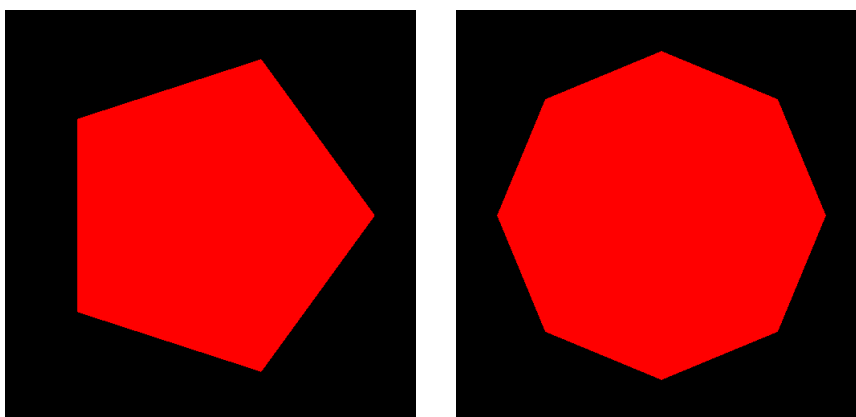


GRAFIKA INTERAKTYWNA

Zestaw 1

1. Napisać funkcję generującą geometrię oraz tworzącą obiekty VBO wielokąta foremnego. Argumentem tej funkcji ma być liczba kątów wielokąta oraz jego wielkość. Program główny ma rysować wygenerowany wielokąt używając do tego celu programu cieniowania (shader wierzchołków i fragmentów), który umożliwi rysowanie obiektu jednym ustalonym kolorem. Kolor ma być podany przez program główny tylko raz dla całego obiektu, a nie osobno dla każdego wierzchołka.

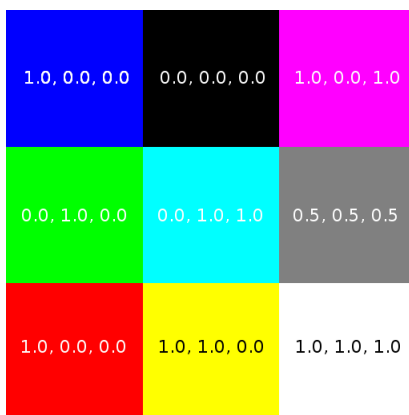


Wskazówka. Do wygenerowania wierzchołków wielokąta foremnego możemy wykorzystać wzór parametryczny okręgu:

$$\begin{aligned}x &= x_0 + r \cos \varphi, \\y &= y_0 + r \sin \varphi,\end{aligned}$$

gdzie $(x_0, y_0) \in \mathbb{R}^2$ to środek okręgu, $r > 0$ to promień okręgu, a $\varphi \in [0, 2\pi]$ jest parametrem. Aby wygenerować wierzchołki n kąta foremnego wystarczy przedział $[0, 2\pi]$ podzielić na n części (podprzedziałów) i obliczyć współrzędne (x, y) dla lewego końca każdego z podprzedziałów.

2. Załóżmy, że obszar w oknie $([-1, 1]^2)$ dzielimy na 9 kolorowych obszarów następująco:



Wewnątrz obszaru podany jest jego kolor w przestrzeni RGB. Napisać program cieniowania:

- (a) który każdemu wierzchołkowi rysowanego modelu (np. trójkąt, wielokąt) przydzieli kolor obszaru, w którym ten wierzchołek leży,
- (b) który każdemu fragmentowi rysowanego modelu przydzieli kolor obszaru, w którym ten fragment leży.

Jaka jest różnica w kolorowaniu modelu używając tych dwóch programów cieniowania?

3. Napisać program cieniowania, który dokona obrotu modelu 2D wokół początku układu współrzędnych. Kąt obrotu ma być podawany do shadera z programu głównego tylko raz. Dla przypomnienia wzór na obrót o kąt θ punktu (x, y) wokół początku układu współrzędnych ma postać:

$$\begin{aligned}x' &= x \cos \theta - y \sin \theta, \\y' &= x \sin \theta + y \cos \theta.\end{aligned}$$

4. Środek masy zbioru n punktów o masach m_i umiejscowionych w pozycjach p_i dany jest wzorem:

$$\bar{p} = \frac{\sum_{i=1}^n m_i p_i}{\sum_{i=1}^n m_i}.$$

W przypadku gdy wszystkie masy są sobie równe wzór upraszcza się do postaci:

$$\bar{p} = \frac{\sum_{i=1}^n p_i}{n}.$$

Napisać program, w którym kolor c każdego renderowanego punktu wielokąta o wierzchołkach $\{p_1, p_2, \dots, p_n\}$ będzie wyznaczany w następujący sposób:

$$c = (1 - t)R + tB,$$

gdzie R – kolor czerwony, B – kolor niebieski, a t dane jest wzorem:

$$t = \frac{d(p, \bar{p})}{d(p_m, \bar{p})},$$

gdzie $d : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow [0, +\infty)$ – odległość euklidesowa między punktami, p – punkt, który kolorujemy, \bar{p} – środek masy wielokąta (dla równych mas), p_m – wierzchołek wielokąta najbardziej oddalony od środka masy \bar{p} , tzn.

$$p_m = \arg \max_{p \in \{p_1, p_2, \dots, p_n\}} d(p, \bar{p}).$$

Przy pisaniu shaderów realizujących opisane kolorowanie przydatne mogą być funkcje z GLSL ES:

- odległość euklidesowa między p_0 i p_1 :
`float distance(vec2 p0, vec2 p1);`

- obliczenie wartości $(1 - t) * c0 + t * c1$:
`vec4 mix(vec4 c0, vec4 c1, float t);`

Przykładowe wielokąty pokolorowane zgodnie z opisanym sposobem:

