Automotive Driver Assistant Report

Gabriel Bulai
Enrique Cordero
Kristiyan Dimitrov
Hampus Gunnrup
Nicholas Oloo

Introduction

Background

With people driving everyday, to and from work, during work, to the store, road safety has become an increasing priority around the globe. One of the many issues is the distraction level amongst drivers, especially when it comes to people driving for a living, such as truck drivers. This creates the need for a source of motivation, during a drive. Something that will keep the driver focused and motivated to drive correctly.

Purpose

Our goal is to reduce the distraction of these drivers, by making a pleasant driving experience. The main constraint on the application, is that any action during a drive, has to be made in under two seconds. This puts a lot of emphasis on the interface, and makes planning important for visual design.

Discussion

Driver Safety

The applications' main focus was to provide the driver with assistance whilst keeping all safety regulations while driving. The task ,certainly, was not simple and led to many dynamic changes, in the end it was successful in achieving its goal.

To achieve safety while driving the design and timing were changed numerous times until a satisfactory result was reached. Many a time we came up with ideas but they did not meet our expectations, such as using "smilies" or bar charts to display drastic changes during the drive. Another interesting idea given to us as an advice was filling boxes on the screen but it was disapproved by the developer team and as such was left out. Yet this idea led us to the successful design. Four players which layers with a respective icon within them, each layer changes the colour depending on the score informing the driver of his progress without much effort or changes of focus from the road to the phone. Sadly only that was not enough, thus we integrated an alert system which flashes the screen red with an icon of the measurement in which the driver did a drastic dangerous action. This alert flashes only for a brief moment and gives off a sound signal to fully inform the driver of his violation. If the driver is in no position to glance at the phone he will still be aware that he executed an action which may lead to a dangerous outcome (for example over-speeding) or inefficient driving (excessive fuel consumption). During the driving session he is unable to access the other functions of the application unless he pauses and stops the drive.

The remaining functions of the application are aimed towards stimulation, stimulation to both improve the driver's performance and do so in a safer way. A powerful stimulant is competition, having a ranking system with all registered users the user of the application would naturally want to be better than his friends or the other drivers.To achieve that he has to drive safely and efficiently which subsequently leads to more responsible drivers. Another stimulant is the medals. It taps into the user's sense of achievement and rewards appropriately for his high scores, which require safe driving.

Scrum

The team group, having got experience from the Software Processes course, used Agile and scrum methodologies to divide and organize tasks. Iteratively, the application gradually took shape by integrating new code systematically. Due to the project dimension and lack of expertise, beside additional individual assignments, courses and personal concerns the members decided to have a variation of scrum meetings, held once or multiple times per week, specifically agreed and depending on the workload. The project has been divided in six sprints, each falling in the range of approximately two weeks, and two demo releases which contributed to a better preparation and competition, raising the group motivation.

Each sprint had its own requirements freely proposed, discussed and divided among the group members. In order to create a visual and methodical work process, ''Trello'' collaboration tool was suggested, which led into a better organization structure and effectiveness of the tasks. However, as the project duties went increasing, its use moderately deteriorated, giving more and more space to oral communication, direct contact and group work.

Teamwork

Working in a team consisting of people from different cultures, nationalities and personalities leads to inevitable difficulties , minor or major depending on the experience in handling such matters. At the beginning everything seemed well except a few minor issues when it came to opinion and behavior. Such problems were handled with haste and were resolved relatively fast.

Sadly there were several major issues which were building up but were not brought up with the right emphasis. Having not been resolved on time they became an obstacle which in the end led to an ultimatum from the faculty in charge of the project. Drastic as it may be , it indisputably improved the situation and pushed the team into overcoming the obstacle except for the fact that one member dropped out due to insufficient interest in the program. Afterwards the team really got together , despite a few minor disagreements caused by difference in personality. The work done during a single sprint increased noticeably and all of us noticed the mistakes which we made. One of the biggest mistakes which we made was that we had the wrong definition of the word “done”. We had many components of the application which were not complete, we believed that we had completed the most difficult part of each and that only the simple matter of integrating and connecting was left. Once the time came to assemble a working prototype we had to relocate more time and put more effort than we had anticipated. We took notice of that mistake relatively late but on time and changed our mindset to prevent it from occurring in the future to ensure a successful project.

