



CyberPi
Kurs podstawowy

1. Podejście pedagogiczne

(Krótki opis podejścia do nauczania opartego na zapytaniach i podejścia opartego na doświadczeniach praktycznych)

Makeblock Education zapewnia rozwój wiedzy i umiejętności poprzez połączenie cyfrowych i fizycznych narzędzi, które dzięki wzajemnej współpracy umożliwiają pełne i praktyczne doświadczenie edukacyjne. Uczniowie aktywnie angażują się w lekcje, skupiając się na rozwiązywaniu określonych problemów lub tworzeniu nowych rozwiązań. Umożliwia to rozwój teoretycznych koncepcji na podstawie praktycznych doświadczeń, jak również ponowne stosowanie tych koncepcji w ramach praktycznych zadań, pogłębiając tym samym zrozumienie tematu. Proces zdobywania wiedzy obejmuje uczenie się na błędach poprzez szczegółową i systematyczną analizę procesów.

Dzięki praktycznemu podejściu do nauki uczniowie mogą brać bardziej aktywny udział w zajęciach zamiast ograniczania się do słuchania wykładów.

2. Wprowadzenie do CyberPi

(Krótki opis możliwości zastosowania urządzenia podczas lekcji)

CyberPi to zaawansowany mikrokontroler zaprojektowany przez Makeblock Education do zastosowania podczas lekcji informatyki i lekcji wykorzystujących podejście STEAM.

Możliwości urządzenia pozwalają na prowadzenie interaktywnych, ciekawych zajęć. Dla przykładu, kilka urządzeń CyberPi można połączyć za pomocą bezprzewodowej sieci lokalnej (LAN) w obrębie klasy, aby przeprowadzić interaktywny quiz lub wymieniać informacje między uczniami. CyberPi mogą również łączyć się z Internetem w celu rozpoznawania mowy, zbierania danych lub ich przesyłania za pomocą arkuszy kalkulacyjnych Google.

mBlock5 jest platformą do kodowania CyberPi, zaprojektowaną w celu zapewnienia lepszych doświadczeń edukacyjnych i rozwoju dla uczniów. Dzięki licznym rozszerzeniom w mBlock5, nauczyciele mogą z łatwością integrować na potrzeby lekcji najnowocześniejsze technologie, takie jak Data Science, IoT i AI. Ponadto, poprzez integrację kodowania opartego na blokach kodów i kodowania w języku Python, mBlock5 oferuje uczniom ścieżkę edukacyjną, pozwalającą im rozwijać swoje umiejętności od zakresu podstawowego do rozszerzonego.

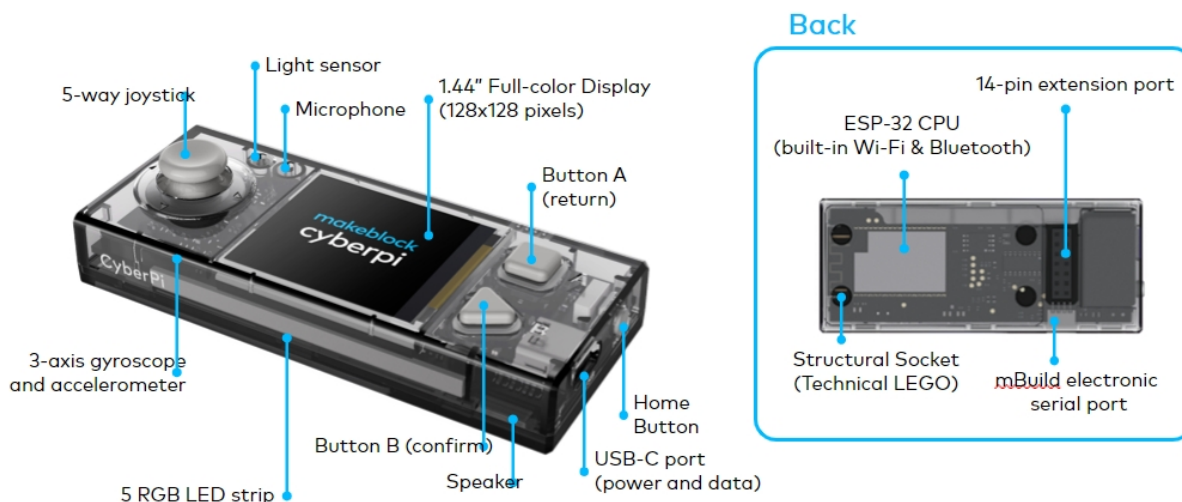
CyberPi posiada wiele wbudowanych, programowalnych czujników i elementów wykonawczych. Uczniowie mają możliwość wizualizacji tekstu, danych i obrazów na kolorowym wyświetlaczu, nagrywania dźwięków lub odtwarzania plików audio w zależności od danych wejściowych z czujników. Pozwala to na tworzenie ekscytujących projektów dydaktycznych dostosowanych do poziomu uczniów, na przykład generatora dźwięku poprzez połączenie diod LED i głośnika w CyberPi lub wykorzystanie modułu jako kontrolera gier, dzięki zintegrowanym przyciskom, joystickowi i żyroskopowi. To tylko niektóre z projektów, które są częścią ćwiczeń wstępnych.

Ich zakres można rozszerzyć, łącząc CyberPi z modułami dodatkowymi i innymi komponentami. Dla przykładu, moduł akumulatora umożliwia przenoszenie i większą swobodę korzystanie z CyberPi, pozwalając na wykonywanie ćwiczeń nawet, gdy CyberPi jest odłączony od komputera. Moduł ten umożliwia również łączenie CyberPi z innymi zewnętrznymi komponentami, takimi jak serwomechanizmy lub silniki DC, w projektach wykorzystujących zjawisko ruchu. Może również łączyć

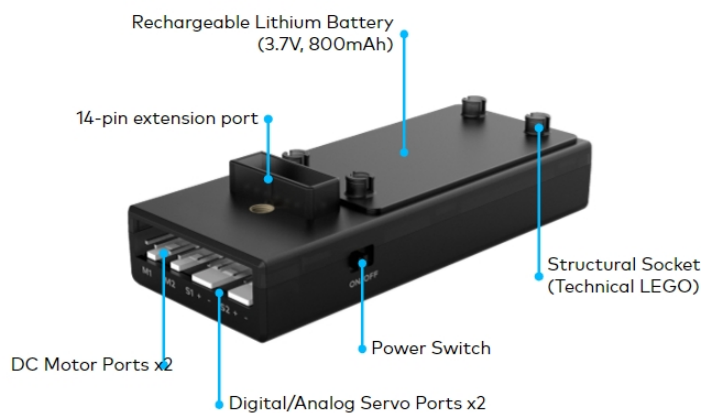
się z diodami LED oraz szeroką gamą czujników i elementów wykonawczych firm zewnętrznych, powszechnie używanych w projektach DIY.

CyberPi jest również kompatybilny z modułami mBuild projektu Makeblock Education. Są to inteligentne czujniki i elementy wykonawcze, które mogą być bezpośrednio podłączone bez konieczności stosowania skomplikowanego okablowania lub konfiguracji, co pozwala uczniom spędzić więcej czasu na wymyślaniu i wdrażaniu rozwiązań.

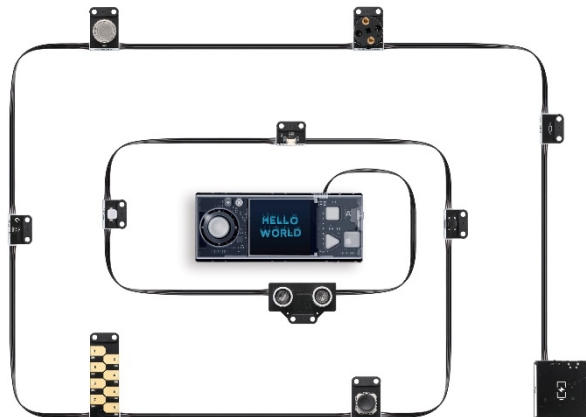
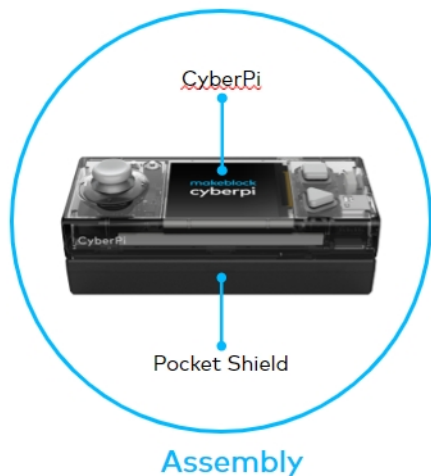
CyberPi jest wyposażony w następujące funkcje:



Mikrokontroler CyberPi



Moduł akumulatora



3. Programowanie CyberPi

(Opis działania oprogramowania mBlock)

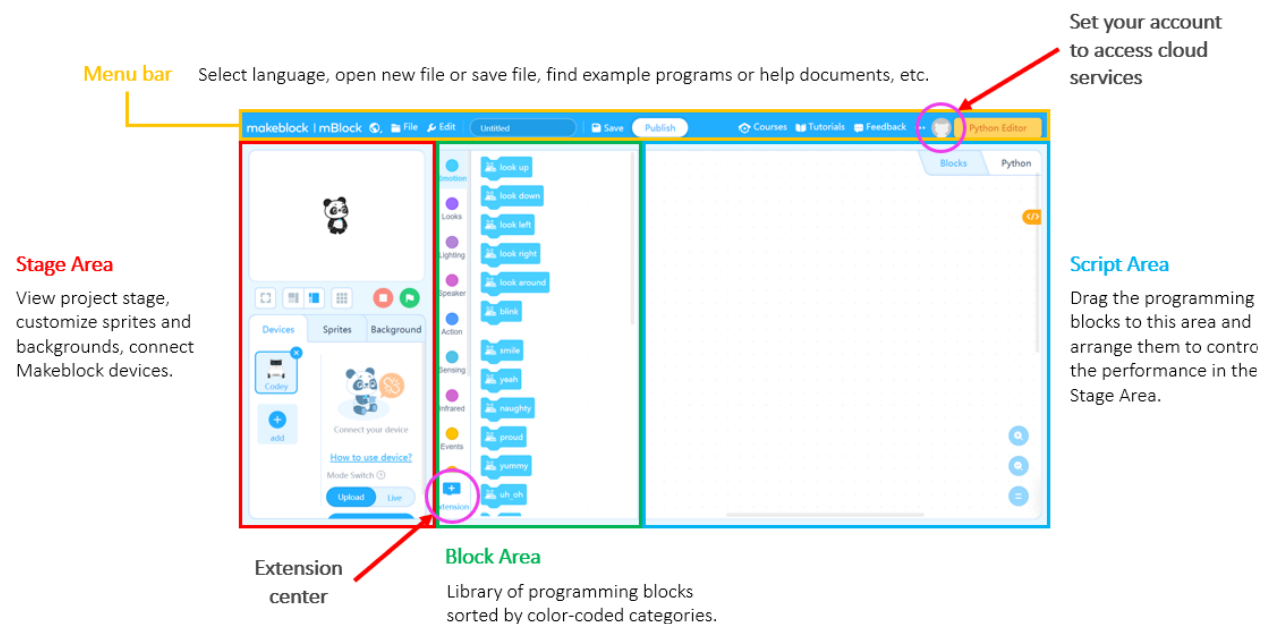
Z mBlock5 można korzystać zarówno za pomocą komputerów, laptopów, jak i urządzeń mobilnych. mBlock5 jest kompatybilny z różnymi systemami operacyjnymi, w tym Windows, Mac, Linux, Chromebook, iOS i Android. Jest to program typu open source, który oferuje możliwość tworzenia nowych rozszerzeń dla oprogramowania i sprzętu, umożliwiając tym samym nauczycielom dostosowanie narzędzia do własnych potrzeb.

Użytkownicy mogą również wyszukiwać i dzielić się projektami w Makeblock Community..

Programowanie oparte na blokach

Rozpoczęcie programowania z mBlock5 jest tak proste, jak przeciąganie i upuszczanie bloków kodów.

Interfejs mBlock5 składa się z następujących części:



Szczegółowe informacje dotyczące charakterystyki i zastosowania mBlock5 znaleźć na stronie <https://education.makeblock.com/help/category/mblock-block-based/>

Programowanie w Pythonie

4. Opis ćwiczeń

(Opis i wyjaśnienie struktury dokumentu)

Ćwiczenia wstępne z CyberPi zostały opracowane dla uczniów w wieku od 11 do 14 lat oraz ich nauczycieli. Poniżej opisano 9 ćwiczeń wstępnych:

Nazwa ćwiczenia	Opis	Kluczowe zagadnienia
1. Zapoznanie z CyberPi	W ćwiczeniu uczniowie zapoznają się z CyberPi - mikrokontrolerem o różnorodnych funkcjach z mnóstwem czujników, przycisków i pełnokolorowym wyświetlaczem. Poznają również kluczowe cechy CyberPi poprzez eksplorację przykładowych programów w mBlock.	<ul style="list-style-type: none"> Komponenty i funkcje CyberPi. Nawigacja po oprogramowaniu mBlock. Ustanawianie połączenia między oprogramowaniem a sprzętem.
2. Generator dźwięku	W tej lekcji uczniowie zorganizują imprezę taneczną wykorzystując wbudowane diody LED i głośnik. Program będzie sterowany za pomocą przycisków CyberPi do wyzwalania zdarzeń i uruchamiania	<ul style="list-style-type: none"> Komponenty wejściowe i wyjściowe CyberPi. Pisanie pseudokodu. Tworzenie programu w mBlock.

	skryptów. Uczniowie zaprogramują również przycisk wyłączający dźwięk i światła oraz przycisk restartujący CyberPi.	
3. Dyktafon	Poprzez połączenie głośnika, mikrofonu i zintegrowanej pamięci, uczniowie przekształcą CyberPi w kieszonkowy dyktafon i urządzenie odtwarzające. Wykorzystując metodę iteracyjną uczniowie będą oceniać swoje projekty i ulepszać zaprojektowane dyktafony.	<ul style="list-style-type: none"> • Nagrywanie dźwięku za pomocą CyberPi. • Odtwarzanie nagrań. • Wykorzystanie iteracyjnego procesu programowania.
4. Dyktafon (ciąg dalszy)	Jest to kontynuacja projektu Dyktafonu. Podczas lekcji uczniowie otrzymają informacje zwrotne od swoich kolegów i koleżanek z klasy i przemyślą raz jeszcze swoje początkowe rozwiązanie. Następnie zaplanują i stworzą projekt dyktafonu o większej liczbie funkcji.	<ul style="list-style-type: none"> • Zbieranie informacji zwrotnych od rówieśników. • Wykorzystanie iteracyjnego procesu programowania.
5. Kontroler do gier	Podczas tej lekcji uczniowie zmienią CyberPi w kontroler do gier łącząc procesy programowania urządzeń i programowania w mBlock. Uczniowie zapoznają się z przykładowymi programami pozwalającymi kontrolować ruch duszków animowanych za pomocą CyberPi. Następnie, z wykorzystaniem metody programowania w parach, uczniowie zmodyfikują istniejącą grę, aby wykorzystać CyberPi jako kontroler do niej.	<ul style="list-style-type: none"> • Połączenie programowania w mBlock i programowania urządzeń. • Programowanie w parach. • Dekompozycja i abstrakcja.
6. Czujnik pomiarowy	Podczas lekcji uczniowie dowiedzą się, w jaki sposób wbudowane czujniki CyberPi przedstawiają głośność i natężenie światła w otaczającym środowisku. Poznają też zasady reprezentacji danych i tworzenia wykresów mierzonych wartości z czujników.	<ul style="list-style-type: none"> • Reprezentacja danych. • Debugowanie programów. • Zasada działania czujników.
7. Mikser kolorów	Podczas lekcji uczniowie zapoznają się ze pojęciem zmiennych w celu	<ul style="list-style-type: none"> • Przechowywanie danych za pomocą zmiennych.

	<p>stworzenia miksera kolorów CyberPi. Program będzie wykorzystywał joystick i przyciski do sterowania wartościami kolorów R, G, B wszystkich wbudowanych diod LED. Zastosowanie instrukcji warunkowych zapewni, że wartości R, G, B będą zawsze znajdować się w określonym zakresie.</p>	<ul style="list-style-type: none"> • Zastosowanie wyrażeń warunkowych.
8. Miernik siły	<p>Podczas lekcji uczniowie stworzą grę, w której gracz będzie potrząsać CyberPi przez dziesięć sekund. Gra będzie zaprogramowana tak, aby wszystkie wyniki powyżej 50 wstrząsów były zapisywane.</p>	<ul style="list-style-type: none"> • Zapisywanie wyników. • Korzystanie z timera CyberPi. • Wyświetlanie tekstu na wyświetlaczu CyberPi.
9. Alarm prezentowy	<p>Uczniowie wykorzystają CyberPi do stworzenia programu, który wykrywa, czy ich koledzy i koleżanki potrząsnęli prezentem urodzinowym. Dzięki wykorzystaniu komunikacji bezprzewodowej, uczniowie będą przysyłać wiadomości pomiędzy urządzeniami komputerowymi, pozwalając jednemu urządzeniu kontrolować inne.</p>	<ul style="list-style-type: none"> • Korzystanie z sieci bezprzewodowych. • Komunikacja między urządzeniami.

Lekcja 1:

Zapoznanie z CyberPi

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Zidentyfikować kluczowe funkcje CyberPi.
- Podłączyć CyberPi do komputera za pomocą oprogramowania mBlock.
- Przeanalizować przykładowe programy dla CyberPi przy użyciu oprogramowania mBlock.

★ Informacje ogólne

W ćwiczeniu uczniowie zapoznają się z CyberPi - mikrokontrolerem o różnorodnych funkcjach z mnóstwem czujników, przycisków i pełnokolorowym wyświetlaczem. Poznają również kluczowe cechy CyberPi poprzez eksplorację przykładowych programów w mBlock.

🔗 Kluczowe zagadnienia

- Komponenty i funkcje CyberPi
- Nawigacja po oprogramowaniu mBlock
- Ustanawianie połączenia między oprogramowaniem a sprzętem

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program dołączony do oprogramowania mBlock

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program dołączony do oprogramowania mBlock

 **Standardy**

Typ	Punkt	Standard
ISTE	6a	Wybór przez uczniów odpowiednich platform i narzędzi do osiągnięcia pożądaných celów.
K12 CS Framework	Practice 2-1	Nawiązywanie współpracy z osobami o różnych perspektywach, poglądach i osobowościach.

 **Plan zajęć (45 minut)**

Czas	Treść
5 minut	<p>Rozgrzewka</p> <ul style="list-style-type: none"> • Zapoznanie z CyberPi
15 minut	<p>Teoria i praktyka</p> <ul style="list-style-type: none"> • Przegląd mBlock 5 • Podłączenie CyberPi <ul style="list-style-type: none"> ○ Dodawanie rozszerzeń CyberPi ○ Testowanie trybu Live • Analiza przykładowego programu <ul style="list-style-type: none"> ○ Testowanie trybu Upload
20 minut	<p>Rozwiązywanie zadania</p> <ul style="list-style-type: none"> • Analiza i przetestowanie przykładowych programów
5 minut	<p>Podsumowanie</p> <ul style="list-style-type: none"> • Podsumowanie cech i możliwości CyberPi • Burza mózgów - pomysły na programy z wykorzystaniem CyberPi • Rozszerzenie lekcji

Rozgrzewka [5 minut]

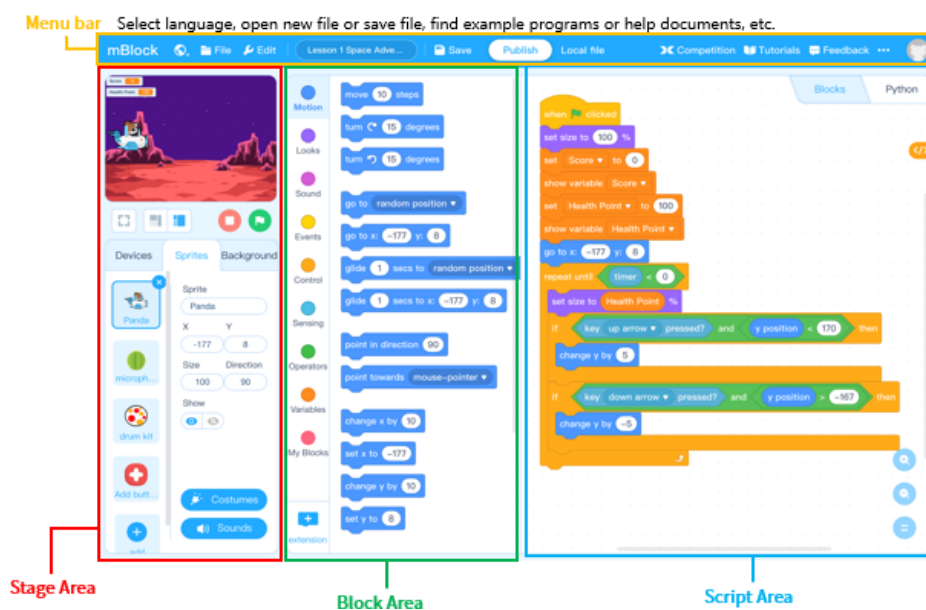
Zapoznanie z CyberPi

1. Na początku lekcji nauczyciel prezentuje uczniom zestaw CyberPi. Po kolei przedstawia następujące elementy, a uczniowie mają za zadanie odszukać je we własnym zestawie CyberPi:
 - a. CyberPi
 - b. Kabel USB-C
 - c. Moduł akumulatora (nie jest dołączony do zestawu podstawowego)
 - d. Czujniki mBuild (nie są dołączone do zestawu podstawowego)
2. Uczniowie zapoznają się z instrukcjami na pudełku oraz instrukcjami szybkiego uruchomienia. Następnie mają za zadanie napisać krótkie podsumowanie, czego się do tej pory dowiedzieli o funkcjach i możliwościach CyberPi.

Teoria i praktyka [15 minut]

Przegląd mBlock 5

1. Uczniowie otwierają oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#).
2. Nauczyciel przedstawia następujące kluczowe obszary interfejsu oprogramowania:



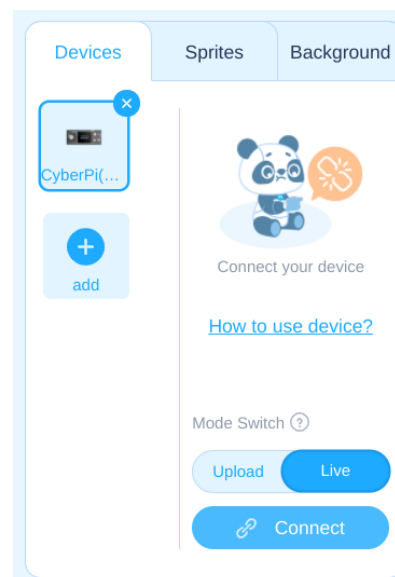
Nazwa obszaru	Funkcja
Pasek menu	<ul style="list-style-type: none"> Wybór języka Tworzenie, otwieranie lub zapisywanie pliku Szukanie przykładowych programów, plik pomocy, itp.
Scena	<ul style="list-style-type: none"> Wyświetlanie etapu projektu Wybór i edycja animowanego duszka i tła Podłączanie sprzętu
Obszar bloków	<ul style="list-style-type: none"> Wyszukiwanie i wybór bloków skryptów zorganizowanych w kategorie oznaczone kolorami Wyszukiwanie i dodawanie rozszerzeń
Obszar skryptów	<ul style="list-style-type: none"> Łączenie bloków w celu tworzenia programów lub skryptów Przeciąganie bloków i układanie ich w określonej kolejności, aby kontrolować duszkami, tłem i/lub urządzeniem

Podłączenie CyberPi

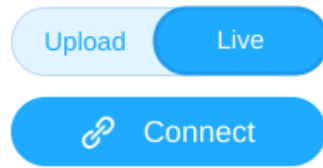
- Podłączyć CyberPi do komputera za pomocą dołączonego kabla. CyberPi powinien się uruchomić, a na ekranie powinien pojawić się ostatnio załadowany program lub menu główne.
- W zakładce **Devices** w programie mBlock kliknąć przycisk **Add**. Wybrać **CyberPi** i kliknąć **Ok**.
- Kliknąć przycisk **Connect**. Następnie wybrać port USB i kliknąć **Connect**.
- Jeśli połączenie przebiegło pomyślnie, przycisk zmieni się na **Disconnect**.

Niepodłączony

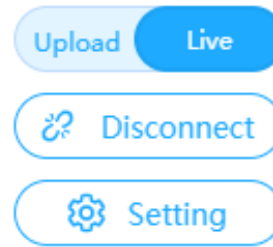
Podłączony




Mode Switch ?



Mode Switch ?




- CyberPi jest podłączony w trybie **Live**. Czas na przetestowanie połączenia.
- W **Obszarze bloków** wybrać kategorię **LED**.

- Kliknąć blok . Na pasku LED CyberPi wyświetlane będą wskazane kolory.

Analiza i przetestowanie przykładowych programów

- Kliknąć przycisk **Tutorials** w prawym górnym rogu. Wybrać **Example Programs**.
- Kliknąć **CyberPi**, aby wyświetlić przykładowe programy dla CyberPi.
- Wyszukać i wybrać program **Rainbow Lights**.
- Zadaniem uczniów będzie odczytanie kodu i odgadnięcie działania, które wywołuje.
- Podłączyć CyberPi w trybie **Live**.

- Kliknąć blok . Na pasku LED CyberPi wyświetlane będą wskazane kolory.
(Uwaga, świecąca żółta obwódka otaczająca skrypt oznacza, że jest on uruchomiony). Kliknąć na bloki, aby zatrzymać program.

- Wyjaśnić uczniom różnicę między trybem **Live** a trybem **Upload**.

Tryb	Opis
Tryb Live	<ul style="list-style-type: none"> Program jest uruchamiany przez komputer (nie jest przechowywany na CyberPi) CyberPi musi być podłączony do komputera Projekt mBlock musi być otwarty Używany do programowania z wykorzystaniem Sceny
Tryb Upload	<ul style="list-style-type: none"> Program jest wgrywany i zapisywany na CyberPi Brak komunikacji z komputerem CyberPi może być odłączony od komputera Oprogramowanie mBlock może zostać zamknięte

- Program zostanie zapisany na CyberPi do momentu wgrania nowego programu

17. Przełączyć CyberPi w tryb **Upload** i kliknąć przycisk **Upload**.

18. Pojawi się okno **Upload Progress**, które zniknie po zakończeniu wgrywania programu. CyberPi uruchomi się ponownie i uruchomi program **Rainbow Lights**. Program ten będzie się uruchamiał przy każdym włączeniu CyberPi.

