UnRealWorld: A Crowdsourcing and 3D Development Platform for Cultural Heritage

Andrea Barillari   Daniele Bernardini
Intranet Standard GmbH, Munich, Germany

Pierluigi Crescenzi
Università di Firenze, Italy

1   Introduction

Crowdsourcing is known as the process of designing and developing a web-based software platform and obtaining content for it through the contributions of a community, including people who are not part of the entity that conceived the platform itself. One of the most popular crowdsourcing platforms is Wikipedia, an online encyclopaedia, which proved during the years that the crowdsourcing model works efficiently without any significant loss of quality (indeed the average article quality increases as it goes through iterations of edits [3]). Another well-known example of a very successful crowdsourcing platform is YouTube, which currently allows billions of users to share originally-created videos [1].

On the other hand, in the last twenty years we have seen an impressive development of 3D game engines, starting from the Doom Engine, produced in the early nineties, which really ushered in a new era of game design and play, arriving to the third version of Unreal Engine, presented in 2004, which was a groundbreaking engine capable of handling high quality graphics and powerful user interface while maintaining good usability (see, for example, [4]). All these game engines, however, were quite low-level tools and most of them were proprietary software, meaning that a "normal" user could not easily master them in order to produce new content. This drastically changed in the last couple of years with the introduction of the latest modern state-of-the-art game engines, such Unity 5 and Unreal Engine 4. These kinds of engines are currently used by millions of game developers [5] (although they also support functionalities that can be used for non-gaming purposes).

The goal of this paper is to propose to applying the crowdsourcing approach to a software platform that uses a modern and state-of-the-art 3D game engine, in order to allow for the generation and manipulation of interactive 3D environments by a community of users, to produce cultural heritage content (see Figure 1). This platform should easily host virtual reconstructions of archaeological sites and monuments, historical buildings and, in general, works of art. These reconstructions should not only be a virtual representation of the current status, but also a representation of their original status. The platform should work with virtual environments that will be completely explorable in first and third person (for instance, using avatars). The environments should also be populated with representations of the people living in a specific epoch. In the hands of expert historians, archaeologists and architects it should be possible to recreate high fidelity 3D reconstructions, which can be shared among academics to improve research work and course quality. With significant crowdsourcing support, different environments should be joined and theoretically the whole world (or at least the most "important" parts of it) could be virtualised and reconstructed. In other words, this platform should become a huge 3D cultural heritage encyclopaedia and repository. Moreover, as shown in Figure 1, the platform could be potentially applied to several other fields, such as, for example, education, tourism, and gaming.

As a first step towards this ambitious goal, and briefly described in the rest of this paper, is a standalone explicit crowdsourcing system, called *Unreal World*, which can be considered a *proof of concept* or a prototype implementation, whose purpose is to verify the system potentials. As a preliminary validation of this proof of concept, we will also shortly describe the contribution in terms of content development performed by a group of history researchers from the third University of Rome.
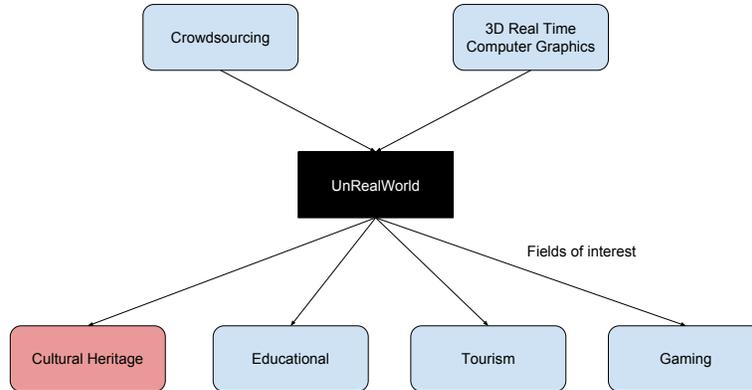
Figure 1: The UnRealWorld platform and possible applications

2 The Platform Architecture

The platform architecture is built using four fundamental elements: a client, an edit project, a server, and a database (in addition for Unreal Engine 4 [6] we also used the Django framework). The database (see the upper left part of Figure 2) contains all the content produced in the platform, which are stored as maps, and in turn contain blocks. The blocks are loaded dynamically at runtime, are independent of the maps, and can include UnReal assets, that is, resources of several types, such as 3D models, textures, animations, and blueprints (i.e., visual script objects). A versioning system has been implemented (see the upper mid part of Figure 2), so that multiple versions can correspond to each map and block. The versioning system is under the control of an administrator, who can approve and reject new versions. In this way, the user will always download (and substitute the current local version with) the latest version of a content. The users have to authenticate in order to modify the contents of the platform. This is done through an interaction with the UnRealWorld server (see upper right part of Figure 2). Currently, the platform includes three different roles: visitors (they can only view and interact with the maps), editors (they can modify the content of the blocks) and administrators (they build the maps, position the blocks, and approve the new versions). Once an editor is inside a map, they can ask to lock a specific block which is not already locked (see the lower left part Figure 2). Once the lock has been obtained, the editor is the only one who can modify the block. This operation is atomic in order to guarantee consistency between different modifications to the same content. Once the lock on a block has been obtained, an editor can download an edit project, which will allow them to modify the content of the block (see the lower mid part of Figure 2). The edit project is a custom project from the Unreal Engine 4 editor, which intuitively acts like a plugin for this editor. Finally (see the lower right part of Figure 2), once the UnRealWorld edit project has been opened in the Unreal Engine 4 editor, the user will authenticate with the UnRealWorld server in order to obtain the block whose content has to be modified. Once the modifications have been completed, the content of the UnRealWorld edit project will be exported by executing an upload to the the UnRealWorld server. The server controls the correctness of the new content and sends it to the database. Once an administrator approves the modifications, the new content will be available to all visitors and editors. A demo of all this process is available in the following YouTube video: `https://www.youtube.com/watch?v=lpeq6PccYGY`.

