Education for Chemical Engineers

A practical development of engineering simulation-assisted educational AR environments

--Manuscript Draft--

Manuscript Number:	ECE-D-20-DIG INV-00009
Article Type:	SI: Digitalisation Education - Tools for Sharing (Invited Authors Only)
Keywords:	chemical engineering, higher education, digitalization, augmented reality, engineering simulations, active learning environments
Corresponding Author:	Tom Van Gerven KU Leuven Leuven, BELGIUM
First Author:	Serkan Solmaz
Order of Authors:	Serkan Solmaz
	Jessica L. Dominguez Alfaro
	Pedro Santos
	Peter Van Puyvelde
	Tom Van Gerven
Abstract:	Engineering simulations have opened several gates for today's chemical engineers. They are powerful tools to provide technical content as physics-based numerical solvers. Augmented reality (AR) and virtual reality (VR), on the other hand, are already underway to digitize environments in many fields. The combination of AR/VR environments and simulations in engineering education has been attracting widespread interest. Literature has demonstrated a massive amount of educational digital environments in several contexts as being complementary to conventional educational methods. Nevertheless, hosting technical content produced by engineering simulations with educational AR/VR is still challenging and requires expertise from multiple disciplines throughout the technical development. Present work provides a facile and agile methodology for low-cost hardware but content-wise rich AR software development. A case study is developed to teach chemical-engineering concepts using a liquid-soap synthesis process. Accordingly, we assess and conclude the digital development process to guide unexperienced developers for the digitalization of teaching content. The present contribution serves as an example of the power of integrating AR/VR with traditional engineering simulations for educational purposes. The digital tool developed in this work is shared in the online version.
Suggested Reviewers:	Juan Garzon fgarzon@uco.edu.co
	Jorge Martin-Guttierez jmargu@ull.edu.es
	Eulogio Castro ecastro@ujaen.es
	Luis Reyes lh.reyes@uniandes.edu.co
Opposed Reviewers:	

Highlights

- Integration of AR/VR with engineering simulations for engineering education.
- A facile and agile development strategy was proposed for non-experts.
- Importance of interdisciplinary skills and collaboration in technical development.
- Enhancing operability of learning environments with AR and engineering simulations.
- Discussions on sustainability of long-lasting digital tools.

A practical development of engineering simulation-assisted educational AR environments

Serkan Solmaz[a], Jessica L. Dominguez Alfaro[a], Pedro Santos, Peter Van Puyvelde and Tom Van Gerven[b]

Department of Chemical Engineering, KU Leuven, Celestijnenlaan 200F, B-3001 Heverlee, Belgium

Abstract

Engineering simulations have opened several gates for today's chemical engineers. They are powerful tools to provide technical content as physics-based numerical solvers. Augmented reality (AR) and virtual reality (VR), on the other hand, are already underway to digitize environments in many fields. The combination of AR/VR environments and simulations in engineering education has been attracting widespread interest. Literature has demonstrated a massive amount of educational digital environments in several contexts as being complementary to conventional educational methods. Nevertheless, hosting technical content produced by engineering simulations with educational AR/VR is still challenging and requires expertise from multiple disciplines throughout the technical development. Present work provides a facile and agile methodology for low-cost hardware but content-wise rich AR software development. A case study is developed to teach chemical-engineering concepts using a liquid-soap synthesis process. Accordingly, we assess and conclude the digital development process to guide unexperienced developers for the digitalization of teaching content. The present contribution serves as an example of the power of integrating AR/VR with traditional engineering simulations for educational purposes. The digital tool developed in this work is shared in the online version.

Keywords: chemical engineering, higher education, digitalization, augmented reality, engineering simulations, active learning environments.

1. Introduction

The utilization of new digital technologies in education has been advocated in the scope of the European Commission in the Agenda for the Modernization of Europe's Higher Education Systems (European Commission, 2011). The progress of digitizing engineering sources, from research to education, has accelerated to find the best practices. Augmented reality (AR) and Virtual reality (VR) technologies have been gaining attention in this realm (Beck, 2019; Liu et al., 2017). Notably, interest of Chemical Engineering in multiphysics, multiscale and spatio-temporal

---

[a] These authors contributed equally to this work.

[b] Corresponding author (tom.vangerven@kuleuven.be)

phenomena has persuaded researchers to seek complementary tools. AR/VR technologies and engineering simulations have been proposed in a series of studies to prevail over shortcomings of traditional methodologies (Ge et al., 2011; Li, 2015; Powell, 2017; Zitney, 2010). However, many research questions are still posed on the technical content, infrastructure, accessibility, affordability, operability, and portability of digital resources (Akçayır and Akçayır, 2017; Beck, 2019). Essentially, arbitrary development procedures of AR/VR tools have generally resulted in single-use digital environments rather than long-lasting updatable applications (Beck, 2019; Porter and Heppelmann, 2017). In that sense, the digital transformation should not only be led by hardware and software but also by the development procedure, by which software and hardware are rigorously blended to provide a sustainable practice of teaching.

1.1. AR/VR technologies

In the new digitalized era, immersive technologies have been commonly used to provide new ways of approaching digital environments. While VR immerses the user in a fully digital environment, AR immerses the user in a blend of real and virtual elements. Since AR only complements rather than replaces reality, in general, VR is considered to be more immersive (Azuma et al., 2001; Brown et al., 2020).

For applications where total immersion is not well-justified, AR systems have proven to be useful. In general, AR systems are characterized by 1) a combination of real and virtual environments, 2) the alignment of digital content with the real world and 3) a real-time interaction (Azuma et al., 2001; Azuma, 1997). For an AR system to be conceived, it requires tracking, registration and visualization, which can be implemented in devices from the low-price range such as smartphones and tablets, to high-price range, like head-mounted displays (HMDs) (Liu et al., 2017). A detailed discussion of both AR and VR falls outside the scope of this paper. Hence, the discussion will be focused on AR, as it is the technology used in the case study.

Depending on the field of engineering, AR applications have demonstrated diverse purposes such as visualizing simulations results (Li et al., 2017), overlaying context-rich data (Salah et al., 2010) or enhancing on-site operations by providing real-time data (Heuveline et al., 2011). Although these applications benefit from the particular characteristics of AR systems, (Li et al., 2017) have identified three drawbacks that hinder its broader implementation. Firstly, the difficulty to transfer a given experience to different settings making them one-time use only. Secondly, the lack of support for having compatibility with several platforms. Finally, the interactions between the user and the experience have not been effectively and/or intuitively implemented and in most cases, the systems are merely for visualization rather than for interaction.

In recent years, there has been an increasing interest in immersive technologies to enhance teaching and learning in various domains. Specifically, for STEM education (science, technology, engineering and mathematics), some researchers have argued that AR has the potential to enhance learning. For example, AR enables the learner to interact with 3D virtual content from different perspectives, thus enhancing their understanding and improving their spatial abilities (Chen et al., 2011; Martín-Gutiérrez, 2017). Additionally, AR provides the possibility to visualize abstract concepts or unobservable (and invisible) scientific phenomena, such as magnetic fields or fluid dynamics (Wu et al., 2013) and allows learners to interact with the content when compared to simulations which offer an outside-observer perspective (Enyedy et al., 2012). Finally, several meta-analyses have concluded that AR can have a positive effect on student's academic performance, achievements and motivation, thus confirming some of the advantages of AR as an educational tool (Garzón et al., 2019; Garzón and Acevedo, 2019; Ozdemir et al., 2018; Santos et al., 2014; Tekedere, 2016; Yilmaz and Batdi, 2016).

Although the educational value of AR is confirmed, the technology also has its limitations. The most-reported challenges that hinder the implementation in education can be listed as follows: AR can be cognitive overloading, it can be cumbersome in the lack of well-designed interfaces, it can be distracting, and it can be hard to implement due to the inflexibility of the content (Bacca et al., 2014; 2014; Garzón et al., 2019; Wu et al., 2013). As a result, further interdisciplinary research, which takes these issues into account, is needed in order to develop qualified educational resources that leverage the advantages of AR (Garzón et al., 2019).

1.2. Engineering simulations

Computer-aided engineering (CAE) simulation tools have become vitally significant in support of engineering decisions. The use of simulation tools in engineering education is also getting prominent to provide required competencies and qualifications for future engineers. From fluid mechanics to process control, simulations can contribute in several fields to boost students' understanding of technical concepts. Multiphysics computational fluid dynamics (CFD) and process simulation tools are the most applied engineering software to research, design, and optimize chemical engineering problems. While engineering simulations can provide educational content to design learning experiences with easy-to-comprehend visualizations, they can also be utilized as computational tools to operate active learning environments, e.g. inquiry-based learning.

Multiphysics CFD simulations are powerful computational tools to solve engineering problems. They can be applied as either a substitute or complementary to experimental campaigns. CFD simulation tools require dedicated personal qualifications to prepare, run, and post-process simulation results. This can prevent lecturers from implementing these tools in their practices. Alternatively, a fully-integrated CFD simulation in AR/VR might present a remedy to prevail over this uneasiness. CFD results in AR/VR environments can also mitigate negative outcomes of complex visualization environments such as misinterpretation of results (Fukuda et al., 2018; Li et al., 2017; Lin et al., 2019).

The integration of CFD simulation results into AR/VR has been investigated following two main approaches to process CFD datasets: re-processing text-based CFD data extracts in the game engine (Berger and Cristie, 2015) and integrating CAD-based 3D visual post-processing extracts with the game engine (Zhu et al., 2019). Several studies have also proposed the update of results in the AR/VR environment by using dedicated cloud servers (Fukuda et al., 2019; Kim et al., 2018). Another study reviewed engineering simulations in AR from a broad perspective (Li et al., 2017). The study concluded that the integration of CFD data with AR/VR is mostly done with very unique elements using specific software, hardware, and dataflow pipelines (Li et al., 2017). Therefore, current data processing pipelines cannot be readily replicated to develop similar environments. In this study we will pursue a recently published methodology which proposes a comprehensive scheme to integrate any CFD simulation data with AR/VR by promoting an extract-based data processing approach (Solmaz and Van Gerven, 2020).

