Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in New Media

# Designing and Developing an Educational VJ Tool

Inka Jerkku

Master's thesis
2024

Aalto-yliopisto
Aalto-universitetet
Aalto University

**Author** Inka Jerkku

**Title of thesis** Designing and Developing an Educational VJ Tool

**Programme** Master's Program in Art and Media

**Major** New Media

**Thesis supervisor** Antti Hietaniemi

**Thesis advisor(s)** Nuno Correia and Matti Niinimäki

**Date** 31.12.2024
**Number of pages** 65
**Language** English

**Abstract**

This research explores the design and development of a VJ (video jockeying) program for real-time visual creation and video editing, aiming to support skill development in educational contexts while functioning as a standalone live performance tool. The program was developed using TouchDesigner, utilizing visual programming to create an independent user interface distinct from the TouchDesigner environment.

The project was motivated by the need for live visuals software accessible to both beginners and experienced users which promotes learning through facilitating understanding of live visuals workflows, foundational principles, and node-based programming in TouchDesigner. Additionally, the program aimed to provide extensibility and customizability, and support users' progression to more advanced systems. Other key design objectives included versatility, practicality, a clear user interface, and compatibility across operating systems and hardware.

The methodology combined a literature review and a user study involving eight participants (beginners and experienced VJs). The design was informed by insights from the literature review, addressing the research question and project objectives. The user study involved interface testing, node-based structure editing inside TouchDesigner, and semi-structured interviews featuring evaluations of the features and user feedback.

Results from the user study were generally positive, particularly regarding learning and creativity, clarity and organization, feature variety, customizability, balance of control and constraints, suitability of the interface for diverse user levels, interface aesthetics, and overall user experience. However, participants, especially beginners, found node editing within TouchDesigner challenging. The research highlights areas for future improvement while emphasizing the program's strengths as a versatile tool for both novice and experienced VJ performers.

**Keywords** VJing, live visuals, VJ software, visual programming, educational software, educational design, TouchDesigner, performance tool

Aalto-yliopisto
Aalto-universitetet
Aalto University

**Tekijä** Inka Jerkku

**Työn nimi** Koulutuksellisen VJ-työkalun suunnittelu ja kehitys

**Koulutusohjelma** Taiteen maisteri

**Pääaine** New Media

**Vastuuopettaja/valvoja** Antti Hietaniemi

**Työn ohjaaja(t)** Nuno Correia ja Matti Niinimäki

**Päivämäärä** 31.12.2024
**Sivumäärä** 65
**Kieli** Englanti

**Tiivistelmä**

Tämä tutkimus käsittelee VJ-ohjelman (video jockeying) suunnittelua ja kehittämistä reaaliaikaiseen visuaalien luomiseen ja videoeditointiin. Ohjelman tavoitteena on tukea taitojen kehittämistä koulutuksellisissa konteksteissa ja toimia samalla itsenäisenä live-esitystyökaluna. VJ-ohjelma ja sen itsenäinen käyttöliittymä kehitettiin visuaalisen ohjelmoinnin avulla TouchDesigner-ohjelmointiympäristössä.

Projektin lähtökohtana oli tarve kehittää VJ-ohjelma, joka on saavutettavissa sekä aloittelijoille että kokeneille käyttäjille: ohjelman tarkoituksena oli edistää oppimista tukemalla ymmärrystä live-visuaalien rakentumisesta, periaatteista ja TouchDesignerin visuaalisesta ohjelmoinnista. Lisäksi ohjelman tavoitteena oli tarjota laajentamismahdollisuuksia ja muokattavuutta sekä tukea käyttäjien siirtymistä kehittyneempiin järjestelmiin. Muita keskeisiä suunnittelutavoitteita olivat monipuolisuus, käytännöllisyys, selkeä käyttöliittymä sekä yhteensopivuus eri käyttöjärjestelmien ja laitteistojen kanssa.

Tutkimusmenetelmiin kuului kirjallisuuskatsaus ja käyttäjätutkimus, johon osallistui kahdeksan henkilöä (aloittelijoita ja kokeneita VJ:tä). Ohjelman suunnittelu perustui kirjallisuuskatsauksesta saatuihin oivalluksiin, jotka vastasivat tutkimuskysymykseen ja projektin tavoitteisiin. Käyttäjätutkimukseen sisältyi käyttöliittymän testaamista, node-pohjaisen rakenteen muokkaamista TouchDesignerissa sekä puolistrukturoituja haastatteluja, joissa arvioitiin ohjelman ominaisuuksia ja kerättiin käyttäjäpalautetta.

Käyttäjätutkimuksen tulokset olivat yleisesti ottaen positiivisia erityisesti oppimisen ja luovuuden, selkeyden ja järjestelmällisyyden, ominaisuuksien monipuolisuuden, muokattavuuden, kontrollin ja rajoitusten tasapainon, käyttöliittymän sopivuuden eri taitotasoille, esteettisyyden sekä yleisen käyttökokemuksen osalta. Osallistujat, erityisesti aloittelijat, kokivat kuitenkin ohjelman muokkaamisen TouchDesignerin oman käyttöliittymän kautta haastavaksi. Tutkimus nosti esille monia tulevaisuuden kehityskohteita, mutta myös ohjelman vahvuuksia monipuolisena työkaluna sekä aloittelijoille että kokeneille VJ-esiintyjille.

**Avainsanat** VJ, live-visuaalit, VJ-ohjelmisto, visuaalinen ohjelmointi, opetusohjelmisto, pedagoginen suunnittelu, TouchDesigner, esitystyökalu

# Table of contents

# Glossary

**Generative**
A process or method in which visuals, sounds, or other artistic outputs are created algorithmically. Generative art often involves systems that produce ever-evolving content based on certain parameters or rules.

**GLSL Shader**
A shading language used in computer graphics to program the way 3D models are rendered, specifically used in OpenGL environments. GLSL shaders are used to create visual effects and enhance the rendering of scenes in real-time graphics applications.

**LFO (Low-Frequency Oscillator)**
A signal generator that produces periodic oscillations at a low frequency, often below the human hearing range. LFOs are used to modulate various parameters in audio and visual systems, such as pitch, volume, or visual movement.

**Midi Device**
A hardware device that sends and receives MIDI (Musical Instrument Digital Interface) messages to control audio and visual systems. Used in live performance setups to trigger sounds, effects, or visuals.

**Modulative**
Refers to the process of varying a signal, parameter, or effect over time, often in a rhythmic or dynamic manner. Modulation is commonly used in audio and visual synthesis to create evolving, changing outputs.

**NDI (Network Device Interface)**
A protocol developed by NewTek that allows for high-quality, low-latency video and audio streaming over a standard IP network. NDI is commonly used in live streaming and broadcast environments.

**OSC (Open Sound Control)**
A protocol used for communication between multimedia devices and software. OSC allows for the real-time transfer of data, including control messages for audio and visual systems, often used in live performance setups.

**Signifier**
In UX/UI design, a signifier is any visual or sensory cue that communicates the purpose or function of an element. It helps guide the user in understanding how to interact with the interface, such as a button icon indicating a specific action.

**System Image**
The collective mental model users have about a system based on their interactions with it. It includes assumptions about how the system works, its functionality, and how to achieve goals within it. A well-designed system image improves the usability and efficiency of software or hardware.

**TouchDesigner: COMP, TOP, CHOP, SOP**

**COMP (Component)**: A container used to group and organize elements of a project.

**TOP (Texture Operator)**: Handles image and texture processing.

**CHOP (Channel Operator)**: Works with time-based data, such as audio, motion, or control signals.

**SOP (Surface Operator)**: Deals with 3D geometry and modeling.

**UX/UI**
**UX (User Experience)** refers to how a user feels when interacting with a product or system, focusing on usability and satisfaction. **UI (User Interface)** refers to the visual and interactive elements of a system, like buttons, menus, and layouts.

**VJ**
A Video Jockey (VJ) is an artist who manipulates video and visual elements in real time, often in sync with music or live performances. VJs use software and hardware to create dynamic, interactive visuals for events and performances.

**Visual Programming / Node-Based Programming**
A programming paradigm where users interact with visual elements (nodes) instead of writing text-based code. Each node represents a function or operation, and users connect them to create workflows. It is commonly used in tools like TouchDesigner.

# 1 Introduction

## 1.1. Background

VJing is the act of live performing and mixing visuals that are projected in the venue, usually in a nightclub to complement the music of the DJ (Alexander, n.d., Chapter 5). The term VJ comes originally from 'video jockey' but nowadays 'visual jockey' is a better suited term. The VJ uses self-made or remixed material or generates visuals in real time (or does a combination of all of these) and they can perform in addition to clubs also for example in theaters, art galleries, big summer festivals and more specialized new media festivals (Faulkner, 2006, pp. 10-12). The visuals often include 3D animation, abstract imagery, filmed material or live camera feed.

I became interested in the VJing scene through observing the background videos on music concerts and wanted to know how to make them myself. In 2018, I got to learn VJing through workshops and other VJs, and eventually bought a VJ software called Resolume Arena (www.resolume.com/software/avenue_arena) for myself. It was fast to learn and beginner friendly and I started VJing myself in different events. Besides Resolume, I had noticed that some other VJs had created their own live visuals programs with TouchDesigner (www.derivative.ca), a node-based software, that had a free non-commercial license. I, too, became intrigued to build my own system and custom visuals with TouchDesigner.

During a course called "Audiovisual Media" at Aalto Medialab in 2022, I decided to finally build my own custom VJ program with TouchDesigner which was taught on the course. I managed to build a working system that will be presented in more detail in chapter 3.2. I created the VJing program solely for myself to use in my own VJing but later in 2023, undergraduate design students used it on the "Koe: Teknologia" ("Experience: Technology") Aalto University course, because the students who did not have Mac laptops could not use the VJ program VDMX (www.vidvox.net) taught in the course by Antti Hietaniemi. I worked as a teaching assistant on the course and since there was no good viable free alternative software for Windows, some of the students ended up using my program.

Hietaniemi had been interested in developing a free multi-platform (both Windows and iOS systems) VJ program that could be used in the course and other workshops. The program would need to be easy to adopt for new learners and work as a learning tool in understanding real time visual performance and everything it entails like creating video material, using correct formats, compositing, layering, controlling settings and effects, audio reactivity,

generative visuals and the live performance setting. Since I had already been building and working on a VJ system, I decided to start developing this new VJ program with TouchDesigner as my thesis project. In this thesis, the terms VJ 'tool', 'program' and 'system' are used as synonyms.

## 1.2 Introduction to VJ Software

There are numerous VJ software applications available that enable users to create, edit, and mix visuals - such as videos or generative content - in real time. In addition to traditional VJ software, there are also node-based visual programming tools. Visual programming allows users to create complex interactive systems by connecting visual building blocks, known as nodes, rather than writing code (The Node Institute, n.d.). Some VJs also utilize text-based live coding (Taylor, 2022) or software not originally designed for VJing, such as game engines (Amagi, 2018), to produce live visuals. This chapter provides a brief overview of various commercial VJ software and visual programming tools.

One of the most widely used and well-known commercial programs for live visuals is Resolume Arena (hereafter referred to as Resolume). Its interface is intuitively structured, making it relatively quick for beginners to learn; for instance, I found the workflow easy to grasp when I first started using it. It offers an extensive range of customizable and automatable effects. Additionally, its newer node-based patching environment, Resolume Wire, allows users to build their own plugins, such as generative visuals and effects (Resolume Wire, n.d.). The biggest barrier to using Resolume is its high cost of €799 (€399 for students), which is a significant investment for those new to VJing.

Another prominent commercial VJ software is VDMX by Vidvox (www.vidvox.net), available for Mac computers. The latest versions, VDMX6 and VDMX6 Plus, offer significant flexibility by enabling users to design a personalized modular interface using smaller, adjustable windows. While VDMX6 includes many built-in features, the Plus version allows users to integrate a paid TouchDesigner or Vuo license for creating generative visuals. VDMX6 is priced at $99, with the Plus version costing $249. A free demo version is also available, though it lacks the ability to save projects.

TouchDesigner by Derivative (www.derivative.ca) is a visual programming tool that allows users to build interactive, real-time multimedia systems. Its capabilities are virtually limitless, as input data can originate from almost any source, with the primary limit being computing power. TouchDesigner is made for visual applications and serves as a versatile tool for VJing. However,

its steep learning curve and inherent complexity can pose challenges for new users. Without a custom-built interface tailored for VJing, the workflow can be tedious. TouchDesigner offers a free non-commercial license with a limited output resolution. Other visual programming environments include Notch (www.notch.one), Jitter in Max (www.cycling74.com/products/jitter), and Isadora by TroikaTronix (www.troikatronix.com/isadora). Pricing of these tools ranges from affordable limited learning licenses like £59 for Notch Builder (Notch, 2024) to professional subscriptions of £99 or £189 per month for Notch Builder, €129 per year for Max (Max, 2024), and $199 per year for Isadora (Troikatronix, 2024). Lifetime licenses are also available, costing €399 for Max and $725 or £875 for Isadora.

There are not many inexpensive and user-friendly software options for a beginner to experiment and get acquainted with VJing before investing into an expensive license. Many existing programs are either expensive, incompatible with certain operating systems, or overly complex. While some free tools developed by independent creators are available, these often suffer from technical issues or risk being discontinued. This lack of accessible software also presents challenges for educators and workshop facilitators, as it can be both difficult and costly to provide VJing tools for a diverse group of students or participants using laptops with varying operating systems and capabilities. As a result, finding a suitable option for beginner VJs is particularly challenging, especially in educational contexts where expensive licenses are not feasible.

## 1.3 Research Question and Objectives

This research focuses on the design and production of a pedagogically suitable VJ program that could be used by students and other people who want to learn Vjing. The main research question is: **how to design a VJ program that works as an educational tool that supports skill development and also works as a performance tool?** Working as an educational tool entails that the tool is suitable for beginners and helps the user to understand the workflow and principles of how real time visuals are created and makes it easier to move onto more complex VJ software in the future. The program will be built with the node based programming environment of TouchDesigner, and helping the user to understand the internal logics of the software would be one additional learning objective. Supporting further skill development means that the program offers room for extensible and customizable features to encourage learning. Thus, even more adept users could deepen their skills in creating real time visuals and effects.

