15-462 Computer Graphics
Fall 2000 Midterm

This midterm will be handed out Tuesday, October 24$^{\text{th}}$ and is due at the beginning of class Thursday, October 26$^{\text{th}}$, no later than 2:00 P.M. Late Midterms will not be accepted without prior arrangements with the Instructor.

You may not discuss problems on the midterm with anyone other than the TAs of 15-462 or the Instructor. You may consult your notes, any book, or any online resource that is not chat, instant messaging, email, zephyr, etc. (The point is that you may only consult published works, and not interactively probe other people's brains.)

You should show all work that is necessary to the problem. You may skip over detailed steps in your computation but not your logic. Questions that ask for proofs will be treated as if this were an upper level math class, and questions that ask you to show something will not be held to as high a standard. Explanations should be legible and in complete sentences, and spelling and grammar counts (especially on the essay).

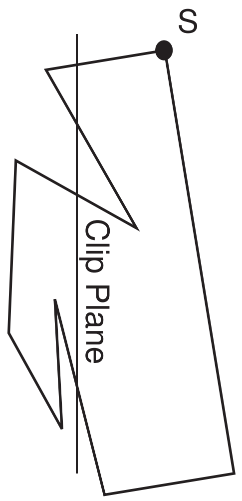

Figure 1: Sutherland-Hodgman Clipping

1 Clipping (50 Points)

1.1

The Sutherland-Hodgman polygon clipping algorithm (Listed on Page 128 of Foley and Van Dam) clips a polygon against a half plane. On figure 1.1, starting at the vertex S and going counter-clockwise, label the order the vetices are output by the algorithm.

1.2

The algorithm used in the previous section clips only against a single half plane. Explain how this can be used to clip against a convex polygonal region.

1.3

Explain why the Sutherland-Hodgman polygon clipping algorithm works for only convex clipping regions.

2 Scan Conversion (30 points)

2.1

Scan converting of polygons using a brute force method involves intersecting each scan line with every edge of a polygon and then sorting these intersections by increasing x coordinate to generate pixel spans to fill in. What is the worst case (big-O) running time of this algorithm in terms of the number of edges in the polygon (E) and the number of scan lines covered by the polygon (S)?

2.2

Using edge coherence, we can speed up the scan line algorithm by using an Active Edge Table, as described on page 98 of Foley and Van Dam. Does the worst case bound on our scan conversion change with this optimization? Why or why not?

3 Hierarchies (70 points)

You have been given several parts with which to assemble a robot to performs the task of grabbing silicon wafers from the wafer fab machine and placing them in the deposition machine. The robot must pull the wafers from the wafer fab machine horizontally, and insert them into the deposition machine horizontally. The robot base will be positioned midway between the slots on these two machines, and the top of its base post is conveniently the same height as the slots in these two machines. A sketch of the layout is in figure 3, and The parts you have to work with are in figure 3.

3.1

Using only these pieces, make a rough sketch of a robot that would accomplish this task. The robot may use each type of piece more than once, except that you are limited to one telescoping piece. One of those pieces must be the base. No need to make a beautiful drawing; it's more important that you label all the parts you're using. Indicate where the parts move using arrows, as we have done above. The arm and fingers have their origins at their axes of rotation (don't worry too much about axes).

3.2

Draw a computer graphics hierarchy of parts and transformations for your robot: use leaf nodes to represent geometric parts (e.g. a hand), internal nodes for trans-

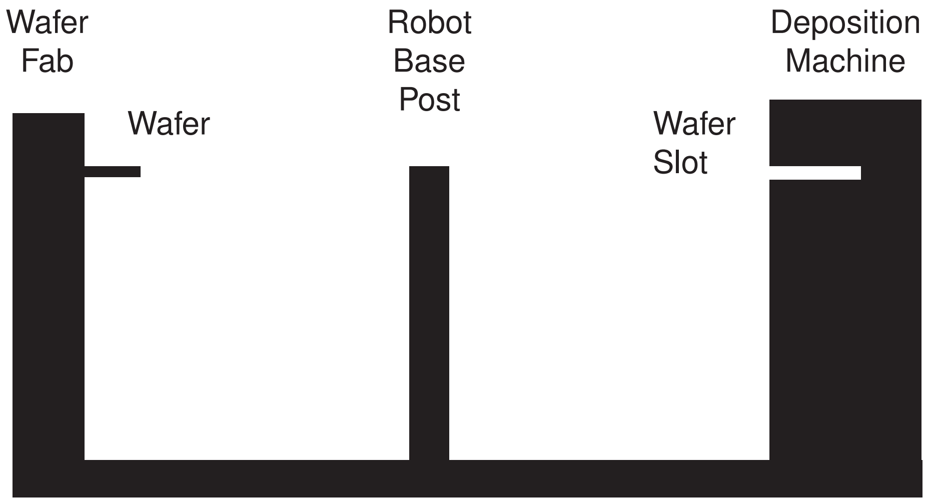

Figure 2: Silicon wafer robot base

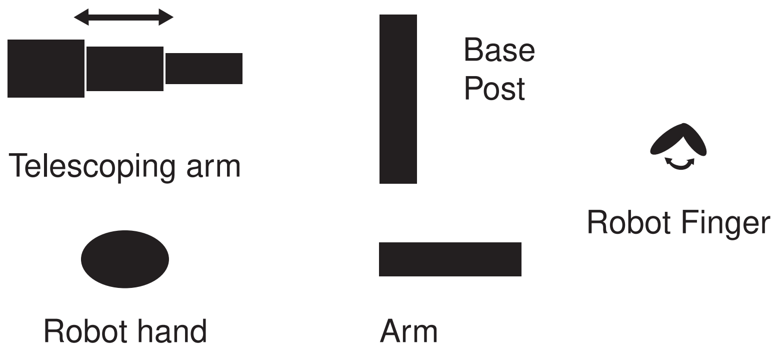

Figure 3: Silicon wafer robot parts

formations (specify type of transformation), and links to represent hierarchy (lower node is the child of higher node). The graph should be a single, connected tree. The branching factor is unlimited. Label each node.

3.3

Write OpenGL-like code to draw this robot using the following routines:

```
glTranslate();  glRotate();
glScale();      glPushMatrix();
glPopMatrix();
```

`drawBase();` – draws the base polygons in the current coordinate system
`drawArm();` – draws the arm polygons (no hand) in the current coord. sys.
`drawHand();` – draw the palm of the hand (no fingers) in the current coord. sys.
`drawFinger();` – draws one finger in the current coord. sys.

No need to specify arguments precisely enough to compile, or worry about the details of parameter ranges, but you should indicate for each transformation which parameters affect it. Some suggested parameters names (if you need more, define them clearly):

e	extension parameter for telescoping part	s	shoulder azimuth angle
f_1	First finger angle	f_2	Second finger angle

For example, a translation affected by parameter t should be written `glTranslate(`t`);`.
Be sure to get the order of routines right.

4 Most of the rest (150 points)

Read the Tech Report at
`http://graphics.lcs.mit.edu/pubs/egwr2000_raytracing.pdf`
and write a one to two page essay answering the following questions in relation to the report:

4.1

What does this report suggest to enhance the quality or versatility of ray tracing?

4.2

Describe how would an implementation of the methods affect the speed of rendering a frame? Also state why it has this affect.

4.3

What parts of the report are not clear or confusing because of background knowledge that you might not have?

4.4

What parts of the future work interest you and why?