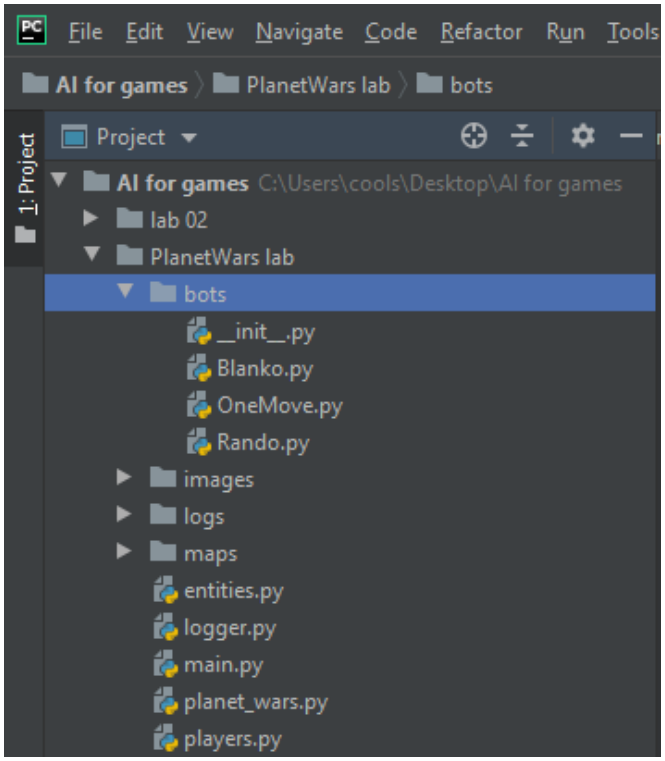
Created a new Bot called it Rando inside the bot directory



Then added some code that was already provided for the new bot and used it, while also amending the players = [“”] called in the main file, so that it is launches one fleet at a time.

```
    if gameinfo.my_fleets:
        return
```

```
if __name__ == '__main__':
    gamestate = open('./maps/map5.txt').read()
    players = ['Blanko', 'OneMove','Rando']
    window = PlanetWarsWindow(gamestate=gamestate, players=players, max_game_length=500)
    app.run()
    window.game.logger.flush()
```

Then Used the code provided to carry out below functionality

Time to attack! The basic design is this:

1. Make sure we have a planet we can launch a fleet from. (Do you know why is this?)
2. Make sure there is at least one planet to attack! (Again, make sure you know why this is.)
3. Select a random source planet from the planets we have (my_planets).
4. Select a random target planet from the planets that are not in our control (not_my_planet
5. Launch a fleet if the sources planet has more than 10 ships.
6. Only launch 75% of the ships available from the source planet.

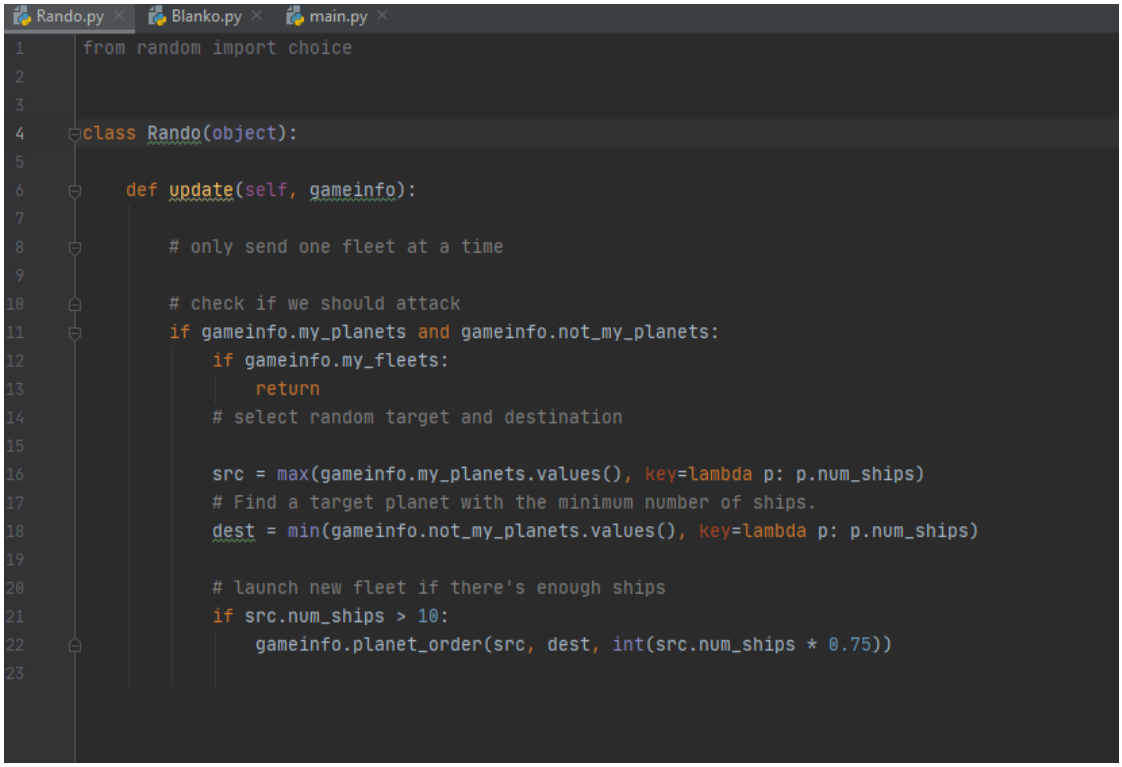
Here’s one way to do all that:

```
        # check if we should attack
        if gameinfo.my_planets and gameinfo.not_my_planets:
            # select random target and destination
            dest = choice(list(gameinfo.not_my_planets.values()))
            src = choice(list(gameinfo.my_planets.values()))
            # launch new fleet if there's enough ships
            if src.num_ships > 10:
                gameinfo.planet_order(src, dest, int(src.num_ships * 0.75) )
```

To change the behaviour of the bot from shooting random targets we used the code provided below: where it used the destination point by finding a planet with the minimum number of ships.

```
src = max(gameinfo.my_planets.values(), key=lambda p: p.num_ships)
dest = min(gameinfo.not_my_planets.values(), key=lambda p: p.num_ships)
```

Rando bot:



Tried using the other provided bot blanko and tested the inverse proportional search

The output was something like this: