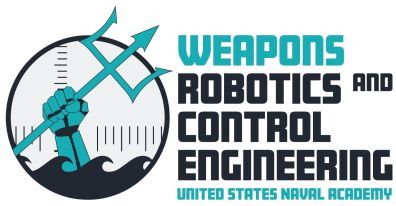
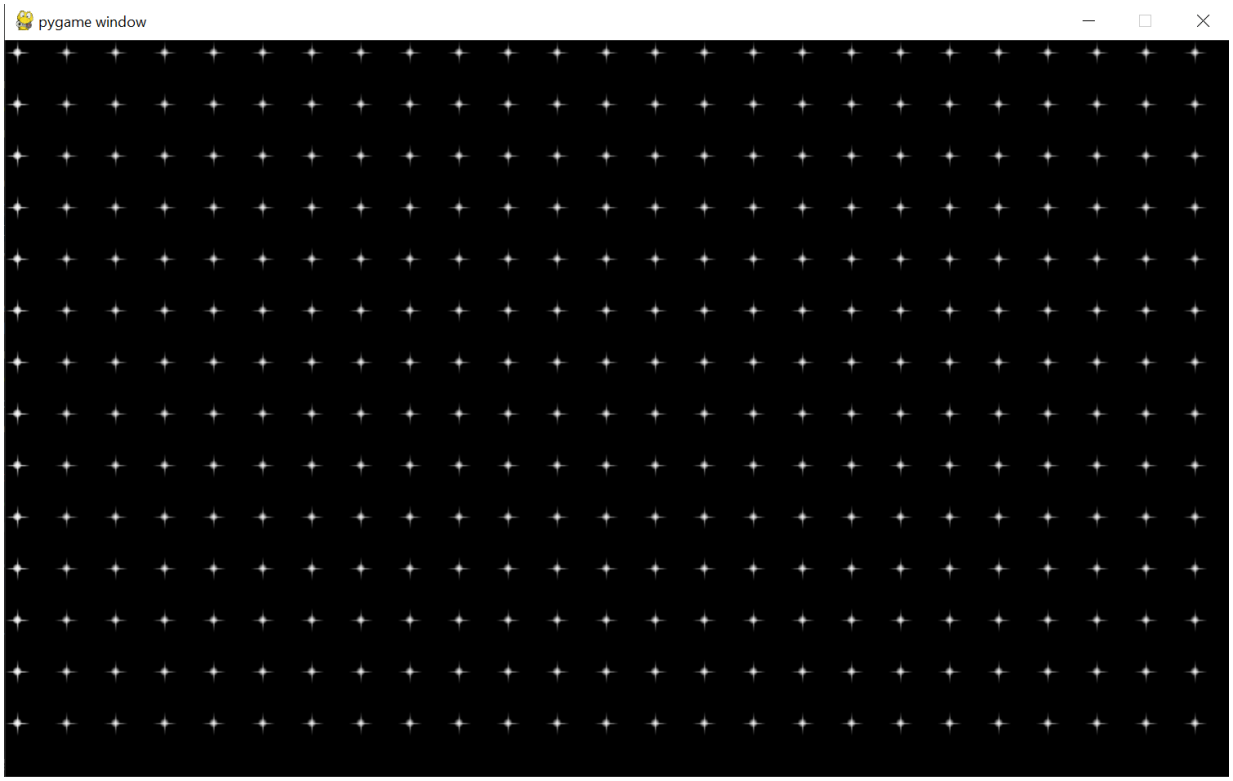


Alien Invasion (Chapter 13)

Create a seperate file in your alien invasion game for each of the following questions. Paste the content of the file into a Jupyter Notebook cell along with a screen shot of the game (2 cells per answer)

13-1. Stars: Find an image of a star. Make a grid of stars appear on the screen.

In [2]:
```
import pygame
import time

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((0, 0, 0))
star = pygame.image.load('particleStar.png')
star_rect = star.get_rect()
screen_rect = screen.get_rect()
screen.blit(star, star_rect)
while True:
    while True:
        screen.blit(star, star_rect)
        star_rect.x += (2 * star_rect.width)
        if star_rect.right > screen_rect.right:
            break
    star_rect.left = screen_rect.left
    star_rect.y += (2 * star_rect.height)
    if star_rect.bottom > screen_rect.bottom:
        break
    screen.blit(star, star_rect)
pygame.display.flip()
time.sleep(10)
```

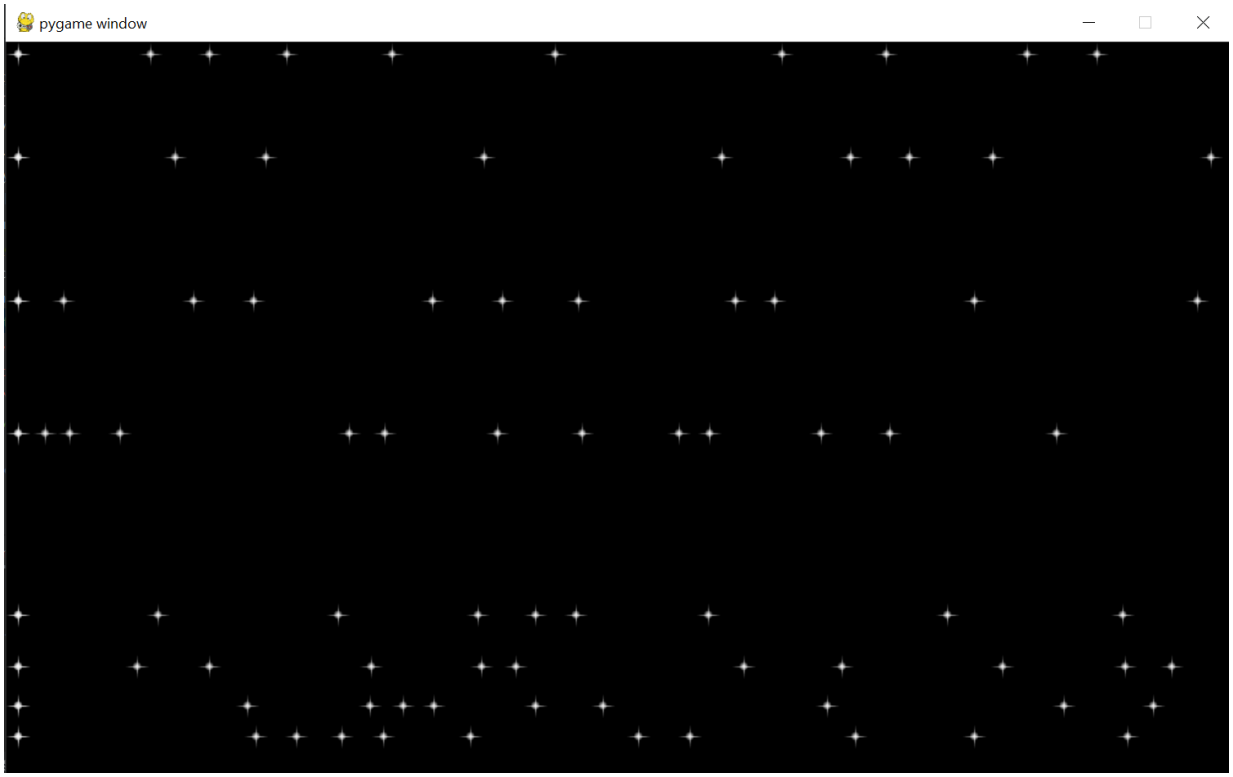


13-2: Better Stars: You can make a more realistic star pattern by introducing randomness when you place each star. Recall that you can get a random number like this:

```
from random import randint
random_number = randint(-10,10)
```

This code returns a random integer between -10 and 10. Using your code from 13-1, adjust each star's position by a random amount.

In [2]:
```
import pygame
import time
from random import randint

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((0, 0, 0))
star = pygame.image.load('particleStar.png')
star_rect = star.get_rect()
screen_rect = screen.get_rect()
screen.blit(star, star_rect)

while True:
    while True:
        randnumx = randint(star_rect.width, 200)
        screen.blit(star, star_rect)
        star_rect.x += randnumx
        if star_rect.right > screen_rect.right:
            break
    randnumy = randint(star_rect.height, 200)
    star_rect.left = screen_rect.left
    star_rect.y += randnumy
    if star_rect.bottom > screen_rect.bottom:
        break
    screen.blit(star, star_rect)
pygame.display.flip()
time.sleep(10)
```

