

NAUKA SCRATCH, LEKCJA 1: JAK ZACZAĆ?

1. Zainstaluj aplikację

Tak naprawdę chodzi o to, żebyś miał możliwość zapisywania wykonywanej pracy (projektów). Pamiętaj o tym, że tylko instalując aplikację masz możliwość tworzenia i edycji projektów na dysku. Natomiast po zarejestrowaniu (punkt 2) możesz przechowywać swoje projekty na stronie scratch.mit.edu, łatwo je udostępniać i edytować przez www.

Jak zainstalować aplikację do edycji offline?

Wejdź na stronę: <https://scratch.mit.edu/download>





Scratch 2 Offline Editor

You can install the Scratch 2.0 editor to work on projects without an internet connection. This version will work on Mac, Windows, and some versions of Linux (32 bit).

Adobe AIR







If you don't already have it, download and install the latest [Adobe AIR](#)

Mac OS X - [Download](#) 
Mac OS 10.5 & Older - [Download](#) 
Windows - [Download](#) 
Linux - [Download](#) 

Scratch Offline Editor






Next download and install the Scratch 2.0 Offline Editor

Mac OS X - [Download](#) 
Mac OS 10.5 & Older - [Download](#) 
Windows - [Download](#) 
Linux - [Download](#) 

Support Materials

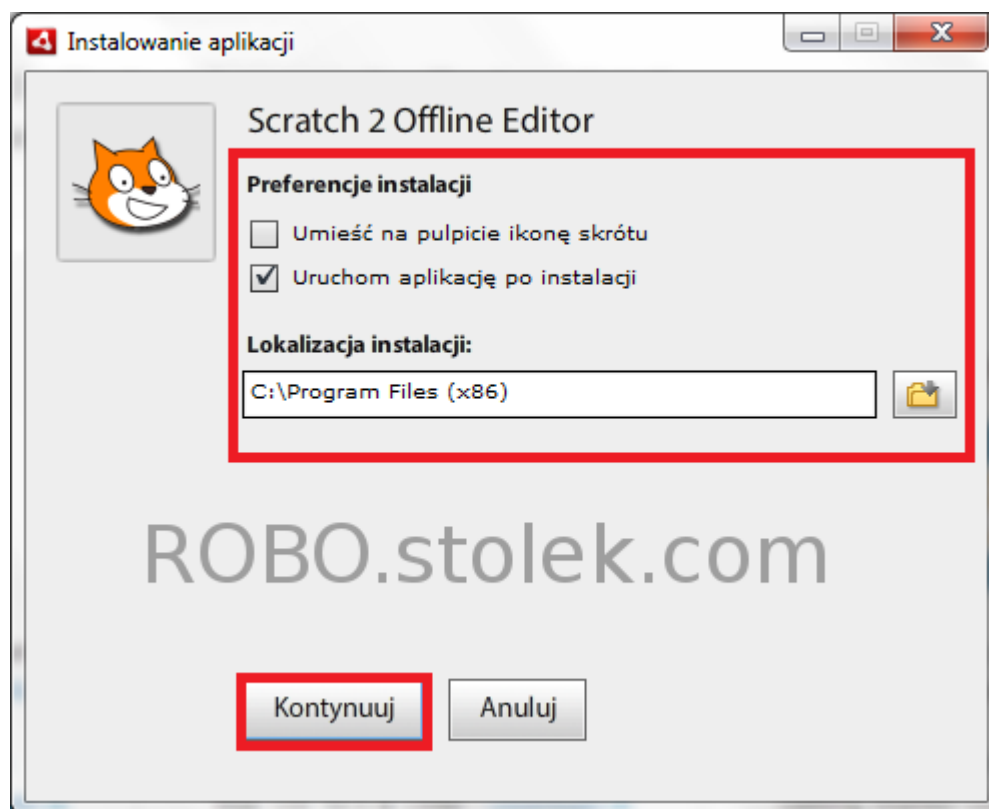


Need some help getting started?
Here are some helpful resources.

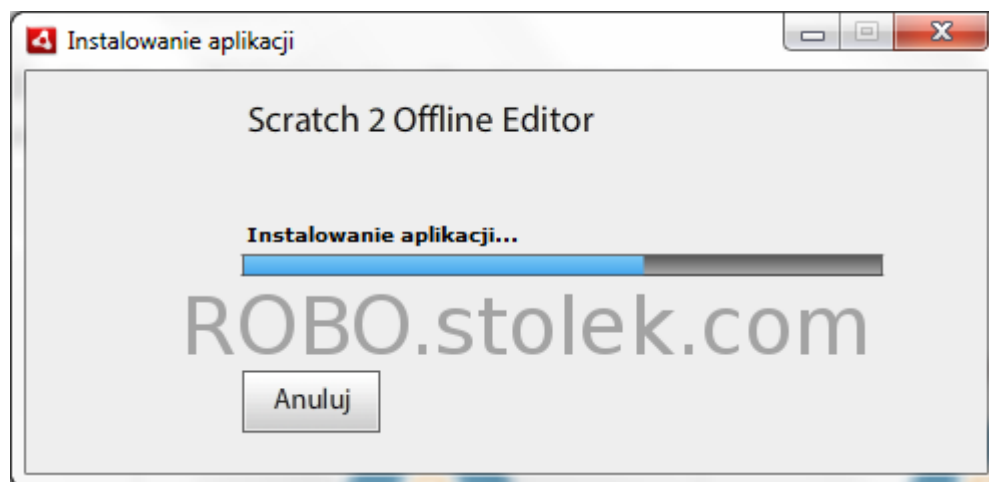
Starter Projects - [Download](#) 
Getting Started Guide - [Download](#) 
Scratch Cards - [Download](#) 

ROBO.stolek.com

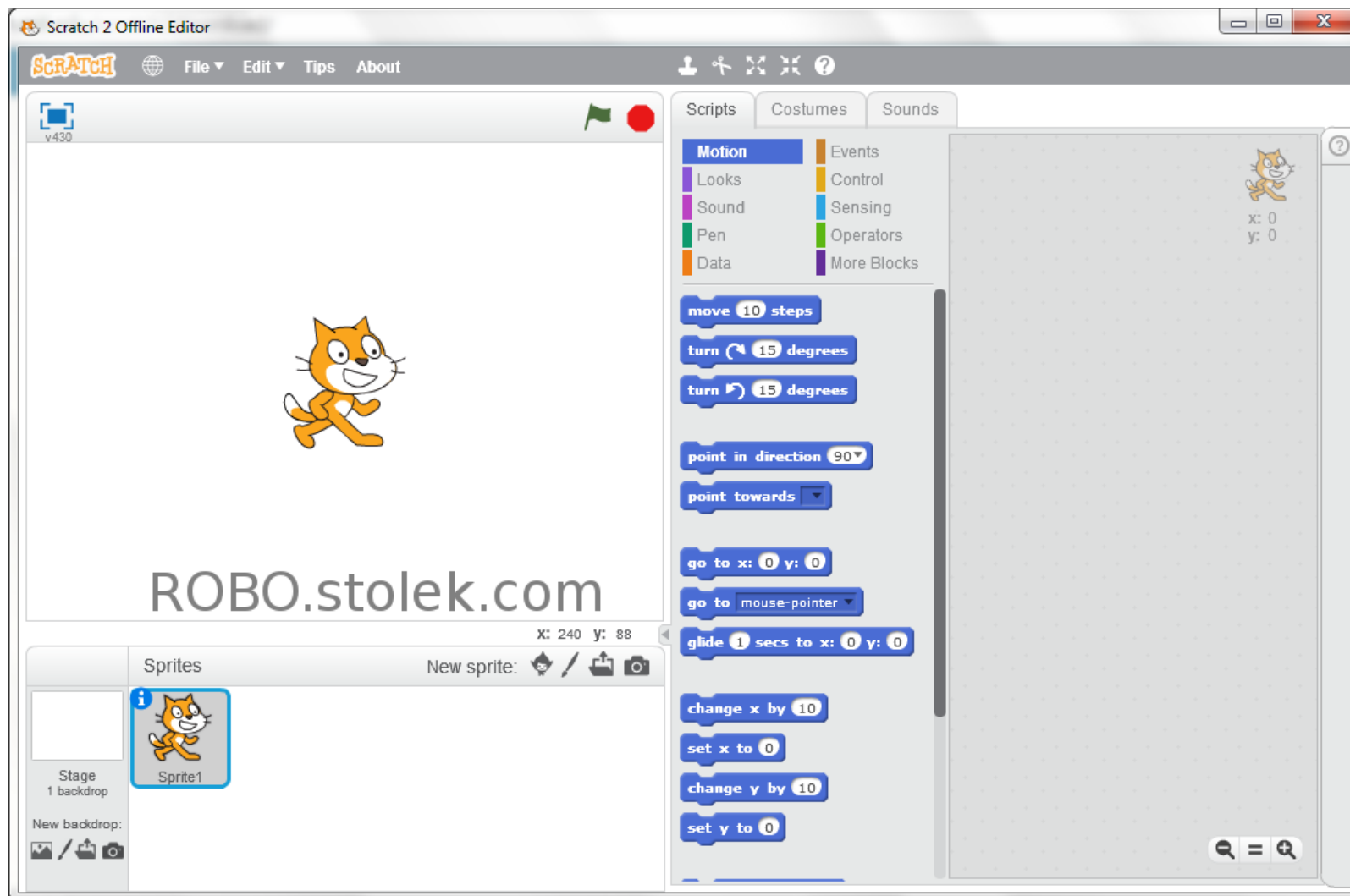
Jeśli nie masz zainstalowanego środowiska AdobeAIR musisz je zainstalować wybierając instalację dla swojego systemu operacyjnego. Po zainstalowaniu AdobeAIR możesz ściągnąć ScratchOfflineEditor i uruchomić instalację:



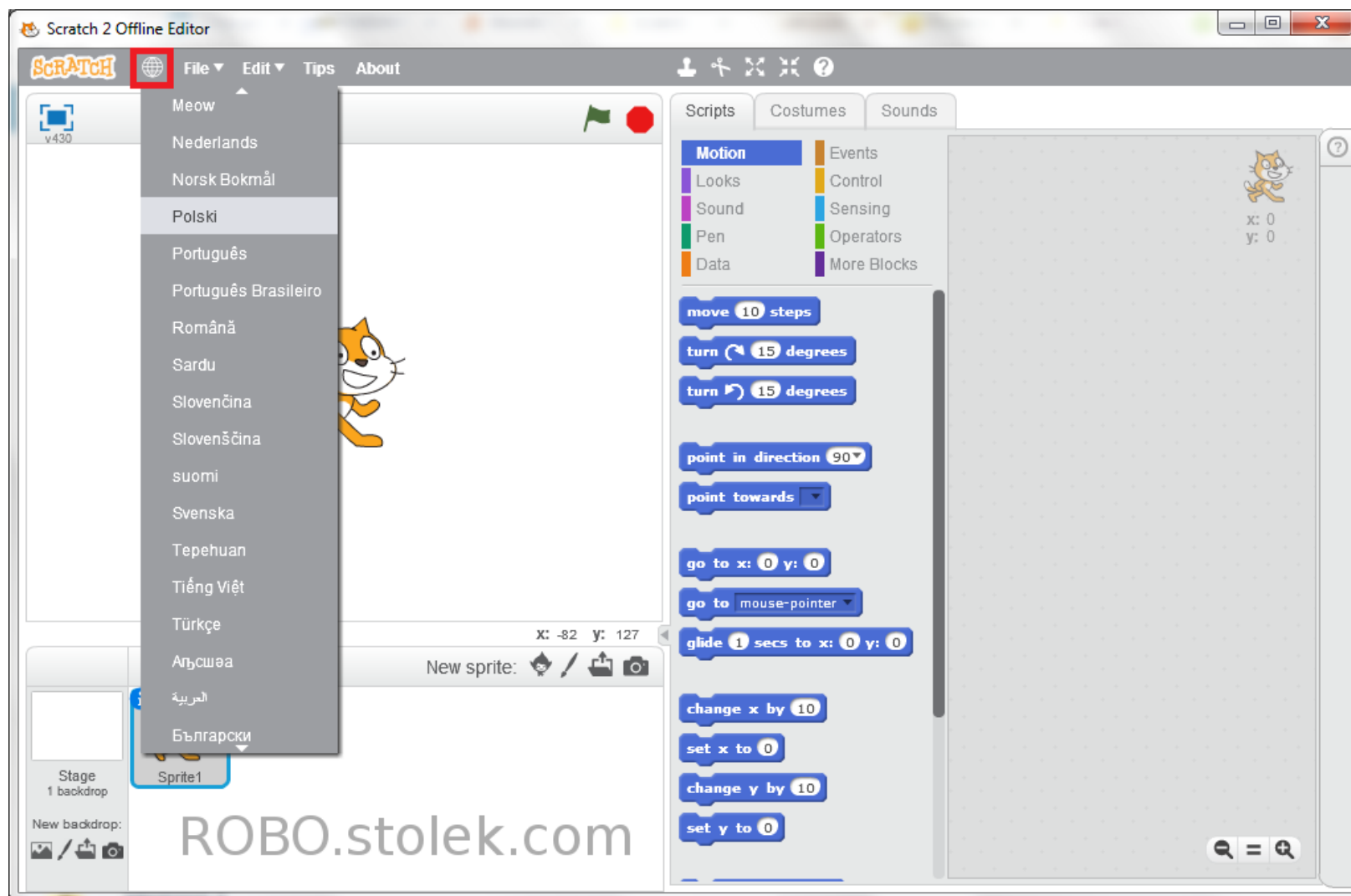
Program instalacyjny poprosi o potwierdzenie podstawowych parametrów instalacji, po wybraniu możesz kliknąć „Kontynuuj”. Program powinien przejść do instalacji:



Jeśli wybrałeś opcję uruchomienia aplikacji, po zakończeniu instalacji zobaczysz następujący ekran:



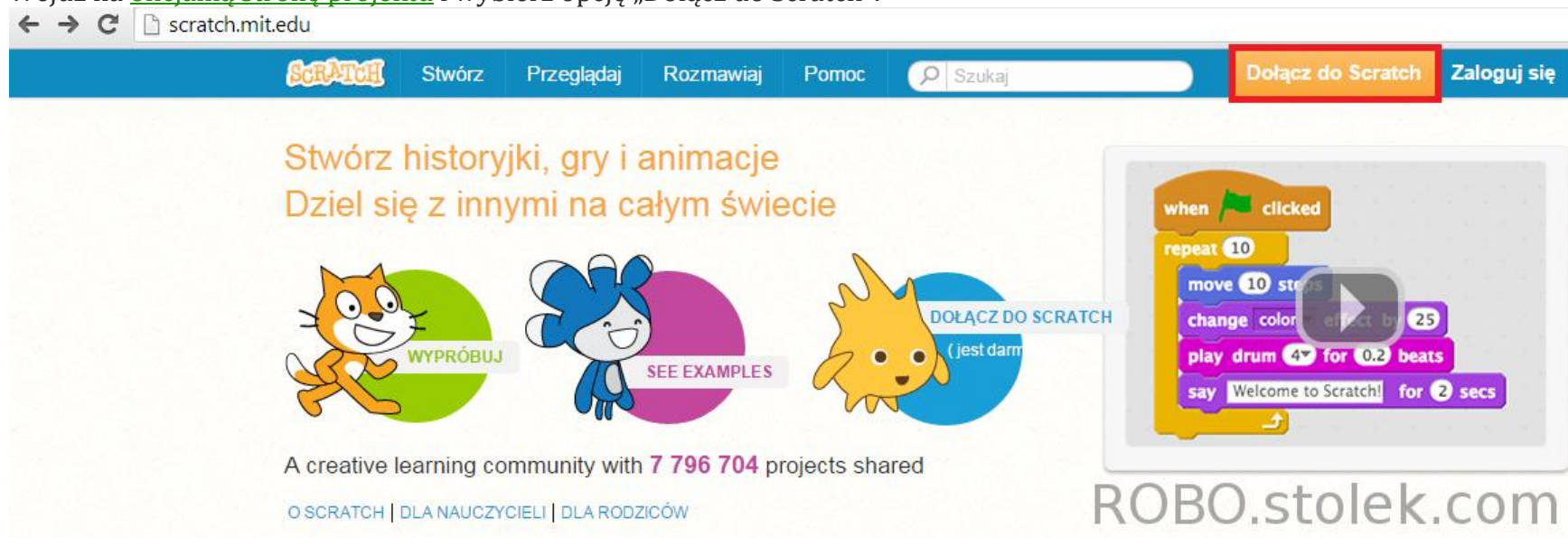
Na początek możesz zmienić język aplikacji na polski, klikając w ikonę globusa:



Jeśli widzisz uruchomiony program to znaczy, że wszystko poszło dobrze. Teraz czas na zarejestrowanie się – do ekranu jeszcze wrócimy.

2. Zarejestruj się na oficjalnej stronie

Wejdź na [oficjalną stronę projektu](https://scratch.mit.edu) i wybierz opcję „Dołącz do Scratch”:



Na kolejnym ekranie musisz wybrać swój login oraz hasło i wcisnąć „next”:

Dołącz do Scratch

X

It's easy (and free!) to sign up for a Scratch account.

Choose a Scratch Username

twoj_login


Choose a Password

.....

Potwierdź hasło

.....

