LunAR Lion

COLUMBIA
SPACE INITIATIVE

Columbia University (116th St & Broadway, New York, NY 10027)

Team Leads:

Kolak, Sophia (sdk2147@columbia.edu, 845-729-4355)
Junior, Computer Science & Math

Radway, Sarah (sar2236@columbia.edu, 513-557-1538)
Junior, Computer Science & Math

Team Members:

Pickard, Aaron (ajp2235@columbia.edu)
Senior, Computer Science (Developer)

Celupica Liu, Iris (icl2113@barnard.edu)
Sophomore, Physics (Developer)

Ahani, Anastasia (aa4277@columbia.edu)
Sophomore, Computer Science (Developer)

Schare, Gregory (gs3072@columbia.edu)
Freshman, Computer Science & Philosophy (Developer)

Cha, Jenny(jc5329@columbia.edu)
Freshman, Computer Science (Developer)

Adib, Sad (sa3783@columbia.edu)
Freshman, Computer Science (Developer)

Zulu, Elijah Temwani(ez2294@columbia.edu)
Freshman, Computer Science (Developer)

Kimberlin, Tommy (tjk2142@columbia.edu)
Freshman, Computer Science (Developer)

Wenzel, Kimi (kvw2109@columbia.edu)
Junior, Computer Science (Developer)

Treasure, Tanner (tt2717@columbia.edu)
Sophomore, Computer Science (Developer)

Slater, James (jls2357@columbia.edu)
Sophomore, Computer Science (Developer)

Lee, Joseph (jtl2164@columbia.edu)
Junior, Applied Physics & Music (Developer)

Joshua Manzorolhagh (yrm2106@columbia.edu)
Junior, Computer Science (Developer)

Ramos, Cesar (cer2178@columbia.edu)
Junior, Computer Science (Outreach)

Fei, Maggie (mf3093@columbia.edu)
Senior, Computer Science (Developer)

Luo, Anthony (al4007@columbia.edu)
Freshman, Computer Science (Developer)

Faculty Advisor:

Feiner, Steven (feiner@cs.columbia.edu, 212-939-7083)

Advisor Signature Date

Table of Contents

I. Technical Section

A. Abstract

B. Design Description

The goal of the LunAR Lion design is to provide astronauts with an intuitive AR user interface to safely carry out tasks in field environments. LunAR Lion seeks to align itself with practices that will make astronauts feel most comfortable; namely by attempting to assimilate to the training and typical procedures astronauts use for other EVA related tasks. These include voice control capabilities intended to mirror the form of communication astronauts use with Mission Control, a SAFER-like backup emergency system, and a similar warning system display to those of Shuttle and beyond, featuring red lighting associated with a given issue when a parameter has been exceeded.

