Collecting, Analyzing, and Curating Virtual Reality Driven Open Source Projects for Research and Developer Community

CAI, Yongjie 1155158879

WEI, Youlin 1155157186

November 29, 2023

The Chinese University of Hong Kong

Supervisor: Michael R. Lyu

FYP Number: LYU2304

2023-2024 Term 1

Contents

Acknowledgements

We express our sincere thanks to our supervisor, Professor Michael R. Lyu, and our advisor, Ms. Shuqing Li. Their invaluable guidance and profound insights greatly contributed to the progression and completion of our final year project.

1 Introduction

1.1 Background on Virtual Reality (VR) and its potential

Virtual Reality (VR) is a pioneering technology that immerses users in computer-generated environments. Via utilizing headsets or other devices, VR creates an experience that closely mirrors reality. Over the past two decades, interest in Virtual Reality technology has surged among investors, developers, and the general public alike. This interest was notably noted when Facebook purchased Oculus for two million dollars. In 2023, the announcement of Apple's venture into the VR market with the launch of the Apple Vision Pro further kindled enthusiasm in the field [1].

However, despite the rapid advancements and growing interest, Virtual Reality remains an emerging technology grappling with a variety of challenges. Issues of reliability, stability, and user-friendliness continue to pose hurdles to widespread adoption. Any minor error, potentially arising from inadequate VR testing, could significantly impair the user experience, discouraging further interest in Virtual Reality products As VR technology continues to evolve, it is crucial for developers to focus not just on innovation, but also on refining the user experience. Rigorous testing, user-centered design, and persistent refinement of the technology are key to overcoming the current challenges and unlocking the full potential of VR.

1.2 The emergence of the Metaverse concept

The Metaverse [2] is a virtual world that includes all virtual realities, augmented reality, and the internet. It is a shared space where users can interact with one another and the virtual environment in real-time. The Metaverse has received a lot of attention because of its potential for socializing, entertainment, and business applications.

1.3 Importance of reliable virtual reality software in accessing the Metaverse

Delivering a high-quality and immersive virtual reality experience is heavily reliant on the use of reliable software. It ensures that the virtual environment looks and feels accurate, the interactions are seamless and responsive, and the performance is smooth and free of any glitches. The absence of reliable software can cause motion sickness, disorientation, or other discomforts that can negatively affect the user's engagement and enjoyment of the virtual experience [3].

Furthermore, accessing the Metaverse requires the use of robust VR software. The Metaverse aims to create a seamless and interconnected virtual world. Therefore, it demands software systems that can handle the complexities of rendering and networking. Without reliable software, the vision of the Metaverse as a fully immersive and interactive virtual universe may remain unrealized.

1.4 Purpose of the study

1.4.1 Motivation

Despite the rapid evolution and growth of Virtual Reality (VR) technology, the industry has recently encountered significant growth challenges. Facebook, the leading tech enterprise renowned for its investment in VR, rebranded itself as "Meta" in 2021, signaling a strategic shift towards the development of the "metaverse". However, according to Meta's second quarter fiscal report [4] in 2023, its Reality Labs division - responsible for VR and Augmented Reality (AR) hardware and software - reported diminished revenue and income compared to the same period in the preceding year [5]. This stagnation is not unique to Meta; other companies in the sector have also faced similar impediments.

An industry report [6] released by the International Data Corporation on October 6, 2023, revealed a four-year consecutive decline in the global shipment of AR and VR headsets. While the decline can be attributed to a various diverse factors, one prominent cause is the underperformance of VR software. A study [7] analysing complaint trends associated with popular VR games highlighted that, although issues such as cybersickness are less prevalent, software bugs persist, ranking fourth among all complaint categories. Initial expectations were high for Meta's flagship metaverse application, an online VR game. However, critiques have surfaced from Meta's employees, who report a significant number of software bugs that have impaired the application's usability [8].

Notwithstanding the clear evidence of industry decline, there is a noticeable dearth of VR datasets available for developers to utilize in implementing and testing their VR projects. This data scarcity could potentially exacerbate the current state of the industry.

With an aim to enhance the user experience of VR technology, we aim to compile a dataset. This repository of information will serve as a valuable resource for developers seeking to refine their VR projects, and will facilitate the development of tools to further the field's progress.

2 Related Work

2.1 Overview of existing research and development in virtual reality software

Virtual Reality (VR) has become a popular topic in recent research, as discovered through Google Scholar. Researchers are primarily exploring its application in data visualization and analysis. They are using VR to process and display complex datasets, which leads to a better understanding and analysis of the data. In particular, studies are exploring how the interactivity and immersive nature of VR environments can enhance data visualization and provide more comprehensive ways of presenting data [9].

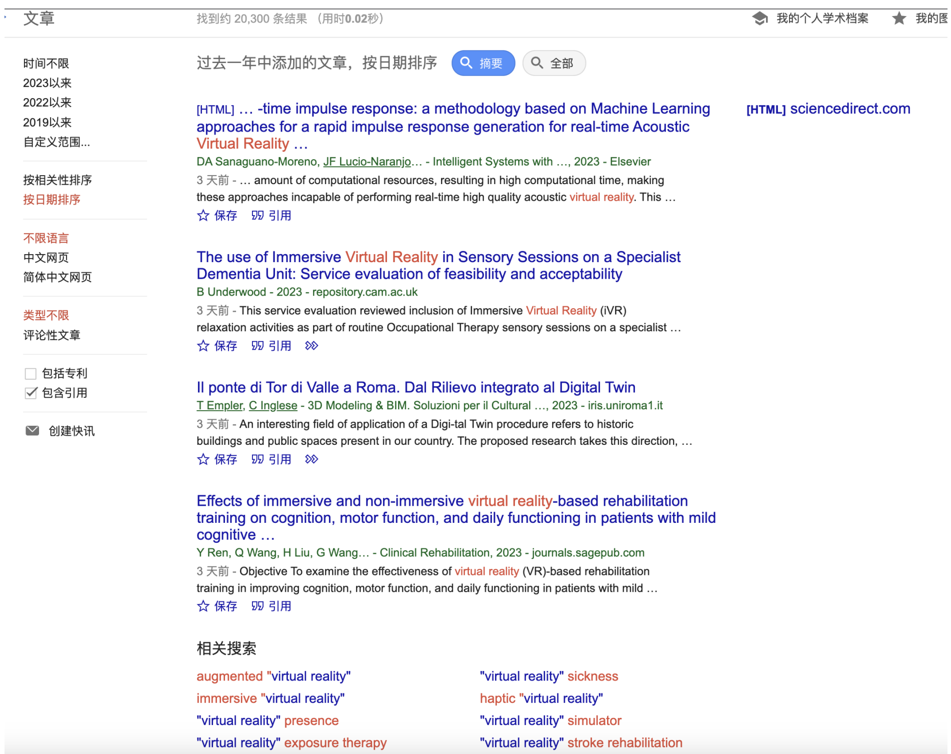


Figure 1: Current Research

Some researchers are examining the visualization and analysis of Earth science datasets using VR technology. They are transforming this data into three-dimensional models within virtual environments, which helps them gain better insights into the Earth's surface topography [10], climate patterns, and other geophysical phenomena. This immersive approach to data exploration helps scientists identify patterns and trends that may be hidden within the data, thus driving advancements in Earth science research.

Other studies are focusing on using VR technology for emotion recognition and eye-tracking research. Researchers are combining VR environments with biometric measurement techniques to observe participants' emotional states and eye movements within virtual scenarios. [11] These studies contribute to a deeper understanding of human emotions and cognitive processes and offer novel research methods and tools for applications such as emotion recognition and psychological health assessment.

Furthermore, VR is being applied in various fields, including education [12], training, and visual analytics. By creating virtual environments and interactive virtual objects, researchers and professionals can conduct simulated experiments, training sessions, and visual analysis of complex

Visualization of large complex datasets using virtual reality [PDF] ieee.org
S Sarathy, K Shujaee, K Cannon - … International Conference on …, 2000 - ieeexplore.ieee.org
… Presented here is a method which uses Virtual Reality (VR) coupled with multi-media …
the much larger datasets typical in the HPC applications. Virtual Reality has been around for …
☆ 保存 ŋŋ 引用 被引用次数：18 相关文章 所有 3 个版本

Slice WIM: a multi-surface, multi-touch interface for overview+ detail exploration of volume datasets in virtual reality [PDF] acm.org
D Coffey, N Malbraaten, T Le, I Borazjani… - … on Interactive 3D …, 2011 - dl.acm.org
… datasets that explores the potential of new interfaces made possible by a virtual reality (VR) …
Slice WIM displays a miniature version of the 3D dataset within a head-tracked stereoscopic …
☆ 保存 ŋŋ 引用 被引用次数：61 相关文章 所有 6 个版本

[HTML] A dataset for emotion recognition using virtual reality and EEG (DER-VREEG): emotional state classification using low-cost wearable VR-EEG headsets [HTML] mdpi.com
NS Suhaimi, J Mountstephens, J Teo - Big Data and Cognitive Computing, 2022 - mdpi.com
… This study aims to use a virtual reality (VR) headset to … VR database that is accessible to
the public and that can potentially assist with emotion recognition studies using virtual reality …
☆ 保存 ŋŋ 引用 被引用次数：27 相关文章 所有 3 个版本

Harnessing the power of immersive virtual reality-visualization and analysis of 3D earth science data sets [PDF] tandfonline.com
J Zhao, JO Wallgrün, PC LaFemina… - Geo-spatial …, 2019 - Taylor & Francis
… Historically, these data sets have been analyzed and quarried on 2D … virtual reality (iVR)
allow for the integration, visualization, analysis, and exploration of these 3D geospatial data sets…
☆ 保存 ŋŋ 引用 被引用次数：30 相关文章 所有 8 个版本

Supporting iterative virtual reality analytics design and evaluation by systematic generation of surrogate clustered datasets [PDF] ieee.org
SK Tadeja, P Langdon… - … and Augmented Reality …, 2021 - ieeexplore.ieee.org
… for synthesizing surrogate clustered datasets for use in virtual reality analytics design and
evaluation… how new insights in VR analytics can be gained using surrogate clustered datasets. …
☆ 保存 ŋŋ 引用 被引用次数：2 相关文章 所有 7 个版本

Eye movement biometrics using a new dataset collected in virtual reality [PDF] acm.org
DJ Lohr, S Aziz, O Komogortsev - ACM Symposium on Eye Tracking …, 2020 - dl.acm.org
… dataset collected in virtual reality (VR) that contains both 2D and 3D eye movement data from
over 400 subjects. We establish that this dataset … an existing, similarly constructed dataset. …

Figure 2: Current Research 2

datasets. This immersive and interactive approach to learning and analysis provides more specific and practical experiences, facilitating better knowledge and skill development.

Based on the Google Scholar search, it appears that existing research lacks content specifically related to virtual reality datasets. Therefore, our research would be highly meaningful in filling this gap. By exploring the creation, processing, and analysis of virtual reality datasets, you can make significant contributions to the application of VR in data visualization and analysis. This will provide richer resources and technologies for other researchers and professionals, thereby driving further advancements in the field of virtual reality.

2.2 Discussion of key contributions and advancements in the field

Firstly, our VR dataset provides an intuitive and immersive data experience. It allows developers and researchers to interact with data in a more intuitive and immersive way, helping them better explain and analyze complex information.

Secondly, the VR dataset drives scientific research and experimen-

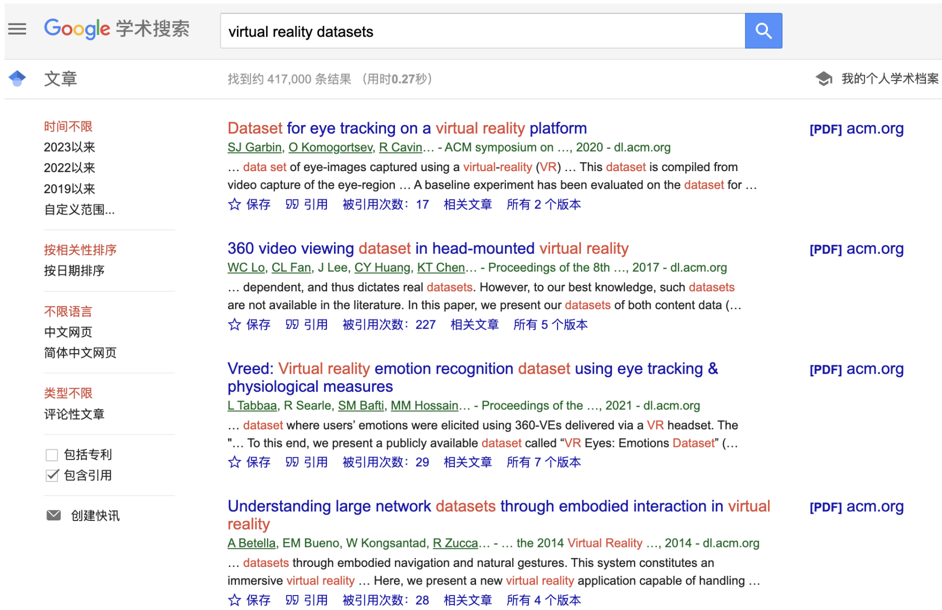


Figure 3: Current Research 2

tation. Our dataset can be used to simulate experimental conditions, observe, test, and analyze phenomena, and explore new hypotheses and theories. By visualizing and interacting with data in a virtual environment, researchers can delve deeper into the relationships and patterns within the data, thereby advancing scientific research.

Similarly, our VR dataset holds great potential value in the field of education and training. It can be used to create virtual laboratories, simulate training scenarios, and interactive learning environments. By utilizing the VR dataset, students and trainers can engage in experiential learning, enhancing learning outcomes and skill development.

Lastly, the VR dataset supports decision-making and problem-solving. By simulating and visualizing data within a virtual environment, decision-makers and problem solvers can better grasp the complexity of issues and explore different solutions and possibilities. This contributes to improved accuracy and effectiveness in decision-making, as well as driving innovation and improvement.Our VR dataset has several valuable applications. Firstly, it provides an immersive and intuitive experience, helping developers and researchers to better analyze and explain complex information.

3 Methodology

3.1 Description of the data collection process from GitHub

3.1.1 The advantages of Using Github for Data Collection

In our data collection process, the objective was to gather information and data related to Virtual Realit (VR) projects. To achieve this, we opted to examine open-source projects focused explicitly on Virtual Reality. This choice was driven by the numerous benefits associated with procuring data from Github, a mainstream open-source platform.

The concept underlying this idea was informed by the methodology employed in the research by Geiger et al. [13]. In their work, a comprehensive list of keywords was devised to filter relevant publications based on their title, thereby facilitating the categorization of collections of Android applications. Given the global popularity of the Android operating system, a multitude of platforms exist from which app-related information can be harvested. However, a comparable wealth of resources for VR development is noticeably absent [7], with a particular dearth of developer communities from which data can be gleaned. Consequently, the decision was made to curate the necessary data from GitHub, one of the most extensive repositories of open-source code.

A primary advantage of Github is its open-source nature, which allows unrestricted access to all public repositories. As a global platform, it hosts an expansive collection of projects from around the world, enhancing the diversity and depth of the data we can cultivate, thereby broadening the scope of our job.

Furthermore, GitHub officially offers an Application Programming Interface (API), which is a convenient tool in our data collection methodology. This API permits us to authenticate as users via an installation access token, enabling us to send frequent data requests. Significantly, the API design allows a high request frequency, with the capability of handling up to 5,000 requests per hour. This high limit facilitates an efficient and robust data collection process, allowing for the extraction of large volumes of data within a condensed time.

Moreover, the datat obtained from GitHub is typically current and precise. The continual updates from developers in the form of new code, bug fixes, and feature enhancements ensure the relevance and timeliness of the data we collect. This ongoing activity allows our job to remain at the forefront of the latest trends and developments within the VR field.

In addition to the raw code, GitHub provides appendix project information. Metrics such as the number of stars, forks, and watchers serve as indicators of a project's popularity and impact within the GitHub community. The number of stars marks the credit of each repository. The ability to access each project's commit history also provides a valuable timeline of the project's evolution, provides materials for inspiration for further analysis of Virtual Reality Test Tool in the next stage. These additional metadata elements add depth to our analysis, offering context and insights beyond the codebase itself.

3.1.2 Identification of Relevant Keywords

The objective of our research is to categorize and analyze repositories related to the domain of Virtual Reality (VR). Our goal is to capture as many VR-related repositories as possible.

We extend our scope beyond the realm of "Virtual Reality" to include overlapping technologies such as "Mixed Reality" (MR) and "Extended Reality" (XR). However, we have excluded "Augmented Reality" (AR) from our keyword list. The rationale behind this exclusion is the operational difference between AR and VR. Specifically, AR applications do not necessarily rely on dedicated VR hardware, instead often utilizing commonly available devices such as smartphones and tablets for overlaying digital information onto the real world. Including AR in our analysis could introduce elements that are not inherently tied to the use of VR peripherals.

To avoid potential omissions due to the use of abbreviations, we have incorporated "VR", "MR", "AR", and "XR" into our keyword list. This strategic approach is designed to offer an extensive and accurate representation of the current state of repositories associated with Virtual Reality technology.

3.1.3 Final scope of repositories

Virtual Reality (VR) has emerged as a focal point, attracting significant interest from developers worldwide. This surge in attention has led to a proliferation of VR-related repositories on GitHub. In order to ensure the quality of our data, it becomes imperative to implement a filtering mechanism to exclude repositories that may lack informative or useful content.

The official GitHub API provides syntax to search for repositories based on their content. This involved defining certain keywords related to VR and related technologies, and subsequently sending requests to the API to retrieve repositories containing these keywords. The response from the GitHub API is a JSON file, containing attributes that provide detailed information of each repository.

Listing 1: Query via Github API

```
base_url = "https://api.github.com/search/
```

```
↪repositories?q={}+in:name,description,readme,
↪topics+stars:{}+pushed:{}..{}&per_page=100".
↪format(keyword, n, since, until)
```

The attributes obtained from the API include the repository title, author, URL, fork count, star count, and the timestamp of the last push operation, among others. These attributes serve as valuable indicators of the relevance, popularity, and activity level of the repositories, thereby aiding in the filtering process.

In our endeavour to maintain a contemporary and relevant dataset, we chose to exclude repositories with a push time preceding January 1, 2020. This decision was predicated on the rapid evolution of VR technology, where older projects may no longer reflect current standards or practices.

Moreover, we established a star count threshold of 10 for the repositories to be included in our study. The star count serves as a measure of a token of repository's reliability and popularity within the GitHub community. Repositories with fewer than 10 stars were thus excluded on the basis that they may not have been sufficiently vetted by the community to warrant inclusion in our analysis.

```
"html_url": "https://github.com/rokups/virtual-reality",
"description": "Stealthy backdoor for Windows operating systems",
"fork": false,
"url": "https://api.github.com/repos/rokups/virtual-reality",
"forks_url": "https://api.github.com/repos/rokups/virtual-reality/forks",
"keys_url": "https://api.github.com/repos/rokups/virtual-reality/keys{/key_id}",
"collaborators_url": "https://api.github.com/repos/rokups/virtual-reality/collaborators{/collaborator}",
"teams_url": "https://api.github.com/repos/rokups/virtual-reality/teams",
"hooks_url": "https://api.github.com/repos/rokups/virtual-reality/hooks",
"issue_events_url": "https://api.github.com/repos/rokups/virtual-reality/issues/events{/number}",
"events_url": "https://api.github.com/repos/rokups/virtual-reality/events",
"assignees_url": "https://api.github.com/repos/rokups/virtual-reality/assignees{/user}",
"branches_url": "https://api.github.com/repos/rokups/virtual-reality/branches{/branch}",
"tags_url": "https://api.github.com/repos/rokups/virtual-reality/tags",
"blobs_url": "https://api.github.com/repos/rokups/virtual-reality/git/blobs{/sha}",
"git_tags_url": "https://api.github.com/repos/rokups/virtual-reality/git/tags{/sha}",
"git_refs_url": "https://api.github.com/repos/rokups/virtual-reality/git/refs{/sha}",
"trees_url": "https://api.github.com/repos/rokups/virtual-reality/git/trees{/sha}",
"statuses_url": "https://api.github.com/repos/rokups/virtual-reality/statuses/{sha}",
"languages_url": "https://api.github.com/repos/rokups/virtual-reality/languages",
"stargazers_url": "https://api.github.com/repos/rokups/virtual-reality/stargazers",
"contributors_url": "https://api.github.com/repos/rokups/virtual-reality/contributors",
"subscribers_url": "https://api.github.com/repos/rokups/virtual-reality/subscribers",
"subscription_url": "https://api.github.com/repos/rokups/virtual-reality/subscription",
"commits_url": "https://api.github.com/repos/rokups/virtual-reality/commits{/sha}",
"git_commits_url": "https://api.github.com/repos/rokups/virtual-reality/git/commits{/sha}",
"comments_url": "https://api.github.com/repos/rokups/virtual-reality/comments{/number}",
"issue_comment_url": "https://api.github.com/repos/rokups/virtual-reality/issues/comments{/number}",
"contents_url": "https://api.github.com/repos/rokups/virtual-reality/contents/{+path}",
"compare_url": "https://api.github.com/repos/rokups/virtual-reality/compare/{base}...{head}",
"merges_url": "https://api.github.com/repos/rokups/virtual-reality/merges",
"archive_url": "https://api.github.com/repos/rokups/virtual-reality/{archive_format}{/ref}",
"downloads_url": "https://api.github.com/repos/rokups/virtual-reality/downloads",
"issues_url": "https://api.github.com/repos/rokups/virtual-reality/issues{/number}",
"pulls_url": "https://api.github.com/repos/rokups/virtual-reality/pulls{/number}",
"milestones_url": "https://api.github.com/repos/rokups/virtual-reality/milestones{/number}",
"notifications_url": "https://api.github.com/repos/rokups/virtual-reality/notifications{?since,all,participating}",
"labels_url": "https://api.github.com/repos/rokups/virtual-reality/labels{/name}",
"releases_url": "https://api.github.com/repos/rokups/virtual-reality/releases{/id}",
"deployments_url": "https://api.github.com/repos/rokups/virtual-reality/deployments",
"created_at": "2019-02-06T11:29:22Z",
"updated_at": "2023-11-24T09:12:01Z",
"pushed_at": "2020-02-13T14:11:36Z",
"git_url": "git://github.com/rokups/virtual-reality.git",
"ssh_url": "git@github.com:rokups/virtual-reality.git",
"clone_url": "https://github.com/rokups/virtual-reality.git",
"svn_url": "https://github.com/rokups/virtual-reality",
"homepage": "",
"size": 503,
"stargazers_count": 272,
"watchers_count": 272,
"language": "C",
"has_issues": true,
```


Figure 4: JSON File that Returned by GitHub API Containin

While not a primary determinant in our filtering process, we also collected data on the fork count of the repositories. The fork count serves as an additional metric of a repository's influence and adoption within the developer community, and will be used as a reference point in our subsequent analyses.

The below figure demonstrates the specific attributes that were selected for extraction from GitHub as part of our research methodology.

```
if(item.get("stargazers_count") >= 10):
    author_name = item.get("owner").get("login")
    repo_name = item.get("full_name")
    html_url = item.get("html_url")
    updated_at = item.get("updated_at")
    created_at = item.get("created_at")
    pushed_at = item.get("pushed_at")
    author_names.append(author_name)
    repo_names.append(repo_name)
    html_urls.append(html_url)
    updated_ats.append(updated_at)
    created_ats.append(created_at)
    pushed_ats.append(pushed_at)
    stars_count = item.get("stargazers_count")
    stars_counts.append(stars_count)
    forks_count = item.get("forks_count")
    forks_counts.append(forks_count)
```

Figure 5: Attributes of Every Repository

3.2 Explanation of the criteria used for repository classification

A collection of over 9,000 repositories was successfully curated, following the application of our set criteria. Our initial data collection process used a crawler script specifically designed to store the fetched repository names and corresponding links in a structured Excel file.

With the primary goal of minimizing data omissions, we incorporated abbreviations in our collection strategy. However, as it is inherent with abbreviations, they often lack clarity and can potentially bear multiple interpretations. For example, the abbreviation "VR" could be deciphered as "Virtual Reality" or "Vehicle Routing". Additionally, "XR" might be a constituent of Unix Commands and not necessarily pertain to virtual reality.

Given the ambiguous nature of such abbreviations, we were compelled to manually inspect each repository to filter out irrelevant entries. This process ensured the relevance and quality of our data.

