

PAKIET MATERIAŁÓW DYDAKTYCZNYCH

do kształcenia na odległość dla nauczycieli
informatyki szkoły podstawowej

Projekt „Wsparcie placówek doskonalenia nauczycieli i bibliotek pedagogicznych w realizacji zadań związanych z przygotowaniem i wsparciem nauczycieli w prowadzeniu kształcenia na odległość”

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Materiał opracowany w ramach grantu przez ZHP Chorągiew Kielecka im. Stefana
Żeromskiego, mgr inż. Pawła Jachowicza

SCENARIUSZ 1 z 5

SCENARIUSZ ZAJĘĆ DLA: UCZNIÓW KLASY IV SZKOŁY PODSTAWOWEJ II SEMESTR

PROWADZONYCH PRZEZ nauczyciela informatyki

TEMAT:

ROZWIJANIE UMIEJETNOŚCI PROGRAMOWANIA. PISZEMY GRĘ „KOCIE SZTUCZKI”.

CELE KSZTAŁCENIA – WYMAGANIA OGÓLNE: (PODSTAWA PROGRAMOWA)

- pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń (E2-PODST-INF-2.0-KLIVVI-II.1.a),
- prosty program sterujący robotem lub innym obiektem na ekranie komputera (E2-PODST-INF-2.0-KLIVVI-II.1.b),
- sterowanie robotem lub obiektem na ekranie (E2-PODST-INF-2.0-KLIVVI-I.2.c),

TREŚCI NAUCZANIA – WYMAGANIA SZCZEGÓŁOWE: (PODSTAWA PROGRAMOWA)

- posługuje się technologią zgodnie z przyjętymi zasadami i prawem; przestrzega zasad bezpieczeństwa i higieny pracy (E2-PODST-INF-2.0-KLIVVI-V.1),
- testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów (E2-PODST-INF-2.0-KLIVVI-II.2),
- wykorzystuje komputer lub inne urządzenie cyfrowe do gromadzenia, porządkowania i selekcjonowania własnych zasobów (E2-PODST-INF-2.0-KLIVVI-III.1.b),
- w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu (E2-PODST-INF-2.0-KLIVVI-I.3),

- wymienia zagrożenia związane z powszechnym dostępem do technologii oraz do informacji i opisuje metody wystrzegania się ich (E2-PODST-INF-2.0-KLIVVI-V.3).

METODY PRACY:

- techniki informatyczne,
- Zintegrowana Platforma Edukacyjna,
- Prezentacja,
- praca indywidualna.

ŚRODKI DYDAKTYCZNE:

- komputer z systemem operacyjnym Windows 11,
- przeglądarka internetowa,
- ZOOM.

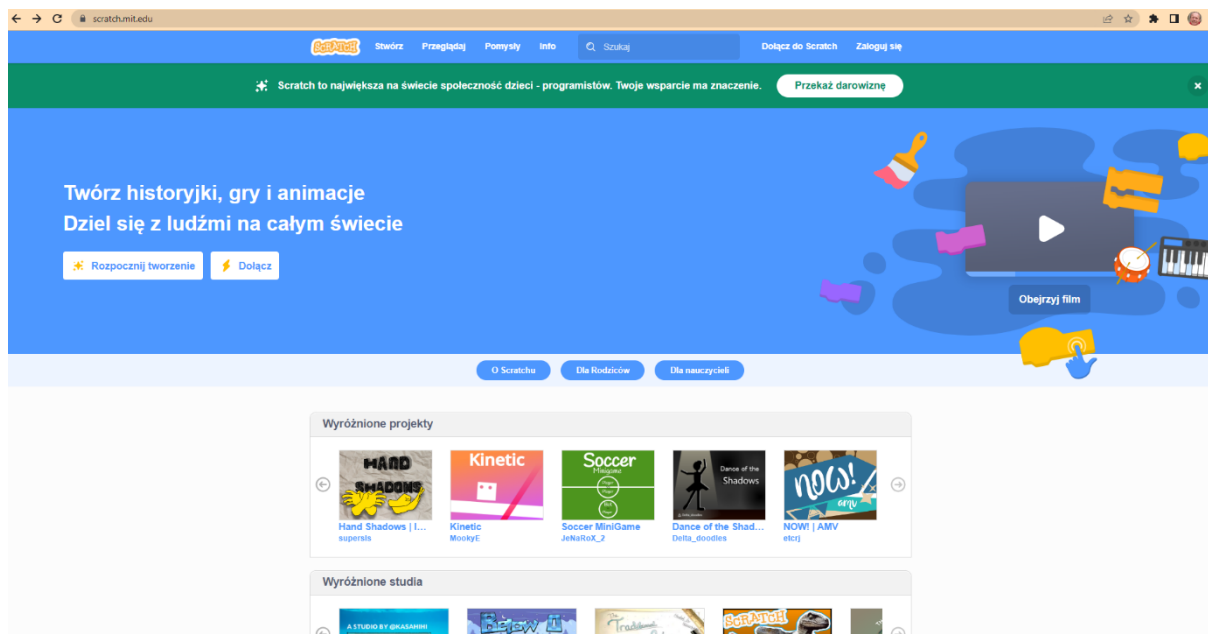
PRZEWIDYWANY CZAS: 45 minut

PROPONOWANY PRZEBIEG ZAJĘĆ:

Część 1: Przywitanie uczniów, sprawdzenie listy obecności, sprawdzenie komunikacji (mikrofony, słuchawki, kamery). Uczniowie są przy komputerach. (3 minuty)

Część 2: Podanie celu zajęć: Nauczyciel objaśnia cel lekcji, którym jest opracowanie prostej gry w programie *Scratch*. Podaje etapy pracy (stworzenie tła, postaci duszków, zaprogramowanie duszków i przetestowaniu gry. Pisanie skryptu gry będzie tematem dwóch kolejnych lekcji. Na tej lekcji uczniowie wykonają tło gry, utworzą duszka małpkę i zaprogramują jej ruchy. Objasnia, że uczniowie posiadający komputery stacjonarne czy laptopy mogą wykonywać zadania równoległe z nauczycielem. Czas 5 minut.

Część 3: Nauczyciel udostępnia swój ekran (przeglądarka internetowa) w systemie Zoom. Wzywa program z platformy: <https://scratch.mit.edu/>. Po wezwaniu platformy ekran przeglądarki ma postać:

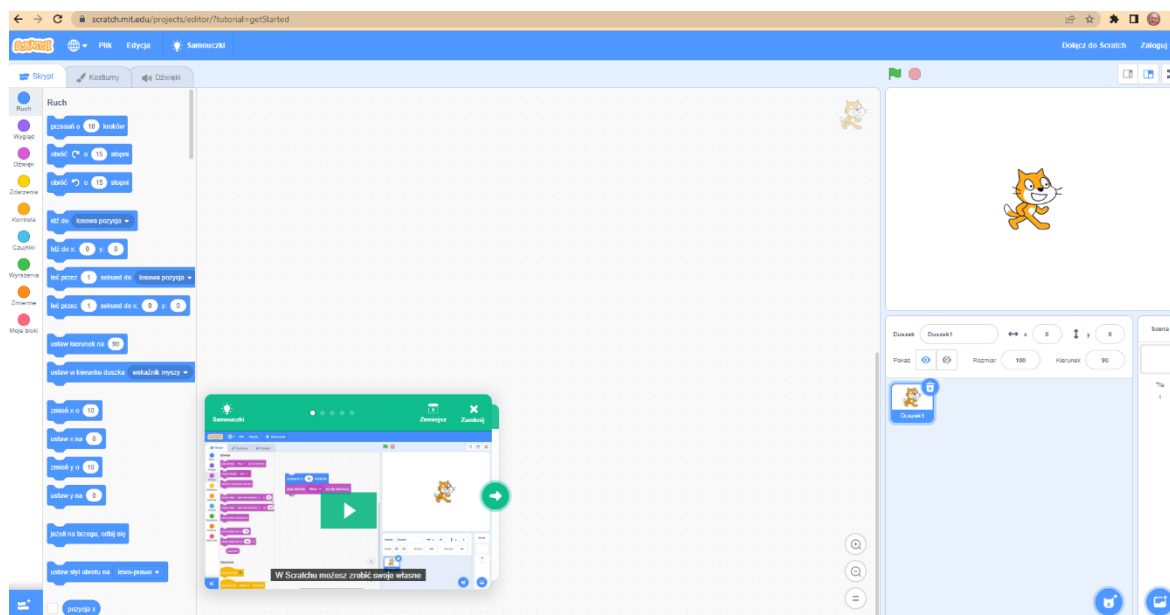


Rysunek 1: Ekran startowy Scratch (opracowanie własne).

Ćwiczenie 1: Stworzenie nowego skryptu, naniesienie tła i potrzebnego duszka.

Nauczyciel wydaje polecenie klasie i sam demonstruje udostępniając cały czas ekran swojego komputera. Czas 7 minut.

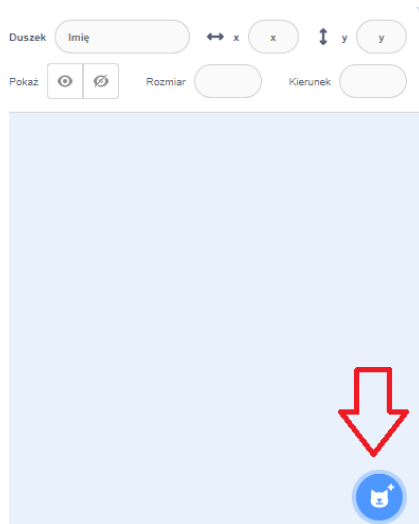
1. Kliknij na Stwórz w górnej części ekranu,



Rysunek 2: Pusty projekt Scratch (opracowanie własne).

