AR TutorialKit: an Augmented Reality Toolkit to Create Tutorials

Federico Meloni[1,*], Alessandra Perniciano[1,*], Giulia Cerniglia[1,*], Vittoria Frau[1] and Lucio Davide Spano[1]

[1]*University of Cagliari, Department of Mathematics and Computer Science, Via Ospedale 72,09124, Cagliari, Italy*

Abstract

Augmented Reality (AR) is a widely used technology in fields such as medicine, engineering, and architecture, and is also prevalent in social media platforms like Snapchat, Instagram, and TikTok. In recent years, the availability of AR applications and improvements in hardware have made it affordable for educational training in various disciplines. However, limited options are available for the general construction of AR tutorials in the literature. Most solutions are specific for particular contexts, such as medical procedures or industry-specific tasks. This paper proposes an AR toolkit that enables novice programmers to create tutorials without topic restrictions. Our aim is to keep improving TutorialKit in such a way that it can be used flexibly and effectively in a variety of different contexts, enabling it to meet the diverse needs and requirements of users.

Keywords

End-User Development, Augmented Reality, Training, Education, Tutorial

1. Introduction

Nowadays, Augmented Reality (AR) interfaces are widespread, finding applications in medicine [1, 2], engineering [3, 4] or architecture [5, 6]. In our daily life, AR features are pervasive in social media platforms like Snapchat[1], Instagram[2] or Tiktok[3], which allow users for instance to apply augmented reality filters to their faces or surroundings. A key AR capability is enhancing the user's sensory experience by seamlessly integrating virtual elements within the real-world environment [7]. This characteristic makes AR well-suited for use in training. In the past few years, the availability of AR toolkit and enhancements in hardware have made it easier to implement AR in educational training. As a result, the use of AR has become affordable for education and training across various disciplines such as industry [8], medicine [9], the military [10], agriculture [11] etc. Augmented Reality is an innovation that has the potential to change where and when education and training take place significantly.

In the literature, there are not many options available to ease the construction of Augmented

IS-EUD 2023: 9th International Symposium on End-User Development, June 06-08, 2023, Cagliari, Italy

*These authors contributed equally.

federico.meloni3@unica.it (F. Meloni); alessandra.pernician@unica.it (A. Perniciano); giuliezz98@gmail.com (G. Cerniglia); vittoria.frau@unica.it (V. Frau); davide.spano@unica.it (L. D. Spano)

[1]https://ar.snap.com/

[2]https://sparkar.facebook.com/ar-studio/learn/articles/people-tracking/people-effects-introduction

[3]https://effecthouse.tiktok.com/

Reality tutorials in the general case. Most of the solutions are extremely focused on a particular domain and are created ad-hoc for a specific context. For example, in the medical field, several applications explain how to do activities such as sterilisation of tools [12] or actual simulations of operations such as thoracotomy[13], but the infrastructure behind each application (medical or other) is strictly personalised for that context.

Our project aims to provide an Augmented Reality toolkit that enables novice developers to rapidly create tutorials in different domains, without the burden of learning all the required aspects of a full-fledged AR toolkit. The toolkit is flexible and applicable in different contexts and domains.

2. Related Work

In the past few years, numerous AR-based training tutorials have been presented[11, 14, 15, 16]. An example of an AR tutorial is provided by the American Fuel & Petrochemical Manufacturers (AFPM), who created a digital toolkit with simulations using virtual and augmented reality to bridge the knowledge gap across employee generations and deal with competence training issues [17]. Interesting work was provided by Eckhoff et al.[18] who developed TutAR which automatically converts videos of hand procedures into 3D AR tutorials with minimal user input. Despite the system being a good example of creating AR tutorials without any programming knowledge, it only accepts videos with hand movements and, when a tutorial is done, it can not be modified. Another fascinating system is Meta-Ar-App[19], which is an authoring platform for collaborative AR that enables teachers to create AR tutorials. One of the most important limitations of this work is related to the tutorial context. Indeed, it is possible to create tutorials only with electronic circuit objects.

Our work enables novice programmers to create AR tutorials without being restricted to a specific field. The structure of tutorials supported by our toolkit is simple, composed of atomic and elementary steps that can be applied in various contexts. As a result, our toolkit allows novice programmers to create AR tutorials that can be tailored to the needs of individual users.

3. The AR Tutorial Kit

As mentioned earlier, our work aims to make augmented reality tutorial creation easier for less experienced programmers, by providing tools that can be adapted as much as possible to any topic, without necessarily having to write complex code. To do this, we created a simple AR tutorial abstraction, consisting of two key concepts: the task and the tool.

- A **Task** is an action that a user must take to finish the tutorial. The aim of the tutorial is to guide the user in achieving a goal by completing all tasks.
- A **Tool** is an object that the user controls to carry out a task.

TutorialKit allows defining tutorials by a structured encoding of tasks and tools. The underlying toolkit support interprets the information and generates the interface for performing the tutorial. The structured encoding consists of a JSON file. For tasks, it includes the following information: title, description, media's URL (image or video, optional), and tool name (identifying the tool to be used for the task, optional). Every task has a tool, which is an object that will be used to do the task. A tool is defined as follows: name, image (needed for the tracking), and the dimension of the hint. Tasks may be grouped in unordered and ordered lists, which express their temporal relationships. Ordered lists define tasks that must be performed in sequence (e.g. executing a recipe), whereas when ordering is not mandatory (e.g. decorating a Christmas Tree), we use unordered lists.

We describe the support to the execution of the tutorials defined by this structure in Section 3.1, while the show the user interface in Section 3.2.

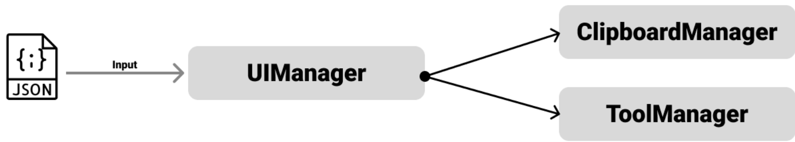


Figure 1: Structure of TutorialKit

3.1. UIManager

The implementation of the toolkit relies on the AR Foundation package[4] in Unity 3D[5]. The solution consists of three modules:

- The **UIManager** is the main class, and it provides represents the entry point for library users and coordinates communications between the other classes;
- The **ClipboardManager** handles information displayed on the clipboard, such as the tutorial's progress of the user, the success or failure callbacks etc.;
- The **ToolManager** identifies the tool in the real environment when performing a task.

The novice developer imports the UIManager in his/her application to implement a tutorial, providing the aforementioned JSON description of tasks and tools. The implementation also requires callback functions for the correct or wrong performance of a given task. After loading the JSON file, the novice developer links the callbacks to trigger positive or negative feedback on a given task, while the interface is completely generated by the underlying toolkit. The feedback consists of a green check or a red cross on the clipboard, for a positive and negative outcome, respectively.

In future work, we would like to open the development of tutorial to non-programmers, through an authoring interface allowing to specify the structure (i.e., the information in the JSON file) and to provide a low or no-code definition of the completion callbacks when a task has been performed correctly or not. This may be achieved by including a set of predefined triggers and actions (e.g., setting an object position or visibility on collision).

[4]https://unity.com/unity/features/arfoundation
[5]https://unity.com/

3.2. Interface

While the UIManager represents the programming interface for the novice developer, the ClipboardManager and the ToolManager are two modules responsible for updating the AR interface. The clipboard is the core of the interface, and it displays the task information and some optional information, which depends on the current task. The Tool Manager is responsible for highlighting the elements required for completing the task in the AR environment, including the item that needs to be handled to finish the task and any external tools that will help do so. Suppose we have a tutorial for making a cake in which one of the task is to whip the cream. Our toolkit could highlight the cream as a necessary object for completing the task, as well as the whisk, which is considered an external object (since an electric mixer might also be used).

3.2.1. Clipboard

The Clipboard is placed in the virtual space of the scene, and it is depicted as a real clipboard. Users interact with an object they are familiar with in suggesting guidance for completing actions. The information in the clipboard contains the task title, its description, a hint about the tool to be used, warnings in case the task can not be performed and feedback on correct or wrong task performance. In addition, in the clipboard interface shows the progress of the activity based on the previously completed tasks.

In the case where tasks have to be performed in an ordered manner (Figure 2), we display a progress bar whose elements change colour as the tasks are performed, ranging from red to green as the bar gets filled like the colours of a battery when it's charging. Specifically, the progress bar is red when a few tasks have been completed, green when the tutorial is done, and transitions through intermediate colours based on the number of completed tasks.

This feature provides users with a visual representation of their progress, which can help motivate them to complete the tutorial.

In the case where the task can be executed in random order (Figure 3), we display a series of spheres, where each sphere represents a task. We display uncompleted tasks with a gray colour, which changes to green or red, respectively, on success or failure.

The interface provides buttons for navigating the task list. When there are random order tasks, the contour of the sphere that represents the task currently displayed in the clipboard is highlighted in yellow. In both cases, the percentage value of the overall progress in task completion is displayed.

The clipboard allows displaying optional information as media elements, such as videos and images that can aid in task performance. Such audiovisual material is displayed on the left side of the clipboard on a virtual screen that can be turned on or off by the user and, in the case of a video, allows the user to play and pause the content.

3.2.2. Tool Manager

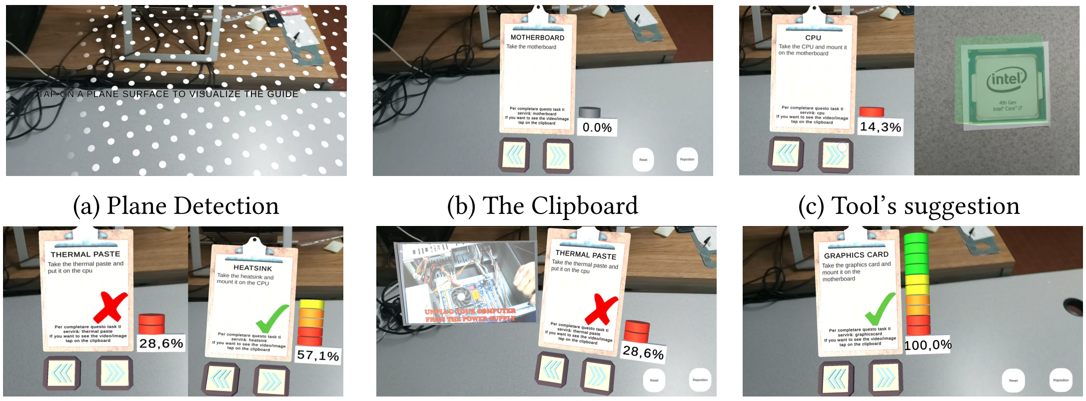
The Tool Manager handles the interface guidance on the tool used for the task accomplishment. Through image recognition techniques, we locate the correct tool for the task displayed in the clipboard. We highlight it in AR by placing a semitransparent green cuboid in the center of it. Our current implementation allows highlighting one tool per task. Such a choice was not made for not for technical reasons, but to force the tutorial designer to divide the tasks into simple tasks so that it is easier for the user to understand and execute them. The highlights for that specific task are turned off as soon as it is completed.

4. Use Cases

In this section, we explain how the application works through two different usage scenarios: assembling a PC (with ordered tasks) and decorating a Christmas Tree (without ordered tasks). These two use cases were chosen because the first use case requires accurate order in assembling the components, whereas in the second it is not strictly necessary.

4.1. Assembly a PC

Suppose novice users bought all the necessary components to assemble their PC at home. They discovered that there is an app that enables them to follow instructions to put the PC together. They open the app on his phone and start the PC assembly tutorial. When they start the tutorial, the app prompts them to rotate the phone screen for a better field of view (Figure 2a). Next, the app detects the planes in the scene and places a clipboard on the real-world surface (Figure 2b). The clipboard suggests the tools users will need to assemble the PC (Figure 2c). To tell if users are executing the task correctly, a green mark is displayed in case of success, and a red sign in case of failure (Figure 2d). The app guides users through each process step, suggesting the appropriate tools and providing visual aids to help them complete the task successfully (Figure 2e). Users complete the PC assembly without any difficulties (Figure 2f).



(a) Plane Detection (b) The Clipboard (c) Tool's suggestion

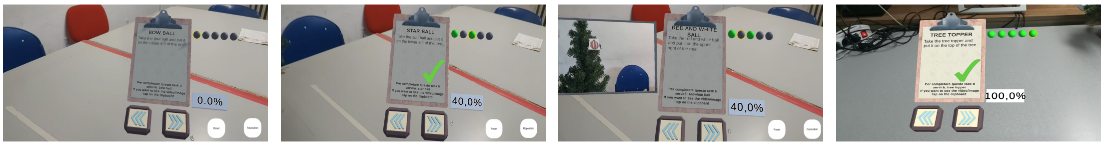
(d) Success/Failure of the task (e) Additional information (f) Tutorial Completed

Figure 2: Assembly a PC

4.2. Decoration of a Christmas Tree

Consider users who want to decorate their Christmas tree but are unsure of where to begin. They launch an app that offers a tutorial to help them get started. They can do the tasks in any sequence they choose because the tutorial, in this instance, is made up of unordered tasks.

When they start the tutorial, they see a clipboard on the right-hand side of the screen (Figure 3a). Each of the spheres on the clipboard is a different task that they must finish. The sphere representing the task they are currently working on is highlighted in yellow (Figure 3b). Users decide to start with the third task, which involves placing a ball on the tree even though they haven't completed the second task yet. As they work on each task, they can tap on the clipboard to see a small panel appear to the left of it. The panel contains an illustrative image to help them complete the task correctly (Figure 3c). Once users have completed all the tasks, the tutorial is finished (Figure 3d).



(a) The Clipboard (b) Unordered manner (c) Additional Information (d) Tutorial Completed

Figure 3: Decoration of a Christmas Tree

4.3. Building the examples with TutorialKit

We discuss here the information needed to construct these two examples with TutorialKit. First of all, the type of tutorial to be conducted must be specified i.e., the way in which the tasks are to be performed, ordered or unordered. In the first example (Assembly a PC) they are ordered, while in the second one (Decoration of a Christmas Tree) they are unordered. Next, we needed to collect all information about the tasks, such as the title, the text that explains the action which the user should do, and all information about the tools, like the name and associated image. The ease of creating these two tutorials lies in the fact that this information is contained within a JSON file that is loaded when the application is started.

5. Conclusion and Future Work

In this paper, we introduced an AR toolkit to help novice developers implement tutorials. The toolkit relies on a JSON description of the tasks involved and on callbacks for identifying the successful or wrong completion of the tasks. It manages the visualisation of a clipboard-based guidance interface and the highlighting of the required tools in the AR environment. We included two examples showing the required information and the resulting tutorials.

In future work, we will perform a thorough examination of our prototype. We plan to conduct user studies with novice developers to collect feedback to enhance our work. One of the most important extensions is an authoring environment for end-user developers. It will provide instruments to manipulate JSON files and handle the success and failure of tasks without writing code. Minor improvements include compatibility with further operating systems and devices, such as AR headsets, and the management of haptic feedback on task completion.

References

[1] M. Hanna, I. Ahmed, J. Nine, S. Prajapati, L. Pantanowitz, Augmented reality technology using microsoft hololens in anatomic pathology, Archives of Pathology & Laboratory Medicine 142 (2018). doi:10.5858/arpa.2017-0189-OA.

[2] H. Al Janabi, A. Aydın, S. Palaneer, N. Macchione, A. Al-Jabir, M. Khan, P. Dasgupta, K. Ahmed, Effectiveness of the hololens mixed reality headset in minimally invasive surgery: A simulation-based feasibility study, Surgical Endoscopy 34 (2020). doi:10.1007/s00464-019-06862-3.

[3] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, J.-K. Kämäräinen, Ar-based interaction for human-robot collaborative manufacturing, Robotics and Computer-Integrated Manufacturing 63 (2020) 101891. doi:https://doi.org/10.1016/j.rcim.2019.101891.

[4] W. Vorraber, J. Gasser, H. Webb, D. Neubacher, P. Url, Assessing augmented reality in production: remote-assisted maintenance with hololens, Procedia CIRP 88 (2020) 139–144. doi:https://doi.org/10.1016/j.procir.2020.05.025.

[5] H. Bahri, D. Krcmarik, R. Moezzi, J. Kočí, Efficient use of mixed reality for bim system using microsoft hololens, IFAC-PapersOnLine 52 (2019) 235–239. doi:10.1016/j.ifacol.2019.12.762.

[6] L. Zhang, S. Chen, H. Dong, A. El Saddik, Visualizing toronto city data with hololens: Using augmented reality for a city model, IEEE Consumer Electronics Magazine 7 (2018) 73–80. doi:10.1109/MCE.2018.2797658.

[7] K. Prasad, J. Winter, U. Bhat, R. V. Acharya, G. Prabhu, Image analysis approach for development of a decision support system for detection of malaria parasites in thin blood smear images, Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology 25 (2011) 542–9. doi:10.1007/s10278-011-9442-6.

[8] S. Feiner, B. Macintyre, D. Seligmann, Knowledge-based augmented reality, Commun. ACM 36 (1993) 53–62. doi:10.1145/159544.159587.

[9] M. Aebersold, T. Voepel-Lewis, L. Cherara, M. Weber, C. Khouri, R. Levine, A. Tait, Interactive anatomy-augmented virtual simulation training, Clinical Simulation in Nursing 15 (2018) 34–41. doi:10.1016/j.ecns.2017.09.008.

[10] M. Chmielewski, K. Sapiejewski, M. Sobolewski, Application of augmented reality, mobile devices, and sensors for a combat entity quantitative assessment supporting decisions and situational awareness development, Applied Sciences 9 (2019).

[11] X. Yang, L. Shu, J. Chen, M. A. Ferrag, J. Wu, E. Nurellari, K. Huang, A survey on smart agriculture: Development modes, technologies, and security and privacy challenges, IEEE/CAA Journal of Automatica Sinica 8 (2021) 273–302. doi:10.1109/JAS.2020.1003536.

[12] V. Krauß, Y. Uzun, Supporting medical auxiliary work: The central sterile services department as a challenging environment for augmented reality applications, in: 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2020, pp. 665–671. doi:10.1109/ISMAR50242.2020.00096.

[13] T. Yonghang, J. Shi, J. Pan, A. Hao, V. Chang, Augmented reality-based visual-haptic modeling for thoracoscopic surgery training systems, Virtual Reality & Intelligent Hardware 3 (2021) 274–286. doi:10.1016/j.vrih.2021.08.002.

[14] K. B. Borgen, T. D. Ropp, W. T. Weldon, Assessment of augmented reality technology's

impact on speed of learning and task performance in aeronautical engineering technology education, The International Journal of Aerospace Psychology 31 (2021) 219–229. doi:10.1080/24721840.2021.1881403.
[15] J.-R. Chardonnet, G. Fromentin, J. C. M. Outeiro, Augmented reality as an aid for the use of machine tools, Research and Science Today Supplement (2017) 25–31.
[16] C.-M. Chen, Y.-N. Tsai, Interactive augmented reality system for enhancing library instruction in elementary schools, Computers & Education 59 (2012) 638–652. doi:https://doi.org/10.1016/j.compedu.2012.03.001.
[17] D. Forest, Training the next generation of operators: Afpm immersive learning, Process Safety Progress 40 (2021). doi:10.1002/prs.12246.
[18] D. Eckhoff, C. Sandor, C. Lins, U. Eck, D. Kalkofen, A. Hein, Tutar: augmented reality tutorials for hands-only procedures, Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (2018).
[19] A. Villanueva, Z. Zhu, Z. Liu, K. Peppler, T. Redick, K. Ramani, Meta-ar-app: An authoring platform for collaborative augmented reality in stem classrooms, CHI '20, Association for Computing Machinery, New York, NY, USA, 2020. URL: https://doi.org/10.1145/3313831.3376146. doi:10.1145/3313831.3376146.