

[**Three-Dimensional Visualization and Animation**](https://fenix.tecnico.ulisboa.pt/disciplinas/AVT351795/2017-2018/1-semestre)

1st Semester 2017/2018



3º Assignment

Project Report

Group 13:

78414 Ricardo Silva

78516 João Loureiro

# Introduction

For this third assignment we migrated our Micromachines application from OpenGL to WebGL. We did everything in WebGL without using the Three.js framework.

We started by migrating and implementing the features we already had in the OpenGL application starting with the basic: the cameras, lights, object movement and collisions.  
We then implemented the techniques we already had: the transparent objects, fog, particle system, 2D lens flare and billboards.

After that we implemented the features specific to this assignment:

1. Bump-Mapping
2. Planar Shadows
3. Planar Reflections
4. Animation
5. Stereo Vision
6. Accelerometer

As explained in the previous report:

* To create the track objects’ position we wrote a script to create a file with the specified positions which in then parsed in our application;
* To texture the table we implemented multi texturing using two textures that are mixed according to a black and red mask;

The link to the sigma machine is the following: <http://web.ist.utl.pt/~ist178414/AVT-WebGL/>  
Or use the following qr code:



# Bump-Mapping

htfrhhtrhrt

## Method

gedrthrethrtyhrt

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – Bamboo table texture | Fig. x – Wood table texture |

# Planar Shadows

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Method

FillerFillerFillerFillerFillerFillerFillerFiller

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – TODO | Fig. x – TODO |

# Planar Reflection

We implemented the planar reflection in the rear-view mirror inside the car, using a stencil.

## Method

To accomplish this goal, we used an extra camera located inside the car and oriented towards the road behind the car. First we render the scene as usual, then we render the mirror with the stencil testing active and finally we render the scene using the backwards camera. By using this technique, we are able to create the illusion of a reflection in the rear-view mirror.

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – View from inside the car | Fig. x – Another view from inside the car |

# Animation

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Method

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – TODO | Fig. x – TODO |

# Stereo Vision

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Method

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – Stereo vision from outside the car | Fig. x – Stereo vision from inside the car |

# Accelerometer

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Method

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Results

|  |  |
| --- | --- |
|  |  |
| Fig. x – TODO | Fig. x – TODO |

# Extras

FillerFillerFillerFiller FillerFillerFillerFiller FillerFillerFillerFiller

## Weather

To add to the atmosphere of the game we change the colour of the sky depending on the current time of the day and weather using the clearing colour of WebGL.

## Scoring System and finishing Line

Our scoring system consists of tracking the time a player takes to complete a lap around the track. The best time and the current time is displayed in the tittle of the window. To complete the system, we added a finishing line, using a texture, to the place where the game starts and finishes counting the time. //TODO TODO TODO

|  |
| --- |
|  |
| Fig. 18 – Finishing Line |

## Custom .obj and .mtl loader

To load .obj models we wrote a custom loader than parses .obj files and the corresponding .mtl files and creates the meshes in our application. The loader retrieves all the necessary elements (vertices, normals, texture coordinates and material) and then creates the VBOs and VAOs necessary to draw the meshes. //TODO TODO TODO

# Conclusion

We completed the project successfully by migrating all the objects/effects, plus some additional effects to complement the game, from OpenGL to WebGL and implementing the six new requirements. During the development of this project we acquired new skills and had the opportunity to learn more about WebGL.