virtual worlds

MDPI

Article

Planogen: A Procedural Generation Framework for Dynamic VR Research Environments

Kaitlyn Tracy [1,*], Lazaros Rafail Kouzelis [2], Rami Dari [1] and Ourania Spantidi [1]

[1] Embedded AI Systems Lab, Eastern Michigan University, Ypsilanti, MI 48197, USA; rdari5@emich.edu (R.D.); ourania.spantidi@emich.edu (O.S.)
[2] Department of Multimedia and Graphic Arts, Cyprus University of Technology, Limassol 4630, Cyprus; lc.kouzelis@edu.cut.ac.cy
* Correspondence: ktracy2@emich.edu

Abstract

This paper introduces *Planogen*, a modular procedural generation plug-in for the Unity game engine, which is composed of two primary components: a character generation module (*CharGen*) and an airplane generation module (*PlaneGen*). *Planogen* facilitates the rapid generation of varied and interactive aircraft cabin environments populated with diverse virtual passengers. The presented system is intended for use in research experiment scenarios, particularly those targeting the fear of flying (FoF), where environmental variety and realism are essential for user immersion. Leveraging Unity's extensibility and procedural content generation techniques, *Planogen* allows for flexible scene customization, randomization, and scalability in real time. We further validate the realism and user appeal of *Planogen*-generated cabins in a user study with 33 participants, who rate their immersion and satisfaction, demonstrating that *Planogen* produces believable and engaging virtual environments. The modular architecture supports asynchronous updates and future extensions to other VR domains. By enabling on-demand, repeatable, and customizable VR content, *Planogen* offers a practical tool for developers and researchers aiming to construct responsive, scenario-specific virtual environments that can be adapted to any research domain.

Keywords: procedural content generation; virtual reality; parametric asset generation

Academic Editor: Daniel R. Mestre

Received: 19 May 2025
Revised: 15 June 2025
Accepted: 30 June 2025
Published: 14 July 2025

Citation: Tracy, K.; Kouzelis, L.R.; Dari, R.; Spantidi, O. Planogen: A Procedural Generation Framework for Dynamic VR Research Environments. *Virtual Worlds* **2025**, *4*, 33. https://doi.org/10.3390/virtualworlds4030033

1. Introduction and Related Works

Digital content creation plays a crucial role in enabling empirical research across fields such as human–computer interaction (HCI), psychology, and education. Researchers often need to utilize controlled, interactive scenarios to test hypotheses, simulate real-life conditions, and collect behavioral data, usually having to create custom scenarios that fit their needs. These scenarios may involve complex environments, task sequences, dynamic agents, or multimodal stimuli, requiring both creative and technical experience. However, building such content typically requires expertise in programming, visual design, or interaction design skills in which domain experts conducting user studies are not regularly trained [1]. This creates a barrier in research workflows, where scientists must either collaborate with technical developers or limit the scope of their experiments to what is feasible with their technical skills.

The need for accessible experimental content creation becomes even more pronounced in the context of virtual reality (VR) applications, where researchers must design immersive 3D environments featuring spatial interactions, space adaptivity, and dynamic

agent behaviors. Popular VR development tools, such as game engines or interaction design frameworks, offer powerful capabilities but present steep learning curves for non-developers. Furthermore, rapidly evolving hardware and the absence of a universal design philosophy create significant obstacles for non-developers [2]. As a result, domain experts often struggle to prototype or adapt VR scenarios for studies, limiting the diversity and adaptability of experimental designs [1,3].

Attempting to alleviate these issues, a growing body of research has focused on scenario creation systems that enable non-technical users to build and customize VR simulations for use in experiments. Published works range from minimal web-based interfaces to rich modular scenario frameworks, many of which have been evaluated in applied settings such as therapy, training, or robotics [3,4]. One such example is the concept of "VR nuggets", a pattern-based authoring approach for educators, which has improved non-experts' ability to create educational VR content [5]. Likewise, the web-based tool *VREUD* enables users to successfully author interactive VR scenes without prior experience [3]. A similar approach allows automotive experts to evaluate driving scenarios using a configurable small-scale environment and block-based scripting, and it was reported to offer high satisfaction, aside from minor interface issues [6].

A recurring theme in these systems is the facilitation of scenario variability—the ability to manipulate different factors (such as varying environments, agents, or conditions) without the need to reprogram the application to account for new variables. To this end, researchers have increasingly incorporated procedural content generation (PCG) methods into VR scenario creators, which can produce diverse virtual environments, objects, and scenarios with minimal effort from the end-user. This methodology allows for the rapid generation of VR-ready interior scenes based on simple textual prompts, utilizing a local library of assets and the ChatGPT 4 API, a popular large language model (LLM) [7]. A driving training application was developed to identify poor driving habits in users and adapt to events that occurred during the experience in real time [8].

Further examples of PCG integration can be found in recent VR applications developed for education and therapeutic purposes. An escape room-based learning platform leverages Unreal Engine's PCG framework to support rapid scene prototyping and reduce development times, enabling the simulation of new educational scenarios with minimal manual work [9]. In a VR safety training system for cyclists, the authors employed procedural generation to automate the placement of hazards and traffic elements, allowing for scalable scenario diversity aligned with real-world hazards and situations [10]. A therapy-focused application in gait rehabilitation used procedural environment evolution to maintain patient engagement, with the virtual world adapting dynamically during walking tasks [11].

The works aforementioned illustrate that PCG is not only a technical convenience but a fundamental requirement for the design of scalable, adaptable, and personalized VR experiences. As VR applications are expected to accommodate more complex requirements, integrating procedural generation becomes crucial in sustaining variability across repeated experiences.

Scientific Gaps and Our Contribution

Most of the existing literature on VR scenario creation tools centers on domain-specific implementations, limited-use prototypes, or systems primarily designed for internal research purposes. These tools often prioritize experimental control over flexibility, offering limited customization options or relying on non-deterministic content randomization to introduce variability. As a result, visual fidelity and scenario structures are typically secondary concerns. Many applications use low-poly assets or abstract representations that

suffice for research goals but fall short in more demanding contexts [12–14]. This is particularly limiting in therapy or high-stakes training, where visual realism and contextual precision can directly impact user immersion, emotional engagement, and overall effectiveness. Moreover, many systems lack modular architectures, are not reusable outside their original teams, and rarely support deployment across diverse user groups or domains [15].

The present work addresses the identified gap in accessible, high-quality VR scenario creation by introducing a Unity-based extension that enables users to generate realistic, customizable airplane interior environments without programming expertise. Designed initially for exposure therapy targeting fear of flying (FoF), the tool allows researchers and clinicians to configure key parameters, such as the aircraft size, occupancy level, and passenger appearance, all through a simple interface. This approach not only fills a gap in the literature regarding domain-specific, usable VR scenario generators but also demonstrates broader applicability.

2. Materials and Methods

The system presented in this work, *Planogen*, is a modular Unity-based plug-in designed to support the rapid prototyping and customization of VR-ready environments and characters for research applications. The architecture of *Planogen* is divided into two independent yet interoperable components: the *PlaneGen* module, which generates modular airplane interiors, and the *CharGen* module, which enables the creation of procedurally varied human characters. Each module can be used independently, allowing the user to employ only the components relevant to their use case.

2.1. Technical Framework

Planogen was developed entirely within the Unity engine, chosen for its ease of integration and widespread adoption in both academia and industry. It is accessible through a simple menu item in the Unity Editor interface. The full functionality of the plug-in was authored in the programming language C#, leveraging Unity's runtime environment to enable in-editor configuration, modular asset loading, and runtime instantiation of scenes and characters. Asset creation and processing were performed in Blender, selected for its open-source flexibility and support for automation via the embedded Python scripting language in Blender version 4.3. Regarding the specific use of the '_FRONT' and '_BACK' empty objects (discussed in more detail in Section 2.2), a custom Blender script was created to automatically place these empties based on the module's visual boundaries, ensuring consistent spacing and seamless snapping when modules were later assembled in Unity.

The base body meshes used in the *CharGen* module (male and female) were sourced from CC0-licensed libraries, while animation data were adapted from Mixamo and then manually edited for compatibility and refinement. All other assets, including clothing, hair, props, and textures, were custom-made by the authors for *Planogen*.

The *Planogen* interface was designed to support use by non-technical users, such as clinicians or domain researchers, through an intuitive Unity Editor panel that allows users to configure key parameters (e.g., aircraft size, occupancy, passenger characteristics) via simple drop-down menus and sliders. No programming or scripting is required to generate new cabin layouts or passenger populations. The inner machinations of these modules are discussed in Sections 2.2 and 2.3. For a technical overview of *Planogen*'s development lifetime, as well as the user creation loop, please refer to the detailed flowchart in Appendix A.

At present, the tool generates static airplane cabin environments with passengers exhibiting minimal movement and interaction capabilities; thus, there are no current configurations that would be considered unsafe or clinically inappropriate. As future

versions of *Planogen* introduce more dynamic scenarios and interactive elements, additional safety validation mechanisms will be incorporated to ensure appropriate scene generation.

Planogen is demonstrated in an airplane cabin context, but its modular prefab architecture and anchor-based alignment system are designed to be generalized across other modular environments, such as trains or buses. Scene-specific modules can be substituted by altering the configuration parameters while reusing the core alignment logic and asset management strategies. For example, future enhancements to the tool could include adapting *Planogen* to generate a modular bus interior. This would involve substituting airplane seat prefabs with bus seat prefabs, adjusting the layout parameters to reflect the aisle width and row spacing typical of bus environments, and reusing the same alignment anchors and passenger placement logic.

2.2. Plane Generator Module

The *PlaneGen* module is a modular environment assembly tool for constructing airplane interiors using predefined 3D modules, which can be changed and customized by the user without issues. Unlike traditional modeling techniques that offer extensive visual control but sacrifice adaptability and reusability, or parametric modeling approaches that generate geometries based on adjustable parameters or conditions [16–18] but offer less visual control, *Planogen*'s *PlaneGen* module operates by combining curated, modular prefabs in a final scene. This method attempts to strike a balance between visual fidelity and adaptability, as well as ensuring ease of integration within existing Unity workflows.

An overview of the plane generation module is shown in Figure 1. Each airplane layout is composed of four types of modules: a *Start* module (such as cabin doors or the cockpit), an *End* module (such as the rear galley), a repeatable *Middle* module (normally the seating rows) that constitutes the majority of the modules, and an optional *Special* module (such as toilets or flight class separators). The user defines how many repetitions of each module type should be included, and the system instantiates them in sequence. The *Special* modules are placed in equal spaces, based on the number of *Middle* modules.

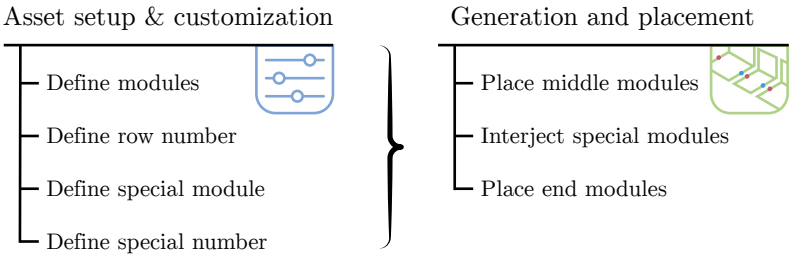


Figure 1. Overview of the *Planogen* plane generation (*PlaneGen*) workflow: After defining the modules and number of instances, the tool generates the environment.

To ensure seamless spatial alignment, each module prefab must include two empty objects (i.e., without geometry data) labeled '_FRONT' and '_BACK', positioned on its horizontal boundaries, to define the start and end of a module. During assembly, the system aligns each new module's '_FRONT' marker with the previous module's '_BACK', allowing the segments to attach precisely. This approach was adapted from previous work [7] and is designed to avoid geometry misalignment, such as gaps or overlapping meshes that could result in z-fighting.

With this method, the customization of the modules is straightforward, allowing non-technical users to utilize any modules that they wish, as long as they contain the necessary empty objects. This approach simplifies the process of constructing large-scale environments while allowing for high control over the layout and visual detail. It is

especially suited for scenarios where esthetic consistency, asset reuse, and predictable layouts are priorities, such as in training or therapeutic VR simulations.

2.3. Character Generator Module

The *CharGen* module is a customizable character generation system that supports both user-defined and default character assets. An overview of the *CharGen* workflow can be referenced in Figure 2. All body meshes are required to conform to a shared skeletal rig and consistent skin weighting, ensuring compatibility across accessories and animation states. Body textures rely on a standardized UV layout, allowing users to extend the included library of skin tones—light, medium, and dark—by simply modifying or replacing the existing textures.

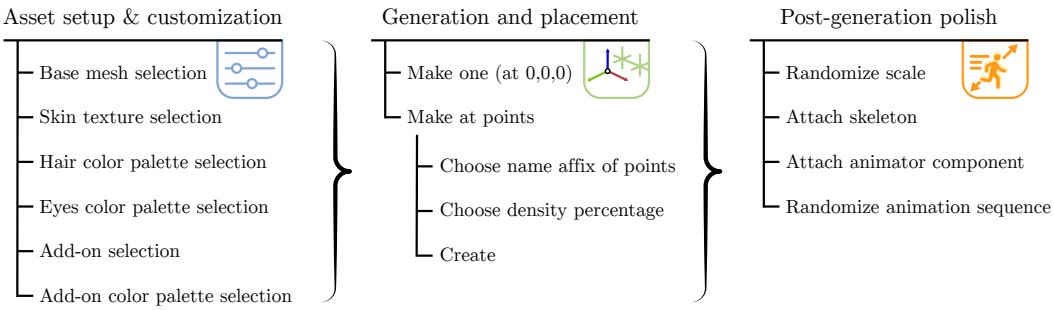


Figure 2. Overview of the *Planogen* character generation (*CharGen*) workflow: First, we define assets such as base meshes, skin textures, accessories, and color palettes. Afterwards, we define the placement mode and positioning. Finally, the tool applies some extra modifications and populates the scene.

Character appearance is procedurally created by sampling from user-defined or default asset sets. These sets include interchangeable components such as hairstyles, eyebrow meshes, clothing items, and accessories. Each of these assets is authored with a grayscale texture, which serves as the base for colorization. During character generation, the system assigns colors to each component based on a palette specified by the user or drawn from the default configuration provided by *Planogen*. These palettes are by default constrained to natural tones (e.g., black, brown, or blonde for hair) but can be customized freely.

Figure 3 illustrates the full set of default assets provided by *Planogen*. Out of the box, each character (male or female) can be customized with three skin tones, four hairstyles, three tops, and one type of bottoms. In addition, female characters have two shoe options, while male characters have one. All of these options can be swapped out or extended by end users.

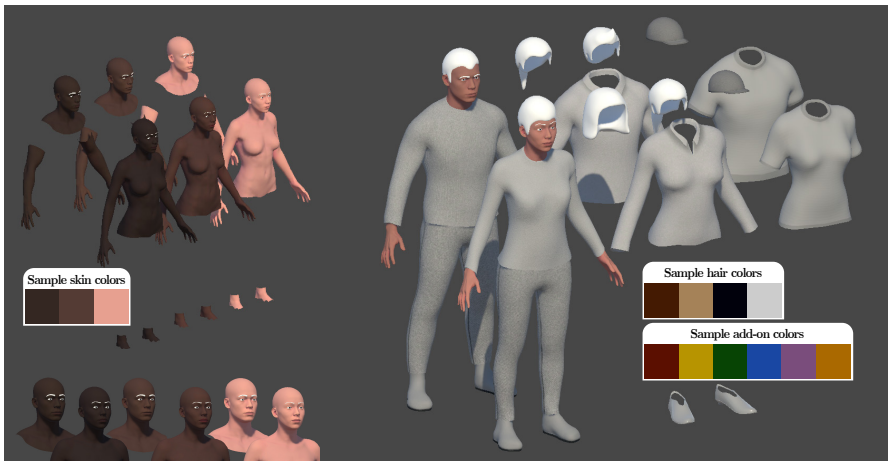


Figure 3. Overview of the default *CharGen* library that ships with *Planogen*.

Color randomization is implemented using Unity's base color input and vertex coloring. The vertex color system applies per-vertex RGB values to the mesh, allowing fine-grained tinting without modifying the original texture. This technique enables visual diversity while maintaining a lightweight and responsive character pipeline. Combined with asset randomization, this approach supports the generation of large populations of varied, believable characters without requiring unique models or textures.

Characters are then placed in the scene either via the *Make One*or *Make At Points* modes. *Make One* instantiates single or multiple characters at world origin and is primarily used for testing asset compatibility. *Make At Points* places characters at the positions of empty GameObjects in the scene whose names match or contain a given string. This allows the user to define spawn points manually or procedurally, while the system populates them with characters. A Density parameter (ranging from 1–100%) controls how many of the matching spawn points are used during character placement. When generating characters in the *Make At Points* mode, the system selects a random subset of the matching spawn locations based on the specified density value. Formally, if the scene contains N objects whose names begin with SEAT, and the user specifies a density D (as a percentage), then the number of characters generated is $\lfloor N \times \frac{D}{100} \rfloor$.

To enhance the visual variation among the generated characters, the system also introduces slight randomization in character scale. By default, male and female base meshes are scaled around 1.75 m and 1.65 m, respectively, with a variation range of $\pm 10\%$, resulting in more natural differences in height across the generated characters. Finally, characters are given an Animator Controller, a component that enables animation states and transitions. For use cases requiring unique behaviors or motions, separate Animator instances can be assigned to each character after generation.

To summarize, the *CharGen* generation pipeline proceeds in three main stages:

1. **Seat detection:** Find all scene objects whose names begin with `SEAT`.
2. **Spawn selection:** Randomly choose spawn points, weighted by the user-specified density.
3. **Character assembly:**
    a. Select a base mesh (male or female) at random.
    b. Assign a skin material.
    c. Attach a top prefab.
    d. Attach a bottom prefab.
    e. Attach a shoe prefab.

Figure 4 provides a scene view of *CharGen* in action. The image highlights automatic seat assignment, scalable crowd densities, and the visual diversity achieved through random mesh, color, and scale variations, all while preserving collision-free alignment within each seat row.

Figure 4. Partial scene view of the random character generation for a density of 1, with the generation button pressed several times.

3. Results

This section will present screenshots and representative use cases to showcase *Planogen*'s accessibility for non-technical users, configurability, and potential for creating immersive VR-ready environments. At the end of this section, we present the results of a user study designed to gauge participants' immersion and satisfaction in a virtual airplane cabin populated with passengers generated by *Planogen*.

3.1. Representative Generated Cabins

A view of the same narrow-body cabin instance synthesized by *Planogen*'s *PlaneGen* and *CharGen* pipelines is shown in Figures 5–7. Collectively, these three perspectives on a single generated cabin illustrate *Planogen*'s ability to maintain structural and esthetic coherence across multiple views, while offering flexible, real-time customization for VR-based research scenarios.

To demonstrate the fidelity and flexibility of the *PlaneGen* module, Figure 5 presents a fully assembled narrow-body aircraft cabin. The final outcome is characterized by consistent seat spacing, overhead bin alignment, and realistic lighting gradients. These are all attributes that are often labor-intensive in manual modeling but are generated on-demand with a single configuration step in *Planogen*.

Figure 6 showcases how the *CharGen* module populates the identical cabin context with a randomized ensemble of virtual passengers. Each avatar's mesh, skin tone, apparel, and scale vary subtly within predefined palettes, as mentioned in Section 2.3, ensuring diversity while preserving a coherent fit within the same seating geometry. Evidently, the passenger placement logic respects cabin constraints (such as head clearance and seat boundaries).

Figure 7 presents a VR user perspective from one of the generated seats. Note that, since *Planogen* is implemented as a plug-in for the Unity game engine, integration with a VR headset is seamless.

Figure 5. Sample aisle view of the plane cabin interior as generated using *Planogen*.

Figure 6. A close-up of the passengers created through the *CharGen* module of *Planogen*.

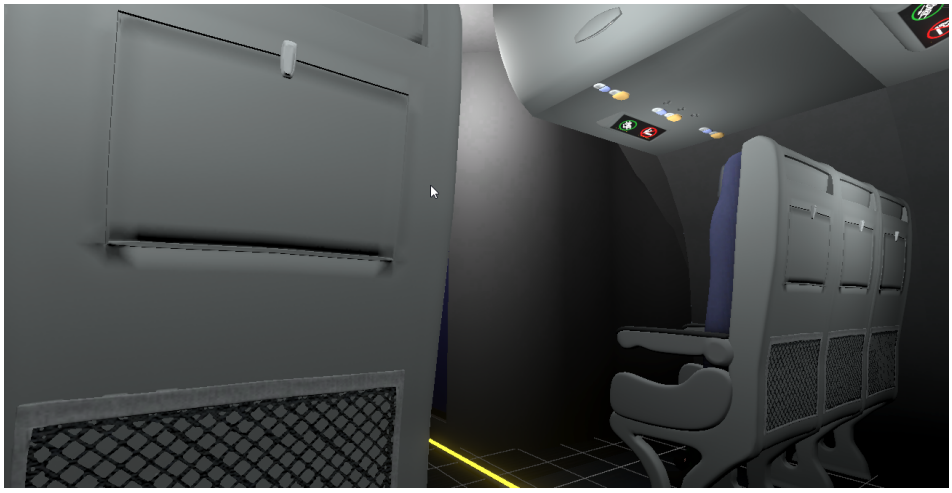


Figure 7. Example VR user view from seat.

3.2. Example Customization Outcomes

Beyond per-cabin customizations, the PlaneGen module supports variable-length configurations to suit different aircraft types or experimental setups. Figures 8 and 9 compare two generated cabins.

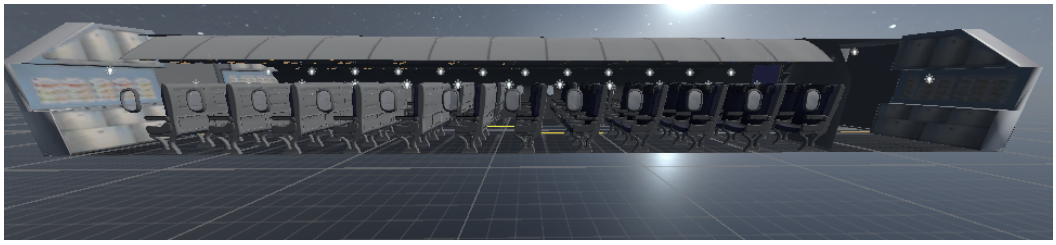


Figure 8. Scene view for a small plane generated with *Planogen*.

Figure 8 shows a smaller configuration that could be optimized for regional flights or prototype testing. The shorter aisle and reduced seat count enable faster load times in the editor and minimal resource consumption in VR. This configuration was generated with the *PlaneGen* module and is composed of one Start_Service_Module Front Cap, 12 Plane_Middle_Module Main Modules, one Toilet_Module Extra Module, and one End_Service_Module End Cap.

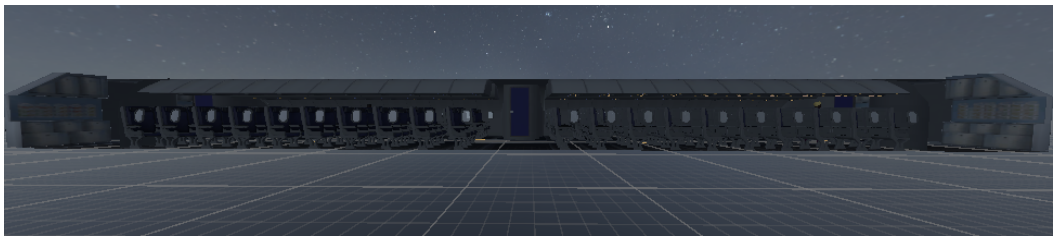


Figure 9. Scene view for a larger plane generated with *Planogen*.

A larger plane configuration is shown in Figure 9. Despite the doubled length, the snapping logic and material assignments remain identical, showcasing the performance scaling and consistency of *Planogen*. This larger configuration was generated with the *PlaneGen* module and is composed of one Start_Service_Module Front Cap, 20 Plane_Middle_Module Main Modules, three Toilet_Module Extra Modules, and one End_Service_Module End Cap.

This ability to adjust the module counts on the fly allows researchers to benchmark the performance impacts of scene complexity (e.g., frame rate, memory footprint). Additionally, specifically in a research environment context, *Planogen* facilitates the comparison of user experiences across cabin lengths (e.g., perceived crowding, spatial orientation). Finally, the main contribution of *Planogen* is the ability to prototype custom aircraft interiors (e.g., premium vs. economy layouts) without additional asset creation.

3.3. Performance and Stability

While *Planogen* is designed as a design-time authoring tool rather than a runtime system, we provide indicative performance metrics to characterize its operating efficiency. *PlaneGen* achieves near-instant generation with peak CPU times of ∼10–73 ms depending on the scene scale, while *CharGen* completes batch creation (50% density) in approximately 4 s, with temporary CPU peaks during instantiation. Currently, the *CharGen* and *PlaneGen* workflows are completed at separate times, ideally starting with *PlaneGen* so that the passengers generated with *CharGen* have seat objects to be placed into. Passengers can be generated and placed in an already populated plane, so, if the user decides that the initial density is not to their satisfaction, instead of deleting all of the passengers and selecting a new density, they can simply press the Generate button on the *CharGen* UI. This generation can be refined by updating specific Seat objects that the user would like to populate using the Match Name in Scene option from the *Make At Points* module. *Make At Points* is referred to as "Generate on points" in the *CharGen* UI, shown in Figure 10. Users can generate multiple plane cabins, each populated with passengers, within one Unity scene. This does not seem to affect the performance or scene stability during VR engagement.

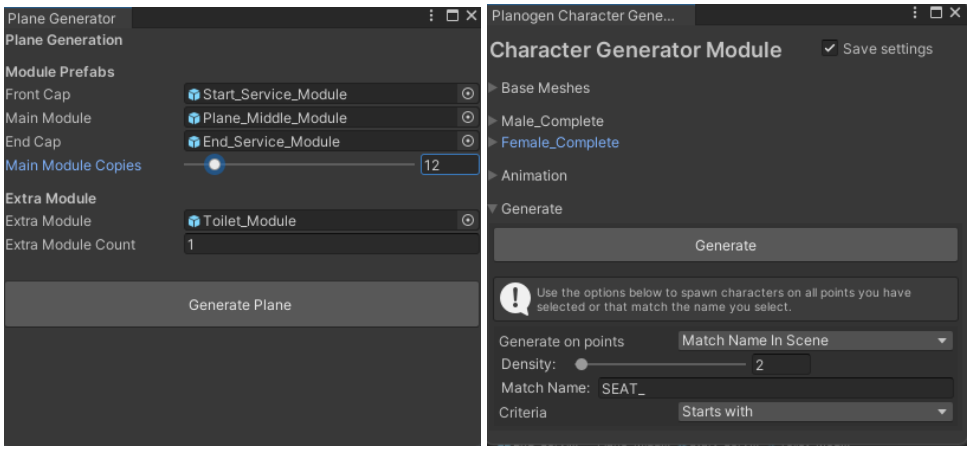


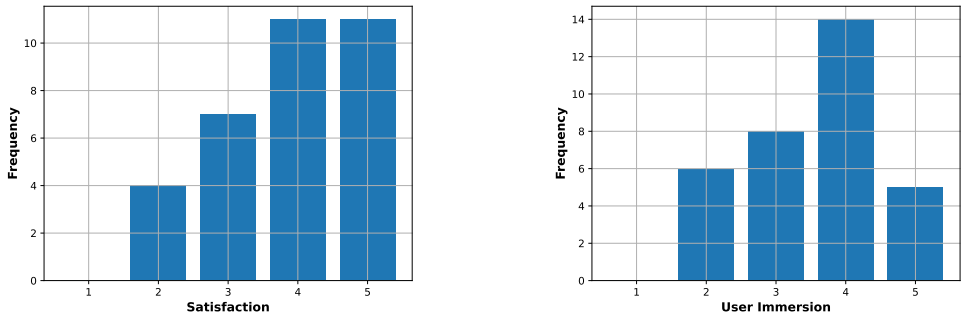
Figure 10. The two user interfaces for the two modules of *Planogen*, namely *PlaneGen* (**on the left**) and *CharGen* (**on the right**).

3.4. Pilot User Study

We generated a virtual narrow-body cabin with 200 seats and a passenger density of 10 (20 total passengers) to keep the cabin sparsely populated and avoid triggering responses beyond our scope. Thirty-three participants (17 female, 15 male, 1 non-binary, with an age range of 18–52, M = 28.4, SD = 6.7) were each seated in a chair with armrests mimicking an airplane seat. This study was reviewed and approved by the Eastern Michigan University Human Subjects Review Committee (UHSRC) under IRB number UHSRC-FY24-25-201. All participants provided informed consent prior to participating in the study. Each participant was given two minutes to explore the environment, and they were informed that they could stand and move around if they wished. Immediately afterward, they were asked by the experiment facilitator two questions (on a 5-point Likert scale):

1. How satisfied are you with the VR flight experience?
2. How immersed do you feel?

The histogram of satisfaction scores is shown in Figure 11a and shows that most participants rated their VR flight experience positively, with a clear peak at 4 and 5. Only a few respondents selected the lower end of the scale (2), resulting in a slight left skew. The mean satisfaction was 3.88 (SD 1.02), indicating generally favorable reactions.



(**a**) (**b**)

Figure 11. Participant satisfaction and immersion ratings (left and right, respectively). (**a**) Distribution of satisfaction ratings on a 5-point Likert scale. (**b**) Distribution of immersion ratings on a 5-point Likert scale.

The immersion histogram is shown in Figure 11b and similarly peaks at 4, with the next most frequent score at 3. Very few participants rated immersion below 3, and fewer still at the maximum score of 5. The mean immersion score was 3.55 (SD 0.97), suggesting a moderate to high presence in the environment.

Figure 12 shows a side-by-side box plot, highlighting comparable medians (4 for satisfaction and 4 for immersion), as well as similar interquartile ranges. Satisfaction shows a slightly higher upper whisker (up to 5) compared to immersion, indicating that high satisfaction ratings were more common than top immersion ratings. Both distributions have minimal outliers.

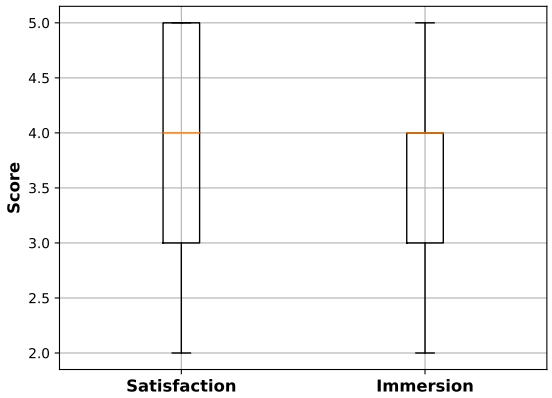


Figure 12. Box plot comparison of satisfaction and immersion scores, showing median, interquartile range, and whiskers. The two metrics demonstrate a similar central tendency and variability, with satisfaction exhibiting a slightly higher maximum.

4. Discussion

Section 3 demonstrated *Planogen*'s capacity to procedurally generate realistic airplane cabins and diverse passenger populations. Its flexibility, modularity, and Unity compatibility make it well suited not only for clinical scenarios but also for applications in education, training, behavioral research, and human–computer interaction studies. *Planogen* offers a simple and efficient user interface to create immersive scenarios, allowing researchers from various disciplines to utilize it effectively. In comparison, other tools that were presented in Section 1 did not appear to offer the same degree of module extensibility, integration into existing projects, or ease of use. For a detailed comparison of *Planogen* with other VR scenario creation tools, consult Table 1. Below, we outline a range of practical use cases and ongoing development efforts for *Planogen*.

Table 1. Comparative analysis of VR scenario creation tools.

	Planogen	**VR Nuggets [5]**	**VREUD [3]**	**VRScenarioBuilder [6]**	**GPT-API Synthesis [7]**
Main output	Plane interior scene	Educational scenarios	Interactive VR scenes	Driving scenarios	Interactive VR scenes
Supports characters	Yes	No	No	No	No
Modular	Yes	Yes	Limited	Yes	No
Narrative creation	No	Limited, template-based	No	Yes (scripted events)	No
UX	Simple dialogs	Visual node-based editor	Wizard-style interface	Scenario-embedded interface	Simple prompt-based
Requires internet	No	No	Yes	No	Yes
Platform-locked	Yes	Yes	Yes	Yes	Yes
Integration into existing projects	Yes, import package	Yes, packaged templates	Yes, packaged output	No, self-enclosed package	Limited, requires scene setup
Extensible	Yes	Partially, via code	Limited	Yes, extensible blocks/scripts	Partially, larger model library
Reusable	Yes	Yes	Yes	Partially	Partially, manual tagging of models

4.1. Practical Applications

Planogen was originally inspired by exposure therapy for fear of flying (FoF), but its utility extends far beyond clinical use. By abstracting away the complexity of scene construction, it enables both technical and non-technical users to produce detailed, interactive aircraft environments for a wide variety of immersive experiences.

4.1.1. Clinical Use: Exposure Therapy for FoF

Fear of flying remains a highly prevalent condition, affecting up to 40% of the population [19,20]. Virtual exposure therapy has shown promise in reducing travel-related anxiety, and *Planogen* offers a novel way to configure graded exposure environments that align with individual symptom profiles. Users can customize the virtual scene in real time, adjusting the crowd density, noise levels, turbulence, and seating arrangements, without requiring Unity programming expertise. This supports the creation of tailored, repeatable scenarios that improve the ecological validity and therapeutic control [21].

Planned features such as AI-driven character behaviors and real-time physiological feedback will help to simulate unpredictable or social stressors (e.g., a disruptive passenger or sudden announcements), which are critical in replicating the dynamic conditions of real-world air travel.

4.1.2. Behavioral Research and HCI Studies

Beyond therapy, *Planogen* is valuable in studying behavioral responses in constrained or high-density environments. Researchers in human–computer interaction, cognitive science, or psychology can use the tool to simulate realistic crowding, interpersonal distance violations, or noise-induced stress. Unlike prebuilt simulations, *Planogen* allows for the controlled manipulation of the cabin layout, character proximity, and demographic variety, all within a reproducible and programmable framework. This makes it well suited for A/B testing, usability studies, and multimodal interaction experiments in high-fidelity VR.

4.1.3. Training and Educational Use

Planogen's procedurally generated scenes also support training use cases. Aviation professionals, flight attendants, and emergency response teams could use it to simulate in-flight scenarios that require communication, crowd management, or de-escalation skills. In educational contexts, instructors can use it to teach human factors and team coordination, or even cultural dynamics in confined public spaces. Because the system supports both randomization and scripting, it can be adapted to different pedagogical needs and VR curricula.

4.2. Limitations and Planned Enhancements

At the time of writing, *Planogen* has not undergone formal user evaluation. While internal testing confirms the tool's functionality and modularity, empirical validation is forthcoming. The reported user study represents an initial pilot study designed to assess the realism, usability, and user experience of the generated VR cabin environments, rather than to clinically validate the system for fear of flying (FoF) treatment. The study sample did not specifically include individuals with FoF, nor was it intended to do so at this stage. We recognize that assessing *Planogen*'s effectiveness in therapeutic applications will require targeted studies with larger samples and appropriate clinical populations. To address this, we are planning further pilot and controlled studies that will recruit participants with FoF in collaboration with clinical partners. These future studies will evaluate both the therapeutic efficacy and the tool's usability among non-technical users such as clinicians and therapists.

Additionally, while *Planogen* was designed to lower the barrier to entry for non-technical users by providing a simple editor-based interface and modular configuration process, the present study did not conduct a formal usability analysis or collect systematic user feedback from this target group. We acknowledge this as an important area for future work. Planned studies will include structured usability testing with clinicians and researchers with varying levels of technical experience, in order to evaluate and further

refine the tool's accessibility and ease of use. Presently, users must configure scenes through Unity's Inspector, which poses a barrier to non-programmers. To address this, we plan to release a user-facing Unity extension with preset templates, intuitive drop-down menus, and scenario logic options. This will reduce the friction for educators, therapists, or researchers who may not be familiar with game engine workflows.

Currently, the tool's character generator offers limited diversity, and the scenarios are relatively static. Upcoming development will introduce greater control over passenger appearances (e.g., age, ethnicity, expression, body type) and behaviors (e.g., motion, emotional state), as well as more detailed environmental triggers (e.g., turbulence onset, lighting changes, announcement cues). In parallel, future evaluation efforts will incorporate more rigorous experimental designs, including controlled comparisons across system versions and expanded participant sampling, to further validate *Planogen*'s impact on the user experience and application outcomes. Additionally, the current version of Planogen does not yet implement formal safeguards to prevent non-technical users from inadvertently generating VR scenes that may be unsafe or clinically inappropriate. However, the scenarios currently supported by the tool are limited to static airplane cabin configurations with minimally interactive passenger avatars and constrained movement, such that no unsafe or hazardous configurations are presently possible. As more elaborate scenarios and user-defined configurations are introduced in future versions of the tool, we will prioritize the development of additional safety mechanisms, including automated validation checks, restricted parameter ranges, and template-based authoring modes, to ensure that the generated VR scenes remain appropriate and safe for clinical and therapeutic use.

Finally, although *Planogen* runs within Unity and supports VR, further testing is needed to evaluate its performance and usability across devices (e.g., Meta Quest, HTC Vive). We also intend to expand its compatibility with biometric sensors, speech-based interfaces, and AI-driven virtual agents to enhance scenario realism.

4.3. Outlook

By reducing the technical overhead of VR content creation, *Planogen* empowers a range of users to prototype, customize, and deploy immersive simulations. *Planogen*'s procedural generation pipeline supports both precision and variability, which makes it ideal for controlled studies, scalable training, and adaptive therapy.

As development continues, we aim to position *Planogen* not only as a therapeutic VR authoring tool but as a general-purpose, research-ready simulation engine for social, cognitive, and behavioral sciences. *Planogen*'s modular architecture invites future extensions to aircraft-adjacent environments such as trains, buses, or waiting areas, further broadening its relevance for real-world stress, safety, and training scenarios.

5. Conclusions

Planogen fills a crucial need for an accessible, high-fidelity VR scenario generator that empowers non-technical users to create and populate virtual airplane cabins without writing a single line of code. By combining a modular Unity plug-in architecture with procedural content techniques, *Planogen* allows researchers, clinicians, and educators to instantly configure cabin layouts, seat densities, and passenger demographics. We validated these claims in a user study of 33 participants, who reported a mean satisfaction score of 3.88/5 and a mean immersion rating of 3.55/5, demonstrating that *Planogen* produces believable and engaging environments.

Beyond lowering development barriers, *Planogen* promotes reproducibility and customization in VR content creation. Future work will expand its scenario diversity, streamline the user interface, and integrate clinical tool compatibility (e.g., biometric sensors and

AI-driven virtual assistants). Future work will also focus on expanding the scope and rigor of our experimental evaluations to further validate *Planogen*'s capabilities across diverse research and clinical applications.

Author Contributions: Conceptualization, O.S., L.R.K. and K.T.; methodology, L.R.K. and O.S.; software, L.R.K. and R.D.; validation, R.D.; formal analysis, K.T. and O.S.; resources, O.S. and R.D.; writing—original draft preparation, K.T. and L.R.K.; writing—review and editing, O.S. and K.T.; visualization, O.S., L.R.K. and K.T.; supervision, O.S.; project administration, O.S.; funding acquisition, O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the James H. Brickley Endowment for Faculty Professional Development and Innovation Grant (No. 003798) and the eFellows Classroom Technology Grant (No. 003594) at Eastern Michigan University.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board of Eastern Michigan University (protocol code UHSRC-FY24-25-201. Date of approval: 27 March 2025).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FoF	Fear of Flying
HCI	Human–Computer Interaction
VR	Virtual Reality
PCG	Procedural Content Generation
LLM	Large Language Model

Appendix A. Conceptualization and Execution Flowchart

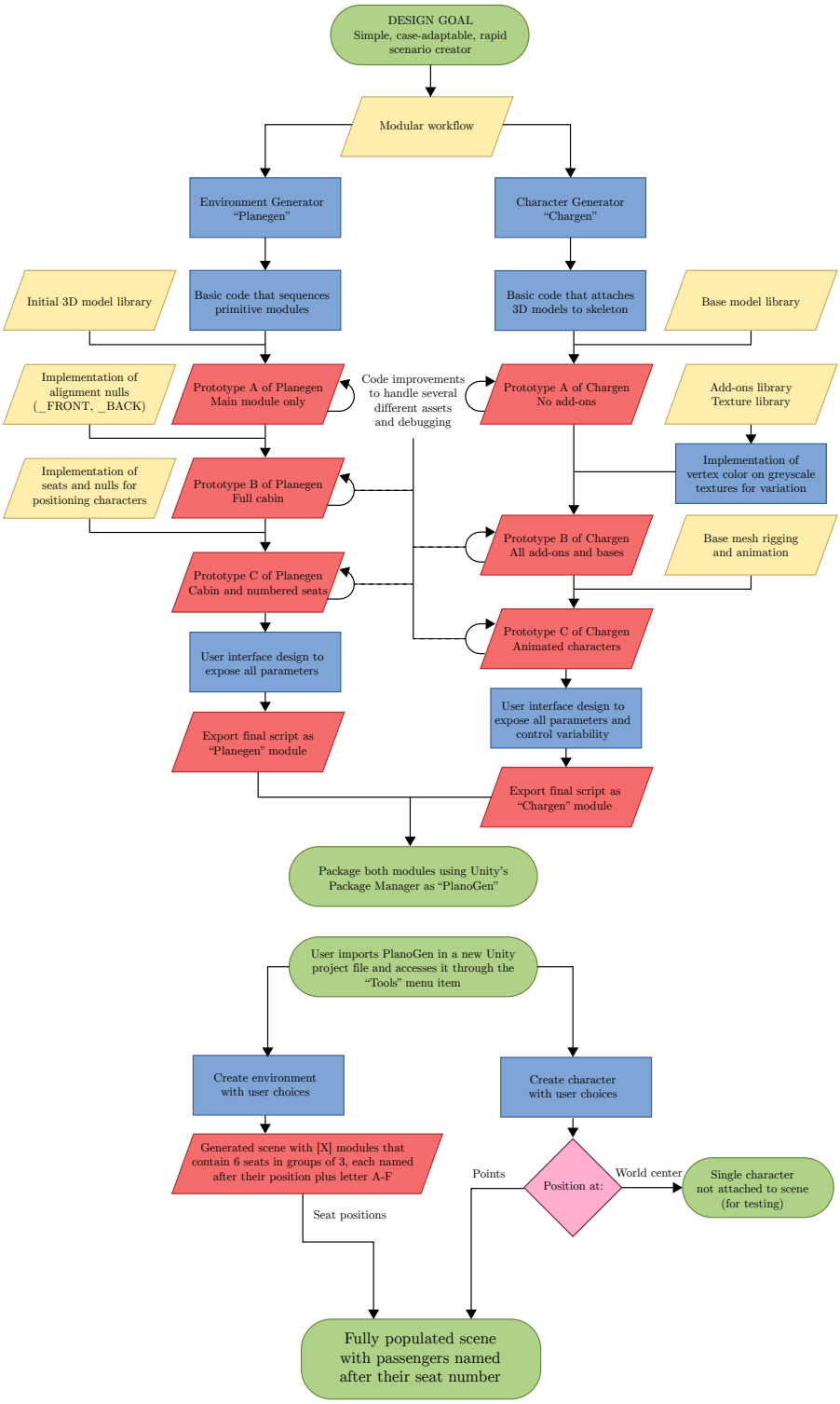


Figure A1. *Planogen*'s conceptualization process and user scene creation flowchart.

References

1. Ashtari, N.; Bunt, A.; McGrenere, J.; Nebeling, M.; Chilana, P.K. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; ACM: New York, NY, USA, 2020; pp. 1–13. [CrossRef]

2. Krauß, V.; Boden, A.; Oppermann, L.; Reiners, R. Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Virtual, 8–13 May 2021; ACM: New York, NY, USA, 2021; pp. 1–15. [CrossRef]
3. Yigitbas, E.; Klauke, J.; Gottschalk, S.; Engels, G. VREUD—An End-User Development Tool to Simplify the Creation of Interactive VR Scenes. In Proceedings of the 2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), St. Louis, MO, USA, 10–13 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–10. [CrossRef]
4. Evain, A.; Fertier, A.; Panzoli, D.; Kropczynski, J.; Halse, S.; Benaben, F. Authoring Training Scenarios "from within" in an Immersive Environment with the CESTATE Framework. *Interact. Learn. Environ.* **2024**, *33*, 2070–2084.
5. Horst, R.; Naraghi-Taghi-Off, R.; Rau, L.; Doerner, R. Authoring with Virtual Reality Nuggets—Lessons Learned. *Front. Virtual Real.* **2022**, *3*, 840729 . [CrossRef]
6. Eroglu, S.; Voigt, A.; Weyers, B.; Kuhlen, T.W. VRScenarioBuilder: Free-Hand Immersive Authoring Tool for Scenario-Based Testing of Automated Vehicles. In Proceedings of the 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Orlando, FL, USA, 16–21 March 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 196–202. [CrossRef]
7. Kouzelis, L.R.; Spantidi, O. Synthesizing Play-Ready VR Scenes with Natural Language Prompts Through GPT API. In *Advances in Visual Computing*; Bebis, G., Ghiasi, G., Fang, Y., Sharf, A., Dong, Y., Weaver, C., Leo, Z., LaViola, J.J., Jr., Kohli, L., Eds.; Springer: Cham, Switzerland, 2023; pp. 15–26. [CrossRef]
8. Lang, Y.; Wei, L.; Xu, F.; Zhao, Y.; Yu, L.F. Synthesizing Personalized Training Programs for Improving Driving Habits via Virtual Reality. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany, 18–22 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 297–304. [CrossRef]
9. Arbesser-Rastburg, G.; Safikhani, S.; Gustin, M.; Hopfe, C.; Schweiger, G.; Pirker, J. Project Beyond: An Escape Room Game in Virtual Reality to Teach Building Energy Simulations. *arXiv* **2024**, arXiv:2407.02981. [CrossRef]
10. Van Paridon, K.; Timmis, M.A.; Sadeghi Esfahlani, S. Development and Evaluation of a Virtual Environment to Assess Cycling Hazard Perception Skills. *Sensors* **2021**, *21*, 5499. [CrossRef] [PubMed]
11. Kern, F.; Winter, C.; Gall, D.; Kathner, I.; Pauli, P.; Latoschik, M.E. Immersive Virtual Reality and Gamification Within Procedurally Generated Environments to Increase Motivation During Gait Rehabilitation. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 500–509. [CrossRef]
12. Thandapani, R.K.G.R.; Capel, B.; Lasnier, A.; Chatzigiannakis, I. INTERACT: An Authoring Tool That Facilitates the Creation of Human Centric Interaction with 3d Objects in Virtual Reality. In Proceedings of the 25th International Conference on Mobile Human-Computer Interaction, Sharm El-Sheikh, Egypt, 22–25 September 2023; ACM: New York, NY, USA, 2023; pp. 1–5. [CrossRef]
13. Andújar, C.; Chica, A.; Fairén, M.; García, O.; Nieto, J.; Tortosa, S.; Insa-Calderón, E. A Highly-Configurable Session Designer for VR Nursing Training. *Heliyon* **2024**, *10*, e39692. [CrossRef] [PubMed]
14. Chard, I.; Van Zalk, N. Virtual Reality Exposure Therapy for Treating Social Anxiety: A Scoping Review of Treatment Designs and Adaptation to Stuttering. *Front. Digit. Health* **2022**, *4*, 842460. [CrossRef] [PubMed]
15. Sobociński, P.; Flotyński, J.; Śliwicki, M.; Maik, M.; Walczak, K. Knowledge-Based Creation of Industrial VR Training Scenarios. In Proceedings of the Communication Papers of the 18th Conference on Computer Science and Intelligence Systems, Warsaw, Poland, 17–20 September 2023; pp. 271–278. [CrossRef]
16. Podkosova, I.; Reisinger, J.; Kaufmann, H.; Kovacic, I. BIMFlexi-VR: A Virtual Reality Framework for Early-Stage Collaboration in Flexible Industrial Building Design. *Front. Virtual Real.* **2022**, *3*, 782169. [CrossRef]
17. Karre, S.A.; Reddy, Y.R. Model-Based Approach for Specifying Requirements of Virtual Reality Software Products. *Front. Virtual Real.* **2024**, *5*, 1471579. [CrossRef]
18. Stemasov, E.; Demharter, S.; Rädler, M.; Gugenheimer, J.; Rukzio, E. pARam: Leveraging Parametric Design in Extended Reality to Support the Personalization of Artifacts for Personal Fabrication. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–22. [CrossRef]
19. Oakes, M.; Bor, R. The psychology of fear of flying (part I): A critical evaluation of current perspectives on the nature, prevalence and etiology of fear of flying. *Travel Med. Infect. Dis.* **2010**, *8*, 327–338. [CrossRef] [PubMed]
20. Cloitre, M.; Koenen, K.C.; Cohen, L.R.; Han, H. Skills training in affective and interpersonal regulation followed by exposure: A phase-based treatment for PTSD related to childhood abuse. *J. Consult. Clin. Psychol.* **2002**, *70*, 1067–1074. [CrossRef] [PubMed]
21. McLean, C.P.; Levy, H.C.; Miller, M.L.; Tolin, D.F. Exposure therapy for PTSD: A meta-analysis. *Clin. Psychol. Rev.* **2022**, *91*, 102115. [CrossRef] [PubMed]