Rozwiązywanie zadania [20 minut]

Analiza i przetestowanie przykładowych programów

1. W tej części zadaniem uczniów będzie przetestowanie 3 z pozostałych programów przykładowych dołączonych do programu mBlock. Niektóre z propozycji to:
 - Buzzer
 - Twinkle Twinkle Little Star
 - Światła reagujące na głos
 - Przypomniczek
 - Prosty timer
 - Licznik kroków
 - Wykres reagujący na ruch
2. Dla każdego testowanego programu uczniowie:
 - Sporządzą opis programu.
 - Zidentyfikują, które komponenty CyberPi są używane do każdego zadania w programie.
 - Wyciągną wnioski dotyczące funkcji poszczególnych fragmentów kodu.

Podsumowanie [5 minut]

Wnioski i burza mózgów

1. Ostatnia część lekcji jest przeznaczona na dyskusję, w której uczniowie podzielą się swoimi ulubionymi przykładami projektów. Należy zachęcać uczniów, aby wskazali funkcje i możliwości CyberPi, które najbardziej ich zainteresowały.
2. Uczniowie powinni również mieć możliwość wyrażenia swoich oczekiwań odnośnie umiejętności zdobywanych podczas lekcji oraz uzyskania odpowiedzi na wszelkie

pytania, jakie mogą mieć.

3. Bazując na przetestowanych przykładach zastosowania CyberPi, warto przeprowadzić wśród uczniów burzę mózgów na temat problemów z życia codziennego, które można rozwiązać za pomocą CyberPi i mBlock.

Rozszerzenie lekcji

- Jeśli uczniowie potrzebują dodatkowej pomocy w korzystaniu z programu mBlock, warto zachęcić ich do zapoznania się z ćwiczeniami wstępnymi: Pierwsze kroki z programem mBlock 5.

Lekcja 2

Generator dźwięku

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Zidentyfikować komponenty wejściowe i wyjściowe CyberPi.
- Napisać pseudokod, aby zaplanować i stworzyć program w mBlock.
- Stworzyć program w mBlock używając przycisków CyberPi do wyzwalania zdarzeń.
- Używać bloków programowania do sterowania głośnikiem i diodami LED.

★ Informacje ogólne

W tej lekcji uczniowie zorganizują imprezę taneczną wykorzystując wbudowane diody LED i głośnik. Program będzie sterowany za pomocą przycisków CyberPi do wyzwalania zdarzeń i uruchamiania skryptów. Uczniowie zaprogramują również przycisk wyłączający dźwięk i światła oraz przycisk restartujący CyberPi.

🔗 Kluczowe zagadnienia

- Komponenty wejściowe i wyjściowe CyberPi
- Pisanie pseudokodu
- Tworzenie programu w mBlock

📋 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)

- Przykładowy program: CyberPi - Lekcja 2 - Generator dźwięku

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)

 **Standardy**

Typ	Punkt	Standard
CSTA	2-AP-10	Korzystanie ze schematów blokowych i/lub pseudokodu do rozwiązywania złożonych problemów jako algorytmów.
K12 CS Framework	Practice 5-2	Tworzenie rozwiązań komputerowych dla celów praktycznych, wyrażenia siebie lub w celu rozwiązania określonych problemów społecznych.

 **Plan zajęć (45 minut)**

Czas	Treść
5 minut	<p>Rozgrzewka</p> <ul style="list-style-type: none"> • Komponenty wejściowe i wyjściowe
15 minut	<p>Teoria i praktyka</p> <ul style="list-style-type: none"> • Planowanie programu za pomocą pseudokodu • Pisanie programu • Randomizacja danych wyjściowych • Restartowanie CyberPi
20 minut	<p>Rozwiązywanie zadania</p> <ul style="list-style-type: none"> • Stworzenie generatora dźwięku
5 minut	<p>Podsumowanie</p> <ul style="list-style-type: none"> • Prezentacja projektów • Rozszerzenie lekcji

Rozgrzewka [5 minut]

Komponenty wejściowe i wyjściowe

1. Na początku lekcji dobrze jest omówić poniższe definicje.

Pojęcie	Definicja*
komponent wejściowy	Urządzenie lub komponent, który umożliwia przekazanie informacji do komputera.
komponent wyjściowy	Urządzenie lub komponent, który odbiera informacje z komputera.

*definicje zaczerpnięte z Code.org – CSD Unit 1

2. Zadaniem uczniów będzie stworzenie w parach lub małych grupach listy komponentów wejściowych i wyjściowych smartfona. Przykłady:

Smartfon	
Komponent wejściowy	i wyjściowe
Mikrofon	Głośnik
Ekran dotykowy	Ekran / wyświetlacz
Przyciski	Słuchawki
GPS	Wibracje
Czujnik ruchu (przechylenie telefonu)	Połączenie internetowe
Czujnik światła	Dioda LED (latarka / lampa błyskowa aparatu)
Kamera	Port ładowania
Połączenie internetowe	Bluetooth
Czujnik temperatury	
Port ładowania	
Bluetooth	

3. Następnie, na podstawie Lekcji 1 - Zapoznanie z CyberPi, uczniowie stworzą listę komponentów wejściowych i wyjściowych dla CyberPi. W przypadku problemów, należy zasugerować uczniom, aby posłużyli się dokumentacją produktu dołączoną do CyberPi.

CyberPi	
Komponent wejściowy	i wyjściowe
Mikrofon Przyciski (A, B i Home) Joystick Port ładowania Bluetooth Czujnik ruchu (żyroskop) Czujnik światła Czujnik głośności/dźwięku	Głośnik Ekran / wyświetlacz Diody LED Wskaźnik LED (pokazujący ładowanie i włączone zasilanie)
Modele z modułem akumulatora i zestawem modułów mBuild	
Czujnik Multi-Touch Slider Czujnik ultradźwiękowy Czujniki innych producentów	Silniki (enkoder i serwo) Diody LED Moduły innych producentów

Teoria i praktyka [15 minut]

Planowanie programu za pomocą pseudokodu

1. W tej części lekcji należy omówić z uczniami, jak ważne jest odpowiednie zaplanowanie programu przed jego stworzeniem. Należy zapoznać uczniów z pojęciem *pseudokodu*, który może być pomocnym narzędziem przy planowaniu projektów w mBlock.

Pojęcie	Definicja
pseudokod	Pisemna sekwencja kroków programu napisana w języku angielskim lub języku ojczystym programisty.

2. Korzystając z poniższego opisu projektu, należy przeanalizować z uczniami proces pisania pseudokodu dla niego.

Generator dźwięku	
Opis projektu	Zaprogramowanie CyberPi w taki sposób, aby po naciśnięciu przycisku A zapalały się diody, a po naciśnięciu przycisku B rozległ się ciągły dźwięk. Środkowy przycisk joysticka powinien zatrzymywać dźwięk i wyłączać diody.
Pseudokod	Naciśnięcie przycisku A:

	<p>Włącza diody LED, które świecą określonym kolorem w sposób ciągły</p> <p>Naciśnięcie przycisku B: Włącza brzęczyk, rozlegający się w sposób ciągły określonym tonem</p> <p>Naciśnięcie środkowego przycisku joysticka: Zatrzymuje dźwięk i wyłącza światła</p>
--	---


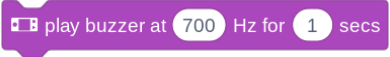


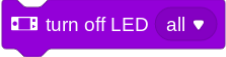
3. W kolejnym kroku zadaniem uczniów będzie przeanalizowanie powyższego pseudokodu i zidentyfikowanie komponentów wejściowych i wyjściowych programu.

Projekt Generатора dźwięku	
Komponent wejściowy	i wyjściowe
Przycisk A Przycisk B Przycisk środkowy joysticka	Głośnik / brzęczyk Diody LED

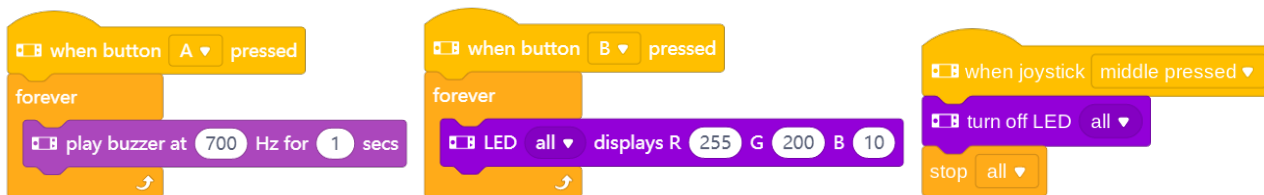
Pisanie programu

1. Teraz, gdy program jest już zaplanowany, nadszedł czas na poznanie nowych bloków potrzebnych do napisania pseudokodu.

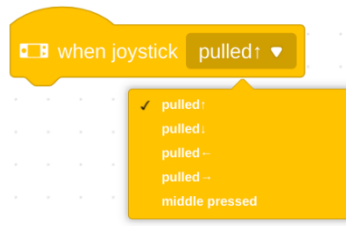
Kategoria	Blok	Funkcja
 Events		Określa działanie na CyberPi, które wyzwała wykonanie określonych akcji.
		
 Control		Instrukcja pętli Wykonuje w sposób ciągły akcje zagnieżdżone wewnątrz bloku.
		Zatrzymuje wszystkie skrypty, w tym wszystkie pętle.

 Audio		<p>Odtwarza określony ton na brzęczyku CyberPi przez określony czas.</p> <p>Zakres częstotliwości: 0 do 1000</p>
 LED		<p>Zapala wszystkie lub pojedynczą diodę LED na określony kolor.</p> <p>Zakres kolorów: 0 do 255</p>
		<p>Wyłącza wszystkie lub pojedynczą diodę LED.</p>

- Należy otworzyć oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.
- W tym punkcie należy przypomnieć uczniom, w jaki sposób przeciąga się i upuszcza bloki z kategorii oznaczonych kolorami w Obszarze bloków. Należy również poinstruować ich, jak zbudować każdy z poniższych skryptów:





Opcję naciśnięcia środkowego przycisku można wybrać w menu rozwijanym.



- Przetestować program w trybie Live i/lub w trybie Upload.
- Należy zachęcić uczniów do eksperymentowania z różnymi wartościami brzęczyka i diod LED, aby zaobserwować, jak zachowuje się CyberPi.
- Rozszerzenie:** Jeśli czas na to pozwala, można omówić zagadnienie wartości kolorów RGB. Można też polecić uczniom skorzystanie z próbnika kolorów w celu zidentyfikowania wartości RGB poszczególnych kolorów.

Randomizacja danych wyjściowych

7. W powyższym kodzie została zaprogramowana określona częstotliwość brzęczyka oraz wartość koloru RGB. CyberPi w nieskończoność powtarza ten sam dźwięk a dioda LED świeci się na ten sam kolor. Poniższy blok może być użyty do wybrania losowej wartości przy każdym powtórzeniu pętli.

Kategoria	Blok	Funkcja
 Operators		Komputer lub CyberPi wybiera losową wartość z podanego zakresu.

8. W tym miejscu należy zmodyfikować poprzednie programy tak, aby zawierały następujące bloki losowe:



Uwaga, zakresy te odpowiadają zakresowi akceptowanych wartości dla każdego bloku.

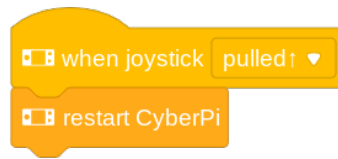
9. Zadaniem uczniów będzie przetestowanie nowego programu. Uczniowie mogą zmieniać wartość czasu w bloku na liczbę dziesiętną, jeśli chcieliby uzyskać dźwięk brzęczyka o większej częstotliwości.

Restartowanie CyberPi

10. Zaprogramowanie przycisku do restartowania CyberPi może być pomocnym narzędziem w kolejnych lekcjach. Należy przeanalizować wspólnie z uczniami proces programowania przycisku do ręcznego restartu CyberPi.

Kategoria	Blok	Funkcja
 Control		Restart lub ponowne uruchomienie urządzenia CyberPi. Po ponownym uruchomieniu CyberPi odtworzy ostatni program wgrany do urządzenia.

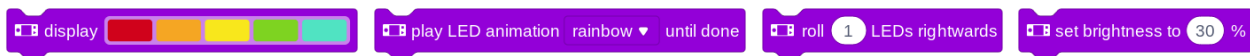
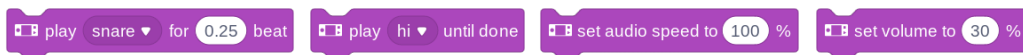
11. Należy polecić uczniom zbudowanie poniższego skryptu i przetestowanie programu:



Rozwiązywanie zadania [20 minut]

Stworzenie generatora dźwięku

1. Należy dać uczniom czas na eksperymentowanie z istniejącym kodem. Warto zachęcić ich do wypróbowania różnych wartości parametrów w blokach,
2. jak również do zapoznania się z innymi blokami z kategorii LED i Audio w Obszarze bloków. Niektóre bloki warte uwagi to:




3. Po zapoznaniu się z różnymi blokami, uczniowie będą mieli za zadanie napisać pseudokod dla ulepszonej wersji projektu **Generatora dźwięku**.
4. Pseudokod posłuży jako wskazówka podczas tworzenia projektu.

Podsumowanie [5 minut]

Prezentacja projektów

1. W ostatniej części lekcji należy podzielić uczniów na pary i polecić im zaprezentowanie swoich Generatorów dźwięku partnerom.
2. Uczniowie powinni zadać sobie nawzajem następujące pytania:
 - o Z jakiej części projektu jesteś najbardziej dumny?
 - o Co było największym wyzwaniem przy tym projekcie?

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można polecić uczniom zaprogramowanie przycisku zatrzymania, używając bloku Repeat Until z blokiem  .
- Za pomocą przycisków joysticka można zaprogramować wiele różnych pokazów świetlno-dźwiękowych.

Lekcja 3

Dyktafon

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Stworzyć program w mBlock pozwalający na nagrywanie i odtwarzanie dźwięku.
- Zastosować proces iteracyjny w celu opracowania rozwiązania problemu programistycznego.

★ Informacje ogólne

Poprzez połączenie głośnika, mikrofonu i zintegrowanej pamięci, uczniowie przekształcą CyberPi w kieszonkowy dyktafon i urządzenie odtwarzające. Wykorzystując metodę iteracyjną uczniowie będą oceniać swoje projekty i ulepszać zaprojektowane dyktafony.

🔗 Kluczowe zagadnienia

- Nagrywanie dźwięku za pomocą CyberPi
- Odtwarzanie nagrań
- Wykorzystanie iteracyjnego procesu programowania

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program:

CyberPi - Lekcja 3 - Generator dźwięku 1

CyberPi - Lekcja 3 - Generator dźwięku 2

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)

Standardy

Typ	Punkt	Standard
CSTA	2-CS-1	Rekomendowanie ulepszeń w projektowaniu urządzeń komputerowych na podstawie analizy interakcji użytkowników z urządzeniami.
K12 CS Framework	Practice 3-1	Identyfikowanie złożonych, interdyscyplinarnych problemów świata rzeczywistego, które mogą być rozwiązane programistycznie.
K12 CS Framework	Practice 5-1	Planowanie rozwiązań programistycznych z wykorzystaniem iteracyjnego procesu, obejmującego analizę i modyfikację planu, biorąc pod uwagę kluczowe cechy, ograniczenia czasowe i zasobowe oraz oczekiwania użytkowników.
K12 CS Framework	Practice 6-1	Regularne testowanie rozwiązań programistycznych z uwzględnieniem wszystkich scenariuszów i przypadków testowych.
K12 CS Framework	Practice 6-2	Identyfikowanie i naprawa błędów z wykorzystaniem systematycznego procesu.
K12 CS Framework	Practice 6-3	Wielostopniowa ocena i udoskonalanie rozwiązań programistycznych w celu poprawy ich wydajności, niezawodności, użyteczności i dostępności.

Plan zajęć (45 minut)

Czas	Treść
10 minut	Rozgrzewka <ul style="list-style-type: none"> • Rozwój rozwiązań programistycznych
10 minut	Teoria i praktyka <ul style="list-style-type: none"> • Tworzenie planu • Stworzenie Dyktafonu 1.0
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Stworzenie Dyktafonu 2.0
5 minut	Podsumowanie <ul style="list-style-type: none"> • Wnioski • Rozszerzenie lekcji

Ćwiczenia

Rozgrzewka
[10 minut]

Rozwój rozwiązań programistycznych

1. Pierwsza część lekcji powinna być poświęcona na przedyskutowanie z uczniami sposobu, w jaki komputery rozwiązują codzienne, rzeczywiste problemy. Należy opracować wspólnie listę rozwiązań programistycznych, z którymi uczniowie i ich rodziny stykają się na co dzień. Przykłady:
 - Smartfon
 - Budzik
 - Alarm w lodówce
 - Samochody
 - Systemy transportu publicznego
 - GPS
 - Mikrofalówka
 - Prognozowanie pogody / aplikacje pogodowe
2. Następnie należy omówić z uczniami zagadnienie ewolucji rozwiązań programistycznych na przestrzeni czasu.

Poniżej podano przykład.

Ewolucja systemów nawigacyjnych

- Zastosowanie drukowanych map drogowych w celu wyznaczenia trasy z jednego miejsca do drugiego przed wyjazdem.
- Korzystanie ze stron internetowych w celu wygenerowania i wydrukowania wskazówek nawigacyjnych przed wyjazdem (np. Mapquest).
- Zastosowanie urządzeń mobilnych do wyświetlania strony internetowej ze wskazówkami nawigacyjnymi podczas podróży.
- Zastosowanie urządzeń mobilnych do przeglądania mapy podczas jazdy (bez podawania wskazówek) (była to pierwsza wersja aplikacji Google Maps, która nie podawała wskazówek dojazdu).
- Zastosowanie urządzeń mobilnych do wyświetlania mapy ze wskazówkami dojazdu.
- Zastosowanie urządzeń mobilnych do korzystania z systemu GPS ze wskazówkami zakręt po zakręcie (na tym etapie obsługa tranzytu ani pieszego ruchu nie była dostępna).
- Zastosowanie urządzeń mobilnych do korzystania z systemu GPS ze wskazówkami zakręt po zakręcie oraz z dostosowaniem trasy na podstawie bieżących danych o ruchu drogowym.
- Zastosowanie urządzeń mobilnych do nawigacji podczas jazdy, tranzytu czy poruszania się pieszo z wykorzystaniem systemu GPS w czasie rzeczywistym.

Systemy nawigacji w smartfonach lub urządzeniach przenośnych ewoluowały z czasem. Na przykład w pierwszej wersji Google Maps brakowało wielu kluczowych funkcji (tj. wskazówek dotyczących poruszania się pieszo, transportu publicznego, ruchu drogowego na żywo, informacji o zamkniętych drogach, informacji o miejscu docelowym itp.), które z czasem zostały dodane w wyniku procesu analizy i oceny rozwiązania w oparciu o opinie użytkowników oraz gromadzenia danych.

[10 minut]

Tworzenie planu

1. W tej części lekcji należy przeanalizować z uczniami rozwiązanie poniższego problemu. Dla ułatwienia podane jest również przykładowe rozwiązanie. Pierwsza wersja rozwiązania powinna być dość prosta. Pozwoli to uczniom na zastosowanie procesu iteracji i dodawanie funkcji w trakcie pracy nad rozwojem dyktafonu o ulepszonych funkcjach.

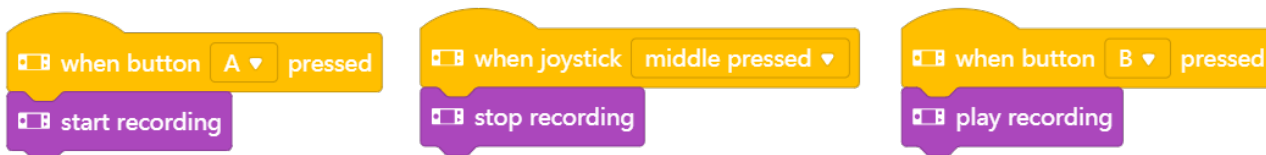
Dyktafon 1.0	
Problem	Jeden z Waszych kolegów z klasy uczęszcza na zajęcia z języka obcego. Chciałby poćwiczyć wymowę niektórych słówek poznanych na zajęciach, ale po powrocie do domu zapomina, jak nauczyciel wymawiał te słowa podczas lekcji. W jaki sposób CyberPi może pomóc Waszemu koledze w rozwiązaniu tego problemu?
Sugerowane rozwiązanie	Nagrywanie nauczyciela wymawiającego słowa, a następnie odtwarzanie nagrań z wykorzystaniem CyberPi.
Pseudokod	Naciśnięcie przycisku A: Rozpoczęcie nagrywania Naciśnięcie środkowego przycisku joysticka: Zatrzymanie nagrywania Naciśnięcie przycisku B: Odtwarzanie nagrania

Stworzenie Dyktafonu 1.0

2. Teraz, gdy program jest już zaplanowany, nadszedł czas na poznanie nowych bloków potrzebnych do napisania pseudokodu.

Kategoria	Blok	Funkcja
 Audio		Rozpoczęcie nagrywania. Czas nagrywania ograniczony do 10 sekund.
		Zatrzymanie nagrywania.
		Odtwarzanie ostatniego nagrania zapisanego na CyberPi.

- Należy otworzyć oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.
- W tym punkcie należy przypomnieć uczniom, w jaki sposób przeciąga się i upuszcza bloki z kategorii oznaczonych kolorami w Obszarze bloków. Należy polecić uczniom zbudowanie poniższego skryptu i przetestowanie programu:



- Po ukończeniu pracy nad prototypem urządzenia należy poprosić uczniów o wskazanie obszarów i aspektów, które można by ulepszyć w projekcie. Wykorzystując metodę burzy mózgów należy stworzyć wraz z uczniami listę potencjalnych sposobów ulepszenia prototypu Dyktafonu. Gdyby uczniowie potrzebowali dodatkowych wskazówek, należy zadać im następujące pytania:
 - Skąd Wasz kolega będzie wiedział, jak korzystać z Dyktafonu CyberPi?
 - Skąd będzie wiedział, czy Dyktafon aktualnie nagrywa dźwięk?
 - Jakie inne funkcje mogą być przydatne dla Waszego kolegi?
 - Jakie inne komponenty CyberPi mogą być wykorzystane do stworzenia rozwiązania o ulepszonych funkcjach? (np. wyświetlacz, diody LED, głośnik, czujnik ruchu)

Rozwiązywanie zadania [20 minut]

Stworzenie Dyktafonu 2.0

1. Przed rozpoczęciem pracy należy wspomnieć uczniom, że programiści i twórcy oprogramowania często ulepszają swoje rozwiązania. (Właśnie z tego powodu mamy aktualizacje i nowe wersje aplikacji)
Tworzenie oprogramowania to proces iteracyjny.
2. Korzystając z listy ulepszeń stworzonej w poprzedniej części lekcji należy poprosić uczniów o wybranie trzech najważniejszych funkcji,
3. które chcieliby dodać do swojego **Dyktafonu**. Następnie, zadaniem uczniów będzie napisanie pseudokodu dla każdej z funkcji. Przykładowy plan podano poniżej:

Dyktafon 2.0	
Cecha 1	Dodanie instrukcji na wyświetlaczu informujących użytkownika, jakie przyciski należy nacisnąć, aby sterować dyktafonem CyberPi.
	Po uruchomieniu CyberPi: Na ekranie wyświetla się napis: "Naciśnij przycisk A, aby rozpocząć nagrywanie. Naciśnij przycisk joysticka, aby zatrzymać nagrywanie. Naciśnij przycisk B, aby odtworzyć nagranie".
Cecha 2	Zastosowanie diod LED w celu sygnalizowania użytkownikowi, kiedy CyberPi jest w trakcie nagrywania dźwięku.
	Naciśnięcie przycisku A: Ustawienie wszystkich diod LED na zielony kolor Rozpoczęcie nagrywania
	Naciśnięcie środkowego przycisku joysticka: Ustawienie wszystkich diod LED na czerwony kolor Zatrzymanie nagrywania
Cecha 3	Naciśnięcie przycisku B: Ustawienie wszystkich diod LED na niebieski kolor Odtwarzanie nagrania
	Dodanie możliwości zmiany głośności CyberPi.
	Po uruchomieniu CyberPi: Ustawienie głośności na 50%.
	Po pociągnięciu joysticka w górę: Zwiększenie głośności o 10%
	Po pociągnięciu joysticka w dół: Zmniejszenie głośności o 10%

Do tej lekcji dołączone są dwa przykładowe programy. Są one przeznaczone do wglądu przez nauczyciela. Uczniowie powinni w miarę możliwości być zachęceni do wykorzystania własnej kreatywności i tworzenia programów według własnych pomysłów.

4. Kolejnym zadaniem uczniów będzie stworzenie **Dyktafonu 2.0** z wykorzystaniem własnych

pseudokodów jako wskazówki.

Podsumowanie [5 minut]

Wnioski

1. W ostatniej części lekcji należy poprosić uczniów o podzielenie się kilkoma pomysłami, które wykorzystali w swoich rozwiązaniach.
2. Należy również przypomnieć im, że w następnym ćwiczeniu będą kontynuować pracę nad projektem Dyktafonu z wykorzystaniem procesu iteracyjnego.

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można polecić uczniom zapoznanie się z UX Design (User Experience Design) i rolę, jaką odgrywa w tworzeniu oprogramowania.
- Można również polecić im zbadanie takich pojęć jak dostępność i użyteczność podczas programowania.

Lekcja 4

Dyktafon (ciąg dalszy)

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Dostarczać konstruktywnych informacji zwrotnych na temat konkretnych rozwiązań programistycznych.
- Zmodyfikować istniejący program w celu poprawy doświadczenia użytkownika.
- Zastosować proces iteracyjny w celu opracowania rozwiązania problemu programistycznego.

★ Informacje ogólne

Jest to kontynuacja projektu Dyktafonu. Podczas lekcji uczniowie otrzymają informacje zwrotne od swoich kolegów i koleżanek z klasy i przemyślą raz jeszcze swoje początkowe rozwiązanie. Następnie zaplanują i stworzą projekt dyktafonu o większej liczbie funkcji.

🔗 Kluczowe zagadnienia

- Zbieranie informacji zwrotnych od rówieśników
- Wykorzystanie iteracyjnego procesu programowania

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi - Lekcja 2 - Generator dźwięku

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)

 **Standardy**

Typ	Punkt	Standard
CSTA	2-IC-22	Współpraca z wieloma uczestnikami poprzez strategie takie jak crowdsourcing lub ankiety podczas tworzenia rozwiązań programistycznych.
CSTA	2-CS-1	Rekomendowanie ulepszeń w projektowaniu urządzeń komputerowych na podstawie analizy interakcji użytkowników z urządzeniami.
K12 CS Framework	Practice 2-3	Uzyskiwanie i uwzględnianie informacji zwrotnych od członków zespołu i innych uczestników procesu oraz przekazywanie im konstruktywnych informacji zwrotnych.
K12 CS Framework	Practice 3-1	Identyfikowanie złożonych, interdyscyplinarnych problemów świata rzeczywistego, które mogą być rozwiązane programistycznie.
K12 CS Framework	Practice 5-1	Planowanie rozwiązań programistycznych z wykorzystaniem iteracyjnego procesu, obejmującego analizę i modyfikację planu, biorąc pod uwagę kluczowe cechy, ograniczenia czasowe i zasobowe oraz oczekiwania użytkowników.
K12 CS Framework	Practice 6-1	Regularne testowanie rozwiązań programistycznych z uwzględnieniem wszystkich scenariuszów i przypadków testowych.
K12 CS Framework	Practice 6-2	Identyfikowanie i naprawa błędów z wykorzystaniem systematycznego procesu.
K12 CS Framework	Practice 6-3	Wielostopniowa ocena i udoskonalanie rozwiązań programistycznych w celu poprawy ich wydajności, niezawodności, użyteczności i dostępności.

 **Plan zajęć (45 minut)**

Czas	Treść
15 minut	Rozgrzewka <ul style="list-style-type: none"> • Zbieranie informacji zwrotnych od rówieśników
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Planowanie i stworzenie Dyktafonu 3.0
10 minut	Podsumowanie <ul style="list-style-type: none"> • Dokumentacja projektu • Rozszerzenie lekcji

 **Ćwiczenia**

Rozgrzewka

[15 minut]

Zbieranie informacji zwrotnych od rówieśników

1. W tej części lekcji wraz z całą klasą należy przeprowadzić ćwiczenie wzajemnej oceny oraz udzielania informacji zwrotnych. W tej części lekcji wraz z całą klasą należy przeprowadzić ćwiczenie wzajemnej oceny oraz udzielania informacji zwrotnych.
2. Należy poprosić uczniów, aby położyli CyberPi i swój plan Dyktafonu 2.0 na biurku i otworzyli mBlock.
3. Jeśli czas na to pozwala można polecić uczniom obchód po klasie w celu oceny projektów kolegów i koleżanek. Pytania pomocnicze do oceny mogą brzmieć następująco:
 - o Jaka cecha danego projektu najbardziej Ci się podoba?
 - o Czy instrukcje korzystania z dyktafonu były jasne? Czy nie było żadnych wątpliwości odnośnie korzystania z urządzenia?
 - o Jaka funkcja mogłaby zostać dodana do projektu, aby uczynić go bardziej przyjaznym dla użytkownika?
 - o Jaka funkcja mogłaby ulepszyć projekt dyktafonu?

Rozwiązywanie zadania

[20 minut]

Planowanie i stworzenie Dyktafonu 3.0

1. W ramach tej części lekcji uczniowie zastanowią się nad otrzymanymi informacjami zwrotnymi i przedyskutują wspólnie listę pomysłów dla **Dyktafonu 3.0**. Kilka pomysłów na ulepszenia:
 - o Dodanie tytułu pojawiającego się po uruchomieniu CyberPi.
 - o Zmiana koloru tekstu wyświetlanego na ekranie.
 - o Ustawienie określonego czasu trwania nagrania zamiast używania przycisku stop.
 - o Zastosowanie joysticka do sterowania głośnością i/lub prędkością odtwarzania.
 - o Wyświetlanie aktualnego poziomu głośności na ekranie lub użycie diod LED do wskazywania głośności.
 - o Animacja diod LED podczas nagrywania.
 - o Zastosowanie joysticka do kontrolowania czasu trwania nagrania poprzez zapisanie czasu trwania nagrania w zmiennej.
2. Zgodnie z procedurą przyjętą w poprzedniej lekcji, należy poprosić uczniów o zasugerowanie ulepszeń, napisanie pseudokodu i stworzenie projektu Dyktafonu 3.0.

Podsumowanie

[10 minut]

Dokumentacja projektu

1. Zadaniem uczniów będzie stworzenie opisu dla zaprojektowanego dyktafonu, z uwzględnieniem zastosowanych ulepszonych funkcji.

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można poprosić uczniów o zebranie opinii na temat projektu od różnych osób (np. rodziców, nauczycieli, przyjaciół, itp.).
- Można też przeprowadzić ćwiczenie polegające na wprowadzaniu poprawek do projektów kolegów i koleżanek.

Lekcja 5

Kontroler do gier

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Rozróżnić pomiędzy programowaniem sceny a programowaniem urządzeń.
- Przeanalizować i opisać sposób funkcjonowania istniejącego projektu.
- Stworzyć program w mBlock używając CyberPi do kontrolowania duszkami.
- Zmodyfikować istniejący program.

★ Informacje ogólne

Podczas tej lekcji uczniowie zmienią CyberPi w kontroler do gier łącząc procesy programowania urządzeń i programowania w mBlock. Uczniowie zapoznają się z przykładowymi programami pozwalającymi kontrolować ruch duszków animowanych za pomocą CyberPi. Następnie, z wykorzystaniem metody programowania w parach, uczniowie zmodyfikują istniejącą grę, aby wykorzystać CyberPi jako kontroler do niej.

🔗 Kluczowe zagadnienia

- Połączenie programowania w mBlock i programowania urządzeń
- Programowanie w parach
- Dekompozycja i abstrakcja

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)

- Przykładowy program:
CyberPi – Lekcja 5 – Pościg CyberPi – Lekcja 5 – Gwiazdne przygody

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program:
CyberPi – Lekcja 5 – Pościg CyberPi – Lekcja 5 – Gwiazdne przygody

 **Standardy**

Typ	Punkt	Standard
CSTA	2-AP-16	Włączanie istniejących kodów, plików i bibliotek do autorskich programów z podaniem informacji o zapożyczeniach.
ISTE	2c	Zrozumienie i szacunek dla praw i obowiązków związanych z korzystaniem i dzieleniem się własnością intelektualną.
K12 CS Framework	Practice 4-1	Identyfikowanie złożonych, interdyscyplinarnych problemów świata rzeczywistego, które mogą być rozwiązane programistycznie.
K12 CS Framework	Practice 7-3	Rozwijanie własnych pomysłów z poszanowaniem praw własności intelektualnej i ze wskazywaniem źródła zapożyczonych elementów.

 **Plan zajęć (45 minut)**

Czas	Treść
5 minut	Rozgrzewka <ul style="list-style-type: none"> Dyskusja o grach wideo
15 minut	Teoria i praktyka <ul style="list-style-type: none"> Przetestowanie przykładowych gier Abstrakcja i dekompozycja
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> Programowanie w parach Modyfikacja przykładowego projektu
5 minut	Podsumowanie <ul style="list-style-type: none"> Poszanowanie własności intelektualnej Rozszerzenie lekcji

Rozgrzewka [5 minut]

Dyskusja o grach wideo

1. W ramach rozgrzewki należy polecić uczniom zadanie sobie w parach następujących pytań:
 - Czy grasz w gry wideo? Jeśli tak, jak często?
 - Na jakim urządzeniu lub systemie lubisz grać w gry?
 - Jaki jest Twój ulubiony kontroler do gier wideo?
 - Jakie cechy sprawiają, że jest to Twój ulubiony kontroler?

Teoria i praktyka [15 minut]

Przetestowanie przykładowych gier

1. W części teoretyczno-praktycznej należy podzielić uczniów w pary, oznaczając partnera A i B.
2. Każdy z uczniów otwiera następnie przykładowy projekt:
 - **Partner A** Lekcja 5 – Pościg
 - **Partner B** Lekcja 5 – Gwiazdne przygody
3. Zadaniem uczniów będzie podłączenie CyberPi w trybie Live i zagranie w grę.
4. Każdy z projektów łączy programowanie sceny i programowanie urządzeń w celu stworzenia gry mBlock, kontrolowanej za pomocą CyberPi. Warto omówić z uczniami następujące pojęcia:

Pojęcie	Definicja
programowanie sceny	Sekwencje bloków programistycznych, które oddziałują na duszki animowane i tło sceny w mBlock.
programowanie urządzeń	Sekwencje bloków programistycznych, które wchodzi w interakcję z fizycznymi urządzeniami komputerowymi.

5. Należy poprosić uczniów o sprawdzenie przydzielonego im przykładowego programu i rozróżnienie pomiędzy programowaniem sceny i programowaniem urządzenia.
To ostatnie dotyczy urządzeń wymienionych w zakładce Devices, podczas gdy programowanie sceny

dotyczy duszków i tła wymienionych w zakładkach Sprites i Background.

Abstrakcja i dekompozycja

6. Z wykorzystaniem myślenia programistycznego oraz abstrakcji i dekompozycji uczniowie zbadają przykładowe projekty i określą, jak zmodyfikować istniejącą grę, aby dodać do niej kontroler CyberPi. Proces abstrakcji pomoże zignorować lub odfiltrować części programu, które są niepotrzebne do wykonania zadania, natomiast proces dekompozycji umożliwi rozdzielanie programowania sceny i programowania urządzeń, które są potrzebne do stworzenia kontrolera.

Pojęcie	Definicja
abstrakcja	Uproszczenie problemu poprzez ukrycie, odfiltrowanie lub zignorowanie niepotrzebnych szczegółów.
dekompozycja	Rozbicie problemu na mniejsze części.

7. Należy poinstruować uczniów, aby uważnie przyjrzeni się przydzielonej im grze i wykonali poniższe zadanie:

Kontroler do gier	
Problem	Twoim zadaniem jest wykorzystanie istniejącego programu i dodanie do niego kontrolera gier CyberPi. Użyj przykładowego programu jako wskazówki, jak wykonać to zadanie.
Zadanie	Objaśnij w formie pisemnej, w jaki sposób CyberPi i scena współdziałają w przykładowej grze.

Myślenie programistyczne	
Abstrakcja	Należy zachęcić uczniów do wykorzystania procesu abstrakcji w celu odfiltrowania niepotrzebnych części programu, które nie wpływają na interakcję z CyberPi. Przykłady: Gwiazdne przygody <ul style="list-style-type: none"> ○ Tytuł, piłka, asteroidy i duszki życia

	<ul style="list-style-type: none"> ○ Tło ○ Skrypty „gameStart” <p>Pościg</p> <ul style="list-style-type: none"> ○ Tytuł i duszki nietoperze ○ Tło ○ Skrypty „gameStart”
Dekompozycja	<p>Należy zachęcić uczniów do dekompozycji poszczególnych części programu. Warto upewnić się, że uczniowie uwzględniają takie szczegóły jak:</p> <ul style="list-style-type: none"> ○ Które przyciski na CyberPi są używane? ○ Które duszki są kontrolowane? ○ W jaki sposób przyciski na CyberPi kontrolują duszki?

8. Po wykonaniu powyższego ćwiczenia uczniowie powinni być zaznajomieni z następującymi blokami:

Kategoria	Blok	Funkcja
 Events		<p>Wysyłanie wiadomości z jednego urządzenia, oraz od jednego duszka lub tła do drugiego. Używane do synchronizacji akcji.</p>
 Control		<p>Instrukcja warunkowa Wykonuje zagnieżdżone akcje, jeśli warunek jest spełniony.</p>
 Sensing		<p>Używany z instrukcją warunkową do wykrywania, czy joystick jest wciśnięty lub poruszony przez użytkownika.</p>

Rozwiązywanie zadania [20 minut]

Programowanie w parach

1. W tej części lekcji uczniowie będą pracować w parach, aby dodać kontroler gier CyberPi do istniejącego projektu. Upřednio należy zapoznać uczniów z rolami partnerów podczas programowania w parach:

Programowanie w parach

Nawigator	Partner, który śledzi ogólny obraz i pomaga określić, jakie kroki należy dalej wykonać.
Driver	Partner, który pisze kod na komputerze.

*definicje zaczerpnięte z Code.org – CSD Unit 1

- Uczniowie powinni wymieniać się rolami co 3-5 minut.

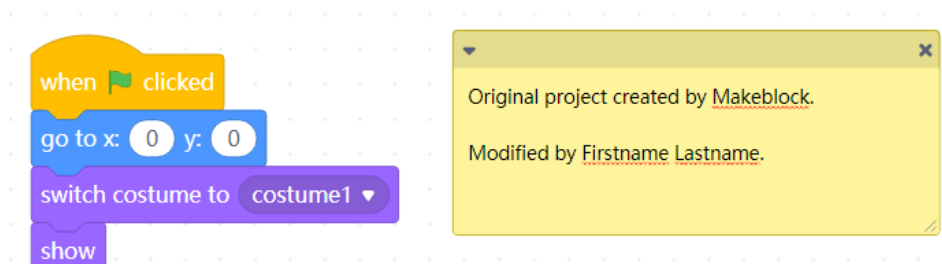
Modyfikacja istniejącego projektu

- Kliknąć przycisk **Tutorials** w prawym górnym rogu. Wybrać **Example Programs**.
- Kliknąć **Stage**, aby wyświetlić przykładowe programy.
- Podczas programowania w parach zadaniem uczniów będzie wybranie przykładowego programu i zmodyfikowanie jego skryptów, aby dodać kontroler do gry.

Podsumowanie [5 minut]

Poszanowanie własności intelektualnej

- Na zakończenie lekcji należy omówić zagadnienie poszanowania własności intelektualnej i podawania do wiadomości faktycznych twórców danych elementów.
- Należy polecić uczniom dodanie komentarza do swojego projektu z podaniem autorstwa konkretnych elementów. Aby dodać komentarz, należy kliknąć prawym przyciskiem myszy na obszarze skryptu i wybrać opcję Add comment.



Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można poinstruować uczniów, aby zapoznali się z pojęciami: własność intelektualna, prawa autorskie, creative commons i cytaty.
- Można również polecić im stworzenie nowej gry, która łączy w sobie programowanie sceny i programowanie urządzeń,
- ewentualnie dodanie do projektu tytułu i instrukcji gry.



Lekcja 6

Czujnik pomiarowy

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Opisać jak czujniki CyberPi wykrywają zjawiska w otoczeniu.
- Usuwać błędy w programach w mBlock.
- Wykonać dokumentację programu za pomocą komentarzy w mBlock.

★ Informacje ogólne

Podczas lekcji uczniowie dowiedzą się, w jaki sposób wbudowane czujniki CyberPi przedstawiają głośność i natężenie światła w otaczającym środowisku. Poznają też zasady reprezentacji danych i tworzenia wykresów mierzonych wartości z czujników.

🔗 Kluczowe zagadnienia

- Reprezentacja danych
- Debugowanie programów
- Zasada działania czujników

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program:

CyberPi – Lekcja 6 – Czujnik pomiarowy

CyberPi – Lekcja 6 – Czujnik pomiarowy V2

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program:
CyberPi – Lekcja 6 – Czujnik pomiarowy

Standardy

Typ	Punkt	Standard
CSTA	2-AP-19	Wykonywanie dokumentacji programu w celu ułatwienia śledzenia, testowania i usuwania błędów.
CSTA	2-CS-03	Systematyczne identyfikowanie i rozwiązywanie problemów z urządzeniami komputerowymi i ich komponentami.
ISTE	5b	Zbieranie lub identyfikowanie odpowiednich zestawów danych, używanie narzędzi cyfrowych do ich analizy oraz przedstawianie danych na różne sposoby w celu ułatwienia rozwiązywania problemów i podejmowania decyzji.

Plan zajęć (45 minut)

Czas	Treść
5 minut	Rozgrzewka <ul style="list-style-type: none"> • Dane i społeczeństwo
15 minut	Teoria i praktyka <ul style="list-style-type: none"> • Analiza danych z czujników • Wykres z czujnika dźwięku
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Wykres z czujnika światła
5 minut	Podsumowanie <ul style="list-style-type: none"> • Dokumentacja • Rozszerzenie lekcji

Ćwiczenia

Rozgrzewka [5 minut]

Dane i społeczeństwo


1. W ramach rozgrzewki należy zaprezentować klasie przykład najnowszego rozwiązania wykorzystywanego w inteligentnych urządzeniach domowych i omówić, w jaki sposób rozwiązanie te wykorzystuje różne czujniki, aby zapewnić konsumentom bezpieczeństwo, wygodę i automatyzację.

Teoria i praktyka [15 minut]

Analiza danych z czujników

1. Uczniowie otwierają oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.
2. Otworzyć przykładowy projekt Lekcji 6 - Czujnik pomiarowy. Kliknąć zieloną flagę, aby uruchomić program. Prześledzić wartości dla CyberPiVolume i CyberPiLightIntensity.
3. Następnie należy poinstruować uczniów, aby uruchomili program i obserwowali wartości czujników oraz wykresy w różnych przypadkach, w tym:
 - CyberPi umieszczony na ławce
 - Zastąpienie CyberPi (natężenie światła powinno się zmniejszyć)
 - Świecenie latarką na CyberPi (natężenie światła powinno się zwiększyć)
 - Zachowanie ciszy (poziom głośności powinien się zmniejszyć)
 - Klaskanie lub mówienie w pobliżu CyberPi (poziom głośności powinien się zwiększyć)
4. Uczniowie powinni również spróbować określić minimalne i maksymalne wartości, które czujniki rejestrują i przekazują do komputera. Zakres rejestrowanych wartości dla czujnika światła i czujnika dźwięku wynosi od 0 do 100.


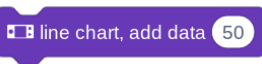
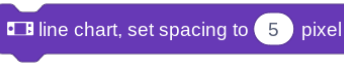
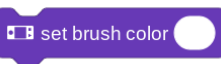
5. Należy zapoznać uczniów z blokami używanymi dla czujników w CyberPi:

Kategoria	Blok	Funkcja
		Przechowuje wartość liczbową reprezentującą natężenie światła zarejestrowane przez czujnik

 Sensing		światła na CyberPi.
		Przechowuje wartość liczbową reprezentującą głośność zarejestrowaną przez czujnik dźwięku na CyberPi.

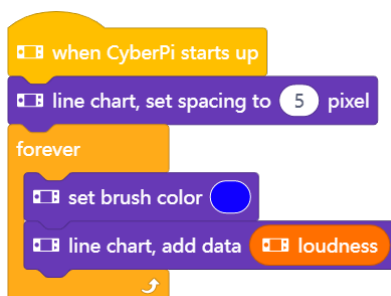
Wykres z czujnika dźwięku

6. Należy zapoznać uczniów z blokami używanymi do tworzenia wykresów liniowych na CyberPi:

Kategoria	Blok	Funkcja
 Display		Umieszcza punkt na wykresie liniowym na wyświetlaczu CyberPi.
		Zmienia poziomy odstęp wykresu liniowego.
		Ustawia kolor wykresu liniowego.

7. Należy otworzyć oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.

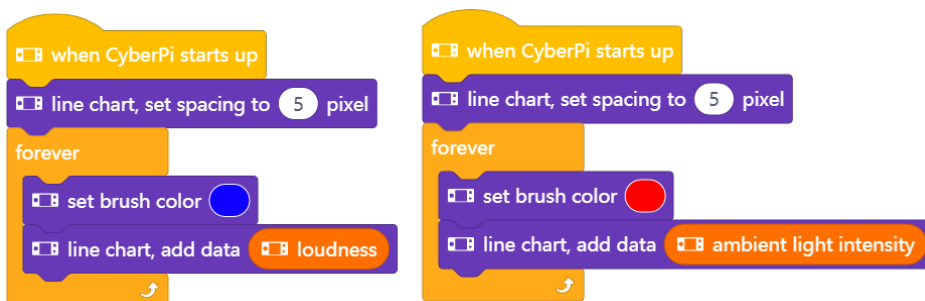
8. Należy polecić uczniom zbudowanie poniższego skryptu i przetestowanie programu:



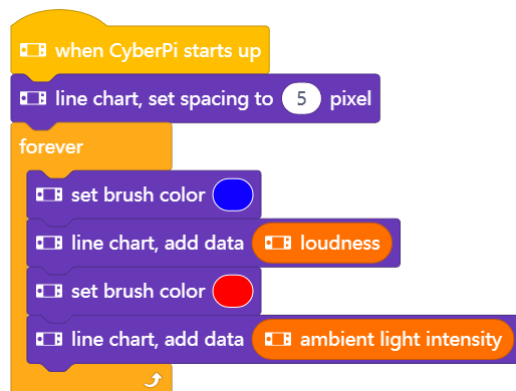
Rozwiązywanie zadania [20 minut]

Wykres z czujnika światła

1. Należy poinstruować uczniów, aby zmodyfikowali kod tak, aby zawierał wykres o czerwonych liniach dla natężenia światła. CyberPi powinien pokazywać oba wykresy jednocześnie.
2. Niektóre rozwiązania mogą wydawać się poprawne na pierwszy rzut oka. Należy jednak zachęcać uczniów do krytycznego myślenia i dokładnego przetestowania swojego rozwiązania. Dla przykładu, poniższy kod nie wyświetla prawidłowo wykresu danych. Podczas testowania uczniowie powinni zauważyć, że głośne mówienie w pobliżu CyberPi sprawia, że zarówno czerwony, jak i niebieski słupek rośnie.



3. Należy motywować uczniów do wytrwałości. Powinni poprawiać swój kod do momentu, aż uzyskają poprawne rozwiązanie. Poniżej przedstawiono poprawne rozwiązanie:



Podsumowanie [5 minut]

1. Podczas programowania pomocne może być prowadzenie dokumentacji, aby ułatwić innym śledzenie, testowanie i usuwanie błędów. Często zdarza się, że nad jednym projektem pracuje wielu programistów. Należy poinstruować uczniów, aby użyli komentarzy do wyjaśnienia sposobu działania kodu.

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można polecić uczniom zaprogramowanie diod LED lub głośników tak, aby reagowały na głośność lub natężenie światła,
- ewentualnie dodanie do projektu tytułu i instrukcji gry.
- Innym pomysłem może być zaprogramowanie duszka tak, aby zmieniał kostium w zależności od natężenia światła lub głośności. (Uwaga, do przechowywania każdej wartości z czujnika potrzebna jest zmienna).

Lekcja 7

Mikser kolorów

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Używać zmiennych do przechowywania wartości.
- Modyfikować zmienne na podstawie danych wprowadzonych przez użytkownika.
- Napisać program w mBlock, który wykonuje określoną akcję, jeśli spełniony jest dany warunek.

★ Informacje ogólne

Podczas lekcji uczniowie zapoznają się ze pojęciem zmiennych w celu stworzenia miksera kolorów CyberPi. Program będzie wykorzystywał joystick i przyciski do sterowania wartościami kolorów R, G, B wszystkich wbudowanych diod LED. Zastosowanie instrukcji warunkowych zapewni, że wartości R, G, B będą zawsze znajdować się w określonym zakresie.

🔗 Kluczowe zagadnienia

- Przechowywanie danych za pomocą zmiennych
- Zastosowanie wyrażeń warunkowych

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 7 – Mikser kolorów

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 7 – Mikser kolorów

Standardy

Typ	Punkt	Standard
CSTA	2-CS-02	Tworzenie projektów, które łączą komponenty sprzętowe i programowe w celu gromadzenia i wymiany danych.
CSTA	2-AP-11	Tworzenie zmiennych, które reprezentują różne typy danych i wykonywanie operacji na ich wartościach.
CSTA	2-DA-07	Reprezentowanie danych przy użyciu różnych schematów kodowania.

Plan zajęć (45 minut)

Czas	Treść
5 minut	Rozgrzewka <ul style="list-style-type: none"> • Gra oparta o instrukcje warunkowe
15 minut	Teoria i praktyka <ul style="list-style-type: none"> • Przechowywanie danych za pomocą zmiennych • Zastosowanie instrukcji warunkowych
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Ukończenie programu
5 minut	Podsumowanie <ul style="list-style-type: none"> • Zmienne • Rozszerzenie lekcji

Rozgrzewka [5 minut]

Gra oparta o instrukcje warunkowe

- W ramach rozgrzewki uczniowie zagrają w grę podobną do dobrze znanej zabawy Szymon Mówi, aby lepiej zrozumieć pojęcie instrukcji warunkowych. Po przeczytaniu każdej z poniższych instrukcji, uczniowie powinni wykonać polecenie. (Można również wykorzystać własne polecenia)
 - JEŚLI** w twoim imieniu występuje litera "S", **TO** podnieś rękę.
 - JEŚLI** masz kota, **TO** zaklaszcz w dłonie.
 - JEŚLI** uprawiasz sport, **TO** tupnij nogą.
 - JEŚLI** masz na sobie skarpetki, **TO** dotknij swoich stóp.
 - JEŚLI** twoje ulubione lody to lody czekoladowe, **TO** powiedz "mniam".
- Należy poinformować uczniów, że polecenia te są przykładami instrukcji warunkowych. Podczas programowania instrukcje te są używane do wykonywania określonych czynności, jeśli warunek jest spełniony.

Teoria i praktyka [15 minut]

Przechowywanie danych za pomocą zmiennych

- Na początku lekcji dobrze jest omówić poniższe pojęcia:

Pojęcie	Definicja*
zmienna	Wielkość, która może przyjmować różne wartości.





*definicje zaczerpnięte z Code.org – CSD Unit 1




- Następnie należy omówić z uczniami poniższe zadanie i pseudokod:

Mikser kolorów	
Opis projektu	Stworzenie programu wykorzystującego joystick i przyciski na CyberPi do sterowania wartościami R, G, B wszystkich wbudowanych diod LED.
Pseudokod	Po pociągnięciu joysticka w górę: Zwiększenie wartości koloru czerwonego (R) wszystkich diod LED o 5

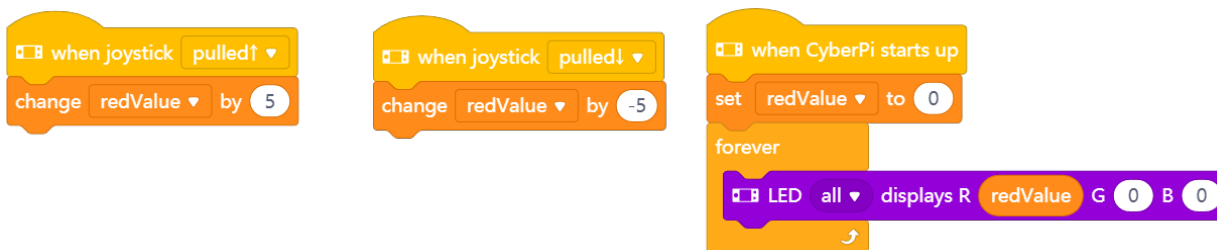
	<p>Po pociągnięciu joysticka w dół: Zmniejszenie wartości koloru czerwonego (R) wszystkich diod LED o 5</p> <p>Po pociągnięciu joysticka w prawo: Zwiększenie wartości koloru zielonego (G) wszystkich diod LED o 5</p> <p>Po pociągnięciu joysticka w lewo: Zmniejszenie wartości koloru zielonego (G) wszystkich diod LED o 5</p> <p>Naciśnięcie przycisku A: Zwiększenie wartości koloru niebieskiego (B) wszystkich diod LED o 5</p> <p>Naciśnięcie przycisku B: Zmniejszenie wartości koloru niebieskiego (B) wszystkich diod LED o 5</p>
--	--

- Aby zaprogramować ten projekt, należy użyć zmiennych do przechowywania wartości każdej z diod LED. Zmienne będą zaczynać się od zera (0) po uruchomieniu CyberPi i będą regulowane za pomocą joysticka lub przycisków.
- Uczniowie otwierają oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.
- Przejdź do sekcji **Variables** w **Obszarze bloków**. Kliknąć przycisk **Make a Variable**.
- Nadać nowej zmiennej nazwę **redValue** i pozostawić zaznaczoną opcję **For all sprites**.
- Następnie należy zapoznać uczniów z następującymi nowymi blokami, które są dostępne w sekcji **Variables**:

Kategoria	Blok	Funkcja
 Variables		Ustawia zmienną na określoną wartość.
		Modyfikuje zmienną o określoną wartość (dodatnią lub ujemną).
		Zwraca aktualną wartość zapisaną w zmiennej.

 Operators		Porównuje dwie wartości i określa, czy pierwsza wartość jest mniejsza od drugiej. Zwraca TRUE lub FALSE.
		Porównuje dwie wartości i określa, czy pierwsza wartość jest większa od drugiej. Zwraca TRUE lub FALSE.

8. Następnie należy polecić uczniom zbudowanie poniższego skryptu i przetestowanie programu, aby zaobserwować, jak sterowanie joystickiem w górę/dół zmienia wartość koloru czerwonego wszystkich diod LED:



```

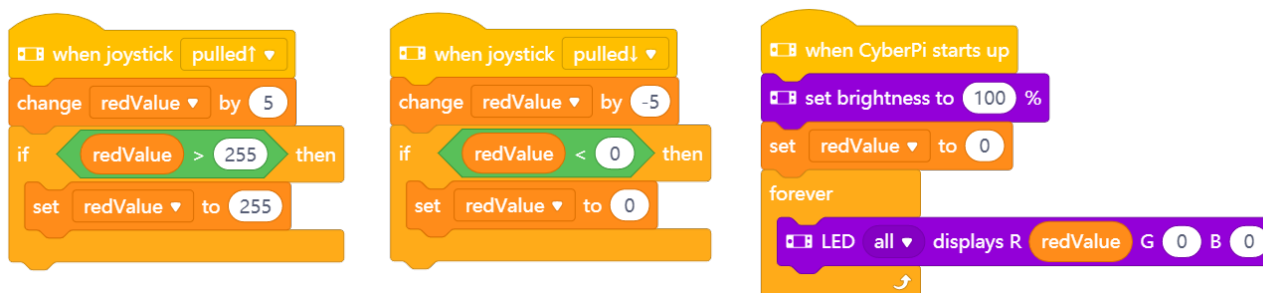
when joystick pulled up
  change redValue by 5

when joystick pulled down
  change redValue by -5

when CyberPi starts up
  set redValue to 0
  forever
    LED all displays R redValue G 0 B 0
  
```

Zastosowanie instrukcji warunkowych

9. Należy przypomnieć uczniom, że wartości R, G, B przyjmują zakres od 0 do 255. W powyższym programie zmienna **redValue** może zostać zmieniona na wartość spoza tego zakresu. Za pomocą instrukcji warunkowych można kontrolować wartości minimalne i maksymalne. Należy poinstruować uczniów, aby zmodyfikowali kod i dodali do programu następujące instrukcje warunkowe:



```

when joystick pulled up
  change redValue by 5
  if redValue > 255 then
    set redValue to 255

when joystick pulled down
  change redValue by -5
  if redValue < 0 then
    set redValue to 0

when CyberPi starts up
  set brightness to 100 %
  set redValue to 0
  forever
    LED all displays R redValue G 0 B 0
  
```

Uczniowie mogą również dodać blok ustawiający jasność na 100% przy starcie programu.

10. Instrukcje warunkowe nie pozwalają na ustawienie wartości RedValue większej niż 255 lub mniejszej niż 0.

Rozwiązywanie zadania

[20 minut]

Ukończenie programu

1. Należy poinstruować uczniów, aby ukończyli program w oparciu o pseudokod podany wcześniej w ramach lekcji. Poniżej przedstawiono przykład ukończonego programu:

```

when CyberPi starts up
  set brightness to 100 %
  set redValue to 0
  forever
    LED all displays R redValue G greenValue B blueValue

when joystick pulled up
  change redValue by 5
  if redValue > 255 then
    set redValue to 255

when joystick pulled down
  change redValue by -5
  if redValue < 0 then
    set redValue to 0

when joystick pulled right
  change greenValue by 5
  if greenValue > 255 then
    set greenValue to 255

when joystick pulled left
  change greenValue by -5
  if greenValue < 0 then
    set greenValue to 0

when button A pressed
  change blueValue by 5
  if blueValue > 255 then
    set blueValue to 255

when button B pressed
  change blueValue by -5
  if blueValue < 0 then
    set blueValue to 0
  
```

2. Pomocne może być dodanie następującego skryptu wewnątrz pętli nieskończonej, aby wyświetlić wartości zmiennych na wyświetlaczu CyberPi:

```

show join join R: redValue join join G: greenValue join B: blueValue at center of screen by small pixel
  
```

Podsumowanie
[5 minut]

Omówienie zmiennych

1. Na zakończenie lekcji należy omówić z uczniami, w jaki sposób zmienne są wykorzystywane do przechowywania informacji w różnych urządzeniach i aplikacjach komputerowych. Można wykorzystać poniższe przykłady:
 - o Urządzenia typu fitness tracker rejestrujące liczbę kroków.
 - o Samochody (z cyfrowym wyświetlaczem) rejestrujące przejechany dystans.
 - o Urządzenia mobilne rejestrujące poziom naładowania baterii i podające go w procentach.
 - o Karty lojalnościowe rejestrujące dokonane zakupy i naliczone punkty uprawniające do zdobycia nagrody.
 - o Gry wideo rejestrujące dane o wynikach.

Rozszerzenie lekcji

- W ramach rozszerzenia można polecić uczniom dodanie do projektu tytułu i instrukcji.
- Innym pomysłem jest utworzenie zmiennej LEDNumber i użycie środkowego przycisku joysticka, aby kontrolować, która dioda jest zmieniana.

Lekcja 8

Miernik siły

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Napisać program w mBlock do zapisu wyników.
- Napisać program, który wykonuje działania przez określony czas.
- Wyświetlać tekst na wyświetlaczu CyberPi.

★ Informacje ogólne

Podczas lekcji uczniowie stworzą grę, w której gracz będzie potrząsać CyberPi przez dziesięć sekund. Gra będzie zaprogramowana tak, aby wszystkie wyniki powyżej 50 wstrząsów były zapisywane.

🔗 Kluczowe zagadnienia

- Zapisywanie wyników
- Korzystanie z timera CyberPi
- Wyświetlanie tekstu na wyświetlaczu CyberPi

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 8 – Miernik siły

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- CyberPi z kablem USB-C
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 8 – Miernik siły

Standardy

Typ	Punkt	Standard
CSTA	2-CS-02	Tworzenie projektów, które łączą komponenty sprzętowe i programowe w celu gromadzenia i wymiany danych.
CSTA	2-AP-11	Tworzenie zmiennych, które reprezentują różne typy danych i wykonywanie operacji na ich wartościach.

Plan zajęć (45 minut)

Czas	Treść
5 minut	Rozgrzewka <ul style="list-style-type: none"> • Omówienie zmiennych
15 minut	Teoria i praktyka <ul style="list-style-type: none"> • Pomiar siły • Zapisywanie wyników
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Stworzenie Miernika siły 2.0
5 minut	Podsumowanie <ul style="list-style-type: none"> • Wykrywanie ruchu • Rozszerzenie lekcji

Rozgrzewka [5 minut]

Omówienie zmiennych

1. W ramach rozgrzewki należy wspólnie z uczniami omówić pojęcie zmiennych i wyjaśnić, w jaki sposób mogą być one używane do śledzenia wyniku w projekcie mBlock. Należy przypomnieć uczniom, jak tworzy się i modyfikuje zmienne w projekcie mBlock.

Teoria i praktyka [15 minut]

Pomiar siły

1. Następnie należy omówić z uczniami poniższe zadanie i pseudokod:

Miernik siły 1.0	
Opis projektu	Stworzenie gry, w której użytkownik potrząsa CyberPi przez 10 sekund i zdobywa punkt za każdym razem, gdy siła potrząsania przekroczy 50.
Pseudokod	<p>Naciśnięcie przycisku B: Ustawienie wyniku na zero (0) Zresetowanie timera Powtarzanie przez 10 sekund Jeśli siła wstrząsów jest większa niż 50, Zmiana wyniku o jeden punkt (1)</p> <p>Naciśnięcie przycisku A: Wyświetlenie wyniku na wyświetlaczu CyberPi</p>

2. Uczniowie otwierają oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Live.

3. Należy zapoznać uczniów z następującymi blokami:

Kategoria	Blok	Funkcja
Control		Instrukcja pętli Wykonuje zagnieżdżone akcje, aż warunek osiągnie wartość TRUE.
Sensing		Ustawia timer na CyberPi na zero (0).
Motion Sensing		Zwraca wartość reprezentującą siłę potrząśnięcia CyberPi.
Display		Wyświetla tekst na wyświetlaczu CyberPi w określonej pozycji i rozmiarze.

4. Należy polecić uczniom zbudowanie poniższego skryptu i przetestowanie programu:

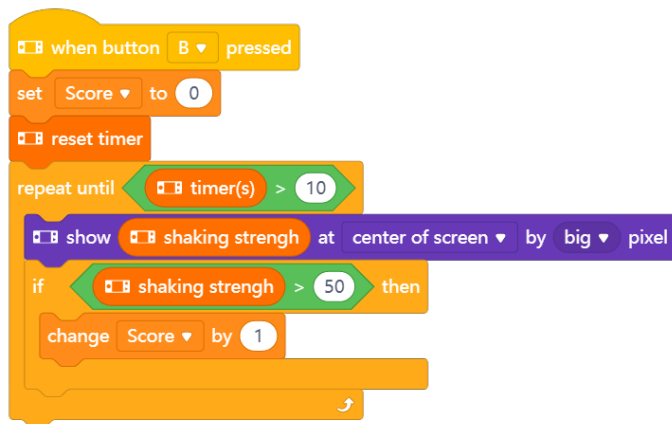
```

when button B pressed
  reset timer
  repeat until timer(s) > 10
    if shaking strength > 50 then
      show shaking strength at center of screen by big pixel
  
```

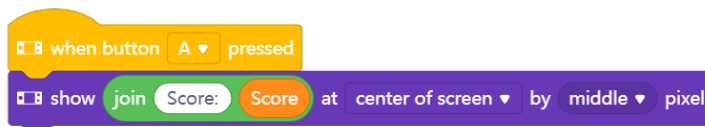
Zapisywanie wyników

5. Nadać zmiennej nazwę **score** i pozostawić zaznaczoną opcję **For all sprites**.

6. Polecieć uczniom dodanie do programu następujących skryptów:



7. Gdy już CyberPi jest zaprogramowany, aby śledził wynik, należy dodać następujące skrypty, aby wyświetlały wynik na ekranie po wciśnięciu przycisku A:



8. Zadaniem uczniów będzie zagranie w grę i odnotowanie uzyskanych punktów.

Rozwiązywanie zadania [20 minut]

Stworzenie Miernika siły 2.0

- Po stworzeniu prostej wersji gry opartej o Miernik Siły, można do niej dodać ulepszenia. Należy polecieć uczniom wskazanie obszarów i aspektów, które można by ulepszyć w projekcie. Kilka pomysłów na ulepszenia:
 - Zastosowanie diody LED w celu sygnalizowania użytkownikowi, kiedy gra jest uruchomiona.
 - Dodanie efektu dźwiękowego przy każdym zdobytym punkcie.
 - Dodanie efektu dźwiękowego po upływie czasu.
 - Dodanie tytułu i instrukcji.
 - Użycie joysticka i zmiennej do modyfikacji poziomu trudności gry (np. siła wstrząsów dla wersji łatwej: 30, średniej: 50, trudnej: 70).
- Zadaniem uczniów będzie opracowanie planu i pseudokodu dla ulepszeń, które chcieliby dodać do swojego projektu

3. Miernika Siły 2.0.

Podsumowanie [5 minut]

Wykrywanie ruchu

1. CyberPi jest wyposażony w 3-osiowy żyroskop i 3-osiowy akcelerometr, które wykrywają ruch, przyspieszenie i wibracje. Blok siły wstrząsów wykorzystuje te komponenty do określenia, jak mocno urządzenie jest potrząsane. W ramach podsumowania należy poprosić uczniów o stworzenie listy urządzeń, które wykorzystują żyroskop. Przykłady:
 - o Telefony komórkowe zmieniające orientację ekranu w oparciu o obrót urządzenia.
 - o Ekran w telefonach komórkowych podświetlające się, gdy urządzenie jest podnoszone.
 - o Kontrolery gier wideo wykrywające ruch.
 - o Odkurzacze roboty wykrywające upadek lub wywrócenie.

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można polecić uczniom zaprogramowanie diod LED tak, aby zapalały się stopniowo w zależności od siły nacisku (patrz przykład).
- Innym pomysłem jest zaprogramowanie gry dla dwóch graczy, w której układ śledzi wyniki obu graczy, porównuje je i ogłasza zwycięzcę.

Lekcja 9

Alarm prezentowy

Tematyka: Informatyka

Wiek uczniów: 11-18 lat

Czas: 45 minut

Stopień trudności: początkujący

★ Cele lekcji

Po ukończeniu lekcji uczniowie będą umieli:

- Wykorzystywać technologię sieci bezprzewodowych do komunikacji pomiędzy urządzeniami.
- Rozwiązywać problemy z wykorzystaniem rozwiązań programistycznych.
- Określić zastosowania komunikacji bezprzewodowej.

★ Informacje ogólne

Uczniowie wykorzystają CyberPi do stworzenia programu, który wykrywa, czy ich koledzy i koleżanki potrząsnęli prezentem urodzinowym. Dzięki wykorzystaniu komunikacji bezprzewodowej, uczniowie będą przysyłać wiadomości pomiędzy urządzeniami komputerowymi, pozwalając jednemu urządzeniu kontrolować inne.

🔗 Kluczowe zagadnienia

- Korzystanie z sieci bezprzewodowych
- Komunikacja między urządzeniami

📄 Lista niezbędnych narzędzi

Nauczyciel:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- Dwa (2) urządzenia CyberPi z kablem USB-C **lub** jedno (1) urządzenie CyberPi i jedna (1) płytki Halocode
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 9 – Alarm prezentowy

Uczeń:

- Komputer z zainstalowanym oprogramowaniem mBlock 5 lub [internetowa wersja mBlock](#)
- Dwa (2) urządzenia CyberPi z kablem USB-C **lub** jedno (1) urządzenie CyberPi i jedna (1) płytka Halocode
- Moduł akumulatora (opcjonalnie)
- Przykładowy program: CyberPi – Lekcja 9 – Alarm prezentowy

 **Standardy**

Typ	Punkt	Standard
CSTA	2-AP-13	Rozkładanie problemów na części w celu ułatwienia projektowania, wdrażania i przeglądu programów.
ISTE	5c	Rozkładanie problemów na części składowe, zdobywanie kluczowych informacji i tworzenie modeli opisowych w celu lepszego zrozumienia złożonych systemów lub ułatwienia rozwiązywania problemów.
K12 CS Framework	Practice 3-1	Identyfikowanie złożonych, interdyscyplinarnych problemów świata rzeczywistego, które mogą być rozwiązane programistycznie.
K12 CS Framework	Practice 3-2	Rozkładanie złożonych problemów świata rzeczywistego na łatwiejsze podzadania z wykorzystaniem istniejących rozwiązań lub procedur.
K12 CS Framework	Practice 3-3	Ocena wykonalności programistycznego rozwiązania problemu.

 **Plan zajęć (45 minut)**

Czas	Treść
5 minut	Rozgrzewka <ul style="list-style-type: none"> • Internet rzeczy
15 minut	Teoria i praktyka <ul style="list-style-type: none"> • Komunikacja bezprzewodowa w mBlock
20 minut	Rozwiązywanie zadania <ul style="list-style-type: none"> • Stworzenie alarmu prezentowego
5 minut	Podsumowanie <ul style="list-style-type: none"> • Pomysły na burzę mózgów • Rozszerzenie lekcji

Rozgrzewka [5 minut]

Internet rzeczy

1. Lekcję warto rozpocząć od omówienia zagadnienia Internetu Rzeczy. Należy zlecić uczniom zapoznanie się z tym tematem i zaprezentowanie innym, czego się nauczyli. Szczegółowa tematyka powinna obejmować sposoby, w jakie fizyczne urządzenia podłączone do internetu są wykorzystywane w następujących zastosowaniach:

- Inteligentne domy
- Monitorowanie zdrowia
- Transport
- Rolnictwo
- Prognozowanie pogody
- Produkcja
- Badania środowiskowe
- Wojskowość

Teoria i praktyka [15 minut]



Komunikacja bezprzewodowa w mBlock


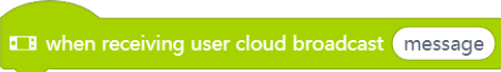

1. mBlock wykorzystuje dwa rodzaje komunikacji bezprzewodowej: Wi-Fi i LAN. Należy zapoznać się z poniższymi informacjami i blokami dla każdego rodzaju komunikacji.

Wi-Fi


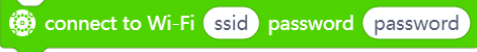




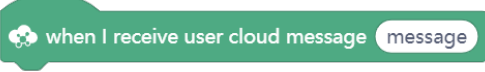

W przypadku połączenia Wi-Fi dane są udostępniane w chmurze. Dane można przysyłać między różnymi urządzeniami i projektami, korzystając z tego samego konta mBlock 5. Odległość fizyczna nie jest w tym przypadku istotna, ponieważ urządzenia te nie muszą znajdować się w tej samej lokalizacji.

Każde urządzenie musi być jednak podłączone do internetu. Poniższe bloki służą do podłączania urządzeń CyberPi, Halocode i projektu mBlock do chmury.

Kategoria	Blok	Funkcja
		Podłącza CyberPi do sieci bezprzewodowej.
		Zwraca TRUE, jeśli CyberPi jest podłączony do internetu.


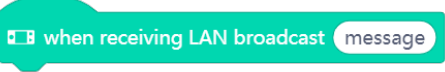
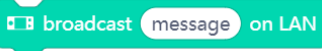
 IoT		Blok zdarzeń, który uruchamia wykonanie zagnieżdżonych akcji po otrzymaniu określonego polecenia w chmurze.
		Przesyła polecenie w chmurze.

CyberPi może komunikować się z płytką Halocode lub duszkami w mBlock poprzez przesyłanie danych w chmurze.

Halocode  Wi-Fi	   	Patrz opis bloków CyberPi powyżej.
Duszek  User cloud message	 	Patrz opis bloków CyberPi powyżej.

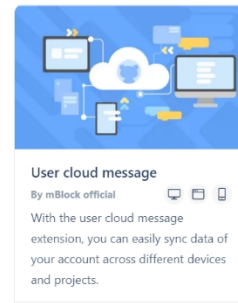
LAN


LAN (local-area network) to sieć, która łączy grupę komputerów lub urządzeń w określonym miejscu. Komputery w ramach danej grupy mogą komunikować się ze sobą. Sieć lokalną można utworzyć pomiędzy kilkoma urządzeniami CyberPi, aby umożliwić jednemu z nich kontrolowanie pozostałych.

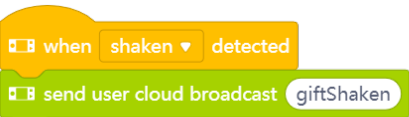

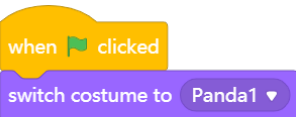
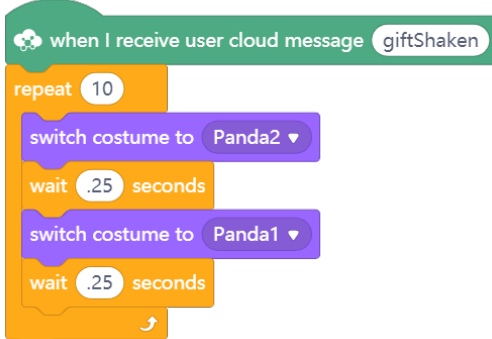
Kategoria	Blok	Funkcja
 LAN		Blok zdarzeń, który uruchamia wykonanie zagnieżdżonych akcji po otrzymaniu określonego polecenia w sieci LAN.
		Przesyła polecenie w sieci LAN.

- Należy wybrać typ komunikacji odpowiedni dla klasy i uczniów. Poniżej przedstawiono przykład funkcjonowania komunikacji bezprzewodowej w programie mBlock.

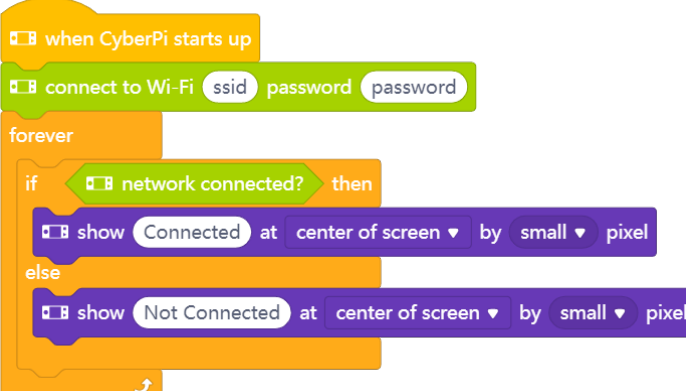
- Należy otworzyć oprogramowanie mBlock 5 zainstalowane na komputerze lub [internetową wersję mBlock 5](#). Dodać CyberPi w zakładce Devices i połączyć się w trybie Upload.



- Kliknąć przycisk  na duszku Pandy w **Obszarze bloków**. Znaleźć rozszerzenie **User cloud message** i kliknąć przycisk **+Add**.
- Poinstruować uczniów, jak zbudować poniższe skrypty:

CyberPi	Duszek
 	 

- Zaktualizować identyfikator Wi-Fi ssid i hasło dla routera bezprzewodowego w danej lokalizacji w celu podłączenia CyberPi.
- Wgrać program do CyberPi i przetestować go. Po potrząśnięciu CyberPi Panda powinna zacząć się poruszać.
- Jeśli program nie działa, należy dodać poniższy kod do skryptu CyberPi, aby rozwiązać problem z połączeniem Wi-Fi.



The image shows a Scratch script for CyberPi. It starts with 'when CyberPi starts up', followed by 'connect to Wi-Fi' with fields for 'ssid' and 'password'. Below this is a 'forever' loop containing an 'if' statement: 'if network connected? then show Connected at center of screen by small pixel', and 'else show Not Connected at center of screen by small pixel'.

Rozwiązanie zadania [20 minut]

Stworzenie alarmu prezentowego

1. W tej części lekcji należy przeanalizować z uczniami rozwiązanie poniższego problemu. Problem ten można rozwiązać na wiele sposobów; jeden z przykładowych został podany poniżej.

Alarm prezentowy	
Problem	<p>Zbliżają się urodziny jednego z Waszych kolegów. Przyjaciel ten lubi potrząsać prezentami, aby dowiedzieć się, co jest w środku. Waszym zadaniem jest stworzenie bezgłośnego alarmu, który bezprzewodowo powiadomi Was za każdym razem, gdy prezent zostanie potrząśnięty.</p> <p>W jaki sposób mBlock, CyberPi i/lub Halocode pomogą Wam w rozwiązaniu tego zadania?</p>
Sugerowane rozwiązanie	<p>Zaprogramowanie Halocode tak, aby wiadomość w chmurze była wysyłana po potrząśnięciu prezentem. Zamontowanie Halocode i akumulatora wewnątrz pudełka przed zapakowaniem prezentu.</p> <p>Zaprogramowanie CyberPi, aby odtwarzał dźwięk a diody LED migały, jeśli zostanie odebrana wiadomość z chmury, że prezent został potrząśnięty.</p> <p>Bonus: Można również zaprogramować duszka w mBlock, aby trząść się, gdy prezent zostanie potrząśnięty.</p>

2. Należy polecić uczniom napisanie pseudokodu dla **Alarmu prezentowego**, a następnie stworzenie projektu z wykorzystaniem pseudokodu jako wskazówki.

Podsumowanie [5 minut]

Pomysły na burzę mózgów

1. Komunikacja bezprzewodowa i wiadomości w chmurze eliminują ograniczenia wynikające z długości

przewodów i lokalizacji urządzeń. Należy zachęcić uczniów do przeprowadzenia burzy mózgów i stworzenia listy pomysłów na programy, które mogłyby wykorzystywać komunikację bezprzewodową.

Przykłady:

- Stacja pogodowa, która przesyła aktualną prognozę do oddzielnego urządzenia.
- Urządzenie do zbierania ankiet umiejscowione w głównym miejscu w szkole, które przekazuje wyniki do klasy.
- Walkie talkie lub wiadomości tekstowe przesyłane pomiędzy urządzeniami.

Rozszerzenie lekcji

- W ramach rozszerzenia lekcji można poprosić uczniów o opracowanie własnego projektu z wykorzystaniem komunikacji bezprzewodowej
- lub o zbadanie sieci bezprzewodowych i ich roli w społeczeństwie.