Unlike CFD simulations, process simulations are used to replicate the behavior of operation units or sections of chemical plants, requiring a lighter computational load and providing a more “bird-eye” view of the dynamics of the process. Many options are available, both commercial, such as Aspen Plus, Aspen HYSYS, UniSim or CHEMCAD, and open-source, such as COCO or DWSIM. Many of the commercial alternatives have their own educational environments focused on the training of operators, known as Operator Training Simulators (OTS), which are used to simulate the task of managing the plant. However, these tools are of limited flexibility and have been generally not used to teach chemical engineering students.

There are many examples of the use of process simulations in the training of operators, both by the use of OTS (Ahmad et al., 2016) or by developing a connection between the simulator and a virtual environment (Manca et al., 2013). Concerning AR/VR technologies, it is worth mentioning that the authors know of no previous case of an AR environment benefiting from process simulation, neither for operators nor for engineering students. On the other hand, there is a growing interest in the use of VR technologies in this field (Manca et al., 2013; Santos et al., 2016). The cases of process simulation used to teach chemical engineering students are more limited, with the only examples found building the case from the ground up without using available process simulation software (Mendes et al., 2010; Ouyang et al., 2018), none of them in VR. This limited previous application should not be a basis for discouragement. In this work we will benefit from the technically simple alternative of retrieving all interesting data from the process simulator and storing it for use in the environment, which proves a useful methodology for basic cases.

Three technical approaches were described on the integrity of process simulators with AR/VR. Firstly, the use of commercially available OTS, which must be discarded because their lack of customization options mean that it can only be used in a standard user interface, and not in AR/VR. Secondly, there exists the option of communicating between environment and simulation, such as in the work of (Santos et al., 2016). Typically this approach uses the OPC (Object Linking and Embedding (OLE) for Process Control) standard, a widespread communication standard designed for industrial use which provides both reliability and robustness, designed with a client-server architecture, although alternative work with the ActiveX/OLE standard exists (Manca et al., 2013). At least one module for the use of OPC in Unity has been developed commercially (https://game4automation.com/), while for ActiveX, further information can be found in (Santos and Van Gerven, 2020). Since this option demands the simultaneous use of a PC along with the mobile phone (as they are not capable of operating most process simulation software), it must be discarded for our case. It is of note that the same challenge would be present with other current AR hardware such as the HoloLens. Thirdly, and most popular in previous work for engineers (Bell and Fogler, 1996; Mendes et al., 2010) is to design the simulation without available process simulation software: this option can be quite technically demanding, although naturally is the one allowing for the highest customization. For this work, we will not depend on these previous approaches, and instead, we demonstrate how to obtain and store a table of results from the process simulation, which then can be used in the environment to create the illusion of active process simulation: we will execute this for a simple case (steady-state simulation with a small number of variables) and will propose alternative methodologies for more challenging scenarios.

The present work studies a facile and agile development methodology on cheap but content-wise rich educational AR applications. Integration of engineering simulations with AR as high-fidelity technical contents provides technically versatile and visually alluring digital educational tools. We experience, assess, and conclude this methodology with a case study of soap making. We also shed light on miscellaneous options that should be determined in the development of long-lasting digital environments.

2. System description

2.1. Methodology

Numerous theoretical frameworks can be found in the literature on the integration and use of digital tools in education. In particular, Technological Pedagogical Content Knowledge (TPACK) (Harris et al., 2009; Ke and Hsu, 2015) and Substitution, Augmentation, Modification, and Redefinition (SAMR) (Hamilton et al., 2016; Romrell et al., 2014) show that the development and sustainability of educational digital environments is strongly related to interdisciplinary skills from engineering, informatics, and educational science. In this study, we, therefore, aim to present a guideline having a basis on aforementioned frameworks in the technical development of digital tools.

Fig. 1 demonstrates the software development methodology proposed in the present work which comprises four main modules: 1) data creation and collection; 2) integration, 3) cross-platform development and 4) digital assets. The data creation and collection refers to all the steps required to create 3D computer graphics, process simulations, and multiphysics CFD simulations. Essentially, these steps are truly independent of AR/VR in the creation. The second module regards integration of outsource data into the Unity game engine. For this purpose, different tools were used which are detailed in further sections. Eventually, a cross-platform development module is proposed in which the designed digital environment, the technical content (simulations), and digital assets (four-module) are deployed via the Unity game engine. This enables a mere environment to control and develop entire processes. It is noteworthy that both AR and VR environments can be created by only switching between software development toolkits (SDKs) in later steps in the development. Finally, a built-in mobile AR application is deployed which is comparably cheap and accessible. In the next sections, each module will be elaborated.

Developing AR/VR tools are generally assumed to be complicated and expensive. They are also arguably developed and designed for a particular scenario in which several technical constraints should be taken into account beforehand such as digital resources, software and hardware. Assessments in the present study are concentrated on the development of a modular AR application with updatable technical content. This will increase the awareness of the design and content editing process while reducing complexity and cost.

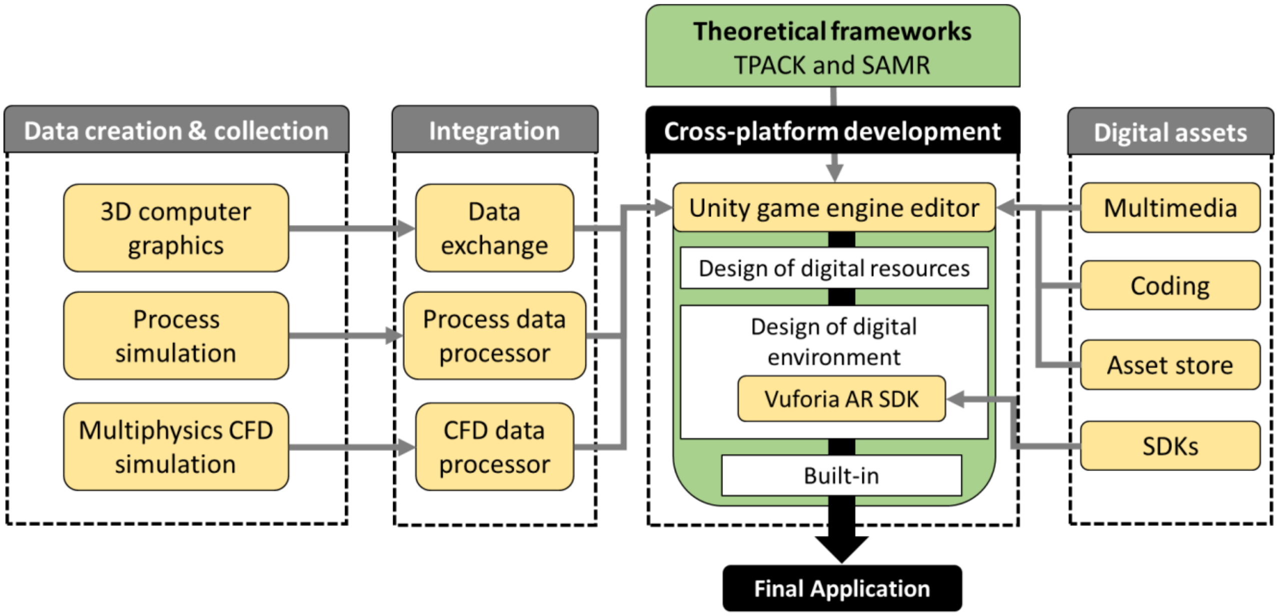


Fig. 1. Proposed software development methodology.

2.2. Development of AR software

The popular spread of AR in mobile devices has led to a variety of options in which AR applications can be built. Table 1 presents the general information of the most popular software for creating AR environments (Laine, 2018).

Table 1. Popular AR platforms.

SDK Name	Operating System (OS)	Application Programming Interfaces (API) availability	Compatible Game Engine	Tracking Target
ARCore	Android, iOS	Java, C#, C++	Unity, Unreal	Plane, marker, point cloud
ARKit	iOS	Objective-C, Swift	Unity, Unreal	Plane, marker, point cloud and face.
Vuforia	Android, iOS, Windows	C#, JavaScript, C++, Java	Unity	Plane, marker, object

Most of the software available to create AR environments provide SDKs written in different programming languages. These programs facilitate a simple integration inside game engines such as Unity and Unreal. SDKs can be very valuable as the integration for AR capabilities might be reduced in effort. The common AR tracking methods reported in literature are marker-based, marker-less based, and location-based tracking. Overall these techniques involve tracking the user's position and orientation by means of sensors available in the device to position the virtual objects relative to the real environment. Marker-based tracking uses computer vision technology that allows comparison against an image or database of images (i.e. the marker), after which 3D virtual objects can be positioned in real-time. If only objects naturally present in the scene are used to determine the position, the method is called marker-less. Location-based AR uses the sensors from the device and the positional data determined by the Global Positioning System

(GPS) to superimpose the virtual content (Andujar et al., 2011; Bacca et al., 2014; Pagani et al., 2016).

ARCore is Google's AR SDK and it supports both iOS and Android devices, whereas ARKit is Apple's AR solution and it is exclusive for iOS devices. While both SDKs can use a marker and marker-less tracking to recognize the same attributes (plane, marker, and point clouds), ARKit can also recognize facial features (Nowacki and Woda, 2020). On the other hand, Vuforia SDK uses computer-vision technology that allows it to recognize images and 3D objects in real-time and subsequently position virtual objects.

In this study, Vuforia marker-based tracking was used due to the support on multiple OS, well-developed technical documentation and smooth integration with Unity. The implementation of the SDK within the AR application is simple and does not require additional coding apart from the software provided by Vuforia, which makes this kit the most suitable for non-expert developers.

2.3. Pedagogical approach

Inquiry-based learning (IBL) approach is part of the everyday activities of any scientist. IBL activities provide the opportunity to hypothesize, explore, validate, discuss and explain scientific ideas (Chiang et al., 2014; Gormally et al., 2009). In this study, the learning tasks are designed to support IBL activities. Along with the different tasks, the AR system provides support to the user by presenting the information of the same phenomena with multiple representations, namely molecular structures, fluid dynamics and process simulations, to gain an understanding of a complex chemical process. As mentioned by (Baptista et al., 2019), multiple representations such as macroscopic, symbolic, and submicroscopic can aid students with a deeper and more structured understanding of a certain concept. The learning environment aims to guide students to complete tasks in the different processes involved in soap production. By interacting with different learning content, the users may gain relevant knowledge, which ideally they can connect, integrate, and implement in other contexts. In addition to the interaction with the content, the results obtained through the experience can direct students to collect and analyze technical information, solving a unique problem upon multiple parameters, and make decisions.

3. Implementation and case study

We performed a case study to scrutinize the software development methodology in the integration of AR software with engineering simulations. The liquid-soap synthesis, from reaction mechanism to industrial production, was chosen as the teaching content due to applicability of different chemical engineering concepts in a single scenerario.

3.1. Soap making as generic teaching content

Saponification involves the ester hydrolysis of a fat or oil (e.g. sunflower oil) using a strong alkali (e.g. KOH) (Fig. 2). As a result of this chemical transformation, a molecule of glycerol and the corresponding fatty acid salt (soap) are produced. Vegetable oils are the traditional raw materials for producing soap. Nearly 98% of these components contain mainly triacylglycerols (TAGs) which are triesters from diverse fatty acids (Dijkstra, 2015; Shipton and Holman, 1994).

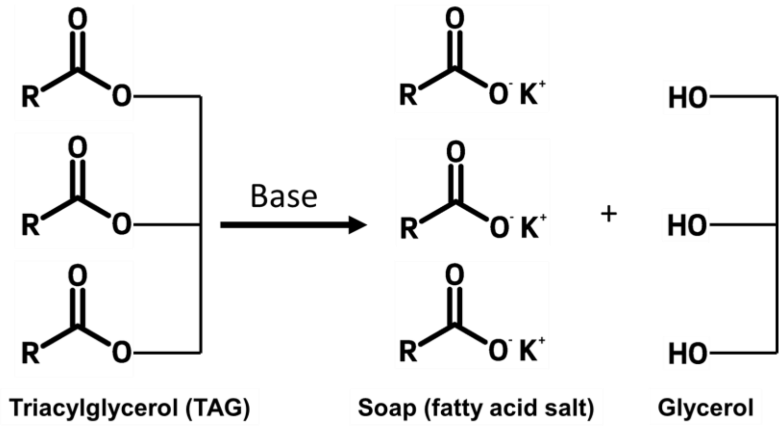


Fig. 2. Saponification reaction.

The reason for the use of soaps as cleansing agents comes from their nature as emulsifiers. An emulsifier is a chemical entity that eases the mixture of two immiscible layers, such as oil and water. The chemical composition of soap provides them with the ability to interact with both oil and water layers. While the large hydrocarbon chain is highly hydrophobic, the carboxylate head is hydrophilic and thus soluble in water (Fig. 3). This double chemical nature allows soaps to be efficient cleaners.

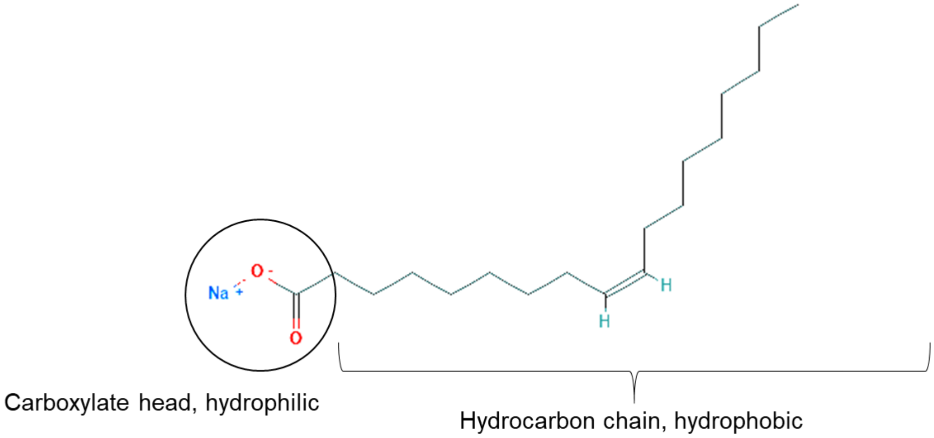


Fig. 3. Soap structure and sodium oleate molecule.

In chemistry, the 3D visualization of molecular structures can help students to understand on a deeper level the underlying mechanism of reactions. In this study, the molecular structures of the corresponding compounds involved in the saponification reaction were included as objects to be visualized through AR. These structures were obtained from PubChem which is a public database of chemical molecules (https://pubchem.ncbi.nlm.nih.gov/) in SDF (structure data file) format. The SDF format is not compatible with the Unity game engine so that the conversion to a compatible data format was performed using Pymol and Blender. Pymol (https://pymol.org/2/) is a commercial open-source software dedicated to molecular visualization with different structural and simulation tools. On the other hand, Blender is a general-purpose modeling program that allows the creation of photorealistic models. In this pipeline, the molecules in Pymol were transformed into a VRML data format as a collection of atoms with exact coordinates, which then Blender stored as meshes, the latter can be stored as FBX format which is then readable by the Unity game engine (Durrant, 2019). Fig. 4 shows the process of data conversion and integration of molecular structures with the Unity game engine.

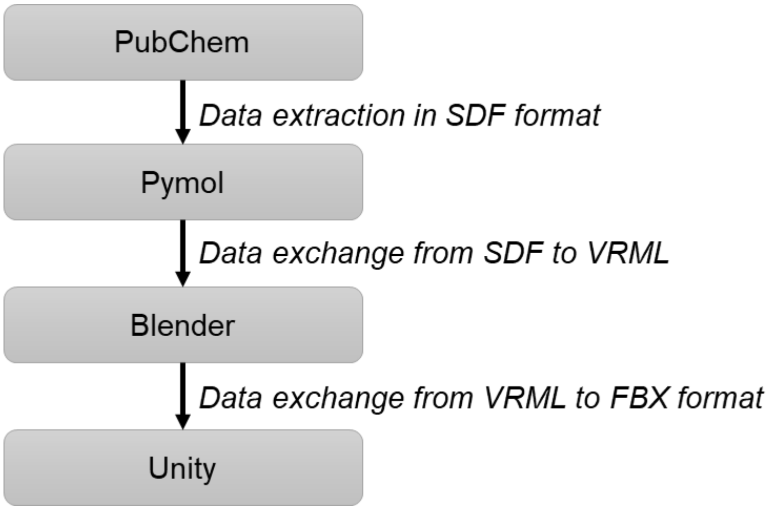


Fig. 4. Software pipeline of molecular data integration with Unity.

3.2. Design of Learning Environment

To give to users a complete overview of the production of soap and the evolution through time, the AR experience includes the following scenarios in sequence: medieval process of soap, modern lab-scale process, CFD and Process Simulation on the industrial scale. Ultimately, the synergy of the different activities in each scenario will help the user to learn a complex chemical process from different perspectives. The learning objectives for each scenario were realized by using the revised Bloom's Taxonomy proposed by (Anderson et al., 2001). This taxonomy is useful because it has served educators to unwrap the learning process and make clear learning objectives, which then can be matched with a coherent type of assessment.

The learning objectives of this case study focus around all aspects of the process of saponification itself, regarding reagents, products, stoichiometry, mass transfer and kinetics. They are:

- To *remember* the names of reactants and products, and associate them with everyday items;
- To *understand* the fact that the reaction consumes these reactants in a certain ratio (stoichiometry);
- To *remember* the effect of mass transfer limitation on the mixing as being dependent on reactor geometry, impeller rotational speed, flow and fluid properties;
- To *understand* that there is no single mixing strategy for efficiency- and cost-optimization
- To *remember* that temperature increases reaction speed;
- To *apply* the knowledge of the saponification reaction to optimally control a continuously-stirred tank reactor (CSTR).

Fig. 5 demonstrates the scenes of application together with the entire narrative. This structure is influenced by the practical guide on active learning of (Felder and Brent, 2016): The topic is chosen to fit with the recommendation to provide meaning (understood as a relationship to the learner's interests, goals, prior knowledge, and past experiences) as an approach to strengthen long-term retention, and the design methodology will follow their advocacy of inquiry-based learning: unlike in traditional education, the design philosophy is to make students work on problems before giving the answers, which has been shown to create desirable difficulties that deepen learning (Roediger et al., 2014).

Additionally, the environment is designed to create a sense of wonder (as defined by Hadzigeorgiou, 2012, who found that fostering a sense of wonder improves results in students: "Situations, phenomena, and ideas that give one the opportunity to admire, to feel a sense of mystery, to be surprised, astonished, bewildered and perplexed"), to make the students realize not only that their own knowledge of soap is limited, but that the explanation of the game itself is also limited. In other words, the game should not create a perception of "now I'm an expert in soap", but instead a sense of wonder of "there is a lot more in soap production than I originally thought".

Finally, the narrative seeks to replicate in a limited fashion the work of a chemical engineer, both to foster understanding of this profession, and to increase interest in chemical engineering.

AR application of the liquid-soap synthesis

1) Introduction to liquid soap synthesis

- A brief introduction is given upon soap and its cleansing action

2) Soap making in medieval time

- The student is tasked with producing sufficient soap to fight the coronavirus pandemic, given medieval instruments to produce it, and explained the reaction sequence.
- Producing a small amount of soap, the option to research the process is activated, which initiates the following stage.

3) Modern lab scale process

- The student is given modern laboratory equipment and can study the molecules that intervene in the reaction and the proportion at which they react.
- After a number of lab experiments, the next stage scale up is introduced.

