ENGINEERING

McMaster University

Infrastructure for Self-Driving Vehicles

Final Report

ENGINEER 2PX3 – Integrated Engineering Design Projects

Tutorial 14

Self-Driving-56

Nivetha Kuruparan (kuruparn) - 400318344

Mohamad-Hassan Bahsoun (bahsounm) - 400309354

Karanvir Braich (braick1) - 400321832

Aithihya Kompella (kompella) - 400310794

Submission Date: April 14th, 2022

Contents

Executive Summary **3**

Introduction **6**

Conceptual Design **7**

Final Proposed Design **11**

Conclusions **11**

References **14**

Appendices **15**

Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Mohamad-Hassan Bahsoun 400309354

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Karanvir Braich 400321832

X____________________

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Nivetha Kuruparan 400318344

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Aithihya Kompella 400310794

X Aithihya

Executive Summary

Traffic can be frustrating for everyone. There are many instances where people have found themselves in situations beyond their control, which negatively affect their commute. Currently, highways are bumper to bumper during rush hour causing many people to drive distractedly or end up trying to speed past other cars. In fact, there were approximately 35,972 accidents in Ontario, where inattentive driving is the leading cause of those accidents [1]. Hence, to combat the issues faced on the highway, this report discusses a redesigned highway infrastructure that can accommodate self-driving vehicles. Self-driving vehicles are not only efficient at communicating with other autonomous vehicles, but they are also better regarding safety and are much more environmentally friendly. In addition, they can accurately detect and change speeds according to vehicles surrounding them [2]. Therefore, a highway design that accommodates self-driving vehicles has the potential to improve the performance of all vehicles. A new design can also help more self-driving vehicles integrate themselves into highways.

In term of the design itself, there were many difficult choices that had to be made to account for different stakeholders and the PERSEID layers [3]. The design consists of one lane that only self-driven vehicles have access to and two lanes that will act as the regular lanes where all vehicle types can drive. Once the self-driven enters the highway, it will go to the self-driven lane when it is safe to do so and remain there until it needs to exit. The only time it will exit the lane otherwise is if it needs to pass. In this case, the self driven vehicle will use one of the regular lanes to safely pass. One of the more difficult choices came when considering the passing as two lanes self-driven lanes would allow self-driven to pass without going into the regular lanes. However, this would have a major impact on the socio-cultural layer as self-driven vehicles are being incentivized. So, the group chose to go with one lane only as opposed to two. Another relevant choice that was made to improve the socio-cultural aspect of the design was introducing on and off-peak hours. During on-peak hours the self-driven lane will be used for only self-driven vehicles to make the flow of traffic more efficient decreasing congestion. During off-peak hours, the human-driven vehicles will be allowed to use the self-driven lane as well as there is not much traffic at these times. As a result, society will not feel as if self-driven vehicles are being incentivized heavily as they do not have major advantages over people who choose to stick with human-driven vehicles.

There are many advantages to this design compared to others. To begin with, this highway does not require that much construction. Most highways are 3 - 4 lanes, so only some would require the addition of one lane compared to other designs that may require more. This design is adaptable to change. During all possible

scenarios (peak/off peak) all vehicles on the road can experience the most effective commute times. A big advantage with this design comes with the performance and the socio-cultural aspects of the design. Performance wise this design results in faster commute times and can accommodate for more vehicles on the highway. In terms of the sociocultural aspects human driven vehicles have more lanes, and current highways have express lanes so they can think of the self-driving lane as a type of express lane.

More considerations need to be implemented to create a fully functional design. One of the most prominent factors that the team has not yet considered is what stretch of highway would the design be implemented on. Currently, the assumption for this design is that it can be enforced on most major highways, such as the 400 series. However, the team has not looked at other uncommon highways. In addition, the physical aspect of the design has not been considered yet. For instance, the number of resources, workload, and amount of time has not been regarded when planning the design. The final design is very general and needs to be customized for different situations. To achieve this, meetings with municipal contractors would have to be set up to further discuss the implementation of the design. For this project, this was not required but would be crucial for a real-life scenario.

Introduction

The Infrastructure for Self-Driving Vehicles project takes a closer at the flaws of the current infrastructure and how self-driven vehicles can be implemented to make improvements [2]. For instance, traffic congestion, reckless driving, and fatal collisions that drivers face daily can all be improved through a newer and more efficient infrastructure. Self-driven vehicles have many advantages over regular vehicles as they can detect their surroundings better by using sensors [2]. Using a systematic approach, reckless behaviour such as sudden lane changes and braking can be prevented to create an overall better driving experience. However, there are still many parameters that must be considered when designing an infrastructure such as speed, safe lane changing distance, and safe follow distance. To determine how each parameter will be adjusted, the main stakeholders were first identified.

The most significant stakeholders that are affected by these issues were deemed to be the average commuter, pedestrians/cyclist, and professional drivers. Through stakeholder analysis done in milestones over the weeks, the primary stakeholder was determined to be the average commuter. The average commuter is somebody who uses the roads regularly to get to a certain destination. This group of people was most important as they are the ones that use the roads most. Commuters must deal with the traffic congestion and reckless acts during their trips and a new infrastructure could minimize these issues creating a better commuting experience. So, the main objective that was considered throughout this design was decreasing the average commute.

However, the safety aspect was kept in mind to ensure that the design does not cause the number of fatal collisions to increase. Keeping all objectives and stakeholder in mind, the refined problem statement was written as follows: “Redesign a stretch of highway that accommodates self-driven vehicles to decrease the average commute time while maintaining a high level of safety”.

Based on the problem statement, each PERSEID layer was looked at to determine objectives and constraints that would be imposed on the final design [3]. As mentioned before, one of the main objectives was to decrease the average commute time which came technical performance and stakeholder analysis. One of the main advantages of a self-driven vehicles over a human-driven vehicles is its ability to detect its surroundings. So, the human errors involved with acts such as changing lanes and sudden braking could be minimized making a more efficient system. In addition, another objective that the group came up with was decreasing the total number of fatalities that occur due to collisions. Along with increased efficiency, self-driven vehicles are also able to perform tasks in a more systematic approach. The algorithm will ensure that the surroundings have been checked thoroughly before performing an action such as lane changing. As a result, reckless acts can be minimized or even eliminated leading to an increased level of safety in a new infrastructure. So, the group made it an objective to decrease the overall number of deaths in the new infrastructure.

The main components that this report will cover are conceptual design, the final proposed design, and some conclusions that the group came to. In the conceptual design section, a few alternatives designs will be examined and compared against chosen design using a weighted decision matrix. In addition, each design will be briefly evaluated through the PERSEID layers. In the final proposed design section, a description of the final design is provided, and specific design choices will be addressed and justified based upon team discussions throughout the weeks. Furthermore, important results will be summarized using drawings, simulation outputs, and figures. Lastly, there will be a brief discussion of how each of the objectives and constraints are met through the simulation. In the conclusion section, the group reflects on what could have been achieved given more time and things that could have been modified throughout the design process to create a better product.

Conceptual Design

Design Alternatives:

There were many potential designs that the team brainstormed that they believed would satisfy parts of the PERSEID method. Below are descriptions of two design alternatives. A brief description of the final design can also be found below, to justify the reasoning for the decision matrices and design evaluation.

<u>Design Alternative 1: 4-Lane Highway</u>

This design is a four-lane highway that includes a dedicated lane for self-driving vehicles, two regular lanes, and an emergency lane, as shown in Figure 1. This design's purpose is to allow self-driving vehicles to drive without any interference from human drivers and allow them to perform as efficiently as possible. When a self-driving vehicle merges onto the highway, it has a chance to drive on the regular two lanes. However, when safe to do so, it must make its way to the self-driving lanes. As we are assuming that all self-driving vehicles can communicate with each other when a vehicle is trying to merge onto the self-driving lane, all the other cars would accommodate their speed. Moreover, this design also allows emergency vehicles to have safe and fast access if an accident were to happen on the road by using the empty emergency lane.

This design also considers if an accident occurs in the self-driving vehicle lane. As seen in Figure 2, the lanes would essentially shift to the right and the left-most lane would become the emergency lane, allowing emergency vehicles to respond to the scene as quickly as possible, as well as not hindering the performance of the other vehicles.

Design Alternative 2: 5-Lane Highway

As seen in Figure 2, this design is a five-lane highway that incorporates a dedicated lane for self-driving vehicles, a passing lane for self-driving vehicles, two regular lanes, and an emergency lane. The purpose of this design is to increase the performance of self-driven vehicles. The left-most lane, deemed the self-driving lane, allows self-driving vehicles to follow a target speed and safe-follow distance. The lane next to it, deemed the passing lane, allows for an efficient merge when self-driving vehicles want to either merge into or out of the self-driving lane. However, compared to Design 1, where there is no passing lane, this lane allows for self-driving vehicles to merge in and out more efficiently as they do not have to encounter human-driven vehicles. Hence, this design minimizes the amount of human error that self-driving vehicles experience on the road. Like design 1, this design also includes two regular lanes and an emergency lane.

Final Design: 4-Lane Highway with Peak and Off-Peak Times

The final design has a similar infrastructure to design alternative 1. It consists of a self-driving lane. two regular lanes, and an emergency lane. It also has an additional feature where it has set times called peak and off-peak. If the highway was set to peak time, the highway would function like design alternative 1 as seen in Figure 3a. If the highway was set to off-peak time, the highway would convert to a regular three-lane highway with an emergency lane, as seen in Figure 3b. A more thorough explanation of the final design is found in the Final Proposed Design of this report.

Decision Matrices:

PERSEID Considerations

When choosing between the designs, the PERSEID layers helped determine which design is better in terms of different considerations. As shown in Table 1, the team had several considerations for each layer that the final design must incorporate.

First, when looking at the technical performance layer, the most important consideration was to ensure that the commute times for all vehicles were as efficient as possible for all types of scenarios that could potentially happen on a highway. This meant that both the human-driven vehicles and self-driven vehicles must experience a faster commute time than on an original highway. It also meant that the performance of all vehicles should not be hindered in scenarios. For instance, if there was an accident, the performance of all other vehicles should not drastically decrease.

Next, when looking at the safety and regulatory layer, the most important consideration was to ensure that the highway was as safe as possible. This meant the number of deaths and crashes do not increase as well allowing emergency vehicles to arrive at an accident on the road as quickly as possible.

For the socio-cultural layer, the largest factor that came into play was not incentivizing self-driving vehicles. Since this design essentially accommodates self-driving vehicles, people that can not afford them should not feel the pressure of buying a self-driven vehicle. Hence, the performance of self-driving vehicles and human-driven vehicles should remain relatively the same to allow for fairness. To elaborate further, if self-driving vehicles have a faster commute time compared to human-driven vehicles, many of those human drivers might find that the highway does not accommodate them. Hence the constraint was implemented that the commute times of self-driving vehicles should not be less than 75 percent of the commute times for human-driven vehicles.

Lastly, the environmental layer looked at considerations such as lane usage. The design should remain conscious of how many lanes it was using and should not precede that of an original highway. It also shouldn't emit more carbon emissions than a regular highway. Another important consideration is that the overall design of the new infrastructure shouldn't use too many materials and resources that can potentially negatively impact the environment.

Constructing the Decision Matrix

Once the considerations were made, a weighting system was created as shown in Table 2, to give a weight to each PERSIED layer. This helps illustrate which considerations are more important and hold value over the others. Overall, the team felt that the performance layer was the most important as the purpose of this design is to accommodate self-driving vehicles on highways. The reason for the accommodation is to allow all vehicles to experience efficient travel times. The second most important layer was socio-cultural. There is a heavy stigma against accommodating the wealthy and the assumption that most people who own a self-driving vehicle have a higher net worth. To ensure that everyone is comfortable with the accommodation of self-driving vehicles, the team deemed the socio-cultural layer to be an integral part of the design. Following is the regulatory layer. All vehicles must be safe on the new highway design, but the team deemed that it was not as important as the first two. This is because self-driving vehicles help prevent any human errors. To elaborate further, the scenario that causes the most accidents is distracted driving [1]. By accommodating self-driving vehicles, it helps limit the accidents that occur due to human error. For a similar reason as regulatory, environmental was put last in the weighting score simply because self-driving vehicles produce less carbon emission and are better for the environment overall. Hence, one of the environmental considerations is already met if more self-driving vehicles drive on the road.

