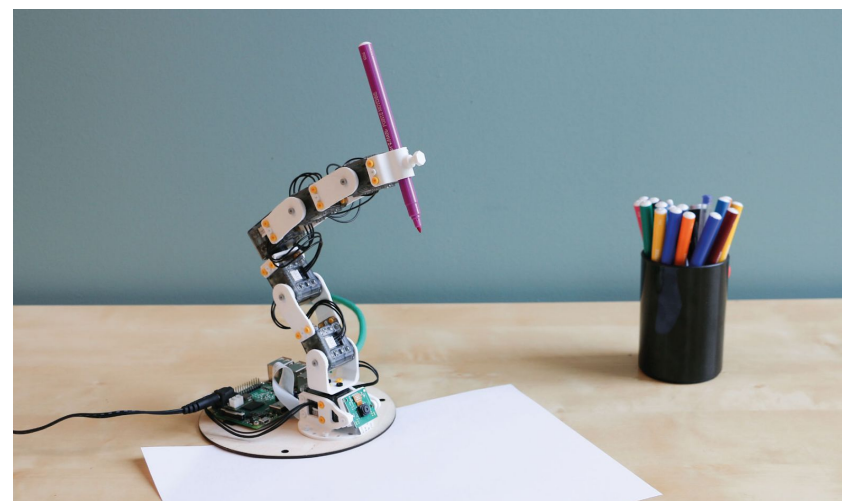


Scenariusz zajęć

Temat: Rysowanie z Poppy Ergo Jr

Cele: Uczniowie nauczą się:

- Składać robota Poppy Ergo Jr
- Używać podstawowych komend w Snap!
- Programować proste ruchy robota Poppy



Odniesienie do polskiej podstawy programowej:

Programowanie jest elementem podstawy programowej w szkole podstawowej. Scenariusz ten może być również przydatny przy wprowadzaniu pojęć z dziedziny sztuki, matematyki i fizyki.

Umiejętności i wiedza, które są rozwijane podczas realizacji tego scenariusza:

- Budowanie robota z otrzymanych komponentów
- Programowanie robota za pomocą graficznego języka programowania
- Rozwiązywanie problemów
- Współpraca

Grupa docelowa: uczniowie 7-8 klasy szkoły podstawowej

Wiek uczniów/klasa: 13-14 lat

Wielkość grupy: maksimum 10 uczniów w grupie

Czas trwania / liczba lekcji: 3 x 45-90 minut

Przygotowanie (niezbędne materiały i pomoce online):

- Poppy Ergo Jr z uchwytem na ołówek
- Komputery z parametrami odpowiednimi do obsługi programu Snap!
- Materiały pomocnicze dostępne na: <https://www.poppy-project.org/en/robots/poppy-ergo-jr>

Wprowadzenie do scenariusza (wskazówki, możliwe sposoby wykonania i sytuacje ryzykowne):

Poppy Ergo Jr to robotyczne ramię zaprojektowane do celów edukacyjnych. Składa się z sześciu motorów umożliwiających wykonywanie różnego rodzaju ruchów oraz elementów drukowanych w 3D. Ramię może mieć trzy różne zakończenia umożliwiające trzy różne sposoby interakcji z otoczeniem: chwytak, klosz lampowy, uchwyt na ołówek. Robot kontrolowany jest przez Raspberry Pi, a dodatkowa kamera umożliwia interakcję z obiektami zewnętrznymi. Jego ruchy mogą być programowane za pomocą wizualnego języka programowania, który umożliwia uczniom wykonywanie wielu interesujących eksperymentów.

Mimo że twórcy uważają, że robota tego łatwo zbudować i wszystko w jego budowie jest proste, my odkryliśmy, że z używaniem Poppy Ergo Jr wiąże się wiele wyzwań. Poniższy scenariusz zawiera podpowiedzi, jak uniknąć niektórych błędów i cieszyć się niezaprzeczalnym potencjałem, jaki robot posiada, gdy chodzi o edukację w szkole podstawowej.

