**ECE 270**



Justin Newman

Quiz #15

SVG Drawing

November 19, 2014

# Statement of the Problem

The purpose of this program is to store the various parameters of shapes to be drawn in structures to be set by user input and passed to functions that will draw to an output file of the .SVG format.

# Description of solution

Several structures are defined that contain the parameters for a given shape to be drawn; for a line it would be start and end coordinates, for a rectangle it would be a starting x and y coordinate and the length and width for the shape.

Rectangle structure:

**struct** Rectangle{

**float** xCorner; //x-coord of upper-left corner of rectangle

**float** yCorner; //y-coord of upper-left corner of rectangle

**float** width; //horizontal size of rectangle

**float** height; //vertical size of rectangle

};

Two functions are set up for each type of shape, one to get user input for each shape and returns a structure to the main draw function which then will pass it to the second function, which will draw that type of shape.

The fill and stroke structures are global variables and can be called and set from any function.

# Output and Testing

This program was simple to test, but not so simple to debug. Testing amounted to verifying that the .SVG file was correctly produced and that the console output was properly recorded.

**Console output:**

Welcome to Justin Newman's ECE270 Quiz #15 SVG drawing program

Please make your selection from the following menu:

L: Draw a line

C: Draw a circle

R: Draw a rectangle

P: Define your own 1-10 pointed shape

Q: To stop drawing

Please enter your selection now:

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired amount of green for your shape's fill color (0-255): 0

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired opacity of shape's fill color (0.0-1.0): 1.000000

Please enter the desired amount of red for your shape's stroke color (0-255): 0

Please enter the desired amount of green for your shape's stroke color (0-255): 0

Please enter the desired amount of blue for your shape's stroke color (0-255): 0

Please enter the desired opacity of shape's stroke color (0.0-1.0): 1.000000

Please enter the desired width of shape's stroke: 1

Please enter the center x-coordinate of your circle: 500.00

Please enter the center y-coordinate of your circle: 500.00

Please enter the radius of your circle: 110.00

Welcome to Justin Newman's ECE270 Quiz #15 SVG drawing program

Please make your selection from the following menu:

L: Draw a line

C: Draw a circle

R: Draw a rectangle

P: Define your own 1-10 pointed shape

Q: To stop drawing

Please enter your selection now:

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired amount of green for your shape's fill color (0-255): 0

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired opacity of shape's fill color (0.0-1.0): 1.000000

Please enter the desired amount of red for your shape's stroke color (0-255): 0

Please enter the desired amount of green for your shape's stroke color (0-255): 0

Please enter the desired amount of blue for your shape's stroke color (0-255): 0

Please enter the desired opacity of shape's stroke color (0.0-1.0): 1.000000

Please enter the desired width of shape's stroke: 1

Please enter the center x-coordinate of your circle: 400.00

Please enter the center y-coordinate of your circle: 400.00

Please enter the radius of your circle: 60.00

Welcome to Justin Newman's ECE270 Quiz #15 SVG drawing program

Please make your selection from the following menu:

L: Draw a line

C: Draw a circle

R: Draw a rectangle

P: Define your own 1-10 pointed shape

Q: To stop drawing

Please enter your selection now:

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired amount of green for your shape's fill color (0-255): 0

Please enter the desired amount of red for your shape's fill color (0-255): 0

Please enter the desired opacity of shape's fill color (0.0-1.0): 1.000000

Please enter the desired amount of red for your shape's stroke color (0-255): 0

Please enter the desired amount of green for your shape's stroke color (0-255): 0

Please enter the desired amount of blue for your shape's stroke color (0-255): 0

Please enter the desired opacity of shape's stroke color (0.0-1.0): 1.000000

Please enter the desired width of shape's stroke: 1

Please enter the center x-coordinate of your circle: 600.00

Please enter the center y-coordinate of your circle: 400.00

Please enter the radius of your circle: 60.00

Welcome to Justin Newman's ECE270 Quiz #15 SVG drawing program

Please make your selection from the following menu:

L: Draw a line

C: Draw a circle

R: Draw a rectangle

P: Define your own 1-10 pointed shape

Q: To stop drawing

Please enter your selection now:

Program terminating...

enjoy your drawing

Your drawing can be found in image.svg in the same folder as this program

**Image produced:**

# C:\Users\Der Meister\Desktop\image.png

# Code

1 #include<stdio.h>

2 #include<stdlib.h>

3

4 //START snippet of SVG code on canvas

5 //Global Variables ............................

6 FILE \*fp; //File pointer for svg image

7 FILE \*output;

8

9 **char** fileName[] = "image.svg";

10

11 **int** imWidth = 1000; //image width (total)

12 **int** imHeight = 1000; //image height (total)

13 //..............................................

14 //END snippet of SVG code from canvas

15

16 **struct** Line{

17 **float** x1; //x-coord of line start

18 **float** y1; //y-coord of line start

19 **float** x2; //x-coord of line end

20 **float** y2; //y-coord of line end

21 };

22

23 **struct** Circle{

24 **float** cx; //center x-coordinate

25 **float** cy; //center y-coordinate

26 **float** r; //radius

27 };

28

29 **struct** Rectangle{

30 **float** xCorner; //x-coord of upper-left corner of rectangle

31 **float** yCorner; //y-coord of upper-left corner of rectangle

32 **float** width; //horizontal size of rectangle

33 **float** height; //vertical size of rectangle

34 };

35

36 **struct** Path{

37 **int** n; //number of points in path

38 **float** x[10]; //array of x-coordinates

39 **float** y[10]; //array of y-coordinates

40 };

41

42 **struct** Fill{

43 **int** red; //set red tone 0-255

44 **int** green; //set green tone 0-255

45 **int** blue; //set blue tone 0-255

46 **float** opacity; //set opacity 0.0-1.0

47 };

48

49 **struct** Stroke{

50 **int** red; //set red tone 0-255

51 **int** green; //set green tone 0-255

52 **int** blue; //set blue tone 0-255

53 **float** opacity; //set opacity 0.0-1.0

54 **int** width; //set width

55 };

56

57 //I'll just declare these down here so the program actually knows what's up

58 //Replaces stroke and fill element declarations from canvas code

59 **struct** Fill **fill**;

60 **struct** Stroke stroke;

61

62 //Prototypes for functions from canvas SVG code

63 **float** rand\_float();

64 **void** setFill();

65 **void** setStroke();

66 **void** drawLine(**struct** Line );

67 **void** drawCircle(**struct** Circle );

68 **void** drawRectangle(**struct** Rectangle );

69 **void** drawPath(**struct** Path );

70 **void** writeSVGHeader(**char** [], **int** , **int** );

71 **void** writeSVGFooter();

72 **void** draw();

73 **void** menu();

74 **struct** Line setLine();

75 **struct** Circle setCircle();

76 **struct** Rectangle setRectangle();

77 **struct** Path setPath();

78

79 /\* MAIN \*/

80

81 **int** main()

82 {

83 srand(time(NULL));

84 output=fopen("quiz15.txt","w");

85

86 writeSVGHeader(fileName, imWidth, imHeight);

87

88 menu();

89

90 draw();

91

92 writeSVGFooter();

93

94 fclose(output);

95

96 **return** 0;

97 }

98

99 /\* RAND FLOAT \*/

100

101 **float** rand\_float()

102 {

103 **return**((**float**) rand() / (**float**) RAND\_MAX);

104 }

105

106 /\* SET FILL \*/

107

108 **void** setFill()

109 {

110 //fill.red = red\_in;

111 printf("\nPlease enter the desired amount of red for your shape's fill color (0-255)");

112 scanf("\n%d",&**fill**.red);

113 fprintf(output,"\nPlease enter the desired amount of red for your shape's fill color (0-255): %d",**fill**.red);

114 //fill.green = green\_in;

115 printf("\nPlease enter the desired amount of green for your shape's fill color (0-255)");

116 scanf("\n%d",&**fill**.green);

117 fprintf(output,"\nPlease enter the desired amount of green for your shape's fill color (0-255): %d",**fill**.green);

118 //fill.blue = blue\_in;

119 printf("\nPlease enter the desired amount of blue for your shape's fill color (0-255)");

120 scanf("\n%d",&**fill**.blue);

