FLOSS BOSS

Kashaf Ahmed

Christine Espeleta

Nemesis Velazquez

Melodie Wang

IGME-587

Travis Stodter

September 10, 2025

Index

Executive Summary

Step into the tiny shoes of a shrimp, tasked with cleaning the enormous teeth of a monstrous lake creature. Towering teeth, covered in grime and plaque, loom above you as you wield only a giant toothbrush and floss pick. Every swipe, scrub, and flossing motion brings a sense of accomplishment as the monster's mouth transforms under your care.

This is more than a game, it's a full-body, high-energy adventure. Feel the thrill of racing against time, the satisfaction of sparkling teeth, and the challenge of battling a relentless, growing mess. It's fast-paced, hands-on fun that engages your instincts, reflexes, and problem-solving skills, turning a simple act of cleaning into a pulse-pounding, immersive experience.

We offer a uniquely physical and digital experience where you, a tiny shrimp, take on a giant lake monster. Exciting, challenging, and entertaining, you're not just playing, you're conquering a beast from the inside out.

Project Scope

Minimum Viable Product

The base concept would require a single screen that displays the teeth of a monster along the upper and lower edges.

The controller would be a large toothbrush used to mimic brushing teeth. When the toothbrush is moved in the air, it would also move on the digital display. When the digital toothbrush is in close proximity to a digital tooth, it would trigger a brushing animation. As a tooth is brushed, plaque slowly disappears until all plaque is removed. Upon fully cleaning the teeth or reaching the time limit, the user would receive a score based on the achieved cleanliness.

A score screen would show the final score for the player.

Deliverables

- Digital design: Teeth design, background, toothbrush design & animation, bacteria design & animation
- Physical components: Giant toothbrush, Giant teeth
- Electrical hardware: Toothbrush chip, Teeth sensors
- Backend code: Chip sensor data usage, etc

- Does not have alternative controllers, additional immersive visuals, chip readers, physical interactions with additional props, or a progress bar that can make backwards progress.

Optimum Viable Product

An optimal project would utilize projection mapping on the teeth and the user's immediate environment. There will be a projection across three surfaces in an arched configuration, creating an immersive environment. There would also be a projection beneath the user's feet on the ground, representing the bottom of the creature's mouth. The projection would include the monster's teeth along the upper and lower edges and a view inside the mouth. Physical teeth will be 3D-printed and placed along the bottom and top edges of the screen in line with the projection. Three large object controllers would be built to control the digital interactions.

The three objects would include a large toothbrush, a floss pick, and a mouthwash shooter. Both the brush and the pick would have a chip near the tip so that any movement of the controller in real life could be tracked and translated to the virtual representation on the screen. Each tooth would include a chip reader that would sense when the controllers are in close proximity to each specific tooth, as well as LEDs. Proximity would allow the user to brush or floss the teeth. The mouthwash shooter would be a point-and-shoot laser connected with a game console's sensor bar. It would be used to shoot and eliminate bacteria in the mouth.

When the user is not actively cleaning, bacteria would regenerate or multiply, and plaque on the teeth would slowly spread to neighboring teeth. A user must balance the different methods of cleaning to achieve a clean mouth within the time limit.

Upon reaching the time limit, a user would receive a score based on the cleanliness of the mouth. If a player were to fully clean the teeth before time runs out, they would receive bonus points based on the remaining time. Total time, score, and cumulative data across all players, and a leaderboard would be displayed on the result screen.

Deliverables

- Digital design: Teeth design, background, toothbrush design & animation, floss pick design & animation, mouthwash shooting animation, bacteria & plaque design & animation
- Physical components: Giant toothbrush, Giant floss pick, Mouthwash shooter (laser), giant teeth
- Electrical hardware: Toothbrush chip, Floss pick chip, Mouthwash shooter laser and sensor bar, teeth sensors
- Backend code: Leaderboard database, Chip sensor data usage, Controller tracking, etc

Risks & Mitigation Plans

The main technical risk will be unfamiliar technologies, as we've never used RFID systems, so we will have to perform extra research on them. Chips have been briefly discussed in the past, and we will need to conduct further research on them. More research would also need to be done on Unity because we only have experience from IGME 201. There could also be complications in having the connection from the physical components to the projected or displayed content. If users are physically brushing a tooth, we have to consider how we would connect the action to the game to show the tooth being cleaned. Similarly, we would have to consider the connection of the mouthwash shooter to the game, such as picking it up and having that show up in the game based on where we are pointing and having it shoot.

Another looming issue would be our time constraints. It would take a long time to make the props since we are making large-scale props, including the large teeth. Testing physical components and getting those working properly would also be time-consuming. If- for some reason- members of the group cannot all meet up, we can give them individual, smaller tasks to work on while the rest of the group can work together on larger elements. If there are new additional requirements, we would split them evenly among all of us. If something were to go wrong with the project, we could adjust the scope by reducing the number of tools or interactable teeth. We could also discuss with the professors about what the next best step would be.

Resources

- 3D printed set of 12 teeth, plus tests - via FabLab
- 7/8 Arduino boards with long jumper wires
- Variety of resistors
- Large flatscreen TV or projector(s)
- 12-16 RFID [NFC] chips and a reader
- If physical teeth, 8 actuators
- LEDs
- Battery pack / 9V Batteries, extensions
- Lightweight prop materials
- Blackout curtains
- Extension cords
- Cardboard/plaster
- Ladders
- Frame (PVC pipes)

Project Architecture

Overview

Our experience starts with the user and how they will interact with our physical stage and props, manipulating them to transmit movement data to receivers in our toothy stage. This data will be managed by an Arduino- or rather, a series of Arduinos- that will communicate with a Unity environment. This virtual stage will either be screened or projected, while a few key data points from the users will be stored online.

Components

The Stage

The main, physical part of our experience will have an RFID system that allows our user to interact with the teeth and- by extension- the display behind them. This RFID system will be managed by an Arduino, utilizing C programming languages.

The Props

The smaller, directly interactable parts- the brush, the gun, the floss- will contain the other half of the RFID system, which allows our experience to register and process the user's interactions. This should be the thing sending out the signals to be processed. The sensors will be tied into our experience with foam props, though durability may require us to switch to a different material.

Digital Integration

The final component will be the display, either connected to a monitor or projected to a surface, which will be what gives the user their feedback and some additional clarity regarding their tasks for maximum accessibility. The integrations will ideally be managed in Unity, while the scores the user accumulates will be randomly attributed to various usernames, and the top scores will be stored in a database hosting site such as Firebase or MongoDB, as those are the systems the developers have the most experience with.

Technologies

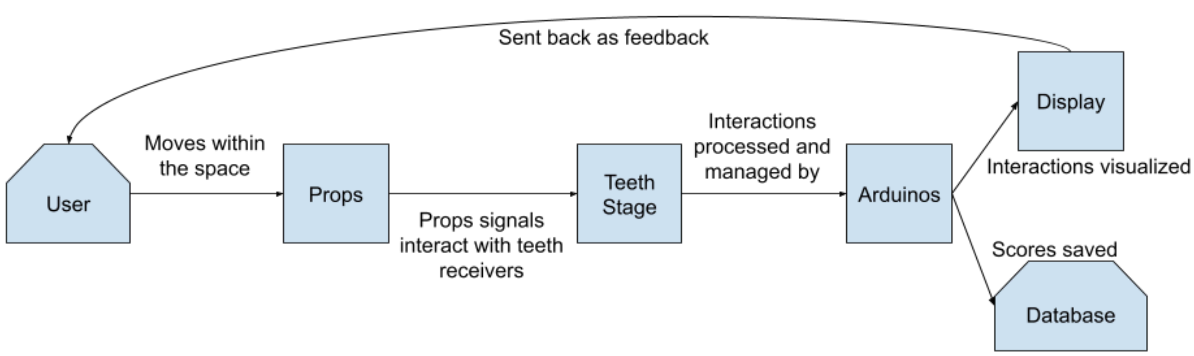
Languages

- C - RFID system, Arduino
- C# - Unity Integration

Services/Platforms

- MongoDB/Firebase - User Score storage
- Unity - Manage controller input, translate to screen

Data Flow



Scalability

Our project is easily scalable due to the use of RFID technology, which is relatively cheap, small enough to integrate into a variety of parts, and easy to use. Since most of our tools rely on this, it would be simple to port over functions for similar tasks. Scalability is crucial to us as things can go wrong at any moment and still having a base game with a variable number of tools will help us let the users walk away with an enjoyable experience, no matter the conditions.

Security

Our project needs minimal security as all parts of our experience require proximity to the setup and therefore only needs to exist locally. While our scoreboard will be assigned with random names to maximize the number of users in the experience, the saved data can be safely hosted on something such as Firebase or MongoDB.

Team Responsibilities

Kashaf “Kash” Ahmed: Will work on all parts of the project, specialist: Arduino, frontend.

Christine “Chrissy” Espeleta: Will work on all parts of the project, specialist: frontend, backend, database.

Nemesis “Neme” Velazquez: Will work on all parts of the project, specialist: calibration, troubleshooting.

Melodie Wang: Will work on all parts of the project, specialist: digital integration.

Bus Factor Mitigation: The bus factor on the Development side is four. None of the devs will be responsible for one specific aspect of the project. Each dev will contribute and be familiar with each aspect of the project. We have a project Discord to share information and concerns, so we will be able to resolve conflicts and communicate ideas quickly. We also shared each other's schedules so we know when would be best to meet and work on this project. Overall, we will all be responsible for everything, based on our familiarity with the software we use.

Audience

This project is designed to appeal to a broad, family-friendly audience, ensuring that players of all ages can enjoy the experience. At events like ImagineRIT, it will engage students, alumni, and families with its humor, immersive gameplay, and interactive physical setup. Similarly, alt.ctrl.GDC attendees will find the shrimp-versus-lake-monster concept both surprising and entertaining (hopefully). The game is built to be instantly accessible, with user friendly controls and a visually striking design that draws people in, encourages hands-on interaction, and keeps them engaged, whether they are casual players or more competitive participants.

Stakeholders

Clients

- ImagineRIT audience
    - RIT Students
    - RIT Alumni
    - RIT Staff
    - Families and acquaintances of any of the above

- alt.ctrl.GDC audience
    - Game Developers
    - Game Enthusiasts
    - As a general rule, families

Team

Developers

- Kashaf Ahmed
- Christine Espeleta
- Nemesis Velazquez
- Melodie Wang

Designers

- Zane London
- Dayne Stein
- Noa Spanier
- Irene Tu
- Danielle Antonacci
- Maya Probeck

Professors

NMID

- Travis Stodter

NMD

- Jason Arena
- Mike Minerva

Quality Assurance Plan

We will consistently test and debug our code as we progress through each iteration of our experience. We will maintain constant communication between us- the developers and the designers- to ensure that we are on the same page throughout the project's development. As developers, we will consult with the designers for the design of our front-end, UI components, and interaction flow. We'll frequently communicate with our professors and advisors to update them on our project's progression and receive valuable feedback. Later in the project, we will assess the project using extensive playtesting before publicly presenting it to larger audiences at ImagineRIT and alt.ctrl.GDC.

Advisors

- IGM faculty
    - Travis Stodter - Arduino and physical hardware integration
    - Austin Willoughby - web development and Node.js integration
    - Jake Adams - creative coding, XR, projection mapping
    - Eric Baker - possible Unity integration
    - Michelle Harris - creative coding and possible p5.js integration
- NMD faculty
    - These advisors can provide valuable input on the design direction and process.
    - Jason Arena
    - Mike Minerva
    - Joel Rosen
    - Hye Jin Nae