Design Document

v.2018.04.09a

Developed by Andrew Gawoski 2018

Table of Contents

Overall Goals

Farming simulation games in the genre pioneered by the *Bokujou Monogatari* series are niche. The genre has close ties to games that include dating-simulation-type social systems which comes off as a very Japanese element in games, which isn't surprising as other standout examples are the *Fire Emblem* and *Persona* series as well as many games developed by Bioware. The *Bokujou Monogatari* series has made a number of changes over the years since pioneering the genre, a few of which might have been steps backwards.

The overall goal of developing *Dawnflower Chronicles* is to create a 3D-realized game with elements similar to older *Bokujou Monogatari* titles, while considering changes made in the genre over the last two decades. For inspiration, I'll look towards the *Bokujou Monogatari* series, the newer *Harvest Moon* games developed by Natsume, *Stardew Valley*, smaller attempts in the genre like *Orange Season*, and standout games that contain system overlap like *Animal Crossing* and the *Persona* series.

Current Design Standards

As time has passed and new entries in the *Bokujou Monogatari* series have been released, players are continuously presented with more choices than in prior titles. These days, *Bokujou* games often have 15 or more different crops (including flowers but not trees) to grow in a season. Not every option is unlocked or available to the player from the beginning of the game, but there isn't much mechanical difference between the various crops. They take different quantities of time to grow, have seeds that are purchased for some value, and produce fruit or vegetables that can be sold at another value. Some crops can be harvested multiple times and others can only be harvested once. *Stardew Valley* and the newer Natsume games have largely maintained this design and have a similar amount of seasonal crops. This creates a general optimization problem where players can maximize their daily profit given the limited amount of space available to grow crops.

Recently, developers have attempted to encourage players to grow a variety of crops rather than having a "mono culture" of the most profitable crops in the game. Arbitrary quantities of particular farm items are now used to unlock new features or crops in the game, sometimes encouraging players to make choices in order to further their progress in a particular game area. Due to configuration of unlock systems (sequential vs parallel), players may sometimes find themselves waiting the majority of an in-game year to grow the crops necessary to progress in the game.

In a genre with multiple varieties of gameplay, the *Bokujou Monogatari* series of games has both moved towards greatly increasing the area players are capable of growing crops in and moved away from needing to tend to crops individually. These choices likely cater to those who want to grow as many crops as possible as well as those who want to spend less time farming and more time with the social aspects of the game through NPC interaction. *Stardew Valley* and other games outside of the *Bokujou* series have maintained the individual

crop design pattern, although most games in the genre allow for some form of assistance in watering crops as it is the most repetitive player action in the crop tending process (*Stardew has sprinklers*).

Geographically, most games in the genre have one or more towns, and a large amount of wilderness usually consisting of a forested mountain with several water features such as rivers, lakes, and sometimes ocean. The forests in these games are usually pretty open and meadow-like, with the majority of trees acting as natural barriers for the area. The mountain is usually created through a gradual increase in area elevation and can be completely forested. It allows for more scenic views of nature, sometimes contains a resource mine, acts as an indicator of geothermal activity for hot springs, and in the *Bokujou* series in particular often has a small pond where a folkloric “Harvest Goddess” resides. Water features primarily allow for area divisions, an increased variety of fish habitats for simple fishing mini-games, and also serve to increase the general aesthetic appeal of in-game areas. NPCs from town can often be found walking through these areas as they go about their daily business, in addition to small timid creatures like squirrels and foxes. Often, several different sizes of rocks, large stumps, smaller branches, wild mushrooms, flowers and fruit can be foraged from this wilderness (found on the ground).

This genre of game heavily features a social system based on increasing affection and or friendship with many different NPCs. Although a fairly simple level system, players are rewarded with unique dialogue options and dialogue-heavy character-developing events upon reaching certain levels. Experience/friendship/affection is increased by speaking to the NPC, giving them gifts, fulfilling their requests or participating in public celebrations or festivals. Recent games have provided more rewards for gaining friendship with non-romantic NPCs such as unique cooking recipes, but the one of the expected “endgame” goals is a romantic relationship with, proposal to, or possibly marriage to, one of several eligible NPCs. You see a similar system in Atlus’s *Persona* series and Intelligent System’s *Fire Emblem* series. Players

have come to anticipate this system, and the blend of farming and social interaction has become a sacred cow of the genre.

Towns typically have a general-store for purchasing seeds, and possibly ordering items that are difficult to find in rural areas. Ranch owners sell animals and animal-specific goods, although its possible some of the tools or goods can be sold at the general store. Often there is a blacksmith of some sort that sells farming tools and upgrades them with resources the player collects, furthermore they can act as a carpenter in upgrading and creating buildings on your farm. However, some games have separated these into different characters. Usually you've got an inn/boarding house of some sort which sometimes contains a restaurant, although often enough the restaurant can be a separate building entirely. Additionally, a clinic is necessary if only for the purpose of selling medicine and acting as an area to lecture the player when they run out of stamina. There is almost always a mayor, who acts as an MC for events, which often take place in a town square or festival area. Also, there's usually a hot-springs area just due to Japanese convention, but *Stardew* maintains this as well. Crops, and other resources to be sold are placed in shipping box near the (primary) entrance to the player's farm rather than needing to visit an NPC at a store.

Pattern Criticisms

Bokujou Monogatari's change to watering and harvesting larger quantities of crops at once developed from just having **too many** crops to take care of. Anecdotally, in an older *Bokujou* game, I had so many crops planted I could not harvest all of them with the time given in a single day. This might have been caused by a difference in the quantity I could tend to and harvest at once. I think it is reasonable to design a game where players cannot plant more crops than they can possibly tend to in a single day, with certain exceptions.

Additionally, there is a needlessly large number of different crops that can be grown in a given season. While increasing the difficulty for players seeking 100% completion in every aspect of gameplay, recent games in the genre have included checklist-style gameplay and gated progression behind growing an arbitrary number of a particular crop. As new crop seeds are also a form of player progression, often these unlocks are inconveniently timed and result in players not having the ability to make choices that they want. The crops usually devolve into a few ideal choices, and the large variety of crops doesn't mean much; the choices are slightly overwhelming.

The *Bokujou* series has moved away from it's older grid and floor-based mine exploration, choosing instead to have a small handful of colorful rocks or cracks in the ground players can hit with their hammer several times daily. The newer Natsume games have a similar design. However, *Stardew Valley*, despite being a newer game implements the style of mines. While I dislike *Stardew's* combat and mine/cave aesthetic, I prefer having plentiful resources where players can continue to search for the items they want. Otherwise, if the game requires certain rare items to progress, players will reset their game until the appropriate item is generated in the world for them.

Bokujou series games have seasonal festivals which are competitive but their results can be highly random. *Stardew Valley* doesn't include festivals that are competitive in the least. Players resort to save scumming in *Bokujou* in order

to receive anything at all from festivals, and the festivals in *Stardew Valley* aren't as exciting as they don't tangibly increase player progression.

