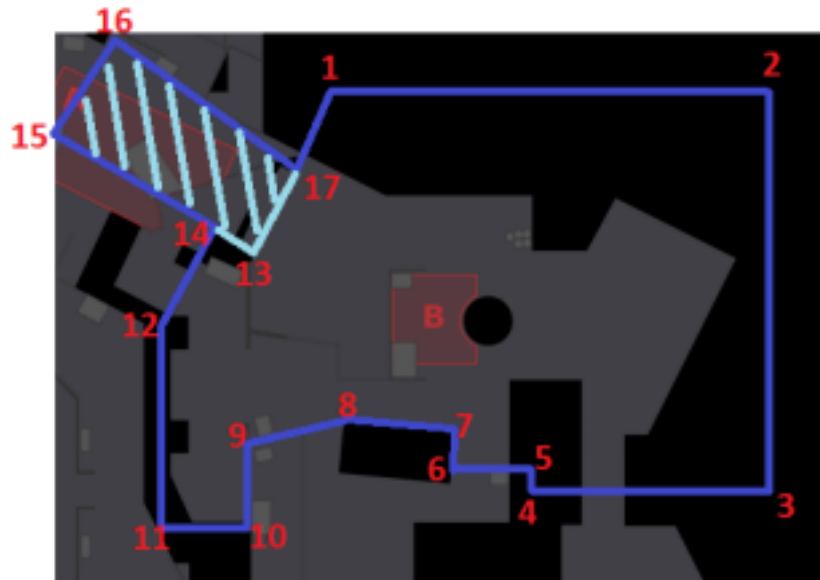


## Task

Your CounterStrike coaching staff is analyzing their opponent's strategies on *BombsiteB*. They have identified a chokepoint of interest (which has been highlighted in **light blue**), and there are a few player location questions that they need to address regarding it.



The coordinate details are as follows:

- Z-Axis Bounds:  $285 \leq Z \leq 421$
- 13:  $[-1735, 250]$
- 14:  $[-2024, 398]$
- 15:  $[-2806, 742]$
- 16:  $[-2472, 1233]$
- 17:  $[-1565, 580]$

## Provided

- Parquet file containing player data during a CounterStrike match.
- Image of radar/map
- (x,y) coordinates of boundary vertices (page 1)
- Data Dictionary (page 3)

You, being the forward-thinking engineer that you are, know that this type of request will be asked multiple times for different sets of coordinates and games. Let's engineer a flexible solution for our team to use.

1. Write a **python** class called *ProcessGameState* that will accomplish the following:
  - a. Handle file ingestion and ETL (if deemed necessary)
  - b. Return whether or not each row falls within a provided boundary
    - i. Be as efficient as possible (minimize runtime)
    - ii. Reduce dependencies outside of standard Python libraries unless you can prove significant improvements.
  - c. Extract the weapon classes from the inventory json column
2. Using the created class, answer the following questions:
  - a. Is entering via the light blue boundary a common strategy used by Team2 on T (terrorist) side?
  - b. What is the average timer that Team2 on T (terrorist) side enters "BombsiteB" with least 2 rifles or SMGs?
  - c. 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"
    - i. Hint: Try a 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:
  - a. could allow our coaching staff to request or acquire the output themselves
  - b. takes less than 1 weeks worth of work to implement

## Submission Instructions + Materials

1. Set up a GitHub account if you do not have one
2. Create a new repository where you will submit your code. You will be sharing this repository with your interviewer(s)
3. Make sure all code is pushed prior to the deadline (anything pushed after will not be reviewed)
4. Please provide the class, script, and outputs/interpretations in your repo
5. Submit the link of this repository to the submission form:  
<https://forms.gle/xVJLf9vKsPgZwvL6>

## Data Dictionary

Each row represents a player's game state per frame (tick).

Attribute	Description	Type
round_num	round	int
tick	time unit (128 ticks per second)	int
side	T (terrorist) or CT (counterterrorist) that team is playing	str
team	team	str
hp	current health points	int
armor	current armor value	int
is_alive	if player is alive or dead	boolean
x	x coordinate	int
y	y coordinate	int
z	z coordinate	int
inventory	what the player is carrying and equipment details	dict/json
total_utility	number of utility (grenades, flashes, smokes, inced) in inventory	int
equipment_value_freezetime_end	value of equipment after purchase time ends	int
area_name	location that the player is	str
seconds	elapsed time (resets to zero after bomb plant)	int
clock_time	in-game clock timer (resets to 0:40 after bomb plant)	str
t_alive	number of T alive	int
ct_alive	number of CT alive	int
bomb_planted	if the bomb has been planted yet	boolean
map_name	map being played	str
utility_used	number of utility used	int
player	player name	str