



Zajęcia 1: Mucha w potrzasku .....	2
Zajęcia 2: Taniec w kosmosie .....	8
Zajęcia 3: Labirynt - cz.1 .....	15
Zajęcia 4: Labirynt - cz.2 .....	22
Zajęcia 5: Memo.....	36
Zajęcia 6: Generator gwiazd .....	46
Zajęcia 7: Brak Internetu.....	50
Zajęcia 8: Scratch FIFA cz.1 .....	60
Zajęcia 9: Scratch FIFA cz.2.....	65
Zajęcia 10: Królestwo .....	69
Zajęcia 11: Quiz .....	74
Zajęcia 12: Arkanoid.....	79
Zajęcia 13 i 14: Wyścigi.....	85
Zajęcia 15: Obrona zamku .....	101



# Zajęcia 1: Mucha w potrzasku

## Zadanie:

Wykonanie gry „Mucha w potrzasku”, która składa się z: samodzielnie latającej i sterowanej przez użytkownika Muchy oraz samodzielnie poruszającej się Żaby.

## Cel:

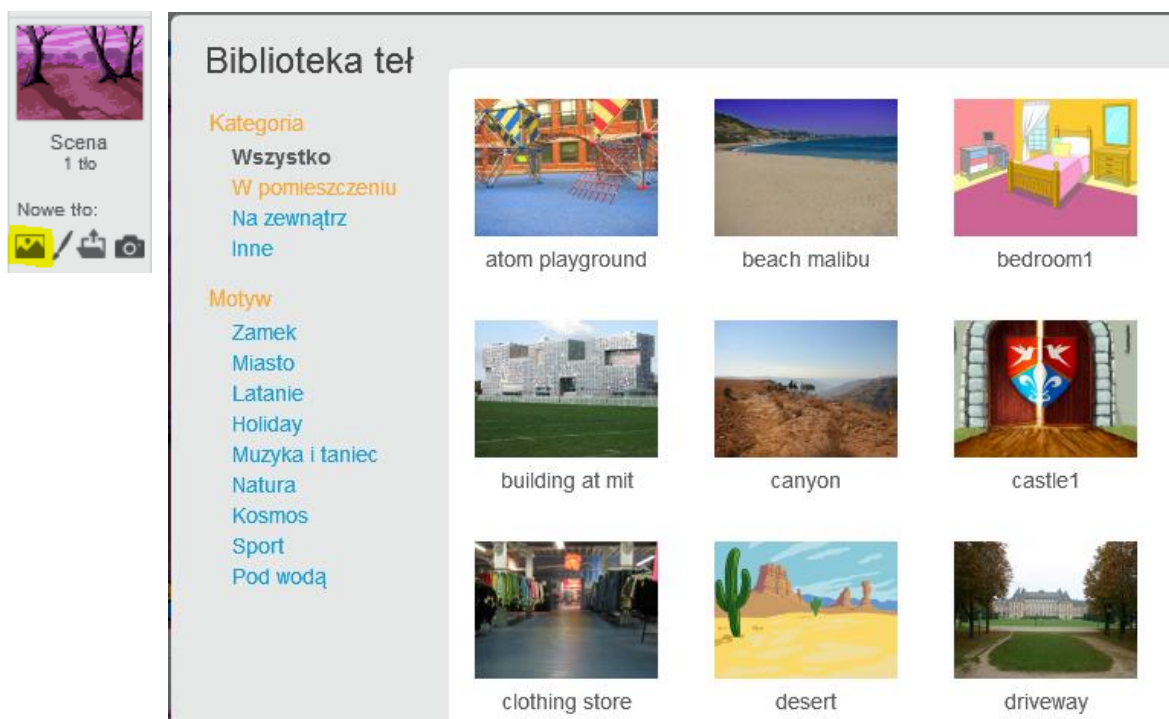
Zapoznanie się ze środowiskiem programistycznym Scratch. Elementy: podgląd, edytor duszka, narzędzia, wybór i tworzenie duszka. Zakładki: Skrypty, Kostiumy, Dźwięki. Edytor obrazu w zakładce Kostiumy.

## Wykonanie:

W kilku zdaniach należy przedstawić dzieciom co to jest Scratch, duszek, kostium itp.

### A. Przygotowanie duszków gry.

1. Wybieramy tło do naszej gry, importując je z biblioteki.



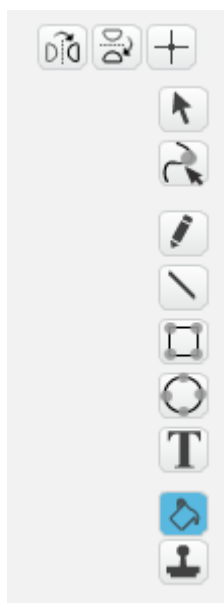
2. Importujemy duszka-Żabę z biblioteki.



3. Klikamy na żabę, następnie przechodzimy do zakładki Kostiumy. Klikamy lewym przyciskiem



myszy na kostium Żaby i wybieramy z listy DUPLIKUJ. Kolejnym krokiem jest przygotowanie drugiego kostiumu Żaby, w którym jej język jest niewidoczny. W tym celu wybieramy narzędzie WYPEŁNIJ KSZTAŁT paska narzędzi znajdującego się po prawej stronie Edytora Obrazu.



Wybieramy z palety kolorów biały prostokąt przekreślony czerwoną linią, jest to PRZEZROCYSTOŚĆ i wypełniamy tą barwą język Żaby.



Następnie klikamy na niebieską ikonkę informacji (i) znajdującą się na ikonie duszka Żaby i wybieramy styl obrotów na lewo-prawo.



4. Kolejnym bohaterem naszej gry jest Mucha, dlatego rysujemy ją w Edytorze Obrazu klikając na ikonę NAMALUJ NOWEGO DUSZKA.

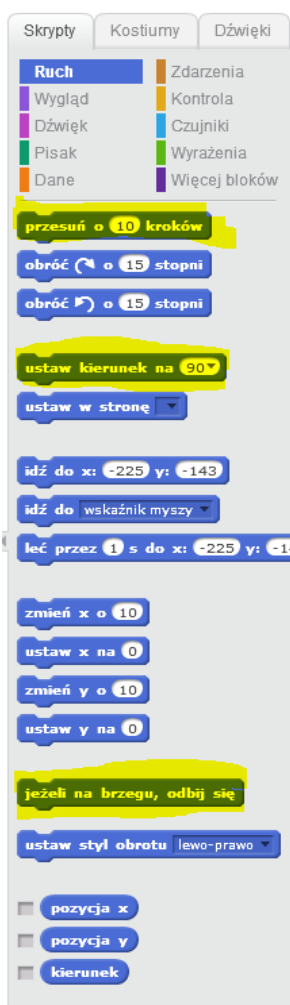
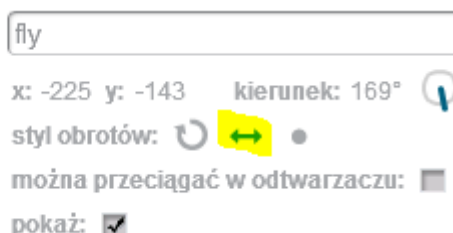




Następnie wykorzystując narzędzia zaznaczone na żółto poniżej malujemy Muchę!



Teraz podobnie, jak poprzednio w stosunku do Żaby, klikamy na (i) znajdujące się na ikonie duszka-Żaby i wybieramy styl obrotów na lewo-prawo.



## B. Programowanie – sterowanie Muchą

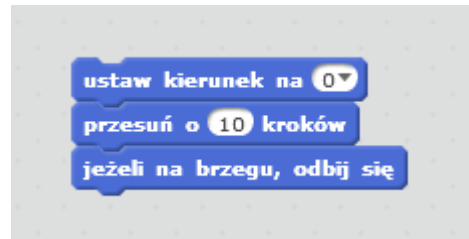
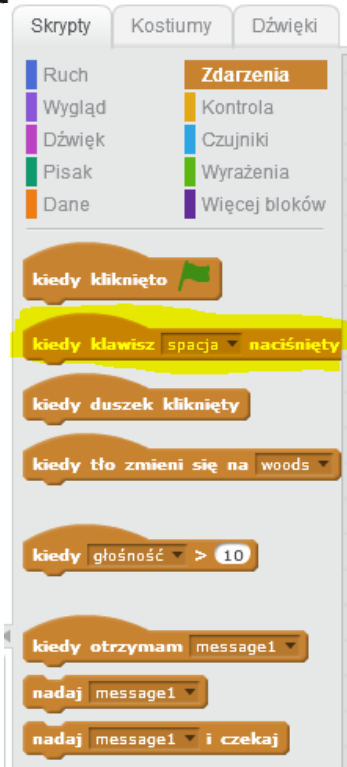
1. Klikamy na ikonę muchy i przechodzimy do zakładki Skrypt.
2. Naszym pierwszym zadaniem jest „zaprogramowanie” Muchy w taki sposób, żebyśmy mogli nią sterować. W tym celu wykorzystamy komendy bloków: Ruchu i Zdarzenia.



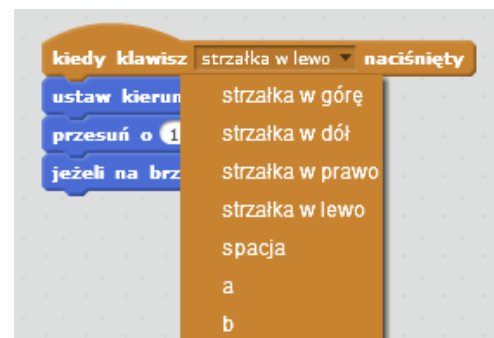
Będziemy poruszać naszą Muchą przy pomocy klawiatury. Użyjemy do tego celu komend:

- bloku RUCHU → USTAW KIEDUNEK NA..;
- bloku RUCHU → PRZESUŃ O10 KROKÓW;
- bloku RUCHU → JEŻELI NA BRZEGU, ODBIJ SIĘ.

Aby móc skorzystać z danej komendy łąpiemy ją w menu, klikamy na nie i przenosimy na obszar Edytora Duszka.

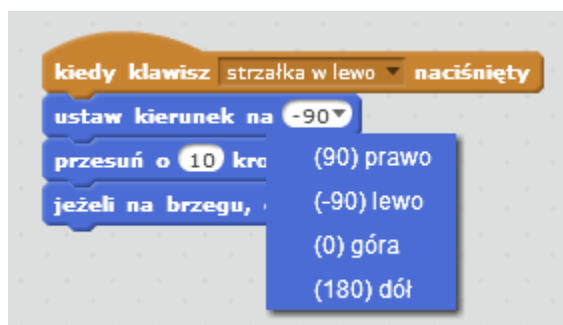


3. Nasz duszek jest już wyposażony w komendy ruchu. Potrzebne są nam jeszcze komendy pozwalające sterować nim za pomocą klawiatury. W tym celu klikamy na blok ZDARZENIA wybieramy komendę KIEDY KLAWISZ... NACIŚNIĘTY i przeciągamy ją na pole edytora.



4. Klikając lewym przyciskiem myszy na komendę KIEDY KLAWISZ... wybieramy z listy DUPLIKUJ i wykonujemy tę czynność 3x. W ten sposób otrzymamy 4 klony bloczków komend. Teraz, klikamy i rozwijamy listę komendy KIEDY KLAWISZ... i wybieramy tam poszczególne klawisze dla pozostałych klonów.

Następnie z odpowiedniej listy dla komendy USTAW KIERUNEK, wybieramy poszczególne kierunki.



5. W kolejnym kroku dodamy samodzielny ruch muchy.
- dodajemy bloczek kiedy kliknięto Zielona Flaga – uruchomienie naszego programu.



- następnie wykorzystamy funkcję Ruch -> obróć o ... stopni. W naszym przypadku wybierzemy 15 stopni. Sprawdź również inne ustawienia.
- kolejno przesuń muchę o 5 kroków i dodaj funkcję odpowiedzialną za odbijanie się od brzegu
- Czy wszystko działa jak należy, mucha się kręci? Niestety nie. Musimy jeszcze dodać funkcję ZAWSZE, którą znajdziemy w zakładce KONTROLA

Poniżej cały program sterowania muchą



## Programowanie - sterowanie żabą

Nasza żaba będzie próbowała złapać muchę. Jeśli ją złapie (dotknie) zmieni kostium i pokaże język.



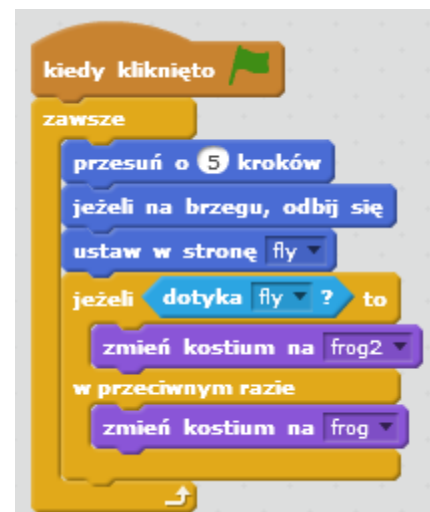


**Podobnie jak poprzednio skrypt ma działać ZAWSZE.**

- w pierwszej kolejności żaba przesuwa się o 5 kroków
- jeżeli jest na brzegu to się odbija
- ustawiamy kierunek ruchu żaby w stronę muchy. USTAW W STRONĘ ....

**Co jeśli żaba złapie muchę?**

- wstawiamy instrukcję warunkową JEŻELI, W PRZECIWNYM RAZIE
- warunkiem zadziałania instrukcji warunkowej jest dotknięcie muchy. Wybieramy funkcję DOTYKA...? Z zakładki CZUJNIKI
- kostium możemy zmienić przy pomocy funkcji WYGLĄD -> ZMIEŃ KOSTIUM NA (żaba z językiem)
- jeżeli żaba nie złapała muchy to wybieramy kostium (żaba bez języka)



**Opcje rozszerzenia gry:**

1. dodanie opcji dźwięku kiedy żaba dotknie muchy
2. zniknięcie muchy po dotknięciu jej przez żabę



## Zajęcia 2: Taniec w kosmosie

### Zadanie:

Wykonanie animacji „Taniec”, która składa się z jednego tańczącego (poruszającego się) duszka.

### Cel:

Wdrożenie się do środowiska programistycznego Scratch. Wyjaśnienie czym jest animacja. Utrwalenie wiedzy dotyczącej poznanych dotąd komend bloków: ZDARZENIA, KONTROLA, WYGLĄD, RUCH, zapoznanie się z blokiem komend: DŹWIĘK. Poznanie PĘTLI ITERACYJNEJ.

### Wprowadzenie:

Prezentujemy dzieciom animację.

Czym jest animacja i jak wygląda proces jej tworzenia? Animacja jest to proces tworzenia ruchomych obrazów.

Jak tworzy się filmy animowane, rysunkowe? Jest to zbiór obrazów, rysunków, różniących się między sobą detalami, wyświetlanych jeden po drugim. W dolnych rogach notesu można narysować „ludzika” różniącego się nieznacząco na kolejnych stronach. Efekt jego ruchu uzyskamy kartkując notes.

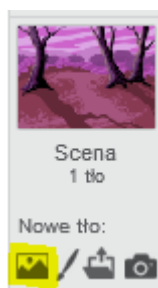
Animacja poklatkowa jest to zbiór zdjęć, gdzie na każdym kolejnym, przekształca się odrobinę obraz (różnica detalu), a następnie wyświetla te zdjęcia w formie filmu.

My zajmiemy się animacją komputerową, ze względu na narzędzie przez nas wykorzystywane.

### Wykonanie:

#### A. Wybór tła oraz duszka.

Wybieramy tło do naszej gry, importując je z biblioteki (pozostawiamy dzieciom dowolność w wyborze).



Proponowane tła z biblioteki: z menu (po lewej stronie) wybieramy kategorie: W POMIESZCZENIU → nazwy tła: SPOTLIGHT-STAGE, STAGE1 i 2, PARTY.

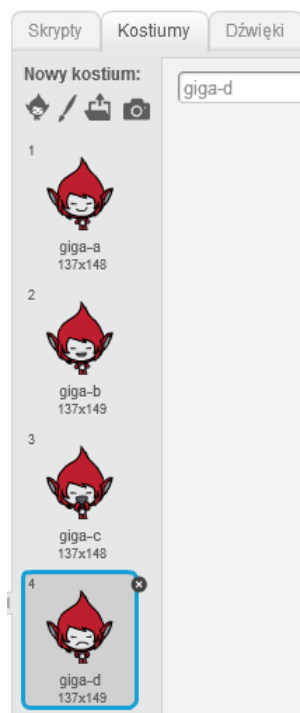
Wybieramy nowego duszka, który dla nas zatańczy.



Proponowane duszki (na ich przykładzie została przygotowana prezentowana animacja): wybieramy MOTYW (menu z lewej strony) → KOSMOS, a następnie duszka GIGA (nie Giga walking), bądź PICO.

Tak zaimportowany duszek jest wyposażony w 4 kostiumy śmiejącego się i smutnego Gigi.





## B. Animacja – intro.

Animacja duszka poprzez zmianę kostiumów.

Tak jak zostało wspomniane, animacja jest to zbiór obrazków wyświetlanych jeden po drugim, różniących się detalami. Dlatego teraz, wyświetlimy śmiejącego się Gigę zmieniając mu po prostu kostiumy.

Do tego celu wykorzystamy blok komend WYGLĄD → ZMIENŃ KOSTIUM NA ... . Aby kostiumy mogły się zmieniać w oknie podglądu duszka, potrzebujemy jeszcze odpowiedniego ZDARZENIA, uruchamiającego naszą pierwszą animację – ZDARZENIA → KIEDY KLIKNIĘTO.

Między komendami zmiany kostiumów, niezbędne jest umieszczenie komendy bloku KONTROLA → CZEKAJ .. s. Jest ona potrzebna, aby użytkownik był w stanie zaobserwować zmianę kostiumów (rozdziela w czasie ustalonym przez użytkownika realizację komend), bez komendy CZEKAJ ... duszek nie poruszyłby się, ponieważ wszystkie komendy wykonałby się w jednej chwili.



Teraz zobaczmy naszą pierwszą animację!

Animacja + monolog duszka.

Na razie Duszek otwiera i zamyka buzię (poprzez zmianę kostiumu). Teraz pora, aby wypowiadał również słowa. Do tego celu wykorzystamy okna dialogowe z bloku WYGLĄD → POWIEDZ ... PRZEZ ... s. Musimy pamiętać, aby nasze okna dialogowe połączyć z kostiumem, w którym Giga ma otwartą buzię (kostium: giga-b) tak, aby wydawało się, że mówi.



## C. Taniec duszka.



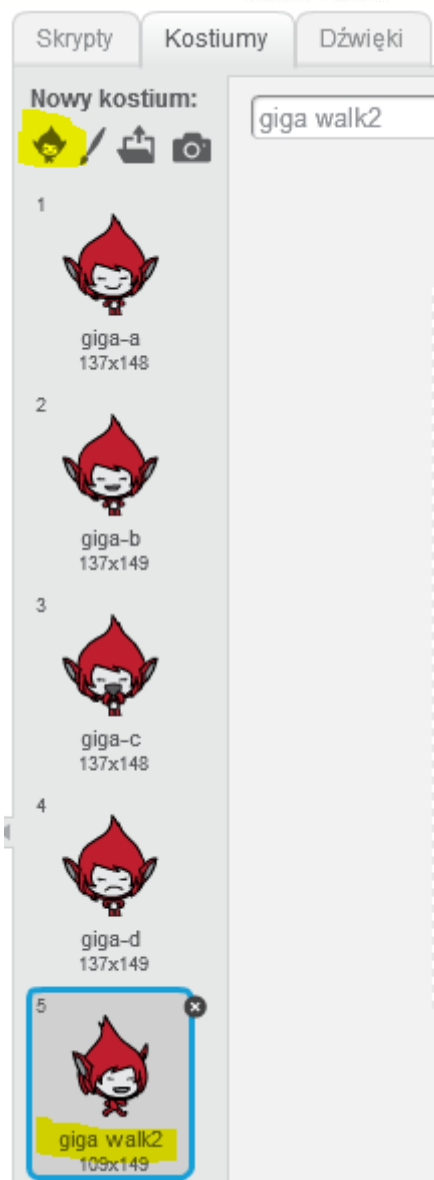
Zajmijmy się teraz samym tańcem naszego duszka. W tym celu zaimportujemy kolejny kostium. Przechodzimy do zakładki KOSTIUMY. Otwieramy bibliotekę, wybieramy kategorię FIKCJA, a z niej GIGA WALK2.

Wracamy do zakładki SKRYPTY. Naszym zadaniem jest teraz wprawienie w ruch naszego duszka. Dlatego z bloku RUCH wybieramy komendę PRZESUŃ O ... KROKÓW.

Aby nasz duszek powtórzył kilkakrotnie swój taneczny ruch, potrzebna nam jest komenda, która wykona PRZESUŃ O ... KROKÓW parokrotnie. Z KONTROLI wybieramy POWTÓRZ ... RAZY – jest to pętla iteracyjna, która w zależności od wskazanej wartości liczbowej, wykona tyle razy komendę umieszczoną we wnętrzu pętli. Aby kroki naszego duszka nie zlewały się w trakcie animacji, w PĘTLI musimy również umieścić komendę POWTÓRZ ... RAZY, aby je rozdzielić.

W zależności od strony (lewa czy prawa), w którą nasz duszek będzie się poruszał, musimy nadać mu kierunek. Dlatego z bloku RUCH, wybieramy USTAW KIERUNEK NA ... . Rozwijamy listę białego pola i wybieramy wartość – prawo (90) bądź lewo (-90).





(Pamiętamy, że przed rozpoczęciem komend poruszających duszka, zmieniliśmy kostium duszka, na nadający się do tańca – giga walk2).

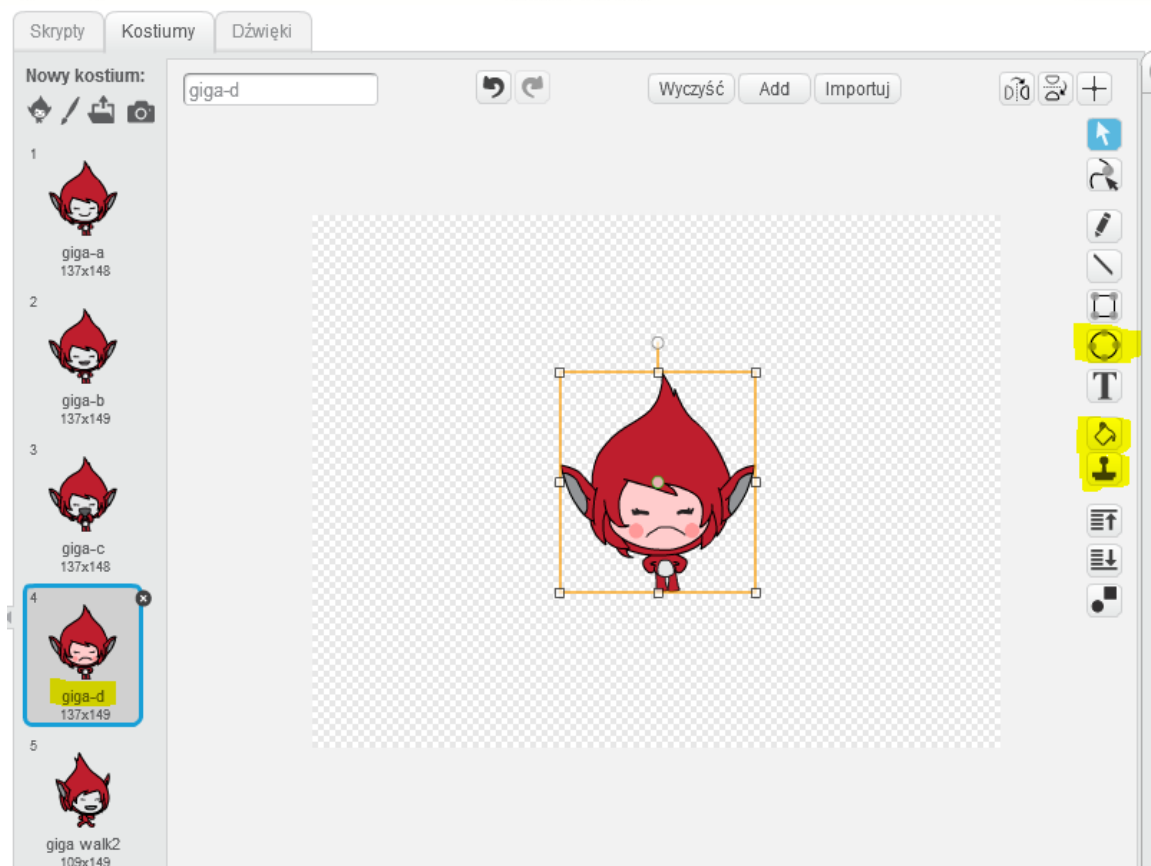
Kolej teraz na kroki taneczne duszka (przód-tył). Jak animować podejście duszka to przodu ekranu i daleko w tył? Bardzo prosto – wystarczy powiększyć i pomniejszyć naszego Gige!

Pamiętamy o zmianie kostiumu na en face (ZMIENŃ KOSTIUM NA giga-b).

Następnie wchodzimy w WYGLĄD → ZMIENŃ ROZMIAR O 10 – w wyniku czego duszek powiększa się, lub → ZMIENŃ ROZMIAR O -10, wtedy duszek pomniejsza się.

Aby animować powiększanie/pomniejszanie duszka w taki sposób, aby wydawało się, że stawia kroki w przód/tył należy umieścić również komendę CZEKAJ .. s (KONTROLA → CZEKAJ .. s)

Komendy ZMIENŃ ROZMIAR O .. oraz CZEKAJ .. s umieszczamy w pętli POWTÓRZ .. RAZY (KONTROLA → POWTÓRZ .. RAZY).

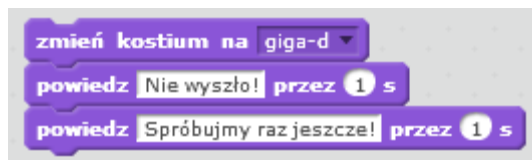


Aby po takich kombinacjach z rozmiarem duszka, zawsze jego kostium wracał do stanu wyjściowego, umieszczamy jako pierwszą komendę bloku WYGLĄD → USTAW ROZMIAR NA 100%.

Zakończymy naszą animację słowami duszka, który informuje nas, że nie udało mu się taniec. W tym celu wykorzystamy nasz jedyny smutny kostium (giga-d), który lekko podrasujemy w naszym edytorze obrazu.

Wchodzimy w zakładkę KOSTIUMY, klikając wybieramy kostium giga-d. W edytorze za pomocą narzędzi wypełnienia, elipsy oraz duplikuj, zawstydzimy naszego duszka.

Teraz zmieniamy kostium naszego duszka WYGLĄD → ZMIENŃ KOSTIUM NA giga-d. Następnie, aby duszek podjął dialog, potrzebujemy WYGLĄD → POWIEDZ „Nie wyszło!” PRZEZ 1 s oraz „Spróbujmy raz jeszcze”.



## Muzyka.

Co to za taniec bez muzyki, dlatego teraz zabawmy się w DJ-ów.

W bloku komend DŹWIĘKI, znajdziemy taką komendę jak: ZAGRAJ bęben 1 PRZEZ 0,25 TAKTÓW. Komenda ta umożliwi nam złożenie przeróżnych instrumentów razem i stworzenie własnej melodii. Otwierając listę białego pola tej komendy, możemy wybrać instrument. Wpisując wartość w kolejne pole, jesteśmy w stanie określić długość danego dźwięku.

Aby muzyczna sekwencja powtórzyła się, umieścimy ją w KONTROLA → POWTÓRZ ... RAZY.



Muzykę włączać będziemy w momencie rozpoczęcia tańca (czyli po zakończeniu jest dialogu duszka), dlatego w tym celu użyjemy ZDZRZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY.





## Zajęcia 3: Labirynt - cz.1

---

### Zadanie:

Wykonanie gry "Labirynt", która składa się na jednego sterowalnego przez użytkownika Duszka.

### Cel:

Pierwsza samodzielna próba powołania do życia sterowalnego duszka. Indywidualne utrwalenie wcześniej poznanych bloków: RUCHU, ZDARZENIA, KONTROLA, dodatkowo WYGLĄD. Zwrócenie uwagi na blok CZUJNIKI. Utrwalenie wiedzy dotyczące PĘTLI NIESKOŃCZONA oraz INSTRUKCJI WARUNKOWEJ.

### Wprowadzenie:

Prezentujemy dzieciom grę.

### Wykonanie:

#### A. Wybór duszka i tła

Importujemy duszka z biblioteki, który będzie głównym bohaterem gry poruszającym się po labiryncie.

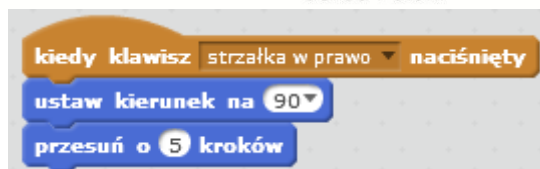


#### B. Programowanie

Zadaniem dzieci jest samodzielne zaprogramowanie duszka w taki sposób, aby móc nim sterować. Omawiamy z dziećmi sposób w jaki zostało to wykonane w przypadku postaci Muchy pochodzącej z gry „Mucha w potrzasku”.

**a.** W pierwszej kolejności korzystamy z komend bloków RUCHU. Dzieci czytają komendy i dobierają je w taki sposób, aby duszek był zdolny do przemieszczania się w konkretnym kierunku (blok RUCHU → PRZESUŃ O 10 KROKÓW oraz USTAW KIERUNEK NA ... - domyślnie ustawiony jest kierunek PRAWO (90) ).

**b.** Aby nasze wybrane komendy działały w zależności od wciskanego klawisza klawiatury przechodzimy do bloku ZDARZENIA. Spośród komend dzieci wybierają właściwą (ZDARZENIA → KIEDY KLAWISZ ... NACIŚNIĘTY). W oknie komendy klikamy na czarny trójkąt i rozwijamy listę klawiszy, a następnie wybieramy → bądź D (w zależności od tego czy będziemy poruszać naszą postacią strzałkami czy klawiszami WSAD).



c. Teraz klikamy prawym przyciskiem myszy na komendę KIEDY KLAWISZ ... NACIŚNIĘTY, rozwijamy listę i wybieramy duplikuj. Powtarzamy to 3 razy otrzymując 4 takie same bloki komend. Następnie w komendach USTAW KIERUNEK NA ... oraz KIEDY KLAWISZ ... NACIŚNIĘTY wybieramy odpowiednie opcje z list, tak aby duszek poruszał się na cztery strony świata.



Uruchamiamy ZIELONĄ FLAGĘ (rozpoczynamy działanie skryptu) i sprawdzamy czy nasz duszek się porusza.

Duszek porusza się, potrafi nawet chodzić po suficie, głową do dołu (czyli obraca się o 360 stopni). Aby poruszał się z lewej do prawej, klikamy na „i” znajdujące się przy Duszku. Po otwarciu nowego menu wybieramy STYL OBROTÓW na LEWO-PRAWO.



## A. Wybór duszka i tła

Następnie wybieramy tło gry. Mamy dwie możliwości: pobranie z pliku (tło przygotowane przez instruktora) bądź narysowanie własnej planszy za pomocą edytora obrazu.





Narysowana przez nas plansza labiryntu musi składać się z dwóch kolorów, które będą rozróżniane przez program. Do tego celu użyjemy takich narzędzi edytora obrazu jak: PĘDZEL, LINIA, WYPEŁNIJ KOLOREM znajdujących się na pasku z lewej strony.

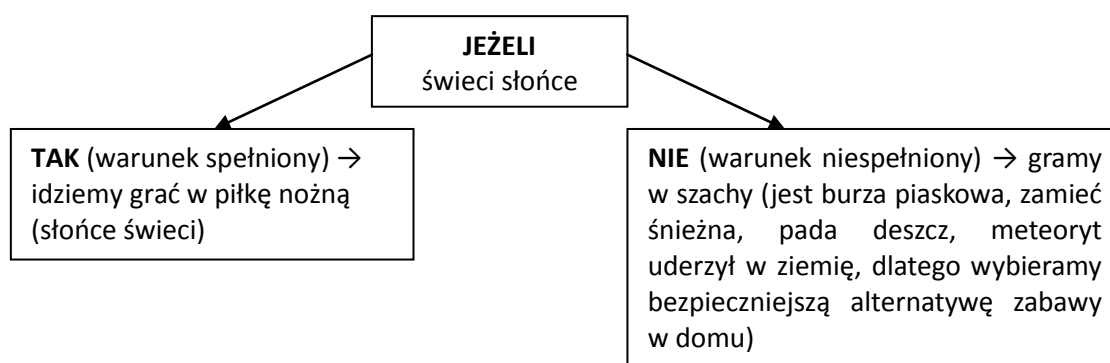
Wybrany przez nas duszek jest za duży i nie mieści się w korytarzach naszego labiryntu. Dlatego korzystając z narzędzia ZMNIEJSZ, dopasowujemy rozmiar postaci do tła. W tym celu klikamy na narzędzie, następnie parokrotnie klikamy na naszego duszka.



## B. Programowanie głównej postaci gry

Nasza postać porusza się po planszy, ale całkowicie ignoruje ściany labiryntu. Dlaczego? Jeszcze nie zaprogramowaliśmy detekcji ścian i dlatego duszek przenika przez ścianę.

W jaki sposób Duszek ma poruszać się po planszy? Tak jak w przykładowym skrypcie. Będzie poruszał się po białych korytarzach, ale nie będzie w stanie poruszać się po czarnym kolorze. Pytamy dzieci czy pamiętają jakiej komendy do tego celu użyć. Wchodzimy w blok KONTROLI, dzieci sprawdzają dostępne komendy i proponują wg nich właściwą. Używamy do tego celu komendy JEŻELI ... TO (blok KONTROLA). Komenda ta, jest to tak zwaną INSTRUKCJĄ WARUNKOWĄ. Jej wykorzystanie pozwoli nam w zależności od tego czy warunek jest spełniony czy nie na wykonanie instrukcji znajdujących się wewnątrz komendy.





