# A Tile Manager for Deploying Scenarios in Virtual Driving Environments

**Piriyakala Suresh and Ronald R. Mourant**

Virtual Environments Laboratory,

Dept. of Mechanical and Industrial Engineering

334 Snell Engineering Center

Northeastern University

Boston, MA  02115-5000

Email: mourant@coe.neu.edu

## Abstract

A tile based scenario system that allows an experimenter to 1) change the order and number of tiles; 2) change the parameters of a scenario via an XML file; and 3) provide both standard performance measures as well as those that are unique to a particular scenario has been developed. Each scenario is viewed as a tile and is defined using the eXtensible Markup Language (XML). This allows many of the scenario's parameters and features to be changed without the need to recompile the program. The software includes a "Tile Manager Interface" that allows an experimenter to build a session that includes a variable number of tiles with each tile representing a scenario. The developed "Tile Manager Interface" is useful in building experimental sessions to evaluate driver performance. Scenarios (tiles) can be presented in various orders allowing for counterbalancing. Performance measures specific to particular scenarios can be defined. The level of difficulty of a particular scenario can be controlled by adjusting parameters in an XML file using a text editor.

## Introduction

### Background

Studies have suggested that PC based virtual driving environments are effective in the evaluation of cognitive functions (Lengenfelder et al., 2002; Rizzo et al., 2003) as opposed to traditional neuropsychological measures. It has also been shown that PC based simulators can be effectively used in novice driver training (Allen et al. 2003). These virtual driving environments can be defined at various levels: using script based or user friendly drag and drop interfaces to define the environment at high levels or programmatically build the environment at a low level. The high level interfaces have been programmed by commercial simulator providers and academic research facilities. They have the user friendliness required by the common user as a trade off against complex environment design. This paper looks at a tile manager that provides scenarios and built-in performance measures with options of inserting, rearranging, and modifying the tiles by ordinary computer users: e.g. medical/behavioral researchers, physicians, etc.

### Objectives

This work was motivated by the need for simple, low cost virtual environments that allow the measurement of driving performance based on presenting participants with scenarios that can be arranged in various orders. A scenario can be dropped into a folder as an XML file. The order of scenario presentation depends on the order of the XML files prescribed in a script file. This allows a researcher to design experiments that are counterbalanced across subjects with a minimal amount of extra work. Each custom built scenario is viewed as a tile and has its own set of associated performance measures.

### Environment

The software engine was built using Java and the Java 3D Application Programming Interface (API). The runtime requirements for this tile manager are a PC with reasonable graphics and the freely available Java 3D runtime software. To extend the scenarios provided with the tile manager the following freely available tools can be used:

- A text editor (may be with xml tag highlighting).
- A CAD tool to design the roads and infrastructure (or a piece of paper and pencil for a rough drawing).
- A photo editor to use custom photos textured on objects (roads, trees, signs etc)

## Scene and Scenario Development

### Defining a tile

A scenario is built on a tile: a section of a route that consists of roads, signs, trees, buildings, traffic elements and scenario objects. A tile need not have a fixed size, but should have uniform starting and ending roads. These tiles can be loaded one after

another by listing them in a text file. One can decide to just try a single scenario, or a subset of the available scenario tiles.
A tile is defined such that at a minimum there should be a road segment. The other objects listed in figure 1 are optional depending on the requirement.

```
<Tile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="tile.xsd" version="0.0.0" author="Priya Suresh"
date="2004-02-05" TileID="1" Description="Gap Acceptance scenario"
StartMidPoint="X = 522.6021 Y = -11205.1630" StartXAngle="117.205"
EndMidPoint="X = -775.7235 Y = -10893.9713" EndXAngle="27.518">
  + <RoadInfo>
  + <TreeInfo>
  + <SignInfo>
  + <ScenarioData>
  + <BuildingInfo>
  + <TextureWallInfo>
  + <PerformanceRequests>
```

**Figure 1: Components of a tile**

Sample objects in the xml file are shown in the following figures.

```
<RoadSegInfo RoadSegID="100">
    <Region>
        <startPointY>0</startPointY>
        <startPoint>X = 3508.2218     Y = -14989.5420</startPoint>
        <endPointY>0.2</endPointY>
        <endPoint>X = 2208.2218     Y = -14989.5420</endPoint>
        <textureFileName>RoadTwoLane.jpg</textureFileName>
        <upOrDown>0</upOrDown>
    </Region>
</RoadSegInfo>
```

**Figure 2: Road definition in XML**

```
<Tree>
    <startPointY>0</startPointY>
    <startPoint>X=3386.5000  Y=-15013.5000</startPoint>
    <endPointY>0</endPointY>
    <endPoint>X=2300.5977  Y=-15013.0080</endPoint>
    <space>60</space>
    <upOrDown>0</upOrDown>
</Tree>
```

**Figure 3: Tree definition in XML**

Each tile can also have specific performance measures associated with it. Currently we have implemented two such specific performance measures. They can be used on the same scenario or alone.

Below we show a sample of the Gap Acceptance performance request in xml format.


```
<PerformanceRequests>
        <GapAcceptance>
            <AutonomousVehicleCrossLine>
                <startPoint>X=-365.1299  Y=-11692.9352</startPoint>
                <endPoint>X=-379.3624  Y=-11665.5196</endPoint>
            </AutonomousVehicleCrossLine>
            <DriverCrossLine>
                <startPoint>X=-481.4499  Y=-11718.5172</startPoint>
                <endPoint>X= 389.4472  Y=-11266.4012</endPoint>
            </DriverCrossLine>
            <StopLine>
                <startPoint>X=-347.5947  Y=-11662.8683</startPoint>
                <endPoint>X=-337.1362  Y=-11683.2809</endPoint>
            </StopLine>
        </GapAcceptance>
</PerformanceRequests>
```


**Figure 4: Gap Acceptance Performance Measure definition in XML**

## Configuring the run

The following parameters can be configured before the run and are shown in Table 1.

**Table 1**
**Configurable Parameters before the run**

| | Parameter | Values |
|---|---|---|
| 1 | Driving control | Steering or Mouse |
| 2 | Head tracking | On or Off |
| 3 | Display car frame | Steering (On or Off), speedometer (On or Off), signals (On or Off) and car frame (On or Off) |
| 4 | Force feedback | Value in Nm (default is 2) |
| 5 | Field of view | Value in degrees |
| 6 | Resolution of display | 640 x 460, 800 x 600, 1024 x 768, 1600 x 1200 |
| 7 | Audio controls | Volume on a scale of 0 to 100 |
| 8 | Performance measures | On or Off |
| 9 | Name of the directory to save the recorded date | There will be a raw data file and processed output files placed in this directory/folder. |
| 10 | List of tiles to be loaded | Name of the xml files listed one on each line |

## Performance Measures

### *General Performance Measures*

We collect raw data at runtime and process the data after the run to produce more meaningful results to the user. A tab delimited output file contains the following parameters as shown in Table 2.

Table 2
Performance data output parameters

| | Parameter | Description |
|---|---|---|
| 1 | Time (s) | Elapsed time since the start of the run in seconds |
| 2 | X Position (ft) | Driver position in the X axis in feet |
| 3 | Y Position (ft) | Driver position in the Y axis in feet |
| 4 | Distdriven (miles) | Distance driven since the start of the run, in miles |
| 5 | SpeedCar (mph) | Speed of the vehicle in miles per hour |
| 6 | Acceleration (mph/s) | Acceleration of the vehicle in mph/s; deceleration will be negative |
| 7 | Gas | Gas pedal value in units returned by the hardware |
| 8 | Brake | Brake pedal value in units returned by the hardware |
| 9 | StrAng (deg) | Angle turned by the steering wheel from the original position in degrees |
| 10 | StrVel (deg/s) | Angular velocity of the steering wheel in degrees per second |
| 11 | TrackerYaw (deg) | Head turn angle as measured by the head tracker, in degrees |
| 12 | Crashed? | Was there a crash at this point in time |
| 13 | StopLineCrossed? | Was the stop line crossed at this point in time |
| 14 | DistToCenterLine (ft) | Distance to the centerline from the driver position is output in feet. Intersections and areas that do not have a visible centerline have an imaginary centerline that smoothly connects the centerlines of its adjacent road segments in the direction of travel. For example if we were to make a turn, the centerline of an intersection will be a curve, and if the requirement was to go straight at an intersection, that centerline would be straight. |
| 15 | Tile Id & Road Id | These variables act as a cross reference to the XML file and will be useful in identifying a certain section of the run for analysis depending on the TileId and RoadId |

Turn profiles of the driver is of interest in subject studies and when testing the fidelity of simulators. It can give a visual presentation of how well the turn was made. To make the data of the driver path meaningful, the road and centerline data were required at the end of the run in a format that can be brought into a spreadsheet or statistical package. For this purpose there are two more data files printed out in a tab delimited text format.

*Sample Plots of Data*

The data files were taken into Microsoft Excel and some plots of data are given.

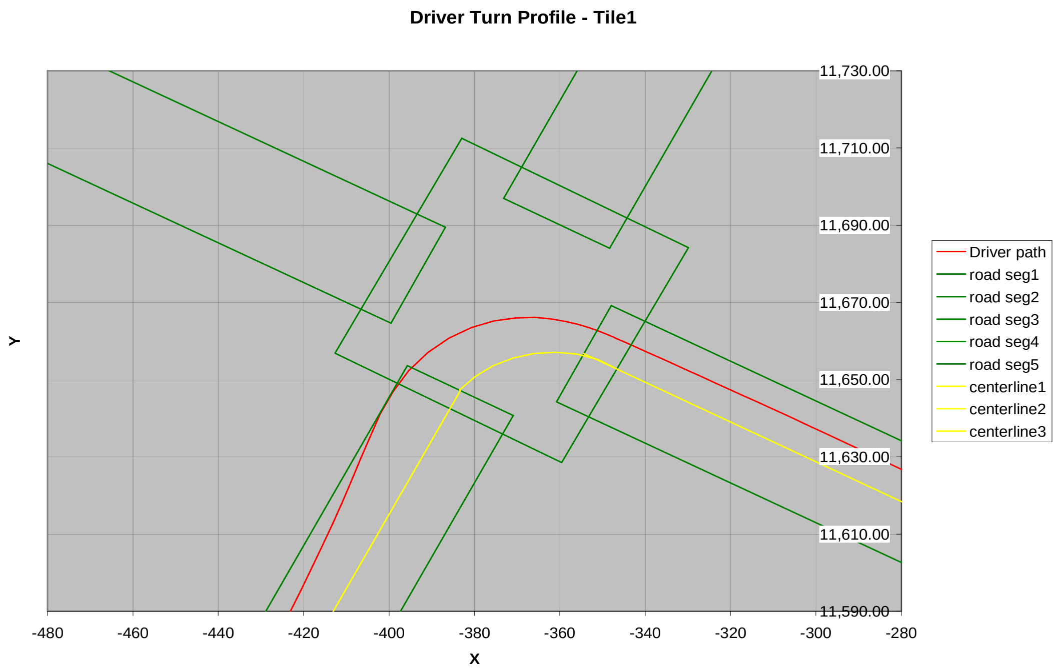


**Figure 5: Driver making an imperfect left turn**

This plot was achieved by plotting the road information, the center line data, and the driver's path superimposed on them. Depending on the length of the turn of interest, data can be chosen for the plot.

In Figure 6 Velocity against Distance Driven gives the speed profile with the stop line super imposed on the graph to show the distance from the stop line the driver came to a stop. It is scaled to show the deceleration when approaching the stop line and the acceleration afterwards. These data were obtained using a mouse for driving: the variations are not realistic as with a steering wheel and pedals.

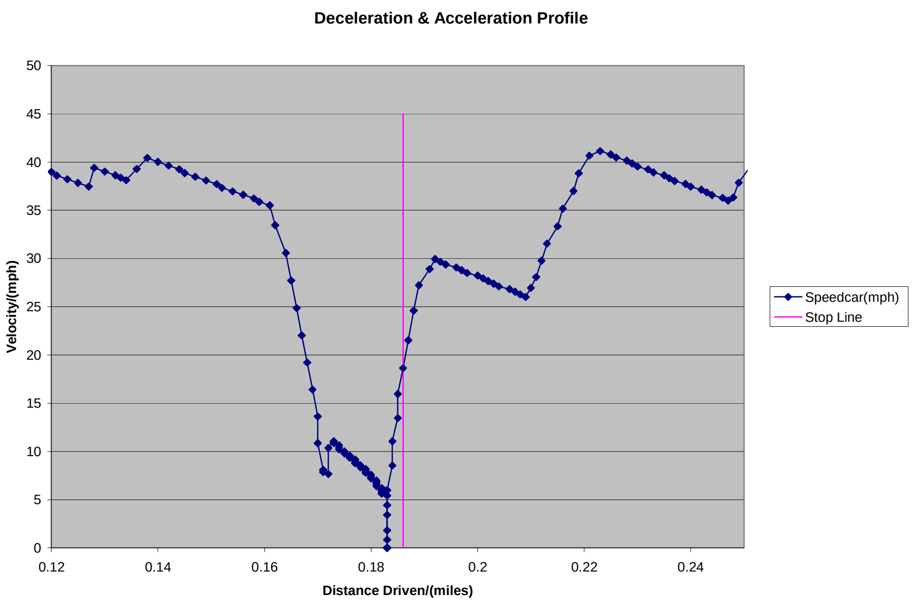


**Figure 6: Driver's deceleration and acceleration profile**

### *Specific Performance Measures*

Each tile can have a specific performance measure associated with it using the XML file. Currently the implemented specific measures are:

1. Gap Acceptance: Print out the available gap times between other vehicles and the gap time chosen by the driver to make the turn. This can be either in a left turn against oncoming traffic or to make a turn at a stop sign among cross traffic.
2. Turn Duration: Print out the time taken for a completed turn. If the driver came to a full stop near the stop line, then the time is calculated since he stopped. Otherwise, the time is calculated based on the time the car crossed the stop line.

```
Summary of results for subject1

Turn duration time for Y-intersection scenario(scenario 2): 5.218998 seconds

Turn duration time for Y-intersection scenario(scenario 4): 5.140999 seconds

Gap times available in Gap Acceptance scenario(scenario 1):
1.5619993 seconds
2.2820015 seconds
3.1089993 seconds
3.203001 seconds
Gap time chosen for Gap Acceptance scenario: 2.2820015 seconds
```

**Figure 7: Specific measures for tiles**

### *Playback*

The collected data allows us to playback the path taken by the driver simulating accidents and other interactions. The experimenter can choose to playback in 2 modes: driver's view and a bird's eye view. The run can be paused for study.

## Conclusions

The Tile Manager software allows users to perform human performance studies by presenting subjects with a different order of scenarios with minimal effort, and collect relevant data. Further, the use of XML to model data allows modifications to the 3D world even after the project has been compiled and delivered. Performance measures have been classified into general measures that are recorded for all the tiles, and specific measures that are put in the XML file for each tile.

Tiles have been used to describe the static scene previously (Papelis et al., 2003; Steed et al., 1999). Our extension includes the dynamic data as part of a tile. Thus when a tile is moved any scenarios (dynamic data) in the tile are also moved.

The idea of specifying a series of tiles in a configuration file could be further exploited to build environments by removing and loading tiles at run time when the environment is too large to fit into the memory (Steed et al., 1999). One would have to consider the time taken to parse the XML file at run time.

## References

Allen R.W., Park G., Cook M., Rosenthal T. J. (2003). Novice Driver Training Results And Experience With A PC Based Simulator. *PROCEEDINGS of the Second International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp165-169.

Allen R.W., Rosenthal T.J., Chrstos J.P., Aponso B.L. (2004). A Driving Simulator for Testing the Visibility and Conspicuity of Highway Designs and Traffic Control Device Placement, Proceedings of the 83rd Annual Meeting of the Transportation Research Board, *Washington, D.C.*

Lengenfelder J., Schultheis M.T., Al-Shihabi T., Mourant R.R., DeLuca J. (2002). Divided Attention and Driving: A Pilot Study Using Virtual Reality Technology. *Journal Of Head Trauma Rehabilitation*, pp26-37.

Papelis Y., Ahmad O., Watson G. (2003). Developing Scenarios To Determine Effects Of Driver Performance: Techniques For Authoring And Lessons Learned. *DSC North America 2003 Proceedings*, Dearborn, Michigan.

Papelis, Y., Ahmad O., Schikore M. (2001). Scenario Definition and Control for the National Advanced Driving Simulator. *ESV: 17th International Technical Conference on the Enhanced Safety of Vehicles*, Report No. 346. Amsterdam, Netherlands.

Rizzo M., Severson J., Crèmer J., Price K. (2003). An Abstract Virtual Environment Tool to Assess Decision-Making in Impaired Drivers. *PROCEEDINGS of the Second International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp40-47.

Steed A., Frécon E. (1999). Building and Supporting a Large-Scale Collaborative Virtual Environment. *Proceedings of 6th UKVRSIG*, pp59-69.