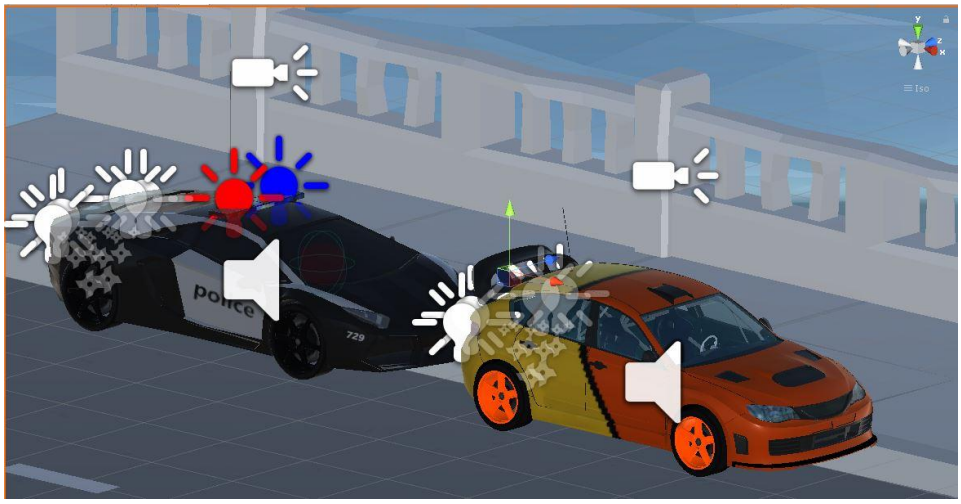
Traffic Stop for Law Enforcement



Victor Ngoubayo

Swetha Thodupunori

Prince Ikegwuono

Bowie State University

COSC 729 – Spring 2021

Dr Sharad Sharma

Major Tools: Unity 3D

GOAL AND OBJECTIVES:

The project is a simulation of a law enforcement traffic stop from the driver and occupants' point of view.

It encourages the user to experience the event first hand and to make decisions that may escalate or de-escalate the situation

The project is based on Unity 3D, with minor polygon editing that was carried out in Google SketchUp.

MODELING:

The envisioned virtual environment is a realistic representation of street and highway traffic, including roads, drivers, pedestrians, traffic signs and buildings. However, this calls for a large expanse and collection of polygons. Reducing scene size was critical and measures were taken to address this. This was achieved by the mixed use of low resolution and high resolution polygons due to the dense geometric requirements to represent such a vast environment. Models were sourced entirely online from 3D databases in order to take advantage of preexisting geometry and textures. Much of the modeling and geometry consists of empty shells that do not include internal details, such as building furniture, car engines, etc.

The environment would include a health bar and onscreen instructions for the user to interact with. This provided opportunities for creativity but this was tempered accordingly due to the screen real estate and presumed high level of screen activity. The health bar inflicted a

penalty of 25 percent of life loss due to colliding with an inanimate object, and 50 percent of life loss for colliding with an avatar.

Figure 1. Bird's eye view of environment

Figure 2. Screenshot of Graphic User Interface (GUI)

PROGRAMMING:

Since the project is based on Unity 3D, C# was the main programming language employed in achieving interactivity and functionality. C# scripts were used to control the health bar, control vehicles, control street lights, trigger enemy and obstacle sensors, animate car lights and control camera views. C# programming was also used to populate the scene with over 500 autonomous avatars.

FUNCTIONALITY:

Functionality focused on six areas of the project, including Interactivity, Vision, Sound, Animation, Sensors and Avatars. This was implemented as follows:

Interactivity:

- Project interactivity was developed entirely and built on Unity3D functionality.
- User will be driving a car in a First Person Point of View (FPPOV).
- User will navigate the car driving experience with mouse and WASD buttons on keyboard.
- Information and instructions is displayed onscreen.
- User would encounter and avoid road hazards or loose health, such as traffic lights, street lamps, cars and pedestrians.
- User would encounter law enforcement and ordered to stop.
- Subsequent onscreen instructions would instruct user to make choices.
- User choice would determine good or bad outcome of the encounter.

- Good encounter would allow user to continue.
- Bad encounter could result in loss of life.

Vision:

- Textures and 3D models were used to provide detailed information.
- Emphasis was placed on those models and textures that provided the user with the best level of interactivity at the sacrifice of low quality textures and models in the background.

Sound:

- City traffic audio sound was used as background ambient sound. This was attached to the camera as an audio source that played upon wake.
- Sound was law enforcement siren, car crash sound and tires screeches.
- Animation: Use of at least three animated objects.
- Sensors: Use at least three different types of sensors (Proximity, Colliders).
- Avatars: Use of custom and inbuilt avatars

Animation:

- Animation was applied to over 500 autonomous animated avatars in the scene. A generic animation clip was applied to all the avatars through the use of C# script.
- Animated cars.

