

Vaja 5– Senčenje z GLSL in teksture

1. Zahteve naloge

Z uporabo OpenGL 3/4 (Core profile) ali WebGL implementirajte senčenje prikazanih objektov. Za zadovoljiv prikaz naj bo prikazani objekt kompleksnejši od kocke (naj ima vsaj 10 oglišč). Implementirajte Phongovo senčenje in poenostavljen lokalni osvetlitveni model z vsaj enim svetlobnim virom. Omogočite tudi nastavljanje parametrov osvetlitve izbranega objekta (npr. barva, razpršenost, odbojnost) in svetlobnega vira (lokacija, intenziteta). Omogočite še vsaj premikanje in rotacijo kamere in objekta, tako da je možno preveriti pravilnost senčenja. Pri tem pazite na izračun osvetlitve ob transformacijah kamere in objektov (rotacija normal, pozicija luči).

Implementirajte še teksturiranje objektov, torej na objekte nalepite teksture. Pri preslikavi teksture na objekt implementirajte:

- privzeto,
- ravninsko,
- sferično in
- cilindrično lepljenje teksture na objekt.

Privzeto lepljenje teksture na objekt pomeni, da uporabite vnaprej izračunane UV koordinate (iz datoteke ali ročno določeno). Uporabniku med izvajanjem aplikacije omogočite izbiro med vsemi navedenimi načini lepljenja tekstur in osmi lepljenja (pri ravninski, sferični in cilindrični preslikavi).

Skupna vrednost naloge je 8 točk:

- Senčenje ter nastavljanje parametrov senčenja in osvetlitve za posamezne objekte in svetlobni vir: 2 točki.
- Transformacije posameznih objektov in pravilno delovanje osvetlitvenega modela pri transformacijah objektov: 1 točki.
- Prikaz tekstur na tleh in objektih: 2 točki (uporaba vsaj privzete preslikave).
- Preslikava teksture na objekt: 3 točke:
 - ravninsko: 1 točka,
 - sferično: 1 točka,
 - cilindrično: 1 točka.

Implementacijo lahko izvedete v programskem jeziku C++, C#, Java, ali JavaScript, pri čemer lahko uporabite ustrezne knjižnice (npr. Qt, OpenTK, JOGL) za izvedbo preprostega uporabniškega vmesnika in sprejemanje uporabnikovih akcij (navadno okno, zajemanje tipk). Sam izris mora biti izveden v samem OpenGL brez dodatnih knjižnic za sam izris.

V primeru implementacije v WebGL pazite na omejitve zaradi OpenGL ES in spletnega brskalnika (https://developer.mozilla.org/en-US/docs/Web/API/WebGL-API/Tutorial/Using_textures_in_WebGL):

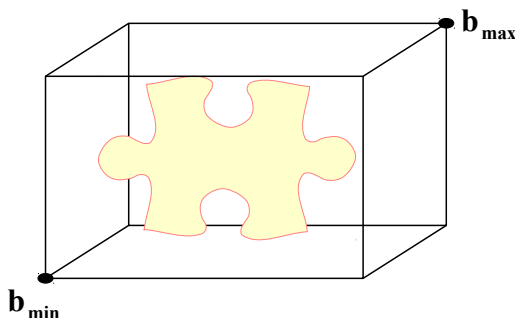
- Dimenzije tekstur morajo biti potence števila 2.
- Zaradi večje varnosti, nalaganje tekstur in objektov iz drugih spletnih strani ni dovoljeno (<https://github.com/mrdoob/three.js/wiki/How-to-run-things-locally>). V tem primeru je najlažje vzpostaviti lokalni spletni strežnik.

Dodatna pomoč

Ravninska, cilindrična in sferična preslikava tekstur

Vsaki točki oziroma oglišču (\mathbf{p}) objekta, na katerega želimo nalepiti teksturo, določimo koordinati piksla texture x in y po enačbah opisanih v nadaljevanju. Za koordinati x in y na teksturi ponavadi raje uporabimo oznaki s in t ali tudi u in v , od tod beseda UV mapping.

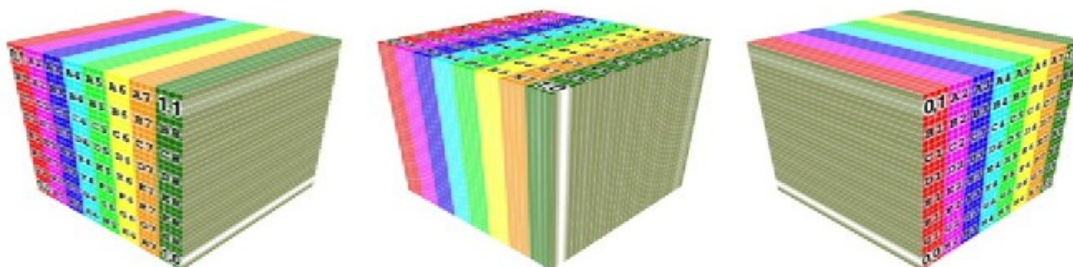
Ne glede na to, kateri način preslikave smo izbrali, je potrebno najprej določiti omejujoči kvader objekta, oziroma površine, na katero bomo nalepili teksturo. Tako bomo preslikali koordinate objekta v območje med 0 in 1. Omejujoči kvader je določen s točkama \mathbf{b}_{\min} in \mathbf{b}_{\max} , glej sliko 1. Torej pri \mathbf{b}_{\min} komponente x , y , z predstavljajo najmanjše vrednosti vseh oglišč objekta, za \mathbf{b}_{\max} velja analogno. Nato je potrebno izračunati vektor \mathbf{d} kot razdaljo med \mathbf{b}_{\min} in \mathbf{b}_{\max} : $\mathbf{d} = \mathbf{b}_{\max} - \mathbf{b}_{\min}$.



Slika 1: Omejujoči kvader objekta

Ravninsko lepljenje texture

V tem primeru je tekstura projicirana kot ravnina na površino objekta, glej sliko 2.



Slika 2:

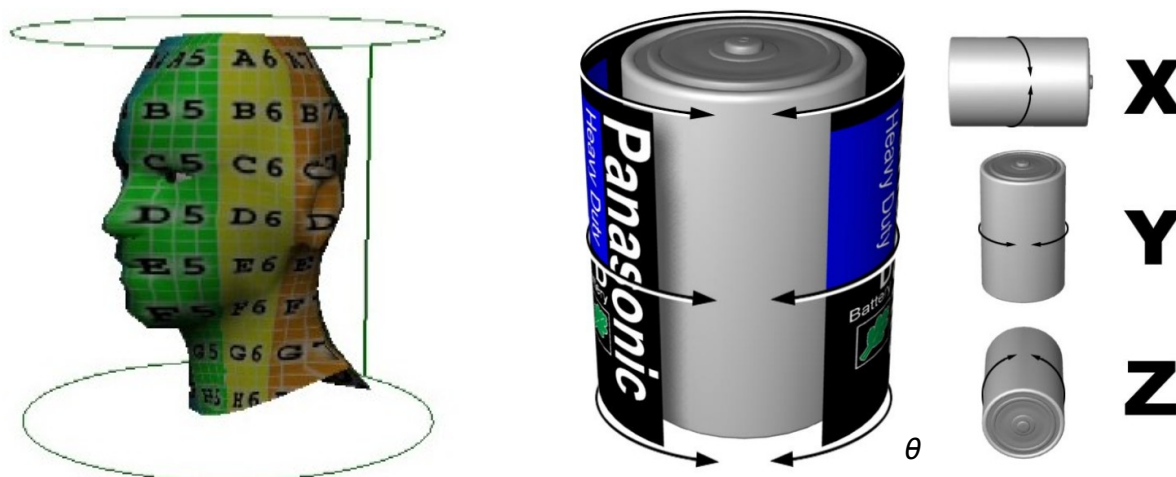
Ravninsko lepljenje texture po oseh X , Y in Z