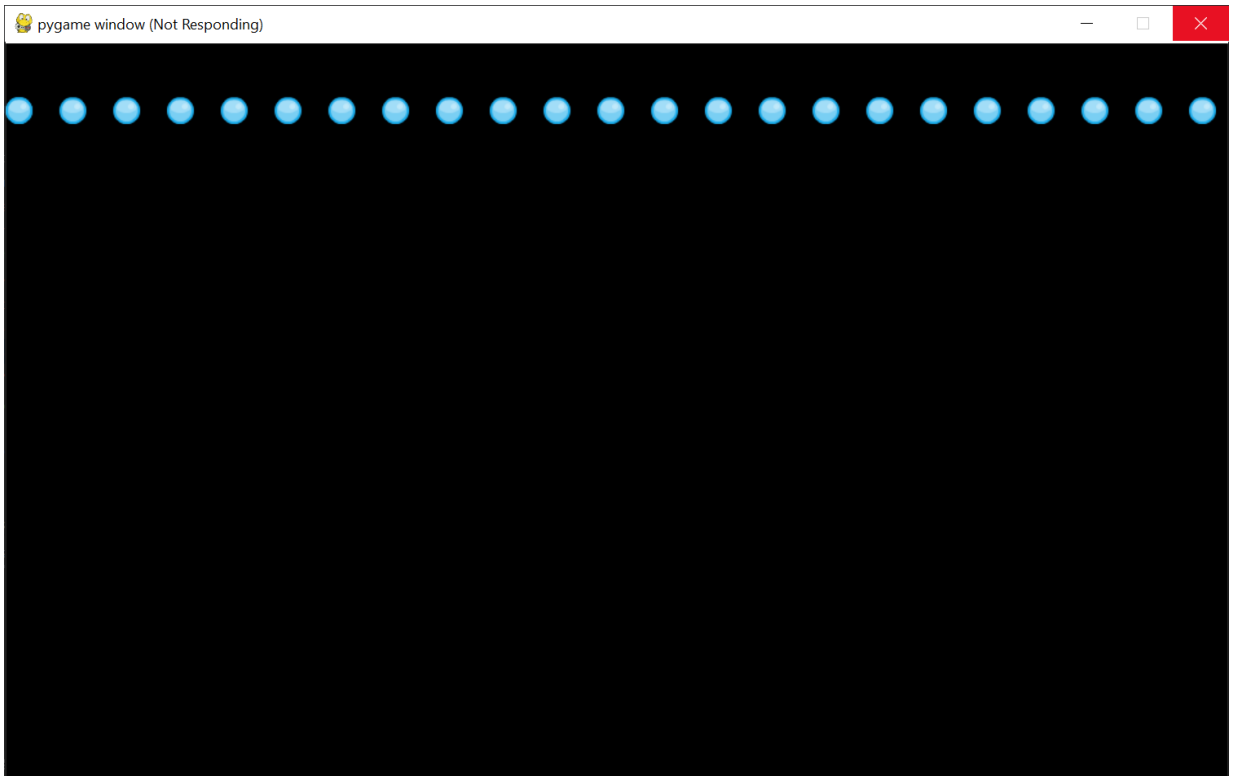


13-3. Raindrops: Find an image of a raindrop and create a grid of raindrops. Make the raindrops fall toward the bottom of the screen until they dissappear.

In [2]:
```
import pygame
import time

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((0, 0, 0))
star = pygame.image.load('ballBlue.png')
star_rect = star.get_rect()
screen_rect = screen.get_rect()

while True:
    screen.fill((0, 0, 0))
    screen.blit(star, star_rect)
    while True:
        screen.blit(star, star_rect)
        star_rect.x += (2 * star_rect.width)
        if star_rect.right > screen_rect.right:
            break
    star_rect.left = screen_rect.left
    star_rect.y += (2 * star_rect.height)
    if star_rect.bottom > screen_rect.bottom:
        screen.fill((0,0,0))
        break
    time.sleep(.2)
    pygame.display.flip()
pygame.display.flip()
time.sleep(10)
```

13-4. Steady Rain: Modify the code from 13-3 so when a row of rain drops disappears off the bottom of the screen, a new row appears at the top of the screen and begins to fall.

In [2]:
```
import pygame
import time

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((0, 0, 0))
star = pygame.image.load('ballBlue.png')
star_rect = star.get_rect()
screen_rect = screen.get_rect()

while True:
    screen.fill((0, 0, 0))
    screen.blit(star, star_rect)
    while True:
        screen.blit(star, star_rect)
        star_rect.x += (2 * star_rect.width)
        if star_rect.right > screen_rect.right:
            break
    star_rect.left = screen_rect.left
    star_rect.y += (2 * star_rect.height)
    if star_rect.bottom > screen_rect.bottom:
        screen.fill((0,0,0))
        star_rect.topleft = screen_rect.topleft
        continue
    time.sleep(.2)
    pygame.display.flip()
```



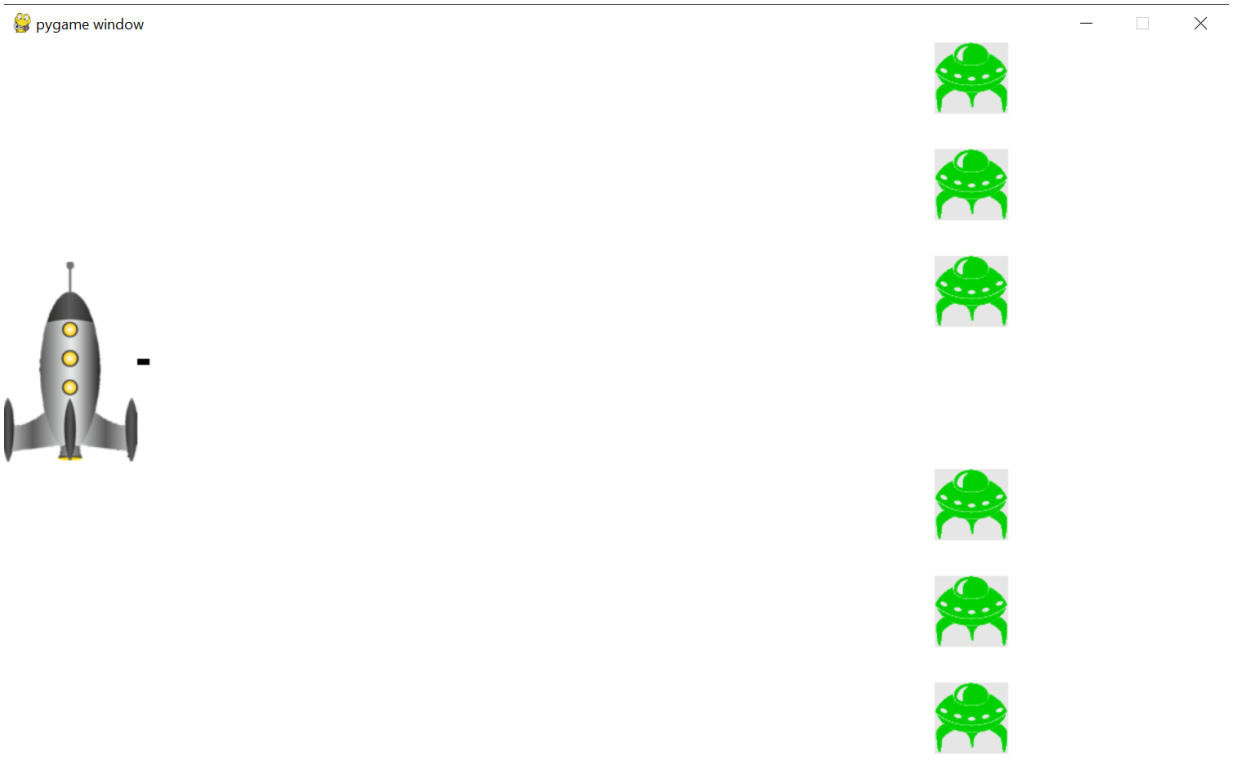
13-5 Sideways Shooter Part 2: We've come a long way since Exercise 12-6, Sideways Shooter. For this exercise, try to develop Sideways Shooter to the same point we've brought *Alien Invasion* to. Add a fleet of aliens, and make them move sideways toward the ship. Or, write code that places

aliens at random positions along the right side of the screen and then sends them toward the ship. Also, write code that makes the aliens disappear when they're hit.

