

fullName:_____ andrewID:_____ section:_____

15-112 S25
Quiz4 version A

Read these instructions carefully before starting:

1. Quiz versions are color-coded. You must have a different version (color) of this quiz than the students sitting to your left and right.
2. Stop writing and submit the entire quiz when instructed by the proctor.
 - Do not unstaple any pages.
 - You must submit the entire quiz with all pages intact.
3. Do not discuss the quiz with anyone else until after 5pm.
 - This applies to everyone, including students in either lecture.
4. Do not use your own scrap paper.
 - You should not need scrap paper, there is plenty of room for you on the quiz.
 - However, if you absolutely must use scrap paper, raise your hand and we will provide some. Then, you must write your andrew id clearly on the scrap paper, and hand in the scrap paper with your paper quiz. We will not grade anything on your scrap paper.
5. You may not ask questions during the quiz.
 - The one exception is for English-language clarifications.
 - If you are unsure how to interpret a problem, just take your best guess.
6. Do not use any concepts (including built-in functions) not covered in the notes through week 4 or beyond unit 3.
 - Do not use lists, tuples, dictionaries, sets, or recursion.
7. Do not hardcode your solutions.
 - We may test your code using additional test cases.
 - Hardcoding will receive zero points.
8. Assume `almostEqual(x, y)` and `rounded(n)` are both supplied for you.
 - You must write all other helper functions you wish to use, unless we specify otherwise.
9. Good luck!

Note on function headers: For both FR1 and FR2, we have provided headers for necessary `cmu_graphics` functions. We recommend that you use these, though you may write additional helper functions if you wish.

Free Response / FR1: Moving Color-Changing Dot [40 pts]

Write an animation such that:

1. At first, a red dot of radius 100 is drawn centered on the window. (Note that this must be true even if the window is not 400x400.)
2. When the user presses the right arrow key, the dot moves 30 pixels to the right. If this would make any part of the dot extend beyond the right edge of the window, then the dot is instead positioned so that its left edge is flush with the right edge of the window.
3. Each time the user presses any key besides the right arrow, the color of the circle toggles between red and blue.

Hint: It may be helpful to sketch the app in different states before you begin

You must follow MVC rules. MVC violations carry large deductions.

Begin your FR1 answer here.

```
from cmu_graphics import *  
  
def onAppStart(app):
```

#Continue your answer to FR1 here:

```
def onPress(app, key):
```

```
def redrawAll(app):
```

```
runApp()
```

Free Response / FR2: Dot with Lines [60 pts]

Draw the app as described in the accompanying video, which we will play in a loop with subtitles (but without audio) throughout this quiz.

Features:

- Set `app.stepsPerSecond` to 10.
- When the app starts:
 - A red dot of radius 10 is drawn at `cx = 100`, `cy = 100`.
- When the 'p' key is pressed:
 - The dot's color is set to blue.
 - A line gradually grows from the center of the dot to the right at a constant rate.
 - After 1 second, this horizontal line is 50 pixels long and stops growing. After the horizontal line finishes growing, a vertical line grows from the center of the dot downwards.
 - After 2 seconds, this vertical line is 50 pixels long and stops growing.
 - If the 'p' key is pressed again at any time, any existing lines disappear and the line-drawing process restarts. (So, the horizontal line begins growing from the center to the right again).
 - The dot should always be drawn on top of any lines.
- When the mouse is pressed:
 - The dot moves its center to where the mouse was pressed.
 - The dot's color is set to red.
 - Any existing lines disappear

Notes:

- **You must follow MVC rules.** MVC violations carry large deductions.
- You may not import or use the `time` module, or any similar modules. As usual, you may not use anything we have not yet covered in the notes.
- You do not need to set any unspecified details (such as the color of the line or the `lineWidth`)

Hint: We recommend keeping your model simple. Consider creating a variable called `app.steps`, which keeps track of the number of times `onStep` has been called since 'p' was last pressed. Then, all you need in order to determine the length of each line is `app.steps` and `app.stepsPerSecond`.

Thus, we calculate the endpoints of each line inside of `redrawAll` before they are drawn, and without storing them in the model. (But be sure to obey MVC).

Partial credit: If you are unsure how to animate the lines, remember that you can get partial credit by correctly implementing other features. For example, make the dot move and change color properly.

In case it helps, an exact transcript of the video is given below:

“To begin with, you should not draw this black line or anything to the right of it. This is not part of your animation.

You begin with a canvas of an unknown size and a red dot of radius 10. That dot is located 100 pixels from the left edge of the canvas and 100 pixels from the top edge of the canvas.

When I press the mouse the dot moves to the location of that mouse press, and when I press the 'p' key the dot turns blue and a line draws to the right and then down. Let's watch that again. We draw to the right and then down gradually, and it takes 1 second to draw the rightward line, and then once that line is done it takes another second to draw the vertical downward line.

If I press the mouse again you'll notice that the lines go away and we return to red. Pressing 'p' begins drawing those lines again, always starting at the center of the dot.

If I press the 'p' key while the lines are actively growing, that process just restarts and the lines begin fresh. Likewise if I press the mouse while the lines are growing then the lines will still go away and the dot will still move and become red. That's it! Good luck.”

Begin your FR2 answer on the next page.

#Begin your answer to FR2 here:

```
from cmu_graphics import *
```

```
def onStart(app):
```

```
def onMousePress(app, mouseX, mouseY):
```

#Continue your answer to FR2 here:

```
def onKeyPress(app, key):
```

```
def onStep(app):
```

#Continue your answer to FR2 here:

```
def redrawAll(app):
```

```
runApp()
```

(No bonusCTs this week)