Decision Evaluation

Shown in Table 3 is the final decision matrix. Some notable comments that can be derived from the table are the relationship between the performance and regulatory layers. As mentioned above, the design needed to find a middle ground when it comes to performance and regulatory considerations. Although an effective design would mean that all cars can travel a lot faster, the maximum speed that a vehicle can drive should still be safe.

Furthermore, shown in Table 4 are the total scores for each potential design. The five-lane highway design scored the lowest. This is due to its very low socio-cultural and environmental layers. Although the design ranked the highest in performance, the number of lanes it uses is higher than the other two designs. Similarly, the two lanes that accommodate self-driven vehicles could be seen as giving lower priority to human-driven vehicles.

Design alternative one is ranked second. Although it surpasses design 3 in the regulatory layer, design two's performance is much lower. This is because the peak and the off-peak notion that design 3 implements increase the performance for all vehicle types in all types of scenarios.

In conclusion, design 3 scored the highest compared to the other two designs. Even though the s between the three designs were not that substantial, the peak and off-peak feature ensures a more effective performance for all vehicle types.

Final Proposed Design

The final proposed design is a highway made up of 4 lanes as shown in Figure 3a and Figure 3b. As briefly mentioned above in the Conceptual Design, this highway has two times, peak time and off-peak time. When the highway is in peak time, the highway will copy the design shown in Figure 3a. This design includes a self-driving lane where only self-driven vehicles are allowed on that lane. In addition, there are two regular lanes where any vehicle type can drive on those lanes. Lastly, there is an emergency lane that emergency vehicles can go through easily in case of an accident. During peak time, all self-driving vehicles will try to merge to the self-driving lane as safely and effectively as possible. Using the assumption that all self-driving vehicles can communicate with each other when a self-driving vehicle is trying to merge in or out, all the other self-driving vehicles can change their speed to accommodate that merge. Furthermore, the human vehicles have two lanes where they can drive as if it was a two-lane highway. When the highway is in off-peak time, the highway will copy the design shown in Figure 3b. The highway would be a typical 3-lane highway with an emergency lane.

To communicate whether it is peak or off-peak time to the drivers, there will be billboards placed along the highways. In addition, communication towers can send signals to self-driving vehicles to indicate whether they need to be driving in one lane or not.

The purpose of this design is to ensure that the performance of all vehicle types is as effective as possible. For instance, if there were only a few cars on the road, having an almost empty self-driving lane isn't feasible. Similarly, if there were only human-driven vehicles on the road, an empty self-driving lane can hinder the performance of human-driven vehicles. Thus, having a peak and off-peak time, allows the highway to be adaptable to certain changes that could happen.

The decision to go with this design was made because it served the objectives and constraints the best. Many decisions throughout the project. For starters, each PERSEID layer was looked at individually. Questions about each layer were answered and a consensus was made about what each layer was trying to accomplish. For each layer, designs that catered toward that layer were made. To decide which layers were more important than others, a problem statement was made. After coming up with several designs for each layer, and understanding which layers were more important than others, some designs were combined, and some were found to be already similar. After choosing 3 of the best designs, each member of the group independently came up with their decision matrix. The reasoning behind the use of a decision matrix was to help analyze several similar designs. The resulting decision matrices were brought together and combined to make a weighted decision matrix to help us make our final decision.

Another decision that was made was who the main stakeholder was going to be. It was decided that the average commuter would be the main stakeholder. The reason for this is that they are the ones who are using the road, and they are going to be impacted most by a change in the highway's infrastructure.

The 4-lane highway with peak/ off-peak times design was chosen because of how it affected the PERSEID layers.

Technical Performance wise the commute time decreased. The one designated lane (for self-driven vehicles) will essentially be an express lane that only self-driving vehicles are allowed in. This design constraint was met. See Figure 4. You will notice that the number of cars that make it through the stretch of highway is almost double that of each human driving lane.

Environmentally it is hard to predict how exactly this design would decrease carbon emissions. It can be predicted that implementing only one new lane will not exceed the constraint provided as just one lane will not be able to accommodate too many new vehicles. This constraint can only be tested via actual implementation and further research. But what can be said is that the maintenance and any further construction of the highways will be made from environmentally stable and friendly materials.

Socio-Culturally this design did meet the constraint. The commute time of self-driving vehicles is no less than 75 percent of human driving vehicles. When you observe the table, you can see that the total number of self-driving vehicles divided by the total number of human-driven vehicles is above 75 percent.

This design will not pose many regulatory concerns as the number of deaths and crashes will likely remain the same or decrease. There will be many drivers who will switch over to self-driven vehicles reducing the number of human-driven vehicles on the roads, thus reducing human error and variables on the road.

Conclusions

During this project, using various methods such as design matrices as well as through simulations run in Python, a final design was chosen and presented to an audience. Given more time, it would be interesting to look at other ways to design the highway and fix flaws that are present in the final design. for example, having three lanes and an emergency lane proves useful in most situations, but if an accident occurs in a center lane and the lane gets blocked, this will pose a problem in terms of what would happen if the other lanes were shifted to adjust and how to let drivers in those other vehicles know what to do. Throughout this project, the team learned about the PERSEID method and how its layers may be used to make effective choices when creating a design for a project of such a large scale. It made the process of creating a final design much easier and allowed for a focus on important factors such as environmental, socio-cultural, regulatory, and performance. This process was also reliable and in general would not need to be changed, however for this specific project, it is believed that ethics would also need to be looked at, especially regarding the way self-driving vehicles interact with their environment. While it may be argued that this is an issue for vehicle manufacturers and policymakers, it must also be examined when making engineering design choices to make highways safer using the policies that come with these vehicles. The team's dynamics were good, and this experience brought to light the importance of communication and collaboration. Through good communication, the tasks at hand were able to be efficiently completed as well as collaboration which allowed for ease in completing the tasks at hand as well. As for the design process, there are some elements we could eliminate; for example, in Week 6, one of the tasks was to come up with a physics equation to simulate the stopping distance of a vehicle. This did not have anything to do with the highway infrastructure that was to be designed, and therefore was not a necessary step in choosing a design. In addition to this, while thinking about wildlife may be important, for this project, it was not a factor and was decided would be a factor to consider for separate infrastructure designs. The group worked ideally and did not face any major hitches or disagreements, however if there is one thing that must be changed for the future it would be that more diverse ideas and conversations need to be had. Many of the ideas that were conceived were agreed upon without much discussion, and there were no counterarguments made to challenge their legitimacy as well. For the future, more creativity is needed to be able to come up with more interesting and innovative ideas as well as to challenge the ideas that others within the group come up with to have stronger design ideas.