Retype your password



ROBO.stolek.com

1

2

3

Next

Kolejny ekran to podstawowe informacje o Tobie, w tym adres mailowy (wykorzystywany do resetowania hasła), po wpisaniu klikamy „next”:

Dołącz do Scratch



Your responses to these questions will be kept private.

Why do we ask for this info [?](#)

Birth Month and Year

- Month - ▼

- Year - ▼

Gender



Male



Female



Country

Poland ▼

Email address



ROBO.stolek.com

1

2

3

Next

Gotowe, jeśli wszystko poszło dobrze powinieneś zobaczyć ekran podobny do tego, możesz przejść dalej klikając „OK, Let’s Go!”:

Dołącz do Scratch



Dziękujemy za dołączenie do Scratch!

You're now logged in.

Scratch is a community of all ages, from all over the world. Make sure your projects and comments are [respectful and friendly](#).

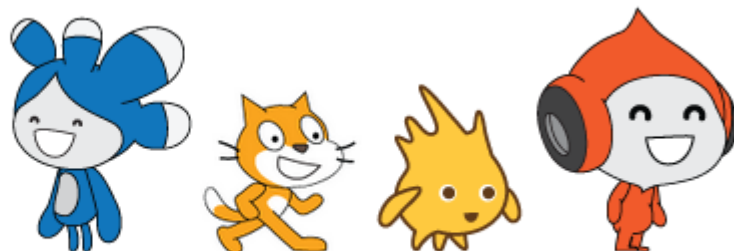
Would you like to:

[Learn how to make a project](#)

[Choose a starter project](#)

[Connect with a Scratcher](#)

ROBO.stolek.com



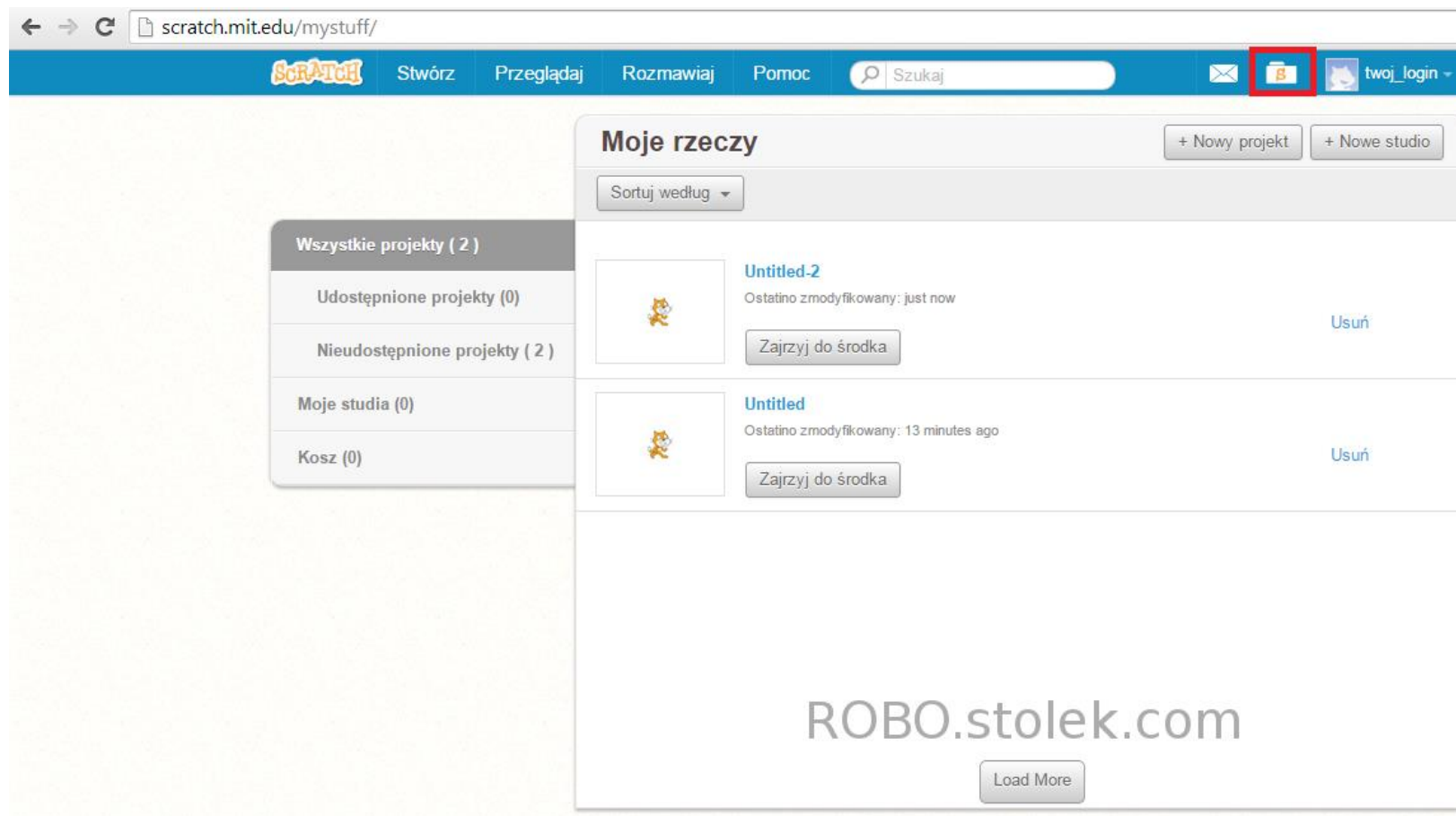
1

2

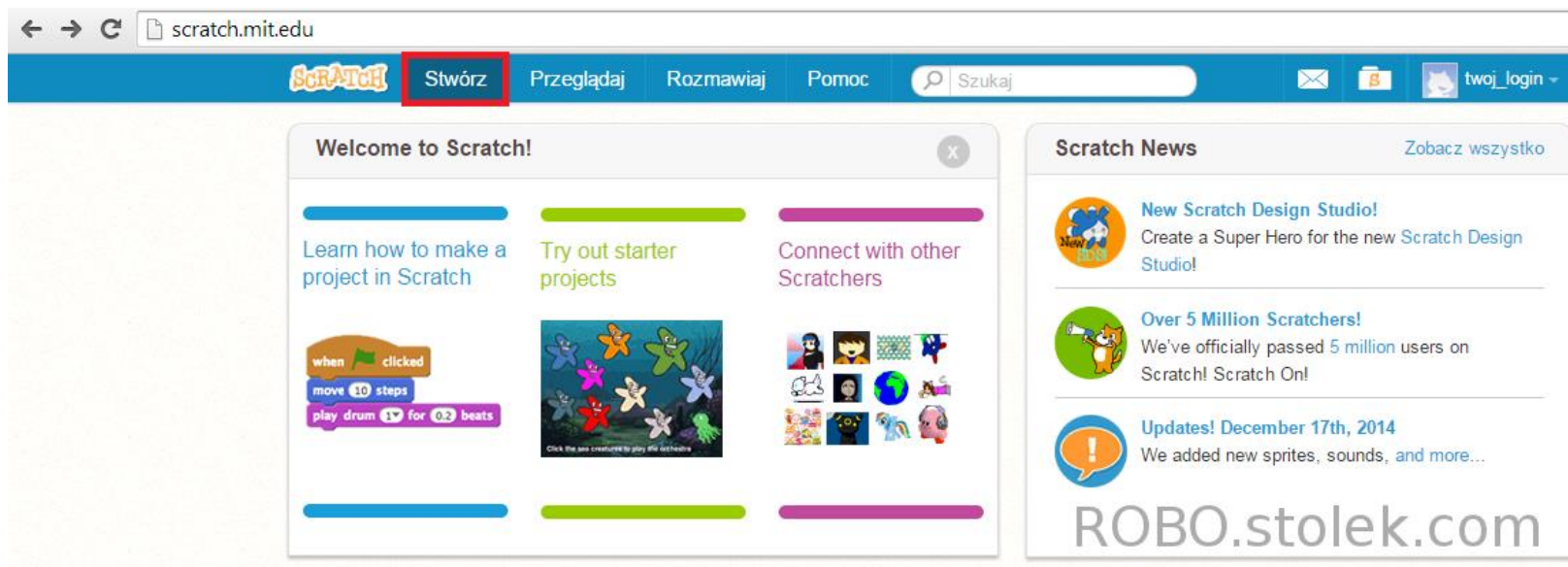
3

OK Lets Go!

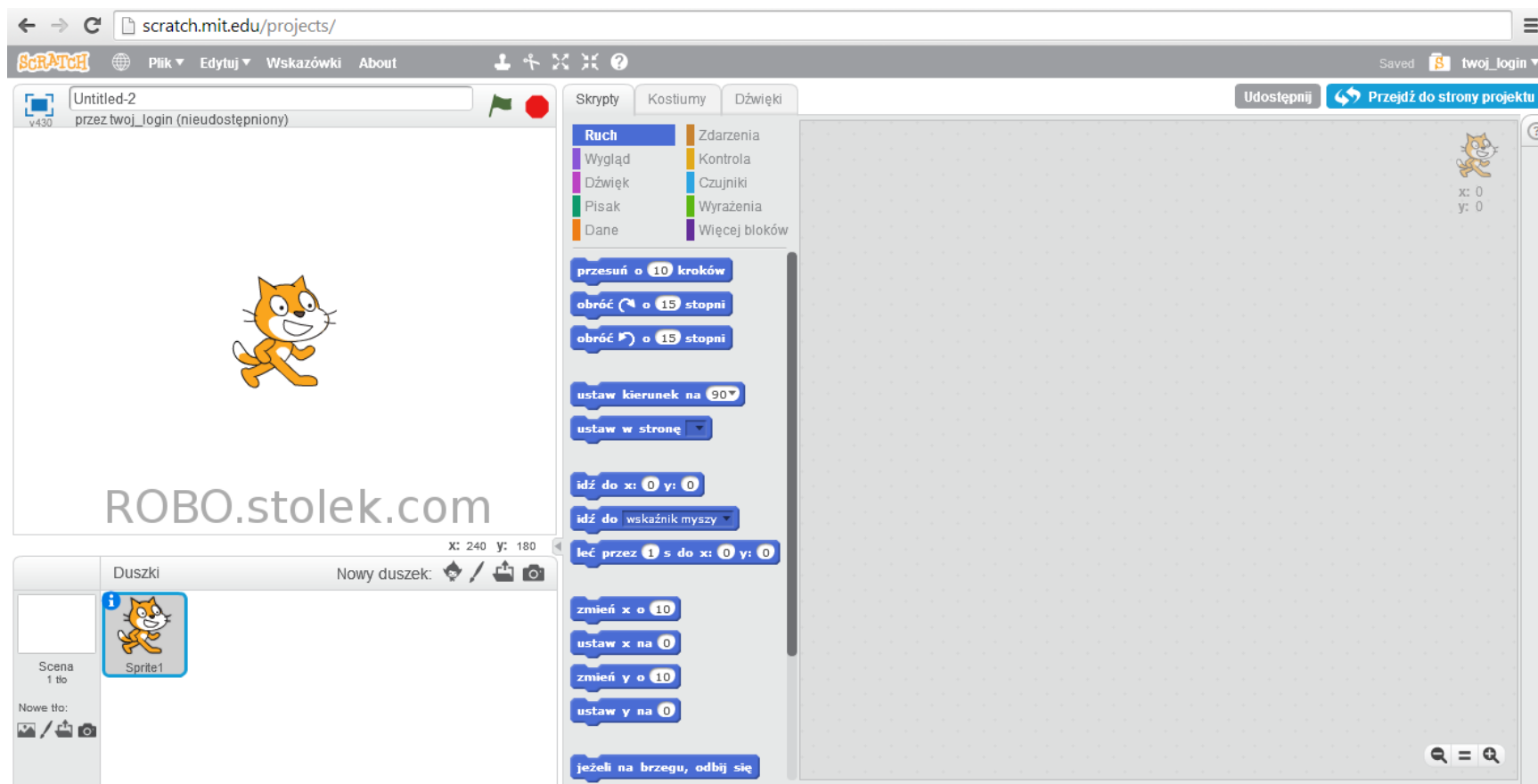
Jako zalogowany użytkownik możesz tworzyć programy, zapisywać je i udostępniać. W prawym górnym rogu masz dostęp do wiadomości (ikona koperty), do zakładki „Moje rzeczy” (ekran poniżej) oraz do ustawień konta. W zakładce „Moje rzeczy” możesz zobaczyć m.in. zapisane oraz udostępnione projekty:



Tyle wstępu. Jeśli chcesz rozpocząć tworzenie nowego programu kliknij na „Stwórz”:

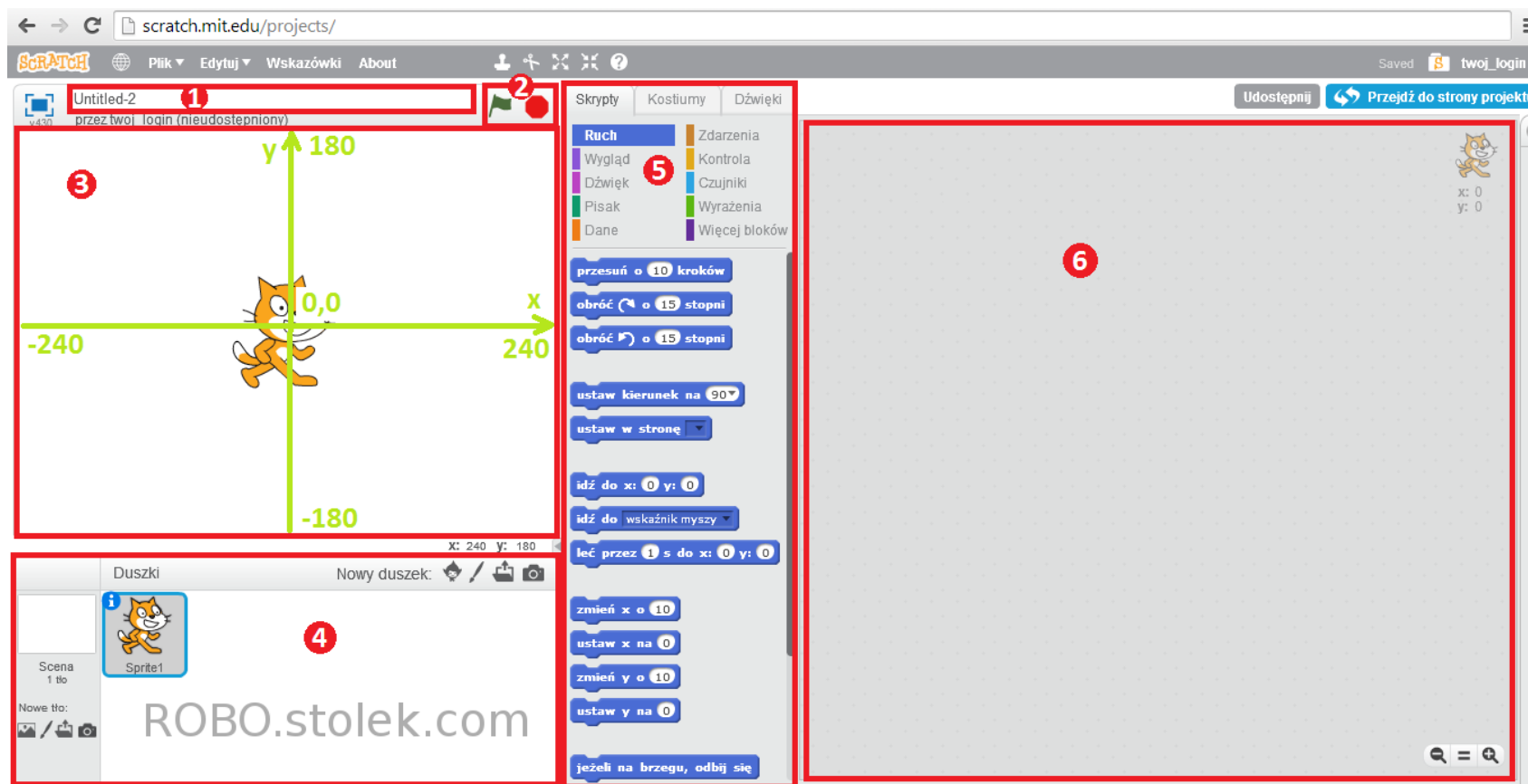


Powinien załadować się ekran podobny do tego poniżej. Wprawne oko zauważy, że ekran jest niemal identyczny jak ten w aplikacji. Właśnie w takim oknie możemy programować:



3. Zapoznaj się z rozkładem okna aplikacji

Poniżej krótkie wyjaśnienie co znajduje się na ekranie tworzenia aplikacji. Opis dotyczy zarówno edytora offline jak i wersji dostępnej przez WWW:



- 1) **Nazwa programu (projektu)**
- 2) Ikony **uruchamiania i zatrzymania** programu. Zielona flaga uruchamia program, natomiast czerwona ikona zatrzymuje go.
- 3) **Scena** (stage). To tutaj zobaczymy wizualne efekty naszego programowania. Scena to tak naprawdę plansza o dwóch osiach współrzędnych. Scena ma wymiary 481×361 punktów. Punkt (0,0) znajduje się dokładnie w środku (domyślnie tam stoi nasz duszek). Dla zobrazowania wymiarów sceny dorysowałem osie wraz z opisem. Dla przykładu: prawy górny róg ma współrzędne $x=240$ $y=180$, natomiast lewy dolny to $x=-240$ $y=-180$.
- 4) **Obszar duszków**. Tutaj zobaczymy wykorzystywane przez nas obrazki postaci (duszków).

