

Lekcja . Gra platformowa

Witam!

Drodzy UCZNIOWIE proszę o zapoznanie się z filmami demonstracyjnymi języka SCRATCH.

Spróbujcie (korzystając z filmu) stworzyć proste programy. Powodzenia

<https://www.youtube.com/watch?v=2fGMt7hk2ng&feature=youtu.be>

<https://www.youtube.com/watch?v=0BYPsODumWY>

<https://www.youtube.com/watch?v=T-l4n4p7g6I>

<https://www.youtube.com/watch?v=YGHPg9M7N2s&feature=youtu.be>

Materiał do realizacji poniżej. Powodzenia

Cele

Celem jest stworzenie prostej gry platformowej umożliwiającej jej szerokie zastosowanie tematyczne.

Cele ogólne

- przypomnienie i utrwalenie poznanych wcześniej poleceń i konstrukcji języka Scratch
- opracowanie projektu będącego bazą gry platformowej
- implementacja projektu w środowisku Scratch

Cele szczegółowe

- uczeń rozumie i potrafi stosować odpowiednie instrukcje strukturalne
- uczeń rozumie pojęcie zmiennej i wykorzystuje je w programie Scratch
- uczeń potrafi konstruować skrypty zawierające zmienną
- uczeń potrafi implementować grę w środowisku Scratch

Informacje

Poziom trudności i proponowane wersje:

- dla klas 4-6 🌟🌟🌟🌟 na przykładzie projektu [Gra platformowa](#)

Etapy realizacji

Wprowadzenie

Czas realizacji 10 minut.

Przedstawienie koncepcji projektu stworzenia bazy gry platformowej. Rozmowa odnosząca się do doświadczeń uczniów, nawiązanie do gier, w które uczniowie grają. Skoncentrowanie się na grach platformowych, w których występuje element poruszania bohaterem. Np.: Lego, Super Mario, Sonic. Przypomnienie realizowanych na dotychczasowych zajęciach, wcześniejszych modułów dotyczących sterowania postaciami.

Tworzenie sceny

Czas realizacji 15 minut

Plansza *intro* i *scena*

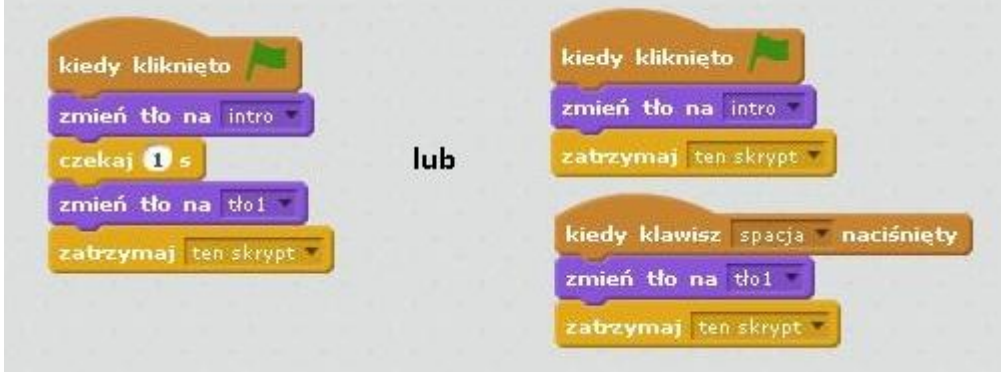
Rozpoczynamy od przygotowania planszy rozpoczynającej grę [intro](#). Wprowadzamy podstawowe informacje np.: tytuł gry. Aby łatwiej w późniejszych etapach konstruować skrypty, nazwijmy nasze tła (np.: *intro* dla tła planszy rozpoczynającej, *tlo1* dla planszy pierwszego poziomu gry)



Skrypt zmiany tła

Na tym etapie możemy wprowadzić prostą, automatyczną zmianę tła lub zmianę tła *intro* na *tlo1* po naciśnięciu odpowiedniego klawisza:

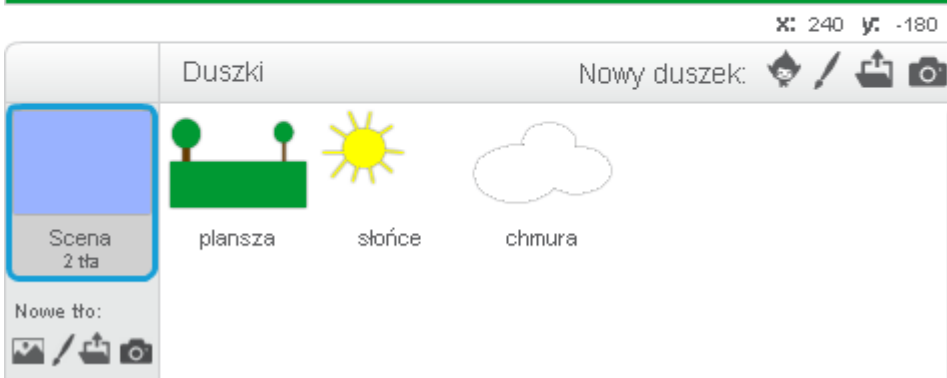
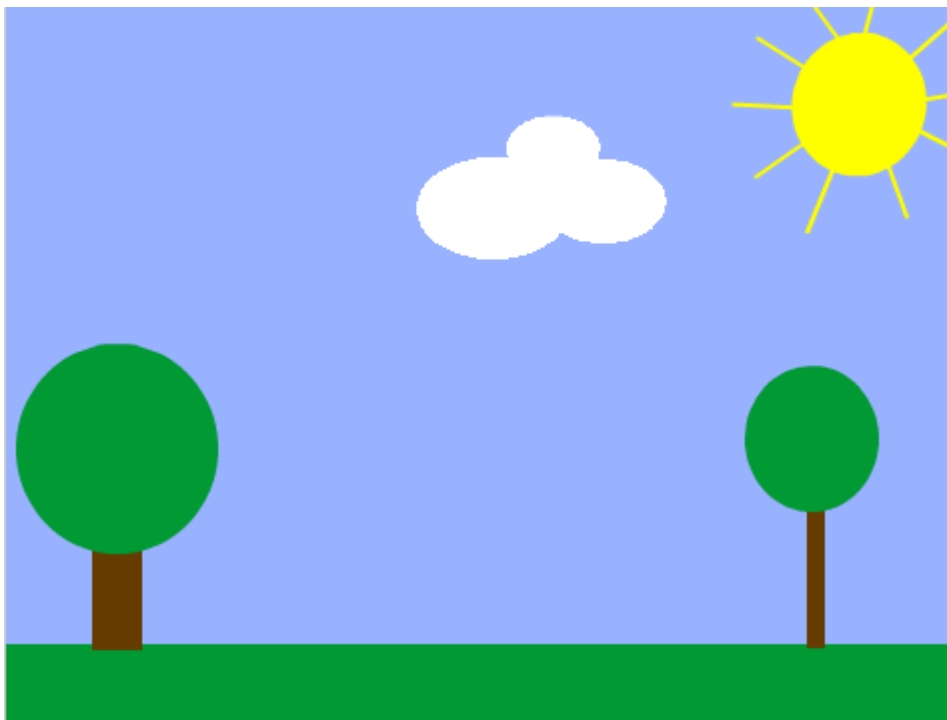
Np.:



Tworzenie duszka *plansza*

Czas realizacji 15 minut W grach platformowych, bohaterowie poruszają się po trasach wyznaczanych przez różne elementy, np.: chodniki, platformy, schody. W prezentowanej wersji przygotowana została zielona *plansza* (trawa i drzewa). Dodatkowo można umieścić duszki stanowiące element krajobrazu, urozmaicenia gry (tu: słońce, chmura).

Uwaga Korzystniejszym rozwiązaniem jest stworzenie prostej planszy i umieszczanie wszystkich elementów oddzielnie (drzewo, jako duszek)



Na tym etapie możemy opisać podstawowy skrypt duszka *plansza* umożliwiający jego ukrycie podczas wyświetlania planszy *intro* oraz pojawienie się w określonym punkcie ekranu, po zmianie tła na *tło1*.

Postać głównego Bohatera

Grafika postaci

Czas realizacji 15 minut.

