



Batyskaf

Cel lekcji

Na lekcji stworzymy grę w Scratchu, w której steruje się Batyskafem, i strzela pociskiem do podwodnych bestii w celu zdobycia punktów. Gra będzie wykorzystywała mechanizmy losowego pojawiania się przeciwników oraz poznamy mechanizm klonowania dysków.

Wytyczne dla lekcji:

Grupowa wiekowa: 10-12

Czas trwania lekcji: 90 minut (2 x 45 min) + przerwa 5 minut, mniej więcej w połowie przebiegu lekcji

Środowisko, narzędzia używane podczas zajęć: Scratch

Instrukcja przygotowania środowiska: [LINK](#)

Ogólny zestaw instrukcji dotyczący prowadzenia zajęć: [LINK](#)

Formy pracy na lekcji:

- **Metoda "krok po kroku"** – nauka poprzez ściśle podążanie za kolejnymi etapami prezentowanymi przez trenera.
- **Inna np. praca samodzielna** - jeśli zostało to wskazane w konspekcie tej lekcji.

Rekomendacje:

- Przećwicz przebieg lekcji zgodnie z konspektem. Pamiętaj żeby lekcję przerobić przed zajęciami z grupą.
- Dostosuj tempo do grupy – ważne jest zrozumienie każdego elementu.
- Aktywizuj grupę - w konspekcie znajdziesz proponowane pytania, na różnych etapach konspektu, które pozwolą Ci zaktywizować grupę.
- Pamiętaj, aby chwalić uczestników za wykonaną pracę czy to samodzielną czy tę grupową, jeśli osiągną zamierzony efekt.
- Przygotuj dodatkowe materiały, jeśli uważasz, że twoi uczniowie mogą potrzebować dodatkowych wyjaśnień i przykładów.

Spis treści

1. Wstęp (5-10 min)

- Rozmowa na luźne tematy
- Przypomnienie zagadnień z poprzedniej lekcji
- Przedstawienie celu lekcji i efektów uczenia się

2. Część właściwa lekcji (70-80 min)

- #Ustawienia początkowe oraz tworzenie sterowania batyskafem
- #Mechanizm strzału - klonowanie
- #Dodanie przeciwników
- #Warunki zwycięstwa oraz przegranej - komunikaty
- Weryfikacja celu lekcji i efektów uczenia się
- Zadanie dodatkowe (opcjonalne)

3. Zakończenie lekcji (5-10 min)

- Podsumowanie lekcji
- Zapowiedź kolejnej lekcji
- Pytania od uczestników

4. Przegląd skryptów z lekcji

- oznaczenie dla kamieni milowych, składników lekcji które wymagane są do pełnej realizacji konspektu lekcji.

Kiedy uznajemy, że dany kamień milowy został zrealizowany?

Kamienie milowe w konspekcie zajęć dotyczą kluczowych etapów realizacji lekcji, a ich ukończenie oznacza że trener zrealizował wskazaną partię materiału, przedstawił zagadnienie, wykonał dany krok w budowaniu projektu.

1. Wstęp (5-10 min)

Rozmowa na luźne tematy - przed rozpoczęciem zajęć, lub zaraz po rozpoczęciu **koniecznie** wykorzystaj czas 1-2 minuty na krótką rozmowę nie związaną z lekcją. Postaraj się nawiązać kontakt z grupą. Zadaj pytania np. *Co u Was słychać?, W co ostatnio graliście fajnego? Jaki film polecacie oglądać? itp.*

W przypadku jeśli już masz zbudowaną relację z grupą pamiętaj o systematycznym dbaniu o nią poprzez krótkie rozmowy na różne tematy, które Was interesują. **Bardzo ważne jest utrzymanie ciągłości relacji z grupą.**

Przeczytaj koniecznie rozdział pt. Budowanie relacji z grupą: [LINK](#)

Przypomnienie zagadnień z poprzednich lekcji

1. Jaką grę robiliśmy na poprzedniej lekcji?

Stworzyliśmy własną wersję gry pacman. W której zaprogramowaliśmy poruszanie się naszej postaci po labiryncie. Dodatkowo dodaliśmy przeciwników, którzy poruszają się po labiryncie. Na planszy rozmieściliśmy kropki, które musimy zbierać.

2. W jakim celu postać pacmana oraz przeciwników miała dodane kolorowe kwadraty?

Dzięki tym kwadratом postacie poruszały się po ścieżce w labiryncie. Stworzyliśmy specjalny warunek że ruch jest tylko wtedy kiedy odpowiednie kolory się dotykają w momencie dotknięcia ścian zmieniał się wykrywany kolor i postać albo się zatrzymywała albo zmieniała kierunek ruchu w przypadku przeciwników.

3. Jakie poznaliśmy efekty, które możemy zastosować na duszku? Jak działa efekt duch? Czy efekty trwale zmieniają wygląd duszka?

Kolor - zmiana koloru Duszka

Rybie oko - zniekształca grafikę Duszka w kolisty sposób

Wir - zniekształca grafikę Duszka w wirujący sposób

Zniekształć - pikselizuje grafikę Duszka

Mozaika - powiela obraz w formie mozaiki

Jasność - zwiększa jasność grafiki Duszka.

Duch - zwiększa przezroczystość grafiki Duszka.

Nałożone efekty nie zmieniają w sposób trwały wyglądu duszka, resetują się wraz z ponownym uruchomieniem gry lub po wykorzystaniu bloku Wyczyść efekty graficzne

Przedstawienie celu lekcji i efektów uczenia się

Cel lekcji i efekty uczenia się - przedstawiamy uczestnikom jakim tematem na dzisiejszej lekcji będziemy się zajmować oraz czego się nauczymy.

Na dzisiejszej lekcji stworzymy grę w Scratchu, w której steruje się Batyskafem, i strzela pociskiem do podwodnych bestii w celu zdobycia punktów. Gra będzie wykorzystywała mechanizmy losowego pojawiania się przeciwników oraz poznamy mechanizm klonowania dysków.

W ramach tej lekcji:

- poznacie technikę klonowania duszków
- poznamy technikę komunikacji się pomiędzy duszkami z wykorzystaniem komunikatów
- stworzymy pełny zakres sterowania góra-dół, lewo-prawo

Prezentacja gotowego projektu: [Link](#)

(Prezentujemy gotowy projekt grupie, chwilę grając i tłumacząc zasady rozgrywki funkcjonalności projektu. Parafrazujemy poniższy opis gry.)

