

COMPUTER GRAPHICS PROJECT REPORT

PREPARED BY

**PRANJAL JAIN - 21UCS155
VAIBHAV RAI - 21UCS225
VANSH GUPTA - 21UCS227**

SUBMITTED TO:-

DR. ALOKE DATTA

Github Link:-

<https://github.com/Pranjal-231003/Computer-Graphics-Spongobob-Squarepaints-Through-Turtle>

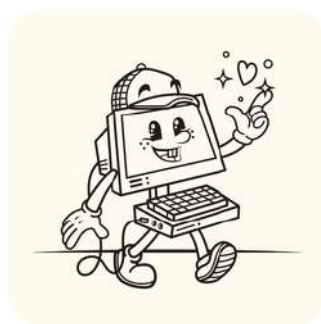
CONTENTS

<u>S.No</u>	<u>Contents</u>	<u>Page-No.</u>
1.	INTRODUCTION	1
2.	TURTLE LIBRARY	2-4
3.	IMPLEMENTATION	5-8
4.	OUTPUT	9
5.	EXPLANANTION	10-16
6.	CONCLUSION	17
7.	REFERENCES	18

INTRODUCTION

Objective:- Make your favorite cartoon character using the knowledge of Computer Graphics

In the realm of Computer Graphics, line and polygon drawing algorithms like Bresenham's Line Drawing Algorithm are utilized to render figures. These algorithms serve to approximate line segments on discrete graphical platforms such as pixel-based screens and printers. The process of line sketching on such media requires approximation, particularly in complex cases. Basic methods are employed to rasterize lines onto the screen or printing surface, typically in a single color.



TURTLE LIBRARY

The turtle library in Python provides a simple and intuitive way to create graphics and drawings using a turtle metaphor, allowing users to control a virtual turtle to draw on a canvas.

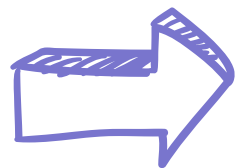
Method	Parameter	Description
Turtle()	None	Creates and returns a new turtle object
speed()	speed	Sets the drawing speed of the turtle
penup()	None	Picks up the turtle's Pen
pendown()	None	Puts down the turtle's Pen
pensize()	width	Sets the width of the turtle's pen
color()	Color name	Changes the color of the turtle's pen
begin_fill()	None	Marks the beginning of a filled area for the turtle to draw

Method	Parameter	Description
end_fill()	None	Marks the end of a filled area and fills it with the current color
goto()	x, y	Moves the turtle to the specified coordinates
setheading()	angle	Sets the orientation of the turtle to the specified angle
forward()	distance	Moves the turtle forward by the specified distance
left()	angle	Turns the turtle counterclockwise by the specified angle
right()	angle	Turns the turtle clockwise by the specified angle
circle()	radius, extent	Draws a circle with the specified radius and extent
done()	None	Stops the turtle graphics window and waits for the user to close it

Thus, using turtle and its vast drawing methods, we created a drawing of the famous cartoon character **“Spongobob Squarepants”** for our project.

Turtle handled the drawing very precisely and in a nice manner. One can install the turtle library on a system by command:

```
>>pip install turtle
```



IMPLEMENTATION

This is the code used to create “Spongebob Squarepants”.

```

1 import turtle
2 draw = turtle.Turtle()
3 draw.speed(1000)
4
5 # Background
6
7 draw.penup()
8 draw.goto(-300, 300)
9 draw.pendown()
10 draw.pensize(5)
11 draw.color("black")
12 draw.begin_fill()
13
14 draw.forward(650)
15 draw.right(90)
16 draw.forward(650)
17 draw.right(90)
18 draw.forward(650)
19 draw.right(90)
20 draw.forward(650)
21 draw.end_fill()
22
23 draw.setheading(270)
24 draw.forward(400)
25 draw.color("#FFB661")
26 draw.begin_fill()
27 draw.forward(250)
28 draw.left(90)
29 draw.forward(650)
30 draw.left(90)
31 draw.forward(200)
32 draw.right(90)
33 draw.forward(650)
34 draw.end_fill()
35
36 draw.forward(650)
37 draw.end_fill()
38
39 draw.color("#B6D7A8")
40 draw.begin_fill()
41 draw.forward(650)
42 draw.right(90)
43 draw.forward(650)
44 draw.right(90)
45 draw.forward(650)
46 draw.right(90)
47 draw.forward(650)
48 draw.end_fill()
49
50 draw.color("#87CEEB")
51 draw.begin_fill()
52 draw.forward(650)
53 draw.right(90)
54 draw.forward(650)
55 draw.right(90)
56 draw.forward(650)
57 draw.right(90)
58 draw.forward(650)
59 draw.end_fill()
60
61 draw.color("#FFDAB9")
62 draw.begin_fill()
63 draw.forward(650)
64 draw.right(90)
65 draw.forward(650)
66 draw.right(90)
67 draw.forward(650)
68 draw.right(90)
69 draw.forward(650)
70 draw.end_fill()
71
72 draw.forward(280)
73 draw.right(90)
74 draw.forward(650)
75 draw.right(90)
76 draw.forward(200)
77 draw.end_fill()
78
79 draw.color("#20B2AA")
80 draw.begin_fill()
81 draw.forward(150)
82 draw.right(90)
83 draw.forward(650)
84 draw.right(90)
85 draw.forward(150)
86 draw.right(90)
87 draw.forward(650)
88 draw.right(90)
89 draw.forward(150)
90 draw.end_fill()
91
92 draw.color("#27286C")
93 draw.begin_fill()
94 draw.forward(150)
95 draw.right(90)
96 draw.forward(650)
97 draw.right(90)
98 draw.forward(150)
99 draw.right(90)
100 draw.forward(650)
101 draw.right(90)
102 draw.forward(150)
103 draw.end_fill()
104
105 draw.color("#3297FE")
106
107 draw.begin_fill()
108 draw.forward(150)
109 draw.right(90)
110 draw.forward(650)
111 draw.right(90)
112 draw.forward(150)
113 draw.right(90)
114 draw.forward(650)
115 draw.right(90)
116 draw.forward(150)
117 draw.end_fill()
118
119 draw.color("#A8BFF7")
120 draw.begin_fill()
121 draw.forward(100)
122 draw.right(90)
123 draw.forward(650)
124 draw.right(90)
125 draw.forward(100)
126 draw.right(90)
127 draw.forward(650)
128 draw.right(90)
129 draw.forward(100)
130 draw.end_fill()
131
132 draw.color("#ACEF7F")
133 draw.begin_fill()
134 draw.forward(100)
135 draw.right(90)
136 draw.forward(650)
137 draw.right(90)
138 draw.forward(100)
139 draw.right(90)
140 draw.forward(650)
141 draw.right(90)
142 draw.forward(100)
143 draw.end_fill()
144
145

```

```

146 draw.forward(100)
147 draw.end_fill()
148
149 draw.color("#20B2AA")
150 draw.begin_fill()
151 draw.forward(100)
152 draw.right(90)
153 draw.forward(650)
154 draw.right(90)
155 draw.forward(100)
156 draw.end_fill()
157
158 draw.color("#75FEE4")
159 draw.begin_fill()
160 draw.forward(170)
161 draw.right(90)
162 draw.forward(650)
163 draw.right(90)
164 draw.forward(170)
165 draw.end_fill()
166
167 draw.forward(650)
168 draw.forward(170)
169 draw.end_fill()
170
171 draw.color("#B6FFB6")
172 draw.begin_fill()
173 draw.forward(100)
174 draw.right(90)
175 draw.forward(650)
176 draw.right(90)
177 draw.forward(100)
178 draw.end_fill()
179
180 draw.color("#90EE90")
181 draw.begin_fill()
182 draw.forward(100)
183 draw.right(90)
184 draw.forward(650)
185 draw.right(90)
186 draw.forward(100)
187 draw.end_fill()
188
189 draw.color("#90EE90")
190 draw.begin_fill()
191 draw.forward(100)
192 draw.right(90)
193 draw.forward(650)
194 draw.right(90)
195 draw.forward(100)
196 draw.end_fill()
197
198 draw.color("#87CEEB")
199 draw.begin_fill()
200 draw.forward(100)
201 draw.right(90)
202 draw.forward(650)
203 draw.right(90)
204 draw.forward(100)
205 draw.end_fill()
206
207 draw.color("#87CEEB")
208 draw.begin_fill()
209 draw.forward(100)
210 draw.right(90)
211 draw.forward(650)
212 draw.right(90)
213 draw.forward(100)
214 draw.end_fill()
215
216 draw.forward(100)
217 draw.right(90)
218 draw.forward(650)
219 draw.right(90)
220 draw.forward(100)
221 draw.end_fill()
222
223 draw.pensize(2)
224 draw.setheading(0)
225 draw.penup()
226 draw.goto(-70, -70)
227 draw.pendown()
228 draw.color("black", "yellow")
229
230 draw.begin_fill()
231 draw.forward(200)
232 draw.left(90)
233 draw.forward(220)
234 draw.left(90)
235 draw.forward(200)
236 draw.left(90)
237 draw.forward(220)
238 draw.end_fill()
239
240
241 # Brown Pants
242 draw.color("black", "#8B6914")
243 draw.begin_fill()
244
245 draw.forward(40)
246 draw.left(90)
247 draw.forward(200)
248 draw.left(90)
249 draw.forward(40)
250 draw.left(90)
251 draw.forward(200)
252 draw.left(90)
253 draw.forward(40)
254 draw.end_fill()
255
256 # Uniform
257 draw.color("black", "white")
258 draw.begin_fill()
259 draw.penup()
260 draw.left(90)
261 draw.forward(200)
262 draw.left(90)
263 draw.forward(40)
264 draw.forward(30)
265 draw.left(90)
266 draw.forward(200)
267 draw.left(90)
268 draw.forward(40)
269 draw.end_fill()
270
271 draw.penup()
272 draw.goto(18, -40)
273 draw.pendown()
274
275 draw.right(30)
276 draw.forward(25)
277 draw.right(95)
278 draw.forward(38)
279

```



```

spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
280 draw.setheading(270)
281 draw.penup()
282 draw.goto(41, -40)
283 draw.pendown()
284
285 draw.left(30)
286 draw.forward(25)
287 draw.left(95)
288 draw.forward(38)
289
290
291
292
293 # Ears
294
295 draw.color("MAGENTA")
296
297 draw.goto(25, 115)
298 draw.pendown()
299 draw.begin_fill()
300 draw.setheading(-30)
301 for i in range(2):
302     draw.circle(15,90)
303     draw.pendown()
304 draw.end_fill()
305
306 draw.penup()
307 draw.goto(-23, 130)
308 draw.pendown()
309 draw.begin_fill()
310 draw.setheading(-40)
311 for i in range(2):
312     draw.circle(6,90)
313     draw.pendown()
314 draw.end_fill()
315
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
317 draw.penup()
318 draw.goto(85, 115)
319 draw.pendown()
320 draw.begin_fill()
321 draw.setheading(-80)
322 for i in range(2):
323     draw.circle(17,90)
324     draw.pendown()
325 draw.end_fill()
326
327
328 draw.penup()
329 draw.goto(110, 85)
330 draw.pendown()
331 draw.begin_fill()
332 draw.setheading(-30)
333 for i in range(2):
334     draw.circle(6,90)
335     draw.pendown()
336 draw.end_fill()
337
338 draw.penup()
339 draw.goto(-25, -15)
340 draw.pendown()
341 draw.begin_fill()
342 draw.setheading(-60)
343 for i in range(2):
344     draw.circle(20,90)
345     draw.pendown()
346 draw.end_fill()
347
348 draw.penup()
349 draw.goto(-45, 10)
350 draw.pendown()
351 draw.begin_fill()
352
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
353 draw.setheading(-30)
354 draw.pendown()
355 for i in range(2):
356     draw.circle(75, -15)
357     draw.pendown()
358 draw.begin_fill()
359 draw.setheading(-42)
360 for i in range(2):
361     draw.circle(17,90)
362     draw.pendown()
363 draw.end_fill()
364
365 draw.penup()
366 draw.goto(95, 10)
367 draw.pendown()
368 draw.setheading(-30)
369 for i in range(2):
370     draw.circle(6,90)
371     draw.pendown()
372 draw.end_fill()
373
374 # Black box
375
376 draw.penup()
377 draw.goto(-50, -40)
378 draw.pendown()
379 draw.color("black")
380 draw.forward(12)
381
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
382 draw.forward(33)
383 draw.right(90)
384 draw.forward(8)
385 draw.right(90)
386 draw.forward(33)
387 draw.right(90)
388 draw.forward(8)
389 draw.end_fill()
390
391 draw.penup()
392 draw.goto(-5, -40)
393 draw.pendown()
394 draw.setheading(0)
395 draw.begin_fill()
396 draw.forward(22)
397 draw.right(90)
398 draw.forward(8)
399 draw.end_fill()
400
401 draw.penup()
402 draw.goto(85, -40)
403 draw.pendown()
404 draw.setheading(0)
405 draw.begin_fill()
406 draw.forward(32)
407 draw.right(90)
408 draw.forward(8)
409 draw.right(90)
410 draw.forward(12)
411

