

# **D7.1 Interim Virtual Studio Pipeline Report**

# PRESENT

Grant Agreement nr	856879	
Project acronym	PRESENT	
Project start date (duration)	January 1st 2018 (36 months)	
Document due:	28/02/2021	
Actual delivery date	26/02/2021	
Leader	Brainstorm	
Reply to	jmontesa@brainstorm3d.com	
Document status	Final	





# Project funded by H2020 from the European Commission

Project ref. no.	856879
Project acronym	PRESENT
Project full title	Photoreal REaltime Sentient ENTity
Document name	Interim Virtual Studio Pipeline Report
Security (distribution level)	Confidential
Contractual date of delivery	28/02/2021
Actual date of delivery	26/02/2021
Deliverable name	D7.1 Interim Virtual Studio Pipeline Report
Туре	Report
Status & version	Submission Version
Number of pages	43
WP / Task responsible	Brainstorm
Other contributors	
Author(s)	Javier Montesa





EC Project Officer	Ms. Adelina Cornelia DINU Adelina-Cornelia.DINU@ec.europa.eu
Abstract	Current works on the Virtual Studio pipeline focus on the adaptation of InfinitySet for the incorporation and use of Unreal Engine and on the implementation of the required modules in Unreal Engine to allow communication between the two graphic engines.
	This document details the architecture design for this integration, the functionality of each of its modules, and the required implementation to cover the requirements and needs identified in the broadcast use case described in WP2.
Keywords	UNREAL ENGINE, INFINITYSET, INTEGRATION, RENDER BUFFERS, PROTOCOL
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	No
Read by partners	No
Mgt. Board approval	No





# **Document History**

.

Version and date	Reason for Change
0.1 02-02-2021	Document created by Javier Montesa
0.2 05-02-2021	Introduced abstract, executive summary, structure and table of contents
0.3 12-02-2021	System architecture section edited
0.4 19-02-2021	Unreal Engine Launcher, InfinitySet Adaptations, and Unreal Engine Plug-in edited.
0.5 23-02-2021	Conclusions, next steps and document revision
0.9 24-02-2021	Version for internal review
1.0 26-02-2021	Revisions in response to review: final version submitted to Commission





# **Table of Contents**

1. EXECUTIVE SUMMARY	6
2. BACKGROUND	6
3. INTRODUCTION	6
3.1. Main objectives and goals	7
3.2. Terminology	7
4. SYSTEM ARCHITECTURE	8
4.1. Unreal Engine Modules	9
4.1.1. Unreal Engine VideoIn	9
4.1.2. Unreal Engine VideoOut	10
4.1.3. Unreal Engine Control Server	10
4.1.4. InfinitySet Control Client	10
4.2. InfinitySet Modules	10
4.2.1. Unreal Engine Launcher	10
4.2.2. InfinitySet VideoOut	10
4.2.3. InfinitySet VideoIn	11
4.2.4. Unreal Engine Control Client	11
4.2.5. InfinitySet Control Server	11
4.3. Technologies and protocols	11
4.3.1. Web Remote Control	11
4.3.2. Shared Memory	12
4.3.3. Brainstorm API	12
5. UNREAL ENGINE LAUNCHER	13
5.1. Functionality	13
5.2. Graphic User Interface	13
5.2.1. "La unch as" selector	13
5.2.2. Project	14
5.2.3. Stats	14
5.2.4. Quality	15
5.2.5. Camera	15
5.2.6. Transfers	15
5.2.7. UE Plugins	15
6. INFINITYSET ADAPTATIONS	15
6.1. External graphics engine manager	15
6.2. InfinitySet VideoOut	16
6.3. InfinitySet VideoIn	16
6.4. Unreal Engine Control Client	17
6.5. InfinitySet Control Server	17





7. UNREAL ENGINE PLUG-IN	
7.1. Unreal Engine Control Server	18
7.2. InfinitySet Control Client	18
7.3. Unreal Engine VideoIn	18
7.4. Unreal Engine VideoOut	19
7.4.1. Main viewport vs. Scene capture	19
7.4.2. Camera Switching	20
7.4.2.1. Recommended workflow. Only main viewport, Cut without preview	20
7.4.2.2. Multiple renders. One main viewport, multiple scene captures.	20
7.4.3. Unreal Engine Video Capture and Delivery	20
8. CONCLUSION	20





# 1. EXECUTIVE SUMMARY

This deliverable covers the activities related to task 7.1 where the integration of InfinitySet and Unreal Engine is being carried out. The two main objectives in this task are first to prepare both engines so they can run together in an easy and seamless way for the operator, and second to connect them so they can share parameters and events, and can stream their renders in real time and in both directions through shared memory.

The document will first introduce the main concepts and overall architecture of the system, then the aforementioned two main challenges will be presented along with details on how they are being achieved, and finally the current functionalities of the system will be introduced and a brief manual on how to make use of them will be added as an annex to the document. The conclusions section, apart from presenting the current development status of the system, will also detail the planned activities still pending to achieve complete functionality.

## 2. BACKGROUND

Two of the most important outcomes of WP2 have been the description of the different use cases and scenarios and the overall architecture of the system and its different implementations for each use case, including a reference implementation to facilitate the technology providers to implement and integrate their modules.

Depending on the specific use case requirements, the Present general architecture may involve different modules and functionalities, but for the development and integration of each of its modules, a reference implementation including all the possible functionalities and requirements was proposed.

Therefore the broadcast use case has been based on the said reference implementation, and the activities and developments made in task 7.1 have been oriented to the integration of InfinitySet and Unreal Engine in such a way that along with the developments carried out in task 7.2, the resulting piece of software will assure compatibility with the rest of systems in Present, while providing all the required functionality to perform the proof of concept tests coming shortly in the project time plan.

Specifically the design of those modules intended to share parameters and events between the two graphic engines have been adjusted from their initial designs and now they are already functional and fit the proposed general architecture in WP2 and its reference implementation.

## 3. INTRODUCTION

One of the main aims on the integration of InfinitySet and Unreal Engine has been to make the system quick to understand and easy to manage by operators. In order to achieve this objective, the plan was to make it possible to put all the systems up and running just from the user interface included InfinitySet, the engine would connect to the so called Brainstorm Unreal Player, a standalone application that once run from InfinitySet would be capable to start the Unreal Engine and load the required project.

During the project design and implementation, this Brainstorm Unreal Player application was found to be possible to be built-in as one more module in InfinitySet, so the now the current design has no pieces but just the two engines. InfinitySet is capable to start Unreal Engine, load the required plug-ins in order to enable bidirectional communication and finally load the required Unreal Engine project along with its specific required plug-ins. The system is now much faster to operate and easy to understand, and its architecture is also easier to explain and detail.