Opis gry

Dzisiejsza gra przeniesie gracza do podwodnego świata gdzie kierując batyskafem będziemy musieli pokonać stado rekinów oraz stoczyć walkę z bossem (część opcjonalna wykonywana w zależności od pozostałego czasu). W grze wykorzystamy po raz pierwszy mechanizm klonowania oraz nadawania komunikatów.

Dodatkowe pytania dotyczące projektu:

1. Co to jest batyskaf i jakie macie z nim skojarzenia?

Batyskaf to podwodny statek badawczy, który umożliwia eksplorację głębin oceanu. Kojarzy mi się z odkrywaniem tajemnic podwodnego świata i futurystycznymi technologiami.

2. Jakie elementy projektu widzicie na demonstracji??

Na demonstracji widzę batyskaf jako główną postać, przeciwników (rekiny i bossa), zmienne takie jak punkty i HP, mechanizm klonowania duszków wykorzystywany do strzałów oraz bloki komunikatów, które sterują zdarzeniami w grze.




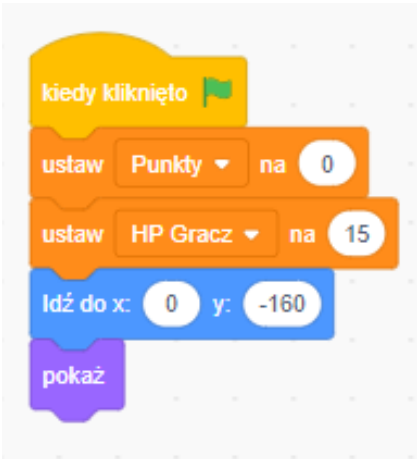

2. Część właściwa lekcji (70-80 min)

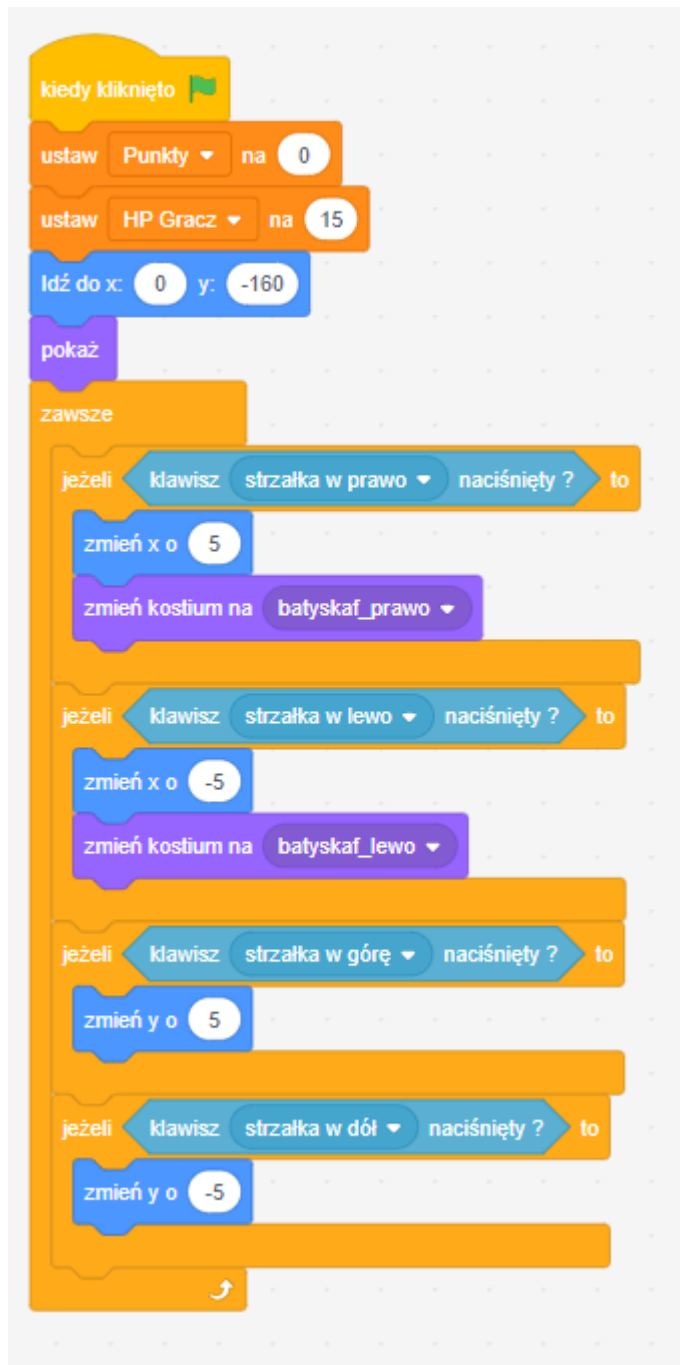
Link do projektu starter: [Link do remiksu](#) lub [Link do pliku](#)

(Udostępniamy projekt starter, szablon wszystkim uczestnikom.

Sposoby udostępniania projektu zostały szerzej opisane w dokumencie **Zestaw instrukcji dotyczących prowadzenia zajęć**, który jest dostępny z poziomu listy obecności oraz sekcji **Nauczyciele** w systemie)

Upewnij się, że wszyscy uczestnicy mają udostępnione swoje ekrany.

 Batyskaf	#Ustawienia początkowe oraz tworzenie sterowania batyskafem Dodamy ustawienia początkowe dla naszego batyskafu ustawimy wartości zmiennych Punkty , oraz HP gracza , pozycję startową oraz pokażemy statek.
	
 Batyskaf	<p>Programujemy sterowanie postaci, będzie ono się różnić znacząco od tego, które tworzyliśmy w grze Pacman. Tworzymy z uczestnikami sterowanie Lewo-Prawo. Sterowanie góra dół staramy się wykonać poprzez pracę samodzielną uczestników.</p> <p>Praca samodzielna Dopasuj stopień pracy samodzielnej do swojej grupy. Możesz też skorzystać z techniki rozsypanki bloków aby ułatwić pracę uczestnikom. Ważne, upewnij się czy uczestnicy wiedzą co jest do zrobienia.</p>



Co możemy jeszcze poprawić w naszym sterowaniu?

Dodać ograniczenie żeby batyskaf nie mógł wypłynąć poza scenę.

Dodajemy na samym końcu skryptu nowy blok, będzie to blokada poruszania się batyskafu poza scenę

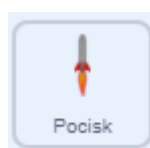
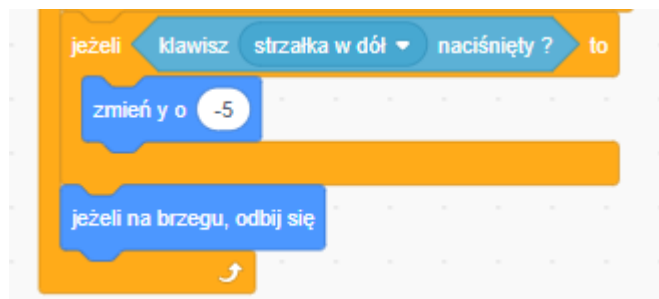
Testujemy działanie pełnego sterowania, oraz zmianę kostiumu, pytamy uczestników co ewentualnie możemy jeszcze poprawić w naszym sterowaniu.

Co możemy jeszcze poprawić w naszym sterowaniu?

Dodać ograniczenie żeby batyskaf nie mógł wypłynąć poza scenę.



Dodajemy na samym końcu skryptu nowy blok, będzie to blokada poruszania się batyskafu poza scenę. Od razu sprawdzamy efekt działania.



#Mechanizm strzału - klonowanie

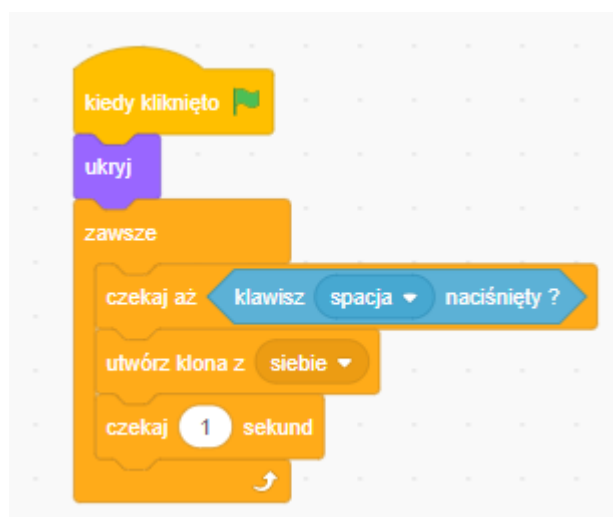
Tworząc strzał wykorzystamy technikę klonowania duszków, dzięki niej będziemy w jednej chwili mogli mieć wiele kopii duszka pocisk na ekranie gry.

Na początku oryginalnego duszka ukrywamy, w grze zobaczymy **tylko jego klony**.

Czekamy aż w grze naciśniemy spację.

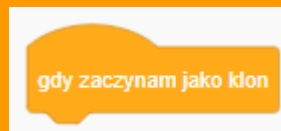
Następnie korzystamy z bloku utwórz klona...który utworzy kopię duszka na wzór oryginału.

Blok czekaj tak naprawdę steruje częstotliwością pojawiania się klonów jeśli chcemy strzelać serią to musimy ustawić mniejszy czas.



Programujemy działanie klonów

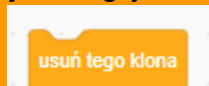
Każdy klon musi wiedzieć jak ma się zachowywać w naszej grze



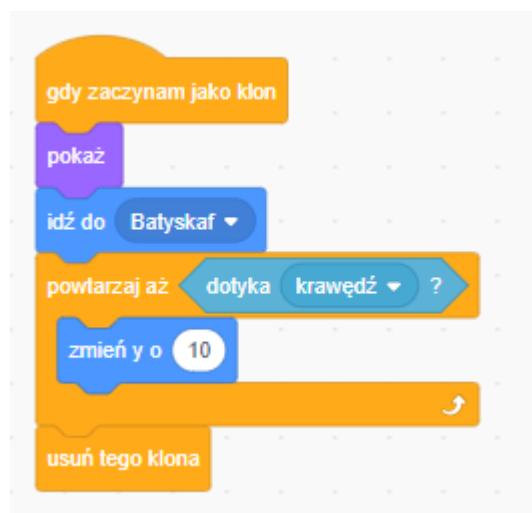
dlatego **należy** skorzystać z bloku
następnie pokazujemy kopię i przemieszczamy ją w aktualną pozycję
naszego statku

Jak myślicie po jakiej osi powinny poruszać się pociski? x czy y?

Niepotrzebne klony musimy usuwać na bieżąco żeby nie stracić na wydajności gry oraz na możliwości klonowania postaci. Służą do tego



blok , pociski kasujemy kiedy dotrą do krawędzi

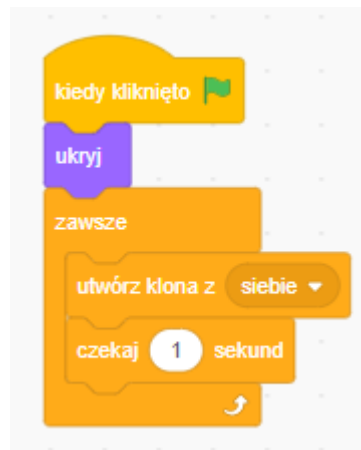


Testujemy działanie strzelania.



#Dodanie przeciwników

Zaczynamy od zaprogramowania mechanizmu klonowania
samoczynnie po określonym czasie.



Określamy zachowanie każdego kłona. Każdy klon posiada wylosowaną pozycję x oraz stałą wysokość pojawiania się. Dodatkowym urozmaicheniem będzie ustawienie losowej wielkości pojawiającego się rekina. W pętli programujemy ruch w dół po osi y, który ma zostać przerwany po dotarciu do krawędzi sceny przez kłona.



Testujemy działanie, na tą chwilę rekin nie reaguje jeszcze na kolizję z graczem oraz pociskiem.

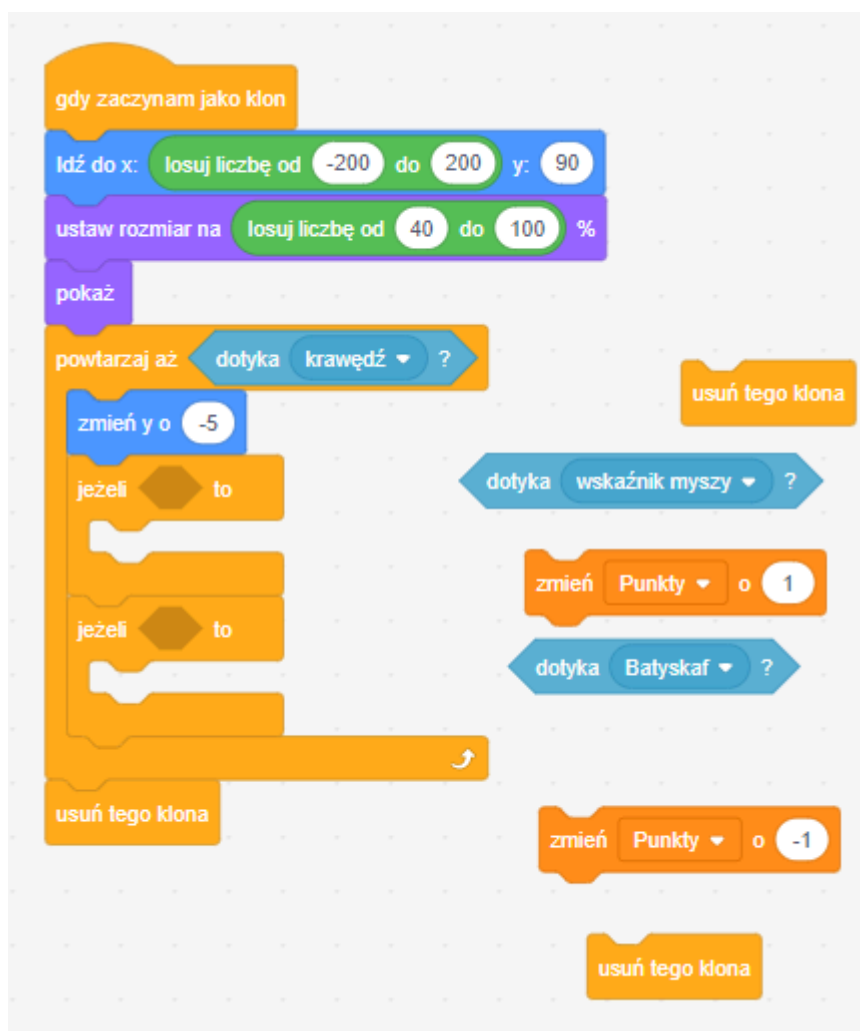


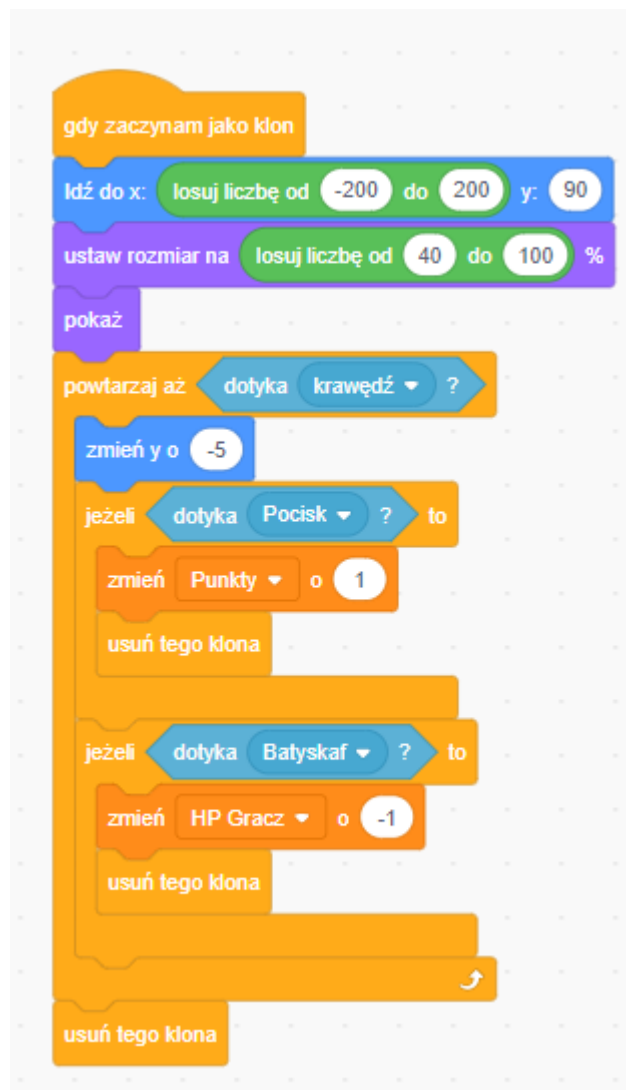
Praca samodzielna
Dopasuj stopień pracy samodzielnej do swojej grupy.
 Możesz też skorzystać z techniki rozsypanki bloków aby ułatwić pracę uczestnikom.
Ważne, upewnij się czy uczestnicy wiedzą co jest do zrobienia.

Zadanie do zaprogramowania:

Jeżeli rekin dotknie pocisku wystrzelonego przez batyskaf wtedy znika a batyskaf otrzymuje punkt. Jeżeli dotknie batyskafu wtedy batyskaf traci jedno życie a rekin znika.

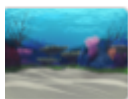
Dodajemy dwa bloki jeżeli... a uczestnicy sami próbują dokończyć.





Testujemy rozgrywkę.

