TrolleyMod v1.0: An Open-Source Simulation and Data Collection Platform for Ethical Decision-Making in Autonomous Vehicles

AAAI Press

Abstract

This paper presents TrolleyMod v1.0, an open-source platform based on the CARLA simulator for the collection of ethical decision-making data for autonomous vehicles. This platform is designed to facilitate experiments aiming to observe and record human decisions and actions in high-fidelity simulations of ethical dilemmas that occur in the context of driving. Targeting experiments in the class of trolley problems, TrolleyMod provides a seamless approach to creating new experimental settings and environments with the realistic physics-engine and the high-quality graphical capabilities of CARLA and the Unreal Engine. Also, TrolleyMod provides a straightforward interface between the CARLA environment and Python to enable the implementation of custom controllers, such as deep reinforcement learning agents. The results of such experiments can be used for sociological analyses, as well as the training and tuning of value-aligned autonomous vehicles based on social values that are inferred from observations.

Introduction

With the rise in technological advancements and investments in autonomous vehicles, the issue of ethical decision-making in such systems is becoming growingly pronounced (Goodall 2014). Similar to human drivers, the Artificial Intelligence (AI) controllers of driverless vehicles will inevitably face moral dilemmas, which cannot be solved by adherence to simple ethical principles (Dwork et al. 2012). A well-known instance of such dilemmas is the *trolley problem* (Jarvis Thomson 1985), where by some means (e.g., brake malfunction), the vehicle is put in a situation that forces the driver to decide between hurting bystanders on one side or another. Considering the enhanced computational and observational abilities of AI, it is expected that autonomous vehicles make *better* decisions than human drivers in such circumstances. However, the definition of *better* is deeply rooted in complex ethical principles and policies that are difficult (if not practically impossible) to formally specify (Greene et al. 2016). Furthermore, such principles are often dynamic across cultural, geographical, and temporal dimensions (Awad et al. 2018).

While the problem of ethical decision-making in AI has remained an open challenge for many decades (Wallach and Allen 2008), recent studies on data-driven approaches to this problem have reported promising advances towards practical solutions (Kasenberg, Arnold, and Scheutz 2018). Another advancement is the proposal to model the dynamics of ethical decision-making within the abstraction of *value-alignment*, which reduces the problem to the identification and measurement of the ethical norms and values of the society, and implementing such values into AI (Arnold, Kasenberg, and Scheutz 2017). The quantification and modeling of such norms and values can be performed via machine learning and parametric techniques, examples of which include those that are based on Inverse Reinforcement Learning (IRL) (Abel, MacGlashan, and Littman 2016) and norm inference (Kasenberg, Arnold, and Scheutz 2018). A major hurdle in such approaches is that of data collection (Kim et al. 2017). Modern machine learning approaches and model-dependent techniques require a large number of samples that provide a representative dataset of the societal choices and ethical values (Noothigattu et al. 2017). In particular, due to the rarity of representative ethical dilemmas in controlled real-world observations, the majority of current studies resort to simulation-based experiments.

However, the experimental setups and configurations utilized by such studies suffer from a number of shortcomings. Firstly, the bulk of such setups do not provide the means for measuring the performance of human operators in real-time, and thus fail to present a baseline for benchmarking the performance of AI against humans. Secondly, the majority of simulation environments used in published experiments (e.g., (Awad et al. 2018)) provide a very simple depiction of the conditions in real-world dilemmas, and hence may fail to account for salient details that are crucial to ethical decision-making. Thirdly, most of the recent experiments are performed in simulation environments that are either not open-source, or are very difficult to reconfigure and customize for new experiments.

In response to the aforementioned gap, this paper presents TrolleyMod v1.0, an open-source simulation platform based on the CARLA simulator for autonomous driving research (Dosovitskiy et al. 2017). Targeting experiments in the class of trolley problems, TrolleyMod provides a seamless approach to creating new experimental settings and environments with the realistic physics-engine and the high-quality graphical capabilities of CARLA and the Unreal Engine (Games 2007). Also, TrolleyMod provides a straightforward interface between the CARLA environment and Python to enable the implementation of custom controllers, such as deep reinforcement learning agents in Tensorflow (e.g., (Liang et al. 2018)) or Pytorch (e.g., (GENANDER and NYLANDER)). The details of TrolleyMod are presented in the remainder of this paper, organized as follows: We present an overview of experimental platforms used in data-driven ethical decision-making studies, followed by the architectural details and components of TrolleyMod. We conclude the paper with remarks on plans for future extensions of this project.

Previous Work

While the philosophical debate and research on ethical decision-making in AI is decades old, the engineering work in this area is very recent (Kasenberg, Arnold, and Scheutz 2018). In particular, the interest in creating moral autonomous agents has rapidly grown in the past few years. Specifically, advances in imitation learning (Taylor et al. 2016), inverse game theory (Wang, Wan, and Wang 2017), and IRL (Abel, MacGlashan, and Littman 2016) have triggered a growing investment into the research on data-driven approaches to the modeling and transfer of human ethics to autonomous agents. Such approaches rely on collection of data on ethical decisions and actions of many human subjects to derive a representative model of the societal ethical principles (Noothigattu et al. 2017).

In the domain of autonomous vehicles, very few platforms for this purpose are reported, and even fewer are openly available to the research community. Of the best known ethical data collection platforms for autonomous vehicles is MIT's Moral Machine (Kim et al. 2018) project, which performs a large-scale crowdsourcing of ethical opinions for a few simple cases of the trolley problem. While this project has yielded many interesting results (Awad et al. 2018), it can be argued that the low fidelity of the experiments, as well as the inconsideration of temporal and cultural externalities, diminish the feasibility of inferring practical ethical policies that can be implemented in real-world driverless vehicles. Furthermore, the Moral Machine experiment fails to account for the effect of environmental nudges (Leonard 2008) in the simulated scenarios. Also, while the datasets and analytics of this project were recently made available, the code for the simulation software itself is not accompanied, and hence does not accommodate the adoption, extension, and customization of this platform for further research.

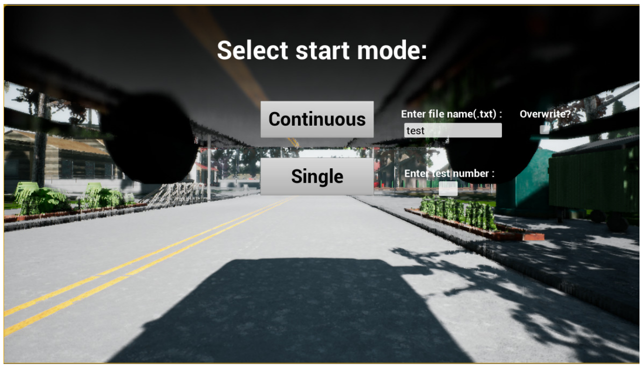
While at a smaller scale, the Ethical Autonomous Vehicles[1] project also provides the means for data collection and experimentation on ethical decision making. Albeit, this project seems to be outdated due to lack of maintenance. Also, the limited scenarios and flexibility of this project, as well as the lack of documentation, may render the usability of this project for extended research infeasible.

In (Frison, Wintersberger, and Riener 2016), authors present an alternative data collection approach using a kinetic high-fidelity driving simulator equipped with a fully automated autonomous driver. Unfortunately, the authors do not provide more information on the setup and replication procedures for their experiments.

While these prior projects succeeded in providing very valuable insights into the problem of ethical decision-making, there remains a need for an open-source, well-documented, high-fidelity, and highly-flexible simulation platform to further facilitate experimentation and research on crowdsourced ethical models for autonomous vehicles.

TrolleyMod v1.0

The goal of TrolleyMod is to provide an easy-to-use and flexible platform for setting up the conditions pertaining to ethical dilemmas. While the current version is mainly focused on the variations of trolley problem, this platform supports seamless customization and adjustment of the experiment to broader classes of ethical decision-making, such as abiding by the traffic laws. TrolleyMod is designed to facilitate experiments that require high-fidelity simulations of ethical dilemmas that occur in driving tasks to observe and record human decisions. The results of such experiments can be used for sociological analyses, as well as the training and tuning of value-aligned AI agents for autonomous driving based on social values that are inferred from observations.



Enter file name(.txt) : test

Overwrite? □

Enter test number :

Figure 1: Startup menu of a TrolleyMod simulation

In TrolleyMod, the trolley problems are set up such that there are one or more victims on each side of a road belonging to a simulation environment. The environment (i.e., map) may be designed to represent realistic conditions in urban, suburban, highway, or custom settings. In each episode of the experiment, the subject is put in the driver's seat of a vehicle that accelerates automatically. The only control actions available to the subject are swerving left or right. Furthermore, the conditions of dilemma can be reinforced via invisible barriers to ensure that the only available options to the driver are to collide with at least one group of victims or other objects. Upon collision, the subject's actions are written into a text file or a network socket for processing. In TrolleyMod, each individual Trolley problem is called a *scenario*. A sequence of scenarios is what constitutes a

[1] http://mchrbn.net/ethical-autonomous-vehicles/

TrolleyMod simulation. The components and procedures of a simulation are detailed as follows:

CARLA

As the name suggests, TrolleyMod is a modification of the CARLA (Car Learning to Act) simulator (Dosovitskiy et al. 2017). CARLA is an open-source simulator for urban driving, designed to support training, prototyping, and validation of autonomous driving models. This platform allows for the flexible configuration of sensor suites and provides various signals useful to the training of driving tasks, such as GPS coordinates, speed, acceleration, and detailed data on collisions. CARLA is implemented over Unreal Engine 4 (Games 2007) to provide flexibility and realism in the physics and high-fidelity visualization of driving environments. CARLA follows a client-server architecture to provide an interface between the world and various types of agents. In this architecture, the server runs the simulation and renders the scene, while the client uses a Python API to interact with the simulation.

The environments of CARLA are composed of 3D models of static objects, such as traffic signs, buildings, infrastructure, vegetation, as well as dynamic objects, such as vehicles and pedestrians. The default sensors in CARLA are comprised of RGB (Red, Green, and Blue) camera and pseudo-sensors which provide semantic segmentations in terms of road, traffic sign, sidewalk, pedestrian, etc. Furthermore, CARLA provides various measurements associated with the state of the agent as well as compliance with traffic rules. Such measurements include orientation and location, acceleration vector, speed, and the cumulative impact of collisions. Signals corresponding to traffic rules include the state of traffic lights and speed limits.

Components of TrolleyMod

There are three types of objects that can be spawned automatically with TrolleyMod: pedestrians, vehicles, and prop objects. These correspond to the Walker, the CarlaWheeledVehicle, and StaticActor classes respectively. The first two classes are native to CARLA, while StaticActor is a product of TrolleyMod. Our extension adjusts these classes to enable their instance to store new information pertaining to trolley scenarios. All three classes share two variables:

1. TestNum: an integer that tracks what scenario is being run.
2. GroupMemberNames: a delineated string that lumps together the names and properties of all victims in the scenario to which the class instance belongs.

Walker actors also retain the following additional information for the pedestrian character: age, gender, ID number of the pedestrian's group in the scenario, size of the pedestrian's group, and special traits (e.g., pregnant, disabled, etc.)

Configuration

TrolleyMod spawns object models into a "level" (i.e., episode of simulation) in the Unreal Engine 4. It takes an input text file, which contains a list of tuples specifying an object (e.g., pedestrian, car, cones, etc.) that can be collided with by the subject and provides any other ancillary data that describes the object. This text is parsed to provide a reference to Unreal Engine's Actor class in memory. TrolleyMod implements a custom format for the specification of objects in a scenario, as detailed in the online documentation[2].

To generate a scenario in TrolleyMod, the experiment designer runs a generation Blutility ((Editor Utility Blueprint)) named ScenarioGen. A blutility is an Unreal Engine object that enables execution of functions outside of the runtime. In TrolleyMod, blutilities are essential for providing a means to quickly modify the CARLA-provided maps to facilitate the generation of various scenarios in a relatively short period of time.

After the ScenarioGen blutility is run, the experiment designer may add other scenarios in unused locations in the map by manually placing Actor objects into the level, as illustrated in Fig. 3. Also, for the data recording script to recognize a new scenario, a Target object must be placed into the map first.

TrolleyMod provides the necessary functions and assets to set up scenarios, but it is up to the user on how to arrange and utilize them in customized environments and settings. Since each user will want to define a simulation differently, TrolleyMod only provides a functional framework. Further details on this procedure is available in the online documentation.

Execution of Simulation

TrolleyMod utilizes a modular structure based on *Blueprint Function Library*, which is an Unreal Engine feature that allows for the functions in the library to be reused over multiple projects. In TrolleyMod, blueprints provide a prepackaged set of functions to be used in the level blueprint of maps derived from CARLA's native environments. Currently, TrolleyMod contains only one blueprint, named "FunctionLibrary". This blueprint contains all of the functions required to run TrolleyMod simulations. When the subject's agent collides with a victim, that result is recorded and the next scenario is spawned. Once all scenarios are complete, the simulation terminates and the results are written to a text file or network socket.

Each TrolleyMod simulation includes an *Event Tick* node, which is called at every frame during the simulation. The purpose of this node is to make the car controlled by the subject to automatically accelerate, thus forcing the subject to decide quickly on what victim to collide with. As depicted in Fig. 2, the initial event object of this blueprint (i.e., *BeginPlay*) first calls the *InitValues* function in order to initialize the environment and object attributes. Then, it binds the subject's agent to a hit event –that is, whenever the agent collides with any object in the map, it will generate an event that calls the *OnActorHit_Event_0 node*. Next, assuming the agent hits one of the intended victims, the blueprint generates a text message describing what the player collided with using the *SetDisplay* function. If the player does not

[2] `https://github.com/zminton/TrolleyMod/wiki`

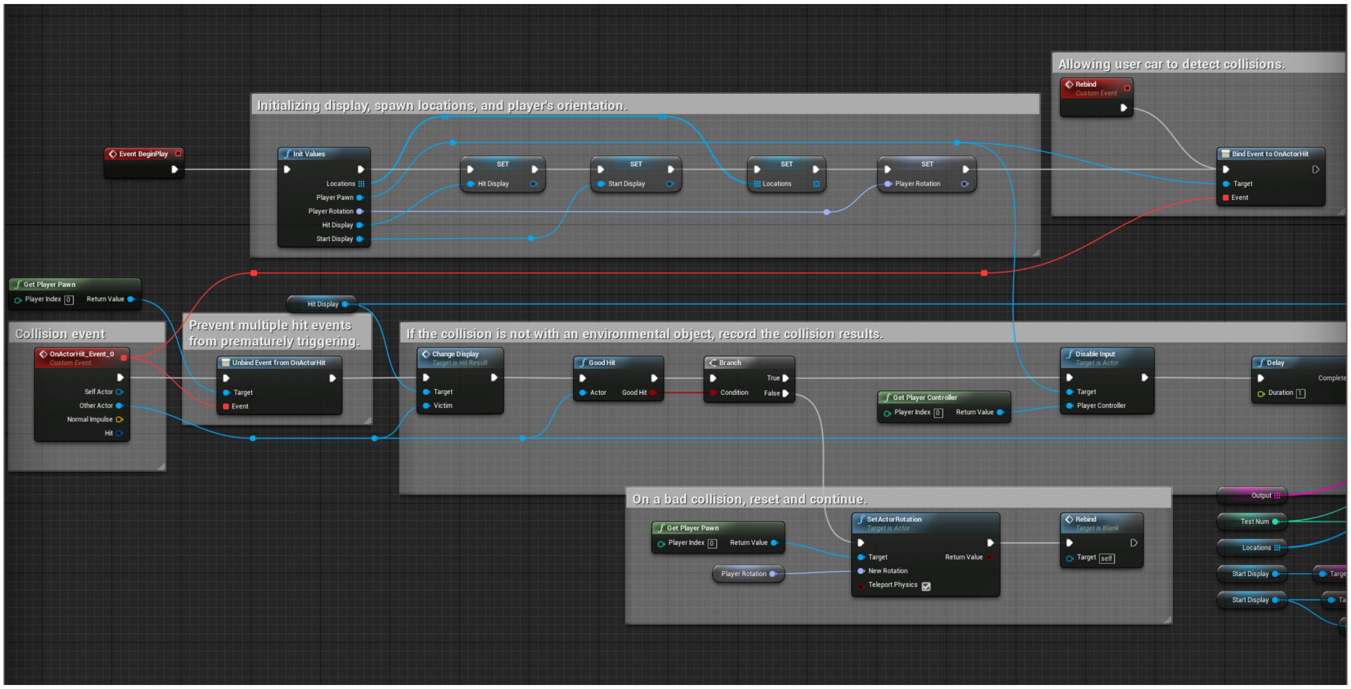


Figure 2: Blueprint functions for execution and control of the simulation

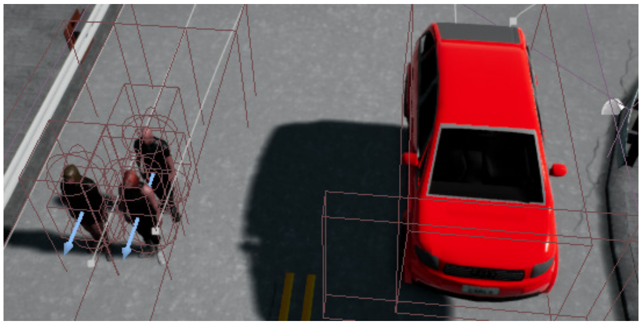


Figure 3: Manual configuration of Actor objects

hit an intended target, the simulation refers to the *PlayerRotation* variable to reset where the car is facing and rebinds hit events to it.

Otherwise, the *CollisionHandler* function analyzes the Actor hit by the player, extracts its data to the Output variable, and then prepares the simulation to move on to the next scenario. This continues until all the scenarios in the *Locations* array have been visited, after which the simulation writes the data to a text file. It is also possible for the player to specify only one scenario to complete using the start menu, and consequently the simulation will terminate after the specified scenario is run.

Future Work

TrolleyMod is a recently conceived and an ongoing project, hence there are many interesting directions for pursuit of its extension. First and foremost in our roadmap is to demonstrate the integration of TrolleyMod with inference techniques such as IRL. Furthermore, our plan for the near-future includes the gradual extension of functionalities to support seamless configuration for more well-known moral dilemmas, and enhance the front and backend to provide better support for handling larger numbers of experiments and volumes of results in a distributed architecture. A similar priority is the development of web and smartphone interfaces to enhance the reach and accessibility of this platform for large scale research. As noted, TrolleyMod is an open-source project with the aim of facilitating research on ethical autonomous vehicles, and is looking forward to contributions and comments from the community.

References

Abel, D.; MacGlashan, J.; and Littman, M. L. 2016. Reinforcement learning as a framework for ethical decision making. In *AAAI Workshop: AI, Ethics, and Society*, volume 92.

Arnold, T.; Kasenberg, D.; and Scheutz, M. 2017. Value alignment or misalignment–what will keep systems accountable. In *3rd International Workshop on AI, Ethics, and Society*.

Awad, E.; Dsouza, S.; Kim, R.; Schulz, J.; Henrich, J.; Shariff, A.; Bonnefon, J.-F.; and Rahwan, I. 2018. The moral machine experiment. *Nature* 1.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the*

3rd innovations in theoretical computer science conference, 214–226. ACM.

Frison, A.-K.; Wintersberger, P.; and Riener, A. 2016. First person trolley problem: Evaluation of drivers' ethical decisions in a driving simulator. In *Adjunct proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*, 117–122. ACM.

Games, E. 2007. Unreal engine. *Online: https://www. unrealengine. com*.

GENANDER, J., and NYLANDER, A. Control of self-driving vehicles using deep learning.

Goodall, N. J. 2014. Machine ethics and automated vehicles. In *Road vehicle automation*. Springer. 93–102.

Greene, J.; Rossi, F.; Tasioulas, J.; Venable, K. B.; and Williams, B. C. 2016. Embedding ethical principles in collective decision support systems. In *AAAI*, volume 16, 4147–4151.

Jarvis Thomson, J. 1985. The trolley problem. *Yale Law Journal* 94(6):5.

Kasenberg, D.; Arnold, T.; and Scheutz, M. 2018. Norms, rewards, and the intentional stance: Comparing machine learning approaches to ethical training. In *Proceedings of the First AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society*.

Kim, R.; Kleiman-Weiner, M.; Abeliuk, A.; Awad, E.; Dsouza, S.; Tenenbaum, J.; and Rahwan, I. 2017. A computational model of commonsense moral decision making.

Kim, R.; Kleiman-Weiner, M.; Abeliuk, A.; Awad, E.; Dsouza, S.; Tenenbaum, J.; and Rahwan, I. 2018. A computational model of commonsense moral decision making. *arXiv preprint arXiv:1801.04346*.

Leonard, T. C. 2008. Richard h. thaler, cass r. sunstein, nudge: Improving decisions about health, wealth, and happiness.

Liang, X.; Wang, T.; Yang, L.; and Xing, E. 2018. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. *arXiv preprint arXiv:1807.03776* 1.

Noothigattu, R.; Gaikwad, S.; Awad, E.; Dsouza, S.; Rahwan, I.; Ravikumar, P.; and Procaccia, A. D. 2017. A voting-based system for ethical decision making. *arXiv preprint arXiv:1709.06692*.

Taylor, J.; Yudkowsky, E.; LaVictoire, P.; and Critch, A. 2016. Alignment for advanced machine learning systems. *Machine Intelligence Research Institute*.

Wallach, W., and Allen, C. 2008. *Moral machines: Teaching robots right from wrong*. Oxford University Press.

Wang, Y.; Wan, Y.; and Wang, Z. 2017. Using experimental game theory to transit human values to ethical ai. *arXiv preprint arXiv:1711.05905*.