



# Skrypt ze scenariuszami zajęć dla Nauczyciela

## Programowania Scratch 2.0



## Spis treści

Temat 1. „Robaczek” .....	3
Temat 2. „Animacja” .....	9
Temat 3. „Abecadło” .....	13
Temat 4. „W bibliotece” .....	18
Temat 5. „Gwiezdna potyczka” .....	22
Temat 6. „Gra przygodowa – małpa w dżungli” .....	29
Temat 7. „Bloons Defense” .....	35
Temat 8. „Lot balonem” .....	46
Temat 9. „Gra platformowa” .....	51



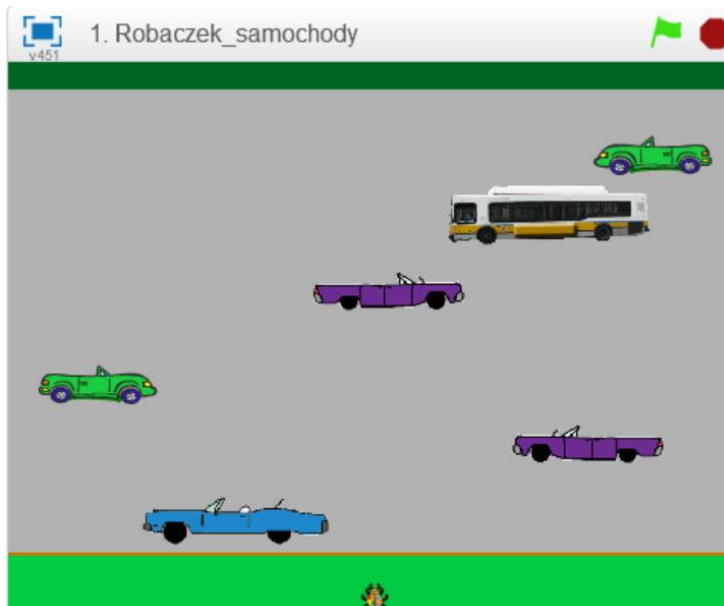
# Temat 1. „Robaczek”

## Proponowany czas realizacji

4x45min

## Krótki opis projektu

Na pierwszych zajęciach uczniowie wykonają swoją pierwszą grę. Zapoznają się ze środowiskiem programistycznym Scratch. Poznają jego podstawowe funkcje. Celem gry jest przeprowadzenie robaczka przez ruchliwą drogę. Na początku zajęć uczniowie wykonają tło gry przy pomocy dostępnego edytora obrazu. Następnie wybiorą duszka z biblioteki oraz napiszą dla niego program. Wykonają, także samochodu, poruszające się pomiędzy krawędziami planszy. W przypadku zderzenia robaczka z samochodem będzie on wracał na start. Po przejściu na drugą stronę drogi, będzie on bezpieczny.



## Przebieg lekcji

1. Przedstawienie uczniom gry „Robaczek” w celu zachęcenia do uczestnictwa w zajęciach
2. Poznanie podstawowych elementów programu (zapoznanie z poszczególnymi zakładkami)
3. Samodzielne wykonanie tła przy pomocy dostępnego edytora obrazu
4. Wybór duszka z biblioteki
5. Napisanie programu umożliwiającego sterowanie duszkiem
6. Przypomnienie definicji układu współrzędnych
7. Narysowanie pierwszej przeszkody oraz wprawienie jej w ruch
8. Dodanie kolejnych przeszkód
9. Dodanie skryptu odpowiedzialnego za powrót robaczka na start
10. Wprowadzenie definicji zmiennej
11. Dodanie zmiennej odpowiedzialnej za ilość żyć
12. Zamiana czarnych bloków na samochody
13. Zapoznanie z blokami z zakładki Wyrażenia, w szczególności z blokiem „... lub ...”
14. Modyfikacja programu robaczka
15. Podsumowanie przebiegu zajęć



## Cele lekcji

Uczeń wie:

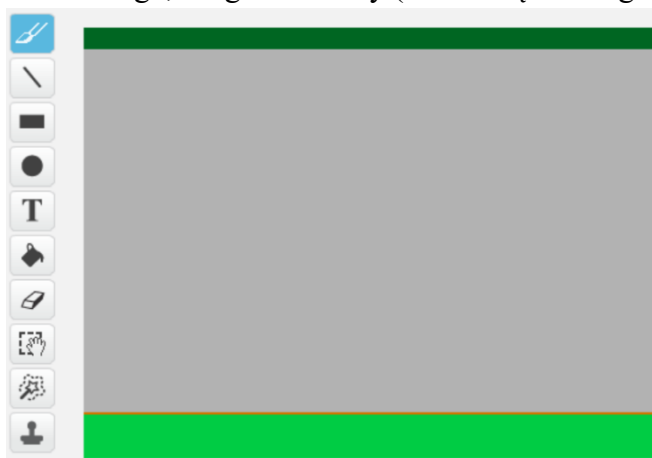
- czym jest pętla i jak można ją wykorzystać w programowaniu,
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

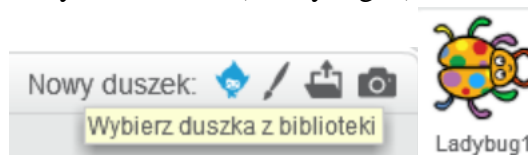
- korzystać z różnych rodzajów pętli,
- za pomocą ciągu poleceń sterować obiektem na ekranie,
- wykonać prostą grę w Scratchu,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej.

## Instrukcja wykonania gry dla nauczyciela

- I. Tło do naszej gry wykonujemy przy pomocy dostępnego edytora obrazu. Przy pomocy figur geometrycznych rysujemy planszę gry składającą się z trzech pól: startowego, drogi oraz mety (każda część innego koloru).



- II. Pierwszego duszka wgrujemy z biblioteki („Ladybug1”).



- III. Klikamy na ikonkę Ladybug1 i przechodzimy do zakładki Skrypty. Cały program będzie się składał z połączonych ze sobą bloczków. Wszystkie bloczki zostały podzielone na osiem kategorii. Każdej odpowiada inny kolor. Umożliwia to szybkie odnalezienie właściwego bloczka.

Na początku ustawimy sterowanie naszego duszka przy pomocy strzałek.

Przechodzimy do brązowej zakładki **Zdarzenia** i wybieramy bloczek „kiedy klawisz...naciśnięty”, z listy rozwijanej wybieramy strzałka w górę. Po każdym wciśnięciu strzałki w górę rozpocznie się wykonywanie tej części skryptu. Do bloczka „kiedy klawisz ...

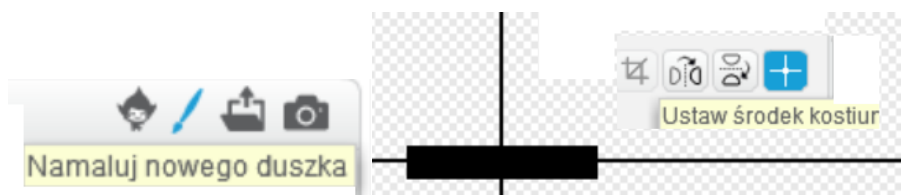
naciśnięty” podczepiamy trzy bloczki odpowiedzialne za ruch (niebieska zakładka Ruch). Na początku ustawiamy kierunek ,w którą ma się przesunąć nasz obiekt – bloczek „ustaw kierunek na 0”. Następnie przesuwamy naszego duszka, o określoną ilość kroków - bloczek „przesuń o 5 kroków”. Im większą wartość wpisujemy tym szybciej będziemy się poruszać po planszy. Ostatni bloczek uniemożliwia opuszczenie planszy „jeżeli na brzegu, odbij się”.



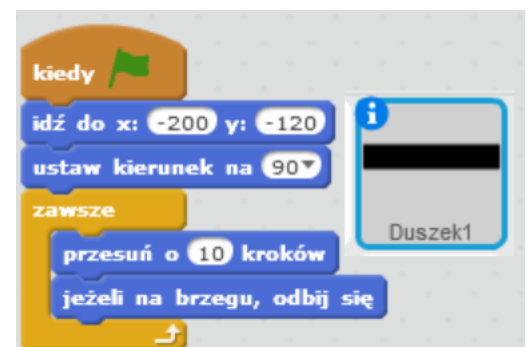
W analogiczny sposób wykonujemy programy dla pozostałych trzech kierunków. Klikając prawym klawiszem myszy na daną część skryptu możemy zduplikować dany fragment wcześniej napisanego programu. Z list rozwijanych wybieramy odpowiednie klawisze oraz kierunki ustawienia naszego duszka.



- IV. Następnego duszka rysujemy samodzielnie przy pomocy edytora obrazu – czarny prostokąt. Środek kostiumu powinien być ustawiony po środku duszka.



- V. Będzie to nasz samochód. Będzie się on poruszał pomiędzy krawędziami planszy. Wykorzystamy do tego pętlę zawsze, znającą się w żółtej zakładce **Kontrola**. Po naciśnięciu Zielonej Flagi zostaje on umieszczony na drodze. Jego kierunek zostaje ustawiony w prawo (90). Zawsze będzie przesuwany się o 10 kroków. Jeżeli będzie znajdował się na brzegu odbije się od niego i będzie poruszał się w drugim kierunku.



- VI. Pierwszy samochód kopiujemy 5 razy. Skrypty poszczególnych duszków będą się różnić pomiędzy sobą współzrędnymi położenia początkowego oraz ustawionym kierunkiem. Współzrędną y zwiększamy o 50 dla każdego kolejnego duszka. Współzrędną x może być dowolna (w zakresie przedziału od -240 do 240), tak samo jak kierunek (-90 lub 90). Poszczególne duszki mogą się różnić pomiędzy sobą również prędkością (wartość, o którą są przesuwane). Przykładowe skrypty poszczególnych duszków znajdują się poniżej.



- VII. Po naciśnięciu Zielonej Flagi robaczek zostaje umieszczony na samym dole planszy (współzrędne startowe).

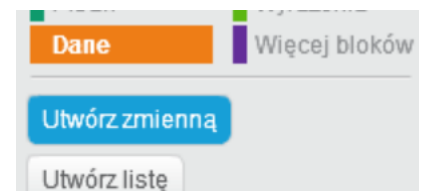
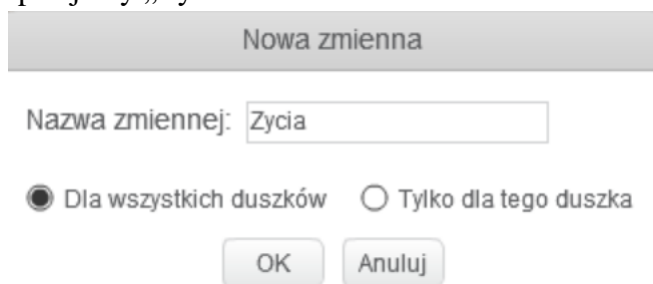
Robaczek ma reagować na zderzenie z samochodami. Użyjemy do tego instrukcji warunkowej „jeżeli ... to”. Duszek będzie reagował na kolor czarny – bloczek „czy dotyka koloru ... ?” z niebieskiej zakładki czujniki. Drugi warunek „jeżeli ... to” sprawdza czy dotarliśmy do końca planszy.

Jeżeli dotykamy koloru czarnego wracamy do położenia startowego. Jeżeli natomiast dotykamy koloru znajdującego się na końcu planszy robaczek informuje nas, że jesteśmy bezpieczni.



Jest to wersja podstawowa gry, którą możemy poddać dalszym modyfikacją.

- VIII. W zakładce Dane tworzymy zmienną, odpowiedzialną za określanie liczby naszych żyć. Klikamy „Utwórz zmienną” i w oknie wpisujemy „Życia”.

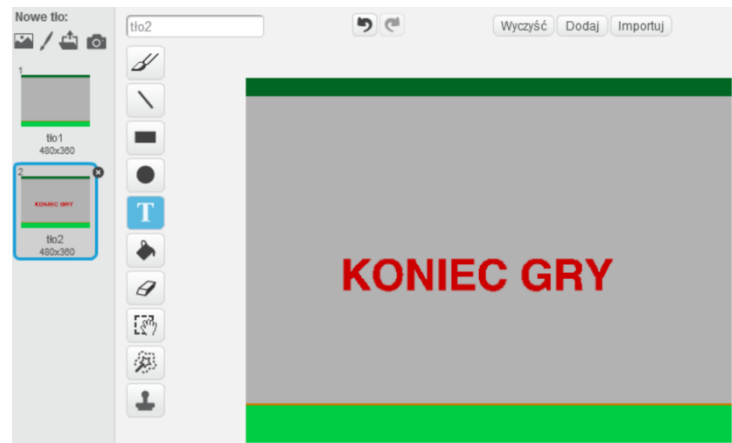




- IX. Dokonujemy modyfikacji skryptu robaczka, tak aby po dotknięciu samochodu zmienna „Życia” była zmieniana o -1. Po rozpoczęciu gry zmiennej „Życia” zostaje przypisana wartość 10.



- X. Duplikujemy tło i przy pomocy edytora obrazu dodajemy napis „Koniec Gry” do tło2.



- XI. Dodajemy skrypt powodujący zmianę tła oraz zniknięcie duszka. Po naciśnięciu Zielonej Flagi robaczek musi się pokazać. Tło zostaje zmieniona na „tło1”. Program będzie oczekiwał, aż punkty spadną poniżej 1 (łączymy ze sobą bloczki „czekaj aż ...” z zakładki Kontrola, „Życia” z zakładki Dane oraz „...<...” z zakładki Wyrażenia).

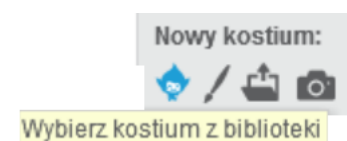
Po spełnieniu warunku program będzie wykonywany dalej. Tło zmieni się na „tło2”. Duszek zostanie ukryty, po czym cała gra ulega zatrzymaniu (bloczek „zatrzymaj wszystko” z zakładki Kontrola).

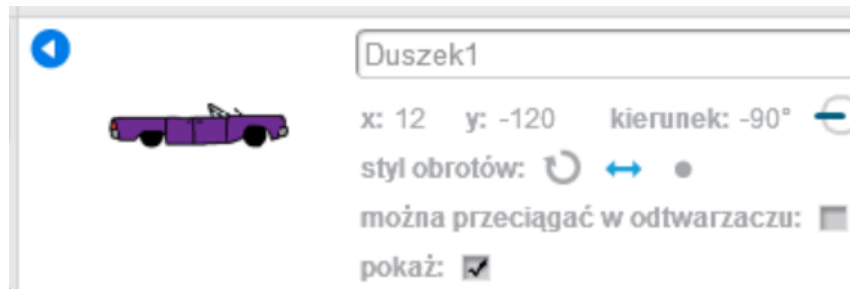


- XII. Aktualnie nasze samochody nie wyglądają zbyt ciekawie. Możemy dodać im nowe kostiumy z biblioteki. Kostiumy pojazdów odnajdziemy w kategorii „Transport”.

Pojazdy poruszając się w lewą stronę obracają się do góry nogami.

Klikając na ikonę „i” na duszku możemy dokonać zmiany stylu obrotu na lewo-prawo. To samo robimy dla pozostałych duszków.

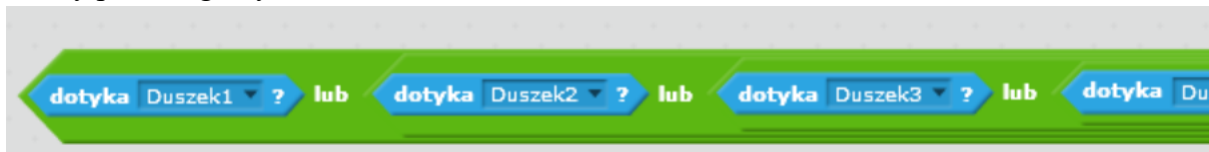




- XIII. Robak nie reaguje od razu na każdy pojazd. Ustawmy, aby reagował on na zderzenie z duszkiem, a nie na konkretny kolor. W tym celu będziemy potrzebować pięć bloczków „...lub...” z zakładki Wyrażenia. Musimy je wszystkie połączyć ze sobą.



Będziemy potrzebować także sześć bloczków „czy dotyka ...?” z zakładki czujniki. Wstawiamy je do połączonych ze sobą bloczków „...lub...”. Z listy rozwijanej wybieramy nazwy poszczególnych duszków.



Całość umieszczamy zamiast bloczka „czy dotyka koloru czarnego?”



### Zadania dodatkowe do samodzielnej realizacji

Dodanie kolejnych pojazdów

Wprowadzenie losowości w ruchu pojazdów przez dodanie bloczka „losuj od ... do ...” z zakładki Wyrażenia.





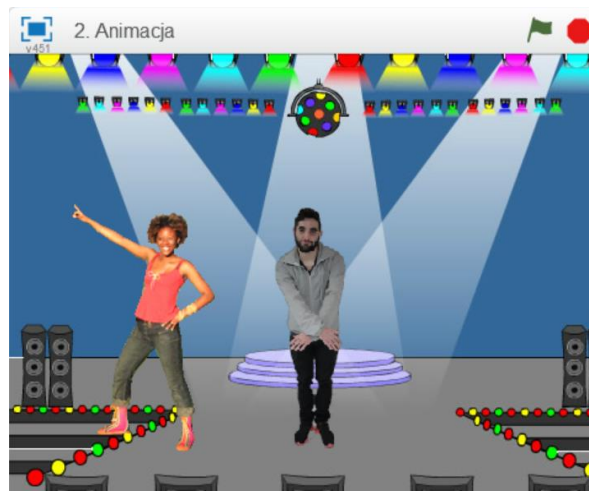
## Temat 2. „Animacja”

### Proponowany czas realizacji

3x45min

### Krótki opis projektu

Przy pomocy dostępnych duszków z biblioteki uczniowie wykonają prostą animację. Po przez połączenie odpowiedniej sekwencji zmiany kostiumów uzyskają wrażenie ruchu postaci. Wykorzystają w tym celu pętle iteracyjne. Zapoznają się także z blockami znajdującymi się w zakładce dźwięki. Zdobyta wiedza będzie bardzo istotna przy realizacji kolejnych projektów. Uczniowie poznają system odbierania i wysyłania komunikatów do porozumiewania się duszków pomiędzy sobą.



