

# Coursework 1 - Games Catalogue

David Frame

40200819@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

## 1 Introduction

In this report, I will discuss and evaluate my games catalogue web app, suggest improvements for the future and look at how I performed when creating this application.

To summarize, the games catalogue allows users to browse a list of games from the home page (see figure 1), or filter them by various categories such as genre or platform. After choosing a category and a specific filter, such as year - 2017, the user will see a list of games matching the filter conditions. Clicking a game shows the game's details, with the details clickable to find similar games. Users can add games to the data set by using the 'add game' screen.

## 2 Design

When designing the URL hierarchy for this web app, I considered how a user would search for a game and how they might refine their searches.

I quickly realized that structuring the URL hierarchy to be completely linear didn't make sense, as I would have to make some filters have priority over others - for example, searching genre and then year of release.

Doing the hierarchy this way would restrict users who, for example, wanted to see all the games released in a particular year without filtering them by genre first.

To remedy this, I designed the URL hierarchy to allow the user to search by any category from the start - with the downside of not being able to filter by multiple things at once.

See figure 2 for a diagram of the structure. The root (/) by itself will redirect to the games page (which shows all games and acts as the home page). Each category can be accessed with /category, for example /genre. The filter option can be chosen using /category/filter, for example /genre/Strategy. The add game page is accessed from /addgame.

## 3 Enhancements

There are various ways that this application could be improved, I have highlighted some of the improvements I would make below.

### 3.1 Allow attributes to be lists

Many games are released on multiple platforms, others span more than one genre or fit into a particular sub-genre. For this reason, it would be better if each of the Game class' attributes could have more than one value when appropriate.

Some of these lists could have a direction, starting with the broad category and then more specific ones, such as sub-genre or less prominent genres that the game also fits into. For example, the genre attribute for a third person shooter could be described like this: [action, shooter, third-person].

### 3.2 Include a search function

Games are currently searchable by all of their attributes aside from their name - arguably the most important one. A search bar would remedy this, I would put it in the red header bar so that it's always accessible.

### 3.3 Integrate a Database

Currently, the data is loaded from a JSON file. If the app were to be used in real life a database would need to be used for scaleability and search reasons.

### 3.4 Edit and delete existing data

The ability to add games within the application is so much faster than editing the file itself. If I developed this app further, I would make pages for editing and deleting existing games as doing that in JSON is very time consuming and not user friendly.

### 3.5 Dynamic navigation bar

The filter options on each page, such as 'Platformer' on the Genre page, are already loaded dynamically by searching through the games themselves. I could take this a step further by generating the navigation bar based on the available attributes in the Game class - so if I added a 'Director' attribute to the Game class, it would automatically be added to the navigation bar.

## 4 Critical Evaluation

I am pleased with this application so far - It allows a user to find games quickly based on a lot of different criteria, adding games is simple and the design is user friendly.

The URL structure, despite not being too deep, is logical and easy to guess - the navigation bar along with buttons at the

top of the screen also make it quick to get anywhere in the app.

The code is flexible - I only use a few HTML files which change based on parameters passed in, which made it possible to change the layout and look of the app very quickly and efficiently. The filter options for each page are loaded in by looking through the games, making it easy to add a new genre/year/etc into a game object as it will automatically appear on the right page.

These new genres/years/etc can then be used as a filter immediately with no extra setup due to my routing solution, which makes use of URL variables to cover all of the app's pages with a small amount of flexible routes.

Any attempt to access a page that doesn't exist shows a custom error page (404) and gives the user a way out with a 'Back to home' button. If a user tries to access a filter option that doesn't exist, for example '/genre/mario', they will see "The genre 'mario' doesn't exist" and be given a 'Back to genre selection' button, or the equivalent based on the category page that they are on.

The use of JSON makes it very quick to add games, especially when using the add game page. The application does need a way to edit and delete games too, as doing that manually is cumbersome and not user friendly - if the users even have access to it at all.

There is a lot of potential for expansion, as I mentioned in the Enhancements section. Most importantly, I think a search feature is essential for this type of application; currently games become harder to find as more are added, which would make the app useless after the list reached a few hundred games.

The game page is very basic, just showing the box art and details. Depending on the reason a user has for using the app, these details may not be enough information. I would improve this by adding more relevant details, such as a carousel at the top with screen-shots/trailers, a description of the game, user reviews/comments and a related games list.

The application's layouts are not as responsive as I would like. It does scale well on various monitors, but the mobile experience leaves a lot to be desired with small icons and poor use of space. If it were a priority, I would make the app fully scale-able so that it worked well on all screen sizes and aspect ratios.

## 5 Personal Evaluation

I learned a lot when building this project and it has given me more confidence in my technical skills. I have always struggled with back-end web development so developing a python flask application quickly and feeling comfortable with it by the end was exciting.

This application was built quickly and efficiently as I spent a lot of time planning how I would approach the URL structure and flexibility of the code. I could have planned my solution for images better as it uses URLs rather than locally stored images - potentially raising licensing issues.

I was apprehensive about switching to a JSON based data structure after having it hard coded, but I was surprised how quickly I implemented it due to some great online help [1]. This encouraged me to go for the 'add game' ability too, which I also completed very quickly, using similar online help [2]. These successful endeavours have made me more confident with pushing myself to go out of my comfort zone, it has made me excited and more ambitious for the second coursework too.

This was the first time I had used LaTeX for a project, so I struggled with it initially. I decided to attempt using it after hearing about the overleaf platform (An online LaTeX environment). I loaded the template and started trying to manipulate it until I felt comfortable. Now that I'm familiar with the technology, I will use it more in the future as it made report writing fast and simple.

## References

- [1] "Reading JSON." <https://stackoverflow.com/questions/40019961/reading-a-json-file-using-python-flask>.
- [2] "Updating JSON." <https://stackoverflow.com/questions/13949637/how-to-update-json-file-with-python>.

Note: Most of my work was based on the workbook, hence the short references list. However, I did use some online help for the JSON implementation, as I mentioned in the personal evaluation section.

## 6 Appendices

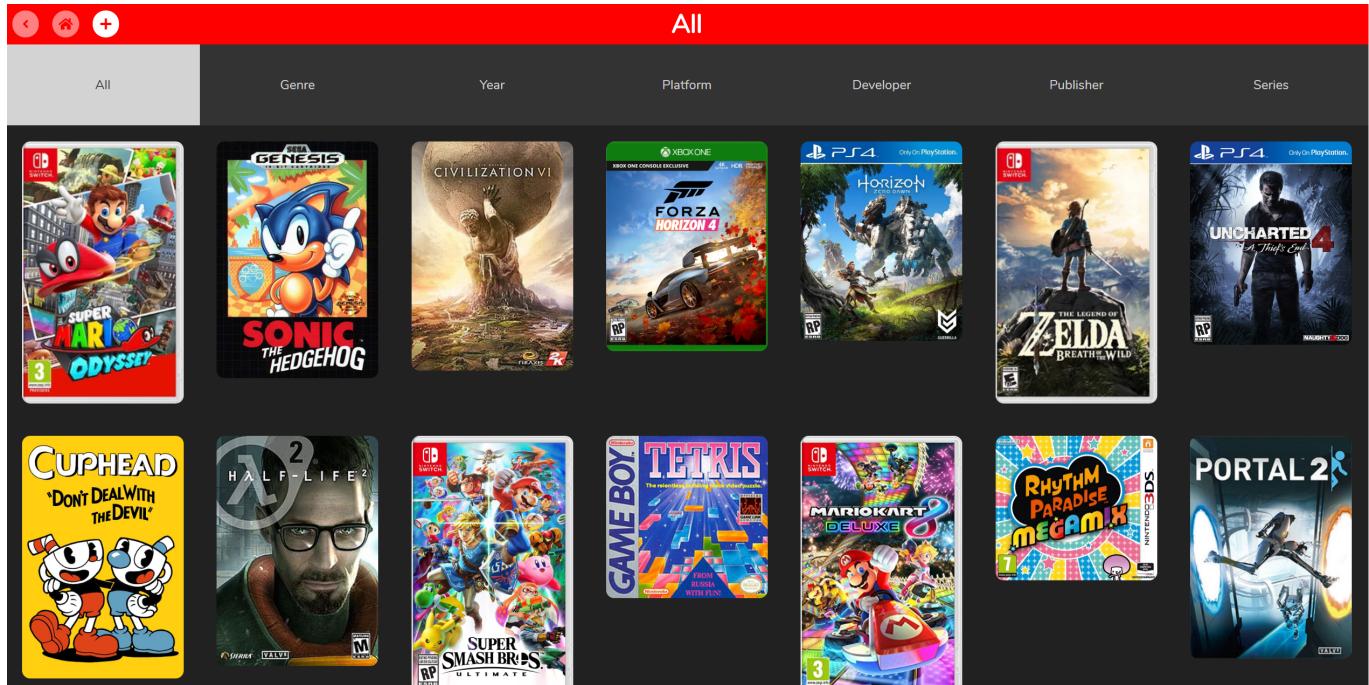


Figure 1: Home Screen - An unfiltered list of games

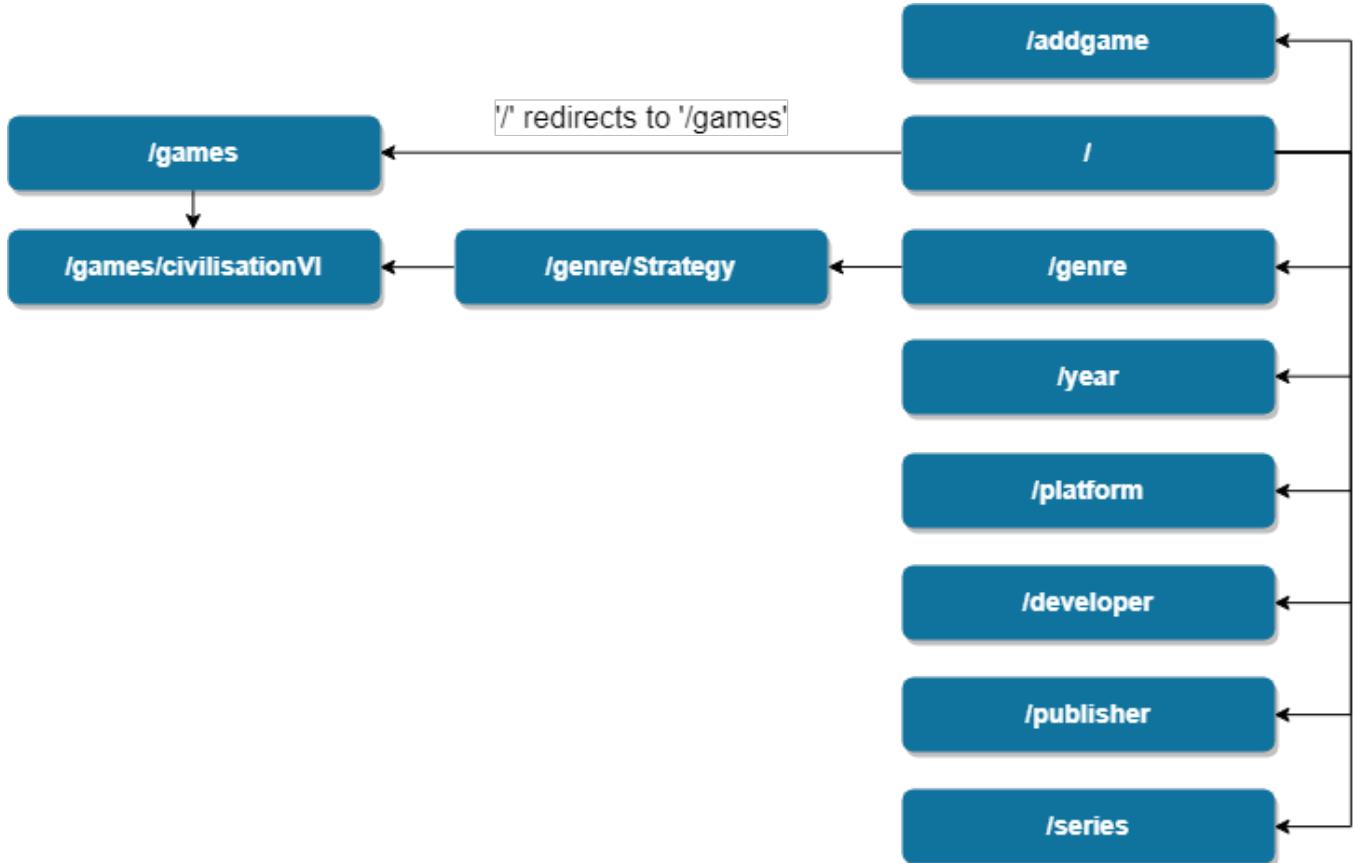


Figure 2: URL Hierarchy - The URL structure of the web app