

# SZTUKA I ALGORYTMY

**Katharine Child** inspirując się artystą Sol LeWitem tworzy z uczniami, podczas lekcji informatyki w szkole podstawowej, prace graficzne oparte na algorytmach



Ściana pomalowana przez Sol LeWitta, w Spoleto's Carandente Museum, była inspiracją do tworzenia z dziećmi algorytmów połączonych z tworzeniem grafiki.  
Zdjęcie: commons.wikimedia.org

Na pierwszy rzut oka informatyka i sztuka to mało pasujące do siebie połączenie. Informatyka kojarzy się z czymś precyzyjnym, zaplanowanym i przewidywalnym, sztuka z kolei z wolnością, ekspresją, kreatywnością.. Sol LeWitt (1928 - 2007) był amerykańskim artystą, który dowodził, że instrukcje tworzenia są równie ważne jak samo dzieło sztuki. Udostępnił on swoje tabele i diagramy, aby inni mogli z nich korzystać i odtwarzać na ich podstawie grafiki i instalacje artystyczne.

Bazując na tym podejściu dostosowałam instrukcje, oparte na zapisach LeWitta, do nauczania informatyki na poziomie szkoły podstawowej.

Pozwalają one na wykorzystanie kolorowych pasków i tworzenie przez dzieci algorytmów w celu narysowania pasiastych obrazów. Algorytm jest to sekwencja instrukcji lub zestaw reguł, pozwalających osiągnąć założony efekt. W naszym przypadku są to dzieła zainspirowane sztuką LeWitta. Po napisaniu algorytmów dzieci pracują w parach. Jedno dziecko pełni rolę pilota, który czyta instrukcje, a drugie dziecko jako robot wykonuje polecenia zgodnie z instrukcją dziecka-pilota. Warto zwrócić uwagę, że LeWitt w niektórych instrukcjach był celowo niejednoznaczny, podczas gdy w klasie dzieci będą musiały być precyzyjne. Dzięki temu powstaną algorytmy możliwe do wykonania przez komputer lub dzieci wcielające się w role robotów.



**KATHARINE  
CHILDS**

Katharine (@IAmKatharineC)  
Koordynator Programu Fundacji  
Raspberry Pi, nauczyciel w  
szkole podstawowej.

## Kamienie milowe

Realizację pomysłu dobrze rozpocząć od pracy offline (bez komputera) z wykorzystaniem materiałów plastycznych umożliwiających tworzenie kolorowych pasków. Dobrze sprawdzają się malowanie na dużych arkuszach papieru, przymocowanych taśmą do podłogi lub kredą na po asfalcie na placu szkolnym.

Podobnie jak w pracach LeWitta, twórcza przestrzeń musi mieć zaznaczone granice. Dzieci mogą wtedy użyć precyzyjnego języka wskazującego położenie kolejnych pasków. Na początku najłatwiej korzystać z poziomych lub pionowych linii, z wykorzystaniem dwóch kolorów, lub kilku barw powtarzających się w sekwencji.

Zadaniem dzieci jest zaplanowanie w dokładny sposób, kiedy mazak lub kreda mają zostać podniesione, a kiedy opuszczone na papier lub asfalt. W czasie drugiego etapu możemy przejść do wykorzystania algorytmu i stworzenia skryptu w Scratchu. To darmowe środowisko pracy dostępne na stronie [scratch.mit.edu](http://scratch.mit.edu). To świetne narzędzie, aby wykorzystać doświadczenie zgromadzone w pracy offline i użyć go do pracy koncepcyjnej nad pisaniem poprawnego programu. Scratch posiada wbudowany edytor graficzny, jednak my chcemy, aby to odpowiednio zaprogramowany duży rysował paski na ekranie. Będzie to wymagało od dzieci dokładnych obliczeń określających pozycję początkową i końcową każdej linii. Uczniowie będą mogli też skorzystać ze zmiennych, aby aktualizować wartość tych pozycji. Jak sekwencja staje się rozbudowana dzieci mogą zauważyć, że wielokrotnie wracają do pozycji wyjściowej: podnieś pióro, wróć do pozycji startowej, przyłóż pióro, narysuj pasek, podnieś pióro, a następnie zmień współrzędne i kolor pióra dla następnej linii. Dostrzeganie tego wzoru pomaga określić, którą część sekwencji umieścić w pętli celem jej powtórzenia.

## Główne trudności

Podczas pisania algorytmów uczniowie często nie rozumieją jak bardzo precyzyjny musi on być, aby został prawidłowo wykonany przez komputer. Te problemy najczęściej wynikają z:

1. *Pominięcia kroków.* Łatwo można założyć, co jest nam potrzebne do stworzenia obrazu lub grafiki. Dlatego dzieci dość często zapominają wyraźnie zaznaczyć kiedy należy podnieść, a kiedy przyłożyć pisak. Praca w parach jest dobrym sposobem na radzenie sobie z tą trudnością. Wspólne debugowanie i doskonalenie algorytmu i skryptu daje lepsze rezultaty.

2. *Braku szczegółów.* Dzieci często używają w algorytmach niejednoznacznych komunikatów. Dodanie sformułowań, które konkretnie wskazują pozycję pisaka może pomóc uczynić algorytm bardziej precyzyjnym. w etapie pierwszym (offline) można do tego użyć takich słów jak: lewo, prawo, pod, dotyka, itp. W etapie drugim (Scratch) można dokonać precyzyjnych obliczeń. Możliwość wymiany przez dzieci rozwiązań między sobą jest dobrym sposobem na poprawę własnego projektu.

Mozesz pomóc uczniom tworzyć precyzyjne algorytmy poprzez poniżej opisane działania. Użytecznym sposobem jest dzielenie zadania na małe kroki i podejmowanie decyzji w jakiej kolejności należy je wykonać. Uczniowie mogą napisać pierwszą część algorytmu i testować ją, sprawdzając jego poprawność, modyfikować to, co zrobili do tej pory i udoskonalać swój projekt. Cykl "napisz-testuj-popraw" jest dobrą praktyką, która powinna stać się rytuałem w tej formie pracy. Można go powiesić w widocznym miejscu w sali zajęć. W czasie pracy można się spodziewać reakcji w postaci: "mam to!", ale równie często: "Nie to miałem na myśli!". Tworzenie warunków do nauki dzięki spotykanym trudnościom i niepowodzeniom buduje przestrzeń do tego, aby błędy nie były problemem same w sobie, tak długo jak nad nimi pracujemy i poszukujemy rozwiązań. Prezentacja rezultatu jest równie ważna jak możliwość zobaczenia przez uczniów błędów, trudności i sposobów debugowania przez które przeszli ich rówieśnicy.

To dodający wiary komunikat wskazujący, że proces "pomyłka - uczenie się" jest czymś naturalnym.

Świetnym miejscem do doświadczenia tego jest współpraca dzieci w parach i wspólna praca nad algorytmem. Niektóre dzieci mogą potrzebować wsparcia, aby zrozumieć że w roli pilotów mają pomóc dzieciom w roli robotów pracować zgodnie ze słyszaną instrukcją, zamiast samodzielnie dodawać ruchy lub inne detale. Na koniec współpracy wytworzona grafika i algorytm są uzupełniającą się całością.

## WIĘCEJ INFORMACJI



Więcej informacji na temat Sol LeWitta:  
[helloworld.cc/Tate\\_LeWitt](http://helloworld.cc/Tate_LeWitt)

Niedokończona wersja projektu w Scratchu, bazująca na pracy LeWitta:  
[helloworld.cc/LeWitt1](http://helloworld.cc/LeWitt1)

Dokończony wersja projektu w Scratchu, bazująca na pracy LeWitta:  
[helloworld.cc/LeWitt2](http://helloworld.cc/LeWitt2)

tłumaczenie na język polski: Krzysztof Jaworski  
[makewonder.pl](http://makewonder.pl)