GAMETOOLS

ADVANCED TOOLS FOR DEVELOPING HIGHLY REALISTIC COMPUTER GAMES

TEST REPORT ON RELEASE PLUG-INS PROTOTYPES

Document identifier:	**GameTools-6-D6.2.1-02-0-1-Project integration, evaluation and test**
Date: (use "update field" Word function, right mouse button)	**04/10/2006**
Work package:	**WP06: Project Integration, evaluation and test** (e.g.: WP6: Integration, Test)
Partner(s):	**All**
Leading Partner:	**INFOWERK**
Document status:	**DRAFT**
Deliverable identifier:	**D6.2.1**

Abstract: The test report on release plug-ins prototypes covers the matching of the requirements and the functionality, the performance of the plug-ins and the assessment of the increase in realism of the resulting 3D – Applications. Furthermore it gives some insight of the ease of integration of each workpackage.

Delivery Slip

	Name	Partner	Date	Signature
From	Martin Kolb	Infowerk		
Reviewed by	Moderator and reviewers			
Approved by				

Document Log

Issue	Date	Comment	Author
0-0		First draft	Martin Kolb
0-1		Gedas second draft	David Gutiérrez

Document Change Record

Issue	Item	Reason for Change

Files

Software Products	User files / URL
Word	gametools-6-d6.2.1-02-2-0-project integration, evaluation and test.doc (use “update field” Word function)

CONTENT

1. INTRODUCTION

1.1. OBJECTIVES OF THIS DOCUMENT

This document is the second deliverable of the “Project integration, evaluation and test” work package. Its aims are to summarize the results in requirement fulfilling, performance, increase of realism and ease of integration. These results were originated by each company partner in their own testing conditions, which will be listed in particular.

1.2. APPLICATION AREA

The testing environment at InfoWerk GmbH has the following specifications:

Pentium D 3.2 GHz

GeForce 7900 GS with 256MB

The testing environment at Gedas Iberia (a member T-Systems Iberia) has the following specification

AMD X2 4800 2Gb RAM

GeForce 7900 GTX with 512Mb

1.3. APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS

Applicable documents

[A1]

Reference documents

[R1]

1.4. DOCUMENT AMENDMENT PROCEDURE

1.5. TERMINOLOGY

Glossary

WP	Workpackage
OGRE	Open Graphics Rendering Engine
DLL	Dynamic Link Library

2. EXECUTIVE SUMMARY:

All work packages work quite well on their own concerning performance and/or increase of realism. However, I was not able to get the work packages written for the same graphics API to run together and referring to some conversations with some of the work package members, there weren't attempts either. Since most of the game developing companies surely would use all the frameworks it would be a hard requirement to make them usable together.

In fact some of the packages are written using OpenGL and some of them are written using Direct3D so is impossible to use all of them at once

3. WP3: VISIBILITY

3.1. MATCHING BETWEEN REQUIREMENTS AND FUNCTIONALITY

Referring to the results created in our test environment given by the DirectX™ tool PIX and the summary given by the OGRE the matching between requirements and functionality is fulfilled.

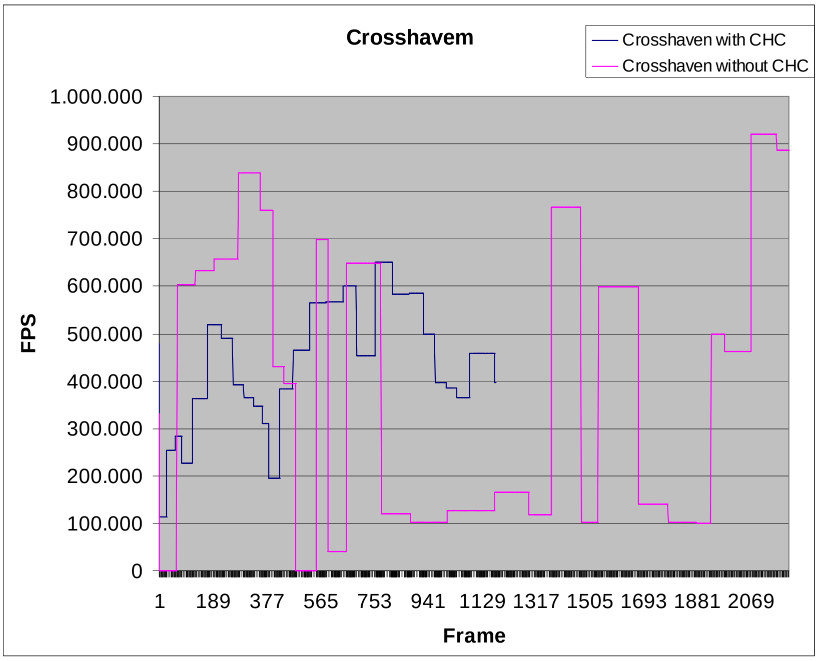
The algorithm works properly and exposes its advantages in appropriate environments. A suggestion would be a more flexible algorithm selection mechanism reacting to the environment and the objects present in the view frustum.

