Georgia Institute of Technology

Evaluation of the HelmetGo Wearable Sonification Application

Project Deliverable #5: Detailed Report and Appendices

Team HelmetGo

Xueying Huang, Siyu Li, Ruixuan Sun, Luming Yin, Mingyuan Zhou

CS 4590 Computer Audio

Maribeth Gandy, Scott Robertson

December 4, 2018

Introduction

With public transportation and shared rides becoming more and more popular in urban areas, riders are in dire need of safe and convenient devices to assist with environmental detection. According to the *US Navigation Device and Usage statistical report* from 2015, mainstream commercial environmental detection and sonification systems are mostly only available for smartphones. However, smartphones can be a major distraction for riders of bicycles or electric scooters. For example, the *Traffic Safety Facts report* from 2016 points out that smartphones are listed as the major distraction factors for drivers and motorcycle riders, subjecting drivers to fatal crashes. Thus, a wearable computing device can be quite useful when creating a safer and more intelligent public transportation system.

The goal of this project is to design an intelligent safety helmet that provides environmental detection of various dangers and obstacles on the road, then sonifies this information to its wearers. In this project, we created a Unity-based prototype of this safety helmet and conducted specific user studies. The design of the device targets novice college student commuters who ride electric scooters on a daily basis. Environment detection allows users to be better aware of the environment. Variations in the sonification mechanism help alert the user when nearby obstacles are detected. For example, when a car approaches the wearer from the left-hand side, the audio system would dictate “left” followed by a stream of sonified beeping noises, where its frequency would shift from low to high as the car gets closer.

Target User Survey

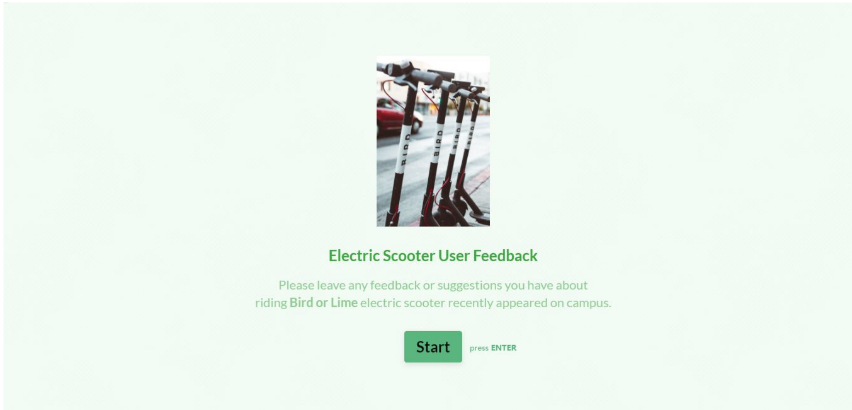

Figure 1. Target User Survey Welcome Page (Typeform link: https://soniahuang1.typeform.com/to/HnEcWP)

To kick off the project, we created a user survey, asking for user feedback of their experiences when riding an electric scooter. We asked them about their perception toward electric scooters. For instance, we specifically asked them how often do they fall off due when trying to observe obstacles, and whether they wear a helmet when riding a scooter. From the target user survey, we confirmed the idea that a helmet with audio alerts can be helpful for college students regardless of whether they are avid or amateur scooter riders.

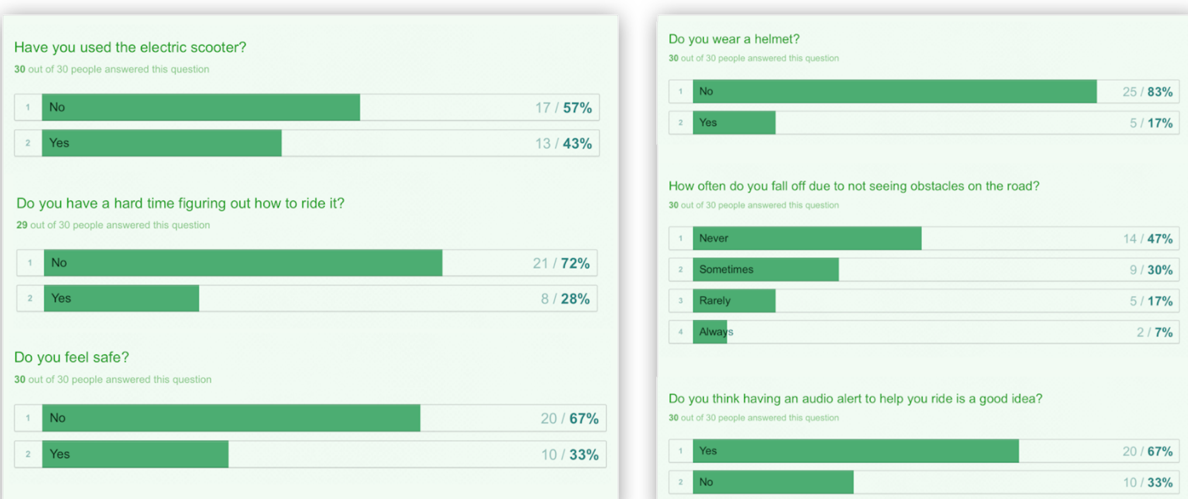

Figure 2. Target User Survey Statistics

As we can see in the target user survey, more than half of the surveyed college students don't feel safe when they are riding an electric scooter. Unaware of its safety benefits, over 80% of them never felt the need to wear a helmet. Over half of participants have fallen off from the scooter due to encountering some obstacles on the road. Finally, most participants agree that a sonification alert system can definitely help create a safer ride.

Design Protocol and Evolution

The first step of our design is to search for a target user group. After analyzing various online resources and recalling our own personal experiences, we decided our potential user group to be frequent urban commuters who have a certain amount of need for using a short-distanced non-motorized vehicle. As the *US Bike Share* statistics illustrates in its yearly report last year, college students can be an obvious target. An example user persona of our solution is shown below:

"Sophia is a third-year college student who has a daily need for commuting between her apartment and the school campus. During weekends, she would also go shopping or biking in the city park. She has bought a student monthly pass for a local bike share plan. To safeguard her trips, she also subscribed to two electric-scooter environmental detection apps on her phone. But the frequent concern she has met during weekdays is that there are always a lot of cars or students sharing the same lane, and it's easy to bump into others if she's not paying enough attention to the road. She knows that non-motorized vehicle is the eco-friendly and sustainable way to go, but the safety problem is an urgent concern for her and her friends."

For college scooter commuters like Sophia, the audio-guided helmet can be a perfect solution. It can feature intelligent guidance based on environmental detection and computer vision, while also ensuring safety protections since the physical form factor of a helmet can provide effective protection for non-motorized vehicle riders when encountering crash (*Bicycle Helmet Statistics*).

With this in mind, we have collectively formed the design idea to install a sonification system along with an environmental obstacle detection sensor on the helmet to satisfy both purposes. We decided to build up an interactive scooter simulation within unity to conduct a series of user studies. After a series of design tweaks and modifications, we evaluated our design under different real-world lighting scenarios. We also compared the effectiveness of our sonification system with a toggle to mute or unmute all artificial sound cues.

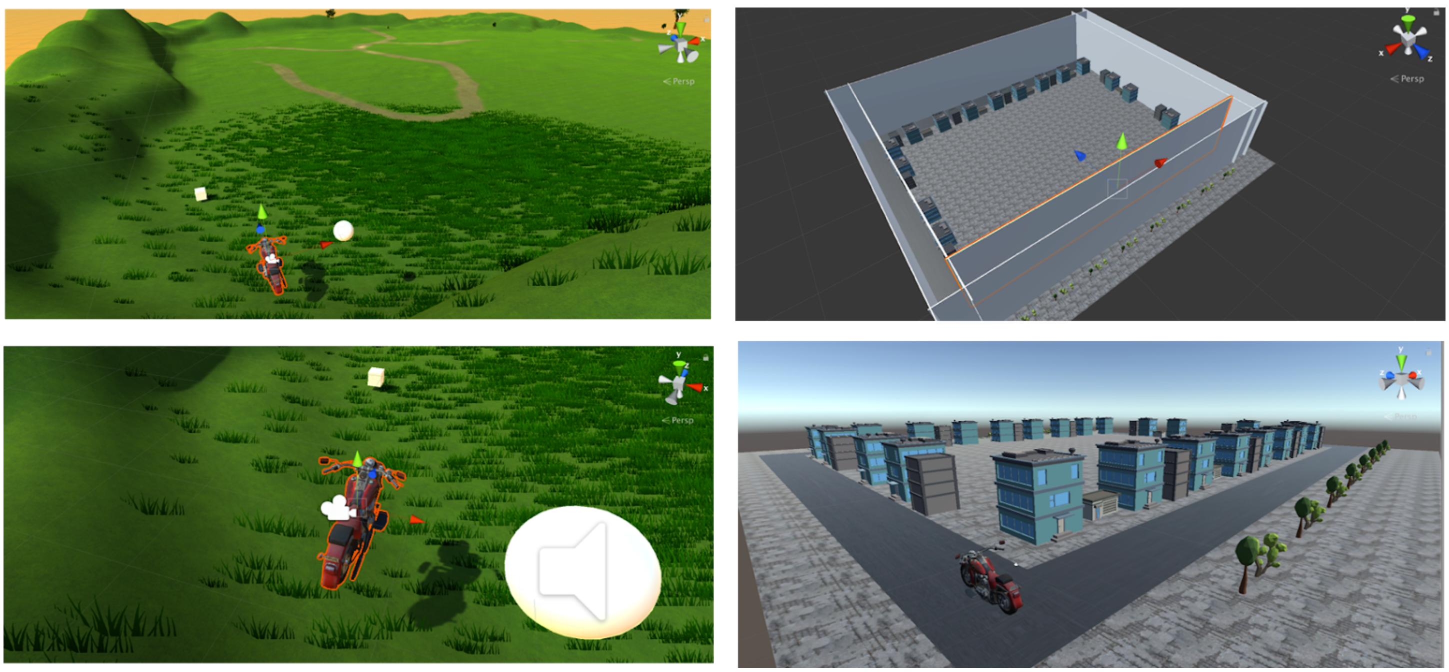

Figure 3. Storyboard: Evolution of Environmental Design

Our project evolved through four phases. The first phase is the initial design, which incorporates primitive environmental detection and sonification by changing the frequency of sound waves based on distance to the obstacle. To achieve this, we designed a simple demo, where the audible cue is dynamically adjusted as a simulated ball rolls on a grass field. In other words, we only sonified the distance between the scooter and one fixed obstacle in the scene in the initial milestone. The sonification was a simple sine wave with varying frequency and loudness, and we found it to be unpleasant and distracting through internal testing.

To address our internal feedback (and many others coming as we progress further along the line), the sonification scheme itself evolved significantly from milestone to milestone. In the second milestone, we decided to sonify differently-sized obstacles with different sound signals and effects, starting with small obstacles, medium-sized obstacles, and ending with large obstacles. Because living animals or human beings deserve extra attention, we intentionally selected a very intrusive and squeaky, high-pitched sound for sonifying living obstacles.

After testing these signals and presenting them in class, feedback indicated that our intentional amplification of living objects may not be optimal. Because it is way too disruptive, it can actually distract riders from understanding and observing the road conditions. As a result, we have chosen a significantly less disruptive sound signal for live animals and human beings, opting for sonifying living obstacles with snippets of a duck quacking. We also added an additional type of sonification for moving vehicles on the road, represented using auditory icon built through a continuous stream of car horn noises.

In addition to improved audio encoding of information where we change the frequency based on the speed of incoming obstacles, the second revised design features improved visuals as well. We replaced the ball with a motorcycle, upgrading our design into one with better looks and more realistically behaviors. We replaced the grass field with buildings and roads, constructing them into a neighborhood of streets and avenues.

Finally, we added direction sonification before the obstacle type sound, and use a priority queue to process and integrate the sonification of all incoming obstacles. This allows us to differentiate and discriminate against obstacle size to live, small, medium and large surrounding objects. For the priority queue, live objects like humans and animals have the top priority, followed by large obstacles like cars, medium obstacles like water puddles, and small obstacles like rocks.

It also became clear through feedback that the simulation, as it was in the third stage, was too overcrowded with busy obstacles that do not accurately reflect riding a scooter in a real world.

To combat this, we streamlined the goal of the game - to travel from one end of the street to the other end, adjusted the acceleration curve of the scooter, enabled it to tilt slightly toward the left and the right when hitting a non-detrimental obstacle (such as a wall, rather than a moving vehicle on the road).

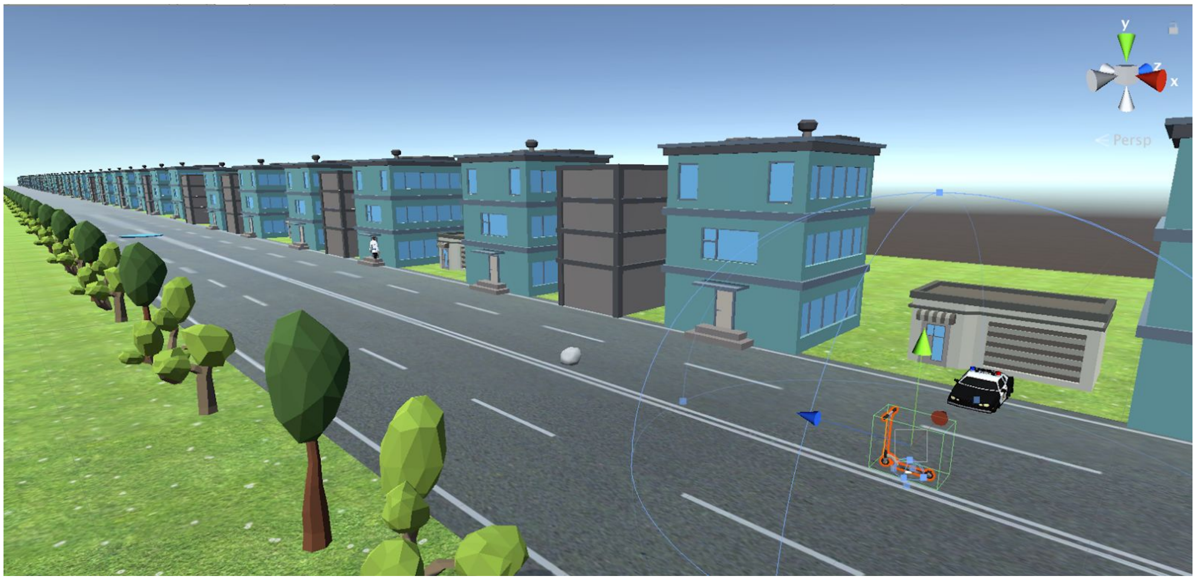

Figure 4. City View Design

To craft a more immersive simulation experience from end-to-end, we also created several eye-candy UI elements to make and the whole program more polished. The player can guide themselves from the starting scene to the ending scene. They can also finetune the behaviour of the world, or to restart the experiment at their own pace.

Figure 5A. Starting Scene

Figure 5B. Ending Scene

After seeding the tentative prototype design to a small group of testers in the participatory design exercise and presenting the prototype in class, we have gathered some additional feedback. For example, the project does not yet consider low-vision scenarios, and visual aspects of the obstacles are too obvious. To address this, we had set up a nighttime lighting environment with much worse visibility of the road and moving vehicles on the road to better simulate low vision scenarios. This allows us to address real-world scenarios where the users are unable to 'see' the obstacles, forcing users to solely rely on the audio cues to react to incoming obstacles. To make these additions more accessible, we added an in-game "pause" menu, allowing players to change the scene, or to switch between muting all sonifications and unmute all sonifications without ever leaving the simulation. In addition, daylight and night mode are also accessible in the menu - this allows us to evaluate how users adjust to different visibility conditions, enabling data collection for both scenarios: a better, clearer sight in daylight mode, and a more ambiguous and unclear sight of the surroundings in the night mode.

