Graphics (INFOGR 2011-2012): Final Exam (T2)

Tuesday, July 3, 2012, EDUC-GAMMA, 09:00-12:00 (time for the exam: 2 hours)

StudentID / studentnummer	Last name / achternaam	First name / voornaam

Do not open the exam until instructed to do so!

Read the instructions on this page carefully!

- You may write your answers in English or Dutch. Use a pen, not a pencil. Do not use red or green.
- Fill in your name and student id at the top of this page, and write it on every additional paper you want to turn in.
- Answer the questions in the designated areas on these exam sheets. If you need more space, make a cross in the designated circle at the end of the problem and continue writing on the additional paper provided by us. You are not allowed to use your own paper. On the additional paper, make sure to clearly indicate the problem number and don't forget to write your name and student ID on it.
- You may **not** use books, notes, or any electronic equipment (including your cellphone, even if you just want to use it as a clock).
- You have max. 2 hours to work on the questions. If you finish early, you may hand in your work and leave, except for the first half hour of the exam. When you hand in your work, have your **student ID** ready for inspection.
- The exam has 8 problems printed on 15 pages (including this one). It is your responsibility to check if you have a complete printout. If you have the impression that anything is missing, let us know.

Good luck / veel succes!

Do not write below this line

Probl. 1 (max. 6 pts)	
Probl. 2 (max. 2 pts)	
Probl. 3.1 (max. 6 pts)	
Probl. 3.2 (max. 4 pts)	
Probl. 3.3 (max. 6 pts)	
Probl. 4.1 (max. 6 pts)	
Probl. 4.2 (max. 3 pts)	
Probl. 5.1 (max. 2 pts)	
Probl. 5.2 (max. 8 pts)	

Probl. 5.3 (max. 3 pts)	
Probl. 6 (max. 8 pts)	
Probl. 7.1 (max. 8 pts)	
Probl. 7.2 (max. 5 pts)	
Probl. 8.1 (max. 10 pts)	
Probl. 8.2 (max. 3 pts)	
Probl. 8.3 (max. 6 pts)	
Probl. 8.4 (max. 6 pts)	
Probl. 8.5 (max. 8 pts)	

Points: ____________ Grade: ____________

Problem 1: Texture mapping

■ **[6 pts]: Texturing (multiple choice questions).** Mark the correct answer. No explanation required. For each of the questions 1.-3., there is only one correct answer.

1. The following procedure

   stripe (x_p, y_p, z_p, w) {
       if $(\sin(\pi x_p / w) > 0)$ return *color1*;
       else return *color2*;
   }

   can be used to create a 3D texture of stripes along the $x-$axis with ...

   A. ... width π.
   B. ... width w.
   C. ... width $\frac{\pi}{w}$.
   D. ... width $\frac{w}{\pi}$.
   E. ... width $\pi \cdot w$.
   F. ... none of the above.

   ___

2. Hermite interpolation uses ...

   A. ... no weights.
   B. ... linear weights, i.e. weights to the power of 1.
   C. ... quadratic weights, i.e. weights to the power of 2.
   D. ... cubic weights, i.e. weights to the power of 3.
   E. ... none of the above.

   ___

3. Bump mapping modifies ...

   A. ... the shape and normal vectors of an object.
   B. ... the shape of an object but not its normal vectors.
   C. ... the normal vectors of an object but not its shape.
   D. ... neither the shape nor the normal vectors of an object.

   ___

Problem 2: Perspective projection

■ **[2 pts]: Projection matrices (multiple choice question).** In the process of perspective projection, we use the following matrix M (among others):

$$M = \begin{pmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x}{2} - \frac{1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y}{2} - \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Which of the following operations is done by M?
Mark the correct option. No explanation required. There is only one correct answer.

A. M is the matrix that maps camera coordinates to world coordinates.

B. M is the matrix that maps world coordinates to camera coordinates.

C. M is the matrix that maps the view frustum to the orthographic view.

D. M is the matrix that maps the orthographic view volume to the canonical view volume.

E. M is the matrix that maps the canonical view volume to the screen window.

F. Neither of the above operations is done by M.

Problem 3: Clipping

■ **Subproblem 3.1 [6 pts]: Clipping triangles.** Assume a plane f defined by

$$f(\vec{p}) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \cdot \vec{p} + 2 = 0$$

and a triangle defined by the three vertices $\vec{a} = (1,2,2)$, $\vec{b} = (2,1,-2)$, and $\vec{c} = (-2,-1,2)$.
Which of the three triangle edges $\overline{ab}$, $\overline{bc}$, and $\overline{ac}$ intersect with the plane? Provide a prove for your answer!

