TagGang Final Proposal

https://accellarando.github.io/TagGang/

Ella Moss
Computer Engineering
University of Utah
Salt Lake City, UT
ella.moss@utah.edu

Dana Escandor
Computer Engineering
University of Utah
Salt Lake City, UT
u1167876@umail.utah.edu

***Abstract*—TagGang is a digital art and engineering project that allows users to collaboratively create a physical canvas of art using modern technology. The project is inspired by the graffiti concept of tagging, where artists work together (or against each other) to create artwork and establish territory. It also draws inspiration from Reddit's /r/place, a collaborative web-based art project that took tagging to the digital realm. TagGang enables users to draw images through gestures via an Xbox 360 Kinect, then choose a location. These images will be drawn on a wall-mounted canvas at that location using a pen plotter. Stretch goals include support for multiple colors and an erase feature. TagGang's goal is to enable collaborative (or even competitive) artwork via a cyber-physical system, while demonstrating computer engineering skills including circuit and hardware design, embedded systems programming, software development, and project management. The end result is a system that invites individuals to come together and create something beautiful.**

***Index Terms*—CNC, vertical paint-pen plotter, GUI, skeletal tracking, art, graffiti, tag, graffiti-bot, G-code, interactive, collaborative**

I. Introduction & Motivation

TagGang is a digital art and engineering project, inspired by the collaborative nature of graffiti and the success of Reddit's /r/place project [6]. In creating a final Computer Engineering capstone project, the team wants to showcase how beautiful the field can be by creating a work of collaborative, interactive art. TagGang's graffiti-bot is their implementation of such a system.

TagGang allows users to draw images in the air via an Xbox Kinect. The system tracks the user's arm movements and gestures to capture and display a drawing. After the drawing is complete, users will specify a small location to plot it on a wall via a traditional graphical user interface (GUI). Finally, a pen-plotter will realize this drawing on the wall. The end result will be a collaborative canvas of art, where multiple users can add their unique contributions to a larger work of art.

In addition to these primary components of the system, the project also has a stretch goal of implementing support for multiple colors. This stretch goal would require support for a tool-changer on the pen plotter, and extensive thought on user-interface implementation details. Even though implementation may prove to be difficult, multiple colors would add a level of visual complexity that adds to the final artwork. Another user-interface stretch goal is an eraser feature, so that users can edit their drawings before sending them to the pen plotter.

TagGang will provide a unique and immersive art experience while also demonstrating computer engineering principles, containing original hardware and original software components. In the end, engagement and collaboration between users is key to making TagGang successful.

II. Influences & Inspirations

TagGang stems from the idea of the interactive capabilities prevalent within the aesthetics of art: fusing select influential aspects from graffiti and Reddit's r/Place.

Graffiti is a form of art through the placement of writings/drawings on walls or other surfaces in public spaces. In graffiti culture, there is an extensive community of graffiti artists who mainly create on an individual level. The use of tags—a stylized signature—separate and distinct these artists from on another and within our project, we loosely define this as a simple drawing. TagGang's name is play-on-words representing the community brought together, like a gang, through the interactive act of putting up a tag—a pun on "tagging".

The website of Reddit had a subreddit called r/Place that acted as an online collaborative art project. The project involved a blank canvas that was divided into individual pixels, which users could color in. Each user was allowed to place one pixel every few minutes to create their own images or to collaborate with others to create larger works of art. TagGang gives individuals their own small amount of space within the set confines of the canvas just like r/Place's use of pixels.

Both graffiti and r/Place allows for individualistic artistic expression within a community that TagGang wants to imitate. However, a community does not guarantee respectful collaboration amongst each other. In graffiti, "going-over" refers to the act covering up one another's tag or street art with one's own. Likewise, r/Place users have the opportunity to go over other user's pixels in order to create their own designs or to sabotage others. TagGang reflects this collaborative nature of conflict by giving TagGang's users the freedom to do such. There is no social experiment to see if users will or will not chose to draw on other's tags.

TagGang is not the first graffiti-bot and we take inspiration from others out there. The GTGraffiti and Hektor graffiti-bots utilize vertical plotters with a cable-pulley system driven by stepper motors that is also found in TagGang [2] [27]. Both inspirations have their respective motives with similarities to TagGang's, so the interactive aspect we have placed distinguishes ourselves from these graffiti-bots.

Fig. 1. "Hektor Titles a Show, Jürg Lehni, 2008 Design And The Elastic Mind, MoMA, New York"

III. Project Tasks

Project tasks will be further elaborated and explained to fully describe how TagGang expects to function and the various aspects found in TagGang's graffiti-bot.

A. User Interface

The user interface will consists of three tasks: "Graffiti Simulator", "Tag Placement Selector", and "Semi-autonomous Tagging" that flow into one another.

In "Graffiti Simulator", as Kinect sensor input is gathered through several SDKs on the Raspberry Pi, it is then modified by a specialized script to send appropriate data in generating G-code. This script deals with necessary pixel and coordinate conversions of the user's tag. Alongside selected placement coordinates in having the plotter tag the canvas as part of the "Tag Placement Selector". Then there is a Python script that reads these G-code instructions and sends them over a serial port to the Arduino Uno to be interpreted with libraries. The Arduino Uno and the custom motor shield/CNC controller provides the interface between the microcontroller and plotter's components as part of the "Semi-autonomous Tagging". These tasks will be broken down further into its respective aspects that comprise it and elaborated on in the following sections.

1) Kinect Hardware & Software: Microsoft's Kinect is motion-sensing input device using an array of cameras and microphones to detect and track a user: it utilizes a combination of depth sensing, skeletal tracking, and voice recognition to capture and interpret the user's movements and voice commands. Being originally developed for the Xbox gaming

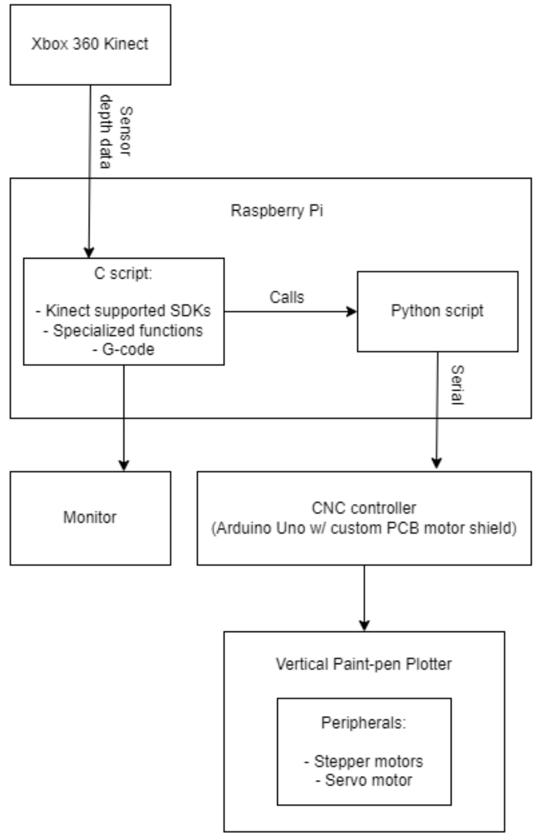

Fig. 2. TagGang system block diagram

console [30], the Kinect for Xbox 360 (model 1414) was the first version from the Kinect product line and the motion-sensing device primarily used for skeletal tracking used in TagGang.

TagGang's Kinect sensor was not intended for use outside of its Xbox console [31] and so some changes to various aspects may be necessary in order to work for our project. In setting up for its intended use, there is proprietary connector cable—called the "USB/Power Cable"—that connects the Kinect sensor and Xbox console together. This cable has a unique 12-pin connector on one end that plugs into the Kinect sensor, a standard USB 2.0 connector on the other end that plugs into the Xbox 360 console, and an AC adapter to supply power to both devices [31].

Alongside that, Microsoft officially discontinued all Kinect sensor manufacturing in 2017 [32] and acquiring a Kinect would have to be through a third-party seller. Due to the nature of Xbox console's being packaged with Xbox-specific Kinect sensors and the large sale numbers of these consoles, there are more of Xbox-specific Kinect sensors readily available on the market and in turn cheaper. Which is why TagGang makes use of the Xbox 360 Kinect sensor over the Kinect for Windows, despite Kinect for Windows being a more optimal choice for TagGang's premise—as it meant for interfacing with an OS machine.

This also means the Kinect is outdated technology, so any official drivers and libraries are no longer supported. Even if we obtained Kinect for Windows, its SDK would also be incompatible with the lack of updates for modern OS machines [33]. TagGang works around this by making use of an open-source SDK, OpenKinect libfreenect—containing drivers and libraries for the Xbox Kinect device on Windows, Linux, and macOS [34]. The libreenect has other SDKs built on it that we would be utilizing to fully set up the software portion in obtaining Kinect sensor data: decisions on what other SDKs to use for our final construction of TagGang are still being made.

2) Skeletal Tracking: Skeletal tracking is key to TagGang, as it constitutes the majority of the user interface. However, earlier experiences in the prototyping phase of the project demonstrated that skeletal tracking (especially on older hardware such as the Xbox 360 Kinect) is not perfect - the data that comes back from the sensor tends to be noisy, jittery, and imprecise. These findings warrant further experimentation with software, and potentially hardware, in order to build a robust system.

The team plans to use a Raspberry Pi as the central device that communicates between the Kinect sensor and pen plotter. Linux drivers and libraries exist for this sensor, but no documentation exists for running this software on a Raspberry Pi [4]. Thus, one of the first tasks must be to evaluate the feasibility and performance of integrating this hardware and software.

This task also includes custom software that interfaces with the Kinect sensor. This software will receive input from a Kinect device, and output an array of 3-dimensional coordinates for the left hand and right hand in the form "(L_x, L_y, L_z):(R_x, R_y, R_z)". This string can be passed to other parts of software as needed.

Rather than passing along raw coordinate data from the Kinect sensor, software should perform pre-processing on the data in an effort to reduce noise. Further experimentation is needed to determine what kinds of de-noising measures will be effective and performant. Smoothing and filtering are two methods that may prove useful and have already been implemented for the Kinect [5].

3) Gesture Recognition & Support: TagGang's Kinect sensor is capable of tracking the movement of hands and fingers to some extent, but it does not provide accurate or reliable finger tracking unlike its updated counterpart—Xbox One Kinect. It can detect the position and movement of the hands and fingers to some degree: though it cannot provide highly-detailed information about the shape or orientation of individual fingers. For this reason, gestures within TagGang will be limited to hand movement detection through the use of limb isolation within skeletal tracking. In polishing TagGang's system further, implementation of gesture recognition is used as indication in whatever means within the graffiti-simulator of the user interface.

At a minimum, the graffiti-simulator user interface should account for user movement to draw, user movement without drawing, and user finished. Outside of that includes gesture recognition of different stretch goals—which is explained in detail in its respective section. It is important to have discernible gestures for these various user movements as it will utilized in system's states that include perceiving the user, starting the graffiti tagging process, and ending the tagging process.

From experience with TagGang's prototype, the specific gestures chosen is more convenient to implement in terms of the data gathered before (i.e., real-time generated depth values represented by the z-coordinate has been calculated). Therefore, the gesture of moving the user's right hand out in front of them at arm's length away indicates user movement to draw—calling this gesture "punching". Likewise, the gesture of pulling/having the right hand back to the user's chest indicates user movement without drawing. The same "punching" gesture with the left hand will indicate the user has completed their tag.

TagGang's Kinect automatically skeleton tracks a user when in range and the user movement without drawing gesture should be synonymous to this initial state. Any gestures to a specific state will stay in said state until another recognized gesture is made. Users will have 30 seconds to draw their tag once the right-hand "punch" gesture has been made: it will send in the user's tag to the "Tag Placement Selector" regardless of whether or not the user was done. This will ensure tags are done in a timely manner to avoid a long wait line and more chances for people to try our graffiti-bot.

4) Preview & Rendering: Visual feedback is necessary when drawing a picture, especially with an interface such as the Kinect. To provide this feedback and allow users to view their drawings, TagGang will display drawings in real-time on a monitor. Data from the gesture-recognition Kinect software will be sent to the preview software, which will plot incoming coordinates on a plane.

Once the image is completed, users may elect to accept the drawing, or clear it and start over—being part of the "Graffiti Simulator" part of the user interface. This will advance them to the next screen on the GUI, allowing them to select a location.

5) Coordinates Selection: Due to inaccuracy and noise in the Kinect data, we have chosen to implement a traditional mouse-driven graphical user interface (GUI) to allow users to select a location for their drawing. If de-noising efforts produce good results, TagGang can instead implement a drawing placement interface via the Kinect. This would produce a seamless, integrated experience for the user, but if it doesn't work well it could instead become a source of frustration. To minimize this risk, the group prefers a traditional GUI, but is still open to implementing this interface over the Kinect if possible. This aspect will be the "Tag Placement Selector" of the user interface.

This stage should also show a rendering of the entire artwork so that users can decide where to place their artwork in relation to other pieces on the wall. This allows for a more collaborative and cohesive final creation. This extra context allows users to make more informed decisions about where to place their artwork to best enhance the composition as a whole. Users will also be made aware about TagGang's aspect of overlapping tags.

B. Vertical Paint-pen Plotter

The vertical paint-pen plotter is a machine that uses a cable-pulley system to control the movement of a paint-pen suspended from above. The motion of the paint-pen is coordinated by the controller that generates G-code instructions based on input from the Kinect sensor. It is responsible for physically drawing the graffiti tags onto a wall canvas in a timely manner—being the "Semi-autonomous Tagging" task of TagGang's user interface.

This semi-autonomous aspect of TagGang's graffiti-bot means the plotter can operate autonomously to some degree, as it requires the collective data of the user's input from both the "Graffiti Simulator" and "Tag Placement Selector" user interface tasks (i.e., once the user has created the tag and selected its placement, the graffiti-bot starts tagging without user intervention outside of a fail-safe event).

1) Controller: A Computer Numerical Control (CNC) board controller is the "brain" of the vertical paint-pen plotter. Interpreting digital signals from the G-code instructions based on the Raspberry Pi's Kinect input: converting them into physical motion via the suspended cable-pulley system driven by stepper motors and actuator of the paint-pen holder.

TagGang CNC controller will utilize a custom Printed Circuit Board (PCB) to act as the motor shield for the Arduino Uno microcontroller in controlling the various motor and switch peripherals. Specifications of the custom PCB such as the motor driver, power regulation/requirements, and protection components are yet to be decided.

Part of the peripherals includes two stepper motors that accurately controls the cable-pulley movement of the paint-pen holder in two directions—consisting of one stepper motor for the x-axis direction (horizontal movement) and the other for the y-axis direction (vertical movement). Also including the actuator attached to the paint-pen holder that moves the paint-pen to make contact with the canvas—a servo motor controls the rotating mechanism to move upwards or downwards. These motor peripherals are chosen in conjunction each other in the cable-pulley system to be able to withstand the load of one another in order to smoothly operate.

The peripherals also include limit switches to prevent the paint-pen holder from moving beyond a certain point of its travel range and ensures that it remains within the limits of the plotting canvas. When triggered, it signals the controller to stop the motion in that direction.

2) Hardware Setup: The hardware setup of vertical paint-pen plotter includes the frame, cable-pulley system, paint-pen holder, and CNC controller. Most parts within this setup are 3D-printed as it has the advantages of custom designs and is optimized for weight reduction without sacrificing structural integrity. TagGang plans to reference Penelope, a similar vertical plotter with 3D printed parts, as a guide in constructing the frame and cable-pulley system [28]. As for the paint-pen holder and actuator mechanism, the gondola-like design of MTvplot will be utilized—Penelope's pen holder takes inspiration from this design as well [29].

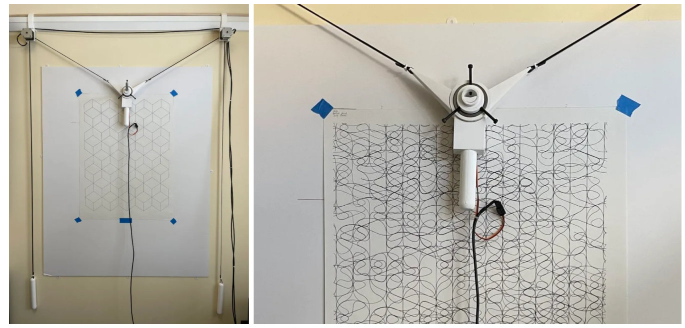

Fig. 3. DavidBliss's Penelope: vertial pen plotter with gondola pen holder design similar to MTvplot

The frame consists of mounts that provides a stable base for the machine. A singular bar will stretch horizontally across a wall—held up by mounting tape—that will have two attachments housing the stepper motors and the cable-pulley system. The limit switches on the frame are placed at the top and bottom of the paint-pen's vertical travel range to serve as end stops. This frame is meant to be portable with the use of mounting tape as a temporary adhesive.

The cable-pulley systems consists of several toothed cables and counterweights creating a pulley-like system that is threaded through various parts of the plotter. Toothed cables provide precise control over movement of the system to ensure accurate and repeatable motion. This toothed cable is threaded through the two attachments held by the frame that also contain a gear and small bearing: these inside parts help in applying pressure to the toothed cables and keep it in constant contact with the teeth of the pulley counterweights.

It is also threaded through the paint-pen holder's gondola arms, wrapping around the paint-pen and actuator mechanism. The actuator mechanism will use the servo motor with an extended arm that rotates to either push the paint-pen holder off the canvas or let it fall onto the canvas.

The counterweights help balance the load of the paint-pen holder and keep it taut: without counterweights, the weight of the paint-pen holder could cause the cables to sag, resulting in inaccurate paint-pen positioning. There are three counterweights; one for each stepper motor as part of the pulleys and the final being for the paint-pen holder—being hollow to hold some type of measured weight that can be added or removed as needed for the right load.

3) Software Generation: After the "Graffiti Simulator" and "Tag Placement Selector" tasks of the user interface, the Raspberry Pi generates the appropriate G-code from the Kinect data. The G-code would typically include information on the paint-pen's position, speed, and other parameters, as well as instructions for moving the paint-pen to specific coordinates on the paper. All said information is then sent to the controller of TagGang's plotter, which would then translate them into the physical motion of the paint-pen on the vertical plotter.

TagGang will utilize the G-code library, GRBL, which is primarily designed for Cartesian coordinates. The use of a cable-pulley system means polar coordinates should be used. Modification the GRBL library to convert polar to Cartesian requires some basic trigonometry knowledge. Configuration to use GRBL includes appropriate settings for TagGang's plotter; the number of steps per revolution for the stepper motors, the maximum speed and acceleration values, and the pin assignments for the limit switches and the servo motor.

C. Stretch Goals

In addition to the primary objectives of the project, TagGang has also designated several stretch goals. These goals are suggestions of design improvements and new features that can be developed on top of the main functionality, time and other resources permitting. This section of the proposal document outlines the various stretch goals and associated tasks.

1) Multiple Color Support: Having color support in our graffiti-bot is the stretch goal TagGang placed highest priority on. We believe it reflects the colorful artistic aspect of graffiti and allows a greater creative opportunity for users—ideally, having a ROYGBIV color range. We reference the DIY Pen Plotter with Automatic Tool Changer as a guide [3]. For multiple color support, an interchangeable tool rack holding different colored paint-pens would be 3D printed and placed alongside the plotter. G-code instructions should be able to map specific paths to these colored paint-pens and allow some mechanism to retrieve and switch to any paint-pen during "Semi-autonomous Tagging". Also implementing specific gestures to change colors during the "Graffiti Simulator" as many times the user desires and support some sort of interface to preview what colors can be selected.

If for whatever reason the multiple color support cannot be implemented, TagGang could change the single-color paint-pen to a multi-colored paint-pen that has different colors in separate sections of the paint-pen tip. It would be a little less to implement than the original multiple color support concept. Where the color-selector GUI is still planned but modification to the paint-pen holder and G-code instructions would rotate the paint-pen to a different section of color.

2) Eraser User Interface: An eraser tool meant for the "Graffiti Simulator" part of the user interface should allow the user to either undo their last stroke or precisely remove a section of their tag at any time before the finished tag gesture. To implement this, a new hand gesture could be made or the user would move their hand to a specific position within the Kinect window to select this tool. Scripts should be in place to account for any erasure before G-code generation so that plotter incorrectly draw the tag. However, this stretch goal is more of an afterthought to maintain realism of graffiti's permanent art: in real-life, artists cannot back-track once ink is laid down.

3) Telematic Mode: Since TagGang is a collaborative and interactive project, the concept of telematic art could be implemented. Telematics refers to the use of computer networks to transmit and receive information over a distance. An application could allow users that are not present for TagGang's physical demonstration to see and contribute in drawing their tags to our graffiti-bot. This would be additional logistics like application design and GUI, network connection, queues, different G-code drawing instructions, camera system, etc,. It could provide more flexibility to users who do not want to physically wait to draw, though these logistics would require more time than other stretches goals and is significantly harder to implement with additional risks.

IV. Schedule & Milestones

To manage these tasks, the project has been broken down into eight major milestones, which each have smaller sections and subtasks to be accomplished. These milestones each have deadlines associated with them to ensure the project stays on track. This section will detail the milestones and individual tasks for TagGang. Refer to Table I for a visual representation of all tasks and milestones in the project.

A. Group Management & Communication

The TagGang project is managed by a two-person team. Both individuals have worked on projects together in the past: from experience, understanding of each others' capabilities and work ethic has solidified how this team operates. Communication is made through various, available mediums such as texting, calling, instant messaging social platforms, and meet ups.

In-person and online meet ups can be established as necessary, which are then recorded on TagGang's website via meeting logs that follow a specific format. This format is distinguished by meet up dates and contains information of meeting notes, assessment of current progress, action items, next meeting expectations, and accomplishments since the previous meeting. Alongside other logistics such as time/duration and member assignments. Some of the following milestones can be worked on concurrently and the group will manage how that will go.

B. Milestone 0: Prototype - April 25

1) Summary: The first milestone - the prototype phase - has already been accomplished. This prototype, demonstrated on April 25th, took data from a Kinect sensor to drive a stepper motor. The prototype consisted of a waving hand based on the Xbox 360 Kinect, an Arduino, and a NEMA17 stepper motor.

2) Description & Operation: Rather than developing entirely new software for the prototype, existing libraries and examples were modified to meet the prototype's goals. A Linux personal computer acted as the main controller for the project, enabling communications between the Xbox 360 Kinect and the Arduino (which connected to a stepper motor). Refer to Fig. 4 for a basic block diagram of the prototype.

Skeltrack is an open-source Linux library for skeleton tracking with the Kinect, which already had some usage examples that could be easily adapted to the prototype's purposes.

TABLE I
PROJECT MILESTONES

Number	Depends on	Description	Assigned to	Due Date
0	-	Prototype	Both	April 25
1	-	Custom PCB designed, other hardware ordered	-	July 14
1.1	-	Order parts	Ella	July 14
1.2	-	Plotter schematic	Dana	July 1
1.3	1.2	PCB design	Ella	July 14
2	-	User interface	-	Sept 22
2.1	-	Kinect support	Ella	Sept 8
2.2	-	Tag Placement Selector	Dana	Sept 8
2.3	2.1, 2.2	Integration	Both	Sept 22
3	1	Hardware assembly	-	Oct 27
3.1	1	PCB assembled/soldered	Dana	Sept 29
3.2	3	3D printed parts printed	Ella	Sept 29
3.3	3.1, 3.2	Pen plotter assembled	Dana	Oct 20
3.4	-	GRBL modifications	Dana	Oct 20
3.5	3.3, 3.4	Testing and documentation	Both	Oct 27
4	2	G-code translator	Dana	Nov 3
5	2, 3, 4	Integration and Testing	Ella	Nov 10
6	5	Stretch Goals	Both	Dec 1
6.1	5	Multiple color support	-	Nov 22
6.1.1	4	G-Code support	Dana	Nov 17
6.1.2	2	Kinect/UI support	Ella	Nov 17
6.1.3	6.1.1, 6.1.2	Integration	Both	Nov 22
6.2	6.1	User interface eraser support	Both	Dec 1
7	5, 6	Demonstration	-	Dec 8

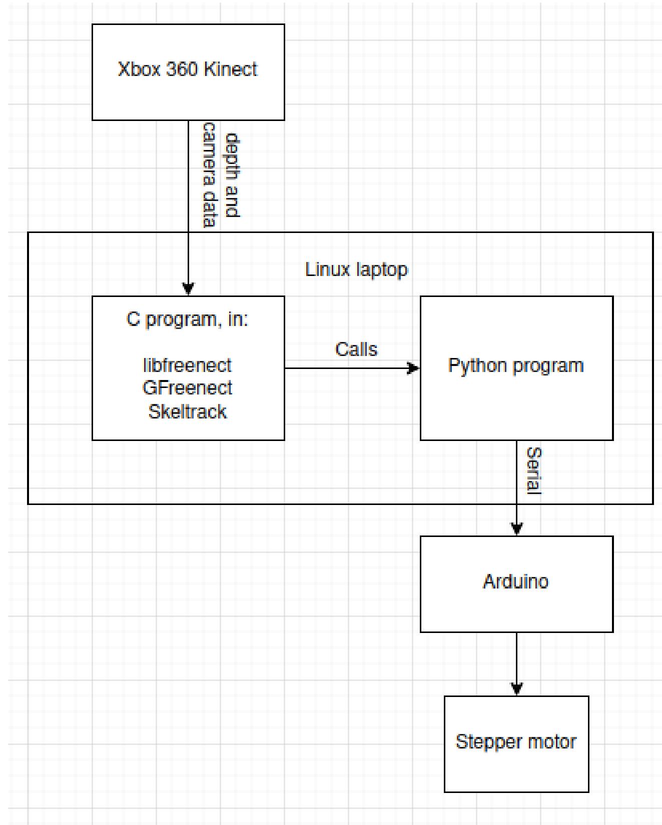

Fig. 4. Prototype block diagram

This example code was modified to track the user's right hand position, then calculate changes in the hand's horizontal position and send new position data to the Arduino, which drove a stepper motor to that position. The Skeltrack examples were written in C, but for portability and ease of use, stepper motor commands were sent via a Python script over a serial port to the Arduino. The Arduino was connected to a stepper motor driver which controlled the stepper motor.

3) Results & Challenges: This prototype demonstrated that it is still possible to receive usable information from the Kinect sensor, despite its age and deprecated software. The team also learned how to control a stepper motor from a personal computer via an Arduino, which is an integral part of TagGang's pen plotter.

The prototype stage had its fair share of issues. Due to the Xbox 360 Kinect's age, the Skeltrack repository hasn't been updated in over ten years and relied on some older software (such as GFreenect and libfreenect). Building these shared libraries so that the Skeltrack examples compiled correctly proved to be difficult, but once the Makefile was configured correctly, compiling the rest of the program was relatively trivial. Fortunately, detailed guides exist online for building and compiling Skeltrack examples [7].

As mentioned previously, the prototype also demonstrated noise in Kinect data. Further research is necessary to determine the best way to denoise this data. Some algorithms have been written to smooth and denoise Kinect data [5], but integrating these algorithms into Skeltrack's existing codebase may be nontrivial. Other skeletal tracking libraries also exist that have not yet been evaluated for noise levels, such as Sen5or's Gestures repository (also built on libfreenect) [8]. Other repositories also exist that rely on OpenNI's skeletal tracking solution [11] [12].

Compatibility with Raspberry Pi hardware is another open question. A Raspberry Pi is a general-purpose computer capable of running a lightweight Linux distro, so the software would theoretically be capable of running [9]. Although the capability seems to be there, evidently the Raspberry Pi hobbyist community has not explored this functionality outside of speculation [10]. This may be another open question that will need to be answered in the final project.

Kinect aside, issues came up in the prototype stage involving the chosen stepper motor driver. The purchased stepper motor driver shield - the X-NUCLEO-IHM05A1 - has Arduino UNO connections, and a GitHub repository was discovered that appeared to support Arduino hardware [13][14]. However, upon compilation it became evident that although the driver was written in Arduino code, it was actually meant for an STM32 device and was not compatible with the Arduino.

After reading the data sheet for the L6208 H-bridge at the heart of the device, it was determined that several pins need to be set to different logical values for the stepper driver to work. The data sheet also specified an internal finite state machine on the driver, which required eight pulses on the "step" pin for a full stepper motor step [15].

Although this stepper driver chips has many useful features, these quirks made it somewhat harder to use than, for example, the A4988 or DRV8825 chip series. The L6208 was recommended because it may run cooler than other options that are prone to overheating [16].

C. Milestone 1: Custom PCBs & Other Hardware Ordered - July 14

The first milestone involves ordering parts from the Bill of Materials, including the first iteration of a custom stepper motor driver board with a microcontroller. This has been broken down into smaller tasks that are delegated to each team member.

The first task is to order parts from the Bill of Materials, listed in the appendix. Sourcing for these parts is not expected to be an issue, except for the Raspberry Pi. Ella's employer, InnoSys, may have some spare Raspberry Pi devices that the team can use; otherwise, Ella has subscribed to stock notifications. Other orders should be able to be placed within a day, and will then arrive within a few weeks (but we will allow up to 2 months in case of shipping delays, part failure, or other risks). It is important to do this step first so that components arrive in time. We expect to start assembling hardware in mid-September, so these parts should be ordered by July 14. Ella will perform this task.

The custom stepper motor driver CNC PCB has four main phases: schematic, layout, manufacture, and assembly. Dana will design the schematic by July 1 so that the next stages can occur on time. This schematic will include three stepper motor driver ICs and required peripheral components as stated in their data sheets, as well as a microcontroller and those required connections [2]. It will include all components and connections that will need to be realized in the final PCB. Other features, such as thermal protection, short-circuit protection, and reverse polarity protection may also be desirable to prevent damage to the PCB and other components.

Ella will be responsible for the PCB layout stage. This stage realizes the schematic into a physical layout, placing components appropriately, and routing traces between components as needed. Form factor is also a consideration at this stage - we want a small form factor that can easily interface with a Raspberry Pi. Raspberry Pi devices have 26 or 40 GPIO pins, depending on model, that allow users to easily extend the Pi's functionality with a "HAT" - Hardware Attached on Top [17]. Because TagGang relies on a Raspberry Pi, it makes sense to design a CNC board in the Pi HAT form factor for maximum compatibility. This HAT form factor has a complete specification, with design guides and other resources, on Raspberry Pi's GitHub [18]. Other guides have also been written that may be more user-friendly [19], as neither team member have designed a Pi HAT before. Trace widths will need to be calculated based on the stepper motor specifications to safely and efficiently handle high currents that will be passing through the board. Finally, once the prototype board has been designed, it will need to be ordered and manufactured. This should be designed and ordered by July 14 through OSHPark, and is expected to arrive by August 25 or so.

Therefore, all parts of Milestone 1 should be completed by July 14, not including the time needed for components to ship. All components should be ordered by this point, including the custom PCB.

D. Milestone 2: User Interface - September 29

The second milestone involves programming the user interface - Kinect and traditional GUI included.

1) Kinect Support: Ella will develop support for the Xbox 360 Kinect. The Raspberry Pi should be able to receive data from the Kinect by July 28, either via Skeltrack or some other library - further research is required. This integration was nontrivial during the prototype stage, so this step will be done early to mitigate risk.

By August 25th, the Raspberry Pi and Kinect should have smooth and reliable gesture tracking, including recognition for all supported gestures (listed in III.A.2). As with the previous task, getting gestures to register reliably (and without too much noise) may prove to be challenging. This task will involve researching and deciding on an option that provides the best results, then implementing it.

Once gestures can be reliably tracked and recognized, they will need to be integrated with other software to actually draw a picture on the screen based on incoming gestures. This task will be done by September 8th.

2) Tag Placement Selector Interface: Dana is responsible for for the "Tag Placement Selector". This will resemble a traditional graphical user interface (GUI) that is driven by a mouse. After the drawing has been completed, the user can use a mouse attached to the Raspberry Pi to select a square on the screen for the picture to be drawn on. This selected coordinate, along with the drawing made by the user, will both be inputs to the G-Code translator (implemented in Milestone 4).

This interface can be developed concurrently with the Kinect support task, so this is also scheduled to be done by September 8th.

3) Integration: Once the GUI and Kinect user interfaces have been developed and tested, the two pieces of the user-experience must be integrated. This will result in a complete user interface where users can draw a picture with the Kinect, complete the picture, then select a coordinate for the picture to be drawn on. The picture will not actually be drawn yet at this stage, but the user-experience part should be done. This integration stage, as well as documentation and testing, should be done by September 22. Because both team members contributed to the parts of the system that are to be integrated, both members will collaborate on this task.

E. Milestone 3: Plotter assembled - October 27

The third milestone requires a total time of 5 weeks to complete. After schematics are made, the hardware construction of the vertical paint-pen plotter will be assembled and wired up to the rest of TagGang's system. Specific G-code instructions from the planned user interface will not be possible yet, but we are able to still test the plotter and custom PCB in the CNC controller will operate with simple G-code GRBL examples.

1) PCB Soldered: After Ella provides the blank PCB and electrical components, Dana will solder the necessary electrical components—following the PCB schematic—that satisfy the specifications in creating the PCB for the CNC controller. At the very most, this process should take 1 week: this accounts for any mistakes made while soldering as well as constant testing to avoid any electrical mishaps.

2) Plotter Assembly: The completed custom PCB CNC controller/motor shield attached to the Arduino Uno. Ella will create custom 3D-printed parts given by Dana's plotter design specifications which will take 1 week. The design includes the frame's bar and attachments for the stepper motor and toothed cable housing, paint-pen holder, and counterweights. Then Dana will assemble the plotter's hardware setup and connect it to the rest of the TagGang system in a temporary environment within 1 week. Modifications to GRBL for polar plotting should be ready in 1 week within the given time frame and before testing of the assembled plotter. The temporary environment will allow basic testing of the plotter which is conducted by both members and subsequently documented, taking 1 week.

F. Milestone 4: G-code translator - November 3

Dana will write the G-code script to control and drive the plotter via the CNC controller and Arduino Uno, taking a total time of 1 week. Within the Raspberry Pi, the appropriate coordinates sent from the Kinect and SDK script will be recorded as the input to be utilized in the G-code instructions. The G-code should already have tested function GRBL examples with converted polar-to-Cartesian coordinates, it will be further implemented with TagGang's full system in mind now. The G-code will also set up the configurations needed to control all aspects of the plotter. The finished G-code script should be sent to CNC controller and Arduino Uno to allow for testing of the TagGang's system in the following milestone.

G. Milestone 5: Integration & Testing - November 10

Testing will occur during and after each milestone to ensure that pieces of the project work as expected. However, integrating different parts into a final product is sometimes difficult and leads to unexpected behavior. For these reasons, the team has designated Milestone 5 for complete integration, testing, and documentation. In this stage the user-interface, G-Code translator, and pen plotter all come together to realize the final TagGang project. This should be done by November 10 to allow time for stretch goals.

H. Milestone 6: Stretch goals - December 1

Stretch goals are only implemented after TagGang's base function is operating correctly and smoothly. As aforementioned, the multiple color support is the stretch goal with the highest priority. In the case that we cannot fully implement it 1 week before demonstration, we will shift our stretch goal priority to an eraser tool.

1) Multiple Color Support: The base plotter design Dana created should allow the addition of a color rack at any time. Ella will then design and 3D print the paint-pen color rack and update the user interface to support this within 3 weeks. Dana will be updating the G-code script to interpret different colors paths, plotter tool changing, and configuration of the plotter's paint-pen holder within the same 3 weeks.

2) User Interface Eraser: Similar to the multiple color support plan, Ella will update the user interface and Dana will update the G-code as necessary with the specific logistics of the eraser tool. Considering this takes less time to implement, whatever knowledge we gain from the color support stretch goal we can apply to this stretch goal.

I. Milestone 7: Demonstration - December 8

This milestone marks the imaginary date of demo day, the TagGang graffiti-bot should be ready for demonstration, we ensure the entire project is completely tested and documented.

V. Risk Assessment & Mitigation

Various issues will be inevitable in the construction of TagGang and bringing awareness of specific issue we might encounter could reduce the likely-hood—if not prevent it—from happening in working to its completion. Acknowledgement of every concern will not be possible before the beginning of TagGang's implementation, but general assessments and their mitigation we can predict are as listed in this section. As a general guide, we want to prevent any extraneous costs, but are open to spending as needed.

A. Component Sourcing & General Design

Acquiring all of TagGang's components could prove to be a substantial difficulty as issues could be out of our control and/or delay our progress a significant amount of time.

Following the prototype build, schematic design and choice of components for the paint-pen plotter and PCB are key in how efficient TagGang will be constructed. Due to manufacturing and shipping times, we have to be confident enough in how we plan to construct these aspects of TagGang as error in either could force us to redo the schematics and/or component selection at any point of time of TagGang's construction—setting us back and compensating from any degree of severity.

Uncontrollable instances of components being Dead on Arrival and component shortages; like TagGang's intended Raspberry Pi unit currently backlogged due to production and supply-chain issues [24]. We can only prepare for unavailable components ahead of time by reconsidering different options and these changes consequently affect surrounding aspects of TagGang's system. For example, in place of a Raspberry Pi unit, a OS machine (PC/laptop) would be utilized like our prototype and—without getting too involved—adjustments would be made to both software and hardware.

Avoidable instances of flawed schematic design and improper component configurations; like inadequate PCB layout, poor soldering, and the possibility of generating electrical overload in stepper motor burnout or a fried PCB. These instances can be circumnavigated by having attentiveness beforehand. Extensive research and thorough understanding of their own respective expected functionality and integration with the TagGang system as a whole would optimize time and resources, but we cannot guarantee the correct implementation on the first iteration.

B. Paint-pen Plotter Operation

Outside of the vertical paint-pen plotter generalities already mentioned, building this machinery has challenges of its own. The plotter's suspended layout requires understanding of the stepper motor's constraints and limitations so that all motors operate as discussed without any mechanical stress. Utilizing cables instead of a two-rod system (CNC-like plotter) reduces load capacity on the torque of the stepper motors. Referencing our stepper motor's data sheet requires knowing how much weight the paint-pen 3D-printed holder and cables are to then apply the maximum load capacity formula provided—we need to keep this in mind before constructing the paint-pen plotter until we know these specifications.

There are several potential issues involved with generating the G-code. We have to make sure that there is not any compatibility issues between any part of the TagGang system, modification to the software is expected to function properly. Likewise, depending on how accurate the data from the Kinect, SDK, and specialized scripts are for G-code input could lead to accuracy problems: replication of the tag, via the plotter's cable-pulley system, might be less precise than desired. Additionally, there is incorrect configurations like calculation of coordinate conversions from polar-to-Cartesian or stepper motors and driver specifications in the GRBL library.

C. Outdated Technology

In conjunction with component sourcing, Microsoft officially discontinued all Kinect sensor manufacturing in 2017 [25] and acquiring a Kinect would have to be through a third-party seller. Due to the nature of Xbox console's being packaged with Xbox-specific Kinect sensors and the large sale numbers of these consoles, there are more of Xbox-specific Kinect sensors readily available on the market and in turn cheaper. Which is why TagGang makes use of the Xbox 360 Kinect sensor over the Kinect for Windows, despite Kinect for Windows being a more optimal choice for TagGang's premise—as it meant for interfacing with an OS machine.

Because TagGang's Kinect sensor was discontinued, This also means the Kinect is outdated technology, so any official drivers and libraries are no longer supported. Even if we obtained Kinect for Windows, its SDK would also be incompatible with the lack of updates for modern OS machines [26]. TagGang works around this by making use of an open-source SDK, OpenKinect libfreenect—containing drivers and libraries for the Xbox Kinect device on Windows, Linux, and macOS.

Learning from TagGang's prototype, there were multiple obstacles and issues related to how our Kinect sensor and libfreenect SDK—and other SDKs mentioned in libfreenect's build instruction—operate together. The libfreenect SDK was made available over a decade ago and any relatively recent updates do not guarantee modern-day function. This was made apparent when we were not able to run the SDK on a Windows OS due to outdated build instructions; however, the Linux OS ran. Therefore, a Raspberry Pi unit with Raspbian OS—a Debian-based OS based on the Linux kernel—acting as our sole machine in TagGang mitigates different OS problems. Additionally, there are some inaccuracies within the libfreenect SDK skeleton tracking—stutter and noise—that could be a risk in how smoothly the "Graffiti Simulator" task in our user interface operates. We can mitigate this through debugging software and fixing values but we are not too concern with perfecting line-tracking. For example, the slight stutter within skeleton tracking could be construed as imperfect line-work or smoothed out in post-process. Ultimately, figuring out if the Kinect will work properly is set early in our milestones.

In the case that the Kinect will not be in our complete favor, switching to a different motion-capture device like the Intel RealSense or the Xbox One Kinect V2 could be an option. Changing motion-capture devices is also an immense responsibility, as even the other options are also discontinued and would provided similar risks to our current device.

VI. Materials

This section of the proposal lists complete bills of materials and a list of potential vendors to buy parts for TagGang. Bills of materials are separated into which part of the project they're for - controllers and connectors (Table III), the CNC board (Table IV), or the pen plotter (Table **??**). Each bill of materials can be found in the appendix. Each part has a primary and secondary vendor listed in case of sourcing issues; details for these vendors can be found in Table II.

TABLE II
Vendor List

Name	Address	Website	Description
Uprok	1594 S. State St, Salt Lake City, Utah 84115	uprokslc.com	Store that sells paint markers and other graffiti supplies in Salt Lake City.
Mouser	1000 N Main St, Mansfield, TX 76063, United States	mouser.com	Commercial electronic component parts distributor
DigiKey	701 Brooks Ave S, Thief River Falls, MN 56701, United States	digikey.com	Commercial electronic component parts distributor
Sparkfun	6175 Longbow Dr, Boulder, CO 80301, United States	sparkfun.com	Electronic component parts distributor geared toward prototyping and hobbyists
Adafruit	150 Varick St, New York, NY 10013, United States	adafruit.com	Electronic component parts distributor geared toward prototyping and hobbyists
Lowe's	1335 S 300 W, Salt Lake City, UT 84115, United States	lowes.com	Hardware store
eBay	N/A	ebay.com	Online store
ECE Stockroom	MEB 2355, University of Utah, Salt Lake City, UT 84112		Sells various electronic components to ECE students
OSHPark	311 B AVE STE B LAKE OSWEGO, OR, 97034-3071, United States	oshpark.com	Producing Perfect Purple PCBs Promptly

VII. Resources

The "Resources" section of this project provides an overview of the people, mentors, materials, and tools that are available to the team as they work on their project. This section describes the various contacts and mentors that can lend their assistance, including a professor and experienced electrical engineers. It also lists the tools and materials available in the on-campus labs that the team can use, such as 3D printers, laser cutters, soldering stations, woodworking and power tools, and various hardware and fasteners. This section serves as a helpful guide for the team as they navigate the resources available to them to successfully complete their project.

A. People and Mentors

The team has various contacts and mentors that may be able to lend their assistance in this project.

First is their professor, Erik Brunvand. He is a professor at the University of Utah in the School of Computing and adjunct professor in the Department of Electrical and Computer Engineering. He will be the main mentor for the project as pieces come together, offering support and guidance.

Ella has some contacts at her place of work, InnoSys. They design and manufacture electronics systems. Specifically, she is in contact with Ryan McCandless and Dr. Larry Sadwick, who are both electrical engineers experienced in realizing embedded projects from conception to completion. The team may also be able to use the electronics manufacturing and soldering equipment at InnoSys, although this has not been confirmed yet.

Ella's father is an electrical engineer with extensive experience in embedded systems, who the team can also turn to for support.

B. Materials and Tools

A variety of tools, materials, and other resources are also available for the team to use. First, they have access to the Senior Hardware Lab in the Merrill Engineering Building at the University of Utah. This includes a rosin printer, laser cutter, solder stations, and other miscellaneous tools that will be useful for the project. Some materials are also available in the lab, such as wire, resistors, wood, and plastic.

Lassonde Studios is another on-campus lab that is accessible to the team. In addition to some of the amenities offered by the Senior Hardware Lab, it also includes woodworking and power tools, as well as a Fastenal vending machine to get hardware, fasteners, and similar offerings.

VIII. Demonstration

TagGang will be demonstrated inside a enclosed space/room with the vertical paint-pen plotter against a wall with a large blank paper (roughly 4'x4') as the canvas. After TagGang is calibrated and initially tested to run properly, users be instructed on how to operate TagGang (e.g., explaining user interface, gestures, capabilities, how it all functions together, etc.)

Firstly, users will be directed to stand in range with the Kinect sensor and facing a monitor. The Kinect sensor will accept a specific hand gesture from the user to begin the "Graffiti Simulator" process and will draw their tag with the various tools—gestures to move with or without drawing, color support, etc.—at their disposal. The user's tag is reflected live on the monitor so they can see what they are drawing. The user will be given a time limit of around 30 seconds before the tag drawn is automatically considered done, regardless of user completion. Otherwise, if the user completes their tag before the time limit, another hand gesture is required to stop the drawing process and is considered complete. This drawing is sent to the "Tag Placement Selector" GUI where we can either reset the tag sent in if complications arise (i.e. if the tag is deemed inappropriate or a logistics issue). If there is no complication, we allow the user to set the coordinates they desired and once selected, the vertical paint-pen plotter should recreate the tag onto the physical canvas as part of the "Semi-autonomous Tagging" task.

Besides the initial instructions and rule to not draw anything inappropriate, we will mention that any coordinate chosen will be appropriate—as long as it is in the measured confines of the canvas—and that they are allowed to chose to draw on top of other's tag.

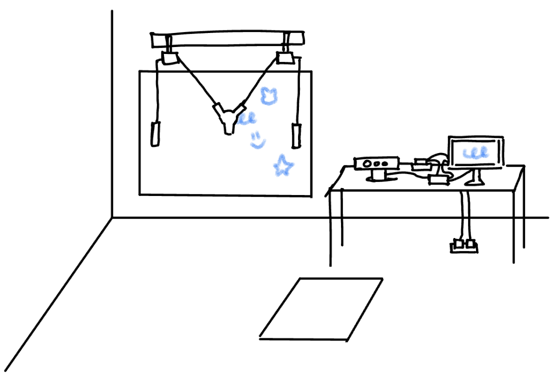

Fig. 5. illustration of preview TagGang demonstration

IX. Conclusion

TagGang is an innovative digital art and engineering project that combines the collaborative nature of graffiti with modern technology to create a physical canvas of collaborative art. Inspired by the concept of tagging and Reddit's /r/place project, TagGang aims to enable users to create artwork using gestures captured by an Xbox 360 Kinect. The project encompasses various aspects of computer engineering, including hardware design, embedded systems programming, software development, and project management.

The primary goal of TagGang is to develop a system that allows users to create artwork collaboratively using gestures and a pen plotter. The project integrates the Kinect sensor to capture user gestures, enabling them to draw, move without drawing, and finish their artwork. Real-time feedback through a preview and rendering system provides users with an immersive experience, allowing them to view their drawings as they create them. Stretch goals include support for color and undoing or erasing brush strokes in the user interface.

The pen plotter, a central component of TagGang, accurately reproduces the artwork on a physical wall. Custom hardware, including a CNC board and stepper motors, is used in conjunction with software that generates G-code from the Kinect data. The pen plotter assembly, incorporating precise guides and limit switches, ensures controlled drawing movements.

TagGang realizes its goals by combining technology and art to create an engaging platform for collaborative artwork creation. The project demonstrates the successful integration of the Xbox 360 Kinect for gesture capture, a user-interface allowing users to preview their art, and a pen plotter for accurately reproducing artwork on a physical canvas. TagGang is a computer engineering project that also provides an opportunity for users to come together and create captivating artwork.

References

[1] grblHAL, "Grblhal/STM32F7xx: Grblhal driver for St STM32F7xx (nucleo-144)," GitHub. [Online]. Available: https://github.com/grblHAL/STM32F7xx. [Accessed: 22-Apr-2023].

[2] G. Chen, S. Baek, J.-D. Florez, W. Qian, S.-W. Leigh, S. Hutchinson, and F. Dellaert, "GTGraffiti: Spray painting graffiti art from human painting motions with a cable driven parallel robot," in 2022 IEEE International Conference on Robotics and Automation (ICRA), 2022. [Online]. Available: https://research.gatech.edu/introducing-gtgraffiti-robot-paints-human. [Accessed: 25-Apr-2023].

[3] Dejan, "DIY PEN Plotter with Automatic Tool Changer: CNC Drawing Machine," How To Mechatronics, 26-Feb-2022. [Online]. Available: https://howtomechatronics.com/projects/diy-pen-plotter-with-automatic-tool-changer-cnc-drawing-machine/. [Accessed: 22-Apr-2023].

[4] Joaquimrocha, "Joaquimrocha/skeltrack: A free software skeleton tracking library," GitHub, https://github.com/joaquimrocha/Skeltrack (accessed May 11, 2023).

[5] Intelligent-Control-Lab, "Intelligent-control-lab/kinect-smoothing: 'Kinect smoothing' helps you to smooth and filter the Kinect depth image as well as trajectory data," GitHub, https://github.com/intelligent-control-lab/Kinect-Smoothing (accessed May 11, 2023).

[6] Lorenz, "Internet communities are battling over pixels," The Washington Post, https://www.washingtonpost.com/technology/2022/04/04/reddit-place-internet-communities/ (accessed May 11, 2023).

[7] T. Naseer, "Installing skeltrack on ubuntu," Installing Skeltrack on Ubuntu, https://tayyabnaseer.blogspot.com/2012/05/installing-skeltrack-on-ubuntu.html (accessed May 11, 2023).

[8] dhaneshp, "Sen5or/gestures: Gesture detection and recognition," GitHub, https://github.com/Sen5or/Gestures (accessed May 11, 2023).

[9] "What is a Raspberry Pi?," Raspberry Pi Foundation, https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/ (accessed May 11, 2023).

[10] tntexplosivesltd, "Board index," Raspberry Pi Forums - Index page, https://forums.raspberrypi.com/viewtopic.php?t=1508 (accessed May 11, 2023).

[11] eddiecohen, "OpenNI/OpenNI: OpenNI," GitHub, https://github.com/OpenNI/OpenNI (accessed May 11, 2023).

[12] antoine1000, "Antoine1000/kinect-skeleton: - processing experiments around skeleton tracking with Kinect," GitHub, https://github.com/antoine1000/kinect-skeleton (accessed May 11, 2023).

[13] "X-NUCLEO-IHM05A1," STMicroelectronics, https://www.st.com/en/ecosystems/x-nucleo-ihm05a1.html (accessed May 11, 2023).

[14] stm32duino, "Stm32duino/X-NUCLEO-IHM05A1: Arduino Library to support a bipolar stepper motor driver based on L6208 component," GitHub, https://github.com/stm32duino/X-NUCLEO-IHM05A1 (accessed May 11, 2023).

[15] ST Microelectronics, "DS2142 - DMOS driver for bipolar stepper motor." [Online]. Available: https://www.st.com/resource/en/data sheet/l6208.pdf. [Accessed: 24-Apr-2023].

[16] Larry, "Alternative to A4988 Stepper Motor Driver?," Electrical Engineering Stack Exchange, 05-Sep-2014. [Online]. Available: https://electronics.stackexchange.com/a/127821. [Accessed: 24-Apr-2023].

[17] John et al., “Introducing raspberry pi hats,” Raspberry Pi, https://www.raspberrypi.com/news/introducing-raspberry-pi-hats/ (accessed May 11, 2023).
[18] Raspberrypi, “Raspberrypi/hats,” GitHub, https://github.com/raspberrypi/hats (accessed May 11, 2023).
[19] J. Buford, “Designing a raspberry pi hat,” Hackaday.io, https://hackaday.io/project/20482-designing-a-raspberry-pi-hat/ (accessed May 11, 2023).
[20] Keith, “How to choose Steppers Motors and controllers for DIY CNC machines,” rcKeith, https://rckeith.co.uk/how-to-choose-steppers-motors-and-controllers-for-diy-cnc-machines/ (accessed May 11, 2023).
[21] Tuenhidiy and Instructables, “Blackboard V-Plotter,” Instructables, https://www.instructables.com/BLACKBOARD-V-PLOTTER/ (accessed May 11, 2023).
[22] R. Weiser, “Custom Arduino Board design,” Developpa, https://developpa.io/custom-arduino-board-design/ (accessed May 11, 2023).
[23] LaskaKit, “LaskaKit/CH340-Programmer,” GitHub, https://github.com/LaskaKit/CH340-Programmer/ (accessed May 11, 2023).
[24] S. M. Sojoudian et al., “Production and supply-chain update,” Raspberry Pi: Production and Supply Chain Update, https://www.raspberrypi.com/news/production-and-supply-chain-update/ (accessed May 10, 2023).
[25] P. Spencer, “Achievement unlocked: 10 Years – thank you, xbox 360,” Xbox Wire, 20-Apr-2016. [Online]. Available: https://news.xbox.com/en-us/2016/04/20/xbox-360-celebrating-10-years/. [Accessed: 13-Apr-2023].
[26] Hickeys, “Kinect for Windows - Windows Apps,” Windows apps — Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows. [Accessed: 14-Apr-2023].
[27] J. Lehni, U. Franke. ”Hektor” juerglehni. [Online]. Available: https://juerglehni.com/works/hektor. [Accessed: 24-Apr-2023].
[28] Davidbliss, “Penelope: A vertical pen plotter,” davidbliss, https://davidbliss.com/2021/09/13/penelope/ (accessed May 10, 2023).
[29] michi-teck, “MTvplot (vertical plotter),” Thingiverse, https://www.thingiverse.com/thing:2371117 (accessed May 10, 2023).
[30] Chen, J. Microsoft Xbox 360 Kinect launches November 4. Gizmodo, 14-Jun-2010. from https://gizmodo.com/microsoft-xbox-360-kinect-launches-november-4-5563148 [Accessed: 13-Apr-2023]
[31] “Connect the sensor to the original Console - microsoft xbox 360 Kinect Manual & Warranty [page 8],” ManualsLib, 23-Mar-2019. [Online]. Available: https://www.manualslib.com/manual/1541058/Microsoft-Xbox-360-Kinect.html?page=8#manual. [Accessed: 13-Apr-2023].
[32] P. Spencer, “Achievement unlocked: 10 Years – thank you, xbox 360,” Xbox Wire, 20-Apr-2016. [Online]. Available: https://news.xbox.com/en-us/2016/04/20/xbox-360-celebrating-10-years/. [Accessed: 13-Apr-2023].
[33] Hickeys, “Kinect for Windows - Windows Apps,” Windows apps — Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows. [Accessed: 14-Apr-2023].
[34] OpenKinect, “OpenKinect/libfreenect: Drivers and libraries for the Xbox Kinect device on windows, linux, and OS X,” GitHub. [Online]. Available: https://github.com/OpenKinect/libfreenect. [Accessed: 24-Apr-2023].

* Primary and secondary vendors, with part number, lead time, unit cost, qty, form factor, packaging * Include other resources you need, ie from the U or other infrastructure * Vendor list: name, address, website, etc

TABLE III
BILL OF MATERIALS - CONTROLLERS AND CONNECTORS

Part	Primary Vendor				Secondary Vendor				Qty	Form Factor
	Vendor	Part No	Unit Cost	Lead Time	Vendor	Part No	Unit Cost	Lead Time		
Raspberry Pi 4 Model B 2GB*	SparkFun	DEV-15446	$45.00	Unknown	Adafruit	4292	$45.00	Unknown	1	Raspberry Pi Model B
Xbox 360 Kinect v1 **	-	-	-	-	-	-	-	-	1	Xbox 360 Kinect
16 AWG wire	Senior HW lab	-	-	-	Lowe's	313146	$6.63	-	25ft	16AWG
Xbox 360 Kinect power/USB adapter **	eBay	274950-400172	$8.59	-	-	-	-	-	1	X360
USB cables, jumper cables, other miscellaneous connectors	Senior HW lab	-	-	-	ECE Stockroom	-	$10.00	-	-	-
									Total:	$70.22

TABLE IV
BILL OF MATERIALS - CNC PCB [22]

Part	Primary Vendor				Secondary Vendor				Qty	Form Factor
	Vendor	Part No	Unit Cost	Lead Time	Vendor	Part No	Unit Cost	Lead Time		
DRV8825 stepper motor driver	Mouser	595-DRV8825-PWPR	$5.81	In Stock	DigiKey	296-29503-2-ND	$5.76	In Stock	3	SMT
ATMEGA328P-PU MCU	Mouser	556-ATMEGA328P-AN	$2.97	In Stock	DigiKey	ATMEGA328P-AN-ND	$2.94	In Stock	1	SMT
CSTNE16M0V-530000R0 crystal	Mouser	81-CSTNE16M0V-530000R	$0.25	In Stock	DigiKey	490-17948-2-ND	$0.25	In Stock	1	SMT
Capacitors (various)	ECE Stockroom	-	-	-	Mouser	-	-	-	5?	SMT, through-hole
Resistors (various)	ECE Stockroom	-	-	-	Mouser	-	-	-	5?	SMT, through-hole
Diodes (various)	ECE Stockroom	-	-	-	Mouser	-	-	-	5?	SMT
USB C connector	Mouser	649-10164359-00011LF	$0.50	In Stock	DigiKey	609-10164359-00011LFTR-ND	$0.50	In Stock	1	SMT
FT231X programmer	Mouser	895-FT231XS-R	$2.34	In Stock	DigiKey	768-1156-5-ND	$2.34	In Stock	1	20-SSOP
TLV75533PDBVR voltage regulator	Mouser	595-TLV7553-3PDBVR	$0.46	In Stock	DigiKey	296-50411-2-ND	$0.48	14 weeks	1	SMT
Miscellaneous LEDs	ECE Stockroom	-	-	-	-	Mouser	-	-	4?	SMT, through-hole
Custom PCB, 5.7 in^2	OSHPark	-	$28.50	4 weeks	-	-	-	-	1	Custom
									Total:	$52.29

* May be able to source from InnoSys/iLumens.
** Already purchased or on hand.

TABLE V
BILL OF MATERIALS - WALL PEN PLOTTER HARDWARE [21]

Part	**Primary Vendor**				**Secondary Vendor**				**Qty**	**Form Factor**
	Vendor	**Part No**	**Unit Cost**	**Lead Time**	**Vendor**	**Part No**	**Unit Cost**	**Lead Time**		
SF2424-10B11 bipolar stepper motor **(1)	Mouser	978-SF2424-10B11	$33.60	In Stock	DigiKey	2300-SF2424-10B11-ND	$27.57	In Stock	3	NEMA17
Servo motor **	Mouser	426-SER0043	$4.00	In Stock	DigiKey	1738-1300-ND	$4.00	4 weeks	1	Servo
5mm timing belt by-the-foot	OpenBuilds Part Store	626-By-the-Foot	$3.49	In Stock	Rubber Belts Online	HTD-5M-15MM	$4.57	In Stock	12	-
3A 24V DC power supply	Mouser	418-TRH70A24-012E03VI	$24.00	In Stock	DigiKey	2034-TRH70A240-01E03VI-ND	$25.47	8 weeks	1	Barrel Plug
Paint markers, various colors	Uprok	-	$5	In Stock	Art Primo	1004665	$3.99	In Stock	7	-
Various 3D printed parts	-	-	-	-	-	-	-	-	-	-