



Create by doing - Action sequencing in VR

Flavien Lécuyer, Valérie Gouranton, Adrien Reuzeau, Ronan Gaugne, Bruno Arnaldi

► To cite this version:

Flavien Lécuyer, Valérie Gouranton, Adrien Reuzeau, Ronan Gaugne, Bruno Arnaldi. Create by doing - Action sequencing in VR. CGI 2019 - Computer Graphics International, Advances in Computer Graphics, Jun 2019, Calgary, Canada. pp.329-335, 10.1007/978-3-030-22514-8_27 . hal-02119288

HAL Id: hal-02119288

<https://hal.science/hal-02119288>

Submitted on 3 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Create by doing – Action sequencing in VR^{*}

Flavien Lécuyer¹, Valérie Gouranton¹, Adrien Reuzeau¹, Ronan Gaugne², and
Bruno Arnaldi¹

¹ Univ Rennes, INSA Rennes, Inria, CNRS, IRISA, Rennes, France

² Univ Rennes, Inria, CNRS, IRISA, Rennes, France

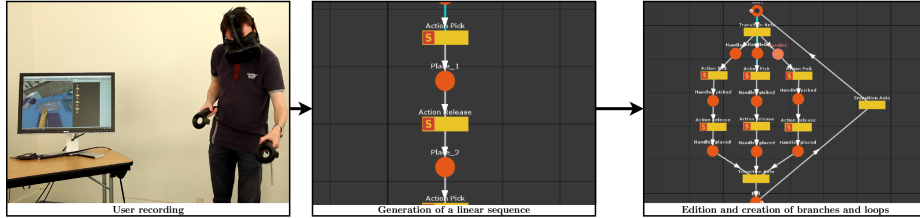


Fig. 1. The proposed workflow starts from the action recording to generate a straight scenario, which can then be edited for a more complex sequence

Abstract. In every virtual reality application, there are actions to perform, often in a logical order. This logical ordering can be a predefined sequence of actions, enriched with the representation of different possibilities, which we refer to as a scenario. Authoring such a scenario for virtual reality is still a difficult task, as it needs both the expertise from the domain expert and the developer. We propose to let the domain expert create in virtual reality the scenario by herself without coding, through the paradigm of *creating by doing*. The domain expert can run an application, record the sequence of actions as a scenario, and then reuse this scenario for other purposes, such as an automatic replay of the scenario by a virtual actor to check the obtained scenario, the injection of this scenario as a constraint or a guide for a trainee, or the monitoring of the scenario unfolding during a procedure.

1 Introduction

Virtual reality has a strong positive impact for the learning of procedures [1], as with serious games [8]. Virtual reality also tends to make students more interested during the learning process, because they can interact with the knowledge presented to them, they get a better understanding of it and feel more involved [3]. Those serious games can benefit from the use of a scenario, which divides the learning into different steps, for a better understanding of each concept that is presented. When we know in advance what are the actions allowed by the

^{*} This work is part of the ANR-16-FRQC-0004 INTROSPECT project, and the SUN-SET project funded by the ANR-10-LABX-07-01 “Investing for the Future” program

scenario, and what is supposed to be the timeline for those actions, the scenario can take the form of an action sequence, with possible choices or repetitions changing the final outcome of the scenario. However, such scenarios are difficult to implement for trainers that are not experts in the domain of computer science, since there is a need for a certain knowledge of coding to author a scenario for virtual reality. At the very least, someone wanting to create a scenario needs to learn how the scenario representation works to be able to use it. This requires time and effort that could be avoided thanks to a higher level authoring tool. Because of this, it is very often a developer who integrates the scenario in the virtual environment. This often leads to a wrong interpretation of the scenario, making it less efficient, and even in the worse case counter productive. On the contrary, explaining a procedure in real life can be easy. A common way of explaining to someone what should be the sequence of actions in a procedure is to show the procedure to the person learning it. This is particularly efficient, since it shows how each action leads to the following one, and at the same time, helps in the achievement of the final goal. It is also a good way to demonstrate the feasibility of the procedure.

We propose to use this metaphor of showing the procedure to explain it, to create scenarios for a virtual environment from this virtual environment. In this paper, after the analysis of the related work in section 2, we present a concept for the generation of a scenario through the recording of users in a virtual environment, through three points:

- The *create by doing* concept to author scenarios, presented in Section 3
- The presentation of a proof of concept based on this paradigm, in Section 4
- The creation of a use case demonstrating our approach in Section 5

2 Related work

Some works proposed to provide a replay function in virtual environments, which can be seen as a way to record the user actions to generate a scenario, or at least a trace of what the user has been doing. For instance, Chan et al. captured the movements of the user to replay them in virtual reality, which allowed to highlight more efficiently the mistakes during the learning of dance moves [2]. However, movements are recorded as a continuous data, without any interpretation or cutting into separate steps. Because of this, the replay is only a video separated from the virtual environment, which cannot be used later if the environment is meant to be modified even the slightest. The video format can only be used to provide indications to the user of an application, making it impractical if the scenario is supposed to constrain and guide the actions of the user. Similarly, Bailenson et al. [1] used the possibility of virtually going back in time, to help the users in the learning of martial arts. For an efficient analysis of the movements of the user, the participants were asked to wear specific clothes in order to help the computer vision algorithm. This is a strong constraint, since it forces the user to have access to a specific equipment to use the application. Another interesting work is the one done by the MIT Education Arcade with

Mystery at the Museum (M@M) [4] and *Environmental Detectives* [5], in which replaying the previous actions is possible, and provided as a means to remind the players of what already happened. Unfortunately, their approach is not really detailed, and the replay function seems to rely more on a log of the events than on a reusable scenario. A scenario that is generated through the execution can also be reused to generate a final product, as done by Paiva et al. [7]. In this application, the child can influence the evolution of the story. Those choices are recorded in order to produce a movie taking into account every choice made by the user. This means that, if a user makes the correct decisions and unfolds the best scenario possible, this unfolding will be reusable for other people only in a passive way. A better possibility would be to create a mixed reality application through an immersive environment, as proposed in [6]. However, the creation of a scenario for a virtual reality application is a tedious task for a non-developer.

3 Scenario creation

The authoring of scenarios for virtual reality is a tedious task, because the necessary knowledge is shared between the domain expert and the developer.

Before we present our approach, it is important to define what we call a "scenario". In our context, a scenario is a sequence of actions, linked in a given order so as to describe a logical progression through time. Of course, this sequence can be enriched with branches, for which the choices of the user will influence how the rest of the scenario unfolds. It should also be possible to repeat the same action or group of actions multiple times, which would be translated by a loop in the sequence of actions.

To make the *create by doing* process possible, the actions must be as atomic as possible, but also discrete in time. Indeed, we focus more on the consequences of an action, rather than on how this action is done. An illustration of this is given in Figure 2. In this precise example, we first take the object, triggering the recording of the **Take** action. Then, we move it as we want, until it goes through a given point in space, which triggers the **Move to** action regarding this point. Of course there could also be other points in space triggering this action. Finally, we release the object on a given spot, with the **Place** action.

We propose to use the *create by doing* paradigm, which is based on the automatic recording of the actions done by a user. Thanks to this, the authoring of a scenario becomes as natural as possible, since it is the very execution of the procedure that generates the corresponding action sequence.

The main steps of our approach are summarized in Figure 1. There are four main steps for the authoring of a scenario with our method:

1. Define a set of possible actions in the virtual environment
2. Run the application in a free mode, with the tool recording the performed actions. At the end of this step, the expert gets a linear sequence corresponding to the actions she performed.
3. Edit the action sequence to make it more complex, by adding branches/loops.

Our paradigm of creating the action sequence by doing actions mostly improves the recording of straight procedures. Indeed, this part can be done by letting the expert choose some actions and execute them freely, to generate the according scenario with the sequence of selected actions. The best way to let the expert select the actions to record in the scenario is to actually let her *do* those actions from within the virtual environment, which is why the recording is done from the application itself. In order to make the actions recorded in the generated scenario easier to recognize, we limit the actions to the ones that are already defined in the environment. This way, the actions can themselves notify the authoring tool, which can in turn generate the action sequence. For the creation of more complex scenarios, with branches and tools, we let the expert modify a straight scenario from a graphic editor. The use of a graphic editor for the final modification of the action sequence is also an efficient way to let the averted user add finer modifications to the scenario. Indeed, it is also possible to add other steps, independent from the actions of a user, and to add direct consequences for the execution of an action, directly from the editor.

4 Proof of concept

In order to verify the practical feasibility and the usefulness of our approach, we built a proof of concept, under the form of a scenario authoring tool.

The authoring tool can be integrated in a virtual environment that is already created. This is done through an automatic generation of the scenario according to the actions that are done by the application designer. For an action to be recognized, it has to be defined beforehand in the environment. In fact, an action is considered as a given behaviour the application executes when asked to. If the actions are continuous over a given period of time, we will prefer to log the beginning and the end of the action. For instance, we will not take into account the fact that a user holds an object in the hand, but rather the moment when the object is taken, and the moment when the same object is released.

Each time an action is recognized, it is logged in the form of a new transition and state, with the transition being triggered by the said action. The scenario being created changes dynamically according to the events in the virtual environment. Because of this, it should not be saved or reused before the end of the recording. However, it is still possible to modify it from the editor at any moment.

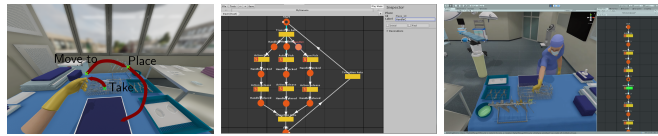


Fig. 2. On the left, the actions recorder for a pick and place task. On the middle, a scenario in the editing tool. On the right, a scenario performed by a virtual agent.

In our proof of concept, we provide three main features, according to the main steps we presented in Figure 1. The first one, which is the core of our *create by doing* paradigm, is the free execution of actions from within the virtual environment to generate a scenario. The user will have two options for the interactions. First, it is of course possible to interact in the ways that have been explicitly defined with the objects in the environment. Instead, the actions define what they can do on some properties. To be involved in specific actions, the objects have to possess those properties. This way, an action can be defined only once, independently of the objects involved. For instance, plugging the outlet of a fridge on the wall is the same action as plugging the outlet for the oven.

When the creation of the linear scenario is done, the expert can move on to the creation of branches on loop on the scenario. To do this, we use the same interface as for the display of the scenario during the authoring, which is the one illustrated in Figure 2. On this interface, the expert can modify the structure of the scenario simply by moving nodes, and by creating/deleting edges.

In order to facilitate the creation of branches, we also propose to use an undo function that can be used from the virtual environment. In the scenario, the undo will change the last state selected in the sequence, to go to the previous step. The actions already recorded will be kept in the scenario as an additional branch. Whenever another action is recorded, the scenario will then create a new branch, starting from the selected state followed by said action.

To help the domain expert to check the scenario, we propose to make a virtual agent automatically play it, action by action, in the virtual environment, as shown in Figure 2. When there are choices to make in the scenario, the virtual agent can be customized to modify how it chooses the actions to perform.

5 Medical procedure training

In the medical domain, the knowledge of the procedures is extremely important, since every member of an intervention must know his/her next action, in order not to slow down the entire group. To help the students in learning the procedures inherent to the preparation of a surgical intervention, we created an application, dedicated to scrub nurses, in which the trainee has to prepare the operating table before the intervention.

In this use case, the trainee has to take some instruments and to place them on the table in a very specific manner according to the nature of the intervention.

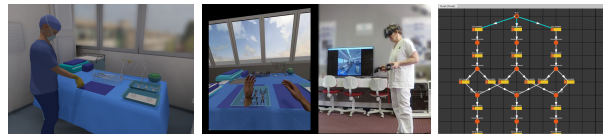


Fig. 3. On the left, the preparation table in the virtual operating room. On the middle, a scrub nurse interacting with the application. On the right, the corresponding scenario

For different interventions, different tools and dispositions might be appropriate, which is why different scenarios can be needed. Even though the virtual environment is a powerful tool for learning, it is by nature supposed to be used for one person at a time. To show more efficiently the correct procedure to a group of students, showing the virtual agent evolving in the virtual environment would be more adequate, as it would be easy to transpose the knowledge gained from seeing the movie, and it is also easy to show a movie to a group. The application simulates an operating room, with the table to prepare, on a HMD. During the learning process, the trainee is helped by the scenario, which highlights the possible actions in the environment to make her move forward in the simulation.

To design the procedure in the virtual environment, the trainer can use the scenario authoring tool to record the correct sequence of actions, to create for instance the scenario provided in Figure 3. This creates a scenario in which the trainee starts from the table with no instrument, and finishes with the table ready for the operation. In fact, the trainer can use the application as a tool in which the students can *learn by doing*, but the trainer can also benefit from the scenario being played at the same time as the trainee executes the tasks to *monitor* her. An overview of the setup, with a real scrub nurse unfolding the scenario, can also be seen in Figure 3.

6 Discussion

Through the creation of our use cases, we noted that our scenario authoring tool could also be easily used for other purposes, illustrated in Figure 4.

Learn by doing When the scenario is done, it can be reused for an application in which a trainee has to complete the tasks described by the scenario.

Learning by seeing The scenario can be automatically replayed by a virtual agent to demonstrate the procedure.

Collaborate by doing Multiple users can collaborate to perform the scenario, possibly with an assignation of roles.

Monitor by seeing The scenario provides two feedbacks: the execution in real-time of the scenario can be displayed on-screen, and the replay can be used afterwards as a tool to repeat a previous execution.

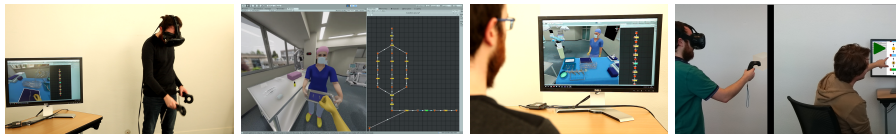


Fig. 4. The additional uses of the scenario, from top left to bottom right: learning by doing, collaborating by doing, learning by seeing, monitoring by seeing

7 Conclusion

In this paper, we presented a tool to easily create scenarios for virtual reality without having to write a single line of code, through the principle of "*creating by doing*". This way, the application designer can start from a simple virtual environment enriched with objects and interactions, and create a scenario through the execution of a simulation in which every action adds a step in a scenario. More functions are also provided in our tool to help in building more complex scenarios, to add choices to the simulation.

The scenario created through this tool can then be used in the virtual environment. Thanks to this, a trainee can then be immersed in an environment with given tasks to complete. Through the completion of these tasks, the student can *learn by doing*, or even *collaborate by doing* when there are multiple students. Furthermore, the scenario can be used to make a virtual agent execute the scenario automatically, which is a great help to demonstrate how the sequence of actions is supposed to be. This demonstration can be done to help the trainees, who can then *learn by seeing* someone else do the work.

References

1. Bailenson, J.N., Yee, N., Blascovich, J., Beall, A.C., Lundblad, N., Jin, M.: The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *Journal of the Learning Sciences* **17**(1), 102–141 (2008), <https://doi.org/10.1080/10508400701793141>
2. Chan, J.C.P., Leung, H., Tang, J.K.T., Komura, T.: A virtual reality dance training system using motion capture technology. *IEEE Transactions on Learning Technologies* **4**(2), 187–195 (April 2011), <https://doi.org/10.1109/TLT.2010.27>
3. Fletcher, J.D.: Does this stuff work? a review of technology used to teach. *Tech-Knowlogia* **Jan-Mar**, 10–14 (2003)
4. Klopfer, E., Perry, J., Squire, K., Jan, M.F., Steinkuehler, C.: Mystery at the museum: a collaborative game for museum education. In: *Proceedings of the 2005 Conference on Computer support for collaborative learning*. pp. 316–320. International Society of the Learning Sciences (2005), <http://dl.acm.org/citation.cfm?id=1149293.1149334>
5. Klopfer, E., Squire, K.: Environmental detectives—the development of an augmented reality platform for environmental simulations. *Educational Technology Research and Development* **56**, 203–228 (04 2007)
6. Lee, G.A., Nelles, C., Billinghamurst, M., Kim, G.J.: Immersive authoring of tangible augmented reality applications. In: *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*. pp. 172–181. ISMAR '04, IEEE Computer Society, Washington, DC, USA (2004), <http://dx.doi.org/10.1109/ISMAR.2004.34>
7. Paiva, A., Machado, I., Prada, R.: Heroes, villains, magicians, & dramatis personae in a virtual story creation environment. In: *Proceedings of the 6th International Conference on Intelligent User Interfaces*. pp. 129–136. IUI '01, ACM, New York, NY, USA (2001), <http://doi.acm.org/10.1145/359784.360314>
8. Zyda, M.: From visual simulation to virtual reality to games. *Computer* **38**(9), 25–32 (Sept 2005), <http://dx.doi.org/10.1117/12.703906>