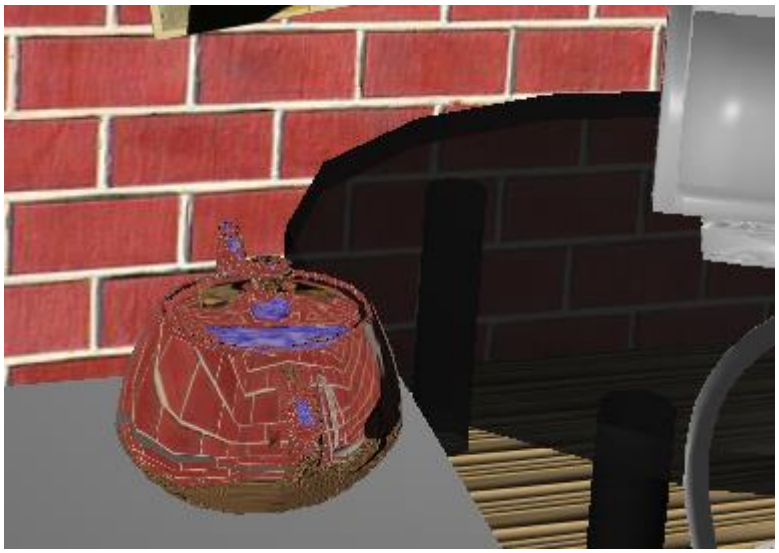
Zadanie 2. Mapowanie środowiska

Zadanie

W pliku „environmentMapper.cpp” należy uzupełnić konstruktor, i metody klasy oraz zmodyfikować shader wierzchołków zawarty w pliku „envMapVS.hlsl” i pikseli z pliku „envMapPS.hlsl” tak, aby uzyskać efekt odbić sceny w powierzchni obiektu czajnika. Wykonanie efektu mapowania środowiska składa się z dwóch kroków:

- renderowanie sceny widzianej ze środka obiektu, w którym ma się odbijać scena, do każdej ze ścian mapy sześciennej
- renderowania obiektu, w którym ma się odbijać scena, odpowiednio teksturowanego przy użyciu mapy sześciennej („envMapVS.hlsl” i „envMapPS.hlsl”)



1. Inicjalizacja tekstur.

W klasie EnvironmentMapper zdefiniowane są potrzebne pola:

- m_nearPlane, m_farPlane – odległość bliższej i dalszej płaszczyzny obcinania dla kamery podczas renderowania sceny do mapy sześciennej.
- m_position – położenie środka czajnika w układzie sceny.
- TEXTURE_SIZE – rozmiar pojedynczej ściany w teksturze sześciennej.

Dodatkowo zdefiniowane są pola dla tekstur i odpowiadających im widoków:

- m_envTexture – tekstura sześcienna, wykorzystywana przy mapowaniu środowiska
- m_envView – widok zasobu dla powyższej tekstury
- m_faceTexture – tymczasowa tekstura do której renderowane będą pojedyncze ściany
- m_renderTarget – widok tekstury ściany, potrzebny do ustawienia tekstury jako wyjściowego bufora kolorów dla potoku renderowania
- m_depthBuffer – widok tekstury głębokości, potrzebny do ustawienia tej tekstury jako bufora głębokości dla potoku renderowania.

Aby poprawnie zainicjalizować teksturę m_faceTexture, należy zmodyfikować obiekt typu Texture2DDescription, na podstawie którego jest ona tworzona, tak aby wymiary odpowiadały rozmiarowi pojedynczej ściany tekstury sześciennej. Dodatkowo należy ustawić pole BindFlags na wartość D3D11_BIND_RENDER_TARGET, aby można było na jej podstawie stworzyć odpowiedni widok, natomiast pole MipLevels ustawić na wartość 1.

Przy tworzeniu tekstury sześciennej m_envTexture, obiekt Texture2DDescription musi określać poprawne wymiary pojedynczej ściany (pola Width i Height), pole BindFlags powinno zawierać wartość D3D11_BIND_SHADER_RESOURCE, pole MipLevels wartość 1, pole MiscFlags wartość D3D11_RESOURCE_MISC_TEXTURECUBE, natomiast pole ArraySize wartość 6.

2. Renderowanie do mapy sześciennej

W metodzie Render klasy RoomDemo należy przygotować rysowanie ścian tekstury środowiska. Wszystkie 6 widoków, z których będziemy scenę renderować będą korzystać z tej samej macierzy rzutowania (którą należy policzyć w metodzie FaceProjMtx klasy EnvironmentMapper, tak by kąt widzenia w pionie wynosił 90°, obraz był kwadratowy, a bliższa i dalsza płaszczyzna obcinania wynosiły odpowiednio m_nearPlane i m_farPlane), dlatego ustawić ją można raz. Każda ściana rysowana będzie do tej samej tekstury tymczasowej. W tym celu w metodzie SetTarget klasy EnvironmentMapper należy przypisać do potoku renderowania :

- Widok celu renderowania tekstury tymczasowej
- Odpowiadający mu bufor głębokości
- Zmianę współrzędnych okna widoku (D3D11_VIEWPORT) aby lewy górny róg znajdował się w punkcie (0; 0), okno miało wymiary równe wymiarom pojedynczej ściany tekstury sześciennej, a bliższa i dalsza głębokość wynosiły odpowiednio 0 i 1.

Przed wyrenderowaniem sceny do każdej ze ścian tekstury w metodzie Render klasy RoomDemo należy:

- Wyczyścić bufor koloru i głębokości tekstury tymczasowej wywołując dla obiektu m_envMapper metodę ClearTarget
- Zmianić macierzy widoku (przekształcenie z układu świata, do układu kamery) tak, jakby kamera znajdowała się w środku czajnika i skierowana była na renderowaną właśnie ścianę (przydatna może być funkcja XMMatrixLookToLH). Macierz ta powinna być policzona w metodzie FaceViewMtx klasy EnvironmentMapper.
- Wyrenderować scenę wywołując DrawScene.

Po wyrenderowaniu ściany wywołana musi być funkcja „SaveFace”, w której należy przekopiować dane z tekstury tymczasowej do odpowiedniej ściany tekstury sześciennej. Służy do tego metoda CopySubresourceRegion w obiekcie context. Poza teksturą docelową i źródłową istotny jest drugi parametr, który określa w której ścianie mają być umieszczone przekopiowane dane (można użyć wartości parametru face). Pozostałe parametry można ustawić na 0.

3. Obliczanie współrzędnych tekstury.

Wprowadzenie

Współrzędnymi dla mapy sześciennej są wektory o trzech współrzędnych. Taki wektor wyznacza kierunek. Promień wypuszczony ze środka sześcianu w określonym kierunku przetnie jego powierzchnię w pewnym punkcie. Tak mniej więcej wygląda schemat adresowania przy użyciu mapy sześciennej.

Jeśli świat ma się odbijać w powierzchni obiektu, to aby wyznaczyć kolor w pewnym jego punkcie musimy promień wzroku odbić od powierzchni obiektu i śledzić jego drogę po odbiciu. W przypadku mapowania środowiska z użyciem mapy sześciennej, po prostu wykorzystujemy kierunek po odbiciu przy adresowaniu mapy sześciennej. Aby uzyskać poprawny wynik należy uważać, aby wszystkie operacje przeprowadzać w układzie sceny.

Wykonanie

Należy uzupełnić shader wierzchołków („main” w pliku envMapperVS.hlsl) tak, aby współrzędnymi tekstury były współrzędne wektora wzroku odbitego od powierzchni obiektu. Pomocna może być funkcja „reflect”. Przekazany do niej promień wzroku musi iść od kamery w kierunku pozycji danego wierzchołka, natomiast wektor normalny przed wywołaniem tej funkcji musi być przekształcony do układu sceny. Nie należy zapomnieć o normalizacji wektorów.

4. Wyświetlanie

W metodzie „DrawTeapot” klasy Room zamiast używać shaderów m_phongVS i m_phongPS, w należy wywołać metodę Begin obiektu m_envMapper. Dodatkowo można odkomentować umieszczony tam kod rysujący jednostkową sferę w miejsce czajnika. Może się ona przydać do testowania poprawności generowania mapy środowiska i obliczania odbić.