3 A Preliminary Experimental Evaluation

In order to simulate a crowdsourcing interaction, we collaborated with Giuseppe Ragone and his research group at the third University of Rome, which fit perfectly with our objective of having someone who might be a future contributor to the platform, while avoiding 3D modeling specialists. The goal of this collaboration
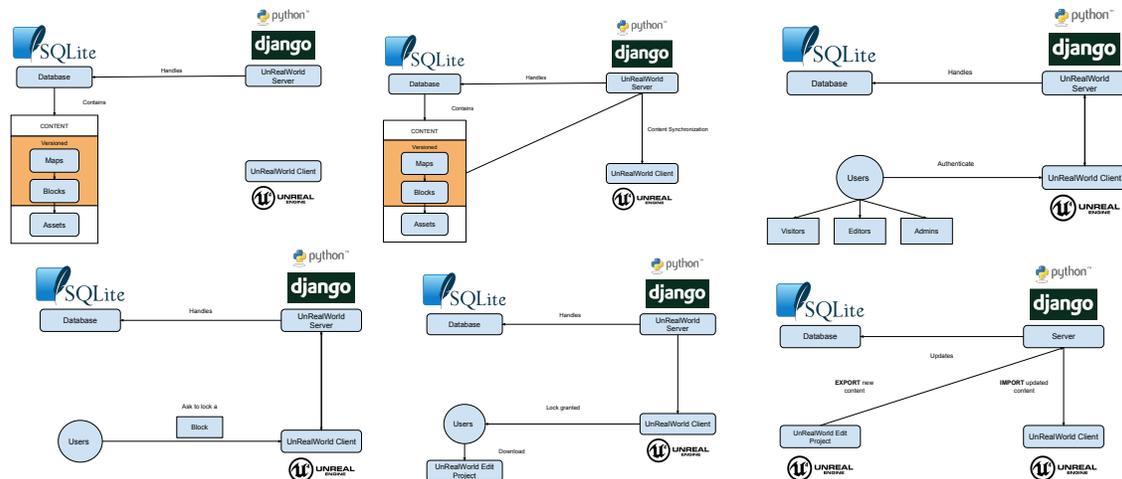
Figure 2: The UnRealWorld architecture

was to virtually reproduce a specific object from the *Cyme* archaeological site, an ancient Aeolian city situated in modern Aliaga, Turkey. The specific object is an exedra, a semicircular recess created to let people sit and talk. It could be considered the ancient greek version of a bench. This exedra was placed inside an agora, which is the central space in the city-states of Ancient Greece. Today the only part of this exedra that remains intact is its base. In order to reconstruct a realistic and plausible version of the exedra as it was in its original state, the research group used information and pictures of other undamaged exedras in their possession. The group modeled the exedra static mesh by using the additive geometry tools provided by the Unreal Engine 4 editor. The group followed a couple of short tutorial videos to learn additive geometry and how to use the fundamental controls of the editor. They did not require other additional external help or assistance in order to produce the static mesh. To increment the static mesh realism we decided to add some texture to it. Unreal Engine 4 uses a collection of textures to generate material, which is an asset whose purpose is to paint the mesh. Since we did not want a generic stone material, the group asked for the help of a graphic designer to create a collection of textures that when putting them together would resemble to the actual stone of the *Cyme* exedra. The final exedra static mesh was imported into the UnRealWorld platform and can now be seen in the demo video previously cited. Overall, this experience shows that the UnRealWorld usability is quite good. In particular, regarding the production and the editing of new content, the usability depends mostly on the usability of the Unreal Engine 4 editor, which in this case, turned out to be not so difficult for users that have no design or computer science skills.

References

[1] J. Burgess and J. Green. *YouTube: Online Video and Participatory Culture*. John Wiley & Sons, 2013.

[2] Django. `http://www.djangoproject.com`.

[3] S. Javanmardi and C. Lopes. Statistical measure of quality in wikipedia. *Proc. 1st Workshop on Social Media Analytics*, 132–138, 2010.

[4] M. Lewis and J. Jacobson. Game Engines in Scientific Research. *Communications of the ACM*, 45:1, 27–31, 2002.

[5] Unity Technologies. `http://unity3d.com/`.

[6] UnReal Engine Technology. `http://www.unrealengine.com/`.