

# Podróżnicy w czasie

## HISTORIA

Nowa historyczna ścieżka przeznaczona jest dla klas V. Oparliśmy się w niej na zagadnieniach związanych z historią Polski – od pierwszych Słowian do Wielkiej Wojny z Zakonem Krzyżackim i jej punktu kulminacyjnego – Bitwy pod Grunwaldem. Ścieżka ubrana jest w fabułę podróży wehikułem czasu wraz z Tymkiem, chłopcem z przyszłości.

Wyzwaniem dla uczniów będzie, podczas poznawania kolejnych etapów rozwoju Państwa Piastów i pierwszych Jagiellonów, „odwiedzać” związane z danym wydarzeniem lokalacje, aby na koniec podróży znaleźć się pod Grunwaldem i uratować prawdę historyczną.

Poza poznaniem najważniejszych faktów z historii kraju i elementów życia codziennego ludzi z danej epoki, uczestnicy projektu poznają podstawy programowania w Scratchu, obsługę m-botów oraz stworzą gry historyczne.

## Autorzy:

Alina Idzikowska, Radosław Potrac, Joanna Płatkowska

## Spis lekcji:

**Lekcja 1:** Spotkanie z Lelum Polelum

**Lekcja 2:** Rybka u Piastuskiewiczów

**Lekcja 3:** Za ciasna korona Bolesława Chrobrego

**Lekcja 4:** Kraj szyty grubymi nićmi – krzyżackie nożyce

**Lekcja 5:** Murujemy nowy dom z Kazimierzem Wielkim

**Lekcja 6:** Jak przeżyć w średniowieczu i nie zwariować?

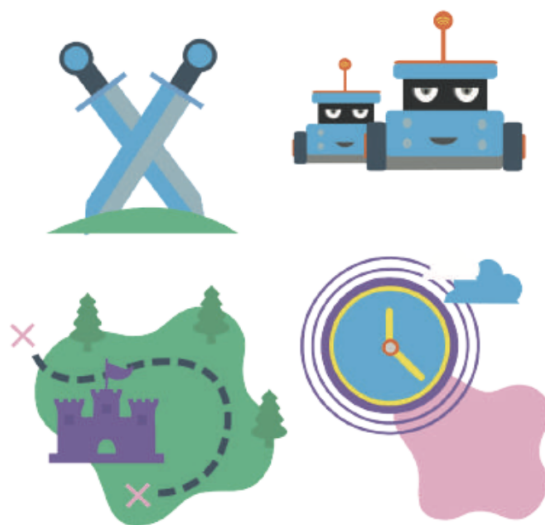
**Lekcja 7:** Polak, Węgier i... Litwin – trzy bratanki

**Lekcja 8:** Ratujemy z Tymkiem victorię grunwaldzką

**Lekcja dodatkowa:** Poznajemy mBota

## Licencja:

Creative Commons Uznanie autorstwa Na tych samych warunkach 3.0



# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 1 :

# Spotkanie z Lelum Polelum

Lekcja, podczas której nauczyciel wprowadza dzieci w konwencję zabawy - lekcja służy wprowadzeniu fabuły, oraz zapoznaniu z życiem Słowian na terenie Polski. Z zakresu programowania uczniowie będą mieli okazję wykonać prostą animację w Scratchu, w której trzeba przejść przez 4 różne mapy - mapę świata, mapę Europy, mapę terenów zamieszkałych przez Słowian, mapę Polski przed 992 rokiem. Będą też mogli wykonać trudniejszą wersję gry - quiz oparty na materiale lekcji, w którym nagrodą za dobrą odpowiedź będzie kolejna mapa.

### Cele zajęć:

Uczeń powinien:

- przedstawiać państwa, które powstały na terenach, na których osiedlili się Słowianie,
- omawiać życie codzienne i organizację państw słowiańskich,
- wiedzieć, kim byli Cyryl i Metody,
- potrafi wymienić nazwy najbardziej znanych ośrodków słowiańskich na terenie Polski.

### NaCoBeZu

Po lekcji:

- przedstawiasz państwa, które powstały na terenach, na których osiedlili się Słowianie,
- omawiasz życie codzienne i organizację państw słowiańskich,
- wiesz, kim byli Cyryl i Metody, potrafisz wymienić nazwy najbardziej znanych ośrodków słowiańskich na terenie Polski.

### Materiały pomocnicze:

- zakodowany list
- podręcznik do historii klasa 5
- materiały dotyczące kodowania, dostępne na końcu scenariusza
- pdf z kartami-nagrodami (wizerunki bogów słowiańskich)
- 1 robot mBot
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ Słowianie → opole → plemię → wiec → drużyna → gród → starszyzna → chrzest → księżę → Cyryl i Metody

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka
- praca z mapą
- algorytmika w programie Scratch,
- gra edukacyjna - grywalizacja.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach, tworzenie animacji/gry.

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia

III Średniowieczna Europa

Uczeń:

2) umiejscawia w czasie i przestrzeni nowe państwa w Europie

Podstawa programowa kształcenia ogólnego dla szkół

podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

1. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:  
2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

3) sterowanie robotem lub obiektem na ekranie;

3. w algorytmicznym rozwiązywaniu problemu wyróżnia

podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

1. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1) projektuje, tworzy i zapisuje w wizualnym języku programowania:  
1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,

2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Lekcja rozpoczyna się wejściem nauczyciela, który opowiada, że zdarzyło mu się coś dziwnego i spotkał po drodze do klasy dziwne stworzenie z listem do jego uczniów. Siada i wyciąga telefon, korzystając z aplikacji wprowadza mBota do sali. Ma on zakodowany list przyczepiony do korpusu.

List ten wyjaśnia, że Tymoteusz - uczeń z XXI wieku - zagubił się w czasie i przestrzeni i potrzebuje pomocy młodzieży. Wszystkim, którzy chcą mu pomóc wrócić do swych czasów, oferuje wstąpienie do Klubu Podróżników w czasie.

Z listu wynika, że jego wehikuł został uszkodzony przez wrogiego agenta i zgubił się w IX wieku w okolicy Poznania i Gniezna.

#### LIST

GA/DE/RYP/PO/LU/KI

Drodzy Koledzy.

Pochodzę z roku 2050. Chciałem zapobiec wielkiej tragedii w przeszłości, ale zostałem zaatakowany przez wrogiego agenta. Mój pojazd zagubił się w czasie i przestrzeni w Gnieźnie u Słowian. Potrzebuję pomocy. Każdemu, kto udzieli mi pomocy, oferuję uczestnictwo w elitarnym Klubie Podróżników w Czasie. Szczegóły następnym razem.

Tymoteusz

Eypezr lpudezr.

Opchpezę z ypil 2050. Chckgłdm zgopbkdc wkdkidj tygadekk w oyzdślpsck, gud zpstgłdm zeygezpnr oyzdz wypakdap gadntg. Mój opjgze skę zgalbkł w czgskd k oyzdstyzdnk w Ankdźnk I Słpwkgn.

Optyzdblję opmpcr. Igzedml, itp lezkduk mk opmpcr pfdylję lczdstnkctwp w duktgynrm lulbkd

Opeyóżnkiów w Czgskd.

Szcdaólr ngstęonrm ygzdm.

TRMPTDLSZ

Po rozkodowaniu listu nauczyciel prowadzi dyskusję, w trakcie której ustala z uczniami przyłączenie się do Tymoteusza i pomoc w rozwiązaniu zagadki (może zahaczyć opowieścią o podróżach w czasie, nawiązać do powieści H.G. Wellsa "Wehikuł Czasu"). Sugeruje, że każde dobrze wykonane na lekcji zadanie pomoże Tymoteuszowi.

Po dyskusji pyta, co z robotem? Sugeruje, że warto się z nim zapoznać, poznać jego język. Pyta, czy dzieci się nim opiekują? (rozmowa o tym, jak opiekować się "normalnym" zwierzęciem - a robotem?) Robot zatem zostaje na razie w klasie, a nauczyciel wprowadza w temat Słowian, nawiązując do listu. Wprowadza cezurę czasową, prowadzi prezentację z pogadanką.

## 2. Część zasadnicza

czas: ok. 15 minut.

Nauczyciel wyjaśnia temat lekcji i przedstawia życie codzienne Słowian, opola, plemiona, wiece, księżęta, drużyny, rodzaj prowadzonej gospodarki, wierzenia. Opowiada o Cyrylu i Metodym, którzy nieśli nową wiarę na ziemię Słowian.

## 3. Podsumowanie i ewaluacja

czas: ok 5 minut

Nauczyciel odwołuje się do NaCoBeZu i podsumowuje lekcję, może też wykorzystać jakiś ulubiony sposób oceny lekcji przez uczniów.

### Programowanie:

Nauczyciel pokazuje uczniom animację, w której zmieniają się mapy - od mapy świata, poprzez Europę, Europę Środkową (opaną przez Słowian w VII wieku) aż do mapy plemion polskich. Sensem animacji jest znalezienie terenów zamieszkałych przez Słowian oraz pokazanie polskich Słowian - ich terenów i plemion. Nauczyciel prosi uczniów o wykonanie podobnej animacji w parach. Opis zadania i materiały pomocnicze znajdują się w materiałach dodatkowych dotyczących kodowania.

### Animacja ze zmianą map:

Opis:

Pierwsze zadanie polega na stworzeniu animacji do zmieniania map. Uczniowie dostają cztery mapy w postaci plików JPG. Mają za zadanie osadzić je jako cztery tła do programu w Scratchu oraz stworzyć skrypt powodujący kolejną zmianę map po użyciu wybranych klawiszy na komputerze. Powinni też narysować stronę tytułową animacji i spowodować, żeby skrypt po kliknięciu zielonej flagi zaczynał się od tytułu.

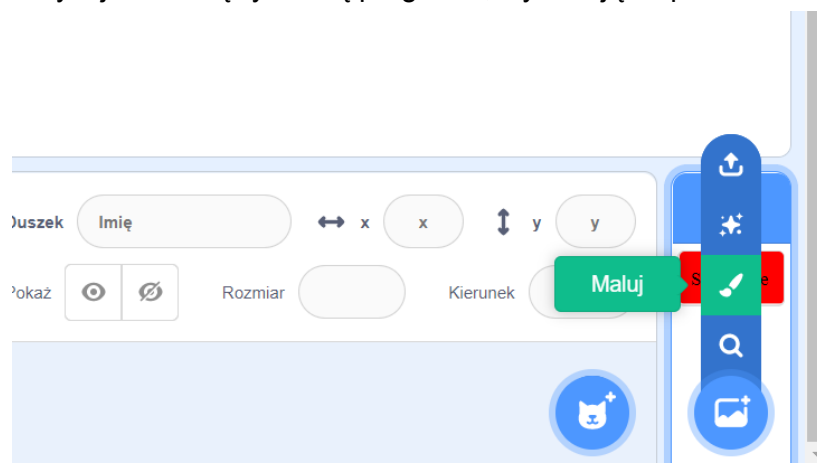
Prosta animacja ze zmianą map <https://scratch.mit.edu/projects/531874982/>.

Otwórzcie ten plik i pobawcie się mapami, naciskając wymagane w opisie klawisze.

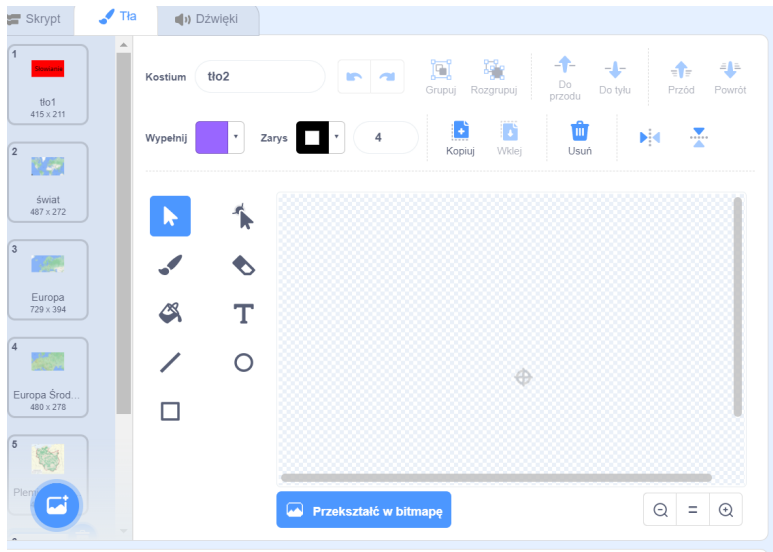
Krok 1:

### Jak zrobić taką animację?

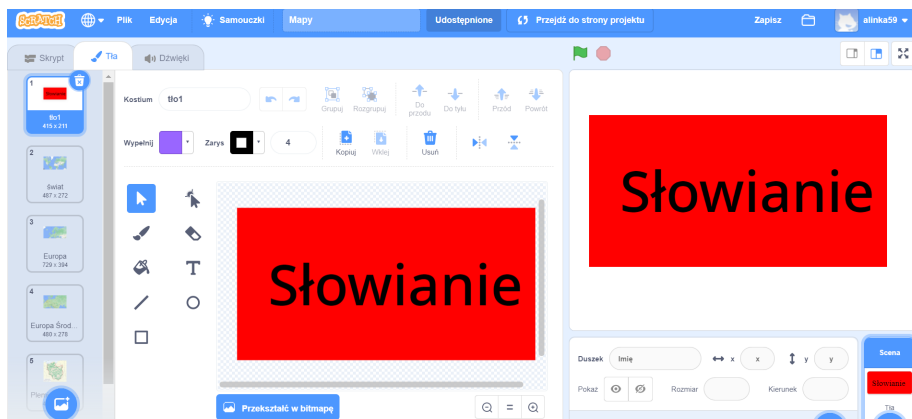
Narysujcie stronę tytułową programu, wybierając z panelu Scena przycisk Maluj:



i korzystając z edytora do tworzenia grafiki:



Może to wyglądać na przykład tak:



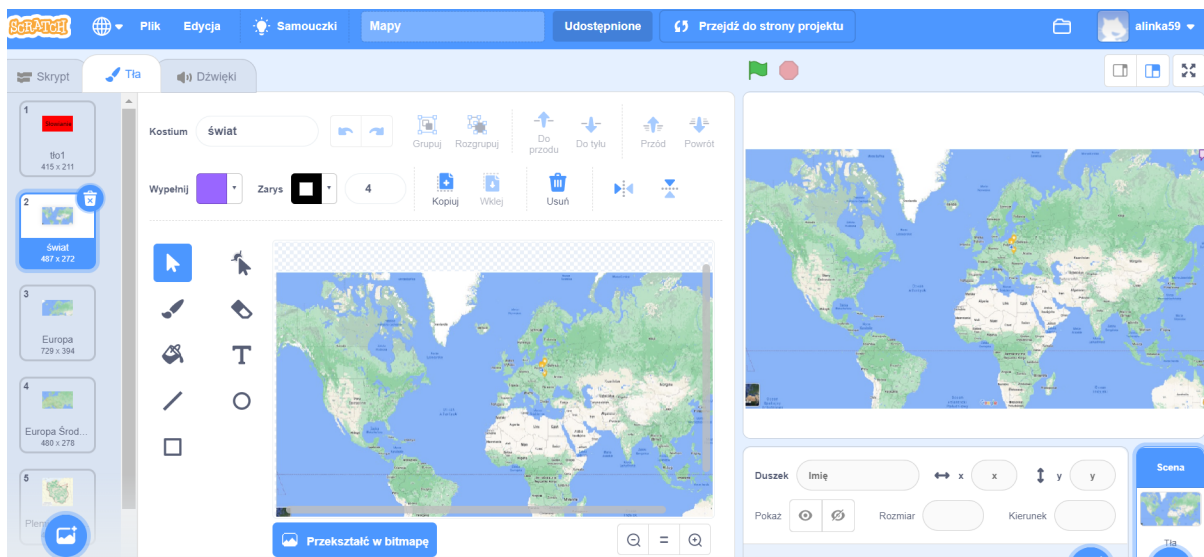
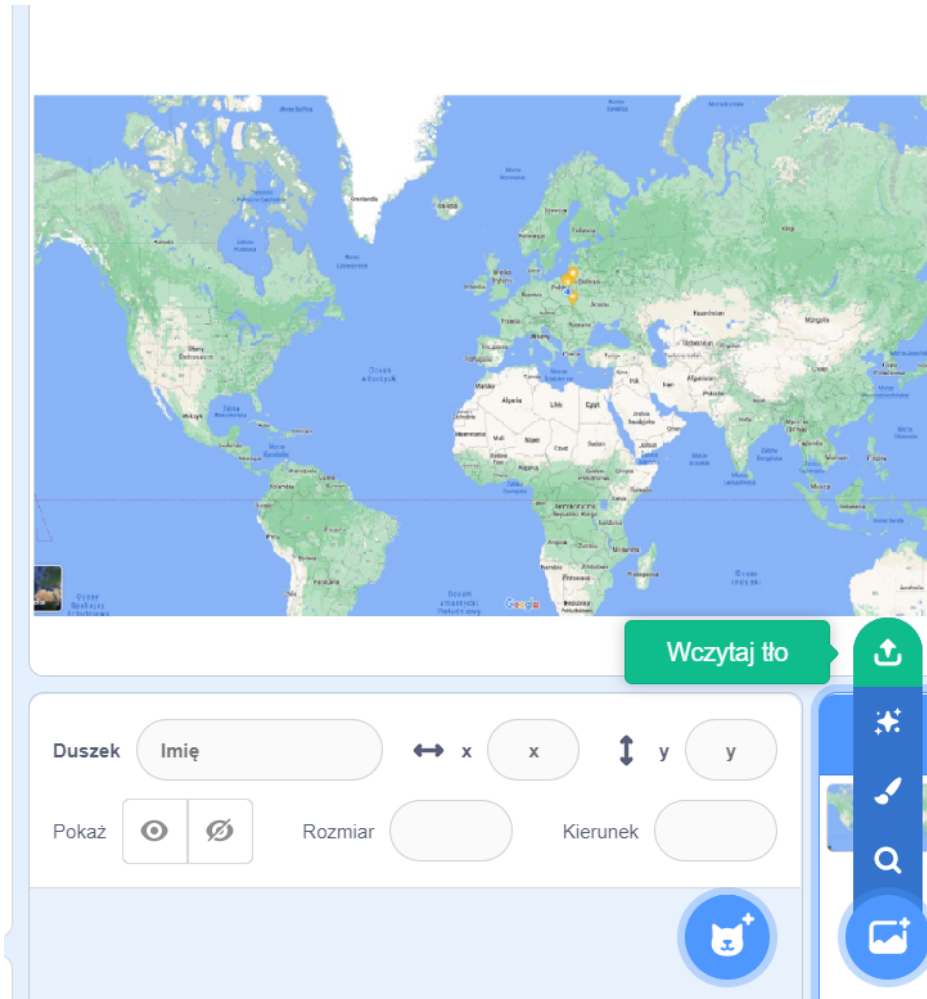
Krok 2:

Następnie wczytajcie z pliku obrazy 4 map (świata, Europy, Europy Środkowej i mapę plemion słowiańskich)

Mapy:

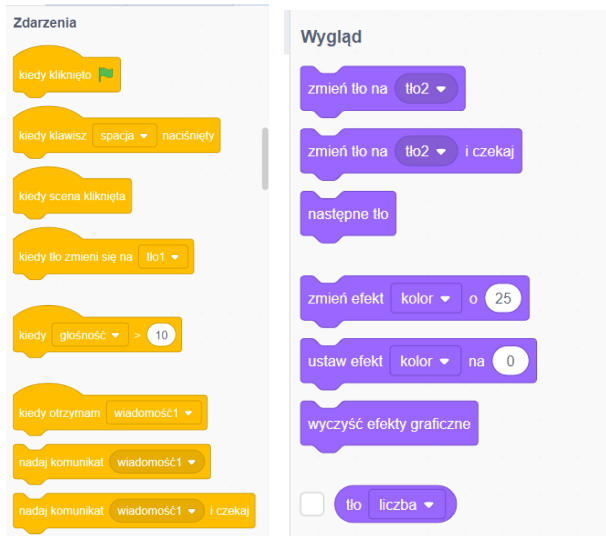
<https://epodreczniki.pl/a/mieszko-i---poczatki-panstwa-polskiego/DsBuZoaw7>

Pozostałe mapy są wycinkami z map Google.



Krok 3:

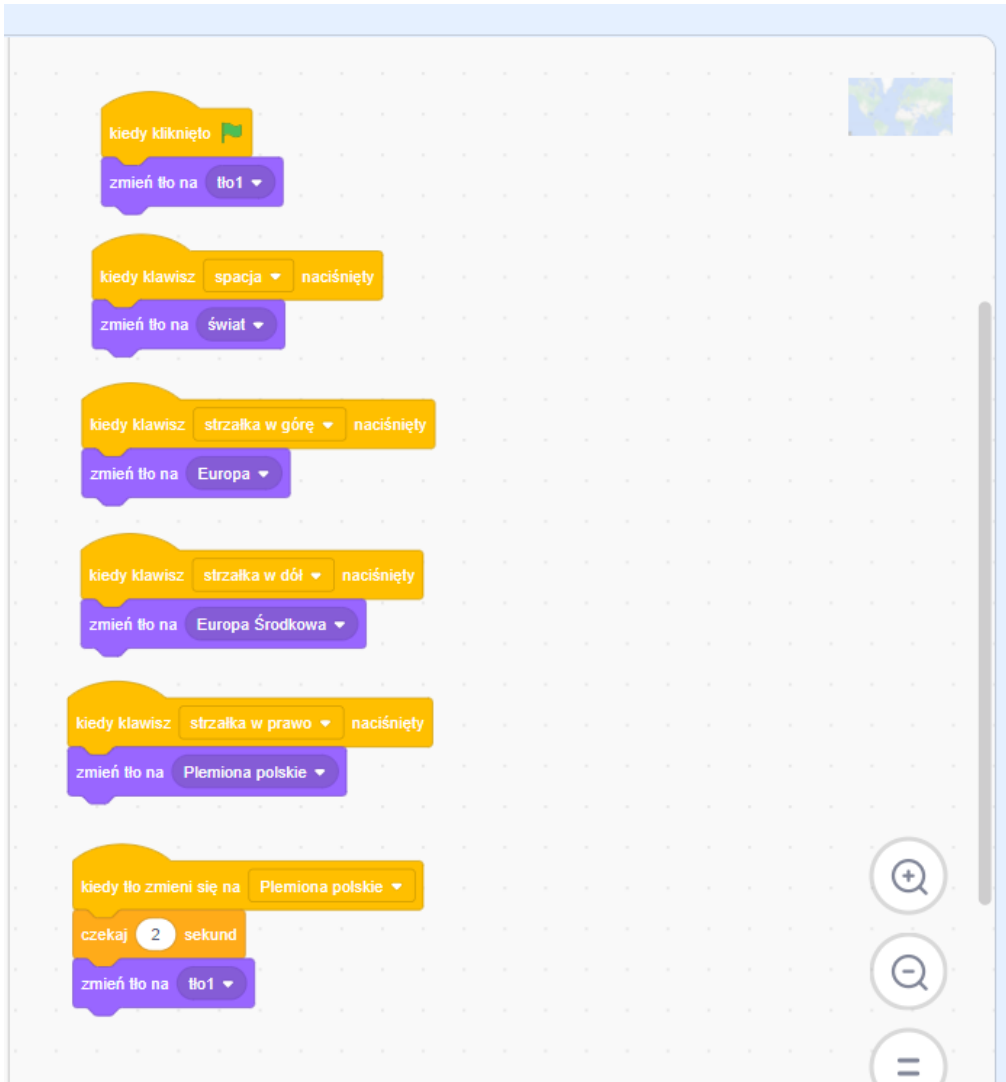
Zaprogramujcie skrypt zmiany map, korzystając z działu bloczków Zdarzenia i Wygląd:



Chodzi nam o to, żeby po naciśnięciu konkretnego klawisza tło zmieniało się na inne; zaś zielona flaga powinna zmienić tło na okładkę. Na koniec działania programu tło znów powinno się zmienić na okładkę, ale nie od razu, np. tak:



Całość może wyglądać tak:



## Uwagi/alternatywy:

**Minimum:** lekcja historii o Słowianach; początek historii o Tymku, który podróżuje w czasie; animacja w Scratchu ze zmianą map.

### Dodatki do wyboru lub wszystkie:

- karty jako nagroda dla uczniów (bogowie Słowian);
- artefakt będący częścią generatora czasu Tymka to ogień w dłoni Swaroga;
- zaprogramowanie gry-quizu w Scratchu;
- narysowanie na dużym papierze kształtu Polski za czasów Mieszka I:

<https://zpe.gov.pl/a/mieszko-i---poczatki-panstwa-polskiego/DsBuZoaw7>

Ta duża mapa będzie wykorzystywana na każdej lekcji - uczniowie będą na niej rysować poszczególne miejsca, gdzie dociera Tymek. Połączenie ich czarną trasą o odpowiedniej szerokości pozwoli jeździć po niej mBotowi. Jeżeli zaznaczymy tylko miejsca - możemy zaprogramować mBota, żeby dojechał do następnego punktu podróży Tymka. Pierwszy punkt - **Gniezno**.

W nagrodę za wykonanie zadań uczniowie otrzymują karty z bogami słowiańskimi. Jedną z kart jest niezbędna do dalszej podróży Tymka - karta Swaroga. Elementem czasowym, który wypadł z generatora czasu w wehikule Tymka jest **ogień w dłoni Swaroga**.



Oczywiście animacja jest zadaniem dla uczniów, którzy nigdy nie mieli do czynienia ze Scratchem. Aby zainteresować także uczniów umiejących programować, proponujemy **grę- quiz z pytaniami**. Tłami mogą być te same mapy i początek może być taki sam, jak w animacji. Tu jednak musimy wybrać duszka, który będzie zadawał pytania. Nagrodą za dobrą odpowiedź jest zmiana tła na kolejną mapę. Uczniowie mogą oczywiście samodzielnie wybrać duszka i wymyślić własne pytania. Główne zadania dla uczniów to:

- zakodowanie duszka tak, aby zmieniał kostiumy,
- pokazał dymek z powitaniem i początkiem gry i zadawał pytania, a potem je oceniał oraz aby po dobrej odpowiedzi następowała zmiana mapy i kolejne pytanie.

Warto zwrócić uwagę uczniów na ewentualność złej odpowiedzi i odpowiednie zapisanie skryptu, żeby gra mogła toczyć się dalej po wpisaniu poprawnej odpowiedzi.

Najpierw pograjcie w tę grę. Udzielajcie odpowiedzi w oparciu o podręcznik, bo program zaliczy Wam tylko jedną poprawną odpowiedź. (Nauczyciel powinien zajrzeć do skryptu, bo musi zostać udzielona dokładnie taka odpowiedź, jaka jest zapisana w programie; wielkość liter nie ma znaczenia).

Gra Słowianie na mapie <https://scratch.mit.edu/projects/534590081/>

Krok 1:

### Zajrzyj do skryptów programu:

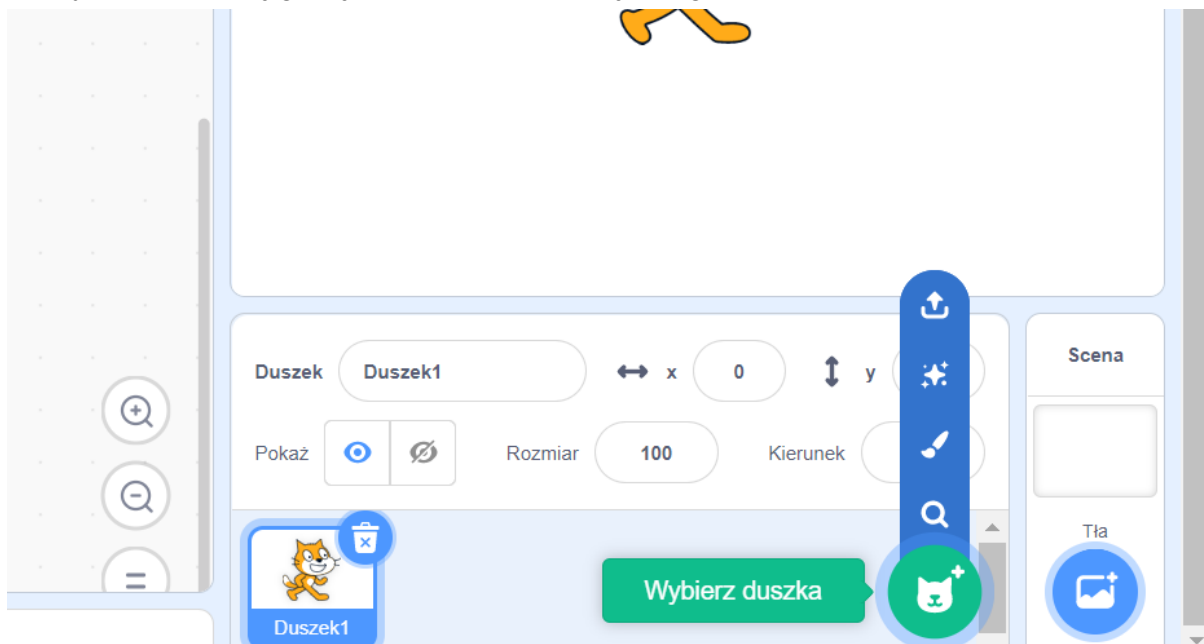
Jakie działania wykonuje duszek?

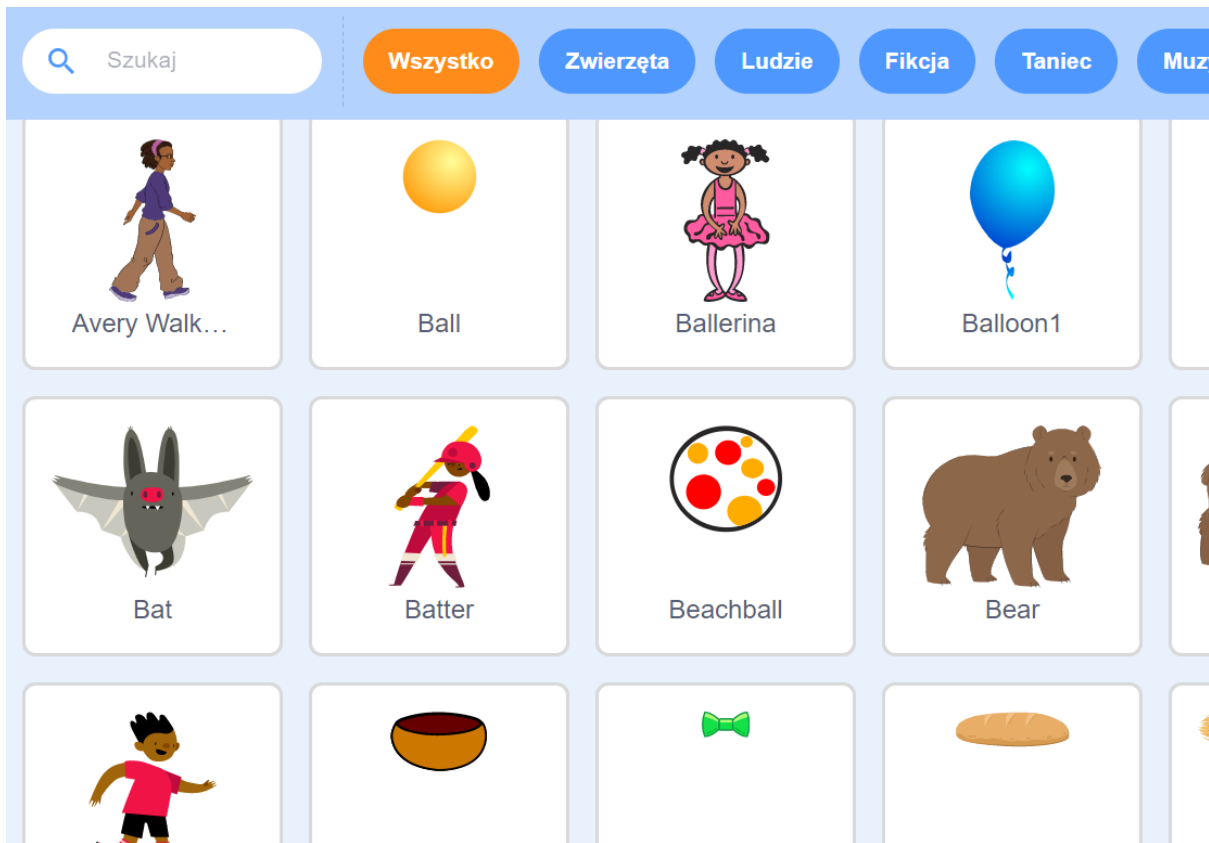
1. Rusza się.
2. Pokazuje dymki z wypowiedziami.
3. Zadaje pytania i pokazuje ramkę na odpowiedź.
4. Po poprawnej odpowiedzi pokazuje dymek z pochwałą i zmienia tło na następną mapę.
5. Po błędnej odpowiedzi proponuje, aby spróbować jeszcze raz i znów zadaje pytanie oraz pokazuje ramkę na odpowiedź.

Krok 2:

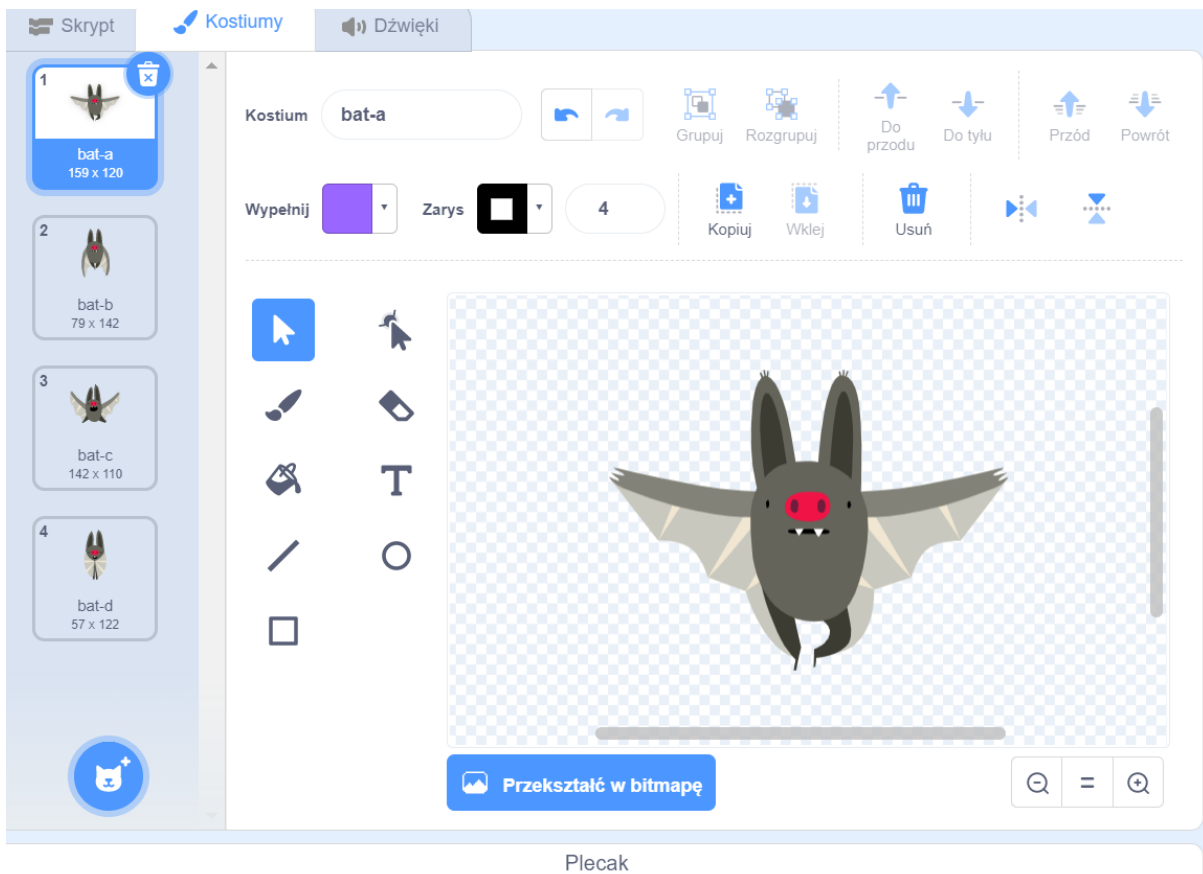
### Zaprogramuj:

Ruchy duszka. W tej grze jest to nietoperz. Znajdź tego duszka w bibliotece duszków.



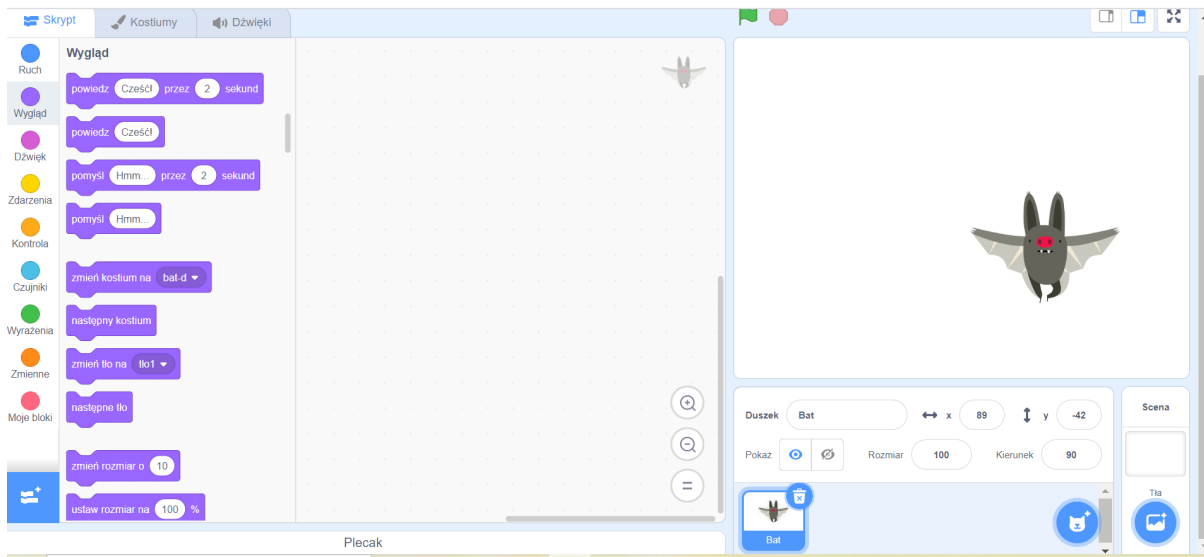


Wybierz duszka kliknięciem i wróć do swojego programu. Usuń niepotrzebnego duszka kota. Zajrzyj do zakładki Kostiumy na górnej belce:

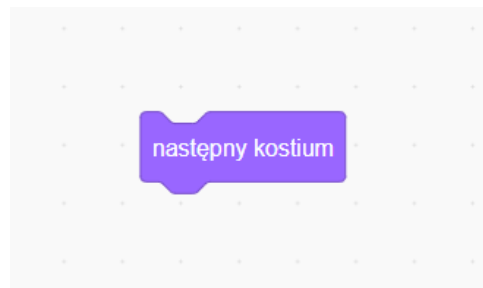
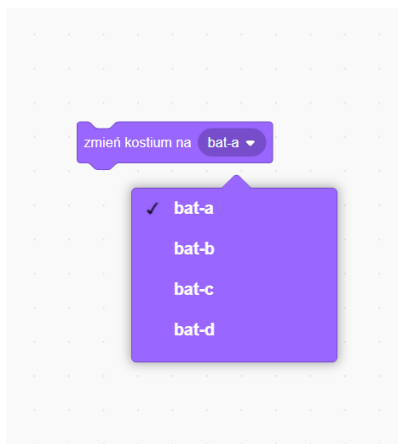


## #SuperKoderzy / Podróżnicy w czasie / Spotkanie z Lelum Polelum

Ten duszek ma 4 kostiumy. Kiedy będziesz klikać po kolei w te kostiumy - nietoperz będzie się ruszał. Teraz trzeba napisać skrypt tak, żeby nietoperz zmieniał kostiumy automatycznie. Przejdź do zakładki Skrypt na górnej belce:

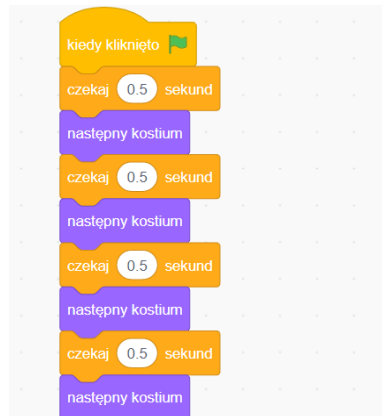
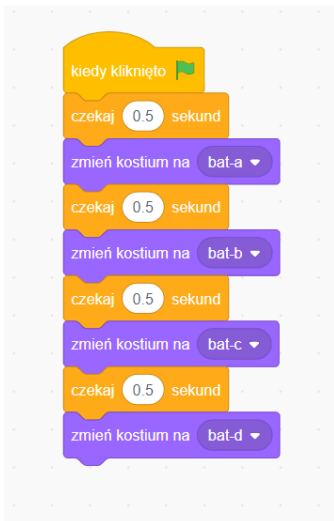


Z lewej strony, w menu bloków programu (są oznaczone kolorowymi kółkami i nazwami) znajdź blocki Wygląd. Znajdziesz tam blok "zmień kostium na". Blok ma menu rozwijalne w strzałce. Jest też blok "następny kostium".



Jak uruchomić nietoperza przy pomocy blocka z zieloną flagą i blocków "zmień kostium" lub blocka "następny kostium"?

Pomiędzy zmianami kostiumu musisz wstawić chwilę czekania z blocków kontroli. Może to wyglądać tak, jak na obrazku z prawej strony lub tak, jak na obrazku z lewej strony:



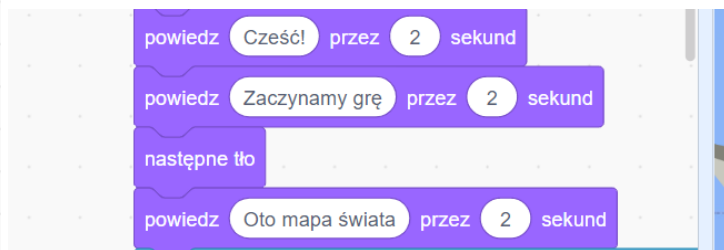
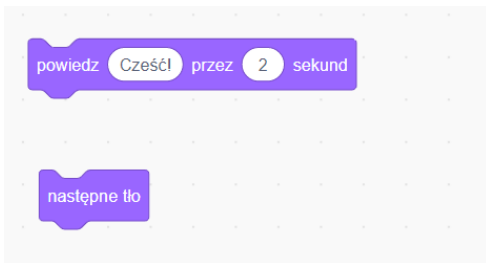
Aby nie trwało to zbyt długo - niech czas pomiędzy kostiumami będzie 0,5 sekundy (pamiętaj, że wpisując czas w okienko obok "czekaj" pomiędzy zerem a piątką, musisz wstawić kropkę, nie przecinek).

Po zielonej fladze też musi być pauza - żeby mogła się pokazać scena z okładką. Programujemy na początek jedną sekwencję ruchu (4 kostiumy), ponieważ dalej nietoperz będzie się ruszał wolniej, w trakcie zadawania pytań.

Krok 3:

**Zaprogramuj przemowę początkową duszka:**

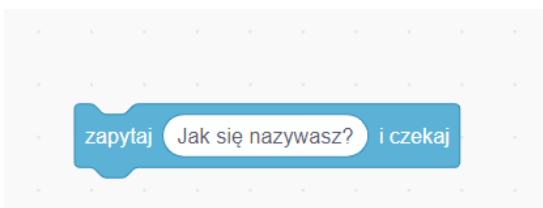
Korzystamy nadal z bloczków Wygląd. Niech duszek mówi po kolei - wykorzystaj trzy razy klocek "powiedz". Będziemy też chcieli zmienić scenę na mapę świata.



Krok 4:

**Zaprogramuj pierwsze pytanie quizu:**

Aby rozpocząć quiz, musimy użyć bloczków Czujniki oraz Wyrażenia. Aby duszek zadał pytanie, na które można wpisać odpowiedź, trzeba użyć bloczka:

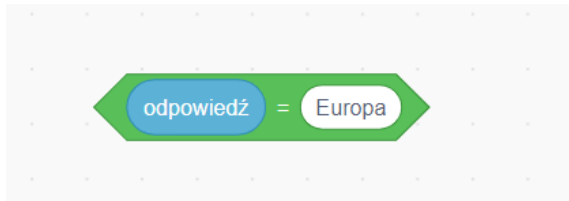


Oczywiście pytanie można wpisać własne.

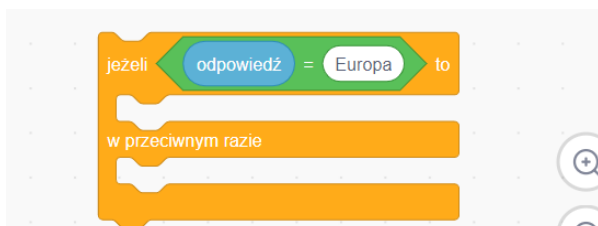
Teraz potrzebujemy bloczka z równaniem z Wyrażeń i bloczka odpowiedzi z Czujników:



Trzeba je połączyć, a w wolne pole wpisać poprawną odpowiedź:

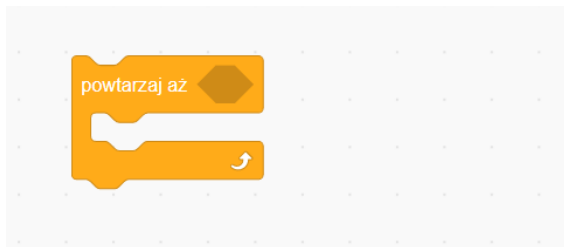


Dzięki temu duszek może rozpoznać poprawną odpowiedź. Ale - jak widać, nie da się takich klocków połączyć z innymi. Trzeba je wstawić w bloczek z działu Kontrola:



Używamy takiego warunku (jeżeli - to; w przeciwnym razie), ponieważ odpowiedź może być dobra, albo zła. Jak zareaguje duszek na poprawną odpowiedź? Potrzebne bloczki znajdziemy w dziale Wygląd. Byłoby dobrze, żeby pochwalił gracza, zmienił kostium i zmienił mapę na mapę Europy. Jeżeli odpowiedź będzie zła, musi powiedzieć o tym graczowi, zadać ponownie pytanie, ale tu też musi być bloczek ze zmianą kostiumu i tła, bo gracz tym razem może dobrze odpowiedzieć.

Jest jeszcze jeden problem. Gracz może udzielić złej odpowiedzi kilka razy. Dlatego w dolną część warunku (w przeciwnym razie) trzeba wstawić dodatkową pętlę:



a w puste miejsce oczywiście trzeba wstawić równanie: odpowiedź=Europa.

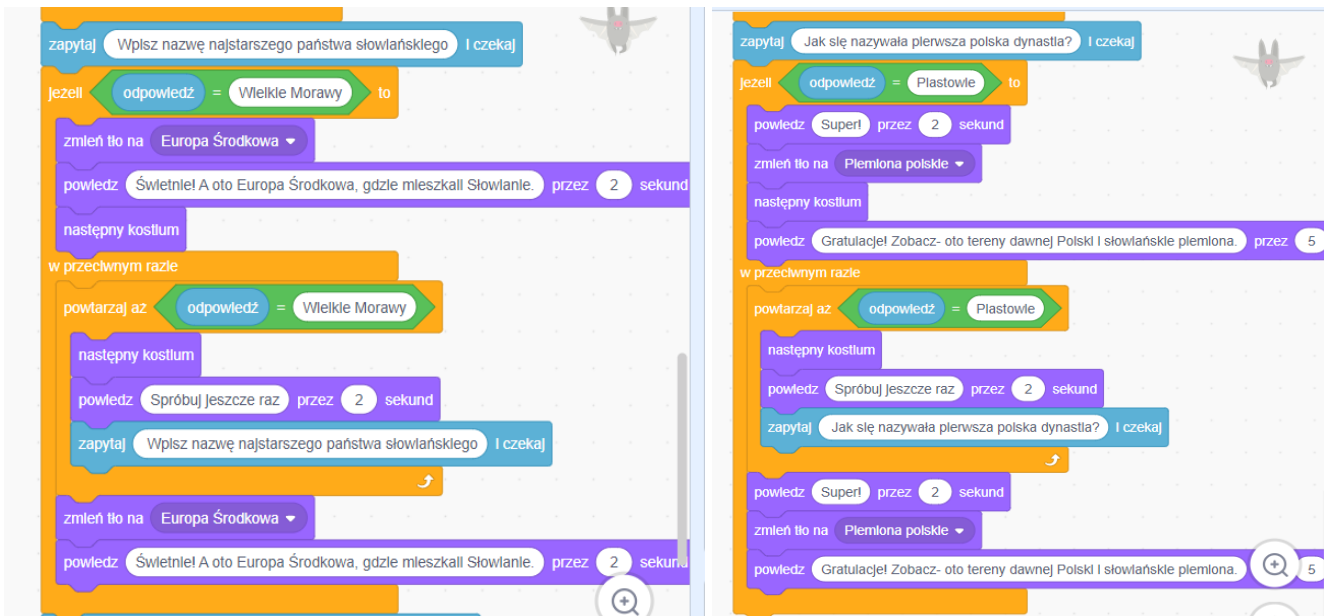
Cały kod pierwszego pytania może wyglądać tak:



Zwróć uwagę, co trzeba wstawić w które miejsce warunku (jeżeli-to; w przeciwnym razie) i gdzie kończy się zakres pętli "powtarzaj aż".

Krok 5:

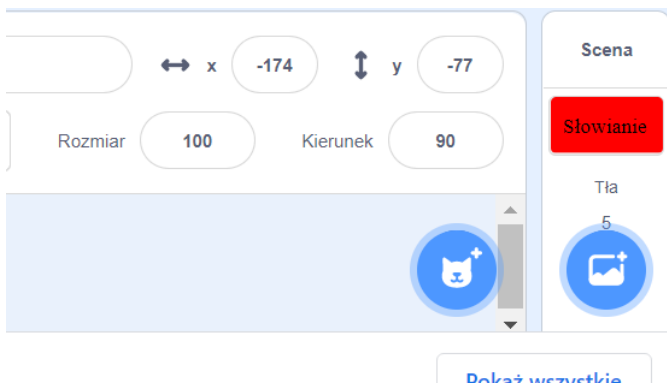
**Zaprogramuj dalsze pytania w taki sam sposób.**



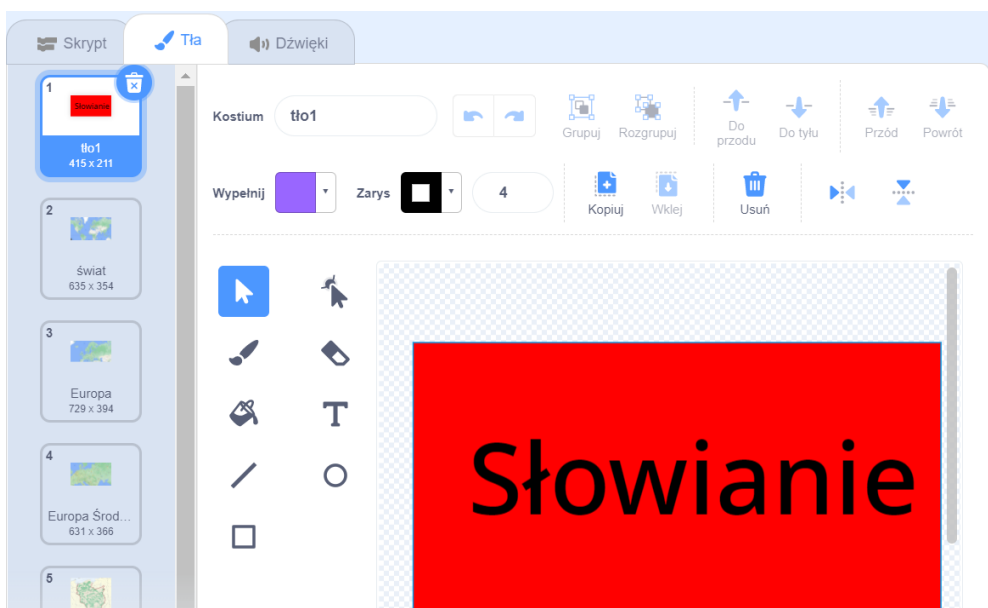
Krok 6:

**Zaprogramuj zmiany sceny czyli okładkę i mapy:**

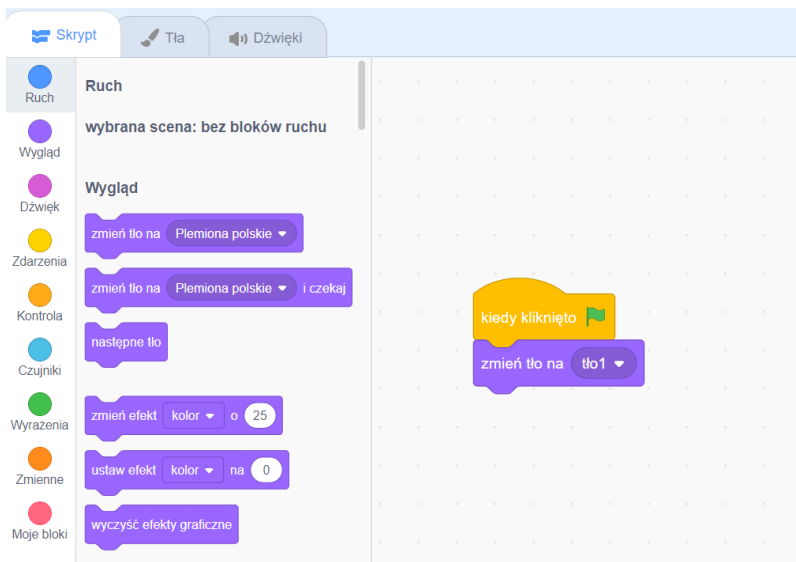
Wejść w panel Scena z prawej strony na dole:



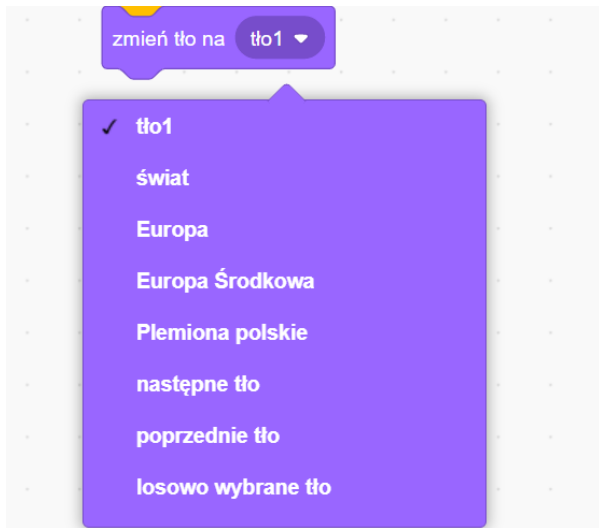
Otworzy się edytor sceny i tła. Na górnej belce trzeba wybrać zakładkę Tła:



Powinny tam być takie same tła, jak w animacji z mapami, czyli okładka z tytułem, mapa świata, mapa Europy, mapa Europy Środkowej (wycinki z map Google), mapa plemion słowiańskich <https://epodreczniki.pl/a/mieszko-i---poczatki-panstwa-polskiego/DsBuZoaw7>. Musimy zaprogramować tylko pojawienie się okładki po kliknięciu zielonej flagi. W tym celu trzeba wejść w zakładkę Skrypt, pozostając w panelu sceny. Skrypt jest taki:



Pozostałe zmiany tła są zaprogramowane w duszku. Pamiętaj, że możliwość zmiany tła kryje się w rozwijalnej liście bloczka “zmień tło na”:





# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 2 :

# Rybka u Piastuskiewiczów

Lekcja, podczas której nauczyciel opowie dalszą część przygód Tymka i wprowadzi uczniów w tajniki powstania państwa polskiego oraz opowie o chrzcie Mieszka I. Uczniowie będą mogli stworzyć grę "Denary Mieszka I".

### Cele zajęć:

Uczeń powinien:

- wykorzystując wiedzę z lekcji o Słowianach, opowiedzieć, jak zorganizowane było plemię słowiańskie,
- wymieniać największe polskie plemiona,
- wskazywać przyczyny i skutki oraz datę przyjęcia chrztu przez Mieszka,
- wiedzieć, kim byli: Mieszko I i Dobrawa,
- opowiadać o powstawaniu państwa Mieszka I.

### NaCoBeZu

Po lekcji:

- opowiadasz, jak było zorganizowane plemię słowiańskie,
- wymieniasz największe polskie plemiona,
- wskazujesz przyczyny i skutki oraz datę przyjęcia chrztu przez Mieszka,
- wiesz, kim byli: Mieszko I i Dobrawa,
- opowiadasz o powstawaniu państwa Mieszka I.

### Materiały pomocnicze:

- nagranie wywiadu z Mieszkiem I (wykonane przez nauczyciela),
- podręcznik do historii klasa 5,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (skarby królów polskich),
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

plemię → drużyna → gród → książę → danina → Polanie i Wiślanie → Piastowie → chrzest → Ostrów Lednicki

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- pogadanka,
- algorytmika w programie Scratch,
- gra edukacyjna- grywalizacja.

### Formy pracy:

- praca z całą klasą
- grupowa - praca w grupach (tworzenie gry).

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia V. Polska w okresie wczesnopiastowskim.

Uczeń:

1) sytuuje w czasie i przestrzeni państwo pierwszych Piastów oraz przedstawia jego genezę; 2) wyjaśnia okoliczności przyjęcia chrztu przez Piastów oraz następstwa kulturowe, społeczne i polityczne chrystianizacji Polski

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka: I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

3) sterowanie robotem lub obiektem na ekranie;

3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:

1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,

2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Nauczyciel opowiada, że dzięki pracy uczniów i karcie z bogiem Swarogiem, Tymek mógł kontynuować swoją podróż i znalazł się w roku 966 na Ostrowie Lednickim. Tam poznał księcia Mieszka I i przeprowadził z nim wywiad, który ukradkiem nagrał smartfonem (ale tylko głos). Z tej rozmowy uczniowie dowiedzą się, jak powstawało państwo polskie i jakie miało problemy.

### 2. Część zasadnicza

czas: ok. 15 minut.

Uczniowie poznają zagadnienia związane z tematem lekcji, słuchając nagranych przez Tymka wywiadów. Mieszko opowiada o różnych plemionach polskich, o swoich wysiłkach, żeby je zjednoczyć lub podbić, przypomina też, jak jest zorganizowane plemię słowiańskie i mówi o swoim rodzie - Piastach. Potem opowiada o problemach z sąsiadami - Niemcami i swoim pragnieniu wzmocnienia władzy poprzez uczynienie z państwa Polan królestwa (a to mogło się stać tylko za zgodą papieża).

W tym miejscu nauczyciel wyłącza nagranie i zaczyna dyskusję z uczniami - co powinien zrobić Mieszko, żeby osiągnąć swój cel, co mu radzą? Mówi o wielożeństwie, władzy papieża i możliwościach działania Mieszka. Po decyzji uczniów opowiada, jak było naprawdę i jakie korzyści miała przyszła Polska z tytułu przyjęcia chrztu przez władcę oraz jakie problemy miał przez to władca. Opowiada też o ostatnich latach rządów Mieszka i walkach z Niemcami i Czechami.

### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

nauczyciel powraca do NaCoBeZu, aby dokonać ewaluacji lekcji.

Po wykonaniu zadania z programowania (poniżej), przekazuje dalsze informacje o Tymku i jego przeciwniku - Jonasie.

Tymoteusz informuje je, że jego wrogiem jest chłopak mówiący po niemiecku, który ma do wykonania w przyszłości jakieś zadanie - zmianę biegu jednej z największych bitew między Polakami i Niemcami. Sposobem, by nie pozwolić przeciwnikom na realizację planu, jest wykonywanie zadań w Scratchu i programowanie mBotów, ponieważ dzięki skupieniu wielu uczniów Tymek będzie mógł wyróżnić falowanie czasoprzestrzeni i dokonywać kolejnych skoków w czasie i przestrzeni. Niezbędne są mu też kolejne karty (tym razem konkretna karta z diademem, który Otto III nałożył na głowę Bolesława, syna Mieszka I - diadem jest kolejnym elementem generatora czasu w wehikule).

#### Programowanie:

Nauczyciel pokazuje uczniom grę "Denary Mieszka I" i poleca, żeby spróbowali ją wykonać w parach. Opis zadania i materiały pomocnicze znajdują się w materiałach dodatkowych dotyczących kodowania.

### Gra Denary Mieszka I:

Gra Denary Mieszka I: <https://scratch.mit.edu/projects/535336422/>

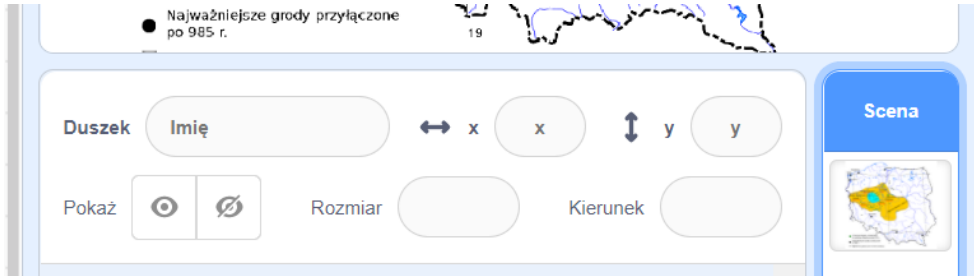
Krok 1:

Zajrzyj do skryptów programu:

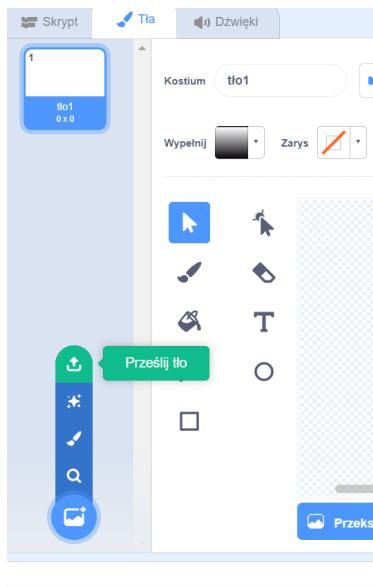
Zagraj w tę grę i zobacz, jak działa, zastanów się, z jakich składa się duszków, a co jest sceną. Potem zajrzyj do skryptów programu.

