Engineering Art: Interactive Installation for Engineering Education

Final Report

Sdmay24-17

Client – Dr. Rachel Shannon

Faculty Advisor – Dr. Rachel Shannon

Karandeep Sandhu – Electrical Engineering

Alexandra Whitmer – Software Engineering

Austin Krekula – Cyber Security Engineering

Seyedehbahareh Hashemimovahed – Computer Engineering

Grant DeWaay – Software Engineering / Interdisciplinary Design

Executive Summary

Development Standards & Practices Used

- ∉ Hardware Practices:
    - o Meticulous design documentation
    - o Comply with industry standards and regulations
    - o Compatibility and integration testing
    - o Prototyping and iteration

- ∉ Software Practice:
    - o Version control and code maintenance
    - o Code review
    - o Security and vulnerability management
    - o Documentation and User Guide

- ∉ Art Practice:
    - o Artistic style and expression
    - o Creative development and conceptualization
    - o Fusion of art and technology
    - o Narrative and emotional expression

- ∉ IEEE Standard:
    - o IEEE P2784 : The Guide for the Technology Evaluation of Smart City Projects.
    - o IEEE 1680 Series: Standards for Environmental Assessment of Electronic Products. Design principles for sustainable and eco-friendly smart city elements in the art piece.
    - o IEEE 802.3: Ethernet
    - o IEEE 802.11: Wireless LAN (Wi-Fi)

- ∉ ISO Standards:

- ISO 9075-13:2016 Information technology -- Database languages -- SQL -- Part 13: SQL Routines and types using the Java TM programming language (SQL/JRT)
- ISO 14882:2020 Information technology - Programming Languages - C++
- ISO 23270:2018 Information technology - Programming languages - C
- ISO/IEC 27001: Provides a systematic approach to managing sensitive company information so that it remains secure. It encompasses people, processes, and IT systems, assessing and mitigating the risks to which they are exposed

Summary of Requirements

- Research and explore the 14 National Academy of Engineering Grand Challenges for Engineering
- Choose several of the challenges to explore further
- Narrow down our options to just one challenge to tackle
- Brainstorm solutions to the redefined problem
- Create a prototype

Applicable Courses from Iowa State University Curriculum

COM S 227, 228

SE 309, 421

DES 150x, 230

List of figures & tables

Figures

Figure 1: Integrated Design and Technical Development Process Map

Tables

1 Introduction

1.1 PROBLEM STATEMENT

In general, the public lacks awareness of the role of engineers and the problems that they solve. The resulting disconnect makes it difficult to know what engineers do and why their work is so important in everyone's day to day lives.

The problem we need to solve for our design project is to educate the public about the contemporary challenges that engineers tackle. More specifically, we were tasked to select one item from the **National Academy of Engineering's Grand Challenges for Engineering.** To do this, we must convey this in an artistic way that demonstrates the use of electrical, computer, software, and/or cybersecurity engineering in our final project.

2 Project Overview

2.1 REQUIREMENTS & CONSTRAINTS

The following are functional and non-functional requirements that include both those stated explicitly by our advisor Dr. Rachel Shannon in addition to requirements intrinsic to creating a project for our intended audience, that being primary school children ranging from 5 to 16 years old.

Functional Requirements

1. Project is derived from one of the 14 **National Acadamy of Engineering's Grand Challenges for Engineering**
2. The installation is educational, and teaches kids about the engineering challenge in the classroom
3. The design should be interactive in a captivating way
4. The design must demonstrate the use of electrical, computer, software and/or cybersecurity.
5. Allows future senior design teams work on problem

Listed nonfunctional requirements are extracted from the chosen Grand Challenge, Improving Urban Architecture, and what would be required to convey the importance of engineers tackling this challenge. Each non-functional requirement has a corresponding number representing the functional requirement that they relate to.

Non-functional Requirements

- Teaches kids about Improving Urban Infrastructure (1)
- Portable design that doesn't take up much space, allows easy transfer of the design in and out of classrooms (2)
- Utilization of technologies that are well documented and easy to learn (5).
- Usage of conductive fabric to enable touch control, giving the hands-on experience that engineering requires (3)
- Questions about infrastructure in conjunction with fun and engaging imagery to provide a fun experience (2)

2.2 ENGINEERING STANDARDS

The standards our project will have are going to come from national, international, and volunteer industry standards organizations such as IEEE, IETF, ANSI. The applicable standards will cover ethernet and wireless communication, computer language use, database use, technological safety, and environmental sustainability.

2.2.1 IEEE Standards:

IEEE 802.15.4: Used for low-rate wireless personal area networks (commonly used in IoT applications).

IEEE P2784: The Guide for the Technology Evaluation of Smart City Projects.

IEEE 1680 Series: Standards for Environmental Assessment of Electronic Products. Design principles for sustainable and eco-friendly smart city elements in the art piece.

Adhering to IEEE standards ensures reliable, compatible, and sustainable technological solutions for urban infrastructure.

2.2.2 ANSI Standards:

ISO 9075-13:2016 Information technology -- Database languages -- SQL -- Part 13: SQL Routines and types using the Java TM programming language (SQL/JRT)

ISO 14882:2020 Information technology - Programming Languages - C++

ISO 23270:2018 Information technology - Programming languages - C

ISO/IEC 27001: Provides a systematic approach to managing sensitive company information so that it remains secure. It encompasses people, processes, and IT systems, assessing and mitigating the risks to which they are exposed.

IEC 60950: Safety of information technology equipment: Pertains to the safety of IT equipment. The standard defines the general requirements for the safety of such equipment, including protection from electrical hazards, thermal risks, mechanical hazards, and more.

IEC 61000: Electromagnetic compatibility (EMC): A series of standards addressing electromagnetic compatibility (EMC). It encompasses both electromagnetic interference (EMI) and susceptibility to interference.

By following ANSI standards we guarantee the quality, safety, and efficiency of urban infrastructure products and services.

2.2.3 IETF STANDARDS

IPv6 (RFC 8200): With the increasing number of devices in smart cities, IPv6 offers a vast address space that can accommodate this growth, making it essential for future-proof smart city deployments.

6LoWPAN (RFC 4944, RFC 6282, and more): This standard describes how to efficiently use IPv6 over low-power wireless personal area networks, which is crucial for battery-operated IoT devices prevalent in smart cities.

RPL (RFC 6550): The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is crucial for establishing routes in low-power and lossy networks, like those in IoT settings.

The general population will benefit from the results of our project since this project is intended to be a display. They will use it by essentially living in these improved urban infrastructure. We plan on sharing this project with the student body of Iowa State so they can learn how urban infrastructure can be redesigned and redeveloped. We will also be able to showcase how this improved infrastructure can be used on the global level. This will also be used in K-12 outreach for the National Iowa Space Grant Consortium. Users of all ages will be able to interact with this project effectively.

3 Project Plan
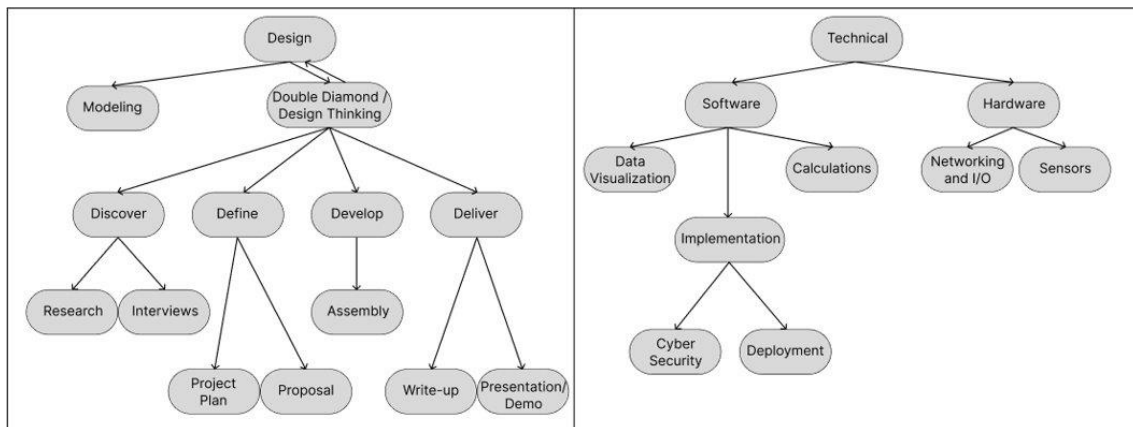
3.1 TASK DECOMPOSITION

Figure 1: Integrated Design and Technical Development Process Map

- Design - Planning out the requirements that are derived from the project proposal
  - Modeling - Includes any artistic elements that enhance the final deliverable
  - Double Diamond / Design Thinking

- ■ Discover - Performing research on the topic we are given so we can make informed decisions regarding how we can build a solution to the problem statement
  - ● Research - Reading text sources that relate to the genre of problem
  - ● Interviews - Talking with experts on campus to learn more about what they would like to see out of our project
- ■ Define
  - ● Project Plan - We now contribute our findings from the Discover phase to culminate a plan that meets the needs and requirements of both the project and the client
  - ● Proposal - Verify with the client that the team's findings and project plan align with their interests and intent.
- ■ Develop
  - ● Assembly - Follow through with our project plan and verify that we are meeting expectations that were set by our teammates, and our client
- ■ Deliver
  - ● Write-up - Write thorough documentation that explains all the steps of the process necessary to operate our project
  - ● Demonstrate - Showcase our project to the client, using technical language that the audience understands
- ● Technical
  - ○ Software - components that visualizes data from the hardware and calculations made on the backend
    - ■ Data visualization - Show data that is expressive and informative that captures the attention of the target audience
    - ■ Implementation
      - ● Cybersecurity - Cybersecurity that ensures the project will stay live without any major interruptions and low maintenance
      - ● Deployment- Deployment that is easy to configure, with simple teardown and setup, even for inexperienced users.
    - ■ Calculations - Calculations that are essential to demonstrate the impacts of designed urban development

- ○ Hardware - components that capture the user input and enables the installation to be interactive for people of all ages
  - ■ Networking- Networking which communicates the state of the system, whether that be user interaction or other state changes, in a timely manner so the software can perform the tasks needed with the lowest latency
  - ■ Sensors - Sensors that are accurate and are tested and measured to gauge the correct amount of force or interaction before certain data is sent to the software.

	A			B			Sculpture	Cool Circuits
			Magazine	Statistics				
			Newspaper article	Research paper	Kids book	Toys		
Sims / City Skylines	Minimalist abstract city	Blocks / Legos	A	B Books / Text/ Noninterractive	C 3 Dimensional model	Website	Urban street public service advertisement	
	FB Electrical mat, put legos on it, shows up virtually	Make it look/ behave like a motherboard	D Moularization	Educating people on the role of engineers of improving urban infrastructure	E Interactive display	Kiosk	E	
			F	G	H Movies/shows			
						Short film	Long film	imagery.
				G			H	

Figure 2: Creative Communication and Media Matrix

3.2 PROJECT MANAGEMENT/TRACKING PROCEDURES

We are adopting waterfall management.

At the outset of our project, we referred to the "14 Grand Challenges for engineers" and selected "restore and improve urban infrastructure" as our primary focus. This decision marked the first phase of our project. In alignment with the characteristics of the Waterfall model, once this phase was set, we proceeded to the next step. Thus, our subsequent task was to delve into the specific issues associated with urban infrastructure.

After in-depth discussions, we identified six primary research areas:

-Power Group
-Energy sustainability
-Cybersecurity
-Civil
-Business
-Urban design

As part of the Waterfall model approach, we believe that a comprehensive review of the work from a preceding phase is vital before embarking on the next. This ensures that we don't overlook any crucial information or essential details.

Consequently, in line with the linear progression of the Waterfall model, the objective of our next phase is to interview experts within our campus who have notable research and insights in the aforementioned areas. Through their professional feedback and opinions, we hope to refine our research directions and strategies, laying a solid foundation for the subsequent development and implementation phases.

We will use Discord for quick communication and for short voice calls when there is an action that needs fast input from multiple members of our team. Figma will be used for visual collaboration in addition to idea generation. The instructor-provided GitLab repository will be utilized for version control of our software components with Git. Additionally, we will be keeping track of milestones and task progress using GitLab's built in Issues and Tasks tracker.

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

For the Spring semester iteration of this project, we wanted to hone in on the development of interactive technologies that enhance the interactive experience for our audience. In particular, we

want to ensure that our sensing technology is consistent and reliable. Additionally, we want the android app to receive the correct data from our serial connection.

Our milestone for the capacitive touch is to reach 95% accuracy when detecting what we call "touch events". A touch event is when our capacitive touch configuration detects that the conductive fabric is being touched. We will say that there is a "miss" in a touch event when the surface does not recognize within 5 seconds that there is a hand touching the fabric. If there is touch event detected but there is nothing touching the surface, that is also considered a miss.

3.4 PROJECT TIMELINE/SCHEDULE
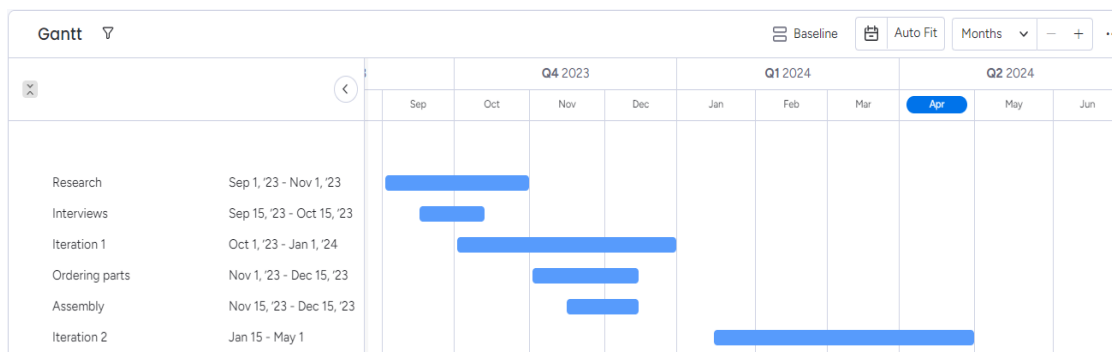
Figure 3: Project Timeline Gantt Chart

After Iteration 1, we had the opportunity to work on our prototype and come up with different ideas to implement leading to the iteration 2. Currently, our final prototype is done during the iteration 2 which is what we are presenting to the industry panel. Using the principles that we acquired from our research, interviews, and ideation, our prototype will reflect some of, although not fully realized ideas that we found.

3.5 RISKS AND RISK MANAGEMENT/MITIGATION (ALLY – IN PROGRESS)

During the early development stage, the risks we faced were low to medium impact. Problems that might affect the long-term success of our project are generally those that impede planning, clarity of vision, coordination, and execution of tasks.

Such risks are:

- Miscommunication
  - Different vision of project outcome
  - Different understanding of requirements
- Disagreement
  - Different design goals

- o Different ideas about implementation
- Misdirection
  - o Planning towards a dead-end
    - ■ Unusable hardware
    - ■ Ill designed software
    - ■ Not seeing solutions
    - ■ Accidental intellectual property theft

These risks are inherent in all team efforts. Our mitigation strategy is built into the waterfall and agile design management methodology. We will be using regular directed meetings and status updates integrated into each sprint to bring up misunderstandings or other issues as early as possible.

The nature of these risks is that they will always be present and no risk management control will be able to completely eliminate them, but the Agile project methodology allows the reassessment of risks and risk mitigation with each sprint, therefore we will be able to regularly reassess these issues.

During the later stage of development, we needed to consider more physical risks that had a medium to high impact. These risks would directly affect the safety of the users and the safety of the team during development.

Such risks are:

- Electrical shock
  - o The user may experience minor electrical shock when touching the display or if there are any exposed wires
- Burning
  - o Soldering is required to complete the project, and teammates are at risk of being burned if appropriate precautions are not taken
- Splintering
  - o Working with the wood could cause physical injury to the teammate
  - o Playing with the display could injure the user if this risk is not proactively accounted for by sanding the display and covering with a protective coat

All of these risks can be accounted for and mitigated by taking the proper safety precautions and using the equipment/technology for its intended purposes.

3.6 Personnel Effort Requirements

Member Name	Task	Hours Spent
Everyone	Research - Everyone is to research areas of our project. • Everyone needs to help with idea generation for iteration 2 that focuses on one of the engineering challenges "Restore and improve urban infrastructure". • Add the idea on Figma board. • Conduct additional research on those topics of interest. • Conduct research on said topic.	• 5-8 hours spent reading documents and conducting research on topics that interest us. • After selecting a topic as a group, spend roughly 5-8 hours researching our topic.
Everyone (individually)	Faculty meeting - Everyone is to research their specific area and discus possible implementation on the generated idea	It is expected that the group member has conducted sufficient research in the topic to be able to provide a rough draft of implementation.
Everyone (together)	Finalizing the problem - As a group we will meet and discuss the information we have gathered and decide on the final idea of iteration 2. We will discuss the design and implementation of iteration 2.	1-3 hours

Everyone (together) and for ordering parts we will select someone during the meeting to do it.	Design brainstorming and ordering parts - We will meet and discuss the design of our project and then begin to order parts. We will select someone during these meetings to conduct research on where these parts can be purchased and order them.	We will discuss our design over the course of 2-3 weeks and will meet regularly during those 2-3 weeks and order the parts as needed. Roughly 3 hours per week will be spent weekly on this.
Everyone	Iteration 2 - in this stage we are focusing on design planning after ordering parts, and conduct testing as needed. It will be very modular to ensure that components of our design are coming together and we will constantly be updating and implementing our design during this phase.	We have been able to narrow down our components and parts needed to the following: Copper sheets Conductive fabric Wooden frame Arduino Uno
Everyone	Iteration 2 - This is a mix of continuously updating our design as we encounter issues and also implementing our design. Additionally, we are focusing on creating software diagrams, and visuals for better explaining our project objectives and goals.	

Table 1: Personal Effort

3.7 OTHER RESOURCE REQUIREMENTS

- **Project Materials and Equipment List**

User Control Surface

- Base and Connection Materials:
    - 4 copper sheets, acting as the base and conduit for connecting wires.
    - 4 conductive fabric sheets , laid on top of the copper sheets for user interaction.
    - Conductive glue, used to adhere the copper sheets to the carbon fiber sheets.
- Flexible Materials:
    - Flexible materials for connecting 4 signal units.
    - Wires and soldering materials for connection to the copper sheets.

Signal Processing

- Core Equipment:
    - 1 Arduino development board.
- Electronic Components:
    - Several jumper wires.
    - 4 1M ohm resistors.
    - 2 LED indicators.
    - 1 PCB board designed specifically for this circuit.

Display

- Display Device:
    - 1 Samsung tablet.
- Connection Converter:
    - 1 USB to USB-C converter.

4 Design

Generally, the public lacks awareness of engineering problems, specifically the 14 Grand Challenges for Engineering, and their states of development. The understanding that the public has of the Grand Challenges (what they are and current state) influences their enthusiasm and engagement. If they do not understand the Challenges, then they will not have enthusiasm or engagement with their solutions. If they understand them then they will (more likely) have enthusiasm and engagement with their solutions. To address this issue, we aim to develop an interactive

installation that appeals to people of all ages and effectively conveys Grand Challenges and their current states of development. An enthused and engaged public is a source of funding, research (new minds) and application. If the public is more educated, then Grand Challenges are more likely to be solved in a faster time. We will use an artistic demonstration to engage and educate the public in the state of urban infrastructure development as a means to ensure they support the research and development of Grand Challenges.

4.1 DESIGN CONTENT

Our project has many design considerations that cover both hardware and software that involves multiple different disciplines that are necessary to research and be informed about. As our project is an interactive installation, we need to think about how every party will interact with our creation. This not only includes the audience that will be tap touching a surface and playing with our project, but also those who need to store and maintain it. As such, we have to ask questions about how quickly it can be assembled and disassembled, and how it can be properly stored so non-technical faculty can keep educating the public about engineering challenges years after we graduate.

4.2 DESIGN COMPLEXITY

Our design requires us to have fast and responsive user interaction that communicates reliably from hardware to software. In our final design, users interact with conductive fabric connected to a tablet through an Arduino configured to detect electrical frequency fluctuation (see image 2 in section 4.6.1.1). The frequency detection software is powered by touché, a technology developed by Disney research labs. This technology is capable of a versatile array of touch identification related tasks; we have used it to identify when a user touches the fabric. The frequency data is processed by the Arduino, then sent to our display through a USB connection where it is displayed. We not only have to have quick turnarounds, use reliable interactive elements in both hardware and software that are informative, but also research and calculate the most important message we can convey with the display that is verifiable and backed by experts in urban designers and civil engineers that are on and off campus.

Due to the need to display our designs in public settings, we are developing a set of hardware devices that interact directly with users. Initially, we planned to use 3D-printed bricks as the medium to trigger the program. However, in practical applications, these 3D-printed bricks could

not be read by microcontrollers due to their insulating material properties. Consequently, we abandoned the idea of placing bricks on the display surface and instead designed a conductive, touch-sensitive surface. This approach ensures that the data can be read by computers and microcontrollers.

4.3 MODERN ENGINEERING TOOLS

What modern engineering tools were used for this design? Their roles.

Hardware:

- PCB Printer – Used for creating our PCB for the hardware
- 3D Printer – 3D prints out the optional enclosure
- Wired Data Transfer – connects the input terminal, signal conversion system and output.
- Tablet – displays android application which visualizes the serialized input of the Arduino in a fun and exciting way
- Conductive material (touch sensitive) - As the input terminal of the entire device.
- Copper – Strengthens the signal of the input, also used as a vector for soldering on wires

Software:

- SolidWorks – Build the 3D model for the system enclosure
- Android Studio – Create an application for visualizing the data inputs by the copper/conductive material
- Arduino IDE – Used to program the Arduino itself to track and process signals from the data inputsgantt

4.4 DESIGN CONTEXT

Describe the broader context in which our design problem is situated. What communities are we designing for? What communities are affected by our design? What societal needs does our project address?

List relevant considerations related to our project in each of the following areas:

Area	Description	Examples

Public health, safety, and welfare	This project is educational, and showcases how the full-scale city can account for public health and safety.	Visualize hospitals and public places, create a more walkable city for the public
Global, cultural, and social	This project has the potential to have an impact in many regions. Since it is a planning tool to be used as a showcase, it has widespread functions.	Rural and downtown city planning, lunar colonization, planetary colonization
Environmental	This project aims to showcase a sustainability aspect. Redesigning cities can in the moment hurt the environment, but there are ways that the city can be built sustainably and with the environment in mind.	Rooftop gardens, urban parks, solar cells, energy efficient lighting
Economic	This project will showcase ways to cost-effectively create a city with a model before actually setting plans for a city design or redesign.	The model itself is a low-cost system, creating these smart cities will prove cost-beneficial in the future
Stakeholders	**Description**	**Impact**

Urban Designers	Groups of people who are using this software for its intended purpose. Such as: City Planners, Park designers.	People designing cities or parks with this software will greatly benefit from the ability to iterate through designs at a higher speed. - Increased iterative design speeds.
Residents	Residents, locals, visitors, business owners, etc. Any person or group that will benefit from well-designed urban layouts.	Increasing the speed of design iterations will reduce the cost for designers to correct mistakes, and incentivize them to design urban areas that more accessible with higher utility and beauty. - Faster error correction means better designed environments
Cities	Cities, States, Countries, or any other groups that may contract the design of an environment consisting of multiple buildings.	This software will reduce the cost to design an urban style environment. - Reduced cost for contracts.

Table 2:   Relevant Considerations Project Areas

4.5 DESIGN DECISIONS

We are creating an Android app using Android Studio with Java as our primary language. While Kotlin is the preferred language for Android development currently, there are many examples that are useful to us, but they are only written in Java. As for our Arduino hardware, we are using the C programming language, as it has most examples out and many libraries to help us send data.

We will be using four pieces of conductive fabric and four copper plates to trigger our input. This will be wired directly to our Arduino which will process the input that gets triggered. The wires will

be soldered onto a copper strip that we will embed into the copper plates. This is done by drilling a hole and soldering the copper strip on both sides since the thermal distribution is too great to solder the wire directly onto the plate.

We will be placing everything onto a printed circuit board (or PCB) for ease of use and to clean up all of the wiring that is currently used with our prototype.

4.6 PROPOSED DESIGN

Upon receiving new design constraints from our client to implement the use of Touché touch technology, we scrapped the idea of using photoresistors and began brainstorming how to implement Touché into our design. Our design now consists of four copper plates, forming a 2x2 grid, being used as conductive surfaces. Each plate will have a wire connecting it to a PCB board which is connected to a tablet. The user will touch one of the copper plates and the signal will be detected by the PCB board and sent to the tablet where our Android app will process the signal and use it to trigger a set of commands that correspond to our app. These commands would be choosing between a set of four options where each copper plate would correspond to one of these choices. Upon a choice being selected the app will update to reflect that change. Our app is a simulation of an urban environment where each choice the user makes will provide new information about that environment or add new components to it. We will be providing a demo of this process at work during our final presentation.

4.6.1 Design

Pictured below is our 2nd Iteration design which we were implementing during the second semester. Upon receiving new design constraints from our client to implement the use of Touché touch technology, we scrapped the idea of using photoresistors and began brainstorming how to implement Touché into our design. Our design now consists of four copper plates, forming a 2x2 grid, being used as conductive surfaces. Each plate will have a wire connecting it to a PCB board which is connected to a tablet. The user will touch one of the copper plates and the signal will be detected by the PCB board and sent to the tablet where our Android app will process the signal and use it to trigger a set of commands that correspond to our app. These commands would be choosing between a set of four options where each copper plate would correspond to one of these choices. Upon a choice being selected the app will update to reflect that change. Our app is a simulation of an urban environment where each choice the user makes will provide new information about that environment or add new components to it. This allows us to create a fun,

interactive, artistic, and informative way to teach kids ages k-12 about ways to improve urban infrastructure which is the goal of this project. Overall our design does meet both our functional and non-functional requirements. Our function requirements were that we wanted to educate the public (kids k-12) on how to restore or improve urban infrastructure. We wanted to do this in a way that implements all of our engineering majors and is done in an artistic manner. Our nonfunctional requirements were that we wanted to make our project portable and functionally fun. The purpose of this project is to create some kind of interactive art piece that can inform kids k-12 how to improve urban infrastructure. This is done by allowing the user interact with our model environment on the app by allowing them to touch one of the copper plates which will correspond to a change in the model environment which will display information in the form of fun facts on ways to improve urban infrastructure. The image below shows a sketch of what our design plan looks like. This project also makes use of all of our engineering majors which was another requirement for our project.
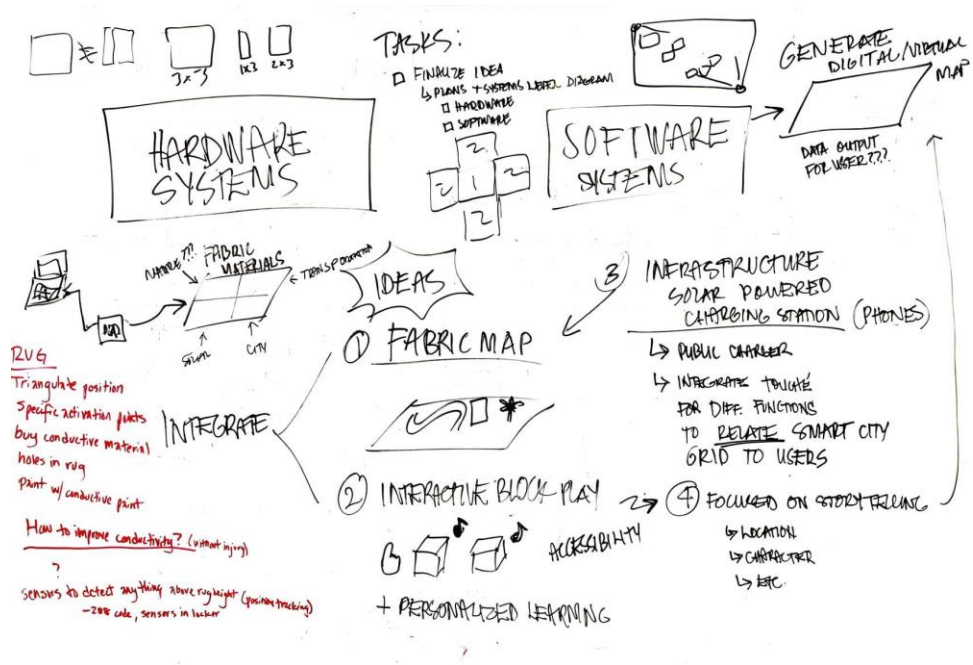
Figure 4E: Touché Version of the Interactive Urban Infrastructure Visualization Concept

Upon receiving new design constraints from our client to implement the use of Touché touch technology, we scrapped the idea of using photoresistors and began brainstorming how to implement Touché into our design. Our design now consists of four copper plates, forming a 2x2 grid, being used as conductive surfaces. Each plate will have a wire connecting it to a PCB board which is connected to a tablet. The user will touch one of the copper plates and the signal will be

detected by the PCB board and sent to the tablet where our Android app will process the signal and use it to trigger a set of commands that correspond to our app. These commands would be choosing between a set of four options where each copper plate would correspond to one of these choices. Upon a choice being selected the app will update to reflect that change. Our app is a simulation of an urban environment where each choice the user makes will provide new information about that environment or add new components to it.
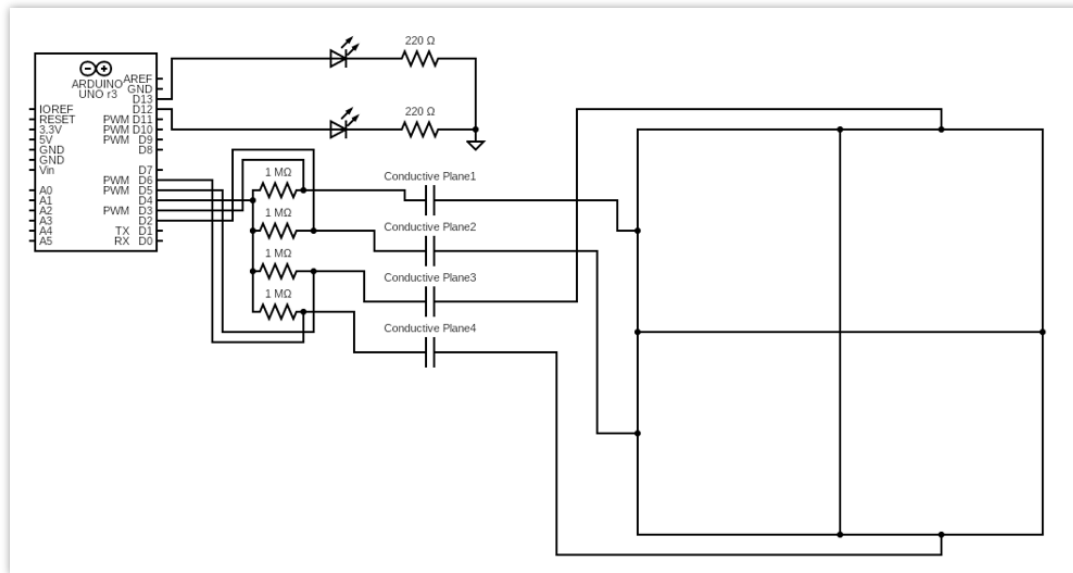

Figure 4F: Touché Circuit Design

The picture above is the design of the Touché circuit that we made. The four copper plates will detect touch of both a persons hand or an object (like a building) and that signal will be sent to the Arduino which will send that information to the Android App which will let the user know that an object has contacted the copper plates. We were initially planning on our app being able to detect the different types of buildings that were placed on the copper plates. However due to time constraints and complexities in the design process of being able to detect the different types of buildings when they contact the copper plates we decided to shift off of that and just focus on the copper plates being able to detect touch. So our design simply became that a person would touch one of the four copper plates and that touch signal would be sent to our app and would correspond to an interactive component of the app.
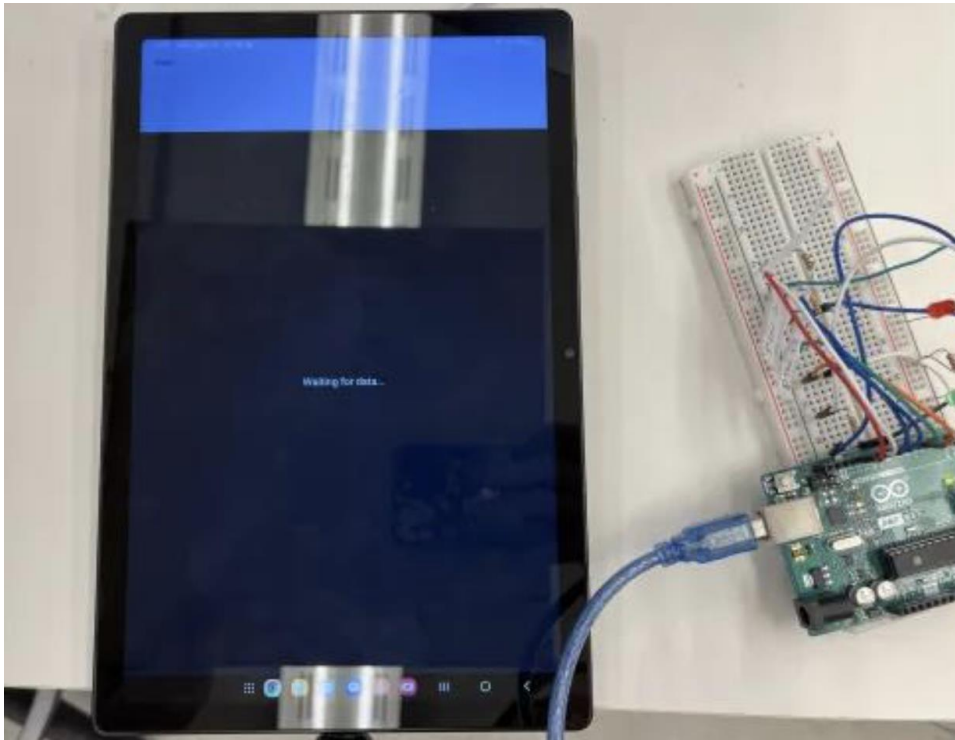
Figure 4G: Touché circuit breadboard design hooked up to tablet

The picture above is our prototype design for the Touché circuit which is connected to the tablet. There is no connection made to the copper plates in this picture as they were in the process of being shipped to us. So, we just used wires and made sure the circuit was functional by touching the ends of the wire with our hands. Touché works by detecting the capacitance when something comes in contact with the conductive material (like the copper plates) and quantifies that amount so that it can be used to detect if something is touching it and what object it could be by looking at the magnitude of the capacitance.

The picture above is what our interactive app user screen looks like. All the information that is received from the four copper capacitive plates will be sent and displayed 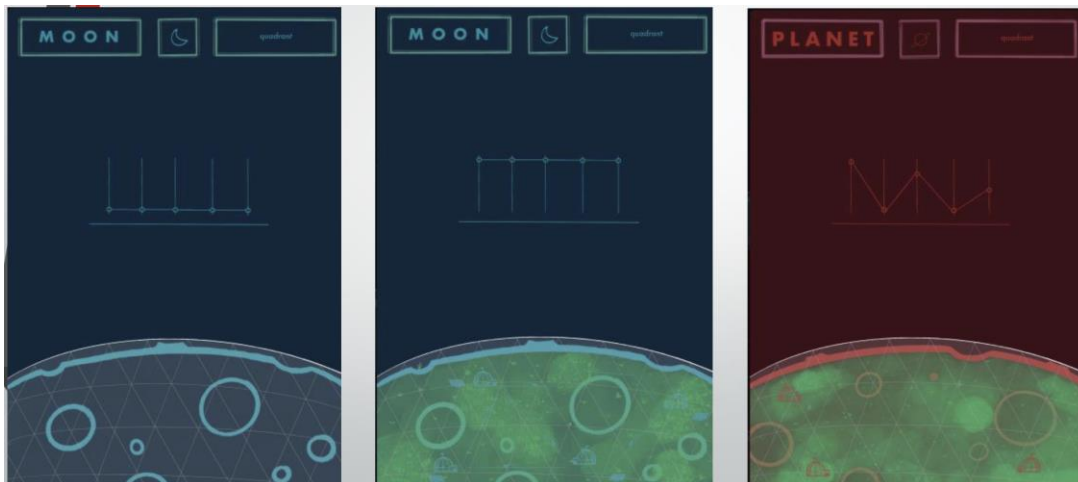here. The app will have interactive components to it where trivia questions will appear on the screen and the user will select the answer to that question by selecting one of the copper plates. That answer will in turn lead to a change in the image that is displayed on the screen. This will serve as our user interaction with our model environment. This meets the educational, functional fun, and artistic requirements for our project.
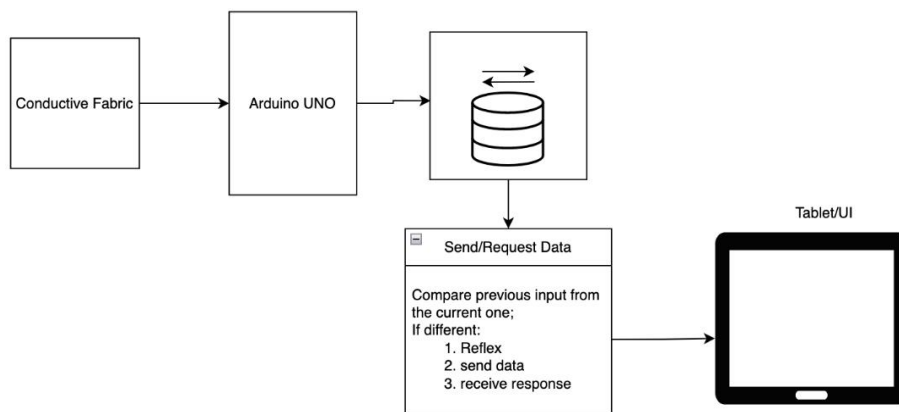
Figure 4I: High Level Systems Chart

The picture above is our high-level system chart which shows how our system will operate. First a question will be displayed on the screen and the user will be given four answer choices which correspond to each of the four copper plates. The user will answer the question by selecting the copper plate and this will generate a change in capacitance and this signal will be sent to the Arduino and then be sent to the tablet to be processed. This will result in a change in the information that is displayed that corresponds to the user's answer choice. By repeating this process over and over again, the user can create their own environment (model city) on the tablet.
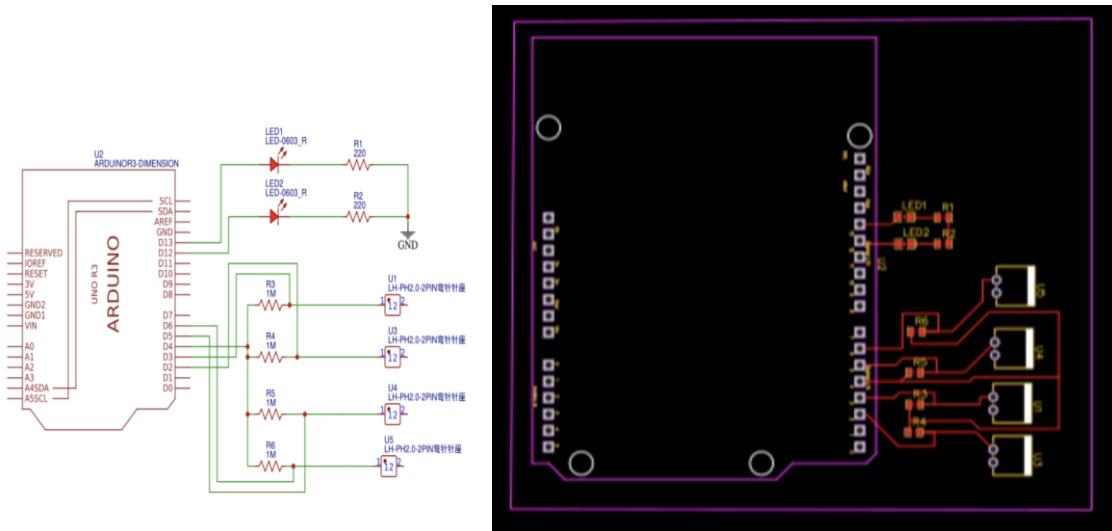
Figure 4J: Touché circuit schematic (left) and Touché circuit PCB design (right)

The pictures above are our Touché circuit schematic and our PCB representation of that same circuit schematic. We choose to implement a PCB board as it would be more visually appealing and easier to maintain compared to a normal breadboard made circuit.

```java
public class UsbBroadcastReceiver extends BroadcastReceiver {

    2 usages
    private static final String TAG = "UsbBroadcastReceiver";

    ≗ Krekula
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (android.hardware.usb.UsbManager.ACTION_USB_DEVICE_ATTACHED.equals(action)) {
            Log.d(TAG, msg: "USB device attached");
            // Handle USB device attached
        } else if (android.hardware.usb.UsbManager.ACTION_USB_DEVICE_DETACHED.equals(action)) {
            Log.d(TAG, msg: "USB device detached");
            // Handle USB device detached
        }
    }
}
```

Figure 4K: Java class UsbBroadcastReceiver

The picture above is one of our Java classes that enables USB broadcast receiving. We chose to use this and the library associated with it for ease of data transfer. Our other option was wireless data transfer, but we decided that this would be unnecessarily complicated to implement, especially since we had all of the hardware on hand.

```java
private boolean setupDataTransfer(UsbInterface iface, UsbDeviceConnection connection) {
    UsbEndpoint endpointIn = null, endpointOut = null;
    for (int i = 0; i < iface.getEndpointCount(); i++) {
        UsbEndpoint ep = iface.getEndpoint(i);
        if (ep.getType() == UsbConstants.USB_ENDPOINT_XFER_BULK) {
            if (ep.getDirection() == UsbConstants.USB_DIR_IN) {
                endpointIn = ep;
                sendToDisplay("endpointIN:" + ep);
            } else if (ep.getDirection() == UsbConstants.USB_DIR_OUT) {
                endpointOut = ep;
                sendToDisplay("endpointOUT:" + ep);
            }
        }
    }

    if (endpointIn == null || endpointOut == null) {
        sendToDisplay("Not all endpoints found.");
        return false;
    }

    UsbSerialDriver driver = UsbSerialProber.getDefaultProber().probeDevice(usbDevice);
    if (driver == null) {
        sendToDisplay("No driver found for device.");
        return false;
    }

    List<UsbSerialPort> ports = driver.getPorts();
    if (ports.isEmpty()) {
        sendToDisplay("No serial ports found on the device.");
        return false;
    }

    // Example setup for serial communication
    try {
        usbSerialPort = ports.get(0);
        usbSerialPort.open(connection);
        usbSerialPort.setParameters( baudRate: 230400, dataBits: 8, UsbSerialPort.STOPBITS_1, UsbSerialPort.PARITY_NONE);
        sendToDisplay("Device is ready for communication.");
        startSerialCommunication();
    } catch (IOException e) {
        sendToDisplay("Error setting up device: " + e.getMessage());
        return false;
    }
    return true;
}

private boolean checkUSBConnection() {
    HashMap<String, UsbDevice> usbDevices = usbManager.getDeviceList();
    Log.d( tag: "Devices",  msg: ": " + usbManager.getDeviceList());
    if (usbDevices.isEmpty()) {
        statusTextView.setText("EMPTY Device List!\nPlease connect Arduino...");
        return false;
    }
    if (usbDevice != null && isArduinoDevice(usbDevice)) {
        statusTextView.setText("Arduino found: " + usbDevice.getDeviceName());
        return true;
    }
    statusTextView.setText("NO Arduino Found!\nPlease connect Arduino...");
    return false;
}

1 usage   ± Krekula
private void requestUsbPermission(UsbDevice device) {
    Intent intent = new Intent( action: "com.example.androidapp.USB_PERMISSION");
    PendingIntent permissionIntent = PendingIntent.getBroadcast( context: this,  requestCode: 0, intent, PendingIntent.FLAG_IMMUTABLE);

    // Log or display status before requesting permission
    statusTextView.setText("Arduino found! Permission request sent...");
    Log.d(TAG,  msg: "Requesting USB permission");

    // Request permission
    usbManager.requestPermission(device, permissionIntent);
}

2 usages   ± Krekula
private boolean isArduinoDevice(UsbDevice device) {
    // Replace with your Arduino's Vendor ID and Product ID
    return device.getVendorId() == 9025 && device.getProductId() == 0x0043;
}
```

```java
private void openDeviceConnection(UsbDevice device) {
    sendToDisplay("...openDevice");

    // Open the USB device connection
    UsbDeviceConnection connection = usbManager.openDevice(device);
    if (connection == null) {
        sendToDisplay("...Error opening device connection.");
        return;
    }

    // Configure USB interface
    UsbInterface iface = configureUsbInterface(device, connection);
    if (iface == null) {
        connection.close();
        return;
    }

    // Setup data transfer
    if (!setupDataTransfer(iface, connection)) {
        sendToDisplay("Failed to setup data transfer.");
        connection.releaseInterface(iface);
        connection.close();
    } else {

    }
}

1 usage   ± Krekula +1
private UsbInterface configureUsbInterface(UsbDevice device, UsbDeviceConnection connection) {
    sendToDisplay("Number of Interfaces: " + device.getInterfaceCount());
    for (int i = 0; i < device.getInterfaceCount(); i++) {
        UsbInterface usbInterface = device.getInterface(i);
        if (usbInterface.getInterfaceClass() == 10 && usbInterface.getInterfaceSubclass() == 0 && usbInterface.getInterfaceProtocol() == 0)
            boolean claimed = connection.claimInterface(usbInterface,  force: true);
            if (!claimed) {
                sendToDisplay("Error claiming interface.");
                return null;
            }
            return usbInterface;
    }
    sendToDisplay("Suitable interface not found.");
    return null;
}
```

Figure 4L: Java class MainActivity setting up data transfer (top left), opening the device connection/configuring the USB interface (top right), checking the USB connection/requesting USB permission/checking if the device is an Arduino (bottom)

The pictures above are portions of our main Java class. These sections specifically set up the data transfer, device connection, USB interface, USB connection, USB permissions, and check if the device that is connected is an Arduino. Our program checks specifically for one Arduino using its product ID, so at the moment, only one Arduino can be used for this implementation. We added checks for errors that display on the device itself for testing purposes, since it is not possible to debug the application through Android Studio while it is plugged into the Arduino.

3rd Iteration

The 3rd iteration of our design is the exact same as the 2nd iteration with the only difference being a change in the PCB breadboard schematic and design. A picture of our updated schematic and PCB

design is shown below for reference. We realized upon testing our PCB board that our circuit wasn't behaving in the manner in which we wanted. The issue was that there were four channels in our design and we expected to see four signals that were sent back to the Arduino however we only saw one. The reason for this issue was that we had all the resistors connected to the D4 pin on our Arduino so everything was being read as one signal. To correct this issue on our new PCB design we just made sure that each resistor was connected up to two pins on the Arduino. For example R3 is connected to pins D12 and D13 on the Arduino. This allows for all the channels to be operational in our model.
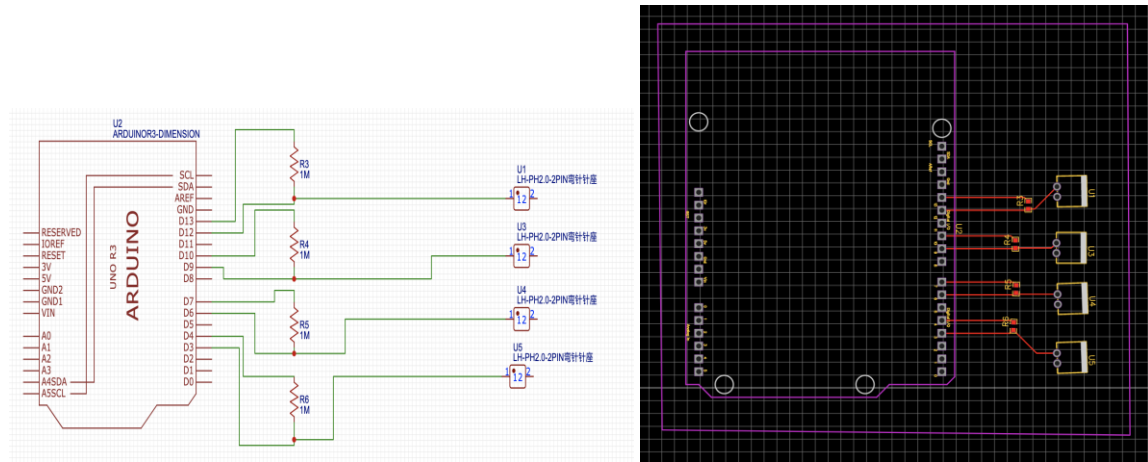

Figure 4K: New Touché circuit schematic (left) and New Touché circuit PCB design (right)

4.6.1.2 Functionality

Describe how our design is intended to operate in its user and/or real-world context. This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

Pictured below is our 2nd Iteration design which we were implementing during the second semester. Upon receiving new design constraints from our client to implement the use of Touché touch technology, we scrapped the idea of using photoresistors and began brainstorming how to implement Touché into our design. Our design now consists of four copper plates, forming a 2x2 grid, being used as conductive surfaces. Each plate will have a wire connecting it to a PCB board which is connected to a tablet. The user will touch one of the copper plates and the signal will be detected by the PCB board and sent to the tablet where our Android app will process the signal and use it to trigger a set of commands that correspond to our app. These commands would be choosing between a set of four options where each copper plate would correspond to one of these choices. Upon a choice being selected the app will update to reflect that change. Our app is a simulation of an urban environment where each choice the user makes will provide new information about that environment or add new components to it. This allows us to create a fun, interactive, artistic, and informative way to teach kids ages k-12 about ways to improve urban infrastructure which is the goal of this project. The purpose of this project is to create some kind of

interactive art piece that can inform kids k-12 how to improve urban infrastructure. This is done by allowing the user interact with our model environment on the app by allowing them to touch one of the copper plates which will correspond to a change in the model environment which will display information in the form of fun facts on ways to improve urban infrastructure. The image below shows a sketch of what our design plan looks like. In section 4.6.1.1 (above) we have described the progress we made on this 2nd Iteration, so you can reference that section for further information.
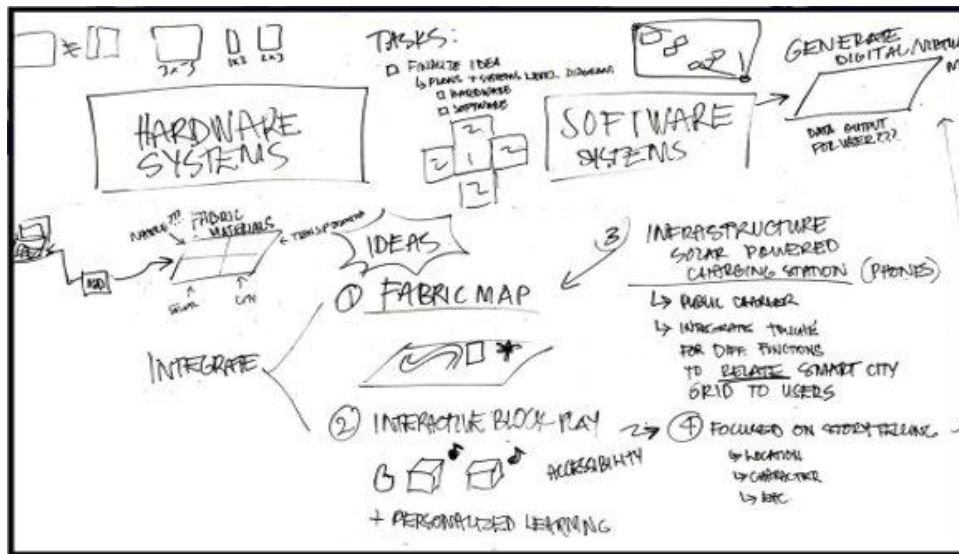

Figure 4.2: Touché Version of the Interactive Urban Infrastructure Visualization Concept

4.6.2 Design (Design Iteration)

As mentioned above for a more detailed description please reference section 4.6.1.1 of this report. The major changes we had in our second iteration was the implementation of Touché conductive touch technology to replace the use of photoresistors and light sensing technology. The major reason for this was because transporting our project around would become a hassle as those parts were easily breakable. Sour our client as us to implement Touché conductive touch technology specifically. The second major change that came in the 2nd iteration of this design is that due to limited time and complexity of the design, we knew we wouldn't be able to implement a model city. So, we opted for a four-channel copper plate setup where we could ask the user questions about urban infrastructure and they could touch the copper plate that corresponds to that answer choice and this would lead to the environment in our app to change to reflect the answer choice they chose and then provide fun facts on how to improve that specific piece of urban infrastructure.
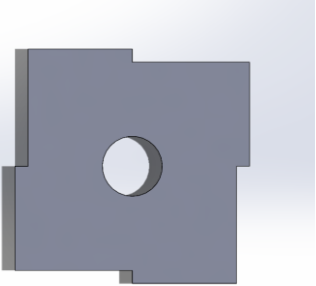
4.6.3 Design 2 (Design Iteration)

Our current enclosure was developed more closely with the non-functional requirements, and takes into account the various stakeholders and users of the finished product.