5) W tej części dostępne są 3 zakładki:

- **Skrypty:** tutaj znajdziemy pogrupowane tematycznie bloczki, z których buduje się programy
- **Kostiumy:** każdy duszek (postać) może mieć wiele kostiumów, w tej zakładce możemy je obejrzeć, zmodyfikować i stworzyć nowe
- **Dźwięki:** miejsce, w którym przejrzymy dostępne dźwięki lub stworzymy nowe

6) **Obszar skryptów.** Miejsce, w które możemy przeciągać bloczki i łączyć je ze sobą tworząc skrypty. To właśnie te skrypty ożywią nasz program.

TWORZYMYS PIERWSZY PROGRAM!

Proponuję zacząć od ułożenia poniższego układu bloczków w swoim edytorze. W następnych krokach przeanalizujemy dokładnie jak ten program działa:






Efekt powinien być taki (program rozpocznie działanie po kliknięciu w zieloną flagę w prawym górnym rogu):

Powyższy program znajduje się pod adresem <http://scratch.mit.edu/projects/11354572>, można go remiksować i eksperymentować do woli.


A teraz przeanalizujemy dokładnie jak ten zestaw bloczków działa. Na początek dwie zasady, które ułatwią tworzenie i analizę programów.

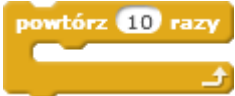


Zasada 1: każdy skrypt (czyli połączony zestaw bloczków) wykonywany jest krok po kroku, zgodnie z logiką bloczków.

Zasada 2: rozpoczęcie wykonywania danego skryptu uzależnione jest od wystąpienia zdarzenia.

Przyjrzyjmy się zatem bliżej bloczkom, z których składa się nasz program. Pierwszy bloczek  (1) mówi, że Scratch powinien rozpocząć wykonywanie kolejnych podłączonych do tego bloku instrukcji w momencie, kiedy zaszło zdarzenie kliknięcia zielonej flagi. A zatem ten skrypt reaguje tylko i wyłącznie na kliknięcie we flagę. Następnym wykonanym blokiem będzie  (2). Ten blok ustawia duszka w pozycji x=0,y=-120 (dla przypomnienia, w [pierwszej lekcji](#) możecie zobaczyć jak wygląda plansza). Kolejnym wykonanym zadaniem będzie  (3) – celem tego zadania jest obrócenie duszka w stronę ruchu (w prawo).

Idąc dalej:  (4) wyświetla „dymek” z podanym przez nas tekstem przez określony czas. Kolejny blok:  (5) odtworzy dźwięk.

Następny blok  (6) jest bardzo ciekawy. Pozwala on na ciągłe powtarzanie bloczków, które są w nim zawarte (przetwarza je po kolei, a po zakończeniu wraca do pierwszego i od nowa je przetwarza). W naszym przypadku, kolejno przetwarzane będą 3 bloki: „powtórz x razy”, „zagraj bęben” i „następny kostium”.


Blok  (7) powtórzy odpowiednią liczbę razy bloki w nim zawarte. W naszym przypadku program 10 razy wykona parę poleceń: przesunięcie o 2 kroki do przodu  (8) i obrót w lewo o 1 stopień  (9). Po wykonaniu zdefiniowanej liczby powtórzeń program przejdzie do kolejnego bloku (tego, który jest przypięty poniżej bloku „powtórz x razy”).

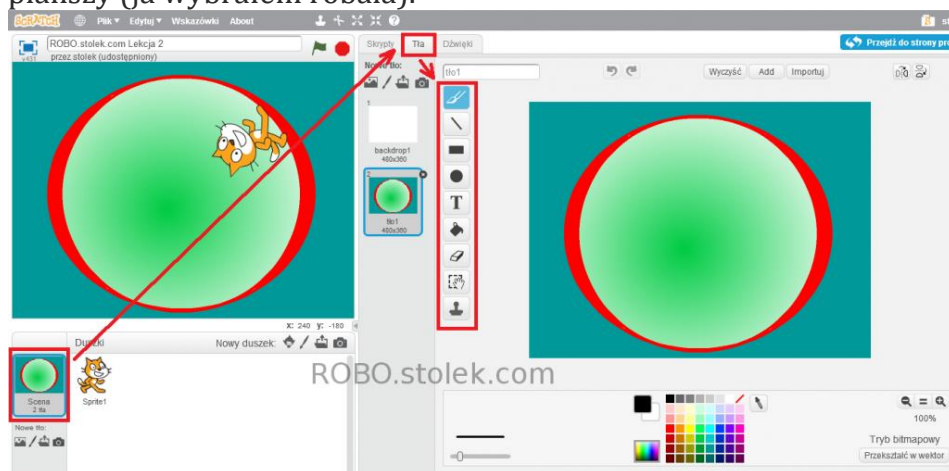
Blok **zagraj bęben 97 przez 0.1 taktów** (10) odgrywa dźwięk bębna przez określoną ilość taktów.

Ostatni element to **następny kostium** (11). Po wykonaniu tego bloku nasz duszek będzie ubrany w kolejny kostium.

UCZYMY ROBAKA CHODZIĆ!

Wracamy i otwieramy poprzednie zadanie.

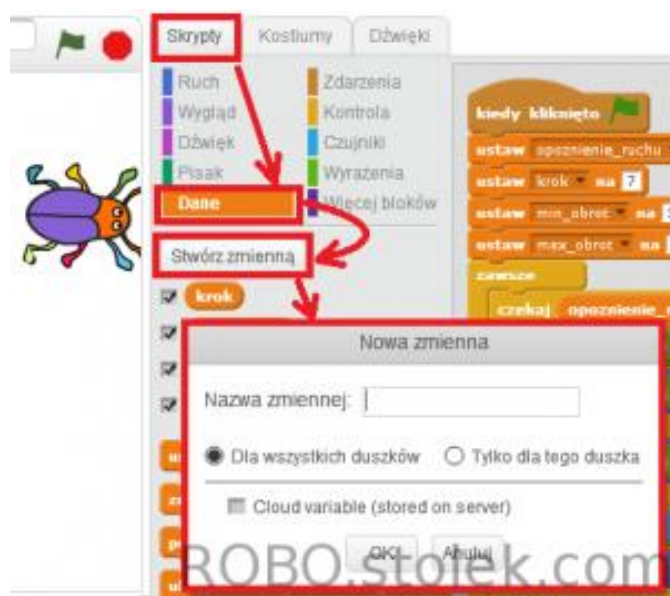
Pierwsza rzecz, jaką trzeba zrobić, to **zmiana postaci** (duszka). Robimy to klikając w obszarze duszków (lewy dolny róg ekranu tworzenia programu) w przycisk Nowy duszek: „Wybierz duszka z biblioteki” . Wybieramy postać, którą nauczymy chodzić po planszy (ja wybrałem robala).



Po wybraniu duszka (i ewentualnie tła) możemy chwilę się zastanowić jak chcemy zrealizować ruch. Ja chcę to zrobić następująco: w każdym cyklu robak powinien obracać się losowo w lewo lub prawo i robić krok do przodu. Pomiędzy cyklami robak powinien nie robić nic poza czekaniem (przez pewien ustalony czas). W chwili wymyślenia programu nie wiem dokładnie jak długi powinien być krok

robaka, w jakim zakresie powinien się obracać (o ile stopni) i jak długo powinien czekać pomiędzy cyklami. Zakładam, że trochę poeksperymentuję z tymi parametrami. I tu z pomocą przychodzą zmienne.




Co to są zmienne? To pewne wartości, które możemy zmieniać, a są dostępne pod zdefiniowanymi nazwami 😊 Już wyjaśniam o co chodzi. Zaczniemy od stworzenia zmiennych, klikamy kolejno jak na obrazku







a następnie wpisujemy nazwę zmiennej, którą chcemy stworzyć (czynności powtarzamy dla każdej zmiennej). W moim przypadku tworzę następujące zmienne:

- opoznienie_ruchu – opóźnienie czasowe pomiędzy kolejnymi cyklami ruchu (krokami),
- krok – o ile robak przesunie się w jednym kroku do przodu,
- min_obrot – minimalny kąt obrotu robaka w każdym cyklu ruchu,
- max_obrot – maksymalny kąt obrotu.

Już po stworzeniu pierwszej zmiennej pojawiają się w zakładce „Skrypty” -> „Dane” bloki, dzięki którym możemy:

- **odczytać** wartość zmiennej  ,
- **ustawić** wartość zmiennej ,
- i **zmienić** wartość zmiennej o zadaną wartość .

Dodatkowo możemy sterować wyświetlaniem wartości zmiennych za pomocą przycisków  i . Wyświetlić i ukryć wartość zmiennej na ekranie można również za pomocą checkboxa przy zmiennej ( ).

Jeśli teraz prześledzimy [program](#) zobaczymy, że na samym początku (po kliknięciu w zieloną flagę) ustawiane są wartości początkowe wszystkich stworzonych zmiennych.

Dalej widzimy pętlę „zawsze”, którą już znacie z poprzednich lekcji. W pętli pierwszym blokiem jest „czekaj opóźnienie_ruchu s”.

Wstrzymuje on wykonanie skryptu przez określoną ilość sekund, w naszym przypadku zgodną z wartością zmiennej opóźnienie_ruchu.

Przyjrzyjmy się dokładniej kolejnemu blokowi, jego zadaniem jest obrót robaka.



Mamy tam duży blok „jeżeli _ to _ w przeciwnym razie _”, którego zadaniem jest podjęcie decyzji czy robakiem skręcamy w lewo czy w prawo. Decyzja opiera się na elemencie (pseudo)losowości. Warunkiem funkcji jeżeli jest porównanie czy wylosowana liczba jest równa 1. Jeżeli jest równa 1 wtedy obracamy duszka w prawo, w przeciwnym razie (czyli jeśli wylosowana jest liczba 2) skręcamy w lewo.

I teraz zobaczymy jak zrealizowany jest sam obrót (bloki „obróć o ...”). Tam również losujemy, tym razem o jaki kąt ma się obrócić robak. I tutaj powinniśmy docenić zastosowanie zmiennych – w obydwu przypadkach zakres losowania określony jest przez zmienne. Zatem w przypadku eksperymentowania z kątami obrotu podajemy tylko jako nowe wartości początkowe zmiennych, a nie nowe wartości

ręcznie do każdego z bloków. Podsumowując tę część: duszka obracamy losowo w lewo lub w prawo o losowy kąt z zakresu [min_obrot; max_obrot].

Idąc dalej mamy taki fragment:



Jego zadaniem jest poruszanie robakiem (wykonywanie kroków), ale w taki sposób, żeby nie wypaść poza planszę. Warunkiem sprawdzanym przez blok „jeżeli” jest czy duszek „nie dotyka krawędzi”. Jeżeli nie dotyka, wtedy robak przesuwa się do przodu o ilość kroków zdefiniowaną przez zmienną „krok”. Jeżeli dotyka krawędzi, to cofa się o dwu krotność wartości zmiennej „krok” (tak sobie wymyśliłem – ma to być efekt odbijania się od krawędzi). Jeżeli okaże się, że po cofnięciu duszek nadal jest na brzegu planszy – odbija się od krawędzi (blok „jeżeli na brzegu, odbij się”). Zabezpieczamy się w ten sposób przed wychodzeniem duszka poza krawędź planszy.

Kropką nad i jest zmiana kostiumu na następny – powinniśmy przygotować taki/takie, które sprawiają wrażenie poruszania się naszej postaci (zakładka „kostiumy”).

Dotarliśmy razem do końca. Czekam na informacje od Was czy wszystko było jasne, ewentualnie co dokładniej wytłumaczyć – piszcie w komentarzach.