

Retro Games Planning

What will my site be about?

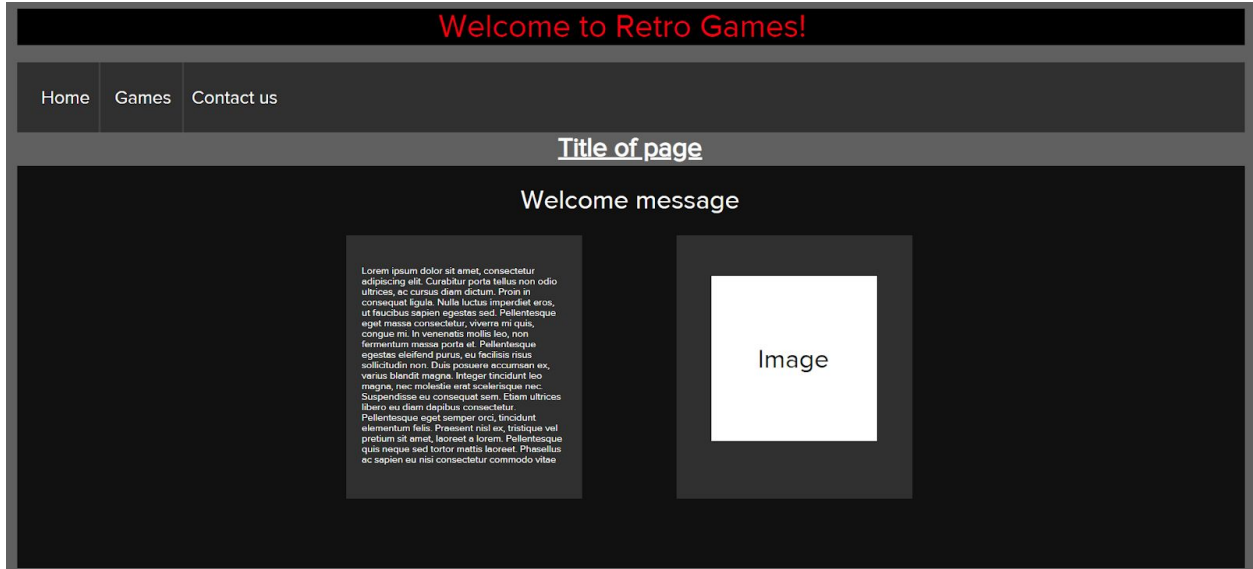
My site will be a database of different retro games, enabling the user to view the differences between each game such as category (e.g Platformer, RPG, Point and click, etc.), company (SEGA, Atari, Nintendo, etc.), and the console that the game was released on.

This site is designed to showcase a collection of my favourite retro games, appealing to an end user that is anyone interested in retro games or games in general.

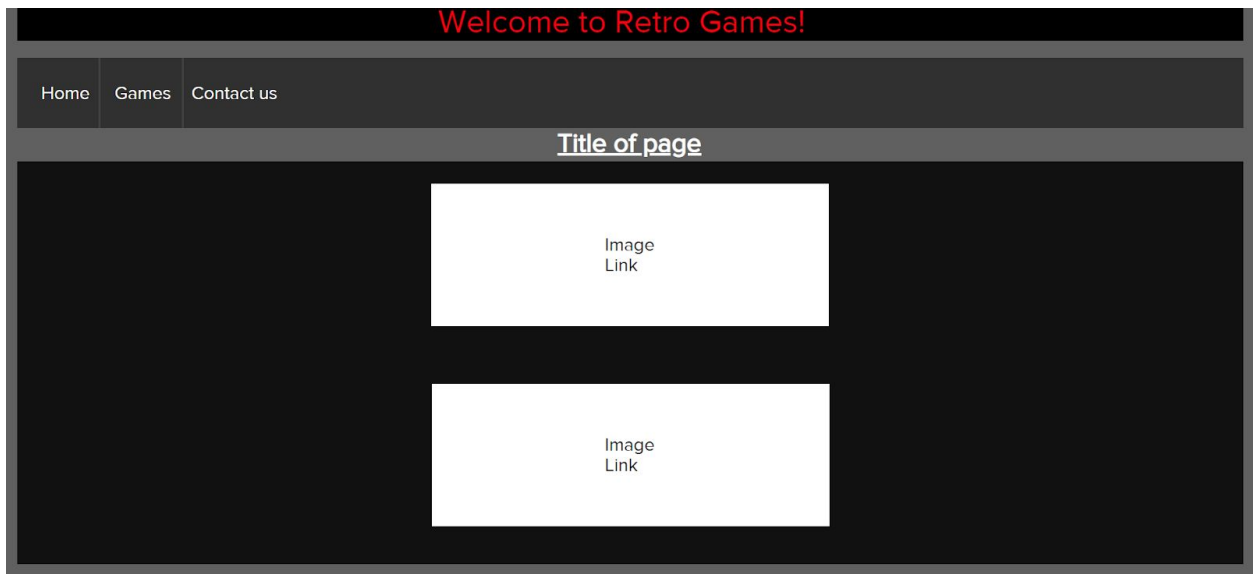
Initial Design screenshots

Site

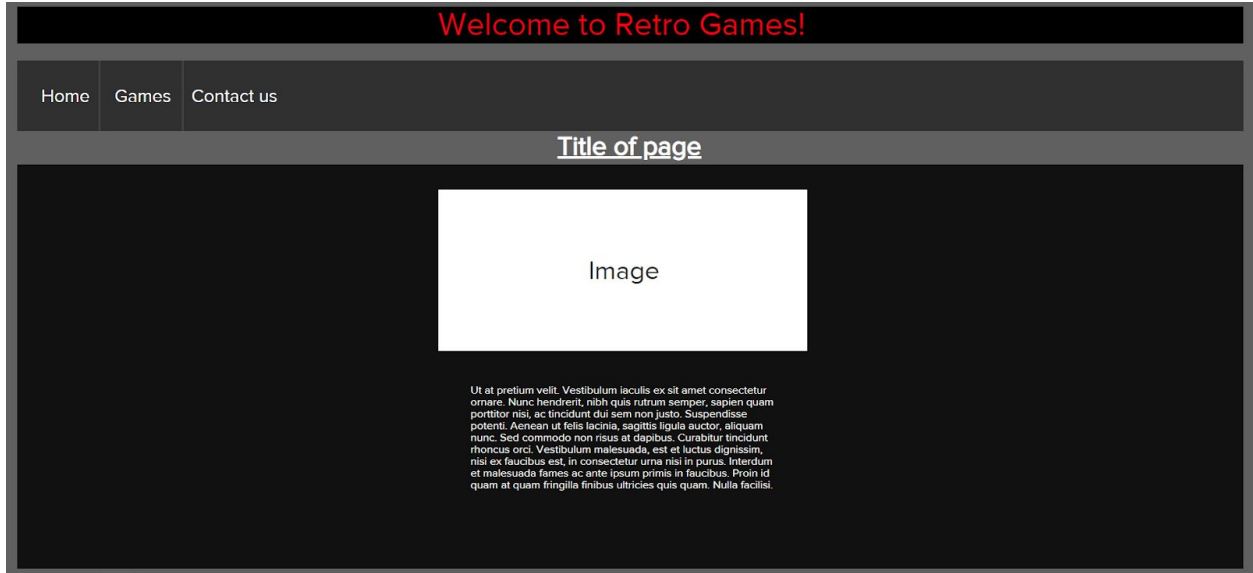
“Home” Page



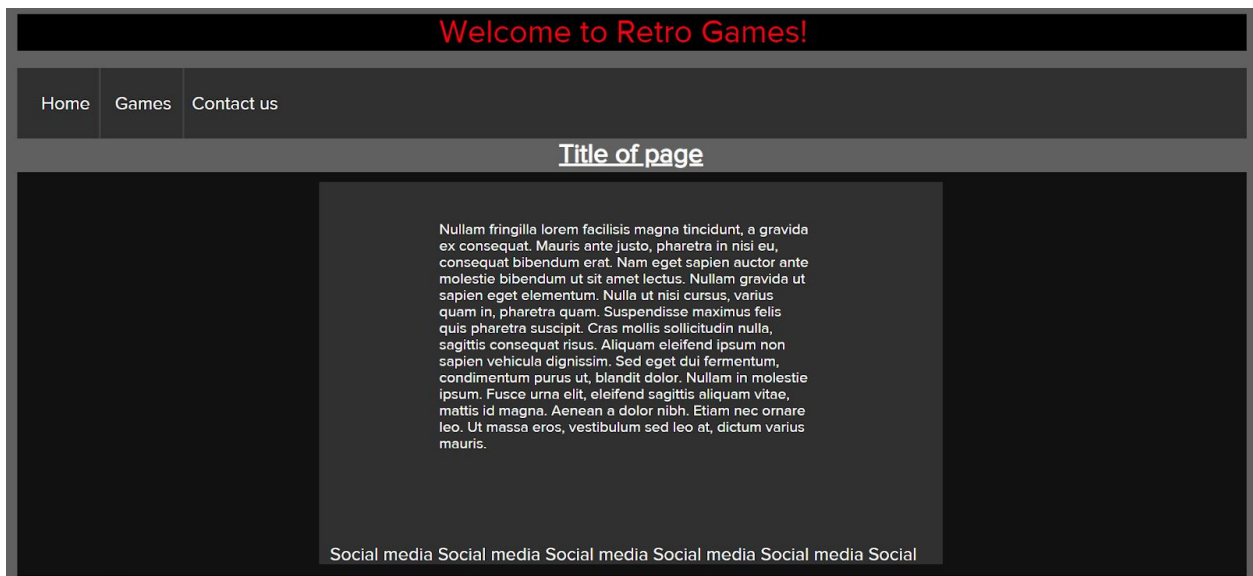
“Games” Page



Page after entering “Image Link” (info on individual games)



