Zena's CS151 Project 6

For this week's project, we had to create an animated scene using some capabilities of the Zelle graphics package for Python. This meant creating the illusion of color changes, movement, and size changes by playing pictures very quickly in succession.

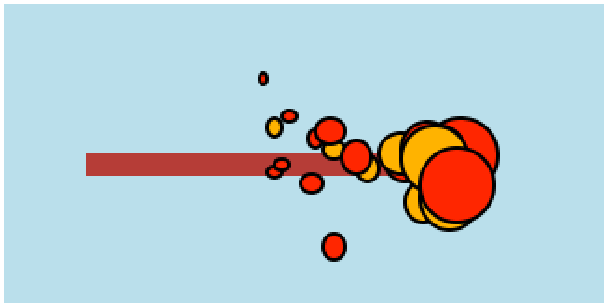
Task one was to create at least two shape groups. In my case, I created a flaming arrow group and a person group. Each shape group, in order to be placed and sized correctly, needed to take in x and y coordinates as well as a scale, which would be given at the end in the test function. For instance, making a simple rectangle for use in animation involved the following code:

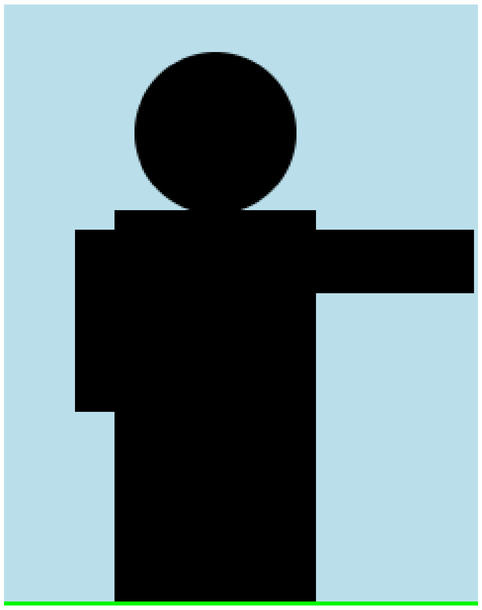
```python
def ground_init(x,y,scale):
    shapes=[]
    r=gr.Rectangle(gr.Point(x,y),gr.Point(x+scale*400,y-scale*50))
    r.setFill("Green")
    r.setOutline("Green")
    shapes.append(r)
    return shapes
```

After being given a location and scale, the function creates an empty list, where the shapes it creates will be stored one after the other. We create a rectangle defined by points (x,y) and (x+scale*400,y-scale*50). We fill it and outline it in green, and we add the resulting object to the shapes list. Since the draw, move, and undraw functions use lists for the sake of convenience when dealing with shape groups with many individual objects (like my person shape group), we simply make a list for each object's initial definition for consistency. It's also important to note that point (0,0) of a given window is in the top left corner, meaning the bigger the y value, the lower the point is in the window. I kept this in mind when deciding where to plot shape groups.
Making the arrow, in comparison to the ground, simply involved more coordinates; the circle part representing the fire would need to be placed on the tip of the arrow. Therefore, the following code was used:

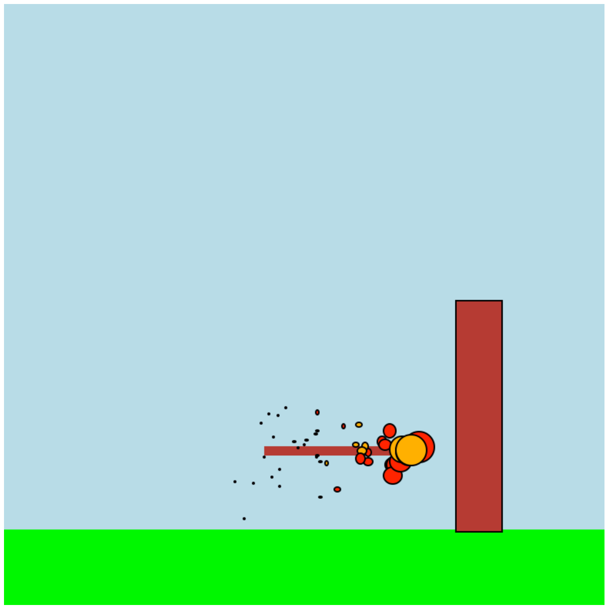
```python
shapes=[]
r = gr.Rectangle( gr.Point(x, y), gr.Point( x+scale*100,y-scale*5) )
r.setFill( 'Brown' )
r.setOutline('Brown')
shapes.append(r)
c=gr.Circle( gr.Point(shapes[0].getP2().getX(),shapes[0].getP2().getY()), scale*10)
```

