



THE LEGEND OF
ZELDATM
TEARS OF THE KINGDOM

GAMEPLAY SYSTEM DESIGN ~ NOVEL SYSTEM
SETTLEMENT BUILDING AND MANAGEMENT SYSTEM
JACK PULLEN

CONTENTS

HIGH LEVEL SUMMARY	3
System Essence Statement	3
Chosen Game Analysis	3
MDA Framework analysis (Mechanics – Dynamics – Aesthetics)	6
System Analysis of Chosen Game	12
Problem Statement	14
How the System Addresses the Problem	14
Core System Concepts	15
Expected Player Experience	15
GAME RESEARCH	16
Design Considerations with Chosen game	16
Similar Games Analysis	16
Similar Systems Analysis	17
Academic Research	19
The “4 Keys 2 fun” as presented by Nicole Lazzaro	19
“Annotation Patterns” as presented by Björk and Holopainen in “Patterns in Game Design”	21
Flow: The psychology of Optimal Experience, by Mihaly Csikszentmihalyi	26
The “Octalysis Framework for Gamification & Behavioural Design” as presented by Yukai Chou	27
ITERATIVE DOCUMENTATION	33
Design Outputs Iteration 1	33
Villager Roles (Alphabetical A-Z)	33
Villager Management	39
Outline of Prototype Contents	39
Iteration 1 Evaluation	41
Rationale for Changes to Iteration 1	42
Design Outputs Iteration 2	42
REFERENCES	S2

HIGH LEVEL SUMMARY

System Essence Statement

An epic adventure across the land and skies of Hyrule awaits in The Legend of Zelda: Tears of the Kingdom. With the people of Hyrule rebuilding after the Great Calamity, the brave protagonist Link tries his hand too. Gather resources and become mayor of a village you design and build, and meet plenty of brand-new people.

Chosen Game Analysis

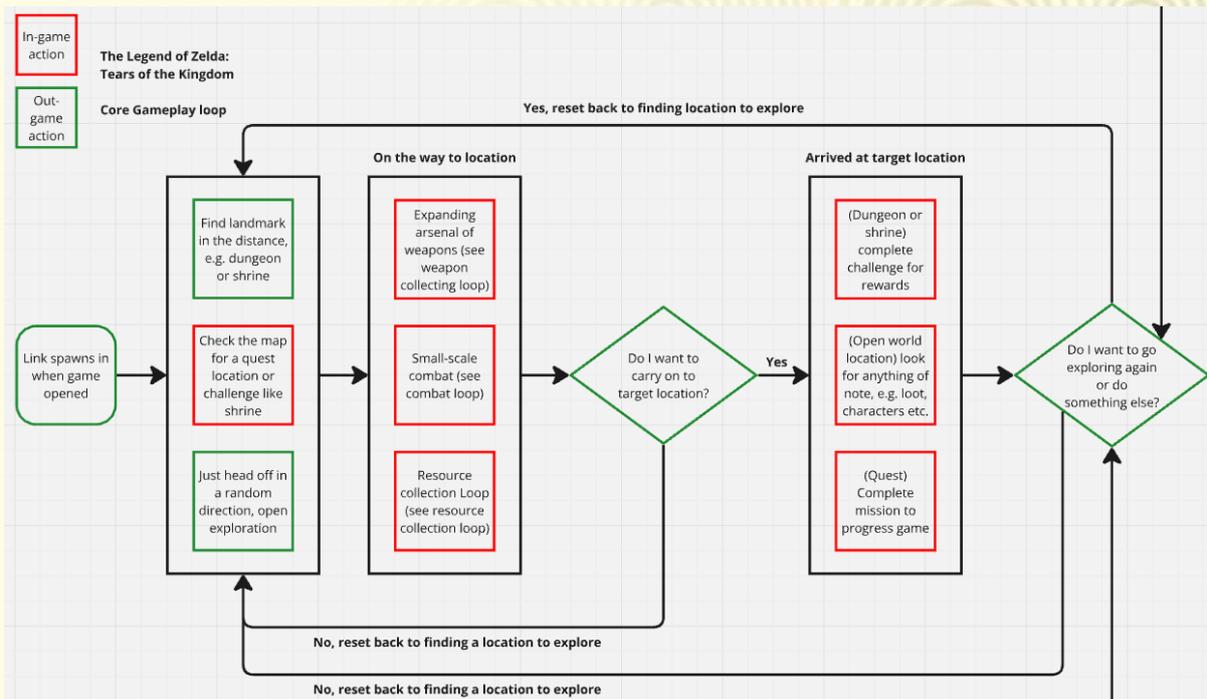
Core gameplay

“The Legend of Zelda: Tears of the Kingdom” (2023) is an action/exploration game developed and published by Nintendo, available exclusively for Nintendo Switch. Sequel to 2017’s “The Legend of Zelda: Breath of the Wild”, Tears of the Kingdom’s (ToTK) sees the main protagonist “Link” defeated, and awakens on a newly arisen archipelago of sky islands, wielding new powers, designed to enhance the open-world and puzzle-solving experiences carried on from Breath of the Wild (BoTW).

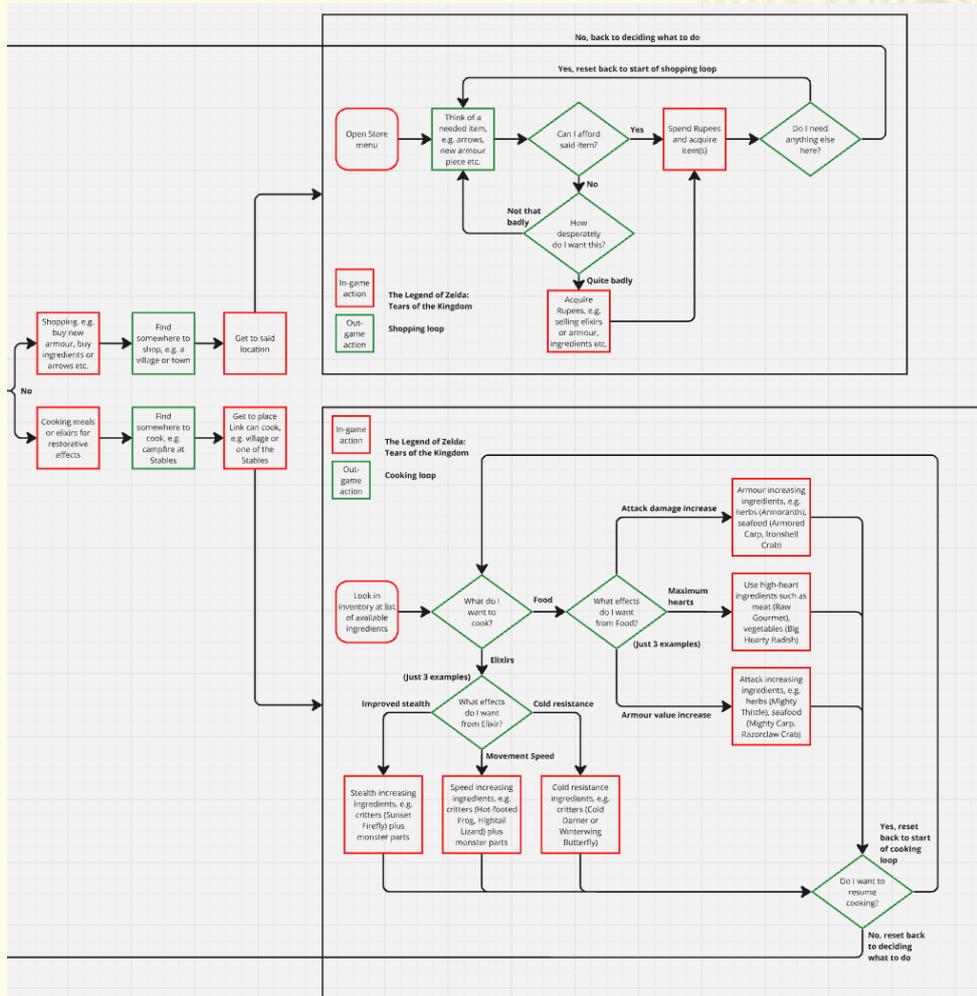
From here, Link goes sets off on a journey, completing a variety of quests taking him to the 4 corners of the known world, in a variety of landscapes and biomes, each with unique environmental hazards, be it the freezing cold of the Rito lands, or the incinerating heat of Goron territory. Across said journey, Link can collect a variety of ingredients, such as meat, fruit and vegetables, or monster parts like horns, wings etc., from which he can cook a variety of meals and “Elixirs”, both granting restorative and stat-buffing effects, such as increased attack power or defence, temporary increased health and/or stamina etc.



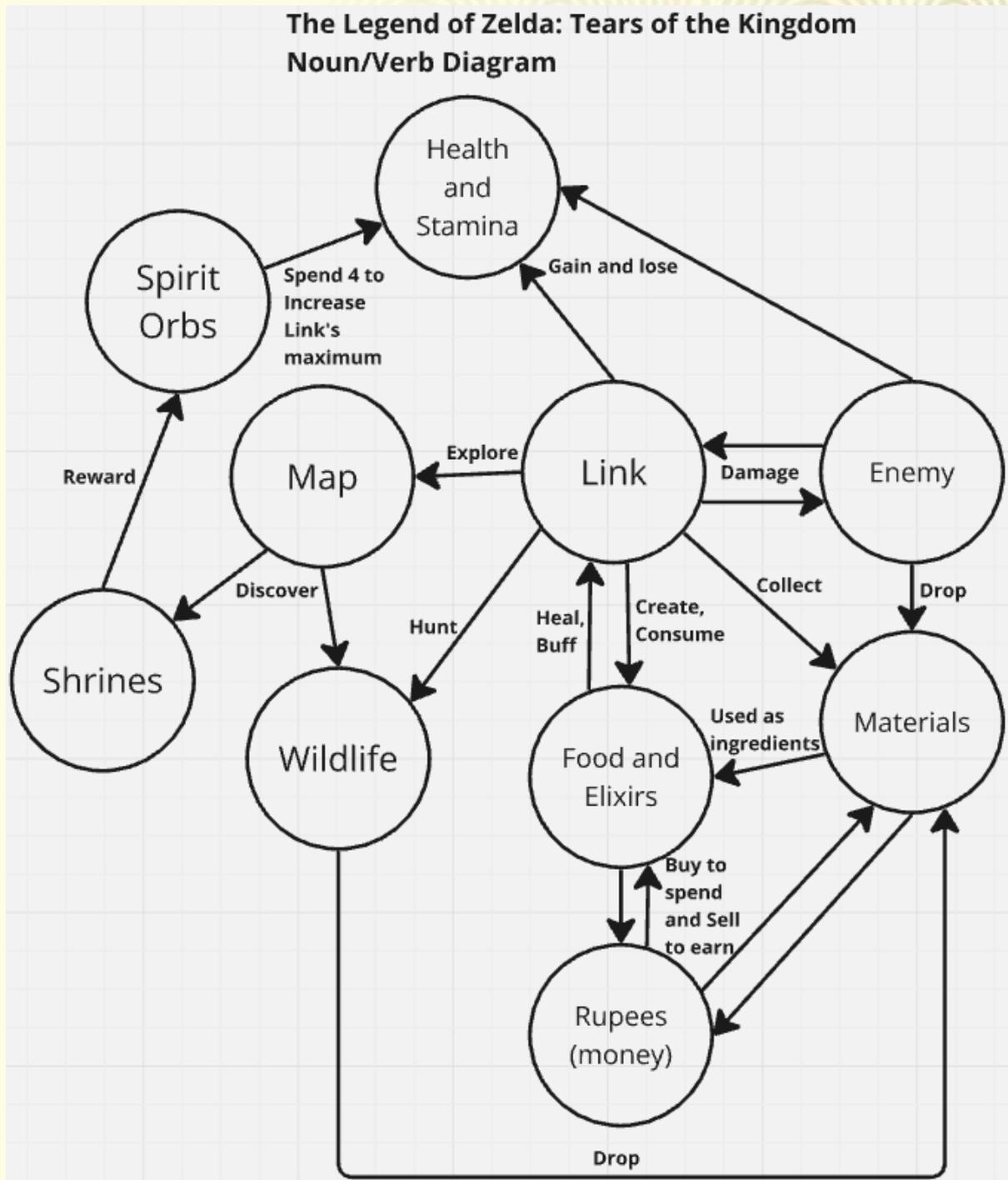
- Core gameplay loop



- Core gameplay loop part 2

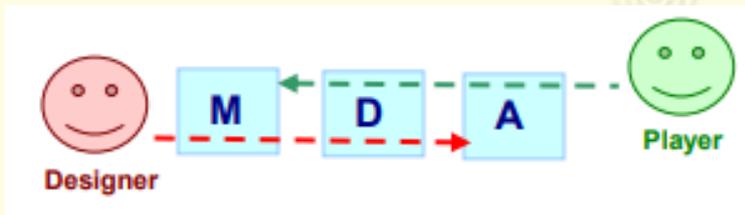


- Noun/Verb Diagram

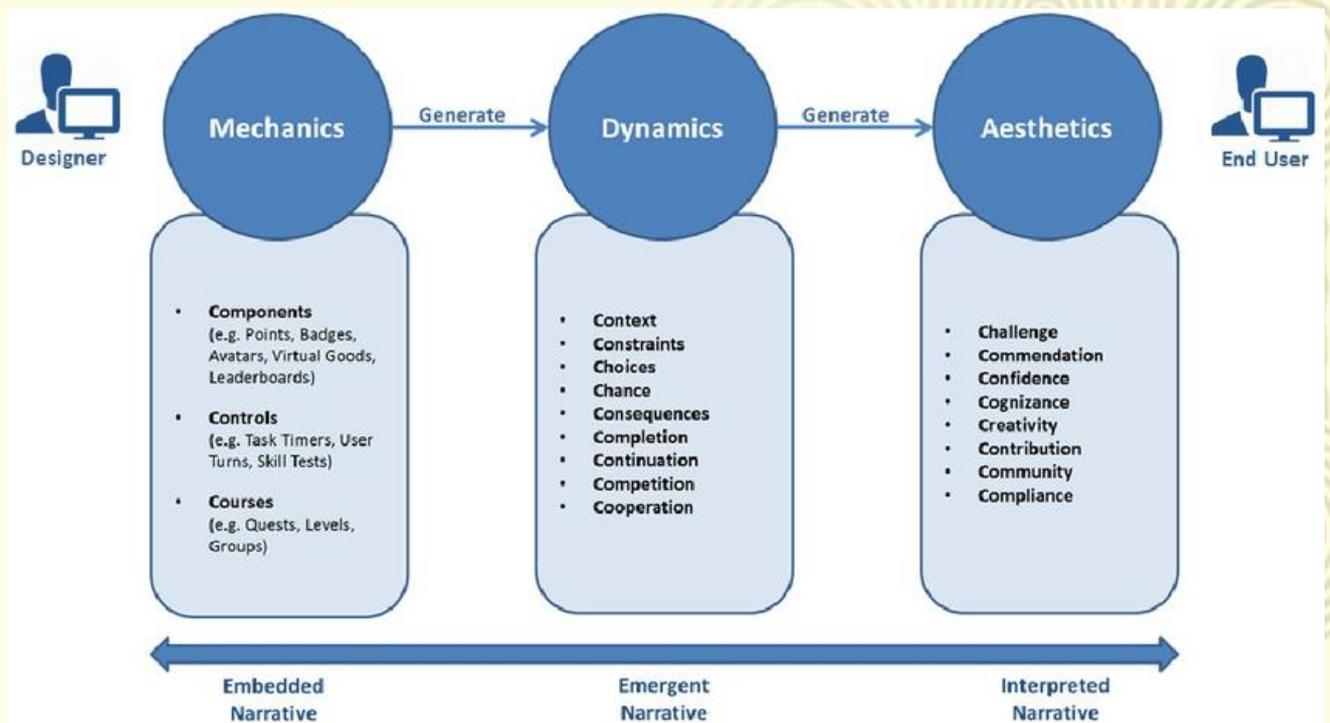


MDA Framework analysis (Mechanics – Dynamics – Aesthetics)

The “MDA Framework” (Hunicke et al, 2004)’, “is a formal approach to understanding games – one which attempts to bridge the gap between game design and development, game criticism, and technical game research.” Here, games are broken down into their mechanics, Dynamics, or how these systems create different types of gameplay, and Aesthetics, or different player responses and receptions to said dynamic(s) (see figure 1). The following is an example list of Mechanics, Dynamics and Aesthetics, followed by a series of applied MDA examples. In coming up with the list of Dynamics specifically, the list was expanded (Ruhi, 2015) with aid of “The MDA Framework and the 20 C’s of Meaningful Enterprise Gamification” (see figure 2).



(Figure 1, the MDA framework diagrams from the original 2004 paper)



(Figure 2, the 20 C’s of Meaningful Enterprise Gamification)



Mechanics

- Combat system
- Health system
- Stamina system
- Movement system
- Equipment system
- Dungeons
- Shrines
- Skyview Towers
- Cooking system
- Environmental effects system - heat, cold, fire, rain, thunder
- Open World system
- Quests

Subsystems

- Equipment system - Weapon durability system
 - Items breaking dealing additional damage
- Open World system - Exploration
- Open World system - Puzzles, e.g. Koroks or chests
- Movement system - Horse riding
- Movement system - Gliding system
- Combat system - Hunting

Dynamics

- Consequences
- Choices
- Constraints
- Exploration
- Boss Battles
- Completing quests
- Hunting system



Aesthetics

- Mastery
- Satisfaction
- Immersion
- Confidence
- Completion
- Freedom
- Sensation (Game as sense pleasure)
- Fantasy (Game as make-believe)
- Narrative (Game as drama)
- Challenge (Game as obstacle course)
- Fellowship (game as social framework)
- Discovery (Game as uncharted territory)
- Expression (Game as self-discovery)
- Submission (Game as pastime)

MDA Frameworks:

- Mechanic

Environmental effects system

- Dynamic

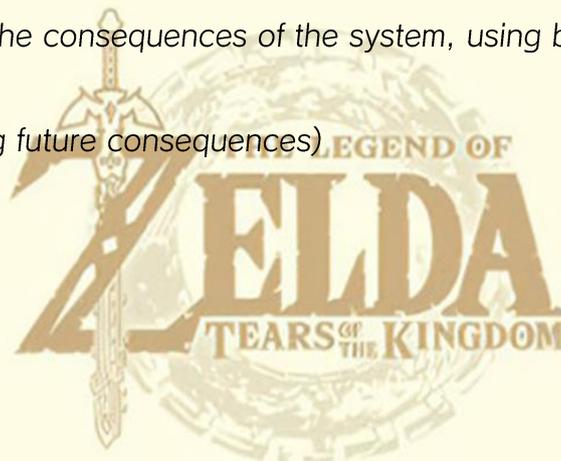
Consequences (of not preparing for environmental challenges)

- Aesthetics

Challenge (initially facing the system)

Mastery (overcoming the consequences of the system, using better equipment, food and/or elixirs)

Confidence (in tackling future consequences)



- Mechanic

Shrines system

- Dynamic

Constraints (specific items or objects needed to progress, e.g. a ball in a socket, creating a circuit, mini boss battle etc.)

- Aesthetics

Challenge (working out puzzles and working their way through the shrine)

Mastery (emerging successful from challenge)

Completion (working towards game completion (either beating the game or 100% completion))

Sensation (use of sounds and particle effects to wow the player)

- Mechanic

Equipment system (weapons and armour)

- Dynamic

Choices (freedom to choose and use any available equipment, e.g. armour, weapons, bow and shields)

- Aesthetics

Satisfaction (the player gets to look and play how they like)

Challenge (challenges to unlock certain armour sets, managing revolving list of weapons as they break and Link grabs new ones)

Expression (the player can mix and match armour, use their favourite kinds of weapons, armour can be dyed)



- Mechanic

Skyview Towers, Dungeon System and Open World System

- Dynamic

Exploration (the player is encouraged to go into the world and explore everything they can)

- Aesthetics

Discovery (uncharted territory, exploring a new world. ToTK even has some new elements for returning BoTW players)

Narrative (the player learns more about the game world as they explore and meet new people and enemies)

Fantasy (new races and enemies, new world)

Challenge (variety of puzzles to reach said locations, e.g. sub quests, obstacle courses etc.)

Completion (gets closer to completion with each discovery, e.g. unlocking more of the map with each Skyview Tower)

Submission (this all takes a long time, but does not lose or bore the player)

Immersion (the player is immersed in the game world during exploration and discovery)

- Mechanic

Cooking system

- Dynamic

Constraints (only maximum of 5 ingredients, meals separate from elixirs)

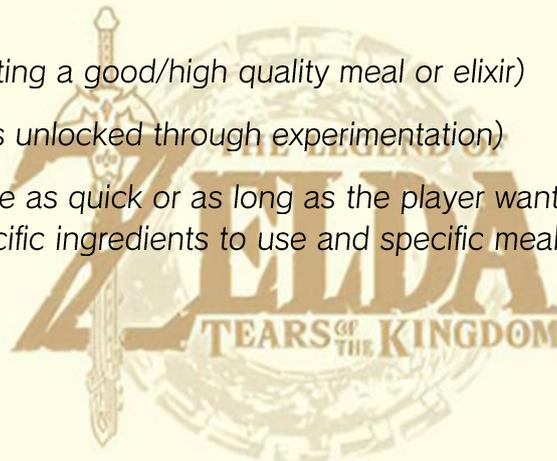
- Aesthetics

Challenge (going out into the world to get higher quality ingredients for better meals and elixirs)

Satisfaction (from creating a good/high quality meal or elixir)

Discovery (new recipes unlocked through experimentation)

Submission (this can be as quick or as long as the player wants, potentially putting a lot of thought into specific ingredients to use and specific meals and elixirs they want to make)



- Mechanic

Combat system

- Dynamic

Boss Battles

- Aesthetics

Challenge (fighting the boss, managing health, stamina, food and elixirs, and equipment durability)

Mastery (beating said boss, maybe after multiple attempts)

Satisfaction (relief in having beaten boss if the player was stuck, and able to claim rewards for beating said boss)

Narrative (story behind the boss, main story, side quest, open world exploration)

- Mechanic

Quests system

- Dynamic

Completing quests

- Aesthetics

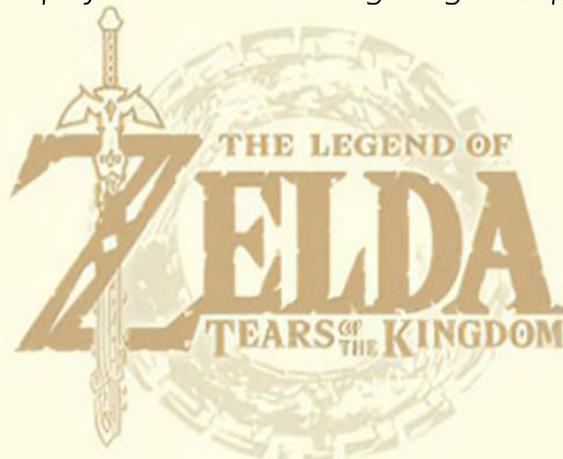
Immersion (engaging with the game world)

Narrative (the story of the mission, e.g. main story, side mission, shrine quest etc.)

Fantasy (the fantasy elements of the quest, e.g. any potential monsters, magic, locations etc.)

Challenge (actually completing the mission, e.g. a fight, puzzle of using certain items or using different movement systems, e.g. gliding, climbing etc.)

Freedom (the player is free to do or not do any quest they want and are not punished for not doing quests, the player consents to not getting said quest's rewards)



- Mechanic

Environmental effects system - Cold weather

- Dynamic

Hunting System – Animal and drops are affected by temperature

- Aesthetics

Discovery (dropped meat in cold weather can freeze to make "Icy Meat", giving Link a temporary heat resistance buff)

Discovery (killing an animal with fire weapon or fire arrow instantly cooks dropped meat for more heart recovery, no need to then go and cook said meat)

Mastery (players can master hunting, and use this system to progress the game)

System Analysis of Chosen Game

Analysis of systems integral to Proposed Novel System

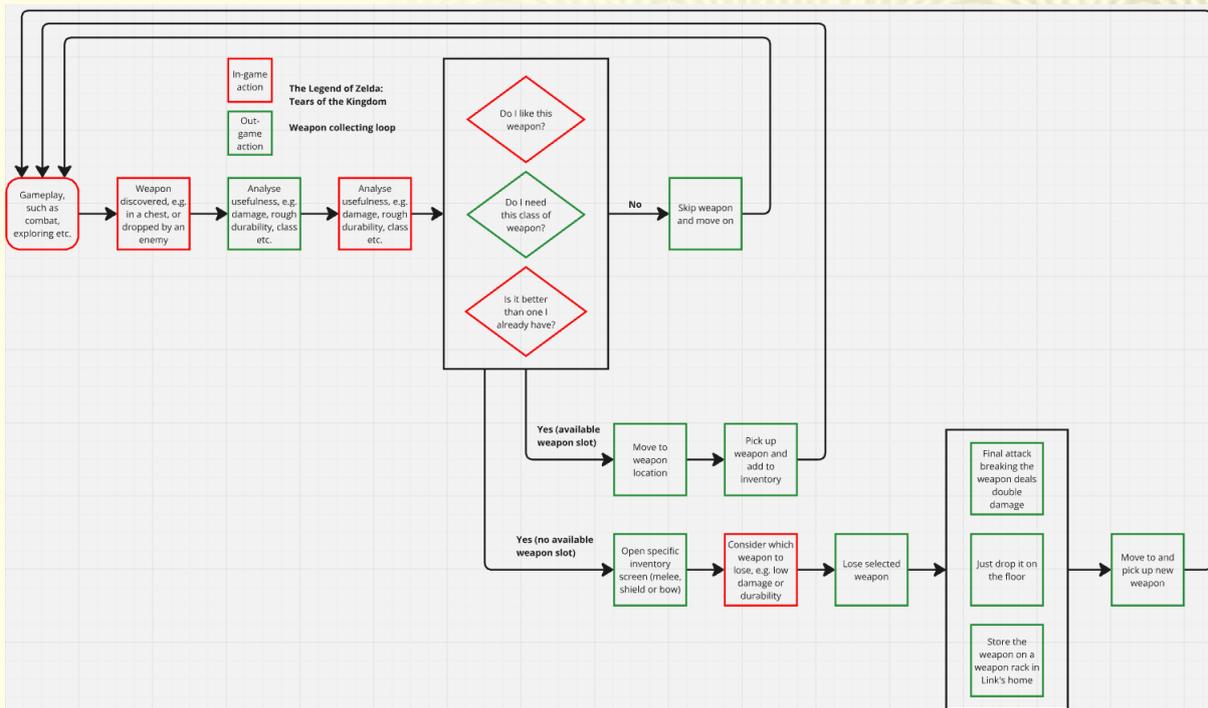
- BoTW and ToTK durability, equipment and weapon-based resource collecting systems

Newly added to BoTW, highly divisive on release but carried over to ToTK, was the addition of durability to weapons. Instead of starting with an infinite durability master sword, Link in BoTW awakens with nothing, and must arm himself with weak, flimsy weapons, slowly building up an inventory of different kinds of weapons, e.g. clubs, swords, spears etc., of slowly increasing damage and durability.