Uwaga! Czas realizacji tej części uzależniony jest od wybranej opcji: wybór postaci lub tworzenie postaci. Umożliwia to zarówno dostosowanie modułu do potrzeb i możliwości uczniów oraz do czasu trwania zajęć kodowania (bloki zajęć trwające 45 lub 90minut). W przypadku, gdy uczniowie chcą zaprojektować

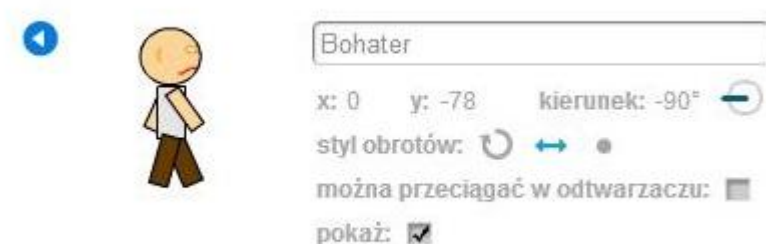
własne kostiumy duszków można zaproponować tę część jako zadanie domowe do samodzielnej realizacji. Proponowanym rozwiązaniem jest samodzielne przygotowanie przez uczniów podstawowej, prostej wersji postaci, z możliwością dopracowania graficznej formy w domu.

Tworzymy prostą postać wraz z kostiumami ruchu:

- *stoi*
- *bieg_lewa* gdy postać wykonuje krok jedną nogą
- *bieg_prawa* gdy postać wykonuje krok drugą nogą



W ustawieniach [styl obrotu](#) duszka zaznaczamy styl obrotu lewa-prawa (a nie 360 stopni)



Sterowanie bohaterem - ruch lewa-prawa.

Czas realizacji: 20 minut

Chcemy aby nasza postać poruszała się zmieniając kostium, imitując naturalny, naprzemienny krok. Mając przygotowane kostiumy postaci, ustawionej w prawą stronę, programujemy ruch w prawo, dla którego najprostszy skrypt może wyglądać tak:



Aby nasza postać poruszała się takim krokiem sterowana klawiszem *strzałka w prawo* dodajemy warunek:



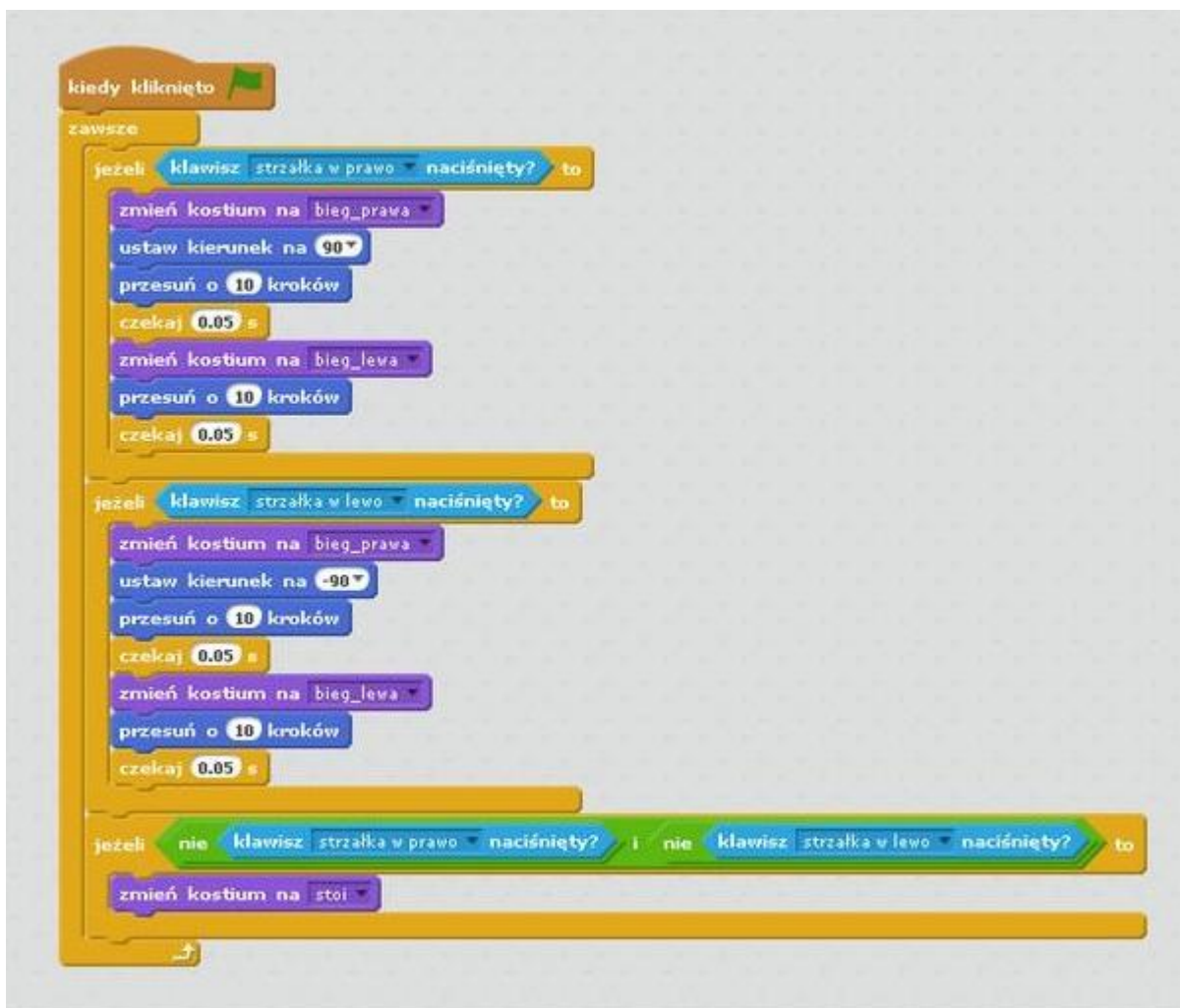
Analogicznie uczniowie mogą tworzyć skrypt dla ruchu w lewo, sterowanego klawiszem strzałka w lewo.

Dyskutujemy o tym, w jaki sposób kontrolować poruszanie postacią tak, aby uniknąć dwóch problemów:

1. nie widać animacji
2. po puszczeniu klawisza strzałki Bohater zatrzymuje się, ale nie wraca do kostiumu "stoi".

Proponujemy przejście klocków z grupy kontrola oraz czujniki. Dyskutujemy wspólnie na temat możliwości utworzenia jednego skryptu, kontrolującego ruch postaci sterowanej klawiszami strzałek oraz uwzględniającego pozycję braku ruchu z kostiumem "stoi"

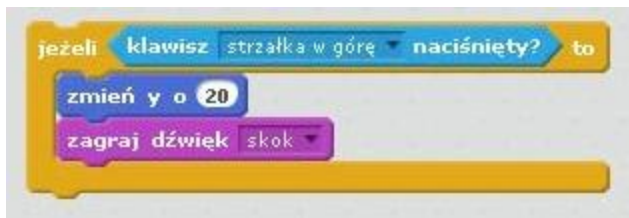
Na tym etapie skrypt sterujący ruchem lewo-prawo może wyglądać w następujący sposób:



Sterowanie bohaterem - skok.

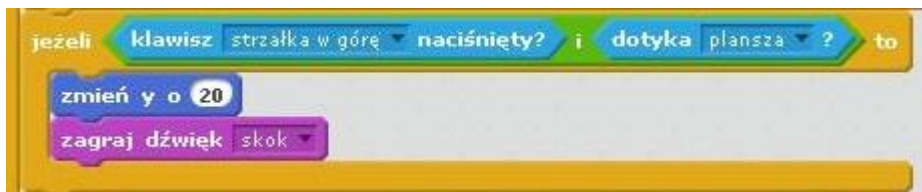
Czas realizacji: 15 minut

Skrypt postaci można rozwinąć o algorytm umożliwiający postaci wybiecie do góry, sterowane klawiszem "strzałka w górę". Analogicznie do tworzonych wcześniej skryptów, można stworzyć następujący schemat (wzbogacony o efekt dźwiękowy):



Uwaga! Warto w tym momencie przetestować poruszanie się postaci. Tak skonstruowany skrypt powoduje, że postać może wykonać skok, odbijając się nawet "od niczego". Warto wskazać, że w sytuacji rzeczywistej możemy skakać w górę wyłącznie poprzez wybicie się od podłoża. Należy dodać warunek możliwości wykonania skoku jeżeli postać dotyka duszka plansza'

Z bloku **Wyrażenia** wybieramy klocek warunku **i**  łącząc nim elementy Bloku **Czujniki**, klocek **strzałka w górę naciśnięta** oraz **dotyka plansza**.



