Daniela Yli-Kiikka

DESIGNING THE USER INTERFACE FOR WIZARD WARS

Thesis
Kajaani University of Applied Sciences
Bachelor of Business Administration
Business Information Technology
Autumn 2015

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ
TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Daniela Yli-Kiikka	
Työn nimi Wizard Wars:in käyttöliittymän suunnittelu ja toteutus	
Vaihtoehtoiset ammattiopinnot Peligrafiikka	Ohjaaja(t) Raimo Mustonen
	Toimeksiantaja
Aika 10.12.2015	Sivumäärä ja liitteet 70 + 1

Tämän opinnäytetyön tavoitteena oli suunnitella korkealaatuinen, toimiva ja käyttövalmis käyttöliittymä *Wizard Wars*-tablettipelille. Käytettävyys ja toimivuus ovat perinteisesti olleet käyttöliittymäsuunnittelun päätavoitteet visuaalisen ilmeen jäädessä taka-alalle. Peliteollisuudessa visuaalinen näyttävyys on kuitenkin korkeassa arvossa, ja se onkin ainoa tietotekniikan osa-alue missä on hyväksyttävää uhrata tietyn verran käytettävyyttä visuaalisen teeman ja tunnelman luomiseen. Siksi aidosti onnistuneen pelikäyttöliittymän tulisikin olla sekä käytettävyydeltään erinomainen että visuaalisesti puoleensavetävä.

Visuaalisen ilmeen suunnittelun lisäksi käyttöliittymäelementtien keveyteen ja yhtenäisen ilmeen luomiseen käytettiin huomattava määrä aikaa. Nämä tavoitteet saavutettiin luomalla modulaarinen käyttöliittymä joka koostuu rajatusta määrästä graafisia elementtejä, joita uudelleenkäytetään useissa eri tarkoituksissa ja useiden eri menunäkymien välillä. Koska jotkut *Wizard Wars*:in keskeisistä menuista ovat suhteellisen informaatioraskaita, myös tablettien rajallisten näyttökokojen optimaalinen hyödyntäminen oli etusijalla.

Kieli	Englanti
Asiasanat	pelisuunnittelu, peligrafiikka, käyttöliittymät, käyttöliittymäsuunnittelu, ikonisunnittelu, vektorigrafiikka, tablettipelit
Säilytyspaikka	☒Verkkokirjasto Theseus ☒ Kajaanin ammattikorkeakoulun kirjasto

KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

THESIS
ABSTRACT

School Business	Degree Programme Business Information Technology
Author(s) Daniela Yli-Kiikka	
Title Designing the user interface for Wizard Wars	
Optional Professional Studies Game Graphics	Instructor(s) Raimo Mustonen
	Commissioned by
Date 10.12.2015	Total Number of Pages and Appendices 70 + 1

The goal of this thesis was to design a high-quality, functioning and implementation-ready user interface (UI) for the tablet strategy game *Wizard Wars*. UI design is traditionally a discipline where visuals take a backseat in favor of usability and functionality. In games, however, visual impressiveness is valued highly, and it is the only medium where it is deemed acceptable to sacrifice some amount of usability to create a more elaborate design to support the theme and atmosphere. Therefore, a truly successful game UI should be able to provide both an excellent level of usability and an appealing visual design.

In addition to designing a visual style suitable for the theme of the game, considerable effort was put into making the UI elements and layout as light and consistent as possible. This was done by creating a modular design made up of a limited amount of art assets, which were then reused for different purposes and between several menu screens. As some of the central menu screens in *Wizard Wars* are quite information-heavy, the optimized use of the limited screen estate in tablets was also a priority.

Language of Thesis	English
Keywords	game design, game graphics, user interfaces, user interface design, icon design, vector graphics, tablet games
Deposited at	☒ Electronic library Theseus ☒ Library of Kajaani University of Applied Sciences

FOREWORDS

When it was time for me to look for a thesis subject there were a few viable and interesting options available. I wanted to delve deeper into a subject that would benefit me in my game artist career, both as an addition to my portfolio and by providing me with useful theoretical knowledge. The fact that many game artists shy away from UIs in favor of more appealing specializations – and thus create a demand for UI artists in the game industry – led me to focus on the UI design process and the theory behind it.

UI design is a section of game graphics I have been very interested in since my first year of studies, as it is one where the final design depends on so many other factors than just artistic vision. Standardized layout and control schemes, typography and iconography, as well as external variables such as the target audience's age and gaming experience all have to be taken into account. Although UI design does not lend itself to as much artistic freedom as some other areas of game graphics, it offers unique and intriguing challenges.

ABSTRACTS

TABLE OF CONTENTS

LIST OF SYMBOLS

Aesthetic-usability effect
A visual design principle observing that aesthetic designs are perceived as being easier to use than non-aesthetic ones.

Diegesis theory
A design theory where game UI elements are divided into four categories: diegetic; spatial; meta; and non-diegetic, depending if they exist within the game world or not and if they are immersive or not.

Graphical User Interface (GUI)
A menu-driven interface, where control options are displayed visually on the screen in the form of buttons, lists, sliders and switches. The player inputs commands by interacting with these UI elements by clicking, tapping and/or dragging.

Heads-up Display (HUD)
Informational UI elements displayed on top of the in-game view, usually in static positions. Their purpose is to provide information the player needs during gameplay.

In-game view
The view of the game during the main gameplay.

Menu screen
A screen containing options, commands, and information that does not have to be accessible or is not relevant in the in-game view. Depending on the game the menu structure can be very complex, and in some genres navigating menus to input commands and access information is a central part of the gameplay.

Modular
A set of assets that can be connected and combined in different ways to create a variety of bigger, more complicated elements.

Pop-up menu

A menu box appearing on top of the current view, covering only part of the screen. Usually used for confirmation dialogue or to group less-used options together under a single button.

Raster graphic

An image made up of a grid of pixels, often referred to as a bitmap. Raster graphics can be scaled down without any effect, but scaling them up results in loss of quality, meaning the final size has to be known when starting the drawing process. Commonly used file formats for raster graphic are .PNG, .JPEG and .GIF (for animation).

Screen aspect ratio

The proportional relationship between a display device's width and height. Common aspect ratios are 16:9 (monitor and TV standard, iPhone5, Android devices), 4:3 (iPad, older monitors and TVs), and 3:2 (iPhone4, Android devices).

Screen estate

The amount of space available on a display device. In game development this is usually measured in pixels (screen resolution), but the physical dimensions have to be considered as well, especially on mobile platforms with small screen sizes.

Screen resolution

The number of pixels a display device can show horizontally and vertically.

Strategy game

A game emphasizing strategic thinking and resource management. The gameplay is usually complex and slow-paced. A game in this genre is classified as either real-time or turn-based, depending on how time passes within the game.

Touchscreen

A display that also serves as an input device by detecting human touch and/or the touch from a stylus pen.

User Interface (UI)

The elements which enable interaction between the player and the game. It consists of the informational and interactive visuals and audios, and the physical interface.

Vector graphic

An image defined mathematically with shapes consisting of Bezier curves linked together. As these shapes are saved as mathematical formulas with color and opacity information, they can be scaled without loss of quality. Commonly used file formats are .EPS, .AI (Adobe Illustrator) and .SVG (Inkscape).

1 INTRODUCTION

In the past few years the game industry has come to recognize the importance of a well-designed UI. More resources are given to both developing the UIs and testing their functionality and usability. With the emergence of easy-to-access and affordable gaming platforms – most notably mobile devices and web browsers – and the free-to-play business model, games and gaming in general have shifted to be more mainstream and approachable. People who have not even touched a console or computer game before download free games on their mobile phones and tablets, or play them on web browsers through their social media or email accounts. With this expansion in target audience usability has become a key element for a game's success, as it is safe to assume a good number of potential players have limited experience and knowledge about common game concepts and mechanics. To get players hooked, a game has to be easy to understand and use right from the first minute, and the game's UI plays a central role in achieving these goals.

The goal of this thesis is to create a high-quality, functioning and implementable UI for the tablet turn-based strategy game *Wizard Wars*. Designing a game UI is a process involving people from various areas of game development. This thesis will present the design process from an artist's point of view and focus on the visual aspects of UI design. Other parts of the user experience, like the control scheme and game mechanics, are examined briefly as well since they significantly affect layout decisions. They are also taken into account in the documentation when they are relevant to the visual design process. The character and environmental assets created for *Wizard Wars* will also be showcased, as they function as a base for the visual style and theme of the UI.

2 THE USER INTERFACE IN GAMES

In the context of game development, UI refers to anything that enables the player to interact with the game. It consists of the game's informational and interactive visuals and audios; and the control system, which in most cases is a game controller, a touchscreen, or a keyboard and mouse. (Novak, 2012, 235-248; Saunders & Novak, 2013, 3).

All mainstream games nowadays have a graphical user interface (GUI), which means the game is controlled through a menu-driven interface without the player needing to have any understanding of the underlying code. The player selects commands from a pre-determined set of options displayed visually on the screen in the form of buttons, lists, sliders and switches. The GUI in a game is further divided into menus, screens with options and information outside of gameplay; and the heads-up display (HUD), informational UI elements displayed on top of the in-game view during gameplay. (Llanos & Jørgensen, 2011; Saunders & Novak, 2013, 24, 78, 168).

Many elements that are typically not thought of as part of the UI – for example game characters and in-game objects – can also contribute toward the interaction between player and game, which makes the definition of a game UI relatively broad. Even though these in-game elements are not included in the scope of UI design, they have to be considered during the design process to decide which feedback and information is given through in-game means, and which ones need separate HUD elements. (Saunders & Novak, 2013, 3, 205-243).

Although game UIs share many traits with productivity application UIs, there is one significant difference; the need to create immersion. Games are designed to entertain instead of being used as tools to help do tasks more efficiently, so value is put into aesthetics in addition to usability, and often even allowed to interfere with it. This leads to UI visuals in games being more animated, detailed and art-heavy than those in productivity applications. (Pagulayan, Keeker, Wixon, Romero & Fuller, 2002, 884-888; Quintans, 2013; Stonehouse, 2010).

2.1 Goals and functions of the user interface

A game’s UI acts as a connection between the player and the game by providing feedback (output) and control (input) to the player, so he/she can learn new concepts and rules, perform actions to progress further in the game, and achieve goals defined by the game and its mechanics. Without this interaction between player and game, the game would be no more than a presentation, animation or static image – in other words, not a game at all. (Novak, 2012, 235-248; Saunders & Novak, 2013, 39-45).

Feedback

By providing feedback and information, the UI lets the player know what is going on in the game. This includes status like current health, available resources and location on the map; goals and the current progress toward achieving them; and reactivity by providing either immediate or delayed effects to player actions. (Saunders & Novak, 2013, 39-45, 205-243).

In the majority of games feedback is provided through visual means, by a combination of HUD elements and in-game characters, objects and effects. The ratio of HUD and in-game elements will vary depending on the game, and in rare cases all information is displayed in-game, removing the need for a HUD completely. Visuals are a reliable way to give feedback, as it is ensured the player will receive them if they are clear and exaggerated enough – after all, a game can’t be played with the display turned off. (Saunders & Novak, 2013, 39-45, Zammitto, 2008).

Although audio is an excellent way to provide feedback in an immersive, non-disruptive way, it can be unreliable. For example on mobile platforms games are often played in the middle of public with their sounds off, and some players’ schedules only allows them to play during the night when they have to keep the volume down. Audio serves as an excellent secondary way to give feedback though, and can add a lot to the atmosphere. A good example of this is horror games, where scares can be gotten from sudden sounds and eerie noises. (Saunders & Novak, 2013, 46-50; Wilson, 2006).

Control

For the player to be able to interact with the game, the UI has to give him/her control to perform the needed actions. Most of the control is provided through physical interfaces, for example a game controller or a keyboard and mouse. Less used controls and controls not needed or accessible during gameplay are provided through interactive UI elements in menu screens. Touchscreens form an interesting borderline case as they need on-screen assets for each interactive element, but on the other hand the whole screen area has the potential to be interactive, which enables more intuitive control schemes, for example swiping on the screen and tilting the device. (Saunders & Novak, 2013, 33-34, 45, 118-121, 167-172).

Controls should always provide some feedback when used to let the player know the game has received the command and is reacting to it in the expected way. For in-game controls, this feedback is shown on the affected characters and objects, and is often reinforced with audio. For example if the player presses the jump button, the player character jumps with a bouncing sound; and if the player lands a hit on an enemy character, the enemy staggers back and lets out a pained grunt. On-screen UI elements should also react to being clicked or tapped on by appearing to be pressed down or changing color or size, accompanied with a fitting sound. (Saunders & Novak, 2013, 45, 224).

Usability

The overall quality and effectiveness of a game UI, in other words how well feedback and control have been executed, is evaluated through usability. Usability is an important attribute in game UI design, as it gauges how efficient, intuitive and satisfying the game is to play. Is the game providing all the information the player needs at any given point in the game? How effortlessly can the player find information and give commands? Is it obvious to the player how they can proceed in the game and what can and cannot be done? Are there tedious and repetitive tasks that require multiple clicks or taps? These are the fundamental questions in game usability evaluation. (Laitinen, 2005; Quintans, 2013; Saunders & Novak, 2013, 55-56).

Usability should always be the primary concern when designing a game UI, as without a high level of usability the player will have a hard time enjoying the game, and in worst cases playing the game at all. It can be argued that "perfect" usability does not exist, as each player has an unique combination of previous experiences from other games, which will alter their expectations and preferences regarding the UI and its design. Even if the player has not played other games before, the UI of the game platform – usually a computer, console or mobile device – has influenced their opinions. (Morrison, 2013; Novak, 2012, 236, 259; Saunders & Novak, 2013, 55-56).

2.2 The value of a well-designed user interface

Up until recent years the UI was often an afterthought in game development. It did not receive much attention and no resources or time was set aside for designing it. Many older games have UIs that are visually clumsy, illogical and awkward to use. A poorly designed UI not only looks bad, it also takes usability and enjoyment away from the game. On the other hand a brilliant UI can significantly improve the gaming experience and raise the game above competing titles. In the current game market there are a few core reasons why a well-designed and visually pleasing UI is important, and why it plays a central role in the game's success. (Fox, 2004, 1-2; Glinert, 2009).

Firstly, the competition in the game market is fierce. With an over-abundance of titles competing for the player's attention, the game industry has reached a point where good just is not good enough. With the free-to-play business model where games can be played on web browsers or downloaded with the press of a button, first impressions are extremely important. Since the player has often not invested anything into the game before trying it out, the threshold to stop playing if the game is not meeting expectations is non-existent. The decision to continue playing or not is made in a few minutes, and even then the average playtime before dropping the game completely is barely an hour. In other words it is extremely difficult to create player engagement in the current game market. By clearly and quickly communicating how the game is played and how to proceed, as well as setting intriguing goals, the UI is essential for trying to hook the player. (Carmichael, 2012; Glinert, 2009; Laitinen, 2005).

Secondly, a big portion of the current player-base is casual, with limited knowledge about common game concepts and mechanics which are taken for granted by long-time players. Since gaming is a voluntary activity meant to bring entertainment, players have a hard time tolerating games that make them feel stupid or frustrated by not properly teaching them how to play it. This does not mean a game cannot be difficult, as often the challenge and sense of accomplishment from overcoming it adds to the entertainment value. However, the difficulty should come from the game mechanics, not from unresponsive controls or from confusing and misleading feedback. If the player struggles with giving commands and understanding why they are losing or cannot progress in the game, the game is perceived as unfair and inaccessible. (Laitinen, 2005; Pagulayan et al., 2002; Saunders & Novak, 2013, 167-191, 195).

Lastly, visual impressiveness is valued highly as a marketing point, and is part of the player's expectations for today's games. It is also one of the main things the player pays attention to when evaluating a game and its entertainment value. It has been observed through the *aesthetic-usability effect* that visually pleasing UI designs are perceived as being easier to use than less-pleasing ones – regardless if they are or not. Aesthetic designs help create positive attitude towards a game, encourages creative thinking and problem solving, and makes players more tolerant of design problems. Therefore, aesthetics can up to a point serve as a "safety cushion" for usability. It won't hide or make players oblivious to any glaring flaws, but it will make players more willing to interact with the UI and figure things out on their own. (Glinert, 2009; Lidwell, Holden & Butler, 2010, 20).

3 VISUAL DESIGN PRINCIPLES

In game UIs visual design is tied closely to usability by making it easier to find and categorize information, and making the game more aesthetically pleasing and efficient to play. There are a couple of significant differences between a game and a productivity application that needs to be taken into account during the UI design process. The first is that games want to create immersion, to make the player forget they are just playing a game. With UI elements this can be done by making them fit into the game's atmosphere through visual style, colors and typography. Games also define their own goals as opposed to productivity applications, where the goals are defined by the user. Since the player is not automatically aware of a game's goals they have to be communicated to him/her through the UI, for example by showing resources and items to gather, and skills and game levels to unlock. (Pagulayan et al., 2002, 884-888; Saunders & Novak, 2013, 55-56, 64-68, 216-221, 233, 238).

A game UI's purpose of providing clear feedback and information, as well as immersion, creates some unique challenges to the visual aspect of the UI design. It is possible to make a perfectly functioning UI with very simple shapes, texts and icons, while only focusing on optimizing the layout and usability. However, to make the UI visually pleasing and adding to the game's atmosphere, it should use the same theme and colors set by the rest of the visuals. Usability should always come first though, so visual impressiveness can only be pushed until it starts interfering with the readability off the UI elements. (Fox, 2004, 27-41; Saunders & Novak, 2013, 64-68, 238-241).

Depending on the game genre and gameplay the requirements for the UI differs greatly. This is due to variations in the amount and type of onscreen information and controls needing to be displayed. No matter how simple or complex the UI is though, it should follow basic usability and visual design principles. Applying these proven principles successfully will go a long way towards creating an aesthetic and coherent UI with a high level of usability. (Fox, 2004, 43-44; Novak, 2012, 256-259).

3.1 Layout and visual organization

Layout refers to how the various UI elements are arranged on the screen, both in relation to the screen and to each other. Aligning UI elements and grouping them with colors, shapes and positioning according to their function creates visual organization. With a visually organized UI the information is faster to find and categorize, controls and commands are easier to learn and memorize, and the layout is more visually pleasing. (Fox, 2004, 53-54).

Alignment

To make the UI appear structured, elements on the screen should be aligned to each other as much as possible. Alignment is one of the most powerful tools for creating an UI that is easy to navigate and find information in. Especially menu screens with repeating UI elements such as inventories and achievement lists make extensive use of alignment. The more complex the design and the information needed to be shown is, the more essential a good alignment grid becomes. (Cooper, Reimann & Cronin, 2007, 296-299).

In the majority of cases UI elements are aligned either by their edges or centers along a vertical and/or horizontal line. More complex forms of alignment are also possible, for example diagonal, circular and spiral, although these usually require the alignment path to be shown to be recognizable. Therefore, they are unpractical for displaying a large amount of information, but can be effective in the main menu to add visual interest. (Cooper et al., 2007, 296-299; Lidwell et al., 2010, 24).

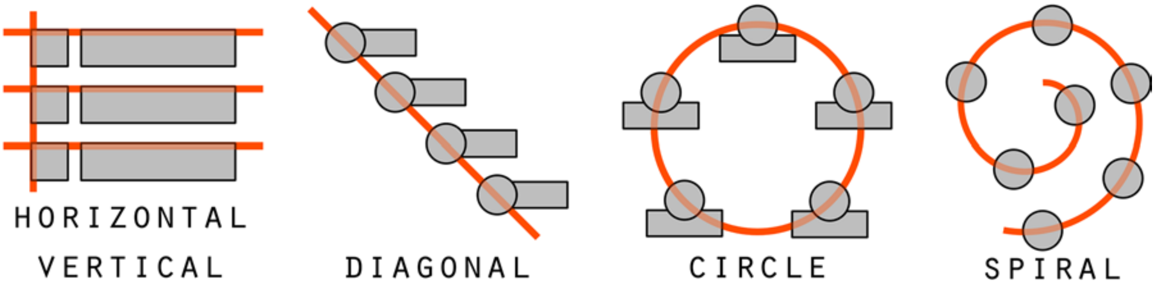


Figure 1. Examples of alignments.

Overlaying is another form of alignment where elements partly intersect with each other, offset by a pre-defined amount. The aligning can be cumulative with more than just two elements stacking up, in which case the offset pattern should be the same between elements, resulting in a cascading effect. This method is especially useful when looking for ways to save screen estate. For example in card game UIs the cards often overlay each other, as the player gets all the needed information by seeing just a part of the card.

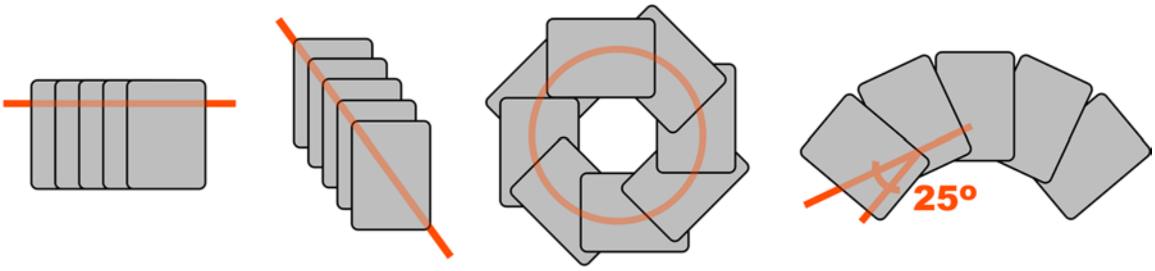


Figure 2. Examples of alignments utilizing overlay.

In text alignment will significantly improve the readability, especially in bigger paragraphs. Left-aligned and justified (aligned to both left and right) text is most commonly used as it is easy for the eyes to jump from one line to the next. The column created by the edge of the text also serves as a visual cue for aligning other UI elements. For titles or smaller quantities of text, for example game options on the main menu, center-aligned text is feasible too. (Galitz, 2007, 578-580; Lidwell et al., 2010, 24).

Grouping

UI elements with shared attributes – shape, size, color and/or proximity to each other – are perceived by the player as being connected. When looking at a game's in-game view or menu screens, the player will subconsciously try to sort both the information and input elements into logical groups. By giving the player clear visual cues on how to group these elements, their ability to understand what is going on in the game and what they can do in each screen increases significantly. (Galitz, 2007, 153-157; Lidwell et al., 2010, 226; Morrison, 2013).

Some grouping methods give much stronger cues than others, as can be seen in Figure 3. Color is by far the strongest, providing the colors used in the UI design are limited. As the amount of colors increase, their effectiveness as visual cues decrease. Color is very useful for situations where two or more elements have a similar function but cannot be placed near each other. Proximity is also very strong, especially if the elements all have the same shape. By placing the elements close together they are regarded as a single, bigger element. Size is only effective when the difference is big enough, and when the elements are limited to only two different sizes. Shape is the weakest cue, and should only be used together with another grouping method. (Galitz, 2007, 153-157, Lidwell et al., 2010, 226).

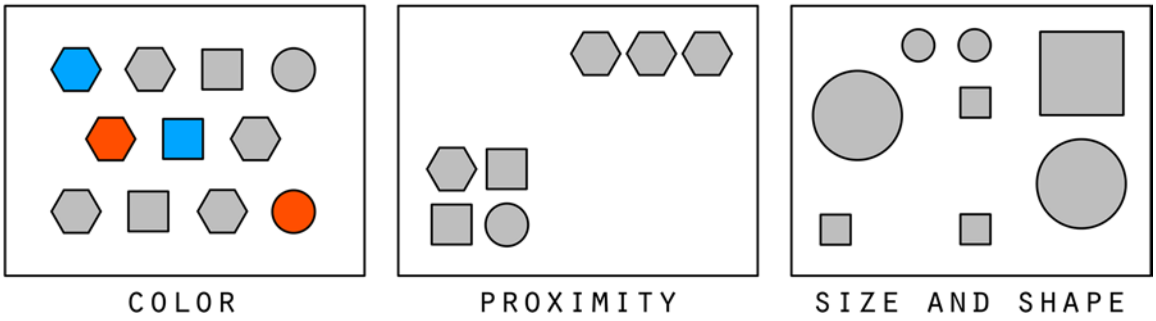


Figure 3. Grouping methods.

Grouping UI elements properly is a very potent way to increase the usability of the UI, and should be strived for especially in the more complex, menu-driven ones. Each group can further be labelled with appropriate titles, such as spells, items and options, to make it easy to refer to them. To optimize the effectiveness of the grouping methods the UI should be as uniform as possible with the minimal amount of different colors and shapes, so both the differences and similarities of the UI elements can be noticed with just a glance. (Galitz, 2007, 153; Lidwell et al., 2010, 226; Morrison, 2013).

Consistency

A game UI should be consistent both in its layout and visual style, meaning elements providing similar information or functions should both look the same and be placed in the same position in each menu screen. This improves usability by making the UI easier to learn, as button functions and visual cues can be recognized between screens. It also makes the UI faster to navigate as the elements are always in the same position, so the player can operate on muscle memory instead of having to spend time locating buttons. For example in Figure 4 the darker grey button is easy to find in every screen, as both its form and position stays the same regardless of what other UI elements are displayed. (Galitz, 2007, 133-134; Lidwell et al., 2010, 56; Saunders & Novak, 2013, 228).

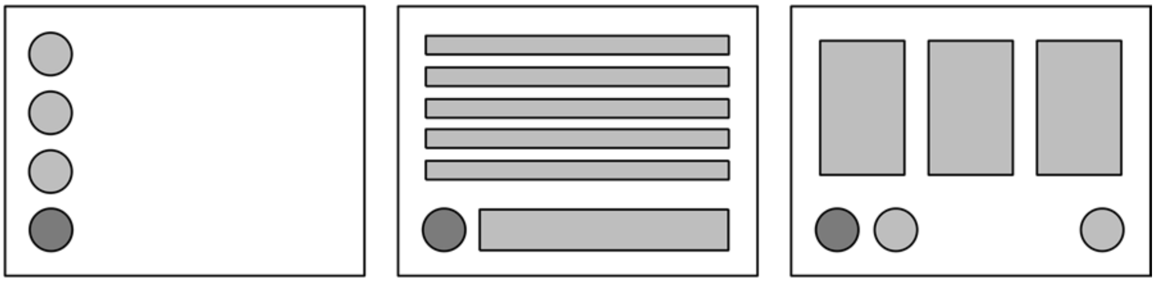


Figure 4. The darker grey button keeps the same form and position between screens.

The UI elements found in the in-game view need to be consistent as well, to give the player reliable information on what is going on in the game world and what they can do next. All push-able switches should look the same, and if a vine plant on the wall can be climbed all vines should be climbable. If the game is able to teach the player to recognize certain interactive in-game objects and their functions, these objects can later be used as motivators and mini-goals. For example if the player knows collecting heart-shaped containers will raise their max health, seeing one on top of a cliff will automatically make climbing said cliff to reach the container a new goal in the game. (Saunders & Novak, 2013, 228).

In a consistent UI design, inconsistency can be used to great effect to draw attention to a particularly important UI element on the screen. If all the buttons are blue, making one orange will automatically cause it to get the player's focus. Inconsistency should be used very sparingly though, as each one will diminish the attention-grabbing effect, and too many will just turn the design into an actually inconsistent one. (Cao, Zieba, Stryjewski, & Ellis, 2015, 25-27; Galitz, 2007, 133-134).

Standards

Most players have previous experiences from other games which will influence their preferences and expectations when it comes to the UI. Games from the same genre and with similar gameplay are expected to have a similar UI layout and control scheme. Many game genres have well-established standards that are in use not only because players have gotten used to them, but because they are actually good and functional. (Fox, 2004, 35; Saunders & Novak, 2013, 126-127, 163).

There are many advantages with using already standardized design patterns. It is faster and easier for the player to learn and memorialize an UI with familiar features, and it reduces the amount of guidance needed when starting to play the game. When the UI looks familiar and functions as expected the game also feels approachable, encouraging the player to try out all the options on the screen. (Cooper et al., 2007, 317-320; Saunders & Novak, 2013, 126-127, 163).

For these reasons standards should always be used when feasible. However, as no two games are the same, there are usually at least small deviations from the standards to accommodate new game mechanics. The more unique a game is, the more there is need to be innovative with the UI – both from a visual design and control scheme standpoint. (Cooper et al., 2007, 317-320; Saunders & Novak, 2013, 126-127, 163).

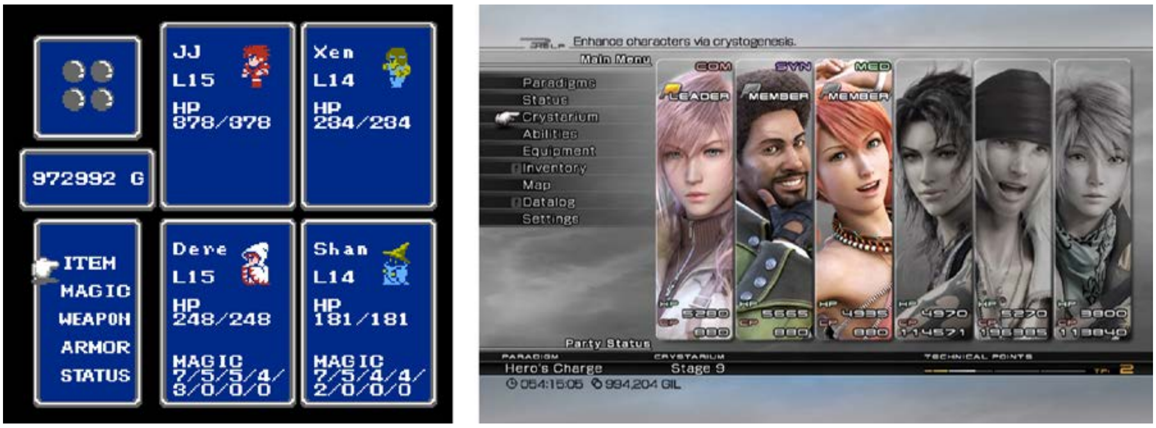


Figure 5. Menu screens from *Final Fantasy* (1987) and *Final Fantasy XIII* (2010).

The screenshots in Figure 5 show two menu screens from the *Final Fantasy* series, a Japanese role-playing game which utilizes menu screens for many commands and options affecting the gameplay, for example item lists, character customization, and choosing the active party members for battles. The UI has gotten more visually impressive as new additions to the series come out, but the basic functionality and information shown has stayed mostly the same.

3.2 Color Theory

Good use of colors has many advantages in UI design. When used correctly color is one of the most effective ways to both group objects and information together and differentiate them. For example if all buttons are in the same color they are easy to recognize on the screen after that color has been associated with buttons; and health and mana bars can be told apart by having them in red and blue, respectively. Color also has a high attention-grabbing capacity, able to make important UI element pop out from the screen and thus making them quick and easy to find. (Galitz, 2007, 691-692, 695-696, 700-701, 704; Saunders & Novak, 2013, 233-234).

As much as color can improve usability it also has the potential to significantly weaken it. Because color has such strong grouping, differentiation, and attention-grabbing properties, it causes confusion and slows down the player's response time when used incorrectly. Therefore, making a UI too colorful is much more risky than making it too monochrome. There is not much use for colors in simple UI designs, but as the complexity of the UI increases, so does the value of effective color usage. (Galitz, 2007, 691-692, 695-696, 700-701, 704; Saunders & Novak, 2013, 233).

Color is also a great tool for creating atmosphere and mood. As a game's UI aims to keep to the aesthetics and atmosphere of the game, the color palette or at least the main color of the UI is usually taken either from the game's theme or gameplay. Colors also have an "in-built" mood to them in the way they are perceived. For example red and orange hues are warm, urgent and dangerous; blue hues are cold and calming; and green hues are natural and refreshing. The saturation level also matters, with high saturation and pure hues perceived as childish and casual; and desaturated colors as more sophisticated and serious. (Fox, 2004, 29-31, 44-45, 49; Saunders & Novak, 2013, 233).

Choosing colors

The main challenge when deciding on the color scheme for an UI is to find colors that look good together. A good color scheme uses a very limited palette, as each color added to the design will lessen the impact a color can have in both conveying information and getting attention. It is recommended that no more than five colors should be used, although the fewer the better. The more complicated a menu screens is, the more colors it tends to need. (Cooper et al., 2007, 292, 311; Galitz, 2007, 700-701, 704; Saunders & Novak, 2013, 233-234).

The color wheel shown in Figure 6 is a visual design diagram that divides the color spectrum into twelve basic hues for easier visualization. It can be used as a starting point for finding harmonious two-, three or four-color combinations based on proven formulas. All the possible color combinations can be seen by rotating the formulas inside the color wheel. (Fox, 2004, 45-46; Williams, n.d.).

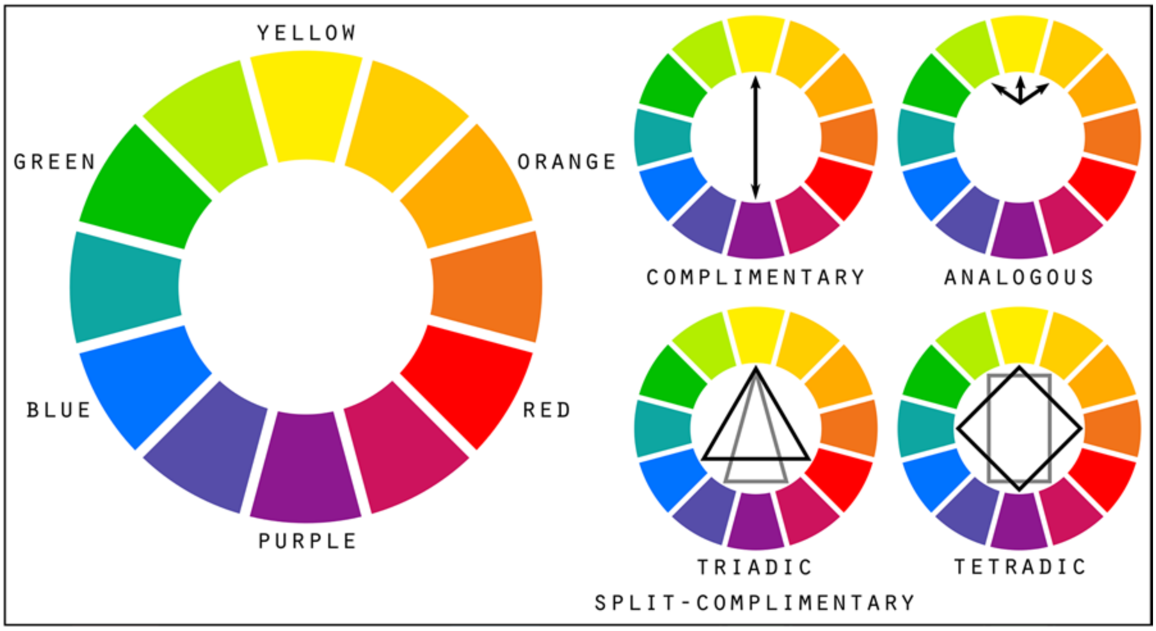


Figure 6. The color wheel and the formulas for finding harmonious color combinations.

The hues are very rarely used in their pure forms except in small amounts to grab attention, for example in button assets. Colors are altered by desaturating them and making them either lighter or darker. Moderately and highly desaturated colors make for more appealing color combinations, as they aren't as tiring for the eyes to look at. As a general rule the interactive elements should be the ones with the highest saturation, the foreground elements should have a medium saturation, and the background a low saturation. In addition to hue, UI elements should also have variations in their brightness to be easier to tell apart from each other. (Cooper et al., 2007, 311-312; Galitz, 2007, 694, 708-709; Lidwell et al., 2010, 48, Williams, n.d.).

Considerations when using colors

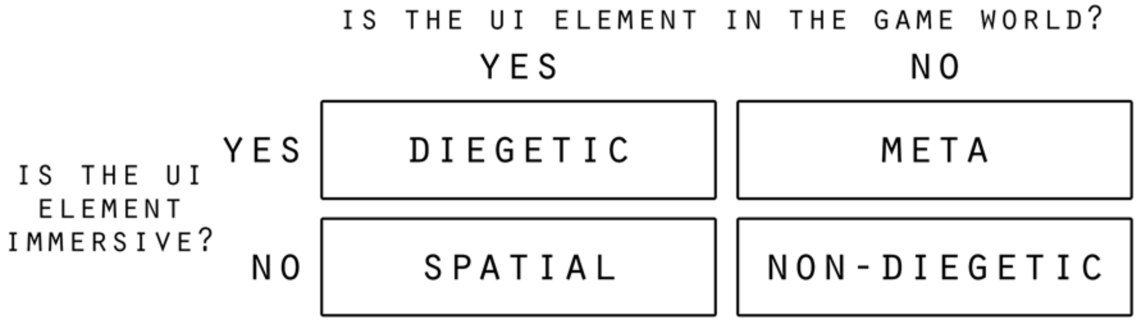
Color-viewing deficiencies are common enough that color should never be used as the only differentiation between two UI elements or to show status changes within an element. For example a status change from normal to poisoned cannot be shown only with a color change to the health bar, there also needs to be an additional indicator like a status icon or a noticeable change in the affected game character. The most common deficiency is the inability to separate red from green, and although total color blindness does exist it is extremely rare. People suffering from color-viewing deficiencies can still recognize differences in brightness as long as they are extreme enough, so a good way to check if a screen is still eligible for them is to turn it into grayscale. (Cooper et al., 2007, 312; Galitz, 2007, 697-698, 703, 713-714; Saunders & Novak, 2013, 58-59).

It is also useful to be aware of the cultural differences in color associations, and how they can either help or harm the usability. These color meanings are usually tied to emotions or basic moral and social concepts. Although these cultural differences are relevant, they are hard to utilize in actual UI design unless the target audience is narrowed down to a specific country or cultural group. The social concepts, such as green means go and red means stop, can be more widely utilized even though they are not completely global. (Galitz, 2007, 698-700, 714).

3.3 Diegesis theory – merging the HUD with the game world

Traditional HUD elements such as bars, numbers, texts and icons are effective at giving feedback and information to the player, but they break the immersion by adding an artificial layer on top of the game world. Making the HUD immersive is difficult as it is displayed during the actual gameplay, when clear feedback and information, as well as immersion, are crucial. For this reason the goal in most cases becomes to design a HUD that breaks the immersion as little as possible. (Saunders & Novak, 2013, 64-68, 238-241; Wilson, 2006).

It is generally agreed upon that the best HUD is one that is not noticed at all – it displays the needed information and then gets out of the way. A way to reduce UI elements in the HUD or in some rare cases even completely remove the need for one, is to display information inside of the game world and/or on game characters. The diegesis theory helps analyze the UI elements by dividing them into four categories: diegetic; spatial; meta; and non-diegetic, depending if they exist within the game world or not and if they are immersive or not. (Russell, 2011; Saunders & Novak, 2013, 64-68, 238-241; Zammitto, 2008).



		IS THE UI ELEMENT IN THE GAME WORLD?	
		YES	NO
IS THE UI ELEMENT IMMERSIVE?	YES	DIEGETIC	META
	NO	SPATIAL	NON-DIEGETIC

Figure 7. The four UI element categories in the diegesis theory.

Diegetic

A diegetic UI element is both immersive and integrated into the game world, meaning it is visible to both the player and the characters in the game. Since the game characters can see the element too they are able to react and interact with it in a natural and immersive way. Visible changes to a character's armor and weapons depending on what is equipped; a building or vehicle breaking down gradually as it takes damage; an item that can be picked up or interacted with in some other way; and a crossroad sign in the game world showing where each road leads, are examples of displaying information to the player through diegetic UI elements. (Russell, 2011; Saunders & Novak, 2013, 78-81, 238-241).

Spatial

A spatial UI element exists within the game world but does not belong there, which means it is not immersive. The game characters are unaware of the element and as a result they cannot react to or interact with it. Examples of spatial elements are a unit selection marker in a strategy game; a quest icon over a NPC character's head; and an outline for a character taking cover in a first person shooter. (Russell, 2011; Saunders & Novak, 2013, 78-81).

Meta

A meta UI element does not exist in the game world but is immersive by maintaining the game's narrative through visuals and functionality. The most common way to achieve this is by breaking the fourth wall, by adding overlays that look like they come from inside the game world on top of the in-game view. Games with a first-person view are the main group using these kinds of meta elements. As the player character is not visible on the screen the meta elements replace the character sprite or model as a way to provide feedback. Blood splattering on the screen when the player character takes damage; and the screen cracking to show a helmet or visor is about to break, are commonly used meta elements. (Saunders & Novak, 2013, 78-81; Stonehouse, 2010).

An UI element can also be classified as meta by mimicking real-world items and behaviors, and/or by including objects present in the game world. Many minigame interfaces are an example of this, resembling an object or composition seen in the main gameplay – often the one the player has to interact with to start the minigame. In addition to the visuals, the control scheme can also add to a meta element's immersion. For example instead of typing the numbers with the keyboard in a mobile phone UI element, the player hovers the mouse over the phone buttons and clicks on them, mimicking the way they would use a real mobile phone. (Saunders & Novak, 2013, 78-81, 238-241; Stonehouse, 2010).

Non-Diegetic

A non-diegetic UI element is neither immersive nor integrated into the game world. It is displayed on top of the in-game view as an overlay, usually in a static position. Most traditional UI elements like health bars, score counters and equipment icons are non-diegetic. Although non-diegetic elements are mostly non-interactive there are exceptions, for example a skill button in a role-playing game that is clicked on to activate the skill; and a mini-map in a strategy game that is clicked on to switch the view to the corresponding area on the in-game map. (Saunders & Novak, 2013, 78-81; Stonehouse, 2010).

Overall, UI design methods and trends in the game industry are constantly developing to be more and more diegetic. With technology that enables realistic graphics, the wish to give the player a cinema-like experience by making the UI as immersive and non-disruptive as possible is understandable. It can be argued that the more realistic graphics, the more immersion is desired in a game. In some game genres and platforms, immersion is not as big a consideration. For example mobile games, where the average playtime is five to fifteen minutes and the venue of play are coffee breaks and public transports, it is futile to try to get the player immersed. In these games usability is held at an even higher value, both from a visual quality and feedback clearness standpoint. (Saunders & Novak, 2013, 64-81; Wilson, 2006).

4 SHOWCASING WIZARD WARS

One of *Wizard Wars'* original design goals was to recreate the gaming experience found in many older PC strategy games – most notably *Age of Wonders: Shadow Magic* and *Age of Empires II* – for a tablet platform. The mix of resource management; troops and buildings development; and exploration of a huge random-generated map makes for an intriguing and rewarding experience. However, it was discovered early on that major changes were needed to the initial game design to make the game more suitable and enjoyable to play on a tablet.

Creating a functional touchscreen control scheme for the initial game design proved to be impossible. Strategy games are traditionally played on PC with a keyboard and mouse – a physical interface that lends itself to very complex control schemes. Unfortunately the touchscreen, although intuitive, is quite limited in input variations. Controls such as the mouse's right-click and scroll wheel, mouse-overs, keyboard shortcuts, and the SHIFT and CONTROL modifiers are hard to replicate. And most PC strategy games use all of these! Prototypes which utilized the iPad's multi-touch gestures were tested, but none of them were precise enough. Using them would have made the game much less casual-friendly and easy to get into, as the average iPad user is not familiar with most of the gestures. Also, this choice would have practically limited the game to only iOS tablets, as Android and Windows devices would not have supported all of the gestures.

Therefore, it was instead opted to simplify many aspects of the gameplay. Eight troops per player decreased to one wizard; a map consisting of 101 tiles went down to 19 (although with explored tiles getting replaced with new ones); wizard and monster stats went from health, mana, attack, defense, magic and magic defense to just health and mana. Not only did this simplify the control scheme, it also significantly cut down on the needed menu screens and art assets.

The final game concept and the visual style of the character and environmental assets will be showcased next, to give better context to the visual design process of the UI discussed in the next chapter.

4.1 Overview of the game

Wizard Wars is a turn-based strategy game developed for tablet platforms, with the first version being available only on iPad. The game can be played either multiplayer online between three players, or single player against two AI opponents. The main target audience is small social circles who want to play a game with a tabletop feel to it without the players having to be in the same room. It can also be enjoyed by any player who likes tabletop, multiplayer and/or strategy games. As the game has some amount of complexity and the player needs to be able to read to enjoy it, it is not suited for younger children.

Gameplay

The player takes on the role of a wizard who is racing against opposing wizards to recover a powerful magical crystal that has fallen from the sky. The task is not an easy one. The fallen crystal's magic is constantly warping its surroundings, transforming the terrain from lush forests into barren wastelands and back again. Its powers have also attracted many monsters and magical creatures, not to mention the other wizards and their powerful spells! Thankfully the area is also filled with old artifacts and powerful items to help the wizard get stronger and be able to face the challenges ahead.

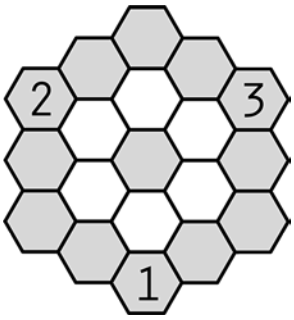


Figure 8. A wireframe of the game map with the three wizards' starting positions marked.

The gameplay takes place on a map consisting of 19 hexagon-shaped tiles forming two rings around the central tile, as can be seen in Figure 8. Each player starts the game on a predetermined tile on the outer ring from which they begin exploring the map. By default they can explore one tile per turn. Each tile is occupied by a monster which the wizard has to battle; a chest with items or resources to pick up; or an event that can be either accepted or declined. The majority of the tiles are monster tiles, as they are the most interactive ones and provide players with crystal shards, an essential resource in the game. In the case of two wizards exploring the same tile, they first have to fight each other, after which the loser is pushed back to their previous tile and the winner gets to explore the tile as normal. The contents of a tile is hidden until a wizard explores it, after which it becomes visible to all wizards. If the wizard who explored the tile fails to or chooses not to interact with it – for example doesn't manage to defeat the monster or doesn't want to complete a trade offered by an event – another wizard is free to use the tile, providing they can make it there in time…

A few turns after a tile has been explored it will crumble and be replaced with a new one. This will be indicated a turn before it happens, so players know to evacuate the tile. Even if a wizard happens to be on a tile when it crumbles they won't be lost forever, instead reappearing on the new tile. The wizard will lose a turn though, understandably being dazed after their experience of falling into a swirling vortex. The tiles will get harder over time to provide new challenges and better items as the wizards grow stronger.

The game is won by exploring the tile containing the fallen crystal and defeating the powerful monster guarding it. Sometimes the tile containing the fallen crystal will be among the starting tiles and sometimes it will not appear until later. This helps make the game experience more varied from game to game. At times the crystal's location will be known from an early stage – even if it is explored it will not crumble away like normal tiles – turning the game into a race of who can get their wizard strong enough to kill the crystal guardian first. It is also much easier for players to stop each other from reaching the crystal, as they know which tiles and wizards to bombard with their disruptive spells. At other times the game will progress to a stage where everyone is strong enough to beat the guardian, but are racing to find the right tile. In total there are sixty tiles in each game of Wizard Wars, so the crystal is going to appear with certainty before long. After all sixty tiles have appeared on the map, the tiles will stop crumbling away, further hastening the finding of the crystal.

The player has three resources to manage: health, mana and crystal shards. Health helps the wizard endure battles against monsters and other wizards. If a wizard's health drops to zero, they will collapse and need a turn to recover. Letting this happen will waste valuable turns, so the player has to keep an eye on their wizard's health and judge whether they will be able to survive a potential battle or not. Mana lets the wizard cast spells, many of which have significant effects on the gameplay and other players. Crystal shards are used primarily to upgrade spells and unlock new ones, but many events also allow the player to trade in crystal shards to either increase other resources' max values or obtain powerful items.

Crystal shards have to be collected by exploring tiles and defeating monsters, while health and mana regenerate a certain amount each turn, determined by the resource's max value. This gives incentive for players to raise their max values, sometimes even offering up some of their precious crystal shards to achieve it. Maintaining a good balance between all three resources and using spells wisely is the key to victory.

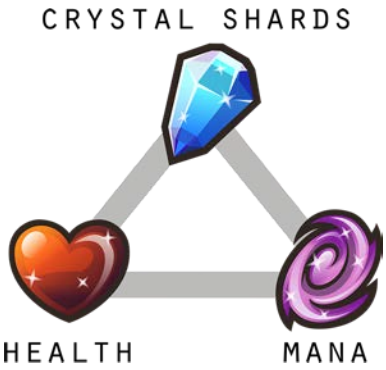


Figure 9. The player has to manage and maintain a balance between the three resources.

The main way to interact with other players in *Wizards Wars* is by casting spells. Each wizard has five spells suited for their playstyle, three of which can be used during regular gameplay and two which are available only in battles. The spell effects vary a lot, with some targeting one or multiple tiles, some opposing wizards and monsters, and some the casting wizard themselves. Each spell can be upgraded twice with a set amount of crystal shards. Two of the more powerful spells (one normal and one battle spell) are also locked at the beginning of the game, and are unlocked by upgrading other spells to max level. There is no limit to how many spells can be cast during a turn or battle, providing the wizard has the required mana for it.

To reduce the amount of idle time for players, *Wizard Wars* uses a turn-based simultaneous execution model. This means all players take their turns and input their commands at the same time. The turn timer is set to 150 seconds (2 minutes and 30 seconds) in multiplayer, although in testing the majority of players finished their turns in between 40 and 90 seconds. A player can end their turn before the 150 seconds are up, and if all players opt to do so, the turn ends prematurely. The timer is there mostly as a safety measure to stop players from ruining the gaming experience by refusing to end their turns. In single player there is no turn timer, so the turn has to always be ended manually. The AI wizards are set to finish their turns in 20 seconds, so they won't slow the player down. In preliminary testing a game usually lasted between 15 to 20 minutes, depending how quickly players completed their turns.

Progress and monetization

By completing a game of *Wizard Wars* the player earns coins, the amount depending on if they won the game or not. These coins are used as in-game currency to buy new tiles from the game's collection. The bought tiles will appear in future games, adding more variation to the gameplay with challenging monsters, special events, and items with unique abilities. In multiplayer games only one of the players need to have the tile unlocked for it to have a chance to appear, giving the other players a preview of it and perhaps prompting them to buy the tile if they liked it. In addition to the tiles, one new wizard can also be unlocked with a huge amount of coins. There are also plans to include a special tile as an extra reward now and then for winning a multiplayer game, for example during holidays such as Christmas and Halloween.

Wizard Wars is at the moment planned to be free-to-play, with themed tile-sets that can only be bought with micro-transactions. The price hasn't been decided on yet, but I would expect for it to be between 0,99–2,99 euros. Each tile-set would contain new tile graphics appropriate to the theme of the set (lava pit, snowy mountains etc.) and 8 new tiles (4 monsters and 4 event/items). A few of the wizards are also planned to be purchasable with real money, for a similar price as the tile-sets.

4.2 Character and environment visuals

The only requirement for the visual style in *Wizard Wars* was for it to be set in a traditional high-fantasy world – the natural place to find wizards, monsters and magic artifacts. As the game is developed for tablet platforms for a more casual audience, a colorful, stylized approach was chosen to make the game feel both accessible and visually interesting. The assets were done in vector graphics to give them a clean, glossy feel.

Characters

After the game design was changed from having multiple troops into a single wizard per player, it opened up resources to focus on each character more, and to really differentiate them from each other not only with colors but through poses and animations as well. The characters seen in Figure 10 were the first assets finalized for the game, and the ones that set the colorful, somewhat "bulky" visual style, with exaggerated heads, hands and feet and clearly defined shadows. Especially the bulkiness and shadowing style later translated over to the UI elements.

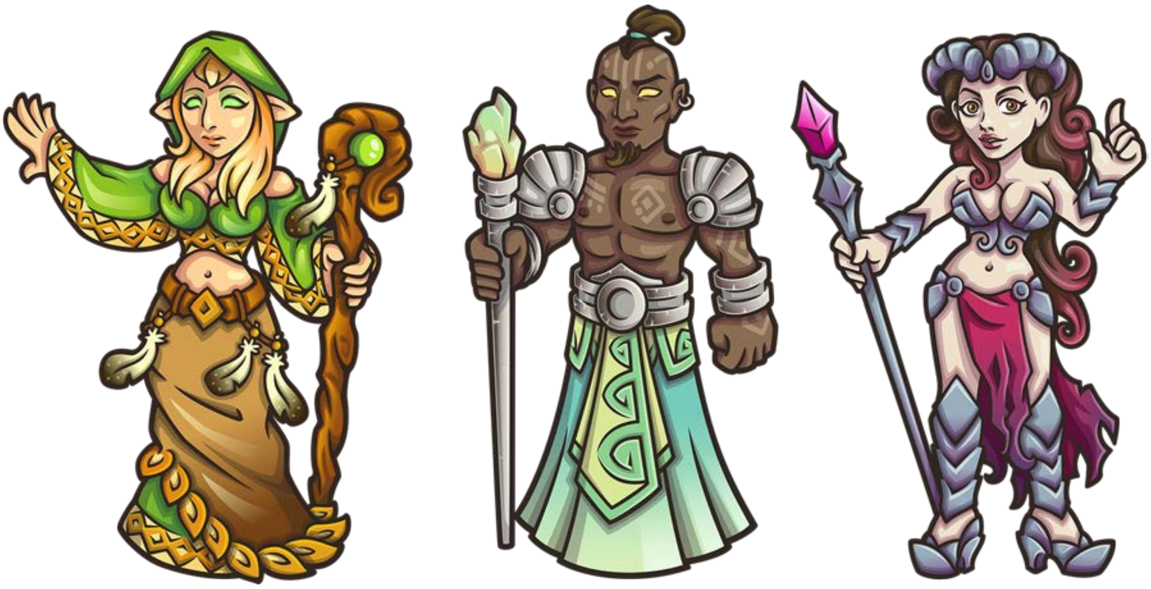


Figure 10. The Druid, Djinn and Sorceress; three of the wizards in *Wizard Wars*.

In-game assets

Wizard Wars' fantasy setting was able to fully utilize the colorful palette with its magical creatures, mystic forests and arcane wastelands. The wizards as well as the monsters and structures translated well into smaller, isometric sprites. The tiles were harder as they needed to be left relatively free in the middle so characters and structures could stand on them without looking out of place. There also needed to be some variation in the tiles, as it would have looked strange if the map was made up of only two repeating assets. Therefore, three variations of each theme of tiles were created. As exploration is a central part of the gameplay the player should feel excited about exploring new tiles on the map, not only for the content to be found in them but from a visual standpoint as well.

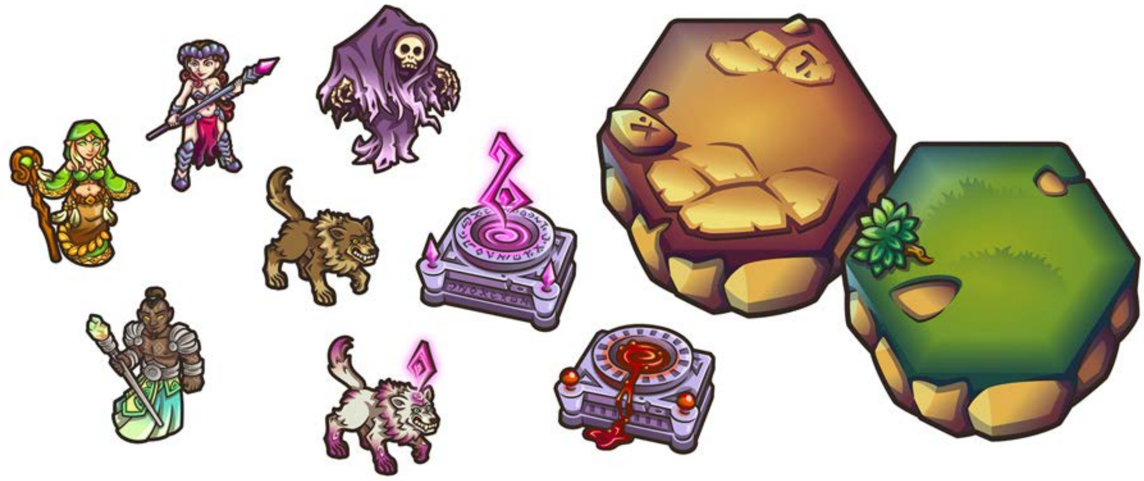


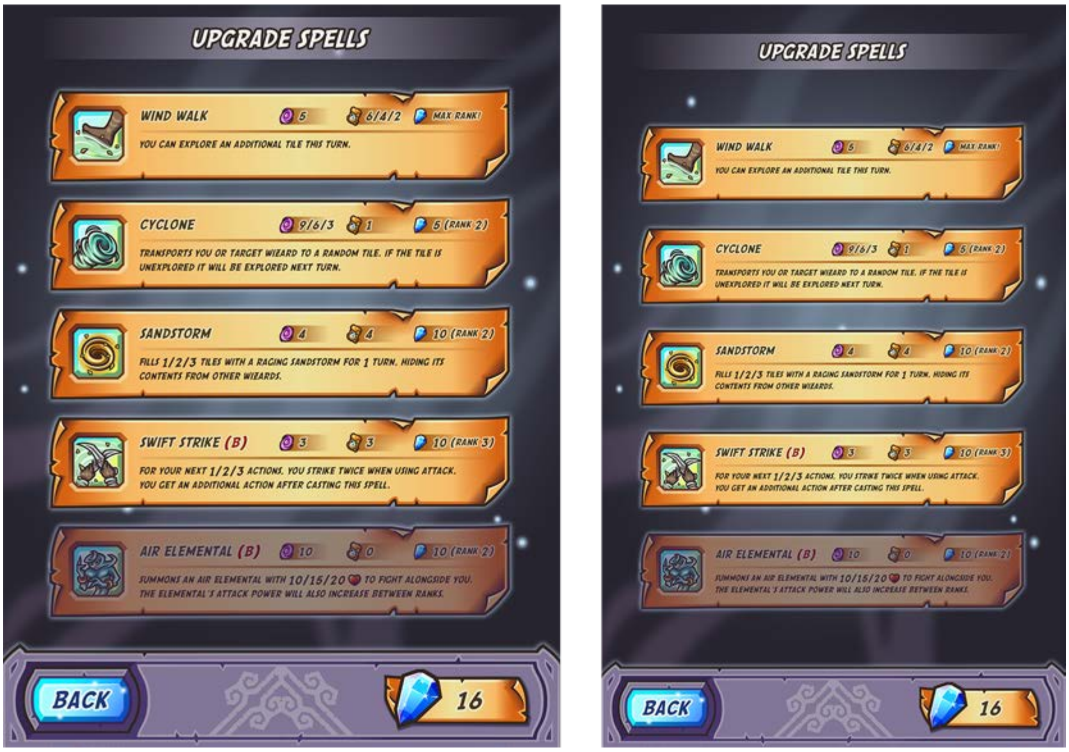
Figure 11. A sample of character and environmental assets.

5 DESIGNING THE USER INTERFACE

From a technical standpoint the only real restriction when designing UI visuals for modern tablets is the physical screen estate. The devices' hardware and screen pixel densities have reached a point where artists do not have to worry about their art not looking sharp enough or animating smoothly enough. The game file size is worth some consideration too, although it is not as critical on tablets as it is on mobile phones. Many popular games have file sizes well over 200 MB with some rare ones even breaking the 1 GB mark, and getting to these numbers require a significantly bigger game than *Wizard Wars* is planned to be.

The UI art assets were still wanted to be kept as light as possible to optimize performance for older devices. Small games are also faster to download and do not take up as much storage space on the device – qualities that are always appreciated by the player. Another goal was to keep the UI easily modifiable into different screen aspect ratios, as there are many tablet models with much wider screens than the iPad's 4:3 aspect ratio.

With these things in mind a modular UI design method was chosen, where assets can be re-used across multiple screens and combined in different ways to create a variety of bigger UI elements. This method is useful even if size is not a consideration, as it brings unity and consistency to the design. Also, as a result of the modularity, the second goal of making the UI transferrable between screen aspect ratios was achieved almost automatically, as can be seen in Figure 12. By having the majority of the UI elements "floating" on the screen over a simple background, instead of aligning them with a themed background image, the different aspect ratios could be achieved by simply changing the distance between elements. Admittedly the second method has the potential to create more immersion and be more visually impressive, but it would have been significantly more work- and art-heavy.



WIND WALK 5 6/4/2 MAX RANK!
YOU CAN EXPLORE AN ADDITIONAL TILE THIS TURN.

CYCLONE 9/6/3 1 5 (RANK 2)
TRANSPORTS YOU OR TARGET WIZARD TO A RANDOM TILE. IF THE TILE IS UNEXPLORED IT WILL BE EXPLORED NEXT TURN.

SANDSTORM 4 4 10 (RANK 2)
FILLS 1/2/3 TILES WITH A RAGING SANDSTORM FOR 1 TURN, HIDING ITS CONTENTS FROM OTHER WIZARDS.

SWIFT STRIKE (B) 3 3 10 (RANK 3)
FOR YOUR NEXT 1/2/3 ACTIONS, YOU STRIKE TWICE WHEN USING ATTACK. YOU GET AN ADDITIONAL ACTION AFTER CASTING THIS SPELL.

AIR ELEMENTAL (B) 10 0 10 (RANK 2)
SUMMONS AN AIR ELEMENTAL WITH 10/15/20 ❤ TO FIGHT ALONGSIDE YOU. THE ELEMENTAL'S ATTACK POWER WILL ALSO INCREASE BETWEEN RANKS.

Figure 12. A *Wizard Wars* menu screen in 4:3 (left) and 16:9 (right) aspect ratios.

To help figure out what assets were needed, the UI design process was started by drawing a flowchart. It defined all the menu screens in the game, what information and controls each screen would contain, and how the screens connect to each other. Then all the pop-ups were added and filled with the same details. The full flowchart for the UI of *Wizard Wars* can be found in the Appendices.

With the flowchart done it was possible to get a good picture of the things that would affect the layout and visuals. The number of buttons in each screen and popup, the amount and length of the texts, and the type of information displayed (resource, description or command input) proved to be the most essential. If the same information or type of information was shown in more than one screen, the placement should be the same across screens.

The following UI design documentation first goes over the establishing of the visual style and creation of the art assets, before moving on to the design and layout choices for the in-game view and each menu screen. As iPad is the primary development platform for *Wizard Wars*, all the screen mockups will be displayed in a 4:3 screen aspect ratio.

5.1 Setting the visual style

As the character and environmental assets are made in vectors, the choice to create the UI design in vectors too felt natural. The same clean and glossy visual style with strong outlines, solid colors and sparingly placed gradients would be easy to reproduce. Vector graphics also have one notable advantage to raster graphics: the ability to change an asset's size without quality loss. Devices with larger screen resolutions are coming out at regular intervals, which will require art assets to be changed into bigger sizes as well to appear sharp on the new screens.

Although the core of the visual style was gotten down relatively fast, deciding on the smaller details like font configurations and button and icon shapes was an undertaking stretching throughout the whole UI design process. It was decided early on that each wizard would have a different color scheme to set their portraits and spells apart from each other. As these color schemes were saturated and colorful, the core UI elements were kept more neutral and desaturated so they would be compatible with each wizard's color scheme. A subdued base would also help other saturated parts of the UI, for example buttons and resource indicators, to pop out.

The fantasy theme of *Wizard Wars* prompted a yellowed parchment design for the information boxes, and the purple-hued stone buttons and dark purple background followed without much effort. Purple is yellow's complementary color, helping it pop out even when it is not that saturated, and it also has an arcane feel to it that fits well with the game's atmosphere. By desaturating and adding greyer tones to the purple, it stopped clashing with the more saturated colors and was instead seen as a base or background element. As the purple surface could be boring in larger quantities, an ornament was designed to be added on top of it when needed. Variations of this ornament have also been used in other UI assets as decoration. The blue hue for the buttons was chosen because of its color harmony with both the yellow parchments and the purple stone and background. In Figure 13 it can be seen that the main colors of the UI elements follow a split-complimentary color wheel formula very closely.

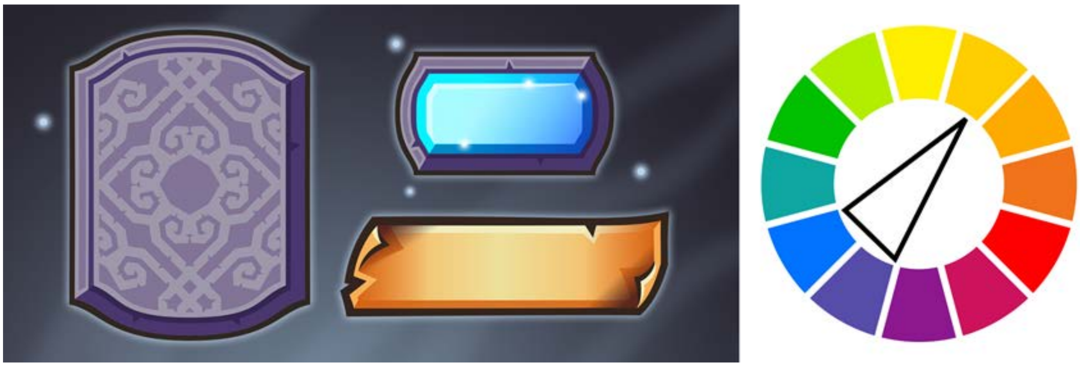


Figure 13. The visuals that make up the core of *Wizard Wars'* UI, and their color wheel formula.

