//
// The following is the documentation for XXX coding assessment project and how to compile the codes
// Presented by Chun-Yin Huang
// Jun 5th, 2020
//
// Library Used: OpenCV 4.1.0
// Additional Source Code: CVUI (https://github.com/Dovyski/cvui/releases)
//

/******************** How to Compile ********************/

* In these projects, I utilized an open source, cvui, which computes GUI system using only OpenCV library.
* The IDE I used was Xcode from MacOS, but I have tested that the following compile message could successfully build the program on a Unix system if OpenCV library is installed correctly.
* For Windows, a properly OpenCV linked VS project with the attached 'cvui.h' file in the same directory with 'main.cpp' should work fine.

Compile message on Unix system:

```
g++ main.cpp -o project -I/usr/local/include/opencv4/opencv -
I/usr/local/include/opencv4 -L/usr/local/lib -lopencv_gapi -lopencv_stitching -
lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -
lopencv_dnn_objdetect -lopencv_dnn_superres -lopencv_dpm -lopencv_highgui -
lopencv_face -lopencv_freetype -lopencv_fuzzy -lopencv_hfs -lopencv_img_hash -
lopencv_intensity_transform -lopencv_line_descriptor -lopencv_quality -
lopencv_rapid -lopencv_reg -lopencv_rgbd -lopencv_saliency -lopencv_sfm -
lopencv_stereo -lopencv_structured_light -lopencv_phase_unwrapping -
lopencv_superres -lopencv_optflow -lopencv_surface_matching -lopencv_tracking -
lopencv_datasets -lopencv_text -lopencv_dnn -lopencv_plot -lopencv_videostab -
lopencv_videoio -lopencv_xfeatures2d -lopencv_shape -lopencv_ml -
lopencv_ximgproc -lopencv_video -lopencv_xobjdetect -lopencv_objdetect -
lopencv_calib3d -lopencv_imgcodecs -lopencv_features2d -lopencv_flann -
lopencv_xphoto -lopencv_photo -lopencv_imgproc -lopencv_core
```

/******************** PART I *********************/

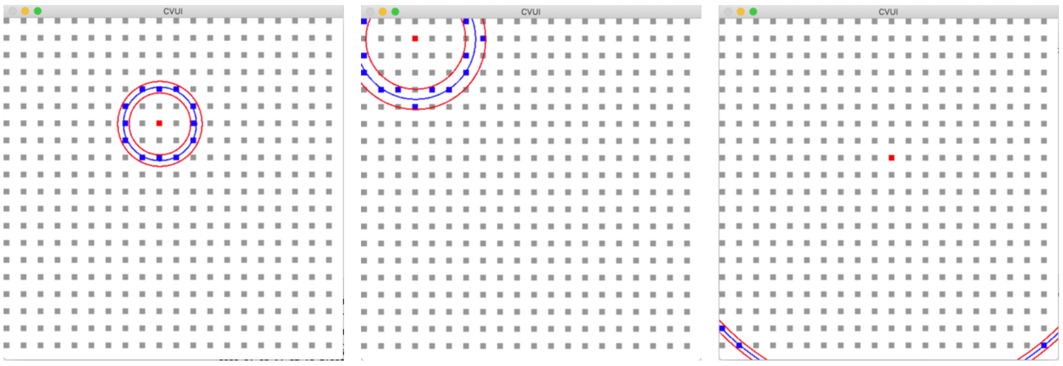
//---------- Design Flow

1. Parameters: point size: 9*9, patch size: 27*27, image size: 540*540
2. Draw the initial image
3. Acquire the starting point and the ending point by detecting left-click down and up, respectively
4. Use a src_ref matrix to represent the selected points and find the center and the radius.
5. Plot any point that is close enough to the circumference.
6. Reset the program if right-click is detected.

//---------- How to Use

1. Run the program
2. Left-click on a gray point to indicate the desired center, hold
3. Release left-click on a gray point to indicate the desired circumference point
4. Right-click on any point to reset the system

//---------- Test Cases



/******************** PART II ********************/

*There are two versions, one of them used Object Orientated coding.

