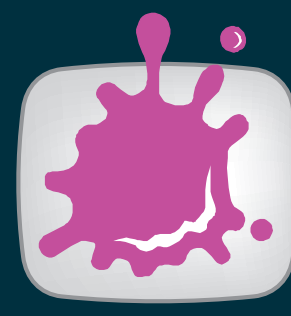


Live Coding of a VR Render Engine in VR

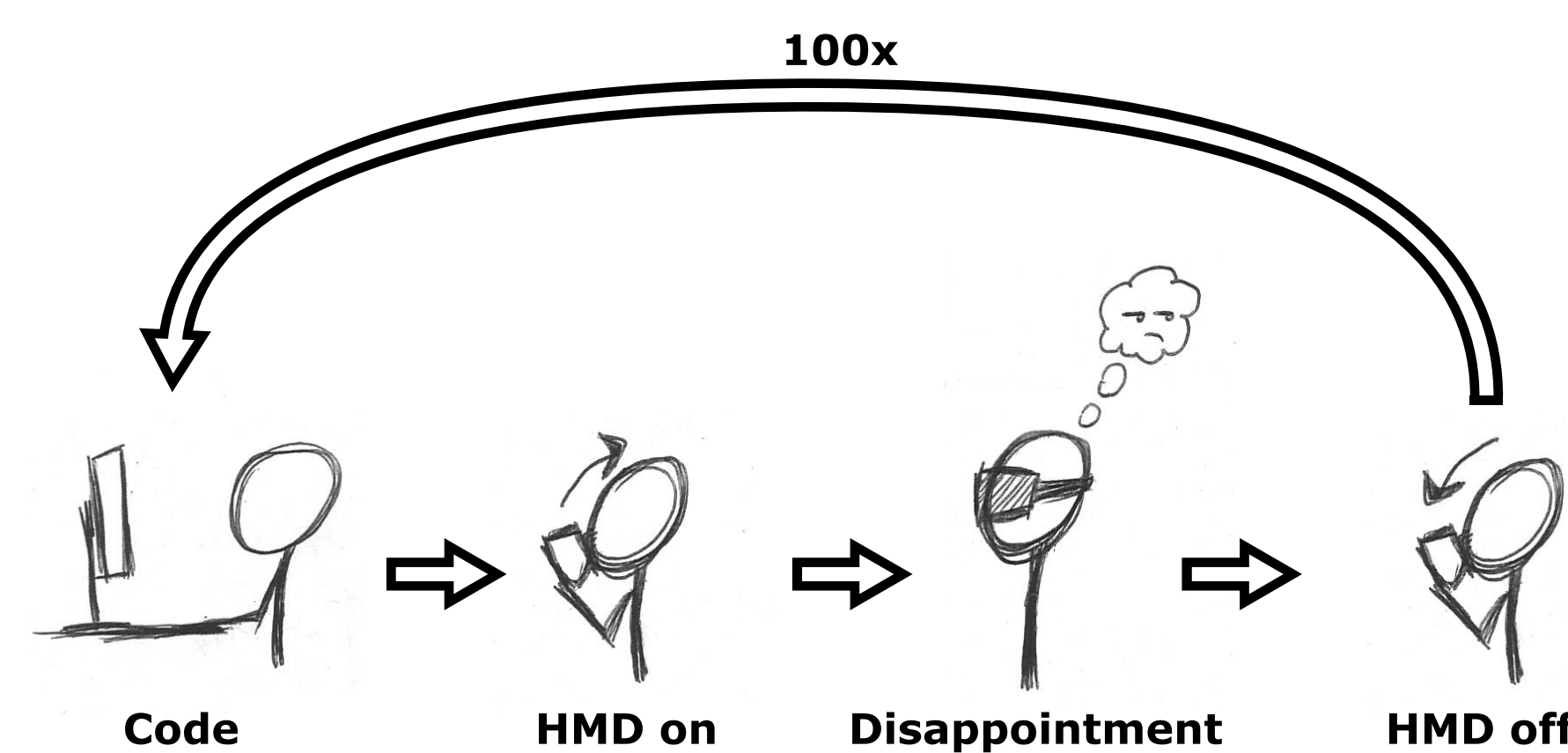


FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics

Markus Schütz, Michael Wimmer
TU Wien,
Institute of Visual Computing & Human-Centered Technology

Abstract

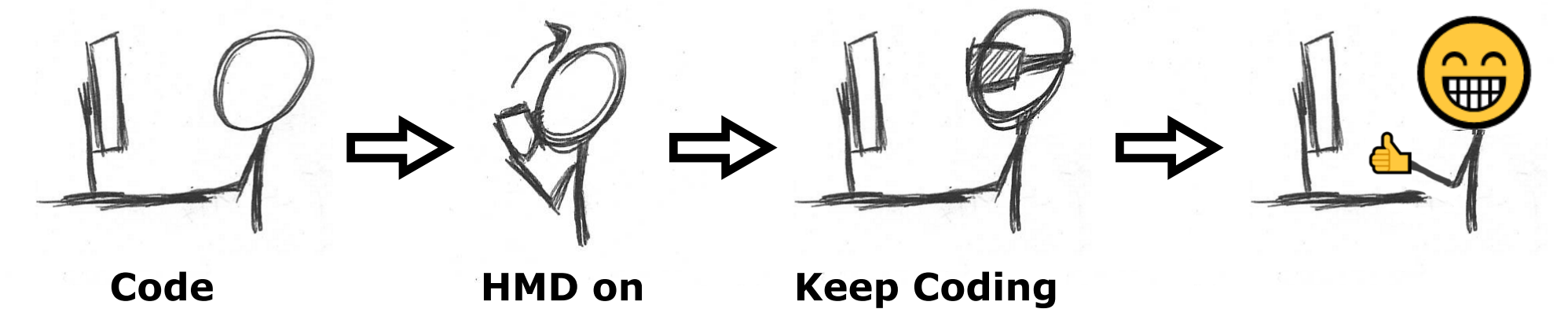
Live coding in virtual reality allows users to create and modify their surroundings through code without the need to leave the virtual reality environment. Previous work focuses on modifying the scene at runtime [1, 2, 3]. We propose an application that allows developers to modify virtually everything at runtime, including the scene but also the render engine, shader code and input handling, using standard desktop IDEs through a desktop mirror.



Common VR Dev Workflow

VR and desktop experiences are very different, and many iterations are necessary to get things right. Switching between desktop and VR is cumbersome and time consuming.

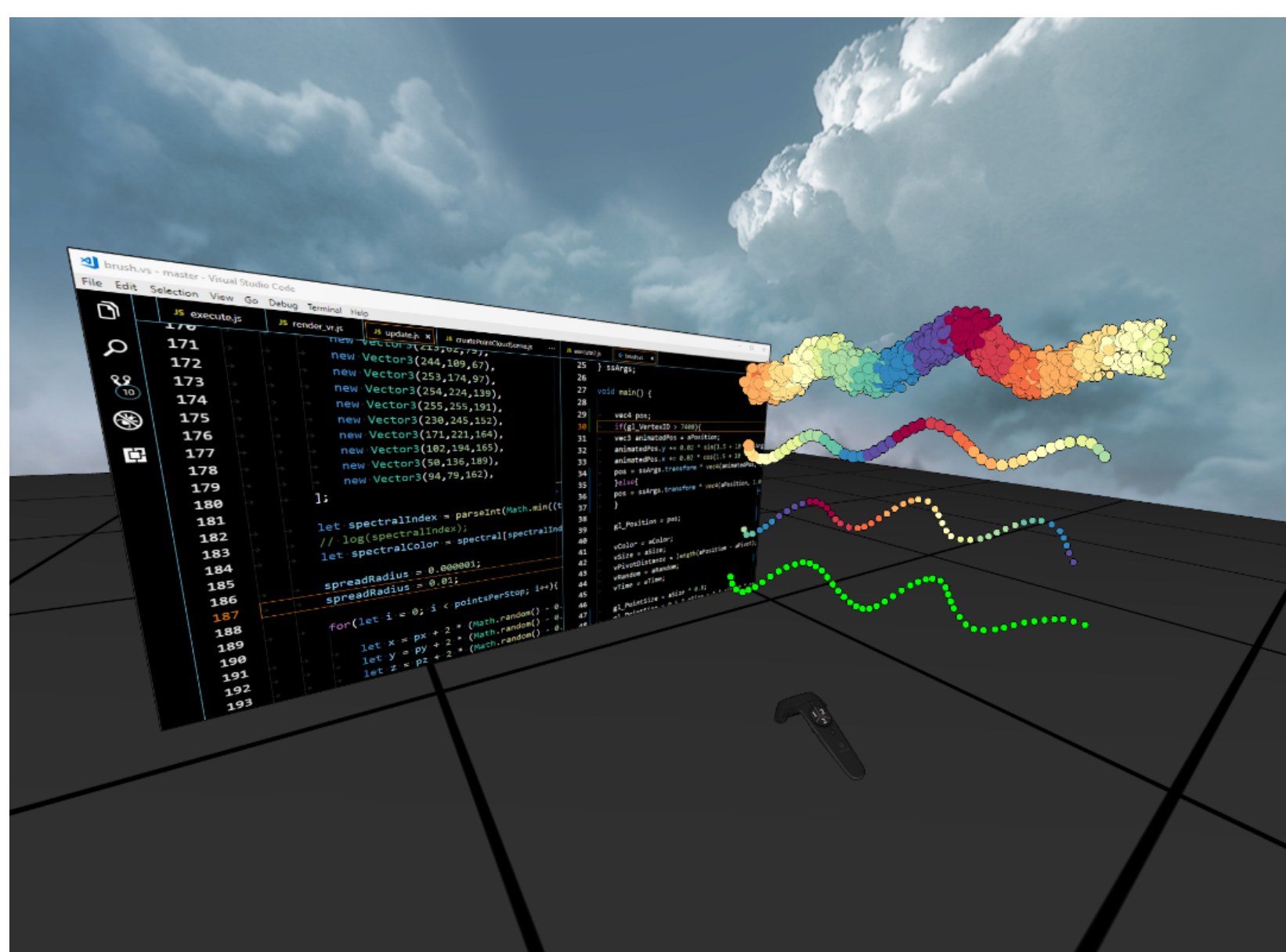
Problem Statement



Our VR Live Coding Workflow

Basic development on desktop, and tweaking and polishing in VR to improve the VR experience directly in VR, without the need to take off the HMD or restart the application.

Use Cases



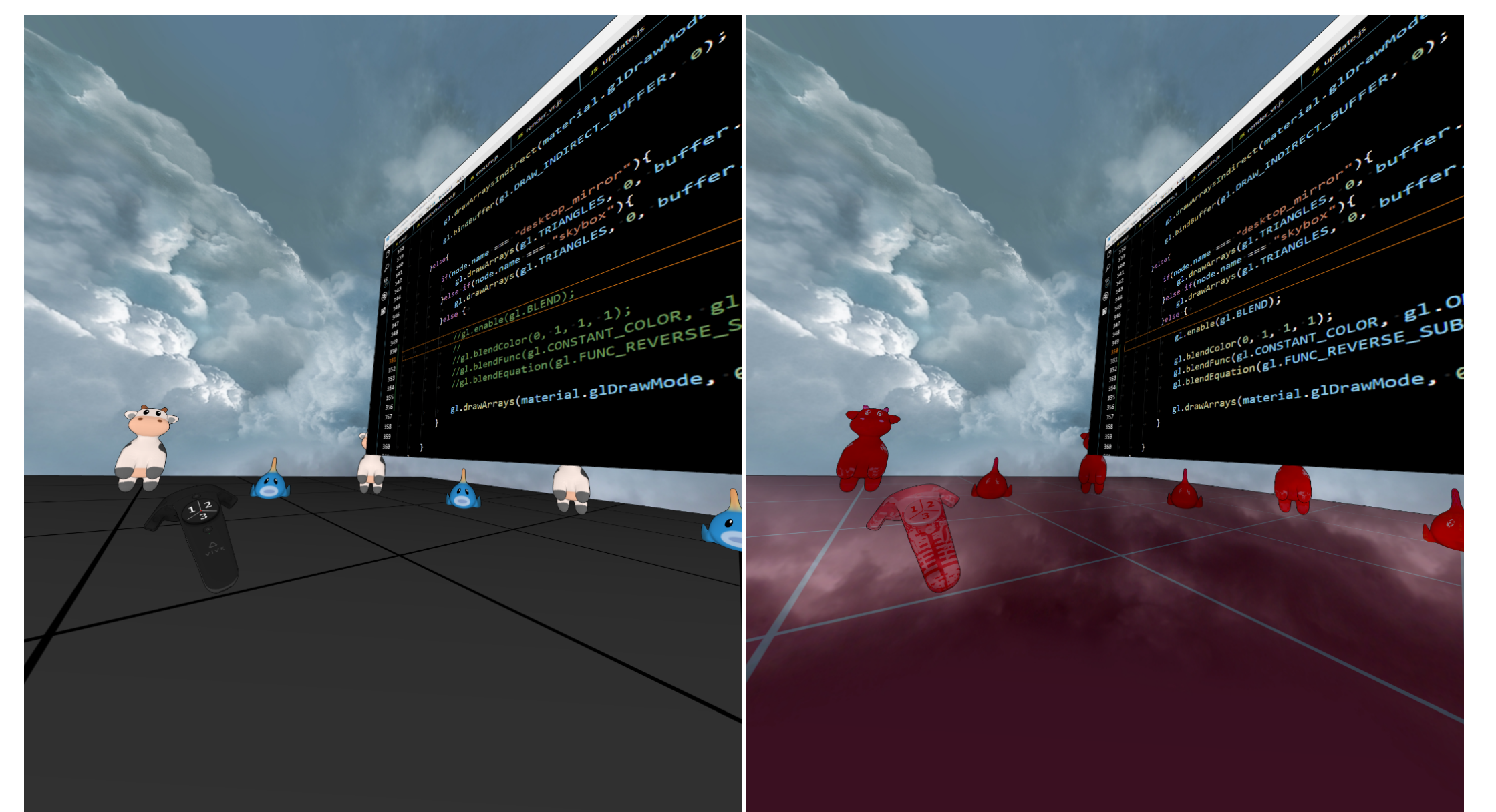
Tweaking Brushes

Modify brush generation and animation at runtime. Experiment with colors, size, spread and animate the vertices.



Drag & Drop Scene Nodes

Pick objects in the scene and drop them in the source code to generate a selector. Then write code that affects the selected object.



Modify OpenGL Draw Calls

Change render logic and draw calls and their parameters at runtime. Experiment with parameters, fix bugs, adjust input to shaders without restart.

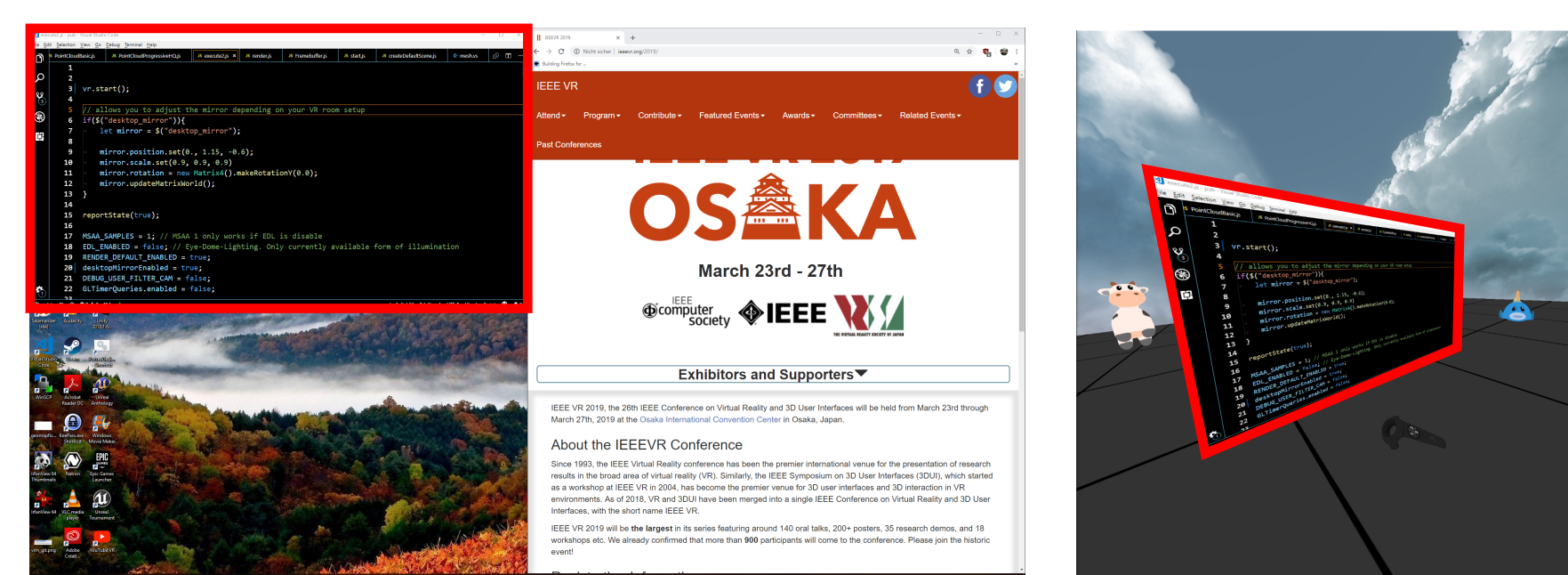
Implementation

This framework is implemented in C++ using OpenGL 4.5, OpenVR and the V8 JavaScript engine. The entire rendering engine is written in JavaScript through JS bindings to the OpenGL and OpenVR APIs. The scripts can be modified at runtime by editing and saving the respective JavaScript files. Saving (ctrl+s) immediately replaces old code with the new code at runtime. DirectX is used to retrieve a texture of the desktop and then pass it to OpenGL to use for the desktop mirror.



Desktop Mirror

The desktop mirror shows part of the desktop monitor inside the VR environment, which allows developers to use any IDE of their choice to code in VR, and even use browsers to look up documentation or find solutions without leaving the VR environment.

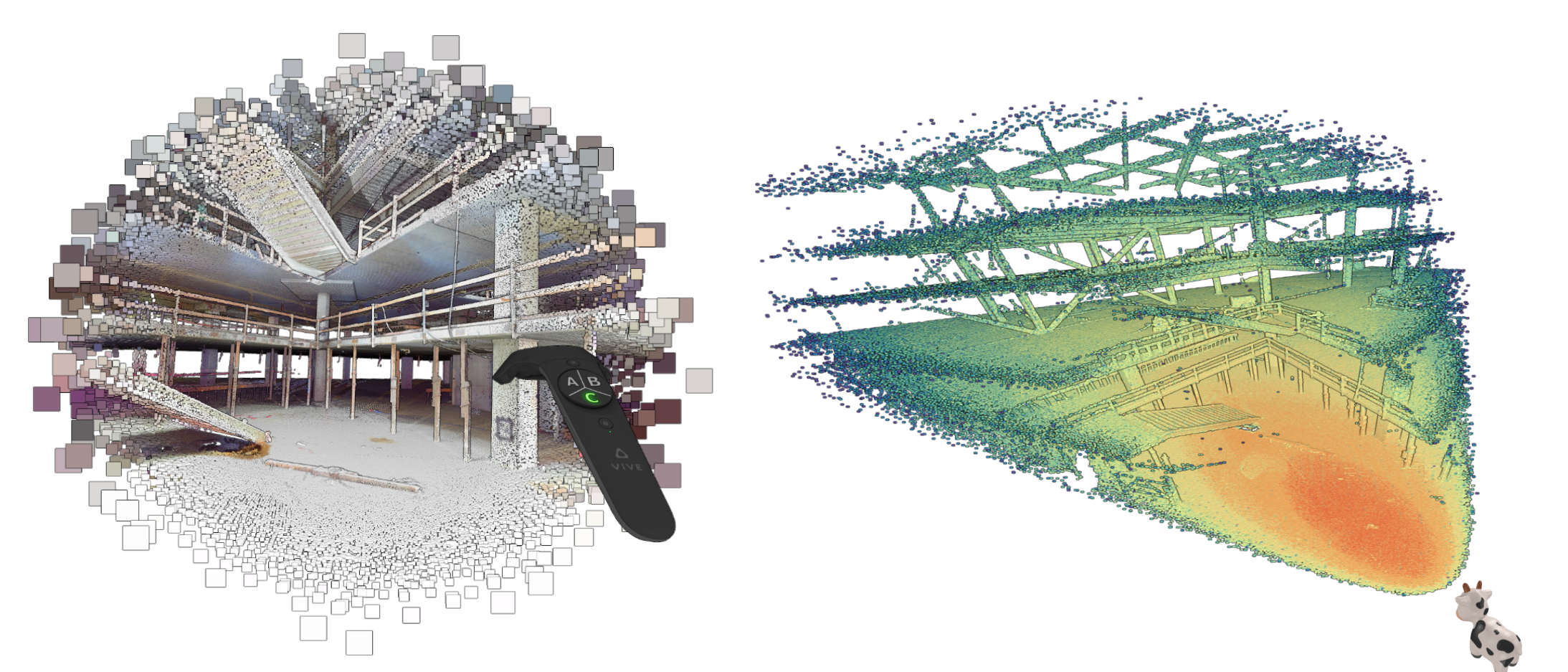


(a) Desktop

(b) VR

Results

This live coding framework was developed to aid us in the development of “Real-Time Continuous Level of Detail Rendering of Point Clouds” [4], which is also accepted as a paper at IEEE VR 2019. We used it extensively to prototype, tweak and polish our point-cloud rendering algorithm, both in desktop and VR mode. In VR mode it was especially useful to us to work on issues that are not apparent in desktop mode, such as aliasing artifacts and flickering issues that look fine on desktop but pose a problem in VR.



The source code is available at:
<https://github.com/m-schuetz/Fenek>

Limitations & Future Work

- See-through is not yet supported, so developers need to be able to use their keyboard blindly. We will evaluate the usage of the VIVE cameras or third-party cameras as part of future work.
- This framework is intended to enhance regular coding workflows but not completely replace them. Some changes are easier done on desktop, others are easier applied by restarting rather than changing the running state, and many changes are easier done by modifying code directly at runtime in VR.

Acknowledgements

This research was enabled by the Doctoral College Computational Design (DCCD) of the Center for Geometry and Computational Design (GCD) at TU Wien.

References

- [1] L. Iannini. Rumpus: A livecoding playground for room-scale vr. <http://rumpus.land/>.
- [2] R. Krahn. Codechisel3d: Live programming with three.js and webvr. <https://robert.krahn.net/past-projects/live-programming-with-threeand-webvr.html>.
- [3] B. Peiris. Riftsketch: An html5 live-coding environment based on webvr. <https://github.com/brianpeiris/RiftSketch>.
- [4] M. Schütz, K. Krösl, M. Wimmer. Real-Time Continuous Level of Detail Rendering of Point Clouds, In Proceedings of IEEE VR 2019