Figure 6. The menu when the simulation is paused

The earcons and auditory icons are played in loops with increasing or decreasing frequency to signify the distance between the rider and the obstacle. When a new obstacle is discovered by the virtual helmet, we dynamically switch the playing earcon or auditory icon to the one that reflects the new obstacle. Because different types and categories sound signals are initially not perfectly aligned, the user experiences a jarring transition between the signal for one obstacle type to another. To resolve this, we lined up all sound signals to the rising and falling beat to the best of our ability in a non-linear editing software and adjusted the playback speed of certain signals by modifying the velocity ramping curve for ones that cannot easily match up with others. After re-exported the loops to bring back into the Unity prototype project, we have discovered a much smoother transition between the sound signal of one type of obstacle to another.

With users previously only able to identify the type of obstacle they are dealing with but not the direction obstacles are from, they have rightfully found the lack of orientation-aware sonification a bit frustrating. Therefore, in addition from making obstacles distinguishable in a 2-dimensional perspective (via proportionally increased frequency as distance decreases), we also added direction as another aspect of designing the sonification, sonifying the orientation of the obstacle, categorizing it into the following: front, back, left-front, left-back, right-front, and right-back. While we sonify both whether the obstacle is in the front or at the back, and to the left or to the right in a majority of cases, we have concluded during discussions in design meetings that in some cases, left or right signals can be redundant. More specifically, when the angular difference between the scooter and the obstacle falls between 88° and 92°, we intuitively perceive the rider and the obstacle to be in the same “straight line”, and remove the “left” or “right” qualifiers form the speech signal.

Through user feedback in later stages of prototype design and development, we received several suggestions that the specific phrasing of “back” is confusing to riders. A user phrased his doubt with the following response”: “does ‘back’ mean I should move toward the back, or should I avoid what’s coming from my back?” To clarify the meaning behind “back”, we replaced it with a synonym with much higher clarity, “behind”. In case this is still not intuitive and obvious enough, we recorded 6 additional variants of each existing orientation speech, appending “obstacle” before each orientation or orientation pair. Because the extended variants (i.e: “obstacle behind”, or “obstacle left-back”) are more than 40% longer in terms of time duration compared to the original recordings, we realized riders may quickly get tired of the “obstacle” prefix. As a result, we only play the variants with the “obstacle” prefix during the first four speech signals - after four consecutive “hinted” speech signals, the prefix is dropped to reduce the speech’s cognitive load and keep the rider clear of redundant phrases.

Finally, while the background ambient sound is not a primary part of the auditory design of our project, it is nonetheless important in creating a realistic simulation environment. We evaluated over fifteen pre-recorded stock audio clips from roads of varying traffic - and eventually determined on one that features common sounds that resemble college students’ real-life scooter rides the best: pedestrians chatting in multiple languages (which are very common in college campuses), slight noises of car wheels contacting the ground and generating friction noises, music playing through portable Bluetooth speakers, then created a seamless loop through ocenaudio.

In all, we carefully considered and selected unique earcons and auditory icons for all obstacles. The same level of care and polish extends to the overall simulation system, including the sound effect of various UI states. For example, for the starting and ending scenes of the simulation, we added sound snippets from the Star War movie and the Super Mario's game. For alerting obstacles to the user, we use auditory icons such as duck quacks to mimic familiar, natural auditory sounds that already exist in the real world.

Unity Prototype

We delivered our project prototype using the Unity SDK. Essentially, we created a Unity simulation as our prototype - with sonification layered on top of ambient soundtracks, simulating how our proposed helmet would improve the safety of scooter transportations in the street.

We created a cityscape scene in Unity as the base world map for evaluating various sonification techniques - in other words, it all begins with a city storyboard. This Unity simulation revolves around an emulated experience of riding a scooter in a busy city. As the user rides on the road, he or she should avoid human, animals and various obstacles by carefully listening to various sonified cues and hints. The scooter is modeled against a common model seen in the real world, namely the “Harley-Davidson scooter”. A custom C# script named playerController allows the player to control the scooter from a first-person perspective. Several obstacles, including but not limited to a police car, a walking kid, and a water puddle all represent distinct situations that we might encounter on a city road. Because obstacles appear in a dynamic and animated fashion, the rider’s encounter to these obstacles can feel “alive” and genuine.

Figure 7. Example of city storyboard: the scooter encountered police car in the first-person view

Moreover, as detailed in the final paragraphs of the previous section, “Design Protocol and Evolution”, we employed multiple sonification soundtracks to inform the user where obstacles may lie ahead. The sonification system can help users predict the distance and direction to the obstacles so that users can take actions before colliding into them. More specifically, these cues are implemented as AudioSource components, which are then controlled by the playerController script. We modified the script so that it encoded two types of output audio sources: one represented distance and another represented direction. We derived the formula of “pitch = 1 + (200 - distance) / 550” to calculate the frequency of repeated sonification. This way, as the scooter moves closer to the obstacle, the frequency increases to warn the rider of this shortened distance.

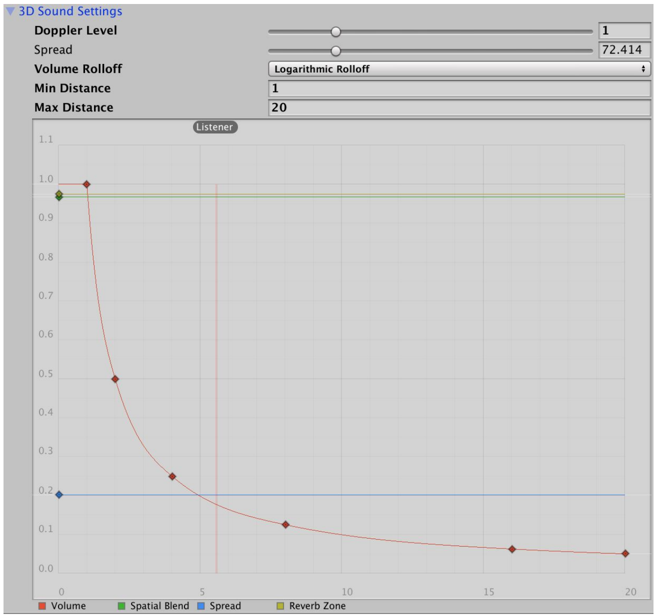

Figure 8. The relationship of sound via distance

In addition, we provide more obvious speech-based cues for the initial discovery of each obstacle. While short and succinct, these cues are meant to be obvious and undisruptive. As detailed and justified in the “Design Protocol and Evolution” section, it is divided into four main categories (and 15 unique variants), including but not limited to “left-front”, “left-back”, “right-front”, and “right-back”. This allows the user to be informed about the orientation of various obstacles to successfully avoid them.

Final User Study and Analysis of Statistical Results

We have conducted several simulation and gameplay sessions throughout the design process to evolve our sound and interaction design, gathering qualitative feedback through brief one-to-one interviews. The outcomes of these shorter studies are documented and elaborated earlier in this document, under the “Design Protocol and Evolution” section.

This particular section focuses on discussing the final user study, which we conducted after finalizing the prototype design and implementation. The final user study primarily focuses on gathering quantitative results with the scientific testing method, evaluating its effectiveness with experimental groups and control groups. We divided our target users into four categories:

- The first group played the game in no-sound daylight mode.
- The second group played the game in sonification system in daylight mode.
- The third group played the game in no-sound night mode.
- The last group played the game in sonification system in night mode.

For each group, we counted the total distance they traveled and gathered feedback on their view on various features of the sonification system. To gather more quantitative responses, we sent them another Typeform Survey.

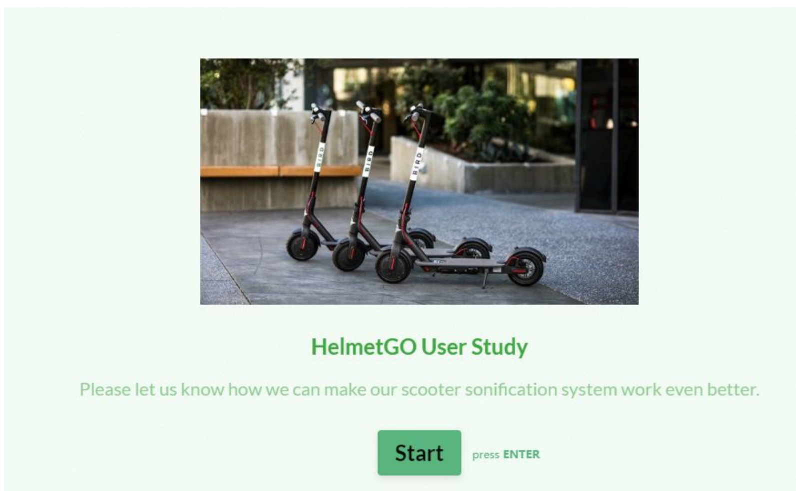

Figure 9. HelmetGo User Study Welcome Page (Typeform link: https://soniahuang1.typeform.com/to/UkQYoz)

In the final user study survey, we asked our users which specific sound notification helped them avoid obstacles. Many participants chose the duck quacking sound as the most helpful auditory icon. The questionnaire also asks about players' feedback on the effectiveness of sonifying distance through the change of frequency. Whether spoken directions in the sonification are helpful or distracting is also covered by the questionnaire.

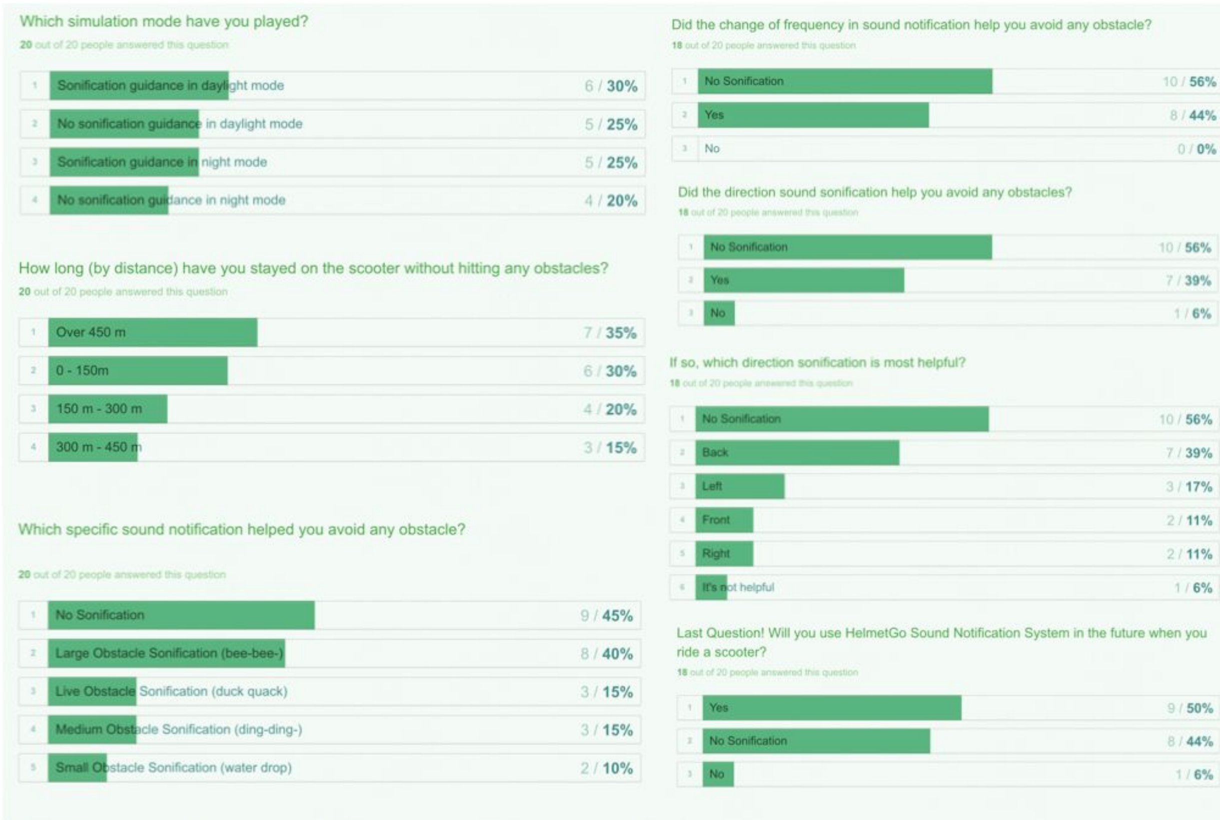

Figure 10. User study results specific to the simulation experience

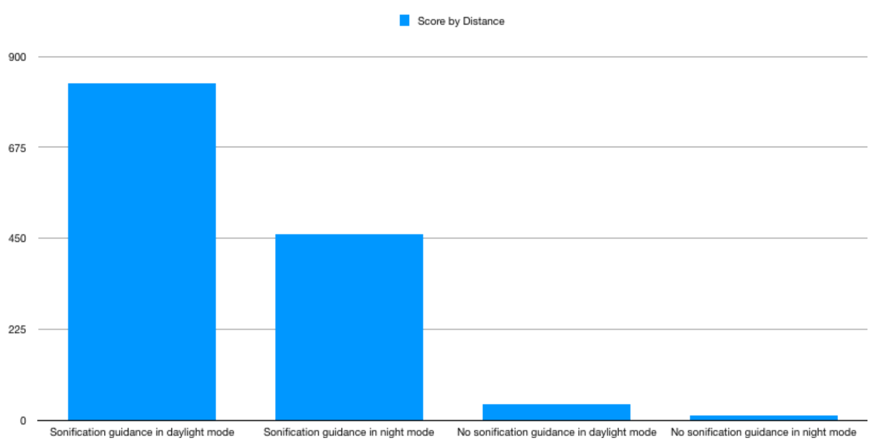

Figure 11. User Study Statistics

By analyzing telemetry data, we have discovered that the second group (sonification, daylight) statistically aced the simulation, while the fourth group (sonification, night) came at a close second in terms of overall performance, which are measured by the total distance safely traveled before hitting an obstacle. Then comes the first group (no sonification, daylight), and the third group (no sonification, night) comes last. This is mostly aligned with our expected results, and it is worth noting that the experiment suggests having sonified audio signals can compensate for the lack of visibility during night time - which suggests our experiment to be successful. On a related note, for every participant from the second group and the fourth group (a.k.a: users who participated in the simulation with sonification turned on), regardless of whether they rode in daylight mode or night mode, indicated the change of frequency in sound sonification has helped them avoid obstacles. These combined observations helped us verify the effectiveness of our approach to convey distance through frequency variations.

In addition to surveying about the generic prior experience of riding scooters, participants of the user study also answered questions specific to the effectiveness and usability of our prototype solution. Notably, among all users who traveled on the scooter without hitting any obstacles for over 450m (35% of all participants), all those 7 users were participating in the simulation with sonifications - where 4 are participating in daylight mode and 3 are participating in night time mode.

Last but not least, we tasked participants with the ultimate question - “How likely will you use the HelmetGo Sound Notification System when they ride a scooter?” Surprisingly, although not completely shocking to all of us, is that 95% of all participants indicated they will likely make use of our guidance system if it is ever fully developed and becomes production-ready.

Evaluation and Conclusion

The usability of this sonification system is validated through the results of user testing (Figure 11). Using the designed sonification system, participants’ performance significantly improved, regardless of the overall brightness or visibility. This is important because it proved that the designed audio system is helpful in terms of reducing the risk of a crash or an accident. Although the experiment is not conducted in real life but is rather simulated, we believe the Unity project incorporates enough realism to conduct a useful experiment. Cars coming behind riders or pedestrians, young kids running on the street, and squirrels crossing the street are all common encounters when navigating on the road.

However, when comparing between day and night mode both with sonifications, the results from night mode is only about ⅔ as good as it is under day mode. While it is still not ideal, it does offer tangential improvements to the rider's safety - considering the vision is limited. Assuming our participants all share similar motor skills, are similarly experienced in operating a mouse, this tells us the pros and limitations of our approach. Because we conducted the final user study with participants who have never participated in prior work-in-progress test sessions, and that everyone only goes through the simulation once in one mode, we are able to eliminate most of the bias and unfairness - it is not possible for a participant to simply score higher due to experience in a prior testing session (because everyone only runs through the experiment once).

However, the design project is still, after all, a simulation - which is not, and can never be fully representative of real-life scenarios. Besides, the current user study is limited in its sample size - where larger control groups and experiment groups, as well as varying levels of light sources (rather than only two - daylight and night light) can tentatively improve its accuracy. If we are to conduct further research, more data should be gathered for more accurate results.

Moreover, we can also improve the Unity simulation game to enhance its realism, which can be achieved in the following ways:
(1) the appearance of objects so it looks more realistic;
(2) the comprehensiveness and complexity of moving objects;
(3) scooter control and lights so the user can better manipulate and perform.

If we are to commercially build and design the hardware and software for this helmet sonification system, rather than merely emulating the experience in Unity, we should conduct real-life experiments with participants in a safe and controlled, but still representative manner. Deploying helmets with our sonification alert system built in can still be a massive undertaking with even more unique engineering and design challenges. Nonetheless, given the limitations of our software-based simulations and the scope for our user studies, we believe that our results show that the HelmetGo wearable sonification system has the potential to successfully allows novice and avid college student to commute more safely via scooters.

Appendices and Deliverables

Sound Files:

The following sound files are generated from Google's text-to-speech engine. They are used in early versions of the Unity prototype, but are eventually replaced with more natural-sounding ones after feedback from potential users. There are only 4 variations of this primitive text-based read-aloud sonification.

CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/front.wav

CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/back.wav

CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/left.wav

CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/right.wav

Later, the recordings are replaced with more natural-sounding counterparts, generated using the Siri voice on an iOS device with the Shortcuts app from Apple. The recordings are extracted from a screen recording, since there is no direct audio file output support on iOS. We have recorded 15 brief sound cue variants specific to each and every scenario - with varying lengths and levels of details. They are located under: CS_4590_HelmetGo_Prototype/Assets/tts

Onboarding only: "Obstacle [direction left or right] [direction front or back]"

- CS_4590_HelmetGo_Prototype/Assets/tts/obs_left_front.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/obs_left_back.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/obs_right_front.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/obs_right_back.mp3

Onboarding only: "Obstacle [direction front or behind]"

- CS_4590_HelmetGo_Prototype/Assets/tts/obs_behind.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/obs_front.mp3

After onboarding has ended: “[direction left or right] [direction front or back]”

- CS_4590_HelmetGo_Prototype/Assets/tts/left_front.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/right_back.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/left_back.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/right_front.mp3

After onboarding has ended: “behind”, “right”, “left”, “front”, and “in front’

- CS_4590_HelmetGo_Prototype/Assets/tts/behind.mp3
- CS_4590_HelmetGo_Prototype/Assets/tts/right.mp3 (unused)
- CS_4590_HelmetGo_Prototype/Assets/tts/left.mp3 (unused)
- CS_4590_HelmetGo_Prototype/Assets/tts/front.mp3 (unused)
- CS_4590_HelmetGo_Prototype/Assets/tts/infront.mp3

Additionally, we sonify the size of the obstacle while identifying whether they are alive with the following audio clips. They are obtained from freesound.org, and the specific author (original uploader) of the audio clips are listed in detail under the “Reference” section. Audio clips are normalized, synced up and looped for them to fit well together in our project.

- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/large_obs.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/large_obstacles_rep.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/live_obs.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/live_obstacles_rep.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/medium_obstacles_rep.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/mid_obs.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/small_obs.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/small_obstacles_rep.wav

Other miscellaneous sound clips used includes the background loop for the street’s ambient sound, Star Wars’ Main Theme audio for the onboarding screen, and a sound effect extracted from Super Mario Bros for the “Simulation Over” screen.

- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/background.wav

- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/Star Wars Main Theme (Full).mp3
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/audio.wav
- CS_4590_HelmetGo_Prototype/Assets/AudioBatchA/smb_mariodie.wav

Software: Source code of our deliverable software is located under CS_4590_HelmetGo_Prototype_Source. It is compatible with the Unity SDK of version 2018.2.6f1 or later. A pre-compiled executable macOS binary is available as CS4590_HelmetGo_Prototype.app in the zipped project folder.

Video: A screen capture, alongside with in-line keyboard video feed and closed caption is available under CS4590_HelmetGo_Recording.mov, which is under the root directory of the project.

Tables of data: Refer to the "User Study" section for tables of data, as well as visualized diagrams and charts.

Note: many appendix items (redesigns [page 6 and 8-11], storyboard [page 5, figure 3], user survey tables[page 3 and 13-14], and diagrams [page 14]) are already incorporated in-line in the main report, hence they are not repeated again in this section.

Member Work Contribution:

All:

- Participated in the evolving project design
- Participated in the participatory design exercise
- Contributed to the presentation slides, the progress report, and this final report

Xueying Huang:

- Ran and recorded pre-project surveys
- Synthesized character movement in the virtual world

- Manually added and pre-programmed obstacles as needed

Siyu Li:

- Set up the camera in Unity and wrote code for collision detection
- Designed and created distance frequency correspondence
- Designed and implemented UI elements

Ruixuan Sun:

- Simulation environment setup in Unity
- Collected and edited obstacle audio effects
- Ran and recorded the participatory design exercise

Luming Yin:

- Integrated audio signal manipulation into the Unity project
- Designed and evaluated several sonification schemes
- Added directional sound and in-game menu for sound mode toggles and light switches

Mingyuan Zhou:

- Created and adopted vehicle models into the virtual world
- Integrated audio into various Unity scenes
- Compiled and analyzed user study data

Build & Running Instruction

Running the project:

We are supplying a pre-compiled executable version of our project for macOS, under the name of as CS_4590_HelmetGo_Prototype.app. Double click on the app package to launch it on a compatible Mac. To run on Linux or Windows, either build it first in Unity and run your compiled executable (preferred approach - there's no frame drops or performance issues if you compile it first), or launch Unity, and navigate to Assets -> Scenes -> start, then click the "Play" icon (running the simulation directly in the Unity editor may lead to frame drops or performance issues).

Building the project:

On a compatible computer running Unity 2018.2.6f1 or later, open the CS_4590_HelmetGo_Prototype_Source project folder. Under File -> Build Settings, make sure “Scenes/start”, “Scenes/SampleScene” and “Scenes/endScene” are included in “Scenes In Build”. Then click “Build”.

To use the project to test out sonification mechanisms:

To test out our design, first launch CS_4590_HelmetGo_Prototype.app. You will then be greeted with a resolution selection pane. After selecting a resolution based on the configuration of your computer, you will be presented with a welcome screen titled “Helmet Go”.

Instructions for the simulation experience:

- On the welcoming screen, you will see three buttons: “Start”, “Start without Sound”, and “Exit”.
- Click on the Start button to start and you are controlling a Harley-Davidson scooter in the first-person view. To run the scooter-riding simulation with only ambient sound and without any sonified audio cues, click on “Start without Sound”.
- After the simulation has started, You can control the scooter with Up, Down, Left and Right keys on your keyboard to control the scooter’s moving direction while avoiding crashing into any obstacles.
- During the game, click the escape key on your keyboard. Then, you will see five options: Mute/Unmute, Night Mode, Daylight Mode, Exit and Restart.
    - Click Mute/Unmute to switch the game with No Sound mode and Sonification Mode.
    - Click Night Mode to switch into a darker scene. Click Daylight Mode to switch back.
    - Click Exit to shut down the sonification system.
    - Click Restart button to return to the start page.

- If you hit an obstacle, you will go to an End scene of “You Crashed”. Click Restart to go to the onboarding page and start again and click “Exit” to quit the sonification system.

(References continued on next page)

References

"A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System." *NHTSA*, U.S. Department of Transportation, 15 Dec. 2016.

"Animals FREE Pack." *Unity Asset Store*, Breviceps, 1 Feb. 2014.

"Beep on highway at 50-80 km/h." *Freesound*, CarlosCarty, freesound.org/.

"Busy road at urban city" *Freesound*, zaem, freesound.org/.

Crusader, Safety. "Bicycle Helmet Statistics." *Helmet Statistics*, Helmets.org, 12 Nov. 2018, helmets.org/stats.htm.

"Game Over Sound Effect Loop." *Freesound*, Penetrermind, freesound.org/.

Gordon, Kyle. "Navigation Devices & Usage." *Statista*, www.statista.com/topics/2221/navigation-devices-and-usage/.

"Harley-Davidson Model." *Unity Asset Store*, EverydayGameObjects, 2 Sep. 2017.

Ink, Social. "Bike Share in the U.S.: 2017." *National Association of City Transportation Officials*, nacto.org/bike-share-statistics-2017/.

"Kajamans' Roads." *Unity Asset Store*, CC4SL, 2 Sep. 2015.

"Opening Scene." *Freesound*, klankbeeld, freesound.org/.

"Poly City - Free Cartoon Pack." *Unity Asset Store*, VFX Master, 15 Aug. 2016.

"Rock Prefab." *Unity Asset Store*, MeshDiffuse Pro, 21 Jan. 2018.

"Star Wars Main Theme (Full)" *Freesound*, fonsflorance, freesound.org/.

“Turn Data Collection into an Experience | Typeform.” *Online Survey for HelmetGo*, CS4590 HelmetGo, www.typeform.com/cs4590-helmet-go/.

“Vehicle - Pro 3D Models.” *Unity Asset Store*, Pro 3D Models, 31 Nov. 2014.

“Rock at waterpond” *Freesound*, jsharick, freesound.org/.