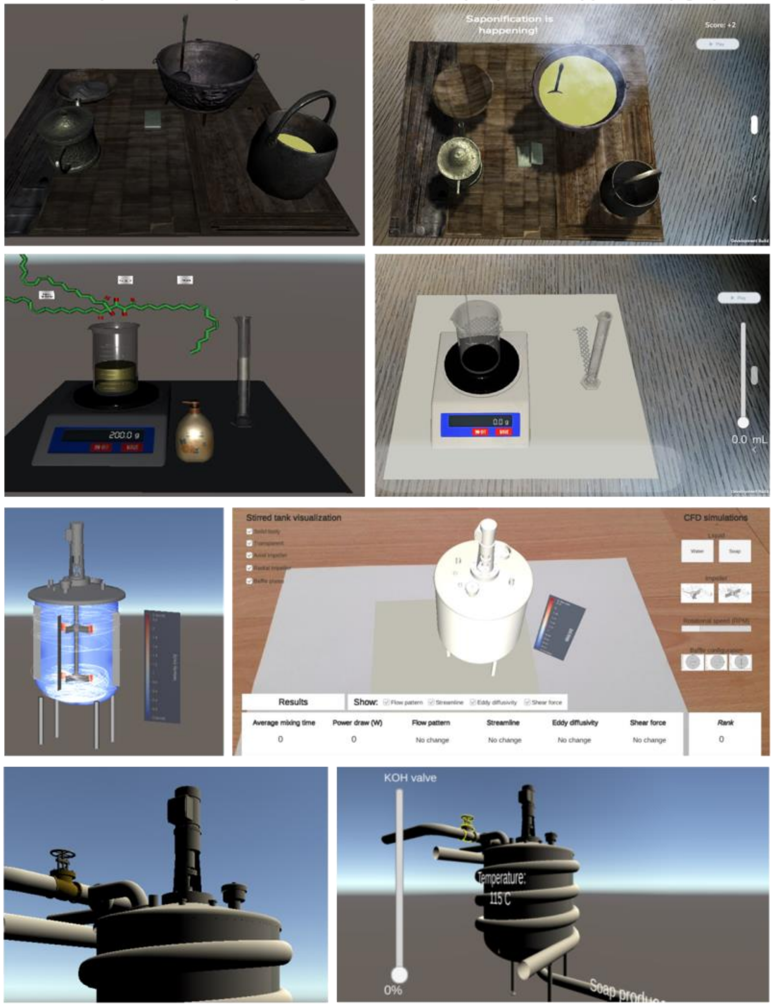
4) Scale-up mixing with CFD simulations

- The challenge of scaling up mixing is presented, and the student can explore several mixing options and decide on optimal mixing tank configuration through CFD simulations for different liquids, impellers, baffle plate configurations, and mixing speed.
- Student's decision comes through finding optimal case upon mixing efficiency, time and cost.

5) Industrial process simulations

- The student is finally placed in control of a CSTR, with the ability to change the ratios of the inputs, the speed of mixing and propeller, and the flow of steam used for heating. The experience will continue until the student has produced the required amount of soap, before which the student can choose to return to any previous stage while the plant continues to run in the setup chosen.

Example scenes in sequence: game engine editor (left) and AR application (right)



6) Final

- The student is asked to use liquid soap for hand-washing, and taught of how many drops of liquid soap is required for a proper wash.

Fig. 5. Narrations and captions from the scenes in AR application.

3.3. Augmented reality in the game engine editor

The Unity game engine serves as a software environment and robust production tool of interactive content such as 3D animations, VR, and AR experiences. In addition to the engine, the online asset store works as an online marketplace for developers to obtain a variety of resources thus reducing the burden of developing every asset from scratch (https://assetstore.unity.com/?on_sale=true&orderBy=1&rows=96). As mentioned previously, Vuforia SDK can be used inside the Unity game engine as an extension/plugin and its functionalities can be leveraged inside the engine. Fig. 6a shows the basic scene hierarchy of an AR application using Vuforia SDK inside the Unity game engine. The basic elements of an AR scene are 1) AR camera, 2) AR image target and 3) the 3D objects used inside the application. The 3D objects can be related to different resources, according to the function or the application it is designed for. These resources include audio source, animations, scripts, other 3D objects, materials, and particle systems. This hierarchy is also illustrated in Fig. 6b. The hierarchy window shows a list of objects that are present in the scene. In this example, the "boiler" is one of the 3D objects which has attached two resources: another 3D object ("oil") and a particle system ("smoke"). These resources attached to the boiler also have elements that modify, describe, or control their behavior such as animations, scripts, and materials.

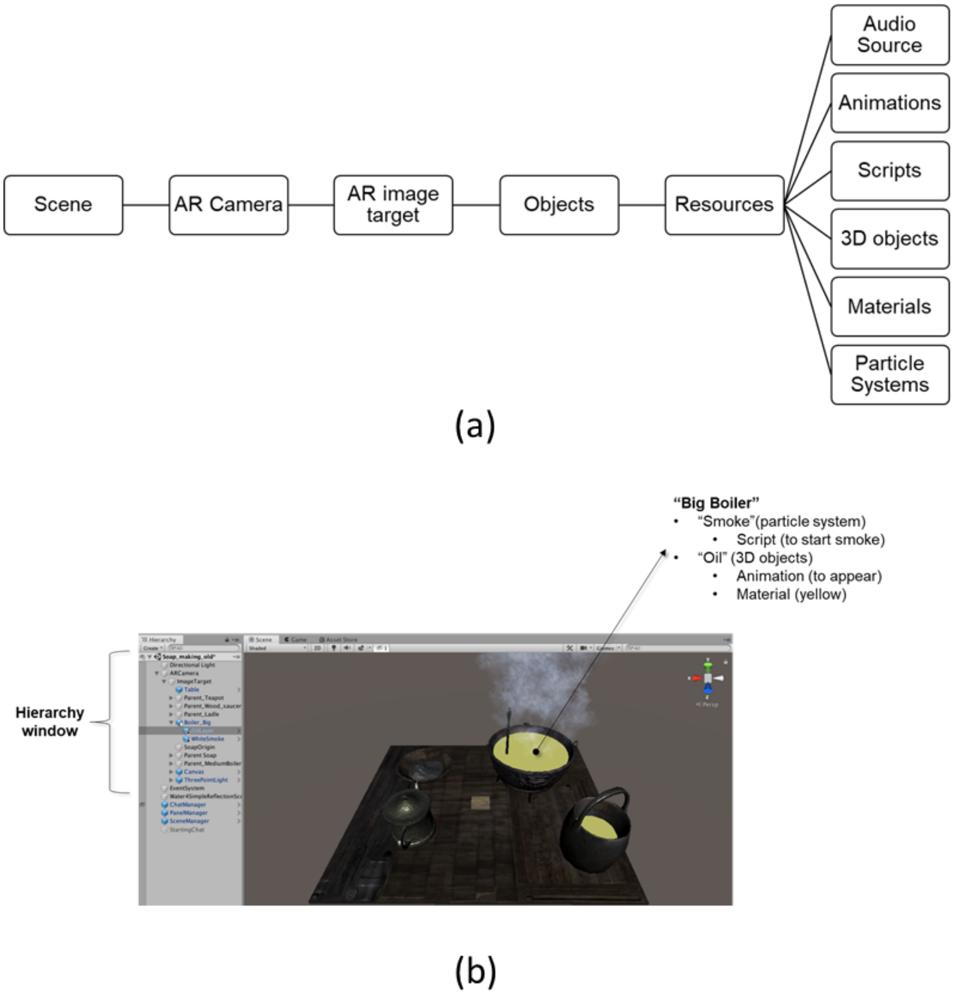


Fig. 6. AR development; (a) basic AR scene hierarchy inside Unity game engine; (b) snapshot of a scene inside Unity game engine.

3.4. Engineering simulations in scale-up and industrial process

Scale-up is a costly procedure with trial-and-error physical experimentations. Increasing stirred tank capacity by simple dimensional scale-up rules can cause falsified scale-up procedures. Each year in America 10 billion dollars is lost due to inefficient scale-up of mixing (Paul et al., 2004). Theoretical design methodologies and engineering simulation tools can provide cost- and time-effective solutions. In the present work, we aim to show how process and CFD simulations can aid scale-up and be integrated with AR/VR. Aiming to serve these tools to non-experts such as bachelor students, experts should optimize the entire simulation from pre-processing to post-processing.

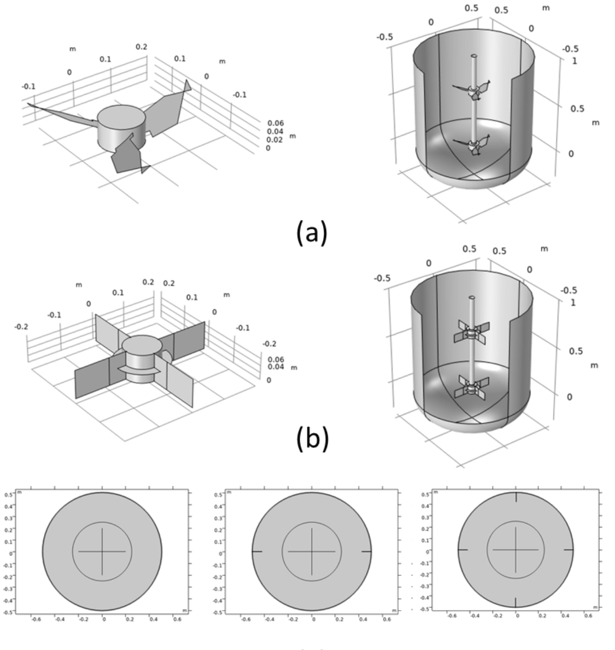
3.4.1. Multiphysics CFD simulations

The primary aim of scale-up in mixing is to obtain a similar performance that is achieved in smaller scales such as laboratory experiments. Both geometric similarities and mixing characteristics can be the first parameter to initiate a scale-up study. Simulating a complete process with CFD is truly computationally expensive and time-consuming. Thus, available theoretical and empirical works should be taken into account to obtain a preliminary design on the targeted scale. A validated CFD simulation can later on in the process be used to investigate, design and optimize parameters for dedicated applications in a sufficiently time- and cost-effective manner. In this study, we started the scale-up work with constant impeller tip speed that may facilitate similar shear and blending characteristics at different scales (Paul et al., 2004).

The liquid soap synthesis process is composed of multiple steps, including reaction kinetics and liquid-liquid mixing. The process is initiated with a clear liquid and completed with a certain yield of liquid soap, with several metaphases during the production (Zoller, 2008). Different design constraints might emerge due to changes in rheological specifications and flow conditions. The Handbook of Industrial Mixing has already shown proven theoretical and empirical design approaches on macromixing (Paul et al., 2004). Therefore, an initial design study was completed to obtain a baseline stirred tank model that can fundamentally help for the large-scale liquid soap sythesis process. A conventional stirred tank with an internal impeller can be used for low- and medium-viscous liquids up to 10,000 cP in scale-up (Paul et al., 2004). Both axial and radial impeller types were considered due to the change in rheological properties of working fluids from water to liquid soap. Radial impeller types can ensure sufficient mixing for relatively highly viscous liquids such as liquid soap. Baffle plates are broadly recommended and easy-to-apply internal structures to increase mixing performance. While baffles in low-viscous fluids can help to prevail over isolated mixing regions, they might harm flow patterns at higher viscosities or lower Reynolds numbers (Paul et al., 2004).

CFD simulations were performed with COMSOL Multiphysics 5.5. A Dell (Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60ghz 2.60 GHz with 128 GB installed RAM) workstation was employed to run simulations. Both liquid soap and water were examined to assess system parameters. Liquid soap was considered Newtonian with a viscosity of 1000 cP and a density of 1050 kg/m³ (Zoller, 2008) whereas values of 1 cP and 1000 kg/m³ were used for water, respectively. Flow modeling was decided according to the Reynolds number between laminar and k-w SST turbulence model. Fluid flow with $Re \geq 20{,}000$ was accepted as turbulent (Kresta et al., 2016).

Two variations were compared to understand in-depth the role of the design parameters with a stationary solver reaching results in quasi-steady states. Firstly, the 3D stirred tank was examined for axial and radial flow impellers, for low- and medium-viscous fluids and for varying mixing speeds. In the second study, a 2D mixer with a radial impeller was applied to assess the baffle plate orientations for low- and medium-viscous fluids. Run-times on average for the 3D stirred tank and 2D mixer were 25 and 3 min, respectively. Fig. 7 demonstrates the axial and radial impellers with the tank geometry used in this study.



(a)

(b)

(c)

Fig. 7. Preliminary geometry design; (a) axial impeller (3-pitched blade turbine) in 3D stirred tank; (b) radial impeller (4-blade Rushton turbine) in 3D stirred tank; (c) baffle plate orientations in 2D mixer.

Table 2 presents a summary of the CFD simulations performed. We used direct and derived values for the flow patterns, turbulence quantities, mixing time scales and power numbers to interpret simulation results. Results essentially outline that a stirred tank reactor with radial turbine blades (Rushton turbine), 100 RPM rotational speed and two baffle plates constitutes the best mixing performance within the studied operational ranges for liquid soap making in large scale. This final design showed a good agreement with practices available in the literature and industry (Kresta et al., 2016).

Table 2. Simulations with relevant parameters utilized in the assessment.

Simulation		**Input**				**Output**
Geometry	**ID**	**Impeller type**	**Number of baffles**	**Liquid**	**Stirring speed (RPM)**	**Assessment**
3D stirred tank	**s1.1**	Axial	-	Water	100	Impeller
	s1.2			Soap	100	
	s2.1	Radial		Water	100	
	s2.2			Soap	100	
	s3.1			Water	50	RPM
	s3.2			Water	100	
	s3.3			Water	200	
2D mixer	**s4.1**	Radial	-	Water	100	Baffle
	s4.2		-	Soap		
	s5.1		2	Water		
	s5.2		2	Soap		
	s6.1		4	Water		
	s6.2		4	Soap		

COMSOL multiphysics exports simulation results only in text (CSV and TXT) or VTU file format without any data loss. In the present work we applied a similar approach to integrate CFD data in the game engine which provides replicable data processing. Re-processing of a text file to generate visual representations in a game engine is not an efficient manner of integration due to redundant re-processing of extracted data. Instead, extract-based visual representations 3D file format can be exported and processed which only requires exchanges between compatible file formats (Solmaz and Van Gerven, 2020). Fig. 8 shows the software pipeline of CFD data integration with Unity game engine. Data extraction in text (for engineering analytics in CSV format) and multimedia files (for image, video, etc.) can be directly imported in the game engine.

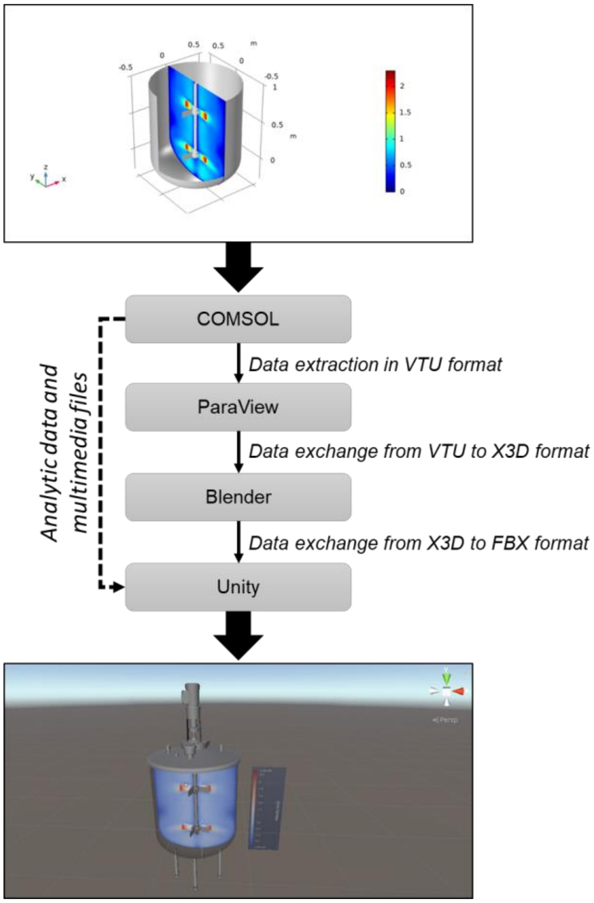


Fig. 8. Software pipeline of CFD data integration with the Unity game engine.

3.4.2. Process simulations

The behaviour of a CSTR conducting the saponification reaction is implemented in Aspen Plus, a process simulation software widely used for chemical processes. The sunflower oil is represented by trilinolein, and a user defined compound is used to represent potassium linoleate, with the only information needed being the molecular weight, which is known; the formation enthalpy, which is calculated from the enthalpy information (Pita Cañola and Pincay Durán, 2012), and the vapor pressure which is set as -10^{20} bar, indicating to Aspen Plus that the compound will not evaporate. The ideal property method is selected, the kinetic constant of the reaction (Asiagwu, 2010) and the activation energy (Reyero et al., 2015) are determined from existing resources. The simulation is not rigorous, since neither the presence of two phases nor the effect of electrolytes in the reaction are taken into account, but is sufficient to show the consequences of variable changes to the student.

Three variables can be changed by the student (Table 3): the flow of the potassium hydroxide (KOH) 50 wt.% solution, which the student will perceive as opening a valve; the kinetic constant, which the student will perceive as changing the mixing speed (using an existing correlation (Almilly, 2014)); and the heat duty, which the student will perceive as controlling a flow of vapor (see Table 3). Instead of directly communicating the Unity game engine and the simulation as previously highlighted (Santos et al., 2016; Santos and Van Gerven, 2020), we discretize these variables and run the simulation for a number of iterations to store all possible solutions in a table inside the game. For this particular case, we choose to discretize in 10 intervals, for a total of 1000 runs. The values stored are the three input variables, the resulting mass flow of soap, and the temperature in the CSTR.

To automate simulation runs we create a C# project in Visual Basic 2019 and follow the automation guidelines in the Aspen Plus user guide by Aspen Technology Inc. (Aspen Technology Inc, 2001) in Chapter 38: first, the type library happ.tbl must be included in the C# project; then, the path for each intended variable must be located in the variable explorer inside the Aspen Plus simulation. Finally, a simple loop is implemented where the input is changed and results are stored in a 2D array. A step by step guide of interaction between C# and Aspen Plus can be found in the literature (Higgins, 2016).

Table 3. Variables from process simulator to be integrated with the game engine.

Variable	Environment element	min	max	Units	progression
KOH 50 wt.% mass flow	valve	0	70	kg/h	linear
Reactor heat duty	steam flow	0	2	kcal/s	linear
Kinetic constant	mixer speed	1.16E-05	1.16E-03	s^{-1}	logarithmic

Once the data has been obtained, it can be stored directly in a binary file by using the BinaryFormatter class, and then introduced into the assets of the Unity environment and loaded at the start of the scene. When the player changes any input, a search is conducted to find the corresponding outputs, which are used then for any necessary environment calculations.

4. Results and discussion

The AR application aimed to show how one can integrate and reuse resources from one scenario to another which ultimately makes the integration easier, reproducible, and scalable. The integration of digital resources in Unity game engine facilitated not only the use of other AR applications but also opened the possibility to be used using other target platforms (e.g., VR or other SDKs). For example, the "chat" integrated into the AR application, was obtained from an online tutorial. This script allowed us to add messages and information to certain objects regardless of their scenario or characteristics. Another example of tools available in the Unity game engine to ease the reproducibility is the "prefabs". Prefabs systems allow to "create, configure and store all the properties of objects” which can later be reused as such in the application.

The assessment in the development of AR application was extended to provide long-lasting digital environments. Software development tools, technical content, digital resources, and software development strategy were accordingly elaborated in the following sections.

4.1. Software development tools

In this study, the Vuforia SDK was used to implement the AR system using a marker-based technology, and the development was done in the Unity game engine. One of the main advantages of using the game engine as the environment to build the application is that unlike the officially integrated development environments (Android Studio for Google's Android operating system or Xcode for macOS) which are specific to the operating system, in the Unity game engine most of the resources can be reused (e.g., prefabs, scripts, multimedia) and only the SDK specific setups have to be developed. For instance, the basic setup of the AR scene using Vuforia presented in Fig. 5 may differ when using either ARCore or ARKit, but the objects used can be the same.

