Traffic Cop: the serious game report

By

Tianyu Lei

Submitted to the Faculty

Of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

In

Interactive Media and Game Development

May 2016

Approved:

Prof. Lee Sheldon

Prof. Brian Moriarty

Prof. Charles Rich

Abstract

Traffic Cop is a game where the player become a traffic police officer trainee and direct the traffic to reduce the car accident rate in MA. The game is a top-down view traffic simulation and the player need to respond to the violations correctly.

The game is aimed at new drivers ages from 16-24 and the purpose is to change the drivers' attitude about safe driving. Player will learn what they should stay focused on while driving and what the consequences may be when they choose to violate the traffic signal.

Acknowledgements

Contents

List of Figures

Chapter.1 Introduction

1.1 Background

According to a recent report from Allstate Insurance Company, Massachusetts has the worst

drivers throughout America as presented in Figure 1.

Breaking: Massachusetts drivers really bad

Boston drivers were 2.58 times as likely to file a claim as other drivers, by far the worst measured.

City	Accident likelihood relative to national average
Boston, MA	**2.58**
Worcester, MA	**2.21**
Baltimore, MD	2.14
Washington, DC	2.06
Springfield, MA	**1.93**
Providence, RI	1.87
Glendale, CA	1.79
San Francisco, CA	1.65
Philadelphia, PA	1.64
Los Angeles, CA	1.63

SOURCE: Allstate

Figure 1: Report from Allstate Insurance Company

Among 200 biggest cities in America, Boston ranked No.1 in 2015 for drivers to be more than

two-and-a-half times likely to file a car insurance claim than an average driver in the United

States. Worcester ranked No.2 in this list and Springfield was No.5, which means when thinking

about the worst drivers throughout the country, 3 cities from Massachusetts was on the worst 5

list.

This was not the issue that just appeared this year. In the same report for 2014, Worcester was

the worst city on the list, while Boston followed as the second and Springfield as the fourth.

Driving safety has become a big issue that cannot be ignored in MA.

1

What caused so many car accidents? According to a list of top 25 causes of car accidents, distracted drivers ranked No.1 on the list, followed by speeding, drunk driving, and reckless driving. All these 4 reasons that cause car accident are something that the drivers know what they are doing is wrong, but they are going to do it anyway. In other words, less caution of the drivers on the road causes car crashes.

In this case, a team of four members, which includes me, Daniel Acito, Yingying Chen, and Caitlin Malone was intended to develop a serious game to improve the situation. The serious game was focusing on new drivers (between the 16-24 years old) to change their attitude about safe driving. To do so, we decide to let the players of the game to not be a driver but the opposite character, a traffic cop trainee. Players can get to know the importance of safe driving from a different perspective. They can see what will happen if the drivers' caution drops and realize the consequences of ignoring the rules when they fail to stop the violations.

The project continued through part of the background research, need and task analysis, basic game design to a paper prototype. From the point on, I start to work alone.
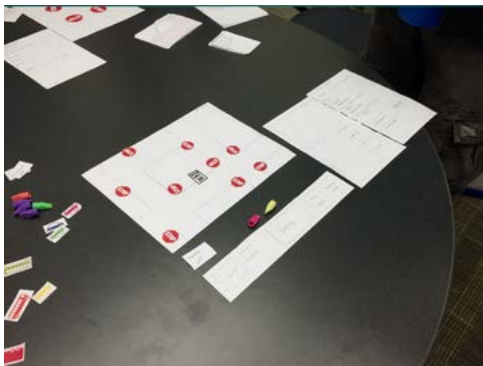
Figure 2: Paper Prototype picture 1

Figure 3: Paper Prototype picture 2

1.2 Related games

1.2.1 Driver's Ed Game

This game has two different version, one is simply using a background of driving, but the content is nothing more than a questionnaire.

Figure 4: Questionnaire type game

The second version is a top down view, and the player can use arrow keys to control their cars to fulfill certain tasks. If the player did anything wrong, a guide character will pop out and tell them what they should do. Player won't get to the next level until he did everything right.

Figure 5: Car control game

3

Many other driving rule games a similar to this game, players control their vehicle and follow the instruction of the game. However it is too passive in a state that may have the worst drivers in America. The drivers in Massachusetts should not only know the rules, but also be able to notice quickly if there is a violation nearby.

1.2.2 Papers please

Papers please is a game that have no relationship to driving rules, but it's interface could be a valuable reference if I want to build a game and let the players play as the police officer.

Figure 6: Paper Please game interface

However instead of putting all information into one screen, I feel I can separate them up into difference interfaces, so the players will have better views of the road when they are trying to find out the violation. After that, they can pull over the car and a new interface will pop out and they can decide what to do there.

1.3 Target user

The target user of this game are new drivers whose ages are between16-24.

According to an report from CDC (Centers for Disease Control and Prevention),

> *"Young people ages 15-24 represent only 14% of the U.S. population. However, they account for 30% ($19 billion) of the total costs of motor vehicle injuries among males and 28% ($7 billion) of the total costs of motor vehicle injuries among females."*

Drivers in this age group are more likely to make mistakes than others, a development of good habits could benefit them for decades in their future lives while bringing safety to other drivers on the road.

1.4 Need and Task Analysis

1.4.1 Need

Problem Statement

- There are many fundamental rules of driving in city communities that are commonly not followed

- This includes obeying traffic signs, lines, etc.

- Many people are aware of the basic the rules of the road, but choose to ignore them.

- People tend to be distracted while driving (by phone or chat) even they know it may be dangerous.

Learning Objectives

- Being fully cognizant of the rules of a given road

    o Observing street signs, lights and lines on the highway

- Recognize the consequences of bad driving habits

    o Not using turn signals, or ignoring speed limits

- Always stay cautious and observing the surroundings.

- Recognize the violations of other drivers and stay out of trouble

1.4.2 Task

Objectives

- To remind drivers, both old and new, of rules of driving that are not commonly followed or are easy to forget

- To promote good driving habits

- To encourage people to drop bad habits that they may have developed or are in the process of developing

Steps to take

- Show players different driving situations

- Have players respond to the situation by choosing their next course of action

- If user responds incorrectly, show them the consequence of their action

- Let the user try again, and repeat until they choose the most appropriate action

Chapter 2: Designs

2.1 Game Overview

Traffic Cop is a 3D top-down view traffic simulation game. In this game players play as a traffic police officer trainee who's main object is to direct the traffic and keep the number of car accidents under a certain level. The method to control the game is simple point and click, players can pullover a car, talk to a driver or give a ticket easily.

The game is developed on Unreal Engine 4, and is basically designed for Windows, but it can also be IOS or other systems that UE4 supports.

2.2 Story

In the game of Traffic Cop, players will mainly play as a traffic police officer trainee. They are going to get through different situations and witness all kinds of violation. During the game, players will get to know when a violation may occur and what consequences it may cause. They are also going to know the importance of driving safely and cautiously. At the end of the game, the trainee becomes a well-qualified traffic police officer and the players become drivers with better awareness.

2.3 Mechanics and UI Design

2.3.1 The top-down view

In order to let the players have a better view of the traffic and get the traffic information as much as possible, a top-down view was introduced and a game HUD was designed for this point of view.

The game HUD was mainly inspired by RTS games like StarCraft 2 and Red Alert 3. In these games, players need to constantly process large amount of information during the game play, which is similar to Traffic Cop.
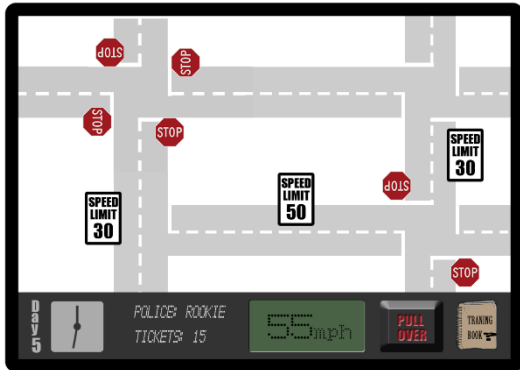
Figure 7: Traffic Cop UI Design

Figure 8: StarCraft 2 Interface

After the game was built in UE4 the UI can be visualized as Figure 9.
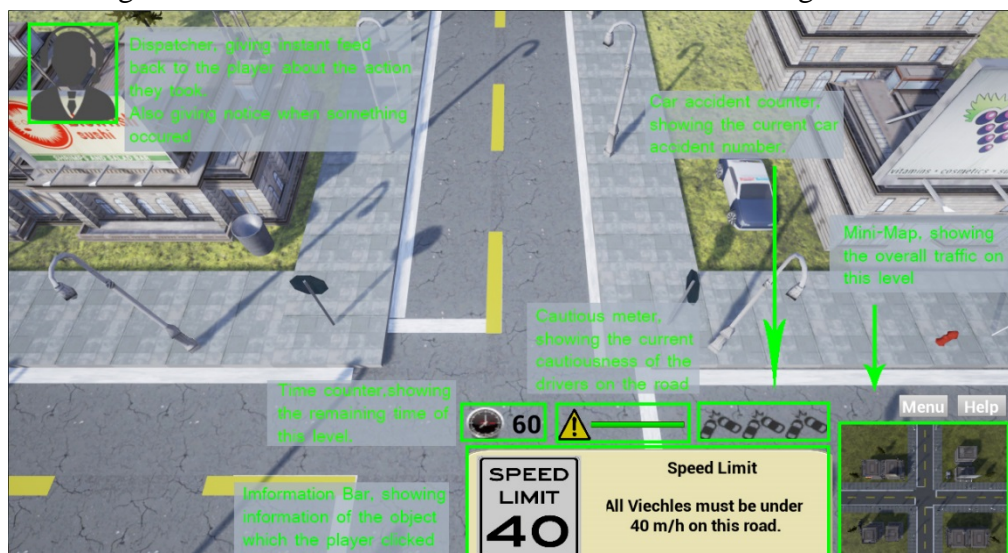
Figure 9: Traffic Cop actual interface

There are many moves the players can do under this interface, they can use mouse to maneuver the viewport, and click on any game related object to see its information at the information bar. When the player clicks on a viechle, the information bar will show the speed of the car and there will be a "Stop" button which can link to the pullover interface.

2.3.2 Pullover and ticket interface

When the players choose to stop a car, the game will pause and come to the pullover interface. This interface is designed for the players to gather further information from the driver they pullover and decide what to do with them. The pullover interface included the ticket interface on the original design.

Figure 10: Pullover interface design

Figure 11: Papers Please interface

This design is mainly inspired by the game Paper Please, which has an effective interface for the player to make decisions from multiple information.

But in the actual game build, the pullover interface and ticket interface were separated because not every driver needed to be given a ticket.

The pullover interface in game can be seen in Figure 12.
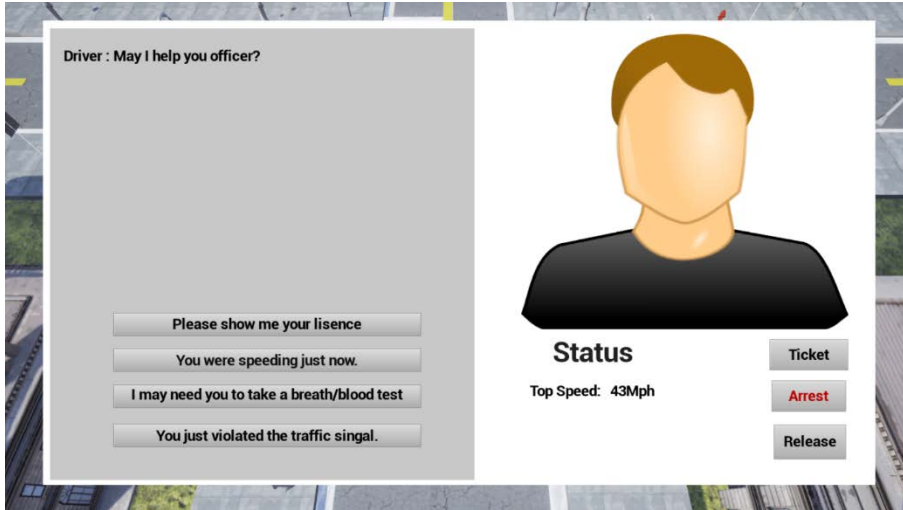
Figure 12: Pullover actual interface

The player can use the buttons on the left to have conversations with the driver. These conversation will be generated randomly, but the player will get the information of the driver after all. The player can also choose what to do with the driver. Clicking on the Ticket button will jump to the ticket interface. Clicking on the Arrest button the driver will be arrested and the car will be deleted from the game, while the player gain/loss point accordingly. Clicking Release button will set the driver free.
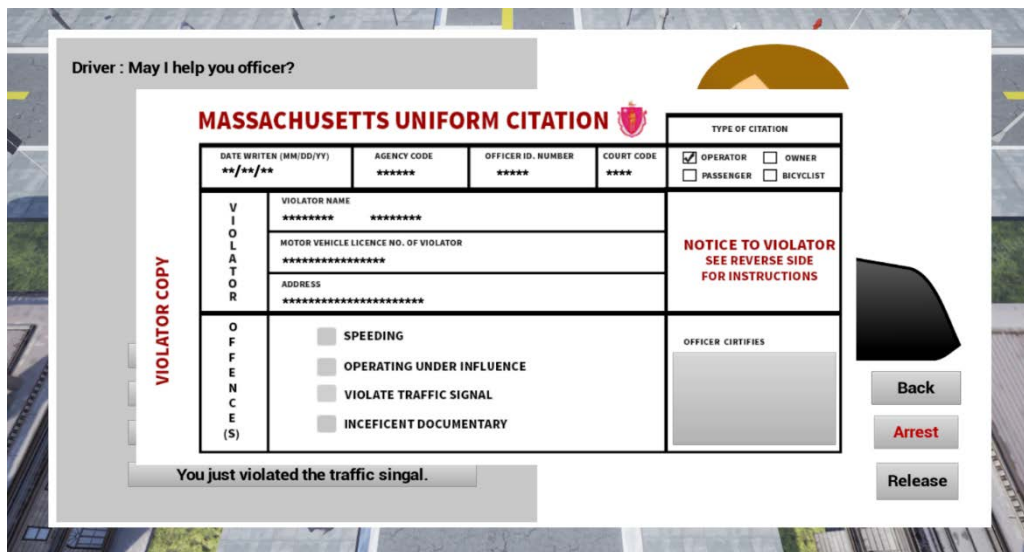
Figure 13: Ticket interface

Meanwhile, the Ticket interface as seen in Figure 13 allows the player to click on the check box to identify the violations that the driver has done and click on the signature button to certify the ticket. The ticket can only be certified with at least one violation checked. The player can also click the Back button to go back to the pullover interface.

The Citation Ticket as seen in Figure 14 is designed according to the Massachusetts Uniform Citation because the target user is mainly focused on the drivers in MA.

Figure 14: MA Citation Form

2.3.3 Information Bar

Information bar is the place where the player can view their interested information. It is a combination of the help book and speed meter in the original design.

Figure 15: Help book design

Figure 16: In game Traffic Signs

When the player clicks on the traffic sign in the game scene, the information of the traffic sign will be shown immediately on the information bar as seen in Figure 16.

11

This design is because it is better for the player to have the information instantly in the game instead of breaking the experience and searching for the answer in a book. Also looking for and clicking on a traffic sign for multiple times will help them to link the shape of the sign to what the sign is standing for, which will be helpful in their actual driving in the future.

2.3.4 Dispatcher

Dispatcher is character that will keep giving the player an update of the traffic situation. It will notify the player once a violation has appeared and will also give an instant feedback when the player choose to give a ticket.

It is important in a serious game to have a quick feedback once the player makes an action. No matter what they did is right or wrong, the instant feedback will help them to have a better understanding of what they should do and what they shouldn't do in game. This may soon transfer to their real life and guide their driving habits into the right way.

2.3.5 Mini-map

Mini-map is a tool to help the player to have an overview of the game level, since the traffic zone is too big to fit into a viewport. The player can use the minimap to monitor the area that is not on their main screen, and find out the violations in his or her blind spots.

2.3.6 Cautious Meter

Cautious Meter is a color meter that shows the current cautiousness of the drivers in the game area. The meter will rise or fall according to the actions the player takes. If the player is doing good, punishing the offenders correctly, the drivers will stay cautious because they know if they did anything wrong, the traffic police officer will know. However, if the player gives wrong tickets or ignores violations, drivers on the road will drop their cautiousness because they may not get caught even if they did something wrong.

When the cautious meter drops to a certain point, a new violating behavior will appear. The lower the meter goes, the worse the drivers' behaviors will be. However, if a car accident happens, the cautious meter will rise back to a certain level because the drivers finally realize the safety on the road is something related to their lives.

2.3.7 Car Accident Counter

The car accident counter is the place where the player can see how many car accidents have already happened in their jurisdiction. As a trainee, the player needs to keep the number of car accidents under 3 to finish the training and become a formal traffic police officer. The counter will show 3 grey car crash icons at the beginning. With every car accident, one icon will turn into red. Once all 3 icons become red, the game will be over and the player can choose to retry or not.

