**Project 3 Proposal:**

**Data Visualisation Track:**

Topic: (Custom) 🡪 Steam Games Dataset analysis + visualisation

**Source files:**

1: Kaggle.com, **Martin Bustos, Steam Games Dataset**

url: <https://www.kaggle.com/datasets/fronkongames/steam-games-dataset>

Licence: MIT (url: <https://www.mit.edu/~amini/LICENSE.md>)

Hereby according to the author said: All data has been collected thanks to the Web API provided by Steam (Steam Spy). Only games (no DLCs, episodes, music, videos, etc) currently released have been added.

2: Kaggle.com, **Nik Davis, Steam Store Games (Clean dataset)**

url: <https://www.kaggle.com/datasets/nikdavis/steam-store-games>

Licence: Attribution 4.0 International (CC BY 4.0) (url: <https://creativecommons.org/licenses/by/4.0/>)

Hereby according to the author said: data gathered via using Steam Store and SteamSpy API.

In the first url, please only download the ‘**games.csv’** file. In the second url, please only download the **‘steam.csv’** file. As I merge two of the csv files together for this project.)

**Reason for choosing this topic:**

I am interested in providing some analysis and visualisation of the world most widely used PC game platform: Steam. (Which is a cross platform webpage & software that enable users to purchase, download, share, create, and publish game/videos/comments/or even real trading using literal currency.)

In this project, the focus is on visualising game data.

Here are the questions that will be analysed and visualised:

1: what are the 5 most popular games from 2020 to 2024?

2: what are the relationship between user review (positive or negative) and games’ price?

3: Which publishers have the highest positive ratings?

4: Do games with certain genre or descriptors tend to be more popular?

**For all the questions above, the approach is below:**

1: use pandas to read csv & perform data cleaning ->

2: save the cleaned dataframe and output it, create tables using the appID as foreign keys to create schema and load data to Postgres (this can be done either manually / using sqlalchemy + psycopg2) ->

3: read data & perform visualisations (propose the NEW library for plotting: Seaborn), save all the static plots using .savefig().

4: create flask API for all the plots above, then user can view all static charts via visiting urls.

5: Include a html page for dynamic plotting, done by using D3.js and plotly.js, add dropdown menu that when user select different menu the page shows different graphs.

References:

Martin Bustos Roman. (2022). Steam Games Dataset [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DS/2109585>

Nik Davis. (2019). Steam Store Games (Clean dataset) [Data set]. Kaggle. <https://www.kaggle.com/nikdavis/steam-store-games>