Scena

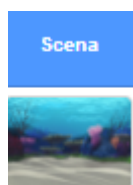
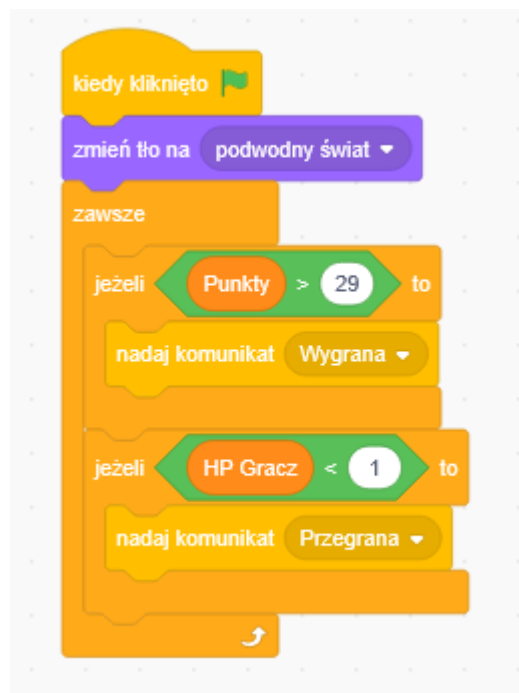


#Warunki zwycięstwa oraz przegranej - komunikaty

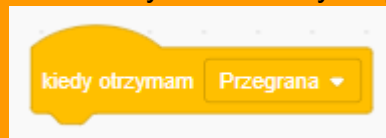
Ustawiamy początkowe tło dla naszej gry. Dodatkowo tworzymy warunki zwycięstwa oraz porażki

Jak myślicie kiedy w naszej grze powinniśmy odnieść zwycięstwo? kiedy przegrać?

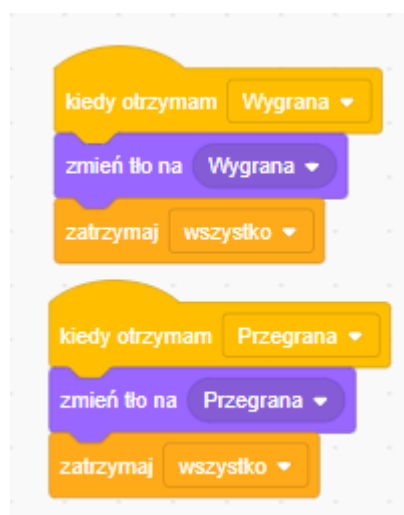
Wykorzystamy teraz komunikaty, które mogą być wysłane w dowolnym momencie naszej gry, duszki mogą odbierać komunikaty wysyłane pomiędzy sobą i odpowiednio reagować



Odbieramy komunikaty za pomocą bloków:



Ustawiamy odpowiednie tło w zależności od komunikatu i zatrzymujemy rozgrywkę

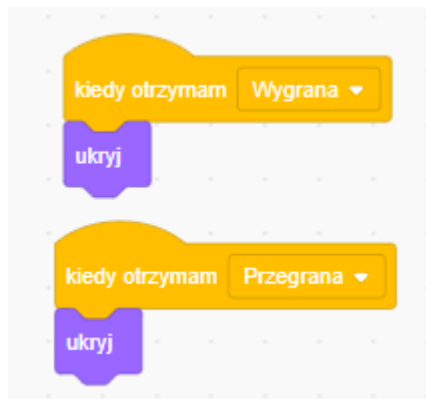


Testujemy działanie.

Co jeszcze na tą chwilę powinniśmy poprawić po wygranej lub przegranej grze?
Ukrycie duszków



Do duszka Batyskaf dodajemy bloki który ukryją duszka w momencie wygranej lub przegranej.



Testujemy poprawność działania całego projektu.

Weryfikacja osiągnięcia celu lekcji i efektów uczenia się

Sprawdzamy działanie projektu u wszystkich uczestników, uczestnicy grają, testują swoje projekty.

Skontroluj u wszystkich uczestników czy osiągnęli określone cele czy nie mają jakiś błędów. Zadaj pytania dotyczące projektu w miarę możliwości każdemu uczestnikowi. Skorzystaj z możliwości dodawania punktów za aktywność uczestników, możesz wykorzystać to jako formę zachęty do aktywizacji.

Przykładowe pytania:

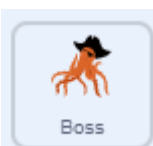
- 1. Czego nauczyliśmy się dzisiaj o mechanizmie klonowania?**
Nauczyłem się, że mechanizm klonowania pozwala na dynamiczne tworzenie kopii duszków w grze, co jest bardzo przydatne przy tworzeniu strzałów lub generowaniu przeciwników. Dzięki temu nie musimy tworzyć wielu duszków a wystarczy kopia jednego.
- 2. Jakie bloki warunkowe wykorzystaliśmy i dlaczego są one ważne?**
Wykorzystaliśmy bloki 'jeżeli...' do sprawdzania, czy użytkownik nacisnął odpowiedni przycisk, co uruchamia określone akcje, np. ruch batyskafu lub strzał. Te bloki są ważne, ponieważ pozwalają na kontrolowanie warunków działania gry – wykonanie określonej akcji tylko wtedy, gdy spełniony jest określony warunek.
- 3. W jaki sposób komunikaty umożliwiają współpracę między duszkami w grze?**
Komunikaty pozwalają na przesyłanie informacji między różnymi duszkami.

Na przykład, gdy jeden duszek (pocisk) trafia przeciwnika, wysyłany jest komunikat, który może spowodować zmniejszenie HP gracza lub aktualizację punktacji. Dzięki temu elementy gry współpracują ze sobą.

4. **Jakie warunki muszą być spełnione, aby klon został usunięty?**

Klon zostanie usunięty, jeśli spełniony zostanie warunek w skrypcie, np. gdy dotknie krawędzi sceny lub przeciwnika. Można to zrobić za pomocą bloku 'jeżeli ... to' i polecenia 'usuń ten klon' w odpowiednim miejscu programu.

