2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) | 979-8-3503-4839-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/VRW58643.2023.00102

Inceptor: An Open Source Tool for Automated Creation of 3D Social Scenarios

Dan Pollak*
Efi Arazi School of Computer Science, RUNI

Jonathan Giron†
Sammy Ofer School of Communications, RUNI

Doron Friedman‡
Sammy Ofer School of Communications, RUNI

Abstract

Inceptor is a tool designed for non-expert users to develop social VR scenarios that includes virtual humans. The tool uses a text based interface and natural language processing models as input, and generates complete 3D/VR Unity scenarios as output. The tool is currently based on the Rocketbox asset library. We release the tool as an open source project in order to empower the extended reality research community.

1 Introduction

The process of 3D creation involves significant barriers. As much as creating movies, it requires several fields of expertise such as 3D modeling, scene direction, programming, interaction design, and other roles that are project dependent. Although technology improves and removes these barriers, using 3D real-time engines such as Unity and Unreal Engine, the learning curve and complexity of creating complete animated scenes with these engines may be off-putting for novices.

This is even more evident in creating scenarios involving virtual humans. Software-controlled human avatars are highly desirable for academic research, simulations, and storytelling. Tools for human models and animation creation have progressed significantly, but the composition and direction of human avatars in a virtual environment remain a challenging task. While the idea of automatic director is not new [4], the technological tools supporting it are becoming more available and sparse.

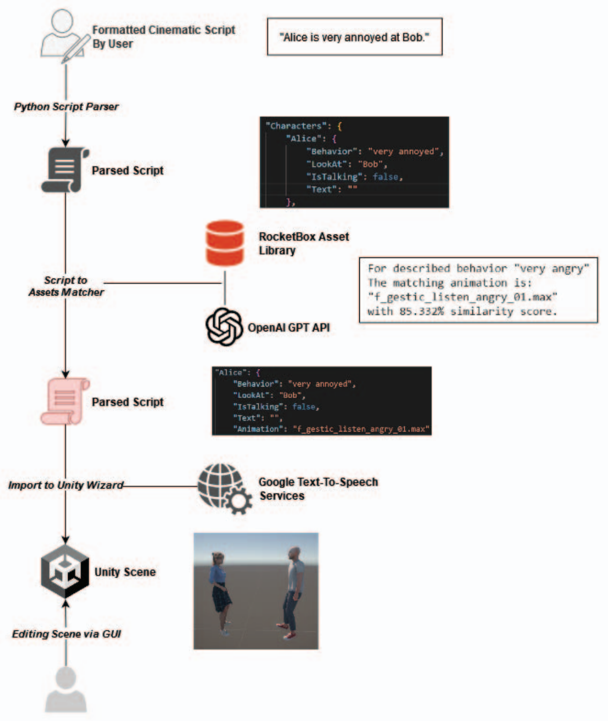
Therefore, we introduce *Inceptor*, a tool for scripting and directing 3D scenarios with human avatars. Inceptor is an add-on for the Unity engine that adds a suite of graphical user interface (GUI) elements that enable the creation and directing of multiple coordinated human avatars, either by GUI or by English-based scripting language.

1.1 State of the Art

Currently, there is no one-stop software for creating fast 3D scenarios. There has been a lot of research on animation and control of virtual humans [7], include several platforms [10].

Unity or Unreal Engine are currently the most popular real-time 3D engines that allow creating 3D scenarios, but they require expertise with the software and at least basic programming knowledge. While there are tools that simplify the process, such as Unity's Timeline and Animator tools, the expertise barrier is still significant.

Tools developed for 3D storytelling such as SceneMaker [1], Aesop [11], El-Mashad et al. system [3] and Toontastic 3D allow directing 3D scenarios using text interfaces and simplified GUI. Most of the tools use graphic engines that are not open to extensions. Furthermore, the output of these systems is usually a 2D video of the 3D environment from a directed camera point of view in the environment.



```
"Characters": {
    "Alice": {
        "Behavior": "very annoyed",
        "LookAt": "Bob",
        "IsTalking": false,
        "Text": ""
    },
```

```
For described behavior "very angry"
The matching animation is:
"f_gestic_listen_angry_01.max"
with 85.332% similarity score.
```

```
"Alice": {
    "Behavior": "very annoyed",
    "LookAt": "Bob",
    "IsTalking": false,
    "Text": "",
    "Animation": "f_gestic_listen_angry_01.max"
```

Figure 1: A schematic diagram of *Inceptor* process.

Our proposed system employs Unity as its engine, a popular and modern real-time 3D engine that is free to use for non-commercial purposes. The system has been designed to be extensible, enabling the community to incorporate additional content and features into the platform. In addition, we utilize Unity's built-in tools for integrating with various virtual reality (VR) headsets.

2 Conceptual Framework

To facilitate the creation of complex social scenarios, we have implemented some constraints on the scripting and directing process. We have divided the script into "Clips" in which all human avatar (”Characters”) initiate each "Behavior" simultaneously with the other characters. Each behavior can consist of a specific animations and audio clips. This allows us to divide the scene into a storyboard and specify the behavior of each character within a specific time frame.

The scenario creation process starts with writing a *Cinematic Script*, which is then processed by pairing the behavior described in the script for each character into animations, and an avatar according to the character's description. We pair the avatars and behaviors that were described in the cinematic scripts to assets from our asset

[*] e-mail: dan.pollak@post.runi.ac.il
[†] e-mail: jonathan.giron@runi.ac.il
[‡] e-mail: doronf@runi.ac.il

979-8-3503-4839-2/23/$31.00 ©2023 IEEE
DOI 10.1109/VRW58643.2023.00102

library, according to the most similar asset. We determine which asset is the most similar using the assets description and GPT-3 sentence embedding. The text is converted into audio files using Text-To-Speech services. After loading the cinematic script in Unity, we choose an environment fitting our scenario. We provide several basic environments, and for different scenarios you may choose to define a custom environment. Lastly, the infrastructure allows for specific behavior editing using the GUI tools in Unity, which allows to preview easily each clip and edit it manually. (See fig. 1).

As the system is open-source, we encourage the integration of third-party software add-ons, such as avatar and animation creators or lip-sync and inverse kinematic plugins, to enhance the range and accuracy of animations. We also provide a means to override the default pipeline and break the "clip" patterns, allowing characters to perform behaviors asynchronously. In addition, the system has been designed to communicate with and receive inputs from other sources, enabling it to connect to AI-based platforms that can guide character behavior. This makes the platform suitable for experimentation and testing of virtual human features, such as connecting a machine learning model that generates facial animations based on character dialogue, or a model that selects which clip to play in response to user input.

2.1 Cinematic Script and Assets

The backbone of each scenario is the cinematic script. This script is written in a format similar to film scripts, describing what each character is saying or doing in every clip of the scenario. Although the formatting of the script is strict, but there are no limitations on the language vocabulary.

The written script is then parsed then transformed into a format that can be utilized by the Unity framework. We use GPT-3 [2] sentence embedding to choose the closest animation description from each behavior description, and closest avatar from each character description. As 3D asset library to match from, we chose Rocketbox [6] and Movebox [5] by Microsoft for its extensive collection of both avatars and rigged animations. We manually wrote descriptions for most avatars and animations in the library in order to enable the matching process.

Lastly, the spoken text for each character is being parsed for each line he speaks. The character audio clip is then created by an external Text-To-Speech provider. We chose to use Google's TTS service, as it allows several voices and allows up to 1 million characters per month for free. Using the character description, we generate a different voice for each character. The audio generation is a process that happens only once per cinematic script, and then saved as audio files in the Unity project. As for lip-sync, we use Rocketbox's implementation of the Oculus Lipsync tool [15].

2.2 Assembly and Fine Tuning

Inceptor is imported to Unity using a Unity Package. After importing the package, the user can start a wizard that will guide him through the scene initialization, giving him a chance to review the choices of the language model. After modifying the parameters, the scene will be built according to the cinematic script. From there, the in-Unity scripts will create the Animators and Prefabs according to the cinematic script parameters, and save it as an independent scene. While the framework takes care of character behavior, it does not create the environment itself and the scene composition. There are two types of scenarios: linear – essentially a movie, and operator controlled. The latter option is useful for VR experiments, or for any scenario in which there is a human operator; each clip is automatically turned into a GUI button, and these can be selected by the operator in real time (e.g., as in Nakash at el. [12] experiment) After the scene is done, the user can use a preview tool to fine tune and adjust each clip to its liking. For Unity "power users", the scene is completely open to manipulation using Unity's tools and they will be able to edit it freely.

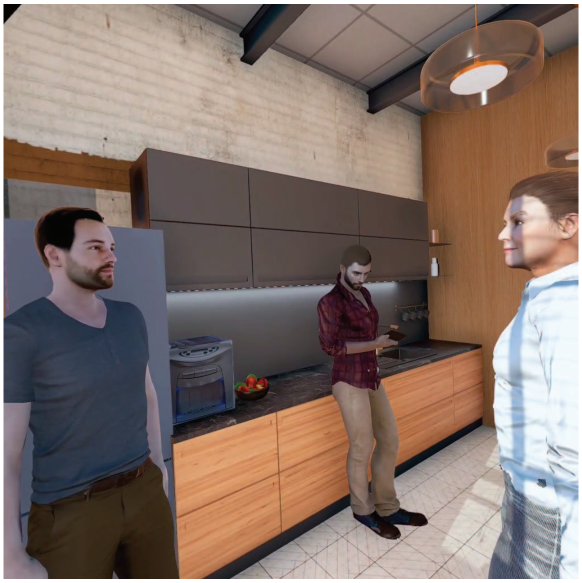


Figure 2: *A* snapshot from a social scenario with multiple characters, generated with the help of Inceptor.

3 Results

Inceptor is a tool designed for non-expert Unity users to easily create social VR scenarios featuring human avatars. It was initially developed for researchers, particularly psychology labs, to enable the creation of experiments in a shorter time frame and at a lower cost. However, we believe that Inceptor has the potential to be used by a variety of audiences, including technical users who may find it useful as a shortcut for creating social scenarios. In the hands of creators, Inceptor allows for the creation of 3D social scenarios, particularly in VR, reducing barriers to creating VR applications for small-scale projects such as short films and simulations. As such, it has the potential to facilitate the advancement of the XR community.

Within the scope of the Advanced Reality Lab, former but similar versions of Inceptor were used to create several simulations leading to research papers such as Nakash et al., and tool is used by a large number of undergraduate and MA students in psychology and human-computer interaction.

4 Conclusion and Future Work

In this work, we provide a proof of concept tool, which was developed from an internal need of the growing number of VR projects in campus. Our system provides an easy method for translating a textual narrative to a virtual social scenario, including avatars, animation and speech output. The system is currently limited: The system can only activate all characters at the same time, relies on assets tagging, and also by the resources library offered by Rocketbox. Another significant limitation is the linearity of the generated scenarios. While it is possible to create a operator controlled interactive simulation, but it is not interactive as a stand-alone experience.

We decided to open source this tool in order to encourage other developers and creators to expand the usage of the tool and its integration to other 3rd party software components and assets. We are currently working to expand the base features of Inceptor, adding movement animation, prop animations, and facial animations. In order to make Inceptor a more reliable tool, we are conducting a user study in order to quantify the degree to which the results of the cinematic scripts adhere to the user's expectations.

In the last few years there have been an overwhelming breakthrough in automatic generation of visual content based on generative deep neural networks; this is not limited to images such as DALL-E 2 [13], but also animation and movies [8], including automatic generation of 3D animation, e.g., AvatarCLIP [9] and MDM Based Generation [14]. Our assumption is that such tools will keep developing, but that a complete text-to-social-VR pipeline based on such techniques is still outside reach. Our approach is complimentary; our tool is aimed at serving as a 3D/VR wrapper for such technologies. Moreover, unlike future end-to-end methods, our approach generates a Unity project that is easily understood, modified, and maintained by humans.

Inceptor is maintained in GitHub and intended for free academic use.

This work was partially supported by projects GuestXR (101017884) and Socrates EU project (951930), which have received funding from the European Union's Horizon 2020 research and innovation program. I would also like to thank Maya Shekel and Alon Shoa for their support and ideas that made this framework come true.

References

[1] M. Akser, B. Bridges, G. Campo, A. Cheddad, K. Curran, L. Fitzpatrick, L. Hamilton, J. Harding, T. Leath, T. Lunney, et al. Scenemaker: Creative technology for digital storytelling. In *International Conference on ArtsIT, Interactivity & Game Creation, International Conference on Design, Learning, and Innovation*, pp. 29–38. Springer, 2017.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] S. Y. El-Mashad and E.-H. S. Hamed. Automatic creation of a 3d cartoon from natural language story. *Ain Shams Engineering Journal*, 13(3):101641, 2022.

[4] D. Friedman. Automating spielberg. In *Proceedings of the EVA 2003 JML Symposium, London*, pp. 1–10, 2003.

[5] M. Gonzalez-Franco, Z. Egan, M. Peachey, A. Antley, T. Randhavane, P. Panda, Y. Zhang, C. Y. Wang, D. F. Reilly, T. C. Peck, et al. Movebox: Democratizing mocap for the microsoft rocketbox avatar library. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 91–98. IEEE, 2020.

[6] M. Gonzalez-Franco, E. Ofek, Y. Pan, A. Antley, A. Steed, B. Spanlang, A. Maselli, D. Banakou, N. Pelechano, S. Orts-Escolano, et al. The rocketbox library and the utility of freely available rigged avatars. *Frontiers in virtual reality*, p. 20, 2020.

[7] J. Gratch, J. Rickel, E. Andre, J. Cassell, E. Petajan, and N. Badler. Creating interactive virtual humans: some assembly required. *IEEE Intelligent Systems*, 17(4):54–63, 2002. doi: 10.1109/MIS.2002.1024753

[8] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[9] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022.

[10] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol. Building interactive virtual humans for training environments. In *Proceedings of i/itsec*, vol. 174, pp. 911–916, 2007.

[11] T. J. Meo, C. Kim, A. Raghavan, A. Tozzo, D. A. Salter, A. Tamrakar, and M. R. Amer. Aesop: A visual storytelling platform for conversational ai and common sense grounding. *AI Communications*, 32(1):59–76, 2019.

[12] T. Nakash, T. Haller, M. Shekel, D. Pollak, M. Lewenchuse, A. B. Klomek, and D. Friedman. Increasing resilience and preventing suicide: training and interventions with a distressed virtual human in virtual reality. In *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents*, pp. 1–8, 2022.

[13] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[14] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[15] M. Volonte, E. Ofek, K. Jakubzak, S. Bruner, and M. Gonzalez-Franco. Headbox: A facial blendshape animation toolkit for the microsoft rocketbox library. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 39–42. IEEE, 2022.