



Z-buffer till textur

Hur vore det att komma åt Z-buffern från en rendering?

Normalt bara hjälpbuffer, slå på och glöm bort.

Men... måste det vara så?

Kan den vara bra att ha?



Kopiering eller FBO

Kan göras med kopiering mellan buffrar.

Nya, snabbare metoden: FBO!

Skapa FBO med enbart Z-buffer!

Lite annorlunda FBO; stäng av de vanliga delarna:

```
glDrawBuffer(GL_NONE);  
glReadBuffer(GL_NONE);
```



Exempel: Tekannans Z-buffer renderad på tekannan





Rita Z direkt?

Vi kan också rita normaliserade djupvärden direkt i utdata (FBO eller framebuffer).

Måste göras med Z-buffern påslagen för att få korrekt VSD.

Vi skapar samma data två gånger!

Bättre att ha en minimal fragment shader.

Bakväg om du har problem med att få ut Z-buffern.



Så nu har vi Z-buffern som indata.

Vad skall vi ha den till?

- Förenklad kollisionsdetektering
- Kollisionsundvikande (speciellt i rörelseriktningen).
 - Voxelisering av konvexa eller nästan konvexa kroppar.
 - SSAO (fö 3)
 - Skuggmappning (fö 3)

Vi återkommer till detta... flera gånger!



Impostoring

Tillämpning av rendering till textur.

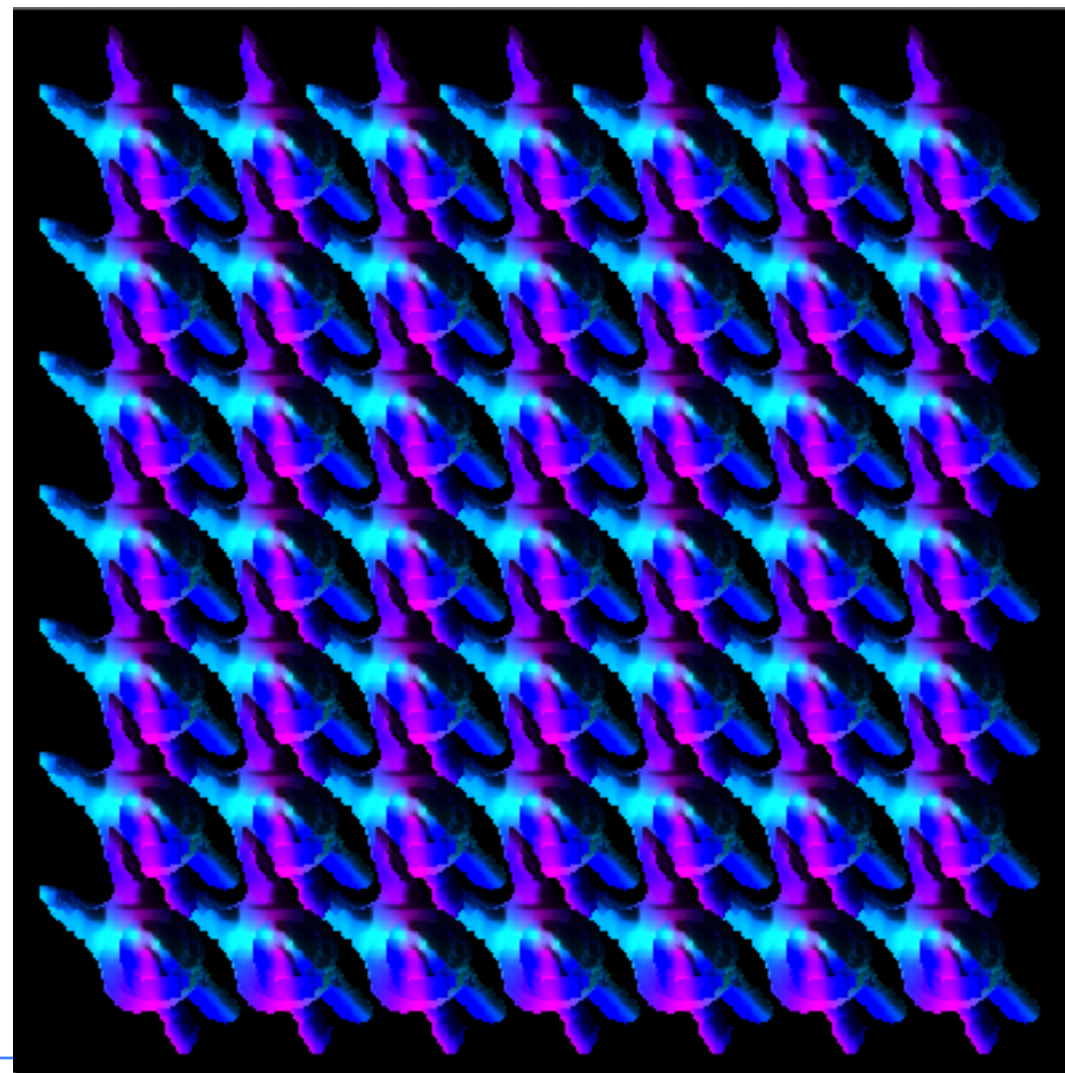
Billboards med bild som renderas online.

- Optimering när en modell skall ritas många gånger i samma frame.
- Optimering när en komplex modell skall ritas många gånger i ungefär samma pose.



Samma flera gånger

"Simple Impostor"-demo, 49 kaniner, men kaninen ritas bara en gång.





Töm, bind texturen

```
// clear the screen  
glViewport(0, 0, width, height); // Fixed size  
glClearColor(0,0,0,0); // Alpha = 0 is important here  
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
glBindTexture(GL_TEXTURE_2D, minitexid);
```



Rendera kaninen men utan glutSwapBuffers()

// Render the bunny to screen (e.g. back-buffer) without displaying it, no
glutSwapBuffers().

```
glUseProgram(bunnyshader);  
modelView = worldToView * Rx(a) * Rz(a/5.0) * S(2.0, 2.0, 2.0);  
glUniformMatrix4fv(glGetUniformLocation(bunnyshader, "modelView"), 1,  
GL_TRUE, modelView.m);  
DrawModel(bunny, bunnyshader, "inPosition", "inNormal", "inTexCoord");  
  
glFlush();
```



Kopiera bilden till textur

```
// Copy the result to the texture "tex"  
glBindTexture(GL_TEXTURE_2D, tex);  
glCopyTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, 0, 0, width, height);
```



Rita texturen 49 gånger med quad

```
glViewport(0, 0, lastw, lasth);
glClearColor(0,0,0,0);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glUseProgram(texshader);
for (GLfloat x = -3; x < 4; x++)
for (GLfloat y = -3; y < 4; y++)
{
    modelView = worldToView * T(x/2.0, y/2.0, 0.0);
    glUniformMatrix4fv(glGetUniformLocation(texshader, "modelView"),
1, GL_TRUE, modelView.m);
    DrawModel(quad, texshader, "inPosition", NULL, "inTexCoord");
}
```



Ännu bättre:

- Rendera till FBO för att slippa kopiera
- Instancing för att rita med färre anrop