One of the significant changes to the design was to add a door on the bottom of the box that could be screwed open to access the internals of the touch surface. This allows instructors and teachers to repair the project in the future. It also inhibits access to the internals for younger children.

The touch surface is constructed using ½ inch plywood, a relatively sturdy yet inexpensive material sanded to be soft to the touch and painted to match the application and branding of our deliverables.

Each wooden part was designed to be easily cut using a bandsaw without any difficult turn.

Part Name	Description	Visual
Top Surface	This part is for the top of the box. The conductive sheet of copper is laid over the Top Surface, and a wire is attached underneath the copper sheet through the 1.75-inch radius hole that is cut. The extrusions on the side act as finger joints that fit snugly on the sides. These sides are equal so we only needed to design one kind of side panel that we could cut four times for each side.	
Sides	One side of the box. The total standing height for the box is be 2.5 inches. Where the initial measurement of the height is 2 inches, and you add the ½ inch height of the finger joints that are joined with the top surface There are additional finger joints on the left and right edges extruding for half an inch each. Half inch finger joints were chosen specifically so we would not need to alter the thickness of the ½ inch plywood that we would be using with the bandsaw. The distance between the bottom of the top surface and the bottom of the	

	box is 2 inches, as the top surface bottom is level with the lower level of the finger joints. One side will have a hole that is drilled and adjusted to include a single wire output	
Felt feet	Felt feet are attached to the bottom of the box's sides in each of the four corners to prevent scratching and damage to the surface.	N/A
Wire port	There is a wire port that is used to connect a separate wire between the Arduino and the surface.	N/A

4.6.3.1 Design Visual and Description

The information for this section has already been described in section 4.6.1.1 so to repeat information we ask that you reference that section of the report. That section of the report describes all the major components of our 1st and 2nd iterations with lots of visuals to help understand what's being discussed. Section 4.6.2 also has some more visuals and information about the two interactions for additional information but the major components are discussed in section 4.6.1.1.

4.7 DESIGN CONSIDERATIONS

When deciding on our design, it was important to consider the functional requirements and all the stakeholders for the project. We had to consider budget, the audience, and how the design would be built and used.

4.7.1 Technology Considerations

We decided to select the technology we were going to use in accordance to our functional and non-functional requirements. In particular, we wanted to ensure that future senior design groups and others will be able to understand and repair the design. As such, we chose hardware and languages with good documentation.

2nd Iteration:

The major change we made in our 2nd Iteration was that we switched from using light sensing technology (photoresistors) with Touché capacitive touch sensing technology at the request of our client. Touché is a capacitive touch sensing technology that was designed by Disney that is able to detect touch on a surface based on the change in capacitance in the circuit. To implement this technology we used the sample code provided by Disney and changed it from one channel to four channels for our design. We also selected to use four copper plates to serve as our

touch sensitive surfaces due to the high conductivity of copper. This would make the surface very sensitive and produce a higher signal compared to materials with poor conductivity.

We made both a prototype breadboard circuit and a finalized PCB board for our circuit. The circuit design was fairly simple and the Arduino we used was the Arduino Uno R3 and we used that model simply because one of our group members had them. We technically could have used any Arduino model however we used the Uno R3 because it was available to us at the time. We chose to make our circuit on a breadboard at first rather than a PCB because we were still in the prototype phase and wanted to be able to play around with components in the circuit which could not be done if we made a PCB model. This allowed us to make mistakes and improve the circuit design freely. Once the design was finalized we implemented a PCB version of it by using the PCB design website EsayEDA to create and order our PCB. The PCB was needed for our project because as we mentioned we wanted to make our project portable and that was not easy to do with breadboard circuits as they could get damaged and break apart.

We implemented our app by using Android Studio because our team members were familiar with the software. The tablet our team used was also an Android so we had to use and Android compatible app designer. The main reason we chose this software was because our team knew how to use this software from prior classes.

4.7.2 Enclosure Considerations

The enclosure for the conductive surfaces was an important design component, as it is the primary way our target audience interacts with the project. We had to make sure the enclosure is repairable with materials that don't need to ordered online, but rather from local hardware stores.

What the enclosure was going to be made of was an important choice to make. We decided on wood, as it would be one of the more inexpensive materials that we could use to construct box out of. Additionally, we looked at other common children's toys, and recognized the historical and longstanding use of wood for toys. Wood also meets the technical requirements of the area surrounding the surface being non-conductive.

The chosen conductive surface to lay on top of the wooden enclosure was chosen to be copper with a overlay of conductive fabric. Copper was chosen because it is extremely conductive and ensures we get the best readings from the ADC feedback. However, we recognize that copper is a very tough material that is not friendly, so we chose to overlay the metal with a softer conductive fabric that is more welcoming that the harsh metal.

For the design of the enclosure, we wanted to make our enclosure construction and assembly to be simple, requiring only one working tool, that being the bandsaw to cut the materials for the enclosure. Additionally, the entire enclosure was designed to be made with only one sheet of 12in x 24in plywood sheet, with every part fitting on the plank and ready to cut.

The surface area of the conductive fabric was cut to a size that is big enough for the hand sizes for our target audience, that being K-12 kids.

4.8 DESIGN ANALYSIS

This design is a significant improvement from our first iteration. It combines the grid spaces that we initially came up with and combines it with touch sensitivity, which allows for touch play. Touch play is essential for fulfilling the requirement of letting this project be accessible to people of all ages, as very young or very old people may not have the capacity to grip small blocks. Touch allows almost any user to interact with the display, and the enclosure allows for a sleek design that also functions as a mild insulator, standardizing the frequency starting point. The tablet also has a simple but fun design for the game portion of the project, and making progress is both visually and analytically appealing. The PCB also contributed to the sleekness of the design, transforming our product from a breadboard containing a jumble of wires to a singular board with four wires and one connecting cable. The project is easy to transport as it is modular, and it is difficult to break.

5 Testing

Testing is an extremely important component of most projects, whether it involves a circuit, a process, power system, or software.

Units being tested include the conductive material's electrical properties, computational geometry algorithms, and the digital display algorithms. Tools might include electrical testing tools and software debugging tools.

Given the complexity of both hardware and software components, the testing strategy will focus on individual unit tests, interface tests between components, integration tests, system-level tests, regression tests, and acceptance tests. A unique challenge in testing is ensuring real-time responsiveness and accuracy in detecting block placements and translating them to the digital display.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for our system/design.

The primary interfaces in the design are between the block placement detection system and the digital display. Testing will ensure that data is correctly and swiftly transmitted from the hardware to the software. Tools would include data logging and visualization tools.

5.1 UNIT TESTING

What units are being tested? How? Tools?

Units being tested:

- **Hardware:**
  - **Surface**
    - Durability Tests: These tests assess the surface's resistance to scratches, dents, and bends. Methods might include hardness testing, drop tests, and bend tests.
    - Environmental Tests: To check the surface's resilience to various environmental factors like humidity, temperature extremes, and UV exposure, environmental chambers are used.
    - Aesthetic and Ergonomic Assessment: This involves evaluating the surface's appearance, texture, and how comfortable it is to handle. User studies might be conducted for this purpose.
  - **Processor**
    - Performance Testing: Benchmarking the processor using various tools and software to assess its speed, efficiency, and ability to handle multiple tasks simultaneously (multi-threading).
    - Stress Testing: Running the processor at full capacity for extended periods to ensure stability and to check for overheating issues.
    - Power Consumption and Heat Dissipation: Testing how much power the processor consumes under various loads and how effectively the heat is dissipated.
  - **Tablet Display**

- Color Accuracy and Brightness Tests: Using colorimeters and light meters to assess color accuracy, brightness levels, and contrast ratios.
- Viewing Angle Tests: Checking how the display's visibility and color consistency change at different angles.
- Touchscreen Functionality: If the display is touch-sensitive, tests are conducted to assess the responsiveness, accuracy, and multi-touch capabilities.

- **Input/output ports**
  - Connectivity Tests: Checking each port's ability to establish and maintain a connection with various peripherals.
  - Data Transfer Speed Tests: Measuring the speed and reliability of data transfer through the ports.
  - Durability and Wear Tests: Repeatedly plugging in and unplugging devices to ensure the ports can withstand regular use.
  - Electrical Testing: Ensuring that the ports provide correct power output and data signals.

- **Software:**
  - **Analog to digital conversion (De-modulization)**
    - Analog to Digital Conversion (ADC) is the process of converting continuous analog signals from our touch surface into discrete digital values. This is a critical step in digital signal processing where real-world analog user inputs are converted into a digital format that can be processed by our software.
    - Accuracy Testing: Compare the digital output with the expected values for given analog inputs to ensure accurate conversion. This can involve using signal generators and oscilloscopes.
    - Noise and Interference Testing: Introduce noise or interference in the analog signal to verify how well the ADC handles real-world imperfect signals.
    - Range and Resolution Testing: Test the ADC across its entire input range to ensure consistent performance.
  - **Data input**

- Data input refers to the process of receiving data from external sources for further processing. This will happen via user input through sensors in our touch surface and through mouse and keyboard input devices.
- Input Validation Testing: Ensure that the software correctly accepts and processes valid inputs from touch surfaces, mice, and keyboards, and rejects invalid inputs.
- Stress Testing: Simulate high-frequency inputs or large volumes of data to test the system's ability to handle heavy loads.

- **Data Scrubbing**
  - Data scrubbing, also known as data cleaning, will involve identifying and correcting (or removing) errors and inconsistencies from data to improve its quality. This step is crucial in ensuring that the subsequent data analysis is accurate and reliable.
  - Data Quality Assessment: Introduce datasets with known errors or inconsistencies and verify that the scrubbing process identifies and corrects these issues.
  - Performance Testing: Measure the efficiency of the data scrubbing process, especially when handling large datasets.

- **Data Parsing:**
  - Data parsing is the process of taking input data and converting it into a more usable format. This will involve interpreting and separating the data stream into discrete data packages that can be used in the processing stage according to a set of predefined rules or formats.
  - Functional Testing: Test the parser with a variety of input data formats to ensure it correctly interprets and separates data as expected.
  - Error Handling: Verify how the parser reacts to malformed or unexpected data, ensuring it fails gracefully or throws appropriate exceptions.

- **Data Processing**
  - Data processing will the transformation of parsed data into meaningful information through a series of actions or operations. This will include data classification based on input source. It will also sort and tag data based block type and location on the touch surface.

- **Data Modeling**

- Data modeling will be the process of taking processed data and translating it into the in memory model of the surface. This will be emplacing the blocks onto the virtualized surface.
- Model Verification: Compare the in-memory model generated by the software with the expected model for given inputs.
- Performance and Scalability Testing: Assess how the data modeling process scales with increasingly complex or large datasets.
- **Rendering**
  - Rendering will be the process of generating a final image from the modeled data. This term will be graphics and visualization, where data is converted into a graphical representation.
  - Visual Inspection: Manually inspect the rendered images to ensure they visually match the expected output.
  - Automated Image Comparison: Use software tools to compare rendered images with a set of reference images for consistency.
  - Performance Testing: Evaluate rendering times and resource usage, particularly under different hardware configurations.

5.2 INTERFACE TESTING

What are the interfaces in our design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

The interfaces necessary for the function of the interactive installation is the interaction between the conductive surface and the microcontroller, and the connection between the microcontroller and the tablet. Additionally, we must ensure that the data that is delivered to the tablet is correctly reflected on the display. These interface tests are essential and tie into our functional requirement of designing and building an accessible that is easy to operate without engineers present.

Key Hardware Interfaces in the Design:

- **Testing Surface Hardware Interfaces:**
  - Connector Durability Testing: If the surface includes connectors (like a detachable keyboard or accessories), test the durability and reliability of these connectors.
  - Signal Integrity Testing: For surfaces that interact with other components (like touchscreens), test the quality and reliability of the signal transmission between the surface and the control unit.

- **Testing Processor Hardware Interfaces:**
  - o Socket/Connection Testing: Test the physical connection between the processor and the motherboard (like the CPU socket) for proper contact and signal transmission.
  - o Bus Interface Testing: Evaluate the communication interfaces (like PCIe, USB, etc.) to ensure they transmit data correctly and at the expected speeds.

- **Testing Display Hardware Interfaces:**
  - o Display Connector Testing: Test the connectors (like HDMI, DisplayPort, VGA) for signal transmission quality and physical durability.
  - o Touch Interface Testing: For touchscreens, test the interface that connects the touch panel to the processing unit, ensuring accurate and consistent touch data transmission.

- **Testing Input/Output Port Hardware Interfaces:**
  - o Port-to-Microcontroller Testing: Check the connection between the I/O ports and the motherboard or the main control unit, ensuring stable and correct data transfer.
  - o Peripheral Compatibility Testing: Test compatibility with a range of peripheral devices to ensure the ports can communicate with a wide array of external hardware.

- **General Hardware Interface Testing Methods:**
  - o Electromagnetic Compatibility (EMC) Testing: Ensure that electromagnetic interference does not affect the performance of the interfaces.
  - o Stress Testing: Subject the interfaces to high data rates, continuous operation, and extreme conditions to assess their durability and performance under stress.
  - o Environmental Testing: Expose the interfaces to various environmental conditions like temperature extremes, humidity, and vibration to ensure they operate reliably under different scenarios.

- **Key Software Interfaces in the Design:**
  - o Analog-to-Digital Conversion Interface:

- The interface between the physical touch surface (hardware) and the ADC software component.
  - o Data Input Interface:
    - The interface through which user inputs (from touch surface, mouse, keyboard) are received by the software.
  - o Data Scrubbing Interface:
    - The interface between the data input stage and the data scrubbing/cleaning module.
  - o Data Parsing Interface:
    - The interface between the data scrubbing module and the data parsing module.
  - o Data Processing Interface:
    - The interface between the data parsing module and the data processing logic.
  - o Data Modeling Interface:
    - The interface between the data processing module and the data modeling component.
  - o Rendering Interface:
    - The interface between the data model and the rendering engine.

- **Testing Composition of Interfaces:**
  - o When testing the composition of two or more interfaces, the focus is on integration testing. This involves verifying that data flows correctly between different modules and that these modules interact as expected.
- **Scenario-Based Testing:**
  - o Create real-world scenarios to simulate how different components interact.
- **Tools for Testing:**
  - o Automated Testing Frameworks:
    - Tools like Selenium, Postman, or JUnit can be used for automating interface tests.
  - o Mocking and Stubbing Tools:
    - Tools like Mockito or WireMock can simulate parts of the system (like external services or data sources) that are not being tested.
  - o Performance Testing Tools:

- Tools like JMeter or LoadRunner to test how interfaces perform under load.
  - Static Code Analysis Tools:
    - Tools like SonarQube can be used to analyze the integration points for potential issues in code quality or security vulnerabilities.
  - Continuous Integration Tools:
    - Platforms like Jenkins or Travis CI to automate testing as part of the development pipeline.

- **Testing Data Flow and Dependencies:**
  - Ensure that data passed between interfaces is correct both in format and content.
  - Test how changes in one module affect the downstream modules (e.g., how data scrubbing impacts data parsing).

- **Testing Error Handling and Failures:**
  - Simulate failures at various interfaces to test the system's robustness and error handling capabilities.
  - User Acceptance Testing (UAT):
    - Review implementation and test with target audience (younger children) to ensure that our goals are being met and our messaging is coming across in a fun and joyful manner.

5.3 INTEGRATION TESTING

What are the critical integration paths in our design? Justification for criticality may come from our requirements. How will they be tested? Tools?

To demonstrate that design requirements are met, end-to-end tests will be conducted with representative users playing with the installation and receiving feedback from the display. Client involvement would include regular demonstrations and feedback sessions to ensure the project aligns with their expectations.

Critical integration paths include the real-time transmission of touch feedback to the digital visualization and the integration of research data into the educational display. These paths will be tested for speed, accuracy, and reliability using software test suites and user testing.

5.4 SYSTEM TESTING

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

Strategy:

For system-level testing, a combination of previously performed unit, interface, and integration tests will be employed. The goal is to ensure that when all components work together, the system behaves as expected. The emphasis will be on real-time responsiveness, accuracy in block placement detection, and the correctness of the educational visualization.

Tools:

- Real-time Monitoring Tools: To assess system response times.
- Data Visualization Tools: To ensure data flow from touch detection to display is correct.
- User Feedback Tools: To get real-time feedback from users during system tests.

Our primary medium of communication, the conductive copper plate, is important to test to ensure that the software properly detects when a user touch event.

5.5 REGRESSION TESTING

Essential aspects of the project that may be more difficult to maintain than other parts include the wooden enclosure and the microcontroller attached.

To ensure new additions or modifications don't compromise existing functionalities, an automated testing suite will be set up. This suite will run previous tests on the system after every significant change.

Critical Features:

- Accuracy of block placement detection.
- Correctness of the educational visualization.
- Responsiveness of the system.

Tools:

- Automated Testing Frameworks: To automate the regression testing process.
- Version Control Systems: To track changes and easily roll back if issues arise.

5.6 ACCEPTANCE TESTING

To demonstrate that design requirements are met, end-to-end tests simulating real-world scenarios will be conducted. Users will touch the surface, and the system's response and educational feedback will be assessed.

The client will be actively involved in this phase, participating in demonstrations and providing feedback. Their insights will ensure that the system not only meets technical requirements but also fulfills the intended educational purpose.

For non-functional requirements, we evaluate the system's performance under different conditions, perform security tests to check if system is protected against unauthorized access, and other security threats.

Acceptance testing is essential to ensure we meet the functional requirements and make the final deliverable kid-friendly and accessible.

We planned on performing one acceptance test with PI515, a Des Moines-based youth organization that empowers youth to pursue professional careers at their *Day of Innovation*. However, we did not attend their event due to conflicts and paper-work issues for the event. We recognize that talking with the demographics that are using a product or service are important to gauge feedback, so we were disappointed that we were unable to participate.

5.7 SECURITY TESTING (IF APPLICABLE)

Given that this is an interactive installation, direct security threats may be minimal. However, to ensure the safety of the system and its users:

- The software will be tested for vulnerabilities, especially if there's any data collection or network connectivity.
- The hardware components will be assessed for safety standards to prevent electrical hazards.

5.8 RESULTS

Results from the various testing phases will be thoroughly documented, with emphasis on how they ensure compliance with the project requirements. Visual aids like graphs, tables, and flow diagrams will illustrate the testing process and outcomes. The narrative will conclude with an assessment of the design's alignment with its intended purpose and requirements.

6 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

6.1 AREAS OF RESPONSIBILITY

Area of Responsibility	NSPE Definition	Corresponding Code of Ethics
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence	IEEE: As engineers we have to maintain high standards in all areas of our own work. The IEEE and NSPE definitions are very similar the only difference is that IEEE also includes ethical conduct in addition to professional conduct. I. To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	IEEE: As engineers we must follow national and international laws and regulations, and not take bribes, and not to cheat customers or take advantage of them. The IEEE and NSPE definitions are very similar the only difference is that NSPE emphasizes the deliverance of products and services at a reasonable cost. I4. To avoid unlawful conduct in professional activities, and to reject bribery in all its forms.
Communication Honesty	Report work truthfully, without deception, and are understandable to stakeholders.	IEEE: We as engineers must be able to give and receive fair criticism, correct any issues that arise, be truthful about data and information, and give credit where credit is due. The IEEE and NSPE definitions are very similar

		however IEEE goes more into detail about the manner in which engineers need to hold one another accountable. I5. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others;
Health, Safety, and Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	IEEE: We as engineers need to respect safety, privacy, well-being, and the environment. The IEEE and NSPE definitions are very similar however IEEE includes more that just stakeholders as it addresses people and environment in general. I1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;
Property Ownership	Respect property, ideas, and information of clients and others.	IEEE: We as engineers cannot do anything that will cause physical, mental, or reputational harm to others. The IEEE and NSPE definitions are very similar but the difference is that IEEE also includes the reputation of others. I9. to avoid injuring others, their property, reputation, or employment by false or malicious actions, rumors or any other verbal or physical abuses;

Sustainability	Protect the environment and natural resources locally and globally.	IEEE: We as engineers need to have empathy for the public. We must think of ethical design processes. We must disclose information that users or consumers should know. The IEEE and NSPE standards are very similar however one difference is that NSPE focuses primarily on the environment and IEEE focuses on general safety, health and welfare of people and environment. I1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment;
Social Responsibility	Produce products and services that benefit society and communities.	IEEE: We as engineers need to help society in general understand the technologies we are creating. The IEEE and NSPE definitions are very similar however IEEE focuses on not only producing products that benefit society but also educating society on said technologies. I2. to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

Table 3: Areas of Responsibility

6.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

For each of the professional responsibility area in Table 1, discuss whether it applies in your project's professional context. Why yes or why not? How well is your team performing (High, Medium, Low, N/A) in each of the seven areas of professional responsibility, again in the context of your project. Justify.

Area of Responsibility	NSPE Definition	Project Application
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence	High – This is our first major project as students so we are following all the procedures and guidelines to implement our project. We are constantly getting are project checked by faculty members to ensure work quality is being met and the highest standard of professionalism is being upheld. It is important for us to have integrity to build our reputation as engineers in the workplace.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Low – We are not really making a sellable product, nor are we working with a large budget so everything will be within a reasonable cost. Our project is also more research based than us implementing a sellable product. We still need to be mindful and reasonable with the cost of items for our project.
Communication Honesty	Report work truthfully, without deception, and are understandable to stakeholders.	High – We are working in a team on this project so our communication skills are paramount to our success in this project. We must be honest about our work with one another and also be honest with our faculty advisor, and the general public. We must not create deception in our work to pass it off as more interesting to the general public or more/less complex to our advisors. We must be truthful

		in communicating our goals and issues with all audiences.
Health, Safety, and Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Medium – Health and safety is always important. However our project isn't really dangerous or harmful to stakeholders. We will create and implement our project in a smart and safe manner that ensures the safety of not only our team members but also the general public. When we make the physical model of our city we will make sure that all safety standards are being met.
Property Ownership	Respect property, ideas, and information of clients and others.	High– We will respect the ideas of our fellow teammates and faculty members. We will respect the input from the general public on our project. We will also give credit where credit is due throughout the implementation of our project by siting the sources we used to get inspiration.
Sustainability	Protect the environment and natural resources locally and globally.	Medium – Our project encompasses sustainability of smart cities by finding ways to improve urban infrastructure. Our research is mainly about this however when we actually make our model we don't really have to worry about it being sustainable because of it's small size. Our model city will not cause much harm on the environment.
Social Responsibility	Produce products and services that benefit society and communities.	High – The purpose of this project is to produce a model city that can be displayed publicly to inform the general public about ways to improve urban infrastructure. We want to produce a high quality product that can convey this to the general public.

Table4: Specific Professional Responsibility Areas

6.3 Most Applicable Professional Responsibility Area

The most applicable Area of Responsibility for our project is Social Responsibility. The entire purpose of our project is to create an interactive public display that educates the public about ways to improve urban infrastructure. Our interactive model city will allow us to convey this information to the general public.

7 Closing Material

7.1 Discussion

Discuss the main results of your project – for a product discuss if the requirements are met, for experiments oriented project – what are the results of the experiment, if you were validating a hypothesis – did it work?

Our project, Engineering Art, asked us to select a problem from 14 Modern Problems that Engineers Currently Face and to implement it in an artistic manner the utilizes and showcases all of our engineering majors. We selected Ways to Improve Urban Infrastructure, and decided to represent it in an artistic manner by creating an interactive STEM tool; although our main target audience is children and students in grades K-12, the design can be used by all age groups. During this project we made use of the double diamond process. In iteration 1, we were able to create a small prototype of our vision for this project. Currently we are working on the iteration 2 by improving the first iteration prototype. We were unable to test and experiment with the K-12 students as our intended audience, however, we were successful in meeting our requirement in testing the components individually, both on software and hardware end. Additionally, we were able to test and confirm the intercommunication between the hardware and software of our prototype. Lastly, we were able to produce a semi-functioning interactive tool. Our design does not meet all the feature requirements we intended to; however, it provides a general idea of our vision.

7.2 Conclusion

Over the two semesters that we have worked on this project, we have realized the uniqueness of this project compared to the others assigned in different senior design groups. While other groups were given simple straightforward projects with sturdy requirements and standards, we were handed a project with near-unlimited possibility and potential. This project allowed us to explore our strengths and weaknesses as both engineers and creatives. After completing this

project, we agree that this is almost similar to a Senior Design Industrial Design project. Selecting each part of the project made us feel like real designers, crafting the most perfect user experience to get the most out of what we have. We wanted to deliver a unique installation to educate the youth about engineering, and we wanted to pursue a different route than a simple tablet application. We wanted to deliver the hands-on experience that engineering offers in a fun and creative way that bridges boundaries and is fun for all ages. We did this by leveraging creative software engineering and hardware design in conjunction with learning traditional skillsets such as woodworking and painting to meet the functional requirements of building an application that educates children. While working on this project, we worked with many tools that we had never previously used. For example, we had to learn woodworking to meet our functional requirements for an aesthetically pleasing design. The decision to undertake new disciplines such as these was intended to elevate our design beyond a simple project, but rather an interdisciplinary deliverable that reflects our stakeholders' and audience's needs and behaviors.

7.3 REFERENCES

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

Luna Optoelectronics, "CdS Photoconductive Photocells," PDV-P9200 datasheet, Oct. 2016

M. Sato, I. Poupyrev, and C. Harrison, "Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects," Disney Research, May 2012.

8 Appendices

Appendix 1: Operation Manual

Parts needed:

1. **Tablet device running the latest version of Android**
2. **Android Application APK Installed on Device**
3. **Arduino Uno with compiled software**
4. **The four conductive surfaces**

Setup:

1. Unload the conductive surfaces and place them next to each other, allow a bit of space between the surfaces to ensure best performance and sensing.
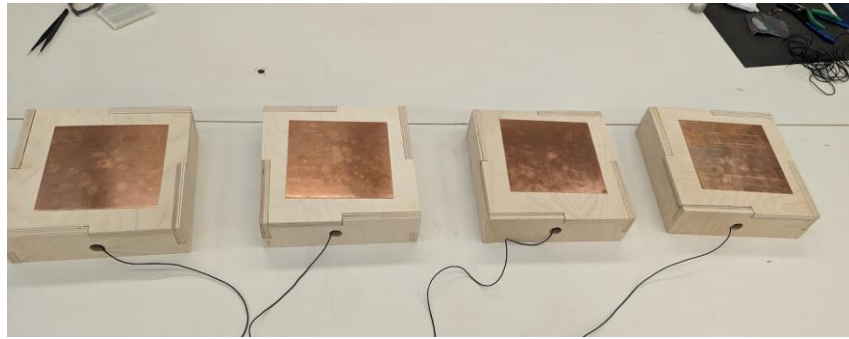
2. Ensure that the surfaces are correctly connected to the Arduino via the circuit board, it is ideal to have each connection correspond to the correct input, so it is less confusing and less wires will cross in the process. These current wires are soldered directly onto the board. This board is designed to go straight onto of the Arduino using the pin layout, so there is minimal work needed to complete setup.
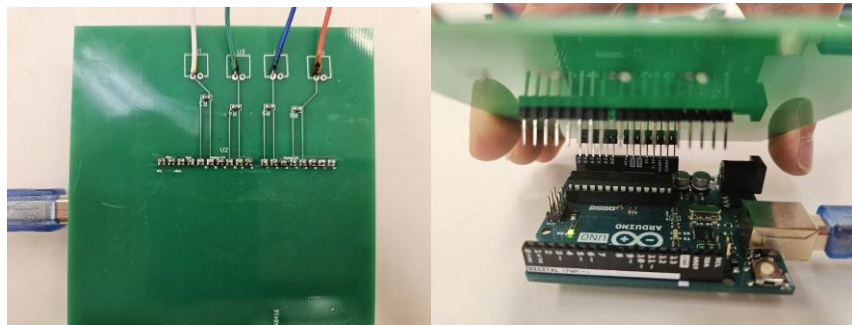

3. Connect the Arduino to the tablet with the installed application by using the USB-B to USB-C cord. The USB-B port is found on the Arduino and the USB-C port is found on the tablet. This connection enables connection between the tablet and the Arduino, enabling the transfer of data such as when the conductive surfaces are touched
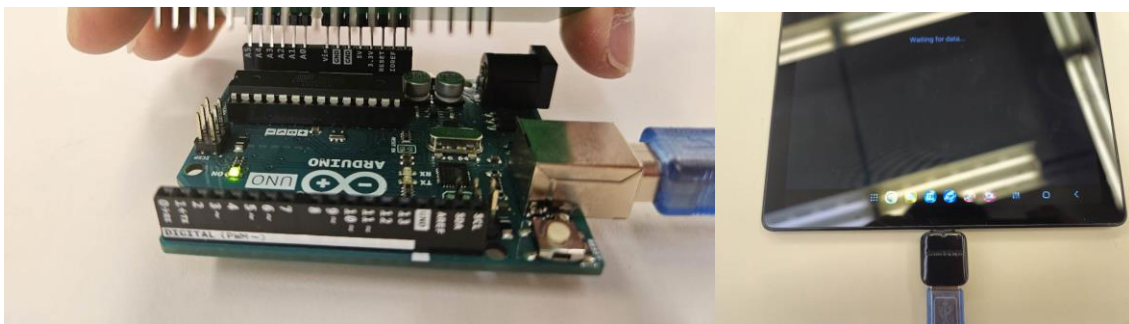
4. Once you do this, you can turn on the application using the tablet. When you first open the application, you should have the cord plugged in, per the previous step. You should be prompted to allow device access. Click accept.

5. Once you accept, click the start USB button at the top to start the game. You should see that the game has started, and you can interact with the application using the conductive surfaces
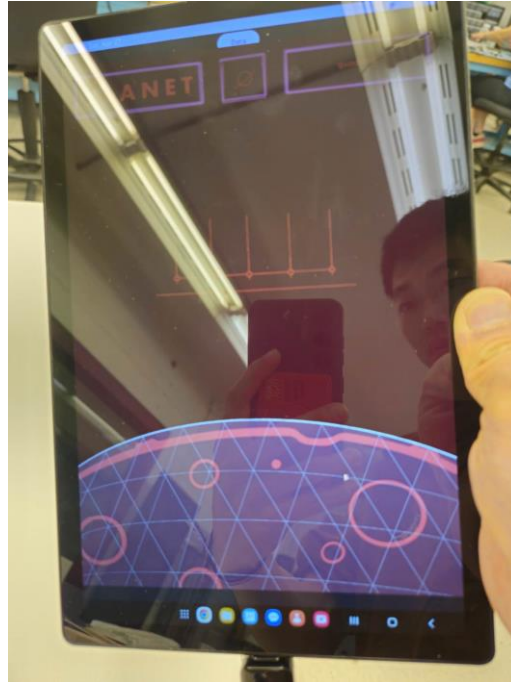
Appendix 2: Previous Iteration

Appendix 2.1: Previous Iteration Design Context

1. Physical layout of the model city will be constructed on a 1 meter by 1 meter (this size may change as the project progresses) square tile made up of scanner conductive material that is able to read the input of the figures that are placed on it and send that information to the computer monitor to visualize the city digitally. This technology is very similar to the Skylanders/Nintendo technology.

2. The models that will be placed on the 1 meter by 1 meter square tile will be made by using SolidWorks and 3-D printer. We want to tackle to art aspect of this project by constructing a model city and to design the aspects of the model city (buildings, roads, power stations) by using SolidWorks to generate layout of the specific components. Then use a 3-D printer to print out those model components. The specific types and number of those model design components has not been decided yet. We chose SolidWorks because it is a widely used CAD software that we have access to here at Iowa State and can learn to operate easily. We would then 3-D print those designs and will attach material to them that allows

the 1 meter by 1 meter scanner conductive material grid to understand where everything has been placed.

3. We will use Python as our primary coding language for the software aspect of this project. Python is a easy coding language to use that a majority of our group members know how to use. This language was primarily chosen due to personal preference over other coding languages. We will need this software to properly analyze our model city and project it onto the monitors.

Appendix 2.2: Previous Iteration Functionality

Pictured below is our 1st Iteration design which we were implementing during the first semester. Our design was creating a model city by 3D printing different model buildings that have a specific amount of holes in them. These blocks would be placed over a photoresistor circuit which would detect and quantify the amount of light that passes through the holes in those buildings. This information would then be sent to a computer where we would have a block rendering of our model city and that information that was gathered by the photoresistor circuit would inform the user about where a specific building is located. There would also be additional fun facts that pop up on the computer as the user cycles through placing different buildings on the different photoresistor circuit grid spaces in our model city. This allows us to create a fun, interactive, artistic, and informative way to teach kids ages k-12 about ways to improve urban infrastructure which is the goal of this project. The purpose of this project is to create some kind of interactive art piece that can inform kids k-12 how to improve urban infrastructure. This is done by allowing the user to place the interactive building blocks over the photoresistor circuits in any combination of their choosing and that information will be sent and displayed to our computer model of our city which will display fun facts about ways to improve that specific buildings infrastructure. The image below shows a sketch of what our design plan looks like. In section 4.6.1.1 we have described the progress we made on this 1st Iteration, so you can reference that section for further information. This design would have met all of our function and non-functional requirements if we had worked on it until completion. However, as mentioned previously we shifted from this iteration to our 2nd iteration
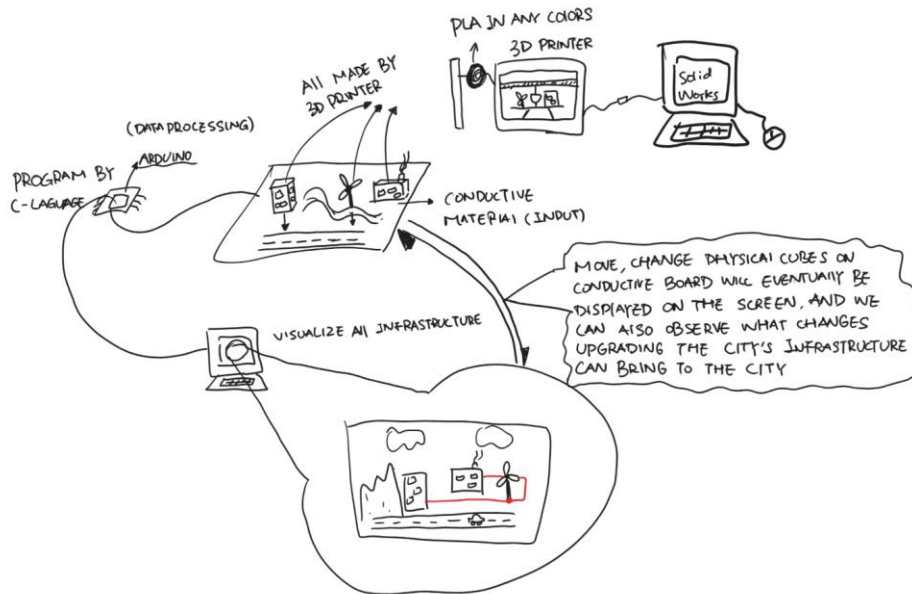
Figure 4.1: Interactive Urban Infrastructure Visualization Concept

We have experimented with the detection of blocks using photoresistors, where we detect the absence of light which signifies that a spot is being covered up. A small demo detailing the direction of our design and project will be shown at the faculty panel presentation. This demo will show the essential aspects of our design at work. These aspects are the detection of blocks and the communication with the frontend computer that shows the current and correct state of the system, which is the current layout of the blocks. We will be showing this demo with a single grid space and various tiles. We have experimented with the detection of blocks using photoresistors, where we detect the absence of light which signifies that a spot is being covered up. A small demo detailing the direction of our design and project will be shown at the faculty panel presentation. This demo will show the essential aspects of our design at work. These aspects are the detection of blocks and the communication with the frontend computer that shows the current and correct state of the system, which is the current layout of the blocks. We will be showing this demo with a single grid space and various tiles.

The primary change to the previous design was the removal of the conductive fabric and switching to a grid based system for sensing and detecting buildings being placed. After researching how we could use conductive fabric to detect where and what kinds of buildings were being placed down, we deemed it infeasible, as we don't have much time to figure out the complicated way that could work. We created an alternative, cost effective way that would be easier to understand. This new

design involves a grid with spots where you can place tiles (buildings) in. Each grid cell has 4 photo resistors on each corner of the grid cell. Each tile has a different shape and they cover a different number of photo resistors. By detecting which photoresistors are covered and which ones are not, we can derive which building is being placed, or there is no building being placed at all.

To reduce complexity of the project, we also have opted to make the buildings that are physically placed into tiles that are easy to store and manufacture. This change allows the system to be stored and transported with less effort in a form factor that a child might be more familiar with. While we are shifting our representation of the buildings from the board and focusing more on the display, we are still intending to make the tiles light up when placed on the grid. We plan to achieve this effect by utilizing the hall effect and magnets. Each tile will have a hall effect sensor paired with an inexpensive cell battery. Each grid space will have a magnet that will trigger the hall effect sensor. When the tile is placed on the grid the internal circuit will light up the LED from the power of the cell battery embedded inside the tile.

We also discovered that there were not many game engines that support Python, so we have decided to steer away from python except for extremely simple testing of basic communication protocols such as serial communication between two systems.

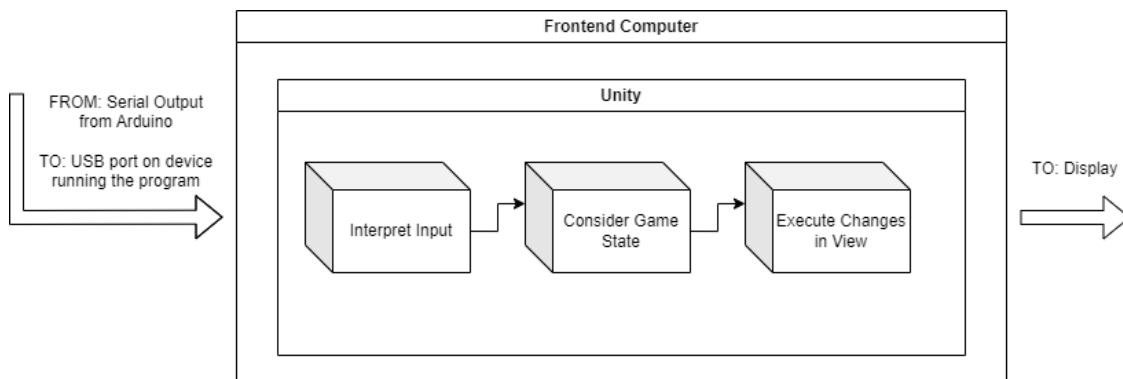*Appendix 2.4 Design Visual and Description*

Figure 5: Arduino- Unity Interaction Flowchart

Using Unity allows us to run our logic and visualization on many different types of hardware, not just a single machine that you must maintain and keep. This download can be publicly available on GitLab or GitHub so even when we leave the university, it can be continued to be studied and used.
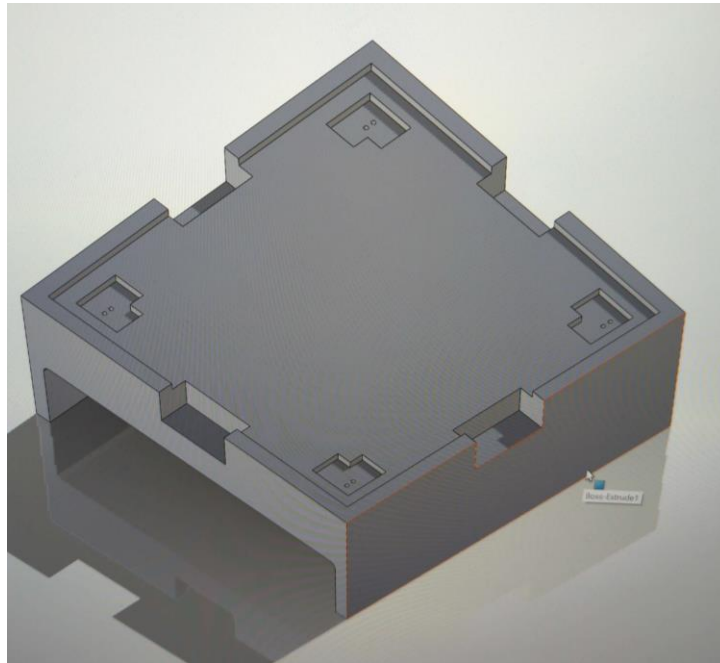
Figure 6: Single Grid Space 3D design

This is the first iteration of a single grid space. The final version of our grid will have many of these connected to each other. Each grid space is exactly 5x5 inches, with space for a 4.58 x 4.58 inch tile that has it's corners shaped according to the type of building the tile represents. The grid has 4 spaces for photoresistors that are adjusted so they aren't too embedded so enough light can hit them, preventing false detections of covering. There are holes for the pins of the photoresistor to go through and connect to a PCB which will feed to an Arduino. This is where the analog signals will be interpreted and delivered to the frontend display via a serial connection using USB.
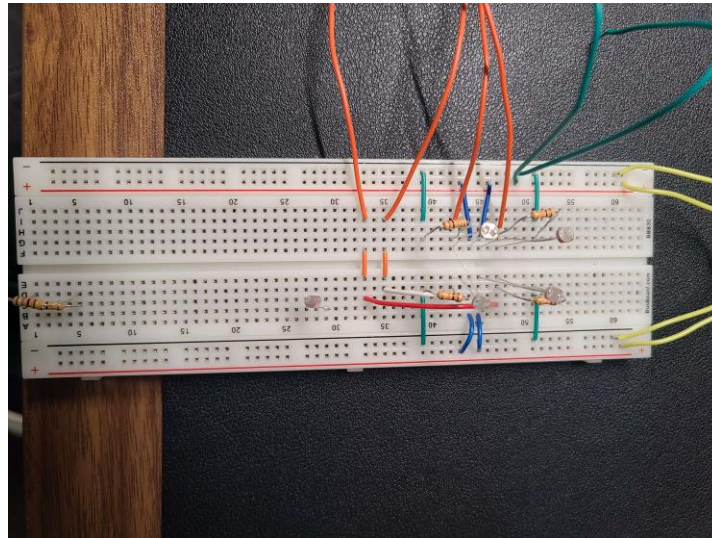
Figure 7: Unit Photoresistor model detection device

These are the internals of the demo being showcased, with the 4 photoresistors being the spots that would be covered up by a tile.
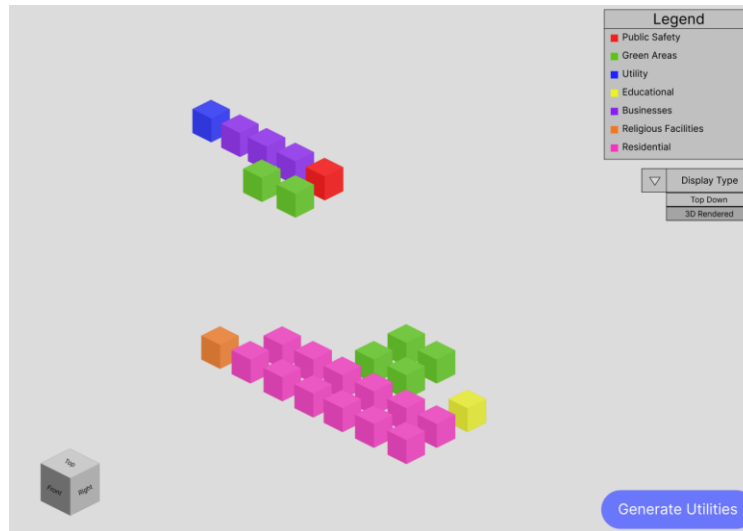
Figure 8: Rendered Output on The Display

This is an example of the rendered output on the display. Right now, each building is represented by a single-colored block, but we want to have each building look like how they look in real life. You can see the task that is prompted to the user in the bottom right corner. This gives a goal and incentive to keep playing around with the architecture and see how well you can build a great town.

We chose to use Unity for the frontend visualization of buildings being placed down and game logic. The reasons Unity was selected can be divided into two categories – function and non-functional reasons.

Functionally, our project requires a real time method of representing objects in a 3d environment that reflects user input (I.e. block placements on the playmat). These specific requirements steered us towards using a game engine as developing these tools from the ground up on a different platform would not only be difficult and time consuming for us to develop, but it would be nearly impossible to maintain. Game engines have a large number of internal tools with exhaustive documentation that lets anyone that is familiar with the engine platform unpack and understand the game logic and perform changes. Additionally, game engines allow the easy export of projects so they can be executed on a wide variety of different machines with different architectures. This feature allows our project to be more portable and not require a specific computer or operating system to run the game. For example, it can be run on a Linux desktop attached to a monitor just as well as a Macbook or a Windows tablet. As long as the hardware has a display and the correct I/O (in this instance a USB port) you can connect the playmat and have the system functional.

To create the hardware components we used a standard breadboard with normal wires, resistors, and photoresistors that were acquired from ETG. The design was fairly simple and different resistor values could be used in place of the ones we implemented. We choose to implement our design using light sensing technology (photoresistors) due to the fact that we could implement multiple units with ease and have it be portable and even adjustable within the model city itself. By detecting which building was placed on a tile based on the amount of light that passed through the holes in the design of said building we could ensure functionality of our design by operating it in a well lit environment. Other designs that we were thinking about implementing were too complicated to implement in the time that we had. So we focused on light sensing to determine what building was placed on a specific tile.

The circuit design was fairly simple as well and the Arduino we used was the Arduino Uno R3 and we used that model simply because one of our group members had them. We technically could have used any Arduino model however we used the Uno R3 because it was available to us at the time. We chose to make our circuit on a breadboard rather than a PCB because we were still in the prototype phase and wanted to be able to play around with components in the circuit which could not be done if we made a PCB model. This allowed us to make mistakes and improve the circuit design freely.

We chose to implement the buildings for our model city by design them on SolidWorks and then 3D printing them. We chose to do this because our team is familiar with the software and we could also design many different versions of each building which helps with the artistic component for our project. By 3D printing the parts we were able to streamline the building process rather than trying to make it by hand.

Appendix 3: Team Contract

Team Name sdmay24-17

Team Members:

1) Karandeep Sandhu

2) Alexandra Whitmer

3) Grant DeWaay

4) Zheyuan Zhang

5) Austin Krekula

6) Seyedehbahareh Hashemimovahed (Bahar)

Team Procedures

3. **Day, time, and location (face-to-face or virtual) for regular team meetings:**

   Thursday from 10 a.m. to 11 a.m. in the Student Innovation Center.

4. **Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**

Discord and Cisco Webex and we also have everyone's phone numbers in case we need to text one another.

5. **Decision-making policy (e.g., consensus, majority vote):**

We will do a majority vote as we want to make sure that everyone's voice is heard and we take everything into consideration.

6. **Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**

We will all be responsible for tracking our own individual progress and how much time we are spending on the project. As for team meetings we have someone in our group (yet

to be determined) take note of what was discussed during the meetings. We will also be using GitLab and google docs and Cybox to store our documents and project material.

Participation Expectations

7. **Expected individual attendance, punctuality, and participation at all team meetings:**
   a. We expect everyone to attend the all the meetings, however we have noticed that not everyone can meet during our work week meeting time due to conflicting schedules so we will schedule meetings on the weekend if need be. However we will keep group members who cannot make to the meetings in the loop by constantly communicating via Discord and Webex. Obviously if there are any conflicts that arise for group members we can adapt and keep them in the loop via discord and webex. We expect everyone to be respectful to one another and participate equally in this project process. we should not have a few group members do all of the work while others do very little and slack off. We expect everyone to do what was assigned to them in a timely manner however if issues come up we as a group will adapt.
8. **Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:**
   b. We expect everyone to do what was assigned to them in a timely manner however if issues come up we as a group will adapt. We plan on hitting deadlines regularly so we don't fall behind.
9. **Expected level of communication with other team members:**
   c. We make use of discord and webex and expect everyone to check the discord at least once a day to ensure nothing unplanned has popped up and everything is moving as scheduled. If something unexpected pops up and needs to be handled immediately then we will communicate via text or groupme as those might generate faster responses.
10. **Expected level of commitment to team decisions and tasks:**
    d. As mentioned above we will use a majority vote to decide what needs to be done and expect everyone to participate in generating ideas, providing feedback and voting. We have a team advisor who will help guide us so they might tell us the direction we need to go but all team decisions will be made by the members.

Leadership

11. **Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):**
    e. Everyone will participate in our team and we all expect that as tasks arise we will distribute them to meet everyone's specific skill set and availability. We have listed everyone's skill sets below. Our client is our advisor so we will meet with them on weekly basis.

     f. Artistic and creative direction that includes non technical elements will be primarily handled by Grant DeWaay in addition to anyone else on the team that expresses interest in contributing to this aspect of the project. This role may transfer to anyone else in the group if there is overwhelming distaste of the ideas being presented, or the practices being implemented. Elements that fall under this category include the following:

    i. Shape and form

    ii. Color palette

    iii. Materials used for non technical elements

12. ex: the enclosure

    iv. Presentation

    v. Messaging

13. NOT including the selection of 1 of the 14 engineering challenges

14. NOT including changing project scope for the sake of greater artistic expression unless agreed upon by the entire group

15. **Strategies for supporting and guiding the work of all team members:**

    g. As mentioned before we have team members from all four of the engineering fields (computer, electrical, software, cyber) and we will use our individual skill sets to help each other succeed. We will try to divide the work fairly and also in a way that allows everyone to work with something they have experience in. If issues arise then we will help each other figure them out.

16. **Strategies for recognizing the contributions of all team members:**

    h. During our weekly meetings we will all touch base and share our individual progress and will be able to see how much work someone is doing by tracking their progress on their tasks.

Collaboration and Inclusion

17. **Describe the skills, expertise, and unique perspectives each team member brings to the team.**

Zheyuan: Matlab, AutoCAD and some simple C language.

Alexandra: Java, frontend development, and some hardware experience.

Karandeep: Circuit design, Matlab, Cadence Virtuoso, and some Verilog and simple C.

Grant DeWaay: Wide array of programming languages along with modeling tools like Solidworks and Grasshopper 3D and manufacturing techniques like 3D printing, sewing, embroidery, and laser cutting. Experience with double diamond design model and design thinking.

Austin Krekula: Security oriented design thinking, information security, risk management framework, Linux OS security, Windows OS security, networking security, applications of cryptography, discrete mathematics, Python, C, Java, Verilog, assembly language

(MIPS & x87 instruction sets), software reverse engineering, digital forensics, embedded systems engineering, microprocessor design, digital logic, algorithm analysis.

Bahar: Java, some simple C, some embedded system and VHDL.

18. **Strategies for encouraging and support contributions and ideas from all team members:**
i. Establish an open communication environment, encouraging team members to share their thoughts and opinions.
j. Ensure everyone has an opportunity to speak.
k. Strengthen trust and collaboration among teams through long-term cooperation.
l. Ensure everyone understands their tasks and expectations.
19. **Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)**

Our team discord is a space for any thoughts and ideas. Please feel free to share if something is making you uncomfortable or unable to complete a task. These messages should be taken without criticism, and we can redirect our team from there.

Goal-Setting, Planning, and Execution

20. **Team goals for this semester:**

Have our project planned out and reach the step where we begin ordering parts. Have a good design document made and do well on the presentation to receive a good grade (A) for the semester.

21. **Strategies for planning and assigning individual and team work:**
m. We want to distribute work in a fair manner so that everyone does a relatively equal amount of work and we also want to assign that work based on everyone's strengths as mentioned above.
22. **Strategies for keeping on task:**
n. We will use our weekly meeting time to go over progress and make sure everyone is contributing and staying on task. For individual strategies that will vary from person to person so encourage all team members to work in a way that works for them without jeopardizing the team.

Consequences for Not Adhering to Team Contract

23. **How will you handle infractions of any of the obligations of this team contract?**

We will reach out to the person to check in. If the person is unresponsive, we will divide the task among other team members and consult with our faculty advisor or TA if need be.

24. **What will your team do if the infractions continue?**

We will set a time for a group team meeting where we will discuss group conduct and resolve issues. If that does not work we will inform our TA or instructor and go from there.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*

b) *I understand that I am obligated to abide by these terms and conditions.*

c) *I understand that if I do not abide by these terms and conditions, I will suffer the*

consequences as stated in this contract.

1) Karandeep Sandhu	DATE 9/10/2023
2) Alexandra Whitmer	DATE 9/10/2023
3) Grant DeWaay	DATE 9/10/2023
4) Zheyuan Zhang	DATE 9/10/2023
5) Austin Krekula	DATE 9/10/2023
6) Seyedehbahareh Hashemimovahed	DATE 9/10/2023

Appendix 4: Code

Our code for our Android application and the implementation of Disney's Touché can be found online within the ECpE Github at [sd / sdmay24-17 · GitLab (iastate.edu)](#). Within this repository are also some images that we used during our first iteration for laser cutting and visual design.