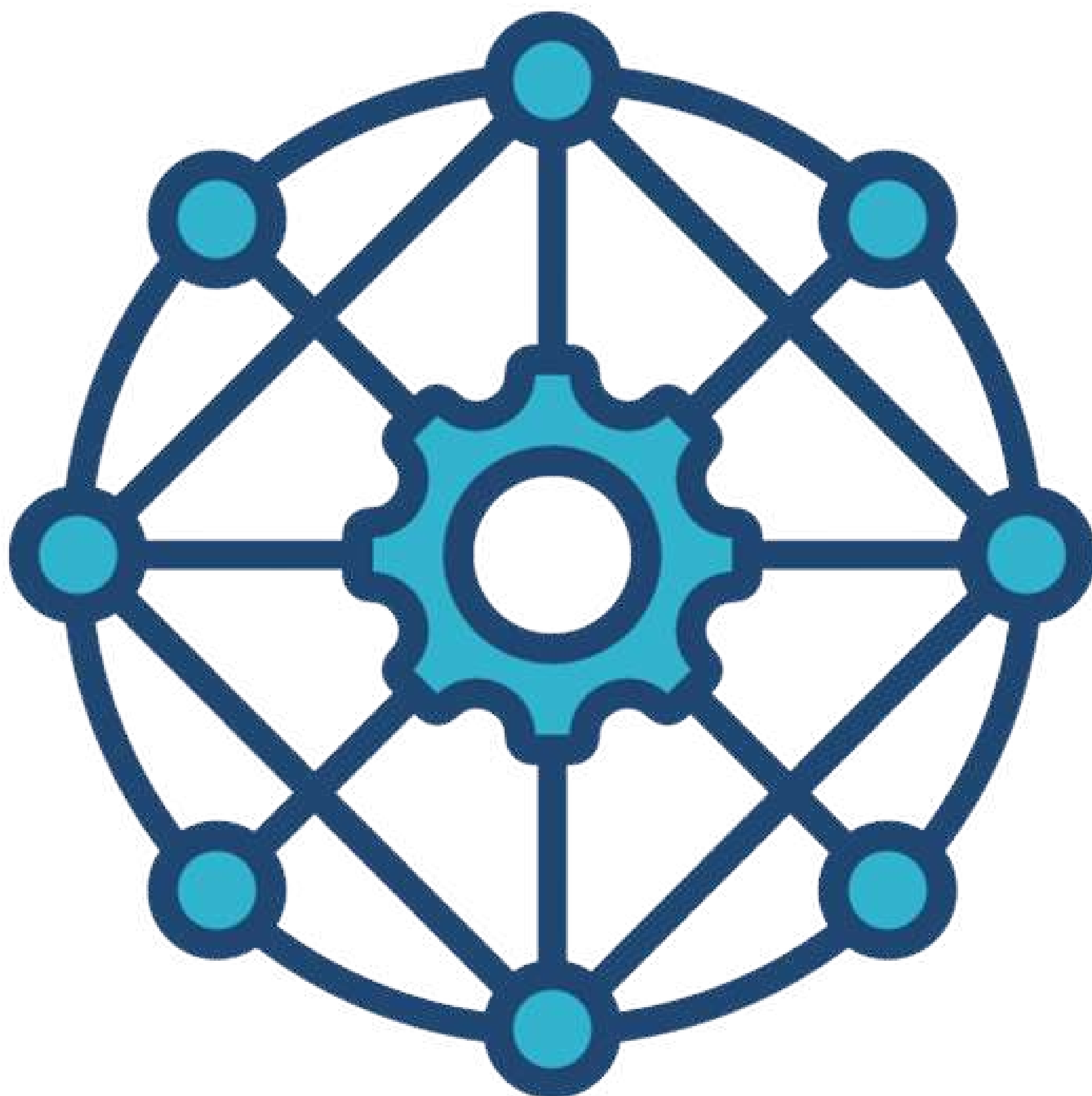


ALGORYTM PRZYSZŁOŚCI

SCENARIUSZE ZAJĘĆ Z ROBOTYKI I PROGRAMOWANIA
DLA NAUCZYCIELI KLAS 1-3

MARIOLA KINAL



STOWARZYSZENIE NAUKOWE PRZESTRZEŃ SPOŁECZNA I ŚRODOWISKO

Redaktor: dr Mariola Kinal

Opracowanie graficzne: Mariola Kinal

Skład, łamanie i rysunki: Mariola Kinal, Anna Świć

Opracowanie okładki: Mariola Kinal

Opracowanie scenariuszy zajęć: Anna Świć – kodowanienadywanie.pl, Mariola Kinal

Recenzja naukowa: dr Katarzyna Myśliwiec

© Stowarzyszenie Naukowe Przestrzeń Społeczna i Środowisko

Ta książka jest dostępna na zasadach otwartego dostępu. Można ją kopiować, udostępniać i wykorzystywać do celów edukacyjnych i niekomercyjnych zgodnie z licencją [Creative Commons – Na tych samych warunkach 4.0 Międzynarodowe (CC BY-SA 4.0)]. Wykorzystanie treści w celach komercyjnych lub w sposób sprzeczny z warunkami licencji jest zabronione. Autorskie prawa do treści książki pozostają przy autorze/autorach. Kopiowanie, publikowanie lub rozpowszechnianie tej książki jest dozwolone jedynie w ramach określonych w warunkach licencji. Zmiany w treści, w tym adaptacje, mogą być dokonywane, pod warunkiem odpowiedniego przypisania autorstwa i stosowania tej samej licencji.

Projekt finansowany ze środków budżetu państwa, przyznanych przez Ministra Edukacji i Nauki w ramach Programu “Społeczna Odpowiedzialność Nauki II”. Projekt “Algorytm przyszłości – robotyka i programowanie dla szkół podstawowych”, ID POPUL/SN/0316/2023/01.

Egzemplarz bezpłatny

ISBN: 978-83-64879-05-0

wydanie I

Rzeszów 2025

O AUTORKACH



Anna Świć – nauczycielka, logopedka, ekspertką edukacyjną, trenerka i miłośniczka wykorzystywania nowoczesnych technologii w pracy dydaktycznej. Od kilku lat zajmuje się wprowadzaniem elementów kodowania, programowania i robotyki do zajęć w edukacji przedszkolnej i wczesnoszkolnej. Stworzyła i koordynowała Ogólnopolski Program „Uczymy dzieci programować” oraz Ogólnopolski Program „Kodowanie na dywanie”.



Mariola Kinal – doktor nauk społecznych w dyscyplinie socjologia, absolwentka studiów magisterskich na kierunku pedagogika przedszkolna i wczesnoszkolna, adiunkt w Instytucie Pedagogiki Uniwersytetu Rzeszowskiego, sekretarz Stowarzyszenia Naukowego Przestrzeń Społeczna i Środowisko, autorka czterech monografii i kilkunastu artykułów naukowych, koordynatorka projektów naukowych i popularyzatorskich takich jak: “Child Research and Education Congress”, “Algorytm przyszłości – robotyka i programowanie dla szkół podstawowych”, “Polska pedagogika przedszkolna i wczesnoszkolna i jej twórcy – wystawa interaktywna”.

SPIS TREŚCI

1. Słowem wstępu, czyli co każdy nauczyciel chce wiedzieć	4
2. Gdybym był_a programist_ką	13
3. 5, 10, 15... zaprogramowane zegary	22
4. Cztery pory roku	35
5. Kosmiczna wyprawa	53
6. Liczby parzyste i nieparzyste	72
7. Poznajemy lektury z programowaniem w tle	83
8. Ruch drogowy bezpiecznie zaprogramowany	96
9. Zaprogramowana ortografia	108
10. Zaprogramowane emocje	136
11. Zaprogramowani na matematykę	146
12. Mała checklista nauczyciela	162
13. Zaplanuj zajęcia!	163
14. Twoje notatki	164

SŁOWEM WSTĘPU, CZYLI CO KAŻDY NAUCZYCIEL CHCE WIEDZIEĆ...

We don't teach kids to code because we expect them to become professional computer scientists. We teach them to code so they can learn to think creatively, reason systematically, and work collaboratively.
Mitchel Resnick¹

JAK ZBUDOWANA JEST PUBLIKACJA?

Drogi nauczycielu_ko, skrypt który czytasz powstał po to, żeby pokazać Ci jak wprowadzać dzieci w świat algorytmów i logicznego myślenia.

Jak zbudowana jest publikacja?

Skrypt zawiera dziesięć scenariuszy zajęć dla uczniów klas 1–3 szkoły podstawowej. Traktuj te scenariusze jako plan, który można zmieniać – możesz rozwijać treści merytoryczne, zmieniać formę pracy, dzielić uczniów na mniejsze i większe grupy w zależności od Waszych potrzeb i możliwości. Każdy scenariusz zaczyna się od wprowadzenia dzieci w opracowywany temat pytaniem kluczowym, zagadką do rozwiązania lub inną formą, która zgodnie ze strategią innowacyjno-percepcyjną R. Więckowskiego zachęci dzieci do aktywnego zaangażowania się w omawianą tematykę. Podczas zajęć uczniowie rozwiązują problemy, sprawdzają stawiane przez siebie hipotezy – pozwól im błądzić, odkrywać błędy, które popełniają i ciesz się z ich satysfakcji, kiedy uda im się rozwiązać problem! Każde zajęcia kończą się ewaluacją – porozmawiaj z uczniami o tym, co zrozumieli, co im się podobało, nad czym jeszcze muszą popracować. Umiejętność oceny procesu, który się przeszło jest równie ważna jak sam proces, nie zapominaj więc o tym. Aby Ci to ułatwić proponujemy ciekawe dla uczniów formy ewaluacji.

Czy wiesz że neurodydaktycy² wskazują cztery filary skutecznego uczenia się: uwaga, aktywne zaangażowanie, informacja zwrotna i konsolidacja. Dzięki podaniu celów lekcji uczniowie wiedzą do czego zmierzają – pozwala im to na zaangażowanie się w proces uczenia się. Popełnianie błędów i naprawianie ich pobudza ciekawość i daje satysfakcję – to wzmacnia motywację wewnętrzną do wysiłku związanego z uczeniem się, a odkrycie informacji uruchamia obwód dopaminowy w mózgu, związany z funkcją nagrody. Informacja zwrotna od nauczyciela lub kolegi_żanki z klasy pokazuje co uczeń robi dobrze, co ma poprawić i w jaki sposób to zrobić. A żeby uczniowie nie tracili ciekawości muszą zdawać sobie sprawę z tego, czego jeszcze nie wiedzą – muszą dysponować zdolnościami metapoznawczymi. Jak ich tego nauczyć? Wprowadzając systematyczną ewaluację zajęć. Warto pamiętać – ewaluacja to nie sprawdzenie wiedzy ucznia, a refleksja ucznia nad jego drogą do realizacji celu zajęć.

[1] M. Resnick, Let's teach kids to code. TED talk, 2013. https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code [dostęp: 22 kwietnia 2025] [Nie uczymy dzieci kodowania, żeby zostały programistami. Kiedy uczymy ich programować, pozwalamy im myśleć kreatywnie, logicznie rozumować i współpracować ze sobą.].

[2] S. Dehaene, Jak się uczymy? Dlaczego mózgi uczą się lepiej niż komputery... jak dotąd. Kraków, 2022.

PODSTAWOWE POJĘCIA, KTÓRE CHCESZ ZNAĆ, ŻEBY ZACZAĆ Z UCZNIAMI PRZYGODĘ Z PROGRAMOWANIEM I ROBOTYKĄ

Możesz od razu przejść do pracy ze scenariuszami, ale mini-słowniczek pomoże Ci zrozumieć czym jest programowanie i robotyka we wczesnej edukacji i ułatwi późniejsze czytanie scenariuszy.

Programowanie – wielu naukowców uważa³, że jest to tzw. trzeci język, dzięki któremu człowiek jest w stanie komunikować się z urządzeniami technologicznymi. Celem programowania w szkole nie jest wykształcenie rzeszy przyszłych programistów. A co nim jest? Uczymy programowania po to, żeby nauczyć uczniów logicznego myślenia, kreatywności i umiejętności rozwiązywania problemów. Programowanie to „przede wszystkim szukanie i wyznaczanie drogi pozwalającej dotrzeć do wybranego celu”⁴. Co ważne programować można (a w przedszkolu i w edukacji wczesnoszkolnej nawet należy) unplugged.

Programowanie unplugged – to programowanie bez użycia komputera. We wczesnej edukacji do nauki programowania można wykorzystywać klocki, zabawki, maty do kodowania, kubki i wiele innych rzeczy codziennego użytku. Przede wszystkim chodzi o to, żeby uczniowie nauczyli się wydawania precyzyjnych poleceń, o naukę szybkiego i precyzyjnego określania kierunków, kodowania, tworzenia algorytmów liniowych, dekompozycji, pętli, instrukcji warunkowej, debugowania⁵.

Robotyka – to dziedzina, która łączy w sobie wiedzę z zakresu techniki, informatyki i inżynierii. W szkole podstawowej to głównie zabawa prostymi robotami, które uczniowie programują, tak by osiągnąć konkretny cel. W naszych scenariuszach dzieci najczęściej programują Photona – jeżdżącego robota wyposażonego w diody, głośnik i szereg czujników. Czy to jedyny robot na rynku? Nie! Możesz pracować z różnymi robotami w zależności od potrzeb, możliwości i preferencji.

Pętla – to polecenie, które wskazuje ile razy powtarzać daną operację. Dzięki pętlom nie trzeba wiele razy pisać tego samego. Stosowanie pętli uczy ekonomicznego myślenia i szukania uproszczeń.

Kodowanie – to pisanie kodu, czyli języka programowego, który jest zrozumiały dla określonej aplikacji⁶. Każdy program to kod.

Debugowanie – to wyszukiwanie i poprawianie błędów w algorytmie lub kodzie. Coś nie działa? Trzeba to sprawdzić! Debugowanie pokazuje dzieciom, że błędy są naturalne gdy działamy, uczy sprawdzania, testowania rozwiązań, cierpliwości i krytycznego myślenia.

[3] B. Koźmińska-Sotśnia, Nauka programowania/ kodowania w edukacji najmłodszych. „Dydaktyka Informatyki”, 13, 2018, s.121-128.

[4] M. Skibińska, J. Zaciewska, Rozwijanie myślenia komputacyjnego u dzieci wczesnej edukacji. „Nauki Humanistyczno-Społeczne”, 453, 2021, s. 39-61, s.44.

[5] M. Roszak, Nauka programowania jako wyzwanie edukacyjne nauczycieli przedszkoli i edukacji wczesnoszkolnej, w: Ewelina Chodźko, Magdalena Śliwa (red.), Interdyscyplinarne badania z zakresu nauk pedagogicznych i humanistycznych. Lublin, 2020, s. 57-66.

[6] M. Roszak, Nauka programowania jako wyzwanie edukacyjne nauczycieli przedszkoli i edukacji wczesnoszkolnej, w: Ewelina Chodźko, 5
Magdalena Śliwa (red.), Interdyscyplinarne badania z zakresu nauk pedagogicznych i humanistycznych. Lublin, 2020, s. 57-66.

Algorytm – to dokładny przepis pokazujący jak krok po kroku wykonać zadanie danego typu, jest powtarzalny i niezawodny⁷.

- algorytm liniowy – sekwencja poleceń, które są wydawane po kolei, a każdy krok wykonywany jest tylko raz. Nie ma w nim warunków: „jeśli... to...”, ani pętli.
- algorytm iteracyjny (z pętlą) – w takim algorytmie czynności powtarzają się określoną liczbę razy (powtórz operację X dokładnie n-razy) lub dopóki spełniony jest warunek (wejścia lub wyjścia), np. przesunąć kubek 5 pól do przodu.
- algorytm warunkowy – to algorytm, w którym pojawia się warunek „jeśli... to...”, np. jeśli wyraz zapisywany jest przez „ó” to zaświeć czerwone światło.
- algorytm zagnieżdżony – to algorytm, który łączy w sobie różne typy algorytmów. Pojawia się gdy na przykład w pętli będzie znajdować się warunek: dzieci mają sprzątać klocki z podłogi (pętla), jeśli jest to klocek LEGO to wrzucają go do niebieskiego kosza, w przeciwnym razie do czerwonego kosza (warunek). Gdy klocki będą posprzątane – zadanie jest wykonane.

Myślenie komputacyjne – to termin, który nie jest Ci potrzebny do pracy ze scenariuszami, ale warto go znać, bo jest podstawą dla programowania i robotyki. Czym jest owo myślenie komputacyjne? To rozwiązywanie problemów krok po kroku z wykorzystaniem czterech technik myślenia komputacyjnego: dekompozycji, analizy, abstrahowania i tworzenia algorytmu⁸.

- Dekompozycja – sformułuj problem i rozłóż go na mniejsze, łatwiejsze elementy.
- Analiza – zastanów się nad sytuacją, poszukaj podobieństw, pomyśl czy widzisz jakieś prawidłowości, które pomogą rozwiązać problem.
- Abstrahowanie – skup się na najważniejszych informacjach, odrzuć niepotrzebne szczegóły.
- Tworzenie algorytmu – stwórz przepis na problem – krok po kroku, następnie przetestuj go i wprowadź poprawki.

Po co kształcić myślenie komputacyjne u uczniów? Przede wszystkim uczy ono brania odpowiedzialności za efekt swoich działań, cierpliwości i krytycznego spojrzenia na to, co się stworzyło. Umiejętność dostrzeżenia problemu, poszukania rozwiązania i debugowania przyda się nie tylko przy programowaniu, ale także przy codziennej nauce i w sytuacjach sprawdzania wiedzy ucznia.

[7] J. Szybowski, Algorytmy w matematycznej edukacji przedszkolnej i wczesnoszkolnej. w: Jaonna Jachimowicz, Zdzisława Zaczona (red.). Współczesna edukacja – wybrane zagadnienia. Nowy Sącz, 2021, s. 54–118.

[8] M. Skibińska, J. Zacniewska, Rozwijanie myślenia komputacyjnego u dzieci wczesnej edukacji. „Nauki Humanistyczno-Społeczne”, 453, 2021, s. 39–61, (s.44).

Z PEWNOŚCIĄ CHCESZ WIEDZIEĆ TEŻ COŚ NA TEMAT POMOCY DYDAKTYCZNYCH, KTÓRE BĘDĄ WYKORZYSTYWANE W ZAPROPONOWANYCH SCENARIUSZACH?

W każdym scenariuszu będziesz korzystać z **maty do kodowania**. Taka mata to zwyczajny dywanik, materiał, arkusz papieru, który jest podzielony na pola – kratownice. Na macie do kodowania uczeń może tworzyć instrukcje, dekodować informacje i tworzyć kod dla np. robota. Na rynku edukacyjnym dostępnych jest wiele mat do kodowania, te które znajdziecie na zdjęciach w scenariuszach są stworzone przez Annę Świć – założycielkę firmy Kodowanie na dywanie (na stronie kodowanienadywanie.pl możecie kodować własne maty!⁹).

Pracując z wykorzystaniem scenariuszy będziecie również wykorzystywać **robota Photona**¹⁰ – warto więc dowiedzieć się czym jest i jakie ma możliwości.

Interdyscyplinarny robot Photon wyposażony jest w czujnik światła, czujnik dotyku, czujnik obrotu, odległości, kontrastu oraz mikrofon i głośnik, a także podświetlane oczy i czułka. Robot łączy się z tabletami oraz smartfonami zarówno z systemem Android, jak i IOS.

Żeby robot wykonywał polecenia dzieci, musi zostać zaprogramowany.

W jaki sposób? Najpierw na urządzenie należy pobrać aplikację: 1) Photon Edu – dedykowana dla szkół, 2) Photon Robot – jeśli twoim celem jest indywidualna praca uczniów, 3) Photon Coding – polecana użytkownikom domowym, 4) Photon Magic Bridge – dzięki której uczniowie mogą rozwinąć dodatkowe umiejętności programowania za pomocą Pythona, Scratcha, MakeCode i JavaScript, 5) Photon AI – w którym można korzystać z dwóch ścieżek edukacyjnych zawierających 10 lekcji, wprowadzających uczniów w świat sztucznej inteligencji.

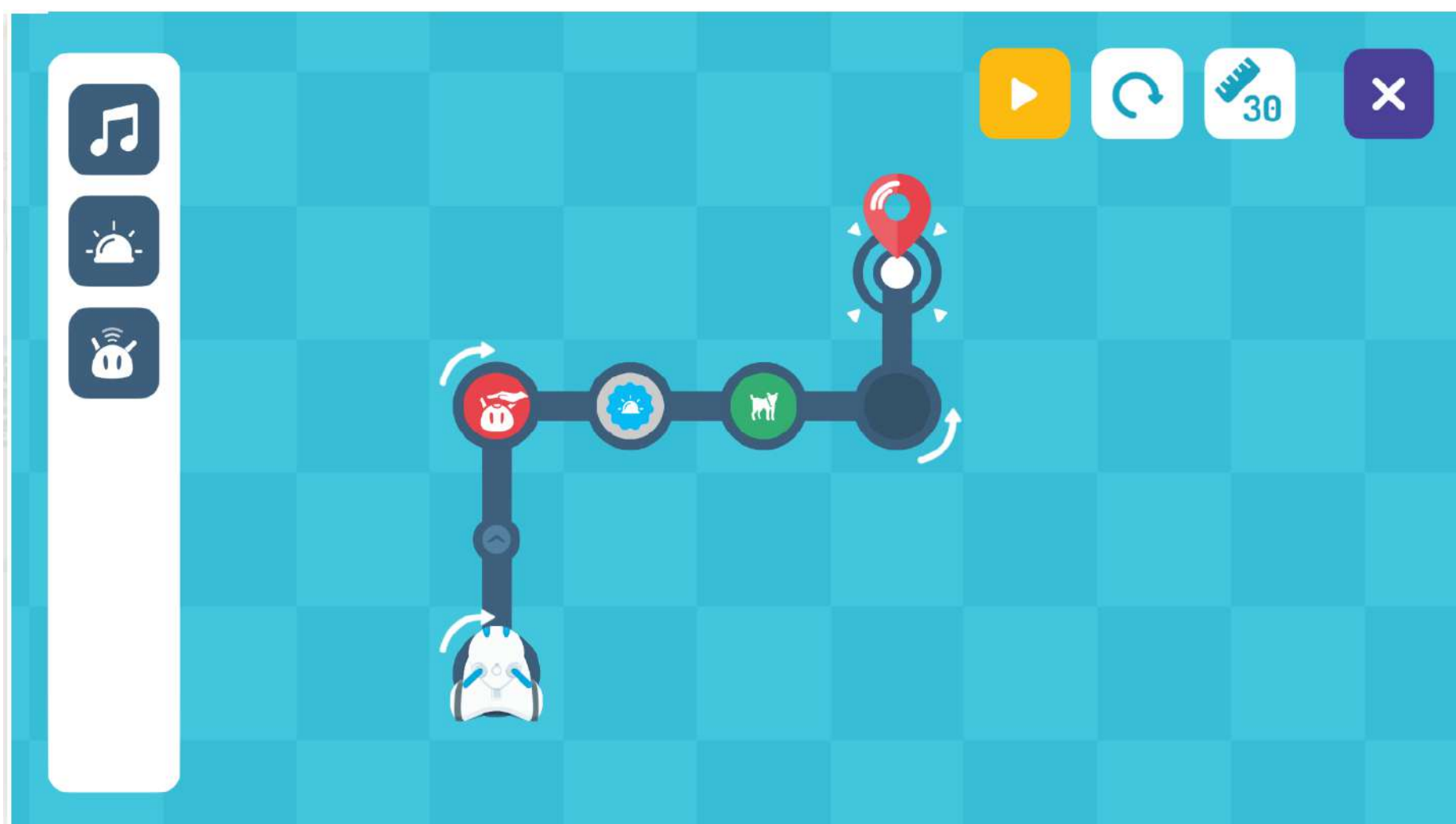
Producent przygotował cztery różne języki programowania, od prostych do bardziej złożonych, z których możesz korzystać podczas pracy z proponowanymi scenariuszami. W proponowanych scenariuszach wykorzystasz trzy z proponowanych języków programowania: Photon Draw, Photon Bridge oraz Photon Blocks. Dostępny jest również Photon Code, który jest najbardziej zaawansowany, więc jeśli uważasz, że Twoi uczniowie mogą z niego korzystać – wybierz właśnie ten.

[9] <https://kodowanienadywanie.pl/>

[10] <https://photon.education/pl/aplikacje/#>

Photon Draw – jest to najprostszy język, przeznaczony dla dzieci powyżej 5 roku życia. W tym interfejsie robota programuje się przez rysowanie palcem ścieżki na ekranie dotykowym. Do narysowanej ścieżki można dodać instrukcje takie jak: wydaj dźwięk, poczekaj na dotyk, zmień kolor^[1].

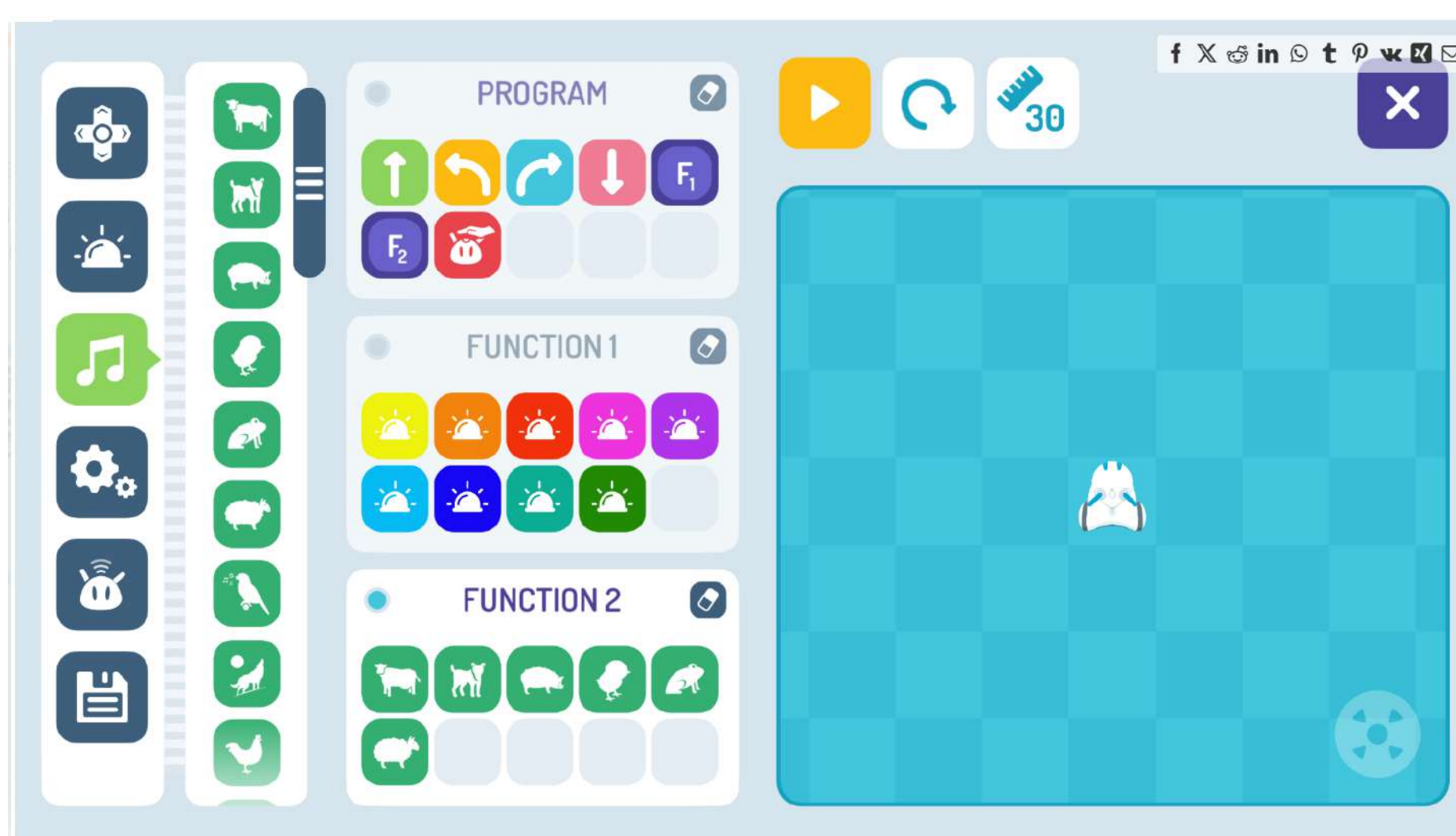
Rysunek 1. Interfejs Photon Draw



Źródło: <https://photon.education/pl/aplikacje/#>.

Photon Badge – to interfejs przeznaczony dla dzieci od 6 roku życia. Programuje się w nim za pomocą graficznych instrukcji, które są układane w odpowiedniej kolejności w oknie programu. Jeden program może zawierać maksymalnie 10 instrukcji. Użytkownik ma możliwość wykorzystania funkcji – F1 i F2 (w oknie F1 lub F2 umieszcza się sekwencję poleceń). Aby wykorzystać funkcję należy umieścić w głównym programie bloczek F1 lub F2^[2].

Rysunek 2. Interfejs Photon Badge



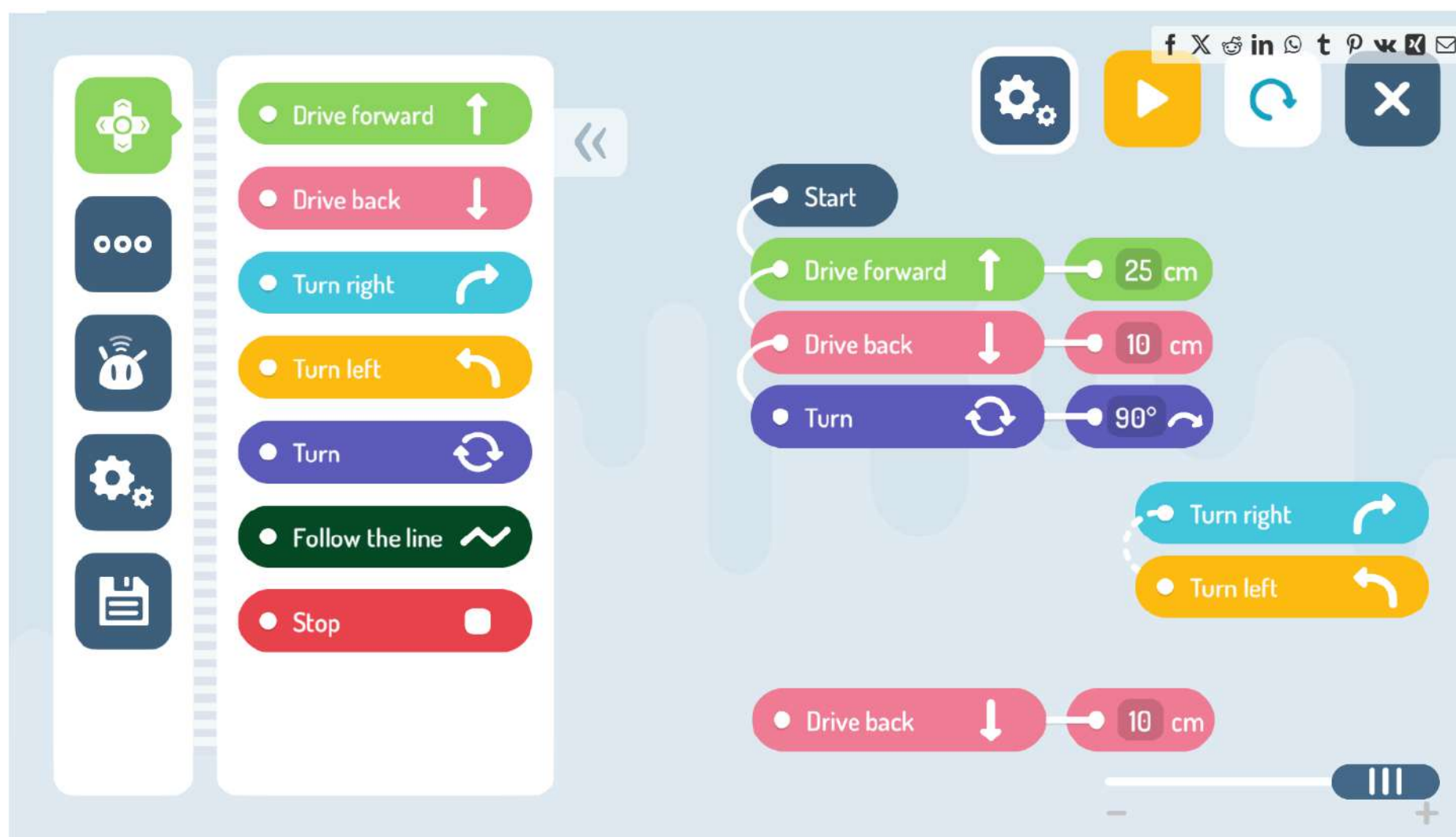
Źródło: [https://photon.education/pl/aplikacje/#iLightbox\[image_carousel_3\]/1](https://photon.education/pl/aplikacje/#iLightbox[image_carousel_3]/1).

[11] O. Syrocka, D. Skrzypek, Recenzja Photona: Narzędzie do nauki czy ładna zabawka?, 2018, online: <https://www.robocamp.pl/pl/blog/photon-robot-edukacyjny-recenzja/> [dostęp: 14.04.2025].

[12] O. Syrocka, D. Skrzypek, Recenzja Photona: Narzędzie do nauki czy ładna zabawka?, 2018, online: <https://www.robocamp.pl/pl/blog/photon-robot-edukacyjny-recenzja/> [dostęp: 14.04.2025].

Photon Blocks – to język programistyczny w graficzno–tekstowej formie. Użytkownik ma do wyboru cztery kategorie: może zaprogramować ruch robota i doprecyzować drogę jaką ma pokonać lub kąt obrotu; może zastosować różne efekty dźwiękowe i wizualne, związane ze zmianą koloru czułek lub oczu oraz wydawaniem różnych dźwięków; może zastosować różne funkcje, np. pętle lub instrukcje warunkowe, przerwanie pętli oraz nadanie sygnału IR oraz może zaprogramować własne funkcje. Aby utworzyć program wystarczy przeciągnąć odpowiedni bloczek na ekran roboczy¹³.

Rysunek 3. Interfejs Photon Blocks



Źródło: [https://photon.education/pl/aplikacje/#iLightbox\[image_carousel_3\]/2](https://photon.education/pl/aplikacje/#iLightbox[image_carousel_3]/2).

Poza matą do kodowania oraz robotem Photon, żeby korzystać ze scenariuszy będziesz potrzebować kolorowych kubków oraz materiałów dydaktycznych, które gotowe do druku, są zamieszczone na końcu każdego scenariusza.

Pewnie zastanawiasz się czy to jedyne elementy, które można wykorzystać przy nauce programowania i robotyki w klasach 1–3? Nie! Jest ich bardzo dużo, więc jeśli masz dostęp do innych, spróbuj dopasować do nich nasze propozycje metodyczne.

Jeśli chcesz poznać więcej wskazówek metodycznych dotyczących robotyki i programowania zapraszam na stronę **www.algorytmprzyszlosci.pl**, na której odnajdziesz cykl krótkich filmików o tym jak uczyć programowania w klasach 1–3.

Przykładowe roboty edukacyjne: Bee–Bot, Ozobot Bit/ Evo, Cubetto, Scottie Go!, Lego® Education WeDo 2.0, Doc – Clementoni.

**Teraz możesz już zacząć czytać scenariusze, chyba że...
chcesz poznać trochę teorii?**

[13] O. Syrocka, D. Skrzypek, Recenzja Photona: Narzędzie do nauki czy ładna zabawka?, 2018, online: <https://www.robocamp.pl/pl/blog/photon-robot-edukacyjny-recenzja/> [dostęp: 14.04.2025].

KRÓTKO O TYM CO, JAK, KIEDY I DLACZEGO UCZYĆ?

Metodyka nauczania robotyki i programowania na etapie edukacji wczesnoszkolnej nawiązuje do ogólnych wskazówek, które możemy znaleźć w dydaktyce nauczania zintegrowanego lub we wskazówkach neurodydaktyków, dotyczących tego jak uczyć efektywnie.

Można jednak stwierdzić, że zabierając się do uczenia z wykorzystaniem robotów i programowania już na wstępie mamy nieco prościej, bo z założenia będziemy badać, tworzyć hipotezy, eksplorować możliwości, projektować, sprawdzać i działać praktycznie – bez takich działań nie ma mowy o robotyce i programowaniu. Chyba nikt nie pomyślałby o tym, żeby uczyć tych umiejętności wyłącznie opowiadając o tym „jak można programować” – bez działania!

Obiecane krótko o ważnym...

Współcześnie w edukacji coraz częściej pisze się i głośno mówi o tym, że uczniowie, aby przyswoili wiedzę muszą działać. Muszą konstruować wiedzę, samodzielnie interpretować zjawiska, rekonstruować posiadane informacje i wyjaśniać pojawiające się problemy. Coraz częściej słyszymy, że uzupełnianie zeszytów ćwiczeń nie jest efektywne, że edukacja powinna wyglądać nieco inaczej – być bardziej spersonalizowana i nastawiona na proces, a nie wynik. Pewnie pomyślicie – teoria sobie, a życie sobie. Nie sposób się z tym nie zgodzić, ale jednak spróbuję Was przekonać do działania, bo jak się okazuje, największą moc zmiany w edukacji mają nauczyciele – choćby w jednej klasie, to już dużo!

To co od wielu lat postuluje się w edukacji nazywa się paradygmatem **konstruktywistycznym** (jest kilka jego odmian, ale jeśli Was to interesuje to odsyłam do książek i artykułów np. D. Klus-Stańskiej^[14]). Założenia konstruktywizmu można streścić w następujących punktach:

1. Żeby uczeń się uczył ważne jest tworzenie warunków, w których będzie mógł samodzielnie rozwiązywać problemy (tak – to zajmie trochę czasu).
2. Uczenie się nowych rzeczy jest dużo łatwiejsze jeśli nawiążemy do tego co uczeń już wie (żeby to zrobić, musimy dać uczniom działać – ich aktywność jest punktem wyjścia dla nauczyciela).
3. Ważniejsze jest żeby uczeń samodzielnie działał i popełniał błędy, szukał gdzie popełnił błąd i żeby te błędy naprawiał, niż żeby nauczyciel podał mu gotowe rozwiązanie (choć tak jest szybciej).
4. Niemożliwe jest żeby uczeń przyjął nauczycielski sposób rozumienia świata, to nauczyciel musi poznać uczniowski sposób rozumienia treści zajęć^[15].

[14] D. Klus-Stańska, Paradygmaty dydaktyki. Myśleć teorią o praktyce. Warszawa 2018.

[15] D. Klus-Stańska, Paradygmaty dydaktyki. Myśleć teorią o praktyce. Warszawa 2018, s.131-167.

W konstruktywizmie nauczyciel stawia przed uczniami problem, który oni, pracując w grupach (lub nie) rozwiązują, a zajęcia kończą się ewaluacją – czyli oceną tego co zrobiliśmy, co nam się udało, a nad czym będziemy jeszcze pracować. Uczenie się to proces, który wymaga czasu.

Seymour Papert był jednym z badaczy, który jako pierwszy połączył edukację z nowoczesnymi technologiami (i został przedstawicielem **konstrukcjonizmu**). Co ciekawe, twierdził, że wykorzystanie nowoczesnych technologii w tradycyjnej dydaktyce jest bezcelowe. Żeby wykorzystywać nowe media w edukacji, należy zmienić sposób pracy z uczniami. Papert zakładał równomierny rozwój trzech aspektów rozwoju poznawczego:

1. Mentalny – aktywne konstruowanie wiedzy w umyśle dziecka;
2. Społeczny – uczymy się przez współpracę i dyskusje z innymi;
3. Materialny – niezbędne jest praktyczne działanie, tworzenie i konstruowanie¹⁶.

S. Papert stworzył język programowania *Logo*, gdyż uważał, że nauka programowania to nauka myślenia, podczas programowania dzieci rozwijają własne strategie uczenia się. Co istotne – celem nie jest samo wykorzystanie robotów, programów i aplikacji – celem jest poszerzenie wiedzy uczniów, testowanie przez nich różnych rozwiązań i samodzielne uczenie się. **Wykorzystanie technologii służy jakiemuś celowi.**

Neurodydaktycy wskazują na cztery filary uczenia się, na które należy zwrócić uwagę projektując zajęcia – **uwaga, zaangażowanie, informacja zwrotna i konsolidacja**. Owe cztery filary są powiązane z tym, co zawiera się w koncepcji oceniania kształtującego.

Uczeń ma motywację do uczenia się kiedy zna cel, do którego zmierza (cel zajęć to ukierunkowanie uwagi ucznia); niezbędne jest zaciekawienie ucznia tematem, dzięki któremu jest w stanie się w pełni zaangażować (tutaj pojawiają się pytania kluczowe, radość z uczenia się, tworzenie hipotez i ich testowanie, popełnianie błędów). Żeby nauka była efektywna niezbędna jest informacja zwrotna – jeśli uczeń wie co konkretnie zrobił dobrze, a co i jak musi jeszcze poprawić, nie działa po omacku. W przypadku programowania i robotyki informacja zwrotna pojawia się w fazie testowania kodu – uczeń sam przekona się czy jego algorytm jest poprawny czy coś należy jeszcze zmienić. Konsolidacja odnosi się do czasu, który jest potrzebny, aby zautomatyzować czynności. Mózg ucznia potrzebuje zarówno czasu intensywnej nauki jak i odpoczynku (relaksacji, zamknięcia oczu, spaceru, aktywności fizycznej itp.), podczas którego zdobyte informacje są utrwalane w mózgu¹⁷.

[16] D. Siemieniecka, B. Siemieniecki, *Teorie kształcenia w świecie cyfrowym*. Kraków 2019, s. 196–200.

[17] S. Dehaene, *Jak się uczymy? Dlaczego mózgi uczą się lepiej niż komputery... jak dotąd*. Kraków, 2022.

Na końcu części „krótko o ważnym” warto wspomnieć o metodzie STEAM, która łączy w sobie założenia konstruktywizmu, konstrukcjonizmu oraz neurodydaktyki. STEAM to podejście integrujące wiedzę z zakresu: S – nauk przyrodniczych, T – technologii, E – inżynierii, A – sztuki, M – matematyki, realizowane w formie angażujących projektów. Jest to metoda doskonale wpisująca się w ideę edukacji zintegrowanej w klasach 1–3, ponieważ umożliwia łączenie różnych obszarów wiedzy w sposób naturalny i praktyczny. Kluczowym elementem pracy metodą STEAM jest współpraca uczniów – nie tylko nad zagadnieniami merytorycznymi, ale także nad rozwijaniem kompetencji miękkich, takich jak: komunikowanie się, negocjowanie, wspólne podejmowanie decyzji, kreatywność i krytyczne myślenie. STEAM opiera się na wielozmysłowym poznaniu, rozwiązywaniu realnych problemów, a także na samodzielnej (nie koniecznie samotnej) pracy uczniów i przejmowaniu przez nich odpowiedzialności za efekt końcowy¹⁸.

Podsumowując: programowanie i robotykę powinniśmy wykorzystywać do poszerzania wiedzy i umiejętności uczniów z różnych dziedzin. Nie po to, żeby wykorzystać roboty, a po to, aby uczniowie działali i przenosili abstrakcyjne koncepcje na realne działanie.

Już wiesz, że programowanie i robotyka to przede wszystkim rozwijanie logicznego myślenia, kreatywności oraz pewności siebie u uczniów.

Wykorzystanie robotów w edukacji to także doskonały sposób na zwiększenie motywacji wewnętrznej do nauki i utrwalanie podstawowych umiejętności – w atmosferze zabawy i odkrywania.

A co jeśli coś się nie uda? Pamiętaj, że na błędach uczą się nie tylko uczniowie, ale też Ty!

Mamy nadzieję, że ten skrypt będzie dla Ciebie inspiracją do tworzenia własnych, angażujących lekcji. Świat edukacji zmienia się razem z technologią. A Ty właśnie pomagasz dzieciom wejść w ten świat z odwagą i radością.

ODRZUĆ ZESZYTY ĆWICZEŃ. WŁĄCZ ROBOTA. I ZACZNIJ DZIAŁAĆ.

[18] A. Bojarska-Sokołowska, STEAM-owe aktywności matematyczne dzieci w wybranych europejskich centrach nauki. „Edukacja Elementarna w Teorii i Praktyce”, 19, 4(75), 2024, s. 39–54.

**GDYBYM BYŁ_A
PROGRAMISTĄ_KĄ**

Na czym polega praca programisty? Jakie kompetencje potrzebne są w tym zawodzie? Dlaczego warto uczyć się programowania? Odpowiedzi na powyższe pytania poszukają uczniowie na zajęciach przeprowadzonych według zaproponowanego scenariusza. Pracując z robotami sprawdzą też jakie możliwości i funkcje dostępne są w poszczególnych interfejsach w aplikacji służącej do ich programowania.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Wprowadzenie w świat robotyki;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie kompetencji miękkich, ze szczególnym uwzględnieniem umiejętności pracy zespołowej

Cele operacyjne:

Uczeń podczas zajęć:

- własnymi słowami wyjaśnia na czym polega zawód programisty;
- wymienia korzyści związane są z nauką programowania;
- opowiada czym jest kod i poszukuje kodów w codziennych sytuacjach;
- analizuje w jaki sposób wykorzystuje się roboty np.: w przemyśle, medycynie, gastronomii;
- programuje robota Photon wykorzystując interfejs Photon Draw, Photon Badge i Photon Blocks;
- chętnie i aktywnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

Ok. 45 – 90 minut

Wprowadzenie

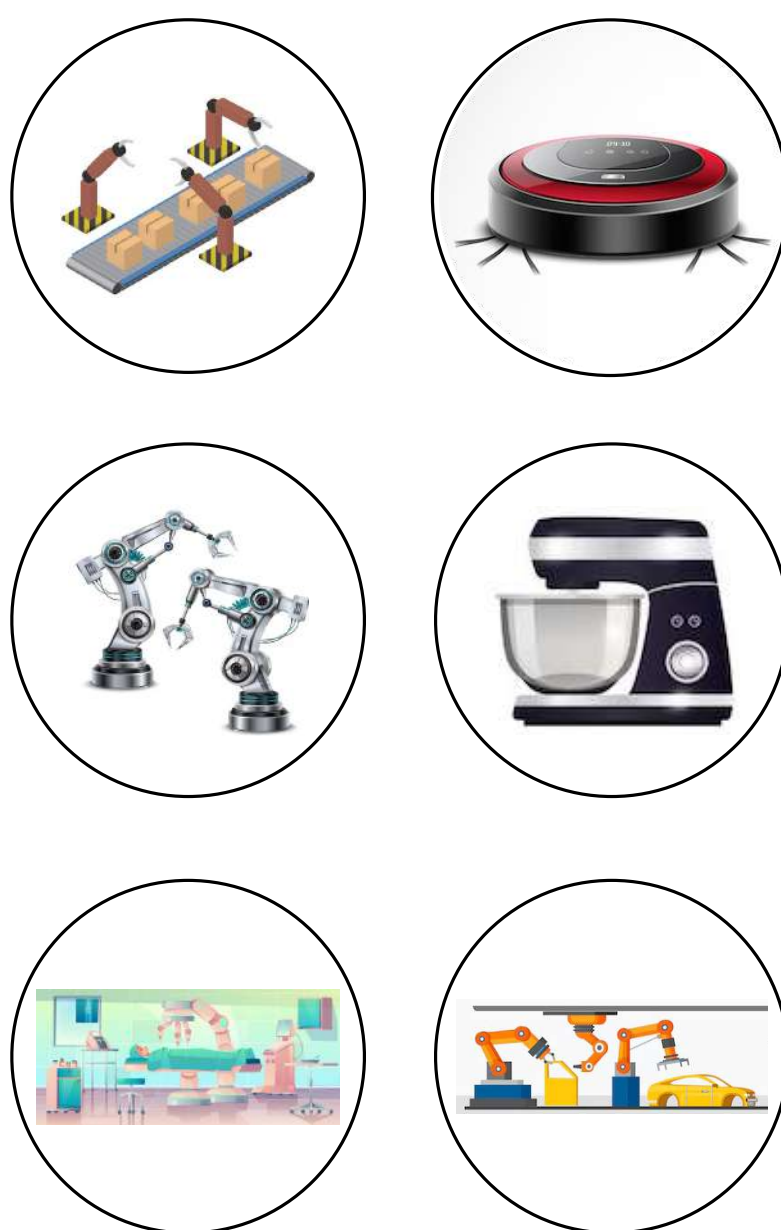
Nauczyciel siada z uczniami w kole i pyta jak ich zdaniem wygląda zawód programisty? Z jakimi zadaniami styka się osoba wykonująca taką pracę? W jaki sposób my korzystamy z pracy programistów. Wspólnie z dziećmi szuka przykładów wykorzystania programowania w codziennym życiu.

Uczniowie rozmawiają o tym, jakie umiejętności są potrzebne do wykonywania zawodu programisty (również o tych mniej oczywistych, przykładowo o umiejętności współpracy czy kreatywności). Swoje spostrzeżenia uczniowie (lub nauczyciel) zapisują na niewielkich kartkach (będą potrzebne w dalszej części zajęć).

Część właściwa

Nauczyciel prezentuje uczniom materiał pomocniczy „Roboty” (rys. 1) i pyta co się na nim znajduje, co łączy przedmioty umieszczone na poszczególnych krążkach. Uczniowie przedstawiają własne interpretacje, wskazują zastosowanie przedstawionych robotów.

Rysunek 1. Ilustracje robotów



Źródło: opracowanie własne A. Świć.

Nauczyciel prosi uczniów, aby przypomnieli sobie gdzie mogą być wykorzystywane roboty, a następnie o podzielenie się własną wiedzą na ten temat z resztą grupy. Kiedy uczniowie zakończą prezentacje swoich pomysłów, nauczyciel może wskazać i wyjaśnić rolę robotów przemysłowych, sposób wykorzystywania robotów, np. w medycynie, wskazuje na roboty domowe, np. kuchenne lub odkurzacze.

Nauczyciel pyta uczniów: co istotnego musi się wydarzyć, żeby roboty mogły spełniać swoją funkcję – żeby mogły „ożyć”? Jeśli uczniowie nie potrafią udzielić odpowiedzi, nauczyciel wyjaśnia, że muszą one zostać zaprogramowane. Podobnie jak są różne funkcje robotów, tak różny może być sposób ich programowania.

Roboty typu *line follower* potrzebują trasy, po której będą podążać, inne roboty mogą być programowalne przy pomocy gestów, a jeszcze inne przy pomocy wizualnego języka programowania takiego jak scratch lub blockly (nauczyciel prezentuje wizualnie różne typy robotów).

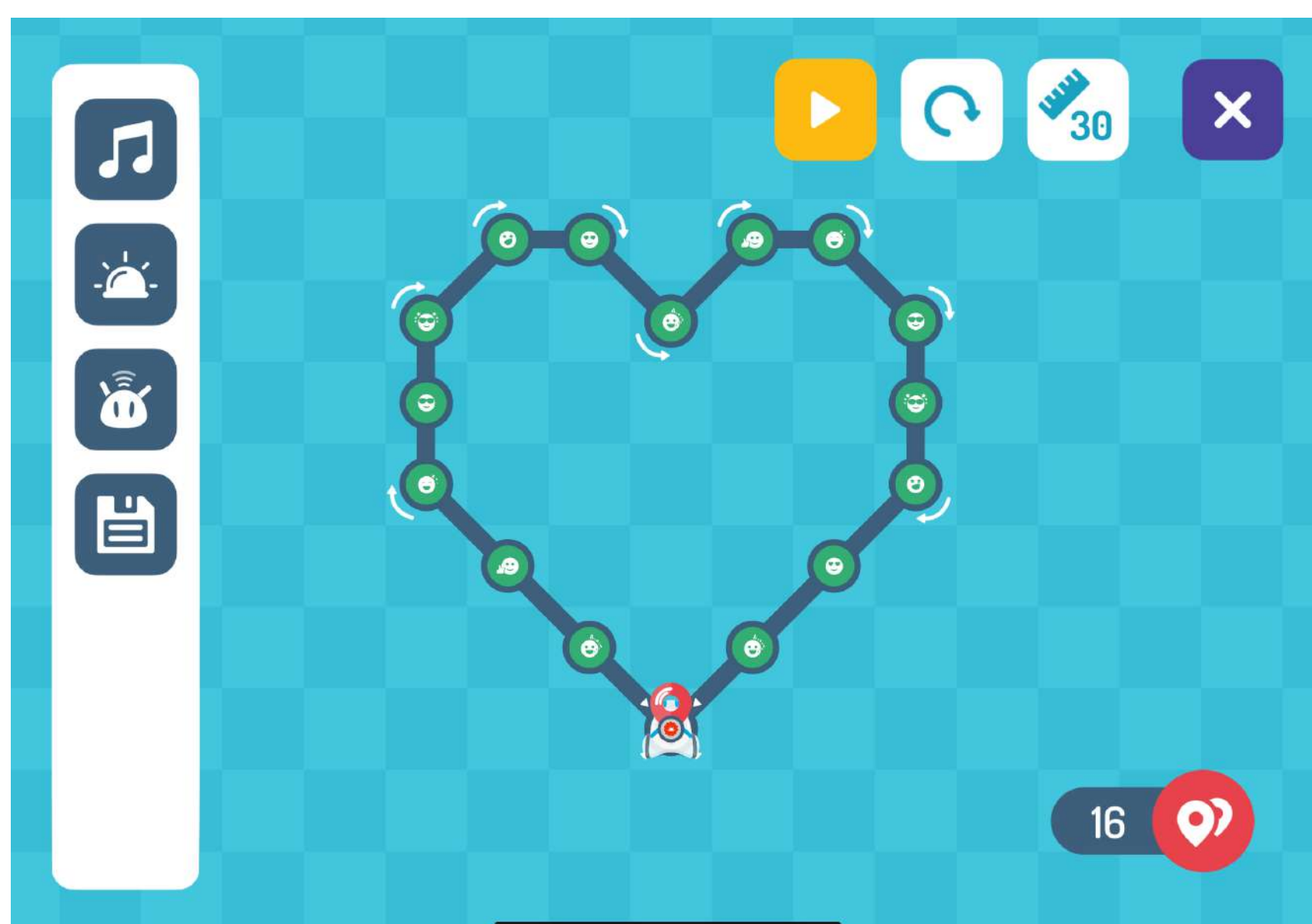
Roboty typu line follower to takie roboty, które jeżdżą tylko po wyznaczonej trasie – zazwyczaj po czarnej linii narysowanej na białej powierzchni. Poruszają się dzięki specjalnym czujnikom na dole, które “patrzą”, gdzie jest linia.

Na prowadzonych zajęciach dzieci będą programować roboty Photon, ale zanim to zrobią, wcielą się w roboty programowalne przy pomocy gestów. Nauczyciel ustala z uczestnikami zajęć, że będą poruszać się zgodnie z gestami, które będzie pokazywał: dłoń przechylona do przodu – krok do przodu, dłoń skierowana do tyłu – krok do tyłu, dłoń skierowana w prawo – obrót w prawo (w miejscu), dłoń skierowana w lewo – obrót w lewo (w miejscu).

Po zabawie, w której uczniowie byli robotami nauczyciel prezentuje roboty, z których będą korzystać podczas zajęć z programowania. Będą nimi roboty Photon. Nauczyciel pokazuje różne interfejsy dostępne w aplikacji Photon Coding (lub Photon Edu) i wyjaśnia jak korzystać z podstawowych funkcji. Pokazuje też akcesoria, które umożliwiają tworzenie za pomocą robota różnych rysunków (zaczep na marker);

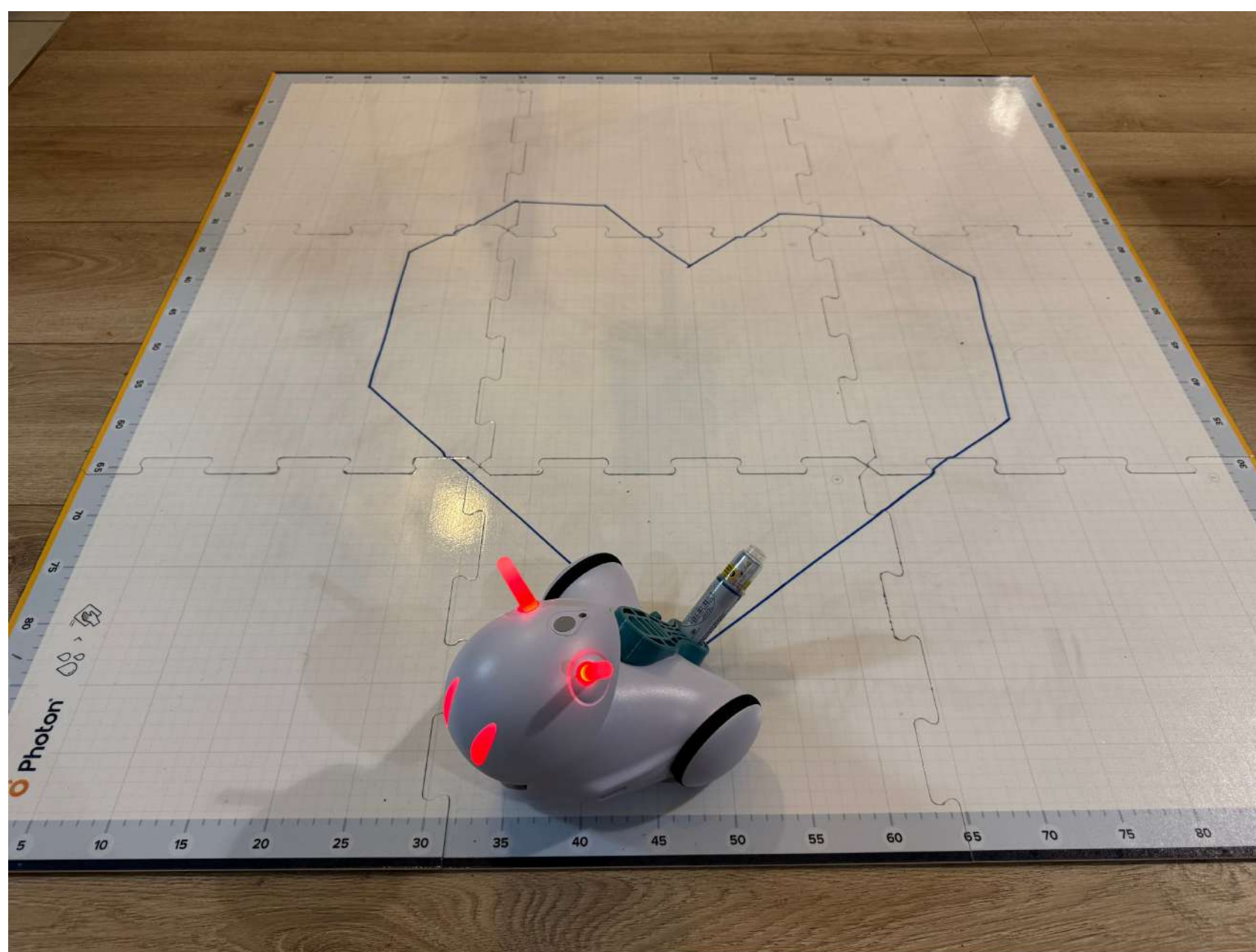
Pierwsze zadanie dzieci wykonają wykorzystując interfejs Photon Draw. Będzie nim takie zakodowanie robota żeby narysował serce, świecił kolorem czerwonym i w trakcie rysowania wydawał różne dźwięki radości.

Rysunek 2. Interfejs Photon Draw



Źródło: opracowanie własne A. Świć.

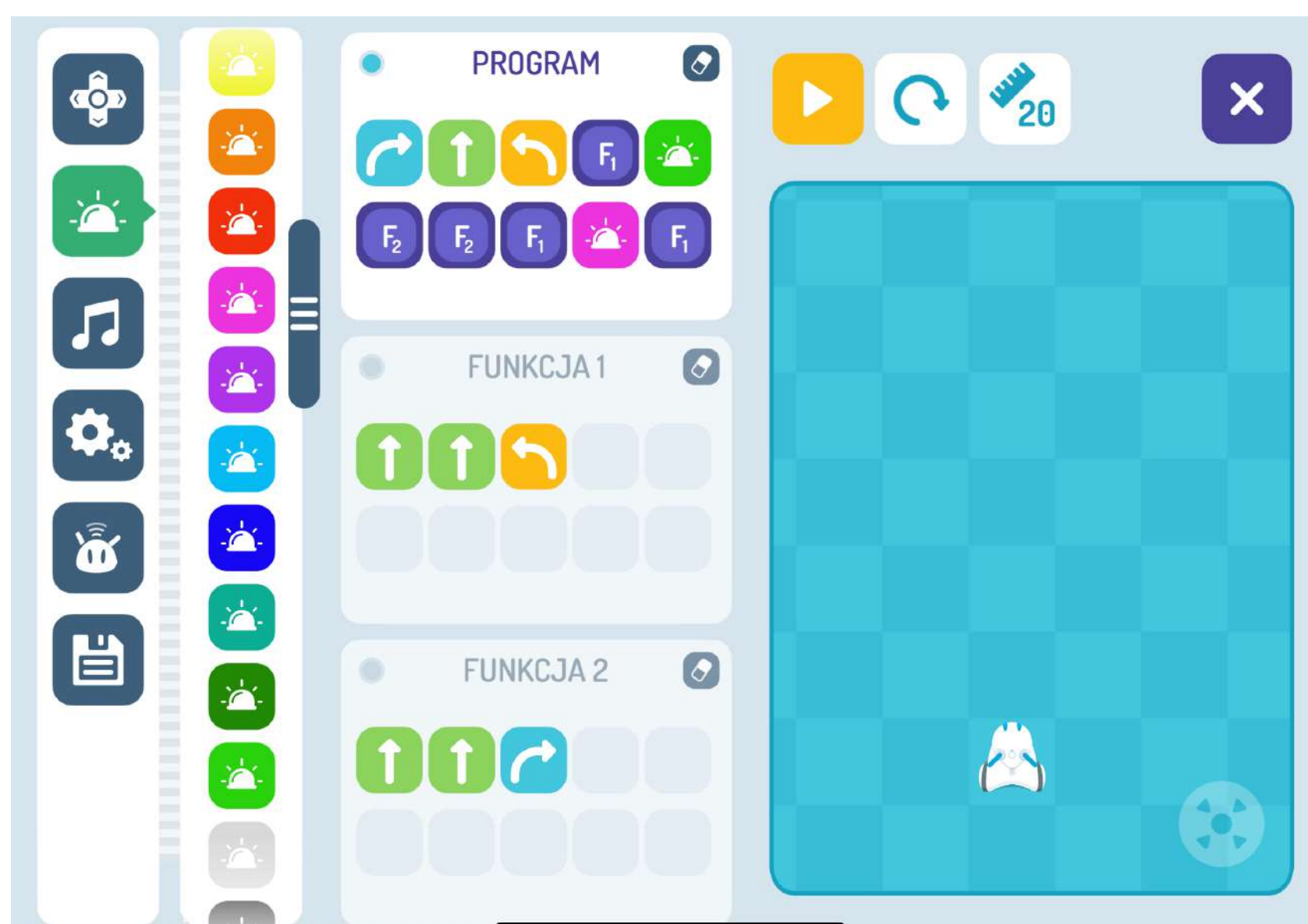
Rysunek 3. Praca robota Phyton wg napisanego przez uczniów kodu



Źródło: opracowanie własne A. Świć.

W drugim zadaniu użyty zostanie interfejs Photon Badge. Tym razem robot będzie musiał pokonać slalom omijając kilka kolorowych kubków. Przejeżdżając obok każdego z nich będzie dopasowywał kolor, którym świeci do barwy kubka.

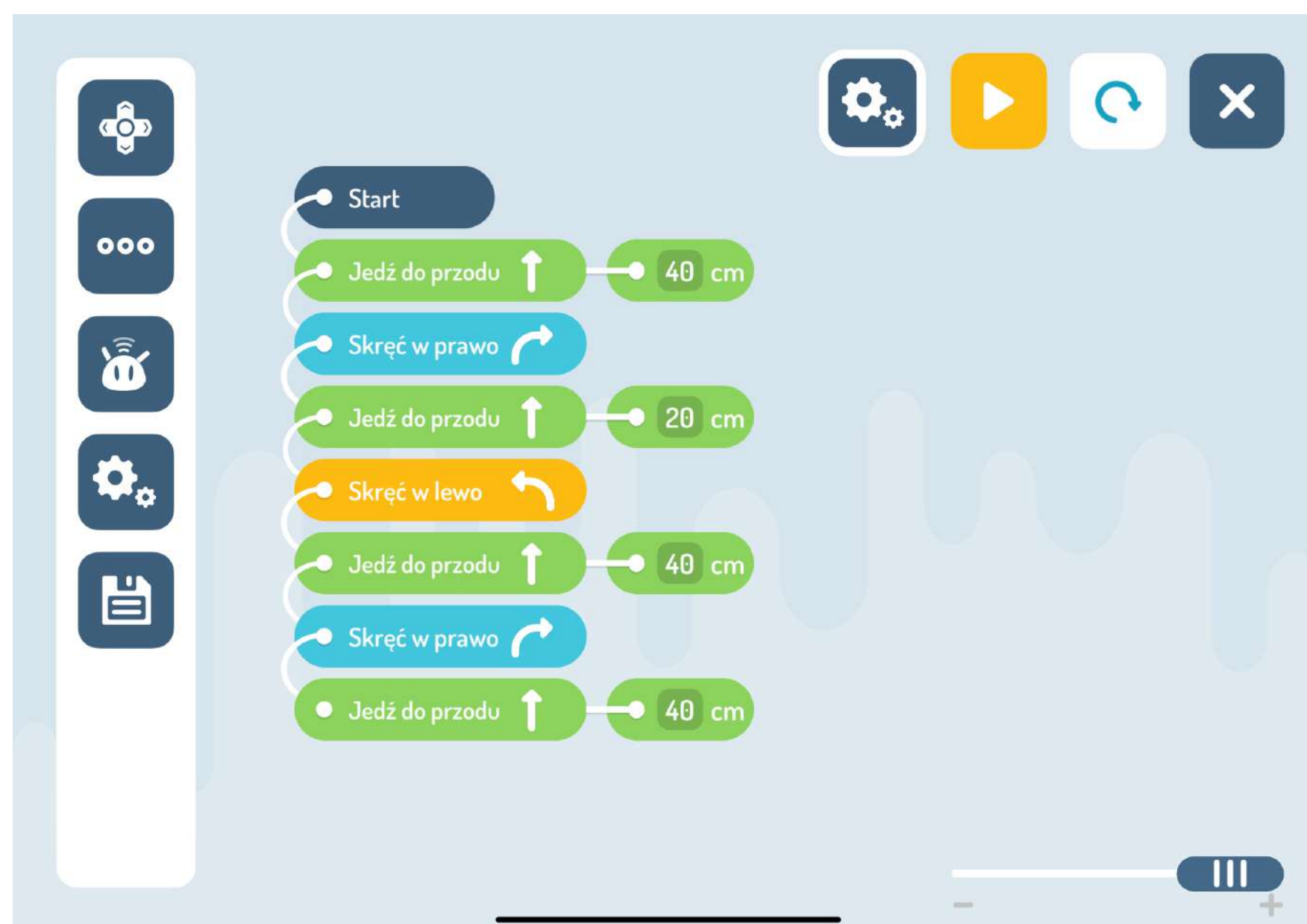
Rysunek 4. Interfejs Photon Badge



Źródło: opracowanie własne A. Świć.

Do trzeciego zadania użyty będzie interfejs Photon Blocks. Uczniowie rozłożą na macie przygotowane wcześniej kartki z napisanymi korzyściami, które wynikają z nauki programowania, a następnie zaprogramują robota w taki sposób, żeby po nich wszystkich przejechał. Jeszcze raz wspólnie omówią czy i dlaczego warto uczyć się programowania.

Rysunek 5. Interfejs Photon Blocks



Źródło: opracowanie własne A. Świć.

Zakończenie zajęć

Nauczyciel prosi uczniów, aby porozmawiali w parach o tym, z którym interfejsem pracowało im się najwygodniej, które funkcje dostępne w robocie najbardziej im się spodobały, co sprawiło im największą trudność. Chętne dzieci wyrażają swoją opinię.

“Strategia zastanów się–znajdź partnera–podziel się”¹⁹

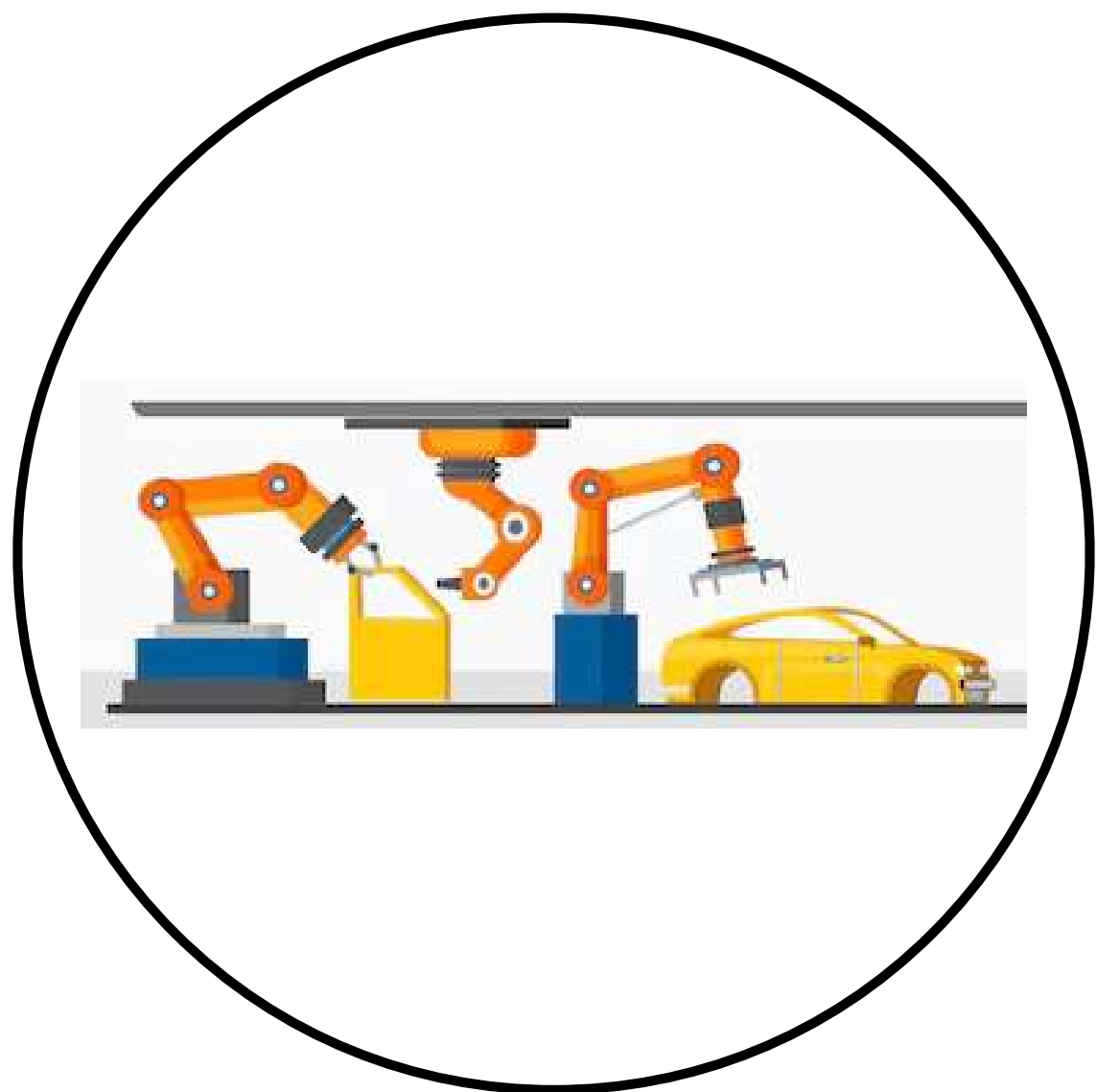
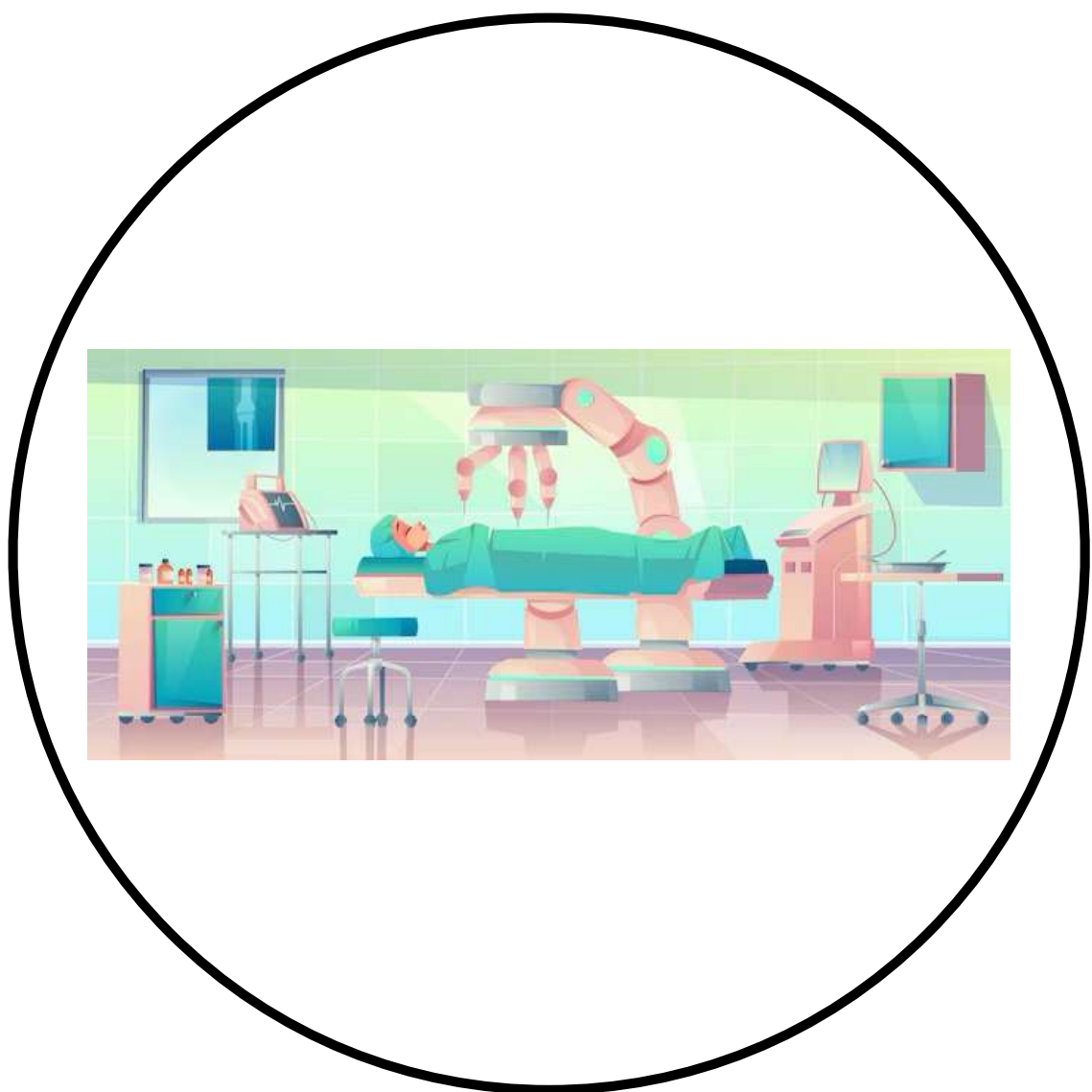
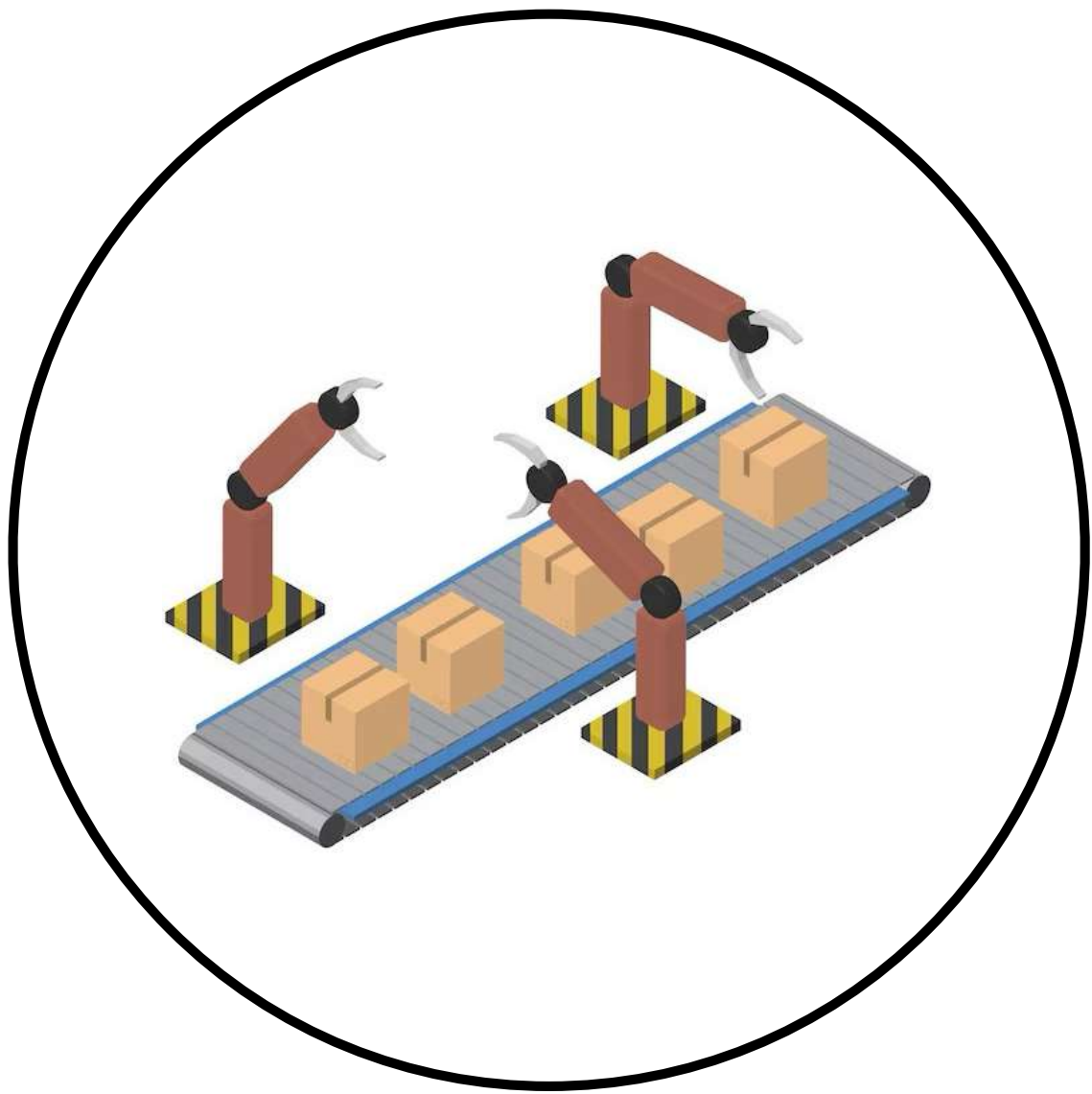
Krok 1. Nauczyciel zadaje uczniom pytanie i daje im czas na to, aby samodzielnie zastanowili się nad odpowiedzią. Można poprosić, aby uczniowie zapisali swoje przemyślenia.

Krok 2. Uczniowie szukają partnera_ów, z którym_i umówią swoje przemyślenia i sformułują wspólną, najpełniejszą odpowiedź.

Krok 3. Uczniowie przedstawiają swoje odpowiedzi na forum klasy.

Co to daje? Więcej uczniów jest aktywnych – myśli nad odpowiedzią, odpowiedzi są pełniejsze i uczniowie uczą się od siebie nawzajem!

[19] F. Lyman, The Responsive Classroom Discussion, w: A. S. Anderson (red.) Mainstreaming Digest. College Park, MD 1981, 109–113.



5,10,15...

ZAPROGRAMOWANE

ZEGARY

Czy pięć minut, to długo, czy krótko? Co można zrobić w piętnaście minut? Czym jest kwadrans? Odpowiedzi na powyższe pytania poszukają uczniowie na zajęciach przeprowadzonych według zaproponowanego scenariusza. Pomoże im w tym zegar, który zostanie narysowany przez robota... jeśli uprzednio dzieci właściwie tego robota zaprogramują.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Poszerzenie wiadomości na temat czasu;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie kompetencji miękkich, ze szczególnym uwzględnieniem umiejętności pracy zespołowej

Cele operacyjne:

Uczeń podczas zajęć:

- poprawnie odczytuje pełne godziny na zegarze;
- własnymi słowami opisuje czym jest kwadrans;
- dokładnie określa czas, który minął między jedną a drugą wskazaną godziną;
- tworzy obrazek na podstawie podanych współrzędnych;
- rozwiązuje logiczne zagadki;
- programuje robota Photon wykorzystując interfejs Photon Blocks;
- zgodnie i efektywnie pracuje zespołowo.

Formy organizacyjne:

- praca grupowa
- praca zespołowa
- praca indywidualna

Środki dydaktyczne:

Mata do kodowania, duży arkusz papieru lub mata do rysowania, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

Ok. 45 – 90 minut

Wprowadzenie:

Nauczyciel wita się z dziećmi. Przedstawia cel zajęć, jakim jest utrwalenie umiejętności posługiwania się zegarem oraz zaprogramowanie robota Photona, tak by narysował zegar, którego tarcza posłuży uczniom do różnorodnych zabaw z czasem.

Część właściwa:

Uczniowie zastanawiają się, jak tarcza zegara narysowana przez Photona może wyglądać. Nauczyciel prosi, żeby każda pełna godzina na tarczy zaznaczona była kreską. Po narysowaniu każdej z kresek robot powinien zatrzymać się na kilka sekund. Będzie to czas, w którym uczniowie napiszą przy kresce odpowiednią cyfrę.

Najtrudniejsze w tym zadaniu będzie ustalenie o jaki kąt powinien obrócić się robot. Pełne koło ma 360 stopni, w naszym przypadku tarcza zegara będzie podzielona na dwanaście części, więc pojedynczy obrót między poszczególnymi odcinkami powinien wynosić 30 stopni (360 podzielone na 12 części daje 30).

Jak wyjaśnić uczniom obrót o x stopni?

Możesz powiedzieć, że "stopnie to sposób mierzenia, który wymyślili ludzie, żeby zmierzyć jak bardzo coś się obróciło".

Zaproponuj dzieciom zabawy, żeby doświadczyły znaczenia tego obrotu na sobie, zanim zaczną programować robota.

W jaki sposób?

Możesz poprosić, aby dzieci stanęły w miejscu i wykonywały Twoje polecenia. Aby ułatwić dzieciom zadanie możesz na podłodze nakleić koło (tarczę zegara) podzielone na 12 części. Obrócenie się o jedną godzinę oznacza wykonanie obrotu o 30°.

Polecenie 1. Pełny obrót (360°)

Polecenie 2. Pół obrotu (180°)

Polecenie 3. Ćwierć obrotu (90°)

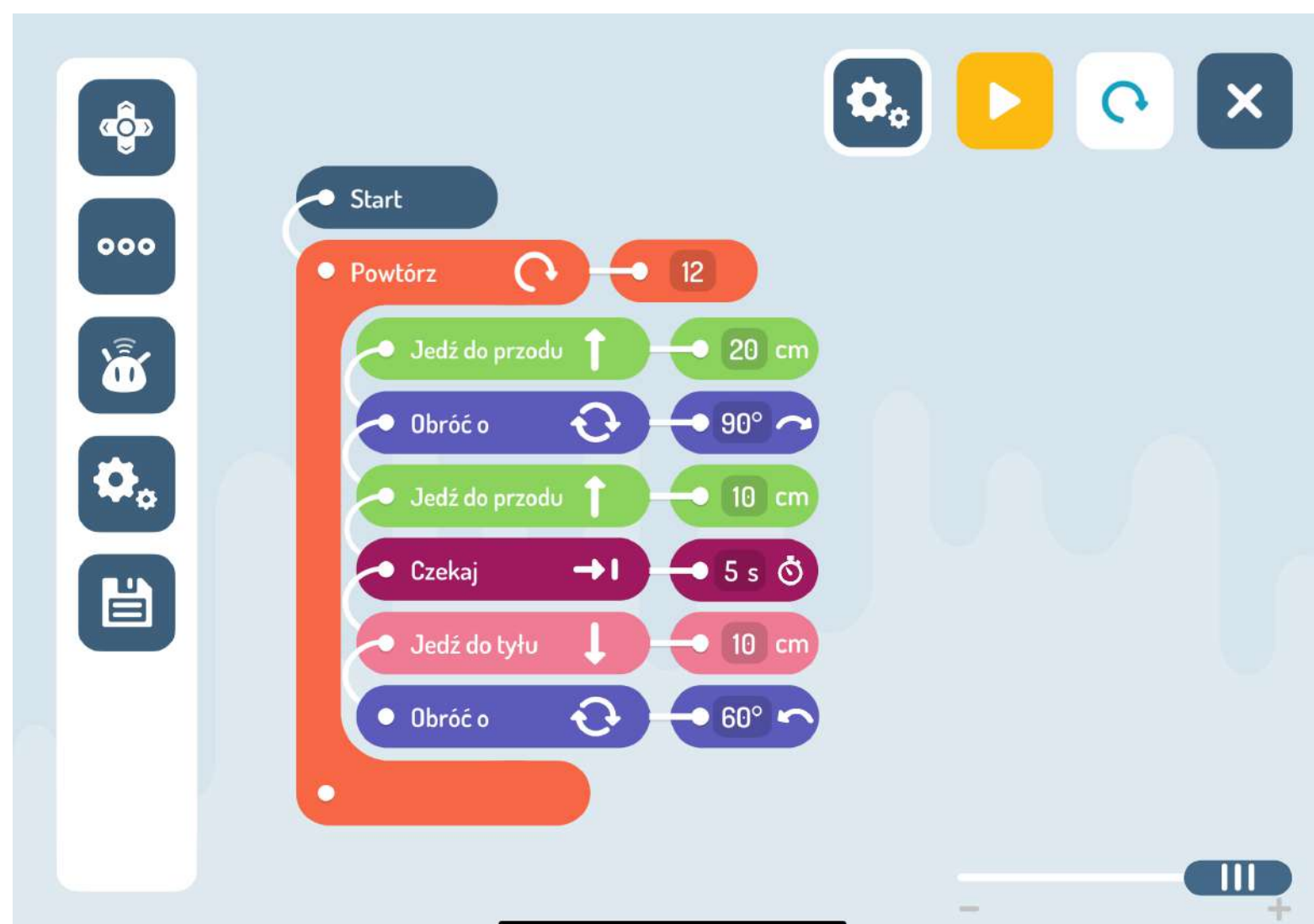
Polecenie 4. Obrót tylko trochę – o jedną godzinę (30°)

Jak inaczej?

Rozdaj każdemu dziecku wycięte koło – poproś, aby dzieci zaznaczyły na nim kolejne godziny. W środek koła uczniowie wbijają pinezkę ze strzałką, a następnie wykonują polecenia nauczyciela, np. przesunąć strzałkę o 180°, 90°, 360°, 30°. Zapytaj uczniów o to, co zrobić, żeby obrócić się o 60°?

Przetestujcie hipotezy (nawet te błędne) w dużej przestrzeni i precyzyjnie na tarczy zegara.

Nauczyciel zwraca uwagę dzieci na to, że robot aby narysować kreskę musi na tarczy zegara obrócić się o 90 stopni w prawo, a następnie przejechać pewien odcinek. Kiedy po narysowaniu kreski cofnie się na tarczę zegara nadal będzie obrócony. Na każdym odcinku robot powinien obrócić się w prawo o 30 stopni. W tym momencie robot jest obrócony o 90 stopni, żeby osiągnąć wymagany obrót o 30 stopni w prawo, musimy obrócić Photona o 60 stopni w lewo ($90 - 60 = 30$).
Rysunek 1. Interfejs Photon Blocks

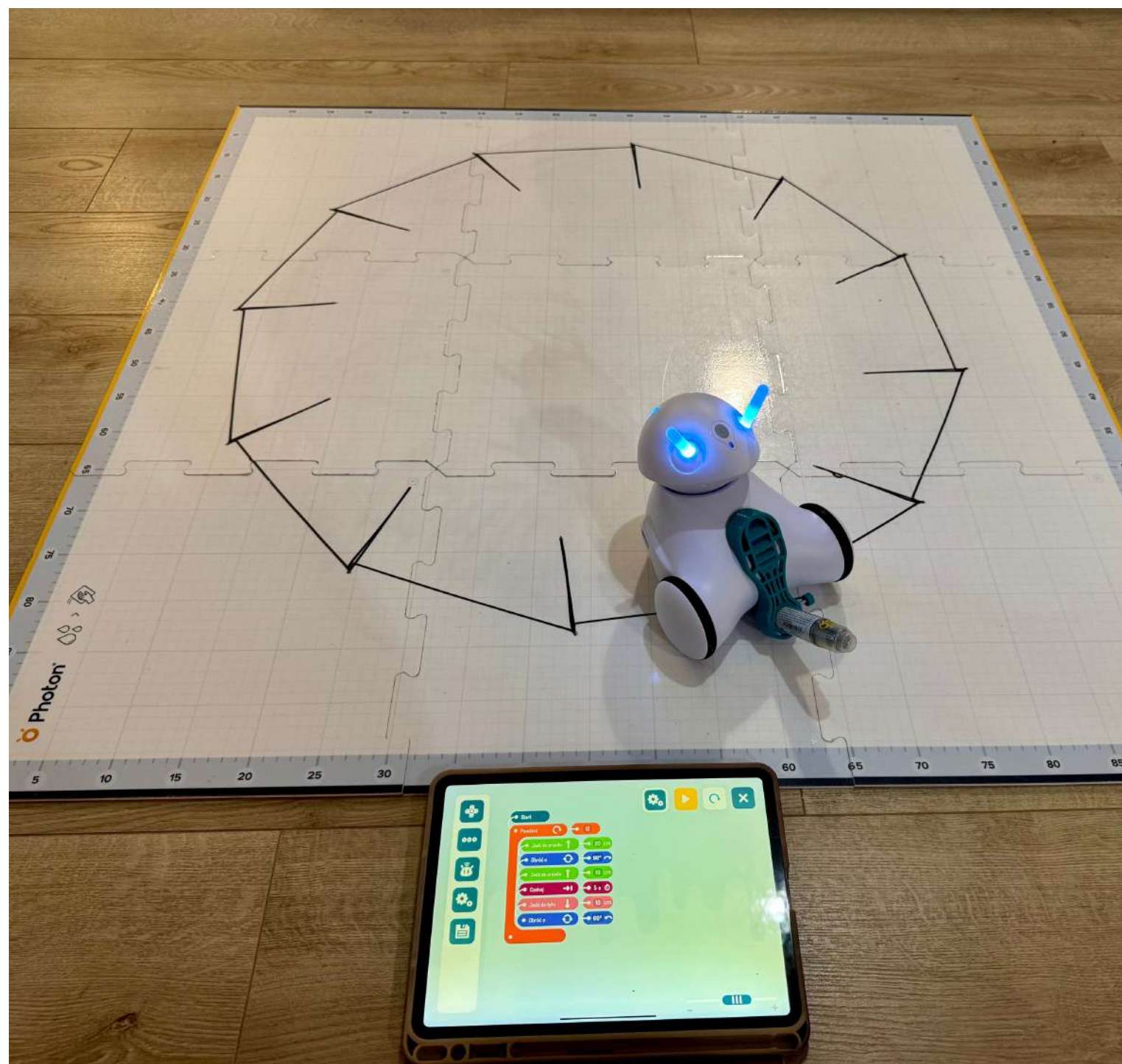


Źródło: opracowanie własne A. Świć.

Po zaprogramowaniu robota uczniowie mocują do niego pisak, a następnie uruchamiają program.

W momencie, w którym robot wykonuje pauzę uczniowie wpisują odpowiednią cyfrę. Opcjonalnie przy wybranych godzinach uczniowie mogą umieścić obrazki przedstawiające czynności, które o tych godzinach wykonujemy.

Tarcza zegara jest gotowa, potrzebne będą jeszcze wskazówki. Do zrobienia wskazówek można użyć klocków, patyczków lub wyciąć je z tektury. Uczniowie uzgadniają w parach jaką rolę pełnią wskazówki na zegarze, za co odpowiada długa wskazówka, a za co krótka. Następnie chętne pary prezentują swoje wytłumaczenie pozostałym dzieciom. Nauczyciel zwraca uwagę dzieci na przestrzenie między narysowanymi kreskami – prosi, aby uczniowie wytłumaczyli co oznacza przesunięcie krótszej wskazówki z jednej kreski na drugą (upływ jednej godziny), oraz przesunięcie w ten sam sposób dłuższej wskazówki (upłynięcie pięciu minut).
Rysunek 2. Narysowany przez Photona zegar



Źródło: opracowanie własne A. Świć.

Rysunek 3. Zegar z ułożonymi wskazówkami i godzinami



Źródło: opracowanie własne A. Świć.

Prowadzący zajęcia pokazuje uczniom krążki „Minuty 1-2”, prosi uczniów o rozłożenie ich na tarczy zegara. Następnie stawia robota na godzinie 12 i prosi uczniów, żeby zaprogramowali go w taki sposób, żeby przejechał np.: na miejsce pokazujące zmianę czasu o 10, 15 lub 35 minut.

Rysunek 4. Krążki „Minuty 1-2”.



Źródło: opracowanie własne A. Świć.

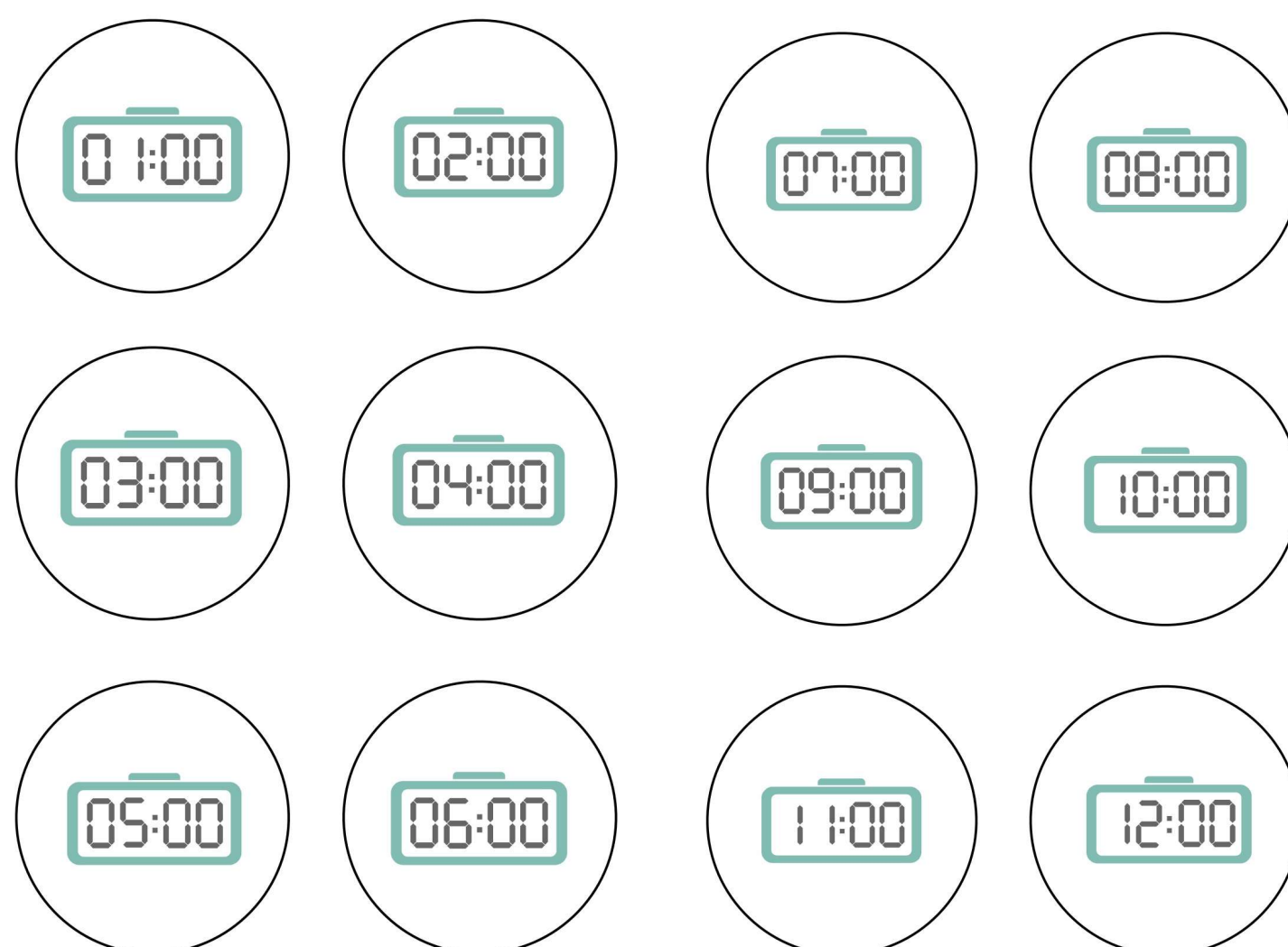
Nauczyciel prosi uczniów, żeby na losowych polach na macie do kodowania rozłożyli krążki z materiału pomocniczego „Minuty 1-2”. Następnie, żeby wybrali pole, z którego wystartuje robot.

Prowadzący zajęcia tłumaczy dzieciom na czym będzie polegało zadanie. Należy zaprogramować robota w taki sposób, żeby przejechał przez wszystkie pola, na których umieszczone są krążki z zegarami, zaczynając od tego, który obrazuje najkrótszy upływ czasu, a kończąc na tym, który obrazuje najdłuższy upływ czasu. Do wykonania zadania uczniowie ponownie wykorzystają interfejs Photon Blocks.

Zakończenie zajęć:

Nauczyciel układa obok narysowanej tarczy zegara krążki z elektronicznymi zegarami. Każdy krążek leży obrazkiem do dołu. Chętni uczniowie podchodzą, losują krążek, sprawdzają jaka godzina zapisana jest na zegarze, a następnie tak układają wskazówki na zegarze narysowanym przez robota, żeby pokazywały wylosowaną godzinę.

Rysunek 5. Tarcze zegara elektronicznego



Źródło: opracowanie własne A. Świć.

Na zakończenie zajęć nauczyciel rozdaje uczniom karteczki zielone i żółte. Na zielonych uczniowie mają napisać, co na zajęciach podobało im się najbardziej, na żółtych co jest dla nich niezrozumiałe. Następnie uczniowie przyklejają kartki w odpowiednim miejscu na tablicy – pod hasłem: „to mi się podobało” – zielone kartki, pod hasłem „tego jeszcze nie rozumiem” – kartki żółte.









CZTERY PORY ROKU

Czy marzec to miesiąc zimowy, czy wiosenny? Do której z pór roku przypiszemy grudzień? Czym jest cykliczność występująca w przyrodzie? Odpowiedzi na powyższe pytania poszukają uczniowie na zajęciach przeprowadzonych według zaproponowanego scenariusza. Rozmowa o cykliczności pomoże zrozumieć uczniom, jaka jest rola pętli w programowaniu.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Utrwalenie wiadomości na temat pór roku;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego i świat cyfrowy;
- Rozwijanie kompetencji miękkich, ze szczególnym uwzględnieniem umiejętności pracy zespołowej

Cele operacyjne:

Uczeń podczas zajęć:

- wymieni nazwy miesięcy we właściwej kolejności i przypisze je do odpowiednich pór roku;
- wyjaśni czym jest cykliczność występująca w przyrodzie;
- stworzy obrazek na podstawie podanych współrzędnych;
- rozwiązuje zagadki logiczne;
- zaprogramuje robota Photon wykorzystując interfejs Photon Badge i Photon Blocks;
- programując roboty wykorzystuje pętle i funkcje;
- chętnie i zgodnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki lub małe kolorowe kartki, dowolny kalendarz, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

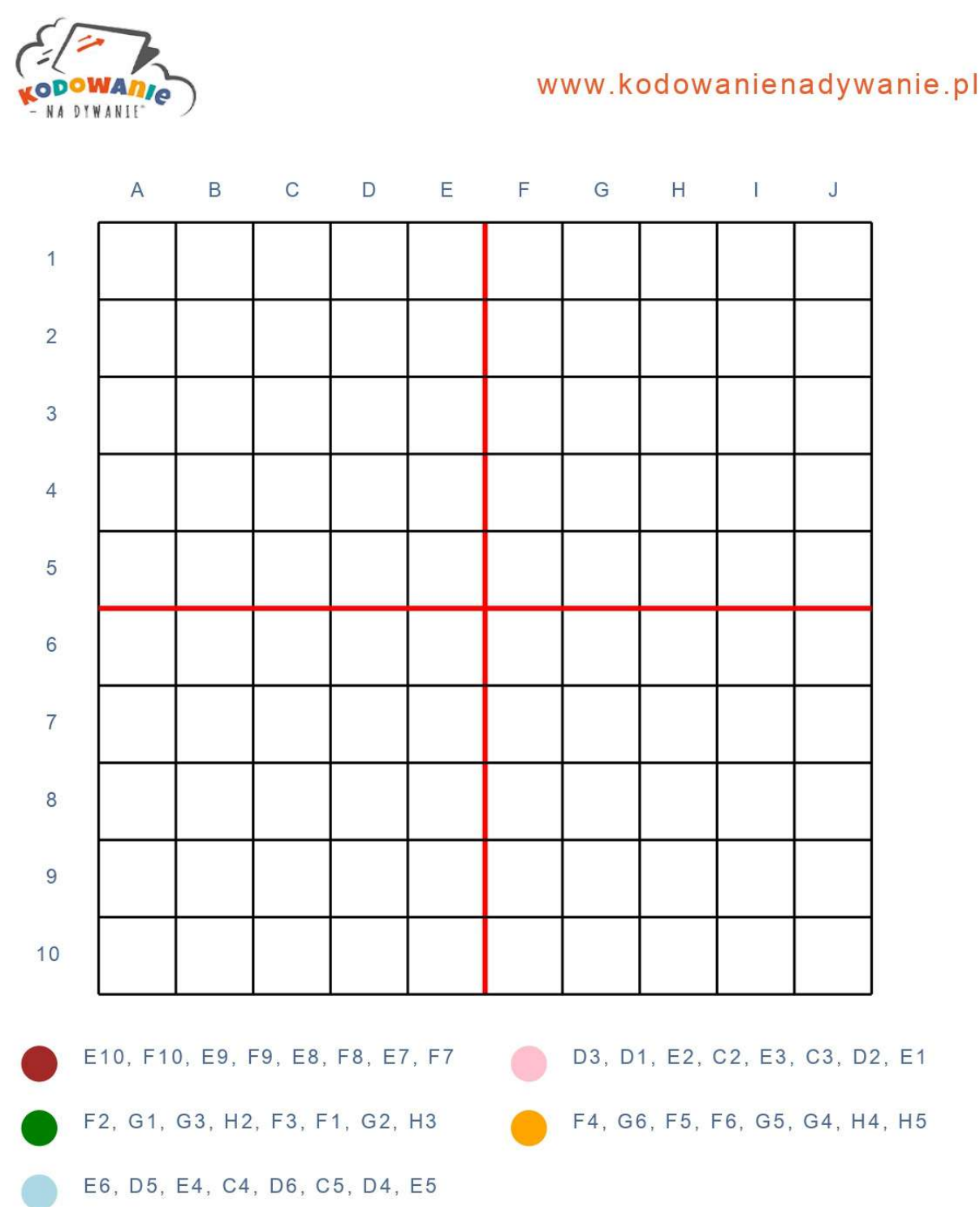
Ok. 45 – 90 minut (czas realizacji jest zależny od pracy uczniów – warto dopasować treści dotyczące pór roku do poziomu wiedzy uczniów).

Wprowadzenie

Nauczyciel wita się z dziećmi i tłumaczy uczniom, że temat dzisiejszych zajęć to tajemnicza zagadka. Kiedy uczniowie wykonają pierwsze przygotowane zadanie otrzymają odpowiedź, która naprowadzi ich na właściwe rozwiązanie i wskaże zagadnienie, które będzie tematem przewodnim dzisiejszego dnia.

Uczniowie rozkładają matę do kodowania klasyczną stroną do góry, obok maty ustawiają kubki. Nauczyciel daje dzieciom kartkę z rozpisanymi współrzędnymi poszczególnych pól i informacją, w jakich kolorach na tych polach powinny zostać postawione kubki lub ułożone kartki (Rysunek 1). Uczniowie układają na macie do kodowania obrazek.

Rysunek 1. Materiał pomocniczy „Drzewo współrzędne”



Źródło: opracowanie własne A. Świć.

Po ułożeniu wzoru (rys. 2) dzieci wyjaśniają co jest pokazane na obrazku i tworzą hipotezy dotyczące tematu zajęć. Uczniowie wyjaśniają, dlaczego korona drzewa zabarwiona jest na cztery różne kolory i co te kolory mogą oznaczać – który kolor wiąże się z którą porą roku i dlaczego? Nauczyciel dopuszcza dziecięce interpretacje. Nauczyciel zadaje pytanie kluczowe: Dlaczego liście zmieniają kolory w różnych porach roku?

Rysunek 2. Obrazek ułożony na macie do kodowania



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

Źródło: opracowanie własne A. Świć.

Część właściwa

Uczniowie dzielą się na zespoły (dzielimy grupę na tyle zespołów, ile mamy robotów). Nauczyciel rozdaje dzieciom obrazki przedstawiające drzewo w czterech odstępach: wiosennej, letniej, jesiennej i zimowej (rys. 3).

Rysunek 3. Obrazki dla dzieci – wiosna, lato, jesień, zima

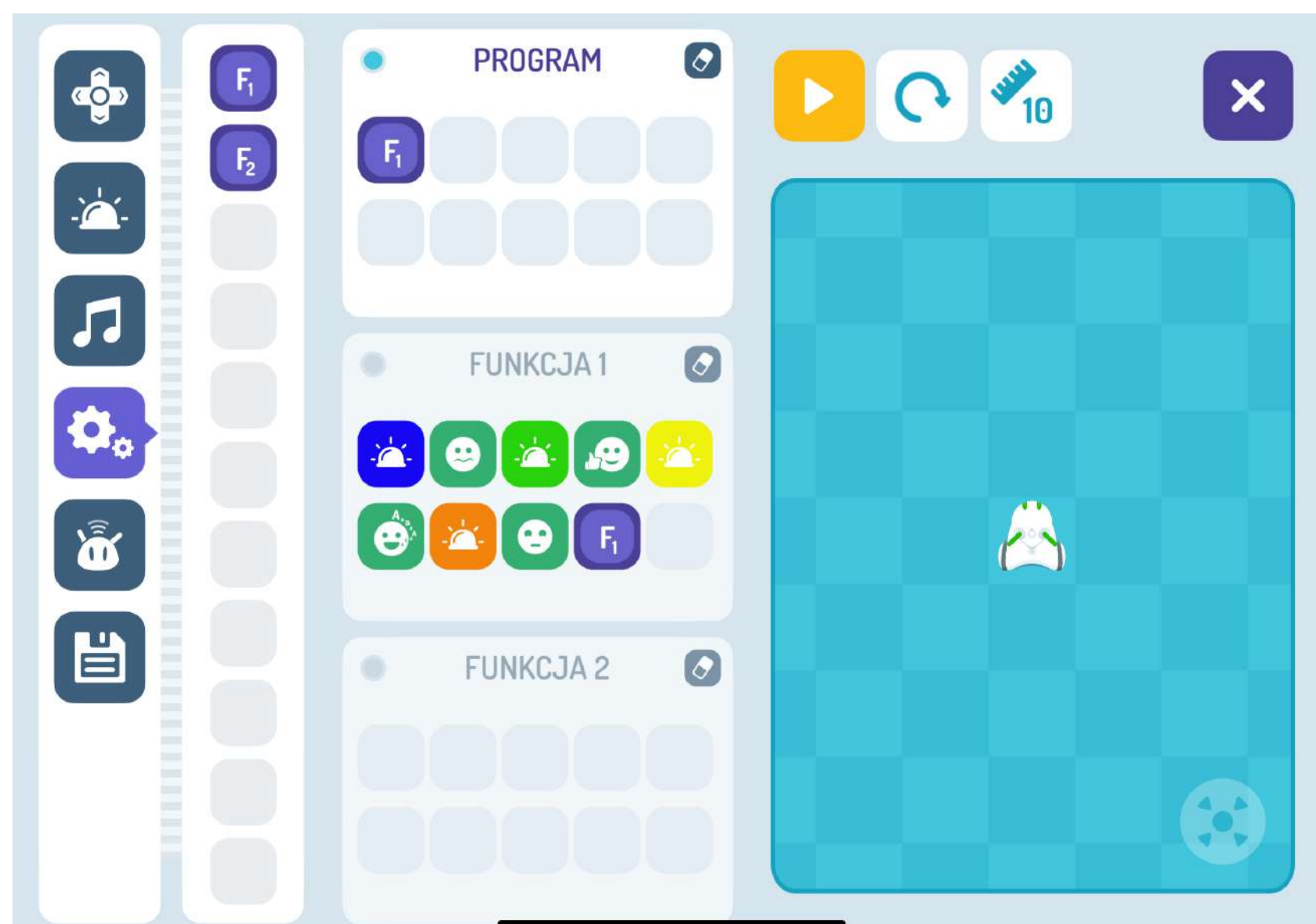


Źródło: opracowanie własne A. Świć.

Każdy zespół układa obrazki w taki sposób, żeby stworzyły koło. Następnie rozsuwa je na odległość, która umożliwi postawienie na środku robota.

Nauczyciel tłumaczy na czym będzie polegało zadanie: należy zaprogramować robota w taki sposób, żeby stale się obracał. W trakcie obrotu robot czterokrotnie zmieni barwę czułek i oczu. Kolor powinien być dopasowany do pory roku, obok której będzie przejeżdżał. Dodatkowo obracając się przy danej porze roku robot powinien wypowiedzieć jej nazwę. Zadanie uczniowie wykonają wykorzystując interfejs Photon Badge (rys. 4).

Rysunek 4. Interfejs Photon Badge



Źródło: opracowanie własne A. Świć.

Prowadzący zajęcia przypomina dzieciom w jaki sposób tworzy się programy w Photon Badge. Zwraca uwagę na to, że nazwy pór roku trzeba nagrać korzystając z zakładki z dźwiękami, w której umieszczona jest możliwość skorzystania z mikrofonu. Zwraca uwagę uczniów również na to, że robot powinien kręcić się stale, a we wskazanym interfejsie nie jest dostępna „pętla zawsze”. Nauczyciel pyta uczniów w jaki sposób można sprawić, aby robot kręcił się stale – sprawdza hipotezy uczniów. Jeśli uczniowie nie znajdą rozwiązania problemu nauczyciel wskazuje jak trzeba sobie poradzić z programowaniem – jeśli w polu funkcji dodamy symbol tej samej funkcji, to będziemy ją stale wywoływać.

Nauczyciel zaprasza dzieci do wykonania zadania. Jeśli będzie taka potrzeba, to pomaga poszczególnym zespołom w trakcie tworzenia programu (rys. 5).

Zespoły prezentują stworzone programy, wykonują jazdę próbną robotów.

Rysunek 5. Jazda próbna robotów



Źródło: opracowanie własne A. Świć.

Kolejne zadanie uczniowie wykonają wspólnie. Ponownie potrzebne będą obrazki symbolizujące pory roku. Dodatkowo potrzebne będą krążki z nazwami miesięcy. Dwanaście osób otrzymuje krążki z miesiącami. Dzieci nie patrząc na to jaki otrzymały miesiąc ustawiają się w jednej linii. Nauczyciel pyta dzieci, czy ich ustawienie pokrywa się z kolejnością w jakiej występują miesiące. Proponuje uczniom ustawienie się we właściwej kolejności, żeby to zrobić dzieci zastosują sortowanie bąbelkowe.

Sortowanie bąbelkowe (*bubble sort*) – co to?

To jeden z najprostszych algorytmów sortowania. Działa on przez wielokrotne przechodzenie przez listę i porównywanie sąsiednich elementów, a następnie zamienianie ich miejscami, jeśli są w złej kolejności. Proces ten powtarza się aż do momentu, gdy cała lista będzie posortowana.

Co robić krok po kroku:

1. Przeglądnij listę od początku.
2. Porównaj dwa sąsiadujące elementy.
3. Jeśli pierwszy jest większy od drugiego, zamieniasz je miejscami.
4. Przechodzisz do kolejnej pary i powtarzasz kroki 2–3.
5. Po każdej pełnej „przebieżce” największy element „wyływa” na koniec listy – stąd nazwa bąbelkowe.
6. Powtarzasz proces dla coraz krótszej części listy, aż nie będzie już potrzeby zamian²⁰.

[20] Ł. Dudziński, Sortowanie bąbelkowe, algorytm klasy $O(N^2)$, 2014: <https://strefakodera.pl/2013/03/sortowanie-babelkowe-algorytm-klasy-on2>, [dostęp 05.05.2025].

Pierwsza osoba z lewej strony sprawdza, czy osoba, która stoi obok niej ma miesiąc, który w kalendarzu jest przed jej miesiącem, czy po. Jeśli po, to zostają w takim samym ustawieniu, jeśli przed, to zamieniają się miejscami. Następnie pierwsza osoba porównuje swój miesiąc z miesiącem osoby trzeciej. W zależności od sytuacji zamieniają się miejscami lub nie. Procedurę „sprawdzania miejsc” powtarzamy dotąd, aż wszyscy będą stali zgodnie z kolejnością występowania miesięcy w kalendarzu, zaczynając od stycznia, a kończąc na grudniu.

Po utrwaleniu kolejności występowania miesięcy uczniowie ponownie zaprogramują robota. Zaczynają od ułożenia obrazków z porami roku (bez zostawiania w środku miejsca na robota). Następnie dookoła układają krążki z nazwami miesięcy, pamiętając, że niektóre z nich powinny być położone na łączeniu dwóch pór roku. Takimi miesiącami będzie marzec, czerwiec, wrzesień i grudzień (rys. 6).

Rysunek 6. Ułożenie obrazków z porami roku oraz nazwami miesięcy



Źródło: opracowanie własne A. Świć.

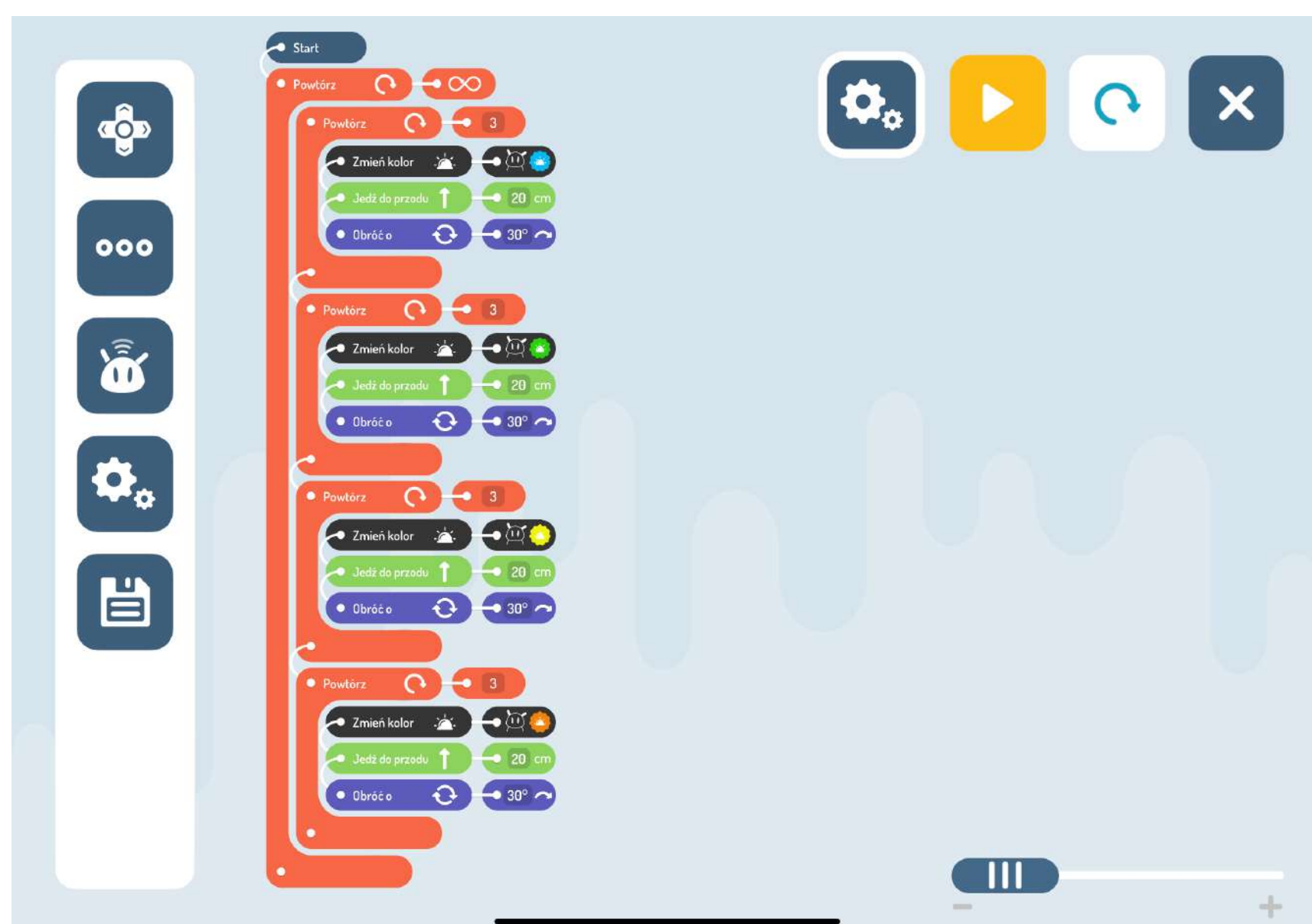
Nauczyciel tłumaczy na czym będzie polegało zadanie: robot będzie stale jeździł dookoła miesięcy. Czułki i oczy robota będą świeciły na kolory, które uczniowie przypiszą do poszczególnych pór roku. Do wykonania tego zadania wykorzystany zostanie interfejs Photon Blocks, który umożliwia zastosowanie pętli.

Uczniowie wspólnie z nauczycielem zastanawiają się jak ułożyć program, który pozwoli na poruszanie się robota dookoła miesiący. Robot nie musi jechać po okręgu. Przejechanie jednego pełnego roku należy podzielić na tyle części, ile jest miesięcy, czyli dwanaście.

Kiedy wiemy z ilu części będzie składał się jednorazowy przejazd możemy ustalić o jaki kąt za każdym razem powinien się obrócić robot. Koło to 360 stopni, przejazd zostanie podzielony na dwanaście części, więc za każdym razem kąt obrotu będzie wynosił 30 stopni. Długość odcinków między obrotami powinna być taka, żeby robot nie wjeżdżał na krążki z nazwami miesięcy, tylko poruszał się na zewnątrz rozłożonych grafik. Tworząc program uczniowie muszą jeszcze uwzględnić zmianę barw.

Uczniowie wspólnymi siłami tworzą program. W razie potrzeby w trakcie tworzenia wykonują próbne jazdy, żeby sprawdzić, czy robot porusza się zgodnie z założeniami.

Rysunek 7. Prawidłowo zaprogramowany interfejs Photon Blocks



Źródło: opracowanie własne A. Świć.

Kiedy program jest gotowy, uruchamiają go i sprawdzają jak zachowa się robot. Jeśli będzie taka konieczność, to uczniowie dokonują modyfikacji w utworzonym programie.

KRysunek 8. Próbną jazdą robota Photon

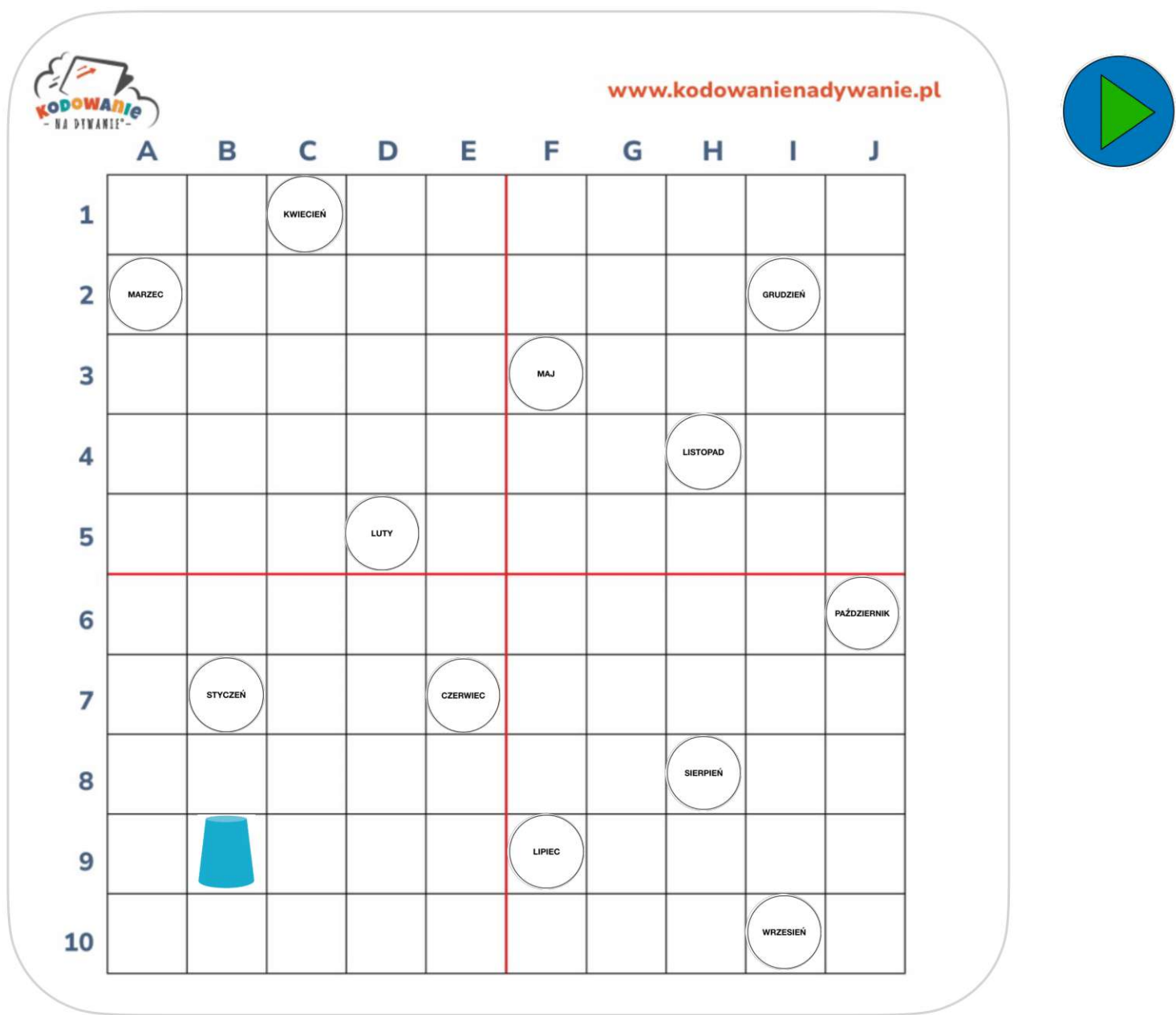


Źródło: opracowanie własne A. Świć.

Zakończenie zajęć

Jako utrwalenie zajęć uczniowie otrzymują kartę, na której znajduje się zadanie do pracy indywidualnej. Na kratownicy rozłożone są nazwy dwunastu miesięcy. Na jednym z pól umieszczony jest kubek. Zadanie będzie polegało na stworzeniu kodu, który poprowadzi kubek przez wszystkie miesiące z uwzględnieniem ich kolejności. Zanim dzieci przejdą do wykonania zadania przypominają w jaki sposób należy stworzyć kod, na co trzeba zwrócić szczególną uwagę (zaplanowanie trasy, tak by przejść od stycznia do grudnia; zamiana planów na kod np. „x kratek do przodu, x kratek w lewo”, testowanie kodu).

Rysunek 9. Mata do stworzenia kodu



Źródło: opracowanie własne A. Świć.

Nauczyciel pyta uczniów o ich wrażenia po zajęciach. Na tablicy zapisuje hasła: „co było dla mnie łatwe”, „co było dla mnie trudne”, „do tych aktywności chcę wrócić”. Uczniowie na karteczkach zapisują (lub ustnie podają) swoje refleksje i przyklejają w odpowiednim miejscu.

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



E10, F10, E9, F9, E8, F8, E7, F7



D3, D1, E2, C2, E3, C3, D2, E1



F2, G1, G3, H2, F3, F1, G2, H3



F4, G6, F5, F6, G5, G4, H4, H5



E6, D5, E4, C4, D6, C5, D4, E5









[illegible]

KOSMICZNA WYPRAWA

Mars, Ziemia, Wenus...jak wyglądają poszczególne planety? Która z planet znajduje się najbliżej Słońca, a która najdalej, która jest najmniejsza, a która największa... na te pytania poszukają odpowiedzi uczniowie podczas zajęć przeprowadzonych według zaproponowanego scenariusza.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Poszerzenie wiadomości na temat układu słonecznego;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie kompetencji miękkich, ze szczególnym uwzględnieniem umiejętności pracy zespołowej.

Cele operacyjne:

Uczeń podczas zajęć:

- wymienia nazwy planet we właściwej kolejności zaczynając od tej, która znajduje się najbliżej słońca;
- programuje robota Photon wykorzystując interfejs Photon Draw, Badge i Photon Blocks;
- Chętnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, duży arkusz papieru lub mata do rysowania za pomocą robotów, papier w żółtym kolorze lub dowolne żółte klocki, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

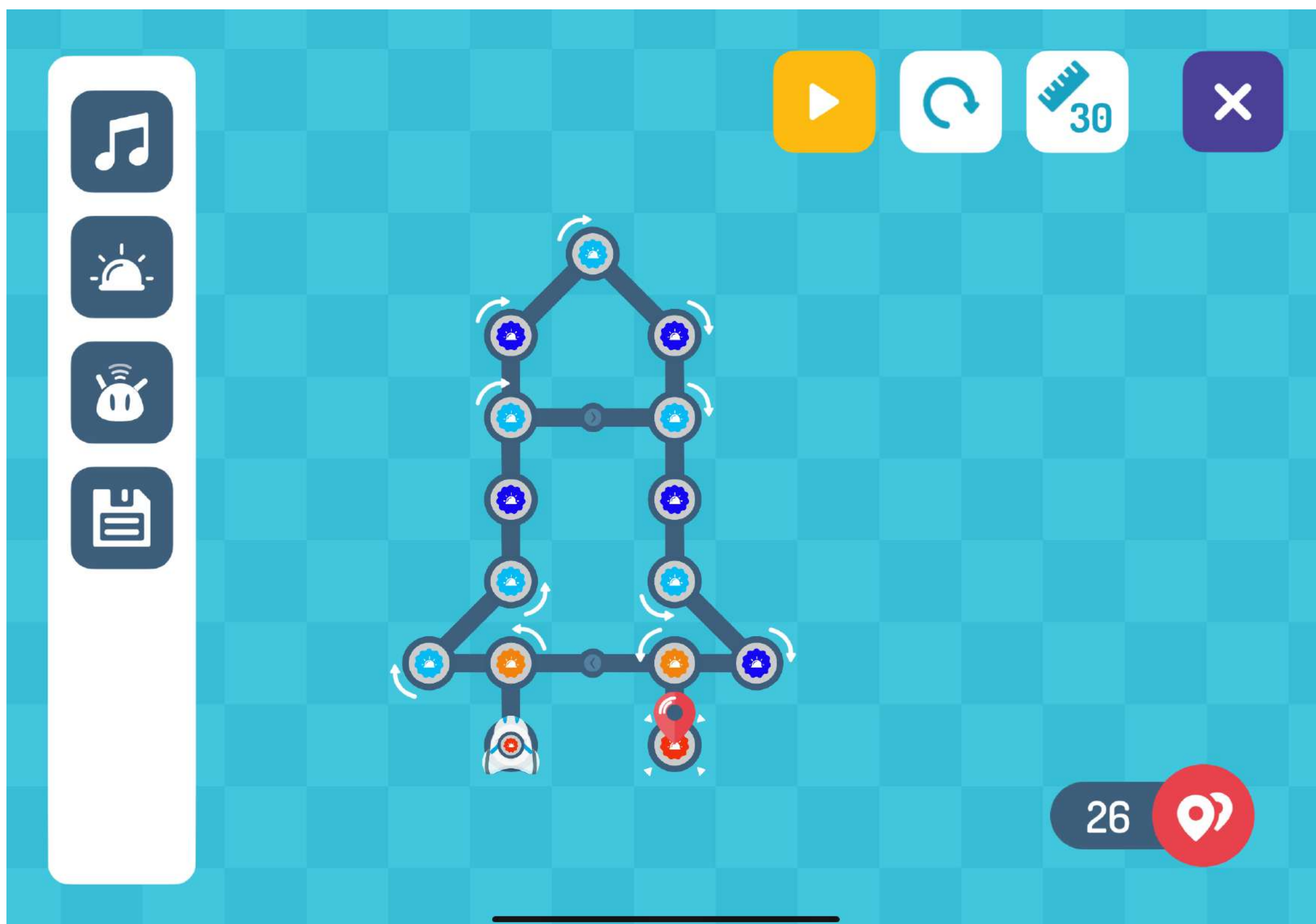
Ok. 45 minut

Wprowadzenie:

Nauczyciel wita się z dziećmi i zaprasza do wspólnej wyprawy, ale żeby dowiedzieć się jaka to będzie wyprawa uczniowie będą musieli odkryć schowane w klasie koperty. Nauczyciel wyjmuję ze słoika 12 patyczków z imionami dzieci (wcześniej nauczyciel przygotowuje słoik lub kubek, do którego wkłada patyczki z imionami wszystkich dzieci w klasie oraz 12 kopert – w każdej kopercie znajduje się jedno słowo, wszystkie słowa składają się na zdanie „Wyberzemy się w niesamowitą podróż kosmiczną i poznamy osiem planet Układu Słonecznego”. Koperty nauczyciel układa w różnych miejscach klasy). Uczniowie, których imiona wylosował nauczyciel będą przez niego sterowani słowami i komendami: np. „dwa kroki w lewo, trzy kroki prosto, obrót o 30 stopni” itp. tak, by dotrzeć do koperty (uczniowie mogą też sterować sobą nawzajem, wówczas dzielimy grupę na roboty i sterujących). Kiedy zebrane będą wszystkie koperty, pozostali uczniowie wydjmują schowane w nich słowa i układają w zdanie. Prezentują cel zajęć całej grupie.

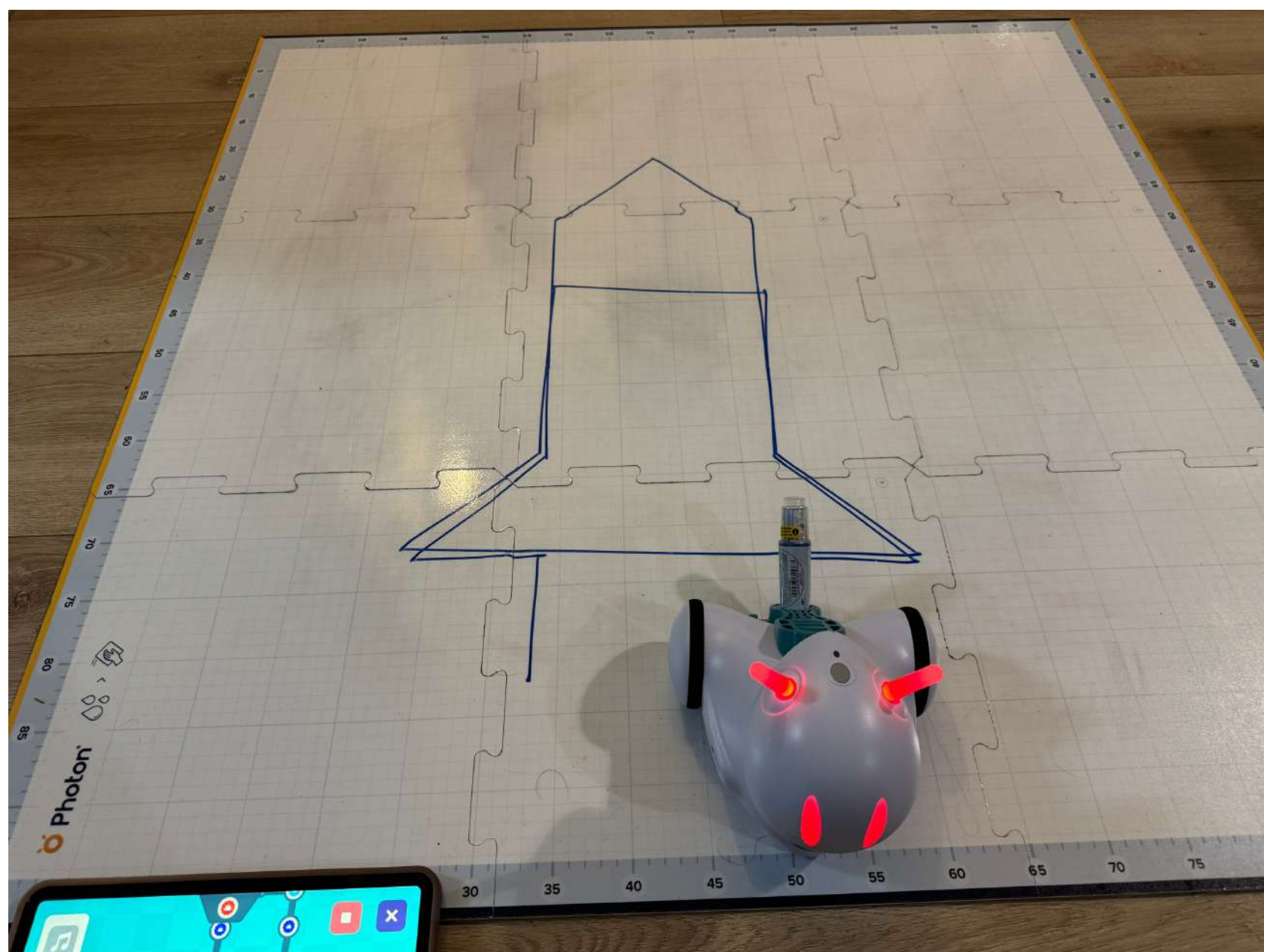
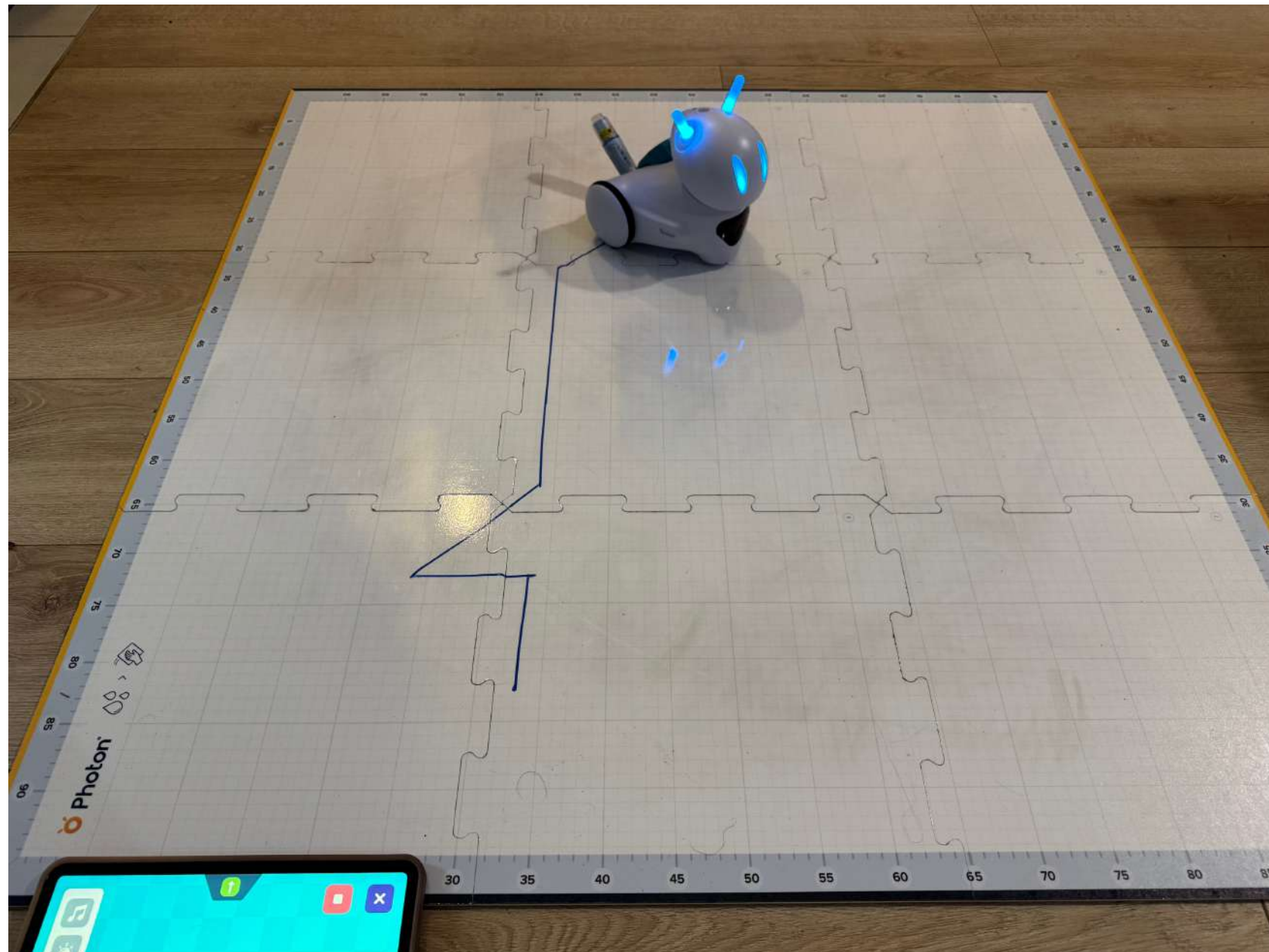
Nauczyciel pyta uczniów, co jest niezbędne do tego, aby odbyć wyprawę kosmiczną? Słuchamy propozycji uczniów, akcentując konieczność posiadania rakiety kosmicznej. Taką rakietę narysuję dla uczniów robot, muszą mu w tym tylko trochę pomóc. Do narysowania rakiety wykorzystany zostanie interfejs Photon Draw. Nauczyciel prosi dzieci, żeby rysując program uwzględniły też dodanie różnych barw (światło czułek i oczu).

Rysunek 1. Interfejs Photon Draw



Źródło: opracowanie własne A. Świć.

Rysunek 2. Robot Photon rysujący rakiety

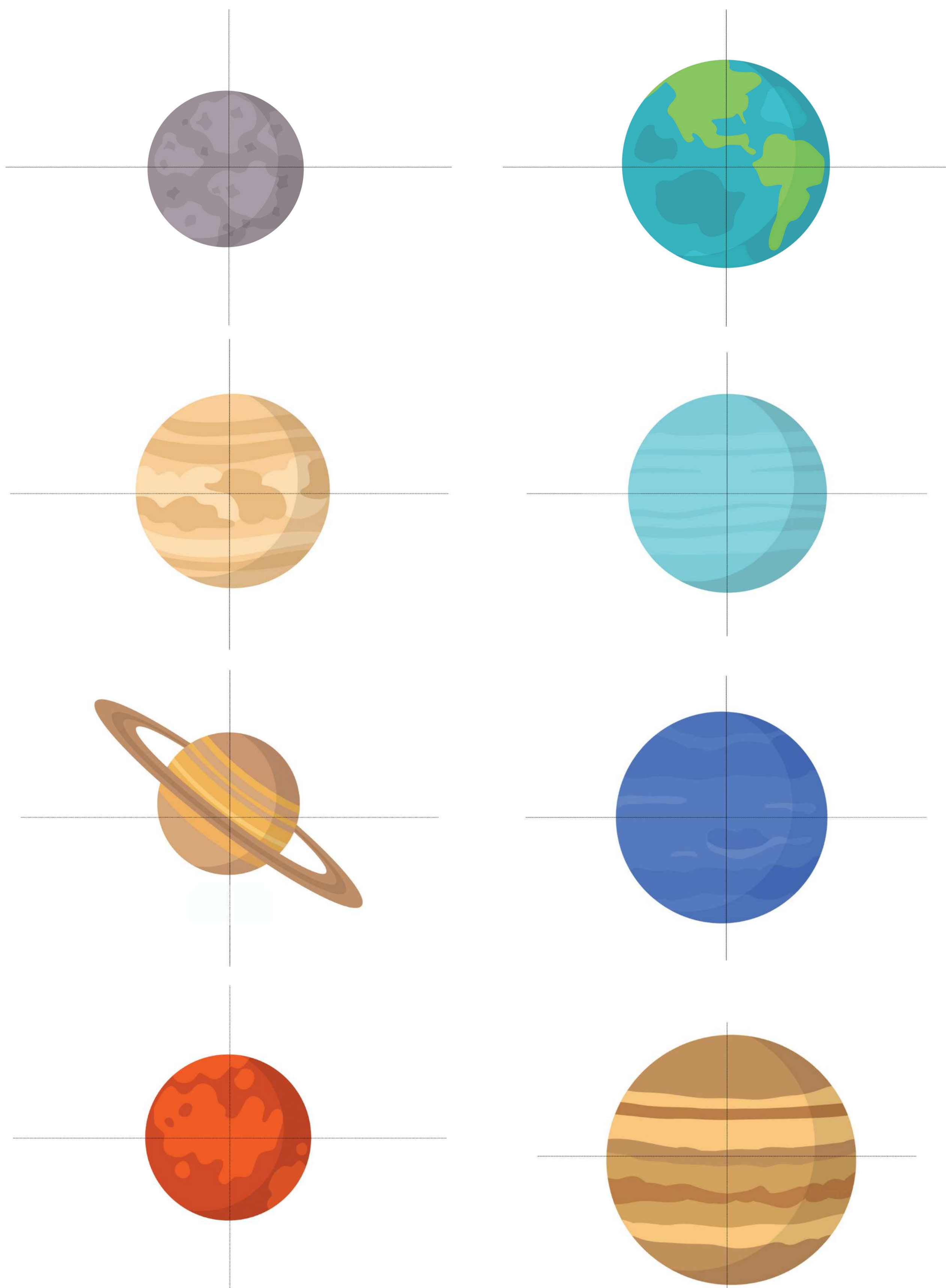


Źródło: opracowanie własne A. Świć.

Część właściwa:

Nauczyciel prezentuje dzieciom puzzle. Kładzie je na środku maty (wszystkie wymieszane). Prosi uczniów o rozdzielenie ich na osobne obrazki, a następnie ułożenie każdego obrazka w innym miejscu maty.

Rysunek 3. Złożone puzzle przedstawiające planety



Źródło: opracowanie własne A. Świć.

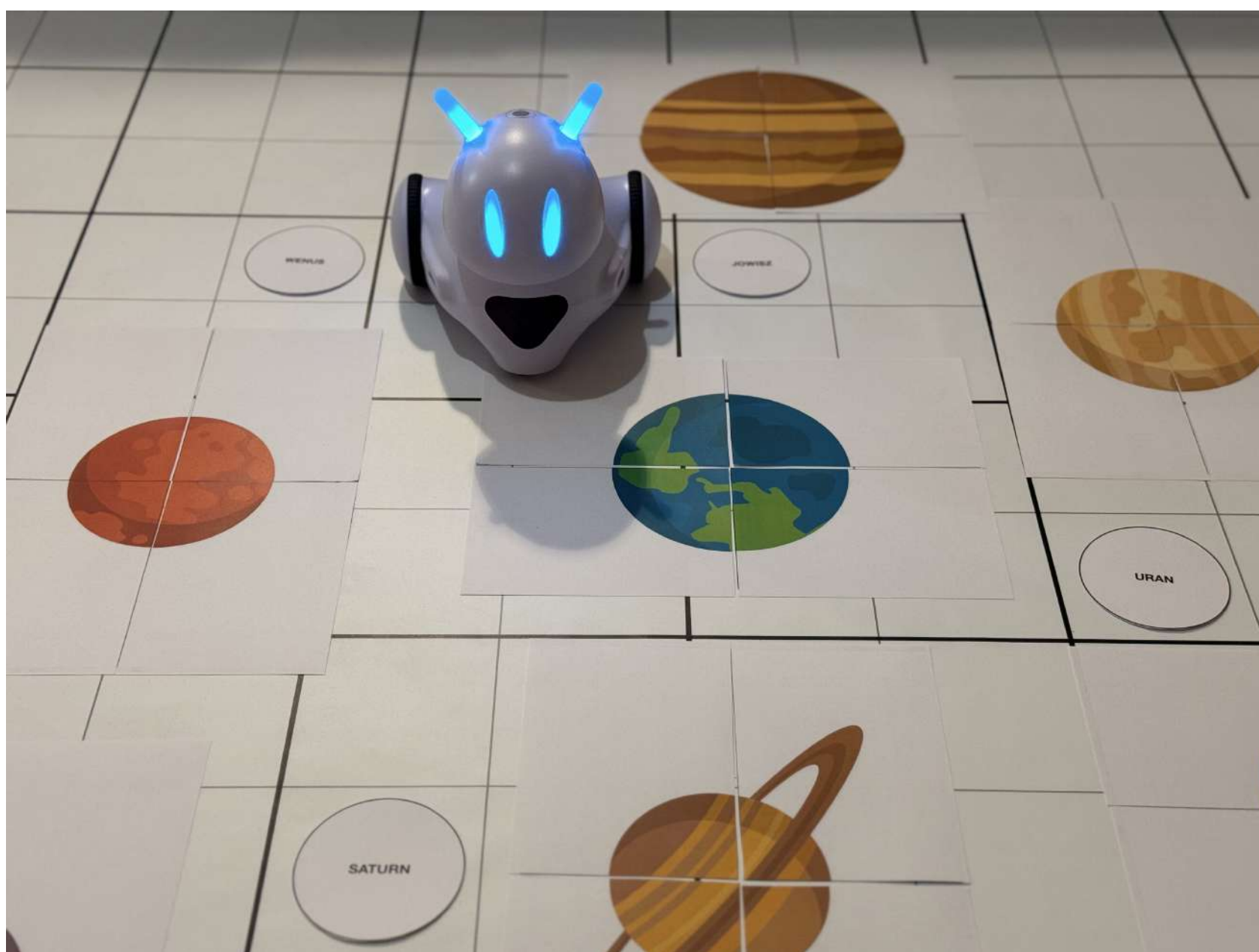
Rysunek 4. Pomieszane puzzle przedstawiające planety



Źródło: opracowanie własne A. Świć.

Po ułożeniu obrazków prowadzący zajęcia pyta, co się na nich znajduje, następnie na losowych polach maty kładzie krążki z nazwami planet. Zadaniem dzieci będzie wybranie jednej planety i połączenie jej z odpowiadającą jej nazwą (zadanie może zostać zrobione w Photon Badge lub Photon Blocks). Symbolem połączenia będzie przejazd robota od planety, do jej nazwy.

Rysunek 5. Robot Photon jeżdżący od planety do jej nazwy



Źródło: opracowanie własne A. Świć.

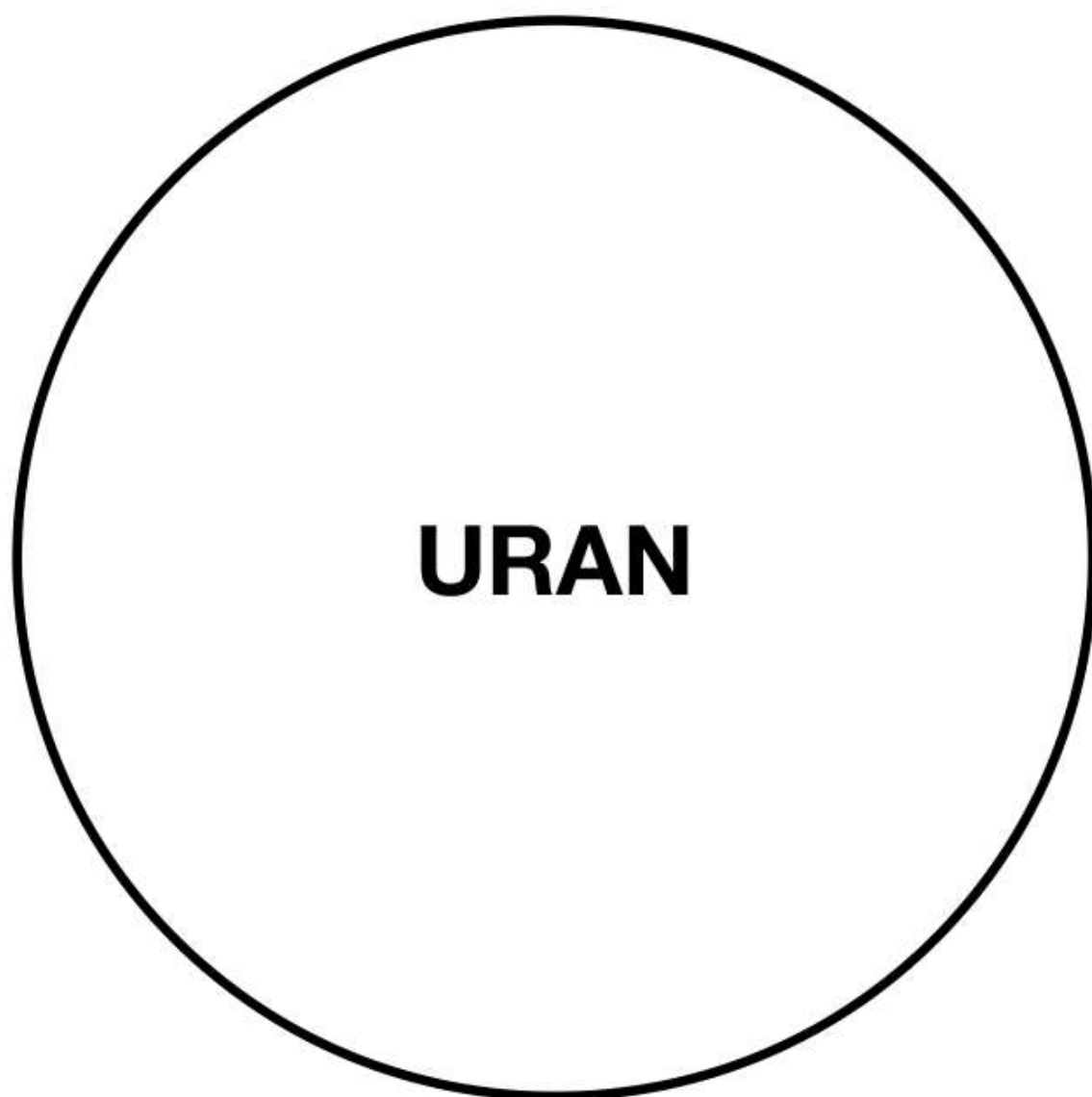
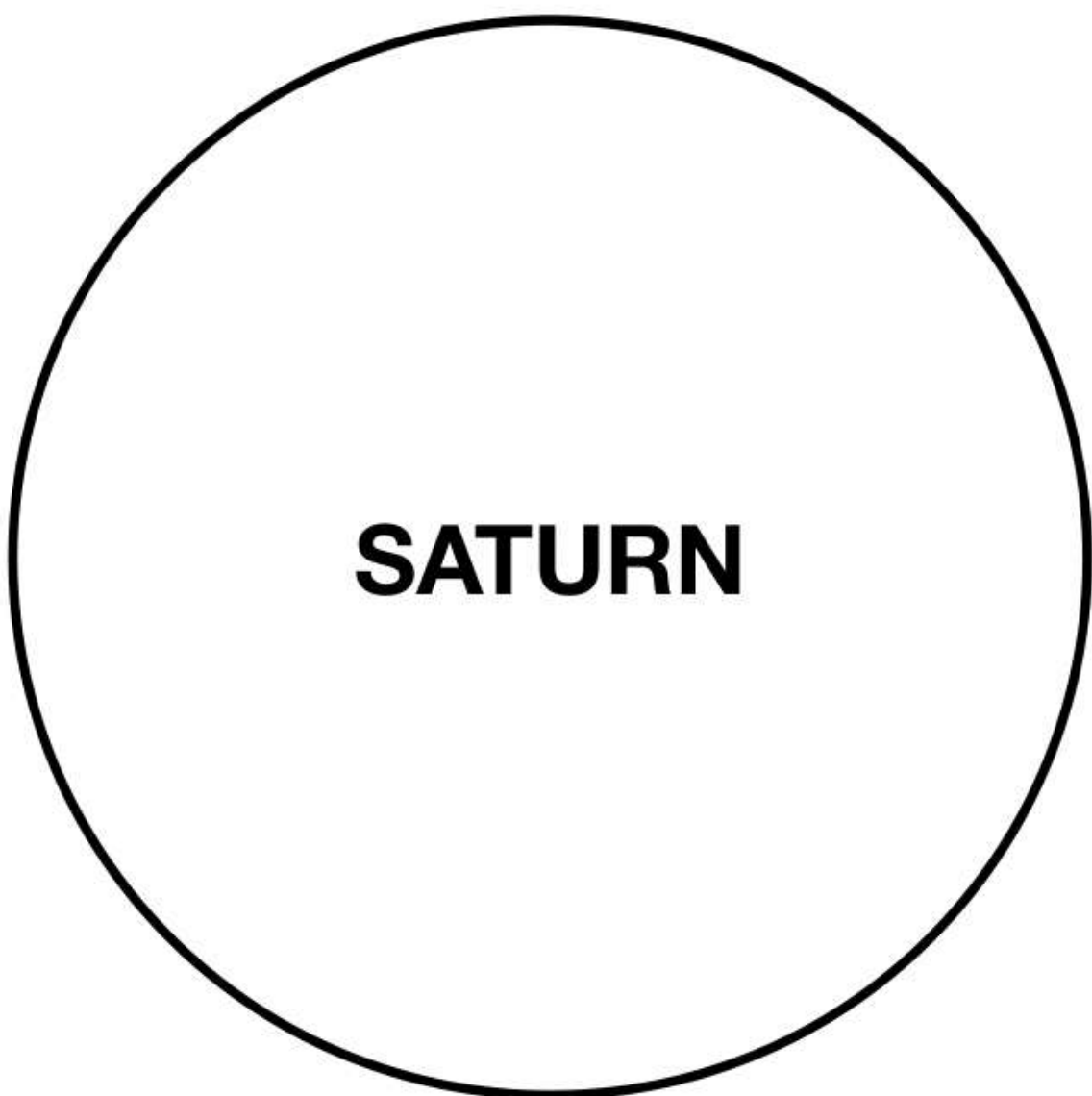
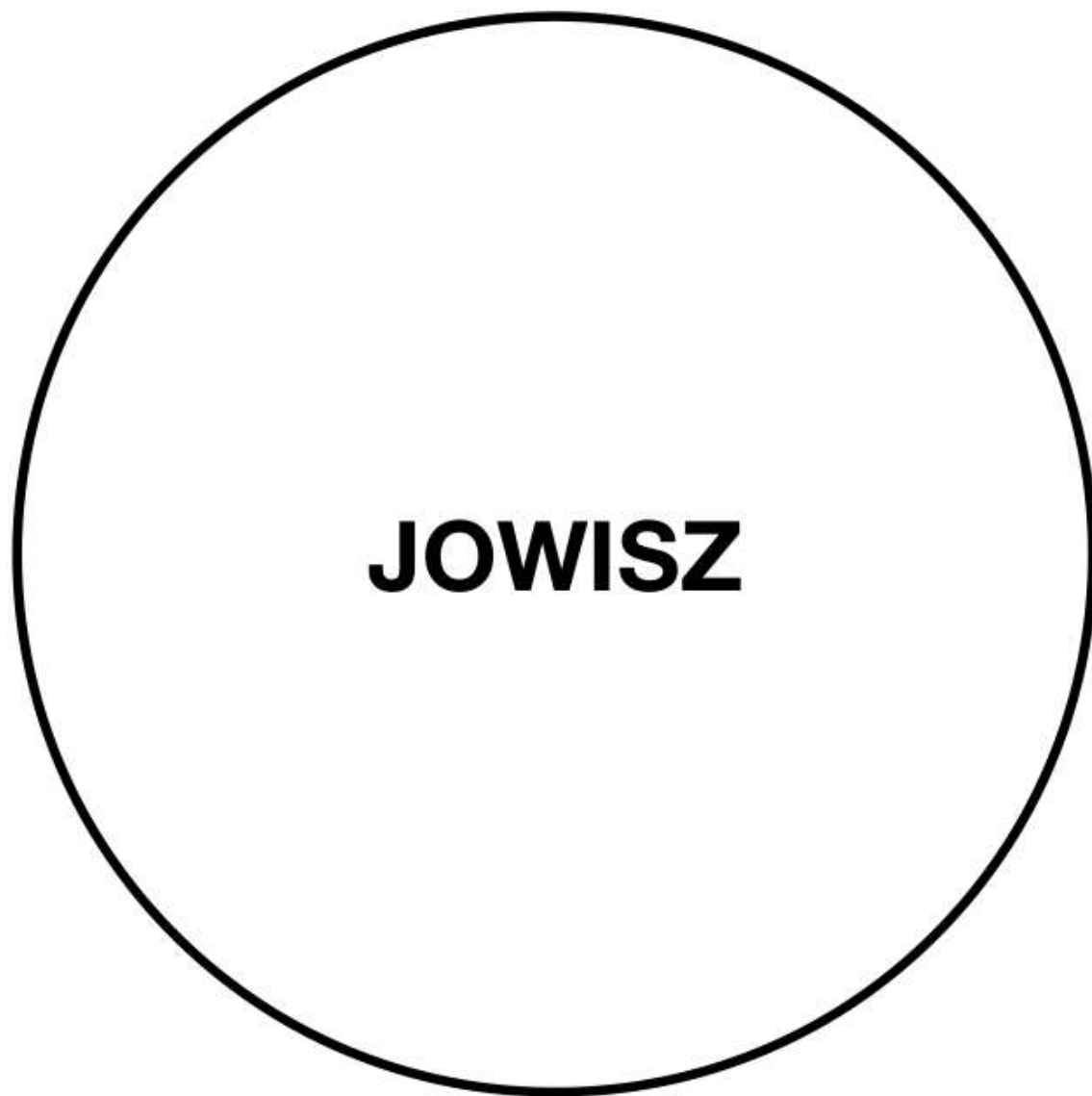
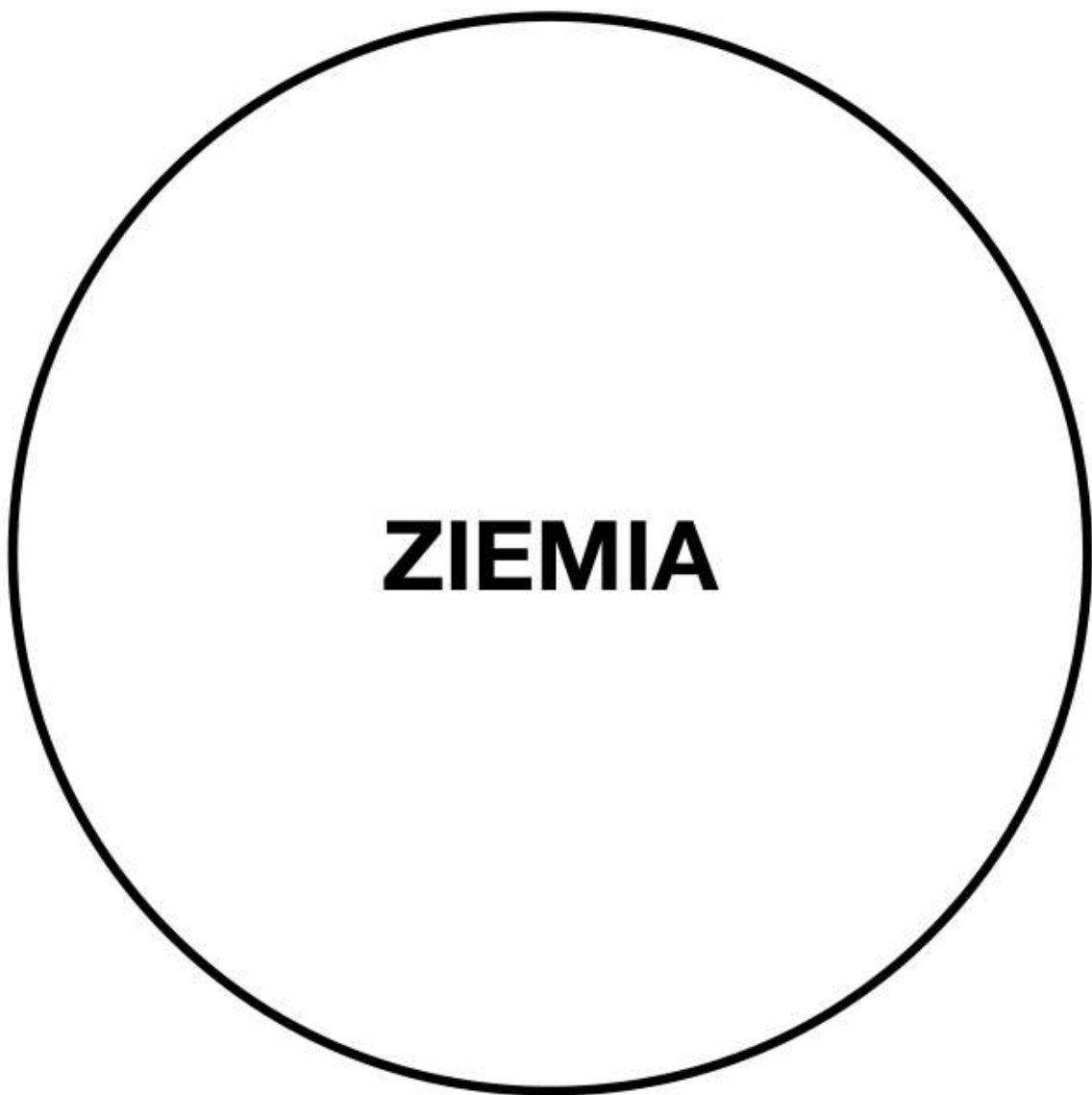
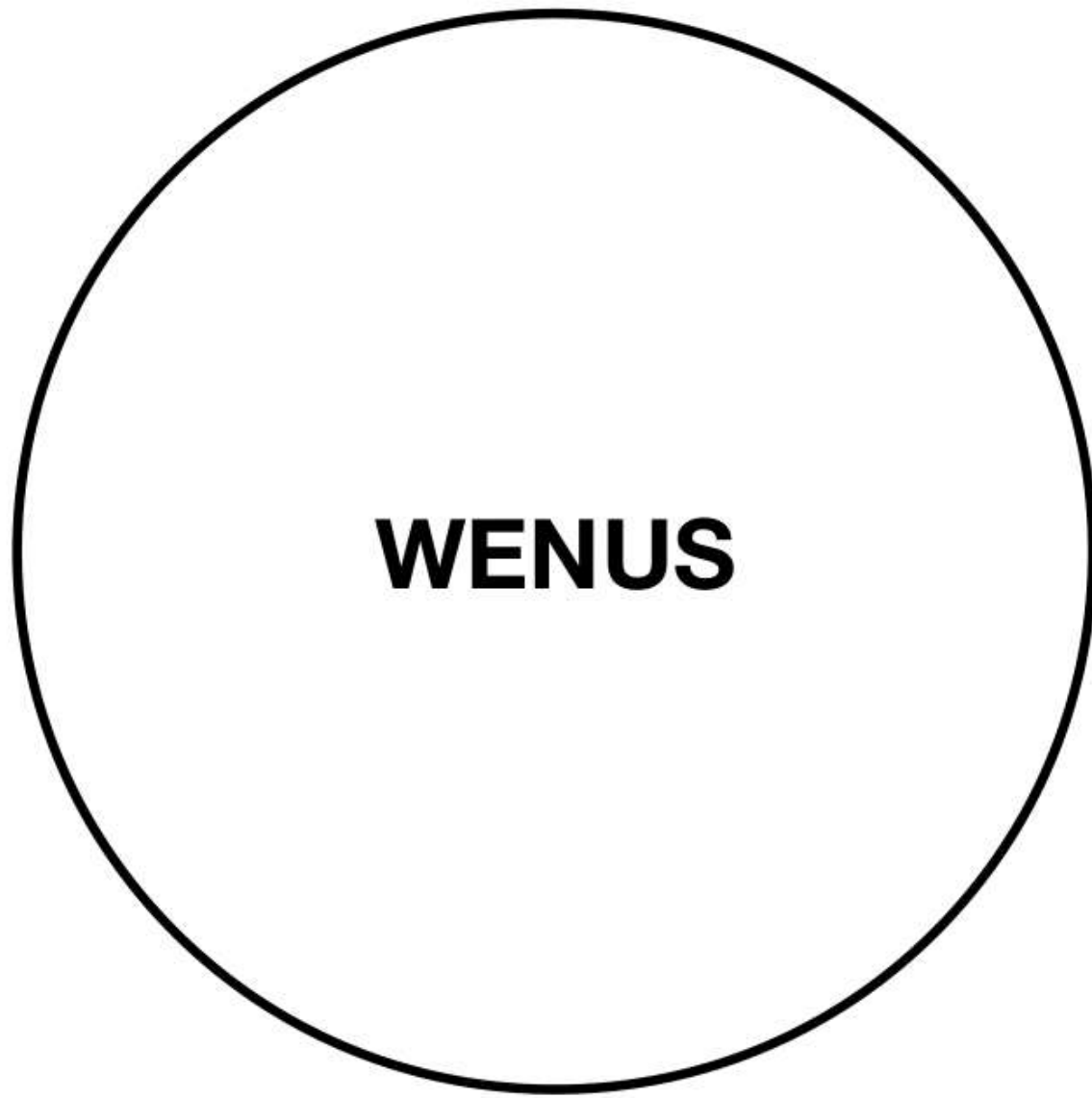
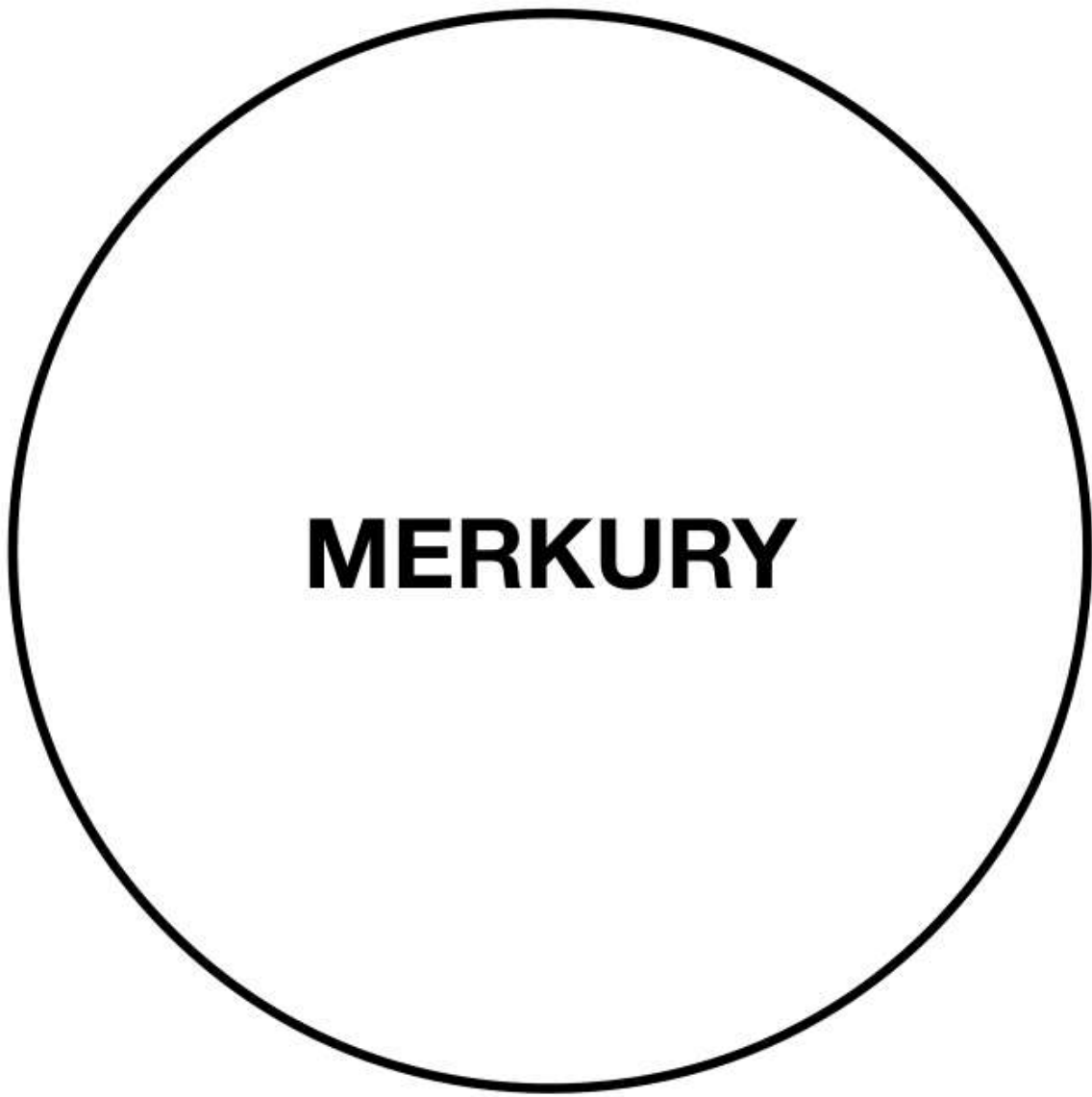
Nauczyciel dzieli klasę na kilka grup (tyle grup, ile jest robotów) i prosi o stworzenie programów. Po wykonaniu zadania grupy kolejno prezentują efekty swojej pracy.

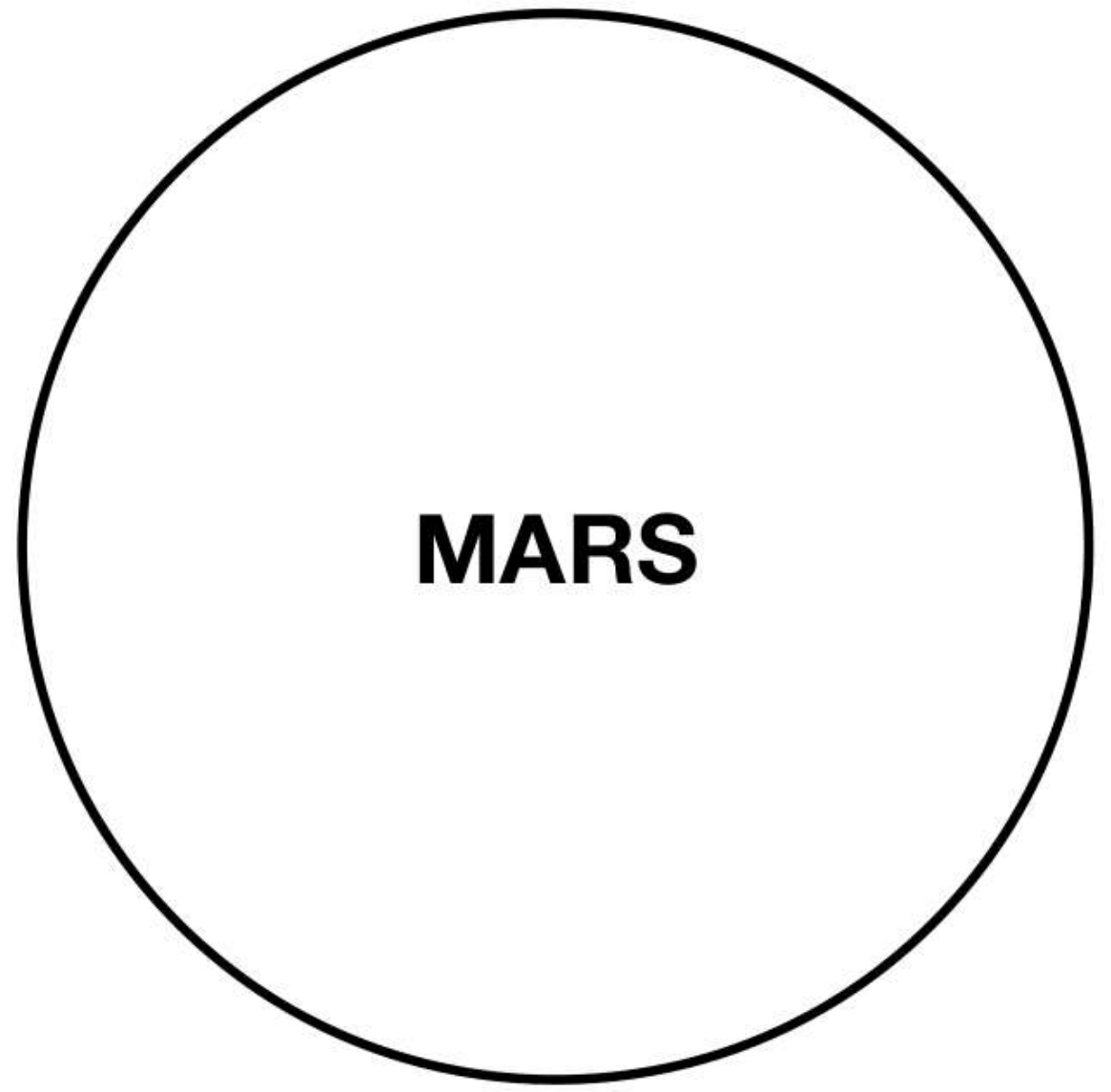
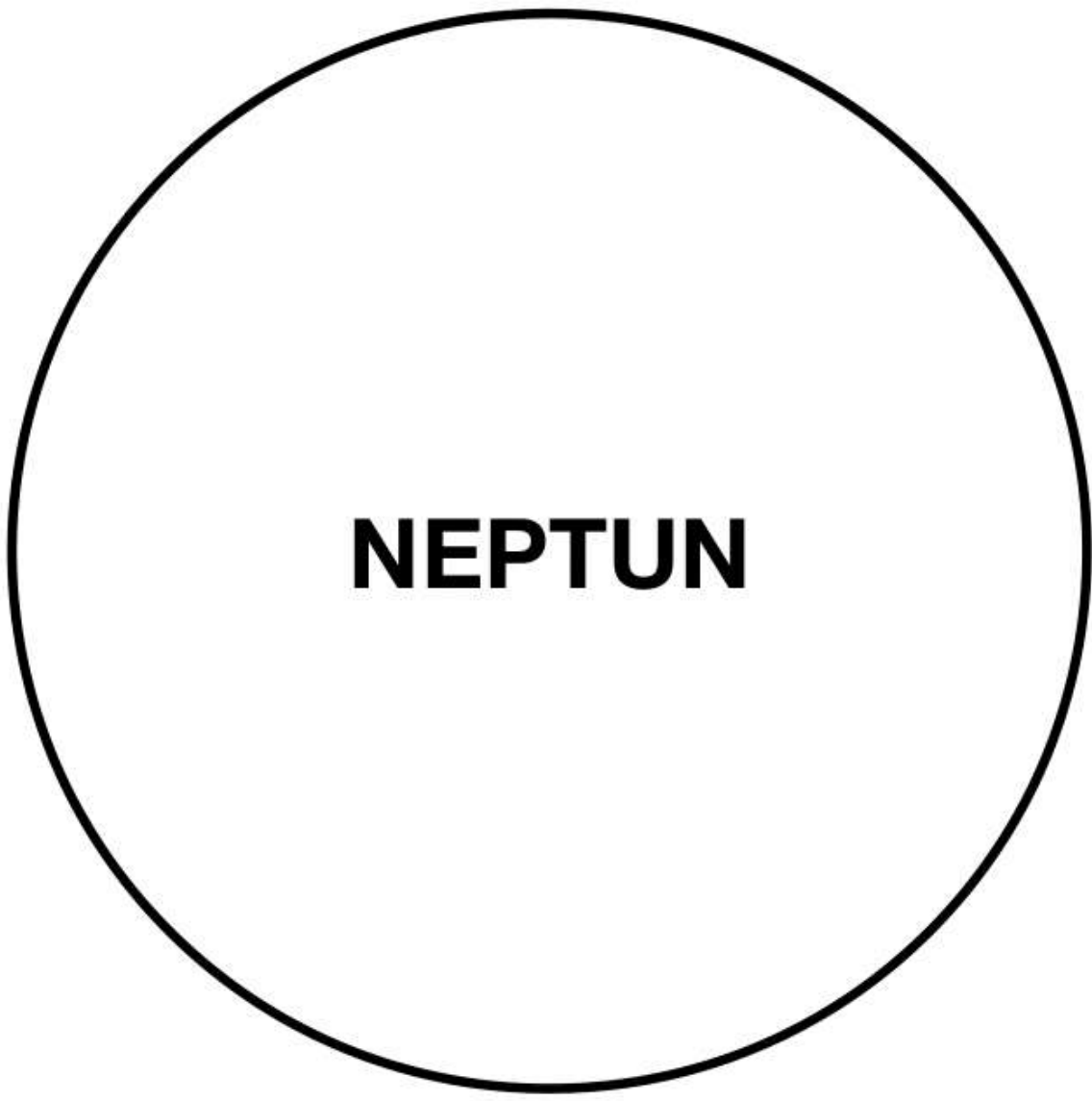
W kolejnym kroku nauczyciel tłumaczy dzieciom, że spróbują ułożyć na macie do kodowania Układ Słoneczny. Słońce może zostać ułożone z klocków lub wycięte z kolorowego papieru, planetami będą puzzle wykorzystane do poprzedniego zadania. Układając puzzle uczniowie postarają się zachować ich układ względem słońca.

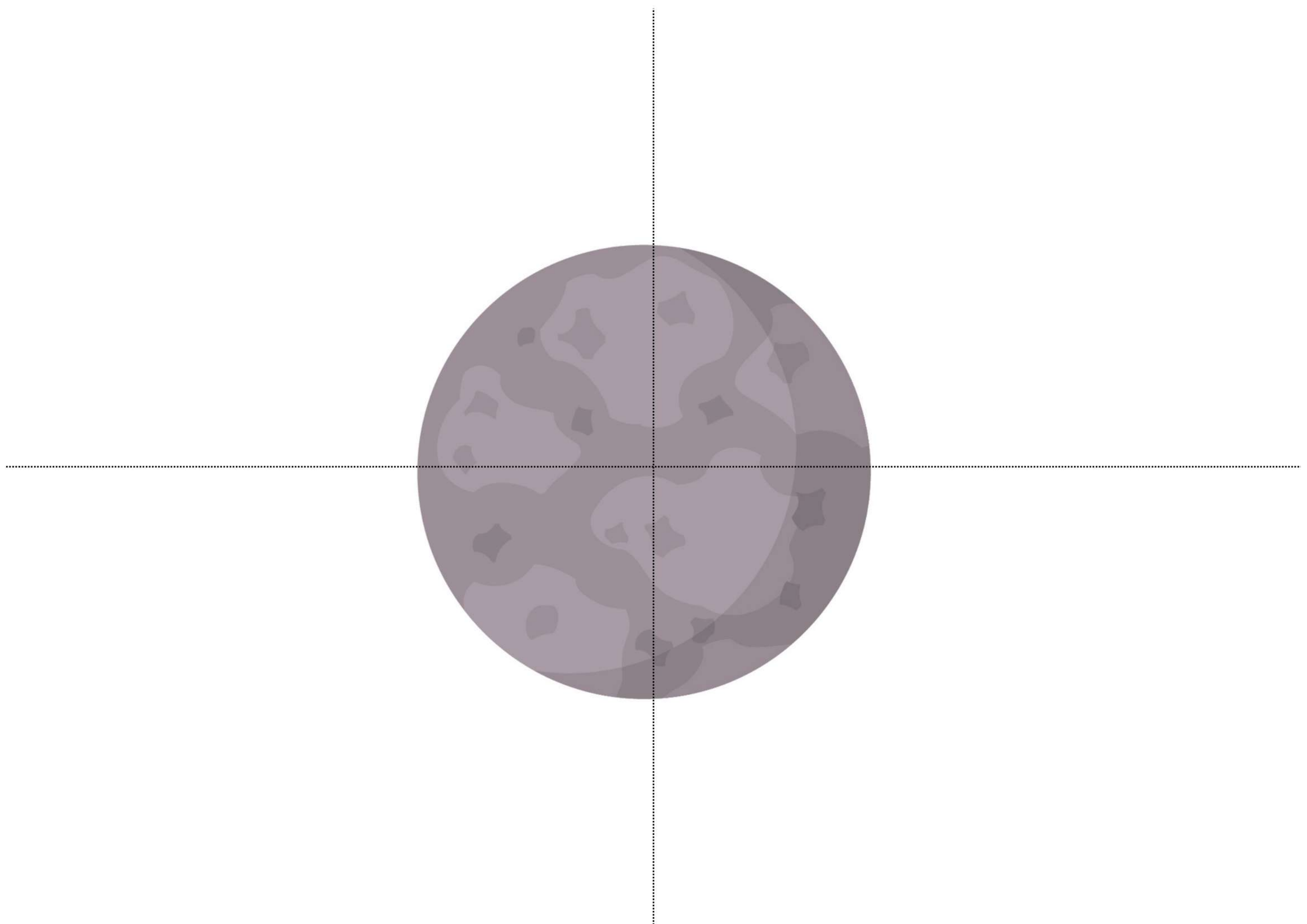
Po ułożonych planetach będzie poruszał się robot. Zacznie od tej, która jest najbliżej słońca, a skończy na tej, która jest od słońca najbardziej oddalona.

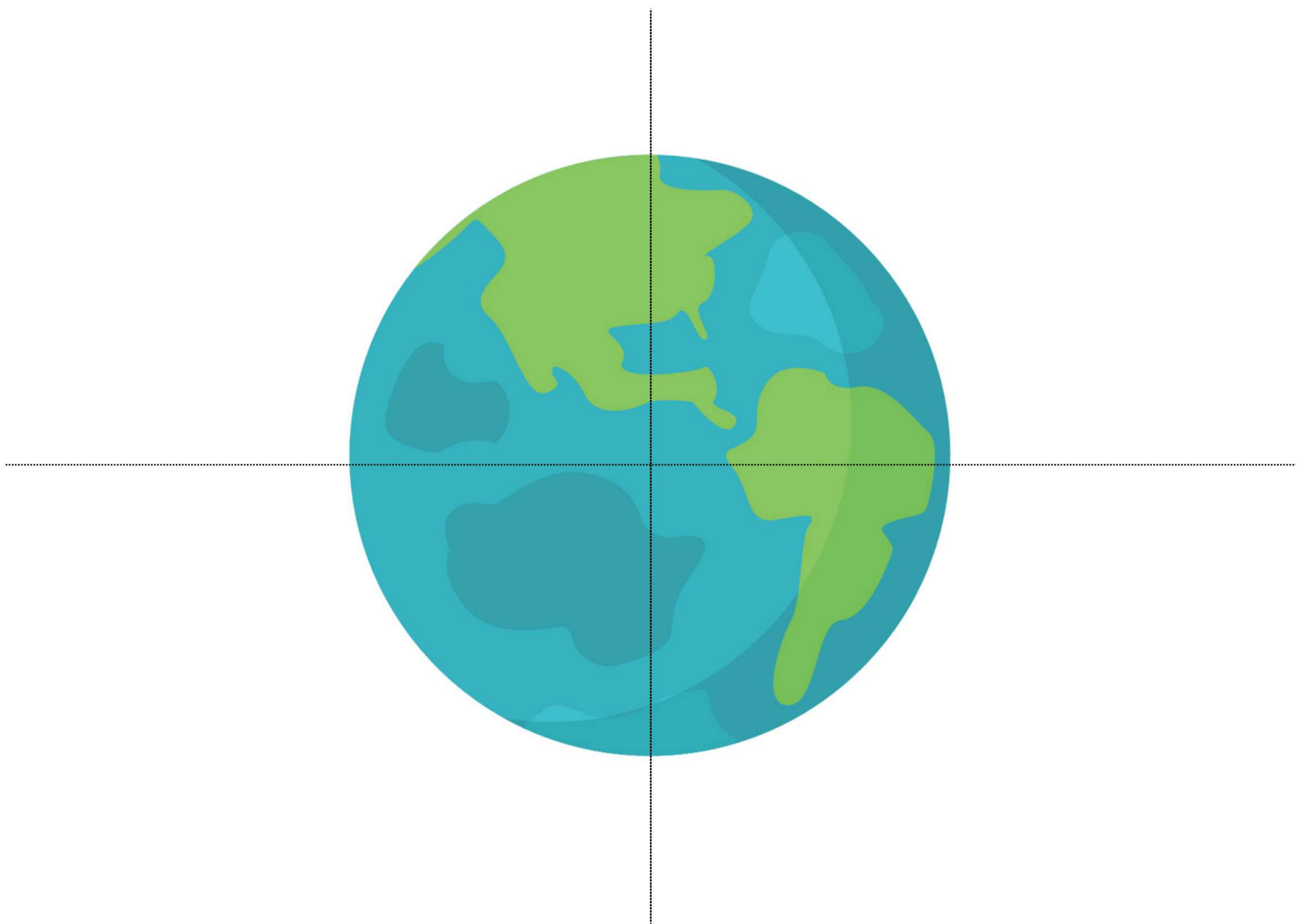
Zakończenie zajęć:

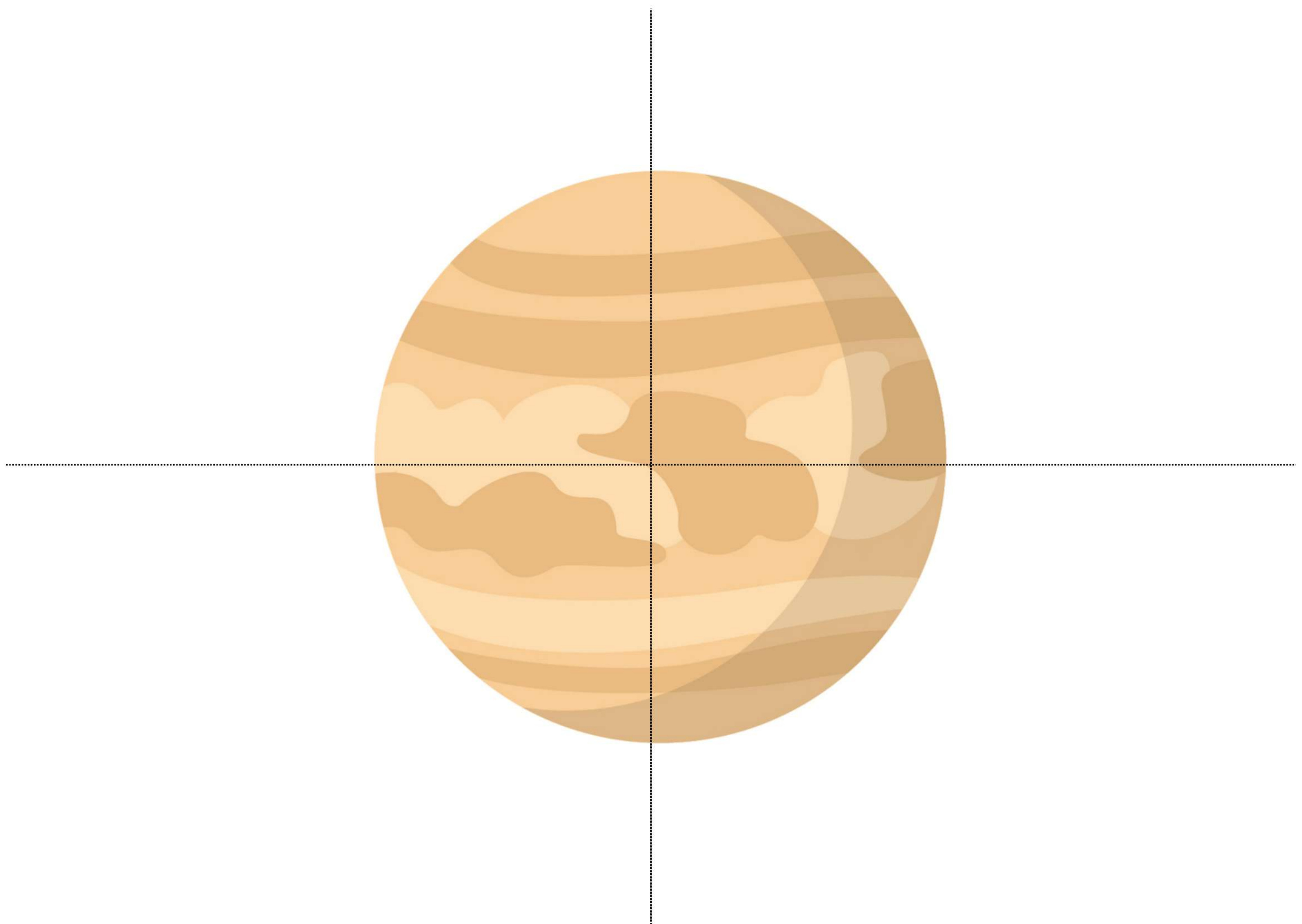
Nauczyciel prosi uczniów, aby na stworzonym przez siebie Układzie Słonecznym zaznaczyli na ile osiągnęli cel lekcji – poznali planety Układu Słonecznego. Wskazanie Słońca oznacza, że uczeń doskonale pamięta wszystkie planety w odpowiedniej kolejności – im dalej od Słońca, tym słabiej uczeń ocenia swoją wiedzę i umiejętności w tym zakresie.

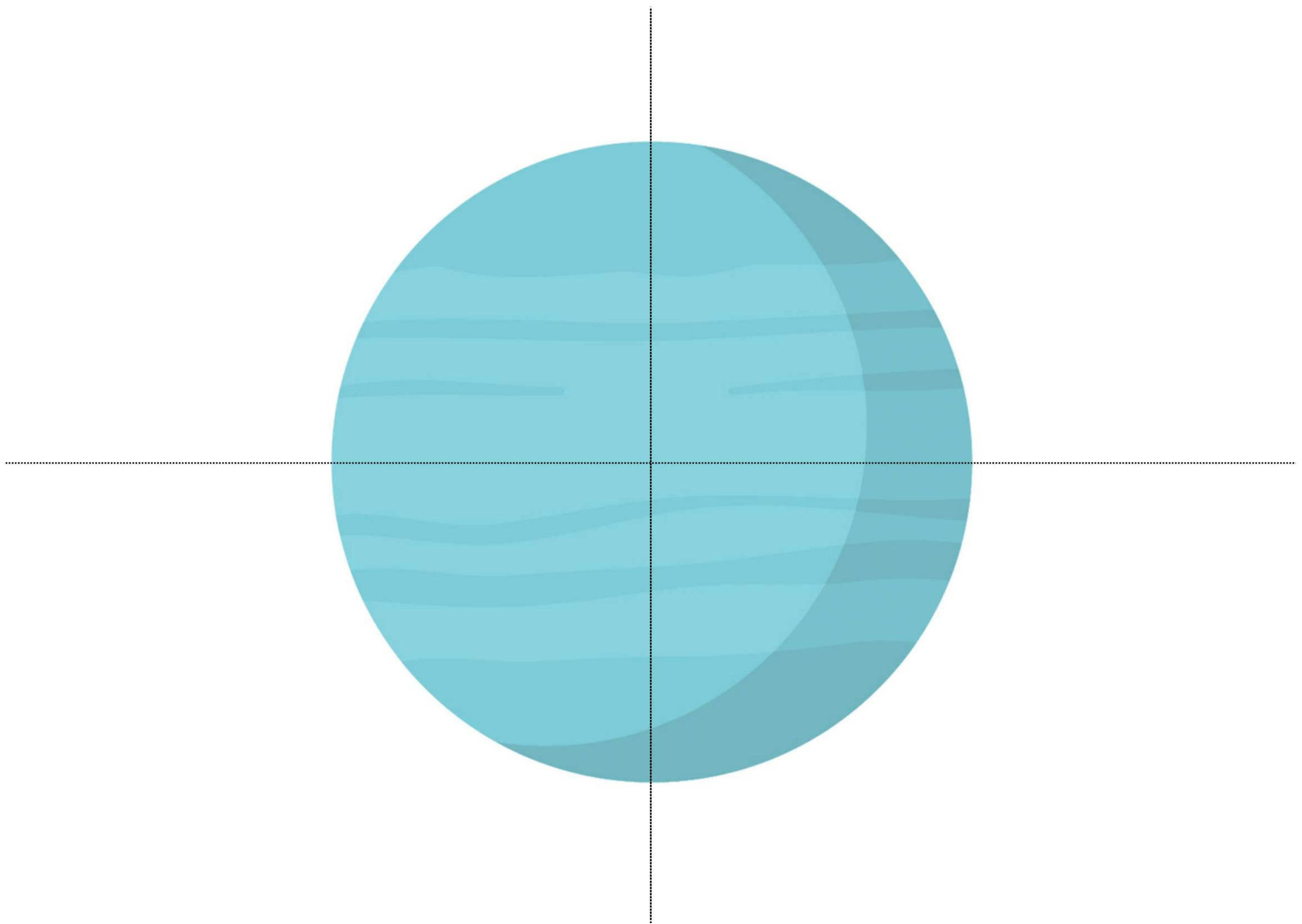


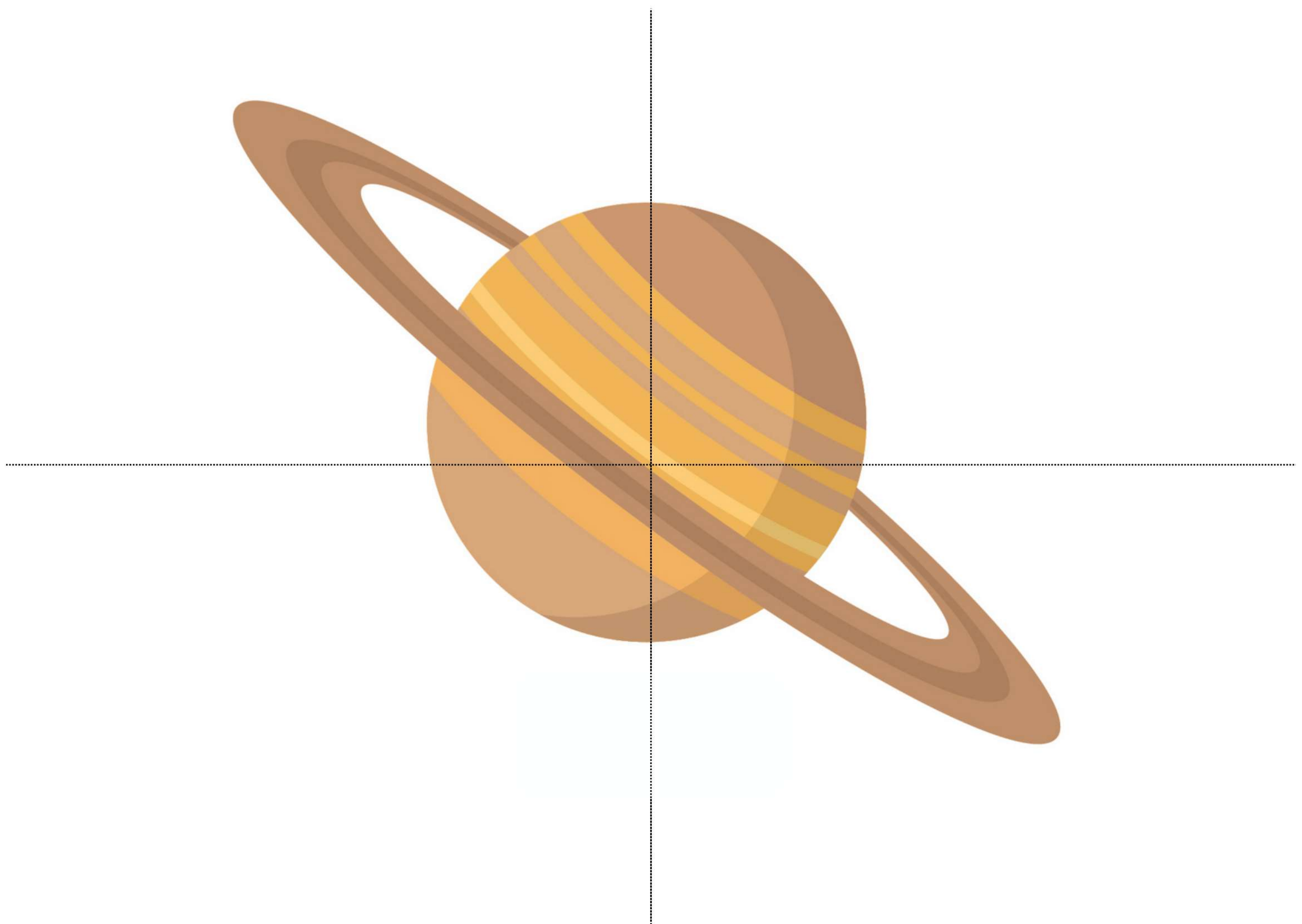


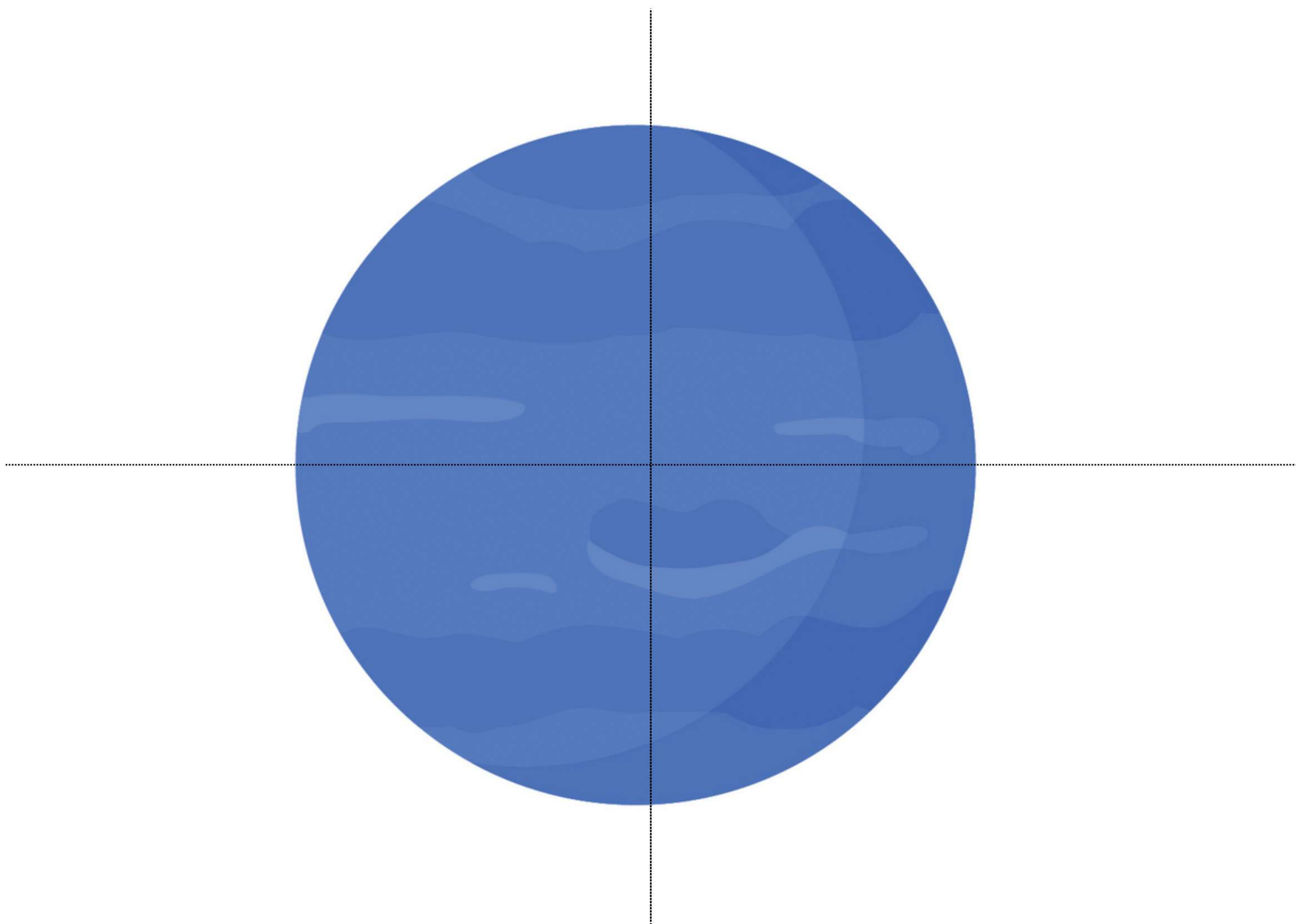


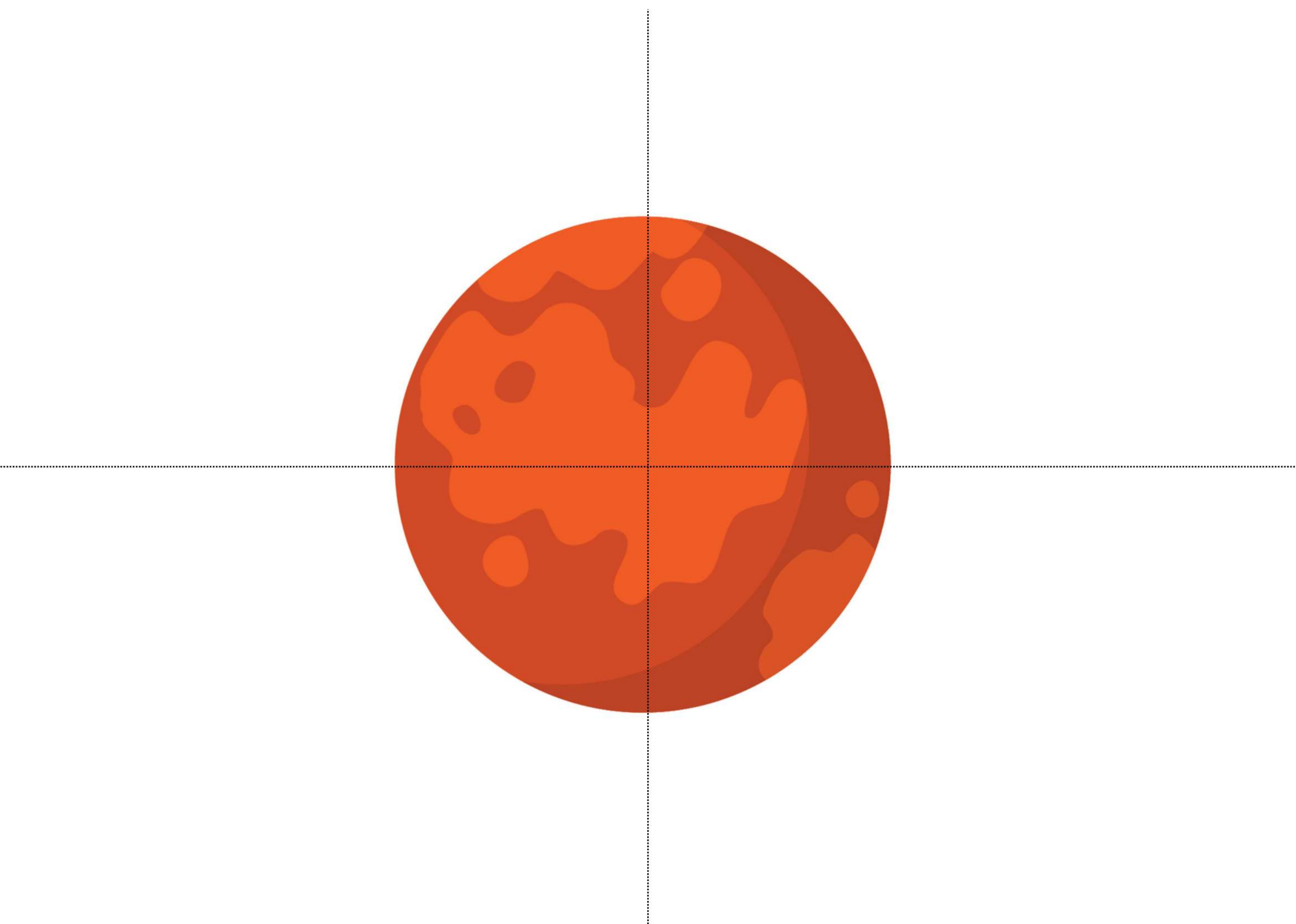


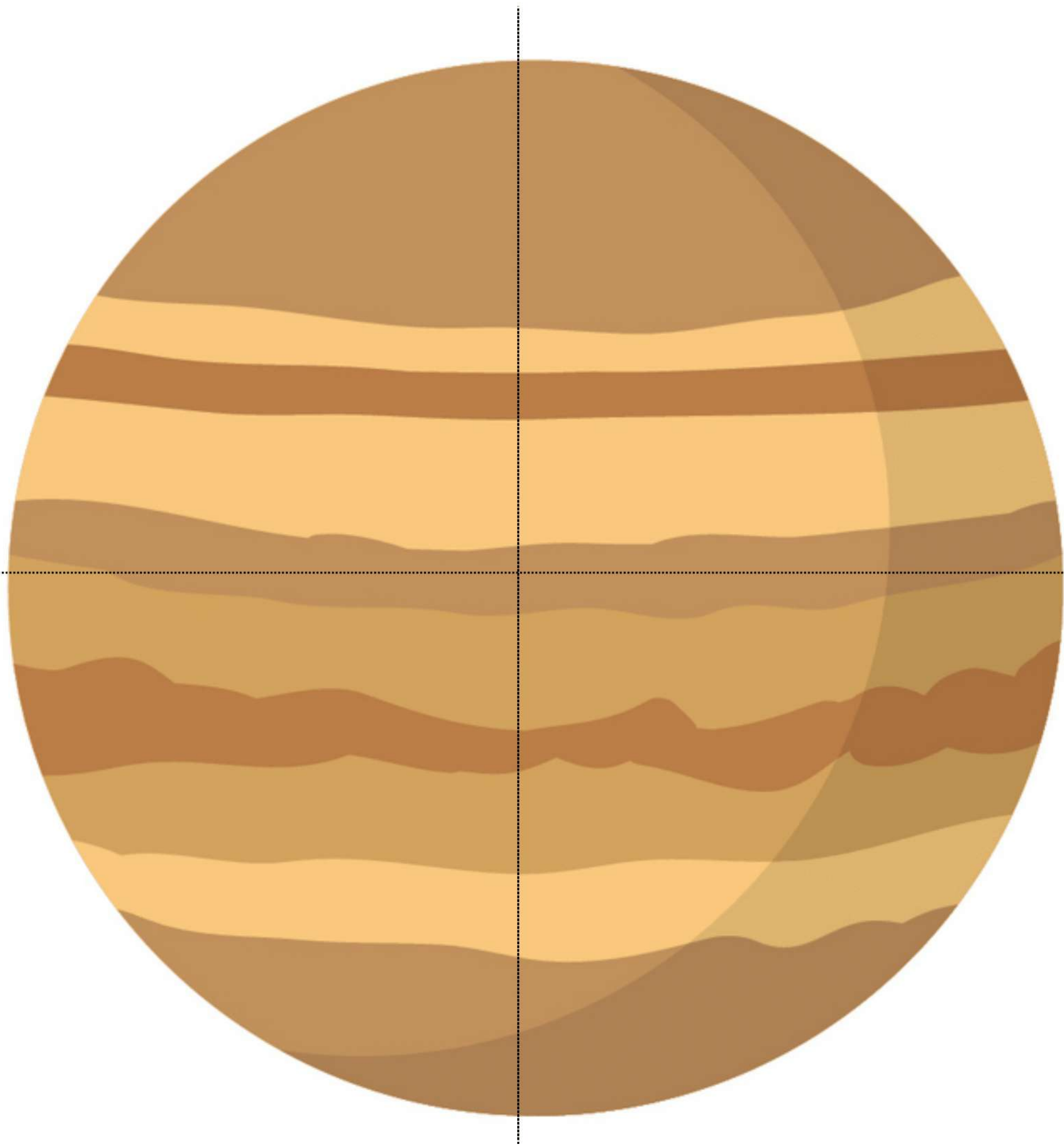












LICZBY PARZYSTE I NIEPARZYSTE

Co to znaczy, że liczba jest parzysta lub nieparzysta?
Jak to sprawdzić? Czy suma dwóch liczb parzystych będzie liczbą parzystą, czy nieparzystą? A suma liczby parzystej i nieparzystej jaką będzie liczbą? Odpowiedzi na powyższe pytania poszukają dzieci na zajęciach przygotowanych według zaproponowanego scenariusza. Pomoże im w tym robot edukacyjny... jeśli uprzednio zostanie przez dzieci prawidłowo zaprogramowany.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Poszerzenie wiadomości na temat liczb parzystych i nieparzystych;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie kompetencji miękkich, ze szczególnym uwzględnieniem umiejętności pracy zespołowej

Cele operacyjne:

Uczeń podczas zajęć:

- Wskazuje które liczby są liczbami parzystymi, a które są liczbami nieparzystymi;
- przelicza w adekwatnym do wieku zakresie;
- dodaje i odejmuje w zakresie adekwatnym do wieku;
- wyciąga wnioski na bazie praktycznego działania;
- rozwiązuje logiczne zagadki;
- programuje robota Photon wykorzystując interfejs Photon Badge i Photon Blocks;
- chętnie i zgodnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, kostka do gry, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

Ok. 45–90 minut

Wprowadzenie:

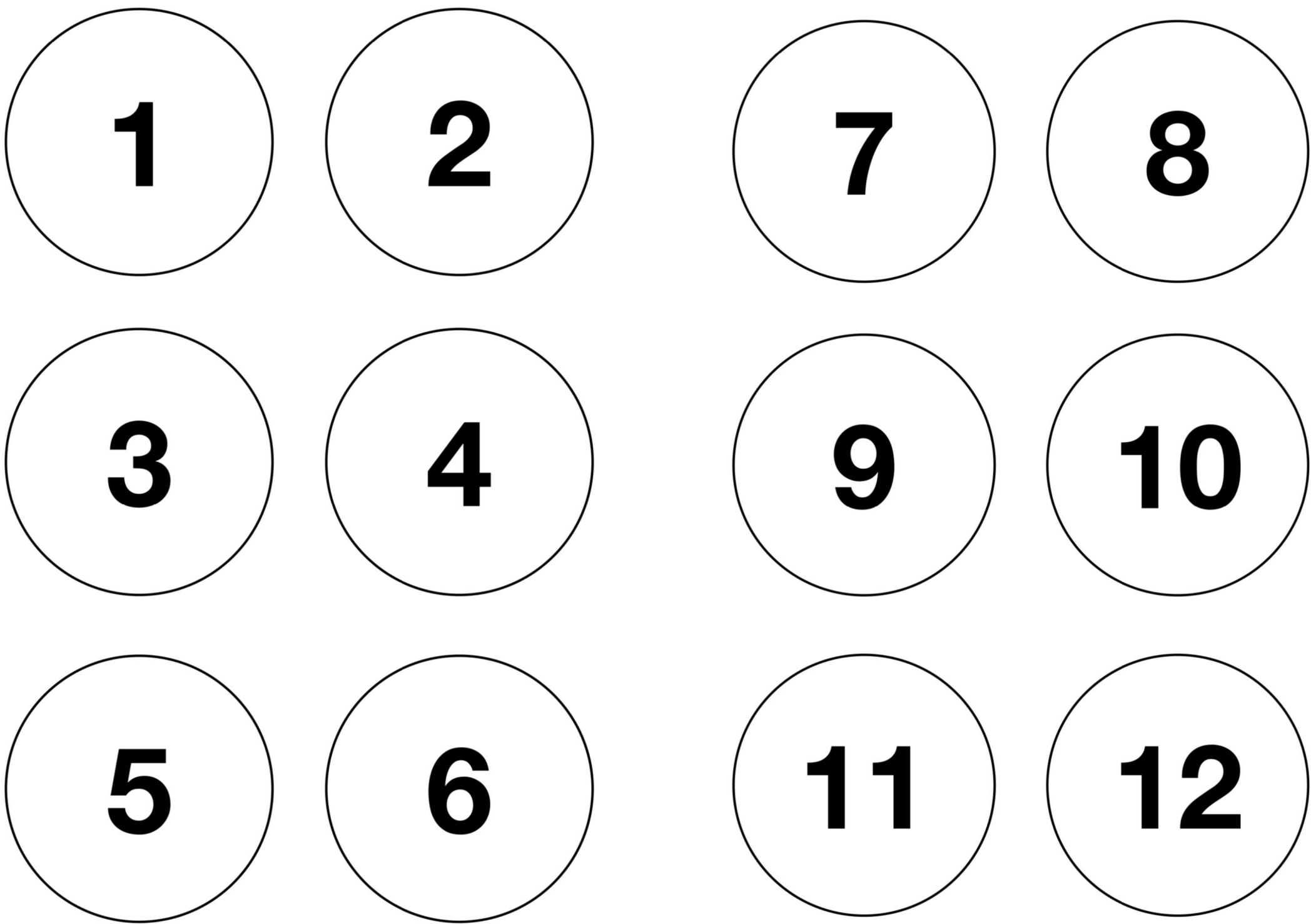
Nauczyciel wita się z dziećmi i zaprasza je do magicznego świata liczb. Pokazuje uczniom kubki w kilku kolorach (każdy kolor ułożony jest tak, że kubek wchodzi w kubek). Nauczyciel wybiera dwie osoby i pierwszy kolor kubków. Wybrane osoby będą dostawały na zmianę po jednym kubku, do momentu aż skończą się kubki. Uczniowie przeliczają kubki, które dostali, sprawdzają czy mają tyle samo kubków.

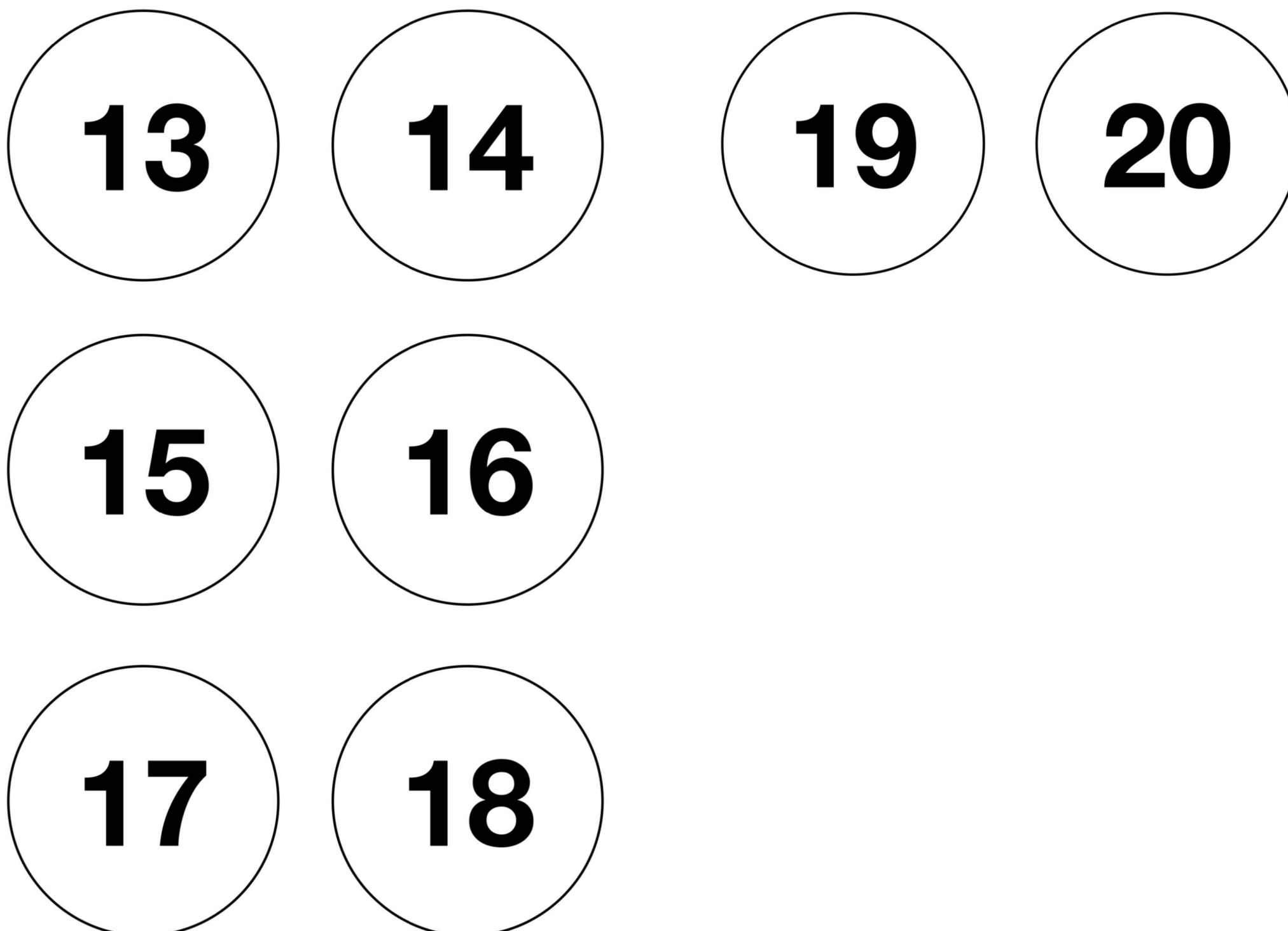
Analogicznie postępujemy z kubkami w pozostałych kolorach. Nauczyciel tłumaczy dzieciom, że jeśli otrzymali po tyle samo kubków, to znaczy, że liczba kubków (suma kubków jednej i drugiej osoby) była parzysta, a jeśli liczba kubków jednej osoby była inna niż drugiej, to znaczy, że liczba kubków (suma kubków jednej i drugiej osoby) była liczbą nieparzystą. Liczby parzyste, to takie liczby, które możemy podzielić na dwa.

Część właściwa:

Nauczyciel rozkłada przed uczniami krążki z liczbami. Prosi, żeby uczniowie ułożyli je w jednej linii w kolejności rosnącej. Dzieli dzieci na dwa zespoły. Prosi, żeby zespoły odczytywały liczby na zmianę: jedną jeden zespół, drugą drugi zespół.

Rysunek 1. Krążki z liczbami





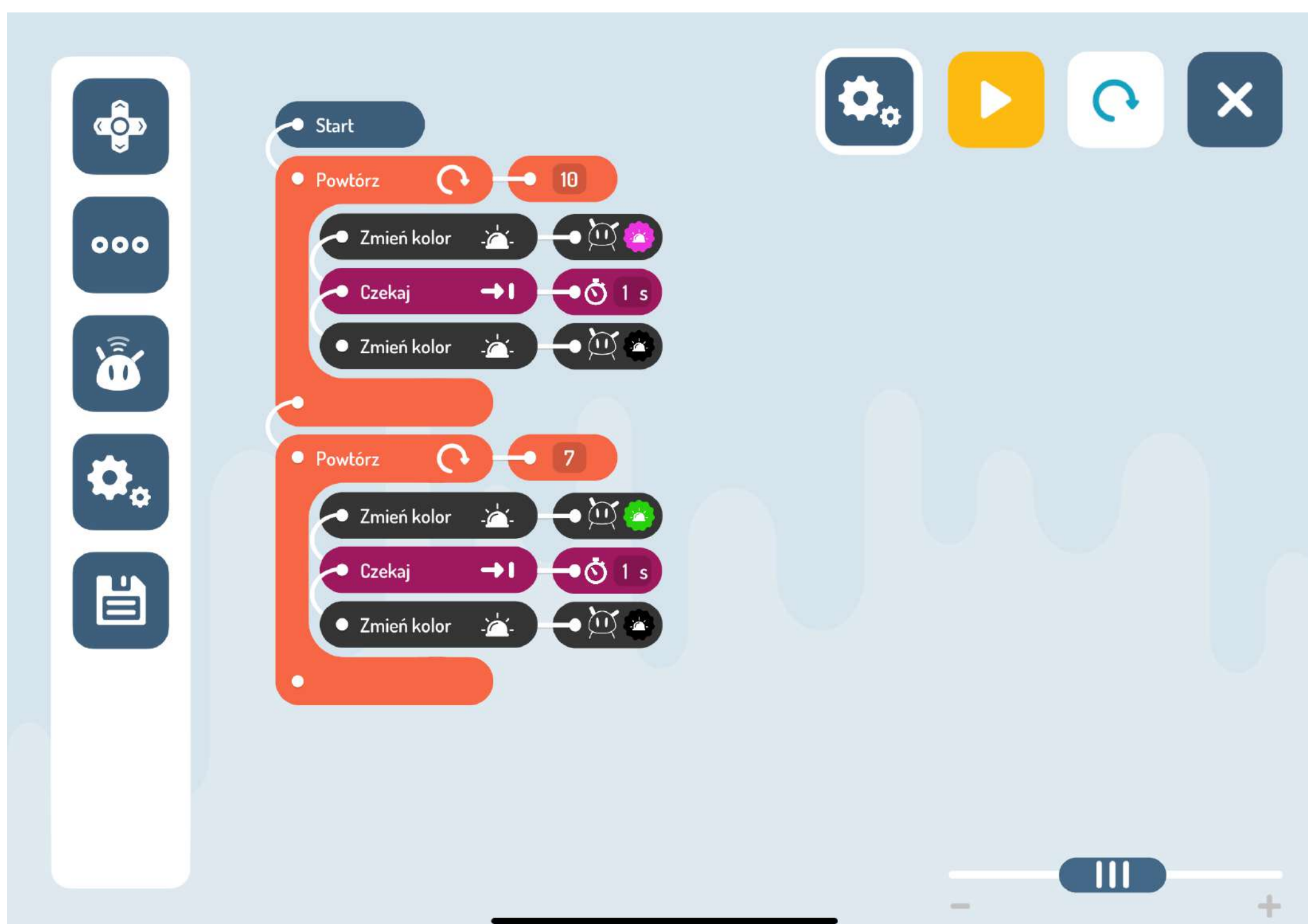
Źródło: opracowanie własne A. Świć.

Nauczyciel pyta czy uczestnicy potrafią nazwać grupę liczb, którą odczytywał jeden zespół i grupę liczb, którą odczytywał drugi zespół (jedna grupa, to liczby parzyste, druga, to liczby nieparzyste). Pyta, która z grup odczytywała liczby parzyste, a która nieparzyste.

Dzieci rozkładają na losowych polach na macie do kodowania krążki z liczbami (te same, z których korzystali przy poprzednim zadaniu). Wybierają pole, z którego wystartuje robot.

Nauczyciel tłumaczy na czym będzie polegało zadanie: uczniowie wybiorą sześć liczb, po których przejedzie robot. Trzy z tych liczb muszą być liczbami parzystymi, a trzy nieparzystymi. Kiedy robot najedzie na daną liczbę zaświeci czułkami i oczami tyle razy, ile wskazuje liczba, na którą najechał. Jeśli ta liczba będzie liczbą parzystą, to robot będzie świecił na różowo, jeśli ta liczba będzie liczbą nieparzystą, to robot będzie świecił na zielono. Nauczyciel zwraca uwagę uczniom na to, że jeśli ustawią barwę np.: na różową i wybiorą liczbę powtórzeń, to nie osiągną zamierzonego efektu, bo robot będzie świecił nieprzerwanym światłem, tylko dłużej. Pyta uczniów, czy wiedzą jak zaprogramować robota, żeby uzyskać efekt świecenia. Jeśli uczniowie nie wiedzą, nauczyciel wyjaśnia, że żeby osiągnąć efekt, który pozwoli policzyć ile razy zaświecił robot trzeba pomiędzy barwami (różową lub zieloną) wstawić barwę czarną (na czarnej nie będzie efektu świecenia).

Rysunek 2. Interfejs Photon Blocks



Źródło: opracowanie własne A. Świć.

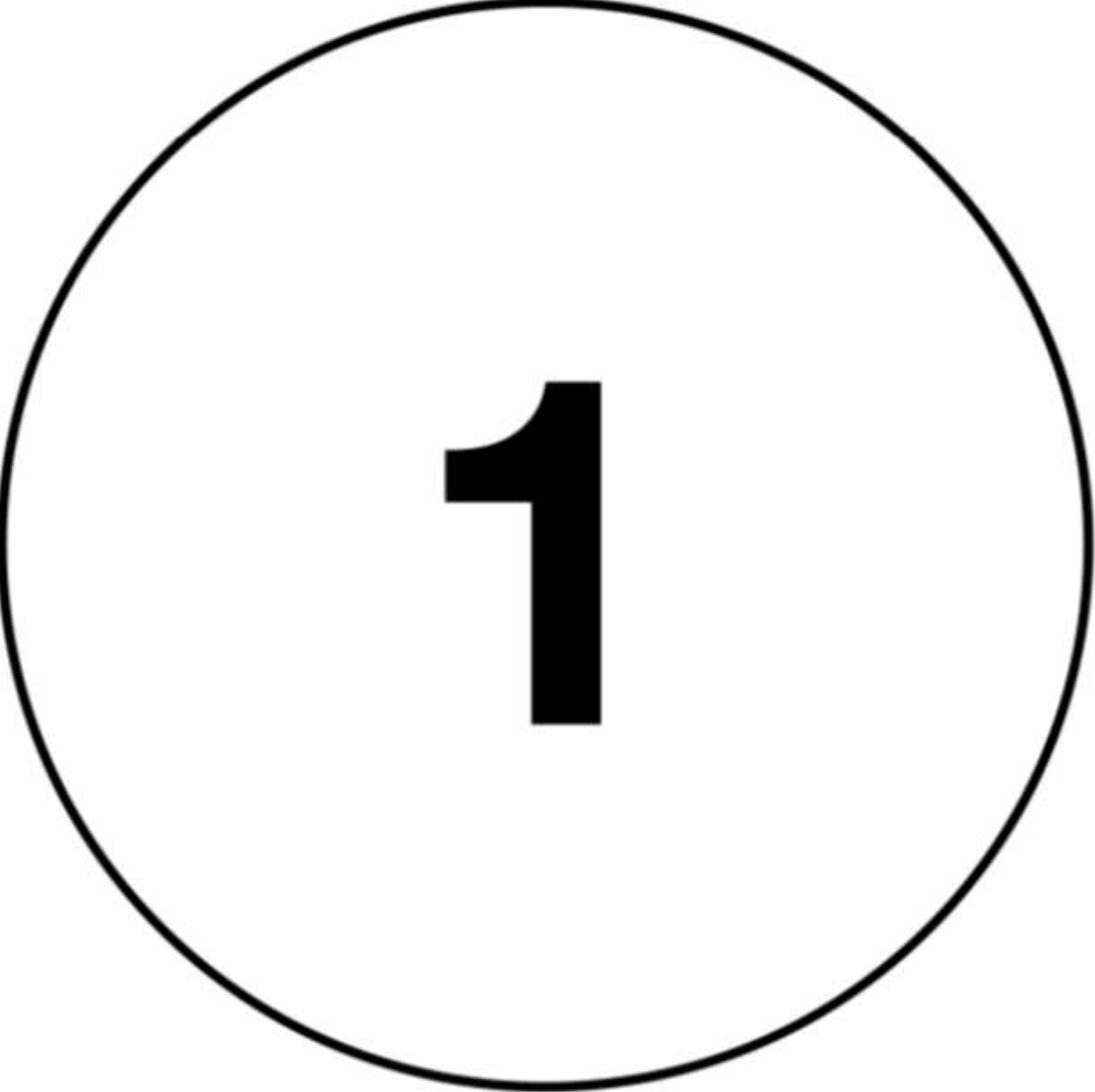
Po ułożeniu programu uczniowie wykonują próbę. Sprawdzają, czy robot faktycznie przejedzie przez wybrane pola i czy zaświeci odpowiednimi barwami tyle razy, ile wskazują liczby znajdujące się na trasie jego przejazdu.

W kolejnym ćwiczeniu na macie do kodowania pojawią się zapisane działania (odejmowanie i dodawanie w zakresie odpowiednim do możliwości dzieci). Uczniowie dokonują obliczeń, a następnie programują robota w taki sposób, żeby przejechał wyłącznie po liczbach parzystych (wynikach). Po wykonaniu zadania jeszcze raz przyglądają się działaniom pod kątem sprawdzenia, czy występuje jakaś prawidłowość np.: suma dwóch liczb parzystych będzie liczbą parzystą, suma liczb parzystej i nieparzystej będzie liczbą nieparzystą.

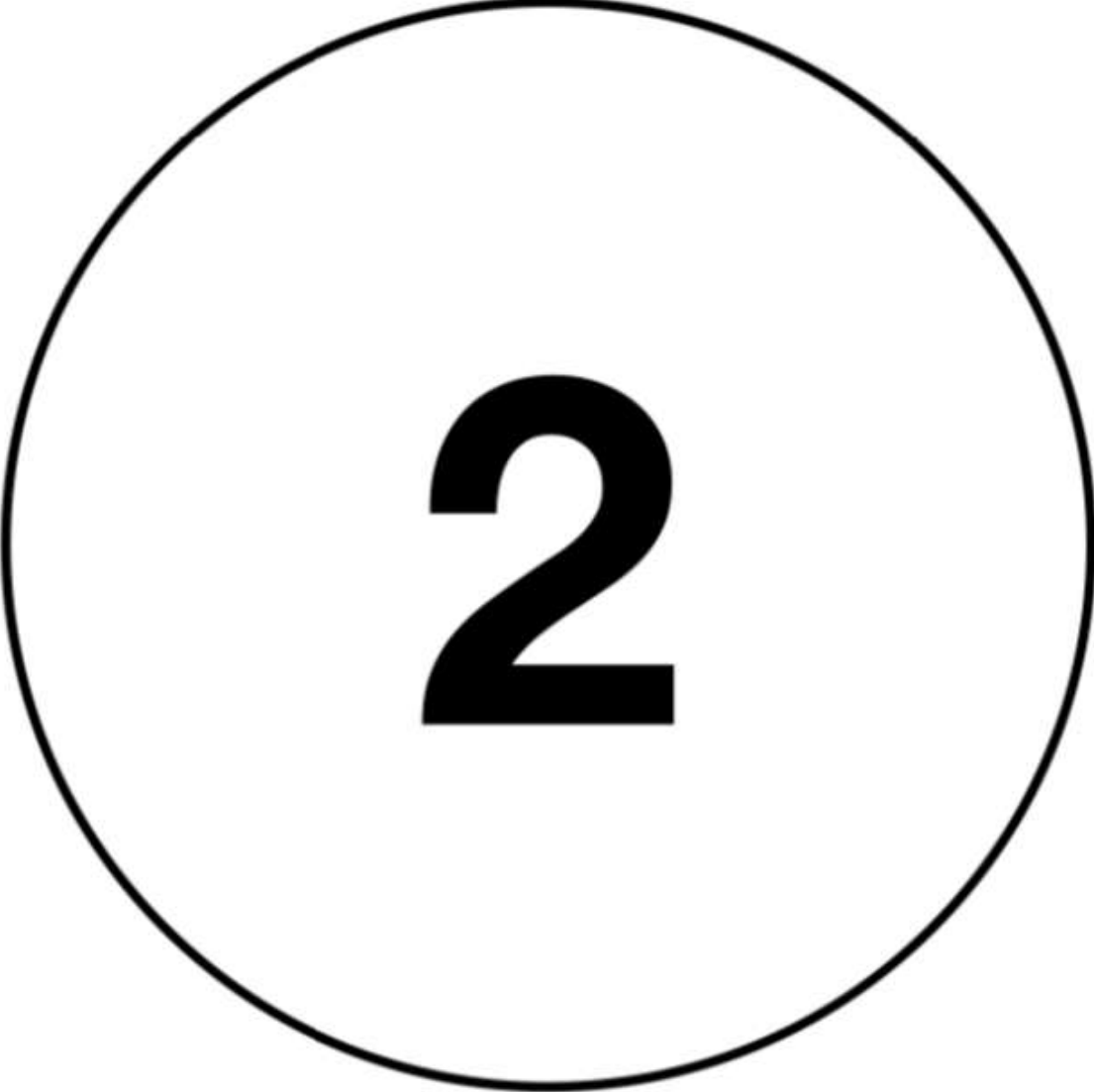
Zakończenie zajęć:

Na zakończenie zajęć nauczyciel zaprasza dzieci do gry losowo zręcznościowej. Każdy zespół rzuca dziesięć razy kostką. Za każdym razem, kiedy wypadnie liczba parzysta zespół otrzyma tyle kubków, ile wskazał rzut kostką. Jeśli wypadnie liczba nieparzysta, zespół nie otrzyma kubków. W ten sposób zespoły zbiorą materiał do budowania wieży z kubków. Po dziesięciu rzutach każdy zespół będzie budował wieżę. Wygra zespół, którego wieża będzie najwyższa i nie przewróci się przez pełną minutę.

Nauczyciel pyta dzieci, który fragment zajęć podobał im się najbardziej, a który chcieliby zmienić. Dziękuje uczniom za zaangażowanie na zajęciach.