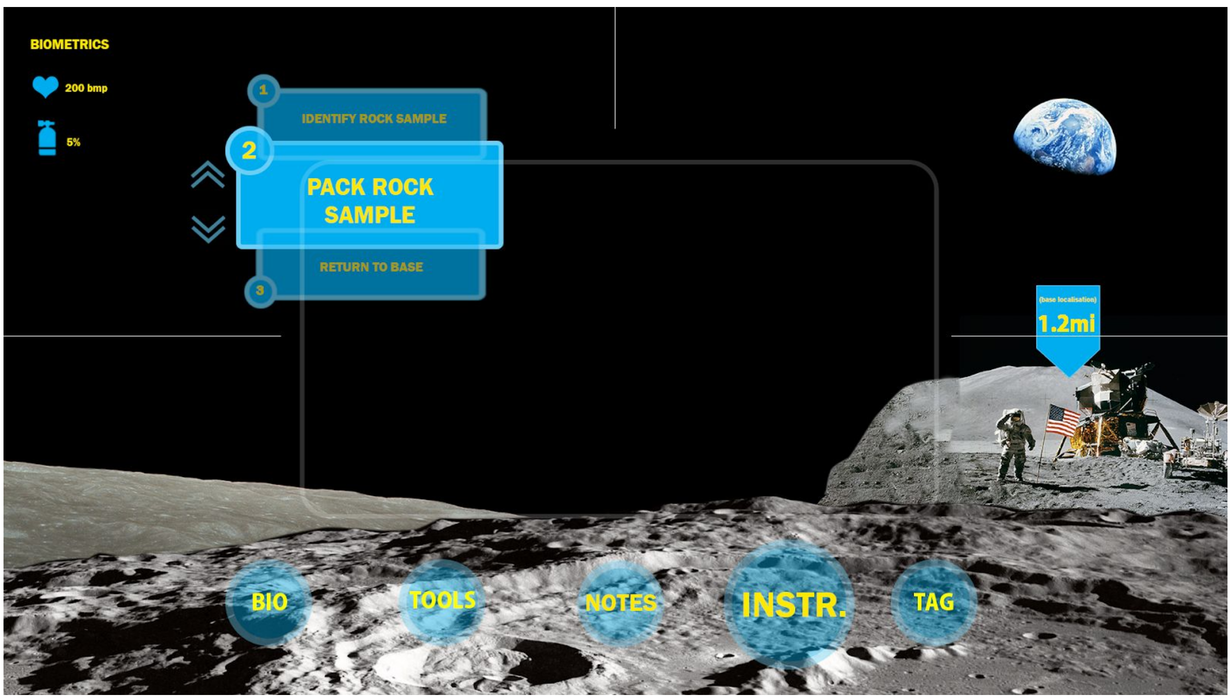


IMAGE 1: The main features of the baseline User Interface; featuring biometrics in the top left, a task bar along the bottom, and an example of the instruction display in the mid left.

The baseline user interface consists of a panel containing suit vitals embedded in a fixed location for easy and immediate access, as depicted in the upper left corner of the image above. Suit vitals and other data obtained from the telemetry feed will be continuously displayed in this fixed location. In the case that this data cuts out or is presumed to become inaccurate, the latest data will continue to be displayed, but the relevant display will turn red and the astronaut will be issued a warning that the feed has been lost (see below for more details on warning messages).

Additionally, there will be a collapsible Mac-style taskbar accessible along the bottom of the field of view. The astronaut may call for the task bar, and select the icon that they wish to use. In Image 1, the user has selected "Instructions" and is being led through the instructions for a geo-sampling mission. After the selection has been made, the tool bar will minimize to the bottom of the screen, to prevent interfering with the astronauts line of sight. However, it can always be called back to re-select or deselect.

For the sake of redundancy, the astronaut will be able to interact with these UI elements using a variety of methods. The primary method of communication with the system will be voice-activation using existing packages for voice recognition and interaction. Astronauts will use predefined commands to select and interact with various UI elements such as note taking and repair instructions. Examples may include "continue to next instruction" or "expand suit vitals." Emphasis will be placed on clear, concise, unambiguous commands with loose syntactic requirements (e.g. "next instruction" and "next task" are also valid voice input commands). The computer will always be listening for commands, but in order to execute a command, it must be activated using a specific reserved keyword (e.g. "computer," "Cortana," "Alexa," "Roaree"), like in existing human-computer voice command systems. After a keyword has been recognized, the following command will be executed. If the astronaut is done with a command and becomes silent, the computer will continue listening for a particular amount of time (e.g. five seconds) until requiring the activation keyword again. Keywords such as "cancel" or "stop listening" will round out the syntax of voice commands which will help avoid the computer becoming confused while the astronaut communicates with ground control or fellow astronauts. It is comparable to having a CAPCOM on the ground and a CAPCOM with you on the surface.

The other main method of activation is eye-gaze. If the astronaut focuses their gaze on a particular element (such as a taskbar icon) for several seconds, the element will be activated or expanded. This can also be used in conjunction with voice commands. For example, if the astronaut wishes to move a UI element around, they could focus on it, say, "move this," shift their gaze to the desired location and say, "over here" to move the icon. There are numerous possibilities for this method of combined interaction.

Lunar navigation will be aided by environmental distance markers, as shown above with the "1.2 mi" marker to the lunar lander. Though not a thorough substitute for traditional navigation, this convenient on-screen element will aid astronauts as they explore the lunar surface and complete mission tasks, as well as serve as a helpful aid in case of disorientation.

Using depth illusion, most UI elements will appear at a distance to avoid requiring astronauts to refocus their eyes between close-up text/icons and far-away physical objects.

For example, distance markers should appear near the location they mark (and thus change based on distance), and likewise, instructions appear within the environment. Alternatively, the suit vitals will maintain a consistent apparent depth such that astronauts always know where to focus their vision when checking information.

