Human Systems Engineering and Design (IHSED2025), Vol. 198, 2025, 277–288
https://doi.org/10.54941/ahfe1006807

TAC-Twin: A Rapid Framework for Personalized Doppelgänger Avatar Creation Using a Modular Virtual Human Pipeline

Arno Hartholt, Kevin Kim, Edwin Sookiassian, Andrew leeds, and Ed Fast

USC Institute for Creative Technologies, Los Angeles, CA 90094, USA

ABSTRACT

We present an end-to-end framework for rapidly creating interactive, personalized avatars for scalable training and simulation applications. Built as an extension of the Virtual Human Toolkit, the framework integrates technologies for audio-visual sensing, speech recognition, natural language processing, nonverbal behavior generation, and high-fidelity text-to-speech synthesis. A personalized avatar is defined here as a real-time, embodied digital representation of an actual individual rather than a generic character. The creation pipeline requires only a single facial photograph, processed through a photorealistic character generation workflow, then refined, customized, and deployed in a real-time 3D environment for integration with conversational AI and synthetic voice generation. The system also supports rapid generation of generic avatars from high-quality synthetic headshots produced by generative AI, enabling the creation of diverse, realistic or stylized cohorts within minutes. Our initial use case examines whether personalized avatars enhance engagement, motivation, and performance compared to generic avatars, with the hypothesis that personalization increases relevance, identification, and learning outcomes. We describe the architecture, avatar creation pipeline, and role of generative AI in accelerating development, and share early implementation insights.

Keywords: Personalized avatars, Doppelgängers, Digital twins, Virtual humans, Embodied conversational agents, System architectures, Toolkits

INTRODUCTION

Virtual doppelgängers, or personalized avatars, are digital representations of real people (Bailenson et al., 2005), that can have many benefits, from increasing engagement (Lucas et al., 2016) to accelerating skill learning (Kleinlogel et al., 2021) and improving health outcomes (Turbyne et al., 2021). However, challenges remain that prevent broad adoption, including 1) creating isolated avatar attributes rather than comprehensive solutions (Liu et al., 2022), 2) proprietary solutions rather than open ones (Song, 2022), and 3) a lack of consensus on the required level of fidelity for different use cases (Mozgai et al., 2022). For the military domain, research on how best to use avatars is lacking (Hudson & Hurter, 2016).

Our work aims to help address these challenges by developing an integrated personalized avatar testbed, TAC-Twin, that combines academic and industry technologies into a common framework in support of rapidly developing new avatar-related prototypes. While our focus is on creating avatars in a military context, the framework uses a generic approach and is applicable to civilian uses.

In this paper, we outline the system architecture, detail the avatar creation pipeline, demonstrate how generative AI accelerates the creation of avatars, and share early implementation insights and lessons learned.

SYSTEM ARCHITECTURE

This work is an extension of the Virtual Human Toolkit[1] (VHToolkit), which aids with the creation of conversational agents (Hartholt et al., 2013). It combines audio-visual sensing, automated speech recognition (ASR), natural language processing (NLP), nonverbal behavior generation (NBG), and text-to-speech synthesis (TTS) into one integrated environment using the Unity game engine, enabling deployment to desktop, mobile, and AR/VR (Hartholt et al., 2019).

The VHToolkit leverages the architecture, extendable API, and cloud AI services provided by RIDE[2] (Figure 1, Table 1), a research & development platform that grew out of the U.S. Army's desire to prototype the next generation training and simulation system (Hartholt et al., 2021). It combines many simulation capabilities into a single framework, including synthetic real-world terrain, support for AI and machine learning (ML) frameworks, and networked multiplayer.

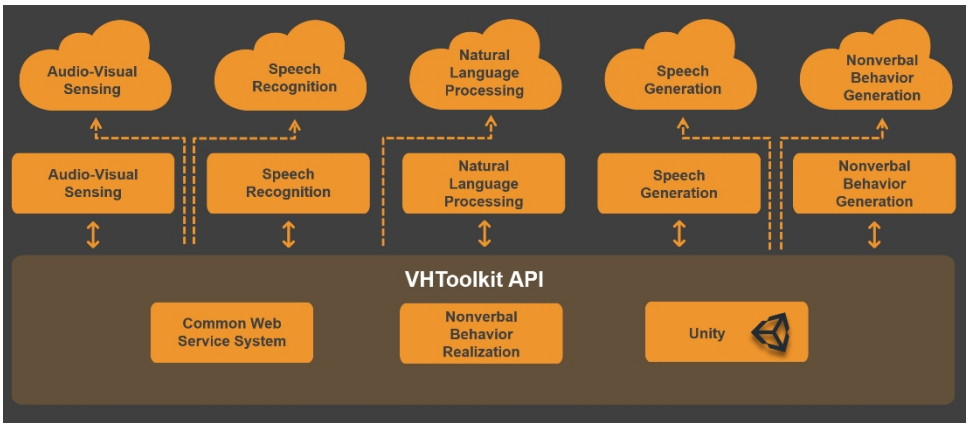

Figure 1: Virtual human toolkit architecture.

[1] https://vhtoolkit.ict.usc.edu

[2] https://ride.ict.usc.edu (Rapid Integration & Development Environment)

Table 1: Technology implementations provided by the virtual human toolkit, for audio-visual sensing, automated speech recognition (ASR), natural language processing (NLP), nonverbal behavior generation (NVB), and text-to-speech synthesis (TTS).

Sensing	ASR	NLP	NVB	TTS
AWS Rekognition (cloud)	Azure Speech (cloud)	Anthropic Claude (cloud)	NVBG (cloud)	AWS Polly (cloud)
DeepFace (local)	Android (local)	AWS Lex V2 (cloud)	NVBG (local)	ElevenLabs (cloud)
	iOS (local)	OpenAI ChatGPT (cloud)		MS SAPI (local)
	Windows (local)	RASA (local)		

TAC-TWIN PIPELINE

The TAC-Twin pipeline is designed as a flexible framework that transforms a single photo of a subject into an interactive, personalized avatar within the Virtual Human Toolkit. At its core, the pipeline integrates commodity software applications that support each stage of this transformation, from generating a 3D likeness to preparing the avatar for real-time interaction. While the architecture is modular and can accommodate different vendors or technologies, our current implementation leverages three widely adopted tools.

For avatar creation and rigging, we rely on Reallusion's Character Creator 4[3], a production-ready 3D character platform capable of exporting to game engines. It was selected for its robustness, interoperability, and suitability for repeatable workflows. To generate high-fidelity 3D heads from photographs, we use the Reallusion Headshot 2 plugin[4], which provides an effective balance between automation and manual refinement. Finally, Unity 6[5] serves as the real-time engine for configuration, animation, and deployment, offering a mature ecosystem and seamless integration with the VHToolkit and RIDE.

These tools represent the current state of the pipeline but are not intrinsic requirements of the architecture. Rather, they illustrate how TAC-Twin leverages available commercial solutions to accelerate avatar creation, while maintaining the flexibility to incorporate alternative technologies as the ecosystem evolves. The following seven steps describe how these components are combined in practice to generate a working avatar, beginning with input capture and pre-processing. The overall process takes approximately 20 minutes (Table 2), and is detailed below.

[3]https://www.reallusion.com/character-creator,Version4

[4]https://www.reallusion.com/character-creator/headshot,Version2

[5]https://unity.com,Version6

Table 2: Approximate time needed for TAC-Twin subprocesses.

Process	Approx. Time (m)
Take subject photo	2
Create generative AI headshot (alternative)	1
Transfer images	1
Generate full-body avatar	2
Refine & customize avatar	6
Export avatar	2
Unity import & setup	4
Build asset bundles	1
Test & validate	1
Configure conversational AI (optional)	3
Total time	~18–22 minutes

Step 1: Input Capture & Pre-Processing

The input data to the pipeline is a photo of a subject's head and face. The subject requires to be well lit, facing straight towards the camera, with a neutral facial expression, and nothing obscuring the face. An image resolution of 4k or higher is preferred, with a suggested lens of 70–200mm in order to minimize distortion of facial features. Supported image formats include .bmp, .jpeg, .png, .tif and.tiff. The input is either photos of real people or headshots created with generative AI.

1A: Photo of Real Person

Our capture setup consists of a white background, with the subject standing in front of two 5700K LED soft box photography light stands, aimed at 45 degrees towards the subject, see Figure 2. This ensures that the subject is lit evenly, with a minimum of shadows. Our capture device is an iPhone 15 Pro Max, mounted in portrait mode on an adjustable tripod. The camera is set to 2x zoom in order to avoid facial distortion. The height is adjusted to be at eye level of the subject. Subjects are asked to look at the camera with a neutral facial expression, to remove glasses, and to move any hair or accessories that obscure the face. While only a single frontal face image is required, two photos are taken in order to minimize chances of movements or eye blinks. In addition, a side view photo is taken for later reference. For privacy reasons, photos are only transferred from the phone to the processing computer using a USB cable and deleted from the phone afterwards.

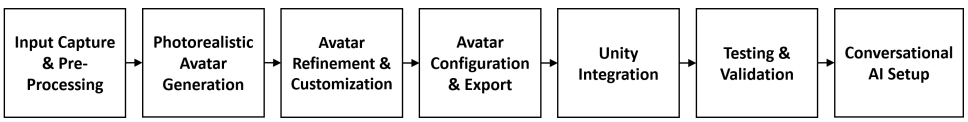

Figure 2: TAC-Twin personalized avatar pipeline.

1B: Generative AI Headshot

As an alternative to photos of real people, the pipeline can ingest headshots created by generative AI, enabling the rapid creation of a wide range of varied avatars. These headshots have the same image requirements as discussed above. We evaluated several generative AI services, including OpenAI, Stability AI, Adobe Firefly, and Midjourney, guided by three main metrics: 1) *Quality*; does the image match the photorealism of a real person, 2) *Lighting*; is the face evenly lit, with a minimum of shadows, and 3) *Adherence*; does the resulting image match the intent of the input prompt, including the ability to create a wide variety of characters. Our preferred choice is OpenAI ChatGPT 4o, which currently provides the best balance between realism, lighting, and variety, see Figure 4.

Step 2: Photorealistic Avatar Generation

The 2D input image from Step 1 is transformed into a full-body 3D avatar character in Character Creator (CC). The process is identical between photos of real people and generative AI created headshots, highlighting the flexibility of our pipeline in rapidly creating both personalized avatars and a varied range of generic characters.

CC Headshot offers two modes for creating an avatar from a photo: Auto and Pro. Auto mode uses machine learning to create a medium resolution avatar with proper head shape, full facial textures, and hair. This mode offers few customization options and is primarily used for crowds and other use cases where no close-ups of the avatar are required. Pro mode requires more manual intervention, but offers increased flexibility and a higher fidelity character that stands up to closer scrutiny. Our pipeline utilizes Pro mode.

CC Headshot Pro mode employs photo re-projection to automatically generate a high-fidelity 3D head based on the input image. CC extracts the facial texture from the input image, as well as skin color and eye color. The resulting head is fully articulated and includes eyes, eye lids, mouth, teeth, and tongue. This head is automatically combined with a generic body, which the user can configure to be male, female, or neutral. Both the body and head are fully rigged and skinned, and compatible with Unity's humanoid character rig. The head contains the appropriate blend shapes for facial animations and lip-synching, ensuring that the avatar can gesture, talk, and play general body and facial animations.

Step 3: Avatar Refinement & Customization

The automatically generated initial avatar character can be optionally refined and customized manually to more closely match the subject. This includes sculpting the head (Figure 3), adding (facial) hair, changing eye color and shape, tweaking skin color, introduce skin blemishes, and adding make-up. Frontal facial sculpting is assisted by the option to automatically overlay the original photo with changeable transparency. The side view photo taken in step 1A can be used here as a reference to sculpt the side profile of the head. Body type, size, and proportions can also be adjusted. CC provides a basic wardrobe with optional purchases. In addition, we created CC-compatible

U.S. Army OCP uniforms, including battle gear, enabling us to create personalized soldier avatars. See Figure 4 for a side-by-side comparison between input images and output avatars in CC.

Figure 3: Capture setup.

Figure 4: CC sculpting tool.

Step 4: Avatar Configuration & Export

Once finalized, the avatar is configured for export. This includes selecting the proper viseme mode for increased compatibility with VHToolkit lip-synching, using subdivided meshes for increased performance, and setting the export target to Unity. The character is exported as a mesh-only FBX with embedded textures.

Step 5: Unity Integration

The resulting character FBX is imported into a dedicated VHToolkit Unity soldier avatar testbed project. Reallusion's Unity plugin creates a Unity prefab for use in one of Unity's main rendering pipelines: URP (Universal Rendering Pipeline), aimed at mobile, web, and AR/VR applications, and HDRP (High-Definition Rendering Pipeline), aimed at higher-fidelity desktop applications.

The examples in this paper are rendered in HDRP. Next, in-house developed automation scripts configure the character prefab for use in the personalized soldier testbed, which includes adding the proper animation suite, dynamic gaze controller, eye blinks, eye saccades, UI portraits, and decals (e.g., combat wounds).

All characters are built as Unity addressable assets, so that they can be used directly in binary applications, without the need to rebuild an executable. Once the asset bundle for the new avatar is built, the character can be reviewed on a virtual turntable (Figure 5) and embedded in a custom scenario (Figure 6).

Figure 5: OpenAI ChatGPT generated headshots and their CC soldier avatars.

Step 6: Testing & Iteration

All soldier avatars are automatically ingested in the personalized soldier avatar testbed, where they can be reviewed on a real-time 3D turntable and embedded within a combat scenario. This scenario includes a main passenger avatar, who can either be a personalized one (i.e., created from a photo of a real person) or a generic one (i.e., created from a generative AI headshot). A second, back-seat avatar is always a generic avatar. These selections are made from a configuration panel, seen in Figure 7, that is automatically populated with all the available avatars, represented by the input images created in Step 1 and set up in Step 5. The semi-automated setup of Unity character prefabs and the ability to ingest new characters into the application without having to rebuild it, combined with the automated population of the configuration panel, ensures that new avatars can rapidly be reviewed and embedded in the intended scenario.

Figure 6: Subject and unity personalized soldier avatar.

Figure 7: Personalized soldier avatar embedded in unity combat scenario.

Step 7: Conversational AI Setup

Optionally, the created avatar can be set up as a conversational AI agent. A dedicated VHToolkit Unity project contains all integrated technologies from Table 1. The character prefab from Step 5 is extended to include the use of a conversational gesture suite and the ability to interpret dynamic nonverbal behavior schedules. The character is then set up to use specific AI technologies (e.g., ChatGPT, ElevenLabs) and configurations (e.g., LLM prompt, cloned voice).

Note that some choices involve trade-offs. For instance, cloud services can use more compute power than a local solution, but require an internet connection and possible subscription. Furthermore, while ElevenLabs provides higher quality voices and voice cloning, it lacks a traditional

phoneme timing schedule required for lip-synching that AWS Polly does provide. Consequently, choices are currently between higher quality voices with lower quality lip-synching or vice versa. The modular architecture of the VHToolkit means it can take advantage of more advanced solutions when they become available.

LESSONS LEARNED

The TAC-Twin effort underscored several practical and conceptual lessons that will inform both ongoing research and production deployment of avatars.

Lighting and Perceptual Fidelity. Our work reaffirmed that environment lighting exerts a stronger influence on perceived fidelity than likeness alone. Flat lighting—whether in real-world capture or generative AI images—produced avatars that felt less lifelike, while carefully controlled lighting substantially improved realism. Subtle appearance factors such as glasses, facial hair, helmets, and makeup dramatically altered recognition of likeness, illustrating how easily perception shifts based on context.

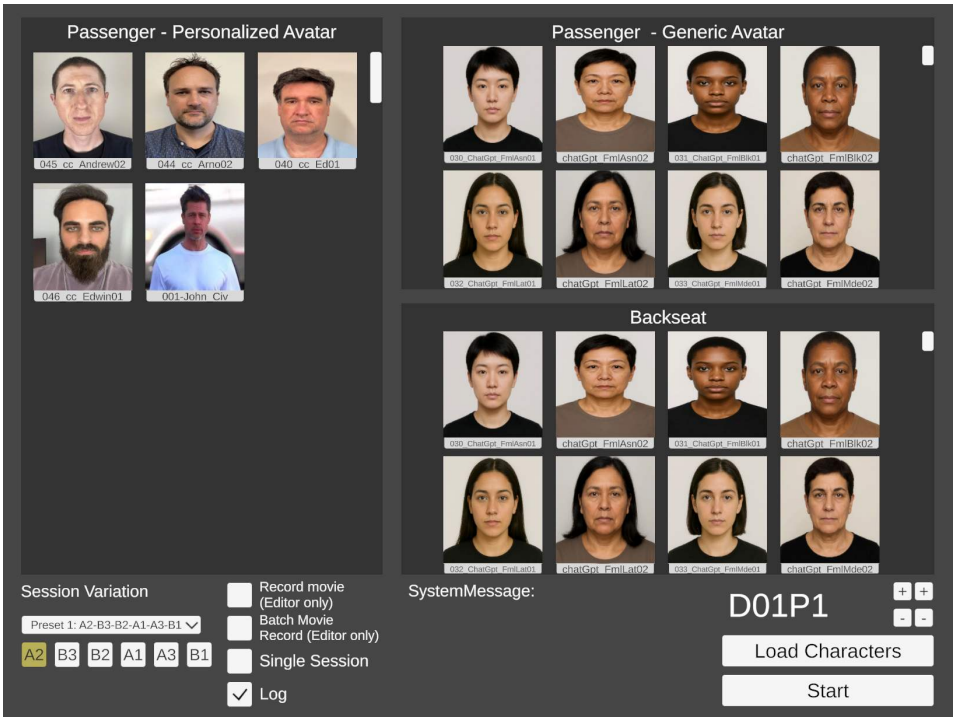

Figure 8: Personalized soldier testbed configuration and selection panel.

Working Within Technological Constraints. Study designs must be tailored to the current limitations of avatar generation technology. For example, the use of helmets and body armor allowed us to sidestep challenges with hair and body realism while still producing recognizable, field-appropriate avatars. Aligning high-quality facial likeness with personalized bodies remains difficult, with hair continuing to be a persistent technical challenge.

Pipeline Robustness and Scalability. Production-ready tools such as Character Creator and Unity enabled us to build a repeatable and extensible pipeline. However, the "last 10%"—the polish phase of smoothing, texturing, and integration—proved disproportionately time-consuming. Clear scoping is critical, particularly when balancing fidelity against scalability. Research-grade software often offers flexibility but lacks the robustness required for repeated deployment in real-world contexts.

Beyond Visual Likeness. Once likeness was established, the challenge became behavioral realism. Movement, gesture, facial expressions, and prosody all proved more influential than static appearance in conveying character and personality. This highlights that the true difficulty in creating convincing avatars lies not in resemblance alone, but in capturing the subtle dynamics of human expression.

Extensibility Through Modularity. The use of principled APIs and a modular architecture allowed diverse input sources—from photographs to generative AI outputs—to be processed through a single pipeline. This extensibility also enabled experimentation across multiple NLP, ASR, and TTS providers. Trade-offs were unavoidable: high-quality cloned voices (e.g., ElevenLabs) lacked phoneme schedules necessary for accurate lip-sync, while lower-fidelity options (e.g., AWS Polly) integrated more seamlessly. Designing with modularity ensured the system could adapt to evolving technologies without requiring fundamental redesign.

In sum, these lessons point toward a central theme: building believable avatars requires balancing fidelity, scalability, and extensibility, with attention not just to likeness but to the lived expressivity that gives virtual humans their impact.

CONCLUSION

The TAC-Twin framework extends the Virtual Human Toolkit by demonstrating how principled APIs, modular architecture, and production-ready tools can accelerate the creation of interactive, personalized avatars. Several core contributions emerged from this work:

- **Principled APIs as Accelerators**. Once ChatGPT was integrated, adding additional LLMs became straightforward; the same was true for TTS vendors. This design reduces overhead and ensures adaptability as the AI ecosystem continues to evolve.
- **Standards-Driven Modularity**. By grounding the system in shared standards, CC avatars, ICT legacy characters, and other models were able to re-use animations and behavior schedules, enabling interoperability across otherwise siloed workflows.
- **Explicit Trade-Offs**. The work surfaced inherent trade-offs between scalability and fidelity (CC Auto vs. Pro mode) and between fidelity attributes (high-quality voice cloning without phoneme schedules vs. lower-quality voices with strong lip-sync). Recognizing and managing these trade-offs is a necessary condition for successful deployment.
- **Designing With Constraints**. Strategic design mitigated technological weaknesses: body armor and helmets masked challenges with hair

and body realism, while well-defined input formats allowed avatars to be generated from multiple sources (e.g., photographs, generative AI headshots).

Looking forward, TAC-Twin illustrates the potential for avatar creation pipelines that are both rapid and extensible, opening pathways for scaling virtual human research and application. Future work should focus on deepening behavioral realism, integrating multimodal generative models, and personalizing avatars at scale for use in health, education, training, and operational domains. As the boundaries between research-grade prototypes and production-ready systems continue to blur, the key will be designing pipelines that balance fidelity with scalability, while remaining adaptable to new technological advances.

ACKNOWLEDGMENT

The authors would like to acknowledge their many VHToolkit and RIDE collaborators, as well as the USC Dornsife Brain and Creativity Institute. Part of the efforts depicted were sponsored by the US Army under contract number W911NF-14-D-0005. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

Bailenson, J. N., Swinth, K., Hoyt, C., Persky, S., Dimov, A., & Blascovich, J. (2005). The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive. *Ieeexplore. Ieee. OrgJN Bailenson, K Swinth, C Hoyt, S Persky, A Dimov, J BlascovichPresence, 2005.ieeexplore. Ieee. Org*, *14*(4), 379–393. https://ieeexplore.ieee.org/abstract/document/6790505/

Hartholt, A., McCullough, K., Fast, E., Leeds, A., Mozgai, S., Aris, T., Ustun, V., Gordon, A., & McGroarty, C. (2021). Rapid Prototyping for Simulation and Training with the Rapid Integration & Development Environment (RIDE). *I/ITSEC*.

Hartholt, A., Mozgai, S., Fast, E., Liewer, M., Reilly, A., Whitcup, W., & Rizzo, A. S. (2019). Virtual humans in augmented reality: A first step towards real-world embedded virtual roleplayers. *HAI 2019 - Proceedings of the 7th International Conference on Human-Agent Interaction*, 205–207. https://doi.org/10.1145/3349537.3352766

Hartholt, A., Traum, D., Marsella, S. C., Shapiro, A., Stratou, G., Leuski, A., Morency, L.-P., & Gratch, J. (2013). All Together Now: Introducing the Virtual Human Toolkit. *Intelligent Virtual Agents; IVA 2013*, 368–381. https://doi.org/10.1007/978-3-642-40415-3_33

Hudson, I., & Hurter, J. (2016). Avatar types matter: Review of avatar literature for performance purposes. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9740*, 14–21. https://doi.org/10.1007/978-3-319-39907-2_2

Kleinlogel, E. P., Curdy, M., Rodrigues, J., Sandi, C., & Mast, M. S. (2021). Doppelganger-based training: Imitating our virtual self to accelerate interpersonal skills learning. *PLoS ONE*, *16*(2 February). https://doi.org/10.1371/JOURNAL. PONE.0245960

Liu, S., Cai, Y., Chen, H., Zhou, Y., & Zhao, Y. (2022). Rapid Face Asset Acquisition with Recurrent Feature Alignment. *ACM Transactions on Graphics*, *41*(6). https://doi.org/10.1145/3550454.3555509

Lucas, G., Szablowski, E., Gratch, J., Feng, A., Huang, T., Boberg, J., & Shapiro, A. (2016). The effect of operating a virtual doppleganger in a 3D simulation. *Proceedings - Motion in Games 2016: 9th International Conference on Motion in Games, MIG 2016*, 167–174. https://doi.org/10.1145/2994258.2994263

Mozgai, S., Winn, J., Kaurloto, C., Leeds, A., Heylen, D., & Hartholt, A. (2022). Toward a semi-automated scoping review of virtual human smiles. *Workshop on Smiling and Laughter across Contexts and the Life-Span within the 13th Language Resources and Evaluation Conference*, 1–5. https://aclanthology.org/2022.smila-1.1/

Song, M. (2022). Meta's Metaverse platform design in the pre-launch and ignition life stage. *International Journal of Internet, Broadcasting and Communication*, *no. 4*(14), 121–131. https://www.earticle.net/Article/A421038

Turbyne, C., Goedhart, A., de Koning, P., Schirmbeck, F., & Denys, D. (2021). Systematic Review and Meta-Analysis of Virtual Reality in Mental Healthcare: Effects of Full Body Illusions on Body Image Disturbance. *Frontiers in Virtual Reality*, *0*, 39. https://doi.org/10.3389/FRVIR.2021.657638