3D Visualization of the University of Massachusetts Boston McCormack Building

A thesis presented by

Nandinii S. Yeleswarapu

to the Honors College and the College of Science and Mathematics
in partial fulfillment of the requirements
for the degree of

Bachelors in Science
in the subject of
Computer Science

University of Massachusetts Boston
Boston, Massachusetts

May 2021

Prof. Daniel F. Haehn Nandinii S. Yeleswarapu

3D Visualization of the University of Massachusetts Boston McCormack Building

Abstract

This project aims to develop a working pipeline to be able to transform 2D blueprints of a building to a 3D model which can then be rendered on the web in an interactive yet easily accessible form. We obtain 2D blueprints for a building designed using Computer Aided Design (CAD) software and use architectural software to convert the blueprints to 3D models. We then use Xeokit, a WebGL framework to render the building on the web.

We use the University of Massachusetts Boston 2D blueprints as our data set, successfully applying our transformation pipeline to the blueprints of the McCormack building, creating the first web based, interactive 3D model of the UMass Boston McCormack building.

Contents

List of Figures

To my family, for their endless support.

Acknowledgments

Throughout the journey of this research project and of writing this thesis paper I have received a great deal of support and assistance.

Firstly, I would like to thank my Thesis Advisor, Professor Daniel Haehn and my Thesis Mentor, Loraine Franke for guiding me throughout the project. I would like to thank them for sharing their knowledge in this field, exposing me to new technologies and new approaches towards solving problems, and most of all for their patience through all my ups and downs in this project.

Secondly, I would also like to thank the Honors College at University of Massachusetts Boston, especially Dean Rajini Srikanth for creating such an amazing opportunity for me to be a part of. I absolutely enjoyed being a part of the Honors College and getting to work on a research project that interested me deeply.

Thirdly, I would also like to the Assistant Campus Planner at University of Massachusetts Boston, Simon Myles for providing us with the campus blueprints and for always being very responsive and giving me informative advice. I would also like to thank Lindsay Kay, creator of Xeokit at Xeolabs for his patient assistance with the technicalities of using Xeokit and one of the authors of the Deep Floor Plan Recognition using a Multi-task Network with Room-boundary-Guided Attention ICCV paper, Zhiliang Zeng for his timely and informative help in trying to figure out and understand the required dependencies and workings of the neural network.

Fourthly, I would like to express my gratitude to Lisa Greggo, the Associate Dean of the Honors College, my Honors Advisor, and my role model. I would like to thank her for being a constant pillar of support. She has inspired me everyday for the past 3 years, guiding me through navigating failure, reminding me to celebrate my successes and reminding me of the potential I have to achieve the goals I set my focus on, both through motivation and by setting an amazing example for me to learn from.

I would like to thank Shivam Gupta, a peer I took Professor Haehn's Graphics course with and a dear friend of mine. I am grateful for his assistance through the course in learning and using Autodesk Revit which formed an integral part of the transformation pipeline.

I would also like to express my gratitude towards all my friends and family members who have supported me patiently throughout this journey. They are the pillars of support, encouragement and love behind my success.

1

Introduction

One of my roles working as an Honors College Community Ambassador has been overseeing the College front desk including but not limited to assisting people who stop by the desk with various questions. When the Honors College office used to be located on the second floor of the Campus Center, I would notice that we often had students, Professors and guests stop by to inquire where certain offices and rooms were located. This observation inspired an initiative to look for more efficient ways of communicating to people on campus not only which building and

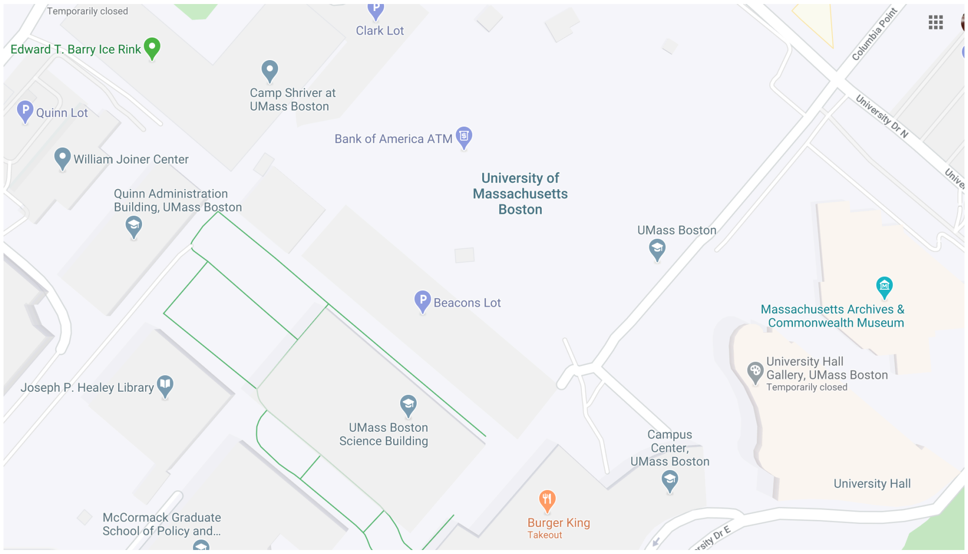

Figure 1.1: Google Maps result for a search of "University of Massachusetts Boston"

floor certain offices were in, but interactive and easily accessible ways to navigate around campus effortlessly. The first step involved looking through existing maps and directories to understand the drawbacks of the existing systems and areas of possible improvements.

1.1 Existing maps and directories of the UMass Boston campus

- **Google Maps**

  Searching "University of Massachusetts Boston" on Google Maps [5] yields an image of the street maps where the university is located with pins positioned where individual buildings and notable areas are, for instance the Campus Center, the parking lots, the Joseph P. Healey library and the Edward T. Barry Ice Rink.

  Google Maps is widely used and therefore requires little to no further instructions to be given to campus members to use the map. As seen in figure 1.1, the map is quite useful when determining driving routes to certain buildings and parking lots but is rather vague and uninformative when trying to navigate to specific offices and areas on campus.

- **2D Campus Map featured on the website**

  The university website features a link to a 'Google Maps' inspired interactive 2D map [7] which places pins on the map where certain departments, colleges and services are located. Hovering over the pins pops up an information box with the name identifying the location, the physical address, a brief description and a link to a website page with more information about the location as seen in figure 1.2.

  The map also allows the user to browse over a list of all the defined pins and to search a specific location's pin under a "Campus Overview" section. The map also supports a "walk" feature which allows the user to view and interact with a live view of the campus buildings and roads using Augmented Reality as seen in figure 1.3. The page also offers a PDF version of the map available for download.

  This map offers more information as compared to the Google Maps version along with interactive features that are easy to learn and use. It also offers more relevant pins to the university such as department and college locations and links to web pages containing more information. Yet, the major drawback associated with Google Maps remains. Navigating to specific offices and rooms within the buildings remains difficult with this version.

- **Building Map**

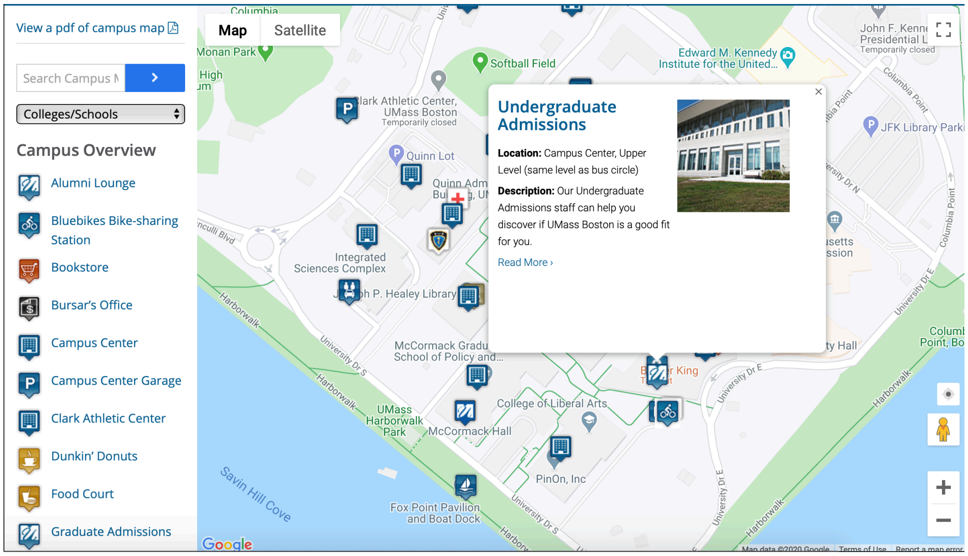

Figure 1.2: Campus Map featured on the university website.

Figure 1.3: Live view of the Campus Center back entrance using the campus map "walk" feature.