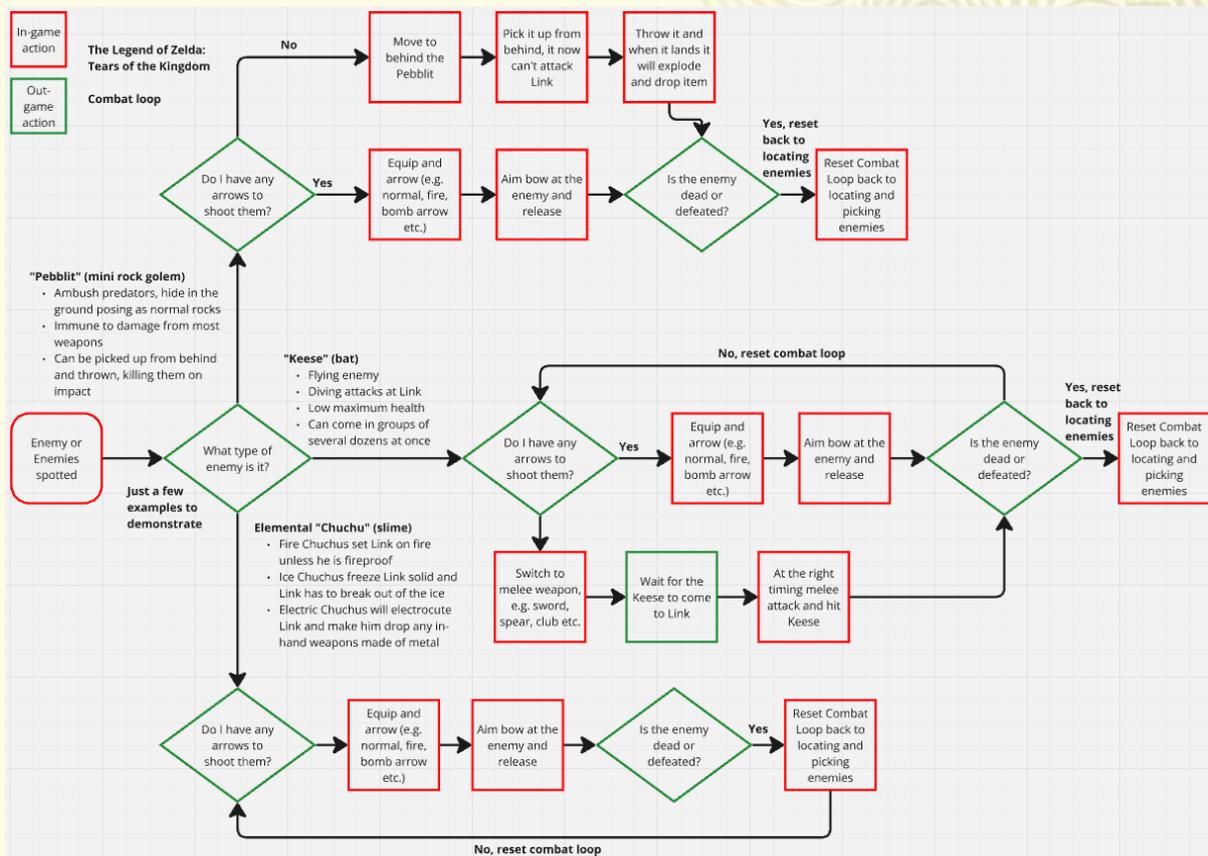
Different weapons will have different effects in relation to one of the core mechanics of the proposed Novel System, resource collecting. Here, blunt weapons like sledgehammers and great clubs are ideal for breaking open ore deposits, which drop items like Salt for cooking, Flint for lighting fires, or gemstones for powerful elemental effects for Link's weapons. These same weapons will conversely do nothing to chop down a tree, to gather wood used for fires, or as a building material for several quests throughout BoTW. Likewise, sharp bladed weapons like swords or axes are ideal for chopping down trees, but will do very little to crack open ores. Players must then be sure to collect both types of weapons for said eventuality.



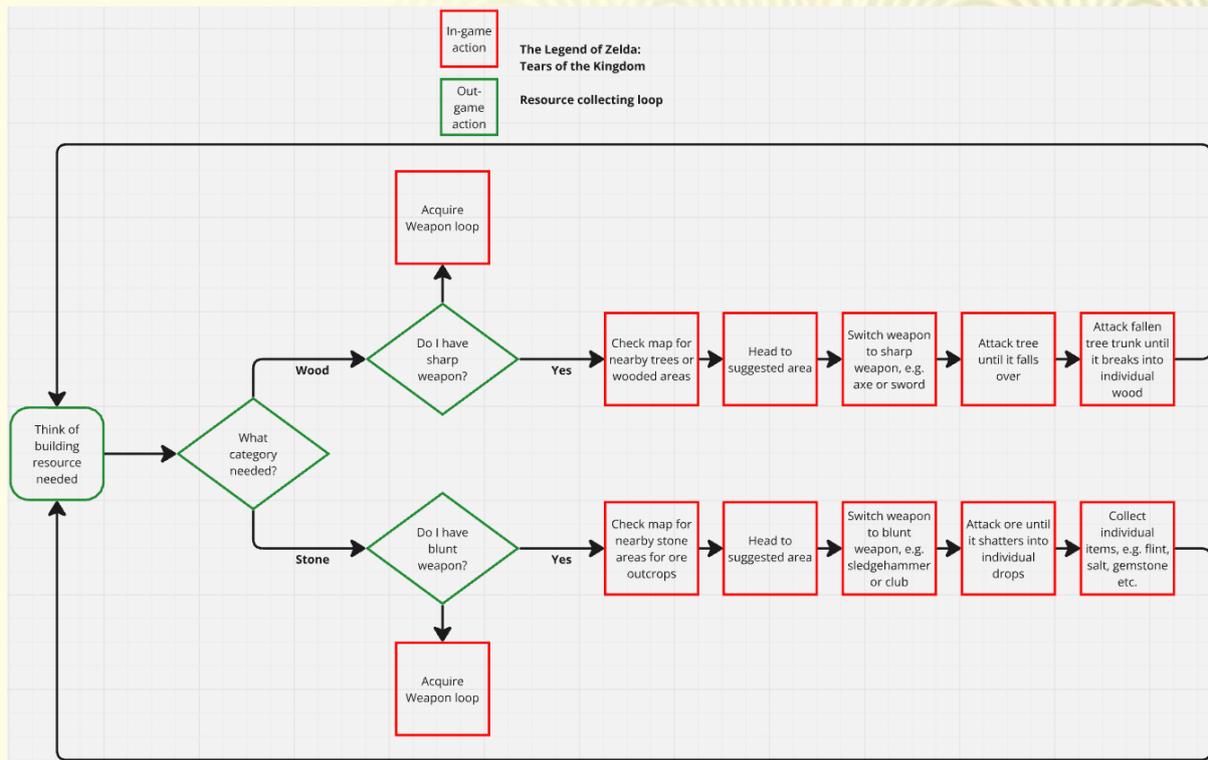
- Weapon collection loop



- Combat loop



- Resource collection loop

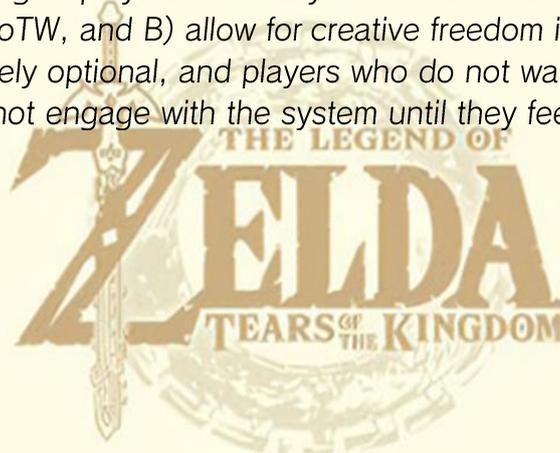


Problem Statement

The Tarrey Town quest line in BoTW was a fun and engaging experience, encouraging players to go all around the world, and engage with several gameplay systems. ToTK has none of this, and thus a new system with more freedom than BoTW would elevate ToTK to not only on par with, but potentially exceed the BoTW experience.

How the System Addresses the Problem

The new system will A) give players the ability to recreate the experience of the Tarrey Town questline from BoTW, and B) allow for creative freedom in so doing. At the same time, this is entirely optional, and players who do not want to, or are holding off for a later date, need not engage with the system until they feel they are ready.



Core System Concepts

- Creative Expression through custom town layouts, decorations, the people there etc.
- Adds a new layer of gameplay, asides from just running around collecting, and completing shrines and dungeons
- Gives back BoTW players what they miss, and give new ToTK players who never played BoTW a similar experience they would otherwise miss out on

Expected Player Experience

The player will get to spend time collecting resources to invest into the system, exploring the world and discovering new locations, items, enemies etc. as they do so, then invest them into the village building system, which will eventually pay itself off, giving Link additional resources like food, elixirs and equipment. In so doing, they will get to individualise their villages, meet new characters, and explore a new side to the game that as of yet does not exist in the Legend of Zelda franchise.



GAME RESEARCH

Design Considerations with Chosen game

- Townsfolk/villagers cannot die

ToTK, like BoTW before it, makes only sparing use of death in its narrative, and thus it would be out of place and stand out to the player for random people to start dying left, right and centre. Typically, if a character is attacked in these games, they will either fight back, or cower until help arrives or they are knocked out, getting up again soon after.

Villagers in the game will thus not die, and will either be knocked out as mentioned, or leave the village. This could be for several reasons, e.g. it would be easier on the player if they ran out of food than watching everyone starve to death.

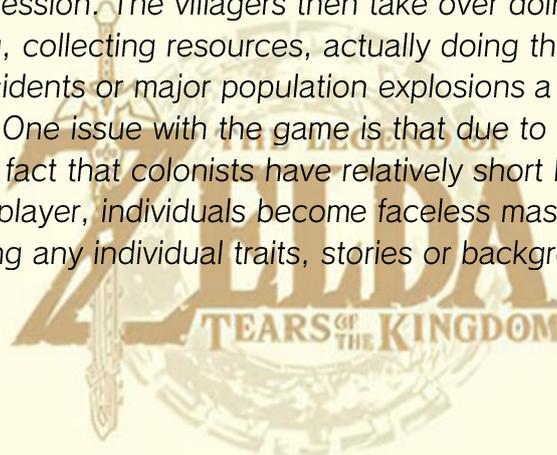
If villagers can then be knocked out or defeated by enemies, this adds importance to getting guards/soldiers, so that they can escort or protect villagers both inside and outside of the village.

- Other villager stuff

Since they cannot die, it would also make sense that they do not age, since the only time people have aged in either BoTW or ToTK was between games, and the time-skip at the start of BoTW. Villagers will thus be created at a certain age and stay that way.

Similar Games Analysis

"Banished" is a 2014 colony building and management simulator, in which the player builds a growing colony of villagers. Here, the player takes a god-like role in designating what buildings to be built and where, as well as assigning a total number of workers in each profession. The villagers then take over doing everything else, including re-populating, collecting resources, actually doing the building etc. This means that bar any incidents or major population explosions a colony should quickly become self-sufficient. One issue with the game is that due to the god-like nature of the gameplay, and the fact that colonists have relatively short lives requiring little management from the player, individuals become faceless masses, differentiated by only a name and lacking any individual traits, stories or backgrounds.



Rimworld (2018), like Banished, is also a colony building and management simulator, this time a pixel-art based, smaller scale version. In contrast to Banished, with colonies potentially of several hundreds of members, Rimworld will typically peak at a few dozen at most, usually even less. Also in contrast, it has a much bigger emphasis on connecting with and experiencing the stories of each colonist, referred to as “pawns”. Like Banished, the player only assigns what to do and the pawns do the work, again playing in a god-like role, the player this time able to have some influence over the pawns themselves, able to direct them to different tasks, without assume total control of said pawn(s).

- Takeaways for Novel System

One of the main takeaways would be to blend both god-like and personal management experiences from both Banished and Rimworld. The idea of being a master architect plotting out everything is ideal to meet the core concept of player creative expression, while the idea of forming inter-personal relations with each colonist is generally a nice thought, just as the player might form relationships with the rest of the cast of ToTK.

Another takeaway from both would be the idea of villager-driven gameplay, not Link doing all the work. As with both games, the idea is to assign buildings and items to be made, and leave it to the villagers to do over time, thus giving Link a reason to keep coming back, as there will always be something waiting for him.

Similar Systems Analysis

- Banished building system

To again reference Banished, the game is played in a top-down camera angle. The player can place buildings in a grid-based system, with each building having a set grid shape. These buildings can be placed, to be built by builders, regardless of available materials, as builders will wait for resources to become available and then start building. This is better than the alternative of having to wait for all the resources to be available before a building can be built, since in the proposed ToTK system Link would have to go out and collect everything for the builders hut, lumberjack and miners huts at a minimum, before a village could become self-sufficient.

- Banished professions system

Each colonist will take up one of many different “Professions” in the game, based on profession availability. For instance, building a “Gatherer’s Hut” will add 4 available “Gatherer” job vacancies, while a “Fishing Dock” will add 4 fisherman vacancies. This was the original plan, on a smaller scale, to be adapted for the ToTK system, wherein each building would grant x number of job vacancies.

- “Freeman: Guerilla Warfare” squad equipment management UI

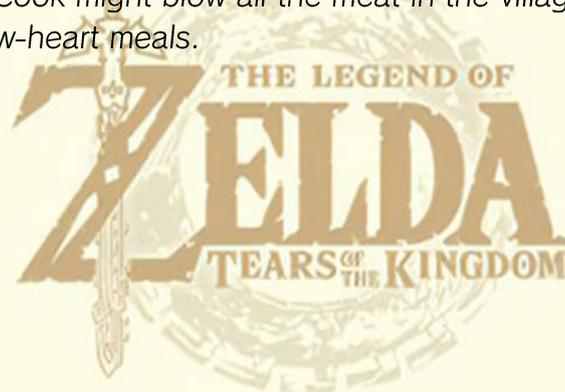
Freeman: Guerilla Warfare is a 2018 strategy game, in which the player forms a squad, and eventually a whole faction of troops fighting to liberate his country from different warring factions and bandits. The specific system to analyse in this is the soldier equipment interface. Here, the player can freely change the equipment of every soldier in his personal army, including their clothes and armour, weapons and special equipment like medkits, grenades etc. Something like this would be ideal for specifically Guard/Soldier management. This could be to assign different weapons to each soldier, e.g. 2-handed versus sword and shield, equipping some or all with bows and different types of arrows etc.

- Rimworld item production and priority systems

In Rimworld, like Banished, players do not create items, but pawns are assigned to set tasks and make things themselves. To do this, jobs like cooking, crafting, tailoring etc. use a priority system in their points of interest, e.g. the oven for the cook, tailoring bench for the tailor etc. Here, the player sets out a list of different items they want made, and the pawn will work down the list in priority order top-bottom, assuming they have the resources to do so. Additionally, if resources for items near the top are missing, but not for items lower down the list, the pawn can skip items until resources become available and work on lower-priority items instead. This can be adapted to the Novel System, in which for the Blacksmith for example, essential tools like Hoes and Sledgehammers can be prioritised for the running of the village, with items like armour or weapons for guards can be done later. Alchemists could prioritise certain elixirs over others, and Fletchers could prioritise certain kinds of arrows.

- Rimworld item total systems

In relation to item production systems, total or target amounts can be changed for each item in each list. For instance, something could be produced over and over again forever, or until X of said item is in storage, ensuring just enough exist at once for the colony to not run out, without filling all the storage areas with something the colony only needs a few of. Gameplay examples of this might include forever making things like meals, while things like spare clothes or weapons the colony will only need a few of. As with the priority systems, Link could limit how many of each item to make, otherwise a Blacksmith might spend all the ore in the village on just making Farming Hoes, or the Cook might blow all the meat in the village on repeatedly making single-item, low-heart meals.



Hard Fun:

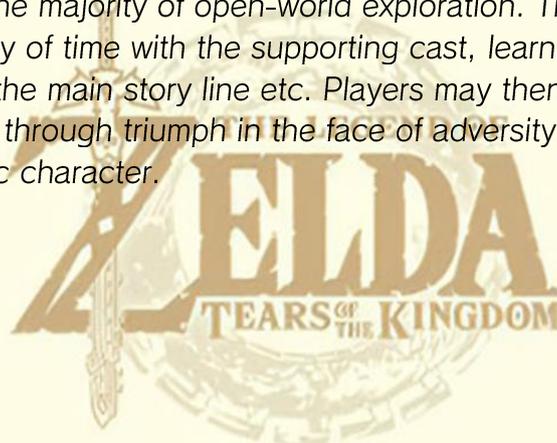
ToTK has plenty of examples of “In the moment personal triumph over adversity”. Every shrine completed for one gives Link a “Spirit Orb”, which both fulfils the player at the moment of beating said shrine, and again redeeming them for permanent stat improvements, namely additional maximum health and stamina. Boss fights, both main mission and random open-world encounters likewise provide these triumphs, as would even just regular small battles against random groups of enemies. In reference to focusing attention “with a goal, constraints and strategy”, this is all catered to in abundance. The goals can be either game-dictated, e.g. the main storyline, collecting all the memories, completing side quests etc., or player-made goals, e.g. “I want to get over there” or “I want to max upgrade X armour”. Constrains come in mainly the early game, with a low health, low stamina character combined with poor quality equipment, that slowly improves over time to eventually overcome said constraints. In strategy, this could include the weapons Link uses, the food and elixirs he makes, each with different effects, and the different gameplay styles one might employ, such as a highly defensive, careful stance versus an all-guns-blazing, hyper-aggressive gameplay style.

Easy Fun:

“Curiosity” is also met in abundance, especially if the player hadn’t played the prior entry “The Legend of Zelda: Breath of the Wild”, which shares a game world and host of characters, but still standing well on its own if they had. New characters, new locations, new storylines and missions await both new and returning players, as well as the usual exploration involved in finding Shrines, “Korok Seeds” and rare weapons and armours from different dungeons around the map. Easy Fun also makes mention of role play. Here, the interactions of Link and the characters around him would count, enhanced by the new village building system.

People Fun:

Being a single-player game, ToTK is not so much to catered here, since people fun focuses mainly on multiplayer. Additionally, much of the game is played with Link being alone, such as the majority of open-world exploration. That being said, the player can spend plenty of time with the supporting cast, learning more about them through side content, the main story line etc. Players may then feel connected to these characters, be it through triumph in the face of adversity, relatedness, or they just really like a specific character.

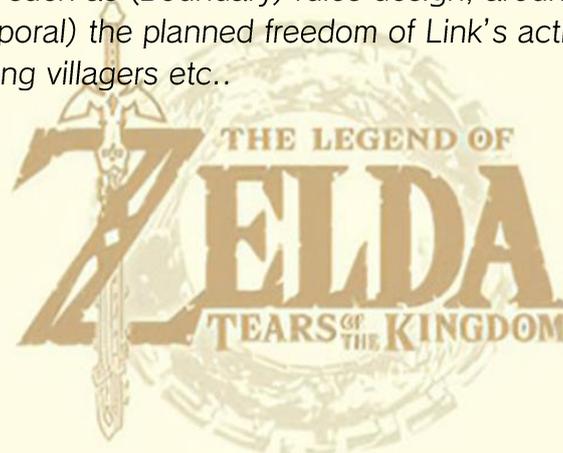


Serious Fun:

The game is not designed to be overly difficult, and the game difficulty scales as the main character improves, and thus not too often should the player not feel relaxed or excited to continue playing. The sensation of relaxedness is aided by the calming atmosphere of the game and its music, leisurely gameplay pace and the freedom to do anything in any order.

“Annotation Patterns” as presented by Björk and Holopainen in “Patterns in Game Design”

According to the Google Books synopsis (Björk et al, 2005)^{iv}, *“Annotation Patterns in Game Design provides professional and aspiring game designers with a collection of practical design choices that are possible in all types of games. These choices, called patterns, are used to illustrate the varying types of gameplay found in games. For the purposes of this book, gameplay is defined as the structures of player interaction with the game system and interaction with other players. This includes the possibilities, results, and reasons for players to play. By putting these elements of gameplay into practical patterns, designers have access to a common set of concepts that can be used by all developers, allowing game projects to be approached with more standard tools. These patterns help designers put their concepts and ideas into words, which makes communication between members much easier. The patterns also help with making design choices, understanding how other games work, and inspiring game ideas. The book itself is divided into two main parts. The first part covers the theoretical aspects of describing games and defining the template used to develop the game design patterns. The second part includes the actual patterns divided into chapters based on the aspect of gameplay they cover. The patterns can be used in any order and referenced as you would a dictionary. By studying these various game design patterns, designers learn about the choices they'll have to make when using a pattern in their own designs, and they'll gain an understanding of what gameplay is, so that they can design better games.”* (see figure 4). Applying these would give a better understanding of both the design of ToTK and the new village building and management systems, such as (Boundary) rules design, around building and management, or (Temporal) the planned freedom of Link's actions, such as freedom when building, managing villagers etc..



Boundary

- Rules

Rules would include systems like maximum weapon, bow and shield counts, which can be expanded by collecting “Korok seeds”, finding them either from completing for example small puzzles or environment traversal challenges. The durability system would count as a rule, as would the final blow of a weapon breaking dealing additional damage. Different combat rules exist, such as the existence of exploitable weak points for most enemies using bows, shields preventing Link from taking damage, and the different ways different classes of weapons behave, e.g. the thrusting of a spear versus the slashing of a sword.

New rules in the proposed Novel System would include buildings requiring X resources and a builder to build them, the different resource collection behaviours of each role, e.g. the Fisherman, Miner or Lumberjack.

- Modes of Play

As mentioned in the Hard Fun aspect of the 4 Keys 2 Fun, Modes of Play would come not from the game itself, but rather how the players play said game. The game itself only really has 1 mode, that being to play as Link, a set character in a set world, with set and outlines goals and objectives. Modes of Play would then be the different ways players play the game, e.g. a hyper defensive/aggressive combat stance, or player-led focuses on different aspects of the game, e.g. exploration or discovery.

- Goals

The end Goal of the game is to beat the final boss and restore peace to Hyrule. To earn said goal, the player can, but does not have to, complete a number of subgoals, such as the main missions around each of the “Sages”, long-dead characters from Hyrule’s past with elemental powers, that can power up Link to make the final boss progressively easier with each upgrade. This is similar to how the prior game Breath of the Wild did subgoals, in which the 4 “Divine Beasts” could be skipped, but in so doing the final boss would be harder and take longer. These subgoals also encourage further exploration of the game world, meeting the different races across the world and the items, weapons and enemies in their different areas.



Temporal

- Actions

The player has free control over their character, able to move freely within the constraints of the human body. The player is able to aim freely, asides from aiming 90 degrees vertical, since this would be very difficult.

- Events

The player can do the main story, side quests or random exploration, during which they might find random events, such as NPC's being attacked, saving them for a reward, or the player themselves being attacked by the "Yiga Clan" of assassins in disguise. In-world events also include the "Blood Moon", in which a cinematic plays of defeated monsters around the world respawning.

- Closures

Closures might include things like the weather, in which each weather condition is set to intervals of 30 minutes in-game, at which point they will either remain the same or change, conditions such as clear skies, rain, thunderstorms etc. Other closures may include loops like the cooking or weapon collecting loops, in which eventually the player will run out of inventory space and close said loops.

- End Conditions

End conditions can be both player and game-generated, for instance the player may call the game done when all shrines have been completed, all Korok Seeds collected, all armour sets collected etc. Game-wise, again these would include the main storyline.

- Evaluation Functions

Evaluation functions would work as outlined in the End Conditions. Technically the game is 'done' at the end of the main storyline, while players come up with and assess their own evaluation functions.



Structural

- Interface

Interfaces would include the main player HUD, including health and stamina, the current time of day and the minimap. Interfaces also include the different menus, such as both the full inventory menus for items like weapons, armour, food, materials etc., as well as the quick-swap menus for weapons, shields, bows and arrows, so the player needs not go into their main inventory, cycle through to the right tab and select every time they want to change equipment. As mentioned in the Game Instance for Holistics, the new Novel System would include several new interfaces, such as the building interface, villager role management menus, Guard equipment manager menu etc.

- Game Elements

As also mentioned in the Game Instance, elements would include Link, NPC characters, enemies, and any materials lying around the map either loose or in collection points, e.g. the difference between a mushroom growing on the ground, or an ore deposit the player has to first break open to collect the contents that then spawn around said deposit.

- Players

There is only 1 real player, being a single player game, the rest made up by AI NPC's. Any non-enemy AI cannot be damaged by link, e.g. villagers or townsfolk, although some characters can be damaged by enemies in open-world encounters, with Link able to save them for a reward. The game makes sure these characters cannot be killed though, in case the player was idle or misses the encounter and had no idea the event happened, so the characters are just knocked out for a moment and then wake up when the encounter ends.

- Game Facilitators

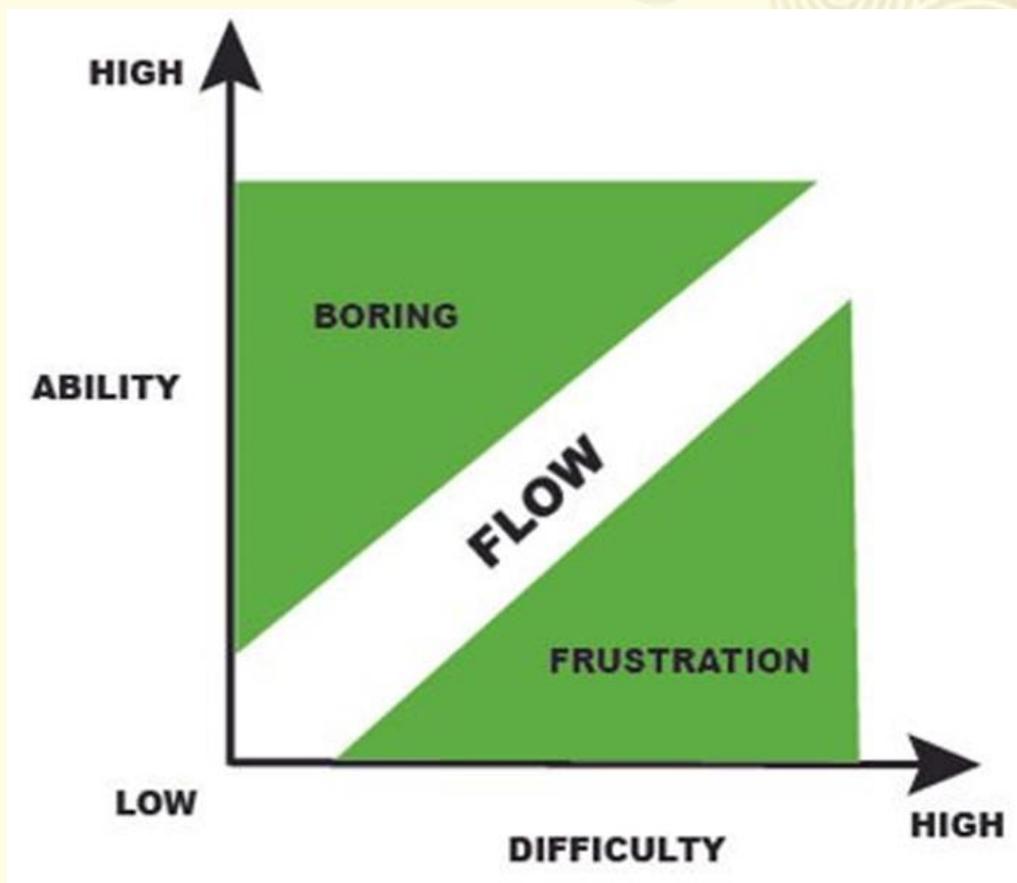
Game facilitators work similar to rules, but in how the gameplay is maintained and synchronized. An example of this could be the time system (see next). Other examples would include loading from a save whenever Link dies, while the game regularly makes sure to autosave in the event of a sudden or unpredictable death, potentially losing the player a lot of progress.

- Game Time

1 in-game day is 24 minutes, or 1 in-game hour per minute, thus an extended play session might last several days or even over a weeklong in-game. When the player leaves and rejoins the game, it remembers the time when the player left and loads in at said time, as not to throw off the player. Blood moons happen every 168 minutes of gameplay, or 1 in-game week of gameplay time. Skipping time does not count, needing to be active gameplay time.

Flow: The psychology of Optimal Experience, by Mihaly Csikszentmihalyi

According to the 1990 book “Flow: The psychology of Optimal Experience” (Csikszentmihalyi, 1990), by Mihaly Csikszentmihalyi, a sensation of “flow” can be achieved in balancing player ability to the difficulty of the game, while an imbalance of either sees players fall out of said flow into either boredom or frustration. While not originally invented for it, the “Flow Theory” has been adopted into games design for decades, and aides in difficulty design. Here then, the difficulty of the village building and management systems shall attempt to be balanced against these metrics.



(Figure 5, flow theory diagram demonstrating the relationship between player ability and game difficulty in relation to the sensation of “Flow”)

One drawback to mention of Flow Theory is that it only mentions ability and difficulty, and thus describing only these two factors on their own makes the gameplay sound boring, so care here will be taken to include the elements of fun, on top of difficulty versus player ability. The system is designed to work an optimal way, with players building certain categories of buildings in certain orders, such as first homes to house villagers, then resource production buildings for food and construction materials, then

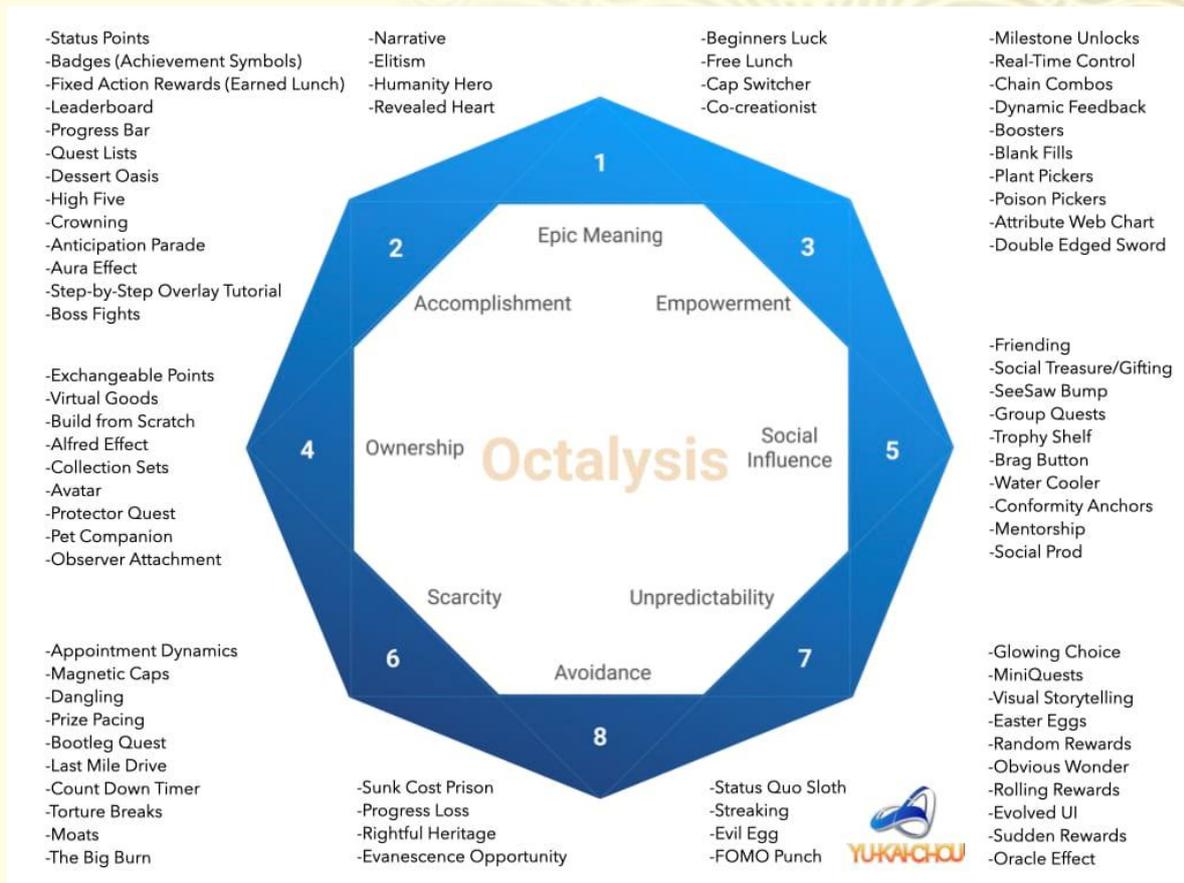
expanding to the rest of the possible buildings. This is not mandatory however, and players of a higher skill level can avoid boredom from a relatively simple and linear path by building different orders of buildings, thus mixing up their experience. Fun-wise, this mixing up of the experience can bring back an element of fun that might be lost from the system after multiple playthroughs of the game. To address the opposite, villagers will be able to be talked to, who will mention any needs they may have or any wishes they would like addressed, such as again resource production buildings like food. This will teach lower ability players how the system works to bring their skill in-line with the difficulty, and thus into a flow state. In reference to fun, this meeting and interacting with different characters can be entertaining, as can exploring their village to find them, and the learning process of finding out what to build.

The "Octalysis Framework for Gamification & Behavioural Design" as presented by Yukai Chou

"Gamification, a design approach centered around human motivation, takes elements from games and applies them to real-world activities. Octalysis emphasizes "Human-Focused Design" instead of mere functionality, optimizing human motivation and engagement within a system.

The framework comprises 8 Core Drives represented by an octagon shape, including Epic Meaning & Calling, Development & Accomplishment, Empowerment of Creativity & Feedback, Ownership & Possession, Social Influence & Relatedness, Scarcity & Impatience, Unpredictability & Curiosity, and Loss & Avoidance. By understanding and implementing these Core Drives, designers can create engaging experiences that cater to intrinsic motivators and promote positive user experiences."^{vi} (Chou, 2015) (see figure 5 for diagram). Such an in-depth breakdown of the core systems, as well as specifically new-system related mechanics, would improve the design depth of the new system, as well as improve design skills going forwards outside of this system.





(Figure 6, Octalysis diagram, this specific diagram found on Visuer Lab – The Octalysis Gamification Framework for corporate eLearning^{vii} (Kho, 2022))

Epic Meaning and Calling

- Narrative for Epic Meaning

BoTW and ToTK are both perfect examples of this, with the player experience epic journeys across the entire world, setting out to defeat the apocalyptic-scale evil “Ganondorf”. Set out on such a journey, players are urged and motivated to do so, creating a sense of purpose, instead of just being put in a world with no major story or plot, to which The Legend of Zelda series players would react with much less enthusiasm. This is aided by the extensive use of cutscenes and voice acting, to better get to know the dangers they are facing, and the characters aiding them along the way.

- Believability is a key factor

Believability in the Epic Meaning and Calling is helped by making the characters around Link believably human, with real individual drives, emotions, and beliefs for each character. Characters carried over from BoTW are much the same as before and have not made any sudden unbelievable or jarring characterisation changes.

Development and Accomplishment

- Importance of Challenges

Challenges are important to ToTK for multiple reasons. If the game was an open-world exploration game without any challenges, it would still be cool to experience, but there would be no motivation or reason to play beyond initial curiosity. Challenges such as environmental danger, e.g. severe heat or cold, shrine puzzles, or boss battles, all add to the motivation to play the game, and thus beating said challenges is satisfying for the player. Chaining successive challenges, such as putting multiple shrines near to each other, or multiple groups of enemies in close proximity would then amplify this effect.

- Need for Goals and growth

To not complete goals or to not grow is to stagnate, which in video games would be a major negative against player's sense of both development and accomplishment. Each system has thus been designed with slow but steady growth in mind, such as Link slowly but surely getting stronger with maximum health and stamina upgrades, his weapons and armour slowly becoming progressively better etc.

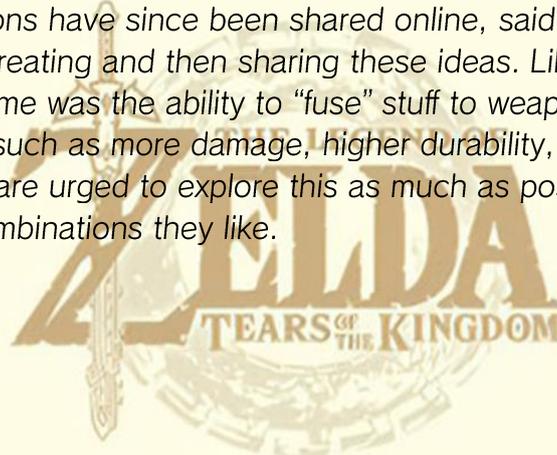
- Never Make Users Feel Dumb

The game features sufficient tutorialisation that users should never feel out of place, or that they are doing something wrong. This has been further reinforced by ToTK being a sequel, and thus players of the prior BoTW have experience in all systems but the newly added ones of ToTK. The game also makes use of non-BoTW, The Legend of Zelda franchise typical systems, such as the lock-on system in melee combat, gameplay tropes like treasure chests, bosses and the hearts system.

Empowerment of Creativity and Feedback

- Enjoyment of creativity

Players are urged to get creative in how they play the game. "Constructs" are machines that can be player-built from pre-made parts, such as batteries, fans, wings etc. A variety of creations have since been shared online, said players getting enjoyment both from creating and then sharing these ideas. Likewise, one of the main new features of the game was the ability to "fuse" stuff to weapons, in order to give them different effects, such as more damage, higher durability, elemental effects like fire or ice etc. Players are urged to explore this as much as possible, and to eventually find new combinations they like.



Scarcity and Impatience

- Motivation through Rarity

Certain weapons are much rarer than others in ToTK, with very few ever available to the player, as well as one-of-a-kind armour sets that can only be found in 1 location, such as a chest at the end of a dungeon with a rare piece of armour, or a rare weapon serving as a shrine reward on top of spirit orbs.

- Perceived versus Actual Value

Rarer items may not necessarily be better, according to the perceptions of each player. If for instance the rarest weapon in the game was a form of 2-handed weapon, but the player didn't like 2-handed weapons, its value would be perceived much differently to a player for whom 2-handed weapons are their favourites, and their go-to class of weapon.

- Evolving UI

The game handles Evolving UI in several interesting and unique ways. In the opening of the game, the player is presented the same full UI they learned playing Breath of the Wild, with all of the old abilities unlocked, and a full maxed-out health bar, as if the player had fully 100% completed BoTW. The opening of the game then plays out in full with Link losing all of his strength and powers, nearly dying in the process. He loses his "Sheikah Slate", an item which in BoTW was responsible for most of his abilities and HUD information, and thus Link slowly regains these HUD elements with a new "Purah Pad", tutorialising each new element for new players who may not have played BoTW.

Unpredictability and Curiosity

- Engagement through Unknown

While the shape of the world remains the same between games, the world of ToTK is shaken up to create a newfound sense of unknown that would have been much more difficult with an identical map. New areas have appeared, some have disappeared, and some have moved, including the physical areas themselves, associated people such as villagers or townsfolk, stables staff etc. Without this, the player could just have waltzed to the exact same places and discovered all the same things again with no new exploration or experiences required, thus making exploration disengaging.

- Finding the Right Risk-Reward Incentive

This could include systems like the weapon collection system. There is sufficient risk that weapons can be lost trying and failing to beat a group of enemies, versus the potential reward of at least on-par, if not even better ones if they are successful. Elsewhere, an element of risk-reward exists in the cooking loop, since the first time the player tries a recipe, they will not know what it will be, or if it will even succeed at

all. They are still motivated to risk it however by the alternative of having nothing at all, making the game more difficult by making combat more punishing, less able to A) deal with threats through elixir buffs, and B) deal with the consequences, as they won't have made any food and thus cannot heal on the fly, instead having to travel to the nearest hotel or stables, then pay to heal themselves.

Loss and Avoidance

- From Small-Scale to Large-Scale

A sense of Loss and Avoidance on a smaller scale can be made through equipment scarcity. If the player broke a load of weapons on a single enemy or group of enemies, and got less out of it than they started with, the player would quickly run out, and have to rely either on disarming enemies, which is riskier, or using other weapons such as bows instead of melee weapons, wasting those weapons instead. Players may then elect to be more careful and strategic about who they fight and why. On a larger scale than that, the player may potentially spend a long time fighting a boss, and either lose near the end, forcing a restart of the whole thing for lost progress, or come out of it with again very little equipment, food and elixirs if they were unprepared for it.

- Cropping your Losses

If the player started to take some losses, e.g. against a boss losing a few weapons and some food//elixirs, and it was obvious that a win was unlikely, an open-world boss is escapable, the player able to crop their losses, and head off, re-build and come back better prepared.

- Grace Systems and Maintaining Credibility

Grace systems would include consumable "Fairies", which sit in Link's inventory, and act as revives when he is defeated, with the fairy flying off and Link getting back up with some health restored.



ITERATIVE DOCUMENTATION

Design Outputs Iteration 1

Villager Roles (Alphabetical A-Z)

- Alchemist

The Alchemist will work similar to the Cook, who produces meals from ingredients prepared by other villager roles. Here, the Alchemist will take monster parts and foraged ingredients to produce various Elixirs, that could be used either by Link or villagers, or sold in the shop for additional income.

- Animal Handler

The animal handler will deal with any domesticated animals and gather any animal products in the process. These will then be taken by Labourers to different locations as required. Each animal will have a set purchase cost, that Link will have to pay to get them, at which point a pair will then start to breed, meaning Link need not buy more. Advancing time by waiting near a campfire would pause the breeding timer for the same amount of in-game time, as this could be interpreted as . These animals and products include:

- Cucco (chicken) – Purchase cost 100 Rupees, breeding time (days) 3
 - o Bird Egg (1 heart)
 - o Raw Bird Drumstick (1 heart)
 - o Raw Bird Thigh (1.5 hearts)
- Hateno Cow - Purchase cost 250 Rupees, breeding time (days) 7
 - o Fresh Milk (0.5 hearts)
 - o Hateno Cheese (1 heart)
 - o Raw Gourmet Meat (3 hearts)
 - o Raw Prime Meat (1.5 hearts)
- Highland Sheep - Purchase cost 200 Rupees, breeding time (days) 4
 - o Fresh Milk (0.5 hearts)
 - o Hateno Cheese (1 heart)
 - o Raw Prime Meat (1.5 hearts)
 - o Wool
- White Goat - Purchase cost 150 Rupees, breeding time (days) 4
 - o Fresh Milk (0.5 hearts)
 - o Goat Butter
 - o Hateno Cheese (1 heart)

- Raw Meat (1 heart)
- Wool

- Blacksmith

The Blacksmith will take ores and wood from the lumberjack and miner to produce tools for the different professions, e.g. Farming Hoes for the Farmer, Sledgehammers for the Miner etc. They will also make weapons and armour for both Link and villagers, namely the Guards. These will be taken by labourers to a different location, e.g. storage or the different worker buildings, e.g. miners hut, guard barracks etc. Each item would also take a different amount of crafting time, e.g. a full armour body (Soldier's Armor) would take much longer than a tool (Farming Hoe).

Tools

- Farming Hoe (Farmer)
 - Damage - 16
 - Durability - 6
- Iron Sledgehammer (Miner)
 - Damage - 12
 - Durability - 40

Weapons

- Soldier's Broadsword
 - Damage - 14
 - Durability - 25
- Soldier's Shield
 - Armour - 16
 - Durability - 16

Armour

- Soldier's set (Helm, Armor, Greaves)
 - Defence value - 4 per piece

- Builder

The Builder is the one actually building the buildings, not Link, who instead places and assigns them to be built. The Builder will use resources such as stone and wood from the main storage building to build them, with different costs per building.

AI Breakdown

- Will work on buildings in order of Link placing them
- Will go and get any materials available and deposit in building

- Any materials are worked on to fill a progress bar, and then go and check for any more available materials

- Carpenter

The Carpenter will be supplied with wood, who will then make wooden furniture to furnish the buildings in the village. This means the village will slowly get more lavish over time and look better, but provides no gameplay benefit.

- Cook

The Cook is responsible for taking raw ingredients supplied either by Link or Fishermen, Animal Handlers, Farmers etc., and making them into consumable meals, both for the villagers and Link.

- Takes any ingredients produces by the different roles to make a variety of meals. This could work by going through a preset list of meals, checking for available ingredients, and doing so on-loop.

- Doctor

The Doctor would heal anyone injured as a result of monster attacks. The player need not necessarily build one, but injury recovery time would be accelerated by the presence of said doctor. Talking to a doctor with patients in the doctor's/hospital would tell the player estimated recovery times of each patient. Talking to the doctor while injured would also allow Link to slowly recover health, similar to how the hot springs around the map heal him.

- Farmer

The farmer would over time produces a variety of ingredients, as long as they have the right tools (Farming Hoe), based on Link designating which ingredients each farm would produce. These could then be taken to the Cook to make meals for the village. These ingredients would change the look of each field, for instance tree-growing fruit would convert the field to a mini orchard.

"Fruits"

- Apple (0.5 hearts)
- Hydromelon (0.5 hearts) (heat resistance)
- Mighty Bananas (0.5 hearts) (increased attack damage)
- Spicy Pepper (0.5 hearts)

"Wild Greens"

- Swift Violet (increased movement speed)
- Warm Safflina (cold resistance)

Mushrooms

- (Big) Hearty Truffle (4/2.5 hearts) (temporary hearts)
- Endura Shroom (0.5 hearts) (stamina recovery)
- Hylia Shroom (0.5 hearts)
- Ironshroom (0.5 hearts) (increased defence)
- Razorshroom (0.5 hearts) (increased attack damage)
- Rushroom (0.5 hearts) (increases movement speed)
- Stamella Shroom (0.5 hearts) (stamina recovery)

Meat

- Raw Gourmet Meat (3 hearts)
- Raw Whole Bird (3 hearts)
- Raw Bird Thigh (1.5 hearts)
- Raw Prime Meat (1.5 hearts)
- Raw Bird Drumstick (1 heart)
- Raw Meat (1 heart)

Seasonings

- Courser Bee Honey (2 hearts)
- Acorn (0.25 hearts)
- Chickaloo Tree Nut (0.25 hearts)

- Guard/Soldier

The Guard/Soldier would act as a village guard, protecting villagers from monster attacks, and taking injured villagers to the Doctor. They could either stay in place guarding a specific location, patrol around the village and surrounding areas, or escort certain villagers. They would be equipped with weapons and armour from the Blacksmith and Fletcher.

- Labourer

Labourers are the backbone of the village, moving goods around from location to location, e.g. taking ore from the miner's hut to the Blacksmith, or food from the Animal Handler to the Cook. Without them, the other roles would have to do the hauling themselves, taking them from actually making things and do their jobs proper. Labourers would also help the Builder to move resources to planned buildings, such as wood and stone.

- Lumberjack

The Lumberjack would go into the world and chop down trees for wood. Trees remember their locations and respawn in-game, thus there would be no danger of permanently scarring the landscape, since when the player closes and re-opens the game all the trees would respawn.

- Miner

Like the Lumberjack, the Miner would go out into the world, this time looking for ore deposits, from which one can mine a variety of materials, including valuable gemstones like Rubies or Sapphires, Rock Salt used for cooking, or in this case Stone, which would be used to build the village. They would be armed with the Sledgehammer to do so.

- Shopkeeper

Somewhat uniquely from other roles, the Shopkeeper would be sent excess materials, to then sell to NPC characters, wandering traders etc. These would whenever possible use the same prices as items found in other stores. At the end of each day the Rupees would be deposited somewhere for Link to access. To prevent abuse by the player, by skipping ahead of time to get free money they haven't earned, the system will freeze for the same amount of time that is skipped. Skipping time multiple times in a row will add the skipped time to a total freeze value. This would be communicated by a "Closed for maintenance" sign on the door.

- Tailor

The tailor would take produced Wool from the Animal Handler to produce a range of clothing sets for the villagers, much like the Blacksmith does Armor for the guards. These would not be taken by Link, unlike most other roles produce, since in so doing he could get several clothing sets basically for free, or at least heavily discounted, instead of going out into the world, finding them and purchasing them as intended.



Villager Management

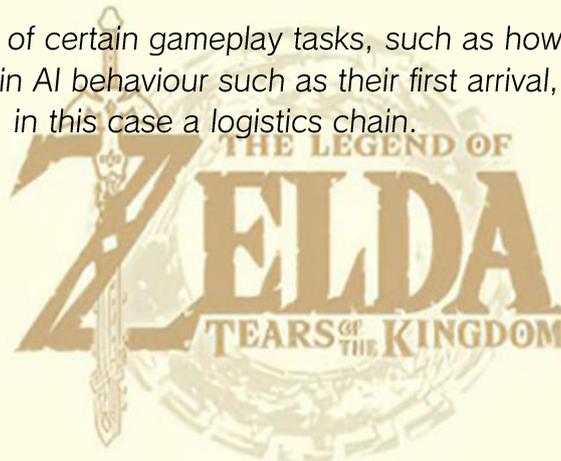
Each villager would have different needs, such as hunger needs. These would go down over time and be replenished when they eat at X time of day. If not enough food is available, they will get increasingly slower at their work, e.g. longer crafting times, slower gathering and fishing etc. If the problem for any villager is too much they will threaten to, and then actually leave the village, unassigning their role.

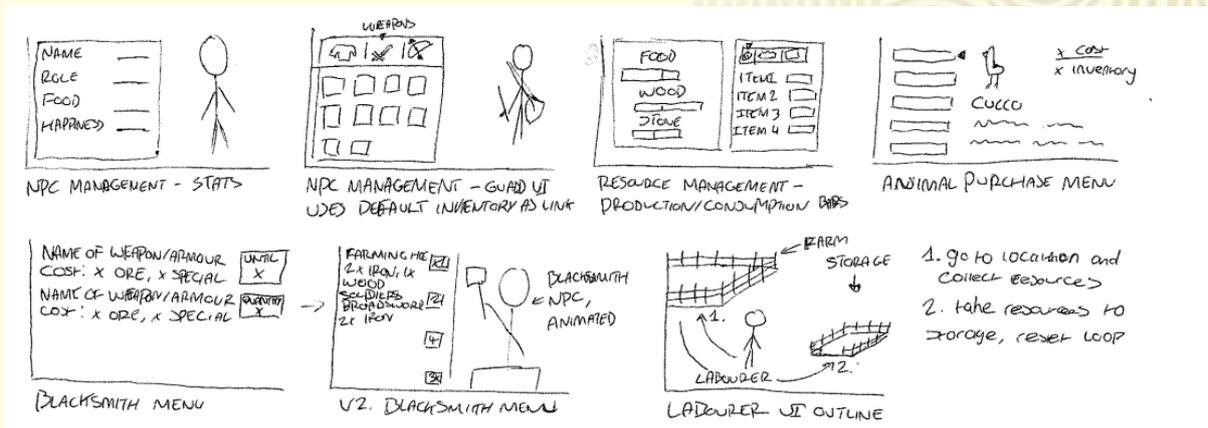
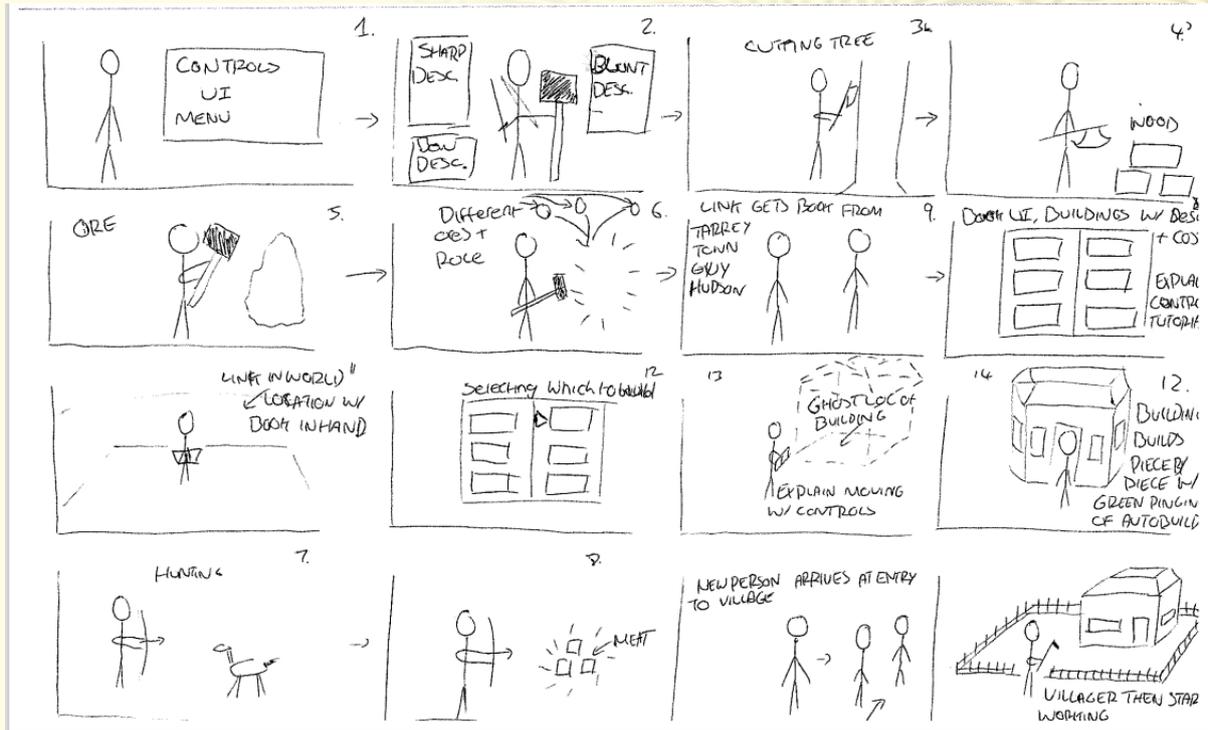
Outline of Prototype Contents

In designing version 1 of the prototype, this ended up looking more like a story-boarded walkthrough of the game using Microsoft Office PowerPoint. This used wireframes for the different UI elements that would have been highlighted, including:

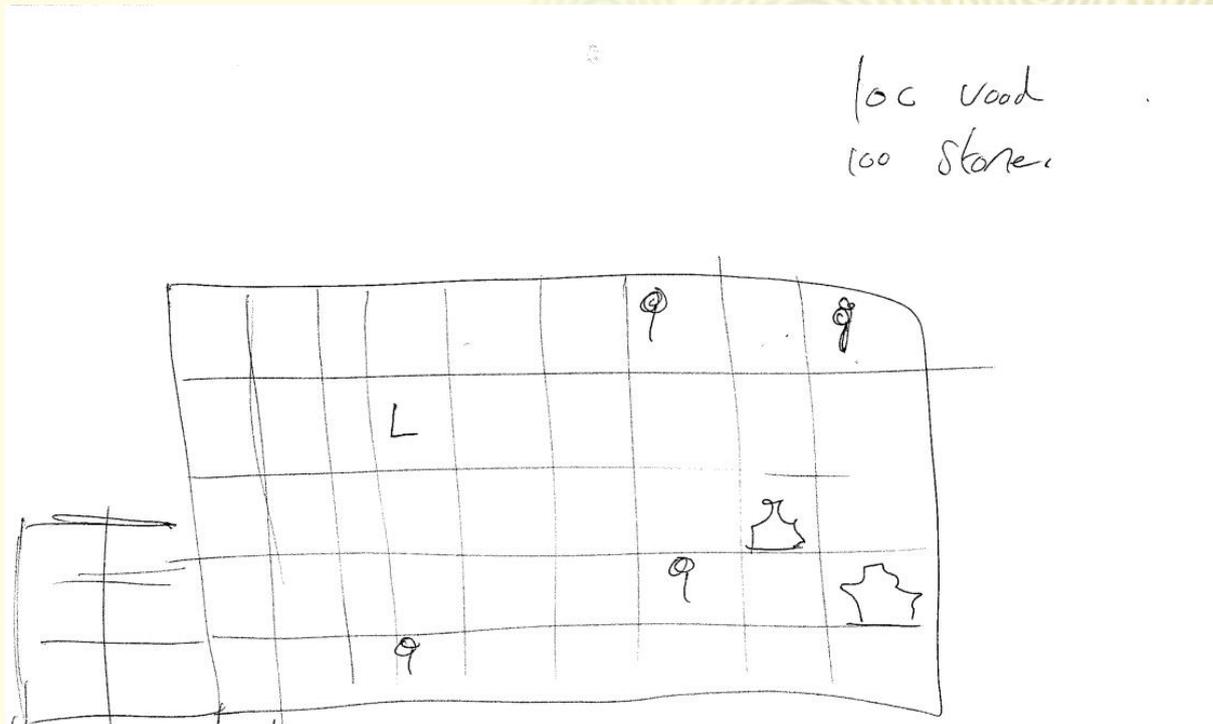
- A book of village building recipes and sizes for each building, here obtained from "Hudson", who in the prequel "The Legend of Zelda: Breath of the Wild" (BoTW) Link helped to build a settlement "Tarrey Town".
- The individual villager stat and management menu. Here, the player can see a villager's name, role (e.g. builder, fisherman etc.), hunger need and happiness.
- The Guard/Soldier role equipment menu. Here, the system uses a UI similar to Link's inventory, with armour, melee weapon and shield, and bow and arrow categories.
- An overall village management menu, including visualisation of food, wood and stone production versus consumption, e.g. food produced versus consumed by the villagers, or wood and stone produced per day versus how much is needed for placed building plans.
- The animal purchasing menu, which is essentially the regular shop menu but with animals added instead of the usual goods.
- Blacksmith production queue menu, with a list of products to be made or upkept to x amount, with a version 2 showing the blacksmith villager animated to the side working.

Here also is an outline of certain gameplay tasks, such as how to collect resources, how to build and certain AI behaviour such as their first arrival, and an outline of the Labourer AI behaviour, in this case a logistics chain.



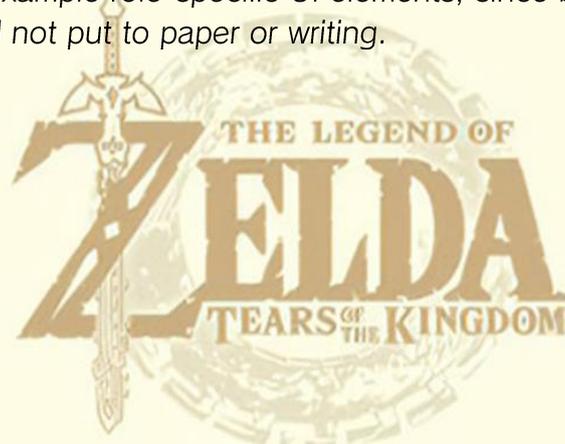


In discussion around making the actual prototype, a mock draft was written up, which helped to clarify the contents and scope of said prototype. Here, a mock grid was made up representing the game space, with an additional side grid representing the village space. The prototype would then be a high-level version of the system, here Link moving around the map collecting resources, then back to the village and building. These resources would be stockpiled and one of the measured player performance stats.



Iteration 1 Evaluation

Iteration 1 was useful for the sake of outlining elements such as the roles, how they work and what they produce or contribute to the village gameplay. The paper diagrams meanwhile were useful to communicate Link's end of the gameplay, including how systems such as resource gathering, designing and building the village, and outlining several example role-specific UI elements, since before this ideas were mainly conceptual and not put to paper or writing.



Rationale for Changes to Iteration 1

A “Heap” is a system with a game that has little effect on the rest of the gameplay or other systems. One of the main points of feedback during development of the project was that Link’s abilities, such as “Autobuild”, which automatically builds the “Zonai Devices” new to ToTK, currently existed as one of these “Heaps”, with no effect on the rest of the village building gameplay. From here, two major problems would need to be addressed:

- As mentioned, address the abilities heap, tying in at least 1, if not more into village building gameplay
- Actually design a proper prototype that could be played by someone, as opposed to a non-playable walkthrough or tutorial as in iteration 1

Design Outputs Iteration 2

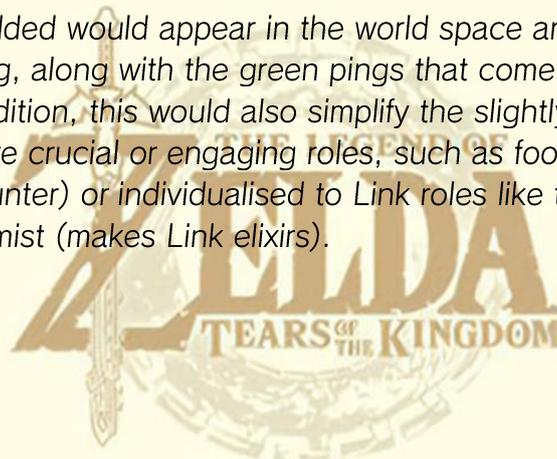
List of changes made to villager roles (plus solving the abilities heap issue)

- Removed the Builder role

In tackling heaped systems, such as the abilities system, the mention of “Autobuild” presented the chance to remove the “Builder” role, whose existence was essentially arbitrary anyway, while tying in player abilities. Here then, as mentioned in the original paper diagrams, the building would be constructed piece by piece, taking ingredients out of either Link’s or the village’s inventory, and adding to each building, snapping parts to it like a Zonai Device as follows:

- First, stone would be added to form the building foundation
- Then, wood would be added to build the wooden frame of the building
- More wood would then build the walls, doors and windows
- Yet more wood would be added to form the roof beams
- More stone would be added in the form of roof tiles, a chimney etc., thus finishing the building

Each of these when added would appear in the world space and snap to the ghost blueprint of the building, along with the green pings that come with using the Autobuild ability. In addition, this would also simplify the slightly long list of buildings, saving villagers for more crucial or engaging roles, such as food production (Farmer, Fisherman, Forager/Hunter) or individualised to Link roles like the Shopkeeper (makes Link money), or Alchemist (makes Link elixirs).



- Removed the Tailor role

Like the Builder role, the Tailor existence as an oddity, as opposed to a directly impactful gameplay feature. The idea that Link could not take the clothes, thus getting them for free instead of finding them in the game world, e.g. a dungeon or chest, or purchasing them from another village or town, made the system less engaging. Additionally, individual villagers are not meant to be massively customisable and individualising them could instead come from role-dependent clothing being set automatically. This also meant the removal of the Wool product, thus simplifying the Animal Handler production chain, and reducing the workload both on them and Labourers.

- Removed the Cook role

Having a Cook, in retrospect, implied that everyone else in the village would be unable to cook, and that one or two people would have to cater for every person in the village. This wasn't terribly immersive, since realistically the villagers would go back to their homes after work and all cook for themselves, thus the Cook role was removed, further simplifying the list to 14 roles.

Outlining the contents and gameplay of the paper prototype

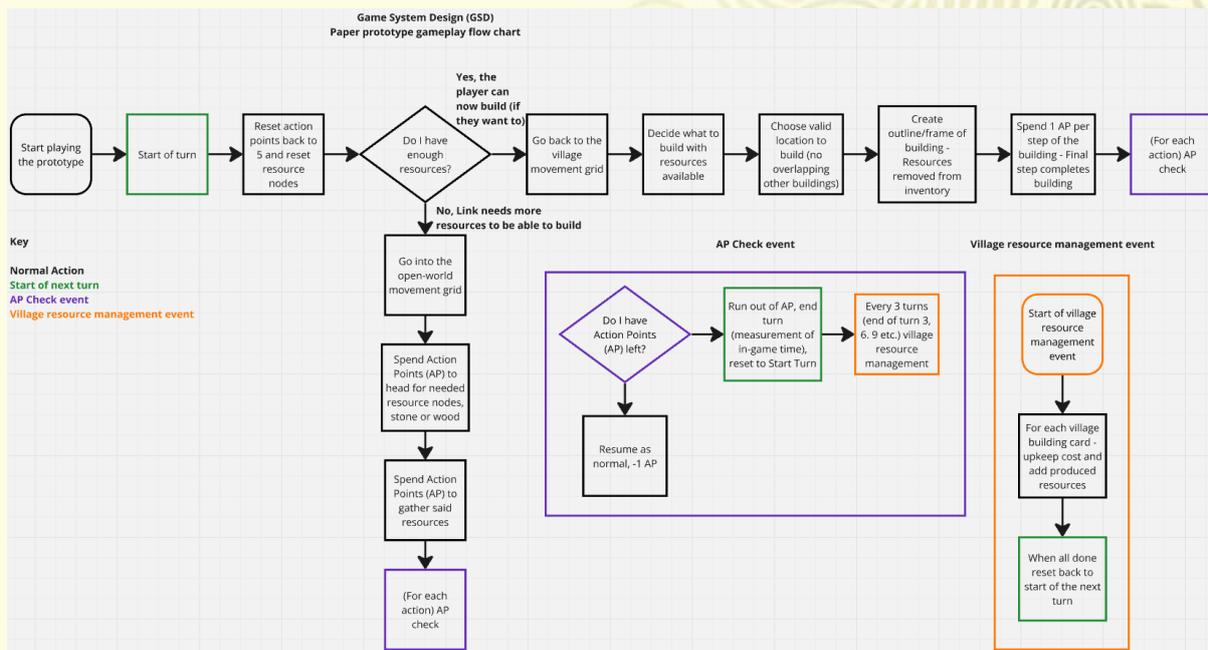
In outlining the contents of the prototype as an evolution of the feedback-based rough, the following contents was proposed:

- The game would work using a grid-based movement system with 2 maps, 1 for the open world map and 1 for the village, with Link able to move between them when he hits the border between them (as with the rough prototype). Link can also move diagonally between tiles.
 - o Separate grids for open world and village, the village grid used for placing buildings and the open world grid having resources
- The game will make use of action points (AP) and turns, with 5 actions per turn. Every 3 turns will represent a day, at the end of which the player will collect resources produced by the village, and spend resources on upkeep, mainly food to keep the prototype simple.
- Resource collection (tokens) (1x1 tiles) – Food (boars), Stone (ores) and Wood (trees) – Each resource is worth 5, so 5 food, 5 stone and 5 wood
 - o Resources are collected when Link walks into their tile and removed from the board, replaced when Link leaves and re-enters the Open-World grid from the Village Grid
- The game uses a simplified version of the building system, where Link can stand next to (but not on) tiles where he wants to build, and can place the blueprint (does not cost AP), with each building step taking 1 action point (see cheat sheet for steps for each building)

- Add building to grid (buildings grid-based) with buildings having different amounts of steps (action points)

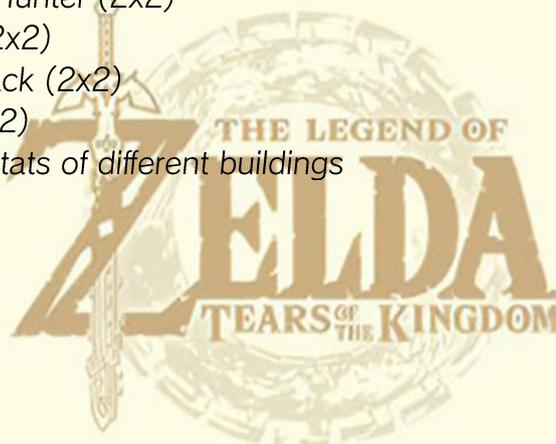
Making and playing the paper prototype

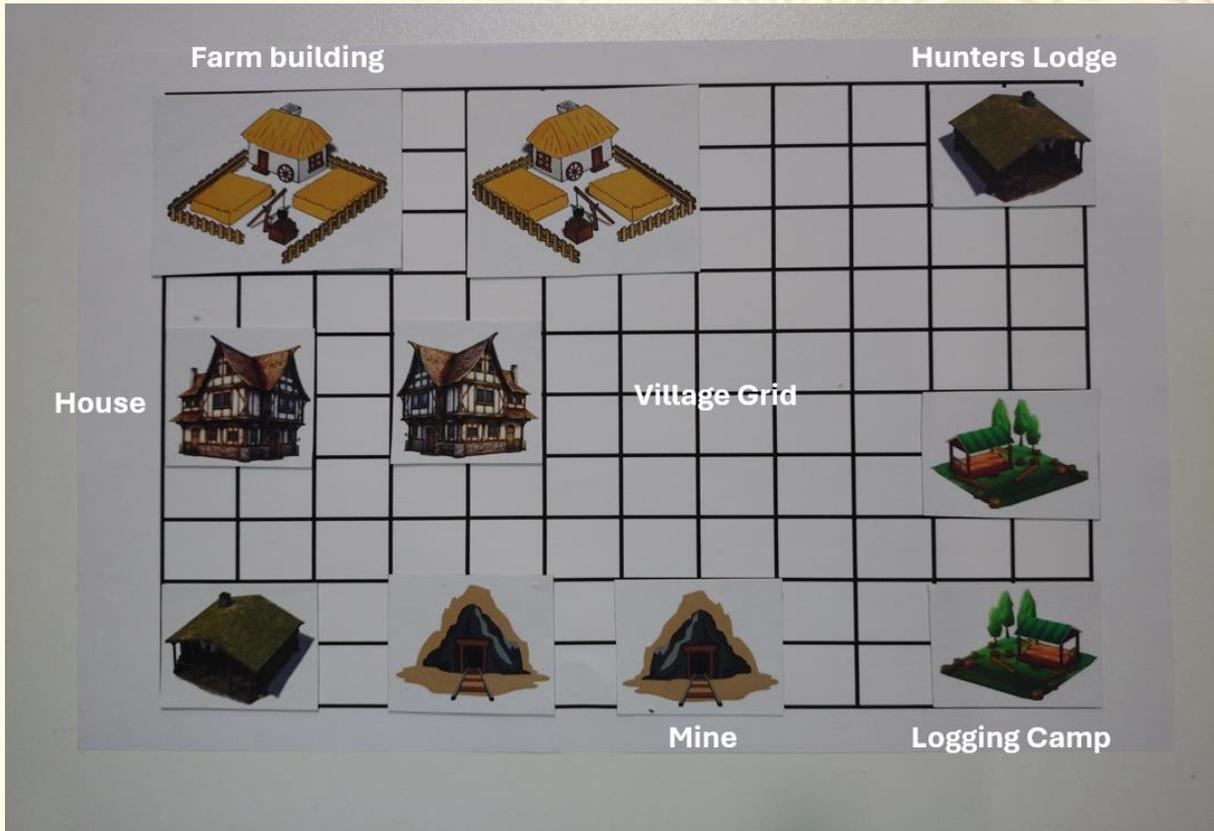
For the gameplay of the prototype, a simple flow chart was produced explaining the flow of a turn, including Action Point (AP) management, resource collecting (open world) and village management events (building the village and collecting resources from it).



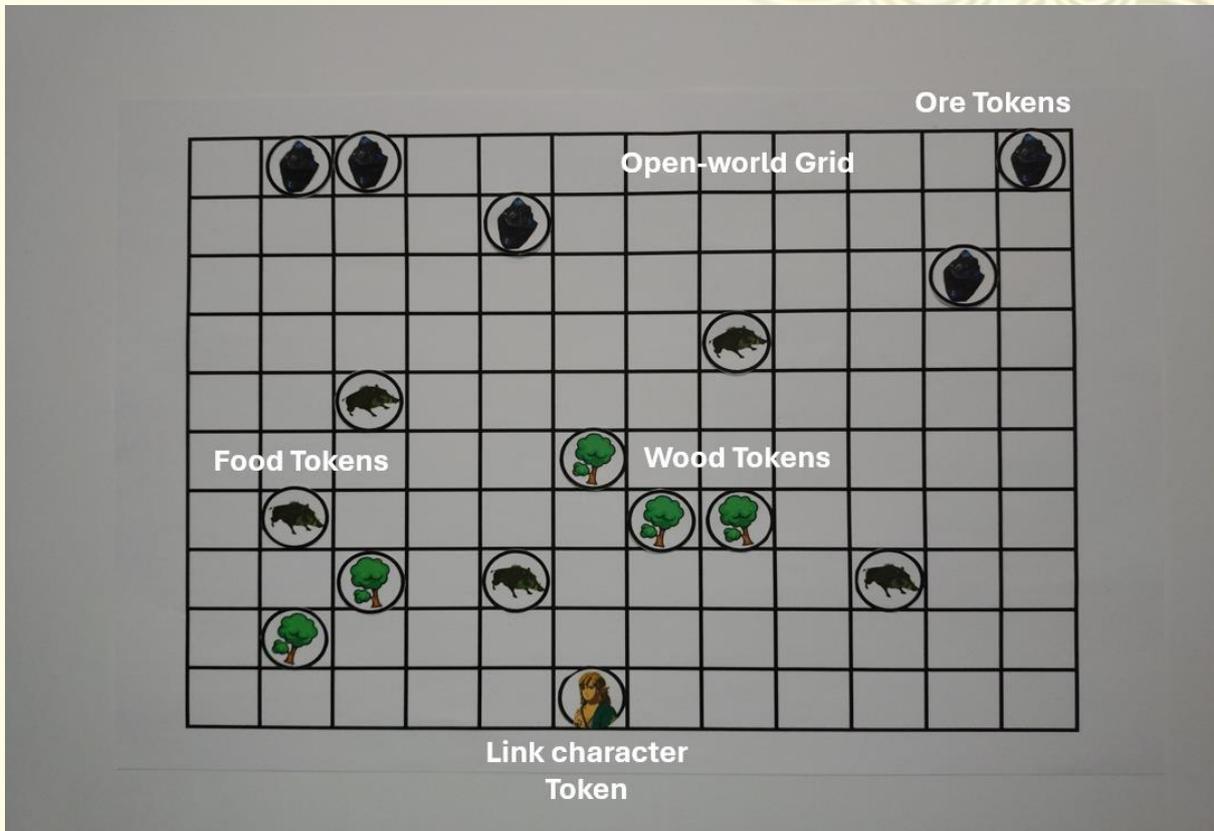
List of needed game pieces:

- 2 grid-based maps, one open world and one the village map (12x10 grids)
- Resource pieces, wood, stone and food (1x1)
- Link character piece (1x1)
- Example buildings for prototype
 - Farm (3x3)
 - Forager/Hunter (2x2)
 - House (2x2)
 - Lumberjack (2x2)
 - Mine (2x2)
- Cheat sheet – Stats of different buildings

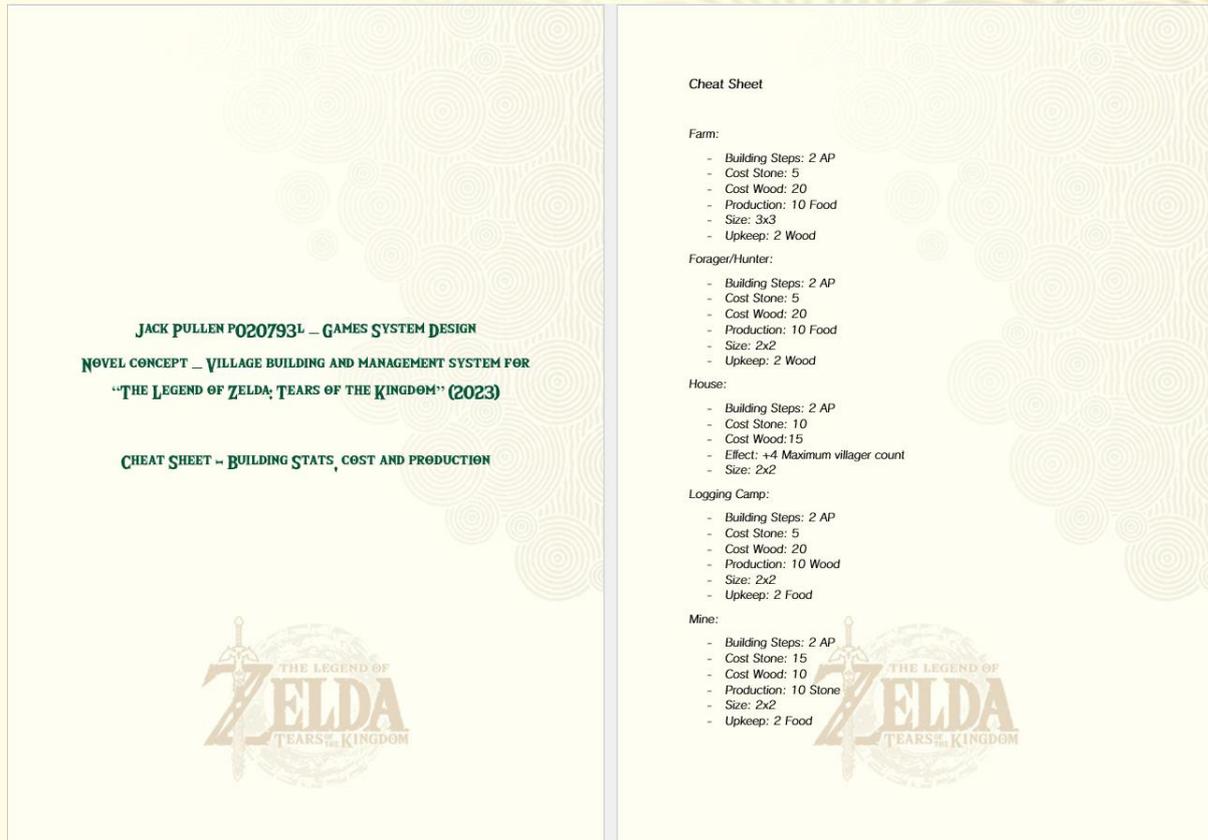




(Example village layout with different buildings as listed)



(Current Open-World grid with different resource nodes, with Stone hugging the sides, Trees in the middle and Boar spread out around the map)



(Cheat sheet image with list of buildings, sizes, costs and production of each)

Limitations of the Paper Prototype pre-test and mitigating said limitations

- The prototype is very high-level, with some features of the design not able to be implemented into the paper prototype for different reasons.
 - o The real system in ToTK would not be turn based. Here, turns instead represent the passing of in-game time, with every 3 turns representing 1 in-game day
 - o There is a lack of low-level content, such as the different production queues, individual villager happiness etc. This has been designed for the main game however, it was just not practical to implement here, the player having to manually set every villager's happiness every turn
 - o There was no way to emulate in-game events like monster raids, since there would be no immediately logical way to implement both Guard and enemy AI into a paper prototype. They are partially designed however with the different equipment stats
 - o It did not make sense here to implement the random loot tables for roles such as the Forager/Hunter, who would collect a random assortment from his loot table every day. However, for the paper prototype this would not matter, as food is food, since link would not be collecting different ingredients to cook different meals in the paper prototype

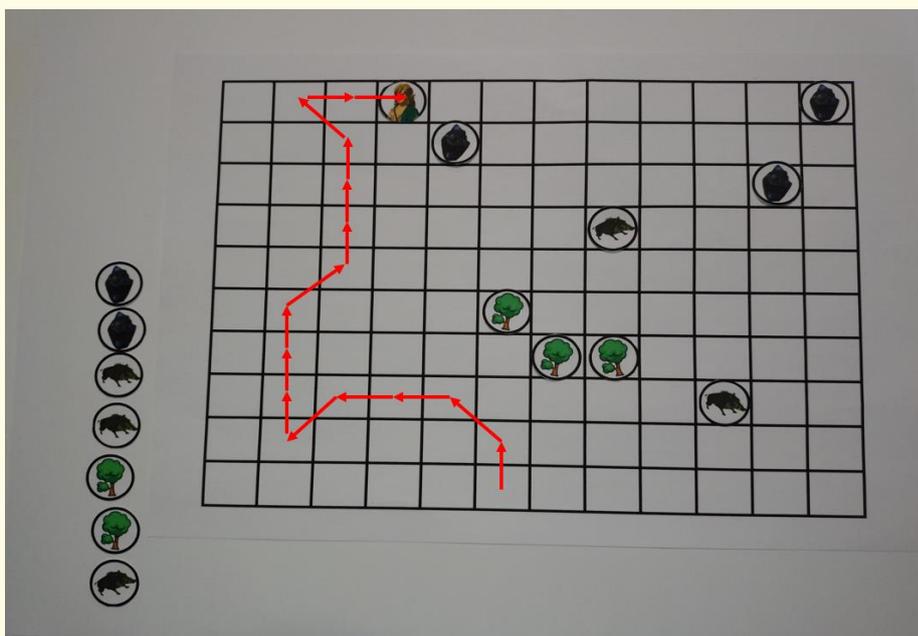
- Another issue is that visualising the village a player builds in the paper prototype compared to in ToTK would be vastly different, since 3D space allows for more freedom, and the ability to walk around and see it from different angled, as opposed to just the top down.

Playtest of paper prototype

Turns 1-3 (day 1)

- +15 food, +10 wood, +10 stone
- 0 Village upkeep
- 0 Village production

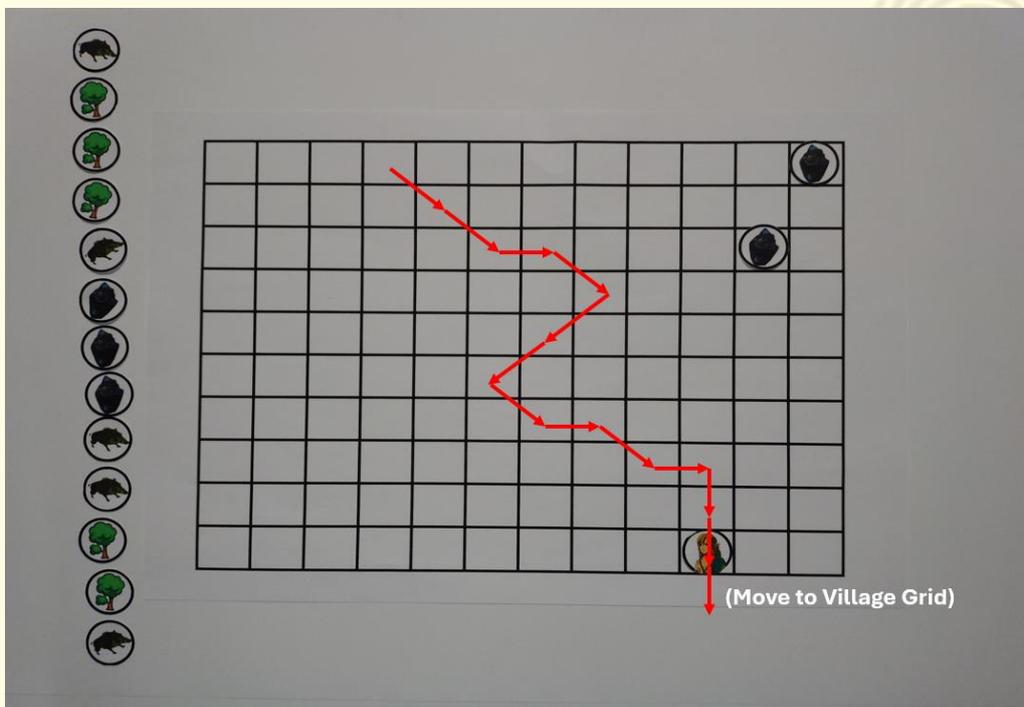
	13 (stone)	14	15 (stone)							
		12								
		11								
		10								
		9 (food)								
	8									
	7 (food)									
	6	4 (wood)	3	2 (food)						
	5 (wood)				1					
					Start					



Turns 4-6 (day 2)

- +10 food, +15 wood +5 stone (total 25, 25, 15)
- 0 Village upkeep – 1 House built (-10 stone, 15 wood) (new total 25,10,5)
- 0 Village production

			Start								
				1 (stone)							
					2	3					
							4 (food)				
						5					
					6 (wood)						
						7 (wood)	8 (wood)				
								9	10 (food)		
									11		
									12		



Resultant thoughts around playtest and immediate improvements

In doing just 2 in-game days of testing there were several notable elements of the games, broken down into pros and improvements for next time the prototype would be tested. Here were also provided some counterarguments to in-game reasons as maybe not to apply said ideas.

Pros

- Testing was actually fairly fun given the whole prototype was moving tokens around a grid. The planning of movement was involved, but not to the point of getting stressed over an optimal path. Many players in ToTK do not follow an optimal path anyway, and as referenced in the MDA framework analysis (Hunicke et al, 2004), non-optimal pathing allows for the dynamic of open-world exploration, and aesthetics of discovery and fantasy.
- The prototype successfully communicated the core concepts of the system, namely resource collection and spending said resources, open-world exploration, building selection and management etc.

Possible improvements for next time (with counterpoints)

- The fact that a day was limited to 15 actions meant that to collect enough resources to build 1 house with some food took nearly 2 full days, with any movement inside the village grid making even the house impossible. 15 actions per day, split into 3 turns of 5 actions felt too short, like Link could realistically do much better in-game than here.
 - o In retrospect the idea of 3 turns of 5 actions was A) weird and B) arbitrary, since only every 3rd mattered, and so they were basically turns of 15 instead
 - While playing the prototype, 2 different ideas emerged, either extending the AP count to 24 in a single turn, thus A) getting rid of the weirdness of multiple smaller turns, and B) tackling the shortness issue
 - Another idea was to instead have 2 turns per day of 12 actions, with the upkeep and production of all buildings halved, in exchange for producing twice per day
 - That being said, when coming up with the original concept, the village building was designed on the expectation that it would be a slow-paced, hands-off system, with Link periodically coming back, spending any resources he had on him (e.g. building), and collecting any produced by the villagers, then running off again

- Another concern was that resources only dropping 5 per node was quite low, and that this would not really be enough to do anything with
 - o In the playtest, collecting nearly everything had Link barely able to afford 1 house with nothing else, besides from having a lot of food for the villagers on him, thus maybe saving Link from having to build food production buildings immediately, instead something like a Mine or Logging Camp
 - 2 solutions that arose during the playtest were either to increase the maximum nodes of each resource from 5, or to have each node give more than 5 resources, e.g. 10 instead
 - Again however, this ran into the hands-off, slow-paced design issue, wherein the player could max out a village in only a few days of playtime and then have nothing to do but collect resources that Link couldn't do anything with

With these few changes made (or not, instead changing expectations of the prototype) the game would be much more fun and approachable, if also quicker paced than originally designed for.



REFERENCES

- ⁱ Hunicke, R., LeBlanc, M. and Zubek, R., 2004, July. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI* (Vol. 4, No. 1, p. 1722) [Accessed 7 Nov 2024].
- ⁱⁱ Ruhi, Umar. (2015). Level Up Your Strategy: Towards a Descriptive Framework for Meaningful Enterprise Gamification. *Technology Innovation Management Review*. 5. 5-16. 10.22215/timreview/918 [Accessed 7 Nov 2024].
- ⁱⁱⁱ Lazzaro, N. (2004). *The 4 Keys 2 Fun* | Nicole Lazzaro's Blog. [online] Nicolelazzaro.com. Available at: <https://www.nicolelazzaro.com/the4-keys-to-fun/> [Accessed 17 Oct. 2024].
- ^{iv} (Björk et al, 2005). *Patterns in Game Design*. [online] Google Books. Available at: https://books.google.co.uk/books/about/Patterns_in_Game_Design.html?id=IFQfyODK4wAC&redir_esc=y [Accessed 17 Oct. 2024].
- ^v Csikszentmihalyi, Mihaly. (1990). *Flow: The Psychology of Optimal Experience*.
- ^{vi} Chou, Y. (2015). *Octalysis: Complete Gamification Framework - Yu-kai Chou*. [online] Yu-kai Chou: Gamification & Behavioral Design. Available at: <https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/> [Accessed 17 Oct. 2024].
- ^{vii} Kho, J. (2022). *The Octalysis Gamification Framework for Corporate eLearning*. [online] VisuerLab. Available at: <https://www.visuerlab.com/2022/01/08/the-octalysis-gamification-framework-for-corporate-elearning/> [Accessed 21 Nov. 2024].

