Introduction

During the second week of exploration, I gradually turned my focus to ‘crashes’ and ‘errors’ in the TouchDesigner software. As a real-time multimedia authoring tool, TouchDesigner strives to provide a smooth and logical workflow that enables users to create dynamic and interactive multimedia works. However, during the first week of reproducing animations, I noticed that the software crashed and lagged when loading too large a file. This experience prompted me to think about whether the software's ‘crashes’ and ‘bugs’ could be seen as a starting point for creation. I chose the *Conditional Design Manifesto* as an analytical tool to revisit my approach and thinking through its core concepts.

Application of theory

Conditional Design Manifesto emphasises that design should be rule-orientated, generating a dynamic process through the setting of conditions, rather than pursuing a defined end result. The setting of rules provides clear boundaries and structure to the design while allowing for unpredictability in the outcome. This approach pays particular attention to the act of designing itself and how the system generates new possibilities in response to conditions. In addition, the manifesto proposes the idea that ‘constraints inspire creativity,’ meaning that constraints are not obstacles to innovation but rather serve as motivators for experimentation and exploration.

These ideas have provided me with new perspectives on my project, allowing me to reconstruct the experimental framework and further deepen the goals of the project. For example, I have divided my experiments into three phases: triggering the crash, collecting the crash material, and integrating and editing the material. In the past, I

triggered crashes in a random way, but now I started to set more explicit rules, such as adjusting parameters to extreme values or overlaying complex components to actively test the boundary values of the software. Through the application of these rules, moments of crashes are no longer isolated phenomena, but the result of system behaviour. These experiments not only revealed the limitations of the software, but also led me to focus on the whole dynamic process: from the triggering conditions of the crash to the formation of its visual representation.

Deepening inquiry

Based on this, I tried to reintroduce the screenshots and videos of these crash moments into TouchDesigner and use its own functionality to conduct iterative experiments. In this way, the software seems to ‘observe’ its own limitations and presents this process of reflection in a visual form. The ‘bugs’ of the software are here redefined as a creative tool, further exploring its possibilities through rule-driven dynamic generation.

Conclusion

In addition, my project reflects the idea that ‘limitations inspire creativity’ ; TouchDesigner's crashes reveal its systemic fragility, but these limitations also create opportunities to explore the tension between control and chaos. By turning constraints into conditions for experimentation, my project breaks with the traditional view of software as a seamless tool and transforms TouchDesigner into a ‘participant’ in the creative process, a system that both generates and analyses its own visual output.
By applying the theory of Conditional Design Manifesto, I no longer see crashes as simple technical failures, but rather as dynamically generated results of design rules.

This perspective not only helped me dig deeper into the potential and limitations of the TouchDesigner system, but also brought new directions for the project to explore. Next, I plan to further refine the rule set, experiment with different iteration patterns, and observe how the rules shape the behaviour and visualisation of crashes.

Reference

Maurer, L., Paulus, E., Puckey, J. and Wouters, R. (2013) *Conditional Design Manifesto*. In: *Conditional Design Workbook*. Amsterdam: Valiz.