



Temat: Kosmiczna podróż

Przedmiot: zajęcia komputerowe/nauczanie zintegrowane

Klasy: 1-3

Autor: Sylwester Zasoński

Czas trwania: 2 godziny lekcyjne po 45 min

Cele ogólne:

- Rozwijanie kompetencji miękkich
- Pobudzanie kreatywności
- Praca manualna
- Łączenie kluczowych obszarów rozwoju

Cele operacyjne:

Uczeń:

- Potrafi szukać sposobów rozwiązania zadania czy problemu
- Wie, jak funkcjonują układy elektryczne
- wykonuje prace, modele, rekwizyty, impresje plastyczne potrzebne do aktywności artystycznej i naukowej
- planuje i realizuje własne projekty/prace; realizując te projekty/prace współdziała w grupie
- programuje wizualnie: proste sytuacje lub historyjki według pomysłów własnych i pomysłów opracowanych wspólnie z innymi uczniami, pojedyncze polecenia, a także ich sekwencje sterujące obiektem na ekranie komputera bądź innego urządzenia cyfrowego
- zapisuje efekty swojej pracy we wskazanym miejscu.
- posługuje się komputerem lub innym urządzeniem cyfrowym oraz urządzeniami zewnętrznymi przy wykonywaniu zadania
- korzysta z udostępnionych mu stron i zasobów internetowych
- wykonuje przedmioty użytkowe, w tym dekoracyjne i modele techniczne
- tworzy polecenie lub sekwencje poleceń dla określonego planu działania prowadzące do osiągnięcia celu







 współpracuje z uczniami, wymienia się z nimi pomysłami i doświadczeniami, wykorzystując technologię

Metody:

Poszukujące, eksponujące

Środki dydaktyczne:

- Klocki lego do zbudowania konstrukcji. W przypadku braku klocków konstrukcje można zbudować z innych elementów
- 2. Zestaw Wynalazcy Makeblock Neuron
- Komputer z aplikacją mBlock oraz adapterem Bluetooth od Makeblock (w przypadku braku kompatybilności z `Neuronem przez wbudowany moduł Bluetooth)

Przebieg zajęć:

Zadanie będzie polegało na stworzeniu gry w aplikacji mBlock, w której sterujesz statkiem kosmicznym i unikasz innych statków (ufo). Przy czym nie będziesz grać za pomocą klawiatury, ale statkiem zbudowanym z klocków lego.

Zapytaj uczniów jak wyobrażają sobie taką grę? Co będzie potrzebne?

Zastanówcie się wspólnie jakich modułów użyjecie? Który z modułów zawartych w zestawie wynalazcy posłuży jako kontroler?

Prawidłową odpowiedzią powinien być żyroskop.

W najprostszej wersji gry statek będzie się poruszał na boki (lewo,prawo). Czyli po osi X.

Zapewne jeszcze uczniowie nie poznali układu współrzędnych i osi X,Y. Wystarczy jednak wytłumaczyć, że przechylając statek z klocków w lewo, statek na ekranie również przechyli się w lewo. Osiągną to zmieniając X o wybraną wartość:

- W lewo będzie to liczba ujemna.
- W prawo liczba dodatnia.

Zanim zaczniesz budowę musicie z uczniami określić wielkość statku tak aby pomieścił wewnątrz potrzebną elektronikę, czyli moduły Neuron. W tym celu należy złożyć układ.

Oprócz żyroskopu potrzebne będzie jeszcze zasilanie i Bluetooth.







Układ powinien wyglądać następująco.



Teraz pora na kreatywną część, czyli budowę statku z klocków. Złożony układ posłuży do określenia jego długości, szerokości i wysokości. Założenie jest takie, aby układ był w środku, czyli zabudowany z każdej strony.

Poniżej przykładowa konstrukcja.



Konstrukcja gotowa? Pora poszukać grafik. Potrzebna będzie grafika statku oraz grafika przeciwnika np. Ufo. Darmowych grafik możesz szukać między innymi na:

https://pl.freepik.com/

https://pixabay.com/

Pamiętaj również o grafice na tło gry, może jakiś kosmos?

Masz wszystko? Czas na kod.







Programujemy!

Programowanie gry oczywiście rozpoczniesz od uruchomienia aplikacji mblock dostępnej na komputery PC i Mac.

Program pobierzesz z:

http://www.mblock.cc/software/mblock/mblock5/

Po otwarciu aplikacji zacznij od usunięcia Codey Rocky, w zamian dodaj Neurona.



