Design documents:

Resouses studied:

1. <http://www.pygame.org>
2. 3 downloaded books. a) Beginning Game Development with python and pygame form Novice to Professional by Will McGugan b)Invent your own computer games with python by AI Swegiart c) Making games with python& pygame by AI Sweigart
3. Terraria (most of my recourses are pulled on terraria.wiki, but not the code though because terraria uses a complete different language and I cannot even open the files)
4. Examples studied: DemonKingDom form pygame.org

Foxen-v0.1.0 from pygame.org

Run! From pygame.org

Micro-Crawl-0.1.0 form pygame.org

BasicTD from pygame.org

And class materials such as: KeyPressed.py by Kosbie

mouseEvent.py by Kosbie

class.py by Kosbie

Problems encountered:

1. Did not know how to use pygame. Read a lot online materials first. Form Apr.1st to Apr.14th
2. Resources needed. Went online to find materials
3. Deal with pygame display and update. You have to call display.flip() every time to actually update the view. And the order of what you are drawing is really important.
4. When I tried adding the pause menu, it often darkens by itself, and I found out it’s because when you call it in a while loop(main game loop), it actually draws on the old one and so the color condenses to deeper. And therefore I created an needUpdate Boolean to help me identify if the menu actually need update(user input recieved). Also I used % in pause menu to help, which I think I was being smart.
5. When I tried to add the monster I actually cannot use inheritance form player cuz they do different stuff. So I have to rewrite the whole monster class and think how can I interact the monster with the player.
6. Basically my game loop is start -> main run function -> call update on player and monster -> get any input -> read the input -> interpret the input -> returns to main run function -> main run function check collisions and everything -> returns output to player and monster -> player and monsters react(move, die, draw things)->main run function calls pygame.display.update() and then does the same thing again. This may not be the best way to make a game but I think this is pretty clear.
7. When the spiky ball just won’t drop, I was frustrated. Turns out that I have to create pygame.sprite.Group to hold all the spiky balls and their information. And then check if it collides with anything with the spiky balls list. This cost me a lot of time.
8. When you shoot the bullets from the rifle, simply getting two points and calculating the slope and line function is not enough. Because the screen use x axis going right and y-axix going down, when you simply get the line and slope and use the line equation on the bullet, it goes the opposite way. I think using a lot of if functions may help, but instead I used vectors to help.
9. The floor collision is actually pretty tough. You have to consider situations that user input but collides and that bottom is larger than top. I drew a lot of rectangles on paper and screen to help me debug and finally, it’s done.
10. The music is not so tricky but I think it’s pretty nice to get it done. I got the music from online, <http://love2d.org/wiki/Free_Game_Resources>
11. Again, time is another factor too. I was sick for the middle week and cannot finish much during that week, so I guess this was one of the problems that annoys me.