//---------- Design Flow

1. Parameters: point size: 9*9, patch size: 27*27, image size: 540*540
2. Draw the initial image
3. Acquire the desired circumference points by detecting left-click up
4. From 3, unmark the point if it has already been selected
5. Find the center by averaging the positions of the selected points
6. Find the radius by calculate the averaged distance from the selected points and

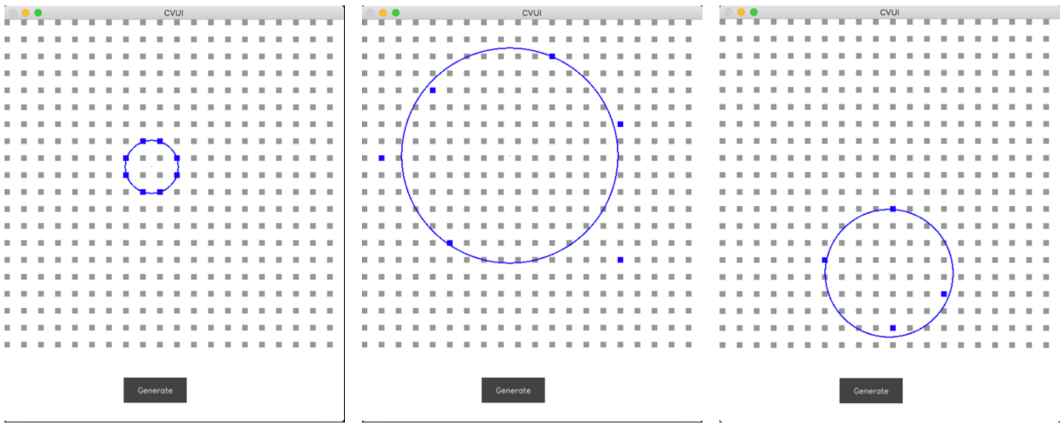
the calculated center

7. Use a src_ref matrix to represent the selected points
8. Plot any point that is close enough to the circumference.
9. Reset the program if right-click is detected.

//---------- How to Use

1. Run the program
2. Left-click on a gray point to indicate the desired circumference point (the selected point will be marked as blue)
3. Left-click on a blue point to delete the desired circumference point (the un-selected point will be marked as gray)
4. Click on the 'generate' button to render the circle
5. From 4, a warning message will pop out if the user press generate button before selecting any point
6. Right-click on any point to reset the system

//---------- Test Cases



******************** PART III ********************/

//---------- Design Flow

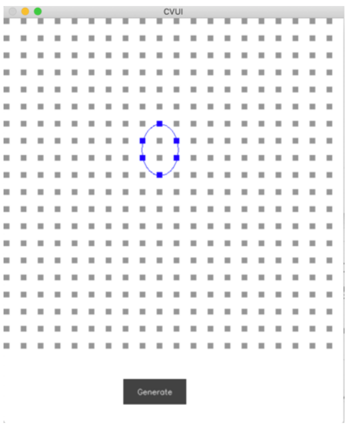
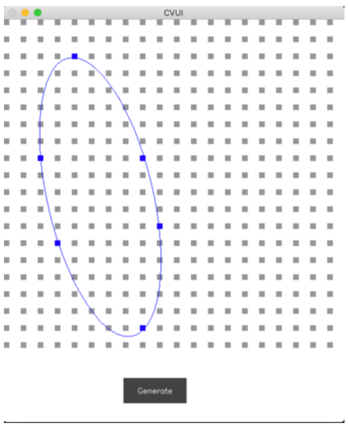
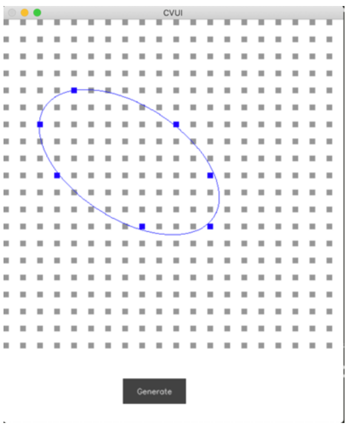
1. Parameters: point size: 9*9, patch size: 27*27, image size: 540(+100)*540
2. Draw the initial image
3. Design a generate button at the bottom of the image
4. Acquire the desired circumference points by detecting left-click up and put them into a vector
5. Use the OpenCV function *fitEllipse* to detect the best fit ellipse
6. From 4, note that to find a best ellipse, the user should select more than 5 points

7. Draw the ellipse on the image using the OpenCV function *ellipse*
8. Reset the program if right-click is detected

//---------- How to Use

1. Run the program
2. Left-click on a gray point to indicate the desired circumference point (the selected point will be marked as blue)
3. Left-click on a blue point to delete the desired circumference point (the un-selected point will be marked as gray)
4. Click on the 'generate' button to render the ellipse
5. From 4, a warning message will pop out if the user press generate button before selecting enough points
6. Right-click on any point to reset the system

//---------- Test Cases


CVUI
Generate

CVUI
Generate

CVUI
Generate