Individual Project Description
Who will you save?

By Lotta Spring
KuMM, WS24/25 LMU

Wörthstraße 34
81667, München
(DE) +49 176 98138086
lotta.spring@remhaug.de

Project-Goal

Our interactive installation **„Who will you save?“** challenges participants with moral dilemmas surrounding medical decisions where another human life is at stake. It highlights how women often lack full control over their own bodies, even when making decisions about themselves.

The installation presents these dilemmas as gradual text prompts on a monitor, allowing visitors to respond using physical buttons. Each participant takes on different roles—a doctor, a relative, and a pregnant woman—over three rounds. Once completed, the collective responses determine the outcome: either the pregnant woman or the child passes away. This interactive experience encourages reflection and discussion on the complexities of bodily autonomy and ethical decision-making, as well putting the responsibility of another humans life into peoples hands.

Concept of the installation and multi-user aspect

The installation is made for three visitors, who take on the roles of doctors, relatives, or a pregnant woman. They answer with yes or no using arcade buttons, and this decides the course of the installation. The users read from lines on a monitor, and there are three different scenarios that the participants get confronted with.

Each person votes yes or no by clicking a green or red arcade button. After three scenes, most of the votes are counted and depending on the users majority of choices a different outcome is shown. In one outcome the heart monitor of the pregnant woman flatlines, in the other one the heart-monitor of the unborn child flatlines - in both cases theres an eerie sound of the monitor flatlining, making the participants have to uncomfortably perceive the consequences of their choices. The multi-user aspect is given by the three roles the users slip into, making decisions as their role. The exhibit can only work when three inputs are given by three different roles: the pregnant woman to represent the person without fully bodily autonomy, the doctor who has to consider medical perspectives and relatives who represent the societal and cultural or religious perspectives that can influence the decisions.

Museum Adaptation and Considerations

The installation is designed to fit well in a museum setting while being quite easy to maintain. It is kid-friendly and not traumatizing, despite talking about death, it avoids graphic or distressing visuals, keeping things simple with heart monitors as symbols instead of anything too intense. To make it more suitable for a museum we wanted to keep the hardware is mostly hidden, and the interaction is straightforward, using clear icons so even those who don’t speak English can still somewhat participate and experiment with it. After each session, the system resets on its own, so the museum staff would not have to manually restart it.

While the exhibit is fully experienceable for hearing-impaired people, the audio and physical buttons make it only partly accessible for blind participants, the experience still relies on visual prompts, which might make full participation difficult. Visitors who do not understand or are able to read in English will sadly also not be able to fully experience the installation.

Technical Components

Arduino Uno: Handles input from the six arcade buttons and sends the data to the Raspberry Pi.

Raspberry Pi 5: Runs the Processing script, processes the input from Arduino, and plays the video based on the decisions.

Monitor: Displays prompts for the participants and plays the final video showing the outcome as well as giving visual button feedback.

Six arcade buttons: Used by the museum visitors to make yes/no decisions (green for yes, red for no).

Cables: Connect the buttons to the Arduino and ensure proper circuit functionality.

Speakers: Play the audio to accompany the video, enhancing the emotional impact of the installation as well as drawing people in, who hear the sound.

Video and AI Sounds: The video to simplify the project, so Processing doesn’t have to run coded animations as well as the layered sounds to make it more immersive for users.

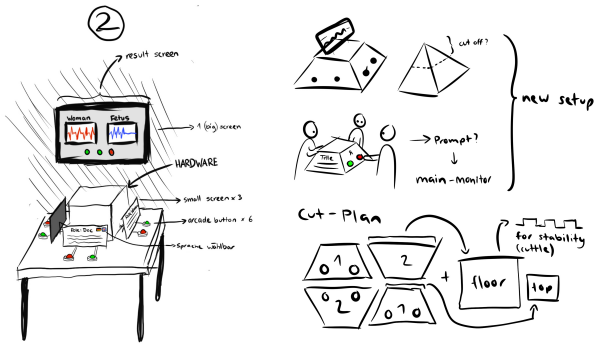
Ethical Questions

The main ethical dilemma the exhibit presents is the question of body autonomy and making a choice about another humans body, as well as comparing two lives, making a decision on which one gets to live. By presenting scenarios in which the museum visitors have to choose between saving the woman who is pregnant with a baby or the unborn child, sacrificing the mother. The installation challenges the influence of external forces—doctors, relatives, social pressures—on personal decisions. The installation highlights how such external forces can dictate or define a woman's independence, often not even making it possible that she gets to decide what happens to her own life, raising essential questions about the extent to which individuals can make decisions about their own bodies in the face of external pressure and societal norms as well as laws in some countries. It makes the

user choose between two human lives, which is ethically extremely challenging, as it also raises the question of whether humans should be allowed to decide about lives of others at all?

Contribution to the project

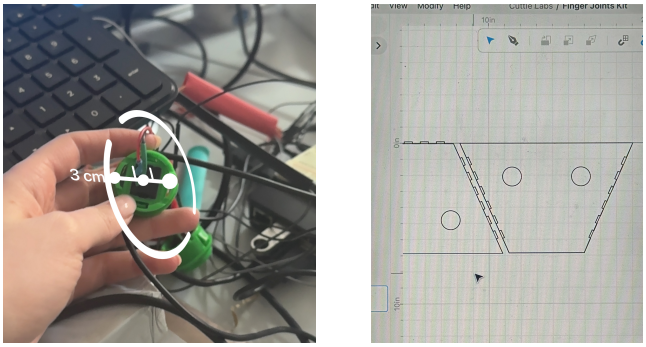
Before starting the process of bringing the installation to life, I started doing the research on what hardware to use and how our concept could be made. After learning more about the Hard- and Software I created a rough plan for how I could set the system up - this step was mostly planning with hardware.



Left: Initial Idea, Right: New Setup-Plan and Cutting Plan

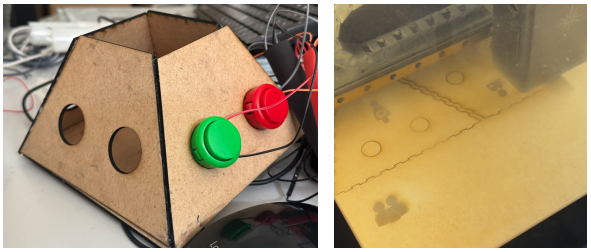
Later on, we had to simplify this plan and settled on one monitor for the prompts, that will display the heart-monitors after the decision making process was done, partly because the Raspberry Pi we were using didn't support more than two screens and we would have potentially needed four, not all of them being able to display mirrored content

For the physical structure of the installation, I had to custom design the laser-cut wooden enclosure from scratch, creating the SVG files in Illustrator and also using the website Cuttle for the edges. The wooden shell could not have been a simple cube, because we wanted a visitor-friendly and inviting physical interface that would be easy to use, therefore I decided to make it slanted.



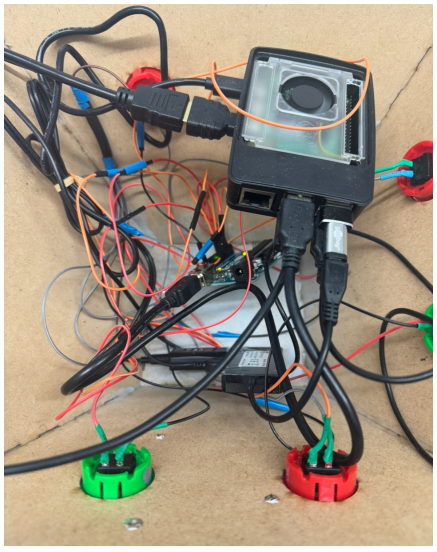
Left: Measuring button sizes, Right: Document in cuttle

The construct has interlocking puzzle-like teeth to hold it together, helping it to securely fit while being slightly angled to make it comfortable to press buttons. To make the different roles more recognizable I looked for different icons for my teammate to engrave into the cutouts. Since no pre-existing designs fit the project's needs, I made my own cutouts, measuring everything precisely to make sure it all fit at the end, during this process I also got familiar with the laser-cutter.



Left: Initial small test-cut, Right: Final Cut with laser-cutter

To hide the hardware components, I drilled two extra holes into the structure, allowing only the necessary cables to stick out while keeping the electronics inside the enclosure. After making the box, I mounted the six arcade-buttons into their designated cutouts. Since the buttons were already wired for initial testing of the code, I had to disconnect all of them and re-solder the common ground connections, and then reconnect the cables to the Arduino, finishing the hardware setup of our project.



Inside view of the re-soldered grounds with blue isolation tape, the Arduino and the Raspberry Pi

For the input system, I initially attempted to use the Raspberry Pi's GPIO pins to read button presses directly. I wrote a Python script to handle the inputs on the Pi and created six threads—one for each button—so that all buttons can be monitored simultaneously. The idea was to prevent any delays in detecting user input. However, running both the Python script and Processing on the same processor caused issues, as they blocked each other from executing properly. This was something I did not

realize before trying it out for two days. Even though the Python script successfully detected button presses, Processing only ran when input is received, which lead to the prompt only being displayed letter by letter, due to the animation I added. Because the threading I tried to implement in python was only simulated by the system and not actually running on a separate processor, I had to find a workaround.

To resolve this, I decided to move the input handling to an Arduino microcontroller. I wrote a small script on my computer, where I had the ArduinoIDE downloaded. I uploaded the code to the Arduino. The Arduino UNO continuously reads button presses and sends the input data to the Raspberry Pi via serial port communication, instead of sockets. The buttons are now wired directly to the Arduino instead of the Raspberry Pi. In the Arduino code, I decided to pre-pair the buttons so that each set of red and green buttons (e.g., R1 and G1 as a pair) only allows one response —either yes(green) or no(red)—per round, this was mainly because Processing was already very slow on the Raspberry Pi and I wanted to avoid adding more elements that Processing would have to handle. Once a button in a pair gets pressed, the other one gets ignored. The Arduino then sends the data to Processing in the format "R1" or "G1" (for the first button pair), "R2" or "G2" (for the second), and so on.

The Processing script handles the main part of the code and the installation overall, since it also provides the visual interface for the visitors, the video, sounds as well as button feedback. It includes an ArrayList for storing user inputs, that it gets from Arduino and another for the different scenario prompts, which then gets shuffled. The prompts are shuffled so it doesn't get too repetitive. My main goal for the GUI was to reflect a neutral, sterile medical aesthetic, using a turquoise color for the text and a font that reinforces the clinical and technical theme, making it look like medical equipment. The draw function is responsible for displaying the GUI, showing the current prompt, and providing feedback to the people interacting with the exhibit. When a button is pressed, a corresponding circle in the color of the button pressed appears on the screen to show visitors that the input has been registered, so their action gets a visual feedback as well as letting them know, if they e.g. pressed too lightly that they need to press again so their choice gets registered. Additionally, a scenario counter is displayed, letting participants know how many scenarios remain (e.g., "1 out of 3 scenarios" or "3 out of 3 scenarios"), I thought this would be important to let people know how many questions would be left so it would be more likely that they finish the prompts without losing interest, as well as providing a sense of orientation.

The void serialEvent function handles the incoming data from the Arduino. It checks for input values such as "G1" or "R1" and counts the responses users have given. The EvaluateOutcome function then determines the final result by comparing the number of green versus red responses, regardless of which specific button was pressed, it only checks for „G" versus „R", which stands for Green and Red. If green responses outnumber red, video1 plays; otherwise, video2 plays.

Since our Raspberry Pi did not support Processing's default video library and had trouble playing the video, I had to use VLCJ for video playback. Additionally, to create a more immersive experience for the visitors, the audio file is played within the Processing script. After three scenarios the evaluation of user responses is complete, theres no new prompt for users, the audio stops, and the corresponding video plays depending on the majority of yes or no answers. After the participants are confronted with the result of their choices for 15 seconds of listening to the heart-monitor flatlining, the video ends and the prompts start again. I added the void reset function to reset all variables, so that the installation returns to its starting state without requiring manual intervention by staff, therefore the next group of participants can just interact with the installation again.

Conclusion

During the project I had to find a lot of workarounds for problems and spent a lot of time on planning, retrying different approaches and also learned a lot better how to code. It was my first time working with hardware and software to this extent so I took a lot away from the experience and I am overall happy with the result, despite the process of getting here was very tiring and nerve-wrecking.

Final Installation Piece in the Uni-Lab