Przed rozpoczęciem zajęć (do przygotowania przez nauczyciela):

- Przygotuj laboratorium komputerowe z wystarczającą liczbą stanowisk.
- Poppy zależy od Snap! (<https://snap.berkeley.edu/>), innej wersji Scratch. Zapewnij stabilne połączenie z internetem dla wszystkich komputerów i uruchom Snap! (otwórz okno przeglądarki na stronie <http://snap.berkeley.edu/run>).
- Od liczby dostępnych ramion robotycznych zależy wielkość i liczba grup warsztatowych. Każdy z robotów składa się z wielu komponentów niewielkich rozmiarów, więc nie więcej niż 2-3 uczniów powinno składać każdego z nich. Większość części można wydrukować w 3D, jeśli chce się obniżyć koszty robota. Wtedy można kupić tylko te części, których wydrukować się nie da.
- Przejdź najpierw samodzielnie przez cały proces konstruowania i programowania robota - radzimy poświęcić przynajmniej weekend na to interesujące doświadczenie zanim robot trafi do klasy.

Przebieg zajęć (około trzy lekcje/sesje):

Każda z kolejnych lekcji jest niezależnym modułem, który może być przeprowadzony autonomicznie. Najlepiej, jeśli wszyscy uczniowie przejdą wszystkie kroki ale to zależy od ich umiejętności, liczby dostępnych komputerów oraz robotów. Pewnym rozwiązaniem może być przeprowadzenie pierwszej lekcji poświęconej składaniu robotów z mniejszą grupą uczniów, złożyc 1-3 ramion, a następne lekcje przeprowadzić już dla całej klasy.

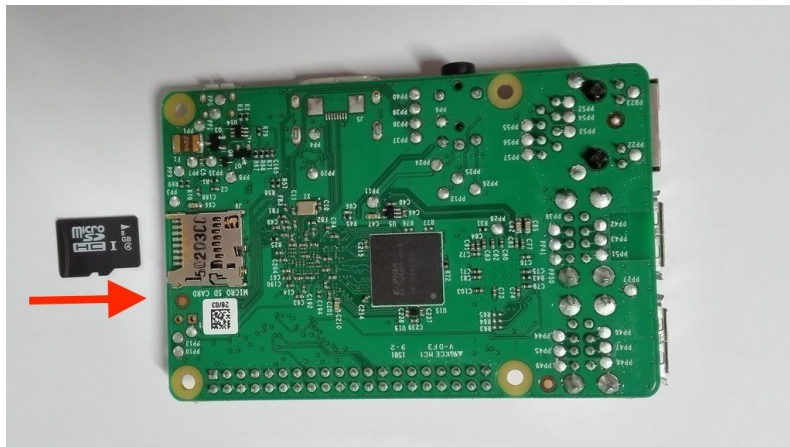
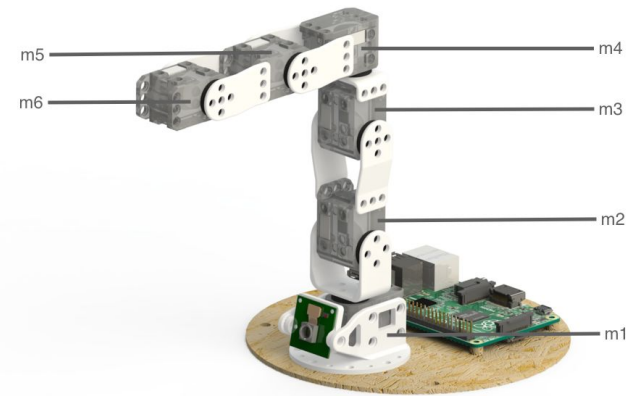
Lekcja pierwsza: Budowanie Poppy Ergo Jr

Twórcy robota udostępnili krótki [przewodnik](#), jak należy składać robota, który zawiera część poświęconą konfiguracji silników, łączeniu komponentów elektronicznych i budowaniu konstrukcji. Powinien on stanowić główne źródło danych na temat procesu, choć nie uczniowie nie muszą przeczytać go w całości. Przygotowaliśmy bowiem tutorial wideo dla początkujących, który



przedstawia proces składania robota. Warto zacząć od obejrzenia go w całości, a następnie składać, oglądając jeszcze raz. Zawsze, gdy nie są pewni kolejnego kroku, uczniowie mogą poprosić o pomoc nauczyciela.

Konfiguracja motorów: serwa powinny być połączone w ściśle określony sposób. Każdy ma unikalne ID.



Połączenia elektroniczne: włóż kartę mikro-SD do Raspberry Pi - wepchnij kartę do złącza aż usłyszysz kliknięcie.

Połączenia mechaniczne: Elementy łączone są przy pomocy specjalnych nitów składających się z dwóch części. Sposób łączenia tych części ma znaczenie, tak więc uważnie przyjrzyj się instrukcji i tutorialowi.

Dobrze jest zacząć od pokazania uczniom gotowego robota, aby pokazać, że działa. Wtedy wykażą się większą cierpliwością przy składaniu własnego.

Przed zakończeniem lekcji pozwól uczniom przetestować złożone roboty. Wejdź na stronę Poppy'ego. Kliknij przycisk "Reboot the robot", aby upewnić się, że oprogramowanie zostało uruchomione przy podpiętych wszystkich silnikach. Przejdź do części "Monitor and Control". Jeśli oprogramowanie zostało uruchomione prawidłowo, powinieneś zobaczyć zielone logo, jeśli nie, będzie czerwone. Jeśli jest czerwone, możesz szukać przyczyny w wiadomościach na stronie "What happened?" Najczęstszą przyczyną jest niewłaściwe połączenie lub konfiguracja jednego lub kilku silników.



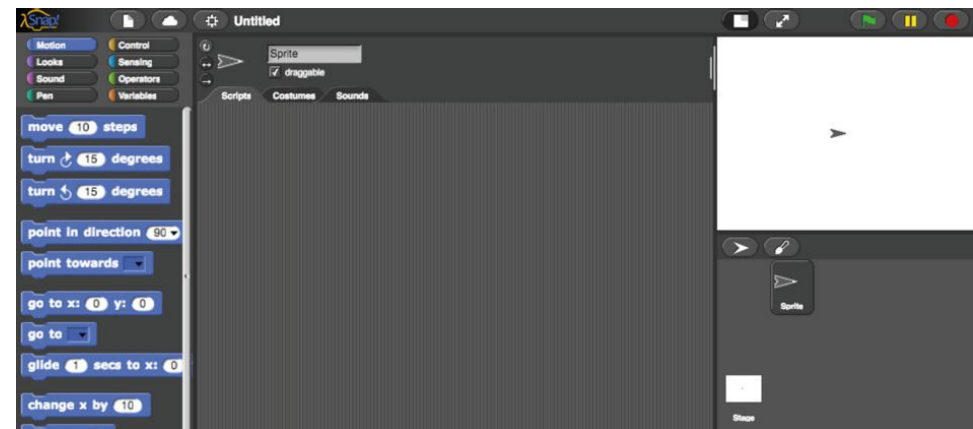
Lekcja druga: Łączenie Poppy Ergo Jr ze Snap!

Snap! jest graficznym językiem programowania, który pozwala użytkownikom tworzyć interaktywne animacje, gry i jednocześnie przyswajać wiedzę matematyczną i komputacyjną. Snap! jest oprogramowaniem typu open source, napisanym w javascript. Możesz używać go poprzez [oficjalną stronę](#) lub zainstalować [jej kopię](#) na swoim komputerze i otworzyć snap.html w przeglądarce. Uczniowie będą musieli poznać podstawowe funkcje Snap!, aby programować Poppy'ego. Temu służy też ta lekcja.

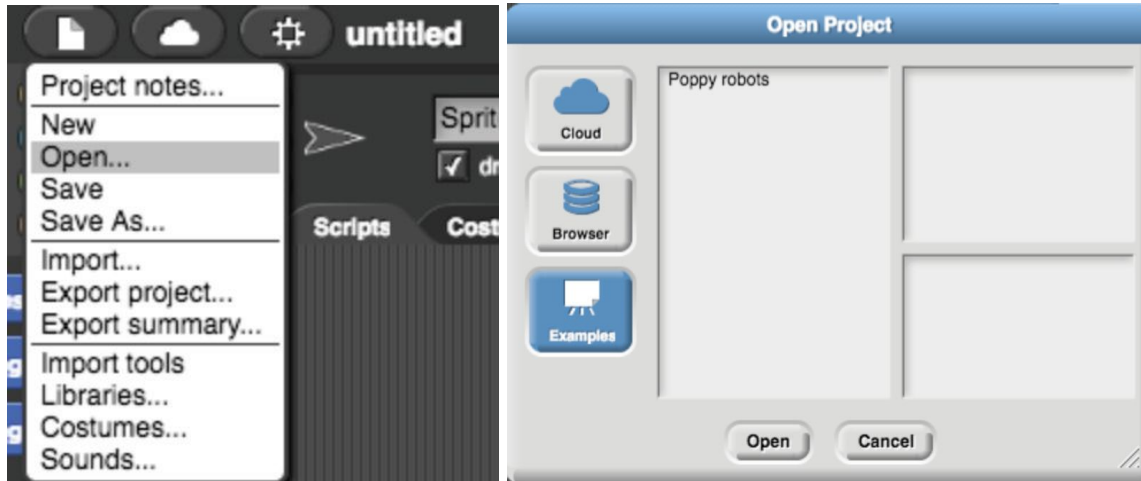
Najpierw uczniowie wybierają Snap! z głównego menu.