Krok 2:

**Zaprogramuj tło dla gry** - osadź mapę jako tło dla gry na scenie. Wejdź w scenę:



w zakładkę Tła na górnej belce i prześlij tło z komputera:



Link do mapy:

Autorstwa Pedros.lol - Z. Kurnatowska, M. Kara, CC BY-SA 4.0,

<https://commons.wikimedia.org/w/index.php?curid=15008968>

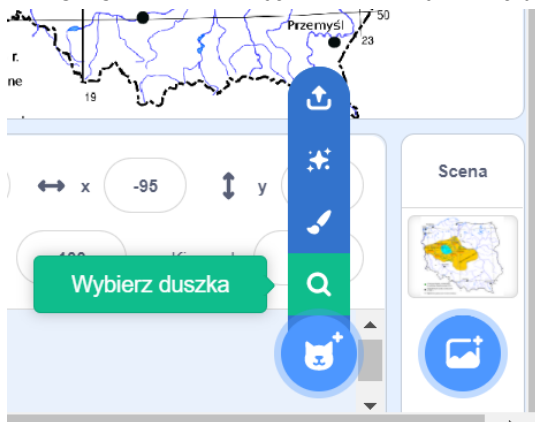
Przekształć mapę na wektor i dopasuj ją wielkością do sceny, ale tak, żeby kształt mapy cały był widoczny:



Postaraj się, aby środek sceny był w środku mapy.

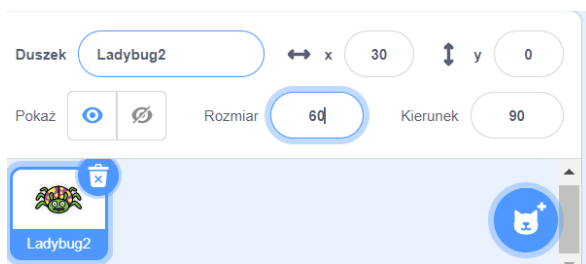
Krok 3:

**Wczytaj duszka.** Wyjdź ze sceny klikając w duszka kota i wczytaj nowego duszka z biblioteki:



W tej grze jest to pająk, ale możesz wybrać dowolnego duszka pasującego do dalszych skryptów. Potem wróć do programu. Usuń niepotrzebnego kota.

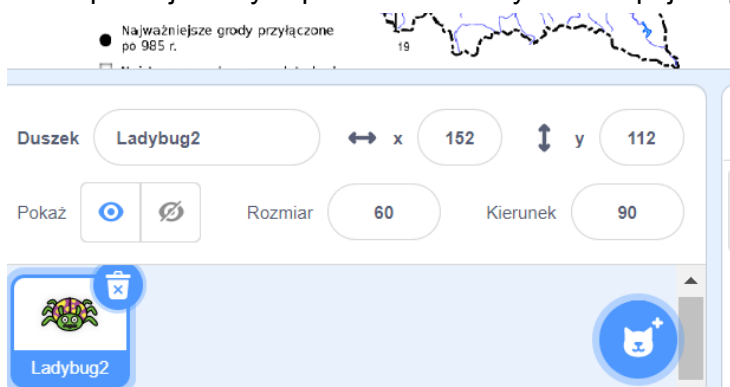
Na górnej belce z lewej strony musisz wybrać zakładkę Skrypt. Ponieważ pająk jest trochę za duży - możesz go zmniejszyć, szukając na dole z prawej strony - owalnej ramki Rozmiar. Normalny rozmiar to 100, lecz jeśli wpiszesz mniejszą liczbę, duszek się zmniejszy:



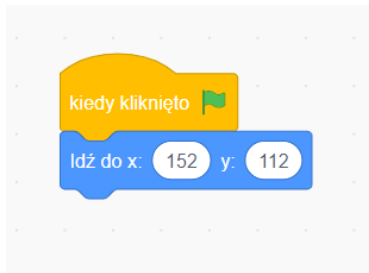
Krok 4:

**Zaprogramuj miejsce pojawienia się i ruchy duszka.**

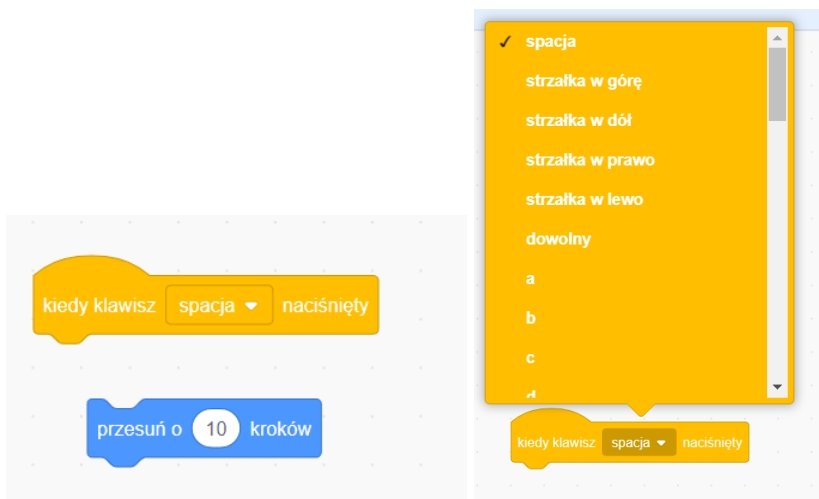
Zdecyduj, gdzie powinien zaczynać grę duszek i przesuń go w to miejsce myszką. Kiedy to zrobisz - na dole z prawej strony w parametrach x i y duszka pojawią się liczby określające jego położenie na scenie:



Aby duszek zawsze na początku gry pojawiał się w tym samym miejscu, trzeba z działu Zdarzenia wybrać bloczek z zieloną flagą, a z działu Ruch bloczek "idź do x... y...". Jeżeli położenie duszka jest nadal takie, jak chcesz - na bloczku pojawią się potrzebne liczby.

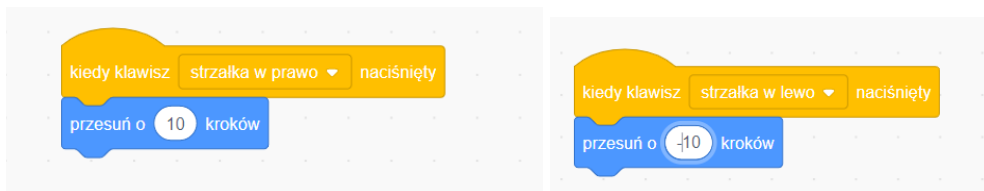


Chcemy, żeby nasz duszek chodził po scenie, poruszany strzałkami na klawiaturze. Potrzebujemy bloczków:



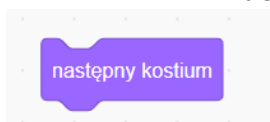
Bloczek “kiedy klawisz spacja naciśnięty” ma menu rozwijalne po kliknięciu w strzałkę.

Jak zaprogramować, żeby duszek chodził w prawo, w lewo, w górę i w dół po naciśnięciu właściwej strzałki? Zwróć uwagę, że można to zrobić, wykorzystując przesunięcie o 10 kroków i -10 kroków:



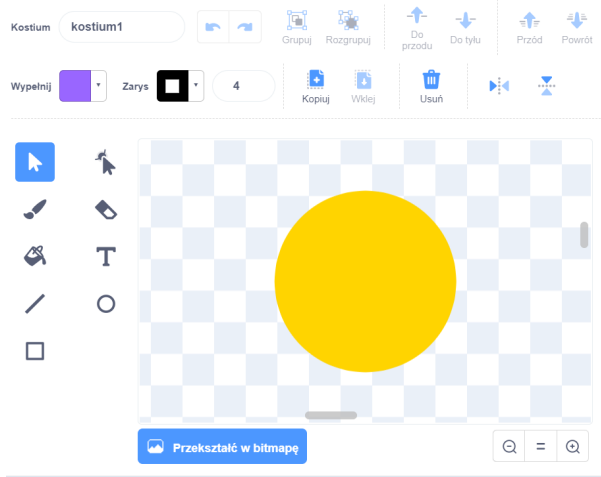
Zaprogramuj podobnie ruch duszka w górę i w dół. Zastanów się, czy można byłoby zaprogramować wszystkie ruchy pająka w jednym skrypcie?

Jeżeli chcesz, aby pajączek przy każdym ruchu ruszał też jamą ustną - dołącz do każdego skryptu bloczek z działu Wygląd, który zmieni kostium duszka.

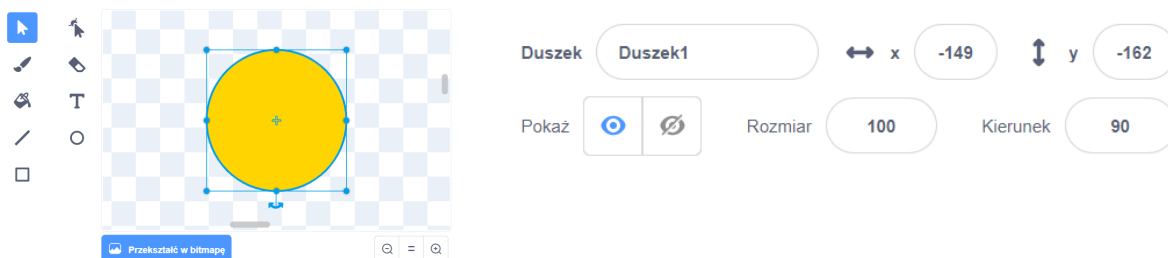


Krok 5:

**Narysuj duszki-denary:** Na mapie widzisz jeszcze kolorowe kółeczka, czyli tytułowe denary Mieszka I. Potrzebujemy tych duszków, żeby pajączek mógł je zbierać. W naszej grze jest ich 10. Są to kolorowe kółka narysowane w edytorze Scratcha (aby tam wejść, musisz z panelu “Wybierz duszka” na dole po prawej stronie wybrać opcję Maluj):



Wielkość denara możesz zmienić w opcji wektorowej - robiąc ramkę dookoła rysunku i rozciągając albo kurcząc kółko, albo zmniejszając wielkość duszka opcją Rozmiar:



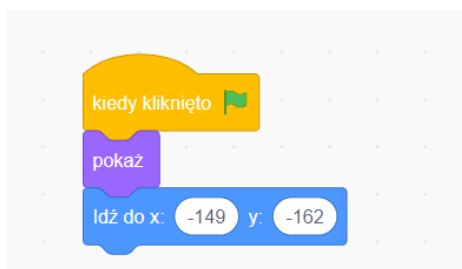
Wystarczy narysować jednego takiego duszka i potem skorzystać z opcji “duplikuj”, ale warto to zrobić po zaprogramowaniu duszka, bo wtedy kopiują się także skrypty duszka (a zmiany w każdym programie duszka-denara są niewielkie); jeżeli chcesz, aby denary miały różne kolory - po zduplikowaniu duszka wróć do edytora graficznego (zakładka Kostiumy na górnej belce) i zmień kolor.

Krok 6:

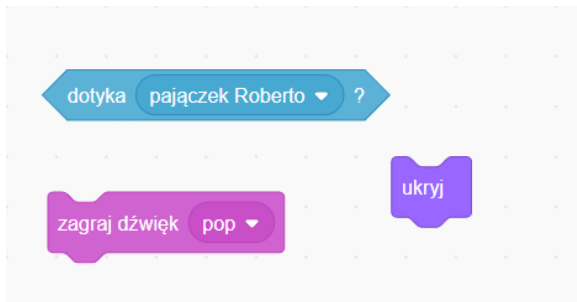
**Zaprogramuj działanie duszka-denara:** zaobserwujcie, co robi ten duszek.

1. Po kliknięciu w zieloną flagę denary znikają z mapy i pojawiają się na dole.
2. Po naciśnięciu spacji denary dość wolno lecą w różne miejsca mapy (te miejsca to główne grody państwa Mieszka I).
3. Kiedy pajęczek dotyka denara - denar znika i wydaje dźwięk “pop”.

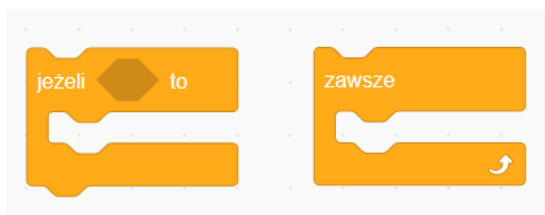
Po kliknięciu zielonej flagi duszek powinien się pokazać (bo zniknął po “zjedzeniu” go przez pajęczka) i znaleźć się na dole mapy, w jednym punkcie - wszystkie denary. Potrzebujemy bloczków:



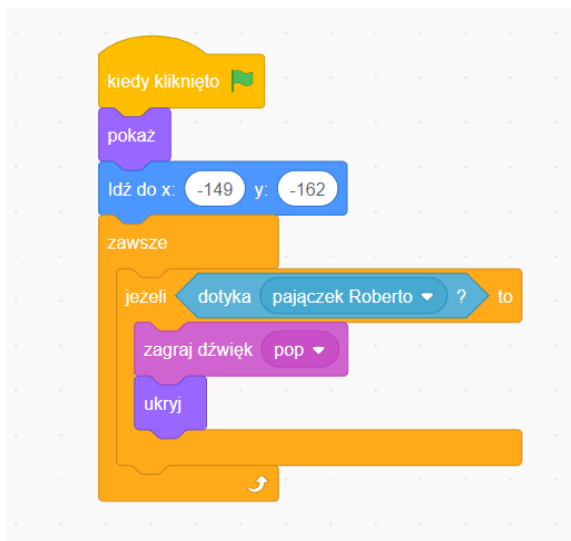
Teraz musimy zaprogramować sytuację, w której po dotknięciu przez pajęczka denar znika, wydając dźwięk. Potrzebujemy bloczków z działów Czujniki, Dźwięk i Wygląd:



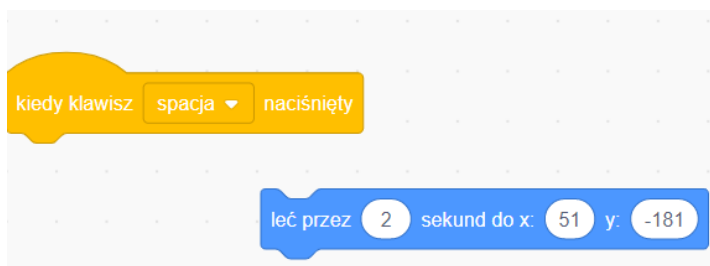
oraz warunku "jeżeli-to" i pętli "zawsze" z działu Kontrola:



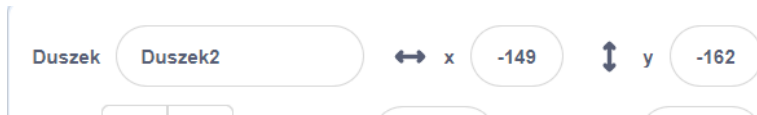
Zastanów się, jak połączyć te bloczki, żeby zawsze - jeżeli pajęczek dotyka denara - denar zniknął i wydawał dźwięk? Cały ten fragment kodu dolepiamy do wcześniej stworzonego kodu - zielona flaga-pokaż-idź do. Całość może wyglądać tak:



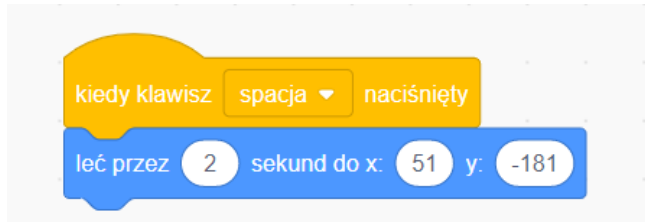
Pozostaje jeszcze zaprogramować lot denarów w odpowiednie miejsca mapy po kliknięciu spacji. Potrzebujemy takich bloczków (działy: Kontrola, Ruch i Zdarzenia)



Chodzi nam o to, aby po kliknięciu spacji każdy denar poleciał spokojnie na swoje miejsce na mapie. Aby wiedzieć, co wpisać w parametry x i y na bloczku "leć przez 2 sekundy do x...y..." musicie rozmieścić denary we właściwych miejscach mapy i spisać parametry z miejsca pod mapą:



Całość skryptu wykonującego lot denara wygląda tak:

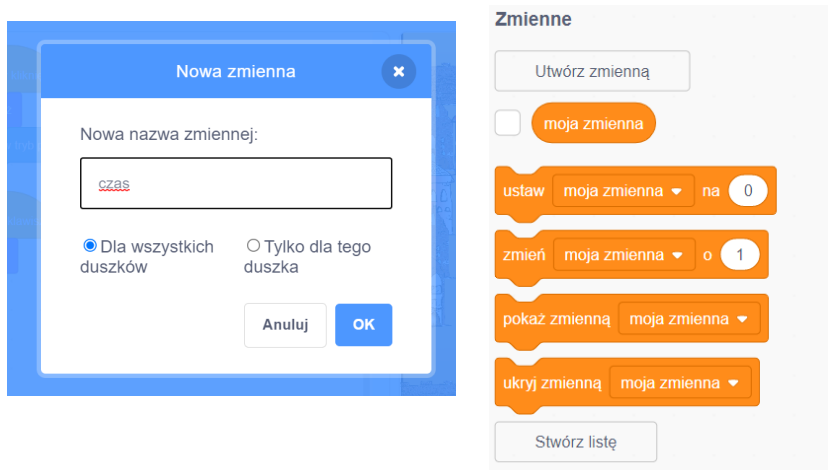


Jeżeli skorzystasz z naszej podpowiedzi i zduplikujesz duszki razem ze skryptami - to właśnie w tym miejscu na bloczku "leć przez 2 sekundy do x...y...." będzie jedyna różnica w skryptach denarów, ponieważ każdy denar po naciśnięciu spacji powinien znaleźć się w innym miejscu mapy. Wszystkie pozostałe skrypty denarów mogą być identyczne.

Krok 7:

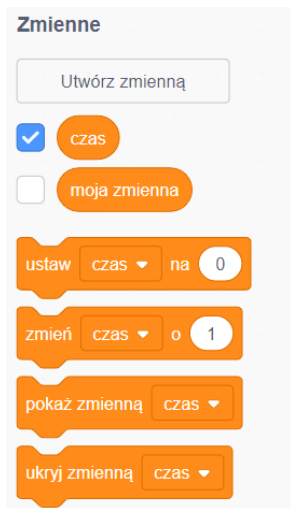
**Zaprogramuj licznik czasu:** pewnie zauważyłaś/eś, że na górze sceny pojawił się licznik czasu, który liczy sekundy i po 30 sekundach kończy grę. Aby zaprogramować ten element, trzeba sięgnąć do bloczków z działu Zmienne.

Aby wykorzystać zmienną, trzeba ją stworzyć i nazwać. Korzystamy z górnego okienka "Utwórz zmienną".



Nasza zmienna będzie się nazywała "czas" i musi działać tylko dla jednego duszka - pająka. Kiedy stworzymy zmienną, pojawią się automatycznie nowe bloczki w dziale Zmienne:



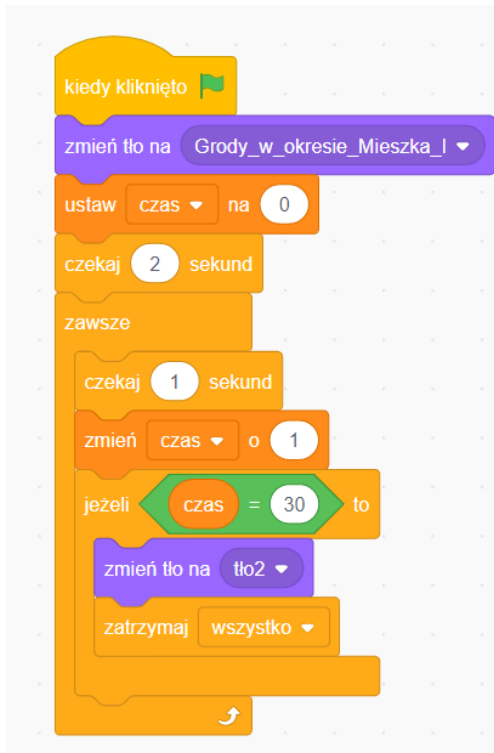


Jeżeli chcemy widzieć zmienną “czas na scenie”, musimy zaznaczyć kwadracik przy zmiennej tak, jak to widać powyżej.

Z działu Zmienne potrzebujemy następujących bloczków:



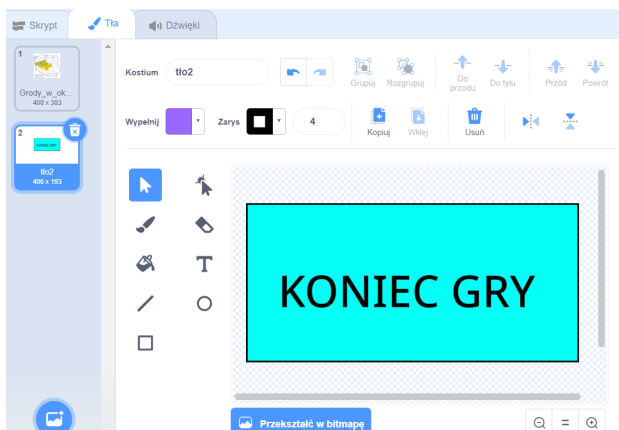
Chcemy, żeby na początku gry licznik czasu się zerował, żeby co sekundę czas zmieniał się o 1 i żeby gra kończyła się, kiedy skończy się zamierzony przez nas czas (u nas jest to 30 sekund, ale u Was może być krótszy). Więc po kliknięciu zielonej flagi należy ustawić czas na zero, przydałoby się też chwilę zaczekać, aby gracz zdążył spacją rozmieścić denary na mapie. Zaprogramuj teraz licznik czasu tak, aby zawsze (bloczek z Kontroli) po upływie 1 sekundy (czekaj 1 sekundę, bloczek z Kontroli) czas zmieniał się o 1 (bloczek ze Zmiennych), a jeżeli czas=30 (bloczki ze Zmiennych i z Wyrażeń) to (warunek jeżeli-to) tło zmienia się na “Koniec gry” i gra się zatrzymuje. Cały skrypt wygląda tak:



Pamiętaj, że zmieniamy tło na końcu gry, więc na początku musimy wrócić do mapy Grodów Mieszka I.

Krok 8:

**Narysuj tło do zakończenia gry:** wracamy do Sceny i zakładki Tła na górnej belce i rysujemy drugie tło - zakończenie gry, żeby nasz skrypt “zmień tło na tło 2” miał sens.



### Uwagi/alternatywy:

**Minimum:** lekcja o Mieszku I i początkach państwa polskiego; kolejny odcinek historii Tymka; zaprogramowanie gry “Denary Mieszka I”.

**Dodatki do wyboru lub wszystkie:**

- karty ze skarbami królów Polski jako nagroda dla uczniów;
- kolejne miejsca na mapie rysowanej przez uczniów na dużym papierze - **Ostrów Lednicki**;
- kolejną częścią generatora czasu będzie diadem, który w przyszłości Otton III włoży na głowę syna Mieszka I - Bolesława.

# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 3 :

# Za ciasna korona Bolesława Chrobrego

Lekcja, podczas której nauczyciel wprowadza uczniów w historię biskupa Wojciecha oraz zjazdu gnieźnieńskiego i koronacji Bolesława Chrobrego. Uczniowie poznają też kolejny fragment historii Tymka i sami wykonują animację o misji biskupa Wojciecha.

### Cele zajęć:

Uczeń powinien:

- potrafić opowiedzieć o panowaniu Bolesława Chrobrego,
- wskazać przyczyny i skutki zjazdu gnieźnieńskiego,
- znać dzieje biskupa Wojciecha,
- opowiedzieć, co się wydarzyło w roku 997, 1000 i 1025.

### NaCoBeZu

Po lekcji:

- wiesz, co się wydarzyło w: 997 r., 1000 r., 1025 r.,
- znasz dokonania biskupa Wojciecha i Bolesława Chrobrego,
- wymieniasz przyczyny i skutki zjazdu gnieźnieńskiego,
- rozumiesz znaczenie koronacji Bolesława Chrobrego.

### Materiały pomocnicze:

- film o Drzwiach Gnieźnieńskich <https://youtu.be/cs9X2RiKb2E>,
- podręcznik do historii klasa 5,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (wizerunki świętych związanych z Polską w średniowieczu),
- zdjęcie Drzwi Gnieźnieńskich,
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ Prusowie → misjonarz → wyprawa misyjna → poganie → relikwie → męczennik → święty → arcybiskupstwo → cesarz → zjazd gnieźnieński → diadem → koronacja → król

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka,
- algorytmika w programie Scratch.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach (tworzenie animacji).

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia V. Polska w okresie wczesnopiastowskim.

Uczeń:

3) charakteryzuje rozwój i kryzys monarchii Bolesława Chrobrego i Mieszka II.

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

3) sterowanie robotem lub obiektem na ekranie;

3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:

1). pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,

2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Nauczyciel opowiada uczniom, że dzięki pomocy uczniów Tymek dokonał kolejnego skoku w czasie i w przestrzeni tak, że znalazł się na dworze Bolesława Chrobrego i poznał biskupa Wojciecha, który był gościem Chrobrego.

### 2. Część zasadnicza

czas: ok. 15 minut.

Biskup Wojciech poprosił Chrobrego o pomoc, ponieważ postanowił wybrać się na wyprawę misyjną, żeby nawrócić pogańskich Prusów. Tymek postanawia towarzyszyć wyprawie, a o historii Wojciecha uczniowie dowiedzą się z filmu - animacji historii pokazanej na Drzwiach Gnieźnieńskich:

<https://youtu.be/cs9X2RiKb2E>

Nauczyciel opowiada też o cesarzu Ottonie III, który postanowił odbyć pielgrzymkę do grobu biskupa Wojciecha i opisuje uczniom zjazd gnieźnieński (czyta fragmenty tekstu źródłowego). Mówi też o planach Ottona III odrodzenia Cesarstwa Rzymskiego, również na ziemiach słowiańskich. Uczniowie dowiadują się o koronacji Bolesława Chrobrego i o wojnach, które toczył.

Tekst źródłowy

ZJAZD GNIEŹNIEŃSKI ROKU 1000 W PRZEKAZIE ANONIMA ZWANEGO GALLEM  
(PRZEKŁAD R. GRODECKIEGO, BIBLIOTEKA NARODOWA SERIA I, NR 59, KRAKÓW 1923)

[6] O pierwszym Bolesławie, którego zwano Sławnym lub Chrobrym.

Pierwszy więc książę polski Mieszko dostał łaski chrztu za sprawą wiernej żony; a dla sławy jego i chwały w zupełności wystarczy [jeśli powiemy], że za jego czasów i przez niego Światłość niebiańska nawiedziła królestwo polskie. Z tej to bowiem błogosławionej niewiasty spłodził sławnego Bolesława, który po jego śmierci po męsku rządził królestwem i za łaską Bożą w taką wzrósł cnotę i potęgę, iż ozłocił - że tak powiem - całą Polskę swą zacnością. Któż bowiem zdoła godnie opowiedzieć jego mężne czyny i walki stoczone z narodami okolicznymi, a cóż dopiero na piśmie przekazać [je] pamięci? Czyż to nie on ujarzmił Morawy i Czechy, a w Pradze stolec książęcy zagarnął i swym zastępcom go porучzył? Czyż to nie on wielokrotnie pokonał w bitwie Węgrów i cały ich kraj aż po Dunaj zagarnął pod swoją władzę? Nieposkromionych zaś Sasów z taką mocą poskromił, że w środku ich ziemi żelaznymi słupami [wbitymi] w rzece Sali oznaczył granice Polski. Czyż zresztą potrzeba dokładnie wymieniać jego zwycięstwa i tryumfy nad ludami niewiernymi, skoro wiadomo, że je niejako swymi stopami podeptał! On to bowiem Selencję, Pomorze i Prusy do tego stopnia albo starł, gdy się przy pogaństwie upierały, albo też, nawrócone, umocnił w wierze, iż wiele tam kościołów i biskupów ustanowił za zgodą papieża, a raczej papież [ustanowił je] za jego pośrednictwem. On to również, gdy przybył doń św. Wojciech, doznawszy wielu krzywd w długiej wędrówce, a [poprzednio] od własnego buntowniczego ludu czeskiego - przyjął go z wielkim uszanowaniem i wiernie wypełniał jego pouczenia i zarządzenia. Święty zaś męczennik płonąc ogniem miłości i pragnieniem głoszenia wiary, skoro spostrzegł, że już nieco rozkrzewiła się w Polsce wiara i wzrósł Kościół święty, bez trwogi udał się do Prus i tam męczeństwem dopełnił swego zawodu. Później zaś ciało jego Bolesław wykupił na wagę złota od owych Prusów i umieścił [je] z należytą czcią w siedzibie metropolitalnej w Gnieźnie.

Również i to uważamy za godne przekazania pamięci, że za jego czasów cesarz Otto Rudy przybył do [grobu] św. Wojciecha dla modlitwy i pojednania, a zarazem w celu poznania sławnego Bolesława, jak o tym można dokładniej wyczytać w księdze o męczeństwie [tego] świętego. Bolesław przyjął go tak zaszczytnie i okazał, jak wypadło przyjmując króla, cesarza rzymskiego i dostojnego gościa. Albowiem na przybycie cesarza przygotował przedziwne [wprost] cuda; najpierw hufce przeróżne rycerstwa, następnie dostojników rozstawił, jak chóry, na obszernej równinie, a poszczególne, z osobna stojące hufce wyróżniała odmienna barwa strojów. A nie była to [tania] pstrokaczna byle jakich ozdób, lecz najkosztowniejsze rzeczy, jakie można znaleźć gdziekolwiek na świecie. Bo za czasów Bolesława każdy rycerz i każda niewiasta dworska zamiast sukien lnianych lub wełnianych używali płaszczy z kosztownych tkanin, a skór, nawet bardzo cennych, choćby były nowe, nie noszono na jego dworze bez [podszycia] kosztowną tkaniną i bez złotych frędzli. Złoto bowiem za jego czasów było tak pospolite u wszystkich jak [dziś] srebro, srebro zaś było tanie jak słoma. Zważywszy jego chwałę, potęgę i bogactwo, cesarz rzymski zawołał w podziwie: "Na koronę mego cesarstwa! to, co widzę, większe jest, niż wieść głosiła!" I za radą swych magnatów dodał wobec wszystkich: "Nie godzi się takiego i tak wielkiego męża, jakby jednego spośród dostojników, księciem nazywać lub hrabią, lecz [wypada] chlubnie wynieść go na tron królewski i uwieńczyć koroną". A zdjąwszy z głowy swej diadem cesarski, włożył go na głowę Bolesława na [zadatek] przymierza i przyjaźni, i za chorągiew tryumfalną dał mu w darze gwóźdź z krzyża Pańskiego wraz z włócznią św. Maurycego, w zamian za co Bolesław ofiarował mu ramię św. Wojciecha. I tak wielką owego dnia złączyli się miłością, że cesarz mianował go bratem i współpracownikiem cesarstwa i nazwał go przyjacielem i sprzymierzeńcem narodu rzymskiego. Ponadto zaś przekazał na rzecz jego oraz jego następców wszelką władzę, jaka w zakresie [udzielania] godności kościelnych przysługiwała cesarstwu w królestwie polskim, czy też w innych podbitych już przez niego krajach barbarzyńców, oraz w tych, które podbije [w przyszłości]. Postanowienia tego układu zatwierdził [następnie] papież Sylwester przywilejem św. Rzymskiego Kościoła.

Bolesław więc, tak chlubnie wyniesiony na królewski tron przez cesarza, okazał wrodzoną sobie hojność urządzając podczas trzech dni swej konsekracji prawdziwie królewskie i cesarskie biesiady i codziennie zmieniając wszystkie naczynia i sprzęty, a zastawiając coraz to inne i jeszcze bardziej kosztowne. Po zakończeniu bowiem biesiady nakazał cześnikom i stolnikom zebrać ze wszystkich stołów z trzech dni złote i srebrne naczynia, bo żadnych drewnianych tam nie było, mianowicie kubki, puchary, misy, czarki i rogi, i ofiarował je cesarzowi dla uczczenia go, nie zaś jako dań [należną] od księcia. Komornikom zaś rozkazał zebrać rozciągnięte zasłony i obrusy, dywany, kobierce, serwety, ręczniki, i cokolwiek użyte było do nakrycia, i również znieść to wszystko do izby zajmowanej przez cesarza. A nadto jeszcze złożył [mu] wiele innych darów, mianowicie naczyń złotych i srebrnych rozmaitego wyrobu i różnobarwnych płaszczy, ozdób niewidzianego [dotąd] rodzaju i drogich kamieni; a tego wszystkiego tyle ofiarował, że cesarz tyle darów uważał za cud. Poszczególnych zaś jego książąt tak okazał obdarował, że z przyjaznych zrobił ich sobie największymi przyjaciółmi. Lecz któż zdoła wyliczyć, ile i jakich darów dał przedniejszym, skoro nawet nikt z tak licznej służby nie odszedł bez podarunku! Cesarz tedy wesóło z wielkimi darami powrócił do siebie, Bolesław zaś, podniesiony do godności królewskiej, wznowił dawny gniew ku wrogom.

### 3. Podsumowanie i ewaluacja

**czas: ok 5 minut**

Nauczyciel nawiązuje do NaCoBeZu, sprawdzając, czy cele lekcji zostały osiągnięte.

Tymkowi znów udało się zrobić skok w czasie. Teraz potrzebuje karty ze świętym Wojciechem (z kolekcji kart z polskimi świętymi), bo elementem generatora jest mitra biskupa Wojciecha. Informuje uczniów, za pośrednictwem nauczyciela, że jego przeciwnikiem jest niemiecki chłopiec, Jonas, z rodu wielkiego mistrza krzyżackiego, zabitego pod Grunwaldem.

### Programowanie:

Zadaniem uczniów na tej lekcji będzie wykonanie animacji w oparciu o zdjęcia - fragmenty Drzwi Gnieźnieńskich. Uczniowie stworzą w parach program, w którym duszek będzie opowiadał dzieje św. Wojciecha i będą zmieniały się slajdy z Drzwi Gnieźnieńskich.

Zdjęcie Drzwi Gnieźnieńskich z Wikipedii:

Autorstwa Nieznany – scanned from Drzwi Gnieźnieńskie vol. 2, Zakład Imienia Ossolińskich, Wrocław, Poland, 1956, Domena publiczna,

<https://commons.wikimedia.org/w/index.php?curid=8467451>

Kolejne tła z animacji są wycinkami z tego właśnie zdjęcia.

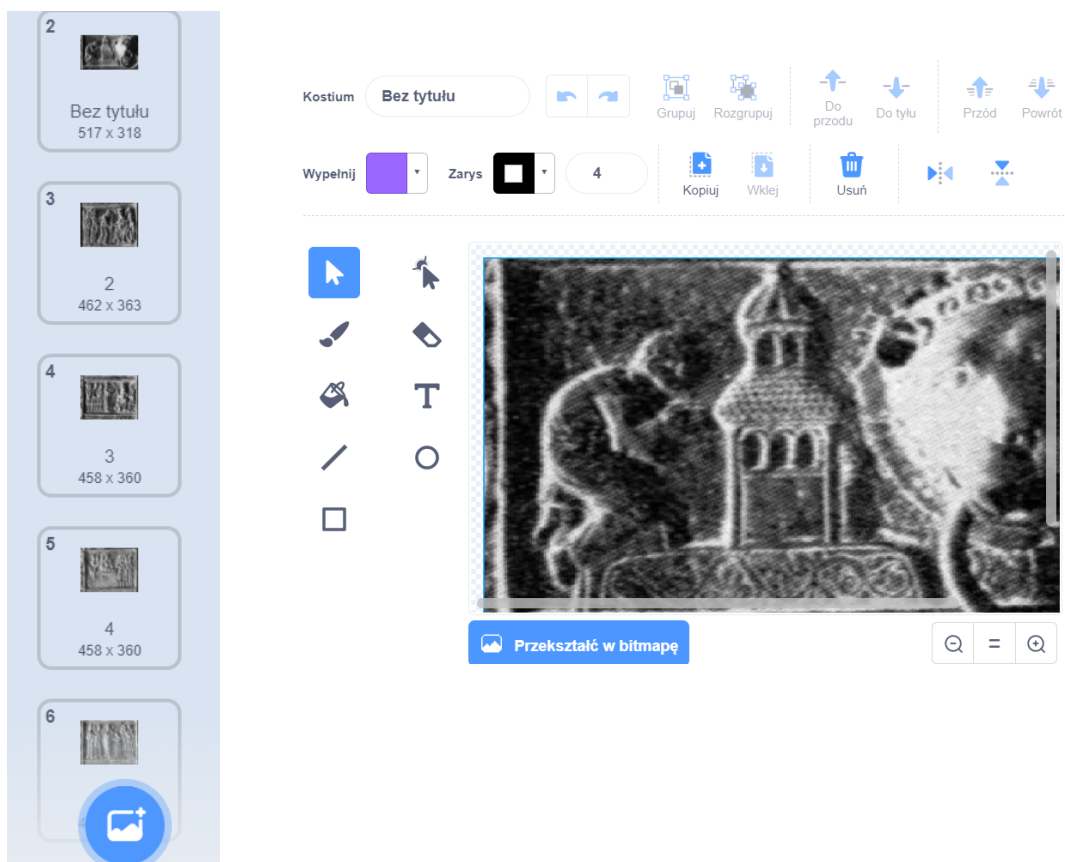
Animacja: <https://scratch.mit.edu/projects/545722352/>

Krok 1:

**Pobaw się tą animacją** i zastanów się, co jest duszkiem, a co tłem.

Krok 2:

**Osadź fragmenty Drzwi Gnieźnieńskich jako tła**, które będą podstawą animacji. W naszej animacji zostały one wycięte z dużego zdjęcia z Wikipedii (link powyżej). Możesz wziąć nasze tła, albo samodzielnie wybrać fragmenty zdjęcia, wyciąć je skrótami win+shift+s, wkleić do painta, ewentualnie zmienić rozmiar i zapisać w postaci JPG lub PNG. Nasze tła wyglądają tak:

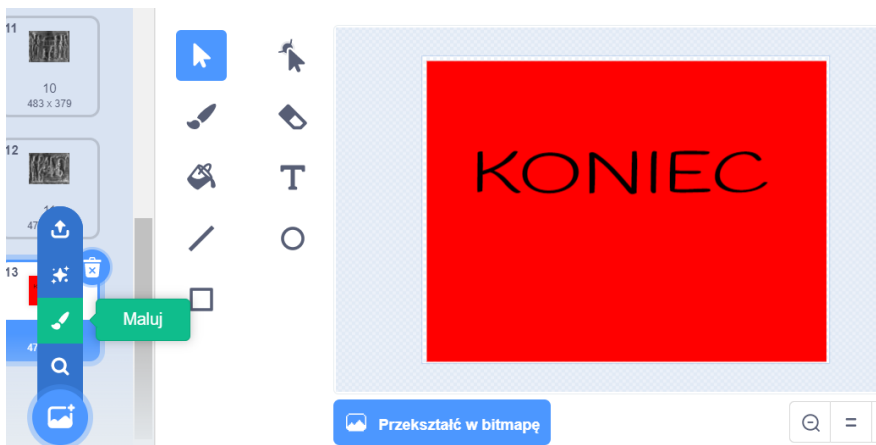
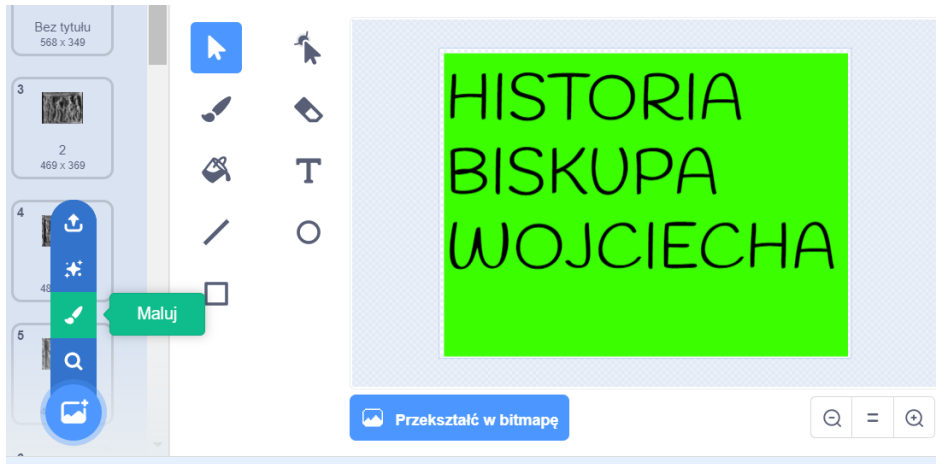


Wybraliśmy 12 kwater Drzwi Gnieźnieńskich (czyli prostokątnych części, przedstawiających sceny z życia świętego Wojciecha). Każde tło powinno zajmować całą scenę. Pomoże Ci w tym edytor graficzny tła w postaci wektora. Tła nie mają w tym programie żadnych skryptów.

Krok 3:

**Osadź okładki animacji - planszę początkową i końcową.**

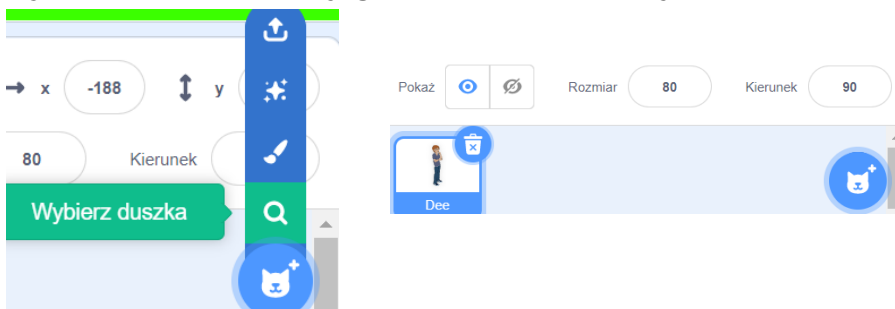
Narysuj w edytorze sceny (zakładka Tła na górnej belce) planszę początkową i końcową, wybierając z menu tła "Maluj":

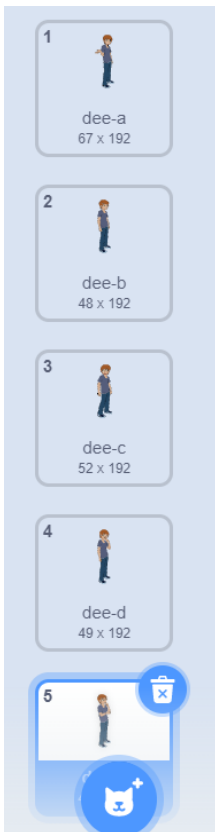


Okładkę początkową przesunij na górę w tłach przy pomocy myszki.

Krok 4:

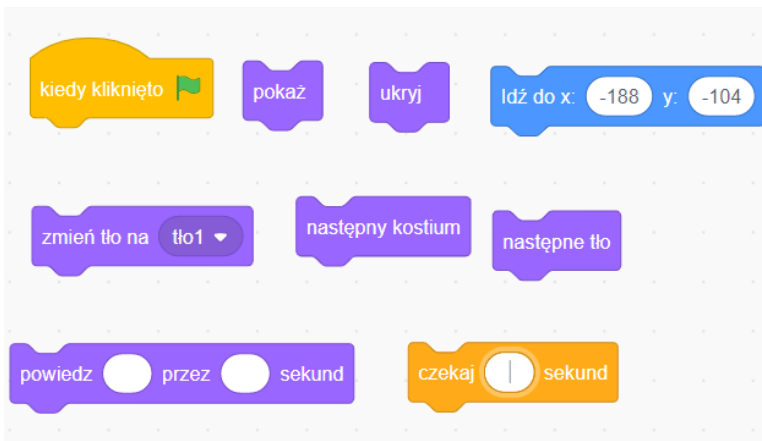
**Wybierz duszka, ustaw jego rozmiar i sprawdź, jakie ma kostiumy:**





Krok 5:

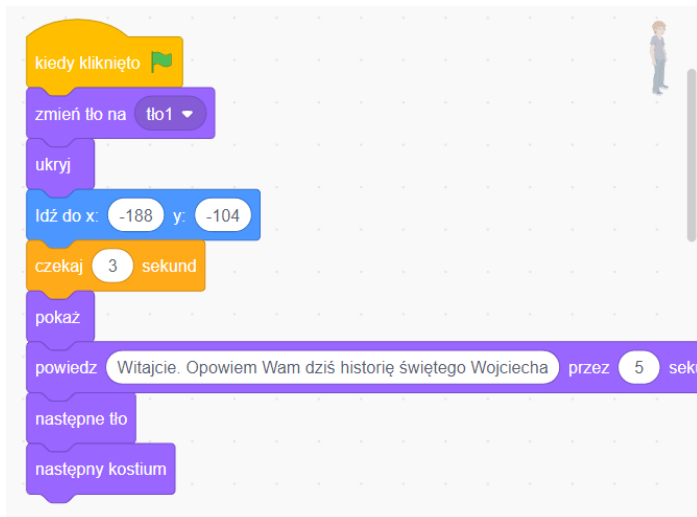
**Zaprogramuj działania duszka.** Co robi nasz duszek? Początkowo go nie widać, potem się pojawia, wita się i zapowiada, co będzie robił. To powinno się stać po kliknięciu zielonej flagi. Zauważ też, że duszek się porusza, czyli zmienia kostiumy. Kiedy kończy mówić, zmienia się tło. Potrzebne nam bloczki to:



z działów: Zdarzenia, Kontrola, Wygląd i Ruch. Zastanów się, jak je połączyć, żeby duszek wykonał zaplanowane działania?

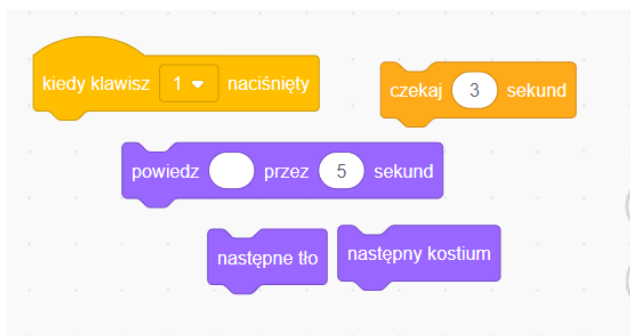
Cały skrypt może wyglądać tak:





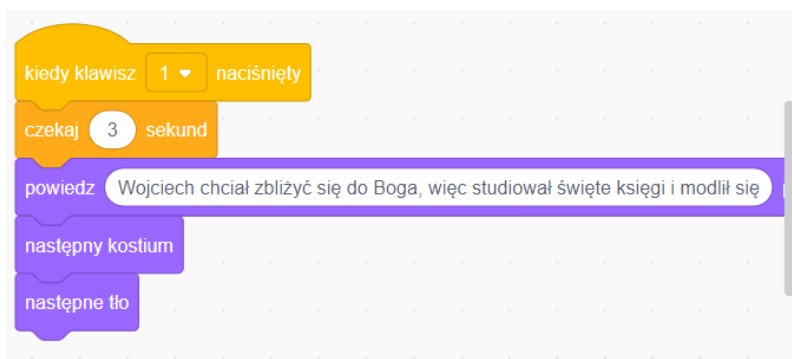
Okladka zmieniła się na pierwsze tło - kwaternę Drzwi Gnieźnieńskich.

Teraz pozostaje nam zaprogramować skrypt kolejnych części opowieści. Każdy kolejny skrypt będzie bardzo podobny, zmieniać się będzie tło - na kolejne kwaterny Drzwi Gnieźnieńskich oraz tekst opowieści naszego duszka. Potrzebujemy bloczków:



Trzeba je połączyć tak, aby po kliknięciu w kolejną cyfrę na klawiaturze, po chwili pauzy duszek opowiedział kolejną część opowieści, zmienił kostium i przełączył nam tło na kolejną część opowieści z Drzwi Gnieźnieńskich.

Gotowy skrypt może wyglądać tak:



Trzeba dać kilka sekund, żeby widz obejrzał obrazek i kilka sekund, żeby widz przeczytał tekst duszka. U nas na koniec każdego skryptu duszek zmienia kostium i zmienia też tło. Zmiana tła może jednak być początkiem kolejnego skryptu, zamiast tak, jak u nas, końcem skryptu poprzedniego. Ostatni skrypt musi się kończyć okładką końcową (która też jest "następnym tłem" w bloczku).

## Uwagi/alternatywy:

**Minimum:** lekcja o dziejach biskupa Wojciecha, zjeździe gnieźnieńskim i koronacji Chrobrego; dalszy ciąg historii Tymka i zaprogramowanie animacji o św. Wojciechu.

### Dodatki do wyboru lub wszystkie:

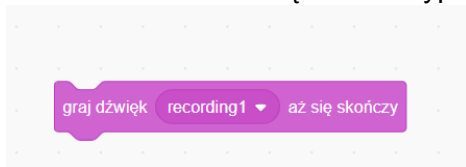
- karty ze świętymi średniowiecznymi, związanymi z Polską jako nagrody dla uczniów,
- kolejny punkt na dużej papierowej mapie - **miejsce śmierci św. Wojciecha**

(za Wikipedią: "Miejsce śmierci, znajdujące się według źródeł w pobliżu pruskiego grodu *Cholin* lub *Chollinun* lokalizuje się albo na południowych brzegach jeziora [Drużno](#) (tradycyjnie w pobliżu wsi [Święty Gaj](#) koło [Pastęka](#) w [powiecie elbląskim](#)<sup>[1]</sup> lub według [Gerarda Labudy](#) w pobliżu wsi [Pachoły](#) koło [Dzierzgonia](#) w [powiecie sztumskim](#)<sup>[3]</sup>) albo na [Sambii](#) (dziś w [obwodzie kaliningradzkim](#), np. w miejscowości [Tenkity](#), dziś Letnoje, lub [Rybaki](#), dziś Primorsk<sup>[4]</sup>).

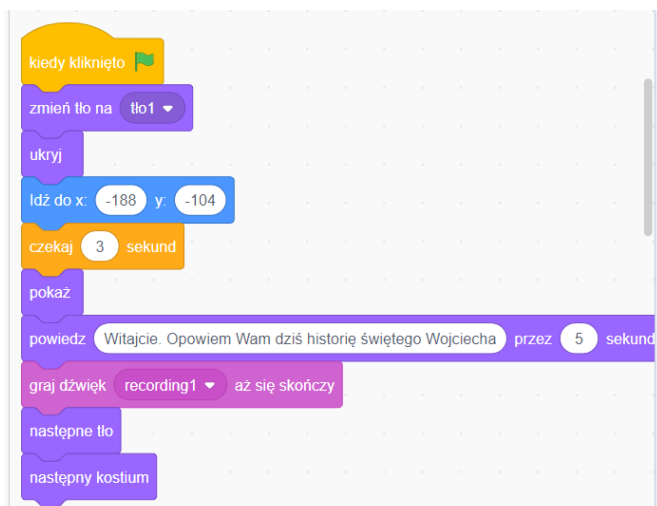
W animacji można wprowadzić nowe elementy - duszek może mówić dzięki nagrywaniu tekstu.



### Wstawianie dźwięku do skryptu:



Tak więc duszek może opowiadać historię Waszymi głosami, można to połączyć z tekstem pisany, albo pozostać tylko przy głosie. W naszej animacji skrypt pierwszy zawiera taki właśnie bloczek dźwiękowy.



# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 4:

# Kraj szyty grubymi nićmi - krzyżackie nożyce

Lekcja, podczas której nauczyciel opowiada uczniom o czasach rozbicia dzielnicowego, dowiadują się też, skąd w sąsiedztwie Polski wzięli się Krzyżacy i jak napadli nas Mongołowie. Uczniowie stworzą grę, w trakcie której będą utrwalali swoją znajomość władców dzielnicowych, a Tymkowi nie uda się powstrzymać Konrada Mazowieckiego przed jego największym błędem.

### Cele zajęć:

Uczeń powinien:

- omawiać konsekwencje rozbicia dzielnicowego w różnych dziedzinach życia,
- wiedzieć, jak układały się stosunki Polski z sąsiadami w okresie rozbicia dzielnicowego,
- rozumieć przyczyny sprowadzenia Krzyżaków do Polski, omawiać proces rozwoju ich państwa,
- wiedzieć, kiedy i z jakimi skutkami Mongołowie najechali na państwo polskie.

### NaCoBeZu

- wymieniasz skutki rozbicia dzielnicowego,
- wiesz, kim byli Krzyżacy i Mongołowie, wiesz, jak układały się stosunki Polski z nimi,
- omawiasz proces sprowadzenia Krzyżaków do Polski,
- wiesz co się stało w: 1226 r., 1241 r.

### Materiały pomocnicze:

- podręcznik do historii klasa 5,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (władcy dzielnicowi),
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ dzielnicowa → rozbicie dzielnicowe → testament Bolesława Krzywoustego → lenno → Prusowie, Litwini i Jaćwingowie → Konrad Mazowiecki → Krzyżacy → zakon → Mongołowie → bitwa pod Legnicą

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka,
- algorytmika w programie Scratch,
- gra edukacyjna- grywalizacja.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach (tworzenie gry).

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia

VI. Polska w okresie rozbicia dzielnicowego.

Uczeń:

- 1) umieszcza w czasie i przestrzeni Polskę okresu rozbicia dzielnicowego;
- 2) opisuje przyczyny i wskazuje skutki rozbicia dzielnicowego;
- 3) umieszcza w czasie najważniejsze wydarzenia związane z relacjami polsko-krzyżackimi oraz zagrożeniem najazdami tatarskimi w okresie rozbicia dzielnicowego.

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

- I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
  1. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
  - 2) sterowanie robotem lub obiektem na ekranie;
  3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
  1. projektuje, tworzy i zapisuje w wizualnym języku programowania:
    - 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
    - 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Podczas tej lekcji, po przypomnieniu testamentu Bolesława Krzywoustego przez nauczyciela w postaci prezentacji mapy Polski podzielonej na dzielnice - uczniowie pomagają Tymkowi rozwikłać zagadkę Krzyżaków. Ponieważ wiemy z poprzednich zajęć, że tajemniczym nieprzyjacielem jest agent mówiący językiem niemieckim - potomek Wielkiego Mistrza Zakonu NMP Domu Niemieckiego w Jerozolimie - Jonas. Skok w przestrzeni Tymka doprowadził nas do Płocka i Księcia Konrada.

### 2. Część zasadnicza

czas: ok. 15 minut.

Nauczyciel wyjaśnia temat lekcji, opowiada o kłopotach podzielonej Polski i przedstawia trudności, jakie napotkał Konrad Mazowiecki w walce z Prusami i Żmudzinami (Litwinami). Opisuje sytuację, w jakiej znalazła się Polska - uwzględniając najazd południa kraju przez Tatarów. Może wykorzystać legendę o hejnale w Krakowie (Klechdy domowe).

Przenosimy się na Mazowsze (Płock) - gdzie książę Konrad ma problem z Jaćwingami i Prusami (a jakże). Panowie z pawimi czubami wygnani z Węgier dostali się na Mazowsze i rozpoczęli to, co potrafili najlepiej - knuć i budować potęgę, korzystając z osiągnięć ludzkości - umiejętności czytania i pisania. Dowiadujemy się w tej części, że Tymoteusz stara się odradzić Konradowi decyzję zaproszenia Krzyżaków, wykorzystując fakt wygnania ich przez króla Węgier, jednak podmieniono mu listy (oczywiście Jonas) - i rozgościli się na darowanych ziemiach (tutaj można pokazać słabość rozbitej dzielnicowo Polski).

### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

Nauczyciel powraca do NaCoBeZu, aby sprawdzić czy osiągnął cele lekcji.

**Programowanie:** uczniowie wykonują grę "Rozbicie dzielnicowe - władcy i ich dzielnice", w oparciu o mapę:

Autorstwa Farary - Praca własna w oparciu m.in. o: Samsonowicz H., Historia Polski do roku 1795, Warszawa 1990, s. 36–38. ISBN 83-02-04285-4., Domena publiczna,

<https://commons.wikimedia.org/w/index.php?curid=10054812>

Link do gry:

<https://scratch.mit.edu/projects/536538925/>

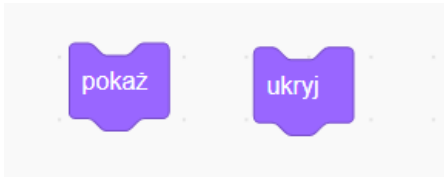
Krok 1:

**Zagraj w tą grę** i zastanów się, jakie elementy ją tworzą? Co jest duszkiem, a co sceną? Na czym polega gra? Co znajdziesz w bibliotece Scratcha, a co musisz utworzyć?

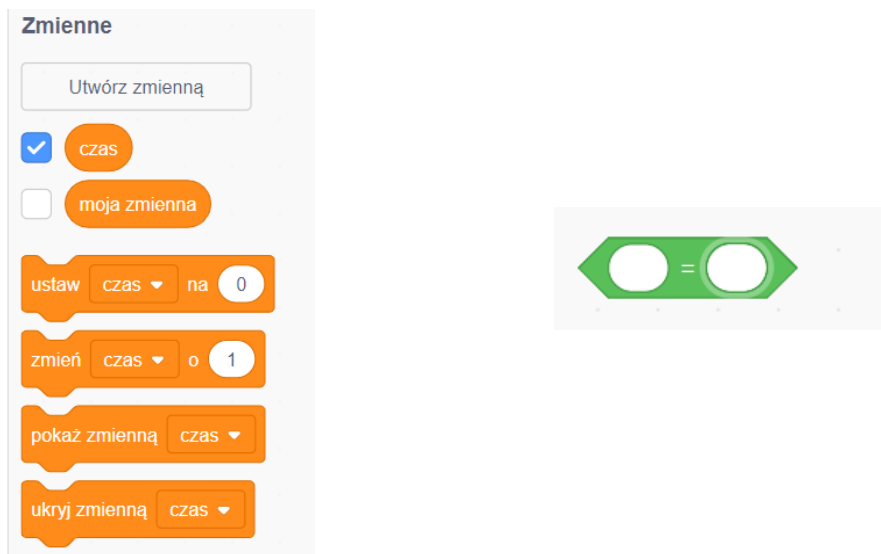
**Zajrzyj do skryptów** i sprawdź czy miałaś/łeś rację.

Krok 2

**Zaprogramuj:** duszka mapę. Mapa pochodzi z Wikipedii (link jest nad linkiem do gry w tym scenariuszu). Gra zaczyna się bez tego duszka (widocznie został ukryty), więc po kliknięciu zielonej flagi powinien się on pokazać. Przydadzą nam się bloczki z działu Wygląd:



Najważniejszym zadaniem duszka mapa będzie liczenie czasu gracza i wykonywanie kolejnych działań wtedy, kiedy czas się skończy. Dzięki grze "Denary Mieszka I" wiesz już, jak stworzyć licznik czasu. W dziale Zmienne utwórz zmienną "czas". Musisz ją zaznaczyć w okienku z lewej strony, żeby było widać licznik czasu na scenie. Zmienna musi działać dla wszystkich duszków. Wykorzystamy też bloczek "zmień czas o" i "ustaw czas na" oraz okrągły bloczek "czas". Przyda się też z wyrażień bloczek ze znakiem =



Korzystając z tych wszystkich bloczków, zaprogramuj następujące działania:

1. Pokazanie się mapy.
2. Ustawienie licznika czasu na 25 sekund (tym razem czas będzie się odliczał do zera).
3. 4,5 sekundy pauzy na działania innych duszków.
4. Zmianę licznika co sekundę.
5. Zniknięcie mapy, kiedy czas się skończy.

Potrzebne będą: pętla "zawsze" i warunek "jeżeli-to".

Przykładowy skrypt może wyglądać tak:

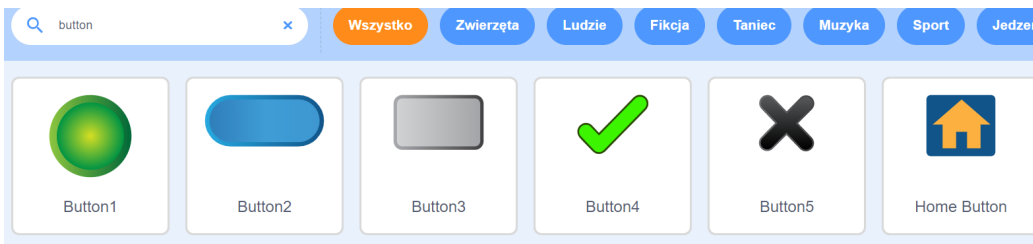


Brakuje nam tu jeszcze zakończenia gry, ale do tego potrzebujemy innego duszka, więc wrócimy do tego skryptu później.

Krok 3:

**Stwórz** duszki z nazwami dzielnic Polski, zgodnie z testamentem Bolesława Krzywoustego.

Te duszki to gotowe przyciski (button) z biblioteki Scratcha. Otwórz Bibliotekę Duszków i wpisz "button":



Potrzebujemy przycisku, na którym można pisać. Przycisków musi być 7- tyle, ile dzielnic do podpisania. Możesz ściągnąć jeden przycisk i zdublować go kilkakrotnie.

Teraz trzeba wejść w kostiumy duszka. Nasz przycisk ma dwa kostiumy - potrzebny nam jest tylko jeden, więc drugi można usunąć (można wybrać dowolny). Po przekształceniu duszka w wektor możemy łatwo zmieniać jego rozmiar i kształt, a po kliknięciu w literę w menu z lewej strony - możemy na nim pisać:



Nazwy dzielnic weź z mapy, jednak muszą być dość krótkie, żeby zmieściły się na przycisku. Sam przycisk powinien w pełni mieścić się na swojej dzielnicy na mapie.

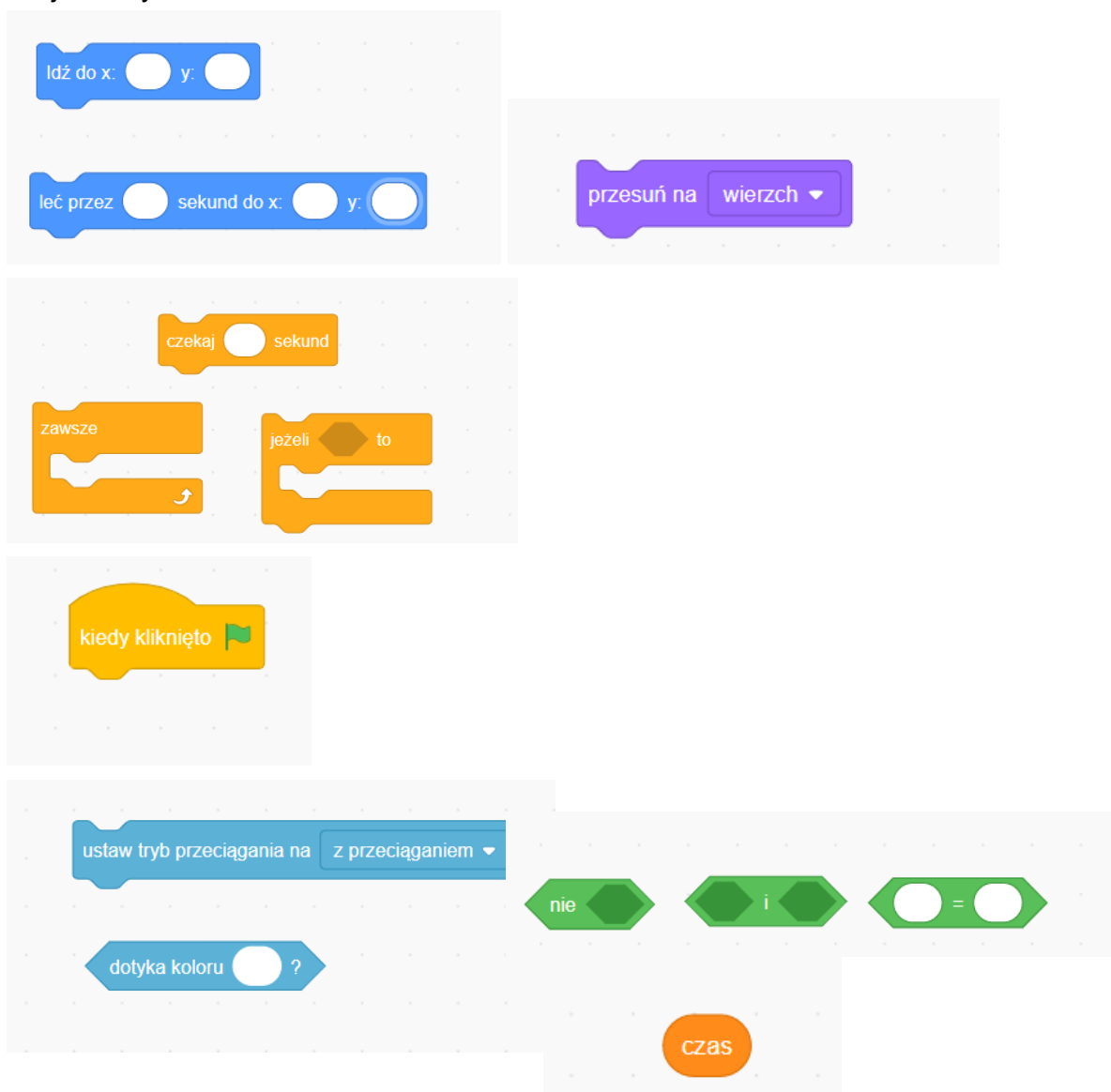
Krok 4:

### Zaprogramuj przycisk:

Zastanów się, co musi robić przycisk.

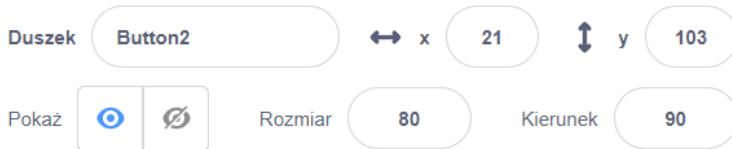
1. Kiedy gra się zaczyna, na scenie widać przyciski.
2. Po kliknięciu zielonej flagi pojawia się mapa i przyciski przesuwają się do swoich dzielnic.
3. Potem uciekają na boki sceny.
4. Wtedy zaczyna się gra - gracz musi w 25 sekund przydzielić nazwę każdej dzielnicy.
5. Przycisk musi więc dać się przeciągać na scenie w czasie gry.
6. Musi zostawać na swojej dzielnicy i nie dać się zostawić na innej dzielnicy (przycisk wraca na miejsce).
7. Kiedy kończy się czas, przyciski wracają na boki sceny, znika mapa i pojawia się nowy duszek, rycerz.

Jak to zaprogramować? Potrzebne nam będą bloczki z działów: Ruch, Wygląd, Zdarzenia, Kontrola, Czujniki, Wyrażenia, Zmienne:



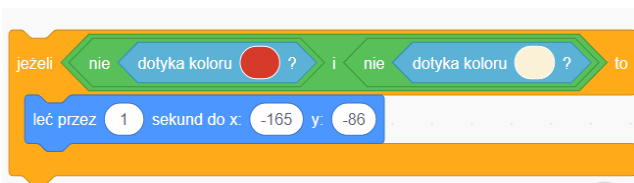
### Podpowiedzi:

- każdy przycisk na początku melduje się na swojej dzielnicy. Wykorzystaj mapę i sprawdź, jakie położenie ma przycisk, kiedy tam się znajduje:

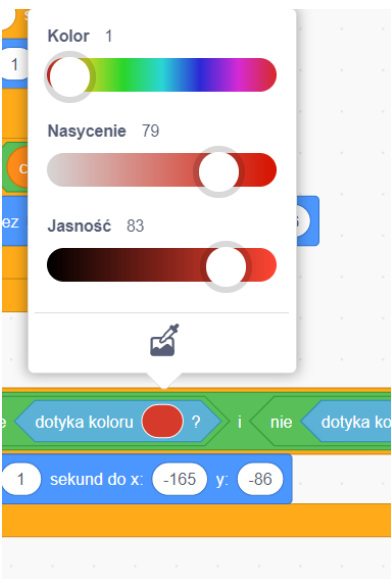


- dobrze będzie przesunąć przycisk na wierzch i trzeba pozwolić na przeciąganie go na scenie,
- po chwili na mapie przyciski powinny polecieć na boki sceny; sprawdź, jakie ma parametry przycisk, kiedy ustawisz go obok mapy,
- kiedy czas=0 przyciski muszą wrócić na te same miejsca obok mapy.

Nową w tej grze rzeczą jest komunikacja pomiędzy duszkami - w tym wypadku pomiędzy mapą, konkretnymi kolorami dzielnic, a odpowiednimi przyciskami. Przycisk musi rozpoznać, czy jest we właściwym miejscu i zareagować, czyli zostać albo wrócić na bok. Możemy wykorzystać czujnik koloru:

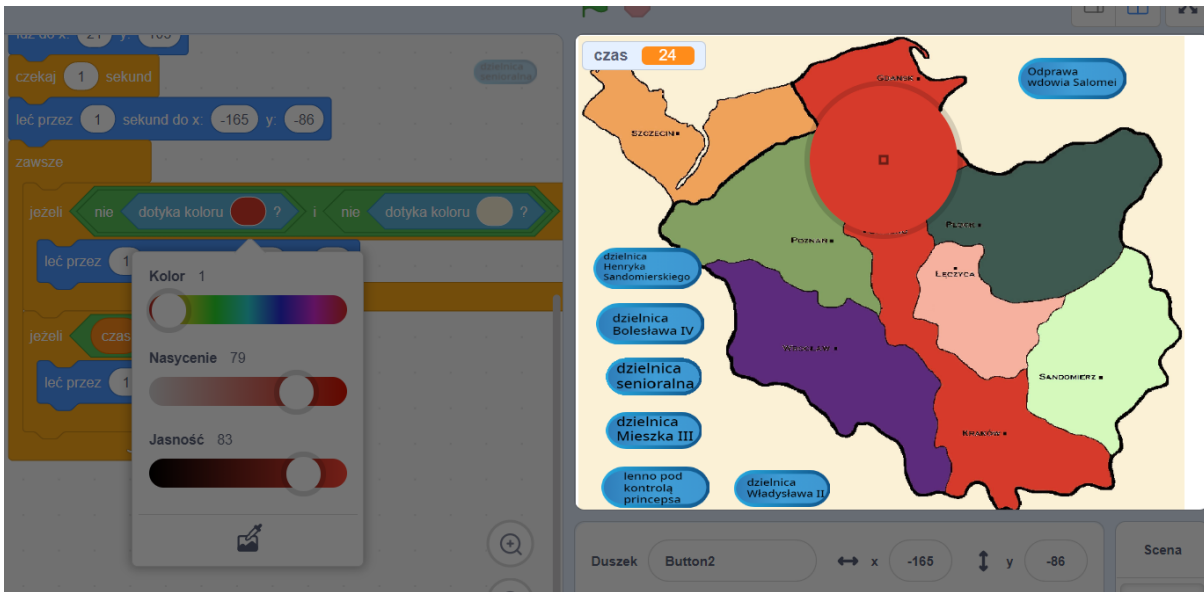


Kolor trzeba pobrać bardzo dokładnie, przy pomocy narzędzia pobierania koloru, które otwiera się, gdy dwa razy klikniemy w pole koloru na czujniku:



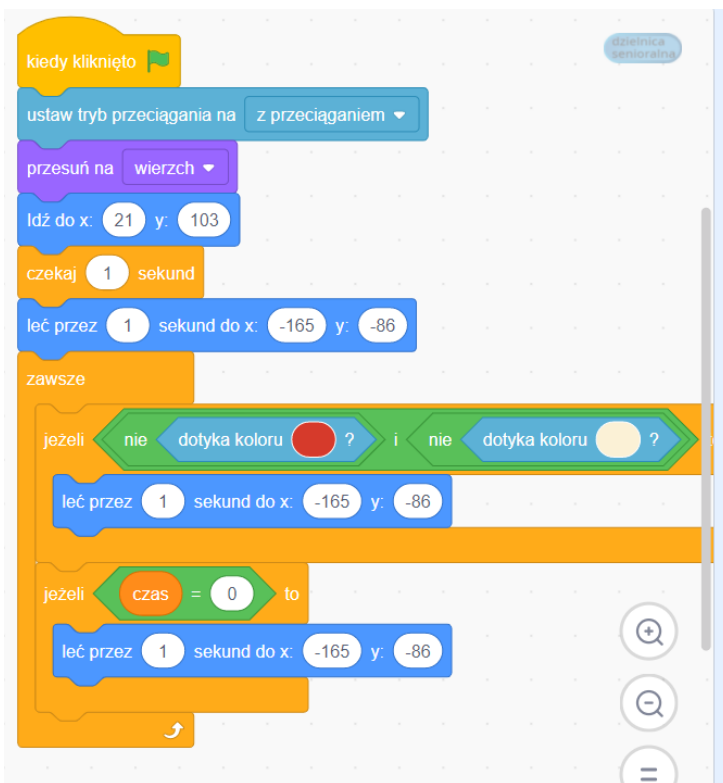
Klikamy w pipetkę na dole białego pola i pobieramy dokładny kolor z mapy czy tła:



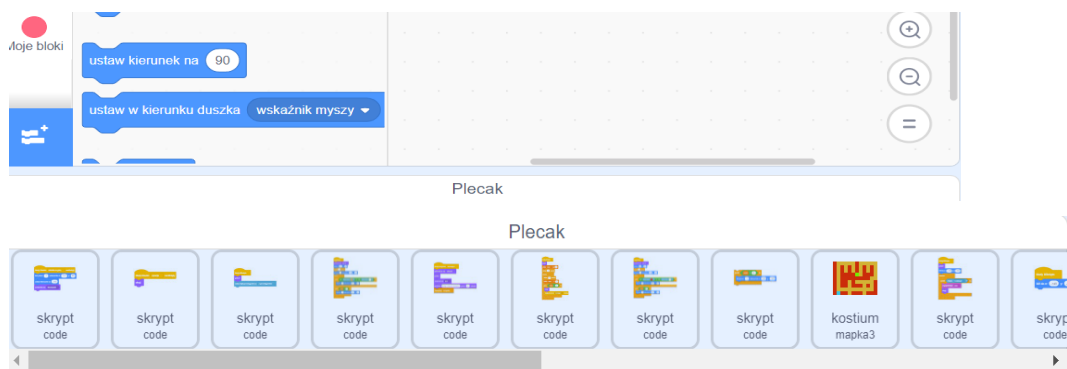


Kolor tła musimy umieścić w skrypcie, bo przyciski dotykają również tła. Muszą więc “uciec” na bok wtedy, kiedy nie są na swojej dzielnicy i nie są na kolorze tła. Problemem jest dzielnica “Lenno pod kontrolą Princepsa”, ze względu na wcięcie w mapie - zdarza się, że inna nazwa tam zostanie. Tu pomogą nam pozostałe duszki, które nie pozostaną na złej dzielnicy, więc i Lenno w końcu znajdzie się na właściwym miejscu, jeżeli gracz zmieści się w czasie.

Pozostaje jeszcze pytanie o miejsce pętli “zawsze” i warunku “jeżeli-to”. Przycisk musi odlecieć **zawsze**, **jeżeli** jest w złym miejscu i **jeżeli** czas się skończył. Cały skrypt może wyglądać tak:



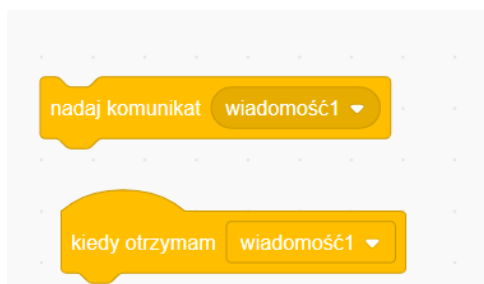
W podobny sposób możesz zaprogramować pozostałe przyciski, a nawet schować sobie cały ten skrypt do Plecak w Scratchu i wyjąć go przy następnym przycisku. Pamiętaj tylko o różnicach w położeniu przycisków na mapie, także na bokach sceny i różnych kolorach dzielnic do czujnika. A oto Plecak:



Kiedy wkładasz skrypty do plecaka, uważaj. Plecak powinien zmienić kolor na niebieski, żeby skrypt w nim został. W przeciwnym razie możesz przypadkowo usunąć gotowy skrypt. Wprawdzie nie musisz kopiować skryptu, bo skrypt schowany do Plecaka od razu jest kopią, ale na początek lepiej skopiuj cały skrypt i tę kopię schowaj w Plecaku, żeby nie stracić swojej pracy, zanim się nie nauczysz, jak działa Plecak.

Krok 5:

**Zaprogramuj duszka Rycerza:** rycerz jest osobą kończąca grę. Nie powinno go być widać na początku gry, więc trzeba go ukryć, gdy klikniemy w zieloną flagę. Rycerz powinien też zakończyć grę w momencie, kiedy czas się skończy. Pamiętaj, że mieliśmy jeszcze wrócić do skryptu mapy? Tam jest osadzony licznik czasu i mapa właśnie powinna uruchomić rycerza. Wykorzystamy kolejny sposób komunikacji pomiędzy duszkami - wiadomość z działu Zdarzenia:

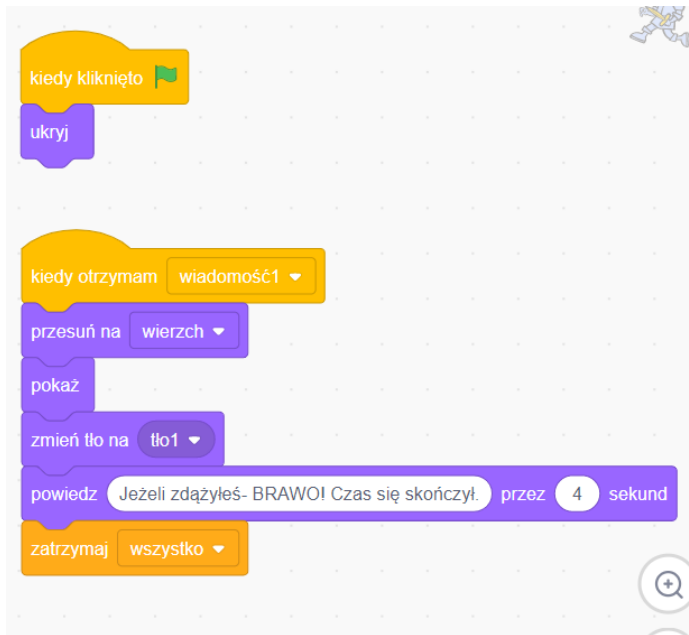


Mapa musi nadać wiadomość, kiedy skończy się czas, a rycerz zacząć działać, kiedy otrzyma tę wiadomość.

Co robi rycerz?

1. Najpierw go nie widać.
2. Pokazuje się nad tłem, czyli na wierzchu.
3. Chowa mapę.
4. Mówi o zakończeniu gry.
5. Zatrzymuje wszystko.

Jak w dwóch skryptach zaprogramować te działania? Gotowe skrypty mogą wyglądać tak:



### Uwagi/alternatywy:

**Minimum:** lekcja o Polsce w okresie rozbitcia dzielnicowego, dalsza część historii Tymka, wykonanie gry "Rozbitcie dzielnicowe - władcy i ich dzielnice".

### Dodatki do wyboru lub wszystkie:

- karty jako nagroda dla uczniów (władcy dzielnicowi);
- kolejnym elementem generatora Tymka jest **chorągiew z krzyżem Konrada Mazowieckiego**;
- kolejny punkt na dużej papierowej mapie - **Płock**, siedziba Konrada Mazowieckiego.

# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 5:

# Murujemy nowy dom z Kazimierzem Wielkim

Lekcja, podczas której uczniowie dowiadują się, jak Kazimierz Wielki odrodził i umocnił Polskę. Tymek, dzięki pomocy uczniów, mógł dotrzeć do Krakowa i dać królowi mądrą radę. Zostanie też zorganizowana parada mBotów z okazji zjazdu monarchów Europy w Krakowie.

### Cele zajęć:

Uczeń powinien:

- opowiadać o problemach Polski za panowania Kazimierza Wielkiego,
- przedstawiać działania Kazimierza Wielkiego na polu gospodarczym, dyplomatycznym, obronnym, prawodawczym,
- przedstawiać przyczyny powołania Akademii Krakowskiej, znać datę jej powstania.

### NaCoBeZu

Po lekcji:

- opowiadasz o problemach Polski za panowania Kazimierza Wielkiego,
- przedstawiasz działania Kazimierza Wielkiego w dziedzinie gospodarki, obronności, prawodawstwa,
- przedstawiasz przyczyny powołania Akademii Krakowskiej, znasz datę jej powstania.

### Materiały pomocnicze:

- podręcznik do historii klasa 5,
- portret Kazimierza Wielkiego autorstwa Jana Matejki,
- materiały dotyczące kodowania dostępne na końcu scenariusza,
- pdf z kartami - nagrodami (polscy naukowcy średniowieczni i renesansowi),
- mBoty2 i komputery lub laptopy, telefony/tablety z aplikacjami do sterowania robotem,
- film instruktażowy: [Meet mBot2](#)
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ Ruś Halicka → kupcy i rzemieślnicy → grosz krakowski → cło → żupy solne → statuty Kazimierza Wielkiego → Akademia Krakowska → uczta u Wierzyńka → dyplomacja → urzędniccy królewscy

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka,
- algorytmika w programie wizualnym, w aplikacji mBlock lub Makeblock.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach z robotem.

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia VII. Polska w XIV–XV wieku.

Uczeń:

1. opisuje rozwój terytorialny państwa polskiego w XIV i XV wieku;
2. analizuje dokonania Kazimierza Wielkiego w dziedzinie polityki wewnętrznej (system obronny, urbanizacja kraju, prawo, nauka) oraz w polityce zagranicznej.

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

- I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
  1. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
  - 2) sterowanie robotem lub obiektem na ekranie;
  3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
  1. projektuje, tworzy i zapisuje w wizualnym języku programowania:
    - 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
    - 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Nauczyciel informuje uczniów, że dzięki ich pomocy Tymek wylądował w Krakowie (o którym mowa była przy legendzie o Hejnale Mariackim) u króla Kazimierza Wielkiego. Prezentuje jego portret z kanonu Jana Matejki. Tymek spotkał się z królem i opowiedział mu swoją historię, król zachwycony opowieścią o szkole zapalał chęcią powołania takiej u siebie. Król zamierza, by Polska była przygotowana do starć z Krzyżakami.

### 2. Część zasadnicza

czas: ok. 30 minut.

Nauczyciel wyjaśnia temat lekcji i przedstawia sytuację kraju. Dzieci wraz z nauczycielem szukają płaszczyzn, jakie powodują, że dany kraj jest silny (finanse, nauka, obronność, wojsko, drogi). Tworzą z nauczycielem poradnik, co zrobić, by Polska była krajem przygotowanym do walki z Krzyżakami w przyszłości. Nauczyciel prezentuje największych studentów i wykładowców Akademii Krakowskiej. Pokazuje, jakie działania podjął król Kazimierz, opowiada o różnych jego reformach i o sukcesach dyplomatycznych (np. zjeździe monarchów europejskich i ucztę u Wierzyńka).

#### Programowanie:

Uczniowie mają tym razem za zadanie zaprogramowanie mBotów na paradę z okazji zjazdu władców Europy w Krakowie w roku 1364 oraz ucztę u Wierzyńka. Uczniowie podzieleni na grupy wymyślają, jak będzie jeździł, świecił i grał ich mBot oraz jak go ozdobią, żeby był efektowny.

Krok 1:

Potrzebujesz programu na komputer lub aplikacji na telefon/tablet do sterowania robotem:

- program lub aplikacja mBlock:  
<https://mblock.makeblock.com/en-us/download/>
- aplikacja webowa Makeblock:  
<https://ide.mblock.cc/>
- aplikacja mBlock:  
<https://play.google.com/store/apps/details?id=com.makeblock.mblock&hl=pl&gl=US>
- aplikacja Makeblock <https://play.google.com/store/search?q=makeblock&c=apps&gl=PL>

Krok 2:

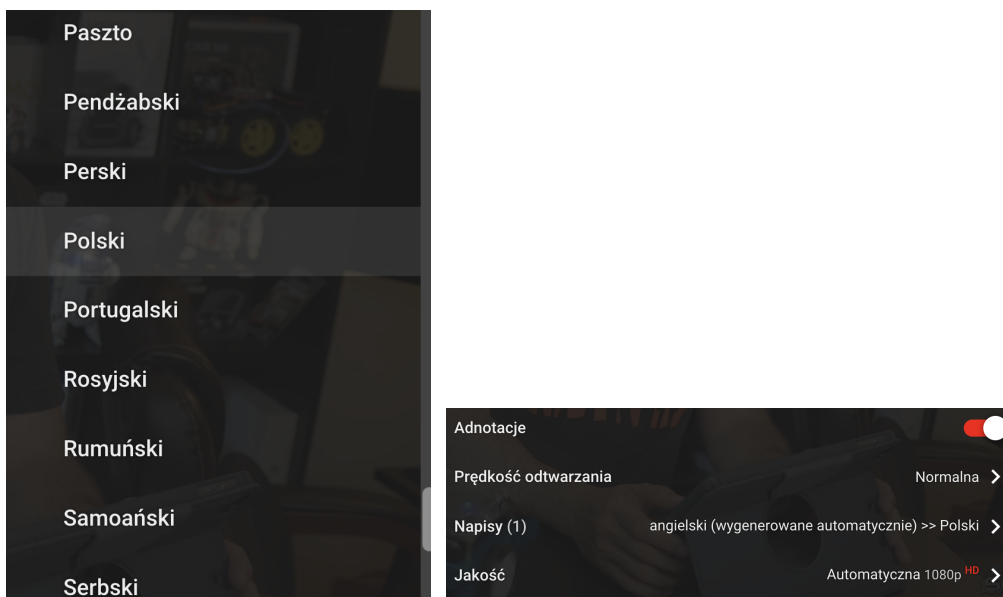
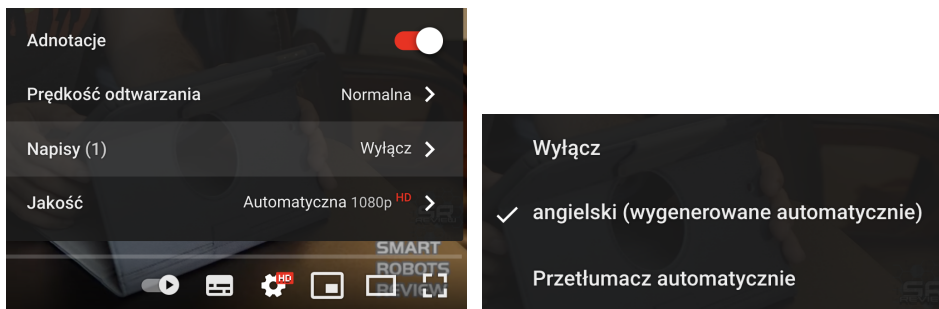
**Zaplanuj i zaprogramuj, co ma robić robot.** Z lekcji o mBotach (lekcja dodatkowa) wiesz już sporo o możliwościach tego robocika. Wykorzystaj je do tworzenia pomysłowych sekwencji ruchu, światła, dźwięku itd. Zainspirować może Cię też film:

 Meet mBot2 oraz zabawy z aplikacjami- mBlock i Makeblock.

Jeżeli nie znasz języka angielskiego, możesz dla filmu ustawić polskie napisy, wykorzystując ikonę:



Następnie przejdź kolejne elementy menu, aby wybrać język polski napisów:



A oto kilka podpowiedzi:

1. Zapoznaj się z możliwościami mBota2, zapisanymi w mikroprocesorze CyberPi. Są tam zawarte między innymi:

- program reagujący kolorami na Twój głos
- dyktafon- który zarejestruje Twój głos
- program zmieniający światła robota
- gra Wesołe liczenie
- trzy programy demonstracyjne dla mBota2- jeżdżenie wzdłuż czarnej linii, omijanie przeszkód i programowanie ruchu robota przy pomocy joysicka

Szczegółowe instrukcje obsługi CyberPi znajdziecie tu:

<https://robotyedukacyjne.pl/wp-content/uploads/2022/07/mBot2-instrukcja-u%C5%BCytkownika.pdf>

Instrukcje działania w każdym z programów zawartych w CyberPi pojawiają się na jego ekranie, a w Ustawieniach można zmienić język na polski.



2. Najwięcej możliwości daje Ci jednak programowanie robota na komputerze, możesz bowiem posłużyć się komunikacją pomiędzy robotem a komputerem w programie mBlock. Komputer może nadawać komunikaty do robota, a robot może mówić, śpiewać, grać, świecić, poruszać się, odpowiadać na pytania itd. Oto przykładowe skrypty:

```
po kliknięciu
  odtwórz cześć
  czekaj 1 sekund
  ustaw głośność na 100 %
  zagraj nutę 60 przez 1 taktów
  zagraj nutę 60 przez 0.15 taktów
  zagraj nutę 60 przez 0.15 taktów
  zagraj nutę 60 przez 0.15 taktów
  zagraj nutę 64 przez 1 taktów
  zagraj nutę 64 przez 0.15 taktów
  zagraj nutę 64 przez 0.15 taktów
  zagraj nutę 64 przez 0.15 taktów
  zagraj nutę 62 przez 1 taktów
  zagraj nutę Otwórz hi-hat przez 1 takt
  zagraj nutę talerz przez 1 takt
  zagraj nutę Zamknij hi-hat przez 1 takt
```

Robot będzie mówił i grał.

A tu skrypt z ruchem, zmianą świateł LED i dźwiękiem:

```
Bloki Py
po kliknięciu
  wyświetl [red, orange, yellow, green, blue]
  jedź naprzód prędkość 50 obr/min przez 5 s
  powtórz 4
    silnik oba koła obróć o 180 °
  odtwórz animację LED Rainbow i czekaj
  LED wszystkie wyświetl R 0 G 255 B 0 przez 1 sekund
  odtwórz wow i czekaj
```

Ciekawe możliwości znajdziecie też w aplikacji na telefon Makeblock

<https://play.google.com/store/search?q=makeblock&c=apps&gl=PL>

Po połączeniu robota z aplikacją możecie na żywo sterować ruchem i dźwiękiem robota oraz światłami.

Natomiast w innej aplikacji na telefon- mBlock można zaprogramować robota tak, jak na komputerze

<https://play.google.com/store/apps/details?id=com.makeblock.mblock&hl=pl&gl=US>



Krok 3:

Możesz ozdobić jakoś mBota - w końcu to uczta u Wierzyńka z okazji zjazdu władców europejskich!

### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

Nauczyciel sprawdza realizację celów w oparciu o NaCoBeZu.

Zakończeniem lekcji jest parada mBotów zaprogramowanych przez poszczególne grupy.

**Dodatki do wyboru lub wszystkie:**

- **kolejnymi kartami** do zdobycia są karty z polskimi naukowcami z epoki średniowiecza i renesansu;
- **astrolabium** z karty z Kopernikiem będzie zagubionym elementem generatora czasu wehikułu Tymka.

**Uwagi/alternatywy:**

**Minimum:** lekcja o panowaniu Kazimierza Wielkiego; kolejna część przygód Tymka; zorganizowanie parady mBotów.

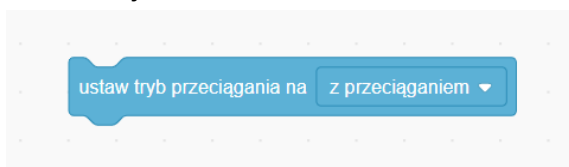
**Dodatki do wyboru lub wszystkie:**

- karty jako nagrody dla uczniów za wykonanie zadania - polscy naukowcy średniowieczni i renesansowi;
- elementem generatora czasu Tymka będzie **astrolabium Mikołaja Kopernika**,
- kolejnym miejscem na dużej mapie papierowej będzie **Kraków**. Uczniowie będą też mogli wykonać animację - miasto średniowieczne do samodzielnego zaplanowania:

<https://scratch.mit.edu/projects/546308691/>.

Wykonanie tej animacji jest dość proste, natomiast każdy z Was może zaplanować w niej swoje własne miasto. Wszystkie obiekty na scenie to duszki - możecie je ściągnąć z naszej animacji, możecie też sami narysować swoje obiekty w edytorze Scratcha (raczej nie pobrać z Internetu, bo obiekty muszą być pozbawione tła).

Kiedy już będziecie mieli duszki - budynki, musicie zdecydować, ile ich będzie (każdy możecie zdublować, klikając w nie prawym klawiszem). Zastanówcie się, jak będą się pojawiały na scenie. Sprawdźcie, jak my to zaplanowaliśmy i wymyślcie swój sposób lub zróbcie tak, jak my. Teraz pozostaje Wam jeszcze narysować albo znaleźć w bibliotece pasujące tło. I jeszcze jedna ważna rzecz - abyście mogli przesuwać każdy obiekt na scenie podczas prezentacji animacji, musicie każdego duszka wyposażyć w bloczek z działu czujniki:



Używaliśmy go już podczas tworzenia mapy władców dzielnicowych w lekcji 4 "Rozbicie dzielnicowe". W normalnej sytuacji, podczas grania w grę czy prezentacji animacji, nie da się poruszyć elementów na scenie przy pomocy myszki. Ten bloczek umożliwi Wam przestawianie obiektów w Waszym mieście. A oto przykładowe skrypty duszków-budynków:



# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 6:

# Jak przeżyć w średniowieczu i nie zwariować?

Lekcja, podczas której nauczyciel wprowadza uczniów w tajniki życia codziennego w średniowieczu - w miastach i na wsiach. Tymek powędruje do Chełma - miasta średniowiecznego, aby poznać jego obiekty i otrzymać od uczniów kolejną kartę z elementem generatora czasu.

### Cele zajęć:

Uczeń powinien:

- potrafić opowiedzieć o wyglądzie, grupach społecznych i zajęciach mieszkańców miasta i wsi,
- rozumieć różne funkcje miast: obronne, polityczne, gospodarcze, kulturalne,
- potrafić omówić proces lokacji miasta,
- potrafić opowiedzieć o życiu codziennym na wsi i zmianach w niej zachodzących pod wpływem nowych metod uprawy i narzędzi,
- rozumieć obowiązki chłopca i pana oraz różnice w ich położeniu.

### NaCoBeZu

Po lekcji:

- potrafisz opowiedzieć o wyglądzie, grupach społecznych i zajęciach mieszkańców miasta,
- rozumiesz różne funkcje miast: obronne, polityczne, gospodarcze, kulturalne,
- potrafisz omówić proces lokacji miasta,
- rozumiesz pojęcia: cech, gildia, mistrz, patrycjusze, plebejusze, pospólstwo,
- potrafisz opowiedzieć o życiu codziennym na wsi i zmianach w niej zachodzących pod wpływem trójpolówki i nowych narzędzi,
- rozumiesz obowiązki chłopca i pana oraz różnice w ich położeniu,
- opisujesz proces lokowania wsi, rozumiesz jego znaczenie, rozumiesz pojęcia: sołtys, zasadzca, wolnizna.

### Materiały pomocnicze:

- podręcznik do historii klasa 5,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (obiekty miasta średniowiecznego),
- strona dotycząca podziemi kredowych w Chełmie <https://podziemiakredowe.com.pl/>,

- animacja średniowiecznego miasta z lekcji o Kazimierzu Wielkim <https://scratch.mit.edu/projects/546308691/>,
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

pańszczyzna → czynsz → sołtys → zasadzca → wolnizna → sołtys → patrycjusze → plebejusze → pospólstwo → cech → gildia → mistrz → lokacja → rada miejska → ratusz → rynek → stan społeczny

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka,
- algorytmika w programie Scratch,
- gra edukacyjna - grywalizacja.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach przy wykonaniu gry.

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia IV. Społeczeństwo i kultura średniowiecznej Europy.

Uczeń:

1. [...] charakteryzuje podziały społeczne w średniowieczu; wyjaśnia pojęcie stanu

2. opisuje warunki życia średniowiecznego miasta i wsi;

3. porównuje kulturę rycerską i kulturę miejską [...].

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

1. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

3) sterowanie robotem lub obiektem na ekranie;

## #SuperKoderzy / Podróżnicy w czasie / Jak przeżyć w średniowieczu i nie zwariować?

3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
1. projektuje, tworzy i zapisuje w wizualnym języku programowania:
    - 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
    - 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Nauczyciel dziękuje uczniom w imieniu Tymka, ich działania przyniosły rozwój edukacji w kraju i wzmocnienie obronności. Skoro mamy już nowe, murowane miasta, możemy przyjrzeć się życiu jednego z nich. Tymoteusz wylądował w Chełmie słynącym ze swych podziemnych piwniczek drążonych w kredowym podłożu. Początki Chełma sięgają wczesnego średniowiecza, od średniowiecza też była tam wydobywana kreda i dziś pozostały po tym magiczne podziemia kredowe <https://podziemiakredowe.com.pl/>

### 2. Część zasadnicza

czas: ok. 15 minut.

Nauczyciel wyjaśnia temat lekcji i przedstawia, jak wyglądało życie w średniowiecznym mieście. Opisuje standardowy układ miasta, opisuje wygląd rynku i układ ulic w miastach lokowanych na prawie chełmińskim i magdeburskim (na prawie polskim i niemieckim). Nauczyciel może się posłużyć animacją z lekcji o Kazimierzu Wielkim, aby uczniowie wymienili elementy miasta średniowiecznego:

<https://scratch.mit.edu/projects/546308691/>.

Opowiada też o mieszkańcach miasta, rzemieślnikach i kupcach, bogatych patrycjuszach i biedniejszych plebejuszach oraz pospółstwie, organizacjach kupieckich i rzemieślniczych, hierarchii rzemieślników. Podobnie opisuje też życie na wsi i jego organizację w średniowieczu.

### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

Nauczyciel przy pomocy punktów NaCoBeZu sprawdza stopień realizacji celów.

#### Programowanie:

Uczniowie wykonują grę- plan średniowiecznego miasta jako labirynt, który musi przejść duszek.

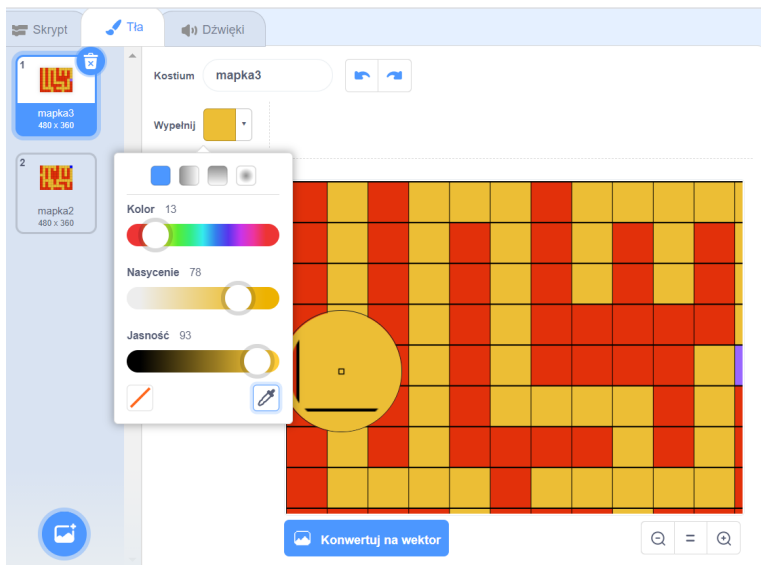
Krok 1:

**Zagraj w grę** i zastanów się czego potrzebujesz, żeby stworzyć taką grę? Co jest tłem, co duszkiem i jakie bloczki są wykorzystane?

<https://scratch.mit.edu/projects/543768190/>

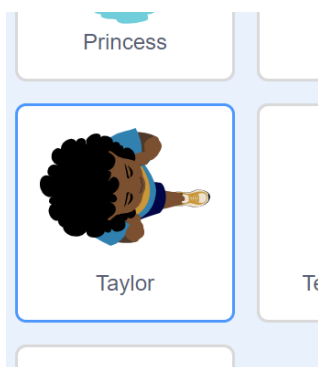
Krok 2:

**Osadź labirynt.** Wyobraź sobie, że widzisz plan średniowiecznego miasta. Ulice są żółte, a domy czerwone. Możesz pobrać mapkę z naszego projektu (są dwie mapki- widać to w kostiumach sceny). Możesz zmodyfikować przebieg labiryntu- korzystając z edytora graficznego sceny (zakładka Tła). Przekształć tło w bitmapę, pobierz dokładnie potrzebny kolor pipetką i zmień kolor wybranej kratki:

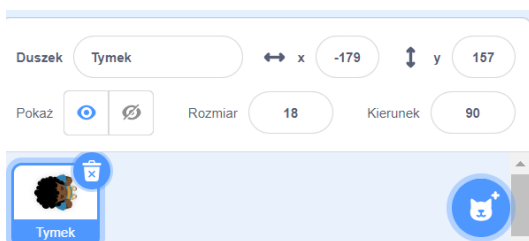


Krok 2:

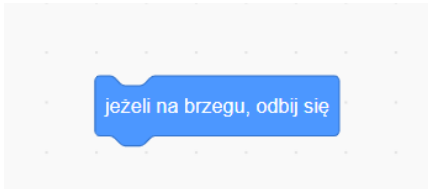
**Zaprogramuj ruch duszka.** Znajdź duszka w Bibliotece Duszków. My wybraliśmy postać chłopca widzianą z góry, aby łatwiej było sterować duszkiem.



Ustaw rozmiar duszka tak, aby mieścił się w żółtej kratce całkowicie.



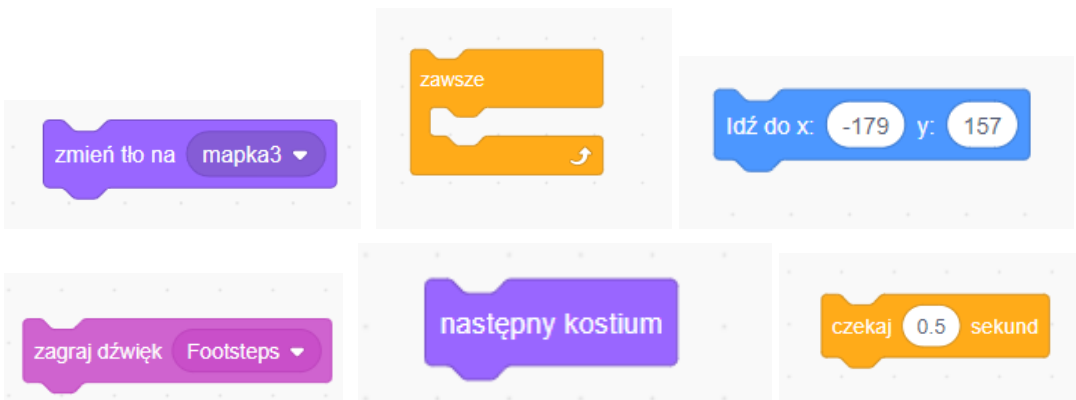
Jak zaprogramować ruchy duszka? Nie może biegać zbyt szybko, więc powinien posuwać się nie więcej niż 10 kroków na raz. Dla kontroli - połącz ruch z jedną ze strzałek na klawiaturze, np. strzałką w górę. Duszek będzie też musiał zmieniać kierunek ruchu, więc zaprogramuj mu przynajmniej jeden sposób obrotu o 90 stopni (tak naprawdę jeden wystarczy, bo można go powtarzać) i połącz z inną strzałką, np. w prawo. Przyda się też taki bloczek (sprawdź dlaczego):



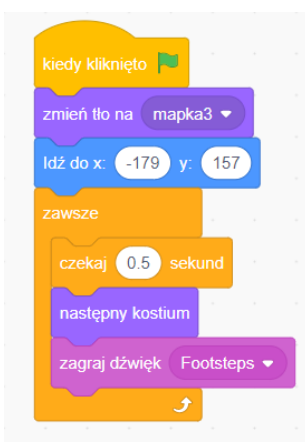
Teraz duszek może chodzić po planszy i obracać się. Skrypt może wyglądać tak:



Aby duszek mógł posłużyć w grze, musi zaczynać w wybranym przez nas miejscu: pierwszej żółtej kratce na górze, z lewej strony. Ponieważ są dwie mapki - tła, bo gra jest dwuczęściowa, na początku powinna się pokazać pierwsza mapka; poza tym duszek ma wiele kostiumów, bo imituje chodzenie. Przydałyby się jeszcze odgłosy kroków z biblioteki dźwięków. Co więc podłączysz do zielonej flagi w kolejnym skrypcie?



Połączenie tych bloczków z zieloną flagą pozwoli nam wykorzystać duszka do naszej gry. Pamiętaj, że kostium nie może zmieniać się bez chwili przerwy i że chodzenie duszka musi działać w pętli. Przykładowy skrypt może wyglądać tak:

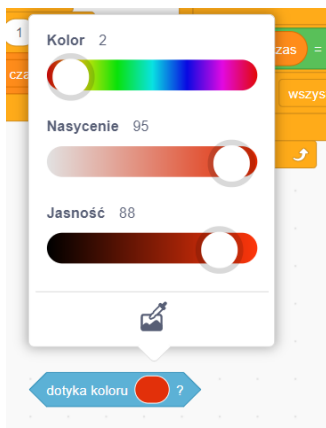


Krok 3:

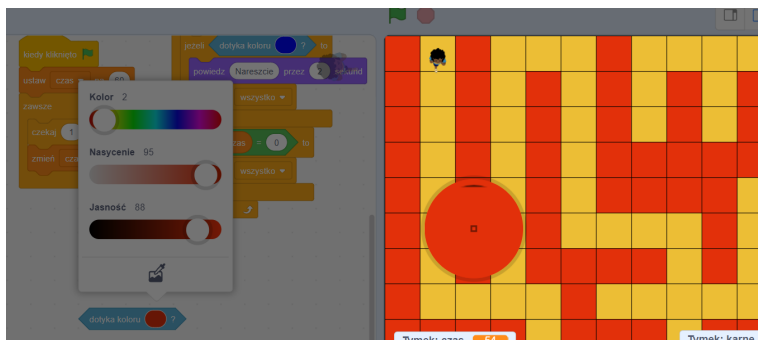
### Zaprogramuj ograniczenia dla duszka.

1. Duszek nie powinien mieć możliwości zejścia z ulicy.
2. Można by dołączyć jakiś alarm, kiedy dotknie czerwonej ściany domu.
3. Kiedy znajdzie się na końcu pierwszego labiryntu, na fioletowej kratce, powinien przenieść się na drugą planszę.
4. Kiedy skończy wędrówkę na niebieskiej kratce drugiej planszy, powinien zareagować i skończyć grę.

To wszystko wymaga uruchomienia czujników koloru i warunku "jeżeli-to" oraz pętli "zawsze":

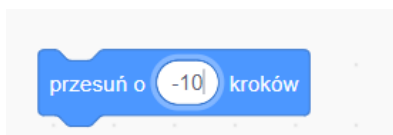


Pamiętaj o pobieraniu koloru pipetką z planszy - komputer reaguje na precyzyjne komendy i lepiej od nas rozróżnia kolory.



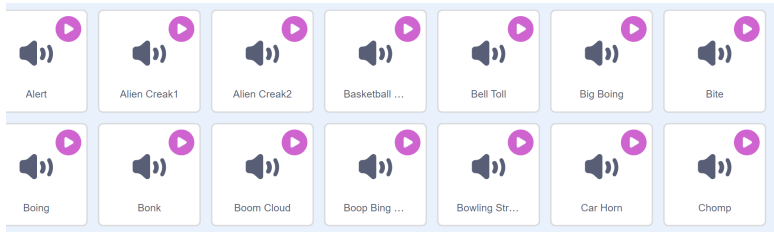
Podobnie możesz pobrać kolor fioletowy i niebieski.

Zastanów się teraz, jak zaprogramować skrypt, dzięki któremu duszek nie wejdzie na czerwone pola. Czujnik to rozpozna, a bloczek ruchu powinien delikatnie odrzucić duszka ze złej drogi (tak, żeby nie wpadł na czerwone pole z drugiej strony). Pomoże nam bloczek ruchu:

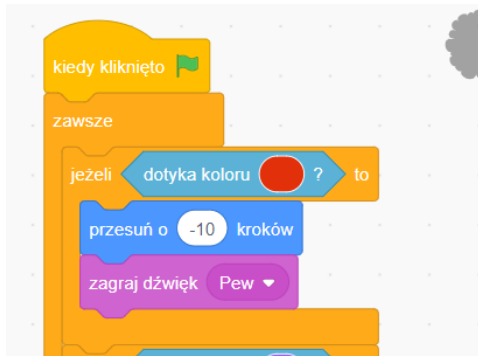


Można też wybrać ciekawy dźwięk z biblioteki dźwięków, który zagra, kiedy duszek zapędzi się na czerwone pole:





Ten fragment skryptu może wyglądać tak:



W podobny sposób zaprogramuj przeniesienie duszka na kolejną planszę i w konkretne jej miejsce - kiedy duszek dotrze do fioletowej kratki na pierwszej planszy oraz reakcję duszka na zakończenie wędrówki i zatrzymanie gry.

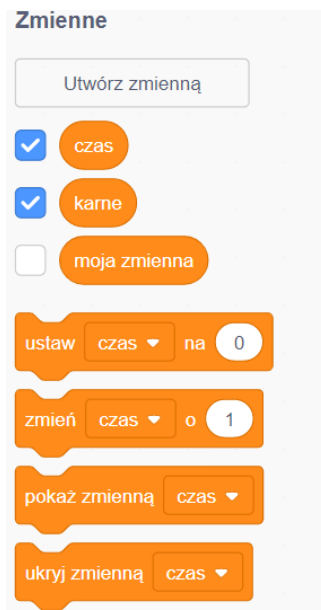
Krok 4:

**Zaprogramuj grę w przejście labiryntu.** Na razie zrobiliśmy zabawę - animację labiryntu. Ale to jeszcze nie jest gra. Aby nasz labirynt stał się grą, musimy wprowadzić elementy rywalizacji - czas i punkty.

Potrzebujesz do tego zmiennych. Co robią te zmienne?

1. Ograniczają czas przejścia obu plansz labiryntu do 60 sekund (można go wydłużyć lub skrócić).
2. Przyznają punkty karne z złamanie reguły nie wchodzenia na czerwone pola.

Jak to zaprogramować? Utwórz zmienne "czas" i "karne" w dziale Zmienne:



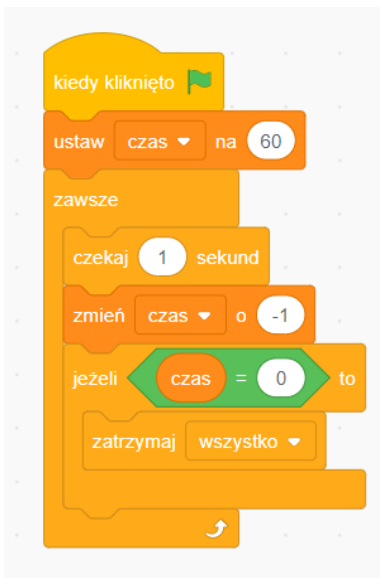
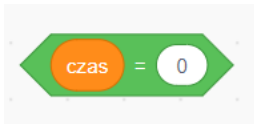
## #SuperKoderzy / Podróżnicy w czasie / Jak przeżyć w średniowieczu i nie zwariować?

Zmienną "karne" możemy wstawić w skrypt z czujnikami - po kliknięciu w zieloną flagę punkty karne powinny być ustawiane na zero, a po każdym dotknięciu czerwonego koloru powinien się naliczać 1 punkt karne.

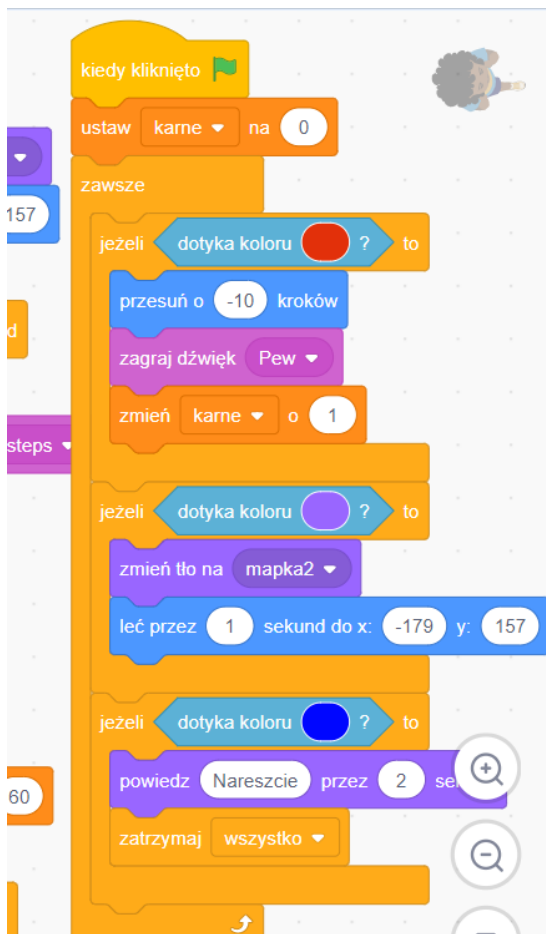
Dla licznika czasu napisz nowy skrypt, znów od zielonej flagi - na początku ustaw czas na 60 i potem co sekundę odejmuj od czasu jedną jednostkę, oczywiście w pętli "zawsze".

Musisz jeszcze sprawić, żeby gra stawała, gdy czas się skończy:

Pamiętaj, że warunek "jeżeli czas=0 to zatrzymaj wszystko" powinien się znaleźć w pętli "zawsze". Cały skrypt czasu może wyglądać tak:



A oto skrypt z czujnikami:



I jeszcze jedna rada - zmienne na scenie mogą zasłaniać planszę, ale możesz je przenieść w dowolne miejsce, łapiąc i przesuując myszką.

### Uwagi/alternatywy:

**Minimum:** lekcja o średniowiecznym mieście i wsi; kolejny etap przygód Tymka; wykonanie gry-labiryntu, przez który musi przejść duszek.

### Dodatki do wyboru lub wszystkie:

- karty jako nagroda dla uczniów za wykonanie zadania - obiekty miasta średniowiecznego;
- artefaktem z generatora czasu Tymka jest wieża obronna;
- kolejnym miejscem na dużej mapie papierowej jest **Chełm**.

Labirynt może być prostą planszą do przejścia, ale też mogą tam czyhać potwory i przeszkody, a podczas przejścia trzeba zbierać jakieś obiekty. Uczniowie mogą zaprogramować na grze labirynt dodatkowe elementy. Uwaga - gra została zaprogramowana w Scratchu2, więc niektórych elementów nie ma w Scratchu3, ale można je pobrać z gry: <https://scratch.mit.edu/projects/12752901/>.

# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 7:

# Polak, Węgier i... Litwin - trzy bratanki

Lekcja, podczas której nauczyciel opowiada o kłopotach Polski i Węgier z następcą tronu oraz o sposobie, w jaki Polska sobie z tym poradziła. Tymoteusz po powrocie z Chełma w Krakowie poznaje młodą dziewczynę, która z siekierą rzuca się na drzwi zamku, by z niego uciec. Ta dziewczyna zostanie Królem Polski. Poznajemy sylwetkę Jadwigi, dowiadujemy się o aranżacji małżeństw i przebiegu takowych.

### Cele zajęć:

Uczeń powinien:

- wymieniać państwa, z którymi Polska zawarła unię, przedstawiać przyczyny i skutki unii,
- rozumieć znaczenie przywileju koszyckiego,
- znać dokonania Ludwika Węgierskiego, Jadwigi, Władysława Jagiełły.

### NaCoBeZu

Po lekcji:

- wymieniasz państwa, z którymi Polska zawarła unię, przedstawiasz przyczyny i skutki unii,
- rozumiesz znaczenie przywileju koszyckiego, wiesz, kiedy i przez kogo został nadany,
- znasz dokonania Ludwika Węgierskiego, Jadwigi, Władysława Jagiełły.

### Materiały pomocnicze:

- podręcznik do historii klasa 5,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (elementy stroju średniowiecznego),
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

unia personalna → unia w Krewie → dynastia → unia → przywilej → uniwersytet → chrystianizacja → Wielkie Księstwo Litewskie → Krzyżacy → przywilej koszycki

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,

- pogadanka,
- algorytmika w programie Scratch,
- gra edukacyjna - grywalizacja.

### Formy pracy

- praca z całą klasą,
- grupowa - praca w grupach przy tworzeniu gry.

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia

VII. Polska w XIV–XV wieku.

Uczeń:

- 3) opisuje związki Polski z Węgrami w XIV i XV wieku;
- 4) wyjaśnia przyczyny i ocenia następstwa unii Polski z Wielkim Księstwem Litewskim;
- 5) charakteryzuje dokonania w dziedzinie polityki wewnętrznej i zagranicznej Jagiellonów w XV wieku;

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:
- 3) sterowanie robotem lub obiektem na ekranie;
3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:
- 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
- 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

czas: ok. 10 minut

Nauczyciel dziękuje za pomoc w naprawieniu pojazdu Tymka, dzięki niemu można ruszać dalej. Tym razem Tymek prosi o pomoc w dotarciu do miejsca, gdzie odbędzie się spotkanie z przyszłym królem Jagiełłą, który podczas chrztu przybierze imię Władysław.

Opowiada, jak wyglądały dawniej zrękowiny (w przypadku Jadwigi jeszcze w dzieciństwie) i śluby.

### 2. Część zasadnicza

czas: ok. 15 minut.

Nauczyciel wyjaśnia temat lekcji i przedstawia prośbę Jadwigi przekazaną Tymkowi (Jadwiga ma 12 lat i Tymkowi świetnie się z nią rozmawiało). Ma wyjść za męża, ale nie wie, kim jest Jagiełło. Jak na mężczyznę w tamtym czasie jest wyjątkowy, kąpie się codziennie.

Wyruszamy zatem do łaźni w Wilnie, gdzie spotkamy przyszłego króla.

Nauczyciel opisuje historię zachowań higienicznych w okresie średniowiecza. Nauczyciel opisuje postanowienia przywileju koszyckiego. Proponuje poszukanie plusów i minusów unii personalnej Polski i Litwy. Opowiada też o dalszych losach młodej pary i ich zasługach dla kraju.

### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

Nauczyciel nawiązuje do NaCoBeZu i sprawdza stopień realizacji celów lekcji.

Karty - stroje średniowieczne.

Artefaktem jest koszula św. Ludwika.

Podczas podsumowania dzieci bawią się grą - wyścigami o rękę króla Jadwigi:

<https://scratch.mit.edu/projects/540885794/>

#### Programowanie:

Uczniowie w parach wykonują tę grę. Wykonanie jej gwarantuje otrzymanie kolejnej części do wehikułu czasu - czyli karty ze strojami średniowiecznymi (koszula św. Ludwika).

Krok 1:

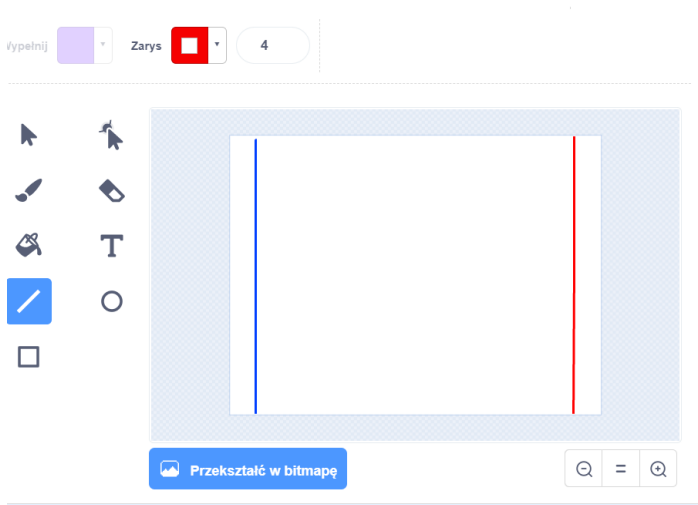
**Po grze podyskutujcie w parach** nad jej konstrukcją. Dlaczego jeden rycerz wygrywa, a drugi przegrywa - i nie jest zawsze tak samo? Jak to się dzieje, że liczony jest czas? Skąd na scenie wzięły się linie startu i mety?

Krok 2:

**Wybierz duszki do wyścigów.** Być może wiesz, że Jadwiga była od dzieciństwa zaręczona z Wilhelmem Habsburgiem, ale później zdecydowano o zerwaniu zaręczyn i oddaniu jej ręki dużo starszemu księciu Litwy - Jagielle. Wyobraź sobie, że obaj kandydaci stanęli do zawodów. My wybraliśmy z Biblioteki Duszków- rycerza. Niestety nie ma on kostiumów i nie może imitować biegania. Za to pasuje do epoki. Oba duszki są takie same, dla odróżnienia zmieniliśmy im kolor piór na szyszaku (zakładka Kostiumy i edytor graficzny).

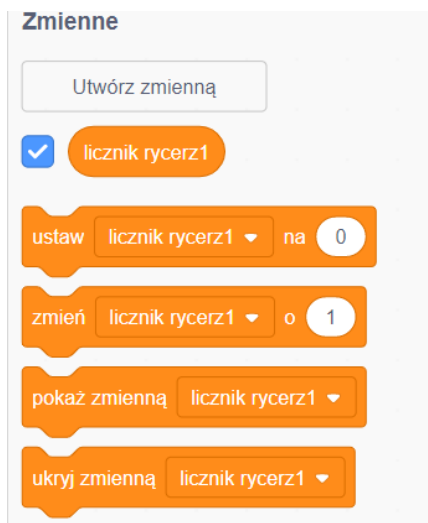
Krok 3:

**Przygotuj scenę.** Jak widzisz na scenie są dwie linie: startu i mety. Trzeba je po prostu narysować w edytorze graficznym sceny - zakładka Tła. Wybierz linię prostą. Liczba obok koloru zarysu to grubość tej linii. Linie muszą być równoległe do prawego i lewego brzegu sceny:



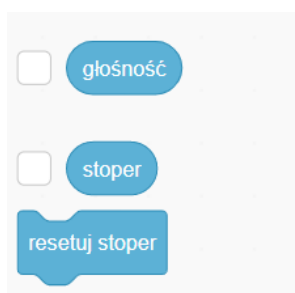
Krok 4:

**Stwórz liczniki obu rycerzy.** To oczywiście zmienne. Każdy rycerz musi mieć swój licznik, widoczny dla jednego duszka.

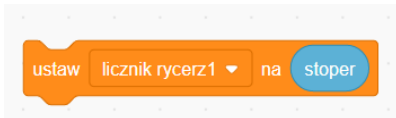


Krok 5:

**Zamień licznik w stoper.** Przedtem licznik liczył nam czas, bo programowaliśmy zmianę co sekundę. Teraz potrzebujemy dokładnego licznika, który działa jak stoper. Znajdziesz potrzebny bloczek w czujnikach:

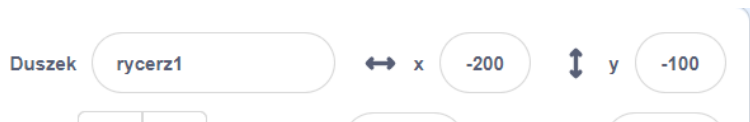


Teraz wystarczy wstawić stoper w licznik:



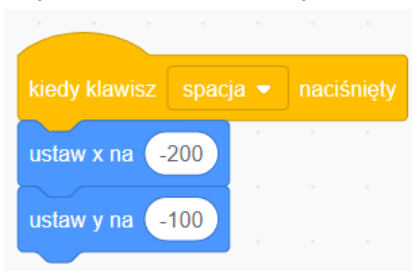
Krok 6:

**Zaprogramuj ruch duszka.** Co zauważasz w ruchu duszka? Biegnie on zawsze po tej samej linii, czyli z punktu widzenia komputera zmienia mu się tylko jeden parametr położenia - zmienna "x", a "y" duszka pozostaje taki sam. Ustaw duszka tam, gdzie ma zaczynać bieg i sprawdź, jaki jest jego parametr x. Oba duszki muszą startować z tego samego pod względem x miejsca. U nas jest to  $x = -200$

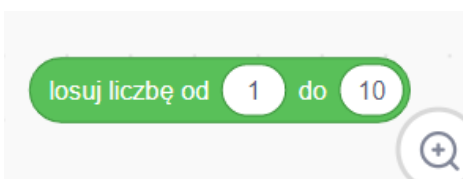


Natomiast y jednego duszka jest 100, a drugiego -100. Oczywiście na początku biegu każdy zawodnik musi się znaleźć na swoim miejscu: rycerz 1 na  $x = -200$ ;  $y = -100$ , a rycerz 2 na  $x = -200$ ;  $y = 100$ .

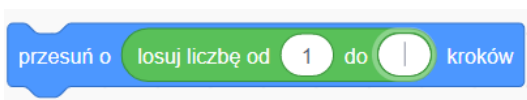
Aby rozdzielić różne czynności duszka, jego ustawienie na starcie podłączyliśmy do kliknięcia spacji.



Na pewno zauważyłaś/eś, że duszki biegają raz szybciej, raz wolniej i nigdy nie wiesz, który wygra. Oznacza to, że komputer wykonuje losowanie. Przyda nam się bloczek z wyrażen:



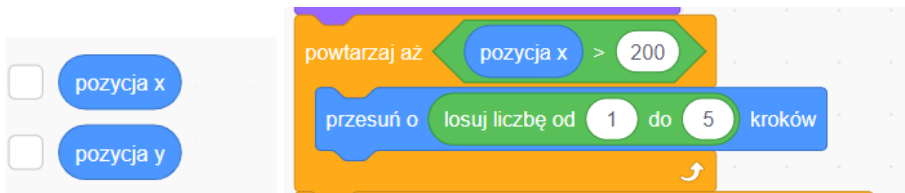
który połączymy z bloczkiem z ruchu:



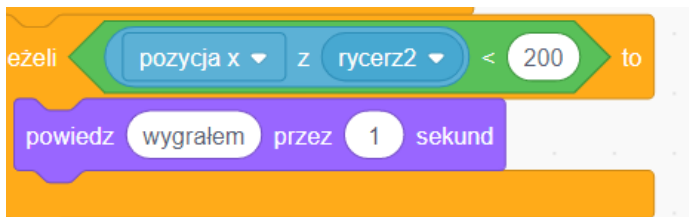
Wypróbuj, do jakiej liczby duszek ma losować ilość kroków, żeby nie biegał za szybko.

Kolejnym pytaniem jest, jak długo duszek ma biec i kiedy oznajmić, że wygrał?

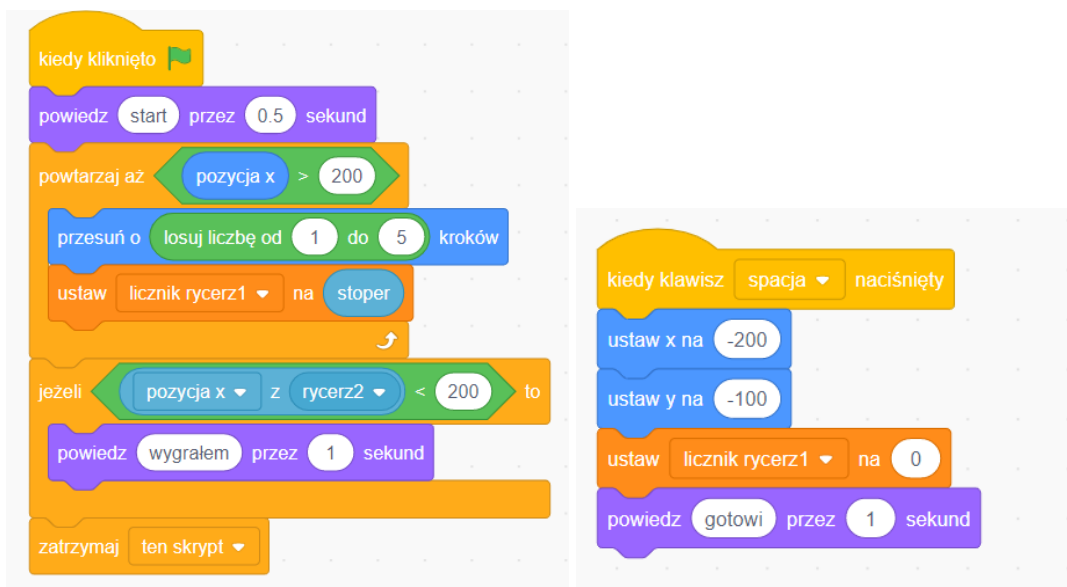
Start znajduje się mniej więcej na linii  $x = -200$ , a meta na linii  $x = 200$  i to określa zakres biegu duszka. Duszek musi biec do mety, czyli powtarzać swój ruch, dopóki się tam nie znajdzie. Ważny jest dla nas tylko parametr x duszka, więc kiedy duszek przejdzie za linię  $x = 200$ , ukończył wyścig. Jak to zapisać?



Pozostaje jeszcze zapisać w skrypcie, jak pokazać zwycięzcę. Jeżeli duszek przekroczył linię  $x=200$ , a jego konkurent jest na  $x < 200$ , to wiadomo, że ten pierwszy wygrał. Powinien o tym powiedzieć, pokaże to także jego licznik. Dla rycerza 1:



Rycerze mogą też wydać komendy przed biegiem i krzyknąć "start". No i oczywiście trzeba na początku wyzerować liczniki. Skrypty mogą wyglądać tak:



Kliknięcie spacji ustawia zawodników i liczniki, zielona flaga rozpoczyna wyścig.

Skrypt drugiego duszka jest identyczny, tylko jego parametry  $y=100$ . Będzie też mówił początek komendy "do biegu". O wygranej rycerza 2 zdecyduje pozycja  $x$  z rycerza 1, odwrotnie niż przy rycerzu 1.

## Uwagi/alternatywy:

**Minimum:** lekcja na temat unii polsko-litewskiej oraz dokonania Jadwigi i Jagiełły; kolejna część przygód Tymka; zaprogramowanie gry-wyścigu kandydatów do ręki Jadwigi.

## Dodatki do wyboru lub wszystkie:

- karty - nagrody dla uczniów za wykonanie zadania - stroje średniowieczne;
- kolejnym elementem generatora czasu Tymka będzie koszula św. Ludwika;
- kolejnym miejscem na papierowej mapie będzie **Krewa** - miejsce podpisania unii polsko-litewskiej.



# Podróżnicy w czasie

**Autorzy:** Alina Idzikowska, Radosław Potrac

## Lekcja 8:

# Ratujemy z Tymkiem victorię grunwaldzką

Lekcja, podczas której Tymek dociera wreszcie do celu, czyli pod Grunwald. Rozegra się ostateczny pojedynek pomiędzy Tymkiem a Jonaszem i mamy nadzieję, że historia nadal będzie wyglądać tak samo - że król Jagiełło wygra pod Grunwaldem, a Tymek będzie mógł wrócić do domu w poczuciu dobrze spełnionego obowiązku.

### Cele zajęć:

Uczeń powinien:

- przedstawiać przyczyny, przebieg i skutki tzw. wielkiej wojny z zakonem.

### NaCoBeZu

Po lekcji:

- przedstawiasz przyczyny, przebieg i skutki tzw. wielkiej wojny z zakonem.

### Materiały pomocnicze:

- podręcznik do historii klasa 5,
- mapa Polski i Litwy w okresie wielkiej wojny z zakonem,
- materiały dotyczące kodowania, dostępne na końcu scenariusza,
- pdf z kartami-nagrodami (elementy wyposażenia rycerza),
- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ Krzyżacy → Żmudź → wielki książę → wielka wojna z zakonem krzyżackim → twierdza → twierdza w Malborku → oblężenie → pierwszy pokój toruński

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- dyskusja,
- pogadanka,
- algorytmika w programie Scratch,
- gra edukacyjna - grywalizacja.

### Formy pracy:

- praca z całą klasą,
- grupowa - praca w grupach przy tworzeniu animacji lub gry.

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, historia VII. Polska w XIV–XV wieku.

Uczeń:

5. charakteryzuje dokonania w dziedzinie polityki wewnętrznej i zagranicznej Jagiellonów w XV wieku;
6. porządkuje i umieszcza w czasie najważniejsze wydarzenia związane z relacjami polsko-krzyżackimi w XIV–XV wieku.

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka: I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. formułuje i zapisuje w postaci algorytmów polecenia składające się na:

- 3) sterowanie robotem lub obiektem na ekranie;
  3. w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

1. projektuje, tworzy i zapisuje w wizualnym języku programowania:
  - 1) pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń,
  - 2) prosty program sterujący robotem lub innym obiektem na ekranie komputera;

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy

**czas: ok. 10 minut**

Nauczyciel mówi, że oto dobiega końca nasza podróż w czasie - jesteśmy pod Grunwaldem. Tymkowi udało się tam dotrzeć dzięki uczniom - mimo uszkodzenia wehikułu czasu (nastąpiła dezintegracja elementów generatora czasu). Krok za krokiem uczniowie pomogli mu uzupełnić brakujące artefakty czasowe - były przekazywane poprzez karty i Tymek mógł skokami przemieszczać się w czasoprzestrzeni, aż dotarł pod Grunwald. Jonas chciał swoimi manipulacjami doprowadzić Jagiełłę do dumnej odmowy przyjęcia dwóch mieczy od Wielkiego Mistrza, a to wywołałoby trwałe zaburzenie czasu i zmieniło wynik bitwy. Na szczęście Tymek ostrzegł króla przed popełnieniem błędu.

### 2. Część zasadnicza

**czas: ok. 30 minut.**

Tutaj stykają się wszystkie elementy układanki. Docieramy do wielkiej wojny. Uczniowie poznają jej przyczyny, skutki, dowiadują się o funkcjonowaniu Zakonu. Poznają przebieg wojny i mądre działania królewskie (most łyżwowy, wielkie łowy i solenie beczek z mięsem). Docieramy wehikułem do Grunwaldu, gdzie zapobiegamy intrydze Jonasa i król Jagiełło może wygrać bitwę. Na karcie z rycerzami średniowiecza odnajdujemy najważniejszą część wehikułu (herb Zawiszy Czarnego, sławnego rycerza spod Grunwaldu) i Tymek może wrócić do swoich czasów.

## Programowanie

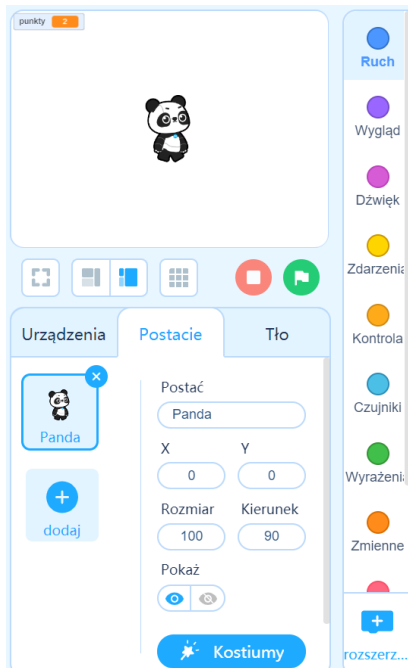
### Turniej Historyczny przy pomocy mBotów:

Podzielcie się na drużyny i przygotujcie powtórzeniowe pytania dla przeciwników. Potrzebny będzie sekundant, który wpisze odpowiedzi przeciwników w program na komputerze lub telefonie. Gdy odpowiedź będzie poprawna - mBot wykona zaplanowane działania, zaś inne, gdy przeciwnicy popełnią błąd.

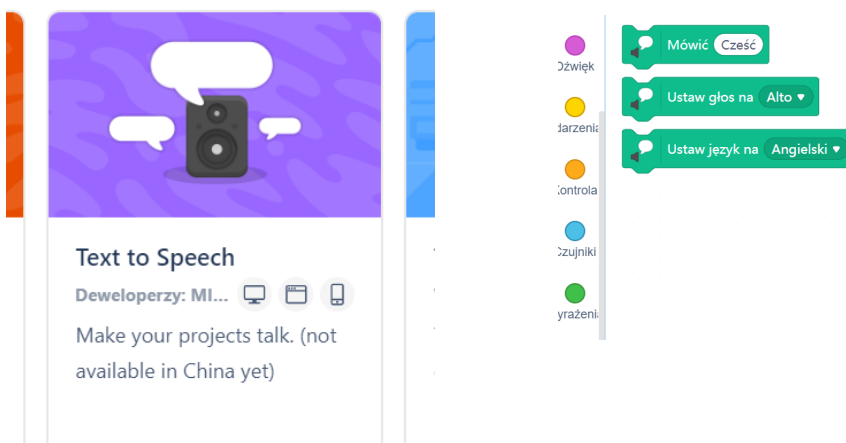
Skorzystamy z programu Make Block i posłużymy się komunikacją pomiędzy duszkiem, podobnym jak w programie Scratch, a mBotem.

Krok 1:

**Zainstaluj rozszerzenie w zakładce Postacie:**



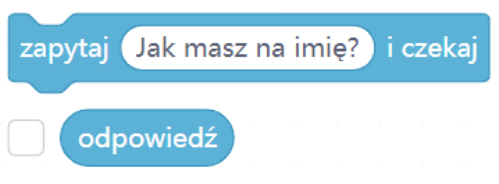
Zainstaluj rozszerzenie "Text to speech". Oto jego bloczki:



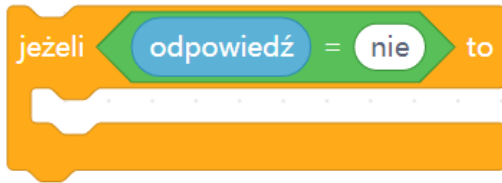
Dzięki niemu program na komputerze przeczyta pytanie, które napiszesz.

Krok 2:

**Zaprogramuj quiz historyczny.** Posłużysz się przy tym bloczkami z działu Czujniki- robiliśmy to już w Scratchu. Bloczek "zapytaj i czekaj" pozwoli Ci zadać pytanie i zaprogramować reakcję programu na dobrą i złą odpowiedź. Potrzebujesz takich bloczków:



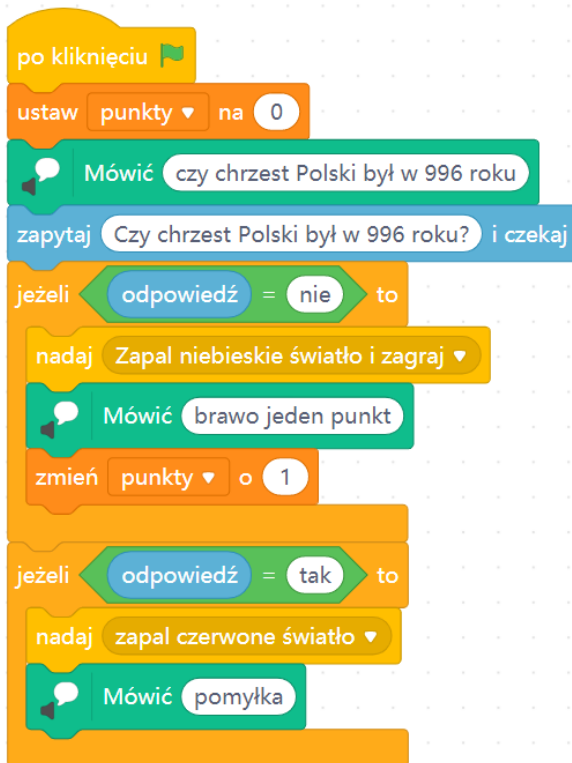
I połączenia odpowiedzi gracza z odpowiedzią prawidłową:



Jaka to ma być reakcja? Jeżeli odpowiedź jest poprawna- mBot powinien zareagować tak, jak to zaplanujesz. Dobrze byłoby udzielić pochwały graczowi i przyznać mu punkt. Żeby móc operować punktami- trzeba stworzyć zmienną "punkty" w dziale Zmienne. Ponieważ mamy rozszerzenie "Tekst na mowę", możemy zaprogramować mówiący quiz. Trzeba też nadać komunikat dla mBota, wykorzystując bloczek "nadaj" z działu Kontrola:



Cały skrypt dla duszka może wyglądać tak:

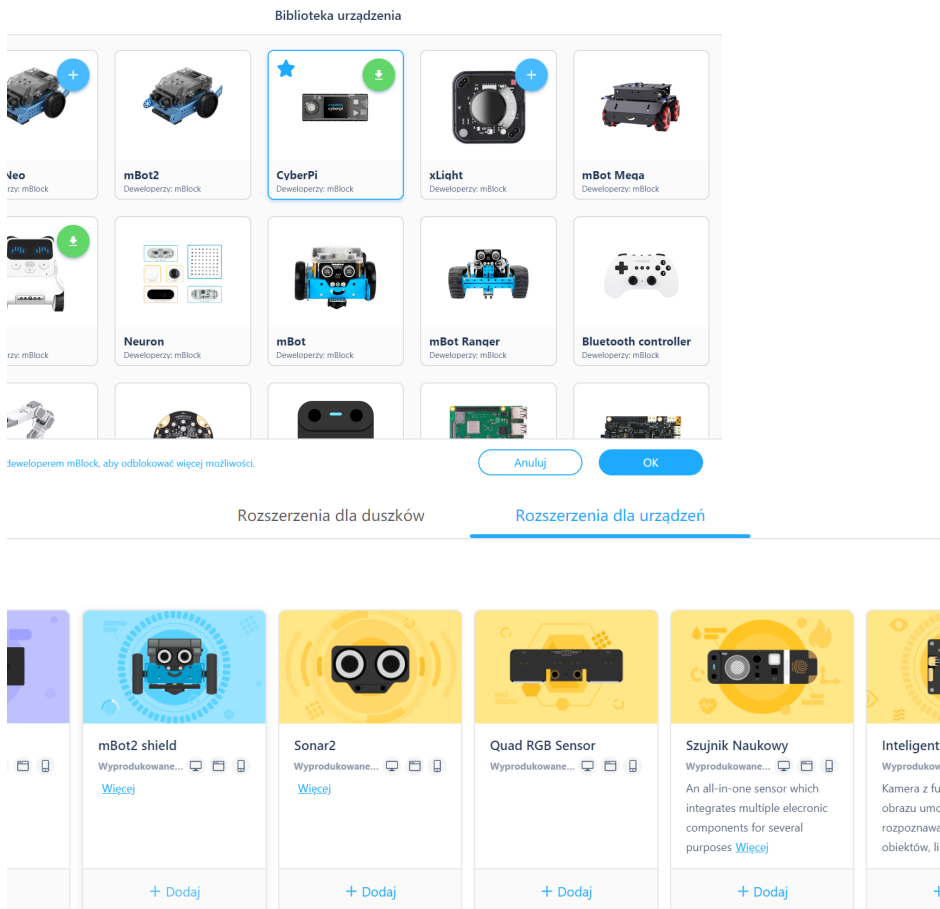


Zauważ- trzeba wyzerować punkty na początku; pytania muszą być wpisane w niebieski bloczek czujnika, a jeżeli chcesz, żeby quiz był głosowy- także w zielony bloczek "tekst na mowę". Zadawaj pytania, na które można odpowiedzieć "tak" lub "nie"- to uprości komunikację z mBotem, chyba, że chcesz napisać mBotowi bardziej skomplikowany program. Jeżeli zaprogramujesz dalsze pytania- mogą one być dalszą częścią tego skryptu, trzeba tylko zaplanować pauzę na działania mBota- np. 5 sekund.

Krok 3:

### Zaprogramuj reakcję mBota.

Teraz wejdź w zakładkę "Urządzenia". Pamiętaj, że musisz dodać z biblioteki urządzeń CyberPi oraz potrzebne Ci rozszerzenia dotyczące mBota2, a potem połączyć komputer ze swoim mBotem.



Co ma zrobić mBot? To już zależy od Twojej inwencji. U nas za poprawną odpowiedź mBot świeci na niebiesko, mówi "wow" i jedzie do przodu. Kiedy odpowiedź jest zła- mBot świeci na czerwono, piszczy i "mówi nie" kręcąc się w prawo i w lewo. Do komunikacji z duszkiem wykorzystujemy bloczek "kiedy otrzymam" z działu Kontrola:

kiedy otrzymam Zapal niebieskie światło i zagraj ▾

Cały skrypt mBota może wyglądać tak:

```
kiedy otrzymam Zapal niebieskie światło i zagraj
LED wszystkie wyświetl R 0 G 0 B 255 przez 3 sekund
odtwórz wow
jedź naprzód prędkość 50 obr/min przez 3 s
```

```
kiedy otrzymam zapal czerwone światło
LED wszystkie wyświetl przez 1 s
zagraj dźwięk 700 Hz przez 1 s
silnik lewe koło (EM1) obracaj 50 obr/min przez 1 s
silnik lewe koło (EM1) obracaj -50 obr/min przez 1 s
silnik lewe koło (EM1) obracaj 50 obr/min przez 1 s
silnik lewe koło (EM1) obracaj -50 obr/min przez 1 s
```

Jeżeli na kolejne pytania quizu będzie można odpowiedzieć “tak” lub “nie”- wystarczy tylko te skrypty. Od Ciebie zależy czy mBot również przejedzie jakąś trasę, kiedy odpowiedzi będą poprawne (ale wtedy robot nie może się kręcić przy złej odpowiedzi, bo zmieni kierunek). Czy można zaprogramować wyścigi mBotów tak, aby dojechały do mety dzięki poprawnym odpowiedziom?

Krok 4:

**Możesz wykonać taki sam quiz na telefonie w aplikacji mBlock.** Nie będziemy go już omawiać krok po kroku- bo etapy są prawie identyczne jak na komputerze. Pamiętaj, że również trzeba pobrać dla duszka rozszerzenie “Text to speech”, jeżeli chcesz, żeby telefon zadawał pytania głosem, a z urządzeń trzeba dodać CyberPi i rozszerzenie dla mBota- tak samo, jak na komputerze. Oto przykładowe skrypty.

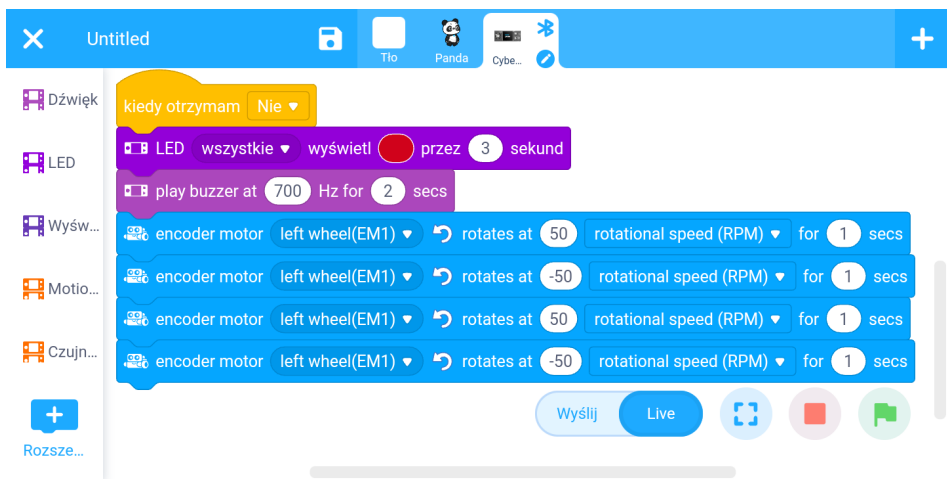
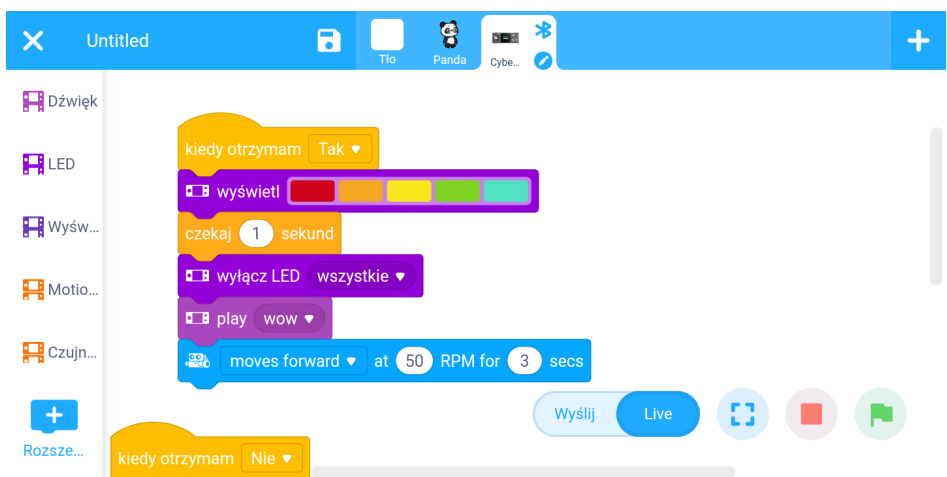
Skrypty duszka:

```
Untitled
Ruch
Wygląd
Dźwięk
Zdarz...
Kontr...
+
Rozsze...

po kliknięciu
ustaw Licznik na 0
Mówić Czy stolicą państwa krzyżackiego był Malbork?
zapytaj Czy stolicą państwa krzyżackiego był Malbork? i czekaj
jeżeli odpowiedź = Tak to
nadać Tak
Mówić Brawo. Jeden punkt.
zmień Licznik o 1
```



Skrypty CyberPi i mBota:



### 3. Podsumowanie i ewaluacja

czas: ok 5 minut

By znaleźć ostatni artefakt - herb Zawiszy Czarnego, sławnego rycerza spod Grunwaldu i móc go odkryć - należy zajrzeć tutaj:

<https://learningapps.org/watch?v=p71678m0521>

Nauczyciel rozmawia z dziećmi, czy podobała im się ta podróż w czasie i czy chciałyby naprawdę odbyć taką wędrowkę? Jakie korzyści i zagrożenia mogłyby nieść taka podróż?

## Uwagi/alternatywy:

### Minimum:

- lekcja o wielkiej wojnie z Krzyżakami i jej następstwach;
- ostatnia część przygód Tymka;
- ostatnim elementem generatora Tymka jest **herb Zawiszy Czarnego**, jednego z uczestników bitwy pod Grunwaldem - znajdziecie go w kartach do tej lekcji lub w zagadce:

<https://learningapps.org/watch?v=p71678m0521>

- Turniej Historyczny przy pomocy mBotów .

### Dodatki do wyboru lub wszystkie:

- karty jako nagrody dla uczniów za wykonanie zadania - elementy wyposażenia rycerza

<https://www.pearltrees.com/radator/lekcja-8/id43618938>

- zaprogramowanie animacji planu bitwy pod Grunwaldem (opis znajduje się poniżej, w dodatkach)

- uczniowie mogą też wykonać **grę - kosmiczny pojedynek** pomiędzy Tymkiem a Jonasem. Gra została zremiksowana z bardzo ciekawego projektu gimnazjalistów (informacje w opisie gry):

<https://scratch.mit.edu/projects/548760462/>.

- ostatnim miejscem na dużej papierowej mapie jest oczywiście **Grunwald**. Można na zakończenie zaprogramować mboty, by przejechały po kolei przez wszystkie miejsca zaznaczone na mapie, dzięki czemu uczniowie przypomną sobie i utrwalą, z jakimi wydarzeniami i osobami się wiązały.

### Plan bitwy pod Grunwaldem [dodatek]

Uczniowie mogą również wykonać animację - plan bitwy pod Grunwaldem. Poszczególne formacje rycerzy widziane z góry (tak, jak się rysuje plany bitew) będą przesuwane tak, aby prześledzić przebieg bitwy.

#### Krok 1:

Przejdź przez tę animację i zastanów się na czym polega:

<https://scratch.mit.edu/projects/548356738/>.

Animacja składa się z dwóch części. Po spacji widzisz po kolei 4 plansze z przebiegiem bitwy (pochodzą z Wikipedii). Po kliknięciu zielonej flagi pojawia się pierwsza plansza bitwy i dodatkowe prostokąty pokazujące oddziały polskie, litewskie i krzyżackie. Dodatkowe prostokąty to ruchome duszki, dla odróżnienia od tła są przekreślone. Animacja polega na tym, że po kliknięciu po kolei strzałek: w górę, w dół, w prawo, w lewo duszki meldują się na właściwych miejscach, potem tło zmienia się na mapkę Google z terenem bitwy, pojawia się kolejna plansza z przebiegu bitwy, kolejna strzałka odsyła duszki na nowe miejsce i znów następuje zmiana mapy i tak dalej. Jak wykonać taką animację?

#### Krok 2:

**Osadź potrzebne tła.** Plansze przebiegu bitwy pochodzą z Wikipedii:

Autorstwa 4C - Own work, based on PNG version created by Screen2, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=1021613>

Autorstwa 4C - Own work, based on PNG version created by Screen2, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=1021618>

Autorstwa 4C - Own work, based on PNG version created by Screen2, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=1021622>

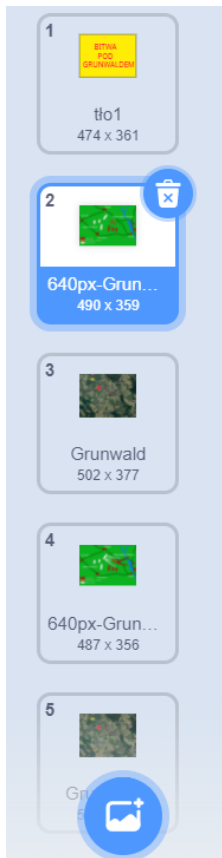
Autorstwa 4C - Own work, based on PNG version created by Screen2, CC BY-SA 3.0,

<https://commons.wikimedia.org/w/index.php?curid=1021628>



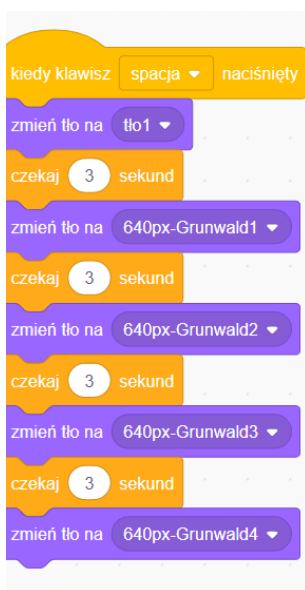
## #SuperKoderzy / Podróżnicy w czasie / Ratujemy z Tymkiem victorię grunwaldzką

Mapa terenu bitwy to wycinek z mapy Google, możesz go jednak pobrać z naszej animacji. Jest to ten sam wycinek zdublowany cztery razy. Pierwszym tłem jest tytuł wykonany w edytorze graficznym, w zakładce Tła. Potem na zmianę następują - plansza bitwy i mapa terenu.



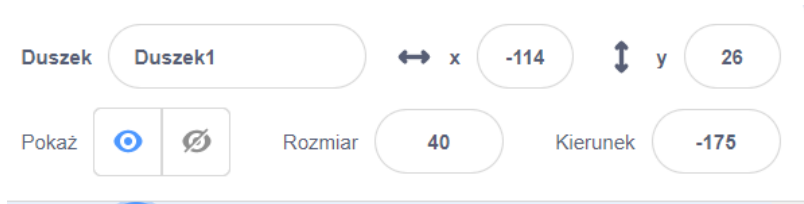
Krok 3:

**Zaprogramuj animację początkową** - zmiany plansz. Po kliknięciu spacji cztery plansze z przebiegu bitwy powinny się dość wolno zmieniać. Pamiętaj, że masz różne tła, a chcesz pokazać tylko plansze z Wikipedii. Na przykład tak:



Krok 4:

**Narysuj duszki** - oddziały. Są to prostokąty o odpowiednich rozmiarach, narysowane w edytorze, w zakładce Kostiumy. Rozmiar prostokąta możesz zmienić za pomocą panelu Rozmiar, pod sceną:



To miejsce w ogóle będzie źródłem najważniejszych informacji w pracy nad animacją.

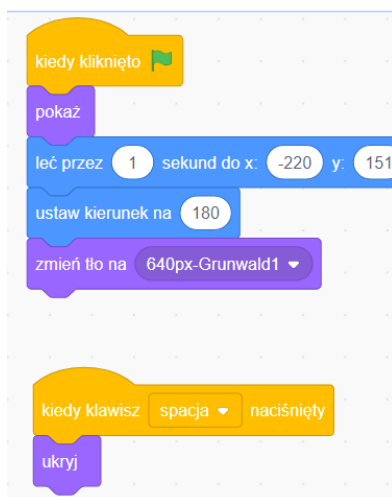
Przekreśl prostokąty, albo zaznacz je jakoś, żeby było wiadomo, co jest duszkiem, a co tłem. Zauważ, że są duszki, których nie widać od początku, ponieważ oddziały zmieniały nie tylko położenie, ale i rozmiary. Niektóre duszki zaś znikają, kiedy ilość oddziałów się zmniejsza.

Krok 5:

**Zaprogramuj początek drugiej części animacji** (pierwsza to tylko cztery zmiany plansz). Co się dzieje po kliknięciu zielonej flagi?

1. Powraca pierwsza plansza z bitwy.
2. Pojawiają się duszki oddziały (a w pierwszej części nie było ich widać).
3. Duszki meldują się na bokach planszy.

Każdy duszek musi ukryć się po spacji, żeby nie przeszkadzał w pierwszej części animacji. Potem musi się pokazać i zameldować na boku, gdzie go ustawimy. Ponieważ duszki będą się obracać, ważną daną jest też kierunek, który widać obok rozmiaru. Oto przykładowy skrypt pierwszego duszka:



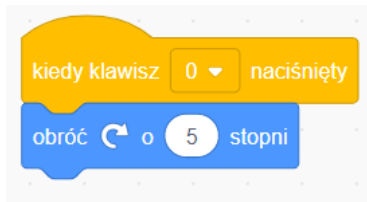
Pozostałe duszki mają analogiczne skrypty, zmieni się tylko położenie. Nie trzeba też dołączać zmiany tła, wystarczy zrobić to raz.

Krok 6:

**Zaprogramuj zmiany położenia oddziałów.** Tu czeka Cię żmudna praca. Pamiętaj, że możesz kopiować swoje skrypty i modyfikować tylko komendy i współrzędne x, y oraz kierunek. Niektóre duszki trzeba będzie w pewnej chwili ukryć, a inne pokazać dopiero na trzeciej planszy. Co robi duszek?

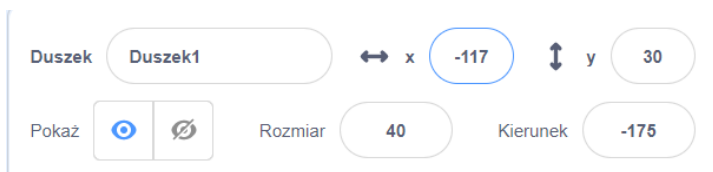
1. Melduje się na swoim oddziale na planszy bitwy po naciśnięciu odpowiedniej strzałki.
2. Czasem zmienia kierunek i rozmiar.
3. Pierwszy duszek zmienia tła - po ustawieniu oddziałów pokazuje mapę Google, a potem kolejną planszę bitwy, pozostałe już nie muszą tego robić.

Jak to zrobić? Trzeba poustawiać duszki na miejscach oddziałów ręcznie. Przyda nam się technicznie taki skrypt:



Pomoże nam ustawić duszka w dobrym kierunku. Każdy duszek musi być uruchamiany innym klawiszem (żeby wszystkie nie kręciły się równocześnie). Można też używać bloczka obrotu bez bloczka komendy - gdy jesteś w skrypcie duszka, kliknięcie w bloczek obrotu obróci duszka. Zmniejszamy wielkość obrotu do 5 stopni, żeby duszek nie obracał się zbyt gwałtownie.

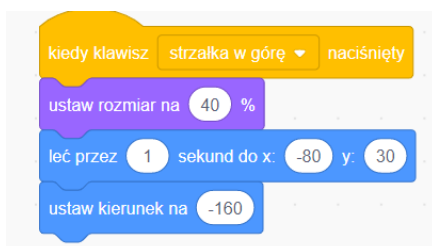
Kiedy już każdy oddział stoi na miejscu, trzeba zapisać to w skrypcie, korzystając z informacji z prawej strony pod sceną:



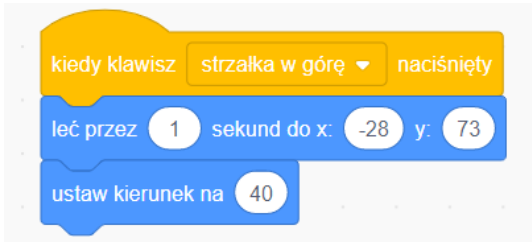
Zobacz, jak wygląda pierwszy skrypt ruchu pierwszego duszka:



A tak wygląda skrypt pierwszy drugiego duszka:

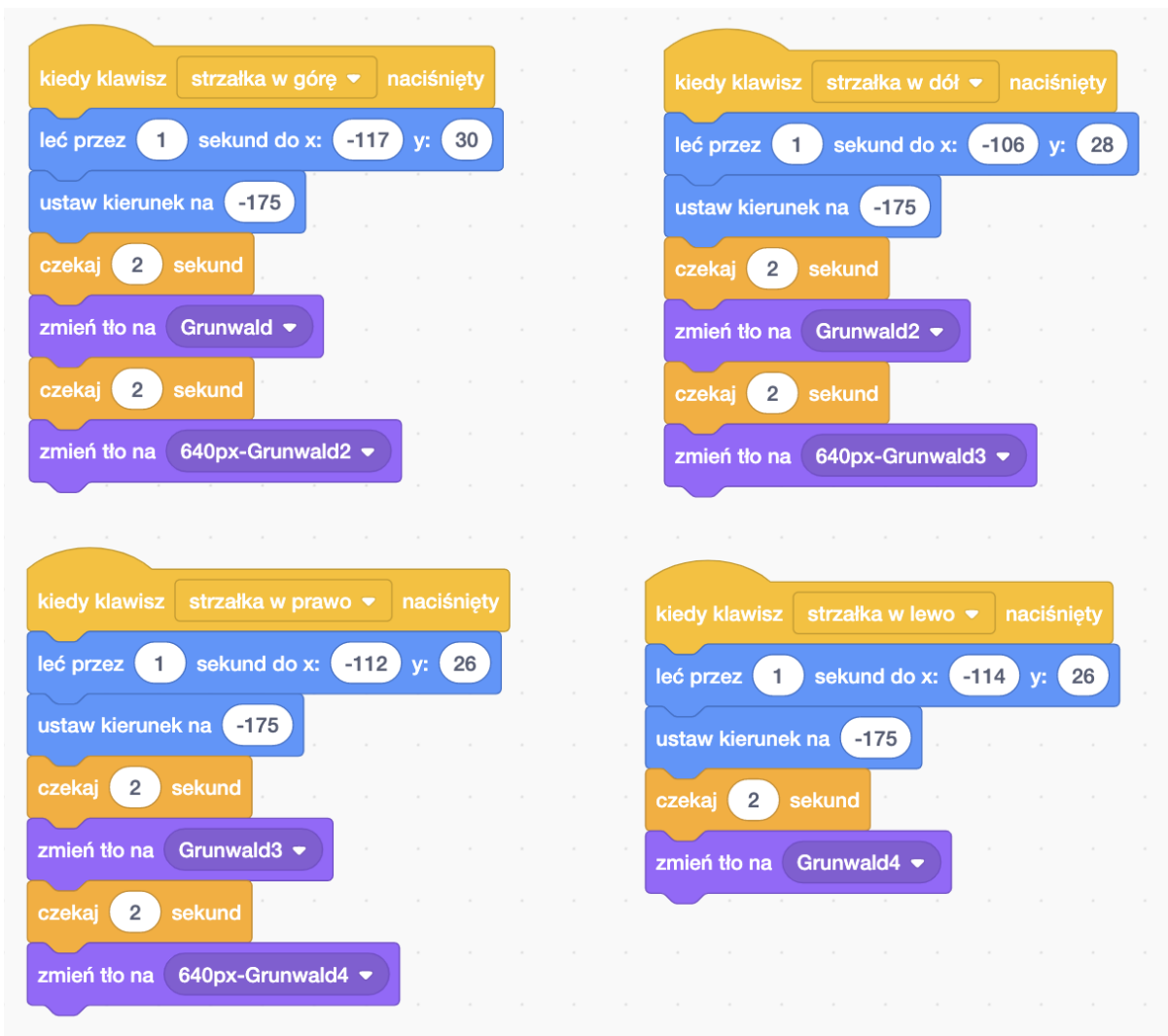


To jedyny duszek, który w pewnej chwili zmienia rozmiar i dlatego od początku musimy ustawić jego rozmiar. Pierwszy skrypt trzeciego duszka jest już prosty:



i z pozostałymi duszkami jest podobnie.

Tę procedurę musisz powtórzyć cztery razy. Każda kolejna zmiana na polu bitwy jest połączona z kolejną strzałką, więc musisz zmienić planszę na kolejną fazę bitwy, poustawiać duszki ręcznie i na bloczkach ruchu spisać aktualne położenie i kierunek tego duszka. Pamiętaj, że możesz zdublować pierwszy skrypt ruchu, a potem zmienić strzałkę i pozmienić x, y i kierunek, w jednym przypadku także rozmiar. Trzy duszki znikają (dwa oddziały krzyżackie i jeden litewski), zaś jeden się pojawia (mały oddział krzyżacki - pozostałości pokonanych). Pierwszy duszek zawsze zmienia tła - plansza, mapa i tak na zmianę. Uważna analiza naszych skryptów pomoże Ci zbudować animację poprawnie. Uruchomienie animacji pokaże Ci błędy w położeniu oddziałów w stosunku do tła i możesz je wtedy skorygować. Powodzenia!



# Podróżnicy w czasie

**Autorka:** Joanna Płatkowska

## Lekcja dodatkowa:

# Poznajemy mBota

Lekcja, podczas której uczniowie zdobędą podstawowe informacje o robotach mBot2 i poznają środowisko mBlock v5.4.0. . Należy ją przeprowadzić przed lekcją nr 5 „Murujemy nowy dom z Kazimierzem Wielkim”.

### Cele zajęć:

Uczeń powinien:

- opisywać, jakie działania potrafi wykonać mBot2
- wymieniać podstawowe elementy mBota2 i jego funkcje,
- tworzyć proste skrypty dla mBota2 w programie mBlock

### Materiały pomocnicze:

- roboty mBot 2 (dla poprzedniej wersji robotów mBot należy skorzystać z archiwalnej wersji tych scenariuszy, dostępnych pod adresem \*\*\*);
- program mBlock lub wersja przeglądarkowa (do kodowania robota, scenariusze napisane są dla wersji V5.4.0):

online: <https://ide.mblock.cc/>

offline: <http://www.mblock.cc/download/off>

- komputery stacjonarne lub laptopy (ze sprawnym portem USB).

### Pojęcia kluczowe:

→ robot → silnik → czujnik → mikrokontroler CyberPi  
→ mBlock

**Czas na realizację zajęć:** 45 minut (1 godzina lekcyjna)

### Metody pracy:

- grupowa – praca w grupach z robotem,
- ćwiczenia praktyczne,
- pogadanka

### Formy pracy:

- praca z całą klasą
- grupowa - praca w grupach z robotem,

### Treści programowe (związek z podstawą programową)

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII, informatyka:

1. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

2. Tworzy polecenia lub sekwencję poleceń dla określonego planu działania prowadzące do osiągnięcia celu.

2. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Uczeń:

1. Programuje wizualnie:

a. proste sytuacje/historyjki według pomysłów własnych i pomysłów opracowanych wspólnie z innymi uczniami.

b. pojedyncze polecenia lub ich sekwencje sterujące robotem lub obiektem na ekranie komputera, bądź innego urządzenia cyfrowego.

3. Zapisuje efekty swojej pracy we wskazanym miejscu.

4. Rozwijanie kompetencji społecznych. Uczeń:

1. Współpracuje z innymi uczniami, wymienia się z nimi pomysłami i swoimi doświadczeniami wykorzystując technologię.

2. Wykorzystuje możliwości technologii do komunikowania się w procesie uczenia się.

Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy IV-VIII; historia:

## Przygotowania przed zajęciami:

1. **Złożenie robota.** W scenariuszach nie przewidujemy osobnej lekcji poświęconej na składanie robotów. Nauczyciel może sam przygotować roboty przed rozpoczęciem realizacji ścieżki przedmiotowej, może także poprosić o pomoc uczniów lub zaproponować przygotowanie robotów np. w ramach lekcji techniki lub informatyki. Obrazkowa instrukcja złożenia robota dostępna jest w materiałach producentach [TUTAJ](#) oraz w książeczce dołączonej do zestawu mBot.
2. **Oprogramowanie.** Przed zajęciami należy także zainstalować na komputerach aplikację mBlock, stworzoną na bazie środowiska Scratch, która będzie nam służyła do programowania robota. Link do pobrania programu: <http://www.mblock.cc/download/> Poniższe scenariusze napisane są z wykorzystaniem wersji 5.4.0
3. **Organizacja zajęć.** Realizacja projektu wymaga pracy w grupach, optymalnie 2- lub 3-osobowych. Do każdej z grup przypisujemy jednego mBota - podział na grupy wynika więc z liczby robotów, którymi dysponujemy.

## Przebieg zajęć:

### 1. Wprowadzenie w tematykę i integracja grupy:

Początkiem lekcji może być prezentacja możliwości robota przez nauczyciela i dyskusja z uczniami na temat ich dotychczasowych kontaktów i doświadczeń z robotami.

### 2. Część zasadnicza

#### Zadanie 1: budowa robota

Nauczyciel prezentuje mBota i rozdaje roboty uczniom. Nim je włączymy i podłączymy do komputerów, konieczne jest omówienie jego budowy.

Nauczyciel wymienia kolejne nazwy części robotów i zachęca uczniów do wskazania, gdzie się one znajdują. Pytania można zadawać po polsku lub w języku obcym. Jeśli uczniowie nie potrafią od razu wskazać wymienionego elementu, można opisać zasadę działania danej części robota - ważne, by uczniowie sami ją odkryli!

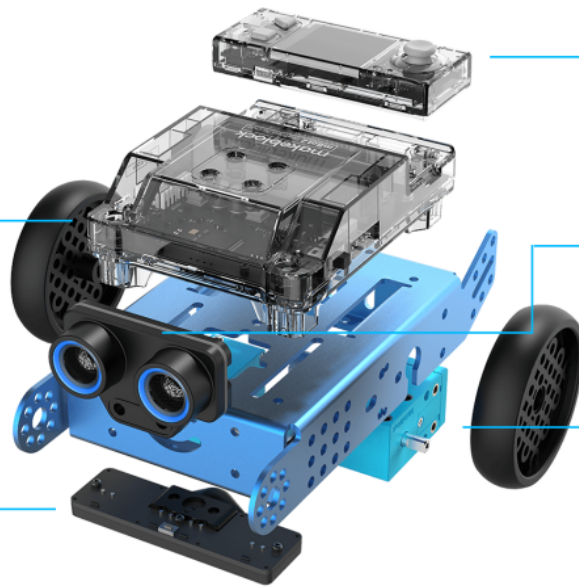
Pod tym linkiem <https://epax.pl/makeblock-robot-mbot2-p-17.html> jest grafika, którą wklejam pod spodem. Akurat u pana Grzegorza takiej nie ma. Może oni (albo ktoś innym, kilka sklepów ma taką grafikę) by udostępnił, albo najwyżej Ty możesz zrobić podobną na podstawie angielskiej wersji, to są generalnie luźne tłumaczenia stąd <https://education.makeblock.com/mbot2/>

### mBot2 sterownik

Zawiera porty do podłączania modułów zewnętrznych i silników oraz zintegrowany akumulator ładowany przez USB

### 4 czujniki linii RGB

Nowy, poczwórny czujnik z rozpoznawaniem kolorów wspiera zaawansowane techniki śledzenia linii.



### CyberPi

Mikroprocesor ESP32 z komunikacją bezprzewodową WiFi i Bluetooth, Zintegrowane czujniki i wyświetlacz. Programowany blokowo i w Python. Może być używany samodzielnie po odłączeniu od robota.

### Nowy czujnik odległości

Pomiar odległości, wykrywanie przeszkód i sygnalizacja przy pomocy 8 diod LED

### Silniki z enkoderami

Nowe silniki z czujnikami obrotu pozwalającymi na pomiar przejechanego dystansu, kontrolę prędkości do 200RPM i położenia z dokładnością do 1 stopnia.

Warto porównać mBota2 do człowieka. “Mózgiem” mBota2 jest mikrokontroler CyberPi, “oczami” jest czujnik ultrasoniczny, “ciałem” sterownik Mbot2, a “nogami” podwozie z kołami. Dodatkowo mBot2 może komunikować się odtwarzając dźwięki, wyświetlając kolory na listwie LED oraz komunikaty czy grafiki na wyświetlaczu. Podobnie jak u człowieka, “mózg” robota może działać tylko, jeśli połączony jest z resztą “ciała” (podwozie) lub bezpośrednio ze źródłem zasilania (podłączony kablem do komputera lub bezpośrednio do kontaktu). “Ciało i nogi” robota (sterownik mBot2 i podwozie) nie są w stanie się poruszać bez “mózgu” (mikrokontrolera).

Jeśli w szkole są dostępne zarówno mBoty2, jak i mBoty w starszej wersji konieczne będzie pokazanie uczniom różnic pomiędzy tymi wersjami. Porównanie tych wersji można znaleźć w Załączniku nr 2.

## Załącznik nr 2

	mBot 1.1, mBot-S Explorer Kit	mBot2
Sterownik	mCore	CyberPi
Liczba programów w pamięci	1	8
Komunikacja bezprzewodowa	Bluetooth lub 2,4G, IR	Bluetooth, Wi-Fi oraz Wi-Fi LAN
Pilot	tak	nie
Urządzenia wejścia i czujniki zintegrowane	Przycisk, czujnik światła	Joystick, przycisk x2, czujnik światła, mikrofon, żyroskop – akcelerometr
Urządzenia wyjścia	Buzzer (odtwarzający pojedyncze dźwięki/nuty o różnej wysokości), RGB-LED x2, wyświetlacz LED 16x8px mono (Explorer Kit)	Głośnik, RGB LED x5, wyświetlacz kolorowy IPS 128x128px
Języki programowania	mBlock/Scratch, ArduinoC	mBlock/Scratch, microPython, Python3
Moc silnika	zależna od poziomu naładowania baterii = im mniej naładowana bateria, tylko wolniej jedzie robot przy tym samym programie	niezależna od poziomu baterii = robot jedzie w takim samym tempie i pokonuje taką samą trasę niezależnie od poziomu naładowania baterii

Pełną instrukcję i grafiki pokazujące poszczególne elementy mBota2 można znaleźć **TUTAJ** oraz **TUTAJ**.

**Zadanie 2: co mBot potrafi?**

Na początek nauczyciel wyjaśnia uczniom, że mBot2 może pracować w dwóch trybach: live lub upload.

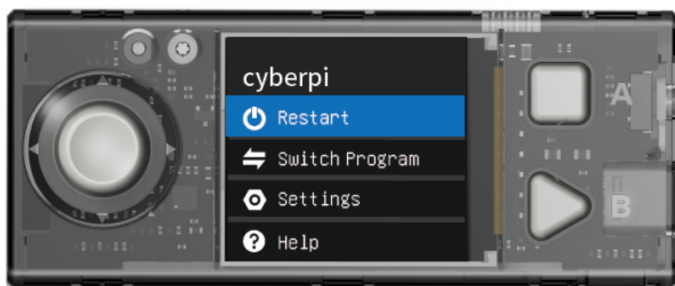
Tryb live polega na ciągłym i bezpośrednim połączeniu robota z komputerem (przy użyciu kabla lub Bluetooth Dongle). Robot wykonuje wszystkie polecenia pisane i na bieżąco zmieniane z poziomu



komputera. W tym trybie robot może komunikować się z postaciami na ekranie komputera. Ten tryb będzie wykorzystywany w naszych scenariuszach.

Try upload polega na napisaniu na komputerze, a następnie przesłaniu (wgraniu) programu robotowi. Jeśli chcemy wprowadzić zmiany do programu, musimy po zrobieniu niezbędnych poprawek, ponownie wgrać go robotowi. Takie programy działają zdecydowanie szybciej i tryb ten jest idealny do bardzo skomplikowanych, wymagających programów. Każdy robot w momencie pierwszego uruchomienia ma standardowo wgrane kilka programów, które warto pokazać uczniom. Nie są one wykorzystywane w późniejszych scenariuszach, jednak mogą one pobudzić ciekawość uczniów i chęć włączenia się w programowanie.

Po włączeniu robota suwakiem pojawia się ekran startowy:



Po wybraniu joystickiem opcji Switch Program możemy wybrać spośród ośmiu wgranych programów. Warto z uczniami spróbować nagrywania i odtwarzania (Magical Recorder), mBot2\_demo 2 (unikanie przeszkód) czy mBot2\_demo 3 (programowanie ścieżki robota). Instrukcje do każdego programu pojawiają się na wyświetlaczu. Żeby wrócić do głównego menu i zmienić program, trzeba nacisnąć znajdujący się po prawej stronie guzik Home.

### Zadanie 3: łączenie mBota z komputerem

Po przeprowadzonych testach nauczyciel zaprasza uczniów do napisania pierwszych programów komputerowych z wykorzystaniem mBota2. Uczniowie siadają do komputerów i łączą roboty z komputerami.

Warto tu zauważyć, że można korzystać z programu mBlock do pisania programu dla robota w wersji online lub offline. Nauczyciel oczywiście wybiera wersję bardziej pasującą do kontekstu szkoły i wydajności łącza internetowego. Przewagą wersji online jest to, że jest ona aktualizowana szybciej i sprawniej niż wersja offline, a także mBot2 jest "gotowy do użycia" w tej wersji. Przewagą wersji offline jest to, że nie jest ona zależna od stabilności łącza internetowego, a także, że pozwala bezpośrednio sprawdzić czas trwania nagrań, co może w znaczący sposób przyspieszyć pracę nad projektem. Równocześnie w tej wersji mBota2 trzeba w dwóch prostych krokach przygotować do programowania. W przypadku wyboru wersji online konieczne jest jednorazowe pobranie i zainstalowanie programu mLink:

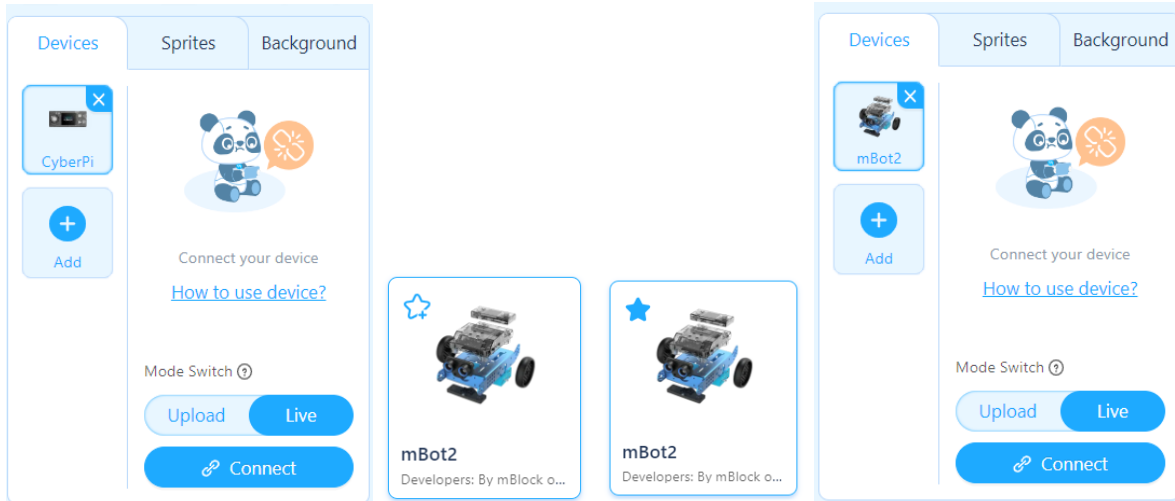
<https://mblock.makeblock.com/en-us/download/mlink/>

#### Kolejne kroki:

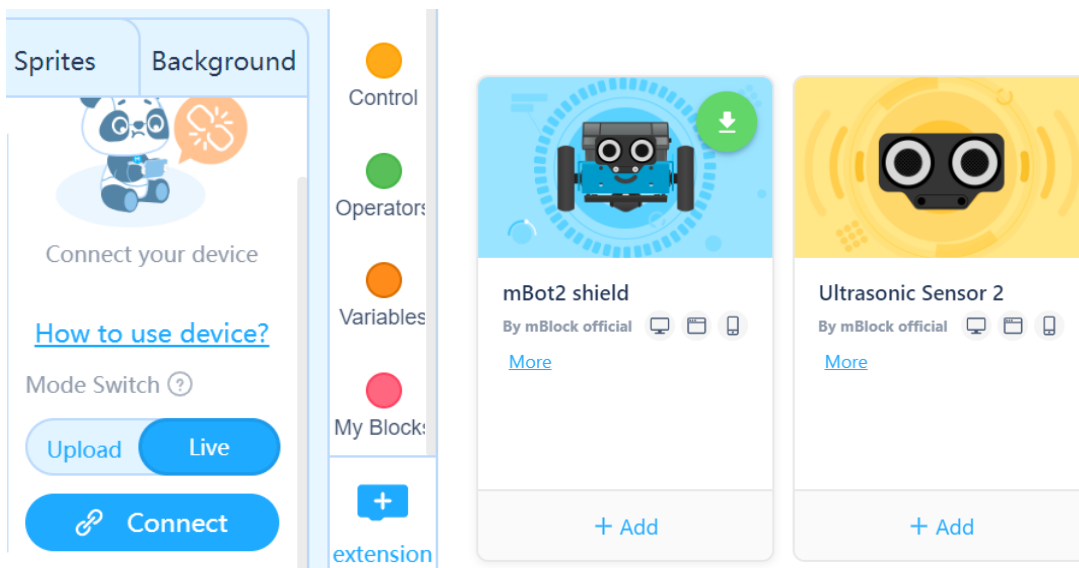
1. Uruchomienie aplikacji mBlock.
2. Podłączenie modułu Bluetooth Dongle; moduł powinien zacząć powoli migać niebieskim światłem.
3. Naciśnięcie na module ikony Bluetooth; moduł powinien zacząć szybko migać niebieskim światłem.
4. Włączenie mBota2; moduł powinien zacząć świecić ciągłym światłem (to znak, że moduł sparował się z robotem).

5. Włączenie mBlocka:

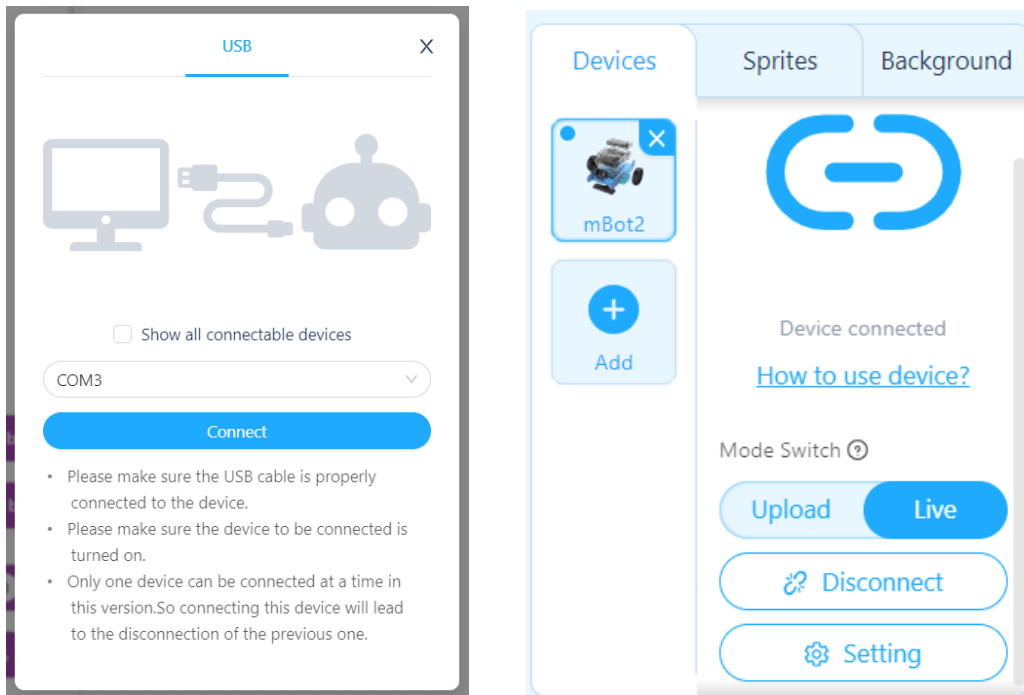
a. w wersji online przy pierwszym uruchomieniu na danym komputerze w lewym dolnym rogu usunięcie CyberPi krzyżykiem i kliknięcie Add w celu wybrania mBota2. Po otwarciu biblioteki urządzeń warto najechać na lewy górny róg pola z mBotem, gdzie pojawi się biała gwiazdka. Po kliknięciu, biała gwiazdka zmieni się na niebieską, a tym samym mBot2 zostanie ustawiony jako urządzenie domyślne i będzie pojawiał się zawsze po uruchomieniu mBlocka.



b. w wersji offline przy pierwszym uruchomieniu na danym komputerze dodanie rozszerzeń (extensions): podwozie (shield) oraz czujnik ultrasoniczny (ultrasonic sensor 2) poprzez kliknięcie +Add. Dodatkowo jeśli obok rozszerzenia pojawia się ikona zielonej strzałki, należy ją kliknąć, żeby zaktualizować dane rozszerzenie.



6. Połączenie robota z komputerem poprzez kliknięcie guzika Połączenie (Connect). Pojawi się komunikat o porcie (standardowo będzie to COM3 lub COM4), gdzie należy kliknąć ponownie Połączenie (Connect). Można sprawdzić czy robot na pewno połączył się z komputerem szukając niebieskiej kropki w lewym górnym rogu jego ikony i zmianie niebieskiego przycisku Connect na biały przycisk Disconnect.

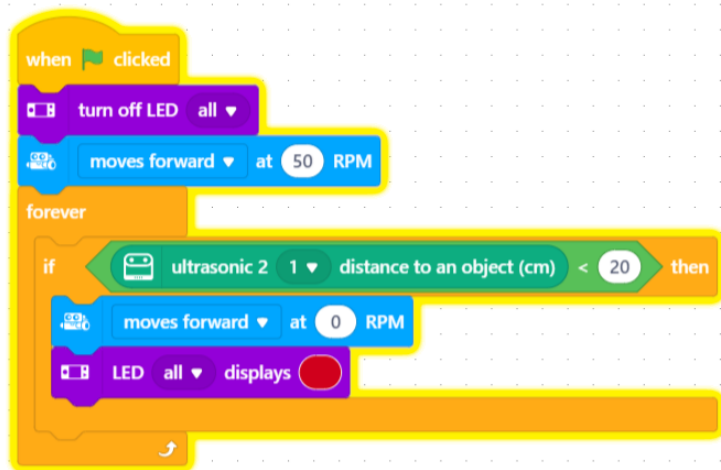


#### Zadanie 4: pierwszy program dla mBota

Nauczyciel pokazuje uczniom, gdzie znaleźć skrypty mBota. Uczniowie przeglądają różne podgrupy tych bloczków i zastanawiają się, z których bloczków mógł korzystać robot wykonując wcześniej odtwarzane programy. W naszym pierwszym programie użyjemy ultrasonicznego czujnika odległości. Chcemy, by robot po napotkaniu przeszkody zatrzymał się i wyświetlił na pasku kolor czerwony. Konieczne jest tu ustalenie, w jakiej odległości od przeszkody ma zatrzymać się robot. W podanych poniżej przykładzie ustalamy, że będzie to wartość 20.

**Wskazówka:** Warto już tu zwrócić uczniom uwagę na to, że roboty programujemy w zakładce Urządzenia (Devices), a postacie na ekranie w zakładce Postacie (Sprites).

Napisanie takiego programu, może być na początku trudne. Jeśli tak jest, nauczyciel podpowiada uczniom np. układając na polu roboczym rozsypankę z wszystkich potrzebnych klocków i zachęca uczniów do poprawnego ułożenia skryptu. Poprawnie napisany skrypt znajduje się po prawej stronie.



### 3. Podsumowanie i ewaluacja – 5 minut

Nauczyciel pyta uczniów, co najbardziej podobało się im w zajęciach, a co najmniej. Pyta także o atmosferę panującą w klasie oraz o tempo pracy. Zapowiada, że na kolejnej lekcji uczniowie napiszą kolejne programy dla mBotów.

Zadanie domowe: proszę obejrzeć filmik na YouTube:

 [Getting started with Mbot2](#)

### 3. Podsumowanie i ewaluacja

Uczniowie dyskutują jak wykorzystać zdobytą wiedzę i jakie programy dla mBota2 mogliby napisać. Zapisują swoje pomysły i realizują je na kolejnych lekcjach.

### Uwagi/alternatywy:

Zachęcamy do utrwalania materiałów z przebiegu zajęć. Mogą to być zdjęcia, filmy, notatki, zrzuty ekranu, ciekawe lub zabawne teksty/informacje, które pojawiły się podczas lekcji. Posłużą one do udokumentowania przebiegu programu #SuperKoderzy.