In addition, the program should work as a proper standalone VJ performance tool: it should be versatile, meaning it should offer enough features, and work

well in a real time performance setting. This also entails that it should have a clear and practical user interface for smooth and intuitive user experience, and it should work on different kinds of computers and operating systems, mainly Windows and iOS. To summarize, here is a list of all the objectives (some of them can overlap):

Objectives – the VJ tool should:

1. be suitable for both beginners and experienced/intermediate users
2. help the user to understand the workflow and principles of how live visual are created
3. make it easier to move on to more complex VJ software
4. help the user to understand the internal logics of the node based programming environment of TouchDesigner
5. provide possibilities for further skill development by offering extensible and customizable features
6. be a versatile and well functioning VJ system for real time performance
7. have a clear and practical user interface
8. work on different kinds of computers and operating systems (Windows & MacOS)

The program is designed for two target groups: 1) beginners which in the context of this research refer to people who are familiar with other creative digital software and the basic principles of image or video editing but who have not used or have only rarely used VJ software or TouchDesigner before, and 2) experienced users who have practiced VJing or used TouchDesigner for projects before. Since the focus of the study is the aspect of further learning, I will not include in the target groups people who already are highly experienced VJs that have been creating their own custom systems for live visual performance, because most likely my VJ tool would not offer anything new for this group of people.

In this thesis I will focus mainly on the objectives that relate to the possible pedagogical use of the VJ tool (first five objectives on the list). A program that helps the user to understand real time visual performance and node based systems and deepen their knowledge by customizing the program can be a valuable tool for students, teachers and other people who want to learn or teach about VJing for example in the context of workshops.

## 1.4 Methodology and Structure

Chapter 1 introduces the background of this study, context of VJ software, research question and objectives, and methods. In order to achieve the thesis objectives, firstly in Chapter 2 I conduct a literature research on the topics of:

the development of VJ culture, educational design and concepts, design approaches to VJ tools and user interface design. Based on the review, I summarize the main educational concepts, VJ tool design approaches and UX/UI principles and I name them EC (Educational Concept), DC (Design Concept) and UXC (User Experience Concept).

Chapter 3 presents the design and implementation process. I start by presenting an earlier prototype of my VJ tool and evaluate it to find out what worked and what did not work to improve the new design. Based on the findings from the literature review (EC, DC & UXC) and the old prototype, I present a new design and explain how the thesis objectives are met in the design. The VJ program is then implemented in TouchDesigner to create a working interface.

In Chapter 4, a small-scale user test of eight testers is conducted to gain data whether the objectives have been successfully achieved. The study population consists of different skill levels of VJ software users so in the user test there are approximately half novices and half experienced users. In the user test, I first explained to the tester how my VJ tool works, both the interface and the node editing inside TouchDesigner. Then the user would try out first the interface and then they would try to customize the program in some way by editing the data flow structure.

In most of the tests, I was constantly present with the participant, observing the situation except for two testers who were physically in another place and performed some parts of the test independently. For all the testers, I explained the program, showed how it works and provided help. After each test, I interviewed the user with 19 structured questions: most of the questions were closed-ended, with a rating between 1 and 10, and some of them were open-ended. The goal of the questions was to examine how well the thesis objectives were integrated in the program to answer the research questions.

Based on the quantitative data from the interview questions, the mean average of each answer was calculated. The qualitative data from the open-ended questions helped to understand the reasoning behind the ratings and identify the challenges and wishes of the users as well as the strengths of the VJ program. There were repeating themes in the answers, but the testers also had varying opinions. My personal user experience with the tool is included as part of research. Based on the collected opinions and remarks, improvements will be made on the tool.

# 2 Literature Review

## 2.1 VJ Culture Development

VJing can be understood as a subcategory of live audiovisual performance, encompassing various forms of artistic expression, such as live and expanded cinema and visual music, in which visuals and sound are manipulated in real-time performance settings (Carvalho & Lund, 2015, pp. 126-128). This research primarily focuses on VJ culture within the context of clubs and live music performances, which also serves as the foundation for the design process of the VJ tool developed as part of this thesis.

VJing emerged in electronic music events during the late 1970s out of a need to enhance the visual experience for audiences since a DJ's presence alone was not stimulating enough for large crowds, leading to the use of screens and projections to create a visual focal point (Crevits, 2006, p. 14). The first artist credited with using the title "video jockey" or "VJ" was Merrill Aldighieri, who was invited to the Hurrah nightclub in New York to present experimental video art to accompany the DJ's music (Kreukniet, n.d.). She was eventually hired for regular performances and was later discovered by individuals who would go on to establish Music Television Network (MTV), popularizing the term "VJ" during the 1980s. By the late 90s, the spread of club culture and renowned DJs increased the demand for VJs (McCarthy & Gibson, 2023, p. 106). Technological advancements, particularly the ability of laptops to process real-time graphics, allowed greater flexibility for VJs and enabled more venues to incorporate live visuals into their performances.

As VJ culture expanded, tools for remixing live visuals also began to evolve. Before the digital era, VJs relied on analog technology such as VCR players and video mixing hardware, but with the emergence of personal computers, the first softwares for live video editing were developed (Spinrad, 2005, pp. 21-24). One of the earliest digital real-time performance tools, Vujak, was created in the 1990s: it allowed users to sample, loop, and replay multimedia content using a MIDI controller (Russell, 1995, p. 32). Another example is VJAMM, released for audiovisual mixing in live settings in 1997 (McCarthy & Gibson, 2023, p. 103). These tools were initially developed to meet the creators' own needs and were later refined for broader use, with some being released as free resources and others as commercial products.

The 2000s marked a significant period of growth for VJ culture thanks to the rapid proliferation and improvement of digital tools for live visual performance: several prominent commercial VJ software applications, such as Modul8, were introduced, and concurrently, many artists chose to develop

custom systems using programming environments like Max/MSP and Processing (Alexander, n.d., Chapter 7). Today, technology imposes fewer constraints on VJing than ever before, and live visuals have become an established field with dedicated festivals, including MUTEK in Montreal, Canada; Mapping Festival in Geneva, Switzerland; and Aavistus in Helsinki, Finland. In the 2010s and 2020s, the live visuals events "have become increasingly complex public spectacles that seek to immerse audiences into total artworks" (Goodfellow & Gibson, 2023, p. 124). A notable example is the work of Anyma, an artistic project co-founded by Alessio De Vecchi and Matteo Milleri, where De Vecchi creates futuristic 3D visuals, while Milleri composes melodic techno music, combining their talents to deliver immersive performances that push the boundaries of contemporary live audiovisual experiences (Groove Atelier, 2024).

## 2.2 Educational Design Research and Concepts

The first part of the research question, "**How to design a VJ program that works as an educational tool that supports skill development (..) ?**", focuses on how to create an educational VJ tool that can be used as an aid in learning and provides further skill development possibilities. In this chapter I want to clarify how I can tie the thesis topic into a larger picture of educational research and framework: I introduce the concept of educational design research and how my thesis research relates to it. I will also present educational concepts that I can apply in the design of the VJ tool to support the research question and the thesis objectives 1-5 which relate to learning and development.

### Educational Design Research

Design research is usually interventionist, iterative and focused on process, utility and theory (Van der Akken et al., 2006, p. 5). The goal is to design an intervention which is based (partly) on existing theory, improve the design through an iterative process, see how practical it is in the real world and finally contribute the results to the theory of design principles. The research is mostly oriented on specific contexts and not on studying isolated variables. The aim of design research is not to test existing theories but to build upon them and evaluate their effectiveness in practice (Walker, 2006, p. 11).

Oftentimes design research can be smaller scale and less controlled instead of being overly-cautious and lengthy to yield reliable results: "Concentrate on the most important design problems, understand them thoroughly, identify the most promising features for the design in light of that understanding, build prototypes with these features, and try them out" (Walker, 2006, p. 11). There should be a correct balance between risk-taking and cautiousness for

optimal results; bold design approaches have the risk of false conclusions but can also yield innovative solutions.

Design research has also been applied in education. The educational design research process starts with the analysis of a learning problem, and there should be a tight connection between the design principles and the features of the (technology-based) practical intervention (Walker, 2006, pp. 9-10). Developing or adapting appropriate assessment methods for analyzing the learning results is an important part of the research. Reeves argues that educational technologists should adopt this type of design research instead of conducting mostly predictive research which tests theory and hypotheses (2006, pp. 57-59). Developing innovative learning environments is better conducted in a real-life context where it will be used and in collaboration with teachers and students instead of working in isolated laboratory environments and applying the findings afterwards in practice (Reeves, 2006, p. 60).

By defining the learning problems and theoretical background, developing practical solutions, testing and improving them, reflecting on the results and tying them back to theory, this thesis contributes to educational design research. As specified in the introduction, the learning problem is that there are not many viable options for low-cost and accessible VJ software that can accommodate learners of different levels (beginners and experienced users), support the learning process of real time visuals and offer extensibility. Next, I will go through some educational concepts that can support the hypotheses and design solutions of this thesis.

## Educational Concepts

In order to answer the main research question, I have gathered here theories and principles that deal with educational topics like learning, understanding, and creativity. The ideas presented in this chapter will be used as foundation and inspiration for designing the educational interface that has the goal of supporting learning and development.

**Constructivism**, a theory based on the notion that knowledge is internally constructed, non-objective and context-based, proposes an approach to teaching and learning: instead of transmitting information to a passive learner, the teacher becomes a facilitator who provides the student "the opportunity for concrete, contextually meaningful experience through which they can search for patterns, raise their own questions, and construct their own models, concepts, and strategies" (Fosnot, 1996, ix). The student is active, autonomous and reflective. In the context of new media, instead of solely practicing skills with new technology or passively receiving information from tutorials, the constructivist pedagogy emphasizes that the student uses the

technology to gather and analyze information, do research, solve problems, develop new concepts and think critically (Schifter & Stewart, 2012, p. 10). Adopting the constructivist approach can support achieving the thesis objectives 2-5: help the user understand live visuals and node-based programming, make it easier to proceed to more complex software, and provide possibilities for further development.

To understand the world, people create mental **conceptual models** based on previous experience and guidance: "Conceptual model is an explanation, usually highly simplified, of how something works" (Norman, 2013, p. 25). They are not always correct if the person has wrong assumptions. People base their conceptual models on the system image of an object - the available information from the perceived structure like affordances, signifiers, mappings, instructions and interaction with these as well as previous knowledge. A good conceptual model gives deeper understanding which makes it easier to predict behavior, navigate novel situations and solve problems. Through good communication, the designer of a product should provide the user with a good conceptual model based on appropriate information, the system image. For the VJ program to help the user understand how live visuals are created and node-based systems operate, and thus move on to more complex software (objective 2-4), it must provide a good conceptual model: informative design and instructions.

**Low Threshold High Ceiling** (or Low Floor High Ceiling), established by Seymour Papert who developed Logo, a programming language for children, is a principle that refers to tasks or tools that anyone can get started on, but which offer more extension opportunities and increasingly sophisticated challenges when going forward (NRICH, 2019). This philosophy is present for example in Processing (processing.org), a programming environment for visual artists, which is suitable for people with no previous coding experience: it constrains the user to a simple stripped-down interface but nevertheless offers possibilities for complex and advanced projects (Reas & Fry, 2018). Resnick argues for an extra dimension to the principle, Wide Walls, which suggests there are multiple possible paths from the floor to the ceiling and thus everyone can choose the personally most engaging and suitable approach (Resnick, 2020).

Low Floor High Ceiling (Wide Walls) is a principle I intend to use as a basis for the design process of the VJ tool to correspond with the thesis objectives (1 & 5): the program should consist of simple elements to be suitable for beginners, but offer possibility for further development with extensible and customizable features for those who already are more experienced or want to challenge themselves.

Creativity, discovery, exploration and innovation can be **accelerated** by "the capacity to locate, study, review, and revise existing projects and performances" (Shneiderman, 2007, p. 22). This can mean open source projects, web page source code and results found from search engines, which help the learner to see what is possible and build on previous works. Analyzing previous projects is a great way to get forward: for example, if someone wants to learn TouchDesigner better, they can examine other projects made with it. Since one of my research objectives (4) is helping the user understand the logics of node based programming in TouchDesigner, I aim to design the program in a way that can clarify how the system works and provide an example case, which accelerates the learning process of the user.

To summarize, here are the four educational concepts I will utilize in the design and the corresponding thesis objectives (EA referring to educational concept):

- **EC1) constructivism:** knowledge is internally constructed, which implies that learning should be active (reflecting, analyzing, searching for patterns, researching, problem solving, developing new concepts and strategies) instead of passive (receiving information) → objective 2-5
- **EC2) conceptual models**: to achieve deeper understanding and problem-solving, the design should communicate a clear system image (available information like affordances, signifiers, mappings, instructions, previous knowledge of user) that enables the user to create a correct conceptual model → objective 2-4
- **EC3) Low Floor High Ceiling:** the program should consist of simple elements to be suitable for beginners, but offer possibility for further development with extensible and customizable features for those who already are more experienced or want to challenge themselves → objective 1 & 5
- **EC4) accelerator:** provide an example that clarifies how the system works, which accelerates creativity, discovery, exploration and innovation of the user when they can study, review and revise the example case → objective 4

## 2.3 Design Approaches to VJ Tools

This chapter addresses the second part of the research question, "How to design a VJ tool that (...) **works as a performance tool?"**, focusing mostly on how to create a versatile and well-functioning VJ system for real-time performance (Objective 6). To gain valuable insights, I examine various design approaches, previous research, and insights from other designers on how to create digital tools for VJs.