The background started out as just a solid color which was deemed visually boring, so subtle swirls and a light-blue glow and sparks were added to make it feel more alive. To make the UI elements tie into the background, the same light-blue glow was also added around them. This light is done in code, so it can be made to pulse or have other special effects.

5.1.1 Buttons and icons

The buttons and icons of *Wizard Wars* can be divided into three categories: the text buttons that are used to enter commands and move around between the menus; the standalone icons used for resources, stats, and symbol labels for buttons; and the item icons in the *Use and equip items* menu screen. There are also the spell icons, but they will be covered later together with the wizard portraits.

Buttons

Due to *Wizard Wars'* gameplay and target audience it felt natural to use text instead of symbols to label most of the buttons. With its spell and tile descriptions *Wizard Wars* is already more text-heavy than the average mobile game, so a little more text would not feel out of place. Text labels also allow buttons to be more descriptive, and if used correctly with the preceding text they leave almost no room for misinterpretation. For example if the player is asked if they want to accept a trade the buttons will say “Reject”, and “Accept”, and if asked if they want to quit the game, “Cancel” and “Quit”. In both of these cases it is very clear what both button do. With symbol labels both of these examples would have been done with an X and a checkmark.

Looking at the flowchart it was concluded that there were two types of text label buttons that needed different visuals to create a clear hierarchy: the menu buttons with bigger impact such as starting a game or leaving a menu screen; and the pop-up buttons with smaller impact such as confirming a purchase or using an item.

It was established early on that the menu buttons would be embedded into a piece of purple-hued stone taking up the bottom of the screen. This would give a feel of the buttons actually belonging in the design instead of just floating randomly on the screen. The menu buttons went through some significant changes during the UI design process, as can be seen in Figure 14. They started out as more ornate and were gradually stripped of details to fit the evolving visual style better. The extra details were mostly there to make the buttons pop out from the background, and as the details decreased the buttons got more colorful instead to make up for it, which proved to be much more effective in differentiating them from the other UI elements.

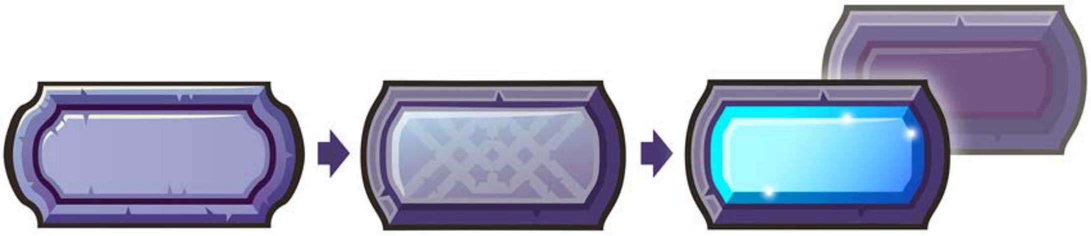


Figure 14. The evolution of the menu button.

The final button is formed of two assets, the blue gem and the purple stone border. They are kept separate to make it possible to add special effects like glows and reflections to just the gem asset. Even if the hidden parts of the stone border is never meant to be visible, it has still been designed to look complete, mostly out of principle, but also as a precaution for slow devices where the two assets may load at different times.

The initial decision to have the buttons in the same purple hue as the background stone was a bad one, and it needed to get changed for the UI to reach the required usability levels. There was some reasoning behind the initial decision though. At this point of the UI design process the *Choose a wizard* screen, where the visual style was defined, was much more crowded with information. Therefore, it was felt that buttons in too saturated colors would overload an already busy screen.

The visuals of the smaller buttons were not defined until later in the design process, when it was time to work on the pop-ups. Not much effort needed to be put into them, as the bright blue used for the menu buttons also worked well with the parchment assets. Using the same visuals also had the added benefit of tying the two button types together.

The flowchart also showed that the amount of text in the buttons could be divided into two distinct categories: the one-word labels and the longer call to actions. Therefore, both buttons come in a long and short version. The inventory and settings buttons were done with symbol labels as they were used in the in-game view as part of the HUD, where a text label would have looked out of place and caused the buttons to be too big.

All in all the final button designs are considered a success. They pop out from the design and the player has an easy time locating them on the screen thanks to the uniform color scheme. As shown in Figure 15 the amount of button assets is also a very limited, which helps to keep the game's file size down.

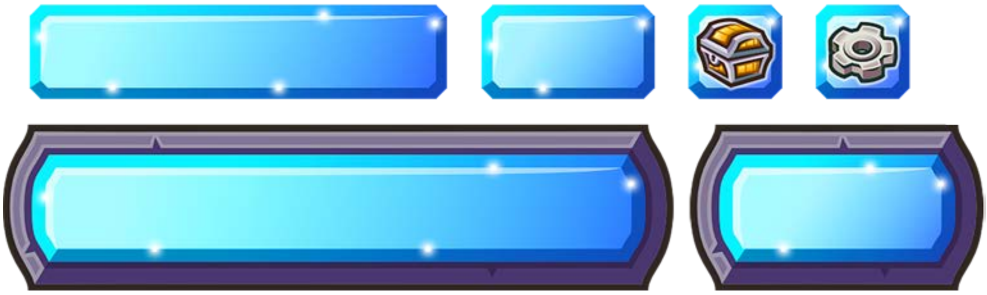


Figure 15. The button assets.

Standalone icons

As can be seen in Figure 16, the first versions of the standalone icons were in a borderline placeholder quality. As the visual style of the rest of the interface became more polished the huge difference in quality could no longer be ignored. Redesigning the icons was not a particularly challenging process but it kept getting pushed back for months, as it required attention to details and perfecting the shapes, which is very tiring to do in larger quantities.

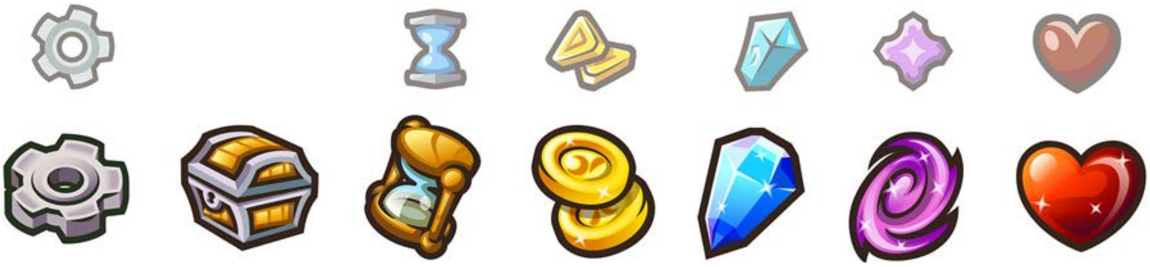


Figure 16. The final icons (bottom) compared to the original version (top).

As these icons are used for different purposes, three versions were made of each of them: a big one for when the resource is gotten as a reward; a small one to be inserted into the font files and used in the description text; and a second small one attached to a gradient bar for the stats in the spell and tile descriptions. The decision to have three versions instead of one asset that would be scaled to fit the different needs were made out of visual polish reasons. Figure 17 illustrates how the manual changes to the outer line thickness makes the icons sink better into the surrounding text. Although not all of the icons are used for all three purposes, the efforts to make the unused ones were nonexistent, and brought the benefit of having the assets ready for possible future use. Some of the icons are also embedded into parchment boxes or used as button labels when suitable.

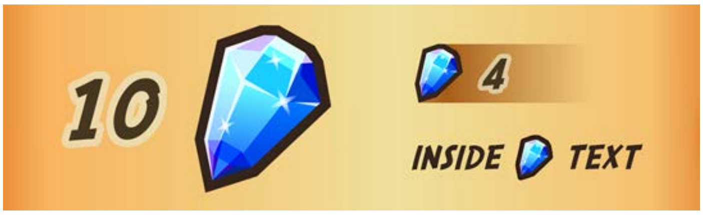


Figure 17. The three versions of the crystal shards icon.

Item icons

The item icons follow the same design used in the wizard spells. However, these icons needed to look different from all the wizard color schemes, which meant saturated pastel colors for the icon bases were out of the question. In addition to this the items are divided into three rarities based on how good they are; common, uncommon and rare, which also needed to be indicated. A metal color scheme solved both of these problems. The metal coloring is single-hued compared to the two-hued wizard color schemes, which creates a big difference; and the gold-silver-bronze hierarchy used to indicate the rarity is already deeply ingrained into our culture. The faint ornament pattern was added to further differentiate the item icons from the spell icons. The pictures of the items were done in the same method as the standalone icons, although with less details to make for a faster production pace. As new item icons need to be added at regular intervals, a too heavy commitment to their quality would have been unwise for both time and resource reasons.

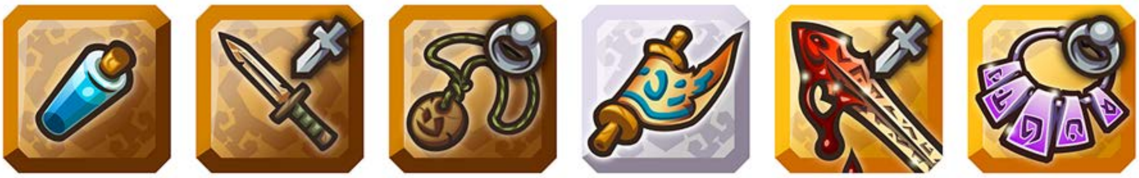


Figure 18. An assortment of item icons.

At a later point in the UI design process it was noticed that the equipment items – divided into weapons, armors, and accessories – were sometimes hard to tell apart by the item picture alone. This was solved by adding a small equipment type indicator to the upper right corner of the icon, as can be seen on some of the icons in Figure 18. This then led to a good UI improvement in the *Use and equip items* screen, as the same icons were added to the bottom of the equipment sockets.

5.1.2 Wizard portraits and spells

To visually differentiate the wizards from each other, it was decided to give each of them their own distinct color scheme. The color schemes were more or less taken directly from the wizards' clothes, and given small saturation and hue changes. This color scheme was then used as a base for the Wizard's portrait and spells. As these assets are present during a game of *Wizard Wars* in both the in-game view and *Upgrade spells* menu, the UI gets somewhat color-themed by the wizard.

The wizard portraits went through some layout changes during the UI design process, as can be seen in Figure 19. The changes were mostly motivated by how they would fit into the in-game HUD. The first HUD portrait was already decent and served the rest of the design well. It was placed at the center of the HUD with spells lodged along the bottom edge, but became obsolete after the gameplay simplifications changed both the layout and requirements of the HUD. The initial *Choose a wizard* menu portrait had a less impressive design, although some parts of it, like the paper with the wizard's name, would serve as a base for the final design.

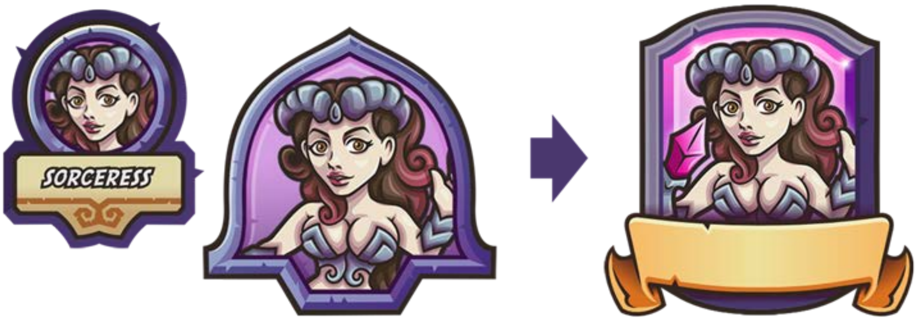


Figure 19. The evolution of the wizard portrait.

The finalized wizard portraits marked the biggest turning point in the UI design process. These portraits were the first assets made with the new, less detailed borders, and the first using an actual parchment with more defined highlights and shadows. The gem parts became more saturated to really pop out from the dark purple background and got a bevel effect on the edges. The sparks were also added to emphasize the gem-like effect. As with the buttons, the wizard portraits are made of separate assets combined together to form the portrait element.

Compared to the other icons made for the game, the wizard spells got significantly more time and effort put into their design. This was for a couple of reasons. Firstly, they would be present in several screens (in-game, *Choose a wizard* and *Upgrade spells*), so the player would spend more time looking at them than the average icon. Secondly, in the in-game view – where they are used to actually cast the spells – they are shown without descriptions, so they have to be memorable and descriptive enough for the player to recognize them. The decision to concentrate more on these icons was also made easier by the limited quantity of them, only five for each wizard, with no indications that more are on the way.

Another goal when designing the spell icons was to create each set of icons to have a strong visual unity. Therefore, identical colors and forms are incorporated into several of the icons belonging to the same set. For example in the Djinn icons in the upper row of Figure 20, four of them have the same pale-green color, three have sand chips flying around, and three have the same wavy wind shapes included into the design.

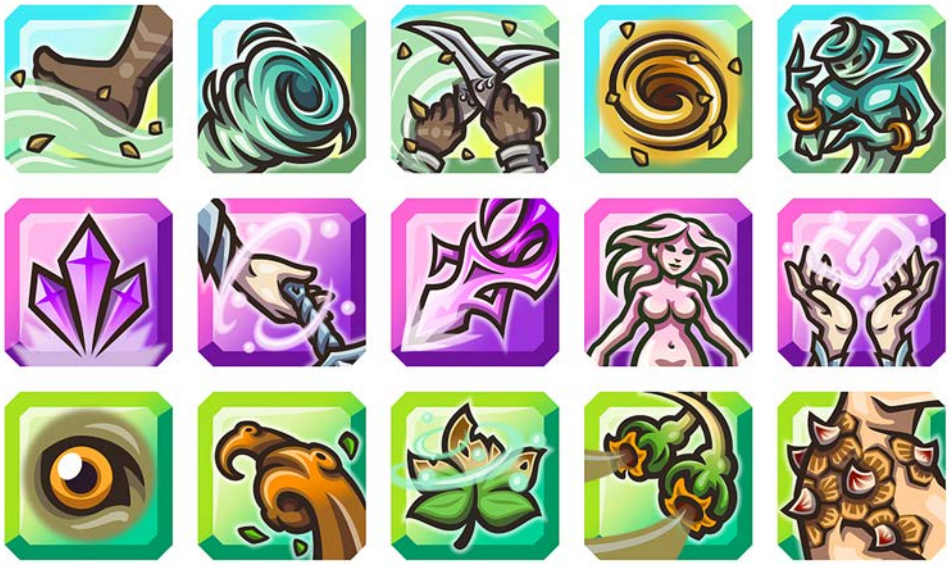


Figure 20. A collection of wizard spell icons.

5.1.3 Pop-ups and boxes

The majority of the text and information in *Wizard wars* is shown on top of boxes and pop-ups made to look like parchments. They went through relatively little change from the initial design to the final one, only getting more vibrant colors and having more tears and folds added for visual interest. Because of the variety in the type and quantity of information the parchments will contain, they had to be made in a few different sizes.

The parchments for the pop-ups were made vertically scalable to better accommodate the different information amounts. The vertical scaling happens by stretching a row of pixels in the middle parts of the parchments by code (the stretching points are marked in Figure 21). The pop-ups come in two different widths, one for smaller announcements and one for information that takes up most of the screen. A version with a place for a title text was also added for both sizes, and proved especially useful for the bigger one.

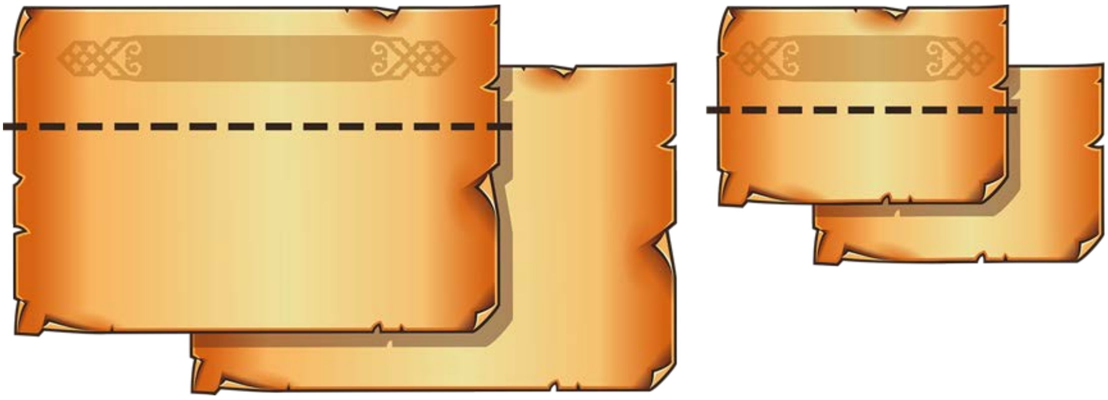


Figure 21. The pop-ups are made of these four parchment assets.

The resource indicators are small parchments only needing to contain a resource icon and a numeric value. They are used in the in-game view to indicate health, mana and crystal shards; and in the *Upgrade spells* and *View and buy tiles* to show the relevant resource. It was decided to include the icons directly into the parchment assets, as it allowed for pixel-perfect positioning between the icons and parchment.

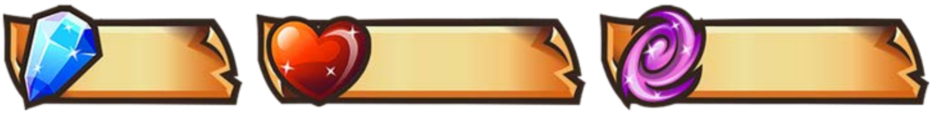


Figure 21. Resource parchments.

The spell and tile description parchments seen in Figure 22 both follow the same layout. There is a place for the icon or sprite on the left, with the rest of the area being divided by a faint line. The upper part is meant for the spell/tile name and stats, such as mana cost and cooldown, and the lower part for a short description. Although the line, socket and ornament details were chosen to be added directly into the parchment assets, the stats icons were kept separate so they could be individually added or removed based on the information needing to be shown.

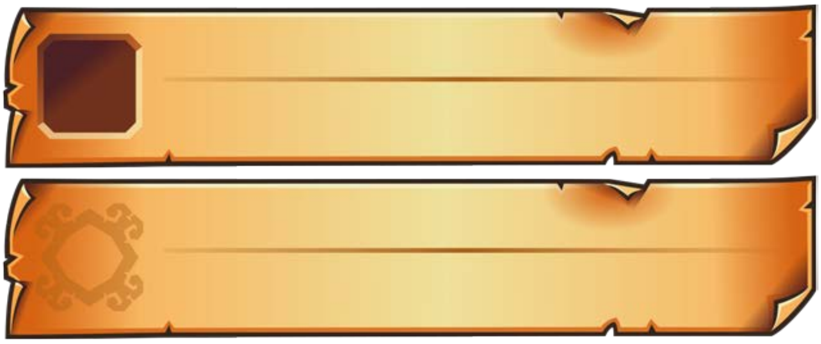


Figure 22. The spell and tile description parchments.

In addition to these art assets, sockets were needed for when a button or icon would be embedded into the parchment. As can be seen in Figure 23 they all follow the same design, with a shadowed upper part and light hitting the lower and right edges. Sockets that are used to indicate item rewards are also surrounded by an ornament pattern for additional impact. As resources can also be gotten as rewards, an asset with the same ornament but without the socket was added to keep the reward visuals consistent.

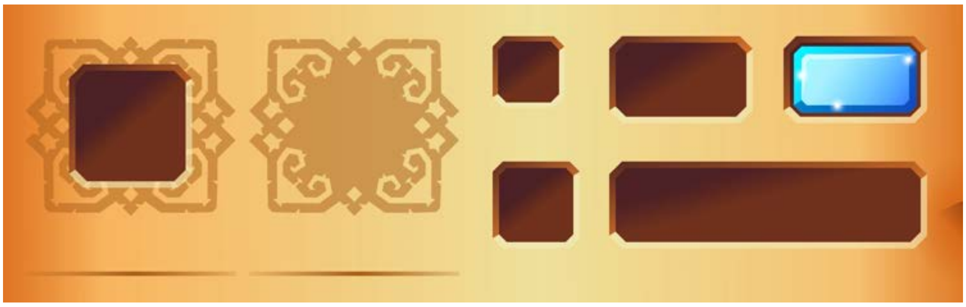


Figure 23. The sockets used to embed buttons and icons into the parchments.

5.1.4 Typography

Choosing a suitable font was essential for the success of the UI design, especially for a game as text-heavy as *Wizard Wars*. The most important attribute when choosing a font is the readability, as text that takes effort to read dramatically decreases the usability of the UI. Therefore, the most reasonable option would be to go with a regular sans-serif, for example Verdana or Arial. However, basic fonts like these would have brought nothing to the theme and atmosphere of the game.

A few tablet games with roughly the same text amounts as *Wizard Wars* were studied to see how acceptable it is to sacrifice some readability for aesthetic reasons. The conclusion was that it is very acceptable. The successful *Kingdom Rush* series has a font very similar to what *Wizard Wars* ended up using, and is using it in smaller sizes. The hugely popular *Plants vs Zombies* uses at least three non-basic fonts – two more than recommended when it comes to UI design.

There was a very specific vision for the font wanted for *Wizard Wars*: a comic-style, all-caps font, with a bit of thickness and sharp edges. It was just a matter of going through the internet font databases to find one that matched the criteria. After looking through more than two hundred fonts and testing out about ten of them in the actual interface, SF Comic Script Regular was chosen. It had all the wanted qualities, and was still easy to read in the spell and icon descriptions – the smallest size it would be used in. It is used without caps, as the flow of the text was smoother without bigger letters at the beginning of a name or sentence. The numbers looking bigger than the text (as they were sized to the caps) caused some hesitance at first, but as the numbers were the part of the description that mattered the most to the player, it ended up looking intentional.

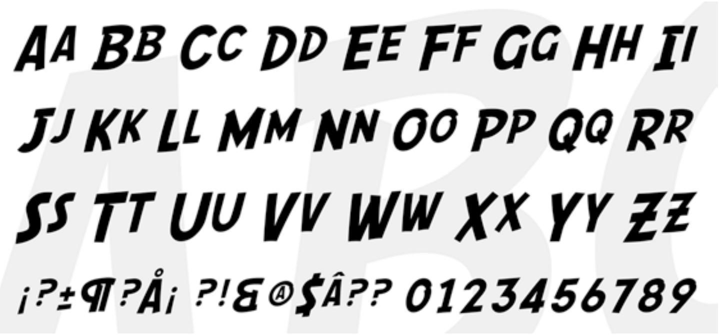


Figure 24. SF Comic Script Regular.

Initially a second font was used for some of the titles. New Rocker has a gothic flair to it that fitted well with the game’s theme, especially in the portrait banners in the *Choose a wizard* screen. It was removed though at the last stretch of the UI design process as the design started to look messy from having two different non-basic fonts and too many size and effect configurations. It was decided to clean up the UI by reducing the fonts to just SF Comic Script, and making the sizes and effects more uniform. In the end the text was reduced from thirteen different configurations down to six, in four sizes and three colors.

SF COMIC SCRIPT 80

Menu screen titles, spell cd, special text

SF COMIC SCRIPT 80

Menu screen buttons

SF COMIC SCRIPT 56

Pop-up buttons

SF COMIC SCRIPT 80

Pop-up titles, menu screen resources

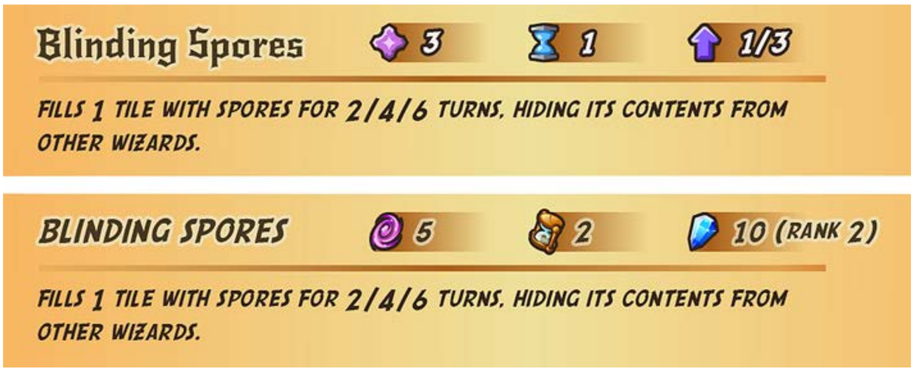
SF COMIC SCRIPT 46

description titles, resources, pop-up text

SF COMIC SCRIPT 32

description text, in-game health and mana, in-game tooltips

The cleanup definitely benefited all the menu screens and UI elements, even if only in subtle ways. To illustrate the impact of these changes, Figure 25 shows the text of a spell icon before and after the cleanup. The title text does not clash with the rest of the text anymore; and the stat numbers next to the title do not pop out as much as before – and neither should they, as they are not an essential enough to be competing for the player's initial attention. In the final version everything is still easily readable should the player want the information, and there is also a better visual hierarchy and unity.



Blinding Spores 3 1 1/3

FILLS 1 TILE WITH SPORES FOR 2/4/6 TURNS. HIDING ITS CONTENTS FROM OTHER WIZARDS.

BLINDING SPORES 5 2 10 (RANK 2)

FILLS 1 TILE WITH SPORES FOR 2/4/6 TURNS. HIDING ITS CONTENTS FROM OTHER WIZARDS.

Figure 25. Font usage in an earlier (upper) and final version (lower) of the spell description.

Localization was also taken into account during the UI design process, especially for the button text. The current font sizes were chosen to make the English text as aesthetic as possible. However, some languages have a much longer average word length, which can be problematic to fit into the button assets. In this case there are two options: reduce the font size; or increase the width of the button assets, which is viable as there is empty space next to the buttons in both the bottom bar and in the pop-ups. The decision how to approach the localization will be made if and when any localization will be done.

5.1.5 Miscellaneous assets

In addition to all the core elements making up the UI, there were a few menus and pop-ups with functions that needed extra assets. One of them was the slider shown in Figure 26. It was made primarily for the *View and buy tiles* menu to enable scrolling through the tiles list, and was also later used in the settings pop-up as music and sounds adjustment bars. The bar asset uses the same pixel-stretching technique as the pop-up parchments to create bars in any needed size. The shading was made in a way that allows the slider to be used both horizontally and vertically without looking out of place.

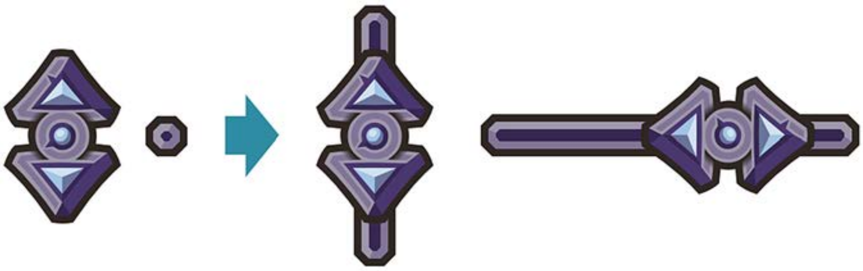


Figure 26. The slider assets.

The other extra asset was the text-field shown in Figure 27, used in the create username and invite friends pop-ups. It was very easy to create, as it only involved making a longer version of the pop-up button socket and creating a light-beige field with shaded edges to go into it. A faded-out call to action was added as the default text, so the player will know right away what information they are expected to enter. The text-fields in the invite friends pop-up also needed options to send and cancel the invite. Therefore, a square socket and small X and checkmark buttons were created in the same height as the text-field, so they could be placed next to it.

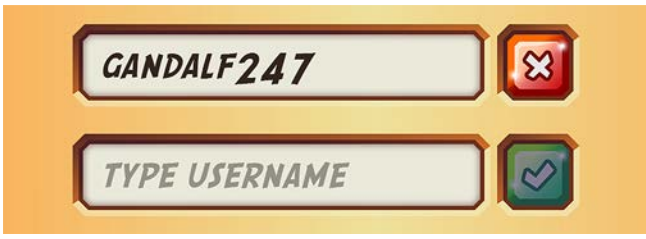


Figure 27. The invite friends text-fields in different states.

The creation of the slider and text-field showed how easy it is to create new assets when the visual style and the core elements of the UI have been well defined and thought out. All of these assets were created by using already established forms and color schemes, and by modifying existing assets.

5.2 Main menu

The main menu screen got relative little attention as far as UI design goes. As it serves as a central hub from where the player can access the other parts of the game, the focus was on making the navigation as quick as possible. It contains four menu buttons and two sub-buttons for the settings and wiki page link. Buttons were made large enough that they are impossible to miss and to decrease the probability of miss-tapping. The menu buttons were aligned vertically along the center of the screen, creating a list of options the player can quickly read through and digest. The sub-buttons were then aligned horizontally with the bottom two menu buttons to create structure to the screen. Without proper alignment it would be easy for the viewer to just assume the buttons were placed randomly.

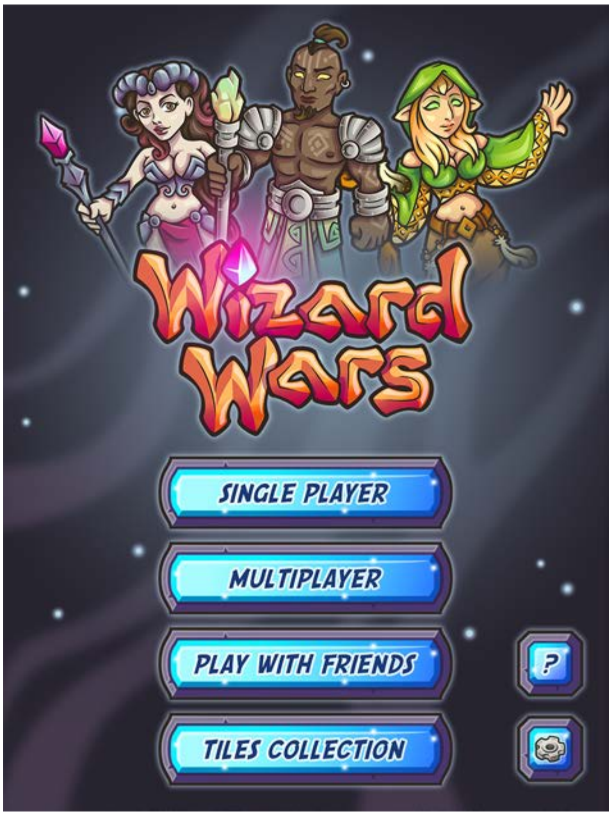


Figure 28. The main menu screen.

It would have been preferable to have more interesting visuals for the top of the screen than the logo and wizards. The main menu is the first thing the player sees when they start the game, so it should set the mood and be as visually impressive as possible. In other menu screens the whole screen estate is used to show information and options, but in the main menu the UI elements are limited to a few buttons, which leaves space for a bigger decorative image. Knowing the production of a full- or half-screen image would be very time-consuming, the decision was made to leave it out from the first version of *Wizard Wars* in favor of polishing other visuals. Figure 29 showcases a few sketches of different main menu images designed around the current button placement. Especially the forest design on the left would be interesting to explore further.

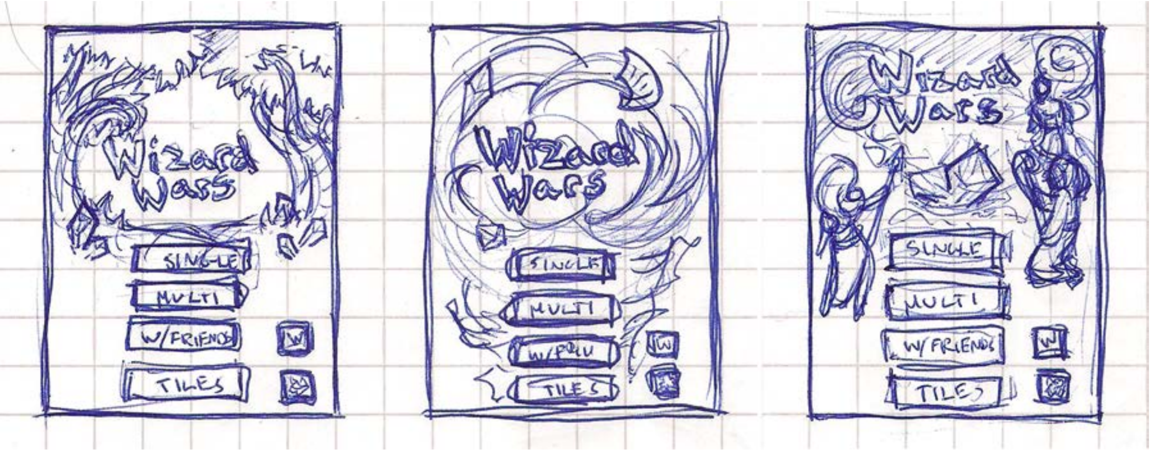


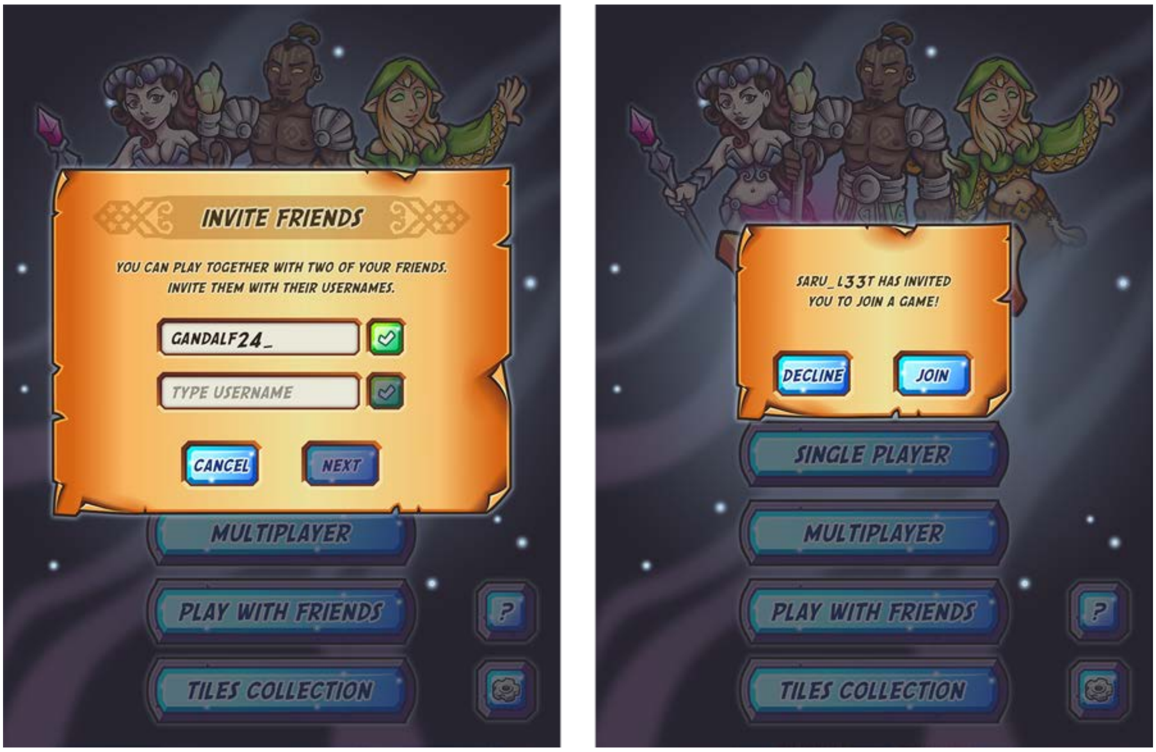
Figure 29. Sketches of ideas for the main menu visuals.

Play with friends

Choosing either the single player or multiplayer game mode takes the player straight to the *Choose a wizard* menu screen, but before starting a play with friends game both the inviter and the invited have to go through an extra pop-up, as shown in Figure 30.

For the invited it is simple. If they are not in the middle of a game of *Wizard Wars* they will get a pop-up informing them they have been invited to a game by the inviter, and they can either join or decline. If the invited isn't currently in the game but has the game installed, they will be alerted through the iPad Notification Center. They can then enter the game and get the invitation pop-up. The invite is active for five minutes or until the inviter cancels it.

The inviter gets a pop-up with an explanatory text and two text-fields asking them to type the usernames of the players they want to invite. Each text-field has a green checkmark icon next to it, which becomes active when the inviter enters the username. The invite is sent by tapping the checkmark icon. When the invite is pending the icon changes from a green checkmark into a red X, which can be used to cancel the invite or remove the invited player after they have accepted the invite. When the two players have been successfully invited, the next button is activated, and by pressing it the inviter and the invited players are brought to the *Choose a wizard* menu screen.



INVITE FRIENDS

YOU CAN PLAY TOGETHER WITH TWO OF YOUR FRIENDS. INVITE THEM WITH THEIR USERNAMES.

GANDALF24_

TYPE USERNAME

SARU_L33T HAS INVITED YOU TO JOIN A GAME!

Figure 30. The invite friends pop-ups shown to the inviter (left) and the invited (right).

Settings

Pressing the settings button opens up a pop-up where the player can adjust the music and sounds volumes, and access options that were deemed to not be used often enough to warrant a button in the main menu screen. The in-game view and tutorial, the other game screens to have access to the settings pop-up, have the options replaced with “Quit game” and “Quit tutorial", respectively. Although the settings popup is not the most logical place for these buttons, the advantages of having a single place to put all of these in most cases redundant options far out-weighted the inconvenience of the player having to learn where to find them.

Unlike the other pop-ups the settings needed to have a close button, so one was added in the top-right corner. Although the buttons in the settings do take the player out of the pop-up, the player needs to be able to exit it without choosing any of the options, as it is very possible they are just there to adjust the music and sounds volumes.

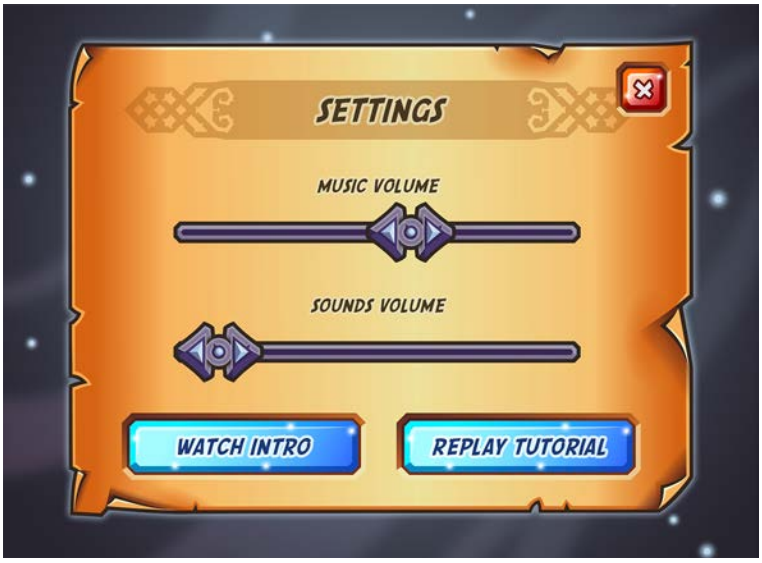


Figure 31. The settings pop-up for the main menu screen.

One-time pop-ups

When the player starts *Wizard Wars* for the first time, they are presented with a couple of one-time pop-ups. When any of the game modes are started a pop-up asking if the player wants to play the tutorial first is launched. Regardless of the player choosing to skip or play the tutorial the pop-up will not be shown again (the tutorial can be accessed later from the main menu settings). The decision to make the tutorial skip-able was based on the fact that there are valid reasons for the player not wanting to play through it. They can be installing the game on a new device, or have already learned the game mechanics from watching a friend play. Or they might simply dislike tutorials and prefer to learn the game the hard way.

The other one-time prompt is asking the player to create their username, and is launched when the player starts a multiplayer or multiplayer with friends game. This one keeps launching until the player creates a username, as it is needed for the player to be able to take part in multiplayer games. The player is returned to the main menu if they choose to cancel the username creation. As it is indicated clearly in the accompanying text that the username is mandatory, the player know exactly why they cannot proceed which should reduce potential frustrations.

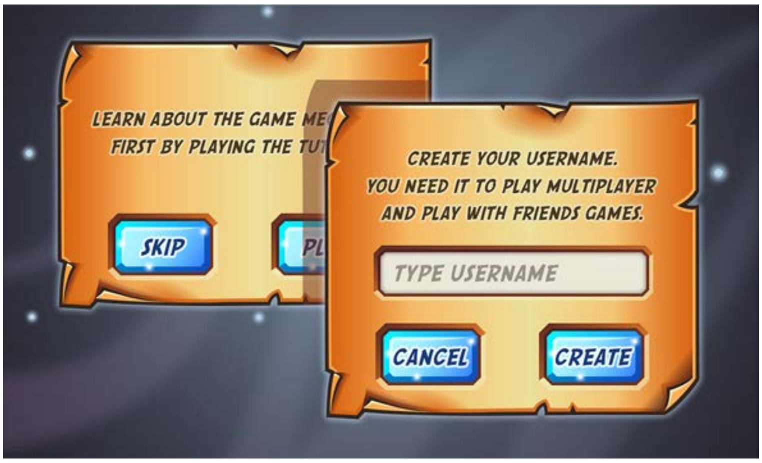


Figure 32. The pop-ups for playing the tutorial and creating the username.

5.3 Choose a wizard

The visual UI design process was started with the *Choose a wizard* menu screen. The whole visual style was – with the exception of a few assets – fully defined during the designing of this screen, making it the only screen to go through several major iterations. The choice to start the UI design from the *Choose a wizard* screen was made for the reason that it was the most information-heavy screen in the game. It contains more or less an instance of every major UI element in the game: a menu button, a wizard's portrait and spell icon, and an informational text box (parchment). Therefore, it was the perfect screen to ensure that all the elements would fit together and the visual style would be consistent.