Graphics will also maintain contrast with the environment to prevent eye strain, using a combination of active contrast/contrast enhance and careful UI design with contrasting colors. The design philosophy behind this is similar to the closed captions used for videos and we will be implementing yellow text within a semi-opaque blue box. Yellow and blue are both uncommon colors on the lunar surface, but they are not so garish and unnatural that they distract. All UI elements, including suit vitals, taskbar, contextual markers, and instructions will follow this theme to maintain visual continuity and readability. Only emergency notices and warnings will break this theme, as they must be obvious and conspicuous.

Primary interaction and control of the UI will be executed through predefined voice commands as described above. In the event that the suit loses pressure or the astronaut becomes unable to speak, a mode of failure will be implemented similar to the pre-existing SAFER system that allows the suit's UI to be navigated using a SAFER-like joystick. SAFER is already an emergency activated system, so this would simply be expanding its scope and usage. Maintaining this operational consistency ensures that emergency protocols are intuitive and familiar to astronauts. This system would allow for some form of communication with the ground in emergency situations, and would allow for the astronauts to still have the information they need in order to return to safety.

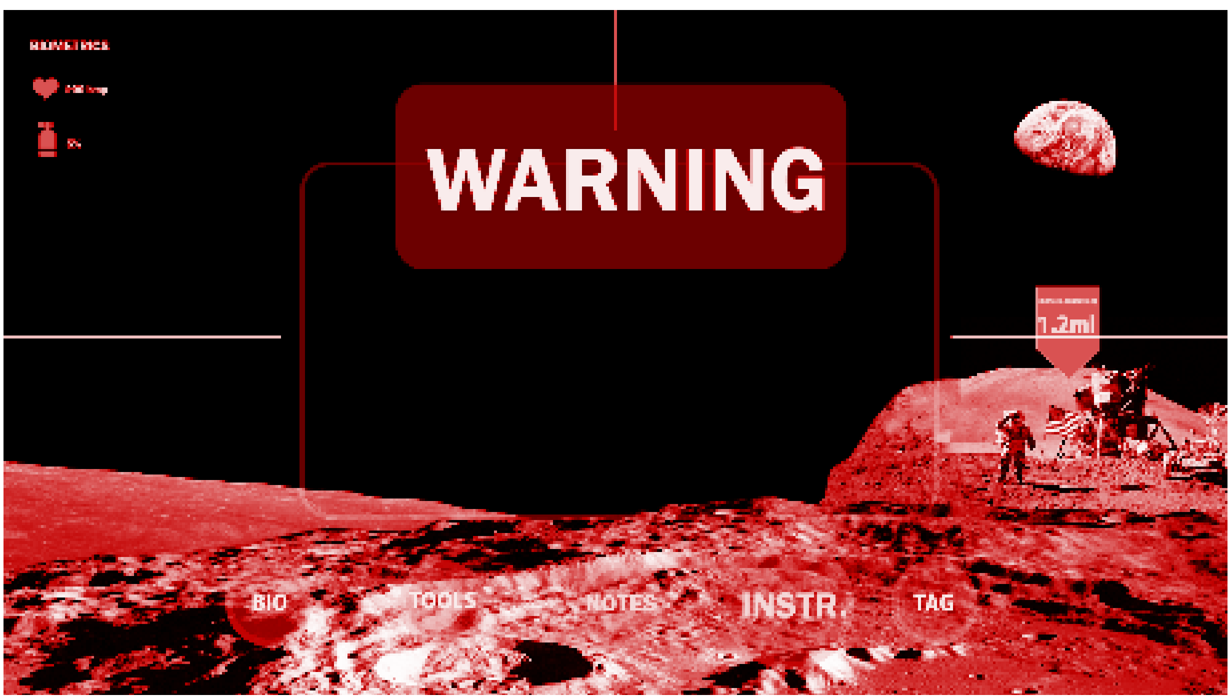


IMAGE 2: Preview of warning display

Additionally, in the unlikely event the astronaut needs to return to a pressurized environment or the astronaut encounters other potentially life-threatening circumstances, the UI will issue a warning to the astronaut based on the danger level of the threat. All warnings, irrespective of danger level, will take the same general form: they will be textually displayed as white text within red boxes at the upper-middle portion of the vision field, and will be accompanied by an audio alert. Warning text should be large so that it calls attention, but not so large that it takes an inordinate amount of time or eye movement to read and process.

The precise nature of these warnings will depend on their severity. For low-level warnings, the box will not flash and will be accompanied by a short "beep" alert sound to grab the attention of the astronaut. For more severe warnings, this "beep" alert will repeat on regular intervals until the issue is resolved. In the case of life-threatening conditions or emergencies, the warning message will be accompanied by a continuous alarm sound and an emergency voice alarm communication (EVAC) that will describe the problem. Additionally, a short textual description of the problem (e.g. "suit pressure low") will display near the red warning sign and pulse to ensure that the astronaut visually notices the message. All audio alerts exist solely as a means to alert and inform astronauts of immediate danger. Though severe alerts must demand attention, we avoid sounds and visual effects that are overly intense as they might detrimentally overwhelm the astronaut in an emergency situation. Also, the astronaut will be able to stop the alarm or voice message at any point in order to ensure they are calm and focused.

C. <u>CONOPS (Concept of Operations)</u>

The CONOPS outlines LunAR Lion's primary goals: accommodating the process of navigating through a UI with a Hololens and EVA gloves, analyzing and displaying relevant task information, and recording log information through photos and notetaking throughout this process.

NAVIGATING THE UI:

In a more standard AR interface, the user would navigate through displayed information using hand gestures, since the Hololens has built in functionality for interpreting hand movements such as sliding and clicking. In our case, however, the astronaut's stiff EVA gloves will limit the range of motion for their fingers, making gestures an unreliable means of interacting with the UI in this safety critical environment. Additionally, we want our UI navigation to be hands-free because the astronaut's hands will oftentimes be busy, for instance, during the tire repair process, or while they are taking samples. Due to the difficulty of reading gestures safely given EVA gloves, and the necessity of allowing interaction without distracting from the task at hand, we completely avoided a gesture based system of UI navigation. Therefore, to navigate through LunAR Lion, astronauts will primarily use voice control to give commands, except in the case where a failure scenario occurs (described below).

Because we will be developing the project in Unity, the voice recognition system will utilize the KeywordRecognizer class, which is automatically implemented in UnityEngine.CoreModule and is imported using UnityEngine.Windows.Speech. KeywordRecognizer will match vocalized phrases to registered keywords. Speaking keywords such as "Menu" will map to a recognized and defined event; in our case, pulling up the taskbar options. For our voice input source, we plan to use the onboard microphone built into the Hololens headset for its improved integration.

For certain parts of the UI, the voice recognition system will be complemented by a head-gaze and dwell system of navigation (1), a method of operating the AR environment which creates a world-locked cursor that follows the position and orientation of the user's head to determine the 3-D generated or real-world object at which the user is looking. This is accomplished by the Hololens sensors creating a vector field and looking for intersections/collisions with interactive Unity objects. Once the object is targeted, a certain amount of time during which the user gaze remains on the object (3-5 seconds) will trigger the action linked to that interactive Unity object. One such example would be a constant taskbar in the user's lower field of view that when targeted and gazed upon will open the icon that is being emphasized. For instance, if the astronaut fixates on the biometrics icon a display will appear giving them detailed readings on their health levels.

As a final failsafe in the unlikely case where voice recognition fails to properly detect the astronaut's intended action, and when head-gaze and dwell does not work as intended, we will have a mechanical device that allows the astronaut to manually move through the UI. This device will be similar to the joystick associated with SAFER, another tool used in emergency situations. There are two modes of failure in which this device could be used; the first being if the system is broken and recognizes it is unable to pick up sound, in which case we will display a failure message, instructing the astronaut to use the joystick. In the second case, where the system mis-interprets speech, we leave it up to the astronaut's discretion to use the joystick instead. In other words, the joystick will be available for use at any given time if the astronaut feels the voice commands are not working properly to ensure the highest degree of safety.

ANALYZE, DISPLAY & RECORD RELEVANT INFO:

Our method for displaying task instructions breaks all instructions into short, logical steps, and displays them sequentially. At the start of the task, we will give a brief overview of the entire procedure, with an estimated time for completion and a list of necessary tools. Then, we begin by displaying the first instruction both in writing, and using a 3D overlay on the object itself indicative of the task. Once an instruction has been shown, we wait for a verbal response from the astronaut confirming that they have completed the step, and would like to continue on to the next instruction. After receiving this confirmation, the completed instruction moves to the bottom right corner of the screen so that it is easily viewable if they want to look back, but not distracting during the subsequent step. As the astronaut progresses through the steps, we will create a visual back stack that allows them to review any completed step of the task, in-case something went wrong at any point. If the user say "go back", they can move back a step and bring the most recently completed task back into their frame. In this way, following the instructions will mimic the experience of reading a book, where you can flip back and forth between "pages" and their corresponding overlays using verbal commands. We also create an option for displaying all written instructions at once if the astronaut is not sure where a mistake occurred and wants to go through everything at once. To avoid the problem where focusing on something in the distance causes objects up-close to blur (and vice-versa) we will be displaying the instructions projected into 3D space with the task itself, moving only to maintain its position in the user's line of sight.

In regards to analyzing relevant information, the Rover Repair Task requires LunAR Lion to identify the tire to be removed, identify the bolt that controls the elevation of the tire, and use and identify tools to change the tire. Such goals require the use of an image recognition service, moreover, one that is able to recognize unique objects such as the rover tires. We will be implementing Microsoft Azure's Computer Vision, one of their Cognitive Services, to this effect. Once the rover task is triggered, we will use Azure Custom Vision, a customizable image recognition system, to build our own image classifier. using images of rover tires and rover parts. This consists of submitting groups of images of variations of the tire flat, as well as images that do not contain those characteristics, and applying those specific image labels, training the model to recognize when a tire needs to be repaired.

Thus, when an astronaut says "Check Tire" with the tire in field of view, the image will be captured locally using ScreenCapture.CaptureScreenshot, uploaded, and the prediction REST API called on the image. The result (ie. whether the tire needs repair) will then be displayed. We are also able to train our Custom Vision model in a similar manner in order to identify the specific bolts used to elevate the tires. Submitting different images of the tire, from face-on, an angle, etc, and labeling a specific image component as the tire elevation bolt, yields a model that is able to identify that specific bolt given the image uploaded when the astronaut calls for a "Check Tire". The identifier is then displayed in the event the tire needing to be repaired returns true.

The instructions for this task will display an image of the tool required for each specific step alongside the standard instruction-display method described above. We will be maintaining a database of the astronaut's "toolbox" containing images of each tool type, and when a certain tool is in the instructions, we will search through our database and then display the image of it.

Additionally, we will be using Vuforia to project and animate the steps of the instructions. This visualization will act as an additional visual reference for the astronauts as they work on a task. To implement this feature, we imported and stored a CAD model from (CITE) of the rover, which gives us an accurate 3D representation of its components. After importing the object files that comprise the rover into Unity, we can manipulate and accesses them as 3D objects. Using Vuforia, we animate and manipulate these images to create a visual overlay of instructions during a task that is presented above the physical object. During a task process, LunAR has a predefined set of visualizations that correspond to each written step of the instructions. For instance, if step x of the tire changing task is to remove bolts, then lunAR will display task x described with words, while the visualization will show the tire with bolts extracted (example below). The combination of written and visual display is given in order to make work as efficient and intuitive as possible.

SCIENCE SAMPLING & NOTE TAKING:

In order to accomplish the science sampling task, LunAR must first be able to locate and navigate to the correct sampling site, which we assume will be marked. Given some visual cue of the region to be sampled, we will be able to record the position of the sampling site in the x,y,z planes by defining a Vector3 variable. As the astronaut moves further and further away from the sampling site, or the origin, we will continually update our current position with another Vector3 variable that we will be calling transform.position. Using vector calculus, we will be able to easily determine the distance between the current position and origin. Additionally, we will also be able to display an arrow that points in the direction of the sampling site; this will be accomplished by constantly calling transform.forward on the current position vector, which returns the angle between the position vector and the origin, and flipping that angle in order to retrieve the direction towards the sampling site.

The next job for LunAR is to measure and record detailed information about objects. Using the Spatial Mapping built into Hololens, LunAR will capture 3D scans of sites and samples, giving us detailed information about the surveyed objects. To capture a 3D scan of a site, the astronaut will give a voice command to start and stop a scan of the site. The scan duration will range from 30 seconds to a few minutes with longer durations resulting in higher fidelity 3D scans. The resolution of the scan will be around 1200 triangles per square meter (there seems to be no benefit beyond this) and we will use the SurfaceObserver API within Unity to access the 3D mesh created and export to an object. This can be added to the notebook entries and integrated with text files including verbal descriptions. It could also be combined with pictures to create a 3D image map. To create 3D scans of samples or objects, we can leverage the existing Hololens application SpaceCatcher.

We can use Holo Measure (3), an existing Hololens application, to allow the astronaut to obtain measurements and distances of samples and surfaces respectively. From there, depending on the size of the material, the screen will display the necessary tool and instructions for data collections. As mentioned earlier, to capture a higher-fidelity 3D scan of a sample, we will leverage SpaceCatcher, an existing Hololens application(4).

For the note-taking portion of the Science Sampling Task, we will be using a voice recorder that will store voice notes as text files in the onboard Hololens storage and then retrieve them using the Unity file manager. We are able to once again implement our voice recognition system, using the KeywordRecognizer class in conjunction with Unity's Dictation Recognizer, in order to translate voice to text in real time to create and save a readable file for future use. The beginning of a note may be triggered with the key phrase "Create Note", and any words uttered afterward will be added to a String array. That array will then be rendered as a complete string, and a system path and file created in the Hololens storage, named by the astronaut using another voice note after a prompt. In order to retrieve that file and display the text in the UI, the astronaut will need to call "Retrieve file_name" and the file will then be displayed using the TextAsset.text method of text files.

Additionally, for the Science Sampling Task, a spoken command will call the method Application.CaptureScreenshot() from Unity, capturing the environment in an image for future use.

Citations:

1. (https://docs.microsoft.com/en-us/windows/mixed-reality/gaze-and-dwell)
2. (https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/home
3. (https://www.microsoft.com/en-us/p/holo-measure/9nblggh52bq1?activetab=pivot:overviewtab
4. (http://spacecatcher.madeinholo.com/

III. Outreach Section

The goal of outreach for our team is to instill a curiosity and passion for space and space technologies in both our own undergraduate community and the NYC community at large.

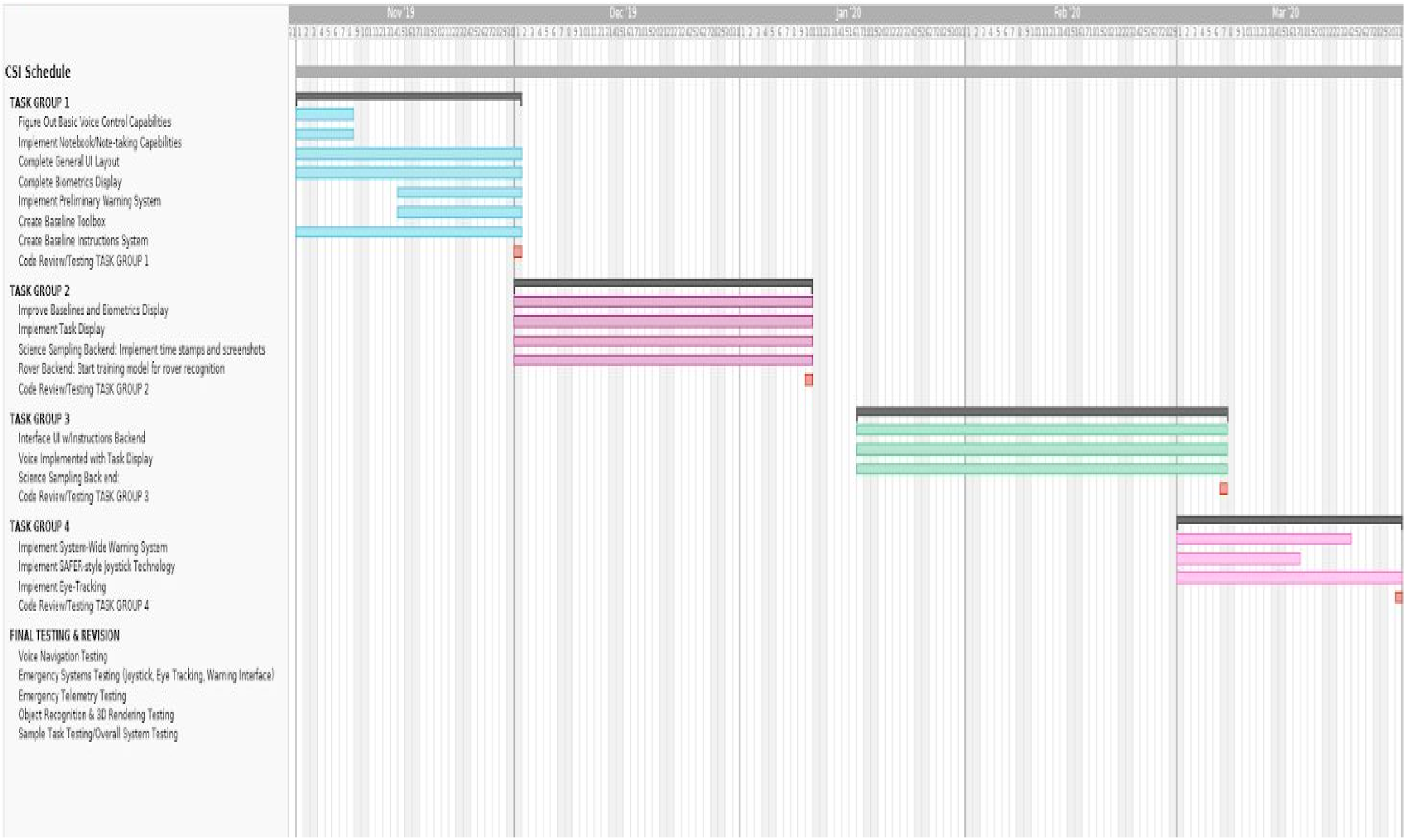
Among our peers, we have already made headway on outreach by registering for Columbia SPLASH, an annual program wherein current Columbia undergraduates teach classes to high school and middle school students which provide them with the opportunity to learn about material outside the scope of their current school curriculum. The purpose of participating in this program is to foster student interest in the subjects of space flight, exploration and engineering and to promote awareness of the opportunities available in higher education and careers. The volunteers from Columbia Space Initiative will receive training on the logistics of the event so that they are prepared to teach classes and volunteer. Columbia Space Initiative's classes will cover topics including the fundamentals of space science, more advanced topics like propellants and orbital mechanics, and the history of the aerospace industry. We hope to inspire the kids in these classes to pursue any interest in space science, engineering, and more.

We are also planning to partner with the Manhattanville Community Center, which has an after school program that promotes volunteering, education, and action in the New York Community and beyond. We would work with them on a variety of programs, from classes about space science to events which help K-12 students see what opportunities exist in the field of space exploration and engineering. Our goal with this project would be to encourage future generations to pursue careers in this field.

Moreover, our team believes that diversity and inclusivity in STEM are of utmost importance, but tend to be overlooked by our community. Thus, a key point for our outreach program is to empower underrepresented students and transmit not only why we are passionate about space and space technologies, but also why they are important and how they can get involved, attempting to eliminate the barrier that is not visible, yet still present. To achieve this, we plan to partner with local organizations, such as ELiTE Education, Hypo The Kids, and Black Girls Code, whose missions are already to empower underrepresented communities. These three organizations either have a base in New York City (specifically Harlem), or are already partnered with Columbia University and its student organizations, making our mission possible without the need of oversaturating the existing infrastructure. We are already in contact with each of these organizations, discussing availability, scheduling, and curriculum, considering that we would like to taylor our outreach to each of these organization's focus.

Finally, we will be partnering with a local elementary school to plan a field trip to the Intrepid Museum or the American Museum of Natural History, with a focus on the STEM exhibits, while talking to the kids about our designs for the new AR Space Suit. We would hopefully encourage these kids to see STEM as fun and engaging while encouraging them to dream big.

SCHEDULE:


Nov '19
Dec '19
Jan '20
Feb '20
Mar '20
CSI Schedule
TASK GROUP 1
Figure Out Basic Voice Control Capabilities
Implement Notebook/Note-taking Capabilities
Complete General UI Layout
Complete Biometrics Display
Implement Preliminary Warning System
Create Baseline Toolbox
Create Baseline Instructions System
Code Review/Testing TASK GROUP 1
TASK GROUP 2
Improve Baselines and Biometrics Display
Implement Task Display
Science Sampling Backend: Implement time stamps and screenshots
Rover Backend: Start training model for rover recognition
Code Review/Testing TASK GROUP 2
TASK GROUP 3
Interface UI w/Instructions Backend
Voice Implemented with Task Display
Science Sampling Back end
Code Review/Testing TASK GROUP 3
TASK GROUP 4
Implement System-Wide Warning System
Implement SAFER-style Joystick Technology
Implement Eye-Tracking
Code Review/Testing TASK GROUP 4
FINAL TESTING & REVISION
Voice Navigation Testing
Emergency Systems Testing (Joystick, Eye Tracking, Warning Interface)
Emergency Telemetry Testing
Object Recognition & 3D Rendering Testing
Sample Task Testing/Overall System Testing