ShapeHub: An Online Collaboration Platform Implementing Version Control for Collaborative Design

by

Qiao Zhang

Submitted to the College of Fine Arts, School of Architecture
on May 24, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computational Design

Advisory Committee:

Professor Eddy Man Kim
Principal Advisor

Professor Daniel Cardoso Llach
Advisor

Abstract

The pandemic has forced a major transition in education: from face-to-face to online education, which poses challenges for students in collaborative design. Recently, the process usually involves communicating through video conferences and chats, copying design files, and continuing to make modifications, which lacks effective communication and may lead to irreversible changes.

In the research, a web-based collaborative design platform is developed, where teammates can share ideas and iterate on others' designs. The platform adopts an asynchronous communication model, but it also supports synchronous design features, such as real-time online chat. It integrates the branch mechanism and history function to implement version control, thereby improving the efficiency and experience of collaborative design. The platform is neither an online version of a traditional architecture studio nor a 3D version of GitHub. It inherits the former's collaborative culture, references the latter's workflow and functions, and develops on this basis to meet the needs of architectural collaborative design and adapt to the characteristics and limitations of digitalization. By rethinking the design process, platform and culture, the research is expected to contribute to the creation of an online collaborative design community.

Keywords

Version Control, Collaborative Design, CSCW, Online Platform, Design Education

Acknowledgments

I'd like to take this chance to express my sincere gratitude to everyone who supported, helped and advised me throughout my graduate study.

First, I'd like to thank my committee members, professor Daniel Cardoso Llach and professor Eddy Man Kim. Thank you for making me a member of the computational design program, giving me valuable guidance and help during my entire graduate study, and enabling me to have more understanding and critical thinking about the entire field of design, including computational design.

I'd also like to acknowledge Yi Cui and Li Cai, my teammates from 17-637 Web Application Development course, because the platform includes some of the features of our collaborative project in 17-637 course.

Additionally, I'd like to thank my family and friends for always supporting and encouraging me.

I'd like to offer my final thanks to the computational design community, which inspired me a lot, and hope this thesis can contribute to it in return.

Contents

1 Introduction **11**

1.1 Research Question 11

1.2 What is at Stake? 11

1.2.1 The Need for an Online Platform 11

1.2.2 The Attention to Students 12

1.2.3 The Attention to Collaborative Design 14

1.2.4 The Attention to Version Control 15

2 Background **17**

2.1 Existing Platforms 17

2.1.1 Collaborative Design Platforms for Architects 17

2.1.2 Platforms Used by Colleges During the Pandemic 18

2.2 Limitations of Existing Platforms 20

2.3 Bibliographical Essay 20

2.3.1 Models for Web-based Education 20

2.3.2 Models for Communication 21

2.3.3 Git and GitHub 22

2.3.4 Versioning as Series 22

2.3.5 Versioning and Collaboration Tools 23

2.4 Limitations of Previous Research 25

3 Hypothesis and Vision **27**

3.1 Hypothesis 27

3.2 Vision 27

4 Methods **29**

4.1 Platform Architecture 29

4.2 Platform Workflow 30

4.3 Core Functions 35

4.3.1 Project Management 35

4.3.2 Design State Management 37

4.3.3 3D Model Management 39

4.4 Auxiliary Functions 43

4.4.1 Register/Login/Logout 43

4.4.2 Chat Synchronously 43

4.4.3 View Public Project Gallery 44

4.4.4 User Profile 44

4.4.5 Follow/Unfollow Other Users 45

4.5 Discarded Functions 45

4.6 Evaluation Indicators 46

5 Use Case Testing and User Evaluation **49**

5.1 User Individual Exploration 49

5.2 Use Case Testing 50

5.2.1 Case 1: Collaboration Between Architecture Students Through the Platform and Rhino 50

5.2.2 Case 2: Collaboration Between Architecture Students Through the Platform and SketchUp 52

5.2.3 Case 3: Collaboration Between Professional and Amateur Students 54

5.3 User Evaluation 56

5.3.1 Feedback from Professionals 56

5.3.2 Feedback from Amateurs 58

5.3.3 Rating Results 59

6 Contributions and Future Work **61**

6.1 Research Contributions 61

6.2 Limitations and Future Works 63

List of Figures

1-1 Communication Channels Present During the Multimodal Critique Session[1] . . . 12
1-2 "Mental Health and Substance Use Fall 2019 vs Spring 2020 (Healthy Minds Study)"[2] . . . 13
1-3 "Mental Health and Substance Use Fall 2019 vs Spring 2020 (ACHA-NCHA)"[2] . . . 13

2-1 The Platforms Used by Universities During the Pandemic[1] . . . 19
2-2 The Collaborative Group Model[3] . . . 21
2-3 Embryological House by Greg Lynn, 1999[4] . . . 23
2-4 Tree Diagram View[5] . . . 24
2-5 GHShot[6] . . . 25

4-1 Platform Architecture . . . 29
4-2 GitHub Workflow . . . 31
4-3 Platform Workflow that Combines Discrete Design States and Continuous Construction History Between Two Design States . . . 32
4-4 Platform Workflow in Practice . . . 34
4-5 Core Functions . . . 35
4-6 Create a New Project . . . 36
4-7 Edit a Project . . . 36
4-8 Edit a Design State . . . 38
4-9 Push a Design State . . . 39
4-10 A Project After Pushing Some Design States. . . . 39

4-11 Export an Object 40
4-12 Import an Object 40
4-13 Import/Export Files from/to SketchUp 40
4-14 Import/Export Files from/to Rhino 40
4-15 Add an Object 41
4-16 Transform 41
4-17 Target Morphing 41
4-18 Record History 42
4-19 Return to any Previous State 42
4-20 Start from a Previous State 42
4-21 Register 43
4-22 Login 43
4-23 Chat Synchronously 43
4-24 View Public Project Gallery 44
4-25 User Profile 44
4-26 Follow/Unfollow Other Users 45
4-27 Results Obtained by the Diff Finder 46

5-1 Design States of Case 1 51
5-2 Design State Structure of Case 1 52
5-3 Design States of Case 2 53
5-4 Design State Structure of Case 2 54
5-5 Design States of Case 3 55
5-6 Design State Structure of Case 3 56
5-7 Chat Room 56

List of Tables

2.1 Features of Autodesk A360 Viewer and Modelo 18

4.1 Project Model 35
4.2 New Project Data 36
4.3 Design State Model 37
4.4 First Design State Data 37
4.5 New Design State Data 38
4.6 Indicators for Evaluating the Platform 47
4.7 Indicators for Evaluating Collaboration 48

5.1 Rating Results for the Platform 60
5.2 Rating Results for Collaboration 60

Chapter 1

Introduction

1.1 Research Question

How can an online collaboration platform that implements version control assist students in collaborative design?

1.2 What is at Stake?

1.2.1 The Need for an Online Platform

When it comes to using the Internet, architectural education never seems to keep up with the cutting-edge technology of the times. Almost every aspect of traditional architectural studio culture depends heavily on the face-to-face contact between professors and students, through hand-drawn sketches, slides, videos, or direct presentation of 3D models in modeling software. Colin Gray and Craig Howard‘s[1] summary of the communication channels present during the multimodal critique session (figure 1-1) demonstrates the universality and importance of direct verbal communication in studio culture. Online platforms are rarely involved in the design process, but they are sometimes utilized to display design results, such as diagrams and videos of the final design. These online platforms did not play an essential role because they "serve only to publish data, offer one-way communication and contain ‘read-only’ content"[7].

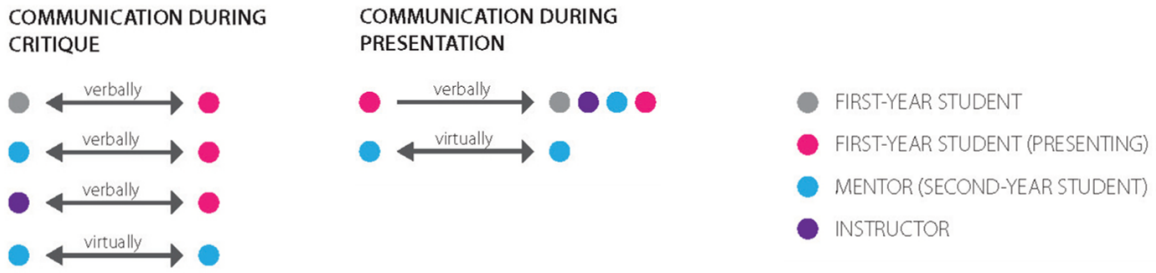


Figure 1-1: Communication Channels Present During the Multimodal Critique Session[1]

COVID-19 has forced a major transition in education: from face-to-face to online education. Educational institutions around the world have responded quickly and creatively to this, including using web conferencing platforms such as Zoom and visual collaboration platforms such as Miro, on which professors and students of architecture conduct two-way communication and dynamic involvement that usually occur in studios. However, these platforms are not satisfactory because online education should not be the online version of traditional education and it requires a shift from the old teaching principles and practices to new ones that adapt to the characteristics and limitations of digitalization. Cook[8] elaborated the shift as follows:

"Building theories of good online education involves asking ourselves which of the principles and practices that have guided instruction in traditional classrooms can be transferred or adapted to work in the new electronic class spaces that have become available to us, which ones must be discarded or replaced, and what new practices must be developed for these new spaces."

The thesis envisions an online platform that adapts and develops the existing studio culture through addition and subtraction.

1.2.2 The Attention to Students

During the pandemic, online platforms are needed for architects and architecture students to conduct two-way communication with collaborators, clients, or professors. I argue more attention should be paid to the architecture students for the following two reasons:

First of all, the functions of existing online collaborative design platforms cannot cater to the needs of architecture students. Platforms like Autodesk A360 Viewer and Modelo allow users to view 3D models and information, share designs, and make comments, which is suitable for presenting completed works. However, for students majoring in architecture, in addition to displaying the design results, the platform is also expected to show different attempts and continuous experiments in the design process.

Besides, the current distributed design education model cannot cater to the psychological needs of students. According to a report by the American College Health Association(figure 1-2, figure 1-3)[2], a higher proportion of college students report symptoms of mental health, including overwhelming anxiety, depression, and loneliness during the pandemic.

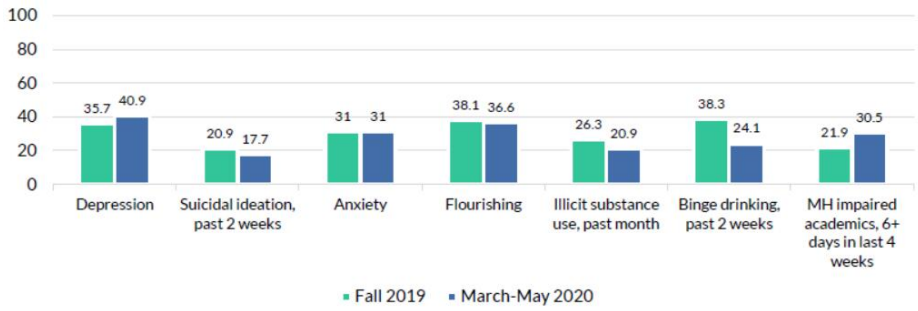


Figure 1-2: "Mental Health and Substance Use Fall 2019 vs Spring 2020 (Healthy Minds Study)"[2]

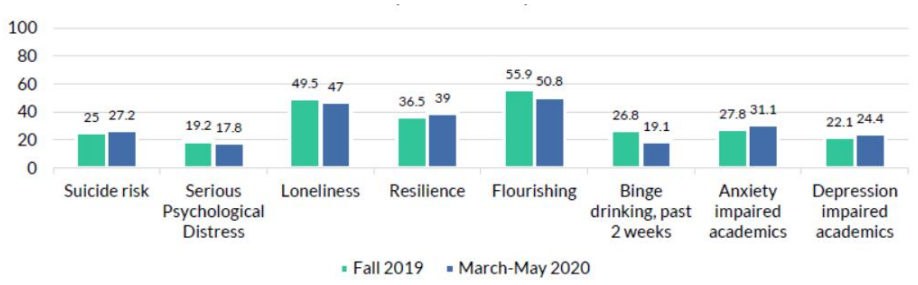


Figure 1-3: "Mental Health and Substance Use Fall 2019 vs Spring 2020 (ACHA-NCHA)"[2]

Benjamin George[9] summarized "the constraints and barriers to the adoption of distributed design education" and proposed mitigation methods for "students feel

social isolation":

- "Select communication technologies to encourage rapport building."
- "Scaffold social interactions into course curriculum."
- "Improvements in communication and collaboration technologies."

Technologies should be improved to assist students to collaborate more easily, reduce the feelings of loneliness and isolation, and increase their sense of community.

1.2.3 The Attention to Collaborative Design

Collaborative design is defined as "an activity that requires participation of individuals for sharing information and organizing design tasks and resources" by Chiu[10]. There are two important aspects in the definition, namely "sharing information" and "organizing design tasks and resources".

In architectural industry, the characteristics of architectural design determine that architectural design requires close collaboration between architects, interior designers, landscape designers, engineers, contractors, and specialist suppliers. In traditional design studios, the collaboration is usually reduced to small group collaboration where Students from the same or different disciplines share their knowledge and ideas, make design decisions, and design new schemes. "Sharing information" is mainly achieved through face-to-face communication between professors and teammates, followed by online chat. "Organizing design tasks and resources" are usually performed by team members modifying different versions of files. However, architecture studios sometimes deliberately overlook the collaborative nature of architectural design and emphasize independent student projects. Admittedly, this helps to train students' personal skills and makes it easier to evaluate them. However, what architectural design needs is not a singular genius, but a member of a diverse team. Online education is even more against this collaborative nature. In online courses, students design their projects at home by themselves, but this situation does not exist in the industry.

At present, the transition to online education in architecture forced by the pandemic is an opportunity to develop the aspects of today's studio culture that disconnect the industry. And this thesis will focus on the development of collaborative design culture. One possible improvement is to change the main way students deliver design results from independent projects to group projects, which may help alleviate a lack of "direct exposure to the thoughts, values and working processes of other people"[1] on online platforms. Another possible improvement is the development of online collaborative design tools. This thesis aims to develop a platform that enables architectural education to embrace online collaborative design and reform the social culture of design schools.

1.2.4 The Attention to Version Control

"In software engineering, version control is a class of systems responsible for managing changes to computer programs, documents, large web sites, or other collections of information. Version control is a component of software configuration management."[11]

Design can be described as a series of divergent and convergent explorations towards the final result. When designers conduct collaborative design, they sometimes face the following bottlenecks:

- Design teams make modifications on different versions of the files.
- Files are exported from one tool / device and imported into another, duplicated, and renamed.
- Files are overwritten, damaged or lost.
- Feedback is separated from the design files.
- Teammates and professors don't know what's changed from one version to the next, and why.
- Inefficient communication.
- After the project ends, its generation history will not be retained.

Students should be provided with a better design workflow platform where each team member can change any file anytime, anywhere, without worrying about making irreversible changes. And team members and the professors can track changes and identify when key decisions were made along the way.

Chapter 2

Background

The chapter first studies the existing online collaborative design platforms for architects and platforms actually used by universities during the pandemic and finds the gap between them and the expected platform. Platforms for architects do not adapt to the educational context while platforms used by the universities do not adapt to architecture majors. Then, through literature review, an appropriate model for web-based education and communication is determined. Based on the analysis of the contributions and limitations of researchers' version control theory and tools, the focus of the thesis will be put forward.