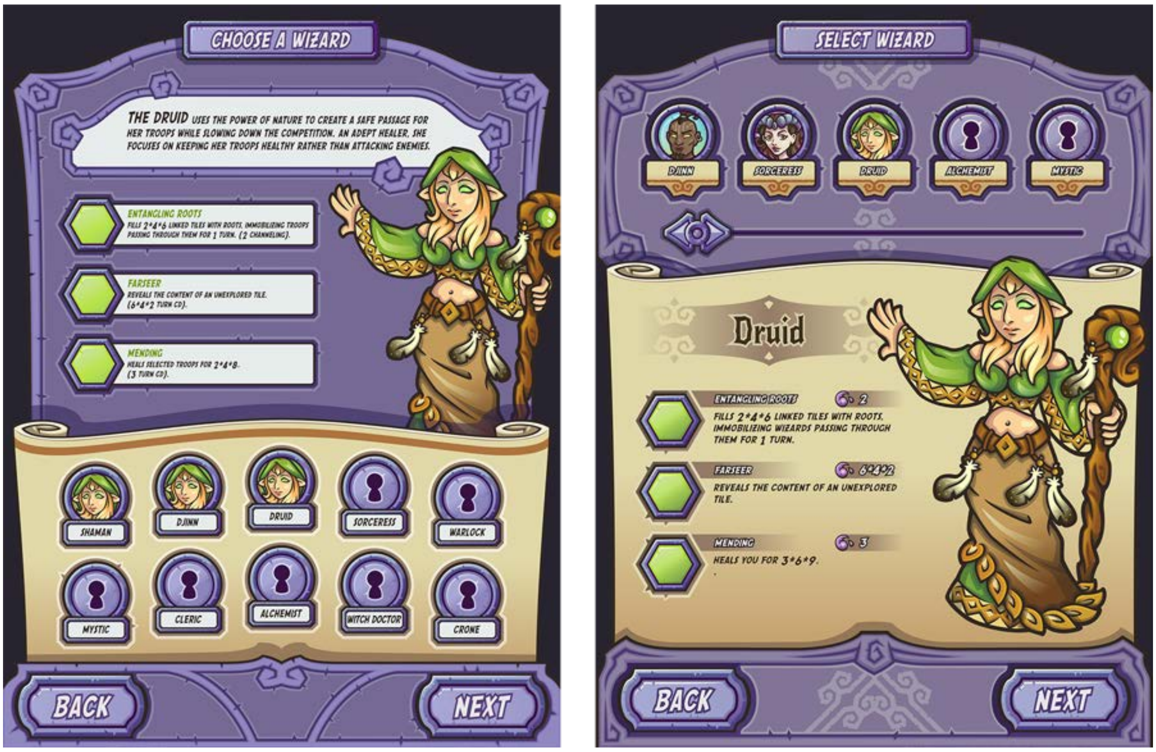
According to the flowchart, the *Choose a wizard* menu screen needed the following information and controls: at least six different wizard portrait buttons with a solution to increase the amount; three spell descriptions (icon and text); the menu title text; and the back and start buttons. It might not seems much, but the physical dimensions of the Ipad screen are small enough that this amount pushed the limits, especially if everything was to be kept at a reasonable size.

The back and start button placement and the purple stone holding them are the only elements that made it to the final design relatively unchanged. This can be largely attributed to the fact that they adhered to actual user interface theory. Ever since the first sketches, the buttons placement at the bottom of the screen felt natural, as the bottom part of the screen will be closer to the player's body and thus easier to reach. In comparison, when reaching for buttons at the top of the screen the player's arms would block part of the screen. This is problematic if the button is used to toggle between information, but the information getting changed would be blocked behind the arm, forcing the player to move his arm between each toggle.

To align the buttons horizontally was both aesthetically pleasing and a screen estate optimizing measure. Having the start button on the right side of the screen and the back button on the left is a concept deeply ingrained in western culture. For example when reading a book, the reader proceeds by turning the right-side page, and goes back by turning the left. This kind of button placement also helps if there are plans to localize the game to a right-to-left culture, as the buttons' positions can simply be swapped around.

A small but important detail which also made it into the final design was to have the menu titles as prompt to actions instead of descriptive text. "Choose a wizard" would have been "Wizard selection" or "List of wizards" in most games. The decision to use prompts instead removed the need for additional explanatory text of what the player is meant to do in each menu screen.

The first few designs seen in Figure 33 were done before the decision to go for a modular UI design, and they had many problems. The purple-hued stone and yellowed parchment theme was established in the first days of the design process and deemed a good color scheme, but they were pushed too hard. The screens were overloaded with the purple and yellow, and by having so much of both colors they ended up washing each other out and losing visual impact. To make matters worse, important elements such as the buttons and the screen title were in a color very similar to the purple that was filling most of the screen.



THE DRUID USES THE POWER OF NATURE TO CREATE A SAFE PASSAGE FOR HER TROOPS WHILE SLOWING DOWN THE COMPETITION. AN ADEPT HEALER, SHE FOCUSES ON KEEPING HER TROOPS HEALTHY RATHER THAN ATTACKING ENEMIES.

FILLS 2*4*6 LINKED TILES WITH ROOTS, IMMOBILIZING TROOPS PASSING THROUGH THEM FOR 1 TURN. (2 CHANNELING).

REVEALS THE CONTENT OF AN UNEXPLORED TILE. (6*4*2 TURN CD).

HEALS SELECTED TROOPS FOR 2*4*8. (3 TURN CD).

FILLS 2*4*6 LINKED TILES WITH ROOTS, IMMOBILIZING WIZARDS PASSING THROUGH THEM FOR 1 TURN.

REVEALS THE CONTENT OF AN UNEXPLORED TILE.

HEALS YOU FOR 3*6*9.

Figure 33. Two of the first designs for the *Choose a Wizard* menu screen.

The most critical problem though was the fact that the screen estate was utilized very badly. The big wizard image definitely gave some visual interest to an otherwise bland screen, but it brought no informational value to the screen while still occupying a large part of it. More importance was placed on ornate background visuals than the interactive UI elements. There were huge gaps between elements and they were made smaller so the screen would not look too crowded.

When it was realized most of the visual clutter came from the background visuals, the design started to evolve. The amount of parchment and purple-hued stone was reduced in favor of the dark background. As can be seen in Figure 34, designs started to utilize floating objects to hold the spell descriptions and eventually moved on to have each spell be its own floating element. The wizard portraits also started to get bigger as more screen estate was freed up.

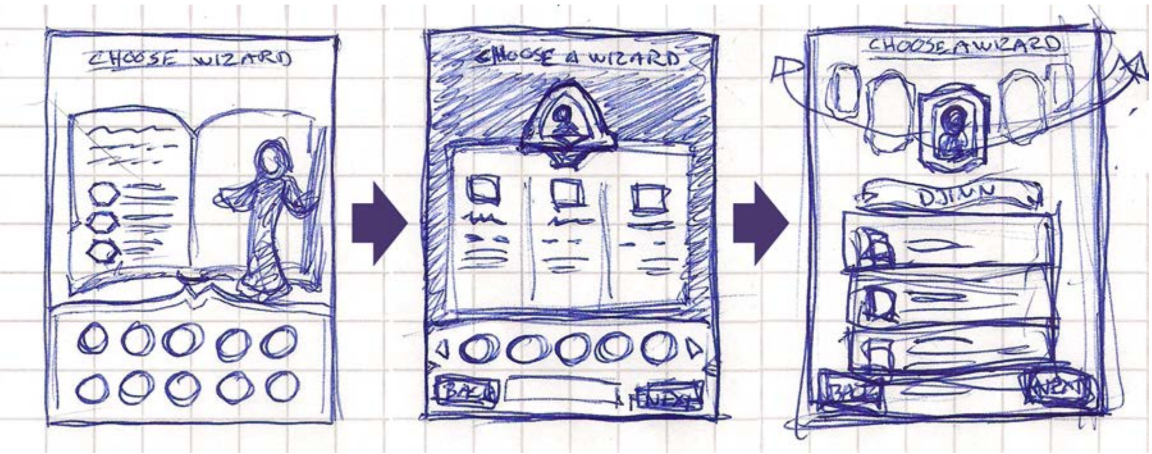
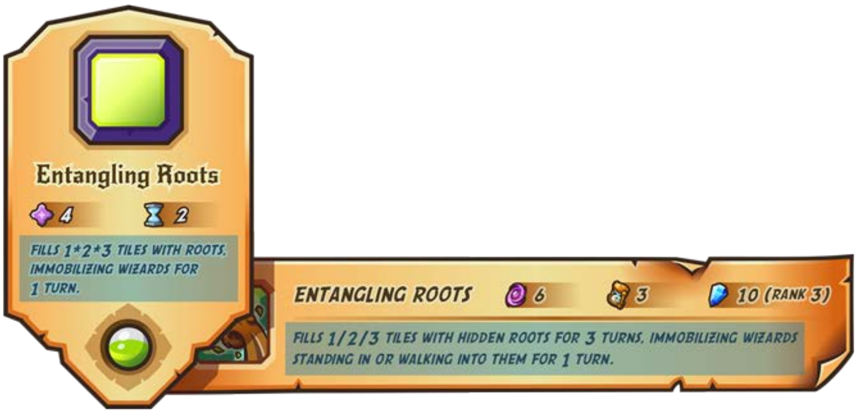


Figure 34. Sketches of more modular versions of the *Choose a Wizard* menu screen.

The first version of the individual spell description was done with a vertical design, mostly out of consideration for the *Upgrade spells* menu screen. It would have allowed for a clear hierarchy between the three basic spells and the two unlockable ones, by placing the basic ones in the bottom row and the unlockable ones on the top row. The main disadvantage with this design was that the readability of the text suffered. It had to be unreasonably small to flow naturally and never fitted more than a few words per rows, which made it unappealing to read. This design was eventually changed for a horizontal version which both utilized the screen estate better and more than doubled the space for the description text, as is shown in Figure 35.



FILLS 1*2*3 TILES WITH ROOTS, IMMOBILIZING WIZARDS FOR 1 TURN.

FILLS 1/2/3 TILES WITH HIDDEN ROOTS FOR 3 TURNS, IMMOBILIZING WIZARDS STANDING IN OR WALKING INTO THEM FOR 1 TURN.

Figure 35. The first and final versions of the spell description, with their text areas marked.

The second big leap towards the final design was the realization that the mandatory tap in the wizard selection process – when the player taps on a wizard's portrait to select it – was not utilized at all in the UI design. Why not have the tap change what information is shown instead of trying to fit it all into one view?

This lead to splitting the information so only the wizard portraits are shown in the default *Choose a wizard* view, which makes for a more visually interesting screen, as can be seen in Figure 36. Tapping on a wizard will center that wizard's portrait and bring out the spell descriptions (or unlock criterion in case of a locked wizard) while quickly shrinking the other portraits out of view. The start button is also activated in this view as a wizard has been selected. By tapping anywhere on the screen the wizard is deselected, and the animation is reversed to return the player to the default view.

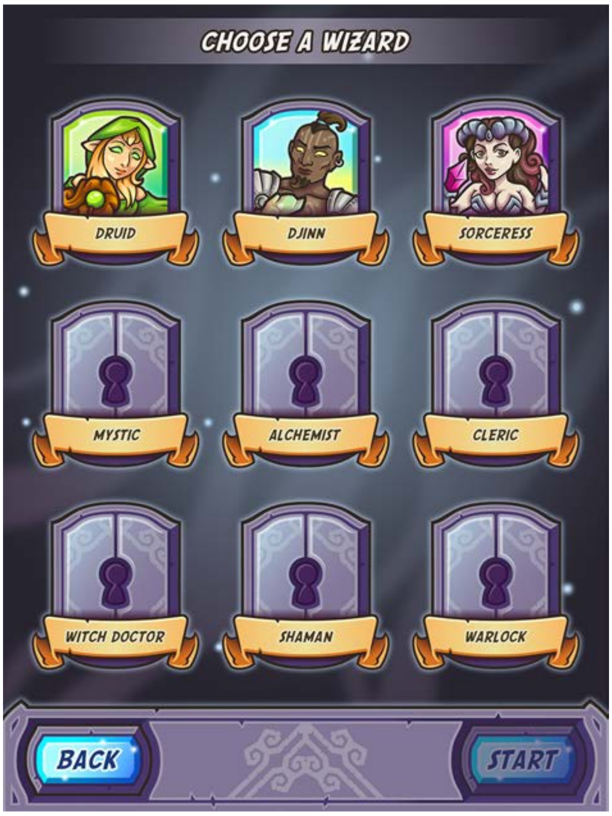
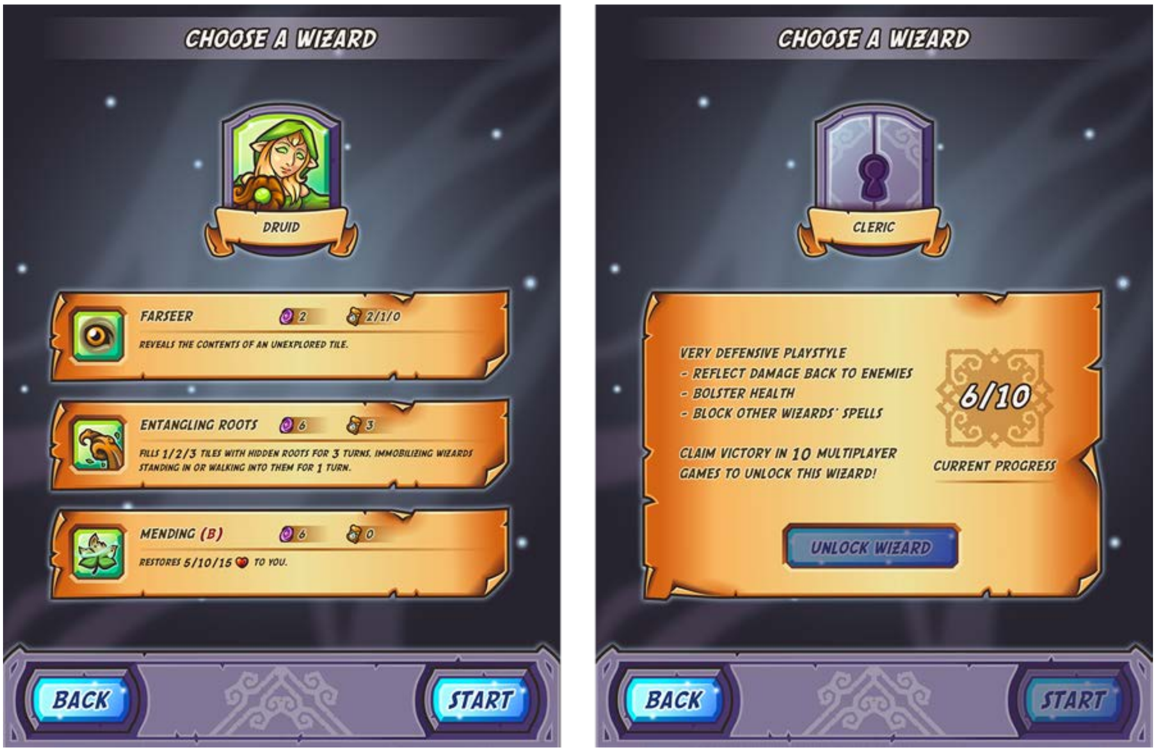


Figure 36. The default view of the *Choose a wizard* menu screen.



REVEALS THE CONTENTS OF AN UNEXPLORED TILE.

FILLS 1/2/3 TILES WITH HIDDEN ROOTS FOR 3 TURNS, IMMOBILIZING WIZARDS STANDING IN OR WALKING INTO THEM FOR 1 TURN.

RESTORES 5/10/15 ❤ TO YOU.

VERY DEFENSIVE PLAYSTYLE
- REFLECT DAMAGE BACK TO ENEMIES
- BOLSTER HEALTH
- BLOCK OTHER WIZARDS' SPELLS

CLAIM VICTORY IN 10 MULTIPLAYER GAMES TO UNLOCK THIS WIZARD!

Figure 37. The views after choosing an available and locked wizard.

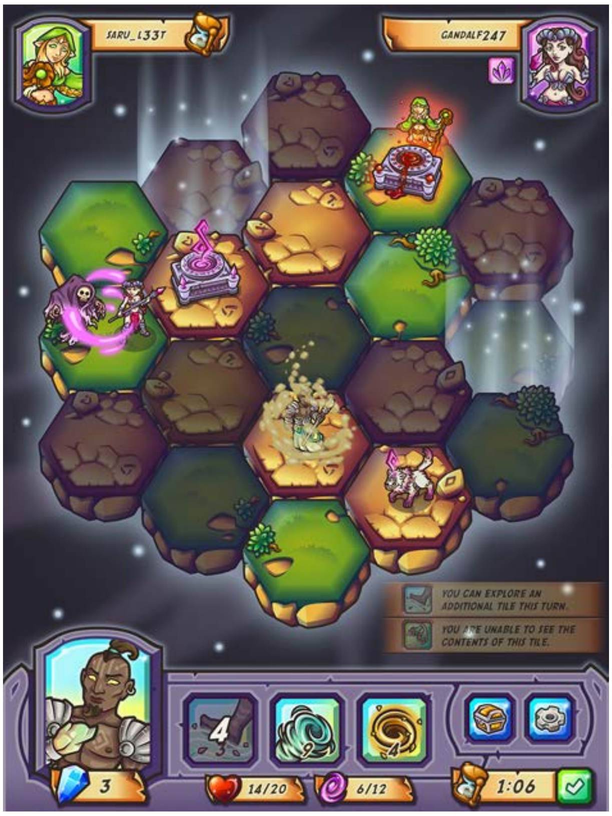
It is acknowledged that there is a flaw in the final *Choose a Wizard* menu screen design. There is no clear way to increase the amount of wizard portraits past nine. There is a couple of easy ways to solve this though. A page change mechanism can be added to the middle of the stone element holding the back and start buttons, or a vertical slider can be added to the right side of the screen. The right-most concept from Figure 34 can also be returned to as a possible solution, with a ring of portraits at the top that can be swiped through, and the corresponding spell descriptions appearing under the portraits. This design was actually very appealing, but was put aside in favor of the current one, as it gave the player a clearer picture of the total number of available wizards.

This will not be relevant for quite some time though. At the time of writing this thesis, there are only spell designs and unlock criteria for three of the locked wizards; the Mystic, Alchemist and Cleric, while the other three only have some preliminary playstyle and spell ideas documentation.

5.4 In-game view and HUD

The main goal for the UI design in the in-game view was to reduce the amount of information and options to display, and thus minimize the HUD's presence in the screen as much as possible. Strategy games are by default very HUD- and menu-heavy, and although the gameplay has been simplified greatly compared to the beginning of the project, there were still a considerable amount of UI elements that needed to be included.

Most of the information in the in-game view is shown by diegetic means, in other words through character and environmental assets, animations and special effects, leaving mostly controls and information needing numeric values to be displayed in the HUD. As shown in Figure 38, the hexagonal shape created by the tiles, with a tip going upwards and another downwards, prompted a HUD design where the elements are positioned in the corners of the screen.



YOU CAN EXPLORE AN ADDITIONAL TILE THIS TURN.

YOU ARE UNABLE TO SEE THE CONTENTS OF THIS TILE.

Figure 38. The in-game view.

The HUD can be divided into two distinct parts: the bottom with all the interactive elements and information concerning the player; and the top with each corner dedicated to one of the opposing wizards. This aligns well with the starting positions of the wizards, as each player will see the in-game view with their wizard starting at the bottom of the screen. As was already rationalized in the *Choose a wizard* menu, positioning interactive elements at the bottom of the screen allows for easier access and improved usability, by not having the player's arm come up to block crucial information every time they want to perform an action. Of course the player will still have to move their hand to the upper parts of the screen to interact with the tiles there, but with this HUD layout the interaction has been kept to the lower part of the screen as much as possible.

Lower HUD

The lower HUD compiles all the information the player needs to know and all the actions they can perform. The wizard portrait is the first element grabbing the attention, and serves as the button for the *Upgrade spells* menu. This felt logical, as the spells are the manifestations of the magical powers residing within the wizard, and tapping on the wizard lets the player also see said powers. Continuing right, the three spells that can be used during regular gameplay are lined up horizontally. As they are the main buttons being used in the screen, they take up the center of the HUD. The *Use and equip items* and the settings buttons are tucked away to the right side of the screen. Even though the *Use and equip items* button is more important than the settings and should thus be bigger in size, they were kept the same for layout reasons.

The resource parchments take up the bottom edge of the HUD. The crystal shards parchment is placed partially on top of the portrait, since the amount of shards is relevant to the *Upgrade spells* menu. The crystal shards are the main resources in the game, so the health and mana parchments were made smaller so they wouldn't fight for the player's attention. They are placed under the spell icons – again for relevancy reasons – as mana is used to cast spells, and the health might be affected by some spells. The turn timer and the end turn button takes up the bottom right of the screen as it is the only space left. It also fits well with the other functions associated with that corner of the screen, as ending the turn and as a result starting the next one is perceived as going forward. The turn timer parchment was kept as big as the crystal shards parchment, as the information was of the same importance. This also balanced the bottom HUD out well, giving the left and right side a similar visual weight.

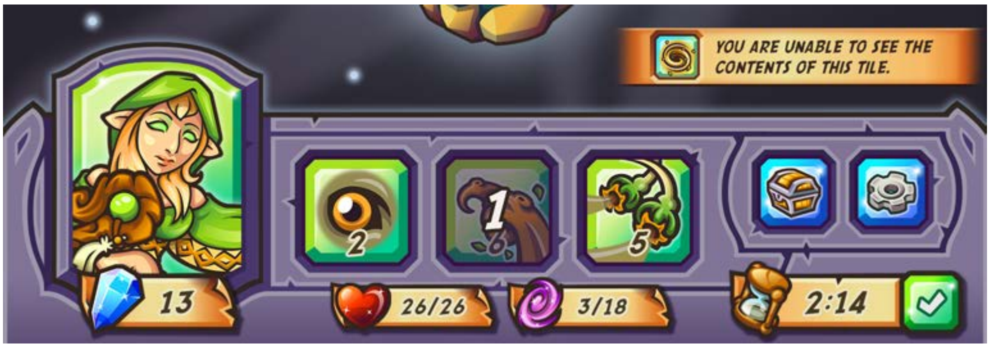


Figure 39. Close-up of the bottom HUD.

The small spell and item effect descriptions floating on top of the bottom HUD gives the player information on the spells and items currently affecting them. This information can also be deciphered from the in-game view when the player gets familiar enough with the game to recognize spell effects. These descriptions are mostly redundant after the player has read them, and because of this they are faded out by default as can be seen in Figure 38, and are only shown in full opacity a few seconds after coming into effect or when getting tapped, as seen in Figure 39. This created a good compromise of the information being out of the way for players who do not want it, and available for players who do. Another solution would have been to hide these descriptions under a button, but it was deemed risky as new players might not find them. In retrospective that risk could have been averted easily by setting the default state of the button to show the descriptions.

