

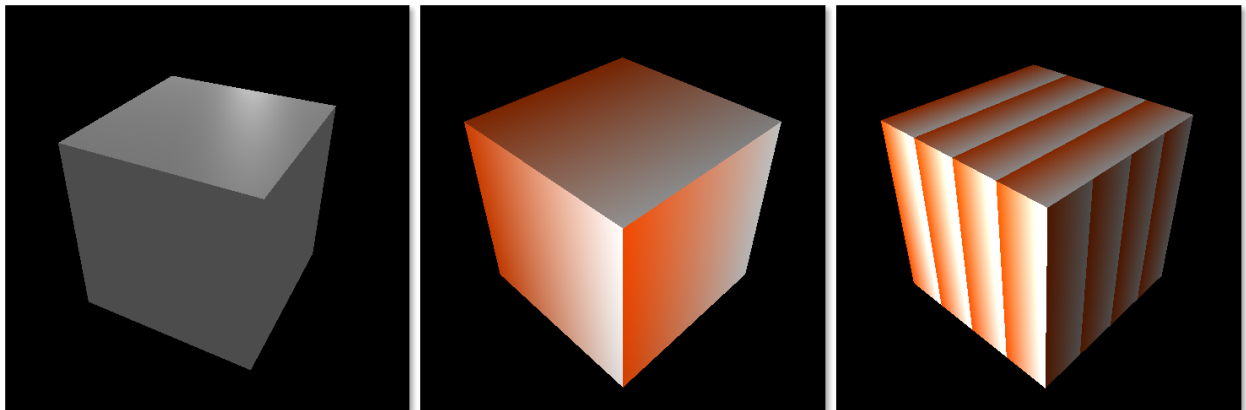
PV112 – osnova pátého cvičení v C++

1. část – procedurální textury

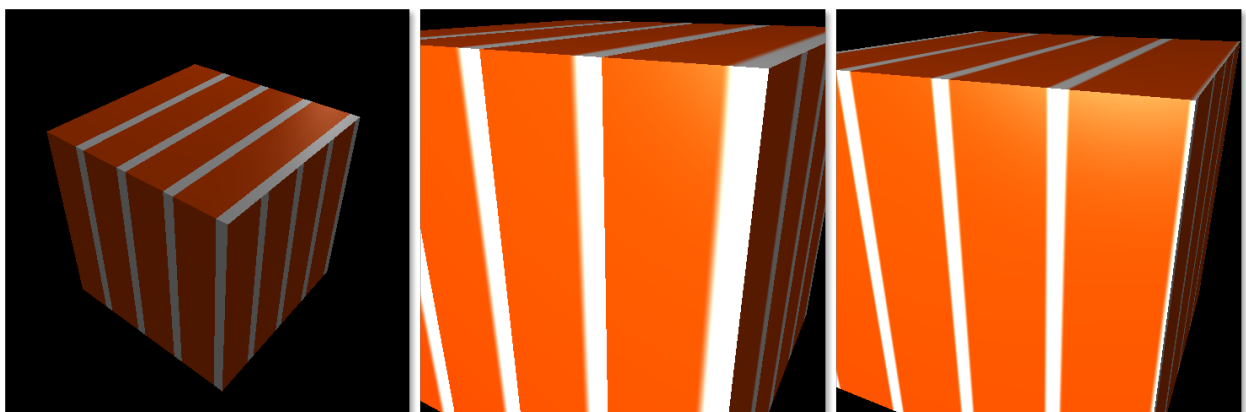
V tomto cvičení nebudeme otexturovat objekty pomocí obrázku; namísto toho barvu každého pixelu vypočítáme na grafické kartě na základě nějakých vstupních dat. Můžeme použít například polohu v rámci modelu (tímto způsobem se dají definovat 3D procedurální textury). My však budeme používat texturovací souřadnice.

Task 1 (pruhy)

Začínáme s bílou kostkou. Obarvíte ji nejdříve přechodem, který posléze zopakujete čtyřikrát. Funkci `mix` už znáte; hodit se může i `fract`, která vrací desetinnou část svého argumentu.



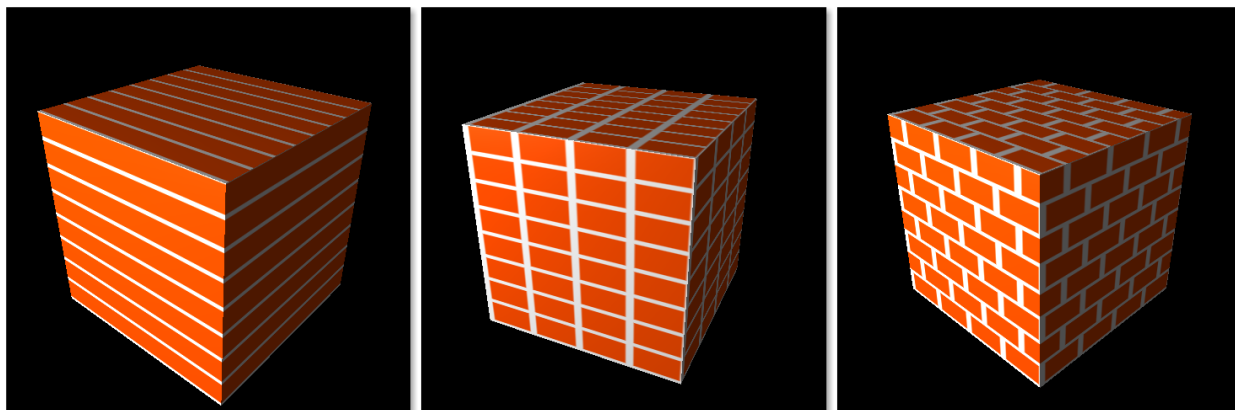
Ostrých přechodů je možné docílit s využitím funkce `step(t, x)`, která vrací 1.0 pro $x \geq t$, nulu jinak. Pozvolný přechod zajistí funkce `smoothstep(t, u, x)`, která vrací 0.0 pro $x < t$, 1.0 pro $x \geq u$ a pro x mezi těmito hodnotami „hladce“ interpolovanou hodnotu mezi 0.0 a 1.0.



Přechodem ze `step` na `smoothstep` se však „zjemní“ pouze jeden přechod, jak ukazuje prostřední obrázek. Vhodnou kombinací dvou `smoothstep`ů je však možné odstranit i druhý ostrý přechod.

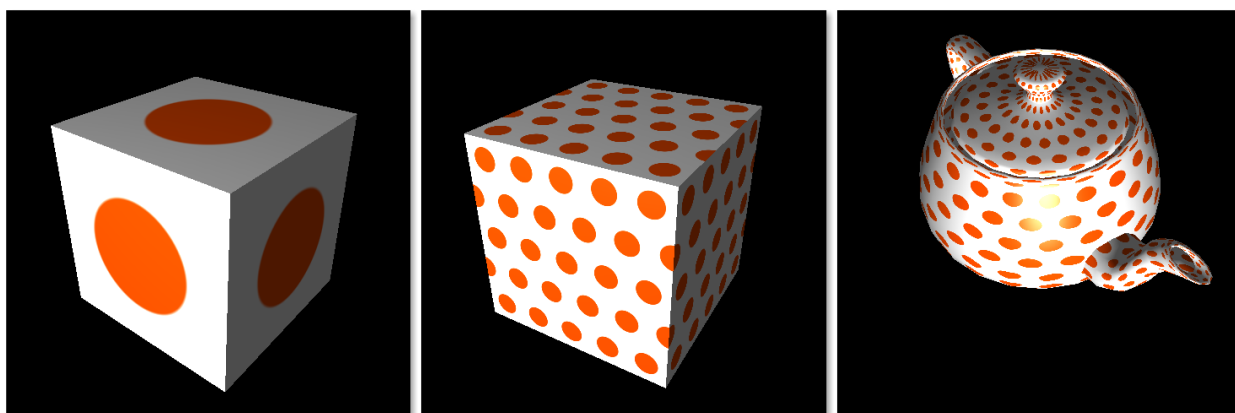
Task 2 a 3 (kachličky a cihly)

Vytvořte osm vodorovných pruhů. Zkombinujte je s vertikálními a vytvořte cosi na způsob kachliček. Tip: nemusíte kopírovat žádný kód, pokud ten stávající vhodně přepíšete na vektory. Nakonec každou druhou řadu posuňte tak, aby výsledek připomínal cihlovou zeď.



Task 4 (puntíky)

Začněte vykreslením jednoho kolečka uprostřed stěny (tip: použijte vzdálenost texturovacích souřadnic od středu). Následně využijte stejný princip jako při kreslení cihel a vytvořte „granko“ efekt jako na prostředním obrázku. A než začnete pracovat na další části, přejděte od krychle k vykreslování konvičky.



2. část – Postprocessing

Namísto vykreslování přímo do okna můžeme vykreslovat do textury. Touto texturou poté obarvíme čtverec, který teprve vykreslíme do okna. Při tomto druhém kreslení můžeme aplikovat efekty na celou výslednou scénu, jako například barevné změny nebo rozmazání.

Task 5 (framebuffer)

Nejprve je nutné vytvořit framebuffer object a k němu připojit texturové objekty pro barvu a hloubku. Textury je potřeba zvětšit na dostatečnou velikost (bude-li násobně větší než velikost okna, pak vlastně provádíme multisampling). Před kreslením scény oznámíme, že se má vykreslovat do tohoto framebufferu a výsledek bude uložen v připojených texturách. Potom stačí tuto texturu nanést na jednoduchý čtverec, který se už vykresluje do okna.

Task 6 (stupně šedi)

Vykreslete scénu černobíle. Neměňte ale shadery používané při kreslení konvičky, upravte postprocessingový shader.

Task 7 (vlnění)

Rozvlňte celou scénu. Do postprocessingového shaderu už si posíláme dobu běhu programu; využijte ji.

Task 8 (wireframe)

Stiskem klávesy L se má scéna vykreslit ve wireframe módu, ovšem nyní to nefunguje správně. Zkuste chybu najít a opravit.