2.1 Existing Platforms

2.1.1 Collaborative Design Platforms for Architects

There is no shortage of excellent products to assist architects in online collaborative design on the market, such as Autodesk A360 Viewer and Modelo, where team members can view 3D models and information, share designs, and make comments in a centralized workspace. Table 2.1 shows the features of Autodesk A360 Viewer and Modelo.

Features	Autodesk A360 Viewer	Modelo
View 2D and 3D files without plug-ins	Yes	Yes
Share files via a unique URL	Yes	Yes
Comment in real time	Yes	Yes
Mobile access	Yes	Yes
Security	Yes	Yes
Cloud storage	Yes	Yes
Find information quickly	Yes	No
Design metadata: see the relationships between files, components or external references	Yes	No
Create 3D animated walkthroughs	No	Yes
Mark up PDF files in the cloud	No	Yes
360-degree panoramic tour	No	Yes
VR	No	Yes
Offline model	No	Yes
Assets folder	No	Yes

Table 2.1: Features of Autodesk A360 Viewer and Modelo

2.1.2 Platforms Used by Colleges During the Pandemic

According to a form on the ACSA Webinar (figure 2-1), during the pandemic, most universities are using platforms they are already familiar with, such as Zoom, Miro, Slack, and Canvas to communicate. It can be shown that some universities are beginning to experiment with new platforms, such as the MOZILLA hub, on which students can "hang" and display their design drawings in a virtual 3D room. Nevertheless, the collaborative design platforms mentioned in section 2.1.1 are never utilized by any university.

PLATFORMS: What software do you recommend, and why?	MEDIA: How have your students presented drawings, narratives, films, models, and other media?	SOCIABILITY + REVIEW CULTURE: How have you arranged critic to student relationships? How have you timed these interactions?
Zoom for video chat, screen share, on-screen drawing - Using Breakout sessions (within Zoom) to share Polls, and drop into the sessions.	Format is key, conventional "boards" won't do anymore. It is good for recording the meeting, the steps that would have been taken in time.	We created smaller break out rooms, where there were hosts in each room. A student was assigned to keeping time and reporting back the tonality of the room. All reviewers were given in advance a expectation of conduct in review statement. This allowed reviewers to be very conscientious of the review culture in ZOOM.
Concept board for screen share and on-screen drawing retains image quality as you zoom in	We can all take clues from Columbia's paperless studio!	Drop-in format: Can anyone from any studio, or any school drop in uninvited?
Microsoft Teams Like Slack, but in Microsoft Office 365 environment	MsTeams = perhaps awkward way of using, but muddling through using drawing tool in PowerPoint, having students export pdf to PPT	I have found the announcements, email list serv, shared folders useful. I have yet to try the sychronous meeting capability.
Miro Like Mural, it is a collaborative platform - potentially good for presentations	Regardless of media, sharing images/files etc. with reviewers/facutly in advance of any live interaction so they can view in advance is helpful	
Mural Mural responded they can tackled unlimited users, but recommend a cap of 25 simultaneously, but others have tackled sessions in the 100s		
Instagram Gallery presentations & shared hashtags	The drawings, collages, questions, and parts of their essays for an Architectural Theory course were presented in plate/slide form. The students were given citations, they were asked in adance if they wanted to share their work in this format.	good idea, but privacy issues. Perhaps students can create second accounts if their main accouont is private?
Voicethread asynchronous presentations and students/prof feedback	Emphasizing video representation of hand-made physical models as final product - views, lighting, sequence of panning.	
Webex good for mark-up options, but expensive		
Blackboard Also called Desire2Learn and used at Universities in Georgia.	PDF upload and file share, PPT file share, or screen share to view active modeling studies or research.	Providing a link to external critics is simple and online review is facilitated in the Collaborate Ultra area intuitively.
PDF for more linear format and narrative, using slides to zoom in/out		If privacy is not a concern, can try livestreaming your private crit sessions to Twitch, YouTube Live, or Facebook Live, they allow commenting and feedback as well.
Slack Way to tie all other platforms together, synced directly to Slack.		Very useful in creating channels of chats, the students can create their own topical channels to share info quickly, if you allow the notification feature to pop up in your computer. When you are working, questions and concerns will come up in your lower righthand corner.
Google Earth site analysis. I created a project and highlighted the site for students to see the extent of their research.	There're differences & similarities in film vs. architecture narrative. Neuroscientific aspects in visual communication are covered by Roland Bleicker and others. Design expertise in corporeal physical connections is extremely underdeveloped.	Since I get only 6 hrs/wk w/ the class, I allow students to text me images of their progress models. I comment asychronously when I have time. Of course this takes more of my personal time.
Google Meet + Google Slides		
Canvas Streamlines digital tools and content for online learning	Webex installed in Canvas for meetings and office hours for one-on-one with students. Students can upload presentations or use whiteboard.	Have not tried yet, but Webex Event can be used to invite individuals and students to review sessions. FERPA issue unresolved. Use WebEx "Office Hours" for one-on-one meetings with students. Students select hour from posted schedule and appointments appear on Calendar.
Bluebeam PDF mark-up platform that also allows simultaneous sharing of screen and marking of pdf using architecture redline and note tools.	Look especialy into the Studio and Session collaborative tool. Very useful for tracking files and mark-ups. All PDF so only works with this file format.	Not great for reviews, but very effective for desk critiques. You can follow a student as they walk through a set and have a really effective mark-up tool set that is embedded in the pdf.
FreeConferenceCall.com - is a free platform (takes donations) that has an interface much like Zoom		
ESRI ArcGIS online, story maps could be great for urban analysis, many universities have accounts, free accounts also available		
Adobe Project Aero (Advanced License, iOS)		
Adobe myportfolio.com sites	Students had first created a series of PDFS related to site drawings, site sections, project diagrams, project floorplans, project axonometrics, and project perspectives.	The products of this format were then plae quickly in myportfolio.com websites in order that classmates/cohorts who were not attending the final sychronous review could see their colleagues work and could be shared with the rest of the faculty.
Adobe XD (Interface and UX/UI Design)		
ADOBE Spark	One can make slideshow/slidedeck with imbedded video for presentations overlap text, titles. This can incorporate a sense of narrative in the presentation.	It is a presentation style that overlaps slides therefore has this opportunity to feel like the presentation is somewhat animated or has moving qualities but are mostly developed from still slides.
ENSCAPE (3D Model sharing both in VR and in a simple web app)		
HUBs by MOZILLA (virtual gallery - can walk around a virtual space)		
WEBJETS (presentation tool)	group work presentations, this is a platform for group collaboration and see the work in progress.	This is a great way for students to prepare work together in group, the faculty member can look onto the work during the project.
HAIKU DECK (presentation tool)	presentation maker. There are many graphic short cuts to a fast presentation built within the program.	This program requires some one to drive the screenshare. It is a program that requires to be in queue.

Figure 2-1: The Platforms Used by Universities During the Pandemic[1]

[1] https://www.acsa-arch.org/2020/03/13/pivot-to-online-learning-discussion-sessions/

2.2 Limitations of Existing Platforms

There is a gap between the existing online platforms and the online collaborative design platform expected by architecture students.

On the one hand, commercial products do not adapt to the educational context. For example, these platforms are more concerned with displaying "results" rather than the "process" of generating design solutions. "Architectural designers are far behind the technology curve when it comes to using the Internet as such – common practice in the industry today is to simply display design work in the form of pictures and videos to potential clients."[12] If a platform only presents the designs better, it is no wonder that students will choose a platform they are already familiar with, such as zoom.

On the other hand, some platforms used by universities during the pandemic are suitable for synchronous or asynchronous two-way interactions between users, but they do not adapt to architecture majors. For instance, Miro allows users to share pictures or texts, but it fails to support the sharing of 3D models that are essential for architectural collaborative design.

2.3 Bibliographical Essay

2.3.1 Models for Web-based Education

"There are various models for web-based education implementations such as The Individualized Instruction Model, which is a text-based system; The Class Model, in which students have the chance to work together and with an instructor; The Integrated Class Model, which is enhanced with research activities and audio and/or visual conferences; The Group Model, which involves the collaboration of a small group; and The Collaborative Group Model (figure 2-2) , which includes a consortium of educators with multiple resources in addition to the activities in all previous approaches." In order to take full advantage of the Internet, "the Collaborative Group Model is the one that best suits the nature of design education"[3] and will be used

in the thesis.

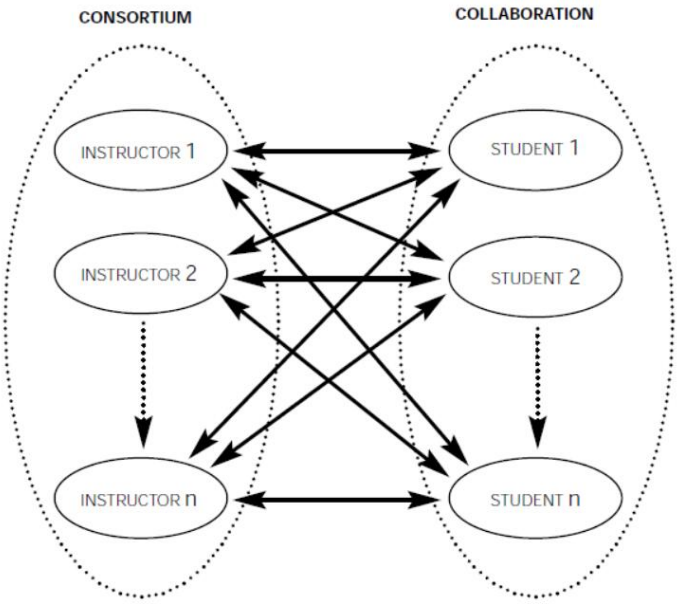


Figure 2-2: The Collaborative Group Model[3]

2.3.2 Models for Communication

"Web-based education can be conducted in two different communication models: synchronous and asynchronous."[3] Synchronous communication is a real-time interaction of geographically distributed participants. It is the most effective and direct system for team members to watch, discuss, and modify the 3D model at the same time. However, there is a time constraint in the system because it is up to the participants to decide the specific time of the meeting. On the other hand, "asynchronous communication is a non-real-time interaction of participants. In this system, there is no time constraint because participants do not have to be online at the same time." [3] Therefore, asynchronous tools can achieve permanent and continuous communication. Synchronous design requires the presence of all cooperating users[13], while asynchronous design requires thinking about how to maintain ongoing communication.[14] Aysu Sagun et al.[3] put forward that "Since synchronous communication provides real time interaction and asynchronous communication has no time restriction for interaction,

it is advisable to apply both systems in a web-based course." The thesis will mainly focus on designing an asynchronous collaborative design platform for version control, but there will be some synchronous design features, such as real-time online chat, so that teammates can communicate more effectively.

2.3.3 Git and GitHub

Linus Torvald wrote the Linux operating system and opened it to hundreds of thousands of developers.[15] ”While writing Linux in 2005, he realized the need for a distributed version control system and created Git.“[16] The GitHub service was developed by Chris Wanstrath, P. J. Hyett, Tom Preston-Werner, and Scott Chacon in February 2008, enabling more people to collaborate on the project. GitHub has always been used for "code and project management for software development" but is now being extended to "other domains that involve collaborative work", such as education.[17] But, as the famous computer science educator Greg Wilson[18] stated, "GitHub for Education" does not necessarily mean "Let's put educational materials in GitHub", but rather, "Let's facilitate a culture of spontaneous-but-structured collaboration and improvement." Similarly, "ShapeHub" can not only implement version control of 3D shapes but also inherit and develop the collaborative culture in traditional architectural design studios.

2.3.4 Versioning as Series

Even though the software engineering field is familiar with the term versioning, in the field of design, the meaning of versioning is controversial. In 2002, Ingeborg M. Rocker[19] termed versioning in the design field, arguing that ”thinking of design no longer as a single entity characterized by an essential form but rather as a series. Each design-event is hereby comprehended as a unique intricate version of a whole series of possible designs - all characterized through continuous similarities rather than clearly defined differences”. The versioning he termed exceeds a simple comparison of iterative architectural models and contains infinite design possibilities. Design

is no longer a process of refining or developing a single entity, but that of generating "one-of-a-kind customized variety"[20]. The Embryological House (figure 2-3) demonstrates "the power and the limitations of versioning-as-series" well[4], where Greg Lynn treated design proposals as a continuous and whole collection, rather than isolated and discrete design states. He combined "unique, intricate variations of each instance and the continuous similarity of its relatives"[4], just like interpolating on a spline. However, whether for feasibility or other reasons, this concept with unlimited possibilities "never left the stage of a sketchy proposal"[4].

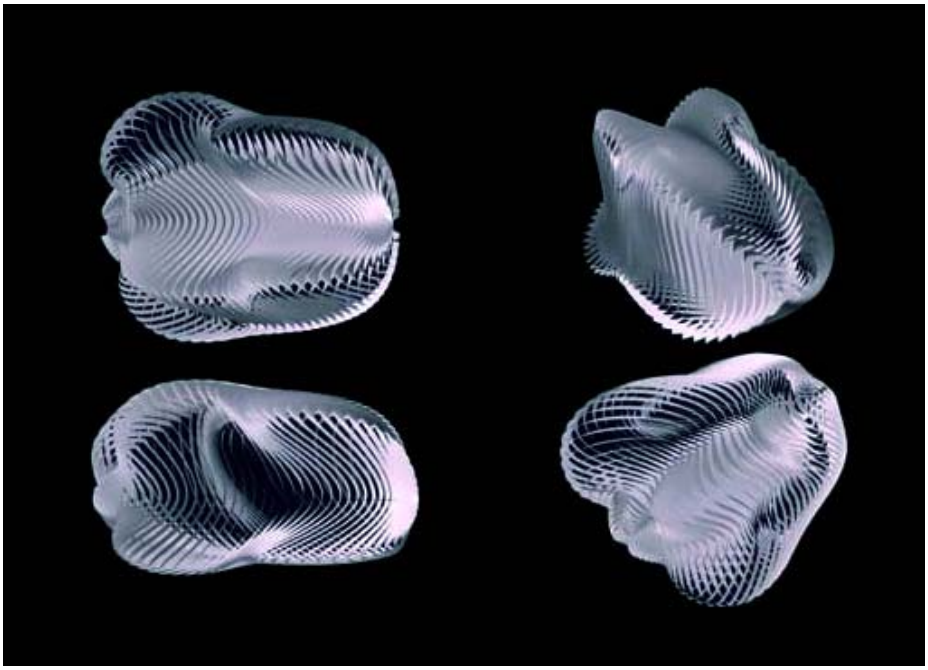


Figure 2-3: Embryological House by Greg Lynn, 1999[4]

2.3.5 Versioning and Collaboration Tools

In practice, versioning paradigm has been explored in the field of 2D and 3D design:

In the field of 2D design, Hartmann et al.[21] developed a tool to revise user interfaces through change tracking, annotations, and alternatives; Kolaric et al.[22] and Zaman et al.[23] respectively developed a graph-based tool to manage multiple design alternatives. The former allows users to create designs using built-in sketching tools, while the latter allows them to produce designs with a built-in generative

system.

In the field of 3D design, Burry et al.[24] developed a tool for file sharing of parametric models; Robert et al.[25] described a multiple-view structure; Sakai and Tsunoda[5] created a web application for people to create simple dwelling like volumes using webGL. It converts the model into a format that can run the Diff algorithm, and creates a tree diagram (figure 2-4) using the parent-branch example similar to the branching mechanism in GitHub.