In conjunction with the data cleansing process, a comprehensive attribute marking procedure was also implemented during the inspection phase. The collected repositories were categorized based on their content from a software engineering viewpoint. For instance, a stand-alone program that can operate under a VR environment was classified as "Application," while a repository containing detailed instructions for operating a VR application was labelled "Tutorial."

Our initial run of labeling was exploratory, as we did not employ a stringent classification system. However, this system was progressively refined and will be further elaborated upon in subsequent sections. The addition of labels to repositories is of significant value as it allows developers to easily locate the types of projects they are seeking, thereby enhancing the overall efficiency of resource discovery.

The process of our investigation included an extensive analysis to determine the presence of code or asset files within each repository. This was an important step in establishing the composition and completeness of each repository. Repositories that were identified as either archived or deprecated were given special attention. The reason for this heightened scrutiny was the recognition of the dynamic nature of dependencies in virtual reality projects, particularly those developed using WebXR technology. Such projects are reliant on a variety of factors which are subject to frequent updates, including hardware specifications, browser compatibility, and API versions.

Without regular updates to accommodate these changes, these projects risk becoming non-functional shortly after being archived, as highlighted by Zubair and Anyameluhor (2021) [14]. This underlines the importance of active maintenance and the challenge of long-term sustainability in the VR realm, especially in the WebXR realm. On a separate note, we conducted a detailed examination of repositories classified as "Application". This involved an in-depth content inspection, followed by the assignment of a categorization scheme. This scheme was designed to differentiate projects based on their scope and complexity, classifying them as demo, medium, or large-scale projects. This categorization provides a clearer understanding of the range and diversity of VR projects found within these repositories.

To ensure the efficiency and accuracy of our inspection process, we adopted a cross-verification method to double-check the content of each repository. This additional step greatly improved our confidence in the data's reliability and accuracy.

Finally, all repositories marked "irrelevant to VR" were excluded from

Item	Options
Related to Virtual Reality or not	yes/no
Related to WebXR	yes/no
Contains Assets/Codes or not	yes/no
Type	(Application, Library, Framework, SDK, Engine, Tutorial, Demo, Tool, Asset, Test, Mod, Plugin...)

Table 1: An example check list.

our dataset. This rigorous process of data cleansing and categorization has not only served to improve the accuracy and reliability of our dataset but also provided invaluable insights into the structure and nature of VR-related repositories on GitHub.

We named this version of the excel file form dataset ”v3-original-project”.

3.3 Discussion of any limitations or challenges encountered during the data collection and classification

During the initial phase of our labeling process, we encountered an unexpected problem. While certain repositories contained our target keywords within their README files, they were in fact primarily associated with Augmented Reality (AR) rather than Virtual Reality (VR). This discrepancy necessitated a refining of our dataset to filter out these irrelevant terms and ensure a more accurate representation of VR-related repositories.

To achieve this, we decided to identify all repositories containing the terms "Augmented Reality" or "AR" in their README files. Once identified, we performed an additional verification step to ascertain whether these repositories were genuinely related to Virtual Reality.

To execute this task, we developed a specialized script. This script differed from our previous data collection methods in that it was designed to specifically select repositories containing our target words from our existing dataset. Consequently, there was no necessity to make additional calls to the GitHub API, which significantly streamlined the process.

Given our objective to optimize efficiency and minimize the time consumption of this task, we chose to leverage Selenium, a powerful web driver renowned for its capabilities in browser automation. Selenium allowed us to automate the process of scanning through the README files of the repositories in our dataset, thereby considerably reducing the

Figure 6: Selenium: A tool that enables broswer automation

manual effort and time required.

This refining phase was a critical part of our data processing pipeline. It ensured the relevance and specificity of our dataset, excluding terms that are not strongly connected to our research target.

4 Data Storage

Although we have conducted a screening process, the number of repositories that successfully passed through our filtering criterion amounted to approximately 1400. These repositories were compiled into an Excel file, as demonstrated below. However, it's worth little that the Excel file was limited to containing only the repository links and some basic attributes, and did not directly include any explicit code.

```
import pandas as pd
import time
import random
from selenium import webdriver
df = pd.read_excel("<file path>")
repo_links = df['URL'].tolist()
target = []
driver = webdriver.Chrome()
search_Text = ["AR", "Augmented Reality"]
exclusive = ["VR", "Virtual Reality", "virtual reality", "xr", "extended reality"]
for url in repo_links:
    try:
        driver.get(url)
    except:
        print("error here")
        continue
    get_source = driver.page_source
    time.sleep(random.uniform(0.5, 1))
    for keyword in search_Text:
        if keyword in get_source:
            for excluded in exclusive:
                if excluded not in exclusive:
                    target.append(url)
                else:
                    pass
        else:
            pass

print(target)
```

Figure 7: Scripts to Exclude AR Only Repository

1	Repo Full Name	Repo Link
2	Magisk-Modules-Repo/PIXELARITY	https://github.com/Magisk-Modules-Repo/PIXELARITY
3	WhiteCrow5/Learning_Notes_Linux	https://github.com/WhiteCrow5/Learning_Notes_Linux
4	phi16/VRC_storage	https://github.com/phi16/VRC_storage
5	krematas/soccerontable	https://github.com/krematas/soccerontable
6	harfang3d/harfang3d	https://github.com/harfang3d/harfang3d
7	OpenHMD/OpenHMD	https://github.com/OpenHMD/OpenHMD

Figure 8: The Repositories Presented in Excel File

For developers who wished to utilize this dataset, the process was rather inconvenient and laborious as it necessitated manually opening each link to access the respective code repository. To enhance the utility and accessibility of the dataset, we conceived a plan to meticulously copy all files and commit histories from each repository on GitHub. Fortunately, Git offers a command, "git clone", which is specifically designed to clone all files and history from a repository.

However, as we set out to implement this solution, we were faced with a new challenge. The sheer volume of repositories, which exceeded 1000, posed a significant problem. The cumulative size of the data from these repositories was considerable and exceeded the available local storage capacity.

In response to this challenge, we sought and were granted access to a remote server provided by CSE department. We subsequently developed a script to clone all repositories on this remote server. While this was a feasible solution, it was not without its challenges. We needed to ensure the script was robust and efficient, capable of handling the extensive volume of data without compromising the integrity of each repository's content.

```python
def slice_df(root_directory, s, f):
    df = pd.read_excel("/home/ylwei1/fyp/forms/v4_projects.xlsx")
    inner_df = df[s:f]
    print(len(inner_df))
    for index, row in inner_df.iterrows():
        name = str(row['Repo Full Name'])
        url = str(row['Repo Link'])
        folder_name = name.replace('/', '_')
        folder_path = os.path.join(root_directory, folder_name)
        os.makedirs(folder_path, exist_ok=True)
        repo_url = f'{"https://@github.com/" + url[19:] + ".git"}'
        time.sleep(random.uniform(0, 0.5))
        process = subprocess.Popen(['git', 'clone', repo_url, folder_path],
                                   stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        stdout, stderr = process.communicate()
        if process.returncode == 0:
            print(f"Clone {name}'s repository into {folder_path} folder.")
        else:
            print(f"Clone failed: {repo_url}")
            print(stdout.decode())
            print(stderr.decode())
```

Figure 9: Code of Cralwer (Command Part)

While the transition to a remote server provided a solution to our storage challenge, it induced a further complication. Each request for a connection to the remote server was restricted to a duration of just two hours. Given the substantial volume of data in our target repositories, this time constraint was insufficient for downloading the entirety of the data.

To expedite the cloning process and circumvent this hurdle, we applied multithreading method. A multithreaded approach enables concurrent execution of two or more parts of a program for maximum utilization of the CPU. Each part of such a program is referred to as a thread, and each thread defines a separate path of execution.

```
if __name__ == '__main__':
    s_time = time.time()
    root_directory = "/data/project/ylwei/github_repos"
    os.makedirs(root_directory, exist_ok=True)
    df = pd.read_excel("/home/ylwei1/fyp/forms/v4_projects.xlsx")
    length = len(df)
    #df = df.iloc[1000:1100, : ]
    process_list = []
    for i in range(10): #1-150 151-300 301-450 ... 1200-1350 1350-1462
        #i: 0-9
        if 150 * (i + 1) <= length: #length = 1462 index = 1463
            s = 150 * i
            f = 150 * (i + 1) - 1
            p = Process(target = slice_df, args = (root_directory, s, f))
            p.start()
            process_list.append(p)
        else:
            s = 150 * i
            f = length - 1
            p = Process(target=slice_df, args=(root_directory, s, f))
            p.start()
            process_list.append(p)
    for p in process_list:
        p.join()
```

Figure 10: Multithread Cloning

The process of cloning such a vast number of repositories was not a trivial task. It required careful planning and execution, particularly considering the broad range of repository sizes and the varying complexity of their histories. However, this approach allowed us to preserve the complete history of each repository, providing a rich and detailed source of data for subsequent analysis and research.

By overcoming these challenges, we were able to create a more comprehensive and accessible dataset. This not only enhances the ease of use for other developers but also provides a more complete picture of the code development and evolution within each repository. This, in turn, can offer invaluable insights for future research in software engineering, particularly in the realm of virtual reality technologies.

The culmination size of collected data reaches 411 gigabytes, which provides sufficient data for developers, serving as a rich and diverse resource.

Figure 11: Size of Dataset

5 Results and Findings

5.1 Presentation of key statistics and insights obtained from the repository analysis

We conducted four rounds of screening and classification, and from 4,423 Github projects, we selected 1,447 projects related to Virtual Reality (VR). We categorized these projects into 122 types of files based on our established classification standards. For instance, Library, Framework, SDK, Engine, etc. To present our classification in a more visual manner, we created a tree diagram to represent each category. Additionally, we included the code that we used to generate the tree and the tools we utilized.

Our dataset classification results are visually represented by Figure 1 to 10.

```
graph TD
   Repo(Repositories) --> VR_Track(VR_Tracking)
      VR_Track(VR_Tracking) --> Motion_T(Motion Tracking)
      VR_Track(VR_Tracking) --> Position_T(Position Tracking)
      VR_Track(VR_Tracking) --> Fullbody_T(Fullbody Tracking)
      VR_Track(VR_Tracking) --> Optical_T(Optical Tracking)
      VR_Track(VR_Tracking) --> Mouth_Tracking(Mouth Tracking)
      VR_Track(VR_Tracking) --> Hand_Tracking(Hand Tracking)
      VR_Track(VR_Tracking) --> Face_Tracking(Face Tracking)
```

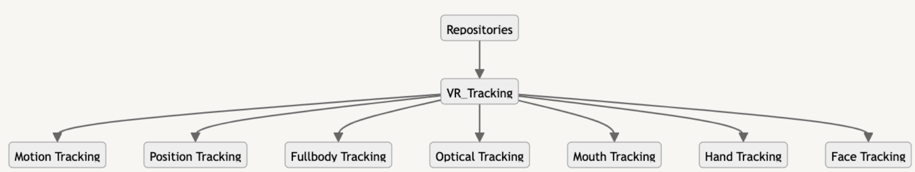


Figure 12: Tracking System

```
graph TD
  Repo(Repositories) --> Documentation(Documentation)
  Documentation(Documentation) --> Other_Doc(Other Documentation)
  Repo(Repositories) --> Specification(Specification)
      Specification(Specification) --> SpecificationC(Specification Collection)
      Specification(Specification) --> APIRe(API, Registry)
      Specification(Specification) --> Other_spec(Other Specification)
  Repo(Repositories) --> Hardware_Introduction(Hardware Introduction)
```

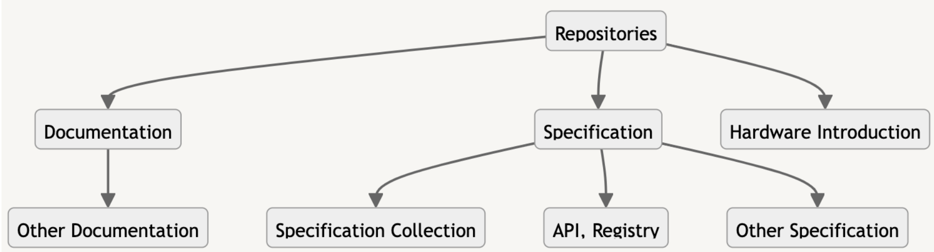


Figure 13: Documentation, Specification, Hardware Introduction

```
graph TD
Repo(Repositories) --> Asset(Asset)
      Asset(Asset) --> Other_Asset(Other Asset)
      Asset(Asset) --> Assetc(Asset Collection)
      Asset(Asset) --> Shader(Shader)

  Repo(Repositories) --> Mod(Mod)
      Mod(Mod) --> Mod_Collection(Mod Collection)
      Mod(Mod) --> Game_mod(Game Mod)
      Mod(Mod) --> Other_mod(Other Mod)
```

