**PROJECT OVERVIEW - CS 1.6 GUNGAME MODE**

CONTEXT

Counter-Strike (publicly known as Counter Strike 1.6) is a multiplayer first-person shooter video game (either LAN or online).

The game has two teams that allow you to have a good experience in the game, these are the anti-terrorists and terrorists. The main objective in the game is to take out the enemy (Terrorists) as well as rescue hostages, prevent them from planting an explosive and in case it has already been planted, defuse it.

Over the years, the game has seen a lot of modifications, one of which is the GunGame (GG) modification. GunGame is a game mode where players progress through a series of weapons by getting a kill with each one. It's a fast-paced and exciting game mode that is popular among Counter-Strike players.

In GunGame, each player starts with a basic weapon, usually a pistol. The objective of the game is to progress through the weapons by getting a kill with each one. When a player gets a kill with a weapon, they automatically advance to the next weapon in the list. The weapons in the list are usually arranged in order of difficulty, with the more challenging weapons coming later in the list.

The game continues until a player gets a kill with the final weapon in the list, which is usually a knife or grenade. The player who gets the final kill wins the game. It's a simple yet exciting game mode that can be played with any number of players.

PROJECT SUMMARY

This project deals with the extraction(daily), transformation and manipulation of data from a web page of a leaderboard of a Counter-Strike 1.6 game server (Game mode: Gungame). This data is then used for the development of a new leaderboard derived from the original one, generating a new ranking system based on statistical calculations with the purpose of helping in-game decision-making based on the resulting data, such as team building for in-game tournaments, analysis and improvement of individual in-game performance, analysis and identification of player behavior patterns, etc.

TOOLS USED

**Python**: ETL - Extraction(Web Scrapping with BeautifulSoup), transformation(Pandas) and loading(Sqlite3).

**SQL**: Sqlite Database for storage, query development and data modelling.

**Windows file system** for file back-ups and documentation.

**Windows Task Scheduler** for process automation.

PROJECT WALKTHROUGH

ETL: Only one Python script is in charge of doing all the ETL processes. It runs daily at 12:00 AM on my personal Windows computer (I used Windows Task Scheduler for this). The script extracts the raw data from the corresponding web page(s), cleans it, backs it up locally into a file system and ingests it through a .csv file into an SQLite database, also located locally in the same directory as the script.

Data is ingested into a database table which will contain the raw data with no transformations.

CREATE TABLE "**raw\_gg\_teamplay\_players**" (

    "POSITION"  TEXT,

    "CS\_NICK"   TEXT,

    "POINTS"    TEXT,

    "GAMES\_WON" TEXT,

    "KILLS" TEXT,

    "ASSISTS"   TEXT,

    "DEATHS"    TEXT,

    "HS"    TEXT,

    "HITS"  TEXT,

    "SHOTS" TEXT,

    "ACCUARACY" TEXT,

    "DAMAGE"    TEXT,

    "TIME\_PLAYED"   TEXT,

    "fecha\_proceso" TEXT

)

Data from the web page(s) is updated in real-time, so the data ingested into the database corresponds to the previous day's batch, with the latest snapshot of the position table.

“fecha\_proceso”: This is a piece of data that the script creates at the time of ingestion. It is the date on which the data has been processed by the script.

A screenshot of a computer

Description automatically generated with medium confidence Raw table sample data

Then the same script executes a sqlite query that takes care of cleaning (transforming) the raw data already ingested into the database and inserting it into a new table:

INSERT INTO cur\_gg\_teamplay\_players(position,nickname,points,games\_won,kills,assists,deaths,kda,headshots,hits,shots,shot\_acc\_ratio,headshot\_acc\_ratio,dmg\_done,dmg\_per\_hit,days\_played,hours\_played,minutes\_played,fecha\_proceso)

SELECT

**CAST**(raw.POSITION AS INT) AS position,

    raw.CS\_NICK AS nickname,

**CAST**(**REPLACE**(raw.POINTS,'.','') AS INT) AS points,

**CAST**(raw.GAMES\_WON AS INT) AS games\_won,

**CAST**(**REPLACE**(raw.KILLS,'.','') AS INT) AS kills,

**CAST**(**REPLACE**(raw.ASSISTS,'.','') AS INT) AS assists,

**CAST**(**REPLACE**(raw.DEATHS,'.','') AS INT) AS deaths,

**ROUND**((**CAST**(**REPLACE**(raw.KILLS,'.','') AS REAL)+**CAST**(**REPLACE**(raw.ASSISTS,'.','') AS REAL))/**CAST**(**REPLACE**(raw.DEATHS,'.','') AS REAL),2) AS kda,

**CAST**(**REPLACE**(raw.HS,'.','') AS INT) AS headshots,

**CAST**(**REPLACE**(raw.HITS,'.','') AS INT) AS hits,

**CAST**(**REPLACE**(raw.SHOTS,'.','') AS INT) AS shots,

**CAST**(**REPLACE**(**REPLACE**(raw.ACCUARACY,',','.'),'%','') AS REAL) AS shot\_acc\_ratio,

**ROUND**((**CAST**(**REPLACE**(raw.HS,'.','') AS REAL)\*100)/**CAST**(**REPLACE**(raw.HITS,'.','') AS REAL),2) AS headshot\_acc\_ratio,

**CAST**(**REPLACE**(raw.DAMAGE,'.','') AS INT) AS dmg\_done,

**ROUND**(**CAST**(**REPLACE**(raw.DAMAGE,'.','') AS REAL)/**CAST**(**REPLACE**(raw.HITS,'.','') AS REAL),2) AS dmg\_per\_hit,

    CASE WHEN INSTR(raw.TIME\_PLAYED,'días') != 0 OR INSTR(raw.TIME\_PLAYED,'día') != 0 THEN **CAST**(**TRIM**(SUBSTR(raw.TIME\_PLAYED,1,2)) AS INT) ELSE 0 END AS days\_played,

    CASE

        WHEN INSTR(raw.TIME\_PLAYED,'horas') != 0 OR INSTR(raw.TIME\_PLAYED,'hora') != 0 THEN

        (CASE WHEN INSTR(raw.TIME\_PLAYED,',') != 0 THEN **CAST**(**TRIM**(SUBSTR(raw.TIME\_PLAYED,INSTR(raw.TIME\_PLAYED,',')+1,3)) AS INT) ELSE **CAST**(**TRIM**(SUBSTR(raw.TIME\_PLAYED,1,2)) AS INT)END)

        ELSE 0

    END AS hours\_played,

    CASE WHEN INSTR(raw.TIME\_PLAYED,'minutos') != 0 OR INSTR(TIME\_PLAYED,'minuto') != 0 THEN **CAST**(**TRIM**(**REPLACE**(SUBSTR(raw.TIME\_PLAYED,-10,3),'y','')) AS INT) ELSE 0 END AS minutes\_played,

**CAST**(raw.fecha\_proceso AS INT) AS fecha\_proceso

FROM raw\_gg\_teamplay\_players AS raw

LEFT JOIN cur\_gg\_teamplay\_players AS cur

ON

    raw.CS\_NICK = cur.nickname AND

**CAST**(**REPLACE**(raw.POINTS,'.','') AS INT) = cur.points AND

**CAST**(**REPLACE**(raw.KILLS,'.','') AS INT) = cur.kills AND

**CAST**(**REPLACE**(raw.ASSISTS,'.','') AS INT) = cur.assists AND

**CAST**(**REPLACE**(raw.DEATHS,'.','') AS INT) = cur.deaths

WHERE raw.fecha\_proceso IN (SELECT **MAX**(fecha\_proceso) FROM raw\_gg\_teamplay\_players)

AND (cur.points IS NULL OR cur.kills IS NULL OR cur.assists IS NULL OR cur.deaths IS NULL)

;

This script has a logic applied so that only new items are inserted in this new table, thus avoiding ingesting repetitive data from players who have not played the previous day or in previous days.

Sample of transformed data:

A screen shot of a computer

Description automatically generated with low confidence

As you see, this leaderboard is ranked on player points, which players gain when they kill, assist or win a game. The data shows that the player with more time played than other players tends to have a better position than those players. (Like player in position 1 that has almost doubled the time played and points of the top 5 players).

At this point, the data is ready to be used for the purpose of the project.

ABT: An analytical base table is made by querying the data in sqlite, showing the rating of each player based on a calculation(defined by me) using the already transformed data. This calculation uses player performance with the purpose of showing off the impact that each individual player has.

rating = (*shot\_acc\_ratio*\*0.25)+*headshot\_acc\_ratio*+(*kda*\*10)+(*wph*\*3)+*dps*+(*points\_per\_min*\*0.10))/10

shot*\_acc\_ratio*: Shot accuracy ratio, how many effective shots(hits) over 100 shots.

*headshot\_acc\_ratio:* Headshot accuracy ratio, how many headshots over 100 hits.

*kda:* (Kills + Assists) / Deaths.

*wph:* How many matches in average the player has won per hour.

*dps:* Damage per second.

*points\_per\_min:* Points per minute.

ABT Sample Data:

A screenshot of a computer

Description automatically generated with medium confidence

At this point, ABT show’s a new leaderboard and a new ranking system base of player performance instead of player points as it was before.