UNIVERSITY OF CALIFORNIA
SANTA CRUZ

THE STRUGGLES OF LEARNING FEATURE-RICH ART SOFTWARE AND STRATEGIES TO HELP

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTATIONAL MEDIA

by

Andrew M. Dunne

June 2024

The Thesis of Andrew M. Dunne
is approved:

Dr. Edward Melcer, Chair

Dr. Adam Smith

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

The Struggles of Learning Feature-Rich Art Software and Strategies to Help

by

Andrew M. Dunne

Today digital art is important as both a hobby and a career, taking many forms from images to animations to 3D models. Creating unique digital art requires feature-rich art software, which can take dozens of hours to learn even the basics of, deterring many aspiring artists. This thesis begins by exploring art software difficulty by examining peoples' learning struggles with feature-rich 3D modeling software through a survey, identifying themes of struggle and comparing them to education and software design theories. Turning its attention from people to software, this thesis then analyzes existing art software and video game tutorials to create a taxonomy of features that art software and games usually have - comparing features of often successful modern video game tutorials with features of often failing art software tutorials. Finally the best tutorial strategies for feature-rich art software are identified by analyzing how previously seen strategies alleviate the learning struggles of artists.

This work is dedicated to my parents for being caring, open-minded, and fun,

and to all of the great friends I've made in the computational media program.

...and to my brother Jason, I guess.

Acknowledgments

Thank you to my advisor Eddie Melcer for accepting me as a student, answering my research confusions, and giving feedback on this thesis, and Adam Smith for his feedback on this thesis and letting me work out of his lab room (in which I've spent hundreds of hours probably). Thanks to Amelia K, Suchir V, and Vivian N for their great work on Gamelike Tutorials for Art Software, and Matthew T for offering to help with the 3D modeling survey. This work wouldn't be complete without Crystal, the CM graduate advisor, who kept me on track, or the 'study' group consisting of Rohan, Jack, and I where we would meet each week to lament the work we hadn't done. Finally I would like to thank my housemate and good friend Kyle for providing thoughtful feedback and listening to my thesis rants.

Part I

Thesis

Chapter 1

Introduction

When was the last time you opened a painting program? Maybe it was Microsoft Paint for a quick doodle, or you opened Photoshop to touch up some photos. Painting programs promise a lot; they can theoretically produce any image whether it's an elaborate scene or a few idle brushstrokes. Painting programs provide tools for modifying their canvas; Microsoft Paint has a handful of brushes, a color picker, shapes, and a few other tools (Fig 1.1). Photoshop on the other end has so many tools, filters, commands, et cetera that listing them here could fill out the rest of this thesis. Because of Photoshop's versatility in tools, which allow users to quickly create and transform images, Photoshop is considered a feature-rich program (and Microsoft Paint is not). The versatility of feature-rich software like Photoshop or Blender (Fig 1.2) has made them extremely popular with millions of users each [20] [36], and feature-rich software are considered standard to learn both for beginners and professionals in many fields.

Feature-rich software let their users work quickly - in theory. However users

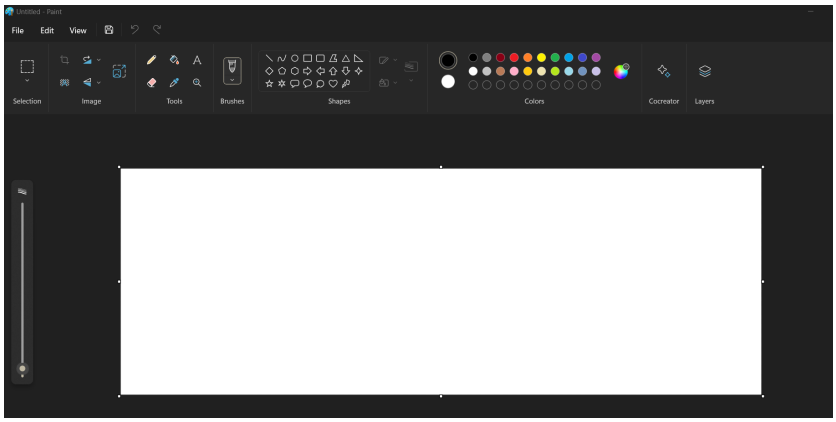

Figure 1.1: Microsoft Paint, a simple painting application, has most of its features visible in the top toolbar.

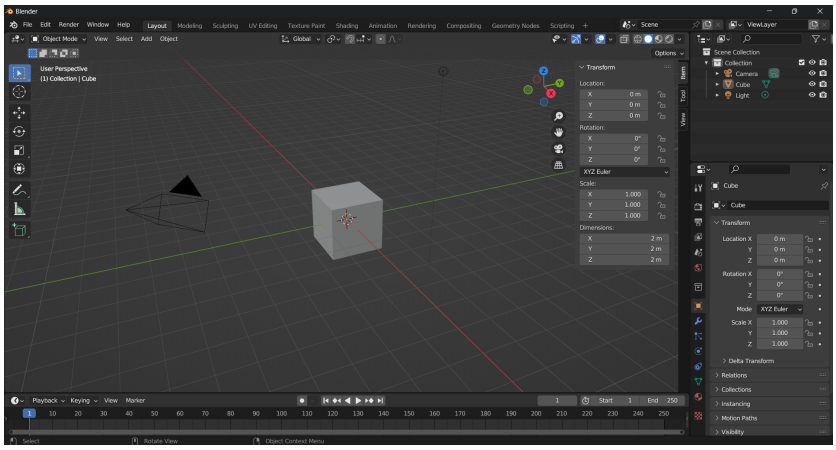

Figure 1.2: Blender, a complicated 3D modeling software, has many menus, modes, tools, etc available on-screen.

must first explore the software's features and fit them into their art-making practice. Beginners struggle to perform even basic tasks because the tools they want are hidden behind hierarchies of menus [37], and even users with years of experience still learn new things about their program every day. It takes a lot of patience for beginners to acquaint themselves with feature-rich software which often leads to frustration and defeat.

Feature-rich software require extensive education to use confidently, so it would only make sense for them to provide that education. However, most feature-rich software approach education as an afterthought, providing only a brief tutorial if any at all. Most commonly they rely on users to educate themselves through second-hand sources such as online videos or forums. Unless they're in a structured educational setting like school, new art software users must be both internet savvy and good at self-regulated learning to succeed [38].

The initial hurdle of learning feature-rich art software leads many aspiring artists to give up. Why does it matter that people are giving up on digital art? Digital art is involved in many big industries in America including gaming, film, architecture, and graphic design, so potential artists in these fields are lost. Digital art education is poor right now and better education has long shown to be connected to economic growth [24]. People who are barred from digital art also can't contribute as well culturally to online art platforms like social media.

To understand more closely what these aspiring artists struggle with, the first study of this thesis surveys 3D modeling software users about their learning experiences and then thematically analyzes their responses [18]. The second study of this thesis

then explores ways to help the onboarding process by comparing well-designed tutorials in video games to the poorly designed ones in feature-rich art software. Finally in the third section of this thesis I review recent developments in art software education, drawing on art tutorial designs from academia along with tutorials in small art software to see what's been done before and determine which tutorial strategies might be effective for feature-rich art software. Ultimately I propose two strategies that I think all feature-rich art software should implement to improve their learnability: UI scaling and obvious foundational tutorials. Even beyond those strategies, I hope this research informs and motivates art software designers to improve their programs' learnability, and for academics the study data inside may be useful.

1.1 Positionality and Motivation

I identify as a privileged and tech-literate artist. The political goal of this thesis is to identify flaws in digital art education in order to distribute the privilege that comes from techno-literacy and make a healthier digital art landscape. "If the self-proclaimed digiterati refuse to promote access and include the dispossessed, their rhetoric of revolution through technology will seem nothing more than the callous justification for unregulated accumulation of personal wealth." [41]

1.2 Research Questions

The main problem this thesis aims to address is; **why is feature-rich art software hard to learn, and what can be done about it?** This is a broad problem, so some questions this thesis answers on the way are listed below.

1. *RQ1: How do people learn to use art software?*
2. *RQ2: What do people struggle with when learning art software?*
3. *RQ3: What strategies do feature-rich art software use to teach their users?*
4. *RQ4: What strategies should popular art software use to improve learnability?*

1.3 Studies

To answer the aforementioned research questions, I conducted two studies:

The *3D modeling survey* is a short online survey sent out to 3D modeling forums online in May 2023. Over the course of 1-2 months 75 respondents of varying expertise in 3D modeling completed the survey and described their experience and struggles with 3D modeling and digital art. The goal of the 3D modeling survey was to understand 3D modelers' learning techniques and struggles (RQ1 & RQ2).

Gamelike Tutorials for Art Software is a taxonomy revealing common features in tutorials of art software vs video games in order to see what video game tutorial techniques might be applied to art software. 50 art software and game tutorials were analyzed for their features in order to build the taxonomy, and following discussion

considers why these differences exist and how they can inform art software tutorial design. The taxonomy aims to answer RQ3, and the discussion that follows aid in answering RQ4.

Chapter 2

Background

This thesis is informed by previous research in education, software design, and related fields. Below are some of the topics important to this thesis.

2.1 Digital Competence

Digital competence describes how competent a person is with digital technologies, including skills such as "ICT skills, technology skills, information technology skills, 21st century skills, information literacy, digital literacy, and digital skills" [32]. This includes art software skills, as using art software involves navigating computers and interpreting user interfaces. Making digital art requires high digital competence firstly because art software is often complex, and secondly because art is already difficult so digital artists need to be very comfortable with computers in order to navigate them and make art at the same time [25]. Without sufficient digital competence, artists grow frustrated at their tools which can lead to lost time [27], or worse yet, may believe they

cannot make what they desire and quit [2].

Although this thesis does not study inequalities in digital competence directly, it is important to understand them as political context for this work. Studies have shown that digital competence varies in the US based on race, age, and education, with the issue of marginalized populations being less digitally competent. Black and Hispanic populations in the US are on average less digitally competent [25] [42] as well as older adults and those with less education [42]. Lower digital competence hinders these populations from engaging with culture [35] and industry [26]. "Compared to digitally literate adults, adults who are not digitally literate have a lower rate of labor force participation and tend to work in lower skilled jobs" [42]. Lack of access to good education and training for digital skills, such as digital art creation, is a contributor to the inequality in digital competence [25]. This thesis aims to build access to good digital arts training, and thus alleviate the inequality in digital competence.

2.2 Feature-Rich Software

Computer applications vary in complexity from simple tools like the Notepad text editor to more powerful programs like Microsoft Word. More complicated software like Word tend to provide more options/tools/features to their users, and are thus called 'feature-rich'. Feature-rich software exist in a number of domains including art, product design, task management, word processing, etc.

In order to support hundreds if not thousands of features inside a single appli-

cation, feature-rich software most commonly use a hierarchical structure of menus and sub-menus [37]. This hierarchy lets users access any feature in only a few clicks, however difficulty lies in knowing where to find a feature within the maze of menus [37]. Feature-rich software have employed various methods to help users find the features they need including search bars (Fig 2.1), hotkeys (Fig 2.2), pie menus (Fig 2.3), panes (Fig 2.4), sidebars (Fig 2.5), icons (Fig 2.5), and workspaces (Fig 2.6). but there is still room for improvement as strategies like command recommendation are still controversial [58] and people continue to struggle with mastering feature-rich art software [33].

The wealth of features available in feature-rich software makes them powerful yet difficult to learn. In response to this, academics have studied software learnability, examining design practices that help new software users become competent and developing methods for evaluating software learning effectiveness [23]. Practices of software learnability studies include UI design [37] [44], tutorial design [15] [13], user studies [23], and interviews [33]. UI design for software learnability includes making changes to software such as managing visible feature count, reducing jargon, allowing customization, and adding icons for easier use and thus learning [10] [48]. Tutorial design, which Gamelike Tutorials for Art Software dives into later in this thesis, studies resources that teach users how software works at a basic level [45] such as manuals [45] and tutorial videos [47], and aims to improve them through experimental design [15] and drawing on theories of multimedia learning [43]. User studies in software learnability test how effective certain teaching strategies are while interviews can explore what processes people actually go through while learning to use feature-rich software.

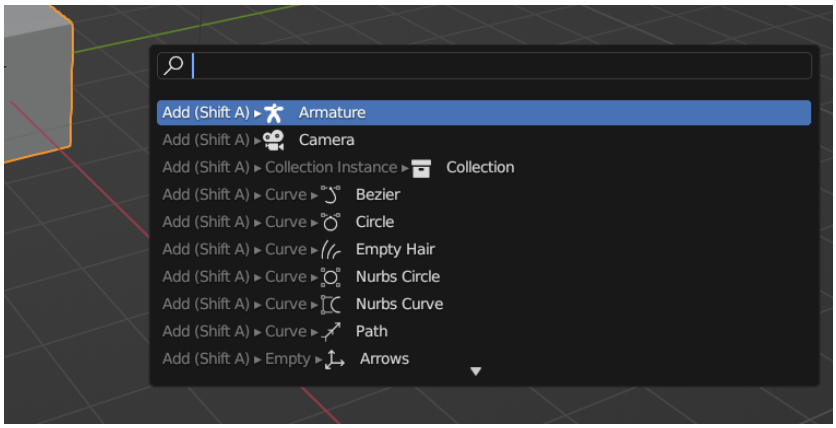

Figure 2.1: Blender's command search bar.

This thesis examines some of the areas that feature-rich *art* software in particular can improve. Although feature-rich software and digitally-created art are decently studied, the particular intersection of feature-rich art software is not. A few studies in the field of creativity support tools [51] examine feature-rich art software through examining how artists use and create tools [39] or by creating their own feature-rich art software [31] but there is little work to be found about commercial feature-rich art software that most people use, much less how those those programs in particular are taught.

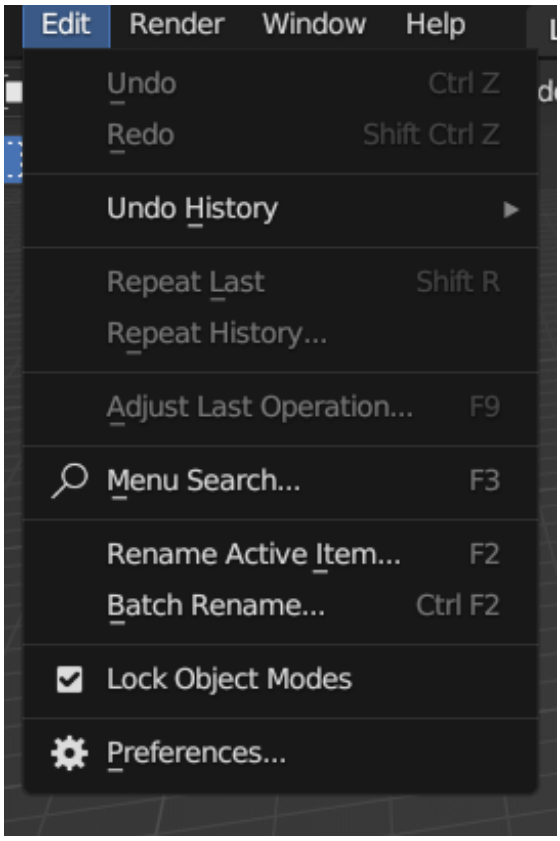

Figure 2.2: Some hotkeys listed beside Blender's 'edit' menu commands let users quickly perform those commands.

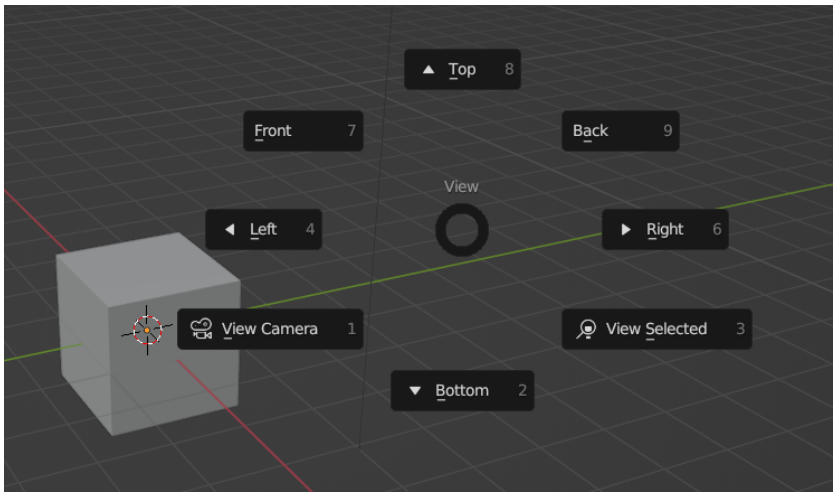

Figure 2.3: A pie menu in Blender to quickly pick viewing angles.

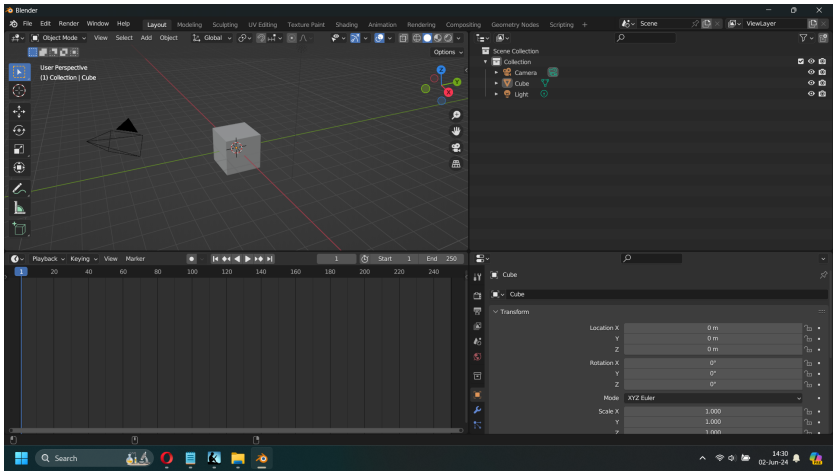

Figure 2.4: Blender's default layout is split into 4 panes.

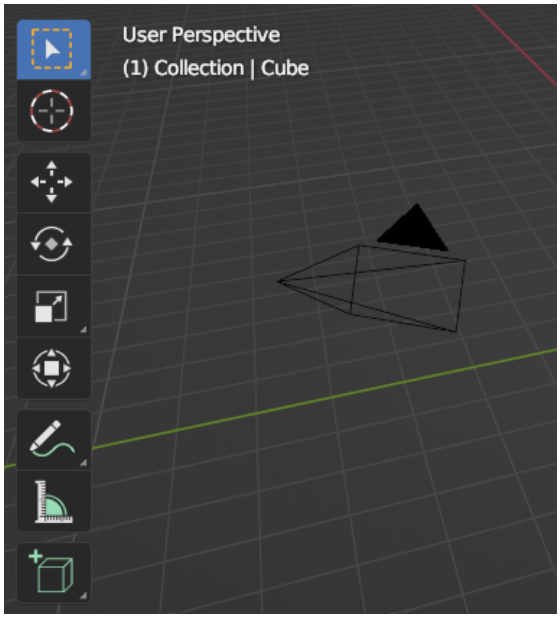

Figure 2.5: The tool sidebar of Blender's viewport, with colorful icons.

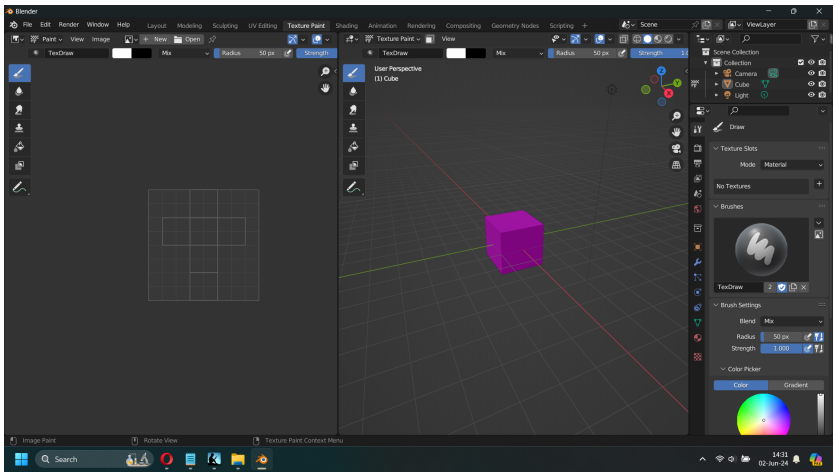

Figure 2.6: Blender's texture painting workspace lays the UI out in a better way for texture painting.

Chapter 3

3D Modeling Survey

3.1 Overview

The 3D modeling survey was a short survey sent out to 3D modelers of various skill levels in order to understand their experiences while learning 3D modeling software. The goal of the survey was to answer the question: "What is stopping people from learning 3D modeling?" The survey had three rating scale questions and two free response questions, and most of the insights of the study come from thematic analysis of the free response answers.

3.2 Related Work

3.2.1 3D Modeling

A lot of digital art is made through a technique called 3D modeling, which involves creating 3D scenes on a computer. 3D modeling began in the 1960s with CAD

(computer aided design) software [6], which is used to design objects virtually before making them physically. 3D models became popular as an art form as rendering and texturing tools developed into the 90s with games like I Robot incorporating realtime 3D graphics and the release of CGI films like Cassiopeia and Toy Story.

To create and edit 3D models, artists use tools provided by 3D modeling software. The most popular modeling software include programs like Maya, Blender, and Sketchup, which let users summon basic geometric shapes like cubes and spheres and then transform them into more complicated objects. Popular modeling software mostly use geometric modeling, which means users interact with the vertices, edges, and faces of objects to edit them. Geometric modeling skills transfer well between modeling programs, but modeling is a very uniquely digital craft that's hard to train outside of modeling software.

3.2.2 Online Learning

Many 3D modelers (and respondents to the 3D modeling survey) learned/are learning 3D modeling not through formal education or mentorship but from online resources. The internet has become a place of education for many people, and this phenomenon has been studied by academics under the term 'online learning' [52]. Unlike structured learning settings like school, online self-regulated learning requires from students self-efficacy and a strong commitment to their goals in order to succeed [38].

3.3 Methods

The data for this study was collected through a survey sent out on May 17th, 2023 and closed on July 11th, 2023. The survey was sent to the public 3D modeling and/or digital art forums seen in Table 3.1. 3D modelers of all skill levels were invited to respond including those who hadn't started learning yet. Besides skill level, common participant demographics (age, race, gender, etc) were not recorded so they are unknown - something worth doing for future studies. The reason 3D modeling and digital art forums were selected is because their members are likely to be actively learning 3D modeling, so their responses will be from recent experience. These forums also contain many self-regulated online learners, who are the ideal audience of this study. Over the duration this survey was active there were 111 responses, of which 36 were removed for being blank leaving 75 real responses. The survey questions can be seen in Table 3.2.

The results of the rating scale questions can be seen below, and the free response questions were assigned codes from which themes were built following Charmaz's constructivist grounded theory [7]. In particular the *initial line-by-line coding* strategies suggested by Charmaz were followed, but in a looser response-by-response style. Memos were developed by expanding on the codes and digging deeper into the meaning of each response. Instead of continuing the grounded theory process with iterative data collection and analysis, my analysis ended at the memo stage and overall five themes were developed from these memos.

Forum Name	Type
r/blenderhelp	subreddit
Polycount forum	forum
blenderartists.org	forum
Art Club	discord server
3D modeling	discord server
Blender community	discord server
Art commissions	discord server
Artists corner	discord server
Blender discord	discord server
Autodesk maya	discord server
Autodesk 3ds max	discord server

Table 3.1: Online locations the survey was sent to.

Question	Response Format
How experienced are you with 3D modeling?	Rating scale (haven't tried) (just started) (novice) (experienced) (professional)
How would you describe your experience of learning 3D modeling?	Free response
How experienced are you with digital art programs? (Photoshop, iMovie, Krita, etc)	Rating Scale (haven't tried) (just started) (novice) (experienced) (professional)
How difficult do you find 3D modeling difficult compared to other types of art?	Rating scale (significantly easier) (slightly easier) (about the same) (slightly harder) (significantly harder)
What impedes you from learning 3D modeling?	Free response

Table 3.2: Survey questions

3.4 Results

This survey resulted in some charts from the rating scale questions, and themes identified from the free response questions.

3.4.1 Data

The first question, "How experienced are you with 3D modeling?". Most participants rated themselves as novices, experienced, or professionals (Fig 3.1).

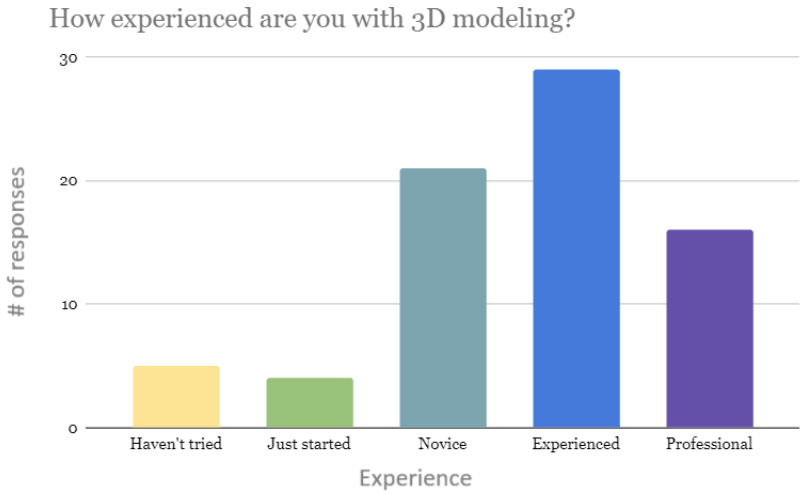

Figure 3.1: Responses to the first multiple choice question, "How experienced are you with 3D modeling?"

The second question, "How experienced are you with digital art programs?", had very similar results to the first question (Fig 3.2). The respondents of this survey mostly have some experience with art software, however many of them still consider themselves novices.

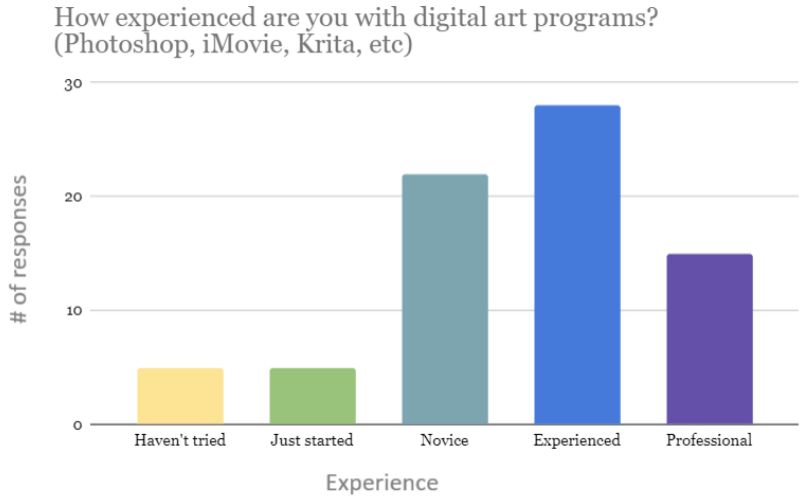

Figure 3.2: Responses to the second multiple choice question, "How experienced are you with digital art programs? (Photoshop, iMovie, Krita, etc)"

The third question, "How difficult do you find 3D modeling difficult compared to other types of art?". There is a slight trend towards respondents considering 3D modeling to be more difficult than other types of art (Fig 3.3).

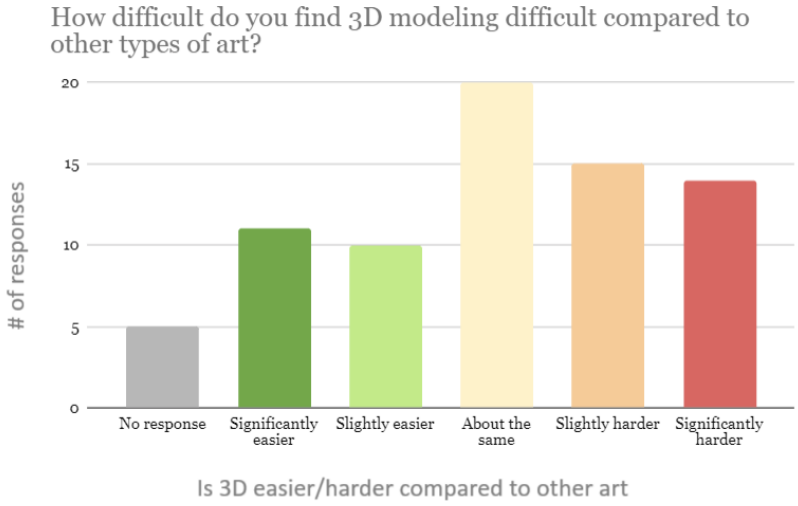

Figure 3.3: Responses to the third multiple choice question, "How difficult do you find 3D modeling difficult compared to other types of art?"

Figure 2.4 is a heatmap comparing respondents' answers from question 1 (Fig 3.1) with question 3 (Fig 3.3). From this heatmap we can see that people who are less experienced with 3D modeling consider it a more difficult form of art, whereas more experienced/professional 3D modelers find modeling to be about as difficult if not easier than other forms of art.

3.4.2 Themes

In answers to the survey's free-response questions, I identified five common themes. Respondents had a variety answers about their experiences and struggles learning 3D modeling, from motivational issues to complaints about learning material to personal histories, which I have distilled into the five themes below.

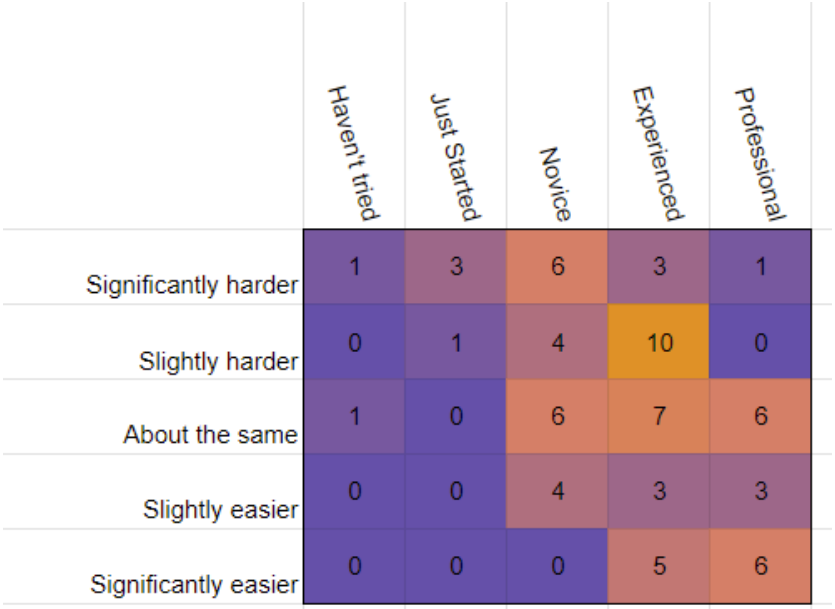

Figure 3.4: How difficult people thought 3D modeling was (compared to other art forms) vs. their experience with it.

3.4.2.1 The Learning Curve

Experienced modelers describe learning 3D like a hill; it's hard in the beginning, but once you learn enough and become proficient, it becomes less frustrating and more fun. They enjoy modeling and say that they may have struggled in the beginning but that the struggles paid off. The experienced modelers are still learning, but they enjoy it. The following response summarizes the journey of experienced modelers well: "The introductory period was difficult as I was self learning, but over time I was able to learn and eventually get a professional grasp of the topic."

Newer modelers on the other hand are overwhelmed by the amount of things they need to learn, are unable to find good resources, and lament the endless ways things can go wrong while modeling. The learning journey is steep, full of splits and

bends, with no end in sight. The exact phrase 'learning curve' came up 4 times in people's answers who identified as 'novice' and 'haven't tried', an example being: "[my experience with 3d modeling was] Hard. Initial learning curve is very steep. Required knowledge seems vast and it is difficult to navigate through all the material to find a pathway to suit your own needs/desires."

The 3D modeling learning curve is seen in figure 2.4 as well. Newer 3D modelers find 3D modeling to be much more difficult than other forms of art, but as people become more experienced with it they find it to be about as difficult as other forms of art if not easier.

3.4.2.2 A Bumpy Road

Several modelers describe the ups and downs of learning modeling. "bumpy", "Interesting, fun, sometimes little bit frustrating," "overcoming speedbumps," "Bumpy road," "stop-and-start". Respondents don't elaborate on what they mean, but a significant number of them described their modeling experience this way.

3.4.2.3 Learning Art is the Hard Part

A couple types of responses point to the idea that, although 3D modeling software can be difficult to learn, the real challenge is learning how to make good art. Anyone can learn modeling tools with some time and effort, but the larger struggle is with improving one's art - a fact which isn't helped by the lack of artistic learning content for 3D modeling. One respondent describes this struggle precisely: "Initially it

was the software primarily [that impeded me]. Now as I progress it [is] the theory that is highly helpful."

3.4.2.4 Controversial Online Tutorials

A lot of people learn from online tutorials, with 13 mentions of 'tutorial', 7 mentions of 'video', and 6 mentions of 'online'. Respondents make a variety of judgements of online tutorials. A repeated sentiment is that there are few good, comprehensive online tutorials for more advanced topics. One respondent is optimistic about learning the basics: "I think learning the basics is fairly simple, there is a ton of tutorials for that. Mastering could be harder," however another respondent implies the bounty of beginner tutorials may not be a good thing: "So much incorrect or subpar information/processes is shared among beginners."

3.4.2.5 Personal Factors Preventing Learning

Whether it's time, motivation, organization, or all three, the most common learning impediment respondents cite are their personal circumstances. Time especially was a common answer, with 18/75 respondents mentioning a lack of time in response to what's impeding them, seven of those answers being solely 'time' or 'lack of time'. Four respondents blamed a lack of motivation and eight mention 3D modeling frustrated them at times. This points to learning 3D modeling requiring a considerable amount of discipline or a moment in your life where you can dedicate yourself to study. "There's a lot to fiddle with when learning about nodes and UV wrapping. I need to just sit down

and really learn it well for a few weeks. So, I would say - [what impedes me is] Myself."

3.5 Discussion

The five major themes identified showcase a variety of learning struggles that 3D modelers face. Some of them are related to the software itself (steep learning curve, controversial tutorials) whereas others attribute struggle to outside factors (personal factors, art is hard). How are these themes connected, and how might we understand them through existing research?

3.5.1 Why is the curve steep? Why is the road bumpy?

When people start learning 3D modeling, they are learning both a new software and a new form of art. This may contribute to the steep learning curve, because new modelers have to balance learning the tools+UI of a new feature-rich software with the challenges of art. The confusing confluence of art and new technology means learning 3D modeling takes a lot of time and energy. This is probably the reason so many respondents described a steep learning curve; when people refer to a steep learning curve, they usually mean a skill that takes many repetitions to become proficient in [3], as 3D modeling does with its many initial difficulties. Respondents say that they are intimidated by the fact that they have to practice 3D modeling for a while before they can feel proficient in it: "Required knowledge [to do 3D modeling] seems vast and it is difficult to navigate through all the material to find a pathway to suit your own needs/desires." "[3D modeling] is very tedious. It takes a lot of time to understand all

the buttons and process of modeling. I spent months personally trying to learn."

The mental energy feature-rich 3D modeling software takes to learn may be a cause for peoples' inconsistent learning schedules; the learning process can easily be interrupted by anything else that takes some mental energy. Besides mental energy another contributor to the bumpy road might be the difficulty in finding good tutorials; learning is smooth when one has access to a good tutorial, but good tutorials can be hard to find in a swamp of overlapping, misinforming tutorials. Finally the bumpiness may also just be the result of standard learning patterns of growth and stagnation.

These ideas of a 'learning curve' and 'bumpy road' encapsulate a lot of the struggles people feel when learning a feature-rich software (RQ2).

3.5.2 Art and Self-regulated Learning

Even without complicated feature-rich software to learn, practicing art is a journey full of struggle. As *Art and Fear* puts it, "Artmaking can be a rather lonely, thankless affair" [2]. In the case of the beginner 3D modeler it's lonely, thankless, and full of confusing UI! One respondent describes a common angst amongst artists: "[3D modeling] seems something that requires artistic talent that I don't perceive myself to have". Talent plays some part, but more importantly art requires strong self-regulated learning, especially for 3D modeling because many people learn it online instead of in social, structured institutions like school. Beginner 3D modelers must organize their learning and motivate themselves, which many respondents struggle with.

3.5.3 On Tutorials

Educational content creators have noticed the steep beginner learning curve, and they have a solution - lots of beginner tutorial videos. While the effort is appreciated, some issues have arisen because of this heap of beginner tutorials: inaccurate information abounds and the supply of intermediate/advanced tutorials is small. Typically tutorials for feature-rich software are best when created by expert users [40], and the rush to help beginners may mean non-expert users are posting tutorials that have errors. Meanwhile the lack of advanced tutorials is noticed by the survey respondents. One respondent describes the situation well: "I think 3D modeling instruction is hampered by both the competitive nature of the field, where experienced modelers have no interest in teaching others their craft and introducing new competitors, and the lowest common denominator effect of YouTube where the most popular 3D tutorials avoid or minimize technical language and concepts in order to capture a wider audience." Aside from a lack of tutorials covering advanced technical topics, there aren't many tutorials on the artistic side of modeling either. "Meanwhile actual artistic tutorials seem to be lacking and often come down to "buy this magic thing that does it for you" which is not helpful."

Another explanation for the lack-of-advanced-tutorials phenomenon could be that beginner feature-rich software users learn from video tutorials, whereas experienced users go to forums for help [11] (RQ1). Feature-rich software users reading this are surely familiar with the pattern of being unable to find a video tutorial and asking on forums as

the second option - thus placing all of the more ‘advanced’ knowledge in forums. More advanced 3D modelers may also be using entirely different software from beginners; one study found that users of ‘mature’ software (stable user base + few updates) were typically much more experienced than users of ‘growth’ software (expanding user base + many updates) [11]. Since the 3D modeling survey was mostly sent out to Blender communities, which is a ‘growth’ software, it could be that growth software in particular have issues with tutorials.

Chapter 4

Gamelike Tutorials for Art Software

4.1 Overview

Gamelike Tutorials for Art Software (GLT4AS) is a taxonomy of the connection between video game tutorials and art software tutorials. For decades video game tutorials have been studied [57] and refined because games with bad tutorials (boring, don't teach content well, etc) aren't fun which detracts from the gaming experience. Games are a type of feature-rich software that try to make their learning fun, unlike feature-rich art software which don't structure their learning much at all. Gamelike Tutorials for Art Software compares tutorials in games and art software, aiming to answer the questions:

1. What are the common differences between existing video game vs art software tutorials?

2. Can the techniques that make game tutorials successful be applied to art software

tutorials as well?

Comparing game and art software tutorials gives us an idea of what art software tutorials look like (RQ3) and game tutorials may provide inspiration for those tutorials (RQ4).

4.2 Related Work

4.2.1 Game tutorial design

Video games, like art software, are complicated software that users need to be taught in order to use effectively; in the case of games, effective use is typically having fun. Video games tutorials must be fun and engaging otherwise the entire point of the game is negated, thus risking poor reviews and low sales. With this incentive, in the 50-odd years of video games' existence game designers have discovered how to craft good tutorials that both educate and entertain.

A principle in game tutorial design is that game tutorials become better (more fun, more educating) the closer they are to 'normal' gameplay; that is, the way players naturally play the game once they already know how it works. There are a few ways this principle is commonly broken, leading to a less enjoyable gaming experience. Games that frontload the tutorial instead of splitting it into smaller pieces at relevant moments risk both boring the player and having them forget information before they need to use it [9] [56]. Other times bad game tutorials will heavily rely on written explanation, interrupting player's experience as they have to switch between reading and whatever

the game's main mode of play is (action, exploration, narrative, etc) [9] [56]. A lack of tutorial as well can lead to frustration as players struggle to progress in a game for reasons they can't understand, causing them to leave the game world entirely to find answers from their friends or the internet [56].

Another principle in game tutorial design is that information should be fed to players at a manageable pace. "A player's willingness to learn grows along with their level of investment" [19]. Frontloading is again an issue as frontloaded players are given more information than they know what to do with. To alleviate this, there are a handful of tactics games use to control the rate of information given to the player. One tactic is called the inverted pyramid of decision making [21], where players only need to make a few decisions when they begin to play but must make more and more decisions as play progresses. Another tactic is to hide game UI elements until they become relevant so that the player doesn't need to worry about elements they don't understand yet [9] [56]. Another tactic some games use is requiring players to complete an action multiple times before letting them progress to ensure that the player is capable before being taught the next thing [56].

Finally, general learning and UI principles apply to good game tutorials as well. Feedback for players' actions in games should be clear and immediate and UI should have recognizable signifiers. Experienced players should be able to bypass tutorials that they don't need, and new players should be able to revisit lessons if they desire.

4.2.2 Gamification

Gamification has been recently defined as "a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation" [29]. 'Gameful experiences' can mean many things, but some studies have found the most popular forms of gamification to be: [17]

1. Points
2. Levels/Stages
3. Badges
4. Leaderboards
5. Prizes and Rewards
6. Progress Bars
7. Storylines
8. Feedback

Gamification has been applied in many areas, including art software tutorials. GamiCAD [15] is a tool designed to test gamification techniques in learning CAD (computer-aided design), which is usually performed in feature-rich CAD software. GamiCAD includes points, storylines, feedback, and rewards from the previous list of common gamification strategies, as well as time pressure and dispersing information between challenges. Sketch-Sketch Revolution [13] is another art software tutorial

which instead of using aesthetic game features like points or music, instead focuses on guiding artists through a step-by-step process similar to game tutorials.

4.3 Methods

To create a taxonomy of tutorial features, GLT4AS began as a literature review of game tutorial design, multimedia learning, user experience, and gamification, followed by an analysis of popular games and art software tutorials. Common features in game and art software tutorials were identified by counting their number of occurrences across a pool of examples. In total 26 mobile game tutorials and 22 art software tutorials were analyzed. Tracked features were determined during the data gathering process to allow for a flexible set of features. After counting feature occurrences, a taxonomy of features was made to compare between games and art software.

4.4 Results

The first taxonomy identified common tutorial features in games (Fig 4.1). Notably every game highlights UI elements when explaining them, and most games have some sort of story or character involved in their tutorial.

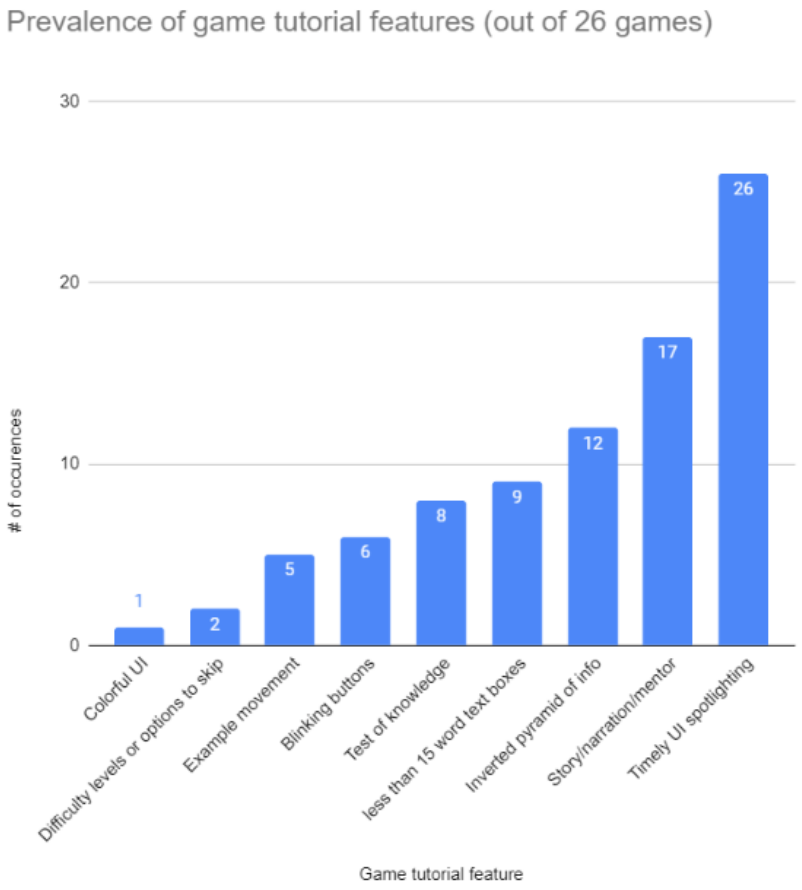

Figure 4.1: The prevalence of certain features identified by the researchers across game tutorials.

The second taxonomy identified common tutorial features in art software (Fig 4.2). All art software tutorials linked to learning materials outside of the software itself, most of them being screen recorded videos where someone walks through using the program.

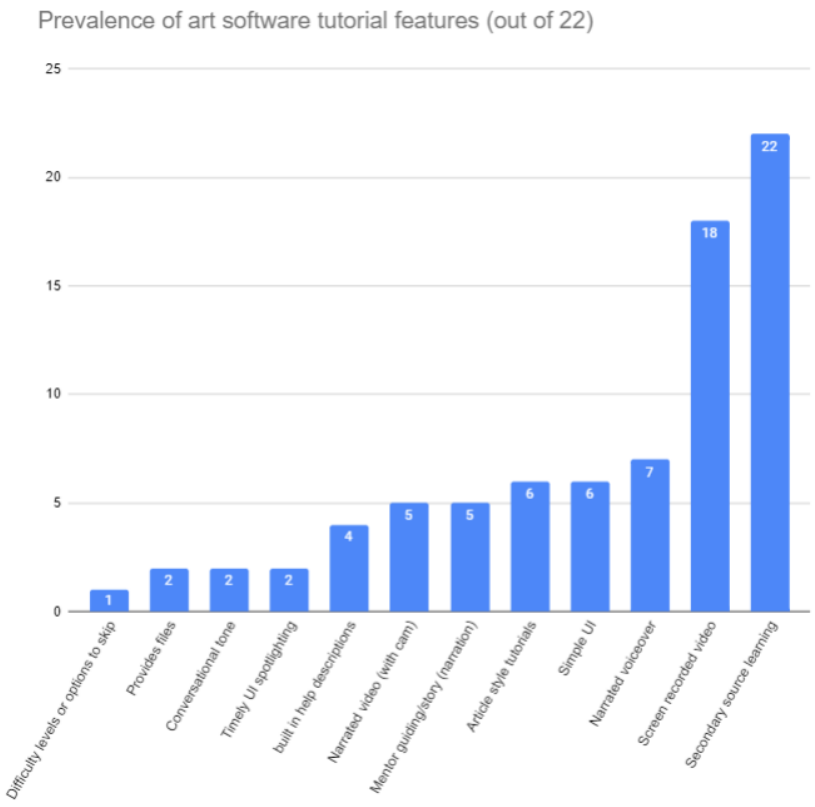

Figure 4.2: The prevalence of certain features identified by the researchers across art software tutorials.

4.5 Discussion

Despite our small sample size, some trends in features were obvious.

The standard art software tutorial is minimal and optional. The first time some art software are opened after downloading there's an optional tutorial prompt which walks the user through a few tools, and that's about it. To teach users, every art software we analyzed redirected users to the internet to learn from a secondary source. Most of the time (18/22) this source is a screen recorded video (or playlist of videos),

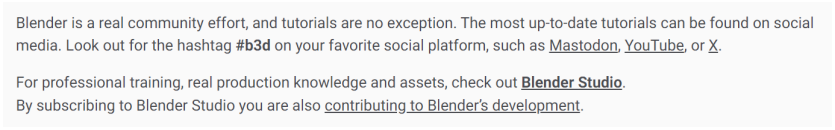

Figure 4.3: The page that Blender (3D modeling software) brings you to when you click on the tutorial button (as of 2024).

where an experienced user explains how the software works and demonstrates how to use its tools. A few art software also linked to text and image based media like PDFs or presentations as tutorials. Most of art software have official tutorials produced by the developers, but many of them also promote users to explore the internet and find community-made tutorials.

The way video games approach tutorials is much different than art software. Whereas art software tend to have minimal built-in tutorials and rely on users to learn from external material, games teach their players within the software itself. Games use a completely different set of teaching strategies because their integration lets them be interactive; for example, video games can progress the tutorial only once the player performs certain actions, and they can hide UI until it becomes relevant. None of the 26 games we analyzed directed players to learning resources outside the game itself.

Art software teach their users outside the program whereas games teach within - why is this? Perhaps the biggest reason is in users' goals. Video games create fun experiences for players by providing distinct goals, whether it's completing a level, finishing a quest, or defeating an evil God [5]. The player gets what they want out of a game by agreeing to complete the game's goals, or sometimes by subverting them. Art-making on the other hand is more ambiguous, its goals constantly changing as the

vision of the artist changes, a process which computers can't measure. Thus art software has taken a more hands-off approach to tutorials than games. The entire experience of games is mediated by the game, but the most that art software can assume about the artist is that they want to know the tools work at a basic level.

Despite the inclination for art software to let users figure things out on their own, I believe users benefit from better built-in tutorials. Artists still need to understand how tools work at a basic level, and many artists want to know how they might use each tool and what they can make. With built-in tutorials artists wouldn't need scrounge for good tutorials online, becoming frustrated when the tutorials they find are out-of-date, too simple, or too difficult [50]. A built-in tutorial can also track the user's actions to provide more immediate feedback and hide irrelevant and distracting UI. GamiCAD, a built-in tutorial for CAD software, has shown benefits of some gamification elements for built-in art software tutorials [15].

An important point to consider with built-in tutorials is their production cost. It's much cheaper to have an experienced user record themselves explaining the software than to develop a tutorial inside the art software, maintaining it through updates. Some feature-rich software like Blender and Photoshop in their 'growth' phase [11] receive continuous updates and maintaining accurate tutorials for them is difficult. It's up to the developers to see if the benefits of built-in tutorials outweighs their production cost.

Although my arguments so far have pitted built-in and secondary source tutorials against each other, there is room for coexistence. Past the basics, it's up to users how to use art software and share their tips, techniques, and opinions. The same phe-

nomenon happens in video games too, where players share advanced knowledge about the game, and emergent strategies with each other through videos and online community.

Chapter 5

The Future of Art Software Tutorials

With growth in digital art industries [53] [30] and social media [55], it's important to consider how art software should be taught in the future. What new strategies are people trying today? If art software tutorials change, what effect might it have? What might better art software tutorials look like? Whereas the 3D modeling survey and Gamelike Tutorials for Art Software study the current status quo, this section describes today's experimental learning strategies and what art software tutorials might look like tomorrow.

5.1 Experimentation in Tutorials Today

The art software tutorials found in Gamelike Tutorials for Art Software were generally uncreative, but that doesn't mean there aren't some interesting art software tutorials out there. The learning problems of feature-rich art software haven't eluded some academics who've tried their hand making experimental tutorials. GamiCAD

[15] is a tutorial for computer-aided design built to study how adding game elements (points, narrative, time pressure) can improve tutorials. Sketch-Sketch Revolution [13] is a painting software tutorial system that allows artists to author their own tutorials inside the painting software, which other artists can follow along to. TutVis [50] is a browsing interface which helps artists estimate the difficulty of tutorial videos before they watch. In a similar machine learning vein, CoDis [59] is a tool that predictively suggest commands to users of feature-rich applications. These tools all demonstrate underexplored techniques to improve feature-rich art software tutorials. Finally, MSWord Personal is an interface layout toggle system that lets users toggle between a full-feature UI layout and a simpler custom UI layout in Microsoft Word [44]. These experiments present many paths that feature-rich art software might take.

Outside of academics, there exist some applications which are oriented as educational stepping stones before people dive into feature-rich applications: Scratch [1] is a programming IDE for beginner level programmers that uses colorful draggable blocks instead of written code, Kidpix [34] is a simple and fun painting program which lets people create interesting images quickly, and Syntorial [54] is an audio synthesizer software that teaches users how each synth setting works and tests them before they can move forward. These applications were built for education, but even simple applications which weren't can give people footing in the basics of a digital craft before tackling feature-rich applications. Painting in MS Paint or 3D modeling in Sketchup can build confidence before tackling Photoshop or Blender. Smaller casual creator [8] programs can also be used to let people explore new artistic tools in a safe setting. Besides being

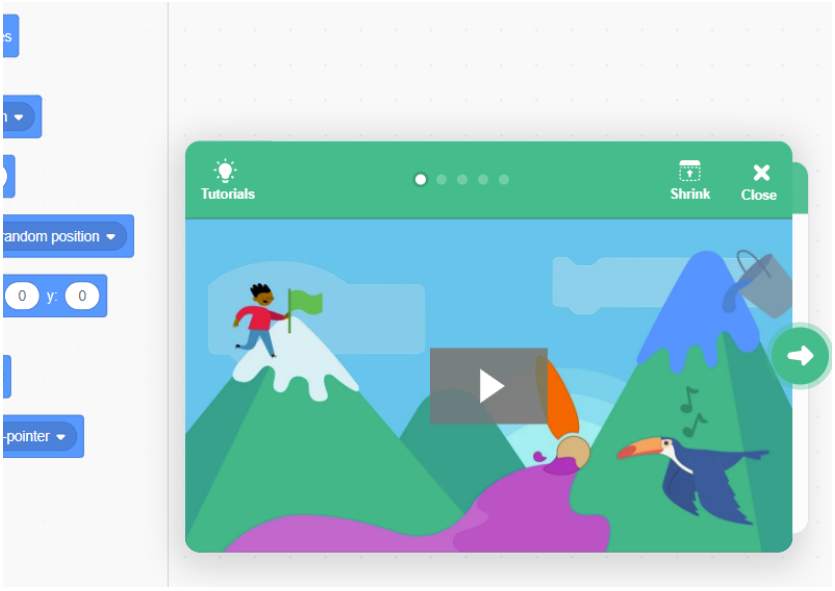

Figure 5.1: Scratch's pop-up tutorial when the program is first opened.

simpler, these smaller and more educational programs tend to *also* have better tutorials for their own tools than their feature-rich counterparts. On starting your first project, Scratch has a video tutorial pop-up inside the app which explains some basics and links to additional resources (Fig 5.1).

5.2 The Consequences of Centralized Tutorials

Suppose art software designers read this thesis and decide to add built-in tutorials to every art software. Even better, suppose those tutorials are great at teaching and everyone who downloads the software follows them! One still must be wary about the effect a universally followed tutorial has on peoples' creativity. As the authors of *Beyond the Artifact* [16] describe: "Designing a tool means structuring and bound-

ing its users' ideas, goals, and intentions. These constraints construct a normative ground—how someone should or could think, act, and express themselves during use". The more tool developers scaffold their users' experience, the more they funnel users into certain creative domains. Although a central tutorial can be helpful, if learning becomes too centralized it can disempower artists and lead to less interesting art [16]. Even a tutorial that addresses all of peoples' needs from the 3D modeling survey could run into a new problem of silently stifling their creativity.

5.3 Speculation about Better Tutorials

Based on previous discussion in this thesis, to better onboard users into feature-rich art software I confidently propose the following two strategies:

1. Feature-rich art software should provide clear option(s) for simpler UI (UI scaling).

2. Feature-rich art software should provide obvious direction to foundational learning content from inside the software (obvious foundational tutorials).

These two strategies address a lot of the themes of struggle found in the 3D modeling survey. Multiple survey respondents mentioned UI impeded them, and anecdotally most modeling beginners I've spoken with complain about confusing UI. Scalable UI will help them find the basic features they need and flatten the learning curve, and studies have shown that allowing users to toggle interface layouts can increase learnability [44]. The fact that UI scaling is a unanimous practice in video game tutorials as well speaks to its effectiveness. Many respondents to the 3D modeling survey also mentioned

becoming frustrated while learning, which overwhelming UI likely contributes to.

The second change, adding obvious foundational tutorials, directly deals the disorganized state of tutorials which was a big pain point in the 3D modeling survey. Cutting out the online tutorial hunt saves beginners time and confusion caused by inaccurate information in some unofficial tutorials. For beginners, the 'bumpy' road should become much more straightforward if they're directed by the software itself to the foundations they need to begin making their own art. Strong beginner tutorials would also allow content creators to focus on teaching more advanced topics, helping address the lack of advanced tutorials.

The two strategies I propose help with the learning curve, bumpy road, controversial tutorials, and organization, they notably don't address motivation or art practice. Problems of motivation and art practice aren't unique to feature-rich art software, and I don't believe they're best solved inside art software. Motivation comes from many places and improving your art is a complicated and deeply personal pursuit. Neither of these things are worth trying to mediate inside a program like Photoshop. Making art, whether using feature-rich art software or not, is always a struggle and trying to alleviate these issues is out of the scope of this thesis.

I've suggested two tutorial strategies, but many more were discussed earlier in the GLT4AS section. Why not implement all the strategies? There is some merit to strategies like point systems, music, and UI highlighting, but I'm suspicious that adding them to art software tutorials may come with notable disadvantages. Although GamiCAD's gamified tutorial with points, music, etc did show improvement over the

non-gamified one [15], GamiCAD did not test gamification strategies individually so it's hard to tell which ones were helpful or not. Music and points don't directly address any of the struggles found in the 3D modeling survey so I can't see how they would be helpful. Even UI highlighting, which is similar to UI hiding, isn't as good for cognitive load and the art software which implements it needs to somehow know when to highlight UI.

It's important that changes to art software don't hinder users, and I don't think UI scaling or obvious foundational tutorials do. Both of them can be harmlessly presented to beginners when launching the software, and don't need to interfere at all with existing users' habits. Art software developers just need to be careful that foundational tutorials don't shrink the software's normative creative ground by promoting certain techniques too much or dissuading others. If feature-rich art software add these two beginner-friendly strategies it could give many new artists a foothold in digital art.

Chapter 6

Future Work

Besides proving that the two strategies I propose actually do work, there is still much to be explored in art software tutorials which are built into the application. Currently the line between art tools and art tutorials is distinct; tutorials exist on the internet and tools exist in the software, but I can see this line being blurred for a better learning and art-making experience, just like how the best game tutorials are invisibly tucked into gameplay itself. Further research in feature-rich art software could develop and test educational artifacts unique to these complicated software, which also respect the user's creativity.

As exploratory research, the 3D modeling survey raises more questions than it answers. Respondents to the survey provided good insight to some issues, but many of them gave very short responses that deserve unpacking. The 3D modeling survey only surveys people about 3D modeling, not all feature-rich art software, so it would be good to see how other digital art forms such as painting or vector graphics fare.

Longitudinal studies on feature-rich art software may help us understand the learning journey better than the single picture in peoples' journeys that the survey revealed. Longitudinal studies could go further by investigating how people learn when provided a tutorial series vs left to their own devices, or the differences in practicing on a easier software then moving onto feature-rich software versus diving right into feature-rich software.

Chapter 7

Conclusion

Despite their importance in industry and culture, feature-rich art software are not great at teaching new users. To understand why this is and what can be done about it, this thesis investigated the experience of 3D modelers, the landscape of art software tutorials, and experimental work in art software tutorials.

The first study, which was a survey sent out to 3D modeling communities online, identified themes of struggle during 3D modeling education. Respondents shed light on which areas of 3D modeling programs are hardest to learn and painted a picture of the 3D modeling journey.

The second study, Gamelike Tutorials for Art Software, was a taxonomy of video game tutorials and art software tutorials based on similarities between games and art software and the field of game tutorial design. The differences between game and art software tutorials highlighted potential strategies for art software tutorials.

The final section of this thesis filled in the gaps of art software tutorials that

the previous studies didn't catch and proposed two strategies, UI scaling and obvious foundational tutorials, which I believe will work best to help beginners learn feature-rich art software.

Bibliography

[1] https://scratch.mit.edu/about.

[2] T. Bayles, D. Orland. *Art and Fear: Observations On the Perils (and Rewards) of Artmaking*. Image Continuum Press, 2001.

[3] D. Benzel, E. Orr. A steep learning curve is a good thing! *The Spine Journal, vol 11 issue 2*, 2011.

[4] J. Carroll. *Annual Review of Psychology*, chapter HUMAN-COMPUTER INTERACTION: Psychology as a Science of Design. Annual Reviews, 1997.

[5] J. Carroll, J. Thomas. Fun. In *ACM SIGCHI Bulletin*, 1988.

[6] https://web.archive.org/web/20080310082944/http://design.osu.edu/carlson/history/timeline.html1960.

[7] K Charmaz. *Constructing Grounded Theory*. Sage, 2006.

[8] Kate Compton. *CASUAL CREATORS: DEFINING A GENRE OF AUTOTELIC-CREATIVITY SUPPORT SYSTEMS*. PhD thesis, UC Santa Cruz, 2019.

[9] Extra Credits. Tutorials 101 - how to design a good game tutorial - extra credits, 2012. https://www.youtube.com/watch?v=BCPcn-Q5nKE.

[10] D. Darejeh, A. Singh. *Journal of Computer Science 9*, chapter A REVIEW ON USER INTERFACE DESIGN PRINCIPLES TO INCREASE SOFTWARE USABILITY FOR USERS WITH LESS COMPUTER LITERACY. Science Publications, 2013.

[11] R. et al. Earle. User preferences of software documentation genres. In *SIGDOC '15*, 2015.

[12] Chung et al. The intersection of users, roles, interactions, and technologies in creativity support tools. In *DIS 2021 Proceedings*, 2021.

[13] Fernquist et al. Sketch-sketch revolution: An engaging tutorial system for guided sketching and application learning. In *UIST '11*, 2011.

[14] Frich et al. Mapping the landscape of creativity support tools in hci. In *CHI 2019 Proceedings*, 2019.

[15] Li et al. Gamicad: A gamified tutorial system for first time autocad users. In *UIST '12*, 2012.

[16] Li et al. Beyond the artifact: Power as a lens for creativity support tools. In *UIST 2023 Proceedings*, 2023.

[17] Nah et al. Gamification of education: A review of literature. In *HCI in Business 2014*, 2014.

[18] Terry et al. *The SAGE Handbook of Qualitative Research in Psychology, chapter 2: thematic analysis.* SAGE Publications, 2017.

[19] George Fan. How i got my mom to play through plants vs. zombies, 2012. https://www.youtube.com/watch?v=fbzhHSexzpY.

[20] The Blender Foundation. 2022 blender foundation annual report. Technical report, The Blender Foundation, 2023.

[21] Tracy Fullerton. Bruce shelley. https://www.gamedesignworkshop.com/bruce-shelley.

[22] A. et al. Graesser. Intelligent tutoring systems, 2012.

[23] T. et al. Grossman. A survey of software learnability: Metrics, methodologies and guidelines. In *CHI '09*, 2009.

[24] L. Hanushek, E. Woßmann. *International Encyclopedia of Education*, chapter Education and Economic Growth. Elseiver, 2010.

[25] I. Hecker and A. Briggs. Overlooked and underconnected exploring disparities in digital skill levels by race among older youth in the us. Technical report, Urban Institute, 2021.

[26] P. Hecker, I. Loprest. *Foundational Digital Skills for Career Progress.* Urban Institute, 2019.

[27] M. Hertzum. Frustration: A common user experience. In *DHRS '10*, 2010.

[28] http://cienciaecultura.bvs.br/scielo.php?script=sci_arttextpid=S0009-67252006000300021.

[29] J Huotari, k. Hamari. Defining gamification - a service marketing perspective. In *Mindtrek 2012*, 2012.

[30] https://www.ibisworld.com/global/market-size/global-graphic-designers/.

[31] J. et al. Jacobs. Supporting expressive procedural art creation through direct manipulation. In *CHI '17*, 2017.

[32] L. Kantosalo and A. Lakkala. What is digital competence? Technical report, EUN Partnership AISBL, 2011.

[33] K. et al. Kiani. Exploring user attitudes towards different approaches to command recommendation in feature-rich software. In *VL/HCC '20*, 2020.

[34] https://www.mackiev.com/kidpix/.

[35] K. et al. Kreslins. *Digital Literacy, Digital Culture and Digitalization in Europe.* Journal of Internet and e-Business studies, 2022.

[36] S. Laborde. 100+ adobe statistics for 2023: Decrypting its staying power. Technical report, TechReport, 2023.

[37] B. et al. Lafreniere. Task-centric interfaces for feature-rich software. In *OzCHI '14*, 2014.

[38] J. Lee, Y. Choi. A review of online course dropout research: implications for practice and future research. In *Educational Technology Research and Development*, 2010.

[39] J. et al. Li. What we can learn from visual artists about sofware development. In *CHI '21*, 2021.

[40] K. Lount, M. Bunt. Characterizing web-based tutorials: Exploring quality, community, and showcasing strategies. In *SIGDOC '14*, 2014.

[41] Peter Lunenfeld. *Snap to Grid*. The MIT Press, 2001.

[42] S. Mamedova and E. Pawlowski. A description of u.s.adults who are not digitally literate. Technical report, US Department of Education, 2018.

[43] R. Mayer. Multimedia learning. *The Annual Report of Educational Psychology in Japan Vol 41*, 2002.

[44] J. et al. McGrenere. *ACM Transactions on Computer-Human Interaction, Vol. 14*, chapter A Field Evaluation of an Adaptable Two-Interface Design for Feature-Rich Software. ACM, 2007.

[45] Hans Meij, Joyce Karreman, and Michaël Steehouder. Three decades of research and professional practice on printed software tutorials for novices. *Technical Communication*, 56, 08 2009.

[46] C. Paul. *Digital Art*. Thames Hudson, 2023.

[47] L. et al. Ponzanelli. Too long; didn't watch! extracting relevant fragments from software development video tutorials. In *ICSE '16*, 2016.

[48] P. Pyla and R. Hartson. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann, 2012.

[49] Razbuten. https://www.youtube.com/watch?v=-nJtd8AJghM.

[50] S. et al. Sabab. An automated approach to assessing an application tutorial's difficulty. In *IEEE VL/HCC '2020*, 2020.

[51] B. Shneiderman. Creativity support tools. In *Communications of the ACM 2002*, 2002.

[52] A. Singh, V. Thurman. How many ways can we define online learning? a systematic literature review of definitions of online learning (1988-2018). In *American Journal of Distance Education*, 2019.

[53] https://www.statista.com/outlook/dmo/digital-media/video-games/worldwiderevenue.

[54] https://www.syntorial.com.

[55] https://ourworldindata.org/rise-of-social-media.

[56] Game Maker's Toolkit. Can we improve tutorials for complex games?, 2021.

[57] M. White. *Learn to Play: Designing Tutorials for Video Games*. CRC Press, 2014.

[58] M. et al. Wiebe. Exploring user attitudes towards different approaches to command recommendation in feature-rich software. In *IUI '16*, 2016.

[59] G. Zolaktaf, S. Murphy. What to learn next: Recommending commands in a feature-rich environment. In *IEEE ICMLA '2015*, 2015.