

Heuristics GDD (=^•_•^=)

[Github](#)

[Drive Folder](#)

[FigJam Board](#)

[Game Script](#)

[Figma Wireframes](#)

Learning Design Overview

Learning Goals

Overall Goal: Players will be able to apply design heuristics to evaluate the user experience of video games and propose more effective alternatives to UX and UI related to the 6 heuristic categories shown in game.

Knowledge/Skills Needed to Play:

- Basic understanding of RPG-like video games and their UI/UX elements
 - Basic RPG vocabulary and understanding of some mechanics (NPC interactions, battles, movement, etc.)
- Basic understanding of what UX is, the goals of UX, and UX-related terms
 - Designing based on user needs, creating an enjoyable experience, feedback, design terminology (typography, spacing, hue, etc.)

Knowledge Goals

- Performance Goals
 - Define Nielsen's 10 Usability Heuristics
 - Define Game Design Heuristics from Hochleitner's Framework
 - Understand how these heuristics fit into our broader heuristic categories
 - Identify where game UX examples violate design heuristics (1+ at once)
 - Compare and contrast different solutions to bad examples of game UX
 - Select the best solutions to improve bad UX examples
- Cognitive Goals
 - Understand how usability errors can affect a player's experience in the context of a video game

- Learn how to procedurally go through an interface to identify usability errors
- Develop a disposition for critically evaluating interfaces with design heuristics in mind
- Transformational Goals
 - Player will come away with a better eye/intuition for effective UX design in terms of heuristics

Potential Transfer

- Potential close transfer to evaluating video game UX and far transfer to being able to use design heuristics to evaluate general UX as a whole (i.e. UX for web)

Learning Domain: User Experience Heuristics

Integrated Heuristics (Reference: [Nielsen's Heuristics](#), [Game Heuristic Framework](#))

1. Consistency / Standards

- a. Follow platform and industry conventions to reduce cognitive load
- b. Heuristics
 - i. N4 – Consistency and standards
 - ii. 12.1 – The interface is consistent in control, color, typography, and dialogue design and as non-intrusive as possible
 - iii. Designers should avoid [out-of-loop failures](#) (i.e. unintended failures that detract from the player experience) such as:
 - Not being able to differentiate enemies and allies because of colorblindness
 - Not being able to read the text on the screen

2. Information Architecture

- a. Organize, structure, and label content in a way that helps users easily find information and complete tasks
- b. Heuristics
 - i. 12.2 – The menu is intuitive and the meanings are obvious and perceived as a part of the game

3. User Control (Including Error Prevention and Recovery)

- a. Allow users to remain in control of the system by showing ways to exit current interactions (cancel, undo, redo, etc.)
- b. Heuristics
 - i. N3 – User control and freedom
 - ii. N5 – Error prevention

iii. N9 – Help users recognize, diagnose, and recover from errors

4. Modality of Communication (Including Labeling)

- a. Use words, phrases, and concepts that are familiar to the user, instead of jargon, and clearly label any information required to use the design

- b. Heuristics

- i. N2 – Match between system and real world

5. Efficient Use of Space

- a. Interfaces should focus on the essentials and that all visual elements support the users' primary goals

- b. Heuristics

- i. 12.3 – The visual representation allows the user to have a clear, unobstructed view of the area and all visual information that is tied to the location

- ii. N8 – Aesthetic and minimalist design

6. Availability of Crucial Information

- a. System should communicate critical info with the user and provide appropriate feedback, make elements, actions, and options visible at all times, and provide help on how to complete tasks

- b. Heuristics

- i. 12.4 – Relevant information is displayed and the critical information stands out. Irrelevant information is left out. The user is provided enough information to recognize their status and make proper decisions.

- ii. N1 – Visibility of system status

- iii. N6 – Recognition rather than recall

- iv. N10 – Help and documentation

- v. Player not knowing what they should do next from confusing quest texts or forgetting information that hasn't been written down anywhere

Heuristic 1: Consistency & Standards

- Definition

- The interface is consistent in control, color, typography, and dialogue design and is as non-intrusive as possible.

- Being consistency and following standards improves learnability of system

- Learning Objectives

- Understand importance of consistent design and difficulties that arise without consistency – without consistency, users have to spend extra time and effort to learn how to perform actions
- Understand which elements of a system benefit from having consistent design
- **Types of Consistency**
 - Internal: Use same design system within a product/family
 - External: Use established conventions
- **Layers of Consistency**
 - Visual: symbols, iconography, and imagery
 - Page/Button: layouts
 - Content: same tone and voice
- **Common Mistakes**
 - Same element in different places
 - Using different colors for same elements
 - Using made-up words for navigation
 - Reassigning function to non-standard function (ex: Press B to jump instead of A)
- **References**
 - [Consistency and Standards](#)
 - [4th Usability Heuristic](#)