References

[1] "23+ Car Accident Statistics in Canada," Carsurance Canada, 2021.

[2] "Project Module - Infrastructure for Self-Driving Vehicles", Project Modules for ENGINEER 2PX3, Department of Engineering, McMaster University, Winter, 2022.

[3] “2PX3 Design Lecture Wk-4 – Human Centered Design, PERSEID framework,” Lecture Slides for ENGINEER 2PX3, Department of Engineering, McMaster University, Winter, 2022.

Appendices

Design Alternatives:

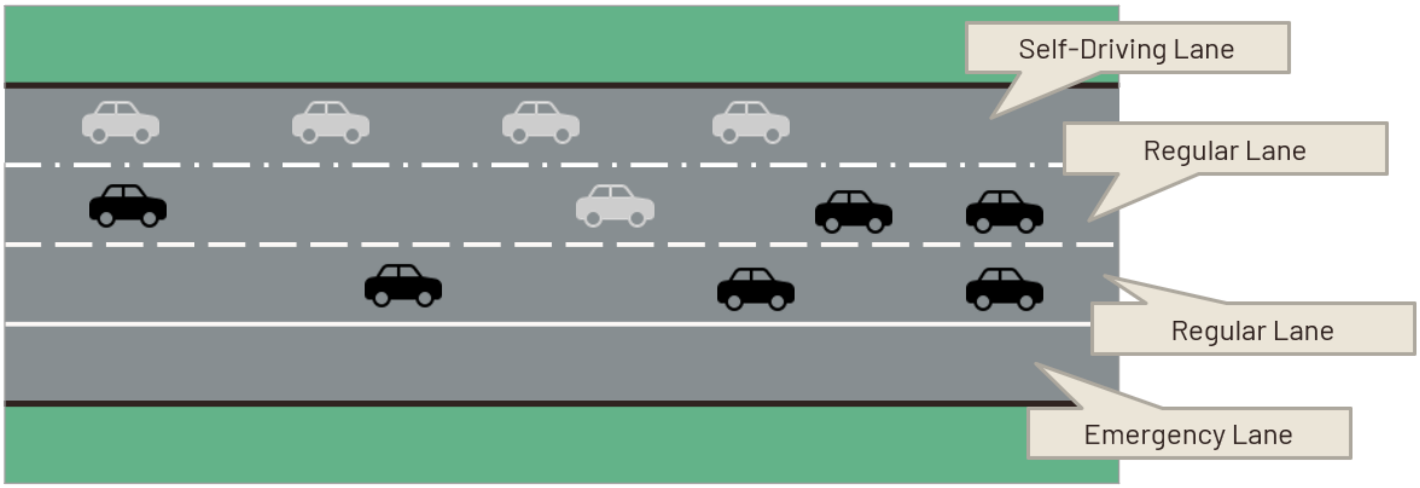


Figure 1: Design Alternative 1: 4 – Lane Highway

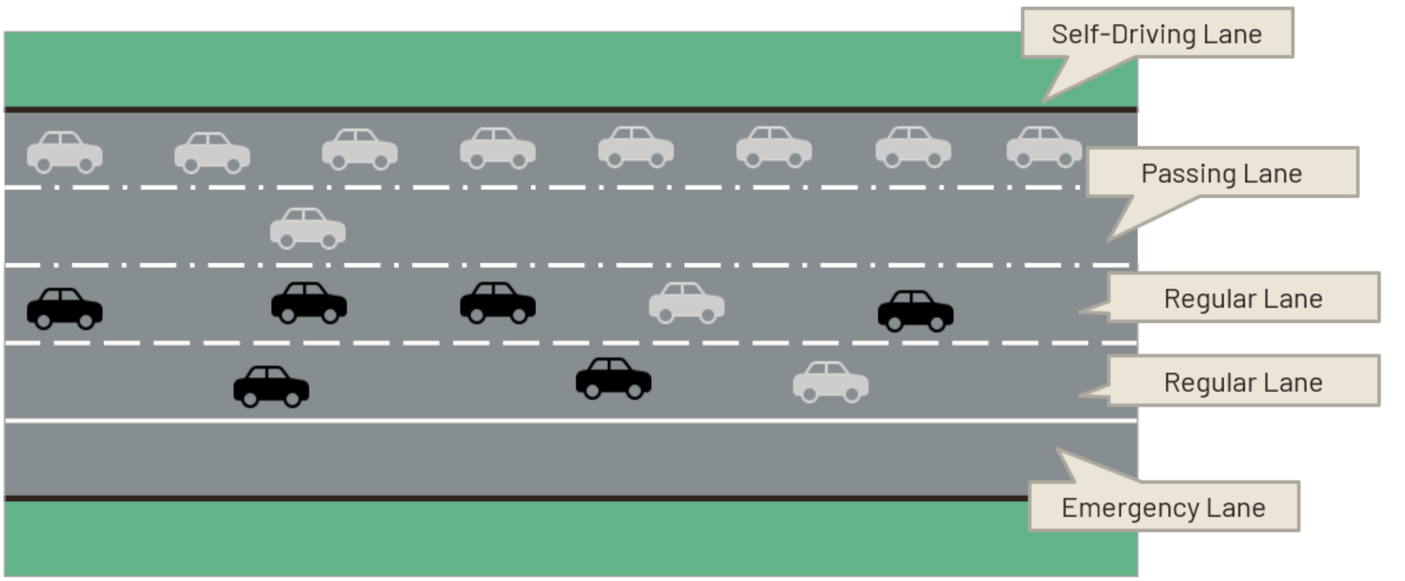


Figure 2: Design Alternative 2: 5 - Lane Highway

Final Design:

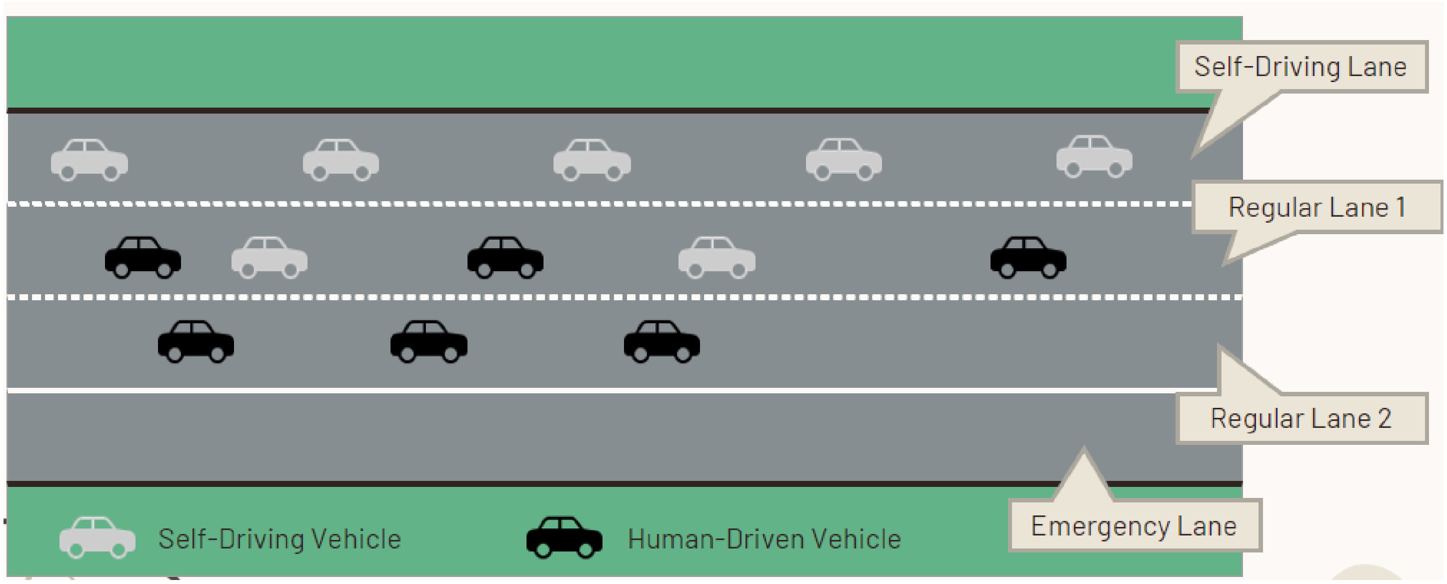


Figure 3a: Final Design – Peak Time

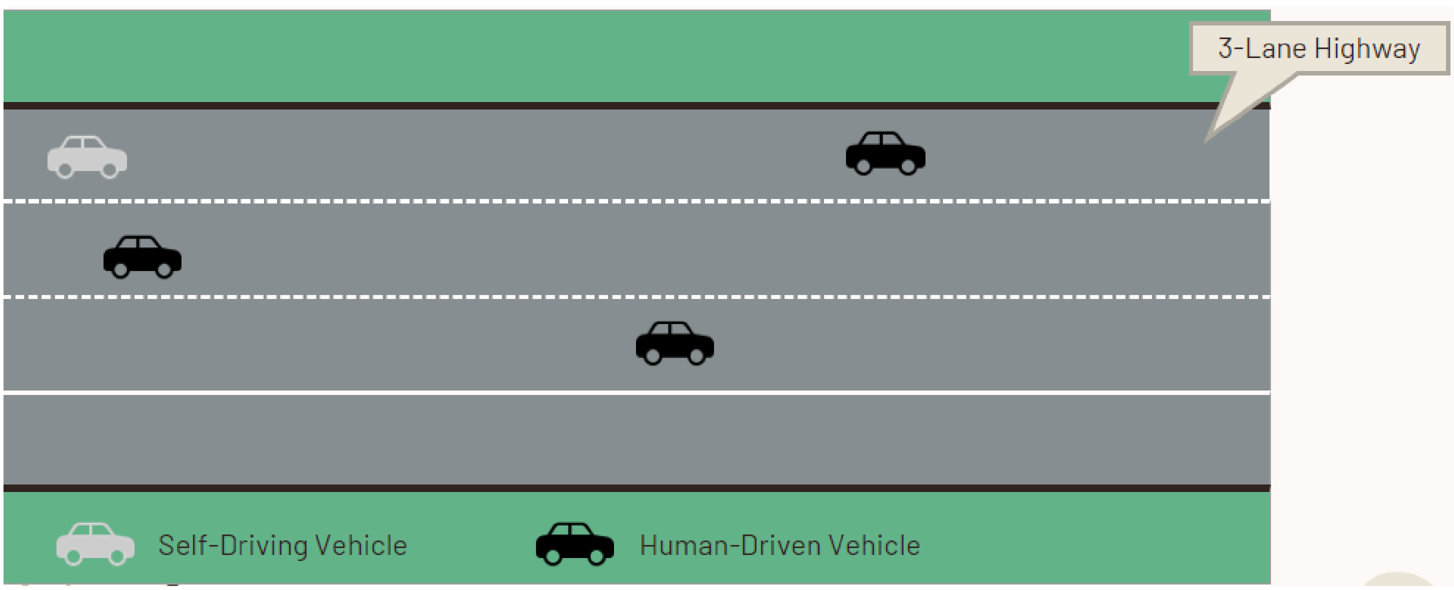


Figure 3b: Final Design – Off-Peak Time

PERSEID Considerations:

	Considerations
Performance	- Commute times for all vehicles must be less than an original highway
Socio-Cultural	- All vehicles given the same priority level - The commute time for self-driven vehicles must not exceed the commute times for human driven vehicles
Regulatory	- Number of deaths doesn't increase - Number of crashes doesn't increase - Accommodation for emergency vehicles
Environmental	- Number of lanes do not increase - Carbon emissions do not increase - Construction of highway must not harm the environment

Table 1: A list of considerations/constraints based on the PERSEID method that are to be implemented in the final design

Decision Matrix:

	Weight
Performance	5
Socio-Cultural	4
Regulatory	2
Environmental	3

Table 2: Determines the weight of each PERSEID layer

	Weight	Design 1	Design 2	Design 3
Performance	5	2	5	4
Socio-Cultural	4	4	2	5
Regulatory	2	4	3	2
Environmental	3	4	1	4

Table 3: Weights assigned to each design based on PERSEID layer

	Total Score
Design 1	10+16+8+12 = 46
Design 2	25+8+6+3 = 42
Design 3	20+20+4+12 = 56

