Assignment 1

Dr. Bastani

Scenario

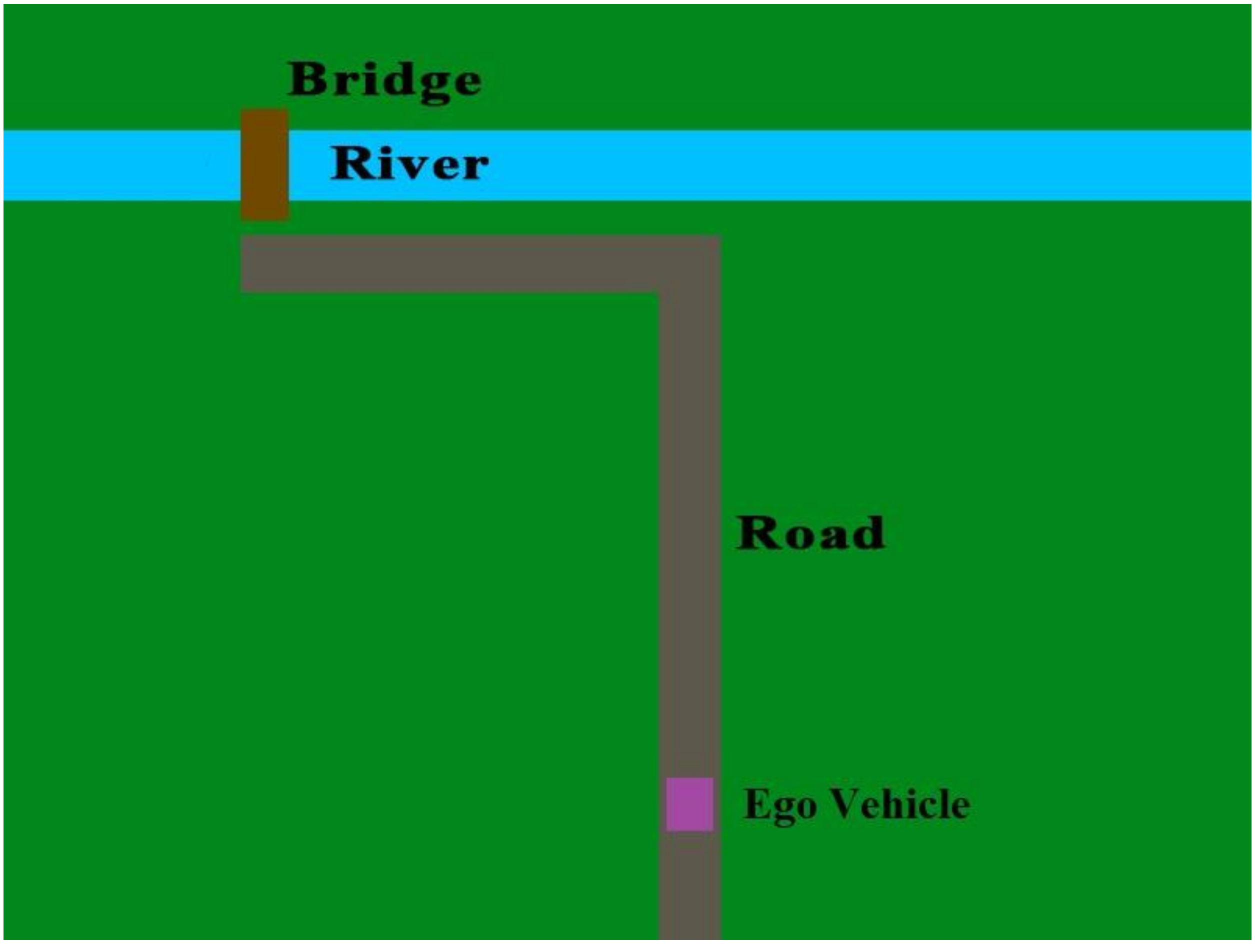

Figure 1

- **Goal**: Move the Ego vehicle across the bridge.
- **Constraints:**
  - Use only the sensor data which is given through the RTOS.
  - Use only the actuators that are given by the RTOS.
  - Stay on the road
  - Don’t fall into the river

Sensors

- DeviceRegsitry.pixels : Readings from a 7 pixel tall by 15 pixel wide color camera.
- DeviceRegistry.compass : Readings from a compass that tells you how different your current orientation is from North.
- DeviceRegistry.speedometer : Readings from a speedometer that tells you how fast the vehicle is moving in m/s.
- DeviceRegistry.microphone : Readings from a microphone that tells you the frequency of nearby sounds.

For more information about these available sensors, please read the Device Registry documentation. Sensors that are not listed here will not update at the beginning of each frame, and will only contain the value 0.

Actuators

- DeviceRegistry.speedControl : The control bus for the acceleration subsystem that controls the speed of the Ego vehicle.
- DeviceRegistry.brakeControl : The control bus for the brakes that can slow down the Ego vehicle.
- DeviceRegistry.steeringControl : The control bus for the steering controller that changes the direction the Ego vehicle is facing.
- DeviceRegistry.transmitterControl : The control bus for the signal transmitter that broadcasts commands over radio frequencies.

For more information about these available actuators, please read the Device Registry documentation.

Additional Environment Information

There are some other entities and objects in this scenario that you must interact with, listen to, or otherwise avoid. Details about them will be documented here.

Color Meanings

Each pixel of the camera contains an RGB color that was detected by a raycast sent out by that pixel. The objects in the scene all have specific colors assigned to them, although not all colors are unique.

Color (RGB)	Objects
⬛ (0, 0, 0)	Empty Space (No object detected)
🟩 (0, 255, 27)	Grass
◼ (120, 120, 120)	Road, Boat, Alarm
🟦 (0, 124, 255)	Road
🟫 (101, 33, 0)	Bridge
🟥 (255, 0, 27)	Boat
⬛ (24, 24, 24)	Boat
🟧 (255, 131, 0)	Boat

Boat

There is a boat which passes through the river at randomized intervals every 5 to 15 seconds. When the boat passes, the bridge will be raised, and the alarm speaker will send out the raising alarm sound.

Bridge

When the boat reaches the bridge, the bridge is raised. After the boat passes, the bridge will remain raised for a few seconds, and then it will lower itself. The timer for the next boat is not reset until the bridge is fully lowered. It is not safe for the Ego vehicle to cross the river when the bridge is raised, therefore the Ego vehicle must wait until it is safe to cross.

Alarm Speaker

The alarm speaker is capable of emitting one of three sound signals when it is commanded to do so via the radio transmitter. These sound signals can be detected by the Ego vehicle's microphone sensor when the Ego vehicle is within 30 meters of the alarm. Each sound signal is associated with a particular state of the environment.

The possible sound signals and their associated states are:

- 0 : Alarm is silent, or too far away from the Ego vehicle to be detected.
- 33 : The Ego vehicle is currently crossing the bridge.
- 41 : The bridge is raised

For example, when the boat is passing through the river, and the bridge begins raising, then the Ego vehicle's microphone sensor will contain the value 41 so long as the Ego vehicle is within 30 meters of the alarm. If the Ego vehicle is not within 30 meters of the alarm, then the Ego vehicle's microphone will contain the value 0, even if the bridge is raised.

The alarm will **not** emit any audio from your computer speakers. You can tell what sound signal is being emitted by the alarm by looking at the color of the sphere at the top of the alarm object, which is located next to the bridge. Blue is audio signal 0, Green is audio signal 33, and Red is audio signal 41.

Alarm Radio Receiver

The alarm speaker will emit its sound signals only when it is commanded to do so via the radio transmitter on the Ego vehicle or the transmitter on the boat. Transmissions from the boat override all transmissions from the Ego vehicle. To transmit a command, you will select give the Ego vehicle's transmitter the radio frequency you wish to broadcast on. The Ego vehicle will then send an "On" command over that frequency at the start of the next frame.

The possible transmission frequencies and the resulting behavior from the alarm are as follows:

- 42.5 : Tell the alarm to go silent (emit sound signal 0).
- 55.6 : Tell the alarm that the Ego vehicle is crossing (emit sound signal 33).
- 78.9 : Tell the alarm that the bridge is raising (emit sound signal 41). The boat will automatically transmit this radio frequency when it passes through the river.

After a transmission has been received by alarm, the alarm will continue to emit that audio signal until it is commanded to emit a different signal.

Problem 1: AVL Environment (10 points)

The UTD Autonomous Vehicle Lab (AVL) is a simulation environment which has been developed as a teaching aid for students to learn about autonomous vehicle and Real-Time Operating System (RTOS) technology. The AVL is implemented in the Unity physics engine, and is populated with various obstacles for your Ego car to avoid. All your scripts will be implemented in C#, which is the native language for the Unity engine.

For this assignment, there will be no other vehicles in the simulation. However, you must still avoid leaving the road or falling into the river.

For Problem 1, you will familiarize yourself with the AVL environment by doing the following:

1. Add a pre-defined debug task to the Task List.
2. Modify the pre-defined debug task to accelerate the Ego vehicle when the “W” key is pressed.

All of the information you need to complete these instructions is contained within the AVL documentation, which can be found here:

- https://github.com/hatfield-c/avl

Problem 2: Custom Control Task (50 points)

Now that you are familiar with the AVL environment, you will implement your own C# scripts which define a set of custom control tasks. You will then remove the previous control task from the Task List, and add your custom tasks in its place. Do not delete the previous task C# file! You will submit it along with the other tasks you create.

The goal of the new control tasks is to successfully navigate from the vehicle’s starting position to the land on the other side of the bridge, as demonstrated in Figure 1. You must make control decisions using only the available sensor data that is provided by the RTOS. You cannot use keyboard controls like was done in the task from Problem 1.

You are encouraged to experiment with the sensor data to understand how it can be used effectively. You do not have to use all the sensors. However, the Ego vehicle will have its starting position on the road and its starting rotation randomized at the beginning of each simulation. Thus you will need to use at least some sensor data to complete the task.

For safety reasons, you must activate the appropriate audio signal when you are crossing the bridge. If you cross the bridge without activating the crossing alarm signal, you will lose points. Review the Additional Environment Information section for information about the alarm, and review the Device Registry documentation for information about how to control the transmitter.

Occasionally, the bridge will raise to allow a boat to pass underneath. When the bridge is raised, a raising alarm will be initiated, which has a different sound than the crossing alarm. You cannot cross the bridge when the bridge is raised, or else the Ego vehicle will fall into the river. Review the Additional Environmental Information section for information about the bridge and boat, and review the Device Registry for information about how to listen for the raising alarm with the microphone.

Problem 3: Documentation (40 points)

You must document and describe your implementation in a final report. This documentation must include:

- Information about the tasks which you defined to control the Ego vehicle
    - Task name
    - Task description
    - Dependencies on other tasks and/or sensor data
- A graph showing the task schedule for at least two cycles of the simulation. Describe the graph. Point out which tasks are being executed, and explain at what point in the simulation the task is occurring at.

Submission Requirements

To receive credit, you must submit the following items within a ZIP file:

- All .cs files that define the classes of your custom tasks.
- The updated TaskList.cs file that adds your custom classes to the Task List.
- Your documentation.
- A text file that contains the NET IDs of all group members.

To be safe, all group members should submit the same ZIP file. If you have an issue with a group member who does not contribute, contact the TA beforehand.