Technical Realization of a 3D Immersive Serious Game for AI Medical-Device Certification Training

A Case Study from the TEF-Health Project

Peter Sykora[1,*], Miroslav Benco[1], and Robert Hudec[1]

[1]Department of Multimedia and Information-Communication Technology, Faculty of Electrical Engineering and Information Technology, University of Žilina, Žilina, Slovakia
*peter.sykora@uniza.sk

Abstract. Certifying an AI-based medical device for the EU market is a complex, multi-stakeholder regulatory process that is difficult to teach through lectures alone. Within Work Package 5 of the TEF-Health project, we are developing a browser-playable 3D serious game that lets engineering students experience this certification process first-hand, as a small team seeking market access for their device. This paper reports the technical realization of the game's 3D immersive component. We review related work on serious games and immersive training, then show how the certification scenario's recurring learner actions – talking to people, e-mail correspondence, form filling and travelling between places – were classified and mapped onto four game mechanics, implemented in Unity with office and hospital environments, an Ink-scripted dialogue system, and a diegetic in-game smartphone bundling chat, mail and tasks. A REST client-server layer synchronises game state with a companion web platform that lets a teacher run the certification-authority role from an ordinary browser without ever opening the 3D client. Interaction is reduced to one-click/one-tap navigation for a non-gamer audience, with colour-blind palettes and localisation for accessibility. We report the current implementation status, our planned evaluation, and the limitations of the approach.

Keywords: Serious Games · Immersive Learning · 3D Game Design · Medical Device Certification · Human-Computer Interaction

1 Introduction

Placing an artificial-intelligence-based medical device on the EU market means satisfying two overlapping regulatory regimes: the Medical Device Regulation (MDR) and, for most AI-enabled devices, the Artificial Intelligence Act [7], which classifies AI safety components of MDR-regulated devices as high-risk and layers additional conformity-assessment and documentation duties on top of existing

© The Author(s) 2026
K. Hollá et al. (eds.), *Proceedings of the SAFAR Scientific Conference 2026 – Building Awareness Through Immersive Learning (SSC-BAIL 2026)*, Atlantis Highlights in Social Sciences, Education and Humanities 57,
https://doi.org/10.2991/978-94-6239-817-7_6

CE-marking. For engineering students, this process is usually taught abstractly, from slides and legal texts, which conveys little of its lived complexity.

Serious games – games whose primary purpose is training, education or simulation rather than entertainment [1, 2] – are an established way to let learners rehearse this kind of procedural, cross-organisational process in a safe, repeatable environment, and gamification, the use of game-design elements in a non-game context [3], helps sustain motivation over an otherwise dry compliance topic. Sect. 2 reviews this body of work.

This work is carried out in TEF-Health, a Testing and Experimentation Facility for Health AI and Robotics funded by the European Union's Digital Europe Programme (grant No. 101100700), building a pan-European reference infrastructure for AI and robotics in healthcare across nine countries [8]. Within TEF-Health, Work Package 5 at the University of Žilina develops a serious-game-based training platform in which student teams play through a certification scenario set at a fictitious hospital, corresponding with mentors and regulatory contacts to bring an AI-based device to market. The platform has two parts: an immersive Unity 3D game and a companion web platform. This paper reports on the technical realization of the 3D immersive component, rather than on the web platform itself or on a summative evaluation of learning outcomes, discussed as future work in Sect. 7.

2 Related Work

Serious games have been used for professional training since at least the 1970s [1], and Michael and Chen's definition of games whose primary purpose is not entertainment still frames most design decisions [2]. Gamification research has since generalised the same vocabulary – points, quests, feedback loops – to non-game contexts such as compliance training [3], of which our certification process is an example.

Most published evaluations of immersive health training concern clinical skills: a review by Kyaw et al. found VR-based education measurably improves health professionals' knowledge and skills [5]. Regulatory process training, our focus here, is comparatively under-served despite a similar procedural, multi-stakeholder structure. On the interface side, Bartle's taxonomy of player motivations [4] places our audience – engineering students playing as a mandatory course component – closer to a socialiser than an achiever seeking mechanical depth, motivating the one-click scheme of Sect. 3; for the interface itself we follow Fagerholt and Lorentzon's diegetic/non-diegetic taxonomy [6], placing interface state inside the game world (Sect. 5) rather than in a heads-up display.

What distinguishes this work from existing educational and serious games is not any single mechanic but the combination of target domain and architecture: to our knowledge, no published serious game specifically targets the AI medical-device certification pathway. Our contribution is therefore twofold: a repeatable scenario-to-mechanic design method that classifies a real regulatory process into a small set of interaction types and maps each to a generic, reauthorable me-

chanic (Sect. 3), and a REST architecture that lets a teacher supervise a session from an ordinary browser without any 3D or game-engine skills (Sect. 6) – unlike existing compliance-training games, which typically hard-code their content, or clinical VR trainers, which typically require the teacher inside the simulated environment.

3 The TEF-Health Serious Game: Scenario and Design Workflow

The game's narrative follows a small team developing an AI-based medical device – for example, an AI-assisted radiology triage tool – that must be piloted and certified for use at a partner hospital before it can be placed on the market. The player's team gathers requirements from clinical stakeholders, corresponds by e-mail with regulatory and hospital contacts such as a named radiology lead, negotiates a pilot deployment, and works through the paperwork this involves, mirroring the structure – if not the full legal detail – of a real MDR/AI Act conformity-assessment pathway.

We followed a six-stage workflow to turn this scenario into a playable system. (1) We wrote the scenario summarised above. (2) We classified every recurring learner action into a small number of interaction types. (3) We designed one game mechanic per type, summarised in Table 1. (4) We designed the GUI/UX around a single, simple input – one-click/one-tap – appropriate for an audience of engineering students rather than experienced gamers (Sect. 2). (5) We implemented client-server support (login, state synchronisation, e-mail) so a session can run over the network in teams. (6) We built a lightweight web platform so a teacher can run the certification-authority role from an ordinary browser, described further in Sect. 6.

We classified actions using two criteria: the communication modality involved – synchronous spoken interaction, asynchronous written correspondence, physical movement between locations, or document completion – and whether the action recurred across multiple stages of the scenario, since one-off actions were absorbed into a neighbouring mechanic rather than given dedicated support. Applying these criteria to the scenario script yielded exactly four recurring modalities, each mapped to one mechanic in Table 1: synchronous spoken exchanges became branching dialogue, asynchronous written exchanges became virtual e-mail, document completion became task/checklist tracking, and physical movement became point-and-click navigation. Team coordination, which is orthogonal to the regulatory scenario itself, was layered on top as a supporting mechanic (team chat) rather than treated as a classified scenario action. Classifying by communication modality, rather than by a domain-specific criterion such as regulatory phase or document type, is what lets the same four mechanics be reused for a different device or jurisdiction by re-authoring content alone, discussed further in Sect. 7.

Table 1. Mapping of classified scenario actions to game mechanics and their implementation.

Scenario action	Mechanic	Implementation
Talking to people	Dialogue	Ink-scripted dialogue trees with NPCs
E-mail correspondence	Virtual e-mail	Mail app: sender, subject, files
Form filling / paperwork	Task tracking	Checklist app, completion per team
Travelling between places	Navigation	Click-to-move, minimap, teleport

4 System Architecture

Fig. 1 shows the resulting architecture. The Unity client [9] – exported to WebGL, with Windows and VR builds planned – owns the 3D simulation: environments, NPCs, dialogue and the in-game interface of Sect. 5. It talks to a REST backend that also serves a web platform used by students and teachers to launch sessions and manage users, tasks and teams. State is organised per team rather than per user: endpoints such as `signIn`, `taskStateUpdate` and `createChatMessage` let the client push state changes, while `teamTasks`, `teamMembersStatus` and a timestamped `checkTeamChatMessages` poll let both the client and the web platform read current team state.

We deliberately avoided a persistent socket connection: short-interval polling of timestamped endpoints is simpler to reason about, degrades gracefully on constrained school networks, and keeps the WebGL build small. Because every piece of pedagogically relevant state is available through the same API, the teacher dashboard can mirror what is happening inside the 3D world without the teacher ever opening the game client, directly realising workflow step 6.

On security, every endpoint other than `signIn` requires a bearer token issued at login; the token also encodes the caller's role, so the backend can restrict teacher-only operations such as team and user management to authenticated teacher accounts, and all traffic runs over HTTPS. Team-state synchronisation builds on the same timestamped-polling pattern used for chat: each client tracks the timestamp of its own last successful read of a given endpoint and requests only records newer than that, so state converges across the Unity client and both web portals within one polling interval without a shared session or broadcast mechanism. Because the fields that matter for grading and oversight – task completion, chat messages, position – are append-only or owned by a single team member, conflicting writes to the same field are rare in practice, and where they occur the backend applies a last-writer-wins rule per field rather than a more complex merge. On network failure, the client queues outgoing state changes locally and retries with backoff until the server acknowledges them, so a dropped connection on lecture-hall Wi-Fi degrades to a delayed sync rather than lost progress, and content that needs no server round trip – navigation, dialogue already downloaded to the client – remains playable offline in the meantime.

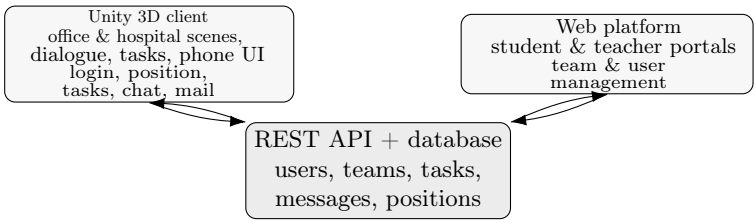


Fig. 1. Client-server architecture linking the Unity 3D game client and the web platform through a shared, team-oriented REST API.

5 Realizing the 3D Environment

5.1 Asset Pipeline and Environments

3D assets are authored in Blender and imported into Unity. Two environment sets are complete: an office (open-plan desks, meeting room, manager's office, kitchen) representing the students' own workplace, and a hospital (exterior building, ambulance bay, reception and waiting area) representing the site where the device is piloted (Fig. 2a); two further asset sets – a dialogue-box UI skin and the in-game phone menu of Sect. 5.4 – are shared across locations. Models are kept low-to-mid poly with baked, PBR-style materials, keeping load time within browser-friendly bounds for the WebGL build.

5.2 Navigation and Wayfinding

Travelling between places, one of the classified action types in Table 1, is realised through click/tap-to-move locomotion: the player clicks a point on the ground to walk there and drags to look around (Fig. 2a), rather than the WASD scheme conventional in first-person games. This keeps the game reachable with a single input and transfers directly to touch-screen and VR builds planned for later releases. A persistent circular minimap orients the player locally, while a location-switch ("teleport") system lets a team jump directly between major locations such as the office and the hospital, avoiding tedious manual traversal across a multi-session course.

5.3 Dialogue and NPC Interaction

Talking to people is implemented as a branching dialogue system authored in Ink [10], inkle's narrative scripting language, which compiles human-written dialogue trees to a lightweight runtime consumed by Unity. Writing dialogue in Ink lets non-programmer scenario writers iterate on NPC conversations – for example, a hospital receptionist directing the player to a named radiology contact (Fig. 2b) – independently of engine code, keeping interviews and negotiations directly playable.

5.4 Diegetic In-Game Interface

Rather than overlay a conventional heads-up display on the 3D view, the remaining mechanics – chat, virtual e-mail and task tracking – are housed inside a smartphone the player's avatar carries, following the diegetic-UI approach of Sect. 2. Clicking the phone opens a single grid of app icons (Fig. 2c); each app mirrors a real regulatory-communication channel: Mail exposes sender, subject and file attachments like real correspondence with the certification authority, Tasks lists the certification steps as a checklist (Fig. 2d), and Chat carries team coordination that the supervising teacher can also read from the web platform (Sect. 6). Collapsing every system tool into one familiar object keeps the control surface consistent with the one-click philosophy of Sect. 3 and minimises the gaming literacy required from students who are not necessarily gamers.

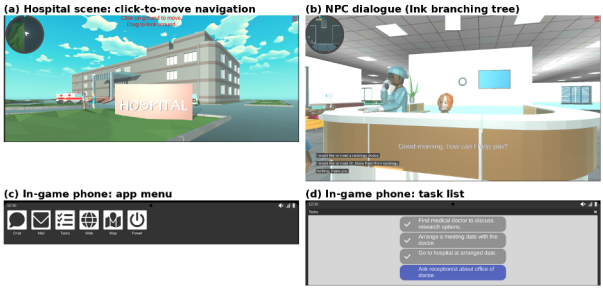


Fig. 2. The 3D immersive component: (a) hospital environment reached through click-to-move navigation, with a location marker and minimap; (b) an Ink-scripted branching dialogue with a hospital NPC; (c) the diegetic in-game phone's app menu; (d) its task list.

6 Accessibility, Localization and Teacher Oversight

UI text is localised at the engine level (English currently supported, with the text-table architecture in place for further languages), and the settings app offers several colour palettes, including ones for colour-vision-deficient players, in line with colour-universal-design practice [11].

Because every piece of teacher-relevant state is exposed through the REST layer of Sect. 4, a teacher supervises a session entirely from a companion web dashboard (Fig. 3) rather than the 3D client: team rosters, an activity log, task completion, and the team's chat transcript are all visible and, where relevant, editable from a single screen, alongside a document exchange for reference material or submissions. This directly realises workflow step 6, and means the accessibility and localisation choices above apply uniformly across the Unity client and both web portals, since all three read and write the same underlying team data.

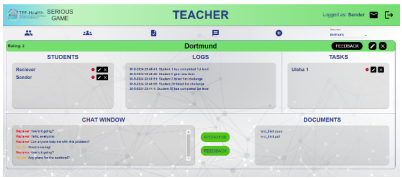


Fig. 3. The companion teacher web platform for one team: roster, activity log, task list, chat transcript and document exchange, all editable without opening the 3D client.

7 Implementation Status, Evaluation and Limitations

The game is at version 0.4 (alpha). *Implemented* and playable end-to-end are one-click navigation, Ink dialogue, chat, e-mail, tasks, an in-game browser, the phone menu, and the REST endpoints of Sect. 4. *In progress* are the location-switch system and remaining level design; an inner-monologue narration mechanic is being explored. *Designed but not yet built* is full scenario coverage across all locations and characters. Remaining work before deployment includes debugging, export and postproduction. We then plan to evaluate the game with a cohort of engineering students enrolled in multimedia- and biomedical-technology courses at the University of Žilina who have not yet received formal regulatory-affairs training, recruited on a voluntary basis alongside their regular coursework. Knowledge transfer will be measured with a short pre-test/post-test on the steps and stakeholders of the certification pathway, complemented by the document-comparison task described above as a practical proxy for procedural understanding. User experience will be assessed with the System Usability Scale [12] administered immediately after play, together with a short Likert-scale questionnaire on engagement and perceived realism and a brief debrief interview to surface issues the quantitative instruments miss.

The design trades mechanical depth for accessibility: one click or tap drives every interaction, and the mechanics in Table 1 are deliberately generic, so new scenarios need only new dialogue and e-mail content, not new code – the game teaches the *shape* of certification more than deep regulatory expertise, a reasonable target for an introductory course. The main limitation is that, at alpha stage, the system has not been evaluated with real students, so we cannot yet report learning outcomes; the evaluation above is intended to close this gap. Scope is also limited to two locations and one language, and the REST/polling architecture would need revisiting for many concurrent teams; unlike the clinical-skills VR literature of Sect. 2, our administrative scenario only partially benefits from that evidence.

8 Conclusion

We presented the technical realization of the 3D immersive component of a TEF-Health serious game that turns a written AI medical-device certification scenario

into a playable world: a scenario-to-mechanic mapping covering dialogue, e-mail, tasks and travel; a Unity/Blender implementation with one-click navigation and a diegetic in-game phone; and a REST client-server layer that lets the same team data drive the game and a teacher web platform simultaneously. The approach is intentionally low-tech at the interaction level, so complexity sits in the certification content rather than the controls, and it is designed to be extended toward classroom deployment and the evaluation outlined in Sect. 7.

Acknowledgments. This work was carried out within Work Package 5 of the TEF-Health project (Testing and Experimentation Facility for Health AI and Robotics) at the University of Žilina. TEF-Health has received funding from the European Union's Digital Europe Programme (DIGITAL) under grant agreement No. 101100700. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA); neither can be held responsible for them.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Susi, T., Johannesson, M., Backlund, P.: Serious games – an overview. Technical Report HS-IKI-TR-07-001, University of Skövde, Skövde (2007)
2. Michael, D., Chen, S.: Serious Games: Games That Educate, Train, and Inform. Muska & Lipman/Premier-Trade, Boston (2005)
3. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining "gamification". In: Proc. 15th Int. Academic MindTrek Conf. pp. 9–15. ACM, New York (2011)
4. Bartle, R.: Hearts, clubs, diamonds, spades: players who suit MUDs. Journal of MUD Research **1**(1) (1996)
5. Kyaw, B.M., Saxena, N., Posadzki, P., Vseteckova, J., Nikolaou, C.K., et al.: Virtual reality for health professions education: systematic review and meta-analysis by the Digital Health Education Collaboration. Journal of Medical Internet Research **21**(1), e12959 (2019)
6. Fagerholt, E., Lorentzon, M.: Beyond the HUD: User Interfaces for Increased Player Immersion in FPS Games. Master's thesis, Chalmers University of Technology, Gothenburg (2009)
7. Regulation (EU) 2017/745 on medical devices, OJ L 117 (5 May 2017); Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act), OJ L, 2024/1689 (12 July 2024)
8. TEF-Health Consortium: Testing and Experimentation Facility for Health AI and Robotics. https://tefhealth.eu (2023)
9. Unity Technologies: Unity real-time development platform. https://unity.com
10. inkle: ink – inkle's narrative scripting language. https://www.inklestudios.com/ink/
11. Okabe, M., Ito, K.: Color universal design (CUD). https://jfly.uni-koeln.de/color/ (2008)
12. Brooke, J.: SUS: a quick and dirty usability scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, I.L. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, London (1996)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