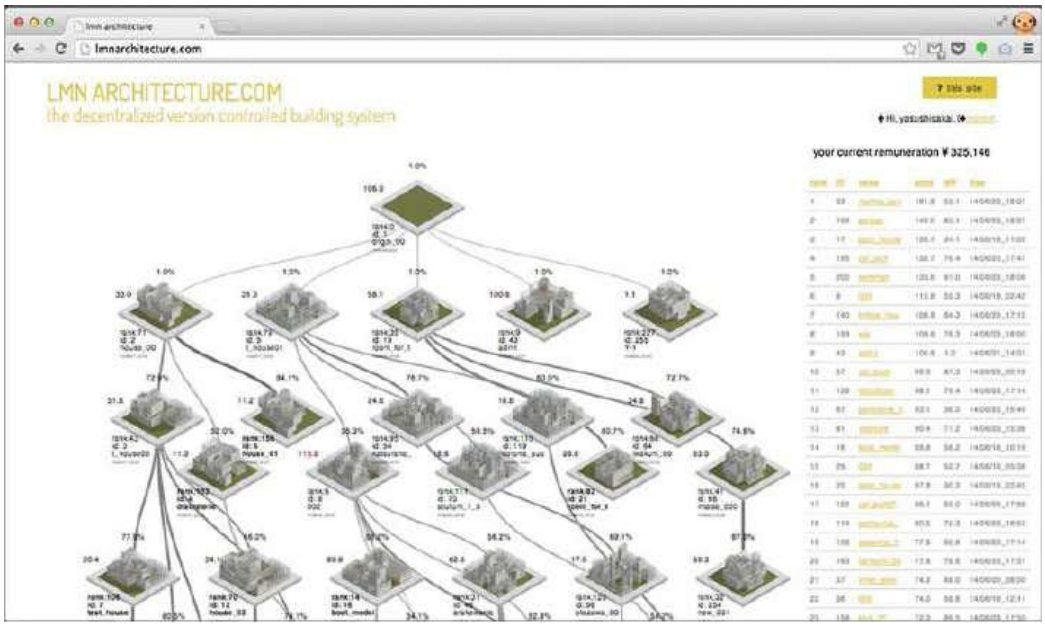


Figure 2-4: Tree Diagram View[5]

Verina and Sam[26][16][6] proposed a collaborative and distributed visual version control tool, GHShot (figure 2-5), for Grasshopper parametric programming. Users can create a new team project online and push/send the design in the Grasshopper to the website. It has certain features of GitHub, such as finding differences. Specifically, red, green, and yellow are used to highlight deleted components, added components, and changed components respectively. In addition, it introduces a branching mechanism similar to GitHub, enabling team members to simultaneously explore multiple possibilities of a design. Also, it will highlight the differences between the two versions of the file and allow the user to merge manually.

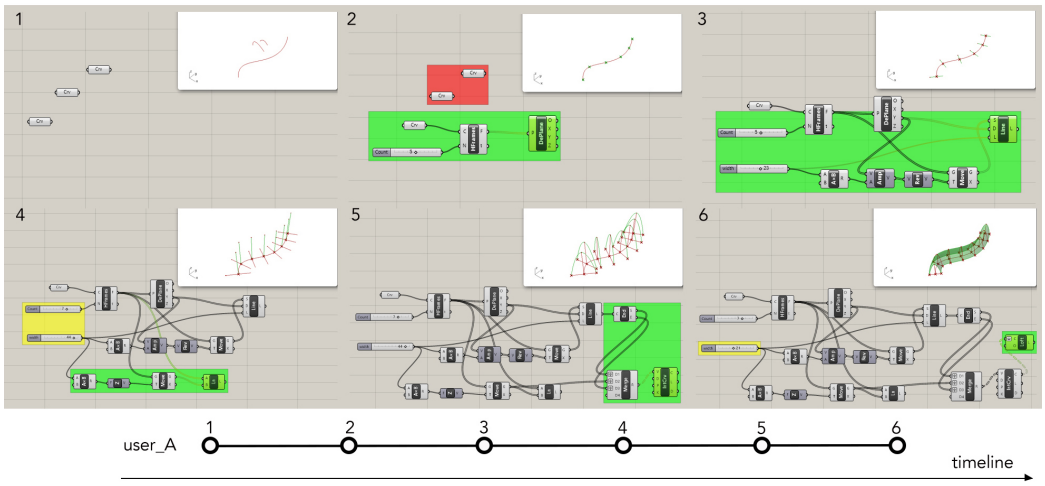


Figure 2-5: GHShot[6]

2.4 Limitations of Previous Research

The section mainly analyzes the controversies and limitations of previous research.

The main controversy is what kind of workflow should be used for version control of the architectural design. Some researchers, including Sakai and Tsunoda, propose a tree structure, while others, including Verina and Sam, argue that it should be similar to the Git workflow. Both of them have implemented the branching mechanism, but cannot reach a consensus on whether to allow two designs to be merged. The divergence stems from the difference in their understanding of versioning in architectural design: the former thinks of versioning as the process of generating a collection of designs, while the latter regards it as the process of developing a single entity. The thesis is more inclined to view versioning as series, so the platform abandoned the merging function.

Previous research has the following three limitations:

First, most platforms developed by previous researchers are not universal. For example, they only support one file format, or only have good compatibility with one 3D modeling software (eg, grasshopper). The platform proposed in the thesis will allow users to upload more file types, such as obj, DAE, etc.

In addition, a majority of platforms can perform version control on uploaded files, but do not support online modifications to models. Admittedly, the design should

be done mainly in the modeling software, but if the platform allows users to modify, discuss and save the phased design online, it will undoubtedly improve the efficiency and experience of online collaborative design.

Last but not least, many platforms provide good support for asynchronous communication between users. For example, users can attach design information when saving/pushing design states for others to view afterward. However, they rarely provide support for synchronous communication. The platform I envision will allow users to interact directly through real-time chat.

Chapter 3

Hypothesis and Vision

3.1 Hypothesis

Alexey et al.[27] proposed that GitHub has created efficiencies and helped improve the way software professionals work. Analogously, I propose the hypothesis that an online collaboration platform that implements version control can enhance the efficiency and experience of students in collaborative design. And the efficiency here does not mean that users can get collaborative design results faster, but can communicate more effectively.

To validate the hypothesis, the necessary web-based infrastructure for version control and collaborative design is first designed and implemented, and then professionals and amateurs are invited to use and evaluate the platform.

3.2 Vision

Facing the future and the "post-epidemic era", the vision of the thesis is to seize the opportunity to eliminate the longstanding drawbacks of traditional architectural design studios, adapt education to the characteristics and limitations of the Internet, and ultimately promote the reform of architectural education. By rethinking architectural education and industry, design process, platform, and culture, the thesis envisions an online collaborative design platform implementing version control for

collaborative design.

On the one hand, the platform is not an online version of a traditional architecture studio, because it pays more attention to the collaborative nature of architectural design that is sometimes overlooked in a traditional studio. Any online user, no matter where they are, whether they can communicate synchronously or not, can take active participation in design collaboration. On the other hand, the platform is not a 3D version of GitHub. Version control on this platform is not to plan a specific path for designers, but to provide them with the possibility of trial and error. Because "having space to showcase the efforts can be especially important in educational settings where problem-solving efforts are, in many ways, more important than the tangible output of a project."[28]

Last but not least, the platform is expected to create an online collaborative design community. On the platform, users can register as community members, where they can interact, learn, and create together. Beyond what the thesis can achieve, my vision is that the platform will one day become an open-source community like GitHub, where all community members can like, comment, share, download, and even modify public projects. It is foreseeable that this platform will bring disputes about ownership and privacy. What needs to be clarified is that just like GitHub, users can choose whether to make the project public. When it comes to private projects, no user other than the project collaborators will access (view or change) the private project, and the ownership of data is the same as that in traditional architecture collaborative design. When it comes to public projects, I envision two options, read-only and fully open source. The former is similar to private projects, and the latter is similar to GitHub's open-source projects. I agree that intellectual property as an asset needs to be protected, but I also believe that architecture is not a closed discipline by nature and should embrace the spirit of open source.

Chapter 4

Methods

In the chapter, the design and implementation methods of an online collaborative design platform for version control are introduced. The chapter documents the choices and trade-offs in the process of designing platform architecture and workflow, implementing core and auxiliary functions, and determining evaluation indicators.

4.1 Platform Architecture

Most of the collaborative design platforms developed by previous researchers follow a multi-layer system architecture, where different layers are responsible for different functions.

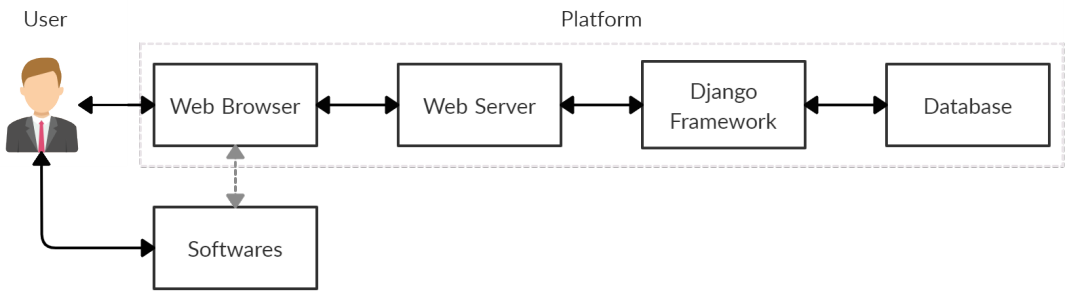


Figure 4-1: Platform Architecture

The platform developed in this research follows the system architecture shown in

figure 4-1, which is similar to that was proposed by Verina and Sam in GHShot.[6] The difference lies in the choice of server and database, as well as the compatibility with the modeling software. The main design decision that needs to be made here is whether to provide complete support for a specific software or to provide general support for more modeling software. Unlike Verina and Sam, who decided to focus on Grasshopper, this platform will support multiple software and file formats.

The following are the four layers that make up the platform structure:

- **Web Browser**

  In the process of collaborative design, users interact with the user interface of the browser. JavaScript is used to handle dynamic functions such as zooming in/out and uploading files. In addition, when processing functions that need to obtain or change data in the database (such as downloading files and pushing), the browser will send a request to the web server.

- **Web Server**

  The web server handles all requests from the browser. It sends back a response to the browser with the requested static contents and forwards the dynamic requests to the web framework. In the platform, the Apache HTTP Server is used.

- **Web Framework and Database**

  The web framework handles dynamic requests. It parses the user's data, stores it in the database, and then returns the data in JSON format to the user, which is then used by the web browser to render the 3D model. In the platform, the Django framework is used.

4.2 Platform Workflow

The design platform refers to the GitHub workflow (figure 4-2) and develops on its basis. Decisions should be made to determine which features to keep, which features to remove, and which new features to add.

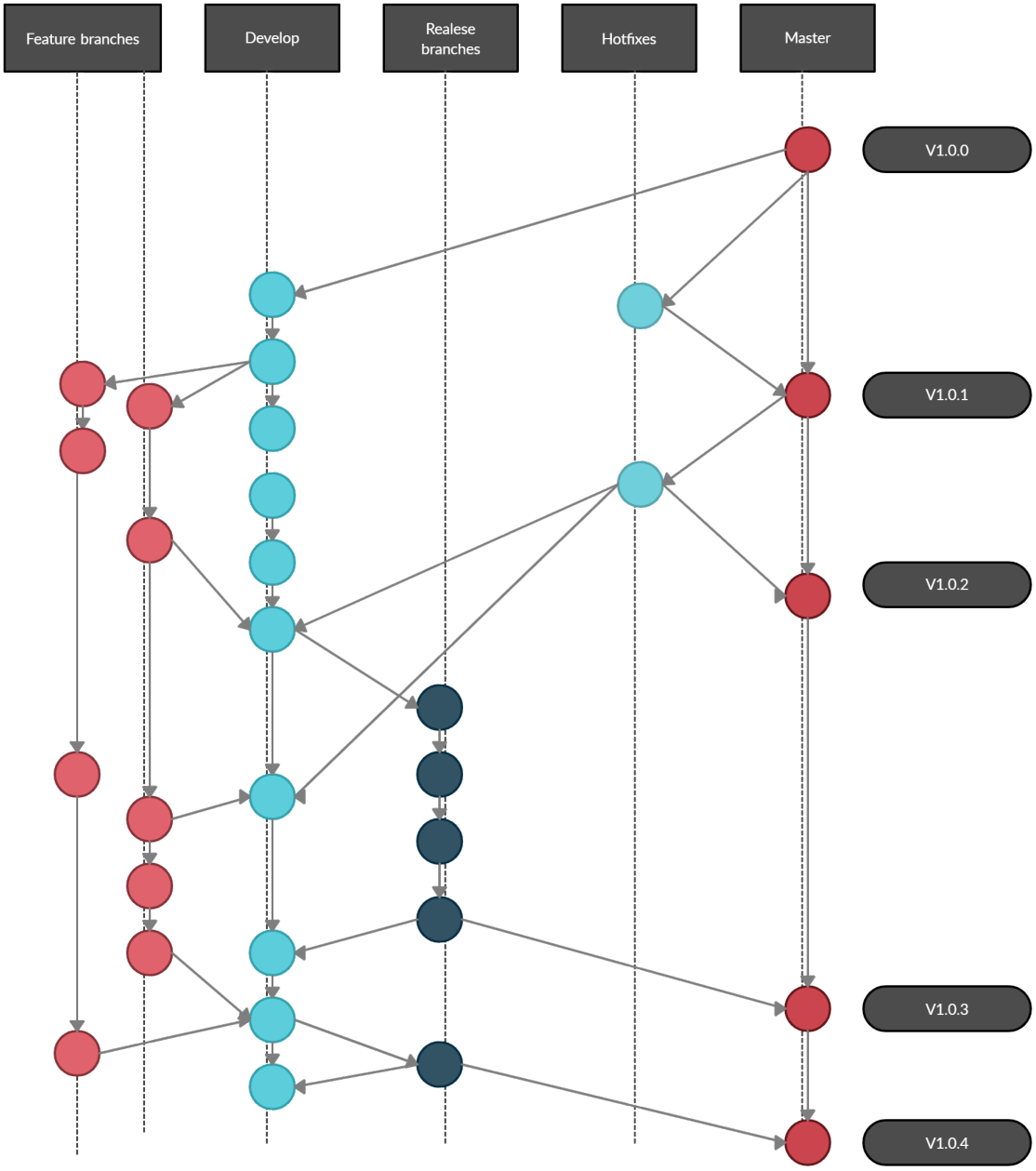


Figure 4-2: GitHub Workflow

There are many reasons behind a design decision, and the first lies in the understanding of versioning in design. Some researchers regard versioning as series, and design as a continuous process of generating a whole collection of entities, while others view it as a process of refining or developing a single entity through discrete design states. The thesis is biased towards the former because I believe version control in

the design field is not to plan a certain path for designers, but to prevent us from eliminating other possibilities in advance. Because of the above reasons, the platform does not have a merging function.

Although the design process is continuous in theory, it is impossible to produce infinite design results in practice. After all, we can't design without making decisions. Therefore, my strategy (figure 4-3) is to combine continuous similarity and discrete difference, that is, to save discrete design states of a project while tracking the continuous construction history between two design states.

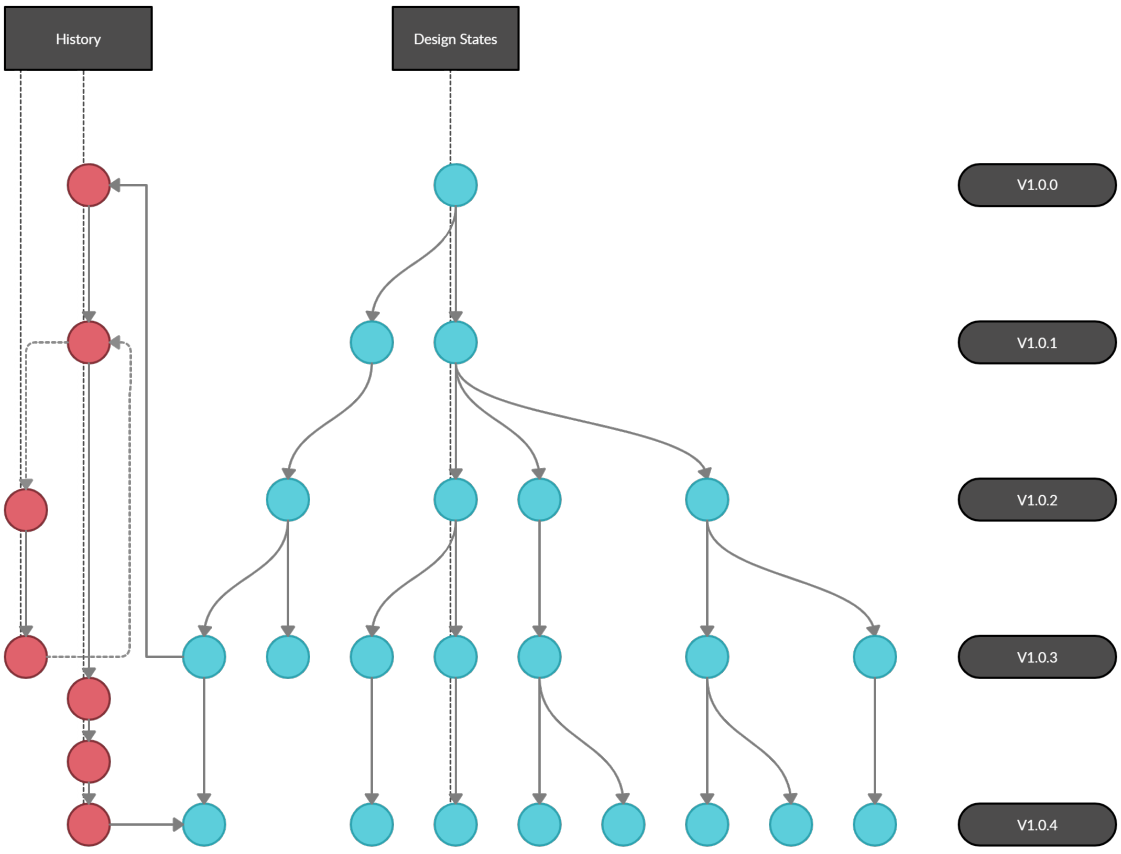


Figure 4-3: Platform Workflow that Combines Discrete Design States and Continuous Construction History Between Two Design States

Another factor to consider when designing platform workflow is the needs of users in the context of the times. Due to the pandemic, students are forced to study in a distributed manner, resulting in unprecedented isolation and loneliness. Compared

with previous research, the platform in the thesis pays more attention to the two-way communication between students. In addition to the asynchronous communication functions implemented by other platforms, a real-time chat function is added to enable direct communication. Figure 4-4 demonstrates the platform workflow in practice.

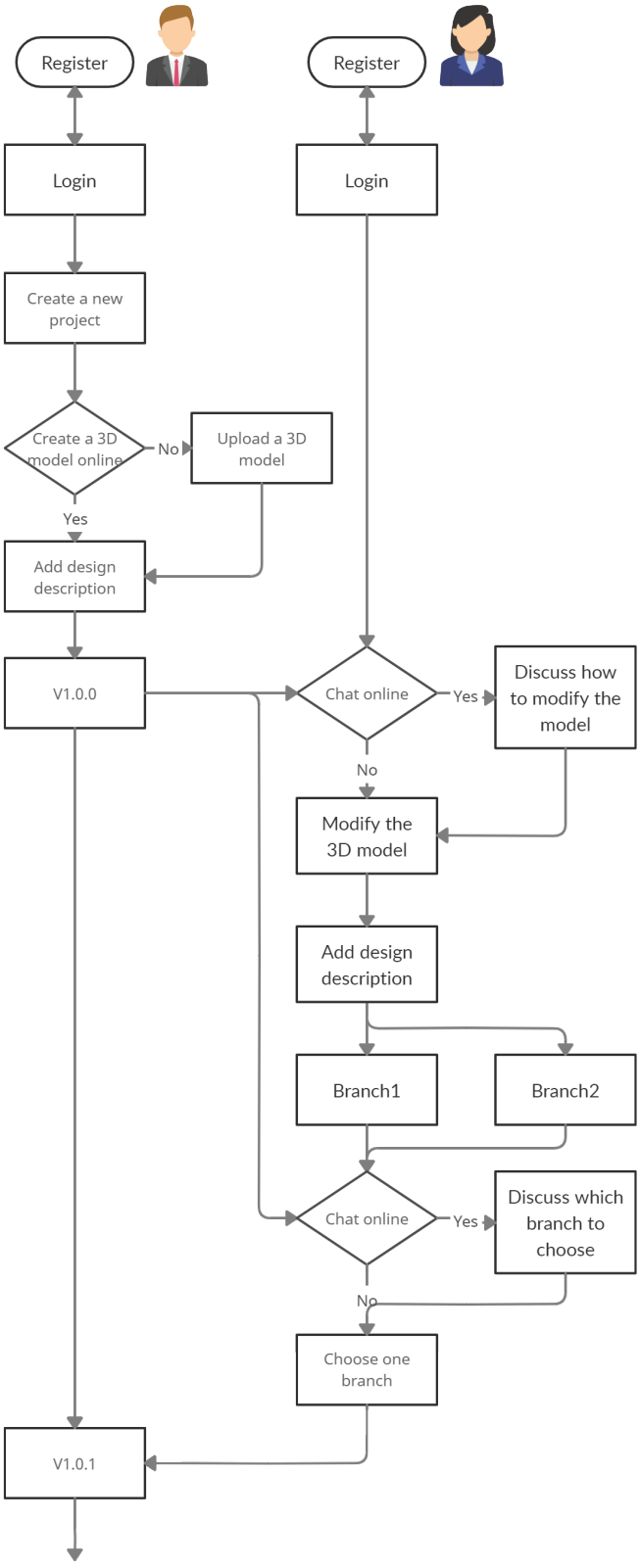


Figure 4-4: Platform Workflow in Practice

4.3 Core Functions

Figure 4-5 shows the core functions required to implement version control, which can be summarized as project management, design state management, and 3D model management.

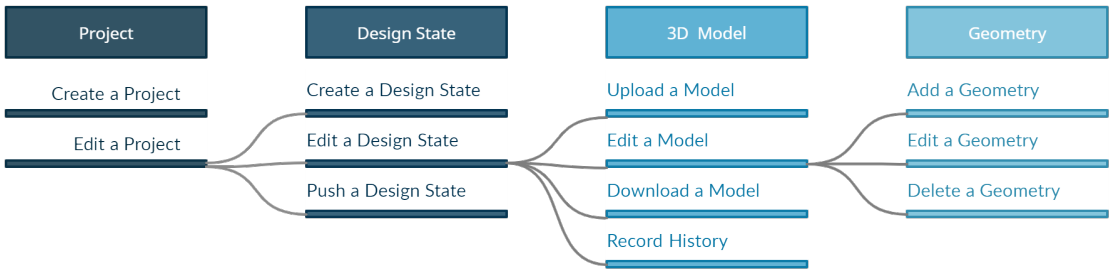


Figure 4-5: Core Functions

4.3.1 Project Management

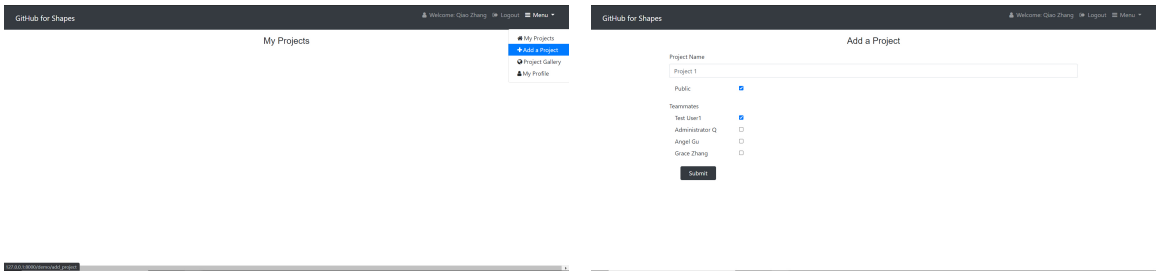
Table 4.1 shows the project model in the platform.

data name	data content
name	project name
time	project update time
user	project creator
teammate	teammates
public	project visibility to all other users

Table 4.1: Project Model

Create a New Project

By clicking "Add a Project" in the menu (figure 4-6 (a)), the user will be redirected to the project adding page (figure 4-6 (b)), where he can enter the project name, choose if to make it public, add team members, and then click "Submit" to add a new project instance (table 4.2) to the database.



(a) Main Page (b) Project Adding Page

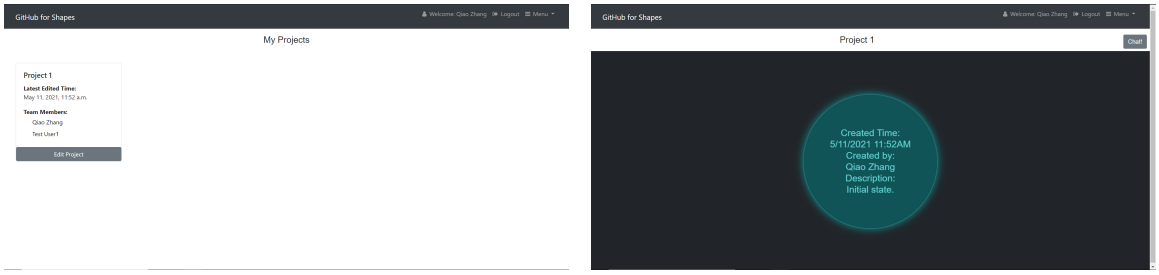
Figure 4-6: Create a New Project

data name	data content
name	user input
time	the time when the "Submit" operation is performed
user	the user who performs the "Submit" operation
teammate	user input
public	user input

Table 4.2: New Project Data

Edit a Project

The main page (figure 4-7 (a)) contains information cards for all projects the user has participated in. An information card includes the project name, last edited time and team member information, on which the user can click "Edit Project" to make modifications. The user will be redirected to the project page (figure 4-7 (b)).



(a) Main Page (b) Project Page

Figure 4-7: Edit a Project

4.3.2 Design State Management

Table 4.3 shows the design state model in the platform.

data name	data content
time	design state creation time
user	design state creator
description	design state description
version	version number
data	JSON data containing 3D model information
parentID	the ID of the previous design state
project	the project to which the design state belongs

Table 4.3: Design State Model

Create a Design State

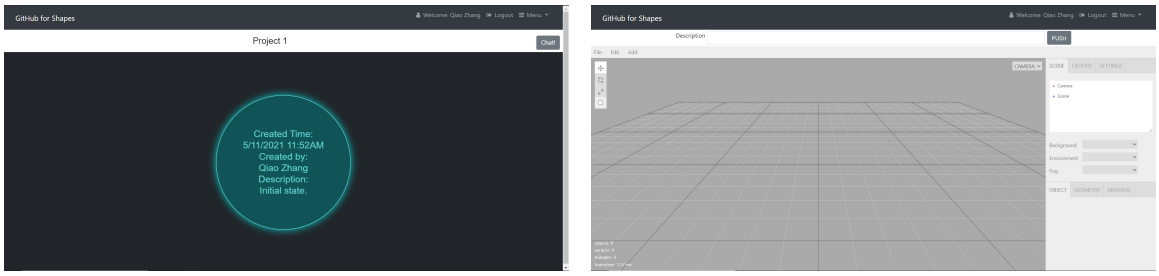
When a new project is created, its first design state (table 4.4) will be created automatically (figure 4-7 (b)).

data name	data content
time	the time when the project is created
user	project creator
description	no description
version	1
data	none
parentID	no
project	the project

Table 4.4: First Design State Data

Edit a Design State

On the project page, click a design state to make modifications (figure 4-8 (a)). The user will be redirected to the edit page (figure 4-8 (b)).



(a) Project Page (b) Edit Page

Figure 4-8: Edit a Design State

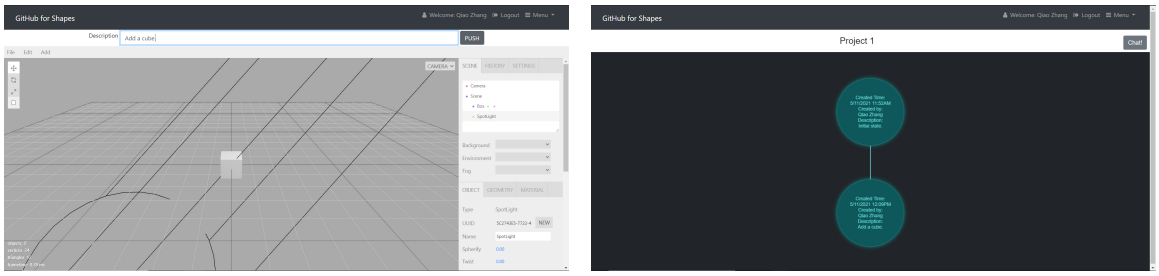
Push a Design State

The research believes that the final result of architectural design should be a collection of designs rather than a single entity, so the "push" here is actually a combination of "push" and "branch" in GitHub. Specifically, GitHub performs "push" and "branch" operations on the latest state to create/save new states, while the platform performs "push" operations on any selected design state to achieve the same function.

On the edit page (figure 4-9 (a)), enter design description in the text box and click "Push" to save a design state (table 4.5). The user will be redirected to the project page (figure 4-9 (b)). Each design state contains information including created time, creator and description. And figure 4-10 demonstrates a project after pushing some design states.

data name	data content
time	the time when the "Push" operation is performed
user	the user who performs the "Push" operation
description	user input
version	add 1 to the version number of the selected design state
data	updated JSON data
parentID	the ID of the selected design state
project	the project to which the design state belongs

Table 4.5: New Design State Data



(a) Edit Page (b) Project Page

Figure 4-9: Push a Design State

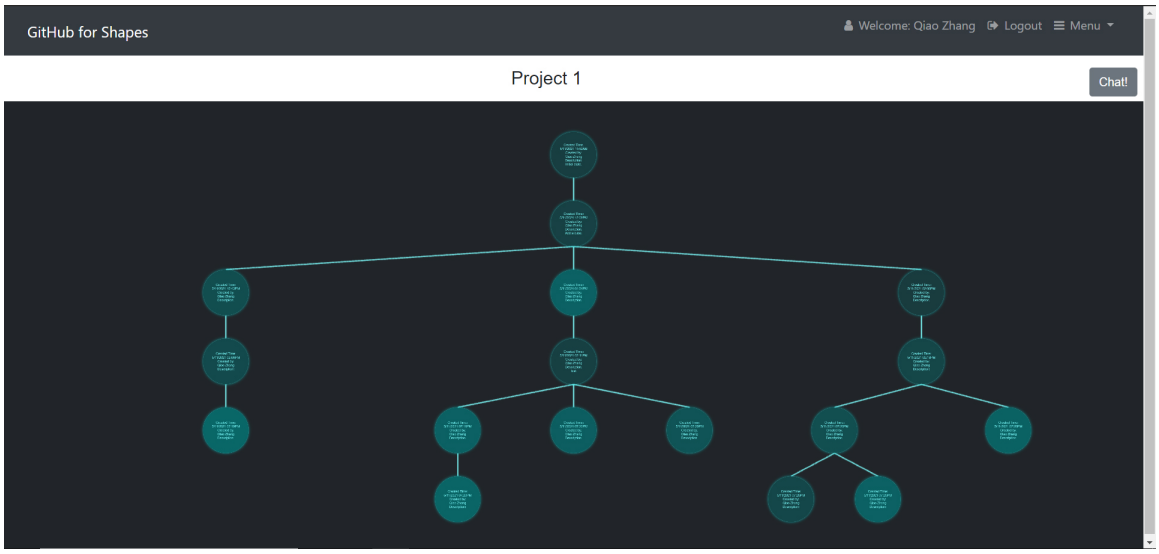


Figure 4-10: A Project After Pushing Some Design States.

4.3.3 3D Model Management

The information of the 3D model is stored in JSON format.

Import/Export a 3D Model

When users click "Export Object/Scene/Design State" on the edit page, the platform will convert an object/a scene/a design state into JSON format and export it (figure 4-11). As shown in figure 4-12, users can click "Import" to import files in different formats (JSON, DAE, 3dm, obj, etc.) using the Three.js file loader. Besides, users can develop designs created by software such as SketchUp (figure 4-13) and Rhino

(figure 4-14) on this platform, and vice versa.

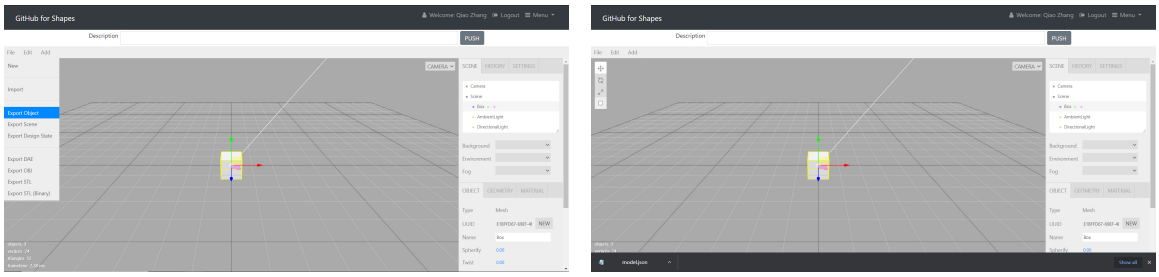


Figure 4-11: Export an Object

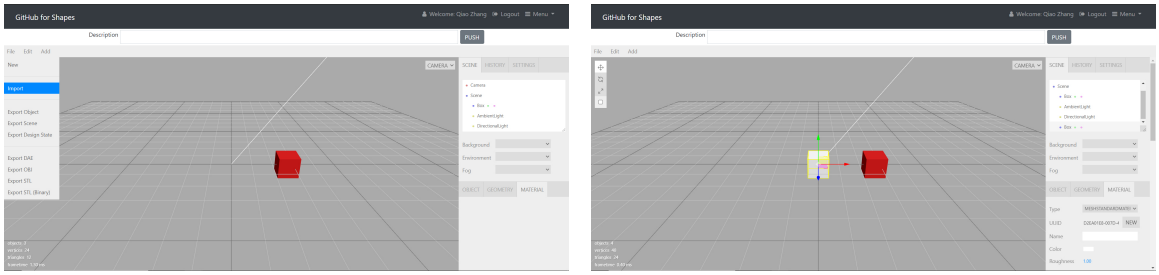
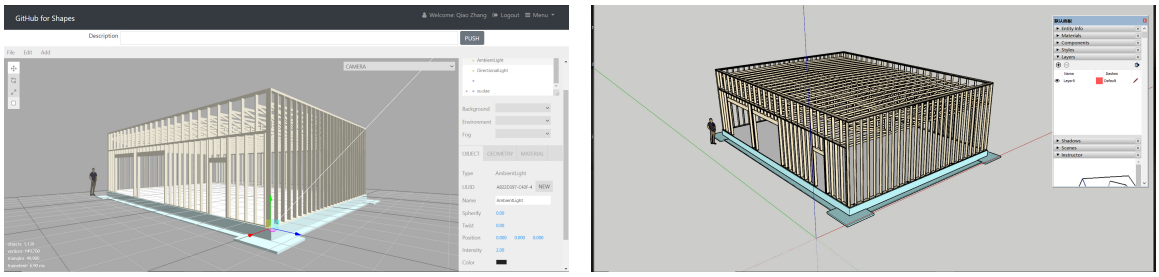
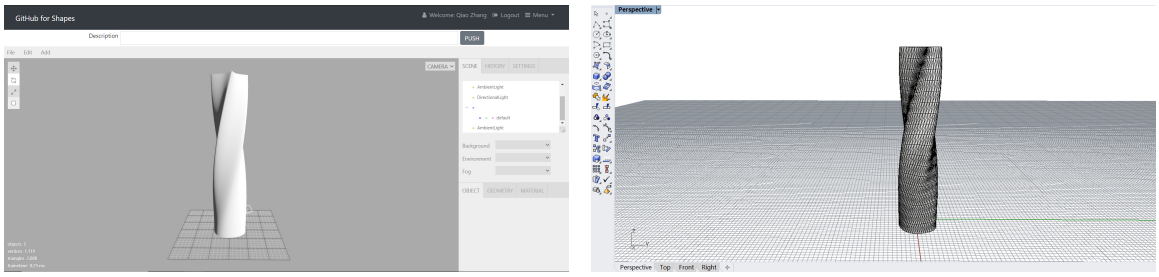


Figure 4-12: Import an Object



(a) SketchUp → DAE File → Platform (b) Platform → DAE File → SketchUp

Figure 4-13: Import/Export Files from/to SketchUp



(a) Rhino(3dm) → Platform (b) Platform → OBJ File → Rhino

Figure 4-14: Import/Export Files from/to Rhino

Edit a 3D Model

On the the edit page, the user can edit the 3D model online with the functions provided by the platform, including adding/deleting geometry (figure 4-15), transforming (figure 4-16), and target morphing (figure 4-17), which are implemented by Three.js.

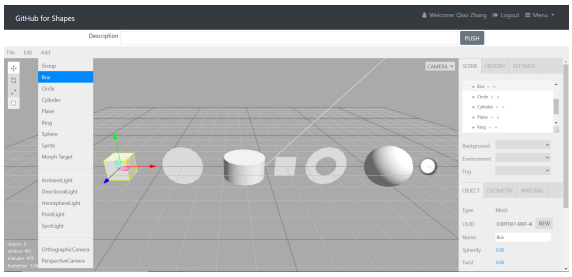


Figure 4-15: Add an Object

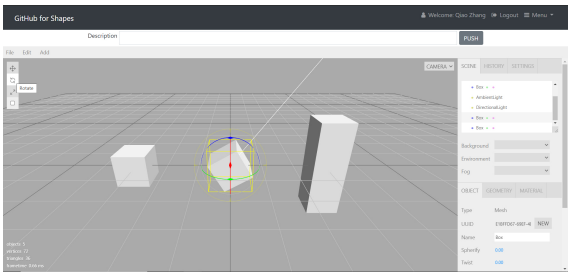
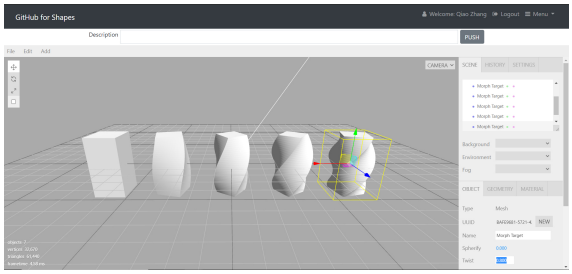
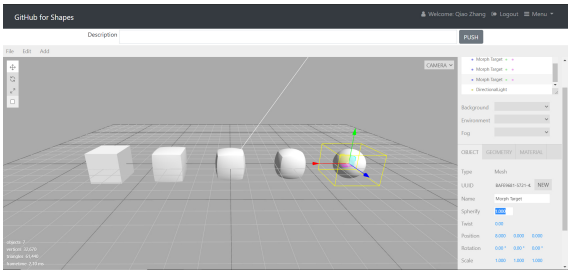


Figure 4-16: Transform



(a) Twist



(b) Spherify

Figure 4-17: Target Morphing

Record History

As explained in section 4.2, the platform allows the tracing of the continuous construction history between two design states. The platform will temporarily store construction history records (figure 4-18) so that the user can compare any state between the starting state and the latest state, and return to any state he is satisfied with (figure 4-19) and start from it (figure 4-20). The user can make multiple attempts and repeated modifications in a short period of time, and push a satisfactory version at any time.

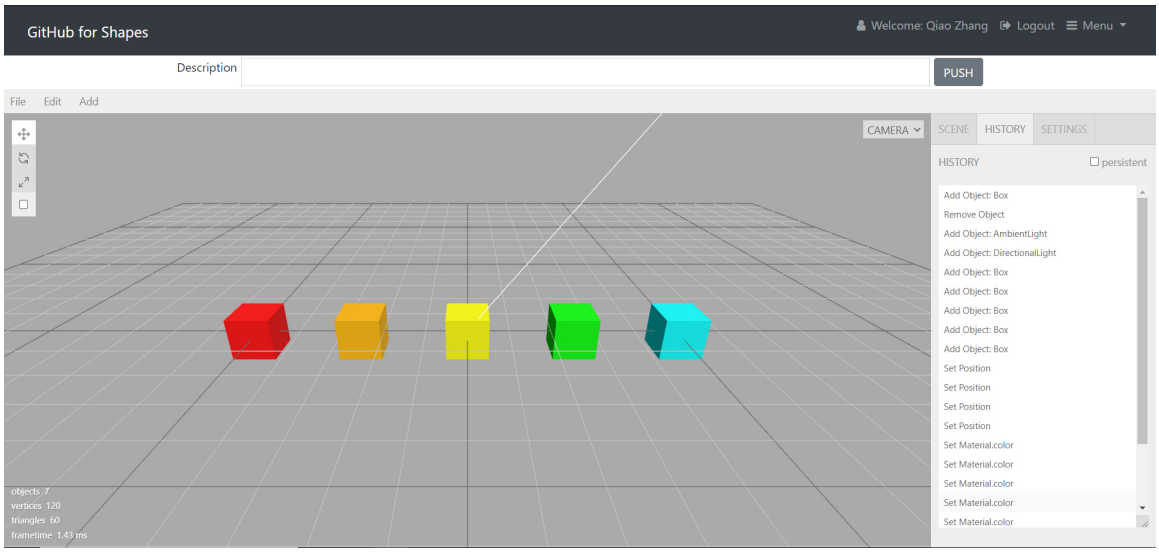


Figure 4-18: Record History

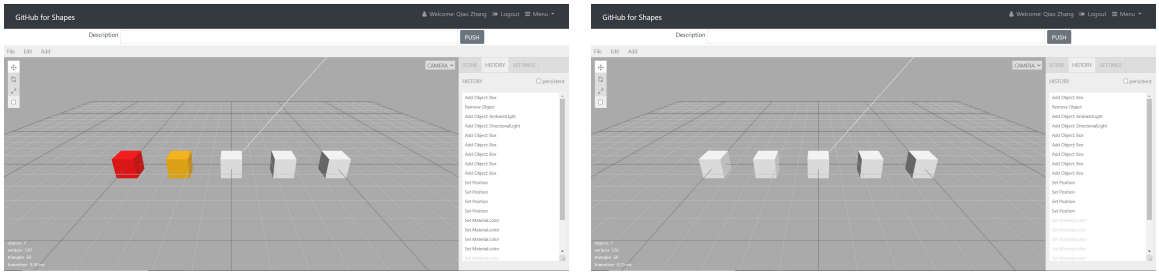


Figure 4-19: Return to any Previous State

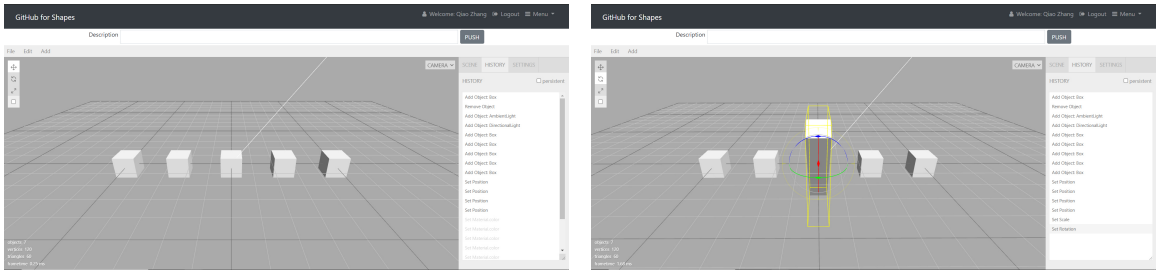


Figure 4-20: Start from a Previous State

4.4 Auxiliary Functions

4.4.1 Register/Login/Logout

Users are required to register a new account (figure 4-21) and login (figure 4-22) to use the online collaborative design platform. This mechanism considers community belonging and information security. Users can cooperate with any member of the community and will not be disturbed by others when sharing data with teammates.

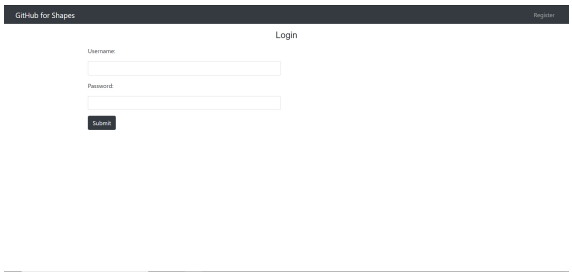


Figure 4-21: Register

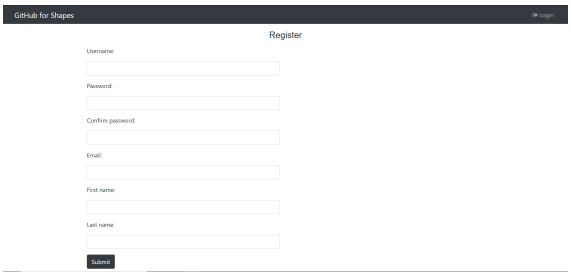


Figure 4-22: Login

4.4.2 Chat Synchronously

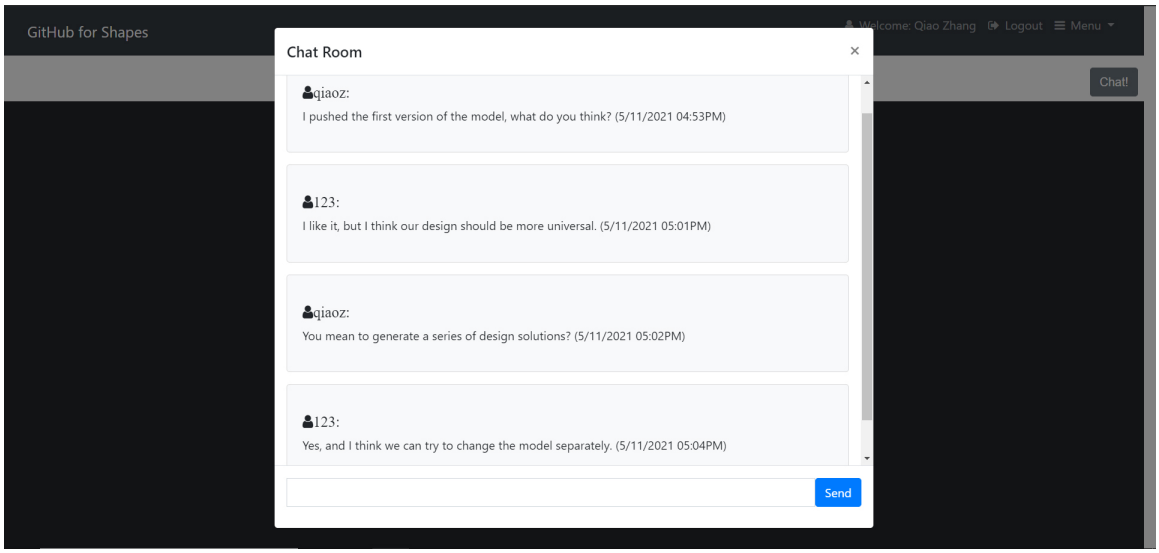


Figure 4-23: Chat Synchronously

To make communication more direct and efficient, a real-time chat function is added to the platform (figure 4-23). In the chat room, users can synchronize with their

teammates to discuss ideas about the latest version of the design. This function is implemented using JavaScript and Ajax.

4.4.3 View Public Project Gallery

By clicking "Project Gallery" in the menu, the user will be redirected to the project gallery page, where users can view public projects created by thousands of other members (figure 4-24).

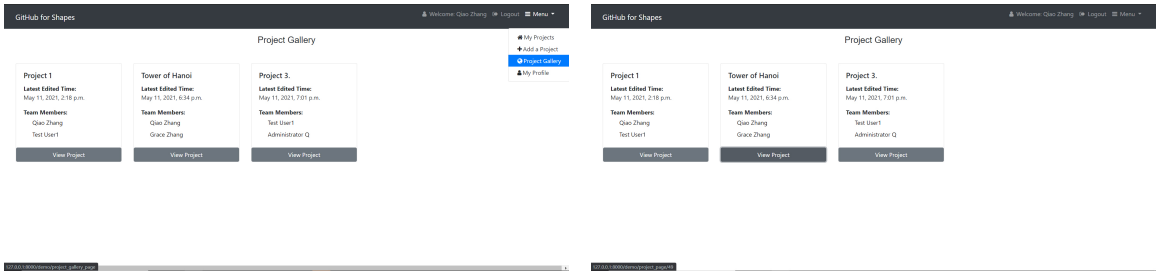
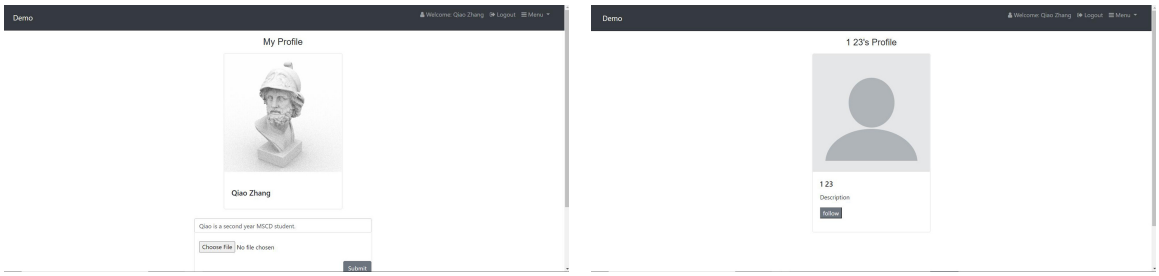


Figure 4-24: View Public Project Gallery

4.4.4 User Profile

By clicking "My Profile" in the menu, the user will be redirected to the user profile page (figure 4-25 (a)), where users can upload profile picture and description.

On the main page, the project gallery page, and the project page, by clicking another user's name, the user will be redirected to the profile page of another community member (figure 4-25 (b)).

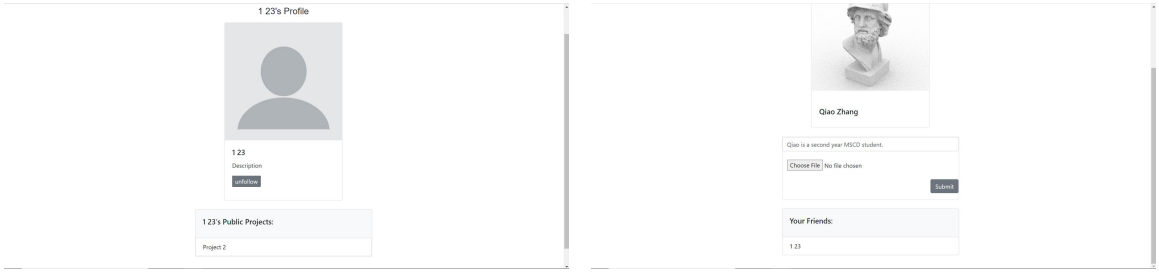


(a) My Profile (b) Other User's Profile

Figure 4-25: User Profile

4.4.5 Follow/Unfollow Other Users

On the profile page of another community member (figure 4-25 (b)), click "follow", a list of his public projects will be displayed (figure 4-26 (a)), and the user can click any of them to view. At the same time, the community member you follow will appear in the "Your Friends" list on the user's profile page (figure 4-26 (b)). Click ”unfollow“ to unfollow a user.



(a) Public Projects List (b) Friends List

Figure 4-26: Follow/Unfollow Other Users

4.5 Discarded Functions

Due to the low priority, the feature of highlighting differences in the GitHub workflow is discarded on the platform. GitHub can highlighting code differences with different colors, which is intuitive and clear. But for the shape, its information is stored in JSON format. I uploaded two JSON files to the online diff finder and figure 4-27 demonstrates the results. Although many differences have been found, it is difficult to tell what changes have been made, because even if I moved only one vertex, other data related to it (position, normal, etc.) have changed. Usually, the difference between two 3D models is very intuitive, and users can know directly without highlighting. Priority should be attached to differences between design states, as this is where key design decisions are made, not trivial differences between 3D models.

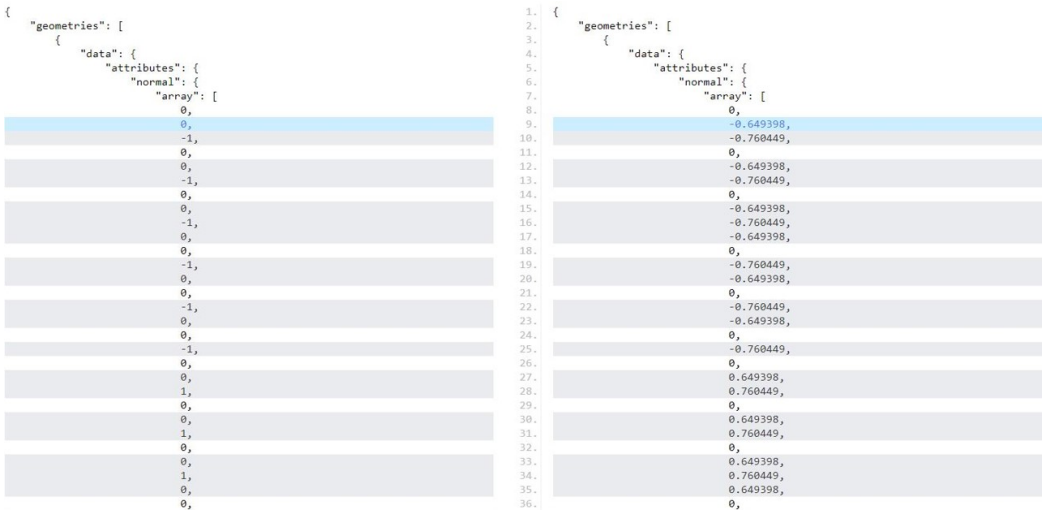


Figure 4-27: Results Obtained by the Diff Finder

4.6 Evaluation Indicators

Professional and amateur volunteer students will be asked to use and evaluate the online collaborative platform to verify my thesis hypothesis. Refer to Tinggui Chen's paper[29], table 4.6 is used to evaluate the user satisfaction of the platform. Evaluation indicators include the degree of satisfaction, the intention of continuous use, quality of interaction, quality of service, platform availability, and so on. Based on the research by Ellen et al.[30], table 4.7 is used to evaluate the collaboration.

The Primary Variable	The Secondary Variable	Indicators
User's willingness to continue using	Recommend to others	You'd like to recommend the platform to others
	Increase the frequency of use	You'll use the platform more often in the future
User satisfaction	Collaborative design needs	You think the existing functions of the platform can meet your collaborative design needs
	Use feeling	You are satisfied with the platform
	Attractive	Compared with offline collaborative design, you think the online platform is more attractive
Platform availability	Easy to learn	The operation of the platform is easy to learn
	Easy to browse	The platform navigation structure is simple and clear, and the web page is easy to browse
	Interface design	The interface design of the platform is reasonable
The quality of interaction	Participation	You will actively participate in the design process
	Feedback	You can get the feedback from your teammates on time
The quality of service	Error handling	When the platform fails, you will know what the problem is and solve it following the instructions.

Table 4.6: Indicators for Evaluating the Platform

The Primary Variable	Indicators
Shared Vision	We have a shared and clearly understood vision
Goals and Objectives	Members understand and agree on goals and objectives
Responsibilities and Roles	Members are clear about their roles
Decision Making Procedures	We have effective decision-making procedures
Changing Membership	We have procedures for changing members
Conflict Management	We are able to manage conflict successfully
Leadership	Leadership is effective and shared when appropriate
Action Plans	Plans are well developed and followed
Relationships/Trust	Members trust each other
Internal Communication	Members communicate well with each other
External Communication	Our external communication is open and timely
Evaluation	We have built evaluation into all of our activities

Table 4.7: Indicators for Evaluating Collaboration

Chapter 5

Use Case Testing and User Evaluation

In the research, professional and amateur volunteer students were asked to follow these three steps to evaluate the online collaborative platform.

- Test users will be asked to explore the platform on their own without any instructions. In the process, they are welcome to ask the developer when encountering a problem.
- Test users will conduct collaborative design with the developer (and other users) on the platform.
- Test users will be asked to rate the platform based on the evaluation indicators in Chapter 4.

Their feedback helps to determine the limitations and gaps of the platform and further guide subsequent development.

5.1 User Individual Exploration

`https://www.dropbox.com/s/eaoira2u707afaz/SHdemo.mp4?dl=0` is a demo video recorded by the researcher, and `https://www.dropbox.com/s/1wxtm33rvyer3dq/SHexplore.mp4?dl=0` is a video of a test user exploring the platform alone.

5.2 Use Case Testing

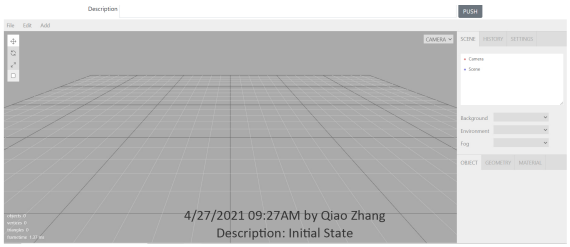
The research conducted three sets of use case tests, including scenarios with different number of users, software used, and user types, in order to perceive and evaluate the platform from a more diverse perspective.

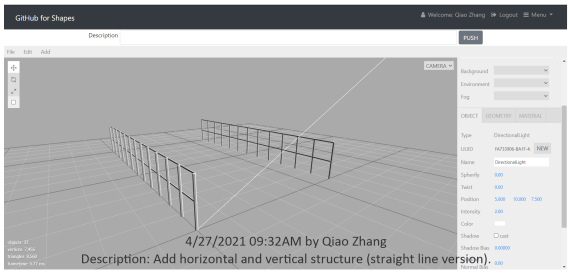
5.2.1 Case 1: Collaboration Between Architecture Students Through the Platform and Rhino

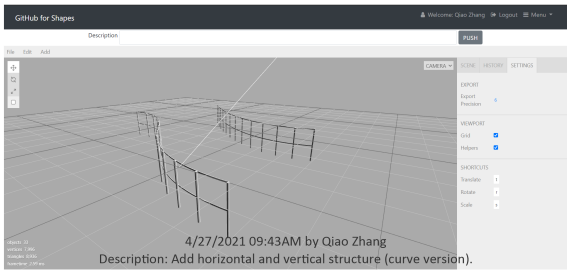
Angel G.[1] (major: architecture, geographical location: China) was the first user invited to test the platform with the developer. She was proficient in various 3D modeling software before the test, so the two collaborators decided to use the platform and Rhino to design a truss.

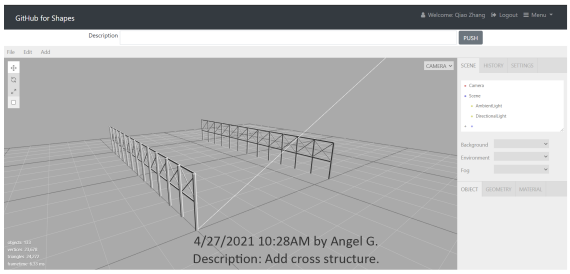
Because the platform exported the 3D model as a mesh, we almost only performed one-way operations from Rhino to the platform. The design process is as follows. I designed two versions of horizontal and vertical structures, of which she chose the straight-line version and added a cross structure. Then I added the roof structure and she refined it with details. Figure 5-1 and figure 5-2 respectively show all design states and design state structure.

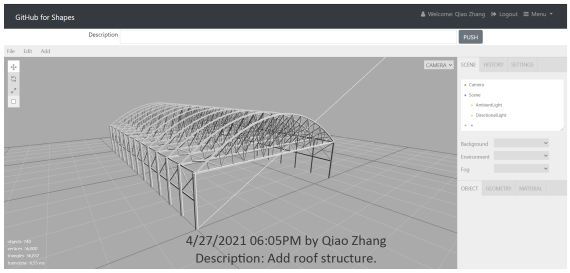
[1]The name of each test user is a changed name.











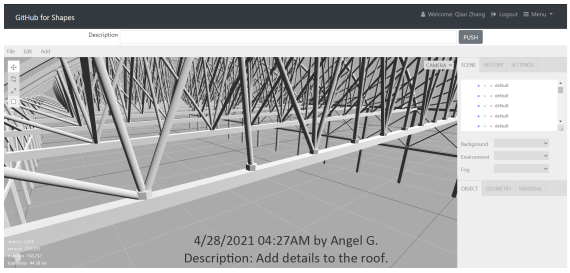


Figure 5-1: Design States of Case 1

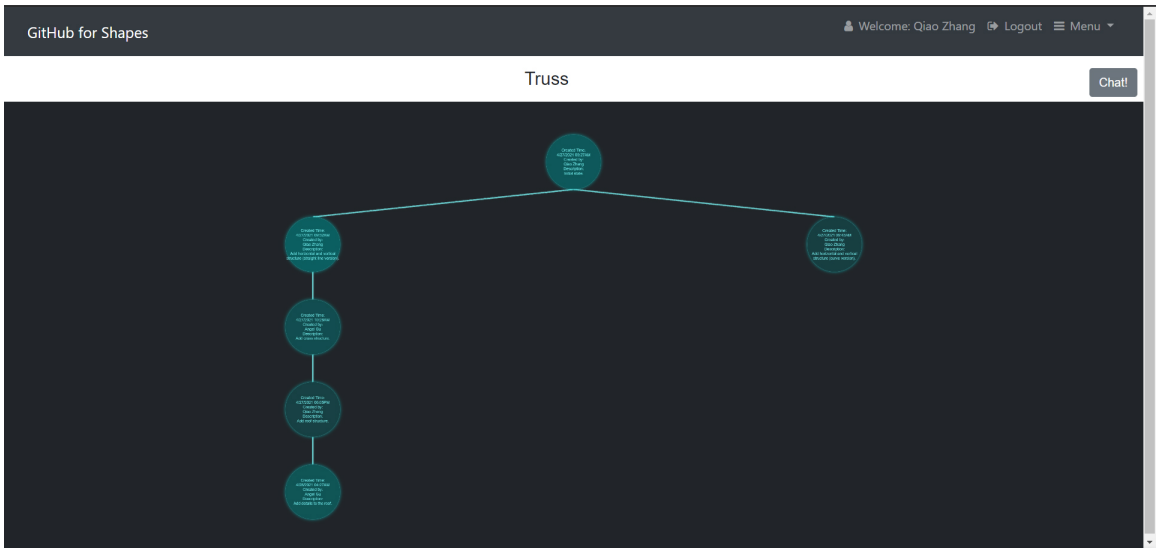


Figure 5-2: Design State Structure of Case 1

5.2.2 Case 2: Collaboration Between Architecture Students Through the Platform and SketchUp

Linda T. (major: urban planning, geographical location: China) and Jason W. (major: architecture, geographical location: China) were invited to test the platform with the developer. The three collaborators decided to use the platform and SketchUp to collaborate in the early design stages of a building complex.

The design process is as follows. The three of us performed a volume design based on the same site model. We discussed and selected a new starting point, on the basis of which two members proposed different design states. After online chatting and making decisions, we no longer tried independently, but iterated on another member's design. Figure 5-3 and figure 5-4, respectively show all design states and design state structure.

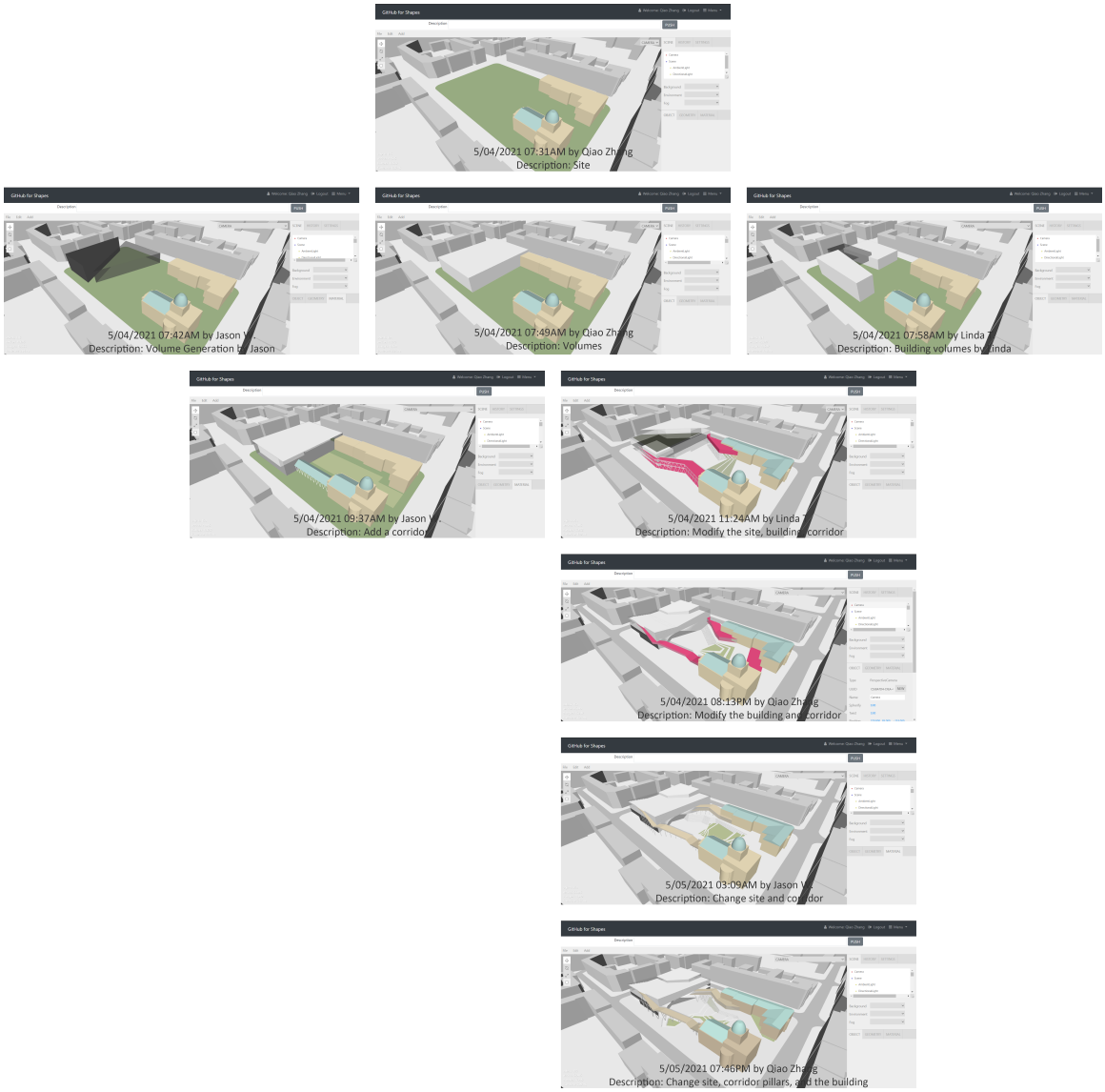


Figure 5-3: Design States of Case 2

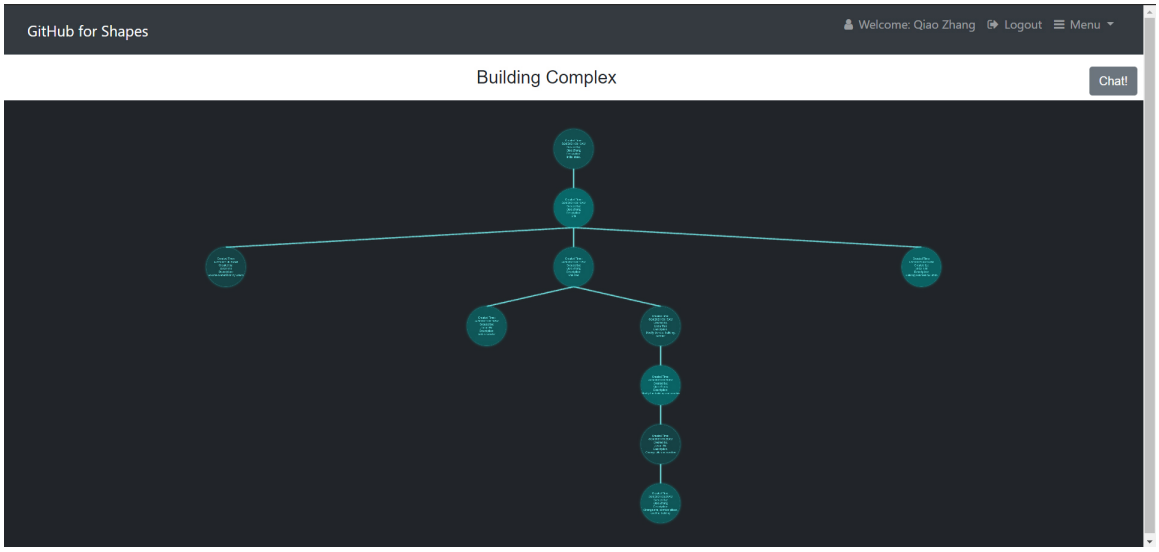


Figure 5-4: Design State Structure of Case 2

5.2.3 Case 3: Collaboration Between Professional and Amateur Students

Grace Z. (major: mechanical engineering, geographical location: Berkeley, US), was invited to test the platform with the developer. Before the test, she was not familiar with 3D modeling software commonly used in architecture, and only used 3D software for robotic arm simulation. The developer collaborated with her on the Tower of Hanoi project, which is not a collaborative design in the traditional sense, but more like a game where a professional guides a non-professional.

The process is as follows. After I created the 3D model, she tried two color schemes. I chose one, and we started playing the Hanoi Tower Game. Since she was not familiar with the game, she made the wrong move. I added a design state, which performed the right move, to the last state so that the branch containing the wrong move can be discarded. Figure 5-5, figure 5-6, and figure 5-7 respectively show all design states, design state structure, and the chat room.

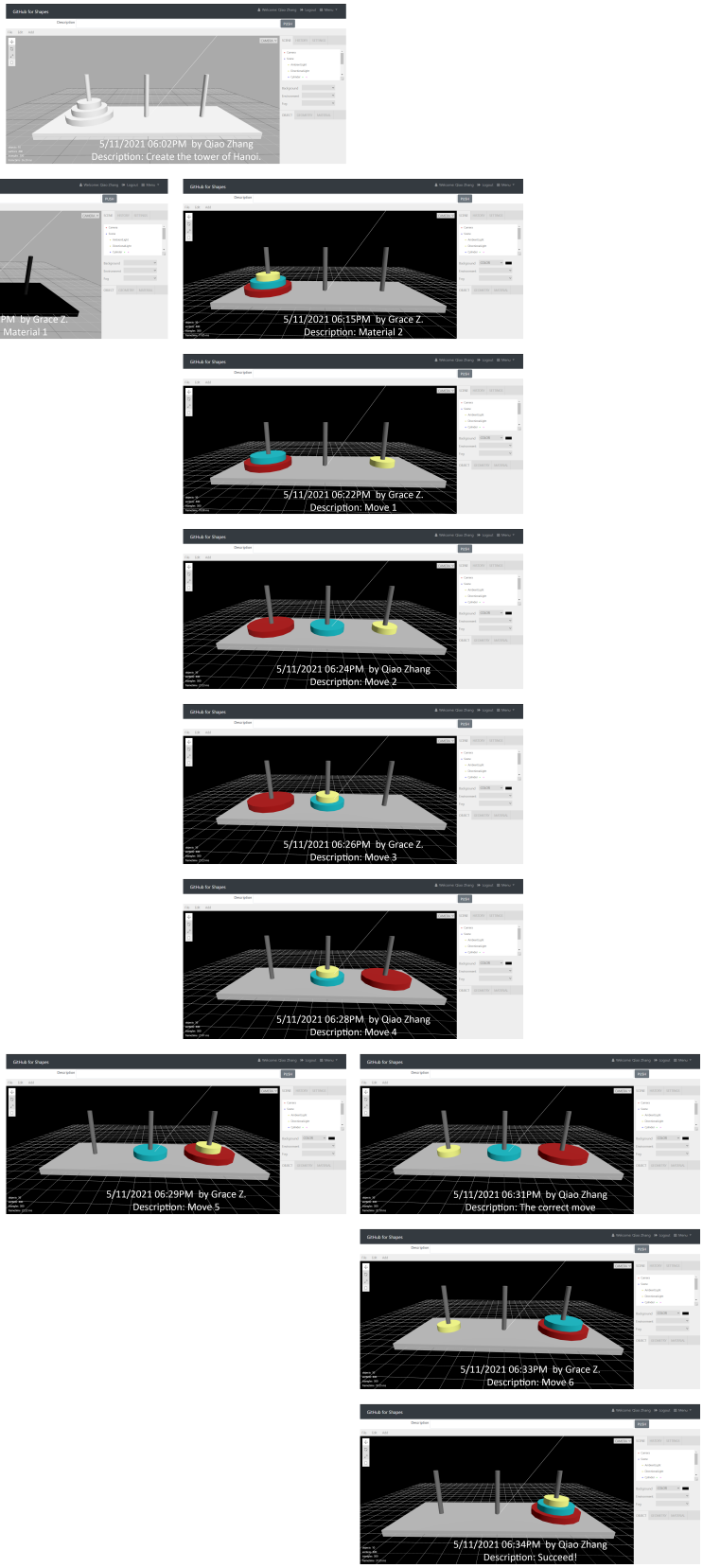


Figure 5-5: Design States of Case 3

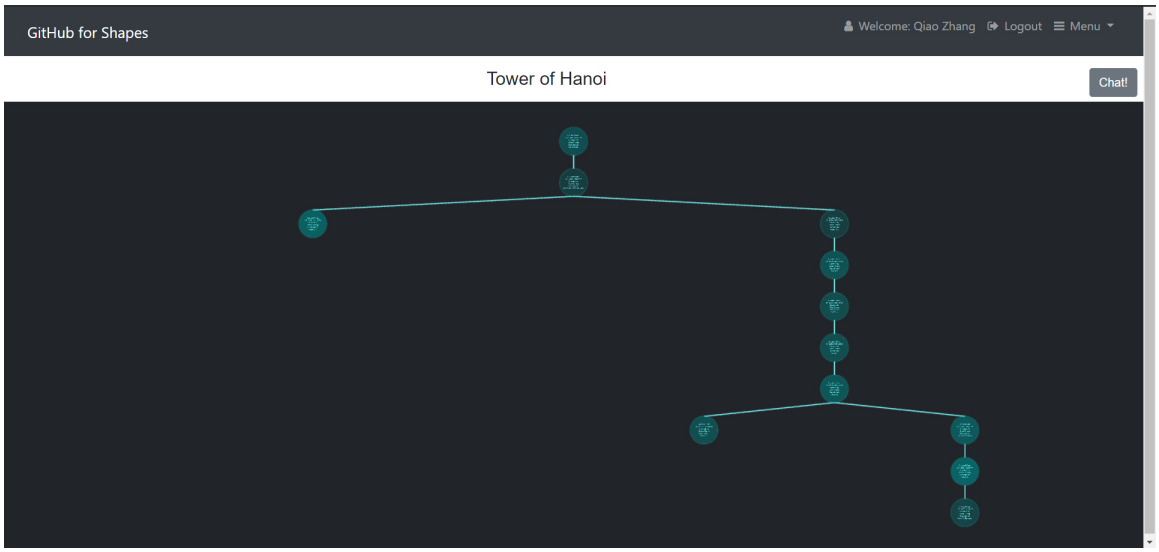


Figure 5-6: Design State Structure of Case 3

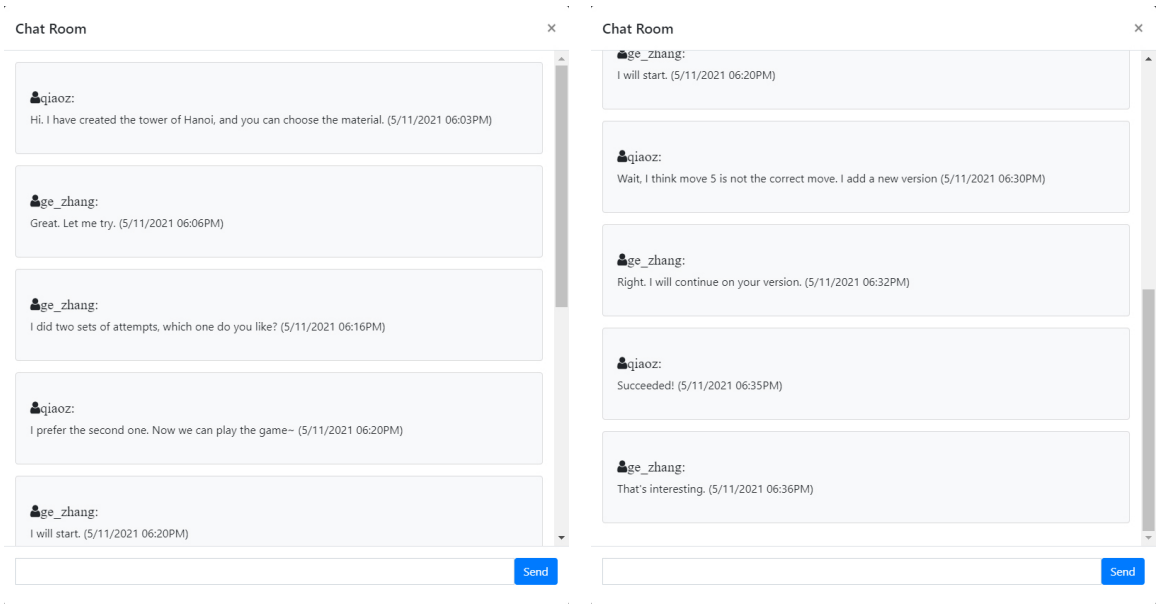


Figure 5-7: Chat Room

5.3 User Evaluation

5.3.1 Feedback from Professionals

The following is some feedback from test users of architecture majors:

Versioning

- Angel has almost no programming experience and has never used GitHub, so she was confused about versioning. She consulted the developer to understand how to perform version control on the platform and suggested adding a website introduction.
- Linda has python programming experience and used GitHub, but because the workflow of this platform is different from that of GitHub, it took her a while to figure it out. A user guide is advised.
- Some users would like to modify a design state (description or 3D model) instead of adding a new one.
- Some users want to delete a design state.

Modeling

- Majoring in architecture, these users often use modeling and rendering software, so they find it easy to learn the modeling part.
- Although it is easy to get started, it is difficult to browse all the functions without instructions.
- Angel believes that the platform needs further support for Rhino. Although files can be imported and exported, since the model exported by the platform is a mesh, it is not convenient to make subsequent modifications in Rhino. Angel deems that the platform enables her to have a more holistic understanding of the entire design process.
- Linda and Jason believe that the platform provides better support for SketchUp, except for the inconvenience of file format conversion when importing files. What they like most is that the platform can retain the type and hierarchical structure of models.

Communication

- Jason did not notice the existence of the chat room at first, but after being notified by the developer, he thought it was the highlight of the collaborative design platform.
- Some users enjoy interacting with other users through features such as "follow".

5.3.2 Feedback from Amateurs

The following is the feedback, mainly experiences and thoughts, from Grace Z.(ME major).

"As a person who barely used any 3D modeling software, I think that this platform is easy to use for beginners. It is a functional platform that can satisfy the basic needs for 3D design. It is not hard to learn basic operations such as adding shapes, rotating the objects, adding lights, changing textures and so on. If there are explanations for different types of light, it would help learners to get a clue what kind of light is more proper to be added to the environment based on various needs.

The feature that surprised me the most was the built-in chat module on the web page. I could communicate with teammates easily, which improve the cooperation efficiency and user experience. Compared with offline collaborative design, this platform is definitely more efficient since you can get feedback timely.

Besides, the idea of keeping the branches of the project is also impressive. I was asked by the developer to collaboratively play a Tower of Hanoi game. Since I am not familiar with this game, I made a wrong move when playing the game. The developer added a new branch, which did the right move, to the last branch so that the branch containing the wrong move could be abandoned. This way, the whole game process can keep going and will not be interrupted by a wrong move. In my opinion, the robustness of the platform is promoted and the efficiency of collaboration is elevated by keeping branches. However, it would be better if branches can be moved. If too many branches are generated, it will waste a lot of memories and also make the user harder to find the branches that they want to edit.

The design of the web page has its pros and cons. I really like the neat and artistic style of the web page. However, I have to say that the web page is not quite easy for novices to understand. I wasn't told by the developer any information about the platform before the test. When going through the web page, I was not sure what I should do. The projects listed in the project gallery, which are fascinating and interesting, couldn't be edited. It took me a while to notice that I could add my own project. As a beginner, I didn't get a clear picture of what I should do using this platform at first sight. It would be better to put some instructions on the website to lead beginners like me. For example, you may write "Give it a try by starting a new project".

Since I have no demand to build up 3D models, I may not use the platform in the future. But I am definitely willing to recommend the platform to someone in need and I will use it by myself if needed."

5.3.3 Rating Results

Test users were asked to rate the platform according to the criteria in Chapter 4, as a supplement to user feedback, with a minimum score of 1 and a maximum score of 5. Table 5.1 and table 5.2 describe the rating results for the platform and collaboration respectively.

Indicators	Angel G.	Linda T.	Jason W.	Grace Z.
Recommend to others	4	3	5	4
Increase the frequency of use	2	4	3	3
Collaborative design needs	3	5	4	5
Use feeling	4	4	5	5
Attractive	5	5	5	5
Easy to learn	3	4	2	4
Easy to browse	3	4	4	3
Interface design	5	5	4	5
Participation	3	5	5	5
Feedback	5	5	4	5
Error handling	4	3	4	4

Table 5.1: Rating Results for the Platform

Indicators	Angel G.	Linda T.	Jason W.	Grace Z.
Shared Vision	3	4	5	4
Goals and Objectives	5	4	5	5
Responsibilities and Roles	5	4	3	4
Decision Making Procedures	4	5	4	4
Changing Membership	4	4	5	5
Conflict Management	5	4	5	5
Leadership	2	3	3	2
Action Plans	4	4	4	4
Relationships/Trust	5	5	5	5
Internal Communication	5	5	4	5
External Communication	4	4	4	5
Evaluation	2	4	3	3

Table 5.2: Rating Results for Collaboration

Chapter 6

Contributions and Future Work

6.1 Research Contributions

First, the direct technical contribution of this research is the development of a web-based collaborative design platform, where teammates can communicate and design collaboratively by sharing ideas and iterating on others' designs. The platform implements version control to improve the efficiency and experience of students in collaborative design.

Secondly, the thesis makes a contribution to architectural collaborative design in the educational context. In the context of work, the existing collaborative design software is more like a tool for architects to show design results to each other or to clients. Clients do not care about the design process, and architects who are accustomed to traditional design software are far behind the technology curve, so the interaction on these platforms is largely one-way. Clients and other architects can provide real-time advice to model publishers, but what they can do is very limited. There are many differences between the collaborative design of students in the studio and that of architects in practice. For example, the collaborative design of students is more experimental, requires two-way communication between teammates, and is less eager to obtain results. The following features of the platform better cater to the needs of students:

- The platform is more tolerant of "mistakes", "experiments", "uncertain", and "deliberation" in the design process. First, the platform records the historical records between the two design states, so the user can try as much as he wants, repeatedly deliberating between several design states, and if he is not satisfied with the current design, he can go back to any previous state. In addition, the branching mechanism also serves "trial and error". There are many starting points but no ending points in this system. Users can design from any design as a starting point, but there is no such thing as the only best result.

- The platform pays more attention to participation and two-way communication in the collaborative design process. For example, it allows online editing of 3D models. Although the functions are not complete, it is much more interactive than a simple viewer. And the platform supports synchronous chat and direct communication, so it is more efficient.

In addition, the research further explores version control in the design field. Although versioning has been introduced into the field of design in previous studies, its definition is controversial. Some researchers believe that design is a series of divergent and convergent explorations towards a final result, while others argue that design is a process of generating "one-of-a-kind customized variety" [20]. This research considers the theories of both parties to obtain a critical understanding of versioning in design in the educational context, and develops a collaborative design tool under its guidance.

Last but not least, "ShapeHub" created a prototype of an online collaborative design community. On the platform, users can view public projects created by thousands of other members, follow them, and even invite them to design together. It inherits the architecture studio culture and the community culture of GitHub, focusing on communication, interaction, sharing, collaboration, co-learning, and co-creation. Although far from complete and mature, the platform has the potential to develop into a collaborative community like GitHub.

6.2 Limitations and Future Works

- **Develop Based on User Feedback**

  A series of design and feedback iterations are expected. Here are some possible improvements based on the users' evaluation and feedback in the use case testing.

  First, adding the "modify/delete design state" features is suggested by some users. Just like GitHub users can use the "git commit -amend" command to modify the commit message and use the "git rm" command to delete files in the repository, adding these features can save memory and help management.

  Besides, several users believe that the platform needs further support for Rhino. The model exported by the platform is a mesh, so users need to perform operations such as MeshToNurb and MergeAllFaces to make subsequent modifications in Rhino. I thought about the problem during development, but since the platform was developed based on Three.js Editor, to implement the function, one needs to be familiar with Rhino's data types and structure and write a file exporter.

  Additionally, some users may have never used GitHub or 3D modeling software before, or it may be the first time to use an online platform for version control and collaborative design. They may not understand the workflow of the platform, its functions, how to use it, or encounter problems that the developer did not expect, so they ask for website instructions and user guidance in their feedback. To help users learn and utilize the platform, a detailed user guide is a feasible solution, but non-professionals have brought me new perspectives and inspiration. Grace Z. wrote 'For example, you may write "Give it a try by starting a new project"', which reminds me that improving the user interface and navigation is also worth a try.

  Last but not least, the researcher abandoned the feature of highlighting differences because she thought it had a low priority. However, it was suggested that the platform interface should move further from the "single state docu-

ment model"[31] of interaction, and the platform should provide multi-viewing functionalities that let designers juxtaposing and comparing designs.

- **Improve the Collaborative Design Community**

  The research only created a prototype of the collaborative design community, but it is far from complete and mature. Researchers need more profound and extensive research and thinking, including but not limited to:

  1. How do community members communicate with each other?
  2. How do community members get the latest developments and announcements of the project?
  3. How do community members connect with the project and make contributions as much as they can?
  4. How does the maintainer of the project communicate with the community?
  5. How does the maintainer of the project organize community members to participate in project decision-making and make more contributions?

  Fortunately, GitHub is a good reference case. It challenged the closedness of the software design field and achieved great success. In my opinion, architectural design is not originally a closed discipline. I hope that the platform will one day become an open-source community for members to interact, learn, and create together. Currently, if they are not collaborators, community members can only view public projects created by others. In the future, we anticipate all community members can like, comment, share, download, and even modify public projects. Possible improvements also include adding interactive communication functions (discussion etc), creating an official bulletin board, and establishing a community knowledge base. Considering security and privacy, this is bound to be a challenging process.

- **Establish an Evaluation System**

  The research regards versioning as a series, but it will be helpful to make design

decisions if the evaluation mechanism for many design states can be introduced. Current design decisions are mainly achieved through text communication between teammates, but it is foreseeable that an evaluation mechanism will help design decisions, where teammates will rate the design states, and the system will suggest "better" design schemes. Although essentially all decisions are made by people, data may provide new perspectives and broader ideas.

- **Extend User Type**

  The target users of the research are students majoring in architecture, and the application scenario is collaborative design. In order to make the research more universal, the types of users should be extended. The researcher invited a non-professional student to test the platform to gain more perspectives, but in view of the differences in the characteristics and needs of different types of users, extending user types requires more use case testing and research. For example, the researcher could map out a range of possible alternative games (e.g. chess) and strategies to the Hanoi Tower game.

Bibliography

[1] Colin Gray and Craig Howard. "why are they not responding to critique?": A student-centered construction of the crit. 06 2015.

[2] American College Health Association. The impact of covid-19 on college student well-being. 2020. https://www.acha.org/documents/ncha/Healthy_Minds_NCHA_COVID_Survey_Report_FINAL.pdf.

[3] Aysu Sagun, Halime Demirkan, and Mesut Goktepe. A Framework for the Design Studio in Web-Based Education. *Journal of Art & Design Education*, 20(3):332–342, 2001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-5949.00282.

[4] I. Rocker and M. Horne. Versioning: Architecture as series? 2008.

[5] Y. Sakai and Daisuke Tsunoda. Implementation of decentralized version control in collective design modelling. 2015.

[6] Verina Cristie and Sam Joyce. 'ghshot': a collaborative and distributed visual version control for grasshopper parametric programming. 09 2019.

[7] Terry. Flew. *New media : an introduction.* Oxford University Press, South Melbourne, Vic. ;, 2nd ed. edition, 2005.

[8] Kelli Cargile Cook and Keith Grant-Davis. *Online Education: Global Questions, Local Answers.* Routledge, April 2020. Google-Books-ID: JjXfDwAAQBAJ.

[9] Benjamin H George. Identification of the Constraints and Barriers to the Adoption of Distributed Design Education. page 181.

[10] Mao-Lin Chiu. An organizational view of design communication in design collaboration. *Design studies*, 23(2):187–210, 2002.

[11] Bryan O'Sullivan. *Mercurial: The Definitive Guide.* O'Reilly Media, Inc., 2009.

[12] Daniel Smithwick. Architectural design 2.0 : An online platform for the mass customization of architectural structures. 08 2010.

[13] Tom Rodden and Gordon Blair. CSCW and Distributed Systems: The Problem of Control. In Liam Bannon, Mike Robinson, and Kjeld Schmidt, editors, *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW '91*, pages 49–64. Springer Netherlands, Dordrecht, 1991.

[14] Jim Budd, Surya Vanka, and Andy Runton. The ID-Online Asynchronous Learning Network: a 'Virtual Studio' for Interdisciplinary Design Collaboration. *Digital Creativity*, 10(4):205–214, December 1999.

[15] Torvalds Linus and Diamond David. *Just for Fun: The Story of an Accidental Revolutionary.* Harper Audio, 2001.

[16] Verina Cristie and Sam Joyce. Ghshot: 3d design versioning for learning and collaboration in the web. pages 1–6, 04 2018.

[17] Terry Griffin and Shawn Seals. Github in the classroom: Not just for group projects. *J. Comput. Sci. Coll.*, 28(4):74, April 2013.

[18] Wilson Greg. Github for education. https://software-carpentry.org/blog/2012/04/github-for-education.html.

[19] Ingeborg Rocker. Versioning: Evolving architectures - dissolving identities 'nothing is as persistent as change'. 72, 01 2002.

[20] Greg Lynn. *Folding in Architecture (1993).* 09 2015.

[21] B. Hartmann, Sean Follmer, A. Ricciardi, Timothy Cardenas, and Scott R. Klemmer. d.note: revising user interfaces through change tracking, annotations, and alternatives. In *CHI*, 2010.

[22] S. Kolaric, R. Woodbury, and H. Erhan. Cambria: a tool for managing multiple design alternatives. In *DIS Companion '14*, 2014.

[23] L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, Naghmi Shireen, and M. Terry. Gem-ni: A system for creating and managing alternatives in generative design. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

[24] J. Burry and D. Holzer. Sharing design space: Remote concurrent shared parametric modeling. 2009.

[25] Robert Aish and Robert Woodbury. Multi-level interaction in parametric design. In Andreas Butz, Brian Fisher, Antonio Krüger, and Patrick Olivier, editors, *Smart Graphics*, pages 151–162, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[26] Verina Cristie and Sam Joyce. Capturing and visualising parametric design flow through interactive web versioning snapshots. 09 2017.

[27] Alexey Zagalsky, Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, and Weiliang Wang. The emergence of github as a collaborative platform for education. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work; Social Computing*, CSCW '15, page 1906–1917, New York, NY, USA, 2015. Association for Computing Machinery.

[28] Tiffany Tseng. Making Make-throughs: Supporting Young Makers Sharing Design Process. page 8.

[29] Tinggui Chen, Lijuan Peng, Xiaohua Yin, Jingtao Rong, Jianjun Yang, and Guodong Cong. Analysis of User Satisfaction with Online Education Platforms in China during the COVID-19 Pandemic. *Healthcare*, 8(3), July 2020.

[30] Ellen Taylor-Powell. *Evaluating collaboratives: Reaching the potential.* Number 8. University of Wisconsin–Extension, 1998.

[31] Michael Terry and Elizabeth D. Mynatt. Recognizing creative needs in user interface design. In *Proceedings of the 4th Conference on Creativity & Cognition*, page 38–44, New York, NY, USA, 2002. Association for Computing Machinery.