

Proposal for Project SteamSuggestion Conferences

Andy Liu, Jacob Samar, Yucheng Ma, and Yiming Sun

Abstract—Our project is to create an algorithm that will effectively recommend steam games based on different properties of the games gathered from databases online. Our project should take in some user information (game ratings, price range, games played, time spent playing a game, genre, etc.) and effectively recommend other steam games to the user.

I. INTRODUCTION

With the gaming industry only continuing to grow, more and more games are produced every year. With such a wide range of options, It is difficult to choose which game to play. Videos games are advertised and played on centralized platforms, one such platform being Steam. Steam is the largest PC game provider, and many different types of games developers sell their games on Steam, from big companies, even small independently operated studios. In May 2019, Steam had a total of 27,033 video games. With that many games, players will have a hard time finding the best choice for themselves, especially for games that seem appealing but are expensive. Customers are forced to buy games blindly (which is a time and money investment) in the hopes that they managed to pick one that they enjoy. Our project, SteamSuggestion, is designed to help players make the right choices on their gaming purchases. SteamSuggestions will filter all games on Steam based on each game's attributes such as price, genres, time investment, game intensity, etc.. Players can use SteamSuggestion's filter to shorten the area of search and then they can decide the final choice of game.

II. MARKETING ANALYSIS

PC gaming is centralized on several large platforms. For our purposes, Steam will be the platform we target. Steam was launched in 2003 and is a distribution platform owned by Valve Corporation. Initially it was used by Valve to push out updates to their games automatically, but grew to accept games from 3rd party publishers. Today, Steam is the largest digital distribution platform for PC gaming. By 2017, users purchasing games through Steam totaled roughly \$4.3 billion USD, representing at least 18% of global PC game sales. By 2019, the service had over a billion registered accounts with 90 million monthly active users, and here are more than 27,000 games on the Steam platform. Due to the popularity of video game trends, the video game industry is developing rapidly. In the future, more games will publish on Steam and more new players will use the Steam platform to pick and download their games. Because of this fierce market growth, game developer companies have increased their investment to make exquisite games. Unfortunately, this

increase in production costs of the game directly increases the price as well. In our dataset, there are least 1,000 games that cost more than \$19.99. The dilemma for many video game players is that there are too many games that look interesting, but buying more than one is too expensive of an option. Even if they could, trying a game takes hours, and no one wants to waste their own time. What the current Steam store offers to players rarely is enough for players to parse all their options. Players are eager to have an independent filter software that will more effectively find them games that they enjoy. SteamSuggestion will help users find a game that will not waste the players' time or money. Many people will enjoy SteamSuggestion because there are not many Steam filter out there. By making SteamSuggestion, players can effectively waste less time finding, and more time playing.

III. DATA SET INFORMATION

Our initial dataset includes multiple useful fields including: name, release date, English, required age, categories, platforms, genres, rating, playtime and price. We attempt to use these different fields to develop a collaborative filtering engine, which is designed to recommend the same games to similar users or recommend a similar game to one a user has liked. We will also use genres, categories, platforms, English, required age, play times, and price as optional user inputs to filter down the results. For example, if users don't want to spend too much time on the game, we can generate a game list based on the average playtime. The rating of the game will initially have the largest weight of the recommendation engine. The games that contains more positive ratings will always on the top of the list. Should we accomplish using the data in this way, we could potentially find other, complementing datasets to augment how we recommend titles. This could include more information Steam's public data or scraped data from the web.

IV. EXPECTED OUTCOME

Our end product should be an application/API that will recommend games to a user based on their inputs. Our program will contain a dataset of Steam games, with each one categorized. Based on these categories, users can filter what type of games they want to play. The category will include prices, required age, playtime, etc. The more filters input by the user the more focused and precise the resulting list of games will be. Our program will help users find the right game(s) for them and save them time and money. The application will come with a display panel that allows users click on it. Through the display panel, the users can then select the categories that interest them. Finally, the display

*This work was not supported by any organization

panel will pop up the result list. This result list will be an image of the game, a description of the game, a rating, and a direct link to the steam store page. The application GUI will be written in pyqt, and the backend will be written in python. The GUI will send user's selection to the backend code, and the backend code will start filtering. Afterward, it will send the result to the GUI, and the GUI will display and beautify the result. If time allows, the program can rank filtered games based on playtime, prices, rating, etc.

V. TEAM ROLE

The team for SteamSuggestion is consist of Andy Liu, Yiming Sun, Jacob Samar, and Yucheng Ma. The main roles are team manager, GUI coders, and backend coder. Team manager will try to keep everything together and make sure nothing go wrong. GUI coder work on the GUI. Blackend coders work on the backbone of the program. They are the most important role, and even GUI and team manager will need to help them if needed. Andy will be the team manager; he will work to keep everyone working on schedule and meeting deadlines. Also, he will verify program function, including QA before any due dates. In addition, he will help Yiming with GUI. Yiming will work on display panel, menu, clickable button, etc. Jacob and Yucheng work on the back end of the program. This will include parsing and storing the data, developing how each game is categorized, and creating the filtering algorithm that will recommend games based on user input. Everyone on the team will be presenting, and each of us will work on each other parts if one of us are lagging behind.

VI. FIGURES AND TABLES

TABLE I
TIME LINE

9/30	Meeting and start on project
10/5	Meeting and discuss ways to implement
10/7	Finalize how to implement
10/11	Start working on the code
10/11 10/15	Work on project(coding for GUI and backend)
10/16	Meeting and progress check(check on coding progress)
10/16 10/20	Work on project(coding for GUI and backend)
10/23	Meeting and progress check(check on coding progress)
10/23 10/29	Work on project(coding for GUI and backend)
10/29	Meeting and progress check (dataset categorized in backend)
11/1	Meeting and progress check (check on GUI)
11/2 to 11/18	Finishing GUI and the back-end coding
11/30	Combine GUI and back-end coding
11/31	From now to presentation: Debugging the program
TBA	Prepare for presentation
TBA	Present project to class