CS3483 Assignment Report

TSOI Yiu Chik, 57225353

# Sections Descriptions

## Global Variables

In this project, multiple global variables were used to cooperate with the draw() function.

A screen shot of a computer

Description automatically generated

Figure 1: Global Variables

Figure 1 shows variables that are storing original image, blurred version of image, real-time camera capture and ml5.js’s hand pose model.

A screen shot of a computer

Description automatically generated

Figure 2: Global Variables

Figure 2 shows variables of image indicator coordinate. “pdisplayx” and “pdisplayy” are storing coordinate before the new index fingertip indicator. “displayx” and “displayy” are for current coordinate of index fingertip. They are initialized with coordinate (0, 0), which is at the origin. An indicatorColor variable is also set, so the indicator is in **green** initially.

A screen shot of a computer program

Description automatically generated

Figure 3: Global Variables

Figure 3 shows variables for hand pose model options, model detection results, index fingertip coordinate and thumb tip coordinate. Both coordinates of index fingertip and thumb tip are respect to video capture’s origin, which is (0, 0) at top-left corner of video source.

A screen shot of a computer

Description automatically generated

Figure 4: Global Variable

There are some variables to cater the requirements, displayed in figure 4. “mode” variable is used for indicating the current mode of interface, 1 for view mode, 2 for freehand drawing, 3 for circle drawing and 0 means there is no active mode. “freeHandImgSet” is used for indicating whether the original image is set in freehand drawing mode, value “true” means the image is set.

“viewBoxSize” is the side length for the square view box in view mode.

“pixelView” is for setting the filling color in circle drawing mode, it is an array arranged in “RGB” sequence.

## preload

A screen shot of a computer

Description automatically generated

Figure 5

The preload function is run before the setup() function. In this function, the hand pose model original images are load into respective variables. Specifically, there is no blurring filter applied to the “blurImg” image as it will be done in later function.

## setup

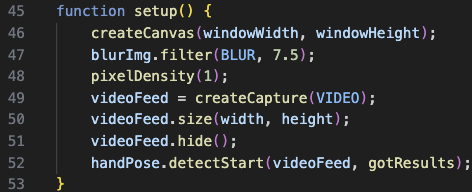


Figure 6

This function is run before the draw() function and after preload() function. It performs the following tasks.

1. Create an empty canvas with user’s window dimensions
2. Apply a blurring filter to the image stored in “blurImg”
3. Set pixel density to 1
4. Capture camera feed in user’s window dimensions
5. Start detecting hands from video feed with the “gotResults” callback

## draw

A screen shot of a computer program

Description automatically generated

Figure 7

As features of interface depend on the current mode, draw() function applies switch cases for each mode. “mode = 1” indicates view mode, draw() function clears the region of displayed image before placing the blurred versio. After placing “blurImg”, a cropped portion of original image is obtained via “cropImage”, which returns a new image with square dimension of “viewBoxSize”. The center of cropped image is equivalent to where the index fingertip indicator was landed in the image region. After cropping, the cropped portion will be placed on the blurred image according to the index fingertip indicator if and only if the hand pose model detects the index fingertip.

For freehand drawing mode, indicated by “mode = 2”, draw() function clears the image region and replace the region to original image in “img” if “freeHandImgSet = false”. Such action will also set “freeHandImgSet = true” to indicate the completion of region reset. The condition is to ensure image reset will only perform once after switching to this mode. After that, stroke detail is set, and a line is draw between previous and current indicator location. As the draw() function executes repeatedly, the trace of index finger indicator is drawn on the image region. An extra condition for drawing the trace is added to ensure the trace will only be drawn when index fingertip is detected.

In circle drawing mode with “mode = 3”, draw() function resets image display region to original image. After reset, a pixel’s color located at index fingertip indicator on image region is obtained via “pixelView = img.get(display, displayy)”. Since the result is in “RGBA” sequence with 4th element representing the opacity, 120 is assigned to “A” value to fulfill the requirement of providing reasonable opacity. Before drawing the circle on image region, a condition is added to ensure both index finger and thumb are detected, thus radius setting can be successfully applied to the circle indicator. The radius of circle is obtained as a portion of distance between index fingertip and thumb tip. After that, a circle is drawn on the image region.

When receiving “e” as input, “mode” is set to 0, and the image region is reset to original image.

A screen shot of a computer program

Description automatically generated

Figure 8

Figure 8 shows the remaining part of draw(). It uses “push()” and “pop()” function to ensure origin translation and element scales only applies to the video capture part but not the whole canvas. It also draws indicators when there is a detected index fingertip. When a key is pressed, it calls the “switchMode()” function to apply mode switching.

## gotResults

A computer screen shot of a program code

Description automatically generated

Figure 9

This function serves as a callback function for handPose.detectStart() in setup(). It saves the detection result (passed to “results” as argument) to global variable “detections”. Additionally, index finger and thumb are saved explicitly to enable convenience access for other function. Such explicit saves are only enabled when a hand is detected, indicated by the condition “detections.length > 0”.

## drawIndicator

A screen shot of a computer program

Description automatically generated

Figure 10

This function serves as procedure of calculating and drawing indicators’ coordinates, used in draw() function. A relative coordinate of index fingertip in video capture is first obtained. After that, the coordinate scales with the image region’s dimension and indicators are drawn on the canvas.

To achieve better user experience, the filling color of indicator is set differently for circle drawing mode and set to green otherwise. In circle drawing mode, the indicator color is as same as the pixel pointed by user.

## switchMode

A screen shot of a computer program

Description automatically generated

Figure 11

This function is called when there is a key press, indicated by the condition of “keyIsPressed” in draw() function. It maps functional keys to mode value and reset “freeHandImgSet” and indicator coordinates.

## cropImage

A screen shot of a computer code

Description automatically generated

Figure 12

This function is used to return a portion of image of “img”, which is an argument of image. Since “x” and “y” is the center of cropped image, the top-left corner of bounding box is calculated. The new cropped portion is returned by “img.get()” function.

# Screen Captures

## Normal mode “mode = 0”

A screenshot of a video game

Description automatically generated

Figure 13: Normal Mode without Indicator

A screenshot of a video game

Description automatically generated

Figure 14: Normal Mode with Indicator

## View mode “mode = 1”

A screenshot of a video game

Description automatically generated

Figure 15: View Mode without Indicator

A screenshot of a video game

Description automatically generated

Figure 16: View Mode with Indicator

## Freehand drawing mode “mode = 2”

A screenshot of a video game

Description automatically generated

Figure 17: Freehand Drawing Mode without Indicator

A screenshot of a video game

Description automatically generated

Figure 18: Freehand Drawing Mode with Indicator and Drawn Trace

## Circle drawing mode “mode = 3”

A screenshot of a video game

Description automatically generated

Figure 19: Circle Drawing Mode without Indicator

A screenshot of a video game

Description automatically generated

Figure 20: Circle Drawing Mode with Indicator and Circle

# Limitation and Possible Improvements

## Freehand Drawing Mode

In this mode, when the index fingertip first becomes undetectable and returned to detection in a new position, an explicit line is drawn on the image region, which is not desirable for user experience. A possible improvement could be an update on the drawing algorithm to prevent such unexpected line.