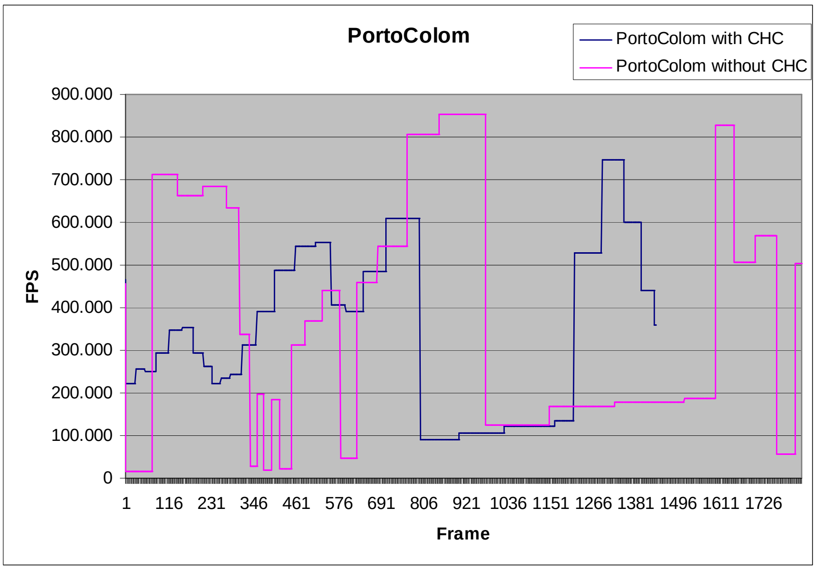
However, the visibility work package was not runnable together with the illumination package.

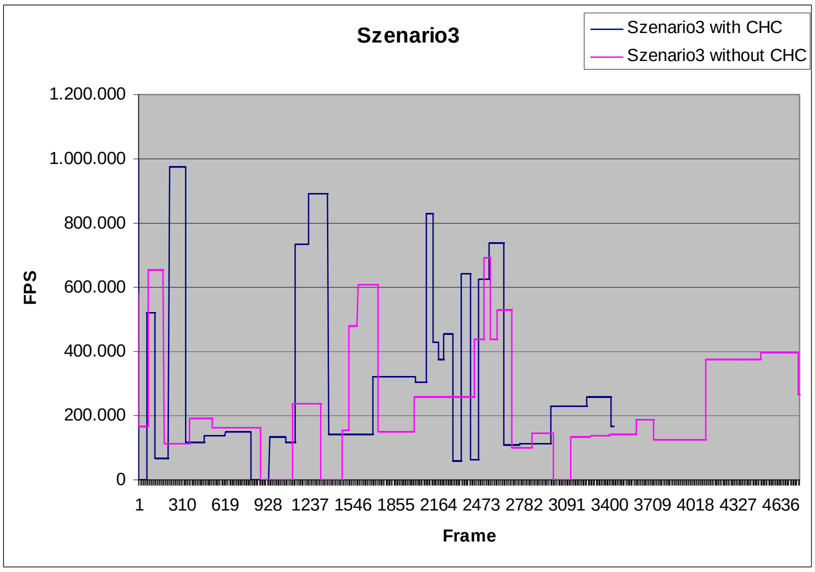
Gedas has not used this module yet.

3.2. PERFORMANCE

Below some test results are mentioned:







3.3. INCREASE OF REALISM

Since this WP3 is fully performance-related the increase of realism is out of the business of this WP.

3.4. EASE OF INTEGRATION

Integration of WP3 is quite easy because of the very open SceneManager system the OGRE provides. After compiling the OGRE solution file with the pre-processor flag GTP_VISIBILITY_MODIFIED_OGRE and the changes made in the OGRE rendering system you can go on with integration of the WP in your project. You just have to register your

SceneManager plugin per DLL and call Ogre::Root::createSceneManager(const string& TypeName) to plug the OcclusionCullingSceneManager into the OGRE rendering system. Configuring the compilation by setting a pre-processor directive is a little bit inconvenient since you have to recompile every other plug-in concerning vital parts of OGRE since the change in object instance size would result in read-write exceptions in run-time.

4. WP4: GEOMETRY

4.1. MATCHING BETWEEN REQUIREMENTS AND FUNCTIONALITY

4.2. PERFORMANCE

Unfortunately we here at InfoWerk weren't able to test the performance in out testing environments, since our meshes were inappropriate to simplify with the GeoTool. The example application worked really good, especially in the LodStrips demo the framerate never dropped below 150 fps.

Gedas has just recently integrated this work package into their engine. They have no benchmark yet because there have been some little problems with the integration and it has worked just mere weeks ago of the writing of this document. With only the first tests, we can say that for the moment it seems that the raw number of polygons from our scenes will be incremented in more or less 60% without losing any performance.

4.3. INCREASE OF REALISM

Since this work package is more performance related, nothing has to be said about increase of realism. However, the major difference to ordinary LOD techniques is, that the popping effect with this LOD technique is not as a recognizable as in normal techniques.

4.4. EASE OF INTEGRATION/USE

GeoTool:

The GeoTool is used to generate the triangle strips called LOD technique and to save the thereby generated meshes. Although the GeoTool seemed to work with most of the configurations tested, it sometimes crashed with any obvious reason. Our meshes at InfoWerk were not usable at all, since it is a hard requirement to use manifold meshes only to generate the triangle strips. However, it worked quite smooth with the models deployed by the work package team.

Gedas has not used the GeoTool package. It has integrated the mesh reduction generation directly into its own editor. It was very easy to integrate into their engine since it was more a work of translation. All the work was about communicating correctly with the library. It was necessary to translate the data to the library structs and translate back the results of the library to the editor classes.

5. WP5: ILLUMINATION

5.1. MATCHING BETWEEN REQUIREMENTS AND FUNCTIONALITY

Matching between requirements and functionality was partly satisfied. The work package teams should put more emphasis on getting their frameworks to work together.

5.2. PERFORMANCE

Glow:

Since the glow effect is a post-processing effect, it is not such a big issue concerning performance for regular PC/Console gaming resolutions.

In Gedas experience it is possible to have some issues when you use very big screen resolutions. In the Gedas reality centre software has to render at resolutions like 2560x1024, and then effects like HDR and Glow produce some framerate drops.

Environment mapping:

Environment reflective and refractive mapping, especially in a dynamic manner, is a really performance decreasing approach to put a realistic impression into practice. However the work package team did a really good job in optimizing performance of this render technique. But also the user can optimise performance with reducing texture size of the cube maps and using material LODs, a feature from the OGRE.

Obscurances:

Since the obscurances are generated as a pre-process, there is no big issue concerning time generation. To use the obscurances when you draw you only have to modify your draw function to use another texture. It could have some performance drops if the scene use more textures than the RAM of the graphic card, so the performance in this case is memory dependent.

5.3. INCREASE OF REALISM

Below are a few screenshots of our demonstrator application using some of the effects developed by this WP. By using approximate respectively physical models for calculating the output textures the effects are convincing and the increase of realism can be - depending on the artwork of course – very high.

Glow:

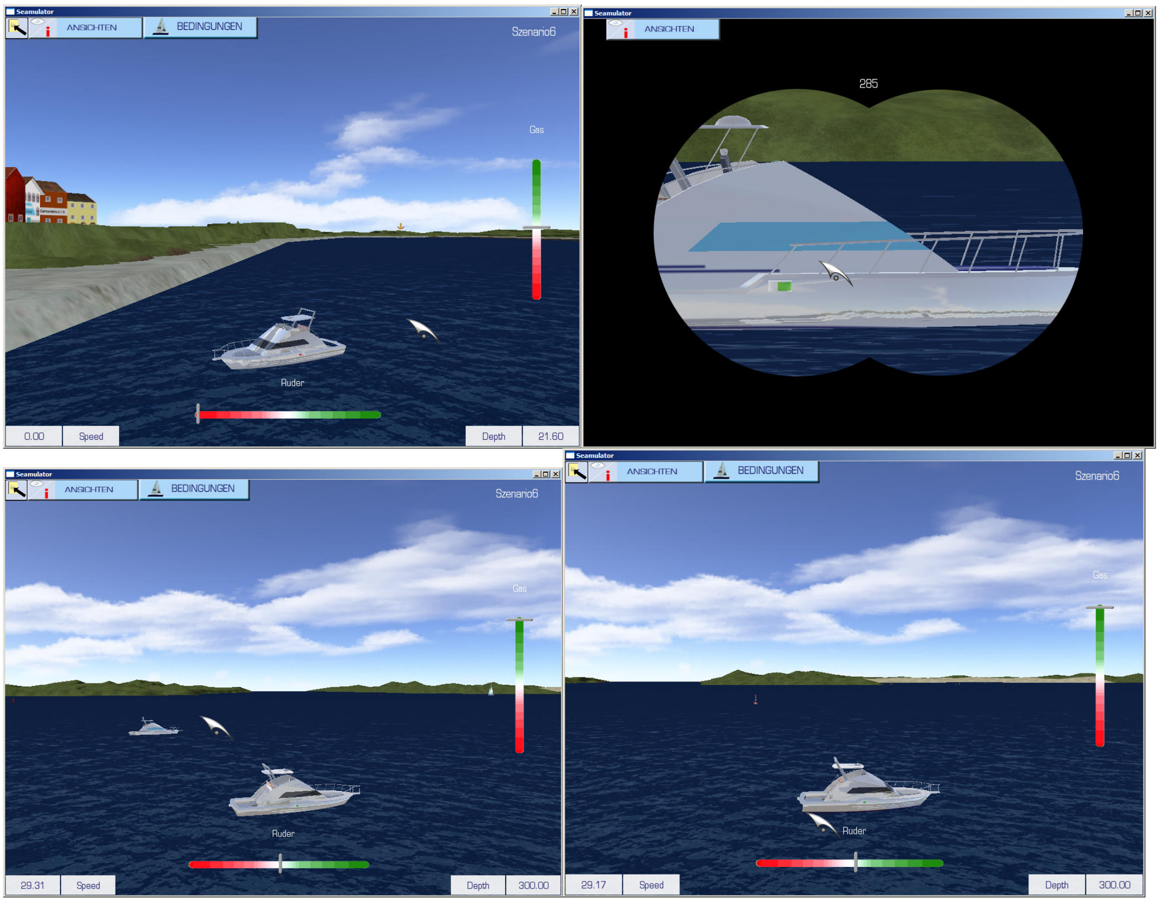
The glow effect satisfies the demand for “simulating” the reaction of the eye respectively its pupil to light. The effect in WP5 provides a convincing result concerning realism and performance.

Screenshot:

Environment mapping:

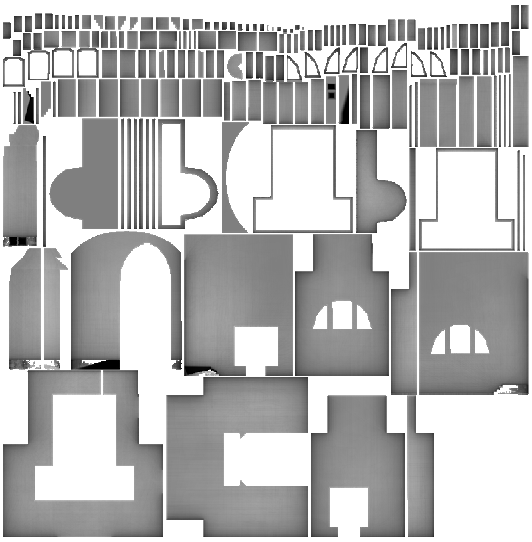
The Environment mapping technique increases realism for reflective and refractive materials, like glass, plain metals and plastic. Since ships and boats in the leisure sector are often made of plain and glossy plastics this technique was quite appropriate for our uses. We had to modify the shader program a little to blend it with the original colour texture. Also a specular map would be usable to define the amount of reflection.

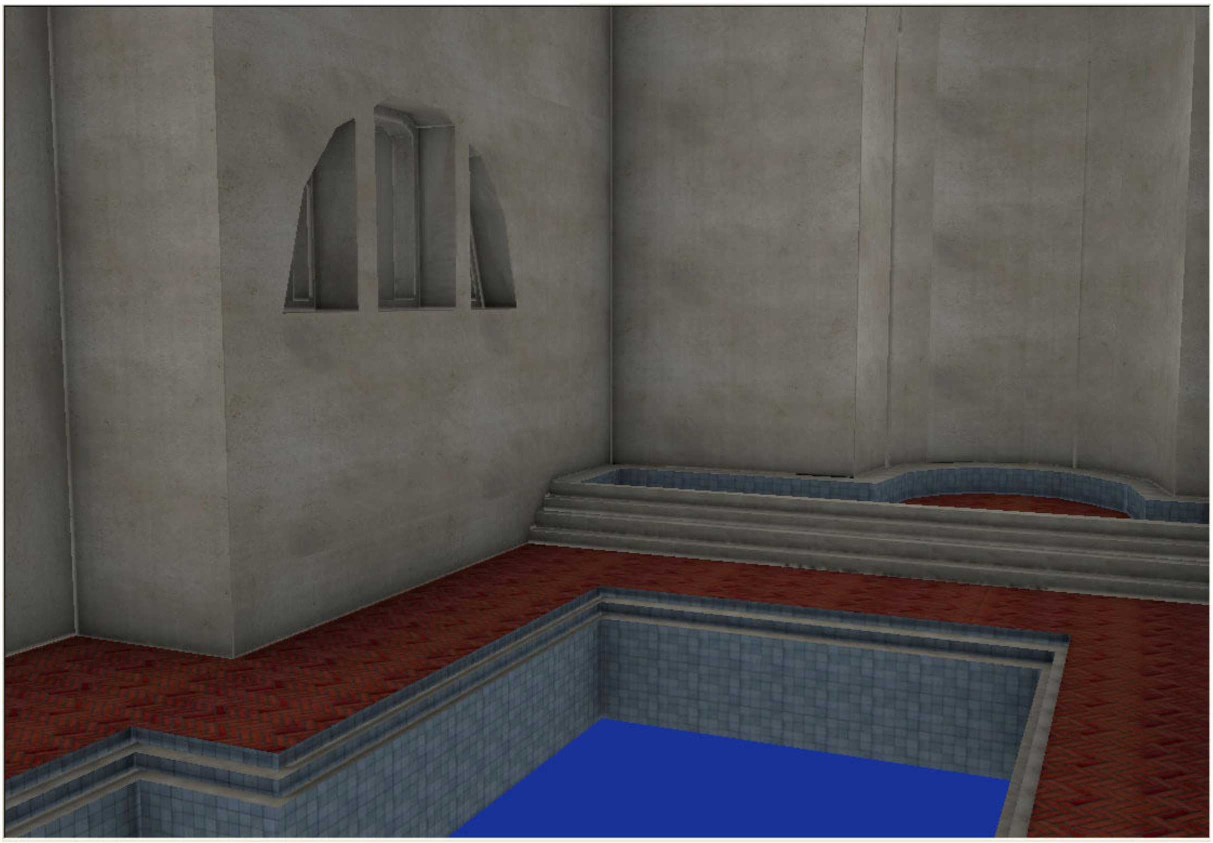
Screenshots:



Obscurances:

Obscurances fake the light reflection between all diffuse objects. We use this texture as the ambient element of the scene. Then in a shader we add this contribution to the diffuse component.





In this screenshot we only see the obscurance with the diffuse texture, there isn't any light in the scene.

5.4. EASE OF INTEGRATION

Since the OGRE has a very flexible material and resource system, it is quite easy to plug new effects into the engine. Compilation of the necessary source files happens like in WP3 with conditional compilation and pre-processor directives. The effects can be easily set up with the .material scripts the OGRE already uses. The glow effect, for example, is a post-processing effect exploiting the OGRE's compositor framework, which are, like material scripts, a convenient method to define effects like bloom, hdr rendering and various filters like an "old tv" effect. The environment mapping technique was also quite ease to integrate since it also uses the common material framework of the OGRE. The shader program needed a few changes to blend the reflective colour with the base texture colour. There was added an "IllumTechniques" section to the material framework.

Unfortunately the illumination work package did not work together with the visibility work package. It would be a hard request to get both work packages to work together.

6. ANNEXES