121 fprintf(output,"\nPlease enter the desired amount of red for your shape's fill color (0-255): %d",**fill**.blue);

122 //fill.opacity = opacity\_in;

123 printf("\nPlease enter the desired opacity of your shape's fill color (0.0-1.0)");

124 scanf("\n%f",&**fill**.opacity);

125 fprintf(output,"\nPlease enter the desired opacity of shape's fill color (0.0-1.0): %f",**fill**.opacity);

126 }

127

128 /\* SET STROKE \*/

129

130 **void** setStroke()

131 {

132 //stroke.red = red\_in;

133 printf("\nPlease enter the desired amount of red for your shape's stroke color (0-255)");

134 scanf("%d",&stroke.red);

135 fprintf(output,"\nPlease enter the desired amount of red for your shape's stroke color (0-255): %d",stroke.red);

136 //stroke.green = green\_in;

137 printf("\nPlease enter the desired amount of green for your shape's stroke color (0-255)");

138 scanf("%d",&stroke.green);

139 fprintf(output,"\nPlease enter the desired amount of green for your shape's stroke color (0-255): %d",stroke.green);

140 //stroke.blue = blue\_in;

141 printf("\nPlease enter the desired amount of blue for your shape's stroke color (0-255)");

142 scanf("%d",&stroke.blue);

143 fprintf(output,"\nPlease enter the desired amount of blue for your shape's stroke color (0-255): %d",stroke.blue);

144 //stroke.opacity = opacity\_in;

145 printf("\nPlease enter the desired opacity of your shape's stroke color (0.0-1.0)");

146 scanf("%f",&stroke.opacity);

147 fprintf(output,"\nPlease enter the desired opacity of shape's stroke color (0.0-1.0): %f",stroke.opacity);

148 //stroke.width = width\_in;

149 printf("\nPlease enter the desired width of your shape's stroke");

150 scanf("%d",&stroke.width);

151 fprintf(output,"\nPlease enter the desired width of shape's stroke: %d",stroke.width);

152 }

153

154 /\* DRAW LINE \*/

155

156 **void** drawLine(**struct** Line line)

157 {

158 fprintf(fp,"\n <line x1 = '%f' y1 = '%f' x2 = '%f' y2 = '%f'", line.x1, line.y1, line.x2, line.y2);

159 fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'", stroke.red, stroke.green, stroke.blue, stroke.opacity);

160 fprintf(fp," stroke-width = '%d' />", stroke.width);

161 }

162

163

164 /\* DRAW CIRCLE \*/

165

166 **void** drawCircle(**struct** Circle circle)

167 {

168 fprintf(fp,"\n <circle cx = '%f' cy = '%f' r = '%f'", circle.cx, circle.cy, circle.r);

169 fprintf(fp," fill = 'rgb(%d, %d, %d)' fill-opacity = '%f'", **fill**.red, **fill**.green, **fill**.blue, **fill**.opacity);

170 fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'", stroke.red, stroke.green, stroke.blue, stroke.opacity);

171 fprintf(fp," stroke-width = '%d' />", stroke.width);

172 }

173

174 /\* DRAW RECTANGLE \*/

175

176 **void** drawRectangle(**struct** Rectangle rectangle)

177 {

178 fprintf(fp,"\n <rect x = '%f' y = '%f' width = '%f' height = '%f'", rectangle.xCorner, rectangle.yCorner, rectangle.width, rectangle.height);

179 fprintf(fp," fill = 'rgb(%d, %d, %d)' fill-opacity = '%f'", **fill**.red, **fill**.green, **fill**.blue, **fill**.opacity);

180 fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'", stroke.red, stroke.green, stroke.blue, stroke.opacity);

181 fprintf(fp," stroke-width = '%d' />", stroke.width);

182 }

183

184 /\* DRAW PATH \*/

185

186 **void** drawPath(**struct** Path path)

187 {

188 **int** i;

189

190 fprintf(fp,"\n <path d ='M%f %f ", path.x[0], path.y[0]);

191

192 **for**(i=1;i<path.n;i++)

193 {

194 fprintf(fp,"L%f %f ", path.x[i], path.y[i]);

195 }

196

197 fprintf(fp,"z' fill = 'rgb(%d, %d, %d)' fill-opacity = '%f'", **fill**.red, **fill**.green, **fill**.blue, **fill**.opacity);

198 fprintf(fp," stroke = 'rgb(%d, %d, %d)' stroke-opacity = '%f'", stroke.red, stroke.green, stroke.blue, stroke.opacity);

199 fprintf(fp," stroke-width = '%d' />", stroke.width);

200 }

201

202 /\* WRITE SVG HEADER \*/

203

204 **void** writeSVGHeader(**char** fileName[], **int** width, **int** height)

205 {

206 //Open file for writing

207 fp = fopen(fileName,"w");

208

209 //Setup SVG header

210 fprintf(fp,"<?xml version='1.0' standalone='no'?>");

211 fprintf(fp,"\n<svg xmlns='http://www.w3.org/2000/svg' xmlns:xlink='http://www.w3.org/1999/xlink' version='1.1' width = '%d' height = '%d'>", width, height);

212 }

213

214 /\* WRITE SVG FOOTER \*/

215

216 **void** writeSVGFooter()

217 {

218 //Closing SVG tag and close file

219 fprintf(fp,"\n</svg>");

220 fclose(fp);

221 }

222

223 /\* DRAW \*/

224

225 **void** draw()

226 {

227 **struct** Line line;

228 **struct** Circle circle;

229 **struct** Rectangle rect;

230 **struct** Path path;

231 **char** selection;

232

233 **do**

234 {

235 scanf("\n%c",&selection);

236

237 **switch**(selection)

238 {

239 **case** 'L':**case** 'l':

240 system("cls");

241 setStroke();

242 line=setLine();

243 drawLine(line);

244 menu();

245 **break**;

246 **case** 'C':**case** 'c':

247 system("cls");

248 setFill();

249 setStroke();

250 circle=setCircle();

251 drawCircle(circle);

252 menu();

253 **break**;

254 **case** 'R':**case** 'r':

255 system("cls");

256 setFill();

257 setStroke();

258 rect=setRectangle();

259 drawRectangle(rect);

260 menu();

261 **break**;

262 **case** 'P':**case** 'p':

263 system("cls");

264 setFill();

265 setStroke();

266 path=setPath();

267 drawPath(path);

268 menu();

269 **break**;

270 **case** 'Q':**case** 'q':

271 system("cls");

272 printf("\nProgram terminating...\nenjoy your drawing");

273 fprintf(output,"\nProgram terminating...\nenjoy your drawing");

274

275 printf("\nYour drawing can be found in image.svg in the same folder as this program");

276 fprintf(output,"\nYour drawing can be found in image.svg in the same folder as this program");

277 **break**;

278 **default** :

279 system("cls");

280 printf("\nPlease make a valid selection from the following menu:\n");

281 fprintf(output,"\nPlease make a valid selection from the following menu:\n");

282 menu();

283 }

284 }**while**(selection!='Q'&&selection!='q');

285 }

286

287 //Displays function selection menu

288 **void** menu()

289 {

290 printf("\nWelcome to Justin Newman's ECE270 Quiz #15 SVG drawing program");

291 fprintf(output,"\nWelcome to Justin Newman's ECE270 Quiz #15 SVG drawing program");

292

293 printf("\n\nPlease make your selection from the following menu:");

294 fprintf(output,"\n\nPlease make your selection from the following menu:");

295

296 printf("\n\nL:\tDraw a line");

297 fprintf(output,"\n\nL:\tDraw a line");

298

299 printf("\nC:\tDraw a circle");

300 fprintf(output,"\nC:\tDraw a circle");

301

302 printf("\nR:\tDraw a rectangle");

303 fprintf(output,"\nR:\tDraw a rectangle");

304

305 printf("\nP:\tDefine your own 1-10 pointed shape");

306 fprintf(output,"\nP:\tDefine your own 1-10 pointed shape");

307

308 printf("\nQ:\tTo stop drawing");

309 fprintf(output,"\nQ:\tTo stop drawing");

310

311 printf("\n\nPlease enter your selection now:");

312 fprintf(output,"\n\nPlease enter your selection now:");

313 }

314

315 //allows the user to set the paramters of a line to be drawn

316 **struct** Line setLine()

