

Hana Mori

# APPROACH TO MULTI-PLATFORM GAME UI DESIGN

Bachelor's thesis  
Game Design

2019



South-Eastern Finland  
University of Applied Sciences



Author (authors)	Degree	Time
Hana Mori	Bachelor of Culture and Arts	April 2019
<b>Thesis title</b>		
Approach to multi-platform game UI design		45 pages 20 pages of appendices
<b>Commissioned by</b>		
Bohemia Interactive a. s.		
<b>Supervisor</b>		
Suvi Pylvänen, Senior Lecturer Petra Oravcova, UX lead		
<b>Abstract</b>		
<p>Game developers often face unique problems when transferring their products between different platforms. Not understanding the differences between these platforms can cause difficulties and dissatisfaction for the end users. The objective of this thesis was to gather and analyse these existing problems in existing games and find solutions for them that would be applicable in future designs.</p>		
<p>The thesis studies different design principles, methods and approaches that are used when designing user experience (UX) and user interfaces (UI) in games. It also analysed existing input schemes in different multi-platform games. These findings were then applied to the design of a keyboard and mouse input scheme for the game Vigor, which uses a controller input scheme at present. The goal of the production was to use the results of the research to create the optimal input scheme and easily understandable user interface for the use of a mouse and keyboard. The side product was creating a set of guidelines for making UX and UI choices that can be applied to the production of other games.</p>		
<p>The result was a set of rules and finished UI mock ups used in the production of the game Vigor. Specifically, the UI was adapted for a keyboard and mouse input scheme and guidelines were set in place on how to approach future designs.</p>		
<p>The finished result was of use in the production of the game and was added to the documentation needed for implementing the design into the product. However, some of the designs were not actually finalized due the ongoing development of the game, changes in design and production, as well as the lack of resources and time needed to conduct user testing. Testing specifically was needed for the finalisation of the results. Despite this, the thesis achieved its goal of providing solutions when designing UI for multiple platforms to a satisfactory degree in regards to the circumstances.</p>		
<b>Keywords</b>		
game design, user interface, user experience, thesis, multi-platform		

# CONTENTS

GLOSSARY .....	6
1 INTRODUCTION .....	8
2 WHAT IS USER EXPERIENCE .....	9
2.1 UX in game development.....	10
2.2 Use of UX in games .....	11
3 UX DESIGN GUIDELINES .....	12
3.1 Breaking down UX .....	12
3.2 Gestalt Principles.....	13
3.3 Affordance, form and function.....	17
3.4 User testing, feedback and analysis .....	17
4 MULTI-PLATFORM GAMES .....	19
4.1 Brief History of Multi-platform games .....	19
4.2 Development focus .....	20
5 PRACTICAL APPLICATION TO VIGOR .....	25
5.1 Preliminary research .....	26
5.2 Analysis of the existing scheme .....	26
5.3 Mock ups .....	28
5.3.1 Shelter menu .....	28
5.3.2 Options menu .....	32
5.3.3 Lobby and in game screens.....	35
6 CONCLUSION.....	38
REFERENCES .....	41
LIST OF FIGURES .....	44
APPENDICES	

Appendix 1. Shelter menu mock ups

Appendix 2. Options menu mock ups

Appendix 3. Lobby screen mock ups

Appendix 4. In game screen mock ups

## **GLOSSARY**

Due to the nature of this thesis, some game industry and design terminology is used. The terms are gathered and explained in the following list with a short explanation for each.

### *Head Up Display (HUD)*

A head-up display, or HUD, is a transparent display that provides the user with information, without making the player look away from their normal point of view. In games, the term refers to the method by which information is visually conveyed to the player while a game is in progress. The HUD can be used to display several pieces of information such as the main character's health, items, and in game indicators simultaneously. (Fagerholt and Lorenzon 2009.)

### *Input Scheme*

An input scheme is a systematic organized configuration of commands, which are mapped to specific keys used to communicate those commands to the computing device. Individual schemes can vastly differ between different devices and types of controllers used to input those commands.

### *Iteration*

The process of repeating a predetermined set of operations to achieve a specific result (Lidwell et al. 2003, 118).

### *Mock up*

A mock up is a form of a prototype, the use of an incomplete model of a design, which is used to explore ideas, refine specifications and test the functionality of the design (Lidwell et al. 2003, 158). It provides a view of what the finished product could look like.

### *Multi-platform or Cross-platform games*

The term multi or cross-platform refers to game software that is available on multiple different devices. In the games industry this refers to games that can be played on PC, different consoles and Mobile. Cross-platform specifically means that players playing the game on different platforms can directly interact with each other, while multi-platform only means that players play the same game, but on separate, different platforms. In this thesis, the term multi-platform will be used.

### *Port*

The term port relates to a piece of software, a game, that has been developed for a specific platform and later transferred to be played on a different platform or multiple platforms.

## 1 INTRODUCTION

The game industry has grown exponentially in the past few decades, transitioning from a fairly niche market into a mainstream industry of giant proportions. In the early days, games were only ever developed for one target device and were limited by the current available technology. As technology advanced, so did the complexity of the games and with it, the need for a clear and readable user interface, to properly communicate the games mechanics to its user, the player.

With the advancement of the internet and online play, as well as more advanced development software, games started being more widely available. While multi-platform games were available decades ago, the biggest surge happened in the sixth generation console games, where porting a game from one console to another became increasingly less difficult. Now, the majority of games usually end up being transferred to others at some later point. The games industry is, after all, an industry, and skipping on profits by not offering the product to as many players as possible seems like a missed opportunity.

Game ports, games that are developed and released on one platform and later adapted and released on others, often face a variety of obstacles. Each platform has a unique and well established input device, where specific keys are used for specific actions, and have done so for years, with very little changes. This makes it easier to design user interface (UI) and user experience (UX) elements in a game for the targeted platform, but problems arise when a game is ported. If done poorly, without understanding the differences, the players playing a ported game can feel frustrated, annoyed and might not have a fun experience, which can have disastrous consequences. User experience specifically, is the discipline most commonly used to solve these issues, as it is used for solving problems that have the end product, and the end users best experience as the goal.

This thesis will try to determine and analyse the problems many developers face when developing games for more than one platform. These most commonly

include input and interaction schemes that are not optimised and are often counterintuitive for the users on the targeted platform. It will start by introducing a brief history of UI and UX design. It will look at the most important basic design principles and guidelines that are used in either or both of those disciplines. Afterwards examples of different games that have different input schemes integrated will be analysed. All of this will be done to help and resolve issues often associated with multi-platform games.

After the theoretical part, the thesis will focus on the practical part, which consists of creating an alternate input scheme, navigation, layout and interaction for the game Vigor. The original input scheme is designed for the use with a controller, while the end product of this thesis will be designed for the use of a mouse and keyboard. The phases of the design will include iterations of mock up schemes, and depending on the time, resource and technological constraints the UI will be prototyped and tested in engine. If possible, some testing will be conducted and the results applied to the product to finalize the design. After the testing is concluded and the end product is satisfactory, the research results will be summarized, analysing how effective the application of research findings and game analysis were.

## **2 WHAT IS USER EXPERIENCE**

Outside of the games industry, specifically in web design, UX is a generally well-established discipline, building on years of knowledge of designing for the end users in different fields (Long 2017). User Experience is a particular discipline of design that centres around psychology of the end user, in this case the player, their behaviours and thinking processes. It applies this knowledge and combines it with collected data, interactive design processes and different types of testing. (Long 2017.) This can also be referred to as all qualitative experience a user makes while interacting with a product (Bernhaupt 2010). UX can also be defined in more precise terms as a means of a person's responses and perceptions that resulted from the use and/or anticipated use of a system, product, or service

(Nagalingam & Ibrahim 2015). UX in games can be evaluated using a variety of concepts including engagement, immersion, playability, flow and fun (Bernhaupt 2010).

## 2.1 UX in game development

Focusing on the impact of different design choices on the player, UX designers in game development are able to reduce key risks that are inherent to such products. An example of such a risk is the so called *complexity creep*, where games are too difficult to comprehend, and parts of the design have to be redone many times over until the player can understand it. (Long 2017.) This is suboptimal since it consumes valuable development time and funds, which are very limited, meaning they have to be redistributed, or other features have to be dropped in order to accommodate the time consuming fixes.

During the development process, teams often falsely predict the reactions of their players, which can happen for various reasons according to Long (2017):

- developers do not perceive the game in the same way a real player would
- they do not design the instructions and prompts for non-expert players as they themselves are experts, meaning the games are often not suitable for players who are casual or novices
- they make incorrect assumptions about their target audience, potentially risking making a game for no one.

Therefore, having a player centric voice can be essential, since it is sometimes hard for developers to understand the difference between what matters to the players and what matters to them. Another term for this is UCD or *user centric design*, which is defined as design that takes into account traditional user needs as well as those that are abstract and subjective in nature (Nagalingam & Ibrahim 2015), for example what an individual player considers fun.

## 2.2 Use of UX in games

The tasks of the UX designers change during different parts of the development process. A UX designer's output can be in various forms depending on the needs and wishes of team members. Most often it is in the form of documentation like wireframes, mock ups, reports, feedback analyses or just working with existing UI designers and game designers and providing feedback directly. It is also stated that UX helps define who the target audience is and looks for good examples from the competition. UX attempts to make the right choices early on, by questioning the decisions as they are being made, not in retrospect. (Long 2017.)

The exact UX tasks depend, as mentioned before, on the current state of the development process and can differ between individual projects, as well as between different UX designers (Tandon 2018). It often also depends on the scope of the project. A small project, like a small role playing game will most likely require less user playtesting and analysis compared to a big multiplayer action game.

According to Long (2017), UX designers in game development can generally be divided into four sub roles:

- The UX Lead, a higher level voice for maintaining a player-centric approach in the development process
- UX Designer, tasked with visual design, applying player psychology and design knowledge to support game and UI design
- Games user researcher, in charge of play testing groups and research
- UX data scientist, who captures and analyses player behaviour through statistical data

Regardless of an individual's role, UX analysis can offer valuable information from the end user to the developer in order to provide the best possible experience.

### 3 UX DESIGN GUIDELINES

Games UX is a discipline that deals with problems in game development, helping the games reach their goal based on experience. It uses formalised processes to uncover weaknesses in a game's design and the ways it informs the player of its intent. Without proper UX approaches, games can and often will fall victim to issues, such as lack of understanding and difficulties in teaching and aiding players different from ourselves. This affects the perceived quality and enjoyment of the game. The UX can be broken down into layers to make it easier to decide where the player is having a negative experience and what tool to utilize to come up with the best solution. (Long 2017.)

#### 3.1 Breaking down UX

In games, UX relates to the games controls and UI. This is a very important issue, as we need to make sure it is supportive and easy to use, since it enables players to utilise the core mechanics of the game. When defining UX, the term most often used is *usability*, however the terms are not freely exchangeable. While usability is considered part of the UX, the latter encompasses more than just usability. According to the standards for human centred design (ISO 9241: 2018), usability is defined as the “extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. “

Usability can be further broken down into individual layers that help us understand how users perceive certain actions. In the words of Nielsen (2012), the five individual components of usability are learnability, efficiency, memorability, errors and satisfaction, however, for the purposes of this thesis, the author chose to instead combine Nielsen's (2012) break down with that of Hodent (2016) and Long (2017).

Hodent (2016) breaks down usability into an informed loop, which encompasses the player's ability to understand given information, memorize it and prevent and

recover from errors. Long (2017) on the other hand, combines usability with learnability. He defines the latter as the player's understanding of the games core rules and mechanics, and the former with the player's success of utilising and navigating those mechanics. These together then lead to what Long (2017) calls *emotional experience*.

Hodent (2015) further elaborates on her break down of usability by emphasizing the importance of context, as usability is influenced by the surrounding information. The information provided to the player should make sense based on what is happening at the current moment.

Understanding this breakdown, a UX designer will use certain methods and approaches to come up with solutions depending on each individual issue at hand. Having unified guidelines on how to approach these issues is important, in order to make the revision process as streamlined as possible. The whole procedure can be divided into four steps which represents the core iteration loop: design, implement, measure and assess. This loop is used to reiterate designs based on feedback to come to the best solution possible. (Long 2017.)

### **3.2 Gestalt Principles**

The Gestalt principles were first introduced by Wertheimer in the years 1923-1938, and were further developed by Köhler, Koffka and Metzger. The principles are often used in graphic design and UI, to help the viewer better understand the information displayed on a screen, increase readability and play with our perception. The human brain is set to create patterns which designers can effectively use when designing game UI. (Vriend 2017.)

There are several Gestalt principles, and they can be of great use for the usability of UI as they are intuitive for the player to understand and vastly improve the interaction with the mechanics the designers have implemented by making the displayed information more readable. (Vriend 2017). While the Gestalt principles

are of great value to when designing for UX and UI, not all were directly applicable in the design part of this thesis. Therefore, the author decided to include only those that were essential for the actual production.

### *Principle of Proximity*

If elements are regionally or chronologically close to one another, our mind assumes they belong together. In games this principle is used when designing an inventory, menus or a HUD in general. Elements that have a similar function or that relay similar information should be placed close to each other, as is demonstrated with the hero information icon in Figure 1. This also means that elements with different function should be placed further apart. (Vriend 2017.)



Figure 1. The proximity of the “!” icon indicates that it provides information for the chosen hero in *Overwatch* (Overwatch 2016)

### *Principle of Closure*

Closure occurs, when an objects’ line or space are incomplete. If enough of the shape is indicated, our minds can perceive it as whole by filling in the missing elements. When designing for closure, it needs to be made sure the player has enough information to fill in the gaps. (Vriend 2017.) Figure 3 illustrates how to

can make game mechanics that revolve around the principle of closure by having the player fill in the missing pieces based on the surrounding elements.



Figure 2. The brain has enough information to assume what the missing pieces are in the board game *Saboteur*. (*Saboteur* 2004)

### *Principle of Enclosure*

Elements can be grouped together and are perceived as a unity, if they are enclosed together. This allows for unification of elements that are otherwise dissimilar. Applying this can be as simple as adding a border around the chosen elements. (Vriend 2017.) In games this is commonly seen in visual novels, or other genres where there is dialogue actively displayed on the screen. This can be observed in Figure 4, where the face of the character is framed with the text, to indicate it is that character's dialogue. It can also be used to indicate interactable areas within the UI by enclosing different visual elements to create new objects.



Figure 3. The enclosure of the character icon and text in *Undertale* implicates they belong together (Undertale 2015)

### *Principle of Figure and Ground*

The human eye can differentiate an object from its surrounding area. A silhouette, a form or a shape are naturally perceived as the point to focus on, while the surrounding area is perceived as background. By balancing these elements, the viewed image can be made more clear, or add interest and subtlety to an image. All of this can be achieved by mixing high contrast shapes and colours, negative space and different patterns. (Vriend 2017.) As observed in Figure 4, this principle is often used for logotypes, or when it is wanted for the player to focus on certain information first.



Figure 4. *DayZ* Logo is an example of using the Figure and Ground principle to add interest in the form of negative space (DayZ 2013)

The negative space in the letters is formed in the shape of the landscape and elements from the game. The added details add visual interest and add subtlety, while the letters act as the background for them. Combined, this creates a very strong and memorable visual element.

### **3.3 Affordance, form and function**

Hodent (2017) states that the primary rule for usability is that form follows function. The player should be able to figure out function from looking at an element with little difficulty. This is also referred to as *affordance*, or the properties of an object that tell the user what actions might be done with it. Norman (2013) refers to affordances as actions that the user considers possible. This means that affordances of a particular object (or UX elements) depend on the users' physical capabilities, their goals and past experiences, which helps speed up the process of the player getting familiar with the game and how to interact with it. A general rule of thumb is that the player should have to think as little as possible about an action. (Krug 2014).

The use of stereotypes in icons may help the player identify their function (Hodent 2015). Hodent also reminds us that ambiguous elements, something that can appear one way or another, should be avoided, as they are confusing and make the player guess. The same applies to the use of similar icons and very similar symbols for different actions. Clarity and consistency are key, but there should be distinctly different forms used for different functions.

### **3.4 User testing, feedback and analysis**

User testing and feedback is a useful tool that provides the designer with both qualitative and quantitative data on how their designs perform in action. However, it often occurs that a studio starts getting player feedback very late in the design process, and is then not able to make proper adjustment fixes to the usability flaws (Long 2017).

Since players lack the language to explain their feelings, it is common that usability frustration influences their feedback on if the game is actually fun. Furthermore, Long (2017) states that the feedback provided by the players is often not worded in a way that can be useful to the developers, and people who are inexperienced in ways of handling it can find it demoralizing. Actions are usually more indicative and can provide us with much more useful information.

When it comes to the testing itself, it is crucial that it is set up properly beforehand. According to Krug (2014), it is important to distinguish two terms. *Focus group* testing, which is where a small group of 5-10 people discuss their opinions about products and their experiences with them. This is a step that should be done before any designing begins, as it offers valuable information on whether you are making the right product.

Unlike focus group testing however, *Usability testing* is about watching one person at a time, while they interact with your design, whether it is a wireframe concept, prototype or the end product. The main difference between usability testing and focus group testing is that in the latter, the designer is actively observing the person who is interacting with your design, instead of listening to them talk. It is a useful tool that should be utilized throughout the production. The number of users being tested is not as important, rather it should be done as often as possible and start early in the production process. (Krug 2014.)

There can be a variety of problems that users encounter during testing, but there are certain types that come up most often (Krug 2014):

- Concept is unclear to the user, meaning the information provided is not understandable enough (e.g. confusing icons in a UI scheme)
- The information required is not there; meaning the designer wrongly anticipated what information the user needed, or it is worded differently than what the user would describe it as

- There is too much noise, which indicates that while all the information might be present, the visual hierarchy needs to be readjusted either by removing some elements, or bringing emphasis to the most important ones.

The results of a usability test should be analysed as soon as possible after a test has been concluded. The severity of problems varies, and the worst should take priority to resolve. The solutions will depend on the resources available at the time, and a complete fix might not always be an option. However, even a smaller fix that decreases the severity of the problem is preferable to no action taken.

## **4 MULTI-PLATFORM GAMES**

As the term suggests, a multi-platform game means the same game is available on a variety of platforms. These can usually be grouped together into several categories for easier understanding.

- The term *Consoles* includes all stationary consoles that are played with a controller (e.g. Xbox and PlayStation). The designs of the controllers vary between manufacturers, but they have the same functionality and will often share the same amount of input options.
- *Handheld devices* refer to portable device that the user operates with their hands and that generally do not require special equipment like a separate monitor. These include tablets, mobile phones as well as handheld consoles like the PSP and the Nintendo switch.
- *PC* is a more generalized term used in this context, and usually requires the mention of the operating system (OS) as well, since not all games are supported on all available OS'.

In this work, the PC will refer specifically to a device that operates on the Windows OS.

### **4.1 Brief History of Multi-platform games**

Multi-platform games have been around for decades, the only difference between then and now being the number of individual titles. They were not uncommon

even in the 16-bit and 32-bit console generations (SNES, Genesis, PlayStation), but their number significantly increased with the 6<sup>th</sup> generation consoles (PlayStation 2, Xbox, Game Cube). The choice to develop for multiple platforms used to be more common with western developers and eastern developers focused more on solely developing for consoles. Nowadays the development focus is equal regardless of the geographical background. Platform choices also depend on the availability and marketing of specific platforms in a region.

Regardless, by the 7<sup>th</sup> generation of consoles, multi-platform development became the standard, especially amongst third party developers, both due to increasing budgets and revenue of mainstream games, as well as standardization of certain components in technology, meaning development became more streamlined as the differences between individual platforms (different consoles, PC) decreased. The latter is also due to the fact that different input devices (controllers, mice, keyboards) are now more a thing of preference and companies are making a bigger effort to make games playable for people with diverse accessibility needs. (Sinclair 2018.)

## **4.2 Development focus**

As stated, the term port relates to a game that has been developed for a specific platform and later transferred to another. The majority of AAA titles are released simultaneously on multiple platforms, however, the exact development process and design highly depend on what the perceived focus or target platform is. As this thesis revolves around the transferring of input schemes between PC and console, mobile platforms will be excluded.

According to Sinclair (2018) the two main differences in UX and UI can be attributed to the different viewing distance and the difference in the number of available keys between a controller and a keyboard. Hence, it is easier to transfer the UX from a console design to the PC instead of the other way around. There are three main approaches that can be taken by companies when adapting the

UIs between the two different platforms. All three approaches are taken from Sinclair's 2018 article.

*The prompt switch* is when the design is the same for both console and PC, only switching the button prompts and icons in menus and HUD, as seen in Figure 5 and Figure 6. While this approach is not inadequate, it clearly favours the use of a controller and can sometimes be counterintuitive to use, especially in cases in which a PC game would usually favour input with a mouse instead of the different keyboard keys. This is one of the most commonly used approach, as it is easy and cheap to develop, though it does put pressure on the player to read over all the key prompts. (Sinclair 2018.)



Figure 5. The menu interactions in *Destiny 2* with controller button prompts (Destiny 2 2017)



Figure 6. The PC version of the menu follows the button prompt switch approach (Destiny 2 2017)

Sinclair uses the game *Dishonored 2* as an example for the prompt switch, the author chose to use *Destiny 2*, as it also follows the same principle. It is worth mentioning that while *Destiny 2* follows the button prompt switch approach in regards to actions, the actual navigation is done through the *free cursor*, which is a free floating cursor akin to the use of the mouse on the PC.

*PC leaning* is when the input scheme is more in favour for PC users and the direct input with a mouse. Still it can be adapted for console use with only minor issues, most commonly the duplication of buttons, double focus and the use of nonstandard buttons for primary actions. The latter is applicable in console-to-PC ports as well. (Sinclair 2018.) An example of this method is the game *Fortnite*, as seen in Figure 7 and Figure 8, taken from Sinclair's 2018 article. In Figure 7, the PC version features a variety of tabs and buttons, which are grouped up and visually distinct. It is designed for navigation with a mouse, which allows the screen to be free of any key prompts.

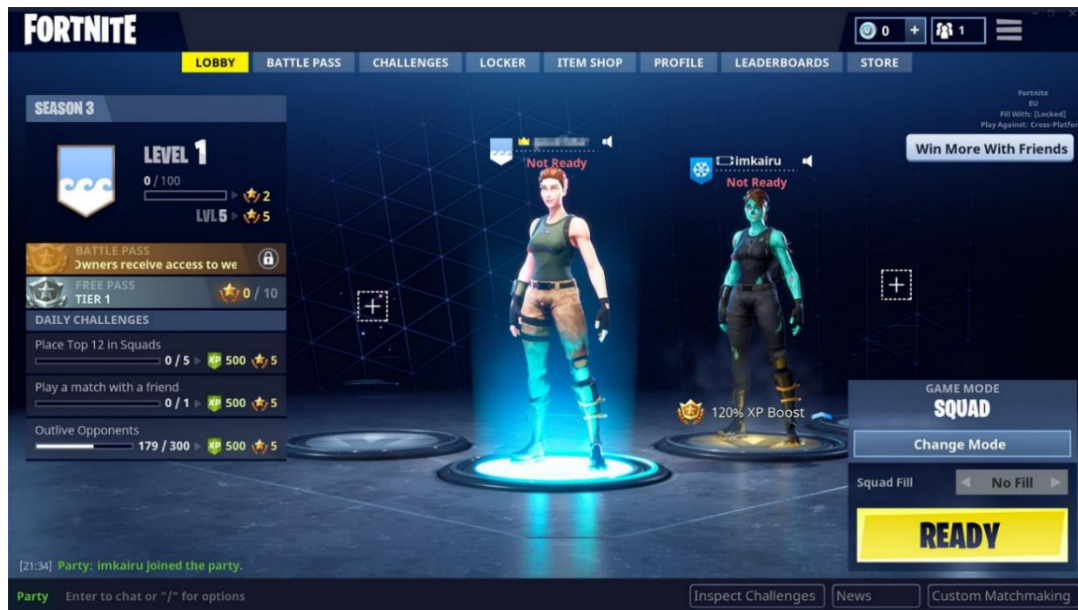


Figure 7. The variety of tabs and options in Fortnite can be easily navigated with a mouse (Fortnite 2017, cited in Sinclair 2018)

Figure 8 shows the same menu with console controls. The layout is largely the same, only key prompts have been added when necessary. Due to the layout and design, there are some issues with double focus (the battle pass, lobby and play button), but nothing is overly conflicting and the UI provides the player with enough information.

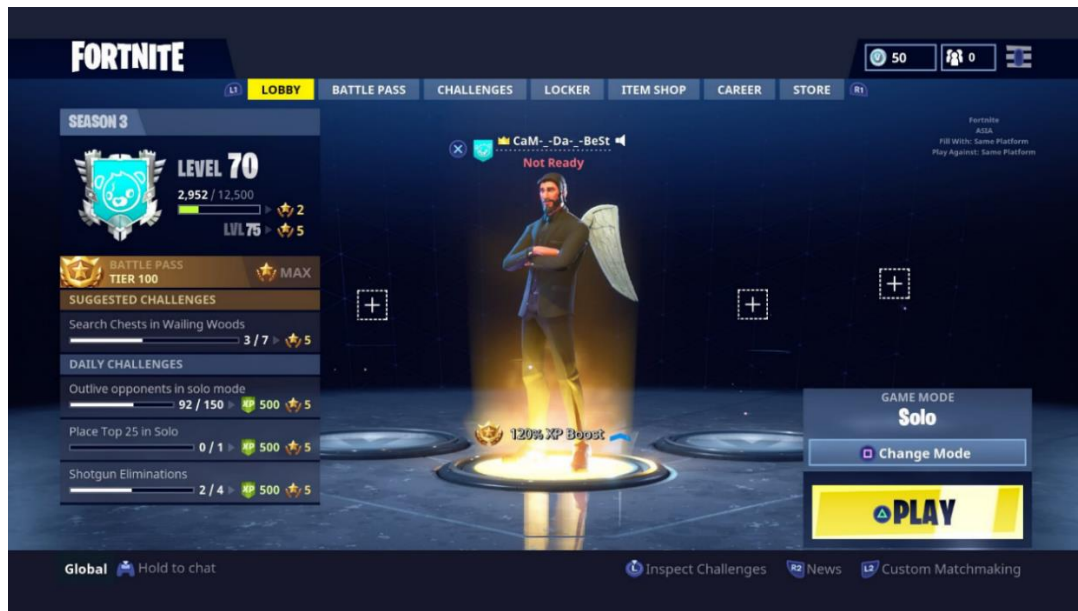


Figure 8. The console version of *Fortnite* is learnable but has some suboptimal UX features like double focus of tabs (Fortnite 2017, cited in Sinclair 2018)

Another example of the PC leaning approach is *The Witcher 3: Wild Hunt*, however there the UX and UI were executed better as it is easier to navigate and use. It makes use of a grid system, which even though it is evident the best way to interact with it would be with a mouse.

Sinclair (2018) refers to the optimal method as being *Secretly Console Focused*. This method focuses on designing for console, however the design is ambiguous and adaptable enough so that PC users cannot pick up on the fact it was not designed specifically for PC. This is done by having a unified high level UI and UX layout, but having different interaction designs for each platform. In some cases, this involves only adding button prompts for console (the commonly used select and commit approach) as seen in Figure 9, while leaving the PC UI without any additional directions as it is intuitive to navigate with a mouse and click when selecting an element, like in Figure 10.



Figure 9. No key prompts on the PC version of the games UI as it is navigated and interacted with a mouse *Overwatch* (*Overwatch* 2016)



Figure 10. Key prompts are displayed when using a controller for the “select and commit” method in *Overwatch* (Overwatch 2016)

This is where the use of Gestalt principles like proximity, enclosure and similarity are of great use. The switch between devices is done by simply toggling off different UI elements which depend on the choice of input device currently in use. The differences are kept at a minimum, but are still intuitive and comfortable to use, based on the input device chosen.

## 5 PRACTICAL APPLICATION TO VIGOR

*Vigor* is an upcoming free-to-play looter shooter game developed and published by Bohemia Interactive. In the game the players’ objectives include building a shelter to survive in post-war Norway and going to different locations and fight other survivors in order to secure resources. The games’ current UI and UX are designed for the use with a controller. As part of this thesis, the current interaction and input scheme will be converted for the use of a mouse and keyboard set up. This chapter will cover the practical execution of this thesis, including the methodology and analysis of the results.

## **5.1 Preliminary research**

The thesis so far focused on summarizing the preliminary study, which is comprised of literature and game study. Following the work of Fagerholt and Lorenzon (2009) this consisted of looking at different related academic articles, relevant literature, analysing examples from similar games and games that use similar mechanics as well as looking for examples from other forms of media, particularly web design.

Based on the research done, the author's goal for the design will be to make minimal changes on the high level UI and only making changes when looking at second and tertiary actions, as advised by Sinclair (2018) in his article. The production part will consist of designing mock ups and iterating on these designs, following researched design principles, which will afterwards be evaluated through a focus test.

## **5.2 Analysis of the existing scheme**

Vigor has several areas where the player directly interacts with the games' UI. The games' main area is the Shelter, the players' home base, where the games' main menu (or shelter menu) can be located. This is where the player conducts the majority of their interactions, which includes upgrading their player base, managing their equipment and customizing their character. The second most important interaction area is the HUD in game and the action the player can undertake while actively participating in a match. The in game scheme includes a variety of actions, however due to the minimalistic design of the HUD, does not feature any prompts while playing and only displays the most crucial UI elements. There are other secondary menus in the game, which include the settings screen, the match lobby and several small menus for special events, like the shooting range challenges. These offer a variety of information to the player, but have a very small number of different actions the players need to use.

Before looking at the menu actions, rules had to be established in order to maintain consistency across all screens. The purpose of this is to make systems more learnable, since consistency allows players to learn new things faster, by focusing their attention on the task at hand, instead of having to re-learn the actions needed to perform the task for every new instance. (Lidwell et al. 2003). It is important to remember that while consistency is important, it should not be sacrificed in place of clarity, meaning that if something can be made significantly clearer to the player by making it less consistent, the designer should rule in favour of the former (Krug 2017). With this in mind, the following rules were set in place:

1. The primary action has to be initiated one of two different ways, depending on the information given. If there is information displayed on focus, the primary action has a prompt displayed for it. This is the *select and commit* method. If there is no information displayed, the primary action does not have a key prompt, only an interactable button. Here the method is *only commit*.

To elaborate, the select and commit approach is done when there is information displayed on focus, as well as when there are multiple actions applicable to the element on focus.

2. The actions should follow the current set up of the console. This means that a primary action with standard input commands for the controller, equates to a primary action in a keyboard and mouse input scheme.

This rule is in place with the target audience in mind with the intent to accommodate console users to a different input scheme.

3. All key input prompts should be clickable with a mouse, with the exception of tab switches and actions that include holding a key as a primary action.

The latter rule is there for the purpose of including players with disabilities, by offering different ways to interact with the UI. While it is set as a rule in this thesis,

it might be left out from the actual in game implementation later on, depending on technical and resource capabilities.

### **5.3 Mock ups**

In order to be able to provide the necessary documentation for the later implementation, mock up prototypes had to be designed to portray the requirements needed for the changes in the input schemes. The mock ups were divided into sets of screens taken from the game, which were then used as a base on which to display the changes on. This was done to more easily maintain consistency and deviate as little as possible from the original designs, while still allowing for changes to be made when necessary.

#### **5.3.1 Shelter menu**

As mentioned before, the Shelter menu is the most complex interaction area in the game, and requires a variety of different actions from the player. This area specifically needed to have rules established, in order to maintain consistency amongst the variety of possible interactions. The controls were divided into two groups based on their purpose. The first group related to navigation and the second to controls that are used to initiate specific actions.

Figure 11 shows what the current input scheme is for the use of a controller. Navigation of the menu is done by the use of the triggers to switch between tabs and subtabs. This is a standard approach and is consistent across all the games menus where tabs are used. To navigate through the various UI elements and options, the game uses the select and commit option, having the player select the element by the use of the left thumb stick and then perform a specific action with it by choosing one of the possible buttons. All of the button prompts with corresponding actions are displayed in the bottom row.

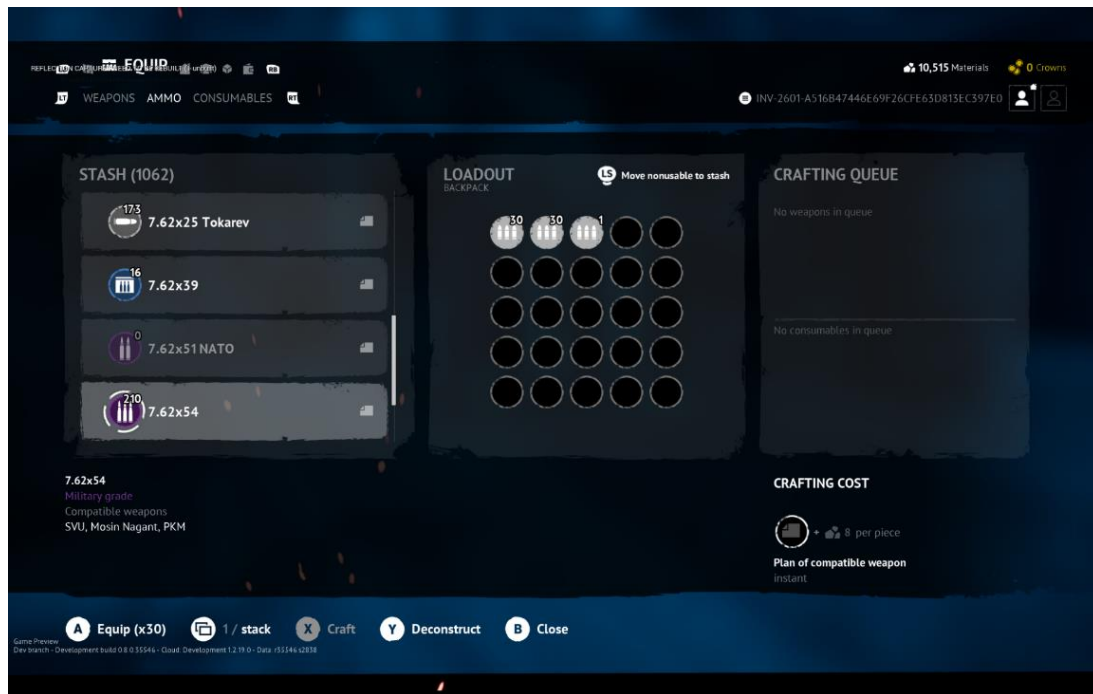


Figure 11. The original equip screen from *Vigor* with the controller key prompts (Vigor 2018)

The equip screen was the first to be designed with the new input scheme and helped solidify the rule for the differences in prompts, based on the information provided for the player. This was primarily done for two reasons. Firstly, it is the screen that the player will interact with the most and by having it be well designed it will help the player navigate other menus they will interact with. Secondly, it is the most interaction heavy, with a variety of different options and a lot of information for the player. By resolving any issues that came up here, it provided a base which made resolving any arising problems in other screens easier.

There were several changes made to the design of the UX and UI, as seen in Figure 12. The navigation of the shelter menu is still done by switching tabs, however this can now be done two ways. The primary way is in line with the new input scheme and is navigated by the use of the primary mouse button. This is common place both in other games as well as the way software is commonly navigated on the computer. The other option is for the tabs to be navigated with the “Q” and “E” keys and the sub tabs with the “Tab” key. The latter was added at a later point after some of the other options were explored. These prompts are

the exception of the clickable prompts rule established earlier, as the tabs themselves are clickable, which makes the rule redundant in this particular case.

The UI of the tabs was adapted to be more easily interacted with a mouse. The icons are now accompanied by text and the spaced a little further apart. This was maintained throughout the whole shelter menu (Appendix 1). As the weapon and other items display some information on focus, which is when they are hovered over with the mouse, the prompt for the primary action, was added at the bottom. All other possible interactions also have their individual prompts displayed.

The keys used for the interactions, were chosen based on what is most commonly used in other games for corresponding actions, as well as what feels most comfortable to use in regards to where the player's hands are positioned on the keyboard. The same keys were then used for any other similar action in other screens. The left mouse button and the "F" key were used as the primary action. The mouse button was chosen for selecting and equipping items, while the "F" key was used when the action was important, but sometimes required a longer interaction (key hold). If there was another prompt needed in a menu, the "V" key was used.

The "shift" key was used for a variety of actions, however they were generally similar in function, usually including some form of switching between different elements. The "escape" key was used in place of the "B" button on the controller. This means it was used as the action to return the user to the previous menu, or to exit the menu completely. The author chose to display the prompt for this even though this key is commonly used for this purpose. The decision was made based on the target audience, as this is not directly targeted at PC users, who would be familiar with this interaction type, but instead targets the users of the Xbox One, who decide to use the keyboard and mouse. The latter might not be as familiar with the use of the escape button; hence the prompt is being displayed with the others in the bottom row. In order to be able to access the options menu from within another menu, a button was added next to the profile option at the

top. The key for accessing the options menu within the game has not yet been determined by the design, at the time this thesis was being completed.

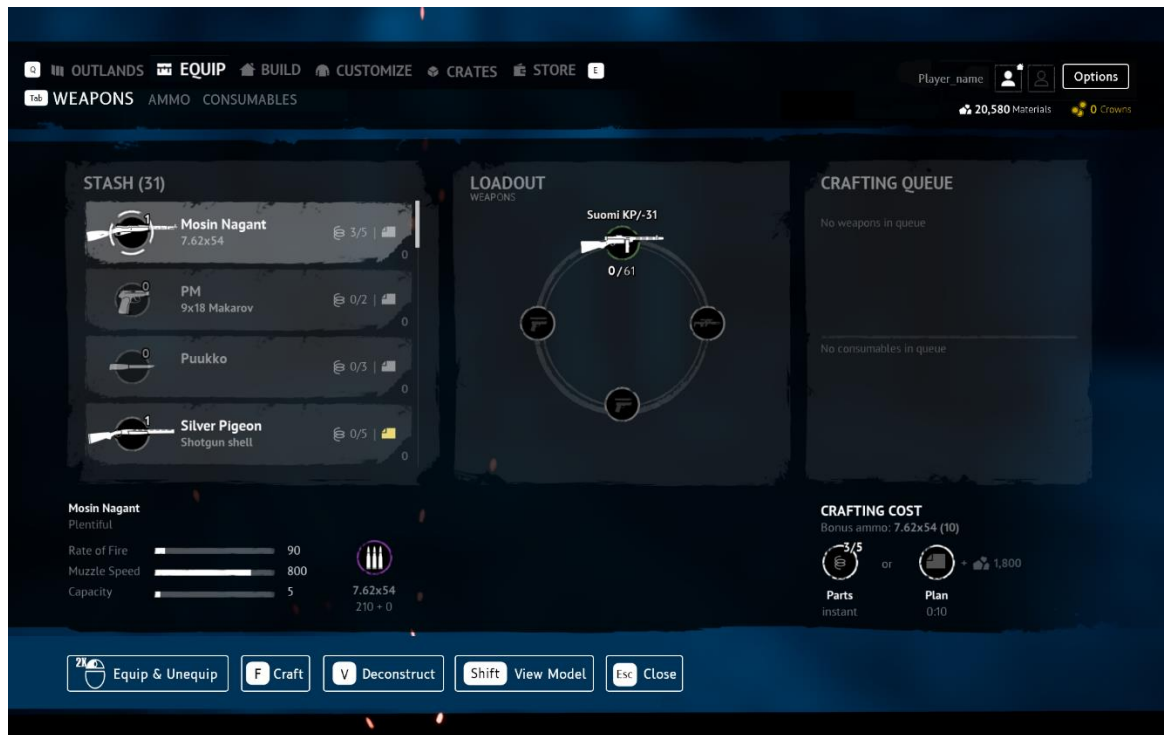


Figure 12. The final mock up design for the equip screen

The displayed key prompts were enclosed with a simple rectangle. This helps indicate that the enclosed elements are clickable as well as that they belong together, following both the Gestalt principle of proximity and enclosure (see Vriend 2017). The same design of enclosing prompts was followed across all screens and instances, including confirmation windows (Figure 13).

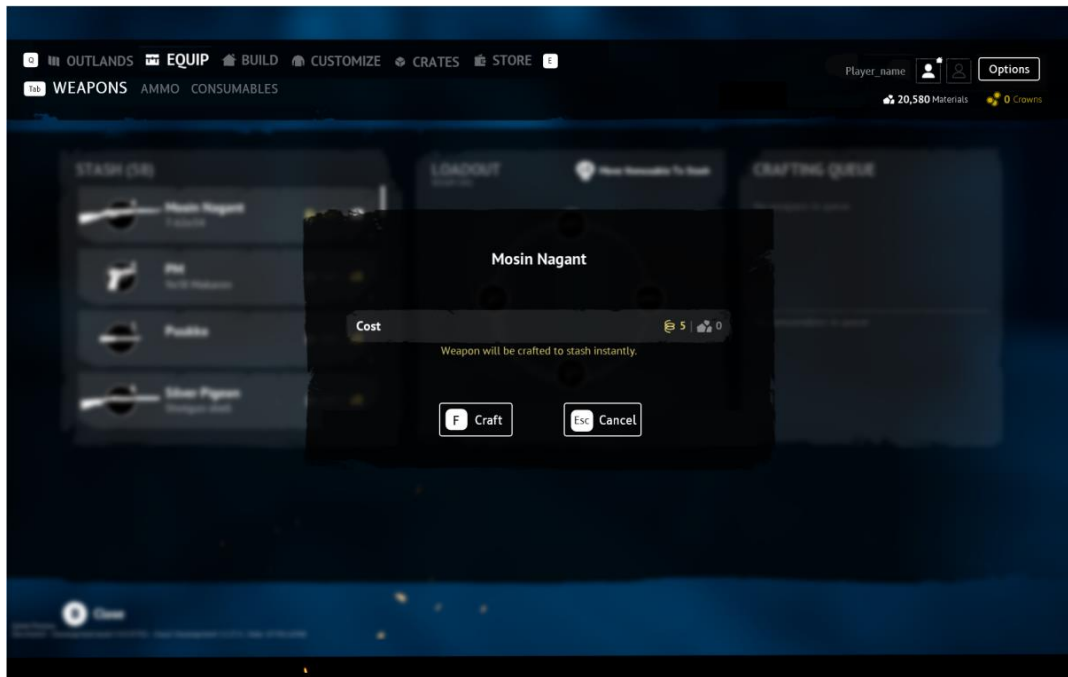


Figure 13. Confirmation window with clickable prompts

There were several instances, where there was no information displayed on focus, in which case a clearly interactable button was chosen as the way to initiate an action, as was established beforehand in the first rule. The entirety of the interaction in these cases are done through the primary mouse button.

### 5.3.2 Options menu

The options menu originally featured the categories as sub tabs, which was changed into a simple selection list (Figure 14). This is in line with other similar instances that can be seen in the shelter menu as well (Appendix 1 and 2). Within the different option categories, the interactions are very straightforward and are primarily contain sliders and check boxes. These can be comfortably navigated and interacted with through the use of a mouse. Therefore, they do not require any significant changes in terms of UI and UX, aside from the different key prompts displayed at the bottom.

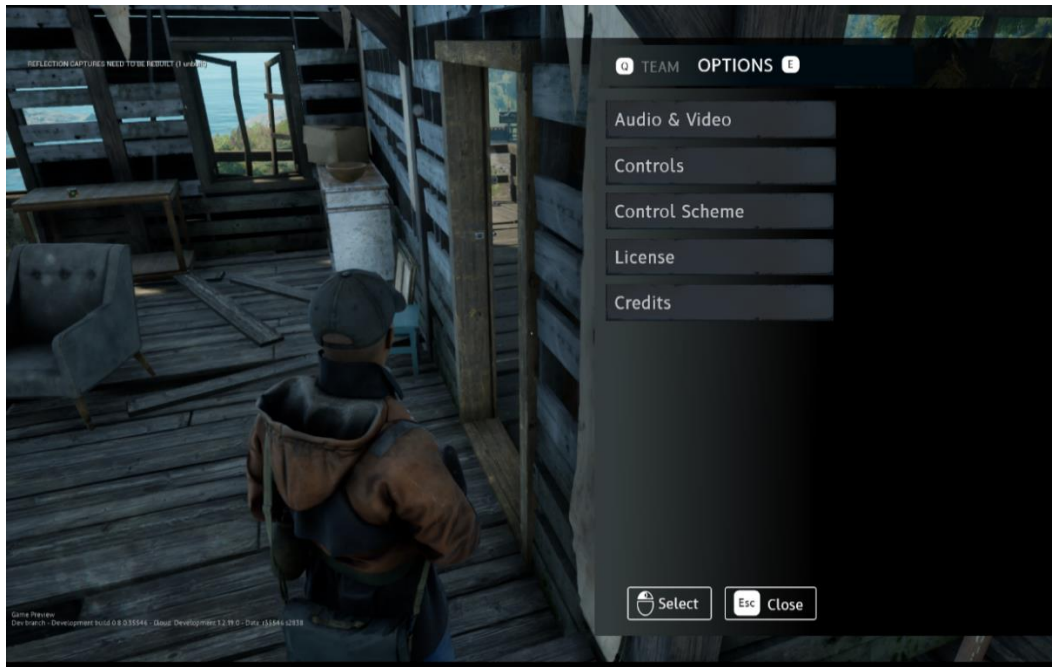


Figure 14. The design for the options menu layout

Two versions of the control scheme screen in the options menu were made. The first one, seen in Figure 15, features a rudimentary visual scheme that displays all the in game controls. The design was based on the currently implemented design, which shows a scheme of a controller with all the possible in game actions aligned with the appropriate input controls. It is important to note here, that the actual propositioned controls in this scheme are not the same as the controls that will be possibly used in the game itself, as the design for this has not been finalized at the time of the thesis.

The second version of this screen, which can be seen in Figure 16, features a list of all the in game actions with assigned controls in a scrollable list. This version was designed with the possibility of allowing key rebinds in the game. For this, a set of screens with different cases and options was designed as well for use in documentation (Appendix 2).

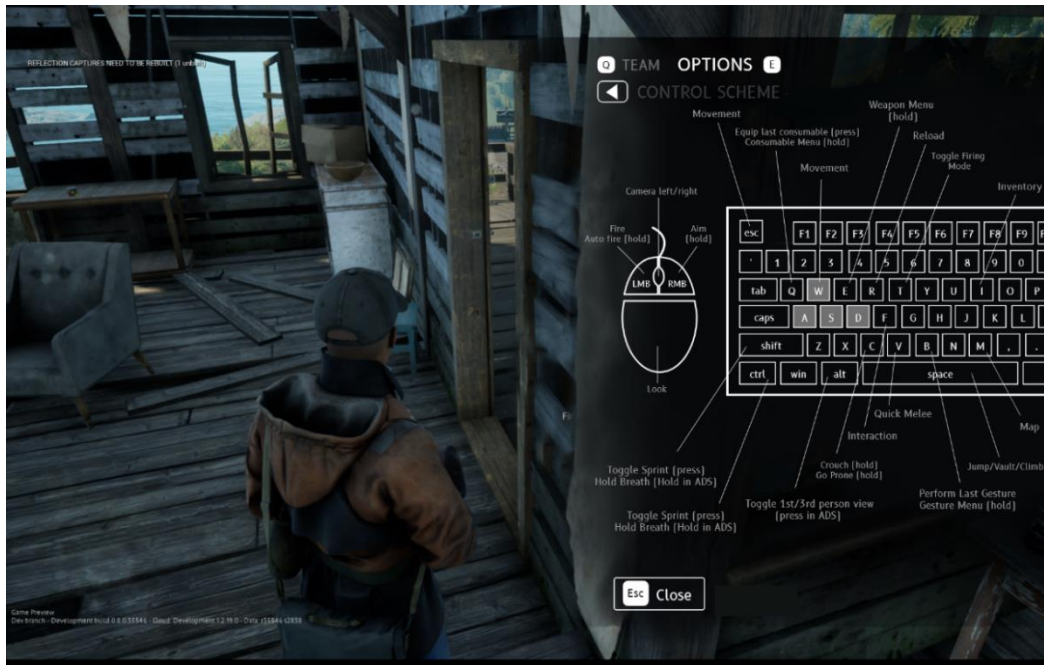


Figure 15. A mock up for the display of the control scheme

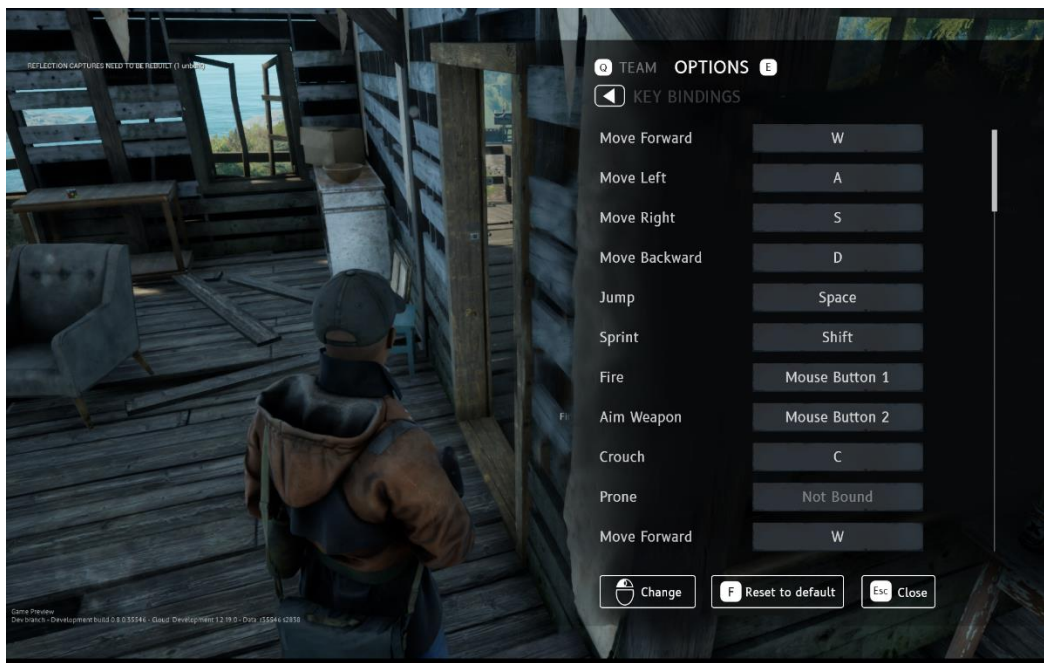


Figure 16. An alternative version of the control scheme design with key rebinding possibilities

A “return” or “back” button was also added for the options menu for the player to be able to return to the overview of the options categories. The same option was also added in the customisation screens in the shelter menu (Appendix 1).

### 5.3.3 Lobby and in game screens

Despite there being no finalized design for the actual in game controls, there were several cases in the actual gameplay that did require revision of the key prompts and controls, in order to carry over the consistency from the main menu and options screens. The lobby screen (Figure 17) is the first area the player sees after initiating an encounter. Here the player can perform several different actions. The main issue here was deciding how to allow the player to most comfortably compare load outs of different players. In a keyboard and mouse input scheme, the most comfortable way of navigation is by the use of the mouse. In this case, however, there are technological limitations that prevent the user to click on the players or their usernames directly, which would be the most natural way to navigate the lobby. Instead, the original design of using the triggers for cycling through the players was carried over from the controller scheme. As this is the same way that tabs are navigated in menus, the decision was made to use the same “Q” and “E” key prompts to cycle through the players, while keeping “F” as the key for toggling the information on and off (Appendix 3).

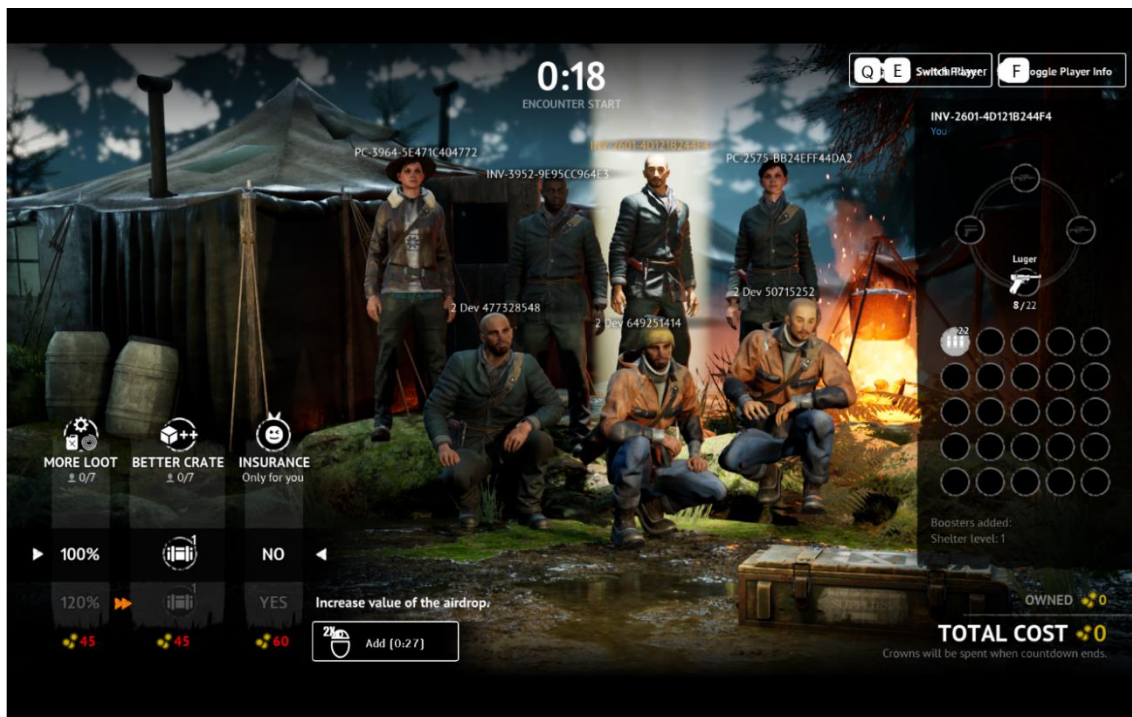


Figure 17. The final design for the lobby screen with the view of the players load out information

The map interaction was another part in game that required UI and UX adaptation (Figure 18). Here, it was most optimal to mimic the way that maps are navigated in different navigation applications (like Google maps) as well as how maps are navigated in other games, and combine it. The map is navigated solely by the use of the mouse, utilizing all three main mouse buttons. The primary mouse button is used to drag and pan across the map. This was deemed most suitable as the maps are small and this is the easiest and most comfortable way. The middle mouse button or scroll wheel is used for zooming and the secondary (right) mouse button is used to mark points on the map.



Figure 18. Final design for the in game map navigation controls

The last in game interaction that needed to be revised for the use of a mouse and keyboard was the actual looting mechanic. Due to the nature of the game, it was important that the interaction is kept as simple as possible to allow the player to react and operate quickly.

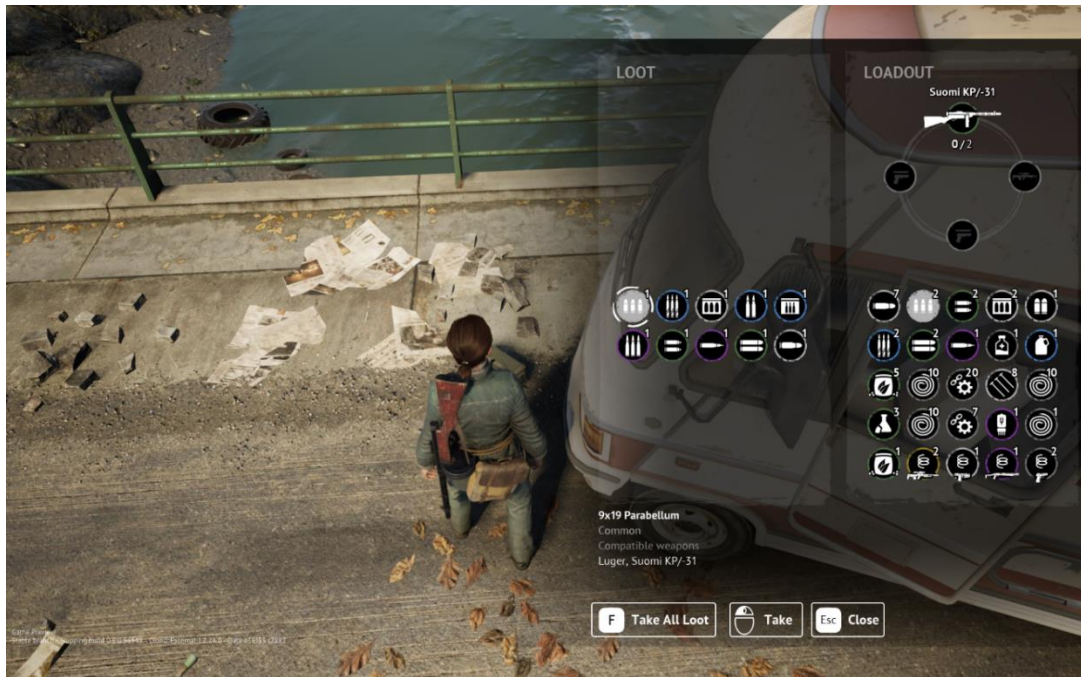


Figure 19. View of the in game looting design scheme

Following the hierarchy of actions and previous options and instances, the “F” key was chosen as the prompt for the primary actions. While not yet determined by the design, it is most likely this will also be the main interaction key in game, following examples from other games. Hence, it made sense for it to be used in this case, as seen in Figure 29. The action of acquiring a single item, as well as removing items from the inventory (Appendix 4) naturally fell to the mouse, as it will be used to also display information on hover. This way the player has the option to quickly process information and react to the UI even when having limited time available. When looting, the player also interacts with their own inventory (Figure 20).



Figure 20. The in game inventory options

In order to allow the player to choose what items they would like to take with them, they were given the option to destroy the ones they do not wish to keep. Even though this would typically be a primary action, the secondary “V” key was picked instead of the “F” key. This was done to accommodate the possible later implementation of the option to inspect the item, and to prevent the possibility of the player to accidentally destroy the item they wanted to keep. The player also has to hold the key rather than press it, to further guarantee any possible mistakes.

## 6 CONCLUSION

This thesis was set to try and determine what the most common problems are that occur when developing games for more than one platform in order to be able to establish guidelines on how to avoid them. The subject of this thesis proved to be broader than initially estimated. There was also only a very narrow pool of references available, which led to somewhat limited research results, hence possibly influencing the actual design of the finished product. Still, the concluded work resulted in a satisfactory solution for the project in the form of a set of rules

and designs. Based on the research and work done, the author suggests that the best approach when designing UI and UX for different platforms is to always keep the differences between them in mind.

Understanding what the individual needs of the users of the different platforms are, helps when making important design decisions that accommodate said needs with only minimal changes required. Optimally, the designer should design for different platforms as early as possible. However, that is often not possible as the scopes and schedules of projects fluctuate and change, and decisions are made later on in the production line. In those cases, the designer should consider the proposed approaches from Sinclair (2018) and, depending on the individual project needs, using and combining them to make a design that is intuitive and comfortable to use on any included platform. The author chose a combination of the *prompts switch* and the *secretly console first* approach with was best suited for the project itself. The actual approach should be chosen on a project by project basis. By always keeping basic usability and design principles mentioned in this thesis at the forefront, achieving a satisfactory UX design for different input schemes should not be an overwhelming task and most common problems should be easily avoided.

In-engine testing was part of the original plan for this thesis work, in order to gather feedback on some proposed UX solutions and polishing of the designs. However, due to the restructuring of the project, changes in schedule and production, it had to be removed from this thesis. Both in-engine testing and focus testing are still planned as part of the implementation of the mouse and keyboard input scheme for the game, but due to the aforementioned changes, it will be done at a later date and will not be included in this work.

Changes in production were always a considered risk for this thesis, and while testing of the mock up prototypes would have been preferred and would have further contributed to the designs, it does not take away from the finished

designs. Rather, the findings help lay out a useful foundation of principles for future multi-platform designs and projects.

## REFERENCES

Bernhaupt, R. (ed.) 2010. Evaluating user experience in games: Concepts and methods. London: Springer.

*Destiny 2*. 2017. PC [Video game]. Bungie: Bellevue.

*Dishonored 2*. 2016. PC [Video game]. Arkane Studios: Lyon.

Fagerholz & Lorenzon. 2009. Beyond the HUD -- User interfaces for increased player immersion in FPS Games. Online publication. Available at:

<http://publications.lib.chalmers.se/records/fulltext/111921.pdf>

[Accessed 15 January 2019]

Hodent, C. 16 March 2015. Developing UX practices at Epic Games. WWW site. Available at: <https://celiahodent.com/ux-practices-epic-games/>

[Accessed 11 January 2019]

Hodent, C. 29 July 2016. The elusive power of video games for education. WWW site. Available at: <https://celiahodent.com/the-elusive-power-of-video-games-for-education/>

[Accessed 11 January 2019]

Hodent, C. 13 August 2017. The gamer's brain: How neuroscience and UX can impact design. Youtube video. Available at: <https://youtu.be/XlpDLa585ao>

[Accessed 11 January 2019]

ISO 9241-11 :en. 2018. Ergonomics of human-system interaction. Terms and definitions. Part 11: Usability: Definitions and concepts.

Lidwell, W., Holden, K. & Butler, J. 2003. Universal principles of design: a cross disciplinary reference. Gloucester: Rockport Publishers.

Krug, S. 2014. Don't make me think, revisited: A common sense approach to web usability. Berkeley: New Riders.

Long, S. 2 October 2017. What is games user experience (UX and how does it help? WWW site. Available at:

[https://www.gamasutra.com/blogs/SebastianLong/20171002/306649/What\\_Is\\_Games\\_User\\_Experience\\_UX\\_and\\_How\\_Does\\_It\\_Help.php](https://www.gamasutra.com/blogs/SebastianLong/20171002/306649/What_Is_Games_User_Experience_UX_and_How_Does_It_Help.php)

[Accessed 10 January 2019].

Nagalingam, V. & Ibrahim, R. 2015. User experience of educational games: A review of the elements. *Procedia Computer Science* 72, 423–433.

Nielsen, J. 4 January 2012. Usability 101: Introduction to Usability. WWW site.

Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

[Accessed 13 February 2019]

Norman, D. 2013. The design of everyday things: Revised and expanded edition. New York: Basic Books.

Sinclair, D. 18 October 2018. Secretly console first: A better approach to multi-platform game UI design. WWW site. Available at:

[http://gamasutra.com/view/news/328311/Secretly\\_console\\_first\\_A\\_better\\_approach\\_to\\_multiplatform\\_game\\_UI\\_design.php](http://gamasutra.com/view/news/328311/Secretly_console_first_A_better_approach_to_multiplatform_game_UI_design.php)

[Accessed 9 January 2019].

Tandon, O. 30 August 2018. Game UX style Guide: Why do you need one?.

WWW site. Available at:

[http://www.gamasutra.com/blogs/OmTandon/20180830/325522/Game\\_UX\\_Style\\_Guide\\_Why\\_do\\_you\\_need\\_one.php](http://www.gamasutra.com/blogs/OmTandon/20180830/325522/Game_UX_Style_Guide_Why_do_you_need_one.php)

[Accessed 10 January 2019].

*The Witcher 3: Wild Hunt*. 2015. PC [Video game]. CD Projekt: Warsaw.

Vriend, S. 28 August 2017. Game design and gestalt laws. WWW site. Available at:

[https://www.gamasutra.com/blogs/SitaVriend/20170828/304420/Game\\_design\\_and\\_Gestalt\\_laws.php](https://www.gamasutra.com/blogs/SitaVriend/20170828/304420/Game_design_and_Gestalt_laws.php)

[Accessed 15 January 2019].

## LIST OF FIGURES

Title of game (edition or version). Year. Platform [Game]. Publisher: Place of publication.

Figure 1. The proximity of the “!” icon indicates that it provides information the chosen hero in *Overwatch*. *Overwatch*. 2016. PC [Video game]. Blizzard Entertainment: Irvine.

Figure 2. The brain has enough information to assume what the missing pieces are in the board game *Saboteur*. *Saboteur*. 2004. [Board game]. AMIGO: Dietzenbach.

Figure 3. The enclosure of the character icon and text in *Undertale* implicates they belong together. *Undertale*. 2015. PC [Video game]. Toby Fox.

Figure 4. Dayz Logo is an example of using the Figure and Ground principle to add interest in the form of negative space. *DayZ*. 2013. PC [Video game]. Bohemia Interactive: Prague.

Figure 5. The menu interactions in *Destiny 2* with controller button prompts. *Destiny 2*. 2017. PC [Game]. Bungie: Bellevue.

Figure 6. The PC version of the menu follows the button prompt switch approach. *Destiny 2*. 2017. PC [Game]. Bungie: Bellevue.

Figure 7. The variety of tabs and options in *Fortnite* can be easily navigated with a mouse. Sinclair, D. 18 October 2018. Secretly console first: A better approach to multi-platform game UI design. WWW site. Available at: [http://gamasutra.com/view/news/328311/Secretly\\_console\\_first\\_A\\_better\\_approach\\_to\\_multiplatform\\_game\\_UI\\_design.php](http://gamasutra.com/view/news/328311/Secretly_console_first_A_better_approach_to_multiplatform_game_UI_design.php) [Accessed 9 January 2019].

Figure 8. The console version of *Fortnite* is learnable but has some suboptimal UX features like double focus of tabs. Sinclair, D. 18 October 2018. Secretly console first: A better approach to multi-platform game UI design. WWW site. Available at: [http://gamasutra.com/view/news/328311/Secretly\\_console\\_first\\_A\\_better\\_approach\\_to\\_multiplatform\\_game\\_UI\\_design.php](http://gamasutra.com/view/news/328311/Secretly_console_first_A_better_approach_to_multiplatform_game_UI_design.php) [Accessed 9 January 2019].

Figure 9. No key prompts on the PC version of the game as it is navigated and interacted with a mouse. *Overwatch*. 2016. PC [Video game]. Blizzard Entertainment: Irvine.

Figure 10. Key prompts are displayed when using a controller for the “select and commit” method in *Overwatch*. *Overwatch*. 2016. PC [Video game]. Blizzard Entertainment: Irvine.

Figure 11. The original equip screen from *Vigor* with the controller key prompts. *Vigor*. 2018. Xbox One [Video game]. Bohemia Interactive: Prague.

Figure 12. The final mock up design for the equip screen.

Figure 13. Confirmation window with clickable prompts.

Figure 14. The design for the options menu layout.

Figure 15. A mock up for the display of the control scheme.

Figure 16. And alternative version of the control scheme design with key rebinding possibilities.

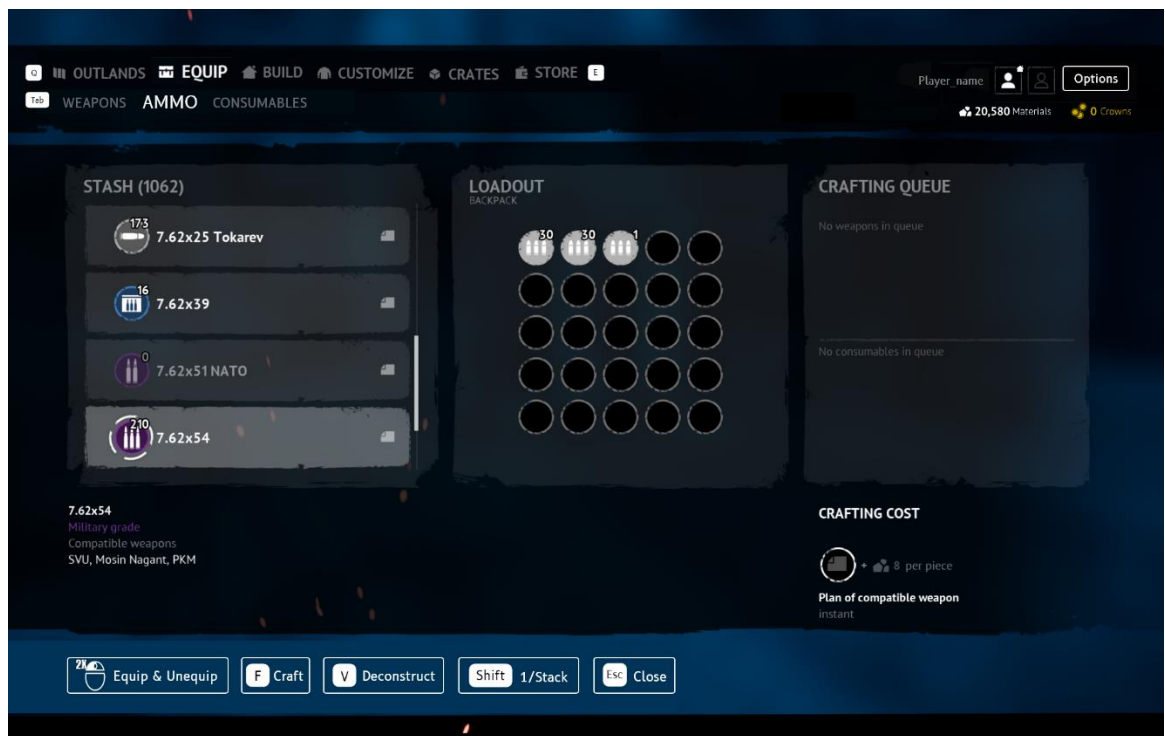
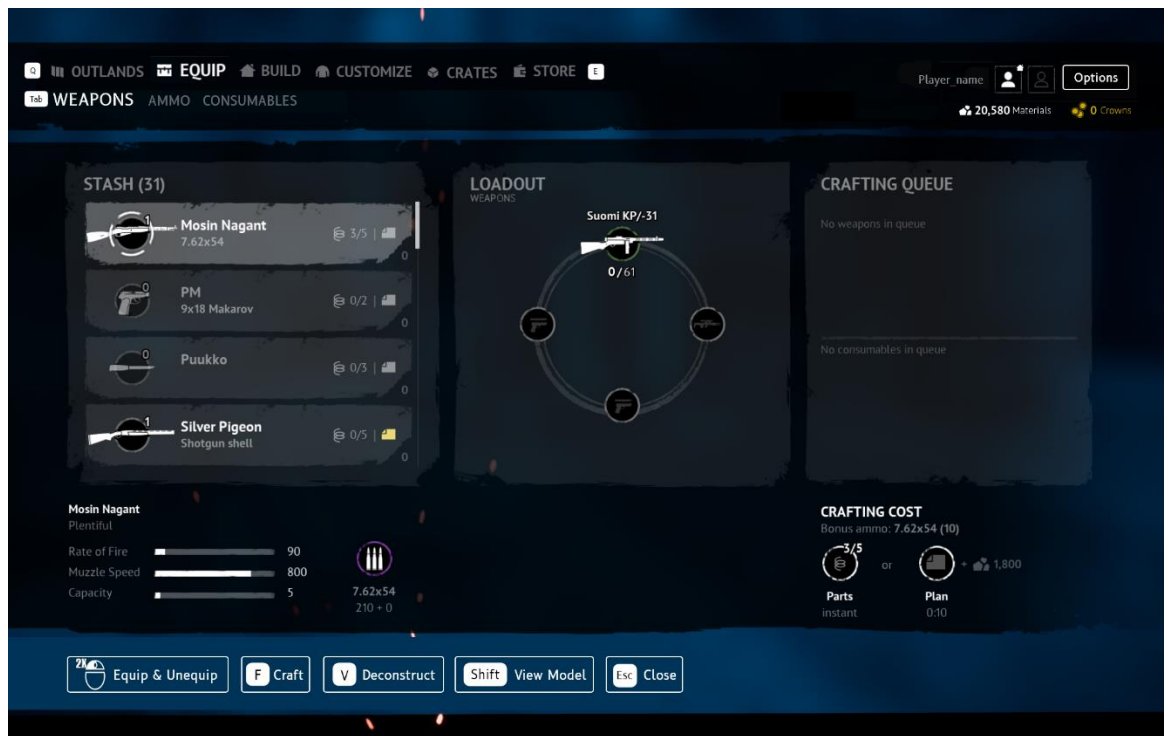
Figure 17. The final design for the lobby screen with the view of the players load out information.

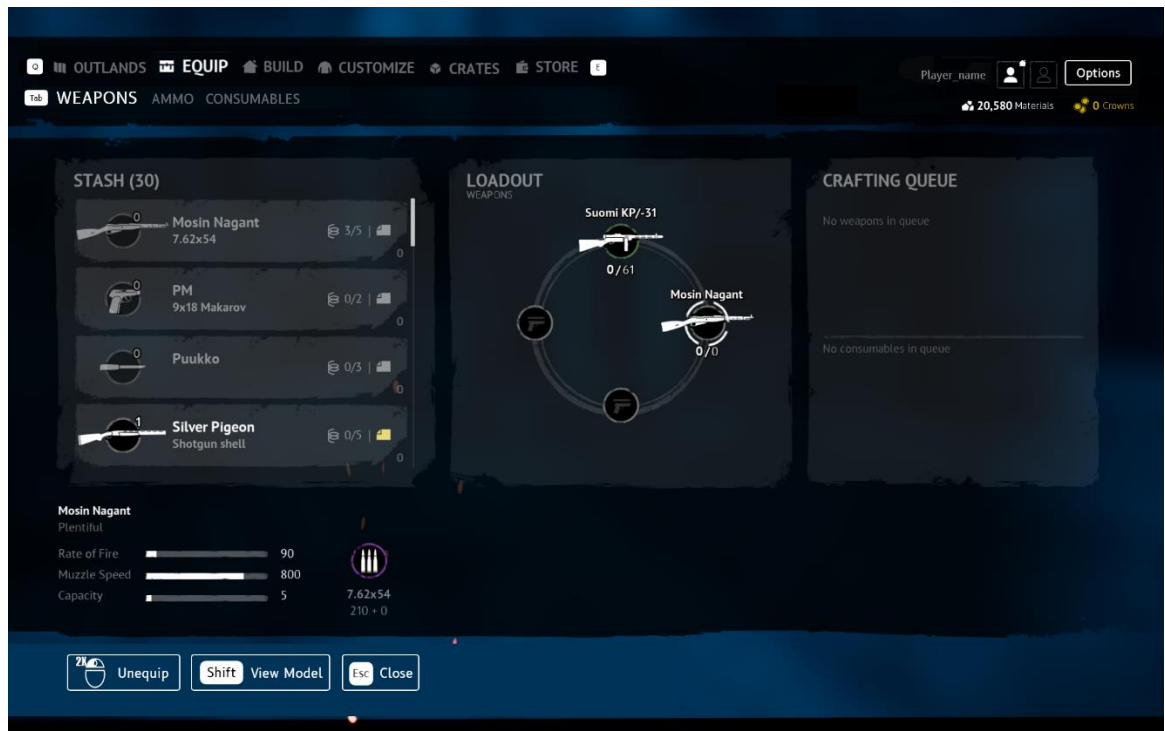
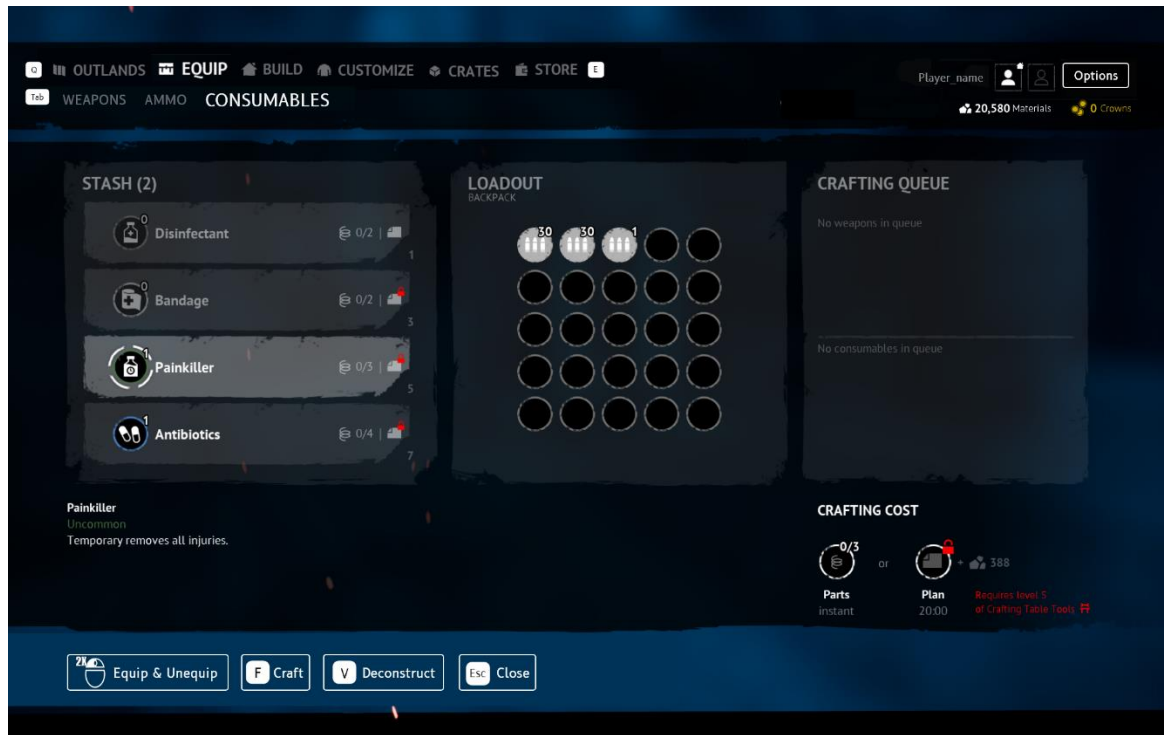
Figure 18. Final design for the in game map navigation controls.

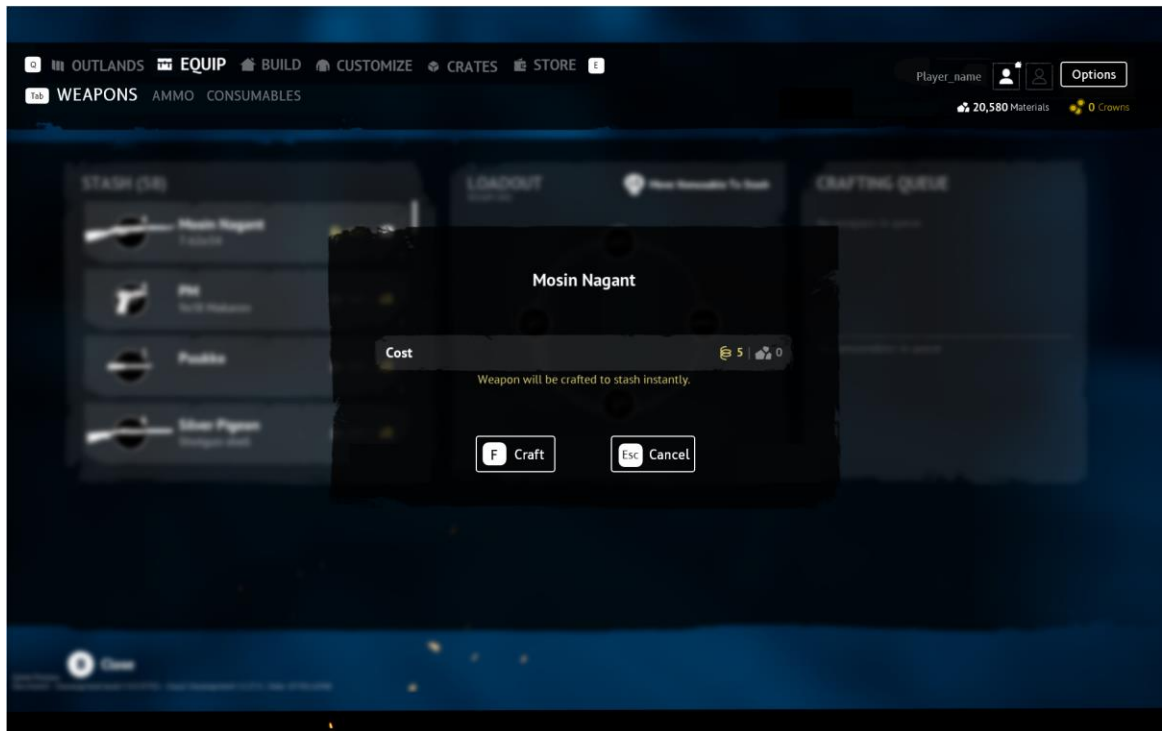
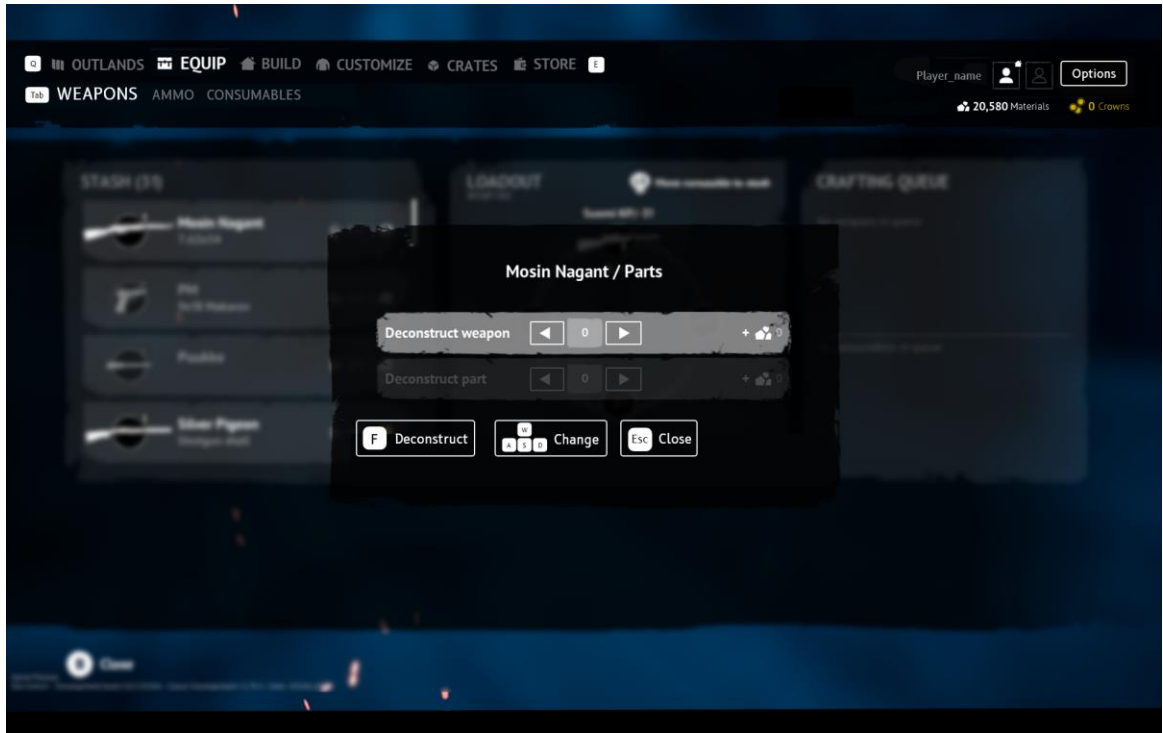
Figure 19. View of the in game looting design scheme.

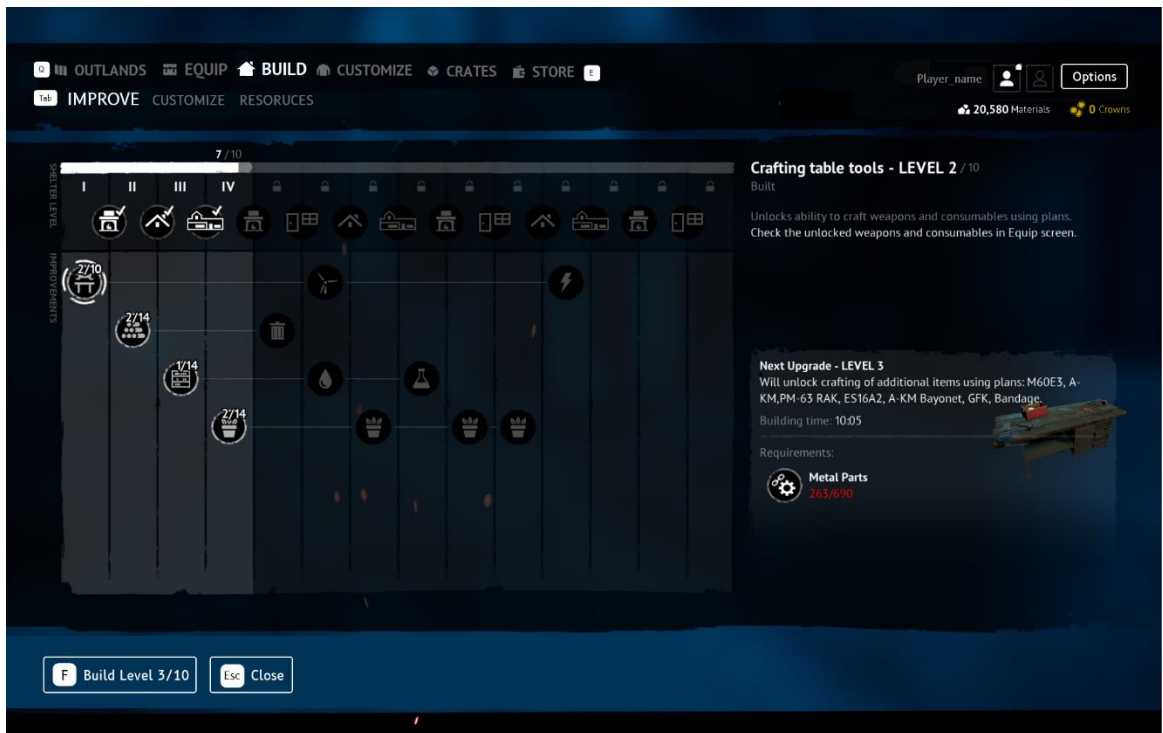
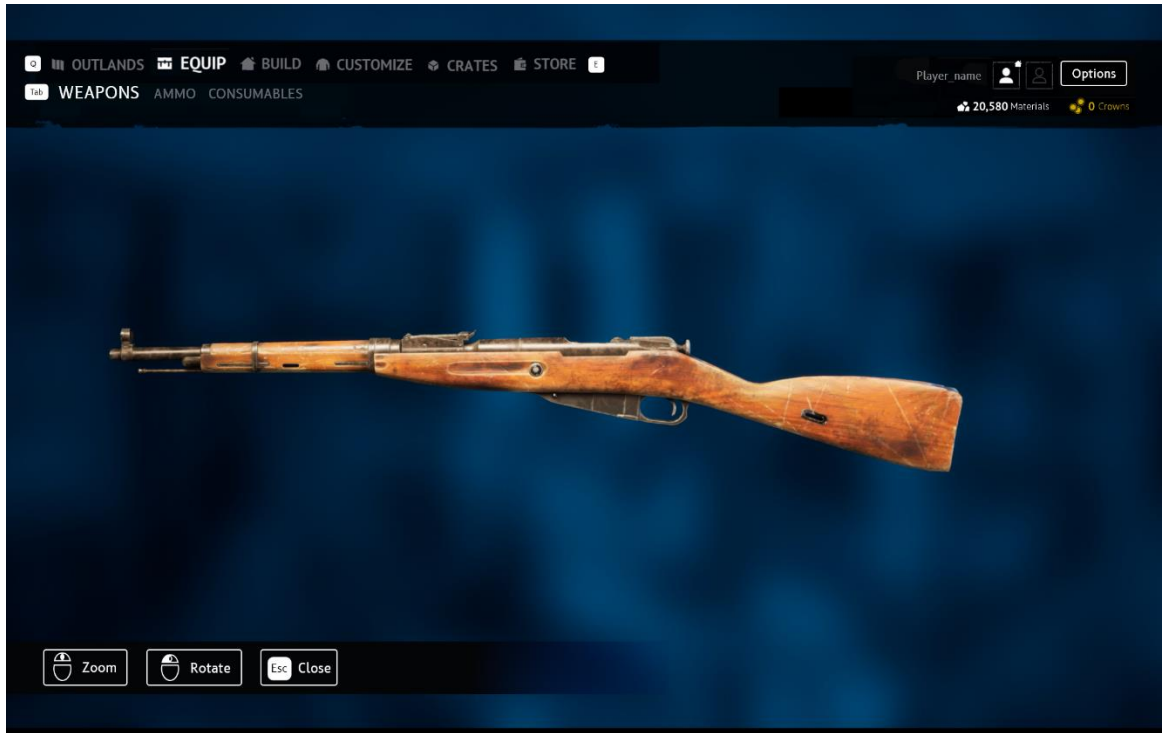
Figure 20. The in game inventory options

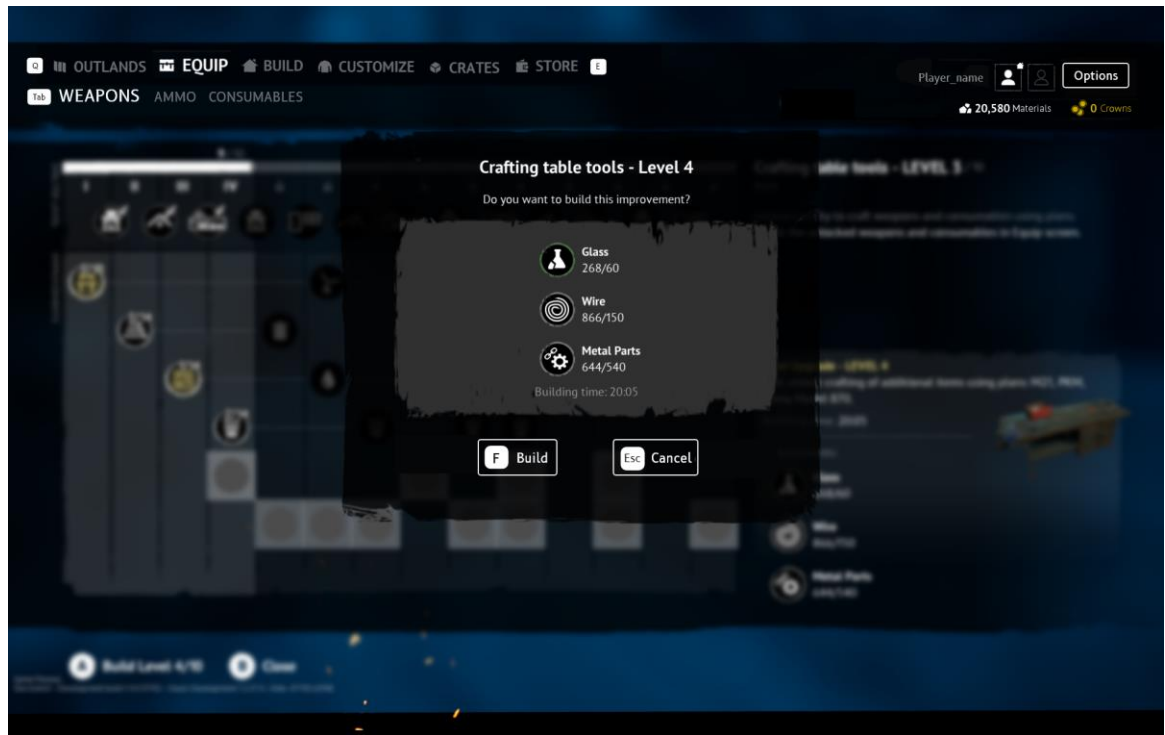
# SHELTER MENU MOCK UPS

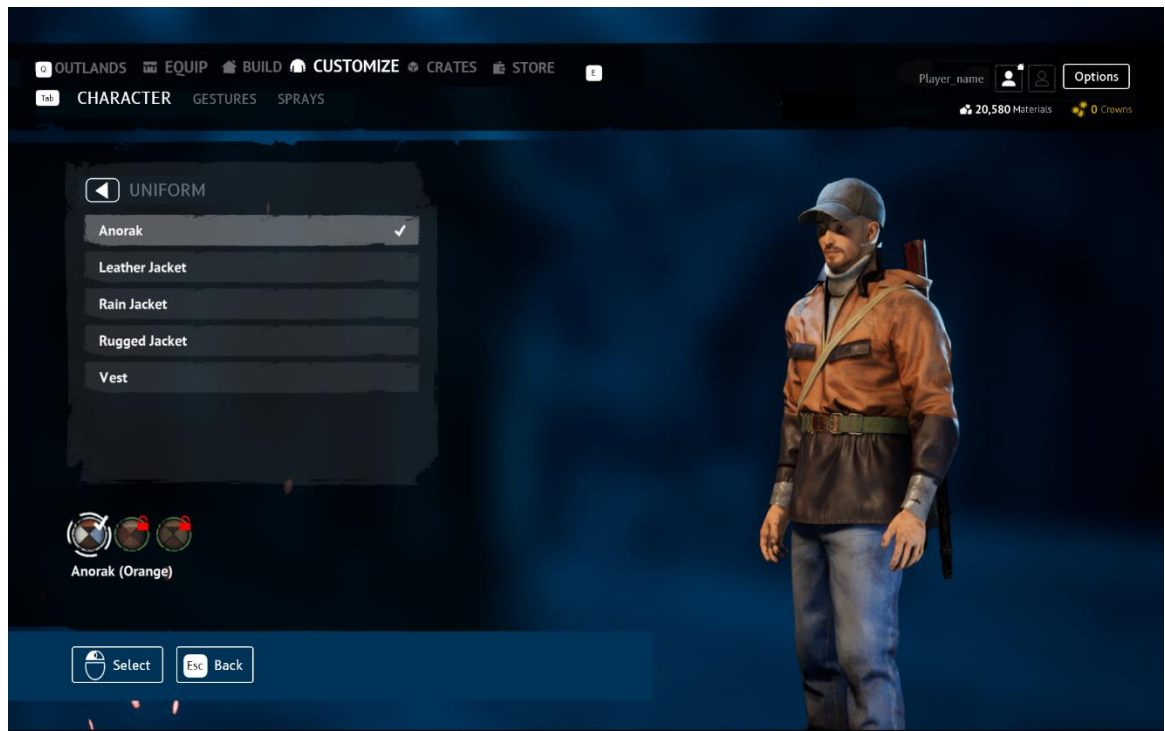


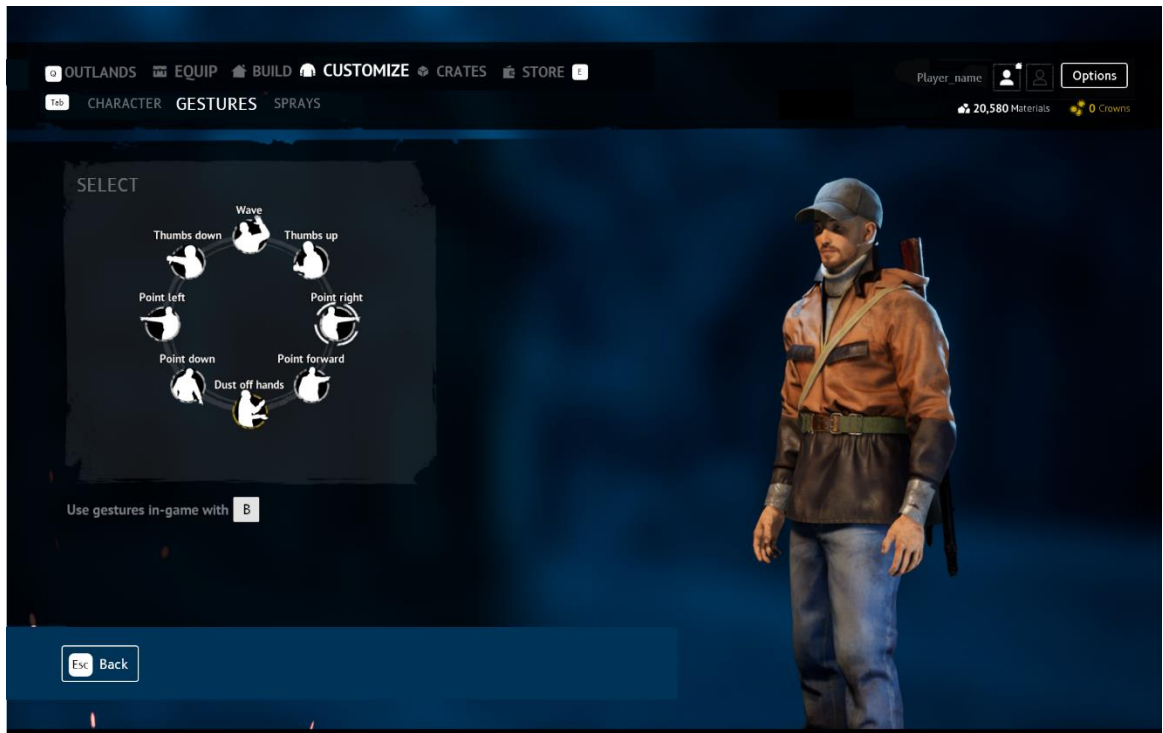


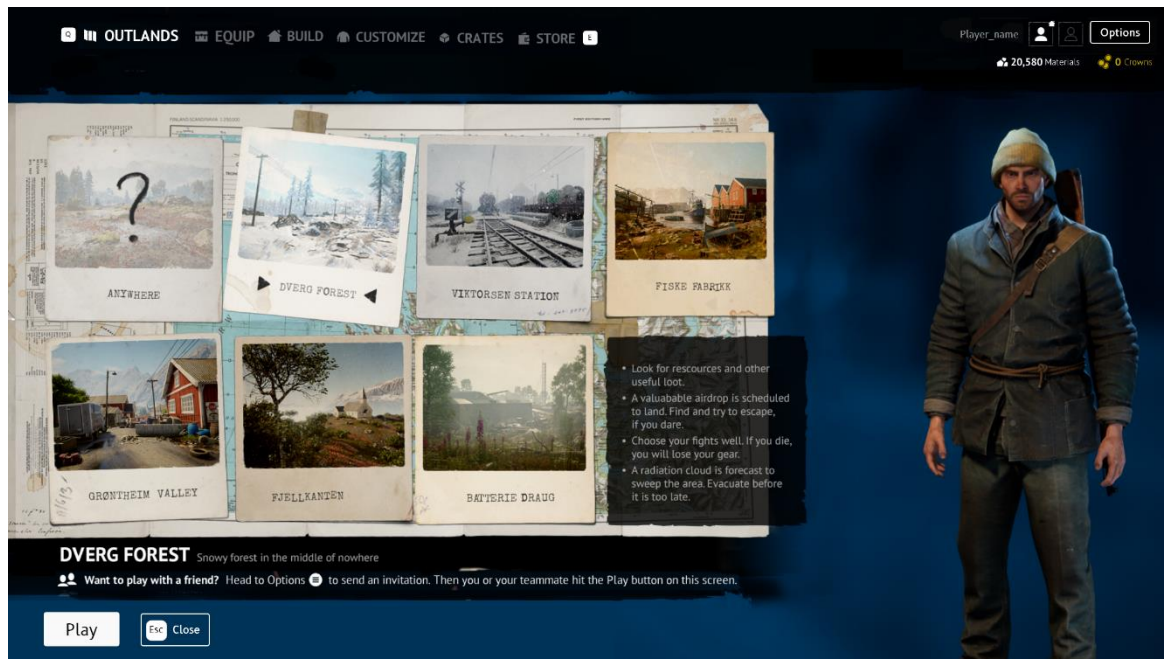
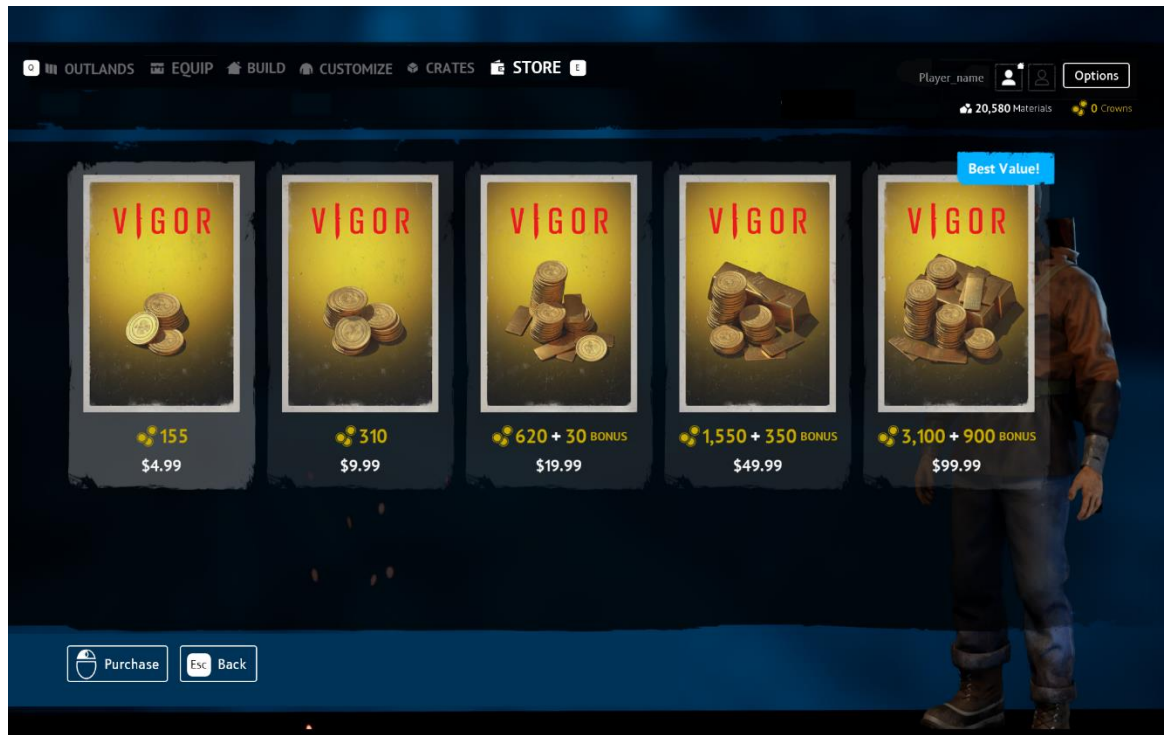




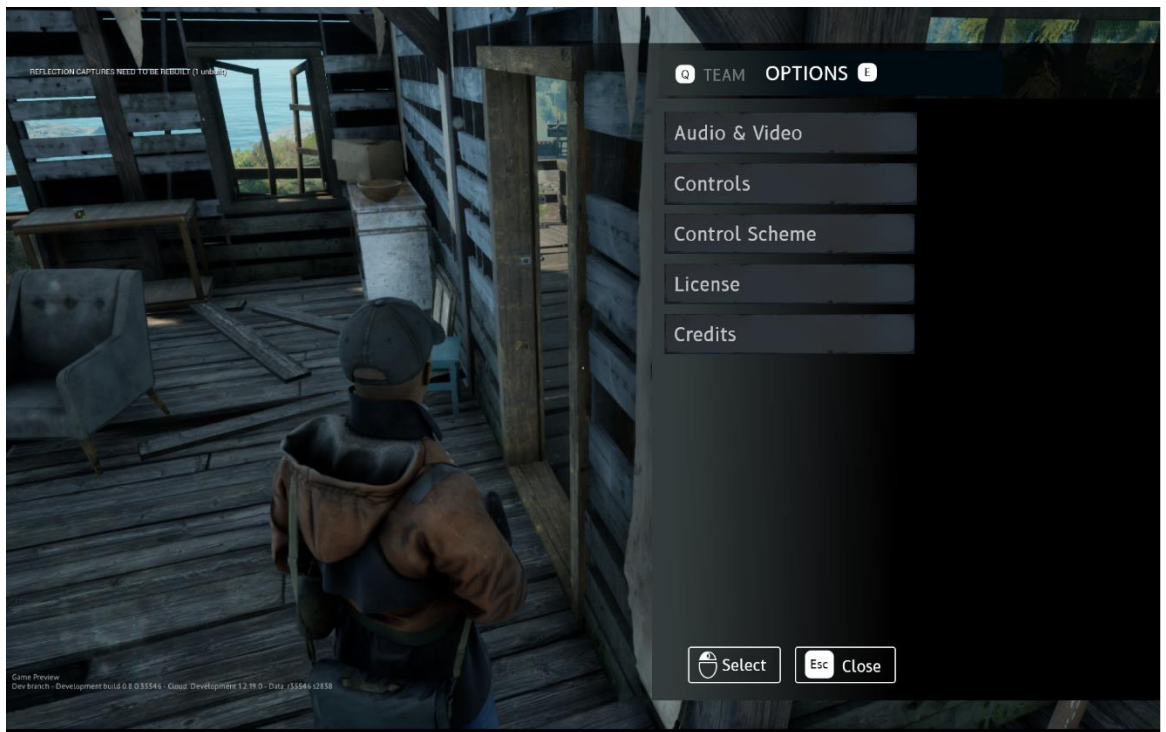
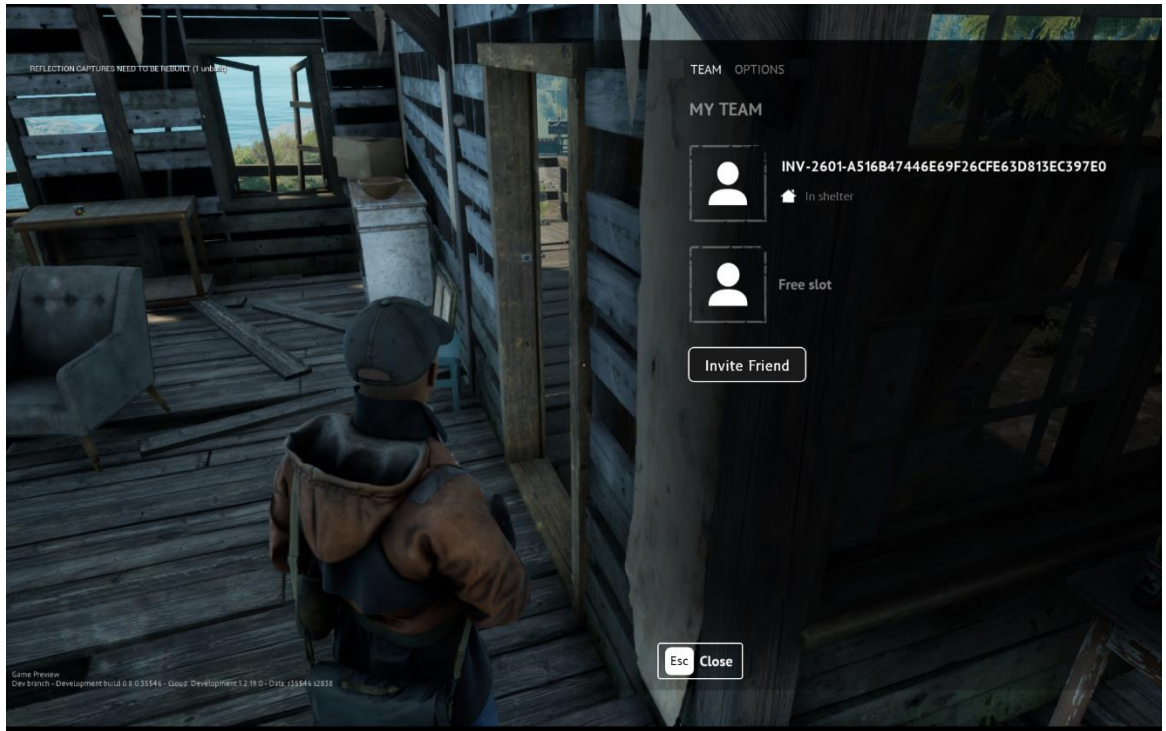


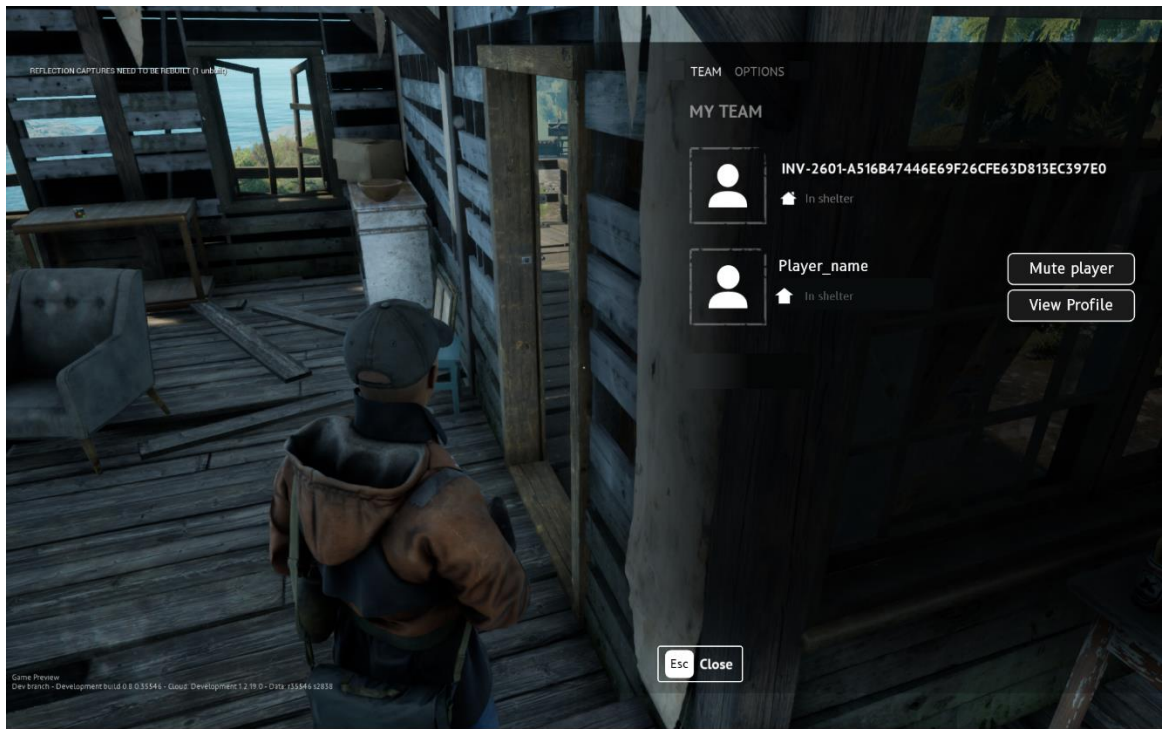
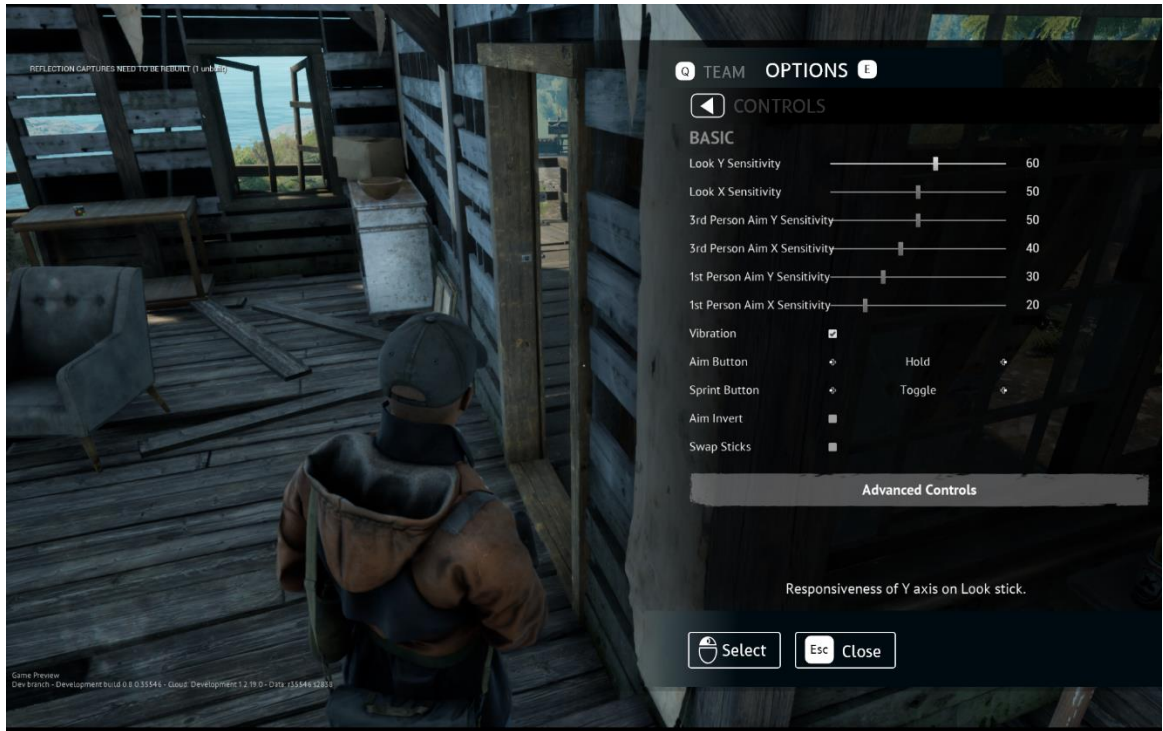


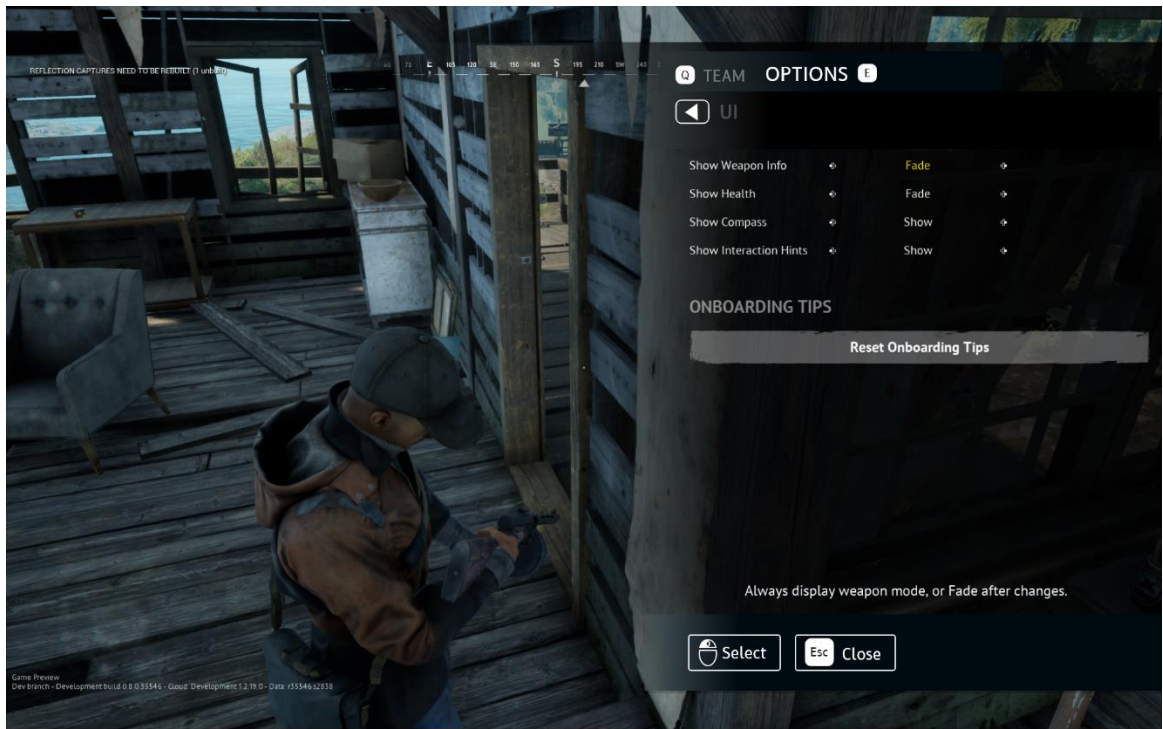
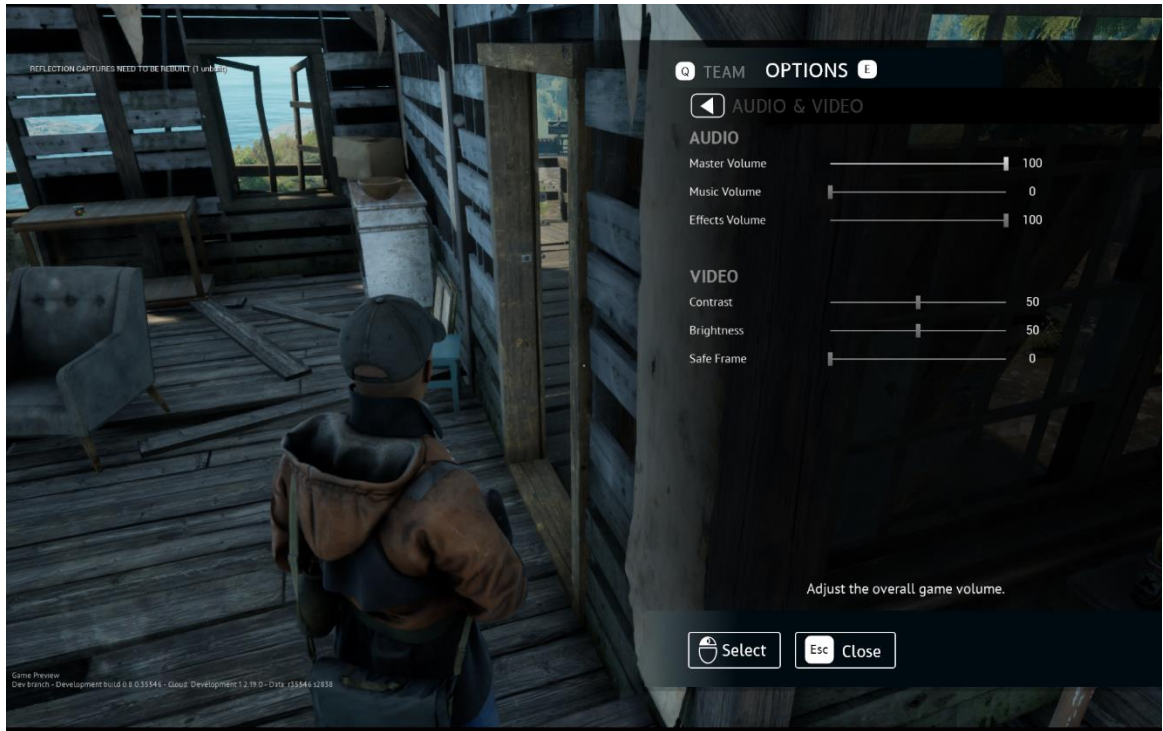


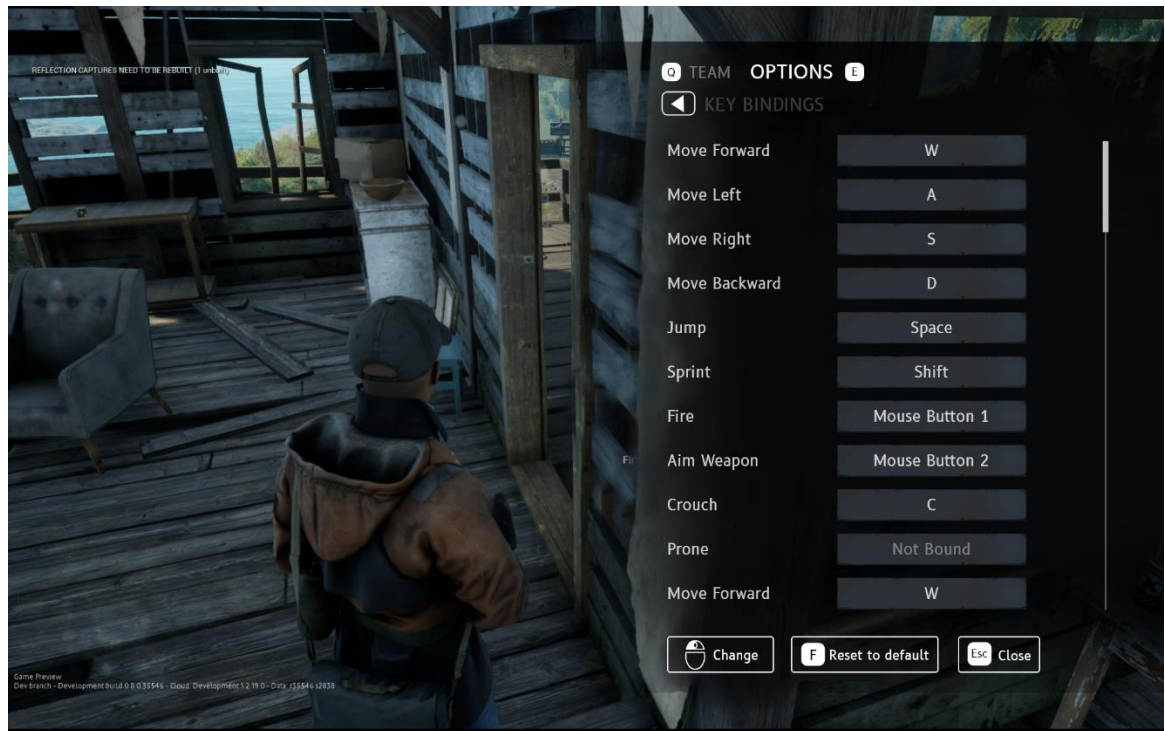
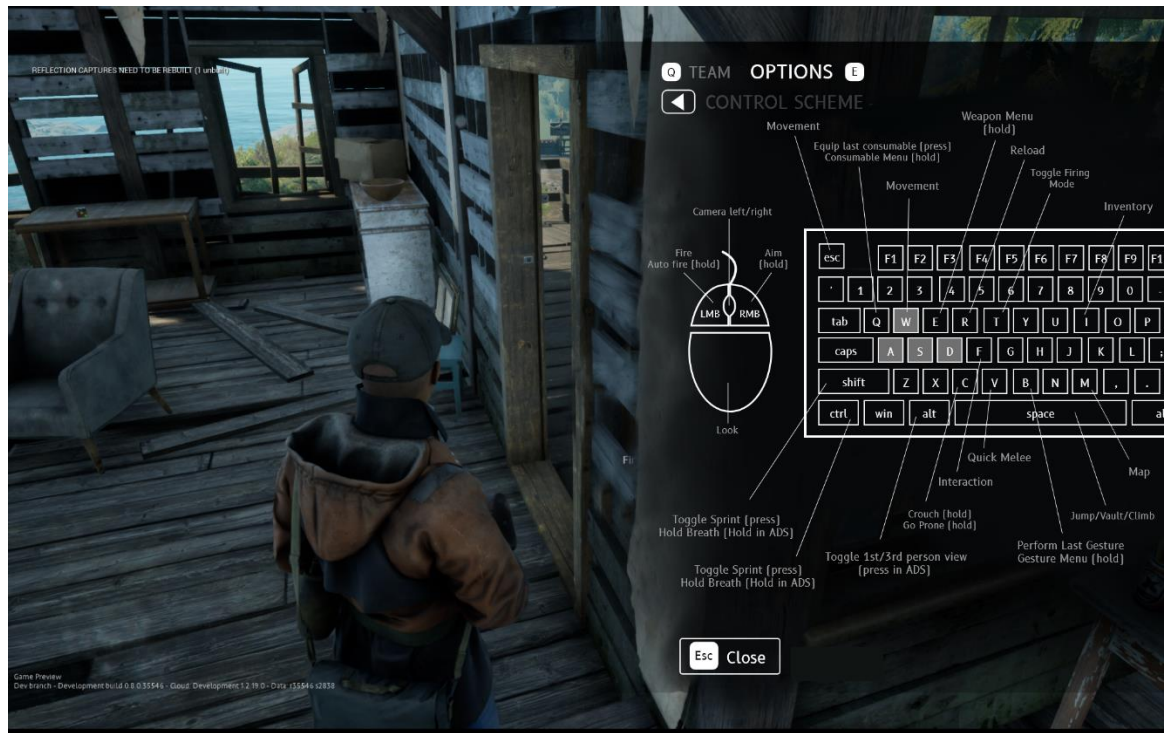


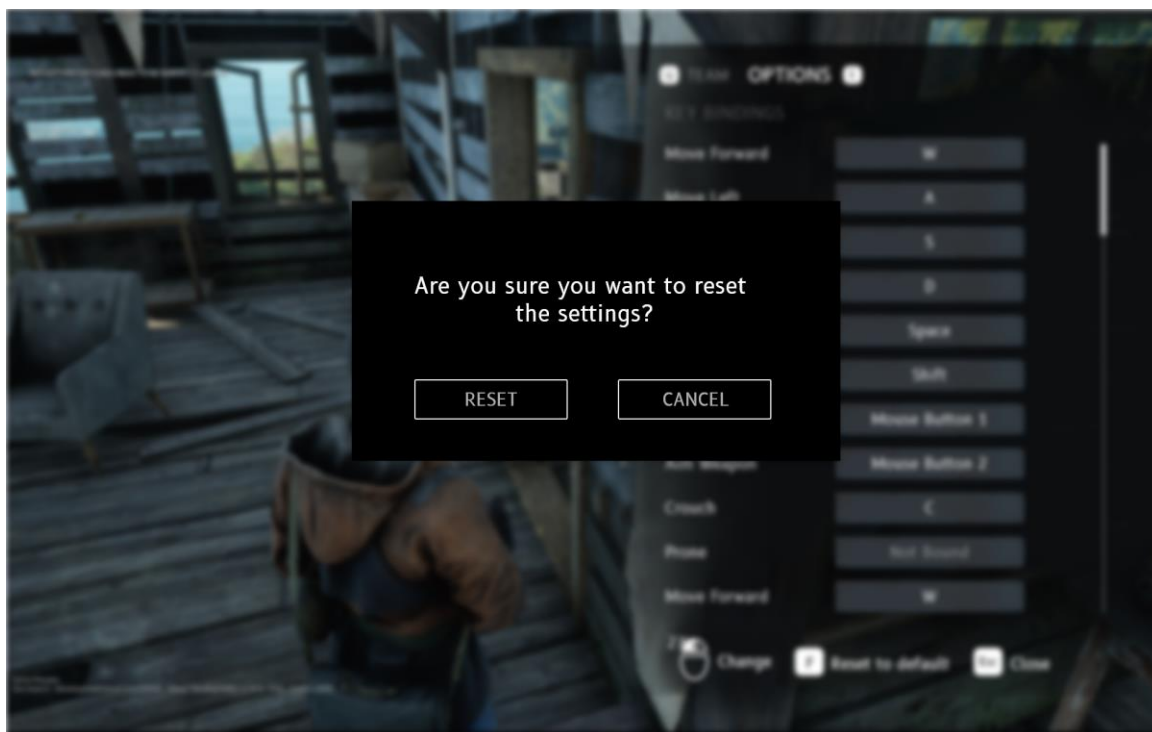
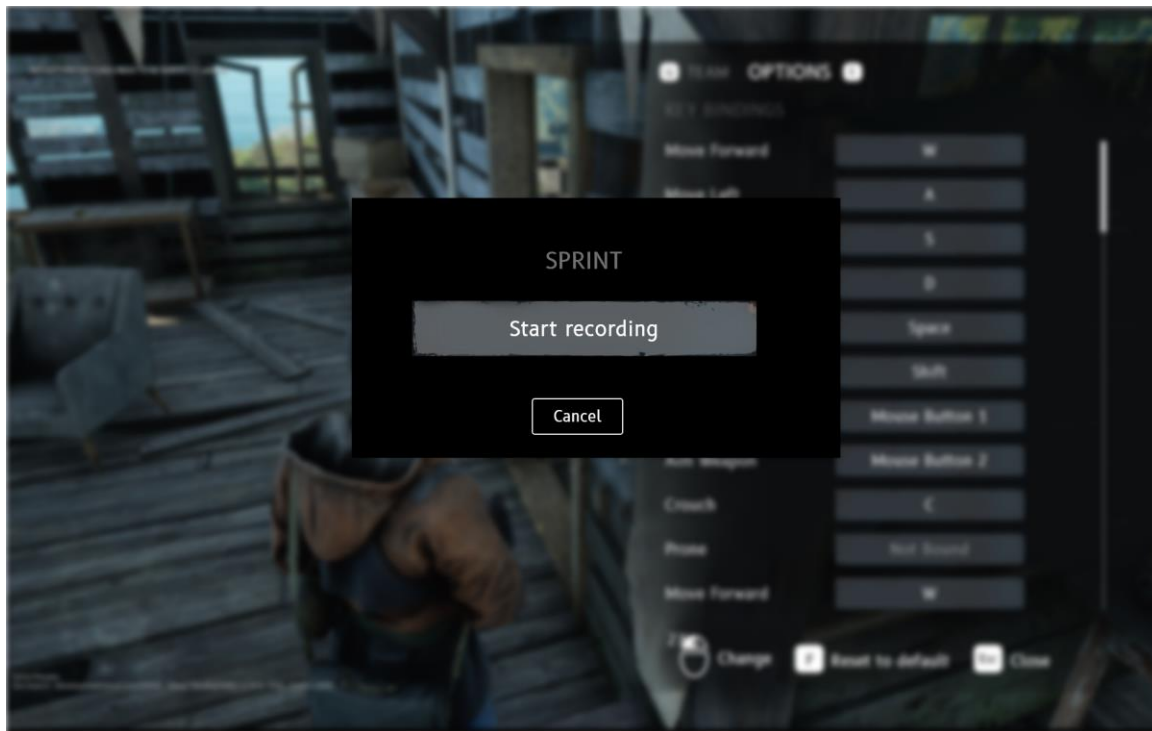
## OPTIONS MENU MOCK UPS

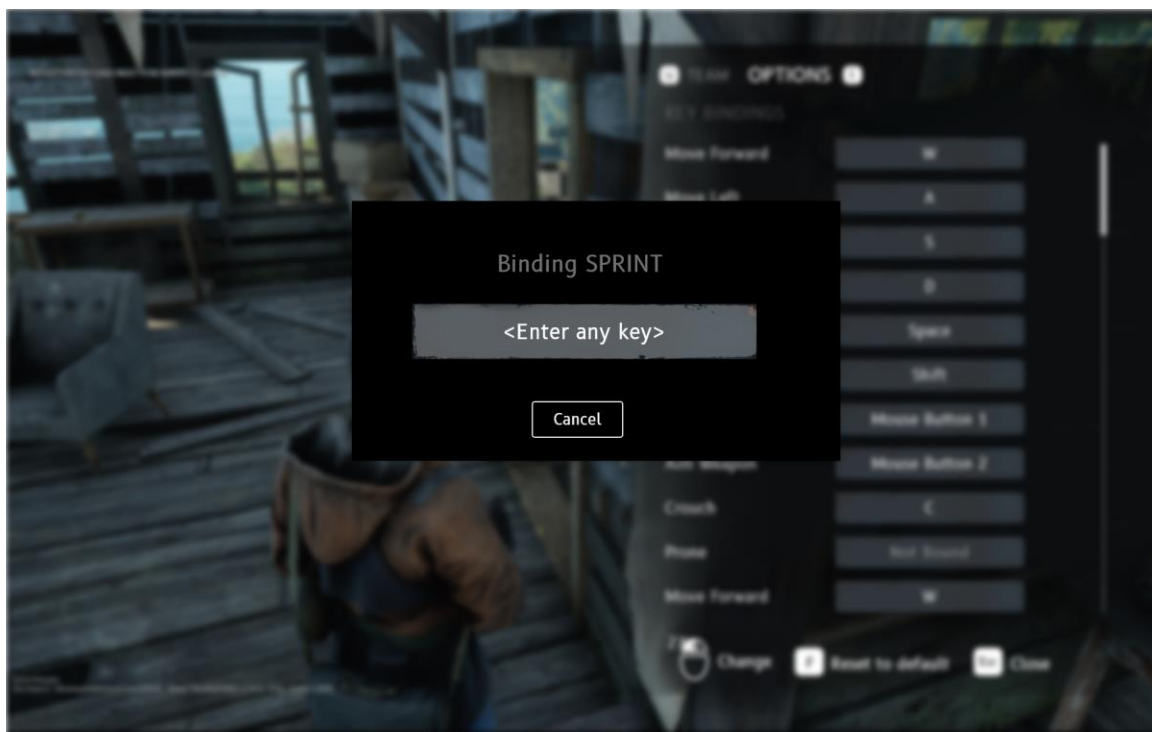
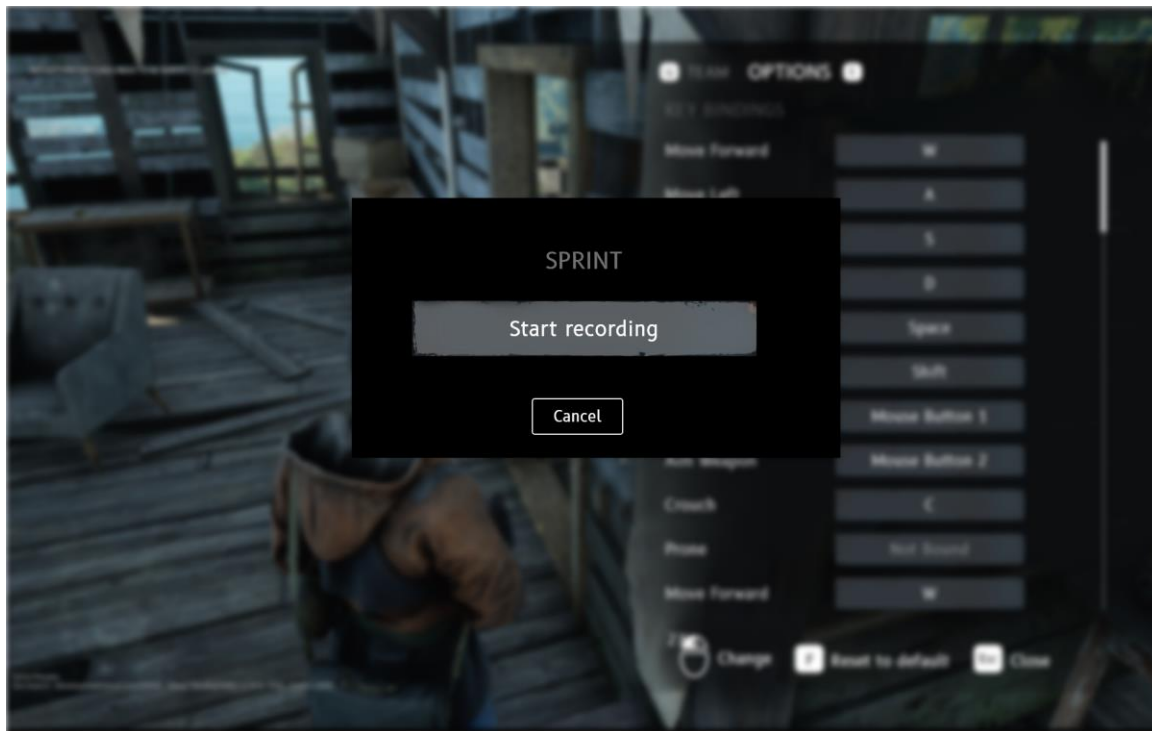


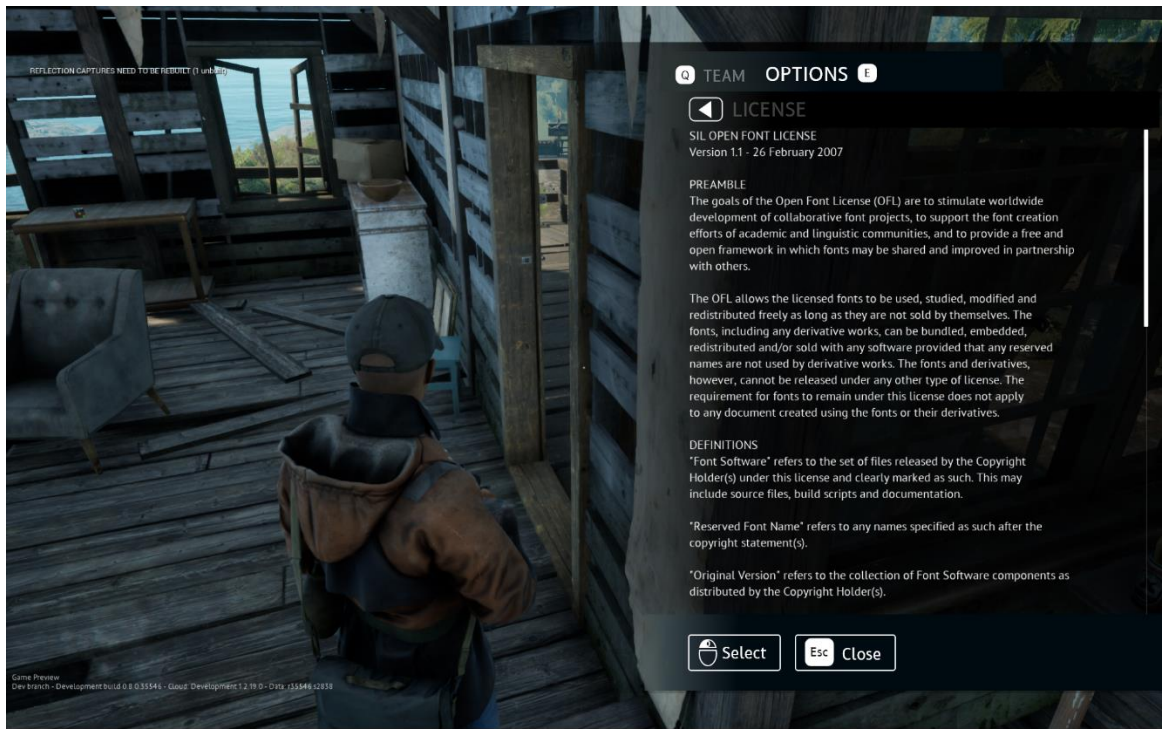
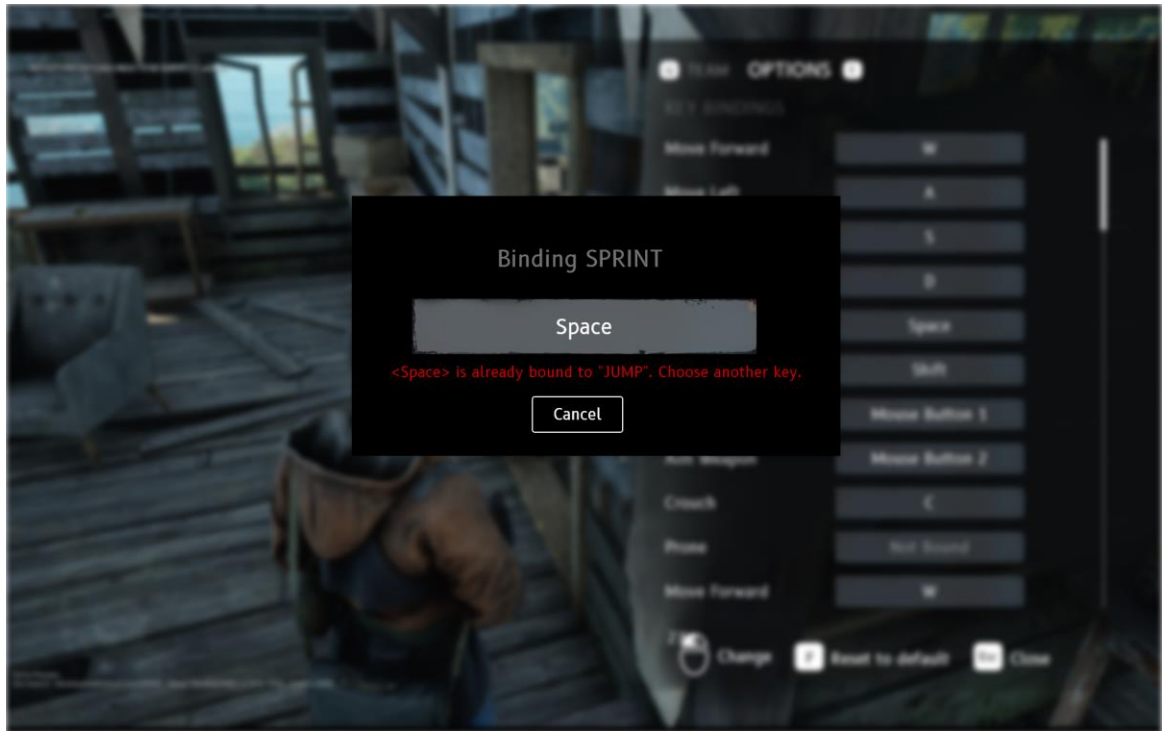




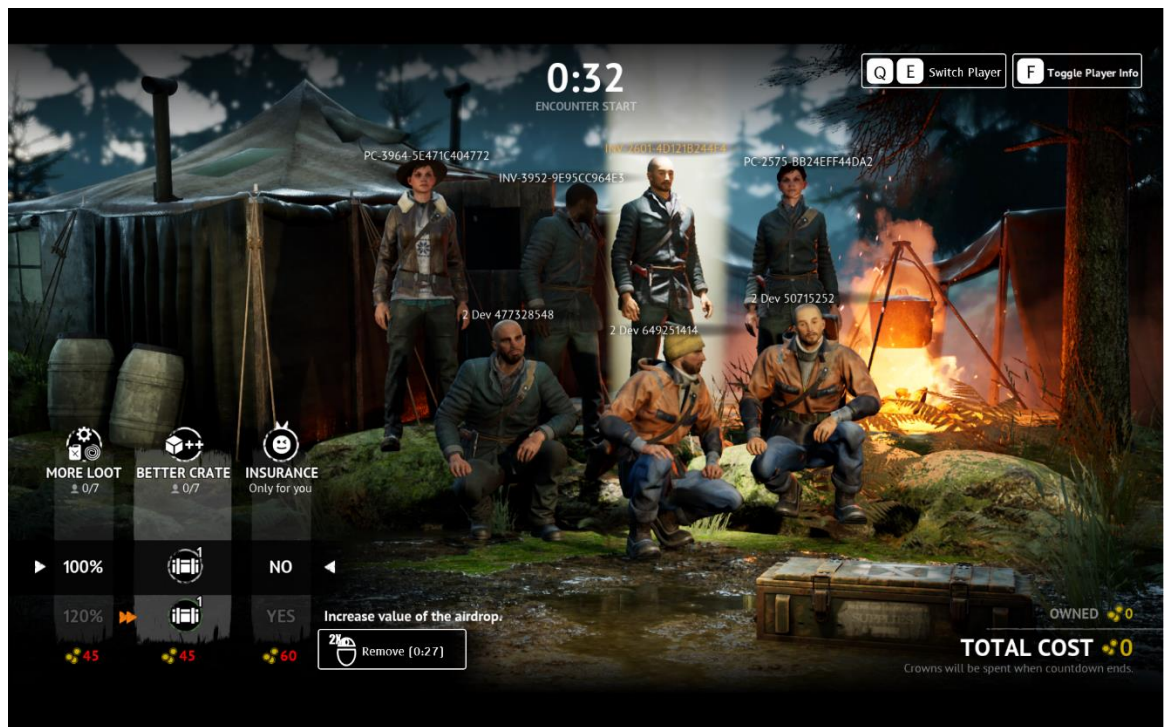
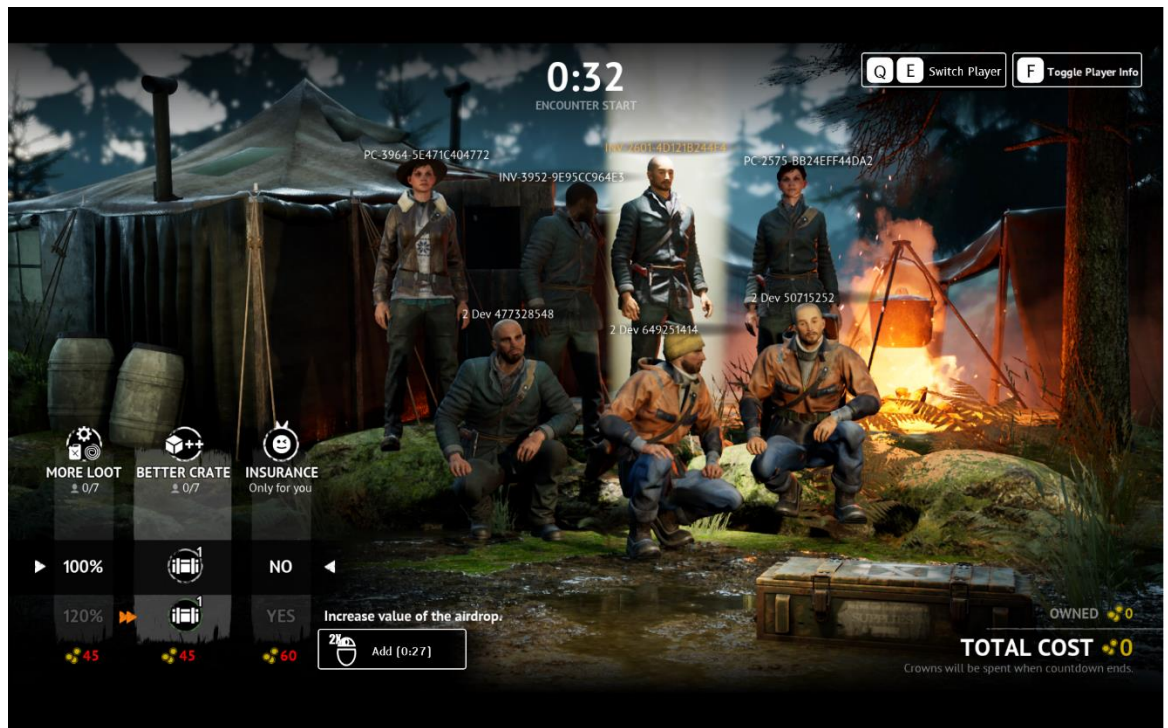


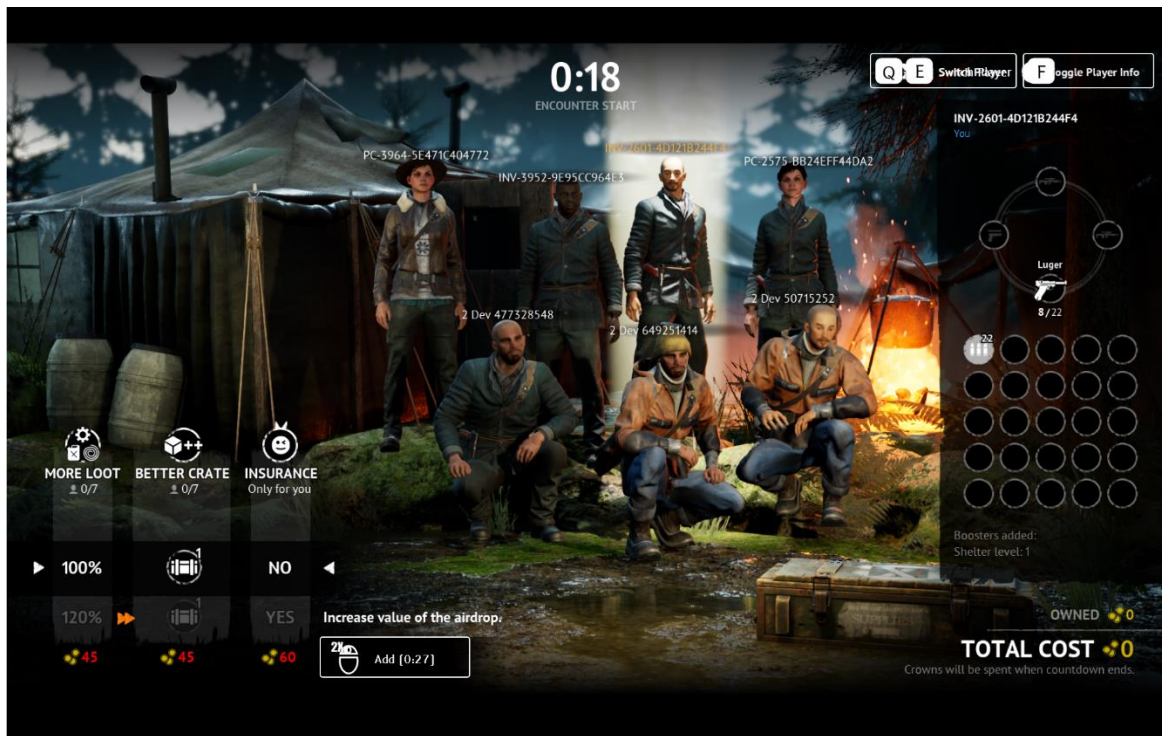
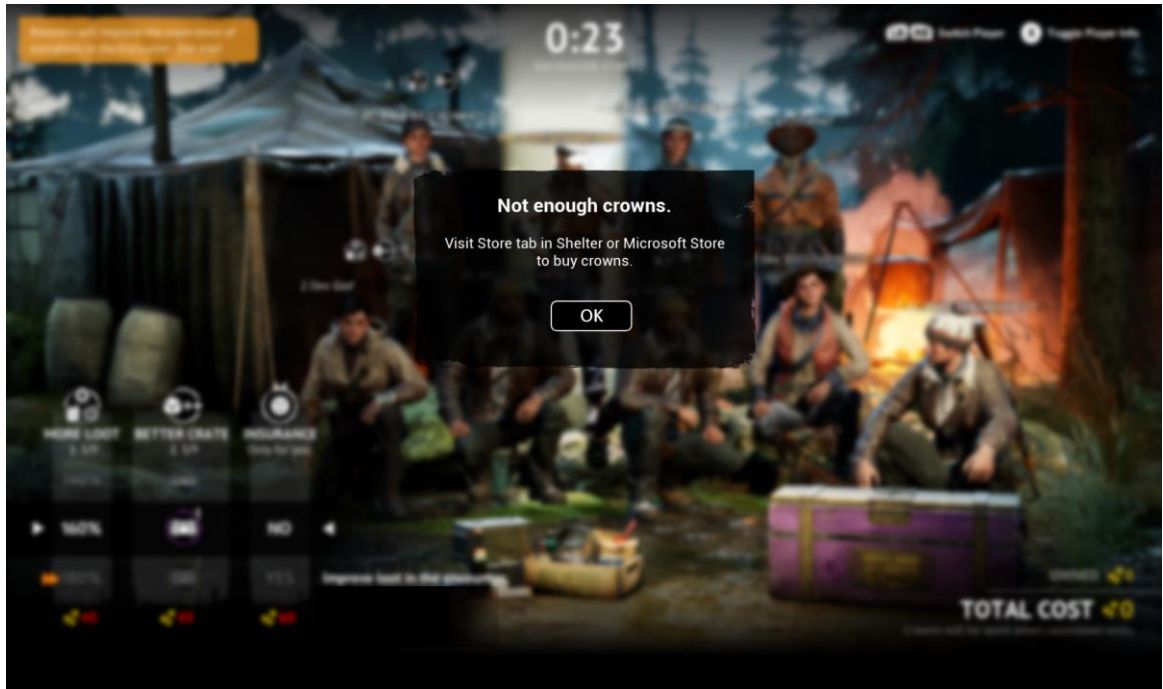






# LOBBY SCREEN MOCK UPS





# IN GAME SCREEN MOCK UPS