In [2]:

```
import sys
import pygame
import time
from pygame.sprite import Sprite

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((255, 255, 255))

sprout = pygame.image.load("C:/Users/m253768/Desktop/EW200/Labs/Lab7/alien_invasi
alien = pygame.image.load("C:/Users/m253768/Desktop/EW200/Labs/Lab7/alien_invasic
character_rect = sprout.get_rect()
screen_rect = screen.get_rect()
character_rect.midleft = screen_rect.midleft
bullet = pygame.Rect(0,0,10,5)
color = (0,0,0)
bullet.midleft = character_rect.midright
alien_rect = alien.get_rect()
alien_rect.topright = screen_rect.topright
clear_alien = pygame.Rect(alien_rect.left, 0, screen_rect.width-alien_rect.left,
clear = pygame.Rect(character_rect.right, 0, screen_rect.width-character_rect.wic
collisions = pygame.Rect.colliderect(bullet, alien_rect)
x=0



while True:
    time.sleep(1)
    clock = pygame.time.Clock()
    for y in range(0, screen_rect.height):
        screen.blit(alien, alien_rect)
        alien_rect.y += 1.5 * alien_rect.height
        pygame.display.flip()
    clear_alien = pygame.Rect(alien_rect.right, 0, screen_rect.width - alien_rect
    screen.fill((255,255,255), rect=clear_alien)
    alien_rect.top = screen_rect.top
    alien_rect.x -= 1.5 * alien_rect.width
    screen.fill((255, 255, 255), rect=character_rect)
    screen.fill((255, 255, 255), rect=bullet)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                character_rect.y -= 40
                bullet.midleft = character_rect.midright
                if character_rect.top < screen_rect.top:
                    character_rect.top = screen_rect.top

            elif event.key == pygame.K_DOWN:
                character_rect.y += 40
                bullet.midleft = character_rect.midright
                if character_rect.bottom > screen_rect.bottom:
                    character_rect.bottom = screen_rect.bottom
```

```
            elif event.key == pygame.K_SPACE:
                while bullet.right < screen_rect.right:
                    clear_bullet = pygame.Rect(bullet.left, bullet.top-20, 50, 10
                    screen.fill((255,255,255), rect= clear_bullet)
                    screen.blit(sprout, character_rect)
                    bullet.x += 10
                    pygame.draw.rect(screen, color, bullet)
                    pygame.display.flip()
                    time.sleep(.02)

                if bullet.right >= screen_rect.right:
                    screen.fill((255, 255, 255), rect=bullet)
                    bullet.midleft = character_rect.midright

    screen.blit(sprout, character_rect)
    pygame.draw.rect(screen, color, bullet)
    if alien_rect.left < character_rect.right:
        alien_rect.topright = screen_rect.topright
        screen.fill((255,255,255), rect=clear)
    if collisions == True:
        screen.fill((255, 255, 255), rect=alien_rect)
        screen.fill((255, 255, 255), rect=bullet)
        x +=1

    clock.tick(60)
    pygame.display.flip()
```