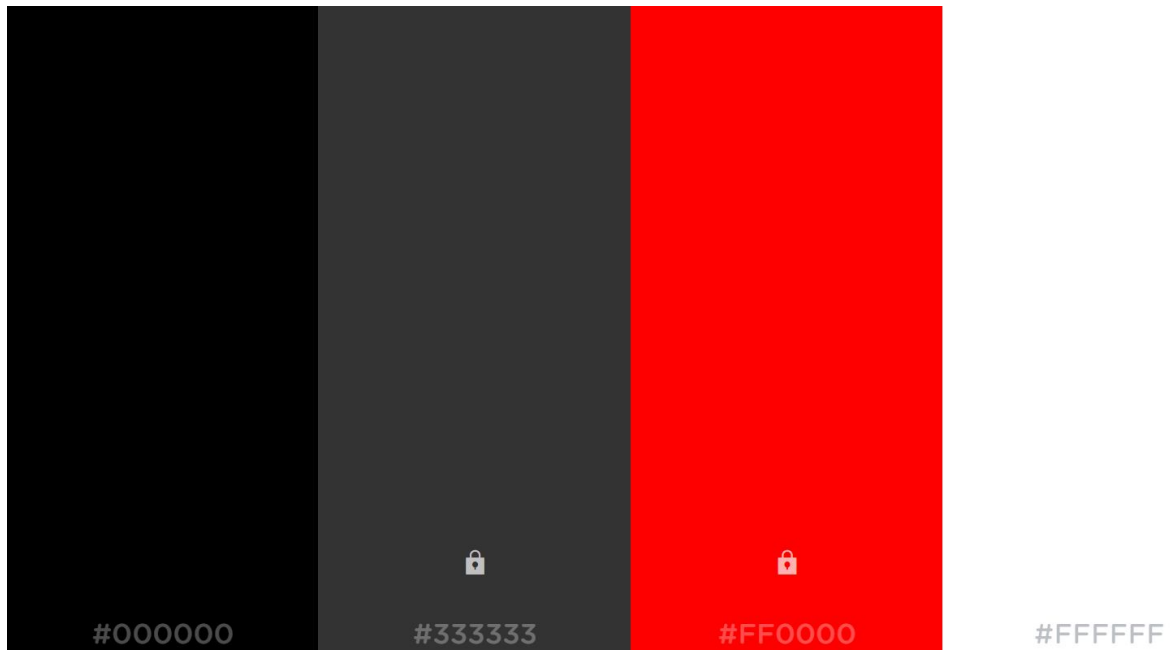
“Contact us” Page



(used wix to build screenshots of pages)

Colour Schemes:

- Darkened base colours for ease on the eyes, especially in low light environments like at night
- High contrast red on black, and white on black/gray



(used <https://coolors.co> to make palette)

Tools to be used

IDE - Atom Text Editor - Best 'IDE' (technically a text editor) for this project as it has full integration with git which makes it extremely easy to track the progression of this site and control versions. I am also a big fan of the ability to add packages to the IDE which make it much easier to develop certain things

Database - SQLiteStudio - Reasonably straight-forward database program for beginners, has an easy to understand language which is used to generate queries and such. Also [creately](https://creately.com) to make the database diagram.

Languages - CSS - Industry standard stylesheet language for web development, nice and efficient for modern day sites, HTML5 - Most modern markup language and easy to integrate with other tools I am using on this project, Python Flask Library - I already have experience using Python, so Flask is easy to pick up and it more than capable of handling the scope of this project, SQLiteStudio - Used for running queries on the database program I am using, SQLiteStudio, also nicely integrates into Flask.

General - Google Chrome - used for both viewing site and searching online for documentation to aid in problems I was experiencing, Git - Used as in my opinion has the best version control services and is easy to pick up and integrate into a project, <https://coolors.co> - Used to create a colour palette for planning, nice and simple website, <https://canva.com> - Used to create design prototypes and screenshots as it is nice and easy to jump into and create design elements as opposed to having to draw them which I am considerably worse at and I think that it would take more time to do.

Techniques - Used a unicode symbol for the search bar in order to provide a more user-friendly

visual alternative to identifying it.



Another technique I used in the site is making a visual slight change of colour on the part of the navbar you are hovering over



I implemented this in order to make the link the end user is about to click easily identifiable in order to prevent them going to the wrong page and wasting time.

Pages Planning

“Home” Page

Will have regular route, render template, function but no queries for home page as it isn't linked to database in any way

“Games” Page

Will have route, render template, and a simple query to select each game from database and display on page

Page after entering “Image Link” (info on individual games)

This page will have an overall route and function but will have a query and render template for each aspect of the game e.g category of game, manufacturer, etc.

“Contact us” Page

Similar to home page, will have regular route to access page with a render template and function but no queries as it is solely html and css on this page, no data to be called from a table in database

Relevant Implications:

Aesthetics - Aesthetics in a website is how the website looks to a visitor. It can include things such as layout, colour usage, and fonts. Maintaining good aesthetics on a site are very important as it manipulates the impression a visitor has of the site which potentially increases how effective the site is in whatever purpose it is built for. Retro Games uses dark, minimalistic, and simplistic colours so that it is easy on the viewer's eyes. It also uses a layout that is natural to follow with header/title at the top of the page, navigation bar right below that, and the main contents of the page in the middle of the page. It is designed this way so that eyes will move downwards and view the page in the correct order. These design elements and others make the site meet aesthetics requirements and improve the viewer's experience of the site.

Usability - Usability of a site is how certain elements of a page are placed and designed so that they are more usable for users. It includes things such as placement of elements, how visually identifiable elements of a page are, and limiting things that may confuse the user. I have met the requirements of usability in the Retro Games site by placing elements of my website in a natural-flowing order, from introductory messages at the top of the page and details of the page towards the bottom, so it is easy to read the site as it is intended.

Intellectual Property - Intellectual Property refers to media that people have created and own, where others are not allowed to use that media without gaining permission or paying a fee first. In my site, I have ensured that all images I use on my site were found on google images and labelled for reuse. This means that there are no intellectual property concerns with media that I use on my site.

Database Planning

Tables:

Category:

This table will hold the data for each category so as to fetch data for the genre/category of each game. Inside this table, I will have the columns: **id** (for identification of each row. Will be a primary key and unique. Will have the INTEGER datatype), **name** (to store and fetch the name of each category. Will be Not Null to ensure I don't leave empty, and will also have a max length of 50 characters, will have the TEXT datatype.), and **details** (to store and fetch the description/details for each category. Will be Not Null to ensure I don't leave empty, and will have a max length of 250 characters, also with a TEXT datatype).

Developer:

This table will hold the data for each developer/company who produced each game in my site. Inside this table, I will have the columns: **id** (for identification of each row. Will be a primary key and unique. With an INTEGER datatype), **name** (to store and fetch the name of each developer. Will be Not Null to ensure I don't leave empty, and have a max length of 50 characters. With a TEXT datatype), **details** (to store and fetch the description/details for each category. Will be Not Null to ensure I don't leave empty, and will have a max length of 250 characters. With a TEXT datatype), **image** (to hold images for each developer so I can fetch the image for the corresponding developer the user will choose. With a TEXT datatype)

Games:

This table will hold the data for each individual game so as to fetch data for the corresponding data from the Category and Developer tables of each game. Inside this table, I will have the columns: **id** (for identification of each row. Will be a primary key and unique. With an INTEGER datatype), **name** (to store and fetch the name of each category. Will be Not Null to ensure I don't leave empty, and have a max length of 50 characters. With a TEXT datatype), **details** (to store and fetch the description/details for each game. Will be Not Null to ensure I don't leave empty, and have a max length of 250 characters. With a TEXT datatype), **image** (to hold images for each game so I can fetch the image for the corresponding game the user will choose. With a TEXT datatype), **developerid** (Will be a foreign key so I can fetch corresponding data for the game's developer information to call into the site. With a INT datatype), and **categoryid** (Will be a foreign key so I can fetch corresponding data for the game's category/genre information to call into the site. With a INT datatype)

***extended planning to include following pages: ***

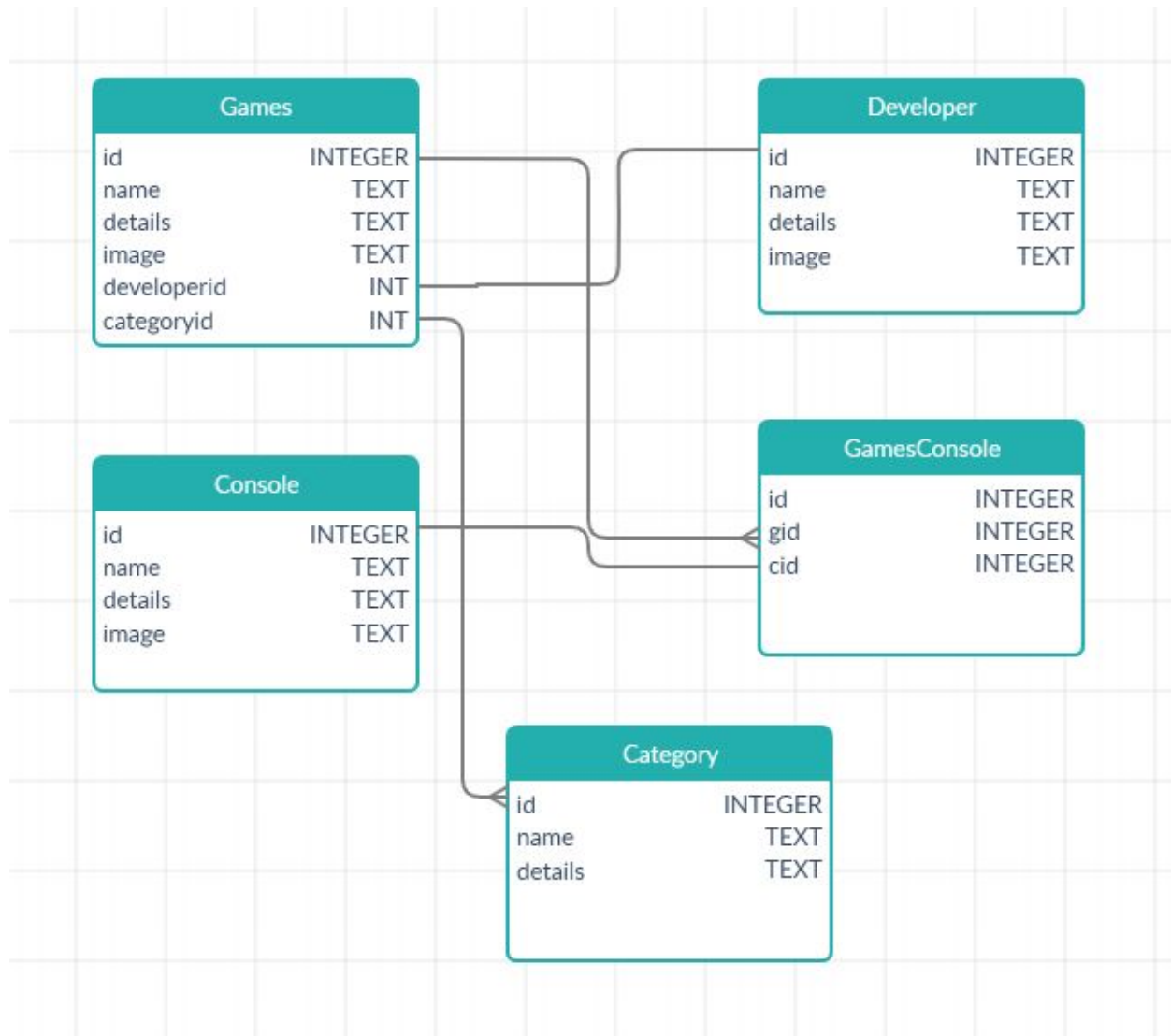
Console:

This table will hold the data for each console that each game was/is available on. Inside this table I will have the columns: **id** (for identification of each row. Will be a primary key and unique in the INTEGER datatype.), **name** (to store and fetch the name of each console. Will be Not Null to ensure I don't leave empty, and have a max length of 50 characters in the TEXT datatype.), and **details** (to store and fetch the description/details for each console so the user can have more information on each console. Will be Not Null to ensure I don't leave empty, and will have a max length of 250 characters in the TEXT data type.), **image** (to hold images for each console so I can fetch the image for the corresponding console that the user chooses. Will be TEXT datatype.)

GamesConsole:

This intermediate table acts to hold temporary data so that I can draw data from my console table and display it on the site. I will have the columns: **id** (for identification of each row. Will be a primary key and unique. With an INTEGER datatype), **gid** (foreign key to store id of games in the games table. Will be Not Null to ensure I don't leave empty), and **cid** (foreign key to store id of consoles from consoles table. Will be Not Null to ensure I don't leave empty).

Visual diagram:



Queries:

- **"SELECT * FROM Games WHERE name LIKE ?"** - gets relevant data from search from Games table in database and puts it into the /gsearch route
- **"SELECT * FROM Developer WHERE name LIKE ?"** - same as above for developer table and /dsearch route
- **"SELECT * FROM Console WHERE name LIKE ?"** - same as above for console table and /csearch route
- **"SELECT * FROM Games ORDER BY name ASC"** - gets all data from games table and prints it into /games route, also sorted in alphabetical order for ease of use
- **"SELECT name, details, image, developerid, categoryid FROM Games WHERE name=?"** - gets the name, details, image, developerid, and Categoryid from Games table
- **"SELECT name FROM Developer WHERE id=?"** - gets names from Developer table
- **"SELECT name from Console INNER JOIN GamesConsole ON Console.id=GamesConsole.cid WHERE gid=(select id FROM Games WHERE name=?)"** - joins intermediate table between Console and Games, 'GamesConsole' and gets data based on what the 'gid' is
- **"SELECT * FROM Developer ORDER BY name ASC"** - gets all data from Developer table and prints it into the /developer route, also sorts it alphabetically for ease of use
- **"SELECT name, details, image FROM Developer WHERE name=?"** - gets name, details, and image from the Developer table and prints it into the /developer/<developer> route
- **"SELECT * FROM Console ORDER BY name ASC"** - gets all data from Console table and prints it into the /console route, also sorts it alphabetically for ease of use

- ***"SELECT name, details, image FROM Console WHERE name=?"*** - gets name, details, and image from the console table and prints it into the /console/<console> route

Relevant Implications:

Functionality - Functionality is the level at which something functions and does its desired job. In terms of my database, this would be how well the database displays data to my page through each query. My database has a lay out which works well with queries to display the data onto the page as it makes use of foreign keys to draw data from relevant tables which may decrease the time for each query to run, therefore allowing the user to load into pages displaying data from my database faster.

Sustainability and Future Proofing - Sustainability and Future Proofing is how much something is designed for the future so that it can have a longer functional life. In the case of my database, this would be how much it is designed so that it can be used into the future and be more adaptable to changes to the overall project. My database has different tables for different data e.g it has a table for data relevant to games, one relevant to developers, and so on. This means that it is much easier to add data to the database as you can just go to the relevant table instead of searching for a place to put it in in one big table. This means that the database is much more allowing for future data being added as it is easier to add and retrieve through a query.

Intellectual Property - Intellectual Property is the legal consideration when using something as it may belong to somebody else and you either have to get permission or pay a fee to use it. When using images in my database, I have only used images which are marked as free to use on google images so that these images are legally fine to use and I don't have to worry about the intellectual property rules around these images.

Testing and Debugging

Navbar:

Tested that Home button on navbar links to the app route '/'



localhost:1111

Tested that the Games button on navbar links to the app route '/games'



🌐 localhost:1111/games

Tested that the Developers button on navbar links to the app route '/developer'



🌐 localhost:1111/developer

Tested that the Consoles button on navbar links to the app route '/console'



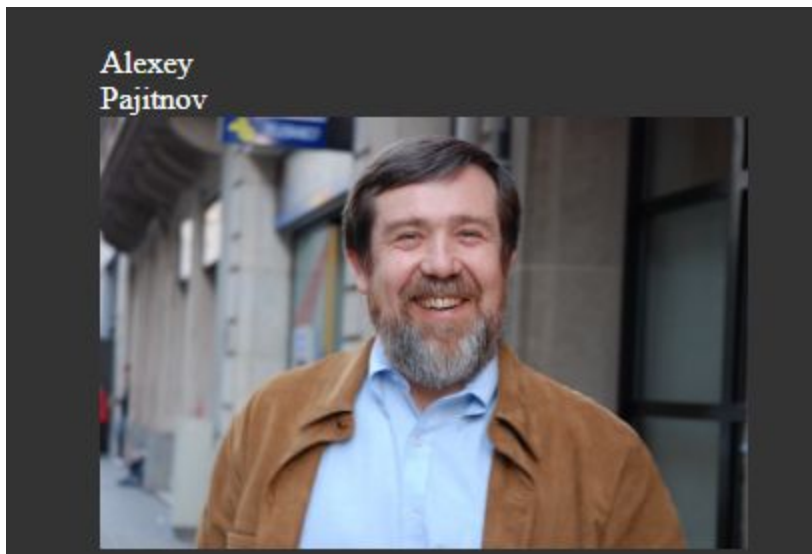
🌐 localhost:1111/console

Tested that the Contact Us button on navbar links to the app route '/contact'



Spot checks:

Ensured that links in list_developer page lead to the correct corresponding show_developer page. Did this by clicking the 'Alexey Pajitnov' developer



This link led to the correct page.

Alexey Pajitnov



Alexey Pajitnov was a Russian video game designer, Alexey developed the famous Tetris game while working at a government funded R&D center

Made sure that the links within the show_games page which lead to the corresponding show_developer and show_console pages work.

Tested this with the links on the 'Doom' game.



This link worked correctly.

id Software



id Software was a recently founded developer (1991). It has been responsible for making famous first person shooter games such as Doom, Wolfenstein, Quake, and Rage.

Console:

Doom was first released on the
[MS-DOS PC](#)

This link also worked correctly.

MS-DOS PC



MS-DOS PC was one of the first operating systems for a PC released in 1984

Made sure that search results would display correctly and link to the correct page. Tested this with the search bar on the `list_games` page.

Searched 'ast' in the bar and got a result of a huge image being displayed although the link itself was working.

Results matching your search:

Asteroids



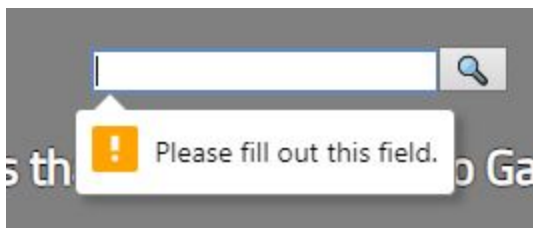
Fixed this by realising that I had changed the name of the div in the css file and forgotten to change it in the search html pages.

Applied this fix to the necessary pages and checked to make sure it worked.

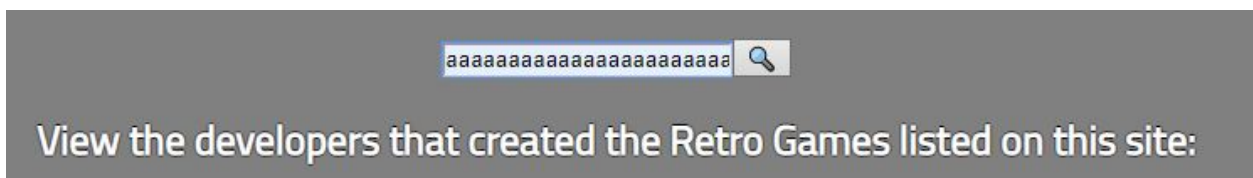
Results matching your search:



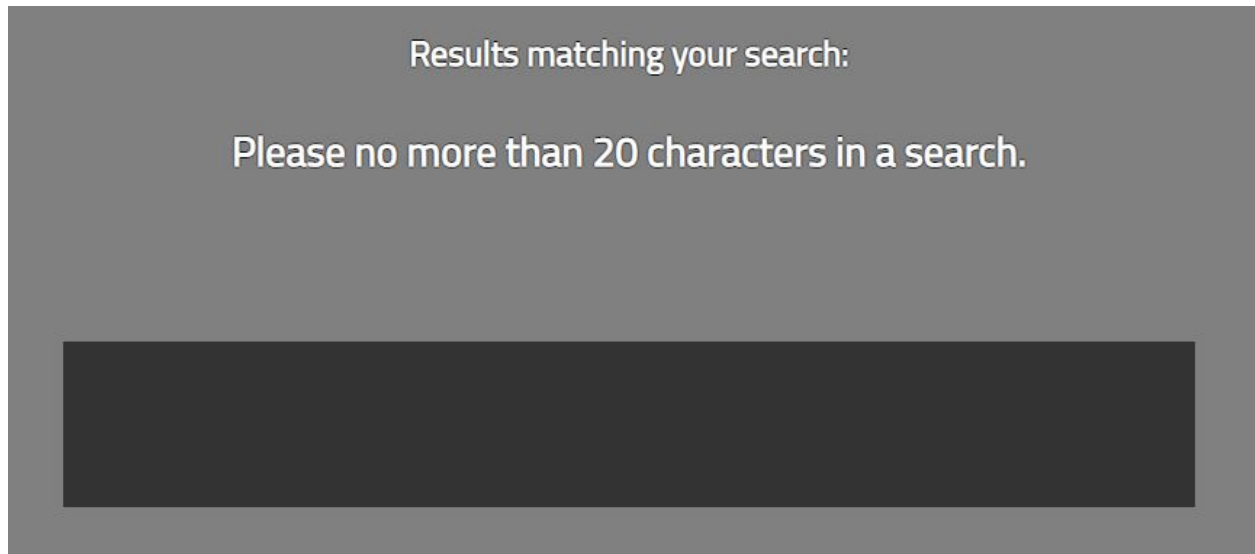
Tested to make sure that if nothing was entered into the search bar, it wouldn't let the search go through but rather state that the user has to enter some data to search.



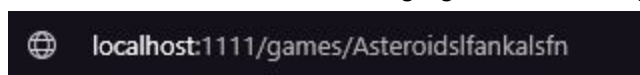
Tested to make sure that if more than 20 characters are entered into the developer search bar, the search would not go through but it would come up with a flash message to tell the user to enter less characters.



