Full Name: ______________________________

Andrew ID: ______________________________

15-462/662, Fall 2017

Final Exam

December 11, 2017

Instructions:

- This exam is **closed book, closed notes, closed neighbor, closed cell phone, closed telepathy, closed internet**.
- You may however use a single 3in x 3in sticky note (or piece of paper) with any information you like written on both sides—-*except* for solutions to previous exams.
- All questions are multiple choice and have five possible answers; there is *one and only one correct answer for each question*. Points are assigned as follows:
  - Correct answer: +1 point
  - No answer: 0 points
  - Wrong answer: -1/4 point
- Good luck!

TOTAL SCORE

Question 1. Suppose you guess randomly on one of the questions on this exam. Assuming there is exactly one correct answer (out of five options), what is the expected number of points you will receive for that one question? Remember that you receive +1 point for the correct answer, -1/4 point for a wrong answer, and 0 points for no answer.[1]

A. 1/5 point

B. -1/5 point

C. 1/4 point

D. 0 points

E. it depends on the question

Question 2. Suppose again that you pick an answer for a question on this exam uniformly at random. What is the variance in the points you will receive for that question?

A. 5/4

B. 1/2

C. 1/4

D. 0

E. it depends on the question

Question 3. Suppose you want to paint your new bedroom purple, but the hardware store has run out of purple paint. However, they still have paints in a variety of shades of red and blue. What information is most useful for predicting how a mix of these paints will appear in your bedroom (assuming you already know what kind of lightbulbs you will use)?

A. their colors expressed as CMYK values

B. their colors expressed as RGB values

C. their locations on a chromaticity diagram

D. their emission spectra

E. their absorption spectra

Question 4. Suppose you have a leather sofa sitting by the window in your bedroom. Over time, the light coming through the window will cause the color of the leather to fade. What is the *minimum* amount of information needed to give a prediction of how your sofa will look after exactly a year of sitting in the sun? (For simplicity, you may assume that the direction of light does not influence the amount of fading.)

A. radiant energy (J)

B. radiance ($W \cdot sr^{-1} \cdot m^{-2}$)

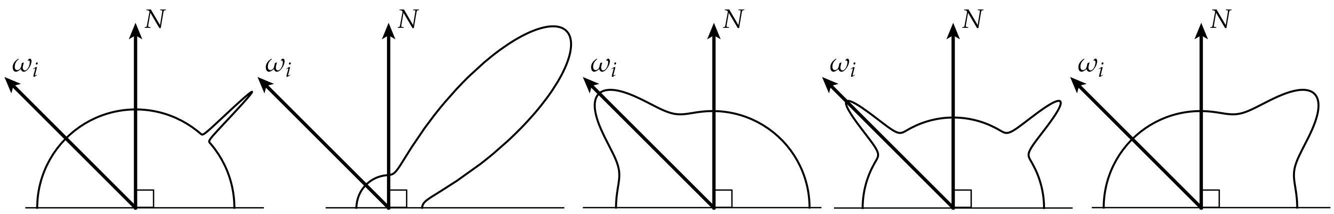
C. irradiance (W/m^2

D. radiant energy density J/m^2

E. radiant flux (W)

[1]Hopefully you now appreciate why being able to compute expected values is useful!

Question 5. To decorate your new bedroom, you decide to make some clay pottery. To glaze this pottery, you first paint it with a bright white diffuse paint, then a dull glossy glaze. For a given incoming direction ω_i, each of the images **(I)–(V)** below illustrates the probability distribution over outgoing directions ω_0 for some material. Which material looks most like your glazed pottery?



(I) **(II)** **(III)** **(IV)** **(V)**

- **A.** **(I)**
- **B.** **(II)**
- **C.** **(III)**
- **D.** **(IV)**
- **E.** **(V)**

Question 6. Suppose you now want to efficiently render the same pottery from the previous question, except now you now omit the final layer of glaze. Suppose also that the room is lit by a long string of small, bright "bistro lights." All of the following techniques will help reduce variance; which one will be especially useful for this particular scene?

- **A.** importance sampling the BRDF
- **B.** multiple importance sampling
- **C.** quasi Monte Carlo
- **D.** stratified sampling
- **E.** bidirectional path tracing

Question 7. Now that you have a nice render setup it's time to add some motion, which you decide to do via some simple physical simulation. You describe the state of the pottery via a 4x4 rotation/translation matrix, which you integrate forward in time using a numerical ODE integrator. Upon running your animation, you notice that the pottery never comes to a stop, but instead starts rolling around faster and faster over time. What integrator are you likely using, and which one might you want to try instead?

- **A.** symplectic Euler, forward Euler
- **B.** symplectic Euler, backward Euler
- **C.** forward Euler, backward Euler
- **D.** backward Euler, forward Euler
- **E.** no choice of integrator will fix this problem

Question 8. A big part of physically-based animation is detecting collisions. Suppose you have a huge apartment, but the only other things in it are a chair, a bed, a Tiffany lamp, and a picture of Albert Einstein. Each of these objects is described as a fine triangle mesh with a nice texture map. If we want to simulate a Roomba navigating the room, what spatial data structure would work best to accelerate collision detection?

A. a regular grid

B. a kd-tree

C. an adjacency list

D. an octree

E. spatial data structures don't help much here

Question 9. Suppose that you know the bounding box of the Roomba intersects a bounding box around the base of the Tiffany lamp, and now need to determine whether two individual triangles (p_1, p_2, p_3) and (q_1, q_2, q_3) from these meshes intersect. Which of the following strategies will give the correct result while doing the least amount of work?

A. Test whether any two vertices from triangle p are on opposite sides of the plane containing triangle q.

B. Test whether any edge from triangle p intersects triangle q.

C. Check if the matrix $\begin{bmatrix} q_1 - p_1 & q_2 - p_2 & q_3 - p_3 \end{bmatrix}$ has positive determinant.

D. Test whether any edge of p intersects q, *or* any edge of q intersects p.

E. Test whether either the ray from p_1 to p_2 *or* the ray from p_1 to p_3 intersects triangle q.

Question 10. An atomic operation in many geometric queries is finding ray-plane intersections. Consider the ray $r(t) := o + td$ with origin $o := (0, 1, 0)$ and direction $d := (1, 1, 1)/\sqrt{3}$, and the plane $N \cdot x = c$ with unit normal $N := (1, 0, 0)$ and offset $c = 10$. Where does the ray intersect the plane?

A. (10,11,10)

B. (10,1,1)

C. (-10,-9,-10)

D. the ray is contained in the plane

E. the ray and the plane do not intersect

Question 11. Why do we use Monte Carlo to integrate the rendering equation, rather than more standard numerical quadrature like the trapezoid rule?

A. to avoid aliasing

B. because standard quadrature is not uniform in area

C. because of the curse of dimensionality

D. so that we can get global rather than just local illumination

E. because not all functions are piecewise linear

Question 12. Which of the following is *NOT* a good argument for using rasterization rather than ray tracing?

- **A.** a rasterizer doesn't have to store the entire scene in memory at once
- **B.** rasterization is supported by modern graphics hardware
- **C.** rasterization provides order-independent transparency
- **D.** in a ray tracer, traversal of spatial data structures can require highly incoherent memory accesses
- **E.** a ray tracer may need to re-build a spatial data structure if the geometry changes

Question 13. Suppose you're driving around in your car wearing AR goggles[2]. When you get to an intersection, instead of a stoplight with a red, yellow, and green light, you see a yellow, red, and magenta light (respectively). What is the AR software doing to the three RGB color channels of your headset?

- **A.** permuting them, so that R becomes G, G becomes B, and B becomes R
- **B.** setting B=R+G, then swapping R and B
- **C.** inverting them (*i.e.*, replacing 1 with 0, and 0 with 1 in each channel)
- **D.** inverting them, then swapping R and G
- **E.** inverting them, then swapping R and B

Question 14. Suppose a 2D lamp is rigged using three points $p_0, p_1, p_2 \in \mathbb{R}^3$ which are determined by two angles θ_1, θ_2, as shown in the diagram. If the first point p_0 is always fixed to the origin, and the lengths $\ell_{01}, \ell_{12} > 0$ are just constants, what is the Jacobian expressing the small change in the positions p_1, p_2 due to a small change in the two angles?

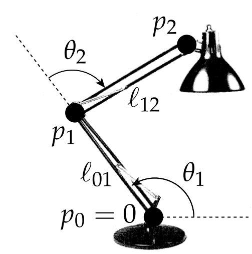
A. $$\begin{bmatrix} -\ell_{01}\sin(\theta_1) & \ell_{01}\cos(\theta_1) & -\ell_{01}\sin(\theta_1) & \ell_{01}\cos\theta_1 \\ 0 & 0 & -\ell_{12}\sin(\theta_2) & \ell_{12}\cos\theta_2 \end{bmatrix}$$

B. $$\begin{bmatrix} -\ell_{01}\sin(\theta_1) & 0 \\ \ell_{01}\cos(\theta_1) & 0 \\ -\ell_{01}\sin(\theta_1) & -\ell_{12}\sin(\theta_2) \\ \ell_{01}\cos(\theta_1) & \ell_{12}\cos(\theta_2) \end{bmatrix}$$

C. $$\begin{bmatrix} -\ell_{01}\sin(\theta_1) & -\ell_{12}\sin(\theta_2) \\ \ell_{01}\cos(\theta_1) & \ell_{12}\cos(\theta_2) \end{bmatrix}$$

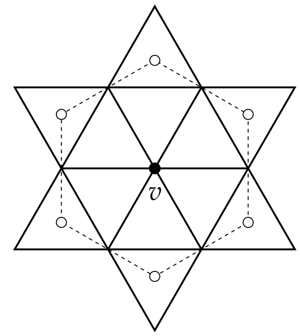
D. $$\begin{bmatrix} -\ell_{01}\sin(\theta_1) & \ell_{01}\cos(\theta_1) & 0 & 0 \\ 0 & 0 & -\ell_{12}\sin(\theta_2) & \ell_{12}\cos\theta_2 \end{bmatrix}$$

E. $$\begin{bmatrix} -\ell_{01}\sin(\theta_1) & 0 \\ \ell_{01}\cos(\theta_1) & 0 \\ 0 & -\ell_{12}\sin(\theta_2) \\ 0 & \ell_{12}\cos(\theta_2) \end{bmatrix}$$



[2]Disclaimer: Professional driver on closed course. Do not attempt.

Question 15. Suppose you have a vertex v in a halfedge mesh, and want to draw the polygon indicated in the figure, *i.e.*, a polygon whose vertices are the centers of the triangles adjacent to triangles that contain v (in some consistent order—doesn't matter if it's clockwise or counterclockwise). Which code snippet does *NOT* correctly extract the marked (white) vertices?



A.

```
HalfedgeIter he = v->he;
do {
   push_back( he->next->twin->face->center );
   he = he->twin->next;
}
while( he != v->he );
```

B.

```
HalfedgeIter he = v->he->next->next;
do {
   push_back( he->next->next->twin->face->center );
   he = he->twin->next->next;
}
while( he != v->he->next->next );
```

C.

```
HalfedgeIter he = v->he->next;
do {
   push_back( he->twin->face->center );
   he = he->next->twin->next;
}
while( he != v->he );
```

D.

```
HalfedgeIter he = v->he->next->twin;
do {
   push_back( he->face->center );
   he = he->twin->next->twin->next->twin;
}
while( he != v->he->next->twin );
```

E. they are all correct

Question 16. Consider an ellipsoid represented explicitly by the function

$$f(\theta, \phi) := (a \cos\phi \sin\theta, b \sin\phi \sin\theta, c \cos\theta),$$

for $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$. If $a = 1$, $b = 2$, and $c = 1$, is the point $p := (\frac{1}{a}, \frac{1}{2b}, \frac{1}{3b})$ inside the ellipsoid?

A. yes

B. no

C. it is exactly on the boundary

D. it is both inside and outside

E. cannot be determined from the explicit representation

Question 17. Which scenario is most likely to result in a large number of incoherent texture lookups (*i.e.*, which will need to inspect many texels that are far apart in memory)?

A. texture magnification with nearest neighbor filtering

B. texture minification with bilinear filtering

C. texture magnification with bilinear filtering

D. texture minification with MIP map filtering

E. texture magnification with MIP map filtering

Question 18. Which of the following statements about aliasing is *FALSE*?

A. aliasing is not just a computer graphics thing: aliasing can also occur in human vision (*e.g.*, wagon wheel phenomenon)

B. when using nearest neighbor filtering, viewing a triangle from a shallow angle will tend to result in more aliasing than viewing it straight on

C. supersampling is generally a more efficient way to reduce texture aliasing than prefiltering

D. a smooth signal can be perfectly reconstructed if you sample it at a rate four times higher than its highest frequency

E. for fast motion, temporal aliasing can be reduced by averaging several consecutive frames

Question 19. In 3D, let R_u^θ denote a rotation by an angle $\theta \in \mathbb{R}$ around a unit vector $u \in \mathbb{R}^3$; let $T_u(d)$ denote a translation by a distance $d \in \mathbb{R}$ along a unit direction $u \in \mathbb{R}^3$, and let $S(a)$ be a uniform scaling by a factor $a \in \mathbb{R}$. Also let x, y, z be a positively oriented orthonormal basis for $\mathbb{R}^3$. Assume sequences of transformations are applied from left to right, *e.g.*, $R_x(1.23), T_y(4.56), S(7.89)$ means *"rotate around x by* 1.23*, **then** translate along y by 4.56, **then** scale by 7.89."* All of the following transformations are equivalent except for one. Which one?

A. $R_y(\pi/2),\ T_y(10),\ S(2)$

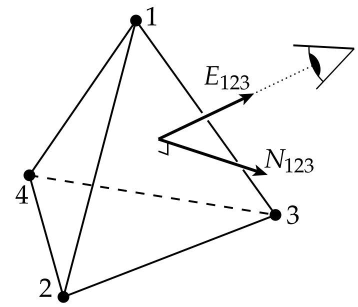
B. $S(2),\ R_x(\pi/2),\ R_y(\pi/2),\ R_z(\pi/2),\ R_y(-\pi/2),\ T_x(10),\ R_y(\pi/2)$

C. $T_x(5),\ R_y(\pi/2),\ S(2)$

D. $R_y(\pi/2),\ S(2),\ T_x(3),\ T_z(-10),\ T_x(-3)$

E. they are all equivalent

Question 20. Consider the tetrahedron depicted in the figure. For a triangle with vertices i, j, and k, let N_{ijk} denote the outward unit normal, and let E_{ijk} be a unit vector pointing from any point on the triangle toward the eye. If $N_{243} \cdot E_{243} = .72$, $N_{123} \cdot E_{123} = -.28$, $N_{134} \cdot E_{134} = -.85$, and $N_{142} \cdot E_{142} = .66$, which edges are on the silhouette of the object from the perspective of the eye? (**WARNING:** the data may not agree with the picture; you should rely on the data, not the picture.)



A. edges 14, 43, 32, and 21

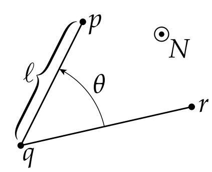
B. edges 12, 23, and 31

C. edges 12, 23, 34, 14, and 42

D. edges 13, 43, and 23

E. cannot be determined from the available information

Question 21. In reference to the figure below, what is an expression for the gradient of the angle θ with respect to the position of the endpoint p of the edge pq? We will use N to denote the unit normal sticking out of the plane, and $\ell := |p - q|$ to denote the length of the segment pq.



A. $N \times (r - q)/\ell$

B. $N \times (p - q)/\ell$

C. $N \times (p - q)/\ell^2$

D. $(p - q)/\ell$

E. none of the above

Question 22. Suppose you wanted to ray trace an animation of a "herd" of 1000 Scotty dogs roaming Flagstaff Hill. The motion of the dogs is determined by an ODE-based flocking algorithm, and each dog is covered by millions of uniformly distributed hairs. Also, each dog exhibits exactly the same motion (i.e., they all lift their left leg at the same time, then the right leg, and so on). What would be a good strategy for building a spatial data structure for this scene?

A. Build a uniform grid for the whole scene.

B. Build a kd-tree for the whole scene.

C. Build an octree for a single dog, and instance it many times inside a uniform grid.

D. Build a uniform grid for a single dog, and instance it many times inside a BVH.

E. A spatial data structure will not help with this scene.

Question 23. Which of the following colors is *NOT* a good representation of CMU's red?

A. CMYK = $(0, 1, .79, 1)$

B. RGB = $(.75, .07, .19)$

C. HSV = $(348°, .9, .76)$

D. all of them are good representations

E. none of them are good representations

Question 24. Suppose you want to numerically integrate the function $f(\theta) := 1 - \cos(\theta^2)^4$ over the interval $[0, \sqrt{\pi}]$. Which of the following pieces of code does *NOT* give a consistent, unbiased estimate of this integral?

A.

```
double estimateA( int nSamples )
{
   double sum = 0.;
   for( double i = 0; i < nSamples; i++ ) {
      double theta = sqrt(pi) * i/nSamples;
      sum += f(theta) * sqrt(pi)/nSamples;
   }
   return sum;
}
```

B.

```
double estimateB( int nSamples )
{
   double sum = 0.;
   for( double i = 0; i < nSamples; i++ ) {
      double theta = sqrt(pi) * uniformRand(0,1);
      sum += f(theta);
   }
   return sqrt(pi) * sum/nSamples;
}
```

C.

```
double estimateC( int nSamples )
{
   double sum = 0.;
   for( double i = 0; i < nSamples; i++ ) {
      double x,y;
      do {
         x = 2.*uniformRand(0,1) - 1.;
         y = 2.*uniformRand(0,1) - 1.;
      }
      while( x*x + y*y > 1 );
      double theta = .5*(atan2(y,x)+pi)/sqrt(pi);
      sum += f(theta);
   }
   return sqrt(pi) * sum/nSamples;
}
```

D.

```
double estimateD( int nSamples )
{
   double sum = 0.;
   for( double j = 0; j < 10; j++ ) {
      for( double i = 0; i < nSamples; i++ ) {
         double theta = sqrt(pi)*(uniformRand(0,1) + j)/10.;
         sum += f(theta);
      }
   }
   return sqrt(pi) * sum/(10.*nSamples);
}
```

E. all functions yield a consistent, unbiased estimate

Question 25. Suppose we have a cubic Bézier curve $\gamma_0 : [0,1] \to \mathbb{R}^2$ and want to find another cubic Bézier curve $\gamma_1 : [1,2] \to \mathbb{R}^2$ such that

- endpoints match ($\gamma_0(1) = \gamma_1(1)$)
- first derivatives match at endpoints ($\gamma_0'(1) = \gamma_1'(1)$)
- second derivatives match at endpoints ($\gamma_0''(1) = \gamma_1''(1)$)
- the first derivative at the other endpoint is given ($\gamma_1(2) = u$, for some constant $u \in \mathbb{R}^2$)

Can it be done? If so, how? (You may ignore degenerate cases.)

A. yes, by solving a linear system involving a real 8×8 matrix

B. yes, by solving two independent linear systems involving real 3×3 matrices

C. yes, by solving a linear system involving a real 3×3 matrix

D. no, the new curve is underconstrained

E. no, the new curve is overconstrained