Graphics 2010/2011

T2

Final exam

Tue, July 5, 2011

COMMENTS AND SOLUTIONS

No responsibility is taken for the correctness of the provided information.

Note that there can be more than one correct solution for most problems. Also, the following information should not be considered as standard solutions, but rather as comments that may go way beyond what was required to achieve the maximum credits for a particular subproblem.

1

Problem 1: Texture mapping

■ **Subproblem 1.1 [1 pts]:** Give a procedure that creates a stripe texture. The stripes should have alternating colors *color1* and *color2* along the $Z-$axis. The width of all stripes should be π.

■ **Solution/comments:** The solution to this problem is similar to the one we had on the slides in the lecture for stripes along the $X-$axis. The only change in the procedure is in the condition (now we check for $\sin z_p > 0$ instead of $\sin x_p > 0$).

stripe(x_p, y_p, z_p){
  if ($\sin z_p > 0$)
    return color0;
  else
    return color1;
}

■ **Subproblem 1.2 [0.5 pts]:** Which of the following statements are correct? List the correct number(s). (This is a multiple choice question. No explanation is required. Listing incorrect answers might result in deduction of points.)

1. Bump mapping causes an apparent change of geometry.
2. Bump mapping produces correct silhouettes of objects.
3. Bump mapping modifies the points for which we do the shading.
4. Bump mapping modifies the normals at the points for which we do the shading.

■ **Solution/comments:** The correct statements are number 1 and 4.

Explanation (not required): Bump mapping changes the normal at the point where we do the shading, but the actual point itself is not modified (hence, 4 is correct, but 3 is incorrect). As a consequence, it only causes an apparent change of geometry (because the actual points are not changes; hence, 1 is correct). This in turn can lead to incorrect silhouettes of objects (hence, 2 is incorrect).

Problem 2: Perspective projection

■ **[2.5 pts]:** Applying the perspective transform matrix P to a point $(x,y,z,1)$ gives us the point $(x_s,y_s,z_s,1)$ as follows:

$$P\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z\frac{n+f}{n} - f \\ \frac{z}{n} \end{pmatrix} \xrightarrow{\text{homogenize}} \begin{pmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ n+f-\frac{fn}{z} \\ 1 \end{pmatrix} = \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}$$

2

Prove that the following statements for the value $z_s = n + f - \frac{fn}{z}$ are correct:

1. Values on the near plane stay on the near plane.
2. Values on the far plane stay on the far plane.
3. Values within the view frustum stay within the view frustum.
4. The order along the $Z-$axis for any two random values is preserved.

■ **Solution/comments:**

1. Values on the near plane stay on the near plane, that means that **if** $z = n$**, then** $z_s = n$:

$$z_s = n + f - f = n = z$$

q.e.d.

2. Values on the far plane stay on the far plane, that means that **if** $z = f$**, then** $z_s = n$:

$$z_s = n + f - n = f = z$$

q.e.d.

3. Values within the view frustum stay within the view frustum, that means that **if** $z > n$ **then** $z_s > n$:

$$z_s = n + f - \frac{fn}{z} > n + f - \frac{fn}{n} = n$$
$$\text{hence } -\frac{fn}{z} > -\frac{fn}{n}$$
$$\text{and that's correct because } z > n.$$

and **if** $z < f$ **then** $z_s < f$:

$$z_s = n + f - \frac{fn}{z} < n + f - \frac{fn}{f} = f$$
$$\text{hence } -\frac{fn}{z} < -\frac{fn}{f}$$
$$\text{and that's correct because } z < f.$$

q.e.d.

4. The order along the Z-axis is preserved, that means that **if** $z_1 > z_2$ **then** $z_{1s} > z_{2s}$:

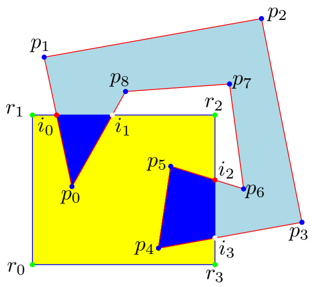
$$z_{1s} = n + f - \frac{fn}{z_1} > n + f - \frac{fn}{z_2} = z_{2s}$$
$$\text{hence } -\frac{1}{z_1} < -\frac{1}{z_2}$$
$$\text{and that's correct because } z_1 > z_2$$

q.e.d.

Problem 3: Clipping

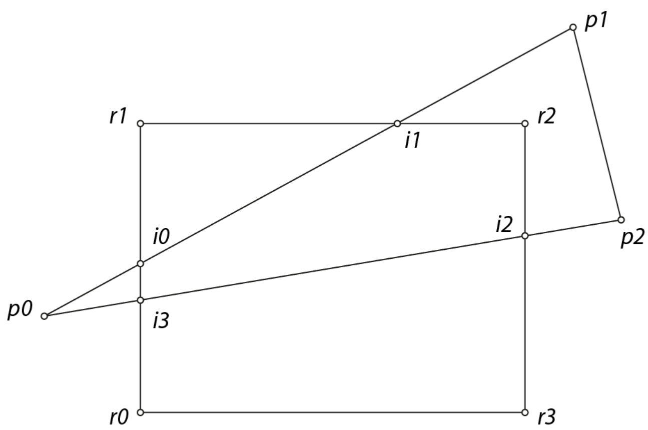
■ **Subproblem 3.1 [0.5 pts]:** Draw an example for a case in which the Sutherland-Hodgman algorithm for clipping arbitrary polygons doesn't work correctly, i.e. the result is a degenerated polygon.

■ **Solution/comments:** Pretty much any case where a polygon intersects with the clipping area in at least two spots that are not connected with each other, e.g. something like this:

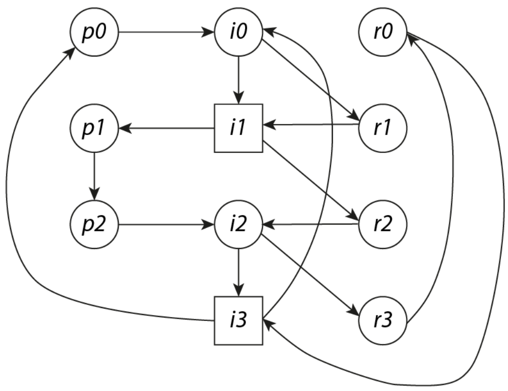


■ **Subproblem 3.2 [1 pts]:** Draw the graph that we get when we apply the Weiler-Atherton algorithm to clip the polygon defined by $p0, p1, p2$ on the clipping region defined by $r0, r1, r2, r3$.

Use the notation given in the image, always start with the lowest index number (i.e. $p0$ and $r0$), and don't forget to clearly mark what nodes in your graph represent incoming and outgoing intersections (e.g. by using a circle for incoming and a square for outgoing edges in your drawing).



■ **Solution/comments:**



Problem 4: Backface culling

■ **[1.5 pts]:** Assume we have a triangle that is part of a closed polygon. The triangle defines a face $f(\vec{p}) = (3,2,0) \cdot \vec{p} - 3 = 0$ (notice that this is the implicit equation of a plane and $(3,2,0)$ is a normal vector of this plane).

1. Assume we place a camera at $\vec{e}_1 = (2,1,0)$. Is it on the positive or negative side of $f(\vec{p})$? Prove or explain your answer.
2. Assume we the camera to $\vec{e}_2 = (1,-4,0)$. On which side of $f(\vec{p})$ is it now – the positive or negative side? Prove or explain your answer.
3. Assume that the triangle is part of our view frustum in both cases. In which case do we remove it when we apply backface culling (none, e_1, e_2, or both)? Explain your answer.

Short explanations are sufficient but no credits will be given for correct answers with missing or wrong explanation!

■ **Solution/comments:** A point $\vec{p}$ is on the positive or negative side of $f(\vec{p})$ if $f(\vec{p}) > 0$ and $f(\vec{p}) < 0$, respectively.

For the first subproblem, we get:

$$f(\vec{e}_1) = (3,2,0) \cdot (2,1,0) - 3 = 3 \cdot 2 + 2 \cdot 1 + 0 \cdot 0 - 3 = 5 > 0$$

Hence, our camera is on the positive side.

For the second subproblem, we get:

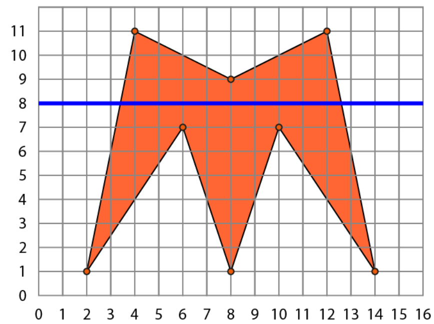
$$f(\vec{e}_1) = (3,2,0) \cdot (1,-4,0) - 3 = 3 \cdot 1 - 2 \cdot 4 + 0 \cdot 0 - 3 = -8 < 0$$

Hence, our camera is now on the negative side.

If a triangle is in the view frustum and our camera is on the positive side of the related face, there is a chance that the camera can see it (if it's not hidden by another object). Hence, we can not remove it if our camera is placed at $\vec{e}_1$. f a triangle is in the view frustum and our camera is on the negative side of the related face, there is no way that the camera can see it (since it is part of a closed polygon), so we can remove it.

Problem 5: Triangle rasterization

■ **[2 pts]:** We want to apply the scanline conversion algorithm to rasterize the polygon shown in the image below.



1. The entries in the Edge Table specify the edges of our polygon. One of them is $1 : (2, 7, \frac{2}{3})$. Explain what each of the four numbers means (a short phrase for each number is enough to answer this quesion; no detailed explanation is needed).
2. The entries in the Active Edge Table specify the edges of our polygon that intersect with the current scanline. Give all entries of the scanline at position number 8.

■ **Solution/comments:** In the Edge Table …

- "1:" gives the scanline where the edge starts
  (= the $y-$coordinate of the lowest vertex)
- 2 is the $x-$coordinate of the lowest vertex
- 7 is the $y-$coordinate of the highest vertex
- $\frac{2}{3}$ is the value of Δ_x for the edge
  (= the change of x when moving from one scanline to the next)

The entries in the Active Edge Table at position 8 are:

- $8 : (3\frac{2}{5}, 11, \frac{1}{5}), (12\frac{3}{5}, 11, -\frac{1}{5})$

Notice that the first value in the first entry is $2 + 7 * \frac{1}{5} = 3\frac{2}{5}$ because we added $\frac{1}{5}$ seven times to the original value 2. In a similar way, we get $12\frac{3}{5}$ in the second entry.

Notice on the mistake in the original exam: There was a mistake in the original exam. Instead of $1:(2,7,\frac{2}{3})$, the following (wrong) Edge Table entry was given: $1:(1,7,\frac{3}{2})$. Once I realized that, I informed the participants and to be fair, told them the solution for the first wrong entry (= x-value of the lowest vertex). My applogies for that. However, it didn't really matter, because the question was to describe what the entries in an Edge Table mean, so the actual values don't matter for the answer at all (which is probably why I didn't notice the mistake, because normally I double or even triple check everything). I understand of course that it can be confusing to have an example with incorrect values, and hence, the grading was done in a very generous way here in order to make sure that no one has any disadvantages due to my mistakes. For example, you also got full credits if you wrote Δ_y instead of Δ_x for the last entry because the wrong example actually does state Δ_y instead of Δ_x. For those who did their exam in another room, I considered in the grading that they couldn't hear my comments during the exam, of course.

Another mishap was that parts of the solution (The two sentences starting with "Notice that the first value ..." from above) did appear in the actual exam! I have no clue on why this happened. There was a misplaced bracket in the latex code that explains how it got in there. However, this doesn't explain why it was not printed in blue (what in turn explains why I didn't see it when printing the final exam). Another thing that I'm totally clueless about is how a handful of students still managed to get that part of the answer wrong.

Problem 6: Radiosity

■ **[1.5 pts]:** The radiosity B_i of a patch A_i can be calculated as follows:

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

1. The factor F_{ij} is called form factor from A_i to A_j and specifies the fraction of the energy leaving from patch A_i that arrives at patch A_j. Its value depends on three criteria. What are these?
2. What do the other three parameters describe that you find in this formula, i.e. what do E_i, ρ_i, and the B_js stand for?

Notice that you do not have to write down any formula here. A short verbal description is sufficient to get full credit (in most cases, just 1-2 words will be enough).

■ **Solution/comments:** Form factors depend on

- the **shapes of the two patches**
- their **distance**
- their **orientation**

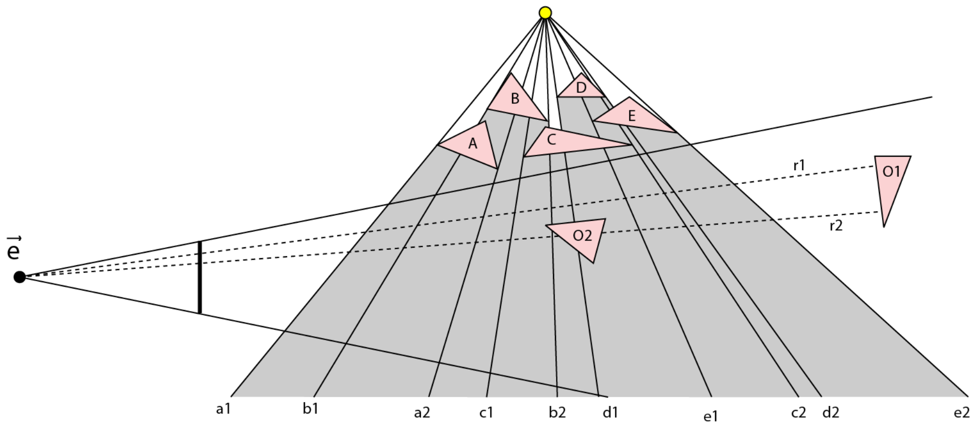
The other values of the formula stand for

- E_i is the **energy emitted by the object** itself
- The B_js are the **radiosities of all the other objects** (i.e. the incomming radiosities)
- ρ_i represents the **reflectivity factor** that specifies how much of the incoming energy is reflected.

Notice that in order to get full credit, it was sufficient to write down the words printed in bold face. A correct description of the reflectivity factor without the explicit usage of the term gave full credit, too.

Problem 7: Shadows

■ **[1.5 pts]:** In the following image, you see a light source (the circle at the top) and five objects creating a shadow (the triangles labeled A till E). The faces of the shadow volumes created by these objects are labeled with $a1, a2$, $b1, b2$, $\ldots e1, e2$ for objects A, B, $\ldots E$ In addition, we have a camera position $\vec{e}$ and two objects $O1$ and $O2$.



We want to use the Stencil buffer to check if the two objects $O1$ and $O2$ are in the shadow or not. We are using a depth-pass approach.

1. Let's look at an intermediate step of the algorithm to fill the Stencil buffer. Assume that we have drawn the shadow faces $a1, a2, b1, b2, c1, d1$ and $e1$. At that time, what are the values in the entries of the Stencil buffer that correspond to the rays $r1$ and $r2$ illustrated by the dashed line in the image? (No explanation needed; just write down the corresponding values)
2. What are the values in these two entries after the algorithm is finished, i.e. once all shadow volume faces have been considered? (No explanation needed; just write down the corresponding values)
3. What does that mean for the two objects? Shortly explain your answer (a very short description is sufficient as long as it illustrates that you understood the algorithm and did not just look at the image to check if the objects are in the shadow or not).

■ **Solution/comments:**

1. The correct answers are **3** for ray $r1$ and **1** for ray $r2$ (or **2**, cf. comment below).

   Notice that you could also interpret the intersection point of ray $r2$ and object $O2$ in the image differently – in which case you would also add $d1$. Hence, writing 2 instead of 1 for ray $r2$ gave full credit here as well.

Explanation (not required): The Stencil buffer is initialized with 0. When passing a front face, we add 1. When passing a back face, we subtract 1. Because we use a depth-pass approach, we only do this till our ray hits the first object. Both rays pass all faces. In the image, front faces are convenientely indexed with odd numbers, whereas back faces have even index numbers. Hence, we can easily see that for ray $r1$, we added 5 front faces (0+5), and 2 back faces (0+5-2 = 3). The same is true for the second ray, but since we ignore faces behind the first object hit by the ray, we only add 3 front faces (0+3-2=1).

2. For $r1$, the value is **0**. For $r2$ it is **1** (or **2**, cf. comment above).
3. 0 means, that we added as many faces to the stencil buffer as we subtracted, i.e. we entered as many shadow volumes as we exited. Hence, we are no longer in one and the object should be drawn in the light.

   Any value > 0 means, that we passed more front faces than back faces, i.e. we entered more shadow volumes than we exited. Hence, we are still in (at least) one shadow volume when we hit the first object, and thus it should be drawn in the shadow (i.e. it is not necessary to explicitly mention that shadow volumes are entered and exited in the way I did it in the description above).

