Statement of Originality

Statement of Original Work: I certify that the concept of the prototype, part of the prototype scene in Unity, and the logic on the interactions between circuit playground's movement and the Unity screen is my own original work. In the construction of this prototype, I also used various external resources, which include the race car object from the asset store, Unity online tutorial, YouTube tutorials, the help from the Unity forum and the tutorials form the Adafruit website. All the external sources I used in prototype two construction will be listed below.

[1] Black race car object in prototypeScreen : PP Super Car 01, Pink Planet Studios; retrieved from [Unity Asset Store] https://assetstore.unity.com/packages/3d/vehicles/land/pp-super-car-01-228682, Last Accessed 29/8/2022

[2] The code that help the camera follow the car object in Camera.cs : Unity tutorial, retrieved from #3 Add an offset to the camera position at https://learn.unity.com/tutorial/1-3-make-the-camera-follow-the-vehicle-with-variables?uv=2020.3&projectId=5caccdfbedbc2a3cef0efe63# , last accessed 31/8/2022

[3] The codes that obtain the keyboard inputs in DriverControl.cs: Unity tutorial, retrieved from https://learn.unity.com/tutorial/lesson-1-4-use-user-input-to-control-the-vehicle?uv=2020.3&projectId=5caccdfbedbc2a3cef0efe63#5cbe3969edbc2a191e639152, last accessed 31/8/2022

[4] The codes that move the vehicle in DriverControl.cs: Unity tutorial, retrieved from https://learn.unity.com/tutorial/1-2-move-the-vehicle-with-your-first-line-of-c?uv=2020.3&projectId=5caccdfbedbc2a3cef0efe63#5ce33718edbc2a232e231e45, last accessed 31/8/2022

[5] The code the change the vehicle direction in DriverContro.cs: Unity tutorial, retrieved from https://learn.unity.com/tutorial/lesson-1-4-use-user-input-to-control-the-vehicle?uv=2020.3&projectId=5caccdfbedbc2a3cef0efe63#, last accessed 31/8/2022

[6] Port setting for unity and Arduino communication in Camera.cs: Class Tutorial, retrieved from https://learn.uq.edu.au/bbcswebdav/pid-8211376-dt-content-rid-47444573_1/courses/DECO2300S_7260_61655/2022-W07%20Tutorial.pdf, last accessed 22/9/2022

[7] Receive port values code in Camera.cs: Class Tutorial, retrieved from https://learn.uq.edu.au/bbcswebdav/pid-8211376-dt-content-rid-47444573_1/courses/DECO2300S_7260_61655/2022-W07%20Tutorial.pdf, last accessed 22/9/2022

[8] Find all the objects with same tag code in Camera.CS: Unity Forum,retrieved from https://answers.unity.com/questions/50208/loop-through-all-objects-with-same-tag.html, last accessed 22/9/2022

[9] Change object color code in Camera.CS: Class Tutorial, retrieved from https://learn.uq.edu.au/bbcswebdav/pid-8211376-dt-content-rid-47444573_1/courses/DECO2300S_7260_61655/2022-W07%20Tutorial.pdf, last accessed 22/9/2022

[10] camera shaking code in camera.cs: Youtube Tutorial, retrieved from https://www.youtube.com/watch?v=BQGTdRhGmE4 ,last accessed 22/9/2022

[11] Port Setting code in P2Arduino: Class tutorial, retrieved from https://learn.uq.edu.au/bbcswebdav/pid-8211376-dt-content-rid-47444573_1/courses/DECO2300S_7260_61655/2022-W07%20Tutorial.pdf, last accessed 22/9/2022
[12] accelerometer code in P2Arduino: Online tutorial, retrieved from https://learn.adafruit.com/circuit-playground-bike-glove/hand-position-and-accelerometer , last accessed 22/9/2022
[13] Set LEDs color in P2Arduino: Class Tutorial, retrieved from https://learn.uq.edu.au/bbcswebdav/pid-8211376-dt-content-rid-47444573_1/courses/DECO2300S_7260_61655/2022-W07%20Tutorial.pdf, last accessed 22/9/2022

Basic Details

Name: Chi Heng Jeffrey Hui
Student number: 46587590
Practical session: Thursday 10am
Tutor: Tahlia Slater
Document title: Prototype 2 Statement of Delivery

The Concept

Long distance driving is very common for most of us, and most of us may experience once before. If you were the driver, you may realize driving long hours is very tired. Especially during night times, you can fall asleep easily. In fact, there are many accidents relate to fatigue driving, some of them involve minor injuries, but some of them are fatal. This is a serious problem, so we should handle this properly. To solve this issue, we need to utilized technologies and design a digital solution to ensure drivers are fully conscious while they are driving, for instance, a system that can keep drivers awake when they don't pay attention to the road. To maintain the effectiveness and a good user experience of the application, it needs to provide different feedbacks based on different scenarios. In example, the application will provide temperature and sound feedback when the detected fatigue level is low. Otherwise, it will provide vibration and aggressive sound feedback when the detected fatigue level is high. By implementing this concept to real life, it can reduce the accidents caused by fatigue driving, so that it can protect everyone on the road.

The Purpose of this testing round

Most application requires input from user, and the system will provide outputs afterwards. In the past, many systems require users input data manually. As technology keeps evolving, automated data input becomes a trend for many applications. Although automated data input brings us a lot of convenience when we use an application, but I also realize automated data input and detection don't always perform very well, which can make things hard, for instance, although Tesla is one of the leaders in self-driving cars, but there are many accidents reported in the last couple of years. It shows there are tradeoffs between convenience and safety while we try to utilize the usage of automation.

In my driver warning system concept, it also involves some automated features. For instance, the warning system will take users eyes movements and head movements as the input and the system will make decision based on these inputs. If the system detects information correctly, it can bring convenience to users. On the other hand, if the collected information is incorrect, it may affect user experience and driving safety, and we can see the tradeoffs between convenience and safety are common decision we need to make across all automated applications.

The first and second paragraph point out the dilemma we are facing, which leads to the objective of prototype 2. In prototype 2, it is to evaluate whether automated data collection as input is a more effective approach, or there are better approaches to handle the data input process in my concept, for instance, trigger the system manually. This is also the research question for prototype 2. We also need to make assumption for the research question, in this case, my hypothesis is "Automated data collection is not always the best way to generate input data for a system, there are many uncertain factors to deal with before it turns to a usable system for real-life application.

Through people testing the prototype, I want to find out how accurate the automated data collection system performs in my concept. For instance, if the driver didn't move, does the system make wrong judgements and count it as moving? In addition, I also interested in finding the tolerance level on wrong data detections and the preferred data input method from testers.

To meet the requirement of the evaluations, there are things I need to include and exclude during the construction of the prototype. In the prototype, I need to include clear signals which can differentiate if the warning is triggered or not. This is very important, since the testers will use this information to evaluate the performance, which is one of the goals of this prototype. On the other hand, we should not include any unnecessary components in the prototype, since it can distract testers, so it may not give us the most accurate results.

The Form of the prototype

As my concept is a warning system on fatigue driving, so prototype 2 will continue using the car driving context I used in prototype one. But prototype 2 will be completely different from prototype 1 since the research question changed from progressive feedback system to the automated data input.

In prototype 2, I want to test whether automated data input is a better approach than other methods. The automated data input I am going to test is the head movements data. Unity, and Arduino are the tools to implement this prototype. I will use Arduino to collect head movements data, and send these data to Unity, which will display corresponding feedbacks to testers.

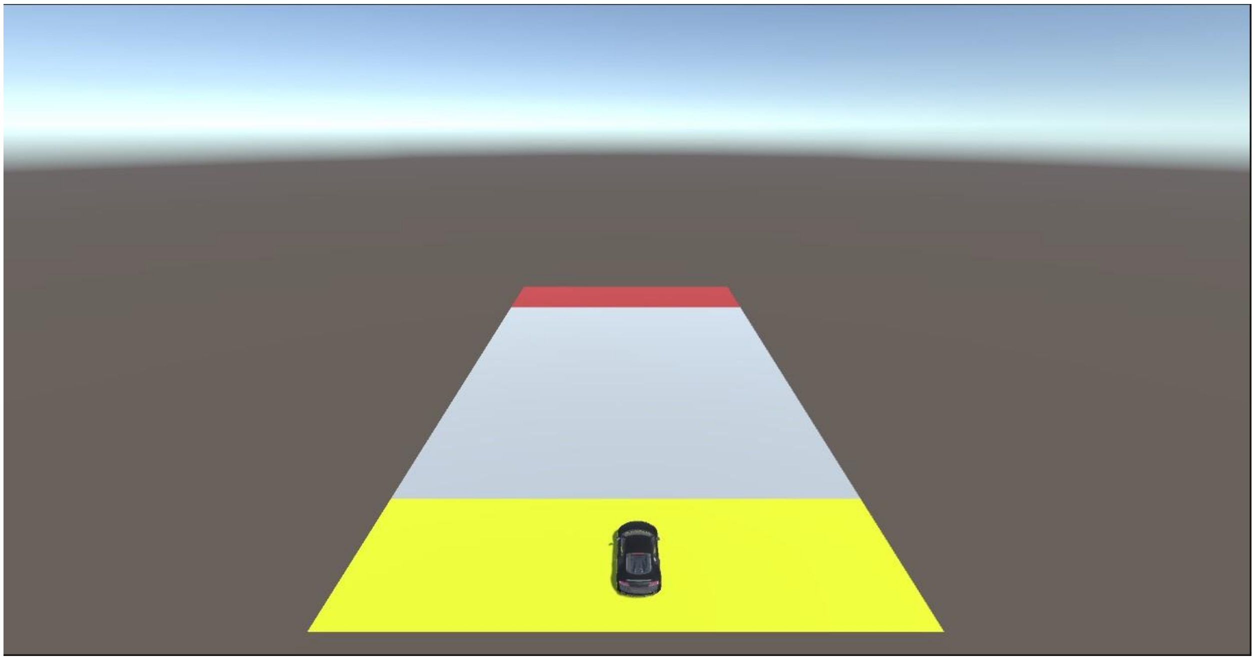
Head movement is one of the indicators to decide whether the driver is falling asleep or not. If a driver falls asleep, there will be abnormal head movement, and driver's head won't keep straight and stable. The circuit playground has a build-in accelerometer, which will be perfect for detecting the head movements. When the driver's head move, the accelerometer values will change instantly, which will be sent to Unity immediately as well. The entire process is programmed, so it is automated and doesn't require testers input. This automated process is exactly what we want to test in the research question.

There are two states to test on the head movement. Firstly, we need to test when the driver is not falling asleep, so the head position should be stable and facing forward. Following, we also need to test the head movement when the driver is falling asleep, which is the abnormal head movements. Testing state one is easy, since we just need testers to keep the circuit playground upright, and don't move. But testing state 2 would be another story since we cannot fully simulate the falling asleep scenario in the testing context. In this case, I can only tell the testers to simulate this state as good as they can. Probably, they can move the circuit playground randomly and observe system responses.

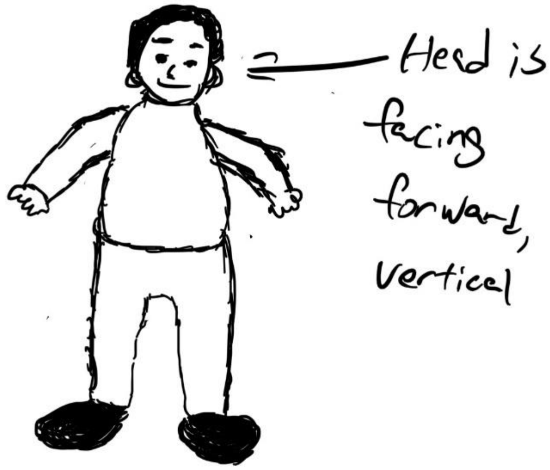
In prototype 2, testers can drive the car. But more importantly, it is about the interaction with the circuit playground. The original position of the circuit playground is upright, which symbolize the driver's head is upright and facing forward. When the circuit playground is placed upright, the LEDs and the driveway in Unity will turn green. Conversely, when the circuit playgrounds aren't upright and create an angle smaller than 90 degrees, it symbolizes the driver's head is moving downwards or start to incline. When the circuit playground starts to incline, the LEDs and the driveway in Unity will turn red. Also, Unity screen will shake, and the circuit playground will ring. This prototype simulates how head movement data are collected automatically from users. By testing this prototype with the testers, I can obtain all the required information to answer my research questions, so that I can make improvements in the future.

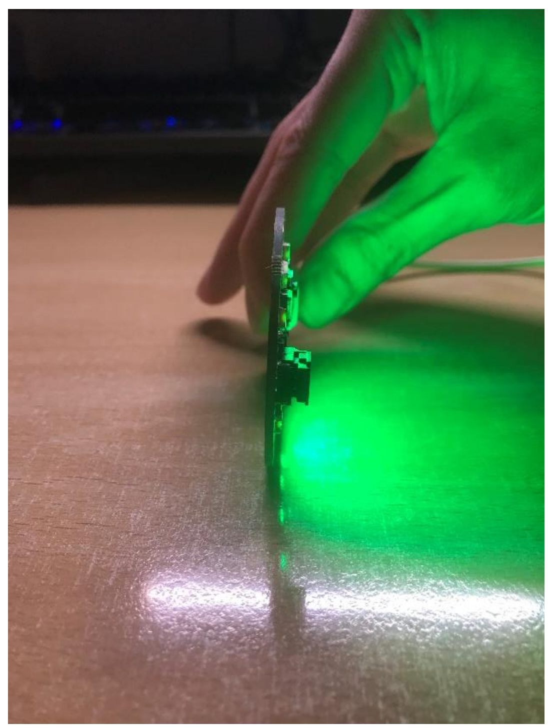
These are some visual aids for better understanding the prototype:

Image 1: This is the Unity screen. It is the settings of the prototype.

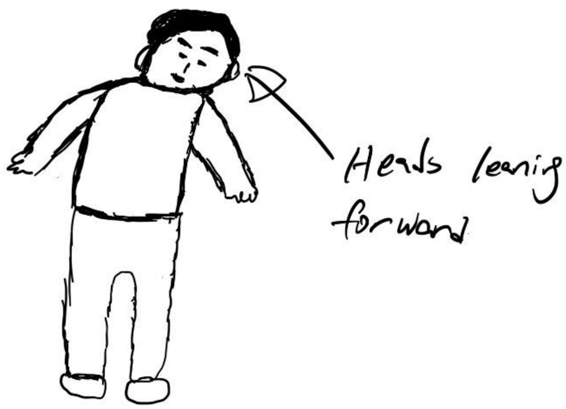


Left is image 2a, right is image 2b. In image 2b, you can see the circuit playground is upright and vertical, which symbolize driver's head is facing forward and vertical, like the position in image 2a.





Left is image 3a, right is image 3b. In image 3b, you can see the circuit playground leaned forward and inclined, which symbolize driver's head learn forward when they fall asleep, which is like the position in image 3a.



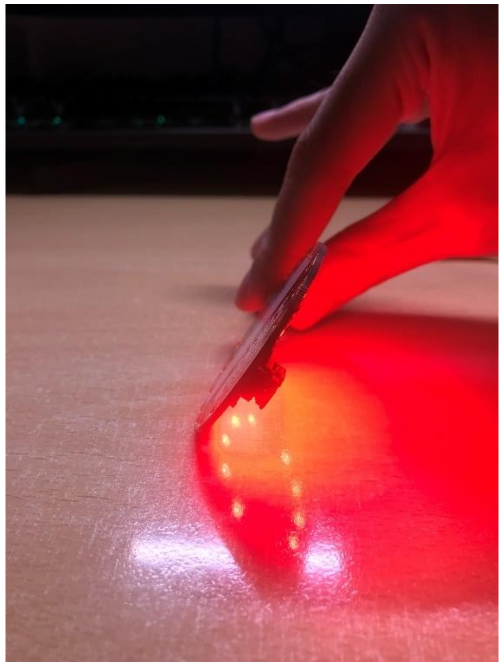


Image 4: When the circuit playground is placed vertically. It means driver facing forward, and paying attention to traffic, so it shows green.

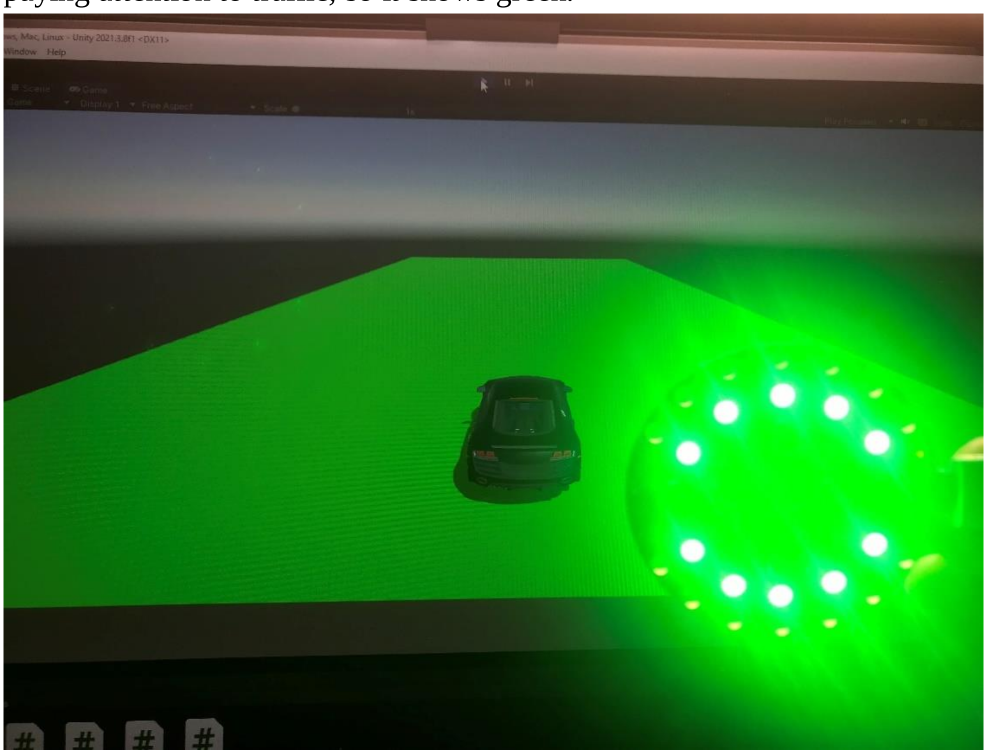


Image 5: When the circuit playground inclined, and create an angle, it means driver’s head keep forward or downwards, which imply driver is not paying attention to traffic, they may fell asleep, so it shows red. In addition, the circuit playground will ring.

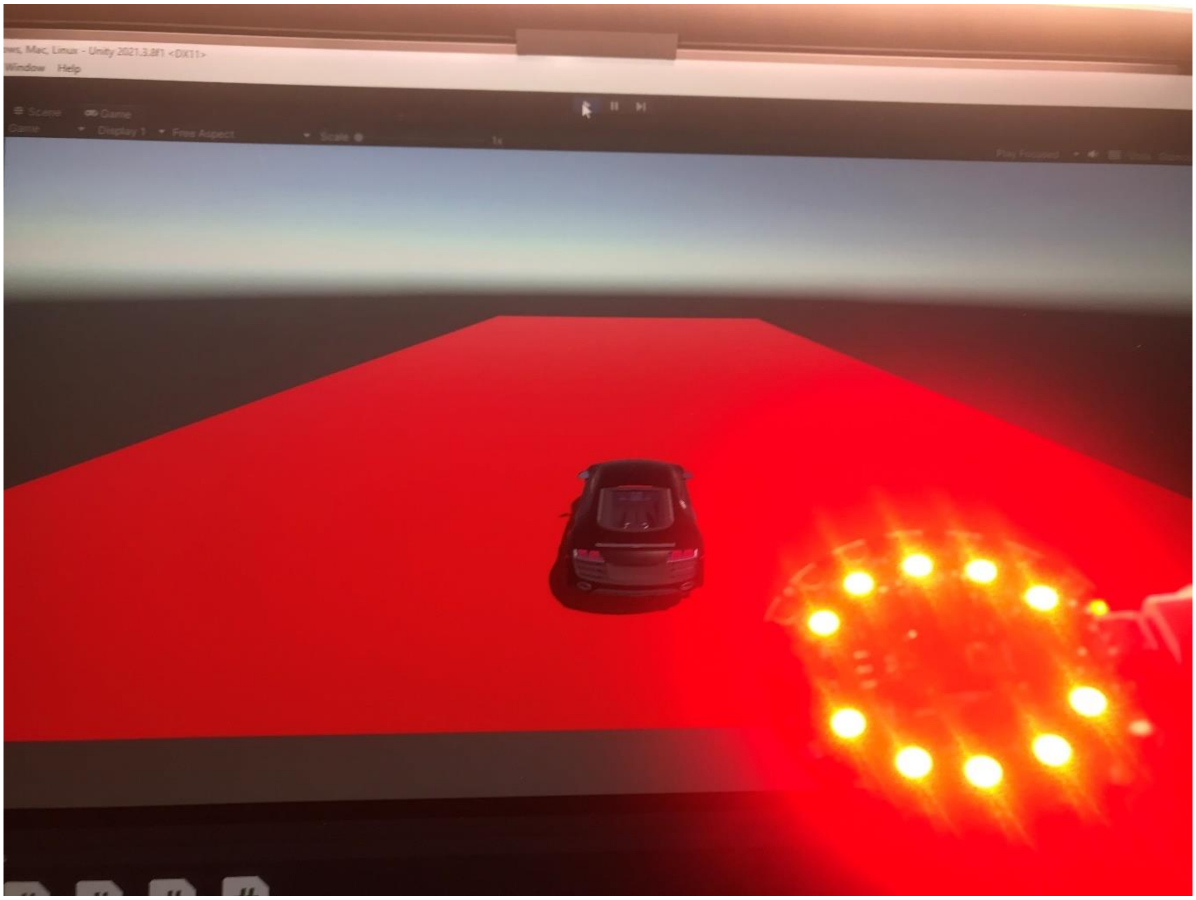
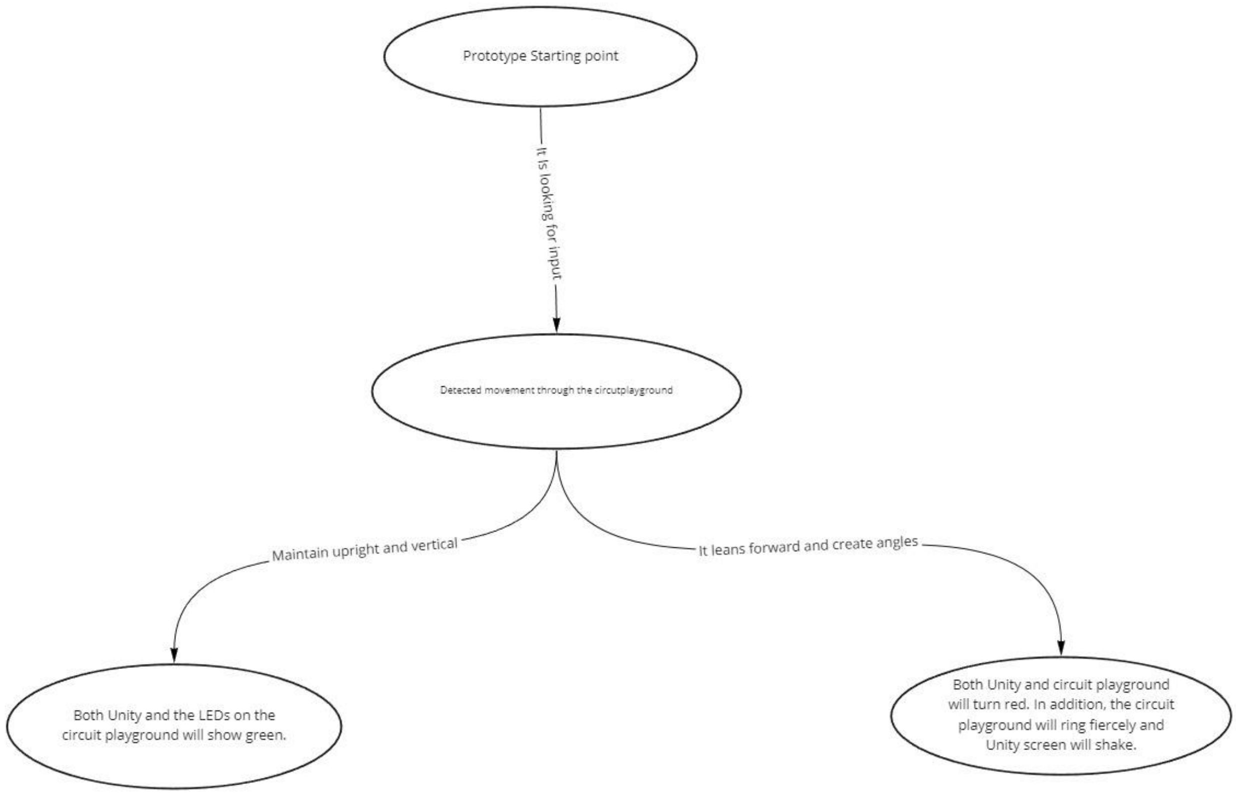


Image 6: This is the interaction diagram which shows the flow of prototype 2.



The Testing Approach

Description:

-what you asked the users to do?

In the testing session, I asked the tester to play with the car. At the same time, I ask the tester to move the circuit playground occasionally. I told them to remember the responses when they move the circuit playground.

Agenda:

1. Firstly, I need to explain the concept to my testers. Then I will tell them what they need to do throughout the testing session with the prototype. Afterwards, I remind my testers to pay attention to the responses when they move the circuit playground.
2. In step 2, I give them time to test my prototype.
3. In step 3, I would conduct an interview with the testers. I will ask them some pre-defined questions, which can help me to understand what people’s thought on this automated data collection system.
4. Give them time to express their thoughts on aspect I didn’t ask.

Process:
When the testers arrive the testing place, I will explain the concept and tell them the intention of this testing session. Afterwards, I will tell them what they need to do and what they need to pay attention. Following, I will give them time to do the test. When the tester finished the testing session, it will enter the interview session. I will ask some question about the experience and performance of my prototype. Finally, I will let them express their thought on other area which I haven't mention.

Method for evaluation:
In prototype 2, I used observation and interview as the methods of evaluation. I picked these two methods since observation can help me to get users responses passively. On the other hand, interview is an active approach. Using passive approach and active approach at the same time can bring us some advantages. During the interview, interviewee may not feel comfortable to express their true thought, so some of their words may not reflect their real thoughts. First feeling is hard to pretend. If we also use observation as the evaluation method, we can compare the result of both methods and validate the real thought from interviewee, which will give us more legit answers to my research question.
Although I will use both evaluation methods in prototype 2's testing sessions, but I will put more effort on interview, and use observation as the secondary method, since interview usually give me more chance to explore the areas, I am interested in.

Evaluation Outcomes & Reflection

Evaluation Outcome:
After observing tester's reaction and interviewing with the testers, I obtained some useful insights from different aspects, which covers the performance of the system, user experience from potential users, and the expectations from potential users.
In terms of performance, most interviewees feel the system is performing adequately. But there are spaces to improve the system. For instance, some interviewees say the system is not sensitive enough. They said they inclined the circuit playground during the test, but the system didn't trigger the warning, so they tried to incline more, and the warning can trigger successfully. After that, they point out the system works, but it is not sensitive enough. In fact, I can also observe this problem during the testing sessions. Testers tried to reduce the angle gradually until the warning triggered. From this observation, it proves the testers feeling is totally correct.
During the testing sessions, interviewee also reflected other performance issues as well. They mentioned false detection happen occasionally. Although it is not very frequent, but they said the false alarm can be annoying, which will affect their experience when using the application.

Beyond performance issues, interviewees also give some feedbacks on user experience. One of the interviewees said the warning system feedback is too aggressive. When the warning system was triggered incorrectly, the aggressive feedback makes him more impatient to the system. Indeed, I also observed one of the interviewees looks confused and shake the circuit playground when the warning system trigger incorrectly.

Lastly, most interviewees also talked about their expectations. During the interview session, majority of the interviewees agree using automated approach to collect information for application input is a good approach, as it can make the application more user friendly. At the same time, interviewees also point out accuracy is something they very concerned. Some of them mentioned if a system doesn’t work well, make it convenient will not add values to it.

Conclusion:

After conducting the testing sessions with several people, the performance of my prototype is better than my expectations. But I also realize there are many spaces for improvements. Through my observations and the interview sessions with the testers, I understand there are performance issues with the design, and it needs to be improved, for instance, sensitivity issue and false detection errors. At the same time, I also have more understanding on the user experience and what potential users expect the application to do, which are some useful insights to improve the application. For instance, the impact of strong feedback in an incomplete system is fatal, so I need to utilize the system or redesign the form of feedback to reduce the impact on users.

Also, I can conclude the hypothesis I made in “the purpose of this testing round” is correct, since the evaluation process tells me automated data collection system doesn’t guarantee success, there are many factors that can affect the result, so developers need to solve every issue before the system getting to a complete state. This also shows there are tradeoffs on automation, and each automated system has its unique way to solve a problem, it won’t be the same. The research questions in prototype 2 is definitely a good choice, it helps me understand the drawbacks on the head movement detection system so I can solve the performance issue before it releases to the public.

Changes I will implement to my concept after getting feedbacks from the testing session:

Although there are some drawbacks on the automated data collection feature in my concept, but most interviewees agree it will bring them benefits while using the application in the future, so I won’t completely remove the automated data collection feature for driver head’s movement. Instead, I will try to utilize the system and balance the pros and cons on the automated data collection system. In example, the build-in accelerometer on the circuit playground has X, Y and Z values. These values are very sensitive, even a very small move can change them substantially. To improve the sensitivity and false detection issue, I can try tuning these values, and do some

testing with different people. I can pick the set of X, Y, and Z values that fit most tester's expectation, which may solve the problem we encountered before.
In addition, the head movement detection system doesn't need to restrict to one approach, it can use multiple approaches at the same time, so that it can use various source of data to make judgements, which will reduce the chance of generating the wrong output. In future development of this prototype, I can add driver's confirm system to the data collection system. This system will only trigger when the system is in an ambiguous state, for instance, the system doesn't feel confident to trigger the system or there is insufficient evidence to enable the warning system.
By tuning the parameters values on the accelerometer and adding an authentication system to my concept, I believe it will make the automated data collection system more reliable in my concept.

Whether the prototype is sufficient to test what I want to test:
Prototype 2 is sufficient to test what I want to test in prototype 2, since it gives me useful insight on the performance of the automated data collection system, and it tells me what people expect from my concept. This information points out the current problems with my concept, so I can decide whether it is a good approach to collect head movements data as input or not. In addition, the information I get from this prototype gives me multiple directions to improve my concept, so it is sufficient to test what I want to test.

Testing Plan

Protype 2 follows the plan in prototype one's statement of delivery, and I didn't make any changes to that, it ends up quite well.

Prototype 3: In prototype 3, I will not follow what I proposed in the statement of delivery of prototype one, since I think exploring the effectiveness between single feedback system and multiple feedback system shares many similarities with my research aspect in prototype one (whether single level feedback system better or multiple level feedback better). Also, this research question is quite simple, which doesn't have too many spaces for exploration, so I decided to switch.
In prototype 3, the research aspect will focus on the feedback when high mode is active. Base on the original design of the concept, when high mode is enabled, it will trigger sound feedback and vibration feedback. Based on the previous experience in prototype 1 and prototype 2, I realize having a strong or aggressive responses may not applicable all the time. When you use sound feedback and vibration feedback nicely, it can give you good effect. If you use it badly, it will bring negative impact to the application. Prototype 3 will try to find out what is the best combination of sound frequency and vibration frequency, so that it can convey a good warning

signal to drivers but won't ruin the user experience. The research question for prototype 3 is "Is having a high sound frequency and high vibration frequency always give us the best effect for driver who fall asleep, or other combinations will give us a better outcome?". This is the research question I will figure out in prototype 3. By the end of prototype 3, I should understand which combinations leads to better outcome, so that I don't need to struggle between the strength of the feedback and the quality of user experience.

Miro Link

https://miro.com/app/board/uXjVPRHuwsY=/?share_link_id=197936182612

Video Link

https://youtu.be/JuuXvboqHn8