KShoot App-> A classic Android-style shooting game using PyGame and Python. It is a classic arcade-styled game filled with 3 different enemies of varying speeds and sizes that give the player progressively more points as the games get more difficult. I created three game modes where the player can see how fast to clear the entire game in free play mode, then in accuracy mode where the player has limited ammo to get as many points as they possibly can. The highest scoring is tracked and saved in an external text file that the player can reset from the menu screen if they ever want to start over the game. The player will LEVEL UP to the higher levels if they get enough scores.

Developing Languages, Tools, and Techniques Needed:

Python 3 https://www.python.org/downloads/

Vscode 1.73 https://code.visualstudio.com/

PyGame https://www.pygame.org/news

pip https://pypi.org/project/pip/

Prerequisites & Setups:

Upgrade pip in Vscode terminal:

```
pip install --upgrade pip
```

Install PyGame with pip:

```
pip3 install pygame
```

Synchronous Note:

Import PyGame in `main.py`:

```
import pygame
```

Initiate PyGame and import 3 levels of backgrounds so we got 3 levels of game backgrounds:

level1 background.PNG

level2 background.PNG

level3 background.PNG

Draw guns and different colors of lasers with various positions:

```
 def draw_gun():
    mouse_pos = pygame.mouse.get_pos()
    gun_point = (WIDTH / 2, HEIGHT - 200)
    lasers = ['red', 'purple', 'green']
    clicks = pygame.mouse.get_pressed()
    if mouse_pos[0] != gun_point[0]:
        slope = (mouse_pos[1] - gun_point[1]) / (mouse_pos[0] -
gun_point[0])
```

Adjust guns sizes to be smaller guns to be at center with better accuracy targeting objects:

```
guns.append(pygame.transform.scale(pygame.image.load(f'assets/guns/{i}
.png'), (100, 100)))
```

gun1.PNG

gun2.PNG

gun3.PNG

Draw levels with coordinates and different targets and initialize enemies coordinates:

level1 birds.PNG

level2 bubbles.PNG

level3 alien spaceships.PNG

Check shots:

```
def check_shot(targets, coords):
    global points
    mouse_pos = pygame.mouse.get_pos()
    for i in range(len(targets)):
        for j in range(len(targets[i])):
            if targets[i][j].collidepoint(mouse_pos):
                coords[i].pop(j)
                points += 10 + 10 * (i ** 2)
```

Now we can click the floating and moving targets to destroy them on board.

Draw Score:

```
def draw_score():
    points_text = font.render(f'Points: {points}', True, 'black')
    screen.blit(points_text, (320, 660))
    shots_text = font.render(f'Total Shots: {total_shots}', True,
'black')
    screen.blit(shots_text, (320, 687))
    time_text = font.render(f'Time Elasped: {time_passed}', True,
'black')
    screen.blit(time_text, (320, 714))
    if mode == 0:
        mode_text = font.render(f'Freeplay!', True, 'black')
    if mode == 1:
        mode_text = font.render(f'Ammo Remaining: {ammo}', True,
'black')
    if mode == 2:
        mode_text = font.render(f'Time Remaining {time_remaining}',
True, 'black')
    screen.blit(mode_text, (320, 741))
```

score drawed level1.PNG

score drawed level2.PNG

score drawed level3.PNG

Menu control and setup:

```
def draw_menu():
    global game_over, pause, mode, level, menu, time_passed,
total_shots, points, ammo
    global time_remaining, best_freeplay, best_ammo, best_timed,
write_values, clicked, new_coords
    game_over = False
    pause = False
```

menu control bar showed.PNG

Finally, Add pause, resume functionalities and background music in different levels.