



Computer Graphics

7,5 högskolepoäng

Provmoment: Tentamen
Ladokkod: NDG011
Tentamen ges för: Systemarkitektprogrammet

Namn: _____

(Ifylles av student)

Personnummer: _____

(Ifylles av student)

Tentamensdatum: 2016-05-30
Tid: 09:00 – 14:00

Hjälpmedel:
Inga hjälpmedel

Totalt antal poäng på tentamen: 36 poäng
För att få respektive betyg krävs:
G=18
VG=27

Allmänna anvisningar:

Skriv dina svar på separata blad

Spara alla filer på F: om inte annat specifikt anges

Se till att lösningarna inte har bindningar till filer på C: i andra mappar än specifikt angivna

Stäng ner samtliga applikationer och logga ut (via ikon) när du är klar med tentamen.

Vid olösliga problem med datorkonto, ta ett nytt och lämna in även detta i tentamensmappen.

Rättningstiden är som längst tre veckor

Viktigt! Glöm inte att skriva namn på alla blad du lämnar in.

Lycka till!

Ansvarig lärare: Rikard König
Telefonnummer: 033 435 5945

- 1) Describe the difference between the additive and subtractive colour systems and why both only require only three primary colour to function. (2p)
- 2) Define a matrix that transforms a point in the homogeneous coordinate system by scaling it by 30% along the Y-axis and moves it 4 units in the z-direction and 5 units along the X-axis. Finally explain how a point in homogenous coordinates is transformed in to the Cartesian coordinate system. You can decide yourself if you want to do a matrix for row vectors or column vectors but clearly state witch you choose (4p)
- 3) In a rendering engine translucent and semi-translucent objects are handled in a certain way in order to avoid artefacts and strange rendering behaviour.
 - a) Explain how translucent objects are handled in a rendering engine, and how the translucency effect is achieved. (1.5p)
 - b) Explain how objects with both fully opaque and fully translucent parts (semi-translucent) are handled in a rendering engine, and how the semi-translucent effect is achieved. (1.5p)
- 4) Pick one of the following shading techniques and explain in detail how the effect is implemented, in the shader as well as the engine;

Normal Mapping

Environment Mapping

Shadow Mapping

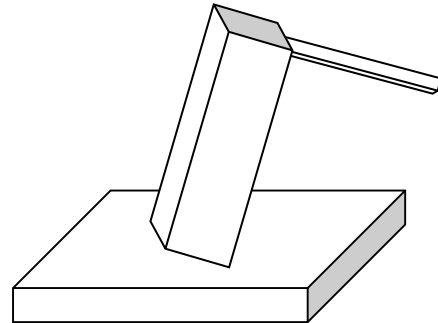
Parallax Mapping

Your explanation should at least include; what is required from the engine side to make the effect work, what the shader need as input in order to achieve the effect, what the main calculation steps are in the shader etc.(3p)

- 5) Hidden surface removal can be performed in either object-space or image-space. Explain the differences. (1p)
- 6) Does the z-buffer algorithm operate in object-space or image-space? Explain why. (2p)
- 7) Consider the task of creating a complex animated model that consists of several moving parts (e.g. a car or a humanoid figure).
 - a) Explain how this model can be constructed non-hierarchically. Describe the limitations of constructing the model in this manner (in terms of animating and rendering the model). Provide an illustrated example. (1.5p)
 - b) Explain how this model can be constructed hierarchically. Describe the benefits of constructing the model in this manner (in terms of animating and rendering the model). Provide an illustrated example. (1.5p)

- 8) Create a program using MonoGame that displays a robot arm that can be controlled (moved and rotated) using the keyboard. The robot arm should consist of a flat square base, a thick lower arm and a thin upper arm. The model should be constructed hierarchically. The camera should follow the robot arm (behind and above it). Use the following key map for your program:

A: turn left
D: turn right
W: move forward
S: move backward
J: tilt lower arm left
L: tilt lower arm right
I: tilt upper arm forward
K: tilt upper arm backward



To prevent the robot arm from falling helplessly into the abyss, you also need to create ground for it to move on. The ground should be created by tiling together a given texture (“stone.jpg”), where each ground tile is drawn using the same (buffered) mesh (i.e., to draw the ground, you should make one draw-call for each tile, using the same vertex buffer but different world transformation matrices). The size of the tiles should be the same as that of the robot arm’s base. You can limit the size of the ground to 50 * 50 tiles (with the robot arm starting near the center). You should use bounding volumes (e.g. BoundingSpheres and BoundingFrustum) to avoid rendering ground tiles that are outside of the camera's view.

The necessary binary contents are provided in a directory on your account. (18p)