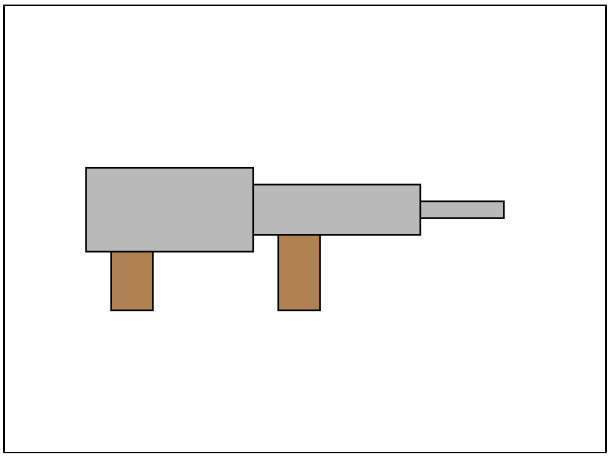
Project6

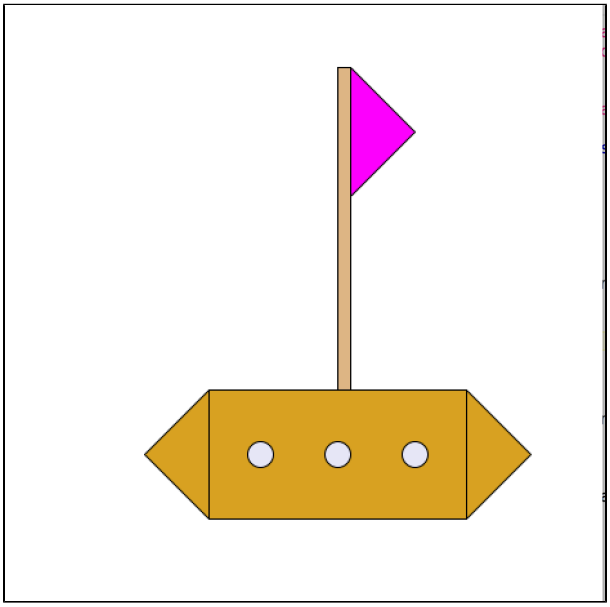
In this project, we were asked to create a collection of complex shapes and objects, very similar to the tasks from project 3. The big difference between project 3 and this project, project 6, is that instead of Python Turtle Graphics, we have started using a new package called, Python Zelle Graphics. With Zelle Graphics, we can all the same stuff from turtle but more efficiently and with more functions. For example, Zelle graphics has shape functions built into the program so to a code a rectangle one can write graphics.Rectangle(point(x,y), point(x,y)). Also, with this new Zelle Graphics package, we learned how to animate objects. The goal of this project was to create a scene out of complex objects and animate an aspect of the scene.

Task 1 asked me to code a few functions for complex shapes using the new Zelle Graphics function. We were told to name these init functions with the word init, i.e. "function_init(x,y,scale)." And, the parameters that each init function takes is x location, y location, and scale. I decided to create objects, two complex objects and one simple background object function. My two complex objects were a cannon and a ship. To create these objects I needed to used rectangles, triangles(polygons), and circles. Here is a snippet of code from my ship function:

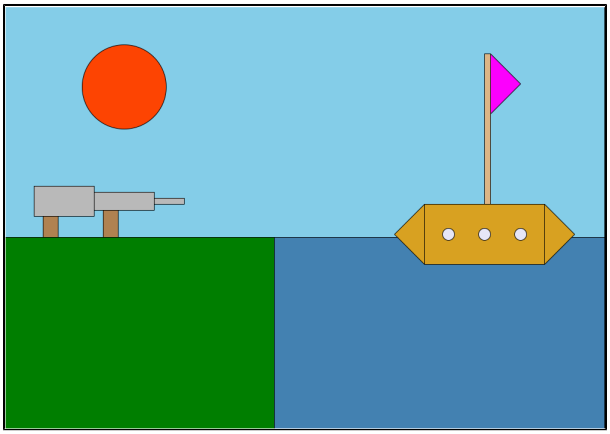
```
def ship_init( x, y, scale ):
    """ Create the list of objects needed to draw a ship
        at position (x,y) with the given scale """

    shapes = []

    '''separate rectangles triangles for the
        body of the ship, mast, and flag and
        circle for the windows'''

    body = gr.Rectangle( gr.Point(x,y), gr.Point(x+scale*200, y+scale*100) )
    body.setFill(gr.color_rgb( 218, 165, 32))
    shapes.append( body )

    mast = gr.Rectangle( gr.Point(x+scale*100, y-250*scale), gr.Point(x+scale*110, y) )
    mast.setFill(gr.color_rgb( 222, 184, 135))
    shapes.append( mast )
```

Here is what my two complex images looked like:





And all together with the background function:



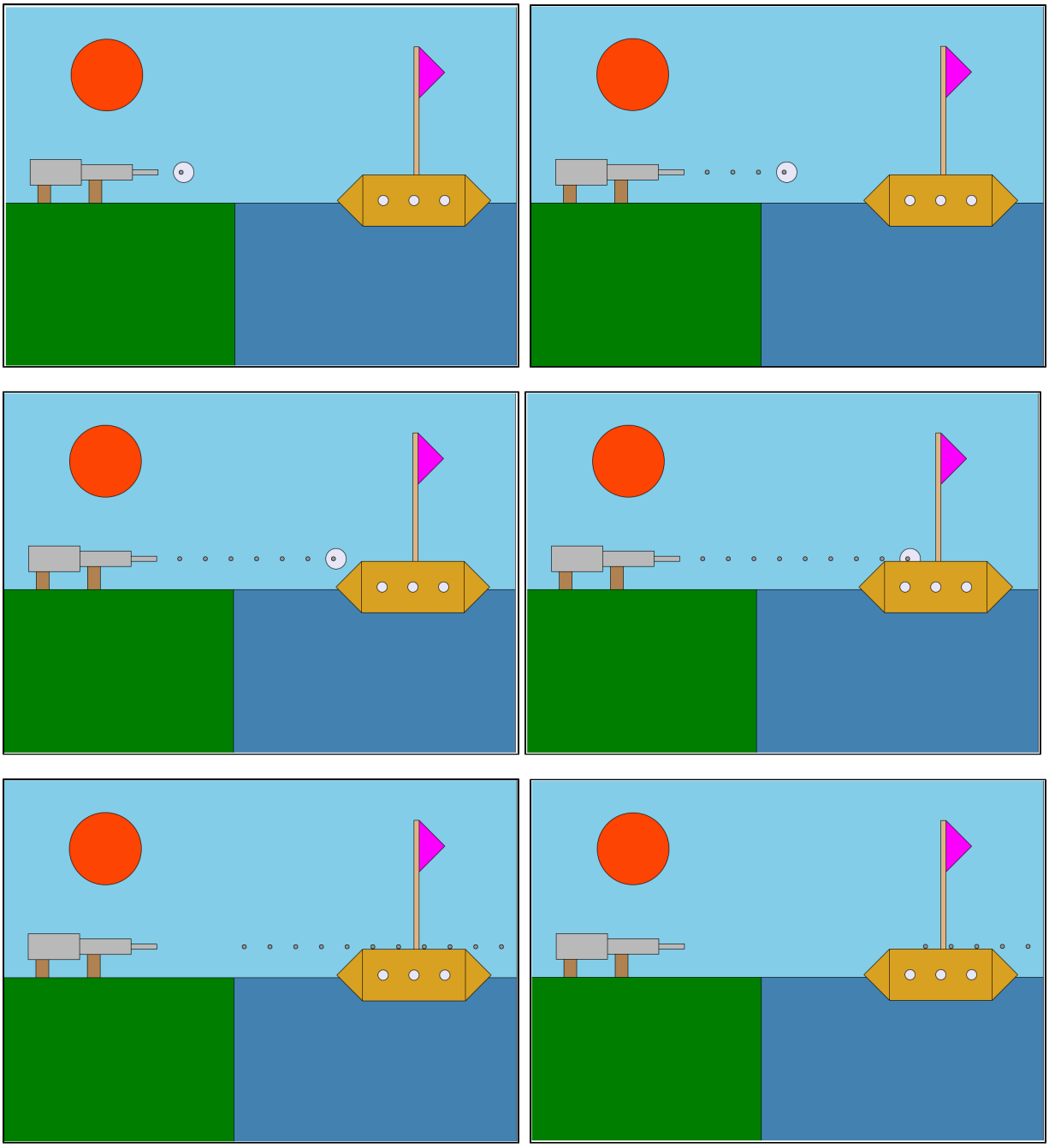
Also, in task 1 we were asked to make animate a single aspect of the complex objects. I decided to animate cannonballs shooting out of my cannon. To do this, I had to create a new functions called cannonball_animation_frame() with the parameters (cannon, frame_num, **ANDDD**) . Inside the animation function, I had to use for loops and if statements. Also, I had to use some new functions called move and **time package.** Her e is a snippet of my code from my animation of the cannonballs:

```
 if frame_num %2 == 0 and len(shapes) < 22:
     c = gr.Circle(gr.Point( newx, newy), 0.4*dy)
     c.setFill(gr.color_rgb(150, 150, 150))
     c.draw(win)
     shapes.append(c)

 for shape in shapes[5:]:
     shape.move( 25, 0)
     center = shape.getCenter()

     if center.y < 0:
         mx = newx - center.getX()
         my = newy - center.getY()
         shape.move( mx, my )
```

Task 2 asked me to create a new file called scene.py and this file was to house the main code for the displaying of my scene. I made a single function called main() that had all the draw functions of each object and the same animation function from complex_shapes.py test function. In the main() function I used had to use a animation for loop for the number of frames to call my cannonball animation. Here is what my final scene main code looked like:

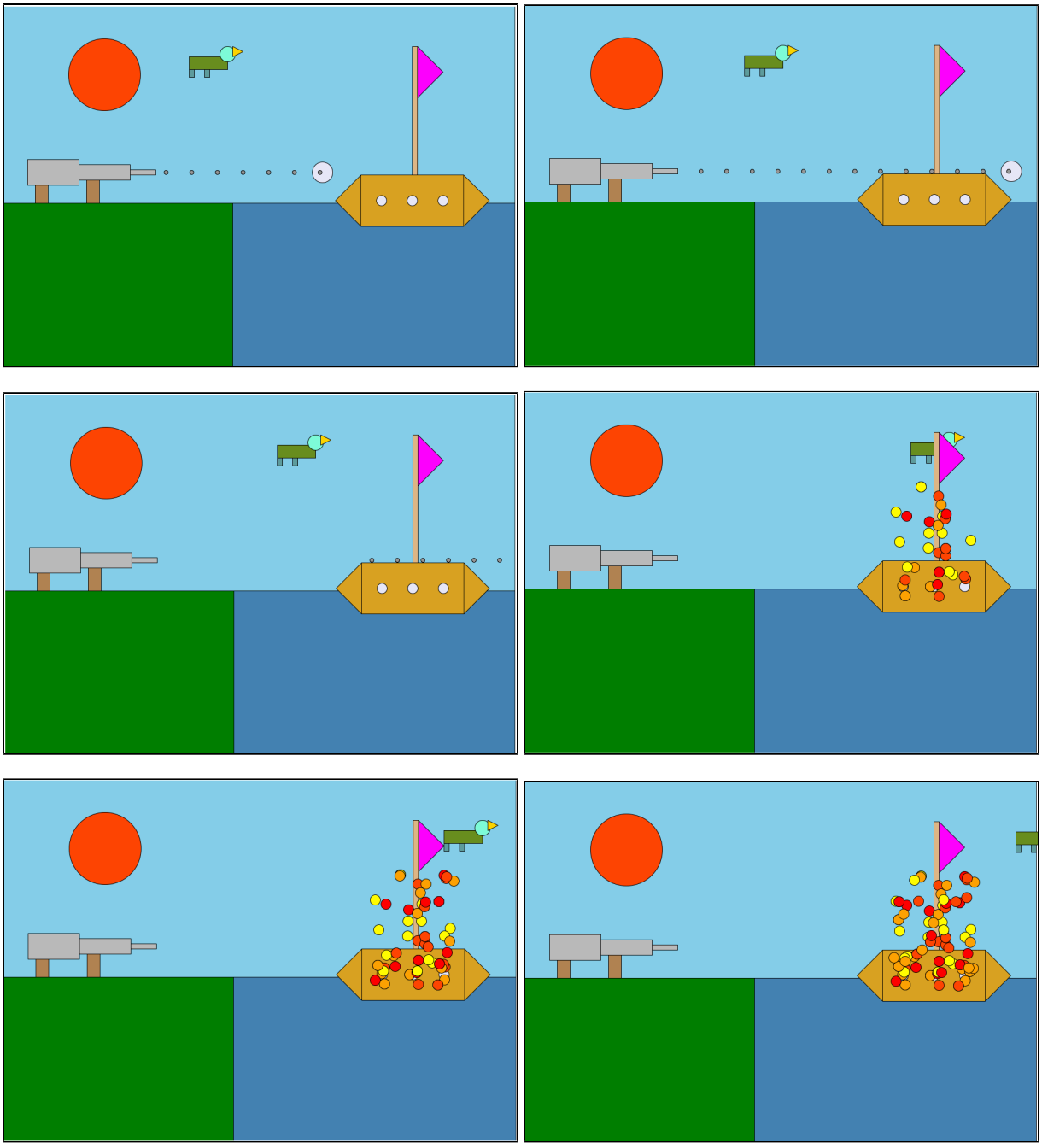
```
def main():
    """ Create a window and plot a scene with a
    of a steam plant in it. """
    win = gr.GraphWin( 'title', 1000, 700, False )

    # the three separate functions for objects
    landscape = cs.landscape_init(0, 385, 1.0)
    cannon = cs.cannon_init( 50, 300, 1.0 )
    ship = cs.ship_init( 700, 330, 1.0)

    #drawing the three separate object functions
    draw(landscape, win)
    draw(cannon, win)
    draw( ship, win)

    win.update()

    # ANIMATION for loop to shoot "big shot' and cannonballs
    for frame_num in range(100):
        time.sleep( 0.15 )
        print frame_num
        cs.cannonball_animation_frame( cannon, frame_num, win )
        win.update()
        if win.checkMouse():
            break

    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

Here is a collection of images to show the animation of the final scene:



After I finished all the required tasks for this project, I completed one extension. **Extension 1** was just two more animations. I decided to make the ship catch fire after the cannonballs passed through it and I added a bird in the sky. The bird is flying above the ship, but when the ship catches fire, the bird speeds off the screen. Here is a look at the animation code I added on for this extension:

Here is a collection of images to show what the animation of the extension looked like:



In this project, I learned an entirely new and different Python Graphics package. I learned how to code with Zelle Graphics and how to animate the objects I coded. I also learned how to use a the new time package to slow down/speed up animation frames. Like always, I polished my skills of for loops, conditional statements, parameters, and lists.

In this project I received help from Professor Taylor, Professor Maxwell, and fellow coder Julia Saul.