

Zadania do wykonania (1)

Na ocenę **3.0** należy przygotować podstawowe środowisko pracy.

Wskazówki:

- zainstalować niezbędne narzędzia i uruchomić przykładowy kod,
 - ▶ kod lepiej pobrać ze strony prowadzącego, niż kopiować ze slajdów,
- przerobić kod – narysować trójkąt z każdym wierzchołkiem innego koloru,
- proszę pamiętać o ustawionych parametrach rzutni:
 - ▶ zakres na osi X: od -100.0 (lewa strona) do 100.0 (prawa strona),
 - ▶ zakres na osi Y: od -100.0 (dół okna) do 100.0 (góra okna),
 - ▶ punkt o współrzędnych ($X = 0, Y = 0$) – w środku okna.

Zadania do wykonania (2)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **3.5** należy napisać funkcję rysującą prostokąt w podanym miejscu.

Wskazówki:

- nowa funkcja powinna przyjmować 4 argumenty:
 - ▶ położenie w osi X – x ,
 - ▶ położenie w osi Y – y ,
 - ▶ rozmiar pierwszego boku – a ,
 - ▶ rozmiar drugiego boku – b ;
- położenie (x, y) może wskazywać środek prostokąta lub jego wierzchołek,
 - ▶ punkt (x, y) określa się wtedy fachowo jako punkt początkowy,
 - ang. *origin point*,
 - ▶ na tej podstawie należy wyznaczyć współrzędne reszty wierzchołków bryły,
 - ▶ do narysowania prostokąta należy wykorzystać dokładnie dwa trójkąty,
- funkcję należy przykładowo wywołać w ramach `render()`.

Zadania do wykonania (3)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **4.0** należy wprowadzić losowość kolorów i deformacje w prostokącie.

Wskazówki:

- proszę rozbudować funkcję z poprzedniego zadania, na przykład:
 - ▶ dodać kolejny argument do funkcji – d – z domyślną wartością 0.0,
 - ▶ nowy argument powinien sterować stopniem deformacji,
 - ▶ można na przykład przeskalać rozmiary boków a i b ;
- uzyskanie losowej wartości w języku Python:
 - ▶ załadowanie biblioteki: `import random`;
 - ▶ przykładowe wywołanie: `random.random()`;
 - ▶ przydatne może być także użycie: `random.seed(...)`;
 - ▶ oficjalna dokumentacja: <https://docs.python.org/3/library/random.html>;
 - ▶ proszę pamiętać o zakresach wybranego wariantu funkcji `g1Color()`;
- w funkcji `render()` umieścić przykładowe wywołanie.

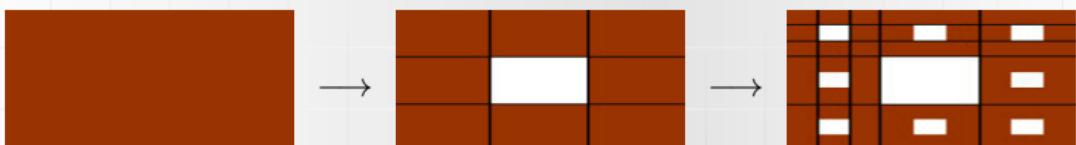
Zadania do wykonania (4)

(po zrealizowaniu zadania poprzedniego)

Na ocenę **4.5** należy narysować fraktal – prostokątny dywan Sierpińskiego.

Wskazówki:

- zasadniczo istnieją dwa podejścia do narysowania tego fraktalu:
 - ▶ rysować poszczególne małe prostokąty w wyznaczonych miejscach, lub
 - ▶ narysować duży prostokąt i pomniejsze w miejscach "wycięć";
- wykorzystać funkcje z poprzednich przykładów:
 - ▶ najpierw narysować zarys fraktalu z ręcznie rozmieszczonych brył,
 - tak wyznaczymy interesujące nas współrzędne prostokątów,
 - ▶ następnie ubrać całość w funkcję rekurencyjną,
 - powtórzyć rysowanie w wyznaczonych współrzędnych.
 - z każdym stopniem rekurencji pomniejszać rozmiary boków;
- stopień samopodobieństwa powinien być parametrem programu.



Zadania do wykonania (5)

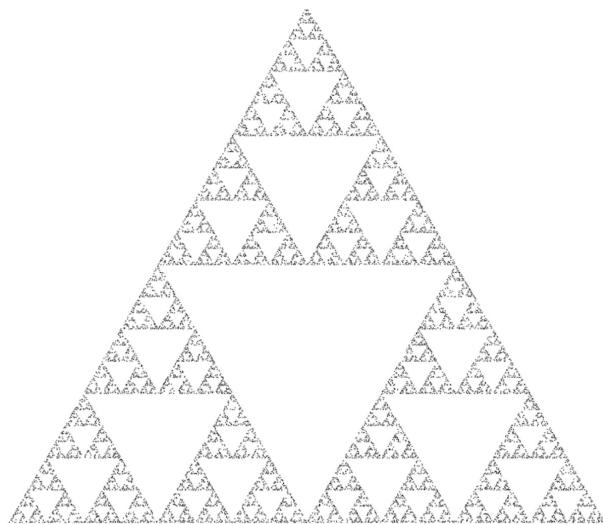
(po zrealizowaniu zadania poprzedniego)

Na ocenę **5.0** należy narysować drugi fraktal.

Wskazówki:

- wybrać jeden z przykładów zaproponowanych jako "zadania domowe",
 - ▶ dokument znajduje się na stronie prowadzącego,
 - ▶ interesują nas po prostu opisane tam pomysły na generację fraktali,
 - ▶ nie trzeba implementować różnych wariantów tego samego fraktala,
- alternatywnie można wykonać poprzedni fraktal w wariantie iteracyjnym,
 - ▶ warto zastanowić się nad dziedziną dozwolonych współrzędnych.

Trójkąt można także uzyskać także inną metodą, przez wyliczanie kolejnych punktów. Obraz trójkąta po wykonaniu pewnej liczby iteracji wygląda na przykład tak.



Zadanie do wykonania

Napisać program rysujący przybliżony obraz trójkąta Sierpińskiego. Program powinien umożliwiać rysowanie trójkąta przy pomocy dwóch algorytmów. Pierwszy opisano powyżej natomiast drugi, pozwalający na narysowanie obrazu trójkąta z kolejno wyliczanych punktów trzeba odszukać w literaturze. W obu przypadkach należy przy w sposób losowy (mniej więcej tak, jak to opisano w instrukcji) dobierać kolory rysowanych elementów.