A demonstration of the ASAP Realizer-Unity3D Bridge for Virtual and Mixed Reality Applications

Jan Kolkmeier ✉, Merijn Bruijnes, and Dennis Reidsma

Human Media Interaction, University of Twente, The Netherlands
[j.kolkmeier; m.bruijnes; d.reidsma]@utwente.nl

Abstract. Modern game engines such as Unity make prototyping and developing experiences in virtual and mixed reality environments increasingly accessible and efficient, and their value has long been recognized by the scientific community as well. However, these game engines do not easily allow control of virtual embodied characters, situated in such environments, with the same expressiveness, flexibility, and generalizability, as offered by modern BML realizers that generate synchronized multimodal behavior from Behavior Markup Language (BML). We demonstrate our integration of the ASAP BML Realizer and the Unity3D game engine at the hand of an Augmented Reality setup. We further show an in-unity editor for BML animations in the same system.

Keywords: BML, SAIBA, Virtual Agents, Unity, Virtual Reality, Mixed Reality

1 Introduction

The Articulated Social Agent Platform (ASAP) provides a collection of software modules for building social robots and virtual humans [3, 5]. ASAP is a SAIBA compliant, OS independent behavior realizer, enhanced with two features that allow for fast and fluent virtual human behaviors: a close bi-directional coordination between input processing and output generation, and incremental processing of both input and output. The default virtual embodiment in ASAP is for the Armandia ECA. It is implemented as a Java OpenGL application that is controlled by directly binding to the ASAP platform code.

In our demo, we show the integration of the ASAP BML Realizer and the Unity3D game engine[1]. For more details and background, we refer to the main paper [1]. This demo is meant to showcase the technology itself, invite others to use it and help solving similar problems for their own engines or realizers. The main software component used in this demo will be made publicly available[2]. In the following we describe what we demonstrate as well as previous work that made use of this technology, illustrating different use cases in which it can be employed.

[1] Developed by Unity Technologies: https://unity3d.com
[2] https://github.com/hmi-utwente/AsapUnityBridge

J. Beskow et al. (Eds.): IVA 2017, LNAI 10498, pp. 223-226, 2017.
DOI 10.1007/978-3-319-67401-8_26

2 Virtual Agents in the HoloLens

We demonstrate the ASAP Realizer-Unity3D Bridge at the hand of a prototype shown in Fig. 1. We use the Microsoft HoloLens™[3] running a ‘client’ Unity scene to the ASAP backend where the agent behavior is being realized. Virtual Agents can be placed in the real world using a uniquely identifying augmented reality marker using the Vuforia augmented reality SDK[4].

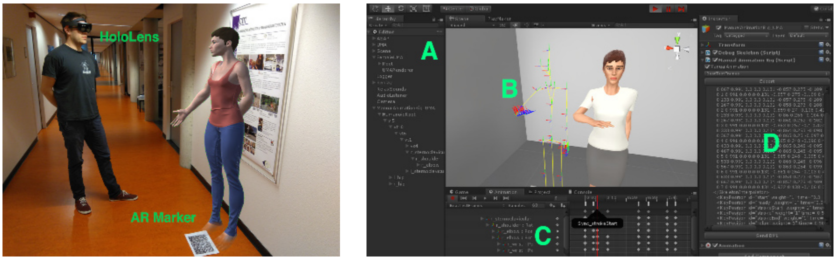

Fig. 1. Left: Using the ASAP Realizer-Unity3D Bridge and HoloLens AR glasses to create a situated embodied agent in our hallway. Right: The Animation Editor view in Unity. A: The skeleton hierarchy. B: The character and the ASAP Animation rig. C: The time-line editor with sync-points as animation events. D: The exported BMLT.

3 Animation Editor for BML

We further demonstrate our unity extension to create BML-animations. Natural behaviors for a virtual agent depends in large part on realistic and appropriate animations. Currently, the ASAP is lacking a flexible pipeline for creating new animations for gestures and postures. For this purpose we extended the Unity editor with a work flow for creating and exporting keyframe animations using the animation time-line (see Fig. 1). These animations can be exported in the *<bmlt:keyframe>* format as described in ASAP’s BML Twente-extensions[5] (BMLT).

Animations and poses are created by animating the H-Anim[6] skeleton – currently using forward kinematics only. The H-Anim skeleton is the same as the one that is used for re-targeting between ASAP and Unity (see above), hence the resulting poses are consistent between the ‘preview’ in the editor and what is produced by ASAP when the generated BML is realized. New BMLs can be used by ASAP to control also other characters in any H-Anim-compatible embodiment.

[3] https://www.microsoft.com/en-us/hololens
[4] https://www.vuforia.com
[5] asap-project.ewi.utwente.nl/wiki/BMLT
[6] http://h-anim.org/Specifications/H-Anim1.1

The BML standard supports the synchronisation between different multimodal behaviors. During realisation, the agent's behaviors can be broken down into phases making it easy to specify alignment of behaviors at meaningful boundaries: at sync-points. To allow this synchronisation with the new animations, named sync-points for gesture animations can be added in the Unity time-line (see Fig. 1 C).

4 Other Use-cases & Projects

The technology presented in this demo is a SAIBA compliant BML realizer in Unity. Besides the demo described above, we have applied the Unity embodiment in several recent projects, three of which we discuss briefly here, highlighting some aspects of the system.

The AVATAR project. In the AVATAR project, a partner developed a Unity environment where a policeman and a witness could interact with the user. The aim was to create an educational setting where social skills could be trained using virtual agents. The behaviors of the agent was driven by ASAP and realized in the Unity world. The facial expressions and lip-sync were procedurally generated by ASAP and combined with animations that were specified in Unity by the partner. The animations were mapped to descriptive BML commands, demonstrating the flexibility of our approach.

Fig. 2. The R3D3 robot (left) with a mixed-modality agent and the Moral Competence Training experiment setup in immersive VR (right)

The R3D3 project. In the Rolling Receptionist Robot with Double Dutch Dialogue (R3D3) project, a robot drives around in a reception area and a virtual agent on a screen in the robot's hands could help people find their way in the building. In this project, ASAP controlled the robot body as well as the virtual agent on the screen (see Fig.

2 left and [4]). In early versions of the system, the agent was situated in an Android tablet using the Unity embodiment. Calculations for the robot's perception as well as the ASAP realizer were executed on a more powerful PC. The tablet only rendered the agent based on the received output of the Unity embodiment over the middleware. This shows the capability of ASAP to run distributed and to control a variety of mixed-modality embodiments.

Training for Moral Competences. This project investigates the possibilities of developing a training agent that help teachers to develop their moral and ethical competences through simulating conflicts. A prototype of this training agent was developed using the Unity embodiment (see Fig. 2 right and [2]). It was necessary to use Unity as we employed a VR headset with integrated eye-tracker, the FOVE 0[7], that was not supported in other 3D engines for ASAP. In this prototype, we made use of the world environment by keeping ASAP informed of the user's head position (which was available in the Unity world frame from the VR headset). This way ASAP could reliably generate mutual and averted gaze behaviors through its IK system.

5 Conclusion

The integration we present here makes it possible to drive characters in Unity with the ASAP BML realizer. It is also possible to control virtual agents on a system with limited computing power such as a tablet through the middleware connection. We demonstrate this specifically with our augmented reality setup. An agent is situated in the real world, rendered through the HoloLens in a Unity scene, while the behavior is computed using the ASAP back-end in a remote location. We further showcase the Unity animation editor as way of quick and flexible creation of reusable BML gesture animations and postures that support procedural change for multimodal synchronization.

References

1. Kolkmeier, J., Bruijnes, M., Reidsma, D., Heylen, D.: An asap realizer-unity3d bridge for virtual and mixed reality applications. In: International Conference on Intelligent Virtual Agents. Springer (2017)
2. Kolkmeier, J., Lee, M., Heylen, D.: Moral conflicts in vr: Addressing grade disputes with a virtual trainer. In: International Conference on Intelligent Virtual Agents. Springer (2017)
3. Kopp, S., van Welbergen, H., Yaghoubzadeh, R., Buschmeier, H.: An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing. Journal on Multimodal User Interfaces 8(1), 97–108 (2014)
4. Linssen, J., Berkhoff, M., Bode, M., Rens, E., Theune, M., Wiltenburg, D.: You can leave your head on - attention management and turn-taking in multi-party interaction with a virtual human/robot duo. In: International Conference on Intelligent Virtual Agents. Springer (2017)
5. Van Welbergen, H., Yaghoubzadeh, R., Kopp, S.: Asaprealizer 2.0: the next steps in fluent behavior realization for ecas. In: International Conference on Intelligent Virtual Agents. pp. 449–462. Springer (2014)

[7] getfove.com