DRIVE2PASS: A 3D GAME-BASED INTERACTIVE DRIVING SIMULATION SYSTEM FOR IMPROVING THE YOUTH DRIVING LEARNING AND TRAINING USING MACHINE LEARNING

Charles Yang[1] and Yu Sun[2]

[1]Northwood High, 4515 Portola Pkwy, Irvine CA 92620
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Youths have a higher car accident rate, so to decrease the percentage, I developed a game that will teach players to practice safer driving behaviors [5]. It is meant to simulate real driving, and teaches the players key individual concepts about road safety. This game puts an emphasis on properly executing blinking, hill parking, and headlights. This addresses the problem in other games where they solely focus on steering and acceleration, as this game also includes other driving elements to promote defensive driving. The intended goal of this game was to teach beginner drivers proper driving etiquette in a safe, risk-free environment and become a potential alternative to the traditional method of driving on real roads.

KEYWORDS

Youth Driving, Machine Learning, 3D Modeling.

1. INTRODUCTION

Nowadays, youth drivers have a higher rate of getting into car accidents. According to Driver Knowledge in 2018: “6.7 million cars involved in traffic crashes”, the main factors to keep driving safe is to manual in keeping your hands on the steering wheel; visual in watching the roadway and traffic around you; and cognitive in keeping your mind on driving safely [1]. “Four common driver behaviors are reported to have caused about 70 percent of those 2016 fatalities. Those are drunk driving, not wearing a seat belt, speeding and distracted driving” [2]. According to InjuryFacts 2021 graph: Average rate of 11.9 deaths per 100,000 population “Peaked among persons age 18-25; high of 18.9 at age 21” [3]. According to Statista 2020: the age at which auto-insurance is most expensive in the U.S. is 18 years old [4]. According to Hedges Company 2019: “The lowest percentage of total licensed drivers is among 16- to 19-year-olds, where just 34.8% of the population has a driver’s license” [5].

I wanted to create a game of my own, and driving seemed like an interesting genre to me. I could develop a game while at the same time introducing a more educational component to give it a practical application, in that it can teach others how to drive. This will accordingly help in reducing injury and fatality rates in car accidents.

The problem is that youth drivers have a higher rate of getting into car accidents. Right now, there are no established solutions for this problem, so I aim to develop an application that can

David C. Wyld et al. (Eds): SOFEA, CTCM, BIBC, SIPR, NCWC, CSEN, EDTECH - 2021
pp. 213-224, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111618

promote safer driving behavior by having youths engage in a driving simulator that teaches them safe driving.

Popular video games like Grand Auto Theft 5 and Mario Kart excludes typical vehicle functions like the turn signal, headlights; also does not educate on proper driving behavior [6].

In this paper, we follow the same line of research by developing a driving simulator composed of levels that feature different driving concepts. Our goal is to teach young drivers proper driving etiquette without having to put themselves in danger on the road. Our method is inspired by a drunk driving simulator in a museum; that simulator allowed the player to experience driving under the influence of alcohol without actually having to drink alcohol [8]. As programmed in the drunk simulator, the player could not accurately control the car [7]. A game that provides a more complex, nuanced driving experience.

This game allows the player to activate the other important vehicle features like the blinker.

The rest of the paper is organized as follows: Section 2 describes the challenges that we encountered during the game development process and experimental design; Section 3 provides the details of the game, including the game structure, level design, and educational reinforcement; Section 4 presents the relevant details about the experiment we did, following by the mentioning related works in Section 5. Finally, Section 6 gives the conclusion remarks, as well as discussing the future work of this project.

2. CHALLENGES

In order to build the tracking system, a few challenges have been identified as follows.

2.1. People are discouraged from investing extraneous resources into driving

As of now, most people learn to drive via the traditional method, which is enrolling into a driving school and practicing to drive as a student driver. Consequently, few people will consider learning virtually as it is not a viable option [9]. I will make my application accessible for anyone learning to drive by making it free to download on PC and iOS. As this application acts as an alternative to the traditional method, users will not have to worry about taking a supplement course and further use up their financial and temporal resources. In addition, this application should be effective enough to the point where people can use this as a substitute to the traditional method. As this application is virtual, it lacks the feeling of realism, so the goal of this app would have to teach users how to follow proper driving etiquette without them actually driving. Thiscan be done by showing them how to follow etiquette so that the users can replicate the behavior in real life.

2.2. How to effectively educate people on appropriate driving etiquette

This application will educate people about driving concepts in a unit format, where each unit focuses on a particular driving concept. A diagnostic test will be conducted at the beginning of each unit to gauge the student’s level of familiarity with the tested concept. One way to teach people to drive safely is to communicate to people the risks of not practicing appropriate driving etiquette. This application aims to allow the students to freely experiment with different actions in the driver's seat that result in different outcomes. This way, the students can digitally experience the consequences and recognize the risks, encouraging them to practice safer driving behaviors to avoid these risks.