Table 4: Total scores for each design

Data

- Ran simulation for 10,000 timesteps
- Collecting Data for:
  - Number of cars that made it to the end for each lane
  - Average number of timesteps for all cars in each lane

Lane	Trial One		Trial Two		Trial Three		Trial Four	
	Number of Cars	Ave. Number of Timesteps	Number of Cars	Ave. Number of Timesteps	Number of Cars	Ave. Number of Timesteps	Number of Cars	Ave. Number of Timesteps
Self-Driving Lane	5968	38.32	5980	38.27	5973	38.26	5914	38.29
Regular Lane 1	3473	38.08	3502	38.07	3548	38.07	3545	38.07
Regular Lane 2	3467	38.02	3498	38.02	3447	38.01	3547	38.01
Emergency Lane	0	0	0	0	0	0	0	0

Figure 3: Simulation data

Code

```python
import random
#===================================================================================================
# For the code to run, it must be run in the code in the Pything console
# The commands to type are as follows:
#To run the Simulation type:
#   sim = Simulation(time_steps) (where time steps is the number of time steps you would like to simulate)
#   sim.run() (this will show you the simulation of the roads)
#
# To get Data type:
#    sim.data (this will give you a 2D array were each array is a lane and
#              shows you how many time steps it took for a car to make it to the end of the highway)
#
# To get how many cars made it to the end of the highway for each lane type:
#   sim.madeIt()

# To get the average time step for each lane type:
#   sim.averageTimeStep
#====================================================================================================


# Parameter definitions
#====================================================================================================
EMPTY = " " # signifies an empty slot in the road
#====================================================================================================

#====================================================================================================
#Speeds for vehicles
SELF_D = 4 # speed for the self-driven vehicles 100km/h to start may change later
SELF_D1 = 5 # speed for the self-driven vehicles 110km/h
SELF_D2 = 6 # speed for the self-driven vehicles 120km/h
HUMAN_D = 4 # speed for the human-ddriven vehicles 100km/h
#====================================================================================================

#====================================================================================================
# Safe follow distances for inbetween vehicles
SAFE_FOLLOW_S = 3 # the travel distance that Human driven vehicles must keep infront of them
SAFE_FOLLOW_H = 4 # the travel distance that Human driven vehicles must keep infront of them
#====================================================================================================

#====================================================================================================
# Integers representiing the lanes of the Highway
LEFT = 0 # Self-driven lane
MIDDILE = 1 # 1st human driven lane
RIGHT = 2 # 2nd human driven lane
EMERGENCY = 3 # emergency lane
#====================================================================================================

```

```python
#==========================================================================================
# Probabilty of generating a vehicle for each type
SELF_PROBABILITY = 0.3 # used in the random spawing of self-driven vehicles going at normal speed
SELF_PROBABILITY1 = 0.2 # used in the random spawing of self-driven vehicles going faster speed
SELF_PROBABILITY2= 0.1 # used in the random spawing of self-driven vehicles going fastest speed

HUMAN_PROBABILITY_M = 0.5 # used in the random spawing of Human-driven vehicles in the middle lane
HUMAN_PROBABILITY_R = 0.5 # used in the random spawing of Human-driven vehicles in the Right Lane
#==========================================================================================

#==========================================================================================
# Other Variables
HIGHWAY_LENGTH = 150 # length of the highway
PRINT_ROAD = True
#=========================================================================================

# Function definitions
#==========================================================================================
class Driver:
    def __init__(self, type, speed, arrive_time):
        self.speed = speed
        self.type = type
        self.arrive_time = arrive_time
        if(type == "S"): # identification of car: is it a human or self-driven car
            self.safe_follow = SAFE_FOLLOW_S
        else:
            self.safe_follow = SAFE_FOLLOW_H
#==========================================================================================

```

```python
#==========================================================================================
class Highway:
    def __init__(self, length):
        # initializing the road
        self.road = [[], [], [], []]
        self.length = length
        for i in range(length):
            self.road[0].append(EMPTY)
            self.road[1].append(EMPTY)
            self.road[2].append(EMPTY)
            self.road[3].append(EMPTY)
#==========================================================================================
    def getCar(self, lane, index):
        # returns what kind of car if any at a given lane and index
        return self.road[lane][index]
#==========================================================================================
    def setCar(self, lane, index, value):
        # sets the kind of car we want a given lane and index
        self.road[lane][index] = value
#==========================================================================================
    def safe_distance_within(self, lane, index, k):
        #Returns the distance until the next car, from index i within k; returns k if all spots are EMPTY
        x = 0
        for i in range(index + 1, index + k + 1):
            if i >= self.length:
                return k
            if self.road[lane][i] != EMPTY:
                return x
            x += 1
        return x
#===========================================================================================================================
    def safe_left_lane_change_from_Right(self, index):
        # Returns true if the self driving vehicle may lane change from the Right into the Middle Lane
        # checks if the index beside it and 2 spaces infront of it are empty
        return self.road[LEFT][index-1] == EMPTY and self.road[MIDDILE][index] == EMPTY and self.road[MIDDILE][index+1] == EMPTY and self.road[MIDDILE][index+2] == EMPTY
#==========================================================================================
    def safe_left_lane_change_from_Middle(self, index):
        # Returns true if the self driving vehicle may lane change from the middle into the left Lane(self-driving lane)
        # checks if the index beside it and 2 spaces infront of it are empty
        return self.road[LEFT][index-1] == EMPTY and self.road[LEFT][index] == EMPTY and self.road[LEFT][index+1] == EMPTY and self.road[LEFT][index+2] == EMPTY
#===========================================================================================================================
```

```
    def print(self):
        # this prints out the road
        s = "\n"
        for i in range(self.length):
            if self.road[0][i] == EMPTY:
                s += "_"
            else:
                s += "S"
        s+="\n"
        for i in range(self.length):
            if self.road[1][i] == EMPTY:
                s += "_"
            else:
                s += "C"
        s+="\n"
        for i in range(self.length):
            if self.road[2][i] == EMPTY:
                s += "_"
            else:
                s += "C"
        s+="\n"
        for i in range(self.length):
            if self.road[3][i] == EMPTY:
                s += "_"
            else:
                s += "C"
        print(s)
#==========================================================================================================
```

