SI@UCF Computer Science Camp
Object Oriented Python and pyGame
Quiz #2 Solutions Date: 6/25/2025

1) (15 pts) The first version of the fruit game showed used inheritance. In particular, the class pictoken inherited from the class token. Here is a listing of the methods in each of the classes. Five of these methods were unnecessary for the correct operation of this version of the fruit game. Please list which method from which class was not needed in the space provided below, along with the reason it was not needed.

```
class token:
    def __init__(self,myx,myy,mydx,mydy,width,height,mycolor)
    def move(self)
    def hit(self, mypos)
    def bounceLeft(self)
    def bounceRight(self, SCREEN_W)
    def bounceUp(self)
    def bounceDown(self, SCREEN_H)
    def draw(self, DISPLAYSURF)

class pictoken(token):
    def __init__(self,myx,myy,mydx,mydy,mycolor,mypic,mypts)
    def hit(self, mypos)
    def draw(self, DISPLAYSURF)
```

Method/Class Not Needed	Reason
1. bounceLeft (token class)	The fruits never bounce off the left side of the screen. So fruit game never calls this method.
2. bounceRight (token class)	The fruits never bounce off the right side of the screen. So fruit game never calls this method.
3. bounceUp (token class)	The fruits never bounce off the top of the screen. So fruit game never calls this method.
4. bounceDown (token class)	The fruits never bounce off the bottom of the screen. So fruit game never calls this method.
5. hit (pictoken)	The hit method that pictoken inherits from Token would work correctly, so there's no Reason to override this method. Instead the Original method from token should be used.
6. draw (token)	There are no regular token objects that get drawn. The draw in pictoken is valid on its own and runs correctly w/o token's draw.

Grading: 1 pt for each method/class, 2 pts for each reason

2) (20 pts) Pygame provides a method to check if a point is within an axis-aligned rectangle. The logic for this method isn't beyond our understanding. (It can be implemented with basic Python tools.) Write a function that takes in a rectangle (left, top, width, height) and a point (x, y), and returns true if the point is contained in the rectangle and false otherwise. (Return true if the point is on the boundary of the rectangle.)

```
# rect[0] is the topleft x value, rect[1] is the topleft y value
# rect[2] is the width, rect[3] is the height
# pos[0] is the x value, pos[1] the y value of the point
def inrect(rect, pos):

    if pos[0] < rect[0] or pos[0] >= rect[0]+rect[2]:
        return false

    if pos[1] < rect[1] or pos[1] >= rect[1] + rect[3]:
        return false

    return true
```

Grading: 10 pts for code solving problem in x, 10 pts for code solving problem in y
Grader decides partial

3) (10 pts) Let the width and height of a pyGame surface be SCREEN_W and SCREEN_H, respectively. If you want to draw a rectangle with width, w, and height, h, perfectly centered on the surface (so the left and right borders are the same width and the top and bottom borders are the same height), what would be the x and y coordinates of the top left corner of the rectangle, in terms of SCREEN_W, SCREEN_H, w and h? Please use only integer operations.

```
topleft_X = (SCREEN_W-w)//2
```

```
topleft_Y = (SCREEN_H-h)//2
```

Grading: 5 pts for each, 1 pt off on each for real number division

4) (20 pts) The following code below is the entire code from updatescores.py from the second version of the Fruit Game. Answer the questions after the code about it. Several lines are numbered to aid in the questions.

```
MAXSCORES = 10
def main():
    scores = open("highscores.txt", "r")
    n = int(scores.readline())                    # line 4
    best = []
    for i in range(n):
        toks = scores.readline().split()
        item = []
        item.append(int(toks[1]))                 # line 9
        item.append(toks[0])
        best.append(item)
    scores.close()
    newname = input("What is your name?\n")
    newscore = int(input("What was your score on the game?\n"))
    newitem = [newscore, newname]
    best.append(newitem)
    best.sort(reverse=True)                       # line 17
    newscores = open("highscores.txt", "w")
    cnt = min(MAXSCORES, len(best))               # line 19
    newscores.write(str(cnt)+"\n")                # line 20
    for i in range(cnt):
        newscores.write(best[i][1]+"\t"+str(best[i][0])+"\n")
    newscores.close()
main()
```

(a) (3 pts) What is the largest number that could be read in for n on line 4?

10 **(3 pts all or nothing)**

(b) (7 pts) Why do we append int(toks[1]) on line 9 instead of toks[0]?

We need to sort the entries by score, and Python sorts lists by the first item in the list. Since we want to sort by score first, we have to put this first in our list, item, so that this entry, score followed by name, is sorted by score and not name, when we call the sort function on line 17.

(c) (3 pts) Why do we put reverse=True on line 17?

We want to print the scores to the file from largest to smallest, which is reverse numerical order. We must specify this flag so that Python's sort function sorts from largest to smallest.

(d) (4 pts) Why do we use the min function on line 19?

We don't want to store any more than MAXSCORES (10) scores in the bestscores file, highscores.txt. This minimum function ensures that cnt doesn't get set to more than 10.

(e) (3 pts) Why can't we put a comma between cnt and "\n" on line 20?

The write function doesn't take multiple parameters. It only takes one string. If we were to put a comma, we would be passing the write function multiple strings, which isn't allowed.

Grading for parts b, c and d is up to grader discretion

5) (5 pts) What does the following segment of code do to the image person:

```
curW = 2*person.get_width())
curH = 2*person.get_height())
person = pygame.transform.scale(person, (curW, curH))
```

This doubles the size of both the width and height of the image, which can be thought of as "making the image twice as big", but really, the total area of the image (in pixels) will be 4 times the previous area.

6) (5 pts) Why is the method shown in question 5 from pygame.transform useful in making games?

Many images you might want to use in a game that you download from online might be bigger in their original file than you want them to display on the pygame screen. This method, scale, allows one to get the image that they want and create a similar one of the desired size.

7) (10 pts) In the minesweeper game shown in class, a timer printed how many seconds in the game had elapsed since the player started. How was this timer implemented? Provide as many details as possible, including what data was stored, when, what type that data was and how it was used to display the time elapsed?

The event that "starts" the game is when the user clicks on a valid square to uncover it. When that mouseButtonDown event occurs for the first time, a variable, called startT gets set equal to time.time(), which is the number of seconds after a set point in the past. (Usually January 1, 1970.) Then, every time the game loop runs, time.time() – startT is calculated. This represents the current time minus when the game started. Although both of these numbers are big, since they are relative to the same point in the past, their difference represents how long the game has been running. This new value is a double, but is then converted to an integer via the int function. Then, this integer value is rendered to the screen, every time the game loop runs.

8) (10 pts) In the sushi game, there are two classes that inherit from pygame.Sprite.sprite. These classes are Obj and Player. In particular, these two classes have different update functions. Why do they have different update functions and what do each of the two update functions do differently?

The Obj class is used for both nigiri objects and detos objects. These objects only move down and just fall off the screen. The Player object, however, wraps around the screen if it goes too far off the right. Also, the Player object can only move left and right, it can't move up or down at all. Thus, two classes were needed for two different update methods, since each of these types of objects moves differently.

Grading for 5, 6, 7 and 8 is up to grader discretion

9) (5 pts) What animal appears on the Panda Express logo?

Panda **(Grading: Give to all)**