Evil Genius Project Documentation

Project Github link - https://github.com/rohan17398/Evil

1. Introduction

The Evil Genius project is a Python-based data analysis and visualization tool designed to analyze gameplay data from a first-person shooter (FPS) game. The project focuses on processing the game state data and extracting valuable insights related to player strategies, boundary analysis, weapon classes, and CT player positions in specific areas of interest. This document provides detailed information about the project's Python class, ProcessGameState, and demonstrates how it can be used to answer specific questions about Team2's tactics on the T side and CT player positions.

2. ProcessGameState Class

2.1 Class Overview

The ProcessGameState class is designed to handle file ingestion, perform data extraction, transformation, and loading (ETL) operations if necessary, and provide functionalities to analyze the game state data. It utilizes standard Python libraries such as Pandas, NumPy, and Matplotlib to efficiently process the data and minimize dependencies on external libraries.

2.2 Class Methods

2.2.1 __init__(self, file_path)

(1.a from the assessment pdf)

Description: Initializes the ProcessGameState class instance by loading the game state data from the specified file path.

Parameters:
file_path (string): The path to the game state data file.

2.2.2 filter_within_boundary(self, x_min, x_max, y_min, y_max, z_min, z_max)

(1.b from the assessment pdf)

Description: Filters the game state data to include only the rows that fall within the provided boundary.

Parameters:
x_min (float): The minimum x-coordinate of the boundary.
x_max (float): The maximum x-coordinate of the boundary.
y_min (float): The minimum y-coordinate of the boundary.
y_max (float): The maximum y-coordinate of the boundary.

z_min (float): The minimum z-coordinate of the boundary.
z_max (float): The maximum z-coordinate of the boundary.

Returns:
filtered_data (DataFrame): The filtered game state data within the specified boundary.

2.2.3 extract_weapon_classes(self)

(1.c from the assessment pdf)

Description: Extracts the weapon classes from the inventory JSON column in the game state data.

Returns:
weapon_classes (Series): A series containing the extracted weapon classes.

2.2.4 is_common_strategy_used(self, boundary_x_min, boundary_x_max, boundary_y_min, boundary_y_max)

(2.a from the assessment pdf)

Description: Determines whether entering via the light blue boundary is a common strategy used by Team2 on the T side.

Parameters:
boundary_x_min (float): The minimum x-coordinate of the light blue boundary.
boundary_x_max (float): The maximum x-coordinate of the light blue boundary.
boundary_y_min (float): The minimum y-coordinate of the light blue boundary.
boundary_y_max (float): The maximum y-coordinate of the light blue boundary.

Returns:
result (string): A message indicating whether entering via the light blue boundary is a common strategy or not.

2.2.5 avg_enter_time_with_rifles_smgs(self)

(2.b from the assessment pdf)

Description: Calculates the average timer that Team2 on the T side enters "BombsiteB" with at least 2 rifles or SMGs.

Returns:
avg_timer_str (string): The formatted average timer in the format "MM:SS".

This is the output to above two questions -

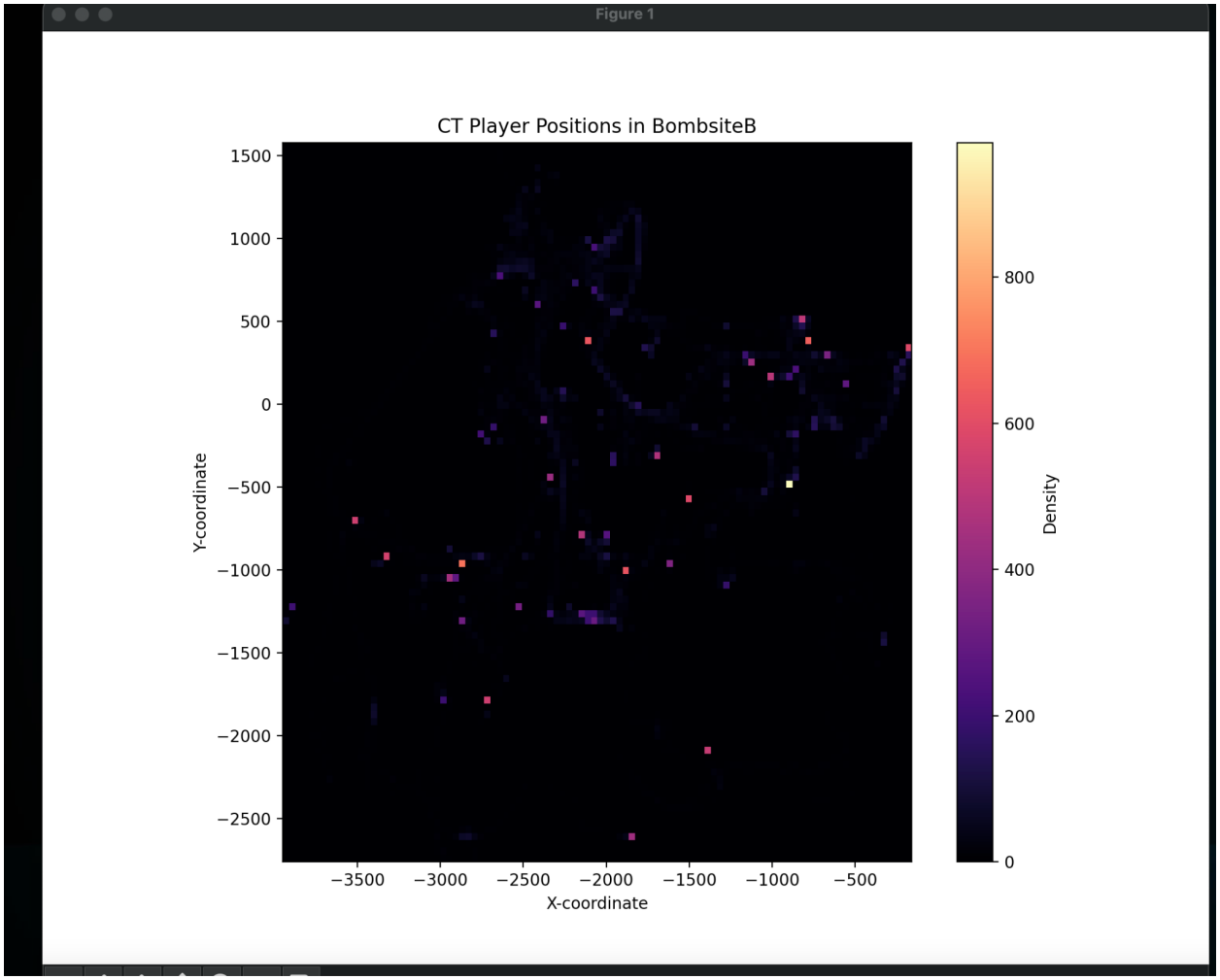
```
→  EvilGenius git:(main) python3 orchestrator.py
Is entering via the light blue boundary a common strategy used by Team2 on T side?  --->  It is not a common stratergy to enter via light blue b
oundary

Average timer that Team2 on T side enters 'BombsiteB' with at least 2 rifles or SMGs --->   1:17
```

2.2.6 heatmap_waiting_spots(self)

(2.c from the assessment pdf)

Description: Generates a heatmap of CT player positions inside "BombsiteB" based on the game state data.

The following below is the heatmap generated from this function.



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:

a. could allow our coaching staff to request or acquire the output themselves
b. takes less than 1 weeks worth of work to implement

To facilitate the engagement of our stakeholders, specifically the CS:GO coaching staff, and empower them to access and request the output of the Evil Genius project, I propose the development of an interactive web-based dashboard. This solution aims to provide a user-friendly and accessible interface, enabling the coaching staff to interact with the project's functionalities without requiring any technical expertise or coding skills. The implementation can be achieved within a week, offering the following benefits:

Intuitive User Interface: The web-based dashboard will feature a graphical user interface (GUI) that ensures ease of use. The coaching staff can effortlessly navigate through the dashboard, leveraging familiar controls and input options.

Visual Data Representation: By harnessing the plotting and visualization capabilities of libraries such as Matplotlib and Seaborn, the dashboard will present the analysis output in visually appealing charts, graphs, and heatmaps. This facilitates the comprehension of insights related to Team2's strategies, average timers, and CT player positions.

Flexible Customization: The dashboard will incorporate interactive elements like input fields and dropdown menus, empowering the coaching staff to customize the analysis parameters. They can define boundary coordinates or select specific game scenarios, tailoring the analysis to their unique requirements and obtaining relevant insights.

Export and Download Functionality: To facilitate further analysis or presentation needs, the dashboard will include options to export the generated analysis results in formats such as PDF or CSV. This capability allows the coaching staff to download and share the output conveniently.

Real-time Updates: If the game state data is regularly updated, the dashboard can be designed to fetch the latest information and perform real-time analysis. This ensures that the coaching staff receives up-to-date insights, enabling informed decision-making.

Implementation Timeline:

To meet the one-week implementation timeframe, we will adopt an agile development approach, adhering to the following schedule:

Day 1: Requirement Gathering: Engage with stakeholders to gather comprehensive requirements, including desired features, visualizations, and customization options.

Day 2: Design and Wireframing: Create a visually appealing design and wireframes for the web-based dashboard, emphasizing an intuitive user interface, seamless navigation, and effective visualization of analysis output.

Day 3: Front-end Development: Implement the front-end components of the dashboard using web technologies such as HTML, CSS, and JavaScript. Incorporate interactive controls, input fields, and visualizations as per the design specifications.

Day 4: Back-end Development: Develop the back-end functionality of the dashboard using a web framework like Flask or Django. Implement data processing, analysis functions, and seamless communication with the game state data.

Day 5: Integration and Testing: Integrate the front-end and back-end components, ensuring smooth communication between the dashboard and the underlying analysis code. Conduct comprehensive testing to identify and resolve any issues or bugs.

Day 6: Deployment and Documentation: Deploy the web-based dashboard to a server or hosting platform, making it accessible to the coaching staff. Prepare comprehensive documentation and user guides to facilitate their usage of the dashboard.

Day 7: User Training and Feedback: Conduct training sessions with the coaching staff to familiarize them with the dashboard's features and functionality. Gather their feedback to incorporate any usability improvements or additional functionalities.

By adhering to this timeline, we can successfully develop and deploy the interactive web-based dashboard within a week. This solution will enable the CS:GO coaching staff to effortlessly request and acquire the output of the Evil Genius project, empowering them to make informed, data-driven decisions in their coaching endeavors.