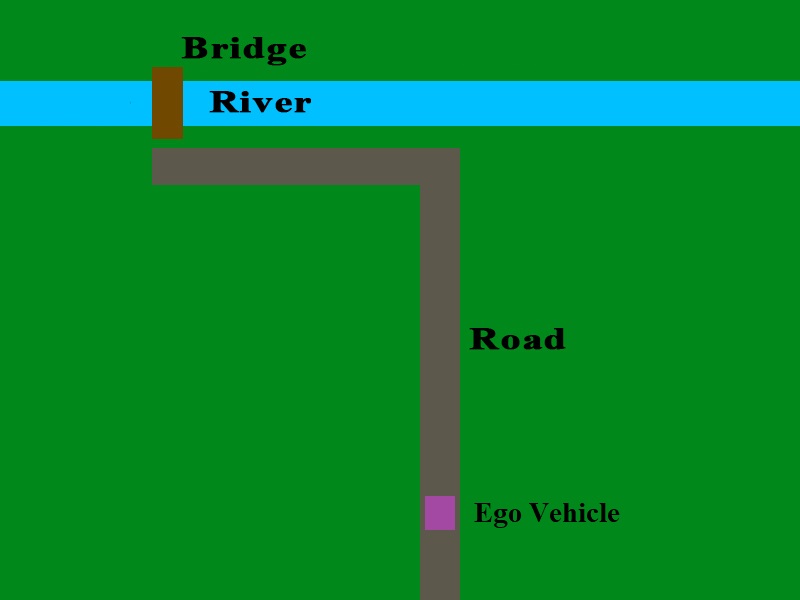
**Assignment 1**

CS-XXXX.YYY

**Scenario**

Figure

* **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**

* A 5 pixel by 5 pixel color camera
* A compass which tells you how different your current orientation is from North.
* A speedometer
* A microphone

**Actuators**

* Acceleration Subsystem
* Brakes
* Steering Controller
* Signal Transmitter

**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 RTOS scheduler that is located on the Ego vehicle object.
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# script which defines a set of custom control tasks. You will then remove the previous control task from the RTOS scheduler, and add your custom task into the scheduler (do not delete the previous task C# file!)

The goal of the new control task 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 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 position on the road and its rotation randomized at the start of each simulation. Thus you will need to use at least some sensor data to complete the task.

For safety reasons, you must transmit a signal that activates a unique audio alarm when you are crossing the bridge. If you cross the bridge without activating the crossing alarm, you will lose points.

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.

**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
  + Is it periodic, aperiodic, or sporadic?
  + Period
  + Deadline
* Functional and non-functional requirements of the system
* A graph showing the task schedule for at least two cycles of the simulation.
  + This can be generated by examining the debug options located in RTOS panel on the Ego vehicle object in the Unity scene.
  + Describe the graph. Point out which tasks are being executed, and explain at what point in the simulation the task is occurring at.

Please note that while bad documentation may lose you at most 10 points, providing no documentation will result in an automatic -50 points penalty.