Coding Two: Advanced Frameworks

U000126 Assignment Element 2: Final Project

Simple calligraphy

General idea: use the mouse to draw a circle, and then draw a line. Change the thickness of the drawn line and make the thickness of the line transition smoothly. Change the thickness difference on a line segment and replace the bifurcation line.

```
def setup():#initialization function, only run once
    size（800,600）#Set the width and height of the screen
    strokeWeight（10）#Set the thickness of the line
    background（255）#Set white background
    maxThickness=25#Set the thickest stroke

def draw(): #Draw function, run repeatedly every frame
    return#The function returns directly

#line (x1, y1, x2, y2), draw a line segment between the vertices (x1, y1), (x2, y2)
Record the position of the last frame of the mouse (lastX, lastY), and draw a line with the current position of the mouse (mouseX, mouseY)

def mousePressed():#Execute when the mouse button is pressed
    global lastX,lastY,vx,vy,lastThickness #Global variables
    lastX=mouse X#When the mouse is pressed, the starting point coordinates of this pen
    lastY=mouseY
    vx=0#The speed of movement is initialized to 0
    vy=0
    lastThickness=1#When the mouse is just pressed, the stroke thickness is 1

def mouseDragged(): #Execute when the mouse button is dragged
    global lastX,lastY,vx,vy,lastThickness #Global variables
    vx=0.7*vx+0.3*(mouseX-lastX)#Get the current movement speed and keep continuous interpolation
    vy=0.7*vy+0.3*(mouseY-lastY)
    v=squrt（vx*vx+vy*vy）#absolute value of current moving speed

#The faster the speed, the finer the strokes
   nextThickness=maxThickness-v
    if nextThickness <0: #Prevent the thickness from being less than 0
      nextThickness =0
   #The thickness of the brushstroke also needs to be continuous to prevent too drastic changes
 nextThickness=0.5* nextThickness+0.5*lastThickness
# Divide the line segment line (lastX, lastY, mouseX, mouseY) into n segments and draw them separately. The faster the mouse moves, the larger n is. Int() converts decimals to integers
```

```
n=10+int(v/2)#The faster the speed, the higher the score
for i in range(1,n+1):#divide the two points before and after the mouse into n segments to draw
    x1=map(i-1,0,n,lastX,mouseX)#corresponding coordinates of the two vertices before and after
    y1=map(i-1,0,n,lastY,mouseY)
    x2=map(i,0,n,lastX,mouseX)
    y2=map(i,0,n,lastY,mouseY)

#Corresponding to the thickness of this short paragraph
  thickness=map(i-1,0,n, lastThickness,nextThickness)
#Define the offset offset=2, draw the main stroke line (x1, y1, x2, y2) with the thickness + offset. Set the stroke thickness to thickness, draw a line line (x1+offset*2, y1+offset*2, x2+offset*2, y2+offset*2) at the bottom right of the main line, and draw a line at the top left of the main line line (x1-offset,y1-offset, x2-offset,y2-offset)
strokeWeight(thickness+offset)#The thickness of the main line plus an offset

#Divide the line segment line (lastX, lastY, mouseX, mouseY) into n segments and draw them separately, and draw the main line
line(x1,y1,x2,y2)#Draw a line
  strokeWeight(thickness)# The offset is drawn below to simulate the effect of brush bifurcation drawing
  line (x1+offset*2,y1+offset*2, x2+offset*2,y2+offset*2)
  line (x1-offset,y1-offset, x2-offset,y2-offset)



  lastX=mouseX #Update the coordinates of the previous point
  lastY=mouseY
  lastThickness=nextThickness#Update the thickness of the previous stroke

def keyPressed():#When any keyboard key is pressed
    background(255)#re-fill the screen with white
```