3D Visualization of a building

Nandinii Yeleswarapu and Shivam Gupta
{n.yeleswarapu001},{shivam.gupta001}@umb.edu
University of Massachusetts Boston

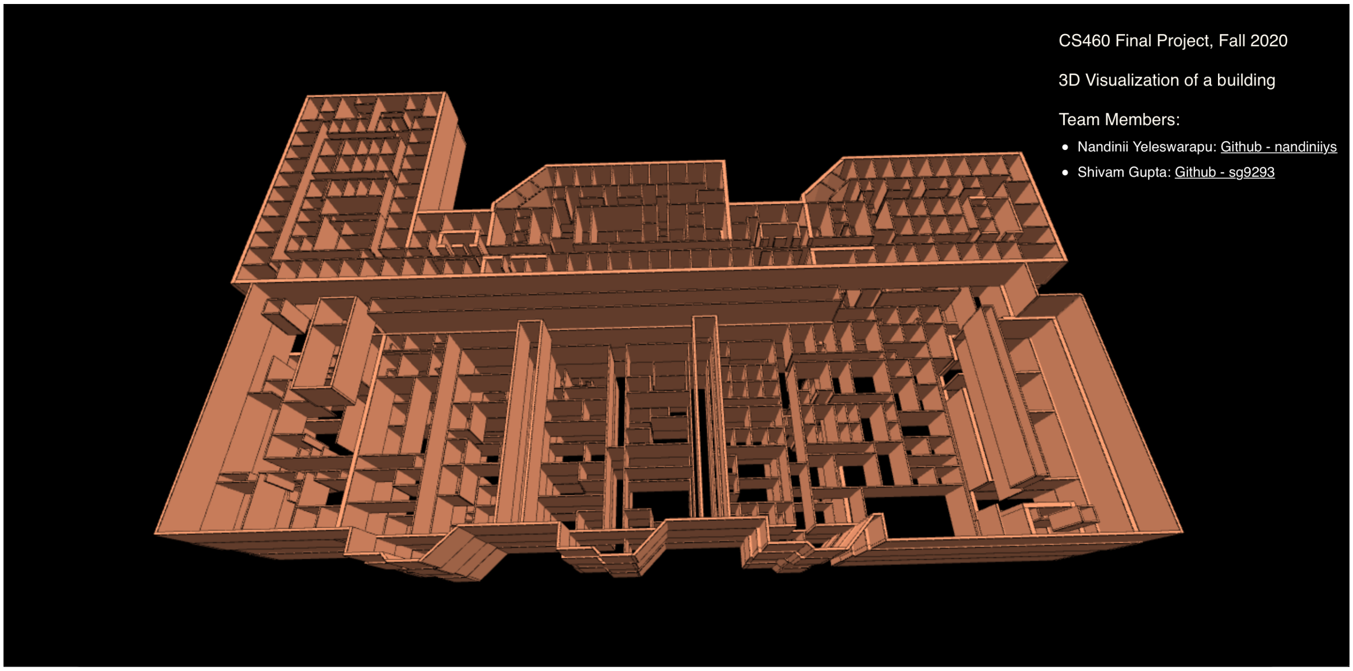

Figure 1: A Safari browser view of the 3D rendering of the UMass Boston McCormack building.

ABSTRACT

This project aims to develop a working pipeline to be able to convert a 2D blueprint of a building to a 3D model which can then be rendered on the web in an interactive yet easily accessible format. 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.

KEYWORDS

WebGL, Visualization, 3D models, BIM, CAD

ACM Reference Format:
Nandinii Yeleswarapu and Shivam Gupta. 2020. 3D Visualization of a building. In *CS460: Computer Graphics at UMass Boston, Fall 2020*. Boston, MA, USA, 4 pages. https://CS460.org

ACM ISBN 1337.
https://CS460.org

1 INTRODUCTION

The inspiration for this project was rooted in our personal experiences on campus as freshmen. We both recall how daunting and overwhelming the campus would look to us and how often we got lost in our first semester at UMass Boston.
Our initial vision was to create a 3D web based interactive view of the UMass Boston campus to provide our community with a way to navigate the campus accessibly.
The need for such a one stop handy campus representation has become even more evident during the current pandemic. With the campus essentially being shut down to visitation and all departments having to navigate around work remotely, such a representation opens up a wide range of possibilities for departments to continue to engage students on campus, this being particularly useful for organizing campus tours for potential students.
In our 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 conversation of a building from 2D blueprints to 3D web rendered interactive 3D models.

2 RELATED WORK

With the increasing popularity and growth of the technology industry we have seen a demand in the industry to push computer programs to new limits. Limits that provide programmers with so much potential to develop powerful and impactful programs. One such field has been using WebGL to produce realistic, interactive and easily web accessible graphical representations of almost anything one can imagine of. Thus, when conducting research on the ways in which we could approach our goal we came across several tools and frameworks, each more powerful than the last, that it made narrowing down one to choose on a project in itself. Our project can be understood in two part: converting 2D blueprints to 3D models; and rendering the 3D models on the web. For the first part of out project we looked at various architectural softwares such as Planner5D [1] and SketchUp [6] before deciding to use Autodesks' Revit [5] due to its wide range of features, exportable file formats and free student version which proved economical. For the second half of the project, although we were more comfortable with WebGl frameworks learned in class such as XTK [3] and Three.js [2], we chose to use Xeokit by Xeolabs [4]. 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 we would need to render our building on the web thus making it our choice.

3 METHOD

For this project, we chose to attempt the visualization of the UMass Boston McCormack building. We obtained the McCormack building 2D blueprints for individual floors in .dwg (CAD) format from the universitys' Assistant Campus Planner, Simon Myles with the help of Professor Daniel Haehn and Honors College Associate Dean, Lisa Greggo. We used Autodesk Revit to create 3D models for the individual floors. We then used Command Line Tools provided by Xeolabs to convert the file format for the 3D floor models to a format supported by Xeokit and then used Xeokit to load the models to the web.

3.1 Implementation

Firstly, we used the software Revit to create 3D models for each building floor. To do this, we first import the .dwg format 2D blueprints we have into Revit and place walls on the floor plan that Revit can identify. We then export the floors structural data and metadata as a .ifc (IFC) format file from Revit.
Secondly, we forked the Xeokit repository from github to our desktops and downloaded the tools we would need to facilitate needed file conversations.
Thirdly, we used Command Line Tools provided by Xeolabs [7] to convert the .ifc (IFC) file to .dae (COLLADA) file and then to a .gltf (GLTF) file to a .xkt (XKT) file. We also extract material information and metadata from the IFC file in JSON format. 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.
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 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.

```
const mck1 = xktLoader.load({
    id: "myModel",
    src: "./models/xkt/Mck/Mck01.xkt",
    metaModelSrc: "./metaModels/Mck/Mck01.json",
    position: [0, 0, 0],
    edges: true
});
```

Having successfully constructed a working pipeline to convert blueprints to 3D models 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 show clearly the division of rooms within the buildings for each floor.
Our future proposed vision is to edit the source code to make the visualization look aesthetically pleasing by changing the background colors and adding images to make the background look as realistic as possible.

3.2 Milestones

This section explains how we structured our project goals to be able to complete this project during the proposed time line and before the deadline.

3.2.1 Milestone 1. We convened over scheduled zoom meetings to discuss the Revit software and share knowledge and understanding so as to learn and become comfortable with using the software.

3.2.2 Milestone 2. We convened over zoom meetings to go through example source codes provided by Xeokit and provided APIs to understand how we could use Xeokit to load our models on the web.

3.2.3 Milestone 3. We divided amongst ourselves the floors of the McCormack building to convert into 3D model.

3.2.4 Milestone 4. We divided amongst ourselves the floors of the McCormack building to convert into 3D model thus completing the 3D modeling well in time.

3.2.5 Milestone 5. We created a python script which allowed us to easily convert all the .ifc files we had to .xkt files to render using Xeokit.

3.2.6 Milestone 6. We loaded all the models for the various floors into one scene and positioned the floors in a manner that rendered them as a complete fluid building.

3.3 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 we both have Mac books. 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 trace walls over all the blueprints and given the size of the campus 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.

4 USAGE

Since this project makes use of an open source SDK by Xeolabs and the code provided, we have uploaded all the files and folders needed to be able to use this pipeline. Given the sheer size of the repository needed to perform the file format conversions and the features of the tool we weren't able to upload to GitHub all the files used in this project. Instead we uploaded the files that we edited and created to out GitHub repositories and added this section to explain how to go about visualizing our final project using Xeokit.

- Step1: Fork the following repository and clone it to your computer, https://github.com/xeokit/xeokit-sdk
- Step2: Add the Mck.html file to the examples folder directly.
- Step3: Add the Mck*.xkt files to the examples/models/xkt folder.
- Step4: Add the Mck*.json files to the examples/metadata/xkt folder.
- Step5: Launch local server and run the following command: http://127.0.0.1:8000/examples/Mck.html

5 RESULTS

At completion of our project, we were able to render the McCormack building in a web browser successfully as in image 3 allowing full interaction and the ability to zoom in and out of the rooms on any floor as in image 4. We were also able to ensure that the various floors which were loaded as different models were aligned and uniform thus rendering the building realistically. The alignment of the floors can be seen in image 2.

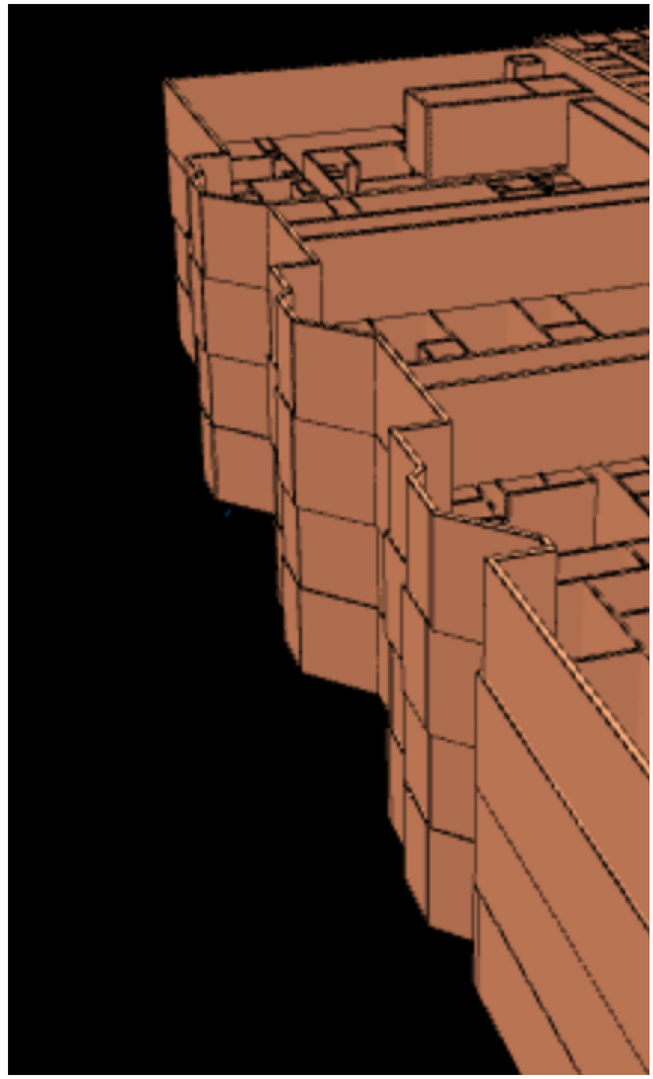

Figure 2: A section view of the building to illustrate floor alignment.

6 CONCLUSIONS

We believe that the working pipeline we were able to achieve 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.

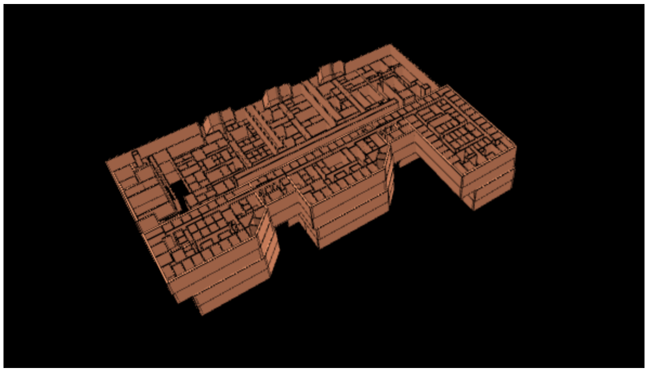

Figure 3: A focused view of the rendered building.

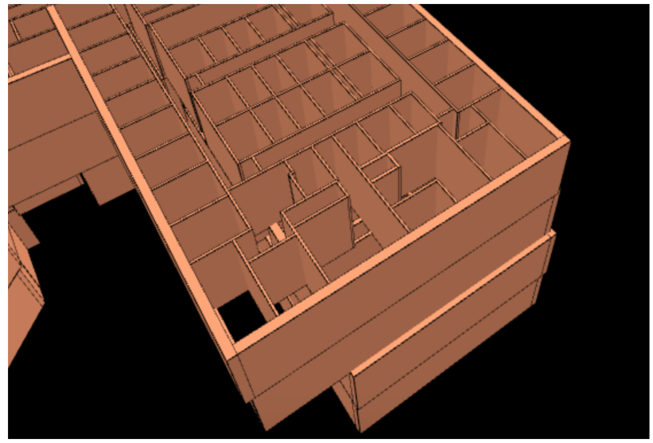

Figure 4: A zoomed in view of the rooms on the top floor.

REFERENCES

[1] Autodesk. Inc. [n.d.]. *Autodesk Revit.* https://planner5d.com

[2] Ricardo Cabello et al. 2010. Three.js. *URL: https://github. com/mrdoob/three.js* (2010).

[3] Daniel Haehn, Nicolas Rannou, Banu Ahtam, P. Ellen Grant, and Rudolph Pienaar. 2012. Neuroimaging in the Browser using the X Toolkit. *Frontiers in Neuroinformatics* (2012).

[4] Lindsay Kay. 2020. Xeokit. *URL: https://github.com/xeokit/xeokit-sdk* (2020).

[5] Planner5D,UAB. [n.d.]. *Planner5D,UAB.* https://www.autodesk.com/products/revit/overview

[6] Trimble. [n.d.]. *SketchUp.* https://www.sketchup.com

[7] Xeolabs. 2020. Creating Files for Offline BIM. *URL: https://github.com/xeokit/xeokit-sdk/wiki/Creating-Files-for-Offline-BIM* (2020).