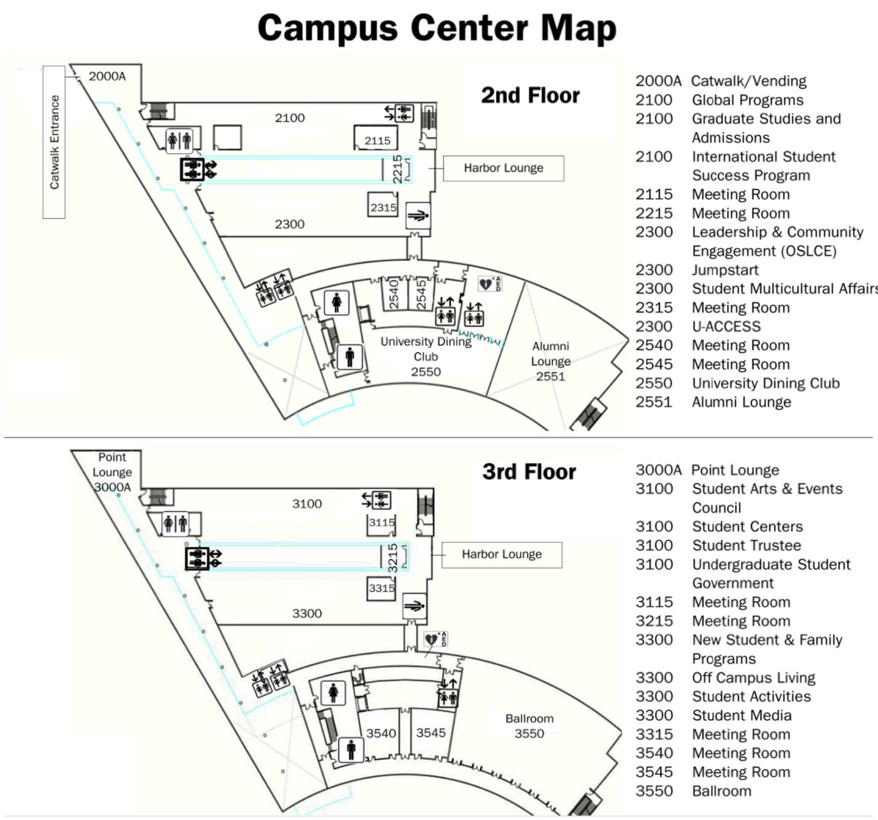

Figure 1.4: A section of the PDF to the Campus Center Building Map

The Campus Center web page [6] offers a link to a PDF document consisted of basic, simplified 2D floor plans of different floors in the Campus Center building and list directory beside the floor plans of the rooms and offices located of particular floors as seen in figure 1.4.

This map provides the additional advantage of introducing an internal view of the associated building over the two maps covered prior. This map specifically aids in locating exact whereabouts of offices with both titles and room numbers used. The drawbacks include: this form of map only exists for one building in the whole campus and remains a 2D oversimplified representation of the actual space which can be confusing to rely on when faced with all the additional aspects in the real building.

1.2 The new 3D model

Putting together the individual benefits from each of the existing maps available and brainstorming ideas to improve on their drawbacks led to the origination of a new model. The new model presents 3D scaled models of all the buildings on campus, along with labeled offices and, an interactive means of navigation around the campus. This new model would be accessible directly on any web browser without needing special downloads and installations. There are many reasons why an interactive campus 3D map as described above is not only an aesthetic improvement to the existing maps but an essential functional addition to the university website:

1) University of Massachusetts Boston is a public research university with students from diverse backgrounds, including Boston residents, out-of-state students and international students from over 136 countries[8]. This means that only a small fraction of students enrolled every year have the opportunity of attending the campus tours. A 3D scaled, accurate and interactive map would allow a greater fraction of incoming students to experience a virtual "campus tour" with as much detail and exposure as the on campus tours. Students would further be able to explore the campus in a flexible manner, allowing them to become more comfortable and feel more welcome come first day of classes.

In fact, this is more important now than ever. The COVID-19 pandemic has reshaped the idea of normal as we know it. In today's time, most campus activities are remote and off-campus. This new model would allow students to experience a sense of connectivity with the university even as they are attending events off-campus. Using AI, and tools and softwares similar to FRAMEVR[4], remote events could be conducted on this 3D campus model allowing attendees to create and utilize avatars to interact during the event.

2) The UMass Boston campus frequently hosts multiple events, conferences and lectures by guest speakers. Thus, we often have visitors who aren't familiar with the campus and find it difficult to make their way to event venues and around. This can be an inconvenience given it creates confusion, the need for an usher and sometimes may result in avoidable time delays. This new model for a campus map aids in easing the process of hosting events and inviting visitors allowing them to make their way to the venue in a timely manner and be able to explore the campus afterwards.

3) This map would also allow university administrators to designate classes and office spaces more efficiently and comfortably.

4) The map can also be further developed to provide administrators with a 'one-stop' with tools and features to access blueprints to all buildings and modify the 3D models to aid in restructuring and redesigning buildings, significantly aiding future construction planning.

The initial vision for this project was to apply the new 3D model vision to all the university buildings thus becoming the first 3D web based interactive view of the whole UMass Boston campus to provide our community with a way to navigate the campus accessibly. Unfortunately, in the initial stages of research we realized that this was a huge task with multiple obstacles to tackle along the way. Thus, we narrowed our project scope to being able to formulate an efficient and successful work flow to facilitate the transformation of a large industrial scale building from 2D blueprints to 3D web rendered interactive 3D models with the goal of applying the new 3D model to one of the UMass Boston buildings, the McCormack Building.

2

Related Work

Accomplishing the 3D Visualization of the McCormack building could be split into two major tasks: converting 2D blueprints of the building into 3D models and deploying the 3D model to be accessed easily on any internet browser directly from the university website.

2.1 Converting 2D blueprints to 3D models

At the beginning of the project, we predicted that tackling the conversion of 2D blueprints to accurate and scaled 3D models would be the toughest obstacle in this project to cross. As the project progressed we could conclude that indeed it would be. We studied two routes of trying to achieve this conversion:

- Manually, using architectural modeling software such as SketchUp and Revit.
- Automatically, using multitask Deep learning neural networks.

2.1.1 Manually converting 2D blueprints to 3D models

The softwares involved in manually converting 2D blueprints to 3D models are readily available, offer several templates to work with immediately and often can be learned within a couple of days. Such software often also provide sophisticated tools which allow intricate and aesthetic designing of buildings using building materials, light and water fixtures, stairways, parking spaces, and a variety of furniture options. Despite the numerous advantages of architectural modeling softwares, they posed several challenges when we were trying to use them to convert the building map.

The softwares are mainly designed and built for architectural purposes and thus proved to be more efficient when the 2D floor plans were directly drawn in the software itself or when provided as an AutoCAD file. The blueprint files we obtained from the university were raster images in JPEG and PDF formats. This meant that first we would need to trace over the walls and sections in the floor plans and then generate the 3D models. For simple floor plans with limited rooms and geometrically shaped structures like those of apartments, it would be straightforward to trace over the walls in a short period of time. But, for floor plans of buildings such as those of the McCormack building which has complex layouts, this would not be a time efficient solution.

Among the several softwares available online, we specifically worked with SketchUp 2019[11] and Autodesk Revit 2019[2] when exploring various ways to convert the floor plans.

- **General Overview**

  SketchUp belongs to the Computer Aided Design (CAD) software family while Revit belongs to the Building Information Modeling (BIM) software family.

  CAD "refers to the use of computers systems to aid in creating designs, allowing engineers to create higher-quality drawings for products or parts more quickly"[10].

  While, BIM "is software that applies CAD concepts to designing buildings, creating models that include not just the physical but also the intrinsic properties of a building"[10].

- **Price**

  SketchUp offers a free online browser-based version with almost all the features included in the latest desktop version available and 10MB free online storage. The online browser-based version has the added advantage of not being constrained to hardware requirements and can be accessed from a broad range of devices.

  SketchUp offers an educational version for a discounted subscription fee charged per year after a 30 day free trial period once the desktop version is installed and signed into.

Revit costs a significant amount more than SketchUp per year but the educational version is offered through Autodesk for a year for free.

As a student or faculty member Revit is definitely more economical.

- **Software and Hardware requirements**

  SketchUp is more versatile and supports both Windows and Mac Operating Systems but cannot be run on BootCamp, VMWare, Parallels or similar environments. Revit only supports Windows but can be run on BootCamp, VMWare or such similar environments.

  SketchUp recommends a device with 8GB RAM and 700 MB of availbale hard-disk space while Revit recommends a device with 8GB RAM and 35 GB of available hard-disk space.

  Personally owning a Mac, SketchUp was much more convenient to install and set-up and supported smooth operation. I had to install Revit via first installing Windows on VirtualBox which greatly limited space availability and led to the performance of my laptop and the software in effect drastically reducing.

- **User Interface**

  SketchUp has a simpler UI making it easier to learn the tools and features available as seen in figure 2.1. The tools available are also much easier to operate thus making learning how to use the software generally faster.

  Revit offers more technical tools, thus having a more complex UI which displays the diversified toolbar of features as seen in figure 2.2. This makes

Figure 2.1: Cropped view of the toolbars available in SketchUp.

learning how to use Revit much more complicated and takes longer to familiarize oneself with the features.

- **Ease of conversion between 2D to 3D**

  For architectural buildings it is much easier to convert 2D images to 3D models using Revit compared to SketchUp.

  Consider for instance walls; in SketchUp lines are simply recognized as lines thus we would have to trace the inside of all rooms and sections as well as the outside ensuring that the traces close (form closed loops). The space between the closed loops is then recognized as a wall which you could use a tool to raise to scale. In Revit, a line is automatically recognized as a wall

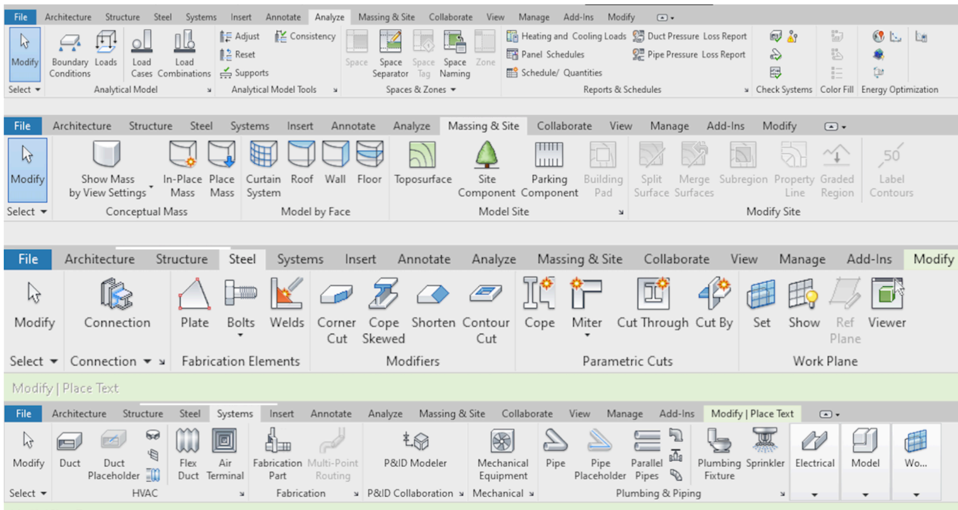

Figure 2.2: Cropped view of the toolbars available in Revit.

and a 3D model is simultaneously generated as we continue to work on the 2D model, we only need to specify the height of the wall when drawing the line.