Tak skonstruowany skrypt umożliwia bohaterowi małe podskoki. Warto dostosować siłę skoku (zmiana wartości Y) np.: do kształtu planszy, Można również wprowadzić element *power-up* (Patrz: ...)

Sterowanie bohaterem - pozycja *chowa się*

Czas realizacji: 10 minut

Skrypt można rozwijać o dodatkowe zachowania postaci wprowadzając dodatkowe, dopasowane do sytuacji kostiumy, np.: wciśnięcie klawisza strzałka w dół powoduje "unik" i schowanie lub schylenie się postaci:



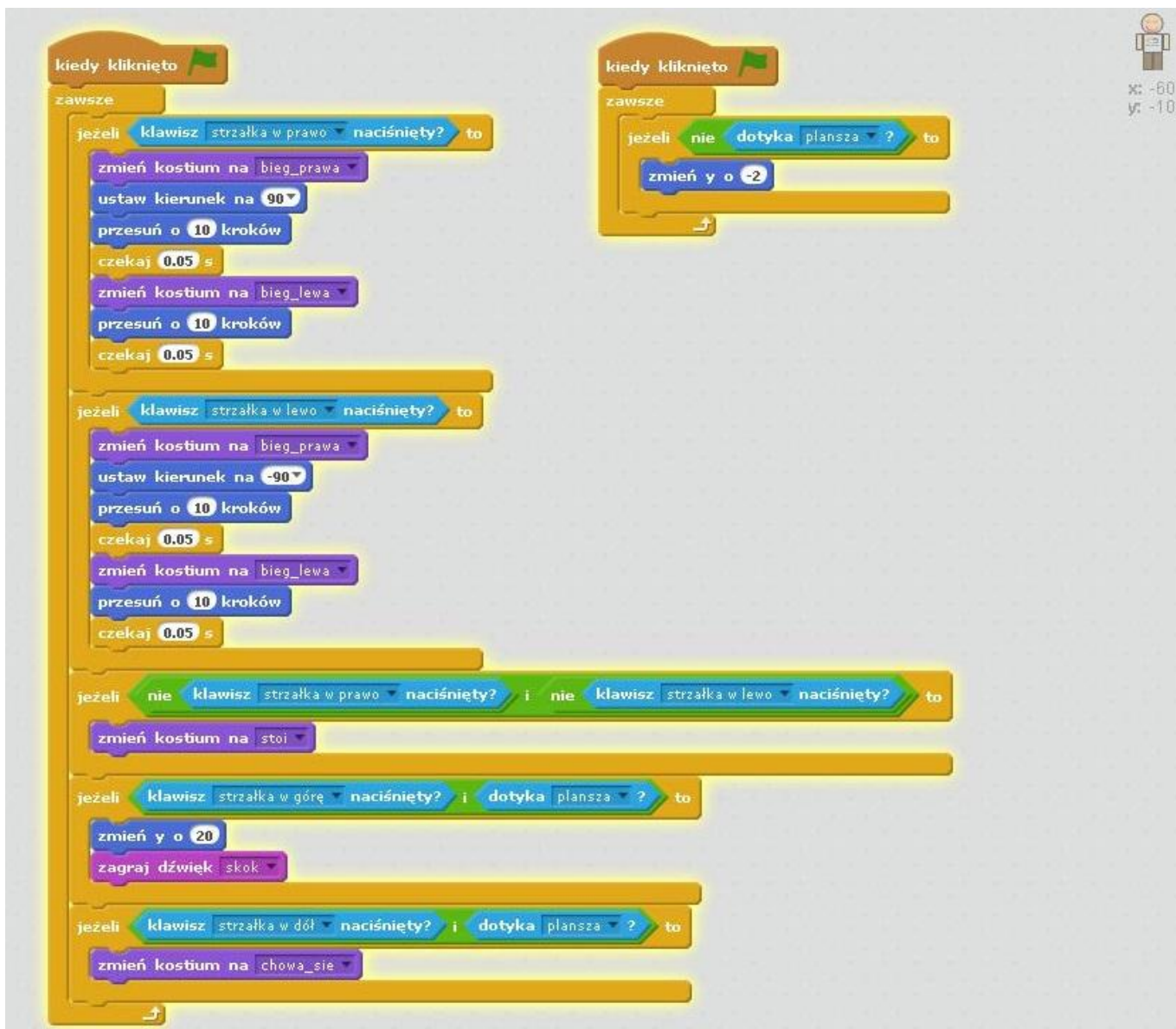
Sterowanie bohaterem - spadanie

Co się dzieje gdy postać poruszającą się po planszy, dotrze na koniec jakiegoś elementu i wyjdzie "poza"? Poszukując odpowiedzi na to pytanie docieramy do hasła "spadanie". Skrypt bohatera można rozwinąć o prosty algorytm umożliwiający postaci uproszczony ruch *spadania*. Jeśli chcemy by po rozpoczęciu gry, bohater zawsze spadał gdy nie ma podłoża pod nogami, wybieramy z bloku **Kontrola** klocek **zawsze** oraz **jeżeli** i łączymy je z wyrażeniem *nie* oraz elementem Bloku **Czujniki** **dotyka plansza** uzyskując następujący zapis:



Skrypt głównej postaci

Proponowany skrypt poruszania się głównej postaci na tym etapie wygląda następująco:



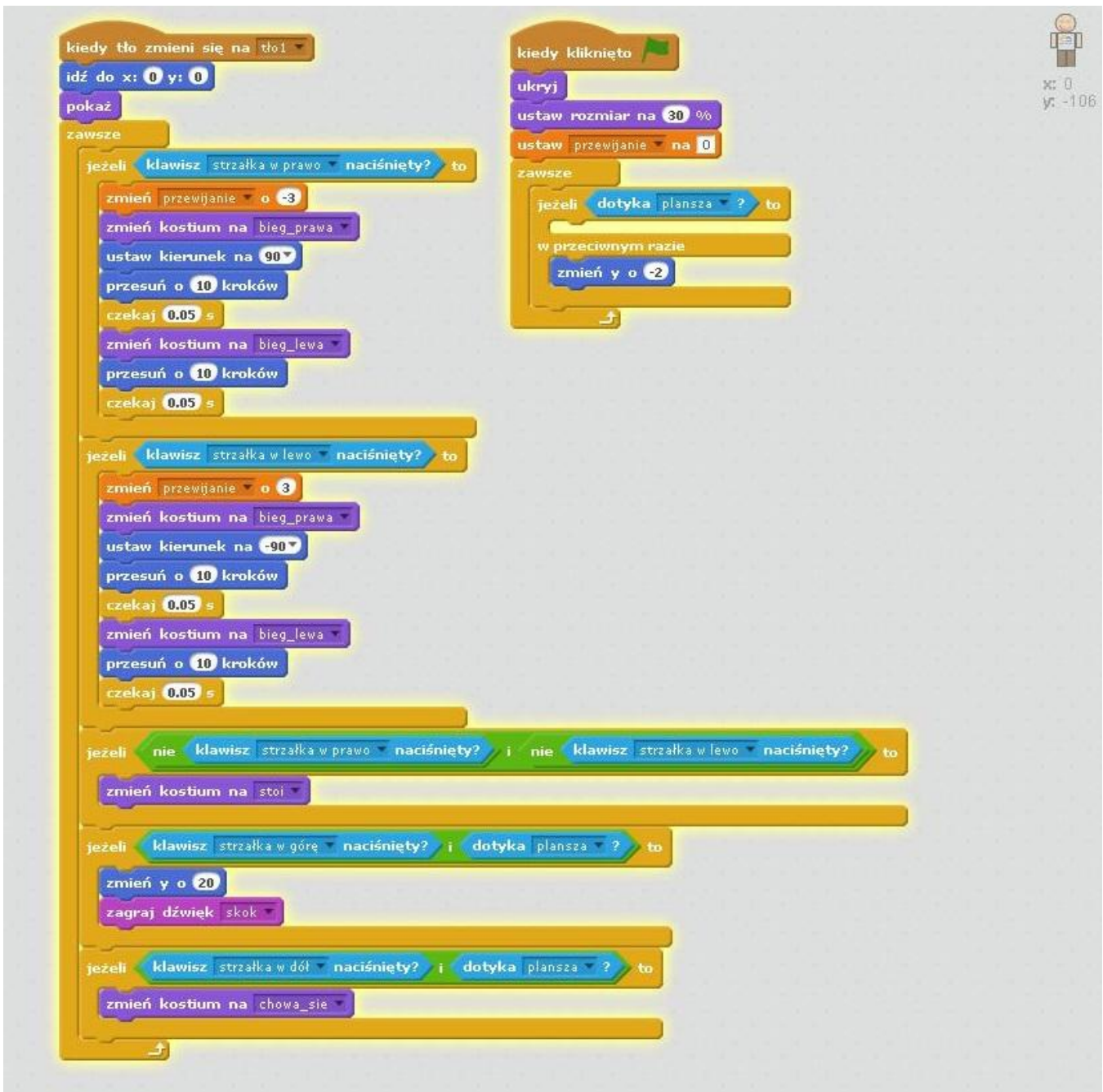
Efekty dodatkowe

Efekt przewijania

Tak skonstruowana baza gry platformowej umożliwi poruszanie się postacią Bohatera w obrębie widocznego na ekranie tła, po *planszy*. Jednym z charakterystycznych elementów gier platformowych jest efekt przewijania, czyli przesuwania się duszka *plansza* stosownie do poruszania się postaci Bohatera. Efekt przewijania można zaimplementować poprzez uzależnienie położenia duszka *plansza* od pozycji poruszającego się Bohatera.

Skrypt Bohatera zmodyfikowany o efekt *przewijanie*

W bloku *Dane* wybieramy opcję *Stwórz zmienną*, nadajemy jej nazwę np.: *przewijanie*. W chwili rozpoczęcia gry, wartość *przewijanie* równa jest 0, natomiast zmienia się w chwili poruszania bohatera, dlatego do skryptu postaci Bohatera dokładamy klocki z bloku *Dane* *ustaw przewijanie na 0* oraz *zmień przewijanie o...*. Zmianę danych *przewijanie* ustalamy dostosowując do potrzeb gry, w prezentowanym projekcie wartość ta wynosi [3]. Przy czym gdy Bohater porusza się w prawo, wartość równa się -3, a gdy Bohater porusza się w lewo +3.



Skrypt duszka *plansza* zmodyfikowany o efekt *przewijanie*

Równocześnie zmianie ulega skrypt duszka *plansza* ponieważ ruch tego elementu ma być uzależniony od zmiany pozycji Bohatera:



Power Up!

W wielu grach platformowych Bohater może złapać obiekt, który sprawia, że na pewien czas jego moc rośnie. Odwołujemy się do doświadczeń uczniów, możemy również posłużyć się przykładem Super Mario, który zjadając grzyba Mega Mushroom powiększa się i nabiera mocy ([klip](#)). Ten grzyb stanowi w grze Mario Bros *power up* (ang. zwiększenie mocy). Dyskutujemy o pomysłach stworzenia duszka reprezentującego ulepszenie postaci Bohatera (tzw "upgrade").