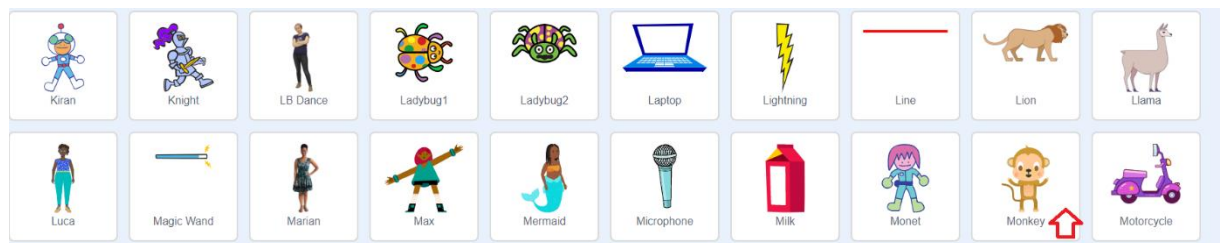
2. Usuń duszka Duszek1 (kot)

3. Dodaj duszka małpka z galerii duszków. Nauczyciel przypomina, gdzie uczeń ma kliknąć, aby uruchomić galerię duszków.



Rysunek 3: Ikona galerii duszków (*opracowanie własne*).

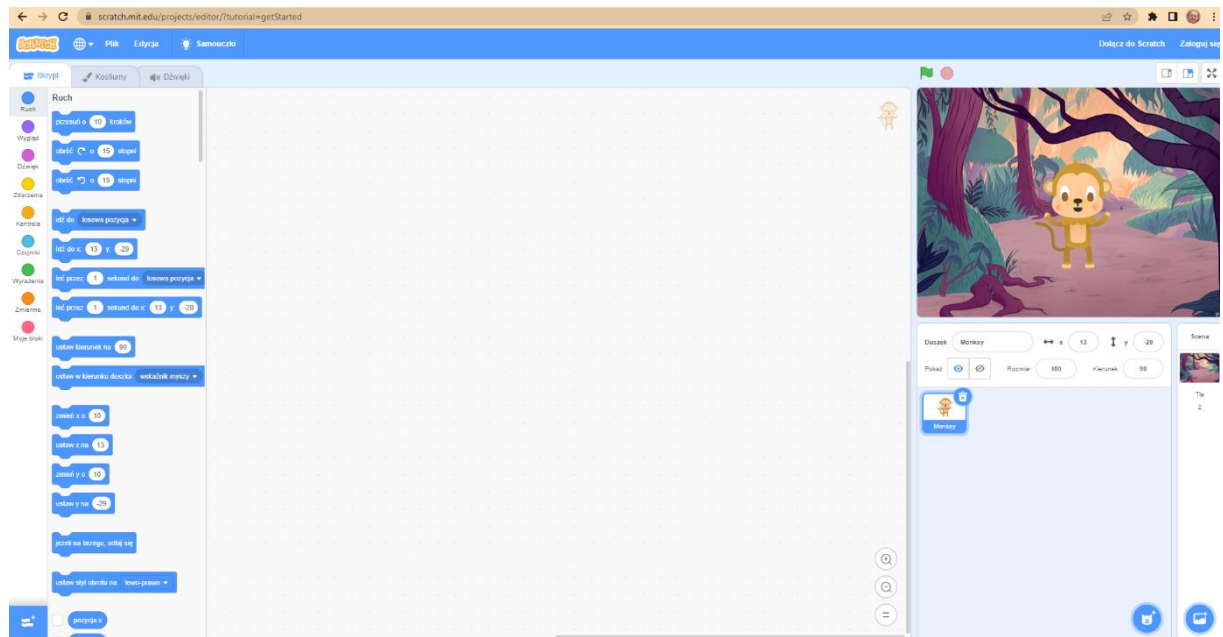
4. Wybierz duszka *Monkey* (małpka).



Rysunek 4: Duszek *Monkey* (*opracowanie własne*).

5. Dodaj tło z galerii tła. Nauczyciel przypomina, gdzie uczeń ma kliknąć, aby uruchomić galerię tła.

6. Wybierz tło *Jungle* (dżungla). Ekran powinien wyglądać następująco:

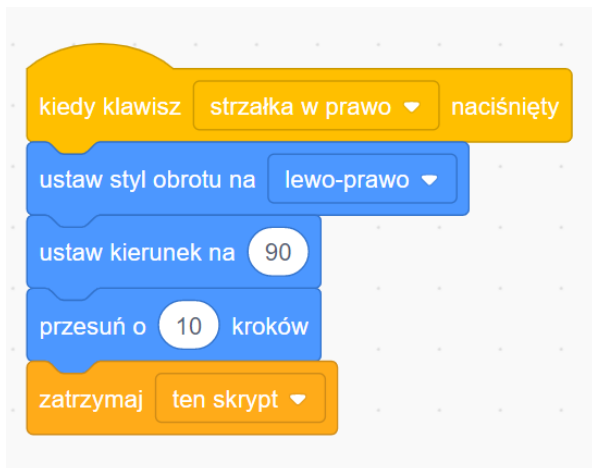


Rysunek 5: Ekran po wykonaniu punktów 1-6 (*opracowanie własne*).

Ćwiczenie 2: Napisanie skryptu na przesuwanie się duszka w lewo, prawo, do góry i do dołu. Nauczyciel, jeśli możliwe, przy pomocy uczniów, pisze skrypt duszka, który za pomocą klawisza strzałka w prawo będzie przesuwał duszka w prawo. Czas 10 minut

Wykaz rozkazów:

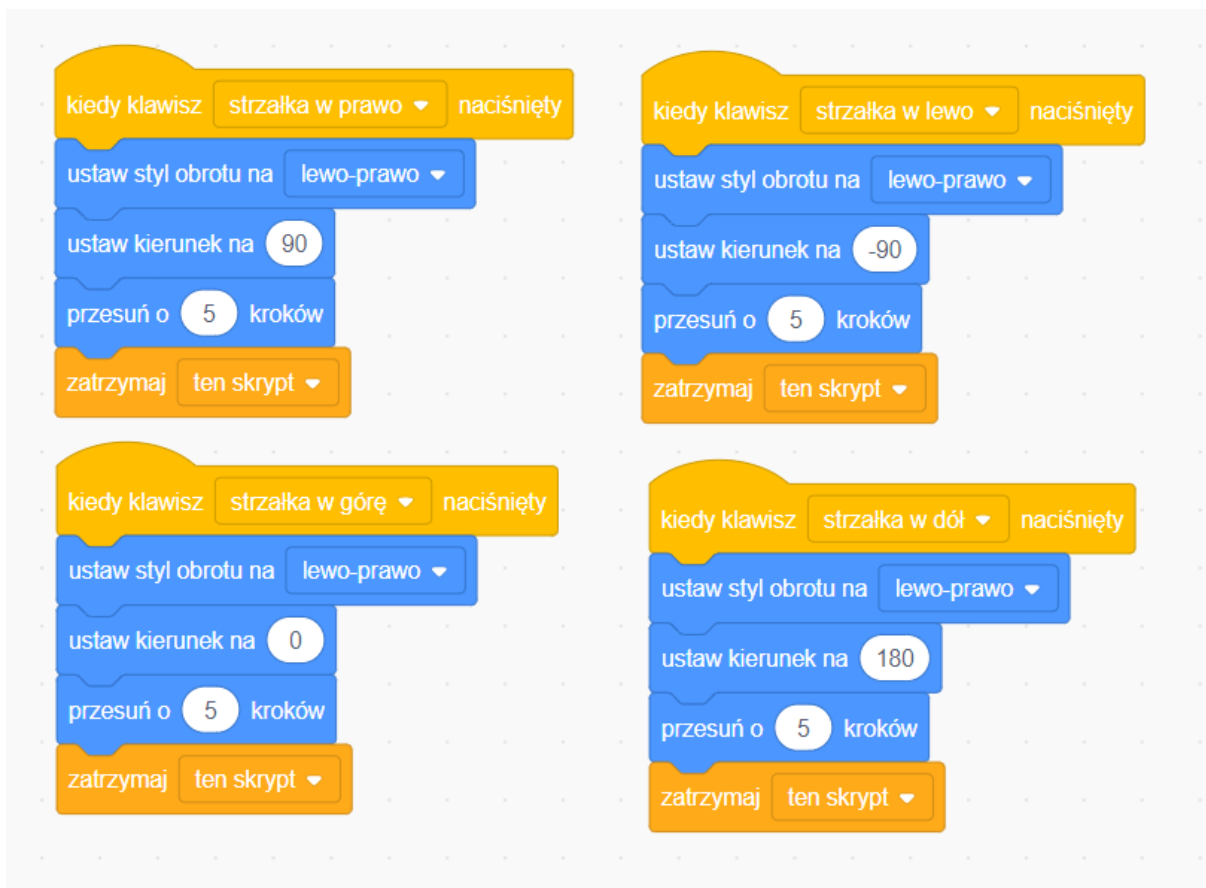
- **Grupa:** Zdarzenia, **rozkaz:** Kiedy klawisz strzałka w prawo naciśnięty,
- **Grupa:** Ruch, **rozkaz:** Ustal styl obrotów lewo-prawo,
- **Grupa:** Ruch, **rozkaz:** Ustaw kierunek na 90 stopni,
- **Grupa:** Ruch, **rozkaz:** Przesuń o 10 kroków,
- **Grupa:** Kontrola, **rozkaz:** Zatrzymaj ten skrypt.



Rysunek 6: Skrypt, za pomocą którego przesuńiesz duszka o 10 kroków strzałką w prawo (*opracowanie własne*).

Ćwiczenie 3: Zmień prędkość duszka. Nauczyciel słucha propozycji uczniów, co trzeba zrobić, aby przyspieszyć, czy opóźnić poruszanie się duszka. (zmiana ilości kroków). Nauczyciel prosi uczniów o zmianę ilości kroków we władnym skrypcie w przedziale od 1 do 15. Uczniowie mają dobrać szybkość przesuwu duszka według własnego uznania. Czas 5 minut.

Ćwiczenie 4: Opracuj następne skrypty – duszek powinien poruszać się w każdym kierunku. Możesz wykorzystać duplikowanie skryptu. Czas 10 minut. Skrypt po wykonaniu powyższych czynności powinien wyglądać następująco:



Rysunek 7: Skrypty do przemieszczania się małpki (*opracowanie własne*).

Część 4: Zapisanie na własnym dysku, zadanie pracy domowej. (5 minut)

Nauczyciel przypomina uczniom o konieczności zapisania własnej pracy na komputerze. Uczniowie to wykonują. Nauczyciel kończy udostępnianie ekranu w Zoom. Zadaje pytanie odnośnie trudności w wykonaniu pracy i wyjaśnia wątpliwości.