Compared to other SDKs, Vuforia is only compatible with Unity and runs under any operating system. From the developer's point of view, the choice of both the game engine and target device may motivate the selection of the adequate SDK to add AR systems. For example, since Vuforia is not compatible with Unreal, a developer intending to create an app compatible with Android and Apple devices and using the Unreal engine will not be able to leverage this SDK and, as a result, creating two versions of the app may be needed, one using ARKit and another ARCore. Moreover, the Unity game engine offers a wider variety of compatible data formats (FBX, DAE, 3DS DXF, C4D, JAS, IXO, BLEND and SKP) when compared to Unreal (FBX, OBJ and SRT). For cases like the one presented in this study where the data processing pipeline includes different software, a tool such as the Unity game engine offers greater data compatibility.

4.2. Integration of technical content

4.2.1. Molecular visualizations

One of the aims of this study was to explore how to integrate molecular structures inside the AR application. The intention was to complement the learner’s understanding of the saponification reaction by means of molecular visualization. In this study, it was identified that the integration of molecular structures in the AR environment as objects requires data processing. Although this integration is relatively easy, it requires more than one software to integrate with a game engine.

Interestingly, this need has already been tackled by some researchers who, for their own needs, have developed tools that can be used for this purpose, such as openbabel (O'Boyle et al., 2011), blendmol (Durrant, 2019), and unitymol (Lv et al., 2013). These are some available resources developed as chemicals tools or dedicated plugins for molecular visualization which reduce or try to solve some of the issues that one encounters in this data processing pipeline. This information can be used to develop targeted AR experiences exploring and enhancing the interaction with real chemical databases and molecular simulations to provide more realistic scenarios.

4.2.2. Multiphysics CFD simulations

A complete CFD simulation, from the generation of geometry to post-processing of results, requires a skilled expert to make several intermittent decisions. Each action taken by experts should refer to a logical reasoning to justify the work carried out. Any incorrect setting in the simulation might cause indecisive and misleading results. Once all settings have been validated, the expert can hand over the tool to a student to carry out further explorations on results and even simulations. CFD software developers and academic institutions worldwide have been promoting validated CFD simulations. These might serve lecturers less experienced in CFD to utilize meaningful CFD results in education.

CFD simulation results were integrated with AR in the present work. Visual representations (3D extractions from CFD results), supplements (colormap, charts, graphs, and relevant multimedia) and engineering analytics (analytic data) were located in the AR environment. AR/VR integrated CFD data were directly stored in the game engine editor and AR application. Almost all CFD solvers enable automated parametric analysis so that a broader database can be entailed to AR applications.

Furthermore, once a CFD tool is validated, an inexperienced CFD user can start carrying out simulations with limited options. In that case, a two-way coupling is required between the CFD solver (for example a connected workstation and cloud server) and the AR application instead of generating and collecting CFD results in the game engine editor and AR application in advance. It is noteworthy that CFD runtime should be taken into account which may vary from minutes to days. Developers should also pay attention to re-run simulations from the input given by the user in the AR application. Commercial software might hinder a direct connection due to the concerns over licensing and data encryption.

4.2.3. Process simulators

A simple approach was used to obtain useful data from process simulations that can be easily implemented for the learning experience. The main limitations of this simple approach are that it will only work for deterministic systems in steady-state, and the search time increases exponentially with the number of inputs. However, these limitations are not severe when dealing with many topics of interest in chemical engineering, with the exception being those that are inherently dynamical or deal with a great number of input variables, such as process control, design, or intensification.

Approaches to tackle these limitations include implementing simulation-environment communication with OPC or ActiveX as shown in the bibliography, or a metamodeling approach where a surrogate model of the real system is acquired and replicated inside the environment. While the Unity game engine does provide ML-Agents, a machine learning tool for this purpose , it is only intended for modeling the behavior of agents from data obtained inside the engine itself, and it currently does not explicitly allow for models extracted from external data. Regarding the use of simulation-environment communication, a private module to ease the use of the Unity as an OPC client (https://game4automation.com/), or previous work on communicating Unity with ActiveX (Santos & Van Gerven, 2020) are available. The implementation of either alternative is technologically more challenging, in contrast with the work shown.

4.3. Other digital resources

Multimedia and 3D models are often one of the key elements of digital environments. Finding a desired digital content for an engineering subject can be quite source-demanding. Developers should be able to generate their content or be aware of outsourcing facilities. Multimedia and 3D models can be stored and imported from an offline local disk to the game engine editor. They can be controlled over connected web or cloud services by hosting data in remote directories. This may aid the developers to store and update objects based on address of directories instead of using game engine editors directly. In this part, we assessed our experience on multimedia, 3D models, and animations throughout the development of the liquid soap making study.

4.3.1. Multimedia

The study showed that many digital multimedia models are supported by the Unity game engine. Some of these are video player (AVI, MP4 and MPEG), audio resource (AIFF, WAV, MP3 and Ogg), image (JPEG, GIF, PNG, TIFF, and PSD) and text file models with various formats. Multimedia can be either generated or imported in the game engine as GameObjects. Additionally, the handling of the multimedia resources can be done directly in the engine, or it can be done through scripting in C# with Unity's API for user-specific demands.

4.3.2. 3D models

A digital 3D model fundamentally holds information about geometry as well as texture, animation, and the scene of both 2D and 3D computer graphics. Several data formats have been released to date concerning different aspects of applied fields. 3D models are vital for engineering. Computer Aided Design (CAD) systems offer special data formats and software in support of design and modeling. Several open-source (e.g., GrabCAD, Autodesk, Sketchfab and Clara.io) and paid databases are available online providing 3D models on miscellaneous engineering subjects. Notably, 3D models used in CAD systems are encoded with dedicated 3D file formats to secure analytical data for design and modeling. These formats generally differ from the ones used by game developers (Smith et al., 2014). Therefore data processing, exchanging between data formats, is necessary to import the intended model in the game engine. A 3D computer graphics software, such as Blender, can effectively be utilized. The developer can find data formats supported by each software in the technical documentation of the relevant software.

4.3.3. Animations, visual effects and physics

The animation system of the game engine provides a comprehensive development tool to produce 2D and 3D animations. Any object in the game engine can be used to create an animator and animation clip. Animations from external resources can be imported in the game engine as long as a suitable 3D file format is selected such as the FBX or DAE format.

From sunlight in the sky to smoke propagation in a complex building, visual effects for a wide range of cases can be created in the editor with the Visual Effect Graph and Particle System tools. The Visual Effect Graph tool is mainly advantageous for large scale and user-controlled effects through the use of 2D and 3D objects. On the other hand, the Particle System, a special unit of visual effects, enables the users to simulate real-time effects with a cluster of individual particles. It can be a wise option to create dynamic and spontaneous behavior of gas and liquid effects. Further rendering might be required for both tools in order to adjust the appearance of objects and particles.

Dynamic real-world interaction in the game engine is only possible with attributes of physical action and movement. The Unity game engine comprises different physics packages to simulate forces acting on individual and interactive objects. Physics engines help developers to demonstrate reliable physical incidents such as acceleration, gravity, collision, buoyancy, and many others. Several free and paid packages in Unity Asset Storage may help to simulate the concerned physics. For instance, NVIDIA FleX for the Unity game engine is a particle-based real-time physics simulator including physics of fluid. Basic fluid properties such as density and viscosity can be simulated in the game engine (Sagheb et al., 2019). It is noteworthy that physics engines fundamentally aim at real-time and visually credible results for computer graphics. Hence, neither high numerical accuracies nor physical validation should be expected with those simulation tools. Additionally, applying real-time physics-based simulation would require more powerful user hardware to run with.

4.4. Software development strategy and post-treatment

4.4.1. Stand-alone vs content-publishing models

Strategies to develop an application with a game engine is generally configured based on the needs of management and post-treatment of digital resources (e.g., multimedia). Literature (Messaoudi et al., 2015; Porter and Heppelmann, 2017) and our own assessment highlighted that stand-alone and content-publishing models are two unique strategies that might be pursued by the developers. In the stand-alone strategy, any update should be processed in the game engine editor, and the application should be rebuilt subsequently. Assets uploaded in the game engine editor are stored in the project's asset directory. In this sense, data is preserved by the game engine and application itself. Any change in the asset directory can automatically update assets inside the game engine editor. Players, who had already downloaded the final application, should again update or download the new version from the relevant repository (e.g. GooglePlay, AppStore or Github). On the contrary, the content-publishing strategy (via local or connected servers) gives flexibility over management and post-treatment. Assets, individually or in chunks, can be linked to a server and retrieved in the final application during run time. UnityWebRequest offers a modular content-sharing system that relies on HTTP requests for devices with server connections. Likewise, the Unity game engine provides specific asset management tools to deal with data in stand-alone and content-publishing models which is elaborated in the following section.

To update data directly from a connected server in the content-publishing model, the relevant asset should be stored through a static URL. This means that even though the asset is changed, the same URL address can be preserved to update the data. Both local and connected facilities can be considered for the content-publishing model. Speaking of multimedia, it was observed that videos can directly be streamed in the application by setting a URL in GameObject as long as the format complies with the game engine editor. For instance, video format in URL from YouTube is not compatible with the game engine; thus, it cannot be streamed directly. The utility of the YouTube videos in external directories may also not abide by copyright agreement. Furthermore, Vimeo, another online video hosting platform, enables direct support to access video files with URLs in MP4 format. Alternatively, any other web hosting platform (Google Cloud, university infrastructures, etc.) would rather be preferred to stream videos with static URLs. As long as the data with a compatible format is stored and can be accessed publicly with a static URL, the application should run the content without issue. Further analysis showed that other multimedia types should be controlled over a C# script, Unity data management tools, or external plugins (e.g., AppCrafted for live texts from the web browser) to stream data from a server.