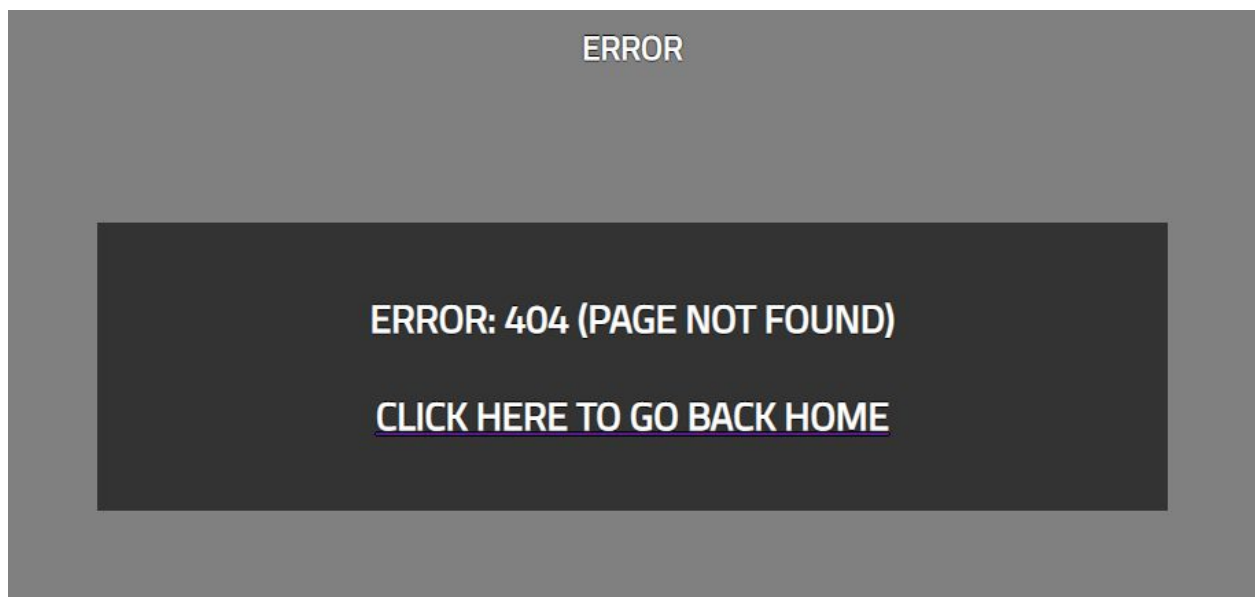
The message came up.



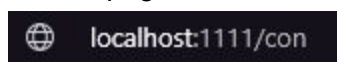
Tested to make sure that changing URL on show_games page would result in a 404 page.



It did



Also tested to make sure that a 404 page would come up if the URL was changed for the contact page.



It did.

ERROR

ERROR: 404 (PAGE NOT FOUND)

[CLICK HERE TO GO BACK HOME](#)

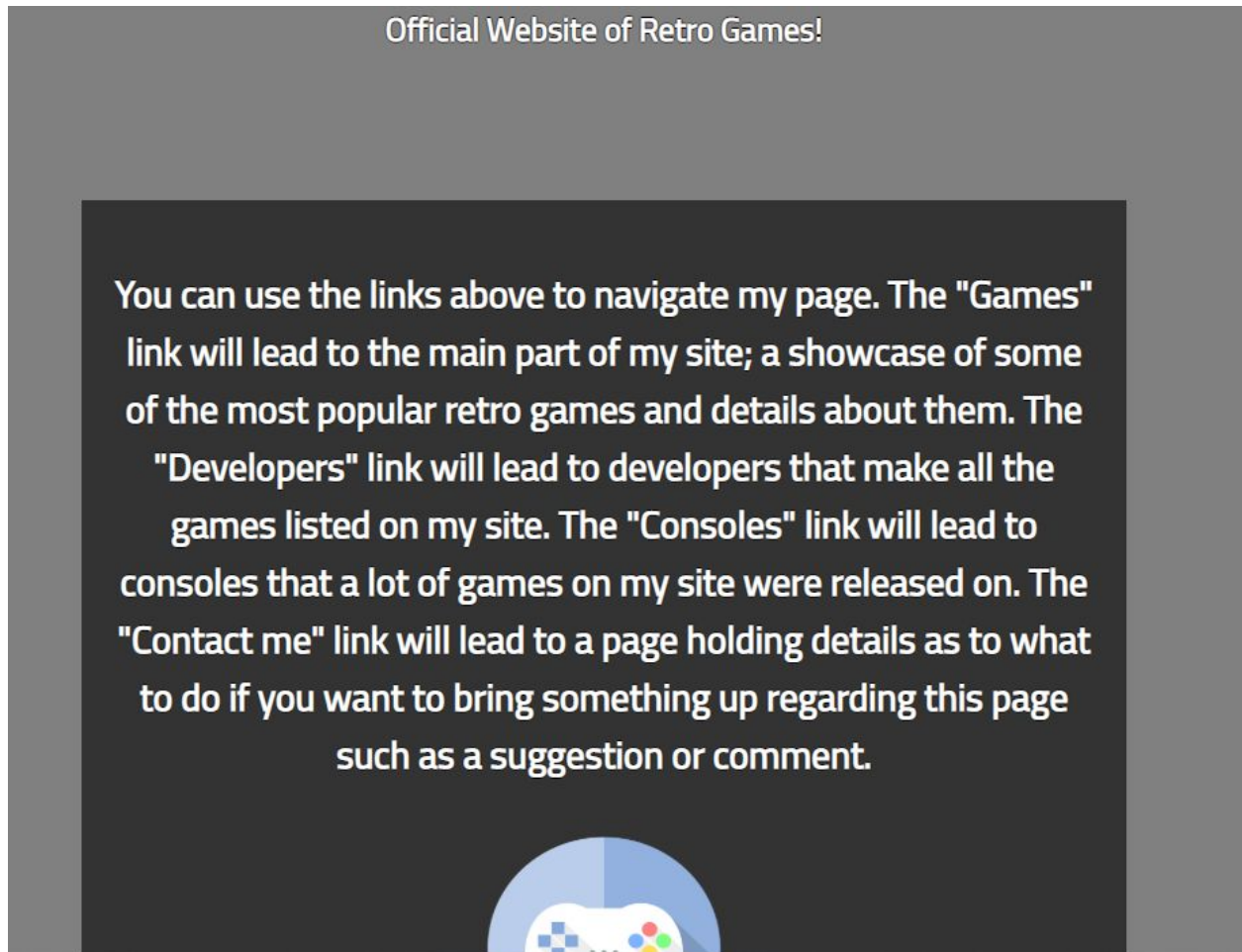
Checked to make sure that the link to go back to home page worked on the 404 page.

ERROR

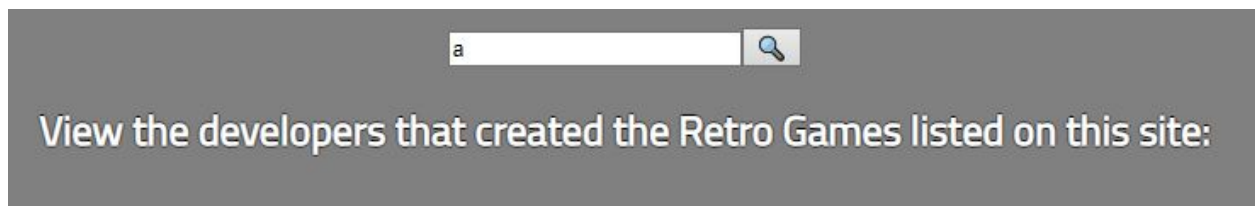
ERROR: 404 (PAGE NOT FOUND)

[CLICK HERE TO GO BACK HOME](#)

It did.



Checked to match results of a search in developer with a view created in my database, to check that the website is serving the correct data.



Taito



Atari



Alexey
Pajitnov



Namco



id
Software



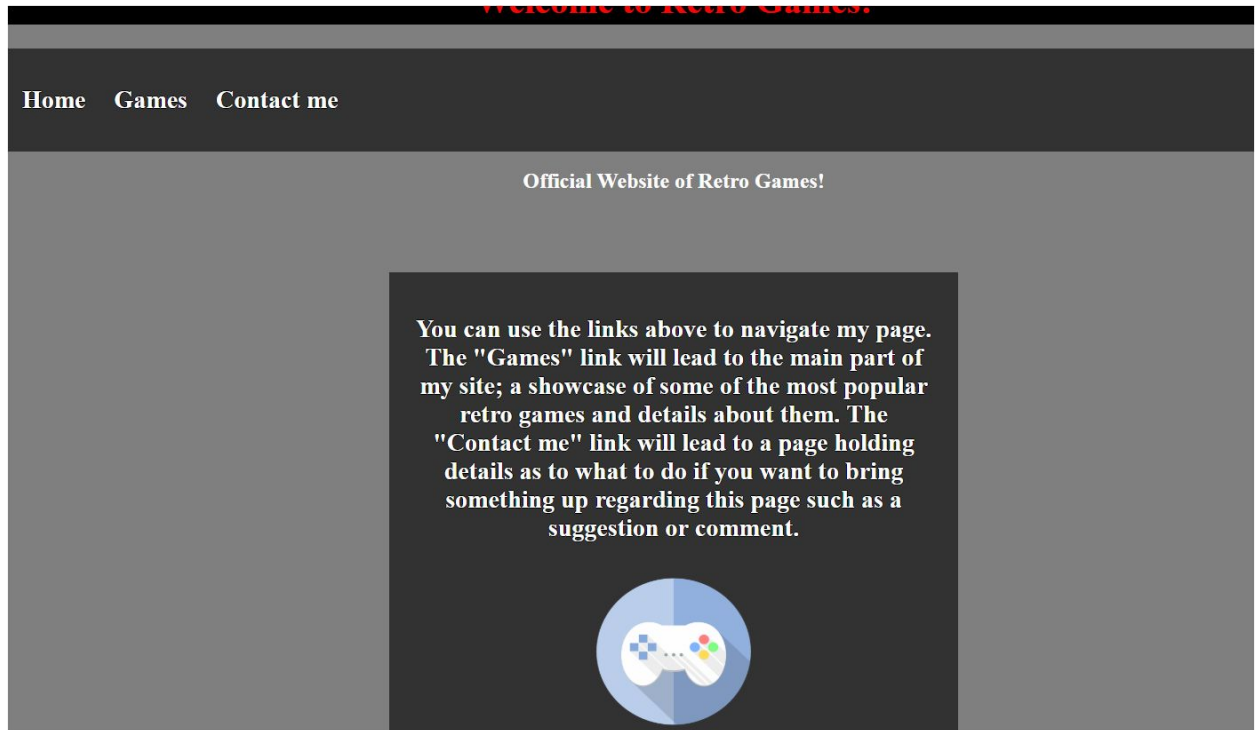
The results were the same as the view:

	id	name	details
1	1	Taito	founded in 1953, where the company imported vodka, vending machines, and jukebo
2	2	Atari	created in 1972, Atari was an important pioneer in the early gaming industry making g
3	4	Alexey Pajitnov	a Russian video game designer, Alexey developed the famous Tetris game while worki
4	5	Namco	a company created in 1955 in Japan, they are the creators behind many popular retro
5	6	id Software	a recently founded developer (1991). It has been responsible for making famous first p

