CS 184 Summer 2025
Exam 2

July 14th, 2025
Time Limit: 110 minutes

Name: _____

SID Number: _____

Room: _____

Instructions:

- Please wait to look past this page until official start time.
- You must carefully read and sign the Honor Code below.
- Initial the top of every page, in case pages separate while scanning.
- This exam is closed book, except for two 8.5×11 page of notes (double sided), handwritten.
- This exam is: 110 minutes long, 140 points, 16 pages (including this cover page) and 7 parts.
- Problem difficulty varies, so move on if feeling stuck on a problem.
- Please detach the last page of scratch paper, which will not be graded.
- If you need to make assumptions to answer a question, write these assumptions into your answer and the assumptions box at the back of the exam.
- For multiple choice questions, please fill the bubble completely next to your answer. If you simply draw a line or a check-mark or are unable to erase any mistakes, we will not count your answer for credit.

Question	Points
1	20
2	20
3	25
4	20
5	15
6	25
7	15
Total:	140

Honor Code and Academic Integrity Certification

The Honor Code is the commitment and work of students individually and as a community to uphold honesty and integrity. This commitment is comprehensive. Specifically on exams, it means not giving or receiving any help, or using any resources that are not permitted. The Honor Code requires that students take an active role in seeing to it that others as well as themselves uphold the letter and spirit of the Honor Code.

By signing I certify that:

- All the work submitted in my name for this exam is my work alone.
- I have not given or received any help during this exam.
- I have not used any resources during this exam aside from my own page of notes.
- I will inform course staff immediately about any Honor Code violations I become aware of.

I understand that a false statement on this declaration is a violation of the Honor Code, and would be subject to the highest level of disciplinary sanction under the University Code of Conduct.

Signature: _____

Date: _____

1. (Total : 20 points) True / False

   Mark each statement True or False. (1 point each)

   (1a) T ◯ F ◯ **[Soln: F]** Total internal reflection can only happen when light approaches a material with a greater index of refraction.

   (1b) T ◯ F ◯ **[Soln: F]** When calculating the impact of a 3D Gaussian Splat on an arbitrary point p, we only need to know the color (RGB) at the Gaussian mean μ.

   (1c) T ◯ F ◯ **[Soln: F]** An octree is always faster than a BVH.

   (1d) T ◯ F ◯ **[Soln: F]** The circle of confusion takes on the shape of the sensor.

   (1e) T ◯ F ◯ **[Soln: T]** Humans are generally more sensitive to changes in luminance than to changes in color.

   (1f) T ◯ F ◯ **[Soln: F]** Every unique color induces a unique (S, M, L) response from human cone cells.

   (1g) T ◯ F ◯ **[Soln: F]** A spectrometer only allows you to analyze a narrow band of wavelengths from a light source.

   (1h) T ◯ F ◯ **[Soln: F]** In the color space Y'CbCr, the luminance value is distributed across all three channels using a gamma curve.

   (1i) T ◯ F ◯ **[Soln: T]** This convolution filter will extract vertical gradients (horizontal edges):

$$\begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix}$$

   (1j) T ◯ F ◯ **[Soln: T]** BRDFs model how light is reflected from a surface.

   (1k) T ◯ F ◯ **[Soln: T]** Even with five colors of linearly independent spectral power distributions, you cannot reproduce all visible colors.

   (1l) T ◯ F ◯ **[Soln: F]** If we switch the incident and outgoing directions, an anisotropic BRDF will always give us different values.

   (1m) T ◯ F ◯ **[Soln: T]** Verlet integration can increase the stability of simulations, although it isn't physically accurate.

   (1n) T ◯ F ◯ **[Soln: F]** Forward Euler decreases in stability as the timestep approaches zero.

   (1o) T ◯ F ◯ **[Soln: F]** A solution to the inverse kinematics problem always exists, though in some cases there may be infinitely many solutions.

   (1p) T ◯ F ◯ **[Soln: T]** sRGB does not encompass the entirety of human color perception.

   (1q) T ◯ F ◯ **[Soln: T]** You can dynamically chose the step size for forwards Euler to improve stability.

   (1r) T ◯ F ◯ **[Soln: T]** Photogrammetry differs from inverse rendering techniques in that it results in a surface defined by a 3D mesh.

   (1s) T ◯ F ◯ **[Soln: T]** As light travels through a participating medium, it can be partially absorbed and scattered at any point.

   (1t) T ◯ F ◯ **[Soln: F]** The force acting on an object attached to one end of a spring is proportional to the mass of the object on the other side.

2. (Total : 20 points) Monte Carlo Ray Tracing and Russian Roulette

In Monte Carlo ray tracing, after every bounce, we decide probabilistically whether a ray continues with probability p_r, or terminates with probability $1 - p_r$.

Let T be the random variable representing the total number of bounces the ray makes before it terminates (inclusive of the current bounce).

Suppose $p_r = \frac{3}{5}$.

(2a) [2 points] What is $P(T = 1)$?

◯ 0 ◯ $\frac{1}{5}$ ● $\frac{2}{5}$ ◯ $\frac{3}{5}$ ◯ $\frac{4}{5}$ ◯ 1

(2b) [2 points] What is the probability that the ray will terminate after bouncing at least three times (assuming the ray has more than 3 possible bounces)? i.e., compute $P(T \geq 3)$.

◯ $\left(\frac{2}{5} + \frac{3}{5} \cdot \frac{2}{5} + \frac{3}{5} \cdot \frac{3}{5} \cdot \frac{2}{5}\right)$ ◯ $\left(\frac{2}{5} + \frac{3}{5} \cdot \frac{2}{5}\right)$ ◯ $1 - \frac{2}{5}$

● $1 - \left(\frac{2}{5} + \frac{3}{5} \cdot \frac{2}{5}\right)$ ◯ $1 - \left(\frac{2}{5} + \frac{3}{5} \cdot \frac{2}{5} + \frac{3}{5} \cdot \frac{3}{5} \cdot \frac{2}{5}\right)$

(2c) [3 points] What is the expected value $\mathbb{E}[T]$?

◯ $\frac{2}{5}$ ◯ $\frac{3}{5}$ ◯ 1 ◯ $\frac{5}{3}$ ◯ 2 ● $\frac{5}{2}$

(2d) [3 points] How does decreasing p_r affect rendering performance?

● Increases ◯ Decreases ◯ Stays the same

◯ I don't know but I'll take 1 point instead

(2e) Let C_t be the light contribution of a ray that survives exactly t bounces and then terminates. Then, the total light contribution is:

$$I = \sum_{t=1}^{\infty} C_t$$

The basic Monte Carlo estimator for this is:

$$I_{\text{basic}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{M} C_t$$

for N rays, which is unbiased as $M \to \infty$.

We want to rewrite this estimator for Russian Roulette using $T \geq 1$ as a random variable representing the number of bounces the path survives.

2e.i. [8 points] Derive an expression for an unbiased Monte Carlo estimator I_{rr} of I that incorporates the survival probability for a single ray. Prove the estimator is unbiased.

Solution:

$$\hat{I}_{\text{rr}} = \sum_{t=1}^{T} \frac{C_t}{\Pr(T \geq t)} = \sum_{t=1}^{T} \frac{C_t}{p_r^{t-1}}.$$

$$\mathbb{E}[\hat{I}_{\text{rr}}] = \sum_{t=1}^{\infty} C_t \, , \frac{\Pr(T \geq t)}{p_r^{t-1}} = \sum_{t=1}^{\infty} C_t = I.$$

2e.ii. [2 points] Using I_{rr}, regardless of your answer to the previous part, derive an expression for $I_{\text{rr,N}}$ with N rays.

Solution:

$$\frac{1}{N} \sum_{i=1}^{N} I_{rr}$$

3. (Total : 25 points) Cameras and Lenses

You have a highly customizable camera and you'd like to purchase camera parts that give you specific settings. You go to Sriram's Camera Company for his special August 7th sale, only to realize that certain camera parts have already sold out. Your goal is to figure out which missing part you'll need to buy in the future to satisfy your camera needs.

(3a) [2 points] You want an FOV of 90° but Sriram only has sensors of size of 2mm. What focal length, f, would you need to set your camera to?

Solution: $f = 1\text{mm}$

(3b) [2 points] You want an focal length of $20\sqrt{3}$mm with an FOV of 60°. What size sensor, h, would you need to buy?

Solution: $h = 40\text{mm}$

(3c) [2 points] If you wanted the same FOV but you also wanted to increase the focal length, how would your image quality generally change?

◯ Decrease ◯ No change ● Increase

(3d) [2 points] You really want to take photographs in dark situations. What is the equation for exposure?

Solution: Exposure = irradiance · time · gain

(3e) [2 points] You currently have a focal length of 30mm, and by installing a random aperture, you notice your camera has F-number 4. What is the aperture diameter, d?

Solution: $d = 7.5\text{mm}$

(3f) You have a couple options of aperture diameter, shutter speed, and ISO gain to work with. You experiment with a couple and notice that you'd like your exposure to be equivalent to ISO 800, F:2, and a shutter speed of $\frac{1}{250}$ seconds.

3f.i. [2 points] If you can only find a sensor with ISO 400, what would your F-number need to be to have the ideal exposure?

Solution: $F = 1.4$

3f.ii. [2 points] If you can only find a shutter with a shutter speed of $\frac{1}{125}$ seconds, what would your ISO gain need to be to have the ideal exposure?

Solution: ISO = 400

3f.iii. [4 points] If you can only get your camera to F:1 and your ISO gain to 200, what would your shutter speed need to be in order to have the ideal exposure?

Solution: Shutter Speed = $\frac{1}{250}$

(3g) Imagine you have an ideal thin lens with a focal length of 1m.

3g.i. [2 points] How far away from the lens do parallel rays, perpendicular to the lens, converge?

Solution: 1m

3g.ii. [3 points] You are looking at a dog which is 1 meter tall and 10 meters away from the lens. Calculate the position of the dog's image on the other side of the lens.

Solution: $\frac{10}{9}$m

3g.iii. [2 points] What is the magnitude of magnification of the dog by the lens?

Solution: $\frac{1}{9}$

4. (Total : 20 points) Animation and Simulation

(4a) Consider a 3-D right-handed mechanical system made up of two connected pistons:

- The first piston is fixed at the origin $(0, 0, 0)$ and extends vertically along the z-axis. It can move straight upward by any amount $h \geq 0$, so its endpoint is at $(0, 0, h)$.
- Attached to the top of this piston is a second piston that lies in the horizontal xy-plane at height $z = h$. This second piston can rotate freely around the vertical axis (i.e., rotate in the xy-plane), and it can also extend outward by any amount $r \geq 0$. The angle of rotation is given by θ, measured counterclockwise from the positive x-axis.

In short:

- $h \geq 0$ controls the vertical extension of the first piston,
- $r \geq 0$ controls the length of the second piston,
- $\theta \in [0, 2\pi)$ is the rotation angle in the xy-plane.

4a.i. [4 points] Write an expression for the euclidean coordinates (x, y, z) given (h, r, θ).

Solution: $(x, y, z) = (r\cos\theta, r\sin\theta, h)$

4a.ii. [4 points] Write an expression for the piston configuration (h, r, θ) given the euclidean coordinates (x, y, z).

Solution: $(h, r, \theta) = (z, \sqrt{x^2 + y^2}, \arctan\frac{y}{x})$

4a.iii. [1 point] Are the inverse kinematics solutions of this problem unique?

● Yes ○ No

4a.iv. [1 point] Do inverse kinematics solutions exist for all points on our 3D space?

○ Yes ● No

(4b) We are now in a 1-dimensional space with a spring centered at 0m connected to a 1kg point mass located 10m away in the negative x direction, with zero initial velocity. The spring constant is 2N/m with a rest length of 5m.

4b.i. [4 points] Using forward Euler, what is the position of the point mass at time $t = 1$ using timestep $\Delta t = .5$ and $\Delta t = 1$

Solution: With $\Delta t = .5$, $x^1 = -7.5$m

$$\begin{aligned} x^1 &= (x^{0.5}) + (v^{0.5}) \cdot 0.5 \\ &= (x^0 + v^0 \cdot 0.5) + (v^{0.5}) \cdot 0.5 \\ &= (-10 + 0 \cdot 0.5) + (v^0 + a^0 \cdot 0.5) \cdot 0.5 \\ &= (-10) + (0 + 10 \cdot 0.5) \cdot 0.5 \\ &= -10 + 2.5 \\ &= -7.5 \end{aligned}$$

Solution: With $\Delta t = 1$, $x^1 = -10$m

$$\begin{aligned} x^1 &= x^0 + v^0 \cdot 1 \\ &= -10 + 0 \cdot 1 \\ &= -10 \end{aligned}$$

4b.ii. [6 points] Using modified Euler, what is the position of the point mass at time $t = 1$ using timestep $\Delta t = .5$ and $\Delta t = 1$

Solution: With $\Delta t = .5$, $x^1 = -5.3125$m

$$
\begin{aligned}
x^{0.5} &= x^0 + \frac{1}{2}(v^0 + v^{0.5}) \cdot 0.5 \\
&= -10 + \frac{1}{2}(0 + v^0 + a^0 \cdot 0.5) \cdot 0.5 \\
&= -10 + \frac{1}{2}(0 + 5) \cdot 0.5 \\
&= -10 + 1.25 \\
&= -8.75 \\
x^1 &= x^{0.5} + \frac{1}{2}(v^{0.5} + v^1) \cdot 0.5 \\
&= -8.75 + \frac{1}{2}(5 + v^{0.5} + a^{0.5} \cdot 0.5) \cdot 0.5 \\
&= -8.75 + \frac{1}{2}(5 + 5 + 7.5 \cdot 0.5) \cdot 0.5 \\
&= -8.75 + 3.4375 \\
&= -5.3125
\end{aligned}
$$

Solution: With $\Delta t = 1$, $x^1 = -5$m

$$
\begin{aligned}
x^1 &= x^0 + \frac{1}{2}(v^0 + v^1) \cdot 1 \\
&= -10 + \frac{1}{2}(0 + v^0 + a^0 \cdot 1) \\
&= -10 + \frac{1}{2}(0 + 10) \\
&= -10 + 5 \\
&= -5
\end{aligned}
$$

5. (Total : 15 points) Radiometry and Photometry

(5a) [3 points] The magical planet of Salongo has a radius of 15 kilometers and is completely transparent. You take the elevator down to the center of the planet and look up at the surface. An alien draws a circle on the surface. From your perspective, it appears like the size of the sun on earth (the sun subtends a solid angle of approximately $6 \cdot 10^{-5}$ steradians). What is the radius, r, of the circle that alien has drawn on the surface of Salongo?

Solution: $r = \sqrt{\frac{6 \cdot 15^2 \cdot 10^2}{\pi}} = \sqrt{\frac{13500}{\pi}}$m

(5b) [3 points] Consider a small isotropic light source with a total luminous flux of 1500 lumens. You create a reflector so that somehow all the light exits evenly through only one face of a cube surrounding the light. What is the luminous intensity of the light at that face before and after you create the reflector? Use the simplest units possible.

Solution: Before: $\frac{375}{\pi}$ candela

Solution: After: $\frac{2250}{\pi}$ candela

(5c) Photometry for Comfortable Reading

5c.i. [6 points] You have a light bulb rated at 500π lumens. To achieve 500 lux on the reading surface, by what height H above the surface must the bulb be mounted? Express your answer as simplified as possible with the correct units.

Solution: $\frac{1}{2}$m

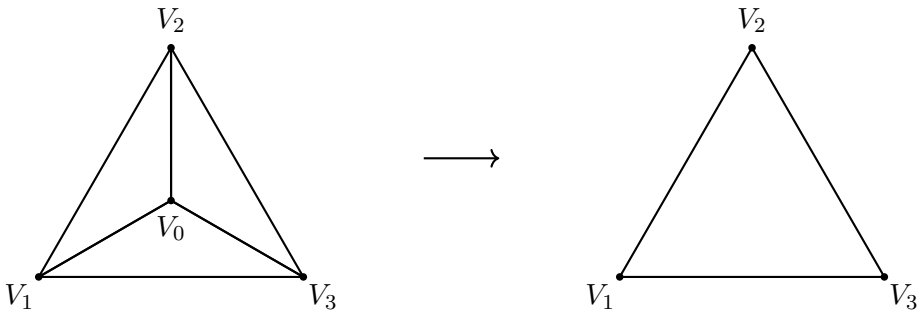
5c.ii. [3 points] By raising the lamp a little, you notice that the measured illuminance drops to $\frac{1}{9}$ of the original value. What is the new height, H', as a multiple of the old one, H?

Solution: $H' = 3 \cdot H$

6. (Total : 25 points) [Coding] Halfedges

We're working on a mesh modification software, and we would like to add some new commands. Fill out the code below to implement the new features.

(6a) [15 points] Let's consider a simplified case where three vertices (V_1, V_2, V_3) forming a triangle are all connected to another vertex, V_0, which is strictly inside. We would like to delete V_0 and in the process, remove any unnecessary faces, edges, and halfedges.



Fill out the code below to implement vertex deletion. (Hint: You should remove two faces, three edges removed, six halfedges, and one vertex. You should also have to reassign pointers for one face and three halfedges)

```
// Remember to delete items after reassignment
// For example, mesh.deleteVertex(v) deletes the vertex v.

void deleteVertex(VertexIter& v0, HalfedgeMesh& mesh) {
  // Technically, you would have to delete halfEdges,
  // but we did not specify this in the problem.
  HalfedgeIter h01 = v0->halfedge();
  HalfedgeIter h02 = h01->twin()->next();
  HalfedgeIter h03 = h02->twin()->next();
  HalfedgeIter h12 = h01->next();
  HalfedgeIter h23 = h02->next();
  HalfedgeIter h31 = h03->next();

  FaceIter f1 = h01->face();

  h12->setNext(h23);
  h23->setNext(h31);
  h31->setNext(h12);

  h12->setFace(f1);
  h23->setFace(f1);
  h31->setFace(f1);
  f1->setHalfedge(h12);

  h12->vertex()->setHalfedge(h12);
  h23->vertex()->setHalfedge(h23);
  h31->vertex()->setHalfedge(h31);

  mesh.deleteFace(h02->face());
  mesh.deleteFace(h03->face());
  mesh.deleteEdge(h01->edge());
  mesh.deleteEdge(h02->edge());
  mesh.deleteEdge(h03->edge());
  mesh.deleteVertex(v0);
}
```

(6b) [10 points] A vertex's degree is defined by the number of outgoing edges it has. We'd like to find the vertex with the greatest degree in the mesh. Assume the `Vertex.degree()` function does not exist.

Fill out the code below to find the vertex with the highest degree.

```
// Remember to account for boundary elements

VertexIter greatestVertex(HalfedgeMesh& mesh) {
  VertexIter best;
  size_t bestDeg = 0;

  for (VertexIter v = mesh.verticesBegin(); v != mesh.verticesEnd(); v++) {
    if (v.isBoundary()) continue;
    HalfedgeIter h = v->halfedge();
    if (!h) continue;

    HalfedgeIter h0 = h;
    size_t deg = !(h0->isBoundary());
    while (true) {
      while (h->isBoundary()) {
        h = h->twin()->next();
        if (h == h0) {
          break;
        }
      }
      if (h == h0) break;

      deg++;
    }

    if (deg > bestDeg) {
      bestDeg = deg;
      best = v;
    }
  }
  return best;
}
```

7. (Total : 15 points) Raytracing

We would like to trace a ray as it flows through a scene consisting of partially transparent spheres.

(7a) [10 points] Given a Sphere p and Ray r, return the number of intersection points that the ray has with the sphere. Hint: this should be almost exactly like Homework 3.

```
class Ray{
    // Public
        Vector3D o;
        Vector3D d;
    // ...
}

class Sphere : public Primitive{
        Vector3D center;
        float radius;
}
int num_intersections(Sphere p, Ray r){
    // Note that we didn't require you to use EPS_F
    Vector3D oc = r.o - p.center;
    float a = dot(r.d, r.d);
    float b = 2.0f * dot(oc, r.d);
    float c = dot(oc, oc) - p.radius * p.radius;

    float det = b*b - 4.0f*a*c;

    if (det < EPS_F) return 0;
    else if (det > EPS_F) return 2;
    else return 1;
}
```

(7b) [5 points] Fill out the following code to find the total number of intersection points a ray has with all the spheres in the scene.

```
int total_intersections(
    std::vector<Sphere*>::iterator start,
    std::vector<Sphere*>::iterator end,
    Ray r
) {
    int count = 0;
    for (auto it = start; it != end; it++) {
        count += num_intersections(*it, r)
    }
    return count;
}
```

Congrats on finishing the exam!

```
class Halfedge {
    // public
    Halfedge *twin;
    Halfedge *next;
    Vertex *vertex;
    Edge *edge;
    Face *face;

    void setNeighbors(
        HalfedgeIter next,
        HalfedgeIter twin,
        VertexIter vertex,
        EdgeIter edge,
        FaceIter face
    ); Sets the neighbors 0
}

class Vertex {
    // public
    Point pt;
    Halfedge *halfedge;
}

class Edge {
    // public
    Halfedge *halfedge;
}

class Face {
    // public
    Halfedge *halfedge;
}
```