2.3.8 The Compliment Letter

Once the player have complete a game level, they will receive a compliment letter from MA Police Department for successfully reducing the car accident rate. This is the interface to summarize the gameplay in the level and will show the players how they did. The design of the letter is based on the real MA State Police letter style.
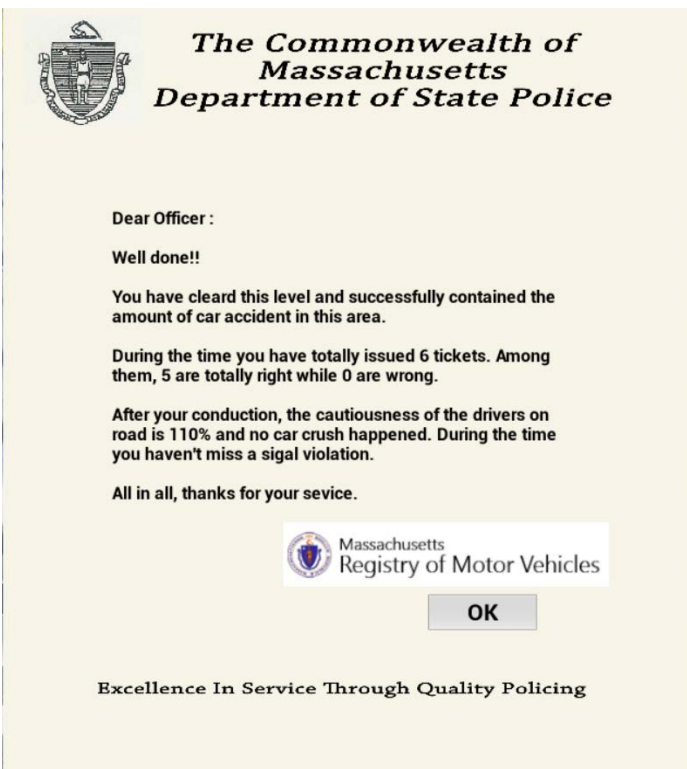
Figure 18: Compliment Letter
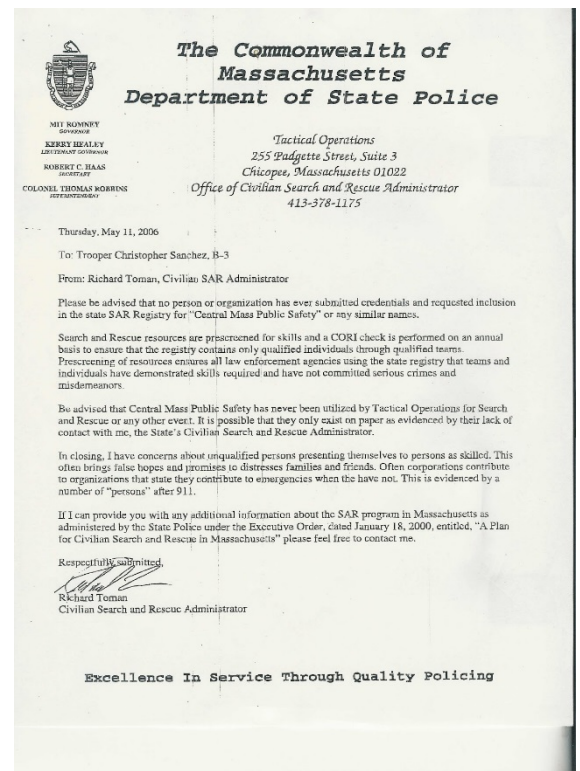
Figure 17: MA State Police letter

2.4 Level Design

Since the purpose of the game is to change the attitude of the drivers about safety driving by showing them the consequences of violating the rules, it is better to show them the violations which is more likely to cause a bad result.

Also, in the real world, there is always a grey area between violating a traffic rule or not. For example, in the real world, driving over the speed limit by one mph or even five, won't get a driver a ticket, but in the game world, even one mph over the limit will be considered as speeding. This is mainly because the game is trying to persuade the drivers to driving safely. If the game conveys messages such as "it is ok to speed just a little", it may influence the driver's future actions, which is against the purpose of the game.

2.4.1 Level 1

As the beginning of the game, one of the most commonly seen traffic situation was introduced, the four-way stop intersection.
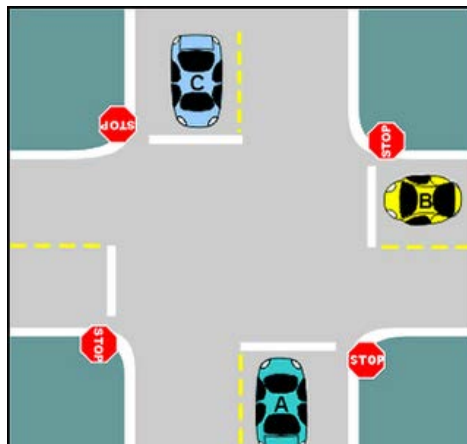
Figure 19: Four-way Stop Intersection

And the level set up is shown as Figure 20.

Figure 20: Level 1 Game Scene

Inside this level, four different violation may appear, and they are:

- Driving without a valid driver's license.

- Speeding (ranked no.2 on car accident causes.)

- Running stop signs (ranked no.7 on car accident causes.)

- Drunk driving (ranked no.3 on car accident causes.)

These 4 violations appear under certain circumstances:

- Speeding and driving without license appear randomly on every generated car.

- Running stop signs appear when the driver's cautious drop under 50%.

- Drunk driving appear when the driver's cautious drop to zero.

Player should respond to these violations accordingly:

- For speeding, running stop signs, driving without license, player should give them

  a citation ticket with the violation list checked.

- For drunk drivers, player should choose to arrest them immediately.

2.4.2 Level 2

In the second level of the game, the player will be asked to focus on the speeding issue

which causes a large portion of car accidents in the everyday situations.

The landscape below is a highway going through a valley with a slower route on the side.

Figure 21: Highway reference

And the actual level is shown as Figure 22.

Figure 22: Level 2 Game Scene

In this level, there are also four different violation may appear, and they are:

- Driving without a valid driver's license.

- Speeding (ranked no.2 on car accident causes.)

- Drunk driving (ranked no.3 on car accident causes.)

- Illegal car racing/reckless driving (ranked no.4 on car accident causes.)

These 4 violations appear under certain circumstances:

- Speeding and driving without license appear randomly on every generated car.

- Drunk driving appear when the driver's cautious drop under 50%.

- Illegal car racing/reckless driving appear when the driver's cautious drop to zero

Player should respond to these violations accordingly:

- For speeding, driving without license, player should give them a citation ticket with the violation list checked.

- For drunk drivers and illegal racers, player should arrest them immediately.

2.4.3 Level 3

In the third level of the game, players were told that the drunk driving and illegal racing behavior have been reduced thanks to the effort of the player. However, the newly introduced traffic light system makes drivers confused, so the players need to help out. The third level setup is an intersection with a traffic light system and is shown in Figure 23.
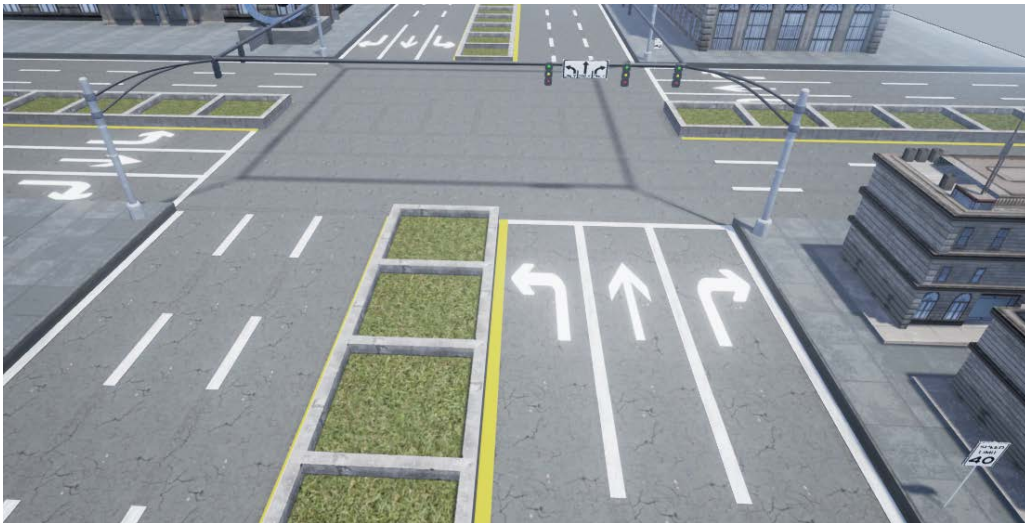

Figure 23: Level 3 Game Scene

18

In this level, four different violation may appear, and they are:

- Driving without a valid driver's license

- Speeding (ranked no.2 on car accident causes.)