- Animated car lights.
- Animated speedometer on GUI.
- Animated camera views.
- Animated special effects, such as smoke

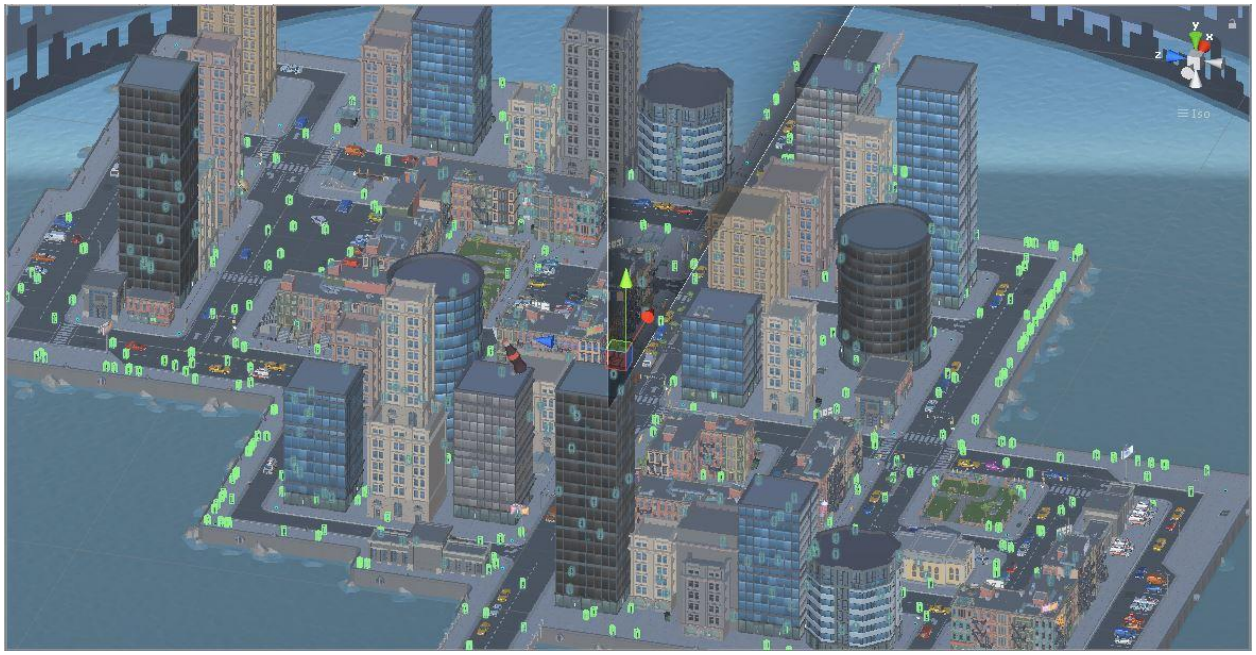


Figure 3. Bird's eye view of the city, showing highlighted avatars.

Sensors:

- Proximity sensors were used to monitor distance between other drivers and pedestrians.
- Proximity sensors are also used to trigger law enforcement chase.

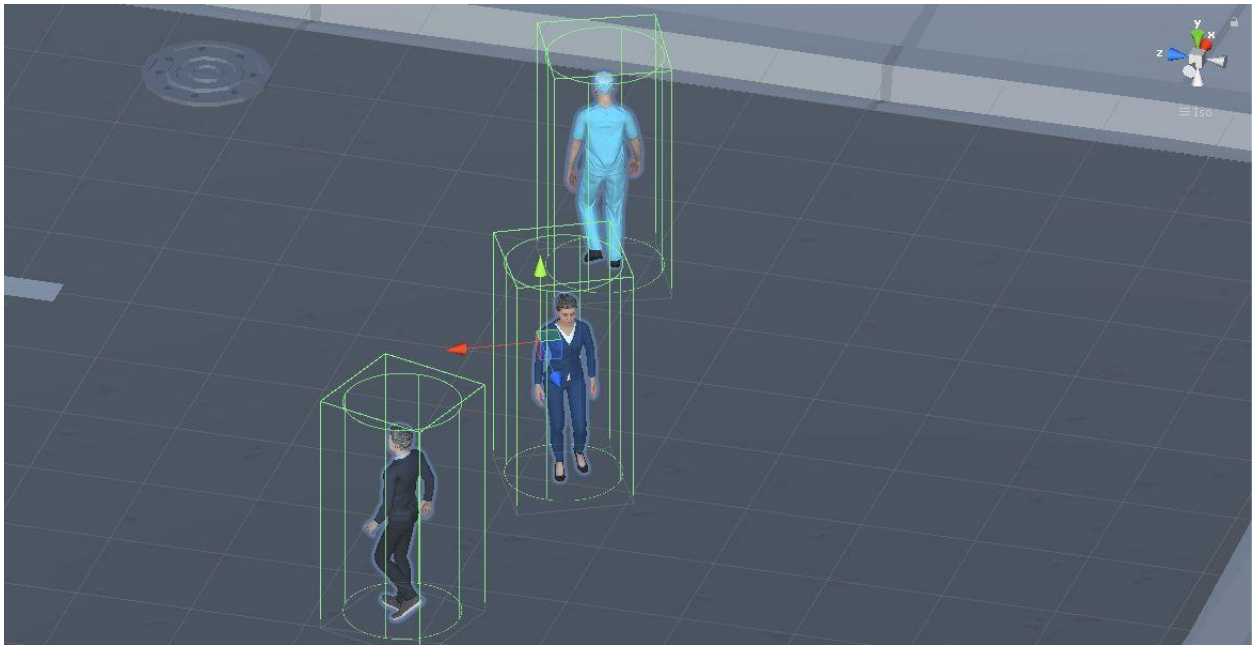


Figure 4. Box collider attached to avatar for proximity sensing.

Figure 5. Box collider attached to road as trigger for law enforcement car chase.

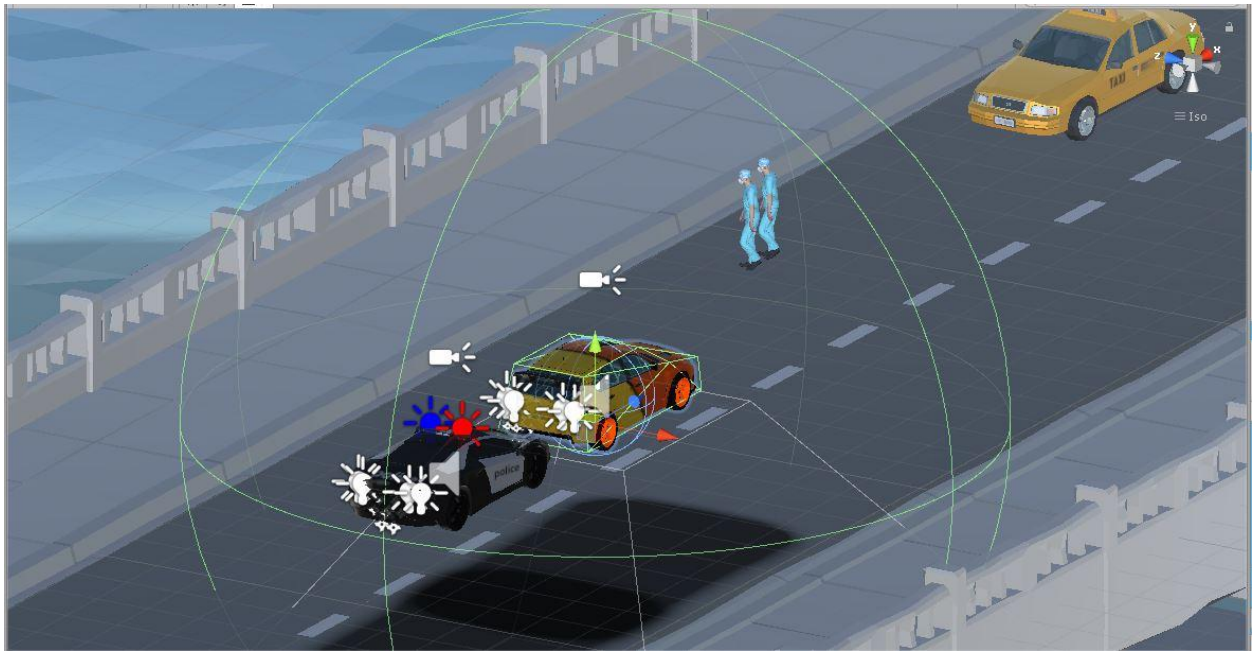


Figure 6. Sphere collider attached to player car as trigger for law enforcement shooting range.

WHY VIRTUAL REALITY?:

Traffic stops are a routine operation of law enforcement these days. It is the most common interactions between a law enforcement officer and a civilian. However, traffic stops are highly risky to both parties due to the uniqueness of each situation. A stop initiated for a simple reason may quickly escalate to a life and death situation.

This virtual reality simulation is developed to assist drivers and law enforcement with current traffic procedures.

- It leverages the futuristic capabilities of immersive virtual reality experiences.
- It can be used in education and training for law enforcement officers and civilians.
- It can be used to assist legal adjudication.

- It can be used for entertainment.

PROBLEMS ENCOUNTERED:

- Limited time.
- Limited expertise in C# scripting.
- Lost time due to debugging errors, else game will not run.
- Computer operating system corruption and computer hardware breakdowns.

REMAINING SHORT COMINGS (TO DO LIST):

- Complete law enforcement car chase.
- Player running away and law enforcement foot chase.
- Law enforcement opening fire during foot chase.
- Law enforcement interview and interaction with player.
- Welcome screen and Game Over splash screen.
- Implementing city traffic light system.

SOFTWARE AND HARDWARE USED:

- Unity 3D.
- Google SketchUp.
- Adobe Photoshop.
- Audacity.

- Dell Precision T5500 Workstation, Dual Intel Xeon X5677, 32GB RAM, Geforce GTX 980 Ti.
- Oculus Rift.

ASSET SOURCES:

Models (royalty free) - Unity Asset Store, TurboSquid, 3D Warehouse, CG Trader.

Animation (royalty free) - Mixamo.

Sound (royalty free) - Zapsplat, BBC, FreeSFX, Freesound.