# EXAM IN

# COMPUTER GRAPHICS

# TSBK07

# (TEN1)

| | |
|---|---|
| Time: | 5th of June, 2014, 8-12 |
| Room: | R41, R42, R34, R44 |
| Teacher: | Ingemar Ragnemalm,<br>visits around 9 and 11 |
| Allowed help: | None |
| Requirement to pass: | Grade 3: 21 points<br>Grade 4: 31 points<br>Grade 5: 41 points<br><br>ECTS:<br>C: 21 points<br>B: 31 points<br>A: 41 points |

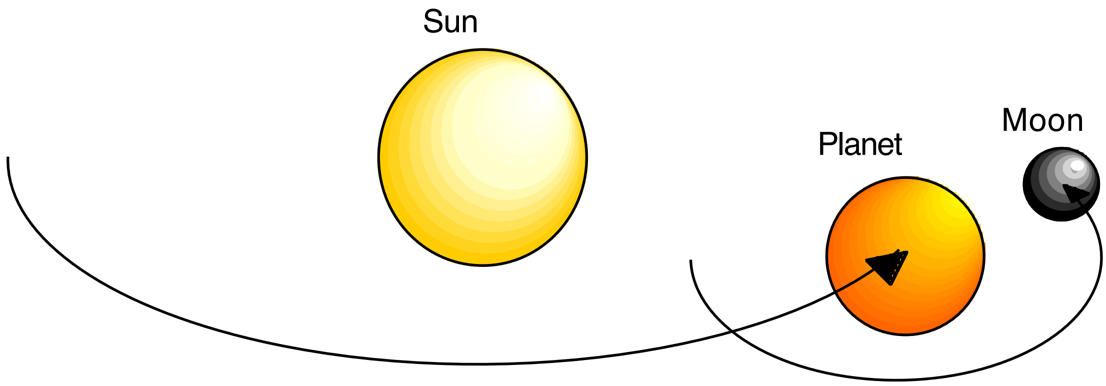
Answers may be given in swedish or english.

Please make a special note if you followed the course before 2012. Some answers may be slightly different depending on that and I need to know what material you studied (old or new) to make fair scoring.

**- Wish us luck!**
**- I wish you skill!**
[Martin Landau, ”Mission Impossible”]

## 1. OpenGL programming

a) The figure shows a small planetary system consisting of one sun, one planet and one moon. The planet rotates around the sun, and the moon around the planet, all in the XZ plane.




The sun is drawn at $x_S$, $y_S$, $z_S$, the planet rotates around the sun at radius $r_P$, with the angle a, and also rotates around its own Y axis by the angle b. Similarily, the moon rotates around the planet at radius $r_m$, angle c, and rotates around its Y axis by angle d.

Function calls drawSun() , drawPlanet() and drawMoon() are provided, all drawing the object centered around origin plus whatever transformation you provide. Write OpenGL (or OpenGL-like) code to draw the planetary system. Shaders do not have to be given but the order of multiplications for the matrices is vital.

(3p)

b) Graphics shaders have a built-in interpolation feature. Describe how it is programmed by a simple code example. (Shader code only. It does not have to be complete code.)

(2p)

## 2. Transformations

a) Give a sequence of 3x3 matrixes that define a 2D transformation that rotates around a point **p** by an angle $\phi$. You don't have to multiply the matrices together. Content of matrices must be given.

(3p)

b) Which one(s) of the following statements are true (given names and symbols used in the course)?

(1) Translation: $T(\mathbf{p})^{-1} = T(\mathbf{p})^T$
(2) Rotation: $R(\phi)^{-1} = R(\phi)^T$
(3) Mirroring: $M = R(\pi)$
(4) Composite transformations: $T(\mathbf{p})*S(a) = S(a)*T(\mathbf{p})$
(5) Composite transformations: $R(f)*S(a) = S(a)*R(f)$
(6) Cross product: $\mathbf{a}\times\mathbf{b} = \mathbf{b}\times\mathbf{a}$

(2p)

c) How do you calculate a normal matrix from a model-to-view matrix?

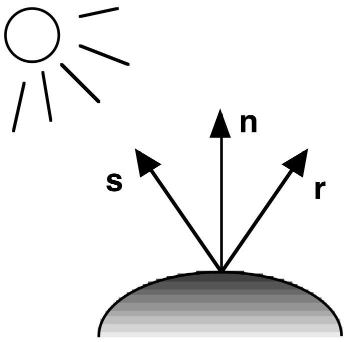
(2p)

## 3. Light, shading and ray-tracing

a) On a written exam, a student gets the task of writing a complete formula for the 3-component light model. The student answers with the following formula:

$$i = k_d i_a + k_d \cdot i_s \cdot \cos(\mathbf{s} \bullet \mathbf{n}) + k_{spec} i_s \cdot \cos(\mathbf{r} \bullet \mathbf{n})$$

together with the figure




The examiner is not happy with the answer. Suggest improvements that will give the student full score.

(3p)

b) Suggest two significantly different graphical effects suitable for ray-tracing that can be solved with jittering.

(2p)

c) How are shadows produced in ray-tracing?

(1p)

## 4. Surface detail

a) A skybox is a powerful yet simple tool for creating an immersive environment. However, it is not quite as simple as it seems at first glance. Describe the method in sufficient detail, including the minor complications that must be taken into account to make it work properly.

(4p)

b) Describe, with formulas, how to map a texture to an object vertex by vertex, using a cylinder as intermediate surface. (You may assume that you have access to both the 1-argument atan and 2-argument atan2.)

(3p)

c) After mapping point-to-point in b), one problem remains. You can not simply map every vertex to texture space and assume a correct result. Why, what will happen? Suggest a remedy.

(2p)

## 5. Curve generation

a) A quadratic Bezier curve is defined by the following formula.

$$\mathbf{p}(u) = (1 - u)^2\mathbf{p_0} + 2(1-u)u\mathbf{p_1} + u^2\mathbf{p_2},\ u \in [0, 1]$$

If we use this on a sequence of control points $\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, \mathbf{p_4}$, it will be applied on three at a time, and these sections will be fit together. On what conditions does it fulfill $C^1$ or $G^1$ continuity in the intersections of two sections?

Prove mathematically that this is the case. State clearly what the condition means in mathematical terms, that is the relationship that you should prove.

(5p)

c) Show, by using a figure and an appropriate table, how the Bresenham line-drawing algorithm works.

Hint: $p_0 = 2\Delta y - \Delta x$
$p_{k+1} = p_k + 2\Delta y$
$p_{k+1} = p_k + 2\Delta y - 2\Delta x$

(3p)

## 6. Miscellaneous

a) Given a triangle as three points **a**, **b**, **c** and a point **p** in the same plane as the triangle, describe how you can determine if **p** in inside the triangle.

(2p)

b) Describe, using a figure, how the Diamond-Square noise/terrain generation algorithm works.

(4p)

## 7. Collision detection and animation

a) Using figures, give two depth cues that can be used to give the impression of depth in 2D animations.

(2p)

b) Describe how 2D collision detection can be performed using the Separating Axis Theorem (SAT). (It is recommended to use figures.)

(3p)

## 8. Visible surface detection and large worlds

a) One of the most common VSD methods have problems with transparency. Describe how this can be handled.

(2p)

b) Describe how view plane oriented billboards can be implemented.

(2p)