One approach to building VJ software is to avoid incorporating overly complex and highly developed effects that constrain the VJ to a predefined aesthetic determined by the developer, according to the co-author of Modul8 software, Dornbierer (2009, pp. 238 & 244). Instead, to maximize the possibilities for improvisation, there should be an emphasis on the flexible combination of different relatively simple effects. This encourages artists to develop their own visual styles: "When you watch an artist creating totally unforeseen and surprising things with an instrument that you have created, you can say that one of your main objectives has been attained" (Dornbierer, 2009, p. 245).

A study by Correia and Tanaka (2014) on user-centered tools for interactive audiovisual performance highlights several features that VJs find essential: modularity, flexibility, ease of integration with hardware and software, diversity of generative features, reliability, speed, and user-friendliness (pp. 87-91). There was also a consensus that existing video manipulating tools were easy to use but there was a lack of simple tools for computer generated graphics (p. 97). One participant wished for a simple mapping of audio-reactive graphics, such as a single-button function. These insights provide practical approaches to achieving Objective 6.

Research by Hook et al. (2011) explored the design of expressive tools for VJs, emphasizing interaction. The VJs involved in the study emphasized the importance of enabling skill development (Objective 5) and having extensive control over the visuals, particularly for live-generated graphics (pp. 1269-1271). On the other hand, they also noted the value of constraining interaction within the technology they used, as these constraints enhanced focus, reduced the overwhelm caused by too many options, and stimulated creativity. Other critical aspects included immediacy and reconfigurable interfaces tailored to specific performances. To address the contrary findings regarding the amount of control and constraints, the authors propose that VJs could "benefit from complex interaction possibilities that are contained within a well-defined space," allowing users to modify and limit the interface to suit their needs (Hook et al., 2011, p. 1273).

Levin (2000) also addresses the topic of balance between complexity and ease of use in performance software, noting that systems rarely achieve both, "for to be so demands that their rules of operation be simple yet afford a boundless space of possible outcomes" (p. 56). These can sound like contradictory directions, but designing such systems is an aspirational goal, in the same way as the Low Floor High Ceiling concept discussed earlier. The software must maintain inexhaustible creative possibilities to sustain user engagement and provide nuanced control to foster personal expression. Levin argues that interactions limited to manipulating predefined graphics result

in shallow and restrictive performance experiences (2000, pp. 46, 54, 56), echoing Dornbierer's earlier observations (2009).

From these findings, I have distilled the key concepts to guide the design of my VJ tool, aligning them with specific objectives. The abbreviation DC refers to a design concept.

- **DC1: Personal Style** – The ability for users to combine simple effects and create their own rather than relying on highly developed effects, avoiding predefined aesthetics (Dornbierer, 2009; Levin, 2000) → objective 6
- **DC2: Compatibility** – Ease of integration with hardware and other software to ensure versatility (Correia & Tanaka, 2014) → objective 6
- **DC3: Skills Development –** The potential for users to develop their skills further (Hook et al., 2011) → objective 5
- **DC4: Balance of Control and Constraint** – Extensive manipulation possibilities for diverse visuals, especially generative, combined with interface constraints to enhance focus (Hook et al., 2011; Correia & Tanaka, 2014) → objective 1
- **DC5: Simplicity and Complexity** – A system that operates on simple principles while providing inexhaustible creative outcomes (Levin, 2000) → objective 1

## 2.4 User Interface Design Principles

In this section I present general user experience and interface (UX/UI) design principles which I find relevant to answer the second part of the research question, "How to design a VJ tool that (...) **works as a performance tool?**". I will focus mostly on objective 7: how to make a clear and practical user interface. I have collected a few principles which I want to focus on out of the best practices of UX/UI design (Yablonski, 2020).

Starting on the visual principles, **Aesthetic-Usability Effect** is a simple yet important rule: beautiful design is seen as more usable as well (Yablonski, 2020). My assumption is that VJs are most likely very visually oriented people so a visually pleasing and unified interface would make the user experience better as well and provide clarity. Another principle relating to visual design is the **Law of Common Region** which states that when a clear boundary defines an area, the objects inside it are understood as a group. Also, the size of and distance to a target affect how fast and correctly the user can access it according to **Fitts's Law**: "fast movements and small targets result in greater error rates, due to the speed-accuracy trade-off" (source).

A principle on building the functions of the system I find useful is **Jacob's Law**: the users have a background using other programs and thus they want

your program to work in a similar way (Yablonski, 2020). If the user does not have to learn entirely new models, they can focus most of their energy on the actual tasks. This means that when the user moves on to learning new programs, it becomes easier to comprehend them if they work in a similar way to the previous program. This can be connected to one of the research objectives: how to make it easier to move on to more complex VJ software (objective 3).

I hope to find ways to apply these basic but important principles when designing the functions and different user interface elements of the VJ tool. Here is a list of the UI/UX concepts and which objectives they answer (UXC referring to user experience concept):

- **UXC1) Aesthetic-Usability Effect:** beautiful design seen as more usable → objective 7
- **UXC2) Law of Common Region:** when a clear boundary defines an area, the objects inside it are understood as a group → objective 7
- **UXC3) Fitt's Law:** the size of and distance to a target affect how fast and correctly the user can access it → objective 7
- **UXC4) Jacob's Law:** the users have a background using other programs and thus they want your program to work in a similar way → objective 3

# 3 Design and Implementation

Before creating the new design, I present the initial prototype (Prototype 1) of the VJ tool I made in 2022 in Chapter 3.1. I will evaluate Prototype 1 based on my own user experience and that of undergraduate students who used Prototype 1 on a course. The insights on the first prototype support the design of the new Prototype 2 by demonstrating what can be improved and what worked well.

In Chapter 3.2., I present the design process of the new Prototype 2, which was initially built in the Figma application before the technical implementation. In Chapter 3.2.1., I explain the main features of the design, and in Chapter 3.2.2., I present how the design objectives are integrated in the new design. The objectives are based on the educational concepts, design approaches, and UX/UI principles presented in the literature review. After the design is finalized, I implement the new VJ prototype in the TouchDesigner development environment to build a working program, as described in Chapter 3.4.

## 3.1 Old Design: Prototype 1

In spring 2022, I created the first iteration of a VJ system interface (Prototype 1.1) during the *Audiovisual Studio* course at Aalto Media Lab. This project was initially intended for my personal use, as I wanted to explore what it would take to design such an interface in TouchDesigner. Inspired by other TouchDesigner interfaces built by VJs, I aimed to create my own version, enabling me to design custom effects and generative visuals. The intention was to further develop the initial version after the course ended.

The development process resumed in autumn 2023 when I worked as a teaching assistant for Antti Hietaniemi's course *Koe: Teknologia* ("Experience: Technology") for undergraduate design students at Aalto University. As some students lacked Mac laptops and therefore could not use the macOS-based VJ software VDMX which was used in the course, my program became a viable alternative for those with Windows systems. This required updating and improving the tool, including adding new effects and addressing usability issues. These updates resulted in Prototype 1.2 (see Image 1). Observing students interact with the tool and gathering their feedback provided valuable insights. Then in spring 2024, I created the most recent iteration, Prototype 1.3, specifically for use in my own VJ set. This version introduced significant changes, including the removal of many previous effects and the addition of more complex generative visuals.

The prototype consists of five main sections: imported media layers, generative visuals, 3D object, composition effects and the final preview with blending mode options and final adjustments (Image 1). The imported media section allows the user to select video or image files as inputs and manipulate them with basic effects. The generative visuals and the 3D object are generated directly within TouchDesigner and these sections let the user adjust parameters to create customizable visuals and activate audiovisual features. Earlier versions of generative visuals included effects like rising sphere particles and moving noise (see Image 1), which were replaced by more advanced visuals in the latest iteration. The composition effects are applied to the entire composition before final output adjustments. The final preview section features two smaller preview windows: *Layer 1: Imports* combines the three imported media layers and *Layer 2: 3D and Generatives* combines the 3D object and generative visual layers. The user can select blending modes within and between these layers, as well as choose which layer (1 or 2) appears on top. However, the order of input layers within Layer 1 and Layer 2 is fixed. The user interface elements include uniformly colored sliders, toggle buttons, and distinctively colored (blue tones) buttons for defining 3D shapes or blending modes. To signify grouping, some sliders are green, even when located in different sections, due to spatial constraints in the *Composition Effects* section.

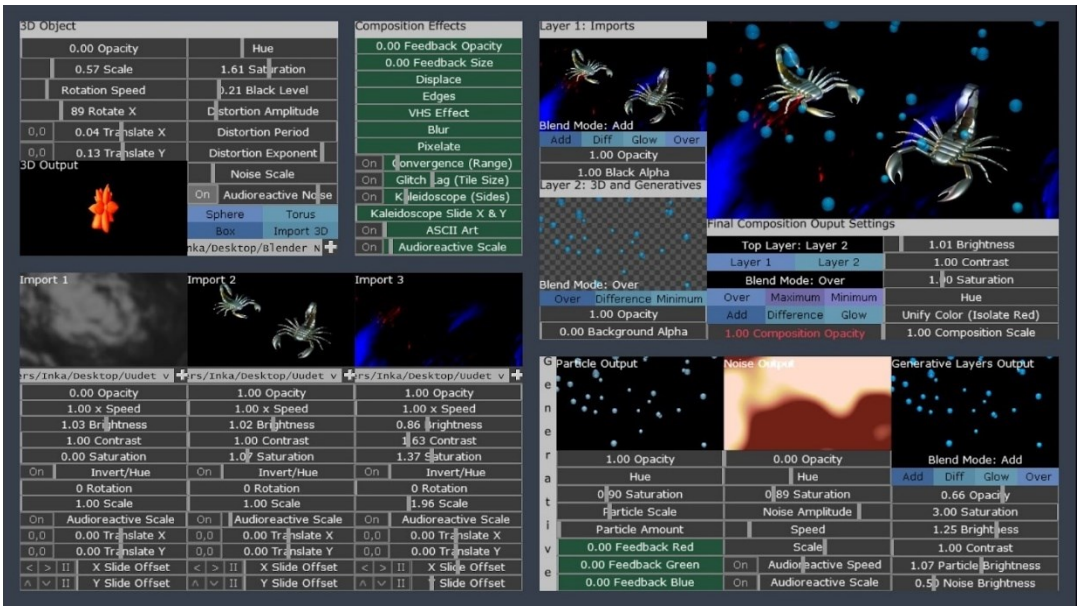


**Image 1:** the second improved iteration, Prototype 1.2, copyright Inka Jerkku, 2023

Based on the initial prototype, my idea for this thesis was born: my goal became to build a new and improved version of the tool, guided by specific design principles and user-centered goals. Because the final product needed to be free or low-cost, I decided to continue development in TouchDesigner.

This platform is free for non-commercial projects (with limited resolution) and compatible with both Mac and Windows systems, making it accessible to a broader audience. Additionally, I was already familiar with TouchDesigner's node-based environment, which suited my limited background in text-based programming. TouchDesigner also allows for the integration of Python code if necessary. For end-users, familiarity with visual programming is not required to use the interface. Nevertheless, the structure of the node-based system makes it possible for users to explore and learn TouchDesigner principles if they wish, encouraging deeper learning.

In the following sections, I will outline the insights gained from these prototype iterations, both from my own experience and from observing others. I will highlight areas for improvement and discuss specific features and functionalities that should be redesigned in the final VJ tool to improve the interface and overall user experience.

### 3.1.1 Observing BA Students' User Experience

I will shortly present my observations during the "Koe: Teknologia" ("Experience Technology") course by Antti Hietaniemi in Aalto University in the autumn 2023 when some of the undergraduate design students attending the course used my prototype VJ tool. This mandatory course was focused on building a VJ performance as a group for a final showcase: the students learned how to make their own material as either 2D or 3D visuals and how to manipulate the videos in real time on a VJ software. The main VJing tool for the course was VDMX but since it does not work on the Windows system, those students who had Windows, around 7 people, would end up using the VJ program built by me (Image 1). I did not conduct any surveys or interviews during the course (at the time I had not decided on this thesis project yet) so I do not have comprehensive data about the students' experiences with my program. However, I made some observations that I will mention here that provide some general information about how they experienced the project.

The first thing to note is that, compared to VDMX, my program had significantly less features and effects and after a while the students would run out of effects to use. In VDMX it is also possible to map individual effects to the audio data or other parameters, which could not be done in my prototype interface, so it was much less flexible in that sense. In general, the students did seem to learn to use my interface quickly and did not have any major problems while using it. I noticed that no one used the generative effects (floating spheres and moving noise) but the generative 3D visual was used. In addition, a few people did not know some of the terms used in the names of the sliders like "alpha". Connecting to a midi device was also a slightly

complicated task and required knowledge about TouchDesigner nodes and code, so I handled that process instead of the students.

During Hietaniemi's course, we integrated GLSL shaders into the TouchDesigner project using an ISF parser add-on (https://github.com/marcinbiegun/isf-touchdesigner). This allowed us to add complex effects from VDMX and other sources into the program. While this greatly expanded the range of effects, it also introduced challenges: The GLSL effects demanded significant processing power, increasing the risk of crashes and overheating. For my own performance in 2024, I chose to remove all GLSL effects to prevent performance issues when adding more complex generative visuals. In the future, I would prefer to build effects directly using nodes in TouchDesigner. While this approach requires significantly more time and effort, it offers greater stability and efficiency during performance.

In general, everyone managed to use my program seamlessly as part of the final group performances. Observing them during the course and answering their questions taught me nevertheless what I should add and what to remove from the tool when designing the new version. In any case, I cannot strive to build a program that would be fully equal to VDMX and have all the same features, because that is out of scope of this thesis. I can, however, aim to design and build a VJ tool that can be used on the same course in the future and include as many useful features as possible.

### 3.1.2 Personal User Experience

Having performed as a VJ at various events since 2018, I find my personal experience with this VJ tool relevant. Although, I acknowledge that since I am the developer of the tool, I already knew exactly how it worked, and thus I cannot assess how intuitive or easy it is to learn for new users. Based on my own user experiences, I will outline what I learned from using the prototypes and suggest improvements.

I personally used the very first version of the interface (Prototype 1.1) twice and the latest improved version (Prototype 1.3) once in live VJing settings. The initial performances took place at a small live streaming event and a student party, while the most recent occurred during a techno club event, where I performed for a short period (approximately 30 minutes). The prototypes generally functioned well during my live performances and I was able to create aesthetically interesting visuals with audiovisual effects, and the system ran smoothly without technical failures. However, the performances revealed some limitations in the interface design:

- Testing compositions in advance was challenging. Often, I had to make changes to the final composition without adequate previews, leaving me to “hope for the best.”
- Without enough pre-made material, the tool's capacity for creating a diverse range of visuals over extended periods of time was limited.
- The generative effects in earlier prototypes lacked the complexity needed to sustain interest in longer performances.

Based on my experiences with the prototypes, I propose the following enhancements for future development:

- Enable users to easily create and add their own generative visuals to the interface.
- **I**dd more input layers to allow greater flexibility in compositing visuals.
- Increase the number of effects that apply to individual layers rather than the entire composition. This would also enhance previewing options.
- Include more blending mode options to expand creative possibilities.
- Incorporate adjustable audio input levels to better regulate audio-reactive effects.
- Expand the scope of transform parameters (e.g., translate and scale) to allow greater freedom in compositing.
- Simplify the process of connecting and configuring MIDI devices.
- Address the need for additional space by making sections scrollable and/or using tabs to organize the interface more efficiently.

These observations and proposed improvements will guide the design and development of the final thesis project. By addressing these issues, I aim to create a more versatile, intuitive, and robust VJ tool. Moving forward, I will keep these remarks in mind during the design phase to ensure the final product meets both my own needs and those of potential users.

## 3.2 New Design: Prototype 2

The new design is based on addressing the research question and objectives, and it incorporates ideas from the literature review. The new prototype (Prototype 2) is an improved version of the first prototype versions (1.1, 1.2 & 1.3) and it is inspired by the case studies. I created the new design of the prototype first in Figma and then built the program in TouchDesigner. In this chapter I will present the main structure and features of the new interface, the reasonings behind the design and the production process.

### 3.2.1 Main Features

The new design plan of Prototype 2 was first created in Figma as a non-interactive visual layout (Image 2). I made multiple versions of the design (Appendix) but here I present the final version. The design contains three stacked horizontal sections which are confined by colored borders. Originally, I planned to use different border and slider knob colors for each section, but I ended up using only one color to avoid visual overwhelm. The top section includes on the left side several layers for imported materials (the user can upload a video or image) titled “Import (number)” which have small preview windows and sliders which can be scrolled down, and on the right side there is a bigger preview for all of the combined import layers. In the middle section, there are in a similar sense multiple layers called “Generative (number - title)” on the left side, and the bigger preview for the combined generative layers on the right. All generative visuals are premade by me, but the user can control their attributes with the sliders which control specific generative parameters, for example the noise distortion of a 3D object. In both “Import” and “Generative” sections, the layer number 1 is shown on top of the other layers, and the same logic applies for the subsequent layers on the respective horizontal section.

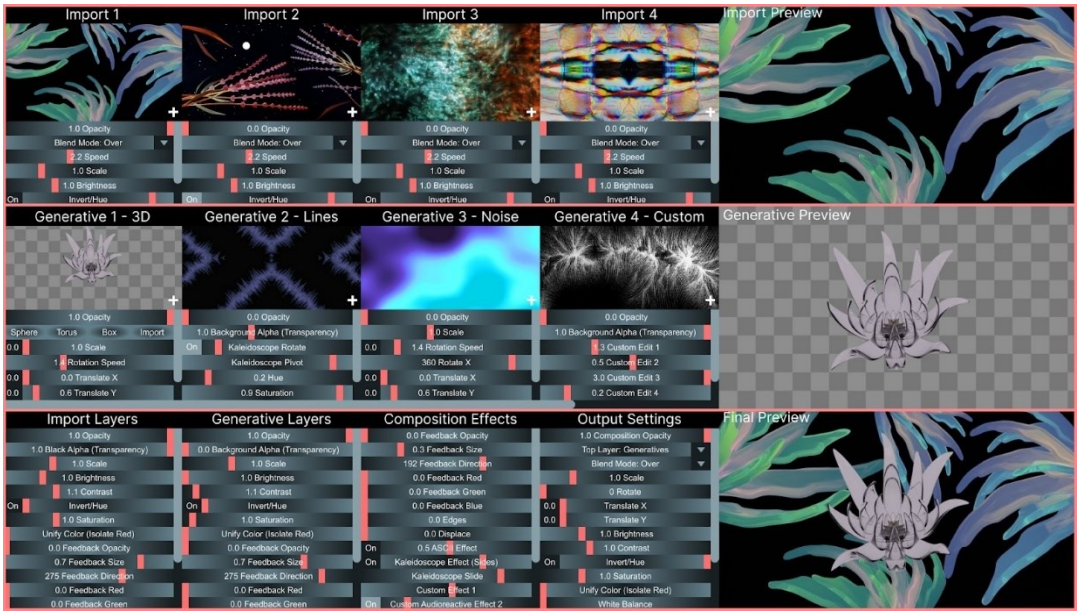


**Image 2:** Prototype 2 design plan made in Figma, copyright Inka Jerkku, 2024

The bottom section is divided into fours subsections (on the left side) and the final composition preview (on the right side): “Import Layers” subsection covers the settings for the combined import layers (top row), “Generative Layers” covers the settings for the combined generative layers (middle row) and

“Composition Effects” controls the effects of the final composition. Hence, there are two main sections of layers, the Import layers and Generative Layers, which are combined together to form the final composition which can be seen in the bottom right corner. Lastly, “Output Settings” subsection controls the final composition as well, providing additional basic and specific output settings like defining the top layer and blend mode. This design plan was the initial starting point, but some things were done slightly differently in the final design.

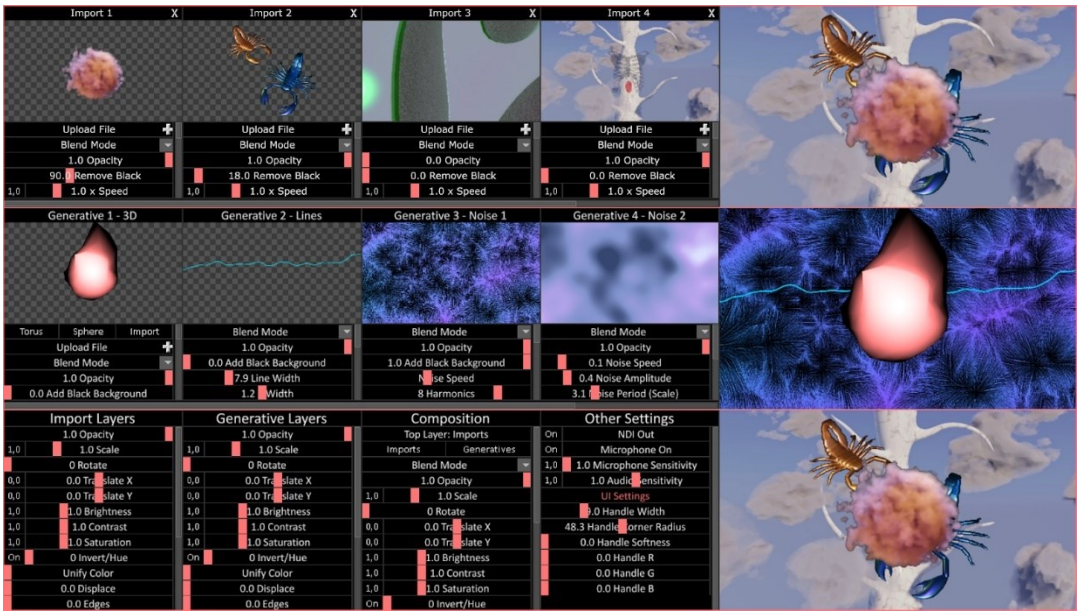

**Image 3:** Prototype 2 final program made in TouchDesigner

Based on the Figma design plan, I built the new Prototype 2 program in TouchDesigner (Image 3). The production process is described more in chapter 3.4. The final structure was the same as in the design plan but some changes were made: Instead of dividing the final composition settings into two subsections “Composition Effects” and “Output settings”, I combined these settings in the “Composition” subsection. Instead of “Output Settings” I added “Other Settings”, which includes for example audio and user interface settings. This subsection also shows the framerate and time and has a button for exporting video. Another major change from the Figma design is the sliders: I decided to make them all black instead of using gradient colors, because I thought gradients would have been too visually overwhelming. The user can change the color of the slider knobs and borders as well as the width, roundness, and softness of the slider knobs. In the final program, each layer has a small X-button in the right corner that disables cooking in the corresponding layer if it is not used to save processing power thus increasing the frame rate of the output.

Here is a link to a video which demonstrates in more detail how the final Prototype 2 works: https://youtu.be/WvBVLH1-kLw

### 3.2.2 Applied Design Objectives

Next, I will describe how the thesis objectives were incorporated into the new design. The objectives are based on the research question: how to design a VJ program that works as an educational tool that supports skill development and also as a performance tool? The objectives 1-5 correspond to the educational approach (works as an educational tool that supports skill development) and 5-8 correspond to the performance tool approach (works as a performance tool). Objective 5 overlaps both parts of the research question. To answer the objectives, I use the concepts defined in the literature review and insights learned from the case studies and Prototype 1 as foundations for the design. The abbreviation EC refers to educational concept (listed in chapter 2.2.), DC refers to design concept (chapter 2.3.) and UXC refers to user experience concept (2.4.)

**Objective 1: be suitable for both beginners and experienced users**

To make the VJ tool suitable for different user groups, I have aimed to make the tool operate on simple rules but offer inexhaustible outcomes (DC5) to follow the Low Floor High Ceiling (EC3) principle. Thus, the basic elements of the program are simple: there are three sections, two of which contain the different layers. Out of these two, one consists of imported materials and the other consists of generative materials. The third section contains the settings for the combined layers and final settings. Although the structure is simple, the program can be infinitely expanded as well: the user can add as many new layers inside the two main sections (which are horizontally scrollable) as they want and create new effects, controls and generative visuals.

**Objective 2: help the user to understand the workflow and principles of how live visuals are created**

To help the user learn more about live visual performance and each step it entails, the VJ tool provides possibilities for the user to actively construct and manipulate the program itself, which encourages subjective, contextual and active learning which includes for example analyzing, doing research, solving problems and developing new ideas (EC1). On top of this, I have strived to create a program that helps the user to form a correct conceptual model based on the system image through signifiers, affordances, mappings, instructions and other available information (EC2). This means that I hope the layout of the interface and the organization of the nodes in TouchDesigner

communicate a clear idea of how the visuals are created, organized, layered and controlled. I will also include comments inside the node system that help the user to navigate and make a separate instruction sheet as well. A good conceptual model helps the user to gain a deeper understanding of the visuals and the system, which supports navigating novel situations, problem solving and future skill development.

### Objective 3: make it easier to move on to more complex VJ software

The approaches mentioned in the preceding objective, the possibility for active learning (EC1) and the communication of a correct conceptual model through a clear system image (EC2), also assist in moving on to more complex VJ software since the user will already understand hopefully at least the basics of real time visual performance tools after using my VJ tool. In addition, when moving onto new software, the user does not need to use as much energy in learning an entirely new model when the software is similar to the one they have previously used (UXC4). That is why I wanted to build an interface that is not too different from other VJ software but includes similar structures and elements. For example, in my VJ tool, there are two main sections consisting of individual layers and these two sections then form the final composition output, which is similar to how there are two main compositions that consist of several individual layers in Modul8. The layers in my VJ tool are both horizontal and vertical: the input layers inside the two main sections are in a horizontal order while the two main sections are vertically stacked - this resembles partially the organization of layers in for example Resolume and VDMX (although the user can also create their own personal interface) which are not identical to my VJ tool but similar in the logic of layers. The main idea of my VJ interface is that it is not fundamentally different in its logic like for example the Vimix program which has a very unique operating principle compared to most VJ software.

### Objective 4: help the user to understand the internal logics of the node based programming environment of TouchDesigner

In addition to providing possibility for active learning (EC1) and communicating a correct conceptual model (EC2), presenting an example TouchDesigner project of a VJ interface with generative visuals can accelerate the learning process of someone who is studying how visuals are made in TouchDesigner and other node-based software (EC4). The example shows how the system works and thus provides more understanding and clarity in comparison to starting from nothing when building a system for real time visuals. The user can examine how the node system is constructed in the

TouchDesigner file and hopefully gain new insights and creative ideas that they can also use in their own future projects.

**Objective 5: provide possibilities for further skill development by offering extensible and customizable features**

This objective relates to the first one, how to design for both beginners and experienced users. Following the Low Floor High Ceiling principle (EC3), the high ceiling is made possible by enabling the user to expand the VJ tool as much as they want by creating new layers, effects, controls and generative visuals. Modifying the program independently supports active learning when the user needs to do their own research and discover new information (EC1), which will then develop the skills of the user (DC3). The possibility of expanding the program in a relatively straight-forward way by using the well-organized node system is one of the core features that bring an educational dimension to the VJ tool.

**Objective 6: be a versatile and well-functioning VJ system for real time performance**

In order to make a diverse VJ program, it should allow the user to express their own style and not predefine the aesthetic of the visuals too much (concept DC1). That can be achieved by letting the user combine a lot of simple effects and create their own instead of incorporating many highly developed and complex effects in the VJ tool. There will be predetermined generative visuals in my program, but their point is to work as a starting point so that the user can later create their own custom generative visuals if they want. A well-functioning VJ program should be easily integrated with hardware and other software (DC2), which can be achieved since TouchDesigner allows almost any input and for outputting the visuals to another software or computer there will be an option to use NDI.