EWALUACJA ZAJĘĆ (sprawdzenie osiągnięcia zakładanych efektów kształcenia)

Praca domowa: Zmniejsz małpkę według własnego uznania. Dodaj nowego duszka – banany. „Zawieś” duszka w na drzewie i dopasuj rozmiar bananów wg własnego uznania. Pamiętaj o zapisaniu swojej pracy. Czas wykonania: na następną lekcję.

BIBLIOGRAFIA

Opracowanie własne

ZAŁĄCZNIKI

EkranScratch.png

EkranScratchNowySkrypt.png

DoGaleriiDuszków.png

DuszekMonkey.png

Ekran-1.png

RozkazyDoRuchuWPrawo.png

SkryptyMałpkiPoLekcji1.png

ProjektMałpieFigue.sb3

SCENARIUSZ 2 z 5

SCENARIUSZ ZAJĘĆ DLA: UCZNIÓW KLASY V SZKOŁY PODSTAWOWEJ I SEMESTR

PROWADZONYCH PRZEZ nauczyciela informatyki

TEMAT:

ROZWIJANIE UMIEJETNOŚCI PROGRAMOWANIA. PISZEMY GRĘ „PRZEJDŹ PRZEZ LABIRYNT”.

CELE KSZTAŁCENIA – WYMAGANIA OGÓLNE: (PODSTAWA PROGRAMOWA)

- pomysły historyjek i rozwiązania problemów, w tym proste algorytmy z wykorzystaniem poleceń sekwencyjnych, warunkowych i iteracyjnych oraz zdarzeń (E2-PODST-INF-2.0-KLIVVI-II.1.a),
- prosty program sterujący robotem lub innym obiektem na ekranie komputera (E2-PODST-INF-2.0-KLIVVI-II.1.b),
- sterowanie robotem lub obiektem na ekranie (E2-PODST-INF-2.0-KLIVVI-I.2.c),

TREŚCI NAUCZANIA – WYMAGANIA SZCZEGÓŁOWE: (PODSTAWA PROGRAMOWA)

- posługuje się technologią zgodnie z przyjętymi zasadami i prawem; przestrzega zasad bezpieczeństwa i higieny pracy (E2-PODST-INF-2.0-KLIVVI-V.1),
- testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów (E2-PODST-INF-2.0-KLIVVI-II.2),
- wykorzystuje komputer lub inne urządzenie cyfrowe do gromadzenia, porządkowania i selekcjonowania własnych zasobów (E2-PODST-INF-2.0-KLIVVI-III.1.b),
- w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu (E2-PODST-INF-2.0-KLIVVI-I.3),

- wymienia zagrożenia związane z powszechnym dostępem do technologii oraz do informacji i opisuje metody wystrzegania się ich (E2-PODST-INF-2.0-KLIVVI-V.3).

METODY PRACY:

- techniki informatyczne,
- Zintegrowana Platforma Edukacyjna,
- Prezentacja,
- praca indywidualna.

ŚRODKI DYDAKTYCZNE:

- komputer z systemem operacyjnym Windows 11,
- przeglądarka internetowa,
- ZOOM.

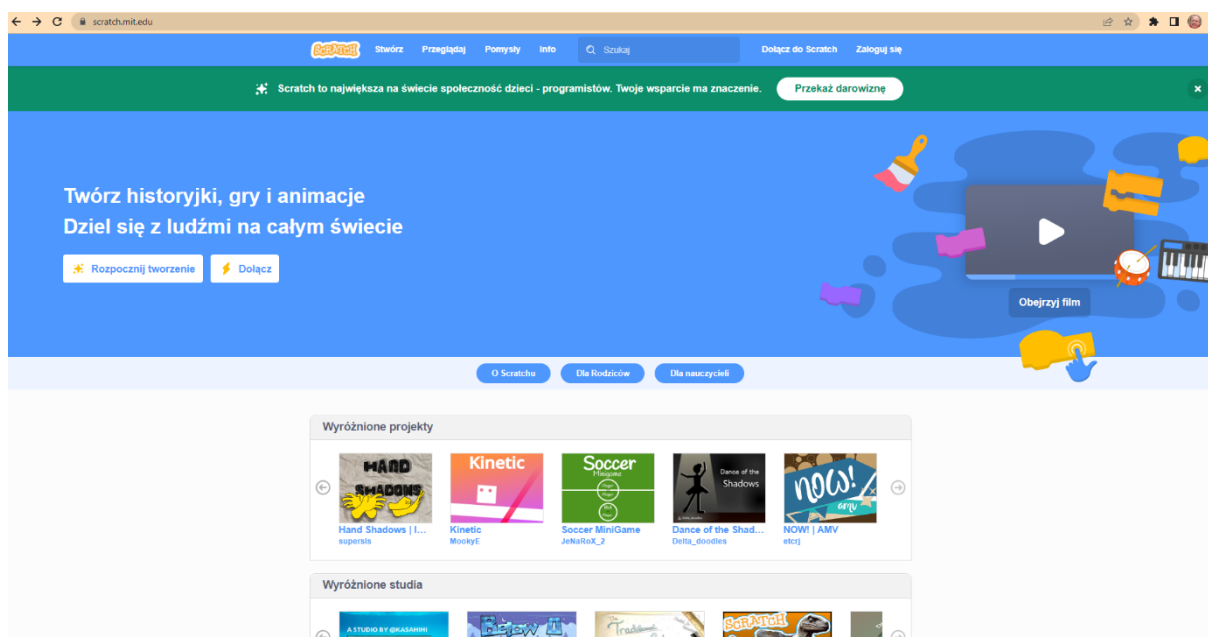
PRZEWIDYWANY CZAS: 45 minut

PROPONOWANY PRZEBIEG ZAJĘĆ:

Część 1: Przywitanie uczniów, sprawdzenie listy obecności, sprawdzenie komunikacji (mikrofony, słuchawki, kamery). Uczniowie są przy komputerach. Czas 3 minuty.

Część 2: Podanie celu zajęć: Nauczyciel objaśnia cel lekcji, którym jest opracowanie prostej gry w programie *Scratch*. Podaje etapy pracy (stworzenie tła, postaci duszków, zaprogramowanie duszków i przetestowaniu gry. Pisanie skryptu gry będzie tematem dwóch kolejnych lekcji. Na tej lekcji uczniowie wykonają tło gry, utworzą duszka robaczka i zaprogramują jej ruchy. Objasnia, że uczniowie posiadający komputery stacjonarne czy laptopy mogą wykonywać zadania równoległe z nauczycielem. Czas 3 minuty.

Część 3: Nauczyciel udostępnia swój ekran (przeglądarka internetowa) w systemie Zoom. Wzywa program z platformy: <https://scratch.mit.edu/>. Po wezwaniu platformy ekran przeglądarki ma postać:

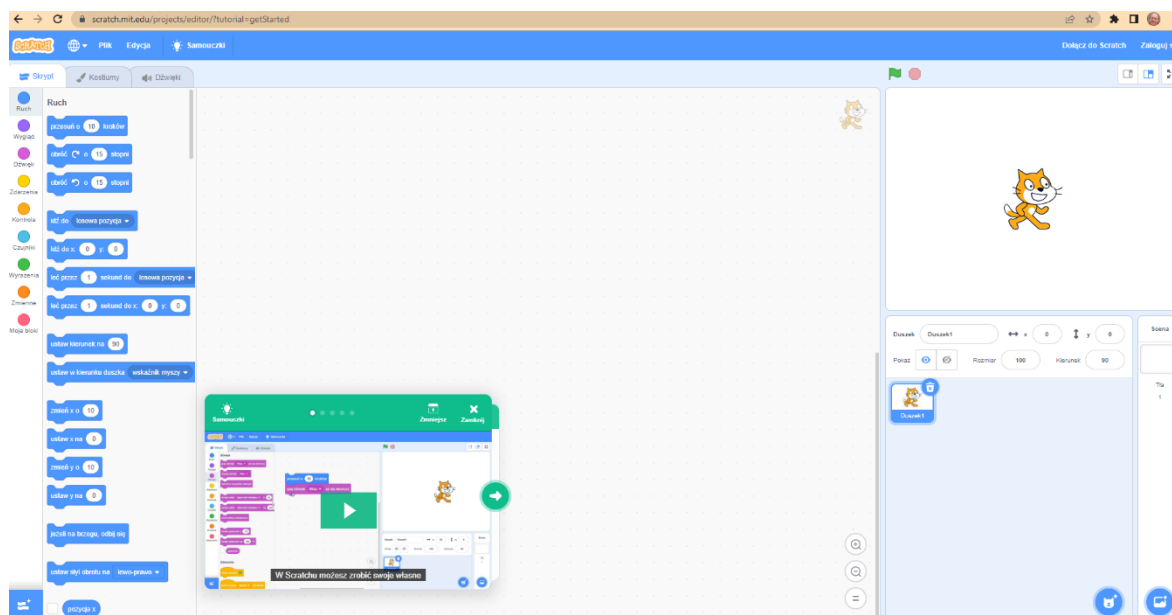


Rysunek 8: Ekran startowy Scratch (opracowanie własne).

Ćwiczenie 1: Stworzenie nowego skryptu, naniesienie tła i potrzebnego duszka.

Nauczyciel wydaje polecenie klasie i sam demonstruje udostępniając cały czas ekran swojego komputera. Czas 9 minut.

7. Kliknij na Stwórz w górnej części ekranu,

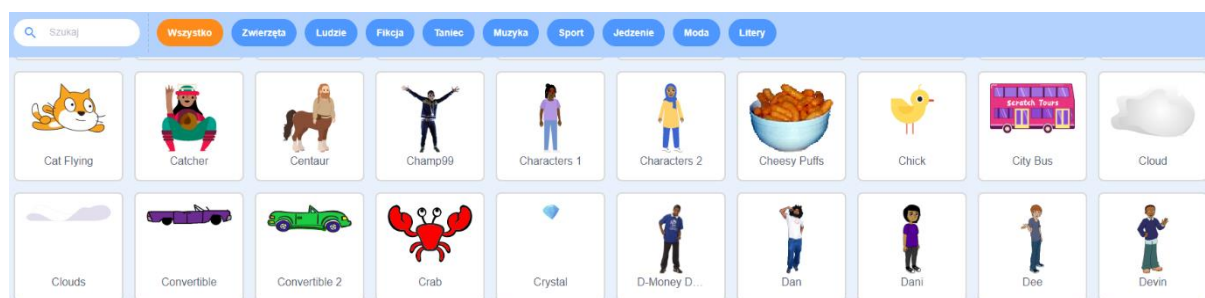


Rysunek 9: Pusty projekt Scratch (opracowanie własne).

8. Usuń duszka Duszek1 (kot)

9. Dodaj duszka kraba z galerii duszków. Nauczyciel przypomina, gdzie uczeń ma kliknąć, aby uruchomić galerię duszków.

10. Wybierz duszka Crab (krab).



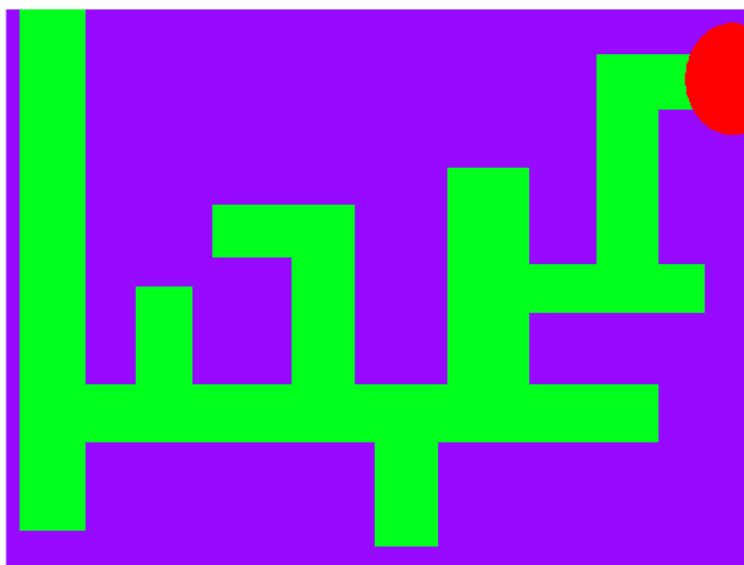
Rysunek 10: Crab (krab) w kolekcji duszków (opracowanie własne).

11. Narysuj tło.

- Kliknij na scenę (prawy dolny róg)
- Kliknij na tła (górny lewy róg)

12. Przy pomocy dostępnych narzędzi narysuj labirynt. Kolory ścieżek i ścian dobierz wg własnego uznania. Pamiętaj, że kolory mają się od siebie odróżniać – tworzyć kontrast.

13. Dodaj METĘ do labiryntu. Nauczyciel pokazuje swój przykład.



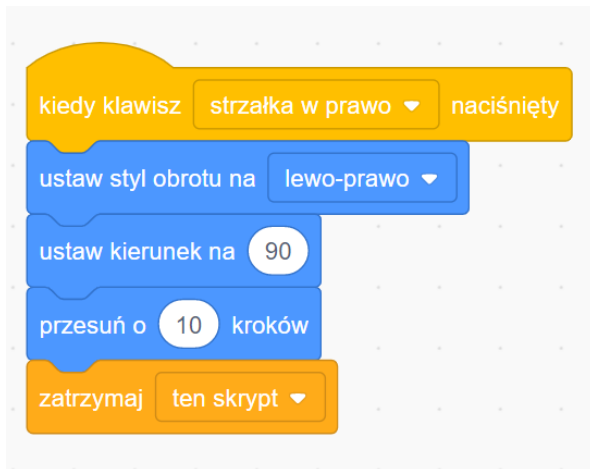
Rysunek 11: Przykładowy labirynt (opracowanie własne).

Ćwiczenie 2: Napisanie skryptu na przesuwanie się duszka w lewo, prawo, do góry i do dołu. Uczniowie przypominają sobie poruszanie się duszka z klasy 4. Nauczyciel, jeśli możliwe, przy pomocy uczniów pisze skrypt na ruch w prawo. Czas 10 minut

Wykaz rozkazów:

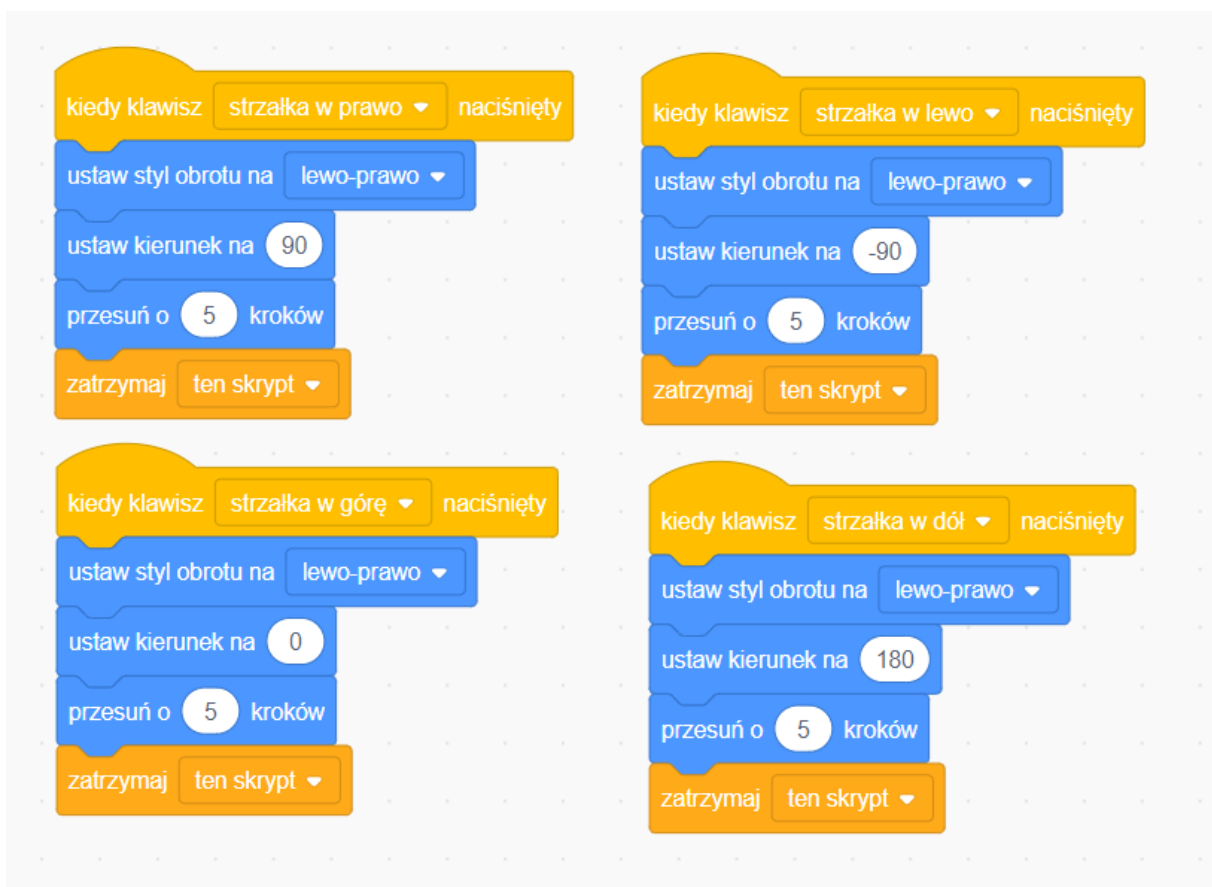
- **Grupa:** Zdarzenia, **rozkaz:** Kiedy klawisz strzałka w prawo naciśnięty,
- **Grupa:** Ruch, **rozkaz:** Ustal styl obrotów lewo-prawo,
- **Grupa:** Ruch, **rozkaz:** Ustaw kierunek na 90 stopni,

- **Grupa:** Ruch, **rozkaz:** Przesuń o 10 kroków,
- **Grupa:** Kontrola, **rozkaz:** Zatrzymaj ten skrypt.



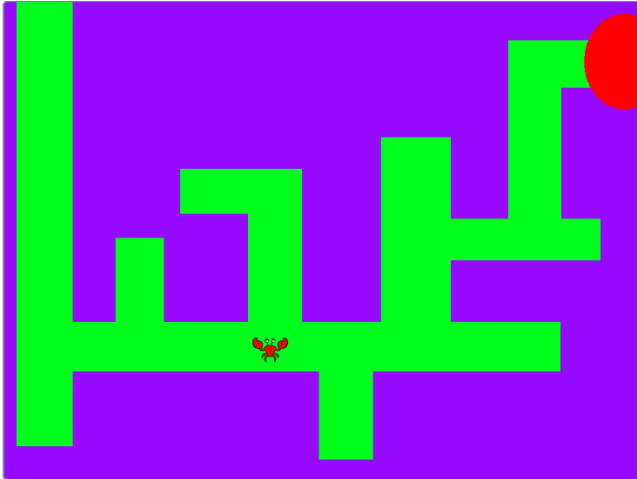
Rysunek 12: Skrypt, za pomocą którego przesuńiesz duszka o 10 kroków strzałką w prawo (*opracowanie własne*).

Ćwiczenie 3: Opracuj następne skrypty – duszek powinien poruszać się w każdym kierunku. Możesz wykorzystać duplikowanie skryptu. Czas 7 minut. Skrypt po wykonaniu powyższych czynności powinien wyglądać następująco:



Rysunek 13: Skrypty do przemieszczania się duszka (*opracowanie własne*).

Ćwiczenie 4: Dostosuj wielkość duszka kraba do wielkości Twojego labiryntu. Nauczyciel objaśnia, że krab musi swobodnie poruszać się w labiryncie. Przy pomocy właściwości rozmiar duszka krab zmień tak, aby duszek mieścił się w labiryncie. Przetestuj swój skrypt, to znaczy przeprowadź kraba z punktu wyjścia na metę. Czas 3 minuty.



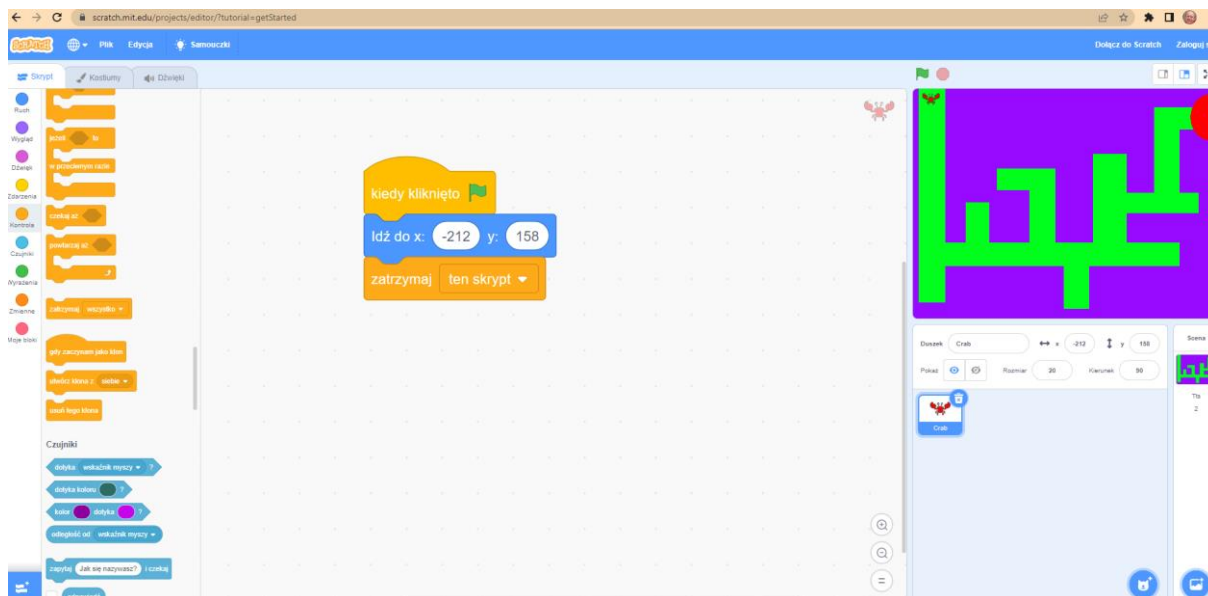
Rysunek 14: Prawidłowo dobrana wielkość kraba do chodnika w labiryncie. (opracowanie własne).

Ćwiczenie 5. Ustaw duszka w punkcie startu przy pomocy myszki (lub innego urządzenia wskazującego). Zobacz wartość x i y dla tego punktu. Napisz pierwszy skrypt duszka. Zadania skryptu:

- Reakcja na zieloną flagę
- Ustawienie w punkcie startowym

Czas 5 minuty.

Nauczyciel przedstawia swój ekran ze skryptem do porównania:



Rysunek 15: Ekran na koniec lekcji (*opracowanie własne*).

Część : Zapisanie na własnym dysku, zadanie pracy domowej. (5 minut)

Nauczyciel przypomina uczniom o konieczności zapisania własnej pracy na komputerze lub we własnej chmurze *Scratch*. Uczniowie to wykonują. Nauczyciel kończy udostępnianie ekranu w Zoom. Zadaje pytanie odnośnie trudności w wykonaniu pracy i wyjaśnia wątpliwości.

EWALUACJA ZAJĘĆ (sprawdzenie osiągnięcia zakładanych efektów kształcenia)

Praca domowa: Wprowadź do skryptu zmienne:

- Punkty – będą liczone, jak krab zje coś dobrego
- Czas – ile czasu potrzebuje gracz na przejście labiryntu.

Skrypt zapisz na swoim komputerze lub w swojej chmurze *Scratch*.

BIBLIOGRAFIA

Opracowanie własne

ZAŁĄCZNIKI

EkranNaKoniecLekcji.png

EkranScratch.png

EkranScratchNowySkrypt.png

PrzykładowyLabiryntMeta.png

RozkazyDoRuchuWPravo.png

SkryptyDuszkaWszystkieRuchy.png

WybórKraba.png

WyskalowanyKrab.png

Labirynt-1.sb3

SCENARIUSZ 3 z 5

SCENARIUSZ ZAJĘĆ DLA: UCZNIÓW KLASY VI SZKOŁY PODSTAWOWEJ I SEMESTR

PROWADZONYCH PRZEZ nauczyciela informatyki

TEMAT:

ROZWIJANIE UMIEJETNOŚCI PROGRAMOWANIA. PISZEMY SKRYPT
ODSZUKANIA NAJWIĘKSZEJ WARTOŚCI NA LIŚCIE W PROGRAMIE SCRATCH.

CELE KSZTAŁCENIA – WYMAGANIA OGÓLNE: (PODSTAWA PROGRAMOWA)

- osiągnięcie postawionego celu, w tym znalezienie elementu w zbiorze nieuporządkowanym lub uporządkowanym, znalezienie elementu najmniejszego i największego (E2-PODST-INF-2.0-KLIVVI-I.2.b)

TREŚCI NAUCZANIA – WYMAGANIA SZCZEGÓŁOWE: (PODSTAWA PROGRAMOWA)

- posługuje się technologią zgodnie z przyjętymi zasadami i prawem; przestrzega zasad bezpieczeństwa i higieny pracy (E2-PODST-INF-2.0-KLIVVI-V.1),
- testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów (E2-PODST-INF-2.0-KLIVVI-II.2),
- w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu (E2-PODST-INF-2.0-KLIVVI-I.3),
- wykorzystuje komputer lub inne urządzenie cyfrowe do gromadzenia, porządkowania i selekcjonowania własnych zasobów (E2-PODST-INF-2.0-KLIVVI-III.1.b),
- w algorytmicznym rozwiązywaniu problemu wyróżnia podstawowe kroki: określenie problemu i celu do osiągnięcia, analiza sytuacji problemowej, opracowanie rozwiązania, sprawdzenie rozwiązania problemu dla

przykładowych danych, zapisanie rozwiązania w postaci schematu lub programu (E2-PODST-INF-2.0-KLIVVI-I.3),

- wymienia zagrożenia związane z powszechnym dostępem do technologii oraz do informacji i opisuje metody wystrzegania się ich (E2-PODST-INF-2.0-KLIVVI-V.3).

METODY PRACY:

- techniki informatyczne,
- Zintegrowana Platforma Edukacyjna,
- Prezentacja,
- praca indywidualna.

ŚRODKI DYDAKTYCZNE:

- komputer z systemem operacyjnym Windows 11,
- przeglądarka internetowa,
- ZOOM.

PRZEWIDYWANY CZAS: 45 minut

PROPONOWANY PRZEBIEG ZAJĘĆ:

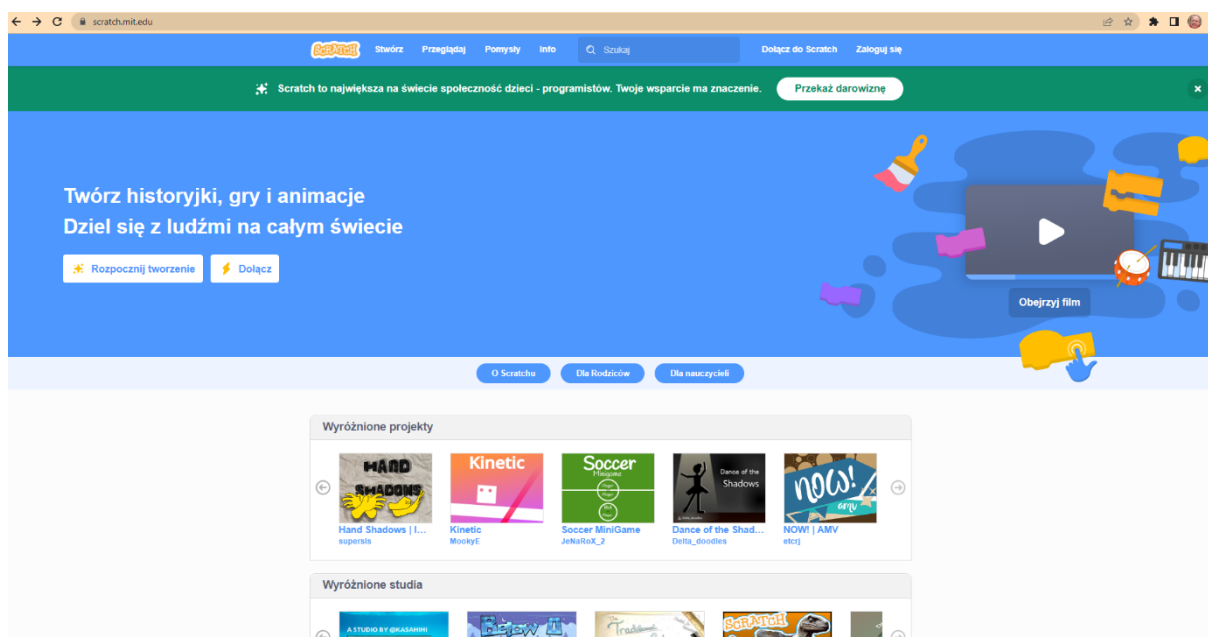
Część 1: Przywitanie uczniów, sprawdzenie listy obecności, sprawdzenie komunikacji (mikrofony, słuchawki, kamery). Uczniowie są przy komputerach. Czas 3 minuty.

Część 2: Podanie celu zajęć: Nauczyciel objaśnia cel lekcji, którym jest opracowanie programu w *Scratch*, którego zadaniem jest:

1. Po naciśnięciu klawisza L nastąpi uruchomienie skryptu, który wylosuje sześć liczb w zakresie od 1 do 100,
2. Po naciśnięciu klawisza N nastąpi uruchomienie skryptu, który znajdzie największą liczbę spośród wylosowanych.

Objaśnia, że uczniowie posiadający komputery stacjonarne czy laptopy mogą wykonywać zadania równoległe z nauczycielem. Czas 3 minuty.

Część 3: Nauczyciel udostępnia swój ekran (przeglądarka internetowa) w systemie Zoom. Wzywa program z platformy: <https://scratch.mit.edu/>. Po wezwaniu platformy ekran przeglądarki ma postać:



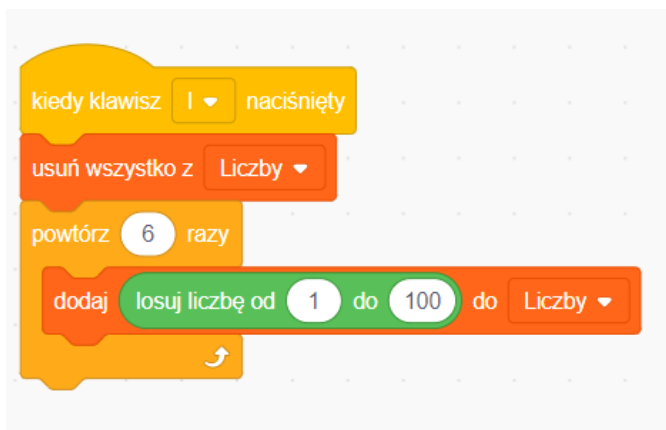
Rysunek 16: Ekran startowy *Scratch* (opracowanie własne).

Ćwiczenie 1: Przygotowanie danych. Nauczyciel poleca uczniom stworzenie nowej listy o nazwie „Liczby”. Prosi o zgłoszenie się chętnego, który udostępni swój ekran i poda czynności, które należy wykonać. Czas 2 minuty.

Ćwiczenie 2: Wypełnienie listy „Liczby” losowymi danymi. Nauczyciel pokazuje czynności, które należy wykonać, aby wypełnić listę liczbami losowymi:

1. **Zdarzenia** – kiedy klawisz „L” naciśnięty,
2. **Zmienne** – usuń wszystko z Liczby,
3. **Kontrola** – powtórz 6 razy, a w pętli,
4. **Zmienne** – dodaj obiekt do Liczby, w miejsce obiekt wstaw **Wyrażenia** – losuj obiekt (dostosuj: 1 do 100)
5. **Kontrola** – zatrzymaj ten skrypt.

Uczniowie w zadanym czasie opracowują skrypt na własnym komputerze. Na minutę przed końcem zadanego czasu nauczyciel pokazuje przykładowy skrypt. Czas 10 minut.



Rysunek 17: Skrypt losujący liczby od 1 do 100 po naciśnięciu klawisza L (*opracowanie własne*).

Ćwiczenie 3: Porównanie zawartości listy „Liczby”. Nauczyciel prosi ucznia, który zgłasza się o wyjaśnienie istoty liczb losowych. Czas 2 minuty.

Ćwiczenie 4: Napisz skrypt na znalezienie liczby największej. Nauczyciel zwraca uwagę, że u każdego ucznia liczba jest inna, gdyż nastąpiło wylosowanie liczby. Czas 18 minut.

Część 1: Pytanie do uczniów: jak wzrokowo szukasz liczby największej? Nauczyciel oczekuje odpowiedzi, że człowiek porównuje po kolei liczby i zapamiętuje największą.

Część 2: Co w programie *Scratch* służy do zapamiętania wartości? – odpowiedź zmienne.

Część 3: Ile razy muszę dokonać porównania, aby odszukać liczbę największą? – Odpowiedź 6 razy.

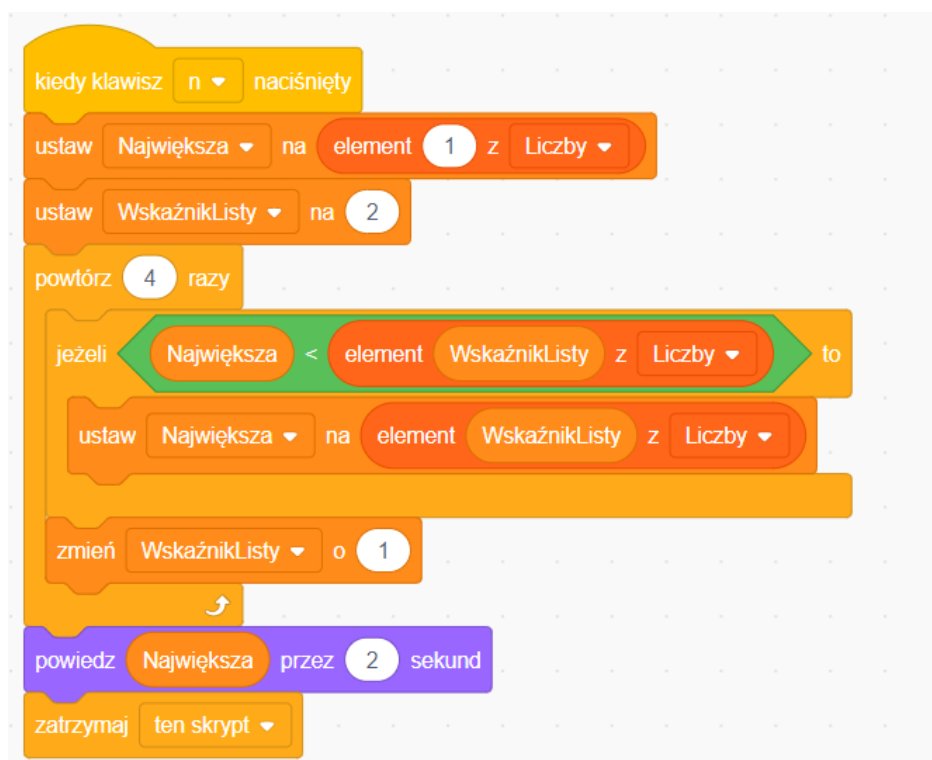
Część 4: Uczniowie podpowiadają nauczycielowi jak należy napisać skrypt. Nauczyciel ocenia podpowiedzi. Błędne podpowiedzi są korygowane logicznymi argumentami i ewentualnie przykładami, dlaczego dana podpowiedź jest błędna. Nauczyciel przypomina uczniom, że do przejścia pomiędzy elementami listy potrzebny jest wskaźnik, czyli zmienna. Zmienna wskazuje, na którym elemencie listy jest skrypt

Po wspólnej dyskusji nauczyciel przedstawia opis wspólnie opracowanego programu.

1. **Zmienne** – utwórz zmienną Największa,
2. **Zmienne** – otwórz zmienną Wskaźnik,
3. **Zdarzenia** – kiedy klawisz „N” naciśnięty,

4. **Zmienne** – ustaw Największa na element 1 z listy,
5. **Zmienne** – ustaw Wskaźnik na 2,
6. **Kontrola** – powtórz 4 razy
7. **Kontrola** – jeżeli Największa < element z listy, który wskazuje teraz wskaźnik
8. **Zmienne** – ustaw Największa na element z listy, który wskazuje teraz wskaźnik
9. **Zmienne** – zmień wskaźnik o 1
10. **Wygląd** – powiedz Największa przez 2 sekundy.
11. **Kontrola** – zatrzymaj ten skrypt.

Po 10 minutach samodzielnej pracy nauczyciel udostępniając ekran wyświetla skrypt.



Rysunek 18: Odnalezienie i wyświetlenie największej liczby (*opracowanie własne*).

Część 5. Nauczyciel prosi wyrywkowo uczniów o objaśnienie, co znaczą poszczególne rozkazy. Czas 5 minut

Część 6: Zakończenie. Zapisanie swojej pracy w chmurze *Scratch* pod nazwą „LiczbaNajwiększaNaLiście” Udzielenie odpowiedzi na ewentualne wątpliwości, zadanie i objaśnienie pracy domowej.

EWALUACJA ZAJĘĆ (sprawdzenie osiągnięcia zakładanych efektów kształcenia)

Zmień skrypt napisany na lekcji wg następujących kryteriów:

1. Wylosuj 100 liczb w zakresie od -100 do 100
2. Znajdź i wyświetl liczbę najmniejszą.

BIBLIOGRAFIA

Opracowanie własne

ZAŁĄCZNIKI

EkranScratch.png

SkryptLosowanie_1-100.png

SkryptOdszukanieNajwiększejLiczby.png

ZnajdzLiczbeNajwieksza.sb3

SCENARIUSZ 4 z 5

SCENARIUSZ ZAJĘĆ DLA: UCZNIÓW KLASY VII SZKOŁY PODSTAWOWEJ I SEMESTR

PROWADZONYCH PRZEZ nauczyciela informatyki

TEMAT:

ROZWIJANIE UMIEJETNOŚCI EDYCJI ZDJĘĆ. KORYGUJEMY ZDJĘCIA W PROGRAMIE GIMP.

CELE KSZTAŁCENIA – WYMAGANIA OGÓLNE: (PODSTAWA PROGRAMOWA)

- tworzenia estetycznych kompozycji graficznych: tworzy kolaże, wykonuje zdjęcia i poddaje je obróbce zgodnie z przeznaczeniem, nagrywa krótkie filmy oraz poddaje je podstawowej obróbce cyfrowej (E2-PODST-INF-2.0-KLVIII-II.3.a).

TREŚCI NAUCZANIA – WYMAGANIA SZCZEGÓŁOWE: (PODSTAWA PROGRAMOWA)

- zapisuje efekty swojej pracy w różnych formatach i przygotowuje wydruki (E2-PODST-INF-2.0-KLVIII-II.4).

METODY PRACY:

- techniki informatyczne,
- Zintegrowana Platforma Edukacyjna,
- Prezentacja,
- praca indywidualna.

ŚRODKI DYDAKTYCZNE:

- komputer z systemem operacyjnym Windows 11 oraz zainstalowanym bezpłatnym programem GIMP w aktualnej wersji,
- przeglądarka internetowa,
- ZOOM.

PRZEWIDYWANY CZAS: 40 minut

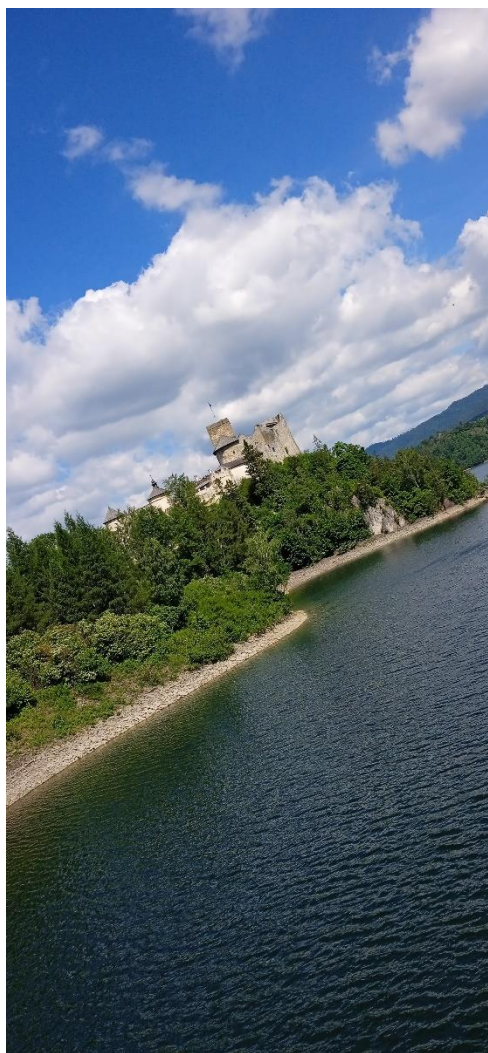
PROPONOWANY PRZEBIEG ZAJĘĆ:

Część 1: Przywitanie uczniów, sprawdzenie listy obecności, sprawdzenie komunikacji (mikrofony, słuchawki, kamery). Uczniowie są przy komputerach. Czas 3 minuty.

Część 2: Podanie celu zajęć: Nauczyciel objaśnia cel lekcji, którym jest skorygowanie zrobionego zdjęcia. Przedstawione przez nauczyciela zdjęcie powinno być obrócone oraz skadrowane. Po wykonaniu tych czynności należy nanieść napis mówiący, co zdjęcie przedstawia.

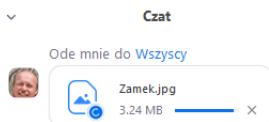
Objaśnia, że uczniowie posiadający komputery stacjonarne czy laptopy mogą wykonywać zadania równoległe z nauczycielem. Czas 3 minuty.

Część 3: Nauczyciel udostępnia swój ekran i pokazuje zdjęcie, które zostało zrobione szybko lub niedbale.

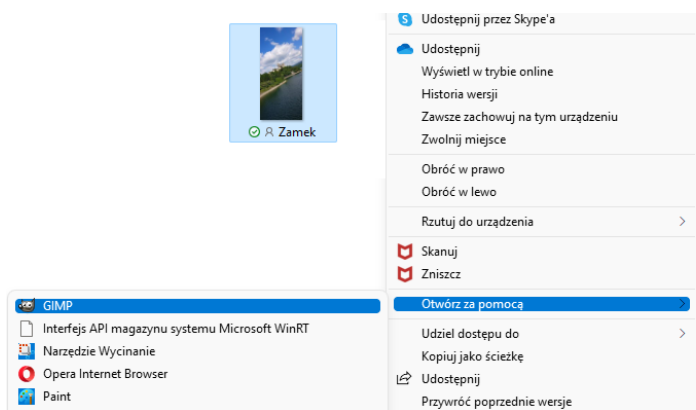


Rysunek 19: Niedbale wykonane zdjęcie. Obrócone i niekadrowane (opracowanie własne).

Część 4: Nauczyciel przesyła zdjęcie funkcją chat w programie Zoom a uczniowie pobierają ten plik i zapisują na swoim pulpicie w folderze Informatyka. Czas 2 minuty.



Rysunek 20: Przesłane pliku zamek.jpg wszystkim uczestnikom lekcji (opracowanie własne).
Część 5: Otworzenie zdjęcia. Nauczyciel i wszyscy uczniowie, którzy pracują równoległe otwierają otrzymane zdjęcie programem GIMP. Nauczyciel przypomina o możliwości skorzystania z opcji „Otwórz za pomocą” w menu uruchomionym prawym klawiszem myszy. Pokazuje na ekranie:

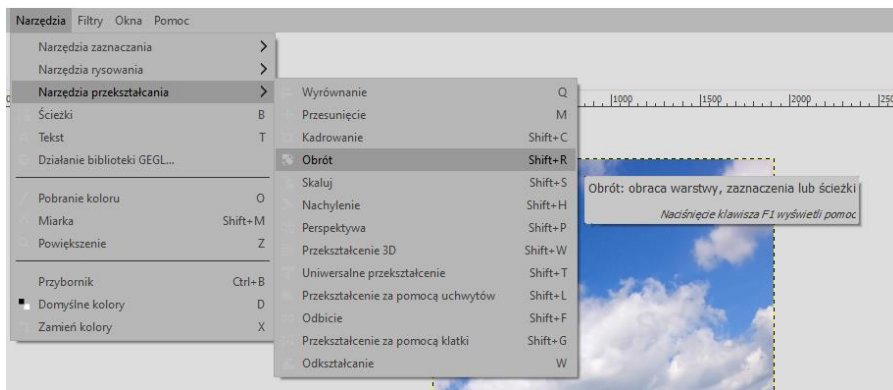


Rysunek 21: Otwórz za pomocą GIMP (opracowanie własne).

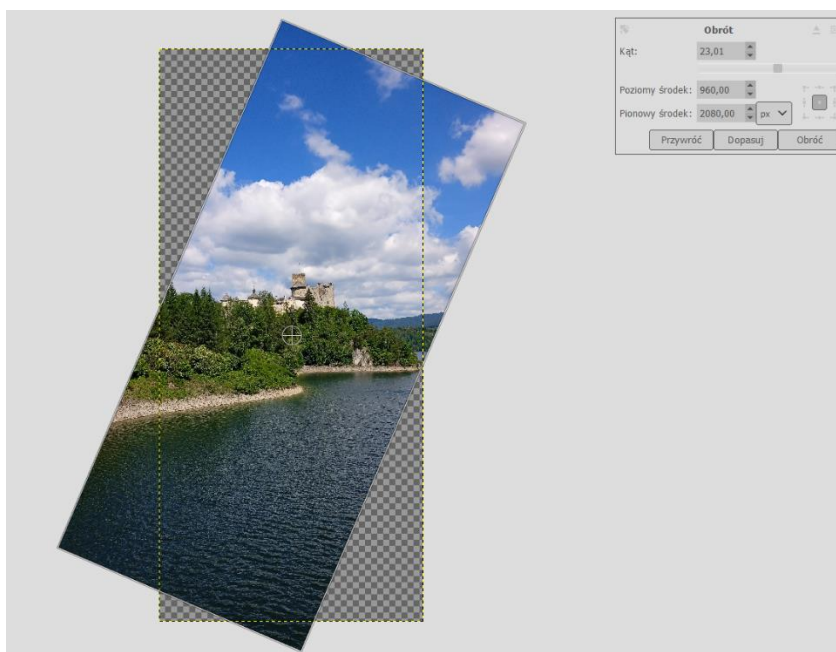
Czas 1 minuta.

Część 6: Obróbka zdjęcia. Nauczyciel pokazuje poszczególne czynności na swoim ekranie. Uczniowie wykonują równoległe. W przypadku problemów u uczniów, podnoszą rączki (na Zoom), udostępniają swój ekran a nauczyciel wyjaśnia, na czym polega problem. Czas: 19.

Ćwiczenie 1: Z menu Narzędzia -> Narzędzia przekształcania wybierz Obrót.

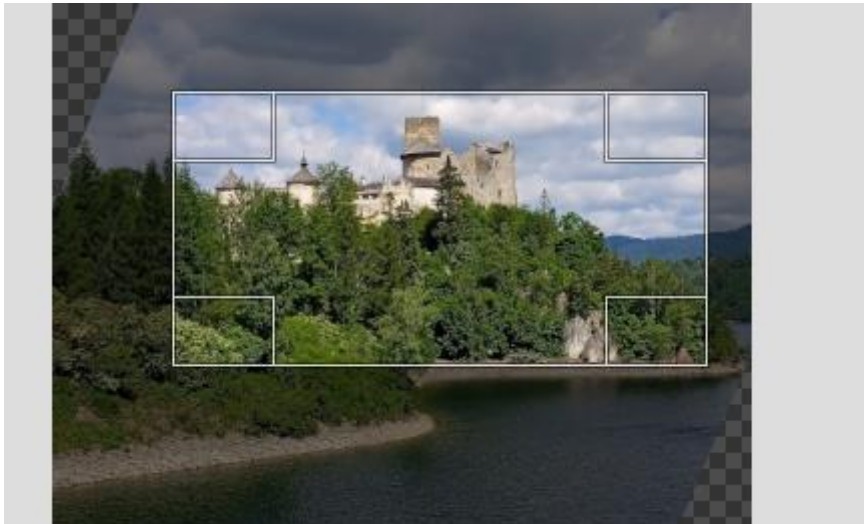


Rysunek 22: Opcja Obrót w menu narzędzia przekształcania (opracowanie własne).
 Posługując się myszką obróć tak zdjęcie, aby zamek stał pionowo.



Rysunek 23: Obrót zdjęcia – przeciągając myszkę (opracowanie własne).
 Po poprawnym obróceniu kliknij na Obróć

Ćwiczenie 2: Kadrowanie. Nauczyciel wydaje polecenie, aby wezwać opcję kadrowanie z menu Narzędzia przekształcania i wykadrować obraz w podobny sposób, jak robi to nauczyciel.



Rysunek 24: Kadrowanie zdjęcia (opracowanie własne).

Ćwiczenie 3: Wstawianie opisu zdjęcia

- Wstaw nową przezroczystą warstwę. Nazwij ją Opis
- Na tej warstwie wstaw napis „Zamek Dunajec w Niedzicy”
- Po wykonaniu tego zadania otrzymujemy obraz:



Rysunek 25: Cel zajęć (opracowanie własne).

Część 7: Zapisanie projektu i eksport do różnych formatów. Czas 5 minut.

Ćwiczenie 1: Zapisz projekt na pulpicie w folderze Informatyka

Ćwiczenie 2: Wyeksportuj efekt swojej pracy do pliku Imie_Nazwisko_r1.png

Plik umieść w chmurze Onedrive.

Część 8: Odpowiedzi na pytania, objaśnienia do pracy domowej (czas 5 minut)

EWALUACJA ZAJĘĆ (sprawdzenie osiągnięcia zakładanych efektów kształcenia)

Zrób własnym aparatem zdjęcie obrócone wybranego przez siebie obiektu.

Następnie obróć je w programie GIMP, wykadruj i wyeksportuj jako imie_nazwisko_r2.png do folderu Informatyka -> GIMP na Onedrive.

BIBLIOGRAFIA

Opracowanie własne

ZAŁĄCZNIKI

GimpKadrowanie.png

GimpObracanie.png

GimpObrót.png

OtwórzZaPomocą.png

PrzesłanieZdjęciaNaCzacie.png

Zamek.jpg

ZdjęciePoObróbce.png

SCENARIUSZ 5 z 5

SCENARIUSZ ZAJĘĆ DLA: UCZNIÓW KLASY VIII SZKOŁY PODSTAWOWEJ I SEMESTR

PROWADZONYCH PRZEZ nauczyciela informatyki

TEMAT:

ROZWIJANIE UMIEJETNOŚCI PROGRAMOWANIA. SZUKAMY W ZBIORZE LICZBY NAJWIĘKSZEJ METODĄ NAIWNĄ.

CELE KSZTAŁCENIA – WYMAGANIA OGÓLNE: (PODSTAWA PROGRAMOWA)

- Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji (E2-PODST-INF-2.0-KLVIIIIII-II.1).

TREŚCI NAUCZANIA – WYMAGANIA SZCZEGÓŁOWE: (PODSTAWA PROGRAMOWA)

- formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków (E2-PODST-INF-2.0-KLVIIIIII-I.1)
- projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice. W szczególności programuje algorytmy z działu I pkt 2 (E2-PODST-INF-2.0-KLVIIIIII-II.1);
- wyszukiwania i porządkowania: wyszukuje element w zbiorze uporządkowanym i nieuporządkowanym oraz porządkuje elementy w zbiorze metodą przez proste wybieranie i zliczanie (E2-PODST-INF-2.0-KLVIIIIII-I.2.b)
- rozwija znajomość algorytmów i wykonuje eksperymenty z algorytmami, korzystając z pomocy dydaktycznych lub dostępnego oprogramowania do demonstracji działania algorytmów (E2-PODST-INF-2.0-KLVIIIIII-I.4)

METODY PRACY:

- techniki informatyczne,

- Zintegrowana Platforma Edukacyjna,
- Metoda podająca,
- praca indywidualna.

ŚRODKI DYDAKTYCZNE:

- komputer z systemem operacyjnym Windows 11
- przeglądarka internetowa,
- ZOOM.

PRZEWIDYWANY CZAS: 40 minut

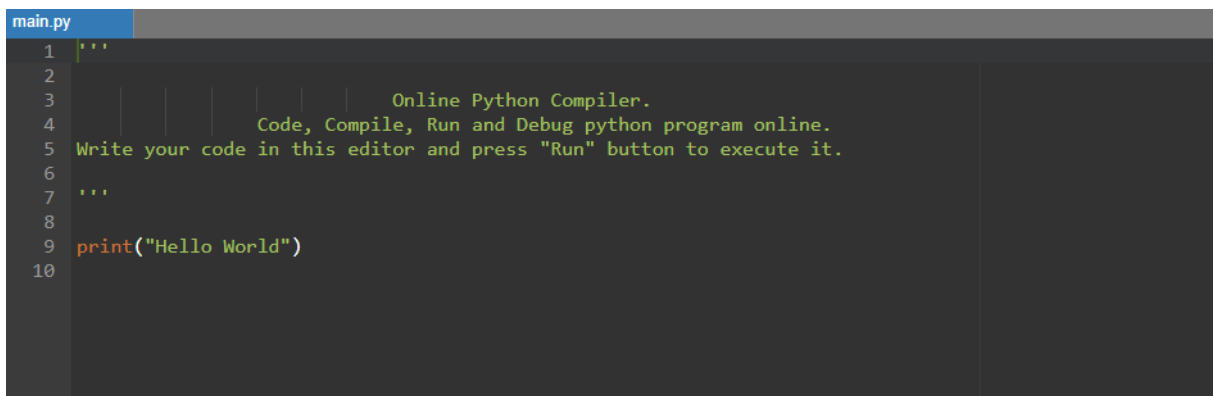
PROPONOWANY PRZEBIEG ZAJĘĆ:

Część 1: Przywitanie uczniów, sprawdzenie listy obecności, sprawdzenie komunikacji (mikrofony, słuchawki, kamery). Uczniowie są przy komputerach. Czas 3 minuty.

Część 2: Podanie celu zajęć: Nauczyciel objaśnia cel lekcji, którym wyszukanie największej liczby z listy nieuporządkowanej w programie Python.

Objaśnia, że uczniowie posiadający komputery stacjonarne czy laptopy mogą wykonywać zadania równoległe z nauczycielem. Czas 3 minuty.

Część 3: Nauczyciel udostępnia swój ekran na którym uruchomiona jest platforma https://www.onlinegdb.com/online_python_compiler. Uczniowie pracujący na lekcji równoległe, wykonują to samo.



```

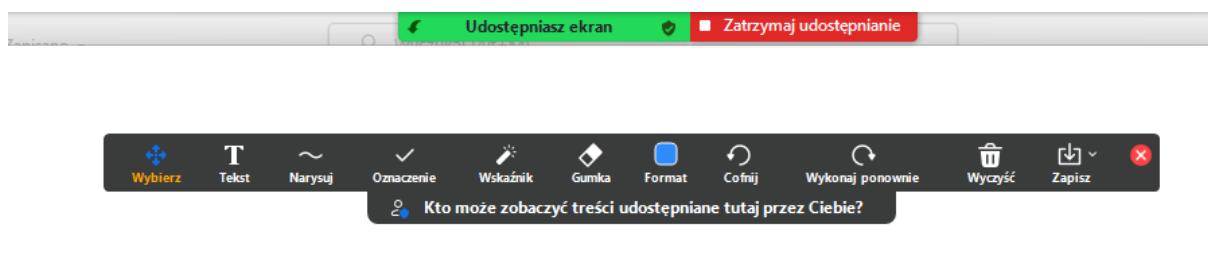
main.py
1 '''
2
3         Online Python Compiler.
4         Code, Compile, Run and Debug python program online.
5 Write your code in this editor and press "Run" button to execute it.
6
7 '''
8
9 print("Hello World")
10

```

Rysunek 26: Ekran onlinegdb (Opracowanie własne)

Część 4: Napisanie programu. Czas 24 minuty.

Ćwiczenie 1: Uczniowie mają utworzyć listę, którą nauczyciel zapisuje na tablicy, jako



`tablica = { 15, -5, 45, 86, 220, -9, 18 }`

Rysunek 27: Tablica z zapisanym zbiorem (Opracowanie własne)

Nauczyciel wybiera ucznia. Uczeń ma udostępnić swój ekran i zapisać taką tablicę w programie Python. W przypadku, kiedy uczeń ma problem z zapisem, nauczyciel prosi o zgłoszenie się innego ucznia do pomocy. Na ekranie wybranego ucznia, jako efekt końcowy powinno pojawić się

```
main.py
1 tablica=[15,-5,45,86,220,-9,18]
```

Rysunek 28: Deklaracja tablicy (Opracowanie własne)

Ćwiczenie 2: Deklaracja zmiennej *Najmniejsza*.

Nauczyciel wybiera innego ucznia i prosi o udostępnienie ekranu i napisanie w następnej linii deklaracji zmiennej *Najmniejsza*. W przypadku, kiedy uczeń ma problem z zapisem, nauczyciel prosi o zgłoszenie się innego ucznia do pomocy. Na ekranie wybranego ucznia, jako efekt końcowy powinno pojawić się:

```
main.py
1 tablica=[15,-5,45,86,220,-9,18]
2 Najmniejsza=None
```

Rysunek 29: Deklaracja zmiennej w Python (Opracowanie własne)

Ćwiczenie 3: Jakbyś wybrał liczbę najmniejszą? Uczniowie mówią, że porównują każdą liczbę ze zbioru do zmiennej *Najmniejsza*. Jeśli w zbiorze znaleziona zostanie liczba mniejsza od *Najmniejszej* to następuje podmiana liczby. Uczniowie wyjaśniają co to jest iteracja oraz jak zapisać iterację w programie Python z uwzględnieniem iteracji na liście. Przykładowy zapis iteracji w przypadku z lekcji


```
tablica=[15,-5,45,86,220,-9,18]
Najmniejsza=None
for element in tablica:
```

Rysunek 30: Iteracja w Python (Opracowanie własne)

Ćwiczenie 4: Zapisanie dyrektywy *if*. Następny uczeń udostępnia swój ekran i zapisuje dyrektywę *if*. W przypadku, kiedy uczeń ma problem z zapisem, nauczyciel prosi o zgłoszenie się innego ucznia do pomocy. Na ekranie wybranego ucznia, jako efekt końcowy powinno pojawić się:

```
1 tablica=[15,-5,45,86,220,-9,18]
2 Najmniejsza=None
3 for element in tablica:
4     if Najmniejsza==None or Najmniejsza >element:
5         Najmniejsza=element
```

Rysunek 31: Skrypt na wyszukanie najmniejszego elementu w zbiorze (Opracowanie własne)

Ćwiczenie 5: Wydruk najmniejszej liczby. Nauczyciel prosi o zgłoszenie się chętnej osoby do zakończenia programu – czyli wydrukowanie liczby najmniejszej.

```
main.py
1 tablica=[15,-5,45,86,220,-9,18]
2 Najmniejsza=None
3 for element in tablica:
4     if Najmniejsza==None or Najmniejsza >element:
5         Najmniejsza=element
6 print("Liczba najmniejszą jest: ",Najmniejsza)
```

Rysunek 32: Kompletny program w Python (Opracowanie własne).

Część 5: Zmień tak napisany program, aby odszukał liczbę największą. Praca samodzielna.
Czas: 10 minut.

```
main.py
1 tablica=[15,-5,45,86,220,-9,18]
2 Najwieksza=None
3 for element in tablica:
4     if Najwieksza==None or Najwieksza < element:
5         Najwieksza=element
6 print("Liczba Największą jest: ",Najwieksza)
7
```

Rysunek 33: Program na odzyskanie liczby największej (Opracowanie własne).

Część 8: Podsumowanie lekcji. Odpowiedzi na pytania, objaśnienie pracy domowej.
Czas 5 minut

EWALUACJA ZAJĘĆ (sprawdzenie osiągnięcia zakładanych efektów kształcenia)

Napisz program obliczający sumę liczby najmniejszej i największej z
zapropionowanego zbioru.

BIBLIOGRAFIA

Opracowanie własne

ZAŁĄCZNIKI

EkranKompilatoraPython.png

PythonDeklaracjaTablicy.png

PythonDeklaracjaZmienej.png

PythonIteracja.png

PythonProgramKoncowy.png

PythonProgramKoncowyZWydrukiem.png

PythonProgramLiczbaNajwieksza.png

TablicaZeZbiorem.png