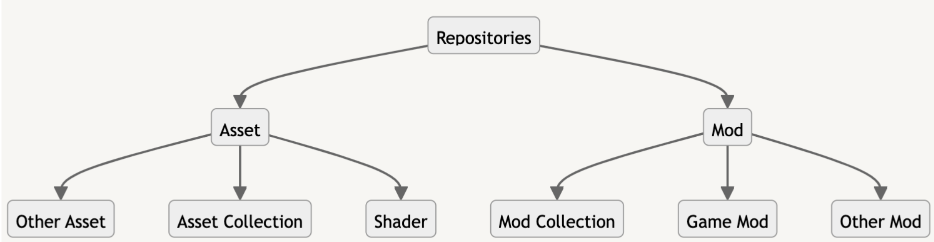


Figure 14: Asset, Mod

```
graph TD
  Repo(Repositories) --> Lib_FK_SDK_Engine(Library, Framework, SDK and Engine)
       Lib_FK_SDK_Engine(Library, Framework, SDK and Engine) --> Library(Library)
       Library(Library) --> LibraryC(Library Collection)
       Library(Library) --> Other_Library(Other library)
       Lib_FK_SDK_Engine(Library, Framework, SDK and Engine) --> Framework(Framewo
           Framework(Framework) --> Test_Framework(Test Framework)
           Framework(Framework) --> RS(Rendering System)
           Framework(Framework) --> OF(Other Framework)
       Lib_FK_SDK_Engine(Library, Framework, SDK and Engine) --> SDK(SDK)
       SDK(SDK)--> Other_SDK(Other SDK)
       Lib_FK_SDK_Engine(Library, Framework, SDK and Engine) --> Engine(Engine)
       Engine(Engine) --> Other_Engine(Other Engine)
```

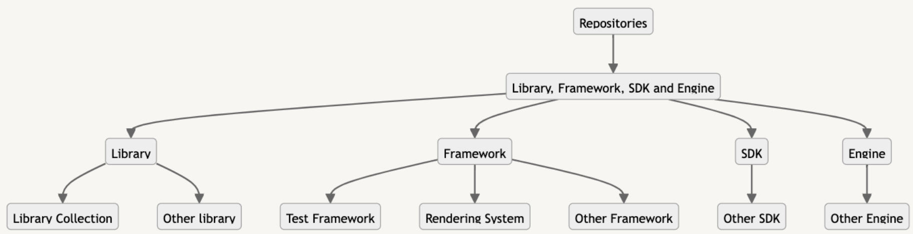


Figure 15: Library, Framework, SDK and Engine

```
graph TD
Repo(Repositories) --> Plugin(Plugin)
    Plugin(Plugin) --> OtherPlugin(Other Plugin)
    Plugin(Plugin) --> UPlugin(Unity Plugin)
    Plugin(Plugin) --> EPlugin(Unreal Engine Plugin)
    Plugin(Plugin) --> IPlugin(Image Plugin)
    Plugin(Plugin) --> VRCP(VRC Plugin)
    Plugin(Plugin) --> OpenVRP(OpenVR Plugin)
```

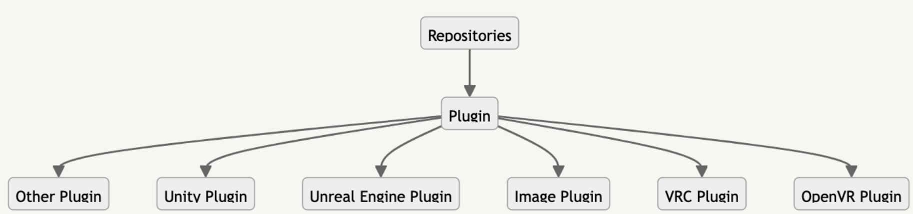


Figure 16: Plugin

```
graph TD
Repo(Repositories) --> App(Application)
    App(Application) --> Appc(Application Collection)
    App(Application) --> Browser(Browser)

    App(Application) --> Demo(Demo)
          Demo(Demo) --> DemoC(Demo Collection)


    App(Application) --> Game(Game)
          Game(Game) --> Gamec(Game Collection)
          Game(Game) --> VR_Game_Port(Ported VR Game)

    App(Application) --> Other_Application(Other Application)
```

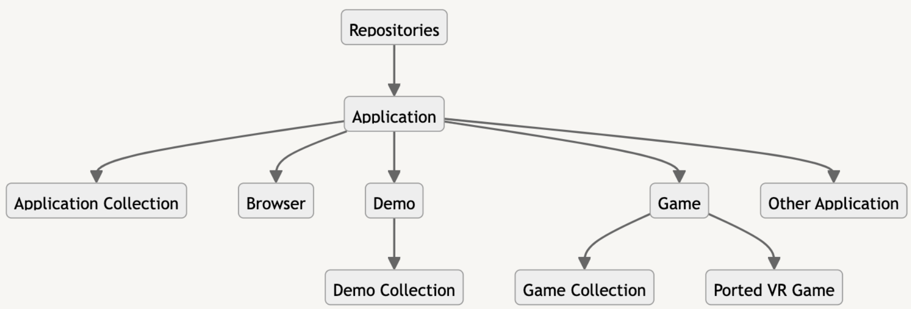


Figure 17: Application

```
graph TD
  Repo(Repositories) --> VR_Resources(VR Related Resources)
      VR_Resources(VR Related Resources) --> Tutorial(Tutorial)
            Tutorial(Tutorial) --> Other_Tutorial(Other_Tutorial)
            Tutorial(Tutorial) --> CourseC(Tutorial Collection)
            Tutorial(Tutorial) --> DIY(Device DIY Tutorial)
            Tutorial(Tutorial) --> Checklist(Checklist for game developing)
     VR_Resources(VR Related Resources) --> RL(Resource List)
            RL(Resource List) --> VRC(VRC Resources Collection)
            RL(Resource List) --> DRL(Development Resource Collection)
            RL(Resource List) --> AwL(Awesome List)
            RL(Resource List) --> Other_Resource(Other Resource List)
```

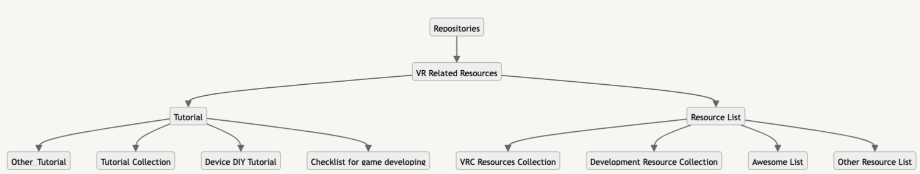


Figure 18: Resource

```
graph TD
  Repository --> Template(Template)
    Template(Template) --> AppTemplate(Application Template)
    Template(Template) --> TemplateCollection(Template Collection)
    Template(Template) --> OtherTemplate(Other Template)
  Repository --> Research(Research)
    Research(Research) --> PaperResult(Paper Result)
  Repository --> Patch(Patch)
  Repository --> Component(Component)
    Component(Component) --> ComponentC(Component Collection)
  Repository --> Driver(Driver)
  Repository --> Emulator(Emulator)
```

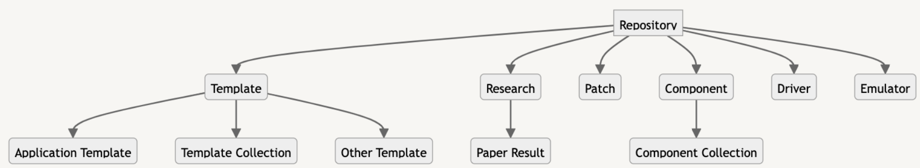


Figure 19: Template

```
graph TD
  Tool(Tool)  --> ToolTest(Test Tool)
  Tool(Tool)  --> DebugTest(Debug Tool)
  Tool(Tool)  --> Tool-bindings(Tool-Bindings)
  Tool(Tool)  --> Inspector(Inspector Tool)
  Tool(Tool) --> ToolCol(Tool Collection)
  Tool(Tool) --> OtherTool(Other Tool)
```

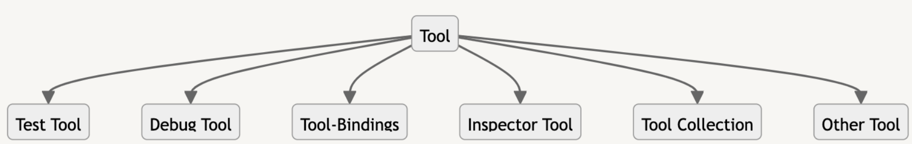


Figure 20: Tool(1)

```
graph TD
  Tool(Tool) --> Utility(Utility)
      Utility(Utility) --> UtilityCollection(Utility Collection)
  Tool(Tool) --> Toolkit(Toolkit)
  Toolkit(Toolkit) -->  UnityToolkit(Unity Toolkit)
  Toolkit(Toolkit) --> MixedRealityToolkit(Mixed Reality Toolkit)
  Toolkit(Toolkit) --> HardwareToolkit(Hardware Toolkit)
  Toolkit(Toolkit) --> Loader(Loader)
  Toolkit(Toolkit) --> OtherToolkit(Other Toolkit)
  Tool(Tool) --> ML(Mod Loader)
  Tool(Tool) --> Client(Client)
  Tool(Tool) --> Polyfill(Polyfill)
```

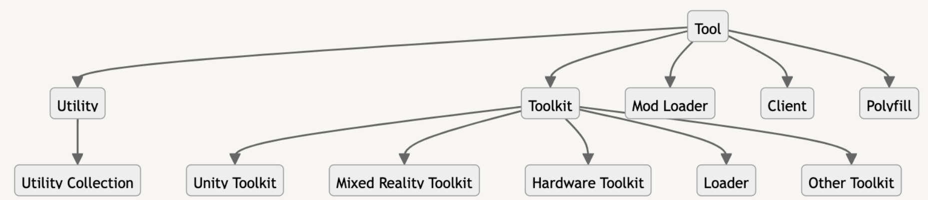


Figure 21: Tool(2)

5.2 Overview of the different categories and their distribution within the dataset

There are 1428 VR repository projects in total, and they are mainly categorized into 5 different categories. Among them, 343 projects fall under the ”Application” category, which indicates that the majority of the projects in the VR field are focused on practical applications like VR games and interactive experiences showing in Figure 11 and Figure 12.

On the other hand, the ”Library” and ”Asset” categories have relatively lower numbers, with 110 and 104 projects, respectively. This suggests that developers in the VR domain tend to use existing libraries and resources instead of developing new ones, which can speed up the development process and improve efficiency.

Furthermore, there are 93 projects in the "Plugin" category and 72 projects in the "Tutorial" category. This indicates that there is a relatively higher interest among developers in creating plugins and providing tutorial resources. Plugin projects may include extensions for VR development frameworks or software, enabling developers to add specific functionalities or integrate with other tools. Tutorial projects serve as valuable resources for other developers, offering guidance and assistance for beginners, skill enhancement, and problem-solving.

To conclude, these statistics shed light on the distribution of different categories within VR repository projects and highlight the focus on practical applications, the trend of resource sharing, and the interest in expanding functionalities and knowledge sharing in the VR domain.