Opposing wizard's portraits

Although the majority of what is going on with the two opposing wizards is seen from the in-game view, there are some information and status updates that are clearer to show by non-diegetic means. These are found in the portraits occupying the top corners of the screen.

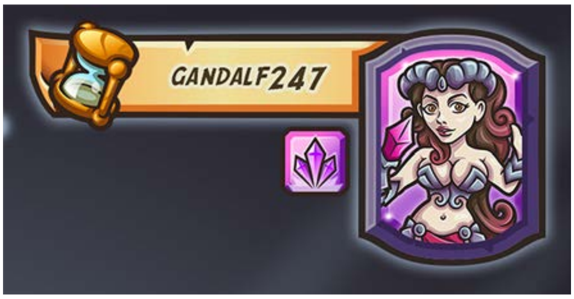


Figure 40. Close-up of one of the opposing wizard portraits.

The opposing player's username (or wizard's name in single player) is show in the parchment jutting out of the portrait. Although this information is mostly of no consequence outside of play with friends games where it is nice to see who is playing as which wizard, it is still a good information to include in case of bad player behavior that needs to be reported. Or in a more positive case, if a player enjoyed a game with another player, they can invite them for future games as they now know the player's username.

The outer edge of the parchment is reserved for the hourglass icon, as shown in Figure 40. When the hourglass is visible it indicates that the player is ready pass the turn, and is waiting for other players to do the same. Possible spell effects affecting the wizard are aligned horizontally in a row under the parchment. The portrait itself is not used for much other than indicating which wizard's information is in which corner. If a player is disconnected, the portrait picture is replaced by a stone texture to inform the other players of the situation.

Battle view

When the player explores a monster tile and as a result encounters a monster, the screen switches into battle view. Instead of seeing the whole game area, the player sees a close-up of their wizard and the monster, giving the impression of the view having zoomed in on the tile. When the battle ends the view switches back to the in-game view, and if the player won the fight an item pop-up showing the reward(s) is launched. There is potential for some very nice special effects to facilitate the switching of views.

In the battle view the opposing wizards' portraits disappear, leaving only the bottom HUD. The spells are changed into the battle spells and an attack button. A health parchment is also added under each enemy on the screen to indicate its current health. The button to end the turn is disabled, as the player cannot end the turn in the middle of combat. The access to the *Use and equip items* menu is kept active though, as the player can choose to spend their combat turn on using an item.

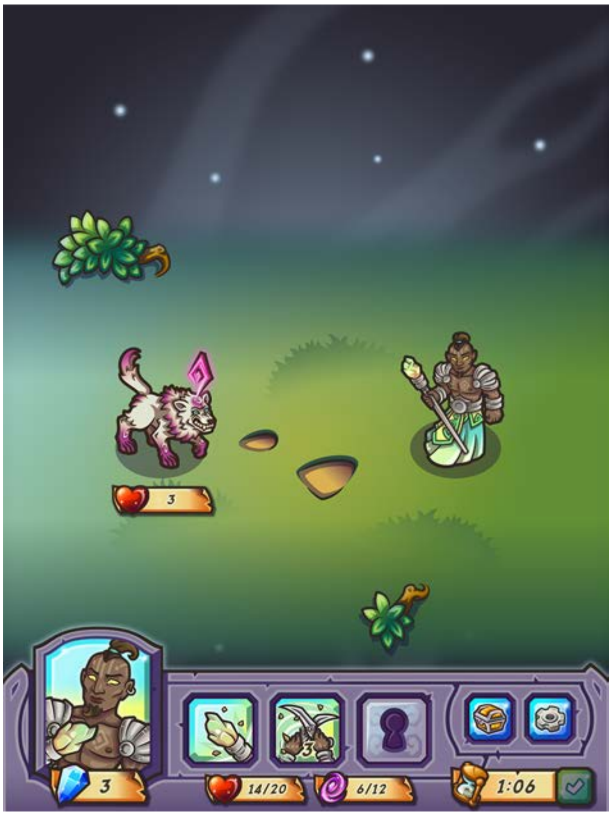
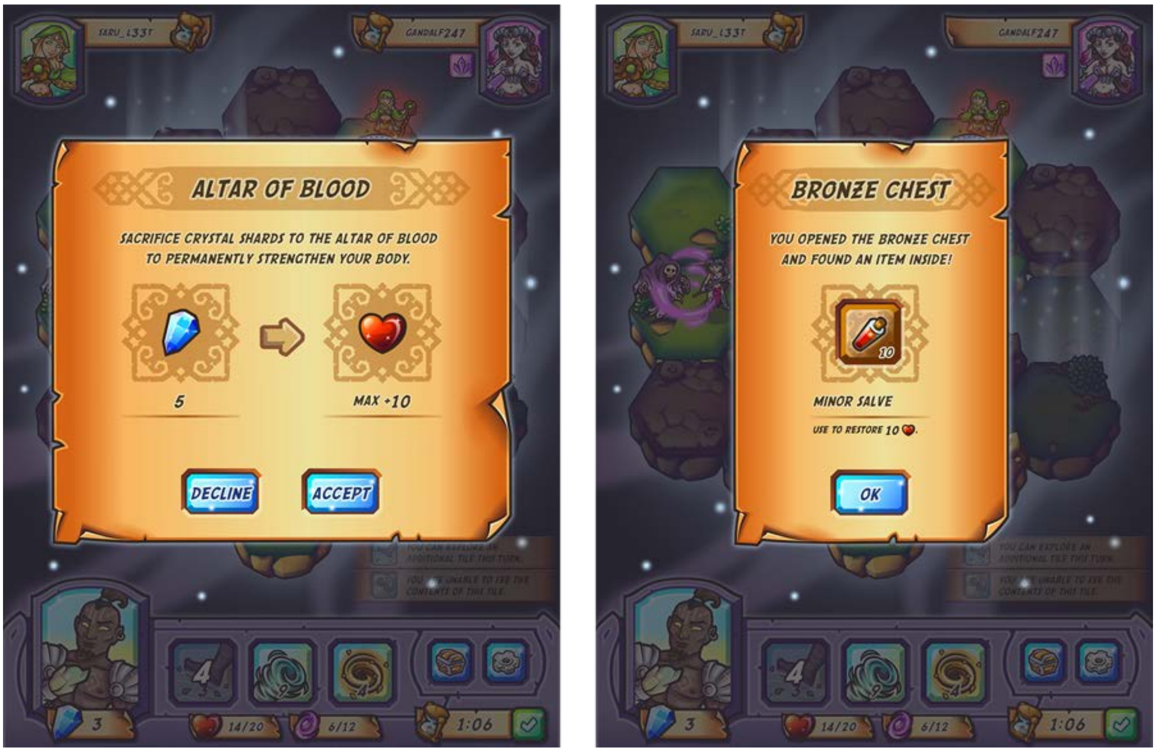


Figure 41. The battle view.

Pop-ups

The in-game view utilizes pop-ups to let the player interact with tiles as they explore them. The event pop-ups revolve around letting the player trade their resources for other resources, max value increases, and items. The information of what is traded for what is shown in the popup together with an explanatory text, and the player can choose to either accept or decline the trade. The item pop-up is simpler, only informing the player of the resources or item they have obtained. The pop-ups utilize the equipment slot UI element found in the *Use and equip items* menu, with a socket-less version for the resources, to display the item and resource descriptions.

In addition to tile interaction, the pop-ups are also used to show the victory and defeat game states. Although these pop-ups were tried to be made more impressive by changing the glow surrounding them, they are still very underwhelming for pop-ups that signal the end of the game. They still serve as a good base though, and the problem can most likely be remedied in future game versions by adding a frame asset around the edges of the parchment.



SACRIFICE CRYSTAL SHARDS TO THE ALTAR OF BLOOD TO PERMANENTLY STRENGTHEN YOUR BODY.

YOU OPENED THE BRONZE CHEST AND FOUND AN ITEM INSIDE!

Figure 42. A sample of event (left) and item (right) pop-ups appearing in the in-game view.

5.5 Upgrade spells

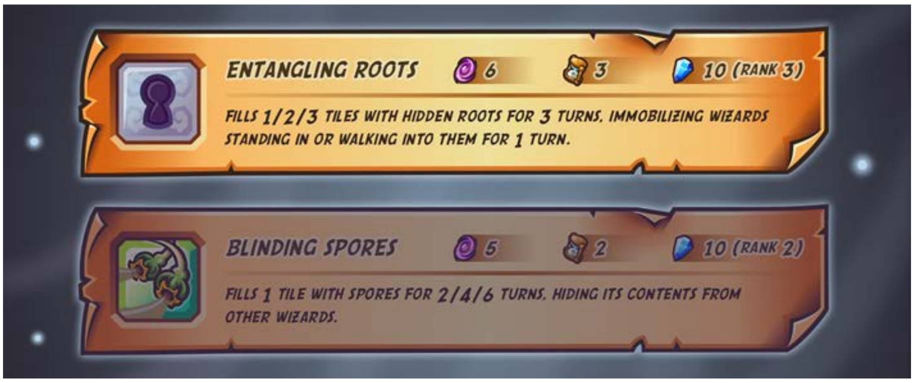
The functionality for the Upgrade spells menu screen is simple. The player taps on a spell description to launch a pop-up to upgrade the spell, providing they have enough crystal shards to do so. If they do not or the spell is already at its max rank, the tap will instead cause the spell description to shake briefly and a rejecting sound to be played.

The layout of this screen was relatively easy to design as the main components – the spell descriptions, had already been defined earlier in the *Choose a wizard* menu screen. This menu screen was referred to occasionally during the spell description design by ensuring the parchments would work visually as the main UI elements in this view, and that the dimensions of the parchments allowed for five of them to fit comfortably on the screen. As the same parchment would be used for the tile descriptions in the *View and buy tiles* menu screen, the width of it was set so the screen estate could accommodate a vertical slider next to the parchments.

The only new elements for this screen were the one-socketed version of the stone button base and a bigger container for the crystal shards resource. The single button was placed on the left side as its function was the same as the other left-side buttons, namely going back to the previous screen. The bottom-right corner also suited the more essential crystal shards better, as it has stronger visual gravity, meaning the eyes naturally move to that part of the screen.

One of the challenges of the *Upgrade spells* menu screen was how to convey the locked state of the spell. Every game of *Wizard Wars* starts with two of the five spells locked. By upgrading the basic spells to their max ranks, the locked spells are unlocked. The spell descriptions still needed to be readable when locked to give the player more incentive to unlock them, and to be able to make an informed decision on which one to unlock first.

The initial solution was to replace the spell icon with the lock icon used to mark a locked spell in the HUD. Although it should have worked in principle, the change was not clear enough as the rest of the spell description still looked normal. It also made the aesthetics of the screen feel a bit off, as the stone icon looked strange inside the parchment and the unified color scheme created by the spell icons was broken. Therefore, it was changed to the whole description having a dark purple color overlay which was set to medium opacity, so the text was still easily readable from under it. This signaled the unavailability of the spell much clearer and did not create any visual discrepancy.



ENTANGLING ROOTS 6 3 10 (RANK 3)

FILLS 1/2/3 TILES WITH HIDDEN ROOTS FOR 3 TURNS, IMMOBILIZING WIZARDS STANDING IN OR WALKING INTO THEM FOR 1 TURN.

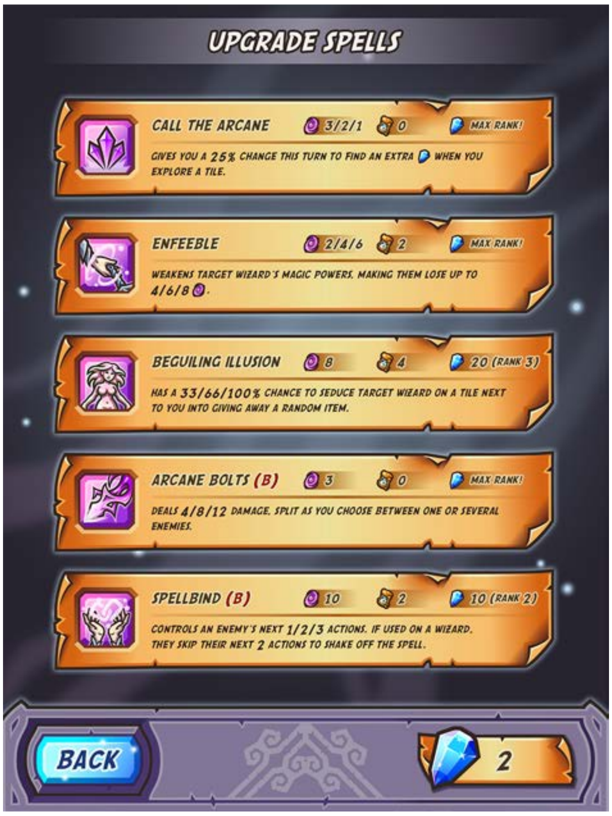
BLINDING SPORES 5 2 10 (RANK 2)

FILLS 1 TILE WITH SPORES FOR 2/4/6 TURNS, HIDING ITS CONTENTS FROM OTHER WIZARDS.

Figure 43. A comparison between the first (upper) and final (lower) spell locked visuals.

Another design problem was how to convey the changes affecting the spells between ranks. Usually there are one or two numerical values changing as the rank goes up, for example the cooldown decreasing or the number of targets increasing. At first the changes were indicated in the upgrade confirmation pop-up, with the current description on the top and the new description at the bottom. This was deemed very clunky, as the player would first have to click on the spell description to open the upgrade dialogue and then scan the descriptions for the differences.

This was solved by indicating the changes directly in the descriptions by separating the values between ranks with a “/” symbol, so a spell having its damage increased by 4 between ranks would have a “deals 4/8/12 damage” clause in the description text. This may create some confusion at first until the player gets the pattern behind it, but makes for a much smoother experience as all the information is displayed in the default view. The changes between ranks are also easy to spot as the x/x/x patterns stand out from the rest of the text.



UPGRADE SPELLS

CALL THE ARCANE ◉ 3/2/1 0 💎 MAX RANK!
GIVES YOU A 25% CHANGE THIS TURN TO FIND AN EXTRA 💎 WHEN YOU EXPLORE A TILE.

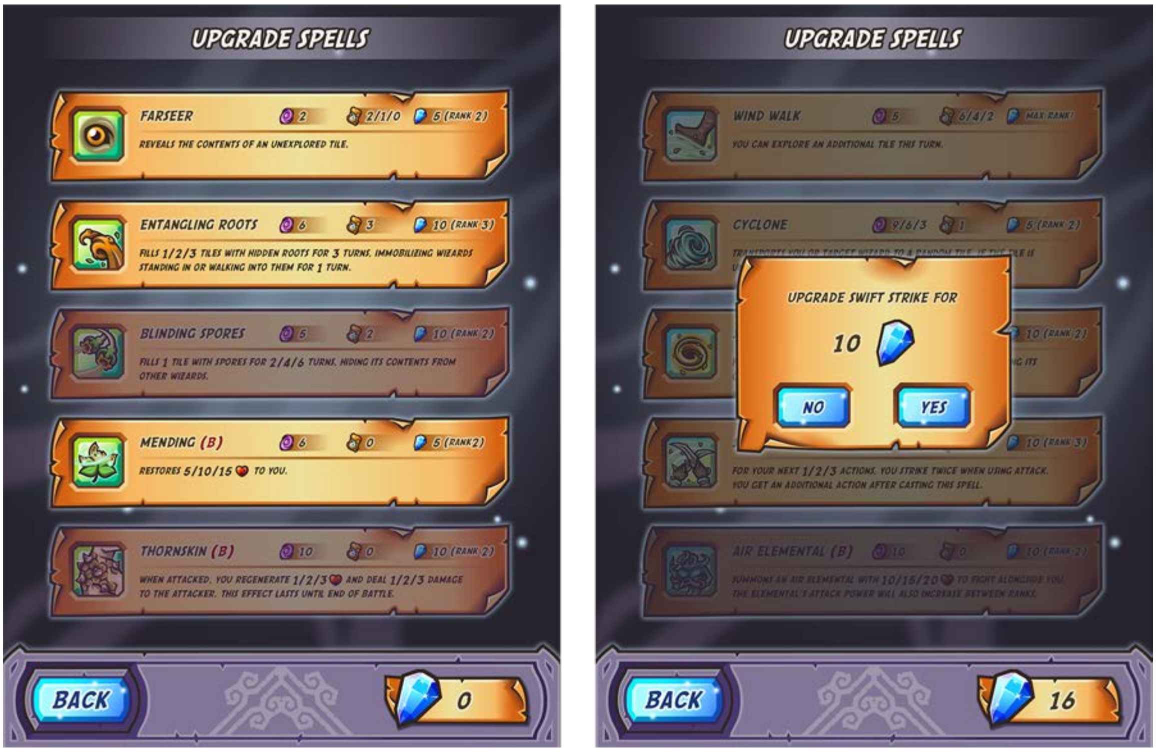
ENFEEBLE ◉ 2/4/6 2 💎 MAX RANK!
WEAKENS TARGET WIZARD'S MAGIC POWERS, MAKING THEM LOSE UP TO 4/6/8 ◉.

BEGUILING ILLUSION ◉ 8 4 💎 20 (RANK 3)
HAS A 33/66/100% CHANCE TO SEDUCE TARGET WIZARD ON A TILE NEXT TO YOU INTO GIVING AWAY A RANDOM ITEM.

ARCANE BOLTS (B) ◉ 3 0 💎 MAX RANK!
DEALS 4/8/12 DAMAGE, SPLIT AS YOU CHOOSE BETWEEN ONE OR SEVERAL ENEMIES.

SPELLBIND (B) ◉ 10 2 💎 10 (RANK 2)
CONTROLS AN ENEMY'S NEXT 1/2/3 ACTIONS. IF USED ON A WIZARD, THEY SKIP THEIR NEXT 2 ACTIONS TO SHAKE OFF THE SPELL.

Figure 44. The *Upgrade spells* menu screen.



UPGRADE SPELLS

FARSEER ◉ 2 2/1/0 💎 5 (RANK 2)
REVEALS THE CONTENTS OF AN UNEXPLORED TILE.

ENTANGLING ROOTS ◉ 6 3 💎 10 (RANK 3)
FILLS 1/2/3 TILES WITH HIDDEN ROOTS FOR 3 TURNS, IMMOBILIZING WIZARDS STANDING IN OR WALKING INTO THEM FOR 1 TURN.

BLINDING SPORES ◉ 5 2 💎 10 (RANK 2)
FILLS 1 TILE WITH SPORES FOR 2/4/6 TURNS, HIDING ITS CONTENTS FROM OTHER WIZARDS.

MENDING (B) ◉ 6 0 💎 5 (RANK2)
RESTORES 5/10/15 ❤ TO YOU.

THORNSKIN (B) ◉ 10 0 💎 10 (RANK 2)
WHEN ATTACKED, YOU REGENERATE 1/2/3 ❤ AND DEAL 1/2/3 DAMAGE TO THE ATTACKER. THIS EFFECT LASTS UNTIL END OF BATTLE.

UPGRADE SPELLS

WIND WALK ◉ 5 6/4/2 💎 MAX RANK!
YOU CAN EXPLORE AN ADDITIONAL TILE THIS TURN.

CYCLONE ◉ 9/6/3 1 💎 5 (RANK 2)
TRANSPORTS YOU OR TARGET WIZARD TO A RANDOM TILE. IF THE TILE IS

💎 10 (RANK 2)
NG ITS

💎 10 (RANK 3)
FOR YOUR NEXT 1/2/3 ACTIONS, YOU STRIKE TWICE WHEN USING ATTACK. YOU GET AN ADDITIONAL ACTION AFTER CASTING THIS SPELL.

AIR ELEMENTAL (B) ◉ 10 0 💎 10 (RANK 2)
SUMMONS AN AIR ELEMENTAL WITH 10/15/20 ❤ TO FIGHT ALONGSIDE YOU. THE ELEMENTAL'S ATTACK POWER WILL ALSO INCREASE BETWEEN RANKS.

UPGRADE SWIFT STRIKE FOR
10 💎

Figure 45. *Upgrade spells* menu screen with locked spells, and a spell about to be upgraded.

5.6 Use and equip items

During the course of a game of *Wizard Wars* the player will collect a variety of items, which will be displayed in the *Use and equip items* menu screen. In the screen the player can view the items and is able to either use or equip them depending on the item type. This screen uses a heavily elongated big pop-up parchment as a background for its UI elements.

The design process of this menu screen was unique from the other ones due to the fact that it had socket assets in a variety of sizes and several short description texts. The screen needed a reasonable amount of item slots and three equipment slots for a weapon, armor and accessory. Planning an alignment grid was essential for this screen, as without it the screen would have lacked structure and looked messy with seemingly randomly placed UI elements.

The initial alignment grid plan can be seen in Figure 46, along with the final grid. The final design follows the initial grid quite closely, although some of the alignments were left out as they were not feasible of include. The vertical alignment line for the right edges of the items slots and the middle equipment button would have pushed the design too far to the sides if included. As the use/equip button ended up being wider than the equipment slots, it and the right-most slot were put in a center alignment instead, while their respective description texts were still left-aligned to each other.

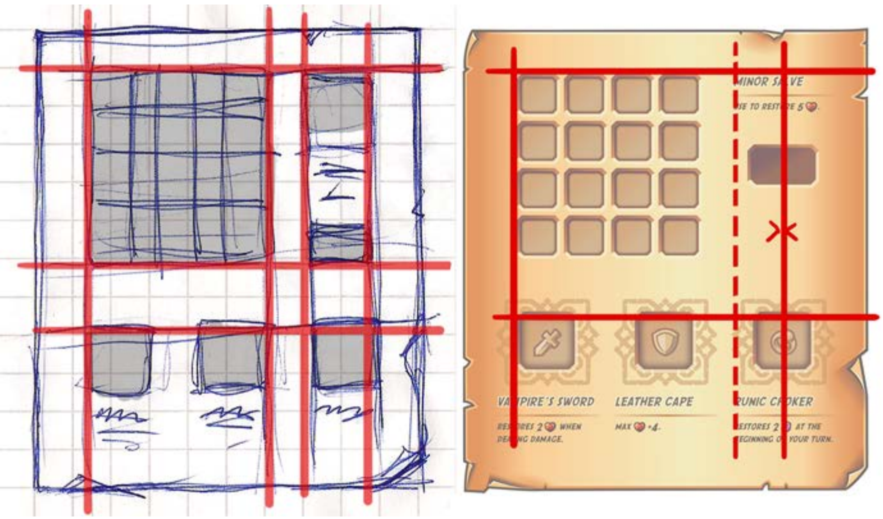


Figure 46. The alignment grid plan and the final alignment grid.

