This exam is closed book and closed notes with the exception of one sheet of notes. Please turn in the notes with the exam. You may use any algorithms or results given in class or in the Mount lecture notes. We do not expect proofs, but do expect you to support answers when asked.

The boxes here are for Gradescope. Put your primary answer in each box. If you have supporting comments, scratch work, or other, put it on other blank sections and we will be able to see and take it into account. If a blank section is small is means the answer is short, but not the reverse.

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Topic</th>
<th>Worth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Short answer</td>
<td>/16</td>
</tr>
<tr>
<td>2</td>
<td>L-System</td>
<td>/15</td>
</tr>
<tr>
<td>3</td>
<td>Navigation and obstacles</td>
<td>/15</td>
</tr>
<tr>
<td>4</td>
<td>CSG and Meshes</td>
<td>/15</td>
</tr>
<tr>
<td>5</td>
<td>Winged edge mesh representation</td>
<td>/20</td>
</tr>
<tr>
<td>6</td>
<td>A*</td>
<td>/19</td>
</tr>
</tbody>
</table>
Problem 1. Short answer (16 pts, 3-4 each). Explanations are not needed, but may be given for partial credit, or to insure we understand your answer.

a) CSG. CSG stands for: (put your answer in the box).
   a. Crazy Stupid Game
   b. Crazy Stupid Gamer
   c. Constructive Solid Geometry
   d. Contact Specular Geometry

b) Mesh. By Euler’s formula a mesh with $v = 10$ vertices and $e = 8$ edges will have $f = ?$ faces?

\[v - e + f = 2\]

So $f = 2 - v + e = 2 - 10 + 8 = 0$ too few faces

c) L-system. If you have an L-system with four production rules, give two ways you might randomize a shape produced by the system.

A) Give each rule a probability, randomly select one
B) Randomly change angle $\alpha$ or length

d) Shading equation. Explain the elements of the specular term $(v \cdot r)^q$. How would you make highlights smaller and tighter?

\[
(v \cdot r)^q \quad v - \text{view vector to camera} \\
r - \text{reflection vector - where light source would reflect} \\
q - \text{phong coefficient} \\
\text{Increase } q \text{ to make highlights tighter}
\]

e) Particle systems. Give five possible properties of a particle in a particle system.

1) Velocity 
2) Position 
3) Lifetime 
4) Mass 
5) Forces that apply (eg, gravity)
Problem 2. L-System (15 points). Given this diagram of recursive step of a possible L-system curve, with the initiator on the left and the next step on the right, answer the questions below.

a) Give an L-system with grammar for drawing the shape defined by the curve.

\[ S = 90^\circ \quad \text{start direction} \rightarrow \]

Initiator: [Diagram]

Rule:

\[ F \rightarrow + F-F-FF+F+FFF-F-FF+F \]

b) Give the fractal dimension of the curve.

\[ n = \# \text{ pieces} = 12 \]

\[ s = \text{scale} = \frac{1}{4} \]

Dimension:

\[ \frac{\log(12)}{\log\left(\frac{1}{4}\right)} = \frac{\log(12)}{\log(4)} \]
C-obstacles. Given the obstacle of the large diamond C in this diagram answer the following:

Sketch below the C-obstacles that result from the moving shapes A and B. Your sketches can be approximate. Use the centroid of each shape.

a) With A

b) With B

c) An alternative solution is to order the edges of C and B by the angle of the line, and then construct a new polygon with the edges of C and B interspersed in order of the angle. Show that solution here using the edges as labeled above.

Order of edges CCW

2 0°
3 45°
1 90°
6 100° < 4 135°
5 225°
A ~ 250°
H 1 270°
Problem 4. CSG and Meshes (15 points).

a) CSG constructs new 3D objects by combining primitive 3D shapes like cubes, cylinders and sphere. The combining operations are union, intersection and difference, plus transformations to position and scale each shape. Assuming you have a binary tree built with primitive shapes at the leaves and each internal node an operation combing its left and right branches. How would you write an algorithm to determine if a single point $P$ is in or out of the shape?

```java
boolean isMember (p, CSGtree)
if CSGtree is leaf return $P$ is $CSGtree$.
else if $CSGtree$ is union
return (isMember($P$,$CSGtree$.left)) || (isMember($P$,$CSGtree$.right))
else if $CSGtree$ is intersection
return (isMember($P$,$CSGtree$.left)) && (isMember($P$,$CSGtree$.right))
else if $CSGtree$ is difference
return (isMember($P$,$CSGtree$.left)) && (not (isMember($P$,$CSGtree$.right)))
```

b) If you have a mesh representation of an object with a vertex list, an edge list, and a face list (this doesn't have to be a winged structure, just a basic set of lists), then how do you apply a geometric transformation to translate, rotate or scale the object?

"The key is that you only have to transform the vertices. The edges and faces only represent relationships between vertices. Only vertices have physical locations themselves."

$so \; \; \; T * Vlist$
Problem 5. Winged edge mesh representation (20 points).
The DECL representation is given on the right. Given this, answer the questions below. First: give short expressions to give:

a) e.dest: the vertex that e goes to

\[ e.\text{twin}.\text{org} \] (or \( e.\text{next}.\text{org} \))

b) e.oprev: the next half-edge that shares e's origin that comes after e in clockwise order about the vertex e.org

\[ e.\text{twin}.\text{next} \]

c) Give an edge e and the face e.left, give an algorithm to give all the faces adjacent to e.left.

```plaintext
current = e  // curr = current
do {  
   output curr.twin.left  
   curr = curr.next  
} until curr == e
```

d) Assuming that the vertex v links to one edge that has \( v = e.\text{org} \), give an algorithm to give in clockwise order all the edges that have v as their origin in any order.

```plaintext
let ec = v.edge  // The one edge that v links to

do {  
   output ec  
   ec = e.twin.next  
} until ec == v.edge
```
Problem 6. A* (15 points). In this problem we consider the performance of Dijkstra’s algorithm and A* search on the graph shown below (see Fig. 2), where the objective is to compute the shortest path from s to t. Each edge (u, v) is undirected and is labeled with its associated weight w(u, v)

For a heuristic we will use the Manhattan distance. The diagram has distance dist1(a,t) = 11 as an example – that is the sum of the x and y distances between the points.

a) Trace the execution of Dijkstra’s algorithm on this graph. For each node indicate the following: (1) list the nodes in the order in which they are processed, (2) whenever a node is processed, indicate which of its neighbors have had their d-values modified, when the algorithm terminates (that is, when t is considered for processing), indicate what the final d-values are for all nodes. If there are ties for which node is to be processed, pick the node earliest in alphabetical order.
b) Trace the execution of A* Search on this graph. Provide the same information as in 2 (a), but also provide the A* f-values for each node that is processed (recall that \( f(u) = d[u] + h(u) \)).

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(u)</td>
<td>9</td>
<td>11</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Init</td>
<td>0:0:0</td>
<td>0:0:0</td>
<td>0:0:0</td>
<td>0:0:0</td>
</tr>
<tr>
<td>1.5</td>
<td>0:4:11</td>
<td>5:6</td>
<td>11:2</td>
<td><strong>Stat with 5</strong></td>
</tr>
<tr>
<td>1.6</td>
<td>0:4:11</td>
<td>5:9:2</td>
<td><strong>Pick b = 11</strong></td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>0:4:11</td>
<td>5:9</td>
<td>11:0</td>
<td><strong>Pick c = 11</strong></td>
</tr>
<tr>
<td>1.6</td>
<td>0:4:11</td>
<td>5:9</td>
<td>11</td>
<td><strong>Pick t</strong></td>
</tr>
</tbody>
</table>
```

c) Remark on the differences (if any) in the length of the path generated and the differences (if any) in the efficiency between these two algorithms.

The final path is the same, but A* is more efficient and takes one step less by skipping node a.