The Journal of Open Source Software

unifir: A Unifying API for Working with Unity in R

Michael J. Mahoney[1], **Colin M. Beier**[2], **and Aidan C. Ackerman**[3]

1 Graduate Program in Environmental Science, State University of New York College of Environmental Science and Forestry, Syracuse, New York, USA **2** Department of Sustainable Resources Management, State University of New York College of Environmental Science and Forestry, Syracuse, New York, USA **3** Department of Landscape Architecture, State University of New York College of Environmental Science and Forestry, Syracuse, New York, USA

DOI: 10.21105/joss.04388

Software

- Review
- Repository
- Archive

Editor: Kyle Niemeyer

Reviewers:

- @kyleniemeyer

Submitted: 05 May 2022
Published: 11 May 2022

Summary

unifir is an R package representing a first step towards an open-source, fully reproducible approach for constructing immersive virtual environments (IVEs). Users are able to construct IVEs from real or simulated spatial data by writing simple, idiomatic R code, allowing the production of IVEs without needing users to be highly familiar with the complicated video game engines used to power these environments. unifir also provides a simple set of classes which are easily extended, enabling future efforts to provide users with a consistent API for IVE design while relying upon unifir as a low-level interface to the Unity game engine and extending its capabilities to new data formats or engine capabilities.

Statement of Need

Immersive virtual environments (IVEs) are an exciting area of research with potential experimental applications in fields varying from science communication (Huang, Lucash, Scheller, & Klippel, 2021), landscape planning (Swetnam & Korenko, 2019), environmental economics (Fiore, Harrison, Hughes, & Rutström, 2009) and beyond. By creating realistically-rendered immersive representations, researchers are able to induce a sense of presence (or "being there") in their audiences (Slater & Usoh, 1993), which may serve to make experiments in IVEs more similar to "real-life" experiences than can normally be obtained through controlled experiments in laboratory settings (Schöne, Wessels, & Gruber, 2019). However, the difficulty of producing IVEs has limited their use, as their creation relies upon a number of software tools that are not commonly used in scientific practice. These tools are often managed through graphical user interfaces, making it difficult to capture the production process in a way that would allow others to reproduce IVEs or replicate IVE-driven experiments. Finally, it is also difficult for researchers to incorporate spatial data into IVEs using these tools in order to produce virtual representations of real-world places (Keil, Edler, Schmitt, & Dickmann, 2021).

The core gap underpinning all of these challenges is a lack of accessible tooling that provides a standardized, programmatic method for creating IVEs from real or simulated data. Some first efforts have been made in this direction; for instance, Keil et al. (2021) surveyed methods for incorporating spatial data representing both terrain and buildings into IVEs, while Mahoney et al. (2022) introduced an open-source toolkit for programmatically incorporating terrain data into IVEs. Further work is needed to expand the amount of IVE production that can be done reproducibly using standard scientific toolkits and to extend these tools to a wider variety of commonly available data types.

The unifir R package addresses these needs by providing an easily-extended toolkit for producing IVEs programmatically. Users are able to construct their IVEs through idiomatic R code, which then is translated into C# and executed within the Unity video game engine in order to produce

IVEs in a fast, reproducible manner. By encoding all of the decisions involved in building a scene in standard R and C# code, unifir hopes to improve the openness and interoperability of IVE development and make these powerful tools more accessible across a broad swath of research domains.

Package Overview

unifir is an open-source toolkit for producing IVEs within the Unity video game engine from simple R programs. To do this, it provides users with methods for creating and manipulating "scripts" (objects of the class `unifir_script`), which are sequential lists of instructions and dependencies for producing IVEs. These "scripts" are themselves composed of "props" (objects of the class `unifir_prop`), which are individual self-contained instructions for how to accomplish some outcome, such as adding 3D models or cameras to the IVE (potentially specifying their positions using spatial data), creating terrain surfaces from standard raster formats, or changing what sets of objects are displayed on opening the Unity engine. Additional functions provide a small set of permissively-licensed 3D models that can be easily incorporated into the produced IVEs. Users sequentially add props to their scripts, with the order props are added to the script determining the order they'll be executed upon IVE construction. Because unifir allows users to write scripts entirely in R, while Unity is only capable of executing C#, these props both provide a method for users to provide inputs in R and specify how those inputs will be translated into a C# program. Upon the user executing a script, unifir translates each prop with its inputs into C# code, opens or creates a Unity project, and executes the resulting C# script inside of the project to produce an IVE.

Due to this straightforward structure, extending unifir is relatively easy. New props only need to provide an interface for accepting user input and instructions for translating that input into C# code, and otherwise can rely upon unifir for all interaction with the Unity engine. Methods for constructing new prop functions are included as part of the package. Advanced methods also allow props to use reflective programming and manipulate other props attached to the same script, a feature used in the base package to improve the efficiency of generated C# by eliminating redundant processes. The intention is for unifir to provide a unifying framework for interacting with Unity from R, providing a standard API for a growing suite of tools for producing IVEs in an open and reproducible manner.

Acknowledgements

This work was supported by the State University of New York via the ESF Pathways to Net Zero Carbon initiative.

References

Fiore, S. M., Harrison, G. W., Hughes, C. E., & Rutström, E. E. (2009). Virtual experiments and environmental policy. *Journal of Environmental Economics and Management*, *57*(1), 65–86. doi:10.1016/j.jeem.2008.08.002

Huang, J., Lucash, M. S., Scheller, R. M., & Klippel, A. (2021). Walking through the forests of the future: Using data-driven virtual reality to visualize forests under climate change. *International Journal of Geographical Information Science*, *35*(6), 1155–1178. doi:10.1080/13658816.2020.1830997

Keil, J., Edler, D., Schmitt, T., & Dickmann, F. (2021). Creating immersive virtual environments based on open geospatial data and game engines. *KN Journal of Cartography and Geographic Information*, *71*, 53–65. doi:10.1007/s42489-020-00069-6

Mahoney, M. J., Beier, C. M., & Ackerman, A. C. (2022). terrainr: An r package for creating immersive virtual environments. *Journal of Open Source Software*, *7*(69), 4060. doi:10.21105/joss.04060

Schöne, B., Wessels, M., & Gruber, T. (2019). Experiences in virtual reality: A window to autobiographical memory. *Current Psychology*, *38*, 715–719. doi:10.1007/s12144-017-9648-y

Slater, M., & Usoh, M. (1993). Presence in immersive virtual environments. *Proceedings of IEEE virtual reality annual international symposium* (pp. 90–96). doi:10.1109/VRAIS.1993.380793

Swetnam, R. D., & Korenko, J. (2019). Can computer game landscapes target new audiences for landscape quality assessment? *Applied Geography*, *113*, 102102. doi:10.1016/j.apgeog.2019.102102