### Przebieg lekcji

1. Czym jest animacja i w jaki sposób wygląda proces jej tworzenia?
2. Prezentacja przykładowej animacji wykonanej w Scratchu
3. Zapoznanie z blockami dostępnymi w zakładce dźwięki
4. Prezentacja systemu nadawania i odbierania komunikatów
5. Wspólne wykonanie krótkiej animacji
6. Samodzielna praca nad wykonaniem własnej animacji przez uczniów

### Cele lekcji

Uczeń wie:

- czym jest animacja,
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

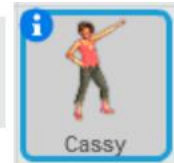
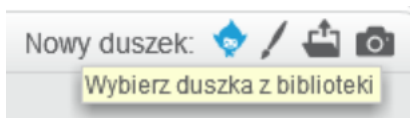
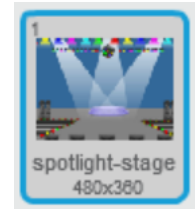
- korzystać z różnych rodzajów pętli,
- wykonać prostą animację,
- za pomocą ciągu poleceń sterować obiektem na ekranie,
- wykorzystać system odbierania i nadawania komunikatów do porozumiewania się duszków

pomiędzy sobą,

- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej.

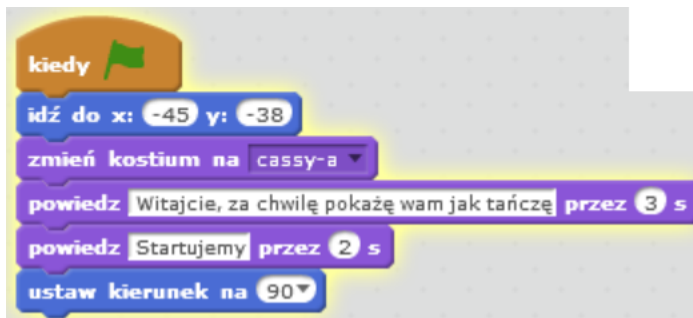
## Instrukcja wykonania gry dla nauczyciela

- I. Tło do naszej gry wybieramy z biblioteki (np. spotlight-stage).
- II. Animacje będziemy wykonywali jako ciąg zmieniających się kostiumów poszczególnych duszków.
- III. Pierwszego duszka wybieramy z biblioteki („Cassy”)



Klikamy na duszka „Cassy” po czym wgrujemy cztery nowe kostiumy z biblioteki: „cassy dance-a”, „cassy dance-b”, „cassy dance-c” oraz „cassy dance-d”.

- IV. Po naciśnięciu Zielonej Flagi Cassy zostaje ustawiona z lewej strony sceny – bloczek „idź do x: ... y: ...”. Kostium zostaje zmieniony na „cassy-a”. Poniżej zostają umieszczone dwa bloczki „powiedz ... przez ... s”, które informują nas o tym co za chwilę się wydarzy. Duszek zostaje ustawiony w prawą stronę.



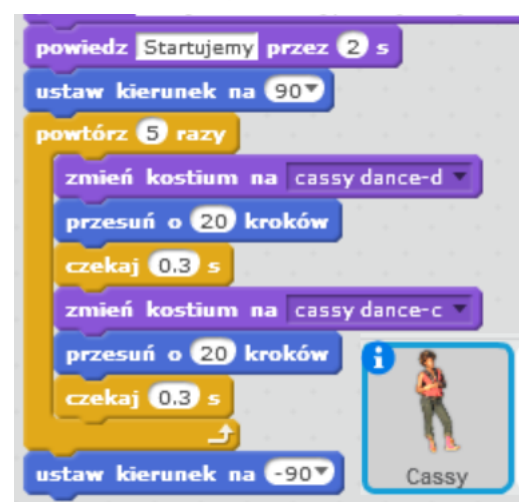
- V. Do przemieszczania naszego duszka będziemy stosować pętlę iteracyjną „powtórz ... razy”. W każdej sekwencji kostium duszka będzie się zmieniał z „cassy dance-d” na z „cassy dance-c”.

Pomiędzy zmianami kostiumów będzie się on przesuwał o 20 kroków.

Bloczek „czekaj 0.3s” umożliwi nam obserwację ruchu. Bez tego bloczka duszek przemieszczał by się tak szybko, że byśmy tego nie zauważyli.

Po dojściu do prawej strony planszy ustawiamy kierunek na „-90”. Duszek obraca się w lewą stronę. Jest odwrócony góra-dół. Klikamy na znaczek „i” na ikonie duszka. Po czym zmieniamy styl

obrotu na lewo-prawo.



styl obrotów: ↻ ↔ ●

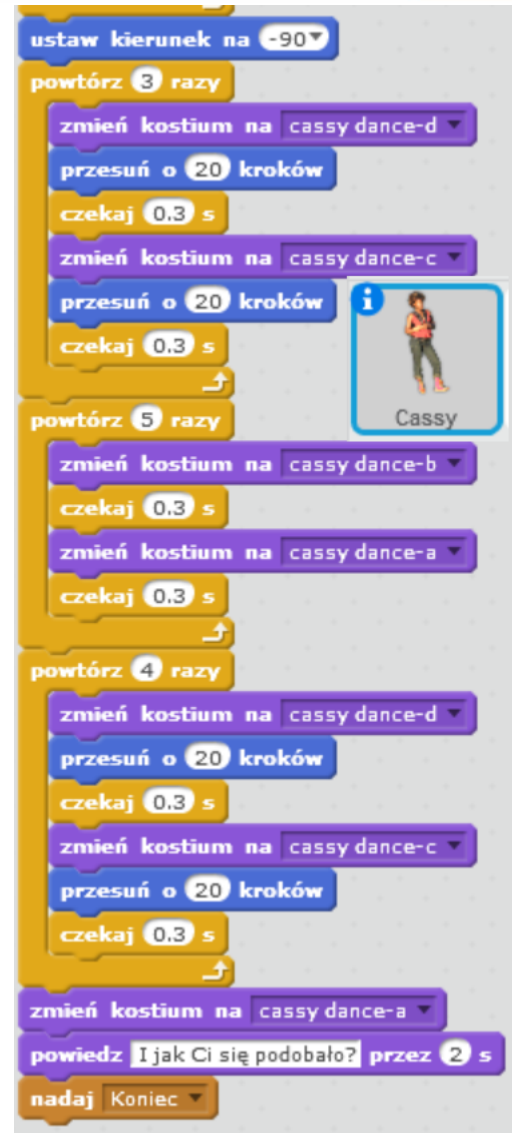
Po obróceniu duszka w lewo ma się on przemieścić na środek planszy. W tym celu ponownie stosujemy pętlę iteracyjną. Tym razem ilość powtórzeń ustawiamy na 3 razy.

Następna pętla odpowiada za taniec duszka na środku sceny. Kostiumy zmieniają się pomiędzy „cassy dance-b”, a „cassy dance-a”, w odstępach 0,3s.

Po wykonaniu sekwencji zmiany kostiumów, duszek przemieszcza się tanecznym krokiem do lewej krawędzi planszy. Ponownie wykorzystujemy pętlę iteracyjną.

Po dotarciu do lewej części sceny kostium duszka zostaje zmieniony na „cassy dance-a”, po czym zostaje umieszczony bloczek „powiedz ... przez ... s”.

Z zakładki Zdarzenia wybieramy bloczek „nadaj message1” i podczepiamy go do naszego skryptu. Z listy rozwijanej wybieramy „nowy komunikat” i wpisujemy „Koniec”.



- VI. Drugiego duszka również wybieramy z biblioteki „CM Hip-Hop”. Zmieniamy jego styl obrotu na lewo-prawo. Po naciśnięciu Zielonej Flagi zostaje on ukryty



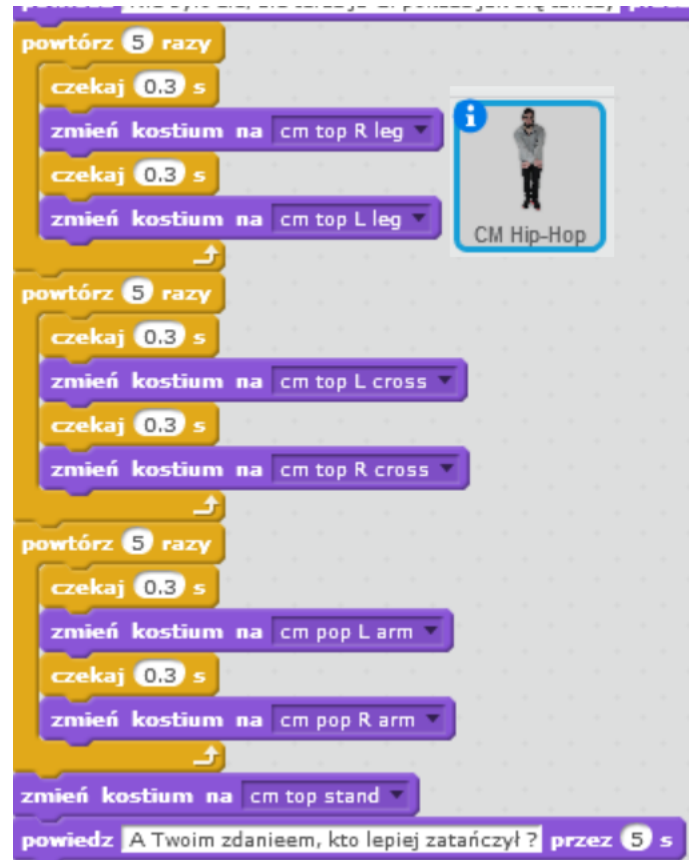
Po otrzymaniu komunikatu „Koniec” pojawia się na środku sceny. Przy pomocy bloczka „pokaż” informuje nas o tym, że teraz on będzie tańczył.





Po przez zastosowanie trzech pętli iteracyjnych będzie odbywała się zmiana kostiumów.

- VII. W skrypcie drugiego duszka umieszczamy drugi skrypt odpowiedzialny za granie dźwięków. Po bločku „kiedy klawisz spacja naciśnięty” w pętli iteracyjnej umieszczamy bloczki odpowiedzialne za granie poszczególnych dźwięków. Znajdziemy je w zakładce dźwięki.



### Zadania dodatkowe do samodzielnej realizacji

Dodanie dialogów pomiędzy duszkami, zmiana wielkości, koloru poszczególnych duszków.

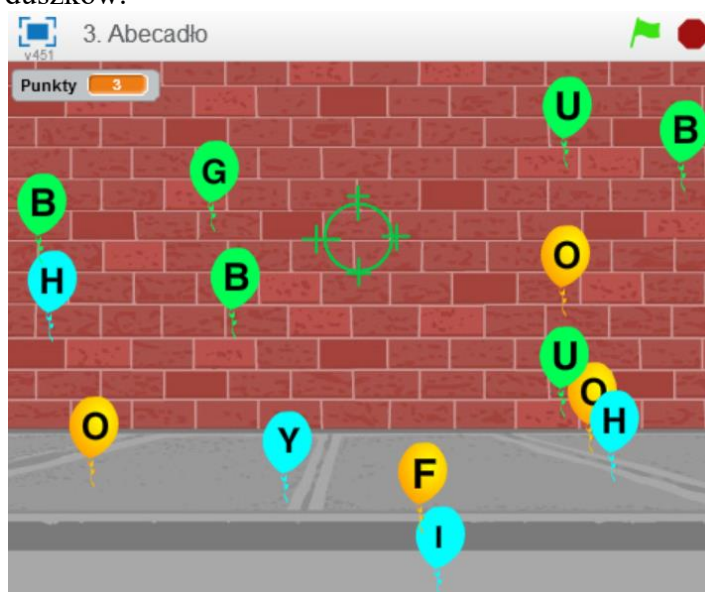
## Temat 3. „Abecadło”

Proponowany czas realizacji

3x45min

### Krótki opis projektu

Uczniowie wykonają grę, w której celownik zawsze będzie podążał za wskaźnikiem myszy. Naszym zadaniem będzie „klikanie” balonów z samogłoskami. Za trafienie w taki balon będziemy otrzymywali punkt. Jeżeli trafimy w balon z spółgłoską jeden punkt zostanie odjęty od aktualnej wartości. W czasie realizacji zajęć uczniowie wykorzystają zmienne, system nadawania oraz odbierania komunikatów oraz generator liczb losowych. Poznają również zastosowania bloczka „połącz ... i ...” z zakładki Wyrażenia. Umożliwi ona wyświetlanie połączonych ze sobą łańcuchów znaków (napisów) z wartościami zmiennych. Wykorzystają także bloczek „sklonuj siebie” do powielania duszków.



### Przebieg lekcji

1. Prezentacja gotowej gry „Abecadło”
2. Dyskusja na temat możliwych rozwiązań programistycznych wykorzystanych do realizacji projektu
3. Przypomnienie informacji na temat zmiennych oraz systemu nadawania i odbierania komunikatów
4. Zapoznanie z bloczkiem generatora liczb losowych z zakładki Wyrażenia „losuj od ... do ...”
5. Zapoznanie z bloczkiem „połącz ... i ...” z zakładki Wyrażenia
6. Przedstawienie działania bloczka „sklonuj siebie” z zakładki Kontrola
7. Wykonanie duszka „Avery”, który będzie wprowadzał nas w świat gry
8. Utworzenie zmiennej odpowiedzialnej za zliczanie punktów
9. Wykonanie oraz zaprogramowanie balona z samogłoskami
10. Wykonanie oraz zaprogramowanie balona z spółgłoskami
11. Ustawienie warunku na koniec gry
12. Wykonanie celownika oraz jego programowanie
13. Podsumowanie zajęć oraz samodzielna praca nad modyfikacjami w grze





## Cele lekcji

Uczeń wie:

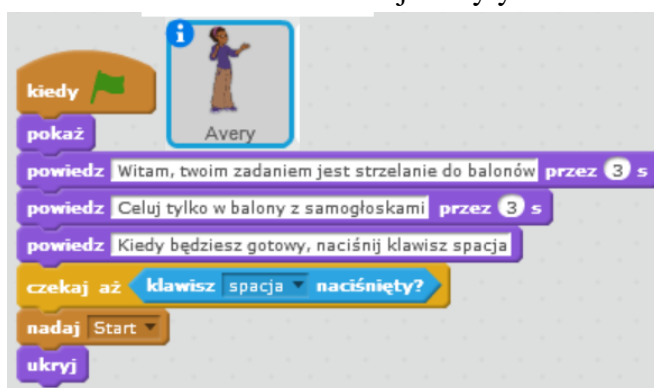
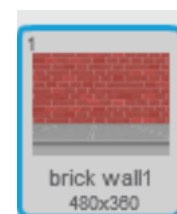
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

- korzystać z różnych rodzajów pętli,
- wykorzystać system odbierania i nadawania komunikatów do porozumiewania się duszków pomiędzy sobą,
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej,
- wykorzystać generator liczb losowych przy wykonywaniu gier,
- użyć bloczka „połącz ... i ...”.

## Instrukcja wykonania gry

- I. Tło do naszej gry wybieramy z biblioteki („brick wall”).
- II. Pierwszego duszka wybieramy również z biblioteki („Avery”). Będzie on wprowadzał nas w tematykę gry. Po powiedzeniu ostatniego zdania będzie czekał na wciśnięcie klawisza spacja. Po spełnieniu warunku zostanie nadany komunikat Start oraz duszek zostaje ukryty.

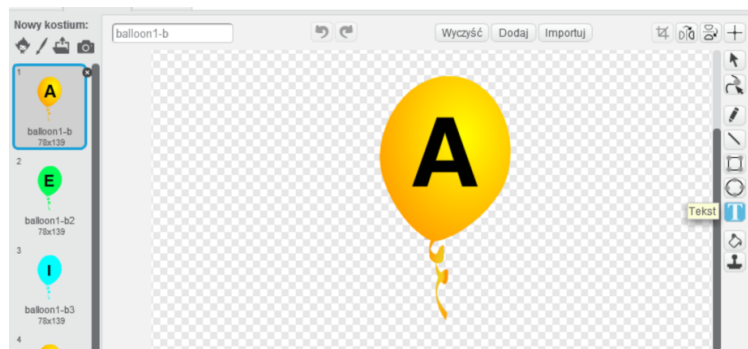


- III. W zakładce dane tworzymy nową zmienną globalną „Punkty”.

- IV. Zamiast tworzyć 6 duszków z samogłoskami wykonamy jednego duszka z 6 różnymi kostiumami. Każdy kostium będzie to osobna samogłoska.

Z biblioteki wybieramy „Balloon1”. Styl obrotu zmieniamy na kropkę (brak obrotu). Nazwę duszka zmieniamy na „Samogłoski”. Wykonujemy 6 różnych kostiumów różniących się kolorami.

Przy pomocy narzędzia „Tekst” umieszczamy na balonach poszczególne samogłoski (A, E, I, O, U, Y).







V. Program duszka „samogłoski” będzie składał się z pięciu podprogramów.

Po naciśnięciu Zielonej Flagi duszek zostaje ukryty. Zmienna „Punkty” zostaje ustawiona na 0.

Po otrzymaniu komunikatu „Start” punkty ponownie są ustawiane na 0. Umożliwi to zapętlenie gry. Poniżej znajduje się pętla iteracyjna odpowiedzialna za klonowanie duszków. Przy każdym przebiegu pętli bloczek „sklonuj siebie” będzie powodował utworzenie kopii duszka. Kolejne duszki będą tworzone w odstępach 1,8 sekundowych (bloczek „czekaj 1.8s”).

Po zakończeniu klonowania program będzie oczekiwał 10s, po czym zatrzymaniu ulegną inne skrypty tego duszka oraz zostanie nadany komunikat „Koniec Gry”.



Z zakładki Kontrola wybieramy bloczek „kiedy zaczynam jako klon”.

Klon zostaje umieszczony w losowym położeniu na planszy (bloczek „losuj od ... do ...”).

Jego kostium zostaje zmieniony na losowy (kostiumy są numerowane od 1 do 6).

Początkowy kierunek balona

zostaje ustawiona na również na losową wartość od 0 do 360.