- Trying to catch up a yellow light signal

- Ignoring a red light signal (ranked no.6 on car accident causes.)

These four violations appear under certain circumstances:

- Speeding and driving without license appear randomly on every generated car.

- Trying to catch up a yellow light signal appear when the driver's cautious drop under 50%.

- Ignoring a red light signal appear when the driver's cautious drop to zero

Player should respond to these violations accordingly:

- For speeding, driving without license, trying to catch up a yellow light signal, players should give them a citation ticket with the violation list checked.

- For drivers who ignore a red light signal, players should choose to arrest them immediately.

There are some difference between the game and the real world. Drivers won't get a ticket trying to catch a yellow light nor do they get arrested for rushing a red light, but the purpose of the game is to make them stay cautious when driving and that's why many rules in the game are much stricter than in real world.

Chapter 3: Implementation

The game, Traffic Cop, was developed on Unreal Engine 4 using blueprint based script. It is a node-based interface that helps people to program without using professional programming language. Traffic Cop, did not use any default template provided by UE4 and was totally build from scratch, and this section is going show how the game was constructed piece by piece.

3.1 Game Commander

Game Commander is the tool that players use to get information from the game and give command to the game. In simple terms, the game commander is the camera controlled by the player. Player can use the commander to move the in game vision around to find out what they are interested in.

There are two ways for the players to move the camera, first one is using wasd keys to maneuver it. This can be achieved easily using the AddActorLocalOffset node. Things need to take care is to set the moving speed and check if the camera has reached the map boundary.

Figure 24: AddActorLocalOffset Node

The second way to move the camera is a bit more complicated. When the player moves the mouse to the edge of the screen, the camera will automatically move towards that direction. This action requires a hitting edge check. To do so, the player needs to get the viewport size and the mouse position first, and calculate the two group of data in percentages, both in X and Y. If the percentage is below of above a certain point, then it should be considered

20

as the mouse has hit the edge of the screen. The point was set to below 2% or over 98% in game as shown in the picture.
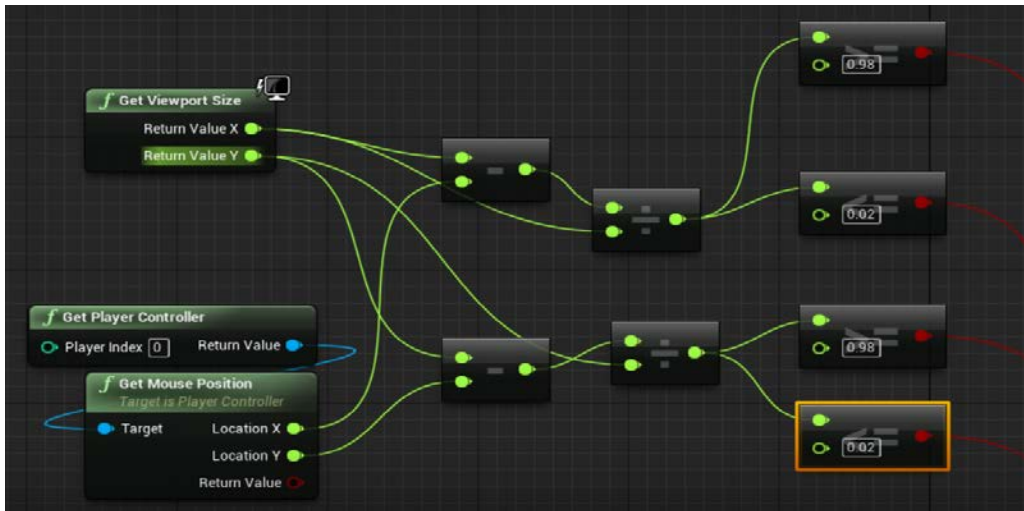
Also, there are two ways to set the map boundary to make sure the camera won't move outside the game level. One is to use the blocking volume to build a wall as the physical map boundary and stop the camera from moving out. Another is to set the map boundary as variables and keep checking the camera's current location. In this case, the camera will stop moving once it's in a certain place (which is the map edge.)

 The commander also contains an Object Selection function, which means once player clicked on a game object, this function will find out the class of the object and show the information of the object on the screen. The game uses a cursor hit result node to get the object reference as shown in the picture. Meanwhile, Cast TO Node, are used to classify the object. (Cast failed means the object doesn't belong to the class of object currently casting towards.)
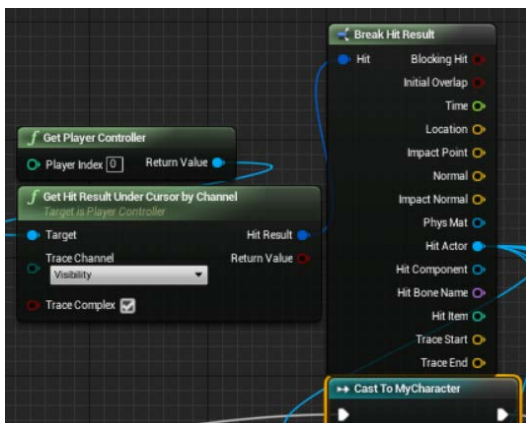
Figure 26: Clicking Object Reference Capture

21

3.2 Game Widget

From game HUD to loading screen, most of the game interfaces that the players see was constructed using the Widget Blueprint. Although it is a basic function in UE4, there are still a few things need to be emphasized.
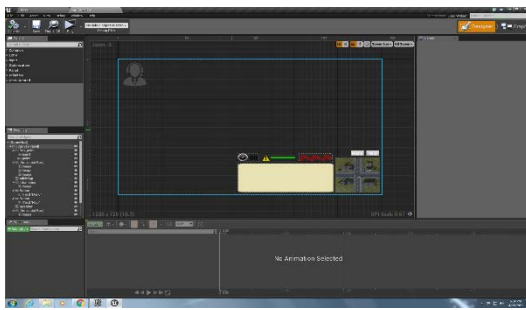
Figure 29: Game HUD widget
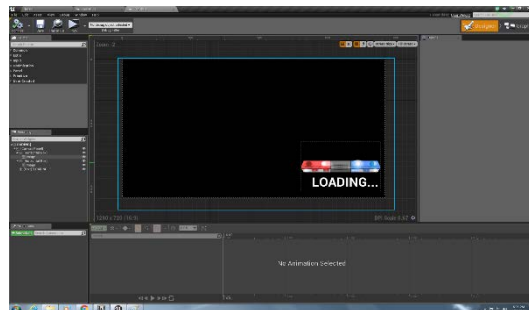
Figure 28: Loading widget

3.2.1 Anchors

Anchor is the point for the content of widget to lock on to. When the game was packaged and ran under a different resolution, the contents will be relocated according to the anchors set for them. It is important for all pieces of the same content to have the same anchor or there may be a misallocation in the actual gameplay, the result may be text were shown outside the textbox or icons appeared in a wrong place. However different content in the same widget should also have different anchors. For example, as the contents in game

HUD, the dispatcher set its anchor to the upper left corner while rest contents were set to the lower right corner.

In this case, even the resolution of the game changed, they will still be separated into different corners during the gameplay and functioning just as the same.

3.2.2 Printing Messages

In some occasions, the in game text should not be shown all at once, but needs to appear character by character, as if someone is typing the messages to the player. This will make the player feel less suddenness and have a better reading experience. However, UE4 doesn't contain such a printout function in the game engine, and to achieve the typing effect becomes a little tricky.

One method to do that in UE4 is first set the get the character array from the text string, Then adding set an integer variable to count the current printing character.

To print the whole text, first get the first character, print it out, set the integer +1, then

delay for a short moment (0.15 second in this game). After that begin the next circle, get

the next character, add it to the text string that had already been printed using an append

node, and print the text again and delay another 0.15 sec. The circle continues till the whole

text was print out, which means the integer number is the same as the character amount of

the text string. (-1 if the integer started count from 0)

The whole process is shown as the *Figure 31*.

 By the way the append node can also be used to start a new paragraph since the UE4

doesn't provide a way to wrap the text except the Auto Wrap Text function.
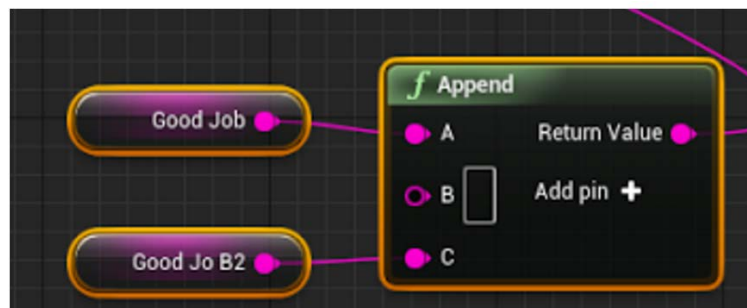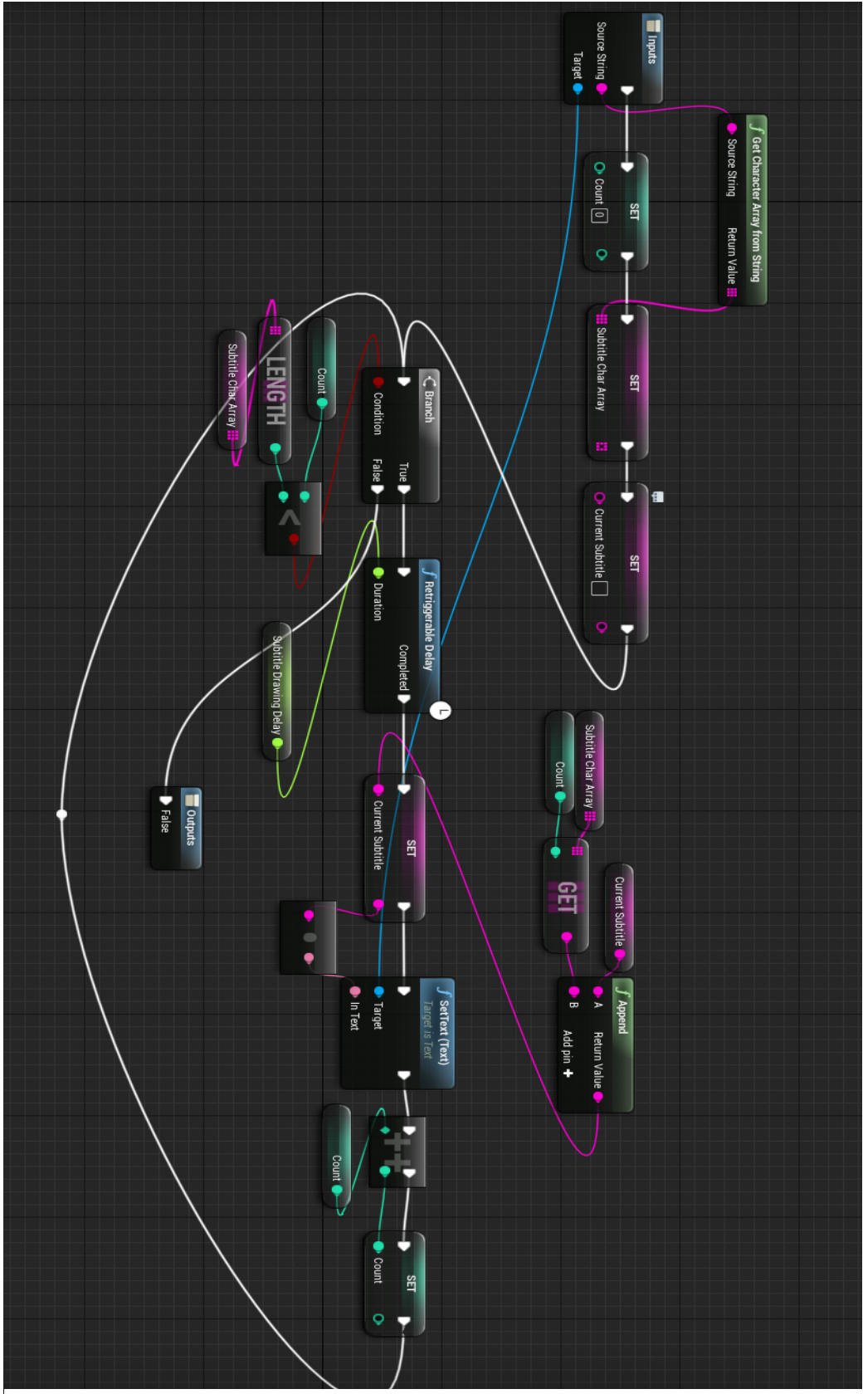

Figure 31: Append node for wrap text

24

Figure 32: Print String Function

3.3 Messages between Blueprints

For linking the whole game together and making it playable, the information transmission between blueprints is very important. These messages could be the player's point and click action data from interface to an actor, and can also be the effect from one actor to another. Usually there are three different methods for these data transmission, and they will be introduced one by on in this section.

3.3.1 Cast To

Cast To is a method to classify the object as mentioned before, but its function is much more than that. In fact, delivering message between blueprints is the main function of this node. Using, Cast To node, can help the current blueprint to get easy access to all the variables and functions that belong to another blueprint. However, one drawback for this node is that it needs a precise reference of the blueprint that is going to be casted to and sometimes it is just hard to get it.

One way to solve this is use make both blueprints cast to the same third blueprint of which the reference is easy to get. For example, Game Instance is such a blueprint used to save variables exclusively, by using the Cast To node, it can even transmit variables through different game levels.
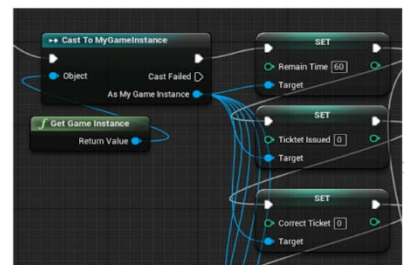
Figure 33: Usage of Game Instance

However, this method can only transfer variables between blueprints but cannot call out the function which belongs to another blueprint that cannot be referenced. This is where the second method is useful.

3.3.2 Function Override

This method used another node called Game Interface, which is especially designed for the function override. In the game this method was used for calling the object select function from the object clicked and running the function on Game Commander.

To achieve the transmission there are totally three steps.

First, create a function called Object Selection in the game interface.

Figure 34: Object Selection in Game Interface

Next, call the function and send the message to the Game Commander.
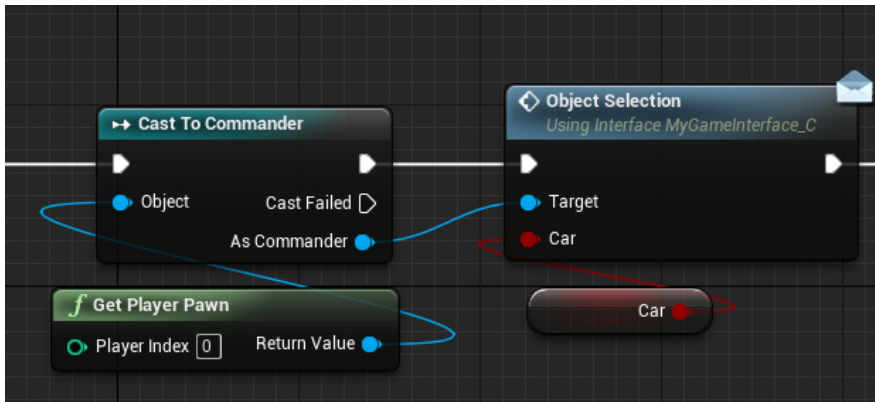

Figure 35: Message sending to Game Commander

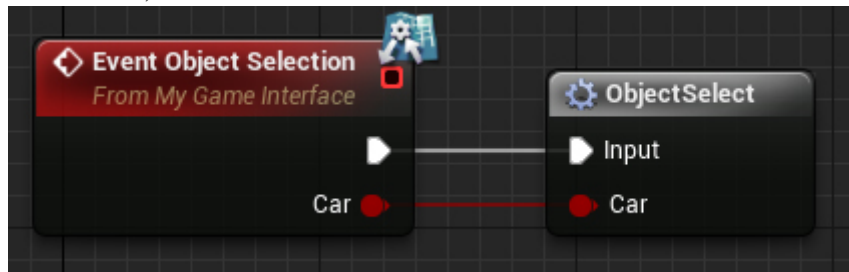And at last, override the function from the Game Commander.

Figure 36: Override Function in Game Commander

However the override can only be used once, one function was override by multiple different blueprints, errors may occur.

3.3.3 Event Dispatcher

Event dispatcher is another way to call out functions from one blueprint to another blueprint.

The strength for this method is that it doesn't require the reference to the blueprint which the function is belonged to, this is really convenient when a function in the level blueprint need to be called since the level blueprint cannot be referenced.

Another blueprint was introduced in this method which is the Game Mode.

To achieve this, there are also three steps.

First, create the Event Dispatcher in the Game Mode.

Figure 37: Event Dispatcher
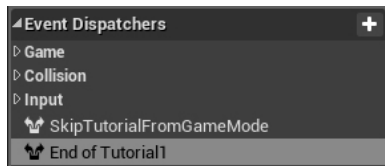
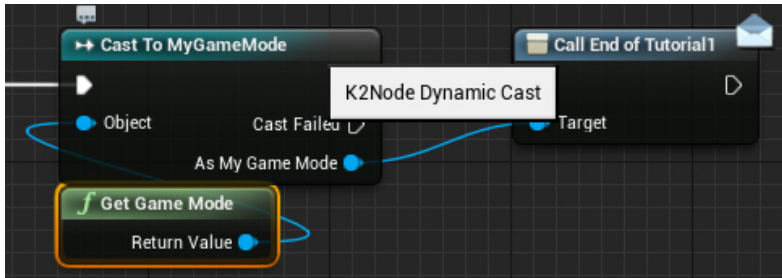Next, call the function from the widget blueprint.

Figure 38: Sending message from blueprint

Finally, bind another Custom Event to this Event Dispatcher in the level blueprint.
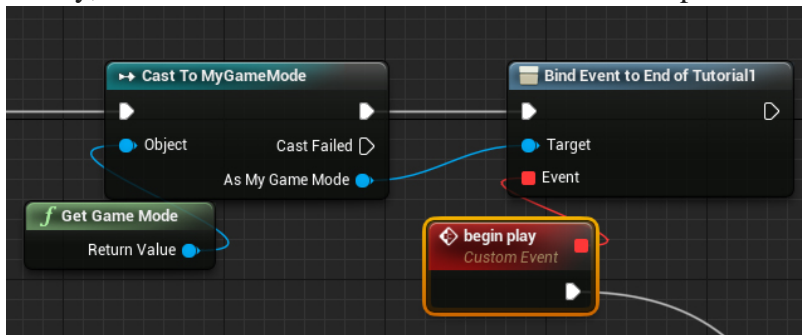

Figure 39: Binding event to the dispatcher

Three different method all have their different strength and weakness, by combining all three of them, the game data transmission can be achieved easily.

28

3.4 AI and Traffic Simulation

In order to make the game possible to serve the original purpose, to show the player what they need to take care of on the road and what the consequence may be if they choose to violate the rules, traffic simulation in one of the most important part in the whole game. Because the traffic need to be self-running and need to react to both situations on the road and different behaviors of the players, a rather complicated AI system is needed.

In this game, the AI behavior tree system was used and all the variables are all saved inside a blackboard data blueprint.

To show the whole AI behavior tree is very difficult, since it's a huge amount of calculation as shown in the figure. It was a part of the AI where a car needs to decide whether to turn or go straight at one direct of the intersection, not to mention there are several direction, multiple type of cars and three levels.
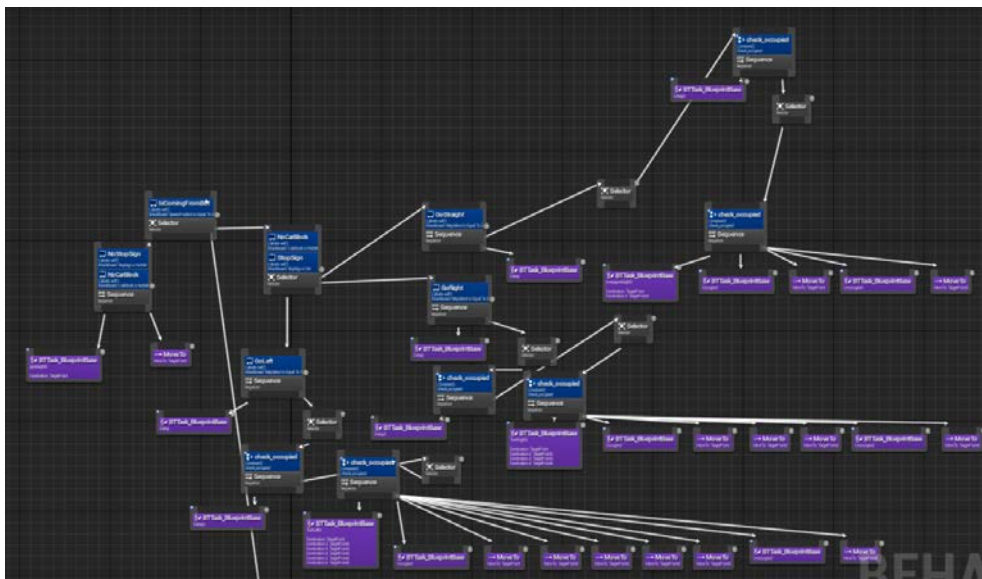

Figure 40: Partial AI behavior tree

However, even the AI behavior tree have so many contents, the basic method is almost the same, and it will be explained in this section.

3.4.1 Behavior Tree Construction

First of all, there are three basic components in an AI behavior tree that can be used:

- Selector - When the actor needs to change the behavior such as from going straight to turning a corner.

- Sequence - When the actor needs to perform a series of actions.

- Simple Parallel - When the actor needs to take two actions simultaneously, like moving and aiming in a shooter game.

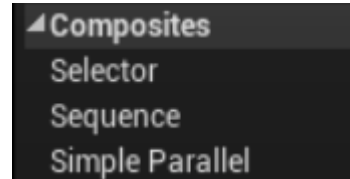*Figure 41: Structure Component*

Use this three components to build the contracture of the behavior trees and add in the task blueprints. Then use decorators to identify what tasks should be taken under which circumstances. The checking value of decorators can be variables from blackboard or other blueprints like the Game Instance as shown in the figure.
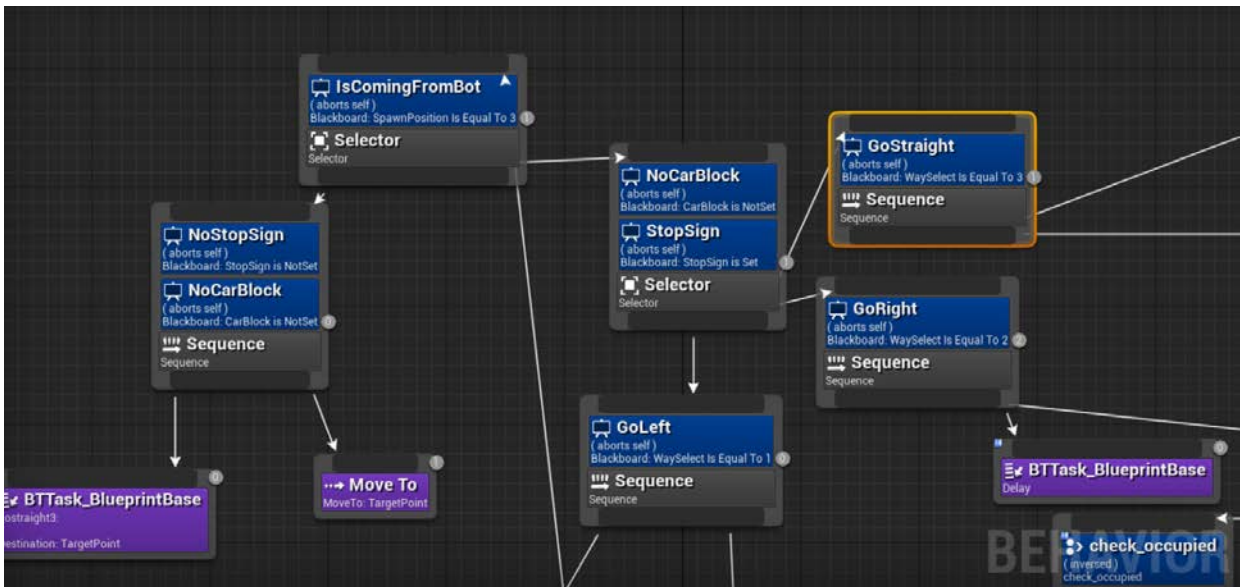

Figure 42: Decorators in behavior tree

In this case, the in game event only needs to change the value of a single component, the behavior of the actor will changed as programed. The figure shows how the behavior of a car was changed when it was blocked by another car and when it reached a stop sign.
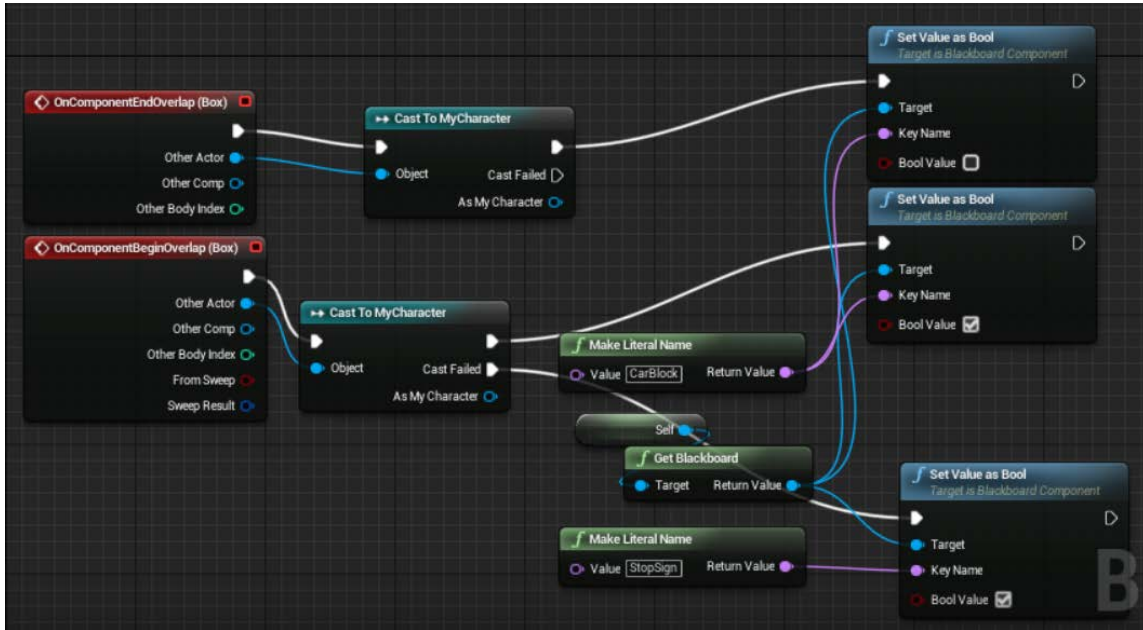
Figure 43: Change value to change behavior

3.4.2 Move the Actor

Another thing important to simulate the traffic system is to make the actors move inside the game level. In the game Traffic Cop, level one and two used the same moving technic, while level three introduced a new moving method.

The way to move the actor in the first two levels is using the moving to function provided by the UE4, which is setting a target point for the actor, and it will move to that point straight forwardly. However, in order to let the car to make a turn, there needs to be a set of multiple points in a curve and ask the actor to move to the point one by one. This not only takes a long time to set up, but it also makes the action of turning kind of choppy and stiff.

31

Figure 44: Simple Move To Function

To improve the situation, a new method was introduced. Thanks to the iTween plug in for UE4, actors now can move following a spline line now, which makes the turning action to be more natural and much smoother. Yet this new method also have a weakness, which is that the action is similar to an animation, so once it was started, the actor can't change its behavior until the whole action is finished. Although this new method is still not perfect, it is still very promising for the future.
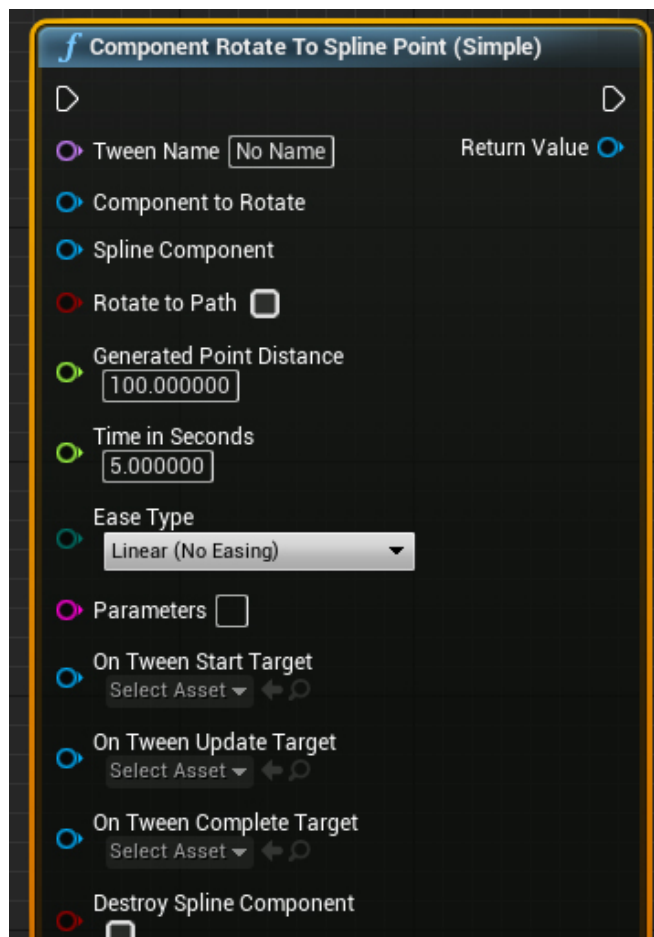
Figure 45: Move to Spline function

3.5 Score Calculation

In order to give feedback to the player about how well they did in game, a functional score calculation system is need. Basically there are two places needed to be checked in the game, one is when deleting a car actor once it has finished all actions and reach the end of the path, and another one is when player is making a ticket.

The first one is relatively easier, the system will check if the actor have already been given a ticket first. If so, nothing need to be done. However, if the actor didn't get any ticket form the player, the system will check if the violation related Boolean of the actor is set to true. Every violation of the actor means a minus point for the player, and the layout of the blueprint is shown as the figure.
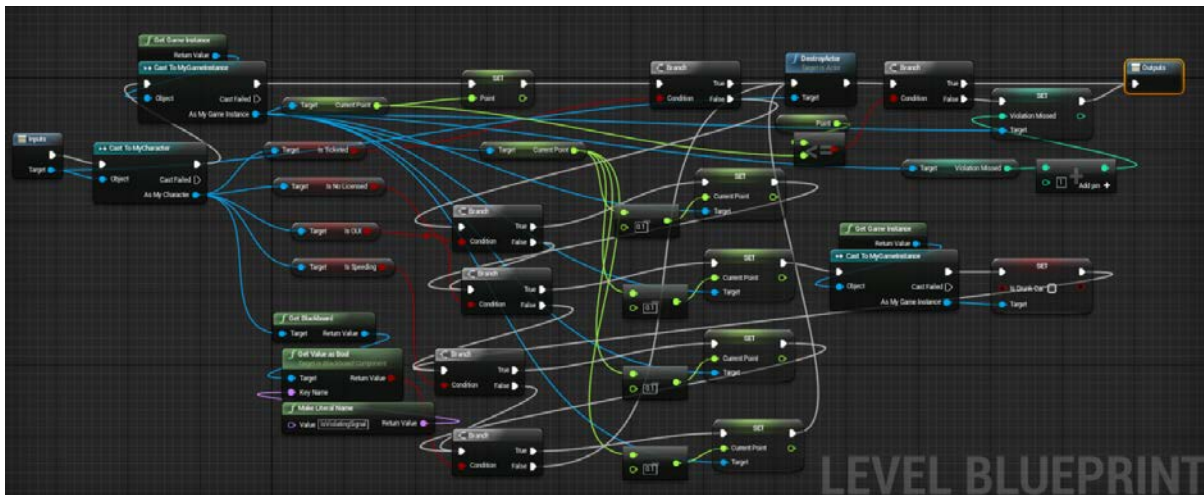
Figure 46: Actor violation checking

However, to check whether the player made every ticket right is a bit more complicated. First, the system will check if the player selected one violation when giving the ticket. If yes, the check if the actor did violated that rule. If the actor also violated the rule, positive point will be given, otherwise there will be a minus point. If the player did not select the violation but the actor did violated the rule, minus point will also be given. However, if the player did not select the violation and the actor did not violate the rule neither, the system will do nothing and start to check next possible violation. It seems to be only four possibilities, but with multiple possible violations, the chart can me very messy as shown in Figure 46.
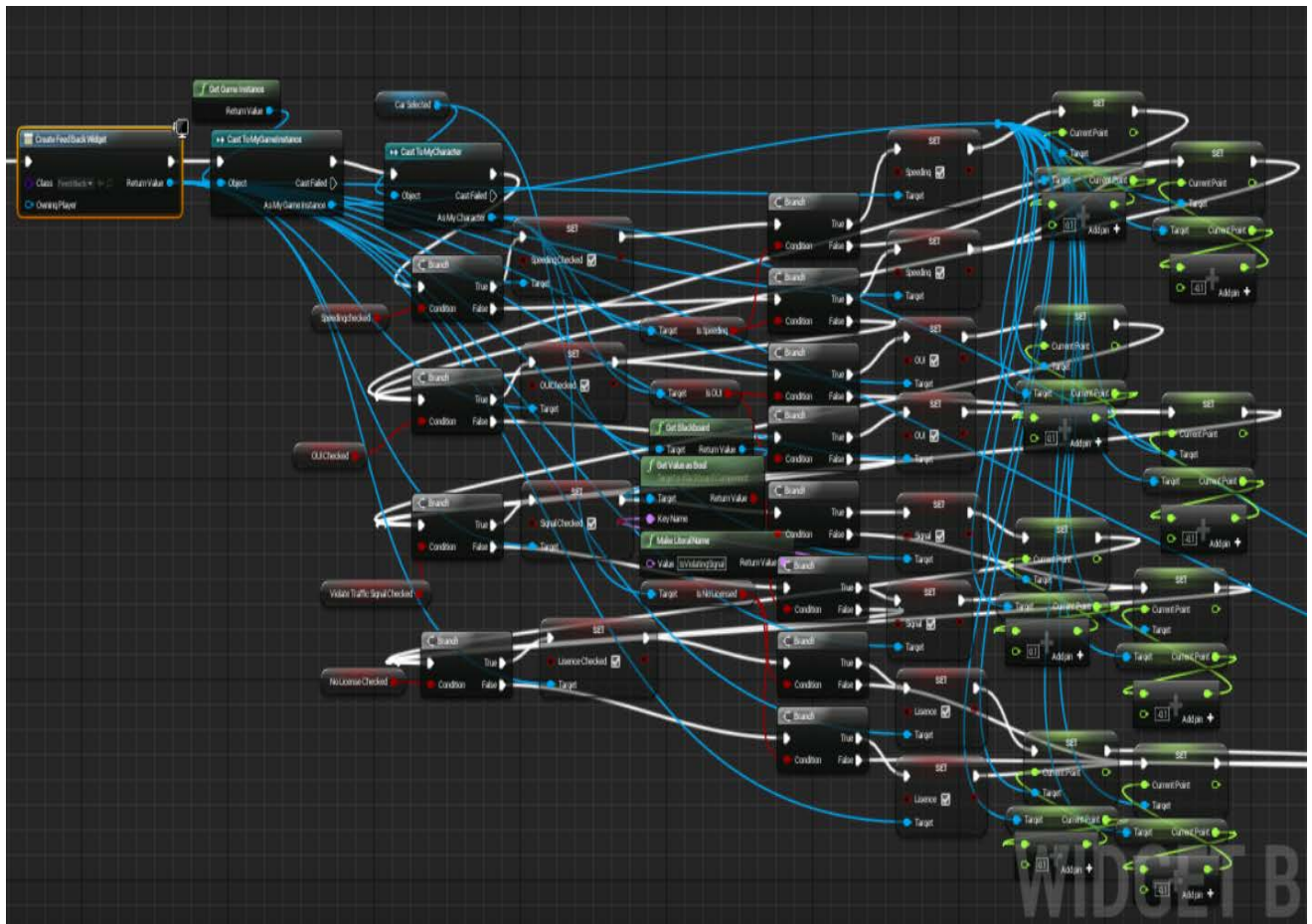
Figure 47: Ticket Checking

Chapter 4: Conclusion

4.1 Playtest

4.1.1 Iterative playtest

During the development of the game, I asked several random people in the lab to playtest it and watched how they do it.

The purpose of the playtest is to check if the gameplay is fun and if there is any defect in design or bugs in game. Many valuable were received during the process and the behaviors of the play testers are very interesting.

There are players trying to beat the game by following the instructions and direct the traffic seriously. Some other players trying the messing the game to find out what trouble they can cause. There are also players who choose to do nothing but just watch the traffic come and go.

But overall, the gameplay was enjoyed and considered interesting.

4.1.2 Targeting test

The project is not able to get through this part due to the scale of the project and the lack pf manpower. However, the test method of the game will still be briefly introduced in this section.

To test if the game actually have the ability to change that attitude of the target user about safe driving, there are four steps.

First is the Pre-Game Test, which is use gather information of the test subjects. Usually the test method is a questionnaire. The information need to gather includes the attitude of the testers about safe driving, their driving record and have they played other games that was developed for safe driving.

Next, in game observation is needed. The behaviors of the test subjects during the game can give valuable information. Do they get touched when they see a car crash? Do they enjoy playing the game? These are all the messages that can help the game to become better.

After the gameplay, there should be a Post-Game Test, which basically have the same contents as the Pre-Game Test. The purpose of this test is to find out if the game really changed how the subjects think about safe driving and how the game worked.

Last but not least, the follow up feedback is important. Because the purpose of the game is to actually change the behaviors of the drivers, it is matter to know how they really behave in driving. Is the messages form the game transferred into their driving habits? How long can the effect of the game last?

Basically this is how the game should be tested.

4.2 Postmortem

4.2.1 What Went Right

Building a game from scratch is not easy, especially for someone working alone, but at the end of this project, a fully functional game was made and it is a major success.

The project management is the one to take the credit, I'd say. Even though my project started quite late compare to other groups, have a schedule and keep working on it every day finally made the completion of the project possible.

4.2.2 What Went Wrong

Spelling error and grammar problems are all over the game, which is mainly because I'm not a native speaker, but if I have more time or a teammate who can speak English well, this problem should have been easily solved.

The game doesn't have a very good game visual and could have better sound effect and BGM. This is also because of the lack of manpower.

Another thing is that the methods to simulate the traffic system is still not perfect, both methods have their weaknesses, it still takes time to fix them.

4.2.3 What I Would Do Differently

First, find a teammate. It feels really bad when sometimes you want to achieve something yet you know it can't be done alone. If I could find a teammate who is good at the visual design, the game could looked totally different.

Second, start earlier, so I would have more time to achieve some features that is already so close to be completed but have to be dropped due to the insufficient time.

4.3 Overall summary

All in all, Traffic Cop is a serious game and its purpose is to change the attitude of new drivers about safe driving.

The game provided a new perspective for the players to witness and experience the importance of driving safely. They can know what they need to be careful with while driving and learn the consequence of disobeying the rules on the road.

The game is still under development, there are bugs need to be fixed and new features need to be added in, but the project is promising and will be continuously built after this report.

Reference

[1] Jack Newsham, 2015, Boston drivers are worst in the US, and Allstate has proof

https://www.bostonglobe.com/business/2015/09/03/boston-drivers-are-worst-data-shows/8XtYYLZof95SvSckwJZt1I/story.html

[2] CBS Boston, 2014, Worcester, Boston Drivers Worst In Nation, New Study Says

http://boston.cbslocal.com/2014/08/26/worcester-boston-drivers-worst-in-nation-new-study-says/

[3] Jim Gorzelany, 2014, Cities With The Best -- And Worst -- Drivers

http://www.forbes.com/sites/jimgorzelany/2014/08/27/u-s-cities-with-the-best-and-worst-drivers/

[4] Allstate, 2014, ABD Report

https://www.allstate.com/resources/allstate/attachments/tools-and-resources/abd-report-2014.pdf

[5] Allstate, 2015, ABD Report

https://www.allstate.com/resources/allstate/attachments/tools-and-resources/abd-report-2015.pdf

[6] Top 25 Causes of Car Accidents

https://seriousaccidents.com/legal-advice/top-causes-of-car-accidents/

[7] Car Accident Attorney, 2013, Top 10 Causes of Car Accidents

http://www.losangelespersonalinjurylawyers.co/top-10-causes-of-car-accidents/

[8] CDC, Teen Drivers: Get the Facts

http://www.cdc.gov/motorvehiclesafety/teen_drivers/teendrivers_factsheet.html

[9] Dow Chang, 2008, Comparison of Crash Fatalities by Sex and Age Group, DOT HS 810 853

http://www-nrd.nhtsa.dot.gov/Pubs/810853.pdf3

[10] Car Crash Statistics Based on Age and Location

http://www.autos.com/driving-and-safety/car-crash-statistics-based-on-age-and-location

[11] Driver's Ed game
https://driversed.com/games/dmv-game.aspx

[12] Driver's Ed game 2

http://www.agame.com/game/drivers-ed-2

[13] Papers Please

http://papersplea.se/

[14] StarCraft 2

http://us.battle.net/sc2/en/