

Project Title: GameFinder

Project Summary:

Our video game search engine, GameFinder, is designed to help users find the perfect game for their preferences and computer specifications. With GameFinder, users can input specific characteristics they are looking for in a game, such as genre, setting, or playstyle, and their computer specs, such as graphics card and RAM. The search engine will then generate a catalog of games that match the user's criteria, ensuring that they find a game that not only fits their interests but also runs smoothly on their computer. Whether you're looking for a fast-paced action game or a relaxing puzzle game, GameFinder has you covered. Say goodbye to endless scrolling through game listings and hello to a tailored, efficient search experience. It's always game time.

Description:

Our video game search engine, GameFinder, aims to solve the problem of finding the right game for a player's preferences and computer specifications. With the increasing number of video games being released every year, it can be overwhelming for players to sift through all the options and find a game that suits their interests and runs smoothly on their computer. GameFinder streamlines this process by allowing players to input specific characteristics they are looking for in a game and their computer specifications, and then generating a catalog of games that match their criteria. The result is a personalized and efficient search experience, making it easier for players to find the perfect game for them. Our goal is to provide a convenient solution for players who want to enjoy the latest games without having to spend hours searching for the right one.

Details About Data:

Game Finder, will use the “Steam Game” dataset. The dataset itself includes 78 columns with different information about games within the Steam platform. These features range from release dates to specific specs about the game. The dataset also includes 13,358 entries. These entries contain the various games on Steam that can be downloaded/ played.

The specific dataset can be found at  games-features .

Link:<https://docs.google.com/spreadsheets/d/1m10Zkm07WIkGzwi36it0g4Fkuw5xx5Q68hCinRHxIx8/edit#gid=1578928233>

With this database, we plan to highlight different features of games currently available on the Steam platform.

Detailed Functionalities:

Filtering games based on provided preferences:

This is the most important function of this application. We will provide dropdown menus, check boxes, or text input fields for users to select their preferences easily. We can then use these inputs to filter out irrelevant games.

Our categories should include:

- Game genre. (This is the most important feature. For this category, we can use check boxes so that users can select multiple genres for more specific filtering.
- **Platform.** (e.g. Windows, MacOS, PS5...)
- **Language.** (English, Spanish, etc.)
- **Content Ratings.** (e.g. PG, R, ...)
- **Price range.** (We can use text input here for convenience.)
- **Minimum requirements.** (Instead of asking users to look up the requirements for specific games, we offer this filter so that they can find games that can run on their computer directly.)

Searching specific games by name:

As a powerful game searching engine, we will offer ways to search games based on the given text. We will try to implement vague search if possible.

Sort outputs based on different algorithms:

After users create their own query, we should display the games in a list. And order matters! So we can either sort the list based on real players ratings/reviews, or display games randomly to help discover more games in this world.

The web application will also have **3-key** user interfaces:

1. **Super User:** Similar to an admin user, this will be a user who has many capabilities. Simple functions that they have will include adding/ deleting games that are no longer available on Steam. Their more complex capabilities include being able to look at registered users. This interface will allow them to see information about the user and allow them to delete inactive users.
2. **Registered User:** Registered users will be referred to as individuals who create an account using our web application. This will allow for users to save preferences about games they enjoy, star those of interest to them, and more complexly save inputs regarding the specs of their computer. This will allow them to more easily “pick-up where they left off” and scroll through games of interest to them. This will be done using filters of game release dates, genre, et cetera.

3. **Guest:** For users only interested in one or two time uses on our web application, this mode offers all the same functionalities and capabilities of a registered user but without saved memory. This essentially means users inputting specs about their laptops won't be saved if they refresh the page or come back to the site later. Similarly, they would not be able to favorite games of interest.

Real-World Applicability

The original dataset for the Steam Game dataset is located at <https://data.world/craigkelly/steam-game-data>, which is a “combination of data from publicly available Steam API's and steamspy.com”.

From the [about](#) page of steamspy.com, we learned that “Steam Spy automatically gathers data from Steam user profiles”. Based on the methods described in the post (<https://arstechnica.com/gaming/2014/04/introducing-steam-gauge-ars-reveals-steams-most-popular-games/>), we learned that all these data comes from real life steam users.

Our web application, GameFinder, builds on the concepts built in this data set by adding an analyzation tool. More specifically, many of the functionalities of our application revolve around providing our users curated information from the larger dataset. Whether that be games they can play on their machine or games of genres they more closely relate to, it adds a seamless connection between the raw set of games on Steam and the platform Steam itself where users can download games but may not have the best insight on.

Creative Component:

Having a subpar computer/ laptop can often turn people away from playing a certain game. In fact, in recent years, it has become more standardized for people who enjoy playing games to buy more and more expensive equipment to run the games they most enjoy playing. While this is a nice luxury for some with higher recreational budgets, it leaves others behind. One creative component that we hope to add to our web application is a small survey set that allows users to input specs about their computer. Our web application will use this data and compare it against games on Steam that support their specific specs and return those. We plan to achieve this by adding key specs that games often do or do not support and inputting those into game availability.

While the applicability may seem insignificant at first glance, this is a large step towards highlighting new games to users with less great specs and even potentially grow popularity for less played games. This feature would in turn benefit both users and game developers to grow a better community for different games on different platforms.

Low-Fidelity UI Mockup:

GameFinder

Login

Search Game or Keyword

filter

Genre ▾

Platform ▾

Rating ▾

Price ▾



Random Ad

Featured:

Game 1

Game 2

Game 3

Game 4

New in 2023:

Game 5

Game 6

Game 7

Game 8

Throwbacks:

Game 9

Game 10

Game 11

Game 12

Add Games

Contact Us

(Subject to change during course of semester)

Rahul Bhatt: Creating the CRUD (Create, Read, Update, Delete) interface, helping the user have a personalized and efficient search experience

Sam Yue Chen: Database programming in SQL (triggers, selecting relevant data, etc.) to help organize data for backend of the web app

Navin Ranganathan: Developing the backend of the web application, organizing all the specifications for helping users find their games.

Nimish Mathur: Helping implement the search feature of our web app, allowing users to type in their game name or other keywords to streamline the users' search process.

Other Notes: Our team has made a group chat and use that as our prime way of communication. We also have implemented checks to help keep each other on schedule for deadlines. This has been worked out so Rahul and Nimish have more overlap with the work related to user interface. Same goes for Navin and Sam who have more support with each other for the back-end.

With these two sets, we hope to have code reviews with the other as a reviewer to make sure all code pushed works as intended!