Emmi Korhonen

INPUT BINDING IN SIMULATOR GAMES

How to design a keybinding interface for hardcore players

Bachelor's thesis

Bachelor of Culture and Arts

Degree programme in Game Design

2022

South-Eastern Finland
University of Applied Sciences

Degree title	Bachelor of Culture and Arts
Author(s)	Emmi Korhonen
Thesis title	Input Binding in Simulator Games How to design a keybinding interface for hardcore players
Commissioned by	Frozenbyte Oy
Year	2022
Pages	42 pages, 26 pages of appendices
Supervisor(s)	Suvi Pylvänen, Senior Lecturer Anna Narinen, Design Lead

ABSTRACT

Keybinding is a part of the game experience players tend to see as a brief interaction and is often overlooked as insignificant.

The objective of the thesis was to provide improvement suggestions to the commissioner regarding their game’s spaceship keybind menu through findings from formal analyses of two other space simulator games and a focus group discussion.

The author of the thesis worked for the commissioner as a user interface designer and produced an updated spaceship keybind menu to the comissioner’s product. The design process of the interface was described and evaluated.

Qualitative methods used were formal analyses of two space simulator games and a focus group discussion in order to provide improvement suggestions to the commissioner regarding the spaceship keybind menu.

The formal analyses found that there were specific design conventions used in keybinding interfaces of simulator games. The focus group discussion revealed common problems players have with keybinding interface and user experience design.

The study found that more time is needed to conduct a more comprehensive research of other simulator games. More focus group discussions or individual interviews with players with even more experience in simulator games are needed but the study provided a good basis for future research on the subject. Additionally, accessibility and controllers in simulator games would make a great research subject of its own.

Keywords: user interface design, keybinding, simulator game, video game, usability

CONTENTS

1 INTRODUCTION

Not many players think of keybinding as a significant part of the gameplay experience and tend to overlook it. The process of keybinding is typically brief or non-existent, however the process is highlighted in simulator games. For games of this genre, the importance of well-designed keybinding interfaces are tantamount for the user experience.

Massachusetts Institution of Technology (n.d.) defines keybindings as a connection between a physical key on a keyboard and an action. Keybindings do not detect upper- or lower-case letters and modifier buttons such as Control are seen as a separate key press (MIT n.d.). The same logic is used in video games as well.

The aim of the thesis is to provide improvement suggestions to improve the spaceship binding menu of the simulator game Starbase (2021) based on findings from other space simulator games and a focus group discussion. These research methodologies are defined more specifically in Chapter 2 of this thesis. The thesis focuses on keyboard and mouse input binding, other controllers are discussed but they do not hold as much influence on the findings.

The interest to the study's subject stems from the fact that the author worked as a user interface designer at Frozenbyte and designed a new version of the existing spaceship keybind menu interface in the summer of 2021. The author suggested to the company to make a thesis about the subject. Even though the author herself has minimal experience in simulator games in general, she felt that conducting the study would be good practice for working life responsibilities.

The research questions the thesis intends to answer are regarding considerations on when designing a keybinding interface to players with varying levels of experience in simulator games. This includes exploring the practices used in designing such interfaces and how to make them approachable. The thesis examines themes relating to user interface design such as usability, feedback, and control. Themes examined more briefly are accessibility and other game controller types. There are no other risks apart of the commissioner choosing not to use the findings of the thesis. There are no costs on the thesis process.

The thesis is divided into seven chapters. The introduction describes what the thesis contains. The second chapter describes the research setting and methods used in the thesis to reach the goals set by the author. The third chapter provides the theory basis on user interface design. The fourth and fifth chapters focuses on finding answers to the research questions. The sixth chapter presents the work done in the project the thesis aims to improve. The seventh chapter examines the findings and how to apply them to the existing design as well as evaluates the thesis process and its success.

2 RESEARCH SETTING AND METHODS

Starbase (2021) is a hybrid voxel and vertex-based massively multiplayer online (MMO) space simulation game developed and published by Frozenbyte for Microsoft Windows. The game's Steam Early Access (Early Access n.d.) launched in the summer of 2021. In the game's universe everyone controls a robotic endoskeleton character, starting in one of the thirty Origin Stations at the beginning of the game. The emphasis is on space exploration, building own spaceships, player versus player (PvP) combat and player to player trading. (Starbase 2021.) The project was completed during the summer of 2021 after which the author conducted research on user interface design with the help of analyses and a group interview on the spring of 2022.

The main goal of the thesis is to offer the commissioner improvement suggestions on their game's key binding functionality and interface. The purpose is to provide as accurate information as possible through research findings. The thesis is a project-based ensemble consisting of the design process of a user interface and the written report comparing and analysing the design to other game interface design and the qualitative methods aiming to answer the research questions.

The main research question the thesis aims to answer is:

- What needs to be taken into consideration when designing a key binding interface for mid-level to hardcore space simulator players?

The sub-questions derived from the main research question are:

- How are input binding interfaces designed in simulator games?
- How to design the interface to be more easily approachable?

The thesis applies the qualitative research process (Figure 1). Qualitative research includes methods such as interviews, focus group discussions, observation, content analysis, visual methods and biographies in order to understand contextual influences on the issues being researched (Hennink et al. 2020, 10). The methods chosen in the design cycle were benchmarking and analysis of input binding interfaces from two space simulator games and a focus group discussion to map out the feelings, wants and expectations the participants have about keybinding.

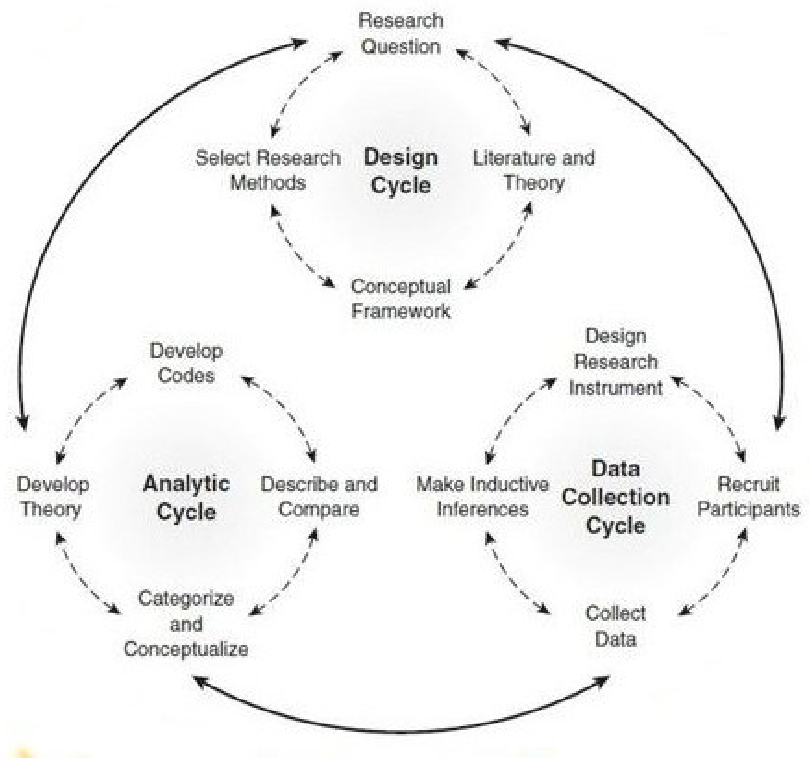


Figure 1. Qualitative research cycle (Hennink et al. 2020, 5)

Benchmarking is a continuous and systematic process where products, services and processes are analysed and compared to learn, understand, and implement the best practices to one's own operation (Tuominen & Niva 2005). The thesis uses this method in the form of a formal analysis (Lankoski & Björk 2015) where the interfaces are broken down into elements and analysed. A focus group discussion is an interactive discussion between pre-selected participants led by a moderator and focuses on specific issues (Hennink et al. 2020, 138). These methods are discussed in more detail in their respective chapters.

The ethical dimensions of the thesis include providing the commissioner with reliable information on improving the input binding interface design. This is done by gathering information on designing an easily accessible and approachable input binding interface from other simulator games and by conducting an interview with individuals who frequently use input binding in their gameplay. Additionally, supporting the ethicality of the thesis is the author's aspiration to improve her skills in game design with the emphasis of user-friendliness. The focus group discussion is recorded and transcribed but the recording is only accessible to the author and the participants if they wish to access it. The identities of participants in the transcript of the focus group discussion will be anonymous.

3 CONTROLS INTERFACE IN SIMULATION GAMES

Saunders and Novak (2013, 143) define simulation games as involving presenting a real-world system in the game. This category of games is diverse and ranges from economic simulations to sports games. Space simulator games can be categorised as vehicle simulations as operating a spaceship is part of the core gameplay. Vehicle simulations aim for realistic controls schemes for the vehicles being simulated and thus have a more special interface needs to them. Even though these games have a lot of keybinds, mastering the realistic interface is appealing for players liking this genre. (Saunders & Novak 2013, 143-144.)

Mapping player actions to computer hardware controls such as console controllers or keyboard commands is called a control scheme (Saunders & Novak 2013, 168). Space and flight simulator games typically have a broad selection of control schemes from keyboard and mouse to console controllers and *Hands on Throttle-And-Stick* (HOTAS) controllers (Figure 2).

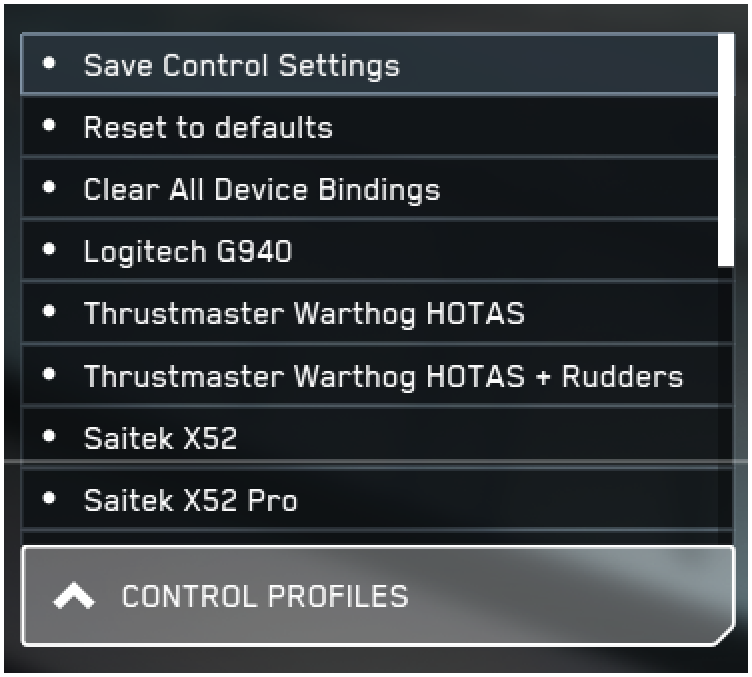


Figure 2. Ready-made control profiles (Star Citizen Alpha 2017)

These profiles have preconfigured binding layouts thus the player has no need to bind actions from scratch to third-party controllers such as HOTAS that have unclear input labels.

Accessibility asserts that designs should be usable by people of diverse abilities, without needing any special adaptations or modifications. The increased knowledge and experience in accessible design has proved that they can benefit every user on any level of capability. (Lidwell et al. 2010, 16.) The importance of accessibility in games is becoming increasingly more essential in game design. According to GDC's report on the state of game industry (2022, 13) 39% of interviewed game developers have implemented accessibility measures to their games.

This progression is opening the world of video games to a larger audience with impairments that would be unable to play games without these measures. Recently, third-party software such as GameGlass (2022) offer touchscreen controls that can be installed on the user's phone or tablet. These types of controls support have the potential to solve accessibility issues involving physical abilities of users and additionally make configuration more linear easier to approach to all users.

There is also legislation overseeing accessibility, the most notable being the Communications and Video Accessibility Act of 2010 or CVAA (Twenty-First Century Communications and Video Accessibility Act 2010). The law affects games sold in the United States of America and requires communication functionality to be as accessible as possible to people with disabilities (Hamilton 2019). For video games, the law requires any communication functionalities and their UIs to be accessible to people with varying levels in senses, motor, and cognitive abilities (McAloon 2019). The law does in most cases impact the rest of the game UI as not having the same accessibility functions in all interfaces affects the overall cohesion and style of the UI.

3.1 Interface elements

According to Churchville (2021) user interface (UI) is the point of human-computer interaction and communication as well as a way for the user to interact with an application or a website. In video game development the term refers to anything helping the player to interact with a game as well as how the game itself communicates back to the player (Saunders & Novak 2013, 3). User experience (UX) is often intertwined with UI design, summarised by Norman and Nielsen (1998) as meeting the exact needs of the end-user without any unnecessary hassle. In video games this means that any aspect of a game's design can have implications to the game's interface (Saunders & Novak 2013, 310).

Video games generally use a subset of the UI known as graphical user interface (GUI). The term refers to an interface with visual indicators or representations the user utilises in communicating with an electronic device. The popularization of GUIs has made computer and other digital technologies more accessible to users lacking technological aptitude or who have other limitations. (Technopedia 2021.)

Game UIs have two primary functions, feedback, and control. Feedback from the UI provides information to the player, often visually through a display such as television screen and through sound effects. Control refers to the game receiving information from the player such as button presses. Primary purposes of feedback are indicating status in the game such as progression toward a goal and teaching new concepts to the player. (Saunders & Novak 2013, 40.)

Usability plays a major role in designing an intuitive and easy to understand interface. Usability as a term refers to user friendliness of an interface by providing information so users know what to do, what options they have and how convenient the interface is for them to use. In video games, considering usability during the design process helps to complement the gameplay and players can make meaningful choices. (Saunders & Novak 2013, 55.)

In Fagerholt and Lorentzon's (2009, 73-75) study of first-person shooters, UI elements are divided into non-diegetic, diegetic, meta-perception, meta-representations, geometric and signifiers (Figure 3).

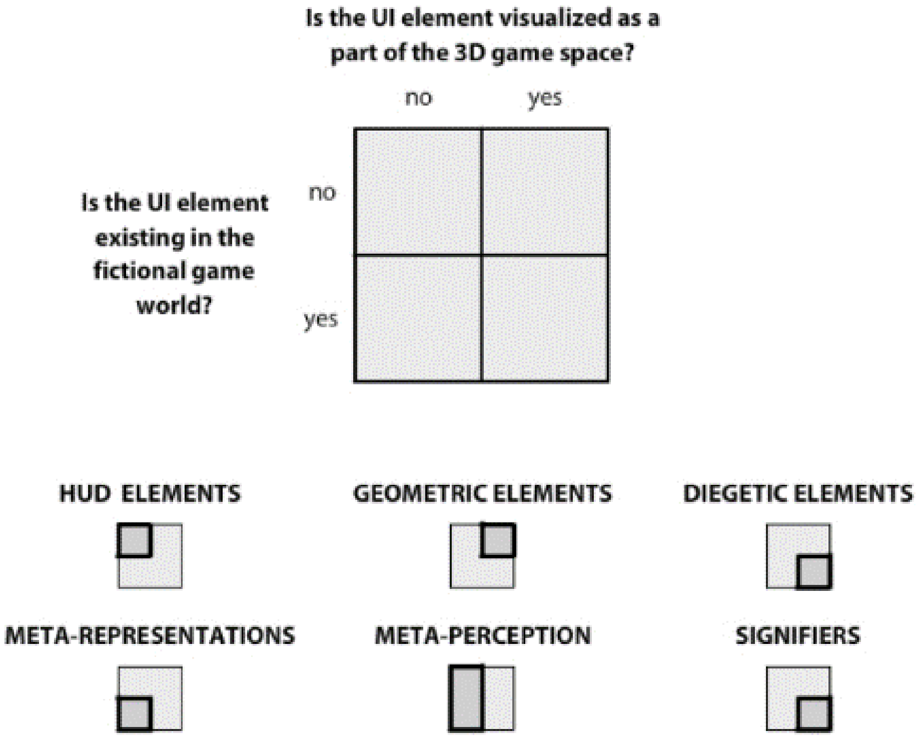

Is the UI element visualized as a part of the 3D game space?
no
yes
Is the UI element existing in the fictional game world?
no
yes
HUD ELEMENTS
GEOMETRIC ELEMENTS
DIEGETIC ELEMENTS
META-REPRESENTATIONS
META-PERCEPTION
SIGNIFIERS

Figure 3. The design space of user interfaces in FPS games (Fagerholt & Lorentzon 2009)

The main function on non-diegetic elements is to convey information on a high level of detail thus they are separate from the game world and only the player can observe them. Diegetic elements exist for both the player and the player avatar. Meta-perception elements can imitate internal perception, connecting to the player's senses and can convey effects such as physical health of the player avatar. Meta-representations exist in the fictional game world but are not in the game space itself. Geometric elements are part of the physical game world but not in the fictional game world and are used to convey spatial information in a game. Signifiers carry information on external entities such as smoke from a fire. (Fagerholt & Lorentzon 2009, 73-75.)

As a controls interface is often non-diegetic, it is separate from the game world itself and thus does not contribute to immersion. Immersion in games refers to players getting involved with the game's experience they forget they are playing a game. Immersive elements in diegetic interfaces appear in the context of the game's fiction and world to provide maximum immersion for the player. (Saunders & Novak 2013, 238.) As controls interfaces typically are there only for the player to remember how to interact with the game, they do not need to be included in the game's world.

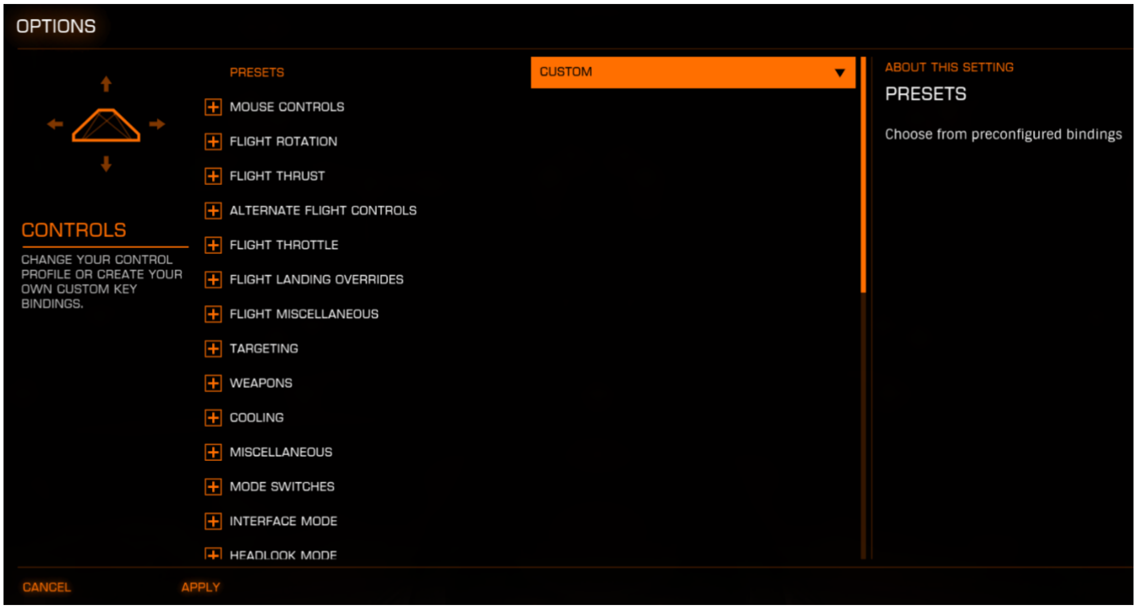
Controls are the core of game interface design as they allow the player to provide input to the game. A well-designed control scheme will reduce frustration and allow concentration on gameplay instead of spending time on the interface. A control scheme is created by mapping player actions to hardware controls such as controllers and keyboard commands. (Saunders & Novak 2013, 167-168.) As the more realistic space simulators might have a massive number of functionalities, a control scheme alone might not be enough for the players to play the game in a way that best suits them.

3.2 Input binding interface

Keybindings are often preferred by advanced players in games as they can greatly improve usability for them. This still requires following the generic and genre-specific conventions if possible while keeping the game's design needs in mind. (Saunders & Novak 2013, 106.) Conventions in UI design refer consistent layouts and patterns that are familiar and easy to remember for most users (Lynch & Horton 2009, 110). However, since the shortcuts are intended for more advanced users, conventions such as easing the learning curve for new players is not a factor. (Saunders & Novak 2013, 106.)

Input binding interfaces in simulator games are often comprised of elements such as confirmation windows, tooltips, right-click menus, and scrollbars. Saunders and Novak (2013, 224-226) describe confirmation windows as preventing the player from performing actions they did not intend to or overwriting a game save, and a tooltip as an interface element displaying information quickly when the cursor is positioned over an item without clicking it. According to Bigelow (2014) right-click menus provide shortcuts for actions anticipated the player might take. A scrollbar is defined by Saunders and Novak (2013, 174) as a control-based solution for limited screen space and are used when an interface contains more information that it can display.

Most space and flight simulators can contain hundreds of different actions, requiring an interface for the player to set the actions they wish to use. The interface is typically non-diegetic with actions categorised to make finding a specific action easier. (Figure 4.)



CHANGE YOUR CONTROL PROFILE OR CREATE YOUR OWN CUSTOM KEY BINDINGS.

PRESETS: CUSTOM

ABOUT THIS SETTING

PRESETS

Choose from preconfigured bindings

Figure 4. Elite: Dangerous' functionalities are under specific categories (Elite: Dangerous 2014)

The most common categories found are those related to general ship control. Multiple inputs of the same controller can be bound to the same action in some games as well. This is especially useful when using a third-party software with a controller together or if the user wants to bind more than one input from the controller they are using to the action.

4 BINDING IN OTHER GAMES

Utilising already established conventions from similar games is a common guideline in interface design. Knowing the game's objectives of interfaces helps determine which guidelines apply and which to ignore or adapt. (Saunders & Novak 2013, 39.) This chapter discusses and analyses the binding systems in other simulation games, where flying on a spaceship in an integral part of gameplay. The games for the analysis are selected using purposive sampling.

The chosen games are Elite: Dangerous (2014) and Star Citizen (2022). Both games are space simulators with an emphasis on operating spaceships. Starbase (2021) can be categorised as quite realistic space simulator with science fiction elements (Figure 5).

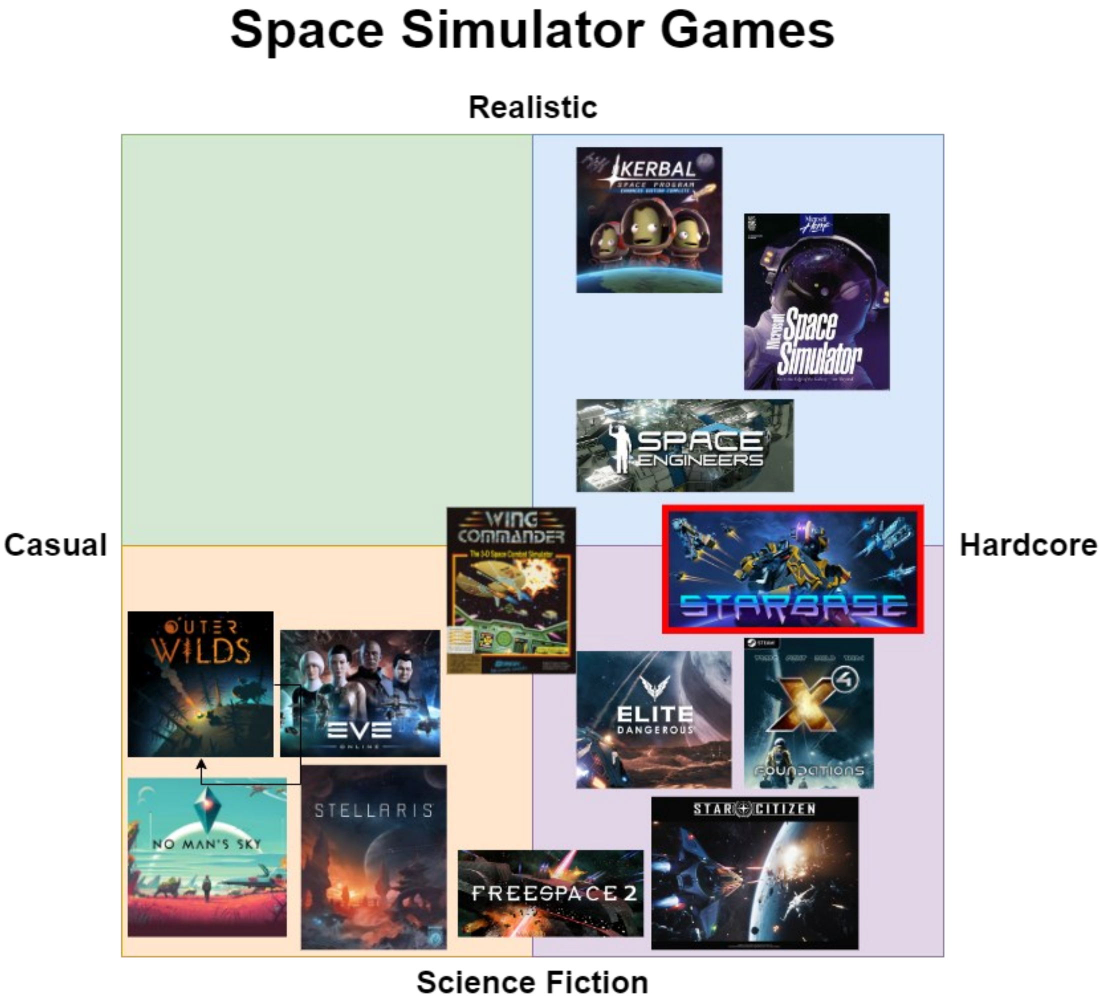


Figure 5. Comparison chart of space simulator games

It is realistic in the sense that the physics have been designed to mimic the real world, however, the player characters are robotic endoskeletons in a distant future somewhere in space and there are no organic lifeforms in the game’s universe at this moment (Starbase 2021).

The analysis is done by utilising the Lankoski and Björk’s (2015) formal analysis of games for binding interfaces. The goal of this formal analysis is to describe the principles of design of the user interfaces examined in detail (Lankoski & Björk 2015, 33). In this analysis, goals have been removed entirely as the functionalities are more important in the context of the thesis. Lankoski and Björk (2015, 23) define formal analysis of games as a study of a game without regarding who specifically are playing a specific instance of the game, focusing only on the formal features of every game.

These analyses study explicitly the key binding interfaces of games, disregarding all other aspects of these games. The author uses a pragmatic approach in this analysis with a customized vocabulary containing the design elements relevant in most formal analyses. The purpose of this vocabulary is to provide coherent yet detailed descriptions on the user interfaces of both games. (Lankoski & Björk 2015, 24-25.)

- *Artifact* refers to the game that is analysed.
- *Primitives* are the basic building blocks of games (Lankoski & Björk 2015, 25), in the context of this analysis the primitives refer to a user interface screen.
- *Components* define the game space (Lankoski & Björk 2015, 25). In this analysis, components are the interactable elements defining the user interface space.
- *Component actions* are the actions coming from the components themselves (Lankoski & Björk 2015, 26). This refers to the actions of the interactable elements.
- *Player actions* are the actions initiated by the player (Lankoski & Björk 2015, 25). This includes player choosing what inputs to bind.
- *System actions* are the actions not originating from either players or components (Lankoski & Björk 2015, 26). In this analysis the term refers to the interface offering additional information to the player.

Similar to Lankoski and Björk (2015, 28), the terms are shown in italics in the analysis of each game for clarity. Each analysis follows the same structure to facilitate the process of comparison between the games analysed. All analyses start with general information on the game and what is covered in the analysis. Then the layout of the UI is described followed by description on actions used by the components, the player and the system. Lastly there will be the analysis section on design and the five major considerations defined by Saunders and Novak (2013, 39) followed by an extension of analysis.

4.1 Elite: Dangerous

Elite: Dangerous (2014) is an open world, space simulation MMO where players pilot a fighter ship in a realistic 1:1 scale of the Milky Way galaxy. As the emphasis is on operating a spaceship, the key binds are an essential part of gameplay. The analysis covers the *Controls* screen accessed through *Options*.

- The UI consists of the following component types: *buttons, collapsibles, dropdowns, sliders, toggles, key binds, analogue inputs locked options, confirmation window* and *scrollbar.*
- The component actions are *key input recording, analogue input recording, dropdown options, collapsible options, toggle options* and *highlight on hover or click.*
- The player actions are *clicking, dragging, binding an input to an action, accepting or cancelling an activity* and *navigating the interface.*
- The system actions are *confirmation, information* and *locking or unlocking components.*

There are 56 *analogue inputs*, 510 *key binds*, 73 *sliders* and 134 *toggle inputs* in total (Appendix 1). All actions are categorised under *collapsibles* (Figure 6).

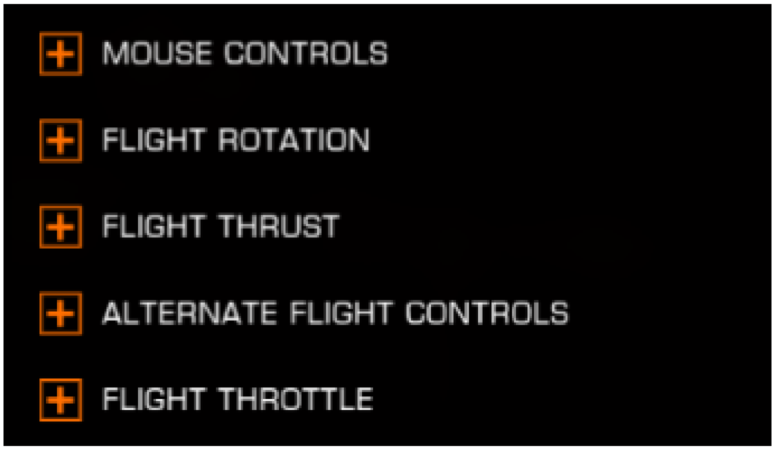
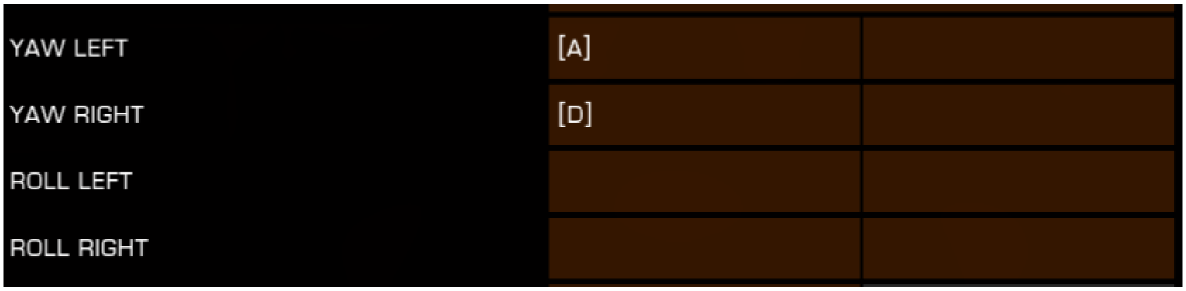


Figure 6. Example of collapsibles (Elite: Dangerous 2014)

Collapsibles are visualised by an orange rectangular box with a plus sign inside. *Clicking* opens the category specific actions and their options. These options are in the forms of *toggles* and *sliders* found under *dropdowns* and additional *collapsibles*. *Dropdowns* are used when there are more than two *toggle options* and *collapsibles* are used for additional action specific configurations.

Key binds are visualised with two boxes next to an action where the player can bind two inputs to the same action (Figure 7). This can be used for controllers such as HOTAS, joysticks or if the player just wishes to have two keyboard buttons performing the same action.

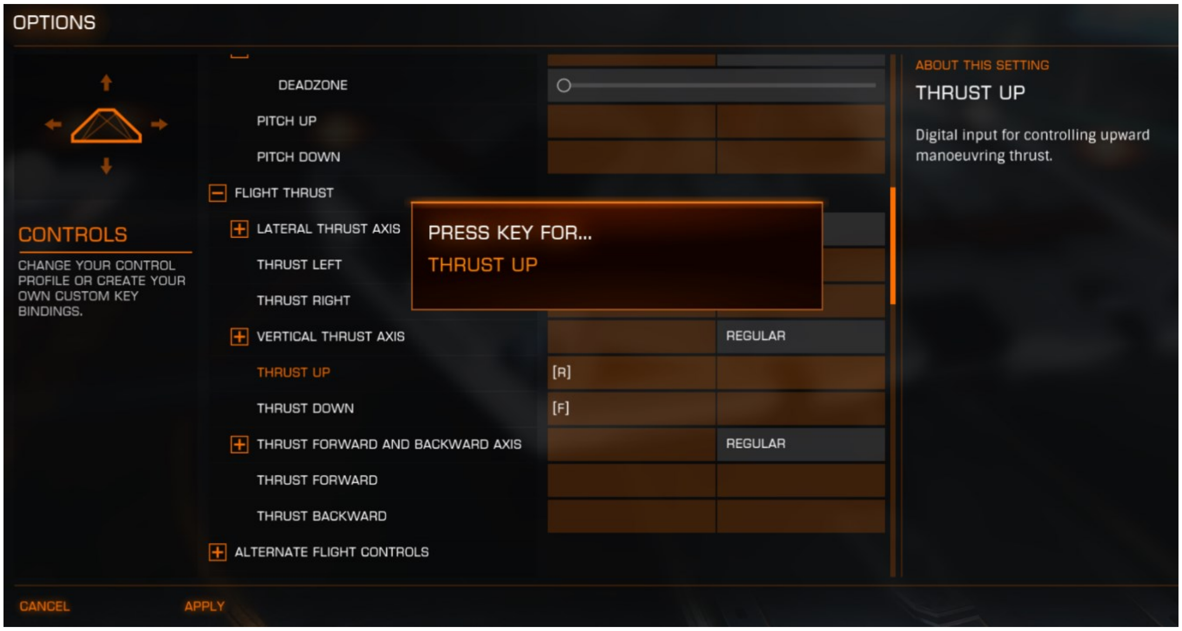


YAW LEFT	[A]	
YAW RIGHT	[D]	
ROLL LEFT		
ROLL RIGHT		

Figure 7. Example of key bind boxes (Elite: Dangerous 2014)

There are two *buttons* on the initial screen layout known as Cancel and Apply, these are used to save or revert the changes made to the keybinds as a whole. Other buttons are found from the *confirmation window*. These are used to confirm or cancel the action being made. *Toggles* provide two or more additional settings for a specific input such as regular or inverted analogue controller axis. *Sliders* are used for either mouse-specific settings or setting a deadzone for an axis. *Locked options* are components tied to the system detecting if an analogue input has been bound to an action. These elements are indicated with a darker grey colour. Clicking or dragging them produces no response.

Component actions are a response to player actions in the UI. To bind a button input, the player *clicks* on a *key bind box*, opening a *key input recording* state (Figure 8).



OPTIONS

CONTROLS

CHANGE YOUR CONTROL PROFILE OR CREATE YOUR OWN CUSTOM KEY BINDINGS.

DEADZONE
PITCH UP
PITCH DOWN
⊟ FLIGHT THRUST
⊞ LATERAL THRUST AXIS
THRUST LEFT
THRUST RIGHT
⊞ VERTICAL THRUST AXIS — REGULAR
THRUST UP — [R]
THRUST DOWN — [F]
⊞ THRUST FORWARD AND BACKWARD AXIS — REGULAR
THRUST FORWARD
THRUST BACKWARD
⊞ ALTERNATE FLIGHT CONTROLS

PRESS KEY FOR...
THRUST UP

ABOUT THIS SETTING
THRUST UP

Digital input for controlling upward manoeuvring thrust.

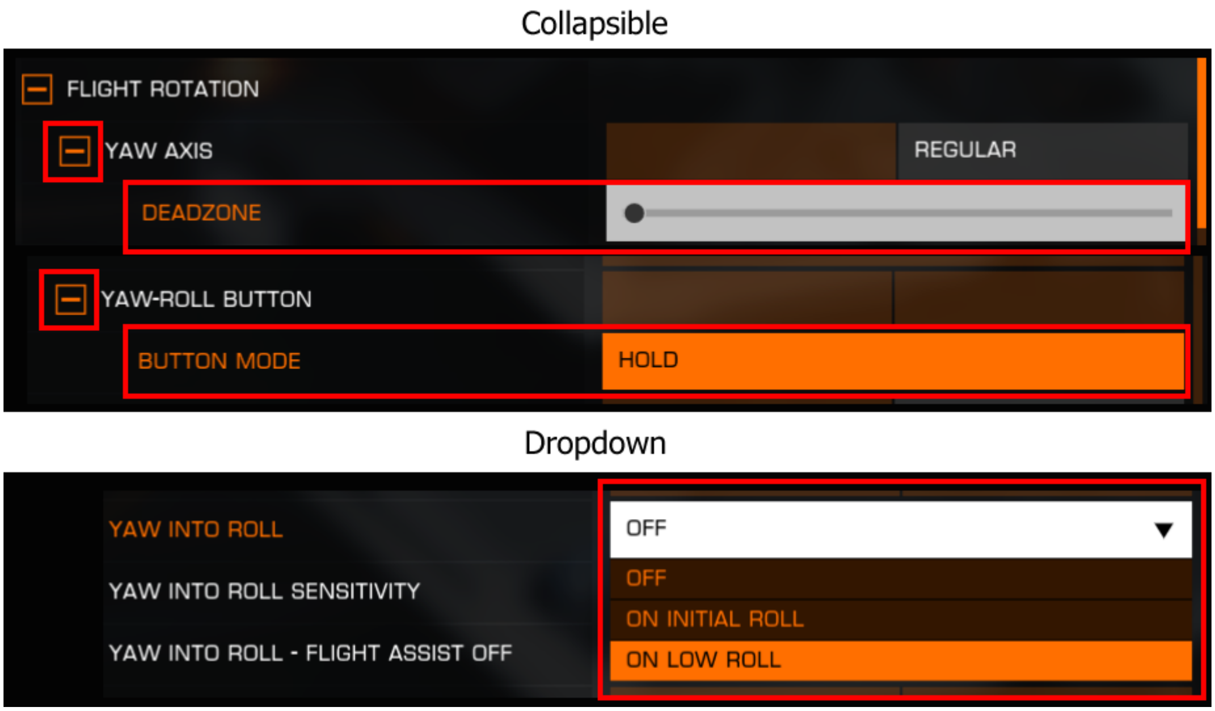
CANCEL APPLY

Figure 8. Key input recording state (Elite: Dangerous 2014)

The player does the same process to access the *analogue input recording* state for controller binding. The *recording state* window stays open until an input is assigned or the process is cancelled by pressing Escape.

The *scrollbar* can be used by the player to *navigate* the categories. However, using the *scrollbar* is not always necessary as the player can *scroll down* with either a mouse wheel or a controller. The *scrollbar* indicates that there is additional content in the interface.

Collapsible options and *dropdown options* are activated when the player *clicks* on either component (Figure 9).



Collapsible

FLIGHT ROTATION

YAW AXIS REGULAR

DEADZONE

YAW-ROLL BUTTON

BUTTON MODE HOLD

Dropdown

YAW INTO ROLL OFF ▼

YAW INTO ROLL SENSITIVITY

YAW INTO ROLL - FLIGHT ASSIST OFF

OFF

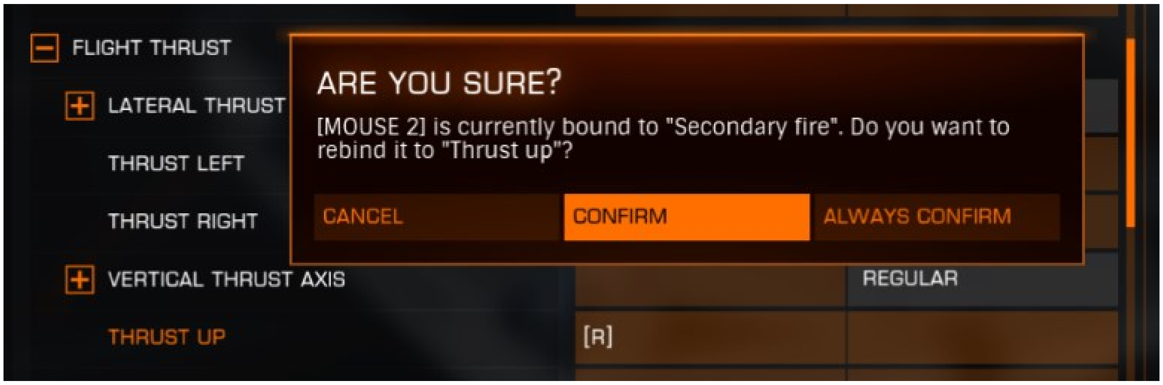
ON INITIAL ROLL

ON LOW ROLL

Figure 9. Additional settings are indicated by either a collapsible box or a dropdown arrow (Elite: Dangerous 2014)

These offer additional settings to a specific action which are configured either via a *slider* or a *toggle. Hovering* over an option highlights the area with orange and a selected area is highlighted with white.

Conflicts detected when binding an input is shown in the form of *confirmation.* When a key is already being used in some other action, the system informs the player about this with a *confirmation window* (Figure 10).



FLIGHT THRUST

LATERAL THRUST

THRUST LEFT

THRUST RIGHT

VERTICAL THRUST AXIS REGULAR

THRUST UP [R]

ARE YOU SURE?

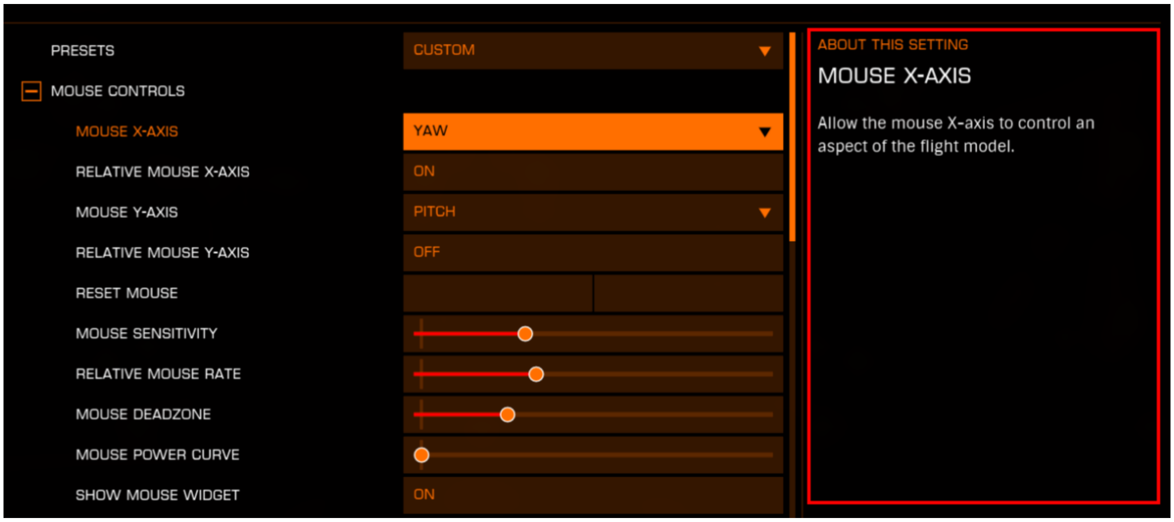
[MOUSE 2] is currently bound to "Secondary fire". Do you want to rebind it to "Thrust up"?

CANCEL CONFIRM ALWAYS CONFIRM

Figure 10. Confirmation window for binding a new input (Elite: Dangerous 2014)

Confirmation windows help prevent mistakes and even if a mistake is made, it can be cancelled by the “Cancel” button at the bottom left of the screen. However, this can create frustration as the binding process needs to be restarted every time this is done. If the constant confirming starts to become tiresome, the confirmation window does offer the option to always confirm the binding changes. This can lead to mistakes but are easily reversible by either cancelling the process by exiting and re-entering the menu. Even if the changes are applied they can be reversed by selecting a profile from the preset *dropdown* at the top of the bindings list.

When the player hovers over an action, explanation of the action’s functionality appears to the left side of the screen (Figure 11). This system action provides the player *information* about what the action is used for.



PRESETS	CUSTOM
MOUSE CONTROLS	
MOUSE X-AXIS	YAW
RELATIVE MOUSE X-AXIS	ON
MOUSE Y-AXIS	PITCH
RELATIVE MOUSE Y-AXIS	OFF
RESET MOUSE	
MOUSE SENSITIVITY	
RELATIVE MOUSE RATE	
MOUSE DEADZONE	
MOUSE POWER CURVE	
SHOW MOUSE WIDGET	ON

ABOUT THIS SETTING

MOUSE X-AXIS

Allow the mouse X-axis to control an aspect of the flight model.

Figure 11. Information on an action’s functionality highlighted with red (Elite: Dangerous 2014)

If there is an analogue input bound to an action, the system *unlocks* the options of the action that can be interacted with. This is indicated by changing the *locked option* colours from grey to dark orange as with all other active components in the UI. *Locking* a component happens when an action does not have an analogue input bound to it.

The genre-specific conventions used in Elite’s keybinding interface are the location of the keybinds, all binds are under specific categories, more than one input can be bound to an action, additional information on an action’s functionalities are provided and ready-made profiles for multiple controllers. The lastly mentioned profiles have become common in modern day simulators as there is a vast amount of controllers the players use on the market today (Verkkokauppa.com 2022).

Flight operation related binds are at the top of the list after mouse controls and easy to find (Appendix 1). Other less important binds for core gameplay comes after. Collapsibles help to keep the binds organised under their respective categories by hiding unneeded information in the layout, thus reducing interface clutter (Saunders & Novak 2013, 228).

Additional information on each action clarifies its functionalities. The interface’s colour scheme is simple, using only orange, white, red and a few shades of grey indicate the state of the elements such as if they’re interactable or activated. The text of the menu is clear, all actions are written in upper case letters and additional information written normally.

Investigating other controller layouts and how their binding processes look like is an easy approach to continuing analysing the interface in the future. The interface could also be examined purely from the accessibility point of view.

4.2 Star Citizen

Star Citizen (2017) is an online space simulation game currently in the Alpha stage of development, current version of the game at the time of the thesis being 3.17. The term Alpha in video game production refers to game testing by players is done in parallel with the development of the game. Alpha testing requires the players to document any failures happening in the game to the developers for them fix the issues. (Ixie Gaming 2021.) The publishing date used in the thesis is when the third Alpha version was published as every major update changes the game greatly compared to what it previously was (Roberts Space Industries 2017).

Although there are other features in the game being on foot and ground vehicle gameplay, the spaceships remain as the priority feature in the game. As the ships are designed to be as realistic as possible, the key bindings are a very important aspect of properly running a ship's systems. (Star Citizen Alpha 2017.) Two different interface primitives are examined: *Controls* and *Keybindings* that are accessed through *Options*.

- The UI components are *buttons, collapsibles, dropdowns, sliders, toggles, keybinds, analogue input, confirmation window* and *curves.*
- The component actions are *key input recoding, analogue input recoding, binding timer, collapsible options, dropdown options, toggle options* and *confirmation windows*.
- The player actions are *clicking a component, dragging or clicking a slider, dragging a curve point, navigating the view, binding an input to an action* and *accepting or canceling an activity.*
- The system actions are *confirmation, disabling or enabling components, prompts* and *information.*

Mouse inversion settings and *sensitivity curves* in the *Controls* menu and the input bindings in the *Keybindings* menu are categorised under *collapsibles* (Figure 12).

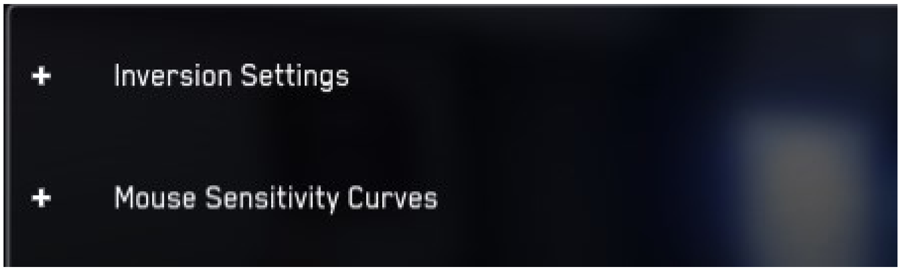


Figure 12. Collapsibles (Star Citizen Alpha 2017)

These are indicated with a plus sign before the category name. *Clicking* a *collapsible* opens the *collapsible options* being a list of actions and their settings.

Controls have 42 *sliders* and 37 *toggles*, *Keybindings* have 26 *analogue inputs* and 462 *keybinds* as seen in Appendices 2/1 and 2/10. The general screen layout in both primitives include *buttons* and a *toggle.* The *buttons* on top of the screen have other *Options* categories. *Clicking* the Back *button* exits the *Options* menu. The *toggle* at the bottom right of the screen changes the controller type. These controller types are keyboard and mouse, gamepad and joystick or HOTAS. Changing the type changes what settings there are available according to the controller selected. The settings components in both primitives are inside an outlined area at the centre of the screen.

Located at the bottom centre of the screen there are *prompt* elements guiding how to operate the menus (Figure 13).

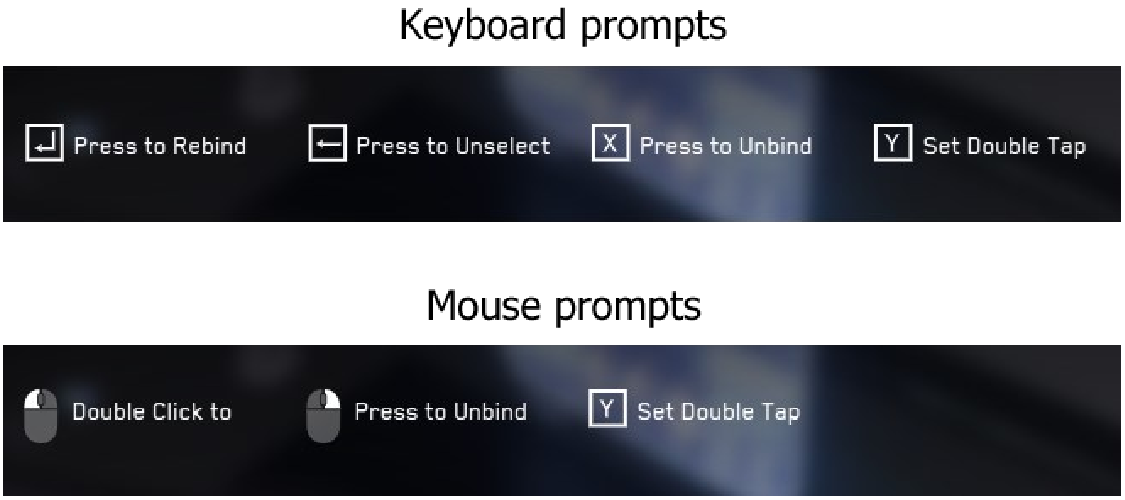
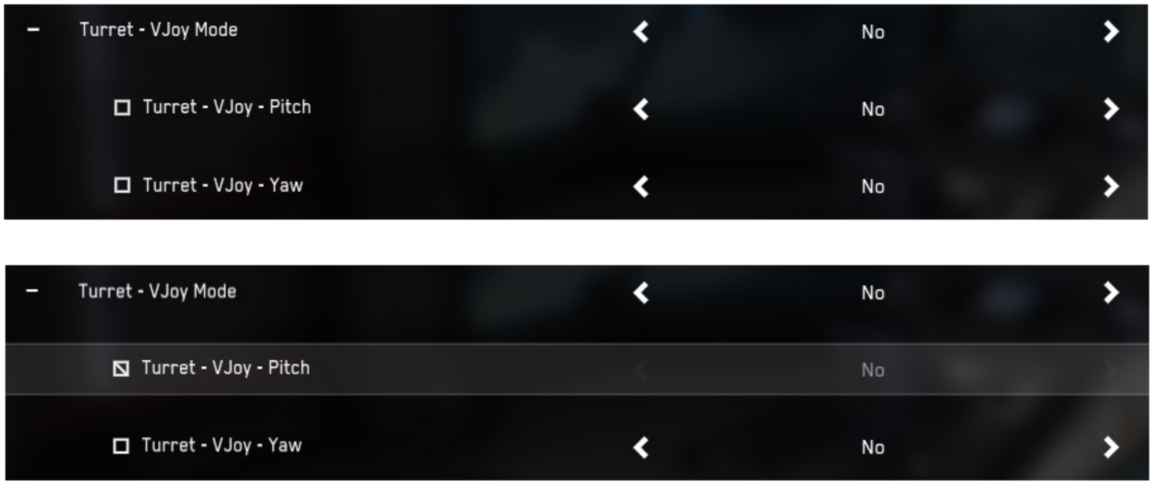


Figure 13. Keyboard and mouse prompts (Star Citizen Alpha 2017)

These *prompts* are found in both *Controls* and *Keybindings.* The information these *prompts* provide changes depending on player actions such as interacting with a *collapsible*. If the player presses Enter, the *prompts* are illustrated as keyboard buttons and to a mouse when clicking a mouse.

The *Controls* UI is used for controller configuration. With a keyboard and mouse setup, this means settings for the mouse. Mouse inversion and *sensitivity curves* have been categorised under *collapsibles.* Other settings are listed after these two categories. *Controls* have two types of *toggles*: the *yes or no toggle* where either option is selected by clicking on an arrow *button* and the *disable-toggle* for enabling or disabling an action (Figure 14).



– Turret - VJoy Mode < No >

□ Turret - VJoy - Pitch < No >

□ Turret - VJoy - Yaw < No >

– Turret - VJoy Mode < No >

☒ Turret - VJoy - Pitch < No >

□ Turret - VJoy - Yaw < No >

Figure 14. Disable-toggle (Star Citizen Alpha 2017)

The *disable-toggle* is a rectangular box before the feature name. *Clicking* it highlights the whole column and a line appears inside the rectangle. This *disables* the action, making the *yes or no toggle* unusable unless the action is *enabled* again.

Mouse sensitivity is adjusted with *sliders* and *curve editors. Sliders* appear next to the action name and next to them are the *curve's* numerical value and an "Edit Curve" *button.* Clicking on the *button* opens a small size, opaque *curve editor* menu. This menu is comprised of a grid with a diagonal line representing the *curve*, the *curve's* value, a *slider* and cancelling and saving *buttons*. The original position of the *curve* is indicated with a dashed line and areas the point of the *curve* being adjusted cannot be moved to are coloured in dark red. (Figure 15.)

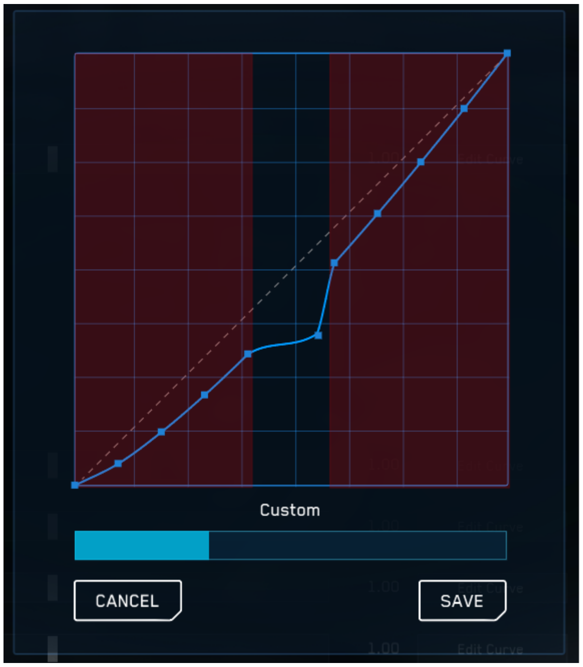


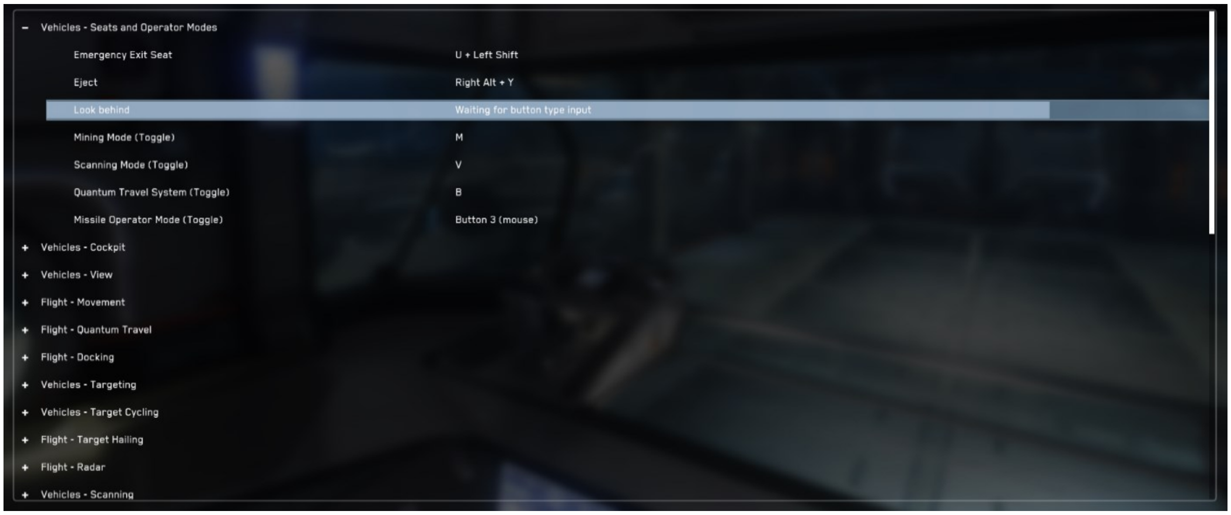
Figure 15. Clicking the curve line creates points of further adjustment (Star Citizen Alpha 2017)

The *curve* is edited by *clicking or dragging the slider* or by *clicking* on the diagonal line in the grid, creating points on the line from which the *curve* can be fine-tuned further. A *confirmation window* opens when the player *clicks* the *curve* or the *slider*, the system *confirms* from the player if they wish to continue modifying the *curve*.

Keybindings UI is used to check the default controller layouts and for input binding. The view opens to a control layout image of a keyboard and mouse, mapping out all default keybindings, pressing the controller *toggle* provides a layout for a console controller. For joystick and HOTAS layouts there is a text asking to refer to advanced keybinding menu for their controls (Star Citizen Alpha 2017).

The UI has two *dropdown* menus. The *dropdown options* containing the Flight and on foot modes are found next to the controller *toggle*. Choosing the mode shows the binding layout with actions related to the mode. The second *dropdown* is found in the advanced controls customization view on the same location on the screen as flight and on foot modes. This menu's *dropdown options* show all the ready-made control profiles the game supports, and the player can save any changes done or reset back to default binds.

Clicking the advanced controls customization *button* contains the customisable keybinds. All keybinds are categorised under *collapsibles.* There is only one binding input for each action. The player assigns an input by *double-clicking* anywhere of an action's column, this *highlights* the column and a *binding timer* is shown visually with a text notification (Figure 16).



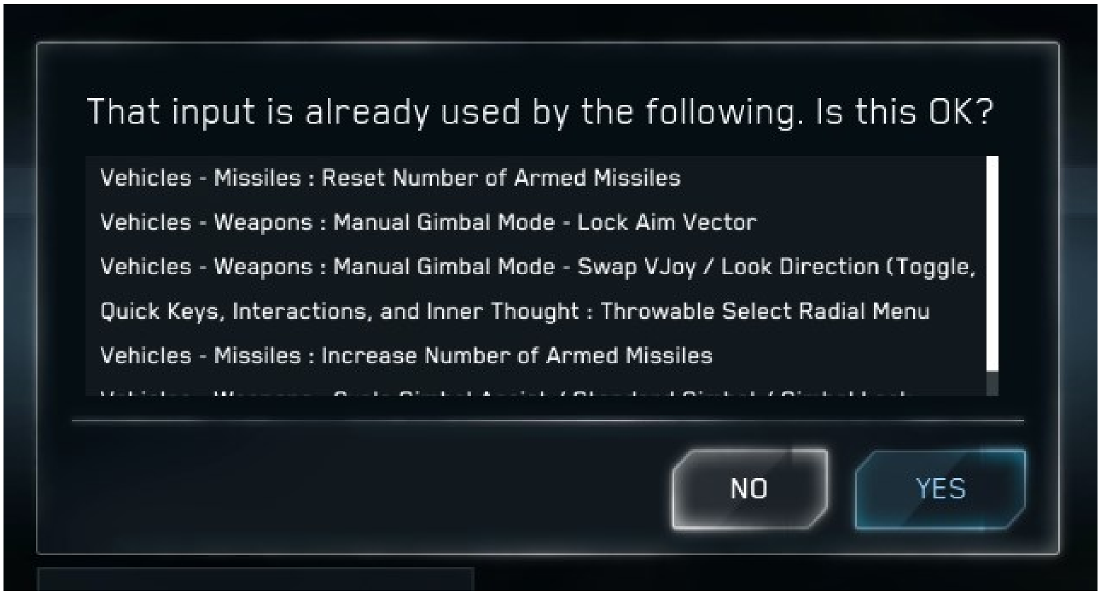
− Vehicles - Seats and Operator Modes	
Emergency Exit Seat	U + Left Shift
Eject	Right Alt + Y
Look behind	Waiting for button type input
Mining Mode (Toggle)	M
Scanning Mode (Toggle)	V
Quantum Travel System (Toggle)	B
Missile Operator Mode (Toggle)	Button 3 (mouse)
+ Vehicles - Cockpit	
+ Vehicles - View	
+ Flight - Movement	
+ Flight - Quantum Travel	
+ Flight - Docking	
+ Vehicles - Targeting	
+ Vehicles - Target Cycling	
+ Flight - Target Hailing	
+ Flight - Radar	
+ Vehicles - Scanning	

Figure 16. Binding state timer (Star Citizen Alpha 2017)

A light blue line on the column shortens with time when in the input binding state. Pressing a key or moving the mouse binds an input to the action.

In addition to an input bind, a double tap function can be added to an action. This function is shown to the player by the system as a *prompt* (Figure 12). The player *clicks* on the action they want to bind and presses the Y key on the keyboard to add the double tapping function. The action can already have an input bound to it or the input can be bound after setting the function. The action can then be performed only by double tapping the input. This can prevent the player from making actions they might not want to do as it requires more than one pressing of an input.

Confirmation is needed from the player when the system detects that the input being bound is already used by another action (Figure 17).



That input is already used by the following. Is this OK?

Vehicles - Missiles : Reset Number of Armed Missiles
Vehicles - Weapons : Manual Gimbal Mode - Lock Aim Vector
Vehicles - Weapons : Manual Gimbal Mode - Swap VJoy / Look Direction (Toggle,
Quick Keys, Interactions, and Inner Thought : Throwable Select Radial Menu
Vehicles - Missiles : Increase Number of Armed Missiles

Figure 17. Input binding confirmation window (Star Citizen Alpha 2017)

The *confirmation window* visualises this by listing all actions using the bind. The player can either cancel or accept the bind assignment.

The conventions used in both primitives are the categorisation of actions and the ready-made profiles for multiple controllers that are presented in a clear manner. Input recording and its timer is shown visually to the player, illustrating how much time there is to choose an input to bind. The other *Options* categories are easily accessed at any point in the UI. Having the preset controls images in *Keybindings* offers a substantial amount of help to the player for quickly checking the default bindings for a keyboard and mouse. However, as there are hundreds of *keybinds,* mapping them to a limited space is difficult and even with the default binds the user can quickly get overwhelmed and confused. These preset images additionally do not change their content information when binding or unbinding an action from a key, making the image impractical for the player if all keybindings have been changed from the defaults.

Prompts are mostly clear with the exception of double clicking the mouse to initiate input binding, the text cuts off which makes it unclear for the player

(Figure 13). The category names do implicate the types of actions listed under them. However, they don’t clearly explain any functionalities and require the player to test them through trial and error thus costing the player time they could spend playing the game itself. The colour scheme is fairly dark but all text use white to pop out. However, the text is fairly small and the outlined area for settings components discussed previously in this chapter, has a lot of space which could be utilised.

The analysis can be furthered by examining the joystick and HOTAS layouts and bindings. The interface can also be analysed from the accessibility point of view and test what types of controllers not included in the controller profile list the game supports and how well does the support work.

4.3 Comparison analysis

The formal analysis showed that in both analysed games keybinds were the most prominent customisation option being over 60% of all other UI elements (Figure 18). As total the numbers were 462 and 510 keybinds that highlights the complexity of keybinding design in simulator games.

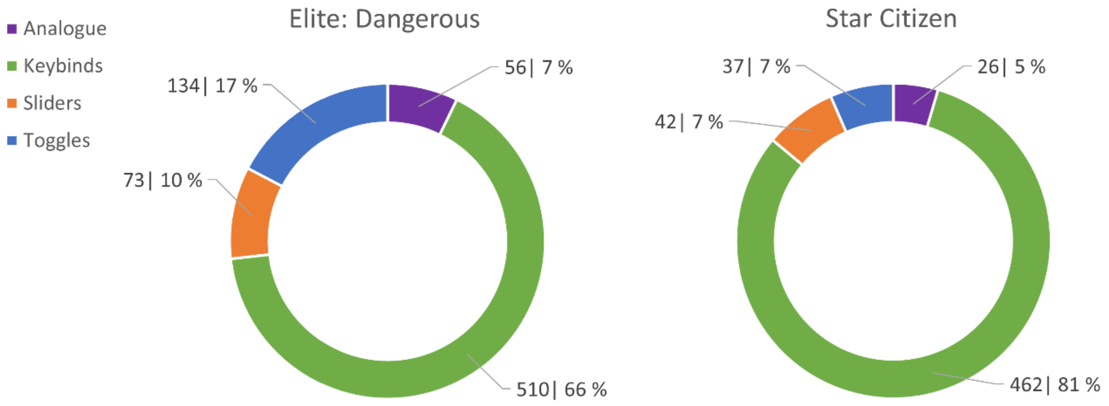


Figure 18. Portion of each input in both UIs

Since the thesis focuses on keybindings, the amount of analogue inputs in the UI is fairly small in both games. Elite combines both mouse settings with input binding whereas Star Citizen has separated sliders and toggles to Controls screen and analogue and keybinds to Keybindings screen. Elite does have fewer functions but more keybinds than Star Citizen because two keybinds can be assigned to one action.

Star Citizen’s preset images are useful for both new and experienced players alike; new players can use it to familiarise themselves to the game’s controls and experienced players are able to quickly check bindings not used often in gameplay. Both games have ready-made input bindings for most common controllers such as gamepad, joystick, and HOTAS. This saves time and the player has no need to bind all inputs to the controller themselves and they serve as a base for building their own custom setup.

Both games list keybinds under categorised lists indicated with a collapsible element. Elite offers clear information regarding action functionalities; Star Citizen unfortunately lacks on this front especially with some category names such as E.V.A (Appendix 2/6). It is important to note that Star Citizen is still currently in the Alpha phase of development and the UI will most likely change with time.

In Elite, the analogue deadzones are set with sliders and are easily found in the menu; in Star Citizen the analogue deadzones are set with curves in the Controls menu. Finding these curves if not knowing what to look for can be difficult. Input binding state in Elite has no set time limit, closing only when an input is assigned or the process cancelled. In contrast, Star Citizen has a set time for input binding, not assigning anything if the timer runs out.

5 FOCUS GROUP DISCUSSION

Saunders and Novak (2013, 102) emphasize how important it is to know the game’s audience and provide an interface best suited for its needs. This chapter covers the focus group discussion where a small group of participants invited by the author discuss their views and experiences with keybindings in video games. Focus group discussions are an interactive discussion between pre-selected participants focusing on specific issues. Generally, the aim of these discussions is to gain a vast range of views on the research topic in an environment where the participants feel comfortable to express them. Focus groups can help the researcher identify a range of issues and the data generated from a group interaction is very different to that of an in-depth interview with a single participant. (Hennink et al. 2020, 138.)

The interview was conducted remotely via a Discord call and lasted for an hour. The themes discussed with the participants were about their experiences with keybinding in video games, what grievances they had about keybinding interfaces and user experience and general improvement ideas regarding usability in such interfaces.

5.1 Focus group discussion design

The author designed a discussion guide to help conducting the group discussion. Hennink et al. (2020, 143) define the guide as a list of topics and series of actual questions acting as a checklist to ensure that all key topics and issues are covered in the discussion. A well-structured guide will help introduce the topic, open the discussion, develop rapport in the group and help focus on key topics, and bring the group discussion to a close.

The discussion guide designed for the thesis follows a funnel structure (Figure 19) with broad opening questions, moving to specific questions and finishing the discussion with closing questions. (Hennink et al. 2020, 143.)

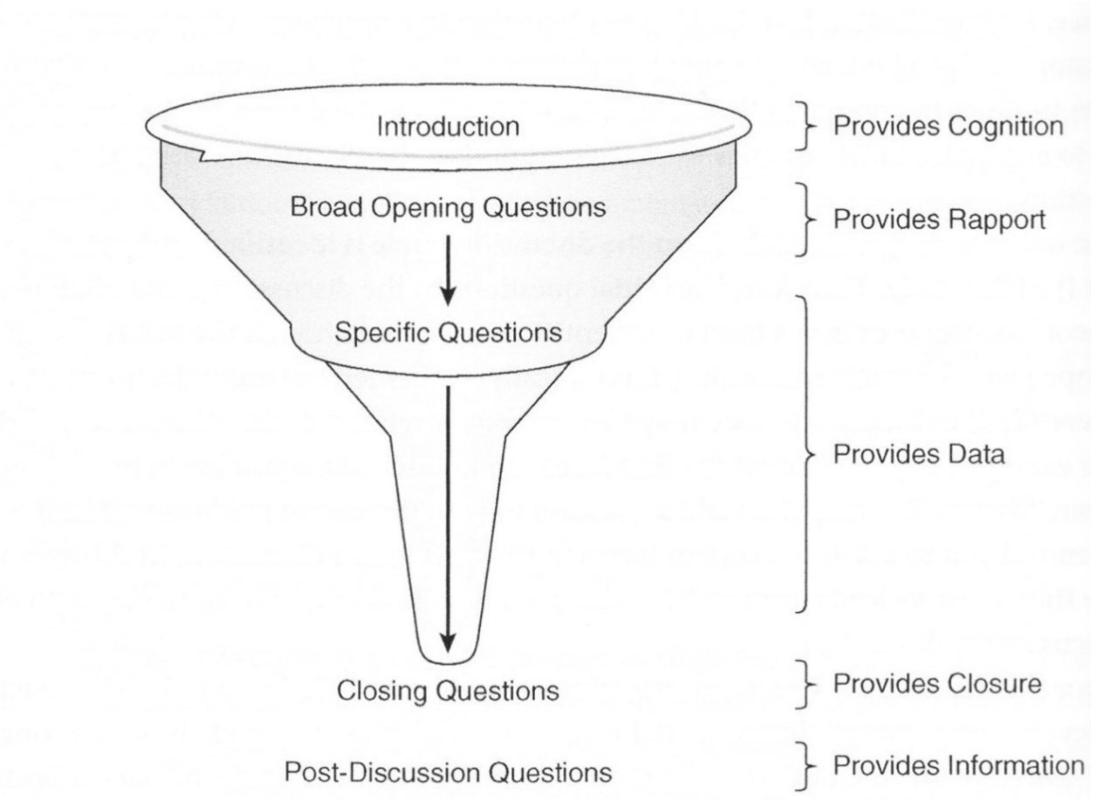


Figure 19. Funnel design of the discussion guide (Hennink et al. 2020, 144)

The questions for the discussion were designed to be short, clear and simple and to be asked in a conversational style. The three aspects of question design were followed to promote discussion, avoiding direct personal questions and having fewer questions compared to an in-depth interview. (Hennink et al. 2020, 147.)

Typically, this type of data collection process is cyclical, meaning that there would be more than one focus group discussion as the first initial discussion identifies issues that can be asked about in subsequent group discussions. This generates richer data and allows exploring the research issues in greater depth. (Hennink et al. 2020, 142.) However, due to time constraints and the size of the thesis, there is only one focus group discussion. The discussion does still generate data for the purposes of the thesis and can act as a basis for future research.

5.2 Conducting the discussion

The group consisted of participants already familiar with each other and the author. Some advantages of pre-existing include easier recruitment, participation is higher due to shared obligation to attend the discussion and less time is typically needed to develop rapport as participants already know each other. Risks such as over-disclosure and reduced confidentiality were not an issue as the topic had no personal dimensions to it. (Hennink et al. 2020, 151.)

The group discussion started with the author providing the participants with general information about the thesis and what themes will be discussed. Participants gave consent to recording the discussion and are referred as participants A, B, C and D. The discussion started with introductory questions about their experience and the usage of keybinding in games generally. The analysis discusses the grievances and improvement ideas the participants had about keybinding regarding user experience and user interfaces.

5.3 Discussion analysis

Participants had varying levels of experience in keybinding, most experiences coming from MMOs. Three participants described how most of their keybinding customisation in games is done in MMOs, more minimal changes are sometimes done in first person shooting games with actions such as crouching and melee attacks seen in Appendices 3/1 and 3/4. Recurring themes that surfaced during the discussion were about user experience, usability, clarity, and feedback.

Participants B and D expressed having a decent amount of experience in simulator games. Participant D described their experiences with HOTAS controllers and mentioned how simulators require regular playing being one of the reasons they rarely use it. Participant B recounted their experiences specifically with vehicular simulators such as ARMA (2013) and some space simulators and survival games. (Appendix 3/2.)

The user experience with keybinding UIs in general was a source of frustration to all participants. The participants all had similar issues with learning and remembering keybinds. The problem accentuates when there’s even a short break from playing a game with complex gameplay and keybinds (Appendix 3/2).

The incoherence of the keybinding UI was described as annoying by the participants. Participant A expressed their distaste toward the keybinding UI in general, describing the UI layout in MMOs as a “cluster” and the process of keybinding “a hassle” (Appendix 3/4). Participant B held the opinion that the game developers think the UI is designed well but it is usually the total opposite (Appendix 3/5).

The participants shared their views on improvement ideas that could be applied generally to all keybinding UIs in games seen in Appendix 3/6. Clarity and simplicity was emphasised by all participants. Finding the keybinding menu as quickly and easy as possible and testing the system to ensure it is properly functional were important aspects to participant B. Participant C suggested a quick check button for seeing most common but not always used hotkeys to help with remembering binds. Participant D proposed features such as highlighting the most important binds, a search feature for finding a specific action, having explanations what the actions are and tutorials to going through keybinding. Participant A mentioned dummy guides for beginners they have found that instructions online are mostly about doing what feels best for the user. Participant A felt this creates a threshold for beginners who don’t know where to start.

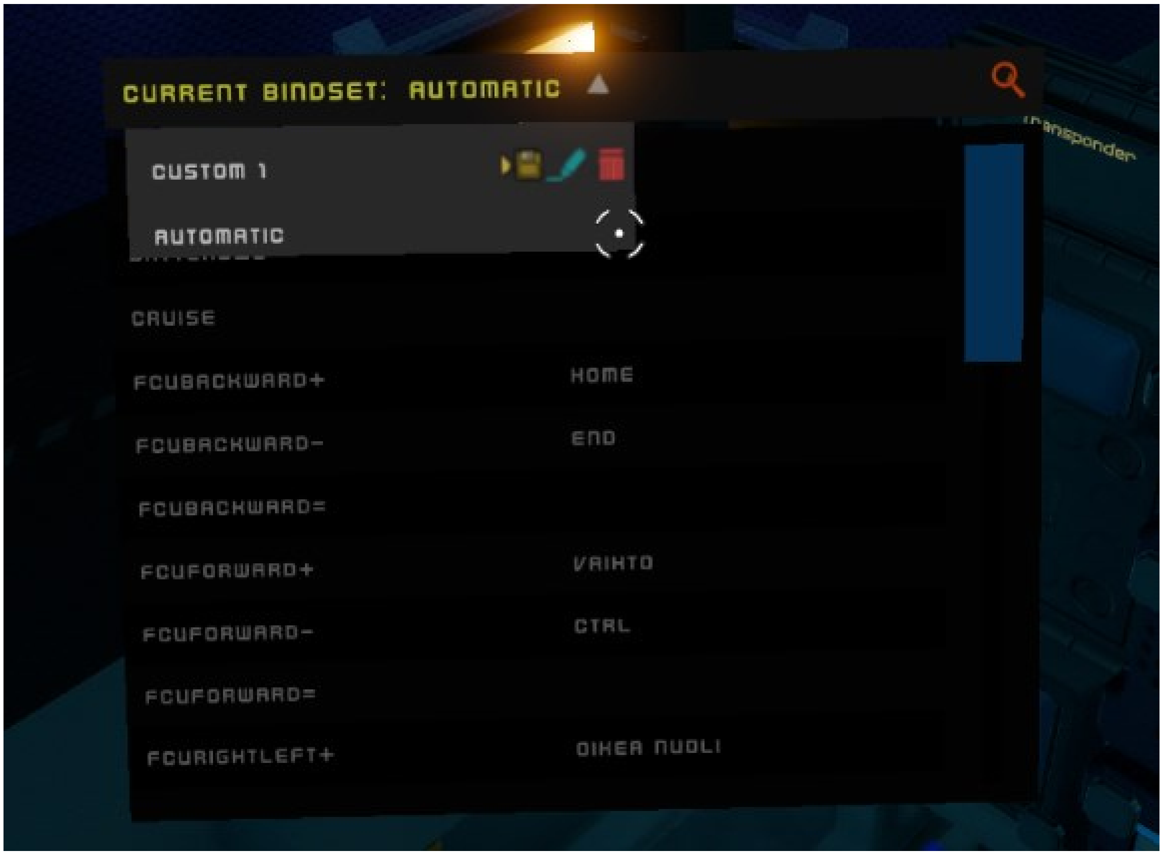
6 STARBASE

Starbase is designed to a particular audience with experience and interest in building their own spaceships and stations from scratch by collecting resources from asteroids and moons and then refining them into building materials. The game has its own coding system for configuring functionalities of spaceships and other machinery. (Starbase 2021.) This chapter covers the design process of the key binding UI for the game’s spaceships.

6.1 Overview of the project

The author started working for Frozenbyte in May 2021 as a user interface designer. The first task was to redesign the input binding menu for spaceships in the game. The original plan was to redesign the key bind-tab in an in-game diegetic Universal Tool menu the player's avatar wields. This was then changed to a separate 2D-window included in a new non-diegetic inventory menu. This menu will contain various windows and its purpose is for the player to access these in whatever situation and as quick as possible. The design process of the project was completed at the end of summer 2021. The UI is not yet physically in the game itself as the new inventory menu itself is still under development. (Starbase 2021.)

The game's standard input binds are accessed in the game's settings and include the basic input binds for operating ships and operate as default input binds for controlling a ship. Changing the standard input binds will affect the default binds in all ships the player uses. Currently in the game the ship specific key binds are in a diegetic menu accessed by sitting on a pilot chair in a spaceship and pressing the letter V on the keyboard (Figure 20).


CURRENT BINDSET: AUTOMATIC

CUSTOM 1

AUTOMATIC

CRUISE	
FCUBACKWARD+	HOME
FCUBACKWARD-	END
FCUBACKWARD=	
FCUFORWARD+	VAIHTO
FCUFORWARD-	CTRL
FCUFORWARD=	
FCURIGHTLEFT+	OIKEA NUOLI

Figure 20. Ship specific key binds accessed by pressing V on keyboard (Frozenbyte 2021)

The idea behind the ship specific key binds is that its system detects all nearby interactable devices in the ship and compiles a list of actions the player can then bind (Starbase 2021).

The menu has a predetermined input binds list, which contains all the basic actions. When an input bind is changed, this automatically creates a custom input binding set. The redesigned menu has the same functioning logic as the ship specific binds list. Additionally, in the redesigned menu’s main purpose is for the player to access the menu quicker and assign pilot chairs from different ships to a desired input binding set. (Starbase 2021.)

The project resulted in over 150 mock up prototypes of the menu. A mock up is a prototype of a design providing designers a method to visualise, evaluate, learn, and improve the design specifications until the wanted result is achieved. The design process of the thesis used evolutionary prototyping where the initial prototype is developed, evaluated, and refined in a loop until it evolves into the final design. (Lidwell et al. 2010, 194.)

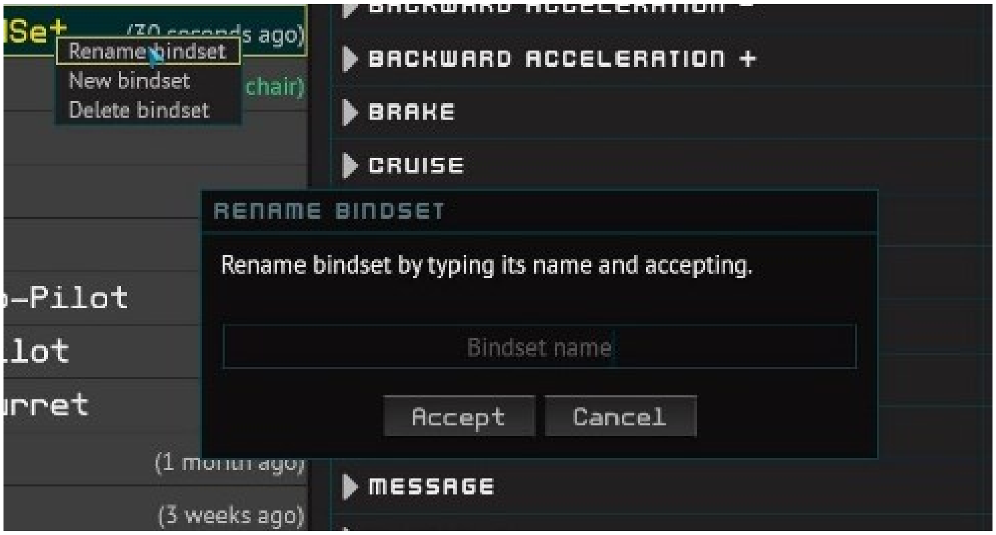
Design iteration occurred between each mock up process. Design iteration occurs when design concepts are explored, tested, and refined, narrowing the wide range of possibilities until the design conforms to design requirements (Lidwell et al. 2010, 142). The interface was designed according to Starbase’s UI style guide. A style guide in UI design is a set of rules dictating how the interface should look and feel. Style guides often contain colour and font directions, logo and icon examples, element spacing and balance rules and other guides regarding the number of objects on screen at a time. (Saunders & Novak 2013, 257.)

6.2 Design process of the interface

Because the interface changed location in the game from the hand-held device of the character to the non-diegetic interface, the hand-held mock ups have been excluded from the thesis and the focus is on the new interface. The first version was based loosely on the ship specific key binding interface seen in Figure 3 with the collapsible custom input binding sets menu. However, the author soon noticed she had misunderstood the core idea of the task and the use of the UI style guide through feedback from colleagues and redesigned the layout.

As the purpose of the key binding interface and general working practices of UI design became clearer to the author thanks to the help from colleagues, a layout following the UI style guide was designed. This layout provided a basis for further iterations. All input binding sets are listed in a navigation tree on the left side of the menu with a search bar helping find a specific input binding set from the list. A button for creating a new custom input binding set is above the search bar and next to it a button for sorting these sets by date. The functionality of the sorting button was unclear thus it was removed. The layout of the menu changed slightly between each iteration composition wise: the button for creating a new input binding set moved to the bottom of the navigation tree, the default input binding set was moved to upper left corner and separated from custom input binding sets and a clear sorting button was added that shows what sorting mode is used.

The interface also contains feedback elements. These elements are mostly entrenched in usability considerations and are designed to make a game more enjoyable and less frustrating to the player. (Saunders & Novak 2013, 224.) The menu uses confirmation windows when creating, renaming, or deleting a input binding set (Figure 21) and when binding an input to an action.

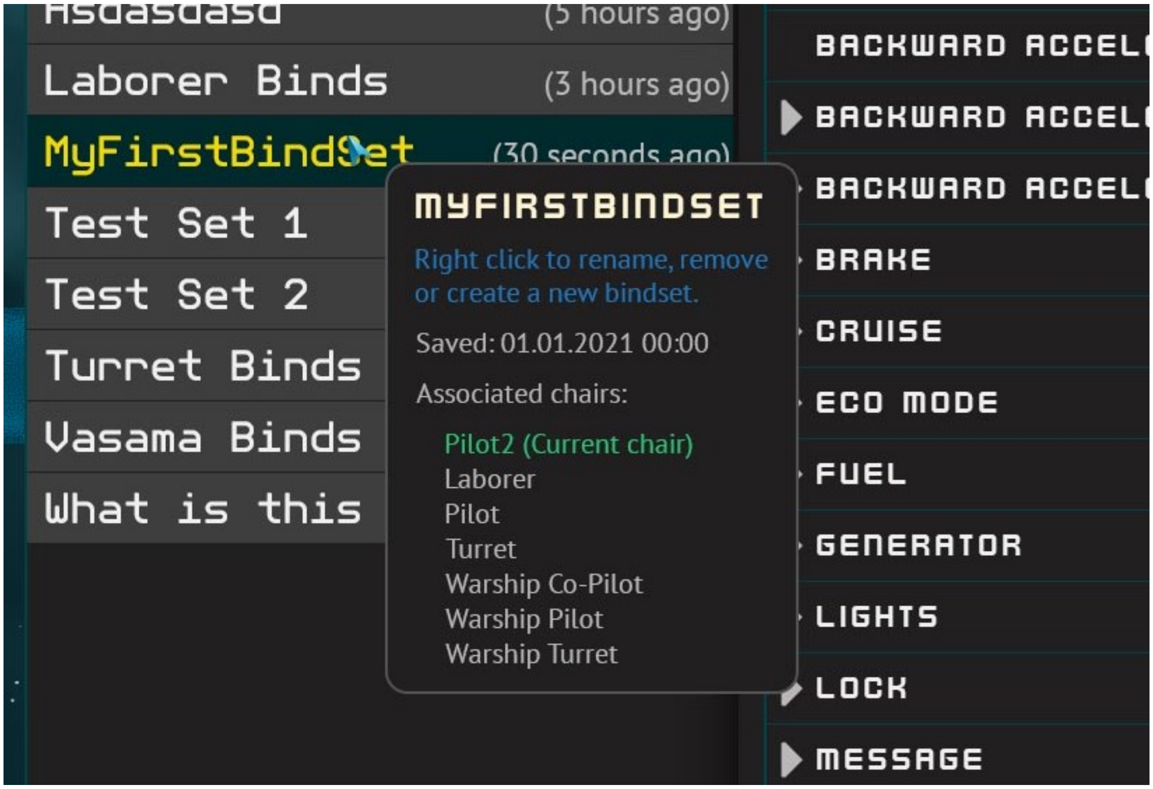


Rename bindset by typing its name and accepting.

Figure 21. Pop-up menu accessed via right click menu for renaming a input binding set

The context menu, or right-click menu that is accessed by right-clicking on an element in the interface. These menus provide shortcuts for actions anticipated the player might take (Bigelow 2014).

Additional information about an input binding set is shown in a tooltip when hovered over any of the custom sets and allows the player to quickly browse through them without needing to click on each one and costing the player time (Figure 22).



MYFIRSTBINDSET

Right click to rename, remove or create a new bindset.

Saved: 01.01.2021 00:00

Associated chairs:

Pilot2 (Current chair)
Laborer
Pilot
Turret
Warship Co-Pilot
Warship Pilot
Warship Turret

Figure 22. Hovering over an input binding set opens a tooltip showing further information

The tooltip shows the input binding set name, instructions on editing the set, date and time the set was saved and pilot chairs using the input binding set in question.

The manner of visually communicating different types of information required colour differentiation (Figure 23). Colour differentiation refers to the process of using colours to improve visibility and provide feedback to the player.

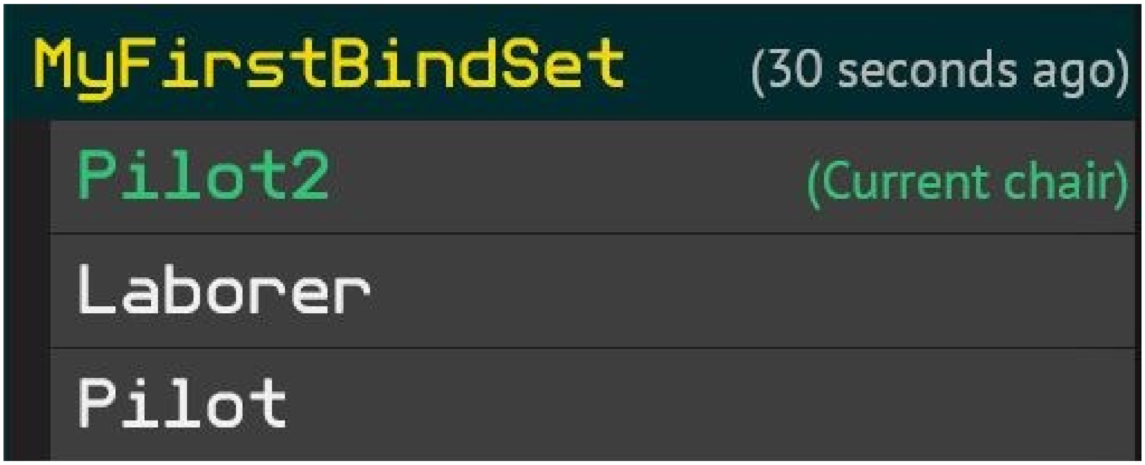


Figure 23. Information inside brackets offer further clarification to the player

This is commonly used to show how seemingly similar interface elements possess different properties such as health and energy bars (Saunders & Novak 2013, 234). Colour differentiation is used in the input binding set tooltip and when showing the active chair in the navigation tree.

To help understand how the binding process is accomplished, the process has been illustrated as a flowchart (Figure 24). A flowchart is one part of interface design documentation helping with communicating design details to a programmer who will implement and create the interface in the game. It is a graphical depiction of the interface's functionality without the need for a detail long description in text. Each interface screen is presented by a box with arrows indicating the relation between the screens and elements to each other. (Saunders and Novak 2013, 225.)

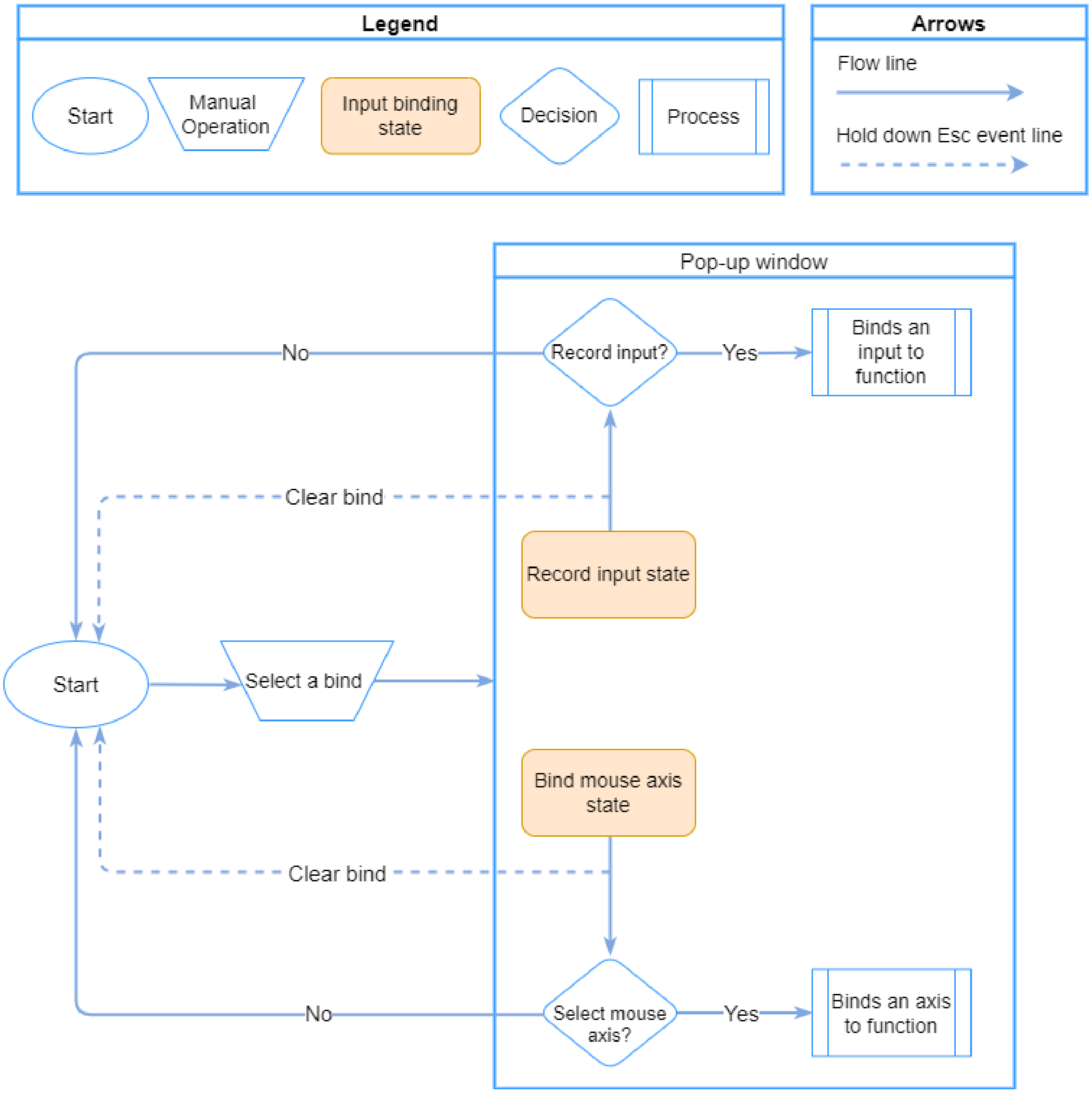
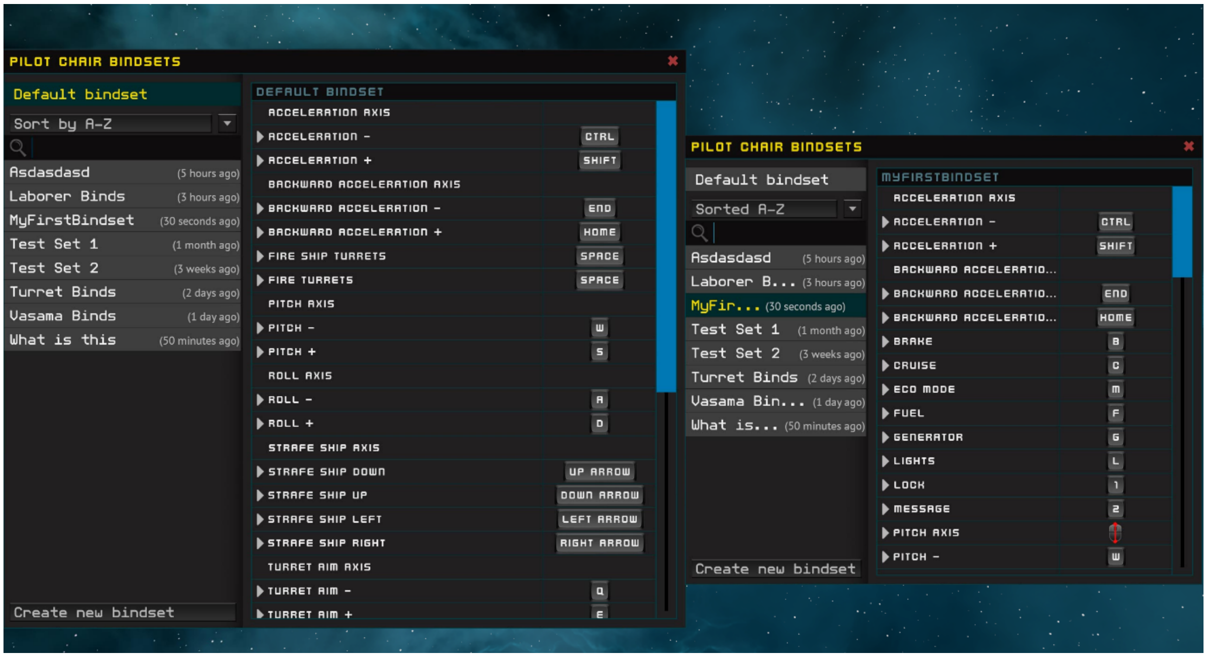


Figure 24. Flowchart of the keyboard or mouse input binding process

The flowchart shows the process of keyboard and mouse input binding. Binding is done by clicking on any of the cells in the binds column on the input binding set list, except any unavailable binds the chair does not detect around. This opens a Bind input pop-up ready to record a keyboard button, left and right mouse button or an analog controller button and axis. The pop-up automatically closes when an input is assigned. (Starbase 2021.)

The final design of the interface when opened has a set default size and can be dragged across the screen. The window is resizable for the player and has no set maximum dimensions. However, the window does have a minimum size that cannot be reduced past its set dimensions. (Figure 25.)



PILOT CHAIR BINDSETS

Default bindset

Sort by A-Z

Asdasdasd	(5 hours ago)
Laborer Binds	(3 hours ago)
MyFirstBindset	(30 seconds ago)
Test Set 1	(1 month ago)
Test Set 2	(3 weeks ago)
Turret Binds	(2 days ago)
Vasama Binds	(1 day ago)
What is this	(50 minutes ago)

Create new bindset

DEFAULT BINDSET	
ACCELERATION AXIS	
ACCELERATION -	CTRL
ACCELERATION +	SHIFT
BACKWARD ACCELERATION AXIS	
BACKWARD ACCELERATION -	END
BACKWARD ACCELERATION +	HOME
FIRE SHIP TURRETS	SPACE
FIRE TURRETS	SPACE
PITCH AXIS	
PITCH -	W
PITCH +	S
ROLL AXIS	
ROLL -	A
ROLL +	D
STRAFE SHIP AXIS	
STRAFE SHIP DOWN	UP ARROW
STRAFE SHIP UP	DOWN ARROW
STRAFE SHIP LEFT	LEFT ARROW
STRAFE SHIP RIGHT	RIGHT ARROW
TURRET AIM AXIS	
TURRET AIM -	Q
TURRET AIM +	E

PILOT CHAIR BINDSETS

Default bindset

Sorted A-Z

Asdasdasd	(5 hours ago)
Laborer B...	(3 hours ago)
MyFir...	(30 seconds ago)
Test Set 1	(1 month ago)
Test Set 2	(3 weeks ago)
Turret Binds	(2 days ago)
Vasama Bin...	(1 day ago)
What is...	(50 minutes ago)

Create new bindset

MYFIRSTBINDSET	
ACCELERATION AXIS	
ACCELERATION -	CTRL
ACCELERATION +	SHIFT
BACKWARD ACCELERATIO...	
BACKWARD ACCELERATIO...	END
BACKWARD ACCELERATIO...	HOME
BRAKE	B
CRUISE	C
ECO MODE	M
FUEL	F
GENERATOR	G
LIGHTS	L
LOCK	1
MESSAGE	2
PITCH AXIS	
PITCH -	W

Figure 25. Menu’s default and minimum size comparison

The pilot chair bindsets menu will be found in the new inventory menu window; however, as the menu is still in the design process the exact location of where to find it is yet to be defined.

7 RESULTS AND CONCLUSION

The main results showed that input binding interfaces in simulator games are designed by utilising conventions and thinking about the game and the player’s needs. Keybinding UIs should list and categorise actions, a logical order being the categories integral to the gameplay itself located at the top of the list. The categories should be elements that both show and hide actions to reduce unnecessary clutter and confusing the player. Today’s simulator games offer support for the most common controllers and have ready-made bindings done to reduce the time and effort the player would otherwise spend on binding. A great convention used by some simulator games is to have the ability to bind multiple inputs to the same action. This provides the player more freedom in customising their setup.

The author’s work at Frozenbyte elucidated the general design pipeline of input binding interfaces. A substantial amount of time in design was spent in finding out the functionalities needed for the interface and multiple iterations had to occur during the process.

In order to design a more easily approachable interface to player with differing levels of experience in simulator games, asking the players themselves offer considerable amount of insight in terms of user experience. Simple UI layout reduces player frustration and having as linear path to the menu as possible eases the binding process and saves time for the player to focus on playing the game itself. A visual controller layout guide similar to Star Citizen can lower the threshold for new players wanting to try simulator games and veteran players can quickly check the default binds. Helping measures in the UI itself such as a search feature to quickly find specific action would be beneficial. Including explanations to the actions would clarify them to the player further.

The study succeeded in establishing a basis for future research regarding the subject and offer improvement ideas regarding the keybinding UI for the commissioner. The study to be more comprehensive requires benchmarking a larger quantity of simulator games to build a list of all conventions used in the keybinding UI. Accessibility issues regarding these games would make a good study subject all by itself. Additionally, more time would be required to specifically find players with more experience in simulator games and conduct either group discussions or individual interviews to gather more data.

REFERENCES

Bigelow, S. 2014. Context menu (right-click menu). Web page. Available at: https://www.techtarget.com/searchenterprisedesktop/definition/context-menu-pop-up-menu [Accessed 11 April 2022].

Bohemia Interactive Studio. 2013. ARMA 3. Video Game. Prague: Bohemia Interactive Studio.

Cloud Imperium Games. 2017. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Fagerholt, E. & Lorentzon, M. 2009. Beyond the HUD: User Interfaces for Increased Player Immersion in FPS Games. Chalmers University of Technology. Department of Computer Science and Engineering. Master’s Thesis. PDF document. Available at: https://publications.lib.chalmers.se/records/fulltext/111921.pdf [Accessed 12 March 2022].

Frontier Developments. 2014. Elite: Dangerous. Video Game. Cambridge: Frontier Developments.

Frozenbyte. 2021. Starbase. Video Game. Helsinki: Frozenbyte.

Game Developers Conference. 2022. State of the Game Industry 2022. PDF-document. Available at: https://reg.gdconf.com/state-of-game-industry-2022?BLG_GDC&_mc=blog_x_gdcsfr_un_x_gdcsf_x_x-13 [Accessed 12 March 2022].

GameGlass. 2022. Games. Web page. Available at: https://gameglass.gg/pages/games2 [Accessed 13 March 2022].

Hamilton, I. 2019. Demystifying CVAA. Blog. 23 January 2019. Available at: https://www.gamedeveloper.com/business/demystifying-cvaa [Accessed 24 March 2022].

Hennink, M., Hutter, I. & Bailey, A. 2020. Qualitative Research Methods. 2nd ed. London: Sage Publications Ltd.

Ixie Gaming. 2021. Three major stages of game testing. Blog. 22 December 2021. Available at: https://www.ixiegaming.com/blog/three-major-stages-of-game-testing/ [Accessed 20 May 2022].

Lankoski, P. & Björk, S. 2015. Game Research Methods. An Overview. Pittsburgh: ETC Press. Available at: https://mediarep.org/handle/doc/14507 [Accessed 21 February 2022].

Lidwell, W., Holden, K. & Butler, J. 2010. Universal Principles of Design, Revised and Updated. Minneapolis: Quarto Publishing Group USA. Available at: https://ebookcentral.proquest.com/ [Accessed 13 March 2022].

Lynch, P. & Horton, S. 2009. Web Style Guide: Basic Design Principles for Creating Web Sites. 3rd ed. Connecticut: Yale University Press. Available at: https://ebookcentral.proquest.com/ [Accessed 8 April 2022].

MIT Media Lab. n.d. Key Bindings. Web page. Available at: https://web.media.mit.edu/~benb/statics/bssmanual/MAIN/Key_Bindings.htm [Accessed 19 March 2022].

McAloon, A. CVAA accessibility rules come into effect for games as FCC waiver expires. Blog. 3 January 2019. Available at: https://www.gamedeveloper.com/disciplines/cvaa-accessibility-rules-come-into-effect-for-games-as-fcc-waiver-expires [Accessed 24 March 2022].

Norman, D. & Nielsen, J. 1998. The Definition of User Experience (UX). Web page. Available at: https://www.nngroup.com/articles/definition-user-experience/ [Accessed 5 May 2022].

Roberts Space Industries. 2017. Star Citizen Alpha 3.0.0. Web page. 23 December 2017. Available at: https://robertsspaceindustries.com/comm-link/transmission/16348-star-citizen-alpha-300 [Accessed 20 May 2022].

Saunders, K. & Novak, J. 2013. Game Development Essentials: Game Interface Design. 2nd ed. New York: Delmar.

Schell, J. 2015. The Art of Game Design: A Book of Lenses. 2nd ed. Florida: CRC Press.

Technopedia. 2021. Graphical User Interface (GUI). Web page. Available at: https://www.techopedia.com/definition/5435/graphical-user-interface-gui [Accessed 14 March 2022].

Tuominen, K. & Niva, M. 2005. Benchmarking käytännössä. Turku: Oy Benchmarking Ltd. Available at: https://ebookcentral.proquest.com/ [Accessed 27 March 2022].

Twenty-First Century Communications and Video Accessibility Act 2010.

Valve. n.d. Early Access. Web page. Available at: https://partner.steamgames.com/doc/store/earlyaccess [Accessed 15 May 2022].

Verkkokauppa.com. 2022. Peliohjaimet ja -tarvikkeet. Web page. Available at: https://www.verkkokauppa.com/fi/catalog/680b/Peliohjaimet-ja-tarvikkeet [Accessed 4 May 2022].

LISTS OF FIGURES OR TABLES

Figure 10. Confirmation window for binding a new input. Frontier Developments. 2014. Elite: Dangerous. Video Game. Cambridge: Frontier Developments.

Figure 11. Information on an action’s functionality highlighted with red. Frontier Developments. 2014. Elite: Dangerous. Video Game. Cambridge: Frontier Developments.

Figure 12. Collapsibles. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 13. Keyboard and mouse prompts. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 14. Disable-toggle. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 15. Clicking the curve line creates points of further adjustment. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 16. Binding state timer. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 17. Input binding confirmation window. Cloud Imperium Games. 2022. Star Citizen Alpha. Video Game. Los Angeles: Cloud Imperium Games.

Figure 18. Portion of each input in both UIs. Korhonen, E. 2022.

Figure 19. Funnel design of the discussion guide. Hennink, M., Hutter, I. & Bailey, A. 2020. Qualitative Research Methods. 2nd ed. London: Sage Publications Ltd.

Figure 20. Ship specific key binds accessed by pressing V on keyboard. Frozenbyte. 2022. Starbase. Video Game. Helsinki: Frozenbyte.

Figure 21. Pop-up menu accessed via right click menu for renaming a input binding set. Korhonen, E. 2022.

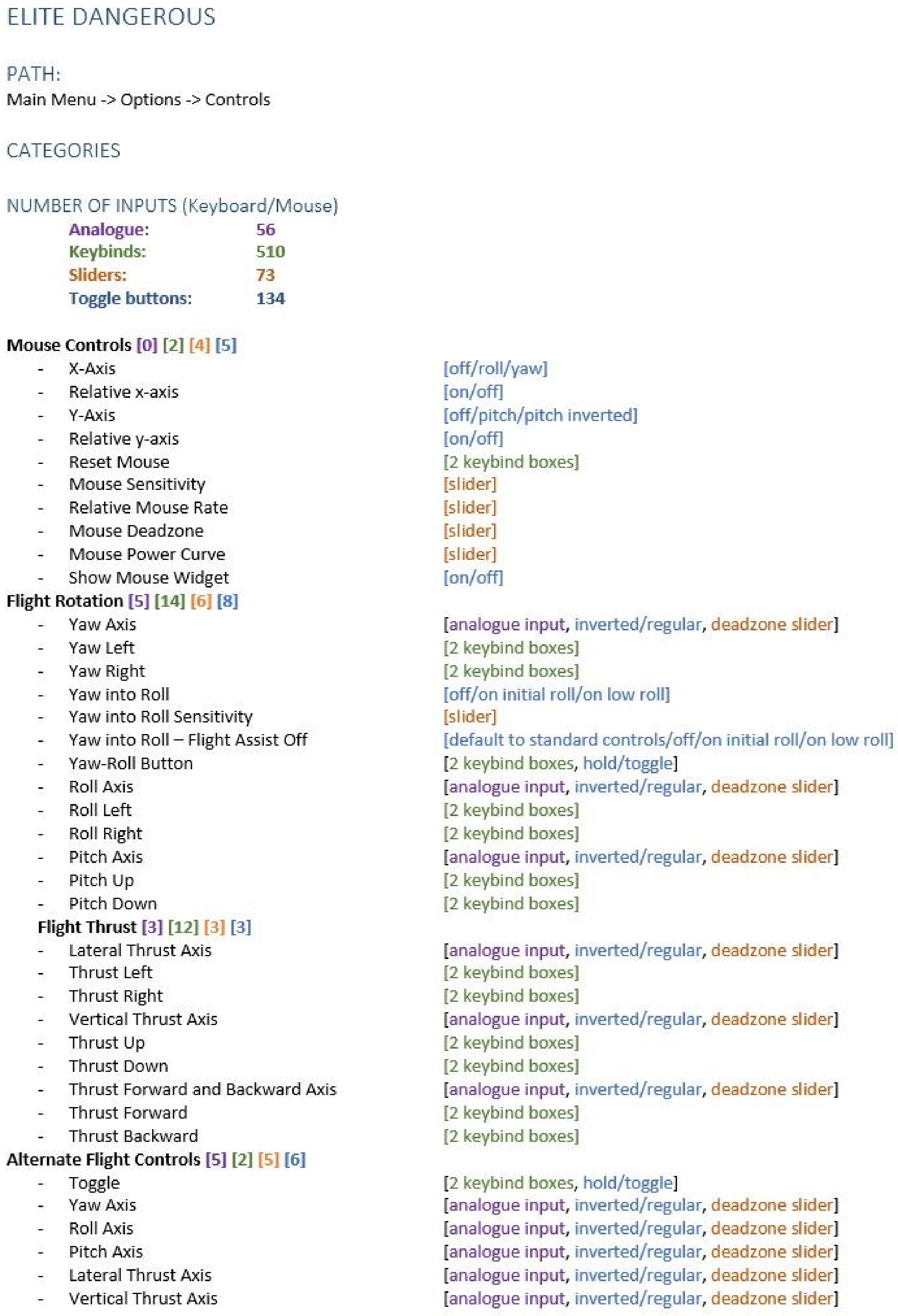
Figure 22. Hovering over a input binding set opens a tooltip showing further information. Korhonen, E. 2022.

Figure 23. Information inside brackets offer further clarification to the player. Korhonen, E. 2022.

Figure 24. Flowchart of the keyboard or mouse input binding process. Korhonen, E. 2022.

Figure 25. Menu’s default and minimum size comparison. Korhonen, E. 2022.

ELITE: DANGEROUS INTERFACE BREAKDOWN



ELITE DANGEROUS

PATH:

Main Menu -> Options -> Controls

CATEGORIES

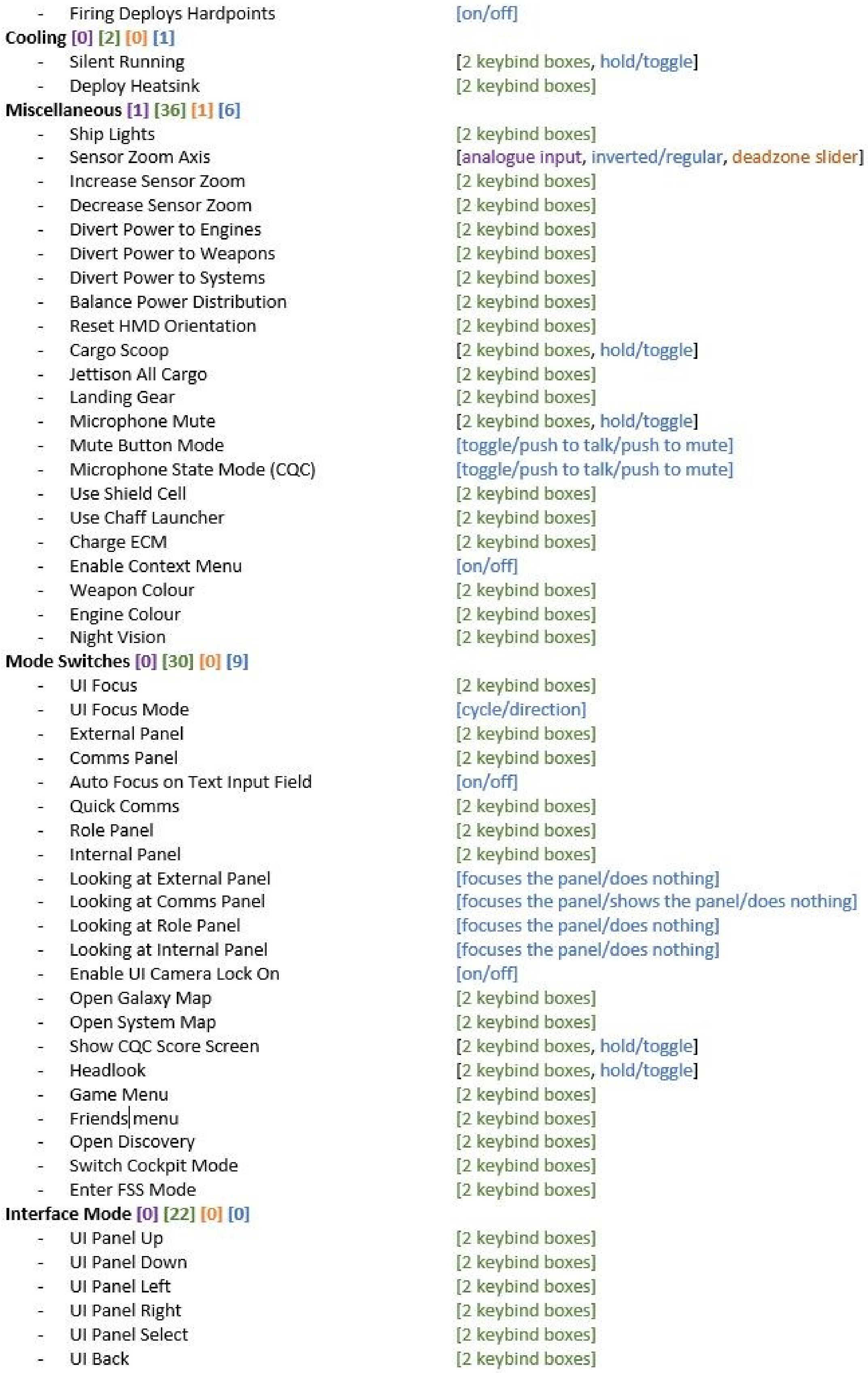
NUMBER OF INPUTS (Keyboard/Mouse)

Analogue:	**56**
Keybinds:	**510**
Sliders:	**73**
Toggle buttons:	**134**

Mouse Controls [0] [2] [4] [5]	
- X-Axis	[off/roll/yaw]
- Relative x-axis	[on/off]
- Y-Axis	[off/pitch/pitch inverted]
- Relative y-axis	[on/off]
- Reset Mouse	[2 keybind boxes]
- Mouse Sensitivity	[slider]
- Relative Mouse Rate	[slider]
- Mouse Deadzone	[slider]
- Mouse Power Curve	[slider]
- Show Mouse Widget	[on/off]
Flight Rotation [5] [14] [6] [8]	
- Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Yaw Left	[2 keybind boxes]
- Yaw Right	[2 keybind boxes]
- Yaw into Roll	[off/on initial roll/on low roll]
- Yaw into Roll Sensitivity	[slider]
- Yaw into Roll – Flight Assist Off	[default to standard controls/off/on initial roll/on low roll]
- Yaw-Roll Button	[2 keybind boxes, hold/toggle]
- Roll Axis	[analogue input, inverted/regular, deadzone slider]
- Roll Left	[2 keybind boxes]
- Roll Right	[2 keybind boxes]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Pitch Up	[2 keybind boxes]
- Pitch Down	[2 keybind boxes]
Flight Thrust [3] [12] [3] [3]	
- Lateral Thrust Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Left	[2 keybind boxes]
- Thrust Right	[2 keybind boxes]
- Vertical Thrust Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Up	[2 keybind boxes]
- Thrust Down	[2 keybind boxes]
- Thrust Forward and Backward Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Forward	[2 keybind boxes]
- Thrust Backward	[2 keybind boxes]
Alternate Flight Controls [5] [2] [5] [6]	
- Toggle	[2 keybind boxes, hold/toggle]
- Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Roll Axis	[analogue input, inverted/regular, deadzone slider]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Lateral Thrust Axis	[analogue input, inverted/regular, deadzone slider]
- Vertical Thrust Axis	[analogue input, inverted/regular, deadzone slider]

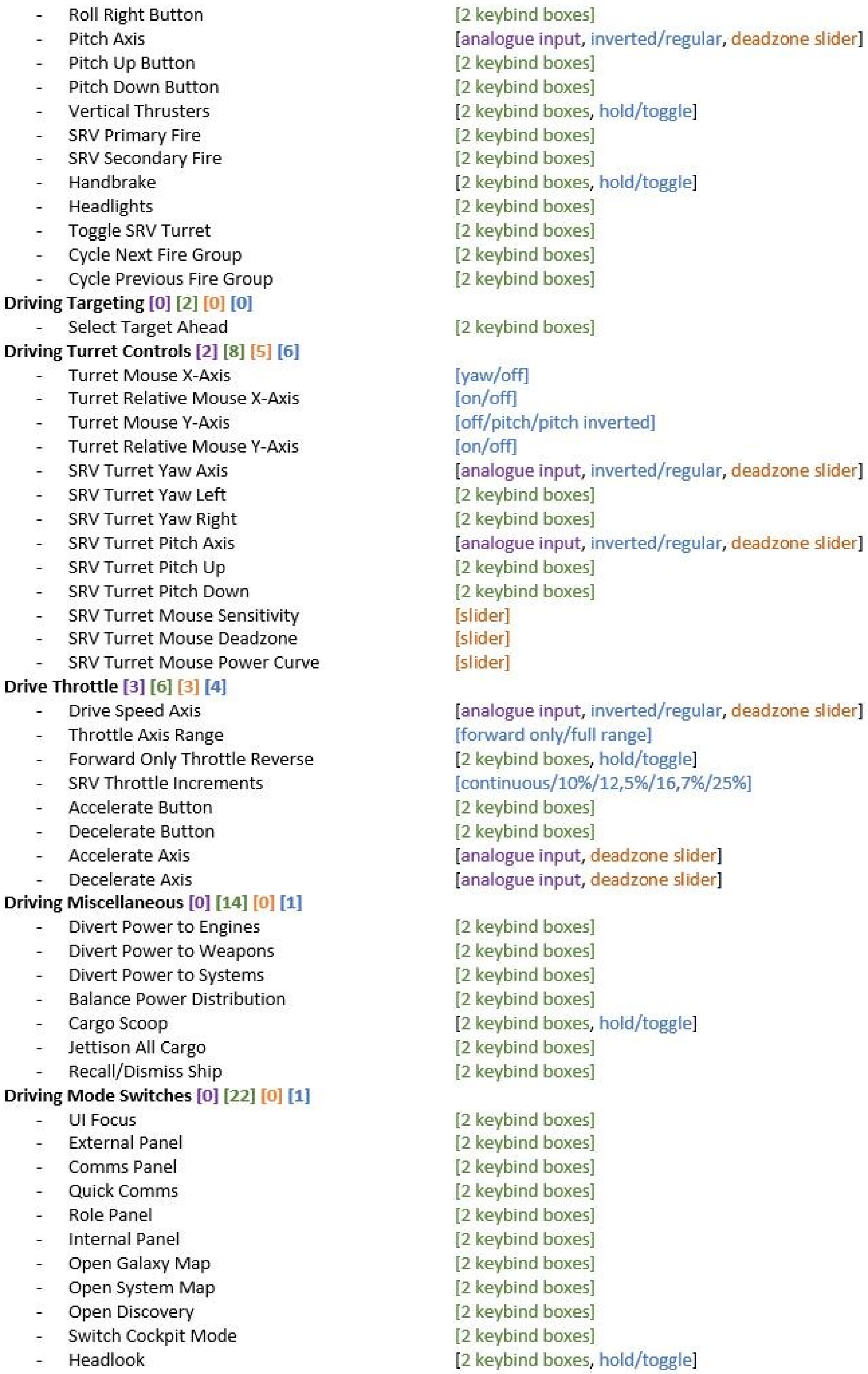
Flight Throttle [1] [8] [1] [4]	
- Throttle Axis	[analogue input, inverted/regular, deadzone slider]
- Throttle Axis Range	[full range/forward only]
- Forward only Throttle Reverse	[2 keybind boxes, hold/toggle]
- Increase Throttle	[2 keybind boxes]
- Decrease Throttle	[2 keybind boxes]
- Throttle Increments	[continuous/10%/12,5%/16,7%/25%]
- Set Speed To (from -100% to 100%)	[2 keybind boxes in each (18 in total)]
Flight Landing Overrides [6] [24] [6] [7]	
- Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Yaw Left	[2 keybind boxes]
- Yaw Right	[2 keybind boxes]
- Yaw into Roll	[default to standard controls/off/on initial roll/ on low roll]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Pitch Up	[2 keybind boxes]
- Pitch Down	[2 keybind boxes]
- Roll Axis	[analogue input, inverted/regular, deadzone slider]
- Roll Left	[2 keybind boxes]
- Roll Right	[2 keybind boxes]
- Lateral Thrust Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Left	[2 keybind boxes]
- Thrust Right	[2 keybind boxes]
- Vertical Thrust Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Up	[2 keybind boxes]
- Thrust Down	[2 keybind boxes]
- Thrust Forward and Backward Axis	[analogue input, inverted/regular, deadzone slider]
- Thrust Forward	[2 keybind boxes]
- Thrust Backward	[2 keybind boxes]
Flight Miscellaneous [0] [14] [0] [2]	
- Toggle Flight Assist	[2 keybind boxes, hold/toggle]
- Engine Boost	[2 keybind boxes]
- Toggle Frame Shift Drive	[2 keybind boxes]
- Supercruise	[2 keybind boxes]
- Hyperspace Jump	[2 keybind boxes]
- Rotational Correction	[2 keybind boxes, hold/toggle]
- Toggle Orbit Lines	[2 keybind boxes]
Targeting [0] [28] [0] [0]	
- Select Target Ahead	[2 keybind boxes]
- Cycle Next Target	[2 keybind boxes]
- Cycle Previous Ship	[2 keybind boxes]
- Select Highest Threat	[2 keybind boxes]
- Cycle Next Hostile Target	[2 keybind boxes]
- Cycle Previous Hostile Ship	[2 keybind boxes]
- Select Wingman 1	[2 keybind boxes]
- Select Wingman 2	[2 keybind boxes]
- Select Wingman 3	[2 keybind boxes]
- Select Wingman's Target	[2 keybind boxes]
- Wingman Nav-Lock	[2 keybind boxes]
- Cycle Next Subsystem	[2 keybind boxes]
- Cycle Previous Subsystem	[2 keybind boxes]
- Target Next System in Route	[2 keybind boxes]
Weapons [0] [10] [0] [1]	
- Primary Fire	[2 keybind boxes]
- Secondary Fire	[2 keybind boxes]
- Cycle Next Fire Group	[2 keybind boxes]
- Cycle Previous Fire Group	[2 keybind boxes]
- Deploy Hardpoints	[2 keybind boxes]

Appendix 1/3



- Firing Deploys Hardpoints	[on/off]
Cooling [0] [2] [0] [1]	
- Silent Running	[2 keybind boxes, hold/toggle]
- Deploy Heatsink	[2 keybind boxes]
Miscellaneous [1] [36] [1] [6]	
- Ship Lights	[2 keybind boxes]
- Sensor Zoom Axis	[analogue input, inverted/regular, deadzone slider]
- Increase Sensor Zoom	[2 keybind boxes]
- Decrease Sensor Zoom	[2 keybind boxes]
- Divert Power to Engines	[2 keybind boxes]
- Divert Power to Weapons	[2 keybind boxes]
- Divert Power to Systems	[2 keybind boxes]
- Balance Power Distribution	[2 keybind boxes]
- Reset HMD Orientation	[2 keybind boxes]
- Cargo Scoop	[2 keybind boxes, hold/toggle]
- Jettison All Cargo	[2 keybind boxes]
- Landing Gear	[2 keybind boxes]
- Microphone Mute	[2 keybind boxes, hold/toggle]
- Mute Button Mode	[toggle/push to talk/push to mute]
- Microphone State Mode (CQC)	[toggle/push to talk/push to mute]
- Use Shield Cell	[2 keybind boxes]
- Use Chaff Launcher	[2 keybind boxes]
- Charge ECM	[2 keybind boxes]
- Enable Context Menu	[on/off]
- Weapon Colour	[2 keybind boxes]
- Engine Colour	[2 keybind boxes]
- Night Vision	[2 keybind boxes]
Mode Switches [0] [30] [0] [9]	
- UI Focus	[2 keybind boxes]
- UI Focus Mode	[cycle/direction]
- External Panel	[2 keybind boxes]
- Comms Panel	[2 keybind boxes]
- Auto Focus on Text Input Field	[on/off]
- Quick Comms	[2 keybind boxes]
- Role Panel	[2 keybind boxes]
- Internal Panel	[2 keybind boxes]
- Looking at External Panel	[focuses the panel/does nothing]
- Looking at Comms Panel	[focuses the panel/shows the panel/does nothing]
- Looking at Role Panel	[focuses the panel/does nothing]
- Looking at Internal Panel	[focuses the panel/does nothing]
- Enable UI Camera Lock On	[on/off]
- Open Galaxy Map	[2 keybind boxes]
- Open System Map	[2 keybind boxes]
- Show CQC Score Screen	[2 keybind boxes, hold/toggle]
- Headlook	[2 keybind boxes, hold/toggle]
- Game Menu	[2 keybind boxes]
- Friends menu	[2 keybind boxes]
- Open Discovery	[2 keybind boxes]
- Switch Cockpit Mode	[2 keybind boxes]
- Enter FSS Mode	[2 keybind boxes]
Interface Mode [0] [22] [0] [0]	
- UI Panel Up	[2 keybind boxes]
- UI Panel Down	[2 keybind boxes]
- UI Panel Left	[2 keybind boxes]
- UI Panel Right	[2 keybind boxes]
- UI Panel Select	[2 keybind boxes]
- UI Back	[2 keybind boxes]

- UI Nested Toggle	[2 keybind boxes]
- Next Panel Tab	[2 keybind boxes]
- Previous Panel Tab	[2 keybind boxes]
- Next Page	[2 keybind boxes]
- Previous Page	[2 keybind boxes]
Headlook Mode [2] [10] [2] [7]	
- Mouse Headlook	[on/off]
- Mouse Headlook Invert	[on/off]
- Mouse Headlook Sensitivity	[slider]
- Headlook Default State	[on/off]
- Headlook Button Increments	[continuous/small-/medium-/large increments]
- Headlook Axis Mode	[accumulative/direct]
- Centre When Headlook Inactive	[on/off]
- Headlook Sensitivity	[slider]
- Headlook Smoothing	[on/off]
- Reset Headlook	[2 keybind boxes]
- Look Up	[2 keybind boxes]
- Look Down	[2 keybind boxes]
- Look Up and Down Axis	[analogue input, inverted/regular, deadzone slider]
- Look Left	[2 keybind boxes]
- Look Right	[2 keybind boxes]
- Look Left and Right Axis	[analogue input, inverted/regular, deadzone slider]
Galaxy Map [7] [26] [7] [7]	
- Galaxy Cam Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Pitch Up	[2 keybind boxes]
- Galaxy Cam Pitch Down	[2 keybind boxes]
- Galaxy Cam Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Yaw Left	[2 keybind boxes]
- Galaxy Cam Yaw Right	[2 keybind boxes]
- Galaxy Cam Translate Y-Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Translate Forward	[2 keybind boxes]
- Galaxy Cam Translate Backward	[2 keybind boxes]
- Galaxy Cam Translate X-Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Translate Left	[2 keybind boxes]
- Galaxy Cam Translate Right	[2 keybind boxes]
- Galaxy Cam Translate Z-Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Translate Up	[2 keybind boxes]
- Galaxy Cam Translate Down	[2 keybind boxes]
- Galaxy Cam Zoom Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Zoom In	[2 keybind boxes]
- Galaxy Cam Zoom Out	[2 keybind boxes]
- Galaxy Cam Set Y-Axis to Z-Axis	[analogue input, inverted/regular, deadzone slider]
- Galaxy Cam Select Current System	[2 keybind boxes]
Driving [3] [30] [3] [13]	
- Drive Assist	[2 keybind boxes, hold/toggle]
- Drive Assist Default	[on/off]
- SRV Steering Mouse X-Axis	[on/off]
- Relative Mouse X-Axis Steer	[on/off]
- SRV Rolling Mouse X-Axis	[on/off]
- Relative Mouse X-Axis Roll	[on/off]
- SRV Pitch Mouse Y-Axis	[off/pitch/pitch inverted]
- Relative Mouse Y-Axis Pitch	[on/off]
- Steering Axis	[analogue input, inverted/regular, deadzone slider]
- Steering Left Button	[2 keybind boxes]
- Steering Right Button	[2 keybind boxes]
- Roll Axis	[analogue input, inverted/regular, deadzone slider]
- Roll Left Button	[2 keybind boxes]



- Roll Right Button	[2 keybind boxes]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Pitch Up Button	[2 keybind boxes]
- Pitch Down Button	[2 keybind boxes]
- Vertical Thrusters	[2 keybind boxes, hold/toggle]
- SRV Primary Fire	[2 keybind boxes]
- SRV Secondary Fire	[2 keybind boxes]
- Handbrake	[2 keybind boxes, hold/toggle]
- Headlights	[2 keybind boxes]
- Toggle SRV Turret	[2 keybind boxes]
- Cycle Next Fire Group	[2 keybind boxes]
- Cycle Previous Fire Group	[2 keybind boxes]
Driving Targeting [0] [2] [0] [0]	
- Select Target Ahead	[2 keybind boxes]
Driving Turret Controls [2] [8] [5] [6]	
- Turret Mouse X-Axis	[yaw/off]
- Turret Relative Mouse X-Axis	[on/off]
- Turret Mouse Y-Axis	[off/pitch/pitch inverted]
- Turret Relative Mouse Y-Axis	[on/off]
- SRV Turret Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- SRV Turret Yaw Left	[2 keybind boxes]
- SRV Turret Yaw Right	[2 keybind boxes]
- SRV Turret Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- SRV Turret Pitch Up	[2 keybind boxes]
- SRV Turret Pitch Down	[2 keybind boxes]
- SRV Turret Mouse Sensitivity	[slider]
- SRV Turret Mouse Deadzone	[slider]
- SRV Turret Mouse Power Curve	[slider]
Drive Throttle [3] [6] [3] [4]	
- Drive Speed Axis	[analogue input, inverted/regular, deadzone slider]
- Throttle Axis Range	[forward only/full range]
- Forward Only Throttle Reverse	[2 keybind boxes, hold/toggle]
- SRV Throttle Increments	[continuous/10%/12,5%/16,7%/25%]
- Accelerate Button	[2 keybind boxes]
- Decelerate Button	[2 keybind boxes]
- Accelerate Axis	[analogue input, deadzone slider]
- Decelerate Axis	[analogue input, deadzone slider]
Driving Miscellaneous [0] [14] [0] [1]	
- Divert Power to Engines	[2 keybind boxes]
- Divert Power to Weapons	[2 keybind boxes]
- Divert Power to Systems	[2 keybind boxes]
- Balance Power Distribution	[2 keybind boxes]
- Cargo Scoop	[2 keybind boxes, hold/toggle]
- Jettison All Cargo	[2 keybind boxes]
- Recall/Dismiss Ship	[2 keybind boxes]
Driving Mode Switches [0] [22] [0] [1]	
- UI Focus	[2 keybind boxes]
- External Panel	[2 keybind boxes]
- Comms Panel	[2 keybind boxes]
- Quick Comms	[2 keybind boxes]
- Role Panel	[2 keybind boxes]
- Internal Panel	[2 keybind boxes]
- Open Galaxy Map	[2 keybind boxes]
- Open System Map	[2 keybind boxes]
- Open Discovery	[2 keybind boxes]
- Switch Cockpit Mode	[2 keybind boxes]
- Headlook	[2 keybind boxes, hold/toggle]

Multi-Crew [3] [26] [3] [7]

-	Mode Toggle	[2 keybind boxes]
-	Primary Fire	[2 keybind boxes]
-	Secondary Fire	[2 keybind boxes]
-	Primary Utility Fire	[2 keybind boxes]
-	Secondary Utility Fire	[2 keybind boxes]
-	Mouse X-Axis	[yaw/off]
-	Relative Mouse X-Axis	[on/off]
-	Mouse Y-Axis	[off/pitch/pitch inverted]
-	Relative Mouse Y-Axis	[on/off]
-	Third-person Yaw Axis	[analogue input, inverted/regular, deadzone slider]
-	Third-person Yaw Left	[2 keybind boxes]
-	Third-person Yaw Right	[2 keybind boxes]
-	Third-person Pitch Axis	[analogue input, inverted/regular, deadzone slider]
-	Third-person Pitch Up	[2 keybind boxes]
-	Third-person Pitch Down	[2 keybind boxes]
-	Multi-Crew Mouse Sensitivity	[slider]
-	Third-person Field of View Axis	[analogue input, inverted/regular, deadzone slider]
-	Third-person Field of View Out	[2 keybind boxes]
-	Third-person Field of View In	[2 keybind boxes]
-	Cycle Cockpit UI Forwards	[2 keybind boxes]
-	Cycle Cockpit UI Backwards	[2 keybind boxes]

Fighter Orders [0] [16] [0] [0]

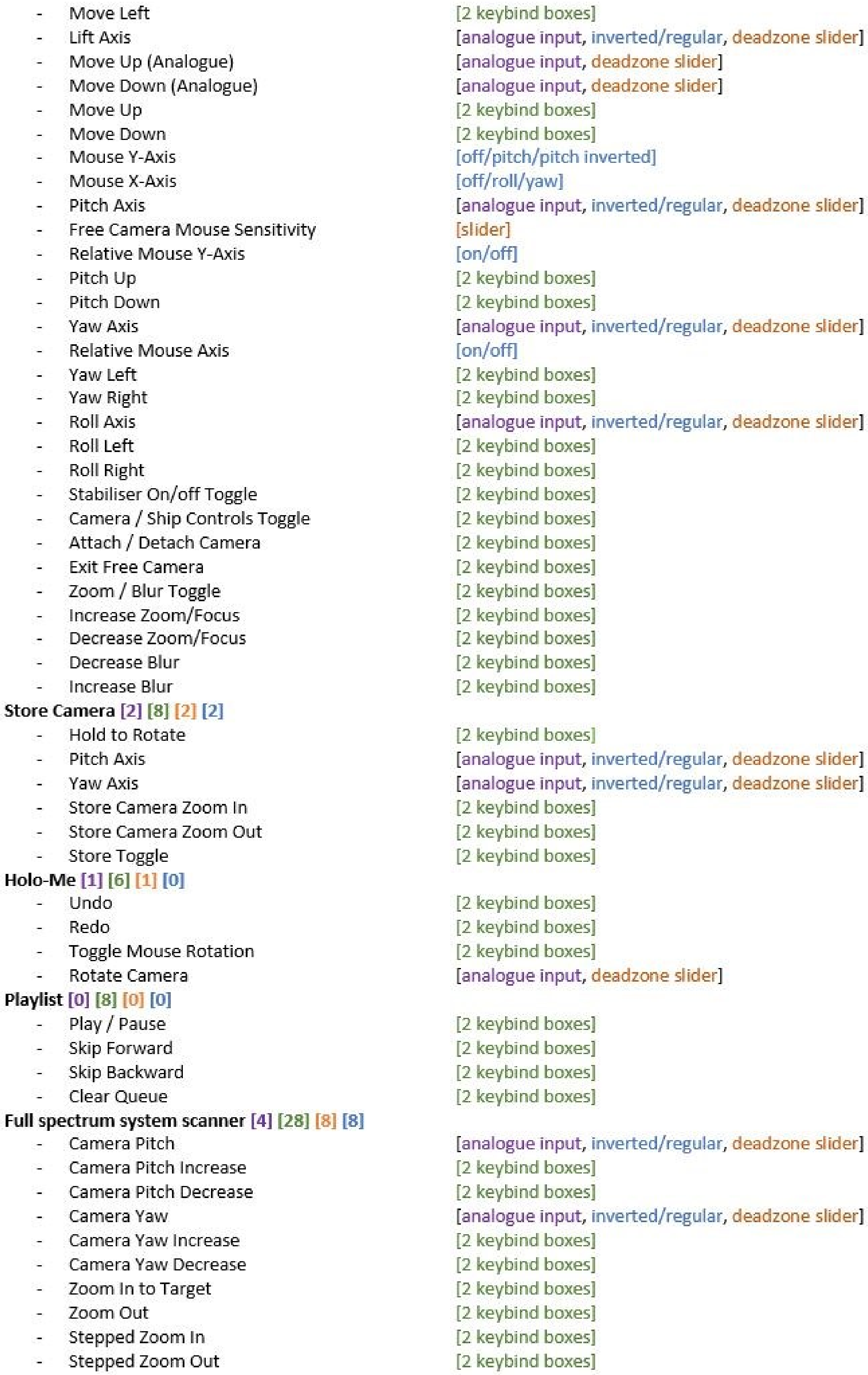
-	Recall Fighter	[2 keybind boxes]
-	Defend	[2 keybind boxes]
-	Engage at Will	[2 keybind boxes]
-	Attack Target	[2 keybind boxes]
-	Maintain Formation	[2 keybind boxes]
-	Hold Position	[2 keybind boxes]
-	Follow Me	[2 keybind boxes]
-	Open Orders	[2 keybind boxes]

Camera Suite [0] [28] [0] [0]

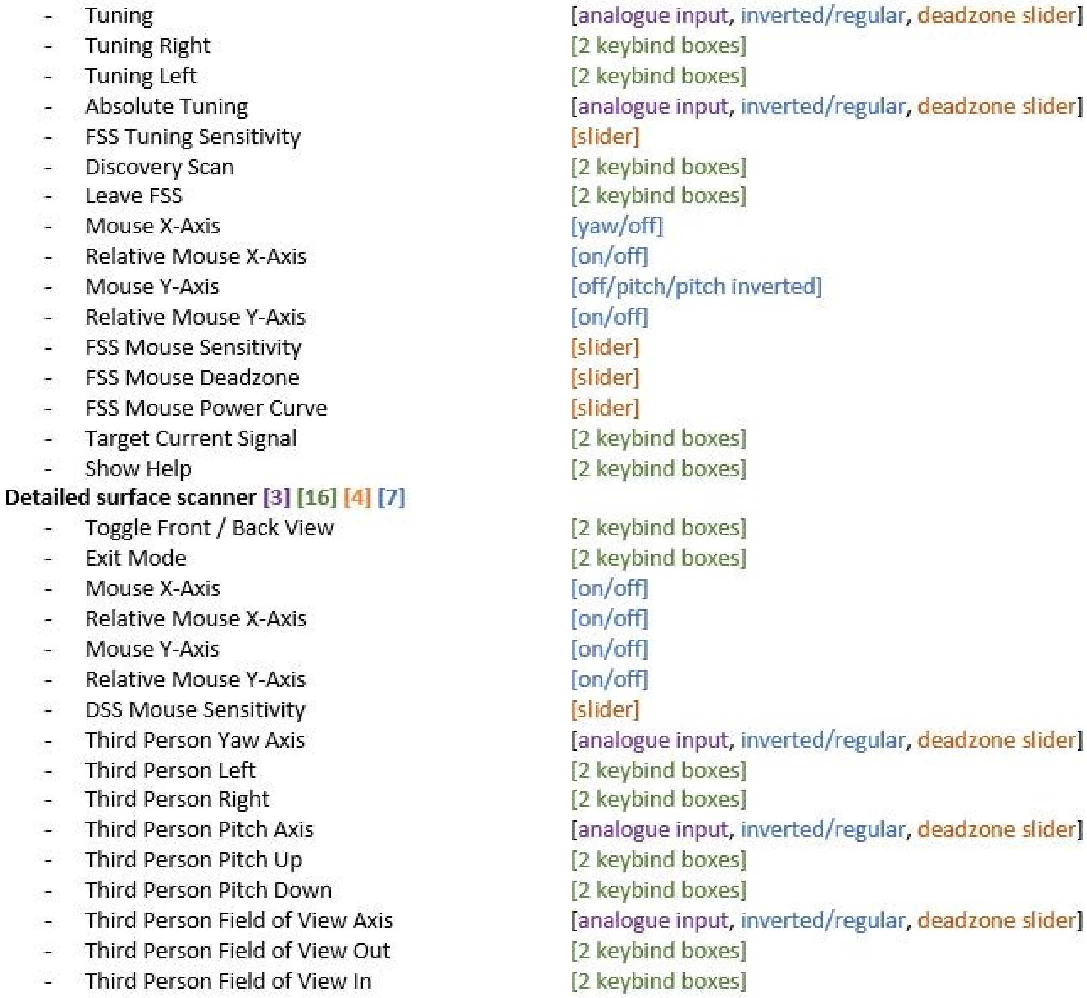
-	Ship – Toggle Camera Suite	[2 keybind boxes]
-	SRV – Toggle Camera Suite	[2 keybind boxes]
-	Previous Camera	[2 keybind boxes]
-	Next Camera	[2 keybind boxes]
-	Enter Free Camera	[2 keybind boxes]
-	Camera – Cockpit Front	[2 keybind boxes]
-	Camera – Cockpit Back	[2 keybind boxes]
-	Camera – CMDR 1	[2 keybind boxes]
-	Camera – CMDR 2	[2 keybind boxes]
-	Camera – Co-Pilot 1	[2 keybind boxes]
-	Camera – Co-Pilot 2	[2 keybind boxes]
-	Camera – Front	[2 keybind boxes]
-	Camera – Back	[2 keybind boxes]
-	Camera – Low	[2 keybind boxes]

Free Camera [8] [50] [9] [12]

-	Toggle HUD	[2 keybind boxes]
-	Increase Speed	[2 keybind boxes]
-	Decrease Speed	[2 keybind boxes]
-	Forward Axis	[analogue input, inverted/regular, deadzone slider]
-	Throttle Axis Range	[forward only/full range]
-	Forward Only Throttle Reverse	[2 keybind boxes, hold/toggle]
-	Move Forward	[2 keybind boxes]
-	Move Backward	[2 keybind boxes]
-	Lateral Axis	[analogue input, inverted/regular, deadzone slider]
-	Move Right	[2 keybind boxes]



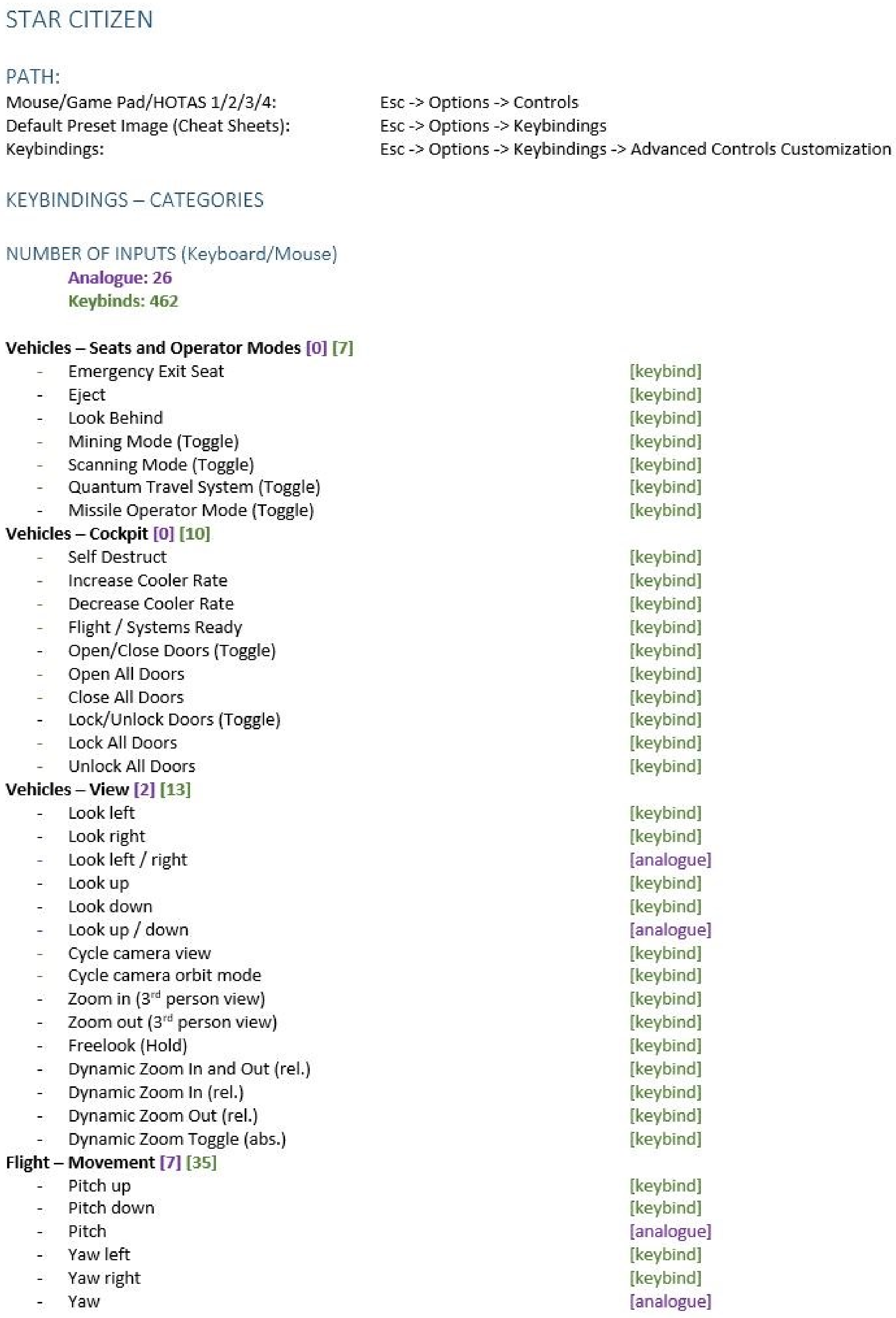
- Move Left	[2 keybind boxes]
- Lift Axis	[analogue input, inverted/regular, deadzone slider]
- Move Up (Analogue)	[analogue input, deadzone slider]
- Move Down (Analogue)	[analogue input, deadzone slider]
- Move Up	[2 keybind boxes]
- Move Down	[2 keybind boxes]
- Mouse Y-Axis	[off/pitch/pitch inverted]
- Mouse X-Axis	[off/roll/yaw]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Free Camera Mouse Sensitivity	[slider]
- Relative Mouse Y-Axis	[on/off]
- Pitch Up	[2 keybind boxes]
- Pitch Down	[2 keybind boxes]
- Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Relative Mouse Axis	[on/off]
- Yaw Left	[2 keybind boxes]
- Yaw Right	[2 keybind boxes]
- Roll Axis	[analogue input, inverted/regular, deadzone slider]
- Roll Left	[2 keybind boxes]
- Roll Right	[2 keybind boxes]
- Stabiliser On/off Toggle	[2 keybind boxes]
- Camera / Ship Controls Toggle	[2 keybind boxes]
- Attach / Detach Camera	[2 keybind boxes]
- Exit Free Camera	[2 keybind boxes]
- Zoom / Blur Toggle	[2 keybind boxes]
- Increase Zoom/Focus	[2 keybind boxes]
- Decrease Zoom/Focus	[2 keybind boxes]
- Decrease Blur	[2 keybind boxes]
- Increase Blur	[2 keybind boxes]
Store Camera [2] [8] [2] [2]	
- Hold to Rotate	[2 keybind boxes]
- Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Store Camera Zoom In	[2 keybind boxes]
- Store Camera Zoom Out	[2 keybind boxes]
- Store Toggle	[2 keybind boxes]
Holo-Me [1] [6] [1] [0]	
- Undo	[2 keybind boxes]
- Redo	[2 keybind boxes]
- Toggle Mouse Rotation	[2 keybind boxes]
- Rotate Camera	[analogue input, deadzone slider]
Playlist [0] [8] [0] [0]	
- Play / Pause	[2 keybind boxes]
- Skip Forward	[2 keybind boxes]
- Skip Backward	[2 keybind boxes]
- Clear Queue	[2 keybind boxes]
Full spectrum system scanner [4] [28] [8] [8]	
- Camera Pitch	[analogue input, inverted/regular, deadzone slider]
- Camera Pitch Increase	[2 keybind boxes]
- Camera Pitch Decrease	[2 keybind boxes]
- Camera Yaw	[analogue input, inverted/regular, deadzone slider]
- Camera Yaw Increase	[2 keybind boxes]
- Camera Yaw Decrease	[2 keybind boxes]
- Zoom In to Target	[2 keybind boxes]
- Zoom Out	[2 keybind boxes]
- Stepped Zoom In	[2 keybind boxes]
- Stepped Zoom Out	[2 keybind boxes]



- Tuning	[analogue input, inverted/regular, deadzone slider]
- Tuning Right	[2 keybind boxes]
- Tuning Left	[2 keybind boxes]
- Absolute Tuning	[analogue input, inverted/regular, deadzone slider]
- FSS Tuning Sensitivity	[slider]
- Discovery Scan	[2 keybind boxes]
- Leave FSS	[2 keybind boxes]
- Mouse X-Axis	[yaw/off]
- Relative Mouse X-Axis	[on/off]
- Mouse Y-Axis	[off/pitch/pitch inverted]
- Relative Mouse Y-Axis	[on/off]
- FSS Mouse Sensitivity	[slider]
- FSS Mouse Deadzone	[slider]
- FSS Mouse Power Curve	[slider]
- Target Current Signal	[2 keybind boxes]
- Show Help	[2 keybind boxes]
Detailed surface scanner [3] [16] [4] [7]	
- Toggle Front / Back View	[2 keybind boxes]
- Exit Mode	[2 keybind boxes]
- Mouse X-Axis	[on/off]
- Relative Mouse X-Axis	[on/off]
- Mouse Y-Axis	[on/off]
- Relative Mouse Y-Axis	[on/off]
- DSS Mouse Sensitivity	[slider]
- Third Person Yaw Axis	[analogue input, inverted/regular, deadzone slider]
- Third Person Left	[2 keybind boxes]
- Third Person Right	[2 keybind boxes]
- Third Person Pitch Axis	[analogue input, inverted/regular, deadzone slider]
- Third Person Pitch Up	[2 keybind boxes]
- Third Person Pitch Down	[2 keybind boxes]
- Third Person Field of View Axis	[analogue input, inverted/regular, deadzone slider]
- Third Person Field of View Out	[2 keybind boxes]
- Third Person Field of View In	[2 keybind boxes]

Appendix 2/1

STAR CITIZEN INTERFACE BREAKDOWN



STAR CITIZEN

PATH:

Mouse/Game Pad/HOTAS 1/2/3/4:	Esc -> Options -> Controls
Default Preset Image (Cheat Sheets):	Esc -> Options -> Keybindings
Keybindings:	Esc -> Options -> Keybindings -> Advanced Controls Customization

KEYBINDINGS – CATEGORIES

NUMBER OF INPUTS (Keyboard/Mouse)

Analogue: 26
Keybinds: 462

Vehicles – Seats and Operator Modes [0] [7]

- Emergency Exit Seat	[keybind]
- Eject	[keybind]
- Look Behind	[keybind]
- Mining Mode (Toggle)	[keybind]
- Scanning Mode (Toggle)	[keybind]
- Quantum Travel System (Toggle)	[keybind]
- Missile Operator Mode (Toggle)	[keybind]

Vehicles – Cockpit [0] [10]

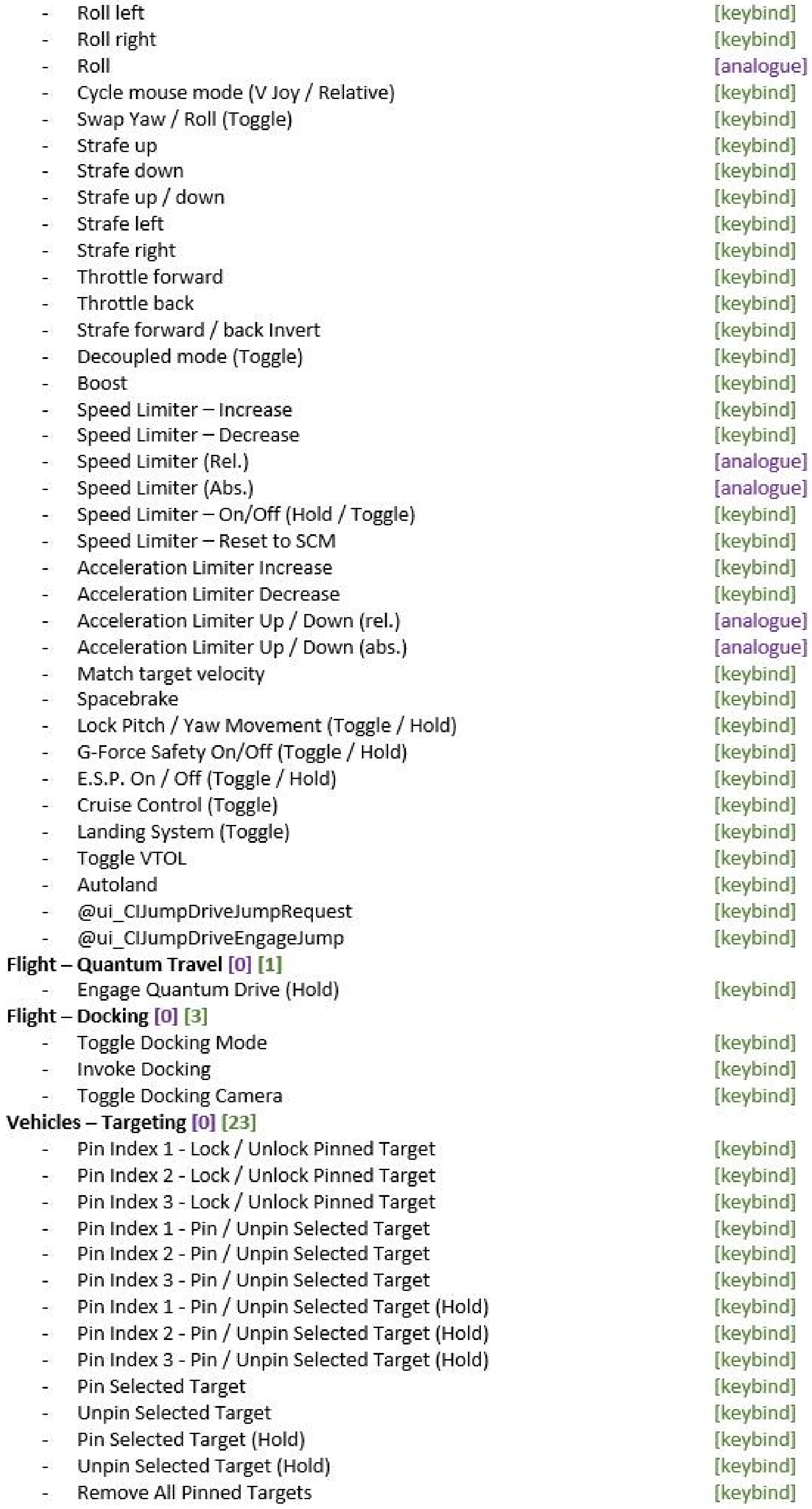
- Self Destruct	[keybind]
- Increase Cooler Rate	[keybind]
- Decrease Cooler Rate	[keybind]
- Flight / Systems Ready	[keybind]
- Open/Close Doors (Toggle)	[keybind]
- Open All Doors	[keybind]
- Close All Doors	[keybind]
- Lock/Unlock Doors (Toggle)	[keybind]
- Lock All Doors	[keybind]
- Unlock All Doors	[keybind]

Vehicles – View [2] [13]

- Look left	[keybind]
- Look right	[keybind]
- Look left / right	[analogue]
- Look up	[keybind]
- Look down	[keybind]
- Look up / down	[analogue]
- Cycle camera view	[keybind]
- Cycle camera orbit mode	[keybind]
- Zoom in (3rd person view)	[keybind]
- Zoom out (3rd person view)	[keybind]
- Freelook (Hold)	[keybind]
- Dynamic Zoom In and Out (rel.)	[keybind]
- Dynamic Zoom In (rel.)	[keybind]
- Dynamic Zoom Out (rel.)	[keybind]
- Dynamic Zoom Toggle (abs.)	[keybind]

Flight – Movement [7] [35]

- Pitch up	[keybind]
- Pitch down	[keybind]
- Pitch	[analogue]
- Yaw left	[keybind]
- Yaw right	[keybind]
- Yaw	[analogue]



- Roll left	[keybind]
- Roll right	[keybind]
- Roll	[analogue]
- Cycle mouse mode (V Joy / Relative)	[keybind]
- Swap Yaw / Roll (Toggle)	[keybind]
- Strafe up	[keybind]
- Strafe down	[keybind]
- Strafe up / down	[keybind]
- Strafe left	[keybind]
- Strafe right	[keybind]
- Throttle forward	[keybind]
- Throttle back	[keybind]
- Strafe forward / back Invert	[keybind]
- Decoupled mode (Toggle)	[keybind]
- Boost	[keybind]
- Speed Limiter – Increase	[keybind]
- Speed Limiter – Decrease	[keybind]
- Speed Limiter (Rel.)	[analogue]
- Speed Limiter (Abs.)	[analogue]
- Speed Limiter – On/Off (Hold / Toggle)	[keybind]
- Speed Limiter – Reset to SCM	[keybind]
- Acceleration Limiter Increase	[keybind]
- Acceleration Limiter Decrease	[keybind]
- Acceleration Limiter Up / Down (rel.)	[analogue]
- Acceleration Limiter Up / Down (abs.)	[analogue]
- Match target velocity	[keybind]
- Spacebrake	[keybind]
- Lock Pitch / Yaw Movement (Toggle / Hold)	[keybind]
- G-Force Safety On/Off (Toggle / Hold)	[keybind]
- E.S.P. On / Off (Toggle / Hold)	[keybind]
- Cruise Control (Toggle)	[keybind]
- Landing System (Toggle)	[keybind]
- Toggle VTOL	[keybind]
- Autoland	[keybind]
- @ui_CIJumpDriveJumpRequest	[keybind]
- @ui_CIJumpDriveEngageJump	[keybind]
Flight – Quantum Travel [0] [1]	
- Engage Quantum Drive (Hold)	[keybind]
Flight – Docking [0] [3]	
- Toggle Docking Mode	[keybind]
- Invoke Docking	[keybind]
- Toggle Docking Camera	[keybind]
Vehicles – Targeting [0] [23]	
- Pin Index 1 - Lock / Unlock Pinned Target	[keybind]
- Pin Index 2 - Lock / Unlock Pinned Target	[keybind]
- Pin Index 3 - Lock / Unlock Pinned Target	[keybind]
- Pin Index 1 - Pin / Unpin Selected Target	[keybind]
- Pin Index 2 - Pin / Unpin Selected Target	[keybind]
- Pin Index 3 - Pin / Unpin Selected Target	[keybind]
- Pin Index 1 - Pin / Unpin Selected Target (Hold)	[keybind]
- Pin Index 2 - Pin / Unpin Selected Target (Hold)	[keybind]
- Pin Index 3 - Pin / Unpin Selected Target (Hold)	[keybind]
- Pin Selected Target	[keybind]
- Unpin Selected Target	[keybind]
- Pin Selected Target (Hold)	[keybind]
- Unpin Selected Target (Hold)	[keybind]
- Remove All Pinned Targets	[keybind]

- Lock Selected Target [keybind]
- Unlock Selected Target [keybind]
- Enable/ Disable Look Ahead [keybind]
- Enable / Disable Target Padlock (Toggle, Hold) [keybind]
- Auto Zoom on Selected Target On / Off (Toggle, Hold) [keybind]
- Switch Flight Lead / Lag Reticle (Toggle) [keybind]
- Enter Remote Turret [keybind]
- Enter Remote Turret [keybind]
- Enter Remote Turret [keybind]

Vehicles – Target Cycling [0] [24]

- Cycle Selection – Back [keybind]
- Cycle Selection – Forward [keybind]
- Cycle Selection – Reset to Auto [keybind]
- Cycle Lock – In View – Back [keybind]
- Cycle Lock – In View – Forward [keybind]
- Cycle Lock – In Vier – Under Reticle [keybind]
- Cycle Lock – Pinned – Back [keybind]
- Cycle Lock – Pinned – Forward [keybind]
- Cycle Lock – Pinned – Reset to First [keybind]
- Cycle Lock – Attackers – Back [keybind]
- Cycle Lock – Attackers – Forward [keybind]
- Cycle Lock – Attackers – Reset to Closest [keybind]
- Cycle Lock – Hostiles – Back [keybind]
- Cycle Lock – Hostiles – Forward [keybind]
- Cycle Lock – Hostiles – Reset to Closest [keybind]
- Cycle Lock – Friendlies – Back [keybind]
- Cycle Lock – Friendlies – Forward [keybind]
- Cycle Lock – Friendlies – Reset to Closest [keybind]
- Cycle Lock – All – Back [keybind]
- Cycle Lock – All Forward [keybind]
- Cycle Lock – All – Reset to Closest [keybind]
- Cycle Lock – Sub-Target – Back [keybind]
- Cycle Lock – Sub-Target – Forward [keybind]
- Cycle Lock – Sub-Target – Reset to Main Target [keybind]

Flight – Target Hailing [0] [1]

- Hail Target [keybind]

Flight – Radar [0] [3]

- Activate Ping [keybind]
- Increase Ping Angle [keybind]
- Decrease Ping Angle [keybind]

Vehicles – Scanning [0] [3]

- Activate Scanning [keybind]
- Increase Scanning Angle [keybind]
- Decrease Scanning Angle [keybind]

Vehicles – Mining [0] [8]

- Fire Mining Laser (Toggle) [keybind]
- Switch Mining Laser (Toggle) [keybind]
- Increase Mining Laser Power [keybind]
- Decrease Mining Laser Power [keybind]
- Activate Mining Module (Slot 1) [keybind]
- Activate Mining Module (Slot 2) [keybind]
- Activate Mining Module (Slot 3) [keybind]
- Jettison Cargo [keybind]

Turret Movement [2] [6]

- Pitch up [keybind]
- Pitch down [keybind]
- Pitch [analogue]

- Yaw left [keybind]
- Yaw right [keybind]
- Yaw [analogue]
- Toggle Turret Mouse Movement (VJoy, FPS Style) [keybind]
- Exit Remote Turret [keybind]

Turret Advanced [0] [8]

- Toggle E.S.P. on /off [keybind]
- Recenter Turret (Hold) [keybind]
- Turret Gyro Stabilization (Toggle) [keybind]
- Turret – Speed Limiter – On/Off (Hold/Toggle) [keybind]
- Turret – Speed Limiter (rel) [keybind]
- [PH] Turret – Speed Limiter – Increase (rel) [keybind]
- [PH] Turret – Speed Limiter – Decrease (rel) [keybind]
- Cycle fire mode (staggered / combined) [keybind]

Vehicles – Weapons [0] [5]

- Fire Weapon Group 1 [keybind]
- Fire Weapon Group 2 [keybind]
- Manual Gimbal Mode – Swap VJoy / Look Direction (Toggle, Hold) [keybind]
- Manual Gimbal Mode – Lock Aim Vector [keybind]
- Cycle Gimbal Assist / Standard Gimbal / Gimbal Lock modes [keybind]

Vehicles – Missiles [0] [14]

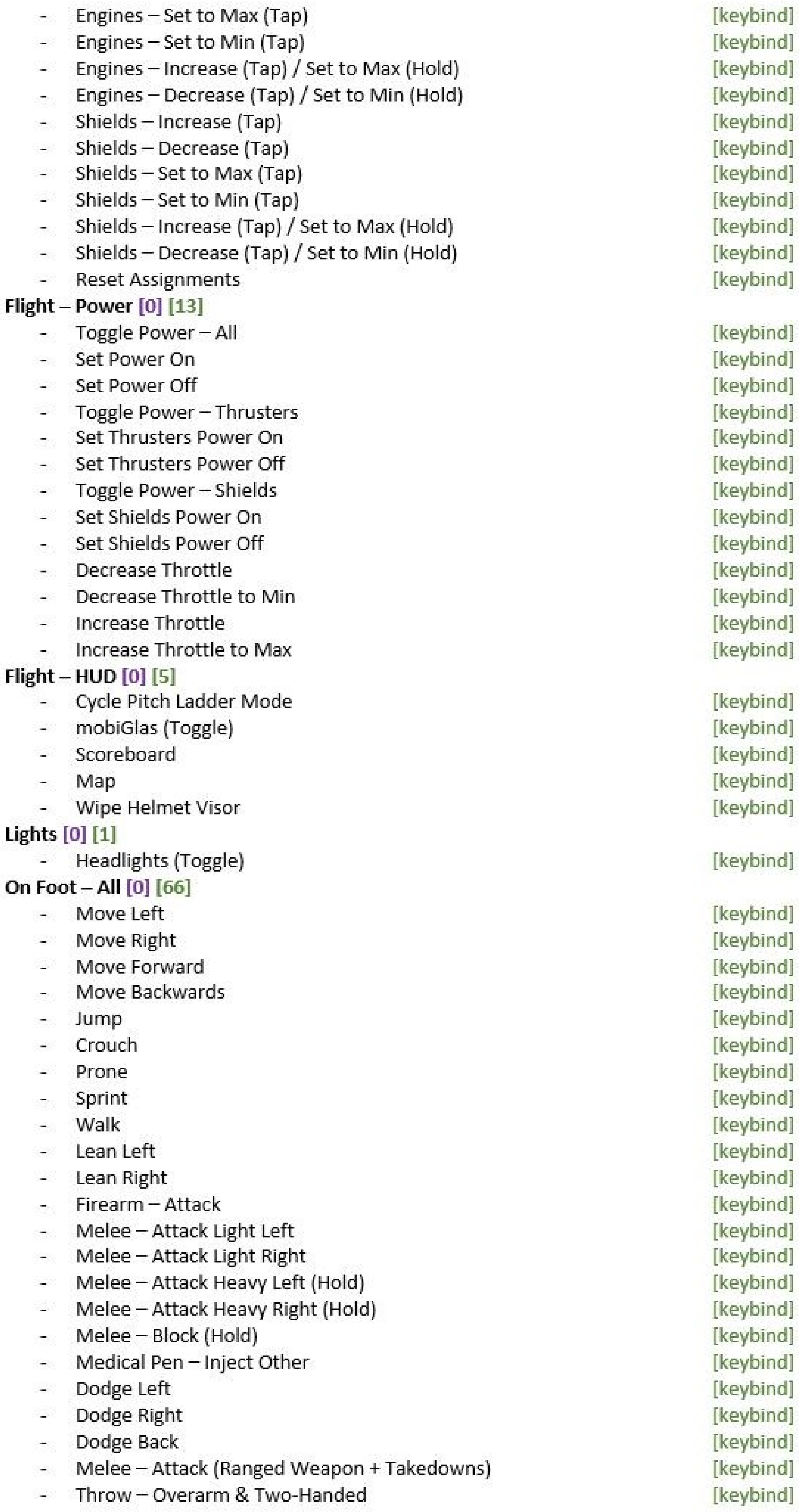
- Launch Missiles (Tap) [keybind]
- Launch Missiles (Hold) [keybind]
- Cycle Next Missile Type [keybind]
- Cycle Previous Missile Type [keybind]
- Increase Number of Armed Missiles [keybind]
- Decrease Number of Armed Missiles [keybind]
- Reset Number of Armed Missiles [keybind]
- [PH] Bombs – Toggle Desired Impact Point (Tap) [keybind]
- [PH] Bombs – Toggle Desired Impact Point (Hold) [keybind]
- Bombs – Increase HUD Range [keybind]
- Bombs – Decrease HUD Range [keybind]
- Bombs – Reset HUD Range [keybind]
- [PH] – Enable Cinematic Camera (Toggle) [keybind]
- [PH] – Enable Cinematic Camera (Hold) [keybind]

Vehicles – Shields and Countermeasures [0] [12]

- Decoy – Launch Burst (tap), Set and Launch Burst (hold) [keybind]
- Decoy – Increase Burst Size (tap) [keybind]
- Decoy – Decrease Burst Size (tap) [keybind]
- Decoy – Panic Launch (tap) [keybind]
- Noise – Deploy (Tap) [keybind]
- Shield raise level front [keybind]
- Shield raise level back [keybind]
- Shield raise level left [keybind]
- Shield raise level right [keybind]
- Shield raise level top [keybind]
- Shield raise level bottom [keybind]
- Shield rest levels [keybind]

Vehicles – Power Triangle Assignment [0] [19]

- Weapons – Increase (Tap) [keybind]
- Weapons – Decrease (Tap) [keybind]
- Weapons – Set to Max (Tap) [keybind]
- Weapons – Set to Min (Tap) [keybind]
- Weapons – Increase (Tap) / Set to Max (Hold) [keybind]
- Weapons – Decrease (Tap) / Set to Min (Hold) [keybind]
- Engines – Increase (Tap) [keybind]
- Engines – Decrease (Tap) [keybind]



- Engines – Set to Max (Tap) [keybind]
- Engines – Set to Min (Tap) [keybind]
- Engines – Increase (Tap) / Set to Max (Hold) [keybind]
- Engines – Decrease (Tap) / Set to Min (Hold) [keybind]
- Shields – Increase (Tap) [keybind]
- Shields – Decrease (Tap) [keybind]
- Shields – Set to Max (Tap) [keybind]
- Shields – Set to Min (Tap) [keybind]
- Shields – Increase (Tap) / Set to Max (Hold) [keybind]
- Shields – Decrease (Tap) / Set to Min (Hold) [keybind]
- Reset Assignments [keybind]

Flight – Power [0] [13]

- Toggle Power – All [keybind]
- Set Power On [keybind]
- Set Power Off [keybind]
- Toggle Power – Thrusters [keybind]
- Set Thrusters Power On [keybind]
- Set Thrusters Power Off [keybind]
- Toggle Power – Shields [keybind]
- Set Shields Power On [keybind]
- Set Shields Power Off [keybind]
- Decrease Throttle [keybind]
- Decrease Throttle to Min [keybind]
- Increase Throttle [keybind]
- Increase Throttle to Max [keybind]

Flight – HUD [0] [5]

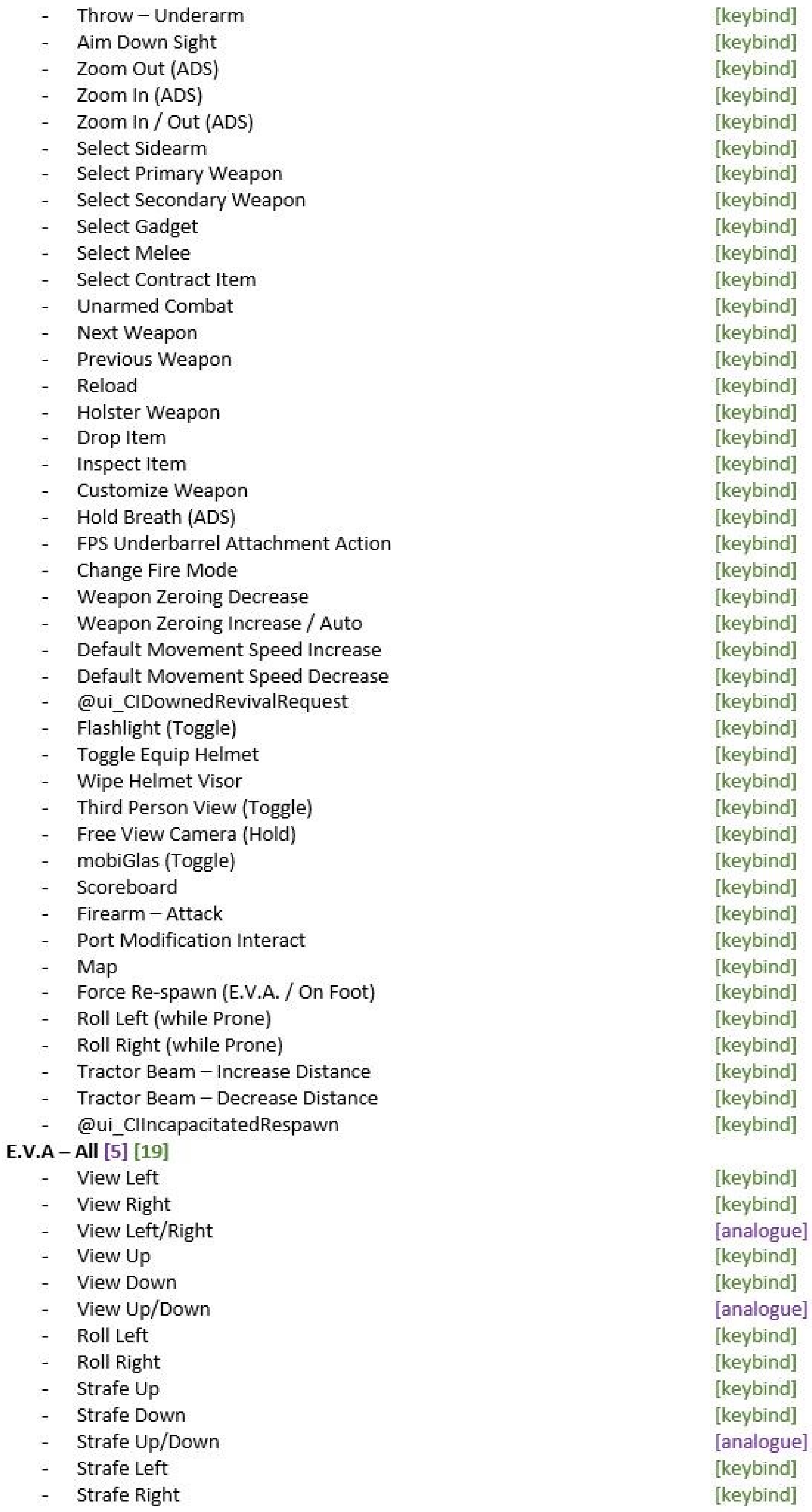
- Cycle Pitch Ladder Mode [keybind]
- mobiGlas (Toggle) [keybind]
- Scoreboard [keybind]
- Map [keybind]
- Wipe Helmet Visor [keybind]

Lights [0] [1]

- Headlights (Toggle) [keybind]

On Foot – All [0] [66]

- Move Left [keybind]
- Move Right [keybind]
- Move Forward [keybind]
- Move Backwards [keybind]
- Jump [keybind]
- Crouch [keybind]
- Prone [keybind]
- Sprint [keybind]
- Walk [keybind]
- Lean Left [keybind]
- Lean Right [keybind]
- Firearm – Attack [keybind]
- Melee – Attack Light Left [keybind]
- Melee – Attack Light Right [keybind]
- Melee – Attack Heavy Left (Hold) [keybind]
- Melee – Attack Heavy Right (Hold) [keybind]
- Melee – Block (Hold) [keybind]
- Medical Pen – Inject Other [keybind]
- Dodge Left [keybind]
- Dodge Right [keybind]
- Dodge Back [keybind]
- Melee – Attack (Ranged Weapon + Takedowns) [keybind]
- Throw – Overarm & Two-Handed [keybind]



- Throw – Underarm	[keybind]
- Aim Down Sight	[keybind]
- Zoom Out (ADS)	[keybind]
- Zoom In (ADS)	[keybind]
- Zoom In / Out (ADS)	[keybind]
- Select Sidearm	[keybind]
- Select Primary Weapon	[keybind]
- Select Secondary Weapon	[keybind]
- Select Gadget	[keybind]
- Select Melee	[keybind]
- Select Contract Item	[keybind]
- Unarmed Combat	[keybind]
- Next Weapon	[keybind]
- Previous Weapon	[keybind]
- Reload	[keybind]
- Holster Weapon	[keybind]
- Drop Item	[keybind]
- Inspect Item	[keybind]
- Customize Weapon	[keybind]
- Hold Breath (ADS)	[keybind]
- FPS Underbarrel Attachment Action	[keybind]
- Change Fire Mode	[keybind]
- Weapon Zeroing Decrease	[keybind]
- Weapon Zeroing Increase / Auto	[keybind]
- Default Movement Speed Increase	[keybind]
- Default Movement Speed Decrease	[keybind]
- @ui_CIDownedRevivalRequest	[keybind]
- Flashlight (Toggle)	[keybind]
- Toggle Equip Helmet	[keybind]
- Wipe Helmet Visor	[keybind]
- Third Person View (Toggle)	[keybind]
- Free View Camera (Hold)	[keybind]
- mobiGlas (Toggle)	[keybind]
- Scoreboard	[keybind]
- Firearm – Attack	[keybind]
- Port Modification Interact	[keybind]
- Map	[keybind]
- Force Re-spawn (E.V.A. / On Foot)	[keybind]
- Roll Left (while Prone)	[keybind]
- Roll Right (while Prone)	[keybind]
- Tractor Beam – Increase Distance	[keybind]
- Tractor Beam – Decrease Distance	[keybind]
- @ui_CIIncapacitatedRespawn	[keybind]

E.V.A – All [5] [19]

- View Left	[keybind]
- View Right	[keybind]
- View Left/Right	[analogue]
- View Up	[keybind]
- View Down	[keybind]
- View Up/Down	[analogue]
- Roll Left	[keybind]
- Roll Right	[keybind]
- Strafe Up	[keybind]
- Strafe Down	[keybind]
- Strafe Up/Down	[analogue]
- Strafe Left	[keybind]
- Strafe Right	[keybind]

Strafe Left/Right	[analogue]
Strafe Forward	[keybind]
Strafe Backward	[keybind]
Strafe Forward/Backward	[analogue]
Brake	[keybind]
Boost	[keybind]
@ui_CIEVAPushForward	[keybind]
@ui_CIEVAPuchBack	[keybind]
@ui_CIEVALaunch	[keybind]
@ui_CIEVADetach	[keybind]
Freelook (Hold)	[keybind]

Ground Vehicle – General [2] [17]

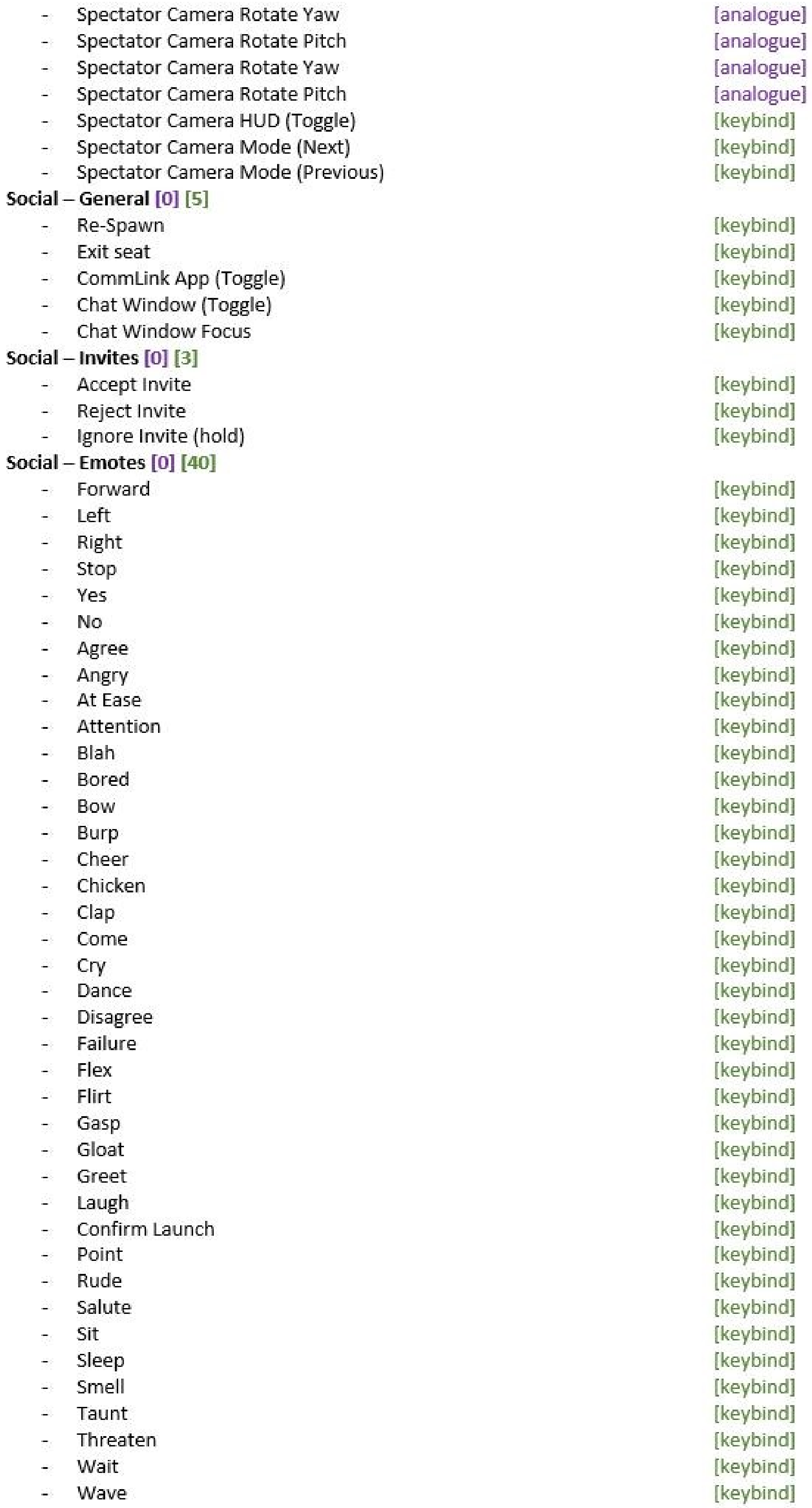
Horn	[keybind]
Cycle camera view	[keybind]
Zoom in (3rd person view)	[keybind]
Zoom out (3rd person view)	[keybind]
Look left / right	[analogue]
Look up /down	[analogue]
Freelook (Hold)	[keybind]
Fire Weapon Group 1	[keybind]
Fire Weapon Group 2	[keybind]
mobiGlas (Toggle)	[keybind]
Flight / Systems Ready	[keybind]
Open/Close Doors (Toggle)	[keybind]
Open All Doors	[keybind]
Close All Doors	[keybind]
Lock/Unlock Doors (Toggle)	[keybind]
Lock All Doors	[keybind]
Unlock All Doors	[keybind]
Map	[keybind]
Wipe Helmet Visor	[keybind]

Ground Vehicle – Movement [4] [14]

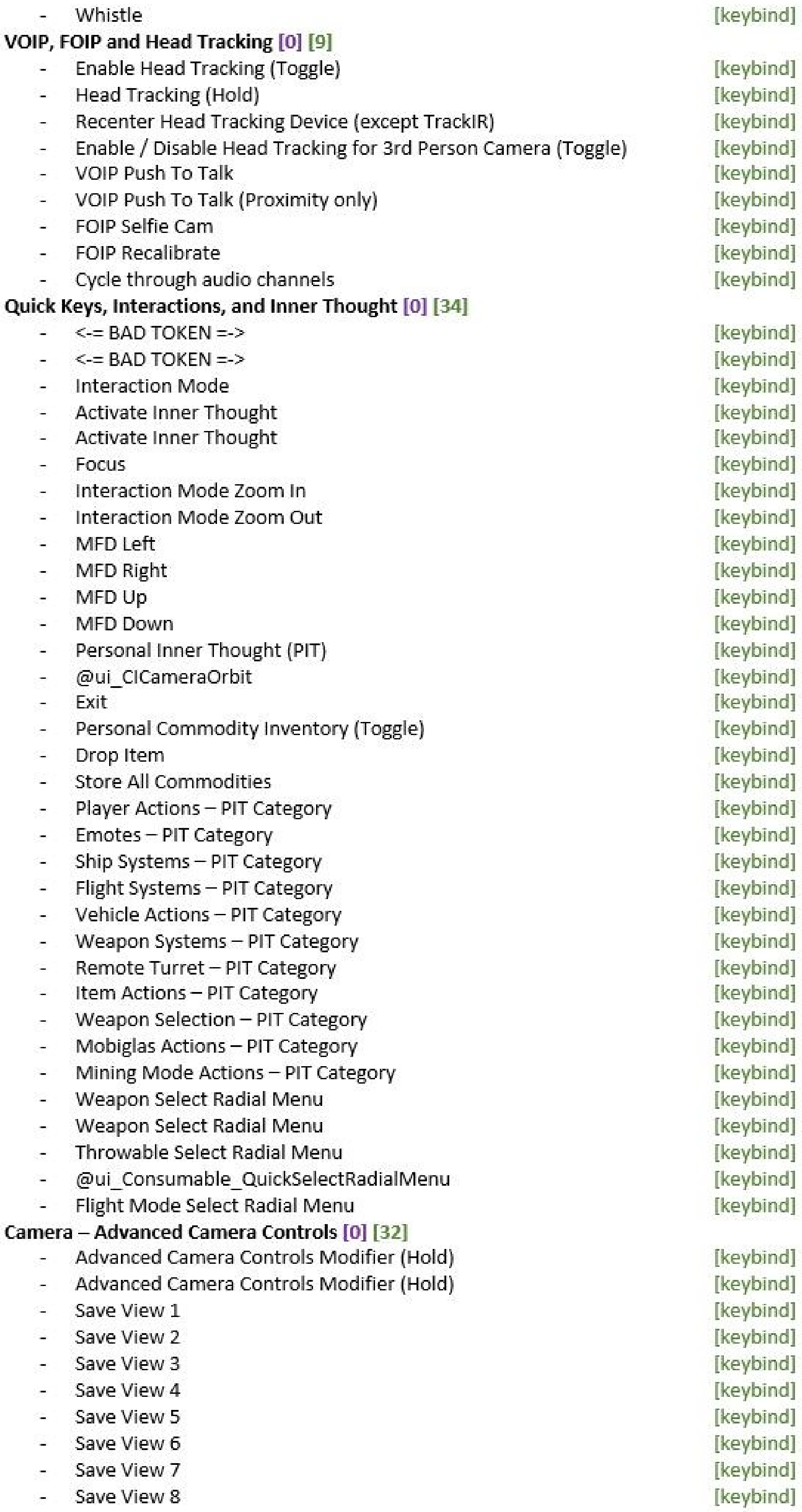
Drive Forward	[keybind]
Drive Backward	[keybind]
Turn Left	[keybind]
Turn Right	[keybind]
Yaw Left / Right (Mouse)	[analogue]
@ui_CI_MGV_Pitch_Up	[keybind]
@ui_CI_MGV_Pitch_Down	[keybind]
Pitch Up / Down (Mouse)	[analogue]
Brake	[keybind]
Primary Fire	[keybind]
Secondary Fire	[keybind]
Dynamic Zoom In and Out (rel.)	[analogue]
Dynamic Zoom In (rel.)	[keybind]
Dynamic Zoom Out (rel.)	[keybind]
Dynamic Zoom In and Out (abs.)	[analogue]
Boost	[keybind]
Lock Pitch / Yaw Movement (Toggle / Hold)	[keybind]
@ui_CI_MGV_ToggleBreakOnIdle	[keybind]

Electronic Access – Spectator [4] [9]

Spectator Camera Target (Next)	[keybind]
Spectator Camera Target (Previous)	[keybind]
Spectator Camera Lock Target	[keybind]
Spectator Camera Zoom	[keybind]
Spectator Camera Zoom In	[keybind]
Spectator Camera Zoom Out	[keybind]



- Spectator Camera Rotate Yaw	[analogue]
- Spectator Camera Rotate Pitch	[analogue]
- Spectator Camera Rotate Yaw	[analogue]
- Spectator Camera Rotate Pitch	[analogue]
- Spectator Camera HUD (Toggle)	[keybind]
- Spectator Camera Mode (Next)	[keybind]
- Spectator Camera Mode (Previous)	[keybind]
Social – General [0] [5]	
- Re-Spawn	[keybind]
- Exit seat	[keybind]
- CommLink App (Toggle)	[keybind]
- Chat Window (Toggle)	[keybind]
- Chat Window Focus	[keybind]
Social – Invites [0] [3]	
- Accept Invite	[keybind]
- Reject Invite	[keybind]
- Ignore Invite (hold)	[keybind]
Social – Emotes [0] [40]	
- Forward	[keybind]
- Left	[keybind]
- Right	[keybind]
- Stop	[keybind]
- Yes	[keybind]
- No	[keybind]
- Agree	[keybind]
- Angry	[keybind]
- At Ease	[keybind]
- Attention	[keybind]
- Blah	[keybind]
- Bored	[keybind]
- Bow	[keybind]
- Burp	[keybind]
- Cheer	[keybind]
- Chicken	[keybind]
- Clap	[keybind]
- Come	[keybind]
- Cry	[keybind]
- Dance	[keybind]
- Disagree	[keybind]
- Failure	[keybind]
- Flex	[keybind]
- Flirt	[keybind]
- Gasp	[keybind]
- Gloat	[keybind]
- Greet	[keybind]
- Laugh	[keybind]
- Confirm Launch	[keybind]
- Point	[keybind]
- Rude	[keybind]
- Salute	[keybind]
- Sit	[keybind]
- Sleep	[keybind]
- Smell	[keybind]
- Taunt	[keybind]
- Threaten	[keybind]
- Wait	[keybind]
- Wave	[keybind]



- Whistle [keybind]

VOIP, FOIP and Head Tracking [0] [9]

- Enable Head Tracking (Toggle) [keybind]
- Head Tracking (Hold) [keybind]
- Recenter Head Tracking Device (except TrackIR) [keybind]
- Enable / Disable Head Tracking for 3rd Person Camera (Toggle) [keybind]
- VOIP Push To Talk [keybind]
- VOIP Push To Talk (Proximity only) [keybind]
- FOIP Selfie Cam [keybind]
- FOIP Recalibrate [keybind]
- Cycle through audio channels [keybind]

Quick Keys, Interactions, and Inner Thought [0] [34]

- <-= BAD TOKEN =-> [keybind]
- <-= BAD TOKEN =-> [keybind]
- Interaction Mode [keybind]
- Activate Inner Thought [keybind]
- Activate Inner Thought [keybind]
- Focus [keybind]
- Interaction Mode Zoom In [keybind]
- Interaction Mode Zoom Out [keybind]
- MFD Left [keybind]
- MFD Right [keybind]
- MFD Up [keybind]
- MFD Down [keybind]
- Personal Inner Thought (PIT) [keybind]
- @ui_CICameraOrbit [keybind]
- Exit [keybind]
- Personal Commodity Inventory (Toggle) [keybind]
- Drop Item [keybind]
- Store All Commodities [keybind]
- Player Actions – PIT Category [keybind]
- Emotes – PIT Category [keybind]
- Ship Systems – PIT Category [keybind]
- Flight Systems – PIT Category [keybind]
- Vehicle Actions – PIT Category [keybind]
- Weapon Systems – PIT Category [keybind]
- Remote Turret – PIT Category [keybind]
- Item Actions – PIT Category [keybind]
- Weapon Selection – PIT Category [keybind]
- Mobiglas Actions – PIT Category [keybind]
- Mining Mode Actions – PIT Category [keybind]
- Weapon Select Radial Menu [keybind]
- Weapon Select Radial Menu [keybind]
- Throwable Select Radial Menu [keybind]
- @ui_Consumable_QuickSelectRadialMenu [keybind]
- Flight Mode Select Radial Menu [keybind]

Camera – Advanced Camera Controls [0] [32]

- Advanced Camera Controls Modifier (Hold) [keybind]
- Advanced Camera Controls Modifier (Hold) [keybind]
- Save View 1 [keybind]
- Save View 2 [keybind]
- Save View 3 [keybind]
- Save View 4 [keybind]
- Save View 5 [keybind]
- Save View 6 [keybind]
- Save View 7 [keybind]
- Save View 8 [keybind]

- Save View 9 [keybind]
- Load View 1 [keybind]
- Load View 2 [keybind]
- Load View 3 [keybind]
- Load View 4 [keybind]
- Load View 5 [keybind]
- Load View 6 [keybind]
- Load View 7 [keybind]
- Load View 8 [keybind]
- Load View 9 [keybind]
- Clear Saved View [keybind]
- X Offset Positive [keybind]
- X Offset Negative [keybind]
- Y Offset Positive / Spectator Freecam Focal Point Forward [keybind]
- Y Offset Negative / Spectator Freecam Focal Point Backward [keybind]
- Z Offset Positive [keybind]
- Z Offset Negative [keybind]
- Increase FoV [keybind]
- Decrease FoV [keybind]
- [PH] Increase DoF [keybind]
- Decrease DoF [keybind]
- Reset Current View [keybind]

CONTROLS - CATEGORIES

Excluded Controls categories:

- GamePad
- Joystick/HOTAS 1, 2, 3, 4

NUMBER OF INPUTS (Mouse)

Sliders: 42

Toggle buttons: 37

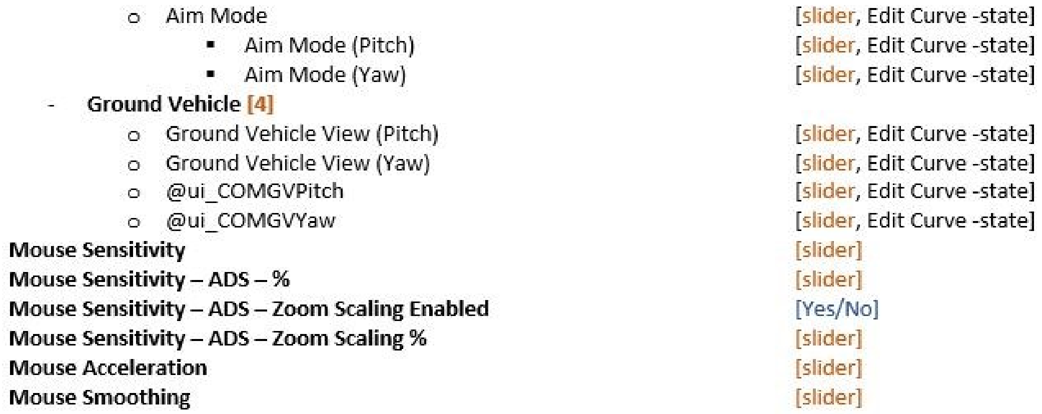
Inversion Settings

- **On Foot** **[2]**
  - On Foot (Pitch) [Yes/No]
  - On Foot (Yaw) [Yes/No]
- **Flight** **[6]**
  - Flight Movement
    - Flight (Pitch [Yes/No]
    - Flight (Yaw) [Yes/No]
    - Velocity Limiter (rel) [Yes/No]
    - Acceleration Limiter (rel) [Yes/No]
  - Free Look Mode
    - Flight View (Pitch) [Yes/No]
    - Flight View (Yaw) [Yes/No]
- **Turret** **[13]**
  - Turret Aim Pitch [Yes/No]
  - Turret Aim Yaw [Yes/No]
  - Turret – Vjoy Mode [Yes/No]
    - Turret – Vjoy – Pitch [Disable], [Yes/No]
    - Turret – Vjoy – Yaw [Disable], [Yes/No]
  - Turret Relative Mode [Yes/No]
    - Turret Relative Mode Pitch [Disable], [Yes/No]
    - Turret Relative Mode Yaw [Disable], [Yes/No]
  - Turret – Speed Limiter (rel) [Yes/No]

- **Any Vehicle** [11]
  - Virtual Joystick Mode
    - Virtual Joystick Mode (Pitch) [Yes/No]
    - Virtual Joystick Mode (Yaw) [Yes/No]
    - Virtual Joystick Mode (Roll) [Yes/No]
  - Virtual Joystick Mode (Gimbal Lock)
    - Virtual Joystick Mode (Gimbal Lock / Pitch) [Yes/No]
    - Virtual Joystick Mode (Gimbal Lock / Yaw) [Yes/No]
    - Virtual Joystick Mode (Gimbal Lock / Roll) [Yes/No]
  - Relative Mode
    - Relative Mode (Pitch) [Yes/No]
    - Relative Mode (Yaw) [Yes/No]
    - Relative Mode (Roll) [Yes/No]
  - Aim Mode
    - Aim Mode (Pitch) [Yes/No]
    - Aim Mode (Yaw) [Yes/No]
- **Ground Vehicle** [4]
  - Ground Vehicle View (Pitch) [Yes/No]
  - Ground Vehicle View (Yaw) [Yes/No]
  - @ui_COMGVPitch [Yes/No]
  - @ui_COMGVYaw [Yes/No]

Mouse Sensitivity Curves

- **On Foot** [3] [slider, Edit Curve -state]
  - On Foot (Pitch) [slider, Edit Curve -state]
  - On Foot (Yaw) [slider, Edit Curve -state]
- **Flight** [7]
  - Flight Movement
    - Flight (Pitch) [slider, Edit Curve -state]
    - Flight (Yaw) [slider, Edit Curve -state]
    - Velocity Limiter (rel) [slider, Edit Curve -state]
    - Acceleration Limiter (rel) [slider, Edit Curve -state]
  - Free Look Mode [slider, Edit Curve -state]
    - Flight View (Pitch) [slider, Edit Curve -state]
    - Flight View (Yaw) [slider, Edit Curve -state]
- **Turret** [8]
  - Turret Aim Pitch [slider, Edit Curve -state]
  - Turret Aim Yaw [slider, Edit Curve -state]
  - Turret – Vjoy Mode [slider, Edit Curve -state]
    - Turret – Vjoy – Pitch [slider, Edit Curve -state]
    - Turret – Vjoy – Yaw [slider, Edit Curve -state]
  - Turret Relative Mode [slider, Edit Curve -state]
    - Turret Relative Mode Pitch [slider, Edit Curve -state]
    - Turret Relative Mode Yaw [slider, Edit Curve -state]
- **Any Vehicle** [15]
  - Virtual Joystick Mode [slider, Edit Curve -state]
    - Virtual Joystick Mode (Pitch) [slider, Edit Curve -state]
    - Virtual Joystick Mode (Yaw) [slider, Edit Curve -state]
    - Virtual Joystick Mode (Roll) [slider, Edit Curve -state]
  - Virtual Joystick Mode (Gimbal Lock) [slider, Edit Curve -state]
    - Virtual Joystick Mode (Gimbal Lock / Pitch) [slider, Edit Curve -state]
    - Virtual Joystick Mode (Gimbal Lock / Yaw) [slider, Edit Curve -state]
    - Virtual Joystick Mode (Gimbal Lock / Roll) [slider, Edit Curve -state]
  - Relative Mode [slider, Edit Curve -state]
    - Relative Mode (Pitch) [slider, Edit Curve -state]
    - Relative Mode (Yaw) [slider, Edit Curve -state]
    - Relative Mode (Roll) [slider, Edit Curve -state]



o Aim Mode	[slider, Edit Curve -state]
▪ Aim Mode (Pitch)	[slider, Edit Curve -state]
▪ Aim Mode (Yaw)	[slider, Edit Curve -state]
- **Ground Vehicle [4]**	
o Ground Vehicle View (Pitch)	[slider, Edit Curve -state]
o Ground Vehicle View (Yaw)	[slider, Edit Curve -state]
o @ui_COMGVPitch	[slider, Edit Curve -state]
o @ui_COMGVYaw	[slider, Edit Curve -state]
Mouse Sensitivity	[slider]
Mouse Sensitivity – ADS – %	[slider]
Mouse Sensitivity – ADS – Zoom Scaling Enabled	[Yes/No]
Mouse Sensitivity – ADS – Zoom Scaling %	[slider]
Mouse Acceleration	[slider]
Mouse Smoothing	[slider]

FOCUS GROUP DISCUSSION TRANSCRIPT

What type of experience with keybinding in games you have?

A: I mostly almost exclusively play MMO's so keybinds are quite familiar to me. I don't really play any simulation games, so I probably won't have that kind of input.

B: I play a lot of different types of games in general. Hard to really pinpoint what kind of games but keybinds, yeah I'm really familiar with them.

C: I'm most familiar with keybinds probably in MMO's what I have played and some strategy games.

D: Well, I play a lot of different games in general, but I also play simulation games, or at least have some experience with them. So I have experience with keybinds from simulation games mostly.

What do you associate with the term 'keybinding'?

A: Well, at least to me in some sense keybinding means that I can customise them to my wants and needs, at least to some extent. I guess that's the main point to me. And that I have some kind of control over it.

B: It really relates a lot to how I play the game, without keybinds certain games can become really difficult to me. So I kinda see them as like a way to play the game proper.

C: To me, the first thing that comes to mind is accessibility, I think it's a very big part of it.

D: What comes to my mind with the term "keybinding" is that it is a key that doesn't necessarily have to be customisable and is just a key that is bound to some kind of action in the game.

How much do you use keybindings?

A: I think that at least for me, most MMO's that I play have so many skills and due to me having tiny, tiny hands, I can only reach up from numbers one to five comfortably. And then six is like, I can reach it, but it's usually some kind of a special action I don't need to press very often. I would say that around 80 percent of my skills in MMO's are in custom keybinds.

B: I customise my controls more in an MMO than in a simulation game. Like, ARMA is one of **the** simulation games for me and because it's a military simulation, most of the controls just kinda make sense to me. Sometimes there can be little thing that might not make sense in the game, but I think because people actually have tested those things a lot in my opinion and it's not a complicated system you have to learn to use like some older simulation games. But in most cases, especially first-person shooters I really don't swap them much. MMO's are pretty much the only ones, especially since I play as a healer, I really need to know my things. I really need to have the heal spells I keep using all the time, so I really need to have them bound on one to five on keyboard. I also need to have my interrupt at a certain button always. In other games I don't touch keybinds too much, maybe to just put crouching to Control or C key but that's it.

C: For me, the thing I spend most of my time rebinding is the melee button in FPS games, because people can't seem to decide if it's the F, C or V key. And sometimes if you have like two buttons for lean like Q and E, you have to use F for interacting and then use C or V for melee which is annoying. But other than that I use them with the couple of MMO's I play.

D: Yeah, I don't really touch them, I have a weird relationship with them and I just think that, oh, I guess the designers are right and these are the keybinds that are best for the game so

I just use them and get annoyed with them quite often though. And like C said, in certain FPS games the keybinds change slightly between games which is annoying. When speaking about simulator games that I play occasionally, especially if using custom keybinds you just have to do that, and it can be quite annoying if you take a long break from a game and it's always a huge setup process too.

What is your preferred platform and controller for playing video games?

A: For me, it has to be PC. I almost never play anything else, and if I need a controller on PC I use Xbox controller.

B: For me, also PC pretty much. And controls mostly mouse and keyboard and if I have to use a controller, it's a PlayStation controller.

C: Yeah, I'm usually on PC and I've got an Xbox controller that I use for platformers and similar stuff.

D: Mostly PC as well and I use Xbox controller or well, if they're driving or third person games I usually use the Xbox controller and with simulation games I do have a HOTAS and a steering wheel and pedals. This relates back to the previous question but especially when using HOTAS, usually there's no default keybindings for them so you basically have to make them from scratch or find binds online that someone else has made. That's kind of a pain in those games. But some simulation games do have default binds like racing games that aren't that complex with their buttons.

Have you ever played space simulator games or vehicular simulator games?

A: So I guess all my simulator experience is from My Summer Car, so not too much.

B: Yeah, I have actually a decent amount of experience when it comes to vehicle simulation games. Like, I previously mentioned ARMA, it's made to be pretty as realistic as possible so they have stuff like tanks, planes, helicopters, bikes... All you can pretty much think of. Aside from that, Space Engineers, definitely. And Empyrion – Galactic Survival, it's a crappier version of Space Engineers but it has some nice things in it but it's more janky and worse basically. I suppose DayZ? Kinda. And other survival games that have cars and similar stuff, so there's quite a few yeah. My Summer Car included.

C: Ah, I've tried to play the old Star Wars X wing and TIE fighter games, the ones from the 90s, but I always hit the wall of "Okay I need to like study for six months before I can operate this damn thing". There's also the thing that if I don't play the game through in like one week, I'm going to forget everything I've learned and start again. But a couple of times I've got enough time in remembering everything and it's been really fun.

D: Yeah, I've played quite many. I'd say I'm like somewhere… well not really, I guess not even close to hardcore, but not casual either on flight simulator games but I've played some Elite, the flight simulator I've mostly played is probably DCS. Microsoft Flight Simulators I played when I was younger, I have also played the new Microsoft Flight Simulator beta a bit and Star Citizen. And of course some driving simulators. I also have bought some expensive controllers like HOTAS, but I rarely use them because the problem with those is that it just takes a long time to get into those types of games and you gotta play them often, especially flight simulators or hardcore simulator games when you have like read a 200-page manual for a plane to actually know how it functions. Like, yeah it can be fun for a while, but it's always "Do I wanna take on the task to learn how this plane works?". It's always a bit of a learning curve to remember what everything does.

Do you ever wish you could change keybindings in a game where there are only default controls?

A: I remember having a really hard time with some Monster Hunter titles, trying to play it on PC on a keyboard was quite the pain. And honestly, why I stopped playing it in the first place was because I just didn't like the experience at all. It's a Japanese port, so basically like Souls games. You can play it on keyboard, but you shouldn't because it's bad. It's really bad.

B: I've played Monster Hunter World on PC and I think it's pretty good, the controls in it are pretty good. Sure, there's some junk, but most of the time it's good.

C: Actually, one thing that came to mind is that when there's a game that you kinda should play on a controller like Dark Souls or something, quite often you can't actually change those controls, which is a bit stupid. Like you're already forced to play with a controller and then you can't even change those. Quite a lot of like 2010 PC ports had a problem similar to that. But yeah, when you can't rebind your controller buttons, that's annoying.

A: That reminds me, the reason why I had a really hard time getting into Elden Ring was because I actually got my Xbox controller just to play it on PC. Before this, I haven't played any PlayStation, Xbox, anything that would require those types of controllers, but I have played on Switch quite, well, not a lot, but when travelling and so on. The thing is that the buttons are in different order, so as I was always trying to use let's say a potion in Elden Ring, in the Xbox button it's the X-button. So I'm always tapping the most up one button because that's where X is on the Switch controller and I always end up doing something stupid.

B: I have a few games too. Mostly a bit older titles on PC. Sometimes I just really wish I could just change the controls themselves, but they just have very specific layouts and you "just have to work with them". In some games like Castlevania I plug a controller and then it's like "oh nice, yeah this is only for Xbox but I have a PlayStation controller" and even though I managed to by jumping through several hoops to get it working, the icons in the game don't change and I have to result to using mods. My beloved Resident Evil games has the same issue. But nowadays most of the time when I just plug my PS4 controller into my PC in lot of the new games, they just work and I don't have to download any fancy drivers, I got all the icons I need. That's one of the positives. It's still not always there but yeah, I wish it was just something people would do more.

C: I think it might be because nowadays most game consoles are basically PCs like from a technical standpoint. So, you can basically just use the same project file in your editor and just export it into the correct system. You really don't have to change everything when you're porting games or try to do a lot of fancy rework.

D: I mean, it's because of Windows basically where that problem comes from. That there was no support in Windows for any other controls than Xbox controllers.

How much customisation do you use in your keybindings?

A: It really depends like when I started playing MMO's, my first MMO was Guild Wars 2 which has quite a lot more less buttons used than your average other MMOs. So, my standard for key binds actually comes from that game, I think I only have bound seven buttons in Guild Wars 2 to custom key binds. And most of the time the buttons they now give for new skills like mounts and everything, it actually makes sense to me. So I've learned and gotten accustomed to those key binds that I have in Guild Wars 2 and every time I try another MMO I always try to get it as close to my Guild Wars 2 key binds as I can. So in Guild Wars

2 I don't have that many custom key binds, but in every other MMO it's like 80% custom key binds.

B: That's a bit difficult to answer, it depends so much on the game. If we're purely talking about keybinds, it really depends so much on the game. In MMOs you actually have to think about that a lot more, as for the rest I just most of the time just go with whatever there is because usually they work pretty well for the most part.

C: Uh, yeah, I don't know, FPSs maybe just a couple, you know, crouch, melee, that kind of stuff. And MMOs, the only one that I've played, or actually that I've **really** played is Final Fantasy 14. And there I think I've redone most of the stuff. The other MMO I have also really played is Runescape, where you don't have key binds because it's basically Diablo, so I don't really have that much experience on that front. But yeah, it really depends on the game, but almost always I change like a couple buttons.

D: I just really don't wanna customise anything really or spend too much time on that in games. Usually, … I do assign them if I have to, but I try to avoid it if I can. But sometimes maybe there's some bindings that you want to use but they aren't even bound. Other than that, I try to stay with the defaults.

How easy do you experience using keybindings?

A: Well, I'd say of course at first if you need to do some new kinds of keybinds that you haven't used in some other games it can be a bit challenging. Especially if it's fast-paced and you're supposed to use the keybinds all the time. And then there's like "Oh, I can't remember quite right like hitting shift and number 5 or control and number 5 instantly", it doesn't come naturally so it takes a bit to do that stuff to get the muscle memory there, but once it's there it's really good. But when you do the keybinding in the user interface? I hate it. I think in most MMOs the UI is most of the time a cluster, it's really bad. Like right now, my most recent keybinds were in FF14, I absolutely hate doing keybinds in there. It's a hassle.

B: From my experience, most of the time I have no problems with those like everything is pretty simple and smooth for me, even in the bit more complex games. But sometimes, there's a bit trouble with what everything does, especially if it's something you don't use often. Actually, when I was playing the latest Tomb Raider, it has like F1, F2, F3 and F4 to quickly craft potions and it doesn't have any icons there, it just shows it to you once and like, I don't ever remember that. I don't really use those keys for that, I'm more accustomed to using a menu. As for the other key binds, the stuff that I use constantly is pretty simple for me, but sometimes when there's a lot of controls, I just can't remember all of them. Especially when it comes to more advanced moves, I have some serious trouble pulling them off because I just don't remember the keys. If there's at least some sort of possibility to have like little icons, it would be more helpful.

C: I've encountered this mostly in older FPS games, they have this where you can bind like many buttons for the same action and sometimes it's really hard to tell or like how to unbind these actions. Sometimes you have a couple of different menus with different kinds of controls, if there's like movement keys and action keys and whatever keys and you're trying to rebind something that is already bound somewhere else, the game just tells you "Hey you stupid idiot, you've bound that somewhere else", but it doesn't like overwrite it or it doesn't show where it's bound. And then you have to find that one key that has like "Q" in it and delete that bind. I think something like, was it Battlefield or something else that I remember having like a really good keybinding system, where it actually showed you everything when you needed it and tells you which keys or which actions share the same key. And you could immediately like jump to the action that needs to be rebound.

D: Yeah, I basically hate the keybinding experience in games with a ton of different keys. In most games it's not really a problem and there's only certain like small amounts of different features and keys in the first place but take specifically simulator games it's such a pain to use the interfaces. Especially when there's things that you don't even understand like "What does this thing do that I'm binding a key for? What are the important things that I actually need to get bound?" Like when there's no default set and you just have to from ground up build bindings yourself, it's annoying. And even when you have bindings or change something, usually there are so many different bindings already and you have no idea like what you even have available anymore to use. Then you just try all of them out until you apply a free key. So, I certainly have those kinds of problems with them.

What is the most annoying aspect about keybinding?

A: Learning. Learning your own keybinds. For example, a new expansion drops in FF. That means you will get around three new skills for every class. So, I'm like "oh no, I have such a good system that these keys are for healing and these for upkeeping damage what do I do now? There's way too many buttons!". It changes my whole system and I have to do some rebinding to make it make sense to me again.

B: Honestly, some interfaces are just plainly annoying to use. People think they designed it really well but in the end they just managed to make it really, really clunky and janky or just overly confusing. Like when they think that "Okay, this will make perfect sense", but no it doesn't because either there's too much stuff on the screen or how it's laid out it's not clear to me. I'm like: "Did I bind this now?". You know those boxes when you click the box and it changes colour and then you press a key to bind it and… most of the time they make sense, but not all the time, sometimes there's like some serious problems, well… okay maybe more minor problems to be honest, but it's still confusing nonetheless. And another thing is that sometimes it's surprisingly hard to find the key binds, like key binds are probably the thing most people want to tweak almost immediately. I would really like to just move through as little amount of menus as possible to get to the controls. That would be nice. When the whole menu system is horrid, it just makes it more painful.

C: Well, mostly kind of the same thing you two said, but maybe an additional emphasis on finding the right kind of layout or the most comfortable layout or whatever you're looking for.

D: There are many annoying aspects I guess. Well, what is the most annoying? I think just like binding a certain function like what is this? What is the keybinding for this thing? I want to do the thing. Just sometimes finding the keybinding from the UI can be a challenge. Especially with like DCS, there's so many of them and then you have to understand like what you are even looking for, so that's certainly annoying. I guess one thing I could mention is when you use HOTAS or these types of controllers and you bind the keys it's hard to actually understand what key you have bound because usually they're labelled like "button 1, 2, 3, 4, 5" and you have like no idea what any of these are. I don't know what the problem with that is or is it with the game or the device manufacturers but it's incredibly annoying and cryptic sometimes. You just have to find something that tells that this key is "button 1" or something like that.

What would make keybinding easily approachable to both new and experienced players alike?

A: Better user interfaces. I honestly really wish there was this system similar to Guild Wars 2 which I really like, it's a trial where you can try out a class in the maximum level where you have all the skills you need. And you have like a target dummy which you can hit and everything and see everything there is to unlock on the character. You can try out like how fluid it feels to play and how to keybind it so it would feel better. I really wish I had that kind of thing more because I think in a lot of games where there's levelling involved and of course I'm an experienced player already, I would like to get into keybind as quick as possible. But if I only have like five skills unlocked for the character, how the hell am I supposed to start the build? If I do the keybindings wrong in the start, then it's going to be really difficult to forget all what I did in the late levels.

B: More simple UI pretty much. In my opinion, simple is sometimes better. Don't hide all the stuff unnecessarily and try to be fancy, just put it somewhere where it's easy to access. And of course test it enough that it makes sense.

C: Actually have rebindable keys, that's always nice. In MMOs you can customise your HUD so maybe add like a quick check button, like press Tab and it shows the most common or the most common but not always in use hotkeys, for example that potion crafting thing in Tomb Raider. Maybe even so that you could like select which buttons it shows when you press the hotkey. It would help with the "I haven't been playing this in three months and don't remember anything" problem.

D: Yeah, especially in games that have a lot of different keybindings I think it would be good to maybe highlight the most important ones, like it's not just a huge list of all the possible things. A search feature for specific input would be really helpful too. You could actually find the things that you absolutely need. Maybe even explanations to what these are in like simulator games. It would make it easier for some people that don't play those that often. Also, better tutorials that can go through keybindings could work as well. Especially after a long while that you can just quickly find what are the most important things that I need to know about keybinds.

A: Oh, actually I wish there was like maybe from the game side or even in the community, but I really wish there was like some people who would make like dummy guides for keybinding. Like, people who don't have experience in these things sometimes don't know that they can actually use letters or maybe combination keys. When I was learning I had to ask my friend who was able to help me get started. But like when I try to look online about, not that I need a guide but at least some pointers, but people respond with "Do what feels best for you", which is… not helpful at all if you don't know what to do in the first place. For example, my friend, Final Fantasy is her first MMO and she's only played for half a year and I just recently realised se actually like clicks everything. She just didn't know how to get even started with the process of learning keybinds, because of course if you have already gotten used to clicking then doing the keybinds is really hard, it takes a lot of effort and time to actually build the muscle memory for the keybinds. And because she doesn't have a preference on what types of keybinds she likes, it's such a hassle because there's no help online about it.