Andrew Wallace

CS2021 – Fall2017

12/04/2018

Wizard Wars

Introduction:

Wizard wars is a simple survival game coded in python using pygame. The objective is to shoot baddies and live as long as possible racking up as many kills as you can. You play a dark wizard who uses only fire spells to decimate the battlefield.

I used the library “pygame” for this project. It is perfect for dealing with the different sprites and has a lot of built in features to make collision detection much easier. Pygame includes computer graphics as well as sound libraries which can also be used. What I used it for was mostly creating character sprites but also to keep track of the games clock time.

So, the beginning of the program starts with importing various classes either created or just from pygame. Directly below that are some global variables I needed to create. We have default color values assigned for when I’m printing text to the screen, the speed of the player as well as the speed of the players projectiles among other things.

In the function section we have a “waitForPlayerToPressKey()” function which does exactly what the title of it is. I will run until the player presses a key. This is used to start/end the game. Directly below that we have the “drawText(text, font, surface, x, y)” function which takes several variables in order to place a specified message on the screen.

I created several pygame groups to manage of the unique sprites that would be on screen simultaneously.

Directly before the main loop I have a small explanation screen on how to play which lasts until the “waitForPlayerToPressKey()” is triggered. In the main loop we set all the movement directions to false initially. This simply says that our little wizard isn’t moving just yet. We create a “keys” variable which uses pygame to listen for a key to be pressed. We have an if block below it which will move the character 5 pixels in the direction of the pressed arrow key. It is formatted in such a way to be able to keep the sprite moving for as long as the key is pressed. This makes it a lot smoother to run.

Below that we have another “keyss” listener which is looking for the asdw keys. This will fire a projectile in the given direction. It is done by having the player press a key, then assigns a temporary variable to a cardinal direction. This tells the cast block (which is directly below) which direction to assign the given projectile.

Below all of that we have a couple blocks which just continuously update the projectiles and moves the baddies toward the player.

Project Results:

In this project I accomplished in learning a whole new way to use python using pygame. I had no prior experience before this but through extensive googling I found enough

information to fully develop a game of my own. I started the project by just finding an example of a game and adapting it to something I was trying to do. An example of this was using the way they made the character move. As well as how to make a “pregame screen” which will wait until the player hits a key to continue.

```
if event.type == KEYDOWN:
    if event.key == ord('z'):
        reverseCheat = True
    if event.key == ord('x'):
        slowCheat = True
    if event.key == K_LEFT or event.key == ord('a'):
        moveRight = False
        moveLeft = True
    if event.key == K_RIGHT or event.key == ord('d'):
        moveLeft = False
        moveRight = True
    if event.key == K_UP or event.key == ord('w'):
        moveDown = False
        moveUp = True
    if event.key == K_DOWN or event.key == ord('s'):
        moveUp = False
        moveDown = True
```

A huge pitfall I found was that I discovered I could not figure out how they interreacted with the character itself. Specifically, when I was trying to make a “baddie”. They used a format like this

```
newBaddie = {'rect': pygame.Rect(random.randint(0, WINDOWWIDTH-baddieSize), 0 - baddi
             'speed': random.randint(BADDIEMINSPEED, BADDIEMAXSPEED),
             'surface':pygame.transform.scale(baddieImage, (baddieSize, baddieSize)),
             }
```

So ultimately I ended up scrapping that entire first program. That set me back a lot. But I was able to get my 2nd doing all the same things as my first program in the same amount of time I spent trying to fix an error in the first iteration. So that was a big confidence boost.

I started using more of class system in the 2nd program which sped up the process a lot. Currently I have one error that I am unable to solve but you probably won't notice if you don't really inspect the program.

```
for s, d in zip(spell, direction):
    if d == "n":
        s.castUp(SPELLSPEED)
        if s.rect.y >= WINDOWHEIGHT:
            spell.remove(s)
            direction.remove(d)
            spell_sprites_list.remove(s)
    if d == "s":
        s.castDown(SPELLSPEED)
        if s.rect.y <= 0:
            spell.remove(s)
            direction.remove(d)
            spell_sprites_list.remove(s)
    if d == "e":
        s.castRight(SPELLSPEED)
        if s.rect.x >= WINDOWWIDTH:
            spell.remove(s)
            direction.remove(d)
            spell_sprites_list.remove(s)
    if d == "w":
        s.castLeft(SPELLSPEED)
```

This is in my main loop and all it does is make the spells move in their dedicated directions until they get to the edge of the screen and then disappear. I just could not get the left side of the screen to work as well as all the others did. So, which the projectile disappears once it hits the end in the north, east, and south directions. I had to just let it keep going in the west direction. It seems like it would have been a simple fix, but I did try everything I could think of and it never worked. So, fixing that is something you could do for a future improvement. Another is adding bosses for X number of enemies killed.

Bibliography:

"Pygame - How To?" *101 Computing*, https://www.101computing.net/pygame-how-tos/. Accessed 30 Nov. 2018.

Pygame.Rect — Pygame v1.9.5.Dev0 Documentation. https://www.pygame.org/docs/ref/rect.html. Accessed 30 Nov. 2018.

"Python - Why Won't the Bullets Fire in the Direction the Player Is Facing and Stay Visible in PyGame?" *Stack Overflow*, https://stackoverflow.com/questions/45665784/why-wont-the-bullets-fire-in-the-direction-the-player-is-facing-and-stay-visibl. Accessed 30 Nov. 2018.

https://inventwithpython.com/chapter20.html. Accessed 30 Nov. 2018.

https://www.pygame.org/news. Accessed 30 Nov. 2018.

Code Appendix:

https://github.com/WallyWallace188/Wizard-Wars