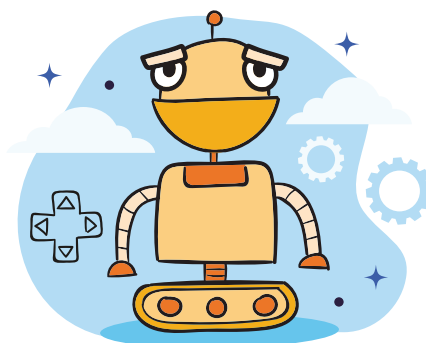




## Współrzędne - Joystick jako narzędzie wprowadzania danych



### Zagadnienia informatyczne

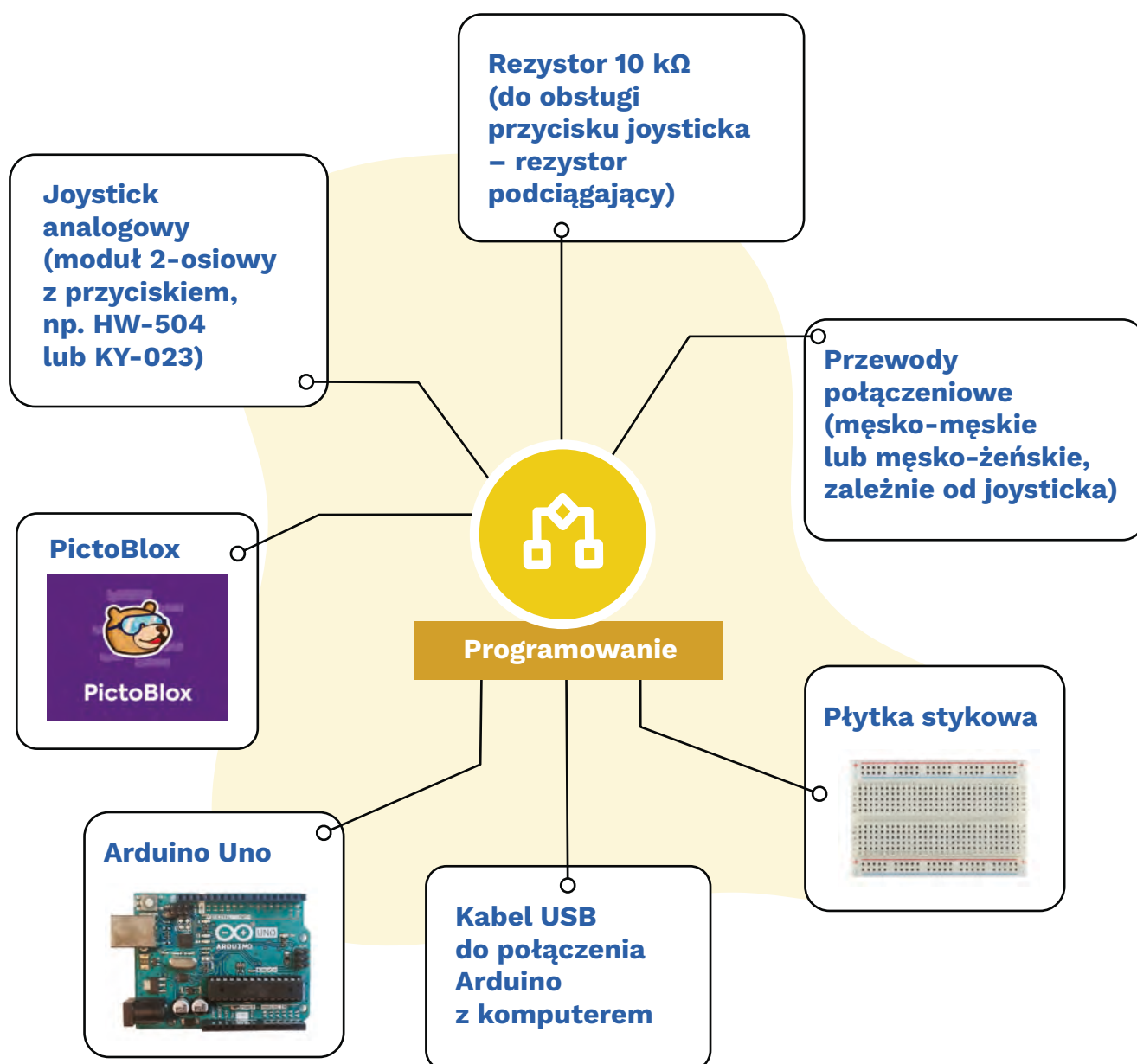
W tej części uczniowie poznają joystick jako urządzenie wejścia umożliwiające sterowanie programem za pomocą ruchu w dwóch osiach oraz przycisku. Uczą się, w jaki sposób Arduino odczytuje wartości analogowe odpowiadające położeniu joysticka oraz jak wykorzystać je do sterowania obiektami na ekranie. Pracując w środowisku PictoBlox, uczniowie sterują ruchem duszka, analizują wpływ wartości liczbowych na położenie obiektu oraz poznają pojęcie „strefy martwej” (deadzone), która pozwala eliminować niepożądane ruchy wynikające z niedokładności pomiaru. Dzięki temu rozumieją, jak urządzenia wejścia przekazują dane do programu oraz jak można je wykorzystać do sterowania systemami interaktywnymi.



### Zagadnienia matematyczne

Część matematyczna koncentruje się na opisie położenia punktów w układzie współrzędnych oraz wykonywaniu obliczeń związanych z przesunięciem punktów na płaszczyźnie. Uczniowie odczytują współrzędne punktów, obliczają nowe położenia na podstawie podanych zależności oraz wykorzystują joystick do wskazywania wyników na ekranie. Dzięki temu uczą się interpretować współrzędne jako liczbowy opis położenia, analizować zależności geometryczne oraz wykorzystywać matematykę do sterowania obiektem w przestrzeni dwuwymiarowej.

## Co będzie potrzebne do module 8?



# Poznajemy świat arduino: Joystick jako urządzenie sterujące

## Czym jest joystick

Joystick jest urządzeniem wejścia, które pozwala użytkownikowi sterować ruchem obiektów w komputerze lub w programie. Jest często używany w grach komputerowych, robotyce oraz urządzeniach sterujących.

Joystick, którego używasz w tym module, składa się z:

- drążka, który można wychylać w różnych kierunkach,
- dwóch czujników położenia:
  - jeden mierzy ruch w poziomie (oś X),
  - drugi mierzy ruch w pionie (oś Y),
- przycisku, który działa po naciśnięciu drążka.

Joystick nie wysyła informacji w postaci tekstu lub słów. Zamiast tego przekazuje wartości liczbowe, które opisują jego położenie.

Program może odczytać te wartości i na ich podstawie wykonać odpowiednią reakcję, na przykład przesunąć duszka na ekranie.

## Jak joystick przekazuje informacje do Arduino

Joystick jest podłączony do Arduino za pomocą pinów analogowych.



Każda oś joysticka działa jak pokrętło, które zmienia napięcie elektryczne. Arduino potrafi zmierzyć to napięcie i zamienić je na liczbę z zakresu:

0 do 1023

#### **Oznacza to:**

- wartość bliska 0 – drążek wychylony maksymalnie w jedną stronę,
- wartość bliska 1023 – drążek wychylony maksymalnie w drugą stronę,
- wartość bliska 512 – drążek znajduje się w położeniu środkowym.

Program może sprawdzać te wartości i na ich podstawie określić kierunek ruchu.

#### **Na przykład:**

- jeśli wartość osi X jest większa niż 512 – joystick jest wychylony w prawo,
- jeśli wartość osi X jest mniejsza niż 512 – joystick jest wychylony w lewo,
- jeśli wartość osi Y jest większa niż 512 – joystick jest wychylony w dół,
- jeśli wartość osi Y jest mniejsza niż 512 – joystick jest wychylony w górę.

### **Przycisk w joysticku (funkcja kliknięcia)**

Joystick oprócz możliwości wychylania posiada również wbudowany przycisk. Przycisk ten działa wtedy, gdy naciśniesz drążek joysticka pionowo w dół.

Działa on tak samo jak zwykły przycisk elektroniczny. Po naciśnięciu przycisku jego stan zmienia się i Arduino może to wykryć.

Joystick posiada specjalny pin oznaczony jako: **SW** (skrót od switch – przełącznik)

Pin ten przekazuje informację o tym, czy przycisk jest wciśnięty.

Program może odczytać stan przycisku za pomocą polecenia: **read digital pin**

Pin cyfrowy przyjmuje jedną z dwóch wartości:

- **0 – przycisk jest wciśnięty**
- **1 – przycisk nie jest wciśnięty**

Dzięki temu program może reagować na kliknięcie joysticka.





## Zadanie 1

### Pierwsze podłączenie joysticka – wykrywanie kliknięcia

#### Cel:

Nauczyć się podłączać joystick do Arduino oraz sprawdzić, w jaki sposób program wykrywa naciśnięcie wbudowanego przycisku i reaguje zmianą stanu diody LED.

#### Informacja — jak działa przycisk joysticka

Joystick oprócz wychylania w osiach X i Y posiada również wbudowany przycisk, który działa po naciśnięciu drążka pionowo w dół.

Przycisk ten:

- łączy pin sygnałowy z masą (GND),
- powoduje zmianę stanu napięcia na pinie cyfrowym Arduino.

Arduino rozróżnia dwa stany:

- **HIGH (stan wysoki)**  
— napięcie ok. 5 V
- **LOW (stan niski)**  
— napięcie ok. 0 V

Aby stan był stabilny, stosuje się **rezystor podciągający 10 kΩ**, który:

- utrzymuje stan HIGH, gdy przycisk nie jest naciśnięty,
- pozwala na pojawienie się stanu LOW po naciśnięciu przycisku.

- Dzięki temu Arduino może jednoznacznie wykryć naciśnięcie joysticka.

## Polecenie

1. Podłącz joystick do Arduino zgodnie ze schematem.

Zwróć szczególną uwagę na:

- zasilanie 5 V,
- masę GND,
- podłączenie pinu przycisku do pinu cyfrowego Arduino,
- rezystor 10 kΩ między pinem sygnałowym a 5 V.

2. Podłącz Arduino do komputera.

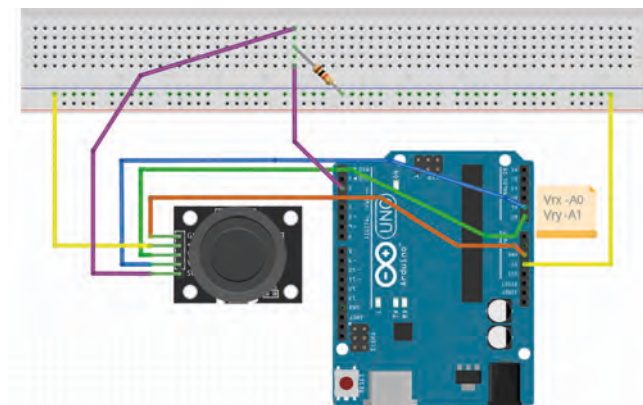
3. Otwórz przygotowany program w PictoBlox.

4. Uruchom program.

5. Naciśnij joystick pionowo w dół i obserwuj diodę LED na płytce Arduino.

6. Zwolnij joystick i sprawdź reakcję układu.

7. Naciśnij joystick kilkukrotnie, aby upewnić się, że układ działa poprawnie.



Rysunek 56

Podłączenie Arduino z joystickiem i rezystorem 10k



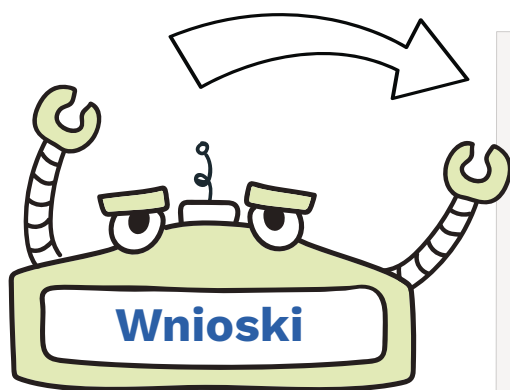
Rysunek 57

Szkic programu



## Kryterium osiągnięcia celu

- ☑ Dioda LED zapala się po naciśnięciu joysticka.
- ☑ Dioda LED gaśnie po zwolnieniu joysticka.
- ☑ Układ reaguje stabilnie i powtarzalnie na kolejne naciśnięcia.



1. Czy Arduino „wie”, że używasz joysticka, czy tylko odczytuje zmianę napięcia na pinie? Co to oznacza dla uniwersalności wejść cyfrowych?
2. Dlaczego program reaguje tylko na naciśnięcie przycisku, a nie na siłę nacisku lub kierunek wychylenia joysticka? Co to mówi o sposobie działania przycisku w joysticku?
3. Czy program reaguje na pojedyncze naciśnięcie, czy na cały czas trzymania przycisku? Jak można by zmodyfikować program, aby reagował tylko raz?



## Zadanie 2

### Sterowanie duszkiem za pomocą joysticka – pierwsza wersja

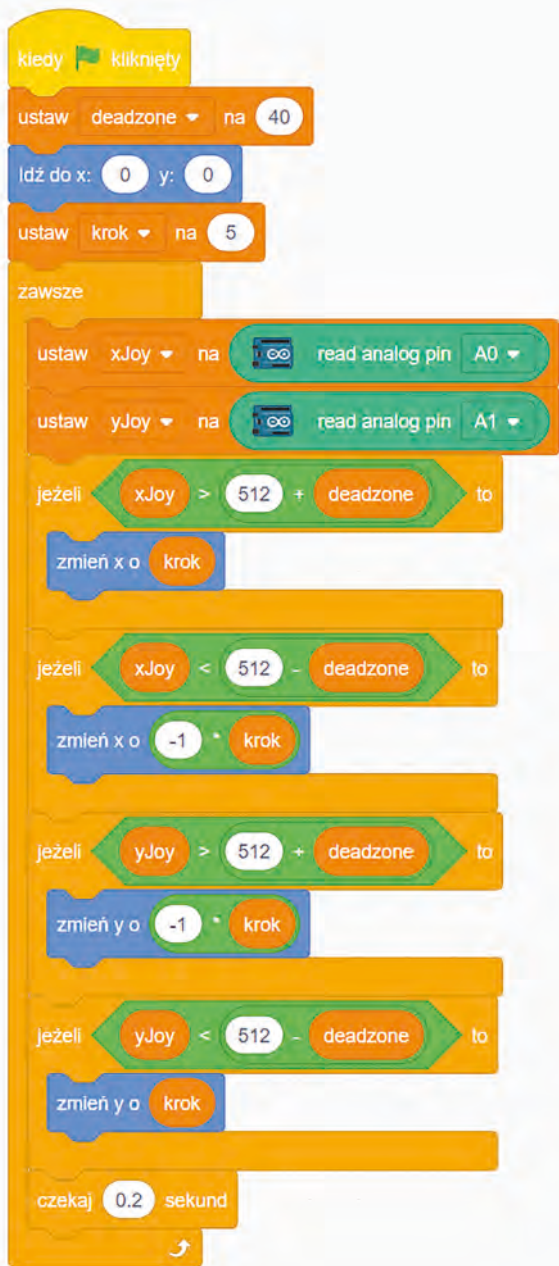
#### Cel:

Celem zadania jest odczyt wartości z joysticka (oś X i Y) oraz sterowanie ruchem duszka na ekranie. Zadanie pokazuje także, że rzeczywiste czujniki nie są idealnie dokładne — nawet w spoczynku mogą generować niewielkie odchylenia od wartości środkowej.

#### Polecenie

1. Uruchom przygotowany program w PictoBlox (zielona flaga).
2. Obserwuj, że duszek startuje w środku sceny:  $x = 0$ ,  $y = 0$ .
3. Wychyl joystick w prawo/lewo i sprawdź, jak zmienia się pozycja duszka na osi X.
4. Wychyl joystick w górę/dół i sprawdź, jak zmienia się pozycja duszka na osi Y.
5. Spróbuj wychylić joystick po przekątnej (np. prawo+góra) i sprawdź, czy duszek porusza się jednocześnie w dwóch osiach.
6. Puść joystick (pozycja spoczynkowa) i obserwuj zachowanie duszka przez kilka sekund.





Rysunek 58  
Szkielet programu do zadania 2

- Jeśli duszek porusza się „odwrotnie” niż intuicyjnie (np. joystick w górę, duszek w dół) — zanotuj to (w uwagach poniżej zobaczysz jak można to naprawić).

## Działanie programu

- Duszek ustawia się na środku ekranu ( $x=0$ ,  $y=0$ ).
- Program w pętli ciągle odczytuje dwie wartości analogowe:
  - A0 → oś X joysticka
  - A1 → oś Y joysticka
- Jeżeli odczyt na osi X jest większy niż 512 – duszek przesuwany jest w prawo o stały krok. Jeżeli mniejszy niż 512 – duszek przesuwany jest w lewo o ten sam krok.
- Podobnie dzieje się dla osi Y (w górę / w dół).
- Program robi krótką pauzę (np. 0,2 s), aby ruch był czytelny.

## Kryterium osiągnięcia celu

- ☒ Duszek **zmienia położenie na osi X i Y** w reakcji na wychylenie joysticka.

- ☑ Da się wymusić ruch po przekątnej (czyli program reaguje na obie osie).
- ☑ W spoczynku joysticka da się zaobserwować jedno z dwóch zachowań:
  - duszek stoi w miejscu **albo**
  - duszek wykonuje drobne, niekontrolowane ruchy (co jest poprawną obserwacją w tym zadaniu).
- ☑ Uczeń potrafi wskazać, czy kierunek osi Y/X jest zgodny z intuicją (tak/nie).

### Ważna obserwacja 1:

#### „duszek sam się porusza”

Może się zdarzyć, że duszek delikatnie przesuwa się, mimo że joysticku nie dotykasz.

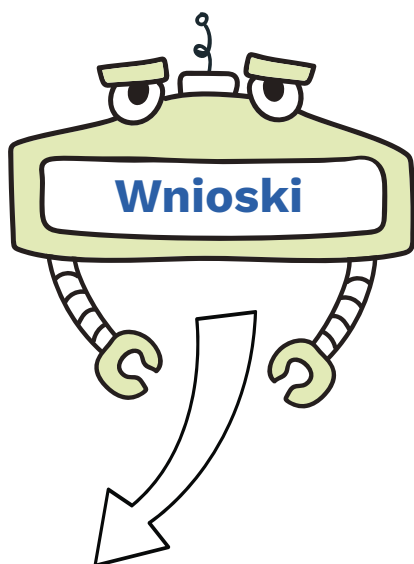
Dlaczego? Bo joystick w spoczynku nie zwraca zawsze dokładnie 512, tylko wartości w okolicy (np. 509–516). Program interpretuje to jako minimalny ruch.

To jest normalne zachowanie i w kolejnym zadaniu zobaczysz, jak to naprawić za pomocą „strefy martwej” (deadzone).





Rysunek 59  
Przeprogramowanie wbrew intuicji tzn. wartości powyżej 512 zmniejszają współrzędną Y a wartości poniżej 512 zwiększają współrzędną Y, niebieskie bloczki operacyjne można przestawić by dopasować do faktycznego działania.



## Ważna obserwacja 2: Kierunki mogą być „odwrotne”

Niektóre joysticki mogą działać tak, że:

- Przykładowo ruch w górę powoduje spadek wartości na osi Y zamiast wzrostu (albo odwrotnie), ewentualnie ruch w lewo na osi X joysticka powoduje ruch postaci w prawo
- zależy to od modelu joysticka i sposobu jego ustawienia (czasem trzymasz go „odwróconego” względem przewodów).

Jeśli duszek porusza się w przeciwnym kierunku niż oczekujesz, możesz to łatwo poprawić w programie – zamieniając znaki przy ruchu osi Y (lub X), np.:

- zamiast zmień y o +krok dajesz zmień y o -krok.

W tym fragmencie programu mamy tak przeprogramowany ruch w osi Y:  
jeżeli > 512 to zmień o -1\*krok

1. Dlaczego duszek czasem minimalnie „dryfuje”, mimo że joysticku nie dotykasz?
2. Jak rozpoznać, że oś w programie działa „odwrotnie” niż ruch joysticka — i co w kodzie/blokach musiałbyś zmienić, żeby to odwrócić?



## Zadanie 3

### Deadzone

– zatrzymaj niekontrolowany ruch duszka



Rysunek 60  
Listing programu moduł8infzad3  
– ustawienie deadzone

## Cel:

Celem zadania jest poprawienie programu sterującego duszkiem za pomocą joysticka tak, aby duszek nie poruszał się samoczynnie, gdy joystick znajduje się w położeniu spoczynkowym. W tym zadaniu wprowadzisz tzw. deadzone, czyli obszar tolerancji, w którym ruch joysticka jest ignorowany. Dzięki temu duszek będzie się poruszał tylko wtedy, gdy joystick zostanie wyraźnie wychylony.

## Polecenie

1. Uruchom program z wprowadzoną strefą martwą (deadzone).
2. Test spoczynku: nie dotykaj joysticka przez 5–10 sekund i obserwuj duszka.
3. Test reakcji: delikatnie wychyl joystick w prawo i sprawdź, w którym momencie duszek zaczyna się poruszać.
4. Zmień wartość deadzone (np. na mniejszą i większą) i porównaj zachowanie sterowania:
  - czy duszek nadal stoi w spoczynku?
  - jak łatwo „ruszyć” postać?





1. Jak zmienia się sterowanie, gdy deadzone ustawisz zbyt małe albo zbyt duże? (co zyskujesz, a co tracisz w obu przypadkach)
2. Dlaczego deadzone to w praktyce „filtr” na dane z czujnika — i czemu ten filtr jest potrzebny nawet wtedy, gdy sprzętu nie dotykasz?

5. (Mini-projekt) Zrób prostą grę typu labirynt albo pobierz gotowy projekt Scratch/PictoBlox i przerób sterowanie tak, aby odbywało się joystickiem Arduino zamiast klawiaturą.

### Kryterium osiągnięcia celu

- ☒ Gdy joystick jest w spoczynku, duszek pozostaje nieruchomy (brak dryfu).
- ☒ Duszek zaczyna się poruszać dopiero po wyraźnym wychyleniu joysticka poza deadzone.
- ☒ Po zmianie parametru deadzone uczeń potrafi zauważyć różnicę w sterowaniu (czułość / stabilność).
- ☒ W wersji „mini-projekt” postać w grze reaguje na joystick, a nie na klawiaturę.

## Wyjaśnienie działania deadzone

Joystick jest elementem analogowym. Oznacza to, że jego wartości mogą się delikatnie zmieniać nawet wtedy, gdy nie jest poruszany.

Na przykład zamiast idealnej wartości: 512 może zwracać: 509, 513, 508, 516. Program bez deadzone traktowałby te zmiany jako ruch. Deadzone tworzy obszar tolerancji: od 472 do 552. Dopiero po przekroczeniu tego zakresu duszek zaczyna się poruszać.

# Odkrywamy matematykę:

## Układ współrzędnych – obliczanie i wyznaczanie położenia punktów

W matematyce często zachodzi potrzeba dokładnego określenia położenia punktu. Aby było to możliwe, używa się układu współrzędnych.

Układ współrzędnych składa się z dwóch osi:

- osi poziomej, nazywanej osią X,
- osi pionowej, nazywanej osią Y.

Obie osie przecinają się w punkcie (0, 0), który nazywa się początkiem układu współrzędnych.

Każdy punkt na płaszczyźnie można opisać za pomocą dwóch liczb zapisanych w nawiasie, na przykład: (3, 5)

Pierwsza liczba oznacza położenie na osi X, a druga liczba oznacza położenie na osi Y.

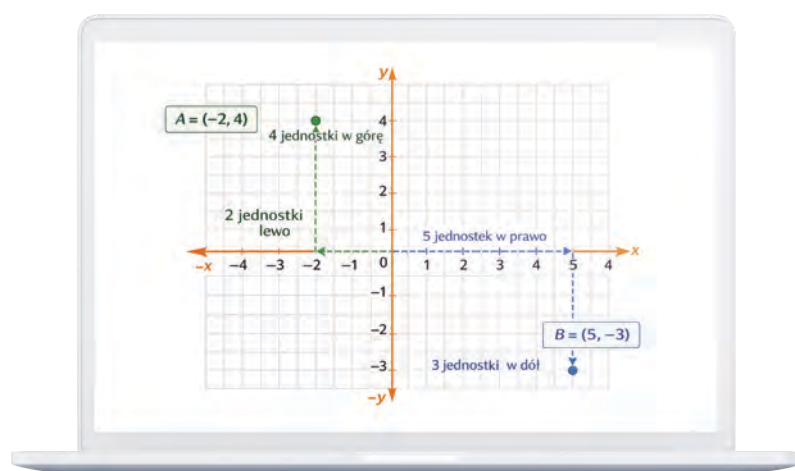
Oznacza to, że aby znaleźć punkt (5,-3):

- należy przesunąć się o 5 jednostek w prawo,
- następnie o 3 jednostek w dół.

Podobnie punkt (-2, 4) oznacza:

- 2 jednostki w lewo,
- 4 jednostki w górę.

Każdy punkt ma jedno, dokładnie określone położenie.



Rysunek XX  
Opis



## Odczytywanie i zapisywanie współrzędnych

Współrzędne punktu zapisujemy zawsze w tej samej kolejności:

$(x, y)$

gdzie:

$x$  – określa położenie w poziomie,

$y$  – określa położenie w pionie.

Na przykład:

**$A = (-2, 4)$**

**$B = (5, -3)$**

Punkt A znajduje się 2 jednostki w lewo i 4 jednostki w górę, względem początku układu.

Punkt B znajduje się 5 jednostek w prawo i 3 jednostki w dół, względem początku układu.

## Przesunięcie punktu

Punkt można przesunąć w inne miejsce. Aby to zrobić, należy dodać odpowiednie wartości do jego współrzędnych.

Rozważmy przykład:

Punkt A ma współrzędne:

**$A = (2, 3)$**

Jeżeli przesuniemy punkt:

o 4 jednostki w prawo oraz

o 2 jednostki w górę,

to nowe współrzędne wyniosą:

**$(2 + 4, 3 + 2) = (6, 5)$**

Widzimy więc, że:

- do współrzędnej  $x$  dodajemy przesunięcie w poziomie,
- do współrzędnej  $y$  dodajemy przesunięcie w pionie.

## Współrzędne w programie PictoBlox

W programie PictoBlox możemy wykorzystać orientację w przestrzeni wg współrzędnych układu osi  $x$  i  $y$ , gdzie centrum ekranu to punkty  $(0, 0)$

A wartości skrajne osi wynoszą:

od  $-240$  do  $240$  w poziomie,

od  $-180$  do  $180$  w pionie.

Joystick pozwala sterować położeniem punktu na ekranie. Dzięki temu można sprawdzić poprawność wykonanych obliczeń.



Rysunek 61  
Oś  $X$ ,  $Y$  w oknie programu Pictoblox

## Sterowanie Joystickiem – współczynnik tolerancji.

W tych zadaniach joystick służy do wskazywania punktów w układzie współrzędnych, ale nie jest on narzędziem idealnie precyzyjnym.

Program uznaje odpowiedź za poprawną, jeśli wskazany punkt znajduje się blisko poprawnego położenia (z tolerancją około  $\pm 10$  jednostek).

Dlatego najważniejsze jest, aby:

- odczytać współrzędne punktów,
- wykonać obliczenia w brudnopisie,
- wyznaczyć dokładne współrzędne szukanego punktu,
- a dopiero potem wskazać jego położenie joystickiem. Joystick służy tylko do zaznaczenia wcześniej obliczonego wyniku.





## Zadanie 1

### Przesunięcie punktu o tę samą odległość

#### Cel:

- **Zrozumieć**, że przesunięcie punktu na płaszczyźnie można opisać jako zmianę współrzędnych  $x$  i  $y$ .
- **Nauczyć** się obliczać nowe położenie punktu na podstawie znanego przesunięcia innego punktu.
- **Zastosować** tę samą zmianę współrzędnych do innego punktu.

#### Polecenie

Program pozwoli Ci wskazać położenie punktów za pomocą joysticka.

1. Sterując joystickiem ustaw krzyżyk w wybranym miejscu i zatwierdź położenie punktu A.
2. Następnie w ten sam sposób wskaż położenie punktu B.

Po wskazaniu punktów A i B program:

- wyznaczy ich współrzędne,
- wylosuje i wyświetli punkt C.

Twoim zadaniem jest wskazać punkt D, który jest przesunięty względem punktu C tak samo, jak punkt B jest przesunięty względem punktu A.

Aby to zrobić:

1. Odczytaj współrzędne punktów A i B:

$$\mathbf{A = (x_A, y_A)}$$

$$\mathbf{B = (x_B, y_B)}$$



2. Oblicz zmianę współrzędnych między punktami A i B:  
**zmiana  $x = x_B - x_A$**   
**zmiana  $y = y_B - y_A$**
3. Odczytaj współrzędne punktu C:  
 $C = (x_C, y_C)$
4. Oblicz współrzędne punktu D:  
 **$x_D = x_C + \text{zmiana } x$**   
 **$y_D = y_C + \text{zmiana } y$**   
Punkt D ma współrzędne:  
 **$D = (x_D, y_D)$**
5. Sterując joystickiem ustaw krzyżyk w obliczonym miejscu i zatwierdź odpowiedź.
6. Program sprawdzi poprawność wskazanego punktu.

### Kryterium osiągnięcia celu

- ☒ Uczeń potrafi wskazać punkty A i B za pomocą joysticka.
- ☒ Uczeń poprawnie oblicza przesunięcie między punktami A i B.
- ☒ Uczeń stosuje to samo przesunięcie do punktu C.
- ☒ Wskazany punkt D pokrywa się z położeniem wyznaczonym przez program.



**Współrzędne punktów:** $A = ( \quad , \quad )$  $B = ( \quad , \quad )$  $C = ( \quad , \quad )$ **Obliczenia:** $\text{zmiana } x = x_B - x_A = \underline{\hspace{2cm}}$  $\text{zmiana } y = y_B - y_A = \underline{\hspace{2cm}}$  $x_D = x_C + \text{zmiana } x = \underline{\hspace{2cm}}$  $y_D = y_C + \text{zmiana } y = \underline{\hspace{2cm}}$ **Brudnopis – miejsce na obliczenia**

\_\_\_\_\_○ **Wynik:  $D = ( \quad , \quad )$**  ○\_\_\_\_\_



## Zadanie 2

### Wyznaczanie czwartego wierzchołka równoległoboku



Rysunek 62  
Widok okna wykonanego zadania nr 2,  
wskazano położenie punktu D

### Cel:

- **Nauczyć się** obliczać współrzędne brakującego wierzchołka równoległoboku.
- **Zrozumieć** zależność między położeniem wierzchołków figury na płaszczyźnie a działaniami na współrzędnych.
- **Zastosować** wzór pozwalający wyznaczyć punkt na podstawie trzech pozostałych.

### Polecenie

Po uruchomieniu programu na ekranie pojawią się trzy punkty:

$$\mathbf{A} = (x_A, y_A)$$

$$\mathbf{B} = (x_B, y_B)$$

$$\mathbf{C} = (x_C, y_C)$$

Punkty te są wierzchołkami równoległoboku. Odcinek AB stanowi jedną ścianę równoległoboku, odcinek BC stanowi drugą ścianę równoległoboku. Twoim zadaniem jest wyznaczyć współrzędne czwartego wierzchołka D.

1. Odczytaj współrzędne punktów A, B i C z ekranu.
2. Oblicz współrzędne punktu D, korzystając ze wzorów:



$$x_D = x_A + x_C - x_B$$

$$y_D = y_A + y_C - y_B$$

3. Punkt D ma współrzędne:

$$D = (x_D, y_D)$$

4. Sterując joystickiem ustaw krzyżyk w obliczonym miejscu na planszy.
5. Zatwierdź odpowiedź przyciskiem joysticka.
6. Program sprawdzi poprawność wskazanego punktu.

### Kryterium osiągnięcia celu

- ☒ Uczeń potrafi odczytać współrzędne punktów z układu współrzędnych.
- ☒ Uczeń poprawnie stosuje wzór na czwarty wierzchołek równoległoboku.
- ☒ Uczeń wskazuje punkt D w odpowiednim miejscu za pomocą joysticka.
- ☒ Wskazany punkt pokrywa się z położeniem wyznaczonym przez program.

**Współrzędne punktów:**

A = ( \_\_\_\_\_ , \_\_\_\_\_ )

B = ( \_\_\_\_\_ , \_\_\_\_\_ )

C = ( \_\_\_\_\_ , \_\_\_\_\_ )

**Obliczenia:**

$x_D = x_A + x_C - x_B =$  \_\_\_\_\_

$y_D = y_A + y_C - y_B =$  \_\_\_\_\_

**Brudnopis – miejsce na obliczenia**

**Wynik: D = ( \_\_\_\_\_ , \_\_\_\_\_ )**





### Zadanie 3

#### Wyznaczanie punktu symetrycznego względem danego punktu

#### Cel:

- **Nauczyć się** obliczać współrzędne punktu symetrycznego względem innego punktu.
- **Zrozumieć**, że punkt symetryczny znajduje się w takiej samej odległości od środka symetrii, ale po przeciwnej stronie.
- **Zastosować** wzory pozwalające wyznaczyć punkt na podstawie współrzędnych punktu i środka symetrii.

#### Polecenie

Po uruchomieniu programu na ekranie pojawią się dwa punkty:

$$A = (x_A, y_A)$$

$$S = (x_S, y_S)$$

Punkt S jest środkiem odcinka AB. Twoim zadaniem jest wyznaczyć współrzędne punktu B — punktu symetrycznego do A względem punktu S.

1. Odczytaj współrzędne punktów A i S z ekranu.
2. Oblicz współrzędne punktu B, korzystając ze wzorów:

$$x_B = 2 \times x_S - x_A$$

$$y_B = 2 \times y_S - y_A$$



Rysunek 63  
Widok okna programu zadania 3 po  
wylosowaniu położenia punktów A i S

3. Punkt B ma współrzędne:  
 **$B = (x_B, y_B)$**
4. Sterując joystickiem ustaw krzyżyk w obliczonym miejscu na planszy.
5. Zatwierdź odpowiedź przyciskiem joysticka.
6. Program sprawdzi poprawność wskazanego punktu.

### **Kryterium osiągnięcia celu**

- ☒ Uczeń potrafi odczytać współrzędne punktów z układu współrzędnych.
- ☒ Uczeń poprawnie stosuje wzór na punkt symetryczny względem danego punktu.
- ☒ Uczeń wskazuje punkt B w odpowiednim miejscu za pomocą joysticka.
- ☒ Wskazany punkt pokrywa się z położeniem wyznaczonym przez program.



**Współrzędne punktów:**

A = ( \_\_\_\_\_ , \_\_\_\_\_ )

S = ( \_\_\_\_\_ , \_\_\_\_\_ )

**Obliczenia:**

$x_B = 2 \times x_S - x_A =$  \_\_\_\_\_

$y_B = 2 \times y_S - y_A =$  \_\_\_\_\_

**Brudnopis – miejsce na obliczenia**

**Wynik: B = ( \_\_\_\_\_ , \_\_\_\_\_ )**



## Czego się nauczyłeś?



### Matematyka

- Nauczyłeś się pracować z układem współrzędnych na płaszczyźnie.
- Potrafisz obliczyć przesunięcie punktu na podstawie różnicy współrzędnych.
- Umiesz wyznaczyć brakujący wierzchołek równoległoboku na podstawie trzech pozostałych punktów.
- Nauczyłeś się obliczać współrzędne punktu symetrycznego względem danego punktu.
- Zrozumiałeś związek między obliczeniami algebraicznymi a położeniem punktów na płaszczyźnie.
- Przekonałeś się, że matematyczne obliczenia można bezpośrednio zweryfikować w środowisku graficznym.



### Informatyka

- Dowiedziałeś się, jak działa joystick jako element wejścia analogowego i cyfrowego.
- Nauczyłeś się odczytywać wartości z osi X i Y oraz wykorzystywać je do sterowania obiektem na ekranie.
- Zrozumiałeś, że czujniki analogowe nie są idealnie dokładne i mogą generować niewielkie wahania sygnału.
- Poznałeś pojęcie deadzone (strefy martwej) i jego zastosowanie w stabilizacji sterowania.
- Zobaczyłeś, jak program interpretuje dane z czujnika i zamienia je na ruch urządzenia lub obiektu w aplikacji.
- Dowiedziałeś się, jak joystick może zastąpić klawiaturę lub inne urządzenia sterujące.

