Examining the Effects of Game Design on Playability

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Software Engineering
May 2025
Kristian Wahlroos

UNIVERSITY OF TURKU
Department of Computing

KRISTIAN WAHLROOS: Examining the Effects of Game Design on Playability

Master of Science (Tech) Thesis, 102 p., 26 app. p.
Software Engineering
May 2025

This thesis explored the game design of a novel roguelike deck-building game and the implementation of its user interface. The thesis made a hypothesis on what would be a good game design for the game and a good design for its user interface. The research data was collected through a series of playtesting sessions, incorporating surveys and interviews. The collected data was analysed with reflexive thematic analysis, to gain design flaw themes. The design flaw themes and general data collected are used to get an estimate on effectiveness and satisfaction of game design's playability in terms of usability. We propose potential future improvements to the future iterations of the game by reflecting on the design flaws found, general data collected and the game's design we had previously explored.

Keywords: design science, game design, playability, user interface, playtesting, roguelike deck-builder, guessing game, game mechanics, gameplay, iterative design

Contents

List of Figures

List of Tables

1 Introduction

Playability has been regarded as the key concept which reflects the overall quality of the game [1]. Playability is sometimes defined as the the quality of gameplay and a game's usability [2]–[4]. It can be seen as the degree on which the game enables the player to achieve game goals with efficiency, effectiveness, flexibility, security and player satisfaction [5]. Playability can be summed up as the design quality of a game, created from its usability, functionality, and gameplay [6].

Playability is implemented through game's design especially game design. [7]. Designing a novel game can be considered as a wicked problem, which makes getting playability correct a difficult process [8], [9]. Design flaws related to playability may be unseen or unpredictable. More iterative approach must be used to guarantee a level of playability. Game development relies a lot on iterative design, prototyping, and especially playtesting [7], [10], [11].

In this thesis, we look at a novel roguelike deck-builder prototype's game design and the implementation of its user interface (UI). We make hypothesis over what we think are good design choices related to the game's design and user interface. Then we do playtest sessions to discover what sort of design flaws we will find in the design while also trying to get an idea of how satisfying the game is. We design some future adjustments to the game's design and user interface based on the existing design and results of the research in preparation for next iteration cycle. We are trying to answer following research questions:

- ***Is the game design shown in this thesis playable in terms of usability as measured by effectiveness and satisfaction?***
- ***What kind of user interface elements does the game's design require to function?***

In Chapter 2, *Background*, we go through what is the design process and space to for creating a roguelike deck-builder game. We start by defining the term game design and look at the process of successfully applying it to design a game. After that, We will examine more specific part of the game design process called playtesting in its own section. In the playtesting section, we look how experts recommend to do playtesting. Lastly, we will have a section for roguelike deck-builder genre, where we try to define the genre, and examine the game mechanics that are typical for the genre to get a better idea of conventions of the game's genre.

In Chapter 3, *Game*, we try to understand the game and its design. In the first section, we will start with initial design principles and criteria that guided the early design. We show the overview of the game and introduce a lot of game's elements. Using the introduced game's elements, we examine the game in depth in the sections 3.3-3.7.

In Chapter 4, *Empirical study*, we examine the process we used to acquiring participants for a study and its research methods used to collect data from the game. We look at surveys, playtest, and interview used in the study, how were they made and why they were used in this study. For the playtest, we compare, how and why, the playtest version differs from the game and its design.

In Chapter 5, *Results*, we look at the results of the study we conducted. First, we look at general things that we found from our research data collection. Then we do reflexive thematic analysis to get a result of design flaws and player satisfaction.

In Chapter 6, *Discussion*, we answer the research questions and sum up the most relevant results from the study and try to construct a design that reflects what we

learned from the study's results.

In Chapter 7, *Conclusion*, we describe what sort of research we did in the thesis and what sort of research findings we got.

2 Background

In this chapter we go through what is the design process and space to for creating a roguelike deck-builder game. Section 2.1 defines the term game design and look at the process of successfully applying it to design a game. After that, Section 2.2 looks into playtesting and how experts recommend to do playtesting. In Section 2.3, we define the roguelike deck-builder genre and examine the game mechanics that are typical for the genre to get a better idea of conventions of the game's genre. These conventions are relevant when we are introducing our design in Chapter 3.

2.1 Game design

There are multiple ways to define game design. Jesse Schell, the author of the book *The Art of Game design: A Book of Lenses* says that "Game Design is the act of deciding what a game should be." [12]. Robert Zubek, the author of the book *Elements of Game Design* [7] describes game design as a design of rules for how things in the game behave. Ernest Adams in his book *Fundamentals of Game Design* [13] defines game design as a process of the following:

- *Imagining a game*
- *Defining the way it works*
- *Describing the elements that make up the game (conceptual, functional, artistic, and others)*

- *Transmitting information about the game to the team who will build it*
- *Refining and tuning the game during development and testing*

Analyzing an already existing game or a game idea is part of the process of game design. Zubek defines three levels as a basis for analyzing games (see Figure 2.1):

- *Mechanics and system*: "*Mechanics* are the game objects and actions that the player interacts with. They can be assembled into *systems* with specific properties."
- *Gameplay*: "*Gameplay* is the process of players interacting with game mechanics."
- *Player experience*: "*Player experience* is the player's subjective experience of gameplay."

Figure 2.1: Three levels as defined by Zubek

Zubek categories two approaches for design:

- *Top-down design*: The design starts with a desirable player experience and the designer splits the experience into various pieces. The designer figures out what kind of gameplay can generate the desirable player experience from the mechanics
- *Bottom-up design*: The design explores the space, builds mechanics and systems first. Mechanics and systems are tested continually with real players to see what kinds of gameplay and player experiences are being produced.

Zubek recommends a hybrid and iterative approach, where the designer works from both ends. The designer should make a lot of top-down plans and bottom-up experiments, to make both designs converge at some point. Prototyping and playtesting early and often is essential for the hybrid and iterative approach.

2.2 Playtest

Playtest is the process of testing games for bugs and design flaws. According to Jesse Schell [10] every playtest is defined by six key questions:

- *What sort of research questions should the playtest answer to?*
- *Who is going to participate to the playtest?*
- *At what point of development should the playtest happen?*
- *At which location should the playtest happen?*
- *What to look for in the playtests?*
- *How should the playtest supervisor influence the playtester?*

Schell says that most likely playtesters should be in one's target demograph- ics. Schell notes that even the target demographic can be split into distinct categories with their own pros and cons:

- *Game developers*: The people who made the game. These types of playtesters have high availability as they are part of the team, but are very biased towards their design, which can make noticing some design flaws unlikely.
- *Friends and family*: People who are friends and family of the developers. These types of playtesters have high availability and have low barrier to give feedback after a study session. The downside is that these types of playtesters often have tendency to hide negative feedback.

- *Expert Gamers*: People who have played every variety of the type of game the developers are making. These types of playtesters know how to use the terminology and have wide knowledge pool for comparing the game to other games in the genre, but they often desire more complex and difficult experiences than more casual players. When following the feedback they give, it might not create an experience that's well suited for casual players.

- *First time testers ("tissue testers")*: People who have never seen the game or its iterations before. These types of playtesters are good at noticing things with "fresh eyes". Good for testing usability, communication and "initial appeal" of the game, but it's difficult to get a lot of feedback about mid and late portions of the game.

Zubek recommends that playtesting should start in earliest iteration with game developers, on later iterations expanding to friends and family and then during late production first time testers to get the first impression right. [14]

Schell says that playtests can happen at the earliest before there is a game or game idea and at the latest playtests can happen after the game is released.

Schell recommends playtesting should happen "when useful, but at least every week". Valve did the same thing during the development of Portal [15]. Every friday would be playtests for Portal according to the developer Josh Weier [11].

Schell lists the following as potential places for playtest: In a studio (Where the game is being developed), in a playtesting lab, at some public venue, at the playtester's home, and online.

Zubek brings the same arguments in his book as Schell related to this playtest supervisor influence [14]. Schell says that the playtest supervisor should consider following questions related to their influece towards the playtester:

- *Should the playtest supervisor be in the same room as the playtester?*

Schell claims that playtesters being aware they are being watched by the playtest supervisor often affects their behaviour, but it is still enough valuable for playtest supervisor to watch that Schell recommends that the playtest supervisors should in most cases watch the playtest.

- *What should be told upfront to the playtester?*

  Telling things upfront might allow making a playtest without built-in tutorial, but the introduction has a lot of room for human errors.

- *What should the playtester supervisor look at?*

  Schell recommends looking at the playtester more than the game, if possible to see the playtester's emotional responses. In an ideal situation, face, hands and the game would all be recorded.

- *What should be logged?*

  Playtester supervisor may log manually or digitally some special variables related to the playtest.

- *Can the playtester be disrupted midgame for further elaboration?*

  According to Schell, it is often a bad idea to disrupt the playtester midgame, but can be valuable if done in a right situation. Schell recommends doing it only when the playtester does something "truly surprising".

  Schell recommends making the playtester do the "think-aloud protocol", if it seems like it wo not affect the playtesters behaviour that much.

Schell says that one can gain even more information with meaningful follow-up questions with interviews and surveys. Schell recommends to do either an interview or a survey after the playtest.

2.3 Roguelike deck-building

Roguelike deck-building is a genre that combines most elements of the deck-building card games and some elements of roguelike games. One of the first games of the genre was *Dream Quest* [16], which did not receive a lot of recognition among the audience, but was considered innovative at the time by industry experts [17], [18]. *Slay the Spire* [19] popularized the roguelike deck-building genre and inspired a lot of games with similar set of mechanics [20]. The developer of *Balatro* [21] said that if he would have allowed himself to try *Slay the Spire* and similar roguelike deck-builder games, he would have taken a lot of its mechanics. [22]

Roguelike is a sub-genre of the roleplaying games. The genre's name comes from the game *Rogue* (1980), from which a lot of games took inspiration from. In the *International Roguelike Development Conference 2008* a definition for roguelike was created called the "Berlin Interpration [23]". In the definition there are 9 high "roguelikeness factors" and 6 low "roguelikeness factors" and their prevalence determined how much the game is a roguelike.

We can see that roguelike deck-building games have elements of roguelike games as most games from the genre have 4 of these 9 high "roguelikeness factors" of the roguelike genre [24]:

- *Random environment generation*: Roguelike deck-builders lack randomly generated world where the possible encounters happen in the same space as the traversal. Roguelike deck-builders has a separation between traversal between encounters and the encounters. Most roguelike deck-builders still include map of nodes with branching paths as a method of traversal between encounters and events (see Figure 2.2). However, a lot of roguelike deck-builder games, for example, *Balatro* [21], *Luck be a Landlord* [25] and *StarVaders* [26] do not use a map of nodes with branching paths for traversal and instead do something more simpler. Some roguelike deck-builders for example like *Roguebook* [27] or

Tainted Grail: Conquest [28] do something more different and let the player have free movement between encounters.

Placement of items (cards, relics) and monsters (encounters) are random to an extent in roguelike deck-builders.

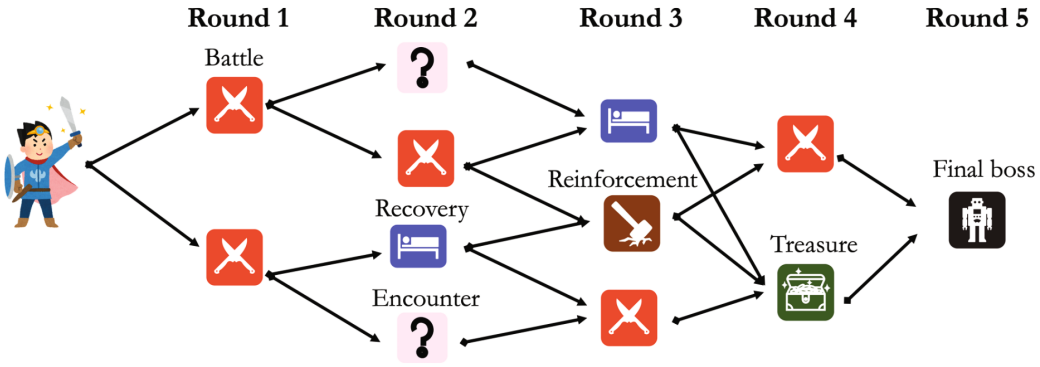

Figure 2.2: Typical roguelike deck-builder's map of nodes with branching paths. [20]

- *Permadeath*: In roguelike deck-builders the player start mostly from the beginning when they lose. Most of the roguelike deck-builders include unlockable cards and difficulty levels, which will not be resetting on loss.

- *Turn-based*: Roguelike deck-builders are often turn-based. In a card playing part of the genre, the player often has a limited amount of cards they can play in a turn. *Ratropolis* [29] is popular example of a roguelike deck-builder that is real-time instead of turn-based.

- *Complexity*: Often roguelike deck-builders are enough complex that several solution can be used to a problem.

Roguelike deck-building games often starts with a pre-defined deck. In roguelike deck-builders, cards are often objects that the player has in their deck and playing them cause some kind of effect. The player may add or remove cards often as a reward or by obtaining from in-game shop. A lot of games in the genre also offer a way to update these cards. Roguelike deck-building games often also have "relics" or equivalent, that are items, which are always in-play. Roguelike deck-builders often include a mix of effects that can be sometimes synergistic. These effects are created by playing cards or are triggered by *relics*. The player does not often have a guarantee that certain cards are offered each playthrough. The combination of increasing level of challenge during a playthrough, synergistic effects, and limited choices pushes the player to adapt and play differently each playthrough. [30]–[33]

Replacing cards with other objects wo not change the gameplay of roguelike deck-builders, which means that exactly using "cards" as objects for deck-building is not necessary for the definition of roguelike deck-builders. Games like *Luck be a Landlord*, *Astrea: Six-Sided Oracles* [34] and *Backpack Hero* [35] are not using cards as object for the deck-building. [32]

The card playing part of the genre equals often combat encounter, but does not

necessarily have to. Games like *Balatro*, *Luck be a Landlord* do not have combat encounters, instead the player has to beat a score by playing cards or in *Luck be a Landlord*'s case symbols.

The genre's combat encounters often are heavily inspired by *Slay the Spire*'s combat encounters (see Figure 2.3). Often the player starts the turn with 3 energy and draws at the start of each turn 5 cards unless the player has received some kind of effect that allows them to have more energy or draw more cards. The player may play cards as long as they can pay the energy cost allocated to that card. Cards can have various effects such as dealing damage, increasing block, causing a debuff, causing a buff, drawing cards, manipulating deck or others. Enemy intention's is displayed on top of the enemy, showing what action the enemy will do at the end of the turn to help the player prepare for that action. If the enemy intent is to attack, a lot of games even display the exact amount of damage the enemies will do at the end of a turn. [36] According to Chris Zukowski genre is one of the most

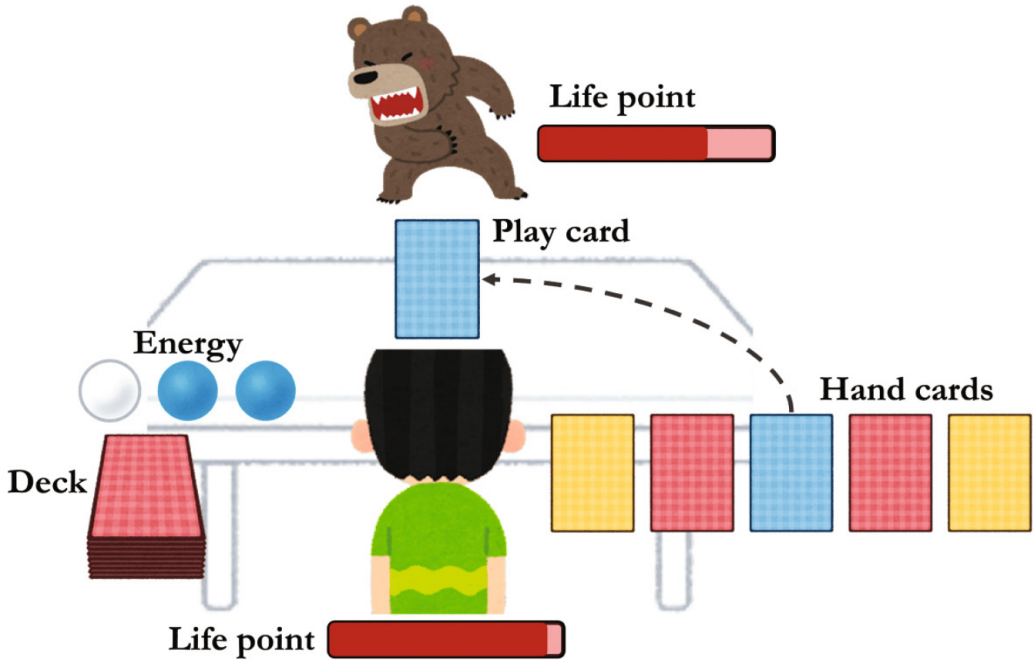

Figure 2.3: The battle event of a typical roguelike deck-building games. [20]

important factors for success of a game. Zukowski analyzed *Video Game Insights'* data and concluded that games with roguelike deck-building Steam tag were one of the most best genres for indie developers to make and sell for profit in Steam in 2022, 2023, and 2024. AAA games and games that did not gain 10+ reviews were for example removed from the data as they did not represent the indie scene according to Zukowski. In the data, games with roguelike deck-building tag that were released in 2023, about 7.78% (7 out of 90 games) roguelike reached 1000+ reviews and in 2024 about 6.71% (11 out of 164 games). To get better idea of what does this data means, let us compare this data to a genre that is one of the least profitable, the puzzle genre. Games with puzzle tag have respectively 0.45% (13 out of 2,913) and 0.36% (14 out of 3,902). [37], [38] According to *Video Game Insights* for Steam games with 1000+ reviews had also median of 30 copies sold per review. [39]

3 Game

The focus of the design in this chapter is on the combat part of the game. Design between combat sequences that do not have significance within three combat encounters are not yet designed. The current design is used to confirm that the first three combat encounters work and explores how they work, before design is created beyond that.

3.1 Design criteria

The game's design explores, in a single-player context, a guessing mechanic combined with a roguelike deck-builder gameplay with a goal that the designed game would be both accessible to a casual player and a more hardcore player. The original idea for the exploration of the design space came from the idea of "mixing" game elements together that did not seem to fit together as a challenge to learn more about game design and to guarantee level of novelty by creating some kind of hybrid genre to an extent.

Originally, the "mixing" process did not include roguelike deck-builder gameplay. After dissecting what the other elements meant in terms of mechanics, gameplay, and experiences, roguelike deck-builder was later added as it was seen fit the best with other elements.

Novelty was driving force to create the hybrid genre as novelty is valuable for game's marketability as it is a way to stand out in the market [40]. However novelty

might have been initially overvalued with the process of creating a hybrid genre. Too much novelty has been known to led to problems with marketability [41]. It is ideal that the game have some novelty, but should also include familiarity, as that way the customers understand better what is being marketed to them and the developer has an already existing target audience to market the game to [42]. Roguelike deck-building genre happened to be inline with goal of marketability

In this thesis, the guessing mechanic refers to an action where the player makes a guess against a pool of potentially correct answers. When the player makes a guess, they get a result back on how correct the guess was. Getting the result narrows down the potential correct answers as long as the guess creates some new information.

Battleship is one of the most notable guessing games with the guessing mechanic. In Battleship there are two players. Both players have a grid that the other side does not see and the game starts with players inserting their ships into their grid. Players alternate turns where they make a guess in form of "shot" to a coordinate. The opposing players reports the correctness of the guess to the player whether that shot hit their ship or not. This way players narrow down the potentially correct answers until they have found all of the ships and game ends. The player whose ships' coordinates are all shot, loses the game.

Guesses can target a single potential correct answer or in some games and versions they can target a group of guesses. There is a known variant for Battleship called salvo variant, where instead of guessing one coordinate at a time, the player guesses a group of coordinates, often five. When calling a group of coordinates the other player reports the correctness as a number of how many shots were hits. [43], [44] This form of guessing is also something that the game artifact has.

Developing a game for low skill level players and high skill level players is an idea that have been in a lot of successful games. The co-founder of Blizzard Entertainment, Allen Adham [45]–[47] claims that their secret at a time for making games

successful games was to appeal to a casual, midcore player, but also to a hardcore player. Adham uses the game of chess as an example of this kind of a game.

Avoiding challenging the low skill level player too much, while not removing challenge from the high skill level player is ideal for the design [48]. Ease of use and usability in general becomes more important the more lower skill level players are accommodated. that is why the design tries to find opportunities to use the Jacob Nielsen 10 usability heuristics a lot. [49] The Nielsen's 10 usability heuristic are the following:

1. Visibility of system status
2. Match between the system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

For the design, heuristics 6 and 7 are used a lot even though it creates problematic situations for the guessing mechanic when the player guesses a group of potentially correct answers. How do we display a guesses with groups in efficient and readable manner? The Section 3.5 tries to answer this question.

Battleship game was used as the base for the guessing mechanic. This was done to get a help from heuristic 2. Some players may use their experience of Battleship to understand how the guessing mechanic works. This makes the starting experience less overwhelming for players with existing experience and makes the learning of basics more pleasant. Secondary effect of using Battleship for base, is that it helps with marketability as it is familiar to a lot of potential players. The video game versions of the Battleship games uses fog of war technique to create opportunities for guessing. Fog of war is defined by Ernest Adams as "a technique of hiding regions or some aspects of the terrain." [50].

Gambler's fallacy is the belief that successive occurrences of one type of random event will make an additional occurrences of that event appear less likely [51], [52]. Making probabilities follow gambler's fallacy makes the game less frustration as it follows the expected belief of gambler's fallacy, which in turn also allows player makes the game more skill based as the player has a guarantee to affect the game in certain intervals [20], [53]. Video game called *Dota 2* is an example of using pseudo-random generation, where the probability of a positive outcome increases after every negative outcome [53], [54]. Guessing mechanic achieves the same effect without relying on pseudo-random generation through narrowing down potentially correct answers. Guessing games can also use visually more intuitive language compared to using percentages.

3.1.1 Reoccuring design explanations

When we look the design in depth in the sections 3.3-3.7 there are reoccuring design explanations that are reused multiple times to explain a design choice. In this section we will examine those design explanations.

We use the term ***hidden depth*** as something that the player believes to have mastered, but as the player experience grows they notices that there is more to

master. The challenge is hidden in plain sight. Ideally this would support the optimum flow experience more by hiding the challenge from a low skill player that has already enough challenge in the game and will give a challenge to a high skill player that is looking for a challenge [48].

For example, *Slay the Spire* has a lot of enemy encounters with predefined attack patterns. The player might at first attribute the attacks to being chosen by random, but as the player experience grows, they can start to notice that there are predefined attack patterns. These kind of realisations allows the player have more complex gameplay without any underlying mechanic changing.

We use the term ***maximising the value of a turn*** for a situation where the player will figure out the most optimal solution for a turn, sometimes to an extent where it can lower the player satisfaction.

Lead designer of *Civilization IV*, Soren Johnson, has this to say about *maximising the value of a turn* [55]:

> Many players cannot help approaching a game as an optimization puzzle. What gives the most reward for the least risk? What strategy provides the highest chance – or even a guaranteed chance – of success? Given the opportunity, players will optimize the fun out of a game.
>
> ...
>
> A phrase we used on the *Civilization* development team to describe this phenomenon is that "water finds a crack" – meaning that any hole a player can possibly find in the game's design will be inevitably abused over and over. The greatest danger is that once a player discovers such an exploit, she will never be able to play the game again without using it – the knowledge cannot be ignored or forgotten, even if the player wishes otherwise.
>
> *Civilization 3* provides a simple example with "lumberjacking" – the

practice of farming forests for infinite production. Chopping down a forest gives 10 hammers to the nearest city. However, forests can also be replanted once the appropriate tech is discovered. This set of rules encourages players to have a worker planting a forest and chopping it down on every tile within their empire in order to create an endless supply of hammers. However, the process itself is tedious and mind-numbing, killing the fun for players who wanted to play optimally.

The game's design is made to accommodate these types of players [56] by mostly minimizing actions that the player could prove to be the most optimal set of actions with an excess time investment. In this chapter giving the player more choices is considered good or bad depending on the context and might seem random what choice is bad or good to the reader of this thesis, but there are some considerations behind these claims. If a choice is considered good or bad in design roughly follows following criteria:

1. *Choosing between actions should not take a long time*: This can happen when the choice is easily optimisable, but still takes considerable effort. When making the player calculate or go through a lot of options one by one, the problem more likely does not fit this criterion.

2. *The longer the choosing process takes, the more meaningful the outcome should be*: The player should not have to spend time on a choice that will give him/her non-existent benefit.

This criteria is very roughly followed. For example in this design and in a lot of turn-based tactics games the player has a lot of options where to move. The player is faced with a lot of situations with non-existent benefits between the choices given to the player. This choice is still included in the design even though it breaks the second criterion.

3.2 Overview of the game artifact

The game is a turn-based roguelike deck-builder which takes inspiration from the game Battleship as a way to introduce guessing mechanics (see Figure 3.1). Both sides have their own 3×3 play area grid and inside the grid there are 3 ships owned by the side they are on. The opponent's play area is initially hidden by a fog of war for the player. Both play areas may also have attack coordinates, which are visual indicators that the opposing side will attack the highlighted coordinate at the end of a turn in a shooting phase.

The player has a deck pile and discard pile. When deck pile is empty, discard pile is shuffled and added to the deck pile. Instead of a hand, the player has 3×3 card grid, where cards are drawn to random positions.

Cards are objects in a game that trigger an action or actions assigned to them upon being played. Cards often have one type of actions. Card actions can be split into three categories depending on the action's target:

- *Defensive*: Defensive effects are targetted to player's play area. Defensive effects are often cards that help protect the player's ships.
- *Offensive*: Offensive cards are often cards that will set an attack coordinate to the enemy's play area.
- *Global*: When played global card's effects will happen regardless of the state of the play areas.

At the start of the turn five cards are drawn to the card grid and player's action points is set to three. Also attack coordinates are set to the player's play area, one for each of the opponent's unsunken ship.

The player can do four different actions during a turn:

- *Choose a coordinate*: This is the main way the player interacts with the game. Choosing a coordinate consumes one action point, meaning that the player can

potentially do the action of choosing a coordinate three times during a turn. Selecting a coordinate plays a card from a card grid with that coordinate and cards adjacent to it.

- *Rotate the card grid*: Rotating a grid is only possible if the player has not played any cards yet on that turn. This will rotate the cards around the middle card slot 90 degree counter clock- or clockwise depending on the direction chosen.

- *Move a ship*: Both sides are rewarded with movement points (called wind power) for each hit on the opposing ships. Each side can consume 10 movement points to move. This action returns fog of war to all or some coordinates depending on the possible range of the movement.

- *End the turn*: Ending a turn will discard the rest of the cards and move the game to a shooting phase. In a shooting phase the player's ships will shoot at the opponent's play area and if the opponent has not yet lost, the opponent ships will shoot at the player's play area. The attack coordinates will determine the coordinates for the ships to shoot in the shooting phase.

For the player, the goal of the game is to destroy all of the enemy's ships while avoiding getting any of the player's ships destroyed. Shooting a ship will cause a damage. Damage reduces ship's hit points unless the ship has been protected enough to mitigate the damage. If a ship's hit points reaches zero, the ship will be destroyed.

Both sides have a fog of war. Fog of war is removed (*revealed*) for a coordinate in two cases:

1. if a ship is destroyed then a fog of war is removed for that ship's coordinate.

2. If there is a full certainty that a coordinate has a ship or does not have ship, then the fog of war is removed for that coordinate.

The game follows Battleship's salvo variant, where a side gets result on how many shots were hit on the other side during the last turn. The result will be used to determine a full certainty situation mentioned in the second case. If full certainty is not reached for a coordinate even after shooting it, then a little hit note is added. Hit note system works as a simplified combat log, which only stores information that is relevant for locating the opponent's ships inside the fog of war. We will refer coordinates with hit notes as partially solved.

If all of the opponents' ships are destroyed, the player wins. After a win, the player has an opportunity to add a new card permanently to the deck. Player is moved next to a map interface, where player have to select a new destination node. Destination nodes range from a new battle, to a node where a player can remove one card permanently from the deck. Going to a new battle resets everything except the player's cards and the locations of player's ships. If player loses even one of its ships during a battle, the player loses and the game ends.

Figure 3.1: Overview of the game

3.3 Game in depth

The game we are looking is in prototype phase and it still uses placeholder graphics and thus animations are only used when they are considered a necessity for the user experience.

The design in depth was split into sections 3.4-3.7, which goes through combat, fog of war and scouting, movement, and card grid and playing cards. However not every designed element fit the categories. The designed elements that did not fit the categories are introduced in Section 3.3.1 and Section 3.3.2.

3.3.1 Infobox

Infobox is a box in the right of the view as seen in Figure 3.1. The player may hover over an element and it explains what is the element in the infobox. Each element has an info entry that can be displayed by hovering over the element.

Infobox was designed to help finding out what each element is for cases where the player would forget a function or a detail of an element.

Infobox was chosen over a tooltip to avoid blocking the view of other nearby elements. The gameplay includes finding patterns through experimenting by hovering on elements. Examining element or finding patterns both include the same action of hovering over elements. However getting tooltips on top of elements can disrupt finding patterns.

3.3.2 Momentum element

"Momentum element" refers here to gameplay and also experiences created by a strong snowball mechanics. Snowball mechanic is a mechanic that makes the player, who is winning, win more. Snowball mechanic is often paired with a catch-up mechanic. Catch-up mechanic is a mechanic that gives the player who is losing a

chance to catch-up the opponents. [57]

Next we will explore momentum element more through things, which can be learned from from the sections 3.4-3.7.

Guessing mechanic was a natural place to start to explore momentum element's game design. Guessing mechanic causes randomness, which works as a catch-up mechanic [57], but also at a start of a battle a way to determine which side is going to be ahead. Guessing mechanic's scouting has a state, where the possible combinations list determines the state. This state can move gradually step by step and sometimes, if lucky, to the finished state.

Momentum element is an attempt and experiment to unify mechanics into the unified player experiences through snowball mechanics. Guessing mechanic is naturally very related to snowball mechanics as it has natural catch-up mechanic, a way to choose who is more likely to snowball and a state with steps to keep the snowballing effect on one side.

Movement and movement points is a way to continue the guessing mechanic's scouting state after it reaches the finished state. Movement points are gained gradually step by step without upper limit, which allows a side's advantage to grow without a limit. The side that scouts faster gains more movement points, which can be used for movement that slows the opposing side's gaining of movement points and dealing damage. When the opposing side deals less damage, less blocking is needed and more attacking can be done which enhances the gaining of movement points.

With momentum element, the player can be faced with multiple novel problems when deck-building. The player can be met with novel choices as to create a deck that will work better than on average when doing catch-up or snowballing. For example, the player might create a deck that is good at the start of the battle with scouting to avoid getting a head start with snowballing. Another example is

a deck that works when catching-up might work for different reasons like it works by scouting a lot, it works regardless of scouting or it works by giving the player defensive options to handle the opponent's snowballing. These example problems and potentially many more, are not in a lot of roguelike deck-builder games as snowball mechanics are not that strong in a lot of roguelike deck-builder games.

Persistent health should be in a design at some point to enable the prevalence of snowballing mechanics more. Snowballing can further amplify the effects of chance. Section 3.4.5 explores the ideas why persistent health points would be good for the design and snowballing makes these reasons more reasonable.

The design related to momentum element has a lot of elements that can be used to measure the winning chance. For a side the health of every ship, the health of every ship after the turn, movement points, and the state of scouting can be measured. Giving weights to these attributes can give a single value as an estimation of how well a side is doing. Comparing that value to the opposing side value gives a simple estimation of the player's winning chance.

Player experiences created by momentum element requires further exploring and experimenting as the effects of it are very unclear in the design. The main assumption is that losing feels bad and winning feels good. As both side's momentum can be measured, the experience can be enhanced with various elements. When the design is not represented anymore with placeholder graphics, it can be an opportunity to try things like dynamic weather and dynamic music to enhance the player experiences.

3.4 Combat in depth

3.4.1 Play area

The player's side and the opponent's side each have their own play area grid (see Figure 3.2). Both sides have their own 3 × 3 play area grid and inside the grid there are 3 ships owned by the side they are on. We will call the cells in the grid, coordinates. If a coordinate has a fog of war on it, we will call it a fogged coordinate. Fogged coordinates can be scouted by shooting to reveal whether there is a ship or not. Coordinates may also have attack coordinate on them to show that the opposite site is going to shoot at the coordinate during a shooting phase.

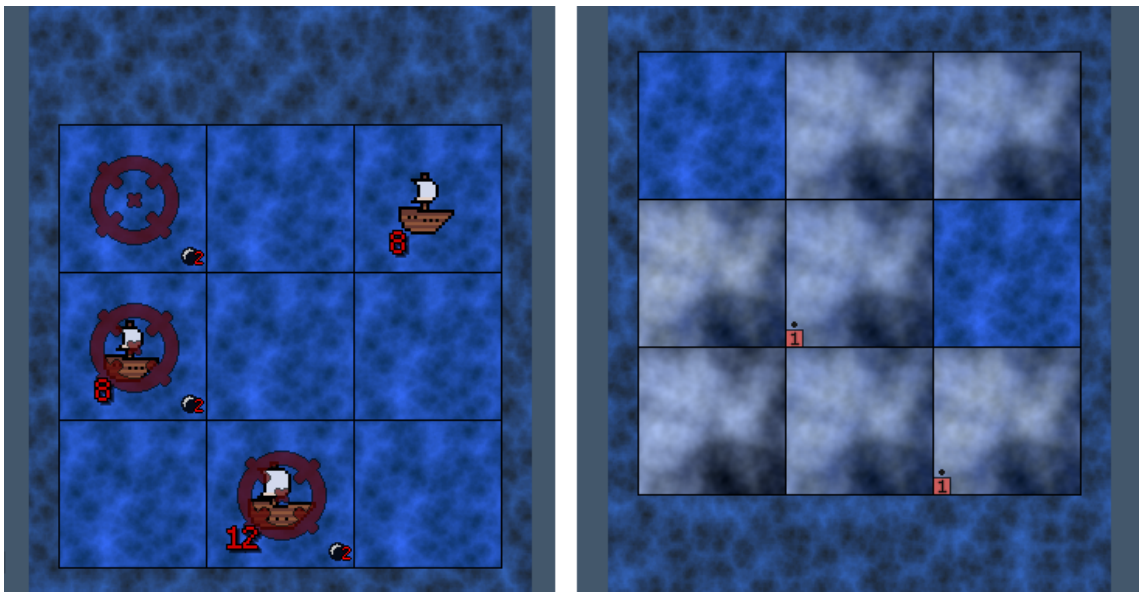

Figure 3.2: Two play areas. The right play area is opponent's as there is visual

3.4.2 Attack coordinate

Attack coordinates are visual markers in the play area, where shots will be fired in the shooting phase (see Figure 3.3). Attack coordinates resemble enemy intent that is familiar from roguelike deck-builders, but instead of displayed on top of attacker, it is instead displayed on the target. Attack coordinates use a red reticle as an

icon and have a 4-frame animation that resembles spinning. In the bottom right corner of an attack coordinate there are icons for shots. These icons are there to show, what kind of shots are already prepared to be shot to the attack coordinate during the shooting phase. All attack coordinates are owned by a ship. Attack coordinate generation is different for the player and the opponent. Immediately after the shooting phase, the opponent's side will generate the attack coordinates, one for each of the opponent's unsunken ship. All of the player's attack coordinates are created from card's played.

There has not yet been designed a consistent rule which ship will get to own which attack coordinate. There is currently no visual indicator for which ship owns which attack coordinates due to the limiting placeholder graphics. Less special unique attack coordinates are used because of the limitations. Currently the design lacks a consistent rule for which player's ship will get assigned the attack coordinate on play. To avoid frustration the player with inconsistency, the player's ship should not have unique buffs or debuffs that affect its shots. Therefore limitations of attack coordinates assignment creates limitations for ship and card designs.

Figure 3.3: Attack coordinate on a fogged coordinate with two shots assigned to it

3.4.3 Shots

Shots are attacks, that are spawned in the shooting phase to attack the attack coordinates. Any ship in the shot's attack coordinate will take damage and this event also grants the side that shot a movement point. Scouting is done using shots.

A shot uses an animation where the shot normally comes from the attack coordinate holder's ship and hits the attack coordinate. There is a time interval between each shot start. The latter shots' animations are faster, which in turn makes the time interval for each shot animation's end shorter. When the opponent's ship that shoots is hidden inside the fog of war the shots spawn from the edge of the opponent's play area instead directly from the ships.

Shots are only instances where the prototype plays an audio. Starting a shot animation starts playing an audio clip that sounds like a cannon firing. Ending a shot animation will play an audio clip that is either sound of a ship being hit by a cannon ball or sound of a cannon ball hitting the sea. If a shot hits a coordinate that is revealed or will be revealed, then the sound played is correct. However if a shot coordinate stays fogged, then the sounds played are added to the same pool and the shot animation end will take a random audio from the pool and plays it.

The exceptions where the shot appears from the corner of the opponent's play area is designed to prevent giving information about ship's locations. If the ability to track the ship's location from the shot's animation would be part of the design, then it could replace to some extent the need for a scouting system as the player can figure out ship's locations through tracking the animations instead of using the scouting system.

The shot animation not always visually coming from a ship has the danger of making the player still assume that it comes from a ship and getting confused. The design has not yet found a satisfying solution for explicitly telling how it works while not breaking fourth wall, so the design is relying on an questionable solution that the player learns it through trial and error. This problem extent easily to the spectators as well as the problem arises from displaying false visual cues without explicit explanation.

The shot's hit audio not always matching the action might create same kind of

confusion as the shot animation not always visually coming from a ship. When a shot hits the fogged coordinate hit audio is added to the same pool and random audio is taken for the fogged coordinate's hit audio, so that the location of the opponent's ship cannot be revealed from the hit audio.

The shot animation has a time interval to avoid playing all shots' audio clips at the same time, making them unrecognizable due to audio overlap. The time interval between shots end is short to make the player not recognize that the audio clips are played in wrong order.

3.4.4 Ship

Ships are the units of the game. Ships can move, shoot, take damage, sink and restrict movement. Every ship has three kind of point attributes assigned to them: health points, block points, and extra damage points (see Figure 3.4). These are displayed in the corners of the ships except block points and extra damage points are not displayed if the value is 0. There are multiple types of ships with their own visuals and properties. Attack coordinates are held by the ship.

3.4.5 Health points

Health points determines the amount of damage a ship can take before it sinks. Health points reset back to full after each battle. Starting health points are determined by the type of a ship. When a ship is shot, it triggers an animation the ship loses health points (see Figure 3.5).

Health points are used to determine the winner and loser of the battle and therefore it is used as a base motivator for all of the actions in the game.

Having own health points for each ship has the following design motivations:

- It is a feature that is shared among many games. The player is expected to be familiar with it so a shorter lesson can be done related to health points.

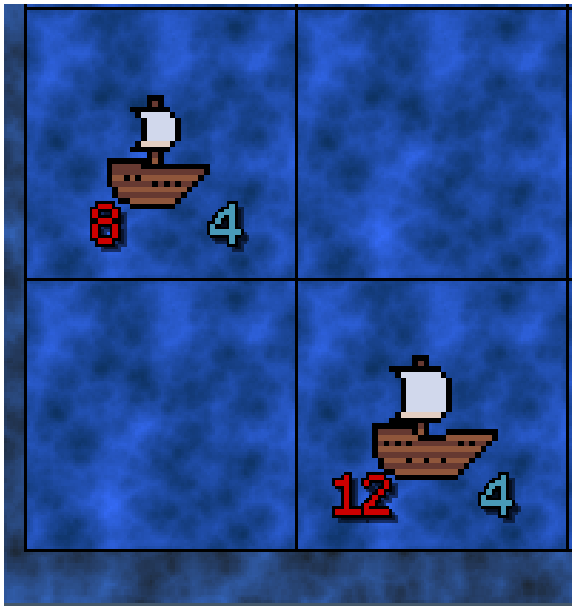

Figure 3.4: Two ship types with health points on the left and block points on the right

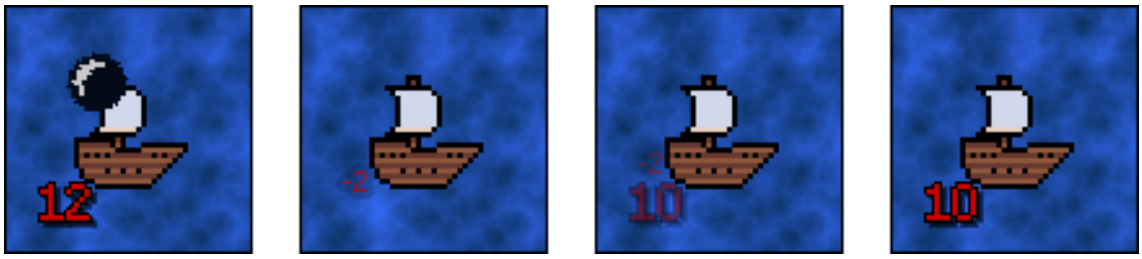

Figure 3.5: Ship's health point losing animation

- Gives player an easy to understand reason to prefer attacking or defending of one ship over an another ship in a lot of situations where there would normally not be one.

- Altenative design such as using shared health pool for a side. Will either sometimes fail to respond to a hit and not move even if a hit has happened or risk of revealing the ship location in the fog of war when that is not the intention. For example, shooting 1 and 2 damage shots in the fog of war and the opponent losing 1 health reveals to the player that the 1 damage shot hit and the 2 damage shot did not hit.

Using persistent health points and not resetting health points after each battle is worthy design detail to consider. When using the persistent variant as an alternative, we are expecting that either the starting health points value is higher or the opponent's shots deal less damage. This is only fair, as the difference between persistent variant and the current health resetting version can be interpreted as that the same pool of health points is expected to last multiple battles instead of one. The persistent variant would have been in the current game artifact, but was not yet implemented due to time constraints. It can have the following advantages to the design:

- *Health points matters more to the player*: Making the opponent lose health and avoiding losing health are the main objectives in the battle and most game mechanics are built around these two. With non-persistent health, when the battle seems to the player like an easy or certain win, it can lead to indifference towards losing player's health points as the loss condition does not anymore motivate the player. From objectives stand-point the player does not have reason to interact with mechanics related to avoiding losing health leading to a shallower than the intended game experience.

  If health points reset after each battle, the player's perceived value of health points often scales with the perceived difficulty of the battle as the perceived likelihood of reaching zero health points increases with increase of perceived difficulty. With persistent health points, the perceived value of health points can be equal in all difficulty levels. This is because losing health points in an easier battle can be equally harmful as losing health points in a more difficult battle when the consequence of losing health points persist between battles.

- *Makes the game more skill-based.*: Law of large numbers states that the average of the results obtained from a large number of independent random samples converges to the true value, if it exists. In this case, sample is interpreted

as a battle's health loss. Without persistent health points having one battle with very bad luck can result in a game over regardless of a skill. With persistent health points in a series of battles the impact of chance starts to converge towards a constant as it follows law of large numbers more and more as the sample (battle's health loss) size increases. The more constant impact of chance increases the likelihood of player's skill making a difference to the process of winning multiple battles.

- *Makes node map better*: With persistent health points it is easier for the players to estimate how strong they are in-game. This helps the player to be less overwhelmed by the node map traversal choices as they can estimate more easily how much risk they can face. If the player has more health, they know that they can take more risks, which allows the player to do more meaningful choices related to the node map traversal.

  Persistent health points enables designing a node that heals. Node that heals makes the node map traversal more approachable than it currently is. It is understandable concept that nodes that deplete health should be balanced with nodes that replenish health.

3.4.6 Block points

Block points is the amount of attack damage a ship can take before the damage affects their health points. After an attack, block points are reduced by the damage value unless block points amount reaches 0, in that case the remaining damage value is reduced from the health points. Block points resets to 0 at the start of the turn. Block points can be gained from playing cards which have a keyword *block* written in them.

Block points provides a clear motivation for the player as not having enough block points will cause player to lose health points, which is a resource that the

player needs to play the game. Block points are there to enable design of cards that provide *block*.

Block points also can be given to the opponent's ships. This can provide a challenge by without teaching the player any new complex rules about the game.

Block points points are hidden, when the value is 0 as there is less information for the player to parse through.

3.4.7 Extra damage

Extra damage points are added to each shot made by the ship. Extra damage points resets to 0 at the start of the turn.

Section 3.4.2 last paragraph explains that in the current design ships should not have buffs assigned to them. This is why in the current design every source of extra damage is always given to each ship on one side.

Extra damage points are there to enable design of cards that provide extra damage.

3.4.8 Sinking a ship and victory/loss conditions

Ship sinks when its health points reaches 0. The player and the opponent have different consequences when a ship sinks. If the player's ship sinks, it will be a game over for him/her. Following consequences happen when an opponent's ship sinks:

1. The sunken ship will not be removed from the play area, instead there is a visual indicator that it has sunken and the ship is immobilised. (see Figure 3.6)

2. Shooting the sunken ship does not grant movement points or any other benefit for the player.

3. The ship will not hold any attack coordinates anymore. All the attack coordinates that would be given to the ship are removed.

4. If the sunken ship is in fog of war, the fog of war will be revealed for that coordinate. If the process of initial revealing leads to full certainty of an other coordinate it will be revealed as well.

5. If it is a last ship on the opponent's side that has not sunken, then the player wins the battle.

Sinking an opponent's ship is designed as a highlight moment for the player. One of the reasons for implementation of the consequence 4 was to enable the player to witness the highlight moment of the game and not be missing the moment due to a ship sinking behind the fog of war. Also the player might have got an expectation from the Battleship board game that if a ship sinks it is told to the player. Having to relearn that rule would work towards more overwhelming tutorial section.

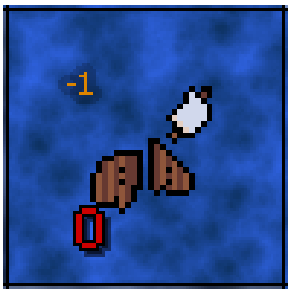

Figure 3.6: Sunken Ship with -1 extra damage

In Section 3.4.2, we learned that attack coordinates are owned by the ships and opponent ship's can have their own signature attack coordinates. The opponent's sinking effect 3 allows the player to make a plan for sinking a specific ship, which will disable the remainder of the battle that ship's attack coordinates. The player may use that information to make a meaningful decision, which ship they will like to sink. This also can have some properties for *hidden depth*, a term we defined in Section 3.1.1.

Which ship is optimal to sink is affected by the following variables:

- *The enemy ships the player is facing*: Some ships are more dangerous than others. Some ships are more durable than others.
- *The cards that player has in their deck*: Certain cards can work better or worse against an unique ship.
- *The current state of the fog of war*: Revealed ships are often faster to sink than ships still behind fog of war.

Each variable in itself can be in multitude of states. The problem of "Which ship is optimal to sink" can be an interesting problem for the player to solve as it should require the player to have a deep understanding of all three variables. Solving the problem gives the player a lot to master and provides a lot of replayability.

The design fails to incentivise the player to scout over maximising dealing damage, even though there is a system where both are option. The problem of "Which ship is optimal to sink" creates a situation where the player might favor scouting over maximising dealing damage. When the player has more meaningful ways to interact with a system, then the system provides more value.

There is likely to be *hidden depth* in this design. Hypothetically the player might make an assumption that only one of the variables is the true key to winning. This is beginner friendly and more simpler way of thinking than thinking all three variables should be considered when making the decision on "Which ship is optimal to sink". When the player is not challenged greatly they do not have to rethink this notion. However, when the player is more experienced and the player has proceed to a higher playthrough difficulty level, the player might get enough difficulties with some combat encounters and might question the notion for "Which ship is optimal to sink".

The big problem with sinking's design is that the enemy's presence becomes

greatly lesser after opponent's ship sinks, but the player's amount of potential actions stays equal lessening the challenge a lot. There is less risk of taking damage from the opponent's ships, when one of the opponent's ships stops generating attack coordinates and attacking due to sinking. When the opponent has sunk, the player loses an attack target, leaving the player with excess action points, which the player may only use to play more defensively and further reducing the threat of the opponent's side. Balancing the game for both, before first sinking and after first sinking is therefore problematic as they are vastly different in levels of challenge in this iteration. The reduction in challenge could be thought as positive as it is a way to provide the player a sense of relief. It's overall good design to mix period of tension with a period of relief. However too long periods of relief, can feel boredom to the player, which is unwanted quality in the design. Card design can be used to adjust this to an extent by making some cards work in a way that they are more powerful at some parts of the battle. For example, a correctly balanced card that gets stronger every time it is played can help speed up the last part of the battle.

3.4.9 Shooting phase

Shooting phase is a phase after ending a turn where both sides' ships shoot at the other sides' attack coordinates. Sides do not shoot simultaneously, instead the player's ships shoot first and after that the opponent's ships shoot. Shots in shooting phase will scout and on hit will deal damage to the opposite side and grant movement points to the shooter's side. After the player has finished shooting, a salvo recapitulation is displayed for few seconds before disappearing and letting the opponent's side shoot. Salvo recapitulation can be seen on Figure 3.7. There are also animations for movement points gaining, which are further explored in Section 3.6.3.

Figure 3.7: Salvo recapitulation appears for a moment. The player has shot once middle left and once bottom left coordinates. Salvo recapitulation tells that there were two hits, so both of the shots were hits.

The player's side and the opponent's side will not shoot at the same for following reasons:

1. *Easier to track one salvo at a time*

2. *The player has more time to reflect on the effects of the player's shots*: When the opponent has enough movement points, the opponent will move immediately after the shooting phase. When the opponent moves it clears a lot of information on what happened in the opponent's play area. This can make the player miss information, which was intended the player gets.

3. *The player gets an opportunity to react to the opponent's ships' attack coordinates with shots*: For example, the player might be able to sink an opponent's ship, which removes one attack coordinate before the opponent's ships shoot. This gives the player an alternative way to avoid damage, which they can figure out how to combine with their other actions that turn. This gives the player an opportunity to show his skill. Being a rare event, it can also provide a positive element of a surprise when and if the player realises that they may

avoid damage after they have mistaken that taking damage is inevitable.

Salvo recapitulation is used as a way to justify a break to the player, where they may reflect on the effects of their shots. It is designed for the same reason as previously listed reason 2. Salvo recapitulation also helps the player to reflect more thoroughly on the effects of the player's shots as it displays what were the attack coordinates used that turn.

Salvo recapitulation gives a lot of information, but does not give any new information. It just recapitulates the information that was displayed before the shooting phase in a different format for few seconds until disappearing. Its goal is to help the player to process information, but has a chance to do the opposite and confuse by adding an extra user interface element for the player to process.

3.5 Fog of war and scouting in depth

3.5.1 Fog of war and revealing

When a coordinate is occupied by the fog of war, the information, whether there is a ship occupying a coordinate or not, is hidden from the other side.

There are multiple things applying and removing coordinate's fog of war. Each battle starts with fog of war applied to every coordinate. When the opponent is moving it applies fog of war to all movable positions. Fog of war will be removed when a coordinate is revealed. Revealing refers to an event where it's displayed to the player whether a fogged coordinate has ship or not. There are following triggers for revealing:

- If a ship sinks then a fog of war is removed for that ship's coordinate.
- If there is a full certainty whether a coordinate has a ship or not then the fog of war is removed for that coordinate.

Fog of war is implemented to enable a guessing mechanic for the game. Fog of war hides the opponent's ships' positions and the player has to make guesses with shots to narrow down the opponent's ships' positions. As guessing narrows down the opponent's ships' positions, the player have less and less to guess. The opponent's movement applying fog of war to all movable coordinates is designed to reintroduce the guessing mechanic to the battle for creating more variety inside the battle and to enhance the momentum element.

3.5.2 Partially solved

Partially solved refers to a coordinate's fog of war status, where the coordinate has been shot, but there is not full certainty whether the coordinate has a ship or not. Partially solved coordinates are cleared from the opponent's side when the opponent's ships move. Partially solved coordinates have always a hit note on them.

Having partially solved coordinates instead of always revealing the coordinate upon hit is designed to create gameplay variety to the guessing mechanic. Partially solved coordinates have often different probability for containing a ship compared to other fogged coordinates (see Figure 3.8).

Partially solved coordinates makes scouting slower compared to always revealing the coordinates upon hit. Partially solved coordinates are great way to achieve slower scouting if it is desirable. As an example for comparison, the card grid size and play area's size increase is the alternative less desirable solution to making scouting slower.

Partially solved coordinates makes it more difficult to judge for the player what is the worst possible outcome for hits during a salvo. The two worst outcomes are doing no hits and doing one hit, when the goal is hitting. With partially solved coordinates both of these outcomes have an unique thing positive to them. With no hits the player gets to reveal all shot coordinates and on the other hand if the player

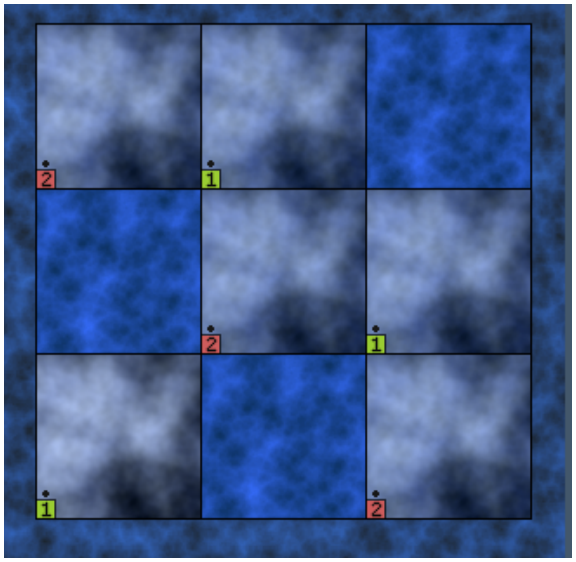

Figure 3.8: In the current state, each sea coordinate has 0/3 chance of having a ship, each coordinate with red hit note has 2/3 chance of having a ship and each coordinate with a green hit note has 1/3 chance of having a ship.

hits once, they know that they did at least some damage to the opponent's ships. Both have positive outcomes that are difficult to measure against each other and to prove which outcome would have been better. Ideally this sort of ambiguousness, where the player is not always sure which outcome is the unluckiest outcome, makes the player less frustrated when it happens as the player has less proof for their unluckiness.

Partially solved coordinates can often allow situations, where the player can easily guarantee that some partially solved coordinates will be solved and get to full certainty after the player's next shooting phase. In these situations, the player may interact with guessing while chances are not affecting the scouting making the interaction with guessing purely related to the player's skill.

When partially solved coordinates mechanic exist, having different shot counts on a salvo can affect where the partially solved coordinates are generated. This allows the player to further strategises with scouting when the player can adjust shot counts to scout better and avoid only partially solving a coordinate (see Figure 3.9).

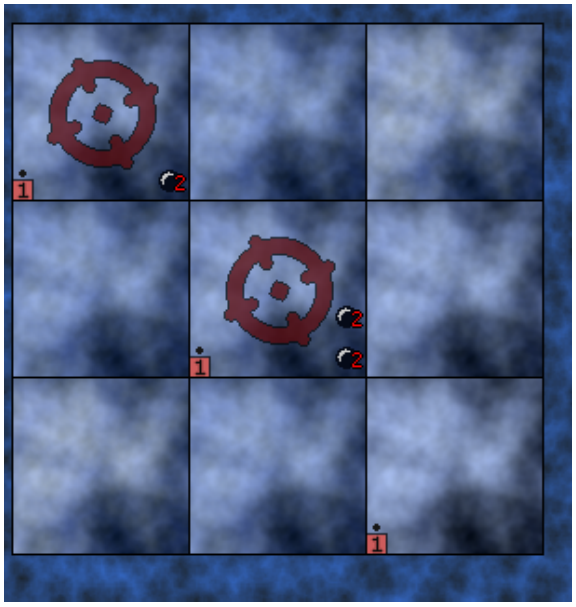

Figure 3.9: The player has a guarantee to reveal all three coordinates with hit notes after shooting like the following picture demonstrates

Partially solved coordinate design has potential design flaw as it can be confusing for the player. The player does not get instant visual feedback how much health is reduced when a ship is hit if it is not revealed instantly. Movement point gaining animation is done less clearly for the player to avoid revealing what is under the partially solved coordinate. The player is also interacting with hit notes, which probably has to be taught to player in order for the player to understand them. As there are visual design problems with partially solved coordinates' design it might harm the marketability of the game as the customer's first impression might be detered by the visual design problems of partially solved coordinates.

There is potential to design shots that could benefit from doing a secondary effect on miss or on hit. However, partially solved coordinates limits the design of these kind of shots. Some secondary effects may reveal partially solved coordinates, when that would not be ideal for the design. For example a simple shot design that cannot be done without revealing the coordinate is a shot that would grant double the movement points if it hits. If the player uses one of these shots, the player will

know if the shot hit or not depending if they gained more movement points than the total amount of hits were.

It is valuable to consider using alternative more simpler scouting design to create visually coherent experience at the start of the game and potentially change to using partially solved coordinates at some point, for example at a higher difficulty. However, making two versions of the same mechanic both work perfectly with existing systems is unlikely, so using only the simpler alternative for scouting design could also be an option.

3.5.3 Hit notes

Hit notes are notes in a partially solved coordinate, which are implemented to help the player to keep track of the information they have about the fogged coordinate. Hit notes are removed, when the coordinate is not partially solved anymore. A hit note has three visual elements to store information for the player:

- *Color*: Each shooting phase has its own color for the hit notes (see Figure 3.10). This information can tell, which hit notes are part of which salvo.
- *Array of circles*: The amount of circles in a hit note represents the amount of shots, that were targeted to hit note's coordinate during a salvo
- *Number*: The number inside a hit note represents total number of hits by the salvo without counting salvo's hits from revealed coordinates. The number is updated, whenever reveal happens that would change the value.

Hit note system is designed as a simplified combat log, which goal is to log and keep information only relevant to the current state of scouting. Minimizing information to only relevant information is designed for the following reason:

- *Reduce the player's cognitive load*

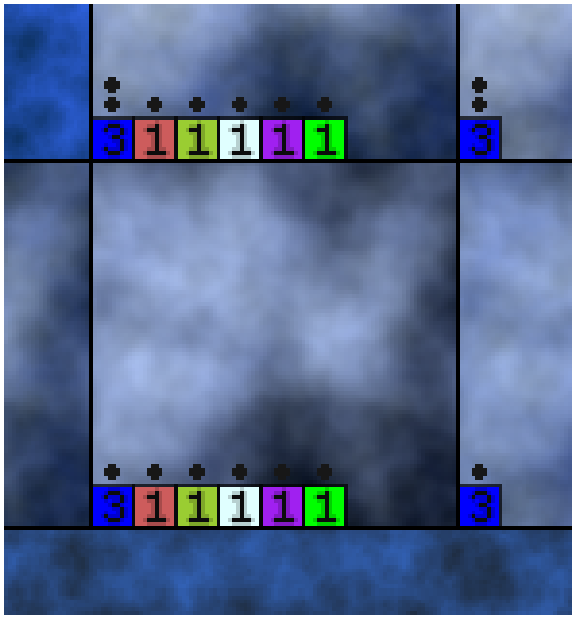

Figure 3.10: Showing all the possible hit note colors

- *Reduce the chance of overflowing the hit notes row*: The hit note system has a flaw and the hit notes row can overflow. If the player plays unoptimally (see Figure 3.10), they have the possibility to create a lot of hit notes to one coordinate.