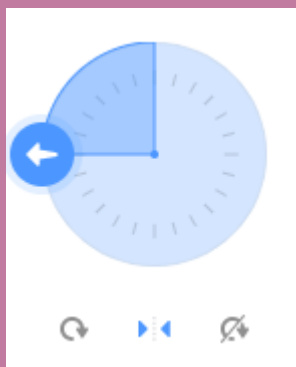
Zadanie dodatkowe (Opcjonalne)



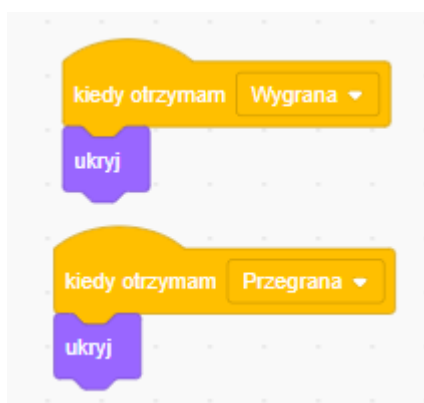
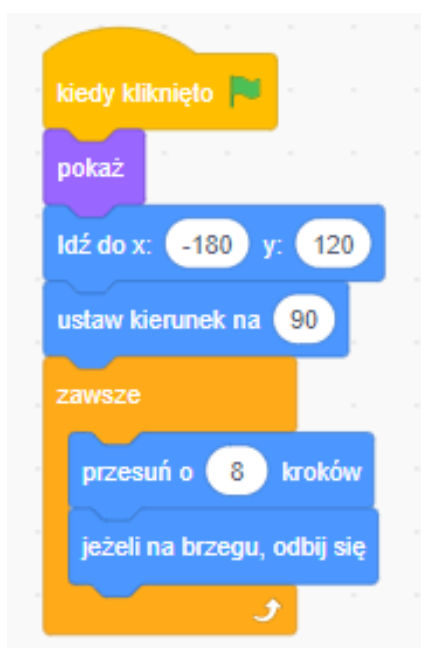
Dodanie przeciwnika - Bossa

Dodanie przeciwnika i zaprogramowanie jego stanu początkowego (pozycji początkowej, pojawiania się za każdym startem gry i ustaleniem kierunku zwrotu postaci). Boss porusza się od krawędzi do krawędzi sceny.

Ważne jest aby styl obrotu Bossa był wybrany na lewo/prawo.



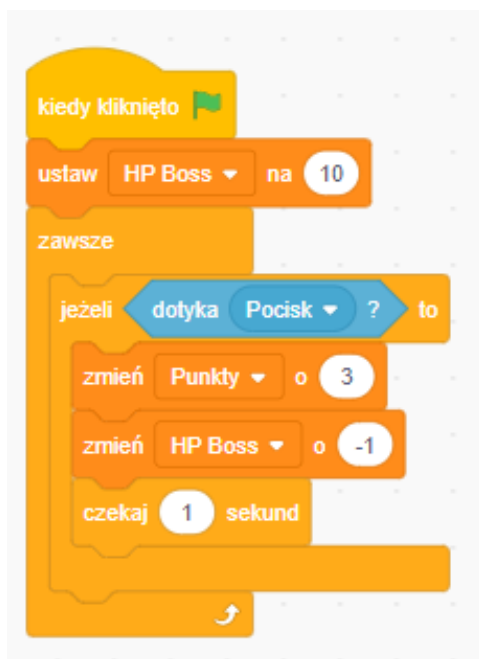
Dodajemy także ukrycie duszka po otrzymaniu komunikatów Wygrana i Przegrana.



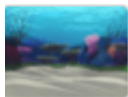


Dodanie zmiennych HP Boss oraz Punkty.

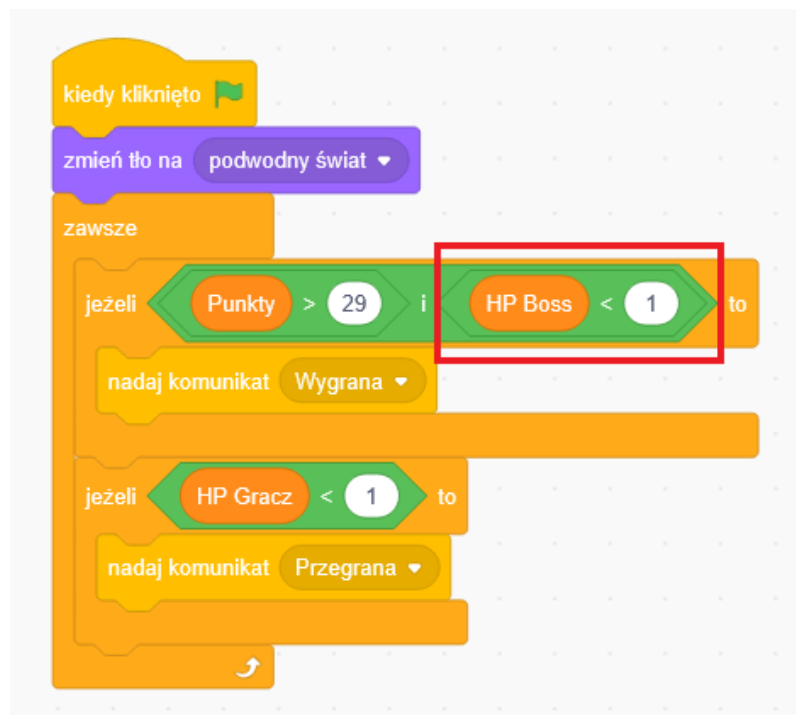
Zmienna **HP Boss** odpowiedzialna jest za punkty życia ośmiornicy
Zmienna **Punkty** to punkty, których zdobywanie jest warunkiem wygranej
Początkowa wartość punktów życia to 10
Zestrzelenie ośmiornicy przy pomocy pocisku zabiera jej jeden punkt życia a gracz zdobywa 3 punkty



Scena

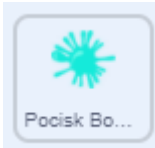


W scenie edytujemy warunek zwycięstwa w grze.
W którym uwzględnimy pokonanie Bossa.



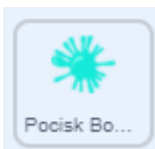
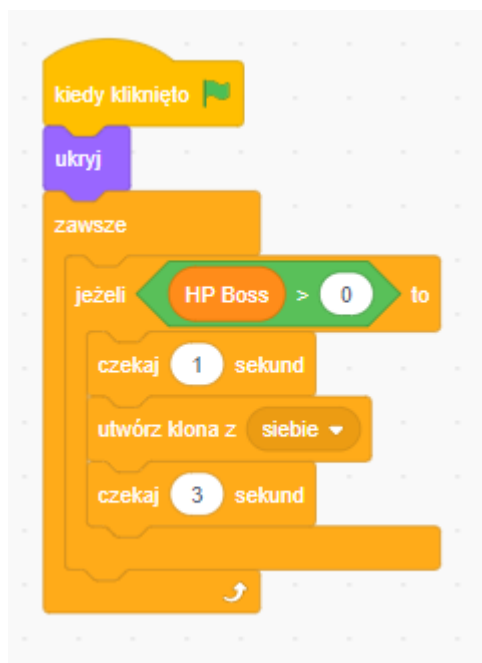
Dodanie efektów po kolizji pocisku z bossem.





