MODDING IN MASSIVELY MULTIPLAYER GAMES

-

A CASE STUDY OF ELITE: DANGEROUS

Juha Syrjä

University of Tampere
Faculty of Communication Sciences
Information Science and
Interactive Media
Master's Thesis
April 2018

UNIVERSITY OF TAMPERE, Faculty of Communication Sciences

Information Science and Interactive Media

SYRJÄ, JUHA: Modding in Massively Multiplayer Online Games - A Case Study of Elite: Dangerous

Master's Thesis, 61 pp., 1 appendix

April 2018

The aim of this thesis was to study played-created third-party tools in the massively multiplayer online game Elite: Dangerous through the lens of modding and argue that Elite tools constitute a type of game mod. Modding historically refers to players modifying game software, but modding in Elite is not sanctioned by the game developer. Elite players have instead created standalone tools (both web applications and downloadable programs) that harvest and utilise the game's informational content, or game data, to enhance Elite gameplay. The existing academic typology of game mods do not account for standalone tools, largely due to a lack of precise definition on what modding is and what player-creations can be considered as game mods.

Both Elite tools and their creators are studied. First, an empirical research on Elite tools was conducted that produced a typology of third-party tools and analysed their relation to the game. Second, a survey on Elite tool developers carried out to find out how they relate their creations as well as inquiring about their modding habits and views on the concept.

Although third-party tools can be considered as game mods as per academic definitions, the main finding of the thesis was that there is a disconnect in how players and academia define mods and modding. For players, game mods are those creations that explicitly alter game software, while academic definitions are more wide-ranging and include the notion of changing how a game is played.

Keywords: games, modding, mods, mmog, elite dangerous

Contents

1 INTRODUCTION..1

2 LITERATURE REVIEW...6

2.1 What is modding?..6

2.2 Evolution of modding..7

2.3 Types of mods..10

2.4 Paratext...13

2.5 Participatory culture..15

2.6 Free labour..18

2.7 Modders and their motivations....................................21

2.8 Working definition of digital game modifications.........23

3 ELITE: DANGEROUS MODDING..26

3.1 Elite: Dangerous..26

3.2 Method and data..28

3.3 Game data...28

3.4 Types of third-party tools..30

3.4.1 API tools...31

3.2.2 Online tools...33

3.2.3 Trading tools...35

3.2.4 Player character tools..37

3.2.5 Overlays..38

3.2.6 Mobile device tools...39

4 ELITE MODDER SURVEY...41

4.1 Method and data..41

4.2 Survey design...42

4.3 Results...44

4.3.1 Background...44

4.3.2 Play...45

4.3.3 Modding..46

4.3.4 Modding in Elite...49

4.3.4 Views on mods and modding..52
5 DISCUSSION AND CONCLUSIONS..54
REFERENCES..59
APPENDIX 1: SURVEY FORM...62

1 INTRODUCTION

Games, particularly commercial ones, are often thought of as closed systems that are designed to be played in a certain way according to the rules set by the game designer. However, games are by their very nature malleable with room for free play and manipulation (Sotamaa and Wirman, 2015), be it through implementing house rules for a board game or creating a variant of a card game. In context of digital games, players have a long history of altering games by tweaking game code, adding or removing content, and generally consuming games in ways unanticipated by their designers. The practice of modifying digital games has become an important characteristic of gaming culture, known as *modding* (from modify or modification) (Laukkanen, 2005; Sotamaa and Wirman, 2015), and it has permeated virtually all types of digital games.

In modding, players create modifications, or *mods*, that vary greatly in terms of how, what or to what extent they alter or add to a game (Sotamaa and Wirman, 2015). Game mods range from altered game code that transform game rules to audiovisual changes and, depending on definition, may even include modifications to gaming hardware (Scacchi, 2010). More ambitious mods called *total conversion mods* may replace all original content of a game, leaving only the underlying game engine untouched, and in the process create new game modes (Laukkanen, 2005; Scacchi, 2010).

Although modding has become a common practice that game companies nowadays support by releasing modding tools and generally opening up the game system for players to tinker with, in online multiplayer games it is typically prohibited or limited to certain aspects in order to ensure that players are on a level playing in terms of game balance. In massively multiplayer online games (MMOGs), in which great numbers of players inhabit the same virtual world simultaneously, modding is typically altogether prohibited or limited to certain aspects of the game, such as the game's user interface (UI). *World of Warcraft* (Blizzard Entertainment, 2004), for instance, has a vibrant UI modding scene where players create add-ons that change the functionality of the UI or reveal information that would otherwise be inaccessible. Perhaps more often than not, however, modding in MMOGs is not possible, which is not to say that players are left without means to change the way the game is is played.

In the MMOG *Elite: Dangerous* (Frontier Developments, 2015), even the UI is off-limits for players to tinker with. *Elite* is a spaceship simulation game set into the far future, when mankind has settled thousands of star systems and space travel is commonplace. *Elite* players captain spaceships to explore the virtual Milky Way, fight off space pirates (or become one themselves) and trade between the myriad settlements of the populated space and so on (*Elite* is introduced in more detail in chapter 1.3). Being an online game, *Elite* game servers are in constant communication with game clients to keep both the players and the game world up to date. In the absence of ways to modify the game software itself, *Elite* players have instead developed tools that observe the traffic between game servers and game clients, copy and store the data, and allow players to sort, filter and manipulate it in ways that are not possible in the game itself. The most common functionality of these third-party tools is to keep track of the game's virtual market data and discover profitable trade routes by cross-referencing the saved data. Considering that the game does not provide players means to keep track of the trading or other relevant data players come across during play, these tools play an important role for Elite players. In other words, third-party tools extend the game's information management capabilities and thus enable players to play the game in a more enriched way - effectively circumventing the game's rules.

The existing academic definition of modding and typology of mods do not account for these kinds of player-creations, largely due to the lack of precise definition on what modding is and what constitutes a game mod (Laukkanen, 2005). As Scacchi phrases it, "if in some way you alter a game, how its [sic] played, or on what it is played, have you created a game mod?" (Scacchi, 2010., under Customizing, tailoring, and remixing game embodiments). Adopting a similarly wide-ranging view, I would argue that *Elite* third-party tools significantly alter how the game is played even though they do not explicitly alter the games formal representation, the game software. As Taylor (2007) points out, game studies has particularly in its early years stressed "designer- or system-centric views on what constitutes a game" (p. 122) and that emergent player behaviours are often situated peripherally to the game. She argues that including player participation and production in games is "defitionally and functionally central" (p. 124) in understanding what a game is. Consalvo (2017) echoes Taylor's sentiment in that game scholars often approach games as the central text of scholarly inquiry, while in some cases games can only be understood in relation to other texts. Citing total conversion mods as an example of games becoming de-centered as the primary text, she argues for fewer central texts and more "study into the relatedness, interconnectedness, and contingent nature of many kinds of popular culture texts" (ibid., 182).

In this thesis I explore the phenomenon of third-party tool development in *Elite* through the lens of modding and aim to both refine the definition of game mods as well as expand the academic typology of game mods. This can be phrased as the main research question:

- RQ1: What are third-party tools and how do their developers relate to them?

In approaching the question, both *Elite* third-party tools and the players who developed them are studied. First, an empirical and mainly descriptive study of *Elite* third-party tools is presented, in which the data set is simplified by identifying the key, illustrative types of tools with examples. The analysis then serves as context for the second part, which presents the survey of the players who develop third-party tools for *Elite*. The survey aims to shed light into the circumstances in which the tools were created, what area of gameplay they touch upon and what is their relevance for the *Elite* players. The survey was divided into sections by themes, which focused on the background of the developers, play habits in *Elite*, modding in general and in *Elite*, and how mods and modding are defined by the respondents. Follow-up questions to the survey were also conducted to expand on the responses to the survey.

The survey themes can be phrased as the following questions:

- RQ2: What are the developers' motivations?
- RQ3: How do the developers of *Elite* third-party tools define mods and modding?

The thesis is structured as follows: Chapter 2 is a literature review that goes through the academic studies relevant to this thesis. At the end of the chapter, I present my working definition of digital game modifications. Chapter 3 details the empirical study of *Elite* third-party tools, while chapter 4 presents qualitative survey of *Elite* third-party tool developers and its results. In chapter 5, the results from the previous two chapters are analysed and discussed, with reflections on further research.

2 LITERATURE REVIEW

This chapter introduces modding by providing an overview of existing academic literature on the topic. The first subchapter discusses the definition of modding in academic literature. The second subchapter goes through the evolution of modding. The third subchapter details types of game modifications. The fourth subchapter discusses mods and other materials surrounding games as paratexts. The fifth and sixth subchapters discuss modding as a form of participatory culture and as unpaid or free labour - respectively. The seventh subchapter is on modders and their motivations to modify games. Lastly, the eighth subchapter presents a working definition of digital game modifications.

2.1 What is modding?

Modding is a participatory practice common in digital gaming culture, where players create, extend and add to games with their own creations (Laukkanen, 2005; Scacchi, 2010; Sotamaa and Wirman, 2015). The outcomes of modding are game modifications, or *mods*, and they vary greatly between different types of games, and in terms of how and to what extent they change the original game product (Sotamaa and Wirman, 2015). Depending on the type of game in question, mods may change how a game appears or sounds like, implement new mechanics or add custom content, such as maps, characters or items. The more large-scale mods, known as *total conversion mods* (Scacchi, 2010), combine all of the above, often replacing all original content and creating an altogether new game in the process.

The definitions presented above are quite broad and understandably so, given the wide range of very different types of games and mods made for them. What exactly is modding and what constitutes a game mod? It is to this discussion that this thesis aims to contribute to by discussing *Elite* third-party tools as a type of game modification and approaching their creation through the lens of modding.

2.2 Evolution of modding

The origins of modifying games likely goes as far back in history as the history of games. Games are malleable systems by design: “every game system has room for manipulation and free play, allowing its players to bend, alter, and transform the game itself” (Sotamaa and Wirman, 2015, p. 1). If you have, for example, implemented a custom rule to a game of cards or chosen to disregard a seemingly silly mechanic in a board game, you have essentially modified the game. Traditional games like chess have gone through such iterations over the centuries; the current version of chess that is today considered to be the “official” version is just one out of hundreds of other variants that survived over the course of centuries to become the dominant version (Laukkanen, 2005).

Similarly, the origins of modifying computer games are in the early days of computing. The first games were made by computer enthusiasts who experimented with the capabilities of computer technology. One of the very first computer games, *Spacewar!*, was made in 1962 by students at the Massachusetts Institute of Technology, and in the following years it spread to virtually every university campus that had a computer. *Spacewar!* and other early games were openly shared, hacked and modified, and their source code reused and changed, an ethos shared by early coders and hackers (Sotamaa & Wirman, 2015).

The title of first game mod is widely attributed to *Castle Smurfenstein*, an art mod that replaced the original nazi-themed characters in *Castle Wolfenstein* (1983) with Smurfs. However, it was In 1993 when id Software's first-person shooter (FPS) *Doom* was released that began the modding phenomenon as it is known today. id's lead programmer John Carmack was encouraged by the modifications made on id's previous game *Wolfenstein 3D* (1992) to purposefully design *Doom* with modding in mind: the game was structured so that game content (e.g. graphics, sounds, levels) were separated as unencrypted WAD files (short for *Where's All the Data?*). This allowed the WAD files to be edited without programming and, perhaps more importantly, shared independently of the game. However, id did not support modders other than making the content readily available. Thanks to the Internet, which was starting to take off when Doom was released, players were able to pool their knowledge on online forums and develop editor tools to modify the WAD files. Were it not for the web, Doom mods would not have found such a wide audience and the first modding scene would have remained a hobby of the few. (Laukkanen, 2005)

With *Doom's* commercial success and modding having played an important role in it lead to other companies to follow suit. Valve's *Half-Life* (1998) and Epic Game's *Unreal Tournament* (1999) were designed from start to be modder-friendly and the companies included modding tools with the games, set up online communities and in-house workshops for modders (Laukkanen, 2005). One of the most famous and successful mods to date is Valve's *Counter-Strike* that was built on *Half-Life*, which attracted more players than previous FPS games combined and it became the first mod to be released commercially in 2001. Even though the Counter-Strike mod was available as a free download, the commercial release sold millions of copies (Laukkanen, 2005).

Although FPS games have played a prominent role in the evolution of modding, the practice has become a popular activity surrounding many other types of games on the PC platform. Other commonly modified game types are simulation, strategy and role-playing games (RPG), which game companies often release with editor tools intended for players to create content with. The RPG *Neverwinter Nights* (BioWare, 2002), for instance, was marketed with a focus on its Aurora toolset, that allowed players to create their own adventures in the spirit of traditional pen and paper role-playing games (Laukkanen, 2005). The virtually endless possibilities the toolset afforded turned out to be a popular one among players, given that it still has an active modding scene 15 years since the toolkits were released.

Although the creation of mods does not require action on the part of a game developer to take place, it is common for games to be bundled with toolkits or other means to facilitate content creation. Perhaps the most widely used digital game distribution platform *Steam* (Valve Software, 2003) also features *Steam Workshop*, which is a hub for player-created content. It also offers tools to make content and an organised way of publishing it for other players to download.

In MMOs like *Elite*, however, modding is often restricted to aspects of the game that do not disrupt the balance among the players. Largely for this reason, altering game mechanics or game world’s physics are usually out of bounds for players, unless of course it is a designed feature. In MMOs it is often the game UI that is opened up for player customisation through add-ons or plugins, which alter the functionality of the UI or what is shown within the game. For example, *World of Warcraft* (Blizzard Entertainment, 2004) has a vibrant user interface (UI) modding scene with the add-ons numbering in thousands and the most popular of them have been downloaded in hundreds of millions. The add-ons provide information about the game that would otherwise be hidden, such as the health of a boss in a raid, or change the menu functionality of the user interface. Blizzard supports the UI modding community by providing an application programming interface (API), through which players can create programming scripts that change the game UI's behaviour.

2.3 Types of mods

Which all player-made content can be considered to be game mods depends on the definition being used. Historically mods are alterations to the game software, be it game client or server, but the academic view on modding is much more broad, including hardware modifications and video productions remixed from gameplay footage.

Scacchi (2010) provides a comprehensive overview into the various kinds of player-created content and activities that can be considered to be results of modding practices. He categorises mods into *user interface modifications*, *game conversion mods*, *machinima* and *art mods*, *custom gaming computers* and *game console hacking* (ibid.).

Game user interfaces constrain and govern what players can do with a game and, in modifying it, players can acquire or display additional information that improve play performance and experience. Scacchi (Ibid.) lists three kinds of UI customisations: player character customisation, customisation of the colour palettes and representational framing of game displays (similar to web browser add-ons) and UI add-ons that modify "in-game information management dashboards". UI add-ons do not necessarily change gameplay rules or functions, but provide additional information about gameplay that may improve gameplay experience and immersion (Ibid.).

Game conversion mods that add to or modify parts of a game are perhaps the most common type of game mods. Partial conversions affect the appearance or capabilities of both player and non-player characters, game objects and resources such as weapons and spells, play levels or maps, game rules or play mechanics. (Scacchi, 2010)

In a study of three different kinds of games and their modding scenes, Laukkanen (2005) further divided game conversion mods into audiovisual modifications and mechanics modifications; the former alter the appearance of a game while the latter result in changes to how the game functions through changes in its code. This categorisation can be applied to game conversion mods in general, although particularly audiovisual mods vary greatly depending on the type of game they are made for. Laukkanen (ibid.) also divided audiovisual mods made for first-person shooter games (FPS) into maps, textures, prefabs (reusable objects), map models, skins (graphics that change character appearance), character and weapon models, sprites and sounds. Mods made for the real-life simulator game *The Sims* (Maxis, 2000) follow a similar pattern, although they only affect the game's visual appearance, consisting of "skins, head and body meshes, walls and floors, objects, hacked objects and houses/lots" (Ibid.).

The more large-scale mods that replace most or all of the original game content, leaving only the game engine untouched, are called *total conversion mods* (Scacchi, 2010). Game engine is the underlying software framework that consists of reusable components that are recycled from one game to another so that the basic game features - such as physics, networking, collision detection - do not need to be implemented for each game anew. Total conversions may result in altogether new games (Scacchi, 2010) and become commercial titles on their own right.

Machinima and art mods are both seen as efforts to mod the play experience by employing games as medium for other creative purposes. In machinima, computer games are utilised for creative expression in producing films made of recorded and remixed gameplay footage, for instance to retell stories about players' experiences in game or to create musical pieces (Scacchi, 2010). Although machinima productions do not modify game software, they are alike game mods in that both make use of the game engine in creating new digital objects and experiences (Sotamaa and Wirman, 2015). Early machinima videos had to be replayed on the game client, but today they are commonly uploaded to video hosting sites such as YouTube.

Art mods alter a game's visual content or gameplay experience in some fashion "through manipulation, intervention, appropriation, or other creative transformation of a game's original visual content as it is consumed by users during during a play session" (Scacchi, 2010, under Machinima and Art mods). Art mods provide a way for artistic expression and social critique, Velvet Strike (1999) for example is an art mod that transformed the multiplayer sessions in *Counter-Strike* into performance art by spraying anti-violence messages on to the game level's surfaces (Sotamaa and Wirman, 2015).

Modding is not limited to game software and can extend to tinkering with gaming hardware. Particularly in fast-paced games like *Counter-Strike* reaction time is in a crucial role, which is affected in part by player skill and in part by the capabilities of the gaming hardware. A computer that performs worse than the opponent's may result in a handicap, which some players attempt to minimise for instance by overclocking the central processing unit (CPU) for increased speed (Simon, 2007). Gaming hardware modding also includes aesthetic customisation of computer cases: purpose is not to enhance the performance of the computer, but to indicate a player's enthusiasm and commitment to games as more than merely a source of entertainment (Scacchi, 2010). The modified cases are shown off in public events such as local area network parties.

While customisation of personal computers is motivated by improved performance of the gaming system, gaming console hacking is about extending the range of experiences that users encounter on gaming consoles. Console hacking is motivated by understanding how the device works to remove limitations set by the device's manufacturer, which allow the device to be repurposed as an alternative personal computer (e.g. capable of running utility software) or to be able to run content protected by vendor-imposed restrictions (e.g. region-protected content). (Scacchi, 2010)

2.4 Paratext

Digital games are surrounded by a plethora of different kind of content that is created by both players and game developers. From game trailers and gameplay videos to walkthroughs and mods, these artefacts play a role in fashioning the experience of playing a game. Consalvo (2009) argues that the concept of *paratext* is useful in making sense of such a system and to help understand the relationship between games, the aforementioned artefacts and the culture around them.

Paratext was developed by Genette (1997) in context of books and other published works of literature that are surrounded by paratextual elements that frame the consumption of a text. For example, a book is accompanied by a cover, preface, table of contents, illustrations and many other things that act as paratexts to the primary text of the book. They help shape the experience of reading, give meaning to the text and aid in understanding it. Paratexts are connected to, but distinct from the text they accompany. Some are physically connected to the primary text (e.g. those of a book mentioned above), while others interact with the text from afar (e.g. a review and other commentaries).

Lunenfeld (1999) applied the concept to digital media and noted that the boundaries of a primary text and paratexts are becoming increasingly blurry and impossible to distinguish from each other. He argues that because digital texts are inherently malleable, readers are able to modify the text (both literally and figuratively) and combine it with other texts. Such interventions rework the primary text by redefinining the meaning of the text and, in especially in case of mods, literally altering the text by tweaking the structure that digital texts are built on.

In her book on cheating, Consalvo (2009) took the concept of paratext and viewed digital games as primary texts, which are surrounded by a plethora of paratextual items. She writes that well before a game is released, communications and artifacts about it begin to appear as players discuss the game on blogs, websites, discussion boards and so on, and game developers provide information about the game via, for instance, gameplay trailers and newsletters. Once a game is released the trickle of information becomes a torrent and possibilities to learn about a game increase rapidly, as walkthroughs and guides appear, players publish gameplay videos and stream their gameplay sessions. Consalvo (ibid.) argues that these elements constitute paratexts to digital games by yielding important information about a game and advice on how to fare better in it, thus shaping the experience of playing a game.

Consalvo (2017) notes that there is a tendency among games studies to position games as the primary text and argues that in some cases games become de-centralised, when the game becomes a paratext to a different text. To support this argument she uses modding as an example of a situation where the centrality of the game as primary text shifts. She points out that mods are traditionally seen as reshaping a player's experience of a game in some way, be it by changing a game's audiovisual representation or changing the options available to a player. However, she argues that mods have the potential to become the primary text when the game serves as a medium for new and different play opportunities. She uses a total conversion mod of *Crusader Kings 2* (Paradox Interactive, 2012) as an example, which replaces the game's original medieval European setting to Westeros, a fantasy world from the *Game of Thrones* books and television show. The total conversion mod retains the game's original idea of vying for land and power via political marriages and conquests with expanded features, but shifts its setting to a fantasy world. The mod completely reshapes the original game and employs the game engine to create a new experience, effectively flipping the roles of a primary text and paratext. Consalvo's (ibid.) argument is that game studies need fewer central texts and more study on the relations, interconnectivity and contingency of the various kinds of popular culture texts.

2.5 Participatory culture

Modding is seen as a type of *participatory culture*, a concept coined by media scholar Henry Jenkins (1992), which describes a form of media consumption where media audiences become active participants - creating and distributing media rather than merely consuming it.

Jenkins (1992) studied fans and fandom surrounding popular media who create their own culture by appropriating and remixing content from mass culture. Fans of a television show might, for instance, write fan fiction with alternate storylines, while fans of graphic novels might draw fan art featuring their favourite graphic novel heroine and so on.

Jenkins *et al.* (2009) defines participatory culture as having relatively low barriers for artistic and civic engagement, with strong support for the creating and sharing of content and some kind of informal mentorship where the most experienced pass along what is known to those with less experience. Members of participatory cultures believe that their contributions matter and have some degree of social connection amongst each other - or at the very least care what other people think of their creations. (Ibid.)

According to Jenkins *et al.* (2009), there are four key forms of participatory culture; affiliations, expressions, collaborative problem-solving and circulations. Affiliations is about formal and informal memberships of online communities, such as Facebook and gaming clans. Expressions refers to creative content creation such as writing fan fiction or modding. Collaborative problem-solving cultures are about people working together to complete tasks and develop new knowledge, for example, writing and editing content on Wikipedia. Circulations refers to shaping the flow of media by podcasting or blogging. (Jenkins *et al.*, 2009)

Although participatory culture is not a new phenomenon, it has become more commonplace due to rapidly advancing digital media technologies that allow consumers to create, contribute and share their content more easily than before (Jenkins, 2003). Internet and the so called Web 2.0 have played a big role in participatory culture becoming a thing for the masses. Web 2.0 refers to the shift from static web content to a more dynamic Internet that places the user in central role.

O'Reilly (2005), who is attributed to having popularised the term, outlined the second version of the Internet being web as platform, where software applications are built as services rather than as desktop applications. In contrast to early websites, Web 2.0 is characterised by rich user experience (e.g. displaying different states in reaction to user input without the need to reload the page), user participation, dynamic data and harnessing of collective intelligence (ibid.). The new web enabled Internet users to contribute by commenting on articles and create user accounts for greater participation, rather than passively browsing the sites. These changes brought about social media, video publishing platforms and blogging tools such as WordPress each of which put people in the center stage for content creation and publishing.

The relationship between the game industry and players is distinct from other participatory cultures on certain aspects. As Laukkanen (2005) points out, games provide a particularly suitable platform for user participation because they are inherently digital and interactive by nature, and because fan communities are virtually built-in into online multiplayer games. While television and movie studios have a notorious reputation for shutting down fan websites on account of copyright violations (Jenkins, 2003), the game industry has been both able and willing to form a mutually beneficial relationship with players by, for example, releasing software development kits (SDKs) (i.e. editors and tools), fostering communities and hosting modding competitions (Postigo, 2010). The game industry is naturally aware of the economical potential mods and modding may bring, as well as of the long-term value that an engaged player base brings. New game concepts also carry a financial risk, which game companies can mitigate by enabling modders and having them test the grounds for new innovations. Furthermore, modding scenes are seen as lucrative breeding grounds for new talent that serve as potential recruitment pool (Au, 2002; Postigo, 2007).

2.6 Free labour

While it may be difficult to evaluate the exact economic value of mods and modding, it is clear that some game companies stand to gain significant economic value from them. Kücklich (2005) writes that modding benefits game industry by increasing brand recognition and customer loyalty, prolonging the shelf-life of a game, producing new innovations and serving as a recruitment pool.

Games with active modding scenes and innovative content can prolong the shelf-life of a game, increase customer loyalty, and popular mods may even be developed into standalone game titles, as the classic example of Counter-Strike shows (Au, 2002; Kücklich, 2005). In some rare cases, popular mods may even result in an altogether new game genres, as exemplified by the Defence of the Ancients (Dota). Dota is a custom game mode created with the editor tools released with Blizzard's *Warcraft 3* (2002) that changed the game's real-time strategy gameplay into a multiplayer online battle arena (MOBA), where five players in two teams each control individual characters and attempt to destroy the enemy team's base. The game mode became a popular phenomenon and several game studios have since developed their own versions of the game, such as *League of Legends* (Riot Games, 2009), Dota 2 (Valve, 2013) and *Heroes of the Storm* (Blizzard, 2015). All three have also become popular electronic sports games, with *Dota 2's* prize pool in 2016 reaching $20 million.

The case of Dota shows that modding also serves as a source of innovation and proving ground for new game concepts. As Kücklich (2005) points out, without the creativity of modders, game companies would find it difficult to come up with new ideas in a high-risk business like the digital games industry. Dota for instance went through a number of iterations and refinement, which would have been a long and costly venture if undertaken by a game company. As such, modding additionally reduces the costs in research and development as well. Furthermore, modders who have taught themselves valuable game development skills provide a recruitment pool for game companies.

The economic benefits of modding are many and while the game industry stands to gain from it, modders are rarely compensated for their part in generating value for game companies. For this reason, modding has been portrayed from the perspective of post-industrial economy as a form of unwaged or *free labour*. Free labour is a concept developed by Tiziana Terranova (2000) that explains how the various voluntary activities on online platforms increasingly benefits new media companies. Activities such as reading, writing, managing websites, modifying software packages are "[s]imultaneously voluntarily given and unwaged, enjoyed and exploited" (p. 1). Free labour is free both in that the labourers give it out of their own free will, and because they are not compensated by the beneficiary of the labour (e.g. social media companies). Although free labour carries a connotation of exploitation, Terranova (2000) notes that it is not necessarily exploited as many people carry out the labour as exchange with each other (i.e. gift economy).

When popular mods are commercialised, they do not strain the game's budget in advertising because players are already familiar with the product. Given that marketing takes a large portion of a game's budget, brand recognition plays an important role. This is evident in the industry increasingly relying on films, books and other media for game titles. Successful games often lead to sequels, for example Activision's *Call of Duty* title includes a total of 15 games to date.

The concept of free labour builds on the concept *immaterial labour* by Lazzarato, Virno and Hardt (1996), where audiences of online platforms as knowledge workers produce the informational and cultural content of a commodity; informational content involves working with computers and cultural content refers to activities that are not typically considered as work and that define tastes, norms, and public opinion. As immaterial labourers, modders are knowledge workers who utilise the same skill set in making mods as game developers do in making games. As Postigo (2010) puts it, "[c]reativity, sociality, technical know–how and thinking are all products (immaterial products) without which digital games could not exist" (Postigo, 2010, under "Theoretical framings of modding").

Modding is comparable to productive types of waged labour in that those who make the goods/content do now own their creations. Game companies routinely require players to agree to end-user license agreements (EULA), which prohibit any unsanctioned manipulation of the game and/or retain ownership of derivative products for the game company (Kücklich, 2005). In terms of ownership of mods, the position of modders is precarious because mods are often created in conditions that prevent their makers from claiming rights (e.g. receiving payments for their work or asking for donations). Modders are usually left to their activities until the issue of money comes into question.

The game industry not only benefits from the modders' labour, but also from the result of their work - the game mods. As Kücklich (2005) puts it, “it seems a radical departure from the established business models of the leisure industries that the games industry not only sells entertainment products, but also capitalises on the products of the leisure derived from them”. For example in 2009, four years after the launch of the Blizzard Entertainment’s *World of Warcraft* (WoW, 2004) MMOG, Blizzard changed their policy on flourishing UI modding scene surrounding the game. The company dictated that all UI mods must be distributed free of charge and that no solicitations for donations can be shown in-game (Kow and Nardi, 2010). Blizzard did not seek to claim copyrights over the mods, however, only to prevent for-pay content. Naturally, the change in policy meant reduced income for many modders, but the modders were more distraught by the perceived breach of an unwritten understanding between Blizzard and the modders. Meanwhile, Blizzard has incorporated the basic features of popular UI add-ons into the game’s default UI, without the authors of the mods receiving as much as recognition (Prax, 2012).

What exactly is the value of modders’ labour? Postigo (2003) estimated that a team of ten modders working on a total conversion mod for a year contribute roughly 10% of a game’s total development budget or over $520,000 a year, when comparing the modders’ roles to professional game development roles and their annual salaries. Building on that estimation, Postigo (2007) later calculated the approximate worth of 39 total conversion mods made for the very popular World War 2 themed FPS *Battlefield 1942* (Electronic Arts, 2002) to be between $10 and $30 millions in salaries, if they were commissioned as paid content.

2.7 Modders and their motivations

According to Sotamaa (2010), there is no typical modder. Players who become modders are representative of the player base of a game at large. For example, Sihvonen (2009) and Sotamaa (2010), who studied modding in *The Sims 2* (Maxis, 2004) and *Operation Flashpoint* (Codemasters, 2001) respectively, found two very different types of modding scenes: modders in the former (a life simulation game) - comprises largely of female modders, while in the latter (a military shooter) all were male.

If modding is done voluntarily and few modders see recompense for their efforts, what then are the motivations for modding? A common sentiment among modders is that modding is done for fun or as sort of a labour of love (Kow and Nardi, 2010), but a more in-depth look has revealed more specific motivations.

Postigo (2007) identifies three themes that surfaced from his study on modder motivations. First, modding is seen as an artistic endeavor or a creative outlet, through which players can contribute something of beauty. For some there is an additional motivation should other players enjoy the creation as well. Second, modding allows one to identify with a game and thus increase the enjoyment derived from playing it. From this point of view, Postigo notes that modders attempt to make games “their own” by creating custom gameplay elements or drawing inspiration from national or popular culture that have some personal meaning to them. Third, modding is seen as a stepping stone to gaining employment in the games industry by gaining the necessary skills and know-how to make games, although today game development is increasingly taught in various educational institutes. (Ibid.)

Sotamaa (2010) complements the motivations listed by Postigo (2007) with five passions for modifying games: playing, hacking, researching, artistic expression and cooperation. Playing and artistic expression are the same as identified by Postigo, with playing referring to the need to extract as much as possible from the game. Programming-oriented modders approach games as complex code-based systems and often seek to understand how they work, while modders who work on for instance historically accurate mods may spend more time researching the relevant era than actually making something. For modders operating in teams, sharing ideas and the enthusiasm for a common goal may be the main source of enjoyment. Particularly among the younger modders, modding is also seen as a way to enter the game industry profession, while older modders are more skeptical about such prospects. (Ibid.)

The above motivations were similar to those identified by Poor (2014), who carried out a mixed-method study on modder motivations and sense of community. He identified similar list of motivations to those of Postigo and Sotamaa (community, play, digital skills, making and remixing), with the key findings suggesting that modders are both young and old, mod more than one game, have a strong sense of community and find enjoyment in helping others.

2.8 Working definition of digital game modifications

As detailed in the literature review chapter, the existing definitions for digital game mods are somewhat vague and understandably so, given that games and the mods made for them vary greatly in scope and type. Consequently the exact definition of game mods is difficult to pin down.

Given that the purpose of this thesis is to introduce a novel kind of modding scene and the artefacts it produces, a more precise definition is necessary. Therefore in this chapter I present a working definition of digital game mods that allows me to explicate the reasoning on how third-party tools may be considered as game mods. The definition is a synthesis of previous, albeit broad definitions covered in earlier in the literature review (e.g. Scacchi, 2010; Sotamaa and Wirman, 2015). The scope of the working definition presented below, however, is limited to player-made technical interventions contents that in some way alter a game or how it is played.

The working definition is as follows:

A mod is custom audiovisual content or software that changes how a game appears, sounds or is operated, and is connected to a game client, server or data.

Custom refers to mods being "made to fit the needs or requirements of a particular person"[1]. The word custom also entails the notion of being unofficial, due to mods being player-made as opposed to being official released by a game developer.

Audiovisual content refers to game assets such as sprites, textures, skins, sound effects and music, while *software* stands for scripts, programs or add-ons that contain programming code. While audiovisual content alters the aesthetics of a game, software typically affects how a game is *operated*. This includes a change in how a game is played, for instance should a mod add a feature or functionality into game play which is not present in the game. Through functional changes, game rules are typically affected in allowing different actions than what is possible with an unmodified game.

[1] http://www.merriam-webster.com/dictionary/custom

Mods are typically collections of files that are attached or included to the game software, be it as installed components or as files that a game engine reads. As such, they are are connected to game software (both *game client* or *server*) intrinsically and therefore have access to the informative content of games, or *game data*, by default. Game data refers to the informative content games operate on, such as details on one’s player character and the game world. Game data is discussed in more detail in the following chapter.

3 ELITE: DANGEROUS MODDING

In this chapter, I present my empirical study of *Elite* third-party tools through the lens of modding. The overall goal of the chapter is to introduce and analyse *Elite* third-party tools to identify their key characteristics, which were then used to form a typology and categorisation of the tools. Together with the working definition, the resulting typology and categories are in turn employed the elaborate the reasoning in considering the third-party tools as a result of modding and to situate them within the existing typology of game mods.

The chapter is structured as follows:

In the first chapter *Elite* is introduced. The second chapter explains the methodological steps taken in the empirical study of *Elite* standalone tools. The third chapter discusses game data, how it is accessed and what its relevance is for standalone tools. The fourth chapter presents the empirical study of *Elite* standalone tools.

3.1 Elite: Dangerous

Elite: Dangerous is the third sequel to David Braben and Ian Bell's classic space trading and combat simulator game *Elite* (*Acornsoft*, 1984). *Elite: Dangerous* was funded through the crowdsourcing website *Kickstarter* and an early version of the game was made available to backers in December 2013, with the official release following a year later in December 2014. The game is set into the far future of mankind in a cutthroat galaxy, where interstellar travel is commonplace and humanity has spread to thousands of star systems. Galactic superpowers vie for power and the control of the populated space in an interstellar war, and while the alien race Thargoids familiar from the original *Elite* are making their return.

While the first three *Elite* games were single player games, *Elite: Dangerous* is a massively multiplayer online game with an open and persistent game world. Persistent worlds exist independently of any people them and continue to develop without any users interacting with them. *Elite's* game world is a virtual representation of the Milky Way, consisting of 400 billion star systems that are procedurally generated, which is to say that they are created by an algorithm as opposed to being hand-crafted.

Players begin their career as a Commander of a lightly armed trading spaceship, with a handful of credits and freedom to choose from various career paths, such as trading, exploration, bounty hunting, piracy and so on. The game is open-ended with no end or goals as such, other than reaching the rank of *Elite* in combat, trading and exploration. Players can shape the game world through their actions, for instance by doing tasks for the galactic superpowers or supporting the myriad minor factions spread around the populated space.

As is common in MMOGs like *Elite*, modding is either altogether prohibited or limited to certain aspects of the game that do not disrupt the game's delicate balance. Frontier has, however, released an application programming interface for player-made applications to connect to, which will be discussed in the following chapter.

3.2 Method and data

The purpose of this thesis is to introduce the phenomenon of standalone tools as part of the broader modding culture. As mods made for various games are often specific to the game in question, some sort of generalisation must be done so that the outcome is applicable to studying similar tools made for other games as well. Therefore, the most suitable method for analysing *Elite* third-party tools is *typification*.

Typification is the study of types or classification of objects according to their characteristics, a method commonly used in archaeology, psychology and sociology. In typification, data is simplified and generalised by reducing the number and variety of objects by identifying their common and typical characteristics. McKinney (1969) writes that the primary purpose of typification and typologies "is to identify, simplify, and order data so that they may be described in terms which make them comparable" (McKinney, p. 1). Individual differences of the typified objects are disregarded if they are not relevant for the purpose at hand. Although in typification the focus is mainly in identifying the commonly occurring characteristics, Saaranen-Kauppinen and Puusniekka (2006) note that also the exceptional and unique characteristics can and ought to be considered.

3.3 Game data

As an online multiplayer game, *Elite's* game world is shared by all of its players. The game servers simulate the game world and govern the state of the game, ensuring that both the game world and the players inhabiting it are synchronised and experience events in the game world the same way. To achieve this, the game servers are in constant communication with players' game clients by sending packets of data back and forth. In *Elite*, the data includes information on player characters (e.g. player ranks, progression and their location in the game world) and the game world, such as market data, details of star systems and various events taking place in the game world concerning the player. There are three ways the data can be accessed, which are explained below in more detail.

In computing, a common way to access a service from outside is to expose an API, which essentially is a controlled way to interact with a service or its contents without having full access to it. On July 20th 2016, Frontier announced[2] the Player Journal, an official API designed to support the development of third-party applications. The API provides a detailed description of events that take place during play and the information is written into a file on players' computer. For example, when a player docks at a station and visits its commodity market, the contents of the market is written into the journal file as an entry.

Before the player journal API was made available, however, *Elite* players had already discovered other ways to access the game's data. Prior to *Elite's* launch, while the game was in beta testing phase, the only way to automatically collect commodity market data was to use tools with *optical character recognition* (OCR). OCR refers to the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. In short, applications using OCR first take a screenshot of what is shown on the game UI, then detect text and numbers on the image and, finally, convert them back into textual format that the applications can use. The OCR method is prone to errors and misinterpretations of the symbols and therefore it is

2https://forums.frontier.co.uk/showthread.php/275151-Commanders-log-manual-and-data-sample mainly used when other ways to access game data are unavailable.

In November 2014, Frontier released an official *Elite* companion application (companion app from here on) for iPhones and iPads, which was meant for players to view their player characters' progression and access station services, such as the commodity market. Using the companion app did not require one to be logged into the game, as it accessed *Elite* game servers via an API specifically made for the companion app, using the player's login credentials. Soon after its launch, players had reverse-engineered the program and were able to access the API it uses. Although the companion app was later discontinued, its API remains active and in use of third-party tools.

As is common with online games, *Elite* game client writes a network log file, the primary function of which is to help in troubleshooting technical issues players might encounter during play. Although most of the information within the file pertain to the technical operations of the game and not to gameplay itself, the log file includes the player's location which many player tools utilise.

3.4 Types of third-party tools

The tools discussed here can be found on *Elite: Dangerous Codex*[3] (EDCodex), a site dedicated to hosting all kinds of resources relevant to *Elite* players. In addition to third-party tools, it contains links to informative forum threads, community websites, guides, wikis[4] and videos - all of which are titled as entries on the site. Entries other than third-party tools are excluded because they are static information resources that do not provide a functional change to how the game is

[3]http://edcodex.info/

[4]A website or database developed collaboratively by a community of users that allow any user to add and edit content.

played.

On 10 May 2017, EDCodex contained a total of 302 entries, out of which 162 were categorized as tools[5]. Out of the 162, 101 entries are relevant for this study as per the working definition. The remaining 61 were left out for reasons such as being generic applications that are not specifically made for *Elite* (i.e. generic applications that can be configured for games in general) or for being discontinued projects. Each tool is marked by one or more subcategory identifiers, based on the area of game play they touch upon or by the technical solution they use, for example exploration and API.

Common to the remaining 101 third-party tools is that they utilize *Elite* game data in some fashion. In addition, they access, extract, store and/or distribute data for other applications to use, depending on the type of tool in question. At a general level, the tools can be divided into applications used locally on a player's computer, those used online (typically via a web browser) or on mobile devices.

In the following subchapters, the types of *Elite* tools are discussed through examples, with an explanation on what aspects of *Elite* touch upon.

3.4.1 API tools

All *Elite* tools utilise the game's data in some fashion for their gameplay-enhancing functionalities. While some games provide an API with an instant access to its data, in *Elite* the game data is limited to what individual players have access to during play, as explained in chapter discussing game data. Certain *Elite* tools, such as *Elite:*

5http://edcodex.info/?m=tools

Dangerous Market Connector[6] (EDMC), are made for the specific purpose of accessing and gathering game data during play.

EDMC's is an *API tool* that is run alongside the game client and it automatically downloads game data for instance when a player docks at a station, jumps to a new star system or scans celestial bodies in a star system. The data is stored on the player's computer for other applications to use or it is optionally sent to online tools and services that are designed to store game data sent by individual players.

While EDMC only accesses game data and sends it onward, other API tools have additional built-in features. *EDDiscovery*[7] for example is a tool that tracks the player's jumps through star systems and, using a trilateration algorithm, plots the visited star systems into a virtual coordinate system. EDDiscovery is particularly useful for players who wish to map and visualize their exploration journeys, as Elite neither stores such information nor provides such mapping features.

[6] http://edcodex.info/?m=tools&entry=150

[7] http://edcodex.info/?m=tools&entry=10

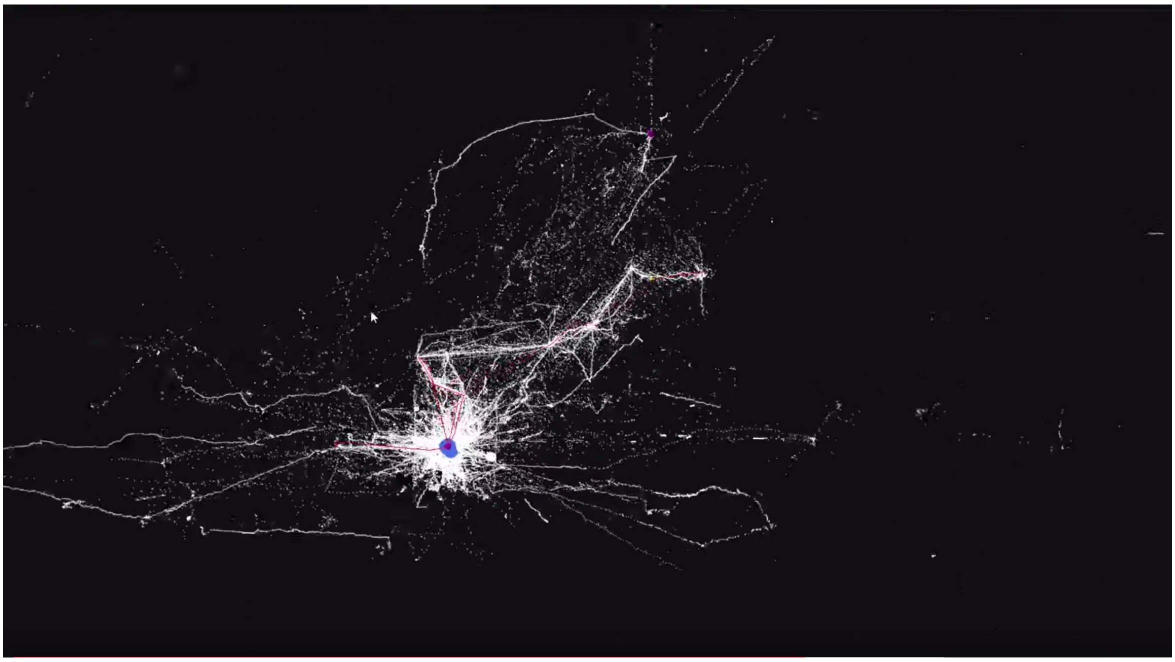


Image 1: Visualisation of a player's travels through the virtual galaxy. The white, central area is the populated area of space.

3.2.2 Online tools

The game data gathered by API tools commonly feature an option to upload the data into online repositories, or *databases*. Databases are organized collections of large and complex data sets that are generally used by more complex online services, such as search engines and web stores. *Elite: Dangerous Database*[8] (EDDB) is perhaps the most comprehensive collection of the *Elite* game data, as it combines data from two other databases that focus on specific types of data; *Elite: Dangerous Star Map*[9] (EDSM) gathers mainly data related to star systems and celestial bodies in them and *Elite: Dangerous Data Network*[10] (EDDN) stores namely commodity market data.

8https://eddb.io/
9https://www.edsm.net/
10http://edcodex.info/?m=tools&entry=24

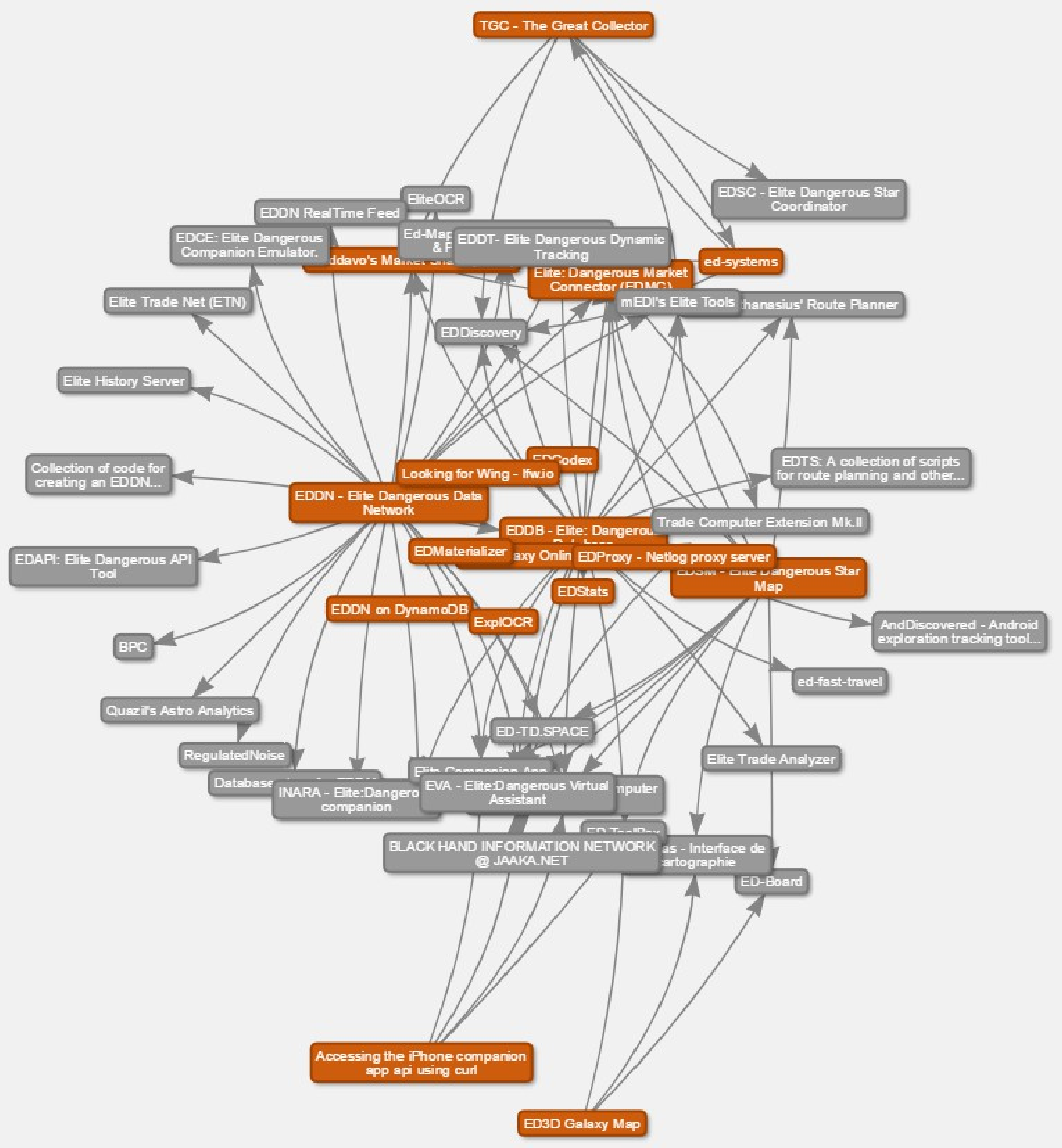

TGC - The Great Collector
EDSC - Elite Dangerous Star Coordinator
EliteOCR
EDDN RealTime Feed
EDCE: Elite Dangerous Companion Emulator.
Ed-Map
EDDT- Elite Dangerous Dynamic Tracking
ed-systems
Elite: Dangerous Market Connector (EDMC)
Elite Trade Net (ETN)
mEDI's Elite Tools
hanasius' Route Planner
EDDiscovery
Elite History Server
Collection of code for creating an EDDN...
EDCodex
Looking for Wing - lfw.io
EDDN - Elite Dangerous Data Network
EDTS: A collection of scripts for route planning and other...
EDDB - Elite: Dangerous
Trade Computer Extension Mk.II
EDAPI: Elite Dangerous API Tool
EDMaterializer
EDProxy - Netlog proxy server
Star Map
EDStats
EDDN on DynamoDB
ExplOCR
AndDiscovered - Android exploration tracking tool...
BPC
ed-fast-travel
Quazil's Astro Analytics
ED-TD.SPACE
RegulatedNoise
Elite Trade Analyzer
Database
INARA - Elite:Dangerous companion
EVA - Elite:Dangerous Virtual Assistant
mputer
BLACK HAND INFORMATION NETWORK @ JAAKA.NET
as - Interface de cartographie
ED-Board
Accessing the iPhone companion app api using curl
ED3D Galaxy Map

Image 2: EDCodex's graph illustrating the ecosystem of Elite tools. Orange items are API tools or databases that relay game data onward to other tools and services. The lines show which tools are connected and the arrows indicate which way data flows.

Both API tools and databases play a crucial role in the ecosystem of *Elite* tools. Any tool that cannot harvest game data on their own rely first on API tools harvest and send data to databases, which are essentially repositories for the crowdsourced game data by players. As shown in the image #, APIs and databases (in orange) are central for a variety of tools, supplying other tools and services with game data via their own APIs.

While some databases only store game data and distribute it onwards, others have additional features built on them. For example, *Elite Galaxy Online* (EGO) comprises of a database and a website built on it, which keeps records of notable discoveries in *Elite* and attributes them to the Commanders who submitted the data. The site organises the data into various categories, for example the most massive black holes discovered, while in *Elite* the first discoveries are only shown upon viewing individual star system maps. As such, EGO essentially stands in for the game's hall of records or achievements.

3.2.3 Trading tools

MMOGs often feature some kind of a virtual market system, where players can purchase equipment, craft materials or other goods, and can sell loot acquired during play. In *Elite*, however, trading can only happen between players and commodity markets, with no possibility for player-to-player trading.

Nevertheless, trading is a core part of *Elite's* gameplay in terms of making profit. The myriad populated star systems have different kinds of economies, which forms the basis of supply and demand. For instance, mineral extraction systems export their yields to systems with refinery economy, while high tech economies export advanced machinery to agricultural systems. While profitable trade routes between two locations can exist, the more lucrative ones often require trading between three or more locations. *Elite* is somewhat ambiguous in informing the player on where to trade, as it only shows where goods are exported to or from , and only as one-way routes. Much like with exploration, the game does not keep track of previously visited commodity markets or their details. Furthermore, as players trade in the various space and planetary stations, their supply, demand and respective prices vary accordingly; prices drop and the more rare commodities might run out of stock, which means that a single trade route will be profitable for only so long.

Given that the in-game ways of figuring out where to trade are so limited, it is no wonder that most of the tools made for *Elite* concern trading. The website *Inara*[11] for example allows one to mark favourite stations, suggests the most profitable export/import goods and where to take them. It also displays the distance of stations from the star, which is important because the closer a station is to the star the faster players make profit, because the most distant stations may take close to an hour to reach. In addition to trading functionalities, Inara also features a more in-depth page for player characters, their fleet of ships and forming of player groups. By default, players must update their information on the site manually, but by using *Inara for EDMC*[12] plugin the process is automated; EDMC automatically transmits new player character data into the site.

Other web tools have more complex functionalities for searching and filtering through game data than Inara. Some databases maintained by players also feature a website that allows players to access its contents directly, as in without some other application. For example, *Elite: Dangerous Database* (EDDB) allows players to search for particular commodities that are difficult to find or to discover "multihop" trade routes involving three or more locations. The site also shows profit per unit sold, totals for individual location and the total gains for the whole route, which is a much more robust way of planning trading than the game affords.

[11] https://inara.cz/

[12] http://edcodex.info/?m=tools&entry=436

3.2.4 Player character tools

In MMOGs like *Elite*, players characters are commonly specialised over the course of gameplay through some kind of a skill or talent tree mechanic. The specialisations allow players to customise and enhance their roles, for instance a warrior character might want to focus on dealing as much damage as possible or to build his character to withstand damage dealt by enemies.

In *Elite*, players always are piloting a spaceship (with the exception of planetary exploration vehicles) and specialisation roles are achieved by outfitting their spaceships with various modules. In this sense, spaceships are extensions of the player character (if not the character itself. The spaceships are designed for specific roles, such as combat, trading and exploration, and perform poorly in others - with the exception of multi-purpose ships that are capable of taking hybrid roles. For example, the spaceship *Vulture* is ideal for combat because of its maneuverability and powerful weapon module slots, but it is not well-suited for trading due to having only a few module slots that can house cargo racks.

Although the outfitting interface in *Elite* has been revised since the game's launch, there are certain aspects of it that have created the need for external tools for designing spaceship builds. For example, one cannot design a build prior to purchasing all the necessary modules. Instead, modules are bought one at a time, which makes it difficult to realise the total cost of the build and other important details, such as its total power consumption. The latter is a critical aspect of the build, because should the total power consumption exceed the power output of the spaceship's power plant, a player might find herself in a situation where the ship cannot deploy weapons without losing power to critical modules such as thrusters or life support.

To overcome the limitations of *Elite's* outfitting interface, *player character tools* like *Coriolis*[13] were developed. Coriolis is a web-based tool that allows players to plan outfitting builds with far more information and options than what the game interface permits. The tool allows players to experiment with builds prior to purchasing any modules or the spaceship itself. Builds made in Coriolis can be compared with each other, exported to other tools and shared with other players via a link.

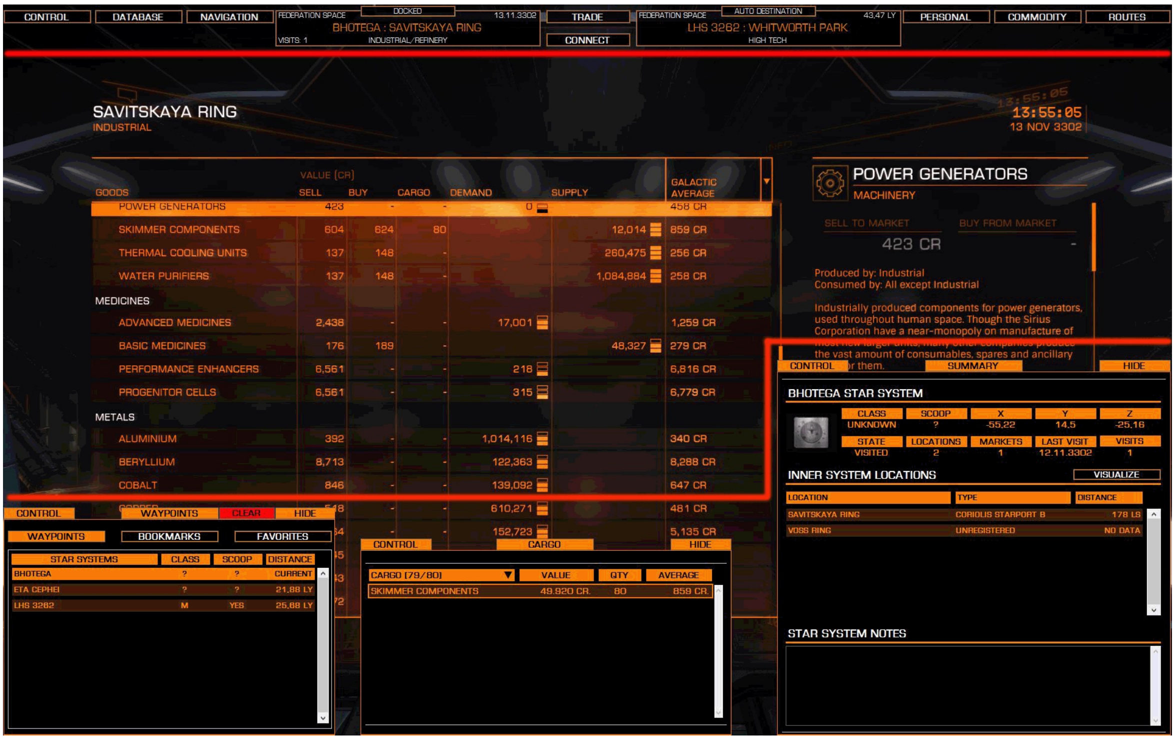
3.2.5 Overlays

Modding in MMOGs is often prohibited or restricted to so that only UI mods are permitted, so as to maintain equal and balanced game experience for all players. UI mods allow players to rearrange and customise game UI's or display additional information and controls that might otherwise be hidden from the player. When modding of the game software is not supported (as it is in *Elite*), the functionality of UI mods can be accomplished within the game by using tools that have an *overlay* feature. Overlays are typically transparent windows or layers that show additional information or controls over the actual content of games and other software. They can be built-in to software or, as is in the case of separate applications like third-party tools they can be set to overlay another program.

In *Elite*, only the *Trade Computer Extension Mk. II*[14] (TCE) features an overlay mode and, like its name suggests, is mainly a trading tool. TCE recognises when the player opens the commodity market interface and automatically shows its panels with extra controls over the trading

[13] https://coriolis.io/
[14] http://edcodex.info/?m=tools&entry=327

interface. Similarly to other trading tools, it allows trade route planning, but without the need to switch to another application - a feature that is particularly useful for players who use virtual reality headsets. In case the other methods of harvesting game data are inaccessible for some reason (i.e. outage in the API), TCE users can revert to using the OCR method that is built-in into the application.



GOODS	VALUE (CR) SELL	BUY	CARGO	DEMAND	SUPPLY	GALACTIC AVERAGE
POWER GENERATORS	423	-	-	0		458 CR
SKIMMER COMPONENTS	604	624	80		12,014	859 CR
THERMAL COOLING UNITS	137	148	-		260,475	256 CR
WATER PURIFIERS	137	148	-		1,084,884	258 CR
MEDICINES						
ADVANCED MEDICINES	2,438	-	-	17,001		1,259 CR
BASIC MEDICINES	176	189	-		48,327	279 CR
PERFORMANCE ENHANCERS	6,561	-	-	218		6,816 CR
PROGENITOR CELLS	6,561	-	-	315		6,779 CR
METALS						
ALUMINIUM	392	-	-	1,014,116		340 CR
BERYLLIUM	8,713	-	-	122,363		8,288 CR
COBALT	846	-	-	139,092		647 CR
[illegible]	[illegible]	-	-	610,271		481 CR
[illegible]	[illegible]	-	-	152,723		5,135 CR

POWER GENERATORS
MACHINERY

SELL TO MARKET	BUY FROM MARKET
423 CR	-

Produced by: Industrial
Consumed by: All except Industrial

Industrially produced components for power generators, used throughout human space. Though the Sirius Corporation have a near-monopoly on manufacture of most new larger units, many other companies produce the vast amount of consumables, spares and ancillary for them.

STAR SYSTEMS	CLASS	SCOOP	DISTANCE
BHOTEGA	?	?	CURRENT
ETA CEPHEI	?	?	21.88 LY
LHS 3262	M	YES	25.68 LY

CARGO [79/80]	VALUE	QTY	AVERAGE
SKIMMER COMPONENTS	49.920 CR.	80	859 CR.

BHOTEGA STAR SYSTEM

CLASS	SCOOP	X	Y	Z
UNKNOWN	?	-55.22	14.5	-25.16
STATE	LOCATIONS	MARKETS	LAST VISIT	VISITS
VISITED	2	1	12.11.3302	1

INNER SYSTEM LOCATIONS

LOCATION	TYPE	DISTANCE
SAVITSKAYA RING	CORIOLIS STARPORT B	178 LS
VOSS RING	UNREGISTERED	NO DATA

STAR SYSTEM NOTES

Image 3: TCE displays additional control panels, marked with red lines on the top and bottom edges of the game view.

3.2.6 Mobile device tools

Modern games, particularly online games, are increasingly played with multiple devices. MMOGs like *Elite* often come with companion apps that provide supplementary functions to gameplay or simply allow players to view game world and player character information. In some games, the role mobile devices take in gameplay is more prominent, for instance the FPS *Battlefield 4* (Electronic Arts, 2013) features a tactical

Commander mode in which one player from each team supports and assigns objectives to the players on the battlefield - a role that can be played on a tablet device as well. As mobile application development is accessible to regular users, it is unsurprising that standalone tools include *mobile device tools* as well.

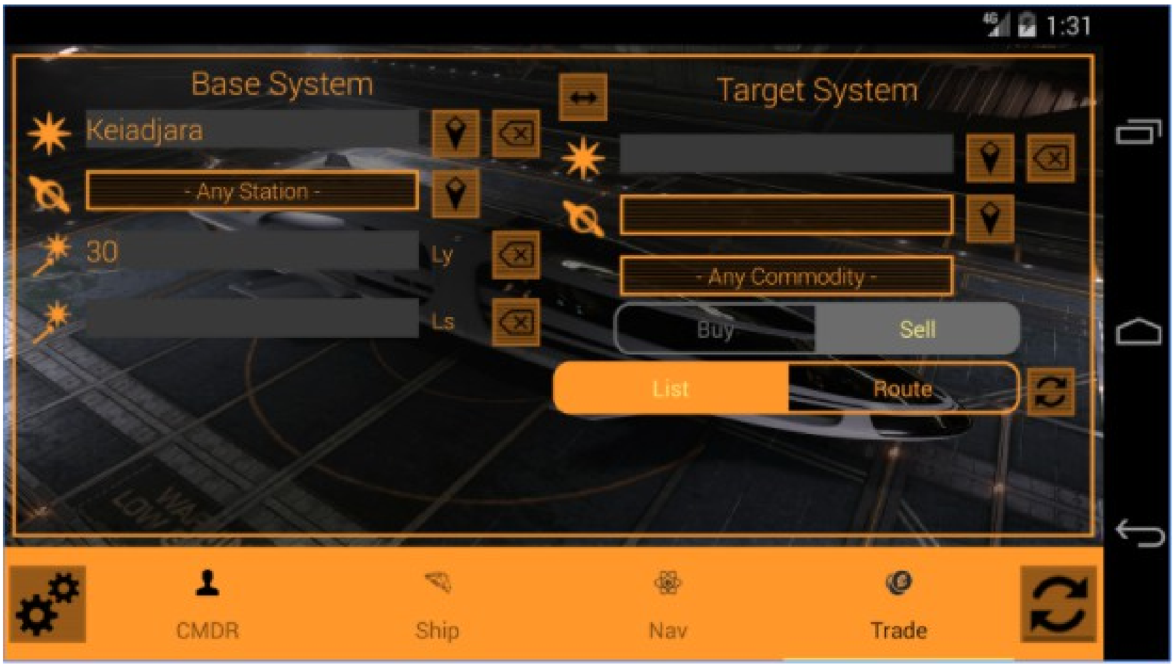


Image 4: EVA's mobile interface for planning trade routes.

Elite: Dangerous Virtual Assistant[15] (EVA) is one such tool (for Android and iOS devices) that keeps track of a player's movements and the data she gathers during play. In that it is similar to *Elite's* companion app, but EVA can also be used to discover trade routes. Similarly to overlays, mobile tools remove the need to switch from the game client to another application.

[15] http://edcodex.info/?m=tools&entry=376

4 ELITE MODDER SURVEY

In this chapter, the *Elite* modding scene is approached from the point of view of the third-party tool developers.

The first subchapter explains the methodological steps taken in gathering the data and how it was analysed. The second subchapter provides an overview of the survey design. The third subchapter reports the results.

4.1 Method and data

The data from the developers of third-party tools was gathered with a qualitative online survey. Interviews were also considered, but given that the target group consists of online game players potentially from all over the world, an online survey seemed the most suitable method. With a self-administered survey the respondents may choose when and where to take it, and complete it at their own pace. Furthermore, the responses to a survey tend to be more thoroughly thought-out compared to interviews (Fielding *et al*., 2008).

Survey is essentially a way to gather information by asking questions (Wolf *et al*., 2016). Survey research has been quick to embrace new technologies, lately the Internet and mobile devices. Online surveying has several advantages in comparison to offline research methods, namely being cost- and time efficient (Fielding *et al*., 2008). By using one of the many free survey tools, costs associated with conducting survey research are minimal and distributing the form via invitation links is effortless. Survey tools also store and process data automatically, which reduces errors in coding the data. Online research methods are also useful because the Internet provides access to potentially vast, diverse and global audience, and practical when studying online special-interest groups (bid.) - as this thesis does.

Online surveying has potential drawbacks, such low response rate, participants quitting the survey before completing it or answering questions selectively (Fielding *et al.*, 2008). The researcher can mitigate such disadvantages by paying attention to the survey design: wording, visual design and layout choices form the main method of communication between the researcher and respondent. Questionnaires that offer a friendly user experience can raise the motivation of respondents and improve the quality of responses. Compared to in-person interviews, qualitative online research may suffer from a lack verbal or body language cues (Fielding *et al.*, 2008).

Because the study relies on the voluntary participation of the respondents, the validity of the data is questionable in the sense that it might not be representative of the whole target audience (Miettinen and Vehkalahti, 2013). However, as the survey presented here is qualitative in nature, it was more important to reach the right kind of respondents rather than aiming for a large number of respondents.

4.2 Survey design

The survey (attached in Appendix 1) consisted of 27 questions that were divided into sections according to the following themes: respondent background, *Elite* play habits, modding in general and in context of *Elite*. Participation in the survey was anonymous. Although several respondents identified themselves by indicating which tool they had created, they are anonymised in reporting the results. The survey was followed by more in-depth questions to those who chose to leave their email address as a point of contact.

As the platform for conducting the survey, I chose *Google Forms*, a free online survey tool by Google. Google Forms is ideal for designing and publishing simple online surveys. For more complex studies, especially those that deal with quantitative data, Google Forms might be unsuitable as it lacks analysis tools. For the survey presented here this was not an issue Google Forms produces mainly qualitative data, with the exception of questions on the respondents' year of birth and time spent playing and modifying. As Google Forms was familiar to me from prior study-related uses and given that the intended respondents are a group online players, choosing Google Forms as the platform made sense.

The target group for the thesis were players who had created third-party tools for *Elite*. Participants were recruited on the official *Elite* forums, EDCodex and on *Reddit*, a popular discussion site. The form was opened for answers on December 1st 2015 and kept open until the end of January 2016, by which time the influx of answers had stopped. Before making the survey public for answers, a test round was conducted.

Following an advertisement on Reddit, feedback was given that revealed two problems in the survey design: the wording and general approach imposed the concept of modding on the respondents, rather than inquiring how they view their activities in creating third-party tools. The feedback did, however, serve a purpose in highlighting a disconnect between the ways modders define mods and modding, and how academic literature defines them. To rectify the oversight, respondents who had included their email address' were asked how they define mods, modding and their productive activities in *Elite*.

4.3 Results

A total of 16 *Elite* players responded to the survey, out of which six indicated that they had not created tools for *Elite*. As the survey was intended for authors of *Elite* third-party tools, those six were excluded from the results. It is worth noting that the excluded respondents participated in the survey in hopes of bringing more attention the third-party tool development and to gain more tangible support by Frontier, ideally in the form of a proper API.

In the following chapters, the results are reported by theme according to the structure of the survey.

4.3.1 Background

The birth years of the respondents were between 1949 and 1986, with an age range of 30–67. Divided into groups, four were 30–35 years old, three 38–42 years old, two aged 50 and the oldest 67 years old. All respondents are males and mostly from European countries; five are from the United Kingdom, two from Australia, and one from France, the Czech Republic and Sweden each. Half of the respondents coming from the United Kingdom is possibly due to the original *Elite* being a British production.

Most responses to question about education reported their level of degree, ranging from high school to a Master’s degree. Three respondents specified to have a Master’s degree in computer or software engineering. With the exception of one business consultant and a freelancer with no further details given, all work in software development in some capacity, including chief technology officer, project manager, software engineer and web developers. Only the respondent who works as a freelancer reported to have working experience in the games industry.

The age of the respondents was somewhat surprising, mainly in that all were 30 or over in age. The older age group represented in the data can potentially be explained by them being returning fans of the original *Elite*. The commonality of information technology and software development answers to the education and occupation questions was not so surprising: the development of full-fledged scale standalone applications requires a more comprehensive command of the programming language used (as opposed to writing scripts that build on existing game software).

4.3.2 Play

To the frequency of play question, two responded that they play *Elite* daily, seven weekly and one player once or twice per month. The estimated time spent playing was reported in two ways; in total hours played and hours in relation to the frequency of play. Those that play daily spend 1-2 hours at a time in game, weekly players 2-10 hours and the sole monthly player estimated 6 hours. Those that reported the total hours spent in game play the game weekly and the play time in hours vary between 130 hours and over 2000 hours. Two wrote that they had started playing *Elite* prior to its launch, which means that they were backers of the game on Kickstarter as only the backers were able to play the game prior to its official release.

Most commonly stated reason for playing *Elite* was love for space, science fiction and spaceship simulators. Four had played the original *Elite* or another of its sequels, with one respondent writing that the original *Elite* got him interested in computing which also resulted in his career as a web developer. For the two respondents interested in exploration, *Elite's* vast game world and the freedom it presents were central to their interest in the game.

4.3.3 Modding

In response to how the respondents came to modify games, the most common theme was that the games they play lacked features that fit their needs. Two expanded on this by writing that because they have the technical skills, it made sense for them to make something to enhance their gameplay. Another two added that the desire to understand how things work and hacking for the fun and achievement played a part in how they came to create custom content for games. The developer of Coriolis wrote that he had come across *Elite: Dangerous Shipyard*, another spaceship outfitting tool, and thought it could be drastically improved and expanded with additional features. Another respondent stated that he tends to create peripheral applications rather than directly modifying games, because the latter "tends to land one in legal trouble."

As for why the respondents create custom content improving game experience and enjoyment was the most common theme. The developer of Inara added that if a mod is also helpful for others, it is an added motivational boost for them. The other themes consisted of making games easier and more immersive, technical challenges and experimenting with tools and technologies they otherwise would not use in their profession.

On the importance of being able to mod the games that are played, the respondents were divided. Half of the responses leaned towards it not being important, with some adding that should they perceive a game to lack a feature they want and if they have the skills to add that feature, the possibility to mod is important. Three respondents wrote that modding is important to them, with reasons such as becoming invested in the game and making it more fun to play with additional features. However, it was pointed out that it really depends on the game in question:

> “Though I have created third-party apps for a number of games, there are others which I have no such inclination towards.
>
> Ironically enough, I've recently started playing *Minecraft*, a game with an active modding community encouraged by the developers; but I have no desire to mod it myself!” (R6)

The respondents described the difference between playing and modding in various ways, with most the responses noting some kind of difference between the two. Playing was described as what or how the game developers want you to play, with modding being about taking the game to the next level and making it more your own. Another commented that playing is only about enjoyment and through modding something is contributed to the community. It was also pointed out that there is overlap in playing and modding, especially when playing the game to gather information and ideas for tools development. However, two respondents felt that that there is no difference between playing and modding:

> “For games like *Elite*, my reasoning is somewhat like this: there's a game universe ‘out there’, virtually existing, and one way of interacting and exploring that universe is through the game client. But another way is to write a third-party app that deals with some of the data from that universe, lets you analyse it, lets you manipulate it; almost making the game a little bit more real. All of which is a very long-winded way of saying that they're both playing, but in different ways ;-)” (R6)

To the question on how modding has affected gaming habits and enjoyment, three wrote that there is little or no effect at all. For the rest, modding enhances the game, making playing more fun and in some cases, there has been a rather significant impact on gameplay. For example, one respondent commented the following:

> “In the case of the trade tool I made for Elite, it had a pretty massive effect on my gaming. Rather than just playing *Elite* for what happened in game I found I was getting a lot of enjoyment from flying about the galaxy updating prices on my site and looking for locations missing from my database.” (R7)

The “freshness” of *Elite* game data is important particularly for trading tools, because the prices of commodities are dynamic in that they react to player activities. For trading tools to be reliable, the market data they operate on must be kept up to date by visiting the commodity markets and uploading their current data into databases.

Being able to make content and applications for games can make players more invested in the games they play, while lack of support for player productivity can lead to a waning interest:

> “Writing any significant third-party app is a heavy investment of time, so I tend to feel more invested in the games I've written such things for. When the developers of a game neglect or actively seek to hinder such development (as has sadly happened in E:D) I find myself wandering elsewhere to engage my interests.” (R6)

Eight out of the then respondents reported having created mods for other games, such as character skins, maps, web-tools and plugins. Modified games included single player games and other multiplayer online games. One respondent had created several third-party apps and websites for the MMO *EVE Online* (CCP Games, 2003) for trading purposes, which served as an inspiration for creating similar tools for *Elite*.

4.3.4 Modding in *Elite*

Each respondent had created at least one third-party application for *Elite*, with two having developed multiple applications or contributed programming code to tools made by others. In describing the circumstances of developing their tools, a common theme was dissatisfaction with *Elite’s* existing features, particularly those pertaining to trading and navigation. As one respondent phrased it, he was frustrated with the inadequate trading tools and attempted to find whether there were existing tools that would work for his needs. Having discovered that such tools existed, he decided he could radically improve them and proceeded to make a tool of his own.

A common functionality among the respondents’ applications was that they automate the tracking of game data during play. Those that had made a trading tool described that their applications essentially serve as a replacement for writing market prices on paper or in a spreadsheet, while exploration tools function as a kind of captain’s logbook. This theme continued in the answers to the following question, which inquired about the respondents' stance on whether the function their applications perform is about customising their play or whether it a feature missing in game. Each respondent wrote that their applications stand in for features missing in the game, with two adding that their applications also serve to customise their play in *Elite*. In the words of one respondent:

> "It still surprises me that the trade tools in the game are so inadequate. I never expected my site to have a use beyond alpha. Since release there have been some months where I've had over 1.2 million sessions says to me there's a need to improve the in-game tools. And mine is just one of many trade sites.
>
> The ships have computers on them, if I was a pilot in the game world the first thing I'd have done is write a database on my ships [sic] computer to save price data from each place I visited for later perusal." (R7)

Alpha refers to an early state of a game in which the game is typically tested by the developers of the game, but in case of *Elite* backers of the game on Kickstarter were also invited to participate. The respondent expected that his website would be made redundant by the finalised game, but instead the website continued to receive large amounts of data queries which indicates a substantial use of the site. Other respondents echoed the sentiment that they expected or wished their tools to be made redundant by changes to the game.

The skills and tools used in making the applications included a number of programming languages and frameworks. Schema design, public relations and OCR theory were also listed as skills used. Half of the respondents work on their applications weekly, one daily and three more rarely. Most respondents indicated that their applications are continuously worked on, with time spent on them ranging between two and 15 hours per week. Only one wrote that he is not working on his tool anymore, stating that the time used in creating his tool had taken roughly 48 hours in total. Another wrote that nowadays he mainly acts as an administrator of his website and only works on the site when he receives feedback or feature requests from other players, which is when he spends about 8 hours a week improving the website.

The responses to the question whether modding in *Elite* should be supported varied, with half of the respondents wanting modding support in some form and three being opposed to being able to modify parts of the game. Those that gave more verbose answers pointed out that an official API meant for the use of third-party applications would be ideal. One was eventually released in form of the player journal API after the survey was conducted.

The respondents described the *Elite* modding scene as small and nascent, but also as being very active, creative, open and healthy considering its size. Its members are in frequent contact with each other to talk about their applications, share information on them, discuss new ideas and, in some cases, to connect each other's applications to share game data between them. In general, the collaboration within the community of player-developers seems to be quite common, as they also test each other's applications to give feedback and find errors to be fixed. Several respondents also wrote that they have contributed programming code to applications made by someone else. Some development projects have been done as a team from start:

> “EDDN was collaborative from the outset, with [a colleague] providing a lot of the momentum, me bringing my programming skills and experience from EVE, a friendly and generous hosting provider donating a VPS[16], and more recently a couple of other programmers getting involved sharing the load.” (R6, edited to anonymity)

Lack of support to their activities by Frontier was often brought up. As the survey was conducted before the player journal was implemented, many respondents specified the lack of a proper API as an indication of a lack of support by the game company. The game company also drew criticism for not encouraging the efforts of the tool developers, referring to the lack of communication with the modding community over the course of *Elite’s* history.

4.3.4 Views on mods and modding

The responses to the follow-up questions on mods and modding were quite uniform in that a mod alters a game, its gameplay, assets or contents somehow. According to the responses, it appears that the primary prerequisite for a mod is that it must modify the game software or its contents. Therefore, standalone applications that do not “directly” affect game software or its files are regarded as extensions, tools or as some of “extended UIs”. As such, none of the respondents considered that their tools warranted the term game mod because they do not alter *Elite’s* assets or contents explicitly.

One respondent further elaborated on extensions and tools in relation to mods. In describing extensions, he made a distinction to mods in that extensions do not directly modify a game or its files, but that they instead “read” files and provide “enhanced features directly related to the game, as some kind of extended UI”. For him, third-party tools are

[16] Virtual private server, used to host online services such as databases and websites.

applications that may partially use game files or game data without being directly related to the game. (R3)

As for defining modding, none of the respondents considered themselves to be modders for making third-party tools. Most referred to their definitions on mods, indicating that because they do not make game mods as per their definition, they consequently are not modders. Here the terminologies of players and academia on mods and modding deviate, as it appears that modding as an umbrella term to capture all sorts of player productivity may not be adequate - at least in the eyes of the players.

5 DISCUSSION AND CONCLUSIONS

The aim of this thesis was to study the creation of *Elite* third-party tools through the lens of modding and to find out what their impact is on how the game is played. The study also inquires about the players' motivations for developing the tools and, as an addendum to the original approach, the players' views on mods and modding are explored. The research questions were:

- RQ1: What are third-party tools and how do their developers relate to them?
- RQ2: What are the developers' motivations?
- RQ3: How do the developers *Elite* third-party tools define mods and modding?

As the first research question consists of two parts, the answer to it requires combining the analysis of *Elite* third-party tools and the survey results. Building on the working definition, third-party tools can be considered as extensions of the game UI in the form auxiliary applications, as they essentially recreate improved versions of the game UI's features and complement the game by adding new ways of utilising game data. As such, third-party tools share characteristics with UI mods that modify a game's information management dashboard (Scacchi, 2010). Although in technical terms third-party tools do not modify the game, they arguably have a meaningful function in how *Elite* is played by providing greater control over the game data than what is possible with just the game. By harvesting and particularly crowdsourcing *Elite's* game data and hosting it on their own servers, players essentially recreate parts of the game world to which the tools are custom interfaces.

The respondents wrote that their applications are mainly intended to amend perceived shortcomings in the game's design. In addition to providing alternate gameplay functions, the possibility to track, visualise and manipulate game data also contributes to players' sense of immersion by making the game world more tangible. Some respondents also wrote that they expected or even hoped that later changes to the game would make their tools redundant. From this perspective, the tool development can also be seen as players giving feedback to the game developers. To answer the latter part of the first research question, the tool developers consider their activities as means to participate in the game's development.

The players developing third-party tools are committed players, whose motivation is to improve the game according to personal needs with an added incentive should other players find the applications useful. This is in line with Postigo's (2003) observation that one of the most important motivations to mod is "the sense of community they derive from the experience" (ibid., p. 599). The respondents also have a history of modding multiple games, which together with their motivations reinforces Poor's (2014) findings. All respondents have a professional software development background and therefore have existing set of skills to create standalone applications, although one respondent took the opportunity to practice a less familiar programming language. Comparing the motivations *Elite* tool developers to those identified by Sotamaa (2010), the most fitting one is unsurprisingly Play, or the "need to extract as much as possible from the game" (ibid., p 8). Additionally, some of the tool development projects have been collaborative from the start and others were later made to work with each other in sharing data, which resonate with the Co-operation motivation (ibid.). To answer the second research question, the third-party tool developers are motivated to improve the game for both themselves and others, and increase their enjoyment of the game.

The results to the third research question provided perhaps the most interesting findings of this study. The respondents see modding as playing, but as something is produced or contributed, the activity takes on new new meanings and characteristics of work. This does not diminish their fun or enjoyment, on the contrary it increase it as players get to make the game more their own or take it to the next level. To answer the research question on that part, players see modding an extension of play through which they can take the game beyond its designed confines and increase their enjoyment of the game.

As for mods and how the respondents define them, they appear to have a clear idea on what constitutes a game mod and define them as creations that directly modify the game software or its contents. Based on this definition, the respondents do not see third-party tools as mods and, consequently, themselves modders in context of creating standalone applications. It appears that players have a system-centric view on what a game is (i.e. a game is its software) and interacting with it is what playing is for them. To complement the answer to the third research question, the players see modding as peripheral activity to playing, mixing elements of play and work due to something being produced and contributed to the game, as opposed to merely consuming the game according to the game developers' vision.

From the perspective of free labour, Frontier arguably benefits greatly from the tool developers' efforts. As the player tools complement the shortcomings of the game's features, they relieve the game developer of (at least some of) the pressure to add similar features to the game. After all, maintaining the player journal API requires far less resources from the game developer than if they were to implement the player tools' features into the game. Interestingly, as professionals in software development, the tool developers utilise the same set of skills that they use in their professional line of work, but in context of *Elite* they give their expertise for free and thus practice what Terranova (2000) calls "gift economy". It is also notable that the implied exploitative connotation in free labour seems remarkably absent here. As Terranova (2000, p. 48) notes, free labour is not necessarily exploited labour as people often carry out their work in exchange with each other. Considering events surrounding the shutdown of player tools, it is the tool developing community clamoring for more support and engagement from Frontier, rather than the company fostering player productivity in hopes of reaping some sort of value from their efforts.

The main contribution of this thesis is to the discussion on what modding is and to complement the existing typology of mods. Although third-party tools may not be considered as mods proper in that they do not alter the formal structure of the game, they arguably provide essential game-enhancing functionalities to games like *Elite* where managing the game's informational content is in a central role of gameplay. Furthermore, playing digital games increasingly involves using auxiliary applications (e.g. mobile companion apps) and player-made third-party tools effectively occupy the same role as the official applications do. This promotes the notion that playing games is more than interacting with the main game object and that modifying games is more complex than merely altering its formal representation.

Even though third-party tools are arguably reliant on game data for their functionalities and exist largely because of *Elite*, in some cases and for certain players the centrality of *Elite* as the main text is not so clear. For example, one of the respondents wrote that the web application he made drastically changed the way he plays the game, so that he would play to visit locations in the game just to update their information on his web application. For him and in this particular case, the application became the central text, rendering *Elite* as the supporting text. As Consalvo (2017) points out, there ought to be more study of "relatedness, interconnectedness, and contingent nature of many kinds of popular culture texts" in game studies and fewer central texts.

Considering the limitations of this thesis, the overall survey design was perhaps too broad and could have been iterated further to be more compact and focused. In retrospect, the survey was conducted perhaps a little too early in the thesis process. Furthermore, even though the survey was tested and amended prior to carrying it out, the wording in some of the questions could have been refined further to reduce the amount of shallow yes/no answers and, consequently, the need for follow-up questions. The survey questions' wording and overall approach assumed the broader academic view on modding, instead of asking how the players see it. This oversight was a blessing in disguise, however, as it helped reveal the drastic difference in the way academia and players view mods and modding.

Future studies could take a closer look at that disconnect and perhaps seek to bridge it. It would also be interesting to see whether the findings presented here would be corroborated by studies in other modding scenes, namely whether modders at large see their activities in the same light as those in *Elite's* modding scene do. Approaching a more complete typology of game mods relies on establishing a more exact definition for modding, and players' input should be included in the process.

REFERENCES

Au, W.J. 2002, *Triumph of the mod*. Available at: https://www.salon.com/2002/04/16/modding/ (Accessed: 8 September 2017)

Consalvo, M., 2009. *Cheating: Gaining advantage in videogames*. Mit Press.

Consalvo, M., 2017. When paratexts become texts: de-centering the game-as-text. *Critical Studies in Media Communication*, 34(2), pp.177-183.

Fielding, Nigel G., Raymond M. Lee, and Grant Blank, eds. *The Sage handbook of online research methods*. Sage, 2008.

Jenkins, H., 2006. *Fans, bloggers, and gamers: Exploring participatory culture*. nyu Press.

Jenkins, H., 2003. Quentin Tarantino's Star Wars? Digital cinema, media convergence, and participatory culture. *Rethinking media change: The aesthetics of transition*, pp.281-312.

Jenkins, H., Purushotma, R., Weigel, M., Clinton, K. and Robison, A.J., 2009. *Confronting the challenges of participatory culture: Media education for the 21st century*. Mit Press.

Jenkins, H., 2012. Textual poachers: *Television fans and participatory culture*. Routledge.

Kow, Y.M. and Nardi, B., 2010. Who owns the mods?. *First Monday*, 15(5).

Kücklich, J., 2005. Precarious playbour: Modders and the digital games industry. *fibreculture*, 5(1).

Laukkanen, T., 2005. Modding Scenes-Introduction to user-created content in computer gaming.

Lazzarato, M., Virno, P. and Hardt, M., 1996. Radical Thought in Italy: *A Potential Politics*.

McKinney, J.C., 1969. Typification, typologies, and sociological theory. *Social Forces*, 48(1), pp.1-12.

Miettinen, J. and Vehkalahti, K., 2013. Verkkokyselytutkimusten otosten valinta. In *Laaksonen, Salla-Maaria & Matikainen, Janne & Tikka, Minttu (toim.): Otteita verkosta. Verkon ja sosiaalisen median tutkimusmenetelmät. Vastapaino. Tampere*, pp.84-104.

O'reilly, T., 2009. *What is web 2.0.* " O'Reilly Media, Inc.".

Postigo, H., 2003. From Pong to Planet Quake: Post-industrial transitions from leisure to work. *Information Communication & Society*, 6(4), pp.593-607.

Postigo, H., 2007. Of mods and modders: Chasing down the value of fan-based digital game modifications. *Games and Culture*, 2(4), pp.300-313.

Saaranen-Kauppinen, A. and Puusniekka, A., 2006. KvaliMOTV-

Menetelmäopetuksen tietovaranto. Available at: http://www.fsd.uta.fi/menetelmaopetus (Accessed: 10 October 2017)

Scacchi, W., 2010. Computer game mods, modders, modding, and the mod scene. *First Monday*, 15(5).

Sihvonen, T., 2009. Players unleashed! Modding The Sims and the culture of gaming.

Sotamaa, O., 2004. Playing it my way? Mapping the modder agency. In *Internet Research Conference* (Vol. 5, pp. 19-22).

Sotamaa, O., 2010. When the game is not enough: Motivations and practices among computer game modding culture. *Games and Culture*, 5(3), pp.239-255.

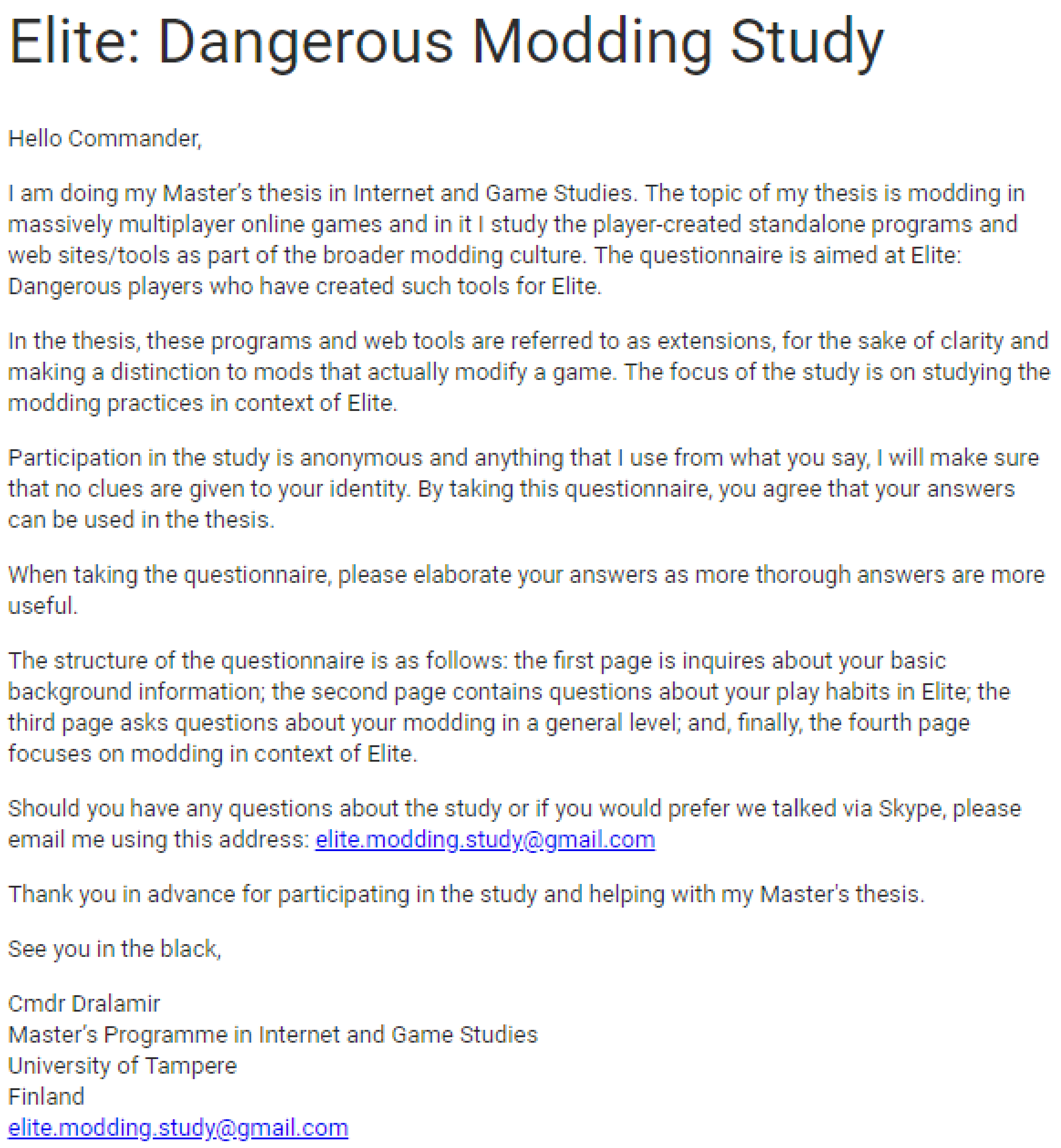
Sotamaa, O. and Wirman, H., 2015. Online games modifications and user generated content. *The International Encyclopedia of Digital Communication and Society*.

Taylor, T.L., 2007. Pushing the borders: Player participation and game culture. *Structures of participation in digital culture*, pp.112-130.

Terranova, T. 2000, "Free labor: Producing culture for the digital economy", *Social text*, vol. 18, no. 2, pp. 33-58.

Wolf, C., Joye, D., Smith, T.W. and Fu, Y.C. eds., 2016. *The SAGE handbook of survey methodology*. Sage.

APPENDIX 1: SURVEY FORM



Elite: Dangerous Modding Study

Hello Commander,

I am doing my Master's thesis in Internet and Game Studies. The topic of my thesis is modding in massively multiplayer online games and in it I study the player-created standalone programs and web sites/tools as part of the broader modding culture. The questionnaire is aimed at Elite: Dangerous players who have created such tools for Elite.

In the thesis, these programs and web tools are referred to as extensions, for the sake of clarity and making a distinction to mods that actually modify a game. The focus of the study is on studying the modding practices in context of Elite.

Participation in the study is anonymous and anything that I use from what you say, I will make sure that no clues are given to your identity. By taking this questionnaire, you agree that your answers can be used in the thesis.

When taking the questionnaire, please elaborate your answers as more thorough answers are more useful.

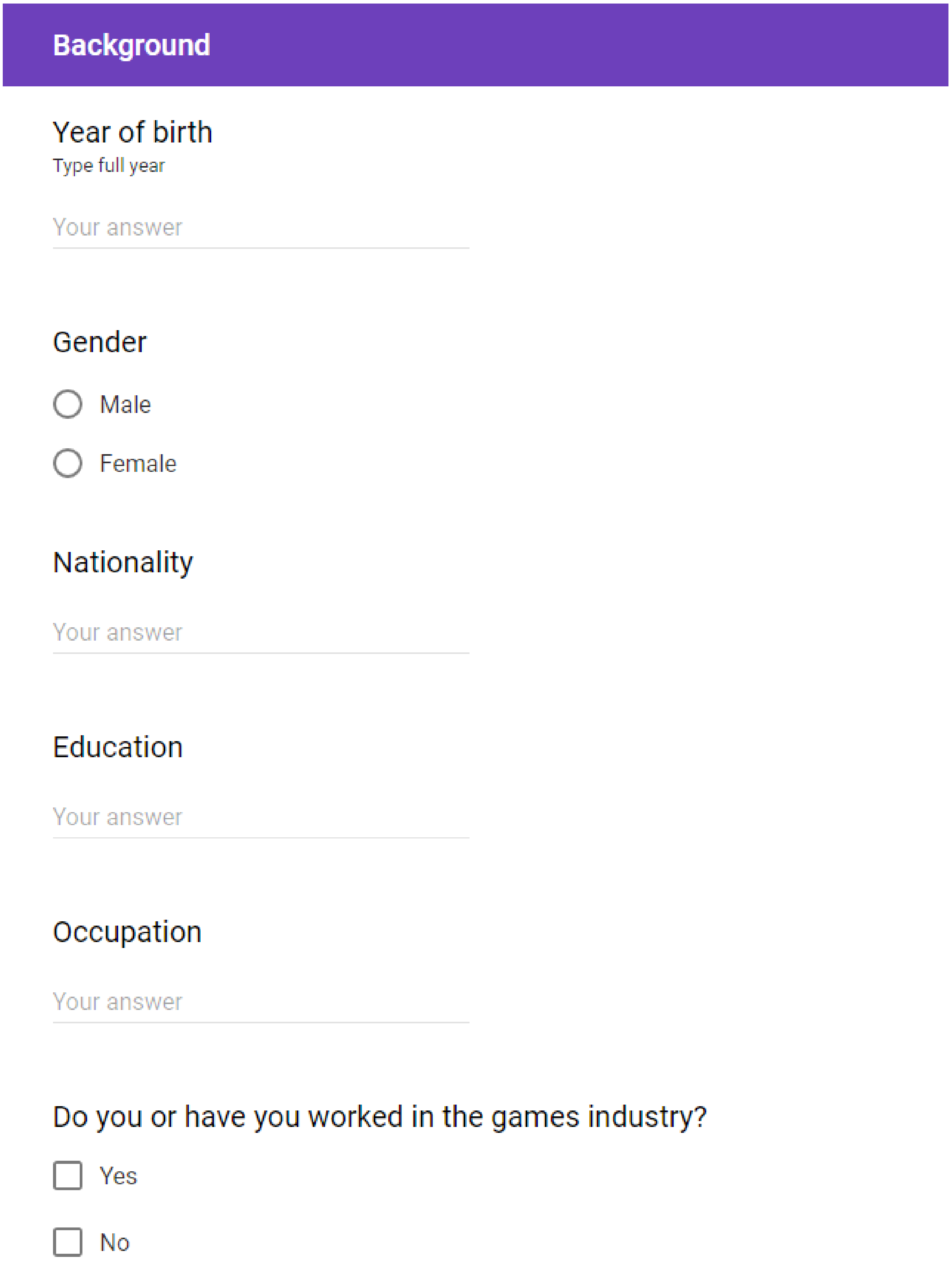
The structure of the questionnaire is as follows: the first page is inquires about your basic background information; the second page contains questions about your play habits in Elite; the third page asks questions about your modding in a general level; and, finally, the fourth page focuses on modding in context of Elite.

Should you have any questions about the study or if you would prefer we talked via Skype, please email me using this address: elite.modding.study@gmail.com

Thank you in advance for participating in the study and helping with my Master's thesis.

See you in the black,

Cmdr Dralamir
Master's Programme in Internet and Game Studies
University of Tampere
Finland
elite.modding.study@gmail.com



Background

Year of birth

Type full year

Your answer

Gender

○ Male

○ Female

Nationality

Your answer

Education

Your answer

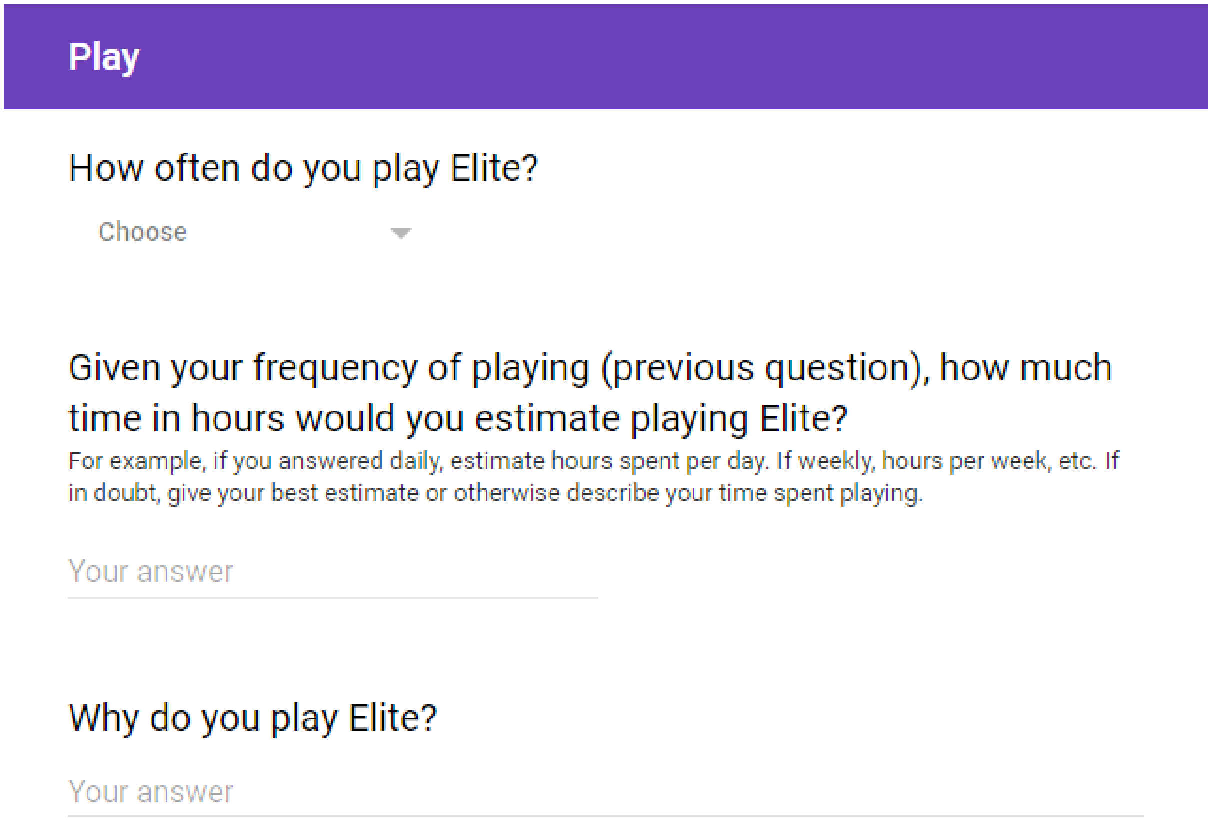
Occupation

Your answer

Do you or have you worked in the games industry?

□ Yes

□ No



Play

How often do you play Elite?

Choose ▾

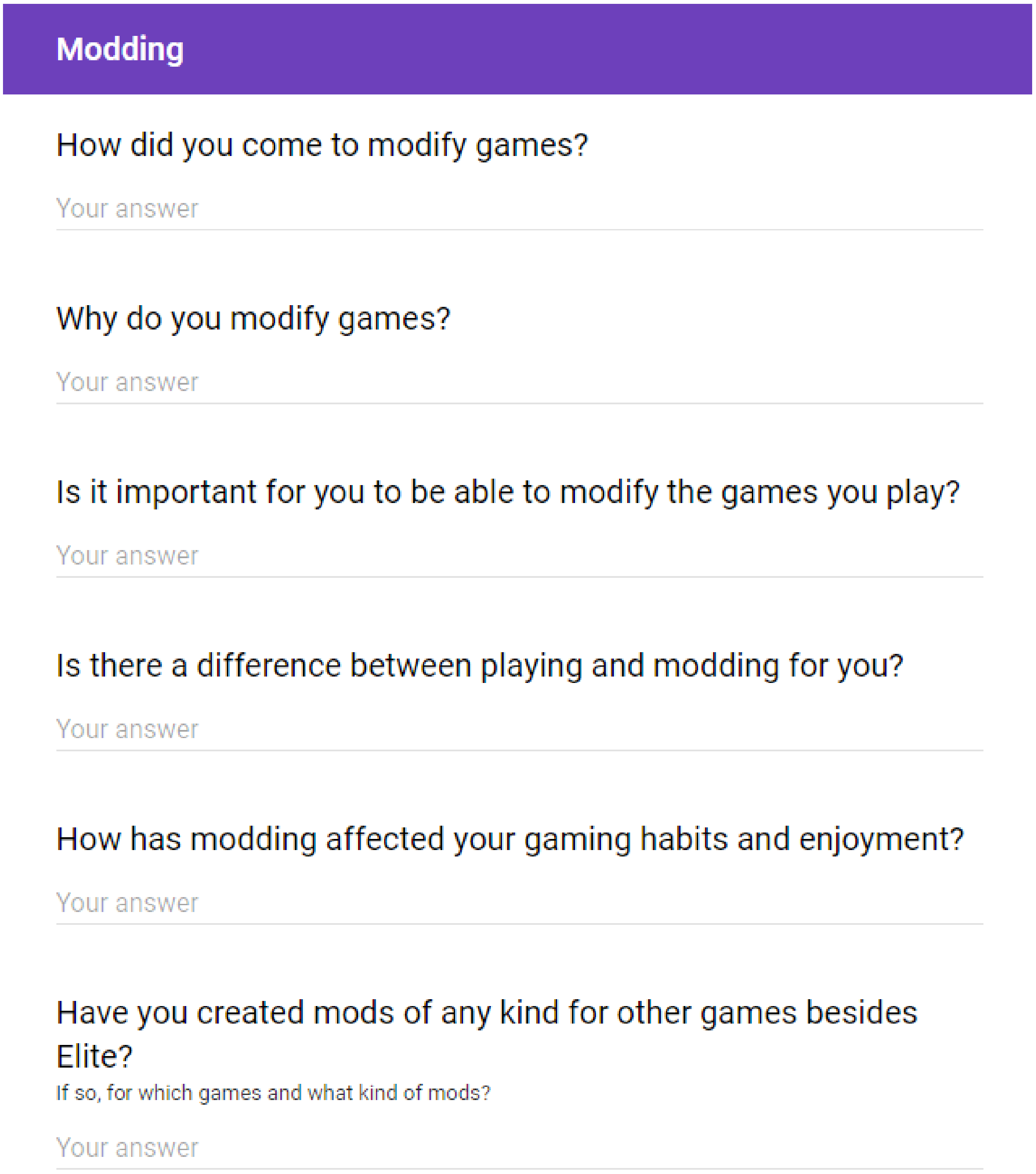
Given your frequency of playing (previous question), how much time in hours would you estimate playing Elite?

For example, if you answered daily, estimate hours spent per day. If weekly, hours per week, etc. If in doubt, give your best estimate or otherwise describe your time spent playing.

Your answer

Why do you play Elite?

Your answer



Modding

How did you come to modify games?

Your answer

Why do you modify games?

Your answer

Is it important for you to be able to modify the games you play?

Your answer

Is there a difference between playing and modding for you?

Your answer

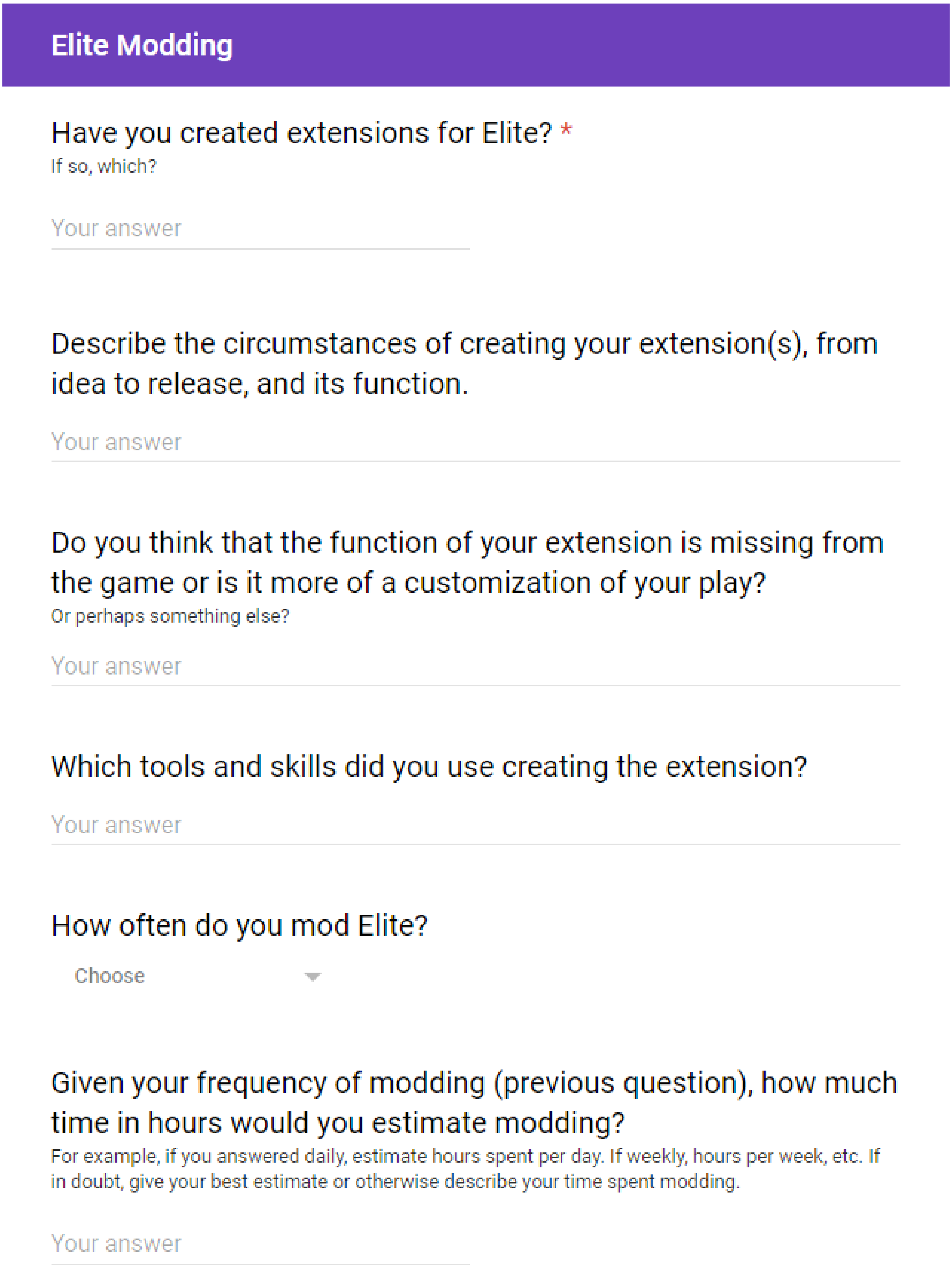
How has modding affected your gaming habits and enjoyment?

Your answer

Have you created mods of any kind for other games besides Elite?

If so, for which games and what kind of mods?

Your answer



Elite Modding

Have you created extensions for Elite? *

If so, which?

Your answer

Describe the circumstances of creating your extension(s), from idea to release, and its function.

Your answer

Do you think that the function of your extension is missing from the game or is it more of a customization of your play?

Or perhaps something else?

Your answer

Which tools and skills did you use creating the extension?

Your answer

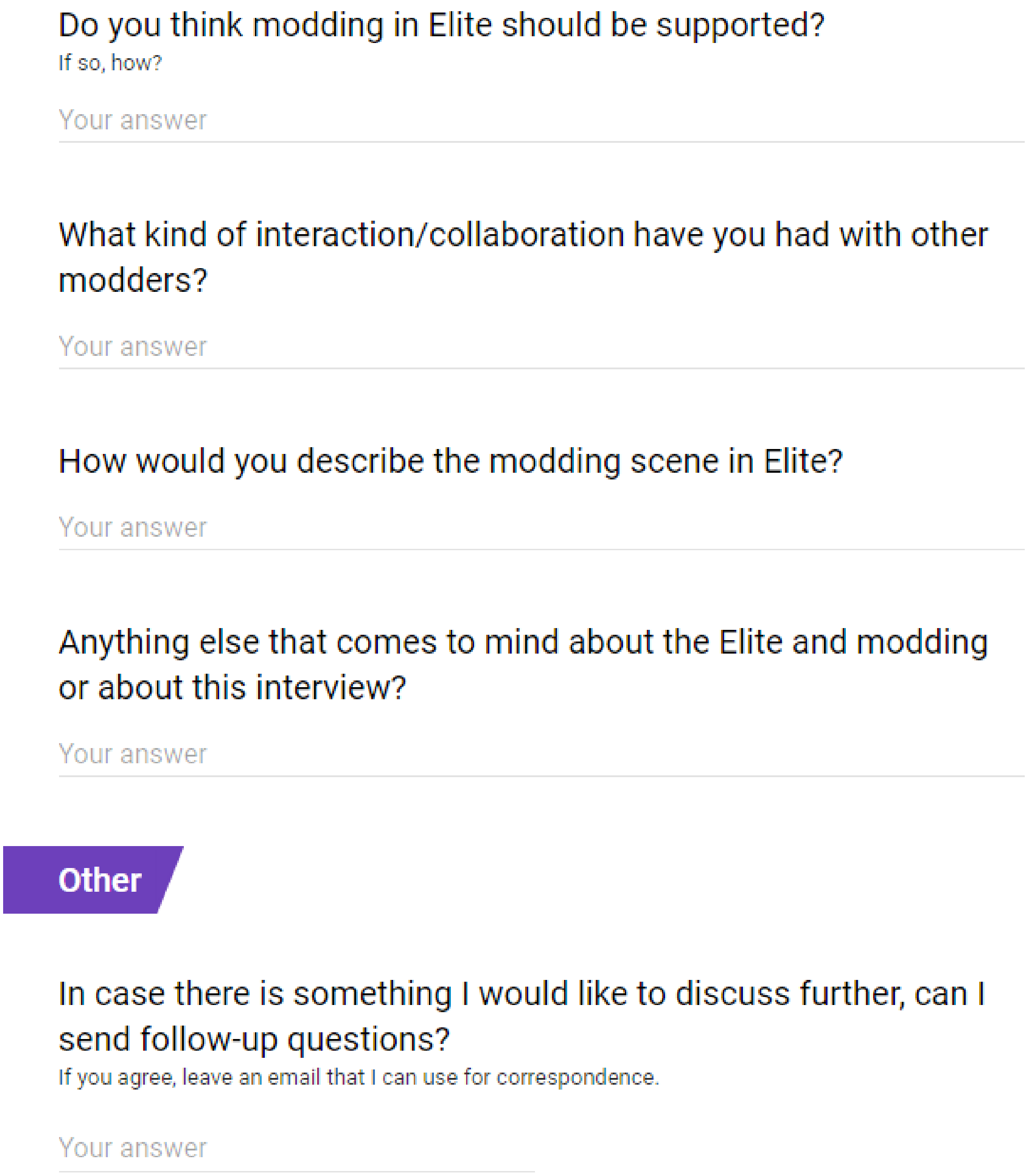
How often do you mod Elite?

Choose ▾

Given your frequency of modding (previous question), how much time in hours would you estimate modding?

For example, if you answered daily, estimate hours spent per day. If weekly, hours per week, etc. If in doubt, give your best estimate or otherwise describe your time spent modding.

Your answer



Do you think modding in Elite should be supported?
If so, how?

Your answer

What kind of interaction/collaboration have you had with other modders?

Your answer

How would you describe the modding scene in Elite?

Your answer

Anything else that comes to mind about the Elite and modding or about this interview?

Your answer

Other

In case there is something I would like to discuss further, can I send follow-up questions?
If you agree, leave an email that I can use for correspondence.

Your answer