DIGITAL TWINS OF THE OCEAN: AUTOGENERATED 3D ENVIRONMENTS FOR VALIDATING OFFSHORE WIND FARM OPERATIONS

Carlos Pereira Santos *and* Harald Warmelink
Breda University of Applied Sciences, Breda, the Netherlands
e-mail: santos.c@buas.nl

Martin Courchesne
CEA Saclay NanoInnov, Gif-sur-Yvette, France

Within the context of the Iliad project, the authors present technical challenges and the first results of having valid 3D scenes of (non-)existing offshore wind farms procedurally and automatically generated within either the Unreal or Unity game engine. The Iliad – Digital Twins of the Ocean project (EU Horizon 2020) aims to develop a 'system of systems' for creating cutting-edge digital twins of specific sea and ocean areas for diverse purposes related to their sustainable use and protection. One of the Iliad pilots addresses the topic of offshore floating wind farm construction or maintenance scenario testing and validation using the Unity 3D game engine. This work will speed up the development of these scenarios by procedurally and automatically creating the Uni-ty 3D scene rather than manually (which is done at present). The main technical challenges concern the data-driven approach, in which a JSON configuration file drives the scene creation. The first results show a base wind farm running in Unreal 5.1. The final product will be able to handle environmental conditions, biological conditions and specific human activities as input parameters.

Keywords: *ocean digital twin, procedural content generation, 3D scene generation, offshore wind farm, game engines*

1. Introduction

The Iliad – Digital Twin of the Ocean project (EU Horizon 2020) aims to develop a 'system of systems' for creating cutting-edge digital twins of specific sea and ocean areas for diverse purposes related to their sustainable use and protection. The project will fuse a large volume of data in a semantically rich and data-agnostic approach to enable simultaneous communication with real-world systems and models. Ontologies and a standard style-layered descriptor will facilitate semantic information and intuitive discovery of underlying information and knowledge to provide a seamless experience. To develop and demonstrate its 'system of systems', the Iliad project relies strongly on 20+ pilots, i.e., actual ocean digital twin in-stances at specific areas for specific purposes [1].

One Iliad pilot concerns the creation of a 3D interactive digital twin of a part of an offshore floating wind farm to test and ultimately validate specific complex construction or maintenance operations scenarios under particular conditions. This could be an existing or soon-to-be-developed farm. The main conditions of interest are weather conditions, which could entail current and extreme conditions observed over a certain time. In terms of scenarios, this pilot will typically deal with construction ships operating large and heavy cranes and other specialised construction equipment to install or upgrade wind turbine components or entire turbines. Since both the vessel and the turbine are floating, wave dynamics are highly important, and thus high-fidelity wave and related or responding physics simulations are paramount. The digital twin would be used by concept engineers and operators to validate and optimise the full operations ahead of time and to define operational windows related to environmental conditions. The primary and practically sole concern is the safety of involved personnel and assets.

One Iliad partner is CEA Saclay NanoInnov (The French Alternative Energies and Atomic Energy Commission). Based in France, CEA is a key player in several areas of research, development and innovation. In particular, the Interactive Simulation Lab develops interactive digital twins for industrial applications. As explained, CEA already uses and applies the Unity game engine to develop, test, and ultimately validate offshore floating wind farm operation scenarios. As such, they are the pilot's leader within the Iliad project. The Unity 3D scene is currently developed manually.

Another Iliad partner involved in this pilot is Breda University of Applied Sciences (BUas). Based in Breda, the Netherlands, BUas' Academy of AI, Games Media offers internationally highly regarded educational and research programmes with innovative technologies (particularly game and media). Over the past 10+ years, several BUas' Games staff developed more and more expertise with procedural and automatic generation of 3D environments using game engines, particularly Unreal, but also Unity. With this expertise and continued interest, BUas sees an opportunity to develop a key system for the Iliad pilot on offshore floating wind farm operation validation, as well as within the whole Iliad 'system of systems' for use by any other interested pilot or future digital twin of the ocean.

In this extended abstract, the authors explain and demonstrate through early prototypes the approach they are currently taking for the procedural and automatic generation of a 3D digital twin of a (non-)existing offshore wind farm with certain parameters working in the Unreal or Unity game engines. Specifically, they describe the main technical challenges in their prototyping with both Unreal and Unity. They subsequently show and briefly explain the first screenshots of a working prototype.

2. Technical Challenges

The main objective of our development work is to be able to have valid 3D scenes of (non-)existing offshore wind farms procedurally and automatically generated within either the Unreal or Unity game engine for a certain/limited set of subsequent use cases, first and foremost for our partner CEA to open up in the Unity editor itself so they can further develop and test the operation scenarios (with limited to no manual 3D scene editing required). Procedurally generated refers to the algorithmic generation of valid 3D content without precise definitions because the algorithms already possess or apply them. This approach leads to diverse results due to the multiple outcomes produced by the parametric algorithms.