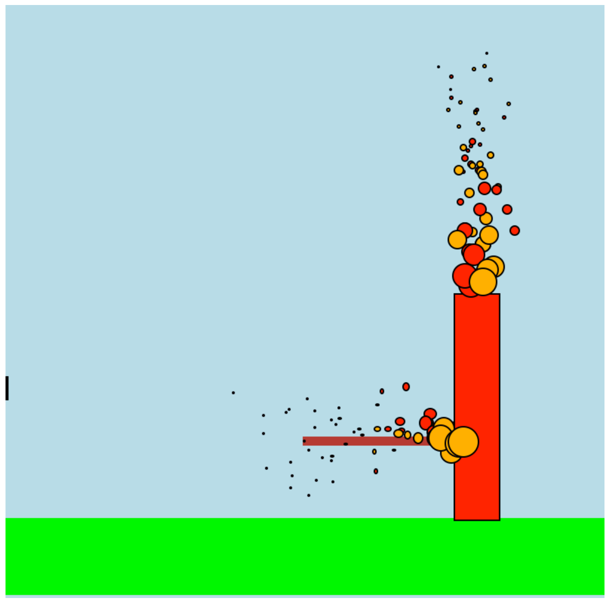
As you can see, the rectangle part was created first, then added to the list. Referring to the rectangle's spot on the list (index 0), the circle's coordinates were decided by using .getP and .getX(). The initial flaming arrow (plus some flames from the animation) and person are shown below.