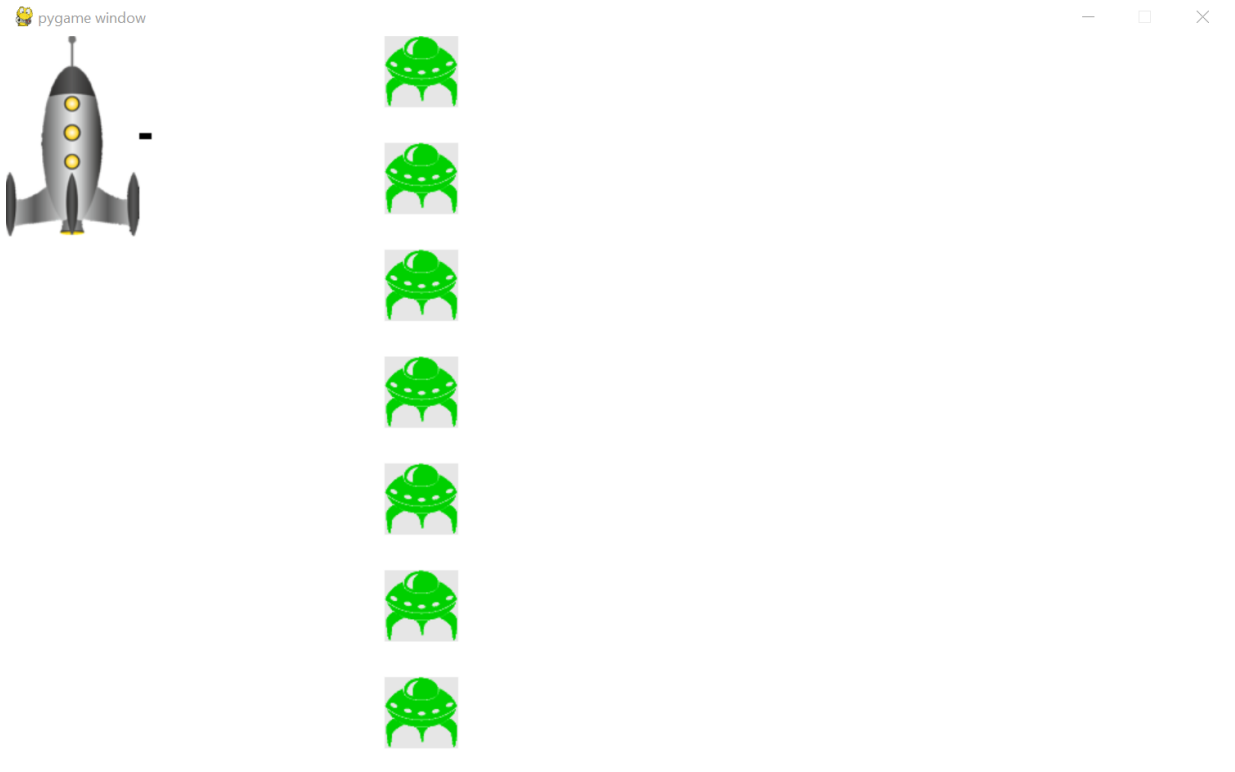


13-6. Game Over: In Sideways Shooter, keep track of the number of times the ship is hit and the number of times an alien is hit by the ship. Decide on an appropriate condition for ending the game, and stop the game when this situation occurs.

In [2]:

```
import sys
import pygame
import time
from pygame.sprite import Sprite

pygame.init()
screen = pygame.display.set_mode((1000, 600))
screen.fill((255, 255, 255))

sprout = pygame.image.load("C:/Users/m253768/Desktop/EW200/Labs/Lab7/alien_invasi
alien = pygame.image.load("C:/Users/m253768/Desktop/EW200/Labs/Lab7/alien_invasic
character_rect = sprout.get_rect()
screen_rect = screen.get_rect()
character_rect.midleft = screen_rect.midleft
bullet = pygame.Rect(0,0,10,5)
color = (0,0,0)
bullet.midleft = character_rect.midright
alien_rect = alien.get_rect()
alien_rect.topright = screen_rect.topright
clear_alien = pygame.Rect(alien_rect.left, 0, screen_rect.width-alien_rect.left,
clear = pygame.Rect(character_rect.right, 0, screen_rect.width-character_rect.wic
collisions = pygame.Rect.colliderect(bullet, alien_rect)
x=0



while x<10:
    time.sleep(1)
    clock = pygame.time.Clock()
    for y in range(0, screen_rect.height):
        screen.blit(alien, alien_rect)
        alien_rect.y += 1.5 * alien_rect.height
        pygame.display.flip()
    clear_alien = pygame.Rect(alien_rect.right, 0, screen_rect.width - alien_rect
    screen.fill((255,255,255), rect=clear_alien)
    alien_rect.top = screen_rect.top
    alien_rect.x -= 1.5 * alien_rect.width
    screen.fill((255, 255, 255), rect=character_rect)
    screen.fill((255, 255, 255), rect=bullet)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                character_rect.y -= 40
                bullet.midleft = character_rect.midright
                if character_rect.top < screen_rect.top:
                    character_rect.top = screen_rect.top

            elif event.key == pygame.K_DOWN:
                character_rect.y += 40
                bullet.midleft = character_rect.midright
                if character_rect.bottom > screen_rect.bottom:
                    character_rect.bottom = screen_rect.bottom
```

```
            elif event.key == pygame.K_SPACE:
                while bullet.right < screen_rect.right:
                    clear_bullet = pygame.Rect(bullet.left, bullet.top-20, 50, 10
                    screen.fill((255,255,255), rect= clear_bullet)
                    screen.blit(sprout, character_rect)
                    bullet.x += 10
                    pygame.draw.rect(screen, color, bullet)
                    pygame.display.flip()
                    time.sleep(.02)

                if bullet.right >= screen_rect.right:
                    screen.fill((255, 255, 255), rect=bullet)
                    bullet.midleft = character_rect.midright

    screen.blit(sprout, character_rect)
    pygame.draw.rect(screen, color, bullet)
    if alien_rect.left < character_rect.right:
        alien_rect.topright = screen_rect.topright
        screen.fill((255,255,255), rect=clear)
    if collisions == True:
        screen.fill((255, 255, 255), rect=alien_rect)
        screen.fill((255, 255, 255), rect=bullet)
        x +=1

    clock.tick(60)
    pygame.display.flip()
```



In [ ]: