Integration of Game Engine Based Mobile Augmented Reality Into a Learning Management System for Online Continuing Medical Education

Robin Horst[1], Dennis Fenchel[2], Reimond Retz[2], Linda Rau[1], Wilhelm Retz[1], Ralf Dörner[1]

Abstract: Physicians must participate in continuing medical education (CME) as part of the medical quality assurance. One possibility is to take online courses in their private living environment. These courses are mostly text- or video-based. Novel technologies such as mobile Augmented (AR) or mobile Virtual Reality (VR) are not yet established although their usage is not out of bounds in private homes anymore. Game engines can facilitate the authoring of applications that utilize VR/AR, as they provide many crucial functionalities out of the box. However, integrating the resulting VR/AR software in online CME courses is not trivial. In this paper, we investigate this integration into an existing learning management system (LMS) for online CME. In the example of a mobile AR application, we propose a system design that extends a course by a mobile AR part. We describe our implementation and how we transition users from their familiar web-interface on the desktop PC to a mobile AR application.

Keywords: Professional Education in Private Living Environments; Online Continuing Medical Education; System Design; Augmented Reality; Virtual Reality; Games Engineering; E-Learning

1 Introduction

Many professionals such as health professionals use their private living environments for continuous education in their field. Continuing medical education (CME) comprises training measures that serve to maintain and permanently update the professional competence of the medical profession. CME also serves as a part of medical quality assurance. These compulsory training activities for physicians are demanded by governmental-related organizations, such as the Accreditation Council for Continuing Medical Education (USA) or the Bundesärztekammer (Germany). Physicians are required to obtain a specific amount of credits, which can be earned through different activities. One of these activities is online training. Online CME is mostly conducted using technology that is already available in the private living environments of physicians, such as common desktop PCs, tablets, or smartphones. Different established media are used here to mediate information, for example, text, images, audio files, or videos. In the context of lifelong learning, private living environments are also expected to support modern technologies for learning, such as Augmented (AR) and Virtual Reality (VR) on mobile devices. Game engines such as Unity

[1] RheinMain University of Applied Sciences, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany, firstName.lastName@hs-rm.de
[2] health&media GmbH, Dolivostraße 9, 64293 Darmstadt, info@health-media.de

doi:10.18420/inf2020_88

[Un20] are suitable tools to support authors in creating applications for such technologies. However, integrating mobile AR and VR applications developed with game engines into existing systems for online CME and courses involves challenges for authors.

In this paper, we make the following contributions:

- We investigate the integration of mobile AR into an existing CME course that is based on a learning management system (LMS). Our goal is to extend a traditional online course with novel technology rather than replacing it entirely.
- In the example of a mobile AR application (app) that we built with Unity, we highlight crucial aspects of its integration in a CME course. Based on a prototype, we report and discuss lessons learned from the implementation process.
- Physicians usually participate in online courses on their desktop PC. To perceive the AR part, they must switch to our mobile AR app within the course. We propose a transitioning technique that guides users from the web-course to our mobile AR app.

This paper is organized as follows. The next section discusses related work. In Section 3, we describe the LMS that we integrate the mobile game engine technology into. In the fourth section, we present our system design that realizes the integration. Section 5 provides a conclusion and points out directions for future work.

2 Related Work

Web-based e-learning courses are widely used for CME. Various technologies can facilitate the authoring of such courses, including LMSs [CGB09]. There exists a variety of LMSs that serve different purposes. Moodle [DT03] (Modular Object-Oriented Dynamic Learning Environment) is an established multi-purpose and open source LMS. There are also more specific LMSs such as OLE [PDG20] (Open Learning Environment). OLE was developed to serve individual requirements of the local learning context of a university.

Different media can be integrated into LMS courses. Persia et al. [PDG20] show that courses using a variety of media highly support learning activities compared to solely text-based education. The user satisfaction was particularly improved when different multimedia were exploited. They also show how to integrate educational videos and smart text in their LMS courses but did not investigate AR or VR technology. Recent work by de Paiva Guimarães et al. [Pa17] propose a tool that utilizes educational browser-based AR content in the SCORM learning object standard. It creates AR learning objects and provides packaged AR applications to Moodle courses that can be executed utilizing a desktop PC and a webcam. It decouples LMSs and AR applications by distributing the applications through online repositories [Pa18]. Therefore, each AR application can be built with separate tools itself, as long as it is packaged with their tool in the SCORM standard. However, each time a course requires AR content, a novel AR application must be downloaded. With a shared

runtime environment, users only would have to download the content of an application. This facilitates the content delivery of AR applications within LMS courses.

Contrary to this approach by de Paiva Guimarães et al. [Pa17, Pa18], Coma-Tatay et al. [Co19] integrate the AR learning content without external applications or plug-ins. They provide a tool (FI-AR) that is based on the open-source software framework FIWARE [FI20]. It utilizes universal web-technology access to visualize AR content within their online courses. The AR content can be executed from both a desktop or a mobile AR runtime environment. Visualizing AR or VR content within an LMS environment brings the challenge of transitioning from traditional media to these immersive technologies. Dodd and Antonenko [DA12] propose to use signaling methodologies to facilitate the transition for users that enter the virtual worlds at the example of a desktop VR app. They provide users visual cues to guide them during this transition and use cues during the virtual learning activities until learners return to the traditional LMS content.

Generally, existing work on LMS and virtual technologies focuses on supporting separate custom-made AR and VR applications. Established authoring tools, such as game engines, are not considered. Furthermore, desktop PCs are targeted. Concepts for integrating AR and VR that runs on a mobile device within online courses are lacking.

3 Existing Continuing Medical Education System

Our existing *arztCME* [hG] e-learning framework for CME comprises both LMS and content management system (CMS) functionality aligned with specific requirements to comply with the certification of CME. The foundation of the CME system is a CMS based on WordPress that offers PHP and MySQL interfaces. It was extended with plugins to provide LMS functionality. The LMS features support authors to create CME courses and tests for assessing the learning success of physicians to give them credits. One CME course takes approximately 45 minutes and is consists of several 'pages' of learning content.

Physicians usually participate through web-browsers on a desktop PC. Some use mobile devices. In addition to static course representations as PDF documents, courses can be represented as multimedia HTML realization within our e-learning framework (Fig. 1). Established media in this realization are images, videos, and texts. Our web-technology can directly integrate them in the HTML environment.

4 Integration of a Mobile Unity Augmented Reality App

There exist ways to include AR and VR technologies directly in a web-browser (e.g., [Li04, La19]). As our online CME courses already take place in a web-environment, a naive integration of these technologies makes them available directly in the web-browser. Unity also offers such a solution. It provides a web-player which can run Unity applications directly

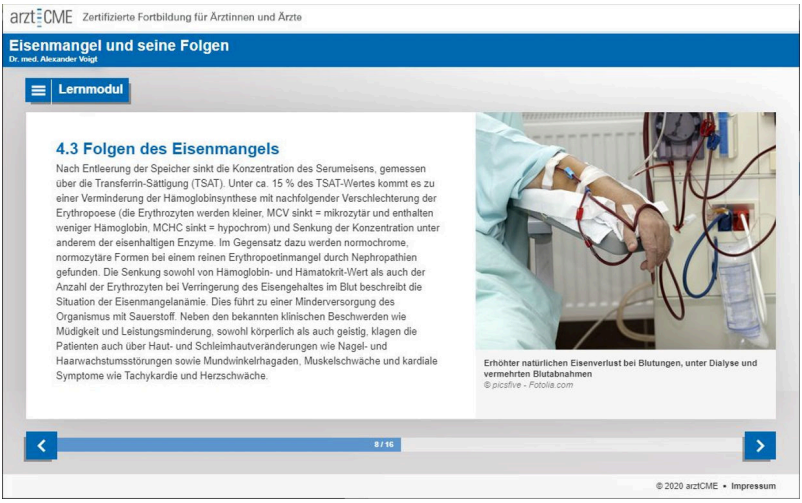

Fig. 1: The view on one page of a traditional online CME course for German-speaking physicians.

in the web-browser. However, this player and specific AR and VR features are designed for web-browsers that run on desktop PCs. Access to these players through web-browsers on mobile devices is possible but not entirely supported. The performance can suffer when using computationally intensive AR or VR functionalities through a mobile web-browser due to the complexity of additional rendering steps that the browser needs to go through [BRR16]. Running these technologies in dedicated apps on a mobile device can increase the performance.

To provide an AR experience at a specific point during the course, users are required to switch from the traditional CME course that runs on their desktop PC to our mobile app. To facilitate the transition between the two environments and to induce the user to do so, we integrated an easily accessible onboarding approach within our system design (Fig. 2). Using QR codes (Fig. 3), an established linking technique within the mobile domain, we can transfer users from a page within the course to the AR app. Furthermore, using dynamic QR codes, created with JavaScript at runtime, we are also able to include context-specific information during the transitional phase. This provides an opportunity to attach a dynamic session ID, that can be used in further steps of the onboarding process to retrieve user-specific data, such as the name of the course that the user is in and the current state within the course.

To proceed in the course, users scan the QR code with their mobile device. In case the users do not have the corresponding AR app already installed on their mobile device, they are redirected to the store app of their mobile operating system (OS) (e.g., Appstore for iOS and Google Play Store for Android). Here they can download our AR app. In case the app already exists on the mobile device, it is opened directly instead. We use Firebase Dynamic

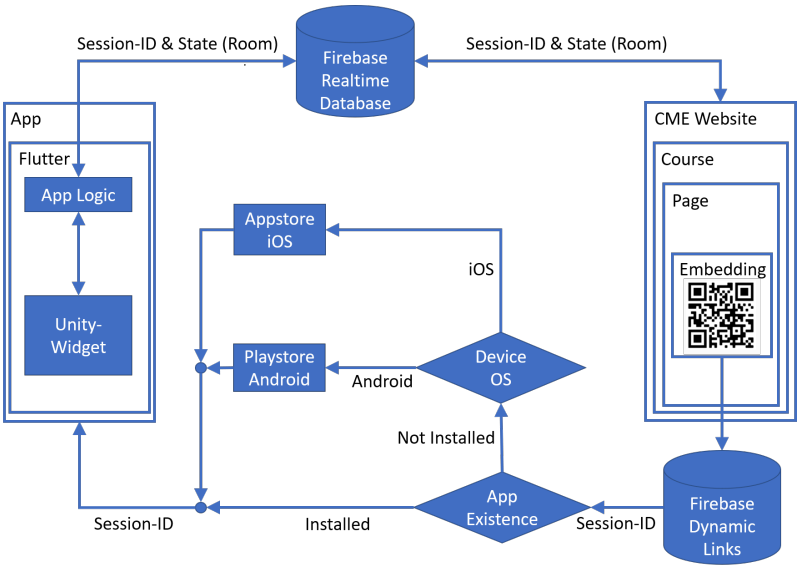

Fig. 2: Our system design extends an existing course on one specific page with mobile AR content. An onboarding process facilitates the transitioning from the website on a desktop PC to the mobile app.

Links [Go20a] to realize these re-directions. Firebase Dynamic Links (FDL) can store the session IDs from the QR code dynamically and transfer them through the installation process to our mobile app. The app can use the session ID to get further information about the user from a corresponding Firebase Realtime Database (FRTD) [Go20a] that is also connected to our arztCME LMS (Fig. 2). It uses an open web socket subscription model for the connection. Both the app and the website can access and manipulate entries in the database in realtime. A basic session management system and IDs that track the logged-in users were already included in the original LMS. This has been extended to include current AR content and the information required for it.

Now users can be directed from a course to a separate AR application for each course that extends it. However, it can be cumbersome for users when they have to download a dedicated AR app for each specific course they participate in. To counteract this, we decided to build a single app that serves as an AR platform for all courses. When using one application that provides the content for all available courses, it can still be tiresome for users to open the app and search the right content for their current course. We make use of the realtime connection of our app to the web-based LMS to open the app at a specific state directly after scanning the QR code instead of just starting the app. The session ID that we transferred throughout the onboarding process with FDLs can be used to query necessary information such as the specific location within a course. Alternatives without FDL would require scanning the QR code multiple times (e.g., at first for linking to the store or opening the app and then for opening the right content within the app).

Fig. 3: A screenshot of a page within a course that includes a QR code used for transitioning between the desktop web-browser and our mobile game engine app.

The app development itself can be divided into two separate blocks. All AR-related functionality was developed in a Unity project, whereas all mobile-related functionalities (e.g., mobile app front end) were developed in a separate Flutter [Go20b] project. The advantage of using Flutter for mobile app development is that it supports cross-platform development. It also has the advantage that user interfaces (UIs) developed with Flutter look more conventional to UIs of regular mobile apps in contrast to mobile game UIs build with Unity. Furthermore, Flutter offers interfaces to a variety of common tools that facilitate mobile app development. However, splitting the development requires the integration of the Unity-based AR functionality within the Flutter app environment. This is a challenging task since Unity development for mobile devices will normally build separate APK or IPA files which cannot easily be integrated into a Flutter app. As of Unity version 2019.3, Unity allows to build a Unity project as a library to be included within other apps. In a Flutter-App, the embedded Unity library can directly be used as a Widget to display its content. This Unity Widget can be developed in the Unity authoring environment without further restrictions except one. Unity Widgets are not directly suitable for integration in a Flutter multi-platform code basis. For example, it is only supported to display AR Widgets in full-screen- mode. We also had to establish an asynchronous communication interface between the Flutter environment and the Unity Widget that is used as a plugin in Flutter. This communication is used to start the app directly with the content while correctly initializing the application's state. For example, separate Unity scenes can be addressed, or the Flutter app can tell the Unity Widget which image targets for the AR functionality to use. The Unity Widget also sends information through the Flutter environment back to the LMS. For example, this is used to share quiz results for later use in the evaluation of the CME course to award the credits for physicians when the course is successfully finished. The resulting app is illustrated in Fig. 4.

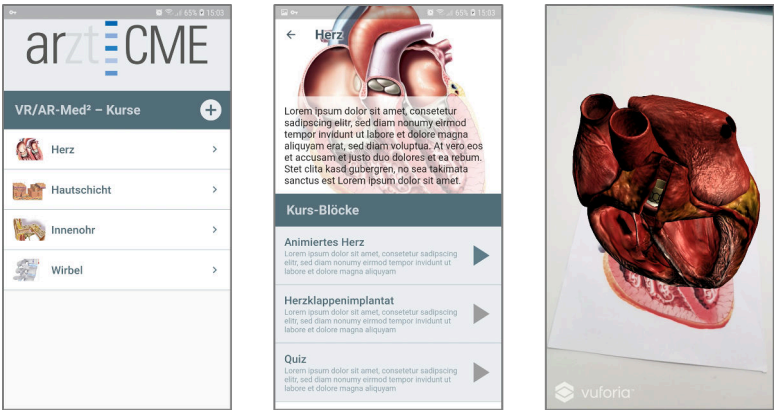

Fig. 4: Left: Start screen of our AR app. Users can select a course. Middle: A course about the functionality of the human heart is selected. Right: A heart displayed with AR technology.

5 Conclusion and Future Work

In this paper, we investigated the integration of mobile AR into an existing LMS for online CME. In the example of extending one of our courses with an AR functionality, we described how we realized this integration. We propose a system design and a suitable onboarding approach that facilitates the transition from the traditional online CME course on a desktop PC to its AR part on our mobile AR app.

Future work relating to our contribution can be divided into two categories. Now that the mobile AR can technically be integrated within our courses, a user test with physicians can be conducted to draw conclusions on the acceptance of such technologies within the CME domain. As these technologies are novel to CME, certification of such content must also be addressed and also privacy aspects of our onboarding approach must be considered. As for now, we do not know of any privacy issues regarding Google's Firebase technology. Secondly, we will explore how new AR content can be delivered to users efficiently. As of now, our app includes the AR content for all courses but when more and more courses get an AR extension, the size of our app will grow vastly. Content can be delivered on-demand. However, delivering novel content to a ready-built Unity application is not a trivial task. Furthermore, virtual assets such as 3D models can have large file-sizes so that downloading on demand may interrupt the learning flow of a course depending on the Internet connection of the mobile device.

Acknowledgments
This project (HA project no. 690/19-10) is financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

Bibliography

[BRR16] Butcher, Peter WS; Roberts, Jonathan C; Ritsos, Panagiotis D: Immersive analytics with webvr and google cardboard. Posters of IEEE VIS, pp. 30–32, 2016.

[CGB09] Cirulis, Arnis; Ginters, E; Brigmanis, K: Virtual reality's technologies use in e-learning. In: Proceedings of the 8th WSEAS International Conference on E-Activities, WSEAS Press, Puerto De La Cruz, ESP. pp. 148–153, 2009.

[Co19] Coma-Tatay, Inmaculada; Casas-Yrurzum, Sergio; Casanova-Salas, Pablo; Fernández-Marín, Marcos: FI-AR learning: a web-based platform for augmented reality educational content. Multimedia Tools and Applications, 78(5):6093–6118, 2019.

[DA12] Dodd, Bucky J; Antonenko, Pavlo D: Use of signaling to integrate desktop virtual reality and online learning management systems. Computers & Education, 59(4):1099–1108, 2012.

[DT03] Dougiamas, Martin; Taylor, Peter: Moodle: Using learning communities to create an open source course management system. In: EdMedia+ Innovate Learning. Association for the Advancement of Computing in Education (AACE), pp. 171–178, 2003.

[FI20] FIWARE Foundation: FIWARE. https://www.fiware.org/, 2020. Accessed: 13.04.2020.

[Go20a] Google: Firebase Realtime Database and Dynamic Links. https://firebase.google.com/docs, 2020. Accessed: 13.04.2020.

[Go20b] Google: Flutter. https://flutter.dev/, 2020. Accessed: 13.04.2020.

[hG] health&media GmbH: arztCME. https://www.arztcme.de/year={2020},. Accessed: 13.04.2020.

[La19] Lam, Kit Yung; Lee, Lik Hang; Braud, Tristan; Hui, Pan: M2A: A Framework for Visualizing Information from Mobile Web to Mobile Augmented Reality. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEE, pp. 1–10, 2019.

[Li04] Liarokapis, Fotis; Mourkoussis, Nikolaos; White, Martin; Darcy, Joe; Sifniotis, Maria; Petridis, Panos; Basu, Anirban; Lister, Paul F et al.: Web3D and augmented reality to support engineering education. World transactions on engineering and technology education, 3(1):11–14, 2004.

[Pa17] de Paiva Guimarães, Marcelo; Alves, Bruno; Martins, Valéria Farinazzo; dos Santos Baglie, Luiz Soares; Brega, José Remo; Dias, Diego Colombo: Embedding augmented reality applications into learning management systems. In: International Conference on Computational Science and Its Applications. Springer, pp. 585–594, 2017.

[Pa18] de Paiva Guimarães, Marcelo; Alves, Bruno Carvalho; Durelli, Rafael Serapilha; de FR Guimarães, Rita; Dias, Diego Colombo: An Approach to Developing Learning Objects with Augmented Reality Content. In: International Conference on Computational Science and Its Applications. Springer, pp. 757–774, 2018.

[PDG20] Persia, Fabio; D'Auria, Daniela; Ge, Mouzhi: Improving Learning System Performance with Multimedia Semantics. In: 2020 IEEE 14th International Conference on Semantic Computing (ICSC). IEEE, pp. 238–241, 2020.

[Un20] Unity Technologies: Unity game engine. https://unity.com/, 2020. Accessed: 13.04.2020.