2.3. How to gather useful and relevant results from this application

The third challenge is that the application needs to collect data from the users that can be used to generate useful and relevant results for the study. The manner in which the data will be collected is by asking the users to fill out a survey outside of the application. This survey will ask for the user's age, gender, time spent driving, rating on the effectiveness of each unit, and the way that they heard of the app. The compiled data will be used by me to generate a data table listing all of these categories, and I will analyze and interpret the data [10]. In order to produce useful and relevant results, I need to record how many people downloaded my application and how effective the app was. I also need to see which age demographic and amount of driving experience is most prevalent in the user base as this app is meant for teenagers and young adults learning to drive.

3. SOLUTION

I developed a single-player game called Safe Drive Test. The game was developed using the Unity game engine, and the code was written using C#. It plays out as a driving sim split into levels that teach individual driving concepts. Each level starts out with the level title screen, followed by an introductory prompt that provides the player a preface to the driving concept that the level is focused on. The player then plays through a pretest that offers them a chance to implement the concept they were taught [11]. After the player completes the pretest, the player is shown a scorecard that details their results. Based on certain conditions unique to each level that the player fulfills, the points associated with the conditions are added to the total score. Therefore, from the scorecard, the player can learn what they did and did not do correctly. Regardless of the player's performance, the player is asked a multiple-choice question that tests their knowledge of the concept. The player is unable to proceed until they correctly answer the question; this was done to reinforce the player's knowledge of the tested concept. The player then plays through the practical test, which builds upon the previous test by providing a different and more complicated scenario. Finally, the player is given a final score based on their performance in the practical test, where their score determines whether or not they can progress to the next level.

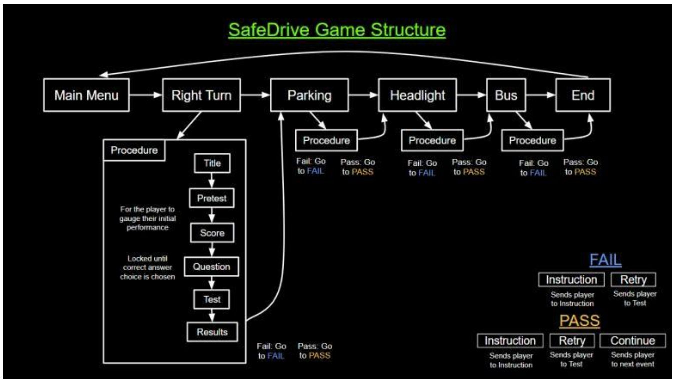
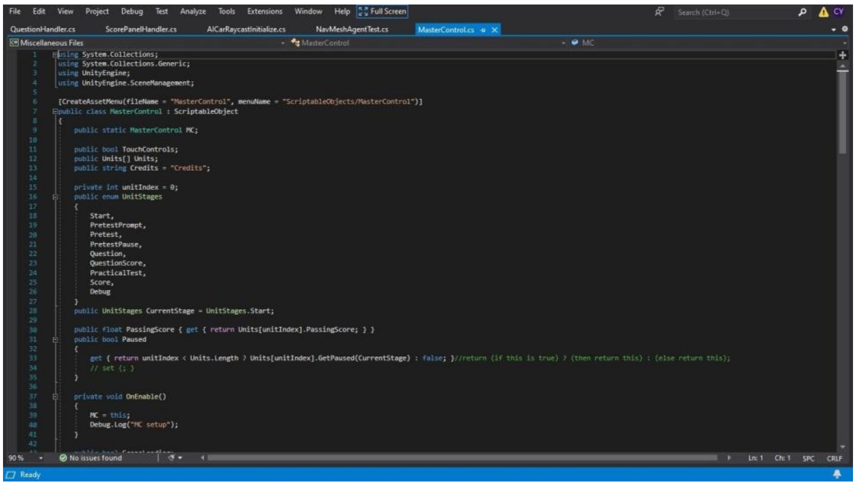


Figure 1. Overview of the game



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

[CreateAssetMenu(fileName = "MasterControl", menuName = "ScriptableObjects/MasterControl")]
public class MasterControl : ScriptableObject
{
    public static MasterControl MC;

    public bool TouchControls;
    public Units[] Units;
    public string Credits = "Credits";

    private int unitIndex = 0;
    public enum UnitStages
    {
        Start,
        PretestPrompt,
        Pretest,
        PretestPause,
        Question,
        QuestionScore,
        PracticalTest,
        Score,
        Debug
    }
    public UnitStages CurrentStage = UnitStages.Start;

    public float PassingScore { get { return Units[unitIndex].PassingScore; } }
    public bool Paused
    {
        get { return unitIndex < Units.Length ? Units[unitIndex].GetPaused(CurrentStage) : false; }//return (if this is true) ? (then return this) : (else return this);
        // set {; }
    }