Vsaki točki \mathbf{p} objekta ali površine, na katero želimo nalepiti teksturo, priredimo koordinati texture s in t po naslednjih enačbah:

Lepljenje v smeri X	Lepljenje v smeri Y	Lepljenje v smeri Z
$s = \frac{p_z - b_{\min,z}}{d_z},$ $t = \frac{p_y - b_{\min,y}}{d_y}$	$s = \frac{p_x - b_{\min,x}}{d_x},$ $t = \frac{p_z - b_{\min,z}}{d_z}$	$s = \frac{p_x - b_{\min,x}}{d_x},$ $t = \frac{p_y - b_{\min,y}}{d_y}$

Cilindrično lepljenje teksture

Pri cilindričnem lepljenju teksture le-to ovijemo okrog objekta, glej sliko 3.



Slika 3: Cilindrično lepljenje teksture

Da bo cilindrično lepljenje delovalo, je potrebno teksturo na nek način ukriviti in jo oviti v eni od izbranih smeri okrog objekta. Podobno kot pri ravninskem lepljenju teksture, je potrebno tudi tukaj najprej določiti obkrožujoči kvader objekta ali površine in to ne glede na smer lepljenja. Tudi v tem primeru je kvader določen s točkama \mathbf{b}_{\min} in \mathbf{b}_{\max} . Nadalje je potrebno določiti center objekta, \mathbf{c} , oziroma površine, ki ga dobimo z naslednjo enačbo:

$$\mathbf{c} = \mathbf{b}_{\min} + \mathbf{d}/2.$$

Potem za vsako točko objekta \mathbf{p} izračunamo \mathbf{v} z naslednjo enačbo:

$$\mathbf{v} = \mathbf{p} - \mathbf{c}.$$

Cilindrično lepljenje teksture je enako projekciji točke na valj. Določanje koordinate s je pri vseh treh smereh lepljenja enako in sicer:

$$s = \frac{\theta}{360}.$$

Kot θ iz parametričnih enačb za valj izračunamo s pomočjo funkcije atan2 (v GLSL $\text{atan}(x,y)$), kot je prikazano na naslednjih enačbah:

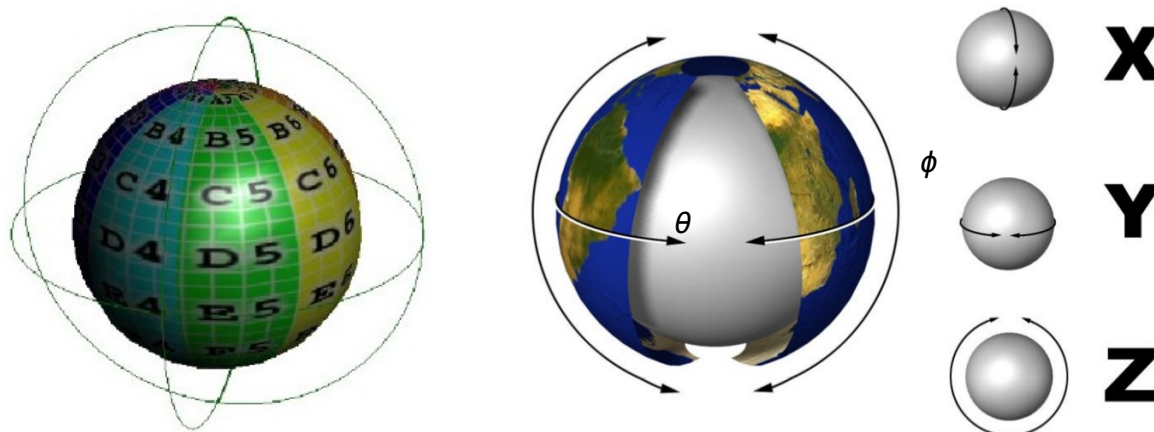
Lepljenje v smeri X	Lepljenje v smeri Y	Lepljenje v smeri Z
$\theta = \text{atan2}(v.z, v.y) + 180^\circ$	$\theta = \text{atan2}(v.x, v.z) + 180^\circ$	$\theta = \text{atan2}(v.x, v.y) + 180^\circ$