3.5.4 Scouting

Scouting refers to an active progression towards finding the ships that are in fogged coordinates. Scouting is a consequence of shooting. Scouting system holds for each play area a list of all of the possible combinations. Possible combination is data of three coordinates. These three coordinates represents location of the ships. In total there are 9 coordinates and the 3 ships can represented as 84 different possible combinations in the grid:

$$\binom{\textit{grid's cell count}}{\textit{ship count}} = \binom{9}{3} = \frac{9!}{3!(9-6)!} = 84$$

When a salvo has landed, all the ship location combinations are checked in the possible combination list and all elements are removed that are proven false using the shots' landing coordinates and hit counts. If every combination in possible combination list has a same value for a specific fogged coordinate then that means that it is fully certain that the coordinate has a ship or not and the coordinate is revealed. The way the possible combination list works can sometimes cause coordinates that was not shot to be revealed as the possible combination list narrows down.

When the opponent moves, possible movable combinations are added immediately to the possible combination list that tracks the opponent's side. When the player moves possible movable combinations are not immediately added to the possible combination list that tracks the player's side, instead they are added when the opponent's possible combination list's size reaches 0. This detail is further elaborated in the next section.

In general the more longer the player has to scout before finding the opponent's ships the less damage they can deal to the opponent's ships. Chance has a significant impact on how long the player has to scout before finding the opponent's ships. Having a larger health pool, persistent health and consistent way to add block points is important in these scenarios to mitigate losing because of being unlucky. Changing the variables related to scouting will change this chance's prevalence and also change the need for larger health pool, persistent health and consistent way to add block.

The two downsides to scouting with possible combination list are:

- *The player might be confused why some coordinates are revealed.*: To lessen the confusion, there could be an animation that reveals the coordinates in patches at a time. A hit note *numbers* could change, which trigger other hit notes *numbers* to be changed in patches. When all the hit notes *numbers* are 0, then the fog of war is removed.

- *The player might prefer spending time and effort on tracking manually the coordinates.*

3.5.5 The opponent's attack coordinate generation

The opponent's side will generate three attack coordinates after the shooting phase. To determine which coordinates in the player's play area are generated for the attack coordinates, one element from the possible combination list is chosen randomly and the attack coordinates are generated where the ships are according to the randomly chosen combination element.

The opponent possible combination list only grows, when the list's size reaches 0. The list grows by adding all 84 combinations before pruning the list. The list is pruned starting by matching it to the most recent salvo's details and going to an older one. If the size of the list reaches 0, the algorithm goes back to a previous state and then starts to consider possible movable combinations. When the possible movable combinations reaches three deep meaning that the movement is considered to happen three times or recent salvo's details list is exhausted the algorithm stops.

The design goals for the opponent's attack coordinate generation logic are following:

- *Support the momentum element*: The opponent hits more shots as the possible combination list shrinks

- *Controllability*: The main reason for the design of opponent's attack coordinate generation logic is to give the player a sense that they are tricking the opponent, which should extend to a point, where the player feels that they have a control of the enemy. The opponent not having information when the player moves is designed to give the player an opportunity to take an advantage of the opponent's lack of information. Predictability of the opponent is

important as the player can predict how the opponent will react to the lack of information from a movement. The player can make a plan how they will use the opponent's logic against them and this can give the player a feeling of control over the opponent's action. This design may be extended by having a mechanic that gives the player an opportunity to give false information to the opponent. For example manipulating the hit information of the opponent's salvo, can narrow down combinations in a way that the opponent will spend multiple turns making wrong guesses based on the false information it received.

Taking advantage of the opponent's attack coordinate generation logic has not been emphasised in the design due to its complexity. An example for a simple to execute trick that the player can do to take an advantage of the opponent logic is to move a ship twice. In this example the player gets a guarantee that the ship that was moved two steps away from all the previous ships' positions is not shot before the combination that are one step away are ruled out in a matter of multiple turns. However, most of the ways to take advantage of the opponent's attack coordinate generation logic are not this simple and require the player to pay attention to what sort of possible combination list the opponent has and what are on the salvo's details list to make a prediction on how do the opponent "repair" its knowledge in a way that the player gets the most advantage with a single movement. To take advantage requires likely too much effort and time for the player to have an enjoyable experience with the mechanic. The player might have an urge to do everything to maximise their value in turns, which can lead to a situation where the player spends a lot of time trying to take advantage of the opponent's attack coordinate generation logic, but is not enjoying the experience of doing so. The design relies on a assumption that the player who can handle the complexity of the mechanic

will use it and those who cannot, will end up avoiding interacting with the mechanic. However, this is unlikely the case and most likely to get a closure for the positive impact of the design choice, the game most likely needs testers that are very experienced with the game.

- *Hidden depth*: The game is designed not to teach explicitly the opponent's logic to the player. This can mean that the less experienced players are not aware that they can even interact with the mechanic. This should make the early player experience less overwhelming.

  The players that have mastered the game to a point where it is not overwhelming to them are more likely discovering new things about the game such as the opponent's attack coordinate generation logic. This gives the experienced player more to master.

3.6 Movement in depth

3.6.1 Movement notes

Movement notes are notes that are added for each of the opponent's revealed ship that will move to show to the player where the opponent's ship may have moved and to keep the information logged for the player. Movement notes have two arrows showing the two possible directions, where the ship may have moved and an icon of a ship (see Figure 3.11). The movement note is added to a coordinate where the revealed ship was before the movement. The movement note is added visually to the same array as hit notes. The opponent's movement removes all hit notes and movement notes from the play area.

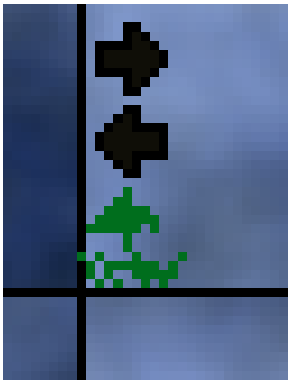

Figure 3.11: Movement note

3.6.2 Movement

Movement is a game mechanic where a movement threshold's amount of movement points are consumed and one side's play area's ship or all the ships move. Movement threshold represents the minimum amount of movement points required to enable movement. By default movement threshold is ten. The player and the opponent do the movement differently.

The player can select a ship for movement when they have movement points equal or over the movement threshold. When selecting a ship, adjacent unoccupied coordinates are highlighted and the player can select one of these coordinates to move the ship to the selected coordinate (see Figure 3.12).

The opponent moves all of its ships at the start of the turn whenever it has movement points equal or over the movement threshold. The opponent's revealed ships move differently than ships still in fogged coordinates. The opponent's ships has following movement order when moving, starting from the ship/ships that move first:

1. *All revealed ships in corners of the play area*
2. *Other revealed ships*
3. *Ships in fogged coordinates*

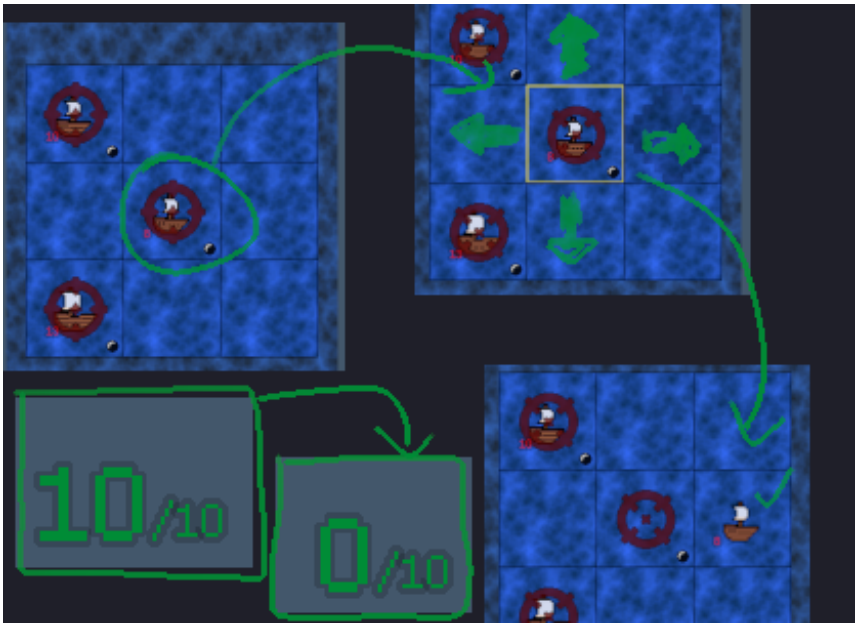

Figure 3.12: Image from the tutorial: Teaching the player's movement

Revealed ships move by choosing two random adjacent coordinates. These two random coordinates' directions from the ship's original position is added to a movement note as two arrows. The movement note will be placed to a ship's original position before the ship moves. From the two random coordinates one is chosen by random as long as it is possible. This chosen coordinate will become the new position for the ship. There are exceptions when the coordinate is not chosen by random from two random coordinates:

- *When an another ship has already been chosen to move to that coordinate*: In this situation, the other coordinate is chosen.

- *If other ships have chosen their random coordinates and the ship generates two random coordinates that would create a situation, where each ship's random coordinates would be shared with other ship*: In this situation the ship changes one of its random coordinate.

- *When the chosen random coordinate is occupied by sunken ship*: In this situation, the other coordinate is chosen. If an another ship has already been

chosen to move to the other coordinate or if there it is also occupied by a sunken ship, then the ship does not move.

Ships in fogged coordinates chooses randomly one unoccupied coordinate in the grid and will move there.

When the opponent moves the play area's fog of war is returned by adding all possible movable combinations to the possible combination list that tracks the opponent's play area. All the possible hit notes and movement notes are removed from the opponent's play area when the opponent's movement happens.

Ships in fogged coordinates randomly choosing one unoccupied coordinate and move there has multiple reason for the design. One of the design reasons is that there are uniform probabilities where it has moved unlike with a lot of other implementation. This enables that the player or the game's UI does not have to keep track of the probabilities apart from how the revealed ships are affecting the probability. If the player keeps track of probabilities, then this behaviour might go to the *maximising the value of a turn* behaviour, which was explored in Section 3.1.1. This design reason is revisited in the *moving a whole fleet makes hit notes visually more manageable* part again in different format.

The player moving with different logic than the opponent is designed for the following reasons:

- *The player gets to make a decision, which ship to protect over an another*
- *Moving one ship has a simpler control scheme than moving three ships*: Simpler movement is faster to execute for the player, the player does not have to consider the order of moving, and the player is less likely to do meaningless choice.
- *Moving a whole fleet makes hit notes visually more manageable*: Moving one ship in a fogged coordinates can make hit notes outdated. Then there would

be two design choices for hit notes unless outdated hit notes would be kept:

- *Update the hit notes*: Updating hit notes starts with keeping some of the hit notes. Keeping the hit notes has the disadvantage of increasing the cognitive load more than not keeping them. There is also real risk of overflowing coordinate's the list of hit notes when they are kept. If the goal is to keep the information up-to-date, there must be some kind of additional note/notes to display possible movement from partially solved coordinates. Keeping the hit notes also mean that the normal hierarchy between movement notes and hit notes is not going to be kept. Because normally hit notes erase all notes for hit note list, movement note will be always displayed the leftmost in the hit note list as it is generated immediately after the movement. Breaking this convention makes the note array less readable.
- *Clear the hit notes*: Clearing the hit notes has the advantage of reducing the cognitive load more than not clearing them. However, tracking outdated hit notes would have given the player an advantage. Therefore clearing the hit notes might upset some players as they might choose to rely on more slower or effort taking methods to keep track the hit notes when the game is not going to do that for them.

Forcing all of the opponent's ships in fogged coordinates, to move at the same time to any positions has the advantages of clearing the hit notes, but does not suffer the disadvantage of clearing relevant information as the movement is uniformly random for ships in partially solved coordinates.

The movement difference between opponent's revealed ships and ships that are in a fogged coordinate is designed to reward the player for following reasons:

- *To give incentive to scout over maximising damage done*: When the opponent

is close to reaching its movement point threshold and there is at least one ship in a fogged coordinate, the player is incentivised to prioritise the scouting over maximising damage done. For example revealing only 2 ships versus 3 ships through scouting can reduce the size of the possible combination list after an opponent's movement happens from 28 to 3 in the best case scenario. If the revealed opponent's ships would use the same movement as ships in a fogged coordinate and the opponent is one turn away from reaching its movement point threshold, the player might feel indifference to scouting. This is because a chance to deal damage once and a chance to get movement points once is a small reward. Whereas reducing the amount of possible combinations after the opponent's movement should be considered a large reward. Reducing possible combinations of the opponent will give the player very probable chance to scout them in a turn or two and giving the player a guaranteed hits.

The player has been given an agency to focus on scouting instead of maximising damage done. It can be good for the design to give the player a reason to use that agency. To sink a specific ship in a fogged coordinate in less turns and limiting enemy movement by revealing a ship are only designed incentives to motivate the player to scout over maximising damage done.

- *To avoid frustrating situations*: When the player has an expectation that they will meet a self-made goal of sinking a ship in a nearby future and that expectation is not met, it is more frustrating the longer it takes to finally finish that goal.

- *To keep the momentum*: The opponent's movement is designed to support the momentum element. The player can get advantage in gradual steps with scouting as revealing each of the opponent's ships increases their likelihood of scouting them after the opponent's movement as well.

The design of the movement might suffer from its complexity and lack of feedback. It might be difficult for the player to understand the opponent's movement when the movement works with different logic compared to the player's movement. The opponent movement has different logic for revealed ships and ships in fogged coordinates, which makes the understanding of the opponent's movement even more difficult. If the player does not learn the opponent movement from a tutorial, they might get an assumption that the ships that are in fogged coordinates move similarly to the revealed ships. This is because the player has not been given a reason to assume otherwise and there is a lack of feedback showing that the assumption is wrong. The player with a wrong assumption must meet the following steps to expose that their assumption is wrong:

1. *Be aware and remember where the opponent's fogged ships could be potentially located.*

2. *Find the opponent's ship after its movement in a coordinate which is not adjacent to the locations where the fogged ships could have been.*

The player may also do other tasks that make perception more difficult between step 1 and step 2, such as spending couple of turns scouting previously revealed ships. The probability of the step 2 happening even if step 1 happens is low. Even when the two steps happen, it does not mean that the player will understand how the movement logic actually works. The player might only realize, that the fogged ships move differently than revealed ships, but not the real movement logic until further experimentation.

The opponent's revealed ships explicitly showing the player using movement notes to demonstrate that it can only move to two possible directions instead of every adjacent directions is designed to reduce the lack of feedback and making the scouting easier and balance the momentum element. When the opponent's revealed

ship moves to one of only two possible directions it is more effortless and faster to scout and reveal those ships compared to if it would be moving to one of every adjacent direction. This might lower the chance of forgetting where the fogged ships were when it takes on average less turns to scout them and therefore help the player, if he does not know that there is a movement difference, to notice the movement difference between revealed ships and ships in fogged coordinates. Moving only to two directions instead of all adjacent directions also lowers game's and especially scouting's difficulty. The player might have difficulties to understand scouting, so the movement of the revealed ship is designed to be one of the simplest scouting problems and might help the player to understand scouting by giving the player simplified first steps.

3.6.3 Movement points

Movement points, which are called in-game wind power, are points that are required and consumed for movement. To the right of the current movement points locates the displayed the movement threshold (see Figure 3.13). There is no maximum amount for obtainable movement points. Both sides movement points labels are visible to the player and they are displayed in the left corners of the game view in their respective sides.

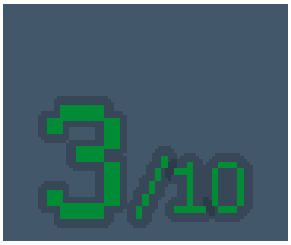

Figure 3.13: Movement points

There are animations for gaining movement points. Gaining a movement point start with spawning an particle image for movement points. The movement point's

location tweens towards the gaining side's movement point label. When the tween is done, the movement point is added to the displayed value. The movement points spawn in different location depending on which side is gaining them. When the player gains movement points, their points are coming from the salvo recapitulation's "hits" text label as shown in the Figure 3.14. When the opponent gains movement points, its points are coming from each hit player's ship.

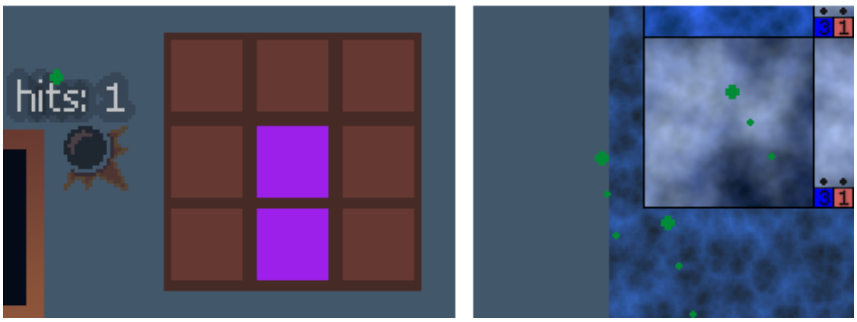

Figure 3.14: Left: Movement points spawning from the salvo recapitulation, Right: Movement points moving towards the opponent's movement points label

Normal design in grid-based roguelike deck-builders is to have cards for movement. As the starting deck size is designed to be small, the player would end up drawing movement cards too often for the opponent's attack coordinate generation logic to work properly. The movement cards are difficult to balance as they often are strong meaning that other cards would also have to be strong and this affects the deck-building in a negative way. Movement points was designed as more of a balanced alternative than the potential movement cards as the frequency of the movement can be somewhat paced and the player has less impact when trying to affect the frequency of the movement. It also does not affect deck-building as negatively as movement cards would.

Movement point gaining is tied to the salvo's hits for the following reasons:

- *To enhance the intended momentum element*: The side with more momentum gains more likely more movement points and further keeps the momentum to that side.

- *Give the player an ability to strategise around gaining movement points*: The player can increase their movement point gaining ability with deck-building by taking cards that help them to maximise their shot count and cards that make scouting faster. If the player wants to move frequently, they may do so by taking previously mentioned types of cards. This makes deck-building more nuanced.

  The movement points system also lets the player strategise during a battle. Unique ships makes movement considerable more valuable in one kind of encounter over other encounters. The player is met in every encounter with a choice to prioritise movement points over some other actions.

Not having maximum amount for obtainable movement points was designed for following reasons:

- *To enable strategies where the player may move twice in a row*: This is a strategy where people who understand the basics of the opponent's attack coordinate generation logic, can take an advantage of the logic and possible feel more skillful.

- *To incentivise saving movement points for later use*: The player is not losing the ability to gain movement points after the movement points amount reaches the movement threshold. Losing the ability to gain movement points because of a limit to obtainable points, would incentivise the player to use the move action so they get the ability to gain points.

  As there is no limit for the amount of movement points the player can have, they are less pressured to move and this supports a design better where they can make future plans for the usage of movement points. The more normal is the saving of movement points, then the opponent's attack coordinates' shots design can be more pronouncedly split into two following categories:

- *Shots that are meant to be blocked.*
- *Shots that are meant to be dodged by moving away.*

The less there is a pressure for the player to use the movement points as soon as they reach the movement threshold, the more acceptable does this kind of design where the player should sometimes hold the movement points might seem to the player.