INSTRUKCJA WARUNKOWA, której teraz użyjemy (JEŻELI .. TO), różni się nieco od komendy, której używaliśmy wcześniej JEŻELI .. TO W PRZECIWNYM RAZIE. Ta druga, zawiera w sobie dwie instrukcje, które może wykonać. W przypadku gdy warunek jest spełniony (prawdziwy – gramy w piłkę nożną), oraz gdy spełniony nie jest (jest fałszywy – gramy w szachy).

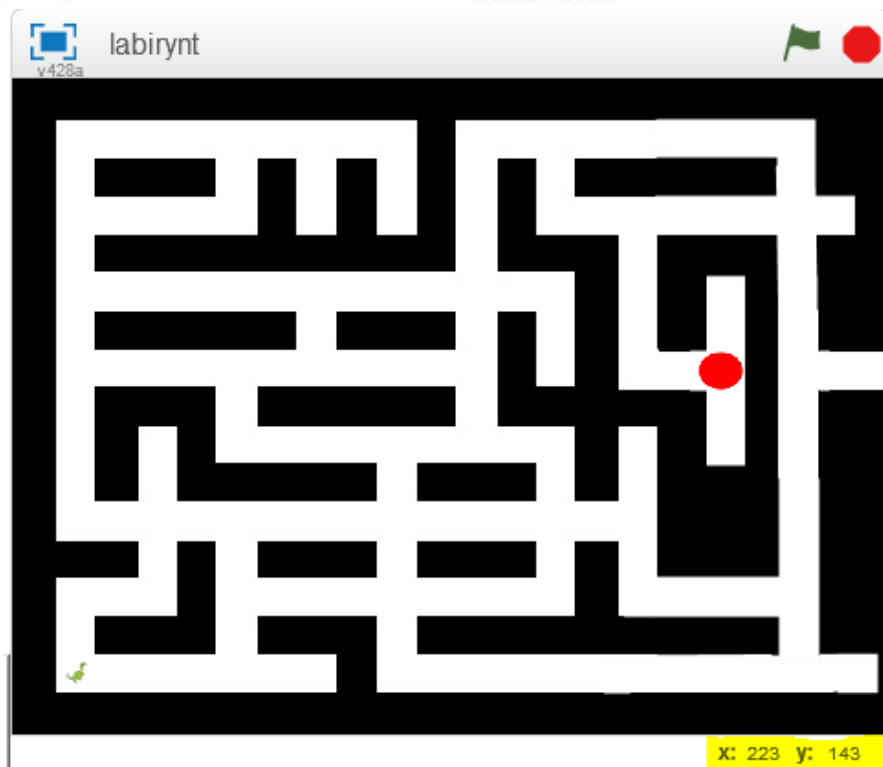
JEŻELI .. TO jest instrukcją, która w przypadku, gdy warunek nie będzie spełniony (fałszywy) nie wykonuje żadnej czynności.

Nasza plansza jest czarno-biała, duszek powinien poruszać się po białych korytarzach, a po czarnym nie. Pytamy dzieci, czy mają pomysł, którą komendę wykorzystamy do tego celu (komenda była używana w trakcie pisania pierwszej gry). Wchodzimy w blok CZUJNIKI i pytamy dzieci, która będzie odpowiednia. Wybieramy DOTYKA KOLORU ... ?. Kolor zmieniamy klikając na kolorowy kwadrat komendy, następnie klikamy kolor tła, na który nasz Duszek ma reagować. Komendę tę umieszczamy w oknie JEŻELI .. TO.

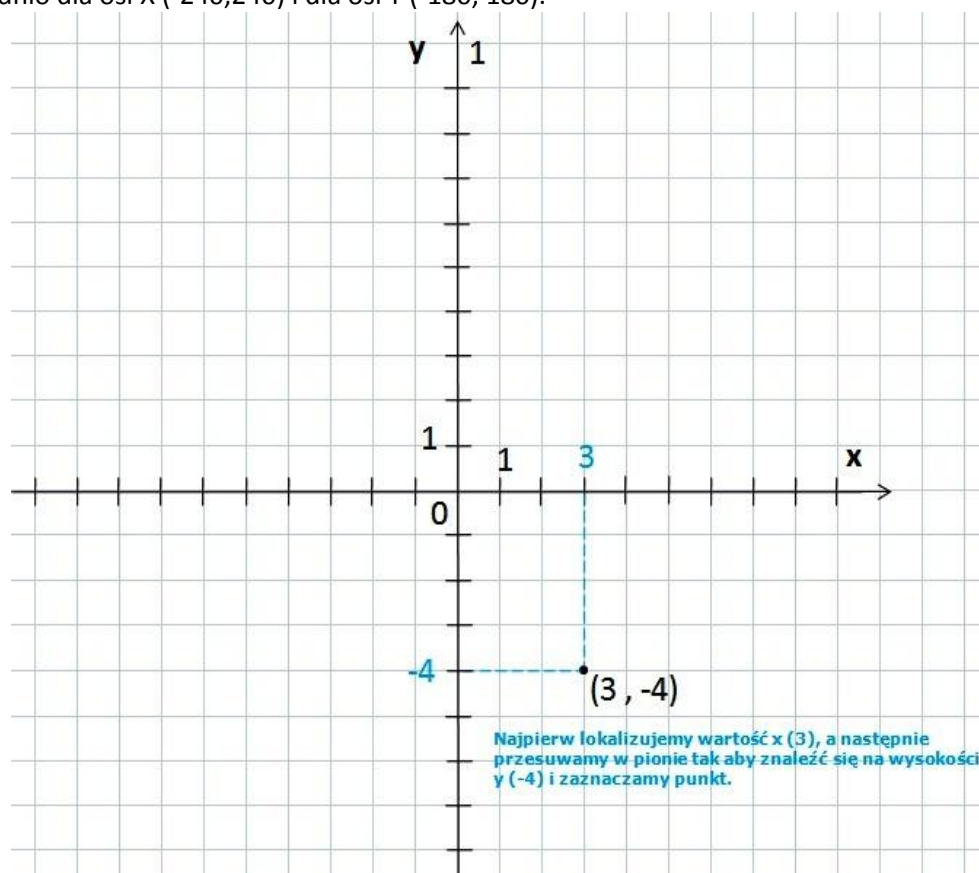
Następnie musimy określić co ma dzieć się w przypadku kontaktu naszego bohatera z czarnym kolorem. Duszek porusza się do przodu o 5 kroków, w momencie gdy dotknie koloru czarnego powinien cofnąć się o 5 kroków (-5). Dzięki temu uzyskamy efekt, w którym duszek dochodząc do ściany labiryntu staje w miejscu i nie może przez nią przejść. Użyjemy blok RUCHU → PRZESUŃ O -5 KROKÓW, którą to komendę umieszczamy wewnątrz INSTRUKCJI WARUNKOWEJ. Następnie INSTRUKCJĘ duplikujemy, tak jak poprzednio, aby otrzymać WARUNEK, który będzie ograniczał duszka, w każdą stronę, w którą się porusza.



Aby nasz bohater gry rozpoczął pokonywanie labiryntu zawsze z określonego miejsca wybieramy RUCH → IDŹ DO X ... Y ... . Określamy tym samy jego początkowe położenie. Zamianę położenia myszki (wartości położenia) można zaobserwować w prawym dolnym rogu podglądu naszej gry.



Rysujemy na tablicy układ i pokazujemy przykładowy punkt znajdujący się w przestrzeni dwuwymiarowej opisanej przez dwie osie współrzędnych. Następnie dzieci przy pomocy współrzędnych kursora myszki określają maksymalne i minimalne wartości naszego układu współrzędnych, odpowiednio dla osi X (-240,240) i dla osi Y (-180, 180).



Ponieważ rozmiar naszego duszka powinien być niezmienny przy każdym uruchomieniu



skryptu ustawiamy jego wielkość używając WYGLĄD → USTAW ROZMIAR NA ... %.

Obie komendy (RUCHU oraz WYGLĄDU) musimy przypisać do odpowiedniego ZDARZENIA, aby program wiedział, w jakim momencie je wykonać. Użyjemy do tego celu ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.

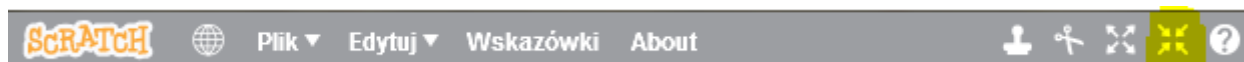
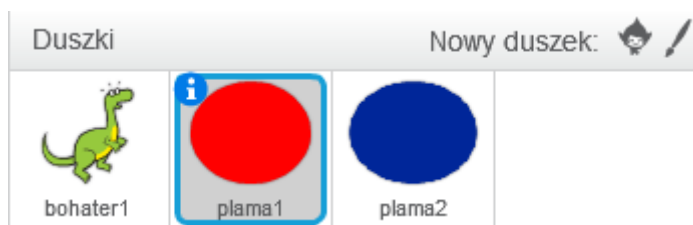


### C. Tworzenie oraz programowanie pozostałych obiektów gry

Gra aby mieć sens, wymaga osiągnięcia jakiegoś celu. Jest tak również w przypadku pokonywania labiryntu. Naszym celem będzie odnalezienie dwóch duszków: plamy niebieskiej oraz plamy czerwonej (w tej kolejności). Tworzymy nasze dwa nowe duszki wybierając NAMALUJ NOWEGO DUSZKA.



Korzystając z narzędzia EDYTORA OBRAZU → ELIPSA tworzymy najpierw jednego duszka, a następnie drugiego, które następnie za pomocą narzędzia ZMNIEJSZ dopasowujemy wielkością do korytarzy labiryntu.



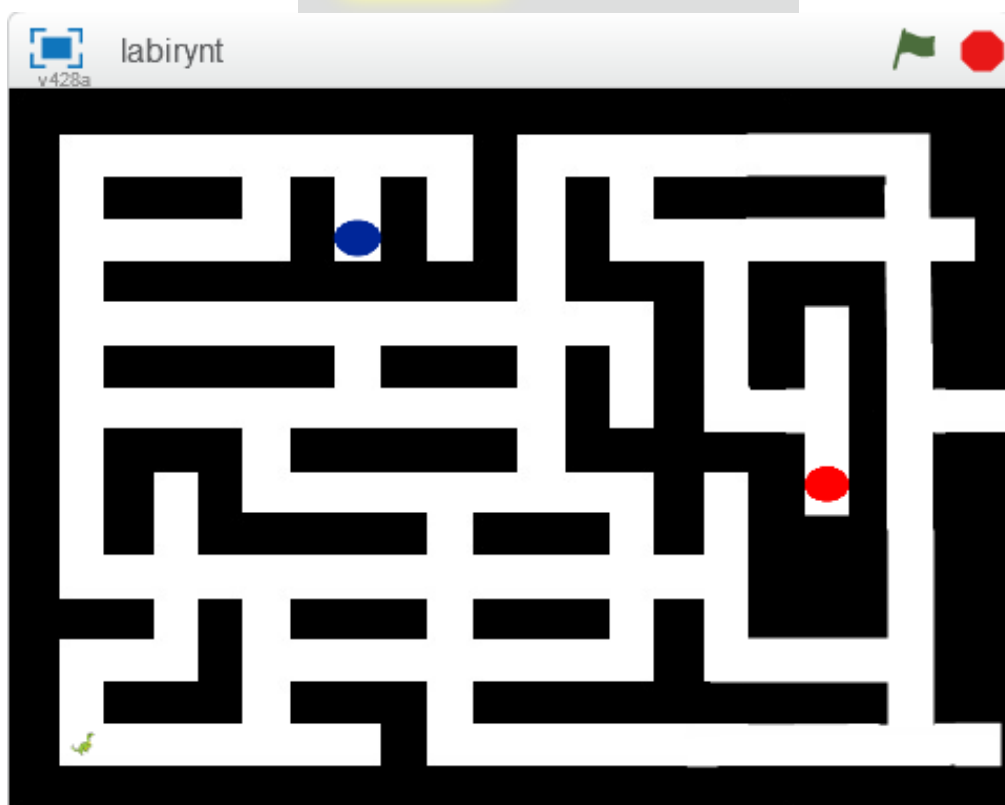
### D. Programowanie plama czerwona

Dotarcie do plamy czerwonej będzie celem naszej gry, które zakończy pierwszą rundę. W pierwszej kolejności umieścimy plamę czerwoną w konkretnym miejscu labiryntu: RUCH → IDŹ DO X ... Y ... .

Następnie, aby nasz Duszek, który zostanie dotknięty przez postać gry, zaczął migać różnymi kolorami użyjemy INSTRUKCJI WARUNKOWEJ. Wybieramy blok KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy warunek z bloku CZUJNIKI → DOTYKA bohater1 ? .



Następnie wewnątrz INSTRUKCJI umieszczamy komendę bloku WYGLĄD → ZMIENŃ EFEKT KOLOR O 25. INSTRUKCJĘ WARUNKOWĄ umieszczamy wewnątrz pętli ZAWSZE (blok KONTROLA), która pozwoli na wykonywanie INSTRUKCJI w nieskończoność, w czasie działania naszego skryptu. Całość przypisujemy pod odpowiednie ZDARZENIE czyli ZDZRZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.



Komenda bloku WYGLĄD → WYCZYŚĆ EFEKTY GRAFICZNE, jest nam niezbędna, aby każdorazowo podczas uruchamianiu skryptu, nasz Duszek odzyskiwał kolor czerwony. Usuwa ona wszelkie zmiany graficzne Duszka, która zostały zapisane przy poprzednim odpaleniu gry.

## E. Programowanie plama niebieska

Dotarcie do plamy niebieskiej jest naszym zadaniem, które musimy wykonać zanim będziemy mogli kierować się i zdobyć plamę czerwoną, która kończy grę.

Duszka programujemy w taki sam sposób, jak plamę czerwoną z dwoma różnicami: efekt



graficzny z jakiego będziemy korzystać to ZMIEN EFEKT JASNOŚĆ O 3 (blok WYGLĄDU).

Również współrzędne plamy niebieskiej będą inne, ponieważ będzie ona znajdować się w innej części labiryntu.



## F. Programowanie głównej postaci gry

Wróćmy jeszcze do głównego bohatera naszej gry. W momencie, w którym dotknie Duszków-plam (KONTROLA → INSTRUKCJA WARUNKOWA), powinien nas o tym poinformować odpowiednim komunikatem (WYGLĄD → POWIEDZ ... PRZEZ ... S). Wybieramy blok KONTROLA → JEŻELI ... TO (INSTRUKCJĘ WARUNKOWĄ) W oknie tej komendy umieścimy komendę, która pozwoli nam wykrywać inne duszki. Blok CZUJNIKI → DOTYKA .. ?

Następnie do środka INSTRUKCJI WARUNKOWEJ umieszczamy okna dialogowe. Blok WYGLĄD → POWIEDZ ... PRZEZ 2 s. Tworzymy dwie identyczne INSTRUKCJE WARUNKOWE dla każdego Duszkaplamy osobno ale z różnymi komunikatami.

Całość (obie instrukcje) umieszczamy w PĘTLI ZAWSZE ponieważ chcemy, aby WARUNKI wykonywały się przez cały cykl życia skryptu.



## Zajęcia 4: Labirynt - cz.2

### Zadanie:

Wykonanie gry „Labirynt”, która polega na pokonaniu labiryntu przy pomocy sterowalnego duszka.



Należy omijać duszki-przeszkody, zbierać duszki-przedmioty i dotrzeć do drzwi. Gra składa się z dwóch poziomów, a celem jest zebrać jak największą ilość punktów.

## Cel:

Utrwalenie wiedzy poznanych dotąd komend bloków: ZDARZENIA, KONTROLA, WYGLĄD, RUCH, CZUJNIKI, WYRAŻENIA, DANE – definicja ZMIENNEJ. Obsługa ZDARZENIA NADAJ wiadomość oraz KIEDY OTRZYMAM wiadomość. Tworzenie i wykorzystywanie INSTRUKCJI WARUNKOWEJ.

## Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy.

UWAGA: Jest to druga część gry, dlatego tworzymy ją, rozbudowując pierwszą część.

## Wykonanie:

### A. Wybór tła oraz duszka

Wybieramy cztery Duszki-przeszkody, oraz trzy Duszki-przedmioty importując je z biblioteki (główny duszek – Duszek-bohater, został zaimportowany w pierwszej części gry).



Do przykładowej gry zostały wykorzystane Duszki-przeszkody zaprezentowane poniżej:



oraz Duszki-przedmioty:



Duszki



Key



Wizard Hat



Magic Wand

Tło czyli PLANASZA 1, zostało przygotowane w poprzedniej części. Teraz narysujmy tło, które będzie PLANASZĄ 2, dla drugiego poziomu gry. Jeżeli nie posiadamy PLANASZA 1, bo brak 1 wersji gry, należy przygotować obie na raz.



Narysowana przez nas plansza labiryntu musi składać się z dwóch kolorów, które będą rozróżniane przez naszą postać. Do tego celu użyjemy takich narzędzi edytora obrazu jak: PĘDZEL, LINIA, WYPEŁNIJ KOLOREM znajdujących się na pasku z lewej strony.



## B. Zmienne

Aby gra miała sens, niezbędny jest jakiś cel. Celem naszej gry jest dotarcie do DRZWI w jak



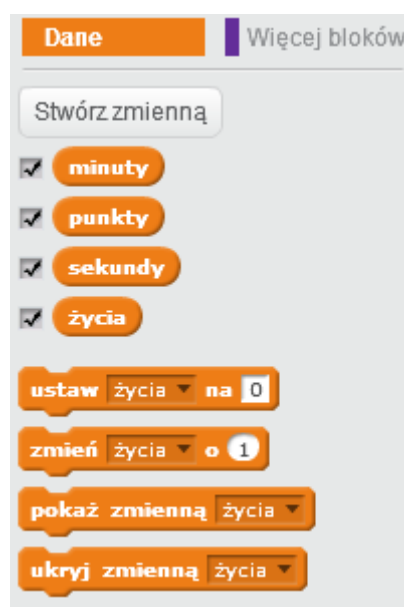


najkrótszym czasie zbierając niezbędne elementy. Ale jak zmierzyć czas pokonania przez nas labiryntu? A co w przypadku jeżeli chcielibyśmy zmierzyć się z kolegą/koleżanką? Jak określić kto z nas jest bardziej zręczny? Potrzebujemy punktów!

Do tego celu niezbędne są ZMIENNE. Są to całe ciągi liter (słowa, skróty), którym przypisujemy wartość liczbową (liczbę), która się zmienia (jest zmienną). Spotkaliśmy się już ze zmiennymi w trakcie ustalania położenia Duszka-bohatera (gra „Labirynt cz. 1”), któremu przypisywaliśmy WSPÓŁRZĘDNE POCZĄTKOWE x oraz y, które zmieniają się za każdym razem, kiedy porusza się duszek. My ZMIENNĄ wykorzystamy do zbierania i wyświetlania punktów, wyświetlania czasu, oraz żyć.

Przechodzimy do bloku DANE, tam deklarujemy cztery zmienne. Klikamy przycisk STWÓRZ ZMIENNĄ → w otwartym oknie wprowadzamy NAZWA ZMIENNEJ: „PUNKTY”, „MINUTY”, „SEKUNDY”, „ŻYCIA”. Wybieramy opcję DLA WSZYSTKICH DUSZKÓW (czyni to zmienną widoczną dla wszystkich duszków), a następnie zatwierdzamy klikając OK.

Razem z nadaniem nazwy zmiennej możemy określić zakres jej widoczności: lokalnie (tylko dla tego duszka) oraz globalnie (dla wszystkich duszków). Zmienna lokalna (tylko dla tego duszka) będzie widoczna (i będzie działać) tylko na liście bloku DANE dla konkretnego duszka, z pozycji pozostałych duszków nie będzie istnieć. Zmienna globalna (dla wszystkich duszków) jest widoczna z pozycji każdego duszka i też dla każdego będzie działać.



Teraz, po deklaracji oraz definicji zmiennej (nadaniu nazwy zmiennej oraz powołaniu jej do istnienia) należy ją zainicjalizować. Inicjalizacja zmiennej to nadanie jej wartości początkowej w naszym przypadku jest to wartość zero (DANE → USTAW ... NA 0).



Zmienne ZAINICJALIZUJEMY dla SCENY. Chcielibyśmy, aby przy każdym rozpoczęciu gry wartości naszych punktów oraz czasu były zerowane, dlatego komendę USTAW ... NA 0 dla wszystkich zmiennych podepnimy pod komendę KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.



Teraz, aby nadać wartości sekundom i minutom w PĘTLI ZAWSZE (blok KONTROLA), będziemy zwiększać sekundy o 1. Czas gry (czyli świata wirtualnego), będzie biegł szybciej niż czas rzeczywisty, dlatego sekundy, będą zmieniać się co 0,2s. Dzięki komendzie CZEKAJ 0,2s skrypt będzie zmieniał sekundy o 1, czekał 0,2s i ponownie zmieniał itd.



W momencie osiągnięcia wartości 60 przez zmienną SEKUNDY, zmienna MINUTY zmieni wartość o 1 (DANE → ZMIENŃ MINUTY O 1), a SEKUNDA zostanie wyzerowana (DANE → USTAW SEKUNDY NA 0). W tym celu użyjemy INSTRUKCJI WARUNKOWEJ blok KONTROLA → JEŻELI .. TO. W oknie komendy umieścimy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI, w którym przyrównujemy zmienną SEKUNDY do wartości 60. Jeżeli warunek jest prawdziwy, to wykonają się komendy wymienione powyżej.



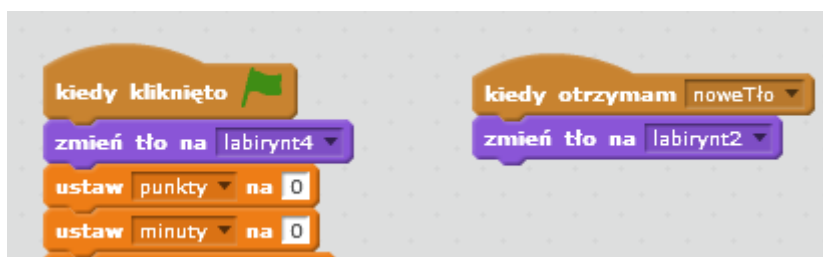
Na potrzeby gry została również stworzona zmienna ŻYCIA. Wyjściowa (początkowa) wartość tej zmiennej to 3, wartość ŻYCIA jest zmniejszana o 1 w momencie dotknięcia Duszków-przeszkód. Gdy zmienna osiągnie wartość 0, wtedy zostanie zatrzymane działanie wszystkich bloków komend, co będzie także skutkowało przegraniem gry (KONTROLA → ZATRZYMAJ WSZYSTKO).





## C. Tło

1. Dla sceny mamy 2 tła, labirynt poziom 1, oraz poziom 2. Musimy ustawić odpowiednie tło w odpowiednim momencie gry. Dlatego tło nr 1 będziemy ustawiać od rozpoczęcia skryptu (KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ), drugie natomiast będziemy ustawiać po przejściu przez Duszka-bohatera DRZWI.
2. Aby zsynchronizować (działać równocześnie, koordynować w czasie) ze sobą działania na duszkach, potrzebujemy ZDARZENIA, które będzie informacją dla wszystkich obiektów gry (Duszków oraz tła), że to jest moment, w którym rozpoczynamy drugi poziom gry, a odpowiednie komendy mają rozpocząć działanie. Wcześniej do tego celu używaliśmy takich zdarzeń jak: KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ, KIEDY KLAWISZ ... NACIŚNIĘTY, KIEDY DUSZEK KLIKNIĘTY.
3. W tym celu wykorzystamy możliwość komunikacji między obiektami gry. Nadamy wiadomość, którą będą mogły odebrać wszystkie Duszki i tła. Blok ZDARZENIA → NADAJ MESSAGE1.
4. Klikamy na czarny trójkąt komendy, rozwijamy listę i wybieramy opcję NOWY KOMUNIKAT. Wyświetla się okno NAZWA WIADOMOŚCI, wpisujemy tam nazwę (np. „noweTło”) i zatwierdzamy klikając OK. Nazwa komunikatu przez nas stworzona pojawia się w oknie komendy.
5. Odwrócimy trochę kolejność i najpierw użyjemy komendy KIEDY OTRZYMAM NOWETŁO, zanim jeszcze nawet ją nadaliśmy. Jest nam to potrzebne do zmiany tła dla drugiego poziomu.



Pierwszy poziom gry rozpoczyna się wraz ze ZDARZENIEM KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Drugi poziom ma początek wraz z przejściem przez Duszka-bohatera DRZWI czyli nadaniu KOMUNIKATU NOWETŁO.

## D. Duszki-przedmioty

**Duszki przedmioty są to elementy gry, które musimy „zebrać”, czyli dotrzeć do nich w zakamarkach labiryntu, aby zdobyć dodatkowe punkty.**

### Klucz

1. Będzie widoczny na obu poziomach gry. Zarówno od momentu ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ oraz ZDARZENIA → KIEDY OTRZYMAM NOWETŁO, musimy nadać mu położenie czyli współrzędne. Blok RUCH → IDŹ DO x ... y ... . Efekt „znikania” KLUCZA uzyskamy używając WYGLĄD → ZMIENŃ EFEKT JASNOŚĆ O 3. Nasz efekt graficzny umieścimy w INSTRUKCJI WARUNKOWEJ, tak aby Duszek „znikał” będąc dotykany przez Duszka-bohatera. Blok KONTROLA → JEŻELI ... TO.
2. Musimy stworzyć WARUNEK naszej instrukcji, dlatego w oknie komendy umieszczamy



CZUJNIKI → DOTYKA GHOST1? Będziemy ten warunek wykonywać ZAWSZE czyli w PĘTLI ZAWSZE jeżeli KLUCZ będzie dotykany przez naszego Duszka-bohatera.

3. Jeżeli wykorzystujemy efekty graficzne, musimy pamiętać o tym, aby je wyczyścić, żeby Duszek mógł odzyskać swój pierwotny wygląd. Dlatego pod oba ZDARZENIA podpinamy WYCZYŚĆ EFEKTY GRAFICZNE.



### Czapka Czarodzieja

1. Czapka Czarodzieja będzie widoczna tylko na pierwszym poziomie gry, czyli od momentu KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Pod powyższe ZDARZENIE podpinamy blok WYGLĄD → POKAŻ.
2. W drugiej części gry będzie niewidoczna, dlatego wybieramy ZDARZENIE od którego rozpoczyna się drugi poziom gry ZDARZENIA → KIEDY OTRZYMAM NOWETŁO. Pod powyższe ZDARZENIE podpinamy blok WYGLĄD → UKRYJ.
3. Pozostałe komendy dla Duszka są identyczne jak w przypadku KLUCZA. Dzieci samodzielnie (nie podglądając) próbują napisać ten sam skrypt.

### Magiczna różdżka

1. Magiczna różdżka nie będzie elementem pierwszego poziomu gry. Pojawi się dopiero w drugim poziomie. Od momentu KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ, nie będzie widoczna. Pod powyższe ZDARZENIE podpinamy blok WYGLĄD → UKRYJ.
2. W drugiej części gry pojawi się, dlatego wybieramy ZDARZENIE od którego rozpoczyna się drugi poziom ZDARZENIA → KIEDY OTRZYMAM NOWETŁO. Pod powyższe ZDARZENIE podpinamy blok WYGLĄD → POKAŻ.
3. Pozostałe komendy dla Duszka są identyczne jak w przypadku KLUCZA. Dzieci samodzielnie (nie podglądając) próbują napisać ten sam skrypt.

### Drzwi

1. Drzwi występują na obu poziomach gry – są naszym celem, przejściem z jednego poziomu do drugiego. Do naszego obszaru skryptu przeciągamy oba ZDARZENIA gry, KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ oraz KIEDY OTRZYMAM NOWETŁO. Komendy umieszczone poniżej równolegle podpinamy pod oba zdarzenia bądź po skończeniu jednego bloku DUPLIKUJEMY go.
2. W zależności od poziomu drzwi będą znajdować się w różnych miejscach, dlatego wartości dla współrzędnych x i y będą różne. Blok RUCH → IDŹ DO x .. i y .. . Aby wyróżnić moment, w którym gracz dotrze do drzwi rozświecimy je migającymi kolorami. Uzyskamy ten efekt dzięki WYGLĄD → ZMIENŃ EFEKT KOLOR O 25. Nasz efekt graficzny umieścimy w INSTRUKCJI WARUNKOWEJ, tak aby Duszek „świecił” będąc dotykany przez Duszka-bohatera. Blok KONTROLA → JEŻELI .. TO. Musimy stworzyć WARUNEK naszej instrukcji, dlatego w oknie



- komendy umieszczamy CZUJNIKI → DOTYKA GHOST1? Będziemy ten warunek wykonywać ZAWSZE czyli w PĘTLI ZAWSZE, jeżeli DRZWI będą dotykane przez naszego Duszka-bohatera.
- Jeżeli wykorzystujemy efekty graficzne, musimy pamiętać o tym, aby je wyczyścić, żeby Dusek mógł odzyskać swój pierwotny wygląd.. Dlatego pod oba ZDARZENIA podpinamy WYCZYŚĆ EFEKTY GRAFICZNE.

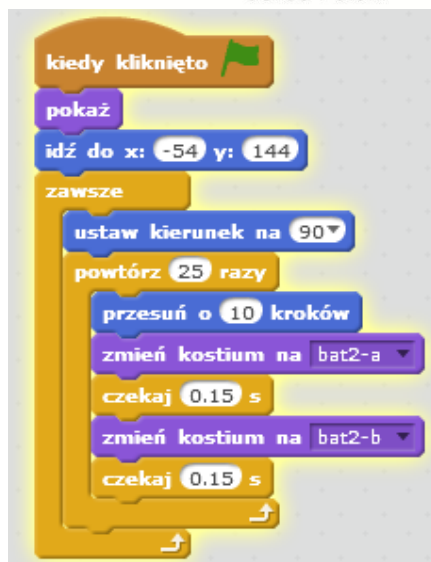


## E. Duszki-przeszkody

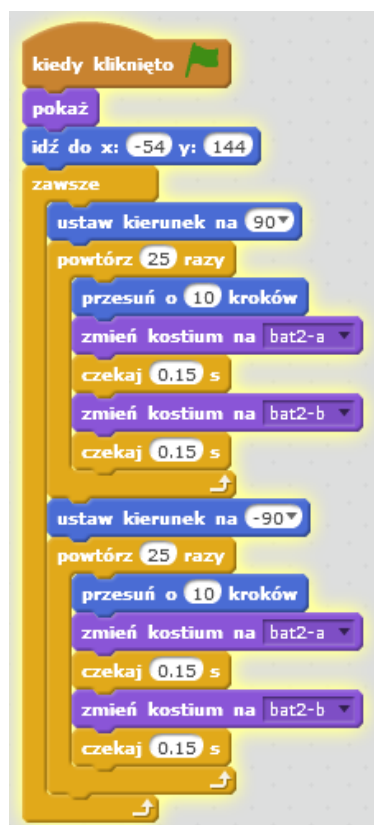
Są to postaci gry, której utrudniają nam przejście gry. W momencie zetknięcia się z Duszkiem-przeszkodą jesteśmy cofani na start i odejmowane jest nam 1 życie z puli trzech. Po trzecim razie umieramy i kończymy grę.

## Granatowy nietoperz

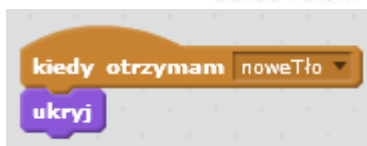
- Występuje tylko na pierwszym poziomie gry, na drugim jest już niewidoczny. Dlatego skrypt dla niego, będzie znajdował się pod ZDARZENIEM KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie, aby pojawił się od początku gry, wybieramy WYGLĄD → POKAŻ.
- Musimy nadać mu współrzędne początkowe, aby zawsze zaczynał latać od tego samego miejsca, wybieramy blok RUCH → IDŹ DO x ... y ... . Ustawiamy kierunek lotu naszego Duszka w prawo, czyli blok RUCH → USTAW KIERUNEK NA 90.
- Teraz, aby stworzyć animację będziemy powtarzać 25 razy sekwencję komend. Do tego celu wykorzystamy PĘTLĘ LICZNIKOWĄ (ITERACYJNĄ). Blok KONTROLA → POWTÓRZ 25 RAZY. Dusek musi się przemieszczać po labiryncie, dlatego, aby wprawić go w ruch, użyjemy komendy PRZESUŃ O 10 KROKÓW.
- Efekt latania uzyskamy dzięki możliwości animacji machania skrzydłami. Użyjemy do tego celu dwóch kostiumów nietoperza: z podniesionymi oraz opuszczonymi skrzydłami. Blok WYGLĄD → ZMIEŃ KOSTIUM NA BAT2-A oraz ZMIEŃ KOSTIUM NA BAT2-B. Zmianę kostiumu z jednego na drugi opóźnimy umieszczając między nimi komendę CZEKAJ 0,15s.



Tak samo piszemy skrypt dla lotu nietoperza w lewo, zmieniając tylko komendę USTAW KIERUNEK NA -90 i doczepiamy do PĘTLI ITERACYJNEJ już istniejącej.