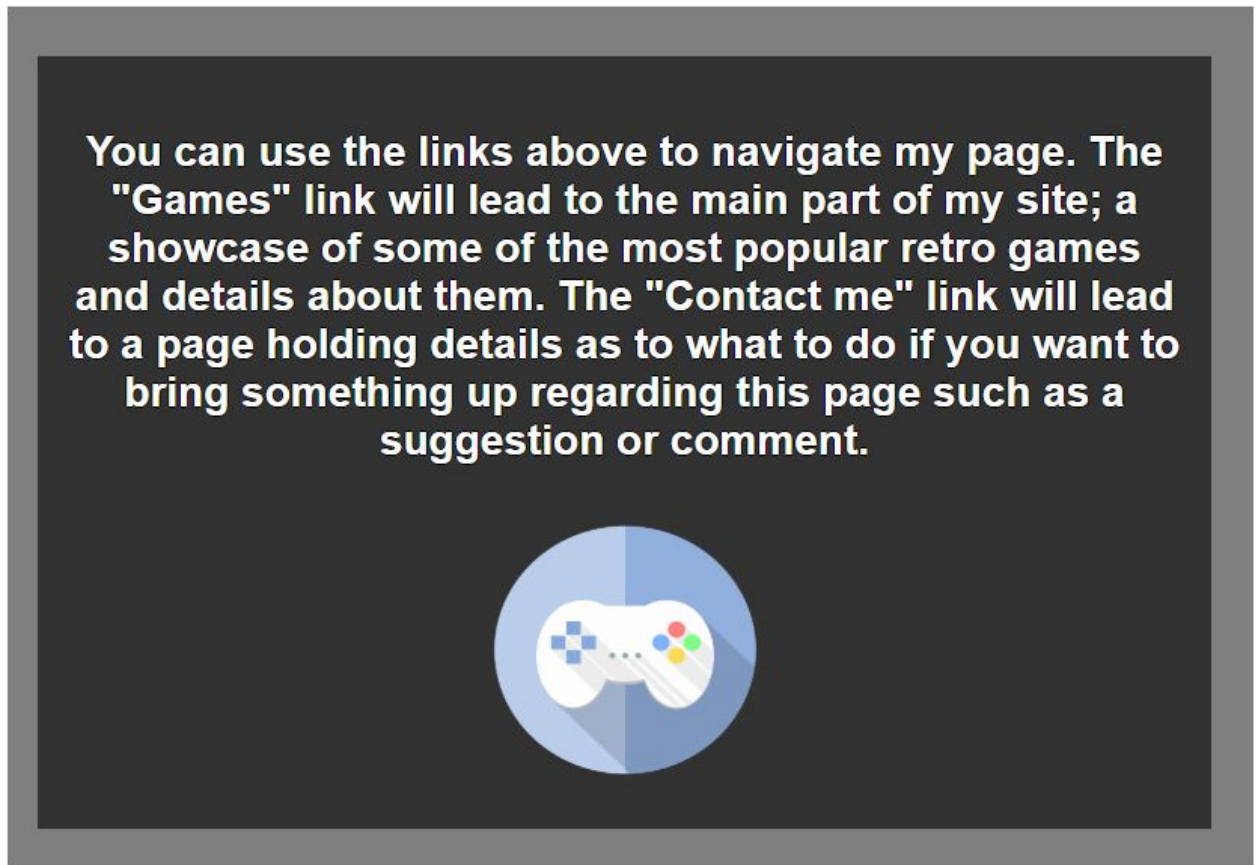
Design Changes

Site:

- Decided to remove large black box on site pages and just have a smaller box in the middle of the page as I thought it became too dark and I favoured more of a lighter, more open feel for the site -



- Changed layout of pages such as the home page to have more of an emphasis on text rather than image -



- Also changed font family to a slightly larger and more suiting family of Arial, Helvetica, sans-serif -



- On the info about individual games page, I refined my planning to have a series of categories for each aspect of the game from each table of my database. -

Space Invaders



Space Invaders is a 2D shooter that lets you attempt to fend off an alien force descending from the sky.

Made by:

Atari

Category:

Shooting: a category involving control of a projectile in order to beat the game - either by killing enemies or clearing your way through obstacles.

Console:

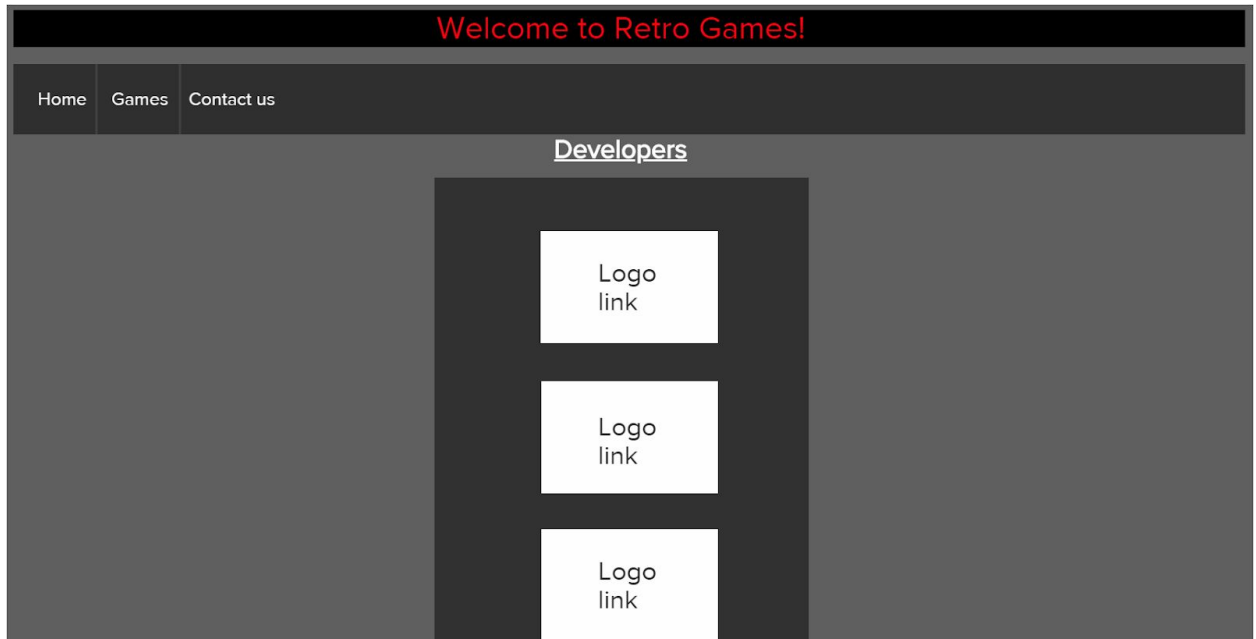
Space Invaders was first released on the Atari 2600

This is to make the layout of information easier and more efficient, so the user doesn't have to read a whole paragraph to find the information they want e.g if they want to see the company who made the game only.

For this change, I needed to make individual queries for each bit of information, for example the console area needed to have a separate query to get the data from the corresponding table in the database. I also needed to add render templates for each query e.g for category I will need my query to get category data and a render template for category

- **“list_developer” page:**

Added another page for information on companies as an additional feature of the site. Planning for this can be seen below where this page will list all of the developers and provide links.



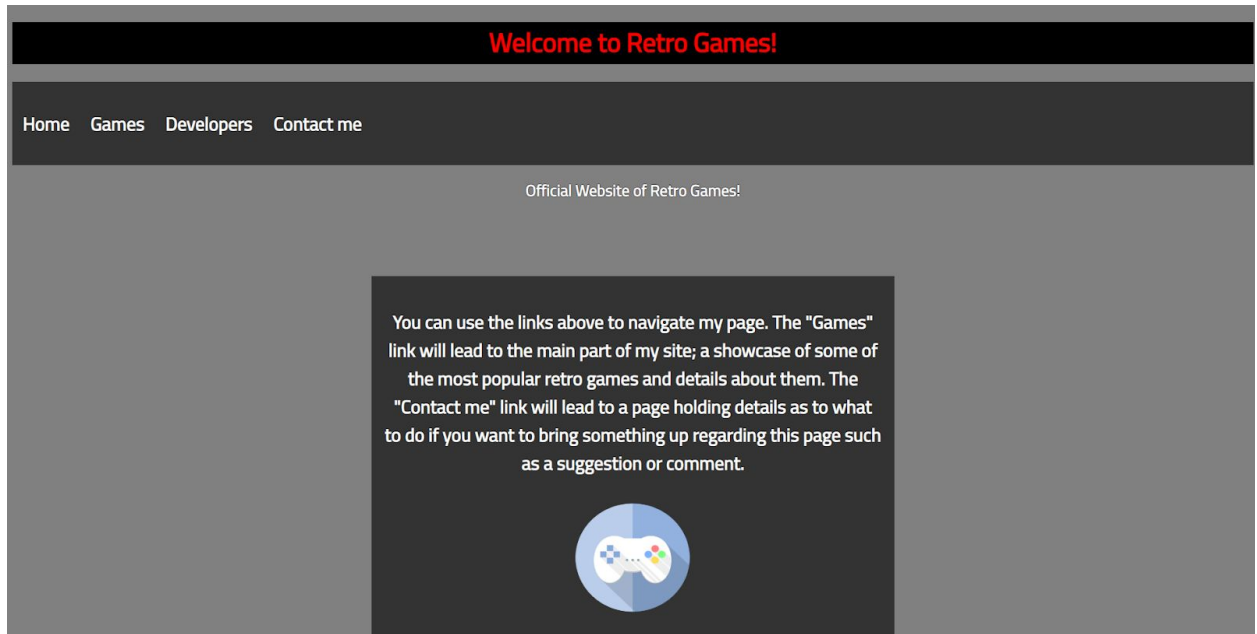
This page will hold info on the company of each game and will be accessible from links inside the pages displaying info on each game.

- **“show_developer” page:**

Added this page to go with the previous page so that it can display more information when each link is pressed. Will have a link to each company/developer in the game page as well. Planning can be seen below.



- Changed font for site again, from the previous font to “Titillium Web” font found off of google fonts - <https://fonts.google.com/specimen/Titillium+Web>. Chose this font as I think that it matches the theme of the site much better than the previous font.



- Decided to add a search function for pages that display games and developers. I decided to add this function to make it easier for the user to find what they are looking for, rather than having to scroll through the list which takes more time. My queries to the database for this will need to include the 'LIKE' operator, which will let me find results in the database which are 'like' what the search input is. This search bar will also be located at the top of both the Games and Developers pages.
- **“list_console” and “show_console” pages**
Decided to add a page on the site to display consoles that games came out on first as this would provide more details to the end user and improve their experience with my site. This will be using my already existing data that displays on the page of each game on what console they were released on, but will hold some more details on the consoles themselves. It will follow design guidelines of the list_developer and show_developer pages.
- Changed information on home screen to fix a problem that was found in Liam’s feedback.

- Changed colour of links from blue and purple to just red as per feedback.
- Added site logo to the tab of page (favicon) for improved aesthetics of site in order to increase the user's experience, also helps them identify which one of their tabs open is my site.

Database Changes:

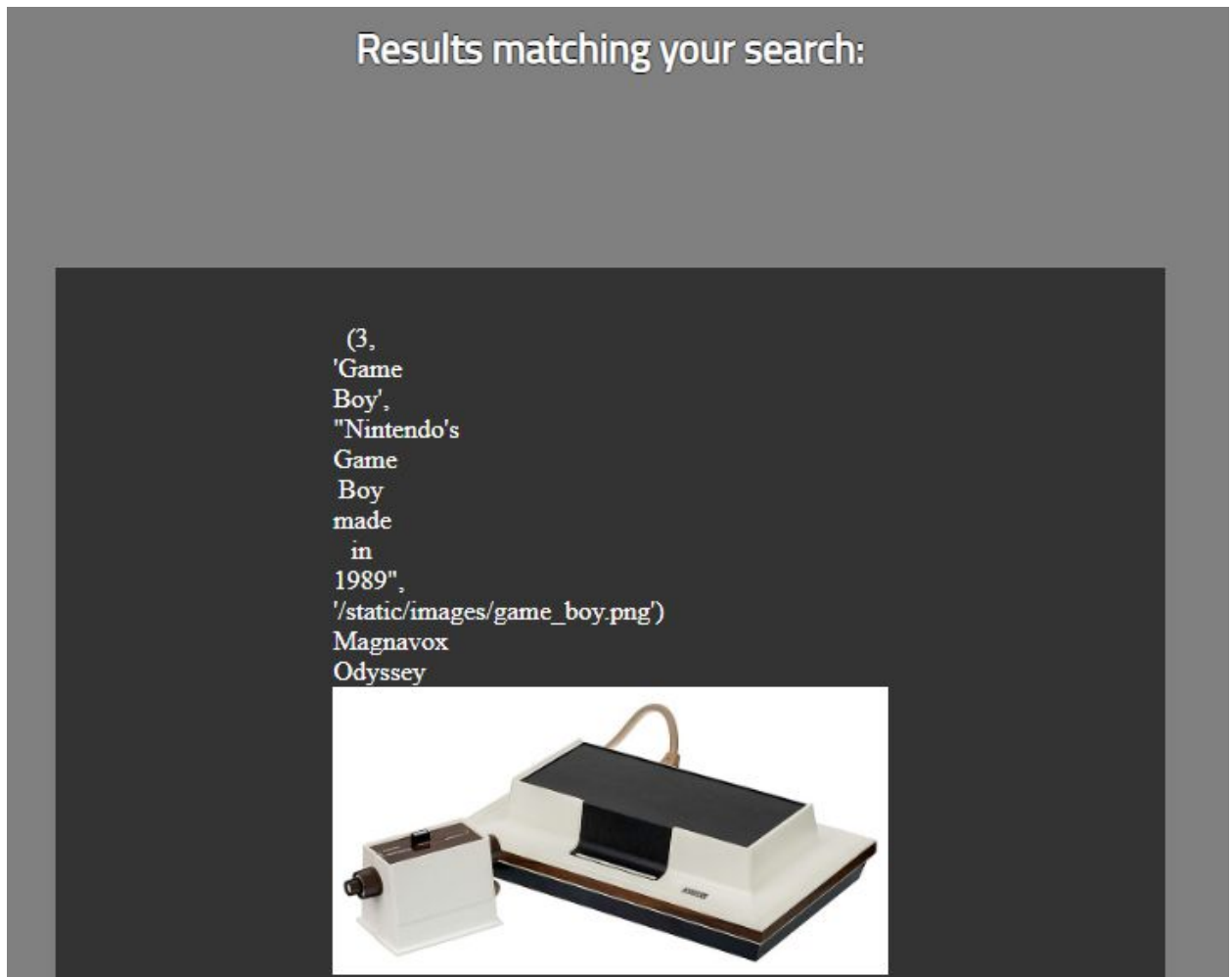
- After testing my site in terms of content, I decided to add another table into my database for the consoles that each game is available on to provide more information to the user. This table will have an intermediate table to my games table in the database.
- Added a table to accompany the extra info on developers. This table will hold the data for each developer that each game was created by. Inside this table I will have the columns: **id** (for identification of each row. Will be a primary key and unique.), **name** (to store and fetch the name of each developer. Will be Not Null to ensure I don't leave empty.), **details** (to store and fetch the description/details for each developer so the user can have more information on each console. Will be Not Null to ensure I don't leave empty), and **image** (to store images for use as links and information on the site).

Feedback

Liam:

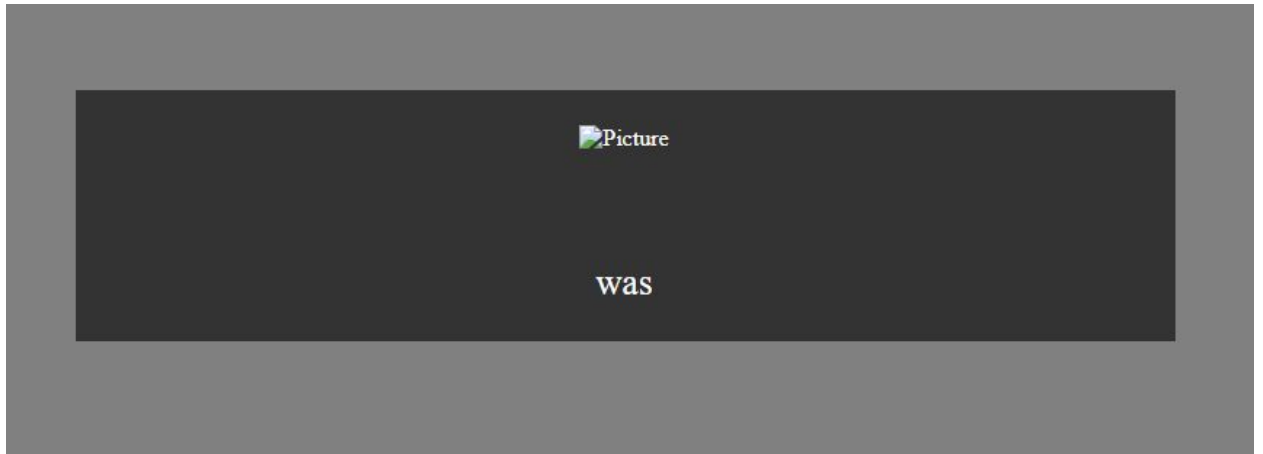
- Need more explanation of how to use site on home page info box i.e doesn't include the developer and console links.
I chose to follow this feedback because it is something that I missed when adding more material on my site, and it is not user-friendly to leave this feedback out.
- Should add a link to the games that each developer made on their page.
Decided not to follow this feedback advice, as I don't want to overcomplicate the site with too many links leading everywhere, and I expect the user to follow the site from left to right on the nav bar so that they see the games first, then the developers.

- Found a problem where when searching, some singular letters such as 'g' can return code instead of the intended result.



Decided to fix this problem as it is not user-friendly. Fixed it by finding an error in my for loop where I was meant to state the variable as 'consoles' instead of 'console'.

- Suggested more details on console info pages, thought "it didn't have enough details".
Added some more small facts to some of the more barren console info pages, including facts that are relevant and that the user would appreciate such as the fact that the game boy was a hand-held console, and that the Atari 2600 pioneered ROM cartridges.
- Found error when changing url on show_games page where the 404 error handler didn't work.



Added a try except to the query and used flask.abort to call the 404 page manually if there is a TypeError.

```
try:
    cur.execute("SELECT name FROM Developer WHERE id=?", (results[3],))
    developer = cur.fetchone()
except TypeError:
    abort(404)
```

Henry:

- Links on show_games page hard to read, contrast on grey background isn't great.

Made by:

[Nintendo](#)

Category:

Platformer: a category that involves movement-heavy gameplay, often involving a lot of jumps and combat.

Console:

Donkey Kong was first released on the [Donkey Kong console](#)

Decided to fix this and change the link colour as I agree that it is hard to read, especially on my laptop's monitor. I changed this to red to match colours of my site more. Also got rid of a different color for visited links because I thought it should be more consistent.

Made by:

Taito

Category:

ports: a type of video game simulating or representing

Console:

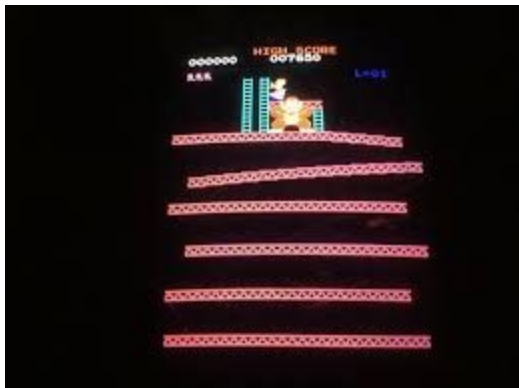
Pong was first released on the

Magnavox Odyssey

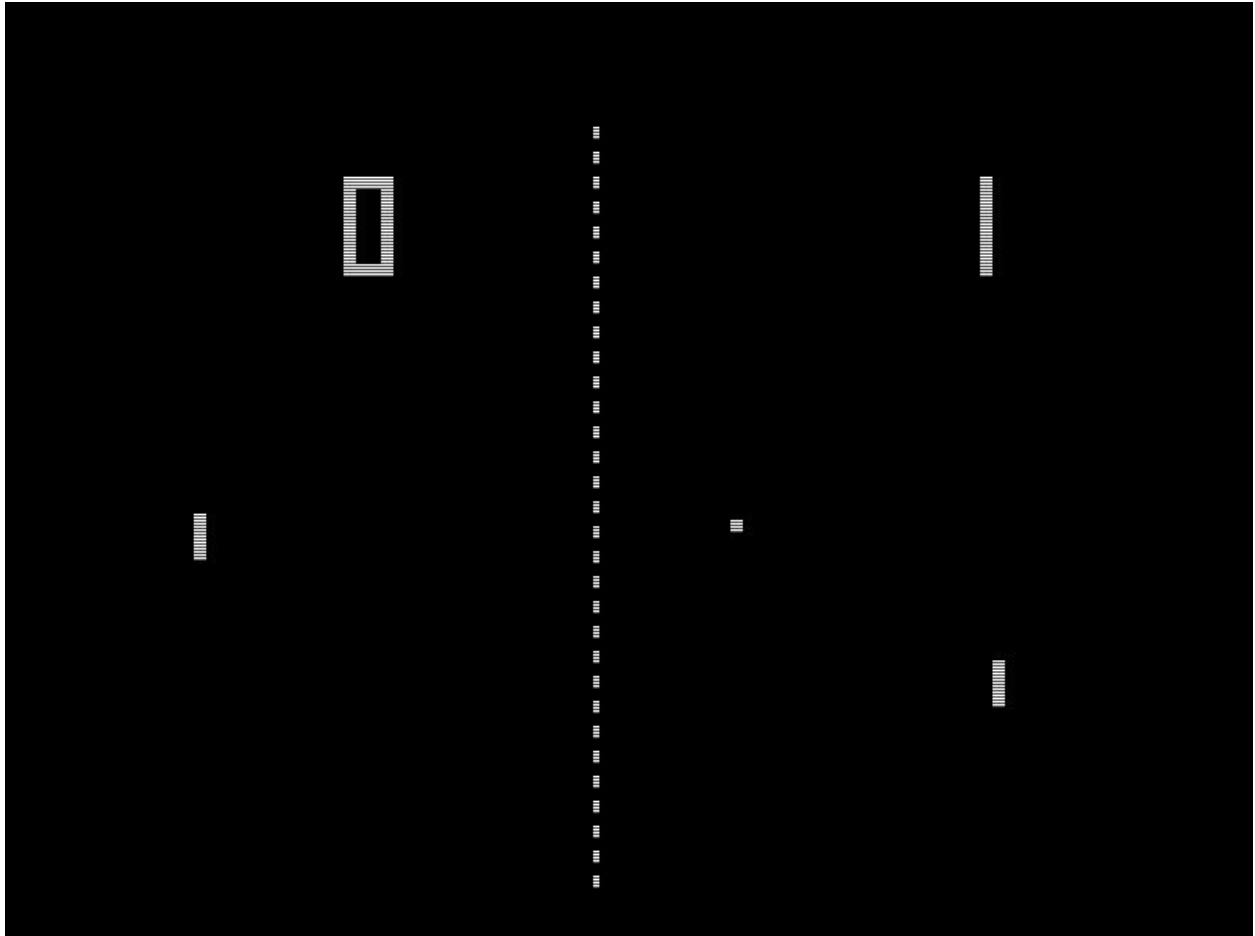
Sources

All images found using Google's "Labeled for reuse" images search filter

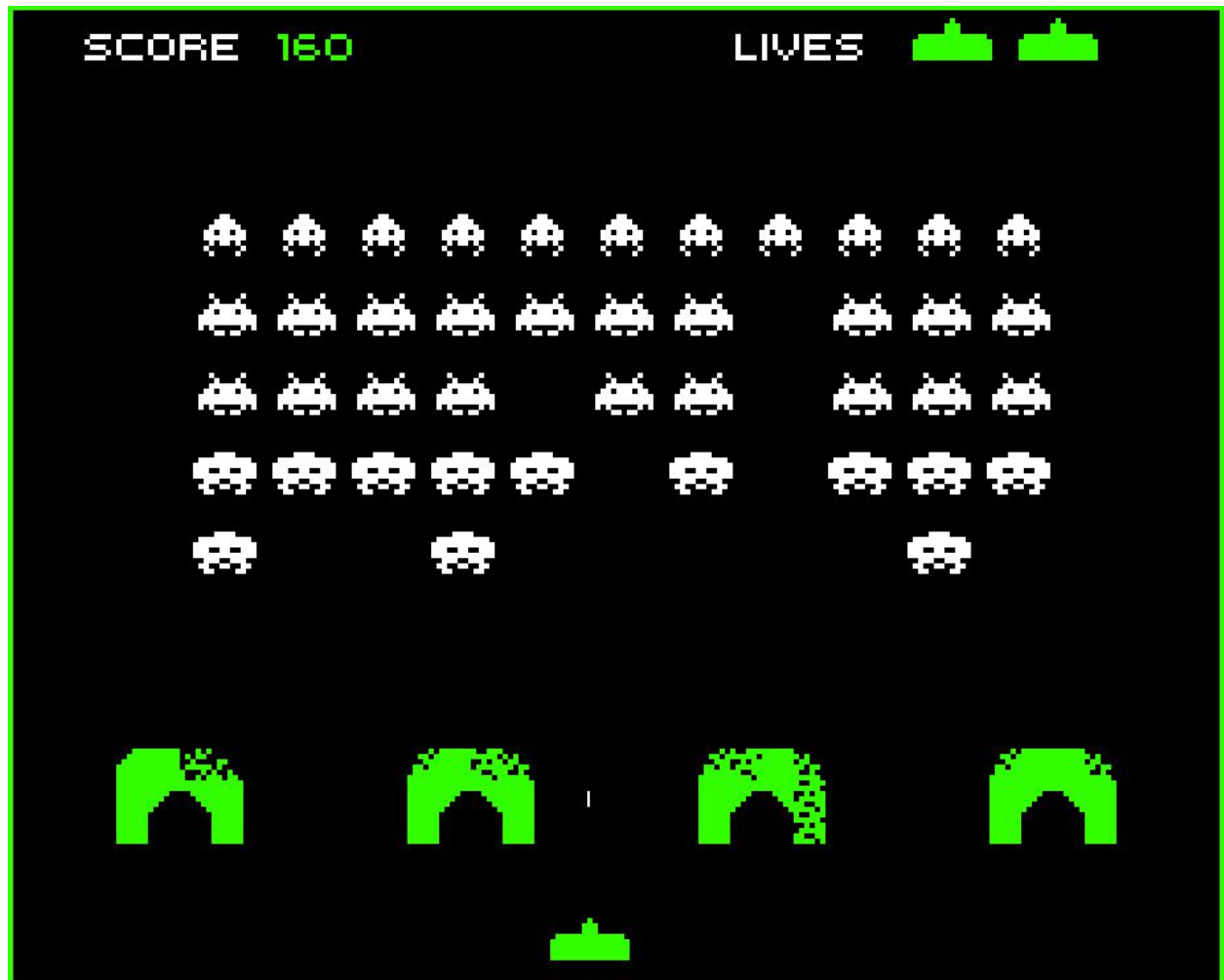
donkey_kong.png - <https://www.flickr.com/photos/vsellis/26889203681/in/photostream/>



pong.png - <https://commons.wikimedia.org/wiki/File:Pong.png>



space_invaders.png - <https://www.flickr.com/photos/bagogames/14750012323>

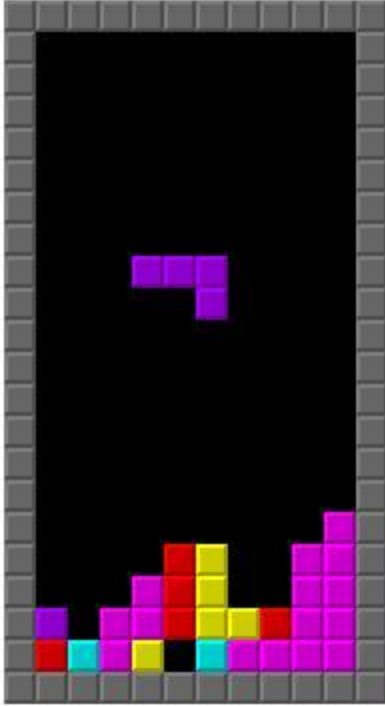


super_mario_kart.png -

https://commons.wikimedia.org/wiki/File:Mario_Kart_Arcade_GP_2-Player_Versus.jpg



tetris.png - <https://commons.wikimedia.org/wiki/File:Tetris.jpg>



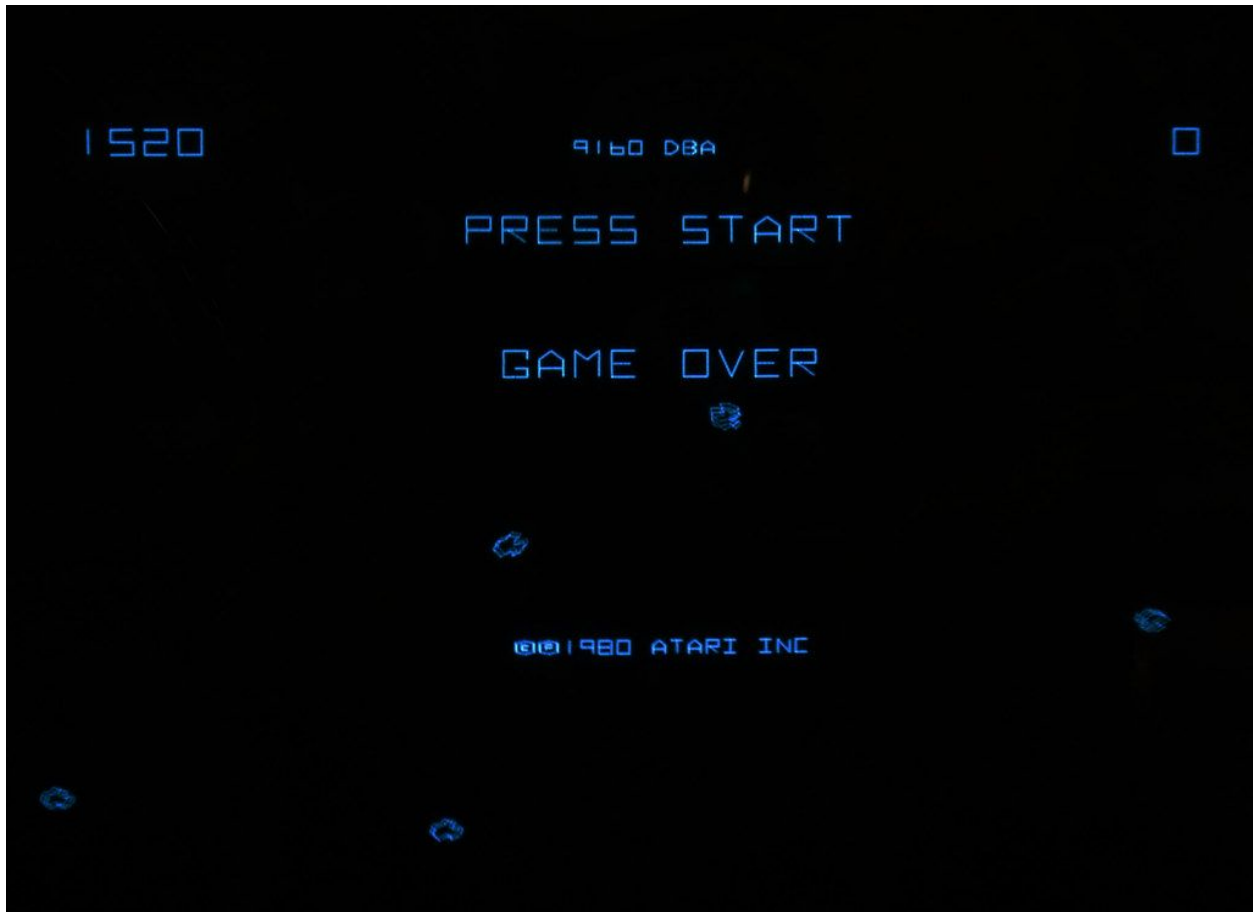
pac_man.png - <https://www.flickr.com/photos/ideonexus/7012962317>

1UP
120

HIGH SCORE
2940



asteroids.png - <https://www.flickr.com/photos/lee-yoshi/15358827250>



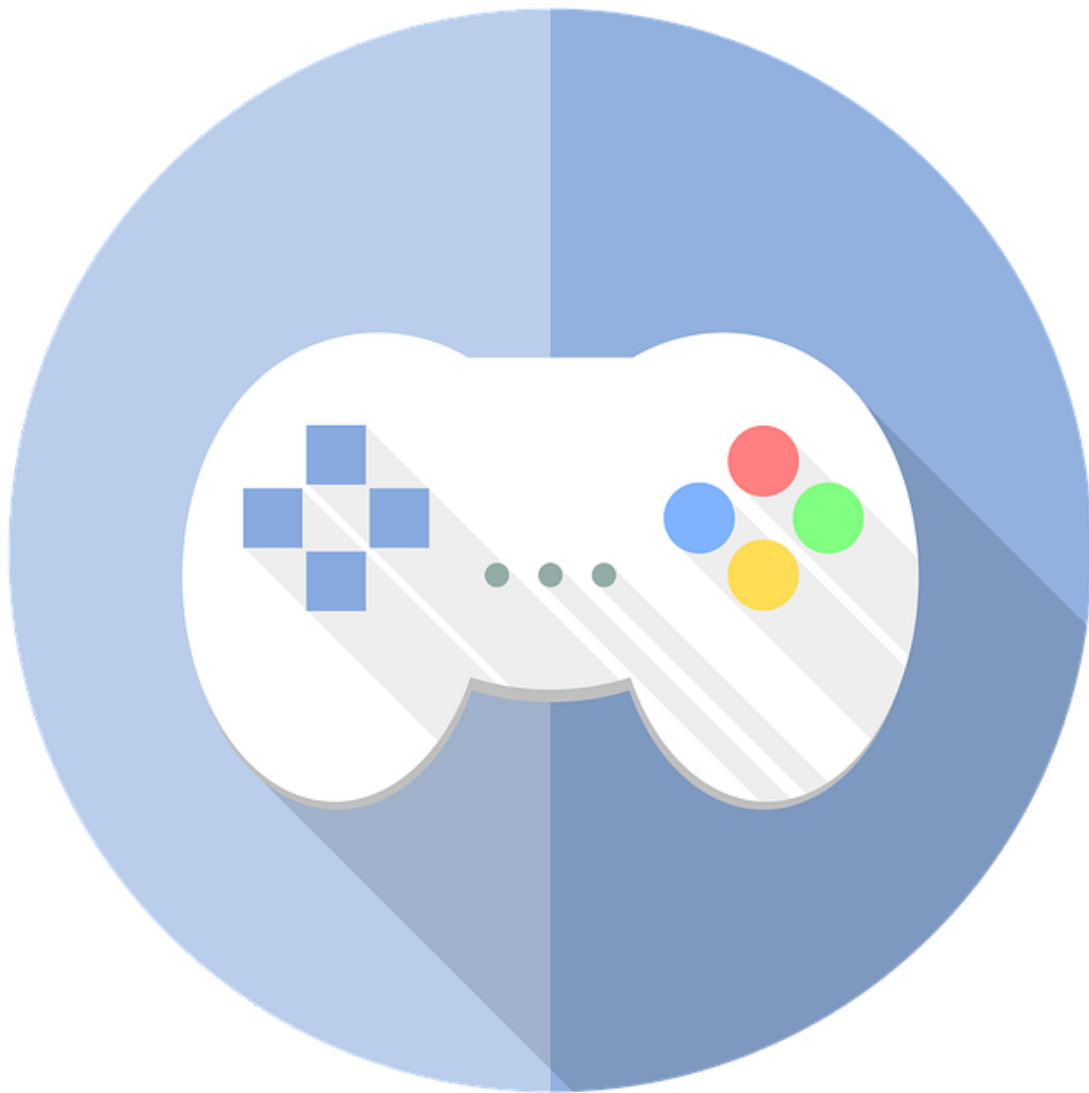
super_mario_bros.png - <https://www.flickr.com/photos/thaivoodoowizard/12352064824>



doom.png - https://commons.wikimedia.org/wiki/File:Freedoom001_01.png



logo.png - <https://pixabay.com/images/search/gamer/>



social_media_image_1.png - https://en.m.wikipedia.org/wiki/File:Instagram_logo_2016.svg



social_media_image_2.png - https://commons.wikimedia.org/wiki/File:F_icon.svg



social_media_image_3.png - <https://commons.wikimedia.org/wiki/File:Twitter.svg>



taito_logo.png - https://ja.wikipedia.org/wiki/ファイル:Taito_logo.svg



atari_logo.png - https://en.wikipedia.org/wiki/File:Atari_logo.svg

ATARI

nintendo_logo.png - <https://ja.wikipedia.org/wiki/ファイル:Nintendo.svg>



alex_logo.png - https://www.flickr.com/photos/eunice_szpillman/2575833305



namco.png - https://commons.wikimedia.org/wiki/File:Namco_logo.png

namco

id_software.png - https://en.wikipedia.org/wiki/Id_Software#/media/File:Id_Software.svg



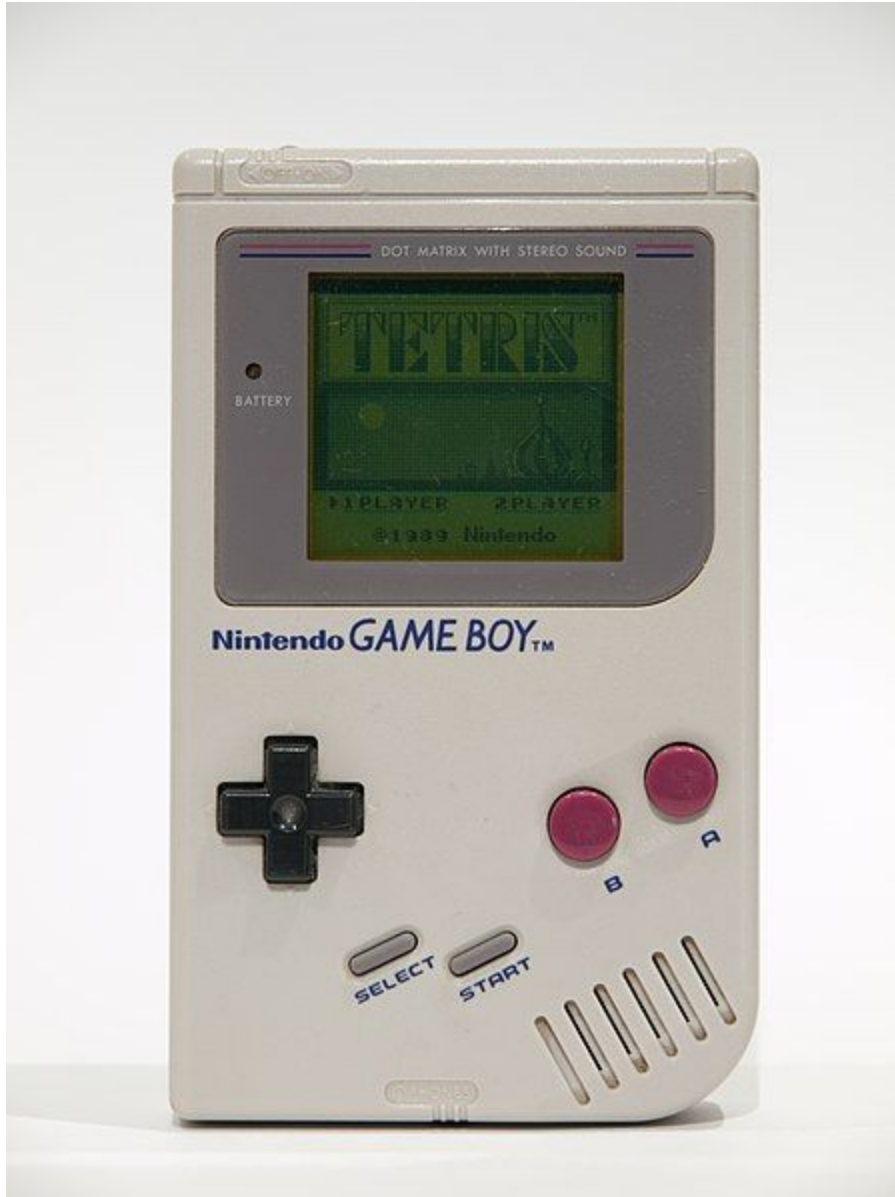
magnavox_odyssey.png - <https://en.wikipedia.org/wiki/File:Magnavox-Odyssey-Console-Set.jpg>



SNES.png - <https://commons.wikimedia.org/wiki/File:SNES-Mod1-Console-Set.jpg>



game_boy.png - https://commons.wikimedia.org/wiki/File:Tetris_on_Game_Boy.jpg



atari_2600.png - <https://commons.wikimedia.org/wiki/File:Atari-2600-Wood-4Sw-Set.jpg>



donkey_kong_console.png - https://commons.wikimedia.org/wiki/File:Donkey_Kong_arcade.jpg



MS_DOS.png - https://en.wikipedia.org/wiki/IBM_PC_compatible



```
22-System Information: Advanced Ballistics, UCJ Cape 1987, 1988, Peter Norton

Computer Name: IBM/PC
Operating System: DOS 3.86
Ballistics Ballistics: October 27, 1987
Ballistics Ballistics: Serial Ports: 1
Ballistics Ballistics: Parallel Ports: 2
Video Display Adapter: Monochrome (VGA)
Current Video Mode: 640 x 480 Monochrome
Available Disk Drives: 2 A: - C:

200 reports 640 0-bytes of memory:
  64 0-bytes used by DOS and resident programs
  136 0-bytes available for application programs
  A search for active memory failed.
  64 0-bytes available memory (A) has 6400-6400
  136 0-bytes available memory (A) has 6400-6400
  200 0-bytes available memory (A) has 6400-6400
  200 0-bytes available memory (A) has 6400-6400

Computing Index (C1), relation to IBM/PC: 1.0
Risk Index (R1), relation to IBM/PC: Not computed. No data specified
Performance Index (P1), relation to IBM/PC: Not computed
C:\>
```