Heuristic 2: Information Architecture

- **Definition**
 - To organize, structure, and label content in a way that helps users easily find information and complete tasks
- **Similar to a...**
 - Skeleton
 - Blueprint
 - Bento Box
 - “9 Squares. Anyone know the intention of the specific placement? Is it about order? Note the center square: devoted to protein. Corners are veg and dessert, sides are rice dishes.... Water, land, plant, animal. Protein is most rare, valuable, special. See the result but also understand the intention.
- **Learning Objectives**
 - Select the best organizational scheme for a given data set
 - Use IA as a blueprint for visual design decisions
- **Understand the Information Ecology**
 - Users: Audience, tasks, needs, experience

- Content: Objectives, data types, existing structure, ownership
- Context: Business goals, funding, politics, culture, technology, resources, constraints
- **Questions to Ask**
 - How does the user think about the content?
 - How does the user interact with the content?
 - What language and terminology does the user use to refer to the content?
 - What is the users' goal?
- **Main components of IA**
 - Organization schemes and structures
 - Hierarchical, Sequential, Matrix, Alphabetical, Chronological, Topic
 - 5 Hat Racks (LATCH)
 - Location, Alphabetical, Time, Category, Hierarchy
 - Choosing one main rack changes the way audience perceives information
 - Labeling systems
 - Navigation systems
 - Structural, Associate, Utility
 - Search systems
- **Common Mistakes**
 - No structure
 - No integration between search and structure
 - Too much categorization
 - Invisible navigation (ex: hamburger menu)
 - Inconsistent navigation
 - Made-up menu options
 - ACNH Inventory: No sorting or searching, non-specific labels like “misc” and “other” for house/DIY inventories



- **References**

- [Information Architecture](#)
- [IA Basics for Designers](#)
- [Beginners Guide to IA](#)
- [IxDS1 IA Lecture](#)
- [Basics of IA](#)
- [Animal Crossing and IA](#)

Heuristic 3: User Control & Freedom

- **Definition**

- Allow users to remain in control of the system by showing ways to exit current interactions (cancel, undo, redo, etc.)

- **Learning Objectives**

- Identify types of UI interactions that make users feel trapped or dissatisfied
- Understand features that support user control

- **Benefits of User Control**

- The ability to get out of trouble encourages exploration
- This facilitates learning/feature discovery, instead of making the user afraid or anxious of making wrong moves

- **Examples of Control**

- Back, Cancel, Close, Undo
- Adding/Saving Items
- Moving to different steps
- Recovering history
- Adjusting quantity
- Confirming decisions (“Are you sure?”)
- Showing game goal at the very beginning/throughout the game
- Retrying levels

- **Examples of Freedom**

- Having multiple pathways to do something
- Baba is You – player is able to change game rules
- Having toggle options for installation/settings

- **Common Mistakes**

- Hidden/Difficult to discover signs
- Extreme punishment for certain actions

- **References**

- [User Control NNGroup](#)

Learning Mechanisms

Induction/Refinement

- *Pretraining*
 - Examples of effective and ineffective use of heuristics will be shown through the player through tutorials and/or dialogue scenes with characters before players are asked to apply them in battle
- *Linking*
 - Players will be introduced to one heuristic at a time, then asked to connect this learning together in harder battles where the UX design presented will have multiple violations — simultaneous application of different concepts
- *Immediate Feedback*
 - When players select a UX solution based on the options we give them, it will immediately be implemented on the screen — feedback will be given through a battle health system, and the user will be able to experience playing through the new UX they suggested; by extension, they will realize whether or not it is an improvement
- *Interleaving*
 - Each battle will show players different heuristic violations (both individually and multiple at the same time). Additionally, practicing evaluation, comparison, and redesign skills will be spread out throughout both battle and overworld gameplay.
- *Application*
 - Battle interactions will require players to identify a heuristic to evaluate specific UX examples and ask them to propose a solution based on a few possible choices
- *Variability*
 - Battles will have different examples impacting different aspect of player UX (game mechanisms, accessibility, readability, ease of understanding) that can help the player abstract these concepts to UX examples outside of our game
- *Metacognition*
 - Players can reflect on their solutions after they make a choice and self-correct in following battles if they make the wrong decision

Targeted Aspects of Player Experience Design

- Immersion – Fantasy + Story
 - Narrative arc, characters have a motivation and personality and are appealing to the player

- Have some kind of interactive storyline to draw the player in
 - Overall situation → Good cats trying to build an interface, bad cats trying to prevent them by wrecking the UI. Player must help good cats fix their interface by learning and applying UIUX principles
- Learn about design principle and better understand it through application
 - Core Gameplay Loop → Interact with the environment around you and get exposed to the different heuristics and to identify good UI/UX when given several choices, the things you learn through this will be applied through battling the enemy cats (inner loop)
 - Inner loop → Apply the heuristics you learn about through battles, where the enemy can wreck your UI/UX, and you can fix it by identifying the UI/UX principle/heuristic, applying further damage to your opponent

Knowledge Elicitation

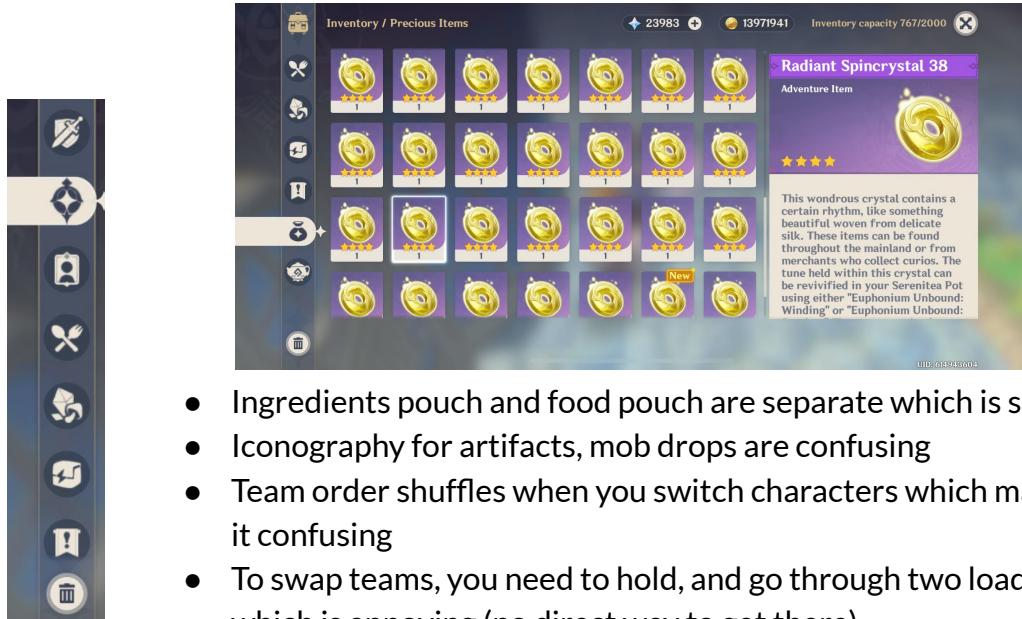
Research Questions

- How easy is it to apply our heuristics to existing game interfaces?
- Which heuristics are most important?
- Which heuristics result in the most improvements?
- How difficult is it for users to propose solutions?
- Generate ideas about how we can “break” UI in the game.

Cognitive Task Analysis

- Method
 - Have user choose their favorite game, a game that they are currently playing, or a game they think has bad UX
 - Ask them to evaluate the game interface
 - Ask them to propose potential solutions
- Participants
 - AL (Novice)
 - Genshin Impact (Mobile)
 - Played for 2 years
 - EX (Sort of novice?? has taken half an HCI class [DHCS])
 - Arknights (Mobile)
 - Played for almost one year
 - CM (Expert, 8 years industry experience in UX Design)
 - Diablo Immortal

- Playing since release
- Participant Responses
 - AL
 - Good
 - Menu buttons are non-invasive, don't obstruct world vision
 - Button placements are intuitive - when you go forward you move forward
 - Spiral Abyss - used to be that you could only see monsters in one menu so you would have to flip back and forth when building ur team, but now it's integrated in the team formation menu
 - You can label pins on the world map
 - Nice that you can jump regions throughout a menu instead of having to scroll to it on the world map
 - Gacha history - used to be in multiples of 6 but now it's 5 so it's easier to calculate pity (multiples of 10)
 - Bad/Solutions
 - Inventory is disorganized, too many materials that are redundant (don't need to see all of them individually)

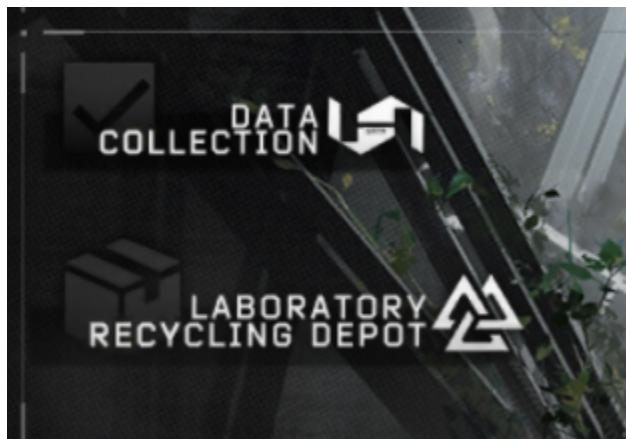


- Quest menu icons are stupid - all of them are exclamation points with some design, so you don't know which one it is until you click on it
 - On PC it doesn't tell you what the key binds are, so you just have to guess
 - For domains, all of them have the same base rewards, so it's annoying that you have to scroll through the rewards list to get to see the special reward
 - The quest names are non-descriptive (fight childe vs. enter the golden house)
 - Volume for clicking buttons at max
 - More specific about currency meant - non descriptive
 - Would like Confirmation button for high value items before you buy them
 - Wish I could change button sizes
- EX
 - Good
 - The level map is also pretty intuitive (sequential left to right to progress through story with some branches for stages)
 - Main page has pretty good layout, like settings / friends / notices are in the corner where you expect them to be, while more important things that get you to actually engage in the game are displayed more for the user. also a good amount of blank space where ur character is so the main screen doesn't feel too cluttered
 - There are back buttons and level exit buttons (3)
 - although sometimes the back buttons dont lead you exactly back to where you were, like they might make you go back to the home page even though you were previously looking at something in the base
 - displays ur sanity and gacha currency (very important) (6)
 - corresponding stores display the currency needed for that store
 - actual game ui also does a pretty good job of displaying relevant info. ex they order the operators by dp at the bottom, display the range of operators when ur placing them down, show which operators can and cant be deployed

- recently they made a thing where the bosses' health bar is bigger than regular enemies so that you can actually tell that it's a boss

- Bad/Solutions

- Some of the wording is a little jargon (like annihilation, contingency contract) like it took me a while to learn what contingency contract was and how to play it. also the instructions for learning how to play is and sss (the more advanced game modes) were less clear. probably could implement some more hand-holdy tutorials
- Arknights has a good amount of currencies (lmd, yellow rocks, orundum, green certs, yellow certs, etc) which makes it confusing for new players because it's unclear which currencies are easily obtainable so it's hard to plan ur finances, but i guess this is a common thing in most gacha games
 - potential solution: you can display a thing on how to obtain this specific currency. ik they do this for mats (like which level you can get the mat from) so they could just do the same thing but for each currency
- in the event ui sometimes it's unclear which button takes you to the stage / store / achievements page, potential solution just make a little icon by the button (i mean there are little icons but make it more obvious, like "box symbol" doesnt really correspond to store in my braiin)



- CM

- Good

- IA of menu elements: codex and inventory on HUD, warband, skills, etc. in a hamburger menu
- Plenty of feedback when in combat

- Visual effects never distract player, only augments the experience
 - Prevents user error when salvaging – shows a confirmation message when salvaging something with a better combat rating than what you currently have
 - Screen turning red when you’re low brings it up in the IA and prevents users from having to look up at the health bar, which is important because this information means life or death
 - Bad / Solutions
 - Really easy to misclick ultimate; if you’re holding it down and want to cancel, the cancel hot zone is very small → the opposite of user control
 - No strong contrast between the player and enemy health bars
 - Especially important for PVP
 - No error prevention for deleting mail that gives rewards or taking portals back to a start of a dungeon
 - Can beat a boss and instantly accidentally take the exit portal that spawns
 - Using services takes too many clicks despite how often you have to use them
 - Limit the number of clicks or at least make the services button larger
 - Missing indicators
 - Really hard to tell legendary and rare items apart, same for legendary crests and eternal legendary crests
 - Default settings are not ideal (especially auto-pickup)
 - Analysis
 - AL
 - Wasn’t hard for participant to propose solutions
 - Many of the things participant identified fall under our heuristics, despite participant not having any background in UX
 - EX
 - Easy for participant to discern both good/bad parts of the game UI and why they were good/bad
-

Game Design Overview

Game Summary/Narrative

- You are inside a cat-themed game/virtual world, and all the NPCs are very aware they are in the game as well
 - They understand that they have UI elements and they actively rely on them to do things
 - You can go into stores / interact with NPCs to find out more about UI in the game and concepts of Nielson's
- The cat world in this game is falling apart because hackers/bugs are taking the form of enemy characters/bad cats and muddling all the UI/UX in the game to lead to a worse user experience

Core Gameplay Loop

- Pretraining: in the town before battle interactions
 - Go around to shops/npcs and are tasked with “fixing” their UI stuff in exchange for a reward
 - That reward will help you later in battle when you have to deal with getting UI fixed and picking a good UI
 - The tasks can be to identify a good UI out of a few choices (which mirrors the task they would do in the interactions)
- Post Battle UX Analysis
 - Evaluate UX at the end of the battle and identify and punish player (less xp, money?) for anything sub-optimal so they can fix it on the next battle
 - To ensure players don't fixate on sub-optimal UI that is still usable but not the best experience

Player's Goals

1. Win a battle against the Boss Hacker Cat to save the world by creating a great experience for all players
2. Win battles against the Boss's Minions to fix specific UX elements within the world
3. Rectify UX failures during battle using the special abilities of your heuristicats

Heuristicats (Allies)

Choose 3 to bring into battle

1. Consistency / Standards (N4, 12.1, 12.5)
2. Information Architecture (12.2)
1. User Control (N3, N5, N9)
 - a. Including Error Prevention and Recovery
2. Modality of Communication (N2)
 - a. Including Labeling
3. Efficient Use of Space (12.3, N8)
4. Availability of Crucial Information (12.4, N1, N6, N10)

Signature Ability

SOLVING A UX PROBLEM DOES BIG DAMAGE OR SOMETHING POSITIVE AND BIG (BREAK?)
 (solving means getting through all 4 steps)

1. First check if there are deficits in the heuristic when a specific heuristic ability is used
2. If there is, they're then provided with a multiple choice of elements which are affected by the heuristic
3. If they choose correctly, there is multiple choice of design standards which could be being violated
4. Finally, there is a multiple choice of possible solutions, which will be applied to your interface regardless of choice

Enemy Cats



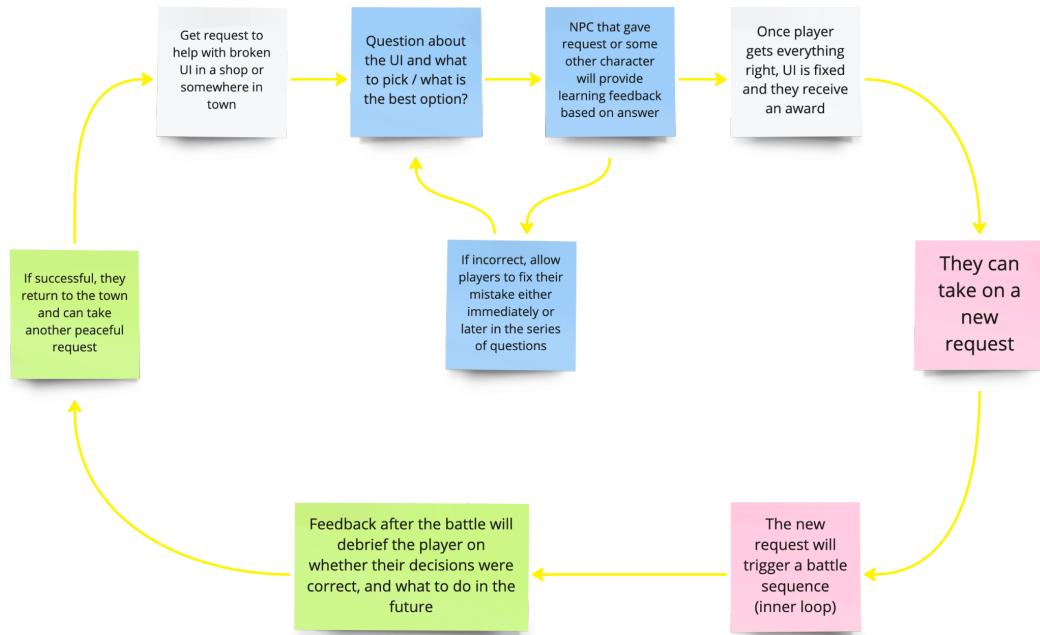
UI Altering Moves

- **General Policy:** Try to use visual elements to convey information rather than text, so that UX matters more rather than content
- **Consistency & Standards**
 - Color coding of mana → skills that use mana don't match
 - Use to show whether a move is affecting allies or enemies
 - Can accidentally heal enemy if this is not consistent
 - Use to show whether a move costs mana or health

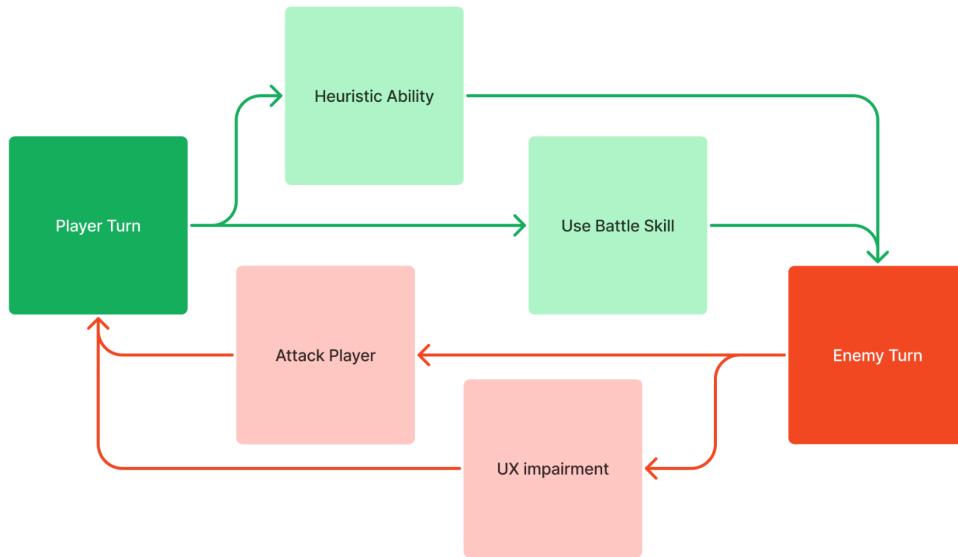
- Could accidentally choose a risky play when trying to choose a safe play if this is off
 - Should still be color coded, but the colors don't match the color of health and mana
- Inconsistency → changing order of action options for every character
- Input does not match the output – e.g. movement doesn't work as expected or moving around menu does not work either
 - Inverse menu navigation → pressing up moves down an element, vice versa
- Placing the pause button in the middle of the screen
- **Information Architecture**
 - Trying to use items results in infinite / very many folder pop ups
 - Bad information organization
 - Organizing UI in such a way that user has to move their mouse around the entire screen to complete actions
 - Making similar sections far from each other (ex: character-related information are in two completely different parts of the screen/menu)
 - Emphasizing the wrong button (ex: having green no and red yes, putting button border around "nevermind" but leaving "accept" only in text, etc)
- **User Control, Error Prevention, and Recovery**
 - Deny user freedom (for each button, have a new pop up that asks them if they want to do something? Y/N, instead of allowing players to just go and click buttons)
 - Remove pause and quit buttons
 - Add ability for user to rewind turns or prevent them from using not-effective skills → take this away as an attack
- **Modality of Communication**
 - Replacing all the dialogue with SAT vocabulary equivalents
 - Ex: "I managed some extra training. I must stay sharp to protect my comrades." → "I have undertaken supplementary instruction, for I must remain astute in safeguarding my fellow companions."
 - Replacing all the dialogue with unreadable cat puns
 - Sword → Meowrd
 - Changing health bar to number (no max value or only the max value)
- **Efficient Use of Space**
 - Extremely cluttered UI (see: [WoW](#)) – include information about cats, items, story, a chatlog, weapons, map, etc. all on the battle screen
- **Availability of Crucial Information**
 - Take away max health indicator

- Delayed response / no response – don't know your health until it drops to 0
- No health going down animation → discrete levels
- Removing hover indicators
- Removing move descriptions
- No way of knowing how to change skills
- Taking away mana number (just showing the bar)

Overworld Interactions



Battle System



Heuristic Ability

1. Identify User Need
2. Identify UX standard being violated
3. Identify a solution

Art Style

- [Pinterest Board](#)
- Style: Isometric, Retro





Concept Art

- Environment: Onsen Cat Town



- Characters: Cats



- Minigames



Playtesting Notes

- arrows not always communicating in bath
- people want to drop cats in the bucket not water
 - ☒ ~~Expand the collider for where players can drop cats~~
- slow progression important —> start with much simpler interface and system
- hover indicator for what heuristic a cat's heuristic ability belongs to
- turn order is useless —> give it gameplay value
- black out screen when UI is changed so people know there is a change
- animate what is being changed (ie: take skill button) in earlier battles, later on it just goes black and people have to figure out
- make the back button disappearing when you have no mana to use battles a lose condition
 - “Not giving users a way to backtrack on and recover from important interactions may lead them to get stuck”
- not thinking about physical magical
- another learning goal —> expose game designers to conventions (ie: be aware of turn order and how that affects the game)
- breaking turn order actually affects the turn order
- Bath/Consistency game right now is more interesting/fun to play the bad version

Appendix

References

- Heuristics
 - [10 Usability Heuristics Applied to Video Games](#)
- Examples
 - [Good Design, Bad Design](#)
 - [Game UI Database](#)
 - [A Diagnostic Taxonomy of Failure in Videogames](#)
- Games
 - [Evoland](#) (UI Progression)

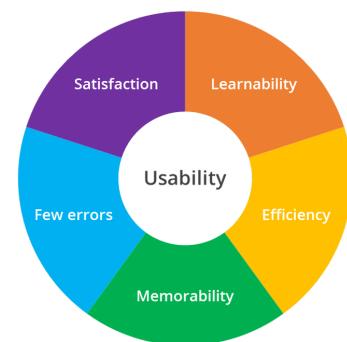


- [Potionomics](#) (Travel between conversations with characters without a map)



Nielsen's Heuristics

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



Nielsen's Heuristics

- 1. Visibility of system status**
 - The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.
- 2. Match between system and the real world**
 - The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.
- 3. User control and freedom**
 - Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.
- 4. Consistency and standards**
 - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.
- 5. Error prevention**
 - Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.
- 6. Recognition rather than recall**
 - Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.
- 7. Flexibility and efficiency of use**
 - Shortcuts – hidden from novice users – may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- 8. Aesthetic and minimalist design**
 - Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.
- 9. Help users recognize, diagnose, and recover from errors**
 - Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.
- 10. Help and documentation**

- It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

Game Heuristics

O12 Menu and Interface Elements (HUD)

- 12.1 The interface is consistent in control, colour, typography and dialog design (e.g. large blocks of text are avoided, no abbreviations) and as non-intrusive as possible.
- 12.2 The menu is intuitive and the meanings are obvious and perceived as a part of the game.
- 12.3 The visual representation (i.e. the view) allows the user to have a clear, unobstructed view of the area and of all visual information that is tied to the location.
- 12.4 Relevant information is displayed and the critical information stands out. Irrelevant information is left out. The user is provided enough information to recognize her status and to make proper decisions.
- 12.5 If standard interface elements are used (buttons, scroll bars, pop-up menus), they are adhering to common game interface design guidelines.

Production Log (↖↖w↖↖)

Week of 4/24 (Final Showcase)

- Environment Art **Jeremia Lo**
 - Bathhouse Interior
 - Boss Cat Shrine
 - Fix Tilemap
- Character Art **Teresa Yang**
 - Bathhouse Cat Manager
 - Character Attack Animations
- UI
 - Button
 - Container for a bar (health bar, mana bar, etc.)
 - 9 Sliceable Dialogue Box
 - 3 MC and enemy portraits for turn order display
- Sound
 - Overworld Music
 - Battle Music
 - Attack SFX
 - Damage SFX
 - Walking SFX

- Dialogue SFX
- **Design Jeremia Lo**
 - Storyboard of the entire game
 - Game Script and Dialogue
 - Intro (enter town from bottom, learn about user control)
 - Bath Game
 - Bath battle
 - Shop Game
 - Shop Battle
 - Final Boss
- **Programming**
 - Finish up consistency bath game
 - Dialogue system
 - Full Battle System Prototype
- **Playtesting**
 - GCS Release
- **Final Showcase Pitch (~15 Minutes)**
 - High-level description
 - Learning objectives
 - Description and example gameplay Nellie Tonev Adrian Ma
 - Justify educational aspect Adrian Ma

Week of 4/17 (Check-In 2 Presentation)

- Presentation
 - Game Direction (Summary - make more specific) Nellie Tonev
 - Current Status
 - Learning Goals: IA and Consistency Adrian Ma
 - Wireframes Jeremia Lo
 - Bath / Overworld Programming Nellie Tonev
 - Battle Programming Adrian Ma
 - Finalized Heuristic Fix Examples Adrian Ma
 - Character Art Teresa Yang
 - Environment Art Jeremia Lo
 - Challenges Teresa Yang
 - Next Steps Jeremia Lo
 - Feedback Document Nellie Tonev
 - Guiding Questions (Check-in 2)
- Next Steps
 - Art
 - Characters

- User Control (main character)
 - Walk Animation
 - Attack Animation
 - Idle Animation
- Consistency
 - Idle Animation
 - Attack Animation
- Information Architecture
 - Idle Animation
 - Attack Animation
- NPCs
 - 3 versions
 - ~~Bath cats (steamed cat?)~~
 - Dirty cat with a bath towel (they all look the same — you can't see their color)
 - Clean bath cat in 4 colors
 - They are in the bathtub
- Enemy Cats
 - Big angry cat
 - Other smaller ones (At least 3 types)
- Overworld Tileset
 - Make some from the battle background
 - Some buildings and stone walkways (different theme than the outside)
- Design
 - Enemies
 - User Control
 - Normal moves
 - UI Altering Moves
 - Consistency
 - Normal moves
 - UI Altering Moves
 - IA
 - Normal moves
 - UI Altering Moves
- Programming
 - Shopkeeper Minigame
 - ~~Bath Minigame~~
 - ~~All Battle Content~~
 - ~~Link Everything Together~~
 - Menus

Implement Art

Week of 4/3 (Check-In 2)

- Design

- Content

- **Finalized Heuristic Fix Examples (element → standard being violated → proposed solution)**

Action Menu (IA) Adrian Ma

- Enemy UI Move: Action Menu IA
- Element Impaired
 - Player Health Display
 - **Action Menu**
 - Turn Order Display
- Standard Violated
 - Size elements by the frequency of their usage
 - Order elements by the frequency of their usage
 - Use intuitive labels for UI elements
- Proposed Solutions
 - A
 - B
 - C

Another IA One Nellie Toney

- Actions get moved into unintuitively-labeled folders (the labels can be cat themed)
- Element Impaired
 - Action Menu
- Standard Violated
 - A
- Proposed solutions
 - A
 - B
 - C

Consistency One Adrian Ma

- Color Coding Inconsistency: alter color coding which indicates whether skills cost health or mana

Another Consistency / Standards One Adrian Ma

- Enemy UI Move: Ordered List Standards
- Element Impaired
 - **Turn Order**
 - Action Menu
 - Enemy Stats Display
- Standard Violated
 - **Organize ordered lists from smallest to largest**
 - Use check-boxes for lists with multiple selections

- Use conventional design patterns that users can recognize
- Proposed Solutions
 - A
 - B
 - C

User Control One Adrian Ma

- Enemy UI Move: Error Recovery
- Element Impaired
 - **Action Menu**
 - Player Mana Display
 - Enemy Stats Display
- Standard Violated
 - Prevent User Errors whenever possible
 - Allow users to quickly backtrack from mistakes
 - Enable users to perform actions that they want to
- Proposed Solutions
 - A
 - B
 - C

Another User Control One Adrian Ma

- Too much Control: Ask user to confirm or deny every action
- Element Impaired
 - Turn Order
 - Player Stats
 - **Action Menu**
- Standard Violated
-

- Narrative
 - First area that you fix is the bath area where you would go and heal after a battle
 - After that, we can fix a shop
 - (stretch goal) final boss after you can buy items & heal
- Characters
 - Motivations
 - User Needs
- Gameplay
 - **Bath (Heal)**: Sorting cats into baths by color / breed / idk
 - **Shop (In-battle items)**: Overworld shop peaceful interaction → shop under attack! Chase after the enemy! → battle inner loop → return to shop with a new heuristicat/new heuristic learning → fix the shop!
- Art
 - Main party members

- User Control (main character)** **Teresa Yang**
 - Walk Animation
 - Attack Animation
- Consistency **Teresa Yang**
- Information Architecture **Teresa Yang**
- NPCs **Teresa Yang**
 - 3 versions
 - **Bath cats (steamed cat?)**
 - Dirty cat with a bath towel (they all look the same)
 - Clean bath cat in 4 colors
- Enemy Cats **Teresa Yang**
- Backgrounds
 - Finalized Battle Background **Jeremia Lo**
 - Wireframe of Shopkeeper Game (Information Architecture)** **Jeremia Lo**
 - Wireframe of Bath Game (Consistency)** **Nellie Tonev**
 - Background — four baths
 - Timer (1 star → 2 star → 3 star)
 - Cats walk in from left and exit at right if you don't sort them
 - Cats have a little UI box over them that shows their color
 - Text, hex code, xcircle, stroop task lol
 - Ways to communicate color
 - YELLOW
 - Color swatch of yellow
 - #FFFF00
 - rgb(255,255,0)
 - hsv(60°, 100%, 100%)
 - Audio
 - Cool or Warm color
 - Positional
 - Colors for cats/cat bracelets/bath label
 - Yellow
- Programming
 - Dialogue **Nellie Tonev**
 - Full Battle Prototype** **Adrian Ma**
 - Shopkeeper Minigame **Adrian Ma**
 - Bath Minigame** **Nellie Tonev**
- User Testing
 - Feedback on Heuristic Fixes Wireframes

Week of 3/21 (Check-In 1)

- Game Design Document

- Learning Overview
 - Learning Goals Jeremia Lo
 - Learning Mechanisms Jeremia Lo
 - Learning Domain
 - Targeted Aspects of Player Experience Design Teresa Yang
 - Knowledge Elicitation @Everyone

- Game Design Overview

- Game Summary Nellie Tonev
- Narrative Nellie Tonev
- Core Gameplay Loop Storyboard Nellie Tonev
- Player Goals Adrian Ma
- Heuristic Cats Adrian Ma

Enemy Cats

2-4 Ideas for UI breaking moves (try to limit to mistakes that would actually be made in practice)

- Adrian Ma
- Teresa Yang
- Nellie Tonev
- Jeremia Lo

- Battle System Storyboard Adrian Ma

Concept Art ([PINTEREST](#))

- Characters Teresa Yang
- Environment Jeremia Lo

- Implementation

- Overworld Movement Nellie Tonev
- Walking into a building (scene swap)
- Battle System Framework Adrian Ma

- Knowledge Elicitation

- Survey + survey recruitment Nellie Tonev
- Adrian Ma
- Teresa Yang
- Jeremia Lo

- Check-In Presentation

- Game Direction
 - Summary Nellie Tonev
 - Overworld Nellie Tonev
 - Battles Adrian Ma
- Current Status
 - Refined Domain

~~Learning Goals~~ Jeremia Lo

~~New Set of Heuristics~~ Adrian Ma

Knowledge Elicitation Jeremia Lo Teresa Yang

- Research Questions
- Methods
- Insights

Programming

- Overworld Nellie Tonev
- Battles Adrian Ma

Concept Art

- Characters Teresa Yang
- Environment Jeremia Lo

Challenges @everyone?

Next Step