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 3.2 [4 pts]: Clipping in the graphics pipeline.** Where is the best position to do clipping in the graphics pipeline? Mark the right answer and shortly explain why this is correct. (Note: 1-2 short sentences can be sufficient to get full credits for this subproblem.)

(I) After the homogeneous divide and before rasterization.

(II) After perspective transformation and before the homogeneous divide.

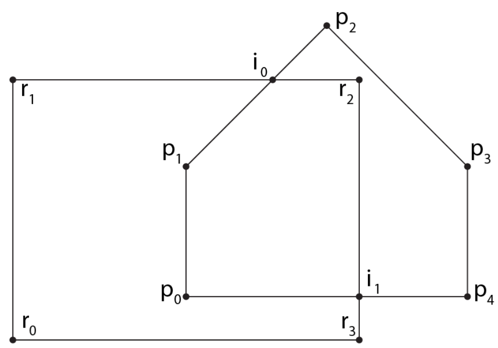
(III) After specification of the view frustum and before the perspective transformation.

(IV) It should be done at none of the above positions but somewhere else.

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 3.3 [6 pts]: Clipping arbitrary polygones.** Below you see a polygone defined by its five vertices $p_0, \dots p_4$. It intersects with a clipping region defined by its four vertices $r_0, \dots r_3$. The intersection points are denoted by i_0 and i_1.



Draw the related graph that is used by the Weiler-Atherton algorithm for clipping arbitrary polygones. Start with vertices p_0 and r_0 and proceed in clockwise order (i.e. start with p_0 and r_0, then p_1 and r_1, and so on). Don't forget to clearly indicate which nodes are outgoing and incoming intersections.

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

Problem 4: Hidden surface removal

■ **Subproblem 4.1 [6 pts]: Culling.** Complete the following sentences in a way that creates a correct statement that clearly specifies the characteristic of the triangles that are removed by that particular operation.

1. Frustum (or volume) culling removes triangles ...

   ___

2. Occlusion culling removes triangles ...

   ___

3. Backface culling removes triangles ...

   ___

■ **Subproblem 4.2 [3 pts]: z-Buffer (multiple choice question).** Values in a z-buffer are often stored as non-negative integers. Because we only have a fixed number of values available for those, mapping our original z-values to them can lead to precision problems. Which of the following statements is correct? (Mark the correct one. No explanation required. There is only one correct answer.)

The precision of the $z-$buffer is always increased ...

A. ... when we move the near plane n and the far plane f further away from the camera.

B. ... when we move the near plane n closer to the camera and the far plane f further away from it.

C. ... when we move the near plane n further away from the camera and the far plane f closer to it.

D. ... when we move the near plane n and the far plane f closer to the camera.

E. ... by none of the above actions.

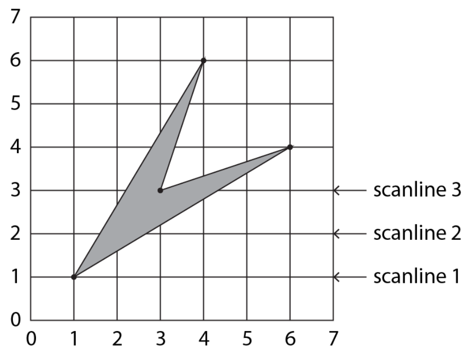
Problem 5: Rasterization

■ **Subproblem 5.1 [2 pts]: Bounding boxes.** Assume a polygone in 2D that is defined by n vertices (x_i, y_i) with $i = 1, \dots n$. The polygone's bounding box is defined as the smallest axis-parallel box that contains the whole polygone. We can specify this box by two points $\vec{b}_0$ and $\vec{b}_1$.
Write down a mathematically correct specification of $\vec{b}_0$ and $\vec{b}_1$.

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 5.2 [8 pts]: Scanline algorithm for polygone rasterization.** In the following, we want to apply the scanline algorithm for rasterization of random polygones in 2D to this example:



- Write down the complete edge table (ET) that is used by this algorithm.

Answer:

○ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

- Write down all entries in the active edge table (AET) when the current scanline is at position 1, 2, and 3, respectively (cf. image).

Answer:

○ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 5.3 [3 pts]: Scanline algorithm for z-buffering.** Assume you want to apply the scanline algorithm from the previous subproblem to linearly interpolate the $z-$values at the edges of the given polygone.

In particular, assume the vertex $(1,1)$ has a $z-$value of 1 and the vertex $(4,6)$ has a $z-$value of 3. Calculate the Δ_z-value that we have to use in this case.

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

Problem 6: Global illumination

■ **[8 pts]: Radiosity (multiple choice question).** Which of the following statements are correct? (Mark the correct ones. No explanation required. Multiple statements might be correct. Marking incorrect answers might result in deduction of points.)

A. Radiosity is a method to calculate local illumination.

B. Radiosity is a method to calculate global illumination.

C. Radiosity is a method to calculate direct lighting.

D. Radiosity is a method that avoids color bleeding.

E. In general, radiosity produces more realistic results than ambient lighting.

F. In general, radiosity is faster than ambient lighting calculation.

G. Radiosity is defined as the amount of energy leaving a patch per unit time per unit area.

H. In radiosity, we distinguish between active lightsources and passive ones (e.g. reflections from objects).

I. Form factors are measured in energy (i.e. Watt W) per surface (i.e. square meters m^2).

J. Form factors depend on the distance between two patches, their emitted energy, and their size.

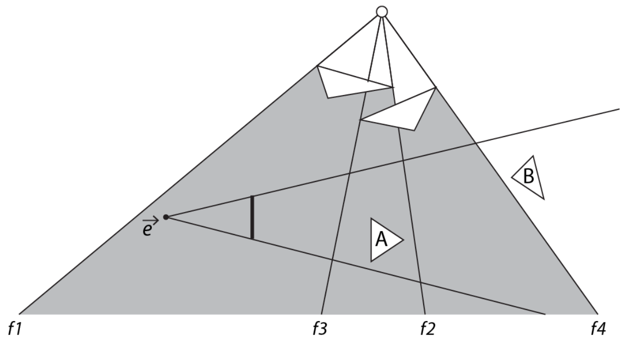
K. Form factors depend on the emitted energy of two patches, their orientation, and their size.

L. Form factors depend on the distance between two patches, their orientation, and their size.

M. None of the above.

Problem 7: Shadows

■ **Subproblem 7.1 [8 pts]: Stencil buffer.** In the following image, we have a light source (the circle at the top), two objects creating a shadow (the two unlabeled triangles at the top) and two other objects (the two triangles labeled A and B). The shadows created by the triangles A and B are not drawn in the picture because they are not relevant for this problem.



We want to render the scene on the screen (indicated by the bold line) towards the direction of our camera (indicated by the vector $\vec{e}$). To check if the triangles A and B are in the shadow or not, we are using a Stencil Buffer.

1. Shortly explain why it is better to use a *depth-fail* approach here, and not a depth-pass approach. (Note: one short sentence could be enough to get full credit here.)

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

2. We denote the faces of the shadow volume created by the first object with $f1$ and $f2$, and the one created by the second object with $f3$ and $f4$ (cf. image above). Assume they are added to the stencil buffer in the order given below. Further assume that we are using a depth-fail approach.

   For each step, fill out the following table, i.e. write down the value of an entry in the Stencil buffer that corresponds to a ray starting at the camera $\vec{e}$ and hitting the triangle A and B, respectively. (No explanation required. Just write down the number of the value that the stencil buffer would have after adding the face to a cell that corresponds to one of the triangles A and B.)

	Triangle A	Triangle B
1. Initialize Stencil buffer:	____	____
2. Add shadow face $f1$:	____	____
3. Add shadow face $f2$:	____	____
4. Add shadow face $f3$:	____	____
5. Add shadow face $f4$:	____	____

■ **Subproblem 7.2 [5 pts]: Fake shadows.**

1. What are the three major problems that can occur when drawing so-called *fake shadows*?

   (I) ______________________________

   (II) ______________________________

   (III) ______________________________

2. The following algorithm uses a Stencil buffer to avoid two of these problems:

   (a) Reset stencil buffer counters.

   (b) Draw scene. When shadow receivers are drawn, increment corresponding stencil buffer counters.

   (c) Project shadow polygones, but only draw/blend for pixel that have a non-zero stencil value.

   (d) Reset stencil entry after drawing.

   Which two problems are solved in this algorithm and which of the lines (a)-(d) solve them? (No explanation required. It is sufficient to write down the two correct pairs of problems (I), (II), or (III) and corresponding line (a), (b), (c), or (d) in order to get full credit for this subproblem.)

   Answer:

   1st pair: ____________ 2nd pair: ____________

Problem 8: Ray tracing

■ **Subproblem 8.1 [10 pts]: Ray-triangle intersection.**

1. Write down a mathematically correct description of a ray $\vec{r}$ that starts at eye vector $\vec{e} = (\frac{1}{2}, \frac{3}{4}, \frac{1}{2})$ and goes through the screen at position $\vec{s} = (\frac{1}{2}, \frac{3}{4}, \frac{3}{2})$.

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

2. Assume we want to check if our ray intersects with the triangle defined by the three vertex vectors $\vec{a} = (1,1,1)$, $\vec{b} = (0,1,1)$, and $\vec{c} = (1,0,1)$. Write down the parametric equation of the plane defined by this triangle. Use $\vec{a}$ as support vector and $\vec{b} - \vec{a}$ and $\vec{c} - \vec{a}$ as direction vectors. (The calculations in the following subproblems should become very simple then.)

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

3. In order to verify if our ray $\vec{r}$ intersects with the above triangle, we want to calculate the intersection point of the ray with the plane from the previous subproblem. We do this by creating a linear equation system using the line and plane equations that we specified above. Create this linear equation system, solve it, and calculate the intersection point.

   Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

4. Use the solution of the linear equation system constructed in the previous subproblem to verify if the ray $\vec{r}$ intersects with the triangle or not. (Hint: remember that the way we constructed our plane is the same way in which we would create a related barycentric coordinates system. If you didn't do the calculation above, but know how to do this test, you can get at least some credit if you write down the correct conditions.)

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 8.2 [3 pts]: Instancing (multiple choice question).** Mark the correct answer. No explanation required. There is only one correct answer.

Assume the following:

- O^* is an object created by multiplying an object O with a transformation matrix M.
- $\vec{r}$ is a ray created by multiplying a ray $\vec{r^*}$ with the inverse of this transformation matrix, i.e. with M^{-1}.
- $\vec{p}_i$ are the intersection points of $\vec{r}$ and object O

Given these assumptions, we can calculate the intersection points $\vec{p^*}_i$ of $\vec{r^*}$ and O^* with ...

A. ... $M\vec{p}_i$

B. ... $M^{-1}\vec{p}_i$

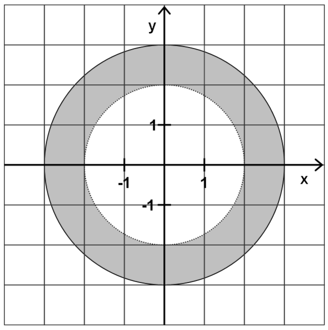
C. ... $M^T\vec{p}_i$

D. ... $(M^{-1})^T\vec{p}_i$

E. ... $(M^T)^{-1}\vec{p}_i$

F. ... none of the above

■ **Subproblem 8.3 [6 pts]: Constructive solid geometry.** We want to create an object in 2D that looks like the ring in the following image. We are using Constructive Solid Geometry and a circle C_1 with radius 2 and a circle C_2 with radius 3. Both circles are centered around the origin.



1. Write down the concrete set operation that has been done to construct our ring.

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

2. Now we want to use Constructive Solid Geometry to calculate the intersection points of our ring with the $y-$axis, i.e. with the line $x = 0$.

   (a) Write down the intervals that you get when calculating the intersections with the original objects, i.e. the circles C_1 and C_2.

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

   (b) Write down the intervals that you get when applying Constructive Solid Geometry to the previously created intervals.

   Answer:

   ◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

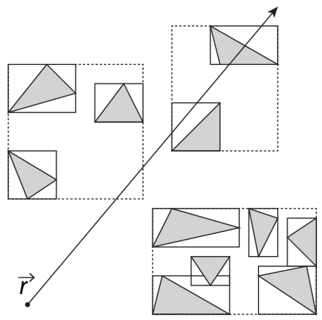
(c) How do you get the intersection points from the calculated intervals?

(Note that it is not necessary to write them down but you should illustrate how to get them. The actual values are easy to see from the image and don't matter here. What we want to know is if you understood the procedure. One sentence can be enough to get full credit for this subproblem.)