I have also learned a lot from the earlier prototypes, and I will use that knowledge to improve the design and features. In chapter 3.2. I have described what kind of changes should be made in the new design: more effects for each individual layer to make previewing them easier (compared to only applying the effects to the final composition), more options of generative visuals, only using effects made in TouchDesigner (compared to using GLSL shaders), more input layers and blending modes, adjustable audio input intensity, bigger scope in the transform parameters, easier connecting to a midi device and more space in the interface by making the different sections scrollable.

**Objective 7: have a clear and practical user interface**

To make an interface that is clearly understandable and practical in live performance, I have strived to make the interface aesthetically harmonious (UXC1), distinctly define the different sections with a colored border (UXC2) and make the slider knobs and buttons correctly sized and located so that the user can reach them quickly and without error (UXC3). There should also be a correct balance of constraint and control in the interface so that it does not feel too simple and limiting or too complicated and overwhelming to help with focusing (DC4). That is why I have divided the interface into three similar looking sections that can each contain many layers and effects which can be expanded - this way the interface is equally simple and complex. Compared to the old Prototype 1, the new interface is clearer and the different elements are organized more logically. The flow of the visuals moves from left to right and from top to bottom, which I find intuitive for the user. The user can also edit the highlight color in the interface to make it more personal and visually pleasing.

**Objective 8: work on different kinds of computers and operating systems**

The last objective is achieved by building the VJ tool with TouchDesigner: the program works on both Windows and MacOS. In addition, the non-commercial license for TouchDesigner can be downloaded for free, so my VJ program can be very widely available. The biggest limit to using the VJ tool is the processing power: the user should have access to a computer with at least a moderately powerful CPU and GPU to run graphics and operations in real time.

## 3.3 Implementation in TouchDesigner

After the design plan was made, I started to implement the program in TouchDesigner. Since I had already built and worked on the previous version Prototype 1 in TouchDesigner, I did not have to start the building from scratch. I did create a new TouchDesigner project, but I used some elements from the old version, for example buttons and effects. Nevertheless, I recreated the whole structure, but the process was faster because I had learned from previous mistakes and knew how to execute the program.

When the user opens the TouchDesigner file (.toe format), they will see the TouchDesigner working environment that shows the inner structure of the nodes in the project file. To use the VJ interface, the user has to press the F1 button that opens the interface view in performance mode, and to close the interface, they have to press the ESC key. When the VJ interface is closed, the

user can navigate inside the TouchDesigner node structure and make edits to the program.

When building the program, I used COMP containers to divide the interface into different sections. Since each part of the interface is inside a specific container, the user needs to navigate to a correct location in the project file if they want to make changes. Inside each container, I have stacked the child elements, for example from left to right or from top to bottom - this is how I organized the layers and the sliders. When stacked, the order of the child element is determined by its location on the X (if stacked horizontally) or Y (if stacked vertically) axis inside the TouchDesigner space. This is the same technique as how I built the Prototype 1 interface, although then I stacked the child elements based on the alphabetical order of their names instead of the X or Y location. Organizing the children based on the name was not ideal since I needed to rename every single one every time I wanted to make a change to the order. Organizing based on the horizontal or vertical location was much easier.

Every slider or button is mapped to a corresponding effect or setting which is located in its allocated container. When the user understands the logic of the program, they can customize the interface: for example, they can add more layers to either the import or generative section, create whole new generative visuals or add new effects and their controls to any layer or section. There are countless tutorials for TouchDesigner on the internet, so the user can basically add almost anything new to the interface. I strived to make the TouchDesigner file as understandable as possible to make customizing it easier: I have named each container, slider and effect to describe its function, and I have added some instruction annotations inside the file.

Most of the settings and effects used in the program are controlled by basic TOPs in TouchDesigner like Transform (scale, X and Y location, rotation), Level (brightness, contrast, opacity), HSV Adjust (hue, saturation), Feedback, Blur, Edges or Displace. There are also some audio-reactive effects which use as input either the microphone (or some other connected audio device) or an uploaded audio file. I did not add many advanced effects, since that is something that the user can build themselves if they want to. On the other hand, some of the generative visuals are relatively complex and predefined, but they work more as placeholders for the user, so they can later build their own custom generative effects. As generative visuals, there are four options: 3D object controls and distorts either a sphere, torus or an uploaded file (.obj); Audio Line shows the visualized audio waveform based on the amplitude; Noise 1 is a more complex noise visualization; and Noise 2 is a very simple visual of a noise. All the generative visuals use SOPs as their basis except Noise 2 which uses only TOPs.

I noticed that the TouchDesigner file can get quite heavy in terms of processing power. Since I want the VJ program to be available for different kinds of computers, I had to think of ways to optimize the file as much as possible. In TouchDesigner, the maximum frame rate for the output is 60 frames per second (fps). To make the frame rate as high as possible and use less computing power, I reduced the video resolution to HD (1280 x 720) instead of full HD. I also added a button to each layer (a small X in the right upper corner) that unables processing in that specific container so that the user can close all the layers they do not need. This was an effective way to increase the frame rate, since especially the generative visuals were using a lot of power. These were the main techniques to optimize the program in addition to some smaller changes to make the program run smoothly. However, the VJ program requires a computer which has at least a moderately powerful graphics card.

# 4 Results

## 4.1 User Study

After the working prototype of the VJ tool was built, I conducted a user test with eight participants who I all knew beforehand. Since the target group of the VJ program is both beginners and experienced users, the testers also had different levels of experience from VJ software. Mentioned earlier in the introduction, a beginner refers to someone who has no experience from VJing (or only very little experience) but is already familiar with creative digital software like for example Adobe software and knows the basics of image or video editing. Half of the participants belong to this group and the other half had practiced VJing for a longer period.

When it comes to using TouchDesigner, three out of the four experienced VJs had been working with TouchDesigner on several projects, and one of them was highly advanced. Also one of the beginners had been working with TouchDesigner before. So in total there were four testers who were intermediate or experienced users of TouchDesigner, and four who had no experience or very little experience. I included one tester who was perhaps too advanced in terms of TouchDesigner for this study to see how they would respond to the VJ tool and if they would find it useful. All in all, the group of testers could be divided into beginner VJs, experienced VJs, beginner TouchDesigner users and experienced TouchDesigner users. Other demographics of the participants were not asked but for context the general age was around 25-40, most of them worked or studied creative media, and the distribution of genders was rather balanced. In my opinion, these attributes were not relevant to the tests, because the testers fit the target group otherwise, so I did not measure them separately.

### 4.1.1 Test Setting

In the user test setting, I would be present with the participant either physically or in two instances, via a video call. In most cases I was present during the whole session but in two cases the user performed some parts of the test independently due to scheduling limitations. The test session started with me explaining and demonstrating how the interface works and how the node editing inside TouchDesigner is structured. Then the tester would first play around with the interface as if they were VJing for a while, for example uploading their own videos. Next, they would try to customize the program in some way, for example adding a new slider and effect, by editing the TouchDesigner nodes to make changes. I would guide and help the participant in

detail since this second part could be challenging to many users. After the test, I would interview the user.

### 4.1.2 Interview Questions and Results

The user test questions were based on the design objectives, presented in Chapter 3.3.2, that have been integrated with the applied findings from the literature review, including the educational concepts, design approaches for VJ software, and UI/UX principles. The user test questions are both open and closed-ended: There were in total 19 predefined questions out of which 16 were numeric ratings between 1 and 10. The ratings followed the logic of semantic differential scales (Preece et al., 2019, p. 281) where the user would have to choose a position between two opposite descriptions to rate a specific feature or element of the VJ tool. In all ratings except for two, 10 was the best possible answer and 1 the lowest, but in two questions number 5 indicated perfect balance between two polarities. The participant could add comments, verbally and I would then write down the comments since I was present, to each rating if they wanted, providing qualitative data.

| | Experienced VJs: | | | | Beginner VJs: | | | |
|---|---|---|---|---|---|---|---|---|
| Question: | User 1 (TD) | User 2 (TD) | User 3 (TD) | User 4 | User 5 | User 6 (TD) | User 7 | User 8 |
| 2 (1-10) | 10 | 8 | 7 | 5 | 10 | 10 | 8 | 7 |
| 3 (1-10) | 8 | 7 | 7 | 9 | 10 | 8 | 8 | 10 |
| 4 (1-10) | 10 | 9 | 3 | 8 | 10 | 8 | 9 | 8 |
| 5 (1-10) | 7,5 | 6 | 5 | 5 | 6 | 10 | 10 | 10 |
| 6 (1-10) | 7,5 | 9 | 7 | 5 | 10 | 3 | 8 | 7 |
| 7 (1-10) | 9 | 2 | 9 | 5 | 9 | 10 | 9 | 10 |
| 8 (1-10) | 10 | 10 | 9 | 8,5 | 10 | 10 | 9 | 9 |
| 9 (1-10) | 8 | 6 | 5 | 7 | 6,5 | 8 | 8 | 8 |
| 10 (1-10) | 10 | 7 | 7 | 8 | 10 | 8 | 5 | 8 |
| 11 (1-10) | 5 | 7 | 5 | 2 | 10 | 5 | 5 | 6 |
| 12 (1-10) | 8 | 10 | 7 | 8 | | 9 | 9 | |
| 13 (1-10) | 6,5 | 9 | 8 | 7 | 7 | 7,5 | 7 | 8 |
| 14 (1-10) | 10 | 9 | 7 | 8 | 8 | 7,5 | 9 | 7 |
| 15 (1-5-10) | 4 | 5 | 5 | 5 | 5 | 3,5 | 5 | 4 |
| 16 (1-5-10) | 3,5 | 6 | 5 | 6 | | 5 | 5 | 5 |
| 17 (1-10) | 7,5 | 10 | 10 | 7 | 7 | 10 | 8 | 10 |

**Table 1:** The quantitative results of the user test. The blue vertical bar indicates the question number, and its scale is in the parenthesis. The horizontal rows specify the answer of each user, first the experienced VJs (pink) and then the beginner VJs (green). TD next to a name means the user is experienced in TouchDesigner.

Next, I will go through all the **user test questions and answers**, including the **mean averages** (rounded to one decimal) of the numeric answers and **summaries of the additional comments**. I have categorized the quantitative answers (in questions 2-17) into five different averages: the total mean average (8 users), the average of the beginner and experienced VJs (4 users each), and the average of the beginner and experienced TouchDesigner (abbreviated TD) users (4 users each). All the individual numeric answers can be seen assembled in Table 1.

**1. Would you describe yourself as a beginner or experienced user of VJ software and/or TouchDesigner?**

- Since the focus of this thesis is on VJing, I have divided the user test participants primarily into beginners and experienced users based on whether they have experience from VJing or not. Having experience in using TouchDesigner is secondary but also important, so I have also categorized the participants based on their experience with TouchDesigner.
- Two participants were total beginners: they had not used either VJ software or TouchDesigner but one of them did have programming experience and had used the nodes in Blender software. One participant had not used VJ software but had used TouchDesigner in a few projects. Another beginner in VJing had tried VDMX software and TouchDesigner slightly but never performed as a VJ or built their own TouchDesigner projects. They had also worked with creative coding and music production, so this beginner was the most experienced one out of the VJ beginners.
- All the experienced VJ software users had performed several times and during several years as VJs. Three of them had used TouchDesigner for VJing and other visual projects before and one of them was very advanced in using TouchDesigner. One experienced VJ had not used TouchDesigner but was slightly familiar with it.

**2. Do you think that using this VJ program supports active learning (reflecting, analyzing, doing research, problem solving, developing new concepts etc.) compared to passive learning (just receiving information)?** Scale: 1 passive - 10 active.

- Total average **8.1**, VJ beginners **8.8**, VJ experienced **7.5**, TD beginners **7.5**, TD experienced **8.8**
- **Comments:** Whether the program supports active learning depends also on the user since they can choose not to edit the program and only use the standard interface, but active hands-on learning through exploration is nevertheless possible. Multiple respondents thought that the user should also have sufficient support, instructions, tutorials or prior knowledge to help the learning process.

**3. Does the interface make it clear how the visuals are created, organized, layered and controlled?** Scale: 1 not clear - 10 very clear.

- Total average **8.4**, VJ beginners **9.0**, VJ experienced **7.8**, TD beginners **9.3**, TD experienced **7.5**
- **Comments:** The interface was mostly understandable to the users, but some parts caused a bit of confusion. The third row section was not as clear as the first two rows, because all the subsections look similar. Using more signifiers like colors or other visual cues to indicate hierarchy and interconnection was suggested. The users did not always immediately recognize the final output screen so it could be emphasized more.

**4. Does the node view make it clear how the visuals are created, organized, combined and controlled?** Scale: 1 not clear - 10 very clear.

- Total average **8.1**, VJ beginners **8.8**, VJ experienced **7.5**, TD beginners **8.8**, TD experienced **7.5**
- **Comments:** For most users the structure of the node view was seen as well organized and clear but there was also some critique: Since every part of the project is inside a specific container, navigating inside the file could be tedious. The views of the operator nodes should be on, otherwise it is more difficult to understand the system. There should also be more comments and instructions inside the node environment and different parts of the system could be highlighted with colored background boxes. To understand the whole logic of the program takes time and needs explanation, especially for a beginner.

**5. Does this program give you any new creative ideas on how you would build your own visuals (in TouchDesigner)?** Scale: 1 nothing new - 5 some new ideas - 10 many new ideas.

- Total average **7.4**, VJ beginners **9.0**, VJ experienced **5.9**, TD beginners **7.8**, TD experienced **7.1**
- **Comments:** For most respondents, there were familiar elements but something new as well, for example how to work within the boundaries of TouchDesigner. For those who had never used TouchDesigner or VJ software, almost everything was considered new and demonstrated an example. One comment noted that experienced users of TouchDesigner may want to stay in their own habit of working instead of adopting a new approach. The interface was also considered good for experimenting with video clips to try out combinations in real time and creating a music video, for example.

**6. Does the interface resemble other similar programs or does it feel very different from them?** Scale: 1 different - 10 similar (what other programs?).

- Total average **7.1**, VJ beginners **7.0**, VJ experienced **7.1**, TD beginners **7.5**, TD experienced **6.6**
- **Comments:** Software that was mentioned included Resolume, VDMX, and Adobe programs like Photoshop, Lightroom, Premiere and After Effects. The slider names were familiar from other software but the three-row-structure of the interface was considered novel.

**7. Do you think there are enough features in the standard interface or do you run out of them quickly?** Scale: 1 not enough features - 10 enough features.

- Total average **7.9**, VJ beginners **9.5**, VJ experienced **6.3**, TD beginners **8.3**, TD experienced **7.5**
- **Comments:** The context of this question refers to making visuals for a basic event. The answers varied more in this question but mostly the users, especially beginners, found the standard features to be sufficient, except some specific effects were missing. The comments mentioned that for example duplicating the visual layers was not possible, but that there were enough import layers.

**8. Do you think you would be able to develop your skills and the program further? For example, create your own custom effects, generative visuals or other features?** Scale: 1 no possibility for development - 10 many possibilities for development.

- Total average **9.4**, VJ beginners **9.5**, VJ experienced **9.4**, TD beginners **9.1**, TD experienced **9.8**
- **Comments:** The answers were very positive and mentioned that the program is a good base and starting point that can be further customized. Using it would also save time since the user does not have to build everything from scratch. Although advanced users might want to develop the program beyond the limits of the current interface. Beginners of TouchDesigner stressed that they would first need to improve their knowledge of the system with the help of tutorials, instructions and guidance to develop the program.

**9. In the standard interface, do you have a lot of control in modifying the visuals or are the effects too predefined? Are you able to express your own style?** Scale: 1 too predefined - 10 lots of control to express my own style.

- Total average **7.1**, VJ beginners **7.6**, VJ experienced **6.5**, TD beginners **7.4**, TD experienced **6.8**

- **Comments:** The generative visuals were considered very predefined but also good examples that the user can use when they first start using the program. The standard interface might restrict the expression and control the user has, but on the other hand they have the possibility to customize and modify the interface if they choose to.

**10. Is the program interface suitable for beginners (easy to learn)?** Scale: 1 not for beginners - 10 great for them.

- Total average **7.9**, VJ beginners **7.8**, VJ experienced **8.0**, TD beginners **7.8**, TD experienced **8.0**
- **Comments:** Most respondents agreed that the interface is well-suited for novice VJs who have prior experience with similar creative software, such as image or video editing tools. Many of the terms used in the interface, such as 'blend mode,' 'hue,' and 'saturation,' are likely to be familiar to these users, while others may be less intuitive. However, some participants thought there should be additional guidance in the user interface, such as improved use of colors or a more intuitive layout.

**11. In the program, is editing the nodes in TouchDesigner suitable for beginners (easy to learn)?** Scale: 1 not for beginners - 10 great for them.

- Total average **5.6**, VJ beginners **6.5**, VJ experienced **4.8**, TD beginners **5.8**, TD experienced **5.5**
- **Comments:** Editing the VJ program with the nodes in TouchDesigner was definitely considered to be more challenging, complex and even intimidating for a beginner than using the interface. Due to multiple containers in different locations, navigating inside the file can be difficult. Learning TouchDesigner in general can be overwhelming, but for someone who is already familiar with visual programming and nodes it could be easier and faster. Learning to edit the program in TouchDesigner can be possible for a beginner and they do not need to know everything beforehand, since they can learn it gradually and with time, according to several answers. Exploring this project can actually help them in the learning process if they are willing and interested to learn. There should also be more instructions in the project file one recipient indicated.

**12. Is the program suitable for experienced users (enough features and complexity)?** Scale: 1 not for experienced users - 10 great for them.

- Total average **8.5**, VJ beginners **9.0**, VJ experienced **8.3**, TD beginners **8.5**, TD experienced **8.5**
- This question refers to both the interface and node editing in the VJ tool. An experienced user in this case is someone who is already

familiar with VJ software and TouchDesigner. The recipients who had no knowledge of VJing and TouchDesigner could not answer this question.

- **Comments:** All agreed that experienced users would know how to use and edit the program. One respondent noted that TouchDesigner projects are often constructed in different ways, meaning that even experienced users may encounter new approaches when working with files created by others. A few answers mentioned that advanced users might not need this program or they prefer to design entire systems from scratch. This program can in any case be advantageous for those experienced users seeking rapid development or experimentation, as it offers numerous predefined features that can be extended or customized as needed. Yet, one beginner VJ remarked that the program still appears to be more suited to novices than experienced users.

**13. Is the design of the interface visually pleasing?** Scale: 1 ugly - 10 very visually pleasing.

- Total average **7.5**, VJ beginners **7.4**, VJ experienced **7.6**, TD beginners **7.3**, TD experienced **7.8**
- **Comments:** The users mentioned that the design is nostalgic, boxy and perhaps could feel outdated to some people. Nonetheless, it is generally regarded as satisfactory, though not exceptional. Positive attributes include its simplicity, black-and-white color scheme, dark aesthetic, and the ability to customize the highlight color. Suggestions for improvement included increasing spacing between elements to create a more breathing layout, enhancing the visibility of the active button color, and repositioning the final output screen in the middle of the layout to make it more prominent.

**14. Are different elements organized and divided in a clear manner in the interface?** Scale: 1 messy - 10 organized.

- Total average **8.2**, VJ beginners **7.9**, VJ experienced **8.5**, TD beginners **8.0**, TD experienced **8.4**
- **Comments:** The interface was generally perceived as well-organized. However, some recipients suggested that the sliders could be divided into separate sections, as their current arrangement - stacked together and highly repetitive - can make locating a specific slider time-consuming. At the same time, they acknowledged that such a change might increase visual clutter. One participant commented that the purpose of the first two subsections in the third row (Import Layers and Generative Layers settings) was somewhat unclear. Additionally, the scrollbars and the ability to scroll were not always immediately noticed. Responses to question 3 also relate to this question.

**15. Can you click the targets (buttons, sliders, etc) quickly without errors or are they too small/big?** Scale: 1 too small - 5 just right - 10 too big.

- Total average **4.6**, VJ beginners **4.4**, VJs experienced **4.8**, TD beginners **4.8**, TD experienced **4.4**
- **Comments:** Some respondents noted that the sliders and buttons were slightly too small, making it difficult to click or grab them quickly. However, not all users experienced difficulties with the controls; for some, the size was appropriate, particularly since the slider knob width can be adjusted. Suggested improvements included increasing the size or prominence of the final composition opacity slider (given its importance), enabling users to zoom in and out of different parts of the interface, and allowing the expansion of specific sections for greater visibility. In addition, a common feature in other software - moving the slider knob by clicking elsewhere on the slider - was proposed.

**16. Is the standard interface too constrained/simple or complex or does it have good balance?** Scale: 1 too constrained/simple - 5 good balance - 10 too complex.

- Total average **5.1**, VJ beginners **5.0**, VJ experienced **5.1**, TD beginners **5.3**, TD experienced **4.9**
- **Comments:** The balance of the interface was generally deemed appropriate. One user found the limitations of the interface positive, noting that constraints help maintain focus by preventing the possibility of doing too much. For one experienced VJ, the interface was simpler than what they typically work with. On the other hand, two VJs found the interface complex and suggested that certain elements could be combined or contained more. One user remarked that there is extensive repetition.

**17. Do you like using the program?** Scale: 1 not at all - 10 very much.

- Total average **8.7**, VJ beginners **8.8**, VJs experienced **8.6**, TD beginners **8**, TD experienced **9.4**
- **Comments:** The feedback was largely positive, with four people giving the full 10 points. The users described the program as fun and enjoyable. One beginner preferred using the program for testing effects and combining videos to produce new material, instead of using it for live VJing. Critiques included the constraints imposed by computing power and the inability to modulate directly through the interface. One beginner wished for a simpler program.

**18. Is anything missing from the standard interface that you would like to add?**

- **Comments on Audio**: Two participants expressed a desire for the ability to filter audio input into low, mid, and high frequencies and to map the audio inputs to specific sliders, enabling any effect to be audio-reactive. Automatic BPM (beats per minute) detection and alternatively the possibility to manually tap the BPM was also desired.
- **Comments on Individual Effects**: Users made several specific requests related to effects, such as adding a blur effect to the generative 3D layer, allowing the inclusion of multiple 3D objects, and adjusting the speed of the slide effect.
- **Comments on Interface Elements**: One user proposed adding buttons to toggle a layer's opacity on and off or to perform fade-in and fade-out transitions. One significant suggestion was enabling the addition, deletion, and rearrangement of layers directly within the interface. Three participants expressed a strong interest in modulating and automating parameters by mapping them to inputs such as audio signals or LFOs, using drag-and-drop functionality or dropdown menus. Other proposals included a play and pause button, reset buttons, visible status indicators for each blend mode, OSC input, the capability to share outputs across multiple screens, resolution settings, and the functionality to record and export video material.

**19. Do you have any additional comments?**

- **Learning:** A potential user, particularly a beginner, has the option to rely solely on the standard interface without interacting with the TouchDesigner nodes if they find it too challenging. As they gain more experience, users can progressively deepen their understanding of the program and its node-based functionality, although this process requires time and effort. Editing manually through TouchDesigner can be beneficial, as it fosters a deeper understanding of the underlying processes. This VJ tool may lower the barrier to entry for VJing and serve as a stepping stone for beginners. There should be proper instructions and tutorials available on topics such as using the interface for VJing, implementing audio reactivity, and connecting a MIDI device.
- **Other:** When used with a high-performance computer, the program operates smoothly without issues related to performance or frame rate. However, for less powerful computers, a simplified and more optimized version could be beneficial. Unlike Resolume, which requires the purchase of a commercial add-on called Wire to build generative effects with nodes, this program offers this capability for free, making it a more convenient option for users.

## 4.2 Personal Evaluation

Compared to Prototype 1, I am very satisfied with the new version, Prototype 2. The updated interface is, in my opinion, more straightforward, clean, logical, and spacious, making it easier to learn quickly. One significant improvement is the ability to relatively easily add new sliders and effects to expand the program. However, I acknowledge that node editing remains complex and requires time for users to fully comprehend the system. Even I find editing the project file occasionally confusing. That said, the nodes in Prototype 2 are much better organized than in Prototype 1. While users can choose to work solely with the interface, those who wish to customize the program will likely require my guidance. This could involve providing either live instruction or comprehensive, detailed tutorials or documentation.

In December 2024, I used the new prototype at an event to create visuals to accompany music performed by DJs. This was the first time the new program was tested in a real-life setting. There were no issues with setting up the visuals, and I was able to perform VJing as usual. Controlling the visuals felt intuitive and seamless, although I encountered some technical issues that require fixing. For instance, there were scaling problems that caused unwanted image cropping, and some visuals became pixelated when scaled larger. During user testing, we also identified several system flaws, such as the audio sensitivity slider unintentionally affecting the volume of the uploaded audio file and disabling a layer causing issues with blend modes. While these issues are fixable and do not significantly hinder performance, they highlight areas for further refinement.

Although I am pleased with the progress made in the new program, I recognize that it still requires further development beyond fixing technical defects. While the VJ tool is functional in its current state, there are some key limitations that need future development. The two most important features I would add to the interface are: (1) the ability to add, remove, and rearrange layers directly, and (2) the capability to modulate parameters by mapping different input signals, such as audio or LFOs, to specific effects. At present, these operations can only be performed using the nodes and not through the interface. Incorporating these features into the interface would have been too advanced for me to implement within the scope of this thesis research. However, I believe that, with additional guidance and more time, I could achieve this in the future.

# 5 Discussion

Conducting the user tests offered a lot of useful feedback on the new VJ tool. The goal of the interview questions was to examine how well the thesis objectives were integrated in the design to provide answers to the research question. Calculating the mean averages of the quantitative answers of the user groups shows how well different aspects of the design were functioning. Since the averages are categorized between the beginners and experienced users of VJ software and TouchDesigner, it is possible to compare the answers to each other and to the total mean average. The user comments to the open-ended questions provide more context and reasoning to each rating, and clarify the challenges and strengths of the program. Next, I will analyze the data in depth in relation to the thesis objectives.

## 5.1 Discussing the Research Objectives

I analyze how well the thesis objectives were achieved based on the user test results and my personal reflection. Under each objective, I discuss the answers of respective user test questions including the mean averages of the scores and the user comments. I have added in parenthesis, which concepts from the literature review, such as the educational concepts (EC), design concepts (DC) or user experience concepts (UXC) are mentioned in the text. These approaches are discussed in Chapter 3.3.2. “Applied Design Objectives” in detail.

### Objective 1: be suitable for both beginners and experienced users

The total mean average of the answers to Question 10, whether the interface is suitable for beginners, is 7.9/10 and there are no major differences between the averages of the different users. This indicates a tentative positive result that the interface is accessible for beginners who are familiar with the basics of image manipulation and other creative software. The respondent comments to the question mention that some of the terms in the interface are recognizable while some may not be, which I expected. Even if the user does not understand a specific term in a slider, they can experiment to see what it does. To make the tool even more beginner-friendly, I would consider making the terms simpler and more descriptive. Other suggestions for improvement by the users are to add more guidance in the interface like better signifiers and use of color or more intuitive layout. I will most likely keep the structure of the current layout, but I could emphasize certain essential elements more with color for example.

In Question 11 I asked if the node editing in the program is suitable for beginners. Here the total average was much lower, 5.6/10. VJ beginners gave a slightly higher rating, 6.5, and experienced VJs gave the lowest score of 4.8. According to the comments, the node editing was challenging because TouchDesigner in general is a complex program with a steep learning curve. In addition, the multiple locations of different containers could be confusing. I agree that node-based programming is not for everyone and it requires previous knowledge on the topic. I hope that the program file can be advantageous for those people who are interested in learning TouchDesigner and have already started their learning journey. The participants mentioned that learning is gradual and possible with time and good instructions, so I aim to add more annotations in the file that guide the user. To help with file navigation, I could instruct the user either in person or record a video tutorial that presents the file structure.