The more earlier the player will decide to move the more likely they are accidentally involving themselves with the opponent's attack coordinate generation logic. This might make the player faces the opponent's attack coordinate generation earlier than designed.

The player's movement point gain animation does not spawn from hit ships like the opponent's movement point gain animation to avoid showing, which partially solved coordinate has a ship or not to the player. Teaching the player two very different animations for a same action is not ideal, but it is one solution to not reveal partially solved coordinates, when it is not designed to be intended.

The player does not interact with the movement every turn, so the player might forget to move. The reason to have movement points gain animation included in a prototype, is to remind the player how to gain movement points and the existence of the movement points.

3.7 Card grid and playing cards in depth

3.7.1 Adding a card to a deck

When adding a card to a deck also known as deck-building, the player may choose one of three cards to add to their deck. The player gets the option to add a card to a deck once after each battle as a reward (see Figure 3.15). The player may skip

adding a card after seeing the three cards, if they so please. All three cards choices are often more powerful than the cards the player starts the game with.

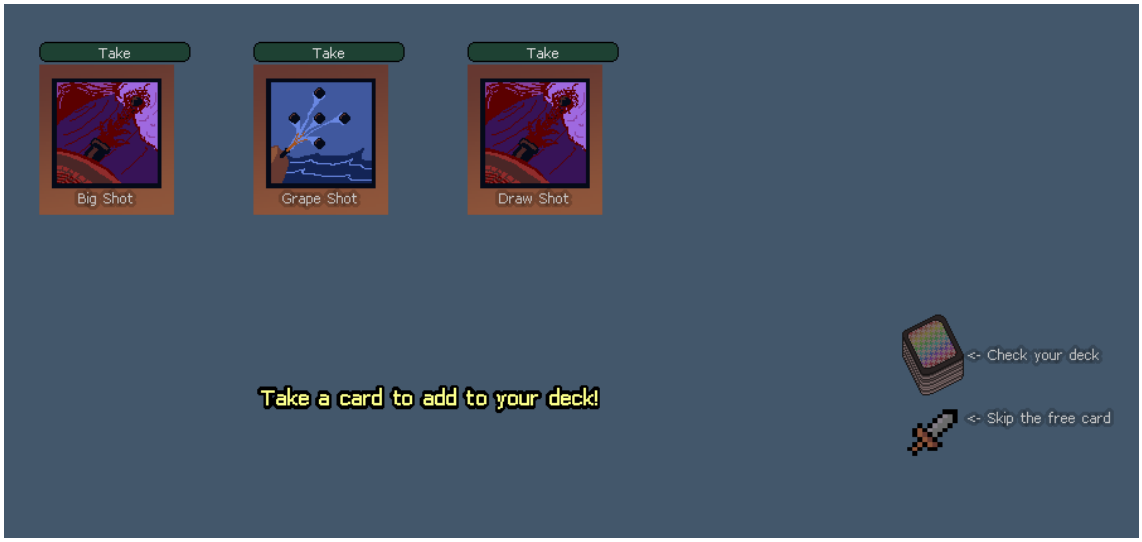

Figure 3.15: The player may add one of the three cards to their deck or skip taking it

Because of fog of war, cards having location in card grid, movement points, and ship having locations, the game artifact has more states than the average roguelike deck-builder. Deck-building can allow specialisation to interacting with a specific variable in an unique way. For example the player could build a deck that somehow specialises in attacking corners of the play area. The player might focus on the corners and think corners differently for that playthrough, but with an another deck the player can put their focus on a different variable. This idea can be used to create even more variance between playthroughs [58].

Giving the player the freedom to build their deck can be creative endeavor for the player and should be encouraged by the design. However, giving the player too much freedom with the deck-building can have negative consequences such as:

1. Some of the players are more likely to use same strategies that they have found success with before and ignoring building the deck towards strategies that they have perceived to lead to lesser success. Ignoring strategies can lead

to shallower experience when the player is refusing to interact with certain content or even mechanics. If the game does not regularly narrow down the freedom and force the player to try other strategies, then the effect of favoring a strategy over another one is further amplified. The amplification happens because the player success rate gap further increases as they start to master the strategies they play and stay more stagnant skill-wise with strategies they are not playing.

2. Too much freedom may overwhelm the player.

Letting the player build the deck as the run progresses has multiple potential design advantages over building the deck at the start of the run. If the player would build the deck freely from a large pool of cards at the start of the run, then they would be incentivised to consider a possibly overwhelming amount of cards, their relations to each other, and after adding a large amount of cards they might have difficulties remembering all the added cards they even have when it matters to the gameplay. Letting the player build the deck as the run progresses reduces these problems as the deck-building happens gradually. Graduality for deck-building has the following advantages:

- It is easier to remember what cards are in the deck during the battle when the cards are introduced gradually There is an exception to this advantage. If the player has picked in multiple runs the same cards in the non-gradual variant, then the player has an advantage for remembering the cards better as they may use previous runs for references to what cards they have in their deck.

- The player is not overwhelmed from picking multiple cards at the same time.

Possibly worthy consideration is a design of letting the player choose a card from one of three cards multiple times at the start of the run instead of spreading the options of adding the cards as the run progresses like it is currently designed. The

alternative design does suffer from the same problem as the earlier alternative design, where it might be difficult to remember the cards, when they are all introduced in the same battle. Choosing cards as the run progresses has multiple advantages over the alternative design of choosing them at the start:

- *The player is less likely to restart a run*: The player may end up having a bad luck with card choices given or otherwise trouble building a good deck. The player may estimate in these kind of situations to have a low chance of winning. The player is less likely to give up and restart the run, if the player has invested more to that deck, which is time in the current design. Not disincentivising run restarting increases the prevalence of the negative consequence 1 as the player is more likely to restart with deck that they have estimated to have a lower chance to win with.

- *Easy way to grow player's power during the run*: Adding cards that benefit the player makes the player's power grow as the run progresses. Growing the player's power allows growing the opponent's difficulty more freely and allowing more different design between earlier and later battles to create more different experiences between those battles.

- *Player may experiences more variety of decks*: When the player gradually build their deck as the run progresses, the deck changes over time. The player gets to experience and play with a slightly different deck between each time card is added. However, a big counter-argument is that the player will start with the same deck every run which makes the start of the run more similar to the start of the other runs.

- *Player can adapt*: If the player is close to losing, the player is inclined to add cards that help more on a short-term instead of a long-term and vice versa. This reaction makes the player play with more diverse decks and experience

more varied gameplay.

The ability to skip a card reward gives the player a lot of freedom to design their deck, but does in a way increase the prevalence of the negative consequence 1. However, not being able to skip cards will cause major balance problems. Having too many cards in a deck will make the design of multiple practical strategies more difficult. The player may get card rewards that do not fit the strategy they are going for increasing the amount of unrelated cards to certain strategy, making the execution of that strategy less and less probable to execute. As the execution of a certain strategy gets less and less probable, the player's power starts to decrease as the game goes on.

3.7.2 Card piles

Card piles are a collection of cards. There are three types of piles:

- *Deck pile*
- *Discard pile*
- *Exhaust pile*

There is also card grid, which can be considered a collection of cards. Card grid is similar to the concept of a hand in other card games except in the card grid, the cards get assigned a grid position attribute when drawn.

At the start of the turn five random cards are drawn from the deck pile to the card grid. If the deck pile is empty when trying to draw cards, then the discard pile's cards are moved to the deck pile so the player can draw the five cards. If both piles are empty, then the drawing ends.

All played cards are moved to the discard pile, unless the card is exhaustible, which moves it to the exhaust pile. At the end of a turn all cards from the card grid are moved to the discard pile.

When a battle ends all cards from discard pile, exhaust pile and card grid are moved to the deck pile. Adding a card, as a post-battle reward, adds it to the deck pile.

There is a separate menu where the contents of the card piles can be viewed, which shows all the cards assigned to the piles.

Drawing limited amount of cards instead of drawing all of the cards is designed for the following reasons:

1. *Drawing limited amount works as procedurally generated puzzle*: There are a lot of different potential combinations of set of five cards that the player can draw each turn. The design aim is that the player faces a problem of optimising the best way to use the drawn cards on that turn, creating a small puzzle for the player. Turns are aimed to be designed as short, so the player can experience a lot of these "mini" puzzles in short intervals.

2. *Less cognitively demanding*: There is less cognitive load when there are less cards that the player is faced to consider. The player cannot interact with the cards in his deck, so the player is less likely to consider those cards when making decisions on what to play. There are, however, some situations where player may benefit on how they interact with the drawn cards based on what they have on their deck. The design matches with idea of *hidden depth*.

As there are a lot of different potential combinations of set of five cards that the player can draw each turn, the player has more variation on what they can play. There is a lot of randomness for which cards are being drawn, but it is not totally random what the player draws. The player will not draw the same card twice until the whole deck has been drawn.

3.7.3 Card grid

Card grid is a 3×3 grid, where cards can be drawn to. When a card is drawn, the card is placed on a random cell without a card in the grid. Cards can be played indirectly from the card grid with the *choose a coordinate* action. The cards placement in the grid will affect, where the cards can be played. The player can rotate card grid, if the player has not used *choose a coordinate* action yet on that turn. Rotating the grid will rotate the cards around the middle card slot 90 degrees counter clock- or clockwise depending on the selected direction (see Figure 3.16).

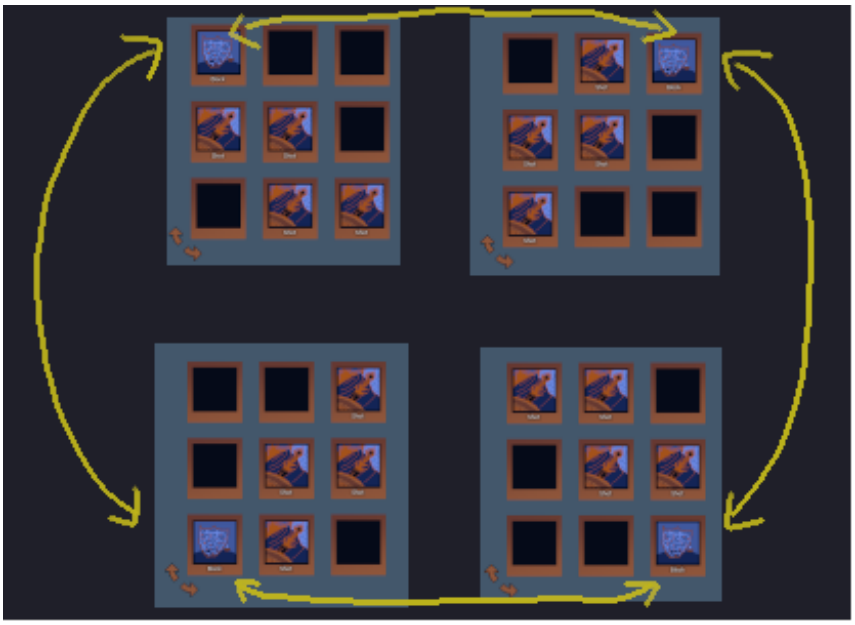

Figure 3.16: Demonstration of the card grid rotation

Card grid is designed to restrict the player so they would target multiple coordinates in a same turn. The game is designed in a way, where doing damage is often more valuable than scouting and it gets increasingly more valuable to damage already damaged target over scouting as the already damaged target is closer to sinking than other not yet scouted targets. Card grid limits by preventing some cards from being played to certain coordinates. When the player has played all possible shot cards to one target, the player may still have playable shot cards, which

the player might use for scouting as they might not have better use for the cards.

Card grid and card grid rotation further makes the drawn set of five cards more of a procedurally generated puzzle. Card grid adds a restraint on how the cards can be played by making their location on the grid matter. The card grid's restraints in itself sometimes prevents the player in a way where the player cannot play any of their shot cards on a ship, leaving the player have "no action" turns.

Card grid rotation allows a way to overcome the restraints of card grid by allowing the player to rotate the cards. Rotating the cards may make some cards go into a more favourable position, but at the same time it might move an another card to less favourable position as a side effect.

Card grid allows potentially superior controller support compared to the alternative of a hand. Hands are 1-dimensional, where the cards are in a single row. Due to card grid being 2-dimensional it has better navigation for stick compared to the hand. When the card grid has not too many cells and is symmetrical and static sized, it is not too difficult to map the card grid's cells to controller's analog stick's horizontal and vertical values.

3.7.4 Action points

Action points (see Figure 3.17) are resource that are required and consumed for the use of *choose a coordinate* action. One action point is required and consumed every time the action is used. Action points are set back to three at the start of the turn.

3.7.5 Choosing a coordinate

Choose a coordinate action is the only way to play cards and allows playing up to 5 cards at a time. The *choose a coordinate* action consumes one action point and plays a card from a card grid with that coordinate, if there is one, and cards adjacent to the coordinate. The action requires one action point and this action must play at

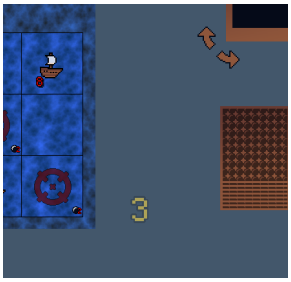

Figure 3.17: Attack points are located between the discard pile and the player's play area

least one card for it to happen. The *choose a coordinate* action can done by choosing a coordinate from the opponent's play area.

The design of the *choose a coordinate* action can be understood better by comparing it to the alternative of choosing a card and choosing a target. The alternative is very common in a lot of roguelike deck-builders. Here are potential advantages that the *choose a coordinate* action provides over the alternative design of choosing a card and choosing a target in the context of current design:

1. *Creates a puzzle with a restriction*: The *choose a coordinate* action creates a restriction by forcing the player to play the card in a coordinate and all adjacent cards to the coordinate. If the player does want to target a card to a coordinate, but wants to target card adjacent to that coordinate, to a different coordinate, then the player might not be able to do both or the player might need to be careful with *choose a coordinate* actions in a right order (see Figure 3.18). Having both, cards with offensive actions, that targets the opponent's play area, and cards with defensive actions, that targets the player's play area, creates these situations more often compared to if there would be only one kind of actions available in the card grid. Having both kind of cards also creates situation where the player's ships' location has an impact related to the opponent's ships' location. However, the interaction is currently underutilized in the design, but could work as a way to generate new optimal locations for

the player's ships for the player to aim for during the battle, if the design would be expanded upon.

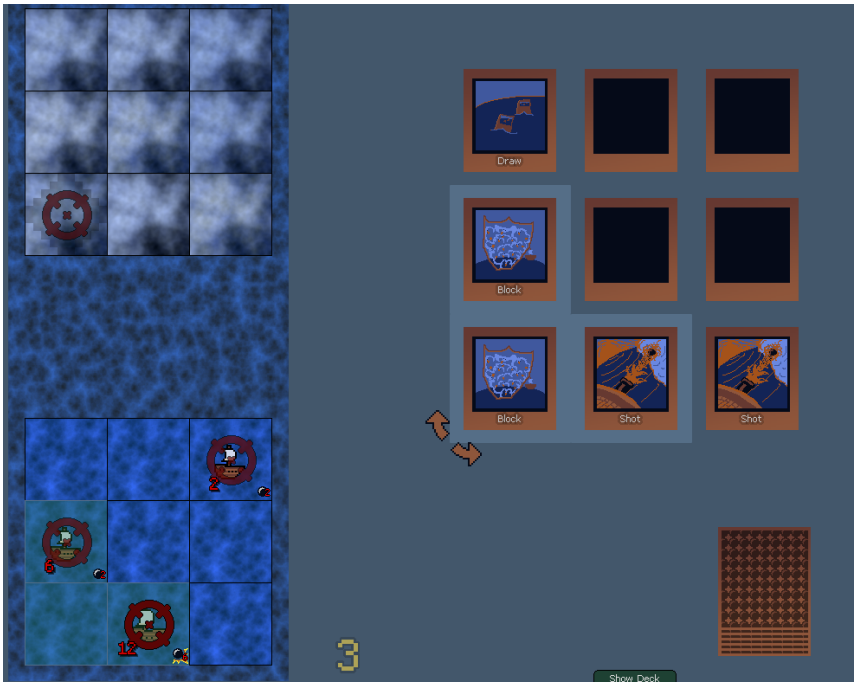

Figure 3.18: Blocking the ship shoots the card adjacent to it as well.

2. *Less button presses to use an action point*: In *choose a coordinate* action the player presses only the target and it chooses the cards automatically. In the alternative design of choosing a card and choosing a target, it is twice as many button presses, as the player also needs to choose a card. Therefore the *choose a coordinate* action is more streamlined than the alternative.

3. *Encourages scouting*: The more cards the player can play, the less likely the player is able to play all of them into a same target. Whenever the player cannot play a card to a same coordinate, they might play them to an another coordinate which, if it's a fogged coordinate, makes the player advance the scouting.

4. *Creates more variation to partially solving*: With the *choose a coordinate* action, the player can play more cards than have targets. If all cards are played

in a turn, then during some *choose a coordinate* actions more than one card is played. There are also situations where no ship has been scouted yet. This means that there are situations where the strictly dominant strategy is to scout using for example two shot cards. Shooting twice into a same coordinate creates variation to the partially solved coordinates, which in turn creates more varied situations for the player to solve.

5. *Enable the design of purely defensive cards*: To have potential to create partially solved coordinates two or more fogged coordinate should be shot during the shooting phase. With *choose a coordinate* action, the player may have purely defensive cards, while still able to shoot to even three different targets. Having purely defensive cards does reduce the impact of scouting as the purely defensive cards are not contributing to scouting unlike offensive cards would. However having purely defensive cards reduces the impact of luck as the player can do defensive actions that are not affected by the uncertainty derived from fog of war.

Choose a coordinate action has at least one disadvantage over the alternative card play design of choosing a card and then choosing a target. When at least one ship has sunk and all of the ships are scouted, the player in most cases runs out of meaningful targets before they run out of actions points. The player is less likely to have to make a difficult choices, when they have excess actions. The opponent also threatens the the player less, so the player can play more offensive cards.

The *choose a coordinate* action could have designed in a way, where it could be activated in the player's play area and/or card grid. Not having multiple places where the coordinate could be chosen was to avoid possible confusion. When allowing for example both play areas to have *choose a coordinate* action, the player might get a wrong first assumption that defensive cards are only activated from the player's play area and the offensive cards are only activated from the opponent's

play area. Showing the player early that they have made wrong assumption can help from confusing the player and may prevent that the player will not make more wrong assumptions based on previous wrong assumptions and further confusing the player. The opponent's play area was chosen over the player's play area for two reasons:

- *The player targets more cards to the opponent's play area*: The player is inclined to play more often purely offensive cards than purely defensive cards as the opponent is not necessarily targetting the player's ship, for example when scouting, and there are more offensive cards in the starting deck than defensive, which forces the player to play more offensive.
- *The movement action input is already in use from the player's play area*: There are less chance for misclicking, when the *choose a coordinate* and movement action are further apart in a separate play areas.

3.7.6 Cards

Cards are objects in a game that trigger an action or actions assigned to them upon being played. Cards are playable only through *choose a coordinate* action. Cards often have one type of actions, but they can also have multiple different types. Card actions can be split in to three categories depending on the action's target:

- *Defensive*: Defensive actions are targeted to the same coordinate in the player's play area as the chosen coordinate in the *choose a coordinate* action and all coordinates adjacent to it.
- *Offensive*: Offensive actions are actions that will set attack coordinates, often one, to the enemy's play area. The default for setting attack coordinate is the chosen coordinate in the *choose a coordinate* action.

- *Global*: When played global actions will happen regardless of the state of the play areas.

3.7.7 Exhaust

Exhaust is an unique card action, which moves the card to the exhaust pile after playing instead to the discard pile. We will call card that has an exhaust action, an exhaustible card. Card that is moved to the exhaust pile stays there until the end of the battle, meaning that exhaustible cards are usable once per battle. Exhaustible cards have often more powerful actions than other cards have.

Cards that are one time use per battle are normal design tool in a lot of rogue-like deck-builders to create variety by letting the player have more powerful cards than normal cards and by being less risky to add to the deck. To motivate the player to pick exhaustible cards over other non-exhaustible cards, the exhaustible cards are designed to be more powerful than their non-exhausting counterparts. As exhaustible cards are one time use per battle, their addition to the deck has less of a negative impact compared to non-exhaustible cards. If the player draws an exhaustible card that they prefer not to redraw in the battle, they may play it to exhaust it away to the exhaust pile.

Exhaustible cards generally reduce the player's power as the fight goes on, which will compensate for the opponent's power loss that happens when at least one of the opponent side's ships sinks. As the player exhaust their more powerful cards to sink a ship, then the player is left with less powerful non-exhaustible cards in his deck and this can be considered that the player's power is reduced.

Exhaustible cards allows in the current design situations where the player may want to avoid playing a card from the card grid. The player might get to a situation where they might use the exhaustible card for no value. This is because the player might be inclined to waste a card due to how *choose a coordinate* action forces the

play of the adjacent cards in the card grid as well. This effect is also increased by the fog of war, where the player might target a shipless coordinate as an outcome of trying to scout for a ship. When the player feels like they should avoid playing some cards in a turn, this creates one more way that the player will interact with the *choose a coordinate* action.

Fog of war may work against the design of offensive exhaustible cards. The player might get an urge to take chances when playing exhaustible card on a fogged coordinate. However when taking a chance is not the best play, the player can be in a situation where they might have to fight against that urge. This can hinder the player's enjoyment of the game.

As exhaustible cards are more powerful and there are less downsides to taking them as a card reward, the player is more likely to take them as a card reward. This can create situations, where the player will pick exhaustible cards over cards that are more synergistic with their deck. If exhaustible cards are too powerful compared to non-exhaustible cards, then it might make the deck-building too much about taking strong cards instead of creating interesting strategies by mixing cards creatively.

3.7.8 Starting deck and card examples

The player has a starting deck, that is a deck that they start each game with. We take a look at some of the cards with their in-game names and descriptions to get a better idea what kind of cards are paired with the rest of the design.

The player's starting deck consist of the following cards:

- $6 \times$ *Shot*: Deals 2 damage on hit
- $4 \times$ *Block*: Gives 4 block

There are about 50 cards that the player may obtain as card reward. We are not showing all of the cards, but some to get a general idea of the cards. Here are

5 cards that are included in the game:

- *Gun Powder Del.*: Gives +1 damage for every shot this turn. (No friendly target needed for this buff to come to effect)
- *Grape Shot*: Shoots in + shape dealing two damage to each tile. Exhausts (Usable once per combat)
- *Draw*: Draw 2 cards on play
- *Draw Shot*: Deals 2 damage on hit. Draw a card on play
- *Big Shot*: Deals 6 damage on hit. Exhausts (Usable once per combat)

4 Empirical study

Now that we have looked at the game, its minimal UI design and its game design in depth it is time to use research methods for finding potential problems in design and finding if the game is fun.

I conducted a small study with 5 participants. An average participant finds about 31% of usability problems. With five participants theoretically about 85% of usability problems can be found as long as the chance of finding an usability problem is uniform [59]. Adding more participants has diminishing returns so smaller participant count is advisable. Two things are in favor of larger participant count:

- Preparing and doing a study has a fixed initial cost.
- There is a lot of variation between participants. With a low sample size, it can be difficult to tell which is an outlier view and what is more generalizable view from the participants.

For this study, a try out of heuristic optimal amount of 5 participants was chosen [59].

Some criteria was applied to the participant. The main criterion was that all participants should have interest in gaming. The secondary criterion was that half of the participants were what Schell [10] calls "expert gamers" in the genre. 100 hours of experience with roguelike deck-building genre was required to be categorized as an "expert gamer". We can use that number and adjust it in the future research. The secondary criterion was enforced for about half of the participant to get even

distribution of potentially "expert gamers" and those without experience with the genre.

This criteria meant that the participants were likely target audience or close to it. If the target audience would not find the game fun, then it is even less likely to find outside of the target audience people who will find the game fun. To only have about half of the participants to have the secondary criterion meant that a more broader set of usability problems could be found. The game artifact shared a lot of familiar elements to the roguelike deck-builder genre, so it broadened the usability problems found when about half had prior knowledge on how a lot of elements already work and about half did not.

Following flexibility options were given for each participants (see Table 4.1):

- *Participating remotely*: Participants who participated remotely.
- *Using only text-based communication*: Participants whose only method of communication in the interview and other parts of the study was through text.
- *Friends and family*: Participants that were part of friends and family. These participants are more likely to not reveal negative opinion about the game, but are highly available

Table 4.1: Flexibility options given to participants

	Participated Remotely	Used only text-based communication	Friends and family
Participant 1	✓		✓
Participant 2			✓
Participant 3	✓		✓
Participant 4	✓	✓	
Participant 5	✓		✓

This sort of flexibility was implemented as there were limited options on willing participants that were not in the category friends and family. One of the reason for not finding willing participants was the session's length, which ended up having the average length of about 84 minutes.
The study had four parts:

1. *Demographics survey*
2. *Playtest*
3. *Semi-structured survey*
4. *Interview*

The details of these parts are introduced more in-depth in their own subchapters.

4.1 Demographics survey

The demographics survey, shown in the Appendix B, exists to get information from the participants that is not directly related to the game artifact as that might distract the participant. After this survey the participant can potentially fully focus on the game artifact.
The demographics survey was a short three question survey which was used to gauge participants' roguelike deck-builder experience and how well did they fit the target audience for the game. The survey was done in Google Forms as survey's done in Google Forms are easy to make and easy to share to the remote participant.

4.2 Playtest

The participants played the game artifact in the playtest part of the session. Participants who were remote, played through Steam Remote Play as no executable did not have to be shared and all of the participants already had Steam installed. The participants did not get any help during the playtest as the game artifact's tutorial and usability design was intended to guide the participant fully. The participant's played until they beat three battles, or one of their ship sunk and they got a game over. The game artifact was designed to fit the playtesters' skill in a way that a game over was unlikely to happen. The estimation for a playtest length was 20 minutes.

4.2.1 Differences

As the participants had only one try to play the game artifact, there was less need for making every run different. The game artifact intended length was only 20 minutes, which meant that it was not a lot of work to hand tailor an experience, that would normally be procedurally generated. The experience was made more similar for every participant with the hand tailored experience to make the research's results more easily comparable to each other. Also the playtest's tutorial was easier to implement, when a lot of game artifact's elements were not procedurally generated. These were the implemented following differences and details to the already introduced game's design to make the research setting more same for the participants:

- *Encounters*: Encounters were not procedurally generated to make the playtests experience more similar for each participant. Every encounter had only one ship type.

- *Map*: As the goal was that the participants would play 20 minutes, the participants were most likely not being able to use the map to an extent, where

the branching path would make a big difference to their game experience. The participants had only one linear path in their map that they could take and therefore making the playtest experience more similar for each participant.

- *Card Reward*: At the end of finishing battles, the participant were given a choice to add a card to his/her deck. The choices were hand picked to make the playtest experience more similar for each participant.

4.2.2 Tutorial and progression

Tutorial was designed for the playtest to make so that the participant would not need external guidance. The tutorial section happens during the whole playtest and is tied how the game progresses.

In the playtest following linear map is used (see Figure 4.1). The map starts with participant facing two battle nodes in a row that they have to play in order to proceed. After the two battle nodes, the participant gets to a node that gives the participant the option to remove one of their card permanently from the deck. The last node is also battle node, where the participant faces the final battle. The last battle node does not have a card reward as the game is meant to end after finishing the last battle.

The playtest has in total 13 tutorial modals with 43 pages of tutorial. If the participant completes the game they have 35 or 36 pages with unique text (see Appendix A) and includes 33 unique pictures paired with the pages. We take a look at a table of tutorial modals (see Table 4.2), but first we should be introduced what each column represents in the table:

- #: The id of tutorial modal.
- *Teaches*: Lists which of the mechanics are explicitly taught in the tutorial modal.

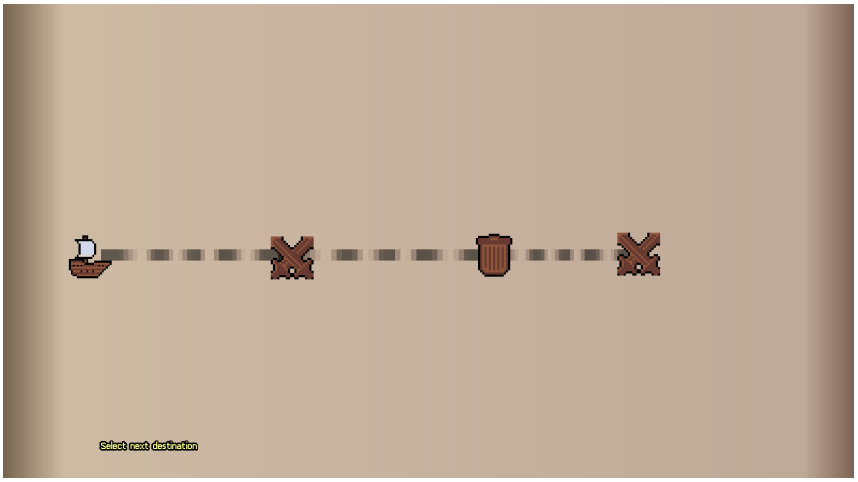

Figure 4.1: The map used in the playtest

- *Trigger*: Lists what triggers this tutorial modal to pop-up.
- *Forced action*: If there is a forced action associated with the tutorial modal, the participant is forced to do the forced action. If the participant does something else, than stated in a forced action then the tutorial modal's last page pops-up. Tutorial modal's last page states in some form the forced action that the player should do as long as the tutorial modal has a forced action associated with it.

The tutorial modals 11 to 13 are special in that the participant will either trigger tutorial modal 11 and 12, or only 13.

Table 4.2: Tutorial modals

#	Teaches	Trigger	Forced action
1.	Win condition and Playing shot cards	Battle 1, start of the turn 1	Forced to click the center cell in the opponent's play area

2.	The impact of playing shot cards and action points	Clicking the center cell in the opponent's play area	Forced to click two cells that will play cards
3.	Ending a turn and what happens at the start of the turn	Playing all cards	Forced to click the end turn button
4.	Scouting logic and hit notes	Battle 1, start of the turn 2	Forced to click the center cell in the opponent's play area
5.	-	Clicking the center cell in the opponent's play area on turn 2	Forced to play card to your old hit note
6.	The opponent attacks as well, what is block and how to play block cards	Battle 1, start of the turn 3	Forced to play the only block card in a way that it will grant block to two ships before running out of action points
7.	Loss condition	Playing the block card, on turn 3	-

8.	Card Grid Rotation	Battle 2, start of the turn 1	Forced to rotate card grid to correct place before playing any cards and play the only block card to the ship in the top right cell in the grid before using all action points
9.	-	Battle 2, start of the turn 2	-
10.	-	Win the last battle	-
11.	Movement points and how to move as a player	Reaching 10 movement points before the opponent does	-
12.	How does the opponent move	The opponent reaches 10 movement points after the player	-
13.	Movement points, how to move as a player, and how does the opponent move	The opponent reaches 10 movement points before the player	-

Here are some playtest specific details for each battle that were added to give a difficulty progression for the participant, to make the playtest more similar for each participant or to make the tutorial's modal match with the playtest:

- *Battle 1*: The participant faced an encounter with three ships with 6 health and shot damage 2.

Two changes were done for the first battle so teaching can be done more evenly between battles:

- *The card grid rotation is disabled for this encounter.*
- *The opponent's movement points are disabled for this encounter.*

There are following turn specific changes that make the game play differently than normally:

- *Turn 1*
  - At the end of the turn, the opponent's ships positions are changed so that the player gets one and only one hit.
  - The opponent shooting is disabled for this turn.
  - Specific cards are drawn to specific cells in the card grid.
- *Turn 2*
  - The opponent shooting is disabled for this turn.
  - Specific cards are drawn to specific cells in the card grid.
- *Turn 3*
  - The opponent attack coordinates are chosen.
  - Specific cards are drawn to specific cells in the card grid.

At the end of the battle 1 the participant had the following card rewards to choose from:

- *Draw*
- *Big Shot*
- *Gun Powder Del.*

- *Battle 2*: The participant faced an encounter with three ships with 10 health and shot damage 2.

  There are following turn specific changes that make the game play differently than normally:

  - *Turn 1*
    - The opponent attack coordinates are chosen.
    - Specific cards are drawn to specific cells in the card grid.

  At the end of the battle 2 the participant had the following card rewards to choose from:

  - *Big Shot*
  - *Grape Shot*
  - *Draw Shot*

- *Battle 3*: The participant faced an encounter with three ships with 16 health and shot damage 2, except every three turns one the attacks had shot damage of 6.

4.2.3 Notable errors of the playtest

There was only one notable bugs which did affect the playtest and one tutorial missed a one important info that the participant could have used.
When the player's play area's ship's health points losing animation is triggered and block has been used to that ship, the blocked amount is incorrectly added to the health points label and the ship loses the added health points, while the block points label is incorrectly not having any kind of animation even though the ship should be visually as well losing block points.

In the last page of the third tutorial modal, the tutorial modal that should teach how to play *block* cards, there is no mention where the player should click to play the block cards. This was only hinted in first tutorial modal's third page with the following tutorial text: "When clicking on the enemy's coordinate grid, all of the highlighted cards are played." (see Figure 4.2).

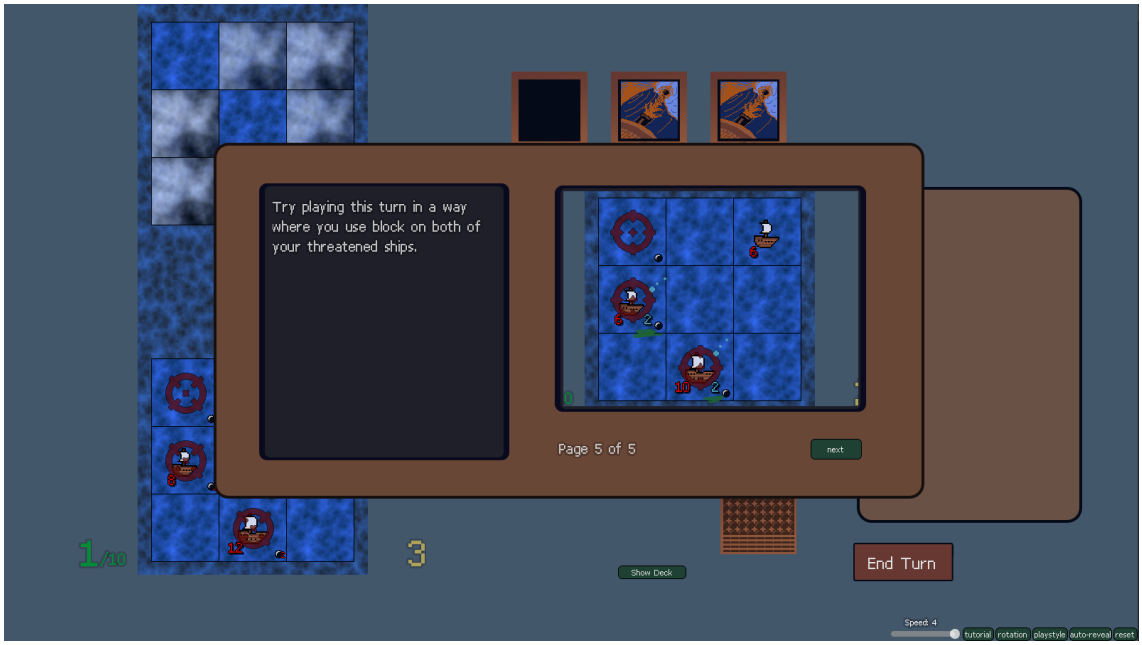

Figure 4.2: The last page of the third tutorial modal

4.3 Semi-structured survey

The semi-structured survey's (see Appendix C) goal is to find problems in the design, which can be further explored in the interview. There were 6 categories for questions: The card grid and playing cards, movement, scouting, combat, clarity, and tutorial. Each category started with likert scale questions and then was followed with open question, where the participant may elaborate their answers to likert scale questions. There was also free form question at the end, where the participant may mention something that they feel like is worth to mention, but did not fit any category. The survey was done in Google Forms as survey's done in Google Forms are easy to make and easy to share to the remote participant.

The likert scale went from 1 to 5 and each number was explained as a following:

1. *Totally disagree*
2. *Somewhat disagree*
3. *Neither disagree or agree*
4. *Somewhat agree*
5. *Totally agree*

All likert scale questions were formatted in a way, where totally agree meant as a positive thing for the quality of the game artifact.

4.4 Interview

The goal of the interview was to further explore the results from semi-structured survey and to figure out do the participant find the game playable. Every interview included the three following questions:

- What did you enjoy the most about the game?

- Do you see potential in the game?
- What steps needs to be taken to reach that potential?

These questions were used to get an idea how playable the game is. Additionally the participants were asked questions that were related to their answers of the semi-structured survey. These questions were asked in most cases when the participants had unexplained answer in a likert scale question that would suggest they had a problem with a playability related to that question. Additionally when there was a room for elaboration in the open questions, a question was created for the interview. When a participant brought up a potential issue and it was concluded as an issue, it was not addressed in the future interviews. This freed time and effort to explore still potential issues in the game artifact more thoroughly.

5 Results

In this chapter we look at the results from the study. We take go quickly about details from the demographics survey and playtest. After that, we will do a thematic analysis on the interviews and semi-structured surveys.

All of the participants were between age of 28 to 42 and their length of the whole session can be seen on Table 5.1. To get a better idea of the participants, Table 5.3 shows the length of the playtest session in turns and in time. The Table 5.2 looks at every participants experience with roguelike-deckbuilders, which roughly correlates on who is an "expert gamer".

Table 5.1: Length of the session for each participant (P)

Participant	P1	P2	P3	P4	P5
Session length	56min	140min	67min	79min	77min

Table 5.2: The experience with roguelike deck-builder for every participant (P)

	P1	P2	P3	P4	P5
Experience with roguelike deck-builders	0h	330h	150h	1000h	0h

Table 5.3: The length of the playtest session in turns and in time (P)

	P1	P2	P3	P4	P5
Battle 1 length	5:26	12:34	10:57	8:24	5:56
Battle 1 turns	6	8	9	9	8
Battle 2 length	2:58	11:16	11:00	5:28	6:36
Battle 2 turns	6	8	9	10	15
Battle 3 length	8:10	21:15	6:22	7:22	2:12
Battle 3 turns	12	12	15	15	6
Average turn length	41s	87s	49s	37s	30s
Total playtest length	17:40	46:55	30:09	22:03	16:03

The participants had a quite a lot of variance for average turn length. Participant 2 took almost three times longer to complete each turn than the participant 5. For participant 5, the game ended on a game over instead of a win. The opponent still had 29 of total of 48 health points left on its fleet. The loss might be able to partially attribute to the speed that the participant played the game with.

5.1 Reflexive thematic analysis

In this thesis reflexive thematic analysis is used to identify and report patterns from qualitative data. Reflexive thematic analysis is used over other thematic analyses as other types of thematic analyses rely on more predetermined themes and codes, which can be too narrow for this thesis' research questions. [60]

5.1.1 Placeholder graphics caused confusion

There were problems with graphical quality that could have caused disruptions in the playtest. Many of the participants considered the placeholder graphics cause confusion. Movement note was the biggest cause for placeholder causing confusion. Participant 2 said the following about the movement notes: "I still do not know

what that green smudge is under the black arrows. Find it confusing.". Participant 2 laughed, when they were told that the green art was representing a ship. Participant 3 asked the following "What is that green booger on a movement note?" and said "Movement note was not pretty, but readable.".

There were various other problems. Participant 4 had problems with the size of hit notes. They wrote that "the numbers was too small" when talking about the hit notes. Participant 2 thought that the card rotator arrows "were maybe little smudgy". Participant 5 wrote the following: "The tutorial section could use better / more highlights on what is what and what should be pressed at any given point."

5.1.2 Blocking was not taught well

Blocking suffered from two implemented errors in the playtest, which we introduced in Section 4.2.3. The first error was a visual bug where instead of a ship having block points losing animation, the lost block points were added to the health and it triggered health loss animation. Participant 1 pondered the following in the survey: "How much damage does it block or does it make your ship gain health points?". Participant 4 stated the following in the survey: "It was hard to tell how much damage each ship was taking since ever though my ships had blocked, it HP went up for that turn due to shields". This error affected how the blocking was understood.

The second error received even more feedback. The tutorial modal that was supposed to teach how to play *block* cards, did not mention where to click to use the *block* cards. This was only hinted in the starting tutorial modal with the following page text: "When clicking on the enemy's coordinate grid, all of the highlighted cards are played". The participant 1 wrote this in the survey about the block: "The block effect could be a bit clearer - I'm not sure if it was mentioned in the tutorial how it works exactly.". Participant 2 wrote this about the block in the survey: "Only after

trying everything and anything else did I find a way to do it. It did not feel intuitive at all that shielding my own ships is actually a shot towards the enemy. Took a good while before it felt natural. Some lore into why it works this way would make it more enjoyable and faster to grasp.". Participant 3 said the following about the block in the interview: "My biggest gripe with the game was that you did not teach how to block".

The participants were all inclined to play the *block* cards by clicking their own ships. Participant 2 said in the survey the following "I caught myself trying to use the block cards by clicking on my own units". Participant 3 also said the same thing in the interview.

As playing block cards is a part of the functionality of the card grid, participants had difficulties with the learning of the card grid as well, which will be further explored in Section 5.3.3.

Participant 1-4 eventually learned to play block cards optimally while participant 5 was unaware that they played block cards wrong until the interview.

The tutorial modal that taught block was also not the simplest possible tutorial for blocking. The tutorial modal's last page had the following text: "Try playing this turn in a way where you use block on both of your threatened ships.". This tutorial modal could be split into two tutorials: One where the goal is to block single ship and one where the goal is to block two ships like in the current tutorial modal. Also the tutorial modal gives the player the responsibility to find one of the two cells in the grid, where playing the *block* card results in two ships getting block points. The tutorial modal could guide, which cell to press to further make sure that the player learns to play *block* cards.

5.1.3 Card grid is not easy to learn

There was a trend of participants not understanding the card grid properly until they got to the second battle and were taught how to rotate the card grid. Participants 1 to 4 could be seen in the playtest struggle to use card grid in the first battle, but use the card grid optimally in the second battle after learning about the card grid rotation mechanic. Participant 2 even said in the interview that "Spinning the card grid made it easy to learn". Participant 5 wrote in the survey that "The ability to rotate grid somewhat helped to understand the card grid mechanic better". Participant 3 said in the interview that "Rotating the card grid was an eureka moment", however it is not clear did this quote reference that the participant 3 understood how the card grid works because of the rotation mechanic or something else. After this feedback, restructuring tutorial to make the card grid rotation tutorial earlier should be consideration in the future iteration. However card grid tutorial also forced the participant to play *block* card correctly. So participants reporting that they learned how to play *block* cards from the card grid rotation tutorial might be a result that some participants learned how to finally play block cards from the tutorial.

There were some learning difficulties with the card grid that probably is not attributed to the difficulty of playing the *block* cards. Participant 4 wrote in the survey the following: "In the beginning of the game, I could not understand how cards were being played since I did not know where to click, let alone that you have to click on the enemy grid to play cards, and it plays the card in the spot on the grid and all cards next to it up, down, left or right.". Participant 3 said in the interview the following: "Card grid is complex in a multitude of ways. ... I think the most difficult mechanic in the game is the card grid.".

Participant 5 did not understand the card grid until in the interview, when it was explained to them. Participant 5 thought that only one card was played with each click, which would be the card that was in the same coordinate. They said in the

interview that “I did not notice nearby highlights.” and to fix this they suggested that “The highlights should be more pronounced.”. Upon demonstration of the card grid mechanic the participant said: “I missed the whole cross mechanic of the card grid.”. The participant did not notice the play areas when using the card grid and vice versa. All of the participants also had this problem to an extent as every participant did not know where to click to play a block card, even though each participant would have already played 4 *block* cards when faced with the tutorial that teaches how to play *block* cards. Combining the three grids into one grid may be a solution that could have helped the participants, but is not a design that works well with the other design. Increasing the highlighting in each grid can help as it makes the principle of similarity more prominent over the Gestalt law of proximity and might help the player visually group the elements in more aligned way to the gameplay. Adding more animations that will guide the focus of the player will help in situations like these, but also moving the card playing input to the card grid will help.

All of the participants assumed first that, if the *shot* cards were played from the opponent’s play area, then the *block* cards are played from the player’s play area. Participant 5 said in the interview “It’s bit weird that you have to shoot the block” while referencing having to click the opponent’s play area for playing *block* cards. To avoid the problem of some players assuming that *shot* cards are played from the opponent’s play area and *block* cards are played from the player’s play area, playing the cards by clicking the card grid might be a better solution. It can also help with the problem of the participants not noticing what happens in the other grids. When the player has to click the card grid it is easier to notice which cards are played compared to clicking the opponent’s play area, assuming that the player has their sight focused on their cursor when doing the action. When each cards start their play animation upon getting played, the player notices these animations more likely

assuming again that the player has sight focused on their cursor on the card grid where they just clicked. Card playing animations demonstrates how card grid works by spawning moving elements towards targets that are affected by the playing of that card.

5.1.4 Movement was not taught well

The participants all had problems with movement. When we take a look at the participants likert scale in Table 5.4 answers we notice something interesting.

Table 5.4: Participants likert scale answers for understanding of how each side moves

	P1	P2	P3	P4	P5
I easily understood how the player can move.	5	1	1	3	1
I easily understood how the enemy can move.	3	4	5	2	4

In general the participants claimed to understand one side's movement while not understanding the other side's movement. Participant 2, 3 and 5 all mentioned that they "did not understand how the player could move" and they did not brought up the opponent's movement in the survey or the interview. Participant 1 had this to say about the movement showing an opposite understanding to participants 2, 3, and 5: "I'm not sure how the movement differs between players and the CPU - the CPU moves all ships?". Participant 4 wrote this about the opponent's fleet's movement "For the AI, it was too confusing and the tutorial made no sense to me as to how they move when a ship is covered in fog or not.".

Tutorial modals related to movement taught a mechanic, but unlike other tutorial modals that taught a mechanic, did not enforce that did the player learn something from a tutorial. Two participant suggested to have a step by step tutorial for movement. Participant 5 wrote "Player also should not be able to skip wind / movement tutorial without moving a ship.". Participant 2 had this to say: "I wish there would

have been more step by step tutorial that would guide me as I move.".

Included in the same tutorial modal were also movement points and their gaining, which was also misunderstood. Movement points are gained from each hit done instead participant 4 attributed the gaining to turns: "The player movement was simply 'after 10 turns, you can move one of your ships one space up, down, left, or right.'".

The participants having confusion related to the movement might be explained partially by its infrequency. Participants 1 wrote the following: "Movement was fine but did not seem like a major mechanic due to how rarely it occurred, perhaps once a game.". Movement requires to have further testing to see its effect. *Rule of doubling and halving* [61], [62] will be used in the future tests to make sure that movement does not happen too infrequently and can actually be tested. This is where a value is more drastically changed than the designer considers to be reasonable to make the value change's impact more distinguishable and to avoid scenarios where the change did not have enough impact and the testing must be redone again.

5.1.5 Partially solved coordinates is a difficult concept

Partially solved coordinates and hit notes were difficult concept to understand for a lot of participants. Participant 3 said this about hit notes "I did not know how to use, but maybe I guess it's possible to learn to use them". Participant 4 did not understand that the hit notes' color just represented which turn the hit note was created: "I could not notice by color if it was a hit or miss.". Participant 4 also missed the concept of partially solved coordinates: "Some things were not explain well like how you're suppose to find an enemy, when you know you hit an enemy".

5.1.6 Tutorial misled into listening to sounds

The fourth and fifth tutorial modal mentioned that audio cues are used to locate the enemy. The purpose of telling the participant about audio cues was to give a lore and logical reason for why partially solved coordinates exists. The purpose was not for the player to listen to the audio cues for gameplay advantage as the hit notes was more efficient way to get the same gameplay advantage. Participants 2-4 got an impression at some point that listening to audio cues was important as a player. Participant 2 said in the interview that "The tutorial made it seem like it was important to listen. I could conclude that it was not necessary from the displayed elements.". Participant 3 asked in the interview "Were you actually supposed to listen in the game?". Participant 4 wrote "For the sound feature, the sounds went too quickly that I could not tell what sound was a 'hit' and what was a 'miss.'".

5.1.7 Mechanics were taught too fast

The previous themes suggests that the playtest might have had too overwhelming of a tutorial. Everything what the participants had problems with were explained in tutorial modals, but were either misunderstood or forgotten by the participants. Participant 4 wrote the following about the tutorial "To me, the tutorial felt a bit too rushed to learn everything". Participant 3 said that "tutorial was not too short and instead there should be more tutorial".

Participant 1 suggested as a way to make the tutorial less overwhelming and avoid teaching too many concepts at the same time, the first battle could be split into two, one without fog of war, and one with. Participant 3 said the following after I brought the participant 1's suggestion: "I think card grid being too complex, the fog of war should be taught in a separate battle. I like the idea of having one fight without the fog of war mechanic so you can properly learn card grid."

5.1.8 The battles were too long

Participant 3 argued multiple times that "Each battle went on for too long.". Participant 3 wrote that "The battles dragged a bit too long than enjoyable -> knowing where the enemy ships were and just repeatedly having to shoot them kind of lacked depth with the cards I had.". Other participants did not brought this theme up, but the participants had a long periods of not getting challenged by the game during the playtest. Further testing is advisable as there is not clear conclusion do the battles feel too long for players, but design in sections 3.4.8 and 3.7.5 predicts that this might be a problem and there were hints of this in the playtests as well.

5.1.9 The game is fun

After asking in the interview what was enjoyable in the game, participants mentioned the card grid and scouting. Participant 4 had this to say when asked what they found enjoyable about the game, which sums up a lot what other participants did say as well:

> 1. The grid based system. This was a first time I tried something like that in a deck-building game, and clicking a space that plays the card on it as well as the cards next to it is a first since you can play 1+ cards for one action point.
>
> 2. Hunting down where enemies are. You basically need to find them on their grid first, and afterwards, you are able to damage them as long as an attack card is next to their spot.

Participant 5 said that they liked it because it was "snack-sized experience". Before mentioning they liked the card grid a lot, the participant 2 said that they "like puzzle games like my game or Into the Breach".

Not a single participant mentioned that they liked the movement mechanic.

5.1.10 The game has potential

When asked did they found potential in the game artifact, every participant said they did. Here we explore some suggestion that the participant mentioned that they want to see in the future iterations.

Progression was obviously missing from the game artifact and the participants stated that they wanted to see more of it. Participants 2, 3, 5 stated that they would like to see more cards and being able to progress so they could get to build a deck. Participant 1 wanted to move more in the map and wished that there would be final boss at the end of the map. Participant 3 stated that they wanted to see more synergies and *relics* to create these synergies and to look at *Slay the Spire* and *Balatro* for inspiration for the *relic* design.

Participant 4 had various suggestions:

- There should even more ways to manipulate the grid so that the player could have an option to play the center card to corners instead of being limited by the randomness.
- The exhaust effect should be reworked so that a card is not wasted if it misses.
- "The player should be able to play their ships at the start of combat in spaces of their choice, instead of being in 3 random spots"
- "Each ship should have their own passive ability and cards they add to your deck to start with, and if you do attack, said ship is the one attacking instead, and they cards are removed from the battle if the ship is destroyed"

6 Discussion

The reflexive thematic analysis suggests following issues with the game artifact.

- *Visual Clarity*

  The game having visual problems that cause confusion or make noticing key elements more difficult.

- *Too much complexity*

  The game having too many mechanics for having satisfying early game experience or the game having too short of a tutorial section.

- *Battle's length*

  The game's battles' length are too long. The game's battles drag for too long.

- *Movement not significant part of the game*

  The movement happens too infrequently in the game.

To fix these issues changes are to be made to next iteration.
There is a lot of research data collected for the current iteration, which means that we have good understanding of this current iteration. *Rule of doubling and halving* [61], [62] will be used for the next iteration as it will be easier to understand what is attributed to the drastic design changes due to familiarity with the current design. Fixing the clarity issues involves experimentation with playing from the card grid instead of the opponent's play area. It also involves a lot of placeholder asset fixes

and adding a lot of animations. Card grid should have stronger highlighting to use more principle of similarity.

Reducing the complexity should start with trying to find mechanics that increase the perceived complexity of the game while not having enough proof of "fun". Movement (Section 3.6), and partially solved coordinates (sections 3.5.2 and 3.5.3) and the opponent's attack coordinate generation (Section 3.5.5) are all possible candidates for this.

Partially solved coordinates did not show enough value to fit the early game experience. It seemed to add complexity without a lot of value for the player. Partially solved coordinates with hit notes will be removed in the next iteration. It will be considered to be added back in later iterations as an optional special modifier, for example, as part of an effect of *Slay the Spire's* equivalent of a *relic* or modifier in a custom game. In the next iteration when a coordinate is hit it will be always revealed instead of sometimes being partially solved.

Opponent's attack coordinate generation may be reworked to much simpler alternative as it has been responsible for a longer iteration cycles due to its implementation cost. Not a single participant reported anything related with opponent's attack coordinate generation and how it actually worked was most likely unnoticed by the participants. Similar to opponent's attack coordinate generation explored in this thesis might be reintroduced in later iterations depending if it still fits the design.

As mentioned in Section 3.5.2 partially solved coordinates makes the scouting slower. Making the scouting faster by removing partially solved coordinates allows for more shorter battles. It also reduces chance elements which will indirectly allow lower health pools as explored in Section 3.5.4, which in turn makes the battles more shorter.

In the next iteration movement could be removed, but instead it will be given one more chance as the thematic analysis was shown to be inconclusive about the value

of movement. As it was inconclusive in the next iteration the movement will be more integral part of the playtest experience to get a conclusion for its value and how well it fits the rest of the experience.

As the scouting becomes faster it allows the design of a more faster movement cycles which is explored in Section 3.5.1. This is in line with the goal to make movement more significant part of the game.

As movement is tied to partially solved coordinates design a new kind of movement design should be done. Multivariate testing will be used to find out what sort of movement design and fog of war design and opponent's attack coordinate generation design is the most ideal for the game's experience. The movement section (Section 3.6.2) explores how all found alternatives for design have design problems.

The battle's dragging in for too long is not addressed except making the battles in overall shorter. Two new mechanics will be added to the next iteration to make "won" battles end faster.

- *Damage by proximity*

  The player uses movement optimally by avoiding opponent damage while keeping the ships in an optimal place for playing defensive cards. The player could exploit this system by moving back and forth, which offers not a lot of interesting challenge for the player. A new mechanic is suggested where the damage increases for each player's ship in range of the defensive card grid area of effect. The player's ideal fleet location changes between each battle as the opponent's ships' location is randomised at the start of the battle meaning that the player may start in not ideal fleet location and work towards ideal location as the battle continues. This causes the player's damage increase as the battle continues and the battle end more faster. The mechanic also makes the movement more significant part of the game.

- *Action modules*

The player has excess action points often when there is one sunken opponent's ship. A new mechanic is suggested where the player may trade action point for a positive effect. For example, the player may use an action point to shuffle their card grid's cards for possible stronger *choose a coordinate* action. These modules works as buttons and may be displayed in the bottom left of the view with an resource, an arrow and an icon of an effect. This also allows create more clarity for movement as movement can be made into one of these modules. Having a button displayed with 10 movement point resource, an arrow, and an icon of a ship with green arrows could help to answer problems discovered in thematic analysis related to participants not knowing how to use the player's movement.

The current iteration's tutorial's goal was to offer similar experience to the real run, where the player does not feel like the procedural generated elements are placed by the developer. Having a tutorial that has significantly different design than the non-tutorial gameplay might turn away potential customers. However the tutorial was overwhelming experience as there were too many mechanics included in it at the same time so this compromise will be trialled in the next iteration. A new kind of tutorial is suggested where it resembles more of a prologue experience and does not resemble a real run. In the tutorial, the player starts with lesser versions of the mechanics where the gameplay is very simple for the player and the player gets the mechanics back gradually between battle sequences. For example, the first battle may start with a 2×2 play area and card grid, only one ship for each side, the player has the ability only draw two cards per turn, playing cards does not play adjacent cards, and cards that are played to defensive area are not played to adjacent coordinates.

We tried to answer to two research questions:

1. ***Is the game design shown in this thesis playable in terms of usability***

as measured by effectiveness and satisfaction?

2. ***What kind of user interface elements does the game's design require to function?***

Playtest and thematic analysis showed that the participants had difficulties with a lot of mechanics. By the end of the playtest, participants were using effectively the taught mechanics except everyone had difficulties with movement, some had difficulties with partially solved coordinates and one participant had difficulties with understanding how to play cards from the card grid. Interview result indicated that the participant found the game satisfying. The answer to the first research question is that the game design is playable, but it can be improved upon.

We explored thoroughly in the design, the design for user interface and its limitations. The participants' feedback showed that there were some usability problems with the user interface, which means that their design need to be iterated more. Participants criticized user interface elements like movement notes, card grid rotation arrows, hit note's size, and lack of highlighting and animations. Fixing these will make the game artifact's user interface function better.

7 Conclusion

In this thesis, we researched the playability of a novel roguelike deck-builder game in terms of its usability and its user interface's functionality. The game was introduced as a roguelike deck-builder game with elements from Battleship board game, especially its guessing mechanic. Section 3.1 explored how the initial design was created and some design principles that were used in the whole of the design. In sections 3.3-3.7, we went in depth in game's design and the design of its user interface. We explored design, how it connects together and alternative designs as considerations for considerations in potential future iterations and as demonstration to understand why sometimes conventional game design does not work.

A study was created to collect data from the game. 5 participants were chosen for the study based on their availability and experience with gaming and roguelike deck-building games. To get more variety to the viewpoints presented, about half of the participants were experienced with the genre, while the other half were not. Qualitative approach was chosen for research of the game. Starting with a demographics survey, followed by the playtest, playtest survey, and interview was concluded to be effective way to collect data about the game for further analysis. Reflexive thematic analysis was used to analyse our findings from the playtest survey and the interview. The reflexive thematic analysis resulted in finding multiple design flaws. The design flaws found were problems with visual clarity, having too much complexity for the player, and major problems with game balance.

The design flaw themes and general data collected were used to estimate effectiveness and satisfaction of game design's playability in terms of usability. We concluded that the game had design problem that affected its usability's effectiveness and satisfaction, but despite this it still resembled a playable game.

Some of the design flaw themes were used to conclude that the user interface had some problems that could be fixed in the future iterations.

We suggested design changes that would improve the playability of the game while trying that the design remains internally consistent. As complexity was big problem, a lot of features that were underused but which increased complexity were either removed or reworked for a future iteration. Some small mechanics were suggested to tie the design together and to help fixing the balance issues the game had. Adding visual highlights, redoing the more unclear art, adding animations to key places, and changing the player's focus when playing were the main ways we suggested to improve the visual clarity of the game.

References

[1] J. L. G. Sánchez, F. L. G. Vela, F. M. Simarro, and N. Padilla-Zea, "Playability: Analysing user experience in video games", *Behaviour & Information Technology*, vol. 31, no. 10, pp. 1033–1054, 2012.

[2] C. Fabricatore, M. Nussbaum, and R. Rosas, "Playability in action videogames: A qualitative design model", *Human-Computer Interaction*, vol. 17, no. 4, pp. 311–368, 2002.

[3] H. Desurvire, M. Caplan, and J. A. Toth, "Using heuristics to evaluate the playability of games", in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1509–1512.

[4] H. Korhonen and E. M. Koivisto, "Playability heuristics for mobile games", in *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, 2006, pp. 9–16.

[5] J. González, F. Montero, N. Padilla-Zea, and F. L. Vela, "Playability as extension of quality in use in video games.", Jan. 2009.

[6] J. Paavilainen, "Defining playability of games: Functionality, usability, and gameplay", in *Proceedings of the 23rd International Conference on Academic Mindtrek*, 2020, pp. 55–64.

[7] R. Zubek, *Elements of Game Design*. MIT Press, 2020, pp. 1–19, ISBN: 9780262362870. [Online]. Available: https://books.google.fi/books?id=0s_tDwAAQBAJ.

[8] C. Cleveland. "The design of Subnautica", Youtube. (Jul. 9, 2019), [Online]. Available: https://youtu.be/7R-x9NSBS2Y?t=2603.

[9] E. J. Conklin, *Dialogue mapping : building shared understanding of wicked problems*, eng. Hoboken, NJ: Wiley, 2006, ISBN: 9780470017685.

[10] J. Schell, *The Art of Game Design: A Book of Lenses*, eng, Third Edition. United States: A. K. Peters, 2019, pp. 479–495, ISBN: 9781351803649.

[11] "Josh Weier interview (Portal 2, Valve, game design, Gabe Newell, play testing, management, crunch, certification etc.)", Listen Notes. (Feb. 26, 2022), [Online]. Available: https://www.listennotes.com/podcasts/kiwi-talkz/126-josh-weier-interview-hRrUGebZ1w9/?t=460.

[12] J. Schell, *The Art of Game Design: A Book of Lenses*, eng, Third Edition. United States: A. K. Peters, 2019, p. xxxv, ISBN: 9781351803649.

[13] E. Adams, *Fundamentals of Game Design*, eng, 3rd ed. Berkeley, CA: New Riders, 2014, p. 31, ISBN: 0-321-92967-5.

[14] R. Zubek, *Elements of Game Design*. MIT Press, 2020, pp. 201–204, ISBN: 9780262362870. [Online]. Available: https://books.google.fi/books?id=0s_tDwAAQBAJ.

[15] *Portal*, https://store.steampowered.com/app/400/Portal/, Accessed: 2025-05-06.

[16] *Dream Quest*, https://store.steampowered.com/app/557410/Dream_Quest/, Accessed: 2025-05-06.

[17] *RDBG #2: Dream Quest (2014)*, https://shrunkenshrine.com/2022/12/17/rdbg-2-dream-quest-2014/, Accessed: 2025-05-06, Dec. 17, 2022.

[18] L. Gordon, "How one of gaming's most intimidating genres spawned a legion of hits", *The Verge*, Jun. 30, 2022, Accessed: 2025-05-06. [Online]. Available: https://www.theverge.com/2022/6/30/23181999/roguelike-deckbuilder-genre-slay-the-spire-inscryption.

[19] *Slay the Spire*, https://store.steampowered.com/app/646570/Slay_the_Spire/, Accessed: 2025-05-06.

[20] S. Nam and P. Rerkjirattikal, "Exploring the impact of randomness in roguelike deck-building games: A case study of slay the spire", in *Integrated Uncertainty in Knowledge Modelling and Decision Making*, V.-N. Huynh, B. Le, K. Honda, M. Inuiguchi, and Y. Kohda, Eds., Cham: Springer Nature Switzerland, 2023, pp. 97–108, ISBN: 978-3-031-46775-2.

[21] *Balatro*, https://store.steampowered.com/app/2379780/Balatro/, Accessed: 2025-05-06.

[22] *The Balatro Timeline*, https://localthunk.com/blog/balatro-timeline-3aarh, Accessed: 2025-05-06, Mar. 6, 2025.

[23] *Berlin Interpretation*, https://www.roguebasin.com/index.php?title=Berlin_Interpretation, Accessed: 2025-05-06.

[24] Y. Liang, "A case study in the game as we descend-key design considerations for implementing controller support in fast-paced deck-building games", 2024.

[25] *Luck be a Landlord*, https://store.steampowered.com/app/1404850/Luck_be_a_Landlord/, Accessed: 2025-05-06.

[26] *Luck be a Landlord*, https://store.steampowered.com/app/2097570/StarVaders/, Accessed: 2025-05-06.

[27] *Roguebook*, https://store.steampowered.com/app/1076200/Roguebook/, Accessed: 2025-05-06.

[28] *Tainted Grail: Conquest*, https://store.steampowered.com/app/1199030/Tainted_Grail_Conquest/, Accessed: 2025-05-06.

[29] *Ratropolis*, https://store.steampowered.com/app/1108370/Ratropolis/, Accessed: 2025-05-06.

[30] K. Stuart, "Dungeon crawler or looter shooter? nine video game genres explained", *The Guardian*, Oct. 11, 2021, Accessed: 2025-05-06. [Online]. Available: https://www.theguardian.com/games/2021/oct/11/modern-video-game-genres-explained-metroidvania-dungeon-crawler.

[31] K. Stuart, "'Dicey Dungeons' will help you understand the best new genre in games", *Vice*, Aug. 13, 2019, Accessed: 2025-05-06. [Online]. Available: https://www.vice.com/en/article/dicey-dungeons-will-help-you-understand-the-best-new-genre-in-games/.

[32] M. Holmes, "The best roguelike deckbuilders: Game recommendations for any card shark", *Rogueliker*, Sep. 13, 2024, Accessed: 2025-05-06. [Online]. Available: https://rogueliker.com/roguelike-deckbuilders.

[33] S. Scaife, "Review: Slay the spire is a thrilling deck-building roguelike", *Slant*, Jan. 23, 2019, Accessed: 2025-05-06. [Online]. Available: https://www.slantmagazine.com/games/slay-the-spire-review/.

[34] *Astrea: Six-Sided Oracles*, https://store.steampowered.com/app/1755830/Astrea_SixSided_Oracles/, Accessed: 2025-05-06.

[35] *Backpack Hero*, https://store.steampowered.com/app/1970580/Backpack_Hero/, Accessed: 2025-05-06.

[36] J. Gollop, "The Gollop Chamber: Rise of the roguelike deck builder", *PC Gamer*, Feb. 7, 2018, Accessed: 2025-05-06. [Online]. Available: https://www.pcgamer.com/the-gollop-chamber-rise-of-the-roguelike-deck-builder/.

[37] *What are the top selling indie games of 2023?*, https://howtomarketagame.com/2024/01/25/what-are-the-top-selling-indie-games-of-2023/, Accessed: 2025-05-06, Jan. 25, 2024.

[38] *What the hell happened in 2024?*, https://howtomarketagame.com/2025/01/15/what-the-hell-happened-in-2024/, Accessed: 2025-05-06, Jan. 15, 2024.

[39] "Further analysis into Steam Reviews to Sales ratio – how to estimate video game sales", *Video Game Insights*, Oct. 18, 2021, Accessed: 2025-05-06. [Online]. Available: https://www.slantmagazine.com/games/slay-the-spire-review/.

[40] J. Schell, *The Art of Game Design: A Book of Lenses*, eng, Third Edition. United States: A. K. Peters, 2019, p. 161, ISBN: 9781351803649.

[41] https://x.com/Franrekk/status/1826419925923270847, Accessed: 2025-05-06.

[42] *Know your game's anchor*, https://howtomarketagame.com/2019/12/23/know-your-games-anchor/, Accessed: 2025-05-06, Dec. 23, 2019.

[43] *Battleship Salvo Game Rules*, https://www.ultraboardgames.com/battleship/salvo-rules.php, Accessed: 2025-05-06.

[44] J. Hinebaugh, *A Board Game Education*. R&L Education, 2009, p. 120, ISBN: 9781607092612. [Online]. Available: https://books.google.fi/books?id=gUK_AAAAQBAJ.

[45] "The secret behind Blizzard's success | Allen Adham, co-founder of Blizzard", Youtube. (May 20, 2024), [Online]. Available: https://youtu.be/aHVC019NbPo?t=590.

[46] *Allen Adham on Blizzard's design philosophy, hiring gamers, and LiveOps: "Strategic depth does not require complexity"*, https://wnhub.io/news/marketing/item-43504, Accessed: 2025-05-06, May 21, 2024.

[47] J. Staats, *The WoW Diary: A Journal of Computer Game Development [Second Edition]*. Source Point Press, 2024, p. 45, ISBN: 9798888760178. [Online]. Available: https://books.google.fi/books?id=x64I0AEACAAJ.

[48] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (Harper Perennial Modern Classics). HarperCollins, 2009, ISBN: 9780061876721. [Online]. Available: https://books.google.fi/books?id=epmhVuaaoK0C.

[49] J. Nielsen, "10 usability heuristics for user interface design", *Nielsen Norman Group*, Apr. 24, 1994, Accessed: 2025-05-06. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/.

[50] E. Adams, *Fundamentals of game design*, eng, 3rd ed. Berkeley, CA: New Riders, 2014, p. 508, ISBN: 0-321-92967-5.

[51] R. Nickerson, *Cognition and Chance: The Psychology of Probabilistic Reasoning*. Lawrence Erlbaum, 2004, pp. 50, 242, ISBN: 9780805848984. [Online]. Available: https://books.google.fi/books?id=wZ0gXKy53egC.

[52] J. Schell, *The Art of Game Design: A Book of Lenses*, eng, Third Edition. United States: A. K. Peters, 2019, p. 208, ISBN: 9781351803649.

[53] *Random Distribution*, https://dota2.fandom.com/wiki/Random_Distribution, Accessed: 2025-05-06.

[54] *Random Distribution*, https://liquipedia.net/dota2/Random_Distribution, Accessed: 2025-05-06.

[55] *GD Column 17: Water Finds a Crack*, https://www.designer-notes.com/game-developer-column-17-water-finds-a-crack/, Accessed: 2025-05-06, Jun. 12, 2011.

[56] *Timmy, Johnny, and Spike*, https://magic.wizards.com/en/news/making-magic/timmy-johnny-and-spike-2013-12-03, Accessed: 2025-05-06.

[57] G. Elias, R. Garfield, and K. Gutschera, *Characteristics of Games* (The MIT Press). MIT Press, 2012, pp. 106–120, ISBN: 9780262017138. [Online]. Available: https://books.google.fi/books?id=QVP8AQAAQBAJ.

[58] J. Bycer, *Game Design Deep Dive: Roguelikes* (Game Design Deep Dive). CRC Press, 2021, p. 32, ISBN: 9781000361988. [Online]. Available: https://books.google.fi/books?id=FrsYEAAAQBAJ.

[59] J. Nielsen, "Why you only need to test with 5 users", *Nielsen Norman Group*, Mar. 18, 2000, Accessed: 2025-05-06. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/.

[60] S. McLeod, *Thematic analysis: A step by step guide*, Jun. 2024. DOI: 10.13140/RG.2.2.13084.71048.

[61] J. Schell, *The Art of Game Design: A Book of Lenses*, eng, Third Edition. United States: A. K. Peters, 2019, pp. 245–247, ISBN: 9781351803649.

[62] M. Beyer, A. Agureikin, A. Anokhin, *et al.*, "An integrated process for game balancing", Sep. 2016. DOI: 10.1109/CIG.2016.7860425.

Appendix A Tutorial texts

Here is table of texts used in the tutorial. The number represents the tutorial modal. Each row has one list of text used in that tutorial modal. Each element of the list represents the text of one page of a tutorial modal.

	Tutorial text
1.	• Welcome to the ship game! There are 3 enemy ships hiding inside the enemy's fog. Your goal is to sink the enemy ships by playing cards. • Hovering over the enemy's coordinate grid highlights your cards. • When clicking on the enemy's coordinate grid, all of the highlighted cards are played. • Played shot cards will be targetted to the coordinate you clicked. • Try clicking the center coordinate on the opponent's side.

2.	• Great! Your shot card got used to the selected coordinate. • Clicking a coordinate used one action point. Try using the rest of your action points as well by clicking on the enemy grid and playing your cards.
3.	• Great! Now press end turn. Ending a turn discards your cards and moves you to the attack phase. After the attack phase, you start a new turn by drawing new cards and replenishing your action points.

4.	• Nice shots! Did you hear that? You hit an enemy! You heard two splashes which were misses and one wooden crack which was presumably an enemy ship. • Audio cues is the main way to rule out whether a coordinate has a ship or is shipless. Ruling out will clear the fog automatically. • Whenever you cannot deduce what's exactly in the coordinate, a little hit note will be added. • The number inside hit note's box shows how many shots were hits with that turn's barrage. • The amount of shots to the coordinate is displayed by the number of circles/balls on top of the box. • Let's rule out what's under the fog. Click on the middle coordinate in enemy's coordinate grid to play two shots there.

5.	• Nice! Target the last card on your old hit note, and press end turn. The shot coordinates from your last turn will then each receive a different amount of shots. Listen how many times a ship is hit to find out which tile has the ship.

6.	• Great! You have ruled out the location of one of the enemy ships! • Hmm... Oh no! It seems that the enemy fleet is ready to attack your fleet. It is targetting two of your ships. Worry not, we have means of defending against such attacks. • Unlike played offensive cards, which are targetted to a single coordinate, played defensive cards are targetted to every highlighted coordinate. As an example, a played block target can potentially give block to every ship as long as they are inside the highlighted area. • Playing defensive cards gives your targetted ships block. Block reduces incoming attack damage, but wears off at the start of your next turn. • Try playing this turn in a way where you use block on both of your threatened ships.

7.	• Great! Being defensive is important. Losing even one of your ships is a game over! • Now use what you learned and sink the enemy fleet!
8.	• Hmm, it seems that your ship has a card rotator. Let me just put it on... Okay, it's on! • Card rotator allows you to rotate the card grid 90 degrees. • Card rotator can be used as long as you have not played any cards that turn. • One of your ships is threatened. Try to protect the ship this turn by using the card rotator first and then use one of your actions to play the block card to that ship.
9.	• Great! Go beat them!

10.	• Congratulations! You have finished the playtest and become a currently reigning champion of the great lake.
11.	• Oh, look at that! You have 10 wind power. • Each hit on a sailing ship grants a side 1 wind power. Wind power is used for movement. • When you have 10 wind power, you can click one of your ships and click an adjacent tile to move it there. This consumes 10 wind power.

12.	• Oh no... the enemy has 10 wind power. • Enemy movement is different from the player's movement. When enemy has 10 wind power, the enemy will consume the wind power and move the whole fleet. • Enemy ships move differently based on whether you have located them or not. • Located enemy ships move to two possibilities near them. There is a little movement note in the corner to show, the two possible move locations. • Not exactly located enemy ships on the other hand can move to any coordinate and any distance. This is one of the reason, why locating a ship is important.

13.	• Oh no.. enemy has 10 wind power • Each hit on a sailing ship grants a side 1 wind power. Wind power is used for movement. • When you have 10 wind power, you can click one of your ships and click to adjacent tile to move it there. This consumes 10 wind power. • Enemy movement is different from the player's movement. When enemy has 10 wind power, enemy will consume the wind power and moves the whole fleet. • Enemy ships move differently based on, if you have located them or not • Located enemy ships move to two possibilities near them. There is a little movement note in the corner to show, the two possible move locations. • Not exactly located enemy ships on the other hand can move to any coordinate and any distance. This is one of the reason, why locating a ship is important.

Appendix B Demographics survey

This survey is part one of a Master's degree thesis research for University of Turku. There are 4 parts for the research participant and are as follows:

1) Pre-playtest survey, where participant gives relevant background information.

2) Playtest

3) Post-playtest survey, where participant answers questions about the playtest.

4) Interview, where participant is asked questions based on the survey answers and also includes more free form conversation. In text or audio.

It takes about 40-50 minutes total to complete all 4 parts. The goal of the survey is to measure how well are the game mechanics communicated and how well the game mechanics are creating an optimal game experience. The game that is playtested is one part roguelike deck-builder game and one part Battleship-like guessing game. Before starting you should have been given interview id, which you can enter below.

1.	Insert the given interview id.
2.	About how many hours have you played roguelike deck-builders before (two significant digits eg. 250h, 0.2h, 3100h)
3.	List your favorite roguelike deck-builders, starting from your most favorite and going towards your least favorite. Leave empty, if you have none.
4.	How old are you? ○ 13-17 ○ 18-22 ○ 23-27 ○ 28-32 ○ 33-42 ○ 43-52 ○ 53+

Appendix C Semi-structured survey

This survey is part three of a Master's degree thesis research for University of Turku.

1) Pre-playtest survey, where participant gives relevant background information.

2) Actual playtest.

3) Post-playtest survey, where participant answers questions about the playtest.

4) Interview, where participant is asked questions based on the survey answers and also includes more free form conversation.

There are following categories of questions:

The card grid and playing cards, Movement, Scouting, Combat, Clarity and Tutorial.

These category starts with rating scale statements, which you can give rating from 1 to 5. The ratings from 1 to 5 are as follows:

1) totally disagree

2) somewhat disagree

3) neither disagree or agree

4) somewhat agree

5) totally agree.

Category ends with open question, where you can explain reasoning to your

answers, if you want to. All statements are numbered for easier refering in the open question. Adding e.g. "1)" is enough to explain, that you are explaining reasoning for the first statement's answer.

There is free form category at the end, where you can comment or give opinion about something that was not addressed with the statements, but might be relevant information for improving the game.

1.	Insert the given interview id.
2.	I easily understood how to play shot cards. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

3.	I easily understood how to play block cards. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
4.	Playing cards from the card grid was an enjoyable mechanic. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

5.	I prefer playing from a card grid more than playing from a hand. (for example playing from the Slay the Spire's hand) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
6.	I easily understood how to use card grid rotation. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

7.	I found that card grid rotation added to the experience of playing from the card grid. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
8.	Here you can give specifics on why you agreed or disagreed with the statements related to the card grid and playing cards.
9.	I easily understood how the player can move. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

10.	I easily understood how the enemy can move. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
11.	Movement notes were easily readable. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

12.	It did not cause any problems that the enemy and player had different rules for movement. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
13.	I easily understood how to gain wind power. (movement points) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

14.	Movement was an enjoyable mechanic. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
15.	Here you can give specifics on why you agreed or disagreed with the statements related to the movement.
16.	Hit notes were easily understandable. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

17.	It was easy to follow why a fogged coordinate/tile would turn into a sea or ship coordinate/tile. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
18.	Not knowing at the start where the enemies were made the game experience better. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

19.	Hit notes are a good thing, compared to always revealing the coordinates on shot. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
20.	Here you can give specifics on why did you agree or disagree with the statements related to fog and locating enemy ships.
21.	The game's difficulty was right for me. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

22.	The pacing in the combat was good. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
23.	The combat felt best, when there were still some ships under the fog. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
24.	Here you can give specifics on why did you agree or disagree with the statements related to combat.

25.	I was not confused by the shots not coming from the ships. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
26.	The UI elements were clear enough. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
27.	Here you can give specifics on why did you agree or disagree with the statements related to clarity.

28.	I was not overwhelmed by the amount of stuff I had to learn by the tutorial. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
29.	The tutorial had a very good pacing. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

30.	It worked that the hit notes were taught in the beginning. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
31.	I felt tutorials on mechanics, did not have anything unnecessary. (e.g. explaining too many times the same thing, explaining obvious very stuff) ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

32.	The tutorial did not left anything out that felt like important information. ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
33.	Here you can give specifics on why did you agree or disagree with the statements related to tutorial. Especially statement (4) and (5).
34.	Do you have comments about something, that was not in this survey?