Snap! jest podzielony na trzy części: Przy programowaniu Poppy Ergo Jr będziecie używać głównie strefy z lewej i na środku.

- Z lewej znajduje się lista dostępnych bloków (instrukcje), zebranych w ramach różnych kategorii.
- W środku jest strefa tworzenia skryptów, gdzie składa się bloki ze sobą.
- Z prawej: u góry znajduje się wyświetlacz, który kontrolowany jest przez skrypty, poniżej umieszcza się obiekty które mogą rozwijać się powyżej.



Otwórz projekt "Poppy robots", aby uzyskać dostęp do specjalnych bloków Poppy'ego. Aby to zrobić, kliknij: File> Open> Examples> Poppy robots> Open

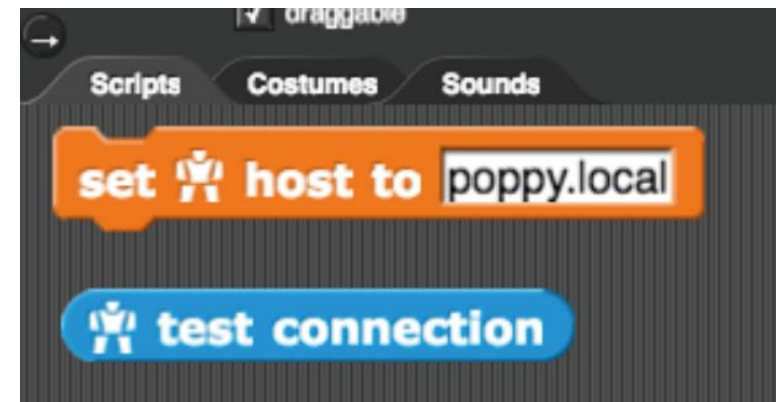


Wskazówka: Aby uzyskać informacje dotyczące bloku, kliknij prawym przyciskiem myszy na blok a następnie "pomoc".

Powinieneś zobaczyć dwa bloki, które pojawią się w zakładce skryptów.

Sprawdź, czy jesteś połączony z robotem:

- wpisz nazwę robota lub jego IP w blok (tutaj nazwą robota jest "poppy")
- Kliknij na blok "test connection", aby sprawdzić połączenie z robotem
- Jeśli pojawi się komunikat o błędzie, sprawdź połączenia silników (sprawdź też dokumentację, jeśli to konieczne).



Zapisz swój program: możesz zapisać go na swoim komputerze, klikając: Files> Export> right click> Save as albo zapisać w chmurze, ale to wymaga założenia konta użytkownika:

- Otwórz menu chmury w pasku narzędzi
- Wybierz "sign up" i zarejestruj się zgodnie z instrukcją
- Sprawdź email aby uzyskać hasło startowe

Przed aktywowaniem Poppy Ergo Jr, zawsze sprawdzaj właściwą pozycję robota. Teraz jesteś gotowy zacząć kolejne ćwiczenia!

Lekcja trzecia: Jak rysować z Poppy Ergo Jr

Podczas tej lekcji uczniowie nauczą się jak można rysować przy pomocy robota i aplikacji Snap! pisząc skrypt za pomocą bloków.

Bloki w Snap! możesz wyszukiwać poprzez:

- kolory/kategorie (każda ma inny kolor)
- słowa kluczowe (kliknięcie prawym przyciskiem na po lewej stronie i szukanie bloków) gdzie można wpisać słowa kluczowe robota, aby wybrać jedynie bloki przypisane do Poppy Ergo Jr.