Suitability for experienced users, comprising both interface and node editing, was rated 8.5/10 in total in Question 12, which is higher than the scores for beginners. Only two VJ beginners answered this question, since they could only speculate. In the comments the users state that the program is especially suitable for experienced users who are interested in rapid experimentation and development, because the program provides a solid foundation for that. I am pleased to hear this because achieving this was my goal. According to one comment, even experienced users encounter new approaches when examining someone else's file, so I aim to add better instructions in the file that help with understanding. The users also remarked that very advanced users might not need this program, because they want to develop their systems. This is fine, because these people would not be my target group anyway.

Based on the user test results, the VJ program is complex enough for experienced users and the interface is suitable for beginners, demonstrating a balance of simplicity and complexity (DC5). On the other hand, beginners might struggle with modification of the file in TouchDesigner although it can be possible with enough time and guidance. These insights reveal that if the program is used in educational settings, I will need to evaluate based on the skill levels of the students whether they should only use the interface or do node-editing as well. I am content with the results but want to implement some of the suggested improvements to make the tool truly Low Floor High Ceiling (EC3).

### Objective 2: help the user to understand the workflow and principles of how live visuals are created

Whether the program supports active learning (EC1) compared to passive is rated 8.1/10 in Question 2. There is some variation between the answers, for

example VJ beginners and experienced TouchDesigner users give an average score of 8.8 while experienced VJs and beginners of TouchDesigner rate it 7.5. According to the comments, it depends a lot on the user whether the learning will be active, but there is the possibility for it because the user can choose to actively examine and modify the project file if they want to and know how to. To encourage and enable the user to make edits, there should be enough instructions available, the comments mention. Using the interface should hopefully also provide means for actively learning live visuals performance, but the users do not comment on that. The results are in any case positive and indicate that active learning which helps the user to better understand the creation of live visuals is possible.

The interface clarity and understandability got a total score of 8.4/10 in Question 3. Beginners of both VJ software and TouchDesigner gave higher points 9.0 and 9.3 respectively. Experienced users gave scores of 7.8 (VJ ) and 7.5 (experienced). The users agree that the interface is mostly clear with a few exceptions such as difficulties in understanding the functionality of the third row and which preview is the output. These issues could be solved with better visual cues like distinct borders or highlight colors. Overall, the users have understood how the interface functions and how the visuals are created and controlled.

In a similar sense, in Question 4, the clarity and understandability of the nodes was rated 8.1/10. Again, the beginners give higher scores, 8.8 from both groups of beginners and 7.5. from both the experienced groups. This question highlights the clarity of the organization and control of the nodes in the project file and not node-based editing in general, meaning that even if a user does not understand node-based programming in TouchDesigner, they can still evaluate if the nodes are clearly structured. The average score is highly positive, but the critiques include difficulties in navigating the nodes due to separate containers, lack of instructions in the file, and lack of colored backgrounds to highlight different sections. I can add the two last mentioned suggestions, but I will still keep the container-based structure in the file, since the interface is built on this specific way of organization.

The user tests yield encouraging results that the program enables active learning, a constructivist approach to education (EC1), and provides an understandable interface and node structure for modification that support the creation of a correct mental conceptual model (EC2) for the user. Further improvement is still advisable.

### Objective 3: make it easier to move on to more complex VJ software

Comparing how similar the interface is to other software was rated 7.1/10 in Question 6. The average answers of different groups differ at most 0.5 points from the total average, so there are no major differences. Elements like the slider names and effects were familiar from other software, but the structure was considered to be novel. Making the program resemble other creative and live visuals programs was intended as a way to mitigate the adoption of this tool for beginners and to help the user to move on to other more complex VJ software in the future, based on Jacob's Law (UXC4). This way my VJ tool can work as a stepping stone for other programs like VDMX or Resolume as well. The results show that even if the program has a novel structure, it contains mostly familiar elements. Other elements that support the transition to more complex programs are the possibility for active learning (EC1) and communication of a clear system image to enable the creation of a correct system model (EC2) which were reviewed in the previous question and showed positive results.

### Objective 4: help the user to understand the internal logics of the node-based programming environment of TouchDesigner

As already mentioned in Objective 2, the clarity and understandability of the node structure is rated 8.1/10 with a little higher points from beginners (8.8) than experienced users (7.5), indicating an understandable system image to support the development of a correct system model (EC2) of the node structure. The adopted constructivist approach to enable active learning (EC1) was also rated 8.1/10, indicating a successful application of constructivist ideas, which was discussed previously in Objective 2 more in depth; the possibility to modify the program in the node environment brings about more understanding of the internal logics of TouchDesigner.

Question 5 asks if the program offers new ideas for the user about how they would build their own visuals, especially in TouchDesigner. The total average score is 7.4/10, where 1 refers to no new ideas, 5 to some ideas and 10 to many ideas. VJ beginners give a much higher score of 9. TouchDesigner beginners give a bit higher score as well, 7.8. The lowest average score comes from the experienced VJs who rate it 5.9 while experienced TouchDesigner users give a score of 7.1. The varying ratings might imply that experienced users already know more and thus the program does not present much new ideas. Providing new ideas and demonstrating an example (EC4) is a way to accelerate the learning process of the user. Clearly, this approach is more useful for novices than experienced users. The test results are promising that this VJ program

can work as a source of inspiration and accelerator for further understanding of node-based programming.

### Objective 5: provide possibilities for further skill development by offering extensible and customizable features

Question 8 is asking if the user feels they would be able to develop their skills and the program further by for example adding new effects or generative visuals (DC3). The general score is very positive with the average of 9.4/10. The averages of each group are all very similar, and the highest score, 9.8, comes from experienced TouchDesigner users. This demonstrates that the possibility for further skill development exists even for beginners, but it is up to the user if they choose to go down this path.

The TouchDesigner environment in itself provides this possibility for expansion and customization, but I argue that it is not enough. The project file needs to also be understandable and logical for the user. For example, the previous Prototype 1 was executed in TouchDesigner as well, but it would have been almost impossible for someone other than me to customize the file, because the organization was so chaotic and without any logical structure that editing that file was even complicated for me.

The integration of new features needs to be facilitated as well. I have done this by categorizing different sections distinctively and located the nodes in such a way that for example adding new layers, sliders, and effect is as straight-forward as possible. If the user wants to add a layer, they need to copy an existing layer and re-connect some parts such as the correct blend mode, so some amount of manual work is needed but it is not overly complicated. Copying and arranging sliders is easy as well, but the user needs to choose the right kind of slider, because there are a few different kinds. To create a new effect, the user needs to navigate the correct effects container and choose an appropriate part in the effects chain to add and connect a new TOP node for example.

This possibility of customizing the nodes enables active hands-on learning (EC1), which is agreed by the users, as described in Objective 2. It also supports the HIgh Ceiling philosophy (EC3) that more experienced users can improve their skills by working and modifying this program. Beginners can choose to work with the standard interface or gradually learn more about node-based programming and customize the program. All in all, the results and my experience with the tool indicate that this objective has been achieved.

### Objective 6: be a versatile and well-functioning VJ system for real time performance

Regarding versatility, the total average of the answers to Question 7 about the sufficiency of features in the interface is 7.9/10, which would imply that there are enough features for a basic VJ event. This question refers to the standard interface without any modifications. In this question, there is a lot of variability between the average score of experienced VJs (6.3) and beginner VJs (9.5). The answers of TouchDesigner beginners (8.3) and experienced users (7.5) vary a little bit but not as much. It is possible that beginners may not know what is a sufficient number of effects and features for a VJ performance. If we take the average score given by the experienced VJs, 6.3., we see that the standard interface is not yet as versatile as it could be and needs more improvement. On the other hand, since the program enables the user to add their own effects, the limits of the basic interface are not so severe. If the user wants a quick and diverse VJ tool that does not require any programming or customization, then this program would not be optimal.

In Question 9, I ask if the user has a lot of control in editing the visuals and expressing personal style through the interface compared to very predefined visuals and aesthetics (DC1). The total average score is 7.1/10 and the beginners give slightly higher scores of 7.6 (beginner VJs) and 7.4 (TouchDesigner beginners) while experienced users give lower scores of 6.5 (experienced VJs) and 6.8 (experienced TouchDesigner users). One of the most predefined aspects in the interface is agreed to be the generative visuals, but it is also acknowledged that they work well as examples. The generative visuals are an integral part of the interface, and I agree that they have a very predefined aesthetic, so there is a risk that if the user uses these visuals often, they create a repetitive and predictable visual style. However, the node editing enables the user to do any changes they want to and know how to make, even though the standard interface is restricting how much the user has control. That is why I have intended the generative visuals to work more as placeholders that the user hopefully would replace later.

One quality of a well-functioning VJ program is the ability to integrate other software and hardware with it (DC2). Unfortunately, these features were not separately tested in the user tests. However, based on my work with the previous Prototype 1 and its application on the design course, I was able to utilize the NDI protocol inside TouchDesigner to enable importing video streams from other software or other computers by using a common network. Integrating a MIDI device with several physical sliders and knobs was also successful, and I was able to perform as a VJ with a physical MIDI, as described in Chapter 3.2.1. This previous experience proves that this kind of integration

is possible in the VJ program, and I will later implement these features to the new prototype as well.

**Objective 7: have a clear and practical user interface**
I wanted to make an aesthetically pleasing user interface to support the user experience and clarity (UXC1) so Question 13 is about the visuality of the interface. The users answered 7.5/10 on average, 10 signifying a very visually pleasing interface. All of the groups answered in a similar sense so there are no major differences between the averages. According to this score, the aesthetic of the interface is found sufficiently pleasing. I am satisfied with this result, since even though the visuality of the interface is not the most essential part of the program, it is nevertheless important for the user experience. Improvement suggestions by the users that I want to implement later are the adding of more space between the interface elements and making the active button color more prominent, otherwise I would keep the visual design the same as it is now.

Question 14 is asking if different interface elements are categorized in a clear manner, including the understanding of what elements belong together and how they are divided (UXC2). The total average score is 8.2/10, 1 referring to messy and 10 to well-organized, and once again all the ratings of the different groups are close to each other. This demonstrates a positive result that the users understand the categorization and division of the interface. The different sections are confined and divided in an understandable way, although the repetitive stacking of the sliders and the similarity of the subsections on the third row are mentioned to be distracting. The highlighted color border might not be sufficient in itself to signify all the different sections and their relations in the interface and thus a better division might be needed. This could perhaps be achieved by using more colors or other visual signifiers, more subheadings, or even an altered layout.

In Question 15, I am asking about the size of the buttons and sliders and if they can be clicked fast without error (UXC3). In the numeric scale, number 1 indicates the controls are too small and 10 indicates they are too big while 5 signifies perfectly sized controls. The total average of answers is 4.6 and the answers of the different groups vary between 4.4 and 4.8. According to these results, the elements are on the most part correctly sized. The ability to adjust the width of the slider knobs certainly contributes to the positive results.

I also ask about the balance of constraint and control (DC4) in the interface in Question 16: 1 refers to too much simplicity and constraint, 10 on the contrary refers to too much complexity and 5 indicates a perfect balance. The average score is 5.1 and the group averages fluctuate between 4.9 and 5.3, so there is no notable difference between the groups. One user describes the

limitations as a positive element that limits the overwhelm of having too many options. These initial results indicate that the program is not too constraining or complicated for the user.

Finally, the average score of enjoying using the program is 8.7/10, with quite even answers from all groups, indicating a generally positive user experience in Question 17. The program was described as fun and great for experimenting with videos. The lack of modularity in the interface was one prominent critique that lowered the user experience. Based on all of the previously mentioned results that relate to this objective, it can be stated that the different elements, including the visuality, categorization, size of the controls, and the balance of constraint and control, succeed in making the interface clear and practical for the user, thus providing a positive user experience.

**Objective 8: work on different kinds of computers and operating systems**

The last objective was not directly asked in the user study, but the users had both Windows and macOS laptops to run the program, and in both cases the program could run without problems caused by the operating system. However, processing power will limit the possibility of available computers to run the program. I aimed to make the program file as optimized as I could, but it nevertheless requires a powerful graphics card to run smoothly with a high frame rate. The most effective way to optimize the program is to disable the unused layers, especially the generative layers, which can increase the frame rate substantially. Thus, the last objective is partially achieved but more work remains to be done on building a VJ tool that even less powerful hardware can handle.

## 5.2 Limitations of the Study

The user study provides some general feedback on the new prototype and its features, but there are several limitations on this study: It included only eight participants so the average scores are not sufficient for generalizations. For further research, more extensive studies should be conducted. Another substantive limitation is the short scope of the user tests. In order to truly study the learning processes of the users, there should be longer studies with multiple meetings and test settings. In further testing, the users could for instance perform with the VJ tool, modify and personalize it for their own needs, and work with it for an extensive period of time. These processes could then be observed to analyze if the users are truly able to develop their skills and perform with the program in various settings. Testing should also be conducted in real life learning settings like courses and workshops

Since I was mostly present with the testers, I was able to guide them in detail. This resembles a real-life learning situation such as in a school environment. However, there should also be tests where I am not present, and the user needs to learn the VJ tool on their own with provided instructions like tutorials. There is also a risk that some of the interview questions might be misinterpreted by the participants. Some questions cover several concepts that might not come across to the user. I have done my best to explain each question in detail for the participants in the test setting to avoid any confusion, but it is nevertheless possible.

The user test conducted in this research is not yet fully adequate to provide profound answers on the topic of learning live visuals. However, it provides an overview of the tool and was mostly sufficient for the aims of this research, providing initial positive results. For deeper inspection, long-term research in various real-life contexts is needed.

