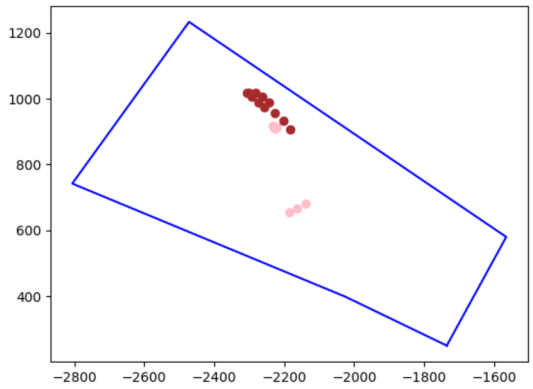
1. Is entering via the light blue boundary a common strategy used by Team2 on T (terrorist) side?

No, the only one time Team2 entered via light blue boundary happened on round 16 with player 5 and 9. And here is the trace of 2 players' movements.




```
state = ProcessGameState.ProcessGameState("game_state_frame_data.parquet")
state.filter_by_side("T")
state.filter_by_team_name("Team2")
boundary_xy = [[-1735, 250], [-2024, 398], [-2806, 742], [-2472, 1233], [-1565, 580]]
boundary_z = [285, 421]
state.check_if_any_row_in_boundary(boundary_xy,boundary_z)
state.write_to_csv("result1.csv")
```

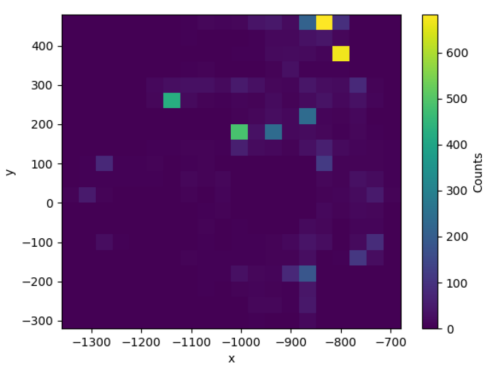
Script Explanation:
First filter the dataset with T side, Team2.
Then given x,y,z boundary for an area to create a polygon and judge if a row is in this polygon, the cache_dataframe will save the qualified rows.
Write the qualified rows into a csv file.

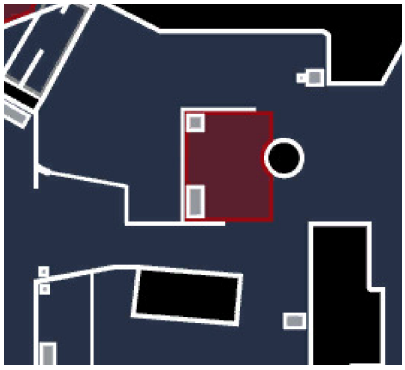
2. What is the average timer that Team2 on the T (terrorist) side enters "BombsiteB" with at least 2 rifles or SMGs?
The average time is 29 seconds after the game starts. Happens on round 21,28,30.

There is some ambiguity, I ignore the case that the player is not alive after entering the bombsiteB, so if there are 2 players entering bombsiteB one by one, the first player dies before the second player enters, would this be considered a valid case?

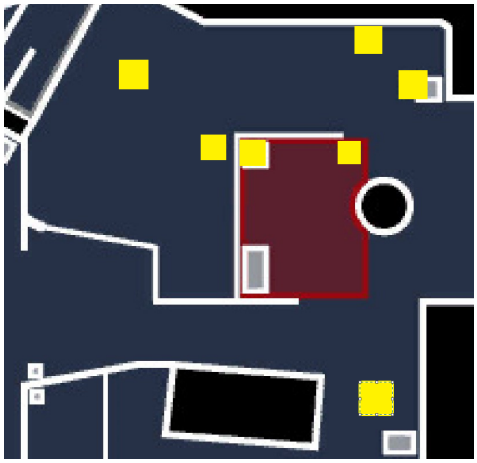
```
state = ProcessGameState("game_state_frame_data.parquet")
state.filter_by_side("T")
state.filter_by_team_name("Team2")
state.filter_by_area_name("BombsiteB")
state.filter_by_alive(True)
state.extract_xyz()
state.extract_weapon()
print(state.calculate_avg_time())
state.write_to_csv("result1.csv")
```

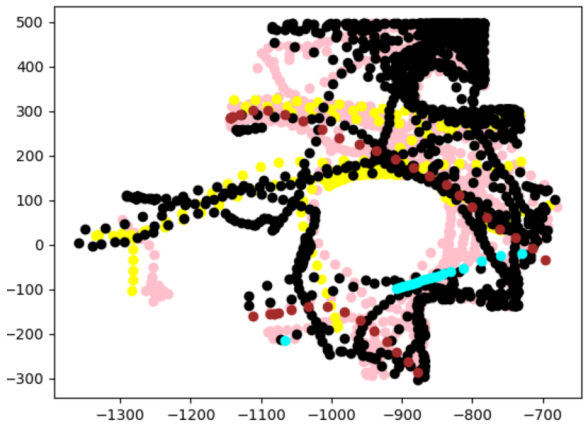
3. Now that we've gathered data on Team2 T side, let's examine their CT (counter-terrorist) Side. Using the same data set, tell our coaching staff where you suspect them to be waiting inside "BombsiteB"





The left picture is the heatmap generated by code. The map picture for bombsiteB is on the right.





The left picture is a rough combination of a heatmap with the game map. The yellow squares are the places that CT from Team2 will more likely stay at bombsiteB.
And the right picture is the trace of Team2 CT on bombsiteB as reference.
The result will be much more precise if I can get the x,y,z boundary of bombsiteB like the light blue area.

```
state = ProcessGameState.ProcessGameState("game_state_frame_data.parquet")
state.filter_by_side("CT")
state.filter_by_team_name("Team2")
state.filter_by_area_name("BombsiteB")
state.generate_heatmap()
```

3. (No Coding) Most of the time, our stakeholders (in this case, the CS:GO coaching staff) aren't tech-savvy enough to run code themselves. Propose a solution to your product manager that:

1. For some simple requests like filtering by team, round_number, player, etc, the code can be reused easily and stakeholders can get the output csv file. But stakeholders can do the same thing using excel after converting the parquet file to csv.

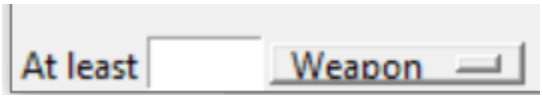
(a simple mockup)

For question 1 and 3, the stakeholders can implement themselves using simple GUI

2. A more reusable code means stakeholders can more easily acquire the output.

   The issue is in question 2 that needs to satisfy the " least 2 rifles or SMGs". The variation of this requirement can be " most 2 rifles or SMGs", "least 1 rifles or SMG", " least 2 grenade or SMGs", "no rifle" … Also, the requirement can also not be limited to weapon_class, then the reusability of the code will be challenged.

   If the query is limited to "least (1,2,3,4….) [weapon]", a simple mockup can be used.



something like this

   If there is a new requirement that needs an update on the class and script. Stakeholders can send a request and the request will be distributed to a developer so that a hard coded version should be released in hours, it can be quicker if the developer is on call. If there are multiple similar requests, developers will consider adding the mock up