Problem 8: Ray tracing: basics

■ **[0.5 pts]:** Which of the following statements are correct? List the correct number(s). (This is a multiple choice question. No explanation is required. Listing incorrect answers might result in deduction of points.)

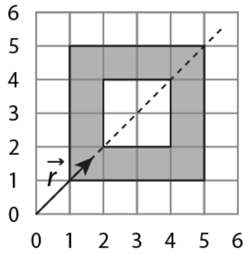
1. Viewing rays for calculating orthographic views have the same direction.
2. Viewing rays for calculating orthographic views have the same origin.
3. Viewing rays for calculating perspective views have different origins.
4. Viewing rays for calculating perspective views have the same direction.
5. If the coefficients β and γ of the barycentric coordinates of an intersection point are both between 0 and 1, the point intersects with the triangle that defines this barycentric coordinates system.
6. If a closed planar polygon and a ray that we shoot from a point $\vec{p}$ into a random direction have an uneven number of intersections, then the point must be within the polygon.

■ **Solution/comments:** The first and the last one are correct. Short explanations (not required):

1. Yes (that's why it's an orthographic projection)
2. No (their origin is the point on the screen onto which we are projecting)
3. No (they all start at the camera position; similarly to perspective projection where we project everything towards the camera position)
4. No (otherwise we would have an orthographic projection)
5. No (the correct condition is that both must be larger than 0 and their sum must be smaller than 1)
6. Yes (each intersection point represents either an exit from or an entry into the polygon; "exits" and "entries" alternate when we walk along the ray; hence, if we have an uneven number, we crossed 1 more entry than exit, so we must have started inside)

Problem 9: Ray tracing: constructive solid geometry

■ **[1 pts]:** Assume we have two squares as illustrated in the image below. Using Constructive Solid Geometry (CSG), we want to create the gray object that is also illustrated in the image.



1. Let $S1$ denote all the points covered by the bigger square and $S2$ the ones covered by the smaller one. Write down the set operation that you have to do to describe all points covered by the gray object.

2. Now we want to calculate the intersection points of the ray defined by the vector $\vec{r}$ illustrated in the image with the newly generated gray object. Write down how we do this with CSG.

   Notice that no complex calculations are required here. You can read the values from the image. However, it is not sufficient to just write down the result. It should be clear that you know how to get it using CSG.

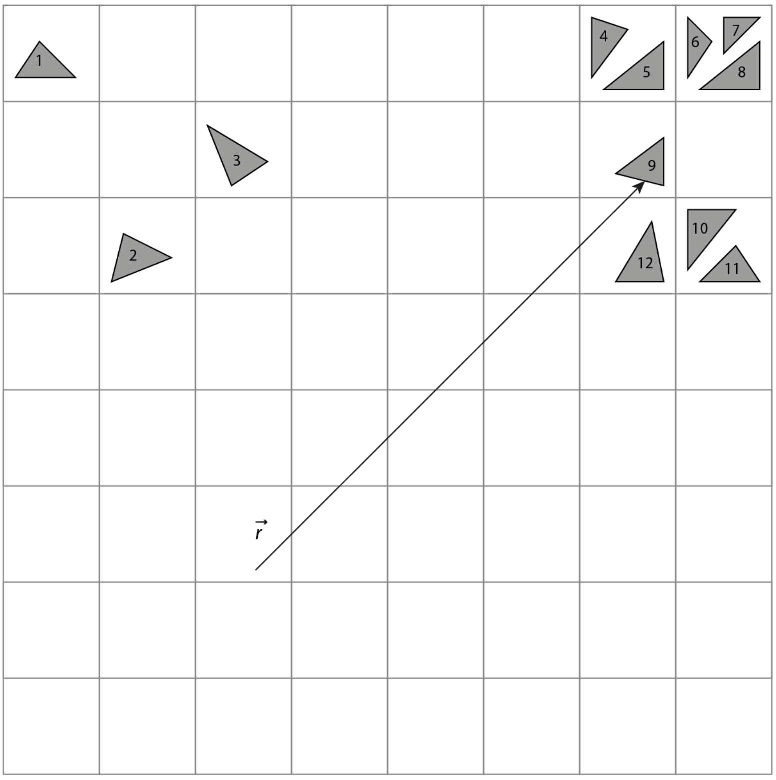
■ **Solution/comments:**

1. The correct set operation is $S1 - S2$. (Notice that using the terms "difference" or "exclusion" instead of the mathematical notation "-" gave you full credits, too, as long as it was clear that you exclude $S2$ from $S1$ and not the other way around.

2. We see from the image that the interval of all points of the ray intersecting with $S1$ is $[(1,1),(5,5)]$. The one for the intersection points with $S2$ is $[(2,2),(4,4)]$. The intervals for combined objects are calculated by applying the same set operations to these intervals that are applied to the base objects. Hence, they are $[(1,1),(2,2)]$ and $[(4,4),(5,5)]$. The borders of these two intervalls are the intersection points that we are looking for.

Problem 10: Faster ray tracing

■ **[1.5 pts]:** Assume we want to apply some space partitioning approach to the scene below in order to speed up our ray tracer.



1. We want to apply the Quadtree approach in order to particion the space in cells that contain at max. *three* objects. Starting with one huge cell that contains all triangles $[1,2,3,4,5,6,7,8,9,10,11,12]$ write down the cells that you get *in each step* of the algorithm.

   Write down the cells containing triangles using the same notation as for the starting cell indicated above. In addition, write down how many empty cells you have in each step.

2. If you do the intersection tests with the given ray $\vec{r}$ using the space partitioning that you just created with the Quadtree approach, how many false positives do you get?

3. How many false positives do you get if you apply Uniform Spacial Subdivision instead of the Quadtree approach when you use the same stop criteria (i.e. max. 3 objects per cell)?

■ Solution/comments:

1. 1st step: 2 empty cells plus $[1,2,3]$, $[4,5,6,7,8,9,10,11,12]$
   2nd step: 4 empty cells plus $[1,2,3]$, $[4,5,6,7,8,9]$, $[10,11,12]$
   3rd step: 5 empty cells plus $[1,2,3]$, $[4,5]$, $[6,7,8]$, $[9]$, $[10,11,12]$

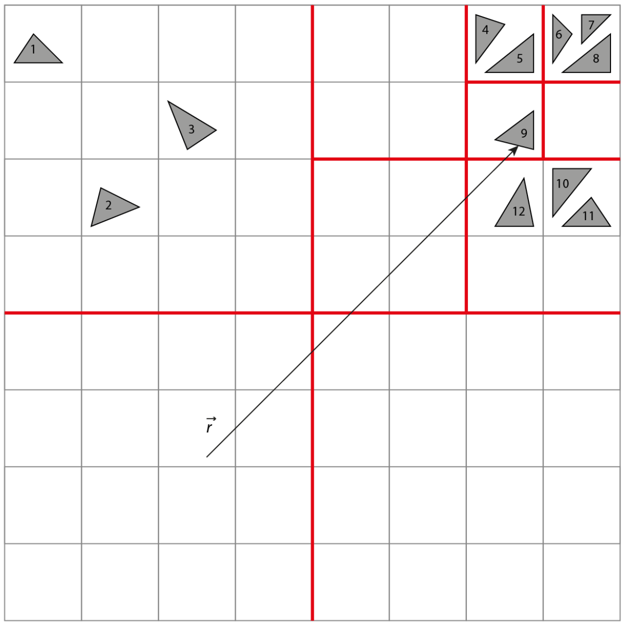
2. The correct answer is 4.

   Explanation (not required): Below you find an image illustrating the final partitioning of the space. We see that the ray crosses three empty cells, and one cell with a triangle that it doesn't intersect with (number 12), until it reaches the cell where it hits the first triangle (number 9). Hence, we have 4 cases where we have a positive intersection test with the cell, but a negative result when we look for intersecting objects in those cells.

I noticed that many students assumed that only the cells with a triangle that is not hit by the ray count as a false positive. That is technically wrong, because we only know that there are no triangles within a cell once we looked, i.e. you have to do a check based on a hit of the ray with the cell (i.e. a positive hit) that turns out to give no results (i.e. a negative hit) and thus is a false positive. However, since I never used an example with such a case in the lecture, I decided to be generous in the grading and also gave you some credits if you made this mistake.

3. The correct answer is 8.

   Explanation (not required): Because the final grid that we get conveniently matches to the gray grid illustrated in the image, we just have to count all cells that are crossed by the ray but don't have any intersection with an object. We see from the image that these are 8.



2