## 5.3 Summary

To answer the research question, "**How to design a VJ program that works as an educational tool that supports skill development and also works as a performance tool?**", I have defined eight research objectives for designing a practical VJ tool:

1. be suitable for both beginners and experienced/intermediate users
2. help the user to understand the workflow and principles of how live visual are created
3. make it easier to move on to more complex VJ software
4. help the user to understand the internal logics of the node based programming environment of TouchDesigner
5. provide possibilities for further skill development by offering extensible and customizable features
6. be a versatile and well functioning VJ system for real time performance
7. have a clear and practical user interface
8. work on different kinds of computers and operating systems (Windows & MacOS)

To meet these objectives, I have used existing educational and design concepts as a baseline for the design of the tool. The initial results from the conducted user test provide positive feedback that most of the objectives have been achieved. To summarize the findings of this study, here are the most relevant findings in relation to the research question:

## Educational Objectives

The interface is generally accessible for beginners, but improvements can be made by simplifying terms and making them more descriptive and adding guidance through visual signifiers, color cues, and layout adjustments. Beginners find node editing challenging, requiring prior knowledge of TouchDesigner. Suggestions to improve learning include adding annotations and clear instructions and providing tutorials (video or in-person) to explain file structure and navigation.

The program supports active learning, allowing users to explore and modify files actively to understand live visuals creation and workflow. Clear interface and well-structured nodes help users develop correct mental models, although some confusion arises from container-based organization and missing visual signifiers. Adding more instructions and background colors can address these issues.

Customization and extensibility scored highly, showing potential for skill growth: Logical structure makes adding new effects and layers straightforward and active learning through hands-on exploration supports skill development. The program provides new ideas for creating visuals, especially for beginners. However, experienced users benefit less, as the novelty is limited. Familiar elements help transition to more complex VJ software like Resolume or VDMX. Active learning and clear system images encourage understanding of advanced concepts.

## Performance and Design Objectives

The interface is versatile enough for basic VJ events, but experienced VJs rated it lower, indicating room for improvement in feature variety. Integrating other software and hardware, such as MIDI devices and NDI video streams, was tested successfully in previous iterations and should be expanded in future prototypes.

The ability to express personal style through the interface is moderate. The generative visuals, while useful as placeholders, limit creativity due to their predefined aesthetics. Node editing enables full customization for users with the skills to modify visuals. The program is capable of functioning in real-time performance scenarios, provided the hardware is powerful enough. Optimization (e.g., disabling unused layers) can help ensure smooth operation.

The program functions on both macOS and Windows, but its reliance on powerful hardware (e.g., a capable GPU) limits accessibility. Further optimization is needed to make the tool usable on less powerful devices.

The user interface is visually pleasing and well-organized, though improvements could include: adding more space between elements, enhancing active button colors, and refining section distinctions with better visual cues or subheadings. Control sizes are rated slightly below optimal, with slider width adjustability seen as a positive feature. Balance of constraint and complexity is deemed appropriate, limiting overwhelm while offering sufficient control. Overall enjoyment of using the program is high, with users appreciating its experimental potential. However, modularity is a desired improvement.

**Context of Design Research**

The iterative and interventionist nature of educational design research (Van der Akken et al., 2006) mirrors the process of developing the VJ tool. The research involves refining the software through feedback from real users (e.g., beginners and experienced VJs), ensuring that the tool addresses practical educational and performance needs. The practical focus aligns with Reeves' (2006) argument that real-life contexts, rather than isolated lab environments, are ideal for developing innovative learning tools.

# 6 Plans for Future

## 6.1 Future Applications

Like its predecessor, Prototype 1, the new VJ program is expected to be utilized during the bachelor-level VJing course at Aalto University in early 2025. While the primary software for the course will remain VDMX, its macOS exclusivity necessitates an alternative for students using Windows laptops. Although my VJ tool does not offer an interface which is as versatile and customizable as in VDMX, previous research and testing indicate that it is a competent solution for beginner VJs learning the fundamentals of live visual performance. In fact, even the earlier prototype was sufficient for the course's requirements. Given that the students are likely to be beginners, it is expected that they will primarily use the VJ interface rather than engage with node editing in TouchDesigner.

Beyond its application in the Aalto course, I have considered organizing workshops where participants could be introduced to this VJ tool and learn its functionalities. These workshops could be held in various contexts, such as festivals, residencies, or other events. Expanding the user base and gathering feedback from a diverse audience would provide valuable insights for further development. My aim is to continue refining the program and incorporating additional features that have been suggested during prior user tests. Future development could potentially be supported by a working grant and involve collaboration with other contributors. The following sections will outline plans for documentation, maintenance, and distribution of the completed program.

## 6.2 Documentation and Maintenance

If the program will be distributed to a broader audience, comprehensive user guidance will be essential. This may include written instructions, screenshots, or video tutorials. Until now, I have primarily been available in person to assist users with questions or issues. However, this level of support will not always be feasible, which means that accessible and clear instructional materials are necessary. Once the current version of the program is finalized, I intend to produce a concise PDF guide featuring illustrative screenshots and explanations of the program's main functions. Users will also need a basic understanding of TouchDesigner to operate the program effectively. Fortunately, there are numerous online resources and tutorials available to support this learning process. If the VJ tool undergoes significant further development, I would consider producing simple online video tutorials as well.

In terms of maintenance, my involvement will likely be minimal unless the program is being used for specific purposes, such as a course or workshop. In such cases, I will ensure that the current version operates smoothly on both macOS and Windows systems and make any necessary updates or fixes. If the program is distributed online for individual users, I would not commit to providing regular updates. Nonetheless, the program is designed to be robust, relying solely on basic TouchDesigner nodes and avoiding imported add-ons, external code libraries, or complex Python scripts. Any technical issues that arise should be relatively straightforward for an intermediate TouchDesigner user to troubleshoot within the project file. Additionally, if a user contacts me for technical assistance, I would be happy to offer support.

## 6.3 Distribution: Open Source or Commercial

A critical consideration for the program's future development is determining whether it will be distributed freely for use, modification, and redistribution or whether there will be restrictions or commercial objectives. This decision remains unresolved at this point. Given that the program was initially developed for educational purposes and aims to provide an accessible option for beginner VJs, it will include at least some degree of openness. For example, if the program is used in a course or workshop that I facilitate, it will always be free for the students. Furthermore, those with access to the program will also have the TouchDesigner project file available for inspection and modification. This transparency will enable users to learn from and build upon the program rather than restricting them from accessing its inner programming.

However, I am unlikely to release the program under a public domain CC0 license, which would renounce all copyrights and allow unrestricted use, modification, and resale without attribution (Creative Commons, n.d.). Instead, if I opt for a Creative Commons license, it would likely be CC BY-NC-SA, which permits modification and distribution under the same licence for non-commercial purposes while requiring credit to the original creator. This would ensure that users can access the program for free, customize it to their needs, and share further adaptations with the community. At the same time, it would prevent commercialization of the program by others and require that all further adaptations be shared under the same license. Requiring attribution would also guarantee that interested users could contact me directly.

If the program undergoes substantial development and improvement, I might consider selling it at a modest one-time cost. Even in this scenario, the price would remain low, given the limited scope for regular updates. Identifying potential customers could perhaps be managed through online platforms or personal networks. Should the program be commercialized, it would be accompanied by an End User License Agreement (EULA), a contract

between the owner and user of a software, to define the legal boundaries of use, including for example conditions for updates, liability, copyright, and usage restrictions (The Interactive & Immersive HQ, 2019; What is EULA?, 2024).

Ultimately, the decision between a commercial or Creative Commons license will depend on the extent of future development. The current version is unlikely to be sold, but a more advanced iteration with additional features could be marketed. One possibility is selling the program and its rights to the university if they wish to integrate it into more courses. Regardless of the licensing model, my primary goal is to utilize the VJ tool in educational contexts, enabling learners of different experience levels to achieve their objectives in live visual performance.

# References


Alexander, A. (n.d.). **Audiovisual live performance**. *See This Sound Compendium*. Retrieved November 5, 2024, from http://www.see-this-sound.at/compendium/abstract/54.html

Amagi, T. (2018, September 4). **VJ-ing with Unity**. *Medium*. Retrieved December 29, 2024, from https://medium.com/@amagitakayosi/vj-ing-with-unity-e9736793e979

Carvalho, A., & Lund, D. (2015). **The audiovisual breakthrough**. *Fluctuating Images*, 126–128.

Correia, N. N., & Tanaka, A. (2014). **User-centered design of a tool for interactive computer-generated audiovisuals**. In A. Sá, M. Carvalhais, & A. McLean (Eds.), *ICLI 2014 Inter-Face: International Conference on Live Interfaces* (pp. x–x). Porto, Portugal: Porto University, CECL (NOVA University), CESEM (NOVA University), & MIPTL (University of Sussex). Retrieved from https://users.sussex.ac.uk/~thm21/ICLI_proceedings/ICLI-2014-Proceedings.pdf

Creative Common. (n.d.). **About CC licenses**. Retrieved December 29, 2024, from https://creativecommons.org/share-your-work/cclicenses/

Crevits, B. (2006). **The roots of VJing: A historical overview**. In M. Faulkner & D-Fuse (Eds.), *VJ: Audio-visual art + VJ culture*. Laurence King Publishing Ltd.

Dornbierer, Y. (2009). **From video games to audiovisual instruments**. In C. Lund & H. Lund (Eds.), *Audio.Visual – On Visual Music and Related Media* (pp. 238–239, 244–245). Stuttgart, Germany: Arnoldsche Verlagsanstalt GMBH.

Fosnot, C. T. (1996). **Constructivism: Theory, perspectives, and practice**. New York, NY: Teachers College Press.

Goodfellow, P., & Gibson, S. (2023). **The post-conceptual digital era (2000–present)**. In S. Gibson, S. Arisona, D. Leishman, & A. Tanaka (Eds.), *Live visuals: History, theory, practice* (pp. 110–112, 124).

Groove Atelier. (2024). **The artistic vision of Alessio De Vecchi: Anyma's creative force**. *Groove Atelier Crew*. Retrieved December 29, 2024, from https://www.grooveatelier.com/read/the-artistic-vision-of-alessio-de-vecchi-anymas-creative-force

Hook, J., et al. (2011). **A VJ centered exploration of expressive interaction**. In *CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1269–1270, 1273). New York, NY: Association for Computing Machinery.

Kreukniet, M. (n.d.). **What is a VJ?** *Passion for EDM*. Retrieved November 1, 2024, from https://passionforedm.com/blog/what-is-a-vj/

Levin, G. (2000). **Painterly interfaces for audiovisual performance** (Master's thesis). Massachusetts Institute of Technology, Department of Architecture, Program in Media Arts and Sciences.

Max. (2024). **Cycling '74**. Retrieved December 29, 2024, from https://cycling74.com/shop/max

McCarthy, L., & Gibson, S. (2023). **Scratch video and rave: The rise of the live visuals performer (1985–2000)**. In S. Gibson, S. Arisona, D. Leishman, & A. Tanaka (Eds.), *Live visuals: History, theory, practice*, pp. 96, 103, 106

Modul8. (2017). *GarageCube*. Retrieved May 9, 2024, from http://www.garagecube.com/modul8/

Norman, D. A. (2013). **The design of everyday things: Revised and expanded edition**. Cambridge, MA: The MIT Press, pp. 25–33.

Notch. (2024). **Notch Builder**. Retrieved December 29, 2024, from https://www.notch.one/pricing/

NRICH. (2019). **Low threshold high ceiling – An introduction**. *NRICH Team, University of Cambridge*. Retrieved May 7, 2024, from https://nrich.maths.org/10345

Preece, J., Rogers, Y., & Sharp, H. (2019). **Interaction design: Beyond human-computer interaction** (5th ed.). Indianapolis, IN: John Wiley & Sons Inc., p. 281.

Reas, C., & Fry, B. (2019, May 29). **A modern Prometheus**. *Processing Foundation, Medium*. Retrieved May 7, 2024, from https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85

Reeves, T. (2006). **Design research from a technology perspective**. In J. van den Akker et al. (Eds.), *Educational design research*. London, UK & New York, NY: Routledge.

Resnick, M. (2020, June 3). **Designing for wide walls**. *Medium*. Retrieved May 7, 2024, from https://mres.medium.com/designing-for-wide-walls-323bdb4e7277

Resolume Wire. (n.d.). **Resolume Wire**. Retrieved December 28, 2024, from https://www.resolume.com/software/wire

Russell, D. (1995, January 14). **Vujak heralds era of video sampling: Editing tool to have pro, consumer applications**. *Billboard Magazine*. Retrieved November 7, 2024, from https://books.google.com/books?id=tQsEAAAAMBAJ&dq=vujak+magazine&pg=PA32

Schifter, C. C., & Stewart, C. M. (2012). **Technologies and the classroom come to age after century of growth**. In C. M. Stewart, C. C. Schifter, & M. E. M. Selverian (Eds.), *Teaching and learning with technology: Beyond constructivism*. Taylor & Francis Group.

Spinrad, P. (2005). **The VJ book: Inspirations and practical advice for live visuals performance**. Los Angeles, CA: Feral House.

Taylor, R. (2022, June 20). **Live coding**. *Medium*. Retrieved December 29, 2024, from https://medium.com/@royce.taylor789/live-coding-6ac7ead9dbcf

The Interactive & Immersive HQ. (2019, December 21). **Licensing your TouchDesigner work**. Retrieved December 29, 2024, from https://interactiveimmersive.io/blog/touchdesigner-business/licensing-your-touchdesigner-work/

The Node Institute. (n.d.). **Introduction to visual programming with TouchDesigner**. Retrieved December 29, 2024, from https://thenodeinstitute.org/courses/introduction-to-visual-programming-with-touchdesigner/

TroikaTronix. (2024, August 7). **Pricing for Isadora 4 and Izzycast**. Retrieved December 29, 2024, from https://support.troikatronix.com/support/solutions/articles/13000106528

Van der Akker, J., Gravemeijer, K., McKenney, S., & Nieveen, N. (2006). **Introducing educational design research**. In J. van den Akker et al. (Eds.), *Educational design research*. London, UK & New York, NY: Routledge.

Walker, D. (2006). **Toward productive design studies**. In J. van den Akker et al. (Eds.), *Educational design research*. London, UK & New York, NY: Routledge.

What is EULA? (2004). *ServiceNow*. Retrieved December 29, 2024, from https://www.servicenow.com/products/it-asset-management/what-is-eula.html