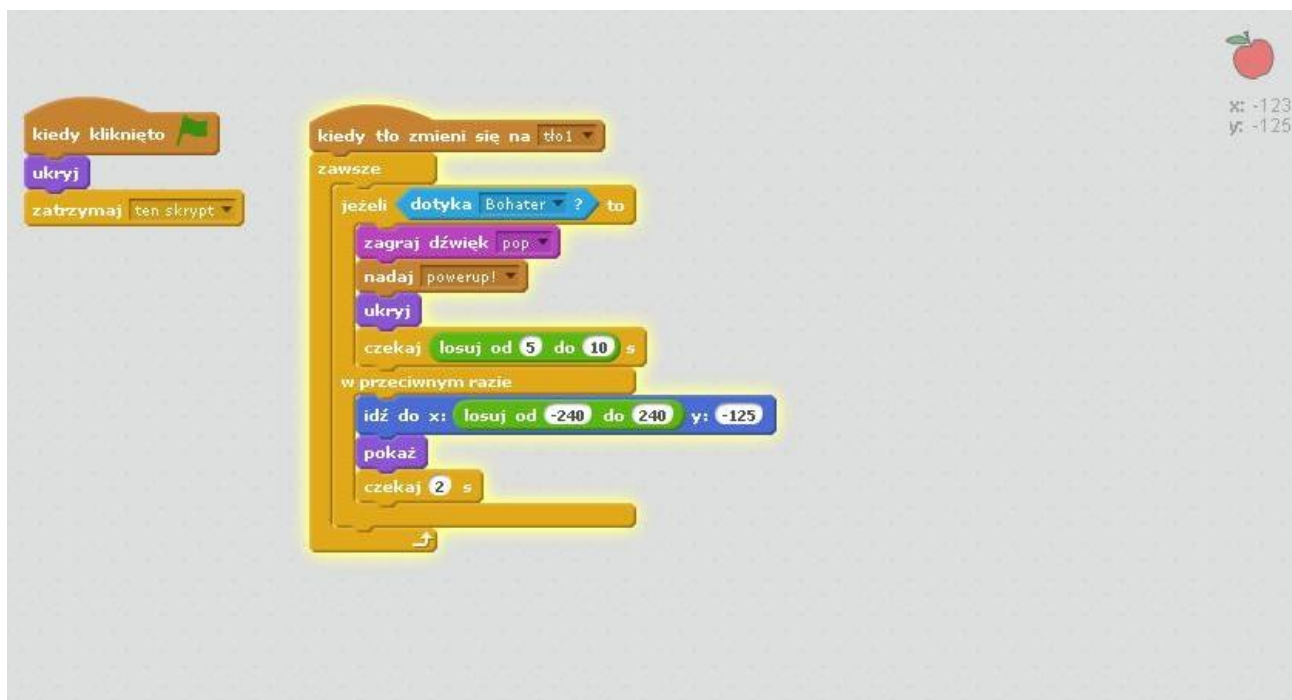
- Czy chcemy aby na planszy był widoczny jeden czy wiele powerup?
- Czy ma mieć stałe czy zmienne położenie?
- Czy power-upy mają pojawiać się w określonych miejscach czy wybranych losowo?
- Co ma powodować zetknięcie Bohatera z elementem "power up"?
- Jaką formę ma mieć power-up? (można wykorzystać formę uniwersalną np.: Super-jabłko lub zaproponować formę nawiązującą do omawianych zagadnień szkolnych np.: litera "ó", liczby pierwsze, figury geometryczne itp)

Uwaga! Jeśli uczniowie planują stworzenie kilku elementów, zachęcamy do rozważenia rozwiązania "klonuj". W scenariuszu prześledzimy tworzenie jednego power-upa, ale pojawiającego się na ekranie w losowym miejscu.

Skrypt elementu *powerup*- duszek *Super jabłko*

Czas realizacji 25 minut.

Tworzymy grafikę duszka lub dodajemy z galerii programu Scratch. Power-up *Super jabłko* będzie pojawiał się w różnych miejscach i po chwili znikał. Gdy nasz bohater dotknie „Super jabłka”, jego postać powiększy się.



Skrypt Bohatera zmodyfikowany o efekt *power up!*

Wykorzystanie w skrypcie powerupa klocka "nadaj wiadomość" skutkuje zmianami w skrypcie głównego Bohatera. Proste powiększenie postaci można kodować w następujący sposób:



Skrypt warto wzbogacić o efekt dźwiękowy. Uczniowie mogą skorzystać z galerii dźwięków dostępnych w programie Scratch lub mogą użyć dźwięków dostępnych w bibliotece np.: freesound.org (dostęp bezpłatny, po rejestracji).