The classic example in procedural 3D content generation is the creation of any physically valid bridge crossing a river. An example more relevant to this pilot could concern the generated weather. On the other hand, automatically generated pertains to generating more specific content with no requirement or intention for multiple outcomes or randomness. An example relevant to this pilot would be the use and placement of a 3D model of a wind turbine.

Given this main objective and taking an agile/iterative approach to the development of this system, we are currently working on the following main technical challenges:

- Defining and prioritising relevant input parameters of different kinds with different units of measurement in an appropriate format of ideally relevant and recent standards. Think of the shape of the wind farm area to be rendered, the number of turbines to position in that area with a certain distance between them, the depth of the sea in that area, etc. Also, think of the existing (GIS) standards for defining each input parameter and the total input parameter set.
- Defining the acceptance levels of the Unity or Unreal scene from audio-visual, user interaction, and coding perspectives. Think of answering the following questions:
  - Will it actually work for generating a scene for particular use cases? More specifically, will the end-user be able to at least explore the 3D scene, also using an appropriate VR headset? Or will the developer wanting to use the created scene be able to subsequently develop the operations scenario with limited to no manual scene adjustments?
  - Will the generated scene actually have the desired quality? More specifically, what quality should the water system, 3D models, animations, etc. attain?
- Developing the actual code to procedurally and automatically generate the Unity or Unreal scene from the input parameters mentioned above at the acceptance levels mentioned above. This challenge also covers dealing with assets, notably 3D model files, audio files, game engine plug-ins or extensions, and any licensing

issues involved.

- Validating the resulting Unity or Unreal scene:
  - o Internally, checking all input parameters to the scene output, especially concerning procedural generation techniques.
  - Externally, by having the automatic procedural generation process be evaluated and tested within their work and the scope of Iliad project, and reviewing and assessing what are the automatic generation and manual adjustments required. Second, by looking for other potential use cases and user feedback on how to apply our software to generate different scenarios and possible applications for offshore wind farms.

3. First Results

In terms of initial input parameters design, we are currently working with the following sets:

- Environmental conditions. Realistic depiction of the situation in a location: bathymetry, sediment layer, wave and weather conditions.
- Biological conditions: Realistic depiction of the fauna and flora based on biology charts and bio-mass information.
- Human Activities: Overview of human-related actions or operations in depicted marine regions. These activities encompass a wide range of human interactions.

To ensure that our work remains within scope, we only consider higher biological groups and specific sectors of human activities. Specifically, we are focusing on wind farms and shipping. Wind farms provide information on energy production using wind turbines, cabling, energy production, and ecological pressures such as noise. The shipping sector includes representation of shipping corridors, traffic, and shipping vessels. We have developed a first/preliminary version of a JSON schema with which to define these kinds of parameters. We subsequently wrote the code for Unreal 5.1 to read and interpret the JSON schema to generate the working scene (Fig. 1). Procedural aspects of the resulting 3D scene have so far concerned how to appropriately animate a 3D asset such as a fish or how to achieve a buoyancy effect with floating as-sets (notably ships, of course).

REFERENCES

1. , (2023), *Iliad - Digital Twins of the Ocean*. https://www.ocean-twin.eu/, lastaccessed31May.

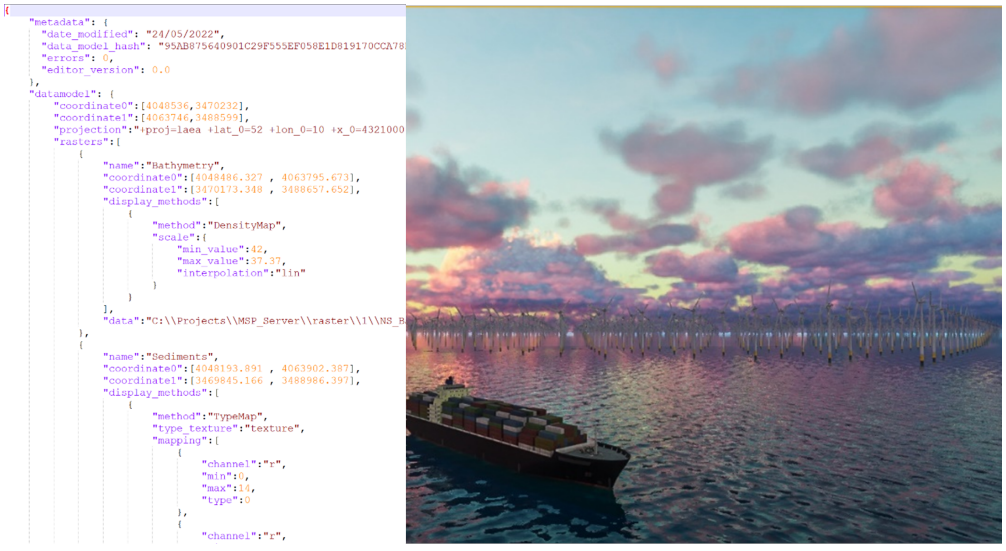

Figure 1: Excerpt of a JSON configuration file (left), leading to an Unreal scene (right).