu kolejowego, zmieniającego światła przez 10 sekund. Dodatkowo można wykorzystać dźwięk ostrzegawczy z głośnika umieszczonego w szlabanie kolejowym.

Zadanie 8: Przejście dla pieszych z przyciskiem

Stwórz przejście dla pieszych wykorzystując sygnalizator dla pieszych. Wykorzystaj przycisk dla pieszych i ustaw czas przejścia na 6 sekund

```
kiedy [ ] Przycisk dla pieszych wciśnięty
Włącz zielone dla pieszych
Wyłącz czerwone dla pieszych
czekaj 6 s
Włącz czerwone dla pieszych
Wyłącz zielone dla pieszych
```

Zadanie 9: Automatyczny przejazd kolejowy

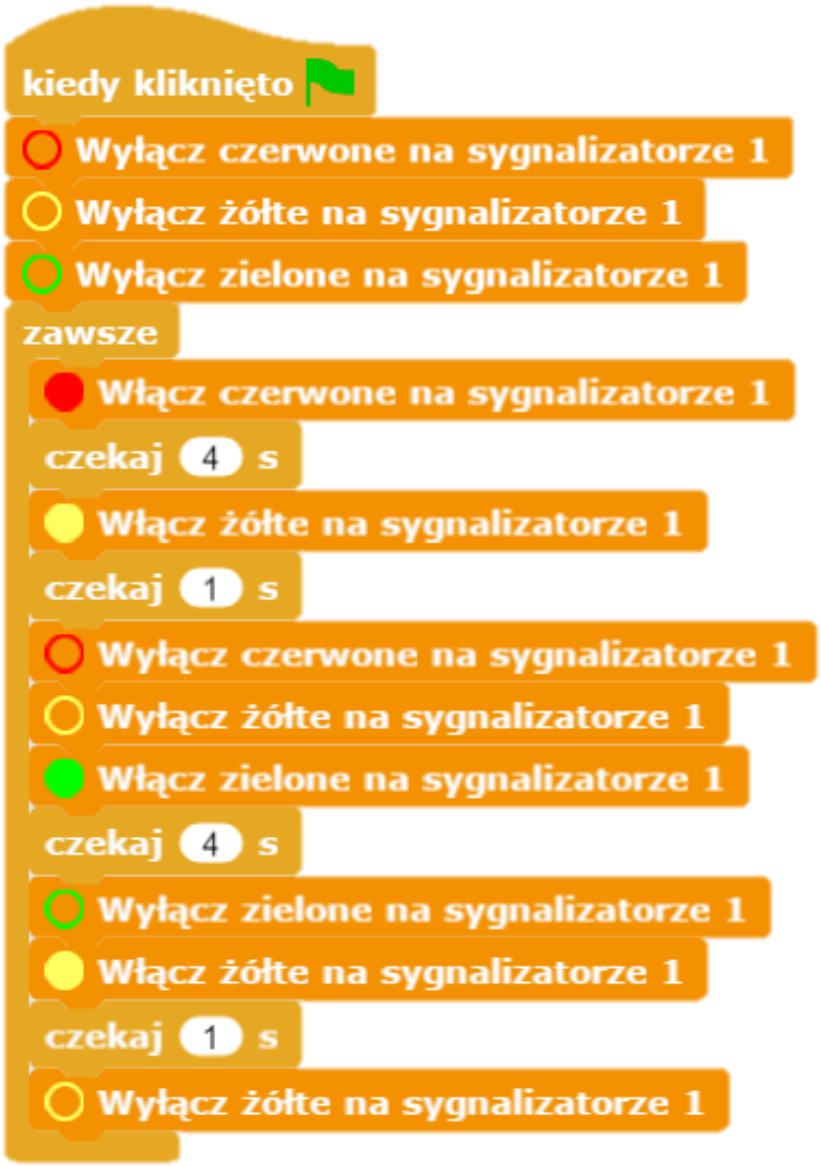
Zbuduj automatyczny przejazd kolejowy. Użyj modeli przejazdu kolejowego oraz dobierz odpowiednie czasy. Możesz użyć dwóch połączonych skryptów.

```
kiedy [⚑] Jedzie pociąg
nadaj uruchom sygnalizator kolejowy do wszystkich
Zamknij szlaban w prawo
czekaj 6 s
Otwórz szlaban
```

```
kiedy otrzymam uruchom sygnalizator kolejowy
powtórz 6 razy
    Włącz lewe światło kolejowe
    Wyłącz prawe światło kolejowe
    czekaj 0.5 s
    Włącz prawe światło kolejowe
    Wyłącz lewe światło kolejowe
    czekaj 0.5 s
Wyłącz lewe światło kolejowe
Wyłącz prawe światło kolejowe
```

Zadanie 10: Automatyczny sygnalizator drogowy

Stwórz sygnalizator drogowy działający w sekwencji: światło czerwone (4s), światło czerwone i żółte (1s), światło zielone (4s), światło żółte (1s), działające w pętli aż do zatrzymania skryptu.



Zaprogramuj Przyszłość

„**Zaprogramuj przyszłość”** to kompleksowy program edukacji cyfrowej skierowany do 222 szkół, 1 200 nauczycieli edukacji wczesnoszkolnej i 450 edukatorów z gmin wiejskich i wiejsko-miejskich wybranych powiatów w całej Polsce. Celem programu jest podniesienie kompetencji cyfrowych nauczycieli i edukatorów, w szczególności wyposażenie ich w kompetencje programistyczne oraz wiedzę i umiejętności dotyczące metodyki nauczania programowania najmłodszych uczniów. Pozwoli im to na prowadzenie ciekawych zajęć z programowania.

Doświadczeni trenerzy przygotują nauczycieli i edukatorów do prowadzenia lekcji z programowania. Nauczyciele poprowadzą takie zajęcia dla uczniów klasa 1-3. Szkoły otrzymają zestawy do prowadzenia atrakcyjnych zajęć z nowych technologii, m.in. roboty, maty edukacyjne oraz tablety.

Zapraszamy do skorzystania ze scenariuszy zajęć z programowania i innych materiałów dydaktycznych,dostępnych na platformie edukacyjnejzaprogramujprzyszlosc.edu.pl, pełniącej rolę bazy wiedzy i forum wymiany doświadczeń.

Program to inicjatywa Fundacji Orange, Stowarzyszenia Mistrzowie Kodowania i Fundacji Netcamp.

orange Fundacja

STOWARZYSZENIE MISTRZOWIE KODOWANIA

NETCAMP Fundacja Rozwoju Branży Internetowej

Fundusze Europejskie Polska Cyfrowa

Unia Europejska Europejski Fundusz Rozwoju Regionalnego

Projekt współfinansowany ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Polska Cyfrowa