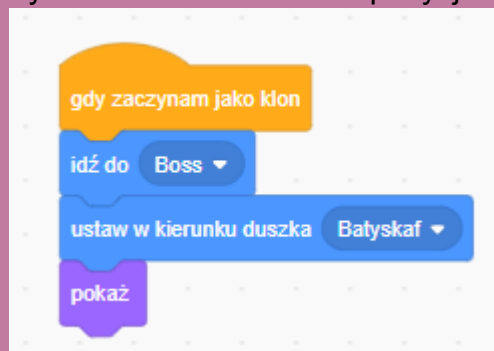
Dodanie pocisku którymi strzela Boss

Dodanie pocisku przeciwnika, który pojawia się co pewien czas pod warunkiem że Boss nie został jeszcze zniszczony.



Pocisk działa jako klon dzięki czemu jest możliwość produkowania większych ilości.

Po utworzeniu ustawia się w kierunku Batyskafu z pozycji startowej jaką jest Boss. Ustawiając kierunek pocisku będzie się on po wystrzeleniu kierować do pozycji Batyskafu

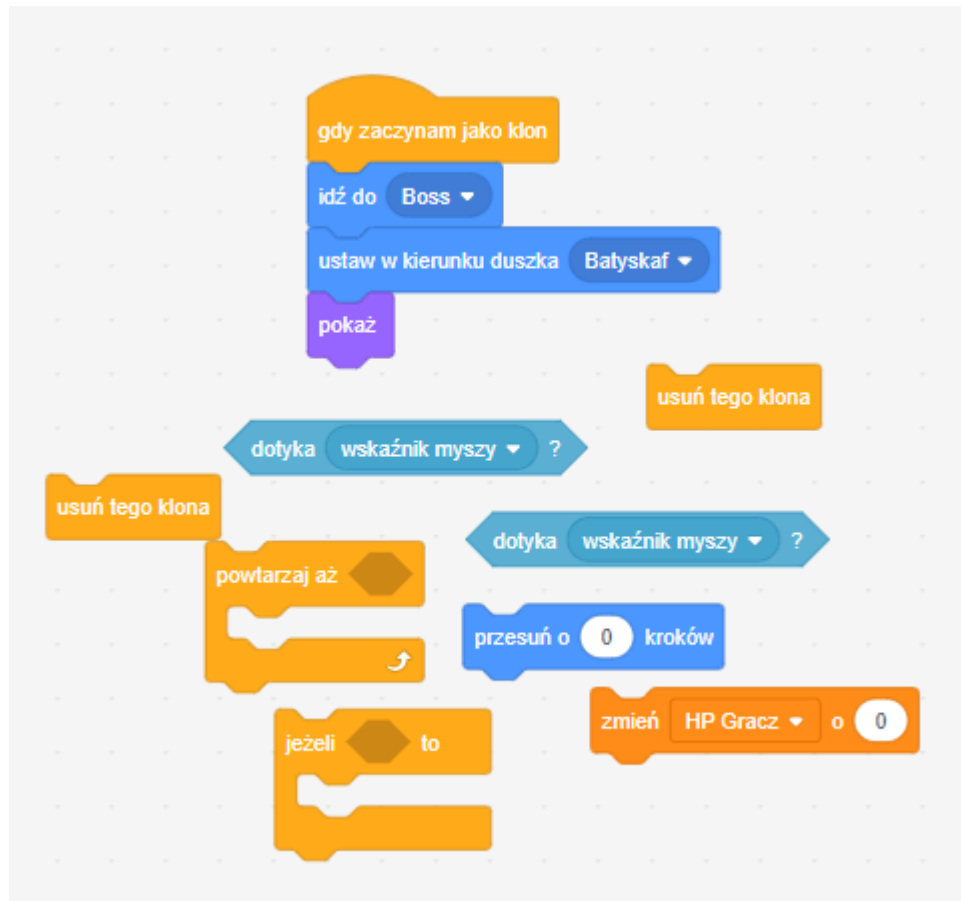


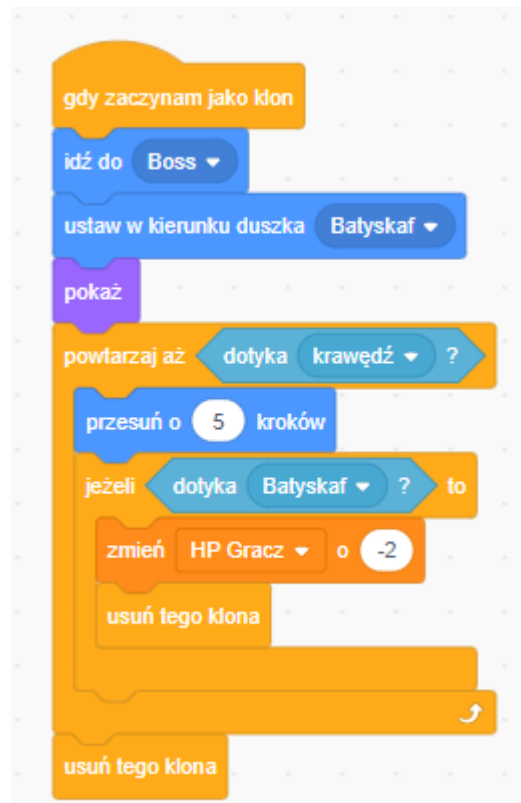
Następnie zawsze przesuwa się w stronę Batyskafu – za każdym razem, gdy nie trafi i dotknie krawędzi to znika. Oczywiście nie podąża za nim przez cały czas, gdyż byłoby to za trudne, gdyby śledził cały czas naszą postać.