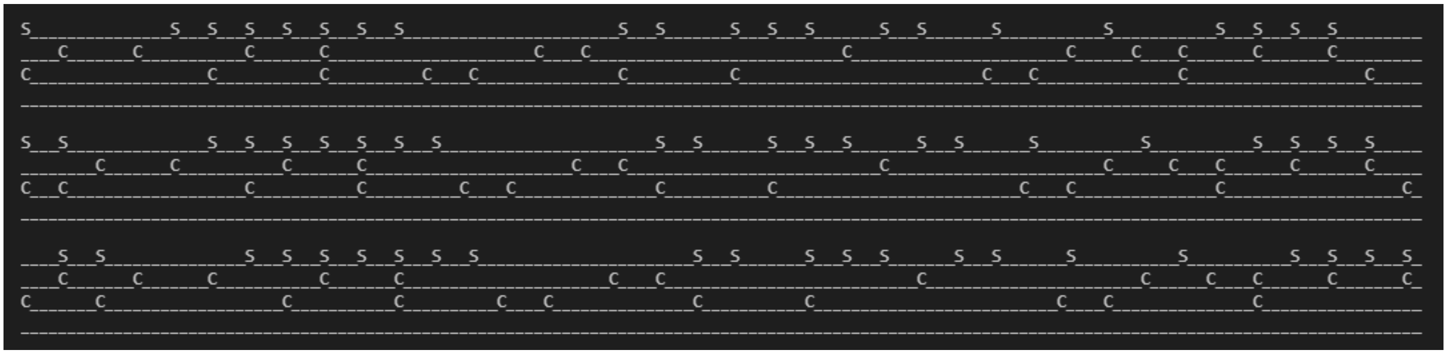
```
#==========================================================================================
class Simulation:
    def __init__(self, time_steps):
        self.road = Highway(HIGHWAY_LENGTH)
        self.time_steps = time_steps
        self.current_step = 0
        self.data = [[],[],[],[]]
#==========================================================================================
    def run(self):
        while (self.current_step < self.time_steps):
            self.execute_time_step()
            self.current_step += 1
            if PRINT_ROAD:
                self.road.print()
#==========================================================================================
    def execute_time_step(self):
        for i in range (self.road.length -1,-1,-1):

            if self.road.getCar(LEFT,i) != EMPTY:
                self.sim_self_driven_lane(i)

            if self.road.getCar(MIDDILE,i) != EMPTY:
                self.sim_human_driven_lane1(i)

            if self.road.getCar(RIGHT,i) != EMPTY:
                self.sim_human_driven_lane2(i)
        self.gen_Cars()
#==========================================================================================
    def sim_self_driven_lane(self, i):
        driver = self.road.getCar(LEFT,i)

        # Remove the car if we make it to the end of the highway
        if driver.speed + i >= self.road.length - 1:
            self.road.setCar(LEFT, i, EMPTY)
            self.data[0].append(self.current_step - driver.arrive_time)
            return

        self.sim_cruise(LEFT, i)
```

```
#============================================================================================================
    def sim_human_driven_lane1(self, i):
        driver = self.road.getCar(MIDDILE,i)

        # Remove the car if we make it to the end of the highway
        if driver.speed + i >= self.road.length - 1:
            self.road.setCar(MIDDILE, i, EMPTY)
            self.data[1].append(self.current_step - driver.arrive_time)
            return

        if driver.type == "S":
            if self.road.safe_left_lane_change_from_Middle(i):
                self.road.setCar(LEFT, i, driver)
                self.road.setCar(MIDDILE, i, EMPTY)
                self.sim_cruise(LEFT, i)
            else:
                self.sim_cruise(MIDDILE, i)
        else:
            self.sim_cruise(MIDDILE, i)
#============================================================================================================
    def sim_human_driven_lane2(self, i):
        driver = self.road.getCar(RIGHT,i)

        if driver.speed + i >= self.road.length - 1:
            self.road.setCar(RIGHT, i, EMPTY)
            self.data[2].append(self.current_step - driver.arrive_time)
            return

        if driver.type == "S":
            if self.road.safe_left_lane_change_from_Right(i):
                self.road.setCar(MIDDILE, i, driver)
                self.road.setCar(RIGHT, i, EMPTY)
                self.sim_cruise(MIDDILE, i)
            else:
                self.sim_cruise(RIGHT, i)
        else:
            self.sim_cruise(RIGHT, i)
#============================================================================================================
    def sim_cruise(self, lane, index):
        driver = self.road.getCar(lane, index)

        x = self.road.safe_distance_within(lane, index,driver.speed+driver.safe_follow)

        if x == driver.speed + driver.safe_follow:
            self.road.setCar(lane, index + driver.speed, driver) #Car moves forward by full speed
        elif x > driver.safe_follow:
            self.road.setCar(lane, index + x - driver.safe_follow, driver) #Car moves forward just enough to maintain safe_distance
        else:
            self.road.setCar(lane, index + 1, driver) #Car moves forward by just 1 spot
        self.road.setCar(lane, index, EMPTY)
```

```python
#===========================================================================================================
    def gen_Cars(self):
        #Self-Driven Lane: The probably of geerating a self-driving car in the left lane
        r = random.random()

        if r < SELF_PROBABILITY:
            r = random.random()

            if r < SELF_PROBABILITY1 and r > SELF_PROBABILITY2 : # this is to generate different speed self driven vehicles
                self.road.setCar(LEFT, 0, Driver("S", SELF_D1, self.current_step))
            elif r < SELF_PROBABILITY2:
                self.road.setCar(LEFT, 0, Driver("S", SELF_D2, self.current_step))
            else:
                self.road.setCar(LEFT, 0, Driver("S", SELF_D, self.current_step))
        #===========================================================================================================
        #Middle Lane: The probably of geerating a human-driving car in the middle lane
        r = random.random()

        if r < HUMAN_PROBABILITY_M:
            r = random.random()
            if r < SELF_PROBABILITY: # may generate a selfdriven vehicle which has to lane change to the left lane
                self.road.setCar(MIDDILE, 0, Driver("S", SELF_D, self.current_step))
            else:
                self.road.setCar(MIDDILE, 0, Driver("H", HUMAN_D, self.current_step))
        #===========================================================================================================
        #Right Lane: The probably of geerating a human-driving car in the right lane
        r = random.random()

        if r < HUMAN_PROBABILITY_R:
            r = random.random()
            if r < SELF_PROBABILITY:# may generate a selfdriven vehicle which has to lane change to the left lane
                self.road.setCar(RIGHT, 0, Driver("S", SELF_D, self.current_step))
            else:
                self.road.setCar(RIGHT, 0, Driver("H", HUMAN_D, self.current_step))
#===========================================================================================================
    def averageTimeStep(self):
        # returns a a list where each index represents the number of cars passed per Lane
        #[ [self-driving lane]   [human driving lane 1]   [human driving lane 2]   [emergency lane] ]
        return [round(sum(self.data[0])/len(self.data[0]),2),round(sum(self.data[1])/len(self.data[1]),2), round(sum(self.data[2])/len(self.data[2]),2), 0]
#===========================================================================================================
    def madeIt(self):
        # returns a a list where each index represents the number of cars passed per Lane
        #[ [self-driving lane]   [human driving lane 1]   [human driving lane 2]   [emergency lane] ]
        return [len(self.data[0]), len(self.data[1]), len(self.data[2]), len(self.data[3])]
#===========================================================================================================
    def average_time(self):
        return sum(self.data)/len(self.data)
#===========================================================================================================
```