To summarize, compared to the stand-alone model, the content-publishing provides an automated and easy updating methodology through connected servers without any work to be done in the game engine editor. Nonetheless, the model may bring several pitfalls that the developers should pay attention to. It requires continuous support from a remote server to host data and provide a link whenever required. User devices should be connected to the system to retrieve data from the host. Paid services might be necessary for long-lasting applications. Developers should put more effort in the game engine editor to set a robust connection between the server and final user application. A blended approach upon stand-alone and content-publishing may be a remedy to deal with user-specific demands.

4.4.2. Data management in game engine

Processing large-size and complex-asset configurations (simulations, 3D models, animations and multimedia) in varying forms may require an intelligent system in the game engine to compress, store, access and retrieve data when needed. The Unity game engine provides different data management solutions to developers' needs. Aside from conventional "File-based" workflows, "Unity package", "AssetBundle" and "Addressable asset system" are other possible methods for managing data in the game engine.Table 4 illustrates a comparison among data management tools, including a case study on data size compression.

Table 4. Comparrison among data mangement stratagies in the Unity game engine; given support (+) and no support (-).

Data management tool	File-based assets	Unity package	AssetBundle	Addressable asset
Data storage	+	+	+	+
Data size compression	-	+	+	+
Data size reduction	-	-	+	+
Portability	+	+	-	+
Browser	-	-	+	+
Runtime load	-	-	+	+
Data management tool	-	-	+	+

Memory management tool	-	-	-	+
Local deployment	+	+	+	+
Remote deployment	-	-	+	+
Stand-alone	+	+	+	+
Content-publishing	with UnityWebRequest	with UnityWebRequest	+	+
Utility	Fast and simple	Fast and simple	Advanced and complex	Advanced and complex
Size of assets (MB)	122	40.6	18.1	18.1

The Addressable asset system is far advantageous due to outstanding support on data management and size compression. Alternatively, AssetBundle can be taken into account for less complex environments. Unity package only serves for compression and storage of the assets securely. It does not provide any additional benefit for managing data inside the game engine. From File-based to Addressable assets, the complexity of development procedures and the need for customization in the game engine editor considerably increases in terms of time, technical development and coding skills.

4.4.3. User experience assessment

Actions in the application taken by users can be tracked and utilized in various manners. Acquisition of logs and analytical information in the application are two significant features to obtain data from the users. Log files can be created automatically in the application and developers can access data from the relevant directory (https://docs.unity3d.com/Manual/LogFiles.html). An order can also be given in the game engine editor to send log files of each user via email to correspondents when the levels in the application are completed. In addition, the Unity game engine also comprises an analytics tool, so-called Unity Analytics, to track the user's actions in the application within customizable features.

On the other hand, a survey, quiz or questionnaire can always be integrated in the application in case an assessment is targeted based on users' responses to questions. In principle, creating a user interface (UI) via relevant scripting may serve this purpose. It was also experienced that there are several free or paid plugins that the developer might benefit from (MaterialUI, Simply Trivia, etc.). Developers can readily modify and generate questions at any level of the application.

5. Conclusion

This study proposed a facile and agile methodology to integrate AR software with technical content such as animations, 3D models and engineering simulations. By using a technical case study to teach the process of soap making, we aimed to share the process to develop a complementary educational application.

Our research underlined the importance of a tailor-made development strategy and competent human resources to develop long-lasting digital environments with desired technical content. We provided further evidence that engineering simulations with AR may facilitate versatile and sustainable educational tools to operate active learning environments. The present study only investigated and presented results on the development work as a guideline. Therefore, further studies need to be carried out in order to validate the educational value of the AR application.

Acknowledgement

This project has received funding from the European Union's EU Framework Programme for Research and Innovation Horizon 2020 under Grant Agreement 812716. This publication reflects only the authors' view exempting the community from any liability. Project website: https://charming-etn.eu/.

Supporting Information

The digital tool associated with this article can be found at https://github.com/alfajess92/SoapMakingProject.

References

Ahmad, Z., Patle, D.S., Rangaiah, G.P., 2016. Operator training simulator for biodiesel synthesis from waste cooking oil. Process Safety and Environmental Protection 99, 55–68. https://doi.org/10.1016/j.psep.2015.10.002

Akçayır, M., Akçayır, G., 2017. Advantages and challenges associated with augmented reality for education: A systematic review of the literature. Educational Research Review 20, 1–11. https://doi.org/10.1016/j.edurev.2016.11.002

Almilly, R.F.K., 2014. Kinetics of the saponification of mixed fats consisting of olein and stearin. Journal of Engineering 20, 144–159.

Anderson, L.W., Krathwohl, D.R., Bloom, B.S., 2001. A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives 41, 352.

Andujar, J.M., Mejias, A., Marquez, M.A., 2011. Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory. IEEE Transactions on Education 54, 492–500. https://doi.org/10.1109/TE.2010.2085047

Asiagwu, A., 2010. Kinetics of saponification of Treculia africana oil using a locally sourced alkaline. Int J Nat Appl Sci 6.

Aspen Technology Inc, 2001. Aspen Plus User Guide.

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., 2001. Recent advances in augmented reality. IEEE Computer Graphics and Applications 21, 34–47. https://doi.org/10.1109/38.963459

Azuma, R.T., 1997. A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments 6, 355–385. https://doi.org/10.1162/pres.1997.6.4.355

Bacca, J., Baldiris, S., Fabregat, R., Graf, S., Kinshuk, 2014. Augmented reality trends in education: A systematic review of research and applications. Educational Technology and Society 17, 133–149.

Baptista, M., Martins, I., Conceição, T., Reis, P., 2019. Multiple representations in the development of students' cognitive structures about the saponification reaction. Chemistry Education Research and Practice 20, 760–771. https://doi.org/10.1039/c9rp00018f

Beck, D., 2019. Special Issue: Augmented and Virtual Reality in Education: Immersive Learning Research. Journal of Educational Computing Research 57, 1619–1625. https://doi.org/10.1177/0735633119854035

Bell, J.T., Fogler, H.S., 1996. Vicher: A virtual reality based educational module for chemical reaction engineering. Computer Applications in Engineering Education 4, 285–296. https://doi.org/10.1002/(SICI)1099-0542(1996)4:4<285::AID-CAE4>3.0.CO;2-9

Berger, M., Cristie, V., 2015. CFD Post-processing in Unity3D. Procedia Computer Science 51, 2913–2922. https://doi.org/10.1016/j.procs.2015.05.476

Brown, M., Mccormack, M., Reeves, J., Brooks, D.C., Grajek, S., Bali, M., Bulger, S., Dark, S., Engelbert, N., Gannon, K., Gauthier, A., Gibson, D., Gibson, R., Lundin, B., Veletsianos, G., Weber, N., Horizon, E., 2020. 2020 EDUCAUSE Horizon Report ™ Teaching and Learning Edition, EDUCAUSE Horizon Report is a trademark of EDUCAUSE.

Chen, Y.-C., Chi, H.-L., Hung, W.-H., Kang, S.-C., 2011. Use of Tangible and Augmented Reality Models in Engineering Graphics Courses. Journal of Professional Issues in Engineering Education and Practice 137, 267–276. https://doi.org/10.1061/(ASCE)EI.1943-5541.0000078

Chiang, T.H.C., Yang, S.J.H., Hwang, G.J., 2014. An augmented reality-based mobile learning system to improve students' learning achievements and motivations in natural science inquiry activities. Educational Technology and Society 17, 352–365.

Dijkstra, A.J., 2015. Vegetable Oils: Composition and Analysis, 1st ed, Encyclopedia of Food and Health. Elsevier Ltd. https://doi.org/10.1016/B978-0-12-384947-2.00708-X

Durrant, J.D., 2019. Blendmol: Advanced macromolecular visualization in Blender. Bioinformatics 35, 2323–2325. https://doi.org/10.1093/bioinformatics/bty968

Enyedy, N., Danish, J.A., Delacruz, G., Kumar, M., 2012. Learning physics through play in an augmented reality environment, International Journal of Computer-Supported Collaborative Learning. https://doi.org/10.1007/s11412-012-9150-3

European Commission, European Parliament, European Council, European Economic and Social Committee, Committee of the Regions (Eds.), 2011. Supporting growth and jobs: an agenda for the modernisation of Europe's higher education systems. Publications Office of the European Union, Luxembourg.

Felder, R.M., Brent, R., 2016. Teaching and learning STEM: A practical guide. John Wiley & Sons.

Fukuda, T., Yokoi, K., Yabuki, N., Motamedi, A., 2019. An indoor thermal environment design system for renovation using augmented reality. Journal of Computational Design and Engineering 6, 179–188. https://doi.org/10.1016/j.jcde.2018.05.007

Fukuda, T., Yokoi, K., Yabuki, N., Motamedi, A., 2018. An indoor thermal environment design system for renovation using augmented reality. Journal of Computational Design and Engineering. https://doi.org/10.1016/j.jcde.2018.05.007

Garzón, J., Acevedo, J., 2019. Meta-analysis of the impact of Augmented Reality on students' learning gains. Educational Research Review 27, 244–260. https://doi.org/10.1016/j.edurev.2019.04.001

Garzón, J., Pavón, J., Baldiris, S., 2019. Systematic review and meta-analysis of augmented reality in educational settings. Virtual Reality 23, 447–459. https://doi.org/10.1007/s10055-019-00379-9

Ge, W., Wang, W., Yang, N., Li, J., Kwauk, M., Chen, F., Chen, J., Fang, X., Guo, L., He, X., Liu, X., Liu, Y., Lu, B., Wang, Jian, Wang, Junwu, Wang, L., Wang, X., Xiong, Q., Xu, M., Deng, L., Han, Y., Hou, C., Hua, L., Huang, W., Li, B., Li, C., Li, F., Ren, Y., Xu, J., Zhang, N., Zhang, Y., Zhou, Guofeng, Zhou, Guangzheng, 2011. Meso-scale oriented simulation towards virtual process engineering (VPE)—The EMMS Paradigm. Chemical Engineering Science 66, 4426–4458. https://doi.org/10.1016/j.ces.2011.05.029

Gormally, C., Brickman, P., Hallar, B., Armstrong, N., 2009. Effects of Inquiry-based Learning on Students' Science Literacy Skills and Confidence. International Journal for the Scholarship of Teaching and Learning 3. https://doi.org/10.20429/ijsotl.2009.030216

Hadzigeorgiou, Y.P., 2012. Fostering a sense of wonder in the science classroom. Research in Science Education 42, 985–1005.