317 {

318 **struct** Line line;

319

320 printf("\nPlease enter the starting x-coordinate of your line");

321 scanf("\n%f",&line.x1);

322 fprintf(output,"\nPlease enter the starting x-coordinate of your line: %.2f",line.x1);

323

324 printf("\nPlease enter the starting y-coordinate of your line");

325 scanf("\n%f",&line.y1);

326 fprintf(output,"\nPlease enter the starting y-coordinate of your line: %.2f",line.y1);

327

328 printf("\nPlease enter the ending x-coordinate of your line");

329 scanf("\n%f",&line.x2);

330 fprintf(output,"\nPlease enter the ending x-coordinate of your line: %.2f",line.x2);

331

332 printf("\nPlease enter the ending y-coordinate of your line");

333 scanf("\n%f",&line.y2);

334 fprintf(output,"\nPlease enter the ending y-coordinate of your line: %.2f",line.y2);

335

336 system("cls");

337

338 **return**(line);

339 };

340

341 //allows the user to set the paramters of a circle to be drawn

342 **struct** Circle setCircle()

343 {

344 **struct** Circle circle;

345 printf("\nPlease enter the center x-coordinate of your circle");

346 scanf("\n%f",&circle.cx);

347 fprintf(output,"\nPlease enter the center x-coordinate of your circle: %.2f",circle.cx);

348

349 printf("\nPlease enter the center y-coordinate of your circle");

350 scanf("\n%f",&circle.cy);

351 fprintf(output,"\nPlease enter the center y-coordinate of your circle: %.2f",circle.cy);

352

353 printf("\nPlease enter the radius of your circle");

354 scanf("\n%f",&circle.r);

355 fprintf(output,"\nPlease enter the radius of your circle: %.2f",circle.r);

356

357 system("cls");

358

359 **return**(circle);

360 };

361

362 //allows the user to set the paramters of a rectangle to be drawn

363 **struct** Rectangle setRectangle()

364 {

365 **struct** Rectangle rect;

366

367 printf("\nPlease enter the x-coordinate of the upper left corner of your rectangle");

368 scanf("\n%f",&rect.xCorner);

369 fprintf(output,"\nPlease enter the x-coordinate of the upper left corner of your rectangle%.2f",rect.xCorner);

370

371 printf("\nPlease enter the y-coordinate of the upper left corner of your rectangle");

372 scanf("\n%f",&rect.yCorner);

373 fprintf(output,"\nPlease enter the y-coordinate of the upper left corner of your rectangle%.2f",rect.yCorner);

374

375 printf("\nPlease enter the width of your rectangle");

376 scanf("\n%f",&rect.width);

377 fprintf(output,"\nPlease enter the width of your rectangle%.2f",rect.width);

378

379 printf("\nPlease enter the height of your rectangle");

380 scanf("\n%f",&rect.height);

381 fprintf(output,"\nPlease enter the height of your rectangle%.2f",rect.height);

382

383 system("cls");

384

385 **return**(rect);

386 };

387

388 //allows the user to set the paramters of a path to be drawn

389 **struct** Path setPath()

390 {

391 **struct** Path path;

392 **int** i,n;

393

394 printf("\nPlease enter the number of corners your shape will have (1-10)");

395 scanf("\n%d",&n);

396 fprintf(output,"\nPlease enter the number of corners your shape will have: %d",n);

397

398 **float** x[n];

399 **float** y[n];

400

401 **for**(i=0;i<n;i++)

402 {

403 printf("\nPlease enter the x-coordinate for point #%d of your shape",i);

404 scanf("\n%f",x+i);

405 fprintf(output,"\nPlease enter the center x-coordinate of your line: %.2f",x[i]);

406

407 printf("\nPlease enter the y-coordinate for point #%d of your shape",i);

408 scanf("\n%f",y+i);

409 fprintf(output,"\nPlease enter the center y-coordinate of your line: %.2f",y[i]);

410 }

411

412 path.n=n;

413

414 **for**(i=0;i<n;i++)

415 {

416 path.x[i]=x[i];

417 path.y[i]=y[i];

418 }

419

420 system("cls");

421

422 **return**(path);

423 };