Koordinato teksture t pri lepljenju v smereh X, Y in Z določimo kot:

Lepljenje v smeri X	Lepljenje v smeri Y	Lepljenje v smeri Z
$t = \frac{v.x}{d.x} + \frac{1}{2}$	$t = \frac{v.y}{d.y} + \frac{1}{2}$	$t = \frac{v.z}{d.z} + \frac{1}{2}$

Sferično lepljenje teksture

Pri sferičnem lepljenju teksture le-to ovijemo okrog objekta v dveh smereh, glej sliko 4.



Slika 4: Sferično lepljenje teksture

Pri sferičnem lepljenju teksture gre za to, da teksturo dvakrat ukrivimo. Lepljenja se lotimo podobno kot pri cilindričnem lepljenju, to je z določitvijo vektorjev \mathbf{b}_{\min} , \mathbf{b}_{\max} , \mathbf{d} , \mathbf{c} in \mathbf{v} . Prav tako pa lahko tudi tukaj koordinate točke zapišemo s pomočjo parametrične enačbe, le da je v tem primeru točka določena z dvema kotoma (θ in ϕ). Koordinati s in t točki \mathbf{p} priredimo z naslednjima enačbama:

$$s = \frac{\theta}{360^\circ}, \quad t = \frac{\phi}{180^\circ} + \frac{1}{2}.$$

Kota θ in ϕ izračunamo glede na koordinate točke \mathbf{p} z naslednjimi enačbami:

Lepljenje v smeri X	Lepljenje v smeri Y	Lepljenje v smeri Z
$\theta = \text{atan2}(v.z, v.y) + 180^\circ$, $\phi = \text{asin}\left(\frac{v.x}{ v }\right)$	$\theta = \text{atan2}(v.x, v.z) + 180^\circ$, $\phi = \text{asin}\left(\frac{v.y}{ v }\right)$	$\theta = \text{atan2}(v.x, v.y) + 180^\circ$, $\phi = \text{asin}\left(\frac{v.z}{ v }\right)$

Implementacija

Teksture nalepimo na naložen objekt tako da:

1. Teksturo najprej naložimo v pomnilnik na grafični procesni enoti.
2. Izračunamo obkrožujoči kvader objekta (maksimalne in minimalne koordinate po vseh oseh (X,Y,Z) , torej \mathbf{b}_{\min} in \mathbf{b}_{\max}).
3. Senčilnikom nastavimo, katero teksturo bomo uporabili, kateri način preslikave tekstur in po kateri osi.
4. Izračunamo koordinate tekstur (s,t) glede na izbran način preslikave tekstur.
5. V senčilniku fragmentov uporabimo barvo iz texture na koordinatah (s, t) .

Koordinate za texture lahko izračunamo v senčilniku oglišč ali fragmentov, lahko pa tudi ob nalaganju objekta. Če koordinati s in t izračunamo v senčilniku oglišč ali fragmentov, moramo s pomočjo uniformnih spremenljivk senčilnika določiti tip in os lepljenja tekstur. Preslikavo lahko namreč izvedemo ravninsko, cilindrično, sferično glede na osi X, Y in Z . Lahko pa uporabimo vnaprej pripravljene koordinate UV . Torej imamo 10 možnih kombinacij.

Dodatna literatura in pomoč:

- <http://www.opengl-tutorial.org/>
- <https://www.opengl.org/wiki/Tutorials>
- <https://open.gl/>
- <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-8-basic-shading/> - primer senčenja
- <http://assimp.sourceforge.net/> - uvažanje modelov iz različnih animacijskih paketov
- <http://freeimage.sourceforge.net/> - nalaganje slik za texture z diska, lahko pa uporabite Qt (QImage - <http://doc.qt.io/qt-5/qimage.html>)