Week 6 Small Groups Handout

Conceptual Questions

Multiple Choice - MVC

Fill in the bubbles to indicate whether the following components of the Snake animation are part of the Model, View, or Controller.

1) app.foodPosition	◯ Model	◯ View	◯ Controller
2) drawSnake()	◯ Model	◯ View	◯ Controller
3) app.snake	◯ Model	◯ View	◯ Controller
4) drawGameOver()	◯ Model	◯ View	◯ Controller
5) app.margin	◯ Model	◯ View	◯ Controller
6) timerFired()	◯ Model	◯ View	◯ Controller
7) app.direction	◯ Model	◯ View	◯ Controller
8) drawFood()	◯ Model	◯ View	◯ Controller
9) redrawAll()	◯ Model	◯ View	◯ Controller
10) keyPressed()	◯ Model	◯ View	◯ Controller
11) takeStep()	◯ Model	◯ View	◯ Controller
12) placeFood()	◯ Model	◯ View	◯ Controller

Fill in the Blanks - getCell Function

Fill in each blank to complete the getCell(app, x, y) function. You may assume pointInGrid(app, x, y) is already written. You may use the attributes app.width, app.height, app.margin, app.cols, and app.rows in your answer:

```
def appStarted(app):
    app.rows = 4
    app.cols = 8
    app.margin = 5 # margin around grid

def pointInGrid(app, x, y):
    # return True if (x, y) is inside the grid defined by app.
    return ((app.margin <= x <= app.width-app.margin) and
            (app.margin <= y <= app.height-app.margin))

def getCell(app, x, y):
    # aka "viewToModel"
    # return (row, col) in which (x, y) occurred or (-1, -1) if
    # outside the grid.
    if (not ____________________________________):
        return (-1, -1)
    gridWidth  = app.width - ____________________
    gridHeight = app.height - ____________________
    cellWidth  = ____________________________________
    cellHeight = ____________________________________

    row = ____________________________________________
    col = ____________________________________________

    return (row, col)
```

Free Response

Blowing Bubbles

Problem Statement

Write an animation with the following features:

1. A dot (bubble) with radius 20 begins at the center of the canvas.
2. Clicking anywhere in the canvas will create another circle (bubble) with radius 20 centered at that location.
3. Every 100ms, each circle (bubble) moves down by 5 pixels.
4. Dots disappear when they touch the bottom edge of the canvas.
5. Pressing 'p' will stop the dots from falling, but more dots can still be created by clicking the canvas.
6. Pressing 'p' again will cause the circles to resume falling.

Connect4

Create the board game Connect4, where:

- We have a board with 7 columns and 6 rows.
- Pressing inside of a column drops a square peg inside of that column (should fall to the lowest position possible - no need to animate it falling, the peg just appears in the lowest position)
- If the column is full, ignore the mouse press
- After a peg has been placed, the game should change players (color of the peg)
- If a player has 4 pegs in a direction, then the game is over and all further mouse clicks are ignored. (the wordSearch code will be given in the starter file)
- Pressing “r” will restart the game

Code Tracing

[CT] Code Tracing 1 - **OOP**

```
from dataclasses import make_dataclass

def ct1():
    A = make_dataclass('A', ['x', 's'])
    M = [ A(x=3,s='a'),
          A(x=1,s='bc'),
          A(x=4,s='de'),
          A(x=2,s='f') ]
    b = A(x=0, s='')
    for a in M:
        if a.x > b.x:
            b.x += a.x
        else:
            b.s += a.s[0]
    return [b.x, b.s]
print(ct1())
```

Solution

[CT] Code Tracing 2 - Bonus CT

```
def ct2(L):
    M = sorted(L)
    N = L
    M.append(L.pop(0))
    N.append(M.pop(0))
    N = N[:2]
    return M, N
L = [3,1,2]

M, N = ct2(L)
print(L, M, N)
```

Solution