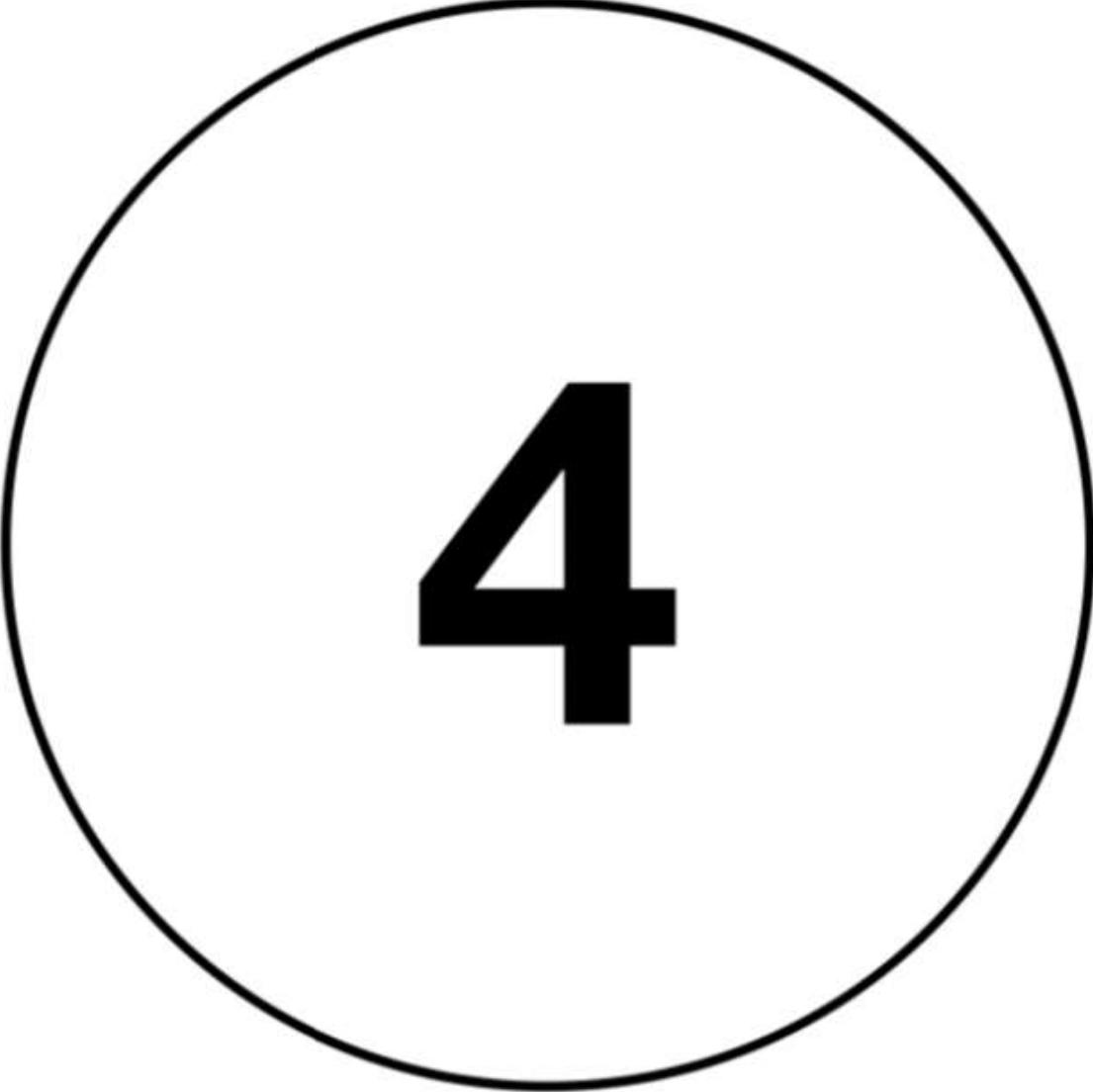
1



2



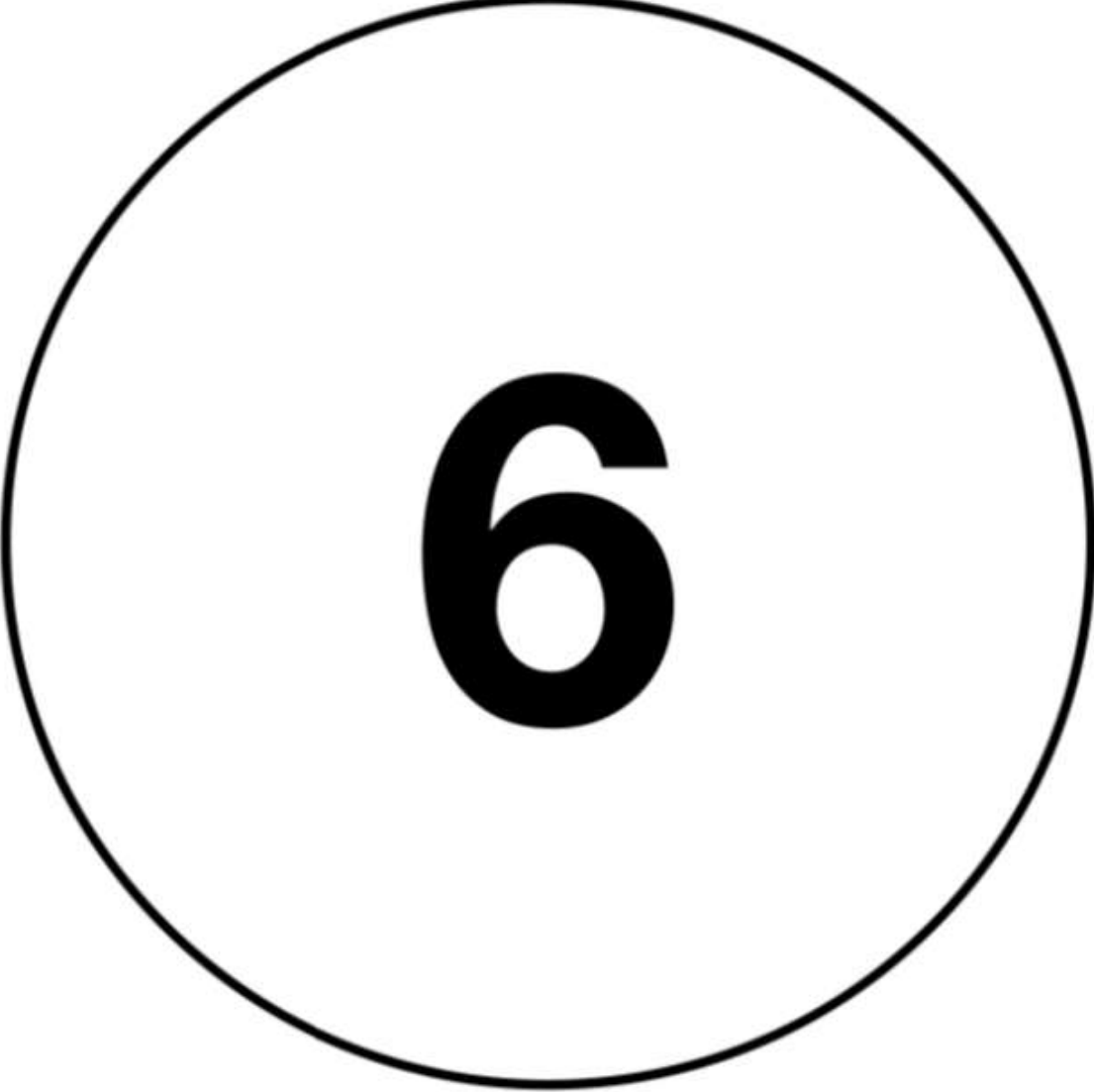
3



4



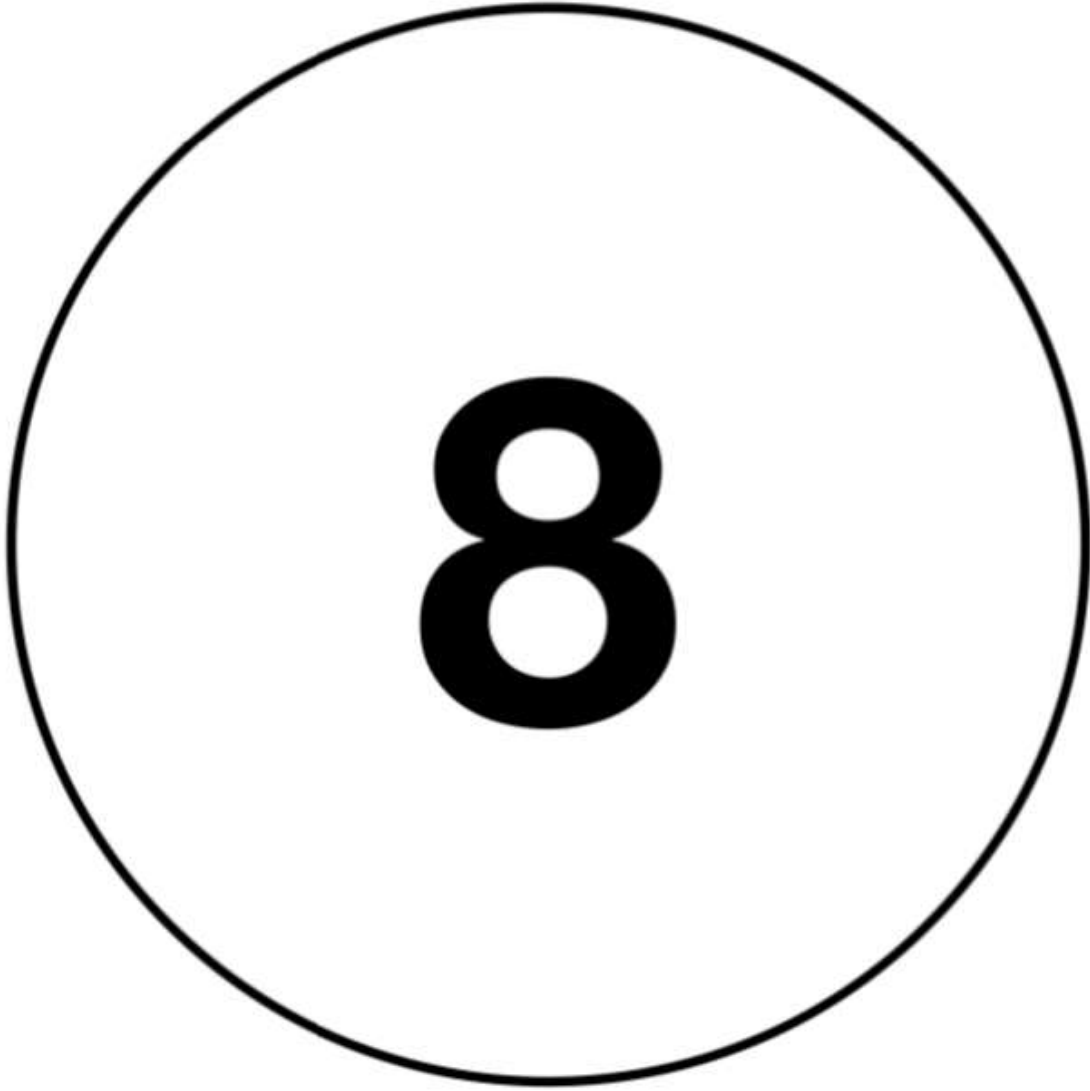
5



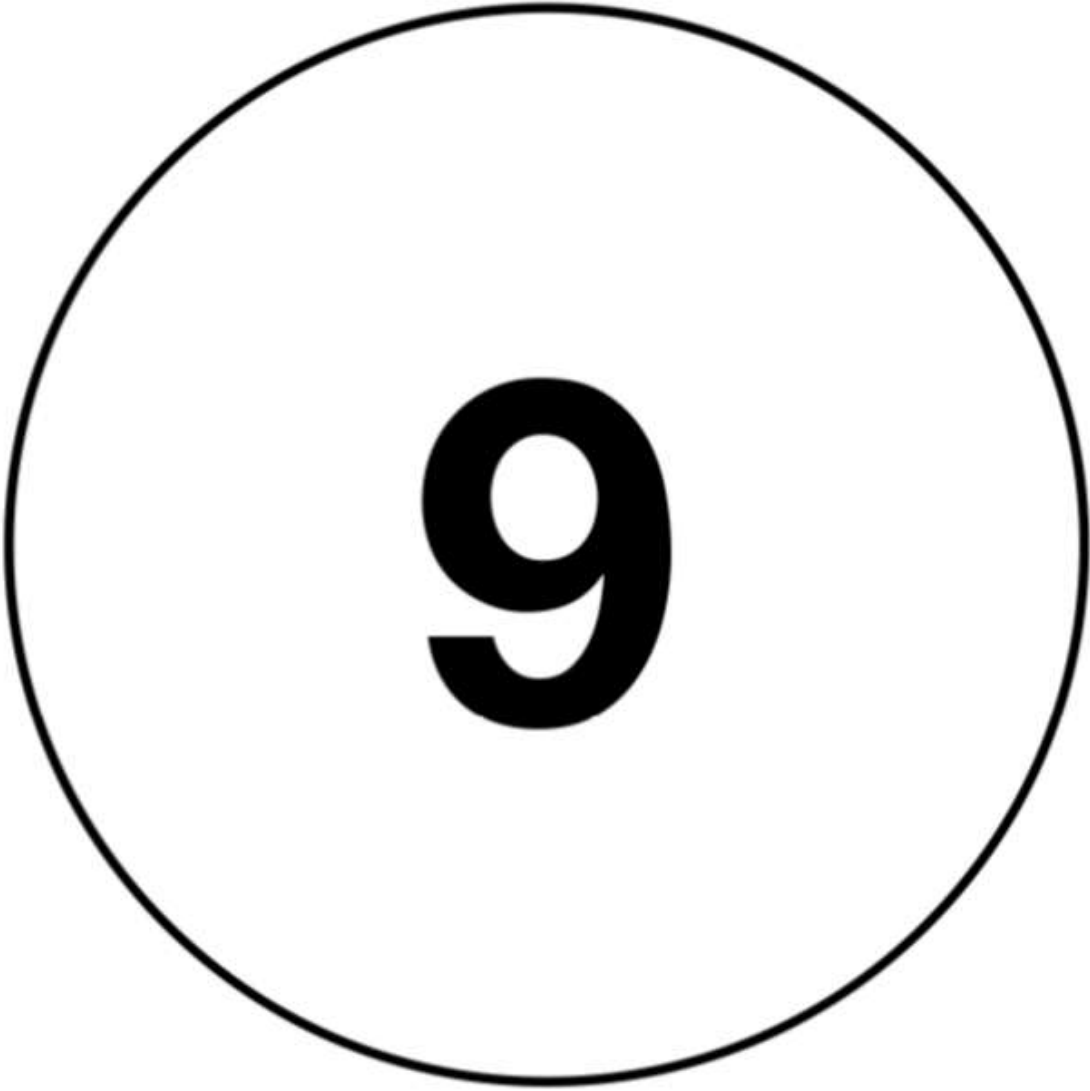
6



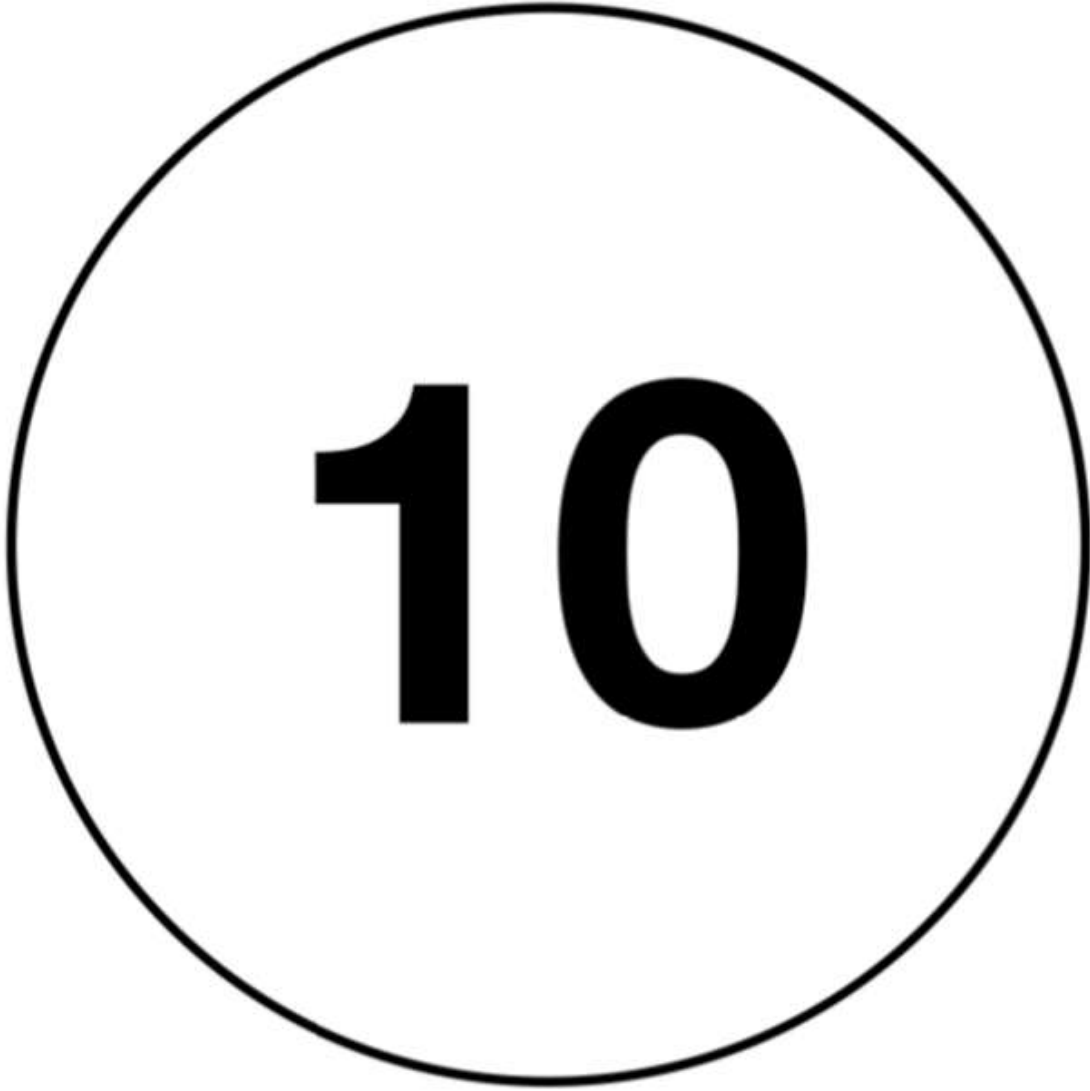
7



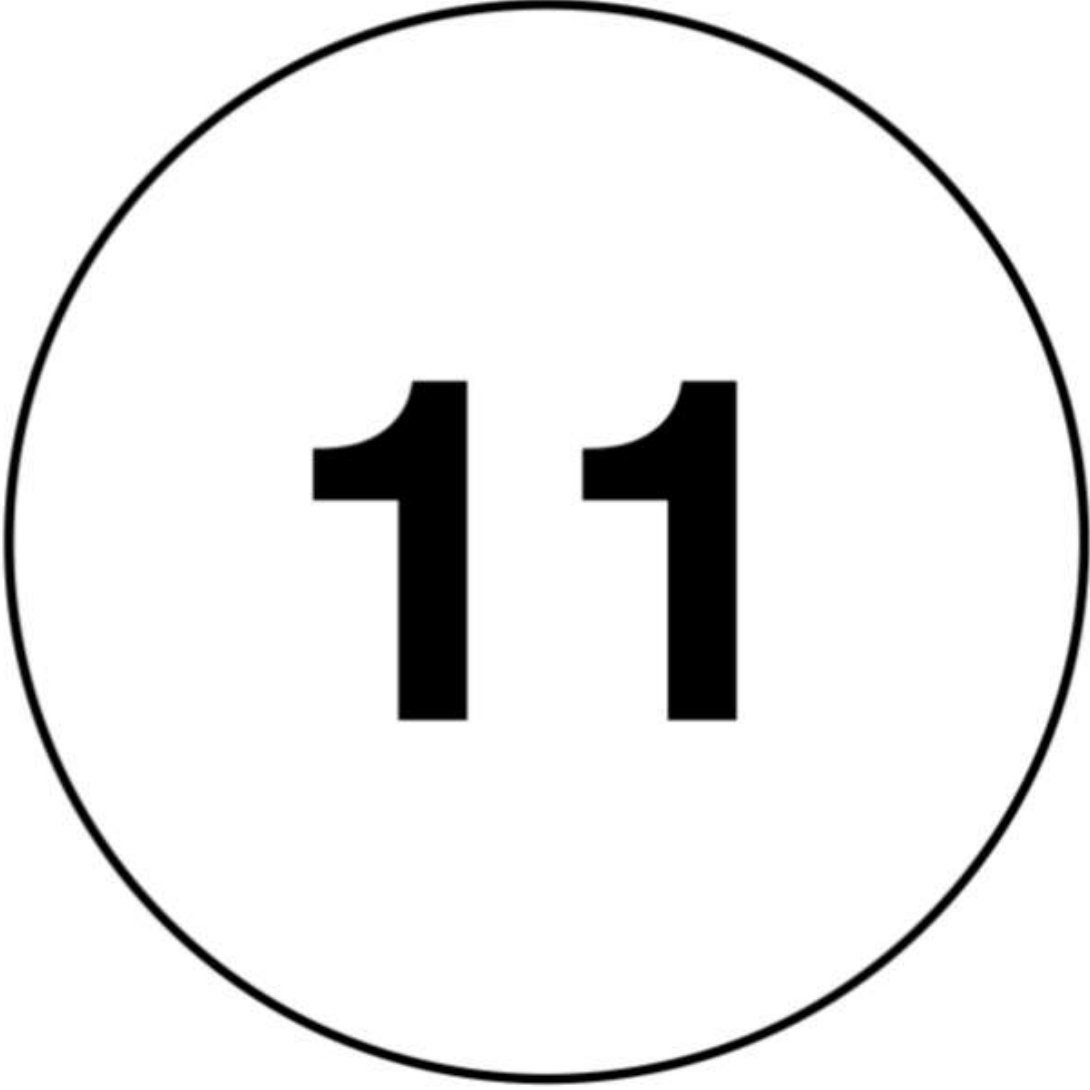
8



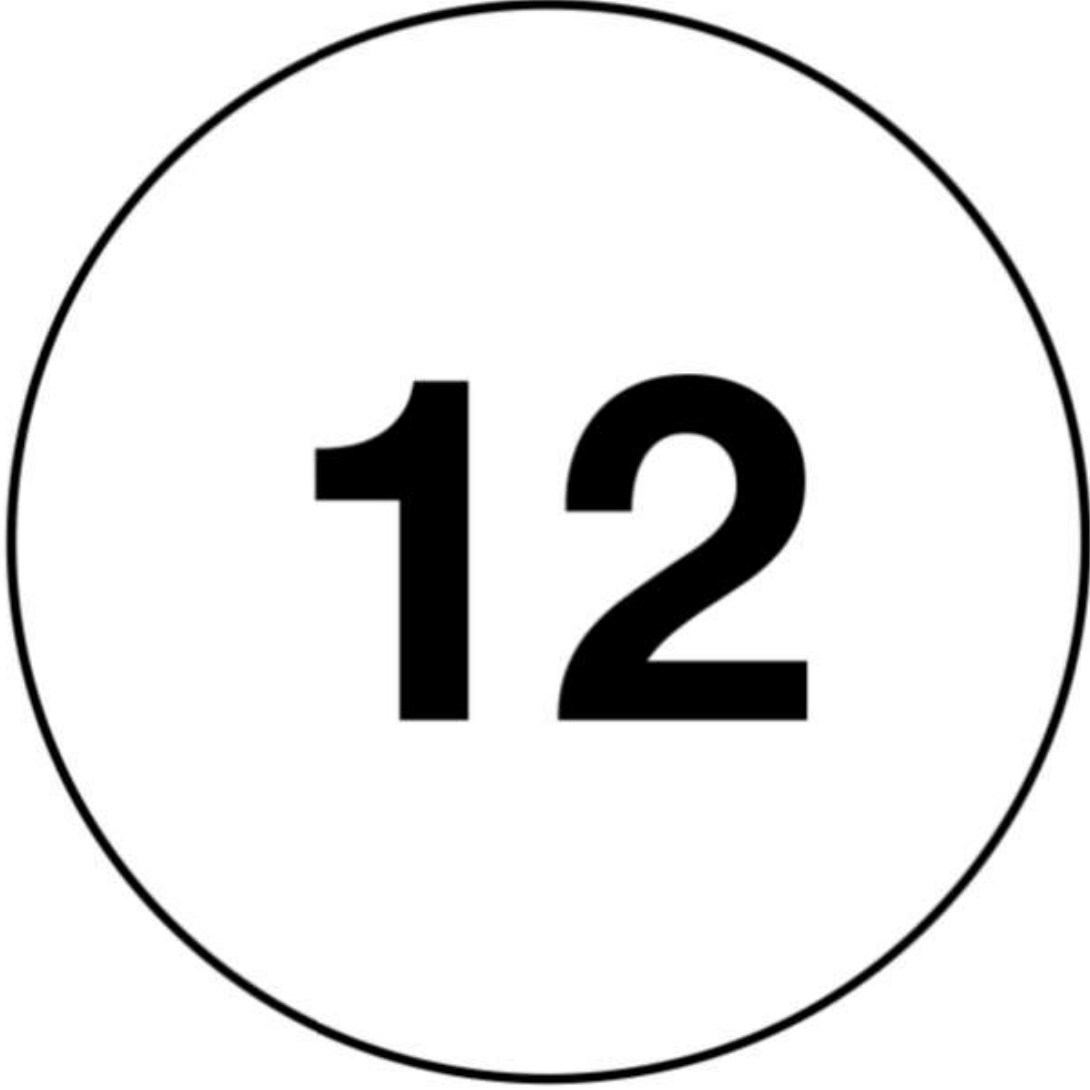
9



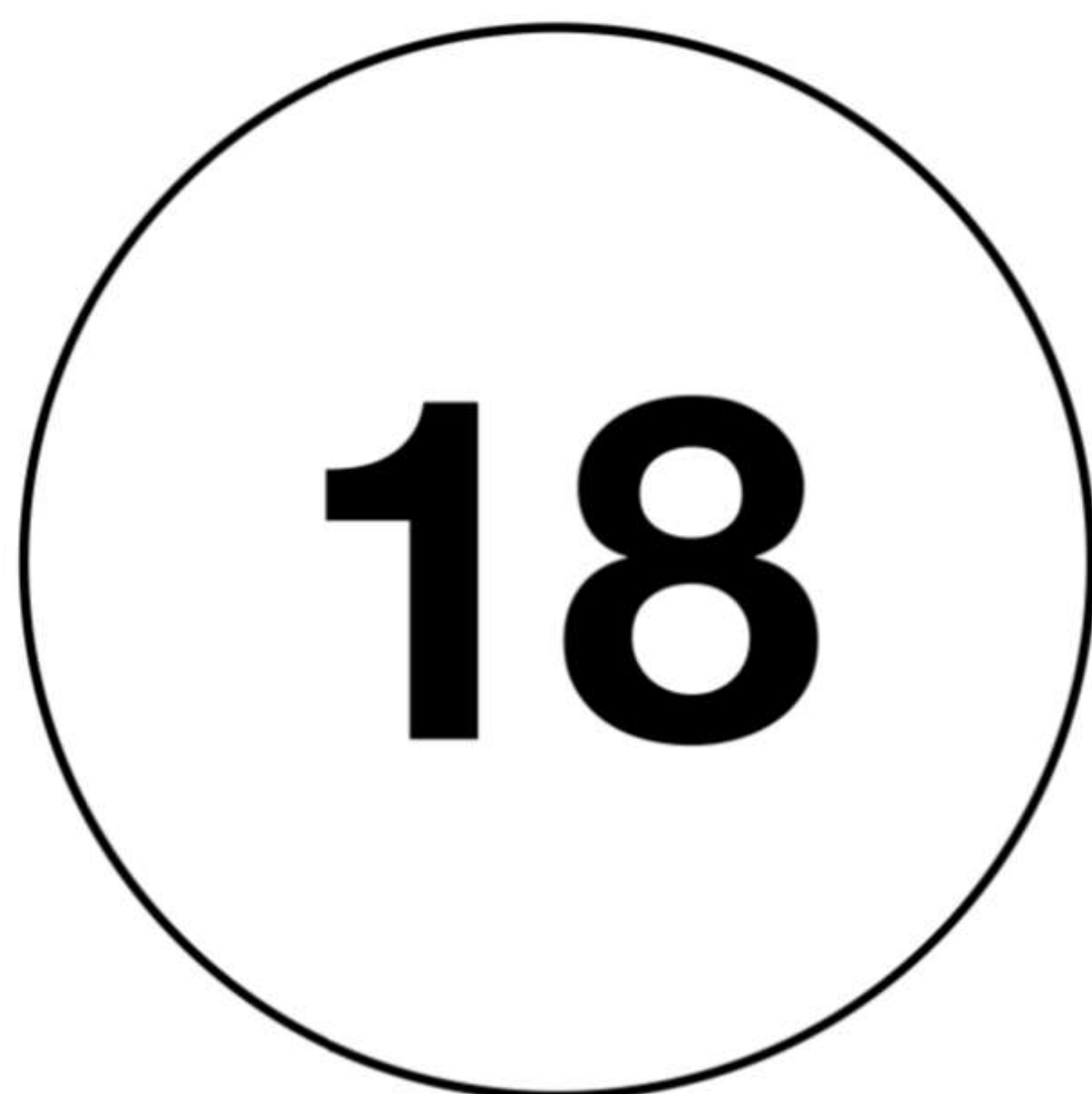
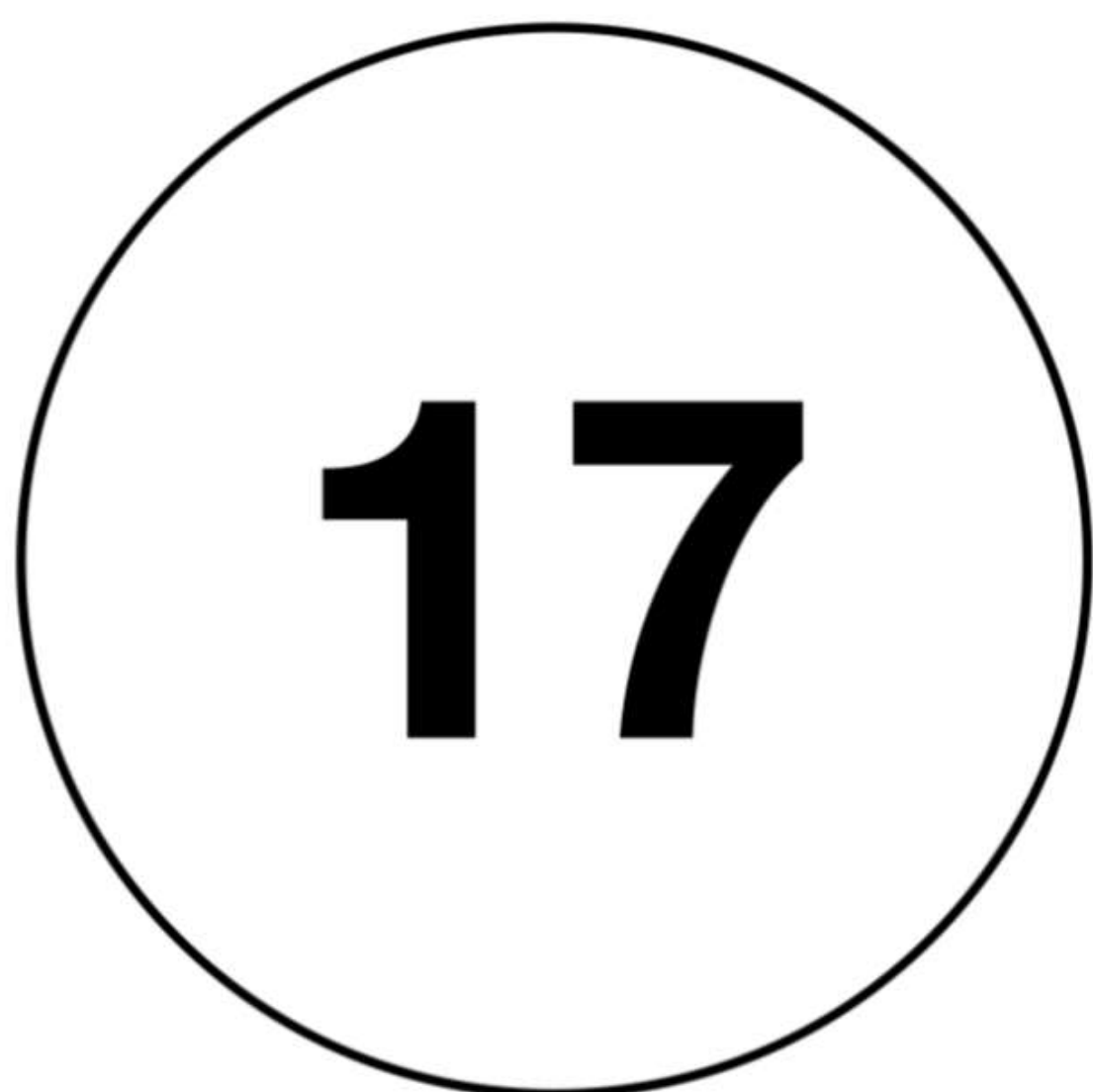
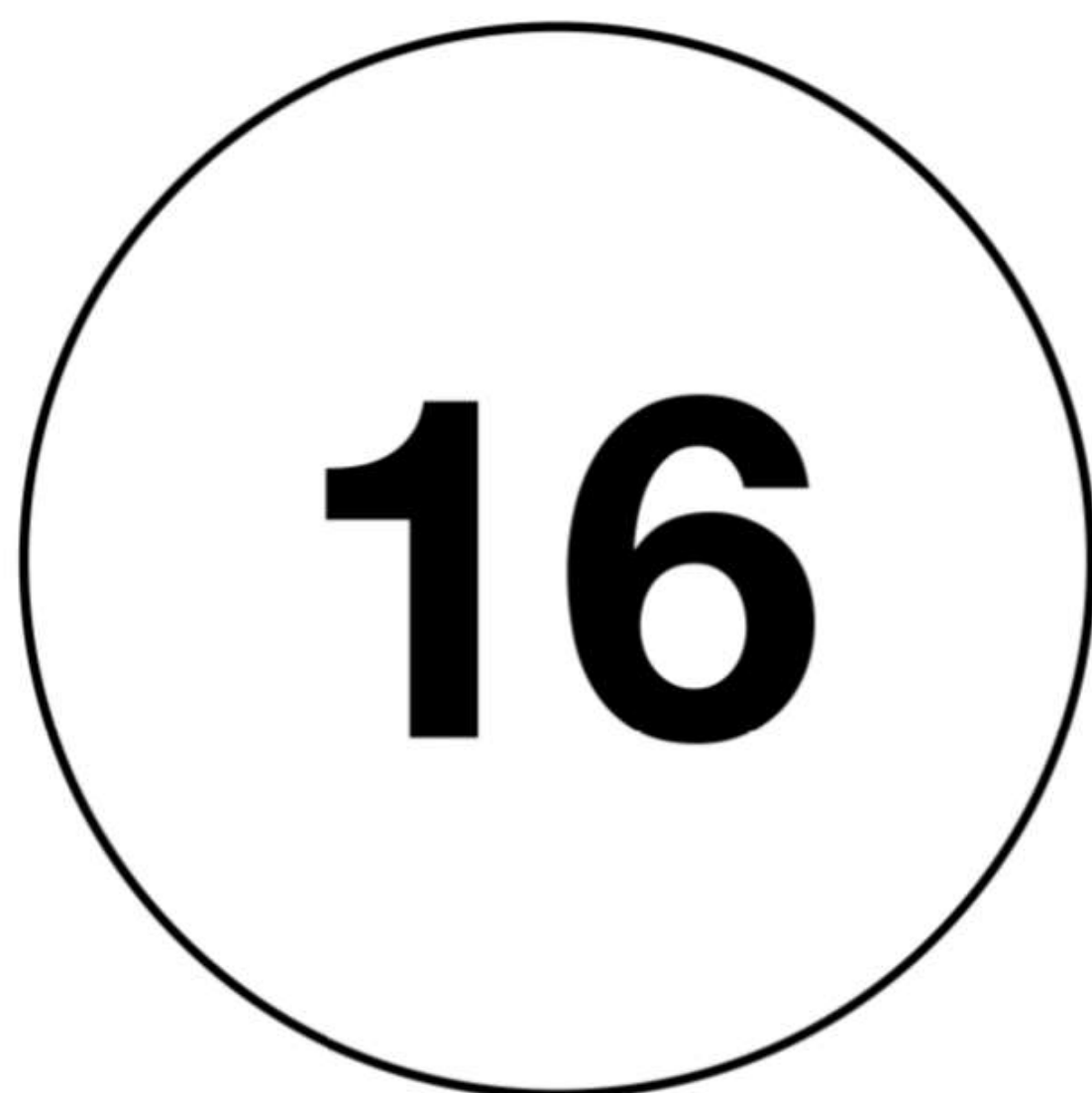
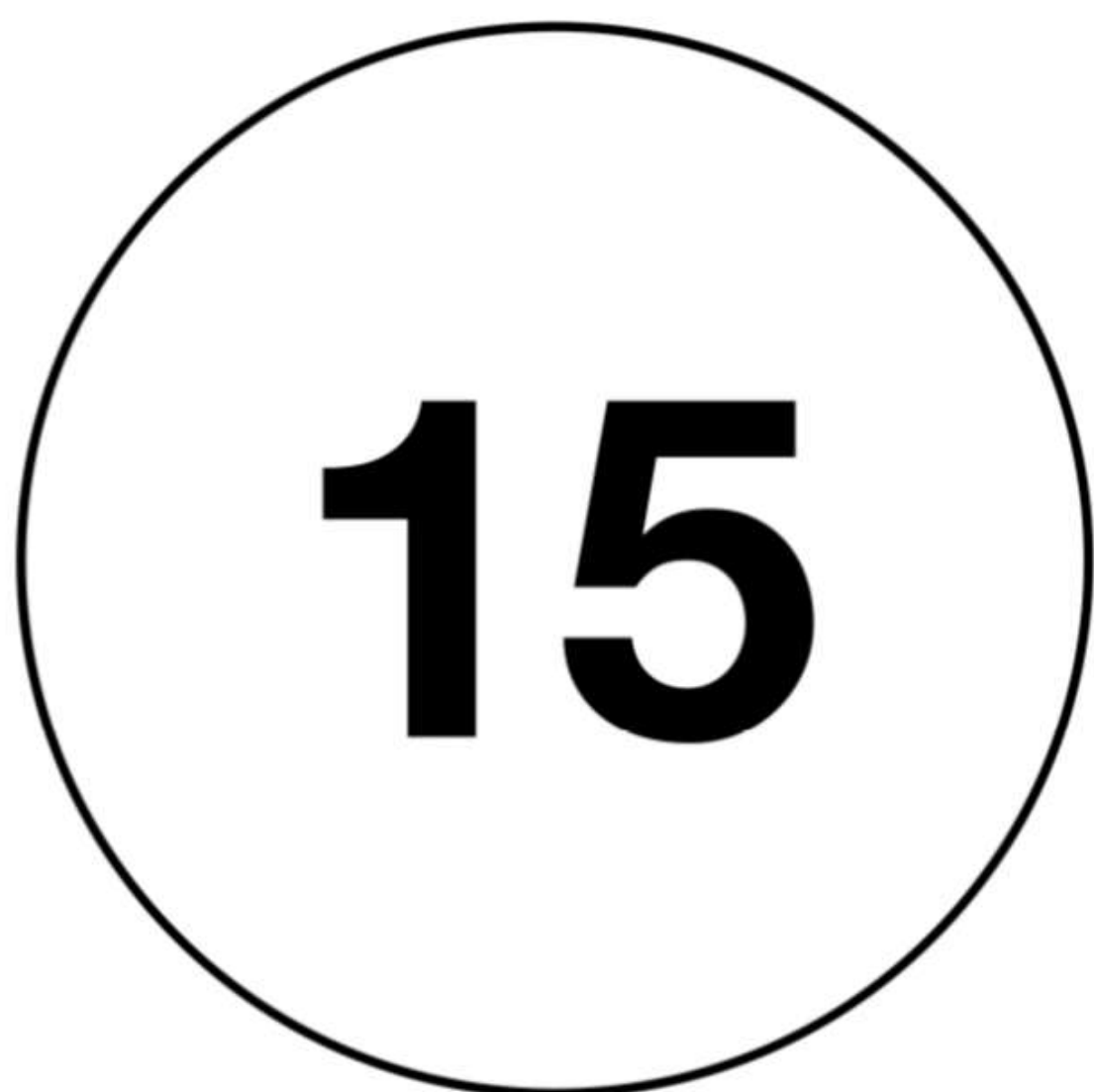
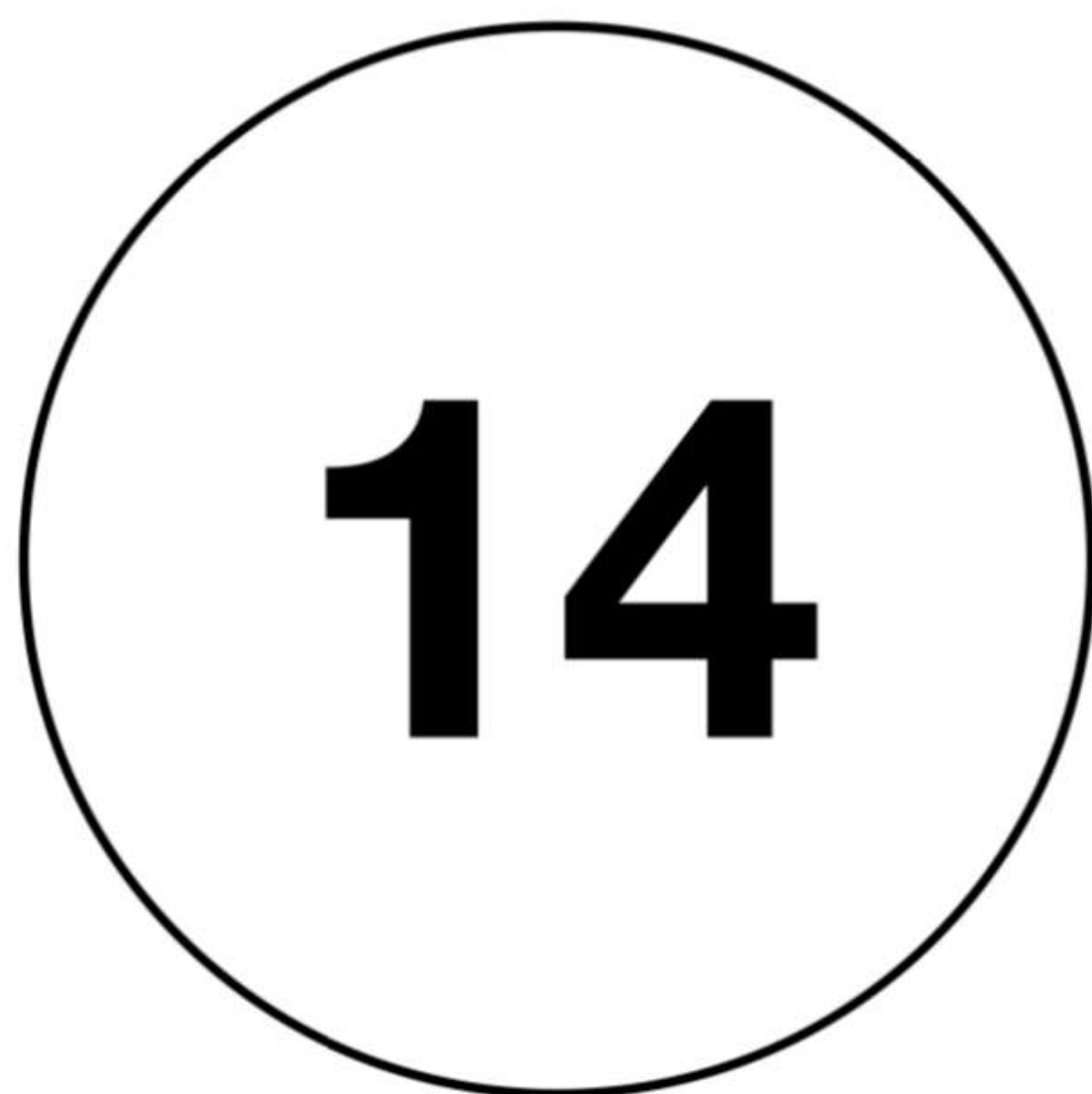
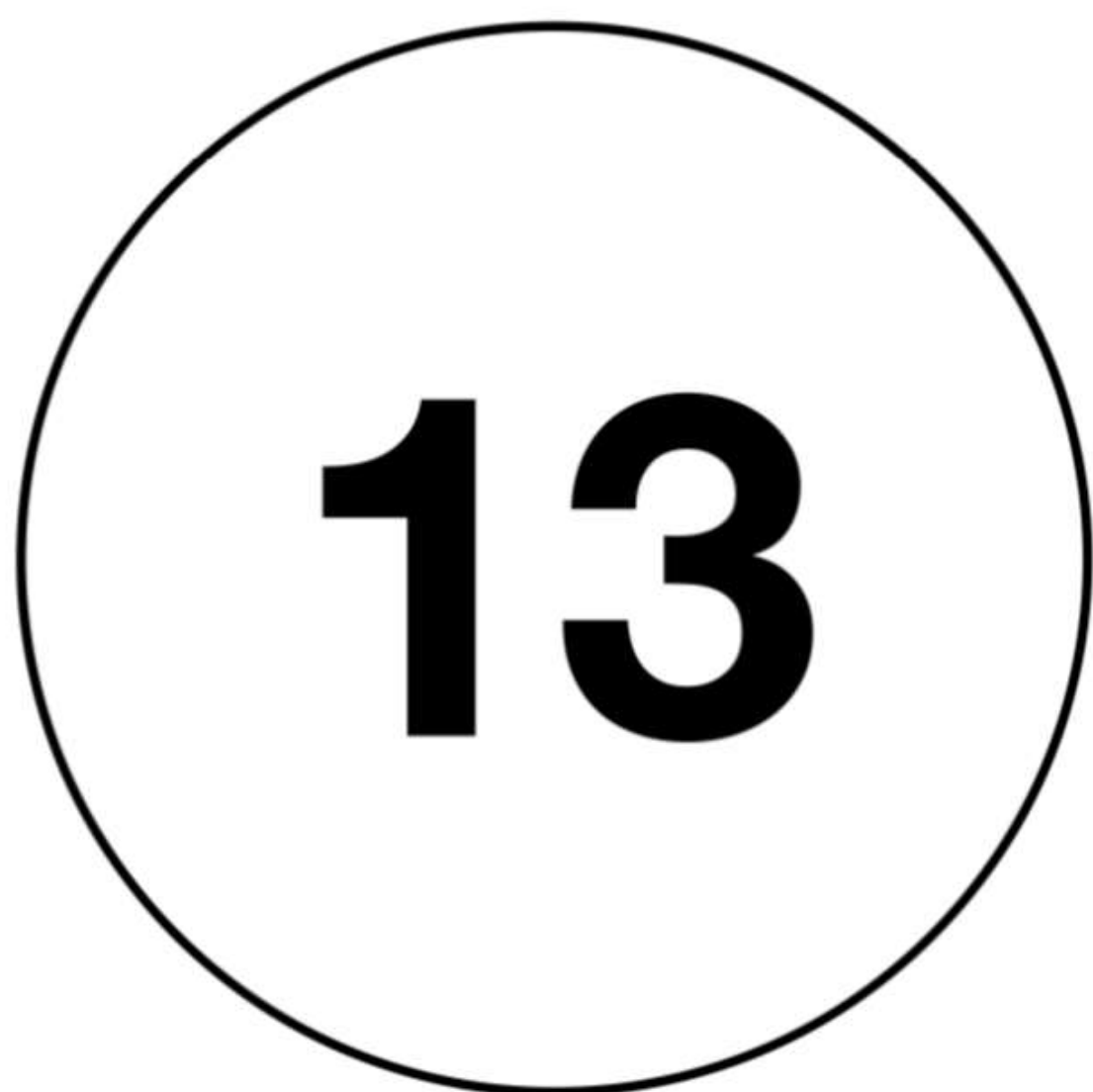
10

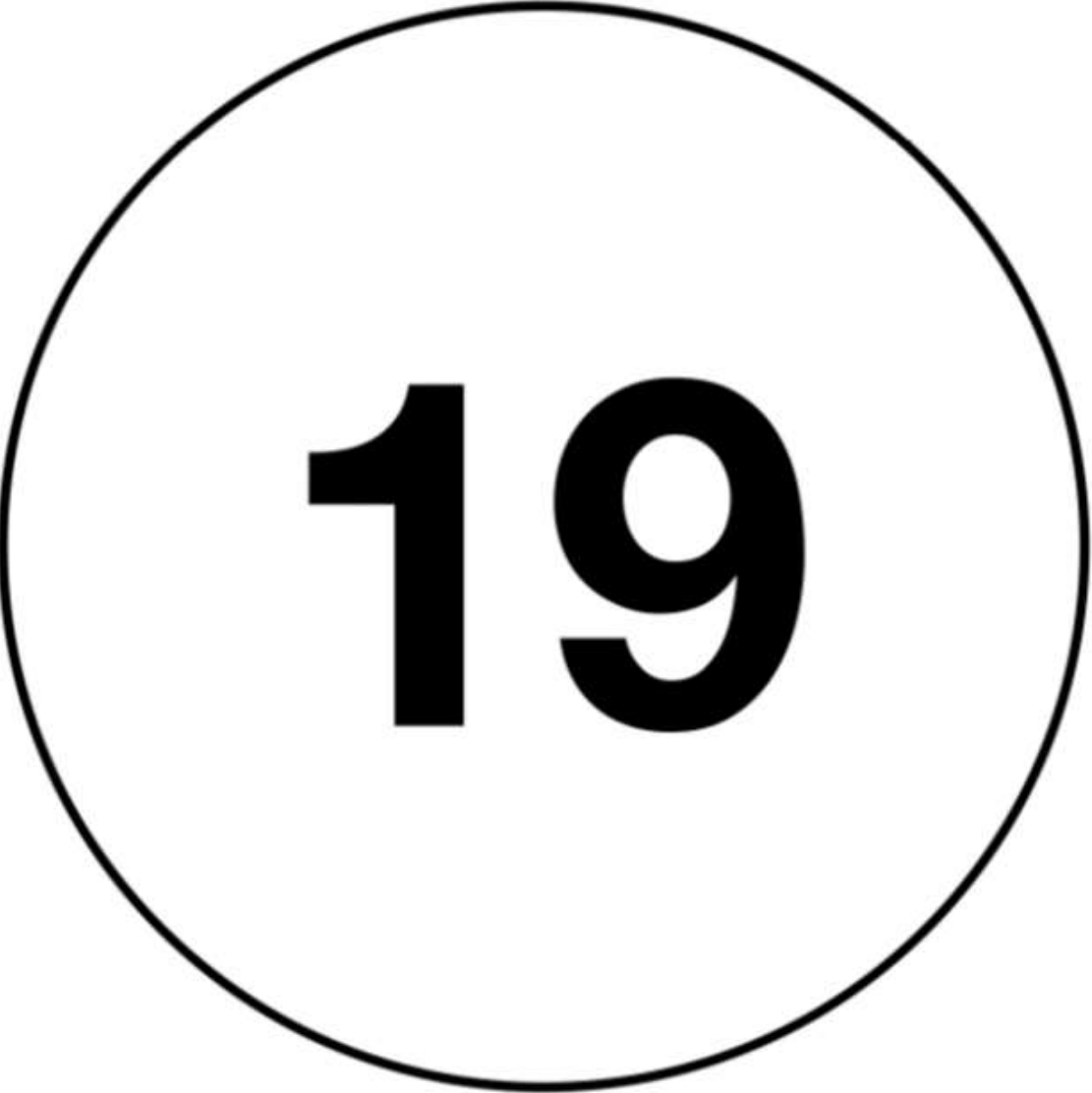


11



12





19



20

**POZNAJEMY LEKTURY
Z PROGRAMOWANIEM
W TLE**

Czy możliwa jest nauka programowania połączona z omawianiem wybranej lektury? Czy na takich zajęciach można zaprogramować roboty? Oczywiście, że tak! Na zajęciach przeprowadzonych według zaproponowanego scenariusza uczniowie nie tylko utrwalał znajomość lektury Waldemara Cichonia „Cukierku, Ty Łobuzie”, ale też poćwiczą logiczne, algorytmiczne myślenie i umiejętność pracy zespołowej.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe.

Cele ogólne:

- Poszerzanie wiedzy na temat lektury Waldemara Cichonia „Cukierku! Ty Łobuzie”;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie miękkich kompetencji, ze szczególnym uwzględnieniem umiejętności pracy zespołowej.

Cele operacyjne:

Uczeń podczas zajęć:

- przedstawia zdarzenia z lektury „Cukierku, ty Łobuzie” zachowując chronologię;
- odpowiada na pytania dotyczące lektury Waldemara Cichonia „Cukierku! Ty Łobuzie”;
- układa obrazki zgodnie z podanymi warunkami;
- programuje robota Photon wykorzystując interfejs Photon Badge i Photon Blocks;
- chętnie i zgodnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

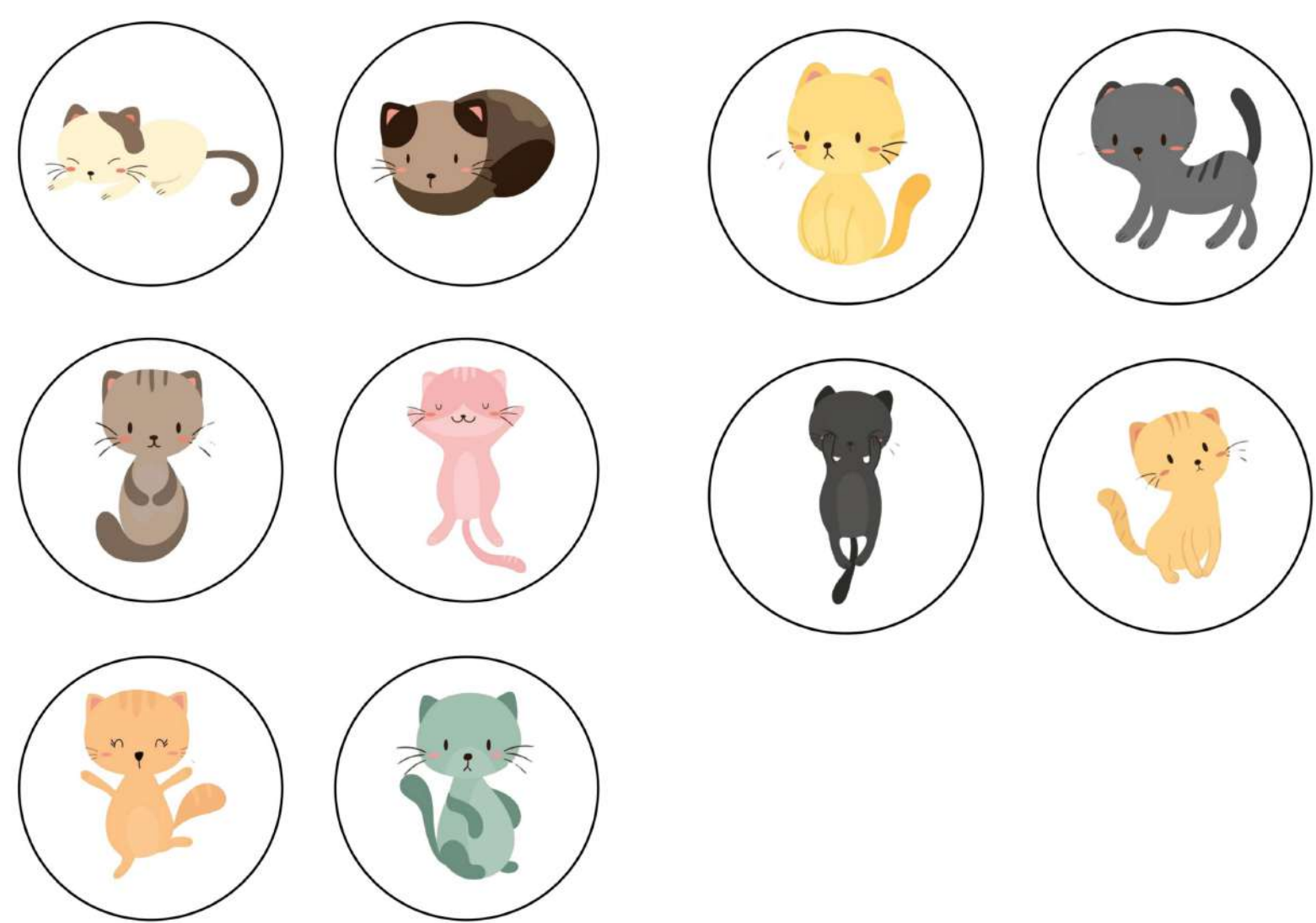
Ok. 45 – 90 minut

Wprowadzenie:

Nauczyciel wita się z dziećmi i mówi, że temat dzisiejszych zajęć poznają, kiedy wykonają pierwsze zadanie.

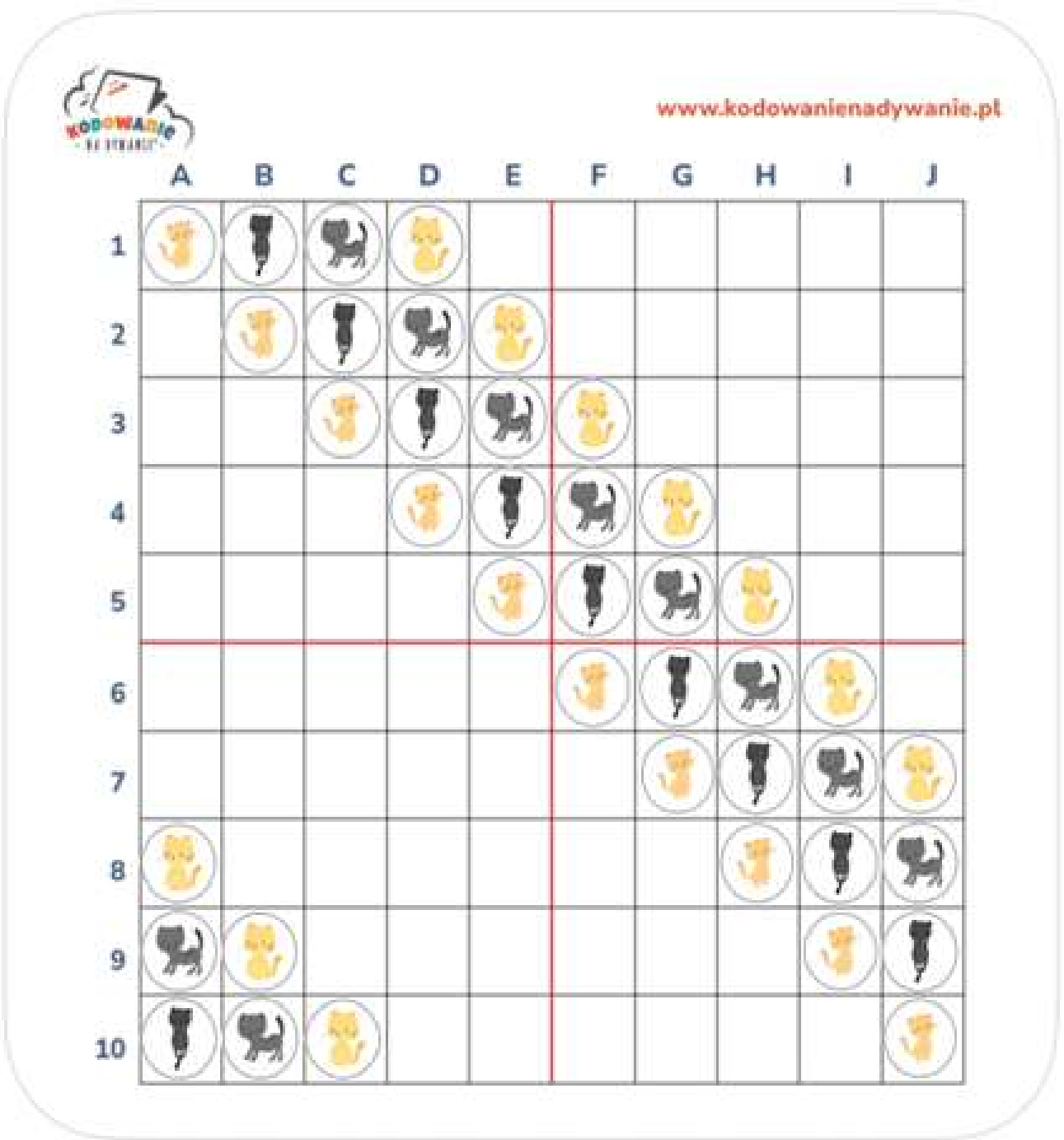
Do pierwszego zadania potrzebna będzie mata do kodowania oraz krążki z materiału pomocniczego „Sudoku 1-2” (wydrukowane w 10 egzemplarzach – lub większej, w zależności od tego, czy praca będzie odbywać się całym zespołem klasowym, czy też w podziale na grupy) (rys. 1). Przygotowana aktywność będzie polegała na takim rozłożeniu wszystkich krążków na macie do kodowania, żeby ten sam obrazek nie powtórzył się ani w linii pionowej, ani w linii poziomej.

Rysunek 1. Materiał pomocniczy „Sudoku 1-2”



Źródło: opracowanie własne A. Świć.

Rysunek 2. Krążki rozłożone na macie do kodowania



Źródło: opracowanie własne A. Świć.

Nauczyciel wyjaśnia dzieciom, że dziś skupią się na lekturze, której bohaterem jest kot o bardzo nietypowym jak na kota imieniu. Pyta uczniów, czy wiedzą, którą lekturą będą omawiać na dzisiejszych zajęciach.

Część właściwa:

Nauczyciel rozkłada krążki z materiału pomocniczego „Obrazki 1-2” obrazkami do dołu, dzieli uczniów na 3-4 zespoły. Każdy zespół podchodzi i losuje po jednym krążku (aż do rozdania wszystkich krążków).

Rysunek 3. Materiał pomocniczy „Obrazki 1-2”

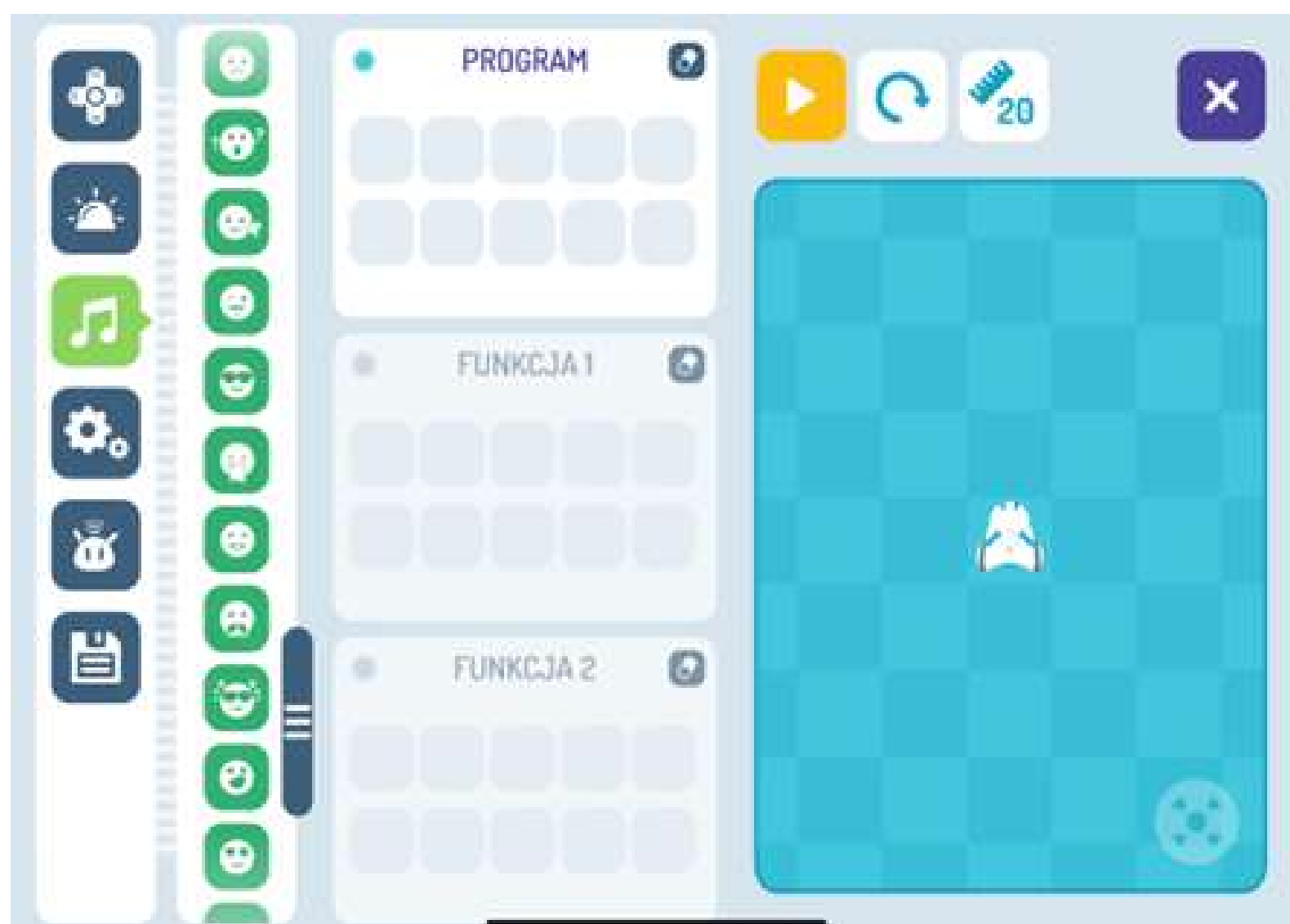


Źródło: opracowanie własne A. Świć.

Każdy zespół ma za zadanie opowiedzieć jaka historia łączyła się z poszczególnymi grafikami lub jaki związek może mieć dana ilustracja z omawianą książką (przykładowo: 4 – Cukierek jest jednym z czworga kociąt, 8 – kiedy Cukierek miał osiem tygodni w jego domu pojawił się Marcel, kartka – wiersz napisany przez Marcela itp. – nauczyciel powinien przyjąć każdą interpretację uczniów, jeśli jest dobrze uzasadniona, nie tylko skojarzenia, które sam wytworzył).

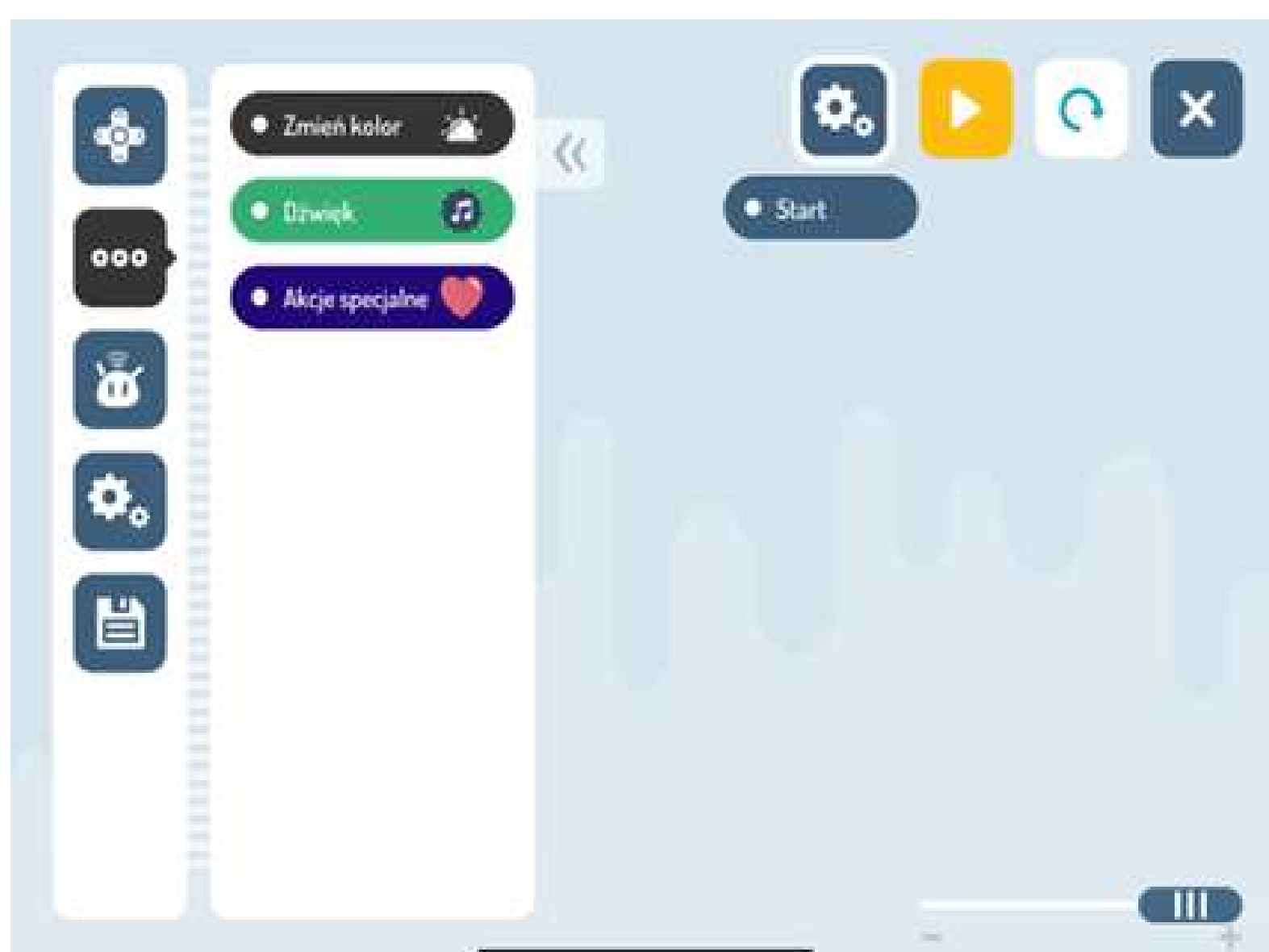
Nauczyciel prosi zespoły o zaprezentowanie swoich interpretacji poszczególnych grafik, a następnie o położenie krążków na losowych polach na macie do kodowania; Nauczyciel tłumaczy na czym będzie polegało zadanie. Robot Photon zamieni się w Cukierka (uczniowie mogą go przy pomocy kolorowego papieru/materiałów ucharakteryzować tak, by przypominał kota). Każdy zespół zaprogramuje robota, w taki sposób, żeby Cukierek (robot) dotarł na pole, na którym znajduje się grafika przedstawiająca najciekawszy fragment książki (grupa musi uzgodnić wspólnie swoje zdanie). Nauczyciel prosi zespoły o zaprezentowanie stworzonych przez siebie kodów. Uczniowie wybierają, czy zaprogramują robota wykorzystując interfejs Badge czy Blocks.

Rysunek 4. Interfejs Photon Badge



Źródło: opracowanie własne A. Świć.

Rysunek 5. Interfejs Photon Blocks



Źródło: opracowanie własne A. Świć.

W następnym zadaniu również zostanie zaprogramowany robot, ale tym razem jeden, wspólnymi siłami całej klasy (lub dalej w grupach). Cukierek, czyli robot będzie musiał dotrzeć na wszystkie pola, na których znajdują się grafiki związane z lekturą, uwzględniając przy tym chronologię zdarzeń. Kiedy robot najedzie na wybraną grafikę, powinien wydać dźwięk emocji. Emocje powinny być dopasowane do sytuacji, które wiążą się z umieszczonymi na krążkach grafikami.

Po zaprogramowaniu robota uczniowie wykonają próbną jazdę. Jeśli robot nie pojedzie zgodnie z ich oczekiwaniami, to odszukają błędy w stworzonym programie, wprowadzą niezbędne poprawki (przygotowanie do debugowania błędów – to proces identyfikowania, lokalizowania i naprawiania błędów (bugów) w kodzie programu) i ponownie wystartują program.

Zakończenie zajęć:

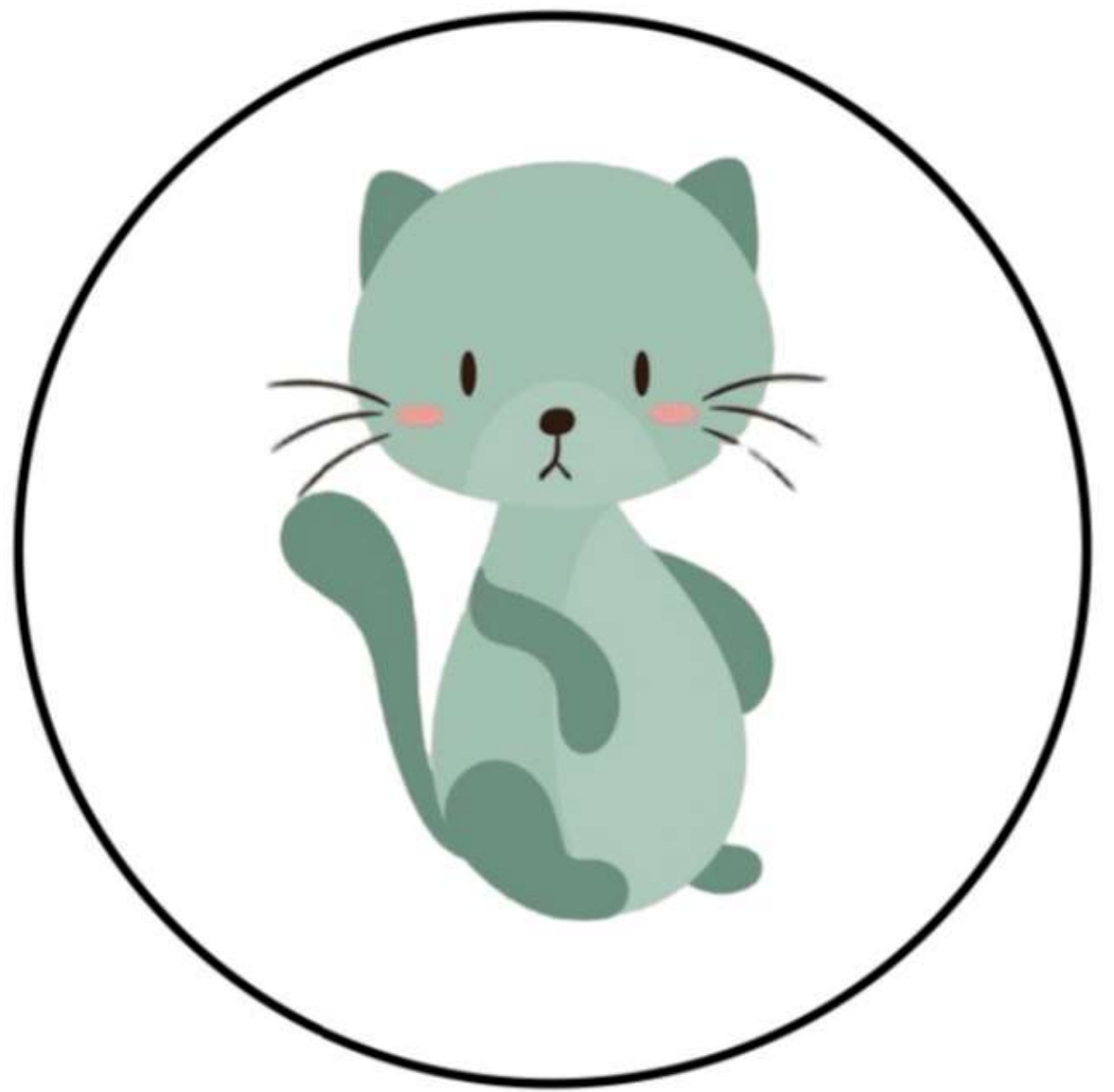
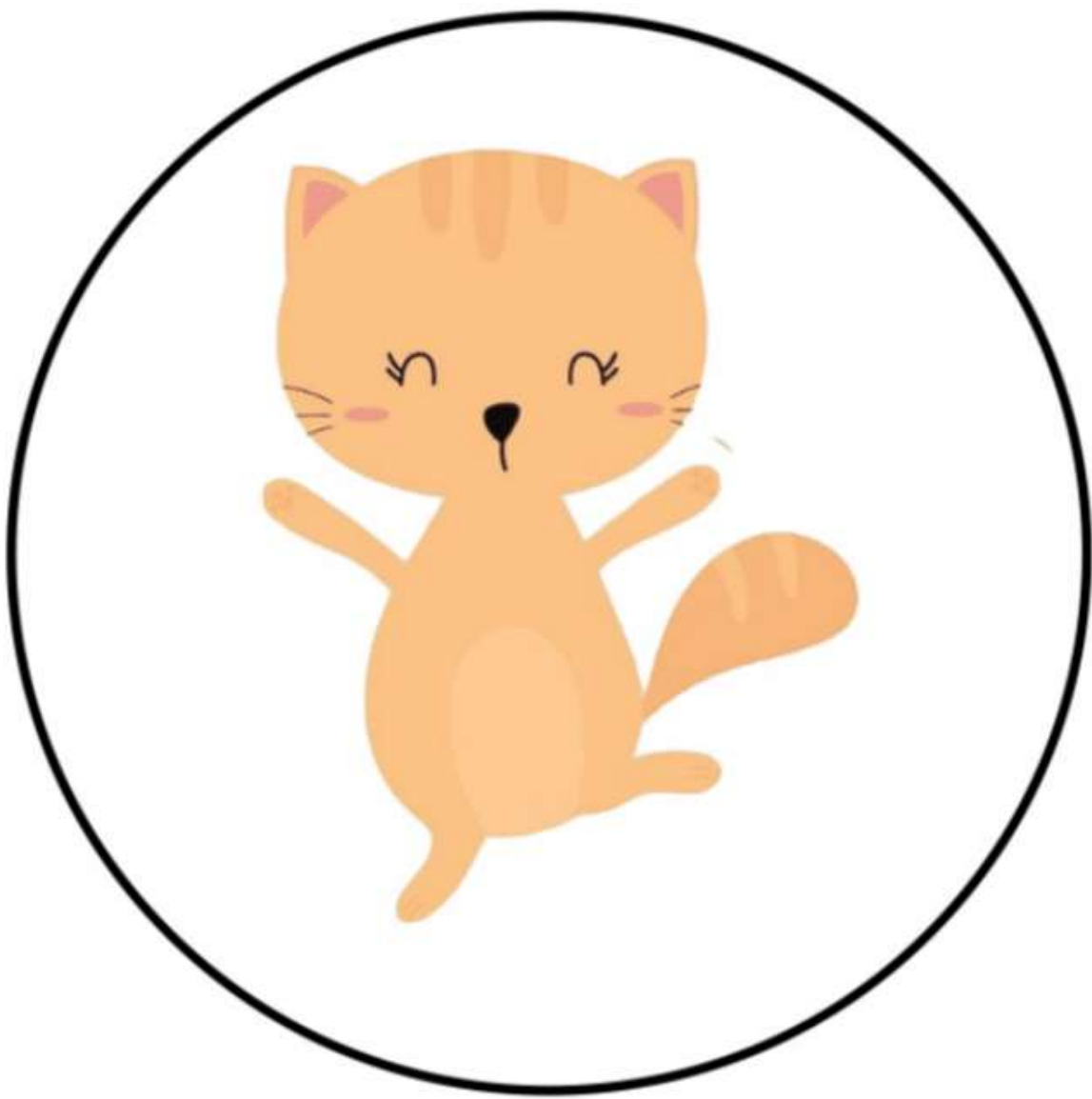
Jako podsumowanie zajęć nauczyciel proponuje dzieciom zbudowanie z kolorowych kubków, na podstawie przygotowanego wzoru, wieży. Prowadzący zajęcia pokazuje uczniom kartkę z wzorem wieży (rys. 6 – „Wieża wzór”). Tłumaczy, że kubki na wzorze są tylko konturowe, nie są wypełnione barwą. Żeby dzieci mogły zbudować wieżę, to muszą najpierw odszyfrować kolor poszczególnych kubków. Pomogą im w tym napisy umieszczone na kubkach i legenda umieszczona nad kubkami. Legenda uzależnia barwę kubka od tego, czy napisane na nim zdanie jest prawdziwe, czy fałszywe. Jeśli prawdziwe, to kubek będzie zielony, jeśli fałszywe, to kubek będzie czerwony.

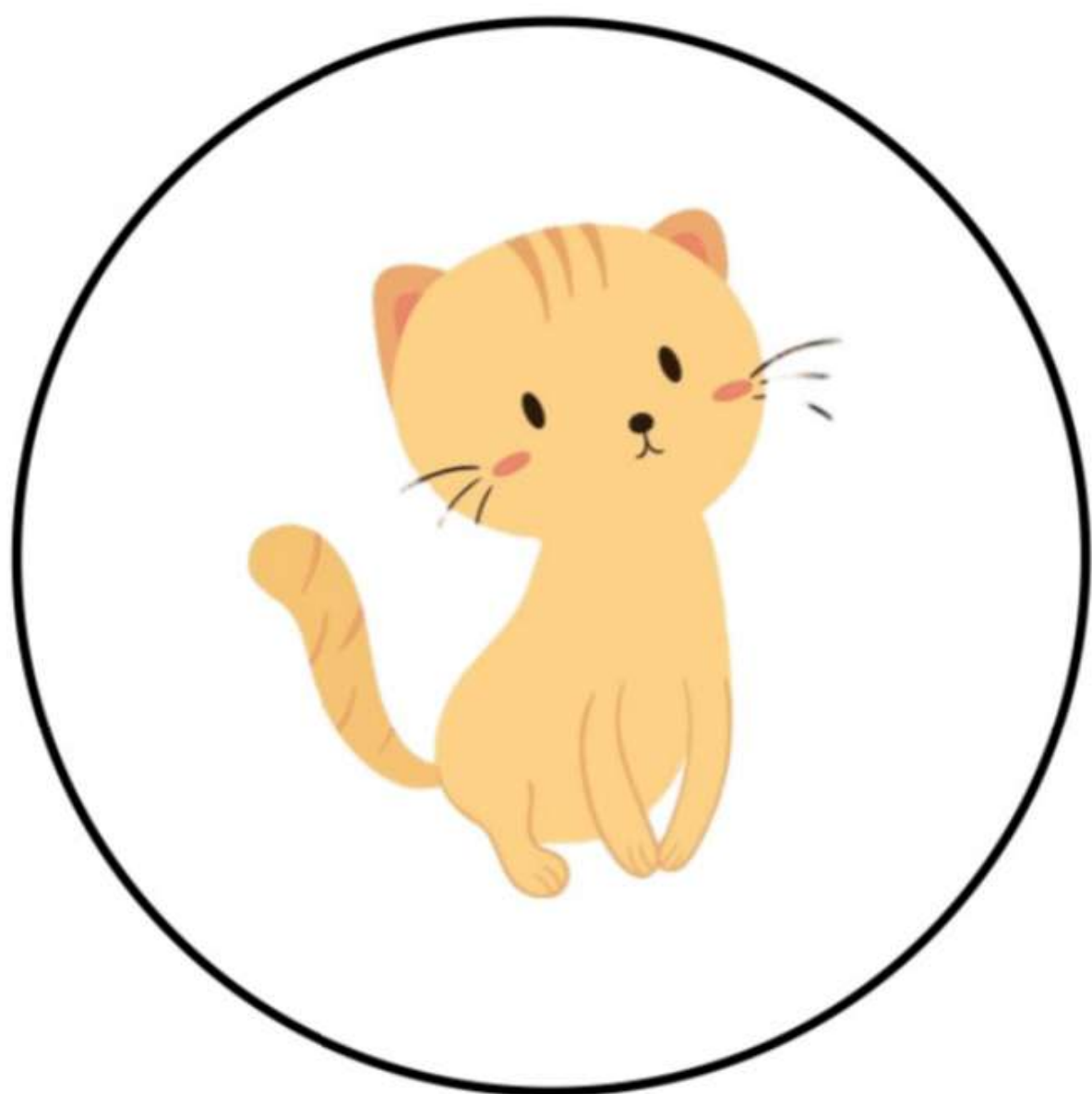
Rysunek 6. Materiał pomocniczy „Wieża – wzór”

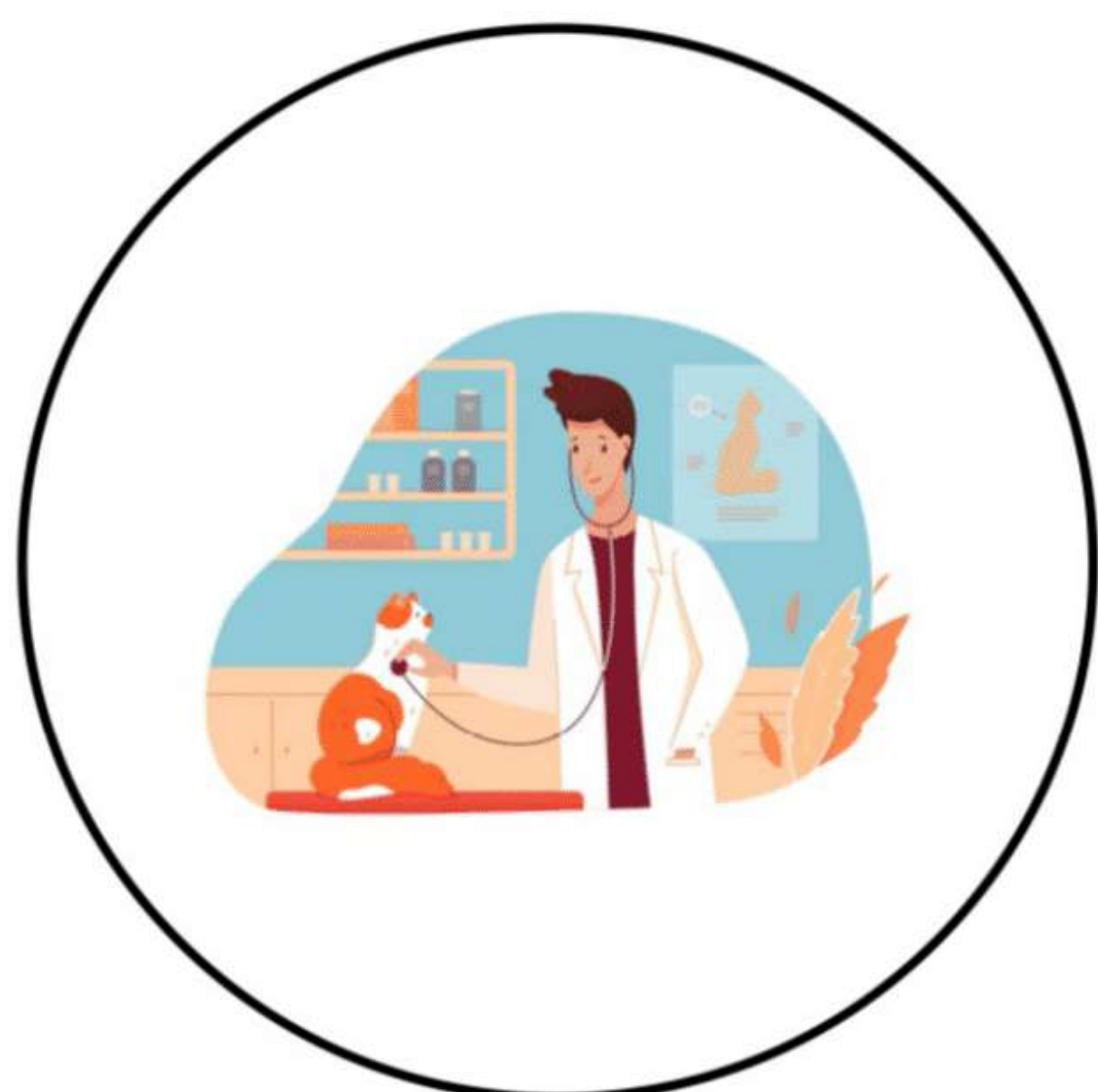


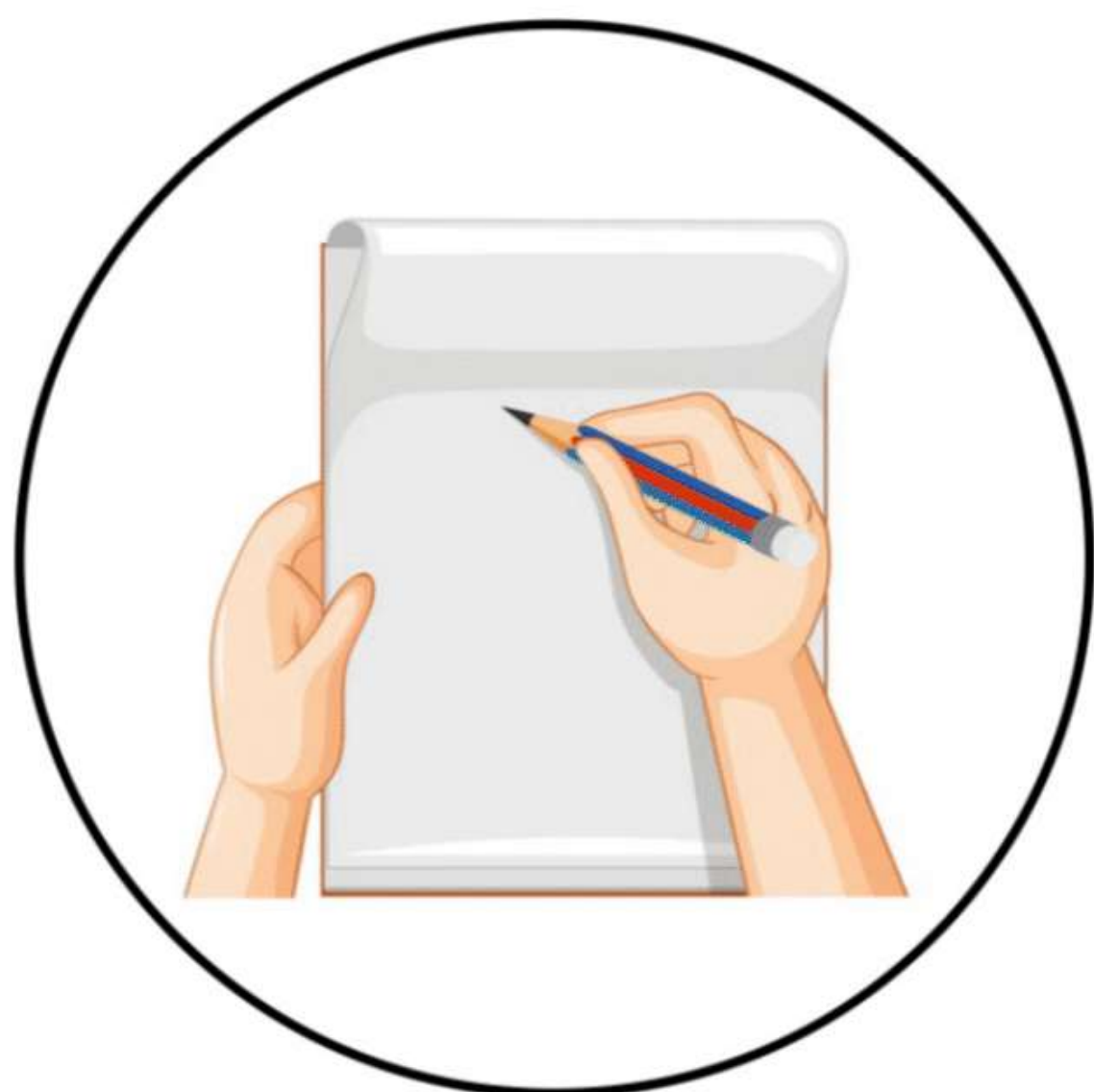
Źródło: opracowanie własne A. Świć.

Nauczyciel dzieli grupę na kilkusobowe zespoły i zaprasza do wykonania zadania. Na koniec prowadzący zajęcia odczytuje zdania, a każde dziecko podnosi zielony, żółty lub czerwony kubek w odpowiedzi. Jeśli zgadza się z przeczytanym zdaniem – podnosi zielony kubek, jeśli nie do końca się zgadza lub nie ma zdania – podnosi żółty kubek; jeśli całkowicie się nie zgadza – podnosi czerwony kubek. Przykładowe zdania: znałam/ znałem wszystkie wydarzenia z lektury; uważam że lektura „Cukeirku, ty łobuzie” jest interesująca; potrafię zaprogramować robota Photon według polecenia; zaprogramowanie robota Photon jest dla mnie problematyczne itp.









FAŁSZ

PRAWDA



CUKIEREK
BYŁ JEDNYM
Z CZWORG
KOCIĄT

CUKIEREK MIAŁ
SZEŚĆ TYGODNI
KIEDY ZABRAŁ
GO DO DOMU
MARCEL

CUKIEREK BAŁ
SIĘ WIZYTU U
WETERYNARZA

CUKIEREK
PRZYCZYNIŁ
SIĘ DO
ZNISZCZENIA
SWETRA MAMY

CUKIEREK
BYŁ JEDNYM
Z SZÓSTKI
KOCIĄT

CUKIEREK MIAŁ
OSIEM TYGODNI
KIEDY ZABRAŁ
GO DO DOMU
MARCEL

CUKIEREK
CZekał Z
NIECIERPLIWO
ŚCIĄ NA
WIZYTĘ U
DENTYSTY

CUKIEREK
UCIEKŁ Z
DOMU

CUKIEREK
NIGDY NIE
JECHAŁ
CIĘŻARÓWKĄ

MARCEL
UCZĘSZCZAŁ
NA TRENINGI
HOKEJA

MARCEL
UCZĘSZCZAŁ
NA TRENINGI
SIATKÓWKI

**RUCH DROGOWY
BEZPIECZNIE
ZAPROGRAMOWANY**

Ucząc podstaw programowania warto odnosić się do sytuacji znanych, bliskich dzieciom, bazować na ich własnych doświadczeniach. Na zajęciach przeprowadzonych według zaproponowanego scenariusza uczniowie stworzą z klocków konstrukcyjnych i materiałów plastycznych miasteczko ruchu drogowego, po którym będą poruszać się zaprogramowane przez nich roboty edukacyjne.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe;
- Kompetencje w zakresie świadomości i ekspresji kulturalnej.

Cele ogólne:

- Poszerzenie wiadomości na temat bezpiecznego poruszania się po drodze;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie miękkich kompetencji, ze szczególnym uwzględnieniem umiejętności pracy zespołowej.

Cele operacyjne:

Uczeń podczas zajęć:

- tworzy obrazek na podstawie podanych współrzędnych;
- wyjaśnia czym różni się sygnalizator przeznaczony dla pojazdów od sygnalizatora przeznaczonego dla pieszych;
- tworzy algorytm bezpiecznego przechodzenia na drugą stronę ulicy;
- tworzy przestrzenne konstrukcje z klocków oraz różnego rodzaju materiałów plastycznych;
- programuje robota Photon wykorzystując interfejs Photon Photon Blocks;
- chętnie i zgodnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, klocki konstrukcyjne, materiały plastyczne, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

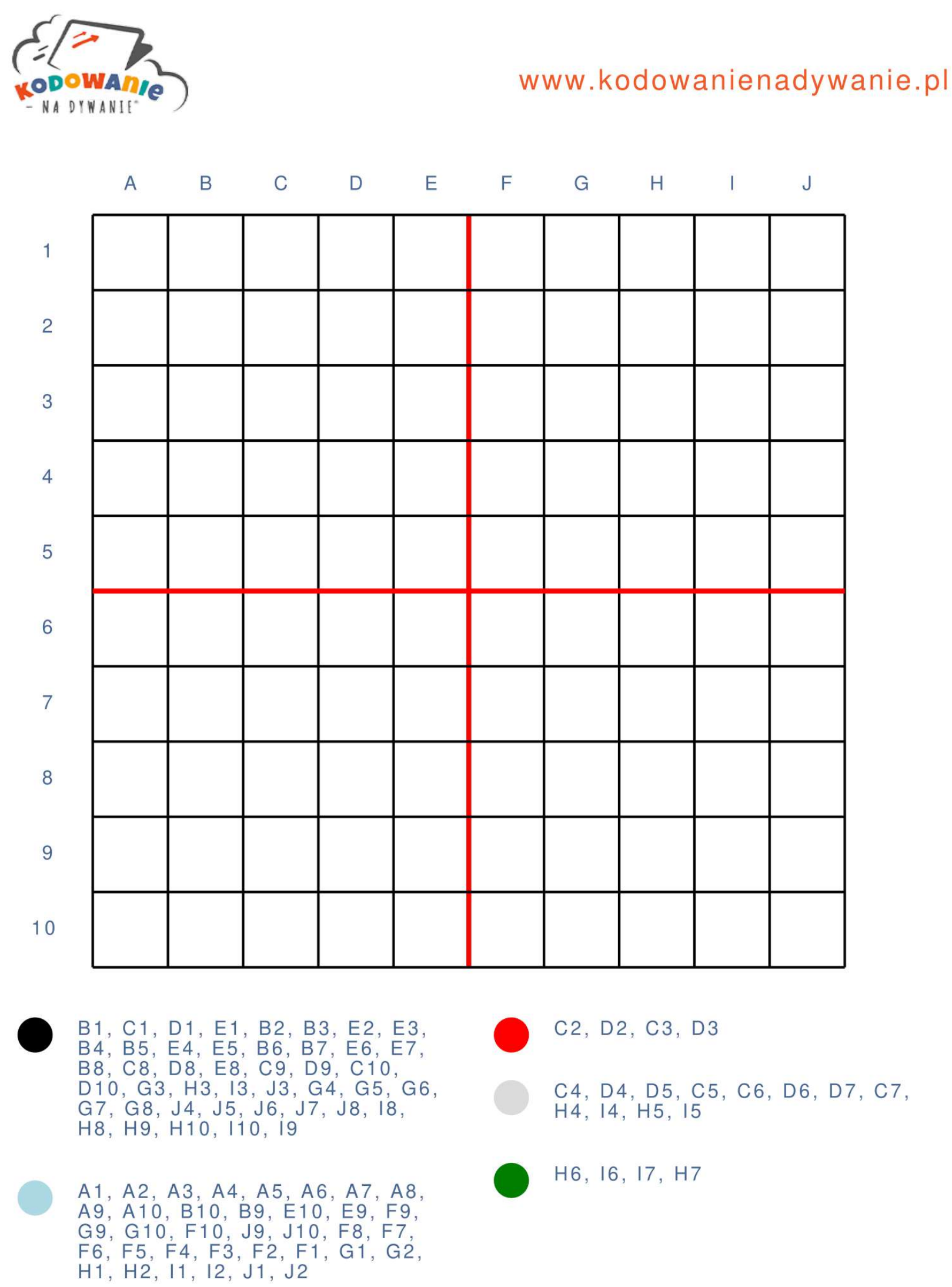
Ok. 45 – 90 minut

Wprowadzenie

Nauczyciel wita się z dziećmi i prezentuje cel zajęć, jakim będzie poznanie zasad ruchu drogowego i bezpiecznego poruszania się po drodze. Zaprasza uczniów do wykonania pierwszego zadania.

Przygotowane zadanie będzie polegało na ułożeniu kolorowych kubków na macie do kodowania, tak by utworzyły obrazek. Wzór uczniowie ułożą na podstawie podanych współrzędnych. Nauczyciel rozkłada matę do kodowania, obok maty stawia kolorowe kubki i kładzie kartę z rozpisanymi koordynatami (rys. 1 – materiał pomocniczy „Sygnalizator współrzędne”). Prosi dzieci o wykonanie zadania.

Rysunek 1. Materiał pomocniczy „Sygnalizator współrzędne”



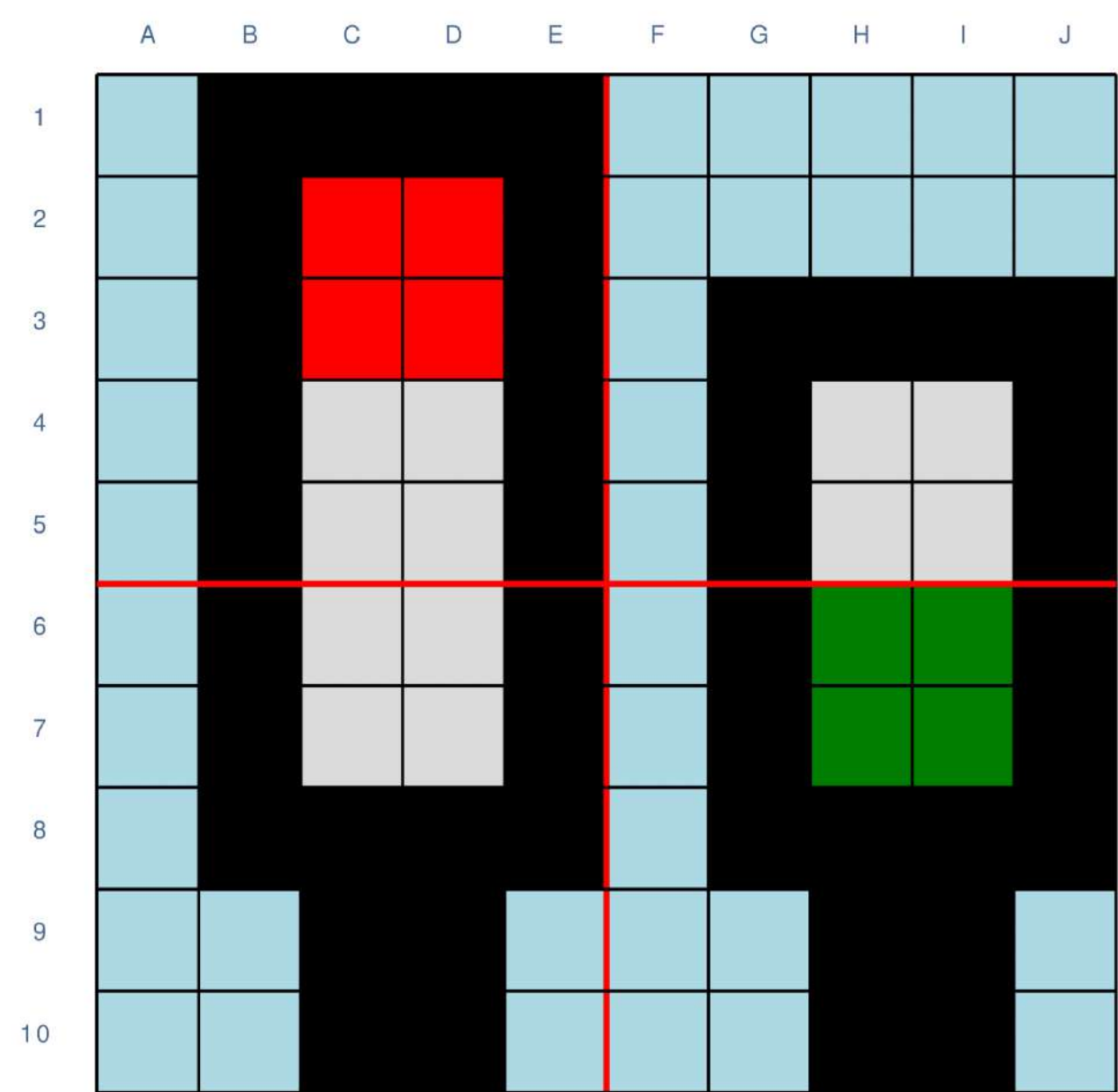
Źródło: opracowanie własne A. Świć.

Uczniowie wraz z nauczycielem omawiają obrazek, który powstał na macie do kodowania (rys. 2). Prowadzący zajęcia pyta o to, czy sygnalizatory z lewej i prawej strony są takie same, czy czymś się różnią. Dlaczego w jednym jest miejsce tylko na dwa kolory świateł, a w drugim na trzy. Wspólnie ustalają, z którego z nich będą korzystać piesi, a z którego kierowcy pojazdów.

Rysunek 2. Rysunek wykonany przez uczniów na macie do kodowania „Sygnalizator”



www.kodowanienadywanie.pl

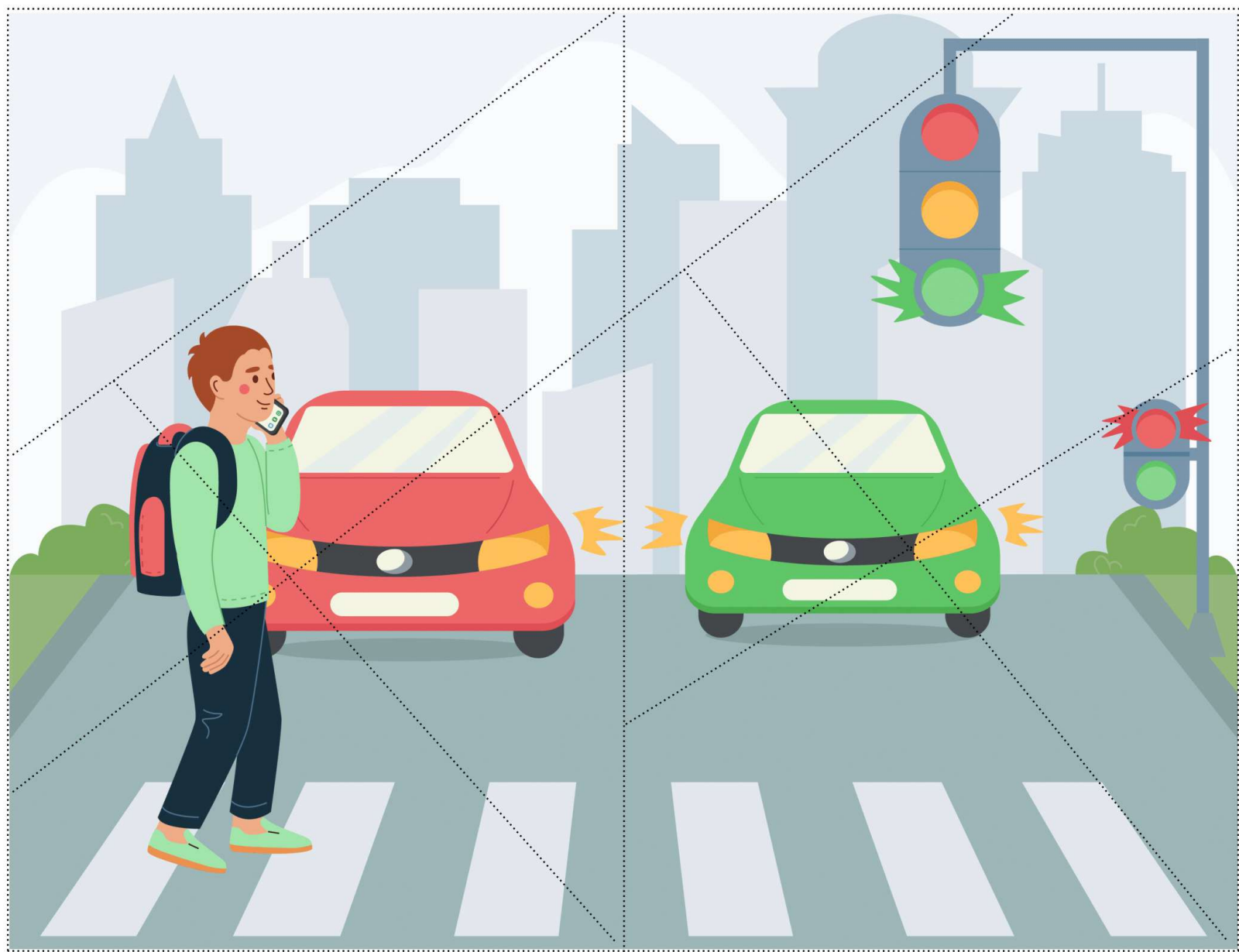


Źródło: opracowanie własne A. Świć.

Część właściwa

Nauczyciel dzieli klasę na kilka grup. Każda grupa otrzymuje do ułożenia puzzle (rys. 3 – „Puzzle”). Po wykonaniu zadania dzieci mówią, co przedstawia złożony w całość obrazek.

Rysunek 3. Materiał pomocniczy „puzzle”



Źródło: opracowanie własne A. Świć.

Prowadzący zajęcia prosi dzieci o zwrócenie uwagi na zachowanie chłopca z ilustracji (np.: rozmawianie przez telefon). Prosi też o ustalenie, czy chłopiec przechodzi na drugą stronę w sposób prawidłowy, czy nie (uczniowie muszą przyrzeć się widocznym na obrazkom sygnalizatorom świetlnym i sprawdzić, które światło świeci się dla kierowców pojazdów, a które dla pieszych).

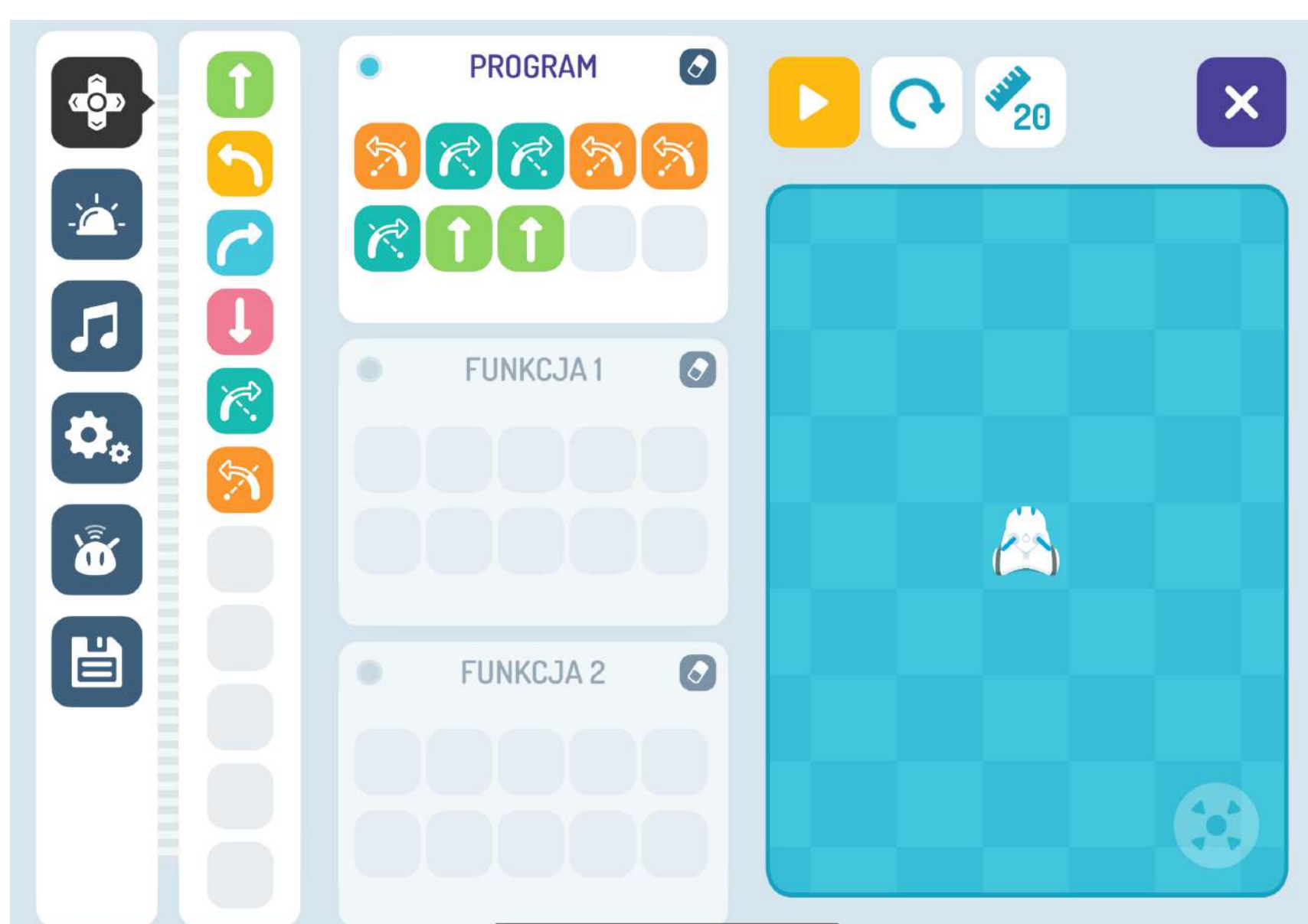
Uczniów można prosić do odpowiedzi korzystając z różnych strategii np. słoik z imionami – imię każdego ucznia zapisane jest na patyczku, a wszystkie patyczki włożone są do słoika; koło losujące – na kole fortuny wypisane są imiona dzieci, losuje się poprzez zakręcenie kołem; generator liczb losowych – każde dziecko ma przydzielony numer, w aplikacji należy wpisać zakres odpowiadający liczbie uczniów w klasie, a następnie losować liczbę itp.

Uczniowie rozmawiają o tym, w jaki sposób można przejść na drugą stronę ulicy, jeśli nie ma na niej sygnalizacji świetlnej.

Nauczyciel rozkłada matę do kodowania (tematyczną z ulicą lub zwykłą, na której przy pomocy kolorowych kartek zostanie ułożone przejście dla pieszych). Stawia robota przed przejściem dla pieszych. Prosi uczniów o takie zaprogramowanie robota, żeby bezpiecznie przeszedł na drugą stronę ulicy. Zwraca uczniom uwagę na to, co oni robią kiedy są przed przejściem dla pieszych – wykonują za każdym razem ten sam schemat. Podchodząc do przejścia najpierw patrzymy, czy nic nie jedzie z lewej strony, jeśli nie jedzie, to patrzymy, czy coś nie jedzie ze strony prawej. Kiedy jest pusto, to ponownie sprawdzamy lewą stronę. Jeśli i tym razem jest pusto, to możemy bezpiecznie przejść na drugą stronę ulicy. Prowadzący zajęcia tłumaczy, że schemat, który wykonujemy zawsze przed przejściem na drugą stronę ulicy można by było nazwać algorytmem bezpiecznego przejścia przez jezdnię.

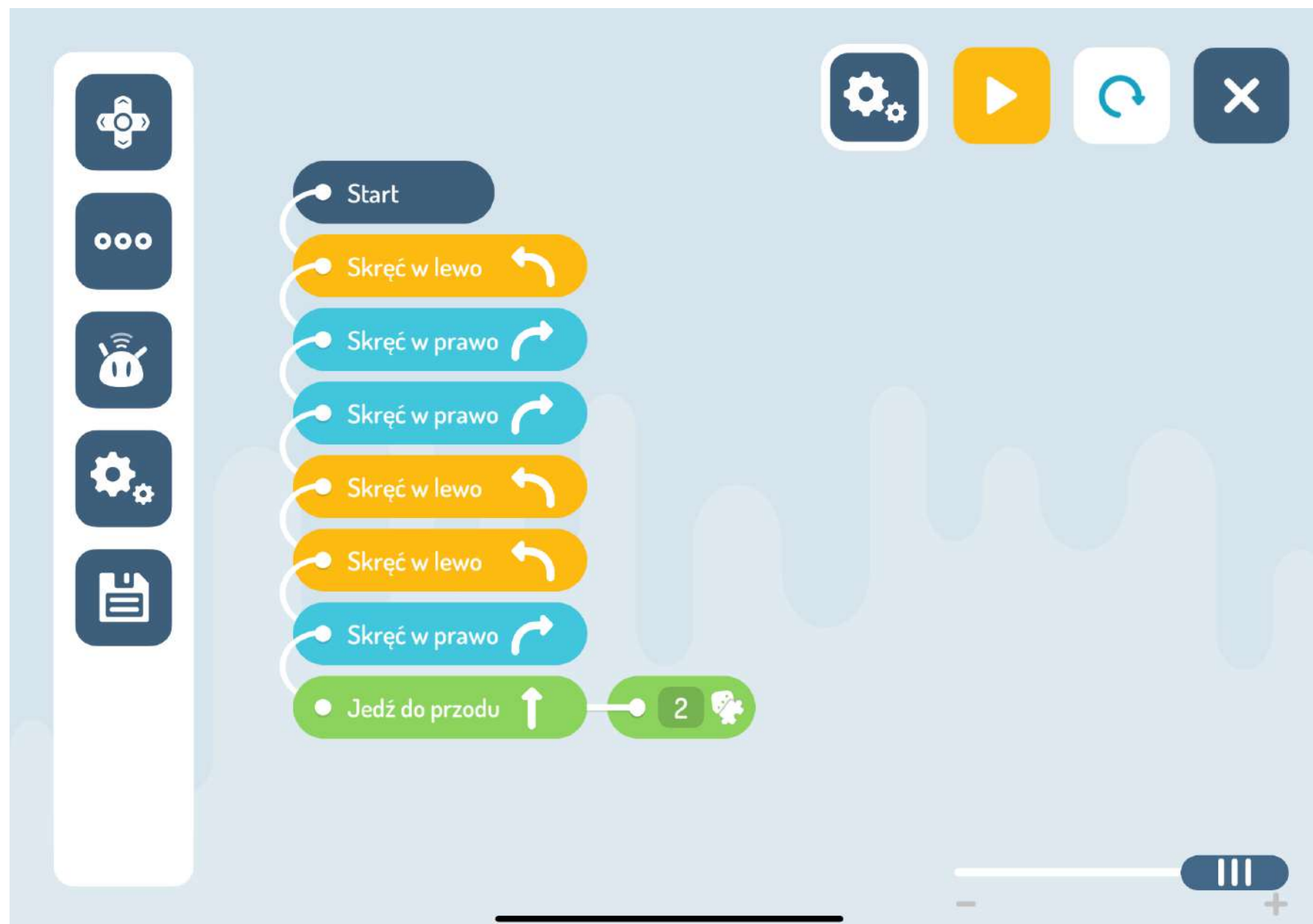
Uczniowie w kilkusobowych zespołach wykonują powierzone zadanie. Mogą wykorzystać interfejs Photon Badge lub Photon Blocks.

Rysunek 4. Interfejs Photon Badge



Źródło: opracowanie własne A. Świć.

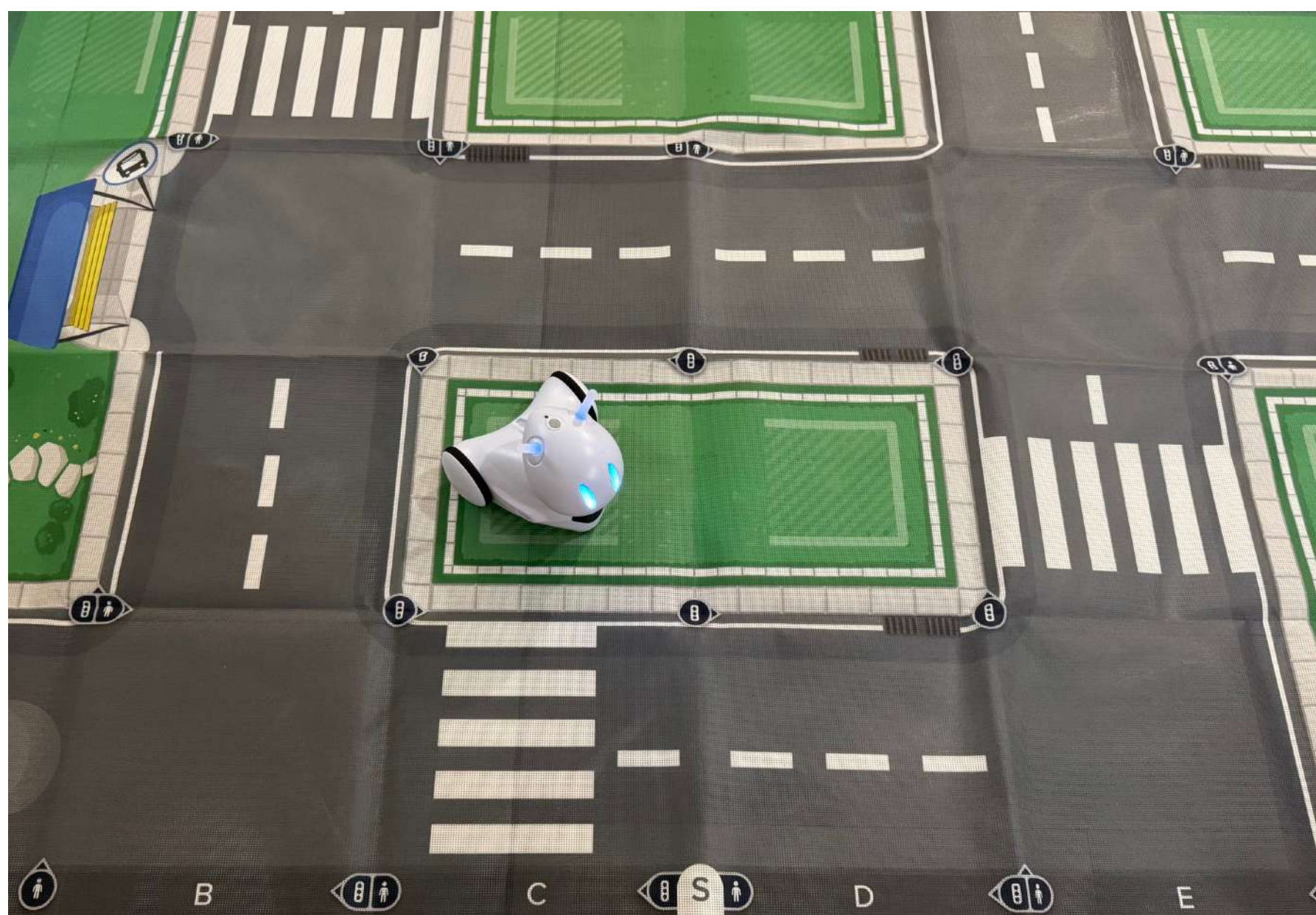
Rysunek 5. Interfejs Photon Blocks



Źródło: opracowanie własne A. Świć.

Zespoły po wykonaniu zadania prezentują stworzone programy i testują je pod kątem sprawdzenia poprawności utworzonego kodu.

Rysunek 6. Robot Photon przechodzący przez przejście dla pieszych

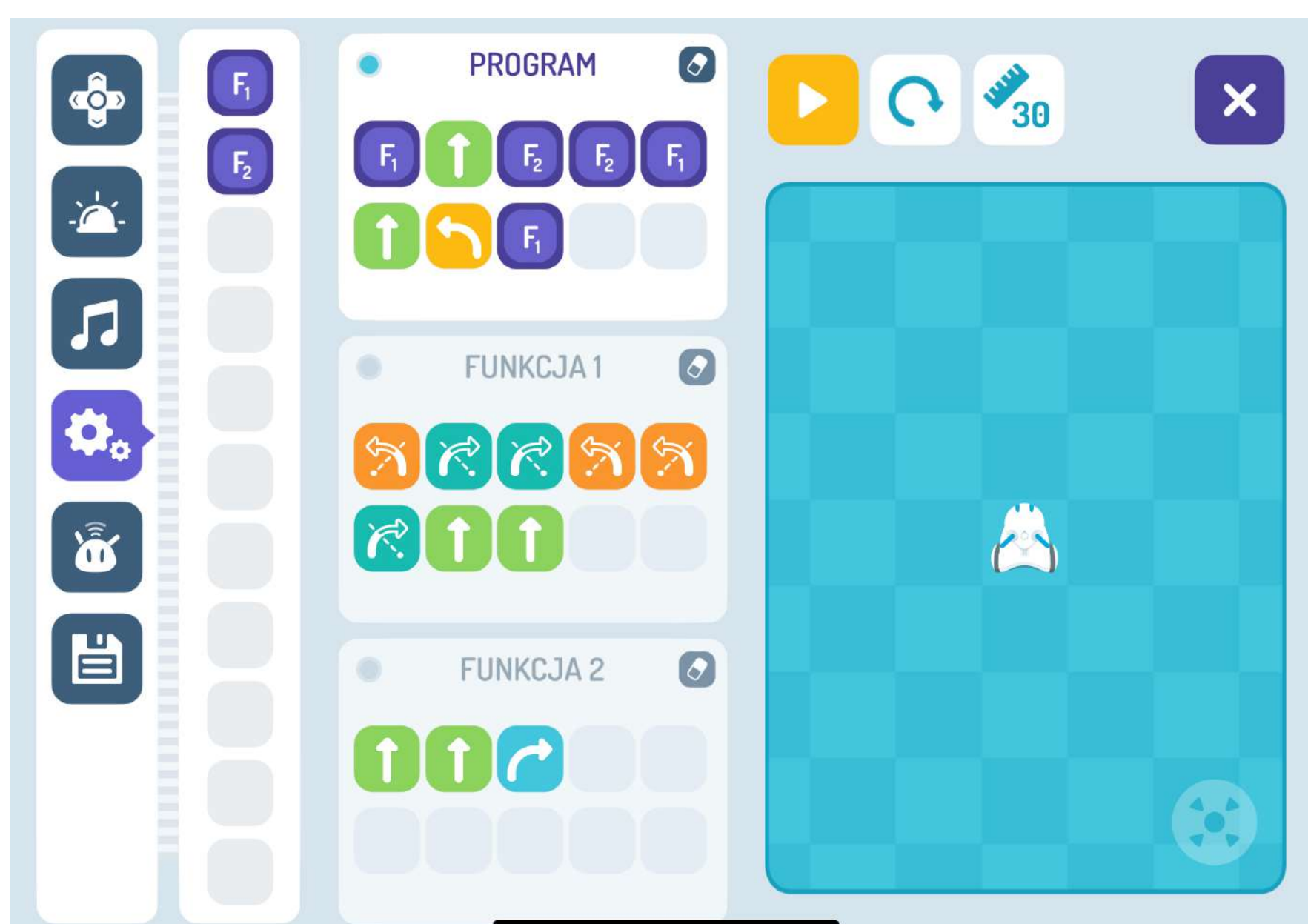


Źródło: opracowanie własne A. Świć.

W kolejnym kroku uczniowie mają za zadanie wykonać trudniejsze zadanie. Robot najpierw przejdzie na drugą stronę ulicy, następnie podjedzie do drugiego przejścia dla pieszych, ponownie przejdzie na drugą stronę ulicy, następnie podjedzie do jeszcze jednego przejścia dla pieszych i po raz trzeci przejdzie na drugą stronę ulicy.

Jeśli uczniowie do poprzedniego zadania wykorzystywali interfejs Photon Badge, to wiedzą, że bezpieczne przejście na drugą stronę ulicy zajęło większą część przestrzeni przeznaczanej na stworzenie programu, a w aktualnym zadaniu robot będzie trzykrotnie przechodził na drugą stronę ulicy i dodatkowo będzie pokonywał jeszcze odcinki łączące poszczególne fragmenty trasy. Cały kod będzie za długi, żeby można było go zmieścić w przestrzeni „Program”. W tej sytuacji potrzebne będzie stworzenie funkcji, która będzie odpowiedzialna za wywołanie w programie sekwencji komend odpowiedzialnych za bezpieczne przejście na drugą stronę ulicy. Część stworzoną w poprzednim zadaniu można „przenieść” do okienka „Funkcja 1”, a w przestrzeni przeznaczanej na program stworzyć jego finalną wersję łączącą bezpieczne przejścia na drugą stronę ulicy „Funkcję 1” z fragmentami odpowiedzialnymi za poruszanie się między przejściami dla pieszych (może zdarzyć się tak, że nadal miejsca będzie zbyt mało, w tej sytuacji należy skorzystać również z Funkcji 2 i w niej umieścić wybraną sekwencję komend, zyskując w ten sposób miejsce w przestrzeni przeznaczanej na stworzenie kodu).

Rysunek 7. Interfejs Photon Badge – przejście przez kilka przejść dla pieszych



Źródło: opracowanie własne A. Świć.

W dalszej części zajęć, korzystając z klocków konstrukcyjnych lub/i materiałów plastycznych dzieci dodają różne elementy do ich miasteczka ruchu drogowego, a następnie programują roboty na przejazdy przez wybrane przez siebie trasy. Warto zaproponować opcję, w której po ulicy jednocześnie będą poruszały się 2-4 roboty i tworząc programy, trzeba będzie uwzględnić nie tylko sposób poruszania się robota, którego programujemy, ale również pozostałych robotów.

Zakończenie zajęć

Jako podsumowanie zajęć uczniowie zamienią stworzone wcześniej miasteczko ruchu drogowego na miasto przyszłości. Projekt wykonują na kartkach w formacie A3. Każdy zespół ponadto stworzy i zaprojektuje znak drogowy, który aktualnie jeszcze nie istnieje – znak, który miałby szansę pojawić się w przyszłości. Zadaniem zespołów jest podanie trzech zasad, które są najważniejsze dla zachowania bezpieczeństwa na drodze.

Po wykonaniu zadania zespoły prezentują wyniki swojej pracy.

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



B1, C1, D1, E1, B2, B3, E2, E3, B4, B5, E4, E5, B6, B7, E6, E7, B8, C8, D8, E8, C9, D9, C10, D10, G3, H3, I3, J3, G4, G5, G6, G7, G8, J4, J5, J6, J7, J8, I8, H8, H9, H10, I10, I9



C2, D2, C3, D3



C4, D4, D5, C5, C6, D6, D7, C7, H4, I4, H5, I5



A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, B10, B9, E10, E9, F9, G9, G10, F10, J9, J10, F8, F7, F6, F5, F4, F3, F2, F1, G1, G2, H1, H2, I1, I2, J1, J2



H6, I6, I7, H7



ZAPROGRAMOWANA ORTOGRAFIA

Pisownie wyrazów z trudnościami ortograficznymi trzeba ćwiczyć. Nie są to jednak ćwiczenia szczególnie lubiane przez uczniów. Na zajęciach przeprowadzonych według zaproponowanego scenariusza połączymy utrwalanie pisowni wyrazów z trudnościami ortograficznymi z programowaniem robotów. W ten sposób nie tylko uczniowie nauczą się podstaw programowania, ale też wykonają więcej przykładów, w których znajomość zasad ortografii trzeba będzie wykorzystać w praktyce.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe

Cele ogólne:

- Utrwalanie zasad ortograficznych
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie miękkich kompetencji, ze szczególnym uwzględnieniem umiejętności pracy zespołowej

Cele operacyjne:

Uczeń podczas zajęć:

- Wykorzystuje poznane przez siebie zasady ortograficzne podczas praktycznego działania;
- tworzy obrazek na podstawie podanych współrzędnych;
- programuje robota Photon wykorzystując interfejs Photon Badge i Photon Blocks;
- przestrzega zasad fair play podczas gry;
- chętnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

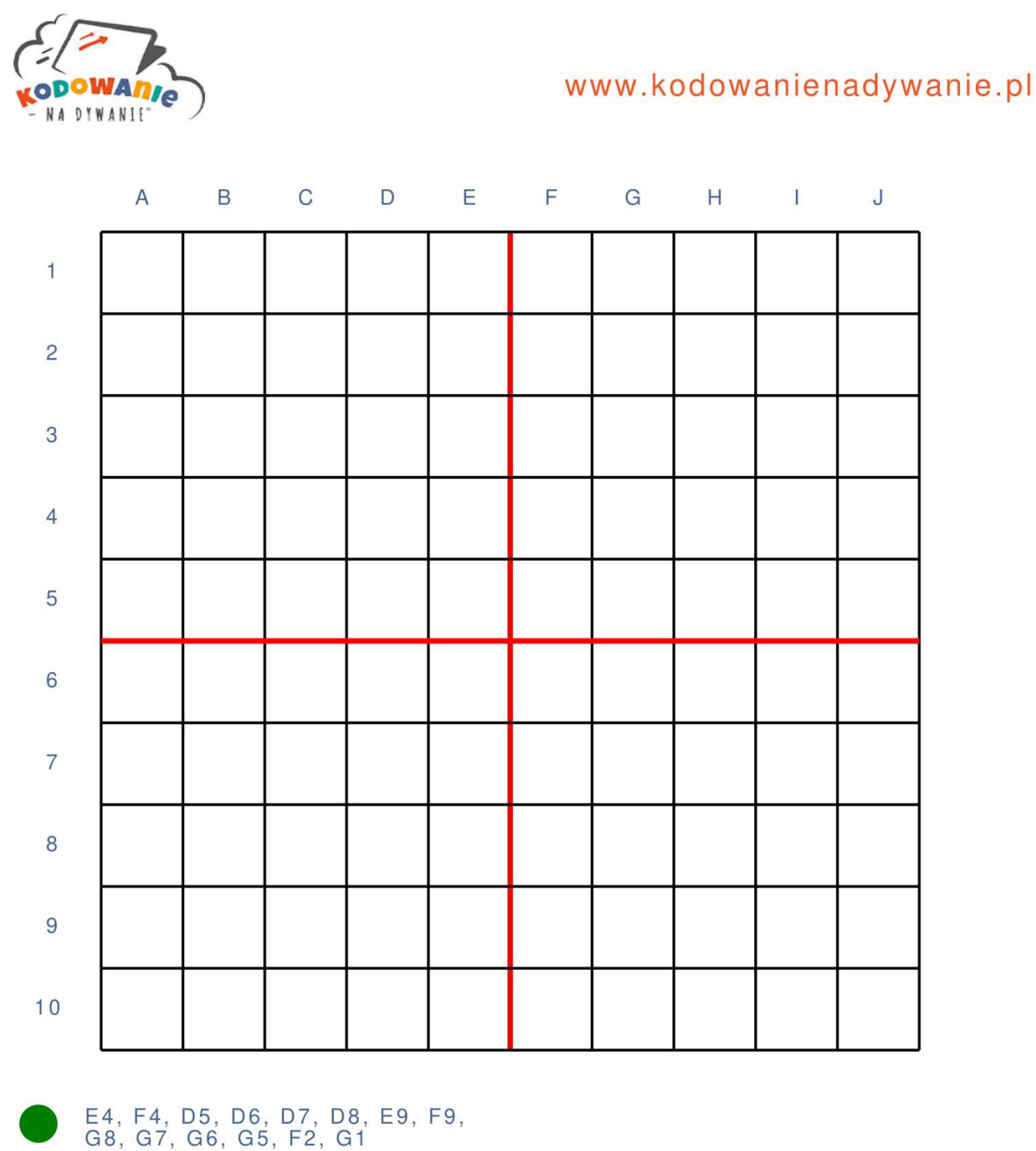
Czas prowadzenia zajęć:

Ok. 45 – 90 minut

Wprowadzenie:

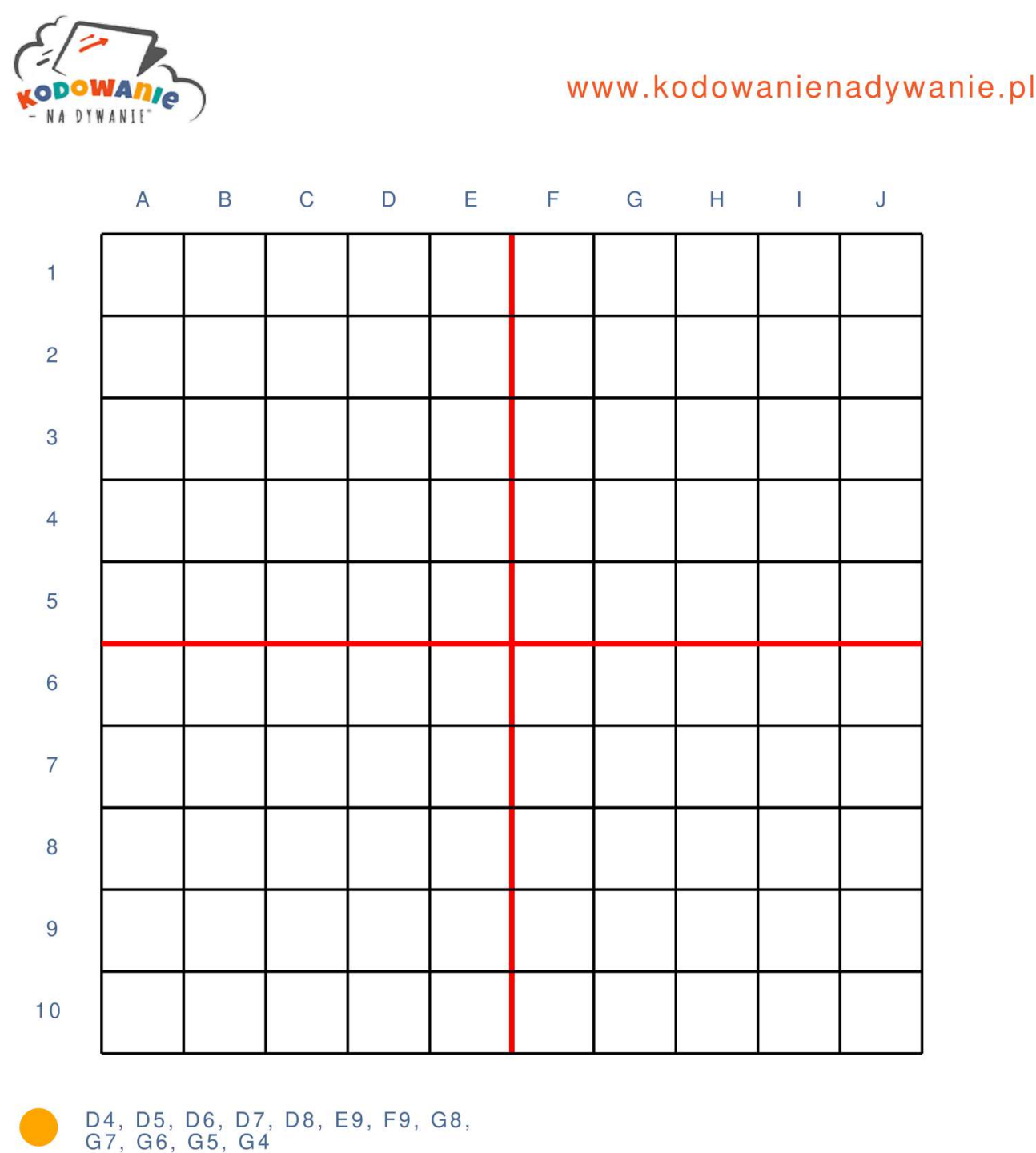
Nauczyciel wita się z uczniami, mówi im, że celem zajęć będzie ćwiczenie, ćwiczenie i jeszcze raz ćwiczenie, a czego? Tego dowiedzą się wykonując pierwsze zadanie. Nauczyciel rozkłada matę do kodowania, obok maty stawia kolorowe kubki i kładzie kartki z rozpisanymi koordynatami (materiał pomocniczy „Ó współrzędne”, „U współrzędne”, „CH współrzędne”, „H współrzędne”, „RZ współrzędne”, „Ż współrzędne”). Prosi dzieci o wykonanie zadania.

Rysunek 1. Materiał pomocniczy „Ó współrzędne



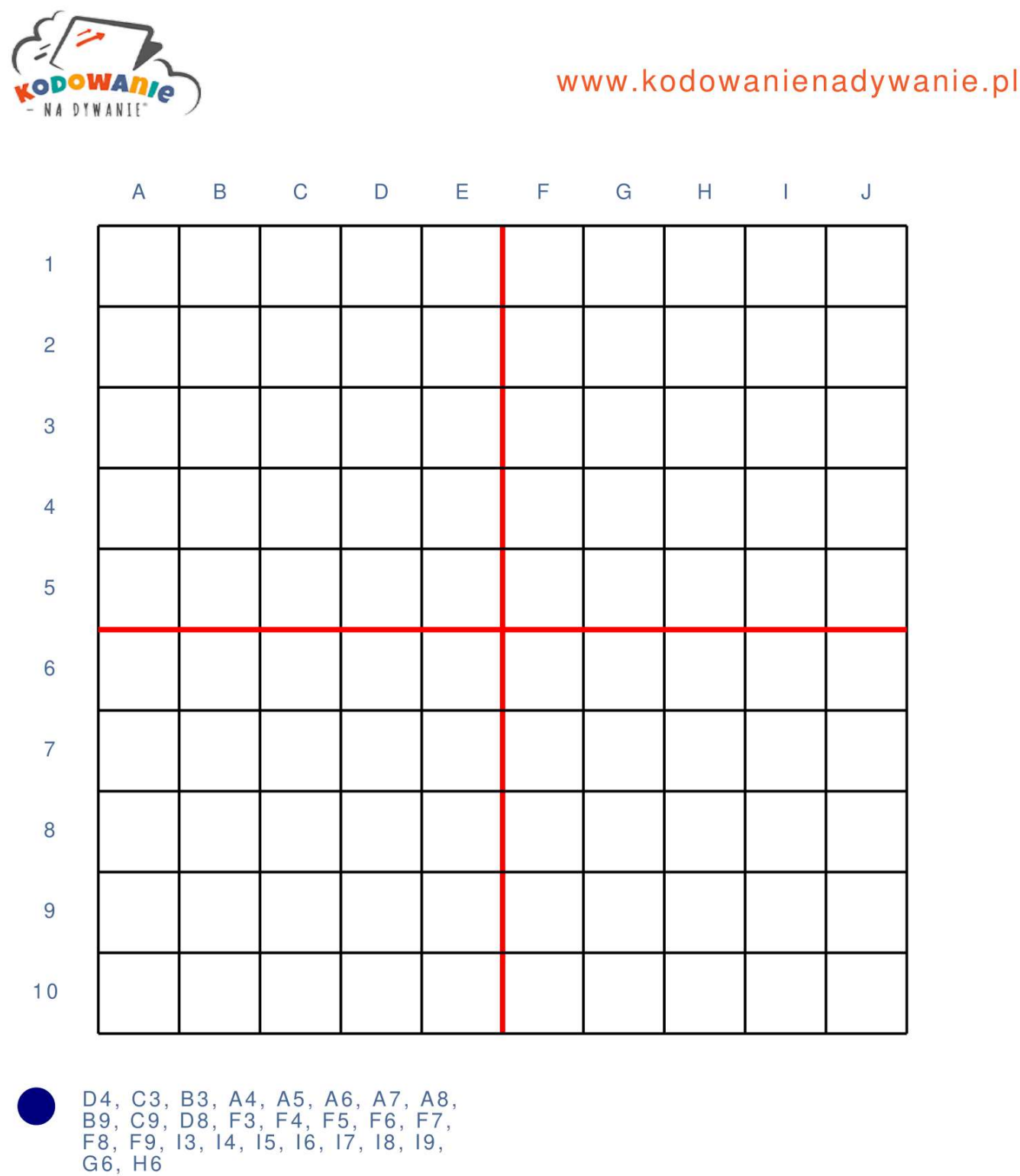
Źródło: opracowanie własne A. Świć.

Rysunek 2. Materiał pomocniczy „U współrzędne”



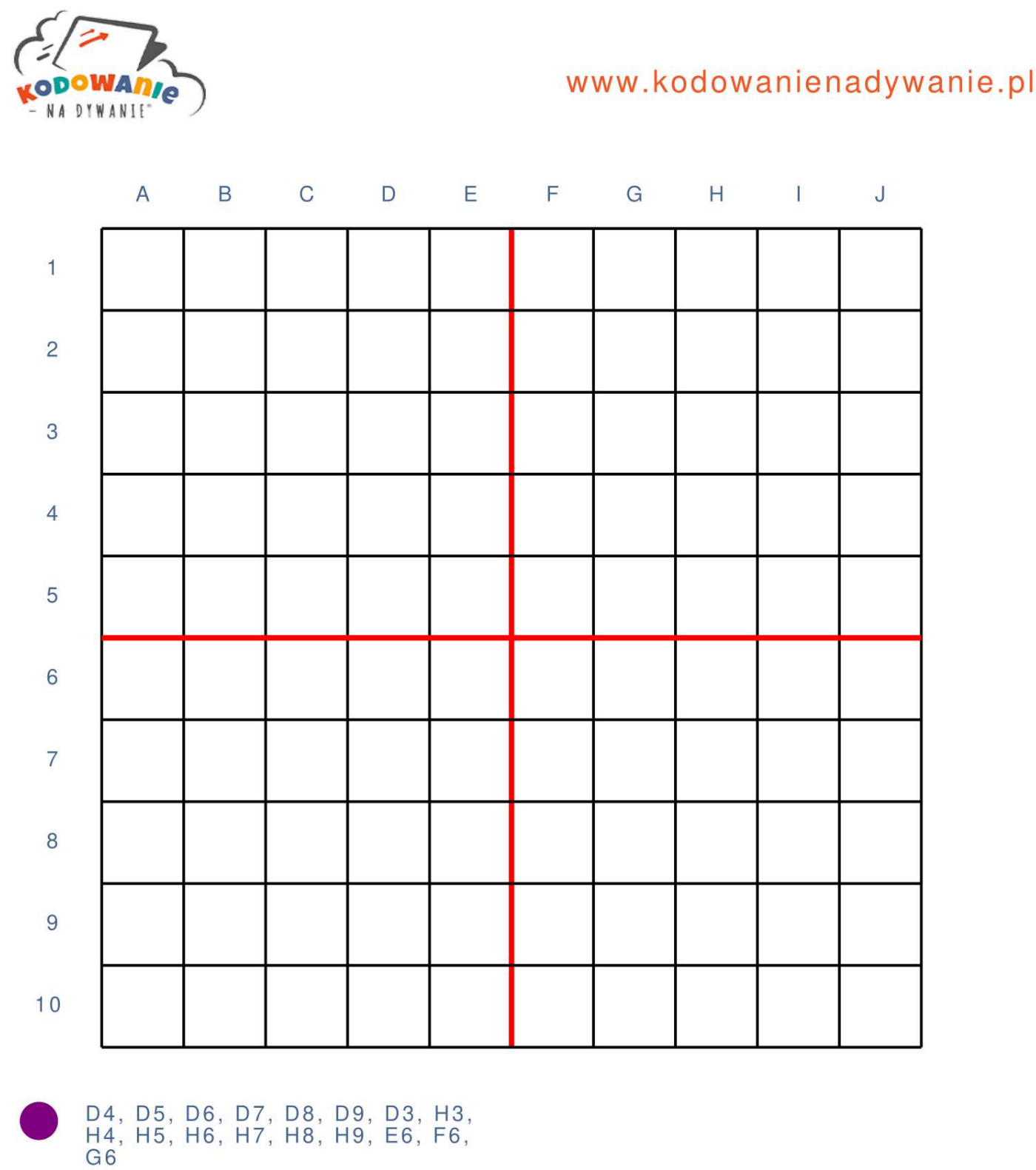
Źródło: opracowanie własne A. Świć.

Rysunek 3. Materiał pomocniczy „CH współrzędne”



Źródło: opracowanie własne A. Świć.

Rysunek 4. Materiał pomocniczy „H współrzędne”

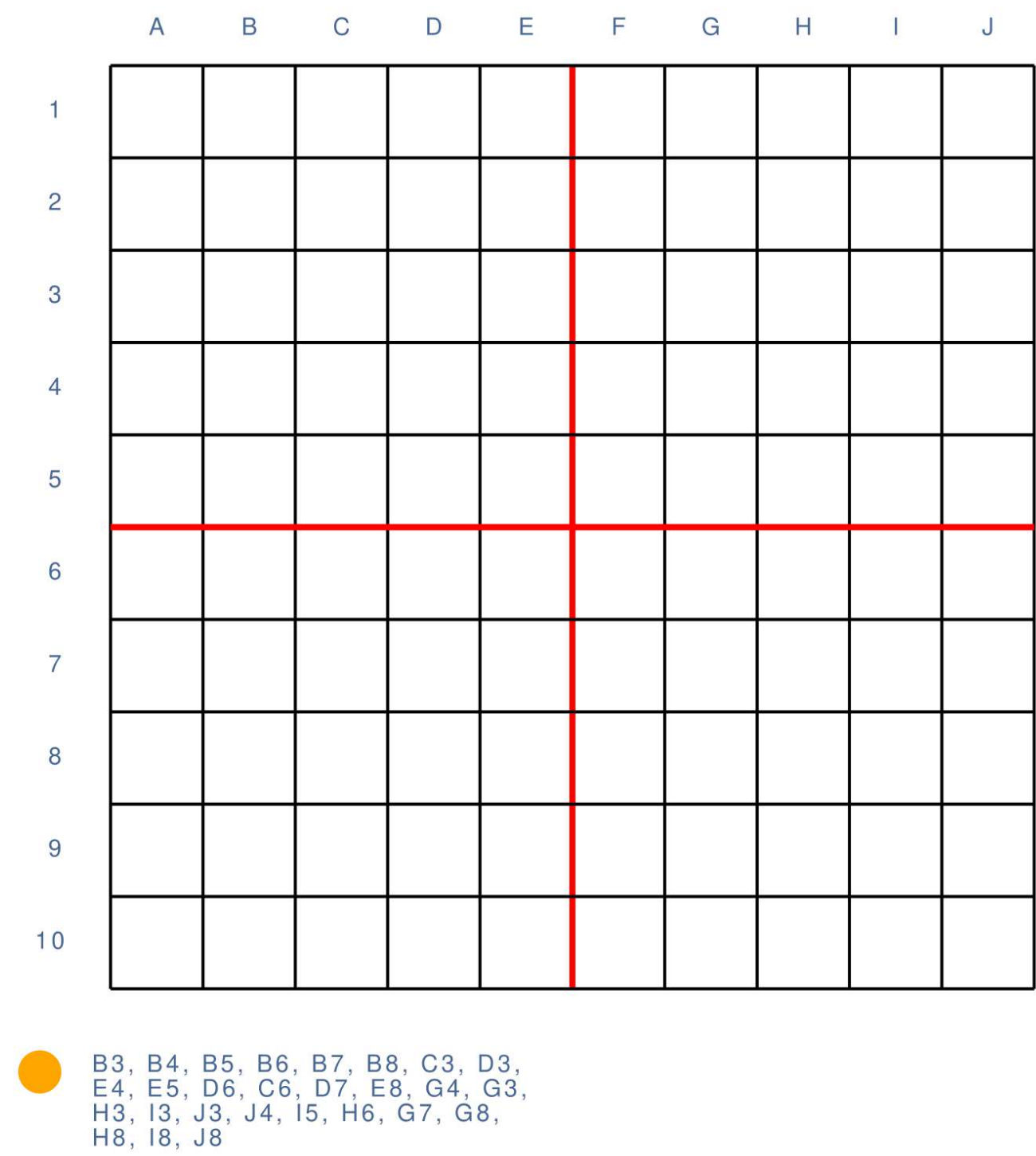


Źródło: opracowanie własne A. Świć.

Rysunek 5. Materiał pomocniczy „RZ współrzędne”



www.kodowanienadywanie.pl

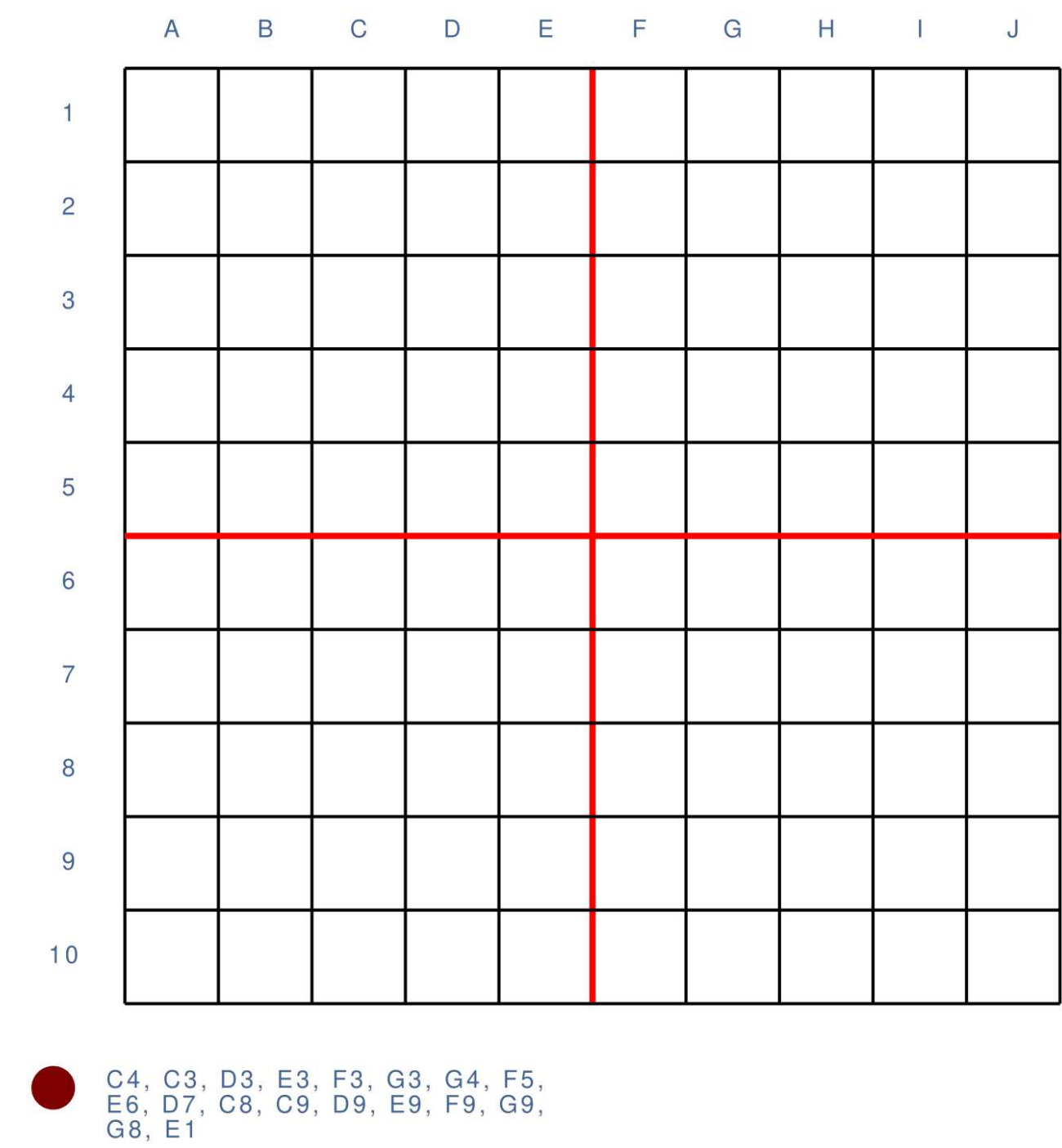


Źródło: opracowanie własne A. Świć.

Rysunek 6. Materiał pomocniczy „Ż współrzędne”

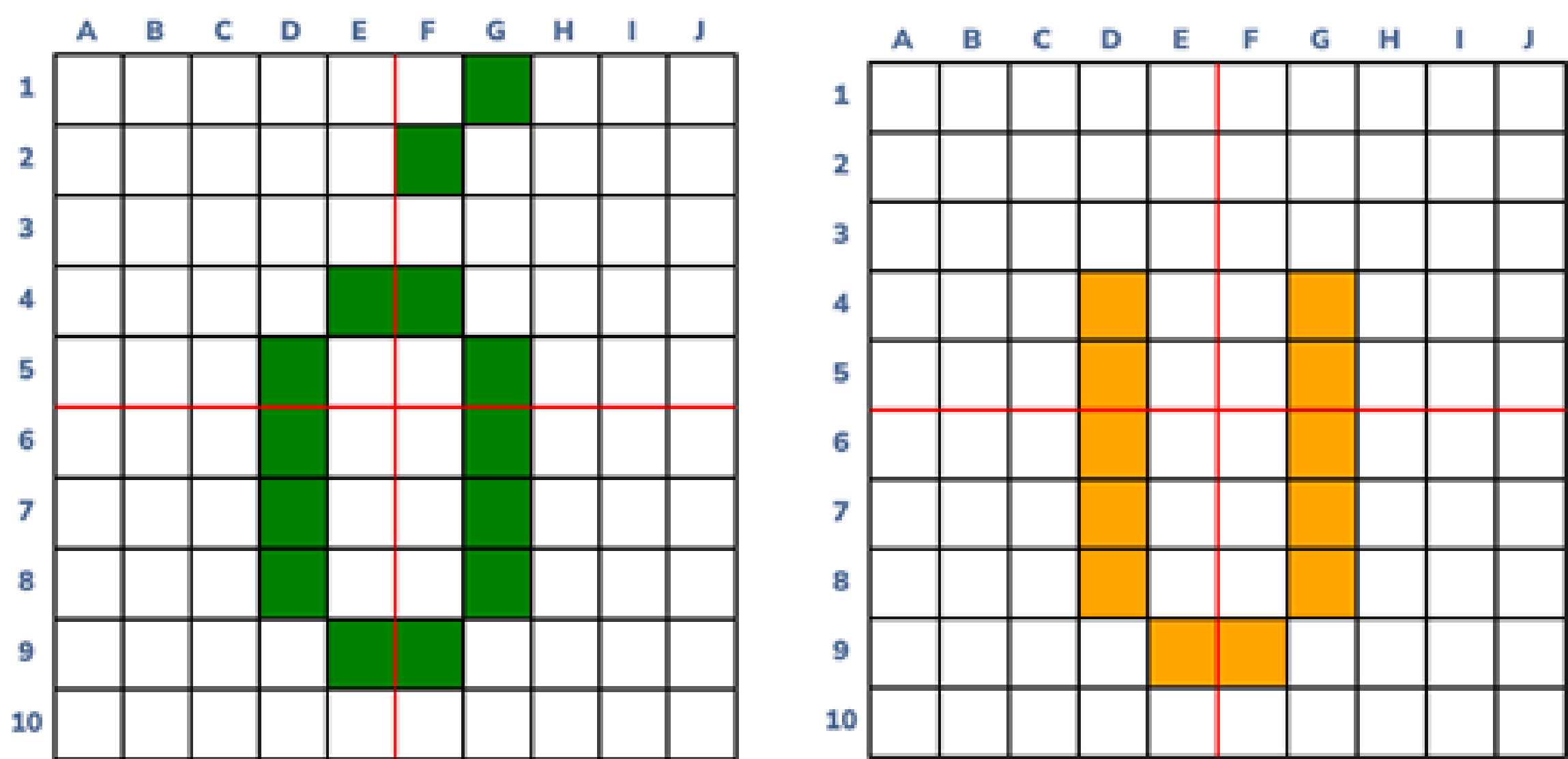


www.kodowanienadywanie.pl



Źródło: opracowanie własne A. Świć.

Rysunek 7. Przykładowe maty po dekodowaniu



Źródło: opracowanie własne A. Świć.

Prowadzący zajęcia pyta uczniów, co pojawiło się na macie do kodowania i w związku z tym będą się zajmować na zajęciach. Uczniowie tworzą cel zajęć: utrwalanie zasad pisowni wyrazów z trudnościami ortograficznymi.

Część właściwa:

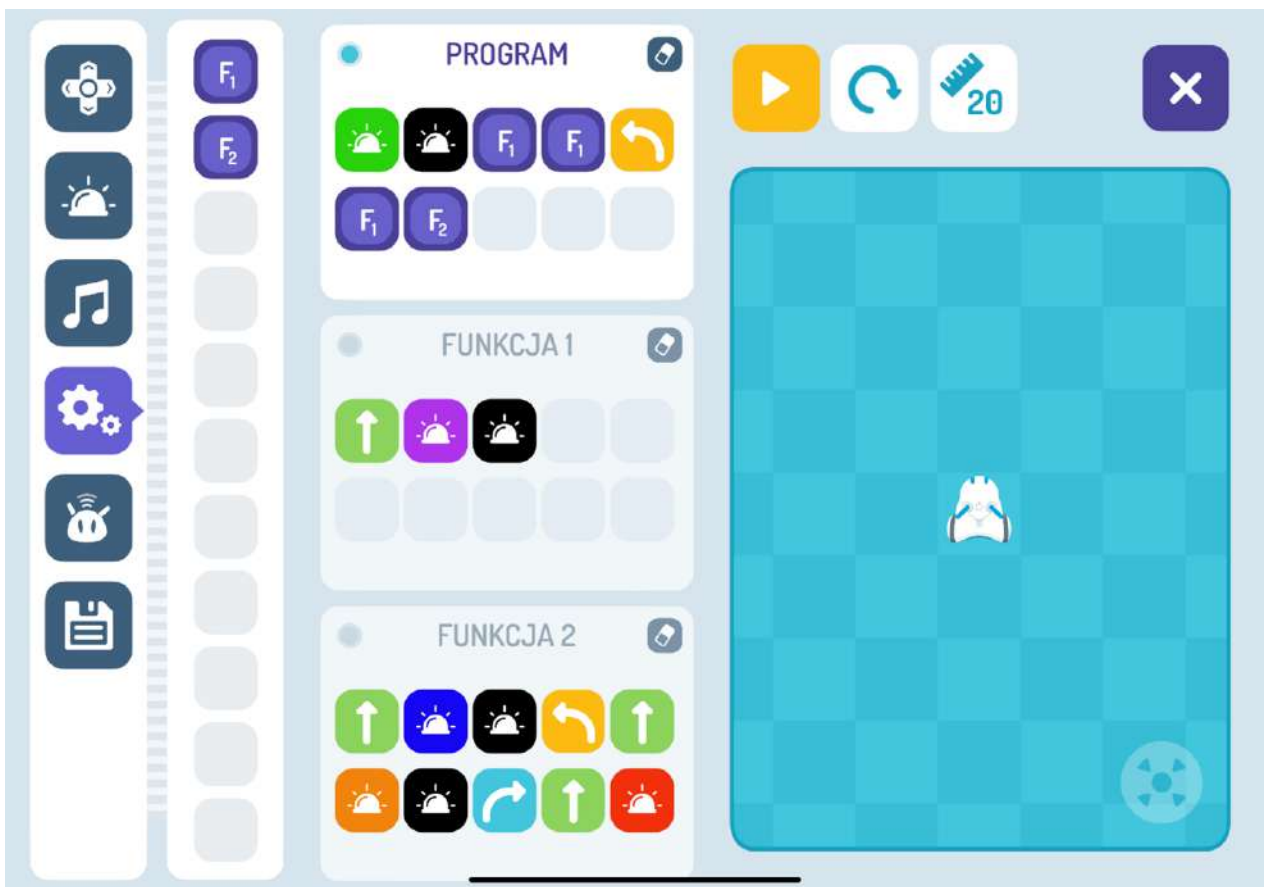
Uczniowie rozkładają na losowych polach na macie do kodowania krążki z wyrazami, które mają luki w miejscach trudności ortograficznych (pomoc „Ortografia”).

Dzieci ustalają kolory, które przypiszą do „Ó”, „U”, „Rz”, „Ż”, „CH”, „H”. Te same kolory zostaną wykorzystane przy programowaniu robotów, więc warto je zapisać na kartce, żeby mieć przez całą zabawę legendę łączącą kolor z wybraną trudnością ortograficzną.

Nauczyciel dzieli uczniów na grupy. Każda grupa wybiera siedem wyrazów, przez które przeprowadzi robota. Robot przejeżdżając przez dany wyraz będzie musiał zmienić na właściwą barwę kolor czułek i oczu (barwa zależna od pisowni).

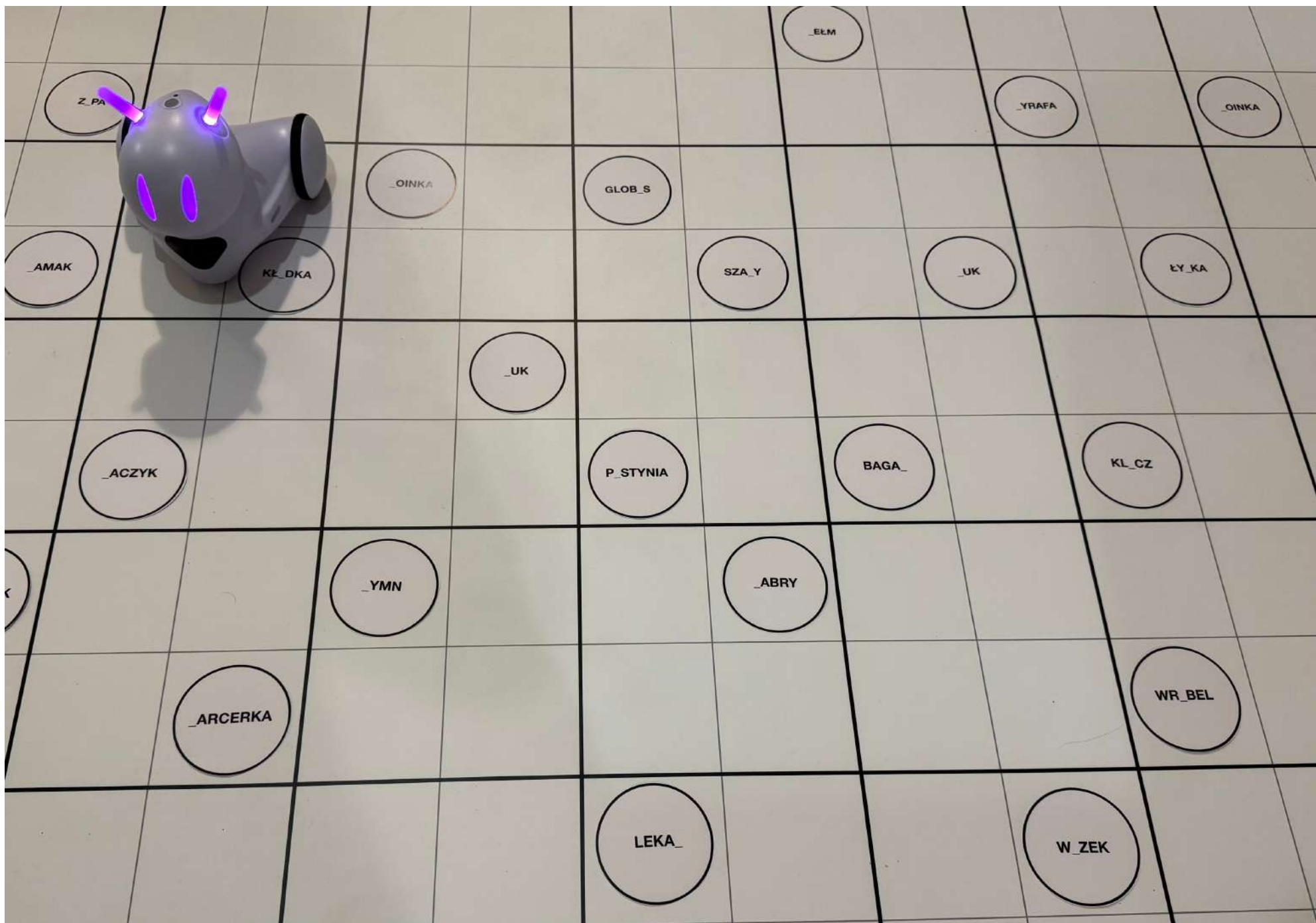
Każdy zespół programuje swojego robota (zespoły mogą wybrać dowolny interfejs, który wykorzystają do stworzenia programu – Photon Blocks lub Photon Badge).

Rysunek 8. Przykładowy interfejs Photon Badge



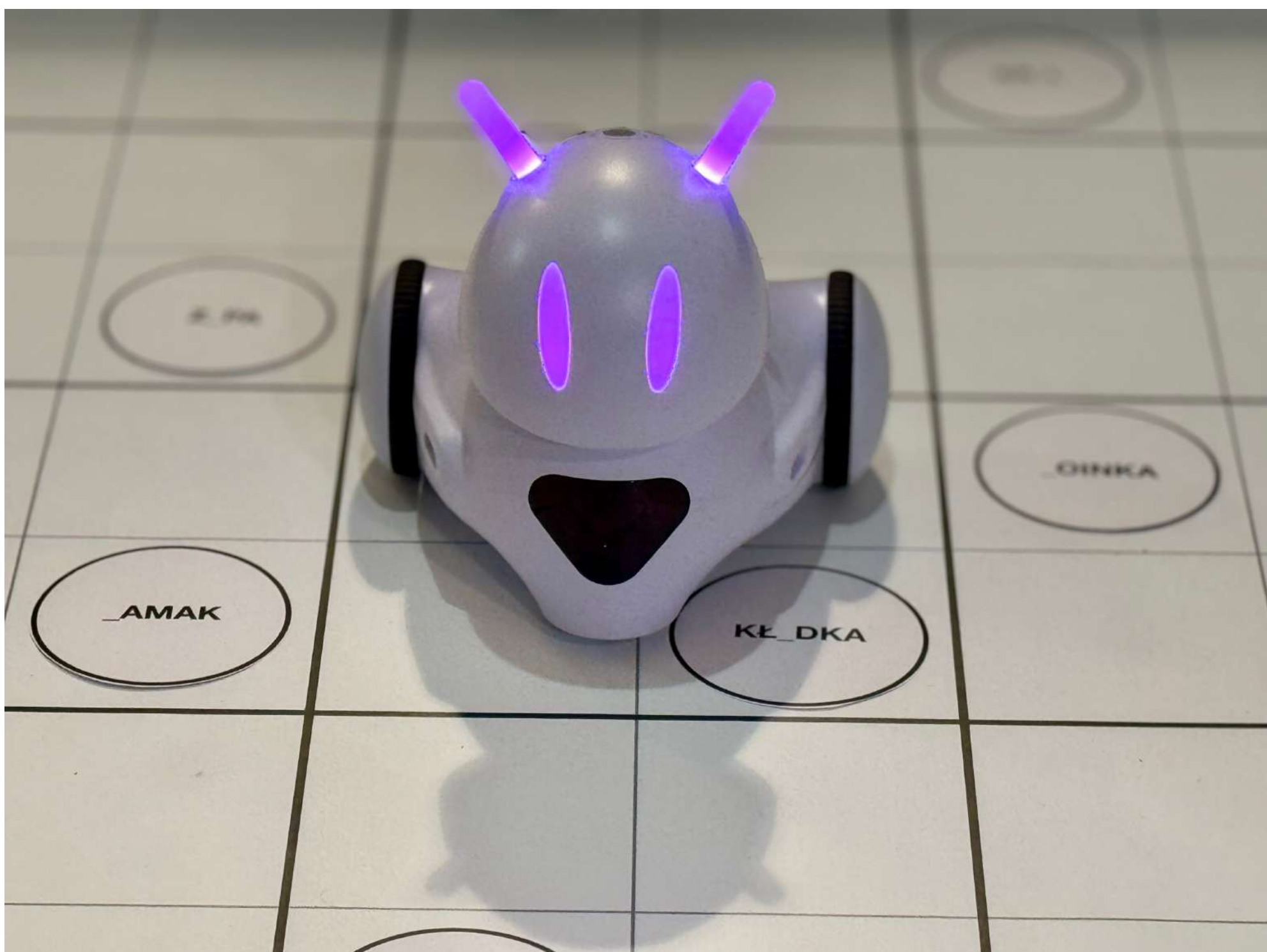
Źródło: opracowanie własne A. Świć.

Rysunek 9. Zaprogramowany robot



Źródło: opracowanie własne A. Świć.

Rysunek 10. Zaprogramowany robot wskazujący trudność ortograficzną za pomocą koloru



Źródło: opracowanie własne A. Świć.

W następnym kroku grupy po kolei startują swoje Photony, a wszyscy sprawdzają, czy wybrana przez zespół barwa na poszczególnych krążkach odpowiada pisowni wyrazów, które się na tych krążkach znajdują.

Prowadzący zajęcia proponuje uczniom zagranie w ortograficzną grę planszową. Do ułożenia planszy zostaną wykorzystane te same krążki, które były użyte przy programowaniu robotów. Planszę tworzymy układając krążki na macie do kodowania (można wybrać 36 krążków i ułożyć planszę 6X6).

Nauczyciel wyjaśnia zasady gry. Każdy zespół otrzymuje pionki (w roli pionków dobrze sprawdzają się kubki. Potrzebna będzie też specjalna kostka do gry (załącznik „Ortografia kostka”). Pierwszy zespół rzuca kostką, a następnie stawia swój kubek na jednym wybranym wyrazie, który pisze się tak, jak wskazała kostka. Kolejne zespoły postępują analogicznie. Jeśli na planszy nie ma już wyrazu pasującego do wyniku rzutu kostką, to zespół traci kolejkę. Wygrywa ten zespół, który jako pierwszy ustawi w jednej linii cztery swoje pionki. Linia może być pozioma, pionowa lub ukośna.

Zakończenie zajęć:

Na zakończenie zajęć każdy zespół tworzy po jednym zdaniu z pięcioma wybranymi wyrazami, które pojawiły się w grze planszowej.

Nauczyciel prosi uczniów aby zbudowali wieżę z kubków – jeśli znali pisownię wszystkich wyrazów z lekcji – wieża powinna być zbudowana z 3 kubków; jeśli znali większość wyrazów – z 2 kubków, jeśli uważają, że jeszcze muszą popracować nad znajomością zasad ortograficznych – z 1 kubka.



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



C4, C3, D3, E3, F3, G3, G4, F5,
E6, D7, C8, C9, D9, E9, F9, G9,
G8, E1

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



D4, D5, D6, D7, D8, E9, F9, G8,
G7, G6, G5, G4



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



B3, B4, B5, B6, B7, B8, C3, D3,
E4, E5, D6, C6, D7, E8, G4, G3,
H3, I3, J3, J4, I5, H6, G7, G8,
H8, I8, J8



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



E4, F4, D5, D6, D7, D8, E9, F9,
G8, G7, G6, G5, F2, G1



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

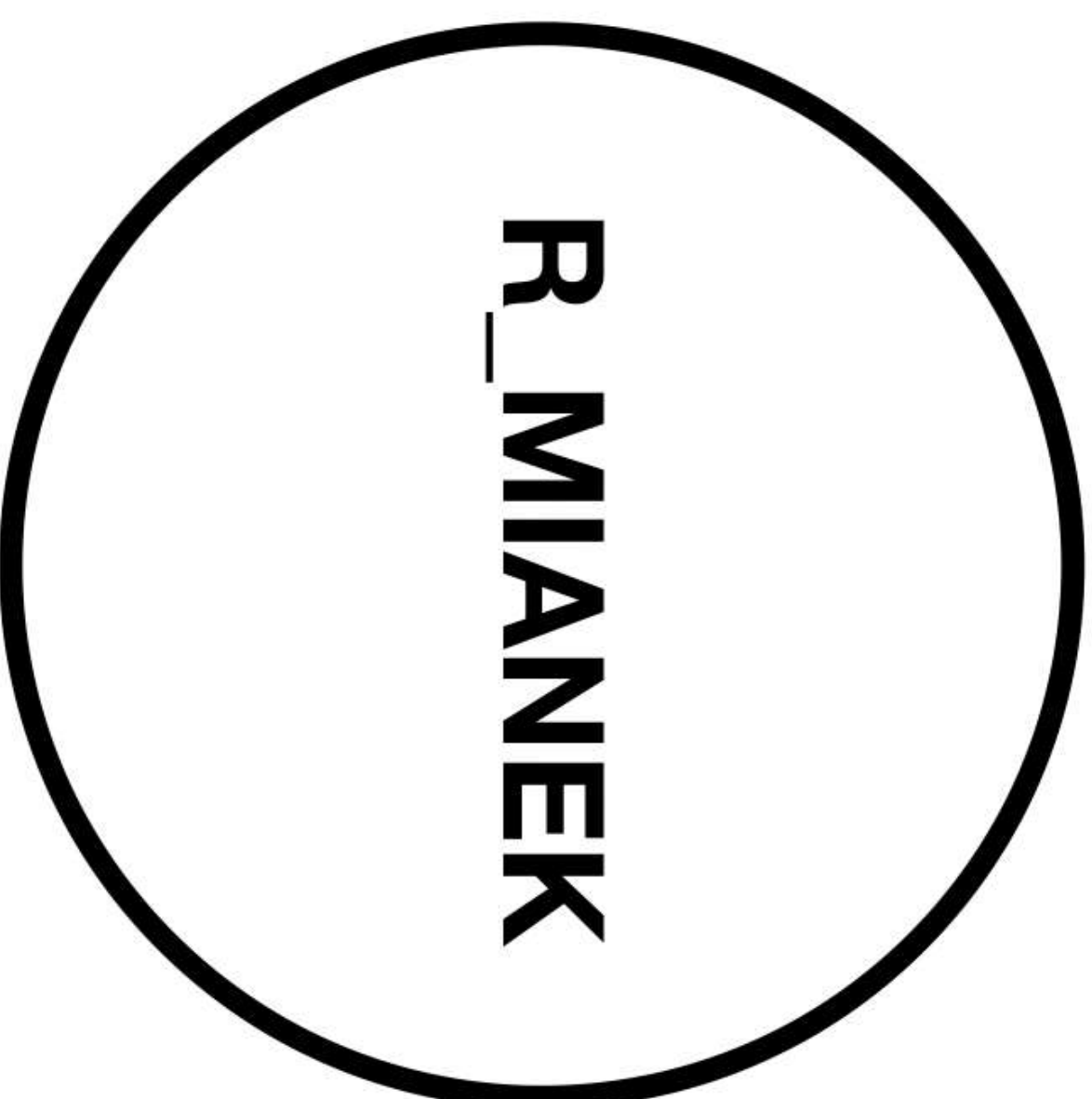
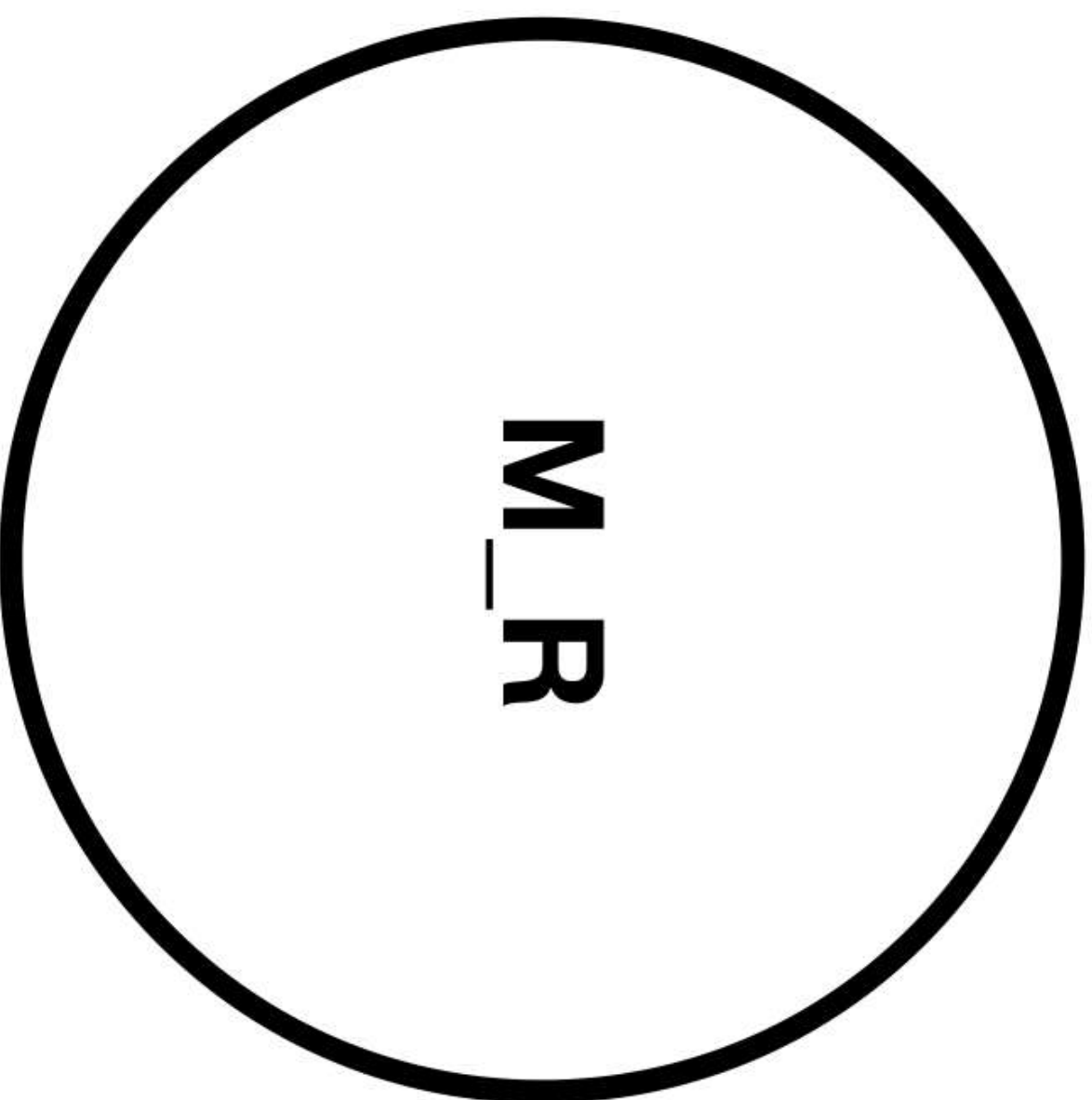
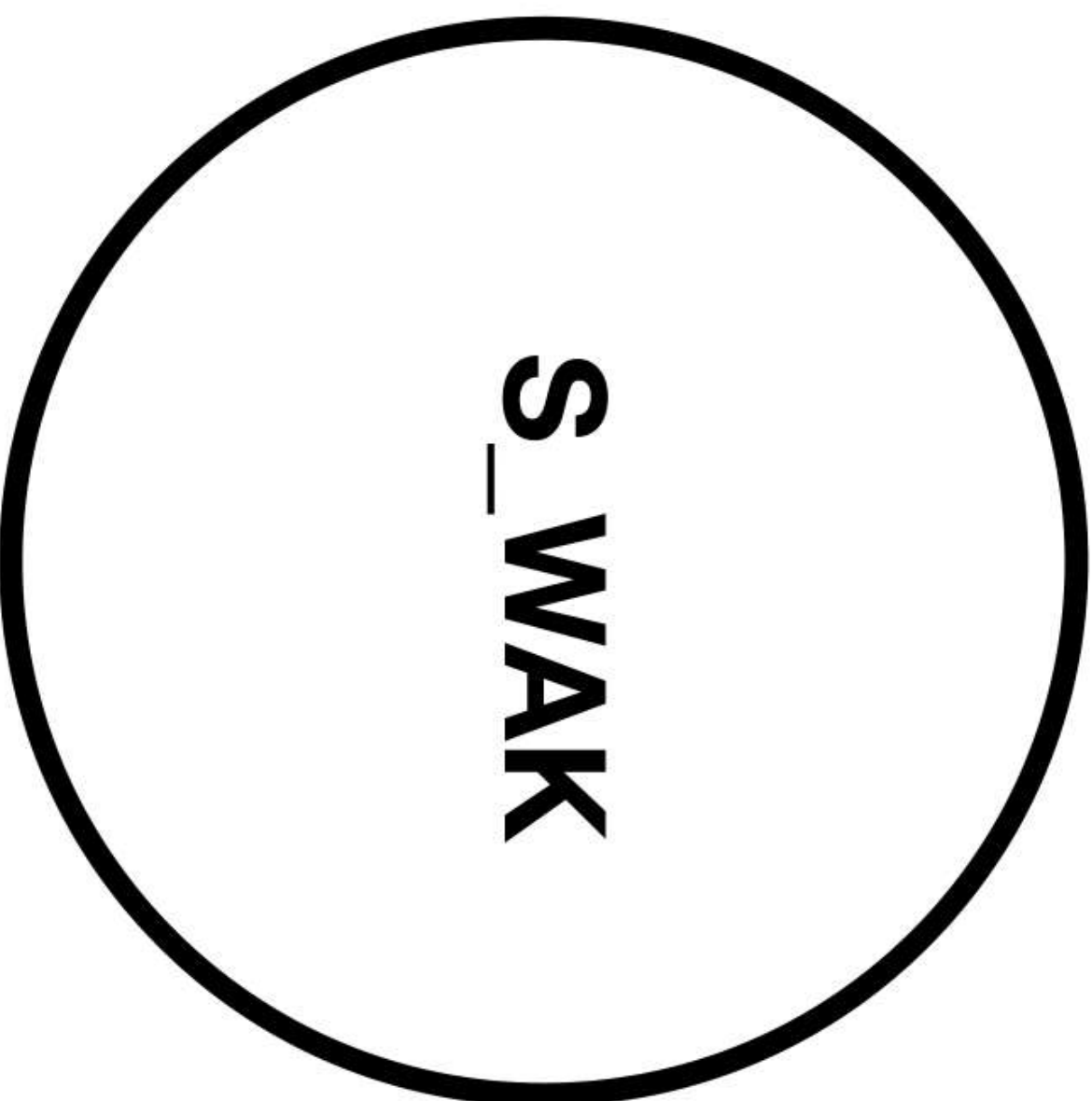
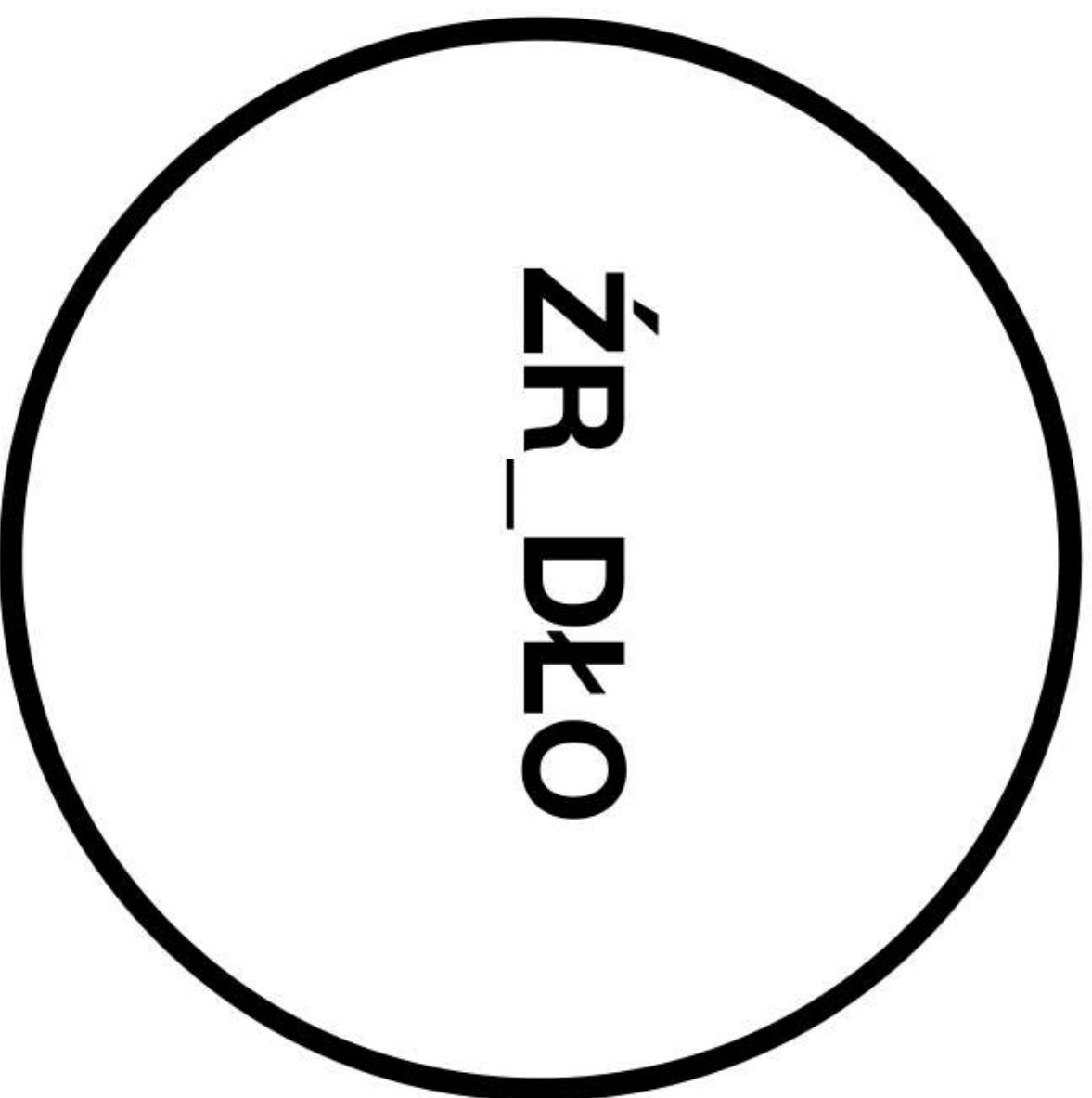


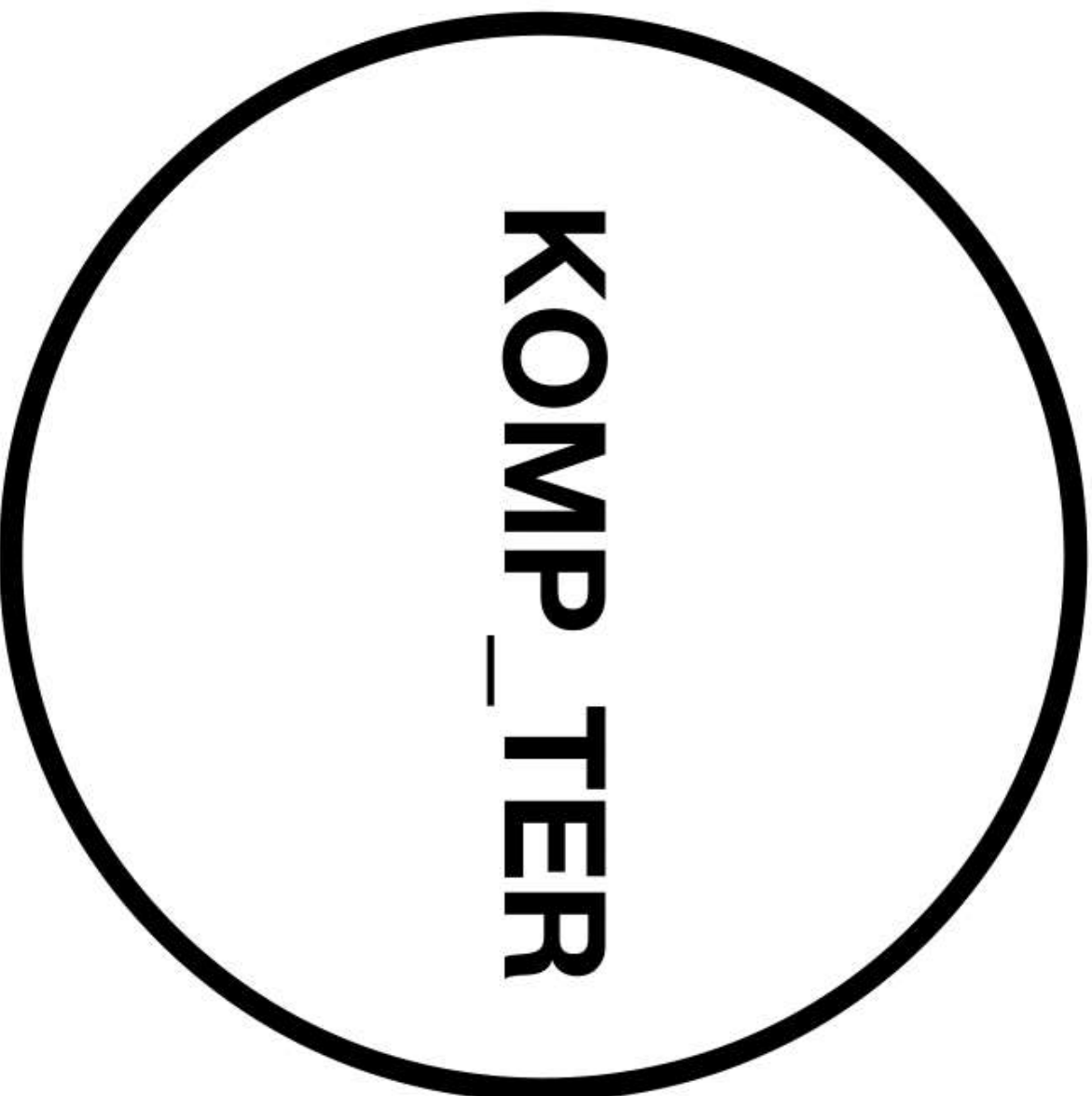
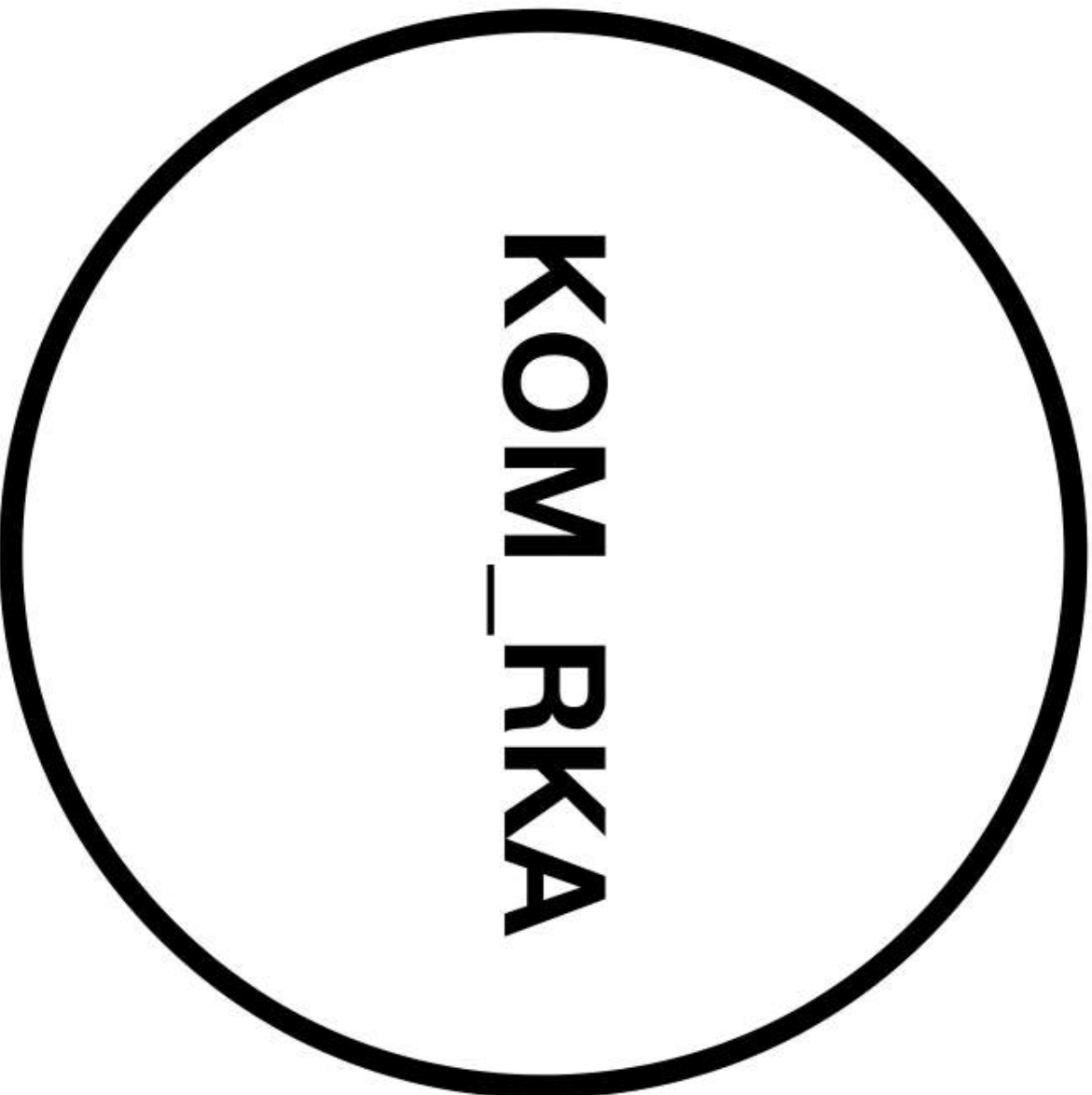
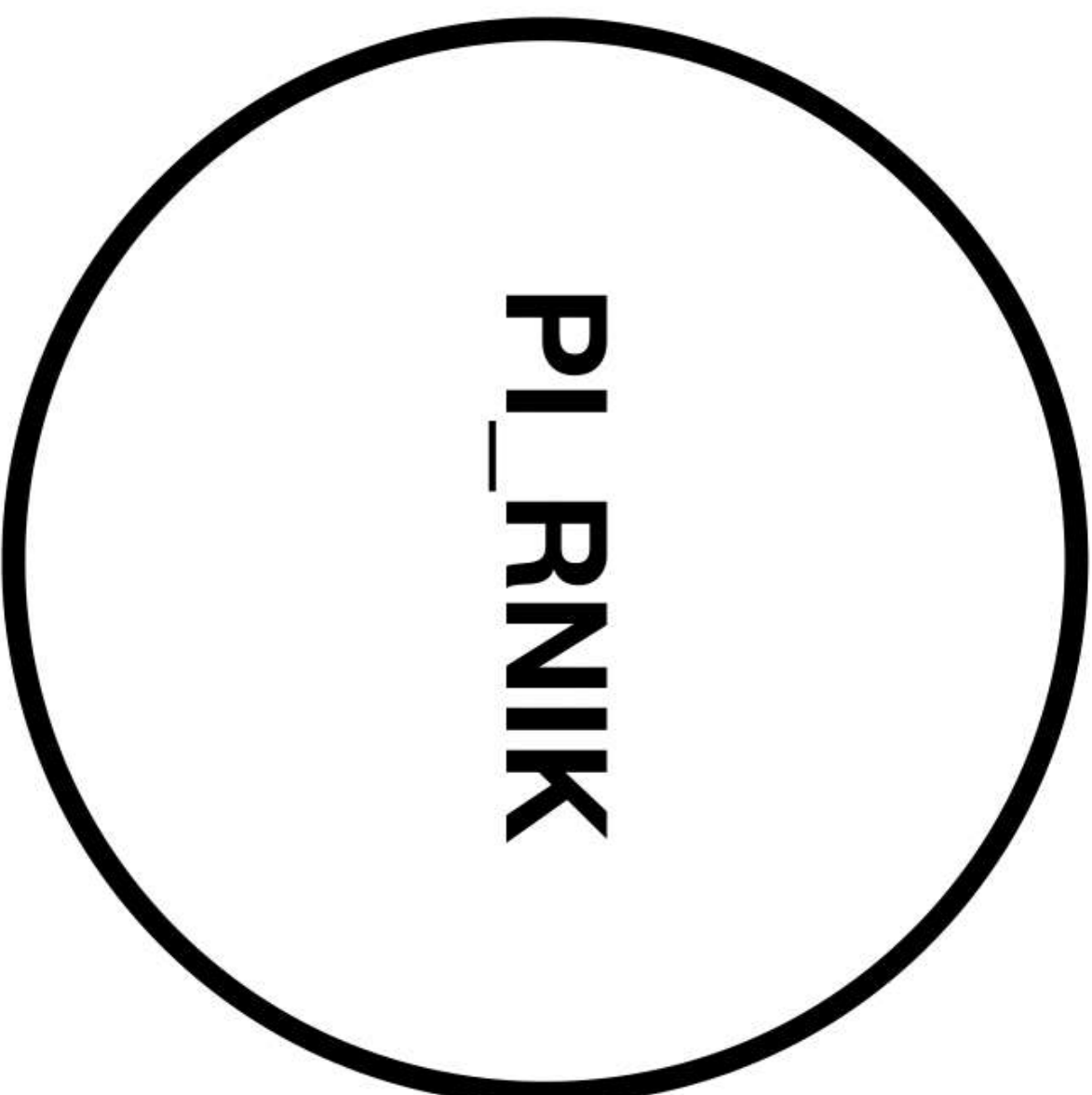
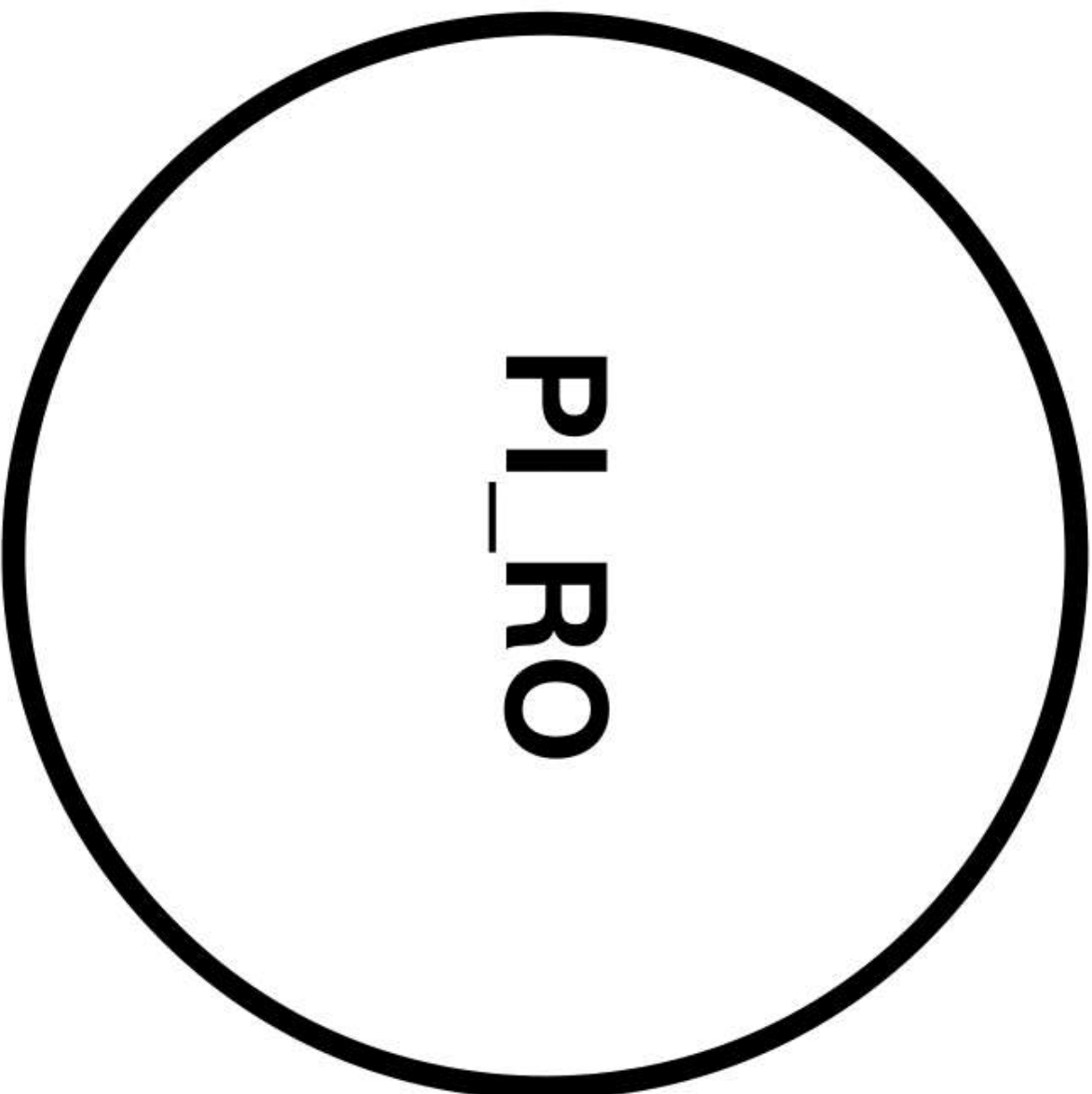
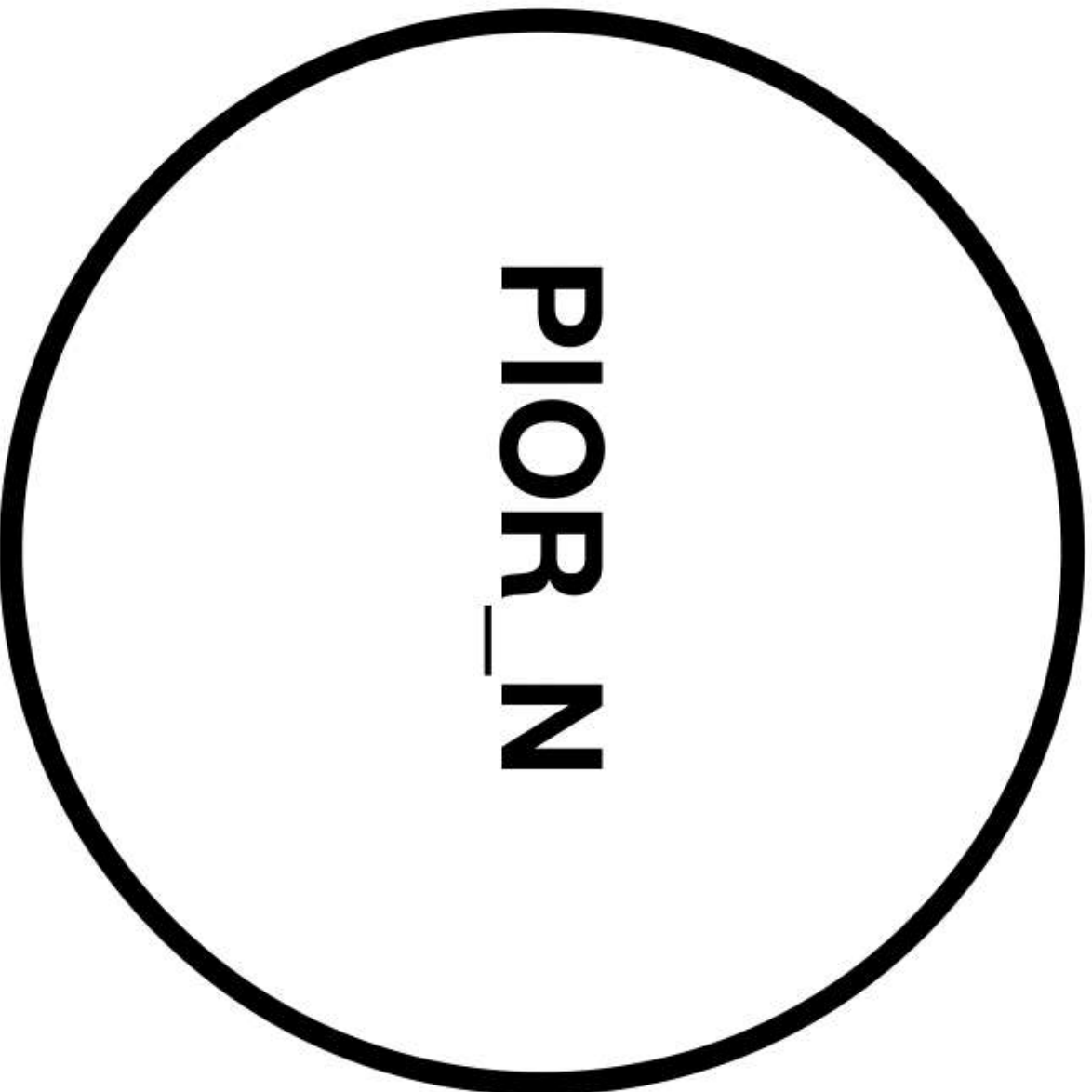
D4, C3, B3, A4, A5, A6, A7, A8,
B9, C9, D8, F3, F4, F5, F6, F7,
F8, F9, I3, I4, I5, I6, I7, I8, I9,
G6, H6

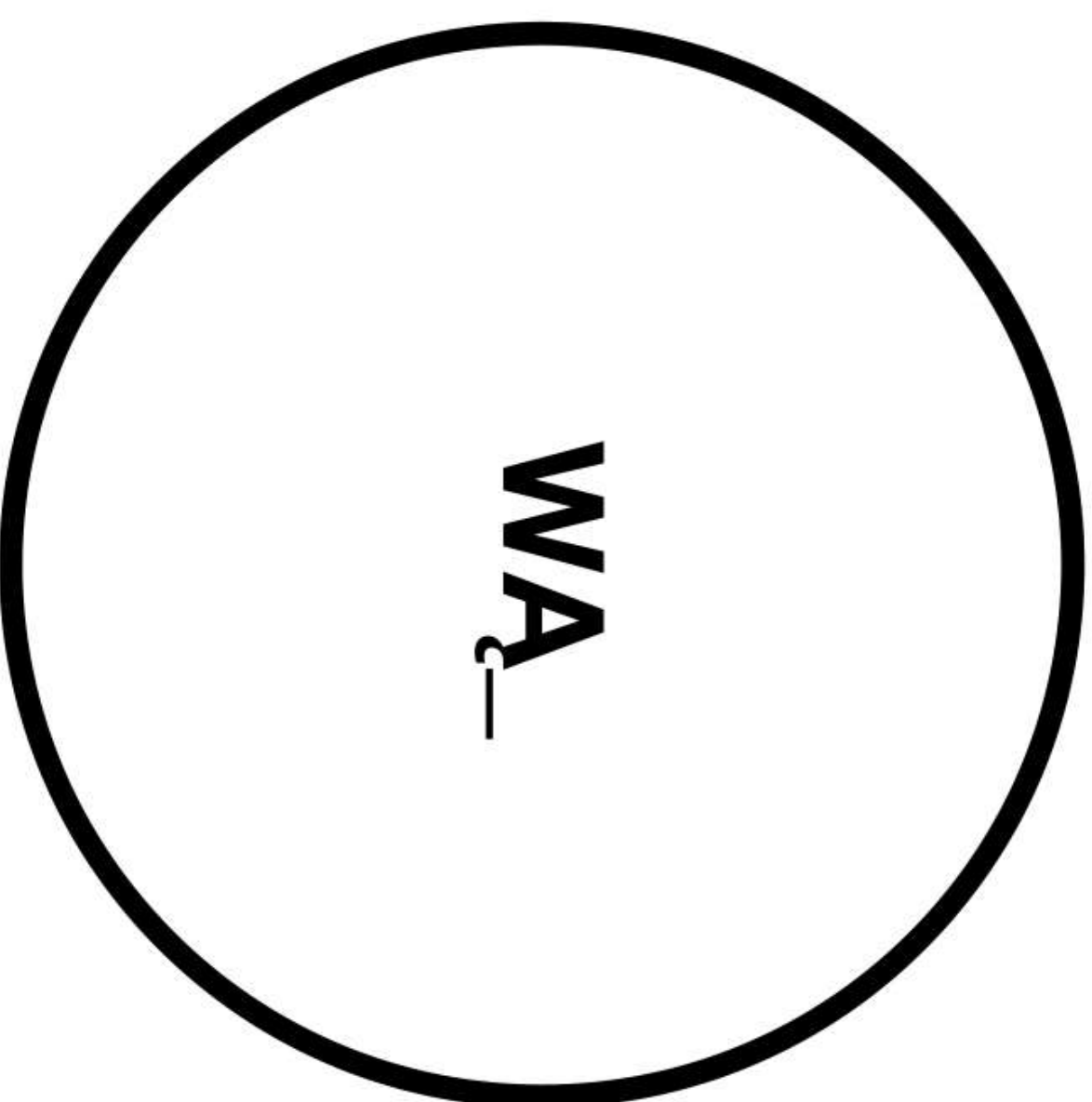
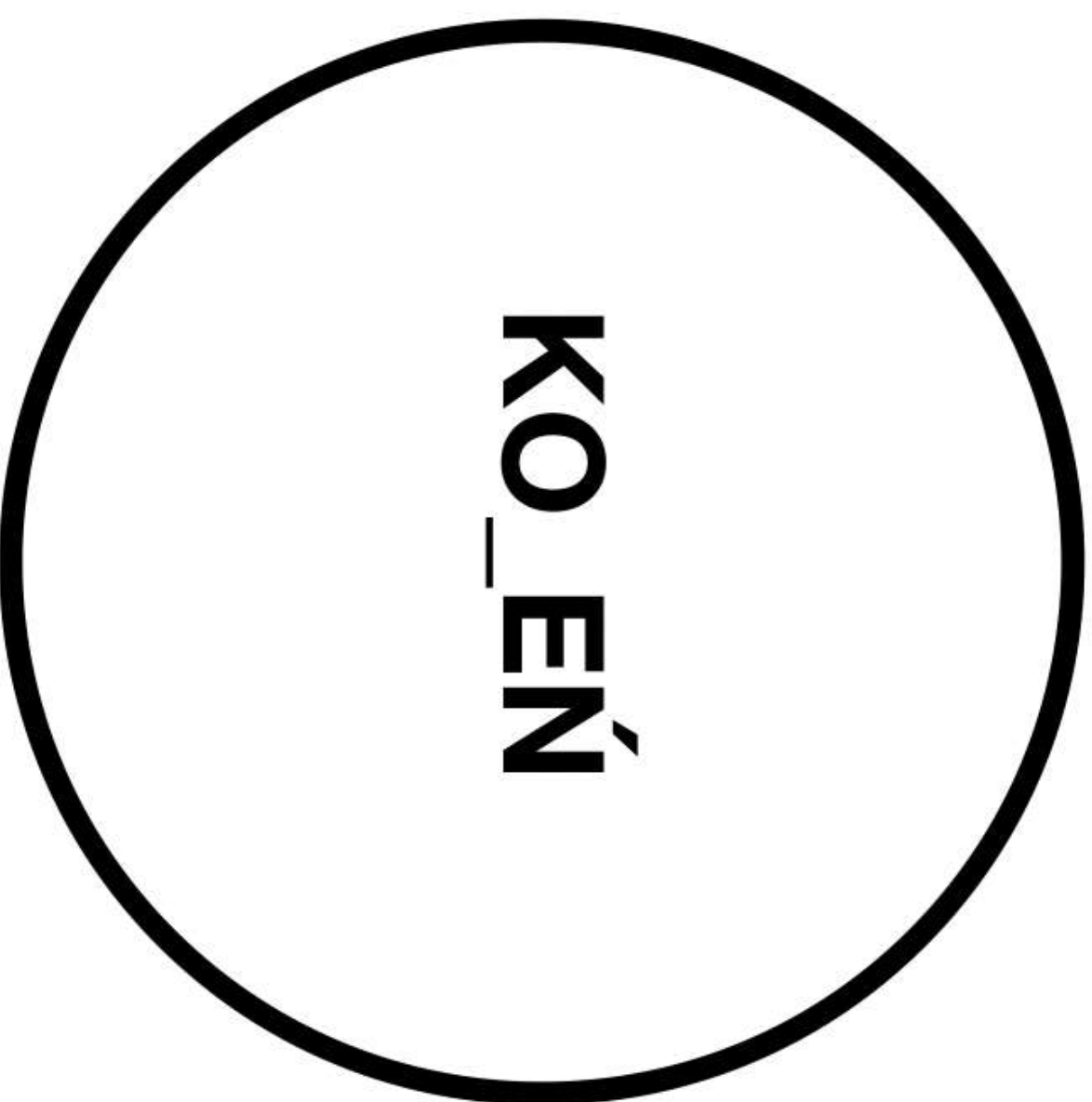
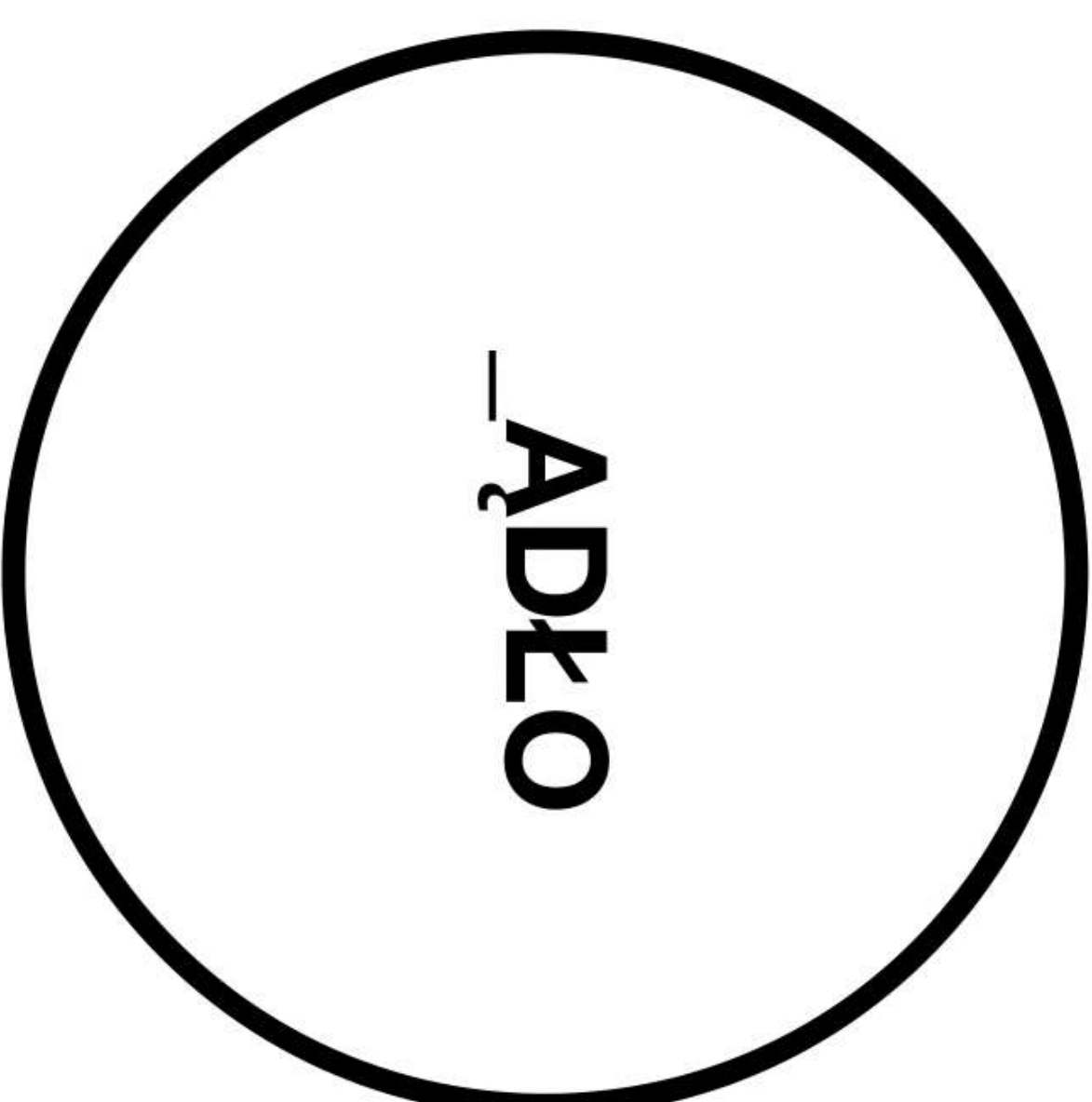
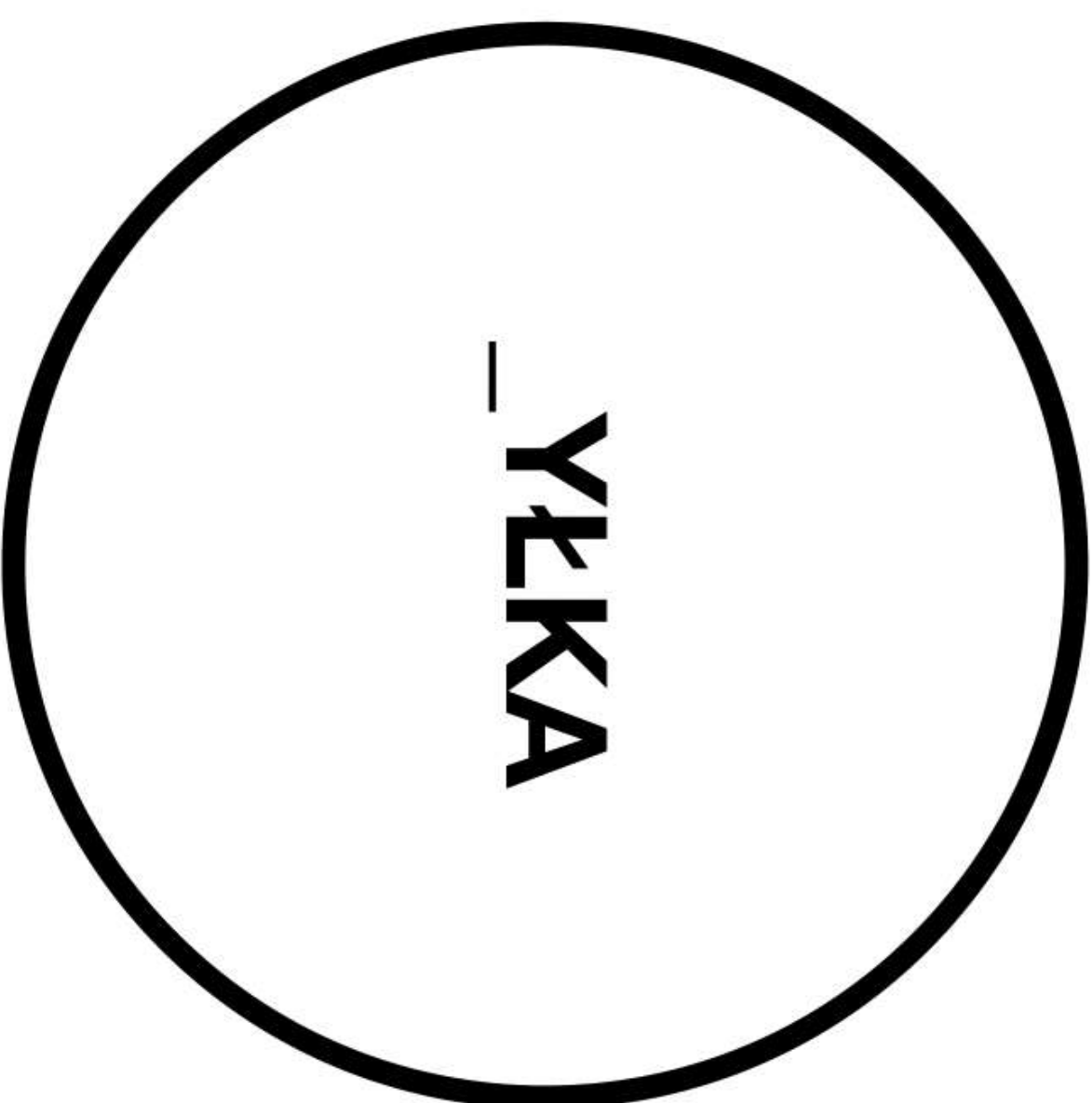
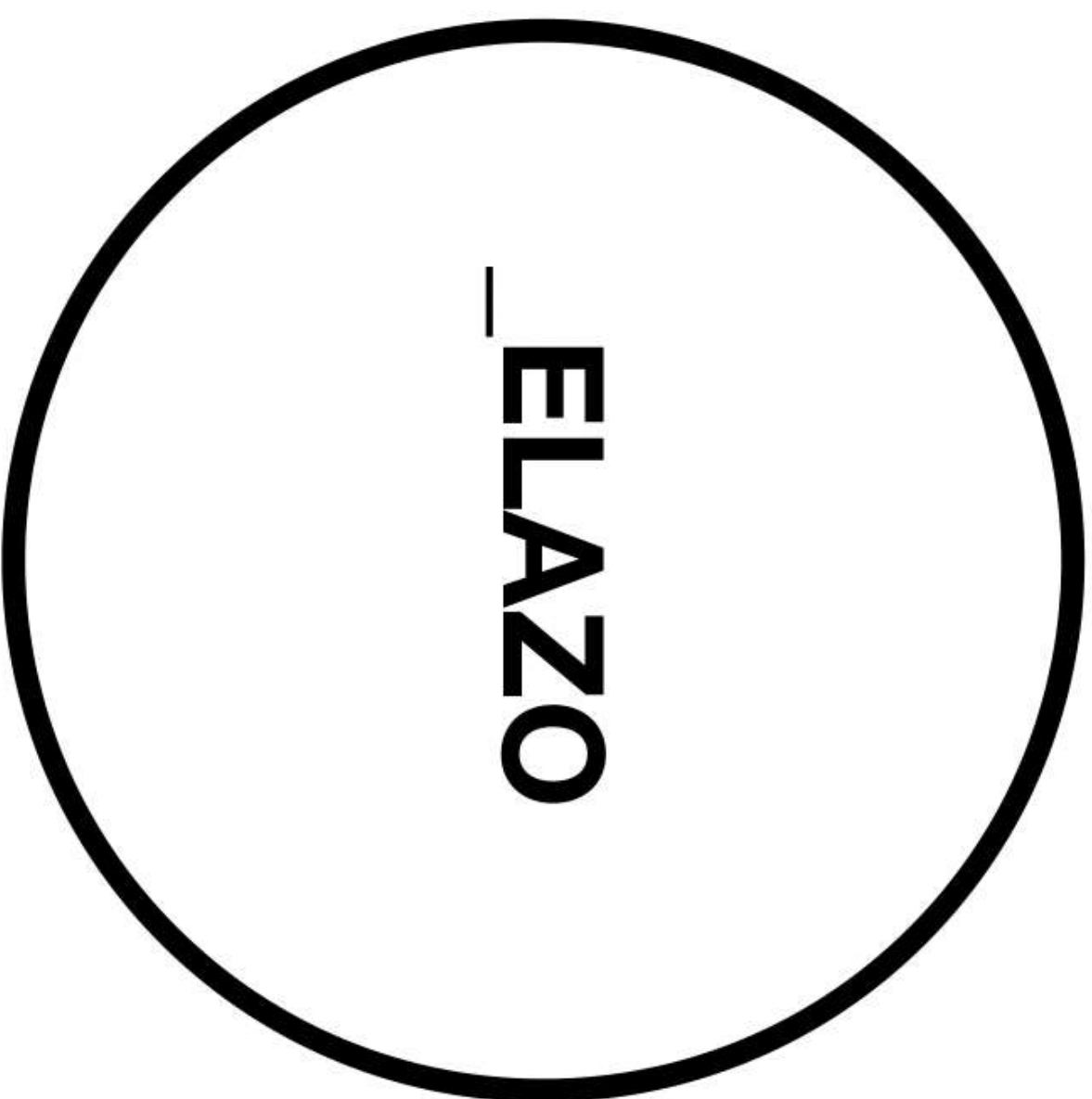
	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										



D4, D5, D6, D7, D8, D9, D3, H3,
H4, H5, H6, H7, H8, H9, E6, F6,
G6







ST_YKAWKA

WIE_A

JE_

KOLE_ANKA

K_ESKO

FILI_ANKA

S_ODY

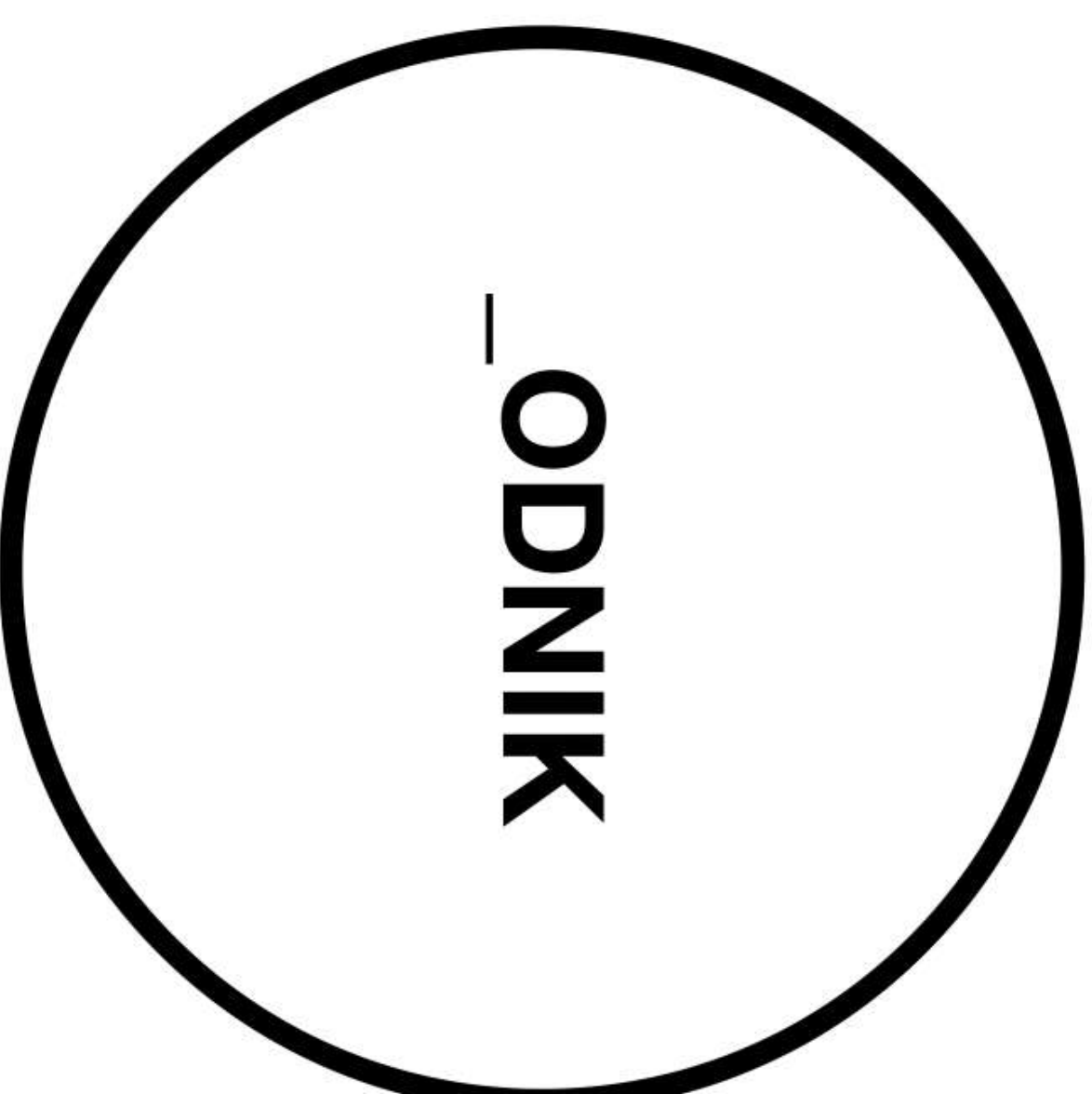
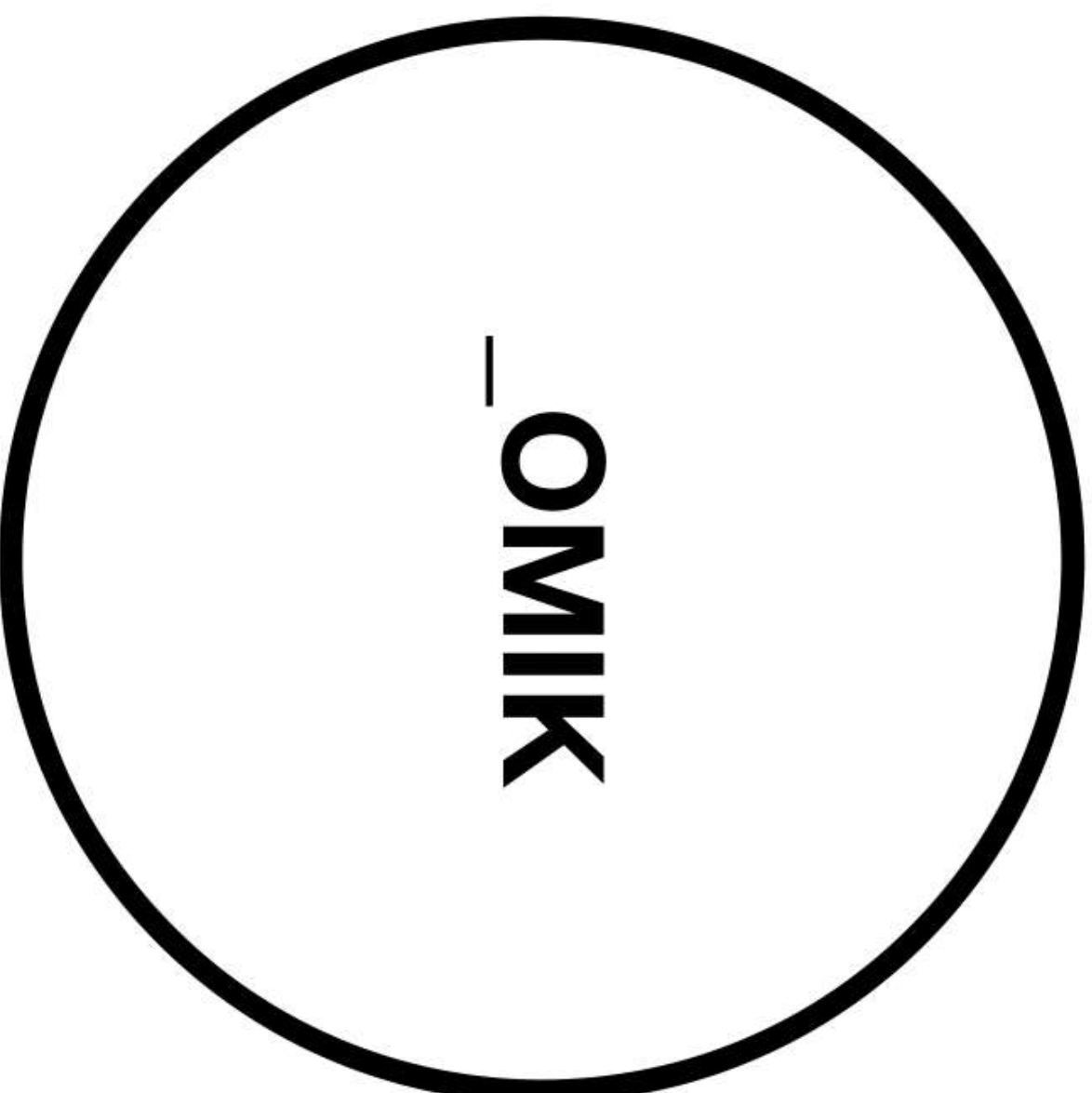
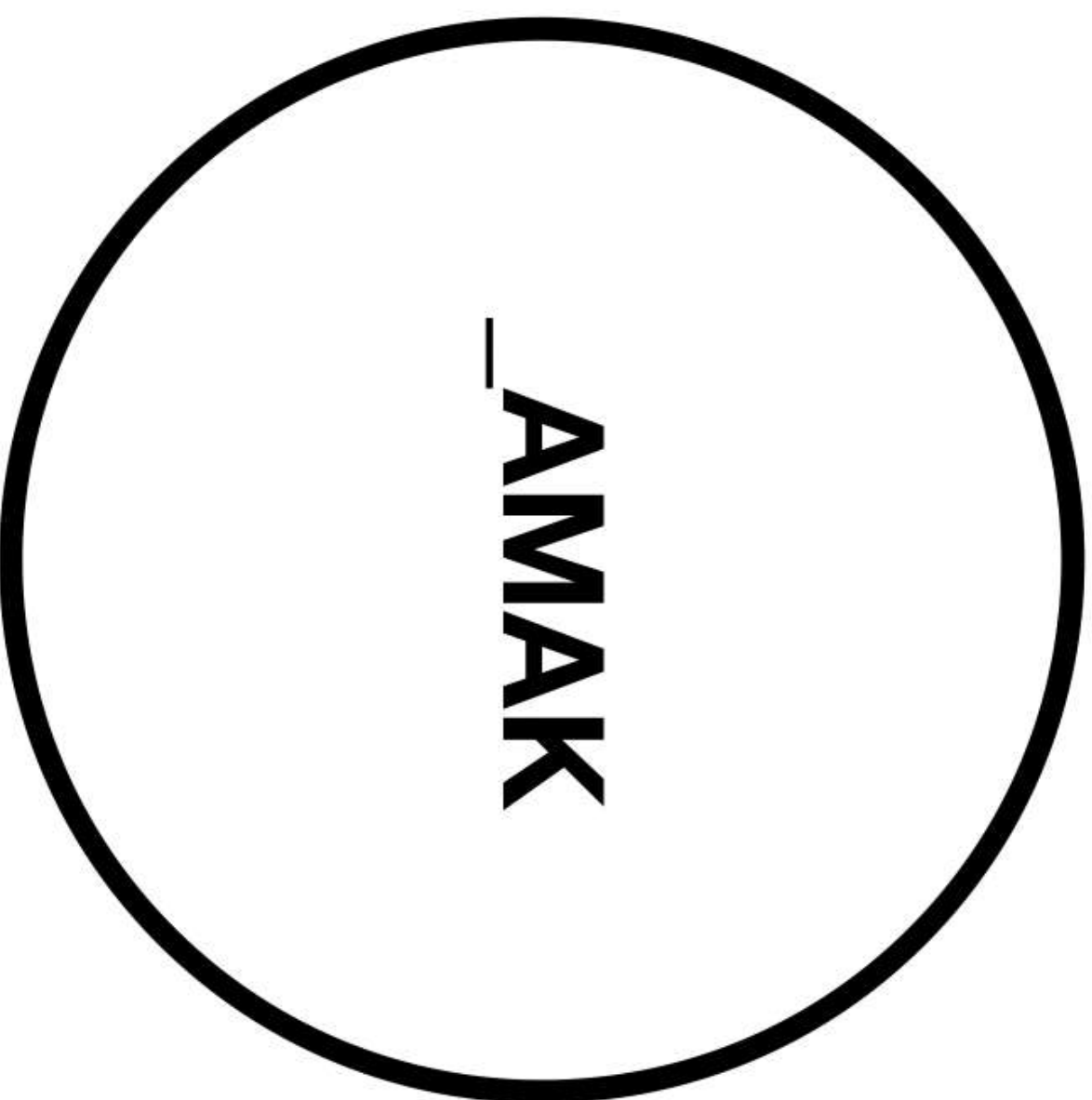
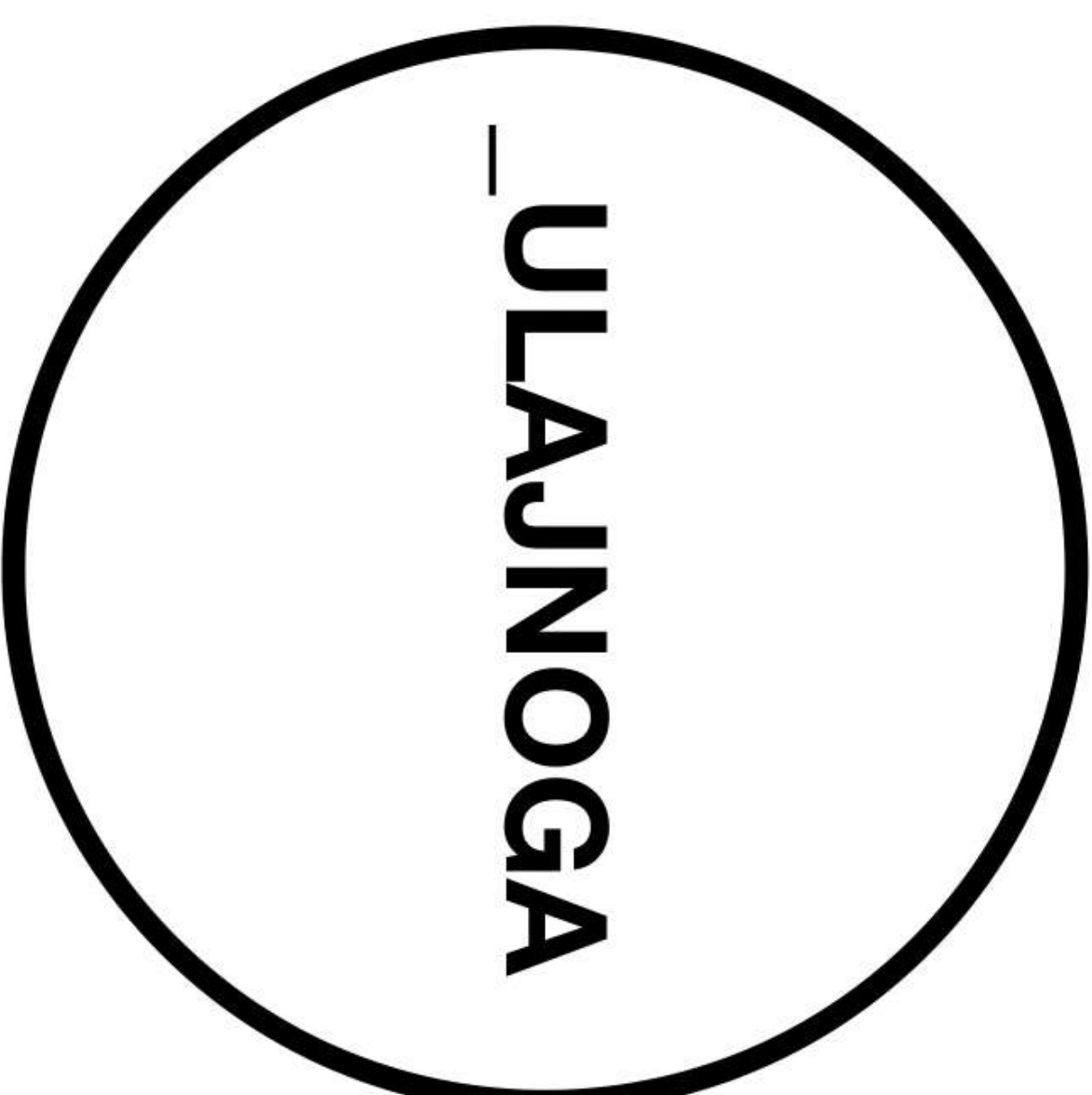
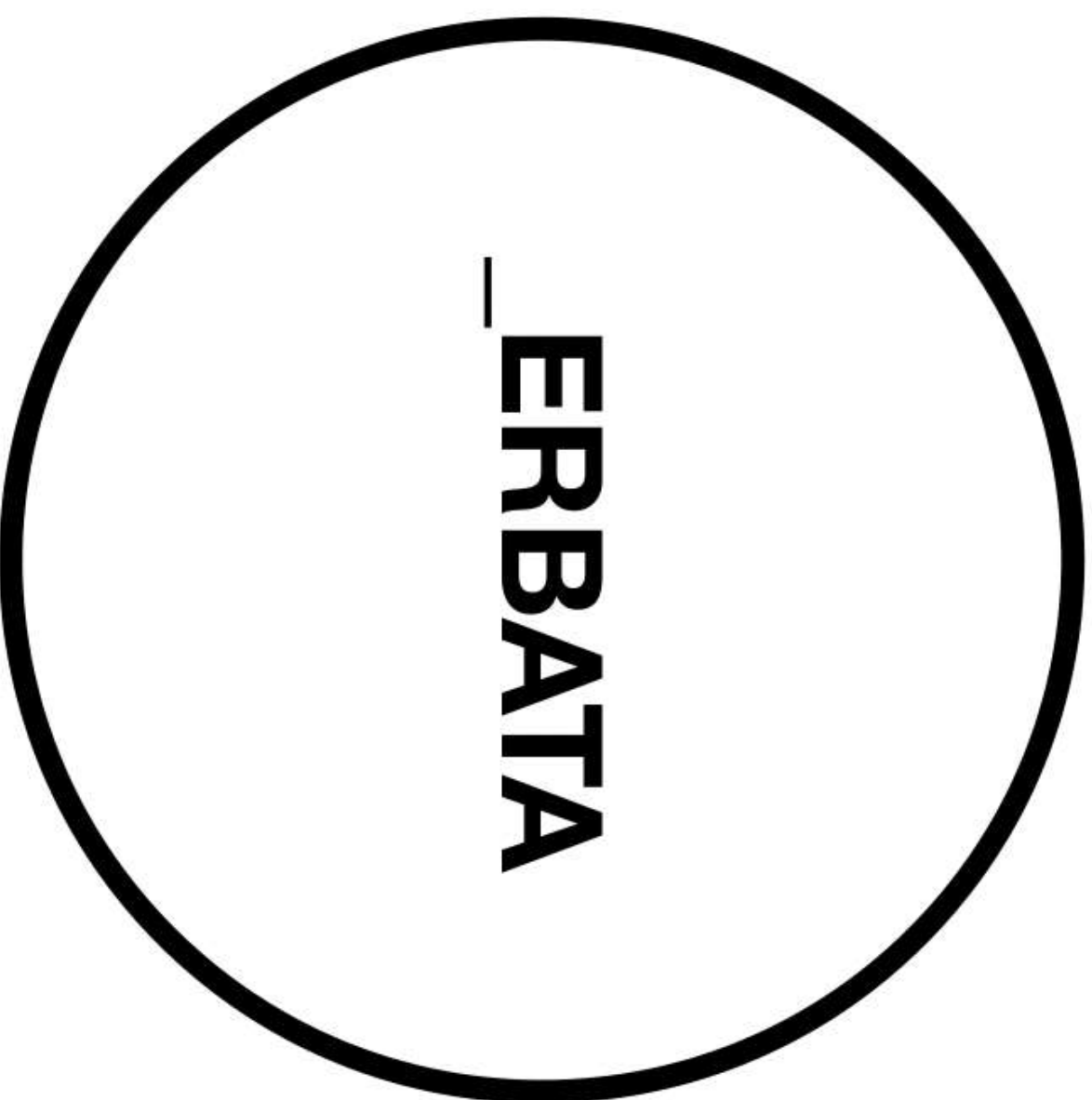
_IPOOTAM

_ARMONIA

_AKKA

_YDRAULIK

_ELIKOPTER



_ORĄGIEW

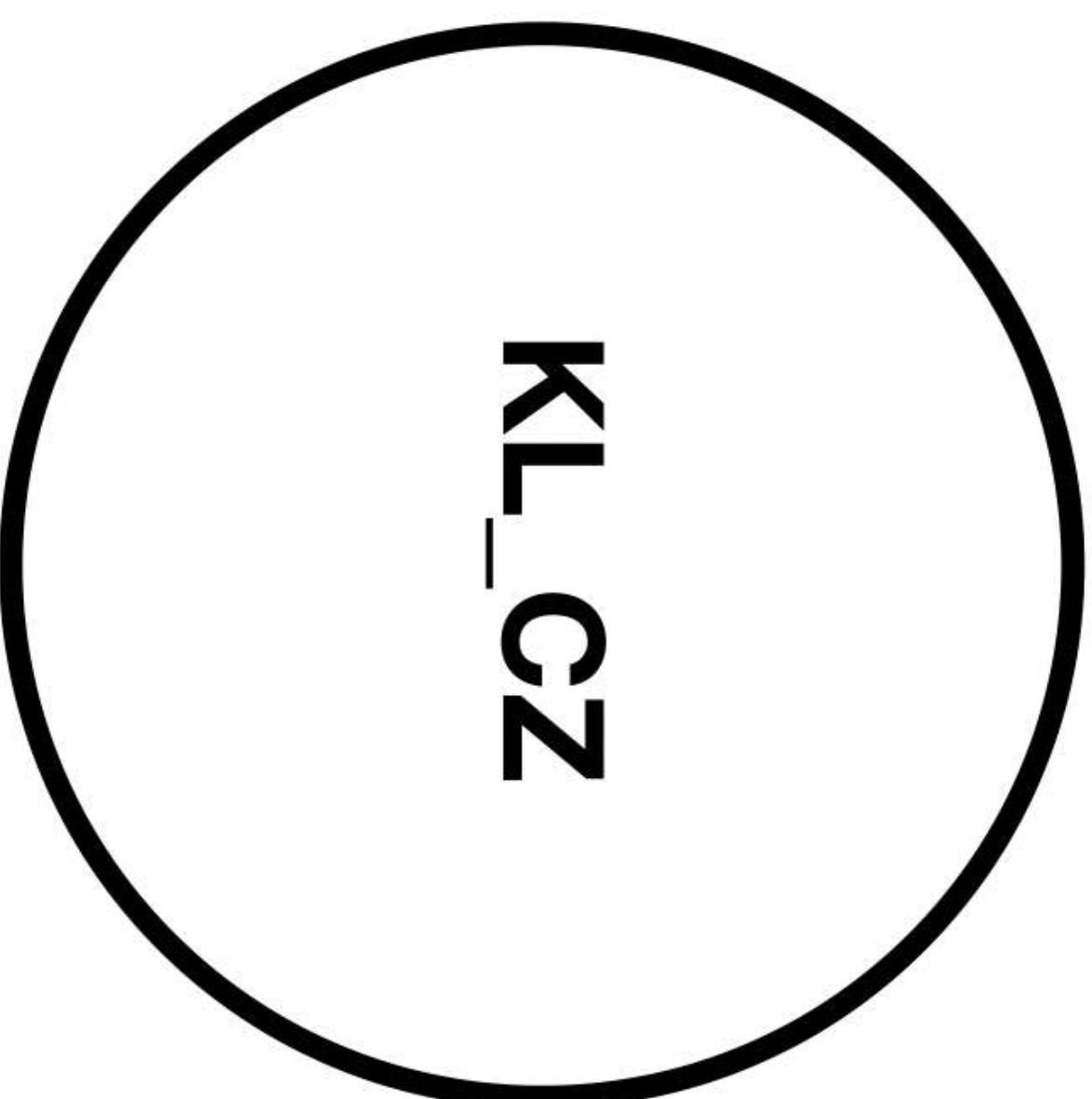
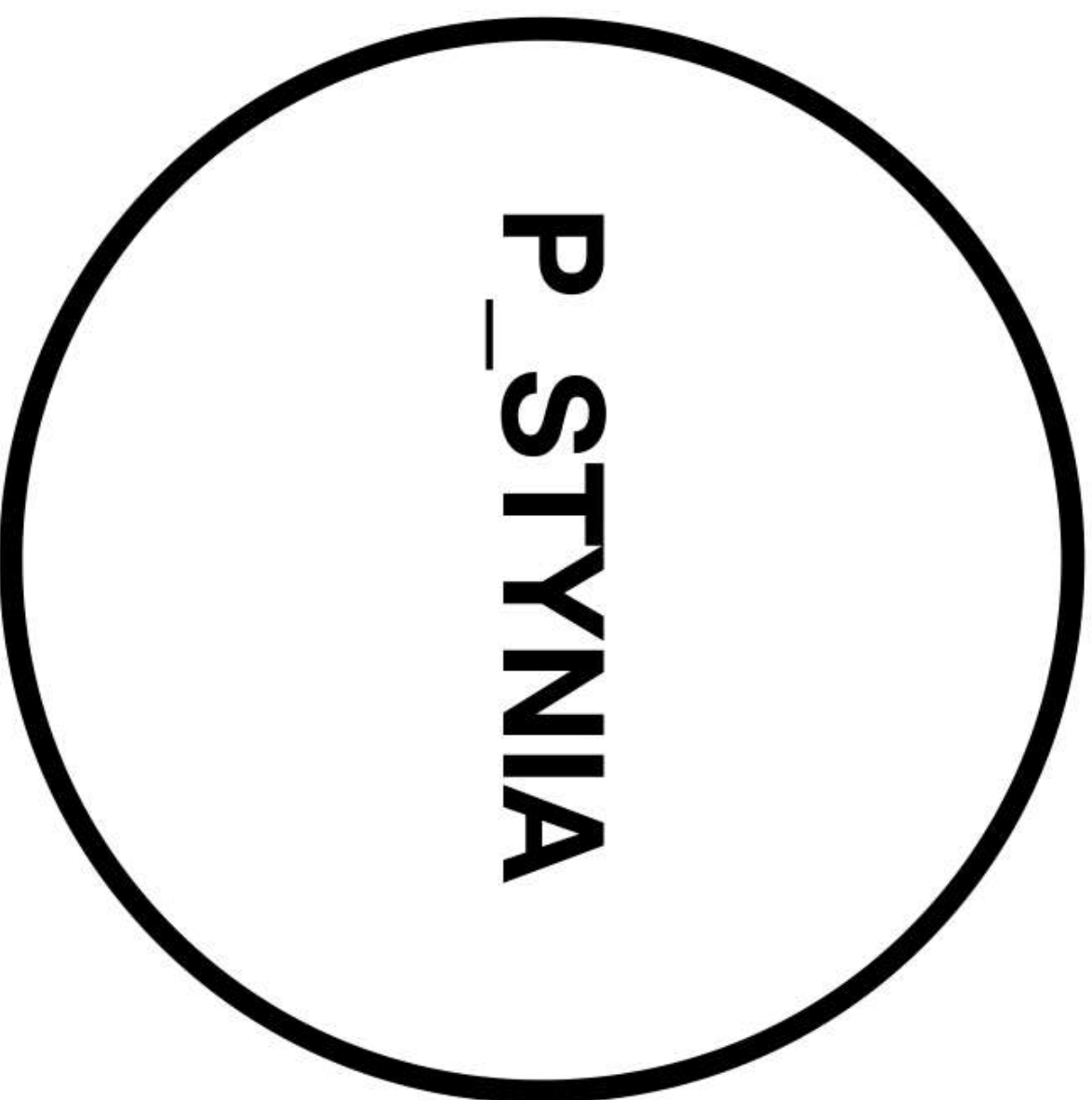
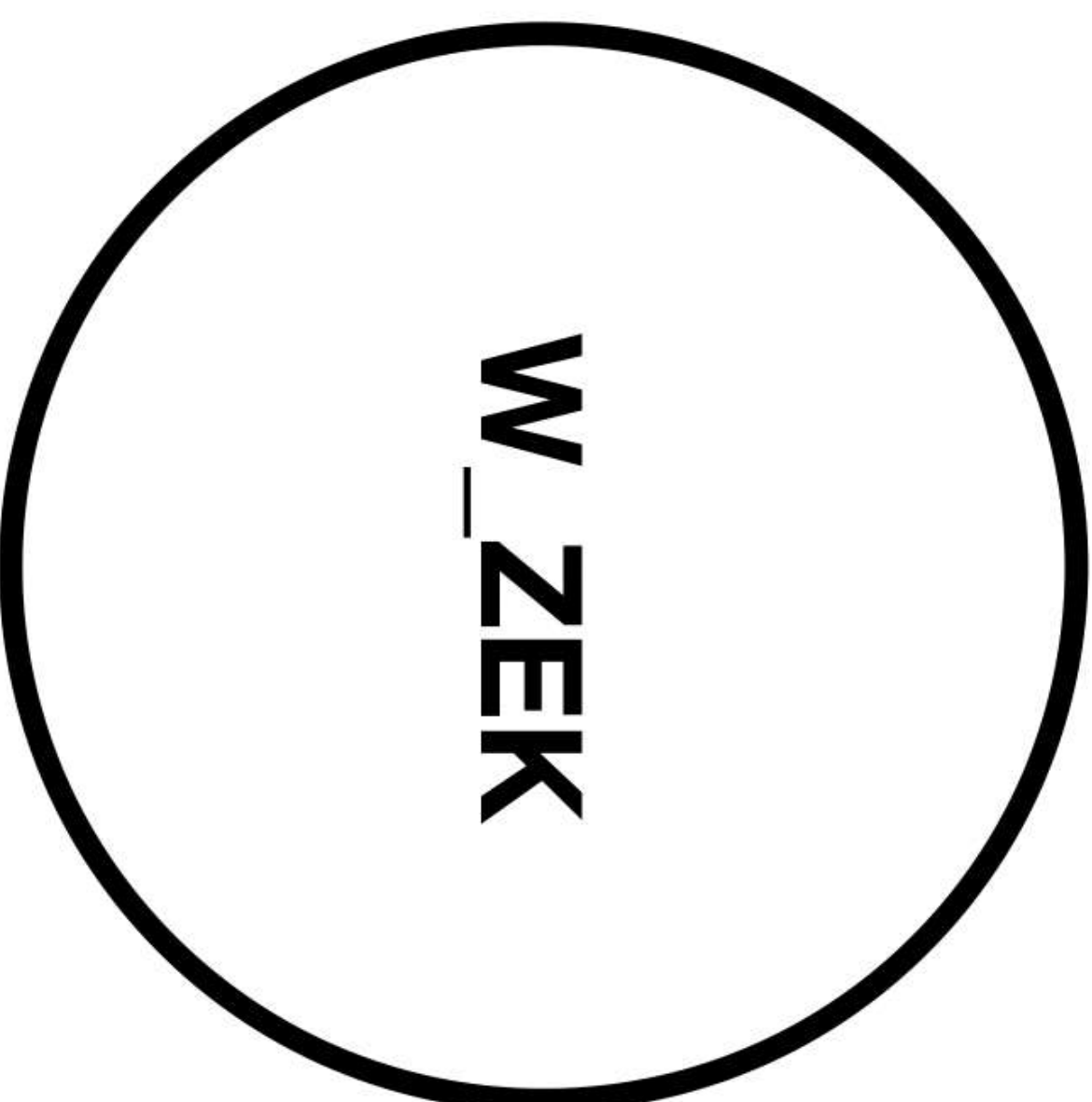
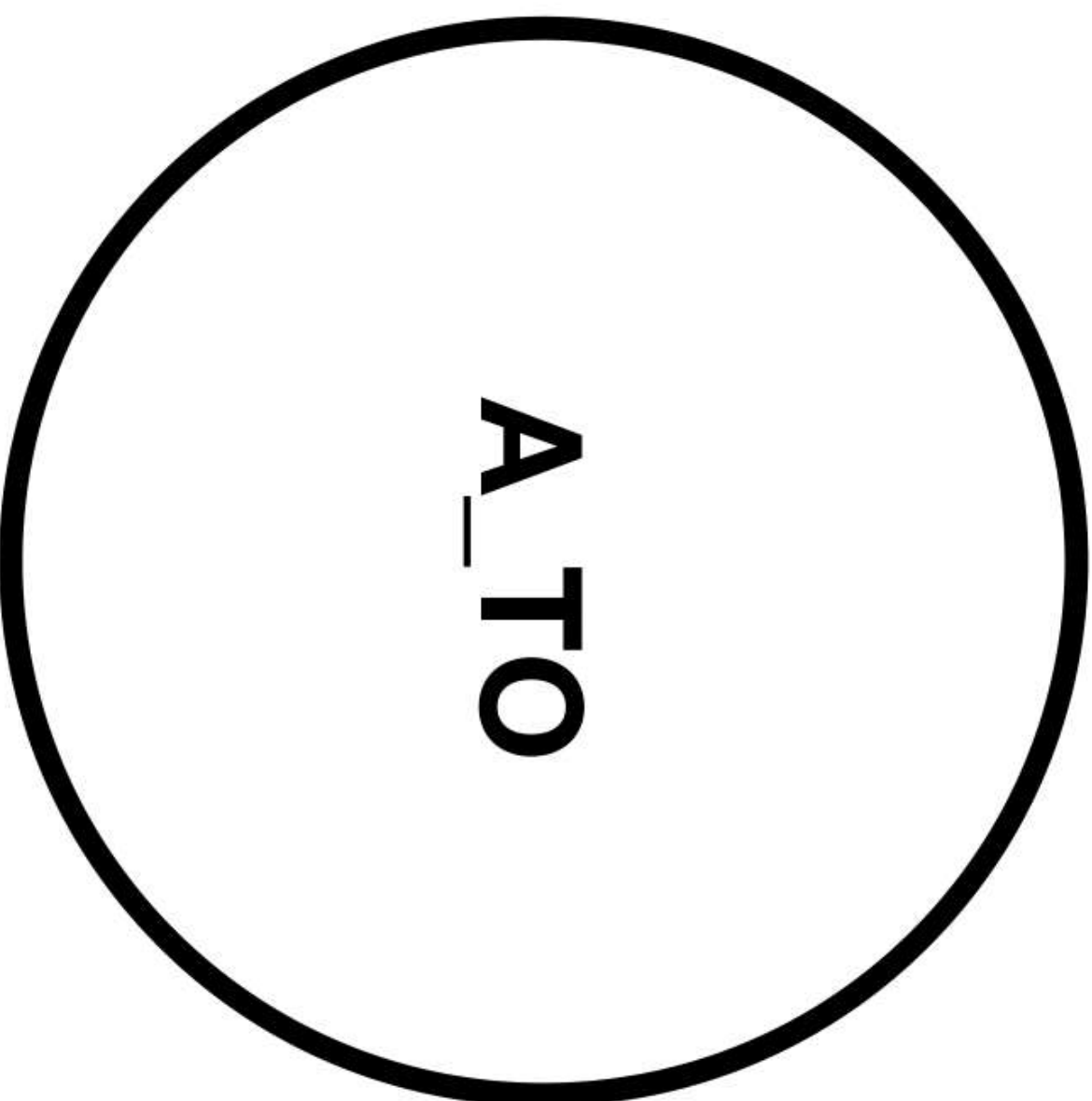
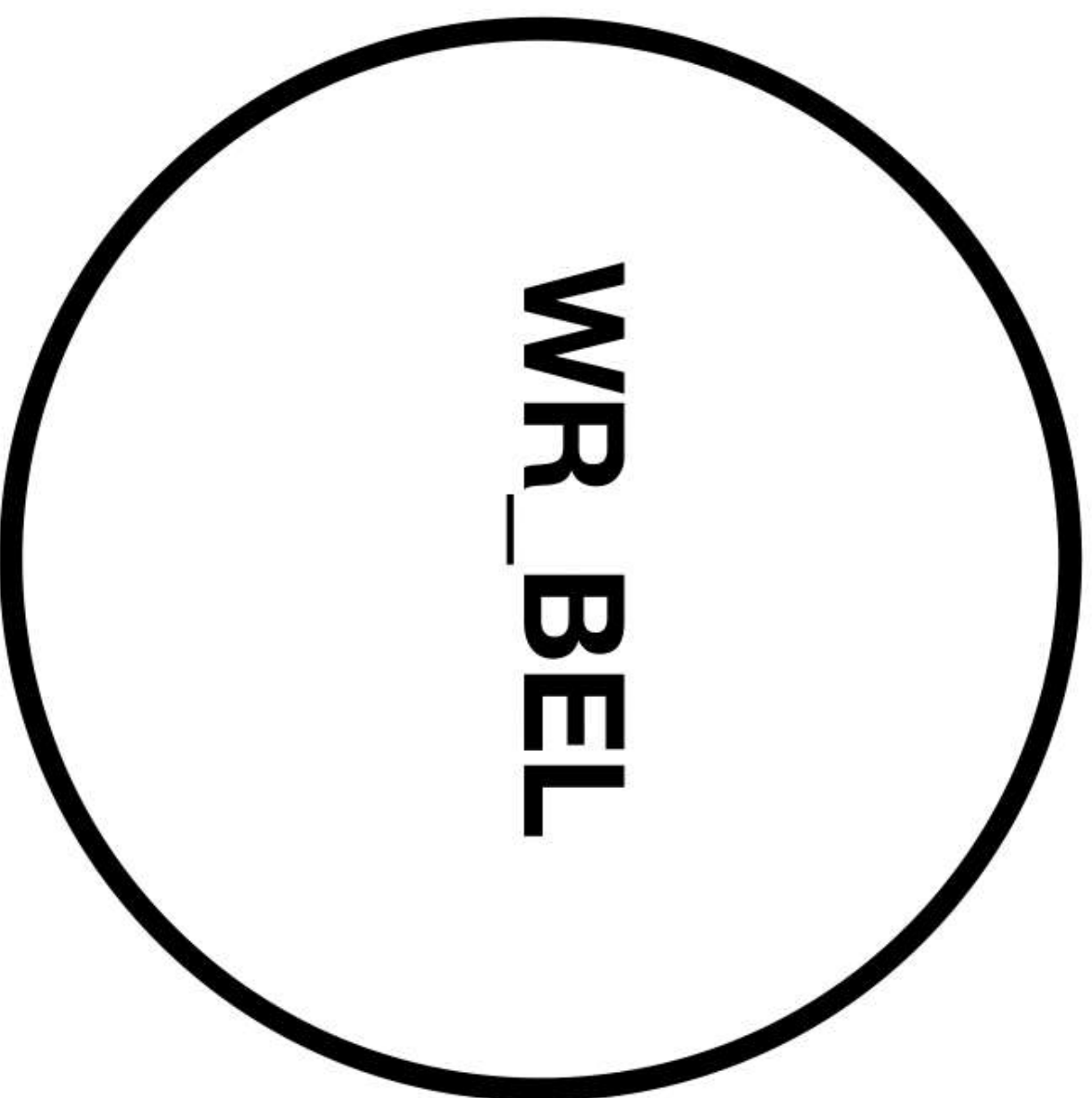
_OINKA

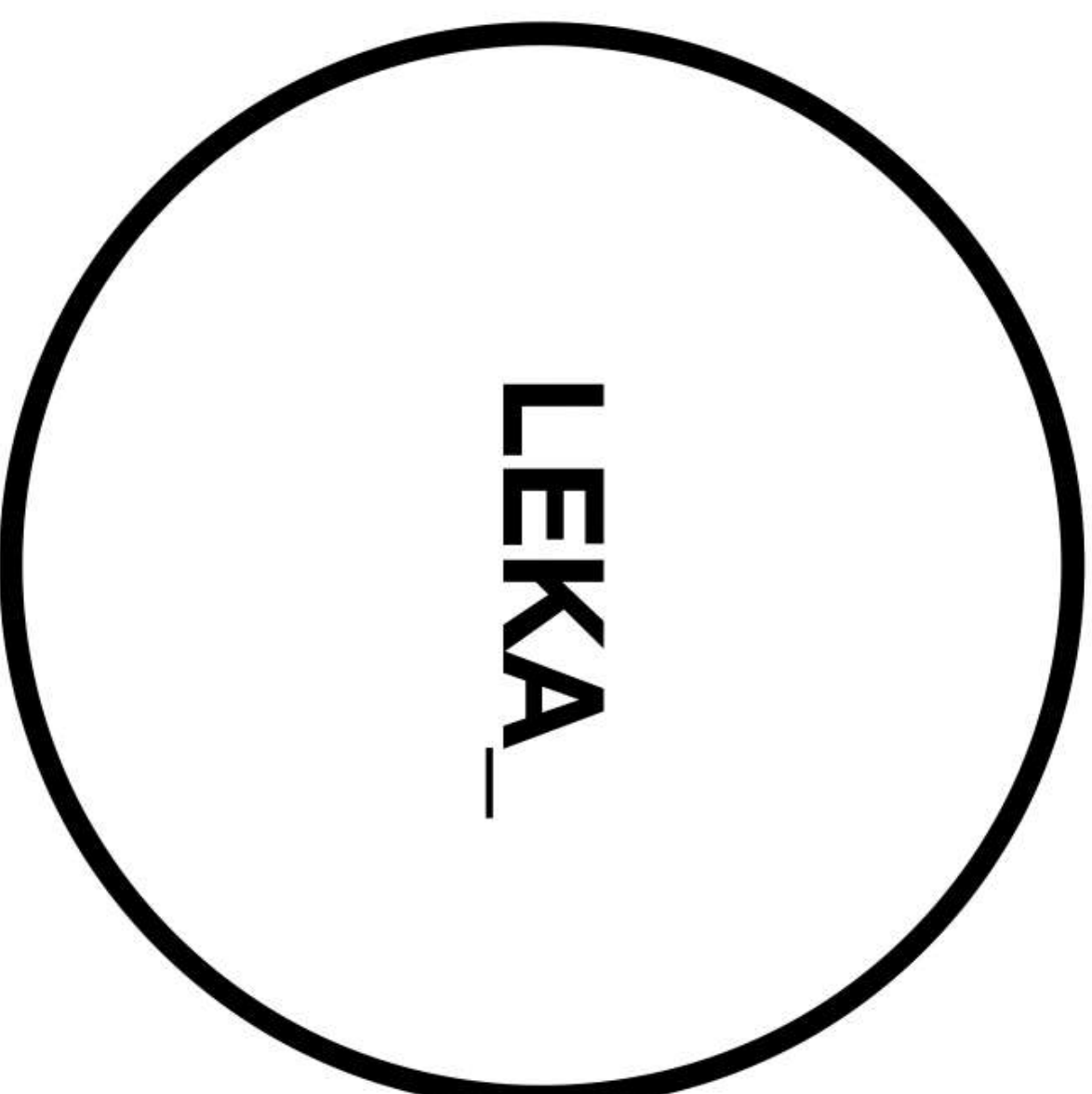
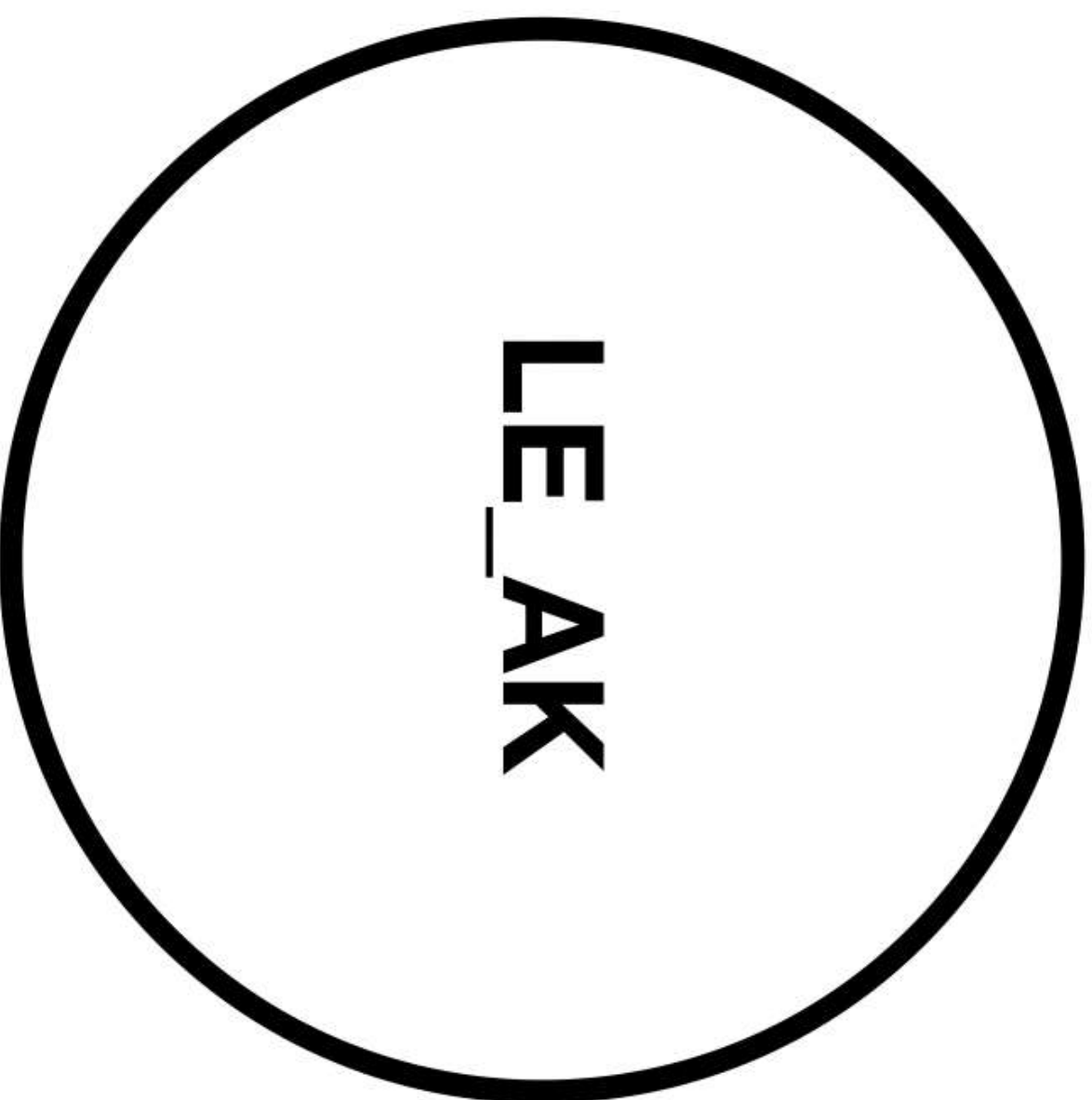
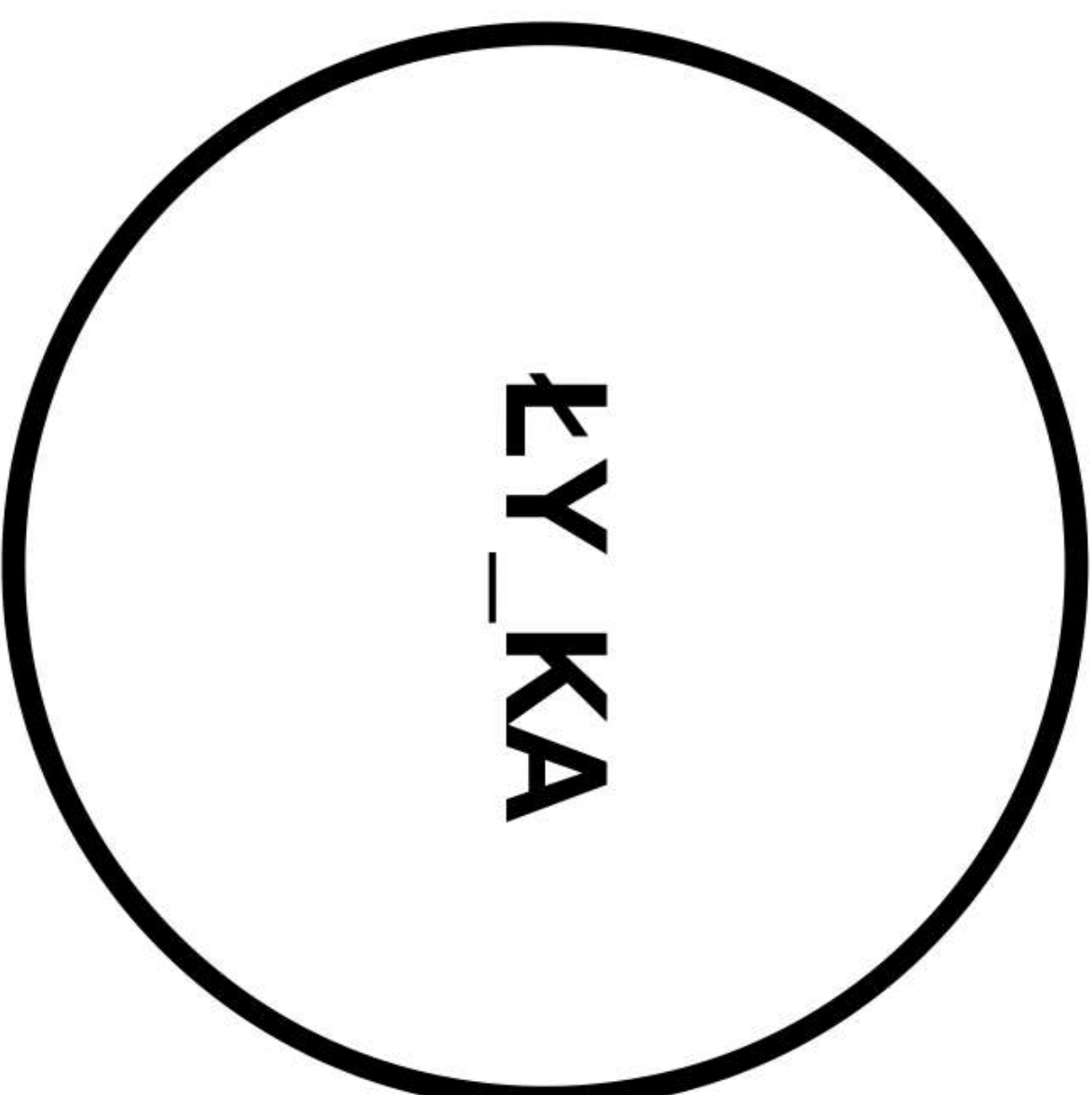
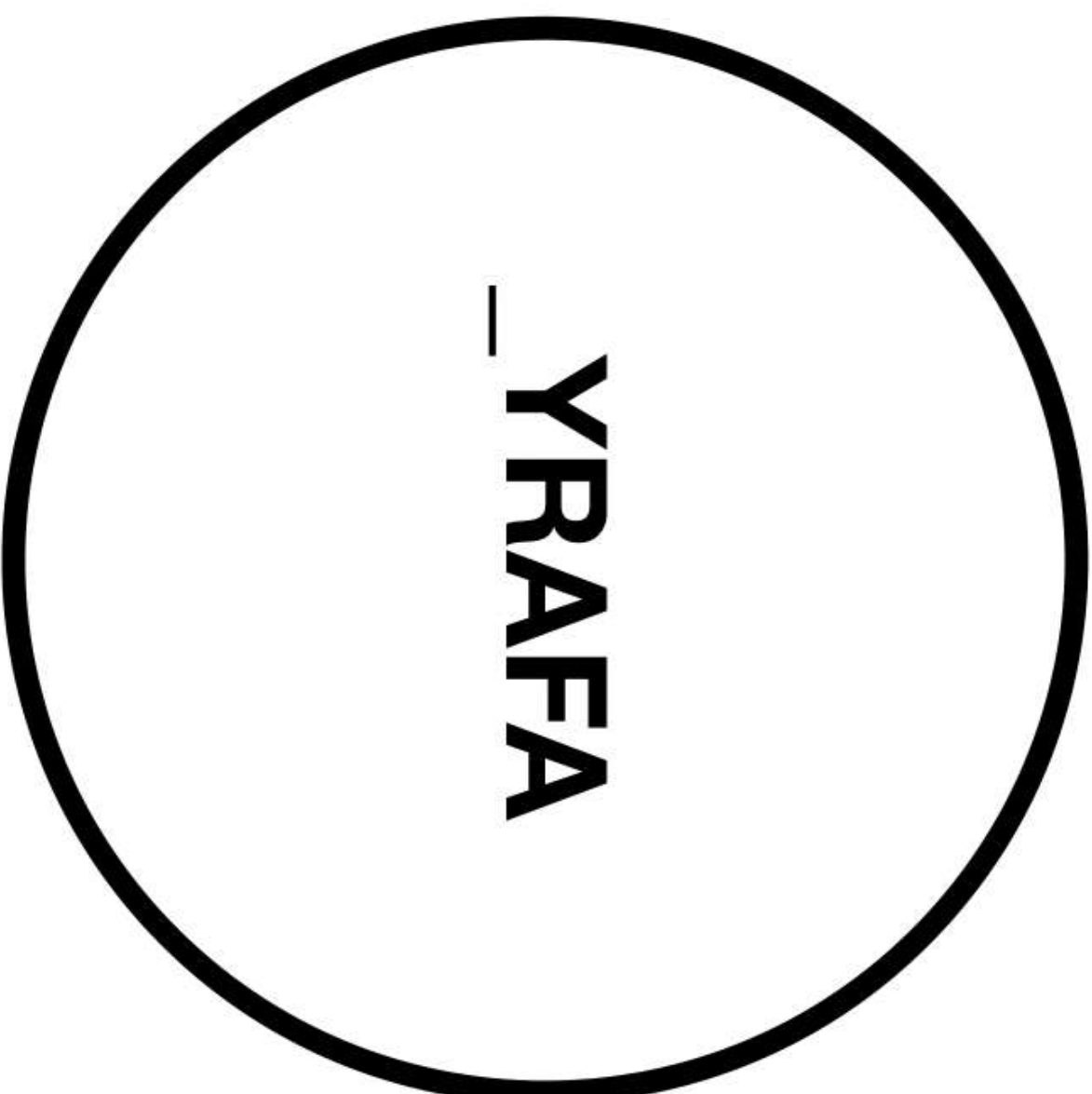
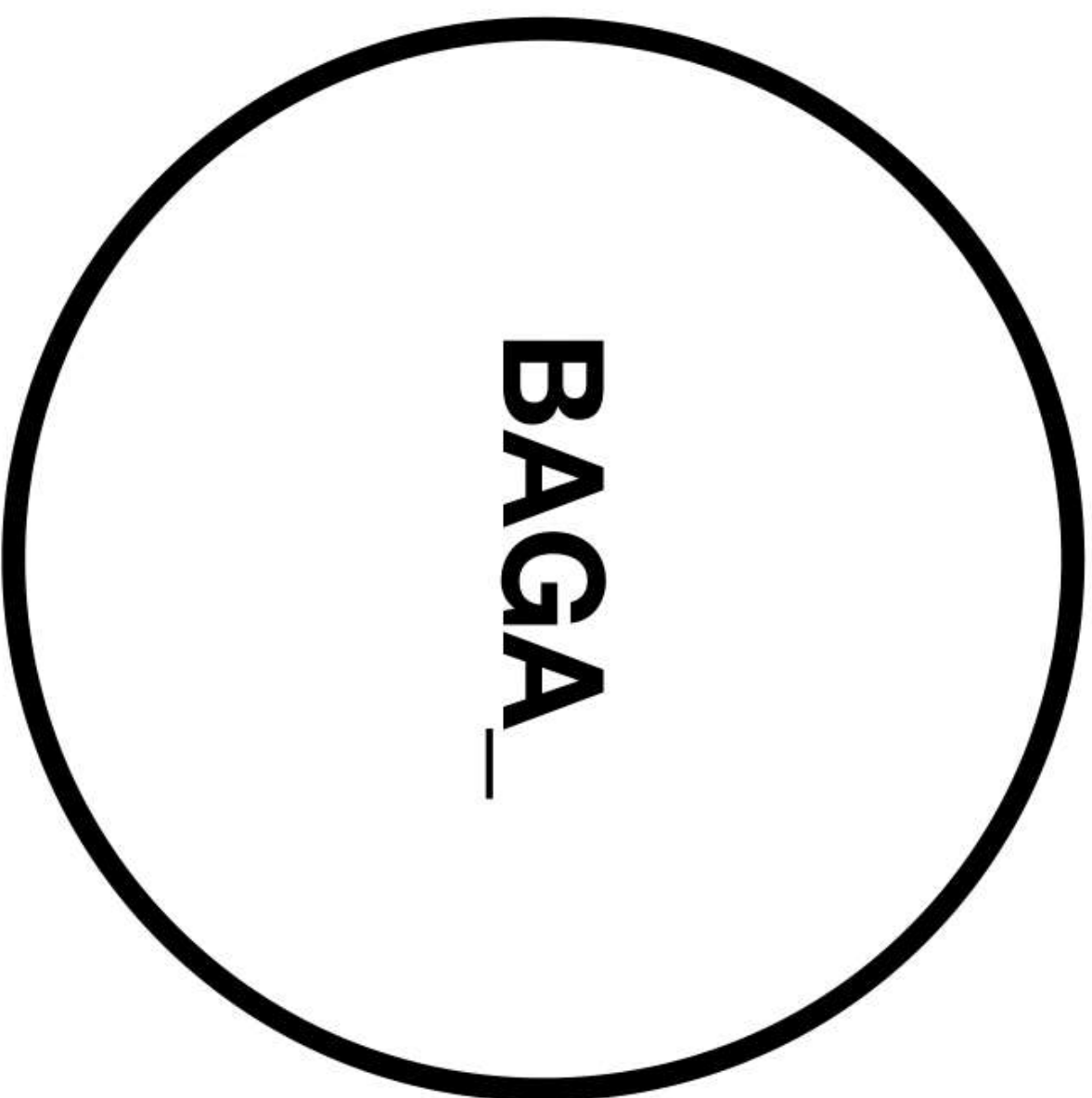
_UK

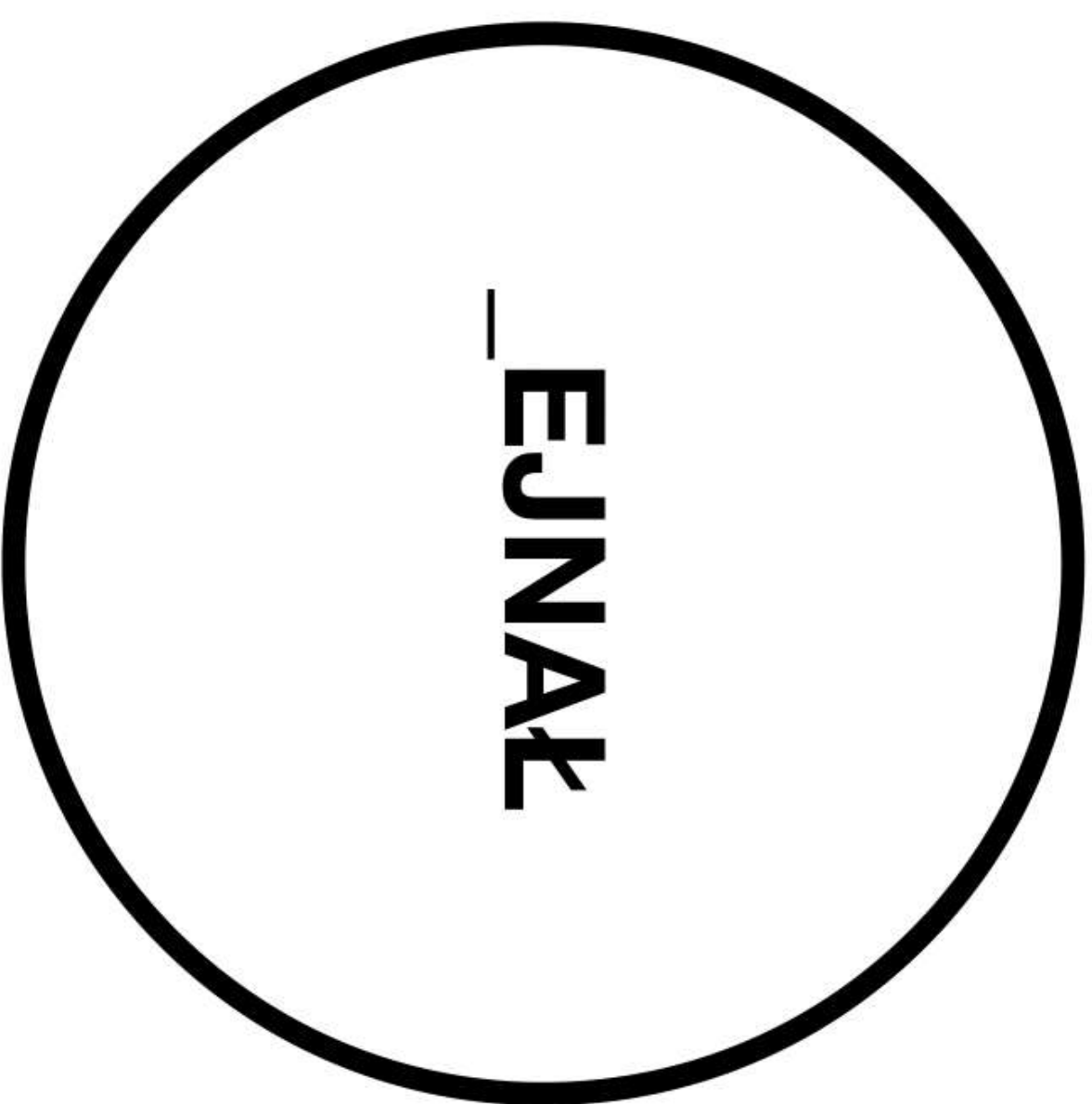
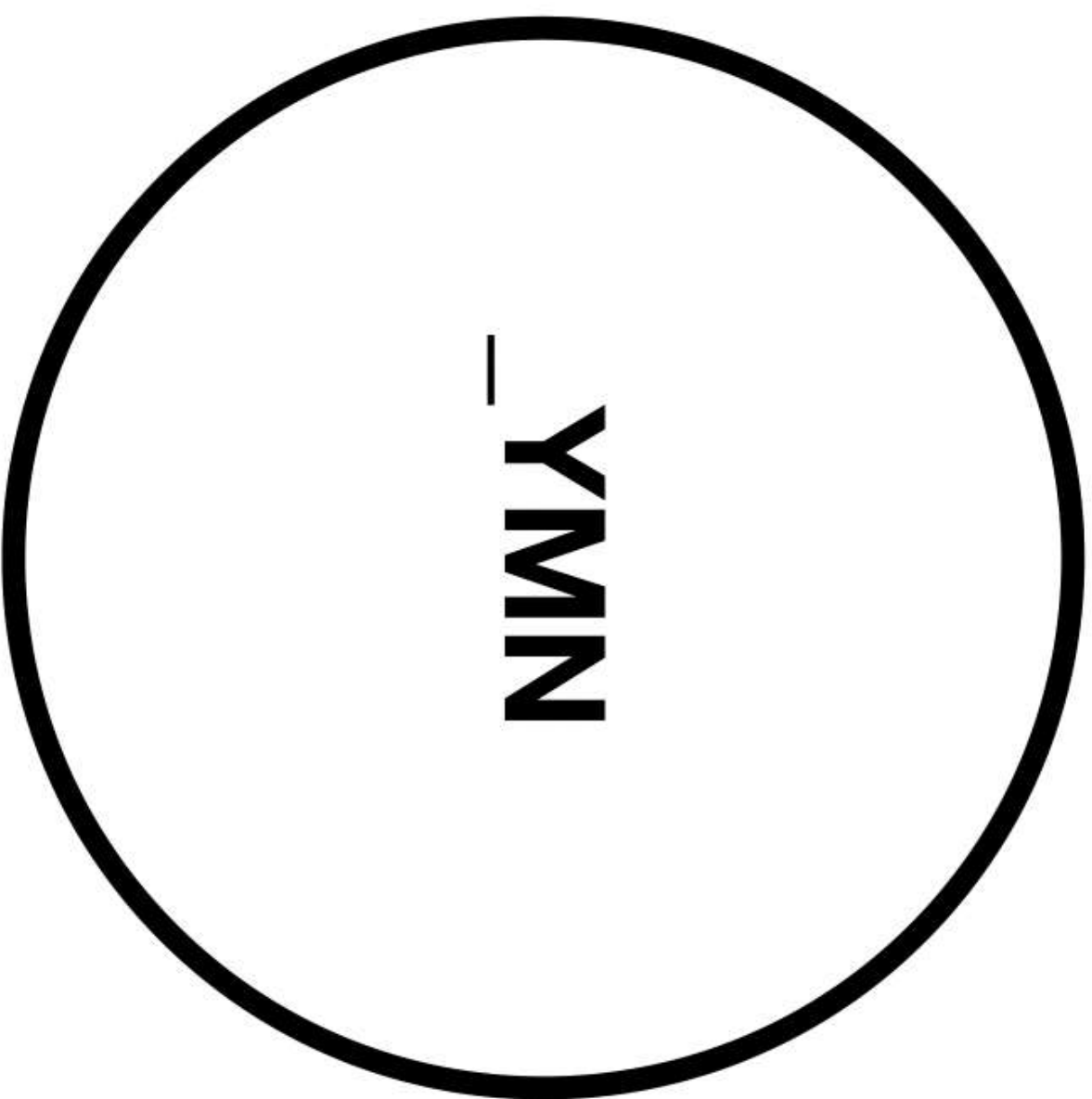
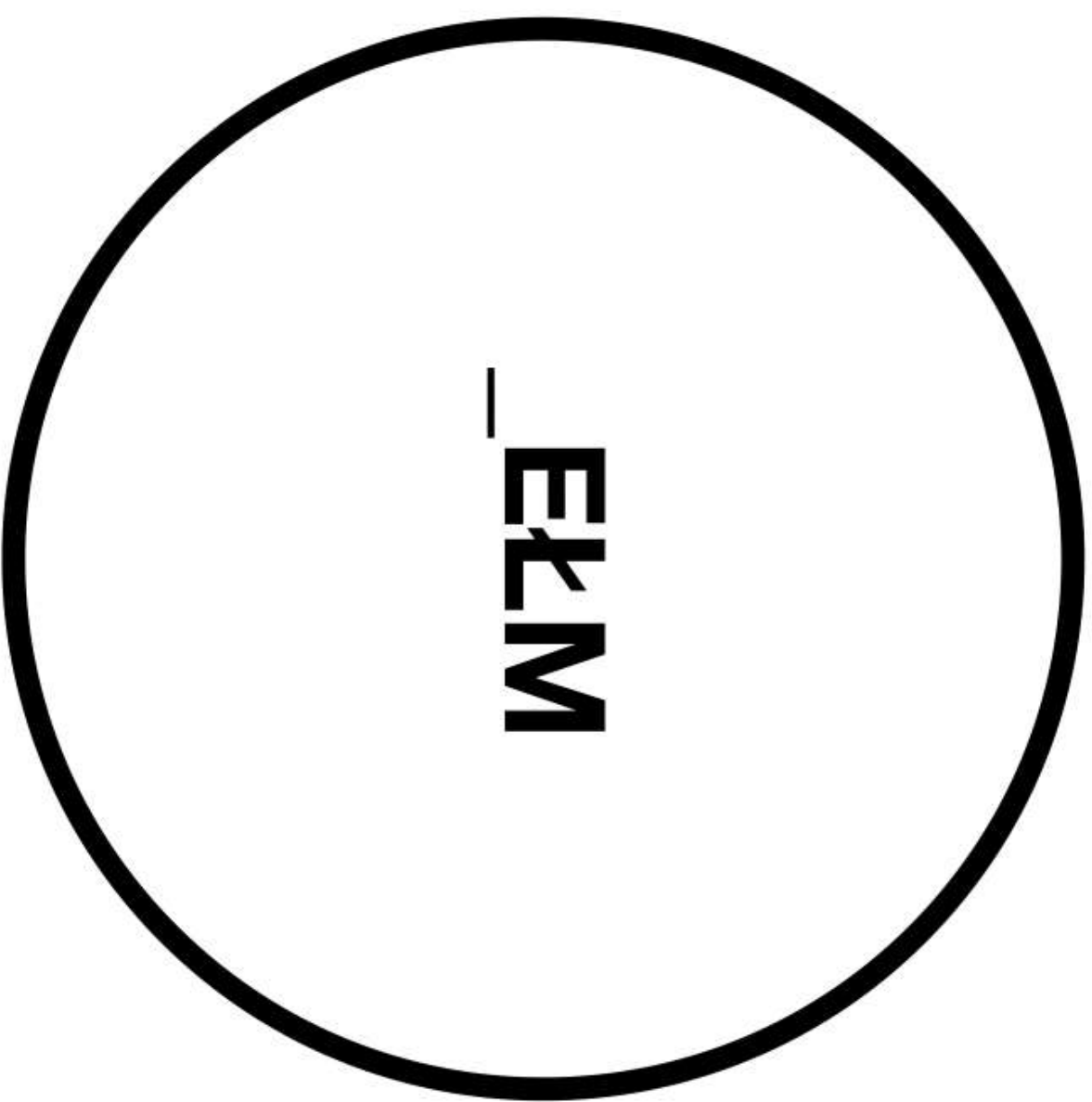
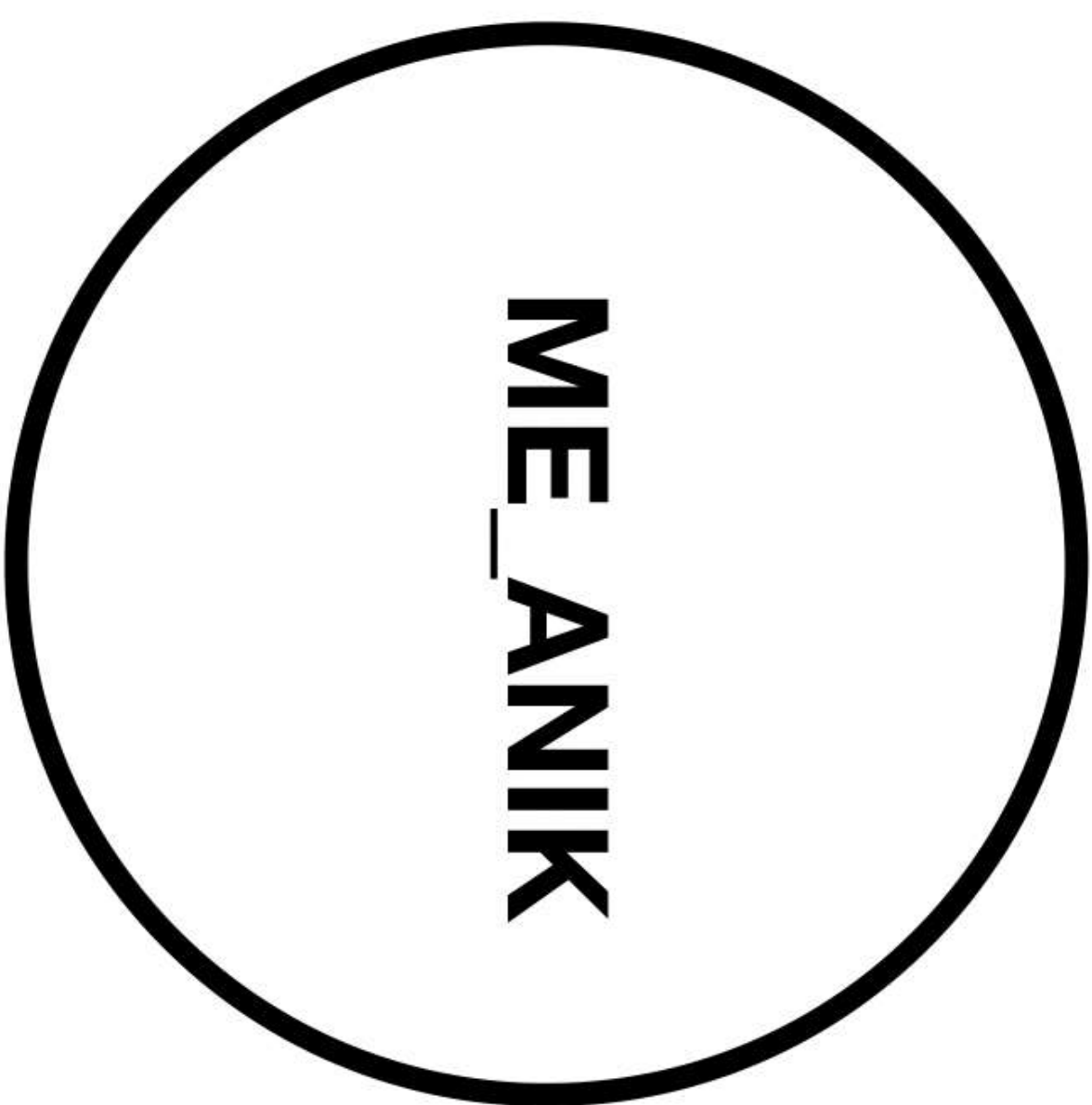
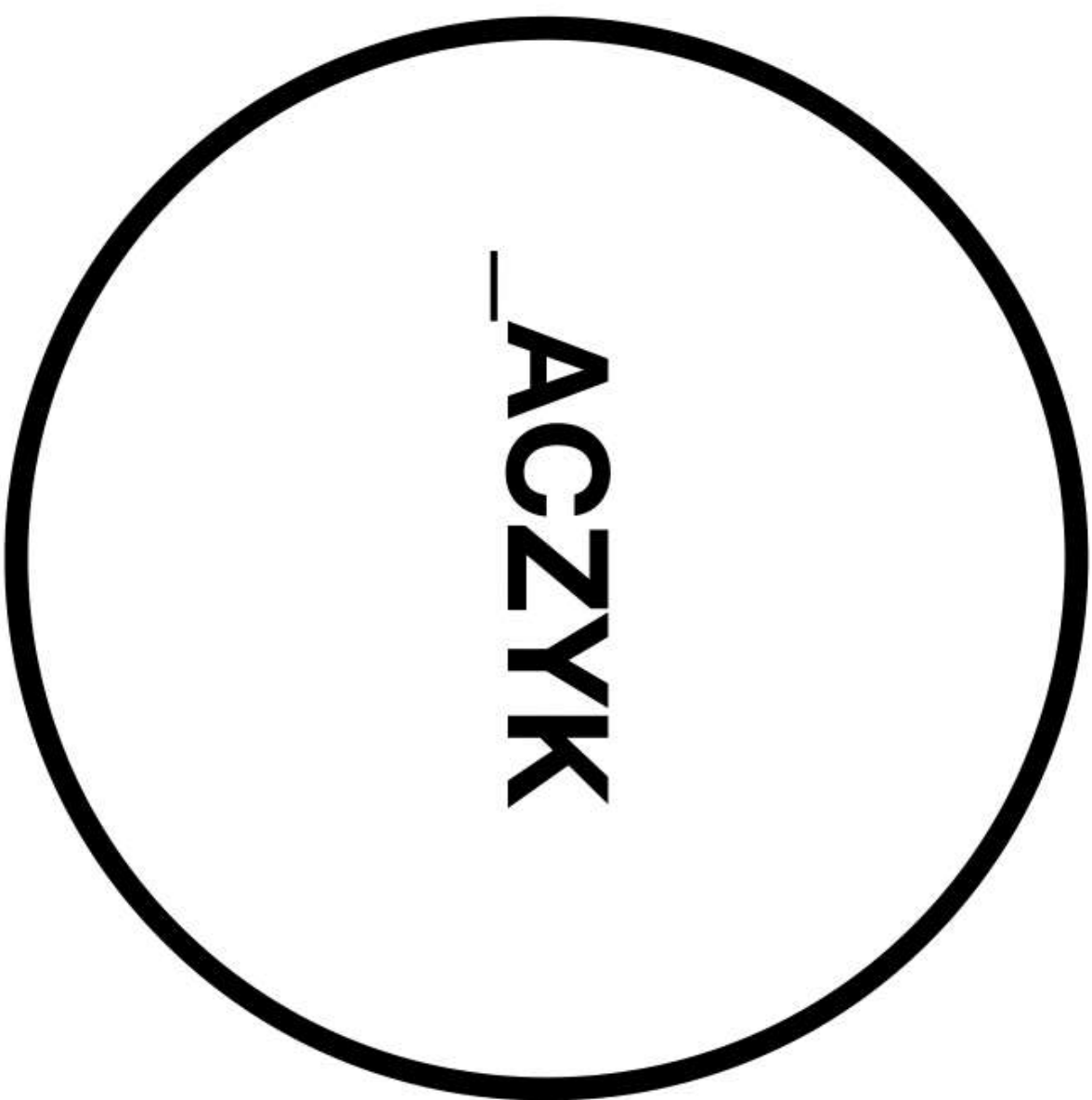
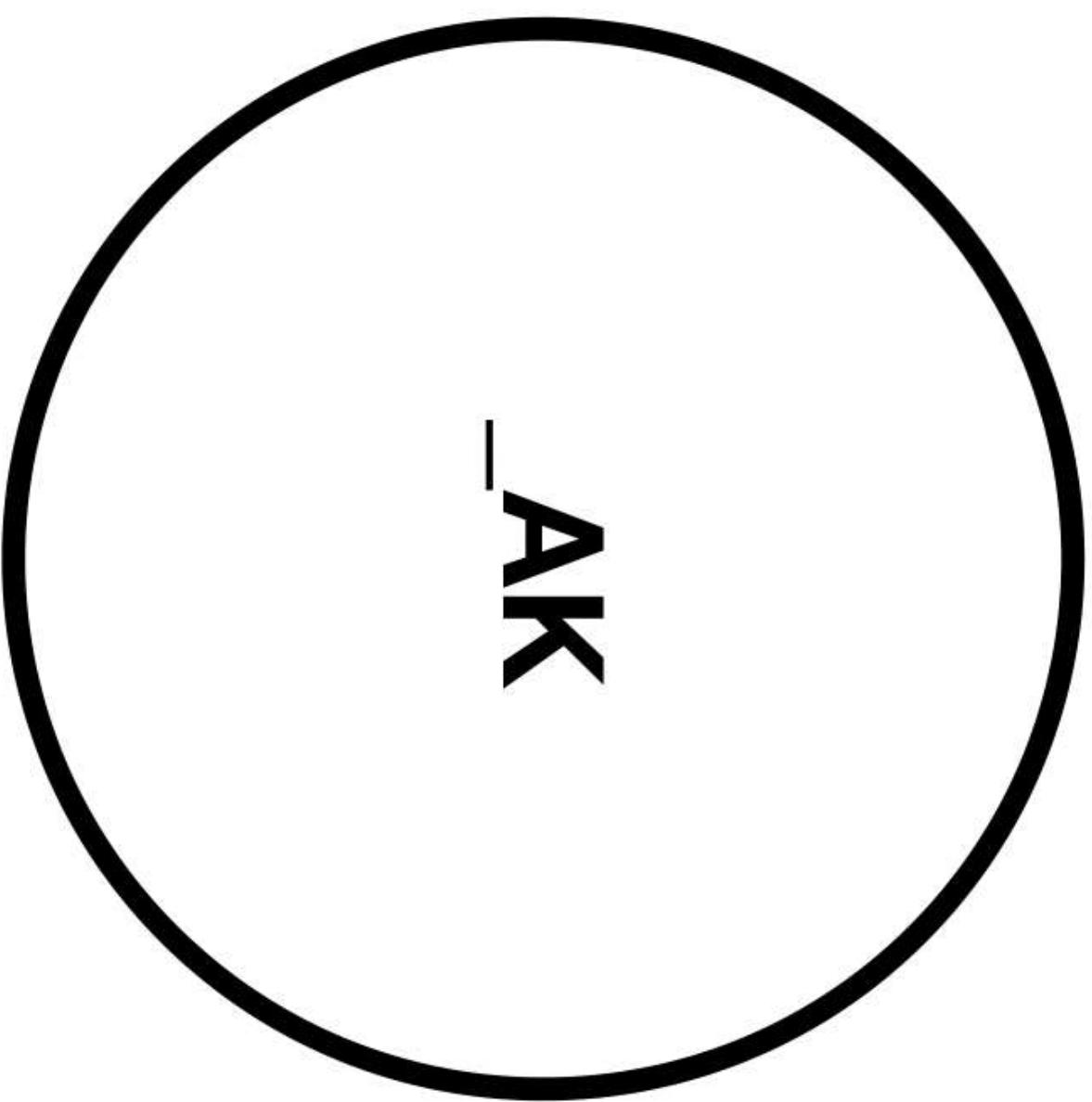
_ARCERKA

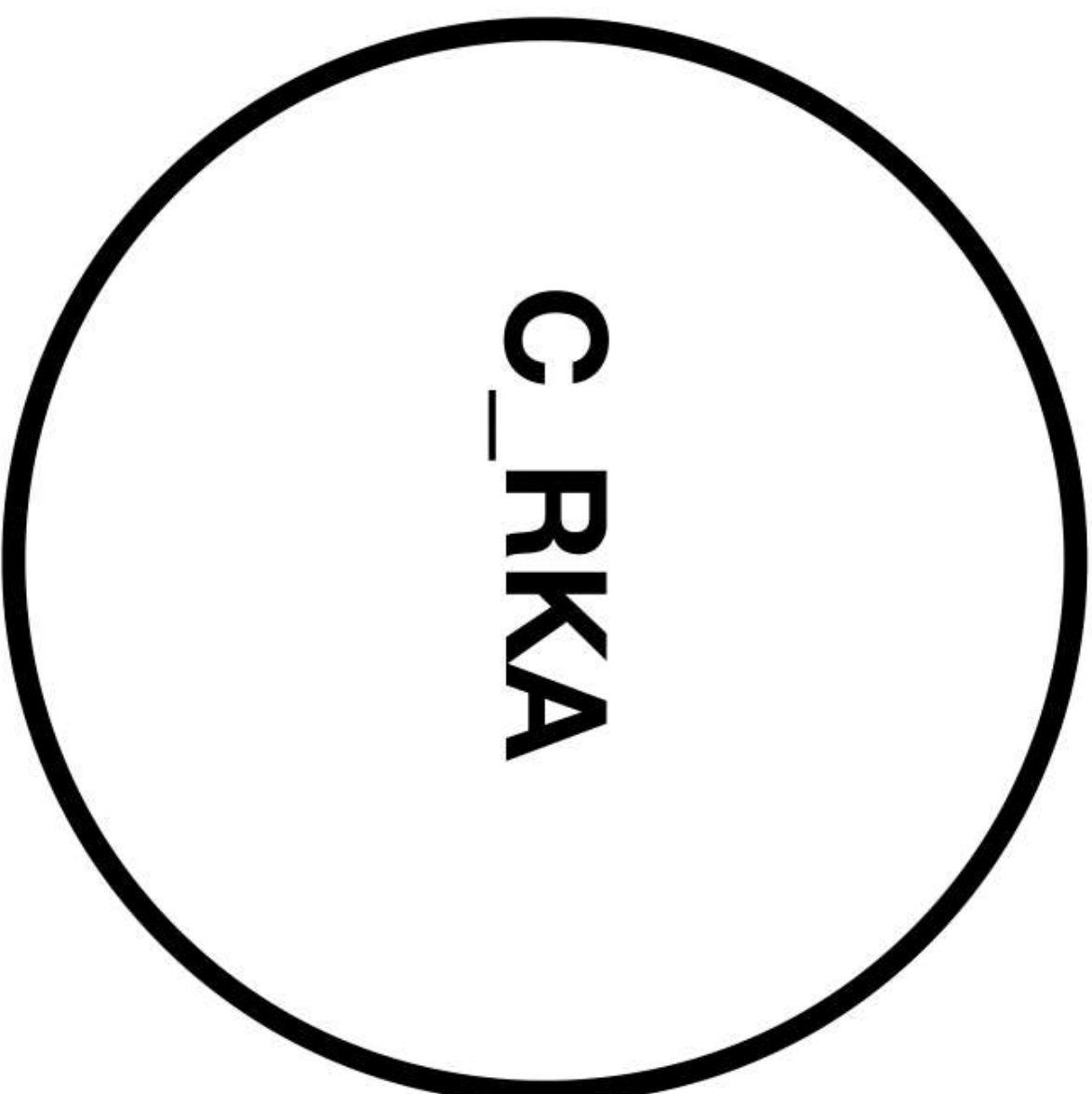
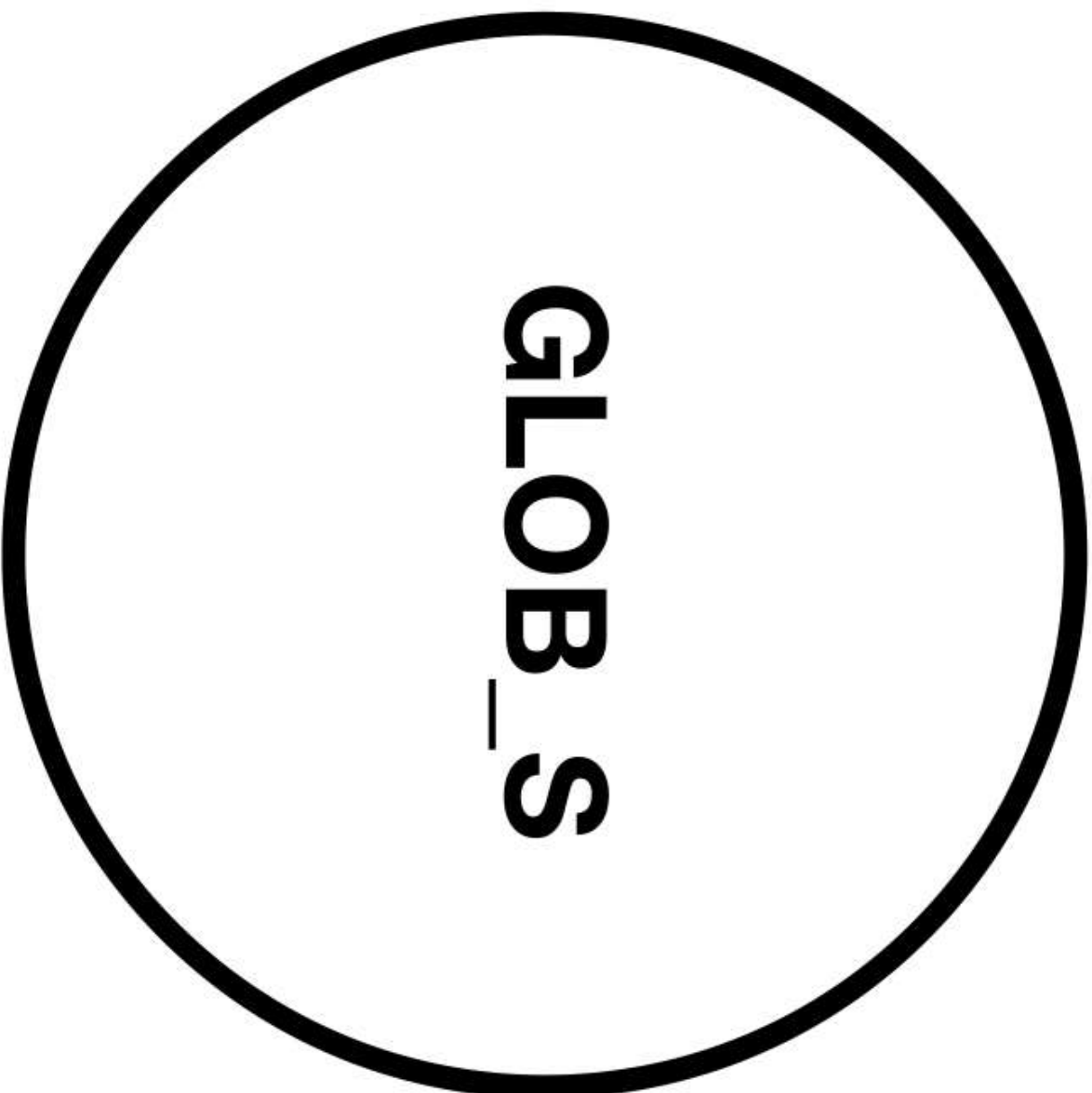
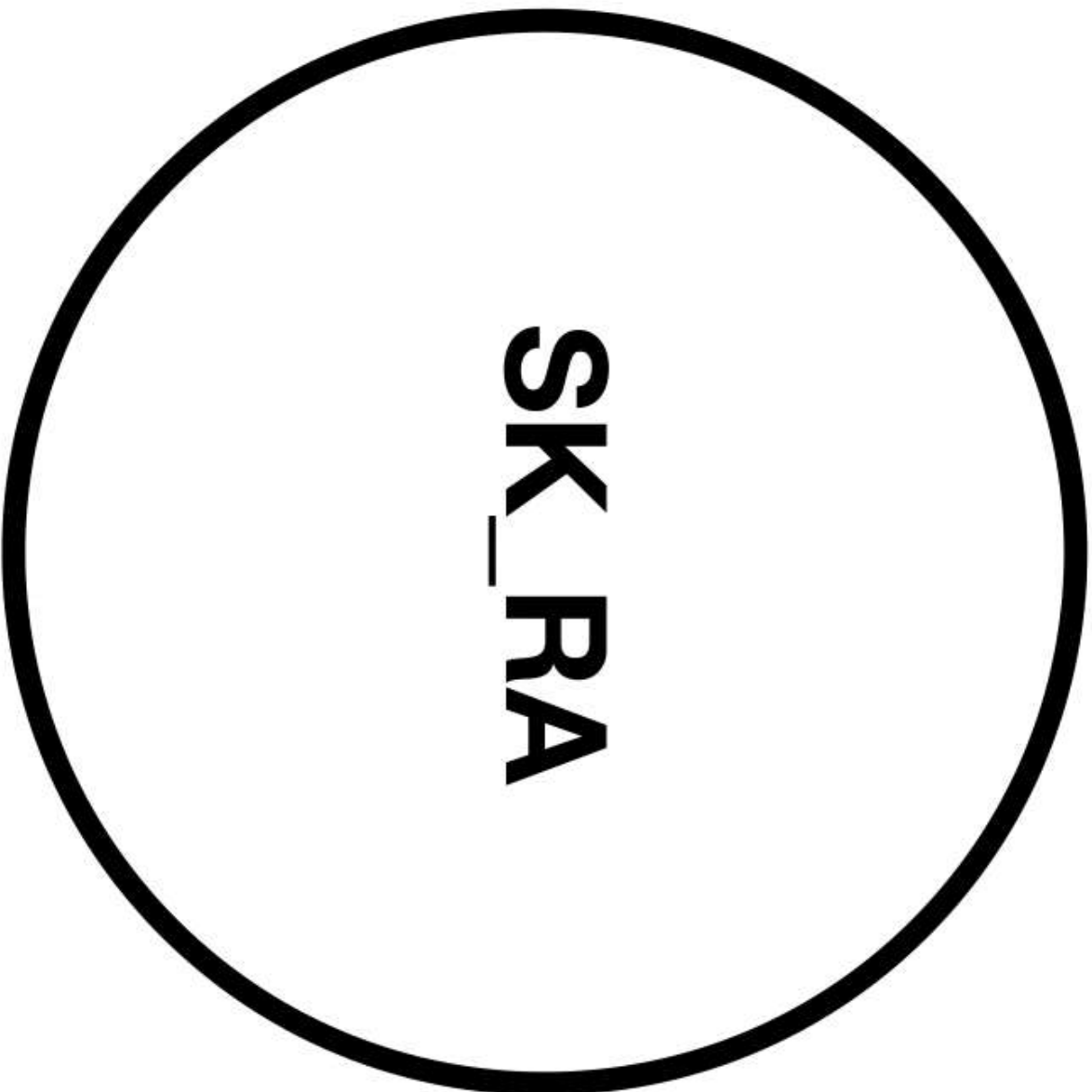
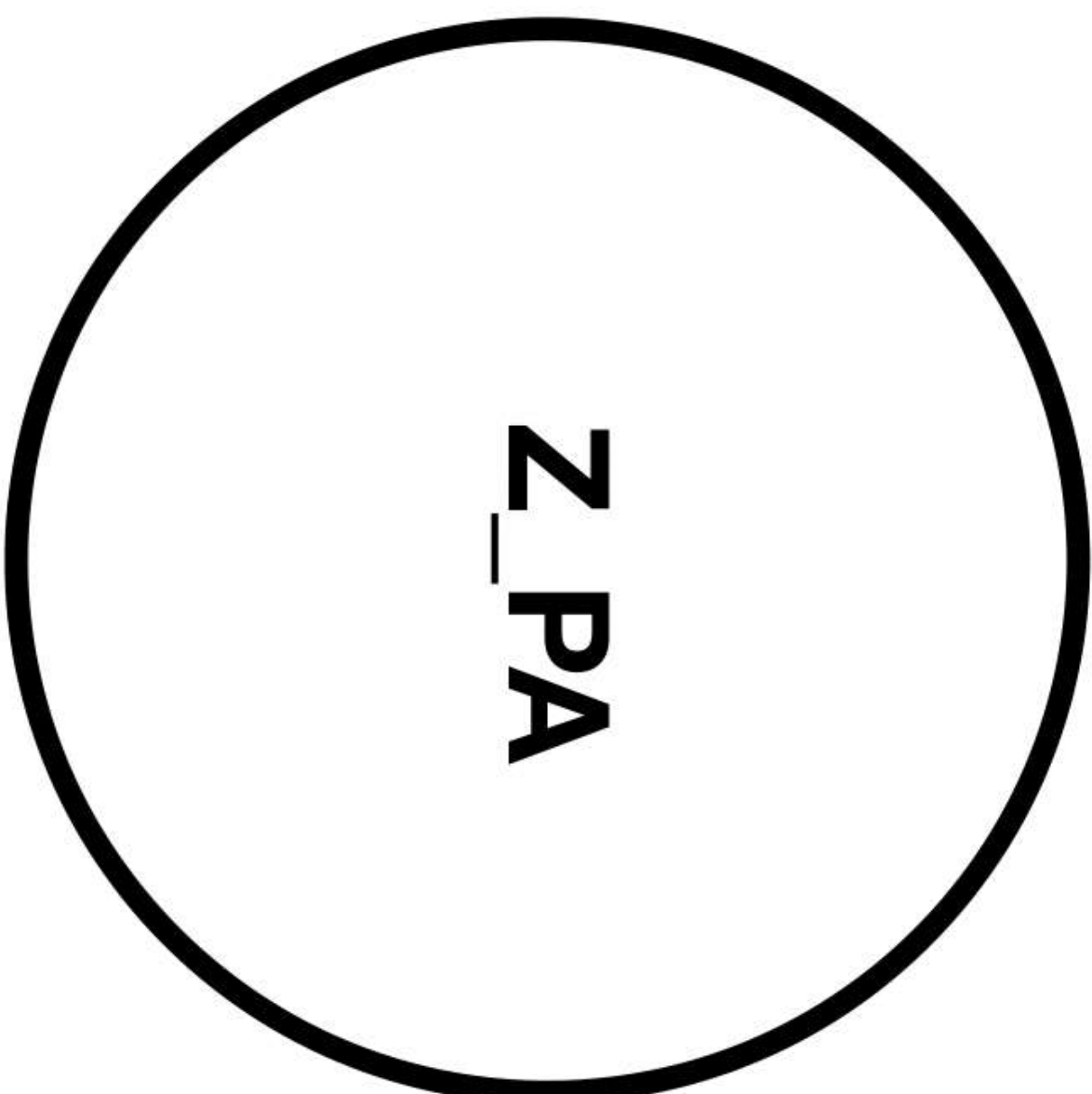
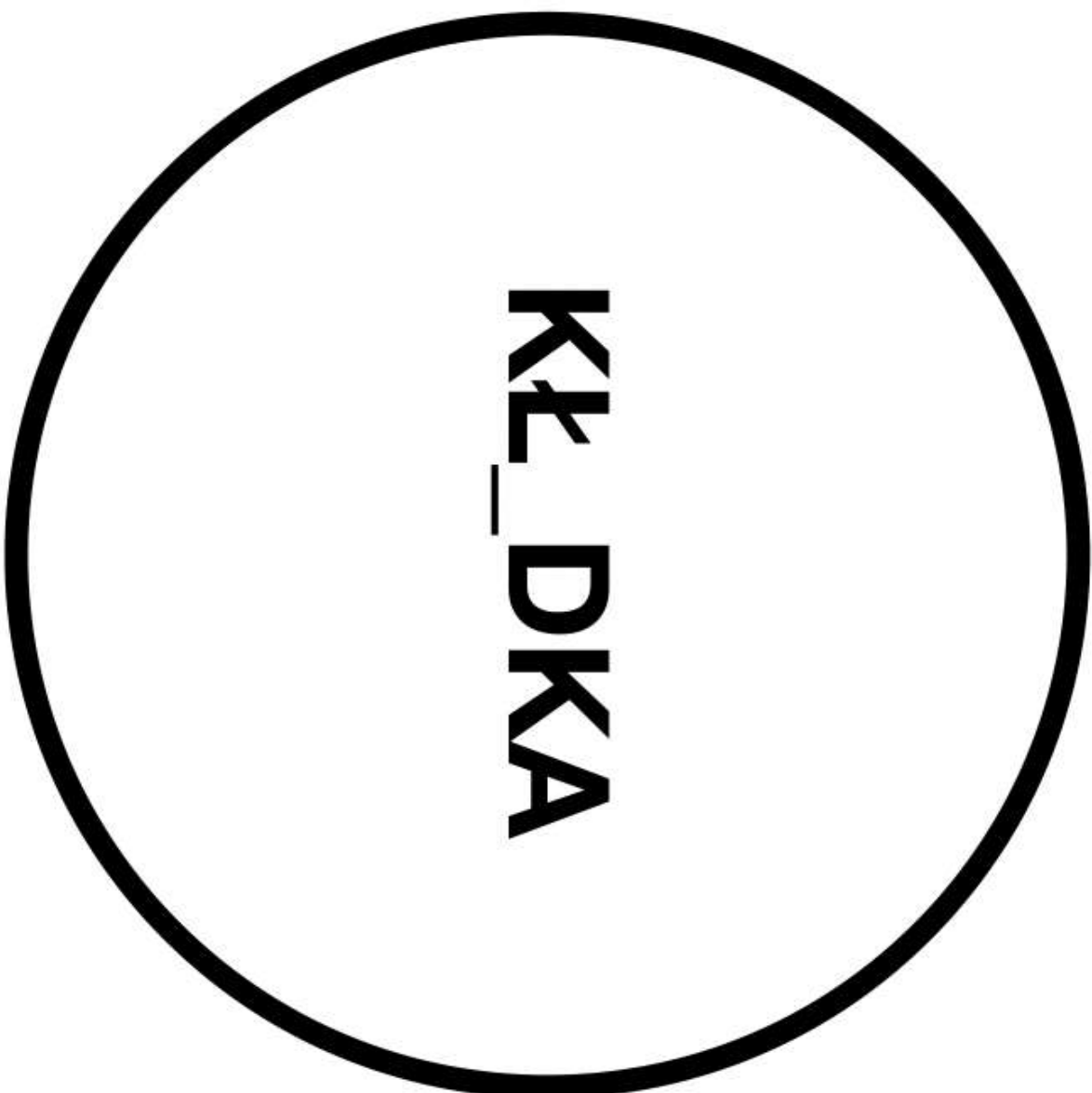
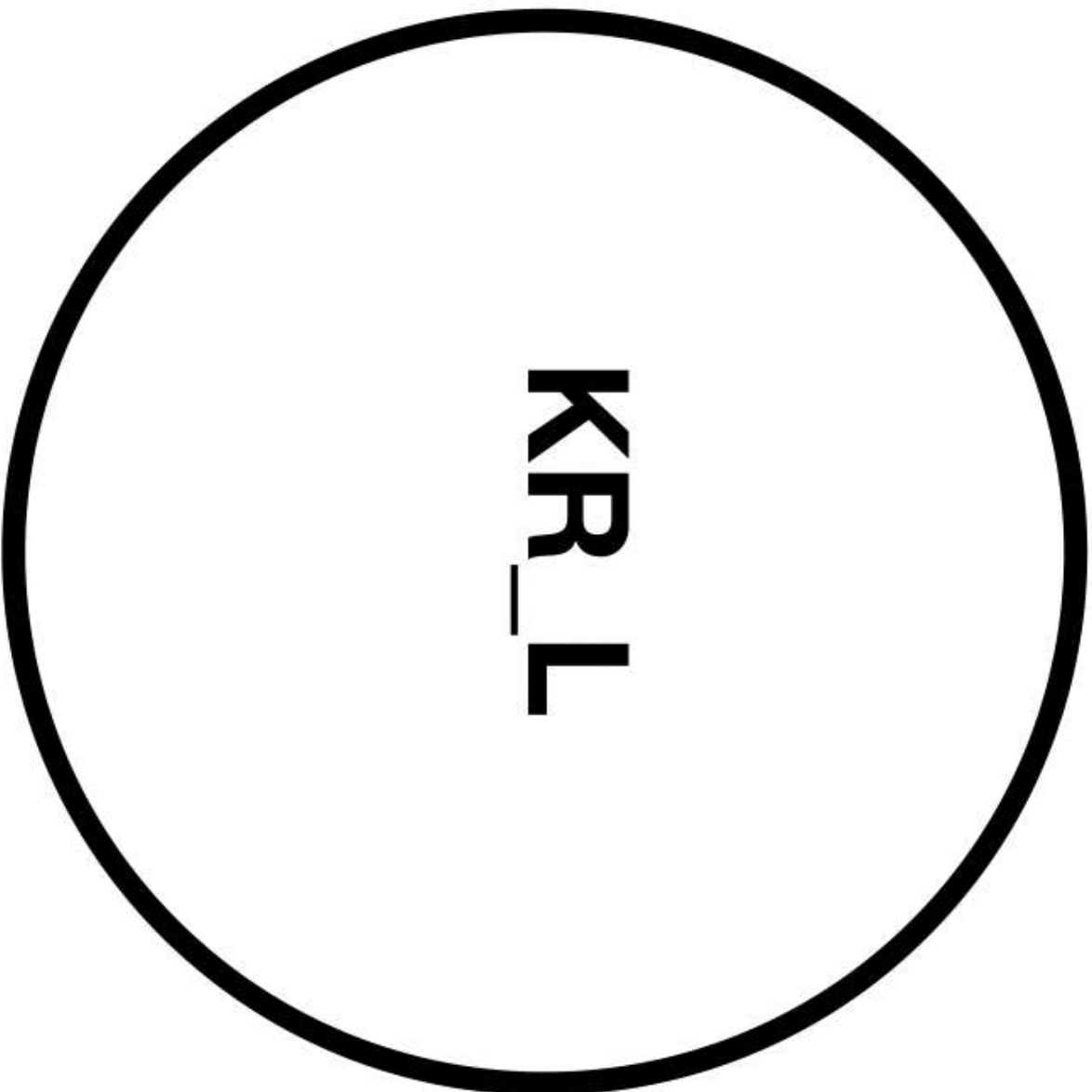
_ABRY

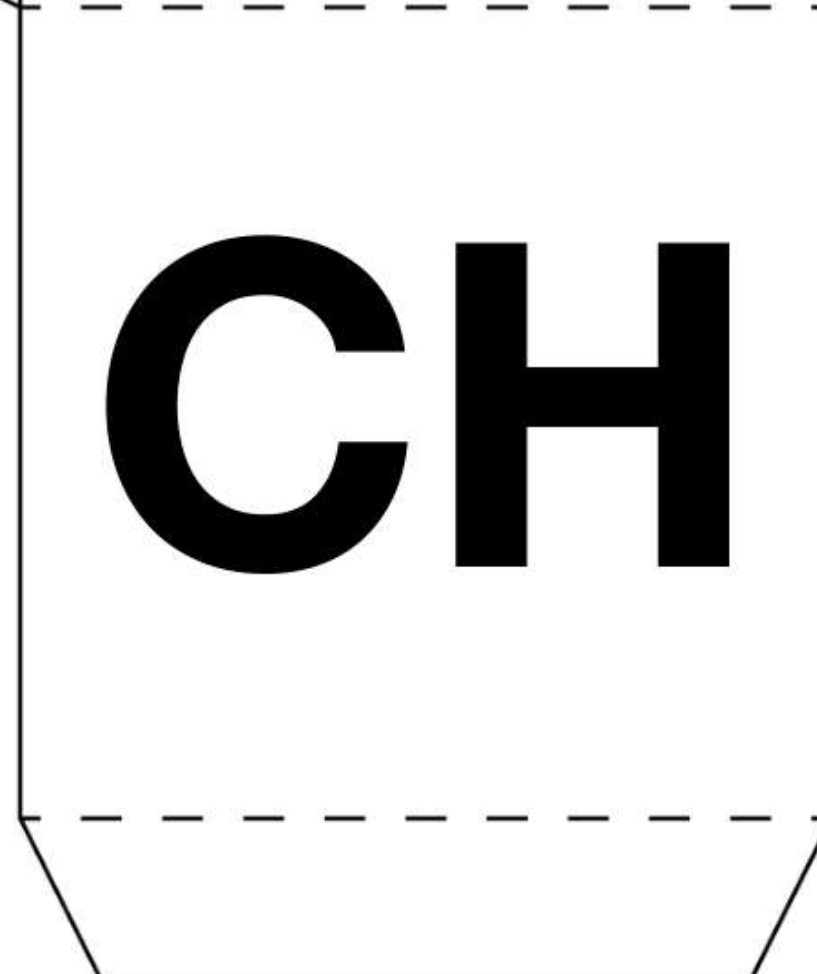
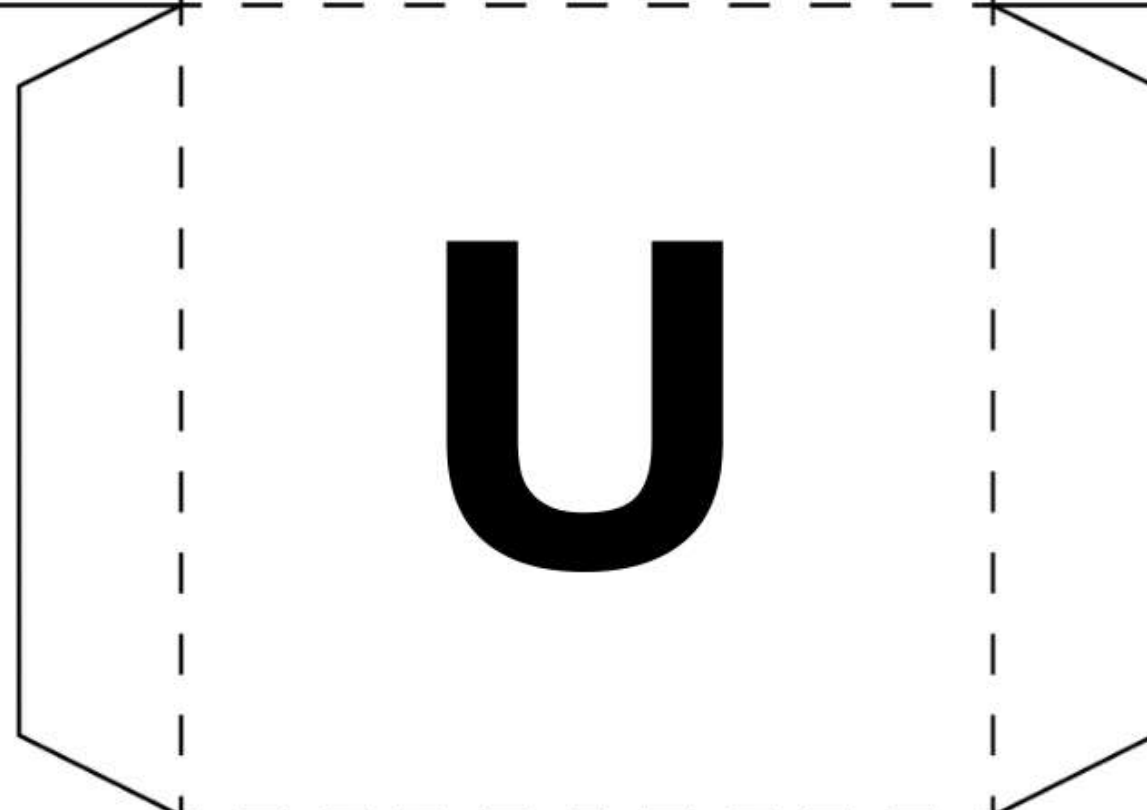
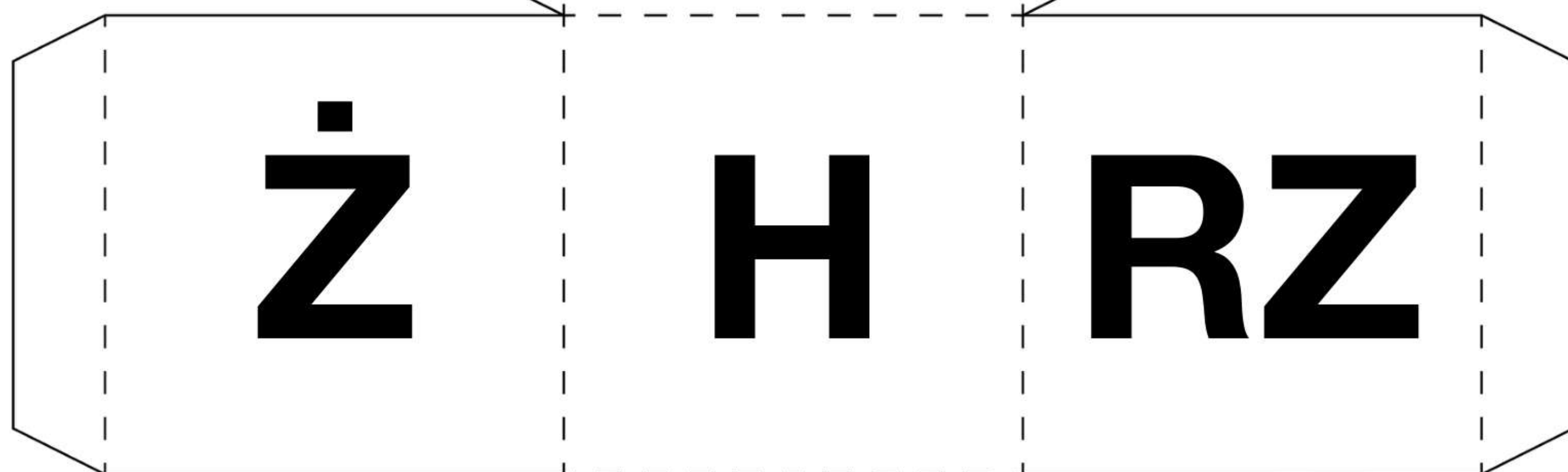
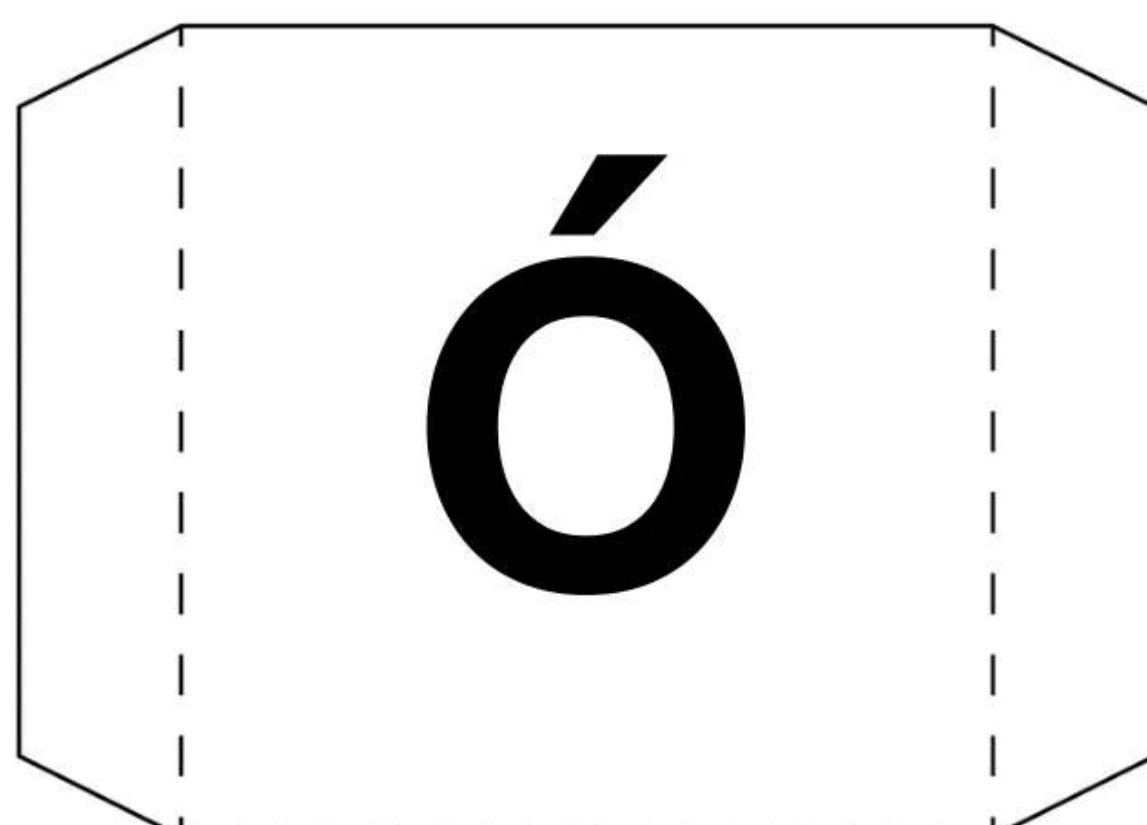
_RABĄSZCZ











**ZAPROGRAMOWANE
EMOCJE**

Czy robot odczuwa emocje? Nie, ale to nie znaczy, że nie możemy zaprogramować go w taki sposób, żeby naśladował sposób ich okazywania. Na zajęciach przeprowadzonych według zaproponowanego scenariusza uczniowie porozmawiają o emocjach, zastanowią się, jakie sytuacje je wywołują, w jaki sposób można radzić sobie z różnymi emocjami. Pomogą im w tym roboty, ale tylko wtedy, kiedy zostaną przez dzieci prawidłowo zaprogramowane...

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe;
- Kompetencje osobiste, społeczne i w zakresie uczenia się.

Cele ogólne:

- Poszerzenie wiadomości o emocjach;
- Rozwijanie logicznego, algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie miękkich kompetencji, ze szczególnym uwzględnieniem umiejętności pracy zespołowej.

Cele operacyjne:

Uczeń podczas zajęć:

- tworzy obrazek na podstawie podanych współrzędnych;
- rozpoznaje i nazywa emocje;
- przypisuje emocje do sytuacji, które mogły je wywołać;
- wykazuje się empatią;
- rozwiązuje logiczne zagadki;
- programuje robota Photon wykorzystując interfejs Photon Badge i Photon Blocks;
- chętnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

Ok. 45 – 90 minut

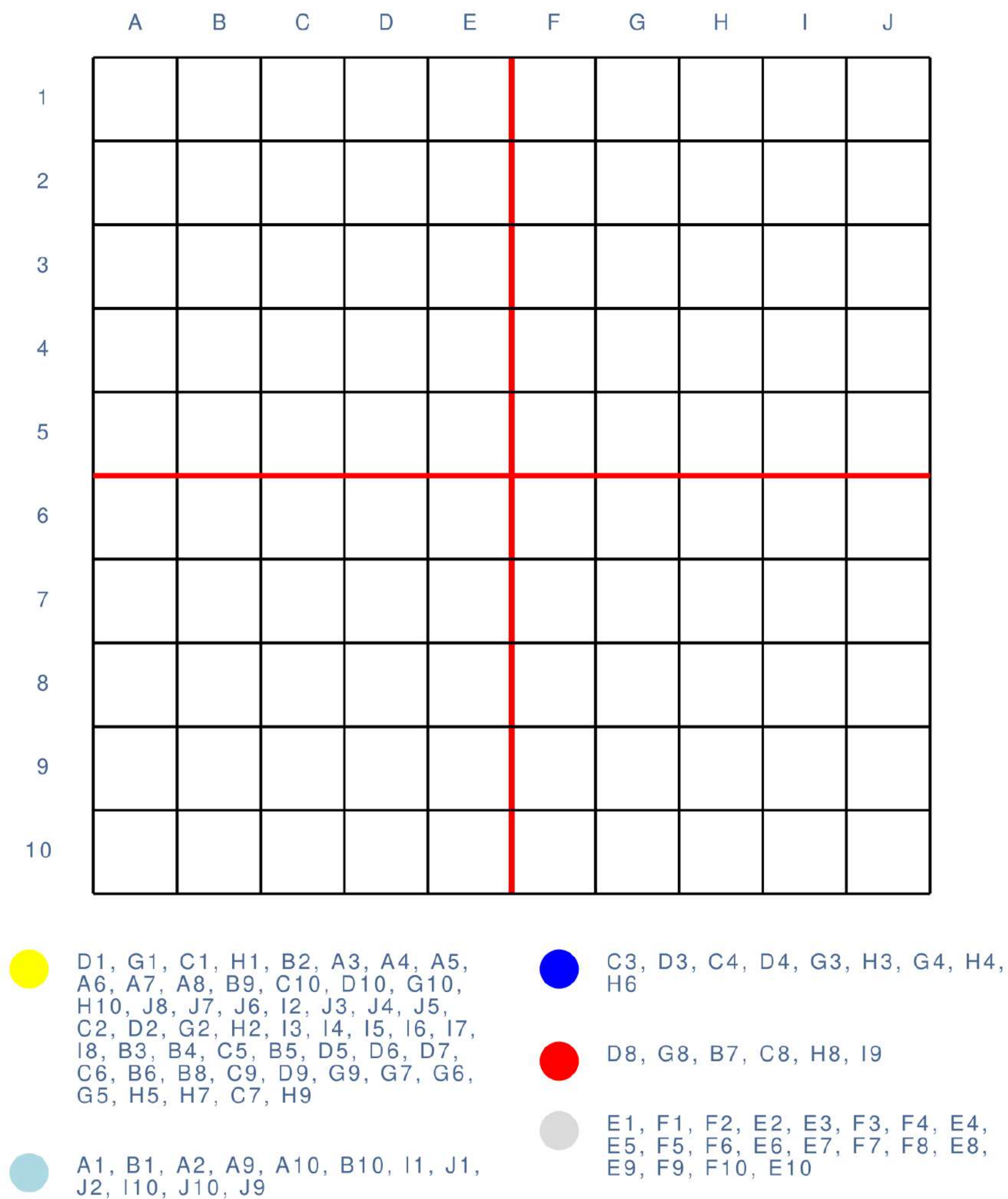
Wprowadzenie:

Nauczyciel wita się z dziećmi. Uczniowie siedzą z nauczycielem w kole, nauczyciel prosi, żeby chętne dzieci opowiedziały o tym, jak się dzisiaj czują. Wypowiada się ten uczeń, który trzyma w rękach totem emocji (totem emocji można wykonać z różnych materiałów – może to być zwykły patyk z przywiązanymi kolorowymi wstążkami). Następnie prosi uczniów, aby powiedziały coś miłego osobie po prawej stronie oraz zachęca uczniów do wykonania pierwszego zadania, dzięki któremu poznają cel zajęć. Nauczyciel rozkłada matę do kodowania, obok maty stawia kolorowe kubki i kładzie kartkę z rozpisanymi współrzędnymi. Prosi dzieci, żeby na ich podstawie ułożyły na macie do kodowania obrazek.

Rysunek 1. Mata do kodowania wraz z podanymi współrzędnymi

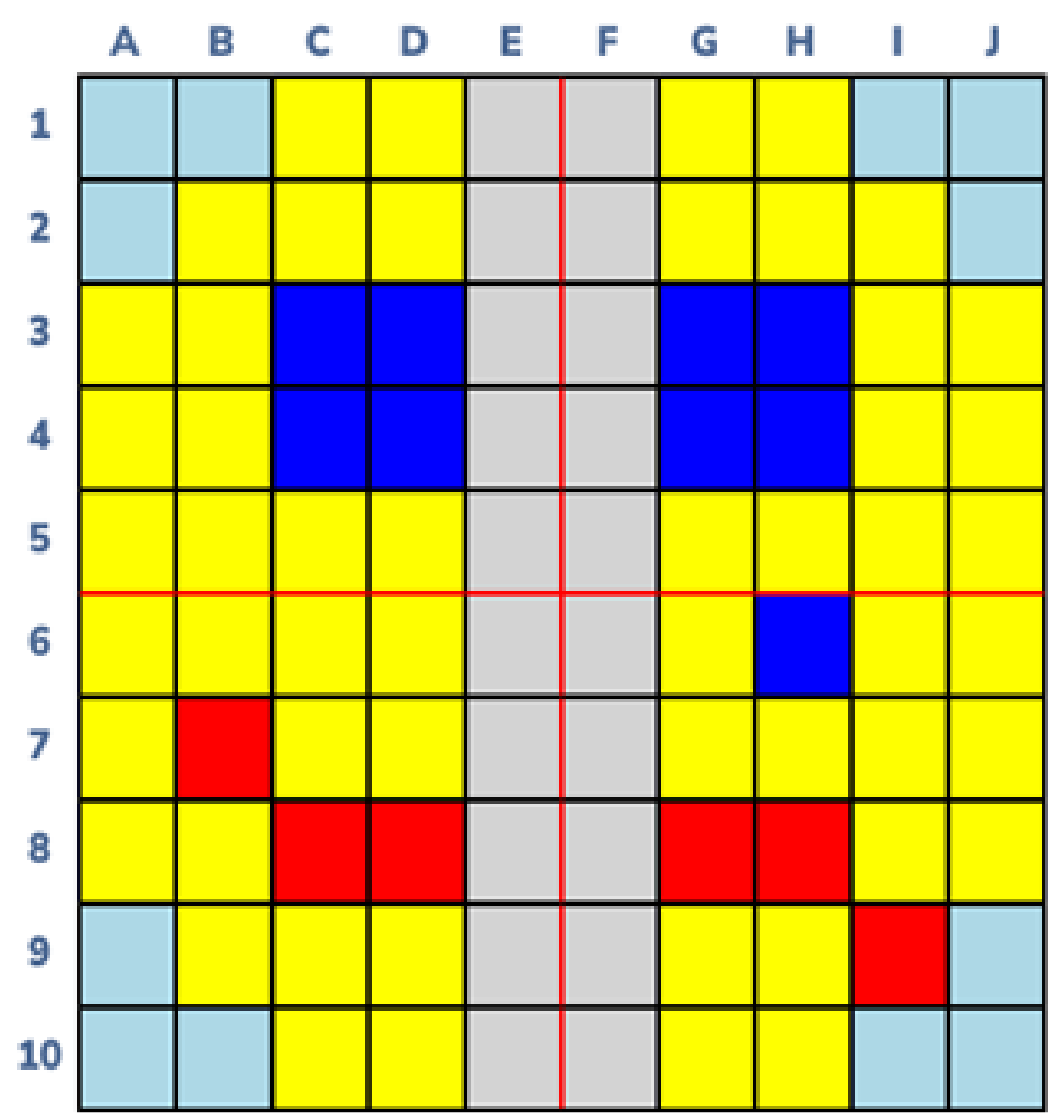


www.kodowanienadywanie.pl



Źródło: opracowanie własne A. Świć.

Rysunek 2. Obraz powstały dekodowaniu współrzędnych



Źródło: opracowanie własne A. Świć.

Po wykonaniu zadania prowadzący zajęcia pyta się jaki obrazek udało się ułożyć na macie. Na obrazku symbolicznie przedstawione są dwie emocje: radość i smutek – celem zajęć będzie wspólna refleksja na temat emocji, które towarzyszą nam na co dzień.

Część właściwa:

Nauczyciel prezentuje uczniom krążki z emocjami (drukuję je w trzech egzemplarzach) – rys. 3). Nauczyciel prosi uczniów, aby każdy zastanowił się nad tym, jakie emocje są pokazane na obrazkach. Następnie ten, kto trzyma totem emocji może nazwać wybraną przez siebie emocję. Na początek nauczyciel proponuje dzieciom ćwiczenie na pamięć, takie które pozwoli jeszcze raz dobrze przyjrzeć się grafikom widocznym na krążkach. Będzie to memory, tylko w zmienionej, utrudnionej wersji. Zasady będą takie same jak w klasycznej wersji, ale zamiast szukać dwóch takich samych obrazków, uczniowie będą szukać trzech (uczniowie mogą grać całym zespołem klasowym lub w podziale na mniejsze grupy).

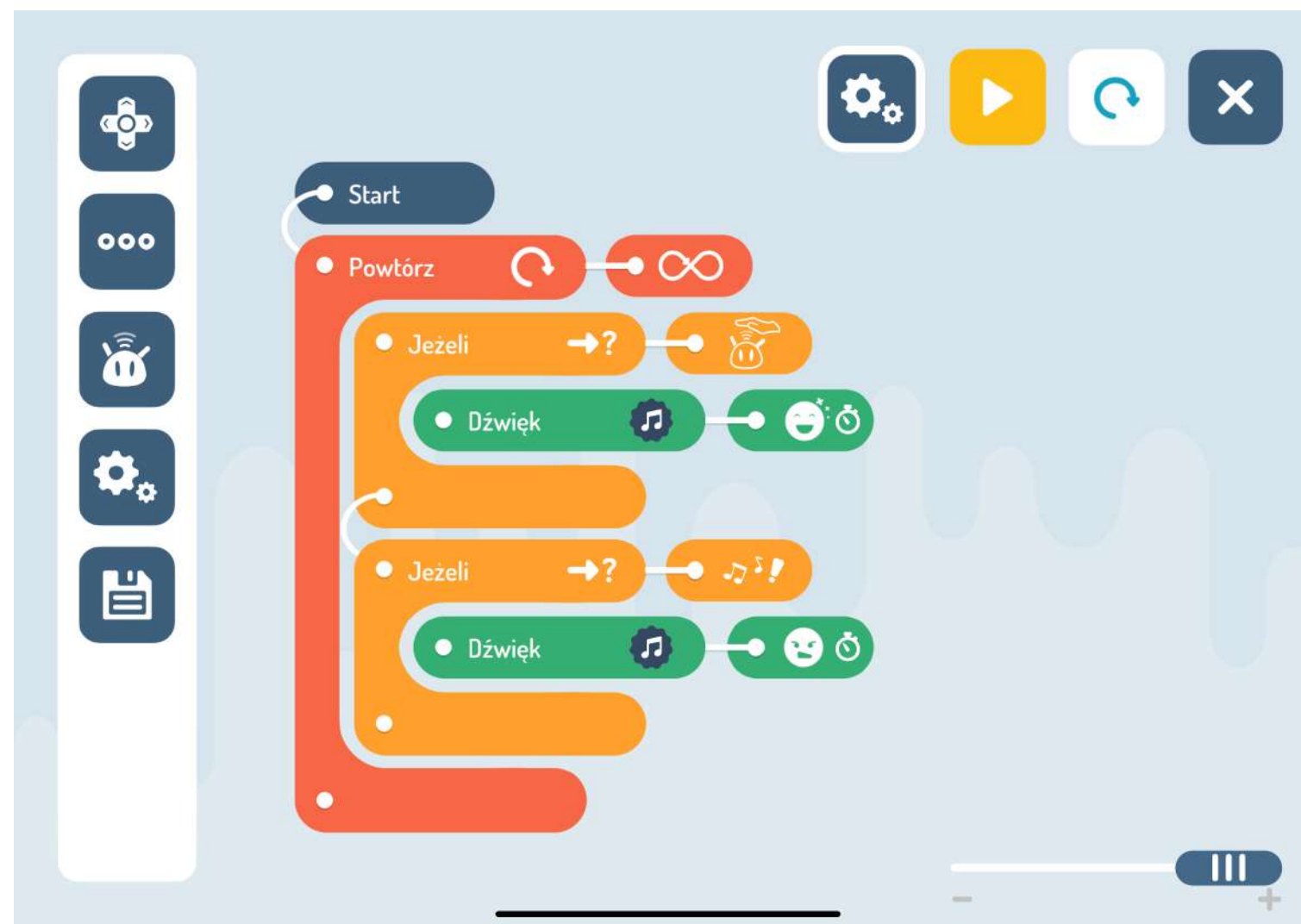
Rysunek 3. Memory – emocje



Źródło: opracowanie własne A. Świć.

Prowadzący zajęcia pokazuje dzieciom aplikację Photon Coding Blocks i wyjaśnia, że można uwarunkować zachowanie robota tym, co my zrobimy. Potrzebne będą do tego czujniki np.: światła, odległości, dotyku, hałasu. Możemy tak zaprogramować robota, żeby wykonał jakąś czynność dopiero kiedy go np.: dotkniemy po głowie lub głośno kłaśniemy. Wykorzystamy taką możliwość do pokazania emocji. Zaprogramujemy robota w taki sposób, żeby okazywał radość za każdym razem, gdy zostanie pogłaskany i okazywał strach za każdym razem, gdy usłyszy hałas. Do stworzenia programu potrzebne będą puzzle „jeżeli... to”. Uczniowie pracują w mniejszych grupach.

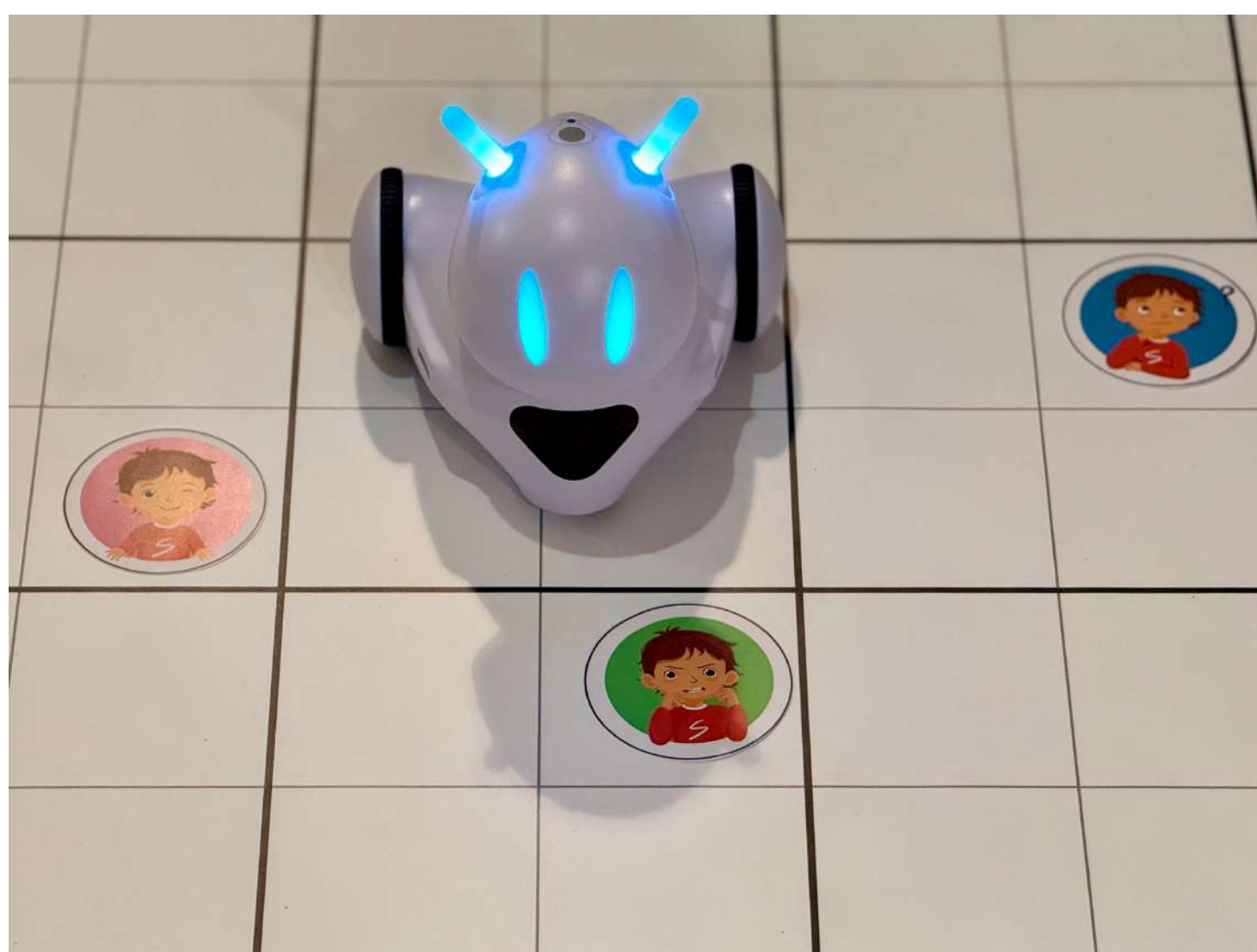
Rysunek 4. Interfejs Photon Coding Blocks



Źródło: opracowanie własne A. Świć.

Nauczyciel zaprasza uczniów do wykonania kolejnego zadania. Rozkłada na macie krążki z emocjami (w losowy sposób) i prosi o takie zaprogramowanie robota, żeby podjechał do każdej emocji i pokazał ją za pomocą dźwięku i koloru. Dzieci mogą wykonać zadanie wykorzystując dowolny interfejs (Photon Blocks, Photon Badge).

Rysunek 5. Robot pokazujący emocje.



Źródło: opracowanie własne A. Świć.

Zakończenie zajęć:

Nauczyciel wraz z uczniami siadają w kole i grają w kalambury. Na początek uczniowie pokazują emocje, a reszta klasy odgaduje hasła (odpowiada ta osoba, która pierwsza podniesie rękę). W dalszej części zajęć, nauczyciel prosi ucznia, który pokazuje, aby pokazał sytuację, która wzbudza daną emocję, np. radość, strach itp. (emocję zna tylko uczeń pokazujący). Pozostali uczniowie próbują odgadnąć co to za sytuacja i o jaką emocję chodziło.

Na zakończenie nauczyciel prosi, aby uczniowie odpowiedzieli na kilka pytań, np. łatwo rozmawia mi się o emocjach, lubię rozmawiać o emocjach, lubię rozmawiać o emocjach, które są dla mnie miłe; lubię dzielić się z bliskimi przykrymi emocjami; mam bliskich, którzy są dla mnie wsparciem. Jeśli uczniowie nie zgadzają się ze stwierdzeniem – kucają; jeśli ich odpowiedź to „jest mi to obojętne/ ani tak, ani nie” – stoją prosto; jeśli zdecydowanie zgadzają się ze stwierdzeniem – stają na palcach i wyciągają ręce w górę.





ZAPROGRAMOWANI NA MATEMATYKĘ

Co łączy matematykę i programowanie? Czy praca z robotami może pomóc zrozumieć różne pojęcia matematyczne? Na zajęciach przeprowadzonych według zaproponowanego scenariusza uczniowie przy pomocy robotów sprawdzą właściwości wybranych figur geometrycznych.

Klasa: 1–3

Podstawa programowa:

I.1.1), I.1.2), II.1.1), II.1.3), II.5.1), II.6.9), III.1.10), IV.2.9), VI.1.1), VI.1.2), VII.1.1), VII.1.2), VII.1.3), VII.4.1), VII.5.1)

Kompetencje kluczowe:

- Kompetencje w zakresie rozumienia i tworzenia informacji;
- Kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii;
- Kompetencje cyfrowe.

Cele ogólne:

- Poszerzenie wiadomości na temat właściwości figur geometrycznych;
- Rozwijanie logicznego algorytmicznego myślenia;
- Rozwijanie zadaniowego podejścia do stawianych problemów;
- Wprowadzanie w świat programowania blokowego;
- Bezpieczne wprowadzanie w cyfrowy świat;
- Rozwijanie miękkich kompetencji, ze szczególnym uwzględnieniem umiejętności pracy zespołowej.

Cele operacyjne:

Uczeń podczas zajęć:

- wyjaśnia dlaczego każdy kwadrat jest prostokątem, ale nie każdy prostokąt jest kwadratem;
- rysuje kwadrat o podanym wymiarze;
- programuje robota Photon wykorzystując interfejs Photon Badge;
- oblicza obwody kwadratów i prostokątów;
- twórczo rozwiązuje problemy;
- chętnie pracuje zespołowo.

Formy organizacyjne:

- grupowa
- zespołowa
- indywidualna

Środki dydaktyczne:

Mata do kodowania, kolorowe kubki, taśma papierowa (lub inna łatwa do usunięcia), roboty edukacyjne Photon, tablet/smartfon, materiały pomocnicze dostępne na końcu scenariusza

Czas prowadzenia zajęć:

Ok. 45 – 90 minut

Wprowadzenie:

Nauczyciel wita się z uczniami i przedstawia cel zajęć: poszerzenie wiedzy na temat właściwości figur geometrycznych. Nauczyciel zaprasza dzieci do zabawy. Jedno z dzieci otrzymuje przedmiot, który będzie wędrował między uczniami (np.: piłka, woreczek gimnastyczny). Nauczyciel mówi nazwę wybranej figury geometrycznej i podaje wybranej osobie wędrujący przedmiot. Osoba, która go otrzyma wskazuje jakąś rzecz, która ma kształt wymienionej przez nauczyciela figury geometrycznej. Następnie mówi inną figurę i przekazuje wybranej koleżance lub koledze wędrujący przedmiot. Osoba, która go dostanie wskazuje rzecz o kształcie figury wskazanej przez osobę, od której otrzymała wędrujący przedmiot i podane swoją figurę... zabawę kilkakrotnie powtarzamy.

Część właściwa

Nauczyciel, przy pomocy taśmy wykleja w kilku miejscach kwadraty i prostokąty (figury muszą być takiej wielkości, żeby można było chodzić po ich obwodzie). Wybiera jedno dziecko, które będzie poruszało się po jednym kwadracie

Jak wybrać ucznia?

można podać do rozwiązania działanie, którego rozwiązanie wskaże numer z dziennika – uczeń, któremu przypisany jest numer zostaje wybrany, np. $12+5$; $39-21$, 4×3 itp.).

Pozostali uczniowie będą wydawać polecenia koledze lub koleżance, na podstawie których możliwe będzie poruszanie się po obwodzie wyklejonej figury. Jeśli kwadrat nie będzie zbyt dużych rozmiarów, to polecenie będzie brzmiało: idź krok do przodu, obróć się w prawo, idź krok do przodu, obróć się w prawo, idź krok do przodu, obróć się w prawo, idź krok do przodu, obróć się w prawo (można też jako miarę stosować stopę – dwie stopy do przodu, obróć się w prawo...).

Nauczyciel tłumaczy dzieciom, że właśnie zaprogramowały kolegę/koleżankę na poruszanie się po kwadracie. Pyta, czy w stworzonym programie jest coś szczególnego (powtarza się sekwencja następujących komend: idź do przodu, obróć się w prawo). W tej sytuacji możemy zastosować pętlę podczas programowania.

W kolejnym kroku inna osoba (znów można ją wybrać przez rozwiązanie działania) będzie poruszała się po sporo większym kwadracie. Program będzie zbliżony, tylko liczba kroków będzie większa. Nie zmieni się natomiast to, że sekwencja komend odpowiedzialna za przejście po jednym boku figury zostanie czterokrotnie powtórzona.

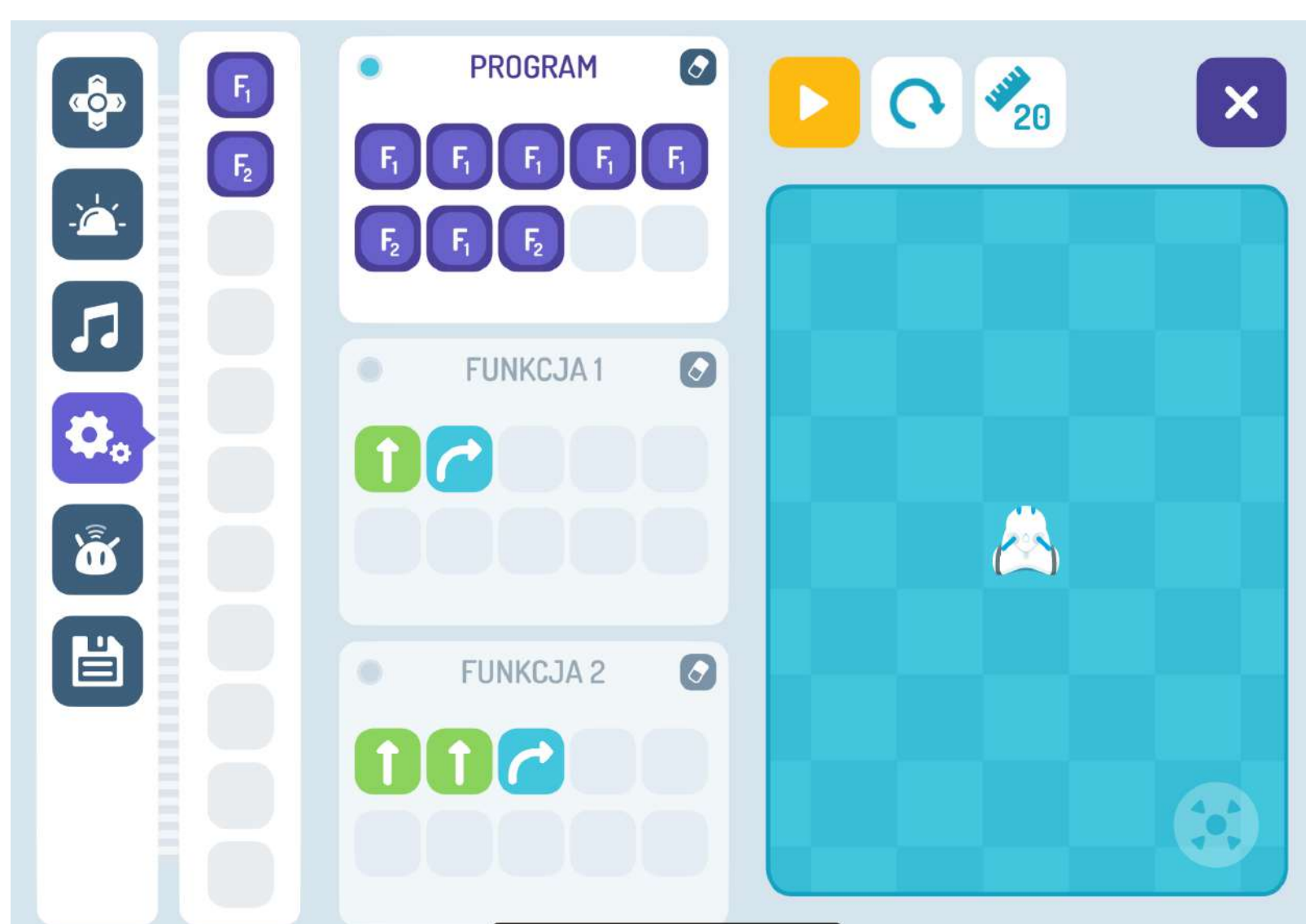
Nauczyciel wybiera kolejne dziecko, które tym razem przejdzie po prostokącie. Zanim uczniowie zaczną wydawać komendy umożliwiające koledze/koleżance poruszanie się po figurze zapyta dzieci, czy ich zdaniem program będzie podobny do poprzednich i też wystąpi sekwencja komend, którą należy czterokrotnie powtórzyć – słuchamy hipotez uczniów, można je testować na bieżąco.

W przypadku prostokątów program będzie wyglądał inaczej, bo boki nie są takiej samej długości. Na dwóch bokach wystarczy wykonać np.: jeden krok, ale na dwóch bokach, konieczne będą dwa kroki. W tym przypadku program może wyglądać następująco: idź krok do przodu, obróć się w prawo, idź dwa kroki do przodu, obróć się w prawo, idź krok do przodu, obróć się w prawo, idź dwa kroki do przodu, obróć się w prawo. Tu również występuje powtórzenie sekwencji komend, ale powtarzana sekwencja będzie dłuższa, a liczba powtórzeń mniejsza.

W dalszej części zajęć uczniowie zaprogramują roboty w taki sposób, żeby najpierw narysowały kwadrat, a następnie prostokąt. Krótszy bok w prostokącie będzie miał taki sam wymiar jak bok w kwadracie, dłuższy bok w prostokącie będzie podwojeniem krótszego boku.

Do wykonania zadania wykorzystany zostanie interfejs Photon Badge. W przestrzeni przeznaczonej na stworzenie programu jest za mało miejsca, więc wykorzystana zostanie „Funkcja 1” i „Funkcja 2”. Nauczyciel prosi dzieci, żeby jako „Funkcję 1” zdefiniowały narysowanie jednego boku w kwadracie plus obrót, a jako „Funkcję 2” narysowanie dłuższego boku w prostokącie plus obrót. Teraz można złożyć cały program, który pozwoli najpierw narysować kwadrat, a następnie prostokąt.

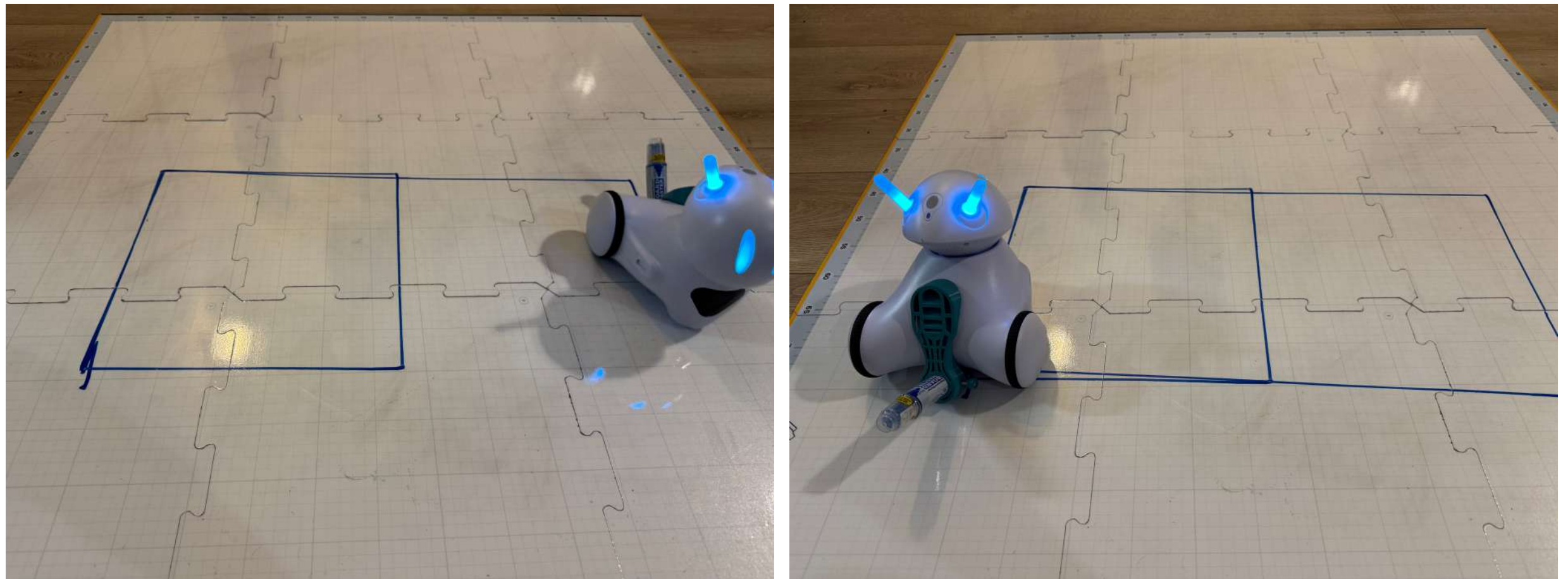
Rysunek 1. Interfejs Photon Badge



Źródło: opracowanie własne A. Świć.

Uczniowie przygotowują duży papier lub specjalną matę do rysowania, mocują flamaster do robota, a następnie dopasowują długość kroku robota do wymiarów kartki, którą przygotowali (kwadrat i prostokąt muszą zmieścić się na kartce).

Rysunek 2. Robot Photon rysujący prostokąt i kwadrat



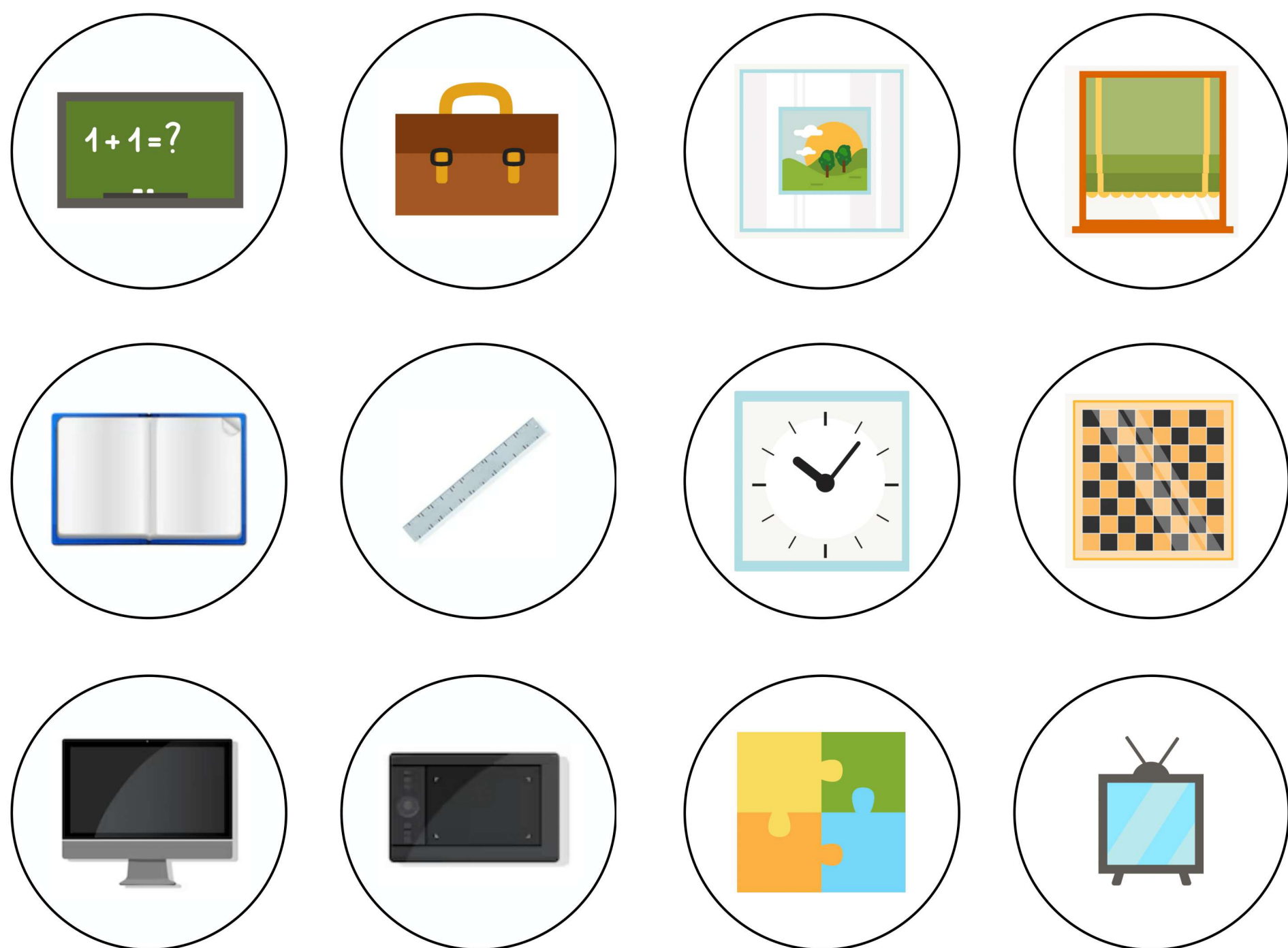
Źródło: opracowanie własne A. Świć.

Kiedy figury są już narysowane nauczyciel pokazuje dzieciom krążki z grafikami przedstawiającymi różne przedmioty (rys. 3 – załącznik „Krążki przedmioty”). Prosi, żeby uczniowie ułożyli je w kwadracie lub w prostokącie, kierując się ich kształtem. Po rozłożeniu wszystkich krążków osoba prowadząca zajęcia zwraca uwagę dzieci na to, że wszystkie kwadraty mieszczą się w prostokącie (każdy kwadrat, to prostokąt), ale prostokąty są poza kwadratem (nie każdy prostokąt jest kwadratem)

Dlaczego każdy kwadrat to prostokąt, ale nie każdy prostokąt to kwadrat?

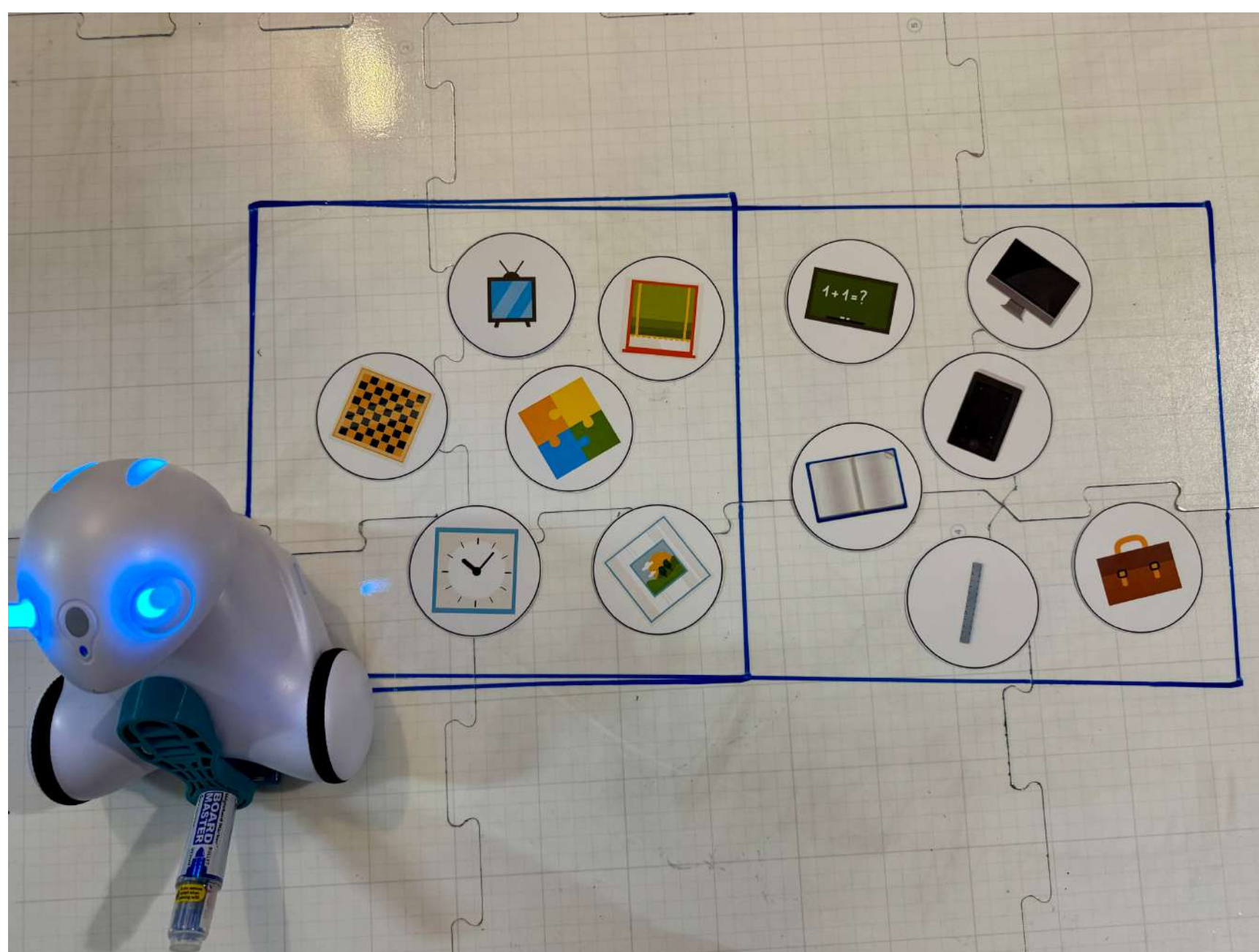
można poprosić uczniów o wyjaśnienie jak rozumieją sformułowanie „każdy kwadrat to prostokąt, ale nie każdy prostokąt to kwadrat”. Jeśli jest to zbyt skomplikowane dla uczniów, warto odnieść się do prostej historii: Dawno, dawno temu żyła sobie pewna rodzina. W tej rodzinie wszyscy mieli proste kąty i przeciwległe boki tej samej długości (warto pokazać co oznacza „przeciwległe” – leżące naprzeciw siebie). Była to rodzina prostokątów. W tej dużej rodzinie znajdowały się też takie rodziny, które miały proste kąty i przeciwległe boki tej samej długości, ale oprócz tego miały specjalną cechę – ich wszystkie boki były równe. W tej rodzinie – rodzinie prostokątów, były też inne rodziny – prostokątów, które nie są kwadratami, ponieważ nie mają tej specjalnej cechy – ich boki nie są równe, więc nie mogą być kwadratami. Dlatego też każdy kwadrat jest prostokątem, ale nie każdy prostokąt jest kwadratem.

Rysunek 3. Krążki przedmioty



Źródło: opracowanie własne A. Świć.

Rysunek 4. Podział przedmiotów prostokątnych i kwadratowych

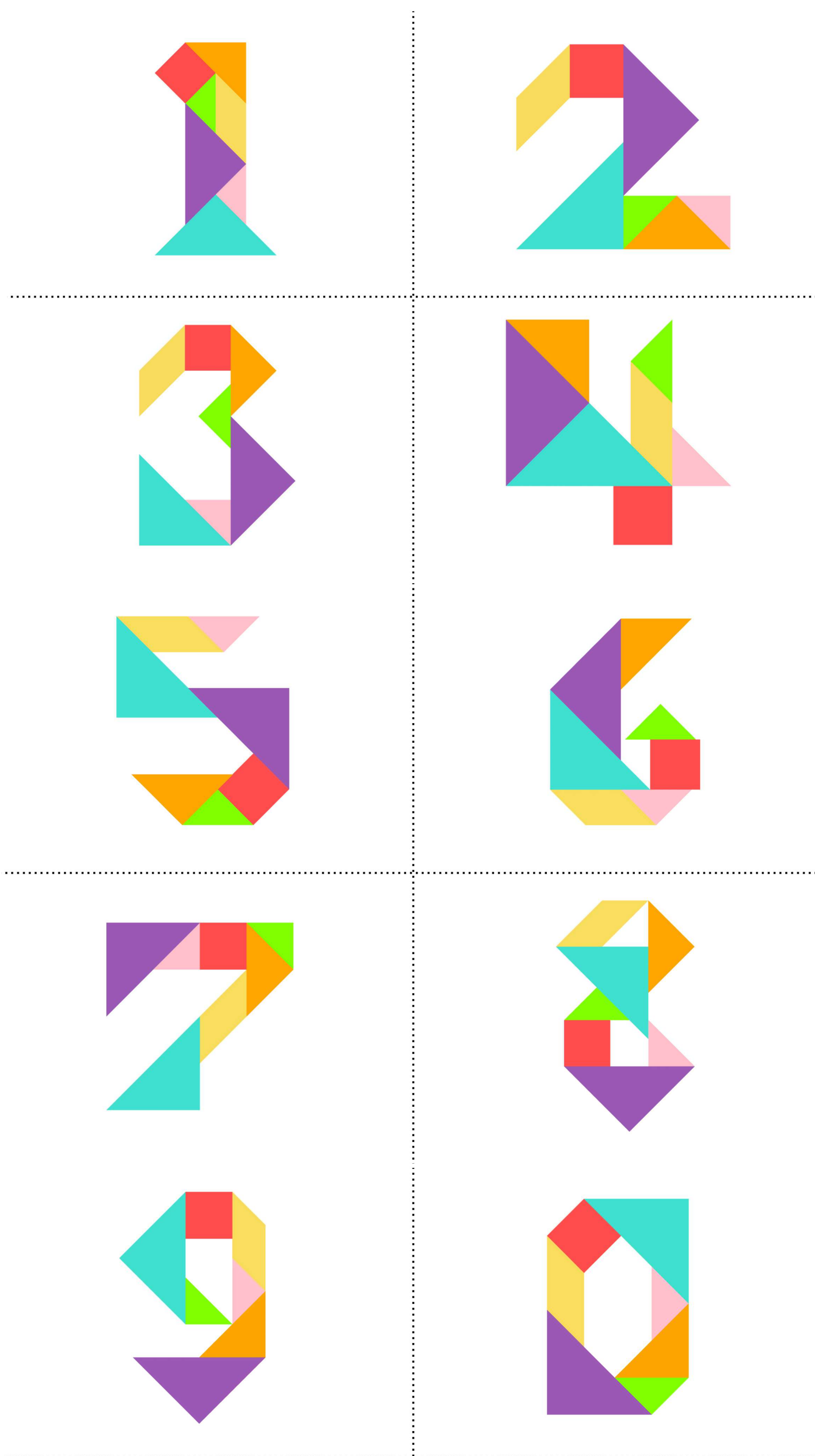


Źródło: opracowanie własne A. Świć.

Wykorzystując podziałkę na macie do rysowania (lub linijkę) uczniowie obliczają obwód narysowanych za pomocą robota figur.

Nauczyciel mówi dzieciom, że różne figury geometryczne można wykorzystać np.: do układania wzorów. Tłumaczy uczniom czym jest tangram. Dzieli uczniów na kilka grup i prosi, żeby spróbowali z otrzymanych figur (rys. 6 – “Tangram”) ułożyć wybraną przez siebie cyfrę (alternatywnie uczniowie układają cyfry na podstawie wzorów – rys. 5 załącznik „Wzory liczb – tangram”).

Rysunek 5. Wzory liczb – tangram



Źródło: opracowanie własne A. Świć.

Rysunek 6. Tangram

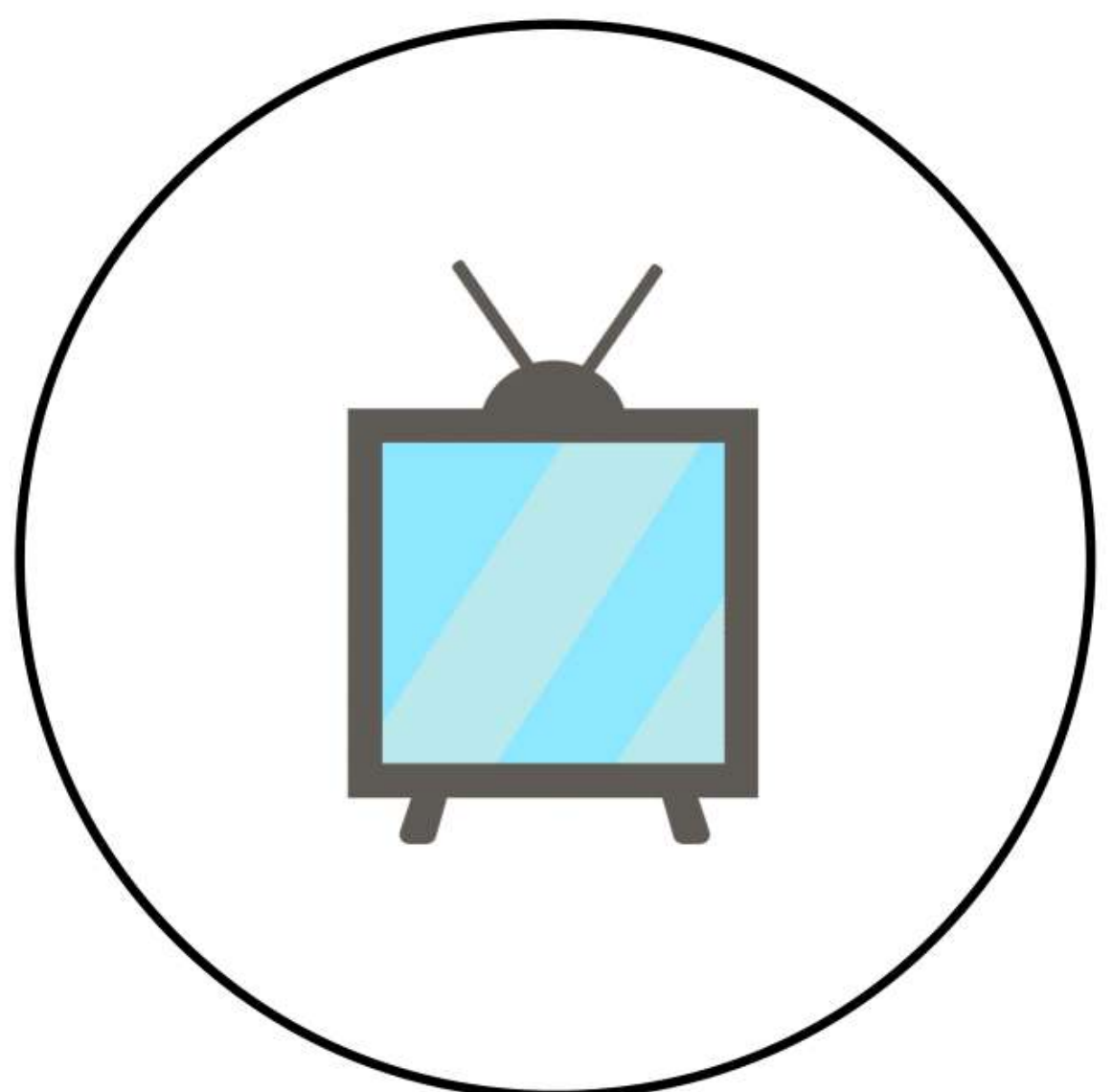
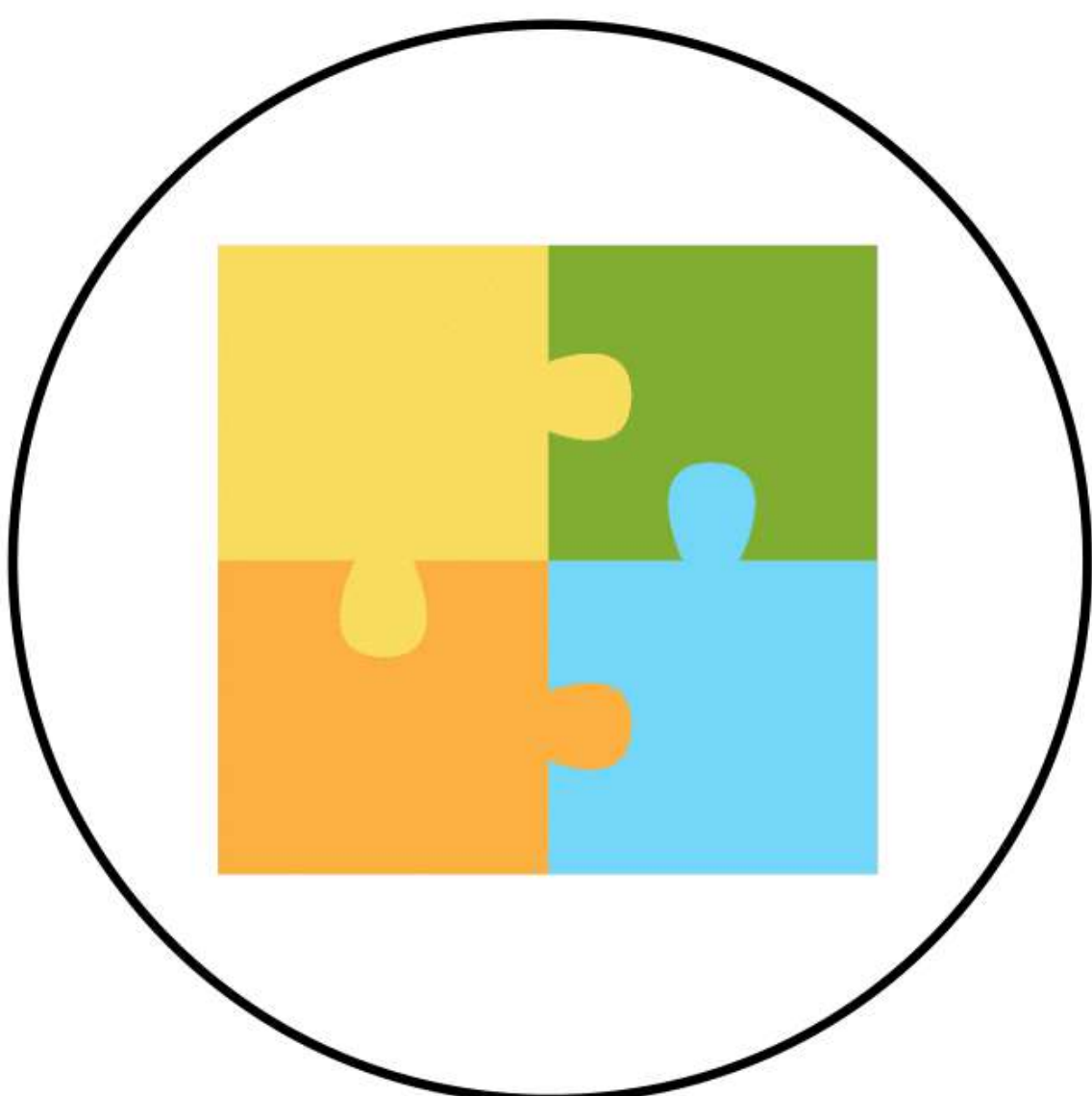
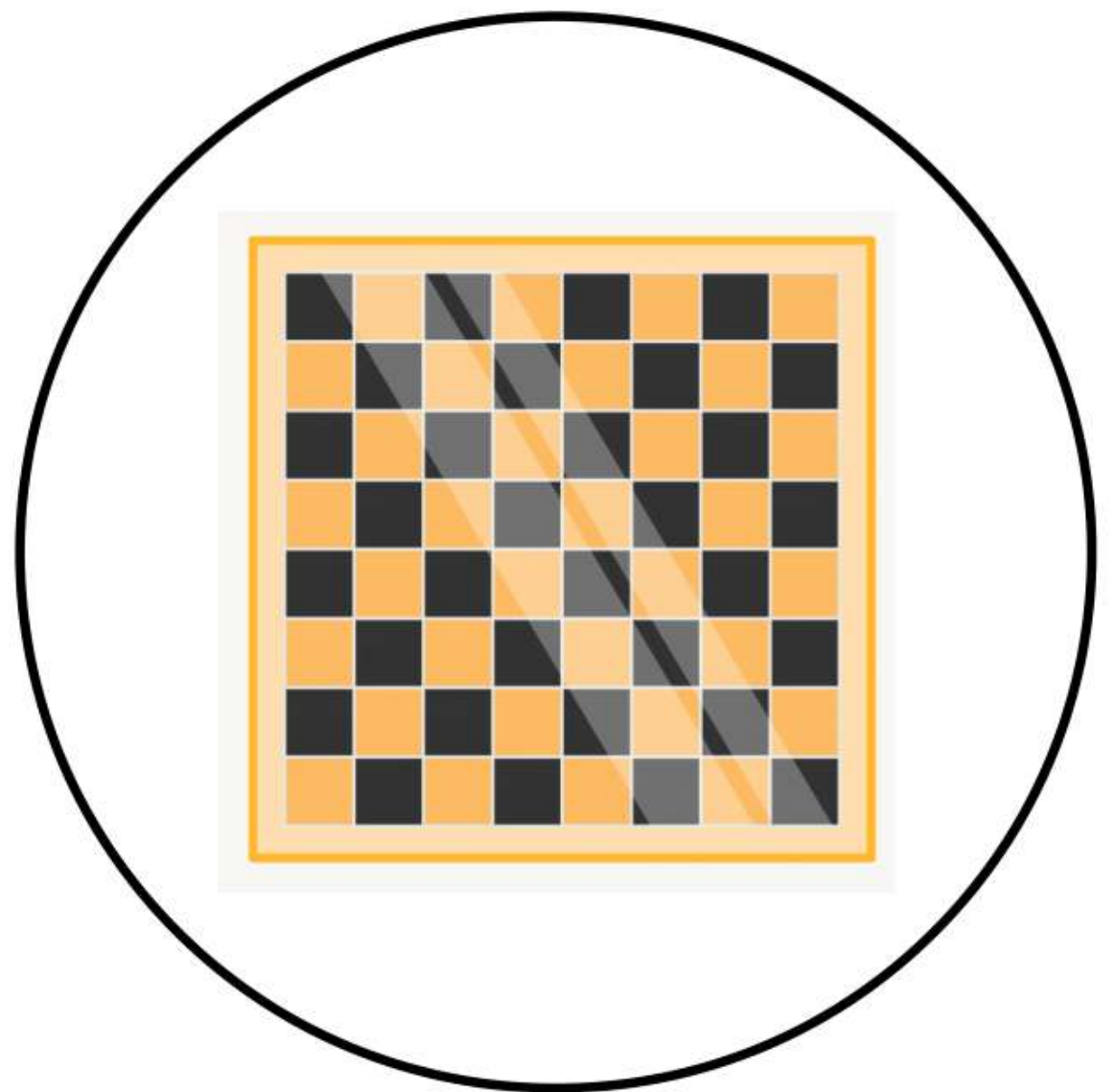
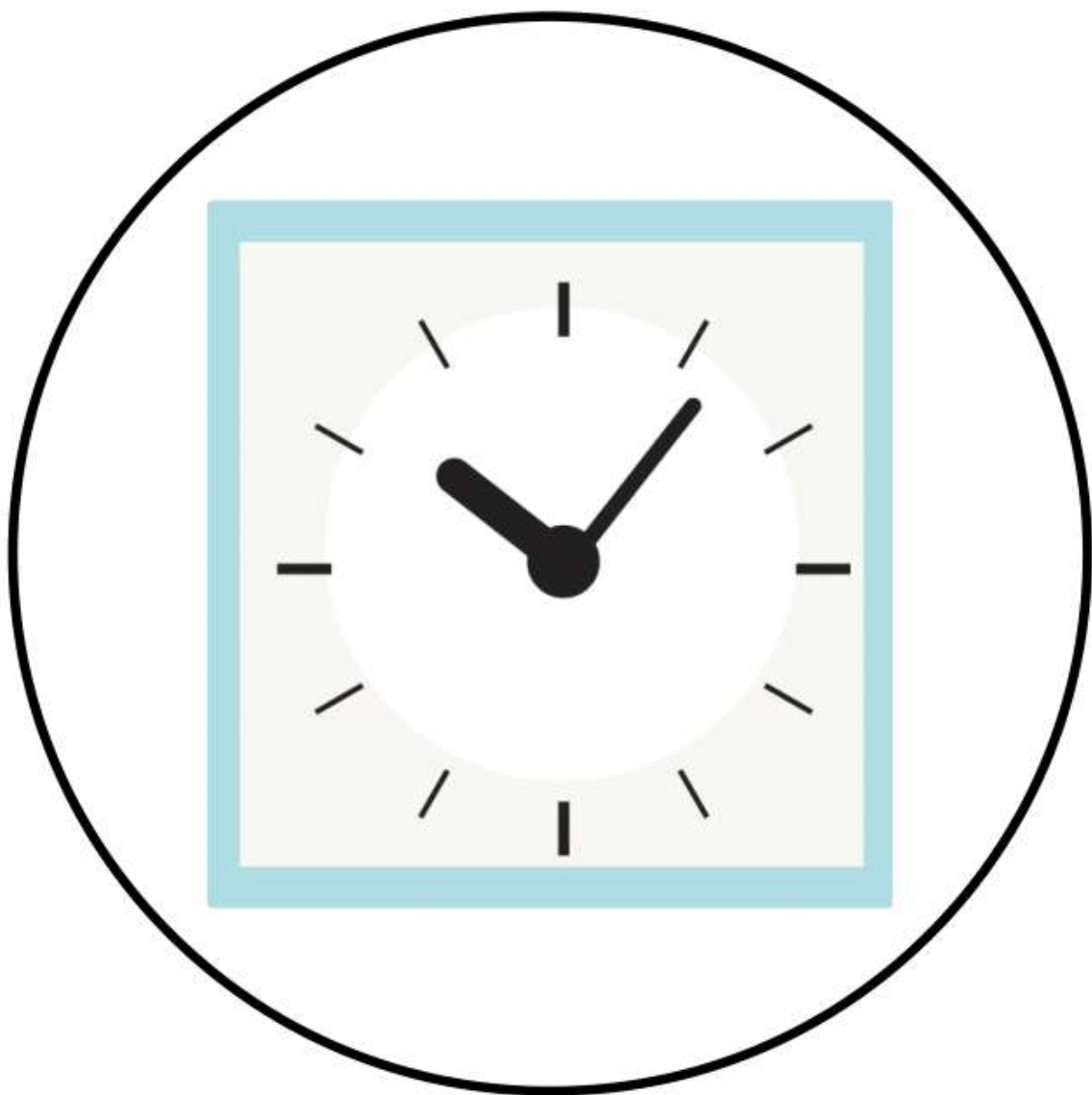
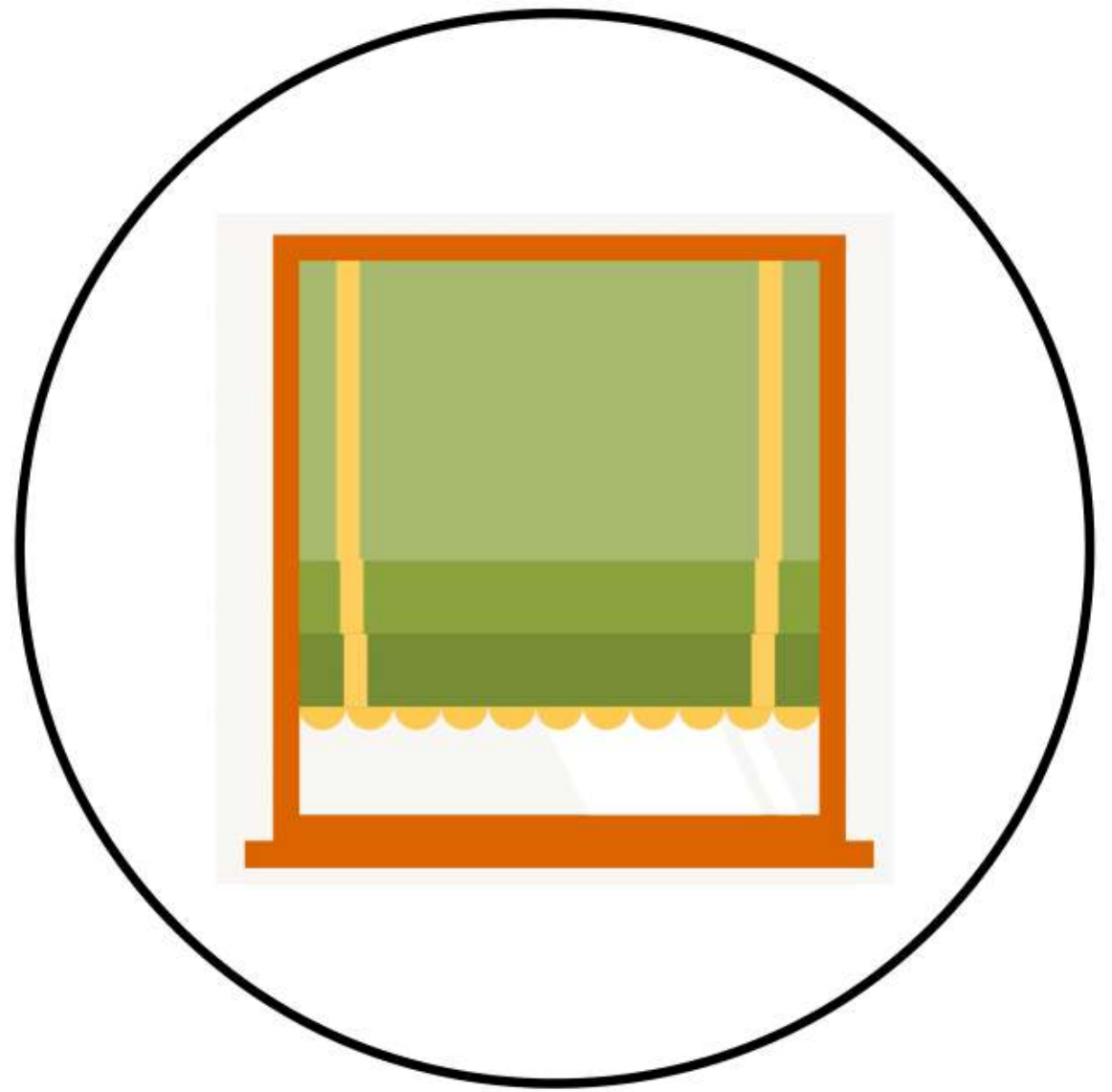
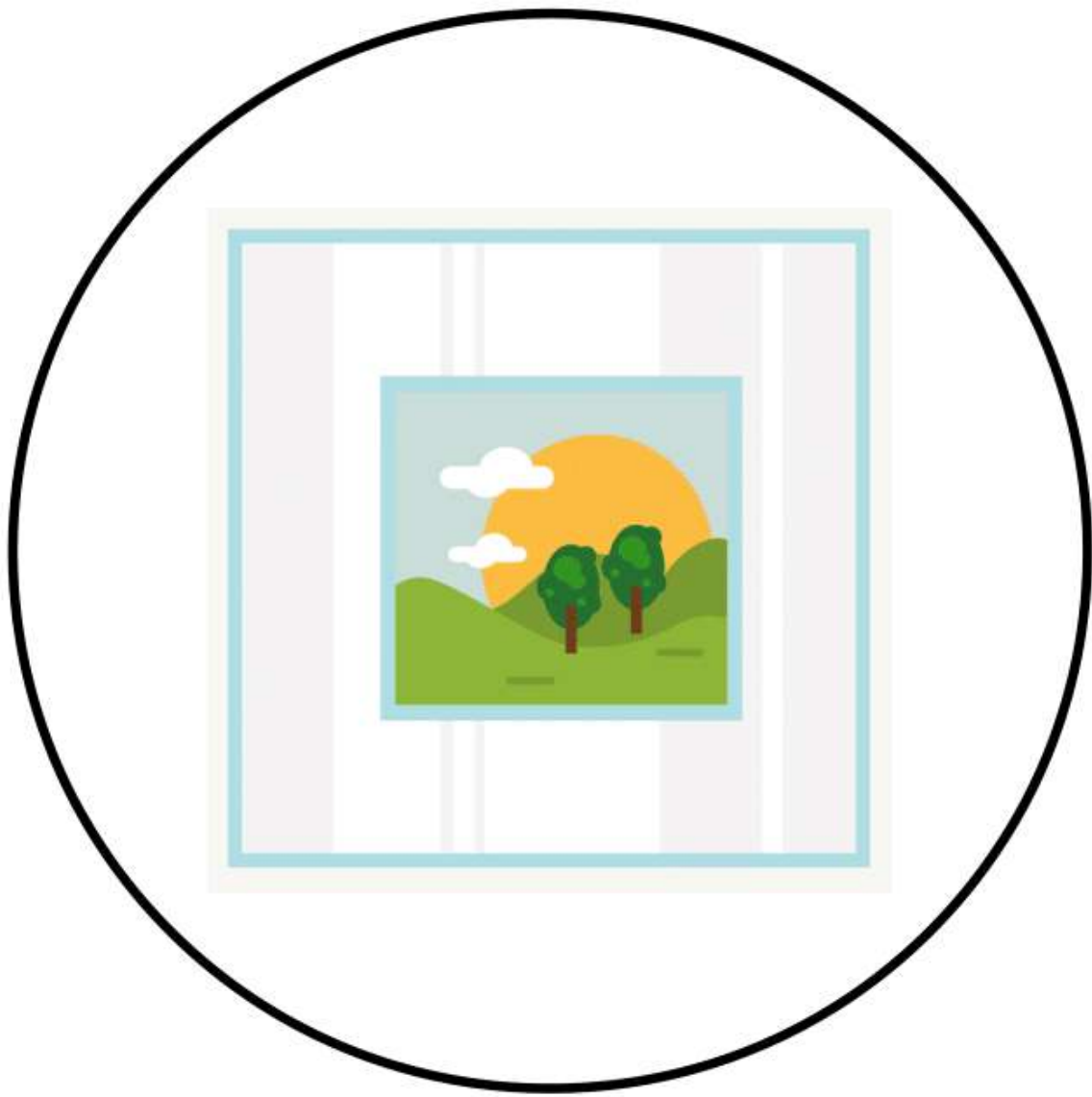


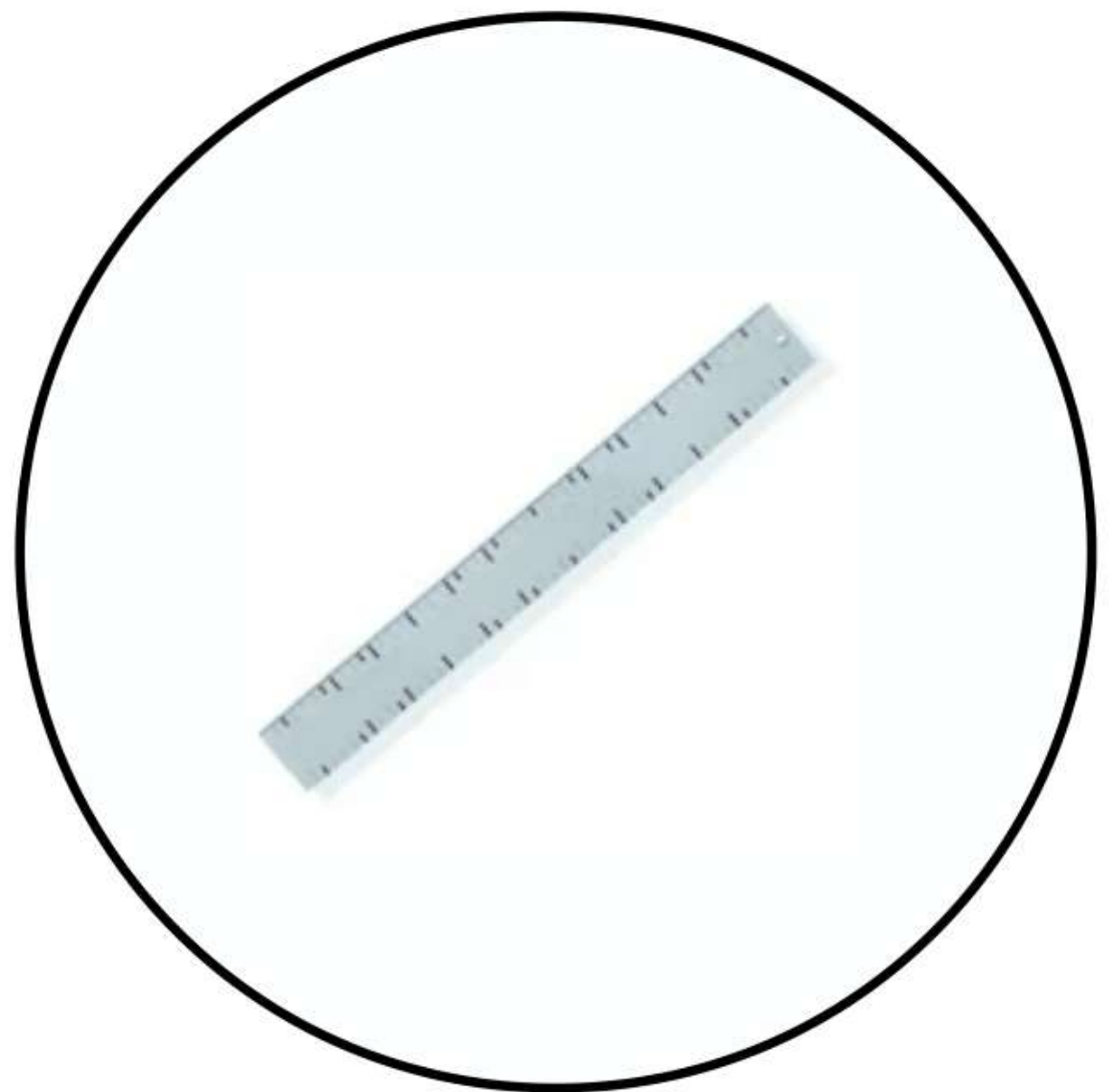
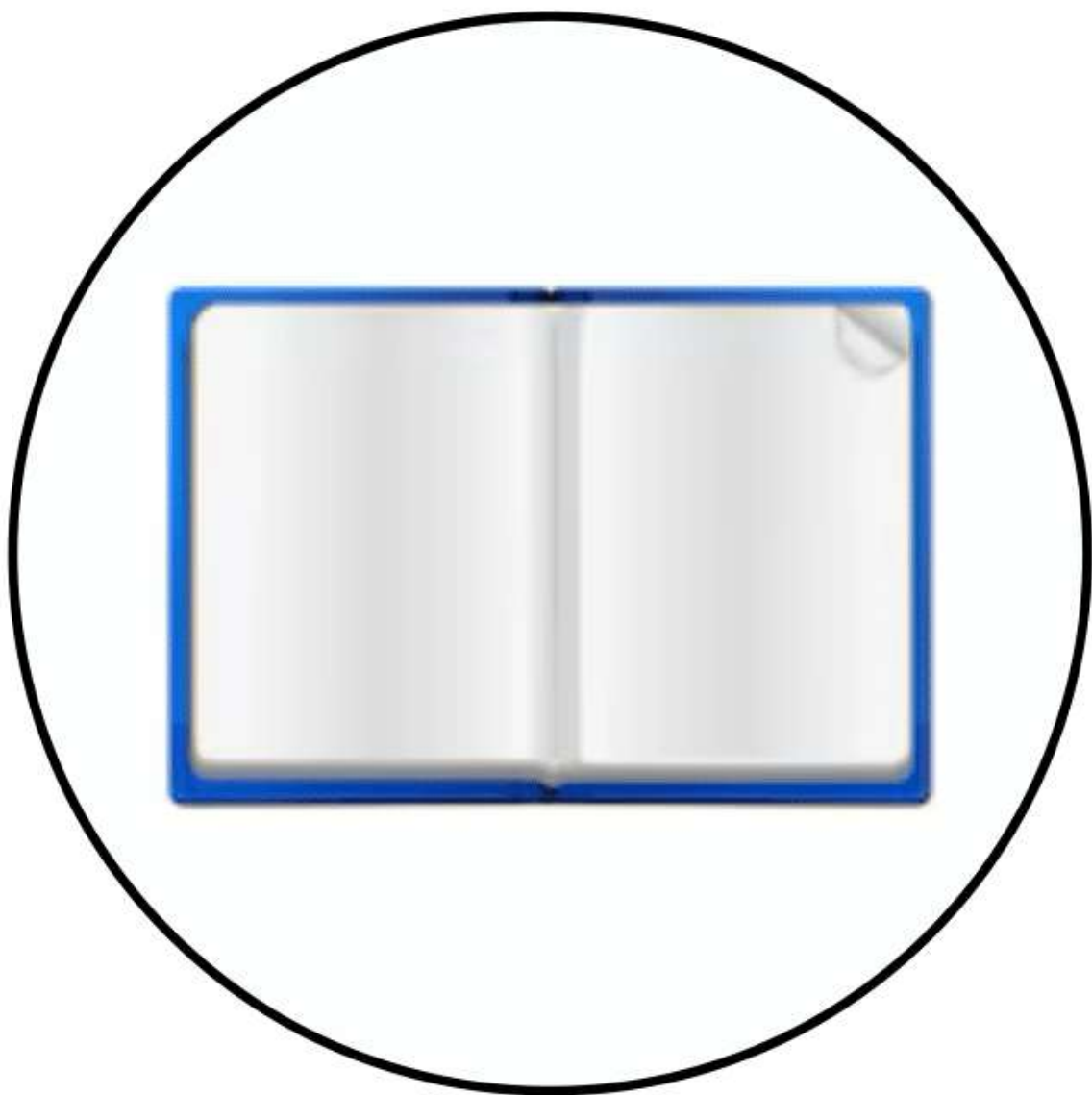
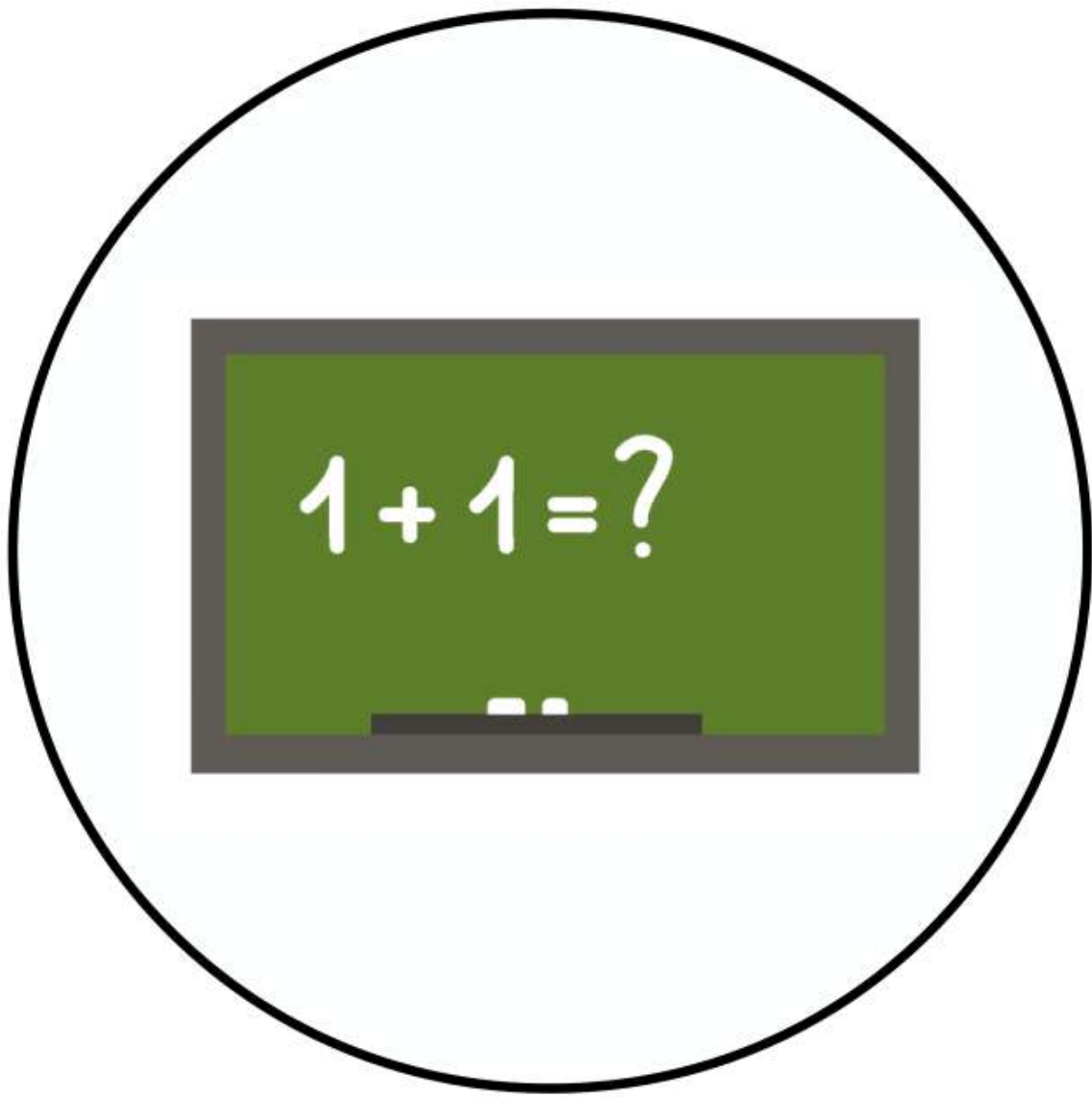
Źródło: opracowanie własne A. Świć.

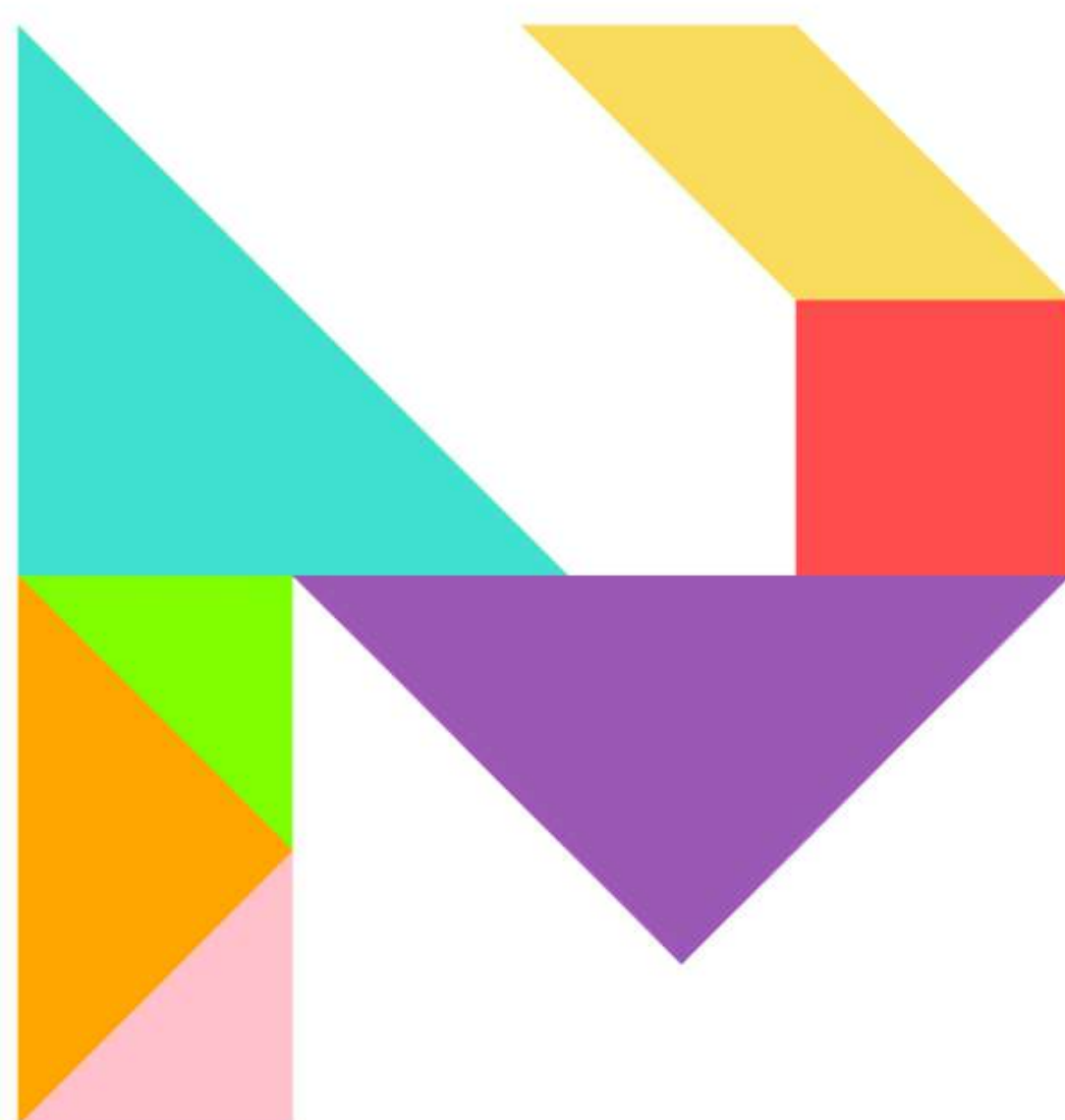
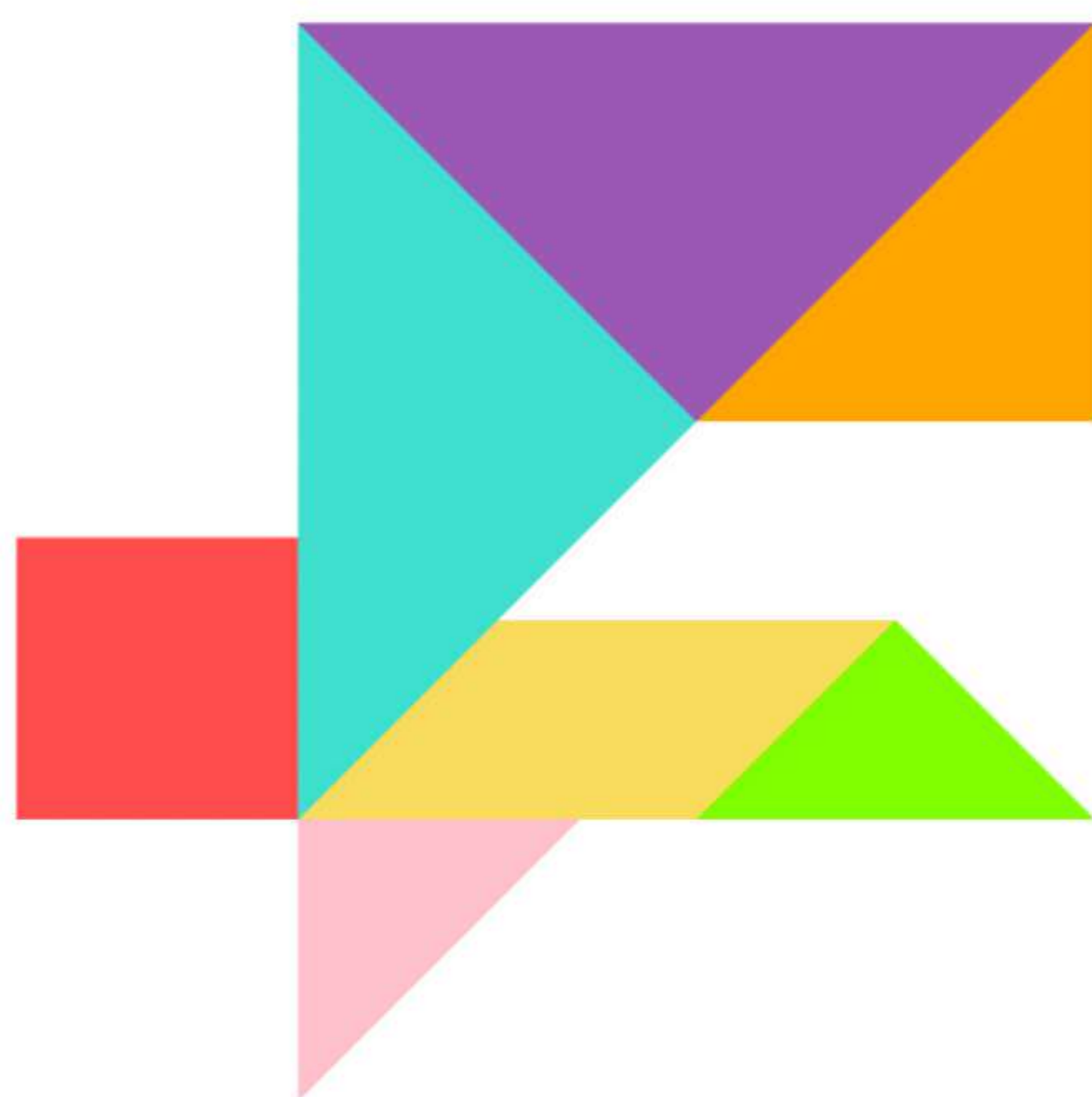
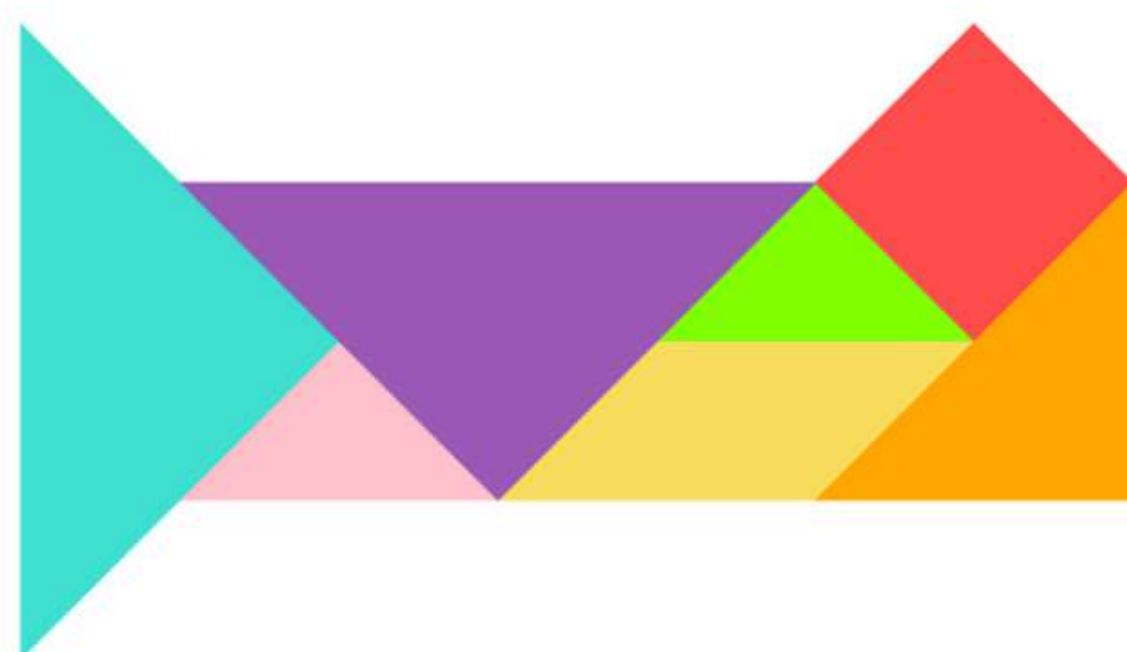
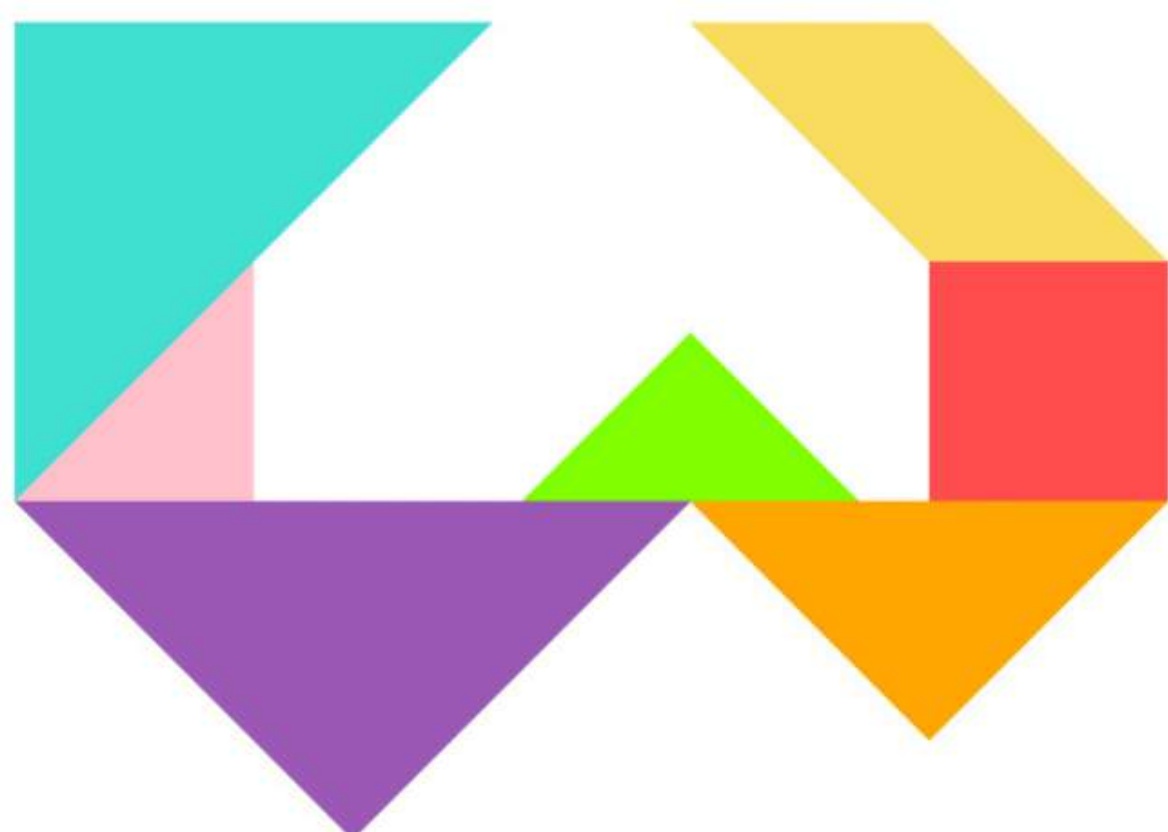
Zakończenie zajęć:

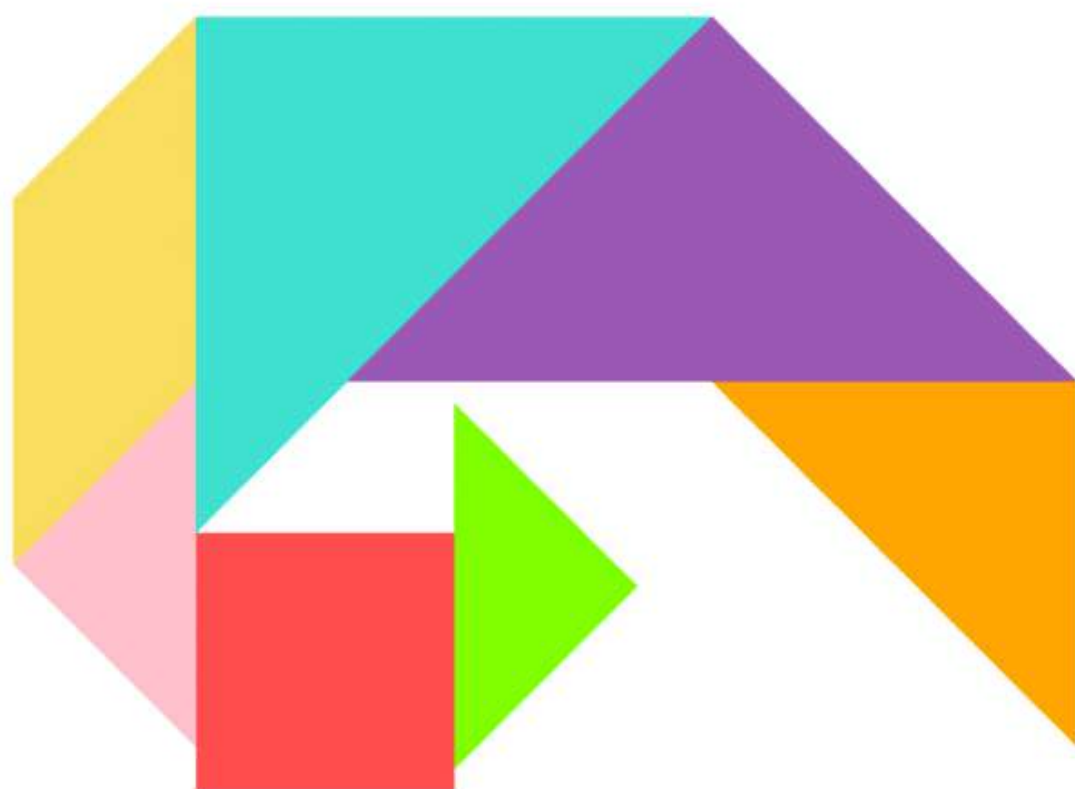
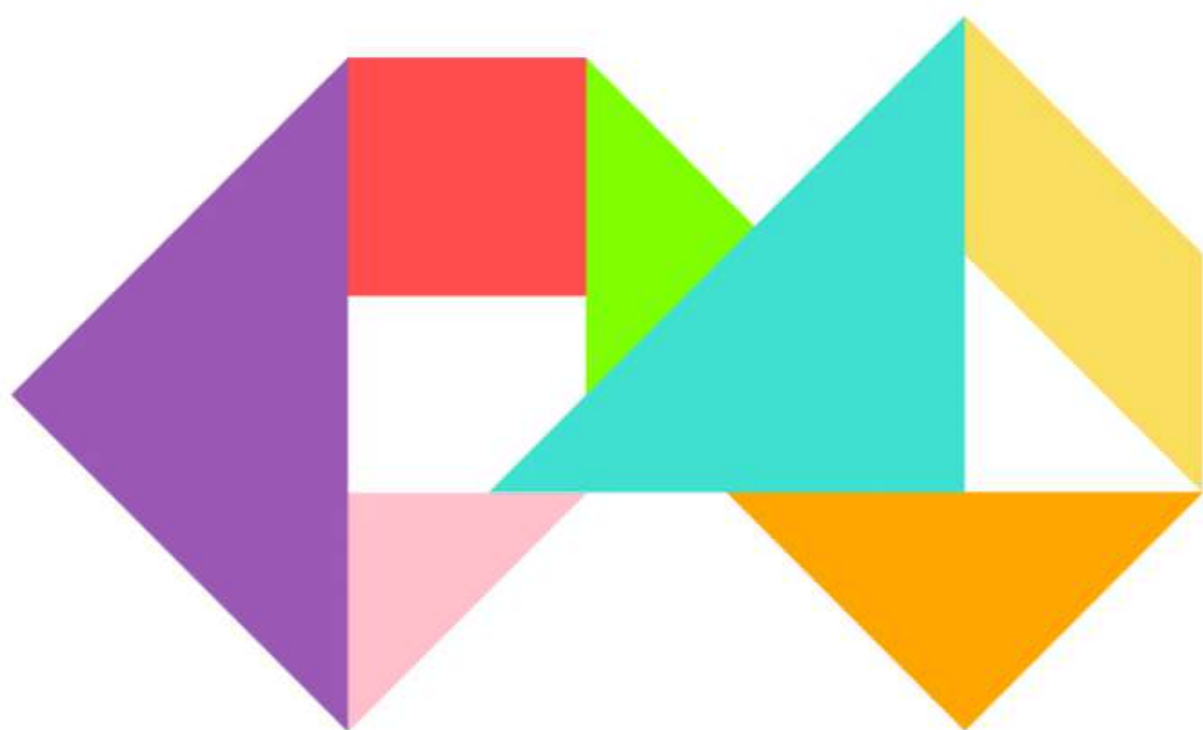
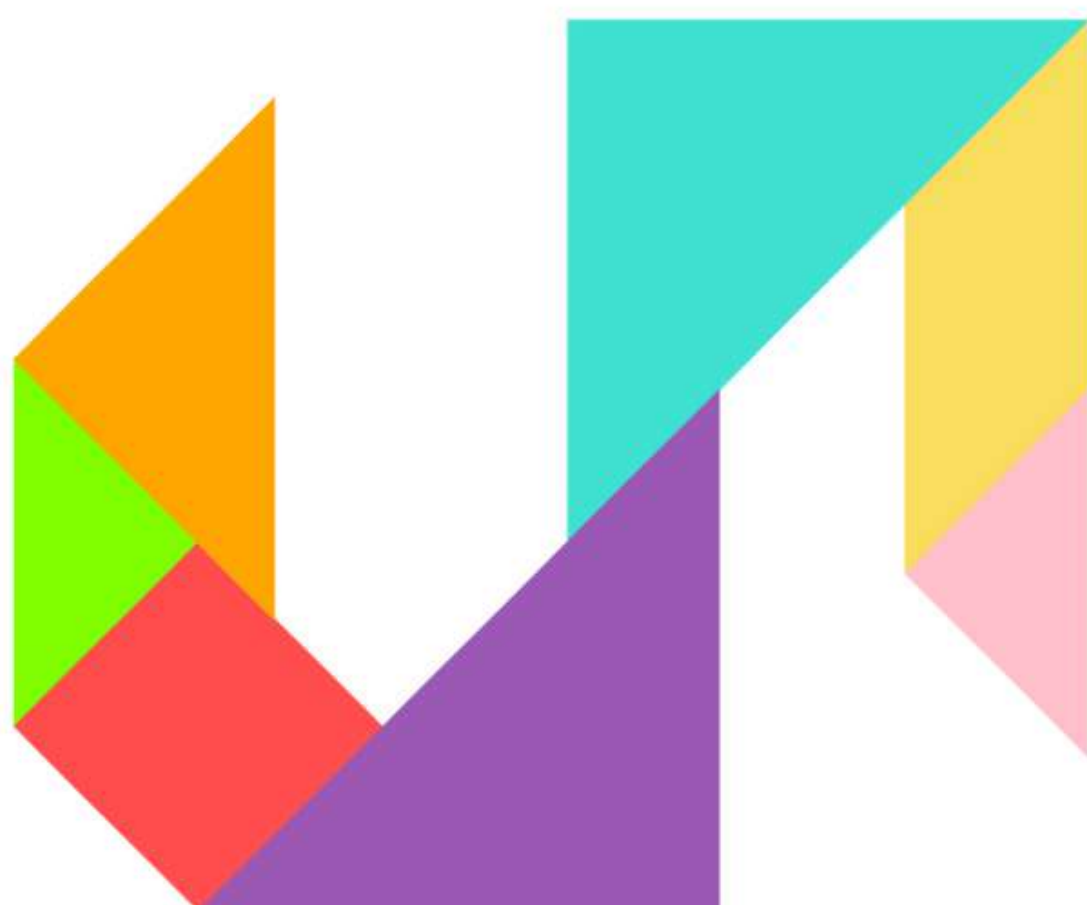
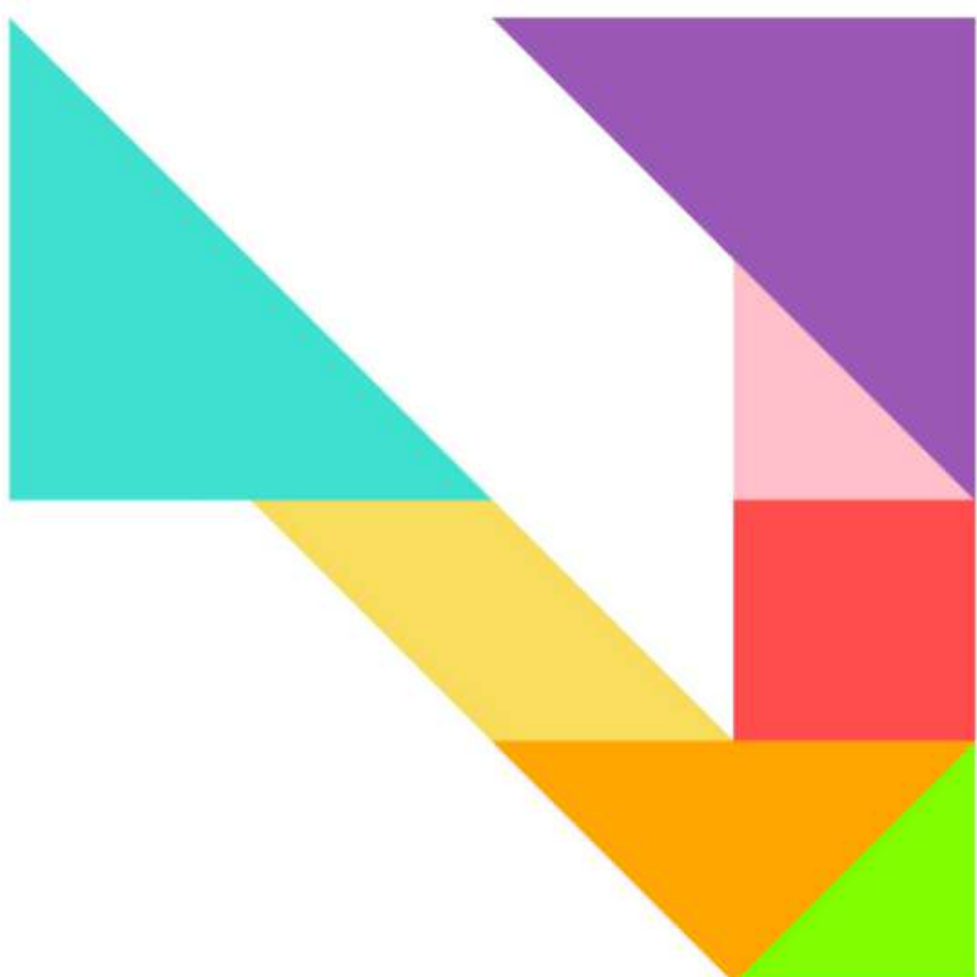
Na zakończenie zajęć uczniowie wykonają pracę plastyczną bazującą na figurach geometrycznych. To nie oni będą rysować poszczególne figury, tylko zaprogramowane przez nich roboty (Photon Badge). Każdy zespół stworzy dowolny obrazek. Po narysowaniu figur zgodnie z koncepcją wybraną przez członków zespołów obrazek zostanie wykończony dowolną techniką plastyczną. Figury można wyciąć i złożyć w całość tworząc mozaikę.

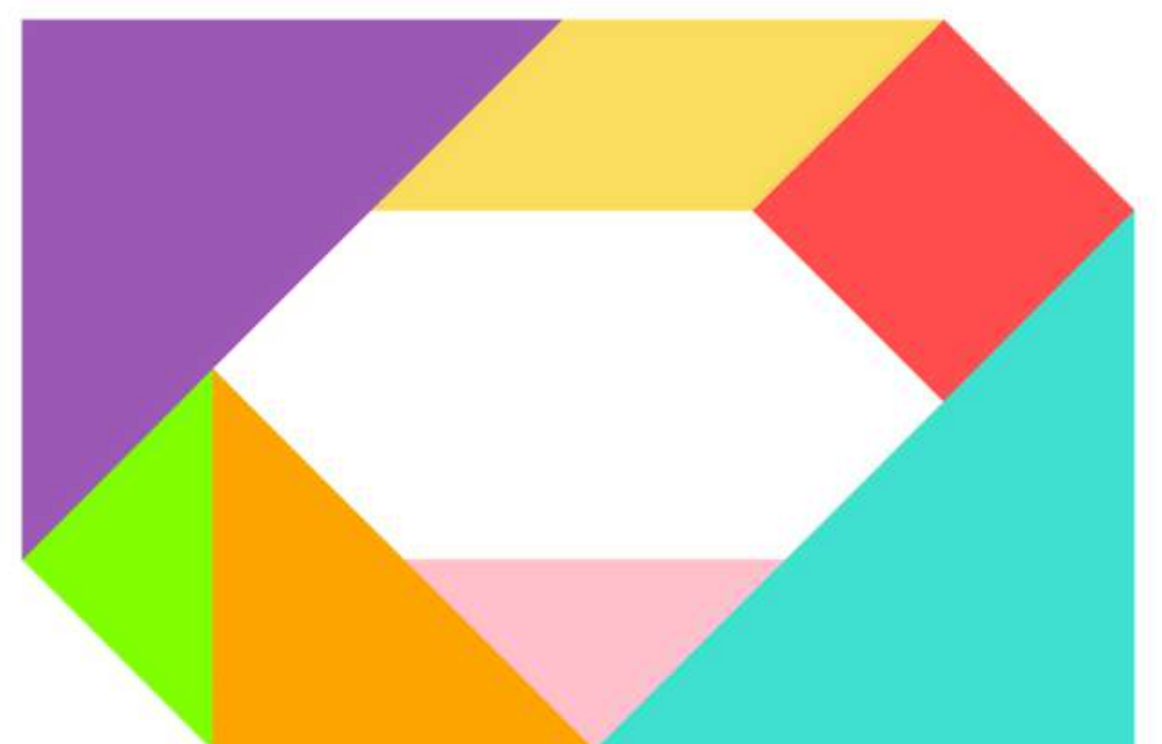
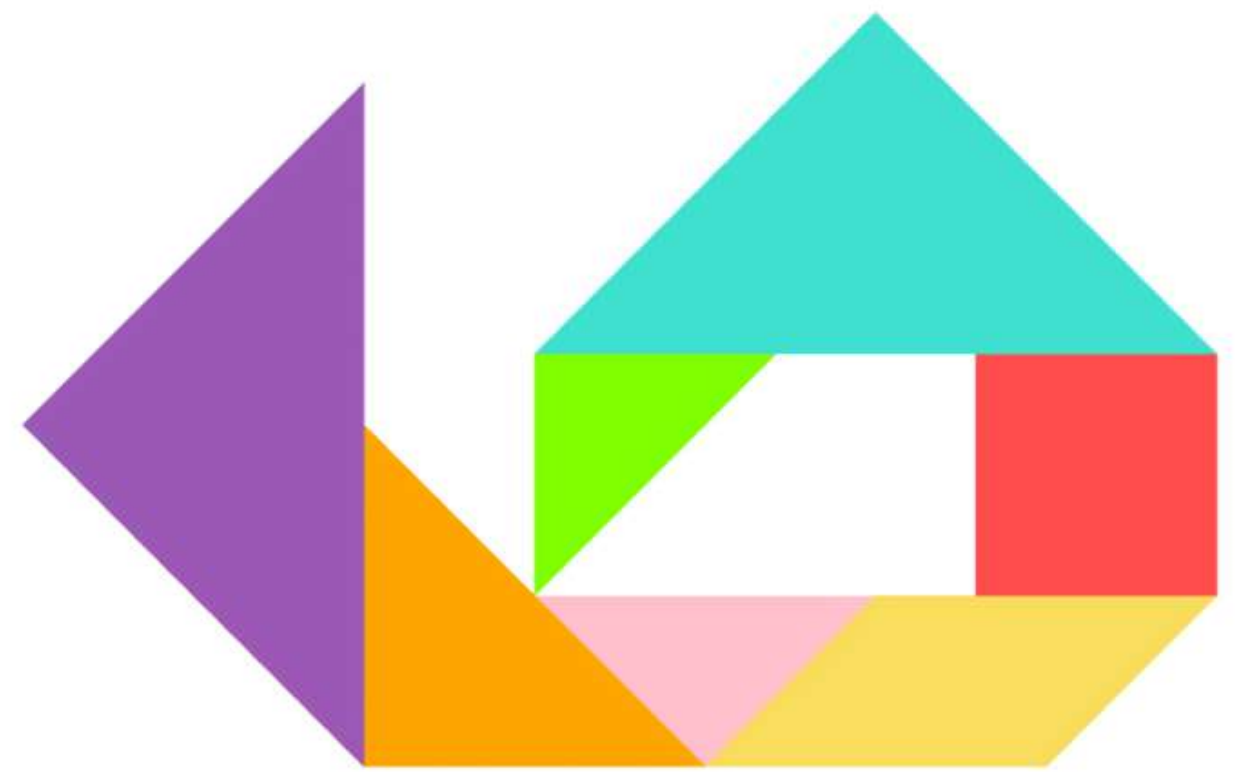
Po wykonaniu pracy plastycznej nauczyciel prosi o ewaluację zajęć. Prezentuje uczniom duży plakat (lub rysunek na tablicy interaktywnej), na którym narysowane są trzy obrazki: robot (oznaczający pracę związaną z programowaniem); tangram (oznaczający pracę z figurami geometrycznymi) oraz pędzel (oznaczający pracę plastyczną). Każdy uczeń ma do dyspozycji trzy kropki, które może dowolnie rozłożyć między te trzy obrazki, odpowiadając na pytanie: która praca sprawiła mi dziś największą przyjemność?













CO DALEJ??
MAŁA CHECKLISTA NAUCZYCIELA:
ZAZNACZ, OD CZEGO CHCESZ ZACZAĆ!

- ☐ ZORGANIZUJĘ PIERWSZE ZAJĘCIA Z ROBOTEM (NP. PHOTON, BEE-BOT, OZOBOT)
- ☐ WYBIORĘ 1-2 POJĘCIA PROGRAMISTYCZNE, KTÓRE WPROWADZĘ NA LEKCJI
- ☐ STWORZĘ WŁASNY SCENARIUSZ W DUCHU METODY STEAM
- ☐ ZAPROSZĘ UCZNIÓW DO WYMYŚLANIA HISTORII Z ROBOTEM W ROLI GŁÓWNEJ
- ☐ POKAŻĘ RODZICOM, JAK WYGLĄDA NAUKA PROGRAMOWANIA PRZEZ ZABAWĘ
- ☐ WYMIENIĘ SIĘ POMYSŁAMI Z INNYMI NAUCZYCIELAMI
- ☐ POZWOLĘ SOBIE PRÓBOWAĆ, TESTOWAĆ, ZMIENIAĆ – I DOBRZE SIĘ PRZY TYM BAWIĆ!

TWÓJ PIERWSZY KROK Z ROBOTYKĄ I PROGRAMOWANIEM!

ZAPLANUJ JEDNE ZAJĘCIA, KTÓRE WPROWADZAJĄ DZIECI
W TEMAT KODOWANIA I ROBOTÓW.

CEL LEKCJI:

Co dzieci mają zrozumieć /poczuć /zapamiętać?
Jeden główny cel, który podasz dzieciom na początku zajęć:

ZASOBY:
Zastanów się z jakiego robota lub narzędzia skorzystasz?

Czy potrzebujesz materiałów dodatkowych? Jakich?

ZABAWA LUB ZADANIE:
Co będą robić dzieci? Jak to powiążesz z realnym życiem?
Przygotuj szkic kilku zabaw i ćwiczeń.

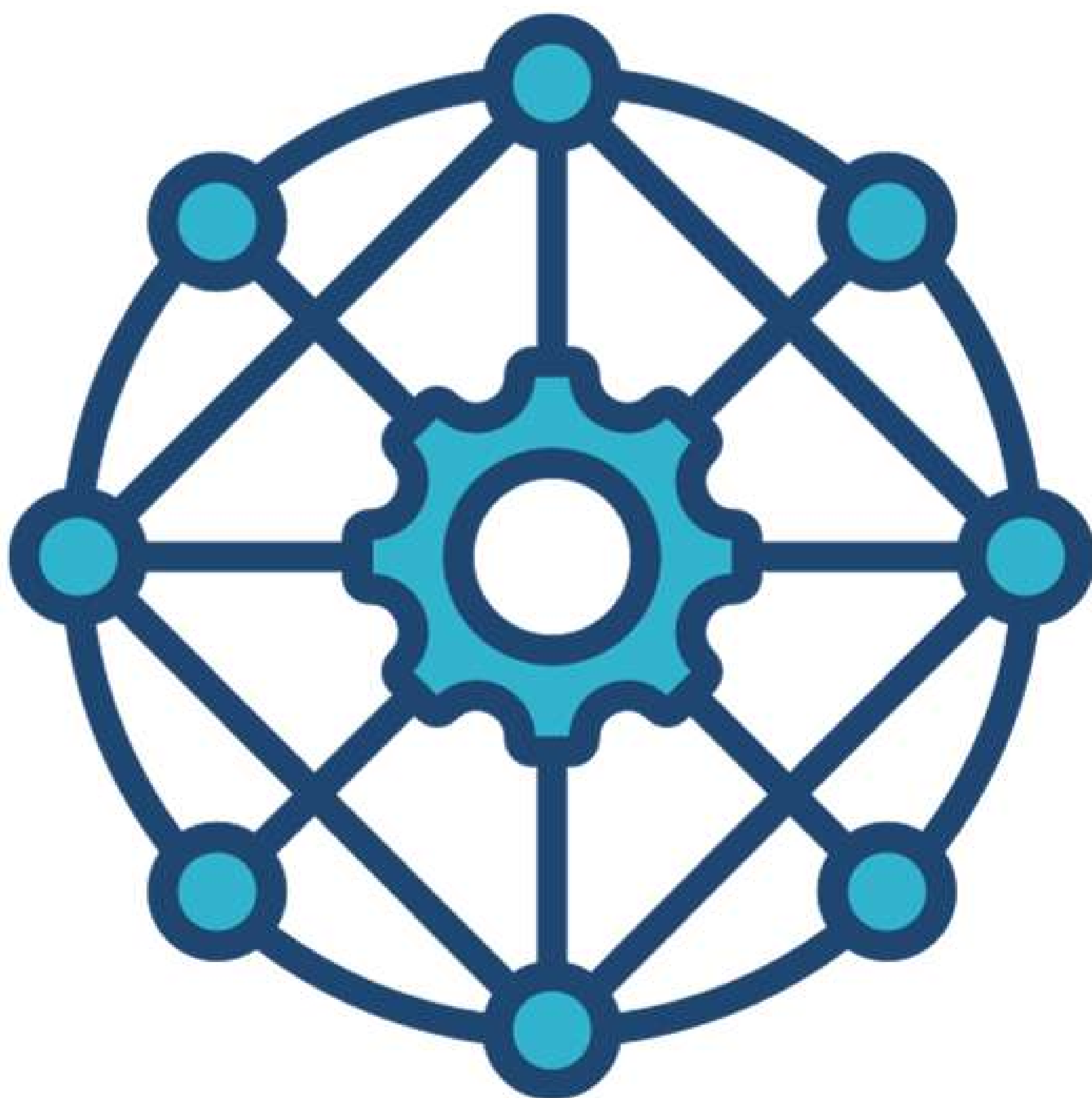
EWALUACJA
Co chcesz sprawdzić na końcu lekcji? O co zapytasz uczniów?

MIEJSCE NA TWOJE NOTATKI:

NAUCZANIE JEST JAK PROGRAMOWANIE – KAŻDY
KROK MA ZNACZENIE, ALE TO FINALNY REZULTAT DAJE
PRAWDZIWĄ SATYSFAKCJĘ.

CZASAMI WARTO PRZEPROWADZIĆ MAŁY
,DEBUGGING', BY UPEWNIĆ SIĘ, ŻE WSZYSTKO DZIAŁA
PERFEKCYJNIE!

POWODZENIA W PODEJMOWANIU KOLEJNYCH, EKSCYTUJĄCYCH WYZWAŃ!



ISBN 978-83-64879-05-0



9 788364 879050