Code Output

```
...
>>> sim = Simulation(10000)
>>> sim.run()
```



```
>>> sim.data
[[38, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 39, 40, 38, 36, 38, 38, 38, 38, 38, 39, 38, 38, 38, 39, 39, 40, 39, 38, 38, 38, 38, 37, 38, 37, 39, 40, 38, 38, 40, 38, 38, 39, 37, 39, 39, 38, 38, 38, 38, 38, 38, 38, 39, 39, 38, 38, 38, 38, 38,
 38, 39, 38, 40, 38, 38, 38, 38, 37, 39, 38, 39, 39, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 36, 38, 38, 38, 38, 38,
38, 38, 38, 38, 38, 38, 38, 38, 35, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 39, 39, 38, 37, 39, 36, 38, 38, 39, 38, 38, 38, 39, 38, 38, 38, 40, 40, 39, 40, 38, 38, 39, 39, 38, 38, 38, 38, 38, 38, 38, 38, 39, 39, 39, 39, 38, 37, 38, 3
8, 38, 38, 35, 38, 38, 38, 38, 38, 39, 37, 39, 38, 39, 39, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 38, 38, 38, 40, 38, 38, 38, 38, 38, 35, 38, 38, 39, 38, 38, 37, 39, 39, 38, 38, 38, 38, 39, 39, 39, 39, 39, 38, 38, 38, 38, 38, 38
, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 39, 37, 38, 39, 38, 38, 38, 38, 38, 38, 40, 39, 39, 37, 38, 38, 38, 38, 38, 38, 38, 38, 37, 39, 37, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 35, 39, 38, 38, 38, 38, 39, 38, 39, 39, 38, 38, 38, 39, 38,
 38, 38, 38, 38, 39, 38, 38, 38, 36, 38, 39, 38, 39, 38, 38, 39, 39, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 39, 39, 38, 38, 38, 38, 37, 37, 38, 38, 38, 38, 34, 39, 39, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 35, 38, 38, 38, 39, 38, 38,
38, 38, 36, 38, 38, 38, 38, 38, 38, 38, 38, 38, 37, 38, 39, 39, 39, 41, 41, 40, 39, 39, 40, 38, 40, 38, 39, 38, 38, 36, 38, 38, 39, 39, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 38, 39, 39, 39, 39, 38, 38, 39, 38, 38, 38, 38, 38, 38, 35, 38, 39, 3
8, 38, 38, 38, 39, 39, 39, 40, 42, 43, 41, 41, 41, 40, 39, 39, 39, 39, 38, 38, 34, 34, 38, 38, 38, 38, 38, 38, 38, 36, 38, 39, 38, 38, 38, 37, 36, 38, 38, 38, 37, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 38, 38, 36, 38, 38, 38, 39, 39, 39, 38
, 38, 38, 38, 39, 38, 38, 38, 36, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 39, 37, 38, 38, 38, 36, 38, 38, 38, 40, 40, 41, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 36, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38,
 38, 38, 39, 38, 40, 40, 38, 38, 38, 38, 37, 38, 38, 38, 38, 39, 39, 37, 36, 38, 39, 38, 39, 39, 41, 40, 38, 38, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 36, 35, 38, 38, 38, 38, 38, 37, 39, 38, 38, 38, 38, 38, 31, 38, 38, 40, 39, 39, 42, 40,
36, 38, 38, 38, 38, 37, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 39, 39, 39, 37, 39, 38, 38, 38, 37, 38, 38, 38, 34, 38, 39, 40, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 39, 39, 40, 39, 40, 39, 40, 41, 38, 37, 38, 37, 39, 40, 37, 38, 38, 36, 3
9, 38, 38, 38, 38, 38, 39, 38, 39, 38, 38, 38, 38, 38, 39, 38, 37, 38, 38, 38, 38, 38, 38, 38, 39, 38, 38, 39, 39, 38, 38, 38, 38, 37, 38, 38, 39, 38, 38, 40, 38, 38, 38, 36, 38, 38, 37, 39, 39, 38, 38, 38, 38, 38, 38, 38, 39, 38, 39, 38, 39, 39
, 38, 38, 38, 38, 39, 39, 38, 39, 39, 38, 39, 39, 38, 38, 38, 38, 39, 39, 38, 38, 38, 38, 38, 40, 38, 38, 39, 37, 39, 38, 39, 39, 39, 38, 38, 38, 38, 36, 38, 39, 38, 38, 38, 38, 38, 38, 40, 38, 38, 38, 38, 40, 40, 39, 38, 38, 38, 38, 39, 39, 39,
 36, 38, 38, 38, 38, 37, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 39, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 38, 37, 38, 39, 38, 39, 38, 40, 39, 39, 38, 38, 38, 38, 38, 39, 38, 37, 39, 39, 38, 38, 38, 38, 38, 35, 38, 38, 38, 38, 38, 38,
```

```
>>> sim.madeIt()
[5959, 3504, 3498, 0]
>>>
```

```
>>> sim.averageTimeStep()
[38.26, 38.07, 38.01, 0]
>>>
```

Design Studio Worksheets

https://drive.google.com/drive/folders/1C_AW4k0I9wvHtVJBBMSY2m1Uln1h7D6v?usp=sharing