    private void OnEnable()
    {
        MC = this;
        Debug.Log("MC setup");
    }
```

Figure 2. Code 1

To establish a proper level progression system, I created an enumeration listing each of the different stages of a level in a specific order. The order is as follows: Start, PretestPrompt, Pretest, PretestPause, Question, QuestionScore, PracticalTest, and Score (the Debug stage was used during game development to find and fix bugs, does not appear in the final version).

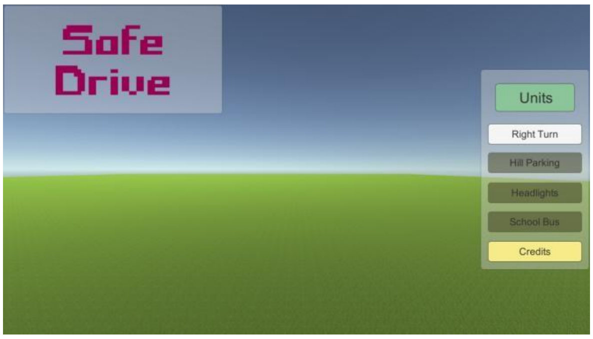


Figure 3. Screenshot of the game 1

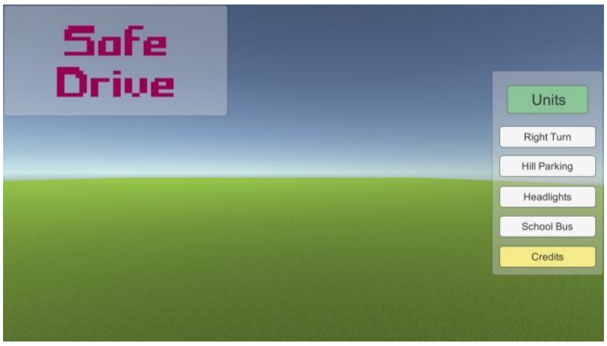
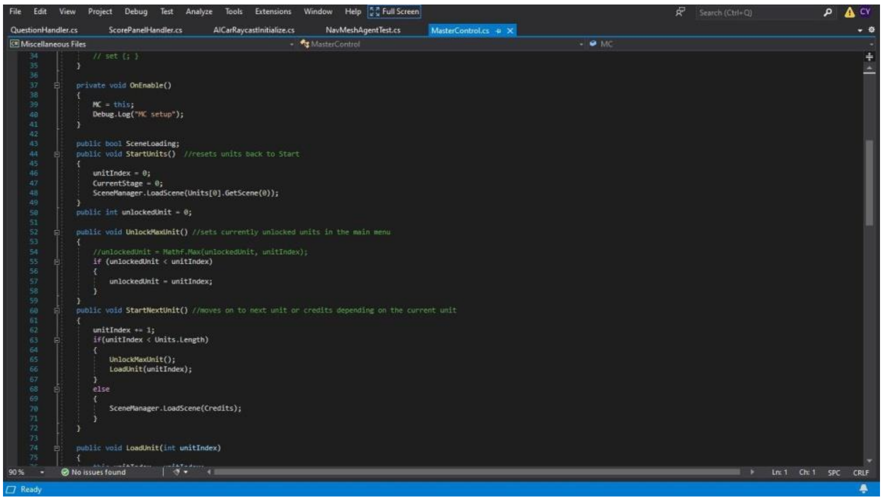


Figure 4. Screenshot of the game 2


File Edit View Project Debug Test Analyze Tools Extensions Window Help Full Screen
QuestionHandler.cs ScorePanelHandler.cs AICarRaycastInitialize.cs NavMeshAgentTest.cs MasterControl.cs
Miscellaneous Files
MasterControl
MC
90 %
No issues found
Ready
Ln: 1 Ch: 1 SPC CRLF

```
        // set {; }
    }

    private void OnEnable()
    {
        MC = this;
        Debug.Log("MC setup");
    }

    public bool SceneLoading;
    public void StartUnits()  //resets units back to Start
    {
        unitIndex = 0;
        CurrentStage = 0;
        SceneManager.LoadScene(Units[0].GetScene(0));
    }
    public int unlockedUnit = 0;

    public void UnlockMaxUnit() //sets currently unlocked units in the main menu
    {
        //unlockedUnit = Mathf.Max(unlockedUnit, unitIndex);
        if (unlockedUnit < unitIndex)
        {
            unlockedUnit = unitIndex;
        }
    }
    public void StartNextUnit() //moves on to next unit or credits depending on the current unit
    {
        unitIndex += 1;
        if(unitIndex < Units.Length)
        {
            UnlockMaxUnit();
            LoadUnit(unitIndex);
        }
        else
        {
            SceneManager.LoadScene(Credits);
        }
    }

    public void LoadUnit(int unitIndex)
    {
```

Figure 5. Code 2

In the first main menu screen, only the first level “Right Turn” is not shaded out, indicating that “Right Turn” is the only currently accessible level. The game features a level selection system where the player progresses through the levels chronologically. After the player finishes a level, he/she is able to play the level, and the new level will be permanently unlocked, which is indicated by the change in color. As shown in the second menu screen, all the levels have been unlocked.

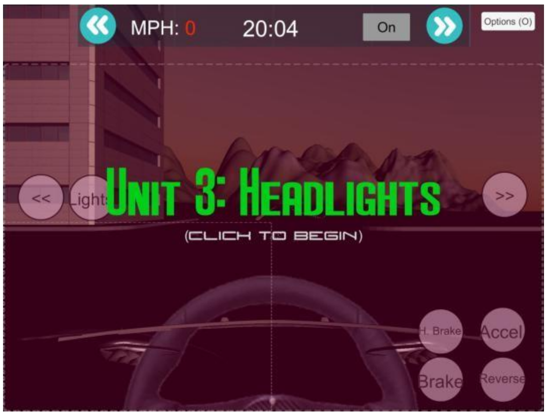


Figure 6. Screenshot of the game 3

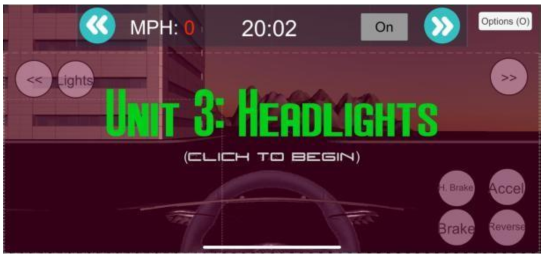


Figure 7. Screenshot of the game 4

This is one of the starting screens that appears at the beginning of each level. The starting screen displays the unit number and level title, e.g. “Unit 3: Headlights.” Each level is categorized into scenes in Unity. When the player finishes a level and continues onto the next, the MasterControl script incrementally adds to the unitIndex int variable to move to the correct level map. If the player selects a level from the level selection, the unitIndex variable will be set to equal the level scene index.

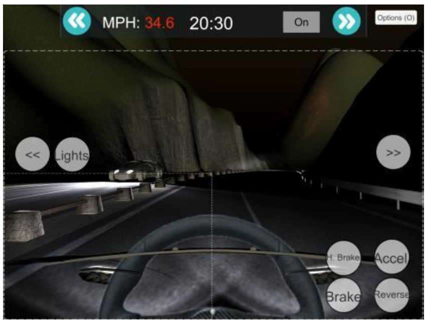


Figure 8. Screenshot of the game 5

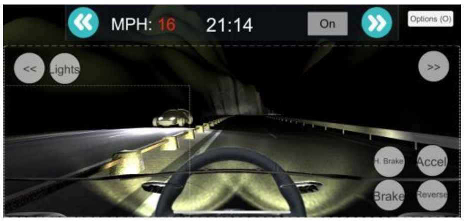
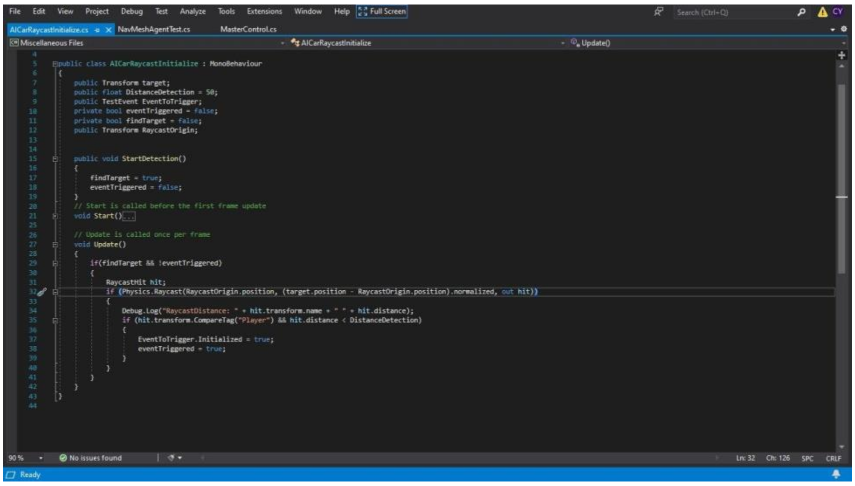


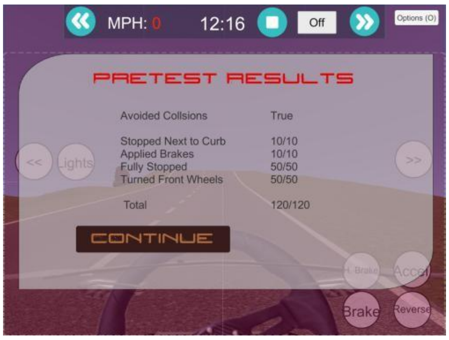
Figure 9. Screenshot of the game 6



```
public class AICarRaycastInitialize : MonoBehaviour
{
    public Transform target;
    public float DistanceDetection = 50;
    public TestEvent EventToTrigger;
    private bool eventTriggered = false;
    private bool findTarget = false;
    public Transform RaycastOrigin;


    public void StartDetection()
    {
        findTarget = true;
        eventTriggered = false;
    }
    // Start is called before the first frame update
    void Start()...

    // Update is called once per frame
    void Update()
    {
        if(findTarget && !eventTriggered)
        {
            RaycastHit hit;
            if (Physics.Raycast(RaycastOrigin.position, (target.position - RaycastOrigin.position).normalized, out hit))
            {
                Debug.Log("RaycastDistance: " + hit.transform.name + " " + hit.distance);
                if (hit.transform.CompareTag("Player") && hit.distance < DistanceDetection)
                {
                    EventToTrigger.Initialized = true;
                    eventTriggered = true;
                }
            }
        }
    }
}
```

Figure 10. Code 3

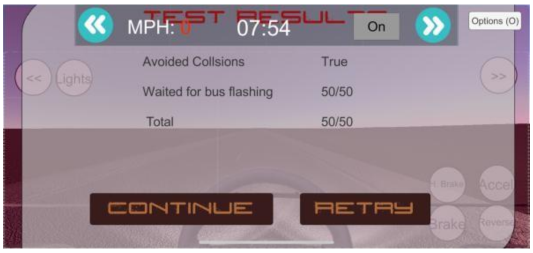
The AI car is able to locate its position relative to the player using raycasting. Both the player and the AI car have a RaycastOrigin object attached that allows the Raycast to calculate the distance between the two. Within the distance, the AI car is able to detect the light state of the player's car. This is important because the score for this unit is calculated based on the player's light state when approaching the AI cars.



PRETEST RESULTS

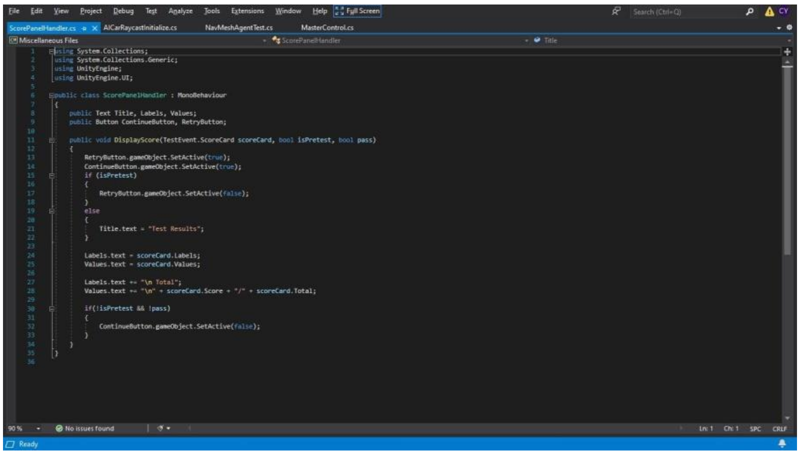
Avoided Collsions	True
Stopped Next to Curb	10/10
Applied Brakes	10/10
Fully Stopped	50/50
Turned Front Wheels	50/50
Total	120/120

Figure 11. Screenshot of the game 7



Avoided Collsions	True
Waited for bus flashing	50/50
Total	50/50

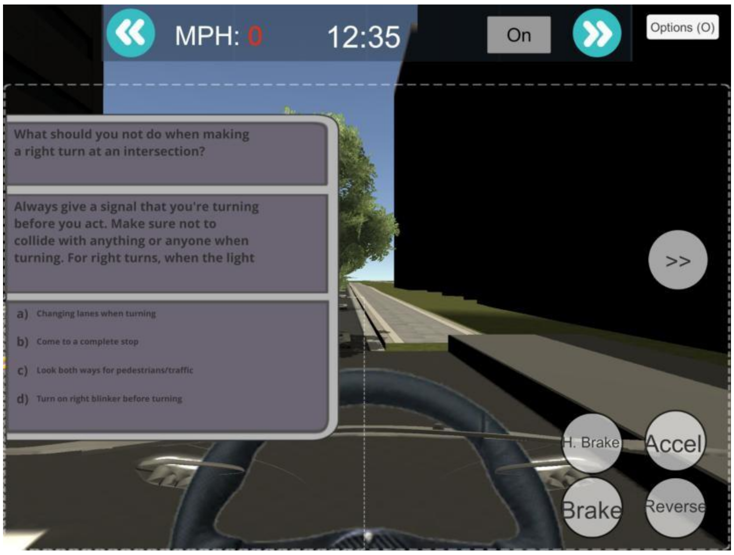
Figure 12. Screenshot of the game 8



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class ScorePanelHandler : MonoBehaviour
{
    public Text Title, Labels, Values;
    public Button ContinueButton, RetryButton;

    public void DisplayScore(TestEvent.ScoreCard scoreCard, bool isPretest, bool pass)
    {
        RetryButton.gameObject.SetActive(true);
        ContinueButton.gameObject.SetActive(true);
        if (isPretest)
        {
            RetryButton.gameObject.SetActive(false);
        }
        else
        {
            Title.text = "Test Results";
        }

        Labels.text = scoreCard.Labels;
        Values.text = scoreCard.Values;

        Labels.text += "\n Total";
        Values.text += "\n" + scoreCard.Score + "/" + scoreCard.Total;

        if(!isPretest && !pass)
        {
            ContinueButton.gameObject.SetActive(false);
        }
    }
}
```

Figure 13. Code 3

Test results are displayed on a scorecard when the player finishes the level. The criteria listed on the scorecard changes depending on the level. The two variants of the scorecard are the pretest and the test variants. Because the pretest is only meant as a self-diagnostics for the player, the retry button is taken out of the pretest scorecard, which is done by setting it to inactive when the bool variable isPretest is true. The kind of criteria shown is determined by Unity objects outside of the script, meaning that each level has its own dedicated objects for listing criteria.



What should you not do when making a right turn at an intersection?

Always give a signal that you're turning before you act. Make sure not to collide with anything or anyone when turning. For right turns, when the light

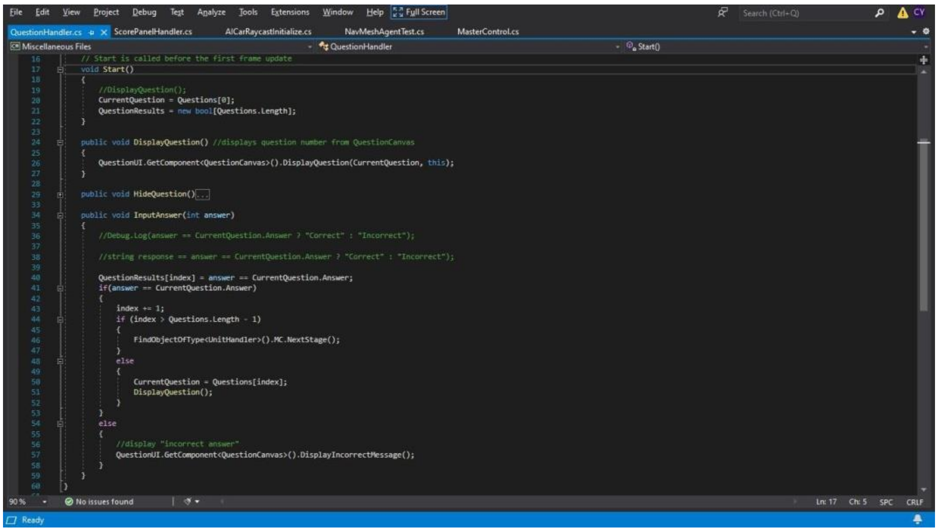
a) Changing lanes when turning

b) Come to a complete stop

c) Look both ways for pedestrians/traffic

d) Turn on right blinker before turning

Figure 14. Screenshot of the game 9



```
    // Start is called before the first frame update
    void Start()
    {
        //DisplayQuestion();
        CurrentQuestion = Questions[0];
        QuestionResults = new bool[Questions.Length];
    }

    public void DisplayQuestion() //displays question number from QuestionCanvas
    {
        QuestionUI.GetComponent<QuestionCanvas>().DisplayQuestion(CurrentQuestion, this);
    }

    public void HideQuestion()...

    public void InputAnswer(int answer)
    {
        //Debug.Log(answer == CurrentQuestion.Answer ? "Correct" : "Incorrect");

        //string response == answer == CurrentQuestion.Answer ? "Correct" : "Incorrect");

        QuestionResults[index] = answer == CurrentQuestion.Answer;
        if(answer == CurrentQuestion.Answer)
        {
            index += 1;
            if (index > Questions.Length - 1)
            {
                FindObjectOfType<UnitHandler>().MC.NextStage();
            }
            else
            {
                CurrentQuestion = Questions[index];
                DisplayQuestion();
            }
        }
        else
        {
            //display "incorrect answer"
            QuestionUI.GetComponent<QuestionCanvas>().DisplayIncorrectMessage();
        }
    }
}
```

Figure 15. Code 4

Within each level, the player is asked a question to test his/her knowledge. The way in which the question is changed for every unit is similar in concept to the score criteria.

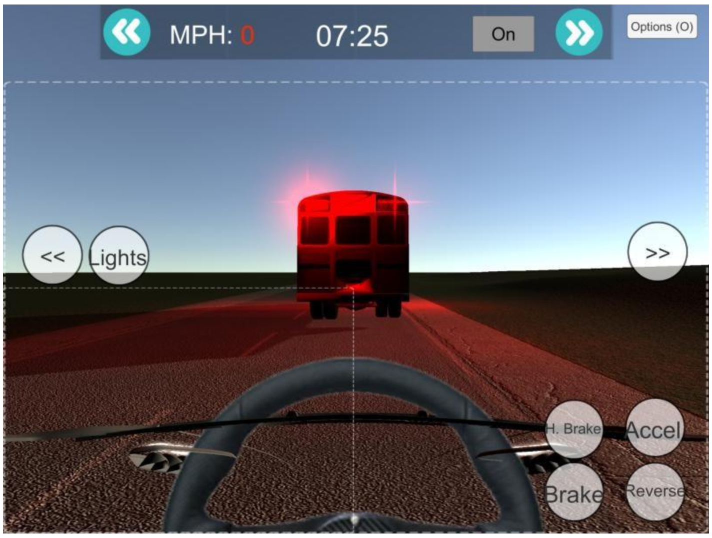


Figure 16. Screenshot of the game 10

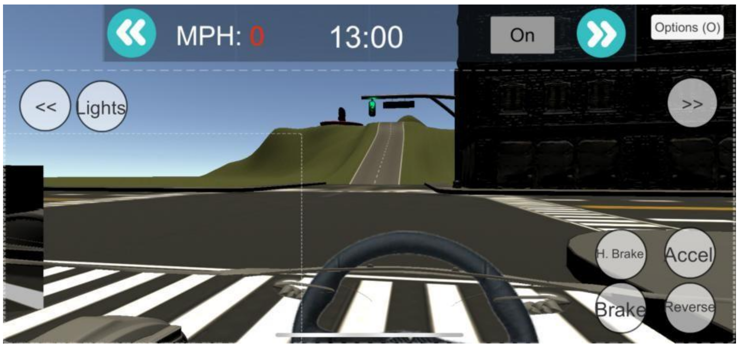


Figure 17. Screenshot of the game 11

Many methods were used to detect and judge the score criteria. For example, area detection, blinker detection, velocity detection, and head turn detection are the detector scripts used in the above two images. The area detection can be used for checking if the car either entered the area designated by the object hit-box or did not enter. blinker detection is used to determine which blinker the car turned on, and when. Velocity detection can be used to detect if the car has stopped or if it's under a certain speed limit. Head turn detection is used to check if the player turned their head by tracking the camera angle.

4. EXPERIMENT

4.1. Experiment

In experiment, We want to test how effective is SafeDrive at teaching players how to drive. To evaluate the effectiveness, We need to collect number of attempts, amount of time played, and helpfulness rating on the game tests. Evaluating the relationship between the number of attempts vs amount of time played and number of attempts vs helpfulness rating on tests will help me determine the effectiveness of SafeDrive in teaching people to drive. In general, the higher the number of helpful ratings, the more effective the game is, and the higher the amount of time played, the higher the level of engagement. I published the app on the app store in February, and as of March 18, out of 731 people who saw the app, 13 people installed it. 100 people ’ s responses were randomly chosen from the player base, meaning sample data comes from simple random sampling

Table 1. Result of experiment

	Passed tests with 1 attempt	Passed tests with2-5 attempts	Passed tests with 6+ attempts	Not passall tests	
play >24h	0	0	1	3	4
play 6-24h	1	6	15	2	24
play 1-6h	7	13	20	6	46
play <1h	6	10	8	2	26
	14	29	44	13	100

Table 2. Result of survey

	Very Helpful	Helpful	No help
Passed tests with 1 attempt	2	9	3
Passed tests with 2-5 attempts	4	19	6
Passed tests with 6+ attempts	5	27	12
Not pass all tests	0	5	8

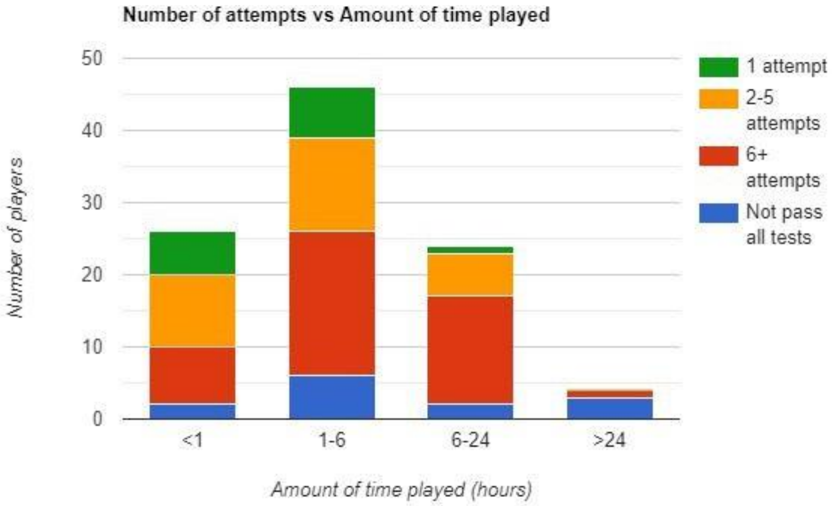


Figure 18. Number of attempts vs Amount of time played

5. RELATED WORK

This research paper is about predicting the potential benefits and the challenges that come with implementing driving simulator training in Norwegian driver education [12]. Although this study did not develop an actual simulator, it did anticipate the potential benefits of sim training based on other countries' results and discussed future challenges in implementing such a system. My study is not directly comparable with this study as the objectives of the two studies differ. My study involves developing the simulation while this study is more oriented towards "investigating the use of simulator training in driver education in Norway, discussing the potential gains and challenges and looking at the possibility of increasing the availability and use of driving simulators" (Taylor & Francis Group 2018). However, unlike this study, my experiment does not rely on results generated from other studies as my own simulation was able to produce concrete results.

Hirsch, Pierro, and François Bellavance did a study on the impacts of driving simulator-based training (DSBT) in driving programs in Canada [13]. This illustrates that the situation is happening around the world.

Similar to my study, Öztel, İsmail, and Ö. Z. Cemil's study also developed a driving simulator game [14].

6. CONCLUSIONS

To address the problem of youths more frequently getting into car accidents, I created a gamethat can teach beginner drivers how to drive safely. The game aims at teaching beginner drivers proper driving in a safe, risk-free environment, which has no solution to solve the problem perfectly on the market yet based on our study [15]. So I used the Unity game engine and C# to finish the whole project. The game plays out as a driving sim split into levels that teach individual driving concepts. Each level has its own individual driving concepts to teach and train. After development work is completed we publish the game on the ios platform, which is very easy to get more downloads and test people. After the experiment, we proved the app actually helps users improve their test score and learn faster.

Driving simulations on a flat screen can never substitute for actual driving. The controls in the sim vs real life are simply too different to compare. However, players will be able to learn concepts that they can apply in real-life driving. They will still have to learn how to drive using a steering wheel.

Although the game is free, the game contains relatively few levels -- too few to encapsulate all driving concepts. In order to include all content from the DMV test, the game development period would have to be greatly extended as it took half a year to create the 4 levels currently in the game.

It takes significantly less time to complete this game than in a traditional driving course. However, based on the results, most players have learned very little from this simulation.

I will have this game support gyro controls so that this can more accurately imitate real-life driving in terms of the steering wheel and head movements. I also plan to develop more levels so that more driving concepts can be added.

REFERENCES

[1] Forster, Yannick, et al. "What and how to tell beforehand: The effect of user education on understanding, interaction and satisfaction with driving automation." Transportation research part F: traffic psychology and behaviour 68 (2020): 316-335.

[2] Schneider, Jörg. Introduction to safety and reliability of structures. Vol. 5. Iabse, 2006.

[3] Rodwell, David, et al. "What, who, and when? The perceptions that young drivers and parents have of driving simulators for use in driver education." Safety 6.4 (2020): 46.

[4] Hirsch, Pierro, and François Bellavance. "Transfer of skills learned on a driving simulator to on-road driving behavior." Transportation research record 2660.1 (2017): 1-6.

[5] Parker, Dianne, et al. "Driving errors, driving violations and accident involvement." Ergonomics 38.5 (1995): 1036-1048.

[6] Squire, Kurt. "Video games in education." Int. J. Intell. Games & Simulation 2.1 (2003): 49-62.

[7] Arnett, Jeffrey. "Drunk driving, sensation seeking, and egocentrism among adolescents." Personality andindividual differences 11.6 (1990): 541-546.

[8] Room, Robin, Thomas Babor, and Jürgen Rehm. "Alcohol and public health." The lancet 365.9458 (2005): 519-530.

[9] Shields, Rob. The virtual. Routledge, 2005.

[10] Raz, Joseph. "Why Interpret?." Ratio Juris 9.4 (1996): 349-363.

[11] Hunt, Shelby D., Richard D. Sparkman Jr, and James B. Wilcox. "The pretest in survey research: Issues and preliminary findings." Journal of marketing research 19.2 (1982): 269-273.

[12] Sætren, Gunhild Birgitte, et al. "Simulator training in driver education–potential gains and challenges." (2018).

[13] Hirsch, Pierro, and François Bellavance. "Transfer of skills learned on a driving simulator to on-road driving behavior." Transportation research record 2660.1 (2017): 1-6.

[14] Öztel, İsmail, and Ö. Z. Cemil. "Traffic Education for Inexperienced Drivers with Virtual Driving Simulator." Sakarya University Journal of Computer and Information Sciences 2.2 (2019): 82-88.

[15] Damodaran, Aswath. "Estimating risk free rates." WP, Stern School of Business, New York (1999).