These differences are illustrated in figure 2.3 and 2.4. Figure 2.5 shows the simultaneous creation of the 3D model in Revit from the 2D blueprint associated.

- **Conclusion**

  Both softwares are incredibly powerful and efficient with features and tools that make architectural designing convenient but SketchUp is more suited more creating detailed commercial designs focused more on interior designing whereas Revit is more suited more large scale industrial projects focused more on building design such as the dataset we have.

  Figures 2.6 and 2.7 show the final 3D model created using SketchUp and the 3D model rendered using Revit.

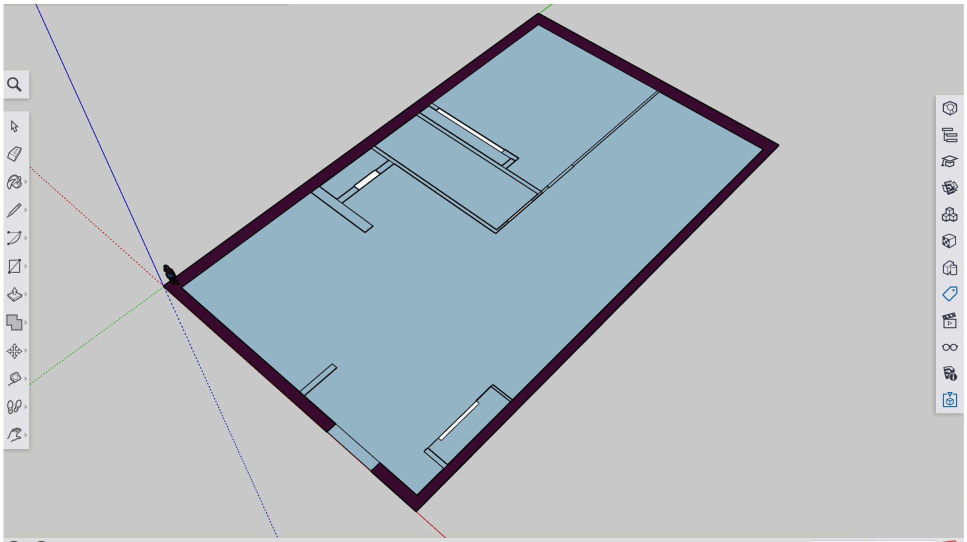

Figure 2.3: 2D blueprint for a sample floor plan created in SketchUp.

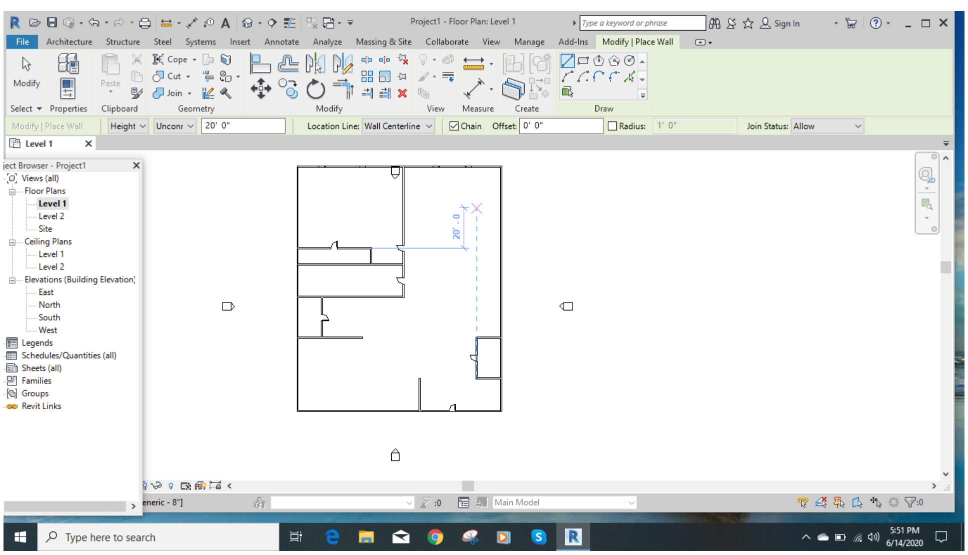

Figure 2.4: 2D blueprint for a sample floor plan created in Revit.

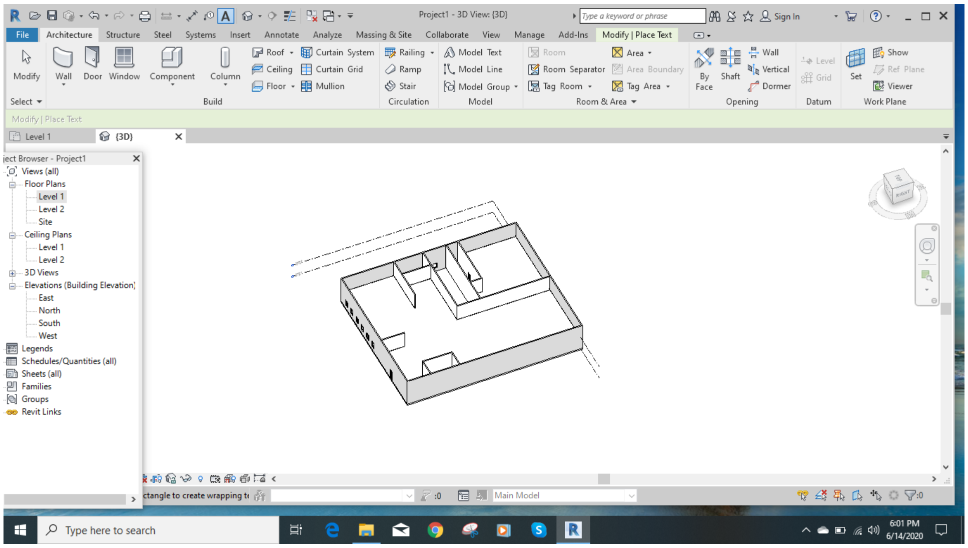

Figure 2.5: 3D model corresponding to the 2D blueprint created in Revit.

2.1.2 Automatically converting 2D blueprints to 3D models

We explored two methods of automatically converting 2D blueprints to 3D models using Deep Learning Neural Networks:

- **Planner 5D**

  Planner 5D[9] is a web based commercial tool that accepts as input a raster image of a floor plan or blueprint, internally runs the image through a neural network model and then outputs a 3D model corresponding to the blueprint.

  The time taken to generate the 3D model depends on the size of the blueprint, in our case, the tool took about 5 to 6 hours to generate the 3D model.

  In terms of generating the 3D model even for such a large plan, this tool's efficiency and accuracy is commendable. This tool required minimal effort

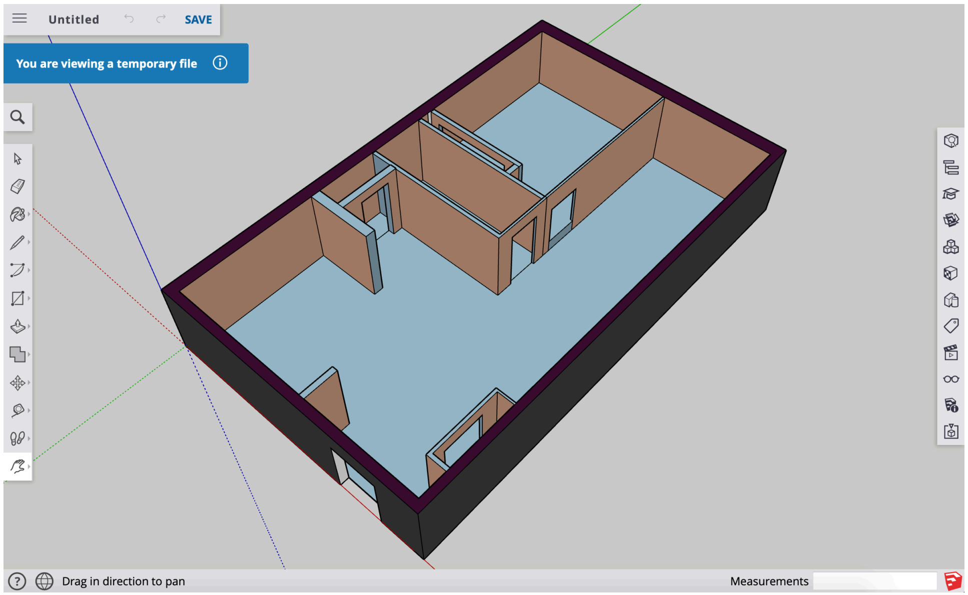

Figure 2.6: 3D model created using SketchUp

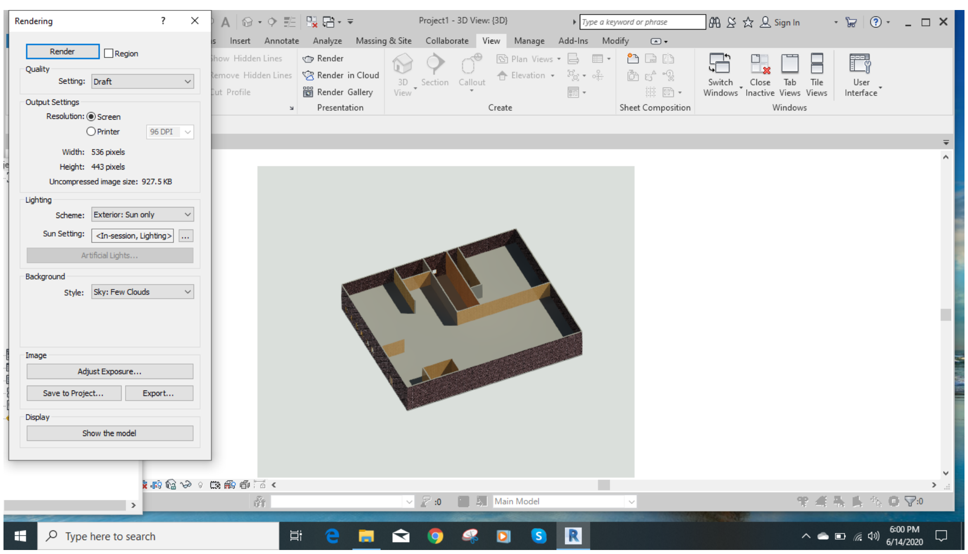

Figure 2.7: 3D model rendered using Revit

from our end for the conversion and provided the most accurate results in terms of wall thickness and placement, and capturing all the rooms and windows.