Answer:

◯ If you need more space, make a mark in the circle & continue on the separate paper provided by us.

■ **Subproblem 8.4 [6 pts]: Hierarchical bounding boxes for faster ray tracing.** Assume we want to calculate the intersections of the ray $\vec{r}$ depicted in the image below with the gray triangles given in this scene. To speed things up, we put bounding boxes around our triangles (indicated by the solid rectangles) and hierarchically grouped them (indicated by the dotted rectangles).



In the following, just write down the correct number (no explanation required). Notice that the first question asks for the total number of intersection tests (i.e. the number of intersection tests with bounding boxes *and* the necessary tests with objects therein).

When calculating the intersections using this structure of hierarchical bounding boxes . . .

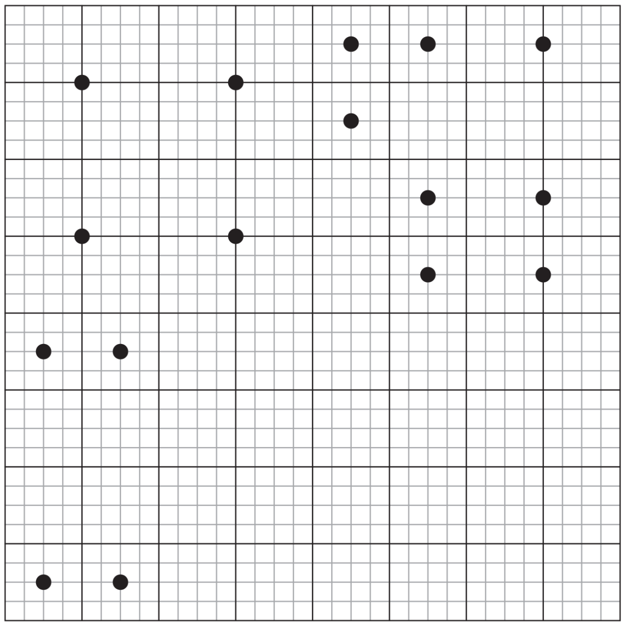
1. How many intersection tests do we have to make? ________________

2. How many false positives do we get? ________________

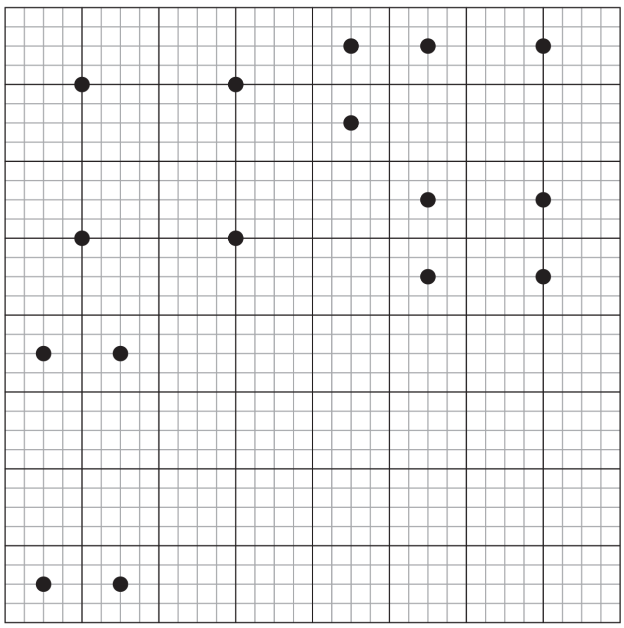
3. How many false negatives do we get? ________________

■ **Subproblem 8.5 [8 pts]: Space partitioning algorithms.** Below you see the illustration of a 2D space containing objects (black dots). The grid structure has no real relevance but was just added to make drawing easier. Now we want to apply different space partitioning approaches to these scenes.

1. Draw the cells into the image that we get when using the *quadtree approach* for space partitioning (which is the 2D version of the octree approach in 3D). Stop the space partitioning once each of the cells contains a maximum of two objects.



2. Now we want to apply the BSP tree approach for space partitioning to the same scene. Draw the resulting cells into the image below. Start with a vertical split and then alternate between horizontal and vertical splits. Stop once each cell contains a maxium of two objects.



Note: if you made a mistake in your drawing that cannot be easily corrected, you can ask one of the assistants for another template (we have a few backups).