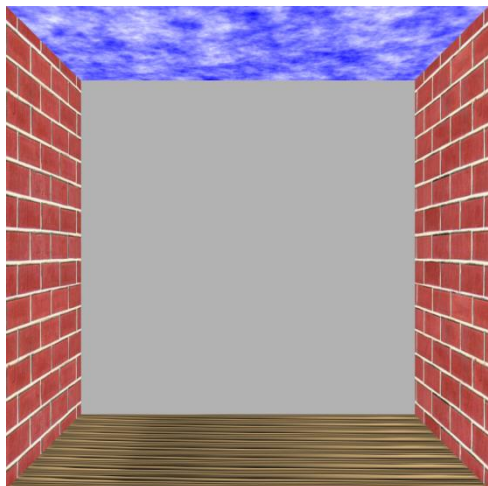


## Zadanie 1. Ściana

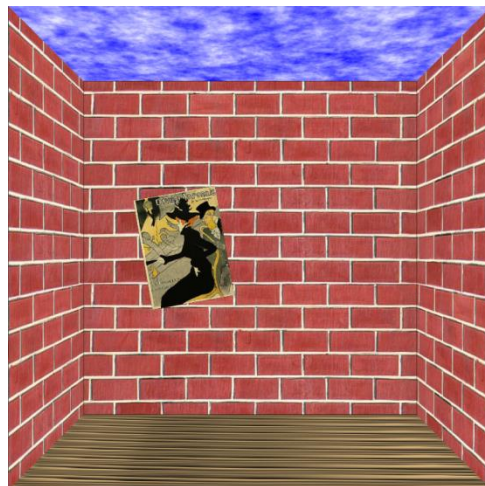
### Zadanie

Należy uzupełnić kod programu o odpowiednie parametry teksturowania tak, aby na ścianę, która w pierwotnej wersji jest jednolicie szara, zostały nałożone dwie tekstury: tekstura cegieł oraz tekstura plakatu w sposób w jaki widoczny jest na zrzucie ekranu poniżej:

**Przed:**



**Po:**



#### 1. Implementacja multiteksturowania z użyciem dwóch tekstur

##### *Wprowadzenie*

Multiteksturowanie to funkcjonalność pozwalająca w jednym przebiegu rysowania siatki trójkątów, pocieniować ich powierzchnię na podstawie funkcji kolorów uzyskanych zarówno z oświetlenia jak i pobranych w określony sposób z kilku kolejno wykorzystywanych tekstur. W multiteksturowaniu można wyróżnić kolejne etapy teksturowania. Z każdym etapem teksturowania związane są:

- obiekt tekstury,
- współrzędne tekstury dla każdego wierzchołka siatki trójkątów (lub sposób ich generowania na podstawie jego położenia lub normalnej z nim związanej),
- sposób połączenia koloru tekstury z kolorem uzyskanym z poprzedniego etapu teksturowania

Współrzędne tekstury w danym wierzchołku siatki definiowane mogą być niezależnie dla każdego etapu teksturowania.

##### *Wykonanie*

Wzorując się na shaderach z plikach textureVS.hlsl i texturePS.hlsl zdefiniuj zmodyfikuj pliki shaderów multiTexVS.hlsl oraz multiTexPS.hlsl tak, by kolory pobierane były z dwóch tekstur a następnie ze sobą mieszanie.

W klasie RoomDemo (pliki roomDemo.h/.cpp) należy w metodzie DrawWalls, przy rysowaniu tylnej ściany należy podmienić shadery na te służące do multiteksturowania. Nie należy zapomnieć o podpięciu do pixel shadera dwóch tekstur wejściowych: ceglanej ściany i plakatu.

2. Ustawienie właściwego trybu zawijania obrazka.

### *Wprowadzenie*

Tryb zawijania ustala się w momencie tworzenia samplera, poprzez ustawienie w strukturze D3D11\_SAMPLER\_DESC odpowiednich wartości w polach AddressU, AddressV oraz AddressW.

Najczęściej używane tryby zawijania obrazka:

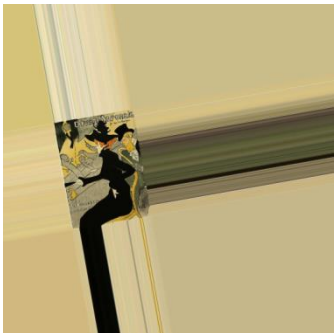
- Powtarzanie (D3D11\_TEXTURE\_ADDRESS\_WRAP)



- Odbicie (D3D11\_TEXTURE\_ADDRESS\_MIRROR)



- Obcięcie do krawędzi (D3D11\_TEXTURE\_ADDRESS\_CLAMP)



- Obcięcie do koloru brzegu (D3D11\_TEXTURE\_ADDRESS\_BORDER)



Kolor brzegu ustala się w tablicy BorderColor będącej polem wyżej wymienionej struktury.

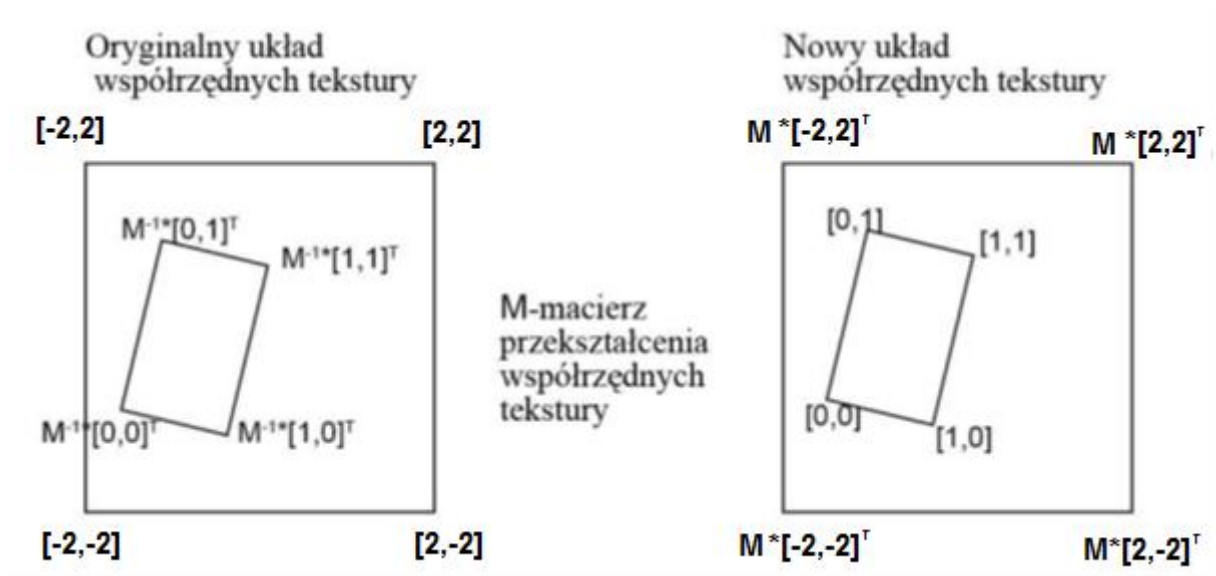
#### Wykonanie

W klasie RoomDemo dodać należy drugie pole samplera `m_samplerBorder`. W konstruktorze klasy RoomDemo należy zainicjalizować to pole tak, aby tekstury obcinane były do koloru brzegu, gdzie kolor brzegu ma składową alpha równą 0.

### 3. Pozycjonowanie tekstury plakatu.

#### Wprowadzenie

Należy wykorzystać macierz przekształcenia tekstury do ustalenia rozmiaru, pozycji i obrotu plakatu na ścianie. Plakat powinien mieć wymiary równe  $\frac{1}{4} \times \frac{1}{3}$  boku ściany. Środek plakatu powinien znajdować się w odległości 0.3 boku ściany od jej lewej krawędzi i 0.5 boku ściany od jej górnej krawędzi. Plakat powinien być obrócony w prawo o 10 stopni. Należy obliczyć macierz przekształcenia współrzędnych tekstury i ustawić ją dla tekstury plakatu.



#### Wykonanie

W metodzie `Room::CreateScene` należy do bufora `m_posterTexCB` wpisać odpowiednią macierz przekształcenia współrzędnych lokalnych wierzchołka do współrzędnych tekstury.

#### 4. Właściwe mieszanie koloru obu tekstur.

##### *Wprowadzenie*

Kolor wynikowy piksela powinien być interpolowany pomiędzy kolorem z pierwszej i z drugiej tekstury na podstawie kanału alpha pobranego z tekstury nr 2.

##### *Wykonanie*

W funkcji main shadera pikseli multiTexPS.hlsl należy pobrać kolory z obu tekstur w odpowiadających im współrzędnych tekstury i przeprowadzić liniową interpolację między nimi na podstawie współrzędnej a drugiego z tych kolorów.

Wtedy wszędzie poza prostokątem plakatu na ścianie, gdzie kanał alfa pobrany z tekstury plakatu będzie równy zero, pojawi się wyłącznie tekstura cegieł, a w obrębie prostokąta plakatu wyłącznie tekstura plakatu. Wcześniej zdefiniowany został sposób zawijania tekstury plakatu z obcięciem do koloru brzegu, który ma składową alfa równą zero, natomiast sama tekstura plakatu ma cztery składowe koloru (to jest ważne, bo nie można pobrać kanału alfa dla tekstury w formacie RGB) z czwartą składową alfa równą jeden dla każdego teksela. To zapewnia poprawne wartości kanału alfa dla przeprowadzenia powyższej operacji.

#### 5. Zmiana domyślnie wybieranego poziomu mipmap (rozmycie)

##### *Wprowadzenie*

Przy korzystaniu z mipmap dla każdego obrazka tekstury tworzone i pamiętane są również kolejne poziomy obrazków, z których każdy następny ma rozmiar dwukrotnie mniejszy od poprzedniego.

Poziom mipmap wykorzystywany przy włączonym filtrowaniu trzyliniowym lub anizotropowym ustalany jest na podstawie wielkości tekseli (punktów tekstury) na ekranie. Można zmienić standardowy sposób wyboru poziomu mipmap, dodając do niego liczbę rzeczywistą, przy czym dodanie jednostki wiąże się mniej więcej z użyciem poziomów mipmap o jeden większych (przy czym zwiększenie poziomu mipmap oznacza użycia obrazka o mniejszej rozdzielczości).

Poziomy mipmap mogą i najczęściej są generowane automatycznie. Teksel następnego poziomu odpowiada średniej wartości kilku (na przykład czterech) tekseli poprzedniego poziomu.

##### *Wykonanie*

Przy tworzeniu samplera m\_samplerBorder należy w polu MipLODBias struktury opisującej sampler podać rzeczywistą wartość dodatnią. Wtedy po uruchomieniu programu będzie można zobaczyć efekt osłabionego filtrowania.