Utwórz dwa skrypty widoczne poniżej, aby móc ustawić Poppy Ergo Jr w konkretnych pozycjach:

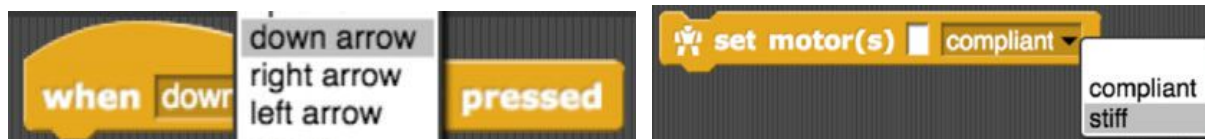


Aby to zrobić:

1. Wybierz trzy bloki i przenieś je na środkową część ekranu:



2. Kliknij na listę i wybierz odpowiednie wartości:



3. Połącz bloki:



4. Zrób to samo z drugim skrytem.

Aby połączyć bloki: przenieś blok w wybrane miejsce przy pomocy myszki. Białe pole oznacza, że bloki pasują do siebie.



Możesz kopiować i wklejać bloki i skrypty: Kliknięcie prawym przyciskiem + duplicate



Aktywuj oba skrypty (biała ramka pojawia się wokół aktywowanego skryptu). Naciskaj \downarrow lub \uparrow na klawiaturze aby kierować robotem i ustawiać go w różnych pozycjach.

Poruszaj robotem za pomocą jego silników.

My użyliśmy następującego bloku aby poruszyć kolejno silnikami robota:



Zanim zaczniesz programować ruch w Snap! upewnij się, że silniki są aktywowane (stiff) i ustaw je wszystkie w wyjściowej pozycji (gdy wszystkie silniki są ustawione pod kątem 0 stopni: wyrównane z nacięciami) klikając w ten blok:






Ustaw silnik m1 pod kątem 90 stopni w dwie sekundy:



Poszukaj bloków przeciwnych i wykonaj skrypt:



Bloki mają różne kształty. Każdy z nich odpowiada konkretnej kategorii:

- Bloki owalne (jak ) nazywane są reporterami: wykonane przywracają wartość.
- Na górze skryptu może znajdować się “Kapelusz”, który wyznacza, kiedy skrypt ma być wykonany. Ich nazwy zwykle zaczynają się od słowa “When” (np. ); skrypt niekoniecznie musi mieć taki początek, ale bez takiego bloku, skrypt zostanie wykonany tylko wtedy, kiedy użytkownik go kliknie.
- Bloki komend (np. ) odpowiadają działaniom.

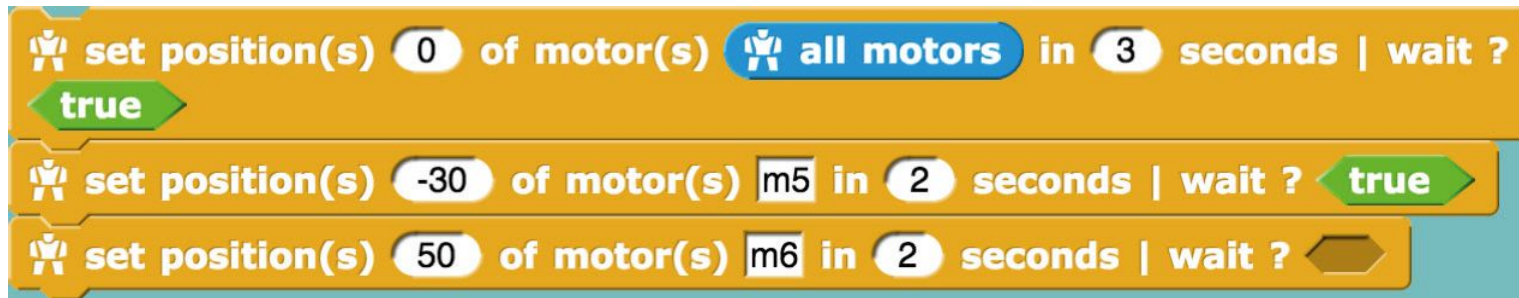
Zmień ustaloną pozycję bloku, aby ustawić motor m1 i motor m6 w pozycji -30 stopni w dwie sekundy.

Z pomocą bloków, które właśnie poznaliśmy, utwórz dwa programy na podstawie poniższej instrukcji:

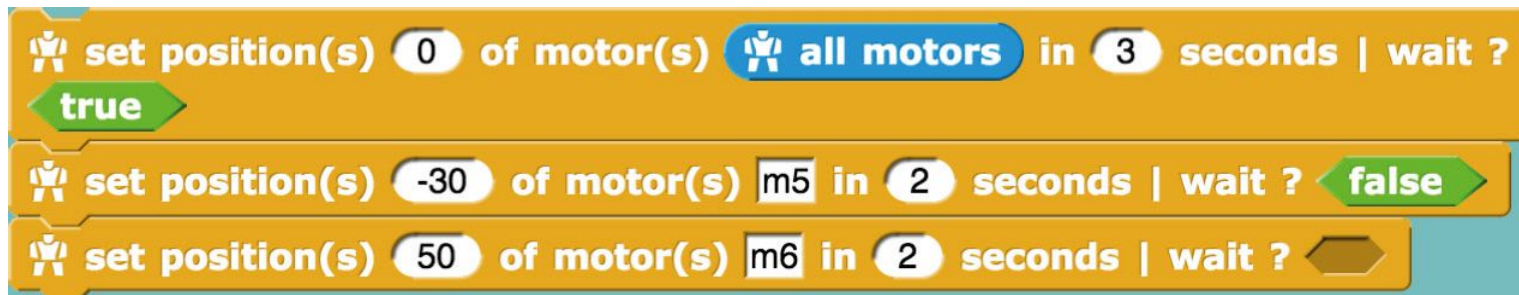
- Naciśnij ⇨ na klawiaturze, potem ustaw wszystkie silniki w pozycji 0 stopni w 3 sekundy.
- Naciśnij ⇐ na klawiaturze, potem ustaw silnik m1 i m4 w pozycji 60 stopni w 2 sekundy.

Teraz użyjemy silników do stworzenia ruchów.

Wykonaj poniższy skrypt i zobacz, co się stanie:




Zastąp blok  blokiem 



Kiedy mamy dwa lub więcej złożonych bloków, w jakiej kolejności wykonają czynności? Co się stanie, gdy ustawimy wartość "true" dla "Wait"? Co się stanie, gdy ustawimy wartość "false"?

Linijki kodu wykonywane są niemal jednocześnie; czasami nawet pozycja z poprzedzającej linii nie została osiągnięta. Komenda "Wait" pozwoli "zaczekać" do momentu kiedy poprzedzający silnik uzyska zadaną pozycję.

Za pomocą bloków, które znasz, wyznacz ruch Ergo Jr, który oznacza "hello", gdy zostanie wciśnięty klawisz b.

- Zaczynij od prostego ruchu a następnie przekształć go w bardziej złożony.
- Wybierz silniki, których chcesz użyć, by wykonać ruch.
- Spraw, aby robot wykonał ruch i obserwuj zachowanie silników.
- Możesz pomóc sobie blokiem  aby poznać pozycję żądanego silnika, zapisać pozycję i wykorzystać go ponownie.
- Zaprogramuj ruch silnik po silniku i za każdym razem przeprowadź test programu.

Nie bój się tworzyć innych ruchów!

Rezultaty uczenia się

Uczniowie będą potrafili :

- Pracować niezależnie, ufając sobie nawzajem
- Postępować zgodnie z pisaną instrukcją i tutorialem wideo
- Złożyć małe ramię robotyczne z otrzymanych elementów
- Użyć programowania blokowego w celu poruszenia robotem
- Współpracować z rówieśnikami przy zadaniach warsztatowych

Źródła:

Scenariusz oparty jest o materiały dostępne tutaj: <https://www.poppy-project.org/en/robots/poppy-ergo-jr>

- Apprendre à programmer Ergo Jr en Snap! Livret d'accompagnement du robot Poppy Ergo Jr (Conception et réalisation : Équipe Flowers Inria et Ensta ParisTech et Poppy Project; Design graphique : Antonin+Margaux)
- Dzień budowania i programowania robota! (<https://www.poppy-project.org/en/posts/poppy-ergo-jr-workshop-at-cern>)