

CSC111 Project Report: Steam Video Games Recommendation and Interactive User Interface Design

Kevin Cai, Junsong Guo, Yiteng Zhang, Patrick Zhou

Wednesday, April 14, 2021

Problem Description and Project Goal

Video games provide people a way to temporarily escape reality and distress. Playing video games is fun but finding the right games to play might take a lot of time and be a gruesome task to some people. Many who don't like the trouble of searching for games simply stick to the few games they already play and are afraid to give new games a try. This causes many good games to be underrated and underappreciated. This is especially true on Steam, one of the largest digital distribution platforms that most people use to download and buy games.¹ Steam currently has more than 40,000 video games and over 10,000 of these games were just added to it last year.² The number of games being added to Steam each year is increasing rapidly, and finding a good game will only take more and more effort and time. Although Steam already offers some ways to recommend people games that are currently trending or games that are highly rated, Steam still does not offer efficient ways for people to just simply find games that are similar to what they have already played and liked. Many who use Steam are young adults who have work and/or school, so they may not have the time to browse through the overwhelming amount of games on steam just to find that one game to play.

So, our project goal is to implement an interactive GUI (Graphical User Interface) that users can use to get recommendations for video games that are on the Steam platform. The recommendations are selected from similar games to the games the user inputs, or selected from the top rated games based on the given genre of games. This program aims to help the user find quality games easily, which in turn will save users a lot of time and effort that can be used to actually play these games and have fun instead.

Dataset description and computational overview

Description of the kinds of data our project use to represent our chosen domain, and how trees and/or graphs play a central role in this data representation:

The original CSV file contains information about the names of games, user reviews, categories, developer, publisher, and price from the past decade.(steam_data.csv) We may filter some variables that do not really aim to suggest the user a new game such as publisher and developer.

Each row of the original CSV file contains 13 entries. However, we may just keep the TOP four important variables: **categories**, **price**, **all_reviews(rating score)** and **URL**, which navigates the users to the corresponding steam page of the game.

- Data Source: <https://www.kaggle.com/deepann/80000-steam-games-dataset?select=steamdata.csv>
- Data zip file: <https://drive.google.com/file/d/1mFJCYoajWEn2Jy2Q2TMuKYjNM9xFVEXP/view?usp=sharing>

Description of the major computations our program performed:

¹Bolding, J. (2019, January 13). Steam now has 30,000 games. Retrieved from <https://www.pcgamer.com/steam-now-has-30000-games/#:~:text=The total does not include,total of over 30,000 games.>

²Clement, J. (2021, February 01). Number of games released on Steam 2020. Retrieved from <https://www.statista.com/statistics/552623/number-games-released-steam/>

First of all, we design and implement a function that can read in the CSV file and creates the corresponding Graph object according to the above description. Our goal is to take this data and represent it using our Graph class, where:

- Each vertex stores the item information, it's kind (game, category, or publisher), and its neighbours. For game vertices, its item is a tuple with three elements, game name, price, and user rating. For category and publisher vertices, the item just stores the name of that category or publisher.
- There is an edge between a game category vertex and a category or publisher vertex if and only if the game is in that category or is made by that publisher. We could “augment” each edge so that it keeps track of this information.
- In order to find similar games given a single game, the program calculates the similarity score of all games in the graph with the given game, like assignment 3. Then the games will be ranked from the highest similarity score to the lowest, and the top 20 games will be returned.
- In order to find the top rated games given some categories, the program finds all the games that share the categories. Then return the top 20 games ranked by their user rating, which is stored in the game's item.

Explanation of how our program reports the results of your computation in a visual and/or interactive way:

To present the results of our program to the user, we plan to use a python library called “tkinter” that provides relatively simple access to GUI elements such as search bars and buttons. This library allows us to implement those features using pure python code, with no need for extra front-end languages such as html, JavaScript and so on. Our GUI will include interactive buttons representing different game categories. As the user clicks on them, a list of overviews of recommended games including game title, price, as well as image will appear on the screen.

Explanation of how our program uses new libraries to accomplish its tasks:

- We used the library “csv” to help read in the dataset, which is a csv file.
- We used the library “tkinter” to implement the GUI using python code.
- We used the library “webbrowser” to help make it convenient for users to browse the detailed information of the game on the steam platform, providing the url of each game in the dataset.

Description of any changes to our project plan between our proposal and final submission

- Instead of using the ‘remi’ python library to implement our GUI, we used another library ‘tkinter’, which has more tutorials and resources online.
- Compared with the content in the project proposal, we did not use the “user review” column in the dataset.
- We added a new feature on the list of recommended games. A link to the steam page of the recommended game is added to the end of the each row so the users can view the information of the games directly.

Discussion section

Results:

In our interactive GUI, users can get their game recommendations from two modes: by the similarity between the game typed in or by categories they choose. In this way, our project goal will be solved.

Limitations:

- In our GUI, users need to type in the exact same name of the game. The letters typed in need to be the same as the name of the game in the dataset. This will cause problems. If there are characters that are difficult to type in the name of a game or the user makes an mistake in typing in the name of the game, a “Game not found in our data base :(” error will occur. This is the most inconvenient problem for users.

- Another problem is due to the restriction of the size of the dataset. If the game is not in the dataset, then our program will not be able to recommend it.

Obstacles:

- We defined a method called `_recom_frames()` in the visualization part of our project. However, we cannot solve the python ta error saying “Optional[Union[]] does not support indexing” occurred.
- Another problem is due to the restriction of the size of the dataset. If the game is not in the dataset, then our program will not be able to recommend it.
- We have also discovered the occurrence of adaptability problems in different systems. When designing the GUI, the colors could not be displayed in another computer system. This problem also took us a lot of time.

Next steps for further exploration:

According to the most inconvenient problem we discussed above, we can improve our game recommendation system by improving the algorithm. When the user enters the game name, it predicts what they will enter and provides the user with several possible results for them to choose from. We believe that will make the system more user-friendly.

References

Bolding, J. (2019, January 13). Steam now has 30,000 games. Retrieved from <https://www.pcgamer.com/steam-now-has-30000-games/#:~:text=The total does not include,total of over 30,000 games>.

Clement, J. (2021, February 01). Number of games released on Steam 2020. Retrieved from <https://www.statista.com/statistics/games-released-steam/>