We have conducted a classification of repositories containing code, focusing primarily on the "Application" category. Our classification is based on the size of the repositories. We have identified three specific

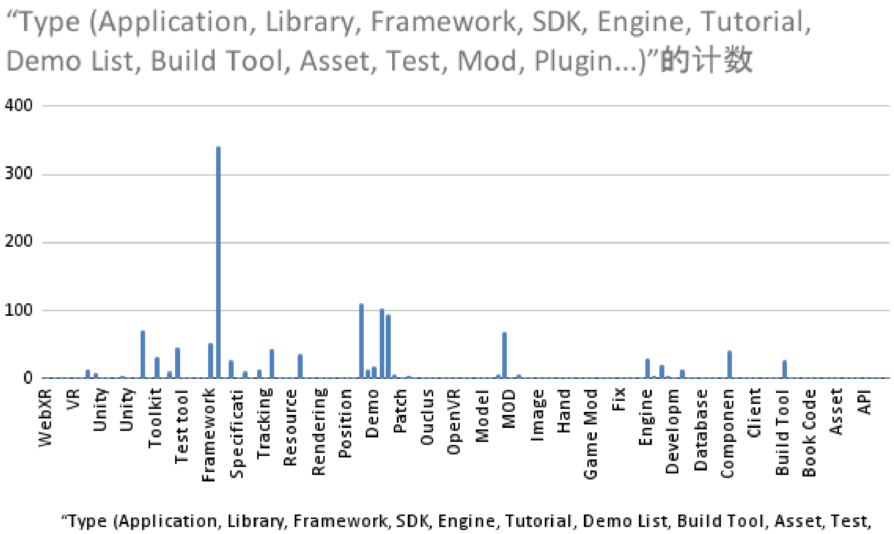


Figure 22: Type

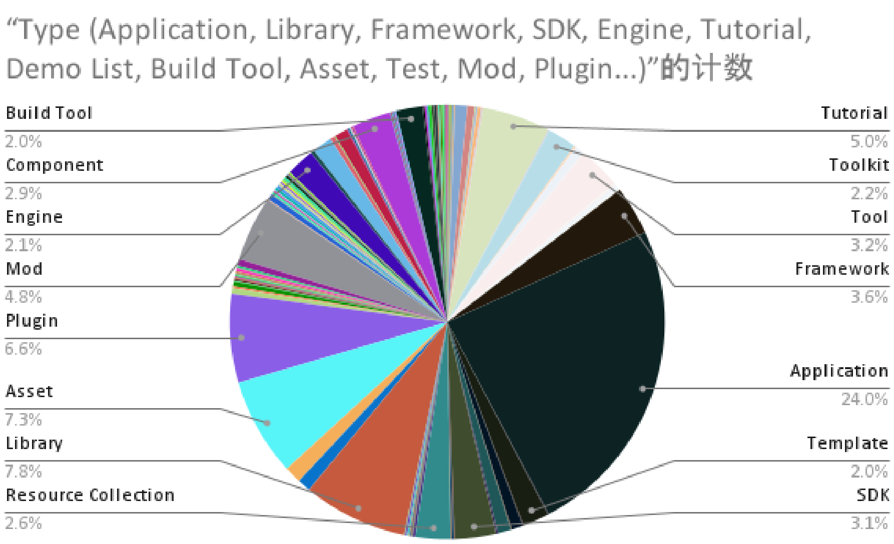


Figure 23: Type 2

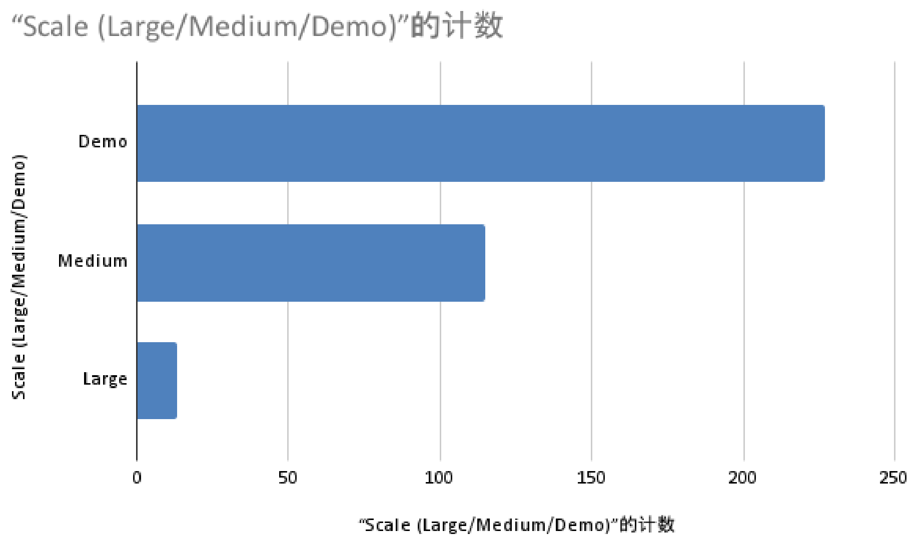


Figure 24: Scale

classifications:

Firstly, we have categorized 225 repositories as Demo level. These projects are relatively small in size and are usually created to showcase specific functionalities or concepts. Demo-level projects serve as demonstrations to illustrate the implementation of certain technologies or methods to other developers. They include basic features or sample code to facilitate understanding and learning.

Secondly, we have classified 110 repositories as Medium level. These projects have a moderate size and tend to be more comprehensive and complex. They encompass a wider range of functionalities and code implementations. Medium-level projects cater to intermediate developers, offering a certain level of challenge and learning opportunities.

Lastly, there are 15 repositories classified as Large level. These projects are sizable and often represent highly complex and comprehensive applications. They feature extensive functionalities and in-depth code implementations. Large-level projects are suitable for experienced advanced developers or can serve as the foundation for large-scale VR projects.

This classification helps developers select appropriate code resources based on their skill levels and project requirements. Projects at different levels provide varying degrees of assistance and guidance, whether for learning, reference, or practical application.

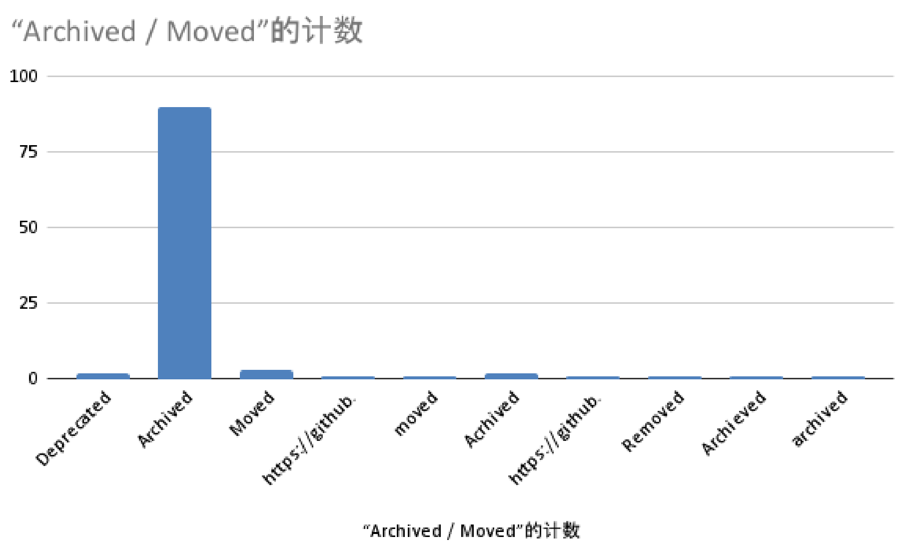


Figure 25: Archived Projects

It has been noted that there are 80 projects which have been marked as "Archived". This simply means that they are no longer in active development, either because they have been completed, discontinued, or replaced by other projects. By being marked as "Archived", it is easier to identify these projects in version control systems, or code repositories, and inform other developers that they are no longer being actively maintained.

It's important to note that these archived projects may have fewer updates or support, but they still have value as they can be used for learning purposes, serve as reference code, or as historical records. They may also contain useful solutions and implementation approaches.

5.3 Identification of notable trends or patterns in virtual reality software development

To illustrate, consider the following statistics. From the period of 2020 to 2021, our dataset comprised a total of 3,589 items. These items were identified based on the presence of the term "Virtual Reality" within their name, description, or README files. It is important to note that in order to gain a holistic view of the total repository count, we did not consider the star count, but instead focused on the presence of the term ”Virtual Reality” for the enumeration of all pertinent repositories.

As we proceed to the period from 2021 to 2022, the data exhibits an interesting increase. The number of relevant items rose to 4,108. Most striking, however, is the substantial surge in the data during the period of 2022 to 2024. The number of items dramatically increased to a total of 16,764, indicating a surprising and significant growth within these two years. This could be possibly attributed to Apple’s announcement of its forthcoming mixed-reality headset, which likely reignited interest and development in the field of Virtual Reality [15].

```
import random
import requests
from datetime import date, timedelta
import time
import json
import pandas as pd

token = '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'
timestamp = ["2020", "2021", '2022..2024']
def get_counts(timestamp):
    for item in timestamp:
        url = "https://api.github.com/search/repositories?q=virtual-reality+in:name,description,readme,topics+pushed:{}&per_page=100".format(item)
        response = requests.get(url, headers={'Accept': 'application/vnd.github.v3+json',
                                              'Authorization': 'token {token}'.format(token=token)})
        jresponse = response.json()
        count_file = json.loads(json.dumps(jresponse))
        total_count = count_file.get("total_count")
        print("The total count of repositories pushed between " + item + " is " + str(total_count ))

get_counts(timestamp)
```

Figure 26: Codes to Count The Number of Repositories

By rearranging the repositories in a descending order based on the start count, we attempted to look for some patterns in the remarkable repositories.

	A	B	C	D	E	F	G	H	I	J
1	Repo Full Name	Repo Link	Check?	Verified?	Related to	Related to Web	Contains	Type (Application, Library, Framework, SDK, Engin		Scale (Lar
5	mrdoob/three.js	https://github.com/mrdoob/three.js	☑	☑	☑	☑	☑	Library		
16	pmndrs/react-three-fiber	https://github.com/pmndrs/react-three-fiber	☑	☑	☑	☑	☑	Library		
20	aframevr/aframe	https://github.com/aframevr/aframe	☑	☑	☑	☑	☑	Application		Medium
21	Unity-Technologies/ml-agents	https://github.com/Unity-Technologies/ml-agents	☑	☑	☑	☐	☑	Toolkit		
26	bkaradzic/bgfx	https://github.com/bkaradzic/bgfx	☑	☑	☑	☑	☑	Rendering Library		
34	bulletphysics/bullet3	https://github.com/bulletphysics/bullet3	☑	☑	☑	☐	☑	SDK		
38	facebookarchive/react-360	https://github.com/facebookarchive/react-360	☑	☑	☑	☐	☑	Framework		
45	olucurious/Awesome-ARKit	https://github.com/olucurious/Awesome-ARKit	☑	☑	☑	☐	☐	Resource Collection		
56	RyanNielson/awesome-unity	https://github.com/RyanNielson/awesome-unity	☑	☑	☑	☐	☑	Resource Collection		
58	microsoft/MixedRealityToolkit-Unity	https://github.com/microsoft/MixedRealityToolkit-Unity	☑	☑	☑	☐	☑	Mixed Reality Toolkit		

Figure 27: Top 10 Repositories

By reorganizing the repositories in descending order based on their star count, we attempted to discern patterns within the dataset, which serves as a measure of their popularity within the GitHub community. A specific focus was placed on the top 10 repositories, which had garnered the highest number of stars.

Within these top-ranking repositories, four were found to be related to WebXR. This prevalence of WebXR-related repositories within the top tier may be indicating an increasing interest in web-based virtual reality technology. This trend suggests a potential shift in the focus of VR development towards more web-centric solutions.

Further analysis revealed that, despite the majority of the collected repositories being categorized as demos and applications, the most popular repositories were predominantly libraries or other developmental tools. This trend could suggest a higher universality of developmental tools and resources compared to stand-alone applications. Developers, while aiming to create unique applications, may find more value in tools and resources that can be used across multiple projects.

Such insights offer valuable implications for future work and research directions. The observed pattern of popularity amongst developmental tools, as opposed to stand-alone applications, suggests that more emphasis could be placed on creating tools for VR-related projects in our subsequent stages. This strategic shift could increase the universality and impact of our contributions to the VR domain, while also aligning with the observed trends within the open-source community.

5.4 Demonstration of the demo created to showcase the dataset

To enhance the presentation of our dataset, we are currently in the process of developing a user-friendly interface. This interface aims to provide an intuitive and efficient way for users to search and explore the content they are interested in.

With our upcoming interface, you will have the ability to filter repositories based on specific categories, ensuring that you find the most relevant resources for your specific requirements. Whether you are a developer working on VR applications, conducting research in VR technology, or simply passionate about VR, our platform will offer a wealth of resources for you to explore and leverage.

Our primary objective is to create a convenient and efficient platform that empowers developers to discover and utilize valuable code and tools. We understand the importance of accessibility and ease of use when it comes to navigating a large dataset, and our interface is being designed with these considerations in mind.

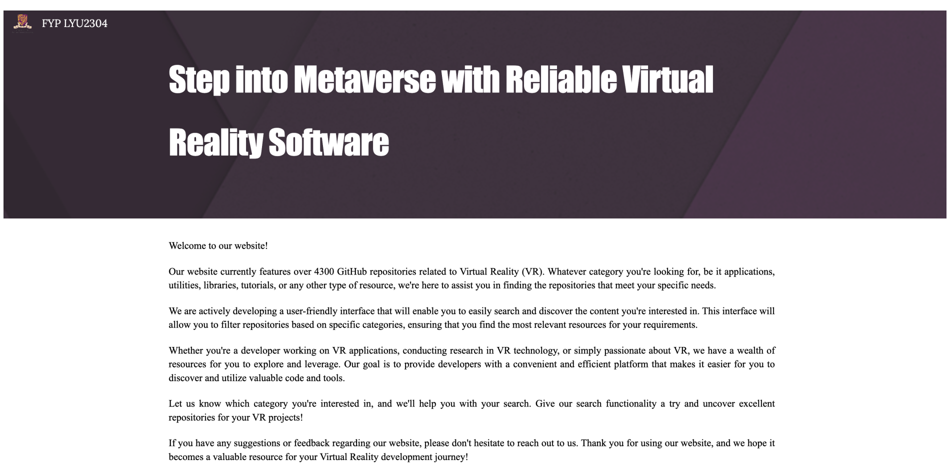


Figure 28: Website

This presentation showcases our dataset table, which is represented in Figure 17.

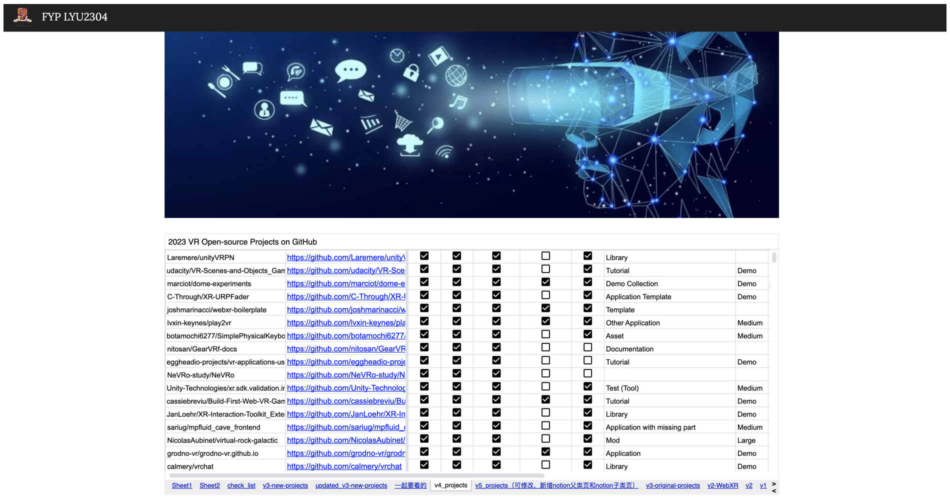


Figure 29: Data sheet

At present, our website has two functions, the first is to search in our database, fuzzy search is also supported, for example, the user input "app", the search module on the website will return all GitHub repository related to application, and automatically download to txt file. The.txt file contains the names and links of all associated repositories.The second feature of this module allows users to upload a text file that was exported by the first function. Once uploaded, the module will assist users in accessing the corresponding GitHub files and retrieve the description files, which will then be returned to the user.

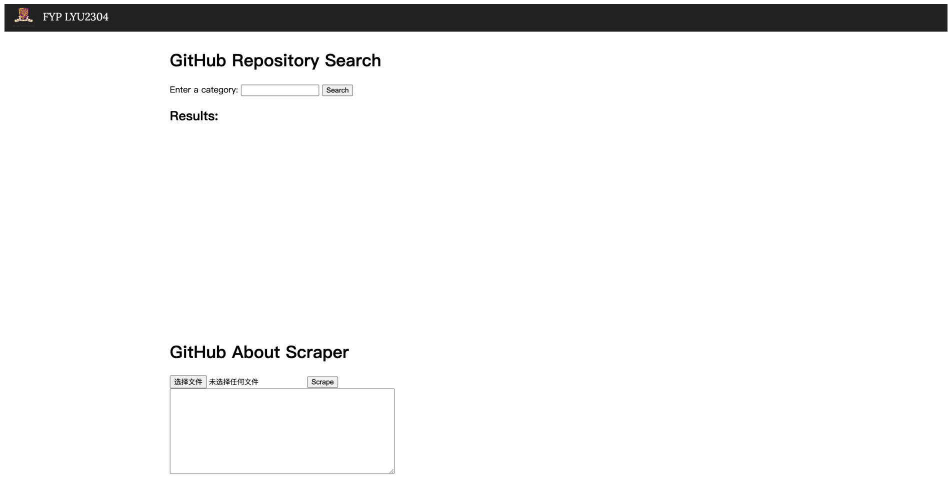


Figure 30: Function showing

Here's an example of how to use the search function.

The following is a sample HTML code.

After locating the file, the website will automatically download it for the user and save it as ”search results”.

This is an example after the file is opened.

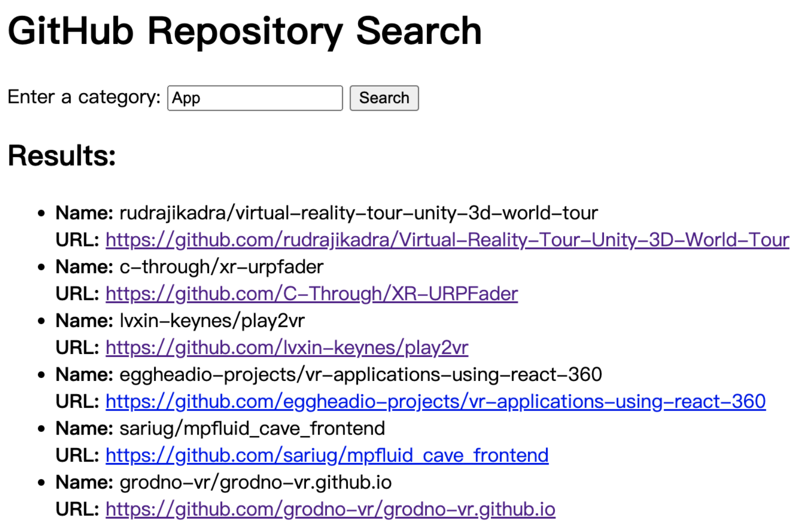
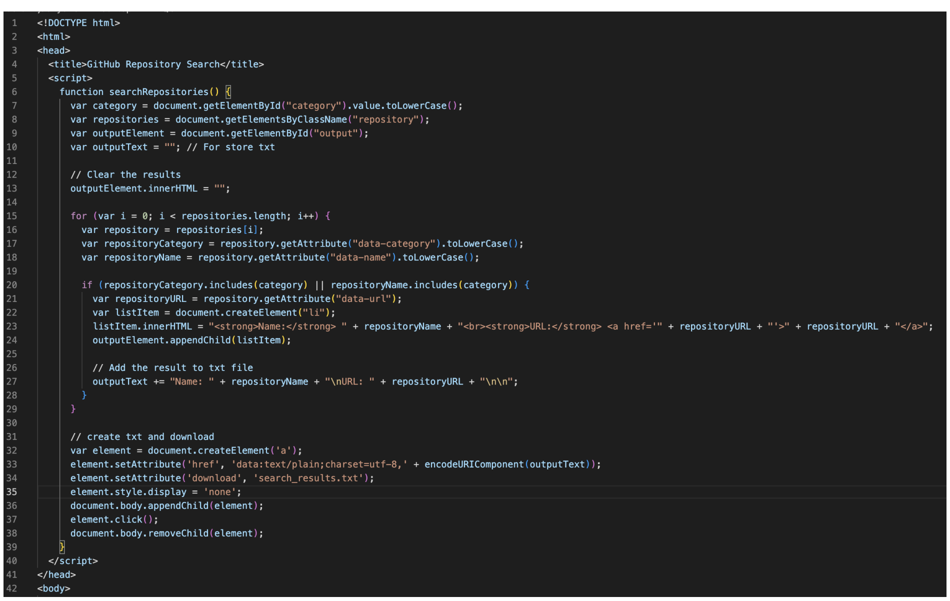


Figure 31: Search Function



```
<!DOCTYPE html>
<html>
<head>
  <title>GitHub Repository Search</title>
  <script>
    function searchRepositories() {
      var category = document.getElementById("category").value.toLowerCase();
      var repositories = document.getElementsByClassName("repository");
      var outputElement = document.getElementById("output");
      var outputText = ""; // For store txt

      // Clear the results
      outputElement.innerHTML = "";

      for (var i = 0; i < repositories.length; i++) {
        var repository = repositories[i];
        var repositoryCategory = repository.getAttribute("data-category").toLowerCase();
        var repositoryName = repository.getAttribute("data-name").toLowerCase();

        if (repositoryCategory.includes(category) || repositoryName.includes(category)) {
          var repositoryURL = repository.getAttribute("data-url");
          var listItem = document.createElement("li");
          listItem.innerHTML = "<strong>Name:</strong> " + repositoryName + "<br><strong>URL:</strong> <a href='" + repositoryURL + "'>" + repositoryURL + "</a>";
          outputElement.appendChild(listItem);

          // Add the result to txt file
          outputText += "Name: " + repositoryName + "\nURL: " + repositoryURL + "\n\n";
        }
      }

      // create txt and download
      var element = document.createElement('a');
      element.setAttribute('href', 'data:text/plain;charset=utf-8,' + encodeURIComponent(outputText));
      element.setAttribute('download', 'search_results.txt');
      element.style.display = 'none';
      document.body.appendChild(element);
      element.click();
      document.body.removeChild(element);
    }
  </script>
</head>
<body>
```

Figure 32: Search code

```
<body>
  <h1>GitHub Repository Search</h1>
  <label for="category">Enter a category:</label>
  <input type="text" id="category">
  <button onclick="searchRepositories()">Search</button>
  <h2>Results:</h2>
  <ul id="output"></ul>

  <script>
    // Repository data
    var repositories = [
      // Add data in here
     {name: "rudrajikadra/Virtual-Reality-Tour-Unity-3D-World-Tour", url: "https://github.com/rudrajikadra/Virtual-Reality-Tour-Unity-3D-World-Tour", category: "Appli
        {name: "sophiabaldonado/layout", url: "https://github.com/sophiabaldonado/layout", category: "Utility"},
        {name: "Twometer/twometer-vr", url: "https://github.com/Twometer/twometer-vr", category: "Hand Tracking"},
        {name: "Laremere/unityVRPN", url: "https://github.com/Laremere/unityVRPN", category: "Library"},
        {name: "udacity/VR-Scenes-and-Objects_Game-Objects", url: "https://github.com/udacity/VR-Scenes-and-Objects_Game-Objects", category: "Tutorial"},
        {name: "marciot/dome-experiments", url: "https://github.com/marciot/dome-experiments", category: "Demo Collection"},
        {name: "C-Through/XR-URPFader", url: "https://github.com/C-Through/XR-URPFader", category: "Application Template"},
        {name: "joshmarinacci/webxr-boilerplate", url: "https://github.com/joshmarinacci/webxr-boilerplate", category: "Template"},
        {name: "lvxin-keynes/play2vr", url: "https://github.com/lvxin-keynes/play2vr", category: "Other Application"},
        {name: "botamochi6277/SimplePhysicalKeyboard", url: "https://github.com/botamochi6277/SimplePhysicalKeyboard", category: "Asset"},
        {name: "nitosan/GearVRf-docs", url: "https://github.com/nitosan/GearVRf-docs", category: "Documentation"},
        {name: "eggheadio-projects/vr-applications-using-react-360", url: "https://github.com/eggheadio-projects/vr-applications-using-react-360", category: "Tutorial
        {name: "NeVRo-study/NeVRo", url: "https://github.com/NeVRo-study/NeVRo"},
        {name: "Unity-Technologies/xr.sdk.validation.input", url: "https://github.com/Unity-Technologies/xr.sdk.validation.input", category: "Test (Tool)"},
        {name: "cassiebreviu/Build-First-Web-VR-Game-Absolute-Beginner", url: "https://github.com/cassiebreviu/Build-First-Web-VR-Game-Absolute-Beginner", category: "
        {name: "JanLoehr/XR-Interaction-Toolkit_Extensions", url: "https://github.com/JanLoehr/XR-Interaction-Toolkit_Extensions", category: "Library"},
        {name: "sariug/mpfluid_cave_frontend", url: "https://github.com/sariug/mpfluid_cave_frontend", category: "Application with missing part"},
        {name: "NicolasAubinet/virtual-rock-galactic", url: "https://github.com/NicolasAubinet/virtual-rock-galactic", category: "Mod"},
        {name: "grodno-vr/grodno-vr.github.io", url: "https://github.com/grodno-vr/grodno-vr.github.io", category: "Application"},
        {name: "calmery/vrchat", url: "https://github.com/calmery/vrchat", category: "Library"},
        {name: "AndreZenner/hand-redirection-toolkit", url: "https://github.com/AndreZenner/hand-redirection-toolkit", category: "Toolkit"},
        {name: "nebriv/VTOLVR-TacviewLogger", url: "https://github.com/nebriv/VTOLVR-TacviewLogger", category: "Mod"},
  {name: "mozilla/webxr-polyfill", url: "https://github.com/mozilla/webxr-polyfill", category: "WebXR polyfill"},
      {name: "Erumite/Eremite_VRC", url: "https://github.com/Erumite/Eremite_VRC", category: "VRC Resource Collection"},
      {name: "Slaynash/VRCModUpdater", url: "https://github.com/Slaynash/VRCModUpdater", category: "VRC Plugin"},
      {name: "Slaynash/VRChatModInstaller", url: "https://github.com/Slaynash/VRChatModInstaller", category: "VRC Plugin"},
      {name: "uezo/aiavatarkit", url: "https://github.com/uezo/aiavatarkit", category: "VRC Plugin"},
      {name: "oestradiol/VRC-Mods", url: "https://github.com/oestradiol/VRC-Mods", category: "VRC Mod"},
      {name: "SimulaVR/Simula", url: "https://github.com/SimulaVR/Simula", category: "VR Desktop"},
      {name: "TheDuriel/DurielsGodotUtilities", url: "https://github.com/TheDuriel/DurielsGodotUtilities", category: "Utility Collection"},
      {name: "vrm-c/UniVRMUtility", url: "https://github.com/vrm-c/UniVRMUtility", category: "Utility"},
      {name: "blepmlem/VRChat-Expression-Utility", url: "https://github.com/blepmlem/VRChat-Expression-Utility", category: "Utility"},
      {name: "shiena/OpenXRRuntimeSelector", url: "https://github.com/shiena/OpenXRRuntimeSelector", category: "Utility"},
      {name: "GeekJosh/BoxVR-Playlist-Manager", url: "https://github.com/GeekJosh/BoxVR-Playlist-Manager", category: "Utility"},
      {name: "TheNewJavaman/unreal-vr", url: "https://github.com/TheNewJavaman/unreal-vr", category: "Utility"},
```

Figure 33: Search code part 2

```
      {name: "OpenVR-Advanced-Settings/OpenVR-AdvancedSettings", url: "https://github.com/OpenVR-Advanced-Settings/OpenVR-AdvancedSettings", category: "Utility"},
      {name: "Caeden117/Curved-UI-Utility", url: "https://github.com/Caeden117/Curved-UI-Utility", category: "Utility"},
      {name: "DavidRisch/steamvr_utils", url: "https://github.com/DavidRisch/steamvr_utils", category: "Utility"},
      {name: "GodotVR/TiltFiveGodot", url: "https://github.com/GodotVR/TiltFiveGodot", category: "Utility"},
      {name: "Das-Kampfnudelholz/Quest-VRC-modder", url: "https://github.com/Das-Kampfnudelholz/Quest-VRC-modder", category: "Utility"},
      {name: "saturday06/VRM-Addon-for-Blender", url: "https://github.com/saturday06/VRM-Addon-for-Blender", category: "Utility"},
      {name: "Lewisscrivens/BetterPortals", url: "https://github.com/Lewisscrivens/BetterPortals", category: "Unreal Engine Plugin"},
      {name: "sigtrapgames/VrTunnellingPro-UE4", url: "https://github.com/sigtrapgames/VrTunnellingPro-UE4", category: "Unreal Engine Plugin"},
      {name: "fweidner/UE4-Plugin-OffAxis", url: "https://github.com/fweidner/UE4-Plugin-OffAxis", category: "Unreal Engine Plugin"},
      {name: "ruyo/VRM4U", url: "https://github.com/ruyo/VRM4U", category: "Unreal Engine Plugin"},
      {name: "guiglass/Unreal_Engine_5_Live_Production_Example", url: "https://github.com/guiglass/Unreal_Engine_5_Live_Production_Example", category: "Unreal Engine

        // Add more repositories here...
      ];

      // create repositories
      repositories.forEach(function(repo) {
        var repositoryElement = document.createElement("li");
        repositoryElement.className = "repository";
        repositoryElement.setAttribute("data-name", repo.name);
        repositoryElement.setAttribute("data-url", repo.url);
        repositoryElement.setAttribute("data-category", repo.category);
        repositoryElement.style.display = "none";
        document.body.appendChild(repositoryElement);
      });
    </script>
  </body>
</html>
```

Figure 34: Search code part 3

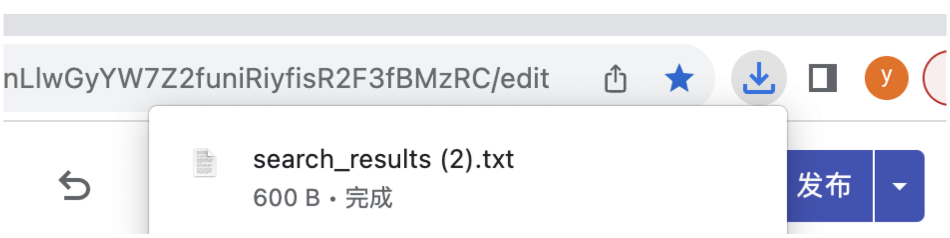


Figure 35: Document Downloading

search_results (2).txt

```
Name: rudrajikadra/virtual-reality-tour-unity-3d-world-tour
URL: https://github.com/rudrajikadra/Virtual-Reality-Tour-Unity-3D-World-Tour

Name: c-through/xr-urpfader
URL: https://github.com/C-Through/XR-URPFader

Name: lvxin-keynes/play2vr
URL: https://github.com/lvxin-keynes/play2vr

Name: eggheadio-projects/vr-applications-using-react-360
URL: https://github.com/eggheadio-projects/vr-applications-using-react-360

Name: sariug/mpfluid_cave_frontend
URL: https://github.com/sariug/mpfluid_cave_frontend

Name: grodno-vr/grodno-vr.github.io
URL: https://github.com/grodno-vr/grodno-vr.github.io
```

Figure 36: Txt file

6 Future Work

6.1 Exploration of potential directions for future research and development

Datasets play a crucial role in comprehending the real world, making data-driven decisions, and driving innovation. [16] They provide us with insights into objective facts, enabling us to identify patterns, analyze trends, and solve problems.

In the future, our dataset could offer researchers and scholars detailed information about VR technology and applications. By searching through the GitHub repositories, they can examine the code, documentation, and related discussions. For instance, they can access relevant discussions and technologies by reading the ”read me” files of each GitHub repository. This can help researchers understand the development trends, popular technologies, and innovative aspects of VR.

Moreover, the code, demonstrations, and applications found in our dataset are highly valuable for learning and understanding VR development and technologies. [17]They serve as references for developers to enhance existing VR applications and tools. Thanks to the open-source nature of GitHub, developers can draw inspiration and even use these codes and knowledge to expedite their own project development. They can also build upon the work of others and propose new innovative ideas.

Lastly, we encourage the community to collaborate with us and actively use and contribute to our dataset. Sharing datasets promotes communication and collaboration among developers. They can share their codes, solutions, and experiences, which fosters cooperation within the open-source community and propels the evolution of the metaverse.

6.2 Potential Use of Databases for Debugging in Virtual Reality Projects

As part of our ongoing research efforts, we aim to not only curate a dataset but also to exhibit how this can be practically utilized by developers. Therefore, one of our primary future objectives involves the development of testing tools based on the dataset assembled in this study. As we move forward, these tools will be meticulously designed to aid developers in accurately identifying and resolving common issues they may encounter during the development process.

Our proposed approach is largely inspired by the research conducted by Nusrat et al [18]. Their study provides a framework for future exploration as they carefully analyzed the commit history of open-source Unity projects on GitHub. By comparing versions of code throughout their development history, they were able to pinpoint the most frequently addressed and updated bugs.

Building on this foundation, our future work will involve a similar analysis of the commit history of popular projects that are frequently updated. Through this analysis, we plan to develop a testing tool designed to pinpoint the bugs most likely to occur in similar projects. By doing so, we aim to provide developers with a tool that anticipates and identifies potential issues, thereby streamlining the debugging process and increasing overall development efficiency.

6.3 Proposal of areas where improvements can be made in virtual reality software

VR datasets can be utilized to enhance performance and user experience in virtual reality (VR) software across various aspects. The following are some of the benefits of using our VR datasets:

1. Improving the quality of graphics in virtual reality can be done through the use of high-quality VR datasets. These datasets can provide realistic references to enhance graphics rendering algorithms and techniques. By capturing and reconstructing real-world scenes, developers can create more accurate textures, lighting, and physics simulations, thereby enhancing the visual realism.

For instance, we have a VR project dataset that includes a real-life scene from an indoor museum. The dataset comprises high-resolution images captured from different angles. Developers can use these images

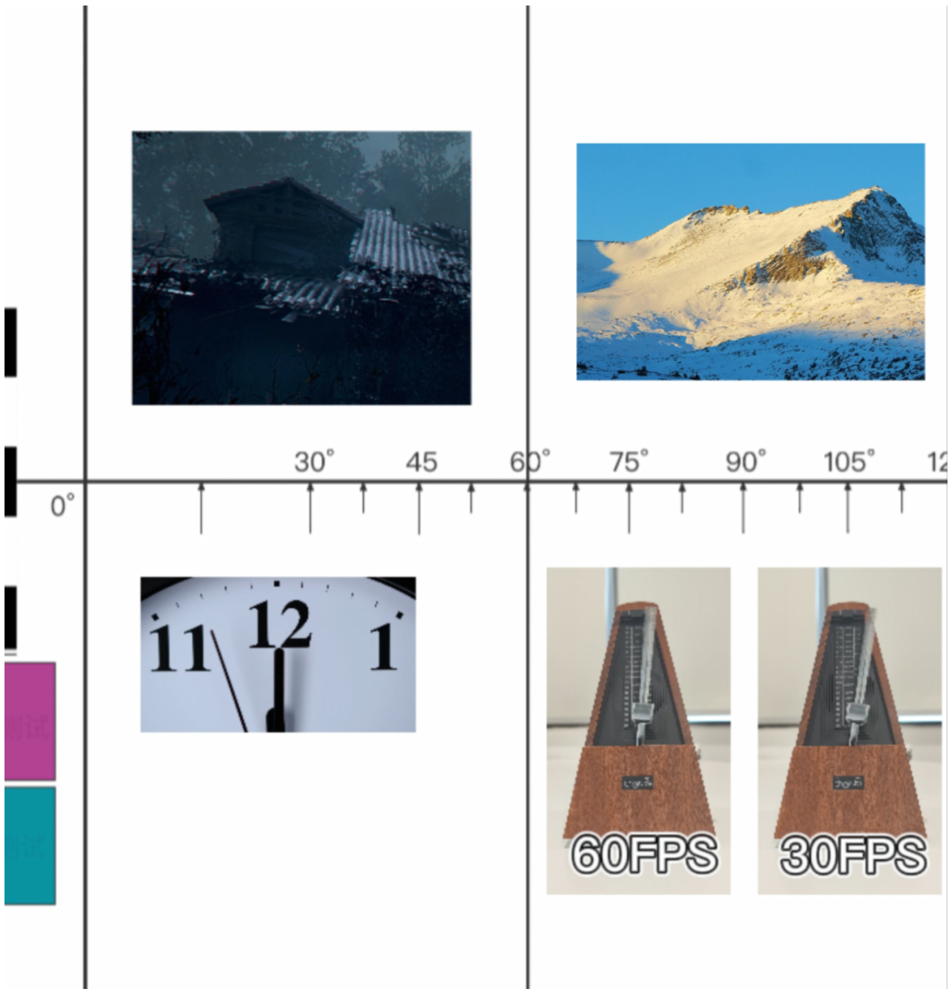


Figure 37: Improve quality of graphics

as a reference to recreate the surface texture of objects in the scene in a virtual environment. This can be done through texture mapping and other techniques, which ensure that the surfaces of objects in the virtual environment look as realistic as those in the real world.

2. Utilizing audio recordings from VR datasets helps in better understanding the sound propagation and reflection characteristics in real environments. [19] This aids in improving spatial audio technology to achieve more accurate sound localization, audio feedback, and environmental sound effects, enhancing immersion.

Our dataset also includes various tools related to sound, which can assist developers in addressing challenges more effectively.

3. Our dataset contains a specialized tracking category that offers developers with useful code snippets and creative ideas to enhance interaction design in virtual reality. By utilizing this data, developers can improve their comprehension of user interaction behavior and create more precise and natural hand tracking and gesture recognition algorithms. These advancements contribute to the development of more immersive and intuitive virtual reality experiences.

4. Our VR datasets are valuable references and sources of inspiration

Repo Full Name	Repo Link	Type (Application, Library, Framewor	Scale (Lar	Deployed	Archived	Remarks	Created a	Pushed at	Updated	Stars	Fork
oculus-samples/Unity-Movement	https://github.com/oculus-samples/U	Tracking					2022-11-0	2023-06-2	2023-06-2	160	28
MerlinVR/PlayspaceTracking	https://github.com/MerlinVR/Playspa	Position Tracking					2021-07-1	2021-07-1	2023-05-1	16	0
ChristophGeske/ARCoreInsideOutTrackin	https://github.com/ChristophGeske/A	Position Tracking					2018-01-2	2022-01-0	2023-06-2	186	27
ju1ce/April-Tag-VR-FullBody-Tracker	https://github.com/ju1ce/April-Tag-V	Tracking					2020-10-2	2023-03-0	2023-06-1	818	51
SummerSigh/ProjectBabble	https://github.com/SummerSigh/Proj	Mouth Tracking					2022-07-0	2023-06-2	2023-06-2	59	6
CatCuddler/BodyTracking	https://github.com/CatCuddler/Body	Motion Tracking					2017-03-2	2021-05-1	2023-01-0	18	16
korinVR/VRGestureRecognizer	https://github.com/korinVR/VRGestu	Motion Tracking					2013-12-1	2021-09-0	2022-10-0	29	7
getnamo/VRArmIK-ue4	https://github.com/getnamo/VRArmI	Hand Tracking	Demo				2019-02-0	2020-04-0	2023-01-0	11	2
jorgejgnz/HandTrackingPack-HapticFeedb	https://github.com/jorgejgnz/HandTr	Hand Tracking					2020-03-0	2020-03-2	2023-03-2	59	13
TebogoNakampe/XRDriveSim	https://github.com/TebogoNakampe/	Hand Tracking					2018-12-0	2020-07-2	2021-01-1	11	6
ju1ce/Mediapipe-VR-Fullbody-Tracking	https://github.com/ju1ce/Mediapipe-	Fullbody Tracking					2021-06-2	2023-05-1	2023-06-2	304	31
maceq687/FullBodyPoseEstimation	https://github.com/maceq687/FullBo	Fullbody Tracking					2022-03-2	2022-12-0	2023-06-2	57	6
MochiDoesVR/Caramel-For-VRChat	https://github.com/MochiDoesVR/Ca	Fullbody Tracking					2022-12-0	2022-12-1	2023-06-1	10	1
Adjerry91/VRCFaceTracking-Templates	https://github.com/Adjerry91/VRCFa	Face Tracking					2021-12-0	2023-06-0	2023-06-2	50	0

Figure 38: Tracking Data

for content creators and developers. Our datasets contain a wealth of information about real-world scenes and elements captured in virtual reality, which can provide insights into the composition, aesthetics, and immersive qualities of various environments.

For content creators, VR datasets are a rich source of reference material that can help them design and craft virtual reality experiences. By studying the scenes, objects, textures, lighting, and other elements captured in the datasets, creators can incorporate realistic and diverse elements into their own virtual reality content. They can draw inspiration from the details and nuances observed in the datasets to enhance the visual quality and overall realism of their virtual environments.

Moreover, VR datasets can inspire innovative and imaginative content creation. By exploring the vast range of scenes and environments within the datasets, developers and creators can identify new ideas, concepts, and storytelling possibilities. They can use the dataset's content as a foundation and then introduce their own creative twists and additions to craft unique and captivating virtual reality experiences.

In summary, Our VR datasets are a valuable resource for content creators, providing them with references, inspiration, and a starting point for designing and developing realistic, diverse, and immersive virtual reality content. By leveraging the information and insights within the datasets, creators can push the boundaries of virtual reality and deliver captivating experiences to users.

5. Our dataset is designed to improve cross-platform adaptation and compatibility. It includes data from various platforms and devices, which helps developers to create software that works seamlessly across different hardware and operating systems.

Our dataset contains information from different platforms, giving developers insights into the characteristics and requirements of various devices. By analyzing this data, they can make necessary adjustments to ensure that their virtual reality software is compatible with different hardware specifications, processing capabilities, input/output methods, and software interface requirements.

With the cross-platform data in our dataset, developers can better understand the differences between various platforms and create software that performs optimally on each device. This results in better performance and user experience, making virtual reality software more accessible and usable.

In summary, our dataset provides valuable references for cross-platform adaptation and compatibility. This helps developers cater to the needs of users across different platforms and devices, resulting in a broader and more extensive virtual reality experience.

6.4 Consideration of the integration of AI technologies (LLM model) in enhancing VR experiences

We are planning to integrate LLM (Large Language Model) technology into our VR dataset, to provide users with a more immersive and personalized experience. Our goal is to create a conversational interface with LLM, which will allow users to make specific queries or requests, and obtain the data they need for their VR experience.

By engaging in dialogue with the LLM model, users can interact with the model directly, and receive relevant information based on their queries or requirements. The model can also provide explanations and analyses of the dataset, thus enhancing the user's experience and increasing the efficiency of developers in the VR environment.

Moreover, the LLM model can offer guidance and suggestions on utilizing the VR dataset. Users can consult the model for the best ways to leverage the dataset, interpret specific data patterns or trends, and address any challenges or issues. By providing practical advice based on its training and knowledge, the LLM model can help users better understand and utilize the VR dataset.

To achieve our goals, we plan to use the OpenAI fine-tuning API to fine-tune the ChatGPT3.5-turbo-1106 model [20]. Fine-tuning OpenAI text generation models requires a careful investment of time and effort, and we plan to utilize techniques such as prompt engineering, prompt chaining (breaking complex tasks into multiple prompts), and function calling to commence the training process. We are currently training the model, and it will be our main focus for the next semester. For detailed information about fine-tuning OpenAI models, you can refer to this link: https://platform.openai.com/docs/guides/fine-tuning.

7 Conclusion

7.1 Recapitulation of the study's objectives and key findings

We conducted a comprehensive analysis of VR developers' concerns and existing VR project patterns, targeting insights for future tool development. Our main data source was GitHub due to its abundance of open-source projects. A custom crawler was used to extract VR-related repositories based on their names, descriptions, or README files. This raw data underwent several filtering and classification rounds to sort repositories from a software engineering perspective, facilitating developers' resource search.

In addition to data collection, we analyzed repositories based on their content. We also reviewed VR-related papers from Google Scholar, uncovering a focus on integrating VR with other fields and enhancing VR technology itself.

To provide easy access to our dataset, we developed a web-based UI for developers, including a search page for quick project category location. Our work offers a valuable resource for VR developers and newcomers to the field.

7.2 Summary of the contributions and implications of the research

We have curated a dataset comprising VR-related repositories, accompanied by a web-based user interface, devised for easy access to the data. Apart from this achievement, our study acknowledges the existence of issues that still require resolution.

Our classification process, despite multiple iterations, has resulted in certain categories that fail to provide clear and precise description of their contents. This issue primarily stems from the intricate nature and high flexibility of VR technology projects. Moving forward, we recognize the necessity for a more nuanced refinement of our current taxonomy to address these ambiguities and better categorize VR projects.

Furthermore, our web-based user interface, currently in its nascent stage, still lacking in terms of performance, particularly during data retrieval operations. This not only hampers usability but also detracts from user-friendliness. As such, future efforts will be directed towards enhancing the interface's performance, with a particular focus on improving search efficiency and overall user experience.

In conclusion, while we have made significant strides in organizing and providing access to VR-related resources, there is a clear path ahead for further improvement and refinement of our offerings.

References

[1] M. O. Onyesolu and F. U. Eze, "Understanding virtual reality technology: advances and applications," *Adv. Comput. Sci. Eng*, pp. 53–70, 2011.

[2] S. Mystakidis, "Metaverse," *Encyclopedia*, vol. 2, no. 1, pp. 486–497, 2022.

[3] S. Weech, S. Kenny, and M. Barnett-Cowan, "Presence and cybersickness in virtual reality are negatively related: a review," *Frontiers in psychology*, vol. 10, p. 158, 2019.

[4] Meta, "Meta reports second quarter 2023 results," Meta, Second Quarter Results, 2023.

[5] S. Kraus, D. K. Kanbach, P. M. Krysta *et al.*, "Facebook and the creation of the metaverse: radical business model innovation or incremental transformation?" *International Journal of Entrepreneurial Behavior & Research*, vol. 28, no. 9, pp. 52–77, 2022.

[6] R. R. Jitesh Ubrani, Ramon Llamas, "Ar/vr another slow year expected for ar/vr headsets before 2024 rebound, according to idc," 10 2023.

[7] R. Epp, D. Lin, and C.-P. Bezemer, "An empirical study of trends of popular virtual reality games and their complaints," *IEEE Transactions on Games*, vol. PP, 01 2021.

[8] A. Heath, "Meta's flagship metaverse app is barely used by the employees building it," 2022, [Online]. Available: https://www.theverge.com/2022/10/6/23391895/meta-facebook-horizon-worlds-vr-social-network-too-buggy-leaked-memo.

[9] C. Donalek, S. G. Djorgovski, A. Cioc *et al.*, "Immersive and collaborative data visualization using virtual reality platforms," in *2014 IEEE International Conference on Big Data (Big Data)*. IEEE, 2014, pp. 609–614.

[10] J. Zhao, J. O. Wallgrün, P. C. LaFemina *et al.*, "Harnessing the power of immersive virtual reality-visualization and analysis of 3d earth science data sets," *Geo-spatial Information Science*, vol. 22, no. 4, pp. 237–250, 2019.

[11] M. Akdere, Y. Jiang, and F. D. Lobo, “Evaluation and assessment of virtual reality-based simulated training: exploring the human–technology frontier,” *European Journal of Training and Development*, vol. 46, no. 5/6, pp. 434–449, 2022.

[12] Q. Zhang, K. Wang, and S. Zhou, “Application and practice of vr virtual education platform in improving the quality and ability of college students,” *IEEE Access*, vol. 8, pp. 162 830–162 837, 2020.

[13] F.-X. Geiger and I. Malavolta, “Datasets of android applications: a literature review,” 2018.

[14] M. Zubair and N. Anyameluhor, “How long do you want to maintain this thing? understanding the challenges faced by webxr creators,” in *Proceedings of the 26th International Conference on 3D Web Technology*. New York, NY, United States: ACM, 2021, pp. 1–4. [Online]. Available: https://dl.acm.org/doi/10.1145/3485444.3495181

[15] P. McGee, “Apple’s vision pro headset attracts investors back to lacklustre sector,” 2023. [Online]. Available: https://www.ft.com/content/83f53d0b-43a6-4005-a89a-633f5770b126

[16] D. Trabucchi and T. Buganza, "Data-driven innovation: switching the perspective on big data," *European Journal of Innovation Management*, vol. 22, no. 1, pp. 23–40, 2019.

[17] B. Xie, H. Liu, R. Alghofaili, and et al., "A review on virtual reality skill training applications," *Frontiers in Virtual Reality*, vol. 2, p. 645153, 2021.

[18] F. Nusrat, F. Hassan, H. Zhong, and X. Wang, "How developers optimize virtual reality applications: A study of optimization commits in open source unity projects," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 473–485.

[19] C. Wakeland, "The importance of audio in vr," *AMD GPUOpen*, 2016. [Online]. Available: https://gpuopen.com/learn/importance-audio-vr/

[20] OpenAI, "Fine-tuning language models," https://platform.openai.com/docs/guides/fine-tuning, 2021.