

fullName: _____ andrewID: _____ section: _____

15-112 F25 Quiz2 FR2 Retake
Time: 20 Minutes

Note: this page must be filled out before you submit the quiz.

Which version of Q2 do you want to be graded (you must check only ONE of these):

☐ Grade THIS version.

or

☐ Grade TUESDAY'S version.

Either way, we will also grade the bonus question on this version should you complete it.

You **must write your name on this paper and hand this back** in immediately after the assessment. If we do not receive it immediately, you will receive a zero on the assessment. **Do not unstaple any pages.** All pages must be handed in intact.

Do not use your own scrap paper. You should not need it, but if you must absolutely have scrap paper, raise your hand and we will provide some. Write your andrewID clearly on it and hand it in with your quiz. We will not grade anything on scrap paper.

You may not ask questions during the quiz, except for English-language clarification questions. If you are unsure about a problem, take your best guess.

Before and during the quiz, you may not view any other notes, prior work, websites or resources, including any form of AI. You may not use calculators, phones, laptops, or any other devices. You may not communicate with anyone else except for current 112 TAs or faculty during the assessment. All syllabus policies apply.

You may not discuss this test with anyone else, even briefly, in any form, until we have released grades. Failure to abide by these rules may result in an academic integrity violation.

Do not use sets, dictionaries, recursion, or anything else disallowed in the original problem.

Do not open this or look inside (even briefly) before you are ready to begin. Do not spend more than 20 minutes on this assessment.

Important notes for the Free Response question:

- You can use the following function without writing it:

```
def getRadiusEndpoint(cx, cy, r, theta):  
    return (cx + r*math.cos(math.radians(theta)),  
            cy - r*math.sin(math.radians(theta)))
```

- Do not assume the canvas is 400x400, but you can assume it is square and at least 400x400.
- You are also given a video which shows how your app should run, but that video is not required. This writeup provides all the details you need.

Quiz2 FR2-Retake [50 pts]

Part 1: drawBasicWheel

Write the helper function `drawBasicWheel(cx, cy, r, hour)` that draws a gold circle of the given center and radius, and then draws a black dot of radius 5 at the given hour in normal clock time except we will use 0 instead of 12 (so 0 is up, 3 is to the right, 6 is down, and 9 is to the left). You can assume the hour is between 0 and 11 inclusive.

Part 2: Moving Basic Wheels

Note: For this part, assume your `drawBasicWheel` function from part1 works properly. Your `redrawAll` should call that function twice. With that, create an animation such that:

- At first, a horizontal line is drawn across the entire canvas centered vertically. Two basic wheels are drawn, one centered in the top half above the line, the other centered in the bottom half below the line. Both wheels have a radius of 75 and start with a start hour of 0.
- The hours on each wheel change by 1 on every 5th call to `onStep`. The hour on the top changes clockwise. The hour on the bottom changes counter-clockwise.
- The top wheel moves to the right by 5 pixels on each call to `onStep`. When the entire top wheel is beyond the right edge, it wraps around so it starts to enter from the left edge.
- The bottom wheel bounces up and down. On each call to `onStep`, its `cy` changes by 5 pixels. If this would make any part of the wheel extend beyond the center line or beyond the bottom of the canvas, the wheel reverses direction (bounces). Note that when the wheel bounces, it is ok if a tiny portion of the wheel briefly extends just beyond the boundary.

Again, a video of this animation will be displayed during the quiz.

Write your solution on the following pages.





Bonus / Optional [2 pts]

Draw a sketch of what the following app will draw:

```
from cmu_graphics import *  
import math
```

```
def onAppStart(app):  
    app.setMaxShapeCount(10000)
```

```
def f(x): return x//33%2
```

```
def redrawAll(app):  
    for a in range(0, 99, 33):  
        for b in range(0, 99, 33):  
            for c in range(a, a+33):  
                for d in range(b, b+33):  
                    drawCircle(c, d, 1,  
                               fill=[None, 'black'][f(a) != f(b)])
```

```
def main():  
    runApp()
```

```
main()
```