Następnie przejdź do zakładki "Sprites" czyli duszków. Teraz dodaj 2 duszki: 1 statek, którym będziesz sterować, 2 przeciwnik.





Ment•r

SYSTEMY AUDIOWIZUALNE



≡				
		5 C	Blocks v	- 100% +
	Looks	switch backdrop to backdrop1 -		· · · · · · · · · · · · · · · · · · ·
Qs	earch	Sprite Library	My Sprite	
Devices Sprites Bo	+	urlo-1608553	-30	
			Cancel OK	

Duszki musisz najpierw wgrać do biblioteki.

Po kliknięciu + pod "Sprites" otworzy się okno z biblioteką duszków "Sprite Library". Obok zobaczysz napis "My Sprite" czyli "Mój duszek" przejdź na tą zakładkę. Kliknij kolejny + i otworzy się okno. Teraz wystarczy wybrać wcześniej ściągniętą grafikę i dodać ją do biblioteki.

Kolejny krok to zaznaczenie dodanej przed chwilą grafiki i dodanie jej do sceny. Grafika ściągnie się w domyślnym rozmiarze, ale nie przejmuj się, że jest za duża. W łatwy i szybki sposób ja zmniejszysz. Poszukaj na dole po lewej stronie przycisku "Costumes" - kostiumy. Kliknij a ukaże ci się menu edycji kostiumu duszka.







Klikając na duszka otrzymasz możliwość zmiany jego wielkości. Wystarczy przeciągnąć jeden z punktów. Pamiętaj, aby zmniejszona grafika była w centralnym punkcie pola edycji.

≡				Untitled)			SLYSK	
		rocket-1	Costume Fill	rocket-147466 • Outline •	Group Ungroup Fer Copy Paste Filp	← ↔ ward Backward Front Horizontal Flip Vertical	4 Back		8
r P			► ✓	ћ 8 Т					
0			/	0					
Devices Sprites	Background				1 <u>4</u> 1				
* +	Sprite O rocket-147466,1280 C O D Size Direction 100 90			Q = Q					
	Costumes Sounds	Add Blank Costume							

Tym samym sposobem dodaj drugą grafikę.

Grafiki gotowe więc do kodu.

Neuron

Programowanie proponuję rozpocząć od kodu Neurona.

Algorytm, czyli ciąg bloków kodu rozpoczynamy od "When clicked" czyli "Kiedy klikniesz zieloną flagę"

Zielona flaga zawsze rozpoczyna działanie aplikacji, dlatego bardzo często znajdziesz ją w przeróżnych kodach.

Przypomnę zasadę działania. Kiedy przechylisz statek z żyroskopem w lewo, rakieta na ekranie ma przesunąć się po osi x w lewo, analogicznie będzie wyglądać sytuacja w prawo.

Żyroskop Neurona będzie komunikować się z duszkiem za pomocą komunikatów.

Tak więc kod powinien wyglądać następująco:

Jeżeli żyroskop przechylony w lewo wtedy

Nadaj wiadomość "L"

Jeżeli żyroskop przechylony w prawo wtedy





Nadaj wiadomość "P"

"L" to skrót od lewo, "P" to skrót od prawo. Nazwy wiadomości możesz nadawać dowolnie. Znajdziesz je w kategorii "Events" - wydarzenia. Do dyspozycji masz 3 podstawowe bloczki.

when 🍽 clicked
when space key pressed
when I receive b 💌
broadcast b -
broadcast b • and wait

- When I recieive kiedy otrzymam wiadomość
- Broadcast nadaj wiadomość
- Broadcast and wait nadaj wiadomość i czekaj

broadcast b -	New Message
 ✓ b f I message 	New message name: hame cannot contain (&<>'")
r start New message	Cancel

Z każdego bloczku wiadomości wysuwa się menu w którym znajdziesz utworzone wcześniej wiadomości oraz "new message" czyli opcję utworzenia nowej.

Po wybraniu "new message" - nowa wiadomość pojawi się okno, gdzie wpiszesz nazwę nowej wiadomości.

Powyżej rozpisany kod ma jedną poważną wadę, zostanie uruchomiony i sprawdzony tylko raz, po naciśnięciu flagi. Aby warunki "jeżeli" - po ang."if" były ciągle sprawdzane,







czyli odczyt z żyroskopu był ciągle weryfikowany, potrzebna będzie pętla zawsze – po angielsku "forever".

Finalny kod powinien prezentować się następująco:



Tak, na początku kodu pojawia się jeszcze jeden nie omawiany wcześniej bloczek "broadcast start". O tym po co akurat tutaj jest i do czego służy dowiesz się za chwilę.

Rakieta

Będziesz programować duszka rakiety, kliknij zatem na niego aby otworzyć panel do układania kodu dla tego duszka.

Zacznij od przeniesienia informacji z żyroskopu tak aby wprawić duszka rakiety w ruch.

Pisałem wcześniej o tym, że duszek będzie się poruszał po osi X. Tak więc poziomo.

Za ten ruch odpowiada przechylanie żyroskopu w lewo/prawo. Tak więc i przy duszku potrzebne są 2 algorytmy:

Kiedy otrzymam L

Zmień x o -10

Kiedy otrzymam P

Zmień x o 10







Osie w mBlock są identyczne jak w Scratch. Dlatego jeżeli chcemy aby duszek poruszał się w lewo, x musi przyjąć wartość ujemną. Im większa liczba tym szybciej będzie się przesuwać.



Gotowy kod możesz przetestować. Połącz komputer z neuronem.

Aby to zrobić przejdź do zakładki "devices" (urządzenia) i kliknij connect/połącz. Wybierz urządzenie z listy i kliknij "connection" - połączenie.







Jeżeli kod działa czas przejść dalej i dodać do rakiety kolejny kod.

Tym razem określimy parametry startowe (czyli po kliknięciu flagi):

Kiedy flaga kliknięta

Pokaż Zmień kostium na rakieta Idź do x 0 y –100

Nadaj start

Po co ci bloczek "pokaż/show"?

To jest dobry moment, aby zastanowić się nad sensem gry.

Samo latanie statkiem jest nudne. W grze będziesz unikać innych statków. A co się stanie, gdy zderzysz się z przeciwnikiem?

Ja założyłem, że rakieta wybuchnie po czym zniknie, aby ponownie móc się pojawić, by móc grać dalej.

Więc skoro znika to musi się gdzieś pojawiać i na pewno pojawi się przy uruchomieniu programu. To taki parametr startowy, tak samo jak wybór kostiumu oraz ustawienie pozycji początkowej rakiety. Dopiero potem, gdy wszystko jest gotowe ten kod może inicjować pozostałe np. Przeciwników.









Kolejny kod to doskonały przykład co się będzie dziać, jak program nada wiadomość "start". Poniższy kod dotyczyć będzie tego co się stanie z rakietą po zetknięciu z innym przedmiotem np. duszkiem "ufo".

Podobnie jak w poprzednim kodzie, 2 znane polecenia: pokaż, zmień kostium na rakieta. Jednak to co powinno cię zainteresować to propozycja tego co może się stać po spotkaniu z przeciwnikiem.

Wystarczy postawić prosty warunek (if) i dopilnować aby był ciągle sprawdzany (forever)

Przykładowy kod:

Kiedy otrzymam start

Pokaż

Zmień kostium na rakieta

Zawsze

Jeżeli dotyka "ufo" wtedy

Zagraj dźwięk "recording1"

Zmień kostium na

Ufo to kolejny duszek, tym razem przeciwnika.

MBlock dysponuje bazą standardowych dźwięków. Póki co jedynym sposobem na dodawanie nowych jest ich nagranie za pomocą "recordera" i mikrofonu. Opcję ta znajdziesz klikając w zakładkę "sprites". Na dole pojawi się przycisk "Sounds" a po jego naciśnięciu przejdziesz do panelu edycji sampla.









W dolnym środku znajdziesz opcje: "record sound" – nagraj dźwięk; "add sound" – dodaj dźwięk. Dzięki temu duszek rakiety po zetknięciu się z innym duszkiem wyda dźwięk

Ciekawą propozycją jest również zrobienie animacji wybuchu po zderzeniu z przeciwnikiem. Do najprostszej wystarczą 3 kostiumy i odstęp czasowy wynoszący ułamek sekundy. Zerknij na kod poniżej.

vhe	n I re	eceive	sta					
ho	w		e ses	9				
wit	ch co	ostum	e to (rak	ieta	1 -		
ore				2	3	3		
	$\overline{\langle}$	touchi	ing	ufo	•	?	> th	
	start	sound	rec	ord	ing	1 •		
	switc	h cost	tume	to (wy	buc	ch1	•
	wait	0.1	secs		•	-		
	switc	h cost	tume	to (wy	buc	ch2	-
	wait	0.1	secs			-		
	switc	h cost	tume	to (wy	buc	ch3	•
	wait	0.1	secs		8	ŝ	ŝ	2
	hide							
	broad	dcast	star	t 💌				

Nie zapomnij tylko dodać i zmienić nazwę kolejnych kostiumów.









Kostiumy możesz zarówno dodawać albo kopiować już istniejący. W tym przypadku dodałem darmową grafikę wybuchu i skopiowałem ją 2 razy. W dwóch kolejnych kostiumach obróciłem ją i zmieniłem jej rozmiar.

Pamiętaj! Gdy animacja się skończy, ukryj duszka i nadaj wiadomość "start"

Przeciwnik – UFO

Zmień duszka na duszka drugiego statku, będącego przeciwnikiem

Zadaniem przeciwnika jest poruszanie się po planszy w losowy sposób z góry na dół. Dlatego też najpierw potrzebować będziesz ustawić duszka w losowej pozycji na osi x i stałej pozycji na oxi Y. Następnie duszek powinien "polecieć" w losową pozycję na dole ekranu i ukryć się, czyli zniknąć. Ten kod zapętlony da efekt pojawiania się kolejnych statków.

Przykładowy kod powinien wyglądać:

Kiedy otrzymam wiadomość start Pokaż Idź do X: Iosuj –200 do 200 y:190 Leć Iosuj 1 do 10 sekund do x: Iosuj-200 do 200 y: -200 Schowaj Nadaj wiadomość start Kodowanienaekranie.pl

13





Zwróć uwagę, że kod rozpoczyna i koczy się nadaniem wiadomości start co tworzy pętlę. Ukrywając na końcu kodu statek, musimy ponownie go pokazać na początku.

		· ·											
-200	to	200	y: (183	 								
to	10	secs	s to x:	picł	< ranc	lom	-20)0 t	:0 2	200) y:	-20	00
	-200 to (-200 to t	-200 to 200 to 10 secs	-200 to 200 y: (to 10 secs to x:	-200 to 200 y: 183	-200 to 200 y: 183 to 10 secs to x: pick rand	-200 to 200 y: 183 to 10 secs to x: pick random	-200 to 200 y: 183	-200 to 200 y: 183 to 10 secs to x: pick random -200 t	-200 to 200 y: 183 to 10 secs to x: pick random -200 to 2	-200 to 200 y: 183 to 10 secs to x: pick random -200 to 200	-200 to 200 y: 183 to 10 secs to x: pick random -200 to 200 y:	-200 to 200 y: 183 to 10 secs to x: pick random -200 to 200 y: -20

Powyższy kod odpowiada za ruch statku przeciwnika, możesz go teraz przetestować.

To prawie koniec programowania, pozostała tylko jedna sekwencja, wiesz jaka?

Brakuje kodu odpowiedzialnego za to co się stanie ze statkiem przeciwnika po dotknięciu rakiety gracza? Powinien zniknąć, prawda?

Kod ten może zaczynać się od bloczka "kiedy flaga naciśnięta" ale musisz pamiętać, że warunek, który się w nim znajdzie (kiedy dotykam rakiety) musi być sprawdzany nieustannie, czyli potrzebujesz pętli.

Przykładowy kod:

Kiedy flaga kliknięta

Zawsze

Jeżeli dotyka duszka wteddy

Ukryj

Czekaj 1 sekundę

Nadaj wiadomość start

Dzięki temu zaraz po uruchomieniu, program sprawdza czy duszek nie dotyka duszka rakiety. Jeżeli tak się stanie wtedy ukrywa duszka ufo i czeka sekundę a potem nadaje wiadomość start. Po otrzymaniu wiadomości start uruchamia się poprzedni kod, czyli duszek ufo pojawia się u góry ekranu i zaczyna lecieć w dół. Taki kod daje wrażenie pojawiania się kolejnych statków ufo a tak naprawdę to cały czas jest ten sam który pojawia się i znika.









Scena

Ostatni i najprostszy etap to dodanie tła do gry. Przejdź do sceny klikając "Backgrounds" obok "Sprites". Wybierz "Costiumes" i podobnie jak w przypadku duszków dodaj tło (add background).

Gra gotowa, czas na zabawę!

Zadanie ekstra:

Dodaj kolejnego duszka przeciwnika latającego po planszy.