Hamilton, E.R., Rosenberg, J.M., Akcaoglu, M., 2016. The Substitution Augmentation Modification Redefinition (SAMR) Model: a Critical Review and Suggestions for its Use. TechTrends 60, 433–441. https://doi.org/10.1007/s11528-016-0091-y

Harris, J., Mishra, P., Koehler, M., 2009. Teachers' Technological Pedagogical Content Knowledge and Learning Activity Types: Curriculum-based Technology Integration Reframed. Journal of Research on Technology in Education 41, 24.

Heuveline, V., Ritterbusch, S., Ronnås, S., 2011. Augmented Reality for Urban Simulation Visualization. INFOCOMP 2011: The First International Conference on Advanced Communications and Computation. https://doi.org/10.11588/emclpp.2011.16.11682

Higgins, S.J., 2016. Design and Optimization of Post-Combustion CO2 Capture (PhD Thesis). Virginia Tech.

Ke, F., Hsu, Y.-C., 2015. Mobile augmented-reality artifact creation as a component of mobile computer-supported collaborative learning. The Internet and Higher Education 26, 33–41. https://doi.org/10.1016/j.iheduc.2015.04.003

Kim, M., Yi, S., Jung, D., Park, S., Seo, D., 2018. Augmented-Reality Visualization of Aerodynamics Simulation in Sustainable Cloud Computing. Sustainability 10, 1362. https://doi.org/10.3390/su10051362

Kresta, S.M., Etchells, I., Dickey, D.S., Atiemo-Obeng, V.A., Victor, A., 2016. Advances in Industrial Mixing. John Wiley & Sons.

Laine, T., 2018. Mobile Educational Augmented Reality Games: A Systematic Literature Review and Two Case Studies. Computers 7, 19. https://doi.org/10.3390/computers7010019

Li, J., 2015. Approaching virtual process engineering with exploring mesoscience. Chemical Engineering Journal 278, 541–555. https://doi.org/10.1016/j.cej.2014.10.005

Li, W., Nee, A., Ong, S., 2017. A State-of-the-Art Review of Augmented Reality in Engineering Analysis and Simulation. MTI 1, 17. https://doi.org/10.3390/mti1030017

Lin, J.-R., Cao, J., Zhang, J.-P., van Treeck, C., Frisch, J., 2019. Visualization of indoor thermal environment on mobile devices based on augmented reality and computational fluid dynamics. Automation in Construction 103, 26–40. https://doi.org/10.1016/j.autcon.2019.02.007

Liu, D., Dede, C., Huang, R., Richards, J. (Eds.), 2017. Virtual, Augmented, and Mixed Realities in Education, Smart Computing and Intelligence. Springer Singapore, Singapore. https://doi.org/10.1007/978-981-10-5490-7

Lv, Z., Tek, A., Da Silva, F., Empereur-mot, C., Chavent, M., Baaden, M., 2013. Game On, Science - How Video Game Technology May Help Biologists Tackle Visualization Challenges. PLoS ONE 8. https://doi.org/10.1371/journal.pone.0057990

Manca, D., Brambilla, S., Colombo, S., 2013. Bridging between Virtual Reality and accident simulation for training of process-industry operators. Advances in Engineering Software 55, 1–9. https://doi.org/10.1016/j.advengsoft.2012.09.002

Martín-Gutiérrez, J., 2017. Virtual Technologies Trends in Education. EURASIA Journal of Mathematics, Science and Technology Education 13. https://doi.org/10.12973/eurasia.2017.00626a

Mendes, D., Marangoni, C., Meneguelo, A.P., Machado, R.A.F., Bolzan, A., 2010. Educational simulator for multicomponent distillation research and teaching in chemical engineering. Computer Applications in Engineering Education 18, 175–182. https://doi.org/10.1002/cae.20134

Messaoudi, F., Simon, G., Ksentini, A., 2015. Dissecting games engines: The case of Unity3D, in: 2015 International Workshop on Network and Systems Support for Games (NetGames). Presented at the 2015 International Workshop on Network and Systems Support for Games (NetGames), IEEE, Zagreb, Croatia, pp. 1–6. https://doi.org/10.1109/NetGames.2015.7382990

Nowacki, P., Woda, M., 2020. Capabilities of ARCore and ARKit Platforms for AR/VR Applications, in: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (Eds.), Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 358–370. https://doi.org/10.1007/978-3-030-19501-4_36

O'Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R., 2011. Open Babel: An Open chemical toolbox. Journal of Cheminformatics 3, 33. https://doi.org/10.1186/1758-2946-3-33

Ouyang, S.-G., Wang, G., Yao, J.-Y., Zhu, G.-H.-W., Liu, Z.-Y., Feng, C., 2018. A Unity3D-based interactive three-dimensional virtual practice platform for chemical engineering. Computer Applications in Engineering Education 26, 91–100. https://doi.org/10.1002/cae.21863

Ozdemir, M., Sahin, C., Arcagok, S., Demir, M.K., 2018. The Effect of Augmented Reality Applications in the Learning Process: A MetaAnalysis Study. Eurasian Journal of Educational Research 18, 1–22. https://doi.org/10.14689/ejer.2018.74.9

Pagani, A., Henriques, J., Stricker, D., 2016. Sensors for location-based augmented reality the example of galileo and egnos. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 2016-Janua, 1173–1177. https://doi.org/10.5194/isprsarchives-XLI-B1-1173-2016

Paul, E.L., Atiemo-Obeng, V.A., Kresta, S.M. (Eds.), 2004. Handbook of industrial mixing: science and practice. Wiley-Interscience, Hoboken, N.J.

Pita Cañola, M.J., Pincay Durán, A.I., 2012. Obtención de jabón a partir de la extracción del aceite de bagazo de café (Master Thesis). Universidad de Guayaquil.

Porter, M.E., Heppelmann, J.E., 2017. A Manager's Guide to Augmented Reality. Harvard Business Review 95, 45–57.

Powell, J.B., 2017. Application of multiphase reaction engineering and process intensification to the challenges of sustainable future energy and chemicals. Chemical Engineering Science 157, 15–25. https://doi.org/10.1016/j.ces.2016.09.007

Reyero, I., Arzamendi, G., Zabala, S., Gandía, L.M., 2015. Kinetics of the NaOH-catalyzed transesterification of sunflower oil with ethanol to produce biodiesel. Fuel Processing Technology 129, 147–155.

Roediger, H.L., McDaniel, M.A., Brown, P.C., 2014. Make it stick: The science of successful learning. Harvard University Press.

Romrell, D., Kidder, L.C., Wood, E., 2014. The SAMR Model as a Framework for Evaluating mLearning. OLJ 18. https://doi.org/10.24059/olj.v18i2.435

Sagheb, S., Bahremand, A., LiKamWa, R., Lahey, B., 2019. SWISH: Shifting Weight-based Interfaces for Simulated Hydrodynamics in Mixed-Reality Fluid Vessels, in: Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '19. Presented at the the Thirteenth International Conference, ACM Press, Tempe, Arizona, USA, pp. 213–221. https://doi.org/10.1145/3294109.3300988

Salah, Z., Preim, B., Rose, G., 2010. An Approach for Enhanced Slice Visualization utilizing Augmented Reality : Algorithm and Applications.

Santos, I., Dam, P., Arantes, P., Raposo, A., Soares, L., 2016. Simulation Training in Oil Platforms, in: 2016 XVIII Symposium on Virtual and Augmented Reality (SVR). pp. 47–53.

Santos, M.E.C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., Kato, H., 2014. Augmented Reality Learning Experiences: Survey of Prototype Design and Evaluation. IEEE Transactions on Learning Technologies 7, 38–56. https://doi.org/10.1109/TLT.2013.37

Santos, P., Van Gerven, T., 2020. Aspen Hysys – Unity Interconnection. an Approach for Rigorous Computer- based Chemical Engineering Training, in: Pierucci, S., Manenti, F. (Eds.), European Symposium on Computer-Aided Process Engineering-30, 48th European Symposium on Computer Aided Process Engineering, Computer Aided Chemical Engineering. Elsevier.

Shipton, M., Holman, J., 1994. Fats and Oils, in: Unilever Educational Booklet: Advanced Series. Unilever, Cham.

Solmaz, S., Van Gerven, T., 2020. Integration of Interactive CFD Simulations with AR and VR for Educational Use in CRE, in: Pierucci, S., Manenti, F. (Eds.), European Symposium on Computer-Aided Process Engineering-30, 48th European Symposium on Computer Aided Process Engineering, Computer Aided Chemical Engineering. Elsevier.

Tekedere, H., 2016. Examining the Effectiveness of Augmented Reality Applications in Education: A Meta-Analysis. International Journal of Environmental & Science Education 11, 9469–9481.

Wu, H.K., Lee, S.W.Y., Chang, H.Y., Liang, J.C., 2013. Current status, opportunities and challenges of augmented reality in education. Computers and Education 62, 41–49. https://doi.org/10.1016/j.compedu.2012.10.024

Yilmaz, Z.A., Batdi, V., 2016. A meta-analytic and thematic comparative analysis of the integration of augmented reality applications into education. Education and Science 41, 273–289. https://doi.org/10.15390/EB.2016.6707

Zhu, Y., Fukuda, T., Yabuki, N., 2019. Integrating Animated Computational Fluid Dynamics into Mixed Reality for Building-Renovation Design. Technologies 8, 4. https://doi.org/10.3390/technologies8010004

Zitney, S.E., 2010. Process/equipment co-simulation for design and analysis of advanced energy systems. Computers & Chemical Engineering 34, 1532–1542. https://doi.org/10.1016/j.compchemeng.2010.02.011

Zoller, U., 2008. Handbook of Detergents, in: Part E: Applications. Surfactant Science Series. CRC Press.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: