

Master's Thesis

**User Interface and User Experience
for a Mobile Game**

Vincent Fischer

16.04.2021

Hochschule Harz

Fachbereich Automatisierung und Informatik

Medien- und Spielekonzeption

- 1. Prüfer:** Prof. Daniel Ackermann
2. Prüfer: Prof. Dominik Wilhelm

**Thema und Aufgabenstellung der Masterarbeit
MA AI 99/2020**

für Vincent Fischer

**User Interface and User Experience
for a Mobile Game**

Over the course of the last two semesters the mobile game "RPG-Trader" has grown from a rough concept into a functional experience. The core gameplay loop and many of the essential systems have been conceptualized and implemented. Some of what has been implemented is not yet satisfactory in terms of user experience. Furthermore, there are central parts of the user interface, like the main menu, that are not yet implemented.

Object of this master thesis is the improvement of existing features and completion of the game's user interface to reach the alpha stage in development. Strong emphasis will be put on the application of user experience design methodology to make sure that the interaction with the game's interface is efficient, satisfying, easy to learn and enjoyable overall. The work encompasses the complete design process and the final implementation in Unity using the programming language C#.

The master thesis contains the following subtasks:

- Analysis of the current state
- User experience methodology research
- Redesign and optimization of existing features
- Design and implementation of new features
- Evaluation of the results



Prof. Daniel Ackermann
1. Prüfer



Prof. Dominik Wilhelm
2. Prüfer

Table of Contents

1. Introduction	11
2. User Experience Design	13
2.1 User Experience Design	14
2.2 The Seven Factors Model	14
2.2.1 Factor 1: Usefulness	14
2.2.2 Factor 2: Usability	14
2.2.3 Factor 3: Findability	14
2.2.4 Factor 4: Credibility	14
2.2.5 Factor 5: Desirability	14
2.2.6 Factor 6: Accessibility	15
2.2.7 Factor 7: Value	15
2.3 User Centered Design	15
2.4 UX Design Methodology	16
2.4.1 Card Sorting	16
2.4.2 Expert Review	16
2.4.3 Eye Movement Tracking	16
2.4.4 Field Studies	17
2.4.5 Usability Testing	17
2.4.6 User Personas	17
2.4.7 Value Proposition.	17
2.4.8 Heuristic Evaluation	18
2.4.9 Stakeholder Interviews	18
2.4.10 User Interview	18
2.4.11 Focus Group	19
2.4.12 Customer/User Journey Map.	19
2.4.13 Competitive Audit	19
2.4.14 Kano Model	19
2.4.15 Information Architecture	20
2.4.16 Key Performance Indicators	20
2.4.17 Product Requirement Document	20
2.5 Approach	21
2.5.1 Initial State	21
2.5.2 Confining Parameters.	21
2.5.3 Prior Work	22

2.5.4	Selected Methodology	24
3.	UX Requirements	27
3.1	Product Requirements	28
3.1.1	General Description	28
3.1.2	Goal	28
3.1.3	Users	28
3.1.4	Opportunity	28
3.1.5	User Interface Components	29
3.1.6	User Requirements	30
3.1.7	Technical Features	30
3.1.8	Monetization	30
3.1.9	Business Requirements	30
3.2	Key Performance Indicators	31
3.2.1	Operational Measurements	31
3.2.2	User Experience KPIs	32
4.	Evaluation	35
4.1	Usability Testing	36
4.1.1	Usability Testing Methodology	36
4.1.2	Planning the Usability Testing	37
4.1.3	Usability Testing Preparation	39
4.1.4	Execution and Results	40
4.2	Usability Analysis	50
4.2.1	The 10 Usability Heuristics	50
4.2.2	Map View Usability Analysis	51
4.2.3	Trade Menu Usability Analysis	54
5.	Design	59
5.1	Design Process	60
5.1.1	Iteration of existing Features	60
5.1.2	Design of new Features	60
5.2	Map View Iteration	62
5.2.1	Travel Flow Redesign	63
5.2.2	Map Object Rework	63
5.2.3	Status Text Field	64
5.2.4	Route Display and Camera Controls	64
5.2.5	Center on Player Function	65
5.2.6	Overlap Prevention	65

5.2.7	Information Window Rework	68
5.2.8	Map Object Icon Rework	69
5.2.9	Button and Toggle Rework	70
5.3	Trade Menu Iteration	71
5.3.1	Value Difference Indication	71
5.3.2	Offer Success Chance Improvements	73
5.3.3	Bartering Chances Indication	74
5.3.4	Adventurer Bar Adjustment	75
5.3.5	Alert and Info Dialogue Improvement	75
5.4	Ambush Interface Design	75
5.4.1	Ambush Interface Flow	75
5.4.2	Ambush Interface Wireframing	76
5.5	Progression Menu Design	77
5.5.1	Progression Menu Flow	78
5.5.2	Progression Menu Wireframing	80
5.5.3	Progression Menu Assets	80
5.6	Intro Sequence and Main Menu Design	82
5.6.1	Intro Sequence	82
5.6.2	Main Menu	82
6.	Implementation	85
6.1	Asset Export/Import	86
6.2	Assembly	86
6.3	Animation	86
6.4	Scripting	87
6.5	Bug Fixing	87
7.	Evaluation of the Results	89
7.1	Improvement	90
7.2	Completion	90
7.3	Summary	90
8.	References	92

List of Figures

Figure 1:	RPG-Trader - Initial State 1, Screenshot taken on OnePlus 6 Smartphone	21
Figure 2:	RPG-Trader - Initial State 2, Screenshot taken on OnePlus 6 Smartphone	22
Figure 3:	Usability Testing Session 1 Analysis Document, Part 1	42
Figure 4:	Session 1 Analysis Document, Part 2	43
Figure 5:	Usability Testing Session 1 Analysis Document, Part 3	44
Figure 6:	Usability Testing Session 1 Analysis Document, Part 4	45
Figure 7:	Usability Testing Session 2 Analysis Document, Part 1	46
Figure 8:	Usability Testing Session 2 Analysis Document, Part 2	47
Figure 9:	Usability Testing Session 2 Analysis Document, Part 3	48
Figure 10:	Usability Testing Session 2 Analysis Document, Part 4	49
Figure 11:	RPG-Trader Initial State - Map Object Overview, edited screenshot.	52
Figure 12:	RPG-Trader Initial State - Offer and Map Button Inconsistency	54
Figure 13:	RPG-Trader Initial State - Settlement Trade Menu	55
Figure 14:	RPG-Trader Initial State - Dungeon Trade Menu	55
Figure 15:	Map Object Redesign - Wireframing Workspace Example.	61
Figure 16:	Map View Iteration - Information Window Visual Design Workspace Example	62
Figure 17:	Final Build - Map Icon Object State: Player Present	63
Figure 18:	Final Build - Map Icon Object State: Player not Present	63
Figure 19:	Final Build - Map Icon Object State: Selected while Traveling	64
Figure 20:	Final Build - Map Icon Object State: Locked Destination	64
Figure 21:	Final Build - Map Icon Object State: Locked Destination Mini	64
Figure 22:	Final Build - Map Icon Object State: Selected Mini	64
Figure 23:	RPG-Trader Initial State - Map View Overlap	65
Figure 24:	RPG-Trader Initial State - Travel Destination Overlap	66
Figure 25:	RPG-Trader Final Build - Player Map Object before overlap occurs	68
Figure 26:	RPG-Trader Final Build - Player Map Object when overlapping with destination.	68

Figure 27:	Redesigned, Tiered Map Object Icons	69
Figure 28:	Redesigned Buttons	70
Figure 29:	RPG-Trader Final Build - Settlement Trade Menu	72
Figure 30:	RPG-Trader Final Build - Dungeon Trade Menu	72
Figure 31:	RPG-Trader Final Build - Dungeon Trade Menu Bartering Chances Indication 1.	73
Figure 32:	RPG-Trader Final Build - Dungeon Trade Menu Bartering Chances Indication 2.	74
Figure 33:	RPG-Trader Final Build - Ambush Interface	76
Figure 34:	RPG-Trader Final Build - Bandit Ambush Fighting Visualization	77
Figure 35:	RPG-Trader Final Build - Progression Menu, Crew View	78
Figure 36:	RPG-Trader Final Build - Progression Menu, Guild Rank View.	79
Figure 37:	RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Stables .	80
Figure 38:	RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Cart-maker	81
Figure 39:	RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Garrison	81
Figure 40:	RPG-Trader Final Build - Progression Menu, Guild Rank Icons and Frame States	81
Figure 41:	RPG-Trader Final Build - Menu	82
Figure 42:	RPG-Trader Final Build - Intro Character Selection	83
Figure 43:	RPG-Trader Final Build - Intro Starting Region Selection	83
Figure 44:	RPG-Trader Final Build - Intro Name Input	83

All figures used in this work are the authors own creations.

1. Introduction

The purpose of this work is to improve and complete the user interface of the RPG-Trader mobile game.

In its initial state, RPG-Trader is an unfinished early development build with an unsatisfactory degree of usability.

The focus of this work lies on the practical application User Experience design theory and methodology, to lift the initial version to a level of usability and completion that allows for playtesting and balancing of the game mechanics.

To this end, a set of commonly used User Experience Design methods is collected. From this collection, applicable methods are selected.

The selected methods as well as the theoretical basis of User Experience Design are applied to present interface design challenges.

New design solutions are generated and implemented, to increase the games enjoyability and its potential to become commercially viable in the future.

2. User Experience Design

In this chapter the topic of User Experience Design theory is discussed. This discussion includes a definition of the term, as it is understood by the author within the context of this work and the summarization of 17 methods that are applied in UX Design practice.

The chapter is concluded with the discussion of methods selected for application to the current state of the RPG-Trader project.

2.1 User Experience Design

In a video essay by the Interaction design foundation (**What is User Experience (UX) Design, n.d.**) it is stated that the term User Experience Design was coined by Don Norman and refers to a process designer use to conceive of and create products and services that provide meaningful and relevant experiences to users.

User Interface (UI) Design and Usability are also terms associated with UX. While both subsets are relevant within the overarching scope of UX design they cannot be used interchangeably (**What is User Experience (UX) Design, n.d.**).

UX design encompasses not only the product and its functionality but also the experiences surrounding it (**What is User Experience (UX) Design, n.d.**).

2.2 The Seven Factors Model

The Interaction design foundation describes the wholistic nature of User Experience Design is through the seven Factors Model (**What is User Experience. (UX) Design, Video 2, n.d.**).

2.2.1 Factor 1: Usefulness

To evaluate how useful a product is, it must be determined what purpose a product has for its target customer (**What is User Experience (UX) Design, Video 2, n.d.**).

The purpose of a product does not need be practical in the sense that it can be used to accomplish a specific task for the users benefit.

Non-practical benefits like fun or aesthetic pleasure may also be considered purposes (**What is User Experience (UX) Design, Video 2, n.d.**).

It follows that usefulness of a Video Game is defined by how much entertainment it provides if entertainment is the intended purpose of the Video Game in question.

2.2.2 Factor 2: Usability

A product is considered usable if it facilitates the achievement of its primary objective in an effective and efficient manner.

Usability provides safety and comfort while using a product and in turn makes the product more attractive (**What is User Experience (UX) Design, Video 2, n.d.**).

2.2.3 Factor 3: Findability

The factor of findability pertains to two distinguishable aspects of User Experience Design.

Firstly, it is concerned with how easy it is for users to find a product (**What is User Experience. (UX) Design, Video 2, n.d.**).

It is of little importance whether a product is highly usable if potential users are unable to find it (**What is User Experience. (UX) Design, Video 2, n.d.**).

Secondly there is the findability of content within a product. This is most relevant for products in which users may choose from a large amount of content. Video on demand services or online shops are examples for products of this kind (**What is User Experience (UX) Design, Video 2, n.d.**).

2.2.4 Factor 4: Credibility

Products can be considered Credible if they inspire trust. Users trust a product if it fulfills the purpose it is designed for consistently, lasts for a reasonable amount of time and provides accurate and relevant information (**What is User Experience (UX) Design, Video 2, n.d.**).

Credibility is also contingent on whether users believe that a products creator has good intentions (**What is User Experience (UX) Design, Video 2, n.d.**).

2.2.5 Factor 5: Desirability

Products can be made desirable through "*branding, image, aesthetics and emotional design*" (**What is User Experience (UX) Design, Video 2, n.d.**).

If for example two articles of clothing are identical in appearance, make and quality, but one is adorned with the logo of a popular name brand, a significant portion of customers will gravitate to the branded article.

2.2.6 Factor 6: Accessibility

For a product to be considered accessible, it needs to be usable for as many people as possible (**What is User Experience (UX) Design, Video 2, n.d.**).

Accessibility is particularly concerned with usability for users who deviate from the average range of ability. This includes people with cognitive and physical disabilities as well as people varying degrees of media literacy in the context of digital products (**What is User Experience (UX) Design, Video 2, n.d.**).

It can be argued that designing with Accessibility in mind results in products that can be used more easily by everyone (**What is User Experience (UX) Design, Video 2, n.d.**).

2.2.7 Factor 7: Value

The value of a product is the sum of all other factors of user experience. It encompasses both the value for the user and the value for the creator (**What is User Experience (UX) Design, Video 2, n.d.**).

2.3 User Centered Design

User Experience Design aims to optimize every aspect of the users experience with a product. Any design or development must be based on the users' needs to do so effectively and consistently (**What is User Experience (UX) Design, n.d.**).

Most user experience design practice follows some form of user-centered design process for that reason (**What is User Experience (UX) Design, n.d.**).

The term User-centered design (UCD) refers to an iterative design and development processes focused on a deep understanding of a products users (**User-Centered Design Basics, n.d.**).

UCD-processes are based on the industry standard ISO 9241-210:2019 Human centered design for interactive systems (**User-Centered Design Basics, n.d.**).

Descriptions of a user centered design process vary from one source to another, but the basic structure remains consistent. It can be separated into three distinctive phases:

Context Research, Specification of Requirements, Design of Solutions and Evaluation (**User Centered Design, n.d.**).

At the beginning of a UCD-Process the context of use is being researched and specified. It must be determined who the users are, what they need and in which context they will use the product (**User Centered Design, n.d.**).

This research is the basis of any of the following steps in the process (**User Centered Design, n.d.**).

The second phase is about the identification and specification of requirements which must be fulfilled for a product to be successful (**User-Centered Design Basics, n.d.**). These requirements are extracted from the previous research and should be both user and business related (**User-Centered Design Basics, n.d.**).

The third phase is about designing solutions with which the previously specified requirements may be fulfilled (**User-Centered Design Basics, n.d.**; **User Centered Design, n.d.**). The design and development of solutions may have varying scopes and fidelities depending on the stage of development and timeline (**User-Centered Design Basics, n.d.**; **User Centered Design, n.d.**).

In the fourth phase it is determined if the designed solutions fulfill the specified requirements (**User Centered Design, n.d.**). This is referred to as evaluation. A product is best evaluated by its intended user (**User Centered Design, n.d.**).

The purpose of the evaluation is to determine if a solution is successful. If shortcomings become apparent in the evaluation the process another cycle of iteration is initiated (User Centered Design, n.d.).

Iterations may start at any of the previous stages depending on the nature of the shortcoming (Zimmermann, 2019).

The User Centered Design-Process offers a structure within which gradual improvement and optimization of a product can be accomplished.

It is commonly used in user experience design practice and therefore it is also applied in this work (What is User Experience (UX) Design, n.d.).

2.4 UX Design Methodology

In UX Design practice countless methods are utilized for various purposes at various stages of development. These methods are procedures with which various sub-tasks, within the broader UX-Design process, can be accomplished.

In the following section commonly used UX-Design methods are collected and described briefly.

Methods appropriate to present design challenges within the current state of the RPG Trader project are then selected from this collection. The descriptions are focused the general definition and intended purpose of various UX Design methods to inform the following selection.

2.4.1 Card Sorting

Card Sorting is a technique originally used in psychological research (7 Great, Tried and Tested UX Research Techniques, n.d.). The method appears to be designed to reveal associations present within a user's mind.

User Experience Specialist Katie Sherwin (2018) describes the card sorting methodology as follows:

To conduct a card sorting exercise designers prepare a set of cards with words and phrases written on them. Users are then asked to categorize these cards in whichever way they see fit. They may also label the resulting categories. The process should be repeated a number of times with different users to reveal patterns.

Card Sorting is a user research method that may be used in the first phase of a UCD-Cycle to determine how users expect different kinds of information within a information architecture to be organized (Sherwin, 2018).

2.4.2 Expert Review

Nielsen Norman Group UX Specialist Aurora Harley (2018) explains that Expert Reviews, involve asking an individual with user experience design expertise to walk through an user interface and give feedback. The Expert applies their skill set to check for compliance with usability heuristics and other principles of usability related fields (Harley, 2018).

According to Harley, Expert reviews have to be conducted by an UX-Expert who is not involved in the designs creation. She states that the value of this method is contingent on the acquisition of a fresh perspective from a knowledgeable but unbiased entity (Harley, 2018).

It may be used as part of the Evaluation-phase in a UCD-Cycle.

Expert Reviews may also be conducted at any other phase if a prototype is available (Harley, 2018).

2.4.3 Eye Movement Tracking

Tracking a user's eye movements while interacting with an interface can help identifying what they are looking for and how they scan through an interface in order to find it (Eye Tracking, n.d.). It also shows what kinds of content or aspects of an interface take up the user's attention the most and what is overlooked (Eye Tracking, n.d.).

2.4.4 Field Studies

The term Field Studies refers to a category of UX-Design methods focused on the observation of user behavior outside of a specific testing environment according to UX research lead Susan Farrel (2016).

She states that the purpose of Field Studies is to better understand user needs as they present themselves when a product is used under realistic conditions (Farrell, 2016).

This method may be used to conduct user research.

2.4.5 Usability Testing

Senior User Experience Specialist Kate Moran (2019) states that Usability Testing or User Testing are widely applied UX evaluation methodology. She explains that Usability Tests are conducted to identify issues with the design of a product or service, uncover opportunities for improvement and learn about the intended users' behavior and preferences.

According to Moran, there are many ways in which Usability Testing may be conducted though they usually involve a Facilitator, a User and a set of tasks that the user is asked to perform. The user's behavior while performing the given tasks is observed and may be analyzed qualitatively or quantitatively or both (Moran, 2019). Usability testing may be conducted in-person, with both the facilitator and user present in a laboratory or office location or remotely over the Internet with the help of communication and online remote-testing tools (Moran, 2019).

Usability tests are a versatile evaluation method and can be used as soon as prototype is available (Moran, 2019).

The method appears to fit best into the evaluation phase of the UCD-Cycle for this reason.

2.4.6 User Personas

UX Specialist Aurora Harley (2015) states that User Experience designers need to empathize with users to be able to execute their role effectively. In her experience is tempting to attempt to design for a vague, abstract group of users rather than for individual human beings.

According to Harley, User Personas are a tool that can help designers foster empathy for the intended users of a product. They are fictional, yet realistic descriptions of an individual user or a small set of users, designed to give product teams a person to relate to and design for.

This is accomplished by utilizing the basic human tendency to be able to empathize more easily with concrete individuals than with abstract groups (Harley, 2015).

By Harley's description, User Personas contain detailed descriptions of an individual including Name, Age, Occupation, Gender, Location. They also contain descriptions of aspects of the users interests and general behavior as far as it is relevant to the present design challenge (Harley, 2015).

Harley suggests that, to avoid the pitfall of constructing a purely fictional representation of the perfect user, personas must be based on user research (Harley, 2015).

This method is most effective if applied early in a design process (Harley, 2015).

2.4.7 Value Proposition

According to UX Specialist Daniel Birch (2020) Value Propositions are short, concise statements describing how your product solves users' problems or benefits the user, what makes it unique and how it sets itself apart from the competition. They communicate the value of a product to its intended customer in a way that is understood and remembered easily (Birch, 2020).

UX Planet Editor-in-chief Nick Babich (2017) argues that Value Propositions also serve to create a consensus about what the product will be within a design and development team (Babich, 2017).

To be able to generate an effective Value Proposition, the products users' needs, pains and gains must be understood first. For this reason, the Value Proposition must be preceded by user research (Birch, 2020).

Value Propositions are best created early in the UCD-Cycle, when a new product idea is considered (Birch, 2020).

2.4.8 Heuristic Evaluation

Heuristic Evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a method with which usability problems in a user interface can be revealed by judging its compliance with recognized usability principles (Nielsen, 1994a).

These principles (heuristics) are a set of broad rules of thumb rather than specific usability guidelines (Nielsen, 2020). They were originally defined by Jacob Nielsen and Rolf Molich in 1990 (Nielsen, 1994a).

Heuristic Evaluations are similar to Usability Tests but differ in the specifics of execution. They are performed by evaluators with a predefined general reference to compare an interface too (Nielsen, 1994a).

Usability Tests are performed by users who are being observed in the attempt of completing a task (Moran, 2019).

As a product evaluation method Heuristics Evaluations appear to fit best in the fourth phase of the UCD-Cycle.

Evaluators may also be guided and helped by a dedicated observer (also called experimenter) (Nielsen, 1994a). This guidance may help overcome technical issues with a unstable, early prototype for example (Nielsen, 1994a).

2.4.9 Stakeholder Interviews

UX design agency founder Adam Fard (2019) describes Stakeholder interviews as one on one discussions with people who have a vested interest in the success of a given product.

Fard explains that stakeholder interviews are used to gain information about user needs, business goals and technical limitations in connection with a product (Fard, 2019).

He lays the methodology out as follows: First the stakeholders need to be identified. Stakeholders are key figures like executives, team leaders and prominent users. The stakeholders then are asked open ended questions. The responses are documented and analyzed to extract findings that can be used to define milestones and objectives (Fard, 2019).

Stakeholder interviews are most useful if they are conducted early in development (Fard, 2019).

Therefore, they would fit best in the first and second phase of the UCD-Cycle.

2.4.10 User Interview

UX Researcher Kara Pernice (2018) describes User interviews as a UX research method in which users are questioned about their use of a system, their behavior and habits. She states that they are conducted to give insights about users' thoughts on a product or individual process within a product. The goal is to gain information to base design decisions on (Pernice, 2018).

"UX Interviews tend to be a quick and easy way to collect user data." (Pernice, 2018)

User Interviews can be conducted at the beginning of a design process, to inform other user research methods and generate ideas. They can also be used as part of other methods like contextual inquiries (type of field study) or usability tests.

In the structure of a usability test they are best done after the user's interaction with the product has concluded. (Pernice, 2018)

2.4.11 Focus Group

According to Jakob Nielsen Focus Groups are “*somewhat informal*” (Nielsen, 1997) group discussions about issues and concerns about and ideas for an existing or planned user interface (Nielsen, 1997).

The techniques have long been popular in market research but have recently become popular in user research as well (How to Conduct Focus Groups, 2020).

Focus Group-Sessions appear to usually involve between 5 and 10 users who are asked inciting questions by a moderator. Answers and discussions are documented and analyzed (How to Conduct Focus Groups, 2020).

Nielsen (1997) states that method reveals what users want from a product. It can reveal preferences and issues. It is not well suited to generate insights on the usability of a User Interface (Nielsen, 1997).

Focus groups can be conducted “*before interface design and long after implementation*” (Nielsen, 1997).

2.4.12 Customer/User Journey Map

Nielsen Norman Group Chief Designer Sarah Gibbons (2018) describes Journey Maps as follows:

Journey maps are detailed timelines of a series of actions a user goes through to accomplish a goal. They also contain additional information about a users’ thoughts and feelings as they complete each action necessary to achieve their goal (Gibbons, 2018).

Gibbons argues that journey maps serve multiple purposes. Creating journey maps necessitates conversation and alignment of mental models within a product team or organization. The process helps in achieving a shared vision. The finished journey map is a useful communication tool (Gibbons, 2018).

2.4.13 Competitive Audit

According to UX educator Jason Withrow (2006) competitive audits are comprehensive analysis of companies and products competing for market share within the same sector as your own. He states that competitive audits are generally conducted with a small set of dominant competitors in-depth, or with larger set of competitors in less detail (Withrow, 2006).

Withrow states that competitive audits contain both quantitative and qualitative data.

He further elaborates that qualitative data in this context are high-level inventories of features offered by a competing product for example.

Quantitative data can be user performance metrics like click through rates on a web service or success rates from user tests (Withrow, 2006).

The deliverable can be used to inform product or company strategies (Withrow, 2006). It can take various forms depending on the product or project in question and factors like the available resources. (Withrow, 2006)

2.4.14 Kano Model

According to design researcher Jan Moorman (2012), The Kano Model was developed by the Japanese academic Noriaki Kano. She states that the model can be used to measure and describe the level of satisfaction the presence or the lack of a feature evokes in users (Moorman, 2012).

Moorman describes that in this model there are five levels of satisfaction, in declining order: delighted, satisfied, neutral, disappointed, dissatisfied. It also considers how the level of satisfaction for a feature evolves over time and how well executed the feature is (Moorman, 2012).

With the Kano Model features can be separated into three categories: Attractive Features, One-Dimensional Features, Must-Have Features depending on the gain in satisfaction relative to the quality of execution (Moorman, 2012).

In UX practice the model can be used as a framework for surveying, measuring and analyzing customer satisfaction for potential features (Moorman, 2012). This can help deciding which features are worth implementing. (Moorman, 2012)

2.4.15 Information Architecture

Information Architecture (IA) does not refer to a specific method but rather to the science of organizing information in digital products in a way that is easy to search through, navigate and understand by users (Arhipova, 2017).

The overarching goal of IA is to develop a structure for user interfaces with which users can find what they are looking for quickly and with little effort (Arhipova, 2017).

Information architecture is rooted in cognitive science (Babich, 2020). Patterns of human cognition like gestalt principles and mental models as well as limitations like cognitive load inform the deliberate organization of information. (Babich, 2020)

2.4.16 Key Performance Indicators

“Key Performance Indicators (KPIs) are the critical (key) indicators of progress toward an intended result” (What is a Key Performance Indicator (KPI)?, n.d.)

KPI.org describes Key Performance indicators as quantifiable measures to analytically determine the progress of an organization towards a specific goal (What is a Key Performance Indicator (KPI)?, n.d.).

Key Performance Indicators can be either leading or lagging. Lagging KPIs indicate past success, while leading KPIs are target measures that need to be met to achieve a future goal (What is a Key Performance Indicator (KPI)?, n.d.).

KPIs are further separated into operational measurements and strategic impact measurements. Operational measurements are focused on the resource input, the efficiency of the processes applied and the outcome of a project (What is a Key Performance Indicator (KPI)?, n.d.).

There is also the related sub section of User Experience KPIs. User experience KPIs are quantifiable measures regarding the usability of a product (Mayer, 2019).

Strategic impact measurements are about intermediate outcomes. (What is a Key Performance Indicator (KPI)?, n.d.)

As a method, key performance indicators can be used to specify business requirements in a user centered design cycle to *“validate the success factor of the solution derived through the UX strategy.”* (Metha, 2019)

2.4.17 Product Requirement Document

Product Requirement Documents (PRD) are UX design artifacts capturing the essence of what a product team is aiming to accomplish. It specifies what a product is, whom it is for and why it is being built. It also contains a list of features.

Product Requirement Document serve to disambiguate a projects vision within a product team and inform the establishment of a schedule for the project. (Cao, 2014)

2.5 Approach

The chosen approach to UX Design for the RPG-Trader Project is discussed in the following section. This discussion includes a summary of the projects initial state, the definition of confining parameters and the selection of methods for the purposes of this work.

2.5.1 Initial State

To define the approach to UX Design in this work the initial state of the RPG Trader Project must be summarized first. Detailed documentation of prior work on the RPG Trader Project can be found in the appendices A and B.

At the beginning of this work the RPG Trader Project is in a pre-alpha state. Most of the core components of the mobile game, namely most of the planned underlying game systems and the UI necessary to complete and repeat the core gameplay loop, are implemented. Figure 1 and 2 show screenshots of the initial state for reference.

The game is technically playable though not complete enough to be considered an alpha version. There is no main menu and the gameplay mechanics and interfaces for the character progression system and the bandit ambush system are not yet implemented.

Character progression and bandit ambush mechanics are considered essential for the game to be enjoyable.

The game design for RPG trader is mostly defined already. Further refinement of the games mechanics is not the focus of this work.

2.5.2 Confining Parameters

Before methods can be selected and the overall approach can be defined the confining parameters must be established.

It must be stated that many of the methods collected in the previous section can only be approximated within the confines of this work.

The first confining factor in this context is the fact that the UX design is conducted by a single individual rather than a team of designers.

The project involves an additional developer who has different responsibilities. This reduces the utility of many of the described methods. UX Design generally ought to be done in larger, interdisciplinary teams.

The second confining factor is the lack of a budget with which some methods may be used to greater effect.



Figure 45: RPG-Trader - Initial State 1, Screenshot taken on OnePlus 6 Smartphone



Figure 46: RPG-Trader - Initial State 2, Screenshot taken on OnePlus 6 Smartphone

The third confining factor is the present global pandemic. As interpersonal contact must be limited to prevent infection and is prohibited by law to varying degrees User Experience Design methods requiring in-person meetings are difficult to conduct legally or safely in this current circumstance.

Besides these limitations there are general parameters that need to be taken into consideration. The time investment of possible methods must be put in relation to the expected impact in respect to the goals of this work.

The chosen methods must also be applicable to the current state of the project, which excludes any that are to be used before development starts and any that need to be applied to a product after deployment.

The factors mentioned above will be considered for the following selection of methods.

2.5.3 Prior Work

The initial idea for the RPG-Trader project was to create a casual mobile game in which players assume the role of a merchant trying to get rich by selling equipment to adventurers in a classical fantasy RPG (Role Playing Game) setting. This basic idea was grown

into a playable early-alpha version as outlined in the "Current State" section of this chapter.

To reach this stage of development the games mechanics, the underlying mathematical system driving the games emergent economy, content, art style and user interface style were designed and implemented.

This part of development was completed by a team of two, within the six months of the university semester, preceding this work.

In the development of the RPG-Trader game prior to this work multiple UX-Design methods were applied.

Prior work is mentioned here to add context and to offer explanation why certain methods were not selected.

After the projects general premise and a set of potential features were defined an informal competitive audit was conducted to evaluate the viability of the game concept within the current mobile market.

The audit involved the collection of download/concurrent player numbers of a small set of potential competitors, as well as a variety of qualitative data.

The features offered by these competitors were compared to the features planned for RPG-Trader.

User Reviews for these competitive products were also considered to identify points of frustration and reveal potential opportunities for RPG-Trader. Three games with a similar premise were considered in this audit.

Competitive audits are generally more detailed than the one conducted was. Furthermore, it is advised to consider at least five top-competitors (Withrow, 2006).

Only three were considered for this audit and of the three only two had sufficient reach to qualify as top competitors within the given market segment.

The formal execution of the competitive audit therefore was not beyond reproach. For the purposes of informing decisions regarding the features, unique selling point and general group of target users it was sufficient.

It can also be argued that the competitive audit has provided evidence that there is a substantial group of users interested in the kind of experience RPG-Trader is aiming to be.

Competitive audits are best conducted early in a development cycle. (Withrow, 2006)

It seems inadvisable to reapply this method on the chance that there might be more insights to be gathered, as the project has moved far past the point at which competitive audits are of most value.

To further inquire if there was a market for RPG-Trader, broader research on the mobile market was conducted. This research focused on the propensity of interest in the genre's RPG-Trader is associated with.

These genres are RPG, Strategy and Simulation. Also considered were the average revenue per paying/active user within the mobile market for these genres, to be able to judge the potential commercial viability of the project.

Based on research of user groups and analysis of user feedback for similar products a persona was generated in prior development.

Prior work has not included the explicit formulation of user and business requirements. This is corrected in this work, before any further development is done since the UCD-Cycle demands the specification of these requirements to compare the developed user interface against. To be able to measure progression and success of the project in the future, Key Performance Indicators are defined as part of the UX-Requirement specification.

It must be stated that RPG-Trader game is an unpublished study project. There is no company behind its development and no existing user base. As a result, the business requirements cannot be researched using methodology like stakeholder interviews at this stage of development.

The structure of information and navigation in RPG trader was developed using information architecture methodology. This took the shape of a custom variation of a flowchart, developed by the author of this work during prior education as an Interaction Designer BA.

The custom information architecture flowchart describes the general structure by which the interface of RPG-Trader is organized but also encodes the type of content with custom Frames.

This flowchart was a valuable tool for designing the information architecture and guiding the implementation of interfaces and features for the development of the RPG-Trader pre-alpha version, that is the starting point of this work.

2.5.4 Selected Methodology

The evaluation and following iteration of present design solutions as well as the development of new design solutions for additional interfaces are the focus of this work.

In addition, the current stage of development necessitates a focus on the Usability aspect of the broader User Experience design field.

Information Architecture

Parts of the flowchart produced in prior work and the practice of information architecture are relevant to the current work and are updated and improved.

User Centered Design

Throughout development the User Centered Design Cycle is applied. As mentioned before, some features are in the evaluation stage and others are still to be designed.

Evaluation

Within the User Centered design process, the functionally and visually implemented interface components are in the evaluation phase.

There are a variety of evaluation methods that are used in user experience design, some of which were selected to be applied in this work.

Usability Testing

The first of these evaluation methods is Usability Testing.

The usability and user experience offered by the RPG trader interface in its initial state may only be determined objectively by evaluating its performance, as it is interacted with by users.

For this reason, usability tests are conducted in this work and their results inform the following iteration of present features.

Usability KPIs

To generate additional usability benchmark data User Experience KPIs were to be measured.

Heuristic Evaluation

Another evaluation method that is considered suitable for the evaluation of the initial state of RPG Trader is the heuristics evaluation.

For the lack of a UX Design team or a budget to incentivize external UX Experts, this heuristics evaluation can only be conducted by the author of this work.

Design Methodology

The improvement of already implemented components of the user interface and the design of new, additional features is based on insight gained through the use of evaluation methodology.

Design decisions throughout this work are informed by the seven factors model of User Experience Design as it is described in the beginning of this work, as well as the demands of the usability principles, the heuristics evaluation methodology is based on.

3. UX Requirements

As described in the method selection section of the previous chapter, no central artifact capturing the UX Requirements of the RPG-Trader project, was produced in prior development.

To prepare for the following evaluation of the initial build UX requirements this circumstance is corrected.

3.1 Product Requirements

The Product Requirements have been researched and conceived during prior development. The purpose of reiterating these requirements here is to give direction and disambiguate the final product.

As such it can be used to evaluate the progress made in this work against. It also helps to ensure that future additions to the interface are considered in the design of the interface components that are part of this work.

Not every aspect and requirement detailed here can be fulfilled within the confines of this work.

3.1.1 General Description

RPG-Trader is a video game, primarily intended for smartphones. RPG-Trader offers the fun and satisfaction of chasing down the best deals and trading profitably in an ever-changing economy.

3.1.2 Goal

The goal of the RPG-Trader project is to develop a merchant-type mobile game that can compete with already established titles within its market segment.

3.1.3 Users

- **RPG Players** - Those who love playing Role-playing Games on PC or console and look for a mobile game to burn time with.
- **Virtual Merchants** - Enthusiastic consumers of merchant-type games who are looking for an alternative to the ones presently available.
- **Strategic Casuals** - Those who enjoy a stress free and relaxed gaming experience but get bored with the repetition of mundane tasks.

RPG-Trader is a game for players who love classical RPGs and strategy games. For those who want to burn time without having to repeat simple tasks. It is a game for players alienated by pay to progress business models, consciously designed to be less enjoyable if no money is spent to hasten progression beyond a certain point.

3.1.4 Opportunity

In reviews of competitors in the given market segment the basic principle of a game about buying low and selling high is received positively.

Yet there are reoccurring points of contention with redundant, non-engaging gameplay loops and the business models. There is an opportunity for a new title in this market segment if these issues are avoided.

3.1.5 User Interface Components

- **Intro Sequence**
This is a sequence at the start of the game within which a player avatar and the player's name can be chosen and the starting region can be selected.
- **Main Menu**
This is a menu with which a new game can be started, settings can be changed, the game can be exited.
- **Character View**
This is an interface component offering access to information about the player character and their crew.
- **Map View**
This is the abstract top-down view of a world map on which players can travel. It is also the part of the interface from which every other part can be accessed.
- **Travel Planning Interface**
This is an interface component with which destinations around the map can be chosen and travel can be initiated.
- **Bandit Attack Popup**
This is a popup that appears when players run into a bandit ambush while traveling.
- **Settlement Trade Menu**
This is the interface component with which players can acquire wares.
- **Trade Guild Menu**
This is the interface component with which players can purchase skill upgrades.
- **Equipment Menu**
This is the interface component with which players can purchase carts and draft animals and hire guards.
- **Dungeon Trade Menu**
This is the interface component with which players can sell wares to adventur-

ers.

- **Dungeon View**
This is the interface component with which players observe fights between adventurers and dungeon inhabitants.
- **Dungeon Info Menu**
This is the interface component with which players can access additional information about dungeon inhabitants.
- **Monetization Menus**
This is a collection of interface components with which various avenues of monetization are facilitated.

3.1.6 User Requirements

The user Requirements are synthesized from User Research conducted before the beginning of this work.

- The interaction flows of the user interface need to be easy to learn.
- The rate player progression within the game needs to be based on player performance.
- Strategic play needs to be rewarded but casual approaches need to result in progression as well.
- The game system needs to be accessible on the surface but offer more complexity if players choose to delve deeper.
- Players need to have or be able to get all information necessary for strategic decision making.
- Waiting times need to be minimized.
- The game system needs to allow for potentially endless gameplay.
- The game needs to be able to be started and stopped at any point without loss of progression.
- The game needs to work and perform well on smartphones.
- The game needs to be stable and free of game breaking bugs.
- The games internal systems need to produce values based on the rules defined in game design.

3.1.7 Technical Features

- **Emergent Economy System**
This is the mathematical, rule-based system driving the dynamic rise and fall of prizes for wares.
- **Bargaining Mechanic**
This is a context specific calculation of sales probability in relation to the asking price of an item.
- **Ambush Mechanic**
This is a context specific calculation for the likelihood of ambushes occurring while traveling.
- **Map Route System**
This is a web of travel-able roadways connecting the settlements and dungeons of the game world

3.1.8 Monetization

RPG-Trader is to be monetized through in game advertisement with the option of permanently removing advertisements from the game for a small fee.

It may also be monetized through paid cosmetics and downloadable expansions.

None of the monetization options will affect gameplay and player progression.

3.1.9 Business Requirements

Until the product is ready to be released further development can be considered successful if the games is feature complete, stable and enjoyable to the users it is intended for.

After release, the product may be considered successful if it reaches download numbers comparable to its top competitors.

Further down the road the products success is contingent on its profitability.

3.2 Key Performance Indicators

There are two sets of Key Performance Indicators that could be relevant to this work.

The first set of KPI's are operational measurements concerning the desired output in comparison to the measurable input with the applied processes in mind (*What is a Key Performance Indicator (KPI)?, n.d.*).

This set of KPI's is useful to measure the efficiency of the processes applied for the purpose of improvement of processes over time (*What is a Key Performance Indicator (KPI)?, n.d.*). The KPI's of this kind provide data with which process optimization can be informed, in future development, after this work has concluded.

The second set are specialized KPI's for the purpose of quantifying user experience. These include but are not limited to the measurement of time users spend on specific tasks and the rate of successful task completion (*Mayer, 2019*). Key performance indicators of this kind can be used as a framework for quantitative usability testing. (*Mayer, 2019*) As Usability tests are part of this work these KPI's are considered relevant as well.

Both types of key performance indicators are part of this work. It must be noted that KPI's in general are of most value if applied over the course of multiple iterations while the constraints of this work only allow for one cycle.

3.2.1 Operational Measurements

This section is about operational measurement KPIs, which are used to inform process optimization (*What is a Key Performance Indicator (KPI)?, n.d.*).

As previously stated, there is no budget for the RPG-Trader project. There is no relevant input of material resources to be noted. The inputs therefore consist mostly of the time invested by the development team. To quantify this input an approximate calculation based on the constraints inherent to this work can be made.

The maximum time given to work on the project as part of a master's thesis is 22 weeks. Assuming a standard eight-hour, five-day work week this translates to a minimum 880 hours of time invested.

At least half of this time needs to be spend researching and writing the thesis document as it is the primary deliverable, leaving 440 hours to work on the project itself.

The next operational measurement to be considered are the applied processes. To specify these processes the desired output needs to be defined in detail first (*What is a Key Performance Indicator (KPI)?, n.d.*).

The desired overall output in the context of this work is an alpha version of the game.

To advance from pre-alpha to alpha stage the existing interface needs to be evaluated and iterated and missing essential features need to be added.

The first process to consider is the evaluation. This evaluation is done by conducting remote usability test and a heuristics evaluation. Both techniques require time as input and produce a list of shortcomings of the UX design as it is initially. The usability tests also provide additional user research data.

The second process to consider is the design of UX solutions for the issues revealed by the evaluation and the new interface components added.

The components are added in this work are the main menu, intro sequence, the bandit ambush events and the progression system interfaces.

The last process to consider is the Implementation of design solutions.

This includes various sub processes from generating design solutions to the production of assets and programming of functionality. The input remains time.

The efficiency of these processes can be measured by comparing the time invested to the output generated.

In the case of RPG-Trader project in the current state the only apparent way to measure the output quantitatively is by checking if a desired feature is present in the end.

After conducting further research and considering the application of operational measurements to the specific circumstances of this work, it was decided not apply operational measurements.

The format and scope of this work did not appear match well with the methodology.

The conscientious application of operational measurement KPIs would also produce significant workload that could not be justified within the confines of this work.

3.2.2 User Experience KPIs

There are two kinds of UX KPIs, behavioral and attitudinal (Mayer, 2019).

Behavioral UX KPIs are focused on what users do when interacting with a product (Mayer, 2019).

Four common behavioral UX KPIs are the task success rate, the time on task, search vs navigation and user error rate (Mayer, 2019).

The task success rate describes how often a specific task is executed correctly (Mayer, 2019). It requires the careful definition of what constitutes success for a given task to generate useful data (Mayer, 2019).

The task success rate is determined by letting multiple testers attempt to complete a task and calculating a percentage of successful task completion from the results (Mayer, 2019).

As a byproduct the reasons for unsuccessful task completion can be determined when measuring this UX KPI (Mayer, 2019).

In the usability testing this KPI is not measured. Task completion and task success are analyzed qualitatively instead. This was decided because it was expected that testing participants would be successful eventually. It was of more interest how often user errors occurred.

User Errors can be measured in two distinctive ways (Mayer, 2019). The first is the error occurrence rate. It is calculated by dividing the number of times a specific error occurs by the number of possible errors for all users (Mayer, 2019).

The second can be called error rate. It is calculated by dividing the total number of errors by the total number of attempts made by the testers (Mayer, 2019).

It was expected, that errors would occur in the user testing. It was not certain what these errors would be. Therefore it was decided to measure the error rate rather than the error occurrence rate.

Time on task measures the average time testers require to complete a specific task (Mayer, 2019).

This KPI is measured in the Usability Testing as a general benchmark.

Search vs Navigation measures how many of a group of testers use a search function to complete a task rather than using the navigation (Mayer, 2019). Depending on the amount of content offered by a product, different outcomes are desirable (Mayer, 2019).

This KPI is not applicable to the initial state of RPG-Trader since the game does not feature a search function.

Attitudinal UX KPIs quantify the feelings of testers have about a product. There are a variety of methods by which Attitudinal UX KPIs can be measured (Mayer, 2019). Generally, they involve a form of question or questionnaire with a number of gradations between “very positive” and “very negative” as possible answer (Mayer, 2019).

At the beginning of this work RPG-Trader does not contain all intended features some of which are considered essential to provide a well-rounded, engaging gaming experience.

Attitudinal UX-KPI's appear to be focused on the customer reception of product as a whole (Mayer, 2019).

It seemed premature to measure the likelihood of referral and similar KPI's for an incomplete, unreleased pre-alpha version.

4. Evaluation

The interface for completing and repeating the core gameplay loop of buying, traveling and selling is fully implemented in the initial pre-alpha state of RPG-Trader.

The third phase of the previously described User centered design cycle is already completed for this part of the interface when this work begins. Following the structure of the user centered design process, the next step is the evaluation of the existing interface.

The following part of this work is about the methods used and findings generated in this evaluation.

4.1 Usability Testing

One part of the evaluation of the initial state of RPG-Trader is Usability Testing. The following section is dedicated to the methodology in theory, the planning, preparation, execution and results of the usability testing conducted for this work.

4.1.1 Usability Testing Methodology

The Senior User Experience Specialist Kate Moran (2019) describes Usability Testing methodology as follows:

In usability tests a Facilitator guides a participant who is given one or more predefined tasks to complete with an interface.

Facilitator

The facilitator's primary objective is to ensure the quality of test results. They give instructions and answer questions the user might have without influencing the user's behavior (Moran, 2019).

Tasks

The tasks are based on the goals of the usability testing. They may be open ended or closely defined. It is advised to take care when wording tasks to avoid misunderstandings and unintentionally influencing the way in which users complete the task. Tasks may be delivered verbally by the facilitator or given to the participant in written form (Moran, 2019).

Participants

Participants are realistic users of the product or service being tested. For a released product this may be a current user. For an unfinished, unreleased product like RPG-Trader, the participants must be people who are part of the games target audience (Moran, 2019).

Remote and In-Person Usability Testing Usability Tests can be conducted either in-person or remotely (Moran, 2019).

In-person usability test sessions are done in a UX-laboratory environment in which devices and interfaces are prepared for the participant to interact with. They involve further equipment to record and observe

the testing for the purpose of documentation and analysis, as well as materials needed by the facilitator. It must be considered that the laboratory environment in which in-person usability tests are conducted does not represent a realistic context of use for most products (Moran, 2019).

Remote usability testing is an alternative in which participants interact with the interface in more familiar environment (Moran, 2019).

There are different types of remote usability testing, which are primarily differentiated by whether the testing is facilitated by a person or an online-remote testing tool (Moran, 2019).

The general procedure for remote usability tests is the same as for in-person testing. The primary difference is that facilitation occurs online, via screen sharing software and/or specialized remote testing software rather than in a laboratory environment (Moran, 2019).

Qualitative and Quantitative Usability Testing

Two different kinds of data can be generated by usability testing, qualitative and quantitative. Qualitative data may be generated by analyzing participant behavior, think aloud methodology and User Interviews after the interaction with an interface has concluded (Moran, 2019).

"Qualitative usability testing is best for discovering problems in the user experience." (Moran, 2019)

Measuring metrics that describe user experience like the behavioral UX KPIs mentioned in the respective section of this work, produces quantitative data. This is most useful for collecting benchmarks, which can be compared against future tests to determine if design changes were effective (Moran, 2019).

Think Aloud

According to the usability consultant and human computer interface research Jakob Nielsen (2012), the Think Aloud-Method is

often used as part of usability testing sessions. He describes, the methodology as follows:

Participants are asked to continuously verbalize their thoughts and feelings as they interact with an interface. By applying this method, more findings can be generated as internal processes of cognition become apparent and can be documented and analyzed (Nielsen, 2012).

4.1.2 Planning the Usability Testing

To conduct the usability tests for RPG-Trader the structure, type and goals must be defined first.

Testing Goals

The primary testing goal was to evaluate the usability of the already implemented parts of the user interface. These parts enabled the core gameplay loop, the process of buying items at one location on the world map, traveling to another location and selling them for a profit.

The usability testing served to determine if this gameplay loop could be completed by realistic users. It was also of interest which user errors or problems occurred in the process. The first test would constitute a first use scenario, since none of the participants had any contact with the game before.

It was not expected that users knew what to do without any explanation of the games structure, mechanics and functions. The final version was intended to offer a form of tutorial to familiarize first time users with the basics.

Regardless, there was a reasonable interest if and to what extent the interface would be intuitively clear to a first-time user without any explanation other than the stating of the games general concept and the task.

The assumption was that the most insight could be gained if two short testing passes were complete by each participant, without any additional input.

The only exception would be, if participants got stuck and needed a hint to complete a

testing pass. With minimal intervention by the facilitator the risk of influencing participant behavior would be reduced.

The first pass served to evaluate the comprehensibility of icons, interaction flow and architecture and familiarize users with the basic structure of the game.

The purpose of the second pass was two-fold. Firstly, to determine if the interaction flows could be retained short term. Secondly to test if, how and to what extent deeper layers of complexity within the games mechanics and interface would be accessible to participants once they were familiar with the basics.

To further increase the amount of qualitative data generated by the test it was decided to ask participants to apply the Think-Aloud method as well.

For the purpose of benchmarking quantitative data was collected as well. Specifically, the behavioral UX KPIs time on task and error rate.

The time on task KPI was measured not only for the duration of the entire main task but also for the three sub tasks "Buying", "Traveling", "Selling" that would be completed multiple times during a single testing pass. This way the KPI would reflect how long players needed to fulfill each step in the core gameplay loop. Buying, Traveling and Selling were enabled through different interface components.

The error rate was measured only for the sub tasks. It was difficult to define what constituted an attempt for the entire testing pass. The most obvious definition was that each testing pass was one attempt. In this case the error rate would be 100% as soon as one error occurred. It was considered unlikely that any testing pass would be without errors.

Counting the errors for each attempt of completing a sub task would not produce useful information either. Players could try to plan a route and fail multiple times until they succeeded.

To produce more useful data other definitions for attempts and errors were used.

For the travel sub task each attempt at planing a route would be counted. If players did not initiate travel before making another attempt or traveled to a place other than the one they intended, an error was counted.

For buying an selling sub tasks each attempted transaction was counted. Whenever the transaction did not go through an error was counted.

For trading in the dungeon trade-menu that meant that offers that were not accepted by adventurers would also be counted as an error. Adventurers not accepting an offer was the expression gameplay mechanic, it was expected to happen. It did not truly constitute an user error. It was still counted as such to calculate the selling KPI. This way the KPI would reflect the rate of successful offer rather than the user error rate. It was decided that the rate of successful offer was more useful data in this specific context.

Testing Structure

Because of the global pandemic present when the tests were to be conducted, a feasible remote usability testing approach needed to be found.

This remote testing would be conducted with the facilitator and participant in their respective home or place of work, connected through the voice and video communication tool Discord (Discord, 2021).

The participant then had to be guided through the process of installing an application file with a test build of RPG-Trader on their personal phone and setting up screen recording software. They also had to be asked to position in a way that allowed for viewing the phones screen through a webcam if available.

The participants were also asked about their age, profession and interests relating to video games.

After the set up was concluded the tests could be conducted, starting with the read-

ing of a scripted explanation of the first task. The participant completes the task while being reminded to continuously express their thoughts and feelings verbally if need be.

After the completion of a testing pass the user would be asked to save a screen recording of the process. The second task is explained and completed the same way.

The testing session would be concluded with an optional impromptu user interview if the participant was willing, and the facilitator detected the need clarify any particular occurrences throughout the testing passes. Lastly the user would be asked to send the screen recordings they took back to the facilitator.

This structure of this remote usability testing approach is based on the methodology of usability testing as described in the previous section. The methodology had to be adapted to the specific context in which the testing was conducted.

The adaptations made, and some of the considerations that went into the structuring of this usability testing should be explained for the purposes of this work.

It must be stated that the Usability testing required participants to set up parts of the equipment, in a way that is not representative of a realistic context of use.

The test is usually set up by the entity conducting the study, rather than the participants. This way it can be made sure that all equipment is in working order and the tests can commence without distractions, more easily.

A lot is being asked of the participants if they are required to do part of the preparation and collection of footage themselves. It also limits the set of possible participants to those who possess the required technical knowledge which in turn may affect the test results.

Considering the limited time and financial budget and the desire to conduct user testing in a safe and legal manner in midst

of a global pandemic, compromises had to be made. The available budget was used to compensate Participants for their contribution.

4.1.3 Usability Testing Preparation

The following section, the preparations that were made before the tests could be conducted are discussed. This includes the preparation of the script used by the facilitator, the adjustments made to the current build of the game and the acquisition of participants.

Facilitator Script Preparation

A facilitator script was produced. This script details which actions to take, in which order and contains a consent form, and two task descriptions, to be read to the participants at the appropriate moments during the tests.

The consent form was to be read at the beginning of the session so participants could state their agreement with the recording of audio and video for the purposes of this usability test. It also served to inform the participants what these purposes were and how personal information would be treated.

Verbal Consent Form Script (Translated)

I agree to participate in the study conducted by the author of the "User Interface and User Experience for a Mobile Game" master's thesis.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

I agree to the recording of my voice and webcam footage for the purpose of later analysis.

I understand that my age, profession and interests may be made publicly accessible as part of the "User Interface and User Experience for a Mobile Game" master's thesis.

Please indicate if you also agree to your full name being used in the same way as age, profession and interests.

Do you have any questions?

Please state your agreement with these conditions to indicate that you understand the information presented to you and that any questions you might have about the session have been answered.

(based on the Consent Form (Adult) Template from usability.gov (Consent Form (Adult), 2021))

Task 1 Script (Translated)

The first task is, to buy an Item in a Settlement, transport it to a dungeon and sell it to an adventurer there.

Task 2 Script (Translated)

The second task is, to do the same thing as you did in the first test but try to make as much profit as possible.

Other verbally delivered information, about the purpose and procedure of the testing, as well as an explanation of the basic premise and core gameplay loop were scripted as well. This way it could be made sure that each participant received the same information before and during the testing.

Testing Build Preparation

The game build that would be used in the usability testing had to be prepared as well. The build available at this point still had several bugs. The game mechanics had not yet been balanced.

To avoid distracting users with issues not immediately related to the design of the interfaces, as many bugs and balancing issues as possible were removed, solved or otherwise accounted for beforehand. Additionally, the starting point for players was adjusted to be more suitable for the tasks the participants were given.

One major bug was, that the camera view transformation "jumped" when zooming which made the navigation on the world map confusing. The responsible script would be reworked later, but for the testing this issue needed to be prevented from occurring provisionally.

This fix would not be sufficient to make the camera control work as intended but it was still preferable to the alternative in this case.

The test build was exported as a .apk file which could easily be shared by various means including text messaging applications. The installation would merely require the .apk file to be downloaded and tapped on the participants smartphone.

Voice and Video Software Preparation

A Discord server dedicated to the Usability Testing was set up. This involved adding three text channels to give participants access the consent form and the two task descriptions at appropriate moments during the usability tests.

To avoid distractions and to be able to reveal the task descriptions just before either of the testing passes, the “roles” feature of discord was utilized (Discord, 2021). There were three roles in total, “participant”, “participant test 1” and “participant test 2”.

The participant role was permitted to view the consent form, participant test 1 and 2 could view the consent form and the respective task descriptions. These roles would be assigned to the participants, one after another, as the testing progressed.

Figure 3 shows a screenshot of the Discord server that was set up.

Acquisition of Participants

Lastly, suitable participants had to be found.

These participants had to be realistic representatives of the intended user group. They also had to be confident to handle the installation of the test build and part of the recording of their usability testing.

Participants were compensated for their contribution.

Only half of the potential participants contacted possessed an android smartphone. It was uncertain if the available build would work with Apple devices at this point. Usability testing was therefore only possible with owners of android devices.

Two participants, who met all requirements, were available in the given time frame.

It would have been preferred to do user testing with at least three participants but at this stage of development any number was certain to produce valuable insight.

4.1.4 Execution and Results

In this section the execution of the usability testing is discussed. Each test was documented by reviewing the footage of the testing multiple times and noting any occurrence relevant to the design of the user interface. The task durations were documented based on the testing footage as well. The KPI results were determined by counting attempts and errors as defined in the planning section of this chapter.

The tests were narrated and following the prepared facilitator script. Participants understood the tasks and were able to complete them eventually. There were no issues while setting up the testing build. In the first testing pass of each testing session a previously undetected bug occurred. This bug prevented the participants from reaching their destination. The affected testing passes had to be restarted. The testing results were taken from the interrupted testing passes up to the point at which the bug occurred as the repeated testing pass no longer constituted a first use. The repeated testing passes went without technical issues.

KPI Result Summary

Pass 1 Average Time on Task: 08:33 min
Pass 1 Average Buying Time: 01:26 min
Pass 1 Average Traveling Time: 08:33 min
Pass 1 Average Selling Time: 05:39 min

Pass 2 Average Time on Task: 16:10 min
Pass 2 Average Buying Time: 03:44 min
Pass 2 Average Traveling Time: 00:59 min
Pass 2 Average Selling Time: 00:57 min

Error Rates Participant 1

Pass 1 Buying Error Rate: 0%
Pass 2 Buying Error Rate: 0%

Pass 1 Travel Error Rate: 86%
Pass 2 Travel Error Rate: 20%

Pass 1 Selling Error Rate: 50%
Pass 2 Selling Error Rate: 59%

Error Rates Participant 2

Pass 1 Buying Error Rate: 0%
Pass 2 Buying Error Rate: 0%

Pass 1 Travel Error Rate: 100%
Pass 2 Travel Error Rate: 50%

Pass 1 Selling Error Rate: 50%
Pass 2 Selling Error Rate: 84%

It must be stated that the explanatory power of the KPIs is low with a sample size of two. The KPIs serve primarily as a benchmark for future testing.

The KPIs can only be used as an additional layer of indication for specific usability issues detected in the qualitative analysis.

Detected Usability Issues

- Center on Player functionality unclear
- Severe intractable object overlap
- Travel planning interaction flow unclear
- Travel status feedback insufficient
- Settlement Map Object information unclear
- The available dungeon information was not utilized
- Settlement Trade Menu item price not perceived initially
- Dungeon Trade Menu adventurer selection interactions not obvious
- The available dungeon information was not utilized
- It is difficult to adjust the bargaining slider element precisely
- The bargaining success chance text element was overlooked completely

- No convenient buying price and profit indication
- The fact, that adventurers refuse to buy the exact same item they already have was not apparent
- It is difficult to adjust the bargaining slider element precisely
- The bargaining success chance text element was overlooked completely by both participants
- There is no convenient way of determining the base value of an item/the price the item was bought for
- The fact, that adventurers refuse to buy the exact same item they already have was not apparent
- The participant the participant documented the buying prices by hand to be able to determine the amount of profit he would make by selling said items later
- Map objects overlap and travel planning node placement was confusing
- Base price and profit/loss were not apparent enough within the trade menus
- The possibility to switch between multiple adventurers in the dungeon trade menu was not perceived
- The case of a player selling two items of the same kind to the same adventurer one after another was not considered in prior development
- Magic Effect iconography was only partially understood

The complete testing analysis documents for both of the conducted testings are displayed in the Figures 3 - 10 on the following pages. The results of the usability testings are used and discussed further in the Usability Analysis section of this Chapter.

RPG-Trader - Usability Testing Session 1

Participant Name: Max Paniagua Türich

Date of Birth: January 17,1990 - 30y.o.

Occupation: unemployed

Device: Huawei p30

Video Game Genres: Shooter Games, Role Playing Games, Strategy Games, occasionally Sidescrollers

User Type: RPG Payer

Testing Pass 1

Total Time on Task: 06:54

Time on (Sub-)Task "Buying": 01:35

Buying Error Rate: 0% (1 Attempt, 0 Unsuccessful)

Time on (Sub-)Task "Traveling": 03:55

Travel Error Rate: 86% (7 Attempts, 6 Errors)

Time on (Sub-)Task "Selling": 00:46

Selling Error Rate: 50% (2 Attempts, 1 Unsuccessful)

Testing Analysis Notes:

Two of the three main map view icons are identified correctly.

The participant attempted to use the menu icon. The icon is not active as the menu was not yet implemented.

The participant partially understood the function of the "center on player" toggle after pressing it once and seeing its effect.

The fact that this element locks the camera position and zoom, to center the view on the player map object, was not apparent to the participant. This led to confusion when the participant tried to zoom and did not realize, that the center on player toggle was still active and prevented this operation.

The participant expressed the expectation that this function would move the camera to the player position and subsequently release controls, rather than fixing the camera transformation to the player position until it was manually deactivated.

Opening the Settlement Map Object sub-menu, the participant falsely interpreted the cities name to be the name of a trader character within the city. This interpretation was based on the settlement race icon featuring an elven face and the fact that the settlement inventory was displayed beneath the name text.

The trade menu itself was found by the participant without issue. This may be because the test build did not feature the two other options within the settlement sub menu, leaving only one button to press.

Figure 47: Usability Testing Session 1 Analysis Document, Part 1

The trade menu itself was found by the participant without issue. This may be because the test build did not feature the two other options within the settlement sub menu, leaving only one button to press.

The participant bought an item without issue. He subsequently realized he had not perceived the items price before buying. He then scanned the interface for this information displayed and found it quickly.

The participant identified the travel planning mode toggle icon correctly. He proceeded to press the toggle. The participant initially expected the travel nodes, displayed in the travel planning mode, to be dungeons which could be selected to set a destination.

The following attempt to plan a route failed multiple times.

The pattern of touching multiple nodes one after another, along the road system to assemble a route between one point of interest and another was not discernable for the participant.

The participant kept the camera zoomed out which led to nodes and points of interest overlapping heavily. This caused the participants to tap and activate elements unintentionally.

The previously mentioned inability to change the zoom level due to the activated camera centering intensified these issues. The participant had to be prompted to deactivate the center on player function.

The participant planned a route to a dungeon eventually but overlapping and the unclear relation between node points and dungeon/settlement map objects caused remained an issue.

It was not clear to the participant why travel could not be started when the planned route did not end at a valid destination node.

After the travel was concluded the participant was unsure if he had arrived successfully.

In the dungeon trade menu, the participant attempted to press the adventurer icons above the adventurer cards section to switch between the multiple different adventurers present at the dungeon. This was a planned but not yet implemented option. After some trial and error, the participant realized that the adventurer objects could be scrolled through with a swiping motion.

The participant was able to sell an item to an adventurer. The participant also attempted to increase the asking price using the bargaining slider. When the offer was rejected, the participant assumed that the asking price had to be reduced for the offer to be accepted.

Detected issues:

- Center on Player functionality unclear
- Severe interactable object overlap at far zoom in plan mode
- Travel planning interaction flow unclear
- Travel status feedback insufficient
- Settlement Map Object information unclear
- Settlement Trade Menu item price not perceived initially
- Dungeon Trade Menu adventurer selection interactions not obvious

Figure 48: Session 1 Analysis Document, Part 2

Testing Pass 2

Total Time on Task: 11:48

Time on (Sub-)Task "Buying": 00:15 ; 00:23 ; Ø 00:19

Buying Error Rate: 0% (13 Attempts, 0 Unsuccessful)

Time on (Sub-)Task "Traveling": 00:30 ; 00:30 ; 00:31 ; 00:30 ; Ø 00:30

Trade Error Rate: 20% (5 Attempts, 1 Unsuccessful)

Time on (Sub-)Task "Selling": 07:08 ; 03:28 ; Ø 05:30 (Ø 24.5s per item)

Selling Error Rate: 59% (32 Attempts, 19 Unsuccessful)

Testing Analysis Notes:

The participant started by buying every item in the starting settlements inventory and travel to another one in order to utilize all their capital and maximize potential profit.

The participant planned a route and traveled to a dungeon without issue. Arriving at their destination he realized that there were no adventurers present. The information, how many adventurers are present a dungeon, is available in the interface but was not perceived by the participant. The participant realized that this information was available eventually but forgot to look up adventurer counts at dungeons when planning travel routes again later.

When selling items in the dungeon trade menu the participant initially tried to offer items to adventurers based on whether they already had an item of this kind equipped.

When selling items in the dungeon trade menu the participant tried to offer items that adventurers did not currently have.

He also tried to increase the asking price in an attempt to increase profits. The participant decreased the asking price when he realized the adventurer would not be willing to buy items at higher prices.

The participant had difficulty adjusting the price precisely.

In further attempted sales the participant discerned that the likelihood of success was connected to more factors than just the asking price.

He eventually suspected that there was a relationship between how much an offered item improves an adventurer's statistics and their willingness to purchase this item.

He also started to attempt to offer items of elvish make to elven adventurers, implying that he suspected that matching item and adventurer culture would be beneficial.

During testing the participant did not notice the text element next to the slider expressing the likelihood of success when selling an item with a percentage value.

When trying to adjust the asking price the participant selected and deselected the offered item frequently. This was done to find out how much the item was bought for, as the price initially displayed when selecting an item is the items base-price, which the participant deduced.

The participant tried to offer the same items that were identical to what the adventurer already had equipped multiple times. It was not apparent to them why this was unsuccessful.

Figure 49: Usability Testing Session 1 Analysis Document, Part 3

Detected issues:

- The available dungeon information was not utilized
- It is difficult to adjust the bargaining slider element precisely
- The bargaining success chance text element was overlooked completely
- There is no convenient way of determining the base value of an item/the price the item was bought for
- The fact, that adventurers refuse to buy the exact same item they already have was not apparent

Figure 50: Usability Testing Session 1 Analysis Document, Part 4

RPG-Trader - Usability Testing Session 2

Participant Name: Pascal Dwojak

Date of Birth: July 25.1994 - 26y.o.

Occupation: assembly worker

Device: Samsung Galaxy Note 10 plus

Video Game Genres: Shooter Games (First Person), Action-Adventure Games, Strategy Games

User Type: Strategic Casual

Testing Pass 1

Total Time on Task: 10:12

Time on (Sub-)Task "Buying": 01:18 -

Buying Error Rate: 0% (1 Attempt, 0 Unsuccessful)

Time on (Sub-)Task "Traveling": 06:14

Travel Error Rate: 100% (Participant was unable to complete task without explanation)

Time on (Sub-)Task "Selling": 01:17

Selling Error Rate: 50% (2 Attempts, 1 Unsuccessful)

Testing Analysis Notes:

The participant tried to tap the settlement map object multiple times. The object does not react. This happened due to a technical issue with the testing build.

The participant was able to open the map object and trade menu eventually. He proceeded to buy multiple items without issue.

The participant scanned the map view and interpreted the dungeon icon correctly. The participant tried tapping dungeon map objects in the expectation that this would allow him to travel to the dungeon location. After trying to select multiple different dungeon map objects the participant suspected that it may only be possible to travel to certain dungeons. This was not the case.

Eventually the facilitator was compelled to hint at the plan mode toggle element at the bottom of the screen. Even after activating the plan mode the participant was unable to discern the interaction flow of the travel planning interface. Further explanation was required.

The participant attempted to tap dungeon map objects again to set their destination, while the plan mode was active. After some trial and error he successfully set a valid route to a dungeon on accident and proceed to travel there.

The participant was confused by the placement of the travel planning nodes.

Figure 51: Usability Testing Session 2 Analysis Document, Part 1

When the participant had reached the dungeon, he opened to the dungeon trade menu. Within the menu he immediately offered an item to an adventurer. He attempted to sell the item at its default price first. When the offer was declined the participant searched for a way to adjust the trade price and discovers the bargaining slider. He proceeded to reduce the asking price and sell the item at a loss.

Detected issues:

- Technical issue with non-reactive map objects.
- The travel planning interaction flow was unclear.
- Map object overlap and travel planning node placement was confusing.
- Base price and profit/loss where not apparent within the trade menus.

Figure 52: Usability Testing Session 2 Analysis Document, Part 2

Testing Pass 2

Total Time on Task: 20:31

Time on (Sub-)Task "Buying": 07:09

Buying Error Rate: 0% (9 Attempts, 0 Unsuccessful)

Time on (Sub-)Task "Traveling": 01:46 ; 01:10 ; Ø 01:28

Travel Error Rate: 50% (2 Attempts, 1 Errors)

Time on (Sub-)Task "Selling": 08:59 (Ø 01:30 per item)

Selling Success Rate: 84% (13 Attempts, 11 Unsuccessful)

Testing Analysis Notes:

The participant opened the trade menu and started purchasing every item available at the location, while writing down the amount of money they started with and the price to which they bought each item on a piece of paper. In the process the participant wonders if they can negotiate the buying price in any way.

The participant proceeded to plan a route to a dungeon. The overall interaction flow of route planning and traveling was understood but the participant did not arrive at the intended destination first. This happened because the participant unknowingly selected a node close to the one he intended to select.

On the next attempt the participant managed to travel to a dungeon, albeit a different one than the one he originally intended to go to.

The participant opened the dungeon trade menu and attempted to sell items to an adventurer. The first few offers were all declined. The participant kept adjusting the offering price and was uncertain why the adventurer was unwilling to purchase the offered items.

The offer success percentage text element was not perceived by the participant. He tried to make offers with a success chance of zero percent.

While adjusting the price the participant had difficulty setting the exact asking price they desired with the bargaining slider.

Eventually the participant tried an item fitting the adventurers class and was able to sell it with some profit. Why this item could be sold at a profit while the others could not be was not apparent to the participant at this point.

The participant then tried to sell another item to the adventurer. This item belonged on the same slot as the item the participant just sold to this adventurer. The participant did not take notice of this circumstance.

After multiple attempted sales were unsuccessful the participant investigated the interface in search of a plausible reason as to why. In this process it was revealed that the participant was uncertain about the meaning of the magical effect icons. He identified the fire and ice magic icons correctly, interpreted the air magic icon as illusion magic and could not determine the meaning of the earth magic icon.

Eventually the participant made multiple profitable sales. Through trial and error, the relationship between potential improvement of the adventurer stats with an item and the success chance of sales was understood. When the adventurer had no money left to spend the participant thought the task completed. The participant made as much profit as he assumed to be possible.

Figure 53: Usability Testing Session 2 Analysis Document, Part 3

Detected issues:

- The participant the participant documented the buying prices by hand to be able to determine the amount of profit he would make by selling said items later.
- Map objects overlap and travel planning node placement was confusing.
- Base price and profit/loss where not apparent enough within the trade menus.
- The bargaining success chance text element was overlooked completely.
- The possibility to switch between multiple adventurers in the dungeon trade menu was not perceived.
- The case of a player selling two items of the same kind to the same adventurer one after another was not considered in prior development.
- Magic Effect iconography was only partially understood.

Figure 54: Usability Testing Session 2 Analysis Document, Part 4

4.2 Usability Analysis

To further evaluate the pre-alpha build, an in-depth usability analysis was conducted. Interaction flows, visual appearance and information architecture of the initial state were critically evaluated based on the 10 usability heuristics, used in the heuristics analysis method.

Heuristics analysis, as defined by Jakob Nielsen and Rolf Molich (1994a), requires multiple evaluators to “*examine the interface and judge its compliance with recognized usability principles.*” (Nielsen, 1994).

Acquiring suitable evaluators to that end was not possible within the confines of this work. This analysis will only be conducted by the author and does not constitute a faithful application of the heuristics evaluation method. The “*recognized usability principles (the “heuristics”)*” (Nielsen, 1994) lend structure to the following usability analysis and are therefore considered useful in this context.

The results of the usability testing factor into this analysis.

4.2.1 The 10 Usability Heuristics

Jakob Nielsen (2020) defines the 10 Usability Heuristics as follows:

1. **Visibility of system status**

The user should always be informed about the current status of the system. To be able to navigate an interface, users must be informed where they currently are and be given immediate feedback about the effects of their actions (Nielsen, 2020).

2. **Match between system and the real world**

The design of an interface should be in accordance with principles users are familiar with through prior experience. This includes principles inherent to the physical world and principles common in other interfaces. Terminology, iconography, symbolism and basic concepts should be familiar to the intended user (Nielsen, 2020).

3. **User control and freedom**

Users should always be enabled to conveniently abort a process, close a window or undo and redo an action. When interacting with an interface, mistakes can occur. It must be made sure that actions do not lead to unwanted consequences. (Nielsen, 2020)

4. **Consistency and standards**

There are two types of consistency, internal and external (Nielsen, 2020).

External consistency implores designers to keep their design in line with established conventions. Users expect all digital products they use to function similarly (Nielsen, 2020).

To achieve internal consistency, similar elements within an interface must look and function similarly. If consistency is maintained in a design, users’ cognitive load is reduced (Nielsen, 2020).

5. Error prevention

Unconscious errors, due to inattention and conscious error due to a misinterpretation of a design have to be avoided. Some errors cause severe problems for the user others are mild and frustrating. High-cost errors should be prevented first (Nielsen, 2020).

6. Recognition rather than recall

To prevent issues arising from overloading limited short term memory, the use of an interface should require as little memorization as possible. The recollection of information should generally be aided by offering cues (Nielsen, 2020).

7. Flexibility and efficiency of use

Different types of users, at different levels of experience should be offered different ways to carry out a process. With shortcuts, personalization and customization interfaces can be made more efficiently usable for a larger set of users (Nielsen, 2020).

8. Aesthetic and minimalist design

An interface should be designed in a way that makes it easy to focus on important things. This means that the information displayed should be limited to what is relevant to the present task. It also means that the interface should not be visually overbearing (Nielsen, 2020).

9. Help users recognize, diagnose and recover from errors

When something goes wrong, users should be informed on what went wrong in a way that is understandable to them. If possible, they should also be offered a way to fix the issue or at least be given enough information to be able to find a solution on their own (Nielsen, 2020).

10. Help and documentation

Users should have easy access to additional information (Nielsen, 2020).

4.2.2 Map View Usability Analysis

In this section it is discussed how the elements of the map view interface prior to iteration worked and were their behavior failed to comply with usability heuristics.

The Map View is the part of the RPG-Trader user interface, with which players can facilitate travel between points of interest (POI) like settlements and dungeons.

POI Map Object States

Each point of interest was marked by an interactive map object that could be selected by the player. Depending on the player's current location in the game world and the kind of POI that was selected, one of two transitions would occur.

If the player character was currently present at the location of the selected POI map object, it transitioned into a selected state as displayed in Figure 11, on the next page. In this state, an information window unfolded and buttons leading to either dungeon or settlement trade menus appeared. At the same time, brackets and frames would appear to indicate that the object was selected.

If the player character was not currently present at the selected point of interest an almost identical transition would occur, but the buttons leading to the trade menus would not appear.

It was designed this way because players should only be able to trade at the point of interest they were present at. Any other point of interest could still be selected but would only display information intended to help players decide where to travel next.

In theory players could deduce why some map objects behaved different than others when selected, there is a player map object indicating the player's current location. Yet it can be argued that the two kinds of selected state were too similar. The system status should be more visible in this regard.

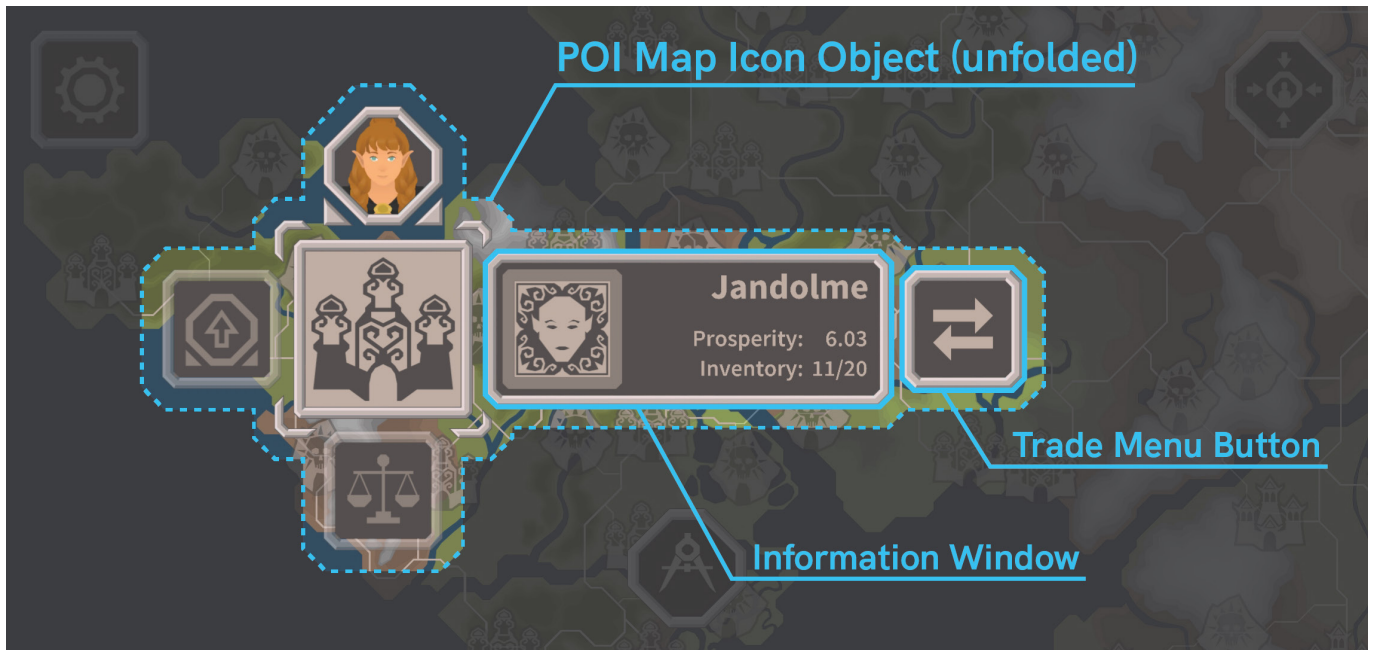


Figure 55: RPG-Trader Initial State - Map Object Overview, edited screenshot

POI Camera Restriction

Further issues became apparent when the camera behavior was analyzed.

When a map object was selected, camera controls were disabled, and the camera was centered on the selected object. The camera could not be moved until the object was deselected.

This constituted a lack of compliance with the third usability heuristic in two ways.

Firstly, the user's freedom to move the camera was taken away upon selection. This would be especially problematic if selection occurred accidentally.

Secondly, there was no obvious and convenient way in which the object could be closed to regain camera controls. The only way to deselect the map object was to tap it again. Players would likely have to discovered this by trial and error.

There were no major consequences in this case, but this behavior could easily cause annoyance and worsen the experience.

POI Information Window Content

There are further issues with the information window, that appeared when a POI map object was selected.

The first thing to note is that the player could only select one POI map object at a time. This meant that the user needed to select and deselect multiple POI-Objects and remember one or more numerical values describing the POIs status.

Players would likely want to compare multiple POIs before deciding on their next destination. This comparison relied on short term memory and did not comply with the recognition rather than recall heuristic.

Information windows, indicated in Figure 11, contained the following information: settlement name, settlement culture, settlement prosperity, number of items in settlement inventory and dungeon name, dungeon type and the number of adventurers at dungeon.

Information about the danger of route was not included in the pre alpha build. Therefore, an important piece of information relevant to player travel decision making was unavailable.

Information windows featured an icon on the left, representing either the culture of a settlement or the type of dungeon. Text elements on the right-hand side of the window expressed the rest of the information.

The settlement culture/dungeon type icon was the visually dominant element within this information window. The icon was misinterpreted by one testing participant. The symbolized information was no more relevant to decision making than the rest, but it was featured more prominently.

In addition, the information contained in the settlement culture/dungeon type icon was redundant, because the same information was already contained in the appearance of the POI icons. These factors made it needlessly difficult to utilize the offered information. This constituted a lack of compliance with the eighth heuristic.

Travel Planning Flow

Both usability testing participants expected to be able to travel to a selected POI directly. This was not possible in the pre-alpha build of RPG-Trader.

Players had to tap an interactable on the bottom of the screen to open the route planning mode and plot a path by selecting a row of “nodes” between their current location and their intended destination to initiate travel.

“Jakobs Law states that people spend most of their time using digital products other than yours.” (Nielsen, 2020)

In this case the most prevalent prior experience users have with an interface for the planning of a route on a smartphone device, likely stems from mapping services like Google Maps.

With the mobile interface of Google Maps, users can select a destination and are subsequently offered a possible route from their current location to this destination (Google Maps, 2021).

In the usability testing, participants expected route planning in RPG-Trader to work similarly and were confused when it did not. It follows that the route planning interface of the RPG-Trader pre-alpha build is not consistent with users’ prior experiences.

Both external consistency and match between system and real world were insufficient in this regard.

The travel planning interaction flow, as implemented in the pre-alpha build, was originally conceived to accommodate a specific gameplay mechanic.

On some road segments on the map, players were supposed to be able to travel faster, on others they would travel with less risk of being ambushed by highwaymen.

This mechanic was intended to make the process of planning routes more interesting. Players had to be given the ability to customize routes in detail. They needed to be able to decide which way to go at any crossing or fork in the road system.

There are perceivable ways in which automatically generated routes could be made customizable. Google Maps for example offers the ability to customize the path to the intended destination by selecting one of the alternative routes offered. (Google Maps, 2021)

The technical realization of functionality such as this was not feasible within the confines of pre-alpha development and would remain so within the confines of this work.

The implemented route planning interaction flow had other issues beyond the lack of external consistency and match between system and real world. It was prone to user errors. The mental model of the user does not match up with the design, leading to conscious mistakes. Overlap and general visual overload on the map view led to accidental selections.

Map Object Overlap

In the process of evaluating the map view interface it became clear that many of the most severe usability issues seemed connected to the overarching problem of element overlap.

The root of this issue lies within the confines of the target device. Smartphones have screens of comparatively small physical size. Yet interactive elements must be big enough to be used comfortably.

To ensure usability and accessibility, map objects were automatically scaled to maintain the same relative size on a smartphone screen regardless of how far the user zoomed in or out. With the quantity of interactable objects that needed to be displayed on the map this lead to overlap.

In design and development prior to this work, measures were taken to combat this issue. Map objects, other than the player map object would be reduced to half their size at a certain zoom threshold to prevent overlap when the player zooms out and the relative, visual distance between points on the map decreases. This proved to be insufficient.

The issues resulting from object overlap were many. Objects could not be differentiated from one another visually, it was difficult to hit the object that was aimed for, the map view appeared cluttered and overloaded, the road system was barely visible beneath all the map objects.

Overlap related issues caused a severe lack of compliance with the first, fifth and eighth usability heuristic. This reduced the overall usability drastically.

Interactable Consistency

There are minor issues with the interactable elements within the map view interface. As Figure 11 shows, Map view buttons are not internally consistent with buttons in other menus regarding their visual appearance and click feedback. Button icons also lack visual consistency.

The visual feedback of the center on player toggle object is also insufficient resulting in a lack of visibility of system status.

Summary

The Map View interface generally fails to comply with most of the usability heuristics at one point or another.

The map view is an integral part of the RPG Trader interface. Solving the usability issues described in this analysis ought to improve the overall user experience significantly.

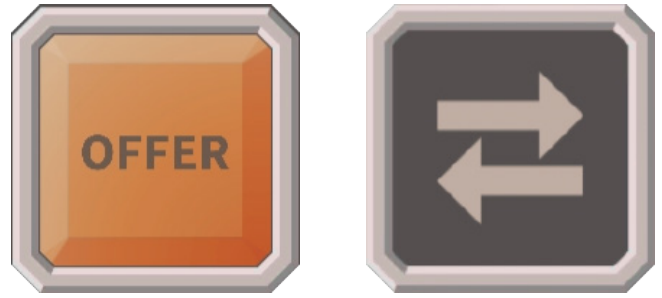


Figure 56: RPG-Trader Initial State - Offer and Map Button Inconsistency

4.2.3 Trade Menu Usability Analysis

In this section, the design of the trade menu interfaces, prior to iteration and its shortcomings are discussed.

Trade Menu Overview

There are two structurally similar but functionally distinct trade menus in the RPG-Trader early alpha interface.

Firstly, there is the settlement trade menu shown in Figure 13, with which players can purchase wares. Secondly, there is the Dungeon Trade Menu, shown in Figure 14 with which players can sell wares to adventurers. In this section, usability issues with the pre-alpha implementation of these trade menus are discussed.

The trade menu structure contains four major elements. There are two inventories, one for the buying entity and the other for the selling entity. Between those menus there is an Item information section.

This section displays information about selected items. There is also an interactive section on the right-hand side, containing buttons for buying and selling items, closing the trade menu and in the case of the dungeon trade menu, a slider with which the asking price for an item can be adjusted.

In the usability testing, these menus worked comparatively well. Participants were able to buy and sell items without requiring any explanation. When they attempted to trade profitably, multiple usability issues became apparent.

Most of these issues relate to what information is given to players and in which way this information is presented.

To illustrate these issues, it is helpful to lay out the information that is given in the trade menu prior to iteration.

The player inventory element was part of both trade menus. It displayed the items currently in the player's possession as well as the player's chosen name, their current gold and the number of items in the inventory compared to the total inventory capacity.

Settlement inventory elements had the same structure but displayed the settlement name and the settlement prosperity rather than player name and gold.

Adventurer inventories displayed an adventurer's current equipment, the gold they possessed, their culture, their class and the sum of their stats. Adventurer inventories were contained within adventurer objects. A dungeon may contain up to eight different adventurers each with their own adventurer object. Adventurer objects can be scrolled through.



Figure 57: RPG-Trader Initial State - Settlement Trade Menu



Figure 58: RPG-Trader Initial State - Dungeon Trade Menu

When an Item was selected, following information was displayed in the center item information section:

- item name
- item price
- the items damage/defense statistics including magical effects
- the culture by which the item was produced
- item tier (comparative quality level)
- the class of adventurer for which the item is best suited

The item information section also contained a contextual dialogue text element, intended to inform the player about the system status in certain cases. This dialogue field was used to ask the player to select an item when none is selected or alerts them if a transaction impossible for any reason.

When an item was selected in the dungeon trade menu, the adventurer statistics displayed beneath adventurer objects transformed and displayed the statistics the adventurer would have if they equipped the selected item.

This served to inform the player how an item would affect the adventurer's statistics.

Lastly there was a small information element that appeared when the bargaining slider element in the dungeon trade menu was moved. This element expressed the likelihood of a trade offer being accepted.

Missing Edge Cases

Players were not adequately informed about some cases that could occur when attempting to sell an item.

The first of these cases occurred, when the offered item was identical to an item the adventurer already owned. It was defined in game design that offers of this kind would never be accepted.

When an item is offered, it is displayed in the adventurer's inventory as if they had it equipped already. Because of this, the item the adventurer currently uses is not visible.

The user must first unselect the item they are currently offering to be able to tell that this item is the same as the adventurer has equipped. A lack of visibility of a specific state of the interface causes confusion in this case.

The second case occurred, when a player sold multiple items of the same kind to the same adventurer, in the same trading session. This case occurred usability testing. This was not an interface issue primarily but an oversight in game design.

It was decided that it would only be possible to sell one item of a kind to each adventurer, each trading session. A trading session would start when a player arrived at a location and end when travel to another location is initiated.

This change in the game mechanics would also have to be accounted for in the interface.

Offer Success Chance Visibility Issue

The bargaining offer success chance text element was overlooked by both usability testing participants. The text element is displayed in Figure 14, next to the bargaining slider handle on the right-hand side.

This element was comparatively small and could easily be covered by a player's finger when the bargaining slider was moved. The bargaining offer success chance was important information. This information was not visible enough, which constituted a lack of compliance with the eighth usability heuristic.

The usability of this element was reduced further because it only appeared when the slider is adjusted, therefore requiring users to recall its value if it was perceived at all.

The approximate success chance of an offer could be deduced indirectly by checking whether the offered item matched an adventurer's attributes or improved their statistics.

The information necessary for this deduction was available in the interface.

To check if an item was appropriate for an adventurer's class or culture players needed to compare the icons in the item information section with the icons above the inventory of an adventurer.

To check if an item improved the statistics of an adventurer, players needed to look at the adventurer's statistics, remember them, select an item and then calculate the difference between the stats with and without the item. This requires recall rather than recognition thereby failing to comply with the sixth heuristic. The process was also tedious.

Arrow icons indicating whether an offered item was better or worse than the one the adventurer had equipped were added to the design in prior work to improve this situation. These arrows are not visible enough and represent difference in statistics without nuance.

Adventurer Stat Bar Transition Issue

If a weapon is offered to an adventurer, the adventurer bars transformed in order to display detailed information about the physical and magic attack components that made up the total attack value. The bar element representing the adventurer's defensive statistics disappeared. There was not enough space available to display both.

If a defensive item (armor/shield) were offered the opposite would happen.

This was an issue because if an adventurer were offered a shield, while having a two-handed weapon equipped, the two-handed weapon would be removed from the adventurer's equipment. This would result in an increase of defensive and a decrease of offensive statistics. The player could not be adequately informed about this circumstance because only the defense statistics bar would be visible.

Lack of Profit Feedback

Players needed to remember the buying price of each item they acquire and recall it when they sold it. This is difficult to do in general and impossible for most people beyond a certain number of items, especially if the game is put down and picked up later.

One participant of the usability testing decided to manually document the prices of all the items they bought to circumvent this issue.

This information should not need to be recalled or manually documented. Buying price, selling price and/or the resulting profit ought to be displayed in the dungeon trade menu.

Lack of Bargaining Slider Precision

There was a usability issue with the bargaining slider element. With this element, the asking price could be set in the dungeon trade menu.

This slider did not allow for precise adjustment. Participants of the usability test were unable to set the offering price to the exact value they desired in some cases.

The bargaining slider had a fixed physical size, but its maximum value was set to the total gold available to the current adventurer. This meant that the value range changed significantly depending on the wealth of an adventurer.

The issue cannot be linked to a lack of compliance with a usability heuristic. As it may cause frustration and reduces accessibility and general usability, it is still worth mentioning.

Summary

The basic functionality of the trade menu interfaces, their overall layout and interaction flows as they presented in this version were usable for usability test participants.

Yet a lack of essential information and many small usability issues make it difficult to interact with the game system in an efficient and enjoyable way.

If these issues are solved, the intricacies of the game design may be more easily grasped and appreciated by players, which should increase the overall enjoyment.

5. Design

This chapter is focused on the processes of designing the next iteration the existing features of RPG Trader and the additional features necessary to achieve alpha status.

In the first section, the general design processes that were applied to the iteration of Map View- and Trade Menu Interfaces and the design of Bandit Ambush- and Progression Interface are described.

The second section is about the discussion of the solutions developed and decisions made in the process of applying the described processes to each of the four mentioned parts of the RPG-Trader interface.

The design of the main menu and intro sequence will be discussed briefly as well but it must be noted that these two aspects of the RPG Trader interface did not receive a proportionate amount of attention due to time constraints.

The design and implementation process does not include the entirety of the RPG-Trader interface as it was envisioned in prior development. Character Screen, Dungeon View and Monetization Interfaces could not be included in the scope of this work.

5.1 Design Process

Throughout this work a user centered design cycle methodology was applied. Of the four major sections of the RPG-Trader interface that are focused on in this work, two were in the stage of iterating functional feature based on usability issues detected in the evaluation phase.

The other two were at the beginning of the cycle, needing to be developed from pre-existing interaction flows of a feature into a new functional part of the interface.

The first step in either case was to re-state what the purpose of a feature is. Purpose in this context meant, what would the feature allow the player to do and why does the player need to be able to do it.

This purpose had already been defined in prior work, when the decision was made to include the feature in question, but it was considered useful to start by explicitly specifying the purpose of each feature that was designed or redesigned in this work.

5.1.1 Iteration of existing Features

The iteration of existing features was a process focused on finding design solutions for the usability issues revealed in the evaluation phase. It started by collecting the usability issues found in evaluation related to part of the Interface element in question. These issues were then brought into context with the interface elements behavior and appearance to trace the issues down to their likely root cause.

When issues and likely causes were considered understood. Possible solutions were conceptualized textually and occasionally with quick sketches.

From all the solutions and possible improvements that could be conceived, those with the largest possible positive impact on usability and user experience, relative to the workload their implementation would likely produce, were selected.

Some of the selected solutions required the adjustment of specific values or the modifi-

cation of smaller components. These could be implemented directly in unity and did not require additional wireframing and visual design.

Other solutions required the redesign of complex interface elements or groups of elements. In those cases, the redesign process was the same as the process used for the new, additional features as it is described in the following section.

The only notable difference is that there were existing assets and wireframes that could be used as a starting point.

5.1.2 Design of new Features

The design process for the new, additional features started by reevaluating the assumptions made, when the interaction flow for the feature in question was first developed in prior work.

Over the course of development, aspects of the overall information architecture had changed, and individual interaction flows had to be updated. This also included the definition of the information the interface in question had to contain.

When this first step was concluded the wireframes of the new interface were constructed to find ways in which all the necessary information and interactive elements, could be fit on a screen.

This was a process of assembling individual elements, often out of the same components that were used in the already existing parts of the interface, fitting them into the same grid, that was used for the rest of the interface and adjusting sizes and layout to generate quick variants. An example for a typical wireframing workspace can be seen in Figure 15.

The best arrangement was then selected. Wireframes for every state the interface and its individual components could assume, were produced this way.

Animated state transitions are an important aspect of the RPG-Trader interface style. The ways in which the various states of an

interface element could be transitioned into one another were a constant consideration during wireframe production.

The wireframes were then used as a reference layer on top of which interface art in the style of the RPG-Trader interface was drawn.

The visual design process was quick for most elements. Basic aspects of the visual design, like colors, font, text sizes and general interface element shapes had been defined in prior development.

For some interface elements significantly more effort was required to arrive at an acceptable visual appearance. Figure 16 is an example for a workspace in which an element of this kind was designed.

The more difficult interface elements generally required the production of many variations until they fulfilled their purpose in a visually pleasing way.

Exports of the work were made at various points throughout the process to examine the appearance on the target platform.

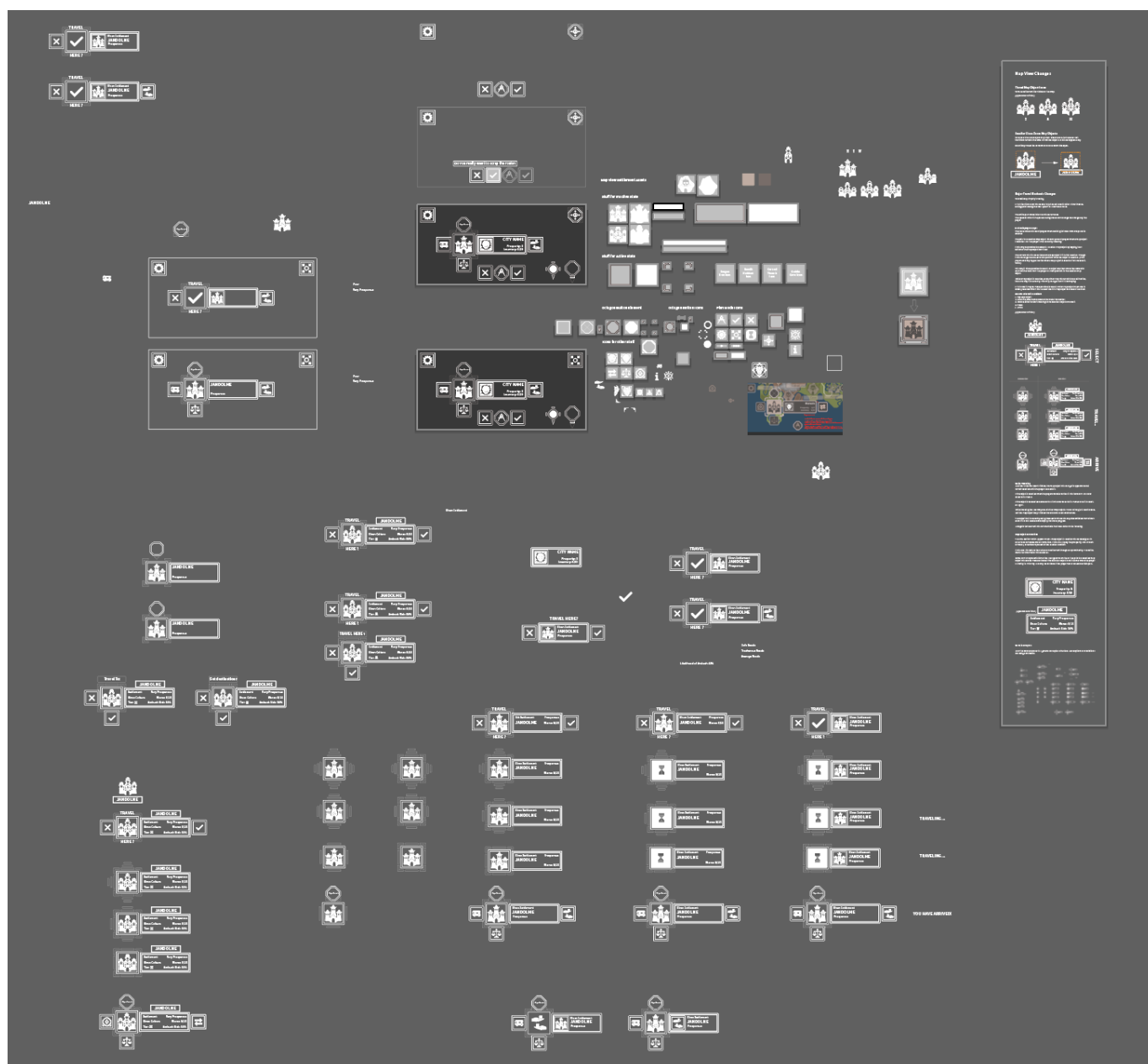


Figure 59: Map Object Redesign - Wireframing Workspace Example



Figure 60: Map View Iteration - Information Window Visual Design Workspace Example

5.2 Map View Iteration

The Map View is the part of the RPG-Trader interface with which the player can facilitate movement from location to location around the game world thereby enabling the “travel” aspect of the core gameplay loop. It is also the “base layer” of the interface from which most other parts can be reached.

The evaluation made it clear that this part of the interface has severe usability problems. The following section of this work is about the design solutions found and the thought processes, considerations and reasoning behind the changes that were made.

To preface the map view redesign process discussion, it is necessary to take inventory of the various UI elements it consists of.

Firstly, there is the map asset in the background it represents the game worlds physical size and shape. On top of this asset there is the road system connecting the 80 settlement and dungeon locations present in this game world.

Settlement and dungeon locations marked by interactable settlement or dungeon map objects. There also is the player map object marking the current location of the player character in the game world.

On another layer on top of the other elements there were three buttons. The button on the top left leads to the settings menu, the toggle on the top right centers and locks the camera on the player location, the button on the bottom activates the planning mode.

To improve the usability the design of most of these elements needed to be changed or redone completely.

5.2.1 Travel Flow Redesign

The first major change that was decided on, was to modify the planning and traveling interaction flow to match user expectations.

In the new version, players would be able to tap a map object and immediately be presented with a route to this destination and the option to travel to travel there directly.

This interaction simplifies the traveling process by removing the necessity (and possibility) of planning a route manually.

It also reduces the number of interactive objects displayed on the map. The plan mode button on the bottom of the screen could be removed, and the node objects that were displayed at each selectable intersection along a route were no longer necessary.

This helps with the issue of overlap and makes the travel planning process much less prone to user errors.



Figure 61: Final Build - Map Icon Object

State: Player Present



Figure 62: Final Build - Map Icon Object

State: Player not Present

5.2.2 Map Object Rework

The visual appearance, states of the object, animated state transitions and interactive elements of map objects need to be changed to accommodate the new interaction flow. An initial concept for these changes was developed at this stage as well.

The initial map object assumed three major states. The default "idle" state when existing on the map without being selected, a selected state for objects at the locations at which the player is currently present and a selected state for objects at locations, other than the one the player is currently present at.

The new map object needed different states now.

Map objects on locations, other than the one the player is currently present at, needed to include interactive elements allowing players to accept or decline an offer to travel there. The final appearance of this state can be seen in Figure 18.

The ability to accept travel would suffice to achieve the intended functionality since selected objects could be deselected by tapping them again but it was decided to add a decline button as well. This way, the two options the player could choose at this point of the flow, would be obvious. To maintain consistency, the player present state received a deselect button, as can be seen in Figure 17.

When players pressed the accept button, travel would start immediately.

While traveling, the destination cannot be changed. This is not optimal, because it can be assumed that players would desire to change their destination during travel in various situations.

The inability to do so could be received negatively. The technical implementation of the necessary functionality was not possible within the given time frame.



Figure 63: Final Build - Map Icon Object
State: Selected while Traveling



Figure 64: Final Build - Map Icon Object
State: Locked Destination



Figure 65: Final Build - Map Icon Object
State: Locked Destination Mini



Figure 66: Final Build - Map Icon Object
State: Selected Mini

Therefore, it was decided to allow for the selection of other map objects while traveling only to view information about them for the time being. This way players could at least look for potential destinations to travel to after they had arrived at their current destination.

It was necessary to design additional states for the map object to communicate the described functional possibilities as well as their limits.

This included one state expressing that a destination was currently “locked in” and being traveled to and another showing additional information, the same way it would when in the regular selected state but without the option to travel there.

Additional states for locked travel destinations and possible destinations were included. These states offered the same functional possibilities but did not include the information element that would usually appear upon selection. The minimized states were included to make it possible to minimize the area these objects would overlap in their selected state to get a better view of the route.

With these additional states, the current system status should remain visible at any point of the traveling process.

5.2.3 Status Text Field

To convey the system status more explicitly a non-diegetic text element was conceptualized. This text element would be used to ask the player whether they wanted to travel to a selected location. It would also display the destination while traveling and announce the arrival at a destination. This way the player would be informed about the state of travel regardless of the current camera position.

5.2.4 Route Display and Camera Controls

Players should still be able to view the route between their current location and a possible destination when selecting a map object. To make this possible with the new travel planning interaction flow, changes to the

camera controls were conceptualized. In the initial pre-alpha implementation the camera would be centered on a map object and the player camera controls were disabled while the object remained selected.

This now had to be changed. If the camera position were to be locked while a map object is selected, players could not move the camera to view the route after they selected a destination.

It was decided to still center the camera on map objects upon selection but release it as soon as it was fully centered. This way the map object interface element is in a convenient position relative to the devices screen initially, but players can immediately move the camera to view the generated route before deciding to travel.

5.2.5 Center on Player Function

The center on player function can be activated with a toggle element on the top right. This function would center the camera on the location of the player and disabled camera controls until the toggle was released. This confused one of the participants of the user testing, mainly because they were unaware that the function remained active after it was pressed rather than centering on the player character and then releasing

automatically.

The function behaves this way to give players the possibility to have the camera follow the player character while traveling, so they would not have to move the camera along manually. This functionality was still considered desirable, but its execution had to be reconsidered.

The first part of the improvements to the center on player function was to improve its appearance to make its current state more visible.

The second was to make the center on player function release automatically once it is fully centered, when the player is not currently traveling. This way the camera controls would never remain locked when there is no practical reason for them to be. Additionally, the camera lock would be released if the player selected a map object while traveling. If players select a map object while traveling it can be assumed the wish to look at this map object, rather than their character traveling.

5.2.6 Overlap Prevention

Even with the improvements to the interaction flow as described, it was evident more measures needed to be taken to reduce overlap, as can be seen in Figure 23, and improve clarity and usability.



Figure 67: RPG-Trader Initial State - Map View Overlap

It was first considered to decrease the visual size of map object elements across the board, while keeping their interactive surface the same size as before. This would reduce the visual overlap without making it too difficult to select objects.

It was decided against this because a reduction in map object size, significant enough to prevent overlap at any zoom level made both the icon objects and their text label difficult to read. This would have resulted in reduced accessibility and would generally be considered bad design practice.

There were two areas in which the overlap of map icon objects was most detrimental. Firstly, when the camera was zoomed out as far as possible. Secondly, when the camera was zoomed in halfway, to the point at which the map objects transformed from their reduced state without a text label into their default state with a text label.

In both states overlap would occur with objects demarcating locations that were particularly close to each other “geographically”.

Conceptually there are multiple ways of preventing this overlap from occurring other than reducing the object size.

The following were considered:

- restricting the camera zoom further so that the first described overlap scenario could not occur.
 - » this would reduce the maximum amount of map objects that players could see and interact with at once and make moving the camera across the map more difficult.
- moving the location of dungeons or settlements farther apart
 - » this would result in a cascade of changes and adjustment to the complex route system thereby causing significant workload for the team
 - » it would also put limit on the physical distance between settlements and dungeons thereby restricting game design possibilities
- readjusting the zoom threshold at which map objects change from their large state to their reduced state.
- reworking the icons themselves to concentrate the same information content in comparable readability, but with a smaller footprint.



Figure 68: RPG-Trader Initial State - Travel Destination Overlap

The last two options did not appear to have obvious down sights, therefore they were chosen to be applied in the new iteration.

Zoom threshold adjustment only required the changing of one value in each of the scripts handling the scaling of map icon objects. Reworking the icons on the other hand was a larger process which is described in detail in a following, section.

Overlap also occurred when map objects were selected and “unfolded” to reveal additional information and sub men access. In this case they would necessarily overlap with most map objects on the screen.

In accordance with the interface style, as defined in development preceding this work, most interface elements consisted of two colors, #544F4E for backgrounds and #BFACA2 for menu icons, text and decorative menu elements. Map icon objects used these colors as well to maintain visual consistency. When selecting a map object, the colors are flipped to differentiate the selected object. At the same time all other map objects would become transparent.

In certain states they would also receive a frame element to create clear separation between the similarly colored selected and all the other, unselected elements.

All these measures were still considered effective for increasing clarity by visually lifting elements that are focused on into the foreground. Therefore, they were also applied to the next iteration. In addition, it was decided to add a subtle 8pixel drop shadow to frames in order to increase contrast further and make specific objects appear physically above others.

Lastly there was another set of overlap issues occurring between the map object representing the current player location (referred to as payer map object in the following) and other map objects in the selected or locked destination state.

When players travel to a destination, the player map object will always overlap with the map object of the destination.

This is because the destination map object covers an area larger than the destination location coordinate, which is just a point at its center. It covers part of the path that the player map object travels on which necessarily leads to the kind of overlap that can be seen in Figure 24, at the last stretch of any journey.

In this situation the player map object can either be rendered in front of or behind of the map object. Neither of these situations is desirable, because in either case overlap occurs and one object becomes temporarily invisible, thereby hurting the visibility of system status.

To solve this, it was considered to make either of the objects become transparent while the overlap occurred. This reduces the visibility of system status as well, albeit to a lesser extent.

It was also considered to make the player “arrive” at the destination just before the overlap occurs. This was difficult to accomplish from a technical standpoint. It also must be considered that map objects are scaled to always remain the same relative size when the camera is zoomed so the area of overlap changes depending on the camera zoom level. This ultimately meant that the same journeys could be longer or shorter depending on the zoom level.

The solution that was to render the player map object behind selected map objects and destination objects but make it transition into a minimized state shortly before it would overlap, while simultaneously activating another player map object connected to the destination, indicating that the player is currently in the process of traveling there.

This way the player would always see where they player currently is, and where they are traveling while overlap was avoided in almost all cases. The this overlap prevention measure can be seen in the Figures

As additional, small improvement, the center on player toggle and main menu button elements were to be moved to the very edge of the screen.



Figure 69: RPG-Trader Final Build - Player Map Object before overlap occurs



Figure 70: RPG-Trader Final Build - Player Map Object when overlapping with destination

5.2.7 Information Window Rework

The information window, that would appear in certain states to offer relevant information about a settlement or dungeon, was also reworked.

It previously presented its information content with bad hierarchization and redundancy. The icon representing a settlements culture was removed.

This information was already contained in the settlement icons. The new settlement

and dungeon icons also included a label in which the name of the location would be displayed. This meant the name text element could also be removed.

There was more space available now to display relevant information about locations. For Settlements, this information was the settlements prosperity value, the risk of traveling there and the number of items in the settlement.

For dungeons it was decided to include the sum of gold all adventurers at the dungeon

location possessed. This would give players a general idea of the profit potential at a dungeon. It also included the risk of travel information and the number of adventurers at the dungeon.

As a result of the removal of name and culture icon the info window could now be made smaller as well. This would help reduce overlap issues.

5.2.8 Map Object Icon Rework

In the process of solving various usability issues with the map view it was decided to improve the icons representing settlements and dungeons within the game world. These icons formerly consisted of a text label and with the name of a location and an icon meant to represent either an elven settlement, orc settlement or dungeon.

The primary issue with these icons was, that they were larger than they needed to be while containing less information than they could.

The text label beneath the icon increased the icons size significantly which contributed to overlap issues. This label was reduced in size and incorporated into the icon. This made the icon occupy much less space thereby helping reduce the overlap issues.

The icons themselves were redone to also represent the Tier of a location. In the RPG-Trader game concept the map is divided in different zones. These zones are differentiated by the tier of the locations within them. Players always start in a tier 1 zone in which cheaper items can be traded with less wealth adventurers and weak bandit attacks.

There are three tiers for locations. The higher the tier, the higher the ambush risk in the region and the higher the profit potential. The tier mechanic is connected to the progression system of the game.

The tier of a location is important for players to know. Players need this information to be able to make educated decisions when traveling around the game world. This information was not apparent from the icons representing the location on the map.

This was corrected by including the tier of a location within the icon design. The tier would now be reflected by the number of spires in a settlement icon and the number of spikes protruding from the skull element of dungeon icons.

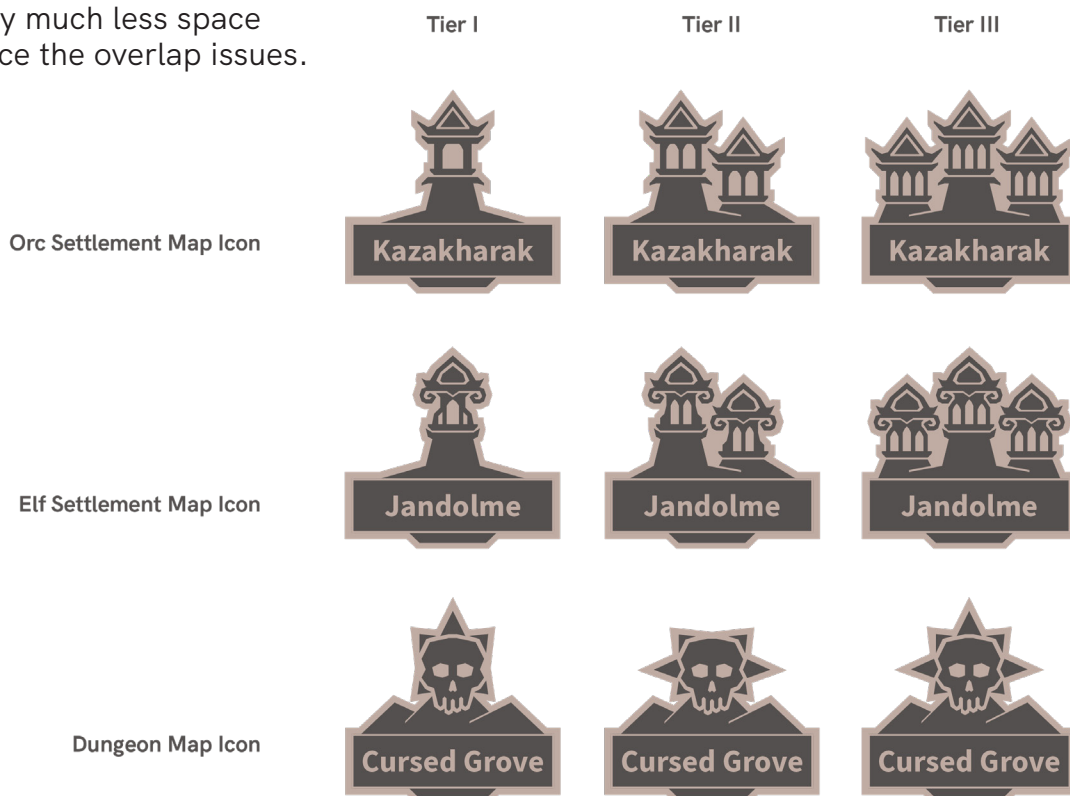


Figure 71: Redesigned, Tiered Map Object Icons

5.2.9 Button and Toggle Rework

It became apparent while working on the map view interface components that the look and feel, internal consistency and iconography of buttons and toggles in the RPG-Trader interface left room for improvement.

It was decided to overhaul these interactive elements to make their function more easily discernible for users and make them more satisfying to use. This was intended to improve usability and user experience across the interface.

There are different kinds of buttons in the RPG-Trader Interface.

- Icon buttons, generally for opening and closing menus.
 - » The icons of these buttons were inconsistent in appearance and were not as readable as they could be.
 - » The buttons looked too similar to non-interactive interface elements
 - » The buttons were internally inconsistent with the style of “buy” and “sell” buttons within trade menus
 - » The button feedback was not noticeable and satisfying enough
- Text buttons, generally for actions within menus
 - » These Buttons were only used for buying and selling interactions and had a special style.
- Icon Toggles, for functions that were switched on until they were switched off
 - » the “activated” state of toggles was not visible enough.

A new style for button elements was established. Instead of a flat dark background the new version used the same style as the metallic frames.

When pressed the shadow, and highlight of this new background would be swapped creating the appearance of the button being pressed in. This improved the feel of pressing the button and created a clearer separation between buttons and non-interactive objects.

In addition, new icons were made. To make the icons more readable, symbols commonly associated with the concept that was to be symbolized by the button were researched and included whenever possible.

It was decided to keep the style of the existing buy/sell button the same but only use it for action text buttons with functions related to the exchange of currency.

Another action text button style was established with the same background and feedback as the icon buttons.

This button style would be used for buttons unrelated to the exchange of currency.

Toggles would also share the background style with icon buttons, but they would receive a dark background upon activation, to create stronger contrast between the inactive and active state while maintaining visual consistency.

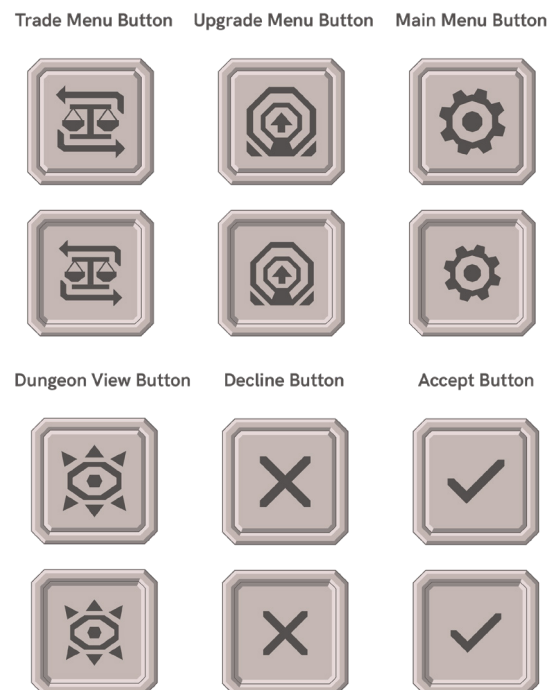


Figure 72: Redesigned Buttons

5.3 Trade Menu Iteration

The trade menus, Settlement Trade Menu and Dungeon Trade Menu are the parts of the interface enabling the buying and selling aspects of the core gameplay loop, respectively. The Settlement Trade Menu can be opened by players when they are at a settlement location. Within this menu players can buy wares.

To preface the following discussion of the improvements made to the trade menu interface its structure is briefly reiterated here.

The settlement trade menu consists of two inventory Interfaces, one for the player inventory one for the settlement inventory. Above each inventory, relevant data about its owner is displayed. When an item is selected in either inventory, relevant information about this item is displayed in the center. With the buy and sell buttons on the right-hand side items can be bought by the player or sold back to the settlement in as long as players change their mind about purchasing an item before traveling to another location.

These wares are then sold to adventurers through the dungeon trade menu which can be opened when players are at a dungeon location.

The Dungeon trade menu has a similar structure but instead of a settlement inventory it has an adventurer selection element. In this element the adventurers present at a dungeon, and their stats and equipment can be viewed. Players can select items in their inventory to offer them to adventurers. They can also adjust the asking price with a slider on the right-hand side.

Various changes and additions were made to improve the usability and utility of these trade menus.

5.3.1 Value Difference Indication

Both trade menus required a form of value difference indication.

Differences in value could be indicated abstractly, with arrows pointing up or down. Arrows of this kind had been used in the dungeon trade menus to indicate if an offered item increased or decreased the statistics of an adventurer.

One problem with this abstract indication was, that it did not inform about the extent of an increase or decrease. The extent could also be indicated abstractly, through the size or shape of an arrow or by using multiple arrows for example. This could offer an approximate indication of how much cheaper than average a specific item was in the settlement trade menu for instance.

It would not inform about the magnitude of an increase or decrease. If a cheap item were three times cheaper than average, the total gold amount saved on the purchase would be much lower than for an expensive item that is three times cheaper than average.

Another problem with the arrow indicators was, that they were the same color as most other elements within its interface. This meant they did not catch the eye.

It could be expected that players would select through one item after another to scan the inventory quickly before making purchasing decisions for example. This meant that any indication that could not be perceived immediately would likely be missed.

There were many similar cases in which the player would benefit from an easily perceivable indication of a given value being higher or lower than another or a factor being positive or negative. A solution had to be found that could consistently be applied throughout the RPG-Trader interface whenever such cases occurred.

What was needed first was way of expressing the general concepts of positive and negative, higher or lower. It was decided to use a simple color code to that end.



Figure 73: RPG-Trader Final Build - Settlement Trade Menu



Figure 74: RPG-Trader Final Build - Dungeon Trade Menu

Green would be used for positive factors and values and red for negative ones. In some special cases gradations between the two could be used. This color code was deemed most likely to match user expectations.

It must be mentioned that red and green were already used in the interface to represent the magic effect types of fire and earth. This meant that the same colors were now used to represent different concepts within the interface.

It was considered to change the colors for fire and earth magic effects, but no suitable alternatives could be determined in time.

It was planned to overhaul the magic effect system interface representation and game mechanical implementation for reasons that fall outside of the scope of this work. This overhaul would affect large parts of the existing interface and would have caused excessive workload. It would have to be revisited after this work has been concluded.

The described color coding pattern was not only applied to the trade menus. It would also be applied to every other element that expressed something positive or negative. Color coding was also used to indicate how high the risk of an ambush on a given route in the map view for example.

In addition to color coding, it was decided to add strings expressing the numerical value of an increase or decrease to value text elements throughout the interface. Increases would be made identifiable as such by adding a plus symbol decreases would receive a minus symbol.

This way players could unambiguously tell if a value would increase or decrease or if currency was lost or gained. The exact magnitude of increases or decreases and whether an increase or decrease was beneficial in any given circumstance would be discernible as well.

The described measures were expected to make it easier to spot potentially beneficial trades in any trade menu. When players scroll through adventurers or select through items in their inventory, they would immediately be able to tell favorable from unfavorable constellations by the amount of green and red in their field of vision.

How color coding and value difference indication appeared in the trade menu after implementation can be seen in Figure 29 and Figure 30.

5.3.2 Offer Success Chance Improvements

Testing participants did not see the offer success chance text element while using the bargaining slider in the dungeon trade menu.

To solve this issue this text element was moved to the center of the interface and its size was increased significantly. It would now be displayed as soon as the player selects an item to sell in their inventory and remain visible until the item is deselected or sold.

The challenge when redesigning this element was to find a space in this menu to place it. It was evident that the element had to overlay part of the menu if it were to be displayed larger. It was decided to place it on top of the item artwork display in the center because the item artwork was visible at two more locations within the menu when selected. The loss of relevant information would be minimal this way. Placement and final appearance of the offer success element can be seen in Figure 30.



Figure 75: RPG-Trader Final Build - Dungeon Trade Menu Bartering Chances Indication 1



Figure 76: RPG-Trader Final Build - Dungeon Trade Menu Bartering Chances Indication 2

5.3.3 Bartering Chances Indication

In the Dungeon Trade Menu, the player was informed of the success of an offer by an animation overlay abstractly suggesting an attempted handshake that is accepted or declined. The player was not informed how often any adventurer had declined their offer and if the adventurer had become unavailable for trade.

To inform players sufficiently about these factors they needed to know how many attempts they had in total and many had already failed. This information had to be connected to each individual adventurer representation.

It was decided to include the information in the adventurer representation objects. The only place within these objects that was suitable, was the adventurer data component at the top. This component contained the name and gold in possession of an adventurer, as well as icons representing their class and culture.

It became clear that this data section had to be resized to make room for an additional element. This increased the total height of the Adventurer representation, which made it necessary to resize the adventurer statistics bar object below.

There are many conceivable ways in which the number of failed offers can be expressed. It was first considered to add a number that decreases or increases with each declined offer.

It was not possible to fit such a number into the available space without scaling it down below the minimum text size, necessary to maintain readability.

It was decided to represent the number of failed offers per adventurer with boxes, with a "x" symbol in the center instead. Four of these boxes, one for each possible attempt, could easily be fit in the available space.

When an offer was unsuccessful, one of the boxes would become red. When all four attempts were spent, the adventurer representation would receive a dark, transparent overlay, as can be seen in Figure 34. This was intended to inform the player that an adventurer had become unavailable and also communicate why the adventurer could not be traded with any more.

5.3.4 Adventurer Bar Adjustment

There were bars below adventurer objects, displaying the current adventurers total attack and defense values. These bars had an animated transition from a state in which both bars were visible into a state in which only one was visible with more detail. The states would change when an item was selected to focus on either defense or attack values. This behavior was determined to be counterproductive in the usability analysis.

The bars size position and animated behavior was adjusted. Both bars would now always remain visible. When an item was selected the value text component would expand and a additional, color coded string would indicate how an item affected the adventurer statistics.

5.3.5 Alert and Info Dialogue Improvement

Menus in the RPG-Trader interface all have an alert and info dialogue element. This element serves to inform the player if they can and cannot do specific things in certain situations and why. This element can be seen in Figure 34, for instance.

Throughout the design of the RPG-Trader interface, care was taken to make almost all things, explained in these dialogue objects, discernible from looking at the interface alone.

The alert and info dialogue element are a redundancy to ensure that users never have to guess why something they thought possible, is not. They also serve as a form of tutorial, by unobtrusively explaining game mechanics that may not be immediately obvious.

In the initial build some cases in which this dialogue needs to offer explanation had been missed. It was made sure that these cases were accounted for in the implementation phase.

5.4 Ambush Interface Design

When players travel from one location on the map to another, they may be confronted by a bandit ambush. The bandit ambush mechanic is intended to make the process of planning routes and traveling more engaging. It adds an element of risk to travel.

How high this risk is, depends on various factors, like the length of a route, the value of items transported and the region that is traveled through. Players must think about the destination they choose, how valuable their freight is and how much they invest in guards to protect themselves. It is an important part of the game concept.

5.4.1 Ambush Interface Flow

The design of the ambush interface started by revisiting the interaction flow, that was produced in prior development. This interaction flow was incomplete and had to be updated.

The flow of interaction with this interface starts with an ambush overlay appearing, regardless of what the player is doing in the current moment. Other menus or unfolded map objects are closed, and the player camera is moved to the current player location.

It was first considered to allow players to resume their activities but be notified of an ambush occurring and informed about the necessity to resolve it before travel can continue. This would have been less disruptive.

It was decided to make the flow disruptive on purpose, to support the notion of immersion within the game world instead. The player is a trader traveling through hills and forests. Suddenly the player is ambushed by dangerous adversaries demanding they hand over their valuables. They would not be able to finish what they were doing before reacting to this situation.

The ambush overlay would need to inform the player about the situation at hand. It would also offer two possible ways of resolving it.

The first being, to pay the bandits off with gold and be let go unharmed. This option would not be given if the player did not have enough gold. Instead, they would be informed about their lack of funds and only be able to select the second option. This option would be, to attempt to fight the bandits off instead.

For players to make an educated decision which of these options they would choose, if there was a choice, they would need information.

This information was defined first because it was the bulk of what the interface would consist of. Relevant information is how much gold the player currently has, how much it would cost to pay the bandits off, the number of items in their inventory and how their crew compares to the ambushing bandits in terms of fighting ability.

This fighting ability is defined by the hit points each party has and their respective attack and defense values.

More detailed forms of comparison could be made available from a purely technical standpoint. It was decided not to offer any more information to allow for a measure of uncertainty and thereby make the decision-making process more engaging.

When players decided to attempt to fight, an animation would play to visualize the battle. The representation components would collide to visualize an attack, as shown in Figure 36.

The fight could be won or lost. In either case players would be presented with a final stage of the interface, informing about the outcome.

If the fight were lost the player would lose gold and items. How much gold and how many items were lost would also be displayed.

If players decided to pay the bandits, the outcome stage would inform about the amount of gold lost.

When players felt sufficiently informed about the outcome of the encounter, they could close the ambush interface and resume travel.

5.4.2 Ambush Interface Wireframing

As soon as the interaction flow was defined, the wireframing phase could begin. The various pieces of information were combined into interface elements. The interface consisted of three major components. One representing the player and their crew on the left.



Figure 77: RPG-Trader Final Build - Ambush Interface



Figure 78: RPG-Trader Final Build - Bandit Ambush Fighting Visualization

One component representing the bandits on the right and a third in the center, offering information and interactive elements.

The wireframes were drawn on top of the wireframes of a trade menu to ensure visually consistent with other parts of the RPG-Trader UI.

At the same time, the ambush interface was designed to appear different than other parts of the interface. It not a menu, like the trade menus for instance, with frame and background covering most of the screen. It was more akin to a popup, demanding attention but leaving more of the map view visible.

The ambush interface consisted of multiple, smaller framed components. Each of the components could be transformed individually allowing for more flexibility when animating the buildup of the interface and the combat visualization.

It was not necessary to export any special assets to be able to assemble the ambush interface. It was designed to consist only of elements that could be constructed out of existing scalable assets and prefabs. This made the process of implementation faster.

5.5 Progression Menu Design

Progression in the RPG-Trader is based on a simple positive feedback loop. Players trade profitably. They can use their profits to buy upgrades. These upgrades help them to trade even more profitably.

The progression system is essential for the game to be engaging. There is little motivation to try to increase profits if there is nothing players can do with the gold they earn. The purpose of the progression system is to give players something to work towards.

The progression menu is the part of the interface that allows players to view, compare and buy two different kinds of upgrades.

Firstly, there are trade guild ranks and perks. The game design of RPG-Trader includes an entity called the trade guild. It is conceptualized as a corrupt organization offering competitive advantages to its members.

Players can obtain guild membership and advance through the ranks by bribing guild officials. With each subsequent rank they gain access to increasingly powerful bonuses, called Perks.

The second kind of upgrades concern the player crew. The player does not travel alone. They have a cart to store their wares in and a draft animal to pull this cart. They may also hire mercenary guards to protect themselves from bandits on the road.

As they gain more money players may buy larger, faster, stronger carts and animals and hire more proficient guards. With a stronger crew, players can carry more wares, travel faster and traverse the more dangerous regions in the game world safely. These can also be purchased with the progression menu.

5.5.1 Progression Menu Flow

In the original interaction flow for the progression system interface, crew upgrades and trade guild advancement were conceptualized to be handled by two separate menus.

Access to these menus was to be given through two buttons, one below a map icon object and one on its left-hand side. Due to the changes made to the map icon objects the left-hand side button was now used as a dedicated close button.

This changed the navigation structure in so far as that the two progression interfaces now needed to be accessible through a single button.

It was considered to add another button to the map object. Various placements for this additional button were tried but none harmonized with the present design language of the map icon object.

In its unfolded state the map object was designed to fit well within the rectangular format of a smartphone in landscape orientation. Button components were placed at deliberate locations around the object. Buttons on the right-hand side lead “forwards” within the core gameplay loop, either by initiating travel or by opening the trade menus. Buttons on the left-hand side lead “backwards” by closing the map icon object.

The player map object was placed on top of the POI map object icon component when players were present at the location. The button leading to the progression menus, was placed below the POI map object icon component, opposite to the player map object. This structure was designed to be consistent with common UI navigation patterns and express the relationship of the object’s components to one another.

There are various ways in which the two aspects of the progression system, crew upgrades and rank progression, could be made accessible through one button.



Figure 79: RPG-Trader Final Build - Progression Menu, Crew View



Figure 80: RPG-Trader Final Build - Progression Menu, Guild Rank View

One option would be a kind of two step selection, pressing the single button first, then being presented with two more buttons to choose between the rank progression and crew progression options.

If it were to be done this way, players would always have to take two actions before arriving at their desired interface and three each time they wished to switch between the two progression system components.

It was considered likely, that players would want to scan through all progression options available quickly in various instances. This would be cumbersome if menu access were structured in the described way.

It appeared to be a better option to merge the two into one combined progression menu.

There was not enough space available to display the entire range of progression options on one menu screen. Therefore, the menu was conceptualized to have two states that could seamlessly be transitioned into each other at the click of a button.

One state would contain all information and interaction necessary to view and purchase crew upgrades. The other would contain a skill tree with which the player could purchase ranks and perks.

Some of the information within the combined progression menu was relevant to both progression system components. All upgrades were purchased with gold. Therefore, the players gold needed to be visible in both states.

Both sub menus would also benefit from a dynamic information and alert dialogue, explicitly informing players if they had insufficient funds to purchase an upgrade for instance.

Most other information was relevant to either crew upgrades or rank progression.

To make educated decisions which rank or perk to buy players needed to know which ranks and perks they currently owned, what the effect of each rank and perk is, what the prices are and how much each level of a perk would increase the bonus.

To make educated decision when purchasing crew upgrade players needed to know what their crew consisted of and what their total crew statistics were.

They also needed to know which upgrades were available for each individual component of their crew, which of those upgrades they already had, what the statistics of each possible upgrade were, the price of each upgrade and if an option was better or worse than the current equipment.



Figure 83: RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Stables

This was a significant amount of information to be displayed on the available screen space. It was not conceivable that everything could be displayed at once without overloading the interface or reducing element sizes below the minimum threshold of readability.

To solve this issue the crew upgrade menu was given two states as well. The first was a default state, in which the current crew could be overviewed. The second was a “shop” state in which the upgrades for each individual crew component could be viewed and purchased.

This way the information content could be divided by contextual relevancy while keeping the design of the upgrade menu internally consistent.

Even with all the design decisions mentioned above, wireframing for the progression menu was the most difficult process of its kind thus far.

5.5.2 Progression Menu Wireframing

Many variants and adjustments were necessary to arrive at an arrangement that contained all the required information in any state of the menu, while maintaining consistency with the rest of the RPG-Trader Interface.

During the process of wireframing plans for the animated transitions between the various menu states were also made. This way it could be made sure that the interface consisted of groups of components that could be transformed in the intended way later.

5.5.3 Progression Menu Assets

Based on the resulting wireframes, the final UI Art was drawn. Besides various frame components and decorative elements, the production of the assets for the Progression menus involved the creation of a new set of icons to represent the various ranks and perks that could be acquired.

These icons also required a special set of frame states to express if they were available, unavailable or already owned in the guild rank view interface.

To keep the workload manageable these icons were designed to consist of an icon image component and a number component to differentiate the five available levels of each perk. The resulting icons and frame states are displayed in Figure 42.

When the UI Art was finished, it could be disassembled into the asset-components. These components were prepared for export so they could be used to construct the progression menu in unity later.



Figure 84: RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Cartmaker



Figure 85: RPG-Trader Final Build - Progression Menu, Crew View Sub Menu: Garrison

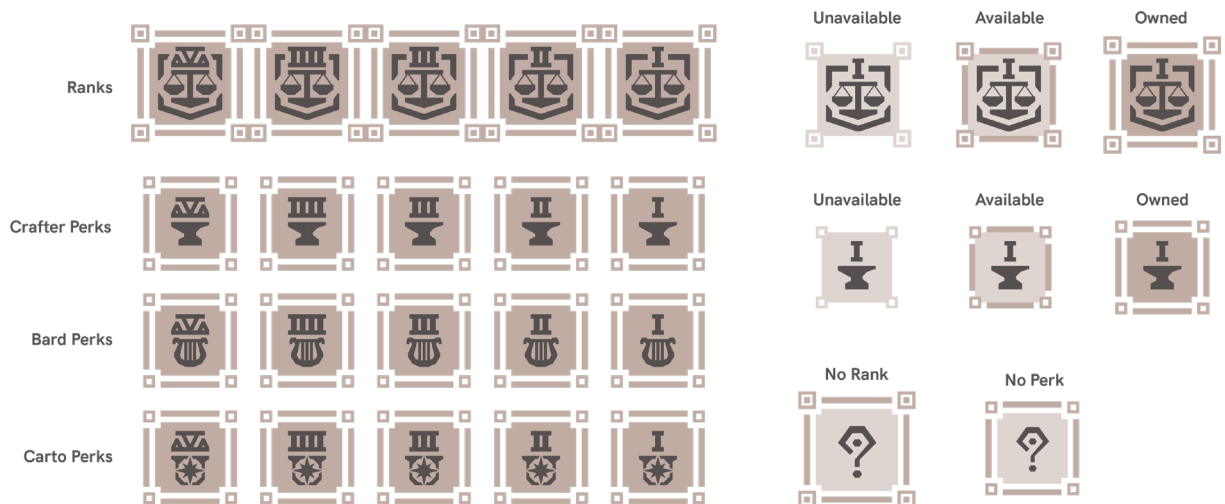


Figure 86: RPG-Trader Final Build - Progression Menu, Guild Rank Icons and Frame States

5.6 Intro Sequence and Main Menu Design

Initially it was planned to complete the interface of RPG-Trader in its entirety. As the work progressed it became clear that this would not be possible within the given timeframe.

The two non-essential features, Dungeon View and Character Screen were excluded from the scope of this work for this reason. What could not be excluded were two features necessary to encapsulate the core gameplay experience. A dedicated way of entering and exiting the experience had to be designed and implemented at the very least.

To this end a simplified version of an intro sequence and a simplified main menu were designed.

The design process of these features did not match the process described at the beginning of this chapter. The resulting design is functional but not as sophisticated and detailed as the rest of the interface. The intro sequence and Main menu are to be considered placeholders.

5.6.1 Intro Sequence

The intro sequence was envisioned as a means of onboarding first time users. It is intended to eventually include character customization, an introduction to the games narrative and a short tutorial.

None of these features are included in the version that could be realized within the confines of this work.

The purpose of the simplified intro sequence is to display the games name, offer the player choice between two possible character models, two different starting regions and the ability to set an individual name for their character when a new game is started.

This purpose would be fulfilled by an animated overlay, displaying the games name first, then transitioning to a selection screen with two selectable character artworks.

When a character is selected a text input field appears with which a name can be set. In the last screen, either an orc or an elf symbol can be tapped to choose if the player starts in orc or in elf territory.

When all selections are made the intro sequence fades out, reveals the map view and the camera is centered on the player map object.

5.6.2 Main Menu

The purpose of the main menu was to offer access to game settings, credits, the ability to start a new game and the ability to close the game.

The simplified main menu can be accessed through a button at the top left of the map view.

The menu would be a simple overlay containing a deactivated settings-button, as settings would not be implemented, a credits-button, a functional game-reset-Button, a functional quit-button and a close-button for returning to the map view.



Figure 87: RPG-Trader Final Build - Menu



Figure 88: RPG-Trader Final Build - Intro Character Selection

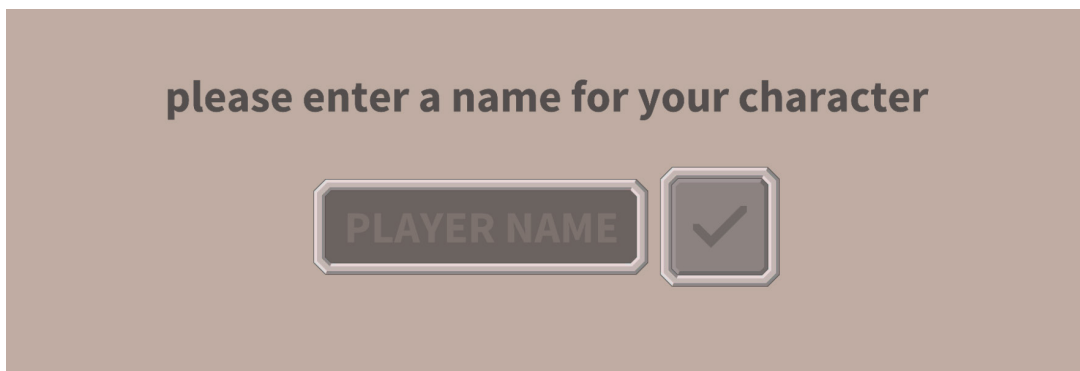


Figure 89: RPG-Trader Final Build - Intro Name Input



Figure 90: RPG-Trader Final Build - Intro Starting Region Selection

6. Implementation

The described design solutions were implemented in Unity. Implementation was the most time-consuming part of the interface improvement and completion effort.

In the following chapter the workflow and the process applied throughout the implementation phase are discussed.

The implementation can be separated into the four sub-processes Asset Export/Import, Assembly, Animation, Scripting and Bug fixing.

6.1 Asset Export/Import

When new parts were to be added to the interface, the first step in the implementation workflow was to export sprites out of the adobe illustrator files in which the visual design had been done. These sprites were then imported into the file structure of the unity engine project.

There are different ways this can be done. For the implementation of interfaces in the RPG-Trader project, sprites were exported out of Adobe Illustrator documents as individual portable network graphics in target resolution.

Most sprites were exported without color, so that the colors could be set and adjusted in engine. Care was taken not to export more individual sprites than necessary.

When importing sprite assets in Unity they can be edited to enable scaling without distortion. This was done with various assets, like frames and button backgrounds. This way a single frame sprite could be used to assemble various objects in the following step.

6.2 Assembly

Frames and backgrounds, icons and text elements were grouped, positioned, layered, colored and received functional components like animator or button components for instance.

For any interface element that occurred at multiple points with slight variations, blank prefabs were produced this way. Care was taken to make these prefabs adjustable. The UI system in -unity makes it possible to use anchor presets to define in which way combined interface elements behave when they are transformed. These anchor presets can be used to ensure that interfaces appear the intended way regardless of screen resolution.

They can also be used, to make every component of a button or frame maintain the same, relative appearance when their parent object is transformed. UI Prefabs were

prepared like this whenever applicable. This way one button prefab could be used for every button element throughout the interface for example. This was intended to accelerate the implementation process by removing the need to assemble similar components multiple times and making it possible to adjust multiple objects by adjusting a single prefab.

Sprites and components were assembled into elements, those elements into groups and those groups into larger sections of the interface.

6.3 Animation

Various components of the RPG-Trader interface would assume different states in different situations. The transitions between those states were animated using the animation system built into Unity.

The process started by creating animator controller components for every UI element that needed to be animated. Within animator controller components, animations were created and connected. Static states of an object were defined first.

The next step was to manually animate detailed transitions between these states.

With Animator Controllers in unity animations can be chained together, through animator transitions. These Controller transitions automatically interpolate between one state and another.

The transitions in RPG trader were made by hand because this allowed for more deliberate and complex behavior. The transition feature was still used, without automatic interpolation, to chain manual transitions together and create a preliminary state transition logic. This made it easier to control the behavior of animated objects with scripts later.

This process was applied to all animated objects, from the simple button click feedback transitions to complex animation state transition networks for elements like the map icon objects for instance.

6.4 Scripting

To make the interface behave in the intended way, scripts were added to various objects. Within these scripts references to the components of an element that needed to be manipulated through code and other system scripts, that contained variables that that needed to be displayed in the interface, were made. When all that was needed had been referenced, the logic was developed.

UI scripts in the RPG trader project read out system variables, apply calculations to transform them into understandable information if need be and write them into text elements and various other forms of data visualization.

They also determine the current state of the interface and facilitate transition into another state when buttons are pressed, or values change.

All UI scripts were written in the C# programming language.

6.5 Bug Fixing

The last step of the implementation process for each feature was testing and bug fixing. Errors and unintended interactions between connected interface components occurred during the Implementation process.

It was important to not only test the functionality of a given interface element that was modified or added, but also determine if the rest of the interface still behaved as expected.

Various bugs occurred throughout implementation. Their causes were as diverse as their fixes. Fixing these bugs generally required determining the root issue by working backwards, from an element behaving in an unintended way, to the variables it depended on, to scripts affecting these variables.

When the root of an issue was found it could be fixed by changing values, correcting syntax or even rewriting part of a script.

The general practice was to test everything, find any element that did not behave the way it was intended to analyze the likely root problem, making changes and testing again until most problems could be solved.

Despite the development teams best efforts there are still multiple bugs in the build that is the final product of this work. Many of these bugs are minor both in cause and effect. The Arrows at the top of the trade menu item info, point in the wrong direction and some values representing Gold currency values in the interface are missing the "G" character at the end for example.

One issue that could be categorized as a bug posed a larger problem. RPG-Trader is designed as a mobile gaming experience. This is why the final build was intended to be an application for android devices.

In build tests it became apparent that the application would not launch on the testing devices available to the team.

Multiple likely reasons for this were determined. The team is confident this issue can be solved. Possible solutions were considered but could not be applied in the available timeframe.

The final builds, that are submitted with this work are PC and Mac versions of RPG-Trader for this reason.

7. Evaluation of the Results

The goal of this work was improvement and completion of the RPG-Trader user interface in order to reach alpha stage in development.

Emphasis was to be laid on the application of user experience design methodology to ensure enjoyability.

To determine if the goal can be considered accomplished each aspect of this goal needs to be evaluated individually.

7.1 Improvement

The interfaces needed for the core game-play loop of the RPG-Trader experience were already implemented when this work began. There were interfaces for buying and selling items and traveling across the game world.

These Interfaces did not provide the desired degree of usability and therefore did not provide a good user experience.

In this work, the map view and trade menu interfaces were thoroughly analyzed using UX Design methodology to reveal the nature of their shortcomings.

New solutions were found for most of the usability issues revealed in the analysis. These solutions were implemented. An iteration cycle for the Map View and Trade Menu UI was concluded.

It is considered likely that the changes made, positively affect the usability of the interface. Subjectively, the interface does provide a much better user experience than it did initially.

An unbiased, objective statement about the improvement in usability and user experience can only be made when the iterated interface is subjected to user testing again.

7.2 Completion

With the addition of the Progression interface, ambush interface, intro sequence and main menu the RPG-Trader game contains all interface components necessary for a complete and enjoyable experience. All implemented game play mechanics are accessible through dedicated interfaces.

Item prices and statistics, in game economy, combat system and progression system have not yet been balanced. The interplay of the game's mechanics could not be evaluated before all the essential interface components had been implemented. With the progress made over the course of this work the game is now considered suitable for playtesting and balancing.

An alpha version, in the context of a software product can be defined as "the first version of a product (such as a computer program) that is being developed and tested" ([Merriam-Webster.com, 2020](https://www.merriam-webster.com/dictionary/alpha)). In this sense the RPG-Trader Game has reached alpha stage.

It must be noted that the games interface cannot be considered fully complete. Further features have been conceptualized in earlier development.

Players are intended to have access to a form of character creation, tutorial and an in-game character screen, offering an overview of data relating to the player and additional statistics about the trading performance over time for example.

There is also a dungeon view interface planned, with which players can access additional information about dungeons their inhabitants. Interface elements related to monetization are planned as well. The interface is also intended to offer audio feedback.

The missing parts of the Interface will have to be added in future development, for the RPG-Trader experience to be truly complete. For the purposes of this work, all that could be done, has been done.

7.3 Summary

The RPG-Trader project advanced from a early alpha into a alpha version. The present features have been designed using UX Design methodology and theory with a particular focus on the usability aspect.

It is considered likely that the games interface is satisfying to use and easy to learn and thereby helps to create a enjoyable gaming experience.

8. References

- "What is User Experience (UX) Design?", Interaction Design Foundation (IxDF), n.d., accessed December 5, 2020. [https://www.interaction-design.org/literature/topics/ux-design#:~:text=User%20experience%20\(UX\)%20design%20is%2C%2C%20design%2C%20usability%20and%20function.](https://www.interaction-design.org/literature/topics/ux-design#:~:text=User%20experience%20(UX)%20design%20is%2C%2C%20design%2C%20usability%20and%20function.)
- "User Centered Design", Interaction Design Foundation (IxDF), n.d., accessed December 5, 2020, <https://www.interaction-design.org/literature/topics/user-centered-design>
- "How to Conduct Focus Groups", Interaction Design Foundation (IxDF), last modified July, 2020. <https://www.interaction-design.org/literature/article/how-to-conduct-focus-groups>
- Zimmermann, Jonas. "Eine Einführung in den User Centered Design Process (UCDP)", embedded-software-engineering, January 2, 2019. <https://www.embedded-software-engineering.de/eine-einfuehrung-in-den-user-centered-design-process-ucdp-a-785958/>
- "User-Centered Design Basics", usability.gov, n.d., accessed December 6, 2020. <https://www.usability.gov/what-and-why/user-centered-design.html#:~:text=User%2DCentered%20Design%20Process&text=Design%20is%20based%20upon%20an,process%20and%20it%20is%20iterative.>
- "Usability Form (Adult)", usability.gov, accessed January 2021. <https://www.usability.gov/how-to-and-tools/resources/templates/consent-form-adult.html>
- Farrell, Susan. "Field Studies", Nielsen Norman Group, last modified October 23, 2016. <https://www.nngroup.com/articles/field-studies/>
- "Eye Tracking", usability.de, n.d., accessed December 10, 2020. <https://www.usability.de/en/services/ux-testing-research/eyetracking.html>
- "7 Great, Tried and Tested UX Research Techniques", interaction-design.org, October, 2020. <https://www.interaction-design.org/literature/article/7-great-tried-and-tested-ux-research-techniques>
- Sherwin, Katie. "Card Sorting: Uncover Users' Mental Models for Better Information Architecture", Nielsen Norman Group, last modified March 18, 2018. <https://www.nngroup.com/articles/card-sorting-definition/>

- Moran, Kate. "Card Sorting: Uncover Users' Mental Models for Better Information Architecture", Nielsen Norman Group, last modified December 01, 2019. <https://www.nngroup.com/articles/usability-testing-101/>
- Harley, Aurora. "Card Sorting: Uncover Users' Mental Models for Better Information Architecture", Nielsen Norman Group, last modified February 16, 2015. <https://www.nngroup.com/articles/persona/>
- Babich, Nick. "Most Common UX Design Methods and Techniques", UX Planet, last modified July 13, 2017. <https://uxdesign.cc/how-to-create-a-value-proposition-83994902bf38>
- Harley, Aurora. "Field Studies", Nielsen Norman Group, last modified February 25, 2018. <https://www.nngroup.com/articles/ux-expert-reviews/>
- Pernice, Kara. "User Interviews: How, When, and Why to Conduct Them", Nielsen Norman Group, last modified October 07, 2018. <https://www.nngroup.com/articles/user-interviews/>
- Gibbons, Sarah. "Journey Mapping 101", Nielsen Norman Group, last modified December 09, 2018. <https://www.nngroup.com/articles/journey-mapping-101/>
- Nielsen, Jakob. "How to Conduct a Heuristic Evaluation", Nielsen Norman Group, last modified November 01, 1994. <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>
- Nielsen, Jakob. "10 Usability Heuristics for User Interface Design", Nielsen Norman Group, last modified November 15, 2020. <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Nielsen, Jakob. "Thinking Aloud: The #1 Usability Tool", Nielsen Norman Group, last modified January 15, 2012. <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>
- Nielsen, Jakob. "Card Sorting: Uncover Users' Mental Models for Better Information Architecture", Nielsen Norman Group, last modified September 13, 2009. <https://www.nngroup.com/articles/discount-usability-20-years/>
- Nielsen, Jakob. "The Use and Misuse of Focus Groups", Nielsen Norman Group, last modified January 01, 1997. <https://www.nngroup.com/articles/user-interviews/>
- Fard, Adam. "10 Usability Heuristics for User Interface Design", UX Collective, last modified May 29, 2019. <https://uxdesign.cc/the-ultimate-guide-to-stakeholder-interviews-understand-your-clients-a3bcf87b6e8b#:~:text=What%20Are%20Stakeholder%20Interviews%3F,help%20simplify%20the%20design%20process.>

- Metha, Bansi. "Gathering UX requirements.", UX Collective, last modified Dec 14, 2019. <https://uxdesign.cc/gathering-ux-requirements-e0580d6a715a>
- Birch, Daniel. "How to create a value proposition", UX Collective, last modified September 10,2020. <https://uxdesign.cc/how-to-create-a-value-proposition-83994902bf38>
- Moorman, Jan. "Leveraging the Kano Model for Optimal Results." UX MAGAZINE Article No :882 (October 09.2012) URL: <https://uxmag.com/articles/leveraging-the-kano-model-for-optimal-results>
- Withrow, Jason. "Competitive Analysis: Understanding the Market Context", boxes and arrows, last modified February 27.2006. <https://boxesandarrows.com/competitive-analysis-understanding-the-market-context/>
- Arhipova, Alina. "Information Architecture. Basics for Designers.", UX Planet, last modified May 25.2017. <https://uxplanet.org/information-architecture-basics-for-designers-b5d43df62e20>
- Babich, Nick. "The Beginner's Guide to Information Architecture in UX.", Adobe, last modified Nov 24.2020. <https://xd.adobe.com/ideas/process/information-architecture/information-ux-architect/>
- "What is a Key Performance Indicator (KPI)?", KPI.org, n.d., accessed December 15, 2020, <https://kpi.org/KPI-Basics>
- Cao, Jerry, "How to Write a Painless Product Requirements Document", STUDIO by UXPin, last modified Dec 05.2014. <https://www.uxpin.com/studio/blog/write-good-product-requirements-document/>
- Mayer, Sandro. "The 7 most important UX KPIs and how to measure them", TestingTime, last modified Jan 29.2019. <https://www.testingtime.com/en/blog/important-ux-kpis/>
- "Alpha", Merriam-Webster.com, accessed March 2021. <https://www.merriam-webster.com/dictionary/alpha#examples>
- Google LLC, "Google Maps", Google Play, Vers. 10.64.2(2021). <https://play.google.com/store/apps/details?id=com.google.android.apps.maps> (accessed April 2021)
- Discord Inc., "Discord", Discord.com, Build Number: 82117 (2021).<https://discord.com/download> (accessed March 2021)

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit bisher bei keiner anderen Prüfungsbehörde eingereicht, sie selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Stuttgart, 16.04.2021

Fischer, Vincent