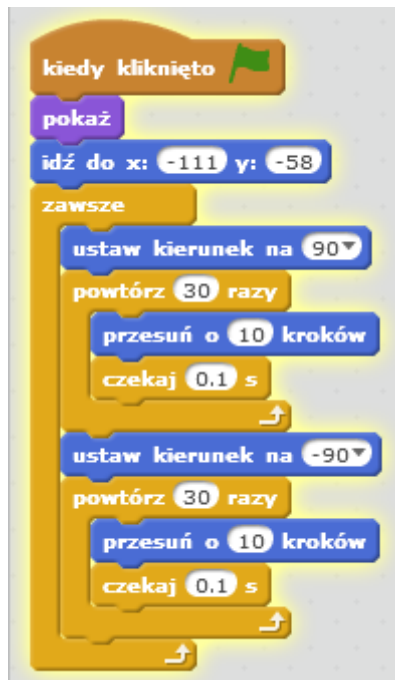
Nietoperz ma być niewidoczny na drugim poziomie gry, dlatego pod zdarzeniem KIEDY OTRZYMAŁO NOWEŁO, podpinamy WYGLĄD → UKRYJ.



## Duch w kapturze

Duch w kapturze, tak samo jak GRANATOWY NIETOPERZ, będzie widoczny na pierwszym poziomie gry, ale na drugim już nie.

1. Dlatego skrypt dla niego, będzie znajdował się pod ZDARZENIEM KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.
2. Następnie, aby pojawił się od początku gry, wybieramy WYGLĄD → POKAŻ. Musimy nadać mu współrzędne początkowe, aby zawsze zaczynał latać od tego samego miejsca. Blok RUCH → IDŹ DO x ... y ... .
3. Ustawiamy kierunek lotu naszego Duszka w prawo, czyli blok RUCH → USTAW KIERUNEK NA 90.
4. Aby stworzyć animację latania będziemy powtarzać 30 razy sekwencję komend. Do tego celu wykorzystamy PĘTLĘ LICZNIKOWĄ (ITERACYJNĄ). Blok KONTROLA → POWTÓRZ 30 RAZY.
5. Duszek musi się przemieszczać po labiryncie, dlatego, aby wprawić go w ruch, użyjemy komendy PRZESUŃ O 10 KROKÓW. Aby nie latał za szybko wykorzystamy komendę CZEKAJ 0,1s.
6. Tak samo piszemy skrypt dla lotu Duszka w lewo, zmieniając tylko komendę USTAW KIERUNEK NA -90 i doczepiamy do PĘTLI ITERACYJNEJ już istniejącej.



7. W tym samym czasie, kiedy duszek się porusza (przez cały czas trwania skryptu) ma być sprawdzany WARUNEK, który pozwoli w momencie zetknięcia się z Duszkiem-bohaterem na zmianę kostiumu. Pod zdarzeniem KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ, umieszczamy PĘTLĘ ZAWSZE, a w niej INSTRUKCJĘ WARUNKOWĄ: JEŻELI ... TO. W oknie komendy warunkowej

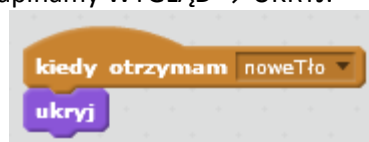




umieszczamy blok CZUJNIKI → DOTYKA GHOST1 ? Jeżeli warunek jest prawdziwy będziemy zmieniać kostium Duszka. Dlatego wewnątrz INSTRUKCJI, umieszczamy komendę WYGLĄD → ZMIENŃ KOSTIUM NA GHOUL-B oraz ZMIENŃ KOSTIUM NA GHOUL-A. Aby wersja „goniąca” kostiumu (kostium b), przez chwilę była widoczna, zmianę kostiumu z jednego na drugi musimy rozdzielić komendą CZEKAJ 1s.



8. Duszek ma stać się niewidoczny na drugim poziomie gry, dlatego pod zdarzeniem KIEDY OTRZYMAM NOWETŁO podpinamy WYGLĄD → UKRYJ.



### Brązowy nietoperz

1. Nie występuje na pierwszym poziomie gry. Jest widoczny od drugiego etapu.
2. Porusza się po labiryncie, dzięki temu samemu skryptowi co GRANATOWY NIETOPERZ.

Różnica występuję w ZDARZENIACH:

Od momentu KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ ---- UKRYJ Duszka.

KIEDY OTRZYMAM NOWETŁO ---- POKAŻ Duszka.

Zmienia się nazwa KOSTIUMU.

Duszek znajduję się w innym miejscu labiryntu, czyli zmieniają się jest współrzędne (wartości x .. i y .. są inne)

Dzieci samodzielnie piszą skrypt dla Duszka.

### Duch

1. Nie występuje na pierwszym poziomie gry. Jest widoczny od drugiego etapu.
2. Porusza się po labiryncie, dzięki temu samemu skryptowi co DUCH W KAPTURZE.

Różnica występuję w ZDARZENIACH:

Od momentu KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ ---- UKRYJ Duszka.

KIEDY OTRZYMAM NOWETŁO ---- POKAŻ Duszka.

Zmienia się nazwa KOSTIUMU.

Duszek znajduję się w innym miejscu labiryntu, czyli zmieniają się jest współrzędne (wartości x .. i y .. są inne)

Dzieci samodzielnie piszą skrypt dla Duszka.

## E. Skrypt Głównego Duszka

Funkcję sterowalności Duszkiem napisaliśmy jeszcze w pierwszej części gry. („Labirynt cz. 1”). Dlatego teraz zajmiemy się kontaktem Duszka-bohatera z pozostałymi Duszkami.



1. Główny Duszek ma występować na obu poziomach gry, dlatego w kodzie wykorzystamy oba ZDARZENIA: KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ oraz KIEDY OTRZYMAM NOWETŁO.
2. Musimy nadać Duszkowi taki rozmiar, aby mieścił się naszych białych korytarzach. Dlatego, blok WYGLĄD → USTAW ROZMIAR NA 15%.
3. Tworzymy INSTRUKCJĘ WARUNKOWĄ, która pozwoli nam na zdobywanie punktów, w momencie „podniesienia” przedmiotów. Blok KONTROLA → JEŻELI .. TO. Komendę umieszczamy w PĘTLI ZAWSZE.
4. W oknie WARUNKU umieszczamy WYRAŻENIE LOGICZNE LUB. Komenda ta, posiada dwa okna, rozdzielone wyrazem LUB pozwala to, w trakcie wykonywania skryptu na wybór jednej z opcji. Wystarczy, że jedna jest PRAWDZIWA (SPEŁNIONA), a wykonają się komendy znajdujące się we wnętrzu INSTRUKCJI WARUNKOWEJ.
5. W oknach WYRAŻENIA LUB umieszczamy CZUJNIKI → DOTYKA KEY? I kolejne WYRAŻENIE LUB wewnątrz okien którego, umieszczamy CZUJNIKI → DOTYKA MAGIC WAND? I DOTYKA WIZARD HAT?.
6. Wewnątrz INSTRUKCJI wkładamy DANE → ZMIENŃ PUNKTY O 10.
7. Aby punkty zbyt szybko nie przyrastały, po uzyskaniu 10 punktów poczekajmy 1,5s z naliczaniem kolejnych. Dlatego do środka INSTRUKCJI wkładamy CZEKAJ 1,5s.
8. W taki sam sposób tworzymy INSTRUKCJĘ WARUNKOWĄ wraz z WARUNKIEM dla przypadku, w którym Duszek-bohater dotyka DRZWI.
9. Wewnątrz INSTRUKCJI, czyli po przejściu każdego poziomu, będą podliczane punkty. W tym celu użyjemy komendy DANE → ZMIENŃ PUNKTY O ..
10. W oknie powyższej komendy umieścimy WYRAŻENIE ze ZNAKIEM DZIELNIA, gdzie umieścimy ZMIENNĄ SEKUNDY / 2. Następnie przy pomocy okna dialogowego wypiszemy wartość zdobytych punktów. Blok WYGLĄD → POWIEDZ .. PRZEZ 2s.
11. We wnętrzu okna komendy POWIEDZ, umieścimy WYRAŻENIE POŁĄCZ, które pozwoli na pojawienie się na ekranie (wypisanie) strumienia słów oraz zmiennej. W momencie pierwszego przejścia przez DRZWI, zostanie nadany KOMUNIKAT, NADAJ NOWETŁO.
12. Ostatnim elementem gry są Duszki-przeszkody, które w momencie dotknięcia cofają gracza na początek labiryntu. Dlatego tworzymy kolejną INSTRUKCJĘ WARUNKOWEJ → JEŻELI .. TO. W oknie komendy umieszczamy WYRAŻENIE LUB, w oknie którego, umieszczamy CZUJNIKI → DOTYKA ... ? Z listy komendy DOTYKA, wybieramy po kolei wszystkie Duszki-przeszkody. We wnętrzu INSTRUKCJI, wkładamy komendę RUCH → IDŹ DO x ... y .., która będzie nas cofać na początek labiryntu. We wnętrzu INSTRUKCJI umieszczamy również odjęcie jednego ŻYCIA, za dotknięcie Duszka-przeszkody, czyli DANE → ZMIENŃ ŻYCIE O -1.



```
when clicked
  go to x: -205 y: -146
  set size to 15 %
  change costume to ghost1

when receive newTlo
  go to x: -130 y: -145
  set size to 15 %
  change costume to ghost2

always
  if any key pressed or Wizard Hat pressed or Magic Wand pressed
    change points to 10
    wait 1.5 s

  if door pressed
    change points to seconds / 2
    say [Zdobyleś i punkty i punktów] przez 2 s
    give newTlo

  if Bat2 pressed or Ghoul pressed or Bat1 pressed or Ghoul2 pressed
    go to x: -205 y: -146
    change lives to -1
```



## Zajęcia 5: Memo

---

### Zadanie:

Wykonanie gry pamięciowej „Memo”, której celem jest rozwijanie u dziecka koncentracji, spostrzegawczości oraz ćwiczenie pamięci.

### Cel:

Utrwalenie wiedzy poznanych dotąd komend bloków: ZDARZENIA, KONTROLA, WYGLĄD, CZUJNIKI, WYRAŻENIA, oraz zapoznanie się z nowymi komendami bloku DANE czyli wprowadzenie definicji ZMIENNEJ. Przypomnienie tworzenia i wykorzystywania INSTRUKCJI WARUNKOWEJ.

### Komentarze:

W zależności od wieku oraz poziomu wiedzy dzieci, można całkowicie pominąć skrypty SCENY oraz wykorzystane w grze DŹWIĘKI, które będą traktowane jako rozszerzenie. Liczba stworzonych kart zależy wyłącznie od upodobań Instruktora.

Ze względu na odtwarzane DŹWIĘKI, w trakcie odstawiania kart, włączany po raz pierwszy skrypt będzie się zaciął, natomiast kolejne uruchomienia będą przebiegać bez przeszkód.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy.

Gra MEMO, dobrze znana dzieciom, polega na odnalezieniu i połączeniu w pary, dwóch takich samych obrazków (kart).

### Wykonanie:

#### A. Powołanie do życia duszków oraz tła gry.

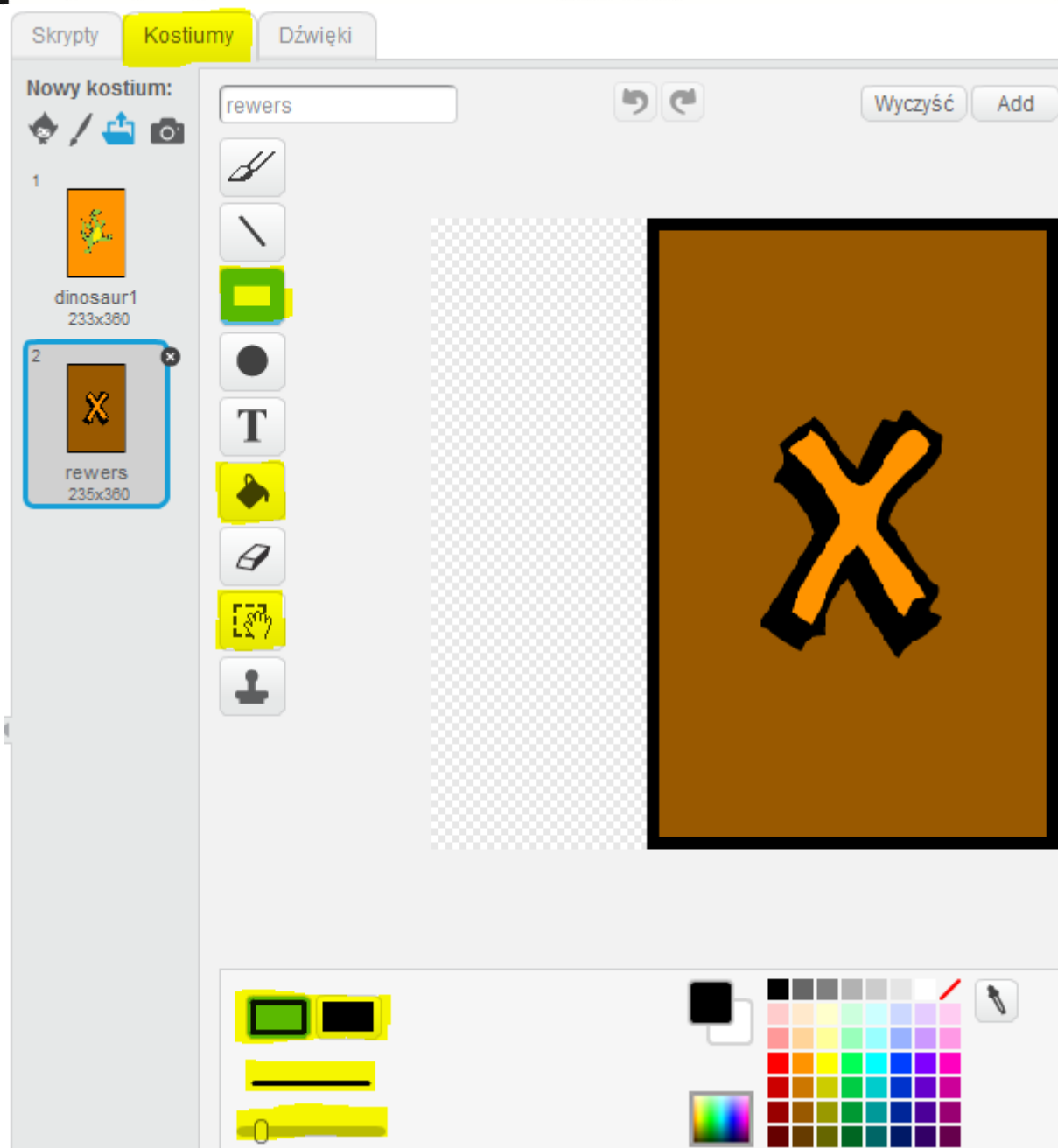
Początkowo tworzymy 4 Duszki, które samodzielnie rysujemy. Będą one zawierać po 2 kostiumy, awers czyli obrazek znajdujący się na karcie oraz rewers, grzbiet karty (wspólny dla całej talii), zakrywający obrazek znajdujący się na niej.

W pierwszej kolejności tworzymy kostium rewersu talii, tak aby swobodnie przekopiować go następnie do kolejnych Duszków. Importujemy z BIBLIOTEKI Duszka, który będzie znajdował się na grzbiecie naszej talii.



Wybieramy ZAKŁADKĘ KOSTIUMY, a następnie przy pomocy Edytora Obrazu dorysujemy kartę do gry. W tym celu wykorzystamy takie narzędzia jak:

- PROSTOKĄT bez wypełnienia/z wypełnieniem,
- WYPEŁNIJ KOLOREM,
- WYBIERZ.



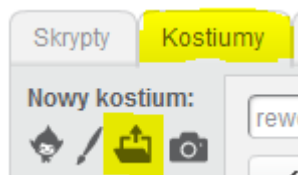
UWAGA: Ze względu na „humory” scratcha zapisujemy każdego narysowanego Duszka w folderze na pulpicie, tak aby nie stracić naszej pracy (nazwa folderu: imie\_nazwisko\_Scratch), tam też zapisujemy nasze projekty.

Następnie klikamy prawym przyciskiem myszy na narysowany kostium i z rozwiniętej listy wybieramy ZAPISZ JAKO PLIK LOKALNY.



Kolejnym krokiem jest stworzenie awersu karty, dlatego powtarzamy kroki 2 i 3.

Następnie tworzymy pozostałe 3 duszki rysując ich awersy. Natomiast rewers (zapisany wcześniej w naszym folderze) importujemy z pliku przy pomocy ikony WGRAJ KOSTIUM Z PLIKU.



Układamy nasze 4 karty w górnym rzędzie, dopasowując ich rozmiar do ekranu. Kolejnym krokiem jest zaimportowanie z BIBLIOTEKI bądź samodzielne narysowanie TŁA SCENY. W tym celu, z ikon znajdujących się po lewej stronie, pod Podglądem Duszków, wybieramy NOWE TŁO → WYBIERZ TŁO Z BIBLIOTEKI albo NAMALUJ NOWE TŁO.



UWAGA: Wersja dłuższa: Duszki-Karty mogą zostać zdublikowane zaraz po ich narysowaniu, dzięki czemu uzyskamy 4 pary kart (Duszków), które będzie trzeba wypełnić skryptem. Wersja krótsza: duplikujemy Duszki dopiero po zapisaniu skryptu.

## B. Zmienne.

Gra MEMO polega na dobraniu w pary kart, które rozłożone są grzbietami do góry. Po odkryciu kart, które nie należą do pary, należy je z powrotem zakryć zapamiętując miejsca, w których dane karty się znajdują. Przyda nam się to w trakcie kolejnych tur (odkryć).

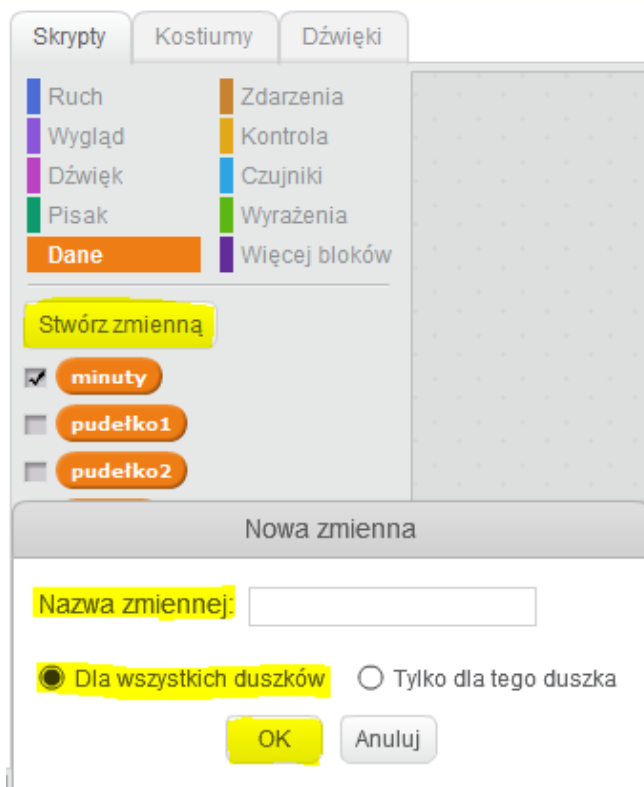
Pytamy dzieci w jaki sposób karty należące do pary będą rozpoznawane przez skrypt? Komputer nie rozpoznaje obrazków znajdujących się na kartach bo nie ma oczu, ani Sztucznej Inteligencji.

Tak samo jak w szkole systematyzujemy dzięki numeracji, sale, szafki, wieszaki w szatni, tak samo w tym przypadku żeby powiązać ze sobą parę tych samych kart nadamy im te same numery. Wykonamy to przy pomocy dwóch pudełek – ZMIENNYCH, które w zależności od klikniętej karty będą przechowywać w danej chwili pojedynczą cyfrę od 1 do 4. Następnie zawartość pudełek (ZMIENNYCH) będziemy ze sobą porównywać czy jest jednakowa.

Czym są ZMIENNE? Są to litery alfabetu, bądź całe ciągi liter (słowa, skróty), którym przypisujemy wartość liczbową, która się zmienia (jest zmienna).

Przechodzimy do bloku DANE, tam deklarujemy dwie zmienne. Klikamy przycisk STWÓRZ ZMIENNĄ → w otwartym oknie wprowadzamy NAZWA ZMIENNEJ: „pudełko1”, „pudełko2”. Wybieramy opcję DLA WSZYSTKICH DUSZKÓW (czyni to zmienną widoczną dla wszystkich duszków), a następnie zatwierdzamy klikając OK.

Razem z nadaniem nazwy zmiennej możemy określić zakres jej widoczności: lokalnie (tylko dla tego duszka) oraz globalnie (dla wszystkich duszków). Zmienna lokalna (tylko dla tego duszka) będzie widoczna oraz będzie działać tylko na liście bloku DANE dla konkretnego duszka, natomiast z poziomu pozostałych duszków nie będzie istnieć. Zmienna globalna (dla wszystkich duszków) jest widoczna z poziomu każdego duszka i też dla każdego będzie działać.

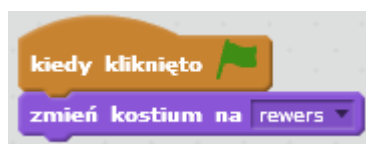


Po deklaracji oraz definicji zmiennej (nadaniu nazwy zmiennej oraz powołaniu jej do istnienia) należy ją zainicjalizować. Inicjalizacja zmiennej to nadanie jej wartości początkowej. W naszym przypadku, nie będziemy inicjalizować ZMIENNYCH, ponieważ będziemy im każdorazowo podczas kliknięcia kolejnego Duszka-Karty nadawać/przypisywać nową wartość (DANE → USTAW ... NA 0).

### C. Karty.

Karty zostały ułożone równo, w rzędach na obszarze Podglądu Duszków. W pierwszym krokiem będzie zakrycie kart w momencie uruchomienia skryptu czyli kliknięcia ZIELONEJ FLAGI. Pytamy dzieci jak to zrobić.

Wybieramy z bloku ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Zdarzenia są to akcje generowane przez użytkownika naszego programu. Zdarzeniami są: kliknięcie przycisku myszy, użycie klawiatury. Naszym zadaniem jest obsługa konkretnego zdarzenia, czyli reakcja skryptu na akcję (działanie) użytkownika. Następnie podpinamy z bloku WYGLĄD → ZMIENŃ KOSTIUM NA REWERS.



Sprawdzamy czy wszystkie karty obróciły się grzbietem do góry.

Jeżeli rozmiar kart nie pozwala na swobodne rozstawienie ich w obszarze Podglądu Duszków, należy użyć komendy z bloku WYGLĄD → USTAW ROZMIAR NA ... %. Pozwoli to na ustawienie jednego rozmiaru dla wszystkich Kart.

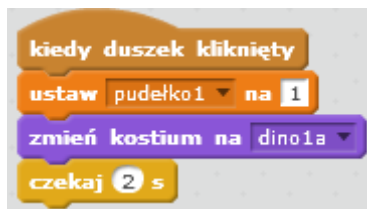
Karty zostały zakryte. Teraz chcielibyśmy, aby Gracz po kliknięciu na wybraną kartę odsłonił ją.





Zanim jednak to zrobi, nadamy karcie numer tak, żeby skrypt rozpoznawał każdą z czterech kart. Dzięki przypisanej przy kliknięciu cyfrze będzie porównywać ze sobą dwie kliknięte karty.

W tym celu wybieramy z bloku ZDARZENIA → KIEDY DUSZEK KLIKNIĘTY. Następnie podpinamy z bloku DANE → USTAW PUDEŁKO1 NA 1. Odkrycie karty zrealizujemy przy pomocy komendy z bloku WYGLĄD → ZMIENŃ KOSTIUM NA Dino1a. W dalszym ciągu podepnimy sobie z bloku KONTROLA → CZEKAJ 2s, tak żeby umożliwić Graczowi zapamiętanie obrazka znajdującego się na karcie.



Powtarzamy kroki 1-7 dla pozostałych trzech kart.

UWAGA: W zależności od poziomu wiedzy grupy oraz czasu duplikujemy zapisany skrypt do pozostałych trzech Kart lub prosimy dzieci o samodzielne napisanie skryptu dla pozostałych kart posiłkując się już stworzonym. Sprawdzamy czy skrypt działa dla wszystkich kart!

UWAGA: W zależności od upodobań Instruktora należy ZDUPLIKOWAĆ każdego DUSZKA w tym miejscu, bądź ZDUPLIKOWAĆ go później z w pełni napisanym skryptem, w którym wprowadzone będą zamiany dotyczące nazwy zmiennych oraz kostiumów tak, żeby uzyskać 8 Duszków-Kart.



Karty są już ponumerowane (pytamy dzieci: Jaki jest kolejny krok, aby gra MEMO została stworzona w pełni?). Naszym następnym zadaniem będzie stworzenie takiego fragmentu skryptu, który będzie porównywał dwie kliknięte po sobie karty (dokładniej ich numery) i sprawdzał czy należą do tej samej pary.

W tym celu wykorzystamy dwie stworzenie przez nas ZMIENNE: PUDEŁKO1 i PUDEŁKO2, których wartości będą ze sobą porównywane. Zrealizujemy to przy pomocy INSTRUKCJI WARUNKOWEJ.

Wybieramy z bloku KONTROLA → JEŻELI ... TO W PRZECIWNYM RAZIE W oknie INSTRUKCJI WARUNKOWEJ umieszczamy z bloku WYRAŻENIA → ... = ... . W lewym oknie ZNAKU RÓWNOŚCI wkładamy ZAMIENNĄ z bloku DANE → PUDEŁKO1, natomiast w prawym oknie ZMIENNĘ PUDEŁKO2.

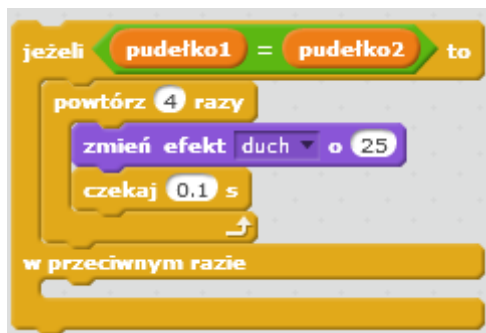
W momencie, gdy powyższy WARUNEK będzie prawdziwy (spełniony), czyli obie ZMIENNE będą



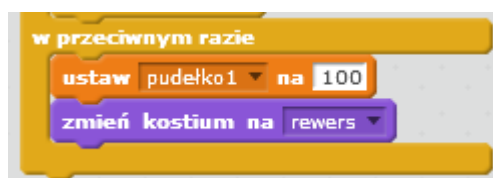
posiadały tę samą wartość, karty znikną (pytamy dzieci: W jaki sposób możemy to zrealizować? Przeglądamy blok WYGLĄD).

W tym celu skorzystamy z bloku WYGLĄD → ZMIENŃ EFEKT DUCH O 25. Czy pozwoli to na zniknięcie karty? Sprawdzamy klikając bezpośrednio na komendę – karta zrobiła się transparentna (przezroczysta).

Całkowite zniknięcie karty zrealizujemy zmieniając wartość EFEKTU DUCH o 100, jak to zrobić? Nie wpisujemy wartości 100 do komendy, ponieważ spowoduje ona natychmiastowe zniknięcie karty, zależy nam na spowolnieniu tego procesu. Zrealizujemy to przy pomocy PĘTLI ITERACYJNEJ z bloku KONTROLA → POWTÓRZ 4 RAZY, dzięki temu osiągniemy pożądaną wartość. Sprawimy, że efekt powolnego znikania będzie zauważalny dodając komendę z bloku KONTROLA → CZEKAJ 0.1s.



W przypadku gdy WARUNEK nie zostanie zrealizowany, zmienimy wartość naszej ZMIENNEJ na wartość, której nie posiada żadna inna karta tak, żeby skrypt nie rozpoznał par kart tam gdzie ich nie ma. W przykładzie wybrane zostały wysokie wartości 100 i 200. Cel ten, zrealizujemy przy pomocy komendy z bloku DANE → USTAW PUDEŁKO1 NA 100. Umieszczamy ją we wnętrzu PĘTLI w części drugiej (warunek nie został spełniony). Następnie zakryjemy kartę, używając do tego z bloku WYGLĄD → ZMIENŃ KOSTIUM NA REWERS.



## ROZSZERZENIE 1:

### A1. Scena i Czas

Rozbudujemy nasz skrypt o TŁO STARTOWE zapraszające Gracza do rozpoczęcia gry oraz TŁO kończące grę.

Wybieramy SCENA → ZAKŁADKA TŁA → NAMALUJ NOWE TŁO. Utworzymy dwie ZMIENNE, które będą odpowiadały za pomiar czasu, w którym Graczowi udało się ukończyć grę. Wybieramy blok DANE → STWÓRZ ZMIENNĄ: SEKUNDY i MINUTY.

ZAINICJALIZUJEMY ZMIENNĄ (PRZYPISEM jej wartość) w momencie włączenia skryptu czyli kliknięcia ZIELONEJ FLAGI. W tym celu wybieramy blok ZDARZENIA → KIEDY KLINIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy z bloku DANE → USTAW SEKUNDY NA 0 oraz USTAW MINUTY NA 0. Podczas



uruchomienia skryptu pojawi się TŁO STARTOWE, natomiast ZMIENNE będą niewidoczne. Wybieramy z bloku DANE → UKRYJ ZMIENNĄ SEKUNDY oraz UKRYJ ZMIENNĄ MINUTY. Następnie podpinamy z bloku WYGLĄD → ZMIEN TŁO NA START.



Grę rozpoczniemy naciskając klawisz spację. W tym celu wybierzmy z bloku ZDARZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY?

Następnie zmieniamy TŁO na tło gry dlatego wybieramy z bloku WYGLĄD → ZMIEN TŁO NA STARS. W dalszym kroku pokażemy nasz zegar, czyli dwie zmienne odliczające kolejno minuty oraz sekundy. W tym celu wybieramy z bloku DANE → POKAŻ ZMIENNĄ SEKUNDY oraz POKAŻ ZMIENNĄ MINUTY.

W kolejnym kroku zaprogramujemy zegar, tak żeby odliczał czas w trakcie gry aż do jej skończenia. Zrealizujemy to przy pomocy PĘTLI NIESKOŃCZONEJ, dlatego wybieramy z bloku KONTROLA → ZAWSZE.

W dalszym ciągu będziemy INKREMENTOWAĆ (zwiększać wartość) ZMIENNEJ o 1, która będzie odbywać się co ćwierć sekundy. Dzięki temu, minuta w naszym wirtualnym świecie gry będzie trwała (pytamy dzieci: ile?) 15 sekund. We wnętrzu PĘTLI umieszczamy z bloku DANE → ZMIEN SEKUNDY O 1. Następnie wybieramy z bloku KONTROLA → CZEKAJ 0.25s.

Kolejnym krokiem jest przeliczanie sekund na minuty po upływie 59 sekund, co zrealizujemy przy pomocy INSTRUKCJI WARUNKOWEJ. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie INSTRUKCJI umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewe okno ... = ... wkładamy z bloku DANE → ZMIENNĄ SEKUNDY, w prawe natomiast, wpisujemy wartość 59.

We wnętrzu INSTRUKCJI przypiszemy ZMIENNEJ SEKUNDY wartość -1, tak żeby w momencie upływu minuty ZMIENNA przyjęła wartość 0. W tym celu wybieramy z bloku DANE → USTAW SEKUNDY NA -1. Następnie wybieramy z bloku DANE → ZMIEN MINUTY O 1.





## B2. Karty

Podczas uruchomienia skryptu pojawia się TŁO startowe, natomiast Karty muszą być wtedy niewidoczne. W tym celu do komend znajdujących się pod ZDARZENIEM – KIEDY KLIKNIĘTO ZIEŁONĄ FLAGĘ, bezpośrednio pod nim, podpinamy z bloku WYGLĄD → UKRYJ.



Następnie, kiedy Gracz naciskając klawisz spacji uruchomi grę, karty muszą się pokazać. Zrealizujemy to przy pomocy komendy z bloku ZDARZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY. Następnie podpinamy z bloku WYGLĄD → POKAŻ.



## A1. Scena

TŁO kończące grę i informujące Gracza o zwycięstwie będzie się pojawiać w momencie kiedy zostaną zliczone cztery odnalezione pary. Policzymy je przy pomocy ZMIENNEJ. W pierwszej kolejności należy ZMIENNĄ powołać do życia, dlatego blok DANE → STWÓRZ ZMIENNĄ → LiczbaPar. Następnie podpinamy INICJALIZACJĘ tej ZMIENNEJ do pozostałych, w SKRYPCIE SCENY. W tym celu wybieramy z bloku DANE → USTAW LiczbaPar NA 0.



## B2. Karty

Zmienna została zainicjalizowana żeby mogła realizować swoje zadanie czyli zliczać odnalezione pary. Należy rozszerzyć skrypt Kart o jedną komendę (pytamy dzieci: Jaką?). Wybieramy z bloku DANE → ZMIEN LICZBAPAR O 1, którą umieszczamy wewnątrz INSTRUKCJI WARUNKOWEJ, sprawdzającej czy odkryte Karty należą do pary.



## A1. Scena

Naszym ostatnim zadaniem będzie opisanie w SKRYPCIE chwili, w której Gracz wygrywa (na niebie pojawiają się fajerwerki i słychać rajska muzykę).

Zrealizujemy to przy pomocy kolejnej INSTRUKCJI WARUNKOWEJ, która pozwoli nam na sprawdzenie czy 4 pary zostały już znalezione czy też jeszcze nie. W tym celu wybieramy komendę z bloku KONTROLA → JEŻELI ... TO. W oknie INSTRUKCJI umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewe okno ... = ... wkładamy z bloku DANE → ZMIENNĄ LICZBAPAR, w prawe natomiast, wpisujemy cyfrę 4. We wnętrzu INSTRUKCJI umieszczamy z bloku KONTROLA → CZEKAJ 1s. Następnie wybieramy z bloku WYGLĄD → ZMIENŲ TŁO NA KONIEC.

Kolejnym krokiem jest włączenie radosnej melodii, które pozwoli, żeby splendor zwycięstwa mógł powoli spływać na Triumfatora. Następnie wybieramy z bloku KONTROLA → CZEKAJ 7.4s (czas trwania melodii).

W ostatnim kroku zakończymy działanie skryptu czyli całą grę. Zrealizujemy to przy pomocy komendy z bloku KONTROLA → ZATRZYMAJ WSZYSTKO.



## Rozszerzenie 2

(w przypadku grupy, która posiada niezbędną wiedzę oraz umiejętności posługiwania się ZMIENNYMI)

Nie ma przeszkód, żeby karty rozkładać w obrębie obszaru Podglądu Duszków przy pomocy komendy bloku RUCH;

losować ich położenie przy pomocy komendy bloku WYRAŻENIA oraz rozbudować grę o kolejne rundy w których:

- czas odsłonięcia kart skraca się,
- liczba kart wzrasta.



## Zajęcia 6: Generator gwiazd

---

### Zadanie:

Wykonanie gry „Generator gwiazd”, która dzieli się na:

- ustalenie poziomu trudności gry
- gra czyli klonowanie duszków
- zmiennie
- wyniki

### Cel:

Dzieci próbują samodzielnie stworzyć zaprezentowaną im grę.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy.

Projektujemy architekturę gry czyli wspólnie z dziećmi wypisujemy na tablicy jej etapy. Ten projekt ma na celu pozwolić dzieciom na planowanie swojej pracy oraz szukaniu własnych rozwiązań.

### Wykonanie:

#### A. Przewodnik po grze.

Importujemy bądź rysujemy swojego duszka-przewodnika. Duszek od momentu rozpoczęcia skryptu ma być widoczny, dlatego używamy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ, pod które to zdarzenie podpinamy WYGLĄD → POKAŻ.

Aby umożliwić użytkownikowi kontrolę nad poziomem trudności gry, użyjemy komendy CZUJNIKI → ZAPYTAJ ... I CZEKAJ (skrypt stanie się interaktywny). Jest to procedura pobierania danych z klawiatury czyli funkcja wejścia danych. W naszym przypadku użytkownik będzie wczytywał dane w postaci liczb całkowitych, które następnie będą przypisywane do zmiennej ODPOWIEDŹ.

W oknie komendy ZAPYTAJ .. I CZEKAJ wpisujemy „Ile?”. Tworzymy zmienną „ILE”, która będzie określać maksymalną ilość gwiazd, która może pojawić się nam na ekranie. Wybieramy DANE → STWÓRZ ZMIENNĄ. Nadajemy nazwę zmiennej i akceptujemy przyciskiem OK.

Następnie pod zadane powyżej pytanie podpinamy DANE → USTAW ILE NA ... . W oknie komendy umieszczamy zmienną CZUJNIKI → ODPOWIEDŹ, do której została przypisana wartość wprowadzona przez użytkownika.

Powtarzamy CZUJNIKI → ZADAJ ... I CZEKAJ tym razem dla poziomu trudności gry, który jest określony, przez ilość sekund w trakcie których gwiazdy mają być wyświetlane. Tworzymy zmienną CZAS, której przypiszemy wartość drugiej odpowiedzi. Wybieramy DANE → USTAW CZAS NA ..., a w oknie komendy umieszczamy CZUJNIKI → ODPOWIEDŹ.

Po ustaleniu parametrów gry możemy ją rozpocząć. Aby pozostałe bloki komend wiedziały, że czas rozpocząć działanie stworzymy odpowiednie do tego celu zdarzenie. Wchodzimy w blok ZDARZENIA → NADAJ ..., a następnie tworzymy NOWY KOMUNIKAT. Wpisujemy nazwę wiadomości i akceptujemy przyciskiem OK.



Teraz duszka-przewodnika gry należy ukryć. Powinno się to wykonać w momencie zakończenia ustalania poziomu trudności gry. Dlatego wybieramy ZDARZENIA → KIEDY OTRZYMAM START. Następnie pod powyższe zdarzenie podpinamy WYGLĄD → UKRYJ.

## B. Gwiazdy

Nasza gra polega na generowaniu i zliczaniu gwiazd, które pojawią się na niebie. Chcąc uniknąć tworzenia 1000 jednakowych duszków (co byłoby odrobinę pracochłonne), wykorzystamy funkcję klonuj, która pozwoli nam na stworzenie dużej ilości takich samych duszków (1000 obiektów zawiesi program).

Klonowanie ma rozpocząć się w momencie zakończenia wprowadzania danych przez użytkownika, dlatego też blok komend rozpocznie działanie od ZDARZENIA → KIEDY OTRZYMAM START. Aby urzeczywistnić klonowanie wybieramy KONTROLA → SKLONUJ SIEBIE. Następnie tworzymy zmienną WYNIKI, która umożliwi nam zliczenie klonów. DANE → ZMIENŃ WYNIKI O 1, którą to komendę podpinamy pod funkcję klonowania (INKREMENTUJEMY – zwiększamy o 1 zmienną WYNIKI).

Powyższe komendy umieszczamy w PĘTLI ITERACYJNEJ, KONTROLA → POWTÓRZ ... RAZY. Za każdym przejściem PĘTLI (wykonaniem się tego co jest w środku) powstanie jeden klon oraz zmienna WYNIKI zwiększy swoją wartość o 1. Nie chcemy za każdym razem uzyskiwać tej samej ilości gwiazd (klonów) na niebie. Ustala to użytkownik podczas podawania wartości zmiennej ILE (ma to miejsce przy pytaniu "Ile?") i jest to maksymalna ilość gwiazd, które pojawią się na niebie. Liczbę gwiazd generujemy losowo, dlatego wybieramy WYRAŻENIA → LOSUJ OD ... DO ... . Umieszczamy komendę w oknie PĘTLI ITERACYJNEJ.

Minimalna możliwa liczba gwiazd, która może zostać wylosowana to 0. Natomiast maksymalną ich liczbę określa zmienna ILE, którą też umieszczamy wewnątrz komendy LOSUJ OD 0 DO ILE.



Czy teraz gwiazdy nam się pokażą? Sprawdzamy to uruchamiając skrypt. Niestety po powołaniu klonów do życia nie określiliśmy co ma się z nimi dzieć



Wybieramy KONTROLA → KIEDY ZACZYNAM JAKO KLON. Teraz musimy pokazać klony duszka, którego ukryliśmy podczas rozpoczęciu skryptu. Pod powyższą komendę podpinamy WYGLĄD → POKAŻ.

Teraz czas nadać różne położenia naszym gwiazdom, dlatego będziemy losować wartości ich współrzędnych. Wybieramy RUCH → IDŹ DO X: ... Y: ... a w oknach komendy umieszczamy WYRAŻENIA → LOSUJ OD ... DO ... . Następnie ustalamy zakresy współrzędnych tak, aby równały się wartości wysokości i szerokości PODGLĄDU DUSZKÓW.

Kolejnym krokiem jest wyświetlanie gwiazd przez konkretny odcinek czasu, który na początku określił Użytkownik. Wybieramy KONTROLA → CZEKAJ ... s, a w oknie komendy umieszczamy zmienną CZAS.

Następnie usuwamy klony tak, aby po określonym czasie gwiazdy zniknęły. Blok KONTROLA → USUŃ TEGO KLONA.



### C. Zakończenie gry

Ostatnia faza gry, czyli sprawdzenie spostrzegawczości Gracza. Teraz musimy porównać ilość gwiazd zaobserwowanych z ilością faktycznie wygenerowaną przez skrypt.

Mamy już wybrane zdarzenie KIEDY OTRZYMAM START, pod którym znajduje się podpięta komenda UKRYJ. Będziemy działać właśnie w tym bloku.

Aby zsynchronizować zniknięcie gwiazd wraz z zadaniem pytania o ilość zaobserwowanych wybieramy komendę CZEKAJ ... s. We wnętrzu okna umieszczamy WYRAŻENIA → ... + ... . Będziemy dodawać do zmiennej CZAS + 1s.

Kolejnym krokiem jest pokazanie naszego duszka-przewodnika, który został ukryty po fazie wstępnej gry poprzez blok WYGLĄD → POKAŻ. Wypisujemy na ekran pytanie CZUJNIKI → ZAPYTAJ "Ile zliczyłeś gwiazd na niebie?" I CZEKAJ.

Następnie musimy porównać wartość wprowadzoną przez Użytkownika z rzeczywistą liczbą wygenerowanych gwiazd. Do tego celu wykorzystamy INSTRUKCJĘ WARUNKOWĄ, KONTROLA ->JEŻELI ... TO, W PRZECIWNYM RAZIE. W oknie INSTRUKCJI umieszczamy znak równości, czyli blok WYRAŻENIA → ... = ... . Porównujemy ze sobą wartość wprowadzoną przez Użytkownika czyli CZUJNIKI → ODPOWIEDŹ ze zmienna WYNIKI.

Jeżeli warunek INSTRUKCJI został spełniony (jest prawdziwy), to wypisujemy komunikat "Dobrze!", za pomocą okna dialogowego WYGLĄD → POWIEDZ .. PRZEZ 2 s. W przeciwnym wypadku (jeżeli warunek jest fałszywy) zmienna ODPOWIEDŹ nie jest równa zmiennej WYNIK, to Użytkownik błędnie policzył ilość gwiazd. Dlatego wypisujemy komunikat "Źle!" za pomocą WYGLĄD → POWIEDZ ... PRZEZ 2 s.







## Zajęcia 7: Brak Internetu

---

### Zadanie:

**Stworzenie bardzo prostej graficznie gry, z wykorzystaniem elementów animacji.**

Gra jest odwzorowaniem dobrze znanej nam gry, którą oferuje nam Google Chrome w momencie, gdy nasz komputer jest niezdolny do połączenia się z Internetem.

### Cel:

Przypomnienie idei programowania równoległego oraz wykorzystanie podstawowych technik animacji.

### Wprowadzenie:

Gra jest odwzorowaniem dobrze nam znanej gry, którą oferuje Google Chrome w momencie, gdy nasz komputer jest niezdolny do połączenia się z Internetem.

Prezentujemy dzieciom grę, początkowo nie zdradzając jej mechanizmu. Zadajemy pytania o zastosowane bloków komend oraz zabiegi programistyczne.

### Komentarze:

W zależności od upodobań Instruktora, stworzenie wszystkich Duszków może odbyć się na początku, bądź też naprzemiennie czyli tworzymy jednego Duszka, a zaraz potem jego skrypt. Jeżeli na ekranie PODLGĄDU DUSZKÓW nie chcemy widzieć części Duszków klikamy na nie prawym przyciskiem i z listy wybieramy UKRYJ.

Pamiętamy, że skrypt piszemy razem z dziećmi, nie przepisując go a wymyślając razem oraz czytając komendy odpowiednich bloków. Pomagamy dzieciom nakierowując je wskazówkami.

### Wykonanie:

#### A. Powołanie do życia duszków oraz tła gry.

Do stworzenia naszej gry potrzebujemy:

- Duszka pokonującego przeszkody,
- przeszkody,
- linię wyznaczającą grunt,
- trzy cyfry odmierzające czas,
- napis GAME OVER.

UWAGA: Duszki mogą zostać samodzielnie stworzone przez dzieci przy pomocy Edytora Obrazu (odwzorowując postacie wykorzystane w przykładowej grze, bądź według własnego pomysłu) albo importując je z biblioteki.

Bohater naszej gry (w przykładzie Dinozaur) będzie wyposażony w cztery kostiumy (które rysujemy bądź importujemy):

- pierwszy kostium jest to strona startowa, która pojawi się po uruchomieniu skryptu, czyli w momencie kliknięcia ZIELONEJ FLAGI.
- dwa kolejne kostiumy są to dinozaury. Na pierwszym Dino ma podniesioną do góry lewą nogę, a na drugim prawą.
- ostatni kostium jest to zaskoczony dinozaur z wybałuszonymi oczami, który wpadł na przeszkodę.



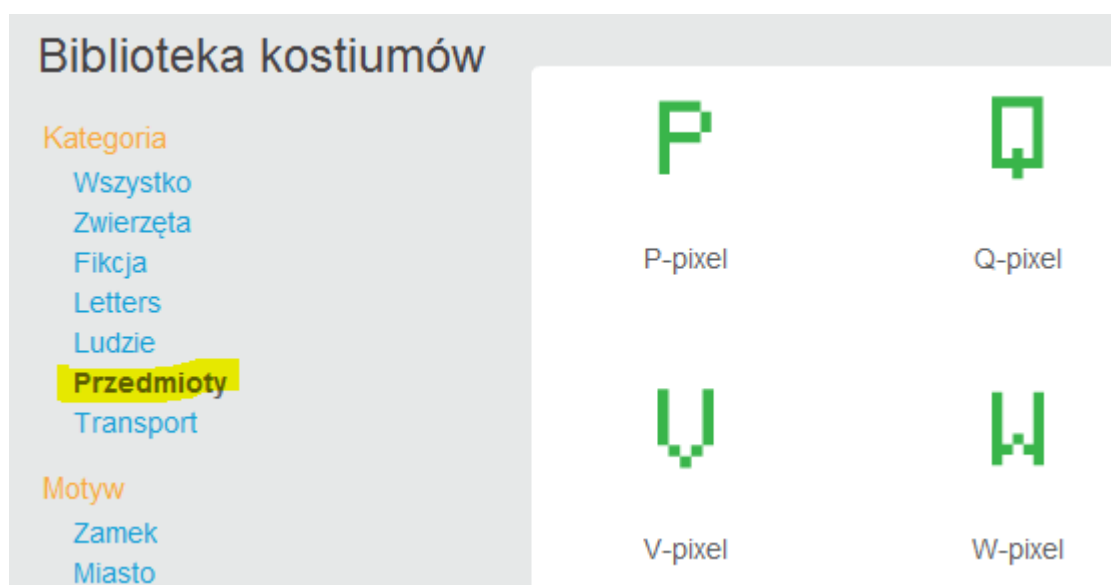
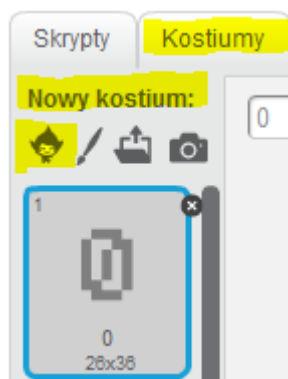
Duszki-przeszkody, czyli kaktusy będą wyposażone w cztery kostiumy (które rysujemy bądź importujemy), różniące się wielkością oraz ilością roślin.



Duszki-cyfry, każda z trzech cyfr będzie zawierała do 9 kostiumów, którymi będą cyfry od 0-9. Duszki importujemy z BIBLIOTEKI, pamiętając, że musimy znajdować się wtedy w zakładce KOSTIUMY.



Z menu znajdującego się po lewej stronie wybieramy kategorię PRZEDMIOTY, a następnie importujemy kostiumy.

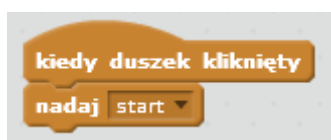


## B. Skrypt bohatera gry.

1. Pierwszym krokiem będzie ustawienie strony startowej gry. Będzie się ona pojawiać po włączeniu skryptu, czyli kliknięciu na ZIELONĄ FLAGĘ. (blok ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ).
2. Następnie ustawimy naszego Duszka w konkretnym miejscu okna PODLGĄDU, a do tego celu użyjemy komendy bloku RUCH → IDŹ DO x: .. y: ..
3. Teraz należy wybrać właściwy kostium bohatera gry, czyli stronę startową przy pomocy komendy bloku WYGLĄD → ZMIENŃ KOSTIUM NA START.
4. Nasz Duszek próbując pokonać przeszkody zawsze będzie obrócony w prawo, dlatego przy każdym uruchomieniu gry musimy nadać mu odpowiedni kierunek. Wykorzystamy do tego komendę bloku RUCH → USTAW KIERUNEK NA 90.



5. Kiedy użytkownik jest już gotów klika na stronę startową (czyli Duszka) i rozpoczyna grę. Dlatego wybieramy ZDARZENIA → KIEDY DUSZEK KLIKNIĘTY.
6. Następnie stworzymy wiadomość, która będzie informacją dla wszystkich Duszków, że należy uruchomić skrypty działające w trakcie gry. Wyciągamy ZDARZENIA → NADAJ ... . Rozwijamy listę w okienku komendy i wybieramy NOWY KOMUNIKAT. Następnie pojawia się okienko, do którego wpisujemy nazwę wiadomości, akceptujemy ją klikając ok i podpinamy pod ZDARZENIE KIEDY DUSZEK KLIKNIĘTY.



7. Gra polega na pokonywaniu przeszkód, dlatego wyposażymy naszego Duszka w umiejętność skakania. Dzieci samodzielnie zastanawiają się przy użyciu jakich komend można to zrealizować. Podpowiedź: Duszek przemieszcza się z punktu A do B i z powrotem. Dzieci sprawdzają komendy bloku RUCH.
8. Duszek będzie skakał przy pomocy klawisza spacji, dlatego wybieramy ZDARZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY.
9. Następnie wybieramy dwie te same komendy bloku RUCH → LEĆ PRZEZ 0.5s DO x: -150 y: 120. Komendy różnią się wartościami dla współrzędnej y, które to pozwalają na przesunięcie Duszka w górę, a następnie w dół. Odpalamy skrypt i sprawdzamy czy działa.



10. Duszek skacze, ale chcielibyśmy, żeby dodatkowo biegał. Wykonamy to przy pomocy prostej animacji (pytamy dzieci czy wiedzą w jaki sposób?), czyli zmieniając kostiumy naszego Dinozaura.
11. Wybieramy odpowiednie ZDARZENIE → KIEDY OTRZYMAM START.
12. Następnie podpinamy dwa razy komendę z bloku WYGLĄD → ZMIEŃ KOSTIUM NA ... . Kostiumy różnią się uniesioną raz prawą, raz lewą nogą Dinozaura.
13. Kolejnym krokiem jest wybranie komendy z bloku KONTROLA → CZEKAJ 0.2 s, która to pozwoli na to, aby kostiumy nie zmieniały się zbyt szybko.
14. Całość umieszczamy w PĘTLI NIESKOŃCZONEJ tak, aby animacja biegu Duszka wykonywała się przez cały czas trwania gry.



Uruchamiamy skrypt i sprawdzamy czy działa.

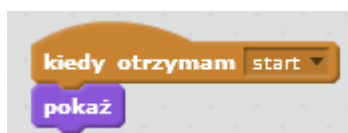
### C. Linia

Narysowaną przez nas wcześniej linię gruntu będziemy ukrywać podczas uruchamiania skryptu tak, aby mogła pojawić się strona startowa. Dlatego wybieramy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy komendę bloku WYGLĄD → UKRYJ.

Kolejnym krokiem jest ustawienie naszego Duszka na odpowiedniej wysokości PODGLĄDU DUSZKÓW. Wykonujemy to przy pomocy komendy bloku RUCH → IDŹ DO x: ... y: ... .

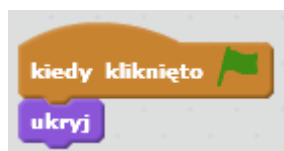


Ukryliśmy naszą linię, ale w trakcie gry ma ona być widoczna tak, aby Dino miał po czym dreptać. Gra rozpoczyna się od komunikatu START, dlatego wybieramy odpowiednie ZDARZENIE → KIEDY OTRZYMAM START. Następnie podpinamy z bloku WYGLĄD → POKAŻ.



### D. Duszki-przeszkody, czyli kaktusy

Tak jak poprzedni Duszek, Kaktusy również będą niewidoczne w momencie rozpoczęcia skryptu, pojawią się natomiast w trakcie gry. Dlatego wybieramy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy komendę bloku WYGLĄD → UKRYJ.

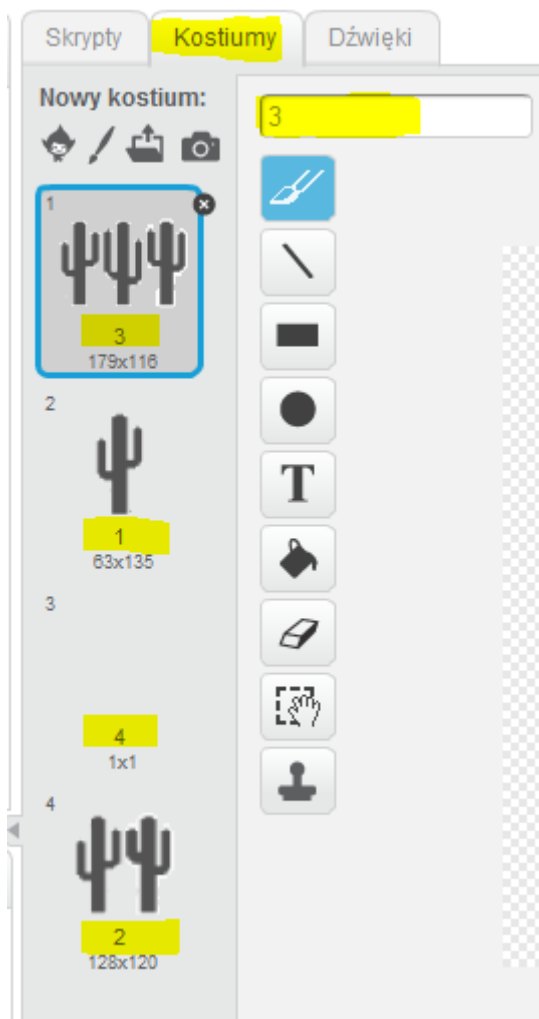


Nasz Dino drepta, ale nie porusza się. To kaktusy płynące od prawej do lewej pozwolą nam na uzyskanie efektu biegu. Ale po kolei. Zakryte wcześniej Kaktusy, odkryjemy teraz. Wybieramy blok WYGLĄD → POKAŻ, które to, podpinamy oczywiście pod odpowiednie ZDARZENIE START rozpoczynające naszą grę, ZDARZENIA → KIEDY OTRZYMAM START.



Nadajemy Kaktusowi jego miejsce/położenie w oknie PODLGĄDU DUSZKÓW. Skoro Duszek ma poruszać się z prawej do lewej (pytamy dzieci gdzie umiejscowimy Duszka i przy pomocy jakiej komendy?) to ustawimy go na prawej krawędzi PODLGĄDU. Użyjemy do tego celu komendy bloku RUCH → IDŹ DO x: ... y: ... . Wartość współrzędnej x będzie wynosiła 240, ponieważ taka jest maksymalna dodatnia wartość osi x, taka jest szerokość PODLGĄDU DUSZKÓW. Kaktusy będą znajdować się na wysokości Duszka-Linii, czyli wartości współrzędnej y, muszą być takie jak naszej linii gruntu (trochę niższe).

Teraz przechodzimy do zakładki KOSTIUMY i tam zmieniamy kostiumom naszego Duszka nazwy na cyfry. W zależności ile tych kostiumów, tyle cyfr nadajemy.



Nazwy kostiumów w postaci cyfr są nam niezbędne, abyśmy byli w stanie te kostiumy losować odpowiednią funkcją, a następnie wynik losowania przypisać odpowiedniemu kostiumowi. Dzięki temu nasze przeszkody nie będą nudne. Zrealizujemy to przy pomocy komendy bloku WYGLĄD → ZMIENŃ KOSTIUM NA ...

W oknie powyższej komendy umieścimy sobie funkcję losującą, która pozwoli nam, na wybieranie niezależnie od nas cyfr od 1 do 4 (w przykładzie). Aby to zrobić wyciągamy z bloku WYRAŻENIA → LOSUJ OD ... DO ...

Kolejnym krokiem jest poruszanie się naszego Kaktusa (pytamy dzieci jaki mają pomysł na to?), który będzie przesuwiał się od prawej do lewej. Ustawiliśmy go na prawej krawędzi, teraz tylko należy wprawić go w ruch. Wybieramy do tego celu komendę bloku RUCH → LEĆ PRZEZ 1,8s DO x: -240 y: -75. Czas mknięcia przez PODGLĄD będzie wynosił 1,8s, który wydaje się być optymalny, ale w zależności od upodobań Twórców gry należy go zmodyfikować.

Celem naszego Duszka będzie wartość osi x, która wynosi -240, ponieważ jest minimalna



wartość osi x dla PODGLĄDU. Uruchamiamy i sprawdzamy czy działa. Całość zamykamy oczywiście w PĘTLI NIESKOŃCZONEJ, tak, aby blok skryptu wykonywał się przez cały czas trwania gry.



## E. Skrypt bohatera gry

Dino już biega, ale nie widać, żeby Kaktusy mu przeszkadzały. (Pytamy dzieci co możemy zrobić, aby zakończyć grę w momencie, w którym Dino wpadnie na Kaktusy?) Wybieramy ZDARZENIA → KIEDY OTRZYMAM START. Następnie podpinamy pod powyższą komendę INSTRUKCJĘ WARUNKOWĄ z bloku KONTROLA → JEŻLI ... TO. Zabieg ten pozwoli nam na sprawdzenie czy Dino dotyka Duszków-Przeszkód czy też nie. W oknie INSTRUKCJI umieszczamy CZUJNIK (pytamy dzieci, która to komenda będzie?) z bloku CZUJNIKI → DOTYKA KAKTUS? Jeżeli Dino dotyka Przeszkodę to jego kostium zmienia się na „dinozaurZaskoczony”. Do tego celu użyjemy komendy bloku WYGLĄD → ZMIEŃ KOSTIUM NA DINOZAUZRASKOCZONY.

Nadamy również od razu KOMUNIKAT o przegranej czyli skończonej grze. Zrobimy to przy pomocy komendy bloku ZDARZENIA → NADAJ GAMEOVER (komunikat tworzymy tak jak poprzednio, wybierając z listy NOWY KOMUNIKAT).

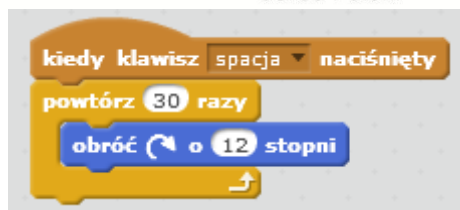
UWAGA: W tym wypadku bardzo ważna jest kolejność trzech ostatnich komend, ponieważ przed zakończeniem działania wszystkich skryptów chcemy, aby Duszek zdążył jeszcze zmienić kostium oraz pojawił napis o zakończonej grze.

Następnie kończymy działanie wszystkich skryptów. Wykorzystamy w tym celu komendę bloku KONTROLA → ZATRZYMAJ WSZYSTKO. Sprawdzamy czy działa.



Aby urozmaicić skoki naszego Bohatera, zrobimy z niego akrobatę. Będziemy obracać go o 12 stopni wokół własnej osi. (Pytamy dzieci: ile razy musimy wykonać/powtórzyć obrót o 12 stopni, tak aby Duszek obrócił się o pełne koło?) Wybieramy z bloku ZDARZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY? Poniżej podpinamy z bloku RUCH → OBRÓC (W PRAWO) O 12 STOPNI. Powyższą komendę zawieramy z PĘTLI ITERACYJNEJ z bloku KONTROLA → POWTÓRZ 30 RAZY. Sprawdzamy czy działa.



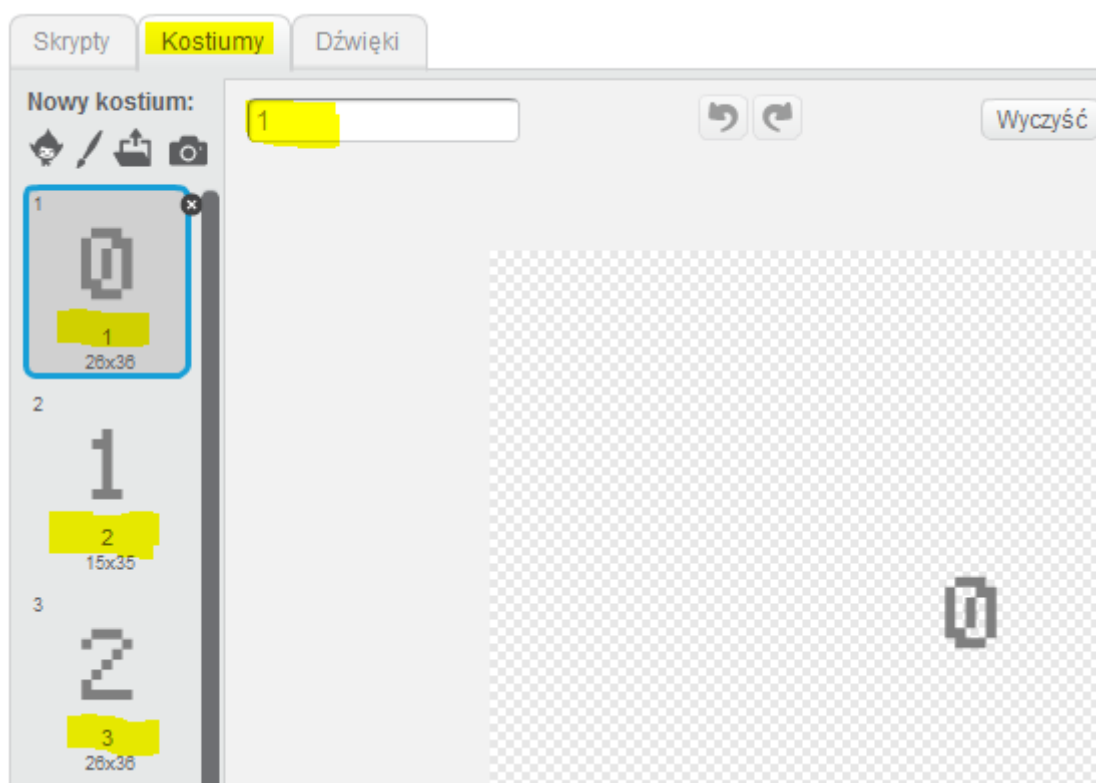


## F. Czas

Mamy bohatera pokonującego przeszkodę. Czego nam brakuje, aby gra była kompletna? W grach najważniejsza jest rzecz jasna dobra zabawa (wersja dla dzieci), ale jeżeli chcemy się zmierzyć z koleżankami/kolegami (czytaj: rywalami) to nie ma reguł, wszystkie chwytaki dozwolone. Teraz na serio: jeżeli chcemy się zmierzyć z innymi graczami, potrzebujemy miarodajnego odniesienia naszych umiejętności (np. refleksu) w tym wypadku będzie to czas, w którym udało się pozostać nam przy życiu. W tym celu na samym początku stworzyliśmy trzy Duszki cyfry od 6 do 10 kostiumów każdy.

Mamy Duszki: jedności sekundy (nazwa Duszka: sekundy), dziesiątne sekundy (sekundy2) oraz minuty (minuty).

W pierwszej kolejności, naszym zadaniem będzie nazwanie odpowiednio kostiumów Duszków. Jest to potrzebne, ponieważ w trakcie gry, będziemy odwoływać się do kostiumów, poprzez zmienne całkowitoliczbowe. Wybieramy zakładkę KOSTIUMY i tam, w lewym górnym rogu Edytora Obrazu, nadajemy kostiumom nazwy.



**UWAGA:** Kostiumy nie są numerowane od 0, tak jak wskazywałoby na to logika, ponieważ w takim przypadku Scratch wariuje. Nazwy nadajemy od cyfry 1 wzwyż.

Będziemy teraz ustawiać Duszki w odpowiednim miejscu Podglądu oraz „nastawiać” zegar. Wybieramy z bloku ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy z bloku WYGLĄD → UKRYJ oraz ZMIENŃ KOSTIUM NA 1.

Kolejnym krokiem jest ustawienie Duszków w jednej linii (wspólna wartość współrzędnej x).



Wybieramy z bloku RUCH → IDŹ DO x: 180 y: 149 (prawy górny róg Podglądu). Następnie resetujemy czas przy pomocy komendy z bloku DANE → USTAW SEKUNDY NA 1.



W momencie rozpoczęcia gry Duszki będą zmieniać kostiumy na odwołującą się do nich zmienną całkowitoliczbową. Wybieramy blok ZDARZENIA → KIEDY OTRZYMAM START. Następnie podpinamy z bloku WYGLĄD → POKAŻ.

Kolejnym krokiem jest zmiana kostiumu, która odbywać się będzie w trakcie całej gry, dlatego wybieramy PĘTLĘ NIESKOŃCZONĄ z bloku KONTROLA → ZAWSZE. W jej wnętrzu umieszczamy komendę z bloku WYGLĄD → ZMIENŃ KOSTIUM NA ... . W okno powyższej komendy wkładamy ZMIENNĄ z bloku DANE → SEKUNDY. Teraz w zależności od wartości ZMIENNEJ SEKUNDY taki kostium ubierze nasz Duszek.



Wykonujemy punkty od 1-10 dla dwóch pozostałych Duszków-Czasu, zmieniając wartości współrzędnej x (współrzędna y pozostaje bez zmian) oraz w zależności od Duszka zmieniając nazwy ZMIENNYCH na SEKUNDY2 oraz MINUTY.

UWAGA: Dla kolejnych Duszków dzieci tworzą bloki komend z pamięci, wzorując się na już stworzonych blokach, bądź duplikują bloki i przenoszą je.

## F1. Sekundy

W tym miejscu zajmiemy się uruchomieniem naszego zegara czyli odliczaniem czasu, co do sekundy. Wybieramy z bloku ZDARZENIA → KIEDY ORZYMAM START. Następnie podpinamy z bloku KONTROLA → ZAWSZE. W tej PĘTLI, odliczany będzie przez cały przebieg gry – czas. Tutaj będziemy inkrementować ZMIENNĄ SEKUNDY. We wnętrzu PĘTLI umieszczamy z bloku DANE → ZMIENŃ SEKUNDY O 1.

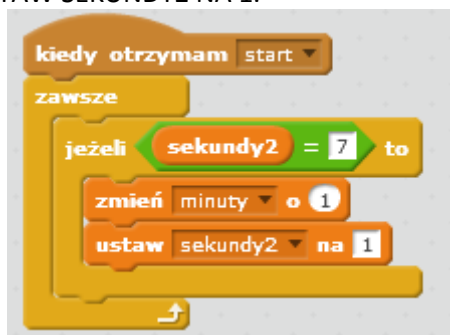
Następnie podpinamy z bloku KONTROLA → CZEKAJ 0,1s. Komenda ta, pozwoli nam na odliczanie wirtualnego czasu w świecie gry. Kolejnym krokiem jest umieszczenie INSTRUKCJI WARUNKOWEJ, która pozwoli nam po osiągnięciu 9 sekundy, na powrót do 0. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewym oknie ZNAKU ... = ... umieszczamy ZMIENNĄ SEKUNDY (z bloku DANE), w prawym oknie natomiast wpisujemy wartość 11. Do wnętrza INSTRUKCJI wkładamy z bloku DANE → ZMIENŃ SEKUNDY2 O 1. Poniżej podpinamy z bloku DANE → USTAW SEKUNDY NA 1.



## F2. Sekundy 2

Wybieramy z bloku ZDARZENIA → KIEDY ORZYMAM START. Następnie podpinamy z bloku KONTROLA → ZAWSZE.

Kolejnym krokiem jest umieszczenie INSTRUKCJI WARUNKOWEJ, która pozwoli nam po osiągnięciu 5 sekundy, na powrót do 0, tak aby ZMIENNA MINUTY została naliczona. Wybieramy z bloku KONTROLA → JEŻELI .. TO. W oknie komendy umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewym oknie ... = ... umieszczamy z bloku DANE → ZMIENNĄ SEKUNDY2, a w prawym wartość 7. Do wnętrza INSTRUKCJI wkładamy z bloku DANE → ZMIENI MINUTY O 1. Poniżej podpinamy z bloku DANE → USTAW SEKUNDY2 NA 1.



Uruchamiamy grę i sprawdzamy czy CZAS jest naliczany.

### Rozszerzenia:

Wprowadzamy kolejną przeszkodę, jaką jest Duszek-Ptak. Może również występować jako urozmaicenie krajobrazu.

Ubogacamy grę o melodie napisane samodzielnie (blok DŹWIĘKI → INSTRUMENTY), bądź importujemy gotowce.



## Zajęcia 8: Scratch FIFA cz.1

### Zadanie:

Wykonanie gry „Scratch FIFA 15 cz. 1”, która składa się ze sterowanego przez użytkownika Piłkarza oraz odbijanej przez niego Piłki.

### Cel:

Przypomnienie jak stworzyć sterowanego duszka oraz nauka, jak poruszać duszkiem, który został popchnięty.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy. Projektujemy architekturę gry czyli wspólnie z dziećmi wypisujemy na tablicy jej etapy. Ten projekt ma na celu pozwolić dzieciom na planowanie swojej pracy oraz szukaniu własnych rozwiązań.

### Wykonanie:

#### A. Piłkarz

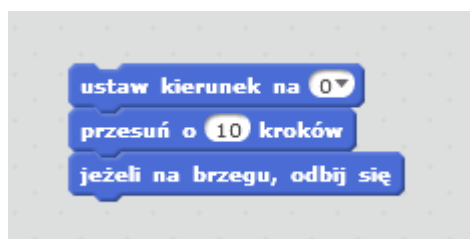
Klikamy na ikonę Piłkarza i przechodzimy do zakładki Skrypt. Naszym pierwszym zadaniem jest „zaprogramowanie” Piłkarza w taki sposób, żebyśmy mogli nim sterować przy pomocy klawiatury. W tym celu wykorzystamy komendy bloków: Ruchu i Zdarzenia. Dzieci próbują przypomnieć sobie i samodzielnie wybrać bloki komend odpowiedzialne za sterowanie duszkiem. Będziemy poruszać naszym Piłkarzem przy pomocy klawiatury. Użyjemy do tego celu następujących komend:

bloku RUCHU → USTAW KIEDUNEK NA... tak, aby nasz Duszek ustawił się w odpowiednią stronę,

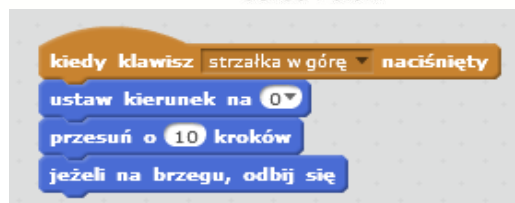
bloku RUCHU → PRZESUŃ O 10 KROKÓW - duszek będzie się poruszał,

bloku RUCHU → JEŻELI NA BRZEGU, ODBIJ SIĘ – duszek nie zniknie nam z ekranu, ponieważ odbije się od jego krawędzi.

Komendy łąpiemy klikając na nie i przenosimy na obszar Edytora Duszka.



Nasz duszek jest już wyposażony w komendy ruchu. Potrzebne są nam jeszcze odpowiednie ZDARZENIE pozwalające sterować nim przy pomocą klawiatury. W tym celu klikamy na blok ZDARZENIA wybieramy komendę KIEDY KLAWISZ ... NACIŚNIĘTY i przeciągamy ją na pole edytora.



Klikając lewym przyciskiem myszy na komendę KIEDY KLAWISZ... wybieramy z listy DUPLIKUJ i wykonujemy tę czynność 3x. Mamy teraz 4 klony blozków komend. Następnie klikamy i rozwijamy listę komendy KIEDY KLAWISZ... i wybieramy tam poszczególne klawisze dla pozostałych klonów. Następnie z odpowiedniej listy dla komendy USTAW KIERUNEK, wybieramy poszczególne kierunki. Spróbujmy teraz poruszyć naszym Piłkarzem!

Aby zawsze na początku rozgrywki nasz Piłkarz zaczynał grę od środka pola ustalamy jego położenie. Wybieramy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Pod tę komendę podpinamy RUCH → IDŹ DO x:... y:... i wybieramy współrzędne środka boiska.

## B. Piłka

Czas umożliwić zawodnikowi odbijanie piłki. Duszek przy każdym rozpoczęciu rozgrywki oraz po голу będzie lądować na samym środku boiska.

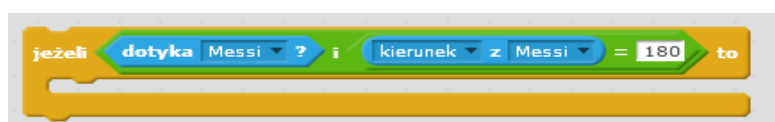
Wybieramy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Pod powyższą komendę podpinamy RUCH → IDŹ DO x: ... y: ... i tym samym ustawiamy Piłkę we właściwym miejscu.

Piłka nie będzie się poruszać niedotknięta przez Piłkarza, dlatego do stworzenia imitacji ruchu kopania piłki niezbędne będą INSTRUKCJE WARUNKOWE. Wybieramy KONTROLA → JEŻELI .. TO. Instrukcja będzie się wykonywać tylko wtedy, gdy zostaną spełnione dwa warunki. Aby oba warunki były sprawdzone w jednym czasie niezbędne będzie użycie OPERATORA LOGICZNEGO AND (i) czyli wykorzystanie KONIUNKCJI. Realizujemy to umieszczając w oknie INSTRUKCJI WARUNKOWEJ z bloku WYRAŻENIA → ... i ... .

Piłka będzie poruszać się tylko w momencie popchnięcia jej przez Piłkarza, czyli dotknięcia Duszka. Wykorzystamy do tego CZUJNIKI → DOTYKA MESSI?, którą to komendę umieszczamy po lewej stronie OPERATORA LOGICZNEGO AND.

Piłkarzem sterujemy, nadając mu odpowiedni kierunek (zwrot) przy pomocy klawiatury. Wyróżniamy 4 wartości : prawo (90), lewo (-90), góra (0) oraz dół (180). Kierunki reprezentowane są przez wartości liczbowe podane w nawiasach. Naszym zadaniem jest obrócenie Piłki w tę stronę, w którą aktualnie jest zwrócony Piłkarz tak, aby uzyskać efekt „kopania” Piłki.

Po prawej stronie OPERATORA LOGICZNEGO AND, umieszczamy ZNAK RÓWNOŚCI, który wybieramy z bloku WYRAŻENIA → ... = ... . Przy pomocy czujnika odczytujemy wartość kierunku, w którym aktualnie zwrócony jest Piłkarz. Dzieci przeglądają listę komend bloku CZUJNIKI i samodzielnie próbują odnaleźć odpowiednią komendę. Realizujemy to przy pomocy bloku CZUJNIKI → POZYCJA X Z MESSI i z obu list wybieramy odpowiednio KIERUNEK Z MESSI a następnie umieszczamy po lewej stronie ZNAKU RÓWNOŚCI. Po prawej stronie wpisujemy wartość pierwszego kierunku np. 180.



Wiemy już kiedy nasza Piłka ma być „kopnięta”. Czego nam jeszcze brakuje, żeby ją wprawić w ruch? Niezbędne są bloki ruchu, które umożliwią jej poruszanie się. Jest to ten sam zestaw, co w przypadku Piłkarza:

- a) bloku RUCHU → USTAW KIERUNEK NA..., tak aby nasz Duszek ustawił się w odpowiedniej stronie;



- b) bloku RUCHU → PRZESUŃ O 10 KROKÓW - duszek będzie się poruszał;
- c) bloku RUCHU → JEŻELI NA BRZEGU, ODBIJ SIĘ – duszek nie zniknie nam z ekranu, ponieważ odbije się od jego krawędzi.

Powyższe bloki podpinamy pod siebie w wymienionej kolejności i umieszczamy we wnętrzu INSTRUKCJI WARUNKOWEJ.



Klikając lewym przyciskiem myszy na komendę INSTRUKCJI WARUNKOWEJ wybieramy z listy DUPLIKUJ i wykonujemy tę czynność 3x. Mamy teraz 4 klony bloczków komend. W poszczególnych WARUNKACH kolejnych INSTRUKCJI, zmieniamy wartości kierunków na: 90, 0 i -90. Następnie klikamy i rozwijamy listy komend USTAW KIERUNEK NA i wybieramy tam odpowiadające powyższym wartości. Wszystkie INSTRUKCJE WARUNKOWE umieszczamy w PĘTLI ZAWSZE tak, aby WARUNKI były sprawdzane przez cały czas trwania skryptu.





## C. Scena

Naszą SCENĄ jest samodzielnie narysowane boisko do gry w piłkę nożną, które wykonujemy w EDYTORZE OBRAZU.

Aby wyniki były miarodajne i można było je porównywać między graczami, niezbędne są punkty. Tworzymy zmienne DANE → TEAM1 i TEAM2, które będą zbierać i zapamiętywać nasze punkty.

Aby okna punktów wyświetlane na Poglądzie Duszaków nie zajmowały nam zbyt dużej powierzchni boiska będziemy je wyświetlać i ukrywać przy pomocy klawiszy. Wybieramy ZDARZENIA → KIEDY KŁAWISZ SPACJA NACIŚNIĘTY oraz KIEDY KŁAWISZ B NACIŚNIĘTY.

Poniżej podpinamy kolejno: DANE → POKAŻ ZMIENNĄ TEAM1 oraz POKAŻ ZMIENNĄ TEAM2, a pod kolejne zdarzenie - DANE → UKRYJ ZMIENNĄ TEAM1 oraz UKRYJ ZMIENNĄ TEAM2.



## D. Piłka

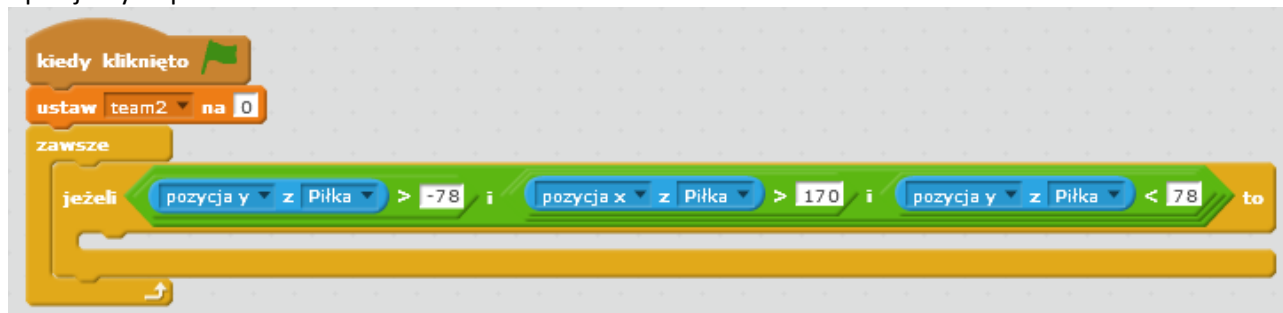
Gole będą liczone przez cały czas trwania skryptu, dlatego wybieramy ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Zaraz po rozpoczęciu gry, wyniki powinny być zerowane. Pod powyższe zdarzenie podpinamy DANE → USTAW TEAM1 NA 0.

Golem nazywamy sytuację, w której piłka wpada do bramki. W naszym przypadku, bramka jest to określona powierzchnia, którą wyznaczymy przy pomocy WSPÓŁRZĘDNYCH. W tym celu wykorzystamy INSTRUKCJĘ WARUNKOWĄ. Wybieramy KONTROLA → JEŻELI ... TO.

Teraz należy stworzyć warunek, który będzie wyznaczał nam zbiór współrzędnych tworzących naszą bramkę. Zrealizujemy to dzięki poznanemu dziś OPERATOROWI LOGICZNEMU AND, blok WYRAŻENIA → ... i ... . W trzech oknach powyższego wyrażenia umieścimy znaki MNIEJSZOŚCI oraz WIĘKSZOŚCI, blok WYRAŻENIA → ... > ... i ... < ... .

Kolejnym krokiem jest sprawdzenie czy nasza piłka znalazła się w bramce, czyli czy wartość pozycji x Piłki jest większa niż współrzędna początku bramki (perspektywa na bramkę ze środka pola). Ustalamy w ten sposób długość bramki. Wybieramy CZUJNIKI → POZYCJA X Z PIŁKA, którą to komendę umieszczamy po lewej stronie ZNAKU WIĘKSZOŚCI. Po prawej stronie znaku wpisujemy odpowiednią wartość (w przykładzie jest to pozycja  $x > 170$ ).

Teraz ustalamy szerokość bramki. Musimy wyznaczyć odpowiedni przedział wartości osi Y, w którym musi znaleźć się Piłka, aby gol był zaliczony. Tak jak wcześniej po lewej stronie znaków MNIEJSZOŚCI i WIĘKSZOŚCI umieszczamy komendę POZYCJA Y Z PIŁKA, a po prawej stronie wpisujemy odpowiednie wartości.



Jeżeli warunek jest spełniony (prawdziwy), należy zaliczyć punkt, czyli podnieść wartość



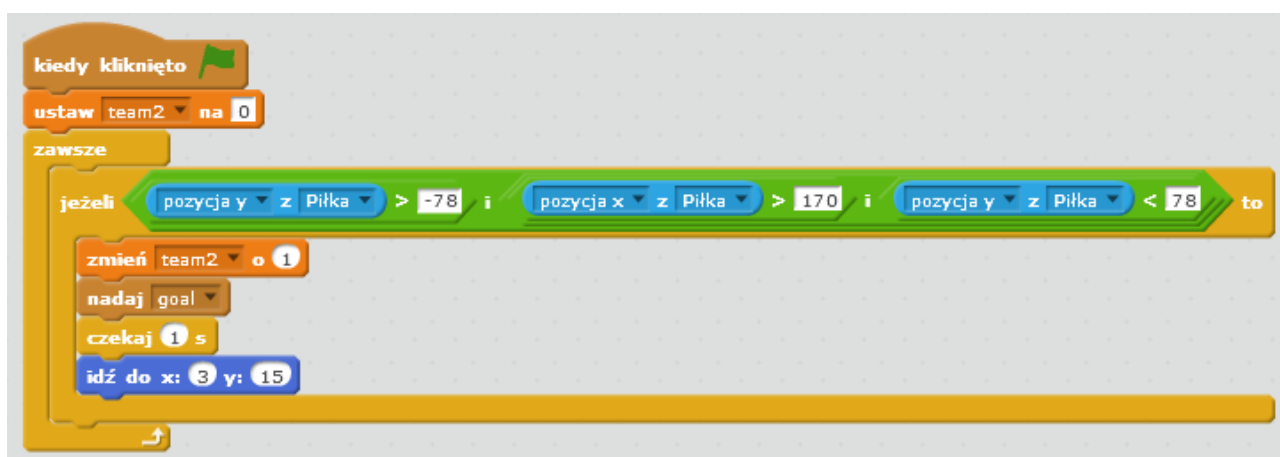
zmiennej TEAM2 o 1. Wybieramy DANE → ZMIEŃ TEAM1 O 1, którą to komendę umieszczamy wewnątrz INSTRUKCJI WARUNKOWEJ.

Teraz nadajemy komunikat dla Piłkarza, który pozwoli na to, aby Duszek cofnął się z pola karnego z powrotem na środek pola. W tym celu wykorzystujemy ZDARZENIA → NADAJ GOAL.

Musimy spauzować na chwilę grę tak, aby Gracz zaobserwował, że gol faktycznie miał miejsce, czyli przetrzymujemy przez sekundę Piłkę w bramce. Przeczekanie umożliwi nam również, precyzyjne naliczanie punktów, ponieważ bez tej czynności punkty naliczane byłyby w błyskawicznym tempie i w ciągu sekundy zdobylibyśmy tysiące punktów. Wybieramy KONTROLA → CZEKAJ 1 s.

Ostatnim elementem gry jest cofnięcie Piłki z powrotem na środek boiska. Wykorzystujemy RUCH → IDŹ DO X: ... Y: ... .

Analogicznie przeprowadzamy zliczanie punktów dla drużyny przeciwnej.







## Zajęcia 9: Scratch FIFA cz.2

### Zadanie:

Wykonanie gry „Scratch FIFA 15 cz. 2”, która zawiera sterowalnego przez użytkownika Piłkarza nr 2, który składa się na multiplayera czyli grę wieloosobową oraz bramkarza.

### Cel:

Tworzenie sterowalnego duszka w nowy sposób oraz powtórka dotycząca animacji.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy. Projektujemy architekturę gry wspólnie z dziećmi. Wypisujemy na tablicy jej etapy. Ten projekt ma na celu pozwolić dzieciom na planowanie swojej pracy oraz szukaniu własnych rozwiązań.

### Wykonanie:

#### A. Piłkarz 2

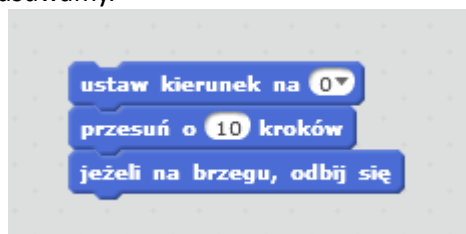
Duplikujemy duszka Piłkarza i przechodzimy do zakładki Skrypt. Naszym pierwszym zadaniem jest „zaprogramowanie” Piłkarza 2 w taki sposób, abyśmy mogli nim tak jak pierwszą postacią sterować przy pomocy klawiatury. W tym celu wykorzystamy komendy bloków: ruchu, zdarzenia, kontroli oraz czujników.

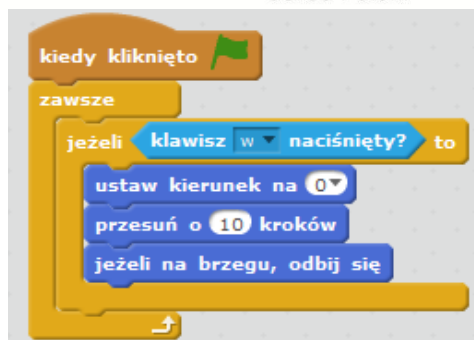
Dzieci próbują same zaproponować i samodzielnie wybrać bloki komend odpowiedzialne za sterowanie duszkiem.

Będziemy poruszać naszym Piłkarzem przy pomocy klawiatury. Przeciągamy blok ZADRZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Pod zdarzeniem podpinamy INSTRUKCJĘ WARUNKOWĄ, która pozwoli nam poruszać Duszkem za pomocą kolejnych klawiszy. Blok KONTROLA → JEŻELI TO. W oknie komendy umieszczamy CZUJNIKI → KLAWISZ SPACJA NACIŚNIĘTY? Wybieramy z listy czujnika, odpowiedni klawisz (używamy klawiszy WSAD). Wewnątrz warunku umieścimy komendy RUCHU, które pozwolą na poruszanie się Duszka po naciśnięciu konkretnego klawisza:

- bloku RUCHU → USTAW KIEDUNEK NA.., tak aby nasz Duszek ustawił się w odpowiedni stronę,
- bloku RUCHU → PRZESUŃ O 10 KROKÓW - duszek będzie się poruszał,
- bloku RUCHU → JEŻELI NA BRZEGU, ODBIJ SIĘ – duszek nie zniknie nam z ekranu, ponieważ odbije się od jego krawędzi.

Powyższe komendy znajdują się w skrypcie skopiowanego Duszka. Niepotrzebne zdarzenia ze skryptu skopiowanego Duszka usuwamy.





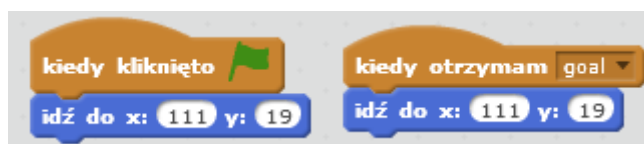
Duplikujemy powyższy blok 3 razy tak, aby w zależności do naciśniętych czterech klawiszy Duszek kierował się w 4 strony świata (bądź Dzieci tworzą bloczki de novo).



Ustawiamy Gracza zawsze w tym samym położeniu kiedy rozpoczyna grę oraz po wbiciu/otrzymaniu gola, czyli ustawiamy go w połowie swojego boiska. Blok RUCH → IDŹ DO X ... Y ... - wybieramy odpowiednie współrzędne.

Komendy bloku RUCHU podpinamy pod odpowiednie ZDARZENIA:

- rozpoczynające grę – KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ;
- KOMUNIKAT – KIEDY OTRZYMAM GOAL.



## B. Bramkarz

Importujemy/rysujemy Duszka, który będzie naszym Bramkarzem (musi posiadać dwa kostiumy, aby stworzyć animację ruchu zawodnika). Dzieci przypominają jak tworzymy animację.

Duszek będzie „poruszał się” przez cały czas trwania skryptu czyli od ZDARZENIA → KIEDY



KLIKNIĘTO ZIELONĄ FLAGĘ. Aby wykonywało się to stale, wykorzystujemy PĘTLE NIESKOŃCZONĄ, blok KONTROLA → ZAWSZE. W pętli umieszczamy WYGLĄD → NASTĘPNY KOSTIUM.

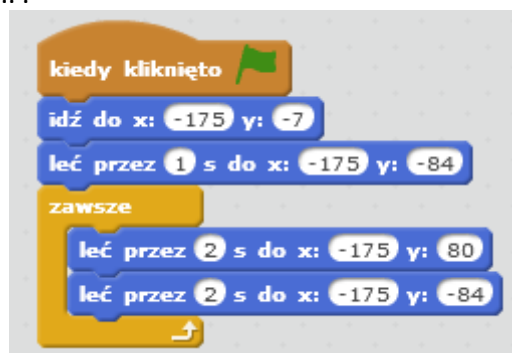
Teraz, musimy uwidocznić zmianę kostiumu, aby nie następowała zbyt szybko. Wybieramy KONTROLA → CZEKAJ 0,15 s, które podpinamy pod komendę zmiany kostiumu (mimo tego, Duszek nie porusza się, co dalej?).



Nasz Duszek będzie poruszał się wyłącznie po zewnętrznej linii boiska. Dlatego należy wyznaczyć mu dwa punkty na osi współrzędnych, które będą wyznaczać szerokość bramki. Użyjemy do tego celu dwóch komend bloku RUCHU → LEĆ PRZEZ 2s DO X ... Y ... .

Umieszczamy powyższe komendy w PĘTLI NIESKOŃCZONEJ, aby Duszek poruszał się przez cały czas trwania gry. Wybieramy blok KONTROLA → ZAWSZE.

Jeżeli Bramkarz ma rozpoczynać mecz na środku bramki musimy nad PĘTLĄ, umieścić jeszcze dwie komendy RUCHU, które pozwolą na umieszczenie Duszka w tym właśnie punkcie. Z kolei następna komenda umożliwi pokonanie połowy długości linii bramki. Blok RUCHU → IDŹ DO X ... Y ... oraz LEĆ PRZEZ 2s DO X ... Y ... .



## C. Bramkarz 2

Bramkarza 2 dzieci wykonują samodzielnie, analogicznie do skryptu Bramkarza.

## D. Piłka

Jesteśmy już w stanie poruszać Graczem 2, ale niestety nie jest on w stanie jeszcze poruszyć Piłką. Jak myślicie co należy wykonać, aby umożliwić rywalizację Piłkarzom na boisku?

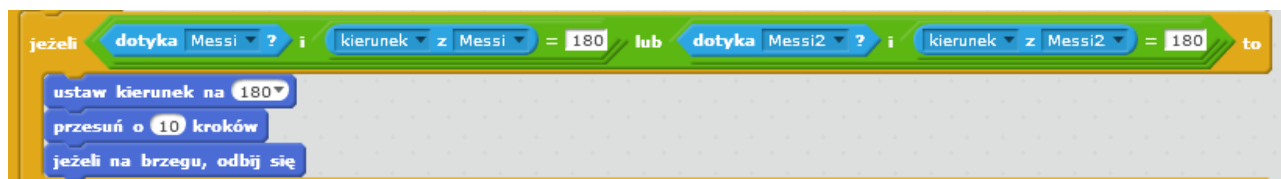
Należy rozszerzyć WARUNKI INSTRUKCJI WARUNKOWYCH, które zostały użyte w części pierwszej gry. Jak to zrobić? Aktualnie zapisany mamy WARUNEK, który sprawdza czy Piłka dotyka Gracza I jaki jest kierunek zwrotu Piłkarza (i - oznacza, że warunki są tak samo ważne i oba muszą być spełnione, aby wykonały się komendy we wnętrzu INSTRUKCJI).

Takie same dwa warunki musimy teraz napisać dla Gracza 2. Wybieramy blok WYRAŻENIA → ... I ... .

W lewe okno powyższej komendy umieszczamy CZUJNIKI → DOTYKA MESSI2. W prawe okno komendy umieszczamy CZUJNIKI → KIERUNEK Z MESSI2.



Stworzyliśmy po dwa warunki dla każdego Gracza. Aby były one wybierane opcjonalnie (w zależności, który Duszek dotyka Piłki), wybieramy WYRAŻENIA → LUB. W oknach po obu stronach umieszczamy komendy dotyczące Duszków.



Analogicznie poszerzamy WARUNKI dotyczące pozostałych zwrotów (kierunków obrotu) Piłki.

Kolejnym krokiem jest cofnięcie piłki na środek boiska, po obronieniu bramki. Do tego celu wykorzystamy INSTRUKCJĘ WARUNKOWĄ, blok KONTROLA → JEŻELI ... TO . W oknie komendy umieszczamy WARUNEK, który umożliwi Piłce na rozpoznanie kontaktu z Bramkarzem. Blok CZUJNIKI → DOTYKA ... ? We wnętrzu INSTRUKCJI umieszczamy komendę odpowiedzialną za wysłanie Piłki na środek boiska. Blok RUCH → IDŹ DO X ... Y ... .



### Rozszerzenie:

- Dzieci dodają muzykę, która towarzyszy wbiciu gola.
- Dodajemy Duszki-przeszkody (poruszające się z punktu A do B), które tak jak Bramkarz odsyłają Piłkę w odległe miejsce boiska.



## Zajęcia 10: Królestwo

### Zadanie:

Wykonanie gry „Królestwo”, która zawiera „sterowalnego” przez użytkownika Duszka. Wyjątkowość skryptu polega na tym, iż bohater gry pozostaje w bezruchu, natomiast poruszają się duszki tworzące tło. W wyniku tego zwiększa się obszar, po którym Duszek może wędrować.

### Cel:

Stworzenie odpowiednich grafik do naszej gry oraz tworzenie sterowalnego tła (a nie Duszka). Tworzenie wrażenia ruchu bohatera, bez jego poruszania.

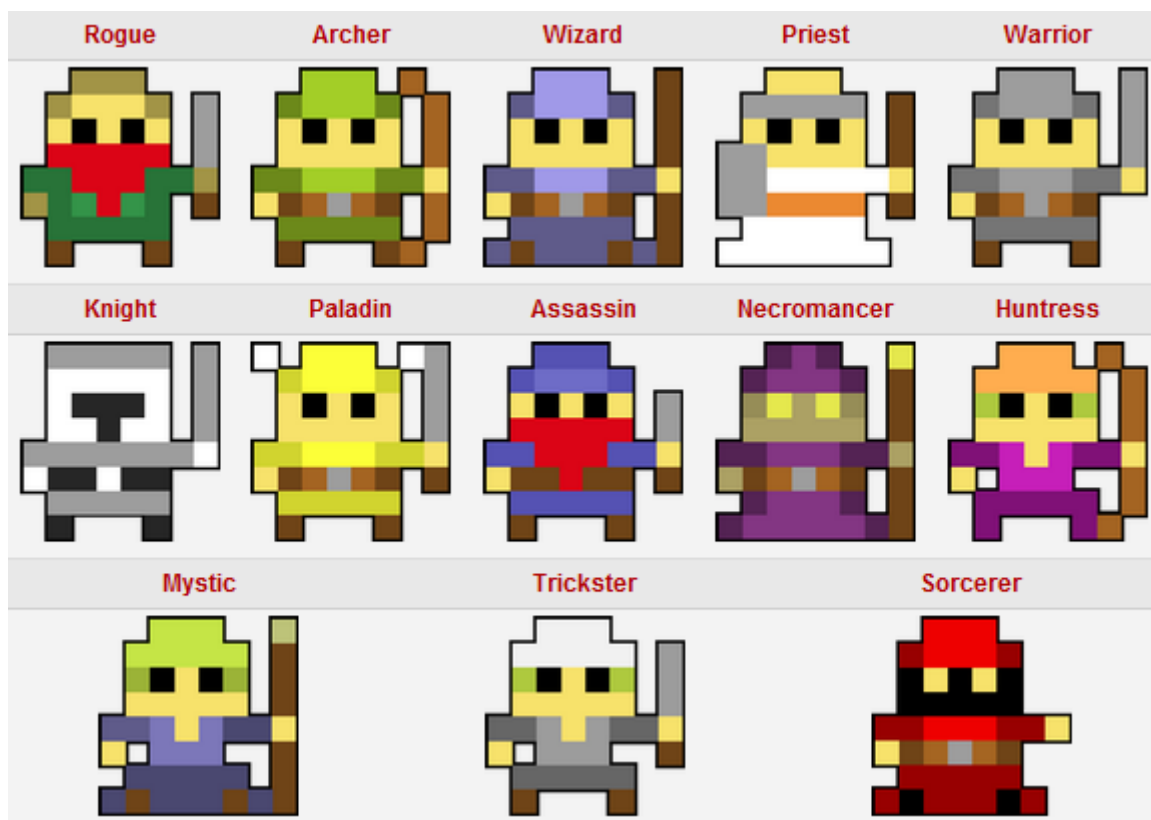
### Wprowadzenie:

Prezentujemy dzieciom grę, początkowo nie zdradzając jej mechanizmu.

### Wykonanie:

#### A. Powołanie do życia duszków oraz tła gry

Dzieci samodzielnie przy pomocy Edytora Obrazu tworzą bohatera gry. Z poniższej listy wybierają postać i używając narzędzia PROSTOKĄT wypełniony oraz WYPEŁNIJ KOLOREM odwzorowują ją.



Duszek będzie poruszał się do przodu, tyłu oraz w prawo i lewo, dlatego należy stworzyć kostiumy, które będą odpowiadały jego zwrotowi w odpowiednim kierunku. Duplikujemy trzy krotnie kostium naszego bohatera i za pomocą narzędzia ZAZNACZ I DUPLIKUJ usuwamy mu oczy bądź



przesuwamy je w prawo/lewo tworząc wrażenie jego ustawienia kolejno: plecami, w prawą/lewą stronę.

Następnie używając narzędzia PRZERZUĆ LEWO-PRAWO (górny prawy róg Edytora Obrazu), tworzymy lustrzane odbicie Duszka, które wzmacnia wrażenie jego obrócenia się w różnych kierunkach.

Czas na tło, które jest Duszkiem (NOWY DUSZEK → NAMALUJ NOWEGO DUSZKA). Dzieci samodzielnie odwzorowują pokazane im tło gry. Tworzą je również w Edytorze Obrazu przy pomocy narzędzia PROSTOKĄT i różnych odcieni zieleni. Rysują drzewa i kwiaty. Następnie duplikują Duszka tło 8 x, tak aby razem z bohaterem gry otrzymać 9 Duszków.

Kolejnym krokiem jest zaimportowanie SCENY. Wybieramy NOWE TŁO → BIBLIOTEKA, następnie dział KOSMOS (menu po lewej stronie) i dwukrotnie klikamy na tło STARS.

## B. Zmienne

Ruch naszego bohatera będzie opierał się na operowaniu zmiennymi. Za pomocą zmiennych będziemy zmieniać wartości współrzędnych, określających położenie naszych Duszków-tła. Tworzymy dwie zmienne x i y. Wybieramy blok DANE → STWÓRZ ZMIENNĄ.

## B. Skrypt TŁA

Ustawiamy pierwszego Duszka-tła w konkretnym miejscu, czyli na środku Podglądu Duszków. Współrzędne środka Podglądu to (0,0). Środek ciężkości Duszka znajduje się na samym jego środku i to właśnie temu punktowi przypisujemy wartość położenia (0,0). Wybieramy blok RUCH → USTAW X NA oraz USTAW Y NA.

Położenie tła będzie zmieniać się w trakcie skryptu, ponieważ zależy nam na uzyskaniu efektu ruchu. Dlatego, nie możemy na sztywno przypisać Duszkowi wartości 0, to będzie jedynie wartość początkowa. Wchodzimy w blok WYRAŻENIA → ... + ..., którą to komendę umieszczamy w oknie powyższych komend bloku RUCHU. W lewym oknie WYRAŻENI umieszczamy zmienne X/Y. W prawym oknie natomiast wpisujemy 0. Zmienne X i Y będą wyzerowywane przy każdym uruchomieniu skryptu, dlatego ich wartość początkowa będzie  $0+0=0$ . Obie komendy mają być wykonywane przez cały czas trwania gry, dlatego umieszczamy je w PĘTLI NIESKOŃCZONEJ, wybieramy KONTROLA → ZAWSZE. Następnie podpinamy komendy pod odpowiednie zdarzenie, czyli ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.



Drugiego Duszka-tła ustawimy już poza Podglądem Duszków. Będzie znajdował się on z jego prawej strony, ale nie będzie widoczny. Szerokość Podglądu Duszków (oznaczana przez oś X) wynosi 480px. Wspominaliśmy o tym, że aby ustawić Duszka w konkretnym położeniu nadajemy odpowiednią wartość jego środkowi. Jeżeli po prawej, niewidocznej już stronie Podglądu Duszków, chcemy ustawić kolejny ekran, musi on mieć szerokość 480px, tak samo jak pierwszy Duszek-tło. Licząc szerokość od jednego środka do drugiego (połowa szerokości Duszka nr1 oraz połowa szerokości Duszka nr 2) wynosi on również 480px.

Dlatego w przypadku Duszka nr 2 wartość x będzie wynosić:  $USTAWA X NA X + 475$ . Odjęliśmy 5px tak,



aby oba tła nachodziły się na siebie (nie było żadnej przerwy między nimi).



Następny Duszek-tła będzie znajdował się po lewej stronie Podglądu Duszków, dlatego teraz wartość współrzędnej x będzie wynosiła -457 (lewa strona osi X jest ujemna).



To samo wykonujemy dla Duszków-tła, które będą znajdować się u góry/dołu Podglądu Duszków. Długość Podglądu wynosi 360, dlatego od środka jedno Duszka do drugiego będziemy mieć również 360. Odejmujemy dodatkowo 5px, ponieważ chcemy, aby Duszki nachodziły na siebie nie tworząc żadnej szczeliny.



Teraz należy ustawić 4 kolejne Duszki-tła w rogach, poza Podglądem Duszków, tak aby stworzyć planszę w kształcie wielkiego prostokąta, którego tylko środek (czyli Podgląd Duszków) jest widoczny. Wartości współrzędnych x i y będą wyglądać w następujący sposób: (475, 355), (475,-355),(-475,355),(-475,-355).

### C. Skrypt bohatera gry

Rozpoczniemy skrypt od zerowania zmiennych przy każdym rozpoczęciu gry, w wyniku czego ustawimy nasze Duszki-tła w odpowiednich miejscach. Wybieramy DANE → USTAW Y/X NA 0 (komendy podpinamy pod odpowiednie ZDARZENIE).

Teraz napiszemy blok komend, który umożliwi nam stworzenie efektu poruszania się naszego Duszka. Blok ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Nasz bohater nie może być nigdy



zastąpiony przez pozostałe Duszki, dlatego kolejną komendą jest WYGLĄD → NA WIERZCH.

Duszek będzie poruszał się dzięki klawiszom (w przykładzie jest to WSAD). W tym celu wykorzystamy czujniki, gdyż poruszający się Duszek jest wtedy czulszy, szybciej reaguje na naciśnięty klawisz. Komendy CZUJNIKÓW umieścimy w INSTRUKCJI WARUNKOWEJ, dlatego wybieramy KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku CZUJNIKI → KIEDY .. NACIŚNIĘTY?. Jeżeli WARUNEK zostanie spełniony, wtedy Duszek zmienia kostium, który ustawia go w odpowiedniej orientacji (pravo/lewo/góra/dół). Wykonujemy to przy pomocy WYGLĄD → ZMIEN KOSTIUM NA ... .

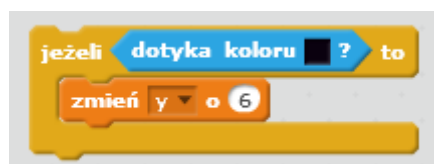
Jeżeli WARUNEK jest spełniony to bohater ma również poruszyć się w odpowiednim kierunku nadanym mu przez klawisz. Otrzymujemy ten efekt poruszając każdego Duszka-tła w przeciwnym kierunku, w którym porusza się bohater. Na przykład: nasz główny Duszek chce przesunąć się do góry, wszystkie Duszki-tła poruszają się w dół. Dlatego wtedy zmieniamy współrzędną Y Duszków o -3.



Analogicznie tworzymy dla pozostałych 3 kierunków świata, pamiętając o tym, że Duszki-tła zawsze poruszają się w przeciwną stronę co bohater gry. Wszystkie cztery INSTRUKCJE WARUNKOWE umieszczamy w PĘTLI NIESKOŃCZONEJ tak, aby wykonywały się przez cały czas trwania gry. Wybieramy KONTROLA → ZAWSZE.

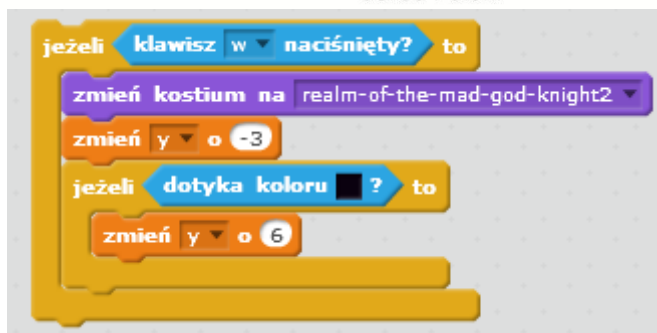
Nasza plansza gry złożona z 9 elementów ma przecież swój koniec. Aby Duszek nie poruszał się po białej, pustej planszy, w pierwszym etapie tworzenia gry zaimportowaliśmy SCENĘ → STARS. Jednak nie chcemy, aby Duszek poruszał się po gwiazdach, ma poruszać się tylko po naszej planecie. Dlatego dodamy kolejny WARUNEK, który nie pozwoli wychodzić mu poza planszę oraz nie pozwoli chodzić po niektórych krzaczkach-przeszkodach znajdujących się na niej. W tym celu ponownie wykorzystujemy KONTROLA → JEŻELI .. TO. W oknie INSTRUKCJI, jako WARUNEK, umieścimy CZUJNIK → DOTYKA KOLORU ...?

Jeżeli WARUNEK jest prawdziwy, wtedy nasz bohater musi cofnąć się o 6 kroków w przeciwnym kierunku z którego przyszedł – oznacza to wpisanie LICZBY ODWROTNEJ (z przeciwnym znakiem). Cofamy się o 6 kroków, ponieważ podwajamy wartość, z którą porusza się Duszek tak, aby cofnięcie było widoczne i uniknąć efektu „blokowania” się Duszka.



Instrukcję zagnieźdźmy (umieszczamy wewnątrz) instrukcji wcześniej już stworzonej.







## Zajęcia 11: Quiz

### Zadanie:

Wykonanie gry „Quiz”, która dzieli się na cztery fazy:

- zaproszenie do gry
- gra
- zmienna
- wyniki

### Cel:

Nauka takich pojęć programistycznych jak:

- zdarzenia (generowanie, obsługa);
- zmienna.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy.

### Wykonanie:

#### A. Skrypt głównego duszka

Pierwszym blokiem komend, będą bloki dialogu wprowadzające nas w grę (WYGLĄD → POWIEDZ). Aby rozmowa mogła się rozpocząć, musimy podpiąć komendy pod komendę rozpoczynającą skrypt (ZDARZENIA → KIEDY KLIKNIĘTO). Ostatnie pytanie duszka brzmi: „Kliknij na mnie, kiedy będziesz gotów”.



Następnie wybieramy komendę: ZDARZENIA → KIEDY DUSZEK KLIKNIĘTY. Jest to reakcja (działanie) naszego skryptu na ostatnie zapytanie dialogu, która daje możliwość użytkownikowi na przygotowanie się i rozpoczęcie gry.

Zdarzenia są to akcje generowane przez użytkownika naszego programu (zdarzeniami są: kliknięcie myszy, użycie klawiatury – są to zewnętrzne urządzenia oraz rozpoczęcie skryptu). Naszym zadaniem jest obsługa konkretnego zdarzenia, czyli reakcja skryptu na akcję (działanie) użytkownika.

To jest nasz główny blok, w którym będzie zapisane działanie quizu. Pod komendą KIEDY DUSZEK KLIKNIĘTY będzie znajdowało się 5 pytań. Pytania będą zadawane komendami (oknami) dialogowymi WYGLĄD → POWIEDZ. Po zadaniu pytania użytkownik wybiera odpowiedź z czterech możliwych, klikając na odpowiedni przycisk (A, B, C i D).



## B. Zmienne

Najważniejszym elementem gry są zmienne. To one pozwalają na wybranie odpowiedzi quizu. Przechodzimy do bloku DANE, tam deklarujemy trzy zmienne (STWÓRZ ZMIENNĄ → wprowadź nazwę zmiennych: „KLIK”, „ODPOWIEDZ”, „PUNKTY”).

Razem z nadaniem nazwy zmiennej możemy określić zakres jej widoczności: lokalnie (tylko dla tego duszka) oraz globalnie (dla wszystkich duszków). Zmienna lokalna (tylko dla tego duszka) będzie widoczna (i będzie działać) tylko na liście bloku DANE dla konkretnego duszka, z pozycji pozostałych duszków nie będzie istnieć. Zmienna globalna (dla wszystkich duszków) jest widoczna z pozycji każdego duszka i też dla każdego będzie działać.



Teraz, po deklaracji oraz definicji zmiennej (nadaniu nazwy zmiennej oraz powołaniu jej do istnienia – stworzyliśmy ją czyli przydzieliliśmy jej miejsce w pamięci komputera, gdzie może istnieć)



należy ją zainicjalizować. Inicjalizacja zmiennej to nadanie jej wartości początkowej w naszym przypadku jest to wartość zero (DANE → USTAW ... NA 0).



### C. Skrypty przycisków (duszków-przycisków)

Skrypt każdego duszka-przycisku składa się z jednego bloku. Przyciski są duszkami, aby można było określić, kiedy duszek został kliknięty. Do tego celu używamy zmiennej KLIK. Zmienna ta, będzie ustawiana na wartość 1 w momencie kliknięcia na duszka (ZDARZENIA → KIEDY KLIKNIĘTO DUSZKA, następnie podpinamy pod zdarzenie komendy zmiennych DANE → USTAW KLIK NA 1)



Aby skrypt był w stanie odróżnić od siebie przyciski będące różnymi odpowiedziami (A, B, C, D) niezbędna jest zmienna, którą ustawiamy dla: przycisku A = 1, przycisku B = 2, przycisku C = 3, przycisku D = 4. Dzięki tym wartościom liczbowym nasze odpowiedzi są rozróżnialne dla skryptu. My użyjemy tutaj zmiennej „ODPOWIEDŹ”.



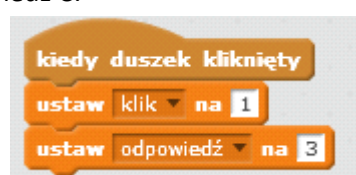
### D. Skrypt głównego duszka

Zmienne są już zadeklarowane, dlatego możemy przystąpić do działania związanego z odpowiedziami na zadane w naszym quizie pytania. Po zapytaniu następuje moment oczekiwania skryptu (nasłuchiwanie) na zdarzenie, czyli w naszym przypadku wybór konkretnej odpowiedzi (kliknięcie duszka-przycisku). Przypominam, że kliknięcie na naszego duszka-przycisku rozwiązaliśmy przy pomocy zmiennej KLIK i ODPOWIEDŹ, które zmieniają swoje wartości w zależności od klikniętej odpowiedzi.

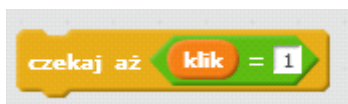
Odpowiedź A:



Odpowiedź C:



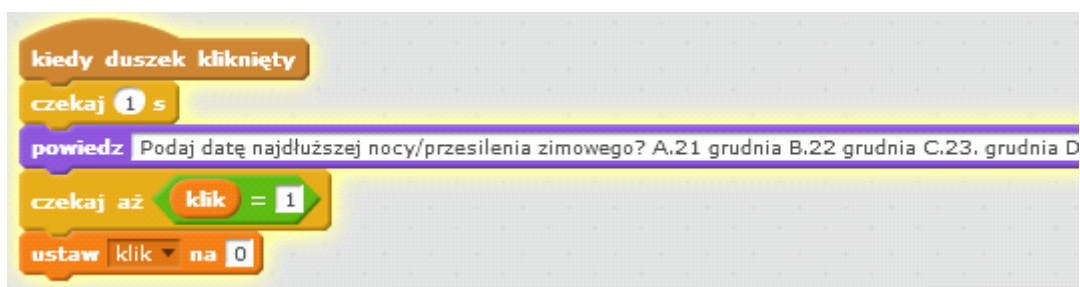
Na wybór odpowiedzi, skrypt czeka przy pomocy komendy bloczka KONTROLA → CZEKAJ AŻ.



Następnie zerujemy ponownie zmienną „KLIK”. Musimy pamiętać o wykonaniu tej komendy po każdym nasłuchaniu zdarzenia (po każdym kliknięciu przycisku – wybraniu odpowiedzi). Postępujemy w ten sposób tak, aby zmienna po przyjęciu wartości 1 (stan: kliknięto) wróciła do stanu



początkowego, czyli do wartości zero (stan: nie kliknięto).



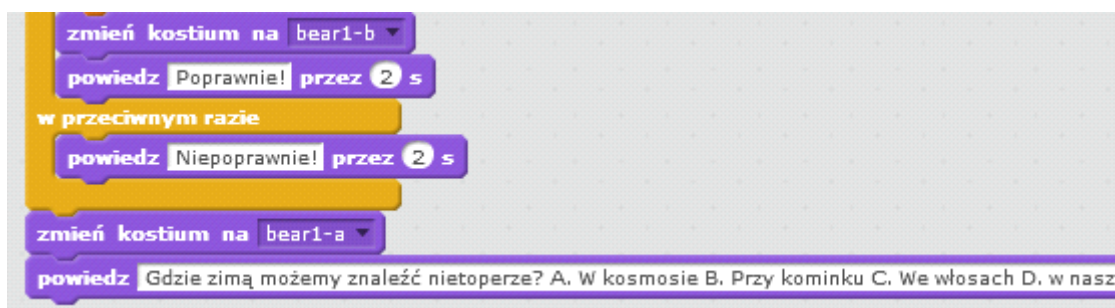
Kolejnym krokiem w programie jest sprawdzenie, która odpowiedź została kliknięta. Wykonujemy to zadanie używając instrukcji warunkowej (blok KONTROLA → JEŻELI... W PRZECIWNYM RAZIE).

Instrukcja warunkowa jest to zabieg programistyczny, który prezentuje dwie instrukcje. Wykonanie poszczególnej instrukcji jest uzależnione od prawdziwości bądź fałszu zdefiniowanego przez nas wyrażenia logicznego. Jeżeli wyrażenie jest prawdziwe wykonuje się instrukcja zawarta w pętli JEŻELI, natomiast w momencie, w którym wyrażenie jest fałszywe wykonuje się instrukcja zawarta w pętli W PRZECIWNYM RAZIE.

W naszym przypadku wyrażeniem logicznym jest sprawdzenie wartości zmiennej ODPOWIEDŹ. Jeżeli wartość zmiennej ODPOWIEDŹ ma równać się przykładowo 3 (a jest to zależne od poprawnej odpowiedzi do konkretnego pytania) i taką też wartość posiada, wtedy główny duszek informuje nas (komenda POWIEDZ), o tym, że odpowiedź jest poprawna („Poprawnie!”). W przeciwnym przypadku, czyli jeżeli wartość zmiennej ODPOWIEDŹ będzie różna od 3, główny duszek poinformuje nas o naszym błędzie („Niepoprawnie!”).



Dodatkowo, po każdej informacji o poprawności naszej odpowiedzi dodajmy animację głównego duszka. Animacja polegać będzie na zmianie kostiumu duszka. Sprawdzamy dostępne kostiumy w zakładce KOSTIUMY, następnie wybieramy z bloku WYGLĄD komendę ZMIEN KOSTIUM NA. Rozwijamy listę znajdującą się po prawej stronie komendy i wybieramy kostium.

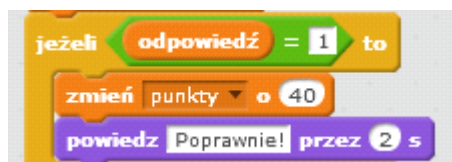




Musimy pamiętać, że przed zadaniem kolejnego pytania, należy zmienić kostium z powrotem na kostium początkowy (patrz obrazek wyżej).

## E. Wyniki

Ostatnim niezbędnym elementem gry jest punktacja i jej obliczanie. W każdej instrukcji warunkowej, po poprawnej odpowiedzi naliczamy punkty DANE → ZMIEN PUNKTY O... .



Kolejnym krokiem, po skończeniu quizu jest podsumowanie punktacji. Chodzi o to, aby określić czy wartość zmiennej PUNKTY jest większa niż 30 lub czy jest równa tej wartości. Jeżeli wyrażenie to jest prawdziwe (czyli wartość zmiennej wynosi więcej niż 30 lub się jej równa), wtedy główny duszek informuje nas o tym mówiąc: „Gratulacje, wygrałeś!”. W przeciwnym przypadku (wartość PUNKTY jest mniejsza niż 30) główny duszek mówi: „Przykro mi, przegrałeś.”. Wykonujemy to zadanie w instrukcji warunkowej (blok KONTROLA → JEŻELI... W PRZECIWNYM RAZIE). W puste okno komendy wkładamy wyrażenie logiczne LUB z bloku WYRAŻENIA (jest to operator ALTERNATYWY). Następnie w puste okna wyrażenia LUB, wkładamy komendy ze znakami większości oraz równości.



Skrypt kończymy komendą ZATRZYMAJ WSZYSTKO z bloku KONTROLA.





## Zajęcia 12: Arkanoid

### Zadanie:

Wykonanie gry „Arkanoid”, która dzieli kod gry, w zależności od duszka na:

- paletkę,
- piłkę,
- cegły,
- klony cegieł,
- piłki-życia

### Cel:

Utrwalenie takich pojęć programistycznych jak:

- współrzędne,
- zmienna.

### Wprowadzenie:

Prezentujemy dzieciom grę. Uczniowie wyróżniają elementy gry, nazywają je i próbują rozszyfrować najważniejsze użyte komendy.

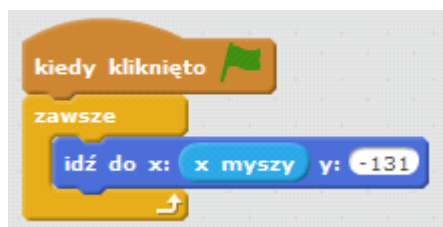
### Wykonanie:

#### A. Paletka

Pierwszym zaprogramowanym przez nas duszkiem będzie sterowalna paletka. Zrealizujemy to zadanie, poruszając nią wzdłuż osi X, wartość osi Y będzie natomiast stała (niezmienna). Paletką będziemy sterować w momencie rozpoczęcia skryptu (ZDARZENIE: KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ). Użyjemy do tego komendy ZDARZENIA -> KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Sterowanie paletką będzie odbywać się przez cały czas trwania gry, dlatego pod powyższe zdarzenie podpinamy NIESKOŃCZONĄ PĘTLĘ. Blok KONTROLA -> ZAWSZE.

Teraz ustawiamy naszą paletkę w odpowiednim punkcie PODGLĄDU DUSZKÓW, czyli nadajemy jej WSPÓŁRZĘDNE. Pamiętamy, że wartość współrzędnej y jest stała, natomiast współrzędna x będzie zmienną.

Wybieramy RUCH -> IDŹ DO x ... y ... . W oknie powyższej komendy umieszczamy z bloku CZUJNIKI -> X MYSZY. Jest to zmienna, do której będzie przypisywana wartość współrzędnej x wczytana z aktualnego położenia myszy.



#### B. Cegły

Głównymi elementami naszej gry jest mur kolorowych cegieł, który będzie znikał w momencie trafienia w niego piłką. Cegły od momentu rozpoczęcia gry do jej zakończenia będą miały przypisane konkretne położenie (współrzędne).

Wybieramy zdarzenie, które zainicjuje pojawienie się naszych cegieł w odpowiednim miejscu. Blok ZDARZENIA -> KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Teraz nadamy cegle współrzędne. Użyjemy do



tego celu RUCH -> IDŹ DO x ... y ... . Po rozpoczęciu gry cegła ma się pojawić, dlatego WYGLĄD -> POKAŻ, którą to komendę podpinamy jako kolejną.

Następnym krokiem jest stworzenie odpowiedniego warunku, który pozwoli po uderzeniu piłki na zniknięcie cegły. Wykonamy to przy pomocy INSTRUKCJI WARUNKOWEJ. Blok KONTROLA -> JEŻELI ... TO . W oknie INSTRUKCJI WARUNKOWEJ umieszczamy WARUNEK, czyli z bloku CZUJNIKI wybieramy komendę DOTYKA PIŁKA ? TO. W momencie, kiedy warunek jest prawdziwy (spełniony), czyli piłka dotyka cegły, ukrywamy ją. W tym celu wykorzystamy WYGLĄD -> UKRYJ. Na wykonywanie powyższego WARUNKU przez cały czas trwania skryptu pozwoli nam NIESKOŃCZONA PĘTLA, blok KONTROLA -> ZAWSZE.



### C. Klony cegieł

Aby nie tworzyć kolejnych nowych duszków, aby każda nowa cegła nie była następnym obiektem, wykorzystamy funkcję KLONOWANIA. Klony będziemy tworzyć na samym początku gry, dlatego wybieramy ZDARZENIA -> KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.

Przy pomocy zmiennej liczbowej, każdej kolejnej cegle będziemy nadawać inną wartość współrzędnej x (przy identycznej wartości y, czyli cegły będą znajdować się na tej samej wysokości) tak, aby powstała pierwsza linia muru. Dlatego, tworzymy zmienną dla każdego duszka oddzielnie (lokalnie) (DANE -> STWÓRZ ZMIENNĄ) "POZYCJAX". Następnie zmiennej nadajemy wartość współrzędnej x pierwszej cegły. Blok DANE -> USTAW POZYCJAX NA -205.

Następnie przy pomocy PĘTLI ITERACYJNEJ (LICZBOWEJ) wykonamy 11 razy klonowanie. Blok KONTROLA -> POWTÓRZ 11 RAZY.

Teraz wybieramy komendę SKOLONUJ SIEBIE z bloku KONTROLA, którą umieszczamy wewnątrz PĘTLI. Poniżej podpinamy DANE -> ZMIENŃ POZYCJAX O 40. Za każdym wykonaniem pętli zwiększamy wartość zmiennej, ponieważ każda kolejna cegła będzie przesunięta o 40px.



Teraz stworzonym już klonom, możemy nadać położenie. Blok komend klonów, rozpoczniemy od komendy KONTROLA -> KIEDY ZACZYNAM JAKO KLON. Każdy kolejny klon będzie wysunięty na wierzch, dlatego pod powyższą komendę podpinamy WYGLĄD -> NA WIERZCH.

Ustalamy pozycję klona po przez RUCH -> IDŹ DO X ... Y ... . Współrzędnej x przypisujemy





wartość zmiennej POZYCJAX. Dlatego w oknie komendy IDŹ DO X ... Y ... umieszczamy POZYCJAX. Wartość y pozostaje niezmienna dla wszystkich cegieł danego koloru, dlatego przepisujemy tę wartość z już ustalonych współrzędnych pierwszej cegły.

Tak samo jak w przypadku pierwszej cegły, klony mają znikać za każdym razem, kiedy trafia w nie piłka. Dlatego stworzymy WARUNEK, który na to pozwoli. Wybieramy INSTRUKCJĘ WARUNKOWĄ, blok KONTROLA -> JEŻELI .. TO. Jako warunek wybierzemy CZUJNIKI -> DOTYKA PIŁKA?, którą to komendę umieścimy w oknie INSTRUKCJI WARUNKOWEJ.

Jeżeli powyższy WARUNEK jest spełniony, czyli piłka dotyka cegły KONTROLA -> USUŃ TEGO KLONA. Aby całość wykonywała się przez cały czas trwania gry, powyższe komendy umieszczamy w NIESKOŃCZONEJ PĘTLI. Blok KONTROLA -> ZAWSZE.



Analogicznie wykonujemy pozostałe rzędy cegieł, zmieniając tylko WSPÓŁRZĘDNE osi y tak, aby każdy kolejny znajdował się odrobinę poniżej.

## D. Piłka

Czas, aby wprawić w ruch naszą piłkę. W momencie rozpoczęcia skryptu ustawimy piłkę "w powietrzu" oraz nadamy jej kierunek ruchu. W pierwszej kolejności wybieramy ZDARZENIA -> KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ.

Kolejnym krokiem jest stworzenie zmiennej "Życia", która będzie określać ilość piłek, którą mamy to wykorzystania w trakcie gry. Blok DANE -> STWÓRZ ZMIENNĄ ŻYCIA.

Teraz zmiennej należy nadać wartość maksymalną, którą będzie oznaczała ile piłek mamy to wykorzystania. Pod zdarzenie podpinamy DANE -> USTAW ŻYCIA NA 3.

Następnie ustawimy piłkę na starcie "w powietrzu". Dlatego wybieramy RUCH -> IDŹ DO X 0 Y 0, są to współrzędne środka OKNA PODGLĄDU. Kolejnym krokiem jest nadanie początkowego kierunku opadania piłki. Blok RUCH -> USTAW KIERUNEK NA ... .

Aby przy każdym uruchomieniu gry piłka nie leciała w tę samą stronę, będziemy losować wartość kierunku. W oknie komendy USTAW KIERUNEK NA ... umieszczamy komendę LOSUJ OD 165 DO 180 z bloku WYRAŻENIA. Następnie sprawiamy, aby piłka była widoczna (będzie ona znikać po każdej utracie życia). Blok WYGLĄD -> POKAŻ.





Pozwalamy graczowi, przygotować się do rozgrywki, dlatego piłka nie będzie odbijać się od momentu rozpoczęcia skryptu (KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ), tylko wtedy, gdy naciśniemy klawisz spacja. Wybieramy z bloku ZDARZENIA -> KIEDY KŁAWISZ SPACJA NACIŚNIĘTY. Jest to zdarzenie rozpoczynające blok komend, który będzie definiował zachowanie piłki w momencie kontaktu z pozostałymi duszkami.

Przez cały czas od momentu naciśnięcia klawisz piłka ma się poruszać. Dlatego pod zdarzenie podpinamy NIESKOŃCZONĄ PĘTLĘ, blok KONTROLA -> ZAWSZE. Piłka będzie przesuwana się z prędkością 5 kroków. Blok RUCH -> PRZESUŃ O 5 KROKÓW. Aby uniknąć "wypadania" piłki poza ekran PODGLĄDU użyjemy RUCH -> JEŻELI NA BRZEGU, ODBIJ SIĘ.

Następnie tworzymy warunek, który pozwoli na zmienny kąt odbicia piłki od paletki. Wybieramy INSTRUKCJĘ WARUNKOWĄ, blok KONTROLA -> JEŻELI, .. TO. Jako WARUNEK wykorzystamy komendę CZUJNIKI -> DOTYKA PALETKA?, którą umieszczamy w oknie powyższej INSTRUKCJI. We wnętrzu WARUNKU umieścimy losowy kierunek odbicia piłki. Blok RUCH -> USTAW KIERUNEK NA .. .

Wartość kierunku będzie nam nieznana, dlatego w oknie powyższej komendy umieszczamy WYRAŻENIA -> LOSUJ OD -60 DO 60. Przy każdym odbiciu piłkę będziemy delikatnie obracać, dlatego poniżej podpinamy komendę OBRÓĆ W PRAWO O 10 STOPNI, z bloku RUCH.

Tworzymy kolejną INSTRUKCJĘ WARUNKOWĄ analogicznie do powyższej. Zmieniamy WARUNEK wykonania się komend w jej wnętrzu. Będzie to dotknięcie cegieł żółtych albo zielonych. Ponownie ustawiamy kierunek odbicia piłki. Tym razem jednak piłka będzie opadać. Duszek i tym razem będzie się delikatnie obracał.

Tworzymy trzecią już z kolei INSTRUKCJĘ WARUNKOWĄ. Tym razem komenda będzie dotyczyć dotknięcia duszka "CZELUŚĆ" znajdującego się przy dolnej krawędzi PODGLĄDU. CZELUŚĆ jest elementem gry odbierającym graczowi życie.

W tym celu analogicznie tworzymy INSTRUKCJĘ WARUNKOWĄ, wraz z WARUNKIEM dotknięcia CZELUŚCI. Jeżeli piłka trafi na wymienianego duszka, ma zniknąć. Dlatego we wnętrzu INSTRUKCJI, umieszczamy komendę WYGLĄD -> UKRYJ. Następnie odejmujemy graczowi jedno życie. Wybieramy DANE -> ZMIENŃ ŻYCIE O -1.

Ponownie ustawiamy piłkę w "powietrzu" (rozpoczynamy grę w kolejnym życiu), dlatego wybieramy RUCH -> IDŹ DO X: 0 Y: 0. Dajemy graczowi chwilę na odsapnięcie po przegranej (i otarcie łez) i wybieramy KONTROLA -> CZEKAJ 3S. Piłka, która po utracie życia stała się niewidoczna, teraz się pojawia. Blok WYGLĄD -> POKAŻ. Ponownie dajemy graczowi chwilę na przygotowanie się. Blok KONTROLA -> CZEKAJ 2S.

Ostatnią INSTRUKCJĄ WARUNKOWĄ naszej gry, są komendy kończące skrypt po utracie wszystkich żyć. Tak jak powyżej tworzymy WARUNEK: JEŻELI ŻYCIA = 0 TO. We wnętrzu INSTRUKCJI umieszczamy KONTROLA -> ZATRZYMAJ WSZYTSKO.



## E. Piłki-życia

Piłki-życia są to trzy kostiumy duszka, które będą informować gracza, ile żyć mu pozostało. Życia będą widoczne od momentu rozpoczęcia skryptu, dlatego wybieramy ZDARZENIA -> KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Ustawiamy duszka w prawym dolnym rogu PODGLĄDU. W tym celu użyjemy RUCH -> IDŹ DO X: ... Y: ... .

W zależności od ilości żyć, czyli od wartości zmiennej ŻYCIA będziemy zmieniać kostium duszka, tak aby uzyskać efekt znikających piłek. W tym celu stworzymy trzy INSTRUKCJE WARUNKOWE, które będą sprawdzane przez cały czas wykonywania skryptu. Wybieramy KONTROLA -> ZAWSZE.

Następnie tworzymy INSTRUKCJĘ WARUNKOWĄ, której będzie odpowiednia wartość zmiennej ŻYCIA. W zależności ile żyć nam pozostało, będziemy wyświetlać kostium o takiej samej ilości piłek.



```
kiedy kliknięto
  idź do x: 202 y: -155
  zawsze
    jeżeli życia = 3 to
      zmień kostium na costume1
    jeżeli życia = 2 to
      zmień kostium na costume2
    jeżeli życia = 1 to
      zmień kostium na costume3
```



## Zajęcia 13 i 14: Wyścigi

---

### Zadanie:

Wykonanie gry „Wyścigi” polegającej na pokonaniu trasy, którą tworzą Duszki-Tory poruszające się wzdłuż osi Y układu współrzędnych. Główny Duszek będzie poruszał się wzdłuż osi X układu współrzędnych.

### Cel:

Opanowanie operowania współrzędnymi układu kartezjańskiego podkreślając fakt iż jest on nieskończony. Duszki w takim wypadku mogą znajdować się poza obszarem PODGLĄDU DUSZKÓW będąc nie widocznymi, jeżeli nadamy im odpowiednie współrzędne.

### Komentarze:

Zajęcia w zależności od czasu jakim dysponuje Instruktor, mogą zostać podzielone na dwa, a nawet trzy zajęcia. W takim wypadku należy zapisywać wersje gier lub/i załadować na serwer Scratch'a.

Aby udało się dzieciom zaobserwować efekt płynięcia toru wyścigowego wcześniej niż w trakcie programowania Głównego Duszka należy zaprogramować działanie zmiennej płynięcie (czyli nie w kolejności zapisanego skryptu).

Propozycje rozszerzenia zajęć znajdują się na końcu skryptu.

### Wprowadzenie:

Prezentujemy dzieciom grę początkowo nie zdradzając jej mechanizmu. Zadajemy pytania o zatasowane bloki komend oraz zabiegi programistyczne.

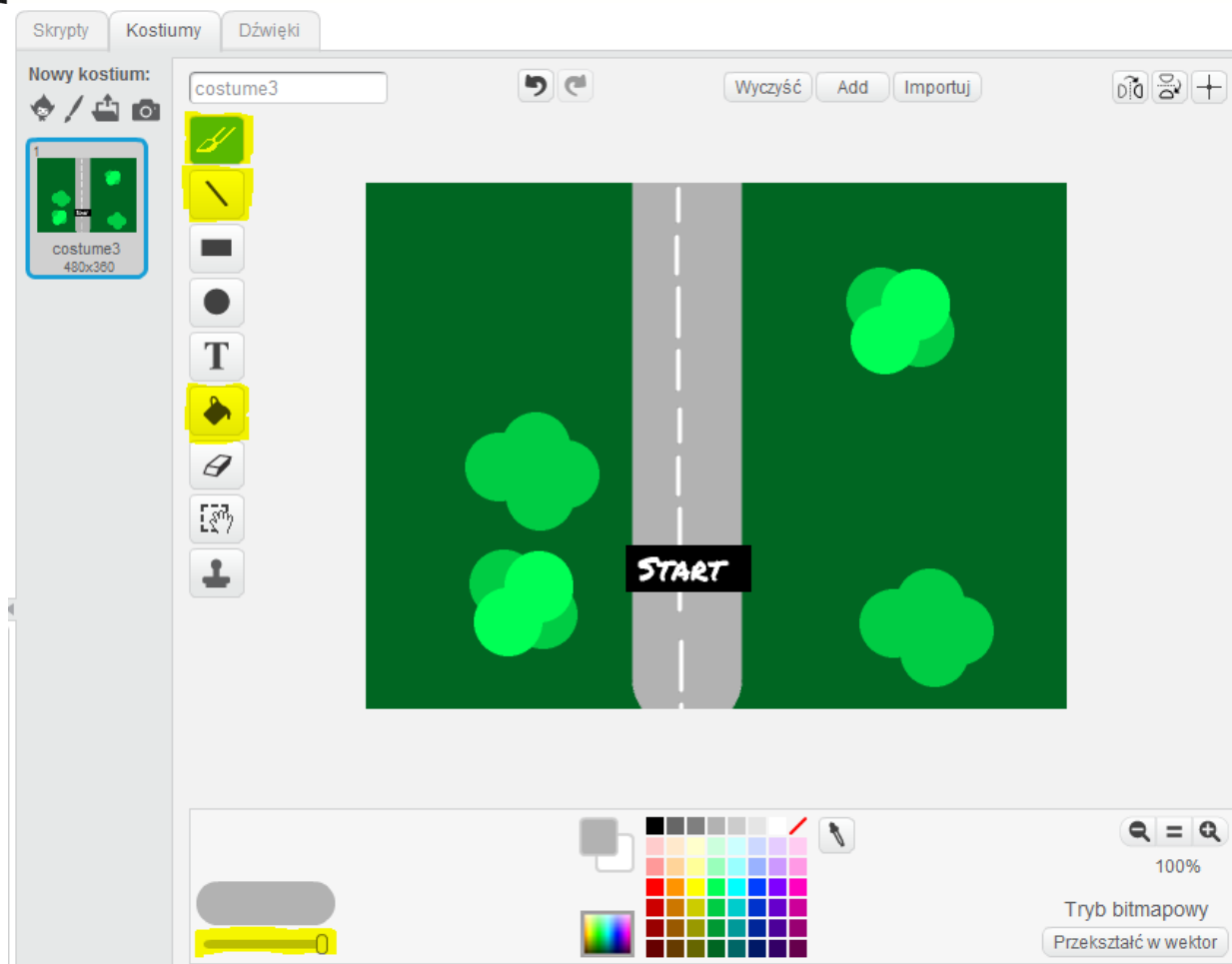
### Wykonanie:

#### A. Powołanie do życia duszków oraz tła gry

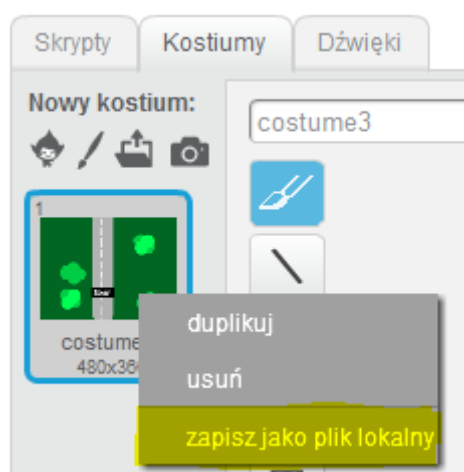
1. Dzieci samodzielnie przy pomocy Edytora Obrazu rysują tor wyścigowy składający się z pięciu Duszków, które wypełniać mają cały ekran. Do tego celu użyjemy takich narzędzi Edytora Obrazu (pasek narzędzi po lewej stronie) jak:
  - PĘDZEL o maksymalnej szerokości, którym wyznaczymy drogę,
  - LINIA, która wyznaczy nam pasy ruchu drogowego (rysując przyciskamy SHIFT, tak aby, uzyskać linię prostą),
  - WYPEŁNIJ KOLOREM, które to narzędzie pozwoli na wypełnienie dwóch części planszy rozdzielonych drogą,
  - PĘDZEL, który pozwoli na narysowanie drzew poprzez nakładanie na siebie czterech kół (rzut drzewa z góry).

UWAGA: Ze względu na „humory” Scratcha zapisujemy każdego narysowanego Duszka w folderze na pulpicie tak, aby nie stracić naszej pracy (nazwa folderu: imie\_nazwisko\_Scratch). Tam też zapisujemy nasze projekty.

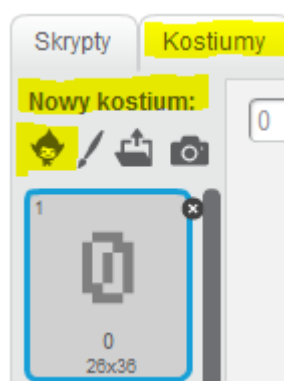
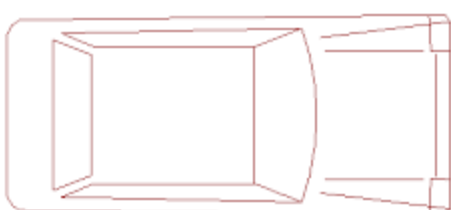
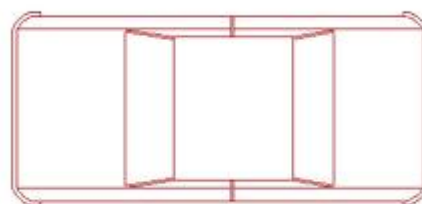
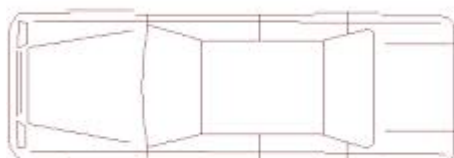
UWAGA: W zależności od upodobań Instruktora tworzymy wszystkie duszki oraz tło na samym początku, bądź po stworzeniu Duszka zapisujemy od razu jego skrypt (przeplatamy programowanie na zmianę z rysowaniem).



Wchodzimy w zakładkę KOSTIUMY. Następnie klikamy prawym przyciskiem na kostium, który chcemy zapisać i wybieramy opcję ZAPISZ JAKO PLIK LOKALNY.



Dzieci samodzielnie rysują dwa samochody, bądź przerysowują szablony udostępnione im przez Instruktora.



Duszki-cyfry: każda z trzech cyfr będzie zawierała do 9 kostiumów, którymi będą cyfry od 0-9. Duszki importujemy z BIBLIOTEKI pamiętając, że musimy znajdować się wtedy w zakładce KOSTIUMY. Z menu znajdującego się po lewej stronie wybieramy kategorię PRZEDMIOTY, a następnie importujemy kostiumy.

### Biblioteka kostiumów

**Kategoria**

- Wszystko
- Zwierzęta
- Fikcja
- Letters
- Ludzie
- Przedmioty**
- Transport

**Motyw**

- Zamek
- Miasto

P-pixel	Q-pixel
V-pixel	W-pixel



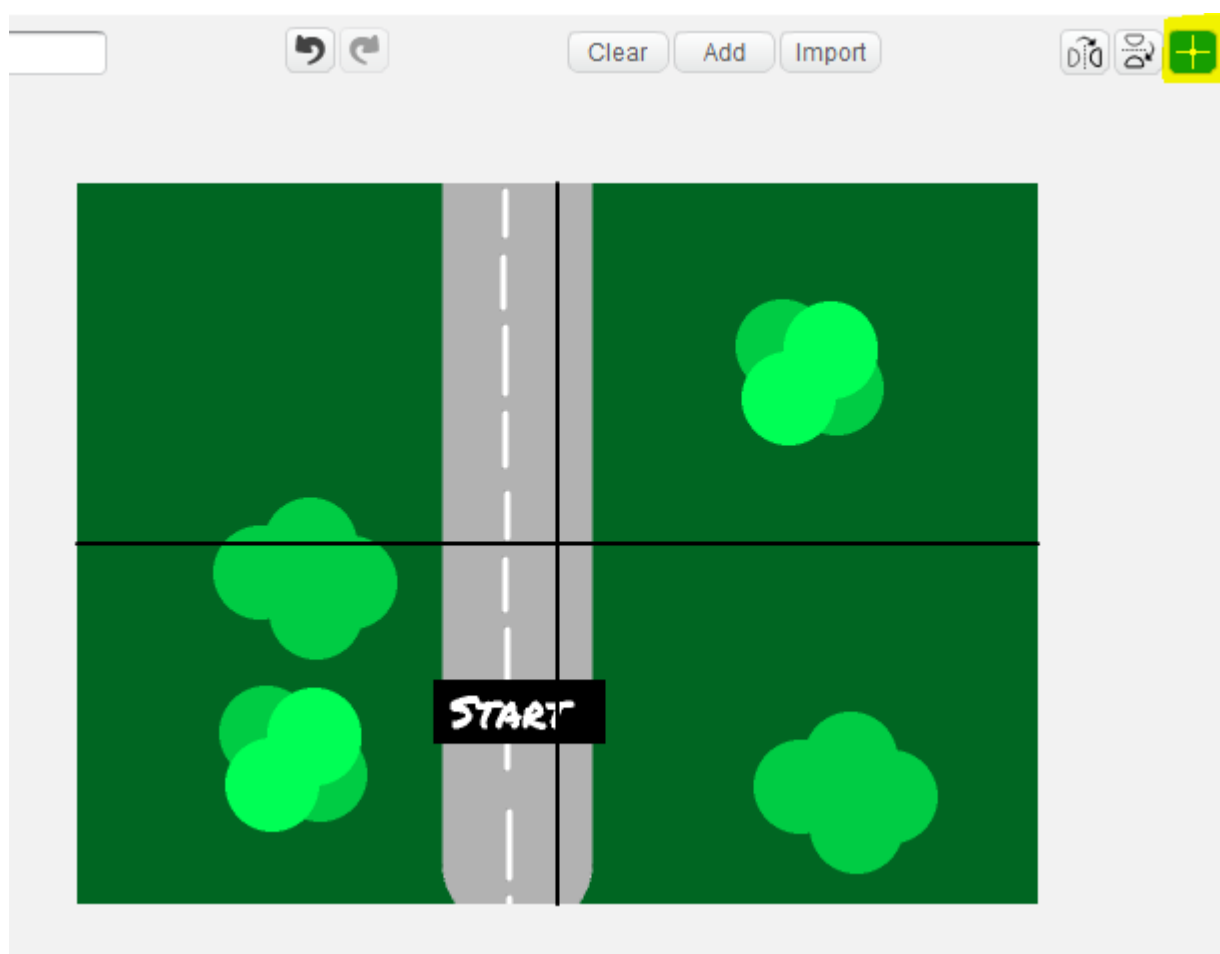
## B. Płynący tor wyścigowy

### B1. Tor 1

Kiedy Duszki toru wyścigowego zostaną narysowane należy je poukładać w taki sposób, aby utworzyły drogę. To one przesuwając się wzdłuż osi Y stworzą wrażenie, że samochód jedzie, podczas gdy tak naprawdę nie będzie się przemieszczał (będzie poruszał się w lewo/prawo wzdłuż osi X).

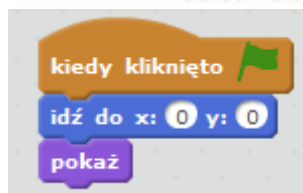
Ustawmy teraz każdą z pięciu plansz toru na swoim miejscu.

Odcinek startowy toru ustawiamy prawie na samym środku Podglądu Duszków (pytam dzieci jaki to punkt? Środek to punkt (0,0)). Ustawimy środek ciężkości kostiumu Duszka właśnie w tym punkcie. Jeżeli nie udało nam się, oznacza to, że przesunięty został środek ciężkości i należy go poprawić w Edytorze Obrazu (zakładka KOSTIUMY).



Ustawiamy naszego Duszka w punkcie (0,0) przy pomocy komendy bloku RUCH → IDŹ DO x: 0 y: 0. Podpinamy powyższą komendę pod ZDARZENIA → KIEDY ZIELONA FLAGA KLIKNIĘTO tak, aby nasza droga układała się na nowo, przy każdym uruchomieniu gry. Wybieramy również WYGLĄD → POKAŻ, aby trasa była widoczna w momencie pojawienia się w obrębie Podglądu Duszków oraz znikła, gdy już przez Podgląd przepływie.





Pojawił się pierwszy odcinek drogi, ale nie od razu rozpoczniemy grę. Dajmy czas graczowi, niech sam zadecyduje kiedy ją rozpocząć (pytamy dzieci: jak to zrobić?)

Wykorzystamy do tego możliwość nadania komunikatu przy pomocy komendy ZDARZENIA → NADAJ START (tworzymy wcześniej komunikat start, bądź wybieramy już udostępniony MESSAGE1). Powyższą komendę podpinamy pod odpowiednie ZDARZENIE, czyli przyciśnięcie klawisza spacja. Wybieramy ZDARZENIA → KIEDY KLAWISZ SPACJA NACIŚNIĘTY.



Po nadaniu wiadomości wszystkie plansze będą mogły zacząć się poruszać w tym samym momencie. Wybieramy ZDARZENIA → KIEDY OTRZYMAM START.

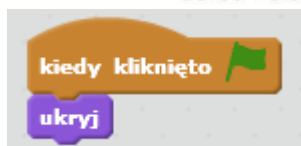
Aby zrealizować, to co zostało opisane powyżej, potrzebujemy ZMIENNĄ „PŁYNIĘCIE”, która pozwoli przesuwać się wszystkim Duszkom równocześnie wzdłuż osi Y, zmniejszając swoją pozycję o -2px. Wybieramy blok DANE → STWÓRZ ZMIENNĄ. Pojawia się okno, w którym wpisujemy nazwę zmiennej. Poniżej możemy wybrać czy zakres widoczności zmiennej jest GLOBALNY (widoczny DLA WSZYSTKICH DUSZKÓW) czy tylko LOKALNY (widoczny TYLKO DLA TEGO DUSZKA). Akceptujemy utworzenie zmiennej klikając ok. Zmienną „PŁYNIĘCIE” umieszczamy wewnątrz komendy bloku RUCH → USTAW Y NA ... . Umieszczamy całość w PĘTLI ITERACYJNEJ, która będzie powtarzała ustawianie Duszka w pozycji Y (zmienna PŁYNIĘCIE nadawać będzie wartość współrzędnej y) 2000 razy. Wykonywać będzie się to tyle razy, ponieważ ZMIENNA nie w każdym kroku będzie zmieniała swoją wartość. W momencie, gdy samochód będzie poruszał się wzdłuż osi X wartość ZMIENNEJ przez tę chwilę będzie taka sama, natomiast ITERACJA PĘTLI będzie się stale przebiegać (patrz: SKYPT GŁÓWNEGO DUSZKA).

Pod PĘTLĘ podpinamy komendę bloku WYGLĄD → UKRYJ tak, aby Duszek zniknął po przepłynięciu przez Podgląd Duszków.

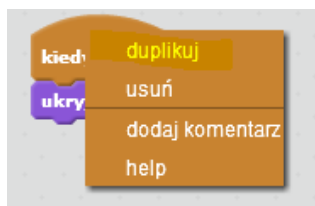


## B2. Tor 2-5

Kolejne tory, będą ukryte w momencie uruchomienia skryptu. Wybieramy blok ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie blok WYGLĄD → UKRYJ.



Następnie klikamy prawym przyciskiem myszy na ZDARZENIE, z listy wybieramy opcję DUPLIKU i przenosimy całość (2 komendy) do następnych 3 torów.



Grę rozpoczyna się w momencie naciśnięcia przez użytkownika klawisza spacji. Wtedy nadajemy komunikat do wszystkich Duszków, że nasza gra rusza. Tory od 2-5 od tego momentu będą ustawione w sobie przypisanej pozycji i rozpoczną płynięcie. Wybieramy komendę bloku ZDARZENIA → KIEDY OTRZYMAM START.

Kolejnym krokiem jest podpięcie komendy bloku RUCH → IDŹ DO x: 0 y: 360. Pytamy dzieci dlaczego takie współrzędne? Wartość współrzędnej y wynosi 360, ponieważ maksymalna wartość współrzędnej osi Y dla Podglądu Duszków to 180, natomiast wysokość całego Podglądu Duszków to 360. Chcemy ustawić Duszka: TOR 2 w prostokącie niewidocznym ponad Podglądem Duszków, czyli musimy zsumować połowę wysokości Podglądu Duszków widocznego dla nas oraz połowę wysokości prostokąta już niewidocznego (współrzędna y jest przypisana środkowi ciężkości kostiumu, stąd połowa wysokości).



	900	
-240	(0,720)	240
	540	
	540	
-240	(0,360)	240
	180	
PODGLĄD	DUSZKÓW	
-240	(0,0)	240
	-180	

Następnie podpinamy komendę bloku RUCH → POKAŻ. Teraz, tak jak w przypadku Duszka: TOR 1 będziemy 2000 razy ustawiać pozycję y Duszka. Wybieramy blok RUCH → USTAW Y NA ... . Wewnątrz okna powyższej komendy umieszczamy dwa WYRAŻENIA ARYTMETYCZNE (komendy z bloku WYRAŻENIA): ze znakiem + oraz \*.

We wnętrzu prawego okna WYRAŻENIA DODAWANIA, umieszczamy ZNAK MNOŻENIA, gdzie działanie mnożenia wykonamy między liczbą 360 (wysokość prostokąta Podglądu Duszków) oraz cyfrą 1. Każdy kolejny tor będzie znajdował się o jedną wysokość powyżej, dlatego mnożyć będziemy przez cyfry od 1-4.

W lewym oknie WYRAŻENIA DODAWANIA umieścimy ZMIENNĄ PŁYNIĘCIE (z bloku DANE).



Całość umieszczamy wewnątrz PĘTLI ITERACYJNEJ, blok KONTROLA → POWTÓRZ 2000 RAZY. Następnie pod PĘTLĄ podpinamy z bloku WYGLĄD → UKRYJ tak, aby przebyta samochodem część toru zniknęła.



UWAGA: W zależności od upodobań instruktora (i czasu, który na to pozwala) dzieci duplikują powyższy fragment skryptu do innych Duszków, bądź samodzielnie tworzą bloki skryptu z pamięci, wspomagając się już napisanym.

Duszek: Tor 3: zmieniamy jego pozycję wejściową na współrzędne x: 0, y: 720 tak, aby znajdował się powyżej już ułożonych torów. Komenda USTAW Y NA PŁYNIĘCIĘ + 360 \* 2 –ZMIENNĄ powiększamy o dwie wysokości Podglądu, ponieważ Duszek: Tor 3 znajduje się powyżej Duszka: Tor 2 i ma do pokonania dłuższą drogę, a więc musi dłużej płynąć.

Duszek: Tor 4: zmieniamy jego pozycję wejściową na współrzędne x: 0, y: 900. Komenda USTAW Y NA PŁYNIĘCIĘ + 360 \* 3. Duszek: Tor 5: zmieniamy jego pozycję wejściową na współrzędne x: 0, y: 1080.

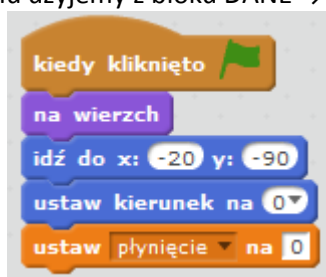
Komenda USTAW Y NA PŁYNIĘCIĘ + 360 \* 4.

### C. Skrypt głównego Duszka

Głównego Duszka czyli samochód ustawiamy standardowo, tak jak pozostałe Duszki, w konkretnym położeniu na Podglądzie Duszków w momencie rozpoczęcia skryptu.

Wybieramy odpowiednie ZDARZENIE z bloku ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy z bloku WYGLĄD → NA WIERZCH, tak aby naszego Duszka inne nie przystaniały.

Kolejnym krokiem jest ustawienie go w odpowiednim położeniu, dlatego wybieramy RUCH → IDŹ DO x: -20 y: -90. Ustawiamy jego kierunek na wartość 0 tak, aby przy rozpoczęciu skryptu skierowany był w górę Podglądu Duszków. W tym miejscu również będziemy po każdej grze zerować naszą ZMIENNĄ PŁYNIĘCIE, w tym celu użyjemy z bloku DANE → USTAW PŁYNIĘCIE NA 0.



Od tego momentu wszystko co programujemy wpływa już bezpośrednio na grę i dlatego rozpoczyna się od ZDARZENIA START.

Wybieramy z bloku ZDARZENIA → KIEDY OTRZYMAM START. Następnie podpinamy PĘTLĘ NIESKOŃCZONĄ z bloku KONTROLA → ZAWSZE. W PĘTLI umieszczamy komendy, które będą wykonywać się bez przerwy w trakcie całej gry. Dlatego do wnętrza PĘTLI wkładamy z bloku WYGLĄD → NA WIERZCH tak, aby żaden Duszek nie zakrył Samochodu w trakcie gry.

Kolejnym krokiem będzie ustawienie NA SZTYWNO pozycji Samochodu na osi Y. Wartość



współrzędnej niezmiennie, przez cały czas trwania gry będzie wynosiła -90 (przykład: będą poruszać się Duszki Torów, a nie Samochód). Do tego celu wykorzystamy blok RUCH → USTAW Y NA -90. Jednocześnie nie poruszając Duszką wzdłuż osi Y, poruszając nią będziemy wzdłuż osi X. Dlatego wybieramy z bloku RUCH → PRZESUŃ O 3 KROKÓW.



Powyżej nadaliśmy Samochodowi możliwość poruszania w lewo i prawo (po osi X), jednak nie udostępniliśmy tego Użytkownikowi. Gracz będzie poruszał Duszką przy pomocy strzałek (bądź klawiszami A i D). Pozwolimy również na to, aby Duszek obrócił się o parę stopni wokół własnej osi, w jedną i drugą stronę tak, aby symulować wchodzenie pojazdu w zakręt.

Wybieramy z bloku ZDARZENIA → KIEDY KLAWISZ STRZAŁKA W PRAWO NACIŚNIĘTY. Następnie podpinamy z bloku RUCH → OBRÓĆ (W PRAWO) O 7 STOPNI. To samo wykonujemy dla skrętu w lewą stronę.

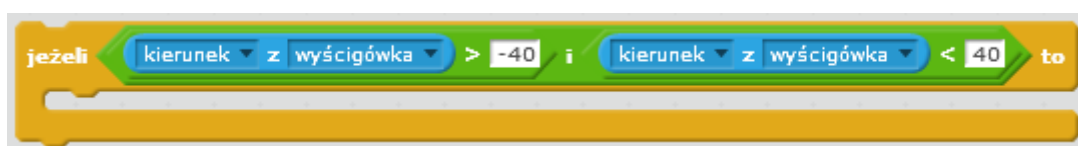


UWAGA: Jeżeli Samochód nie skręca wokół środka ciężkości, należy ten środek wyznaczyć ponownie w zakładce KOSTIUMY → USTAW ŚRODEK KOSTIUMU (prawy górny róg Edytora Obrazu).

W tym miejscu wprawimy w ruch nasz Tor Wyścigowy, poprzez dekrementację (pomniejszanie) wartości ZMIENNEJ PŁYNIĘCIE o 2. Jej wartość (bądź wartość ZMIENNEJ pomnożona dodatkowo przez wysokość, na której znajduje się dany Tor) następnie jest przypisywana Duszkom: Torom jako współrzędna y.

Wybieramy z bloku ZDARZENIA → KIEDY OTRZYMAM START. ZMIENNA PŁYNIĘCIE będzie dekrementowana tylko i wyłącznie w momencie, kiedy Duszek będzie skierowany w górę (KIERUNEK=0 i  $-40 < \text{KIERUNEK} < 40$ ). Do tego celu potrzebna nam będzie INSTRUKCJA WARUNKOWA, sprawdzająca czy Samochód zwrócony jest w dobrym kierunku.

Wybieramy z bloku KONTROLA → JEŻELI .. TO. W oknie INSTRUKCJI umieszczamy z bloku WYRAŻENIA → OPERATOR LOGICZNY ... i ... . W lewym oknie OPERATORA umieścimy z bloku WYRAŻENIA → ZNAK WIĘKSZOŚCI, w prawym natomiast, z bloku WYRAŻENIA → ZNAK MNIEJSZOŚCI. W oknach znaków ... > ... , jak i ... < ... , umieszczamy w lewym oknie z bloku CZUJNIKI → KIERUNEK Z WYŚCIGÓWKA, natomiast w prawym odpowiednio: wartości -40 oraz 40.



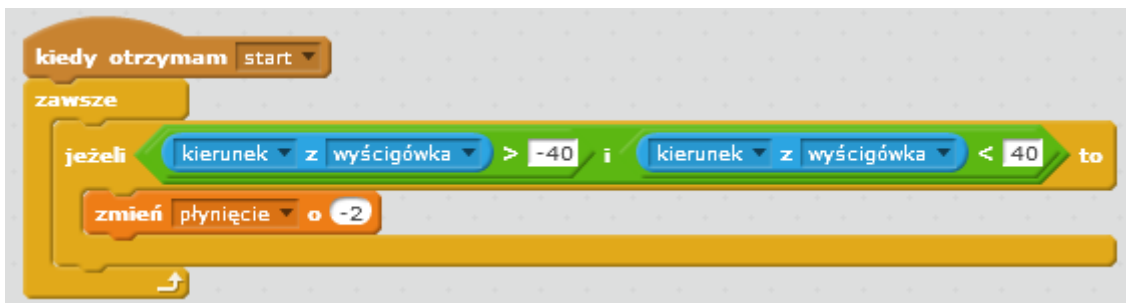
We wnętrzu INSTRUKCJI umieszczamy z bloku DANE → ZMIENŃ PŁYNIĘCIE O -2.

UWAGA: Wedle upodobań Twórcy gry, ZMIENNĄ można pomniejszać (dekrementować) o wartość



większą od 2 i w ten sposób zwiększyć prędkość jazdy Wyścigówki (pośrednio, ponieważ bezpośrednio przyspieszamy ruch Duszków: Torów).

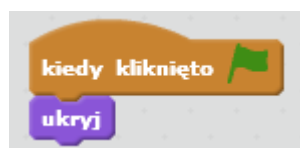
Aby warunek był sprawdzany przez cały czas trwania gry oraz ZMIENNA PŁYNIĘCIE była dekrementowana, całość zawieramy w PĘTLI NIESKOŃCZONEJ z bloku KONTROLA → ZAWSZE.



## D. Auto-przszkoda

Na razie na torze znajduje się wyłącznie nasza Wyścigówka. Żeby nieco utrudnić zadanie zaprogramujemy przeszkodę.

Podczas uruchamiania skryptu Przeszkoda będzie niewidoczna, dlatego wybieramy z bloku ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy z bloku WYGLĄD → UKRYJ.

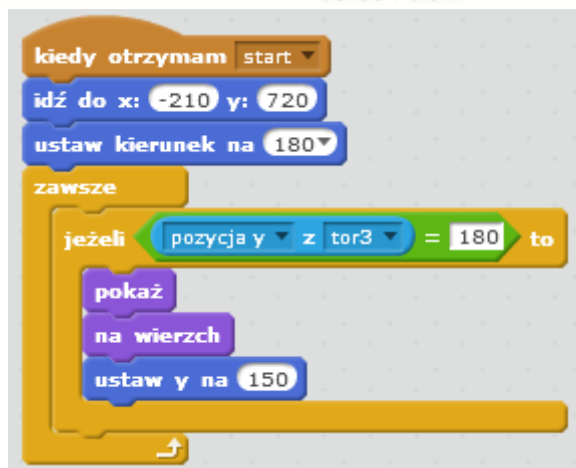


Przy rozpoczęciu skryptu Duszka-Przeszkodę ustawimy w lewym górnym rogu odcinka Toru na którym ma się on pojawić. W przykładzie gry jest to wysokość  $y=720$ , ponieważ jest to ustawienie Duszka:Toru 3.

Wyberamy z bloku ZDARZENIA → KIEDY OTRZYMAM START. Następnie podpinamy z bloku RUCH → IDŹ DO x: -210 y: 720.

Kolejnym krokiem jest ustawienie kierunku samochodu w dół tak, aby uzyskać efekt wymijania Przeszkody przez Gracza. Wybieramy z bloku RUCH → USTAW KIERUNEK NA 180. Następnie podpinamy PĘTLI NIESKOŃCZONĄ z bloku KONTROLA → ZAWSZE. We wnętrzu PĘTLI umieścimy INSTRUKCJĘ WARUNKOWĄ, która sprawdzać będzie czy odpowiednia plansza Toru (Duszek) pojawił się już w oknie Podglądu Duszków. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie INSTRUKCJI umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI ... = ... W lewym oknie ... = ... umieszczamy z bloku CZUJNIKI → POZYCJA Y Z TOR3, w prawym natomiast 180 (jest to maksymalna wartość osi Y Podglądu Duszków, najwyższy punkt).

Jeżeli Duszek: Tor 3 pokaże się w oknie Podglądu Duszków, wtedy nasza Przeszkoda pojawi się również. Dlatego we wnętrzu INSTRUKCJI umieszczamy z bloku WYGLĄD → POKAŻ oraz NA WIERZCH. Następnie ustawiamy Przeszkodę na wysokości  $y=150$  przy pomocy komendy z bloku RUCH → USTAW Y NA 150.



Jeżeli już nasza Przeszkoda jest widoczna oraz znajduje się w odpowiednim miejscu Podglądu, musimy jej jeszcze nadać możliwość poruszania się.

Wybieramy z bloku KONTROLA → ZAWSZE, PĘTLĘ NIESKOŃCZONĄ. W jej wnętrzu umieszczamy blok RUCH → ZMIEŃ Y O -3, tak aby nasza Przeszkoda poruszała się wzdłuż osi Y, z góry na dół.

Następnie sprawimy, że przeszkoda zniknie w momencie pokonanie przez nią całej długości Podglądu.

Wybieramy INSTRUKCJĘ WARUNKOWĄ z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku CZUJNIKI → DOTYKA KRAWĘDZI? TO. Następnie we wnętrzu INSTRUKCJI dopinamy z bloku WYGLĄD → UKRYJ.



Smaczkiem naszej gry będzie możliwość poruszania się Przeszkody nieliniowo. Jazda Duszka w linii prostej nie byłaby wielkim utrudnieniem, będzie nim natomiast zmiana pozycji x Przeszkody w momencie jazdy. Będzie on delikatnie poruszał się od lewej do prawej.

W tym celu wybieramy ZDARZENIA → KIEDY OTRZYMAM START. Następnie całość wykonywać się będzie przez cały czas trwania gry, czyli podpinamy PĘTLĘ NIESKOŃCZONĄ z bloku



KONTROLA → ZAWSZE. We wnętrzu PĘTLI umieścimy INSTRUKCJĘ WARUNKOWĄ z bloku KONTROLA → JEŻELI ... TO. INSTRUKCJA ta będzie sprawdzała warunek, czy nasz odcinek drogi na którym ma pojawić się Przeszkoda pojawił się już w oknie Podglądu.

Dlatego wybieramy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewym oknie znaku ... = ... umieścimy z bloku CZUJNIKI → POZYCJA Y Z TOR3. W prawym natomiast wartość 180 (maks wartości osi Y Podglądu).

Następnie 10 razy powtórzmy losowanie wartości współrzędnej x Przeszkody. W tym celu wybieramy z bloku KONTROLA → POWTÓRZ 10 RAZY. W jej wnętrzu umieszczamy z bloku RUCH → USTAW X NA ... . W okno powyższej komendy wkładamy z bloku WYRAŻENIA → LOSUJ OD -215 DO -170.

UWAGA: W zależności od tego, gdzie na Podglądzie znajdują się droga i jakiej jest szerokości, wartości te należy zmodyfikować.

Następnie podpinamy z bloku KONTROLA → CZEKAJ 1s, pozwalając na to, aby Przeszkoda przejechała pewien odcinek drogi znajdując się w wylosowanej pozycji współrzędnej x.



## E. Czas

Mamy wyścigówkę, tor wyścigowy i przeszkodę ale czy możemy rozpocząć wyścig? Wyścigi nie mają sensu bez pomiaru czasu!

W tym celu na samym początku stworzyliśmy trzy Duszki-cyfry od 5 do 10 kostiumów każdy. Mamy Duszki: jedności sekundy (nazwa Duszka: sekundy), dziesiątne sekundy (sekundy2) oraz minuty (minuty).

W pierwszej kolejności, naszym zadaniem będzie nazwanie odpowiednio kostiumów Duszków. Jest to potrzebne, ponieważ w trakcie gry będziemy odwoływać się do kostiumów poprzez zmienne całkowitoliczbowe. Wybieramy zakładkę KOSTIUMY i tam, w lewym górnym rogu Edytora Obrazu, nadajemy kostiumom nazwy.

UWAGA: Kostiumy nie są numerowane od 0, tak jak wskazywałoby na to logika, ponieważ w takim przypadku Scratch może działać nieprawidłowo. Nazwy nadajemy od cyfry 1 wzwyż.





Będziemy teraz ustawiać Duszki w odpowiednim miejscu Podglądu oraz „nastawiać” zegar. Wybieramy z bloku ZDARZENIA → KIEDY KLIKNIĘTO ZIELONĄ FLAGĘ. Następnie podpinamy z bloku WYGLĄD → ZMIEN KOSTIUM NA 1.

Kolejnym krokiem jest ustawienie Duszków w jednej linii (wspólna wartość współrzędnej x). Wybieramy z bloku RUCH → IDŹ DO x: 180 y: 149 (prawy górny róg Podglądu). Następnie resetujemy czas przy pomocy komendy z bloku DANE → USTAW SEKUNDY NA 1.



W momencie rozpoczęcia gry Duszki będą zmieniać kostiumy na odwołującą się do nich zmienną całkowitoliczbową. Wybieramy blok ZDARZENIA → KIEDY OTRZYMAM START. Następnie podpinamy z bloku WYGLĄD → NA WIERZCH.

Kolejnym krokiem jest zmiana kostiumu przez cały czas trwania gry, dlatego wybieramy PĘTLĘ NIESKOŃCZONĄ z bloku KONTROLA → ZAWSZE. W jej wnętrzu umieszczamy komendę z bloku WYGLĄD → ZMIEN KOSTIUM NA ... . W okno powyższej komendy wkładamy ZMIENNĄ z bloku DANE → SEKUNDY. Teraz w zależności od wartości ZMIENNEJ SEKUNDY taki kostium ubierze nasz Duszek.



Wykonujemy punkty od 1-10 dla dwóch pozostałych Duszków-Czasu zmieniając wartości współrzędnej x (współrzędna y pozostaje bez zmian) oraz w zależności od Duszka zmieniając nazwy ZMIENNYCH na SEKUNDY2 oraz MINUTY.

UWAGA: Dla kolejnych Duszków dzieci tworzą bloki komend z pamięci wzorując się na już stworzonym, bądź duplikują bloki i przenoszą je.

## E1. Sekundy

W tym miejscu zajmiemy się uruchomieniem naszego zegara czyli odliczaniem czasu co do sekundy. Wybieramy z bloku ZDARZENIA → KIEDY OTRZYMAJAM START.

Następnie podpinamy z bloku KONTROLA → ZAWSZE. W tej PĘTLI odliczany będzie czas przez całe trwanie gry, będziemy inkrementować ZMIENNĄ SEKUNDY. We wnętrzu PĘTLI umieszczamy z bloku DANE → ZMIENŃ SEKUNDY O 1. Następnie podpinamy z bloku KONTROLA → CZEKAJ 0,3s. Komenda ta, pozwoli nam na naliczanie wirtualnego czasu w świecie gry.

Kolejnym krokiem jest umieszczenie INSTRUKCJI WARUNKOWEJ, która pozwoli nam po osiągnięciu 9 sekundy, na powrót do 0. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI. W lewym oknie ZNAKU ... = ... umieszczamy ZMIENNĄ SEKUNDY (z bloku DANE), w prawym oknie natomiast wpisujemy wartość 11. Do wnętrza INSTRUKCJI wkładamy z bloku DANE → ZMIENŃ SEKUNDY2 O 1. Poniżej podpinamy z bloku DANE → USTAW SEKUNDY NA 1.



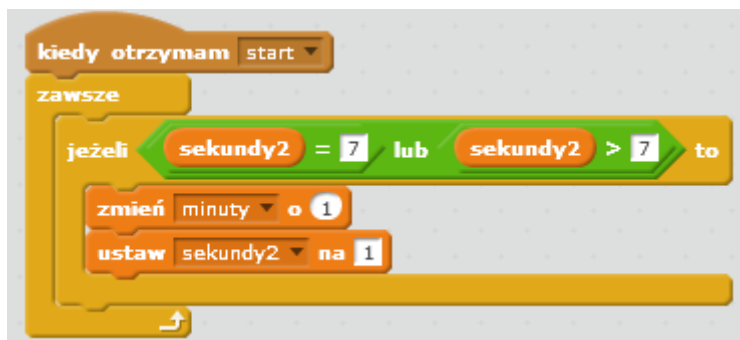
## E2. Sekundy2

Wybieramy z bloku ZDARZENIA → KIEDY OTRZYMAJAM START. Następnie podpinamy z bloku KONTROLA → ZAWSZE.

Kolejnym krokiem jest umieszczenie INSTRUKCJI WARUNKOWEJ, która pozwoli nam po osiągnięciu 5 sekundy, na powrót do 0, tak aby ZMIENNA MINUTY została naliczona. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku WYRAŻENIA → OPERATOR



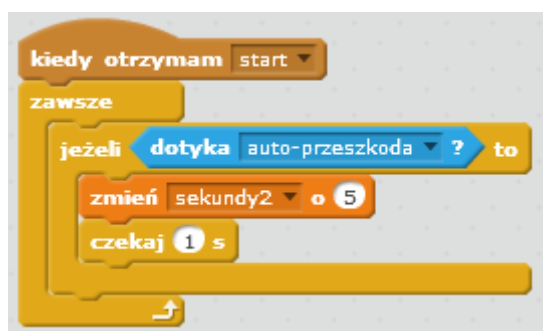
LOGICZNY LUB. Do lewego okna komendy wkładamy z bloku WYRAŻENIA → ZNAK RÓWNOŚCI, w prawe natomiast z bloku WYRAŻENIA → ZNAK WIĘKSZOŚCI. W tym miejscu będziemy sprawdzać czy wartość 6 została przekroczona, bądź ZMIENNA posiada taką wartość. W lewym oknie .. = ... umieszczamy z bloku DANE → ZMIENNĄ SEKUNDY2, a w prawym wartość 7. W lewym oknie ... > ... umieszczamy z bloku DANE → ZMIENNĄ SEKUNDY2, a w prawym wartość 7. Do wnętrza INSTRUKCJI wkładamy z bloku DANE → ZMIENIĆ MINUTY O 1. Poniżej podpinamy z bloku DANE → USTAW SEKUNDY2 NA 1.



## F. Skrypt głównego Duszka

Powróćmy jeszcze na chwilę do skryptu Głównego Duszka. Przeszkoda działa, czas został już zaprogramowany, teraz tylko potrzebujemy poniesienia konsekwencji przez Gracza w momencie zetknięcia Głównego Duszka z Przeszkodą.

Wybieramy z bloku ZDARZENIA → KIEDY ORZYMAM START. Następnie podpinamy z bloku KONTROLA → ZAWSZE. Kolejnym krokiem jest umieszczenie INSTRUKCJI WARUNKOWEJ, która będzie doliczała karny czas, za nieominięcie Przeszkody. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku CZUJNIKI → DOTYKA AUTO-PRZESZKODA? TO. Do wnętrza INSTRUKCJI wkładamy z bloku DANE → ZMIENIĆ SEKUNDY2 O 5. Następnie podpinamy z bloku KONTROLA → CZEKAJ 0,3s, która to komenda nie pozwoli na doliczanie zbyt dużej ilości karnych sekund. Doliczenie ma być jednorazowe. Brak tej komendy będzie powodował ciągłe naliczanie karnych sekund w trakcie zetknięcia.



Ostatnim elementem gry będzie jej zakończenie w momencie dojechania przez Gracza do mety. Pod powyższą INSTRUKCJĄ dopinamy kolejną. Wybieramy z bloku KONTROLA → JEŻELI ... TO. W oknie komendy umieszczamy z bloku CZUJNIKI → DOTYKA KOLORU CZERWONEGO? TO. We wnętrzu INSTRUKCJI umieszczamy z bloku KONTROLA → ZATRZYMAJ WSZYSTKO.



### Rozszerzenie:

1. Grę możemy ubogacić liczniejnymi Autami-Przeszkodami.
2. Możemy również dodać DŹWIĘKI, melodie napisane przez dzieci przy pomocy instrumentów (blok DŹWIĘKI), bądź zaimportowane z biblioteki.



## Zajęcia 15: Obrona zamku

---

### Zadanie:

Wykonanie gry stwarzającej wrażenie rzutu ukośnego.

Gra polega na wystrzeliwaniu strzał w celu obrony zamku.

### Cel:

Przypomnienie idei programowania równoległego, wykorzystanie systemu nadawania komunikatów do komunikacji pomiędzy duszkami.

### Wprowadzenie:

Prezentujemy dzieciom grę, początkowo nie zdradzając jej mechanizmu. Zadajemy pytania o zastosowane bloki komend oraz zabiegi programistyczne.

### Wykonanie cz.1:

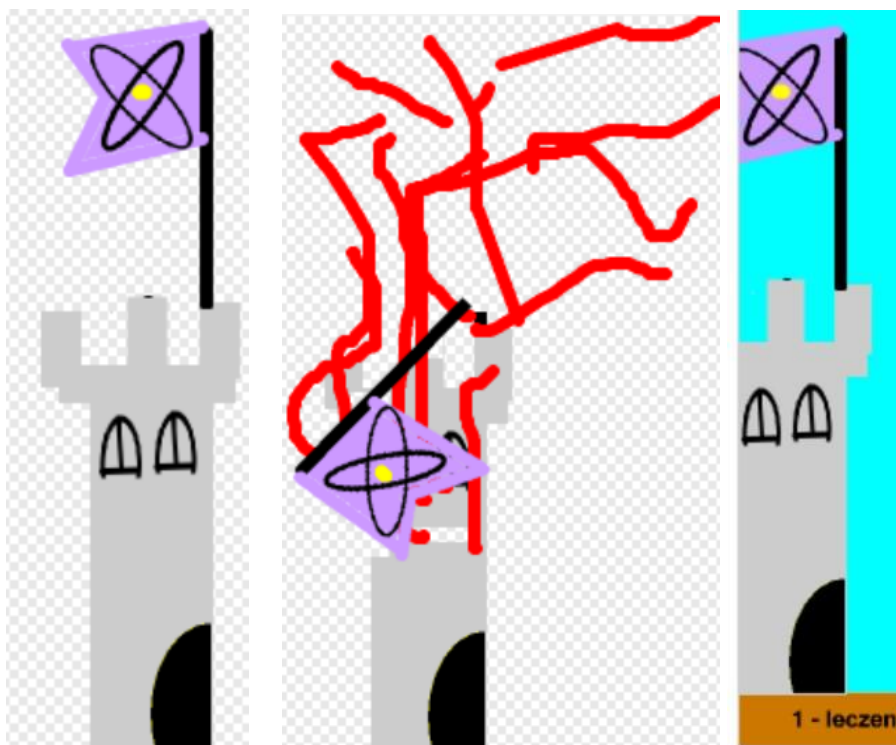
#### A. Stworzenie tła

Tło składa się z dwóch części, niewielkiej dolnej imitującej podłogę (o kolorze brązowym/zielonym) na którym umieścimy ewentualny opis klawiszy funkcyjnych oraz górnej części odwzorowującej niebo.



#### B. Pierwszy duszek - zamek

Dzieci wykonują tego duszka samodzielnie, powinien on być w miarę wąski, aby zajmował jak najmniej miejsca z lewej strony planszy oraz wysoki na całą planszę (można dodać do naszego zamku maszt ze sztandarem wymyślonym przez uczniów). Przypominamy uczestnikom, że po środku naszego zamku będzie umieszczona kusza (tudzież inny przedmiot, który będzie wystrzeliwał pociski). Na tym etapie tworzenia gry możemy wykonać od razu drugo kostium przedstawiający zniszczony zamek w płomieniach



Ustawiamy zamek w odpowiednim miejscu (jak najbardziej z lewej strony, ale tak żeby zajmował całą wysokość planszy, można go rozciągnąć). Na tym etapie ustawiamy wybór kostiumu oraz współrzędne położenia zamku, aby zapobiec jego przypadkowemu przesunięciu.



### C. Łuk/kusza

Uczestnicy wykonują przyrząd który będzie wystrzeliwał pociski, najlepiej łuk/kusza. Będzie się on obracał wokół środka ciężkości, najlepiej ustawić go pośrodku cięciwy. Łuk umieszczamy na szczycie wieży. Po ZIELONEJ FLADZE ustawiamy współrzędne które odpowiadają temu punktowi (przeciągamy bloczek „idź do” współrzędne w których znajduje się środek ciężkości są ustawione automatycznie).

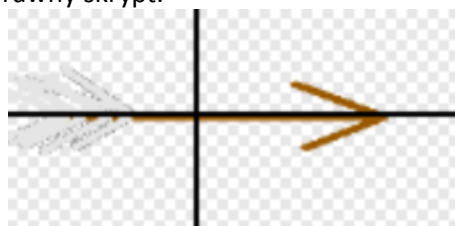
Uczniowie powinni wykonać samodzielnie skrypt przy pomocy którego będzie się on obracał wokół środka ciężkości (analogicznie jak w grze Asteroids).





## D. Strzała/pocisk

Uczniowie wykonują strzałę która będzie wystrzeliwana w chwili naciśnięcia spacji (pytamy się uczniów w jaki sposób rozwiązać tę kwestię – przypomnienie mechanizmu wykorzystanego w grze asteroids). Uczniowie samodzielnie próbują wykonać skrypt strzały. Po 2-5 min wspólnie z uczestnikami wykonujemy poprawny skrypt.



### ZIELONA FLAGA

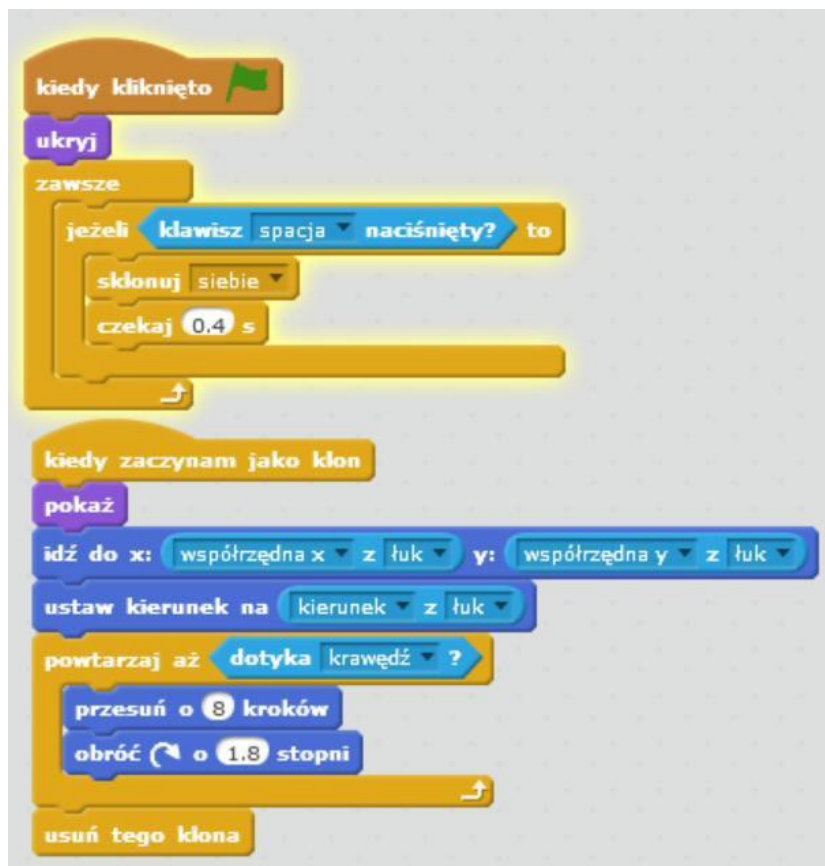
- po naciśnięciu zielonej flagi strzała zostaje ukryta,
- możemy zastosować zarówno polecenie „jeżeli...to” lub „czekaj aż” klawisz spacja naciśnięty (w przykładzie zostało użyte polecenie „jeżeli...to”),
- duszek ulega sklonowaniu – polecenie „sklonuj siebie”,
- wstawienie bloczka „czekaj 0.4s” zapobiega „zawieszeniu” programu, po przez zmianę tej wartości wpływamy na częstotliwość wystrzeliwanych strzał.

### KIEDY ZACZYNAM JAKO KLON

- klon pokazuje się i idzie do współrzędnych, w których położony jest łuk (można przepisać współrzędne położenia łuku, ale warto zapoznać uczniów z bloczkami z zakładki czujniki, które z czytują wartość danej współrzędnej z innego duszka),
- ustawiamy kierunek na (odczytujemy aktualną wartość kierunku w jakim ustawiony jest łuk – bloczek użyty w poprzednim poleceniu tylko wybór innego parametru),



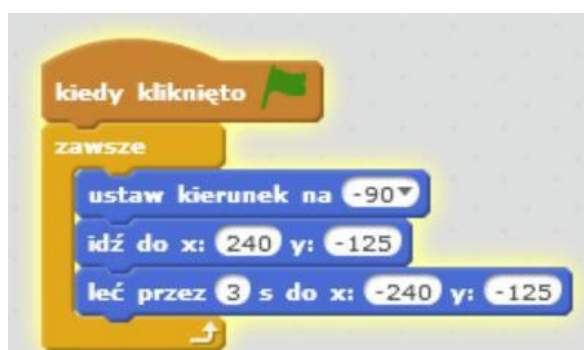
- strzała poleci tak długo, aż nie dotknie krawędzi,
- bloczek „przesuń o x kroków” wpływa na prędkość strzały, a bloczek „obróć o x stopni” sprawia wrażenie jak by strzała była przyciągana przez ziemię.
- gdy strzała doleci do krawędzi dany klon jest usuwany.



Na tym etapie nie dodajemy jeszcze polecenia usuwania klonu w przypadku trafienia przeciwnika.

## E. Pierwszy przeciwnik – Rycerz

Uczniowie starają się wykonać wstępny skrypt który powoduje, że rycerz będzie poruszał się z prawej strony do lewej. Jeżeli im się to uda można przystąpić do dalszej części zajęć, jeżeli nie instruktor wykonuje wspólnie z nimi ten skrypt – analogiczny do przesuujących się „kaktusów” w grze brak Internetu.







Uwaga: Nie chcemy aby rycerze szli pojedynczo. Należy ich klonować, dzięki czemu w łatwy sposób będziemy mogli modyfikować częstotliwość ich pojawiania się.

Na tym etapie tworzenia gry można wprowadzić zmienne odpowiadające za życie zamku (zmienna globalna HP) oraz punkty zdobywane za pokonanie przeciwnika (w tym przypadki złoto - Gold).



#### ZIELONA FLAGA

- po naciśnięciu zielonej flagi rycerz zostaje ukryta,
- duszek ulega sklonowaniu – polecenie „sklonuj siebie”,
- wstawienie bloczka „czekaj x s” zapobiega „zawieszeniu” programu, po przez zmianę tej wartości wpływamy na częstotliwość pojawiania się rycerzy – zastosowanie polecenia losuj sprawi, że będą oni pojawiali się w losowych odstępach czasu.

#### KIEDY ZACZYNAM JAKO KLON I

- klon pokazuje się,
- ustawiamy kierunek w lewą stronę,
- ustawia się z prawej strony planszy,
- i przesuwa się w lewą stronę. Po osiągnięciu współrzędnej zamku odejmuje mu punkty życia, po czym jest usuwany (można zastosować bloczek przesuń o ilość kroków w pętli powtarzaj, aż dotyka zamek,
- zastosowanie bloczku „losuj od... do...” Sprawia, że rycerze idą z różną prędkością,

#### KIEDY ZACZYNAM JAKO KLON II

- klon czeka, aż zostanie ugodzony strzałą,
- jeżeli zostanie trafiony zyskujemy złoto,
- wstawienie bloczka „czekaj 0.05 s” daje to czas na to aby dana strzała zdążyła zareagować na trafienie i mogła zostać usunięta,
- dany klon rycerza zostaje usunięty.

## F. Dokończenie skryptu Zamku



- w tym miejscu można dodać przypisanie wartości zmiennych Gold i Hp.
- zamek czeka, aż jego Hp spadnie poniżej 1,
- ulega wtedy zniszczeniu, zmienia kostium,
- po zniszczeniu cały program się zatrzymuje,
- licznik życia można umieścić nad wejściem do zamku.



## G. Stworzenie dwóch pozostałych przeciwników

### Czarownica

Uczniowie powinni wykonać skrypt tego duszka oraz kolejnego samodzielnie – skrypt identyczny jak w przypadku rycerza, zmianie ulega wartość położenia „y:” na 0 (będzie się ona poruszać nad rycerzem) oraz zmniejszeniu wartości X: na „-180” („wleci” ona wcześniej do zamku odejmując mu życie, wartość ta zależy od wielkości zamku). Można zastosować również wartości losowe czasu w bločku „leć przez ... do x: y:”,



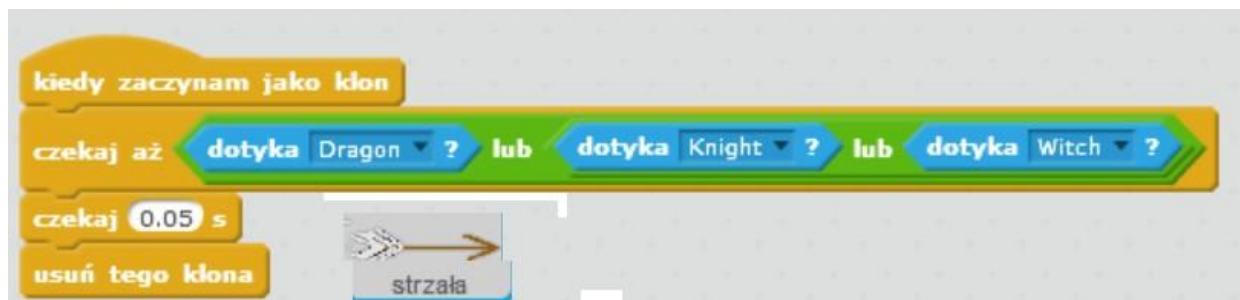
### Smok

Skrypt podobny do skryptu czarownicy, tylko że po zderzeniu z zamkiem smok zaczyna ziać ogniem, zastosowano opóźnienie w pojawieniu się pierwszego smoka – wybór poszczególnych wartości pozostawiamy uczniom.



## H. Dokończenie skryptu Strzały

Strzała po dotknięciu jednego z trzech przeciwników znika (żeby nie było za łatwo), bloczek „czekaj 0.05s” daje czas poszczególnym duszkom na reakcję, że zostały trafione.



## Gra „Obrona Zamku cz.2

### A. Modyfikacja tła

Na dolnym pasku umieszczamy opis poszczególnych rzeczy na które możemy wydać zgromadzone przez nas złoto.

1 - leczenie (+50hp -100G) 2 - burza (100G) 3 - grad strzał (200G)

Wszystkie umiejętności specjalne umieszczamy w skrypcie zamku.

### B. Leczenie

Najprostsza z umiejętności: kiedy zgromadzimy 100G będziemy mogli uleczyć 50HP. Nie został w tym przypadku zastosowany ogranicznik (możemy mieć więcej niż 100HP).



### Skrypt w Zamku :

- po naciśnięciu zielonej flagi program czeka, aż naciśniemy klawisz 1 ale zostaniemy uleczeni wtedy gdy mamy zgromadzone więcej niż 99G, czyli w naszym wypadku 100G,
- jeżeli oba warunki są spełnione tracimy 100G i zyskujemy 50HP (zmiana odpowiednich wielkości).



### C. Burza

Wywołujemy morderczą burzę z piorunami, która dotkliwie rani naszych wrogów. Trwa przez określony czas.

### Skrypt w Zamku:

- podobny do poprzedniego skryptu dla Leczenia, tylko w tym wypadku wciskamy przycisk 2, po spełnieniu obu warunków tracimy 100G oraz nadawany jest komunikat „burza” za chwilę rozpocznie się nawałnica.



Wgrywamy nowego duszka z biblioteki (Lightning), zmniejszamy go do ok 50% początkowej wielkości.

### Skrypt błyskawicy:

PO NACIŚNIĘCIU ZIELONEJ FLAGI

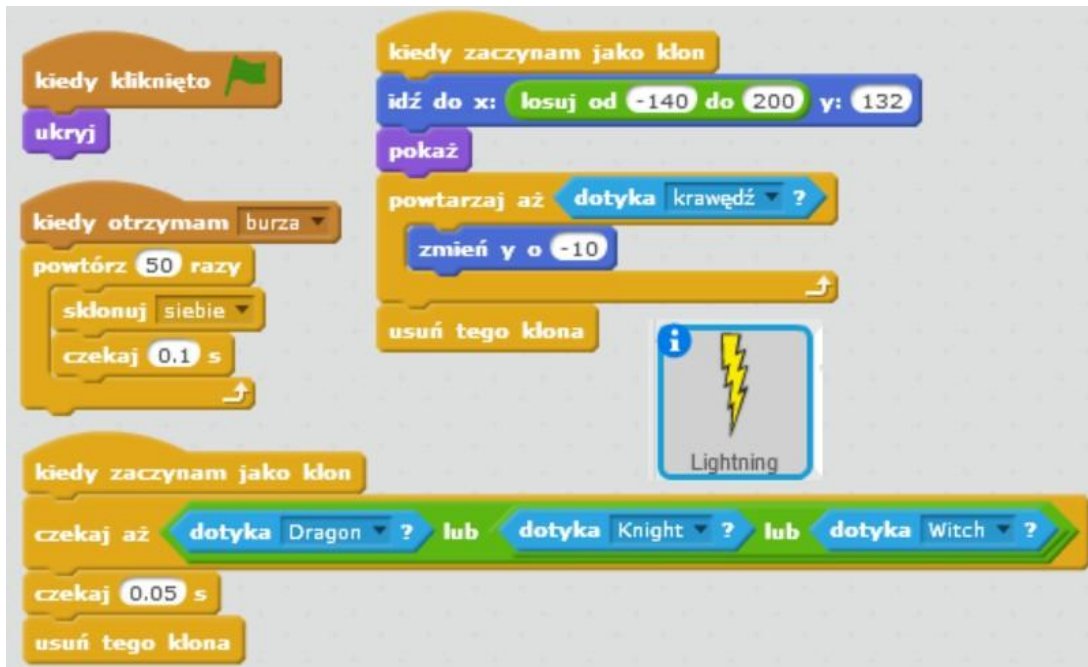
Duszek pozostaje ukryty, KIEDY OTRYMUJE KOMUNIKAT burza klonuje się 50 razy z odstępami 0.1 sekundowymi. Burza trwa 5s. Jej trwanie można modyfikować po przez zwiększenie ilości powtórzeń, wydłużenie czasu pomiędzy poszczególnymi sklonowaniami.



#### KIEDY ZACZYNAM JAKO KLON

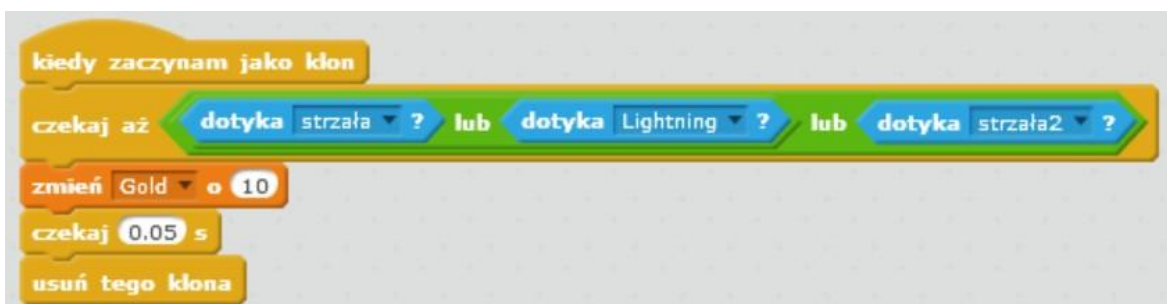
- przenosi się do losowych współrzędnych u góry planszy (y dobieramy tak, aby nam od razu nie zniknął – nie dotykał krawędzi),
- jest już ustawiony w dół więc nie ustawiamy kierunku, przesuwamy w dół tak długo, aż nie dotknie krawędzi, zmieniamy wartość współrzędnej y.

W celu zapobiegnięcia przelatywaniu przez wrogów ustawiamy, że jeżeli dotyka przeciwnika to po 0.05s jest usuwany – można skopiować ze skryptu strzały.



#### Modyfikacja skryptu przeciwników

We wszystkich trzech duszkach przeciwników w miejscu, w którym dostajemy za nich złoto i są oni usuwani dodajemy przy pomocy spójnika logicznego „lub” warunek, że ma to się również stać po zetknięciu z Lightning oraz strzała2 (o strzale2 za chwilę – nie musimy dodawać tego warunku jeżeli nie będziemy dodawać umiejętności specjalnej gradu strzał).





## D. Grad strzał

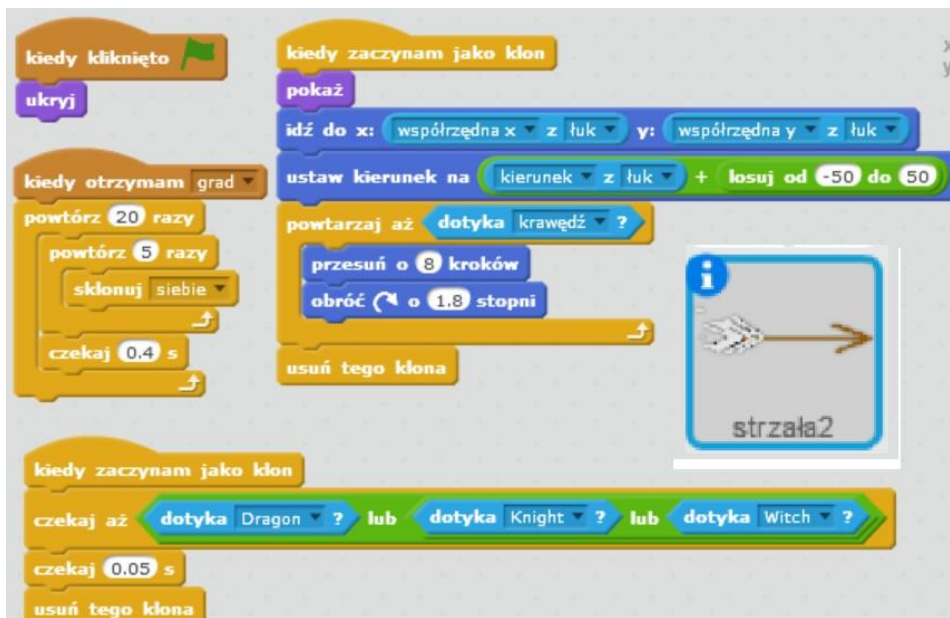
### Skrypt w Zamku:

- podobny do poprzedniego skryptu burzy, tylko w tym wypadku wciskamy przycisk 3 i potrzebujemy większej ilości złota. Po spełnieniu obu warunków tracimy 200G oraz nadawany jest komunikat „grad” za chwilę nasz łucznik zacznie strzelać z dziką furją.



W celu ominięcia pisania całego skryptu strzały kopiujemy naszego duszka „strzała2”, będziemy modyfikować jej skrypt. Uzyskaliśmy duszka strzała2.

### Skrypt strzała2:



Jak widzimy dolna część skryptu pozostała bez zmian (KIEDY ZACZYNAM JAKO KLON). PO NACIŚNIĘCIU ZIELONEJ FLAGI duszek pozostaje ukryty. KIEDY OTRYMUJE KOMUNIKAT grad klonuje się 20 razy z odstępami 0.4 sekundowymi (grad trwa 8s długość jego trwania można modyfikować po przez zwiększenie ilości powtórzeń, wydłużenie czasu pomiędzy poszczególnymi sklonowaniami), za każdą „salwą” powstaje 5 klonów jednocześnie które lecą w losowych kierunkach.

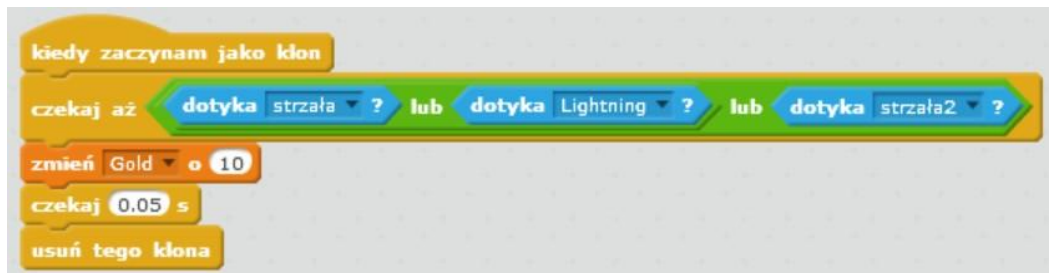


KIEDY ZACZYNAM JAKO KLON (modyfikacja w odniesieniu do zwykłej strzały):

- jedyny element jaki musimy dodać to dwa bloczki z zielonej zakładki wyrażenia „dodawanie” i „losuj od... do...” dzięki temu każda strzała z danej salwy poleci w trochę innym kierunku w odniesieniu od aktualnego kierunku ustawienia łuku.

## Modyfikacja skryptu przeciwników (jeżeli nie zrobiliśmy tego wcześniej)

We wszystkich trzech duszkach przeciwników w miejscu w którym dostajemy za nich złoto i są oni usuwani dodajemy przy pomocy spójnika logicznego „lub” warunek, że ma to się również stać po zetknięciu z Lightning i/lub strzała2 (w zależności czy wprowadziliśmy wcześniejszą modyfikację - burza).



## E. Falanga wrogów

Po wprowadzeniu umiejętności specjalnych gra staje się zbyt łatwa. Dlatego w każdym z trzech duszków dodajemy bloczek, który będzie powodował, że po pewnym losowym czasie dany przeciwnik zacznie bardzo szybko się klonować i podążać w naszym kierunku.



## F. Pomysły na utrudnienie gry do samodzielnego wprowadzenia przez uczestników zajęć:

- dodanie przeciwników dysponujących więcej niż jednym życiem,
- dodanie własnej umiejętności specjalnej,
- wprowadzenie jakiegokolwiek warunku na wygraną gry,
- wprowadzenie zmiennych odpowiedzialnych za prędkość/szybkość klonowania przeciwników.