The most notable drawbacks using Planner5D include: the tool requires a paid subscription which costs 9.99 USD/month for the individual plan which includes only 5 renders and 5 textures per month; the tool only allows rendering the 3D model within their platform and cannot be exported as other formats restricting the ability to deploy the maps on other platforms or websites; the tool also doesn't allow interaction between various projects and each AI recognized 3D model is saved into a new model thus restricting combining the 3D models for different floors we have into one project.

- **DeepFloorTransformation**

We looked into existing work from various papers and found the ICCV paper by Zhiliang Zeng, Xianzhi Li, Ying Kin Yu and Chi-Wing Fu[13] the most promising relative to our needs. Their work involved using a deep multi-task neural network to recognize elements in floor plan layouts. The model they designed performs two tasks: the first task identifies room-boundary elements in floor plans such as walls, doors and windows to identify and separate rooms in the layout; the second task predicts the types of each of the rooms identified such as dining, kitchen, bathroom etc.

The most notable drawbacks of applying this model to our dataset include: the model only analyses the floor plans and identifies various elements, thus we would still need to implement a way to isolate boundary elements and apply a model to automatically raise these elements; the code provided uses

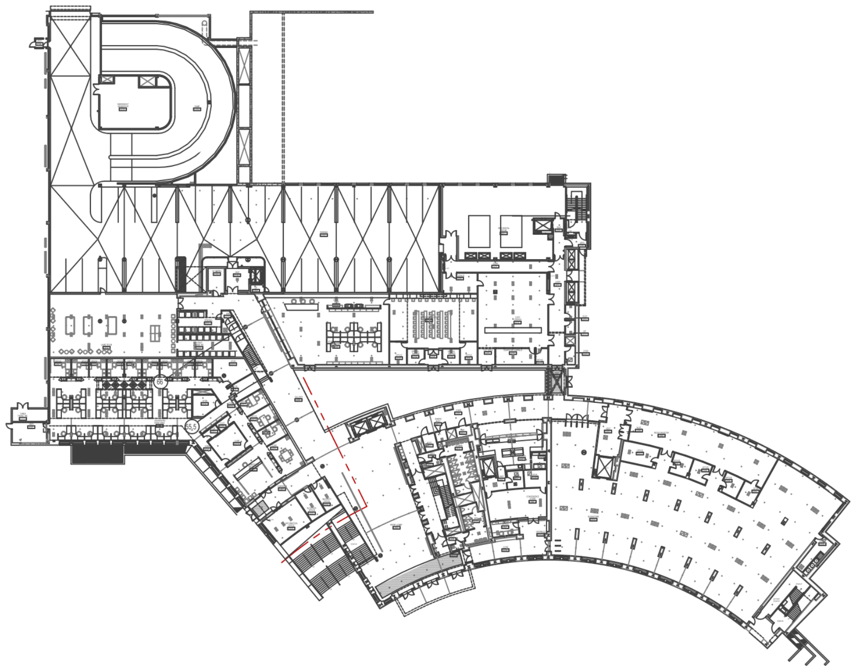

Figure 2.8: 2D blueprint of the Campus Center UL floor as a PNG file provided as input to Planner5D

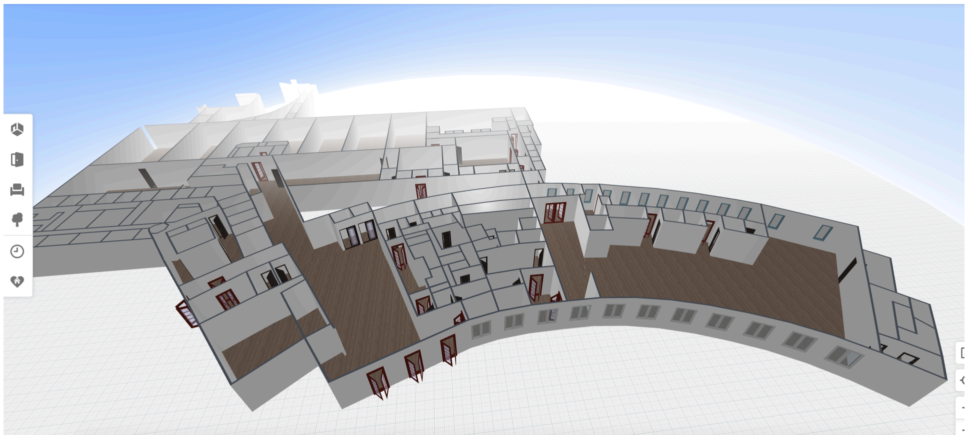

Figure 2.9: 3D model for the Campus Center UL floor generated by Planner5D

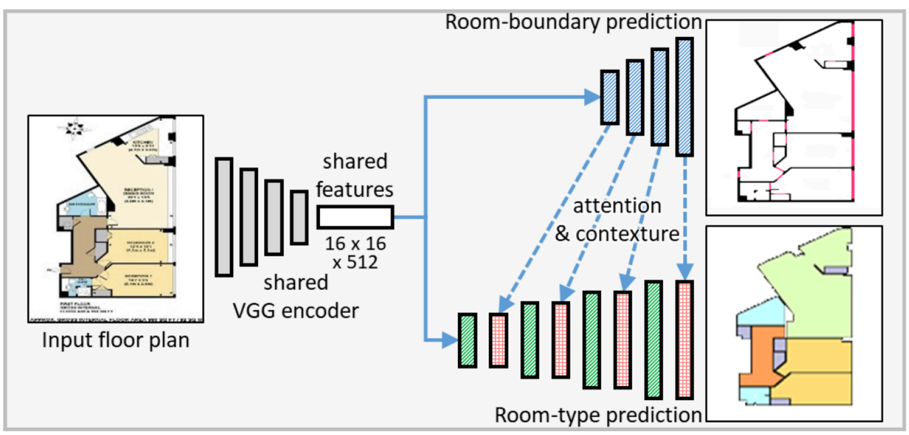

Figure 2.10: An overview of the network architecture used by the ICCV paper by Zheng et al. This figure was taken from the ICCV paper[13]

older versions of TensorFlow and other dependencies which we had to identify thus required creating a special Conda environment with the appropriate dependencies to be able to run the model; and the model worked well on smaller scale apartment style blueprints in simple shape layouts but wasn't able to accurately identify elements and rooms with our data set, thus would need to us to specifically modify and train the model for our data set.

Given our data set was quite restricted in size, we wouldn't be able to train the model as efficiently as we would need. 2.11 demonstrates the results of applying the model to a fragment of the Healey Library 10th floor blueprint containing limited rooms and a symmetrical geometric layout. 2.12 demonstrates applying the model to the McCormack Building 1st floor blueprint containing a greater number of rooms but still retaining the geometric layout. 2.13 demonstrates applying the model to the Campus Center 3rd floor blueprint containing a significantly greater number of rooms and complex shaped rooms. We can see that as the blueprints get more and more complex, the room-boundary identification becomes less and less efficient notable by the color coding in the resulting images.

2.2 Making the 3D model of the campus available online on a web browser

The second half of our project would entail finding a way to make the 3D models obtained from the first half available easily and accessibly through the University website. For the purposes of this project we aimed to make available the models through GitHub pages using a basic placeholder website we create.

During my undergraduate journey at UMass Boston, I took a course with Professor

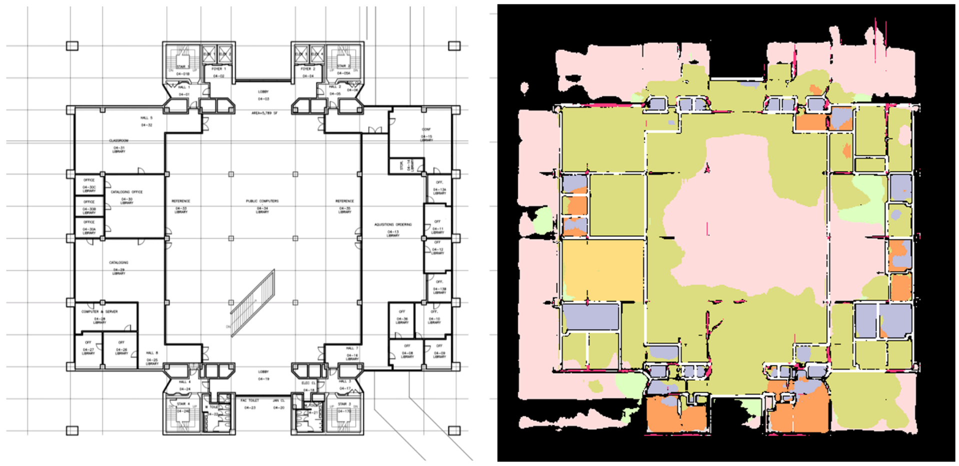

Figure 2.11: Applying the ICCV paper model to a simple campus floor plan layout

Daniel Haehn titled Graphics where I learned about the scope and potential of WebGL and WebGL frameworks such as ThreeJs[3]. Professor Haehn also introduced me to the visualization WebGL framework Xeokit by Xeolabs[12].