Because of the selected item description placed on the right side of the item slots, the vertical layout previously explored in the spell descriptions was revisited. As the item description texts were significantly shorter than the spell ones, it worked out much better. The layout and formatting of the text was taken directly from the spell descriptions, with the same vertical distance between the title and the rest of the text, and with a shortened version of the dividing line between them.

To let the player know which equipment slot responded to which type of equipment (weapon, armor or accessory) they needed to be differentiated from each other. First this was done the lazy way with text, but was later switched to simple icons which brought a sense of polish to the slots. Ornaments were also added around the equipment slots to make them more visually interesting.

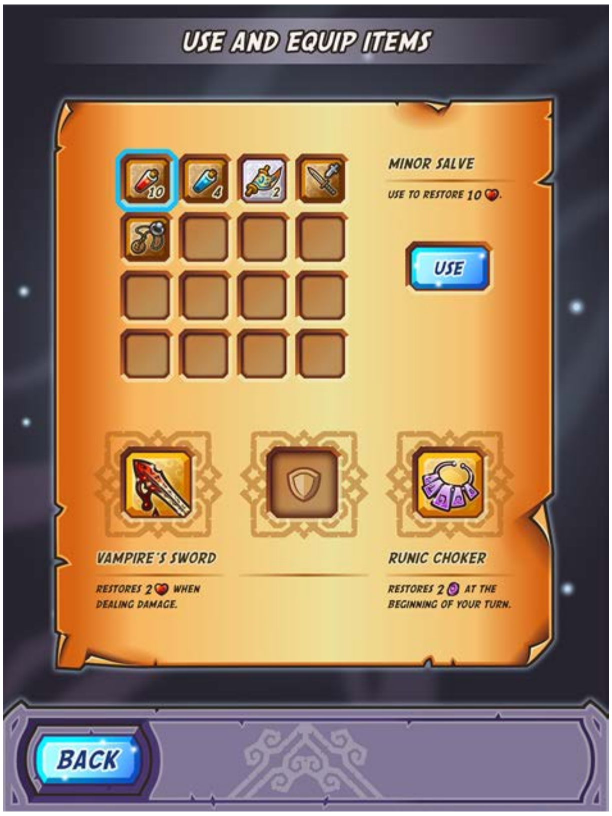


Figure 47. The *Use and equip items* menu screen.

In the end the *Use and equip items* screen ended up looking very subpar compared to some of the other screens, and might get revisited in later versions of *Wizard Wars*. It can be contributed largely to the big parchment, which looks boring with a too elongated pixel stretch. Building up the screen from smaller parchments so that each equipment slot would be its own freely floating element was considered, but there was concerns of making the screen too fragmented. The item icons also play a big part in the aesthetics of the screen, so making them more impressive could solve the issue to some degree. However, the screen has all the required functionality and is easy to navigate, which means it has still successfully reached the required usability levels.

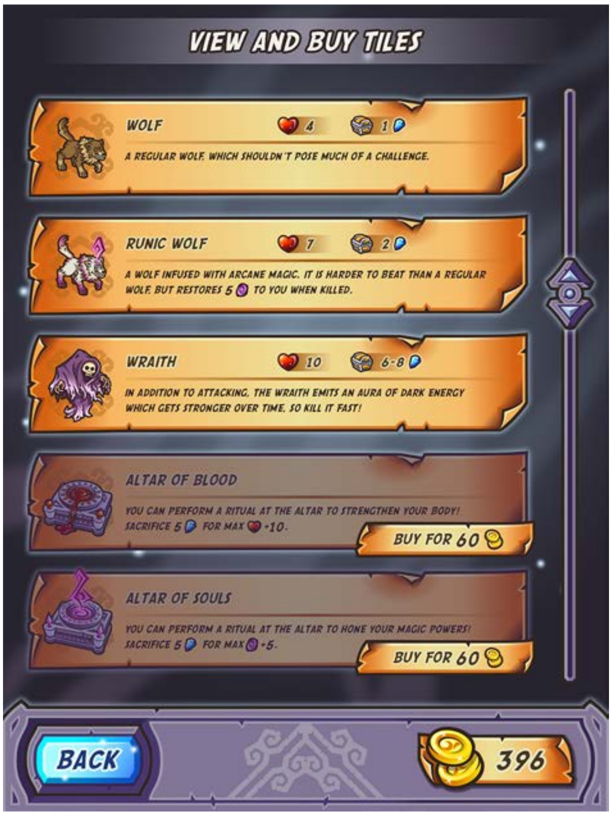
5.7 View and buy tiles

The *View and buy tiles* menu screen follows the same layout set by the *Upgrade spells* screen, with the exception of the tile descriptions being pushed towards the left edge to make room for a vertical slider used to scroll through the tiles list. By tapping on a locked tile the player either launches a pop-up to confirm the buy or the AppStore purchase interface, depending on if the tile is bought with in-game currency or through a micro-transaction.

The only new UI elements that needed to be added were the parchment for the in-game currency coins and a small buy prompt that is put on top of the locked tiles. Creating both elements was a straightforward process, as they were made out of already existing assets.

As a happy coincidence, the sprite size used for the in-game view fit well into the tile descriptions. This removed the need of coming up with a different scaling percentage, and it brings a nice sense of polish when the sprite seen in the description is exactly the same as in the actual game. As each sprite is saved in the same 200 x 250 pixel size their positioning in relation to the description box will always be the same, and thus the tile description assembly happening in the code can be completely automated.

As the information next to the tile name varies depending on the tile type (monster, item or event) each small icon asset and the belonging text is added as a separate element on top of the parchment, so they can be easily switched or left out. This method of assembling UI elements is much more flexible than having to make a parchment version of every possibly icon combo, not to mention the difference in the game file size between these two solutions.



A REGULAR WOLF, WHICH SHOULDN'T POSE MUCH OF A CHALLENGE.

A WOLF INFUSED WITH ARCANE MAGIC. IT IS HARDER TO BEAT THAN A REGULAR WOLF, BUT RESTORES 5 ● TO YOU WHEN KILLED.

IN ADDITION TO ATTACKING, THE WRAITH EMITS AN AURA OF DARK ENERGY WHICH GETS STRONGER OVER TIME, SO KILL IT FAST!

YOU CAN PERFORM A RITUAL AT THE ALTAR TO STRENGTHEN YOUR BODY! SACRIFICE 5 ● FOR MAX ❤ +10.

YOU CAN PERFORM A RITUAL AT THE ALTAR TO HONE YOUR MAGIC POWERS! SACRIFICE 5 ● FOR MAX ● +5.

Figure 48. The *view and buy tiles* menu screen.

6 CONCLUSIONS

The development of *Wizard Wars* has been an ongoing project since 2012, and has been more or less done on the side of fulltime work, hence the long development time. It started as a project which primary purpose was to provide a testing platform for many game design and development methods and ideas, but has since evolved into a full game. In addition to designing the UI, my main goal was to see how far I could push my vector graphics skills with the character and environmental assets. The main non-art focuses of this project were the porting of a traditional strategy game concept to a tablet platform; the testing of touchscreen control schemes; and the creation and balancing of the playstyle and spells for each wizard. Each of these processes would have been very interesting to document more thoroughly and would have made for a great thesis subject. However, in the end I opted for the UI design for the reasons that it was a process I had full control over, and it was the most relevant one for my career.

From a visual standpoint I would summarize the UI design process as starting with something way too elaborate, which was then gradually simplified into a clean and consistent visual style. In retrospective, the visual design broke almost every principle of UI design in its earlier iterations. Poor utilization of the screen estate, no visual hierarchy or structure, too many fonts and font sizes… For this reason, the beginning of the design process was very frustrating for me, as I kept creating designs that just did not work. I genuinely thought I was following design principles, when in reality I was not.

I think the main factor holding me back was that as a game artist with no real UI design experience, I was designing full-screen images instead of UI elements. Too much thought was put into creating a theme for the whole screen instead of focusing on how the information was to be displayed in individual elements, which would then be used to build up the UI as a whole. As a summary, I kept sacrificing way too much usability in favor of an impressive design.

The biggest success of the thesis is definitely the fact that my UI design skills have evolved significantly. Nowadays I think about the design process in a completely different way, being able to deconstruct the UI into smaller elements to tackle one at a time. Adopting useful work methods, such as starting the UI design process by creating a flowchart instead of going straight for the layout sketches, has also proved to be invaluable.

During the thesis project I confirmed the notion that most of the UI design should be done together with the game designer, or whoever is in charge of the user experience and game mechanics. Questions about the control scheme, user flow and gameplay that had not been thought of during the regular game design surfaced all the time. It was surprising for me to notice how much hand in hand gameplay design and UI design go. Many gameplay improvements were thought of during the UI design process and vice versa. The whole gameplay simplification process described briefly in the beginning of Chapter 4 was a direct result of the control scheme limitations of the touchscreen.

One important fact to keep in mind is that I had much more freedom during this UI design process than I would ever have in a "real-world" project for an actual game company. As this thesis project did not have a set deadline, the only limitations for me was the requirements established by the game design and the amount of time I was able to set aside. In a "real" game project there would be many demands and restrictions set on the UI design both from inside the project in the form of time and resource limitations, and from outside in the form of the product owner's and producers' wishes and demands. The fact that this thesis project was not done for a client also means it is harder to measure the actual quality of the UI, as there was no product owner who could say how well the design met their expectations and needs.

Although the goals of this thesis have been met – to design the visual style and create all the art assets for the UI of *Wizards Wars* – the UI design is still not completed. Animations and special effects have to be designed and added to bring the polish and visual impressiveness expected of an industry-standard mobile game. As the UI elements have been designed with these animations in mind, I don't think it will be a difficult undertaking, although it may be a time-consuming one.

Also, the current wish list for *Wizard Wars* is quite long, with many features that might get explored further in future versions of the game. One of the more interesting ones is the map assembly, which lets the players choose what tiles to add to their current game. This would require an extra screen after the *Choose a wizard* menu, as well as some new UI assets, for example a condensed tile description with just the sprite and name, and a way to show and/or regulate the balance between the three tile types (monster, item and event). Another possible feature is letting the players adjust the number of tiles in the current game, to enable both quicker and longer-lasting games. This would require the crystal guardian to have scaling stats, so it is weaker in the quicker games and stronger in the longer-lasting ones. I expect this feature would not be too hard to add as long as the guardian's scaling is balanced properly. The last bigger feature in the wish list is to allow 2-4 players instead of just the default three in the play with friends game mode. This feature is very appealing to include for flexibility and player enjoyment reasons, but it would require considerable amounts of testing to see if it is even feasible.

After I finish the animations and special effects for the UI elements, my work on *Wizard Wars* is done, at least for now. The first version of the game is ready to be launched on iPad except for some programming for the in-game view, and a more extensive testing round to fine tune the balancing of the wizards and tiles, and to polish the gaming experience in general. From there on out the future of the game is mostly decided by if it will get any traction or not. Any potential updates should not be too demanding art-wise, so I expect to be able to continue contributing to the project on the side of my other commitments.

REFERENCES

Books and published works:

Cao, J., Zieba, K., Stryjewski, K. & Ellis, M. (2015). Consistency in UI Design: Creativity Without Confusion. UXPin Inc.

Cooper, A., Reimann, R. & Cronin, D. (2007). About Face 3: The Essentials of Interaction Design. Indianapolis, IN, USA: Wiley Publishing Inc.

Fox, B. (2004). Game Interface Design. Boston, MA, USA: Course Technology/Cengage Learning.

Galitz, W. O. (2007). The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third Edition. Indianapolis, IN, USA: Wiley Publishing Inc.

Lidwell, W., Holden, K. & Butler, J. (2010). Universal principles of design. Beverly, MA, USA: Rockport Publishers Inc.

Novak, J. (2012). Game Development Essentials: An Introduction, Third Edition. Clifton Park, NY, USA: Delmar, Cengage Learning.

Saunders, K. & Novak, J. (2013). Game development essentials: Game interface design, Second Edition. Clifton Park, NY, USA: Delmar, Cengage Learning.

Sears, A. & Jacko, J. A. (2002). The Human-Computer Interaction Handbook. In Pagulayan, R. J., Keeker, K. & Wixon, D. & Romero, R. L. & Fuller, T. (Eds.), User-Centered Design in Games (pp. 883-906). Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Web articles and blogs:

Carmichael, S. (2012). Mobile games winning the war for your attention. venturebeat.com/2012/11/07/console-vs-mobile-gamers/ (Read 15.3.2014)

Glinert, E. (2009). Upping Your Game's Usability. gamasutra.com/view/feature/132499/upping_your_games_usability.php (Read 16.3.2014)

Laitinen, S. (2005). Better Games Through Usability Evaluation and Testing. gamasutra.com/view/feature/2333/better_games_through_usability_.php (Read 15.3.2014)

Llanos, S. C. & Jørgensen, K. (2011). Do Players Prefer Integrated User Interfaces? A Qualitative Study of Game UI Design Issues. digra.org/wp-content/uploads/digital-library/11313.34398.pdf (Read 9.3.2014)

Morrison, B. (2013). Using Psychological Principles for Great User Interfaces. gamedev.net/page/resources/_/creative/game-design/using-psychological-principles-for-great-user-interfaces-r3397 (Read 17.3.2014)

Quintans, D. (2013). Game UI by Example: A Crash Course in the Good and the Bad. gamedev.tutsplus.com/tutorials/aesthetics/game-ui-by-example-a-crash-course-in-the-good-and-bad/ (Read 14.4.2014)

Russell, D. (2011). Video game user interface design: Diegesis theory. devmag.org.za/2011/02/02/video-game-user-interface-design-diegesis-theory/ (Read 16.3.2014)

Stonehouse, A. (2010). User interface design in video games. thewanderlust.net/blog/2010/03/29/user-interface-design-in-video-games/ (Read 11.2.2014)

Williams, S. (n.d). What's the Difference Between a Hue, Tint, Shade and Tone? color-wheel-artist.com/hue.html (Read 10.08.2015)

Wilson, G. (2006). Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design.
gamasutra.com/view/feature/2538/off_with_their_huds_rethinking_.php (Read 17.5.2014)

Zammitto, V. (2008). Visualization Techniques in Video Games.
bcs.org/upload/pdf/ewic_eva08_paper30.pdf (Read 14.4.2014)

Image sources:

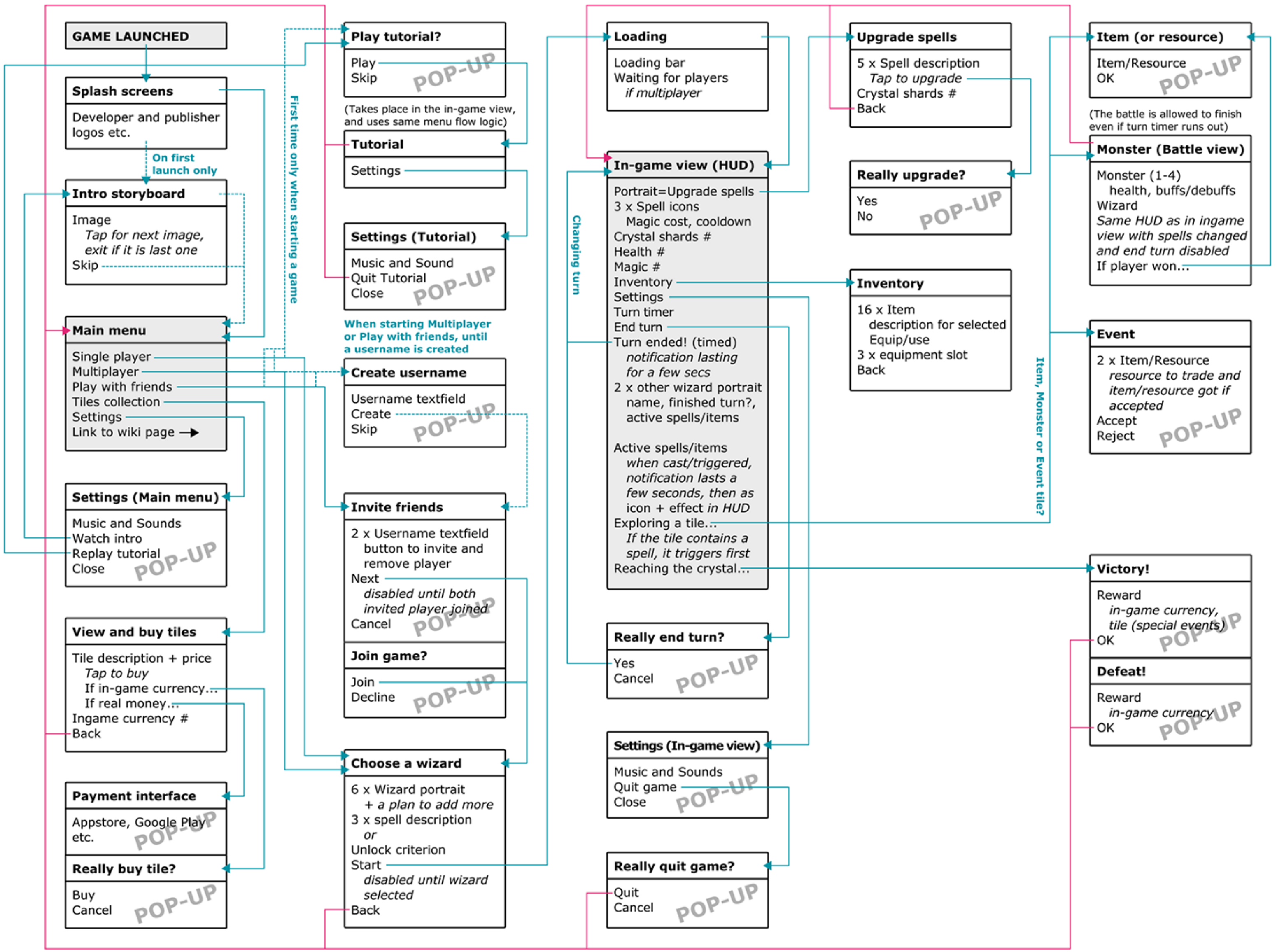
Figure 5. Screenshots from

Final Fantasy	finalfantasy.wikia.com/wiki/Menu_(Final_Fantasy)
Final Fantasy XIII	gametremor.com/2010/12/12/final-fantasy-xiii

Figure 24. Modified from http://www.1001fonts.com/sf-comic-script-font.html#gallery

APPENDIX 1

FLOWCHART


GAME LAUNCHED
Splash screens
Developer and publisher logos etc.
On first launch only
Intro storyboard
Image
Tap for next image, exit if it is last one
Skip
Main menu
Single player
Multiplayer
Play with friends
Tiles collection
Settings
Link to wiki page →
Settings (Main menu)
Music and Sounds
Watch intro
Replay tutorial
Close
POP-UP
View and buy tiles
Tile description + price
Tap to buy
If in-game currency…
If real money…
Ingame currency #
Back
Payment interface
Appstore, Google Play etc.
POP-UP
Really buy tile?
Buy
Cancel
POP-UP
First time only when starting a game
Play tutorial?
Play
Skip
POP-UP
(Takes place in the in-game view, and uses same menu flow logic)
Tutorial
Settings
Settings (Tutorial)
Music and Sound
Quit Tutorial
Close
POP-UP
When starting Multiplayer or Play with friends, until a username is created
Create username
Username textfield
Create
Skip
POP-UP
Invite friends
2 x Username textfield button to invite and remove player
Next
disabled until both invited player joined
Cancel
POP-UP
Join game?
Join
Decline
POP-UP
Choose a wizard
6 x Wizard portrait
+ a plan to add more
3 x spell description
or
Unlock criterion
Start
disabled until wizard selected
Back
Changing turn
Loading
Loading bar
Waiting for players
if multiplayer
In-game view (HUD)
Portrait=Upgrade spells
3 x Spell icons
Magic cost, cooldown
Crystal shards #
Health #
Magic #
Inventory
Settings
Turn timer
End turn
Turn ended! (timed)
notification lasting for a few secs
2 x other wizard portrait
name, finished turn?, active spells/items
Active spells/items
when cast/triggered, notification lasts a few seconds, then as icon + effect in HUD
Exploring a tile…
If the tile contains a spell, it triggers first
Reaching the crystal…
Really end turn?
Yes
Cancel
POP-UP
Settings (In-game view)
Music and Sounds
Quit game
Close
POP-UP
Really quit game?
Quit
Cancel
POP-UP
Upgrade spells
5 x Spell description
Tap to upgrade
Crystal shards #
Back
Really upgrade?
Yes
No
POP-UP
Inventory
16 x Item
description for selected
Equip/use
3 x equipment slot
Back
Item, Monster or Event tile?
Item (or resource)
Item/Resource
OK
POP-UP
(The battle is allowed to finish even if turn timer runs out)
Monster (Battle view)
Monster (1-4)
health, buffs/debuffs
Wizard
Same HUD as in ingame view with spells changed and end turn disabled
If player won…
Event
2 x Item/Resource
resource to trade and item/resource got if accepted
Accept
Reject
POP-UP
Victory!
Reward
in-game currency, tile (special events)
OK
POP-UP
Defeat!
Reward
in-game currency
OK
POP-UP