Gdy zetknie się z batyskafem zmienia jego punkty życia o 2 i znika

Tworzymy rozsypankę bloków uczestnicy próbują samodzielnie

zaprogramować wzorując się na skryptach rekina





3. Zakończenie lekcji (5-10 minut)

Podsumowanie lekcji - na koniec zajęć podsumuj co udało wam się osiągnąć na tej lekcji, jaki projekt zrobiliście, zadania wykonaliście, czego się nauczyliście, postaraj się zaktywizować całą grupę. Skorzystaj z proponowanych poniższych pytań.

Czy podobał wam się dzisiejszy projekt?

Czy macie jakieś swoje pomysły na ulepszenie tej gry?

Zapowiedź kolejnej lekcji - przedstaw w formie zajawki czym zajmiemy się na kolejnej lekcji, pamiętaj nie ma potrzeby prezentacji gry, projektu z następnych zajęć wystarczy o tym powiedzieć.

Czy znacie grę w która możemy zagrać w przeglądarce Google chrome kiedy stracimy połączenie z internetem?

Na dwóch kolejnych lekcjach będziemy tworzyć własną wersję gry T-rex


<https://chromedino.com/> (Możemy udostępnić link uczestnikom na czacie)

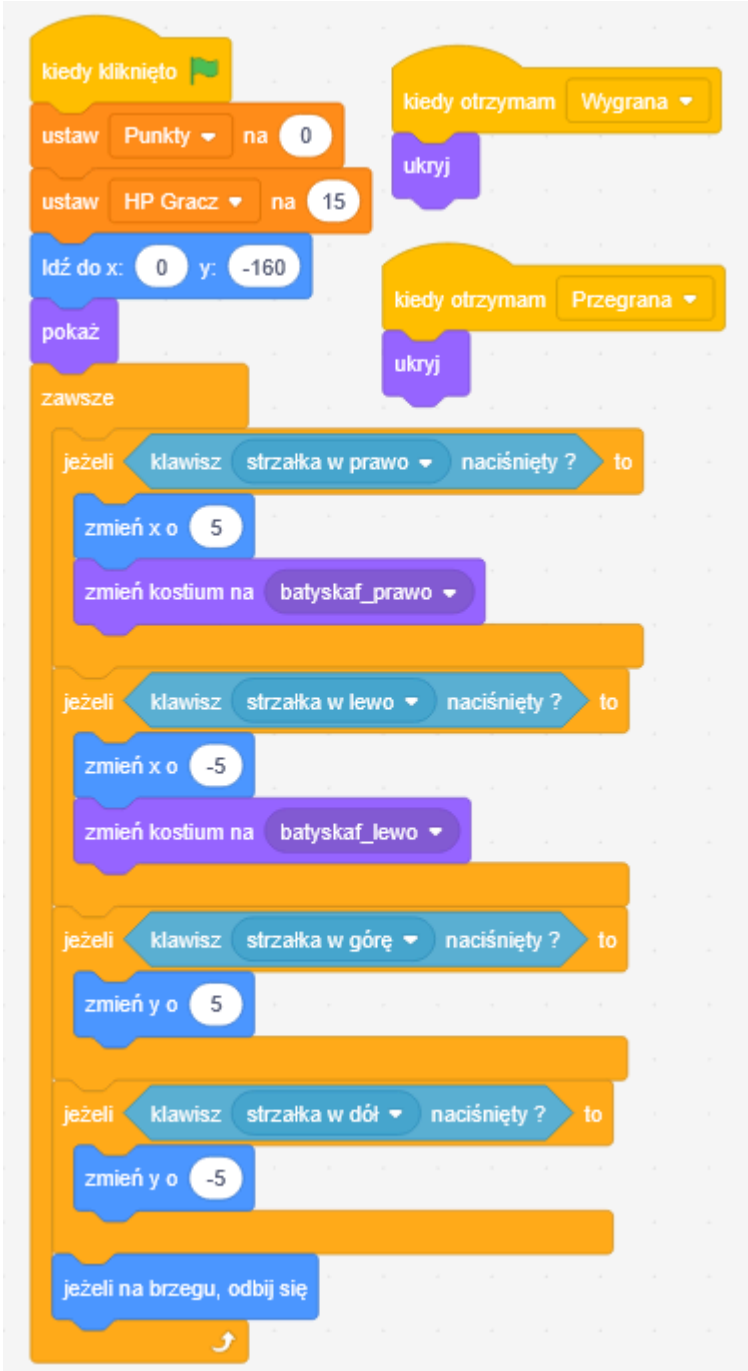
Czeka nas lekcja ćwiczenie refleksu przeskakując nad kaktusami i robiąc unik przed lecącymi ptakami.

Zaprogramujemy mechanizm skoku oraz przewijane tło.

Pytania od uczestników - zanim pożegnasz się z grupą zapytaj na sam koniec czy może ktoś ma jeszcze jakieś pytania dotyczące bieżącej lekcji lub inne.

4. Przegląd skryptów z lekcji


Batyskaf



```
whenClickedFlagClicked
  set Punkty to 0
  set HP Gracz to 15
  move to x: 0 y: -160
  show

whenIReceive Wygrana
  hide

whenIReceive Przegrana
  hide

forever loop
  if key pressed: right arrow
    change x by 5
    change costume to batyskaf_prawo

  if key pressed: left arrow
    change x by -5
    change costume to batyskaf_lewo

  if key pressed: up arrow
    change y by 5

  if key pressed: down arrow
    change y by -5

  if at edge, bounce
```

The image shows a Scratch script for a submarine game. The script is organized into several sections. At the top, there is a 'whenClickedFlagClicked' event that initializes 'Punkty' (Points) to 0 and 'HP Gracz' (Player HP) to 15, moves the submarine to x: 0, y: -160, and shows it. Below this, there are two 'whenIReceive' events: 'Wygrana' (Win) and 'Przegrana' (Loss), both of which trigger a 'hide' action. The main logic is contained within a 'forever loop'. Inside the loop, there are four 'if key pressed' conditions for the right, left, up, and down arrow keys. Each condition triggers a 'change x by' or 'change y by' block and a 'change costume' block. The right arrow increases x by 5 and changes the costume to 'batyskaf_prawo'. The left arrow decreases x by 5 and changes the costume to 'batyskaf_lewo'. The up arrow increases y by 5. The down arrow decreases y by 5. At the bottom of the loop, there is a 'if at edge, bounce' block.