If we could do something differently then we would not have kept our emotions regarding the group and work done to ourselves. We would discuss every issue openly, what we deem appropriate and what inappropriate, as we did in the middle to the end of the project. Aside from that we would complete every feature fully and then deem it as “done” to prevent a possible failure in assembly of the project before the deadline.

Git - Version Control

The group has continuously used the Git Version Control to keep the project files updated and easily shared among the members. Starting with less than basic experience with the program, the group faced an opening that went gradually increasing in experience and performance individually with several attempts. There has been a transition from using git command line at the early start, due to the Git introductory lesson before the project

The use of different branches created by each single phase helped developers to avoid multiple updates into the origin main branch, making the readability and debugging easier and more accessible for potential stakeholders. The transition from command line and Android Studio built in Git service has proved to be useful and less time consuming in the management of the different branches especially when comparing and merging with 'master,

Despite all the encountered problems, Git tool has been appreciated by team members and its usage has been fundamental in the project updates.

Communication

There were four communication methods which we used. The first and most impactful method was face-to-face communication on scrum meetings set by us. It served for evaluation of our current progress, voting opinions, determining our future plans and resolving conflicts of which all of us were aware of.

The second method was using facebook. Which was the main tool for communication amongst team members. If used responsibly, Facebook is a great way of communicating with the team. Indeed, it was easily accessible by every member of the group and it was a good reminder due to the pinning post function which is provided but it had many flaws. The main flaw with Facebook, is that it can be a source of distraction for team members. If at any point a team member is intrigued by something irrelevant to the project, and since there is a lot of content on Facebook, this can greatly distract the members of the team. The problem is that Facebook is often used on a daily basis by people, as a social playground. This is a hard habit to break, and is a hard problem to solve.

The best way would be to not use Facebook, or to at least open separate accounts that only shows content relevant to the project.

Even with separate accounts Facebook can still post some problems. For example, when a group discussion appears and goes on for even just a couple of minutes, it can be hard for some members to follow the conversations. This can result in important information not being passed to every member of the team.

The third method was skype. Skype was for times in which we did not have the possibility to meet and determined it was important and needed to be discussed orally or when one member was in dire need of assistance from another member. The share screen function definitely proved to be useful in such cases.

The fourth and last method which was used is google drive. It was not approved by all members but a consensus was reached that it can be used by whoever wishes to do so. It's primary use was for documentation, allowing multiple members to write at the same time while being able to view instantly the changes which the other person makes. By doing so it prevents repetition. The downside of google drive was that to be fully effective you need the members to be close to each other to discuss while writing to maximize efficiency, by not doing so it once had a negative effect and thus ended up being disapproved as said in the beginning.

If we could do something differently we would definitely have avoided facebook and used a different, more private software for communication which is , just like facebook, accessible through the smartphone and allows file sharing.

Leader role

The different roles each person had in the project, made it difficult when it came to decisions. In the beginning of the project, the team relied on the scrum master for important decisions. Since the scrum master don't have that type if authority, this usually ended up in poor communication. Eventually each person in the team started seeing themselves as both a product owner and a developer. This improved the communication, and the decision handling.

It did, however, post a new problem when it came to decisions. There would often be an extremely unstructured meeting, or part of a meeting, revolving around a specific issue. Each member of the team, would say their opinions on the matter, and no real decision would be made. This lead to unclear goals, and a big amount of time being spent on change of requirements.

The main cause of this issue, is the lack of a proper leader role. Since each member of the team is a product owner, each member had equal authority. This meant that the leader role was divided equally, and decisions were hard to make. A solution to this problem, could have been to decide upon a structure or system, that would define a way of making decisions. In that way, each member could have their ownership in the project, with decisions still being consistent and quick.

The evolution of the GUI

Throughout the project, the visual design has been consistently changing. From the beginning, there was no structure or plan made for the GUI, which led to a confusing interface that distracted the driver.

After the alpha presentation, the interface got more attention from the team. The interface was, however, still not appealing from a user point of view, and the main view seen while driving, distracted the driver.

During the beta-presentation, a lot of criticism were given towards the view distracting the driver, and the team started sketching a design through gimp. After implementing that design into the main driving view, the problem with distracting the driver was solved. The rest of the app, however, was still confusing and did not have the same look and feel as the driving view.

Using the sketched design and the driving view as an inspirational source, the rest of the app was evolved to be more clear and visually fulfilling. The end result is an app that is easy to understand, and does not make it hard for the driver using the app to understand it.

Architecture

Since android uses a structure that separates the concerns of the application, this approach was used throughout the project. The design was frequently modeled and separated into different layers, where each layer has its own level of relation with the application. The highest layer, the UI layer, contains classes that closely communicates with the user interface and is highly application specific. One layer below, in the application layer, the classes gets more independent and can be reused with a different interface. They are still, however, very application specific. In the lowest layers, model and foundation, the classes are even more application independent and can be reused to a large extent. These classes are mostly custom data structures used in the rest of the application.

This layered approach supports the design principles high cohesion and low coupling. To make the application support high cohesion and low coupling even further, the team implemented a controller that communicates between the different layers and classes. This makes changes in the code easier to handle, and thus easier to maintain.

Dependencies

We used several dependencies in our application, which include different libraries provided by AGA and other external libraries from the web.

The following are the four libraries provided by AGA:

1. Automotive API
2. SDP
3. VIL
4. Automotive Service

These libraries were involved in the connection between the application and the simulator.

Other libraries included *mpandroidchartlibrary*, *achartengine*, and *androidplot-core*;
All of these were used for graphs and charts, but in the end we decided to use only one-
mapandroidchartlibrary in order to increase cohesion in the application since it was the most capable library- able to do all the visual tasks - displaying all the different charts and graphs.

Alpha

The alpha version was not very concrete, we had our idea set but as we still had a difficult time
deciding on how to present the information to the driver without distracting them at all.
We had a main view with smiley faces that changed periodically depending on the driver's decision and all the other aspects of the application such as grading system were still very experimental since this was a very early stage of development.

Beta

The beta version was more substantial as we added an alert system, the grading system was much more complex and the main view had a live bar chart that showed the driver's progress in real time. At this stage the application was functioning properly but we still had a problem in deciding what to present to the driver during a driving session. The bar charts were too descriptive and so the driver would have to do a lot of straining in order to gather what was being presented on the screen. We still hard some work to do for the final version in deciding how we should make the main view simple and easily understandable at one glance thus decreasing driver distraction**.**

Final

The final version satisfied all the stakeholders needs. A ranking system was added to the application for social media purposes. The initial plan was to utilize facebook for these purpose but that turned out to be a daunting task due to the limitation of time. We also changed the mainview in order to best satisfy the drivers needs. We decided to use symbols with colours following a traffic light theme of green amber/yellow and red that changed depending on the driver's performance- these ensured that the driver was able to look at the screen and get the presented information as first as possible with no major strain. The graphs and charts were still included in the application but these were now in a different view where the driver could access them for a more detailed assessment when they have completed a driving session. Overall the team was satisfied with the end product as it met most of our requirements; we hoped at this stage facebook would be utilized but that was a secondary requirement.

Database solution

Given the previous experience we had in Java and databases from last semester our group did not focus too much on data storage at the beginning of the project. Our main focus would be the learning curve of learning how Android Studio and Android programming worked. It took some time to get the basic functionality of the project working. This however presented a huge challenge concerning the database because we hit a wall. The group tried to implement an adapted version of the DBHandler from last semester. From the first moment this was a problem and pretty quickly it was noticeable how it would not be possible to do. According to a lot of information in StackOverflow and the Android documentation, the type of connection we applied last semester was considered unsafe and therefore Android would not allow it, which would cause the app to crash. Given that a lot of the functionality of the app relied on the group having a database a solution would have to be implemented and would have to be done fast since a lot of the basic functionality was already up and running and required real data to start testing.

In one sprint it was it decided that a remote host that offered a web service was the way to go. Few to none of the group members had experience in PHP and none of the group members had started to work with Async tasks and Androids way of doing remote connections. However basic functionality such as Login and Register functions came along with the learning curve and the basic functionality of set and get measurements was able to be applied during one sprint.

The database challenge gave us unique experience in real backend applications since it created a web service that could easily be used by a website or even another application. From this point of view it was an enriching experience. It also facilitated that one person took over the back end and therefore the whole architecture of the database is very consistent and so little to no loss of data was experienced.

If we could have done something differently it would have been to fully check the requirements of the language and environment at hand instead of making assumptions. We thought that since it was java then it would work the same. The fact of the matter is that the Android SDK has its own requirements and limitations and should have been checked before hand. Another change would have been to further evaluate the necessity of the database and how it could have been applied. Checking with other groups in the final presentation it was obvious that several other database options existed including the local SQLITE database used by Android and stored locally on the telephone.

Programming language

When starting the semester most people knew Android was going to be our programming language for the project. It was said that it was basically Java with a little bit of XML. Our lack of experience on different SDKs and our comfort in believing it would be easy gave us an incredibly big challenge to tackle early into the project. Java is the main language in Android programming. However it uses the Android SDK to process the code. This brought a whole new set of limitations and requirements along with added functionality. Not only was the Java part going to present a challenge, but it seemed also that XML was not as easy as we thought and it played a bigger role than expected in the application. Clearly the learning curve was highly misjudged and the team did not start early enough to tackle the situation.

The functionality enhancement brought by Android was an incredibly enriching experience. Android seems to take java to a new level. It allows us to do Async tasks, which basically creates a new thread for that specific action and it handles it in a very elegant way. This provides the group members to develop a deeper understanding of limitations and possibilities that all programming languages face. Also the challenge with XML and the deep relationship it has with its given class obliged people to focus on both design and on functionality. This brings a unique opportunity for people to strengthen the areas where they might be a little weaker.

If we could do something different it would have been to try the language as early as possible with different dummy projects. This would have allowed us to experience some of the challenges early on, and possibly the team could have avoided such a steep learning curve that had to be faced during an already tense and time consuming moment of the semester.

AGA

The AGA simulator was a very interesting and exciting tool to use. It gave us as students a real life example of what type of tools could be programmed to sell and to enhance the way we as a society study different ways to live better (driver safety in this case).

The tool has an enormous amount of challenges to use. Because it is still under constant development, different documentation exists for different versions. The website that was given as a suggestion to follow was based on examples that were outdated and incomplete. Also the documentation available with the libraries was very difficult to follow and was very “over the top” documentation. It did not go deep enough in a lot of explanations and it made it difficult to understand exactly what a method did and how to use the specific method.

In our group we had three different operating systems using the simulator. This gave the group the possibility to really test the simulator thoroughly in all the environments. First of all there is no simulator for MAC. This presented a challenge of having to use a simulator aimed for Linux on a Mac computer. Also, AGA has a big difference between its Windows simulator and its Linux simulator. Through the documentation it is stated that different signals update or send a signal with different time intervals. Once the simulator is open these time intervals are visible. For example driver distraction sends a signal every 1000 milliseconds but wheel speed sends a signal every 100 milliseconds. When testing the simulator the reactions were slightly different. For the Linux simulator on a Mac computer all the signals sent the signals as stated except for the driver distraction signal. This was not a big surprise since it is declared on a separate side when opening the listener. However the biggest difference with the windows simulator comes when expecting the signals as the documentation and the simulator state. The windows simulator will only send the signal when the value is changed and then in the time interval stated. The signal is therefore not constantly received by the application. This created a huge problem for the group since the experience testing the app ended up having different results depending on the machine being used.

Something that we would do different given the possibility would be to try to read more thoroughly through the AGA documentation before encountering big problems. Also a robust dummy setup to play with the simulator would have helped to maybe encounter the problems and differences with anticipation. Another variation would be to try to post on the AGA website about our findings and our problems with enough time to see if the owners of the simulator could come with a better solution than to patch around the app instead of actually using the right tools for a platform versatile app.

Conclusion

As the application started to take shape, personal remote working faded, giving space to regular frequent meetings and pair programming. The team appreciated the explanations of design and logic relations among all the components of what would have become the final application. As a lot of new features were part of requirements and therefore implemented, pair programming has been used in most of the cases to brake procedures into smaller parts, increasing efficiency.

For our group, the task evaluation was very difficult therefore consequences have been sometimes unexpected. Although there have been a few problems in the communication area that led into slower delivery times of some parts, but fortunately the determination helped to overcome pressure and personal considerations and concerns keeping focusing on the final objectives and on the groupwork.

In this way, a clear set of rules collectively agreed on was necessary and productive

The absence of a social contract as well of a project plan had a great impact on the management of the group main decisions, evaluations and responsibilities. In similar occasions and for future group projects it will be more effective to use these documents in order to maintain a certain order and to have a source to examine and to rely on if project tasks acquire a bad propensity, trying to adjust flawlessness and flexibility.