

**CO600**

**Ethics Generator**

Technical Report

Kieran D’Arcy, James Scarry, Taylor McNicol, Ibraheem Jhanji

Supervised by R.O. Kafali

**Abstract**

This is a comprehensive report for the CO600 project ‘Ethics Generator’; the report will feature an overview of the entire development process of the project (named Autonomous Ethics by the four developers), as well as discussing the technical and non-technical features that went into its creation. Throughout this report, there will be an outline of our targets/goals that we aimed to achieve with the development of this program.

The report will also feature our reflection of each step of the development and design. Autonomous Ethics was designed and developed by Kieran D’Arcy, James Scarry, Taylor McNicol, and Ibraheem Jhanji.

1. **Introduction**

Autonomous vehicles and the industry they are part of have been facing the issue of moral dilemmas/clashes for many years now. While the technology for these vehicles continue to improve and develop, there is one thing that has kept industry experts at a standstill – that is the issue of deciding what a vehicle should do in the event of an unavoidable accident.

Many have expressed discomfort behind the idea of vehicles making these ethical decisions; morals and ethics are hard to centralize based on location, and it is unlikely that the perfect medium for it is even possible to achieve.

Autonomous Ethics aims to test the morals and ethics in a user’s decision-making. It offers the simulation to the user of being a passenger in a fully automated car; the car is approaching a crossing and cannot avoid an accident. Autonomous Ethics randomly generates various scenarios to present to the user, offering them a selection of three options. The program will measure their responses and collect information such as which age group they favour most, and which character(s) they choose to sacrifice more often.

Autonomous Ethics was built with the intention of being a research tool, offering 28 unique characters for use in the generated scenarios.

1. **Background**

**2.1 The Problem we are Addressing**

As explained in section one, our project was designed for the purpose of being a tool to conduct research in the area of vehicular autonomy and the moral complications that come as a part of that. Autonomous Ethics seeks to offer concise and realistic data in its results.

**2.2 Deciding the Setting**

During our planning stage, one of the key concepts we discussed was how the program was going to work at its core – mainly based off the setting and the functionality of our potential choices. We were able to narrow down to a few different areas of autonomy.

We discussed the possibility of autonomous trains, planes, boats, buses, and cars. Although we discussed the idea of developing the program to cater to more than one area, it was a unanimous decision to ensure quality over quantity. We decided to start on just cars; however later into the project, we began discussing how it could potentially be extended to explore different areas.

**2.3 Creating Uniqueness**

One thing that was important to us while we planned the development of Autonomous Ethics was ensuring its originality and uniqueness as a research tool. We discussed implementing this in a few ways; most notably, the timer feature was one of the first things put forward to make Autonomous Ethics stand out. While the users are taken through their scenarios, they will alternate back and forth between timed interactions and limitless ones. This feature is intended to add a sense of realism to what the program is demonstrating – in a real-life situation, an autonomous vehicle would not have an unlimited amount of time to decide the course of action to take.

Using the timer feature, we can measure differences in user decisions compared to when they had time to think about it and when they did not.

Another feature discussed was the concept of having scenarios given to the user in both first person and third-person perspective.

# **3. Aims**

The key aim of our group project was to provide a greater insight on how the choices we make throughout our daily lives, each show varying ethical and moral thought patterns.

When first discussing our project, we concluded that determining someone’s exact moral code would be next to impossible, given the fact that morals, ethics, and legalities vary greatly across the world. This is something that carries over directly to the autonomous market; our primary aim was to develop something that would allow for a clearer view on an individual person’s moral/ethical decision-making.

To gain as accurate of a result as possible, we wanted Autonomous Ethics to provide a vast range of scenarios, which would put the user at the front of the conflict - they would be the ones to make the decisions that they saw as appropriate. Each scenario would present a selection of varying characters; in any given case, the user must choose from three separate options, resulting in varying casualties.

To avoid extreme bias and controversy, ethnic background is not a factor taken into account in these scenarios.

**3.1 User Requirements**

Among our user requirements, we determined that a user-friendly interface with a high ease-of-use was of great importance. We wanted our users to be able to focus purely on the scenarios presented to them, rather than having to take a great deal of time to learn how the program itself functions.

Instead of being text-heavy, we wanted our project to make use of the imagination, to provide the user with a straightforward explanation of what was happening in any given scenario. As some scenarios would impose a time limit, having our users be able to take to the program quickly and without any confusion was one of the most critical aspects of the development.

We believed that the number of scenarios should be limited to a lower number to keep participants engaged, yet large enough so we could gather a pool of adequate results from each user.

User privacy was also of importance to us; we utilized a database for storing the results from each user. User accounts do not require full names as we wished to maintain a sense of anonymity.

## 3.2 Possible Extra Features

While in the development process, we discussed potential new features both amongst ourselves and with our project supervisor. The features that we ended up adding to our program were some of the following:

* A timer which would alternate between scenarios; this would enforce realism to the scenarios as it would put the user on the spot to make a decision, otherwise Option 1 would be chosen by default. This ties directly to a real-world scenario, as an autonomous car and/or human driver would not have an infinite amount of time for decision-making in the event of an accident.
* A simple scenario/progress counter to keep track of how many scenarios had been completed. This benefits both the data collection and the users themselves.

There were other features discussed that could be potentially added in the future, however these were deemed non-essential and were only to be developed once the foundation of Autonomous Ethics was in a good place. Primarily, this included:

* Different types of vehicles/automation (planes, boats, trains, buses, etc.)
* Injury/death scenarios; some options may result in severe injury/disability as opposed to the death of a character, adding in another layer of moral conflict.

## 3.3 Non-Technical Requirements

Under our non-technical requirements, we concluded that finding results that were as unbiased as possible was of high importance, as mentioned previously. We took steps in our development to ensure that the varying factors between characters did not allow room for racial bias. This is why ethnicity is not a variable in Autonomous Ethics.

As said in ‘User Requirements’, allowing our users to maintain a sense of anonymity is important, given the theme and function of Autonomous Ethics. Full names, DOBs, addresses, and other sensitive information is not required to participate/use our program.

# **4. Development, Planning and Decisions**

## 4.1 Scope

The initial topic of autonomous cars deals with a rather large outlook of different perspectives into the many different ethical theories, which arise over the discussion of autonomous cars. Ethics are a deep aspect of everyday life, yet many people have probably never taken a second thought into them, and most likely only separate things into the two categories of good and bad.

Our project offers an opportunity to explore morals and ethics on a deeper level. Autonomous Ethics provides an explanation of ethical theories to the user; even more so, using the choices made by the user, Autonomous Ethics will provide the user with the knowledge of what ethical theory their decisions are classed as.

Autonomous Ethics splits ethical classes into three subsections: Utilitarianism, Virtue, and Deontology.

## 4.2 Website or Program

Once we’d figured out exactly what our project would entail, we then needed to make a decision on how it would be presented. We initially decided we would create an application for computers, but realised it would be going against one of our technical requirements (access) as we believed that in the future this would then be available for mobile use, therefore allowing access to a larger group of participants. As such, we created a program which was hosted on a relative website - this allowed for quicker and easier access, as well as a method of data storage.

**4.3 Development Resources**

To produce the site we used Visual Studio code in collaboration with Git, in order for all members of the group to keep track of the source code and allow ease of use when coordinating work amongst ourselves. During development, the project was split into two branches which allowed multiple people to work on it at once without overwriting any previous changes (of which could have potentially caused damage to the integrity of the project).

To maintain order and management of our project and all of its resources, we made use of a free project management tool (Basecamp) which enabled the whole team to stay connected, as well as enabling us to keep track of all relevant documentation and project files. Basecamp also allowed us to schedule reminders for both group and supervisor meetings; Basecamp was also efficient in producing a changelog for documents/files that underwent constant updates.

## 4.4 Languages

For our development process, we made use of multiple languages for the generator, the holding website, and Autonomous Ethics itself.

We made use of HTML to develop the core of the website and its functionality. The website was used to display and run Autonomous Ethics for the users; the results page can also be accessed from this website.

CSS was used in conjunction with the HTML and jQuery elements; this was used to style the website and for graphic implementation. On the same note, JavaScript was used majorly in this area to build the main part of Autonomous Ethics; we used JavaScript not only for the main program but also for the scenario generator.

A server with Node.js was used to establish the database in which the collected results and users are stored; jQuery was utilized to establish a connection between the database and the project website.

**4.5 Development Approach**

For our development, we approached it using the Scrum Development Model. After establishing exactly what it was we needed/wanted for our project, the team collaborated and assigned roles based on strengths and weaknesses. It was determined that development of the main program and the generator could begin without having all of the graphics ready; placeholders could be put in place until this was achieved.

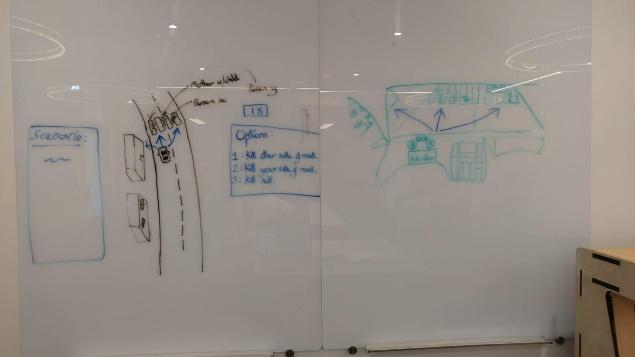
While some members of the team worked on backend development, others worked on the front-end and the graphics design. This allowed for progress to be made in several different areas at once, ultimately combining at the end to deliver the finished product.

This was the most time-efficient method of development as it prevented members of the team from having to wait for something else to be finished before they could continue; it required stricter monitoring and greater accuracy, as well as some additional patching once all features were merged, but as of the final version everything was working as expected.

# **5. Design**

## 5.1 Concept Designs

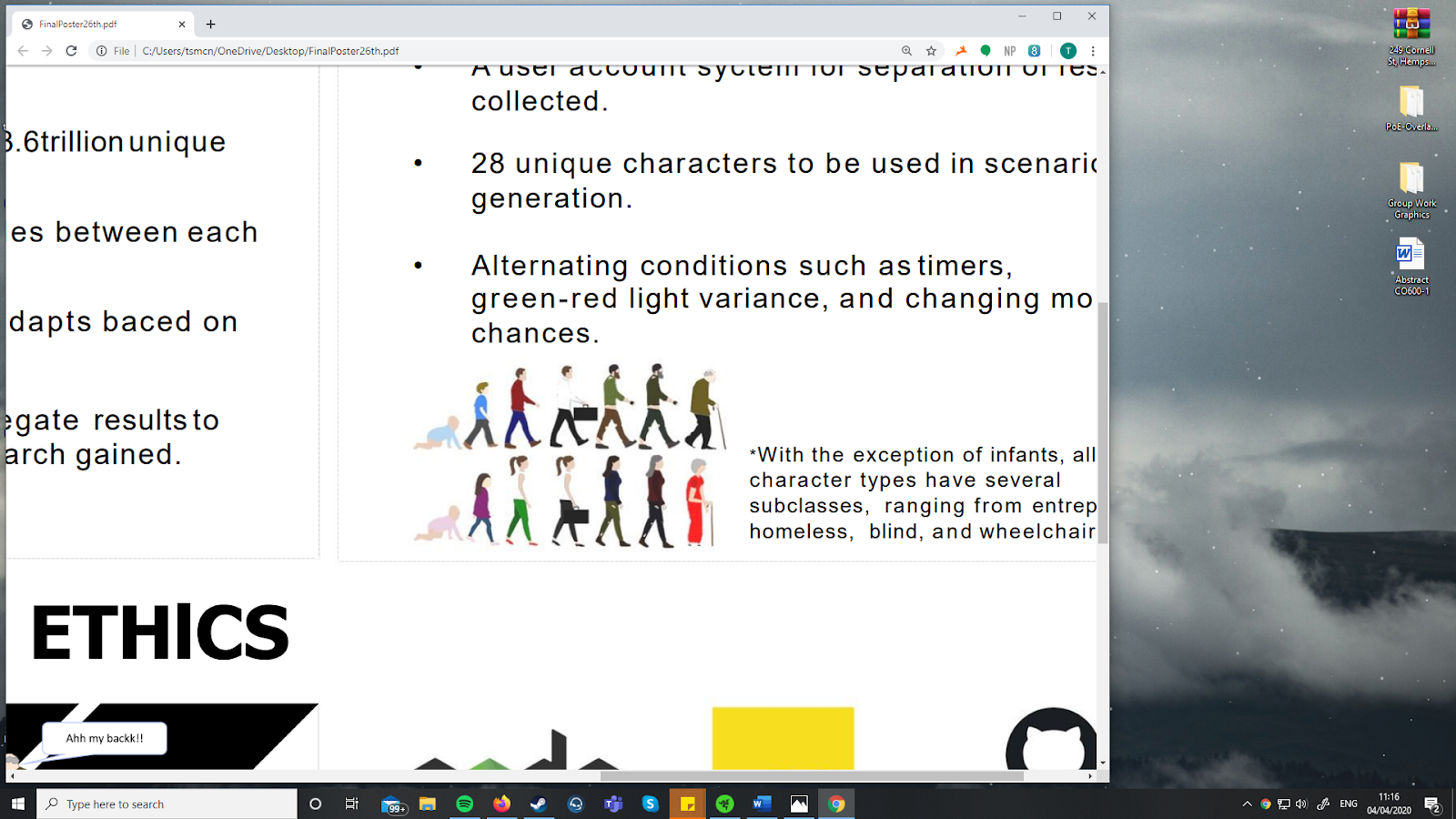
At the beginning of our project, we first had to collaborate and create a rough sketch-up of what we thought the final product should look like.



The above image is our group’s collaborative sketch-up. Everyone was instructed to bring their concept drawing; the above is the finished result of all our ideas pooled together. The imagery seen on the right is supposed to portray our program from a first-person view. It was through our collaborative concept work that we were able to decide which key features we wanted to bring forward, as well as what features could help us stand out.

## 5.2 Graphics Design

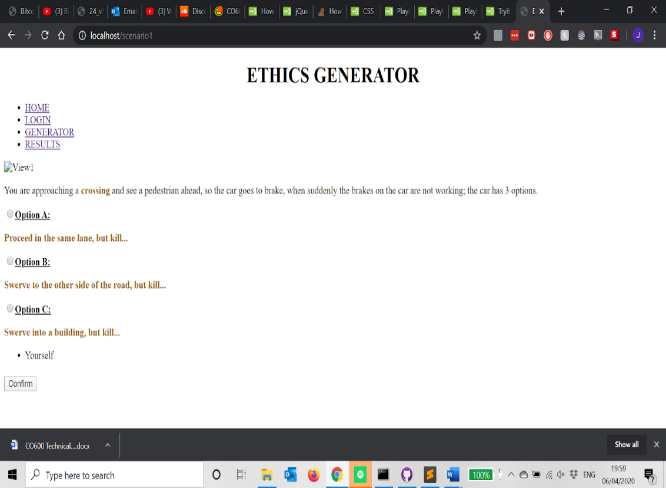
Once development for Autonomous Ethics had begun, there soon came a long list of required graphics that were needed to help piece the functionality and intended purpose together. 28 unique characters were used in Autonomous Ethics, all of them from varying age groups, ability, and class.



The image seen above is a collection of seven characters from each gender; each individual character seen has a unique difference to the others (age, class, profession, etc.) All graphics are not displayed here, however different variations of the unique characters exist, such as being wheelchair bound and/or having the ‘homeless’ modifier.

Alongside the characters that were individually designed, there were separate graphics for the vehicle, the red and green lights, etc. The environment in which our project takes place was built using Blender.

## 5.2 Web Design

This was our first web design layout that structured the basis of our generator page. From here the pre-set options were placed for 2 scenarios, so we had an understanding of how we wanted to proceed. Therefore, it was fundamental to have a clear view of having the navigation bar along the top, scenarios on the left hand side

of the page and eventually have the corresponding images on the right. It was clear to us that this format would help identify what options the user has and to be able to make the clear connection from the options to visual characters on screen.

Another concept we wanted to implement was the use of hover over effects when picking an option so you can identify what side of road each option is. This would allow the user to understand the decisions the car would have to make in the self-driving age, showing the true moral decisions, compared to human reactions.

# **6. Implementation**

## 6.1 HTML Structure

Once planning had been completed, it was time to move onto piecing each part of the project together. This was to start from forming a basic html page that had present, hard coded scenarios, so we could visualize our end goal. When the structure was agreed upon we added in the scenario generator instead of having hard coded options. This would initially allow the generator to work on basic functionality so we could optain an understanding of further areas of development. Use of CSS and JavaScript styled the page in terms of positioning (overlaying images) and having specific formatted navigation bars.

## 6.2 Scenario Generator

The scenario generator is made up of multiple functions in the Index.js file that when put together can generate a new scenario for a client. The scenario generator uses the information stored in the MySQL database to make these scenarios with the exception of some probability logic that decides on other factors of the scenario.

Functions include:

* The “/scene” getter listener function - this function listens out for a get request from the web page to get the scene.
* makeScene() – this function is a compact function made up of all the other functions and some separate logic required to piece the scenario together before returning the complete and correctly formatted scenario ready for use by the web page. This function also determines whether there will be a time with a chance of 50%, it also controls the limit of questions there are before redirecting you the results page
* getPeople() – this function returns a specified number of random people to be used in the scenario.
* getCrossing() – this function randomly returns 1 of the 3 crossing types to be used in the scenario.
* query() – this function is used to query the SQL database.

## 6.3 Image Creation

Image creation for Autonomous Ethics was done using Photoshop CS6. There are 28 unique characters to be used in scenarios, as well as graphics for traffic lights (red/green). With the exception of infants, all characters had several variations: Homeless, entrepreneur, elderly, homeless/elderly, and wheelchair-bound. These images were created with the intention of providing a unique perspective on the potential moral conflicts behind vehicular autonomy.

All of these images were individually mapped to scenarios; each one has its respective properties to enable proper data collection, of which is also used to display the post-participation results for each user.

## 6.4 Mapping Images to Scenarios

The hardest part once the images had all been made for each character (26) and scenarios generating randomly after each selection was being able to map the images that corresponded to the randomly generated character. Through creating a ‘scene’ using jQuery on the generator page that links to the main index.js and external database, we could append selected characters for each scenario with the ID referring to their image. However, to use this in CSS we had to style each character separately as we did not know which character would appear in each scenario. This took a reasonable amount of time due to aligning each character perfectly on their respective side of the road. Having an aspect of visualisation in our project was important so the user could feel the real-life implications it would have making each decision.

**6.5 Index and Database**

The index.js file, made using Node.js, controls everything that happens on the back end of the website including running the web server itself. One of its main tasks includes responding to requests made by clients to view specific webpages or to access specific files and resources such as images, stylesheets or acting as middleware when clients make requests for information that is stored on the SQL database. This saves the trouble of using a separate program to communicate with the database.

The database is a MySQL database that has used to store information used to create the scenarios, e.g. the crossing types, people and all the results from users answering questions.

## 6.6 Results and User ID’s

Creating the results page giving details on the character they killed most, the option they picked the most and how long it took them to answer each question, needed unique ID’s for each user. This would redirect them to a new page, which has a form and saves their unique ID to the database, and then sends them to the generator page.

Control methods for this page was having a new ‘userName’ every time the scenarios were completed, and results generated. Also the ‘userName’ has to be unique and not used twice as this will throw up an error, then a confirmation alert is at the top of the page to make sure you are over 18. This is due to the nature of the ethics generator being a controversial moral dilemma and having challenging options to consider.

Once a user has completed the five scenarios they are redirected to the results page, which uses nested ‘if’ statements to calculate categories they picked the most or least. From this, an ethical report is generated that predominantly identifies either a utilitarian or virtue approach.

## 6.7 Counter and Timer

The counter counts the number of questions answered so the user knows what question they are on, once the user answers 5 questions they are automatically redirected to the results page where they are shown stats for their answers.

The timer counts the time you have left to answer the question, which is 20 seconds, when the timer reaches 20 seconds the page automatically selects option A and then submits that answer and moves you onto the next scenario. The timer only appears on scenarios that have a timer and the probability of a scenario having a timer is 50%.

# **7. Evaluation**

## 7.1 Conclusion

Autonomous Ethics was designed to be a research tool to be used in the area of autonomy; namely, autonomous vehicles. While we initially wanted to develop something that was multi-directional in terms of what variations of autonomy were available, due to time constraints we had to settle on one. We chose to build something focusing around autonomous cars (with the option to expand later). Trying to build something with a unique spin while maintaining potential data soundness proved to be a challenge; the planning stage went on longer than we would have liked, but ultimately resulted in a final product that was - although more refined than we first wanted - clean, easy to use, and effective for its purpose.

Rooted in ethical theory, Autonomous Ethics did achieve its intended purpose. It presents the user with a unique but realistic situation involving a fully autonomous car, and several pedestrians crossing the street. The user is placed in charge of what happens in an unavoidable accident; we wanted Autonomous Ethics to be able to tell users what ethical background their decisions fell within (utilitarian, deontologist, etc). Once the scenarios are completed, the user is given their individual results. However, all results can be per one user or as an aggregated result across all users.

The system offers the user 3 different options to choose from in each randomly generated scenario; a timer is utilized in at least half of the given scenarios to add a refined dataset by putting the user under the pressure of a time constraint. This can spark conflict in someone’s moral code as the program will choose Option 1 by default if the user fails to confirm their choice within the time limit.

The results delivered post-simulation provides the participant with details about the scenarios they worked through, such as which type of pedestrian was sacrificed the most, the age group sacrificed the most and least and the most picked option which informs the user on their ethical approach to making decisions:

* Choosing A shows a Deontological approach.
* Choosing B shows a Virtue approach.
* Choosing C shows a Utilitarian approach.

Autonomous Ethics utilizes 28 unique character graphics to be used in scenarios; all of these graphics depict different pedestrians, all with varying attributes such as age, profession, and disability. All graphics were designed using Photoshop and were then mapped to be used in the program itself. Despite the time it took to find the most efficient way of mapping the images, the generation with the images incorporated now works as intended and has been fine-tuned both for functionality and aesthetic purposes.

There were several features that we wanted to add, but due to aforementioned constraints, we faced issues achieving this. Given the nature of the project, there were some areas of the subject in which none of us were greatly experienced; this included the studying of new programming languages, database implementation, and image mapping. At first, we went back and forth on what the most appropriate language would be for development; following from this, building exactly what we wanted using a combination of JavaScript, HTML, and CSS proved to be difficult. Although our time management was satisfactory, carrying out appropriate research and learning how to utilize certain tools took up a large chunk of time.

Figuring out the image mapping was also another difficult element of the project which consumed a significant amount of time.

As a direct result, this had an impact on the amount of time we had available to dedicate to the development of extra features; despite this, we were able to determine which features were critical moving forward and which ones could be placed later on our roadmap to be added at a later date.

Due to the cancellation of the poster fair it unfortunately made it difficult to receive the desired and crucial feedback to make key improvements or adjustments to our project. This may have given a more tailored final product for the end user, however using the resources available to us we sent it to individuals, and they tested it online. From this we tweaked certain aspects such as the results page display, making it more user friendly and the ability to sign in as a new user once you have already attempted the generator.

Overall, I believe our Ethics Generator project to be a success due to us having a working, functional program that gives the user the experience of the current autonomous car moral dilemma. They get a fundamental understanding of the difficult decisions programmers or artificially intelligent machines will have to make in milliseconds. Furthermore, being able to analyse their thought process at the end, on the results page and get an overview of their results compared to everyone else's, can help them see the current trend of thinking and the issues the autonomous industry is dealing with. Moreover, meeting most of our aims in this project leads me to concluding that we have completed our intended outcomes, with possible further development to expand, specific, autonomous ethics research.

## 7.2 Meeting the Criteria

When starting the project, Autonomous Ethics had a set criterion that, ideally, we would have liked to achieve by the end of the initial development process. Throughout development, certain features shifted in terms of priority - while some of these features were important, we chose to prioritize establishing proper, fluid functionality at the core of the program.

One of the things we wanted to achieve more than anything was a sense of uniqueness. We planned to do this through several different means - one, a timer which alternated between different scenarios. Having some scenarios timed and others untimed allows for a completely different type of data to be collected. It lets us see how a user’s response may differ when they are put under pressure and constrained by time. If a user fails to make a selection in the given time limit, Option 1 is chosen by default. This alone creates the sort of moral conflict that we wished to simulate with Autonomous Ethics, and as a direct consequence, we get the opportunity to compare results to a much finer detail.

On top of this came the scenario generation as a whole; writing out every single variation would be next to impossible, which is why we designed a separate sub-program tasked with generating scenarios to be used in Autonomous Ethics. The generator itself can generate several trillion scenarios using the 28 unique characters that have been designed for the project.

Autonomous Ethics can pull these scenarios from a pool, using them directly in the scenarios presented to the user. This feature was deemed to be of great importance, as not only did it allow great variety, but in the long term saved a lot of time in the scenario generation process.

Having the ability to compare individual results to overall results of the ethics generator means the user can understand the general thinking of other users and can see where they stand ethical. Delivering on these criteria was key for us as it gave a more immersive user experience while giving back more practical data to the user. This can even be used by the user or even third-party companies developing autonomous ethics in the future.

## 7.3 Further Development

Reflecting on our current version of Autonomous Ethics, one aspect we could potentially develop further could be implementing a form of algorithm that customizes proceeding scenarios. This can force the user to choose between options they have been staying away from e.g. children, infants and a disabled homeless person. Furthermore, adding another element in the scenarios such as options where the characters have a percentage chance of death or life changing injuries would give a more realistic feel to how a computer will make the decision in autonomous cars.

Looking into Option C in the scenario generator it represents the user in the car with passengers. A way to intensify this and make it more personal to the user, would be to generate these characters and give them identities such as Grandmother, Son, Father, Cousin etc. This could potentially change how the user evaluates each scenario in reference to Option C and possibly veer away from picking it, due to feeling more emotionally tied to the characters in the vehicle. Finally, a variant we could explore is other types of autonomous vehicles such as airplanes and the potential decisions they would have to make in life-and-death situations.