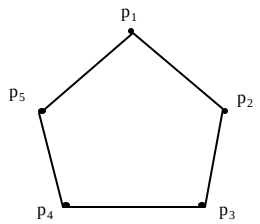
Computer Graphics

Z Sweedyk
Lecture 4
9/11/00

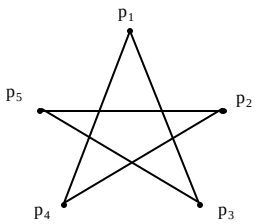
Outline

- **Polygons**
- Filled polygons

Polygon: p_1,p_2,p_3,p_4,p_5



Polygon: p_1,p_3,p_5,p_2,p_4



Order Matters

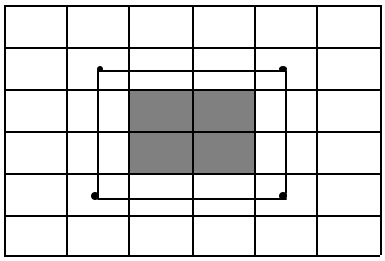
Polygon: Scan Conversion

```
Polygon(p1, …, pn)
  For i=1 to n-1
    DrawLine(pi,pi+1)
  DrawLine(pn,p1)
```

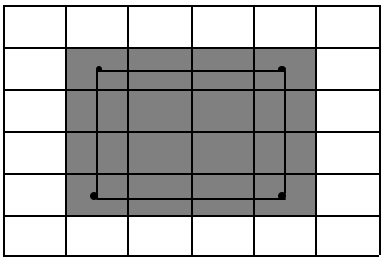

Outline

- Polygons
- **Filled polygons**

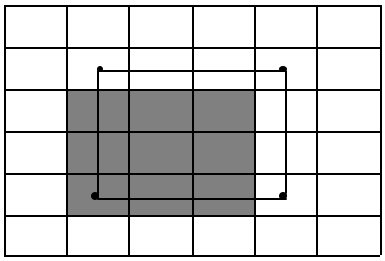
Filled Polygon
Which pixels should be on?



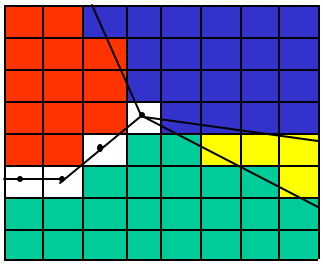
Filled Polygon
Which pixels should be on?



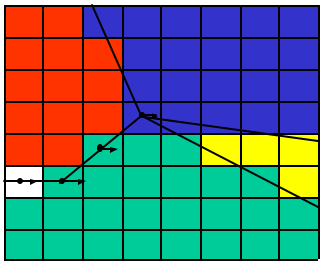
Here we get the same size!



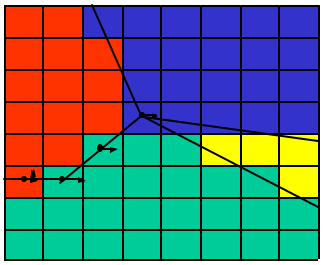
Center Claims



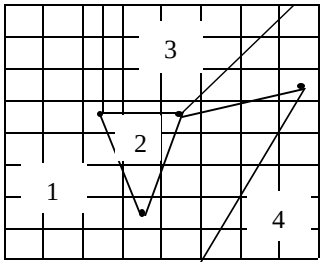
Tie Breaker 1: Entering Owns



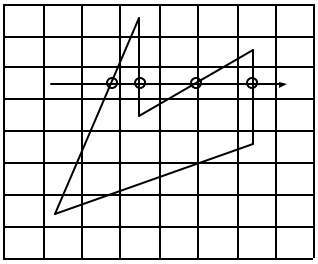
Tie Breaker 2: Up wins



Exercise

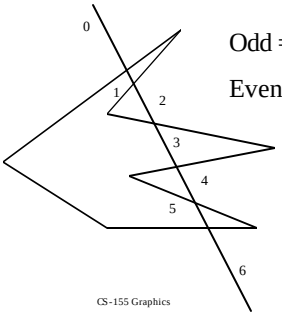


Scan Line Algorithm



1. Compute intersections
2. Order by x-coordinate
3. Use odd-even test to turn on pixels

Odd-Even Test

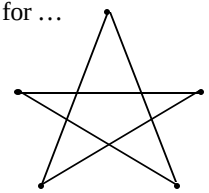


Odd = in

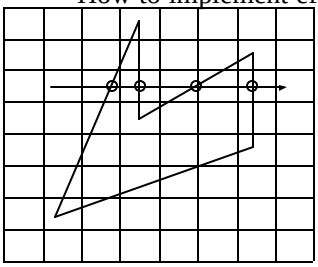
Even = out

Odd-Even Test

May not be what you're looking for … but it's easy to implement



Scan Line Algorithm
How to implement efficiently?



1. Compute intersections
2. Order by x-coordinate
3. Use odd-even test to turn on pixels

Key ideas

- Let S be the set of line segments that intersect scan line i. The set of lines that intersect scan line i+1 is:

  S+new-done

- Suppose that line L=(m,b) intersects scan line i at (x,i). If L intersects scan line i+1 it does so at:

  (x+1/m,i+1)

Data Structures

- Edge Table
- Active Edge Table

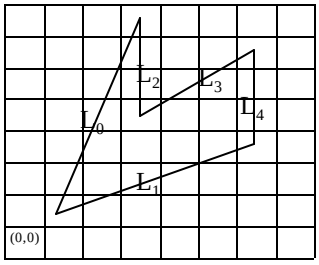
Edge Table (ET)

Yval	Lines
6	
5	
4	
3	
2	
1	
0	

Line Segments "beginning" at scan line 3

Example: Edge Table

Yval	Lines
6	-
5	-
4	L_2,L_3
3	L_4
2	-
1	L_0,L_1
0	-



Example: Edge Table

Yval	Lines
6	-
5	-
4	L_2,L_3
3	L_4
2	-
1	L_0,L_1
0	-

Record with info about L_2

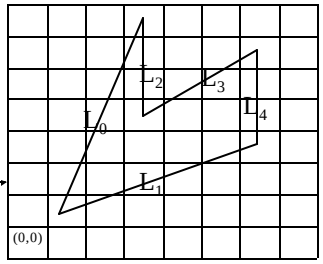
Active Edge Table (AET)

List of the Line segments intersecting current scan line

Example: Active Edge Table

L_0,L_1

Record with info about L_0



Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
- Yval ++
- Update info on line segments
- Add ET[Yval] to AET
- Remove lines from AET that are "done"
- Sort lines in AET by x-intercept at y=Yval
- Choose pixels based on odd-even test

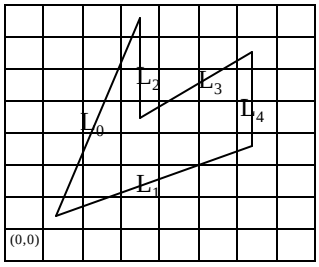
Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
- Yval ++
- Update info on line segments
- Add ET[Yval] to AET
- **Remove lines from AET that are "done"**
- Sort lines in AET by x-intercept at y=Yval
- Choose pixels based on odd-even test

Line Record

Field 1: ymax

line	ymax
L_0	7
L_1	3
L_2	7
L_3	6
L_4	6



Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
- Yval ++
- Update info on line segments
- Add ET[Yval] to AET
- **Remove lines from AET when Yval=ymax**
- Sort lines in AET by x-intercept at y=Yval
- Choose pixels based on odd-even test

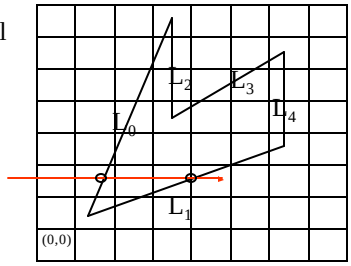
Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
- Yval ++
- Update info on line segments
- Add ET[Yval] to AET
- Remove lines from AET when Yval=ymax
- **Sort lines in AET by x-intercept at y=Yval**
- Choose pixels based on odd-even test

Line Record: CurrentYval=2

Field 2. Xval

line	xval
L_0	4/3
L_1	7/2



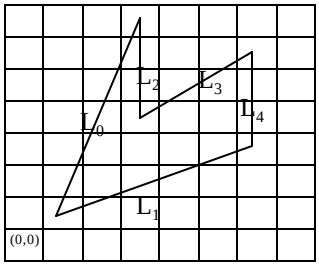
Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
  Yval ++
  Update info on line segments
  Add ET[Yval] to AET
  Remove lines from AET when Yval=ymax
  Sort lines in AET by x-intercept at y=Yval
  Choose pixels based on odd-even test

Line Record: CurrentYval=2

Field 3. 1/m

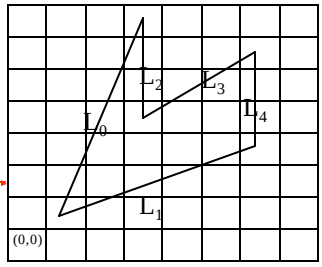
line	xval	1/m
L_0	1	1/3
L_1	1	5/2



Line Record: CurrentYval=2

Field 3. 1/m

line	xval	1/m
L_0	4/3	1/3
L_1	7/2	5/2



Scan Line Algorithm

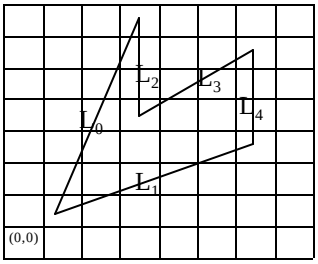
Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
  Yval ++
  Increment xval by 1/m for each line in AET
  Add ET[Yval] to AET
  Remove lines from AET when Yval=ymax
  Sort lines in AET by x-intercept at y=Yval
  Choose pixels based on odd-even test

Scan Line Algorithm

Build ET
Yval=-1
Initialize AET=φ
Repeat until ET and AET are empty:
  Yval ++
  Increment xval by 1/m for each line in AET
  Add ET[Yval] to AET
  Remove lines from AET when Yval=ymax
  Sort lines in AET by x-intercept at y=Yval
  Choose pixels based on odd-even test

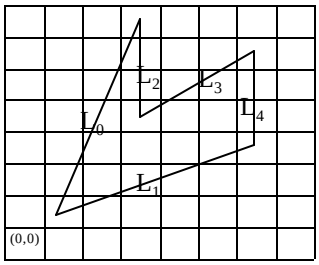
Initialize Line Records

line	ymax	x	1/m
L_0	7	1	1/3
L_1	3	1	5/2
L_2	7	3	0
L_3	6	3	3/2
L_4	6	6	0



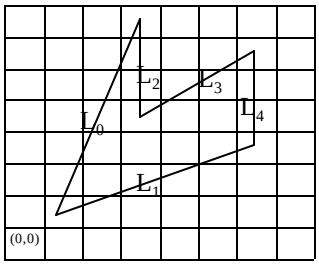
Initialize Edge Table

Yval	Lines
6	-
5	-
4	$L_2 \rightarrow L_3$
3	L_4
2	-
1	$L_0 \rightarrow L_1$
0	-



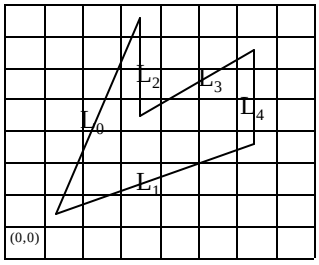
Initialize Active Edge Table

φ



Yval=0

φ

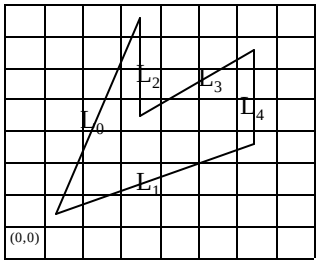


Yval=1

(sorted) AET

line	ymax	x	1/m
L_0	7	1	1/3
L_1	3	1	5/2

Turn on (i,1)
where: $1 \leq i < 1$

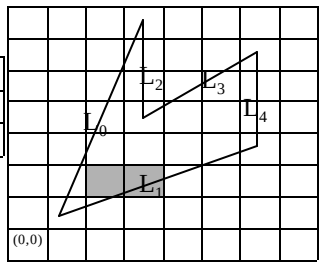


Yval=2

(sorted) AET

line	ymax	x	1/m
L_0	7	4/3	1/3
L1	3	7/2	5/2

Turn on (i,2)
where $4/3 \leq i < 7/2$

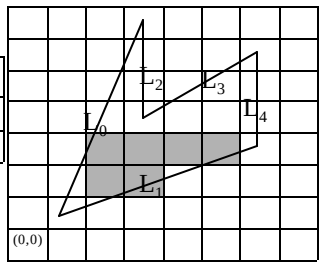


Yval=3

(sorted) AET

line	ymax	x	1/m
L_0	7	5/3	1/3
L_4	6	6	0

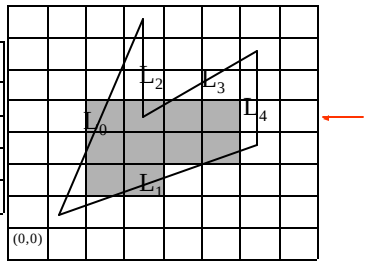
Turn on (i,3)
where $5/3 \leq i < 6$



Yval=4

(sorted) AET

line	ymax	x	1/m
L_0	7	2	1/3
L_2	7	3	0
L_3	6	3	3/2
L_4	6	6	0

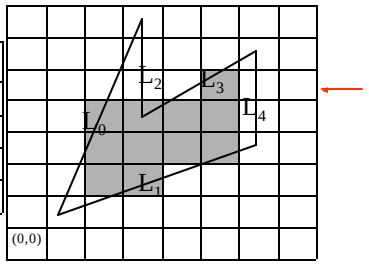


Turn on (i,4) where $2 \le i<3$ or $3 \le i<6$

Yval=5

(sorted) AET

line	ymax	x	1/m
L_0	7	7/3	1/3
L_2	7	3	0
L_3	6	9/2	3/2
L_4	6	6	0

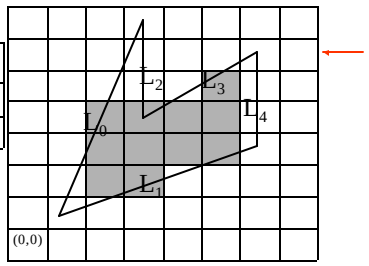


Turn on (i,5) where $7/3 \le i<3$ and $9/2 \le i<6$

Yval=6

(sorted) AET

line	ymax	x	1/m
L_0	7	8/3	1/3
L_2	7	3	0

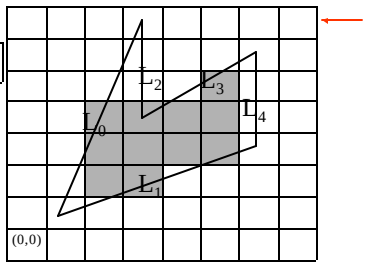


Turn on (i,6) where $8/3 \le i<3$

Yval=6

(sorted) AET

line	ymax	x	1/m



Claims

- AET always contains an even number of lines
- The algorithm implements the correct tie-breaking rules