

# Museum Imaginarium: Project Report for COMP5411

Kerim Kochekov

*Department of Computer Science*

*Hong Kong University of Science and Technology,  
Hong Kong*

Hitarth Singh

*Department of Computer Science*

*Hong Kong University of Science and Technology,  
Hong Kong*

**Abstract**—In this report, we will discuss our COMP5411 course project called *Museum Imaginarium*. Museum Imaginarium is a 3D open-world online game. The main character of our game is a minion ghost who travels through the dynamically generated 3D museum. The images on the frames in the museums are retrieved from the internet and allow the user to enjoy the virtual museum with images of their interest.

**Index Terms**—Virtual Museum, ThreeJS, WebGL Rendering,

## I. INTRODUCTION

A lot of real museums are nowadays providing Virtual Museum facility that allows one to see the museum from the comfort of their homes [1]. As useful as the museums are for knowing more about history and culture, it is quite often the case that we don't find in the museums what we wanted to see. This has given rise to our open-world 3D virtual museum called Museum Imaginarium.

Unlike many other virtual museums which require the user to install an application, Museum Imaginarium can be run online on any modern browser with a click of a button. Museum Imaginarium allows the user to enter keywords related to topics of their interests, e.g., cats, dogs, apple, etc., and our game generates a museum with artifacts related to the entered keywords.

We believe our game could also serve as an educational tool by allowing children, as well as adults, to learn about various things by looking at pictures related to that keyword in a game-like setting.

## II. IMPLEMENTATION DETAILS

In this section we will explain our implementation details as follows:

- 1) We used HTML and CSS as the backbone and style of the project.
- 2) We implement our game in pure JavaScript. We employed the ThreeJS library [2] and jQuery.
- 3) We used Skybox to create a 3D surrounding in the game, and it makes the scene look more realistic. As you can see in Fig. 1, Skybox consists of 6 images which correspond to 6 faces of a cube. Our whole game world is then inside this box.

ThreeJS for the awesome open-source library!

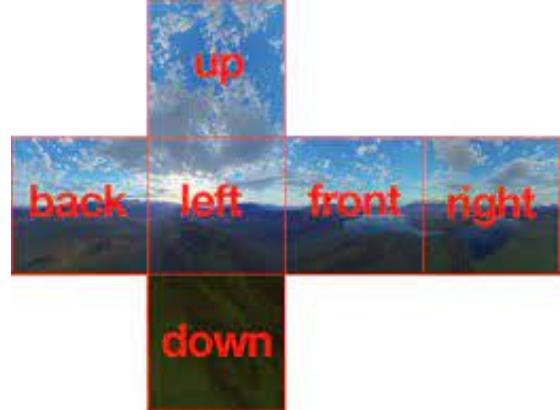


Fig. 1. Skybox Illustration

- 4) For image retrieval we used inferred the Pixabay database. Pixabay has 2.7 million+ in high-quality images, so we used PixaBay API to retrieve images related to user-entered keywords. We use jQuery for AJAX [3] and pass the keywords to PixaBay API, and it returns a JSON object with a link to the 20 most relevant images. We select the first 5-6 images and display them in our virtual museum.

## III. NOTABLE FEATURES

We list the main features of our games in the following:

- 1) Our game allows the user to see images of their interest in the virtual museum.
- 2) It is easy to use and doesn't require any installation. One can run it on any modern web browser.
- 3) The project itself is developable and promising. In the future, if we can put enough time into it, we can add more features and apply some advanced graphics techniques together with smart and fast algorithms. Even some GPT-3 [4] algorithms can be included to present images that have not been seen by anyone before.

## IV. HOW TO RUN THE GAME?

The following points will guide the user on how to run our game.

- 1) Open the file *index.html* using either a local or production server.

- 2) Player will see the greeting screen. Enter the keywords that you are interested in and press the "Enter" button. You can go back to the home screen by pressing "R" button at the anytime of the game. Either way, we included instructions on the greeting page, so please refer to them.
- 3) You will be directed to our web-based game.
- 4) The nicely designed moving guide will greet you to direct you on your movements.
- 5) Player rotates on the current position by using the Left and Right arrow keys of the keyboard.
- 6) Player moves forward and backward by pressing the Up and Down arrow key, respectively.

## V. RESULTS

In this section, we will show screenshots of our game.

- 1) **Greeting Screen.** See the Figure 2



Fig. 2. Greeting Screen

- 2) **Start Screen.** See the Figure 3



Fig. 3. Start Screen

- 3) **In the Museum.** See the Figure 4

## VI. MAIN CHALLENGES FACED DURING THE IMPLEMENTATION

In this section, we will discuss the main challenges that we faced during our implementation of this project.

- 1) We could not find a direct way to use Google API within Javascript, and it also had a very limited number of free requests. We wanted to use pure JavaScript so it can run on any machine without the need for a server.



Fig. 4. Inside the Museum View

- 2) We faced difficulty while implementing our First Person Camera module which allows the minion to move around. It was particularly challenging to make sure that the movement is smooth as we didn't use any library for these functions.
- 3) Loading and rendering new objects were slowing down the game, so we limited the number of objects by 5. We can increase this number in the future with some specific algorithms (do not load the objects that we do not see or load objects with lower resolution for the mesh).

## VII. FUTURE WORK AND FURTHER IMPLEMENTATION

Due to the limitation of time, we were not able to implement many features that we planned our game to have. We hope to add these features in the future. In the following list we state our future goals:

- 1) Try to achieve a more professional first-person view and accurate control-movement system.
- 2) Analysis of different light schemes in the ThreeJS library to have realistic shadows and view.
- 3) Changing the architecture of the museum dynamically depending upon how big or small the museum the user wants to tour. This feature should create well-connected rooms and multiple floors through which the player can move and discover new objects.
- 4) Dynamically arranges the 3D objects around the room in each run of the game so that player never gets bored.
- 5) To have animated objects in the game world that can make the game more interesting.
- 6) Have a goal in the game like collecting hidden flags and giving points for this to the player. This will make sure that the user tries to go through all the rooms and hidden doors of the game and tries to collect all the flags. We believe this will make the game more thrilling.
- 7) To allow the user to take screenshots of the museum from at any given point.
- 8) With the help of the ThreeJS library, support the VR or AR devices to enjoy a virtual museum from a realistic perspective.
- 9) Try to integrate ML models to generate images to present the museum and not have copyright issues.
- 10) We will like to integrate Google Image API with our game to have more relevant images in our game.

## VIII. CONCLUSION

In this report, we discussed our game Museum Imaginarium. It is an enjoyable game that allows the user to tour through a museum of their imagination, hence the name. It was the first breakthrough for some members of our team in the Computer graphics area and Web-GL platform. Hence the development of this game was a nice experience for them as well. We hope to add many more features to the game in the future to make it more interesting and playable.

## ACKNOWLEDGMENT

We will like to thank our friends and family for their support! We will also like to thank Prof. Sanders for teaching and allowing us to use the Three.js library!

## REFERENCES

- [1] TravelandLeisure, <https://www.travelandleisure.com/attractions/museums-galleries/museums-with-virtual-tours>, Retrieved - 20 Nov 2022
- [2] Three.js, <https://threejs.org/>, Retrieved - 20 Nov 2022
- [3] Ajax, [https://en.wikipedia.org/wiki/Ajax\\_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)), Retrieved - 20 Nov 2022
- [4] GPT-3, <https://en.wikipedia.org/wiki/GPT-3>, Retrieved - 21 Nov 2022