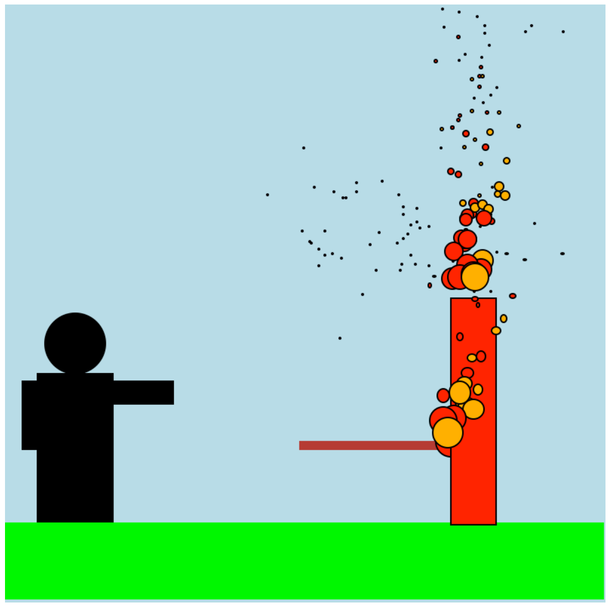


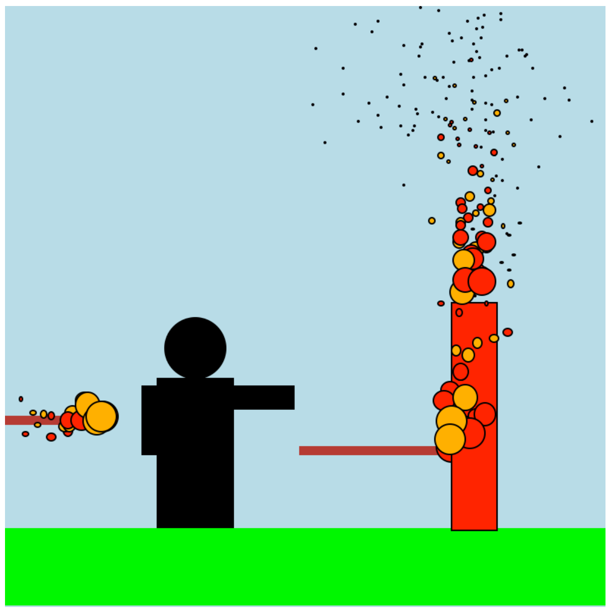
Task two involved animating at least one shape group. I animated the arrow so it would move while producing flames that change color and size until it hit the target, whereupon the arrow's fire would change direction and the target would light on fire as well. I first made it so that every item moved (using the move function on the list returned by the object group initialization) for every given frame, because we want the arrow to continue moving. Then I created 100 flames that got smaller with every frame by using an if statement to check if less than 100 shapes were in the object group list. If there were, they would each get assigned a new circle (size decreasing each time by using a multiple of .9 for the radius) with a new color (either red or orange, picked randomly from a list I provided) by undrawing the one present in the list and redrawing it to be what was just created. After that, the flames' location was decided by using the method mentioned before to pinpoint a spot on the arrow from which they could come. This part is what actually adds an initial flame ball onto the arrow, while the prior process looks for each one that is created and shrinks and changes the color of it. After that, I used an if statement to check if the arrow point was touching the object, and if it was, the object would set on fire, too, in a similar fashion to how the arrow is on fire. At the point the two objects touched, the fire on the arrow also started going more up, which I did by specifying a frame time that would be the point when the objects touch, and saying that if it is past that point, the fire should travel in a different path, since it is, of course, not moving with the arrow anymore. For the animation I also added a ground and sky. The next two pictures show the basic initial animation before I did the extensions.

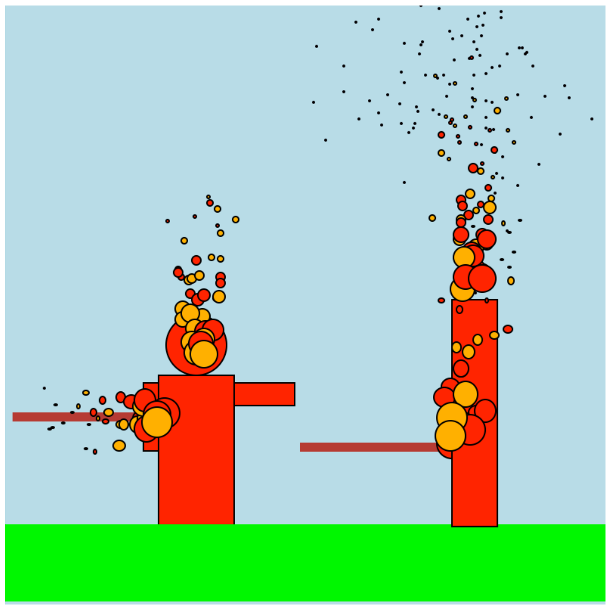




For extension 1, I animated another object: namely, the person. This was simple enough, and just involved telling the person (the whole object list) to move forward for a certain number of frames. Extension 3 called for making a story, which I did by adding in another arrow that shoots the person and lights them on fire after they get close to the other fire. This involved making another arrow because the list needed to be fresh, another function because the target was different this time (the person instead of the wooden target), and another for loop in the main test function because the person could not be moving at the same time as the arrow, so it would need to be after the arrow has finished its animation. It worked in a similar fashion to the other animations; the second arrow traveled until a certain frame number, which was at the time it came into contact with the left torso part of the person. At this time, the person also bursts into flames. The story is a bit strange; a person shoots a flaming arrow at a log and sets it on fire. The person goes to get the arrow, but before they can get it, is shot by another flaming arrow. What goes around comes around. Another way to interpret it is that the person was trying to save the log but was too late, and also got caught in the rain of flaming arrows. Tragic, really.







Through this project, I learned how to animate by using the time package and object movement. I also learned how to get the coordinates of a point of an object and use that to place something else relative to it. Lastly, I learned how to organize groups of objects into lists and use the list format to my advantage when deciding what to do to which parts of an object group.

Helped by: Prof. Maxwell