```

```

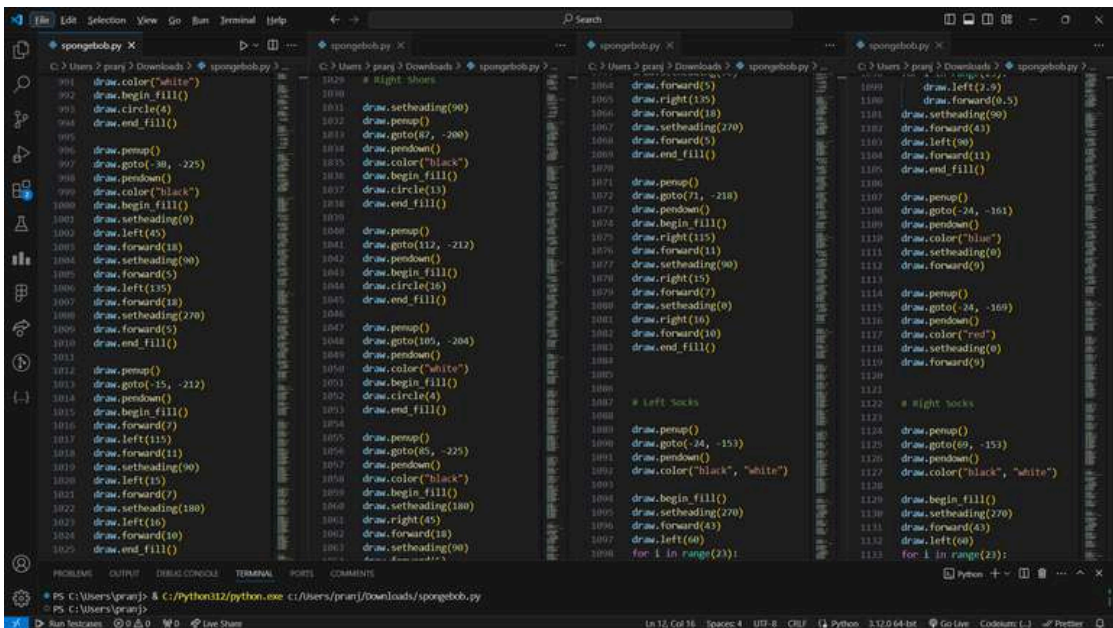
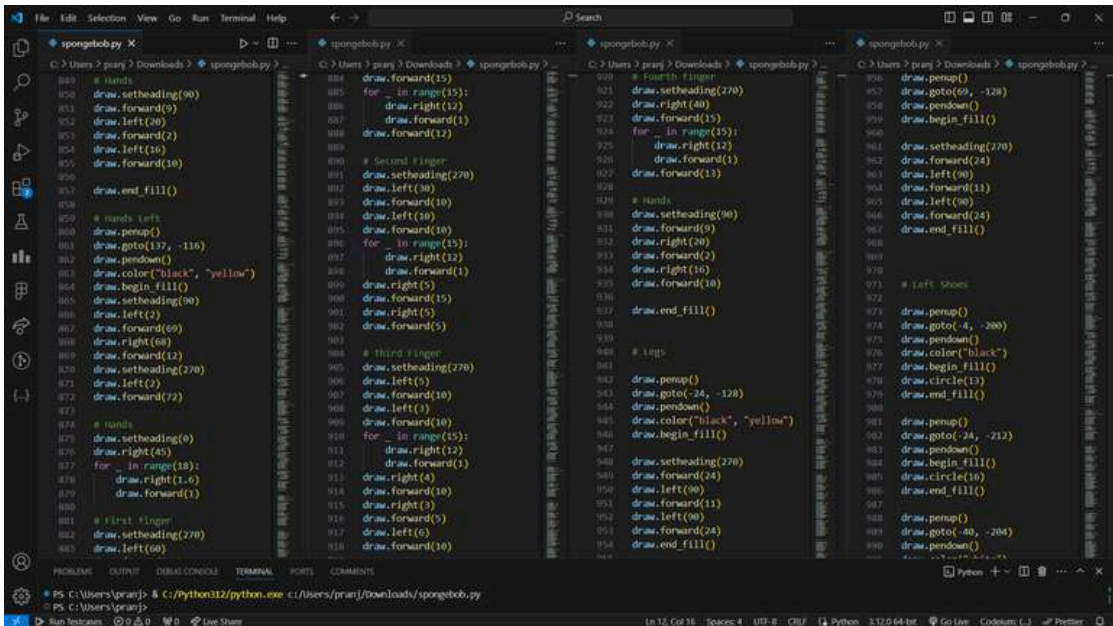
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
412 draw.forward(11)
413 draw.setheading(180)
414 draw.forward(11)
415 draw.end_fill()
416
417 # Red tie
418
419 draw.color("black", "MAGENTA")
420 draw.setheading(180)
421 draw.penup()
422 draw.goto(41, -40)
423 draw.pendown()
424 draw.begin_fill()
425
426 draw.forward(25)
427 draw.left(115)
428 draw.forward(20)
429 draw.setheading(0)
430 draw.forward(10)
431 draw.left(65)
432 draw.forward(20)
433 draw.end_fill()
434
435 draw.penup()
436 draw.goto(24, -59)
437 draw.pendown()
438 draw.begin_fill()
439 draw.setheading(270)
440 draw.right(90)
441 draw.forward(10)
442 draw.left(65)
443 draw.forward(15)
444 draw.left(90)
445 draw.forward(13)
446 draw.end_fill()
447
448 draw.penup()
449 draw.goto(33, 8)
450 draw.pendown()
451 draw.begin_fill()
452 draw.setheading(270)
453 draw.left(7)
454 draw.forward(14)
455 draw.left(90)
456 draw.forward(15)
457 draw.left(90)
458 draw.end_fill()
459
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
460 draw.forward(14)
461 draw.end_fill()
462
463 # Eyes
464
465 # Left eye
466 draw.color("black", "white")
467 draw.penup()
468 draw.goto(-30, 64)
469 draw.pendown()
470 draw.begin_fill()
471 draw.setheading(90)
472 draw.left(25)
473 for i in range(22):
474     draw.right(11)
475     draw.forward(2)
476     draw.end_fill()
477
478 # Right eye
479 draw.penup()
480 draw.goto(5, 64)
481 draw.pendown()
482 draw.begin_fill()
483 draw.setheading(90)
484 draw.right(25)
485 for i in range(23):
486     draw.left(12)
487     draw.forward(2)
488     draw.end_fill()
489
490 # Pupil
491 draw.color("black", "MAGENTA")
492 draw.penup()
493
spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
494 draw.penup()
495 draw.setheading(0)
496 draw.begin_fill()
497 draw.forward(1.5)
498 draw.forward(1)
499 draw.end_fill()
500
501 # Mouth
502 draw.penup()
503 draw.goto(-25, 40)
504 draw.pendown()
505 draw.color("black")
506 draw.setheading(0)
507 draw.right(60)
508
509 for i in range(40):
510     draw.left(1)
511     draw.forward(1.5)
512
513 draw.penup()
514 draw.goto(-30, 34)
515 draw.pendown()
516 draw.setheading(90)
517 draw.left(15)
518
519 for i in range(17):
520     draw.right(6)
521     draw.forward(1)
522

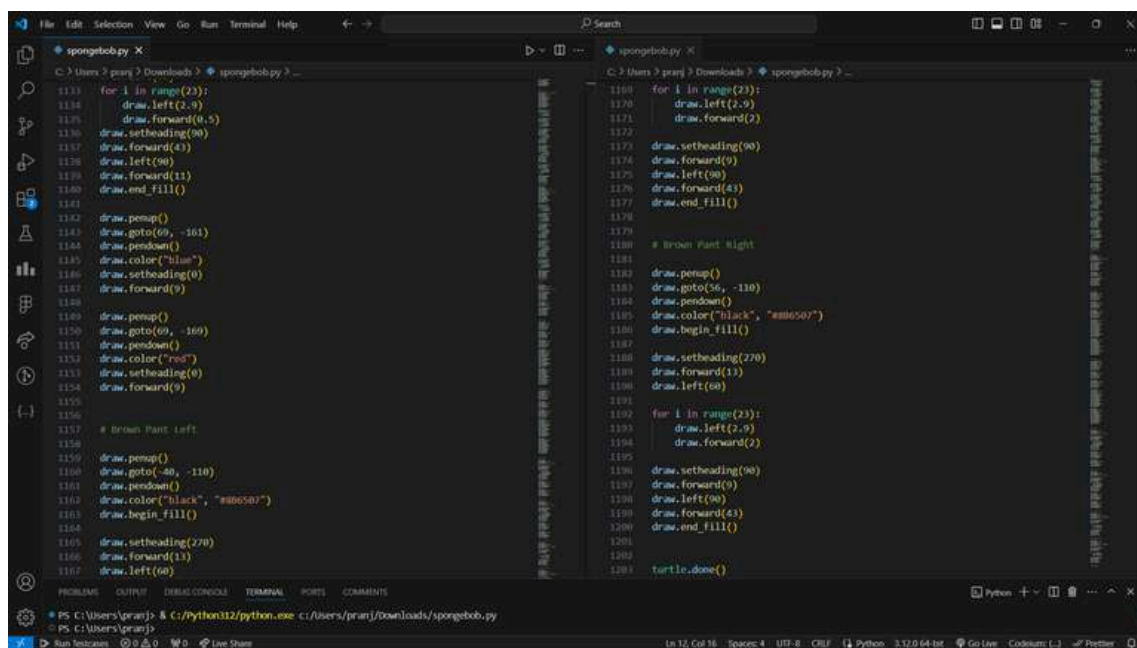
```

```

spongebob.py X
C:\Users\pranj> Downloads > spongebob.py
523 draw.left(50)
524 for i in range(45):
525     draw.left(5.1)
526     draw.forward(1.6)
527
528 draw.penup()
529 draw.goto(-37, 50)
530 draw.pendown()
531 draw.penup()
532 draw.setheading(0)
533 draw.circle(0.8)
534
535 draw.penup()
536 draw.goto(-20, 52)
537 draw.pendown()
538 draw.setheading(0)
539 draw.circle(0.8)
540
541 draw.penup()
542 draw.goto(-20, 52)
543 draw.pendown()
544 draw.setheading(0)
545 draw.circle(0.8)
546
547 draw.penup()
548 draw.goto(-20, 52)
549 draw.pendown()
550 draw.setheading(0)
551 draw.circle(0.8)
552
553 draw.penup()
554 draw.goto(-20, 52)
555 draw.pendown()
556 draw.setheading(0)
557 draw.circle(0.8)
558
559 draw.penup()
560 draw.goto(-20, 52)
561 draw.pendown()
562 draw.setheading(0)
563 draw.circle(0.8)
564
565 draw.penup()
566 draw.goto(-20, 52)
567 draw.pendown()
568 draw.setheading(0)
569 draw.circle(0.8)
570
571 draw.penup()
572 draw.goto(-20, 52)
573 draw.pendown()
574 draw.setheading(0)
575 draw.circle(0.8)
576
577 draw.penup()
578 draw.goto(-20, 52)
579 draw.pendown()
580 draw.setheading(0)
581 draw.circle(0.8)
582
583 draw.penup()
584 draw.goto(-20, 52)
585 draw.pendown()
586 draw.setheading(0)
587 draw.circle(0.8)
588
589 draw.penup()
590 draw.goto(-20, 52)
591 draw.pendown()
592 draw.setheading(0)
593 draw.circle(0.8)
594
595 draw.penup()
596 draw.goto(-20, 52)
597 draw.pendown()
598 draw.setheading(0)
599 draw.circle(0.8)
600
601 draw.penup()
602 draw.goto(-20, 52)
603 draw.pendown()
604 draw.setheading(0)
605 draw.circle(0.8)
606
607 draw.penup()
608 draw.goto(-20, 52)
609 draw.pendown()
610 draw.setheading(0)
611 draw.circle(0.8)
612
613 draw.penup()
614 draw.goto(-20, 52)
615 draw.pendown()
616 draw.setheading(0)
617 draw.circle(0.8)
618
619 draw.penup()
620 draw.goto(-20, 52)
621 draw.pendown()
622 draw.setheading(0)
623 draw.circle(0.8)
624
625 draw.penup()
626 draw.goto(-20, 52)
627 draw.pendown()
628 draw.setheading(0)
629 draw.circle(0.8)
630
631 draw.penup()
632 draw.goto(-20, 52)
633 draw.pendown()
634 draw.setheading(0)
635 draw.circle(0.8)
636
637 draw.penup()
638 draw.goto(-20, 52)
639 draw.pendown()
640 draw.setheading(0)
641 draw.circle(0.8)
642
643 draw.penup()
644 draw.goto(-20, 52)
645 draw.pendown()
646 draw.setheading(0)
647 draw.circle(0.8)
648
649 draw.penup()
650 draw.goto(-20, 52)
651 draw.pendown()
652 draw.setheading(0)
653 draw.circle(0.8)
654
655 draw.penup()
656 draw.goto(-20, 52)
657 draw.pendown()
658 draw.setheading(0)
659 draw.circle(0.8)
660
661 draw.penup()
662 draw.goto(-20, 52)
663 draw.pendown()
664 draw.setheading(0)
665 draw.circle(0.8)
666
667 draw.penup()
668 draw.goto(-20, 52)
669 draw.pendown()
670 draw.setheading(0)
671 draw.circle(0.8)
672
673 draw.penup()
674 draw.goto(-20, 52)
675 draw.pendown()
676 draw.setheading(0)
677 draw.circle(0.8)
678
679 draw.penup()
680 draw.goto(-20, 52)
681 draw.pendown()
682 draw.setheading(0)
683 draw.circle(0.8)
684
685 draw.penup()
686 draw.goto(-20, 52)
687 draw.pendown()
688 draw.setheading(0)
689 draw.circle(0.8)
690
691 draw.penup()
692 draw.goto(-20, 52)
693 draw.pendown()
694 draw.setheading(0)
695 draw.circle(0.8)
696
697 draw.penup()
698 draw.goto(-20, 52)
699 draw.pendown()
700 draw.setheading(0)
701 draw.circle(0.8)
702
703 draw.penup()
704 draw.goto(-20, 52)
705 draw.pendown()
706 draw.setheading(0)
707 draw.circle(0.8)
708
709 draw.penup()
710 draw.goto(-20, 52)
711 draw.pendown()
712 draw.setheading(0)
713 draw.circle(0.8)
714
715 draw.penup()
716 draw.goto(-20, 52)
717 draw.pendown()
718 draw.setheading(0)
719 draw.circle(0.8)
720
721 draw.penup()
722 draw.goto(-20, 52)
723 draw.pendown()
724 draw.setheading(0)
725 draw.circle(0.8)
726
727 draw.penup()
728 draw.goto(-20, 52)
729 draw.pendown()
730 draw.setheading(0)
731 draw.circle(0.8)
732
733 draw.penup()
734 draw.goto(-20, 52)
735 draw.pendown()
736 draw.setheading(0)
737 draw.circle(0.8)
738
739 draw.penup()
740 draw.goto(-20, 52)
741 draw.pendown()
742 draw.setheading(0)
743 draw.circle(0.8)
744
745 draw.penup()
746 draw.goto(-20, 52)
747 draw.pendown()
748 draw.setheading(0)
749 draw.circle(0.8)
750
751 draw.penup()
752 draw.goto(-20, 52)
753 draw.pendown()
754 draw.setheading(0)
755 draw.circle(0.8)
756
757 draw.penup()
758 draw.goto(-20, 52)
759 draw.pendown()
760 draw.setheading(0)
761 draw.circle(0.8)
762
763 draw.penup()
764 draw.goto(-20, 52)
765 draw.pendown()
766 draw.setheading(0)
767 draw.circle(0.8)
768
769 draw.penup()
770 draw.goto(-20, 52)
771 draw.pendown()
772 draw.setheading(0)
773 draw.circle(0.8)
774
775 draw.penup()
776 draw.goto(-20, 52)
777 draw.pendown()
778 draw.setheading(0)
779 draw.circle(0.8)
780
781 draw.penup()
782 draw.goto(-20, 52)
783 draw.pendown()
784 draw.setheading(0)
785 draw.circle(0.8)
786
787 draw.penup()
788 draw.goto(-20, 52)
789 draw.pendown()
790 draw.setheading(0)
791 draw.circle(0.8)
792
793 draw.penup()
794 draw.goto(-20, 52)
795 draw.pendown()
796 draw.setheading(0)
797 draw.circle(0.8)
798
799 draw.penup()
800 draw.goto(-20, 52)
801 draw.pendown()
802 draw.setheading(0)
803 draw.circle(0.8)
804
805 draw.penup()
806 draw.goto(-20, 52)
807 draw.pendown()
808 draw.setheading(0)
809 draw.circle(0.8)
810
811 draw.penup()
812 draw.goto(-20, 52)
813 draw.pendown()
814 draw.setheading(0)
815 draw.circle(0.8)
816
817 draw.penup()
818 draw.goto(-20, 52)
819 draw.pendown()
820 draw.setheading(0)
821 draw.circle(0.8)
822
823 draw.penup()
824 draw.goto(-20, 52)
825 draw.pendown()
826 draw.setheading(0)
827 draw.circle(0.8)
828
829 draw.penup()
830 draw.goto(-20, 52)
831 draw.pendown()
832 draw.setheading(0)
833 draw.circle(0.8)
834
835 draw.penup()
836 draw.goto(-20, 52)
837 draw.pendown()
838 draw.setheading(0)
839 draw.circle(0.8)
840
841 draw.penup()
842 draw.goto(-20, 52)
843 draw.pendown()
844 draw.setheading(0)
845 draw.circle(0.8)
846
847 draw.penup()
848 draw.goto(-20, 52)
849 draw.pendown()
850 draw.setheading(0)
851 draw.circle(0.8)
852
853 draw.penup()
854 draw.goto(-20, 52)
855 draw.pendown()
856 draw.setheading(0)
857 draw.circle(0.8)
858
859 draw.penup()
860 draw.goto(-20, 52)
861 draw.pendown()
862 draw.setheading(0)
863 draw.circle(0.8)
864
865 draw.penup()
866 draw.goto(-20, 52)
867 draw.pendown()
868 draw.setheading(0)
869 draw.circle(0.8)
870
871 draw.penup()
872 draw.goto(-20, 52)
873 draw.pendown()
874 draw.setheading(0)
875 draw.circle(0.8)
876
877 draw.penup()
878 draw.goto(-20, 52)
879 draw.pendown()
880 draw.setheading(0)
881 draw.circle(0.8)
882
883 draw.penup()
884 draw.goto(-20, 52)
885 draw.pendown()
886 draw.setheading(0)
887 draw.circle(0.8)
888
889 draw.penup()
890 draw.goto(-20, 52)
891 draw.pendown()
892 draw.setheading(0)
893 draw.circle(0.8)
894
895 draw.penup()
896 draw.goto(-20, 52)
897 draw.pendown()
898 draw.setheading(0)
899 draw.circle(0.8)
900
901 draw.penup()
902 draw.goto(-20, 52)
903 draw.pendown()
904 draw.setheading(0)
905 draw.circle(0.8)
906
907 draw.penup()
908 draw.goto(-20, 52)
909 draw.pendown()
910 draw.setheading(0)
911 draw.circle(0.8)
912
913 draw.penup()
914 draw.goto(-20, 52)
915 draw.pendown()
916 draw.setheading(0)
917 draw.circle(0.8)
918
919 draw.penup()
920 draw.goto(-20, 52)
921 draw.pendown()
922 draw.setheading(0)
923 draw.circle(0.8)
924
925 draw.penup()
926 draw.goto(-20, 52)
927 draw.pendown()
928 draw.setheading(0)
929 draw.circle(0.8)
930
931 draw.penup()
932 draw.goto(-20, 52)
933 draw.pendown()
934 draw.setheading(0)
935 draw.circle(0.8)
936
937 draw.penup()
938 draw.goto(-20, 52)
939 draw.pendown()
940 draw.setheading(0)
941 draw.circle(0.8)
942
943 draw.penup()
944 draw.goto(-20, 52)
945 draw.pendown()
946 draw.setheading(0)
947 draw.circle(0.8)
948
949 draw.penup()
950 draw.goto(-20, 52)
951 draw.pendown()
952 draw.setheading(0)
953 draw.circle(0.8)
954
955 draw.penup()
956 draw.goto(-20, 52)
957 draw.pendown()
958 draw.setheading(0)
959 draw.circle(0.8)
960
961 draw.penup()
962 draw.goto(-20, 52)
963 draw.pendown()
964 draw.setheading(0)
965 draw.circle(0.8)
966
967 draw.penup()
968 draw.goto(-20, 52)
969 draw.pendown()
970 draw.setheading(0)
971 draw.circle(0.8)
972
973 draw.penup()
974 draw.goto(-20, 52)
975 draw.pendown()
976 draw.setheading(0)
977 draw.circle(0.8)
978
979 draw.penup()
980 draw.goto(-20, 52)
981 draw.pendown()
982 draw.setheading(0)
983 draw.circle(0.8)
984
985 draw.penup()
986 draw.goto(-20, 52)
987 draw.pendown()
988 draw.setheading(0)
989 draw.circle(0.8)
990
991 draw.penup()
992 draw.goto(-20, 52)
993 draw.pendown()
994 draw.setheading(0)
995 draw.circle(0.8)
996
997 draw.penup()
998 draw.goto(-20, 52)
999 draw.pendown()
1000 draw.setheading(0)
1001 draw.circle(0.8)
1002

```

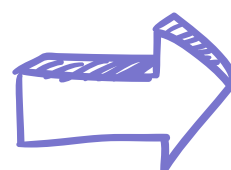





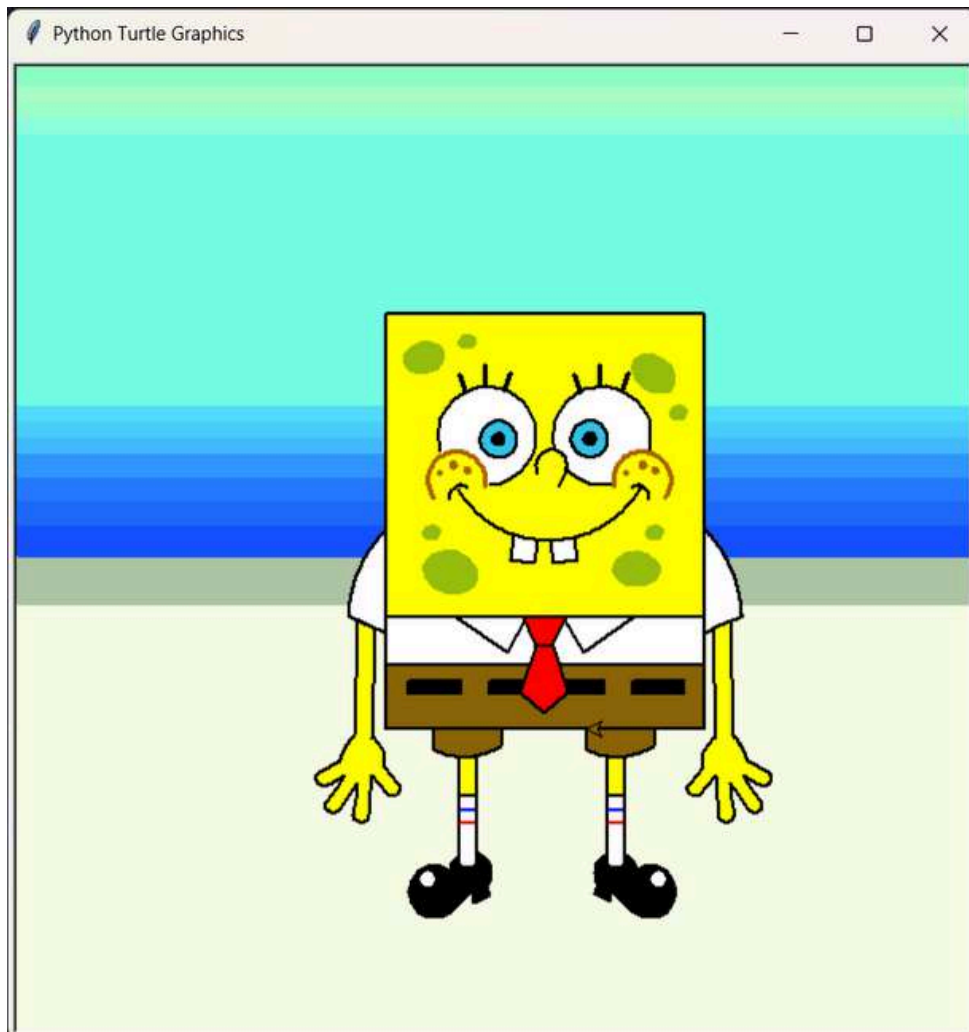
The image shows a screenshot of a Visual Studio Code editor with two Python files open, both named 'spongebob.py'. The left file contains code for drawing the head and body of a spongebob character, while the right file contains code for drawing the pants and legs. The code uses the turtle module for graphics.

```
1133 for i in range(23):
1134     draw.left(2.9)
1135     draw.forward(0.5)
1136 draw.setheading(90)
1137 draw.forward(43)
1138 draw.left(90)
1139 draw.forward(11)
1140 draw.end_fill()
1141
1142 draw.penup()
1143 draw.goto(69, -161)
1144 draw.pendown()
1145 draw.color("blue")
1146 draw.setheading(0)
1147 draw.forward(9)
1148
1149 draw.penup()
1150 draw.goto(69, -169)
1151 draw.pendown()
1152 draw.color("red")
1153 draw.setheading(0)
1154 draw.forward(9)
1155
1156 # Brown Pant Left
1157
1158 draw.penup()
1159 draw.goto(-40, -110)
1160 draw.pendown()
1161 draw.color("black", "#80507")
1162 draw.begin_fill()
1163
1164 draw.setheading(270)
1165 draw.forward(11)
1166 draw.left(60)
1167
```

```
1168 for i in range(23):
1169     draw.left(2.9)
1170     draw.forward(2)
1171
1172 draw.setheading(90)
1173 draw.forward(9)
1174 draw.left(90)
1175 draw.forward(43)
1176 draw.end_fill()
1177
1178 # Brown Pant Right
1179
1180 draw.penup()
1181 draw.goto(36, -110)
1182 draw.pendown()
1183 draw.color("black", "#80507")
1184 draw.begin_fill()
1185
1186 draw.setheading(270)
1187 draw.forward(11)
1188 draw.left(60)
1189
1190 for i in range(23):
1191     draw.left(2.9)
1192     draw.forward(2)
1193
1194 draw.setheading(90)
1195 draw.forward(9)
1196 draw.left(90)
1197 draw.forward(43)
1198 draw.end_fill()
1199
1200 turtle.done()
```



OUTPUT



This is the image of “Spongebob Squarepants” you get after running our code.

EXPLANATION

The forward(x) function in Turtle draws a line of x pixels in the direction the turtle is facing. The concepts of Bresenham's line drawing and circle drawing algorithms are used in this project. It also accommodates Beizer curve algorithm, Scan line algorithm and Flood fill Algorithm.



1. BRESENHAM'S LINE DRAWING ALGORITHMS

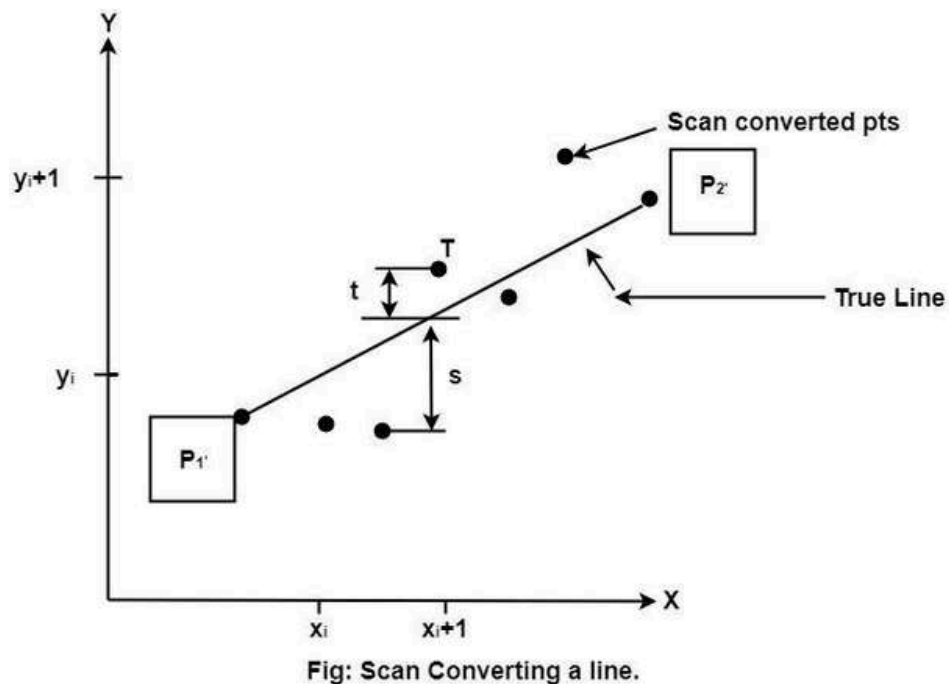
This technique, pioneered by Bresenham, is utilized for scan converting a line.

It's notably efficient as it relies solely on integer addition, subtraction, and multiplication operations, which can be swiftly executed.

The algorithm selects the next pixel based on its proximity to the true line, ensuring a quick generation of lines.

Here's how it works:

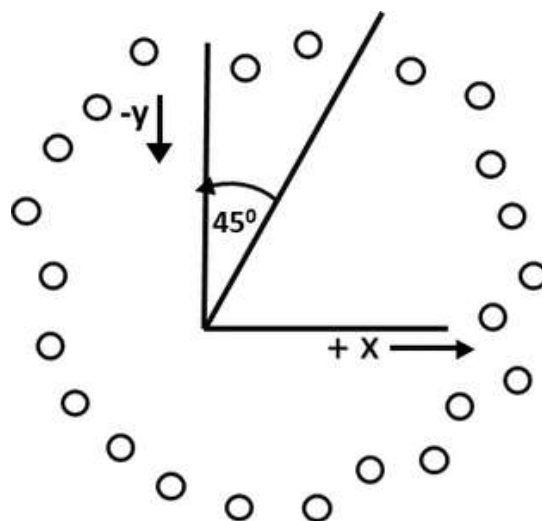
- Starting from a pixel $P1'(x1',y1')$, subsequent pixels are chosen one at a time in the horizontal direction towards $P2'(x2',y2')$.
- At each step, the next pixel is either the one to its right (lower-bound for the line) or the one diagonally up and to its right (upper-bound for the line).



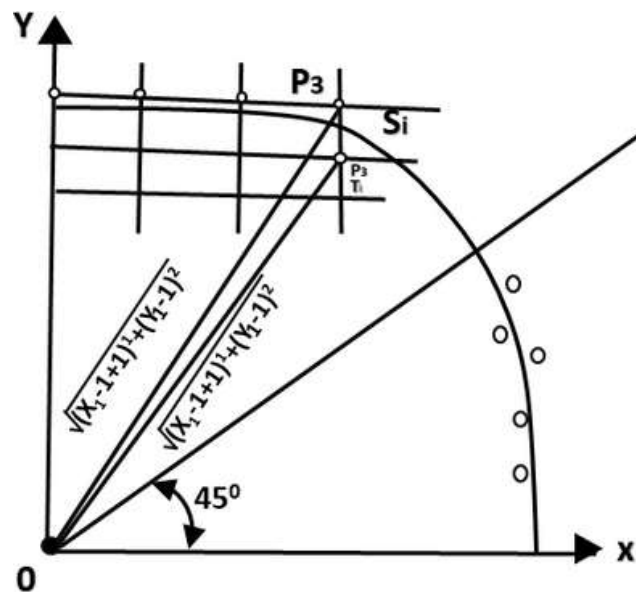
The line is accurately represented by pixels that minimize the distance from the path between $P1'$ and $P2'$.

2. BRESENHAM'S CIRCLE DRAWING ALGORITHMS

Bresenham's algorithm for scan-converting a circle operates in the following manner: Points are generated from 90° to 45° , with movements restricted to the $+x$ and $-y$ directions, as depicted in the figure:-



The most accurate representation of the true circle is achieved by selecting pixels in the raster that are closest to the true circle. Our goal is to generate points from 90° to 45° . Let's assume that the last scan-converted pixel is labeled P_1 , as illustrated in the figure:-

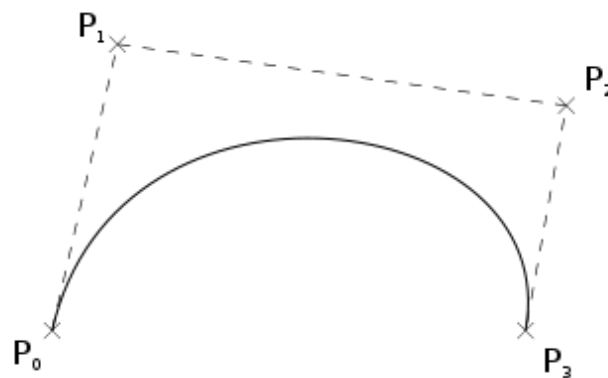


To identify each new point closest to the true circle, you have two options:

- Advance one unit in the x-direction.
- Advance one unit in the x-direction and simultaneously move one unit in the negative y-direction.

3. BEIZER CURVE ALGORITHM

Bézier curves, a fundamental concept in computer graphics, find extensive application in various fields due to their versatility and smoothness. In the context of Turtle graphics, Bézier curves offer a powerful tool for creating complex and visually appealing shapes. Utilizing control points to define the curve's path, the algorithm efficiently interpolates between these points, providing local control over the curve's shape. This makes it ideal for generating smooth and precise curves in Turtle graphics, allowing for the creation of intricate designs and patterns with ease. Whether used for drawing intricate shapes or designing elaborate patterns, the Bézier curve algorithm, with its efficiency and accuracy, enhances the capabilities of Turtle graphics, enabling the creation of captivating visual compositions.

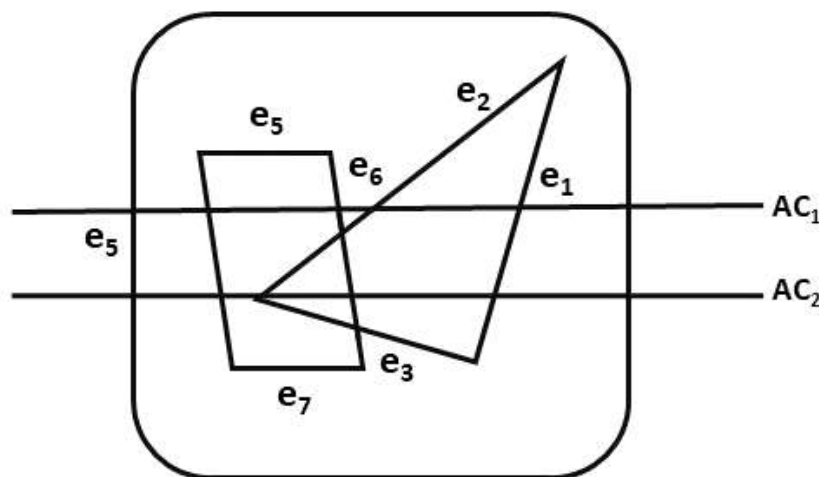


As shown in the diagram , the curve P0 to P3 is drawn by applying Beizer curve algorithm for points P0, P1, P2, P3.

4. SCAN LINE ALGORITHM

The Scan Line Algorithm operates on image space, processing one line at a time rather than pixel by pixel. It utilizes the concept of area coherence, managing edge and active edge lists for accurate rendering. The edge list records endpoint coordinates, while the Active Edge List (AEL) tracks edges intersecting a scan line, sorted by increasing x-values. This dynamic AEL grows and shrinks as needed during processing.

During each scan line iteration, the algorithm determines surface visibility based on intersecting edges, calculating depth for each surface. Visible surfaces receive intensity values stored in the refresh buffer.



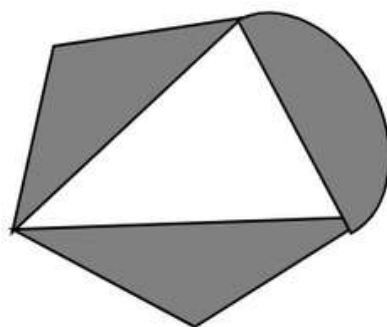
Following figures shown edges and active edge list. The active edge list for scan line AC_1 contain e_1, e_2, e_5, e_6 edges. The active edge list for scan line AC_2 contain e_5, e_6, e_1 .

Algorithm Steps:

1. Initialize data structures: Polygon table with color and edge pointers, Edge table with endpoint and polygon information, and sorted Active Edge List (AEL).
2. Begin scanning each line:
 - Populate the AEL sorted by y-values.
 - Scan until the flag F is active, using a background color.
 - Assign intensity values to visible surfaces based on depth.
 - Utilize coherence for remaining surfaces.
3. End the algorithm.

5.FLOOD FILL ALGORITHM

The flood fill algorithm begins by selecting a seed point inside the region to be filled. This point is then used as the starting point for filling the interior with a specified color. It employs either a 4-connected or 8-connected approach to traverse pixel positions, replacing existing colors with the desired color until the entire interior is filled. This method is particularly useful for filling regions with multiple boundary colors, providing a more efficient alternative to the boundary fill algorithm.



Algorithm:

```
Procedure floodfill (x, y, fill_color, old_color: integer)
  If (getpixel (x, y)=old_color)
  {
    setpixel (x, y, fill_color);
    fill (x+1, y, fill_color, old_color);
    fill (x-1, y, fill_color, old_color);
    fill (x, y+1, fill_color, old_color);
    fill (x, y-1, fill_color, old_color);
  }
}
```

This algorithm is used for the flood fill algorithm to fill the colors in the figure.

CONCLUSION

Through the practical application of Computer Graphics principles in this project, we obtained a deeper understanding of how these concepts are utilized in real-world scenarios. Leveraging the Turtle library, a versatile Python tool, allowed us to create intricate and accurate drawings. By exploring various drawing techniques and algorithms, we were able to achieve detailed and precise visual representations. This hands-on experience provided valuable insights into the practical implementation of Computer Graphics concepts, enhancing our understanding of their significance in graphical applications.



REFERENCES

1. https://en.wikipedia.org/wiki/Bresenham's_line_algorithm
2. <https://docs.python.org/3/library/turtle.html>
3. <https://www.geeksforgeeks.org/bresenhams-line-generation-algorithm/>
4. <https://www.geeksforgeeks.org/turtle-programming-python/>

