2800 Biscayne Blvd
Suite 600
Miami, FL 33137

neocis

Software Assessment

Part 1:

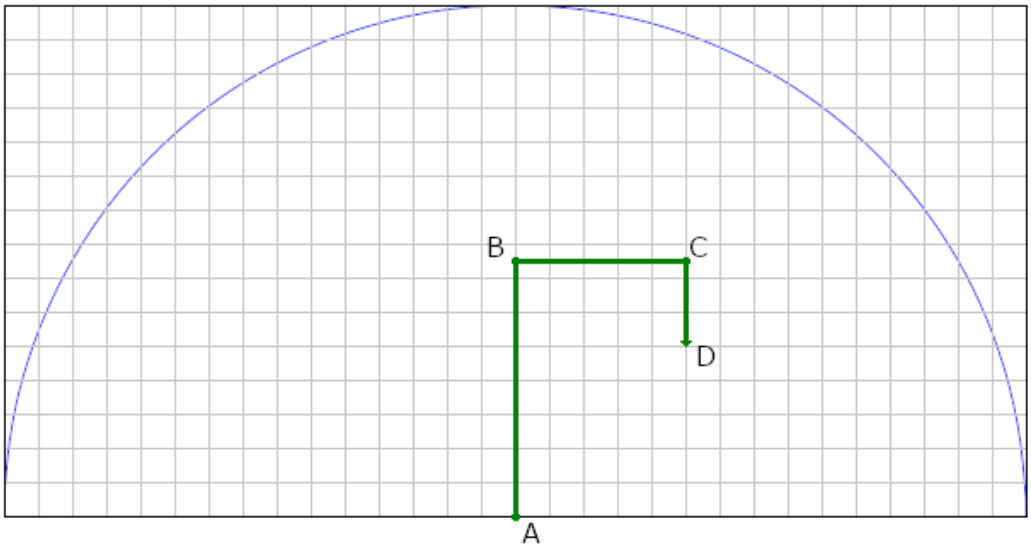
Design a program that simulates a three-segment robotic arm in 2 dimensions.

Create a single window that contains a grid that goes from -300 (left) to +300 (right) on the X-axis and 0 (bottom) to +300 (top) on the Y-axis. Place a gray gridline every 20 units both vertically and horizontally. Draw a blue circle with radius=300 and center at the origin (0,0).

Now add a "robotic arm" that consists of three connected line segments with these rules:

1. The first line segment (A->B) is 150 units long and should always have point A at the origin (0,0). The angle between the segment and the X-axis can vary such that point B can move in an arc.
2. The second (middle) line segment (B->C) is 100 units long and should always have the end at point B connected to the end of the first line segment. The angle between the first and second segments can be any value as long as point B is 100 units from point C.
3. The third line segment (C->D) is 50 units and should always have the end at point C connected to the end of the second line segment. The angle between the second and third segments can vary.

Your grid and arm should look like this:



Add functionality to accept multiple mouse clicks on the grid. Each time the user clicks a point on the grid within the blue circle, find angles of each of the joints of the arm (at points A, B & C) such that Point D is at the point clicked. Draw the arm using those angles. For example, the diagram above shows a solution when the user clicks the point (100,100). Your grid should look like the diagram, except that the letters (A,B,C&D) are not needed. The letters are for explanation in this document.

2800 Biscayne Blvd
Suite 600
Miami, FL 33137

Now modify the robot arm to move Point D from its current position to the new position in a straight line. To accomplish this, when the user clicks a point, draw a line between the current point D and the clicked point. Draw 21 equally spaced dots on the line, with the first dot at the current Point D and the final dot at the clicked location. The Point D of the arm should move from the current position to the clicked position by moving to each of the dots in order along the line to the final point.

Calculate the 21 solutions for the joint angles of the arm segments that place Point D at each of the 21 points along the line. Draw the arm at each of the 21 poses (segment angles) in order. Display the arm at each of the 21 points for at least ¼ of a second before erasing it to draw the next arm pose. There are many joint angle solutions for each Point D, so **choose the solution in which the maximum change of all three joint angles is as small as possible for each step (i.e. minimize "Max Δ ∡" from the table below.)** *Do the math for this yourself! Do not use an optimization function or library.*

Below the grid, print a table with the data for each dot and set of poses (see below). The table should contain the following columns:

1. The number of the dot, from 1 to 21. (Dot #)
2. The (X,Y) coordinate of point B, which is the end of the 1st line segment. (B pos)
3. The angle between the 1st line segment and the Y-axis in degrees. "0" indicates that the segment is vertical, negative values indicate that it is to the left of the Y-axis, and positive values indicate it is to the right of the Y-axis. (A-B ∡)
4. Absolute value of the difference between "A-B ∡" of the current and previous step. (Δ A-B ∡)
5. The (X,Y) coordinate of point C, which is the end of the 2nd (middle) line segment. (C pos)
6. The angle between the 1st and 2nd line segments. "0" indicates that the segments have the same orientation (i.e. the 2nd segment linearly extends the 1st segment with no bend at the joint), negative values indicate a bend to the left (counter-clockwise), and positive values indicate a bend to the right (clockwise). (B-C ∡)
7. Absolute value of the difference between "B-C ∡" of the current and previous step. (Δ B-C ∡)
8. The (X,Y) coordinate of point D, which is the end of the 3rd (last) line segment. (D pos)
9. The angle between the 2nd and 3rd line segments. See "B-C ∡" above for how to measure the angle. (C-D ∡)
10. Absolute value of the difference between "C-D ∡" of the current and previous step. (Δ C-D ∡)
11. The greatest of "Δ A-B ∡", "Δ B-C ∡", and "Δ C-D ∡" on this step. **This is the value to minimize on each step.** (Max Δ ∡)

See the table below for an example of the columns described.

2800 Biscayne Blvd
Suite 600
Miami, FL 33137

Dot #	B pos	A-B ∡	Δ A-B ∡	C pos	B-C ∡	Δ B-C ∡	D pos	C-D ∡	Δ C-D ∡	Max Δ ∡
1	(0,150)	0	---	(100,150)	90	---	(100,100)	90	---	---
2	(-7.55,149.81)	-2.885	2.885	(92.45,149.94)	92.81	2.81	(90,100)	92.883	2.883	2.885
3	(-15.203,149.228)	-5.817	2.932	(84.796,149.769)	95.507	2.696	(80,100)	95.815	2.931	2.932
4	(-22.966,148.231)	-8.807	2.99	(77.026,149.504)	98.078	2.571	(70,100)	98.807	2.992	2.992
5	(-30.851,146.793)	-11.869	3.062	(69.121,149.161)	100.512	2.434	(60,100)	101.867	3.06	3.062
6	(-38.858,144.879)	-15.014	3.145	(61.066,148.76)	102.79	2.278	(50,100)	105.011	3.144	3.145
7	(-46.985,142.452)	-18.254	3.24	(52.843,148.322)	104.888	2.098	(40,100)	108.25	3.238	3.24
8	(-55.216,139.467)	-21.599	3.345	(44.43,147.873)	106.778	1.889	(30,100)	111.595	3.346	3.346
9	(-63.533,135.881)	-25.059	3.46	(35.797,147.439)	108.422	1.644	(20,100)	115.055	3.459	3.46
10	(-71.891,131.65)	-28.638	3.579	(26.916,147.052)	109.778	1.356	(10,100)	118.634	3.579	3.579
11	(-80.23,126.74)	-32.335	3.697	(17.749,146.744)	110.796	1.018	(0,100)	122.331	3.697	3.697
12	(-88.462,121.138)	-36.139	3.804	(8.256,146.548)	111.419	0.623	(-10,100)	126.135	3.804	3.804
13	(-96.474,114.86)	-40.028	3.889	(-1.61,146.495)	111.585	0.166	(-20,100)	130.022	3.887	3.889
14	(-104.131,107.966)	-43.964	3.936	(-11.899,146.608)	111.232	0.353	(-30,100)	133.957	3.934	3.936
15	(-111.286,100.576)	-47.894	3.93	(-22.661,146.897)	110.299	0.933	(-40,100)	137.885	3.928	3.93
16	(-117.799,92.862)	-51.751	3.857	(-33.95,147.354)	108.732	1.567	(-50,100)	141.742	3.857	3.857
17	(-123.561,85.045)	-55.461	3.71	(-45.823,147.948)	106.482	2.249	(-60,100)	145.45	3.708	3.71
18	(-128.502,77.377)	-58.946	3.485	(-58.326,148.618)	103.514	2.968	(-70,100)	148.933	3.483	3.485
19	(-132.485,70.34)	-62.035	3.089	(-70.965,149.177)	100.001	3.513	(-80,100)	152.444	3.51	3.513
20	(-135.512,64.317)	-64.61	2.575	(-83.185,149.533)	96.162	3.84	(-90,100)	156.282	3.839	3.84
21	(-137.757,59.356)	-66.69	2.08	(-94.987,149.748)	92.012	4.15	(-100,100)	160.433	4.15	4.15

Extra Credit:

- Constrain your angle solutions so that all parts of the arm (segments) are completely within the grid.
- Add an option to minimize the largest value in the last column of the table above rather than minimizing it on each step.

General Guidelines for both parts:

Your solution should be easily executable, and well documented. Document all algorithms in full.

Ways to Improve your Chances:

- Good code design & structure
- An object-oriented solution
- Good documentation!

Requirements for submission

- Turn your result in by the end of the second day. Earlier if possible.
- Please use any standard development language (e.g. C/C++, Java, Python, JavaScript or equivalent). LabView, MATLAB or equivalent are not considered to be standard development languages.
- Attach all documentation and source code in an email to **jtieman@neocis.com**.
- Ideally, include a working Windows executable, executable jar file or equivalent of your solution. If that is not practical, include all needed libraries and instructions for properly compiling and executing your application on Windows.