Xeokit is an open-source 3D web SDK created by Lindsay Kay at Xeolabs specifically created to handle rendering large industrial buildings on the web. Xeokit provides a specialized API to handle a large fraction of the functions that are needed to render large scale industrial infrastructure on the web.

On the Xeokit website we were able to run a demo illustrating the UI and interaction provided by models rendered with Xeokit, and the assembling of various floors to form a building. We ran the demo on the OTC Conference Center Building provided as a sample on the Xeokit website. 2.14 demonstrates how Xeokit renders the whole of the OTC Conference Center building and 2.15 demonstrates how Xeokit renders the 6th floor of the building.

The major drawbacks of using Xeokit included: we would have to study and learn how to use the Xeokit API to render our models efficiently; and Xeokit renders

Figure 2.12: Applying the ICCV paper model to an intermediate campus floor plan layout

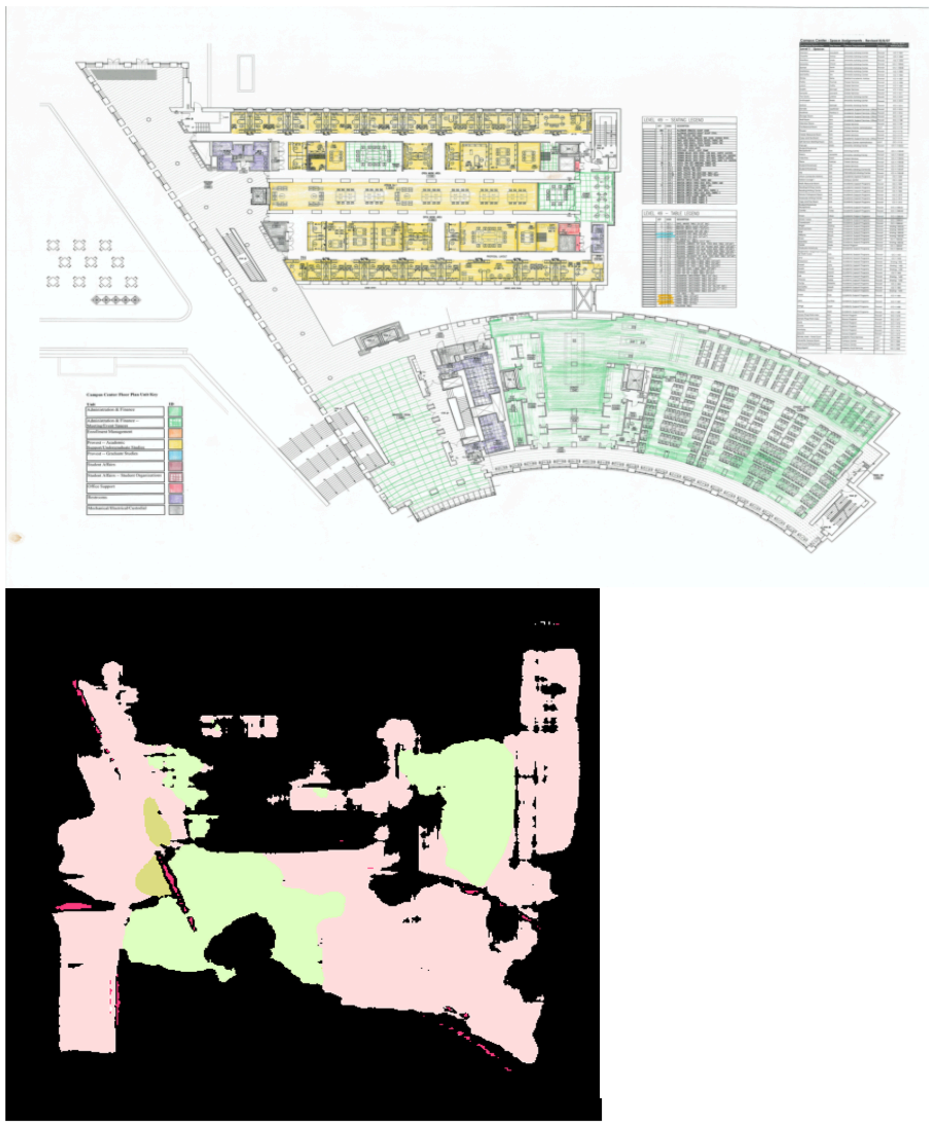

Figure 2.13: Applying the ICCV paper model to a complete campus floor plan layout

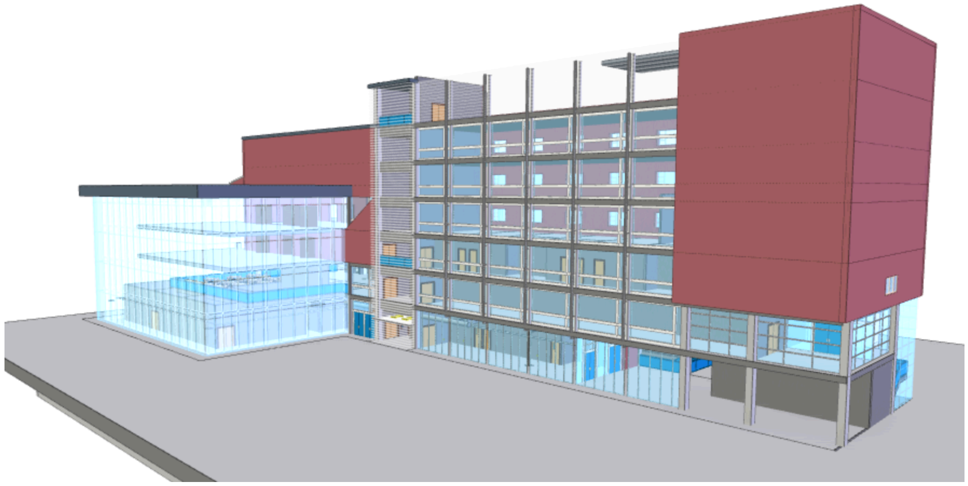

Figure 2.14: The whole OTC Conference Center building as rendered by Xeokit

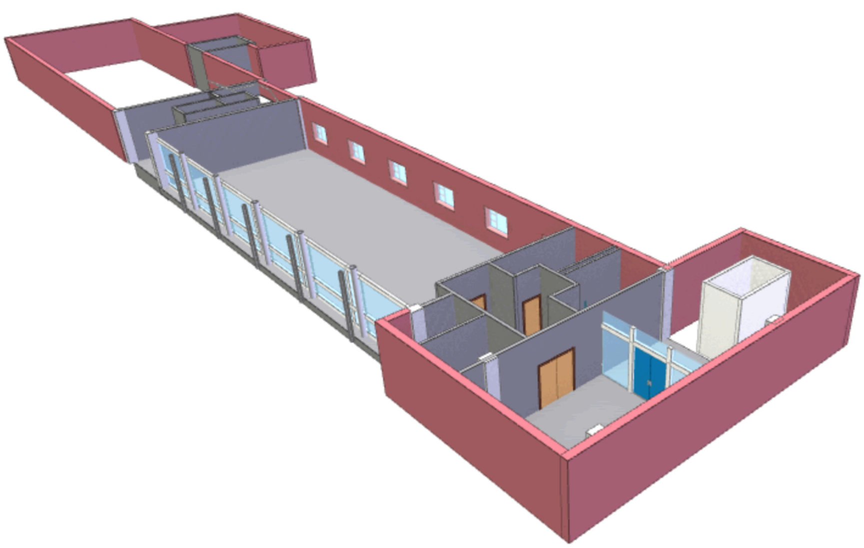

Figure 2.15: The OTC Conference Center 6th floor as rendered by Xeokit

floorplans provided in their proprietary format XKT thus we would need to find a method to convert our Revit files to this format.

3

Our Method

After keenly researching the various methods of generating and rendering the 3D models we finalized our pipeline and goals, choosing to use Autodesk Revit as our tool for converting 2D blueprints to 3D models, and Xeokit to render the models in the web. We aimed to establish this pipeline as illustrated in 3.1 and successfully apply it to the McCormack building of UMass Boston.

We fortunately were able to obtain the McCormack building 2D blueprints for individual floors in CAD (.dwg) format from the university's Assistant Campus Plan-

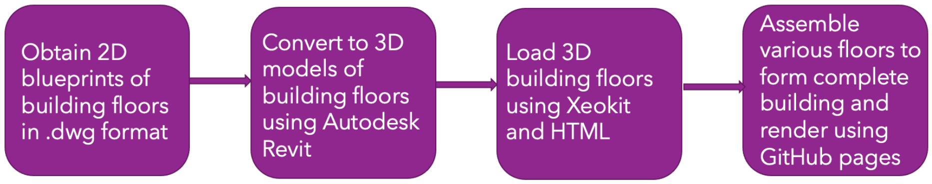

Figure 3.1: Visual representation of the predicted work flow to complete this project

ner, Simon Myles with the help of Professor Daniel Haehn, Loraine Franke and Honors College Associate Dean, Lisa Greggo. The floor plans in this format would make utilizing Revit much more convenient as the wall elements would readily be identifiable in the software.

3.0.1 Implementation

Firstly, we used the software Revit to create 3D models for each building floor. To do this, we first imported the .dwg format 2D blueprints into Revit and selected the wall elements to place walls on the floor plan. We then exported the floors structural data and metadata as .ifc (IFC) format files from Revit. This step of the thesis project was completed as part of the Fall 2020 CS460:Graphics course final project. For this step, a course peer, Shivam Gupta and I worked on converting the 2D blueprints to 3D using Revit together.

Secondly, we forked the Xeokit repository[12] from GitHub to the desktop and downloaded into the same folder the tools we would need to facilitate needed file conversations.

Thirdly, we used Command Line Tools[1] provided by Xeolabs to convert the .ifc (IFC) file to .dae (COLLADA) file and then to a .gltf (GLTF) file and then to a .xkt (XKT) file. We also extracted material information and metadata from the IFC file

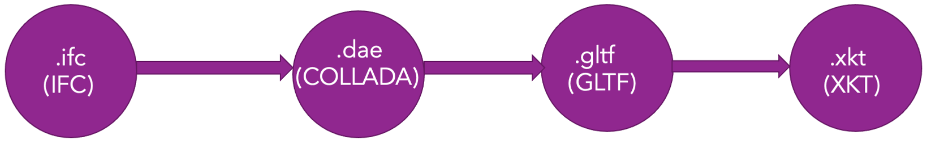

Figure 3.2: Visual representation of process of converting .ifc files to .xkt files

into a JSON format file. A summary of the file format conversions is illustrated in 3.2

Given there were several files that we would need to convert using multiple program runs we wrote a python script to include all the commands instead. The commands that this script encompasses and runs are listed below:

```
./Ifcconvert --use-element-guids [target_filename].ifc
    [target_filename].dae --exclude=entities IfcOpeningElement
./COLLADA2GLTF-v2/COLLADA2GLTF-bin -i [target_filename].dae -o
    [target_filename].gltf
./xeokit-gltf-to-xkt/gltf2xkt.js -s [target_filename].gltf -o
    [target_filename].xkt
./xeokit-metadata-osx-x64/xeokit-metadata [target_filename].ifc
    [target_filename].json
```

3.3 demonstrates the CAD file we obtained from the university vs 3.4 which demonstrates the IFC file we obtained from Revit for the McCormack building's first floor.

XKT is Xeokits' native file format which can then be loaded using the framework. We used example source files provided by Xeokit to load XKT files to load the first floor into the web. We used the Xeokit APIs to modify the code to load several floors of the building into the same scene. We also modified the code to adjust the

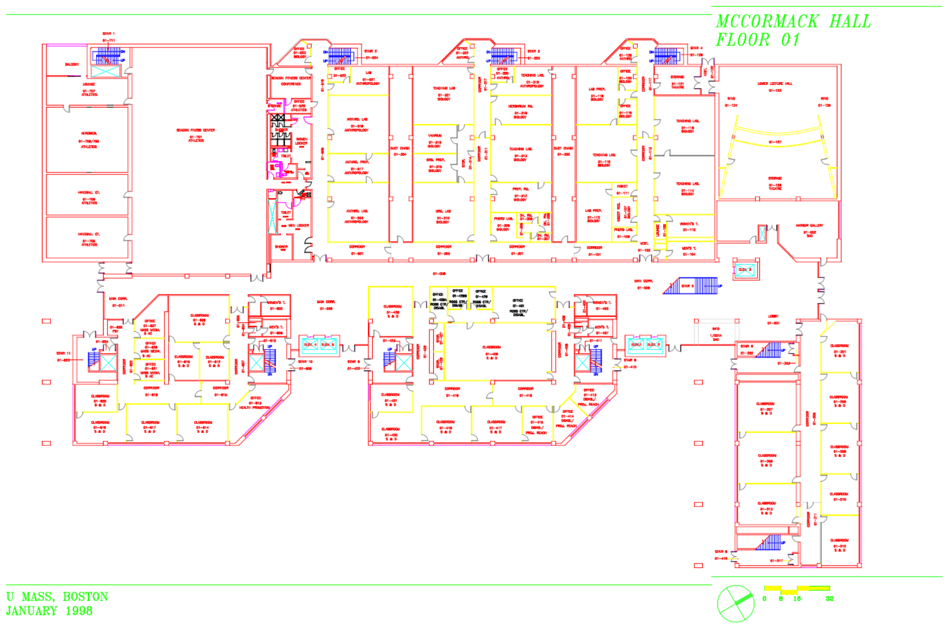

Figure 3.3: CAD file we obtained from the university for McCormack floor 1

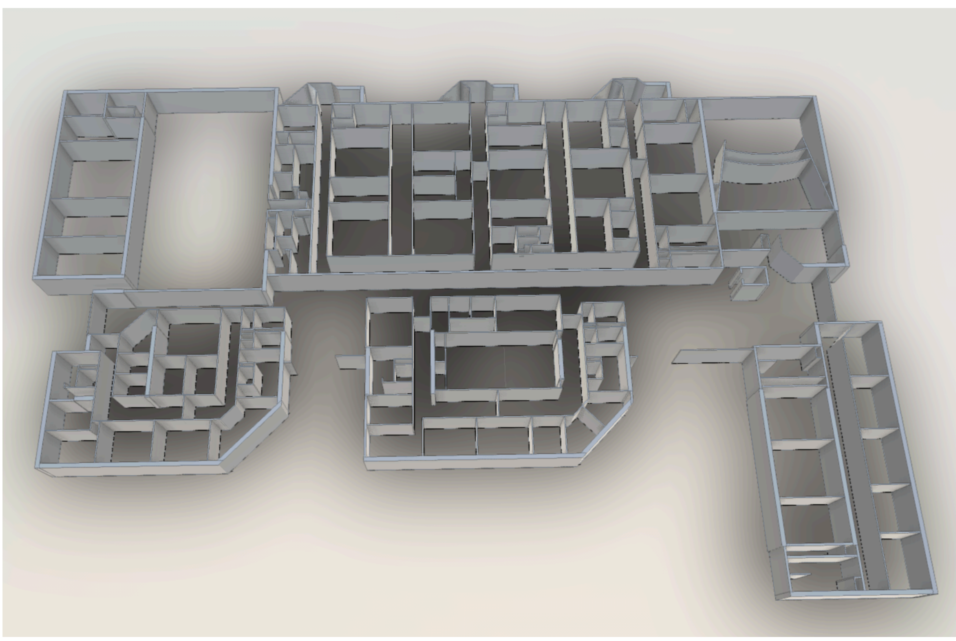

Figure 3.4: IFC file we obtained from Revit for McCormack floor 1

placement positions of the floors such that they accurately stacked on top of each other without overlapping each other.

The following lines of code illustrate how we can use Xeokits' xktLoader to easily load the structural data and materials metadata for the first floor in the scene. The position attribute for the various floors need to be adjusted appropraitely to ensure the floors load above each other and lined at the boundaries to form a complete building model.

```
const mck1 = xktLoader.load({
    id: "myModel1",
    src: "./models/xkt/Mck/Mck1.xkt",
    metaModelSrc: "./metaModels/Mck/Mck1.json",
    position: [0, 0, 0],
    edges: true
});
```

Having successfully constructed a working pipeline to convert blueprints to 3D models and applying it to the McCormack building, we completed our scope for this project. We would like to make a note that the building as of now lacks roof and floor elements which we purposefully left out so that we could demonstrate clearly the division of rooms within the buildings for each floor.

4

Results

At completion of the project, we were able to render the McCormack building in the web browser successfully allowing full interaction with the model and the ability to zoom in and out of the rooms on any floor in the model.

We created a basic website using HTML, CSS and Javascript to load the building model. The website includes several features to allow and ease user interaction with the project and the model:

- **Position Transformation**

Using the track pad or mouse allows the user to adjust the placement of the model along the x-axis and the y-axis.

- **Scale Transformation**

  Using the track pad or mouse also allows the user to zoom into and out of the model. The user can also zoom into certain rooms allowing a view into the interiors of various rooms.

- **Model Rotation**

  Using the track pad or mouse also allows the user to rotate the model around allowing a clearer view of the model from all angles.

  A labelled navigation cube as shown in 4.1 in the right bottom corner of the page makes rotating the model easier and also provides constant knowledge of the direction of the model being viewed at any given point in time.

- Highlighted Entities:

  Hovering the cursor over any of the elements in the model highlights the entity yellow as shown in 4.2 to allow the user to track view movements easily.

- The 2D to 3D building transformation pipeline:

  A labelled flowchart at the top of the page as shown in 4.3 provides a basic summary of the stages involved in generating and rendering the model. Hovering over any of the flowchart elements launches a tooltip as shown in 4.4 providing more information regarding the transformation stage.

- Project Information:

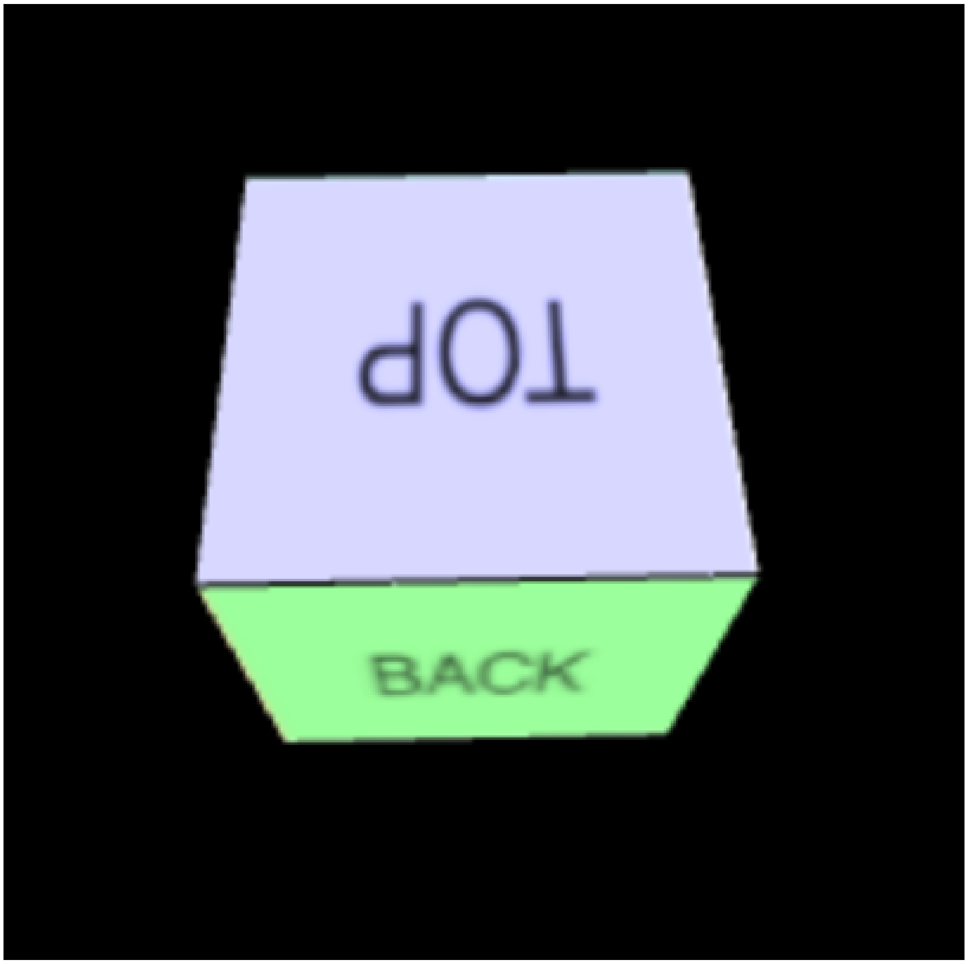

Figure 4.1: Navigation cube allowing a smooth rotation of the model

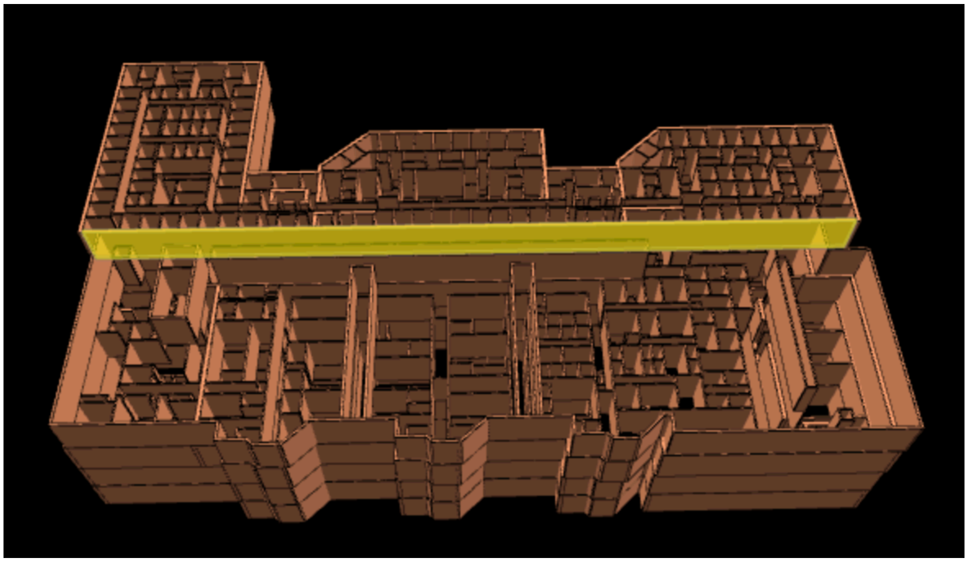

Figure 4.2: Highlighted entity as it is hovered over

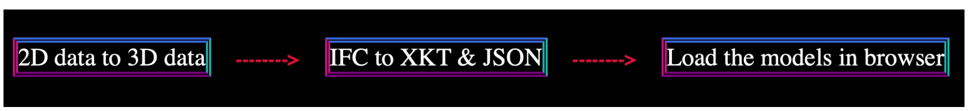

Figure 4.3: The 2D to 3D building transformation pipeline

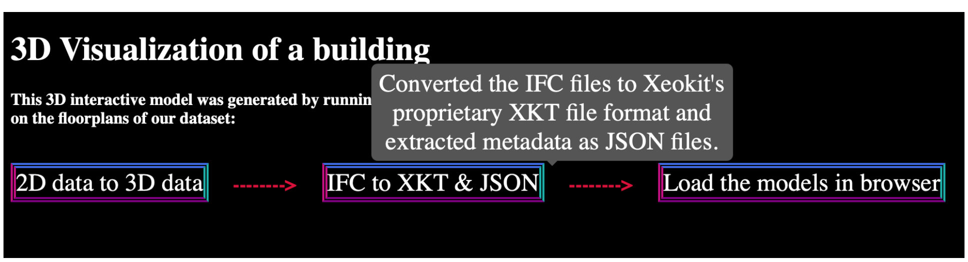

Figure 4.4: Tooltip with more information on the transformation stages

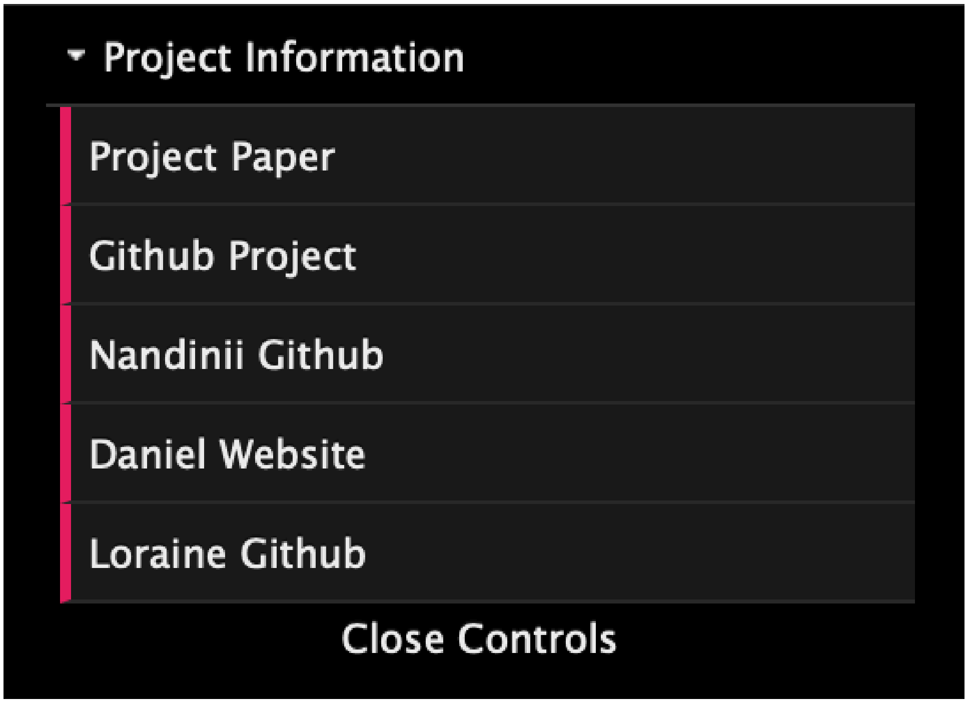

Figure 4.5: Links to websites with more information about the project

A GUI designed using ThreeJs in the right top corner of the page as shown in 4.5 provides links allowing the user to navigate to the project's code base as a repository on Github, to this paper within the Github repository, to my Github page with several other projects I've worked on, a link to my thesis advisor Professor Haehn's website and a link to my thesis mentor Loraine Franke's Github.

4.1 The Github Repository

This section explored the various files and folders included as part of the project code base explaining how this project results can be applied to other blueprints and buildings.

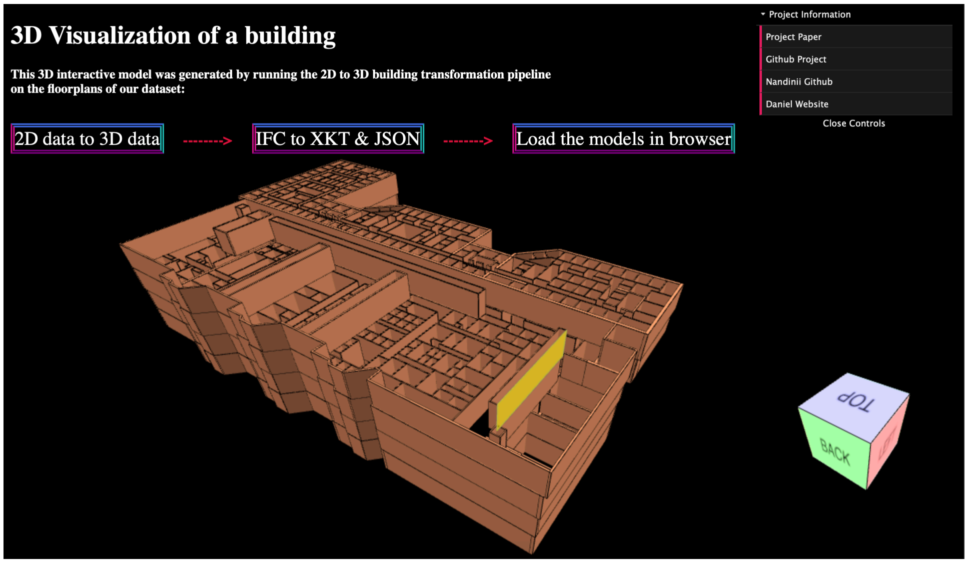

Figure 4.6: The complete website demonstrating the 3D model of the McCormack building

4.1.1 The McCormackBuildingFiles folder

This folder contains all the IFC, COLLADA, GLTF, XKT and JSON files generated and used in this project for the first four floors of the McCormack building rendered in the model.

4.1.2 The LiteMckViewable folder

This folder contains the required XKT and JSON files, plugins, viewer files, and the HTML file needed to render the full website demonstrating the 3D model of the McCormack building as shown in 4.6.

This model can be run using Github Pages by following the link: `https://nandiniiys.github.io/LiteMckViewable/index.html`

4.1.3 The LiteViewableGen folder

After analyzing the stages of the 2D to 3D building transformation and the HTML file rendering the model, we noted that quite a lot of the stages are constant and can be combined into one or more scripts that automate the whole process. This would allow users to apply our 2D to 3D building transformation pipeline to own building blueprints. This folder contains all the tools needed for the file format conversions, required plugins and viewer files, and the scripts as described below:

- **ifcTOxkt.py**

  As discussed in chapter 3, this script takes the name of the IFC file to be converted and runs the script on this file, generating the intermediate COLLADA and GLTF files as well the desired XKT building data and JSON metadata files. The original IFC files needs to be in the LiteViewableGen directory for the conversion tools to have access to the file. The script thus also cleans up the directory after the various files are generated by moving the building files to the contained models directory.

- **htmlTemplate.txt**

  This text file contains base HTML, CSS and Javascript code that loads the various features of the website as discussed earlier leaving out only the code that loads the various floors.

- **htmlGen.py**

  Given the name of the building (requires all the IFC files and thus all the XKT and JSON files to be named consistently as <buildingname>.<floorNum-no preceding 0s>) and the number of floors load combines the data in the html-

Template.txt file with appropraite Javascript code to load all the models producing a concise HTML file named index.html

- **modelGen.py**

  This is the main python script run by the user. This script takes the building name and number of floors and runs all the scripts mentioned above. After a complete run, the user would need to edit the index.html file to adjust the positioning of the various floors according to building dimensions. The model is now ready to be rendered.

5

Discussion

5.0.1 Challenges

We faced several challenges during the course of this project, some of which we were able to conquer and some of which are still in works of progress.

- **Challenge 1:** The Revit software we needed only works on the Windows OS and unfortunately I had access only to a Macbook. Eventually, we were able to request remote access to the UMass Boston Computer Science Lab PCs which are Windows PCs to be able to use Revit.

- **Challenge 2:** Learning how to use remote access and especially being able to manage the files we needed between the two systems was tedious.
- **Challenge 3:** Using Revit in itself to create the 3D models was incredibly time consuming since we had to pick walls over all the blueprints and given the size of the building this was particularly strenuous when trying to make sure we didn't double trace any overlapping walls.
- **Challenge 4:** We were successfully able to add geometrical elements to the scene and map textures to them but given our model isn't of uniform geometry we weren't able to find a suitable function in Xeokit to map realistic textures to the building walls. This is an obstacle that is still a work in progress.
- **Challenge 5:** We wanted to change the background of the scene to resemble the view from the campus one would see if they were to view the building as is rendered. We attempted to use Google Earth to capture scenery that we could replace the background image with. But, we noticed that doing so only obscured the building. We also noted that to be able to make the scene realistic we would have to create a sky box with real time images of the campus surroundings. Unfortunately, this wasn't something we were able to understand how to implement within the time frame we had and thus is an obstacle that is also still a work in progress.

5.0.2 Application and Future work

At the beginning of this project, having been naive to the involvement and depth of designing and implementing a research project on such a large scale, my goals and aims were rather ambitious. Through my journey of learning, testing, failing, fixing and finally completing this project, I had to rethink the scope of the project to a goal that could be achieved within the time frame choosing to prioritize efficiency and quality of the goal rather than the depth. That does mean that there are several improvements and additions that I hope and intend to pursue beyond this publication.

- **The UMass Boston Campus**

  The most glaring next step for this project would be to apply the pipeline we've established to the rest of the campus buildings to achieve the initial vision we had for the first interactive, web-based representation of the university.

- **Unify university blueprints**

  At the time we obtained the blueprints from the university, we recognized that they were all available only either as raster images or as CAD files. All the various floor blueprints for all the buildings are held in different files, some of which with written or typed annotations that are backdated. The completion of applying this pipeline to all the buildings would allow the university to retain blueprints of the whole campus in one unified and accessible location.

- **Update university blueprints**

  Over the years, there have been several changes to the university buildings. On a scale scale, certain walls have been knocked down, rooms moved about and on a large scale, buildings knocked down during the current wave of construction at the university. This new model would allow the university to update the floorplans easily and conveniently, ensuring that the map of the university available to students and members of the community is always accurate and up to date.

- **Maintain university privacy**

  In our initial conversation with Simon Myles, he made us aware that we would have to keep in mind what sections of the campus we label and make visible on the model as there are certain security and privacy sensitive areas that we would have to be mindful of. As the model expands to include the whole campus, we would need to find a way to ensure that these areas are indeed hidden from guest users and only accessible to those faculty and administrative members of the university who have the clearance. This would include designing and establishing a back-end sign-in and authentication feature along which a database to hold clearance levels data for the various sensitive areas on campus.

- **Individual safety**

  Although with the opening of the student dormitories in 2018, UMass Boston still remains a majorly commute campus meaning that the campus usually has a low population density, especially during the later hours of the day. In the recent year, with the ongoing pandemic resulting in majority of student activities and classes being conducted remotely, the population density has further decreased. This often leads to a sense of safety risk and concern for individuals on campus. This new model could be further enhanced to provide a safety feature for individuals to send the UMass Boston Public Safety Department an exact and premise location down to the room they are in, should they find themselves concerned for their safety.

6 Conclusion

We believe that the 2D to 3D model transformation pipeline we were able to design and implement during the course of this project has significant potential in the realm of visualizing 3D buildings. In our research to be able to realize the 3D visualization of the UMass Boston McCormack building we came across several research papers and tools that facilitated only a fraction of this whole process but none that provided a concrete, working and easy to learn methodology. Thus, this project serves two benefits to the greater community: it provides an adaptable, easy to learn way of creating web rendered interactive 3D models from basic 2D blueprints which can be adapted to be used in multiple different projects; and it serves as a starting point to further build on and one day accomplish the visualization of the entire UMass Boston campus as was the initial vision.

A

Appendix

A.1 Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

(a) **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

(b) **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

(c) **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

(d) **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

(e) **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

(f) **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

(g) **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

(h) **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

(i) **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

(j) **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

(k) **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

(a) **License grant.**

  (1) Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

    (A) reproduce and Share the Licensed Material, in whole or in part; and

    (B) produce, reproduce, and Share Adapted Material.

  (2) Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

  (3) Term. The term of this Public License is specified in Section 6(a).

  (4) Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats

whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

(5) Downstream recipients.

(A) *Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.*

(B) *No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.*

(6) No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

(b) **Other rights.**

(1) Moral rights, such as the right of integrity, are not licensed under this

Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

(2) Patent and trademark rights are not licensed under this Public License.

(3) To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

(a) **Attribution.**

(1) If You Share the Licensed Material (including in modified form), You must:

(A) retain the following if it is supplied by the Licensor with the Licensed Material:

(i) identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

(ii) a copyright notice;
(iii) a notice that refers to this Public License;
(iv) a notice that refers to the disclaimer of warranties;
(v) a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

(B) indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

(C) indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

(2) You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

(3) If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

(4) If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

(a) for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

(b) if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

(c) You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

(a) **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**

(b) **To the extent possible, in no event will the Licensor be liable to You on**

any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

(c) The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

(a) This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

(b) Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

   (i) automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

   (ii) upon express reinstatement by the Licensor.

   For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

(c) For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

(d) Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

(a) The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

(b) Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

(a) For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

(b) To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

(c) No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

(d) Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Bibliography

[1] Viewing an ifc model with xeokit. https://www.notion.so/Viewing-an-IFC-Model-with-xeokit-c373e48bc4094ff5b6e5c5700ff580ee, 2012. Online; accessed 19th May 2021.

[2] Autodesk. Inc. Autodesk revit. *URL: https://www.autodesk.com/products/revit/overview*. Online; accessed 19th May 2021.

[3] Ricardo Cabello et al. Three.js. *URL: https://github. com/mrdoob/three.js*, 2010. Online; accessed 18th May 2021.

[4] FRAMEVR. https://framevr.io/. Online; accessed 19th May 2021.

[5] Google Maps. https://www.google.com/maps/place/UniversityofMassachusettsBoston/@42.3146224,-71.0397134,17z/data=!3m1!4b1!4m5!3m4!1s0x89e37afb848c801f:0x8b627c28a174fb5b!8m2!3d42.3146185!4d-71.0375194, 2020. Online; accessed 19th May 2021.

[6] University of Massachusetts Boston. https://www.umb.edu/campus_center/about/building_map, . Online; accessed 19th May 2021.

[7] University of Massachusetts Boston. https://www.umb.edu/map, . Online; accessed 19th May 2021.

[8] University of Massachusetts Boston. https://www.umb.edu/the_university/facts, . Online; accessed 19th May 2021.

[9] Planner5D,UAB. Planner5d,uab. *URL: https://planner5d.com*. Online; accessed 18th May 2021.

[10] Dan Taylor. https://blog.capterra.com/bim-vs-cad-whats-the-difference/, 2017. Online; accessed 19th May 2021.

[11] Trimble. Sketchup. *URL: https://www.sketchup.com*. Online; accessed 18th May 2021.

[12] Xeolabs. Xeokit-sdk github repository. *URL: https://github.com/xeokit/xeokit-sdk*, 2020. Online; accessed 19th May 2021.

[13] Zhiliang Zeng, Xianzhi Li, Ying Kin Yu, and Chi-Wing Fu. Deep floor plan recognition using a multi-task network with room-boundary-guided attention. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. doi: 10.1109/iccv.2019.00919. Online; accessed 3rd April 2021.