

Drogi rodzicu!

W obecnym czasie, gdy zamknięte są szkoły, przedszkola, żłobki mamy dla Ciebie propozycję kreatywnej zabawy dla Twoich dzieci.

Sami zajmujemy się edukacją dzieci w zakresie algorytmiki, robotyki i programowania oraz elektroniki i automatyki i z tych dziedzin właśnie przygotowaliśmy edukacyjne zabawy.

W tej chwili udostępniamy Ci zabawy dla dzieci do klasy czwartej, w przyszłym tygodniu udostępnimy i dla starszych. Udostępnimy również materiały wideo.

Wszystkie zabawy zawarte w tym pdf'ie możecie bez ograniczeń wykorzystywać do celów prywatnych (domowych).

Bawcie się dobrze.

Pamiętajcie #zostańwdomu #toproste .

Z poważaniem,
Zespół RoboBlocks.pl

Ćwiczenia 'unplugged'

Kodowanie 0-1

Do przeprowadzenia ćwiczenia potrzebna będzie kartka, podzielona na równe, niezbyt duże kwadraty (może być wyrwana z zeszytu lub notesu w kratkę), wielkości 10x10 pól. Zadaniem dziecka będzie narysować obrazek zgodnie z kodem: każda nowa linia kodu to nowa linijka na kartce, "0" to brak koloru, "1" - zamalowane pole. Kolejne linijki kodu należy dzieciom czytać (można też przewijać na smartfonie, tablecie, zapisywać lub drukować- dobrze byłoby, żeby dziecko nie widziało całego kodu jednocześnie), sprawdzając czy kolorują właściwą kratkę.

Wyjaśnienie: w ten sposób poznajemy zapis binary (dwucyfrowy), stosowany w informatyce. "1" oznacza "włączony", "0" wyłączony, włączony przybiera kolor odpowiadający kolorowi kredki jaką dziecko maluje. Z jego pomocą, można zaprogramować np. sygnały błyskowe, wysyłane przez urządzenia do komunikacji światłem, język Morse'a itp. Przykładowe obrazki:

Rysunek 1:

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0
0 0 1 0 0 0 0 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Rysunek 2:

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 1 0
0 0 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0
```

Rysunek 3:

```
0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 1 0 1 0 0
0 0 0 0 1 1 1 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 1 1 0 1 1 0 0
0 0 0 1 1 0 1 1 0 0
```

Ćwiczenie to można rozbudować do 16 kolorów (to bardzo podstawowa paleta kolorów używana początkowo w komputerach). Dzięki temu mamy możliwość pokolorować bardziej skomplikowane obrazki. Rodzic ustala jaka cyfra i literka odpowiada jakiemu kolorowi. Przykład:

0 - czarny, 1 - bordowy, 2 - czerwony, 3 - biały

Rysunek jest o wymiarach 9x9:

```
3 3 0 3 3 3 0 3 3
3 0 2 0 3 0 2 0 3
0 2 2 2 0 2 3 2 0
0 1 2 2 2 2 3 2 0
0 1 2 2 2 2 2 2 0
3 0 1 2 2 2 2 0 3
3 3 0 1 2 2 0 3 3
3 3 3 0 1 0 3 3 3
3 3 3 3 0 3 3 3 3
```

Możecie Państwu pokazać dziecku rysunek i poprosić je o stworzenie kodu.

Sami możecie stworzyć proste pixelarty korzystając ze strony

<https://www.pixilart.com/draw>

Programowanie postaci z wyścigiem

Do ćwiczenia potrzeba będzie plansza, podzielona na 64 pola, dwie figurki (idealnie nada się szachownica i szachy), dwie kartki papieru oraz pisaki. Jedno z pól należy oznaczyć w specjalny sposób, kładąc na nim np: monetę, kolorowy guzik. Figurę umieszczamy w rogach szachownicy. Zadaniem graczy będzie zakodowanie ruchu postaci w taki sposób, aby dotarły do wyznaczonego pola. Zasady dotyczące języka i poruszania się, podane są niżej. Ważne, aby uczestnicy nie mogli podglądać swoich kodów w czasie ich pisania. Gdy kod jest skończony, oznajmia swoją gotowość podnosząc rękę do góry. Gracze wykonują swoje kody w takiej kolejności, jak skończyli je pisać na kartkach. Jeżeli kod był wykonany poprawnie, gracz zdobywa punkt. Jeżeli nie, drugi gracz ma prawo wykonać swój kod. Gdy jest poprawny, od zdobywa punkt. Jeżeli żaden z kodów nie był wykonany dobrze, nikt nie zdobywa punktu. Po skończonej rundzie (kiedy któryś z kodów był poprawny lub żaden z kodów nie okazał się trafiony), zmieniamy pozycje dla celu i postaci, przenosząc je na inne pola na planszy. Można grać do 3 rund lub gdy któremuś z graczy uda się zdobyć 3 punkty. Poziom trudności można zwiększać, przenosząc się na większą planszę lub dodając na niej przeszkody z innych figurek, które trzeba ominąć, ołówków, długopisów, tworzących niby-ściany w labiryncie itd.

Język poruszania się- zasady

↑ Strzała w górę - idź prosto

↻ Strzałka skręcająca w prawo - obróć się w prawą stronę (nie mylić ze - skręć i idź)

↻ Strzałka skręcająca w lewo - obróć się w lewą stronę (nie mylić ze - skręć i idź)

↓ Strzałka w dół - idź do tyłu.

Dzieciom uczestniczącym w zajęciach z "Algorytmiki" lub uczestniczącym w półkoloniach można przypomnieć programowanie myszy, Cezarego Piskalskiego. Poruszanie się działa u niego na tej samej zasadzie.

Odszyfrowywanie wiadomości

Poniżej zaszyfrowano kilka haseł. Aby je odczytać, należy odszyfrować je, wedle prostego kodu- każda cyfra przyporządkowana jest kolejnej literze alfabetu. 1 = A, 2 = B, 3 = C itd. (cały rozkład poniżej). Dzieci muszą odszyfrować hasła.

A	1
A	2
B	3
C	4
Ć	5
D	6
E	7
Ę	8
F	9
G	10
H	11
I	12
J	13

K	14
L	15
Ł	16
M	17
N	18
Ń	19
O	20
Ó	21
P	22
R	23
S	24
Ś	25
T	26
U	27
W	28
Y	29
Z	30
Ż	31
Ź	32

Hasła(litery oddzielone są spacją, słowa - myślnikiem)

6 30 12 7 19 - 6 20 3 23 29

12 6 30 12 7 - 28 12 20 24 18 1

22 20 2 20 26 29 14 1

14 20 6 20 28 1 18 12 7

Algorytm robienia kanapki

Dla dzieci starszych, potrafiących pisać. Ciekawy sposób na urozmaicenie przyrządzania posiłku. Aby otrzymać kanapkę z wybranymi składnikami, muszą zapisać czynności, składające się na zrobienie kanapki. Czynności muszą być ułożone w logicznej kolejności i składać się z bardzo konkretnych poleceń (kod "Zrób kanapkę" odpada) Np.:

- podejść do lodówki
 - otwórz lodówkę
 - znajdź słoik z dżemem
 - wyciągnij słoik z dżemem z lodówki
 - zamknij lodówkę
 - połóż słoik na stole..
- itd.

Dziecko będzie mogło poprawiać algorytm ile razy będzie chciało, aż kanapka będzie gotowa. Stopień skomplikowania i dosłowności zależy od osobistych ustaleń z dzieckiem. Jednocześnie należy pamiętać, że jeżeli wcześniej części programu wykonane były poprawnie, nie zmieniamy ich znaczenia w trakcie kolejnej próby. Przykładowo, jeżeli wcześniejsza wersja pomijała otwarcie lodówki, ale my i tak to zrobiliśmy, uznajemy się za mądrego robota i w następnej próbie robimy to samo. Robienie kanapki jest tylko przykładem, można je zamienić na mycie zębów czy też rąk.

Ołówkowy obwód

Kolejna aktywność to nie tyle ćwiczenie czy zadanie do wykonania, co eksperyment z elektrycznością. Najlepiej wykonywać go wieczorem, gdy można zgasić światło. Potrzebna będzie kartka papieru, ołówek, dioda LED, bateria, najlepiej 9V, taśma izolacyjna lub najzwyczajniejsza taśma klejąca, przewód zasilający na baterię 9V *. Ołówkiem rysujemy dwie linie, dość grube. Nie mogą się przecinać ani łączyć ze sobą. Do jednego końca linii przyklejamy diodę, jedną "nóżką" do jednej linii, drugą do drugiej. Następnie do baterii przyczepiamy przewód zasilający, jeden kabel przykładamy do jednej linii, drugi do drugiej. Dioda powinna zacząć lekko świecić (jeżeli tak się nie dzieje, trzeba zamienić kable miejscami).

*Niekiedy udaje się wykonać eksperyment bez tego elementu, jedynie przykładając baterię do linii lub zamiast rysować linie ołówkiem można je wykonać z przewodzącego prąd materiału, jak miedziana taśma samoprzylepna.

1. Domowe konstrukcje

Poniżej znajdują się linki do projektów, które można wykonać domowymi lub łatwo dostępnymi materiałami. Zachęcamy do spróbowania wykonania ich wspólnie z podopiecznymi lub przygotowania dla nich materiałów. W większości przypadków klej na gorąco można zamienić taśmą samoprzylepną, izolacyjną albo typu duck tape.

<https://www.youtube.com/watch?v=C9IRZ4QYXg8>

<https://www.youtube.com/watch?v=7Vb92uRhDpU>

<https://www.youtube.com/watch?v=IK9a6SuitiU>

<https://www.youtube.com/watch?v=ybFy-zyLYco>

2. Kodowanie online

Poniżej zamieszczamy linki do kilku stron, gdzie znajdują się uczące programowania gry. Z racji na używanie języka podobnego do Scratcha, rekomendujemy je dla uczestników kursu Klasa 4 - Scratch + WeDo

<https://blockly.games/?lang=pl>

<https://studio.code.org/s/aquatic/stage/1/puzzle/1>