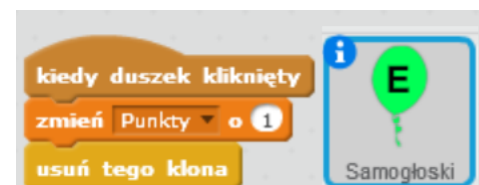
Następnie dany klon się pokazuje i czeka 15s zanim zostanie usunięty z planszy.



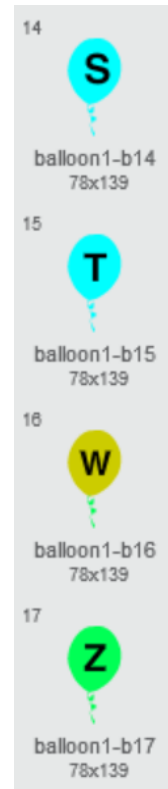
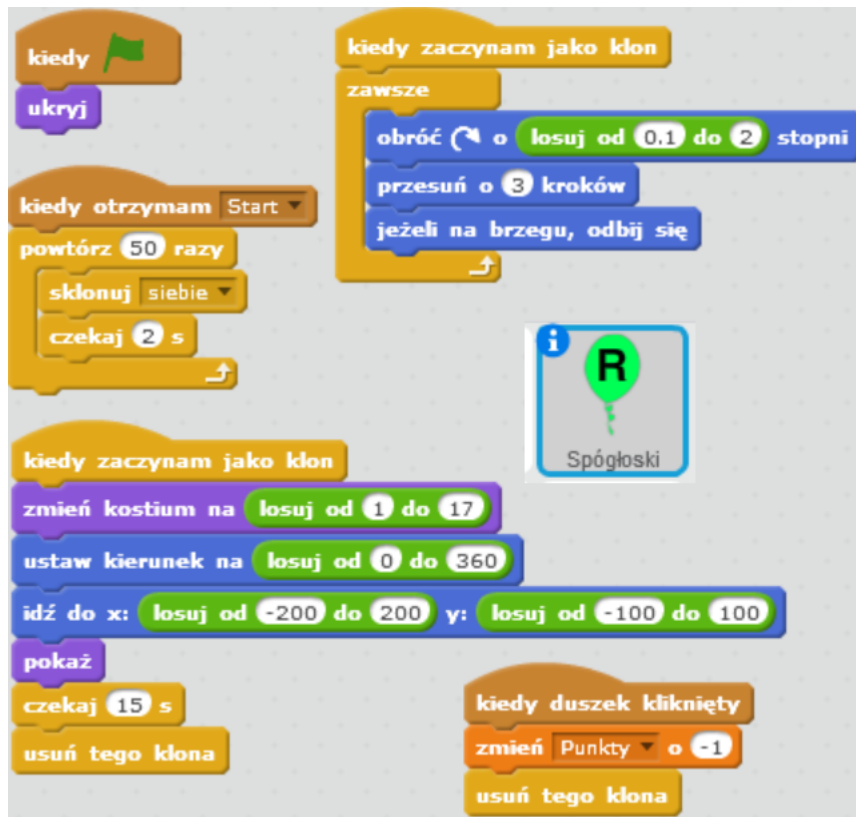
Do kolejnego bloczka „kiedy zaczynam jako klon” podłączona będzie pętla zawsze odpowiedzialna za poruszanie się balona po planszy. Po przez zastosowanie generatora liczb losowych każdy balon będzie poruszał się trochę inaczej. Zwiększenie ilości kroków, o którą jest przesuwany dany klon przyspieszy balona. Jeżeli balon dotrze do brzegu planszy będzie się odbijał.



Ostatni podprogram jest odpowiedzialny za znikanie balonów po ich kliknięciu oraz naliczanie punktów. Do bloczka „kiedy duszek kliknięty” dołączamy bloczek „zmień Punkty o 1” z zakładki Dane oraz bloczek „usuń tego klona” z zakładki Kontrola.



- VI. Kolejnego duszka możemy utworzyć po przez zduplikowanie duszka samogłoski. Zmieniamy jego nazwę na spółgłoski. Przy pomocy edytora obrazu wykonujemy 17 nowych kostiumów z spółgłoskami.



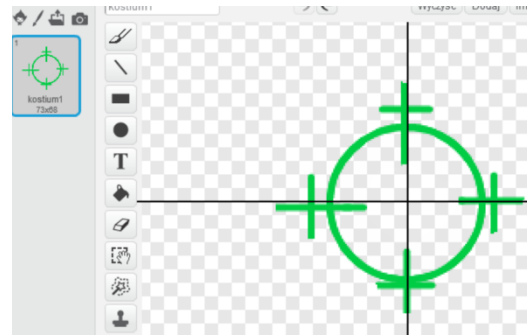
- VII. Dokonujemy następujących modyfikacji w kodzie duszka „spółgłoski”:
- usuwamy bloczek odpowiedzialny za ustawienie punktów na 0,
  - po otrzymaniu komunikatu Start pozostawiamy tylko pętlę iteracyjną, zmieniamy w niej odstęp między utworzeniem kolejnych klonów,
  - kiedy duszek zaczyna jako klon jego kostium może zostać ustawiony na losowy kostium z 17 dostępnych (bloczek „losuj od 1 do 17”),
  - kiedy duszek kliknięty zmieniamy wartość zmiennej „Punkty” o -1.

- VIII. Avery po otrzymaniu komunikatu „Koniec Gry” pokazuje się ponownie. Z zakładki Wyrażenia wybieramy dwa bloczki „połącz ... i ...”. Przy ich pomocy będzie ona mogła nas poinformować ile punktów zdobyliśmy. Jeżeli chcemy zagrać jeszcze raz musimy nacisnąć klawisz spacja. Po spełnieniu tego warunku nadawany jest ponownie komunikat Start rozpoczynający grę. Po nadaniu komunikatu duszek zostaje ponownie ukryty.





- IX. Ostatnim duszkiem jaki dodamy do naszej gry będzie celownik, który będzie „zawsze” podążał za wskaźnikiem myszy. Rysujemy go w edytorze obrazu. **Środek celownika musi pozostać pusty.** Musimy mieć możliwość kliknięcia innych duszków. Przy pomocy narzędzia pomniejsz ustawiam odpowiedni rozmiar naszego duszka. Pamiętajmy o ustawieniu środka kostiumu na samym środku celownika.



- X. Program celownika składa się z 4 bloków: „kiedy Zielona Flaga”, pętli „zawsze”, bloczka ruchu „idź do wskaźnik myszy” oraz bloczka wyglądu „na wierzch”. Ostatni bloczek powoduje to, że celownik nie będzie zniknął za balonami.



### Zadania dodatkowe do samodzielnej realizacji

Dodanie wyboru poziomu trudności na początku rozgrywki

Powiązanie szybkości poruszania się obiektów z liczbą zdobytych punktów



## Temat 4. „W bibliotece”

### Proponowany czas realizacji

3x45min

### Krótki opis projektu

Uczniowie wykonają prosty Quiz matematyczny. Dodatkowo nauczą się w jaki sposób można wprowadzać dane z klawiatury. W czasie realizacji projektu uczniowie będą korzystać z wcześniej poznanych elementów programistycznych takich jak: zmienne, system nadawania i odbierania komunikatów oraz łączenie różnych łańcuchów znaków. Tym razem również wykorzystają zmienną jako znacznika do komunikacji się duszków pomiędzy sobą. Użyją bloczka negacji „nie ...” z zakładki wyrażenia. Zadaniem w grze jest udzielanie prawidłowych odpowiedzi na zadawane pytania matematyczne.



### Przebieg lekcji

1. Przedstawienie uczniom gry „W bibliotece”
2. Przedstawienie możliwości wykorzystania bloczka „zapytaj ...? i czekaj” z zakładki czujniki
3. Wybór tła
4. Wybór duszków do konwersacji
5. Utworzenie zmiennej „Imię”
6. Programowanie „rozmowy” duszków z wykorzystaniem bloczka „zapytaj ... ? i czekaj”
7. Wykonanie dwóch duszków – przycisków TAK/NIE
8. Utworzenie zmiennych „odpowiedz” oraz „Czy\_kliknelismy”
9. Zaprogramowanie przycisków
10. Utworzenie zmiennej „Punkty”
11. Dodanie pytań do Quizu
12. Wykonanie własnego Quizu tematycznego
13. Podsumowanie przebiegu zajęć



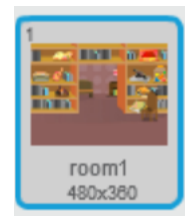
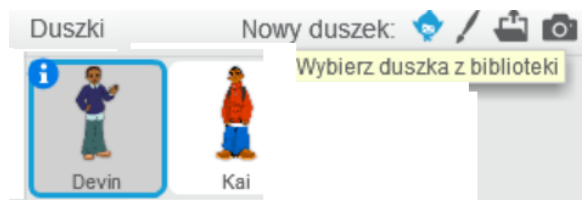
## Cele lekcji

Uczeń potrafi:

- korzystać z różnych rodzajów pętli,
- wykonać prosty Quiz,
- korzystać z zmiennych oraz systemu nadawania i odbierania komunikatów przy tworzeniu gier,
- wykorzystać zmienną jako znacznik do komunikacji się duszków pomiędzy sobą,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej,

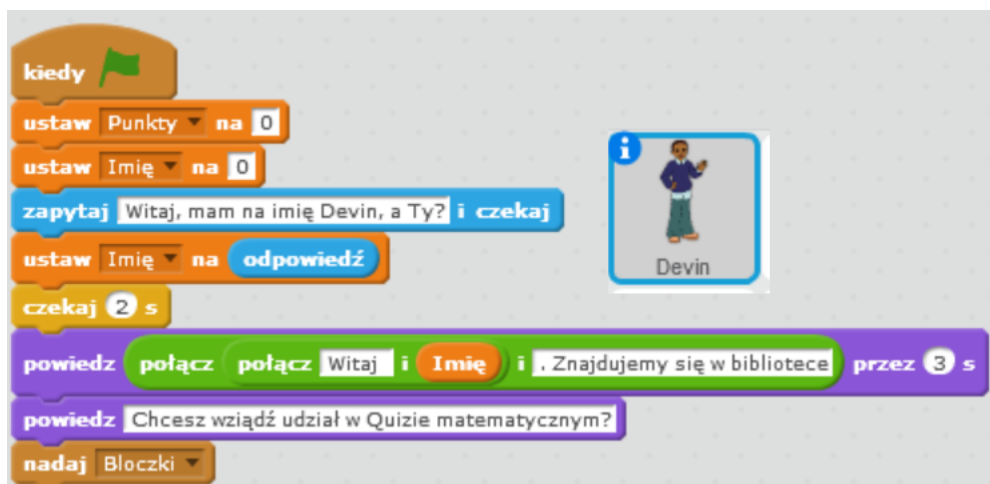
## Instrukcja wykonania gry

- I. Tło do gry wybieramy z biblioteki („room1”).
- II. Wybieramy dwa duszki z biblioteki „Devin” i „Kai” .



- III. Przechodzimy do zakładki dane i tworzymy zmienne „Punkty”, „Czy\_kliknelismy”, „Imię”, „Odpowiedź”.
- IV. Skrypt Devina i Kaia będą wykonywane równoległe. Kwestie wypowiedane przez obydwu duszki będą nakładały się na siebie. Zaczynamy od napisania skryptu Devina.

Po naciśnięciu Zielonej Flagi zmienne Punkty oraz Imię zostają ustawione na 0. Devin pyta nas o imię i oczekuje udzielenia przez nas odpowiedzi. Jeżeli to się stanie zmienna „Imię” zostaje ustawiona na udzieloną przez nas odpowiedź. Następnie czeka 2 s, po czym wita się z nami (do połączenia zmiennej „Imię” z łańcuchem znaków używamy dwóch bloczków „połącz ... i ...”) i zadaje nam pytanie czy chcemy wziąć udział w Quizie. Nadany zostaje komunikat „Bloczki”.





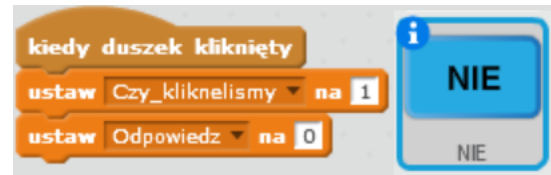
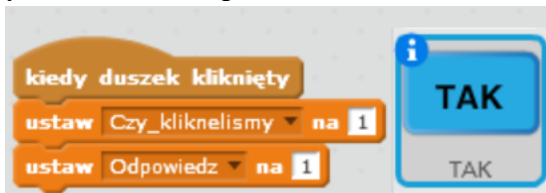
- V. Musimy wykonać dwa przyciski TAK oraz NIE. Przy ich pomocy będziemy odpowiadać na pytania. Będą to dwa duszki.



- VI. Po naciśnięciu Zielonej Flagi oba duszki pozostają ukryte. Wartość zmiennej „Czy\_klikneliśmy” zostaje ustawiona na 0. Będzie ona służyć nam jako znacznik. Wykorzystamy ją zamiast systemu nadawania i odbierania komunikatu. Kiedy przyciski otrzymają komunikat „Bloczki” pokażą się w prawym górnym rogu planszy.



Kiedy Klikniemy przycisk wartość zmiennej „Czy\_klikneliśmy” będzie każdorazowo ustawiana na 1. Przyciskowi TAK odpowiada wartość zmiennej „Odpowiedz” równa 1. Natomiast przyciskowi NIE odpowiada wartość zmiennej „Odpowiedz” równa 0.



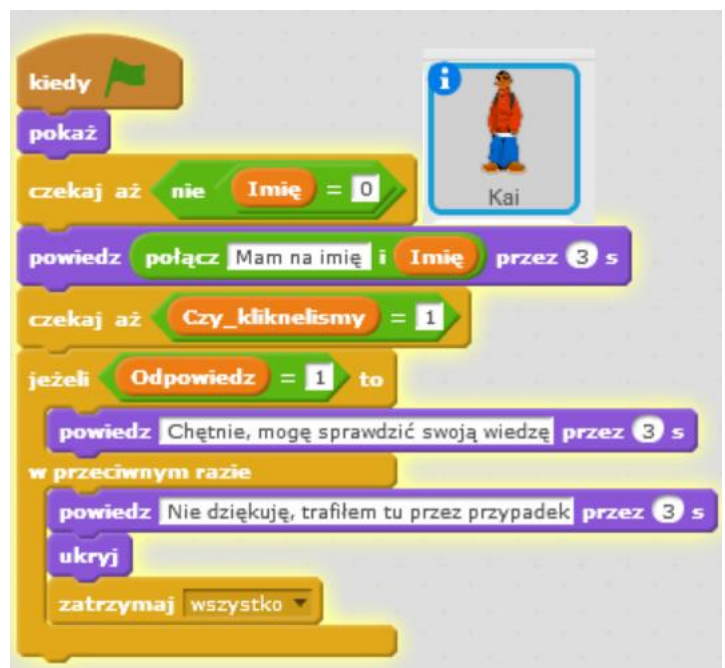
- VII. Przejdźmy do programu Kaia.

Po naciśnięciu Zielonej Flagi duszek się pokazuje.

Będzie oczekiwał, aż wartość zmiennej Imię będzie różna od 0. Stanie się tak wtedy, gdy udzielimy odpowiedzi na pytanie „Jak masz na Imię?”. Po wpisaniu Imienia duszek mówi jak ma na imię. Wykorzystujemy do tego bloczki „powiedz ... przez ... s” oraz „połącz ... i ...”.

Duszek będzie oczekiwał, aż zmienna „Czy\_klikneliśmy” będzie równa 1. Stanie się tak po kliknięciu jednego z dwóch duszków „TAK” lub „NIE”.

Po spełnieniu warunku porównujemy wartość zmiennej odpowiedź z 1. Jeżeli kliknęliśmy duszka „TAK” zmienna „Odpowiedz” jest ustawiana na 1, wtedy duszek mówi, że chętnie sprawdzi swoją wiedzę. Po kliknięciu duszka „NIE” zmienna „Odpowiedz” jest ustawiana na 0. Nie jest spełniony warunek. Wykonuje się instrukcja „w przeciwnym razie”. Duszek mówi, że trafił do biblioteki przez przypadek i znika. Zatrzymaniu ulegają wszystkie skrypty.





VIII. Davin po nadaniu komunikatu Błoczki, również będzie oczekiwał na udzielenie przez nas odpowiedzi (kliknięcie duszka „TAK” lub „NIE”) . Po kliknięciu przycisku będzie czekał 4s. Zada nam pytanie, po czym ustawi wartość zmiennej „Czy\_kliknelismy” na 0.

Znów będzie oczekiwał na kliknięcie któregoś duszka „TAK” lub „NIE”. Po kliknięciu wartość zmiennej „Czy\_kliknelismy” zostaje ustawiona na 1.

Przy pomocy bloczku porównania, porównujemy wartość naszej odpowiedzi (wartość zmiennej „Odpowiedz”) z prawidłową odpowiedzią.



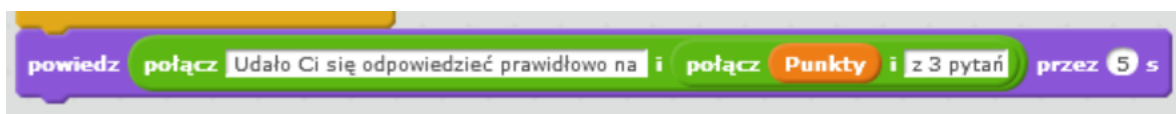
Jeżeli obie liczby są sobie równe, wyświetla się informacja o tym, że udzieliliśmy prawidłowej odpowiedzi i zyskujemy 1 Punkt.

W przeciwnym wypadku jest wyświetlana informacja o tym, że udzieliliśmy błędnej odpowiedzi.

Kolejne pytania tworzymy w sposób analogiczny do pierwszego. Możemy zduplikować część skryptu od bloczka „powiedz Pytanie 1, przez 2s”. Zmieniamy treść pytania. Ewentualnie zmieniamy liczbą odpowiedzialną za prawidłową odpowiedź.



Na końcu skryptu wstawiamy bloczek „powiedz ... przez ... s” informujący nas o ilości zdobytych punktów.



### Zadania dodatkowe do samodzielnej realizacji

Rozszerzenie Quizu o dodatkowe pytania.

Wykonanie własnego Quizu tematycznego z większą ilości możliwych odpowiedzi.

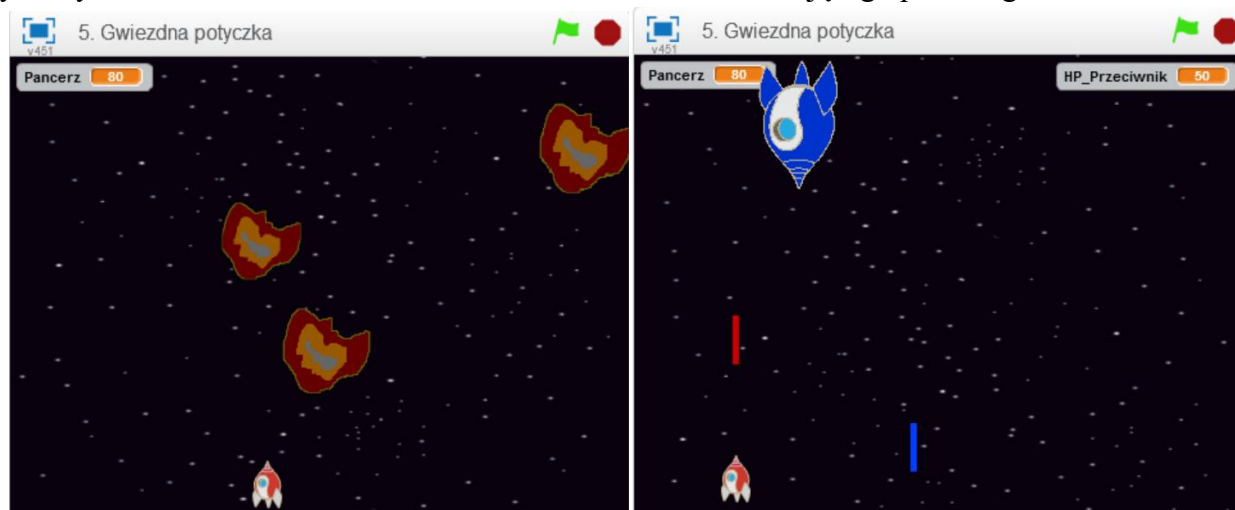
## Temat 5. „Gwiezdna potyczka”

### Proponowany czas realizacji

4x45min

### Krótki opis projektu

Gra składa się z dwóch części połączonych w jedną grę. Na początku rozgrywki lecimy przez kosmos, unikając nadlatujących asteroid. Dodatkowo całe tło pod nami również się przemieszcza. Efekt przemieszczania się gwiazd uzyskujemy, po przez zastosowanie przesuwających się duszków, zastępujących tło. Po pewnym czasie napotykamy na swojej drodze nieprzyjacielski statek kosmiczny, który zaczyna do nas strzelać. Tło się zatrzymuje. Musimy unikać ostrzału, jednocześnie próbując go trafić. Przy wykonywaniu tego projektu będziemy korzystać głównie z systemu nadawania i odbierania komunikatu oraz bloczka klonującego poszczególne duszki.



### Przebieg lekcji

1. Prezentacja gotowej gry „Gwiezdna potyczka”
2. Dyskusja na temat możliwych rozwiązań programistycznych wykorzystanych do realizacji projektu
3. Przypomnienie informacji na temat zmiennych oraz systemu nadawania i odbierania komunikatów
4. Wykonanie przesuwających się duszków, zastępujących tło
5. Wybór statku z biblioteki oraz napisanie programu nim sterującego
6. Narysowanie asteroidy
7. Wykonanie programu umożliwiający klonowanie się asteroidy oraz przelot klonów pomiędzy krawędziami planszy
8. Dodanie zmiennej odpowiedzialnej za określenie stanu życia statku
9. Dodanie warunku odejmującego życie statku po zderzeniu z asteroidom
10. Dodanie drugiego statku – przeciwnika
11. Narysowanie dwóch nowych duszków – prostokątów o różnych kolorach
12. Wykonanie programu umożliwiającego strzelanie
13. Stworzenie sztucznej inteligencji sterującej naszym przeciwnikiem
14. Ustawienie warunków na wygraną oraz przegraną gry



15. Podsumowanie zajęć

16. Samodzielna praca uczniów nad dalszymi modyfikacjami w grze

## Cele lekcji

Uczeń wie:

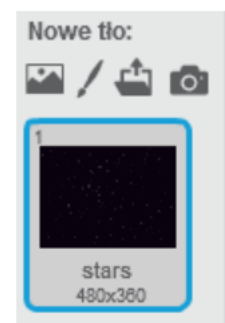
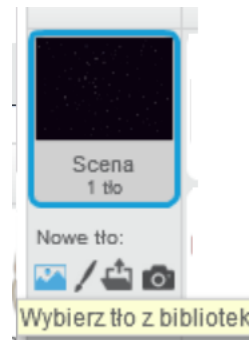
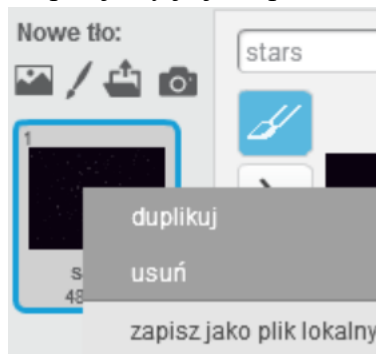
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

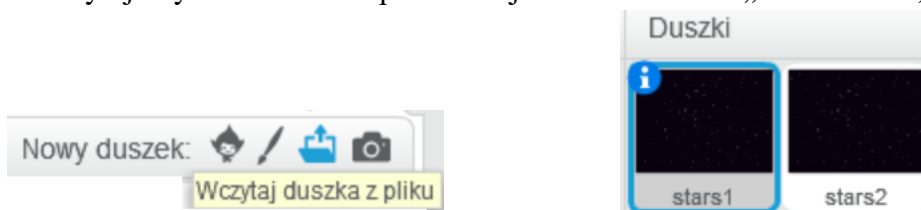
- korzystać z różnych rodzajów pętli,
- wykorzystać system odbierania i nadawania komunikatów do porozumiewania się duszków pomiędzy sobą,
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej,
- wykorzystać generator liczb losowych przy wykonywaniu gier.

## Instrukcja wykonania gry

- I. Tło do naszej gry będą stanowiły dwa duszki, które na przemian będą przesuwać się po ekranie. Z biblioteki wybieramy tło „stars”. Klikamy prawym przyciskiem myszy na tło i zapisujemy je jako plik lokalny.



- II. Wczytujemy dwukrotnie zapisane tło jako nowe duszki „stars1” oraz „stars2”.



- III. Oba tła będą się poruszać synchronicznie przez planszę. Duszek „stars1” zaczyna się poruszać od współrzędnych (0,0) do współrzędnych (0.-350), natomiast duszek „stars2” od współrzędnych (0,350) również do współrzędnych (0.-350), ale w innym czasie.



Duszki będą się przemieszczać na zmianę, w odstępach 3 sekundowych. W tym celu używamy pętli „zawsze” oraz dwóch bloczków ruchu. Czas przemieszczania się tła można modyfikować. Obydwa czasy przelotu powinny być takie same. Czas pierwszego przelotu duszka „stars1” powinien stanowić połowę czasu całego przelotu.

- IV. Kolejnego duszka wybieramy z biblioteki „Spaceship”. Możemy go również narysować samodzielnie przy pomocy edytora obrazu. Styl obrotu duszka ustawiamy na kropkę (brak obrotu). Przy pomocy narzędzia pomniejsz zmniejszamy jego wielkość.



- V. Przechodzimy do zakładki Dane i tworzymy zmienną „Pancerz”.

- VI. Program sterujący naszym duszkiem ma umożliwiać mu przemieszczanie się tylko w prawą i lewą stronę.

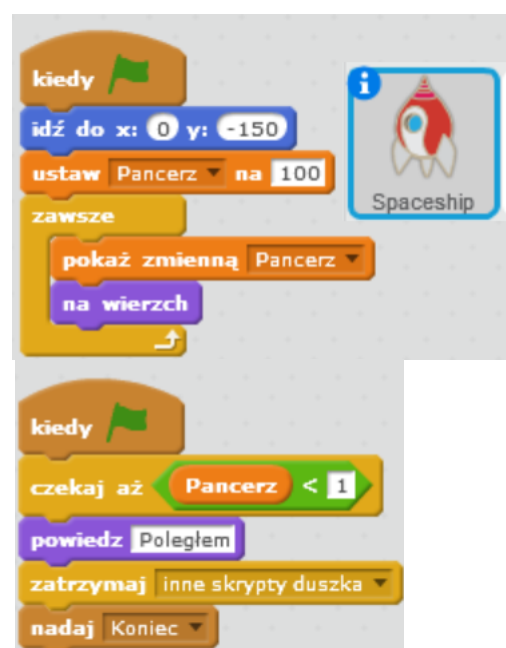


Nasz statek znika pod duszkami tła. W celu uwidocznienia go stosujemy bloczek „na wierzch” umieszczony w pętli zawsze.

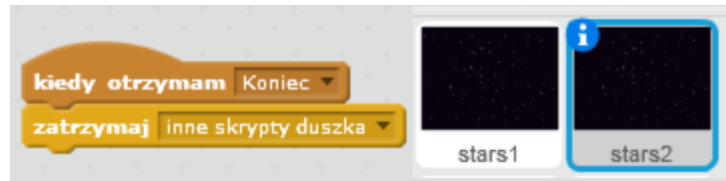
Po naciśnięciu Zielonej Flagi statek jest umieszczany na dole planszy. Zmienna „Pancerz” zostaje ustawiona na 100.

W pętli zawsze po za bloczkiem „na wierzch” umieszczamy również bloczek „pokaż zmienną Pancerz”. Spowoduje on, że zmienna ta będzie cały czas widoczna.

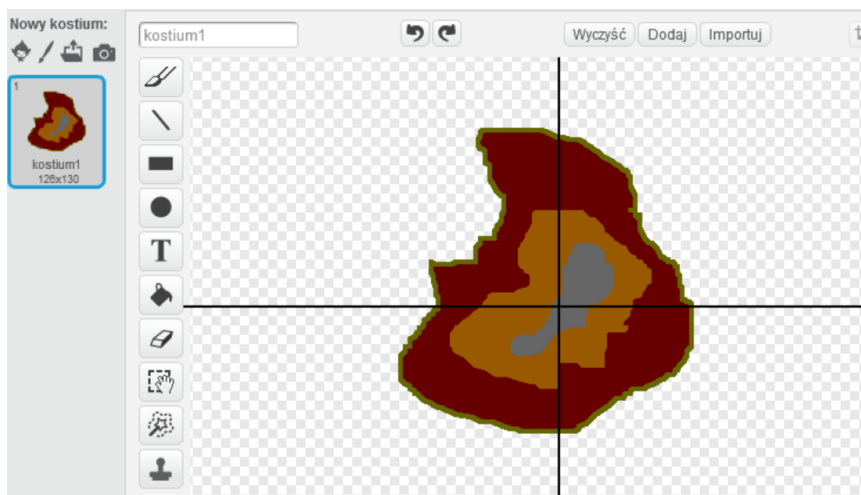
Drugi skrypt zaczynający się od Zielonej Flagi będzie oczekiwał, aż wartość zmiennej „Pancerz” będzie mniejsza niż 1. Po spełnieniu tego warunku duszek informuje nas o tym, że przegraliśmy. Zatrzymaniu ulegają inne skrypty duszka oraz nadawany jest komunikat „Koniec”.



- VII. Duszki „stars1” oraz „stars2” po otrzymaniu komunikatu „Koniec” również się zatrzymują. Skrypt odpowiedzialny za zatrzymanie innych skryptów umieszczamy w obu tych duszkach.



- VIII. Przy pomocy opcji „Namaluj nowego duszka” wykonujemy asteroidę. Pamiętajmy o tym, aby ustawić środek kostiumu po środku asteroidy.



- IX. Po naciśnięciu Zielonej Flagi asteroida zostaje ukryta. Następnie zaczyna się klonować 50 razy w odstępach 0,5 sekundowych. Po zakończeniu pętli iteracyjnej czeka 5 sekund, a następnie nadaje komunikat Walka. Jest to sygnał do przejścia do kolejnego etapu gry.

- X. Kiedy obiekt zaczyna jako klon jego rozmiar jest ustawiany na losową wartość (zakres liczbowy zależy od wielkości narysowanej przez nas asteroidy).

Kierunek ruchu jest ustawiony na 180.

Klon się pokazuje w losowym położeniu przy górnej krawędzi planszy.

Pierwsza pętla iteracyjna powoduje to, że duszek odlatuje od górnej krawędzi planszy. Wewnątrz tej pętli umieszczamy bloczki „jeżeli na brzegu, odbij się na wierzch” oraz „przesuń o 5 kroków”.

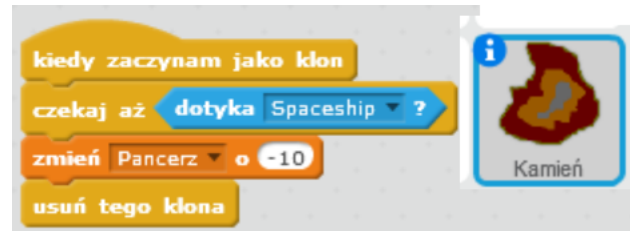
Kolejna pętla jest to pętla warunkowa. Duszek tak długo przemieszcza się w dół planszy, aż nie dotknie krawędzi.

Po zakończeniu wykonywania pętli warunkowej dany klon jest usuwany.

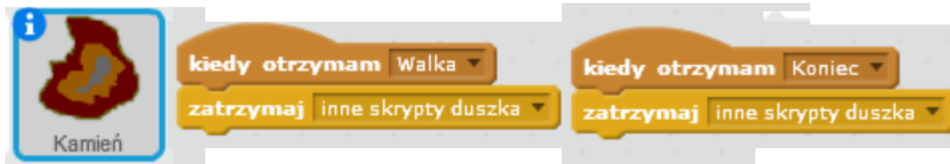




- XI. Po zderzeniu klonu asteroidy ze statkiem wartość zmiennej Pancierz jest zmieniana o -10, a następnie dany klon zostaje usunięty.

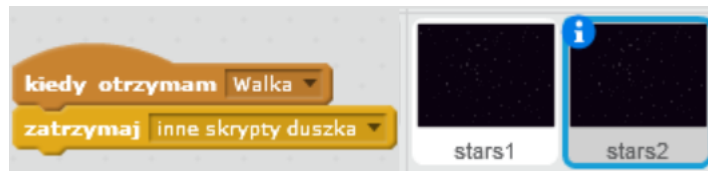


- XII. Skrypty Asteroidy po otrzymaniu komunikatu Walka oraz Koniec ulegają zatrzymaniu.



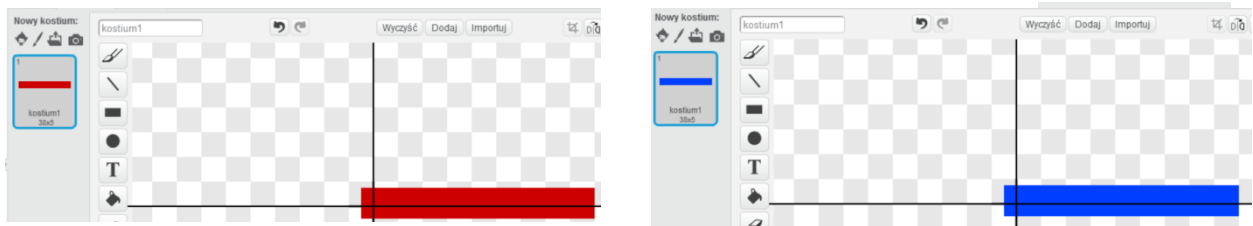
Wykonaliśmy właśnie pierwszą część gry.

- XIII. Skrypty duszków „stars1” oraz „stars2” po otrzymaniu komunikatu „Walka” ulegają zatrzymaniu. Skrypt odpowiedzialny za zatrzymanie innych skryptów umieszczamy w obu tych duszках.

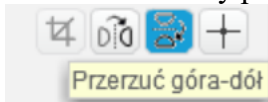


- XIV. Przechodzimy do zakładki Dane i tworzymy nową zmienną „Pancerz Przeciwnika”

- XV. Przy pomocy edytora obrazu wykonujemy dwa kolejne duszki: „Nasz laser” oraz „Laser przeciwnika”. Oba duszki różnią się pomiędzy sobą kolorem. Tym razem środek kostiumu ustawiamy z lewej strony duszka.



- XVI. Przechodzimy do wykonania naszego przeciwnika. Będzie to również statek kosmiczny. Wybieramy go z Biblioteki. Przy pomocy edytora obrazu zmieniamy jego kolor na niebieski. Kostium możemy przetrzucić góra-dół.



- XVII. Program naszego przeciwnika będzie składał się z 3 podprogramów.

Po naciśnięciu Zielonej Flagi duszek idzie do współrzędnych (0,120). Zmienna „Pancerz Przeciwnika” zostaje ukryta, a jej wartość zostaje ustawiona na 100. Sam duszek, również zostaje ukryty.







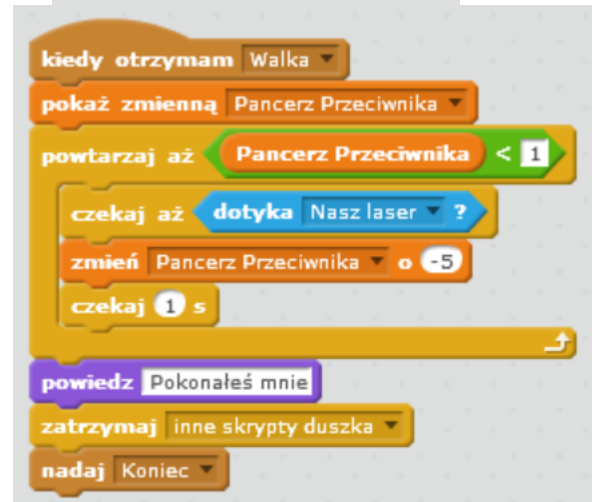
Po otrzymaniu komunikatu „Walka” pokazuje się. Zawsze będzie ustawiał się naprzeciwko statku. Jednak jego położenie wzdłuż osi y pozostaje stałe. W pętli zawsze umieszczamy dodatkowo bloczek „na wierzch”.



Ostatni podprogram statku przeciwnika odpowiada za reakcję na „Nasz laser”.

Po otrzymaniu komunikatu Walka zmienna „Pancerz Przeciwnika” pokazuje się.

Pętla warunkowa będzie powtarzana tak długo, aż zmienna „Pancerz Przeciwnika” nie będzie mniejsza niż 1. W każdej iteracji bloczek „czekaj, aż” będzie oczekiwał na zderzenie z duszkiem „Nasz laser”. Jeżeli warunek ten zostanie spełniony zmienna „Pancerz Przeciwnika” zmienia się o -5. Następnie umieszczony jest bloczek „czekaj 1 s”. Daje on czas laserowi na dotarcie do końca planszy.



Kiedy wartość zmiennej „Pancerz Przeciwnika” spadnie poniżej 1, wyświetli się informacja o tym, że wygraliśmy. Zatrzymaniu ulegną inne skrypty tego duszka oraz nadany jest komunikat „Koniec”.

XVIII. Przejdźmy teraz do zaprogramowania duszków laserów. Zaczniemy od duszka „Nasz laser” – laser koloru czerwonego.

Po naciśnięciu Zielonej Flagi duszek zostaje ukryty.

Kiedy duszek otrzyma komunikat „Koniec” zatrzymaniu ulegają inne skrypty tego duszka.



Po otrzymaniu komunikatu „Walka” zawsze po naciśnięciu klawisza spacja duszek będzie się klonował. Kolejne kolony nie będą powstawały częściej niż co 0.5 sekundy.



Kiedy duszek zaczyna jako klon pokazuje się.

Jego kierunek zostaje ustawiony na 0.

Idzie do duszka „Spaceship”.

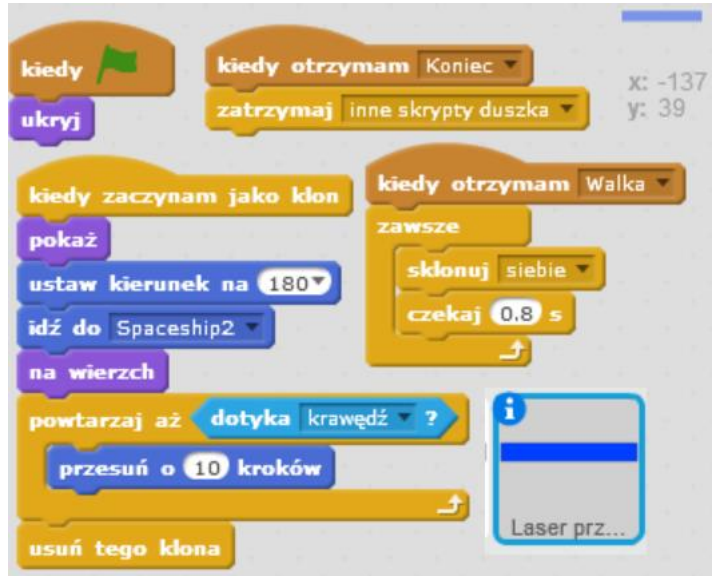
Przesuwa się na wierzch.

Będzie się przemieszczał w górę planszy. Tak długo, aż nie doleci do krawędzi. Kiedy to się stanie dany klon jest usuwany.

- XIX. Program duszka „Laser przeciwnika” będzie wyglądał podobnie do programu poprzedniego duszka.

Po otrzymaniu komunikatu „Walka” duszek będzie się sam klonował co 0,8 sekundy.

Kiedy duszek będzie zaczynał jako klon jego kierunek zostanie ustawiony na 180 oraz idzie do duszka „Spaceship2”. Będzie przemieszczał się w dół.



- XX. Ostatnią rzeczą, którą musimy dodać jest podprogram duszka „Spaceship” powodujący, że duszek „Laser przeciwnika” będzie zmniejszał wartość naszego pancerza.

Po otrzymaniu komunikatu „Walka” za każdym razem po trafieniu przez „Laser przeciwnika” wartość zmiennej „Pancerz” będzie zmieniana o -10.



### Zadania dodatkowe do samodzielnej realizacji

Dodanie wyboru poziomu trudności na początku gry.

Stworzenie kontynuacji gry. Po pokonaniu statku, przeciwnik ucieka, a my lecimy dalej. Asteroidy przyspieszają. Po pewnym czasie znów napotykamy nieprzyjacielski okręt.

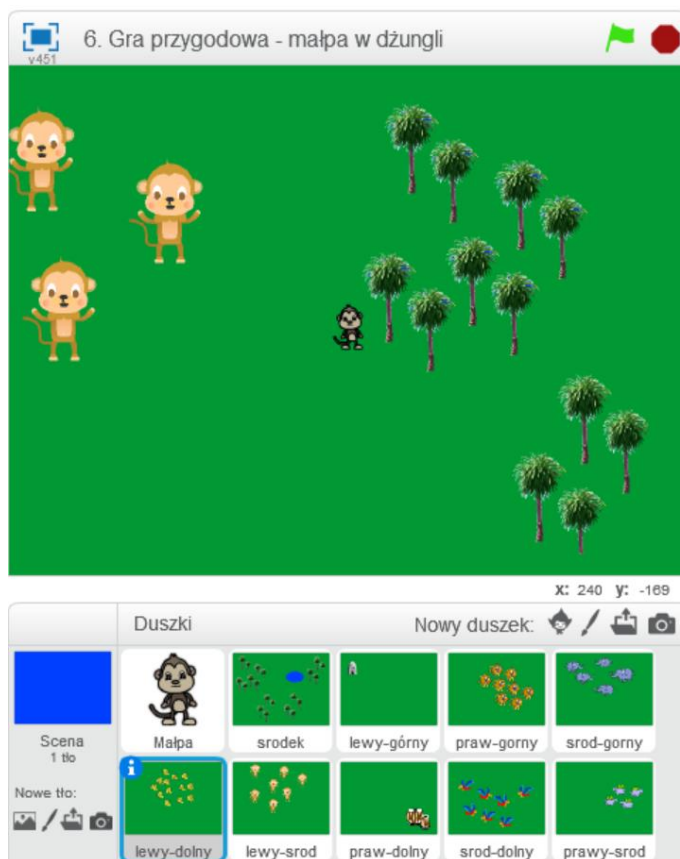
## Temat 6. „Gra przygodowa – małpa w dżungli”

Proponowany czas realizacji

3x45min

### Krótki opis projektu

Na zajęciach zostanie zaprezentowany mechanizm umożliwiający rozszerzenie pola rozgrywki. Zamiast tła zastosowane zostaną przesuwane się duszki. Spowodują one wrażenie poruszania się głównej postaci gry po planszy, pomimo to, że będzie ona cały czas w bezruchu. Główną postacią w grze jest małpa, która przemieszcza się po dżungli. Na każdej z dziewięciu plansz gry znajdują się inne obiekty. Pole gry może zostać dowolnie rozszerzane przez uczniów.



### Przebieg lekcji

1. Przedstawienie uczniom gry przygodowej „Małpa w dżungli”
2. Dyskusja nad użytymi rozwiązaniami programistycznymi
3. Porównanie gry z projektami realizowanymi w czasie wcześniejszych zajęć
4. Wybór duszka z biblioteki oraz wykonanie dodatkowych kostiumów
5. Utworzenie zmiennej odpowiedzialnej za przesuwanie się duszków tła
6. Napisanie programu zmieniającego kostium małpy w zależności od wciśniętej strzałki
7. Wykonanie pierwszego duszka tła - „srodek”
8. Zaprogramowanie przesuwania się duszka po wciśnięciu odpowiedniego przycisku
9. Wykonanie pozostałych ośmiu duszków tła oraz ich programowanie
10. Wspólna praca nad ustaleniem scenariusza gry
11. Ustawienia warunków na wygraną oraz przegraną
12. Samodzielna praca nad modyfikacją gry oraz rozszerzeniem fabuły

## Cele lekcji

Uczeń wie:

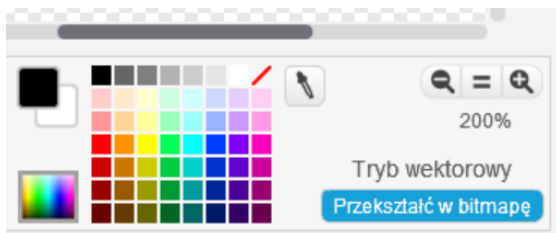
- czym jest pętla i jak można ją wykorzystać w programowaniu,
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

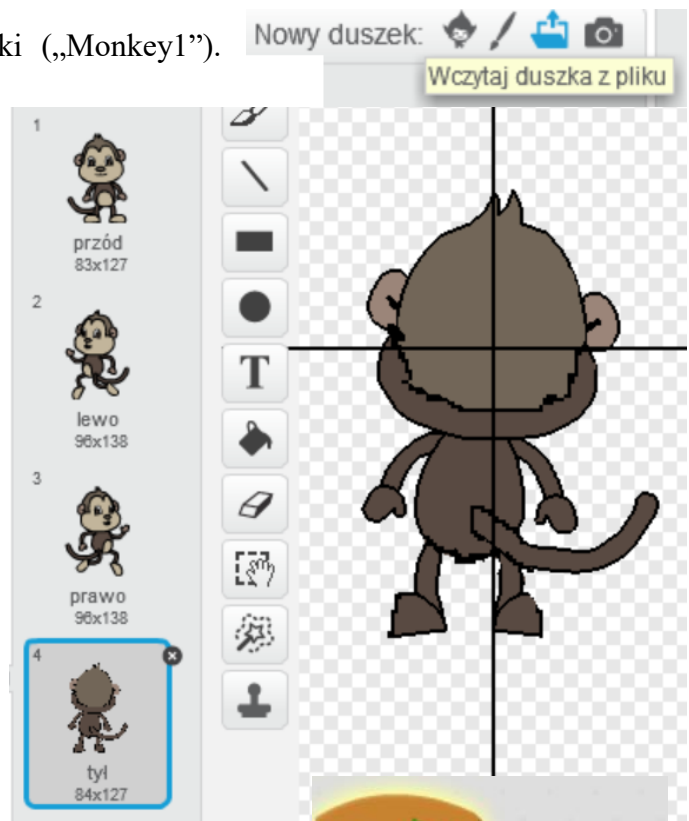
- korzystać z różnych rodzajów pętli,
- za pomocą ciągu poleceń sterować obiektem na ekranie,
- wykorzystać zmienne do ustawienia położeń poszczególnych duszków,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej,
- wykorzystać mechanizm przesuwania się tła do sprawienia wrażenia ruchu bohatera gry.

## Opis wykonania gry krok po kroku:

- I. Główną postać gry wybieramy z biblioteki („Monkey1”). Możemy zmienić nazwę duszka na „Małpa”.



Nasza małpa musi posiadać cztery kostiumy, przedstawiające ruch we wszystkie cztery strony. Przód i prawą stronę już mamy. Kopiujemy drugi kostium i przeliczamy go lewo-prawo. Pierwszy kostium również kopiujemy i przeliczamy lewo-prawo. Przekształcamy go w bitmapę. Przy pomocy dostępnego edytora obrazu przerabiamy odwrócony przód małpy na tył.



- II. Przed przystąpieniem do pisania skryptu małpy, przechodzimy do zakładki Dane. Tworzymy dwie nowe zmienne globalne „położenie\_x” oraz „położenie\_y”. Są one odpowiedzialne za przesuwanie duszków zastępujących tło naszej planszy.





- III. Po naciśnięciu Zielonej Flagi obie zmienne „położenie\_x” oraz „położenie\_y” zostają ustawione na 0.

W pętli zawsze umieszczamy bloczek „na wierzch”. Chcemy żeby ten duszek znajdował się przed duszkami zastępującymi tło (był widoczny).

- IV. Do drugiej Zielonej Flagi przyłączona jest pętla zawsze. Znajdują się w niej instrukcje odpowiedzialne za zmianę wartości zmiennych „położenie\_x” oraz „położenie\_y”.

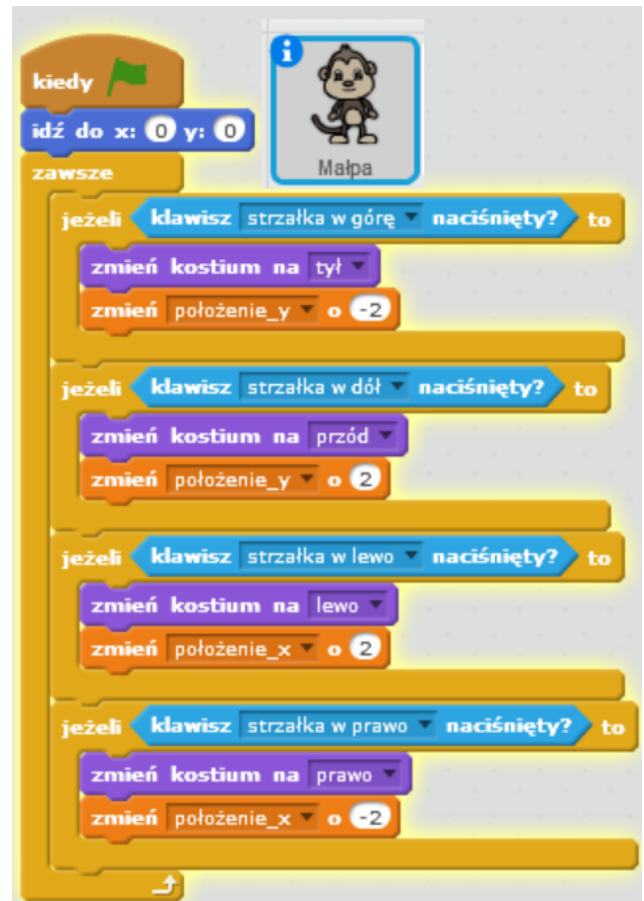
Wewnątrz pętli są umieszczone cztery bloczki instrukcji warunkowych „jeżeli ... to” odpowiedzialne za sprawdzanie, która spośród czterech strzałek jest wciśnięta. Każdej jest przyporządkowany odpowiedni bloczek zmiany kostiumu.

Jeżeli idziemy w górę, wszystkie duszki zastępujące tło są przesuwane w dół wzdłuż osi y zmienna „położenie\_y” zmienia się o -2.

Po naciśnięciu następuje sytuacja odwrotna. Wszystkie duszki będą przesuwać się w górę. Zmienna „położenie\_y” zmienia się o 2.

Gdy idziemy w lewo duszki przesuwają się w prawo. Zmienna „położenie\_x” zmienia się o 2.

Gdy idziemy w prawo duszki przesuwają się w lewo. Zmienna „położenie\_x” zmienia się o -2.



- V. Tło naszej gry może być jednokolorowe (np. ciemnoniebieski).

- VI. Wybieramy nowego duszka z biblioteki (np. Palmtree). Pomniejszamy palmę. Przy pomocy narzędzia „zaznacz i duplikuj” kopiujemy ją kilkakrotnie. Resztę planszy wypełniamy

zielonym kolorem. Używając kształtów możemy narysować sadzawkę.

**Pamiętajmy o tym, że pracujemy na duszku, a nie na tle.**

Będzie to plansza startowa naszej gry. Zmieniamy nazwę duszka na „srodek”.





- VII. Cały skrypt duszka „srodek” składa się z sześciu bloczków. Po naciśnięciu Zielonej Flagi duszek będzie umieszczany w położeniu równym aktualnym wartościom zmiennych „położenie\_x” oraz „położenie\_y”. Wpisywane wartości liczbowe będą określały przesunięcie względem pierwszej planszy.



- VIII. Druga plansza względem pierwszej będzie przesunięta o -470 pikseli wzdłuż osi x. O tyle właśnie będzie zawsze modyfikowana wartość położenia x.

Wykonujemy pozostałe 8 planszy. Na każdej umieszczamy różne elementy graficzne. Przy ich wykonywaniu możemy wykorzystać dostępne duszki z biblioteki. Rozmieszczenie duszków względem siebie oraz wartości, o które będą modyfikowane poszczególne współrzędne zostały zamieszczone w tabeli poniżej. Uzyskamy pole gry składające się z 9 pól.

x: -470; y: 350  lewy-górny	x: 0 ; y: 350  srod-gorny	x: -475; y: 350  praw-gorny
x:-470 ; y: 0  lewy-srod	x:0 ; y: 0  srodek	x-470 ; y: 0  prawy-srod
x:-470; y:-350  lewy-dolny	x:0 ; y: -350  srod-dolny	x:-475; y:-350  praw-dolny



IX. Po wykonaniu duszków dodajemy do nich skrypt odpowiedzialny za ich przemieszczanie.

The image displays eight separate Scratch code snippets, each consisting of a 'kiedy' (when green flag clicked) event block, a 'zawsze' (always) loop block, and an 'idź do x: y:' (go to x: y:) block. The 'idź do x: y:' block contains mathematical expressions for calculating the x and y coordinates based on the object's current position ('położenie\_x' and 'położenie\_y').

- lewy-górny:**  $x: -470 + \text{położenie}_x$ ,  $y: 350 + \text{położenie}_y$
- lewy-srodo:**  $x: -470 + \text{położenie}_x$ ,  $y: 0 + \text{położenie}_y$
- lewy-dolny:**  $x: -470 + \text{położenie}_x$ ,  $y: -350 + \text{położenie}_y$
- srodo-górny:**  $x: 0 + \text{położenie}_x$ ,  $y: 350 + \text{położenie}_y$
- srodo-dolny:**  $x: 0 + \text{położenie}_x$ ,  $y: -350 + \text{położenie}_y$
- praw-górny:**  $x: 470 + \text{położenie}_x$ ,  $y: 350 + \text{położenie}_y$
- praw-srodo:**  $x: 470 + \text{położenie}_x$ ,  $y: 0 + \text{położenie}_y$
- praw-dolny:**  $x: 470 + \text{położenie}_x$ ,  $y: -350 + \text{położenie}_y$



- X. Wykonaliśmy mechanizm umożliwiający poszerzenia pole naszej rozgrywki. Możemy przystąpić teraz do tworzenia fabuły naszej gry. Możemy dodawać pojedyncze elementy duszki do poszczególnych pól.

#### Zadania dodatkowe do samodzielnej realizacji

Rozbudowanie planszy gry o dodatkowe pola, dodatkowe elementy na poszczególnych planszach.  
Stworzenie własnej fabuły gry.



## Temat 7. „Bloons Defense”

### Proponowany czas realizacji

4x45min

### Krótki opis projektu

Wykonanie gry, w której broni się twierdzy przed nadlatującymi balonami. Głównym celem tych zajęć jest przypomnienie idei programowania równoległego, wykorzystania systemu nadawania i odbierania komunikatów do porozumiewania się duszków pomiędzy sobą. Uczniowie poznają zasadność stosowania zmiennych lokalnych.



### Przebieg lekcji

1. Przedstawienie uczniom gry „Bloons Defense”
2. Wykonanie planszy gry
3. Zaprogramowanie ruchu balonów
4. Utworzenie odpowiednich zmiennych
5. Wykonanie pierwszego działka oraz jego zaprogramowanie
6. Wykonanie pierwszego pocisku oraz jego zaprogramowanie
7. Wykonanie kolejnych działek oraz pocisków
8. Modyfikacja skryptu balonu
9. Samodzielne prace na modyfikacjami gry
10. Podsumowanie zajęć

### Cele lekcji

Uczeń wie:

- czym jest pętla i jak można ją wykorzystać w programowaniu,
- w jaki sposób wykorzystać system nadawania i odbierania komunikatów do porozumiewania się duszków pomiędzy sobą,
- czym jest zmienna lokalna oraz globalna.



Uczeń potrafi:

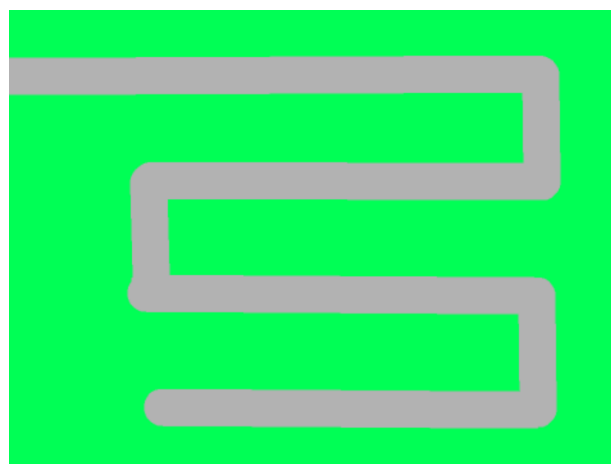
- korzystać z różnych rodzajów pętli,
- wykonać prostą grę z wykorzystaniem zmiennych lokalnych oraz systemem nadawania i odbierania komunikatów,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej,

## Instrukcja wykonania gry

Instrukcja wykonania gry została podzielona na dwie części.

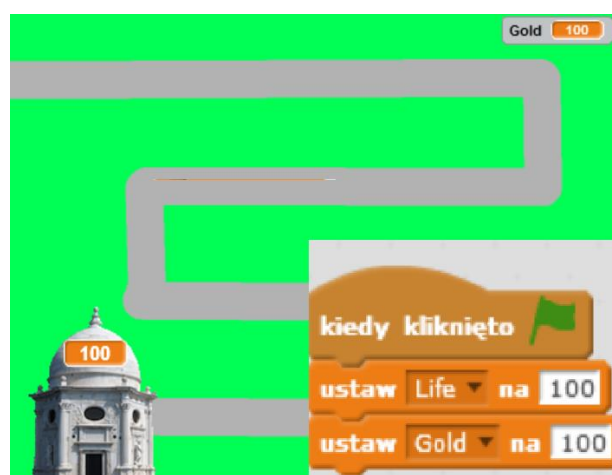
### Bloons Defense cz.1

- I. Wykonywanie gry rozpoczynamy od narysowania tła. Na jednolitym tle rysujemy ścieżkę, po której będą poruszały się balony. Zwracamy uwagę, na to, żeby ścieżki zakreślały pod kątem 90 stopni.



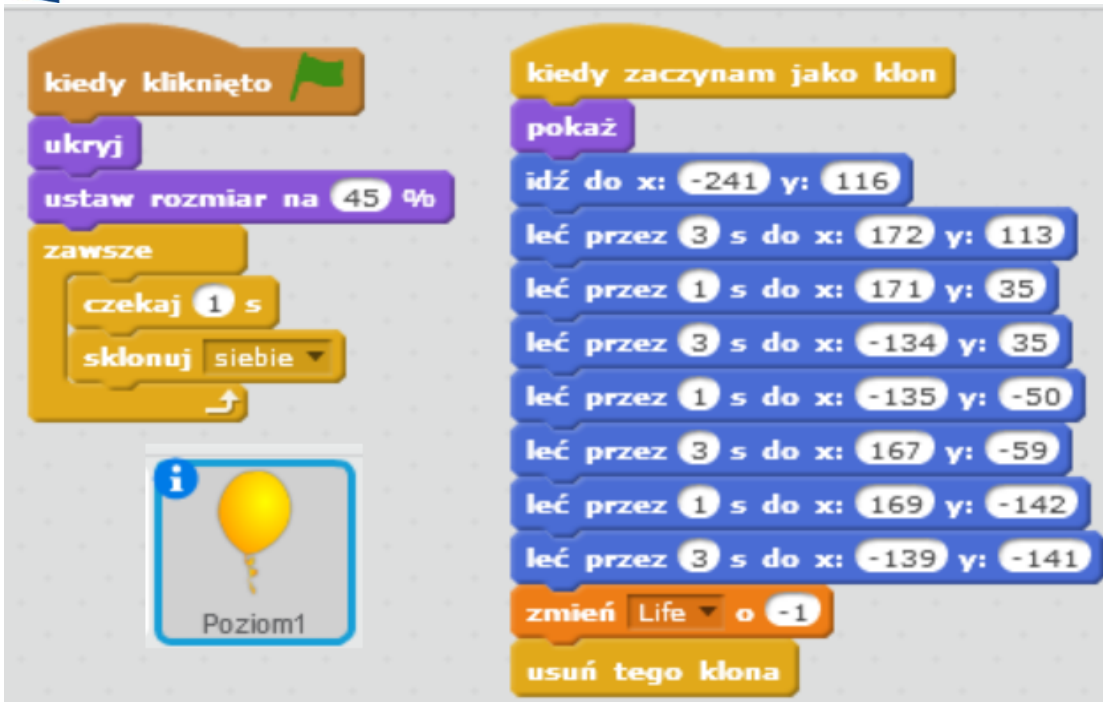
- II. Na końcu ścieżki umieszczamy duszka twierdzy/zamek. Możemy wybrać gotowego duszka z biblioteki (np. „Marble Building”). Tworzymy dwie zmienne – życie twierdzy „Life” oraz „Gold” (złoto które będzie wykorzystywane do budowy i ulepszania wieżyczek strzelniczych).

Obie zmienne na początku rozgrywki (po naciśnięciu ZIELONEJ FLAGI) są równe 100. Punkty życia twierdzy można umieścić na jej kopule.



- III. Kolejnego duszka również wybieramy z biblioteki - „Balloon1”. Balony będą poruszać się po ścieżce. W celu uproszczenia procesu tworzenia gry wykorzystamy jednego balona, który będzie się klonował.





Balon pozostaje ukryty, po czym co 1s będzie ulegał klonowaniu.

Klon balonu pojawia się w współrzędnych odpowiadających współrzędnym początku ścieżki, po czym przesuwa się kolejno pomiędzy współrzędnymi na zakrętach. Po osiągnięciu współrzędnych twierdzy balon odbiera jej 1 życia, po czym jest usuwany.

#### IV. Wykonanie pierwszego działka

Tworzymy nowego duszka – działko1, który będzie posiadał cztery kostium (0, 1, 2, 3).

Pierwszy jest to kwadrat który nic nie robi czeka aż na niego klikniemy, jeżeli to zrobimy i będziemy mieli dostateczną ilość złota zmieni się on w działko.

**TWOŻYMY NOWĄ ZMIENNA LOKALNA (przypisaną tylko do tego duszka) KTÓRA BĘDZIE ODPOWIEDZIALNA ZA POZIOM NASZEGO DZIAŁKA.**

#### **ZMIENNA LOKALNA – „poziom”**

Skrypt zmieniający kostiumy duszka (w celu sprawdzenia poprawności skryptu możemy zmniejszyć wymaganą ilość złota wymaganą do przejścia na wyższy poziom). Do jego wykonania będziemy potrzebowali trzech komunikatów: „1\_1”, „1\_2” oraz „1\_3”.



Program będzie składał się z trzech podprogramów.

1) Po naciśnięciu Zielonej Flagi:

- duszek pojawia się,
- przenosi się do określonej współrzędnej na planszy,
- zmienia kostium na kostium 0 (na początku jest kwadratem),
- ustawiamy rozmiar na x% (tak żeby duszek nie zajmował całej planszy),
- ustawiamy wartość zmiennej „poziom „na 0.

2) Po naciśnięciu duszka są sprawdzane trzy warunki, na tym etapie tylko pierwszy ma szansę zostać wykonany (poziom = 0), przy warunkach stosujemy spójnik logiczny „i”. Sprawdzany jest również warunek na ilość złota. Kolejne poziomy są coraz droższe.

Kiedy duszek jest kliknięty,  $Gold > 49$  oraz  $poziom = 0$ :

- nadawany jest komunikat 1\_1 (komunikat dla pierwszego duszka poziom 1),
- tracimy odpowiednią ilość złota,
- kostium zmienia się na kolejny („1”),
- ustaw rozmiar na x% (wraz z poziomem nasz duszek „rośnie”),
- flaga (zmienna „poziom”) zostaje ustawiona na 1 – pierwszy warunek nie będzie już spełniony,



Po ponownym kliknięciu poziom jest równy 1 jedyny warunek jaki może być realizowany to drugi warunek:

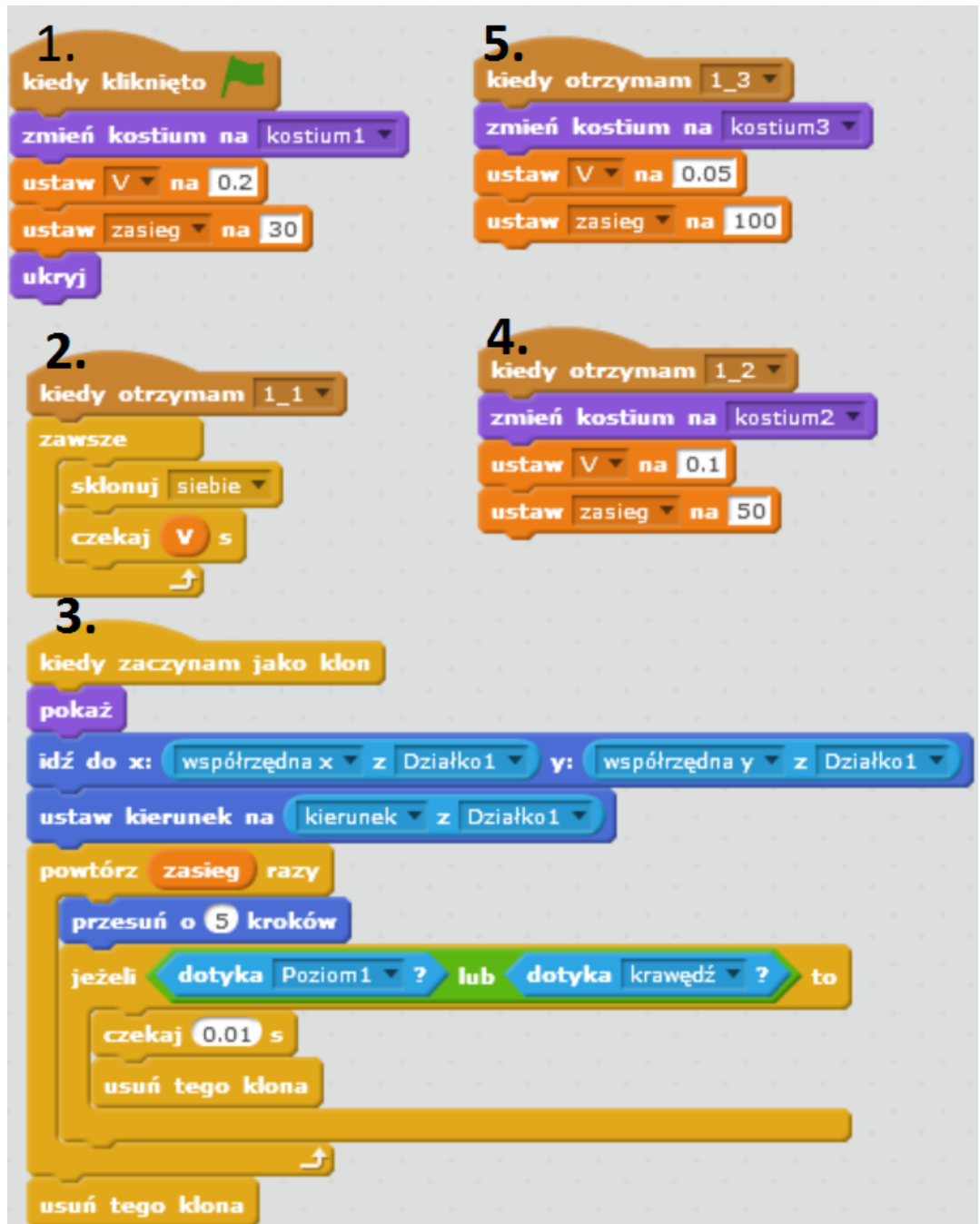
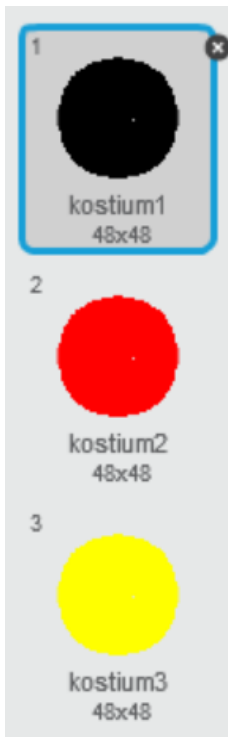
- zostaje nadany komunikat 1\_2,
- tracimy odpowiednią ilość złota,
- kostium zmienia się na kolejny (2),
- ustaw rozmiar na  $x+5\%$  (nasz duszek „rośnie”),
- zmienna „poziom” zostaje ustawiona na 2 – drugi i pierwszy warunek nie będą spełnione, będzie spełniony częściowo trzeci.

Po kolejnym kliknięciu i dostatecznej ilości złota 3 warunek „jeżeli ... to ...” jest wykonany. Wszystkie polecenia są wykonywane podobnie jak poprzednio. Poziom zostaje ustawiony na 3. Zostaje nadany komunikat 1\_3.

3) Kiedy duszek otrzyma komunikat 1\_1 będzie cały czas ustawiony w stronę wskaźnika myszy.

V. Każde działko będzie posiadało przypisany swój własny pocisk, który będzie się klonował.

Malujemy duszka pocisk1, posiada on trzy kolory odpowiadające kolejnym poziomom wieżyczek.



TWORZYMY DWIE ZMIENNE LOKALNE „V” (prędkość, szybkość klonowania) oraz „zasięg”. Odpowiadają one za prędkość oraz maksymalny zasięg naszych pocisków.

1. Po naciśnięciu „Zielonej Flagi”:

- kostium zostaje zmieniony na 1,
- zostają ustawione zmienne „V” na 0.2, a „zasięg” na 30,
- pocisk zostaje ukryty.

2. Kiedy duszek otrzyma komunikat „1\_1”

- zostaje uruchomiona pętla, która będzie działać aż do zakończenia programu
- pociski zaczynają się klonować co V s.

3. Kiedy duszek zaczyna jako klon

- klon pocisku pokazuje się,



- idzie do współrzędnych w których znajduje się działko,
- ostawia kierunek na kierunek działka,
- pętla będzie powtarzana przez „zasięg” razy, im większa ilość powtórzeń tym dalej doleci nasz pocisk,
- w każdym kolejnym powtórzeniu pętli pocisk przesuwany jest o x kroków,
- oraz sprawdzany jest warunek czy nie dotknął balona lub nie doleciał do końca planszy, jeżeli tak to zostaje usunięty,
- bloczek oczekiwania czekaj 0.01s został użyty po to, aby balon zdążył zareagować na uderzenie,
- po zakończeniu pętli klon pocisku zostaje usunięty (kończy się jego zasięg).

4. Kiedy duszek otrzyma komunikat „1\_2”:

- kostium zostaje zmieniony na kostium 2,
- zmienna „V” zostaje ustawiona na 0.1 (pocisk przyspieszy),
- zmienna „zasięg” zostaje ustawiona na 50 (pocisk polecą dalej).

5 . Kiedy duszek otrzyma komunikat „1\_3”:

- kostium zostaje zmieniony na kostium 3,
- zmienna „V” zostaje ustawiona na 0.05 (pocisk przyspieszy),
- zmienna „zasięg” zostaje ustawiona na 100 (pocisk polecą dalej).

VI. Pozostało nam dodanie do skryptu balona instrukcji która będzie powodowała że dany klon po zetknięciu z pociskiem będzie znikał, a my otrzymamy złoto.

Dodatkowo zastosujemy pętlę iteracyjną, która spowoduje, że dany balon będzie musiał zostać trafiony 5 razy zanim zostanie usunięty – będzie posiadał 5 żyć.



## Bloons Defense cz.2

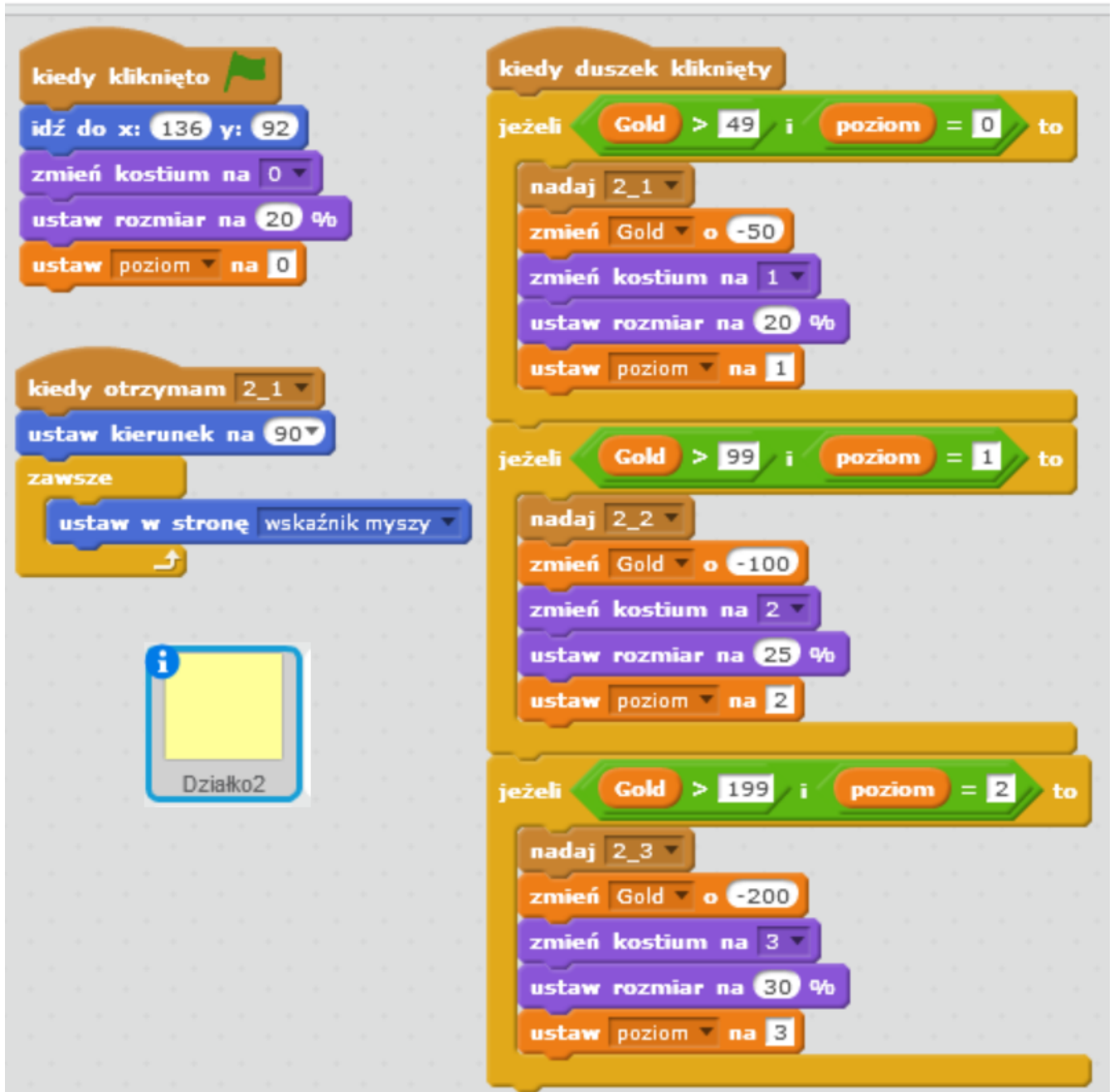
I. Kolejne działka tworzymy po przez zduplikowanie pierwszego. Skrypt po modyfikacjach został zamieszczony poniżej .

Tworzymy trzy nowe komunikaty 2\_1, 2\_2 i 2\_3, zamieniamy komunikaty 1\_1 na 2\_1; 1\_2 na 2\_2, 1\_3 na 2\_3. Zmieniamy również współrzędne początkowe położenia duszka idź do x:... y:..... na współrzędne miejsca, w którym ma być umieszczony nasz duszek. Reszta skryptu nie ulega zmianie, Poziom jest zmienną lokalną i dotyczy tylko działka 2.



Kolejna działka wykonamy w analogiczny sposób (na razie ograniczymy się do 3 działek), po przez:

- skopiowanie działka numer 2
- zmianę współrzędnych początkowych,
- zmianę nadawanych komunikatów na 3\_1, 3\_2, 3\_3.



II. Kolejny pocisk tworzymy po przez skopiowanie duszka „pocisk1”. Zmieniamy nazwy otrzymywanych komunikatów: 1\_1 na 2\_1; 1\_2 na 2\_2, 1\_3 na 2\_3.

KIEDY ZACZYNAM JAKO KLON zmieniamy z Działko1 na Działko2.

Koniec modyfikacji skryptu.

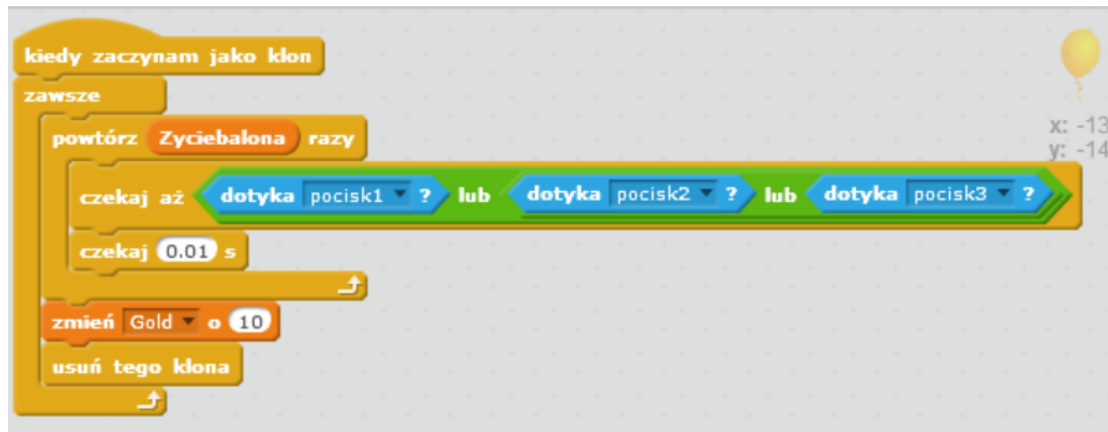


III. Pocisk numer trzy wykonujemy analogicznie:

- kopiujemy duszka „pocisk2”,
- zmieniamy „działko2” na „działko3”,
- zmieniamy otrzymywane komunikaty na „3\_1”, „3\_2”, „3\_3”.

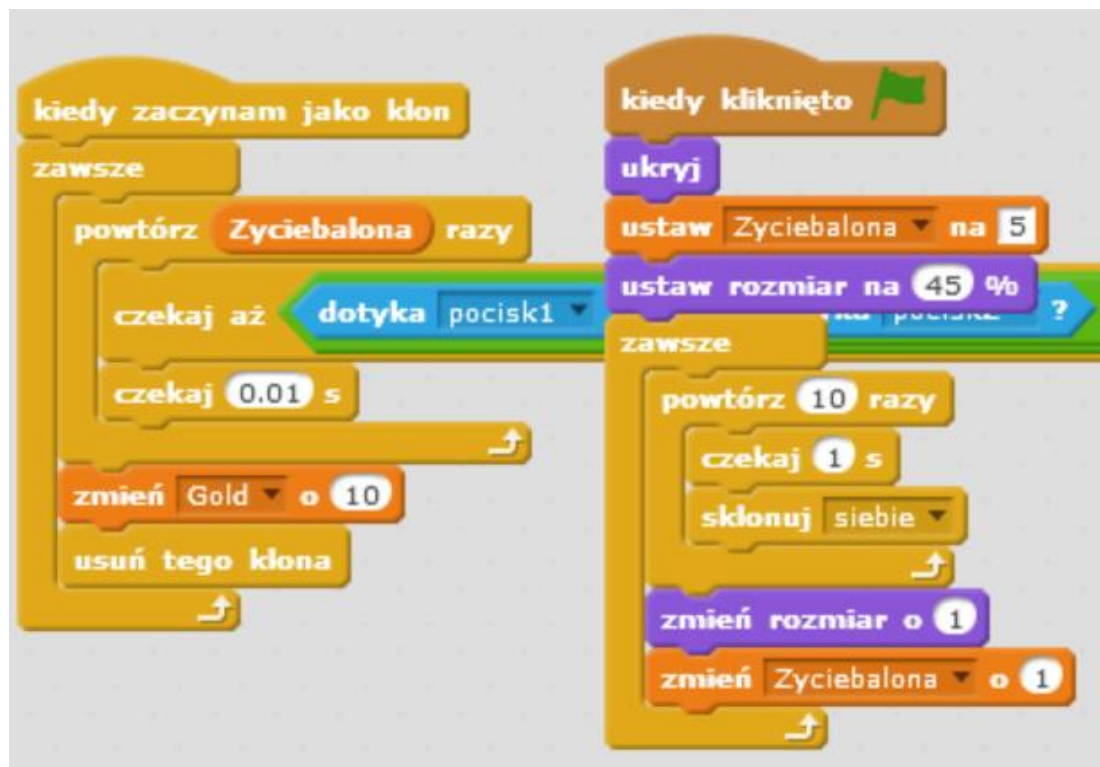
IV. Modyfikacja skryptu balona

Uzupełniamy skrypt balonu o to, żeby reagował na wszystkie trzy pociski (używamy spójnika logicznego lub). Przy dodawaniu kolejnych działek na planszy należy dodać kolejne warunki.



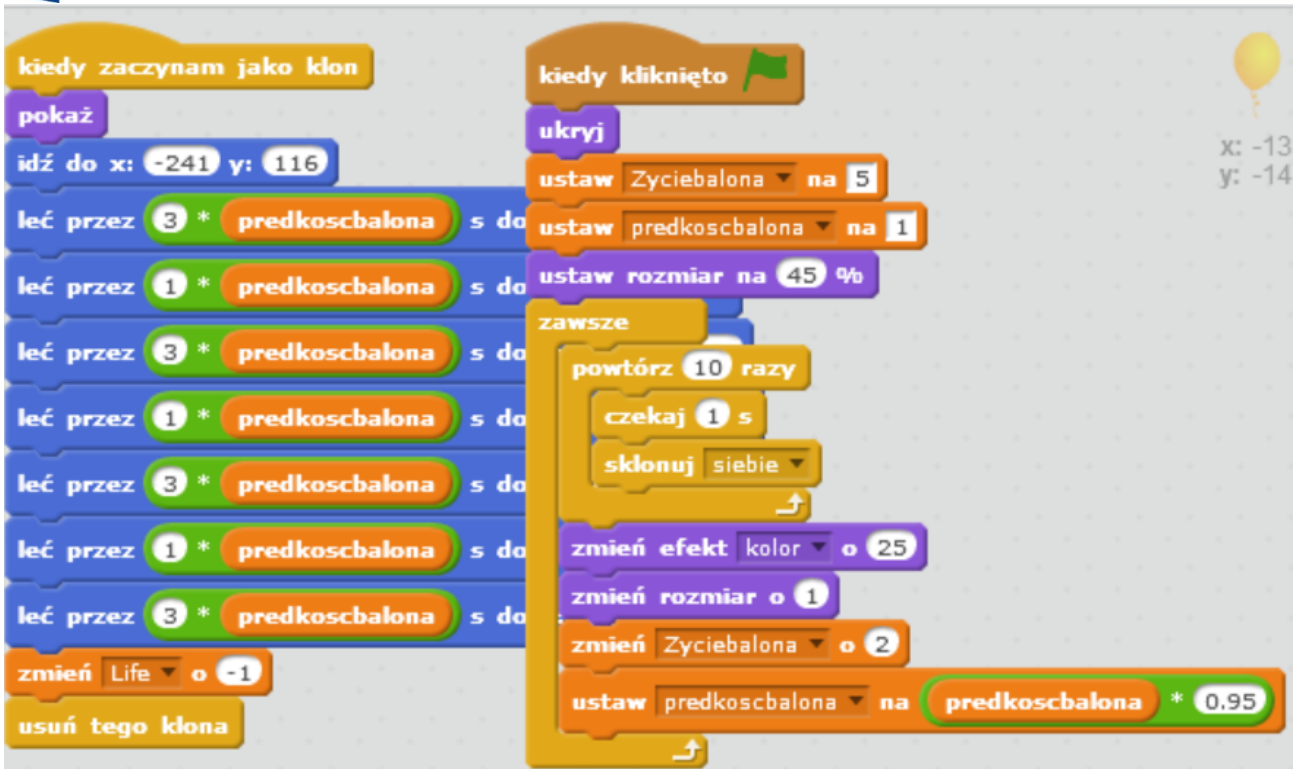
#### V. Dodanie progresywnego zwiększania poziomu trudności.

Dodajmy zmienną lokalną „Zyciebalona” – która np. co dziesiąty balon będzie zwiększana, spowoduje to, że 11 balon będzie trzeba już trafić 6 razy żeby go zniszczyć. Dodajmy również bloczek „zwiększ rozmiar o 1” przez co kolejna fala „balonów” będzie rosła. Można dodać jeszcze zmianę kolorów.



Możemy także sprawić, żeby nasze balony poruszały się coraz szybciej – wprowadzamy zmienną lokalną „predkosc balona” (lokalna na wypadek jakbyśmy chcieli dodać kolejne balony).

Prędkość balona oddziałuje na czas przelotu każdego odcinka trasy przez balon (na początku jest równa 1, dany czas przelotu należy pomnożyć przez prędkość) po 10 balonach następuje zmniejszenie predkoscibalona o 5% (każda fala będzie przyspieszona o 5% w stosunku do poprzedniej).



Po przez modyfikacje wartości o jakie zmieniają się „Zyciebalona” i „predkoscbalona” optymalizujemy grę.

### Zadania dodatkowe do samodzielnej realizacji

Dodanie warunku przegranej, zmiana kostiumu i zatrzymanie skryptów poszczególnych duszków

Dodanie działek

Dodanie dodatkowych przeciwników



## Temat 8. „Lot balonem”

### Proponowany czas realizacji

3x45min

### Krótki opis projektu

Balon znajduje się w jaskini. Po naciśnięciu klawisza spacja zaczyna spadać. Klikając w niego powodujemy, że zaczyna się wznosić. Po chwili znowu zaczyna opadać. Musimy tak nim manewrować, aby nie dotknąć ścian jaskini. Po dotknięciu koloru brązowego balon się rozbija. Dodatkowo balon cały czas leci w przód. Mechanizm przesuwającego się tła powstał na tej samej zasadzie jak w grze „Gwiezdna potyczka”. Jednak tym razem nasze duszki tła rysujemy samodzielnie. Po rozbiciu będziemy informowani jak długo udało nam utrzymać się w powietrzu.



### Przebieg lekcji

1. Przedstawienie gotowej gry „Lot balonem”
2. Dyskusja nad możliwymi rozwiązaniami programistycznymi wykorzystanymi w czasie realizacji projektu
3. Wykonanie dwóch duszków jaskini oraz napisanie ich programów
4. Narysowanie balonu
5. Stworzenie zmiennej Lot oraz Czas
6. Wykonanie skryptu sterującego balonem
7. Samodzielna praca nad dalszymi modyfikacjami gry
8. Podsumowanie zajęć





## Cele lekcji

Uczeń wie:

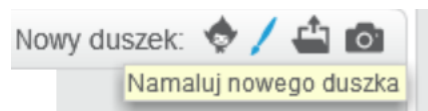
- czym jest pętla i jak można ją wykorzystać w programowaniu,
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

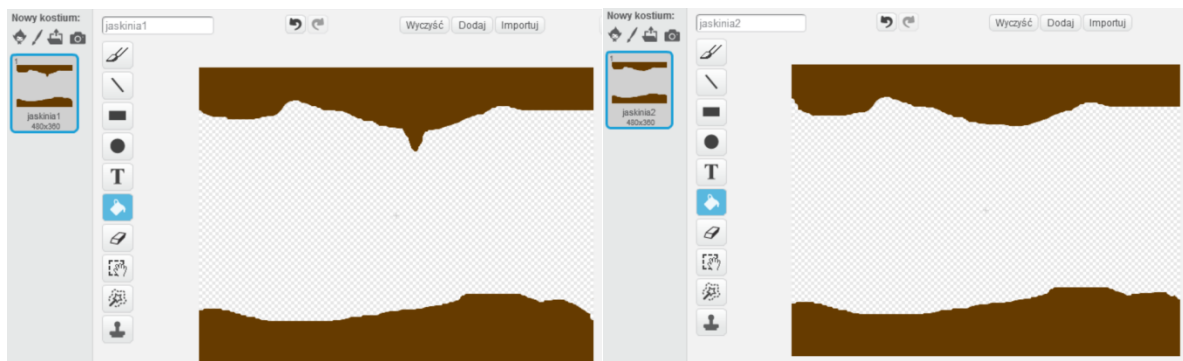
- korzystać z różnych rodzajów pętli,
- wykorzystywać system nadawania i odbierania komunikatów do porozumiewania się duszków pomiędzy sobą,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej.

## Instrukcja wykonania gry dla nauczyciela

- I. Tło do naszej gry będą stanowiły dwa duszki, które na przemian będą przesuwać się po ekranie. Jako tło podstawowe możemy wybrać „blue sky3”. Posiada jednolity kolor.
- II. Duszki rysujemy przy pomocy opcji „Namaluj nowego duszka”.



Będą one zajmowały całą dostępną przestrzeń. Nazwijmy je „jaskinia1” oraz „jaskinia2”. Środek jaskini powinien pozostać pusty. Krawędzie jaskini z lewej i prawej strony na obydwu duszkach powinny mieć taką samą wysokość (będą one nachodzić na siebie).



- III. Oba duszki będą poruszały się synchronicznie przez planszę. Duszek „jaskinia1” zaczyna się poruszać od współrzędnych (0,0) do współrzędnych (-470,0), natomiast duszek „jaskinia2” od współrzędnych (470,0) również do współrzędnych (-470,0), ale w różnym czasie.

Duszki będą się przemieszczać na zmianę, w odstępach 3 sekundowych przez planszę. W tym celu używamy pętli „zawsze” oraz dwóch bloczków ruchu. Czas przemieszczania się tła można modyfikować. Obydwa czasy przelotu powinny być takie same. Czas pierwszego przelotu duszka „jaskinia1” powinien stanowić połowę czasu całego przelotu.

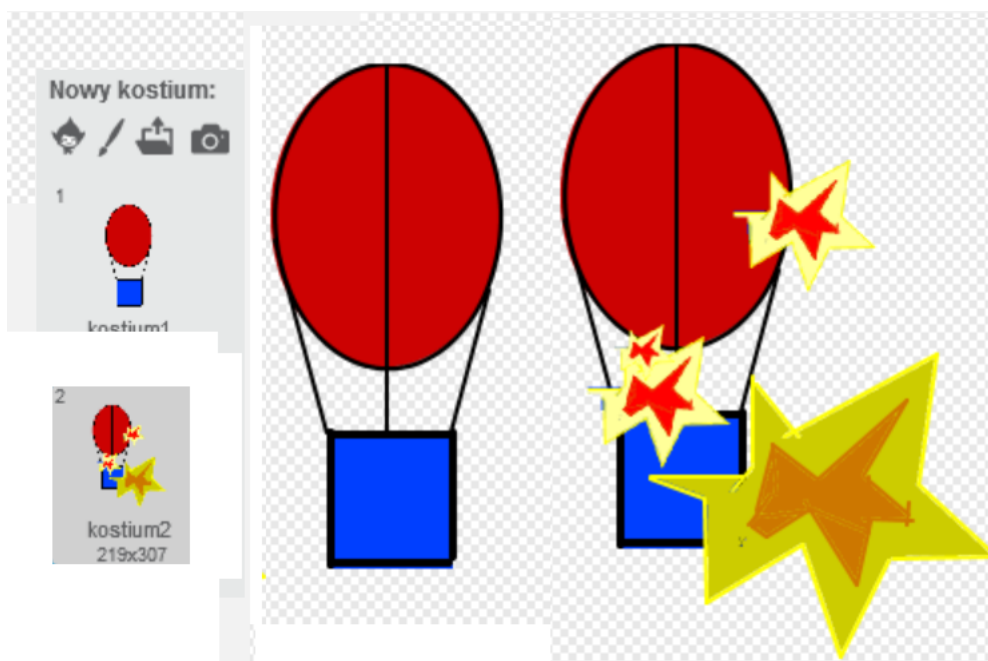
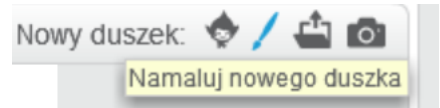


Skrypt odpowiedzialny za poruszanie się duszków będzie rozpoczynał się po otrzymaniu komunikatu „Start”.

Po otrzymaniu komunikatu „GameOver” oba duszki się zatrzymują (bloczek „zatrzymaj inne skrypty duszka”).



- IV. Obiektem który będzie poruszał się w jaskini jest balon. Dodajemy nowego duszka przy użyciu opcji „Namaluj nowego duszka”. Przy użyciu dostępnych opcji rysujemy duszka. Następnie duplikujemy go i wykonujemy drugi kostium rozbitego palona, po przez dodanie iskier.



- V. Przechodzimy do zakładki Dane i tworzymy dwie zmienne „Lot” oraz „Czas”
- VI. Gra nie może zaczynać się od razu po kliknięciu Zielonej Flagi. Pierwszy program balona będzie odpowiada za wprowadzenie nas w tematykę gry. Dopiero, kiedy będziemy gotowi gra się rozpocznie.

Program będzie oczekiwał na naciśnięcie klawisza spacja, kiedy to się stanie czyszczone jest pole wypowiedz (bloczek „powiedz ...” pozostawiamy pusty) oraz nadawany jest komunikat „Start”.



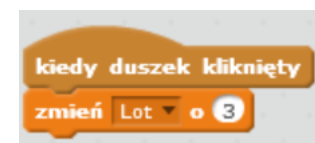
Lot balona będzie zależał od wartości zmiennej „Lot”. Jeżeli będzie ona dodatnia balon będzie poruszał się w górę, jeżeli będzie ujemna balon będzie przemieszczał się w dół. W czasie gry położenie wzdłuż osi x pozostaje stałe. Lot balonu rozpocznie się po otrzymaniu komunikatu „Start”.



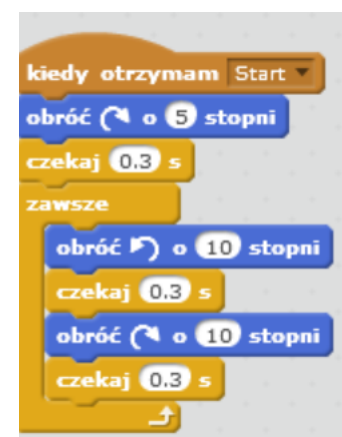
Wartość zmiennej „Lot” po otrzymaniu komunikatu „Start” zostaje ustawiona na 0. Jej wartość będzie zmieniana o -1 co 0,1 sekundy (Balon będzie spadał).



Jeżeli klikniemy w duszka wartość zmiennej „Lot” zwiększy się o 3.



W czasie lotu balon będzie wychylał się raz w prawą stronę, raz w lewą stronę. Użyjemy do tego bloczków „obróć o ... stopni” oraz „czekaj” umieszczonych w pętli zawsze.





- VII. Ostatni podprogram balona odpowiedzialny będzie za pomiar czasu lotu oraz za wykrywanie koloru brązowego (czy nie zderzyliśmy się z ścianą jaskini).

Po otrzymaniu komunikatu „Start” stoper jest zerowany. Program będzie oczekiwał na dotknięcie przez duszka koloru ściany jaskini.

Kiedy to się stanie nadawany jest komunikat „GameOver” (ściany jaskini się zatrzymują), zatrzymaniu ulegają inne skrypty tego duszka, zmienna czas ustawiana jest na zaokrągloną wartość z aktualnego odczytu stopera. Na koniec duszek informuje nas o tym jak długo udało nam się utrzymać w powietrzu.



### Zadania dodatkowe do samodzielnej realizacji

Dodanie duszków, które będziemy musieli zbierać w czasie lotu balonem (np. gwiazdek przelatujących z prawej na lewą stronę planszy).

Przyspieszenie duszków jaskini wraz z upływającym czasem.

Dodanie dodatkowych kostiumów dla duszków „jaskinia1” oraz „jaskinia2”.



## Temat 9. „Gra platformowa”

### Proponowany czas realizacji

3x45min

### Krótki opis projektu

Gra platformowa jest grą zręcznościową polegającą na poruszaniu się bohatera po różnych poziomach (najczęściej wielopoziomowych platformach). Uczniowie wykonają grę, w której główny bohater będzie przemieszczał się pomiędzy poszczególnymi poziomami. Jego zadaniem będzie łapanie gwiazdek poruszających się w sposób losowy po planszy. W czasie wykonywania gry uczniowie wykorzystają całą dotychczasową wiedzę. Stworzą duszka który będzie poruszał się ruchem przyspieszonym oraz będzie potrafił skakać.



### Przebieg lekcji

1. Przedstawienie uczniom przykładowej gry platformowej
2. Dyskusja nad możliwymi rozwiązaniami programistycznymi
3. Wykonanie planszy gry
4. Modyfikacje duszka z biblioteki
5. Wykonanie programu umożliwiającego chodzenie
6. Wykonanie programu umożliwiającego skakanie oraz odbijanie się od ścian
7. Wybór duszków, które bohater gry będzie musiał kolekcjonować
8. Samodzielna praca uczniów nad dalszymi modyfikacjami gry
9. Podsumowanie zajęć





## Cele lekcji

Uczeń wie:

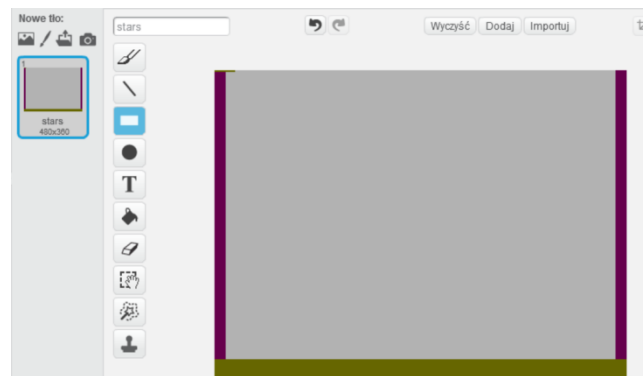
- czym jest pętla i jak można ją wykorzystać w programowaniu,
- jakie rodzaje pętli występują w Scratchu,
- czym jest zmienna.

Uczeń potrafi:

- korzystać z różnych rodzajów pętli,
- za pomocą ciągu poleceń sterować obiektem na ekranie,
- wykonać prostą grę zręcznościową w Scratchu,
- wykonać rysunki i motywy przy użyciu edytora grafiki (posługuje się kształtami, barwami, przekształcaniem obrazu, fragmentami innych obrazów),
- sformułować ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej.

## Instrukcja wykonania gry dla nauczyciela

- I. Zaczynamy od wykonania planszy do naszej gry. Przy pomocy dostępnego edytora obrazu wykonujemy tło składające się z podłogi oraz ścian.



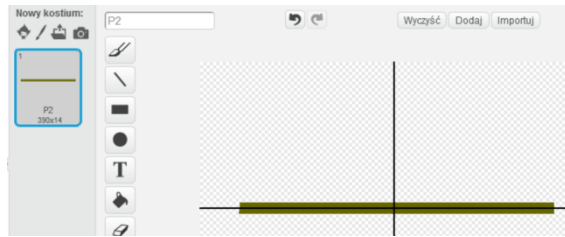
Pozostałe trzy poziomy wykonamy jako duszki. W przypadku modyfikacji gry będzie je łatwiej nam przekształcić. Platformy muszą mieć taki sam kolor jak dolny prostokąt.

Poszczególne platformy powinny mieć szerokość mniejszą od całej szerokości planszy. Kolejne poziomy umieszczamy w równych odległościach od siebie.

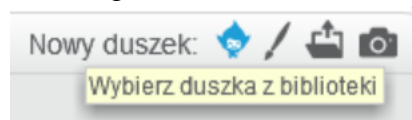


W niektórych platformach możemy pozostawić pustą przestrzeń pośrodku. Ułatwi nam to schodzenie na niższe poziom.

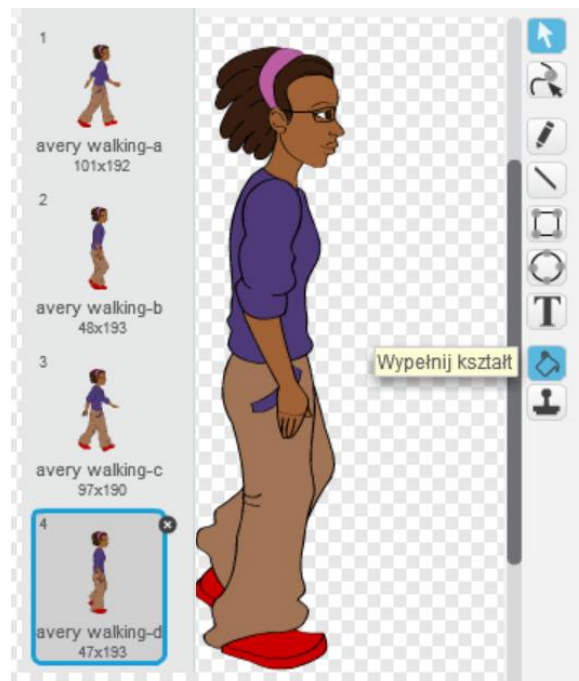




- II. Głównego bohatera gry wybieramy z biblioteki („Avery Walking”).



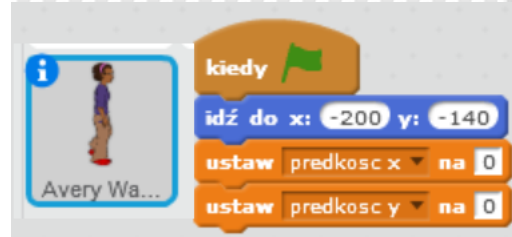
- III. W każdym z kostiumów Avery zmieniamy kolor jej butów na czerwony (dowolny kolor niewystępujący na planszy) przy pomocy narzędzia „Wypełnij kształt”.



- IV. W zakładce Dane tworzymy trzy zmienne „prędkość x”, „prędkość y” oraz „punkty”.

- V. Po naciśnięciu Zielonej Flagi nasz duszek pojawia się w określonych współrzędnych planszy. Zmienne „predkosc x” oraz „predkosc y” zostają ustawione na 0.

Posiada on dwie składowe prędkości x oraz y odpowiedzialne za ruch w poziomie oraz pionie.



- VI. W pętli zawsze umieszczamy dwa bloczki ruchu „zmień x o ...” oraz „zmień y o ...”. Położenie duszka będzie w każdej iteracji zmieniane o aktualną wartość zmiennych „predkosc x” oraz „predkosc y”. W przypadku gdy obie są równe zero, duszek będzie pozostawał w miejscu.



- VII. Duszek będzie poruszał się przy pomocy strzałek. Będzie się on poruszał ruchem przyspieszonym – im dłużej będziemy wciskać strzałkę tym szybciej będzie się poruszał.

Gdy wciskamy strzałkę w prawo wartość prędkości  $x$  zwiększa się o 0,4 natomiast przy naciśnięciu strzałki w lewo zmniejsza się o -0,4.

Po każdym sprawdzeniu tych dwóch warunków duszek będzie „wyhamowywał” - wartość prędkości  $x$  będzie malała, w tym celu zastosowano blok ustaw ... na (za każdym razem prędkość będzie przyjmowała wartość 95% dotychczasowej prędkości).

Ostatni warunek „jeżeli...” odpowiada za odbijanie się od ścian, wartość prędkości  $x$  ulega odwróceniu, mnożymy ją przez -1.



- VIII. Kolejny podprogram Avery będzie odpowiedzialny za zmianę kostiumów w czasie chodzenia. Duszek będzie zmieniał kostium zawsze wtedy gdy zmienna „prędkość  $x$ ” będzie znajdować się w przedziale od -0,4 do 0,4. Dzięki czemu podprogramowi uzyskamy wrażenie ruchu naszego duszka.



- IX. Pozostało jeszcze wykonanie podprogramu odpowiedzialnego za skakanie.

W pętli zawsze umieszczamy instrukcję warunkową „jeżeli ... to ... w przeciwnym razie ...”

Warunkiem, który sprawdzamy jest to czy kolor butów, nie dotyka kolory podłoża oraz czy nie wykonaliśmy skoku (predkosc  $y$  < 0,01).

Jeżeli oba warunki są spełnione wartość prędkości  $y$  zostaje ustawiona na 0, nie będziemy przelatywać przez platformy. Jeżeli program będzie znajdował się w tym miejscu wykonywania skryptu i naciśniemy spację



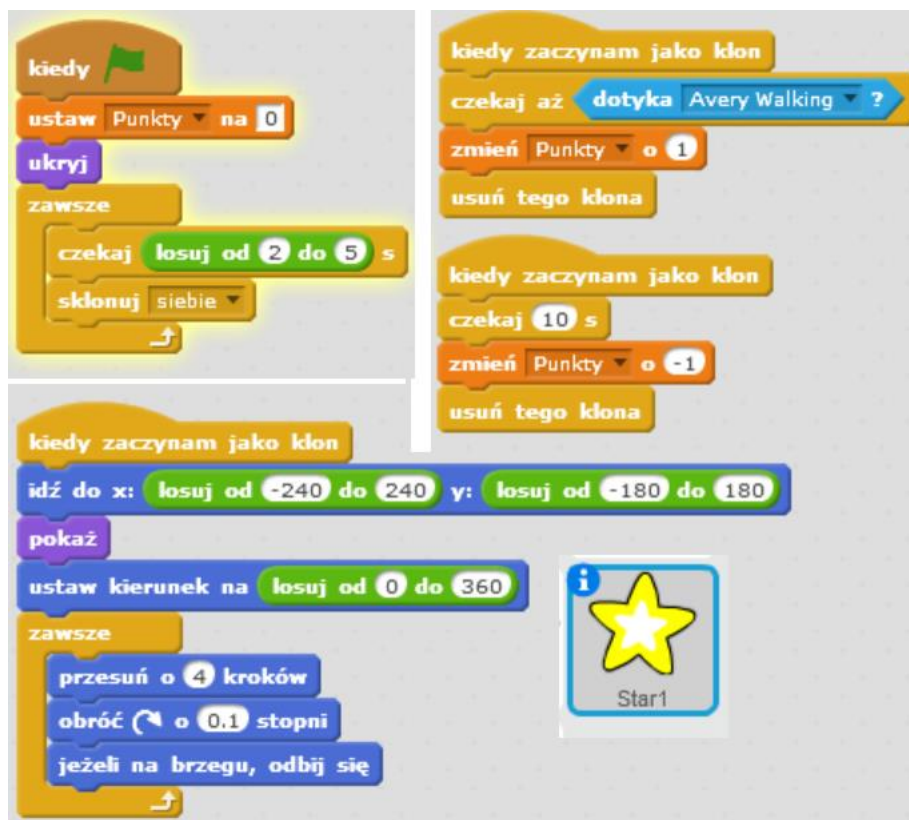


prędkość y zostanie ustawiona na wartość 10 (zaczniemy poruszać się do góry).

Wartość zmiennej „prędkość y” w czasie skoku jest większa niż 0,01 dlatego będzie wykonywany warunek „w przeciwnym razie ...” wartość prędkości y będzie zmniejszana o -0,4 w każdej iteracji pętli zawsze. Gdy wartość prędkości y będzie mniejsza niż 0 zaczniemy poruszać się w dół, tak długo, aż nie dotkniemy podłoża.

- X. Wykonaliśmy właśnie swoją pierwszą grę platformową. Pozostaje teraz samodzielnie pomyśleć nad jej fabułą. Możemy wykorzystać zmienną, którą utworzyliśmy wcześniej („Punkty”).

Z biblioteki wybieramy „Star1”. Duszek będzie klonował się w losowym miejscu na planszy i zacznie się poruszać. Za każdą złapaną gwiazdkę będziemy zyskiwać 1 punkt. Jeżeli nie uda nam się złapać gwiazdki w ciągu 10s będziemy tracić jeden punkt. Poniżej został przedstawiony skrypt gwiazdki. Możemy go dowolnie modyfikować.



### Zadania dodatkowe do samodzielnej realizacji

Stworzenie własnej gry platformowej na podstawie poznanego mechanizmu.