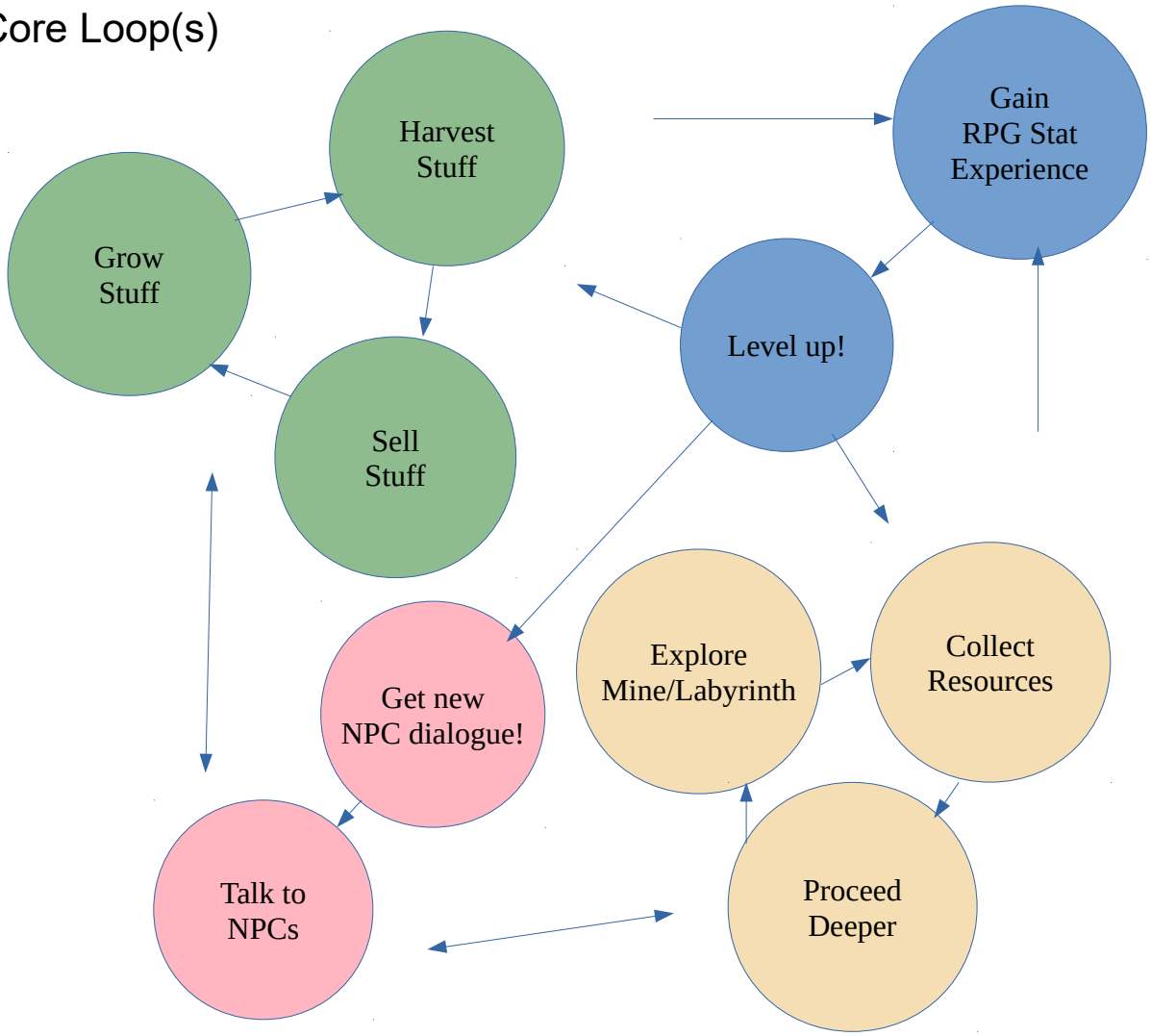
Social systems used in these games are alright, but haven't changed much since the creation of the genre. Recent RPGs with social systems like the *Persona* series of games do similar, but add additional and meaningful complexity by tying it together with RPG elements (player stats) and the game's time management system. While the two aren't perfectly suited to one another, improvements can be made and I can look to other genres for inspiration.

Game Hook and Core Loop

Hook

Dawnflower Labyrinth is a farm-life-simulation game that improves player progression rate through tracking player desires. Different crop choices have noticeably unique growing patterns and NPC interaction integrates player stats in a meaningful way. Players can learn about and explore the world at their own pace as they gradually discover the mysteries of the multi-floor mine and the winding, changing, labyrinthine forest.

Core Loop(s)



Design Pillars

Pillar 1: Farming (Relaxation)

Players will care for animals (Cows, sheep, chickens, at a minimum) and tend to crops (fruits, vegetables, flowers, trees) at their own pace. A non-stressful source of income and progression, players plot and plan the layout of their fields and choose which kinds of animals to house in their barn(s).

It's slow, methodical, and a constant "I will take care of my cows tomorrow because they'll eventually make better milk" and "I will plant corn tomorrow so that I can start harvesting every other day, 8 or 9 days from now." It's relaxing as an optimization game, where the player only becomes more successful and can't really fail. Players have a routine.

Players have their choice of additional sources of income such as foraging and mining that they can choose to spend their time doing. "It's raining and I don't need to water my plants today and all of my animals are safe inside the barn? I can go spend more time doing other things today like talking to NPCs or exploring the world."

Pillar 2: Social NPC Interaction (Affection)

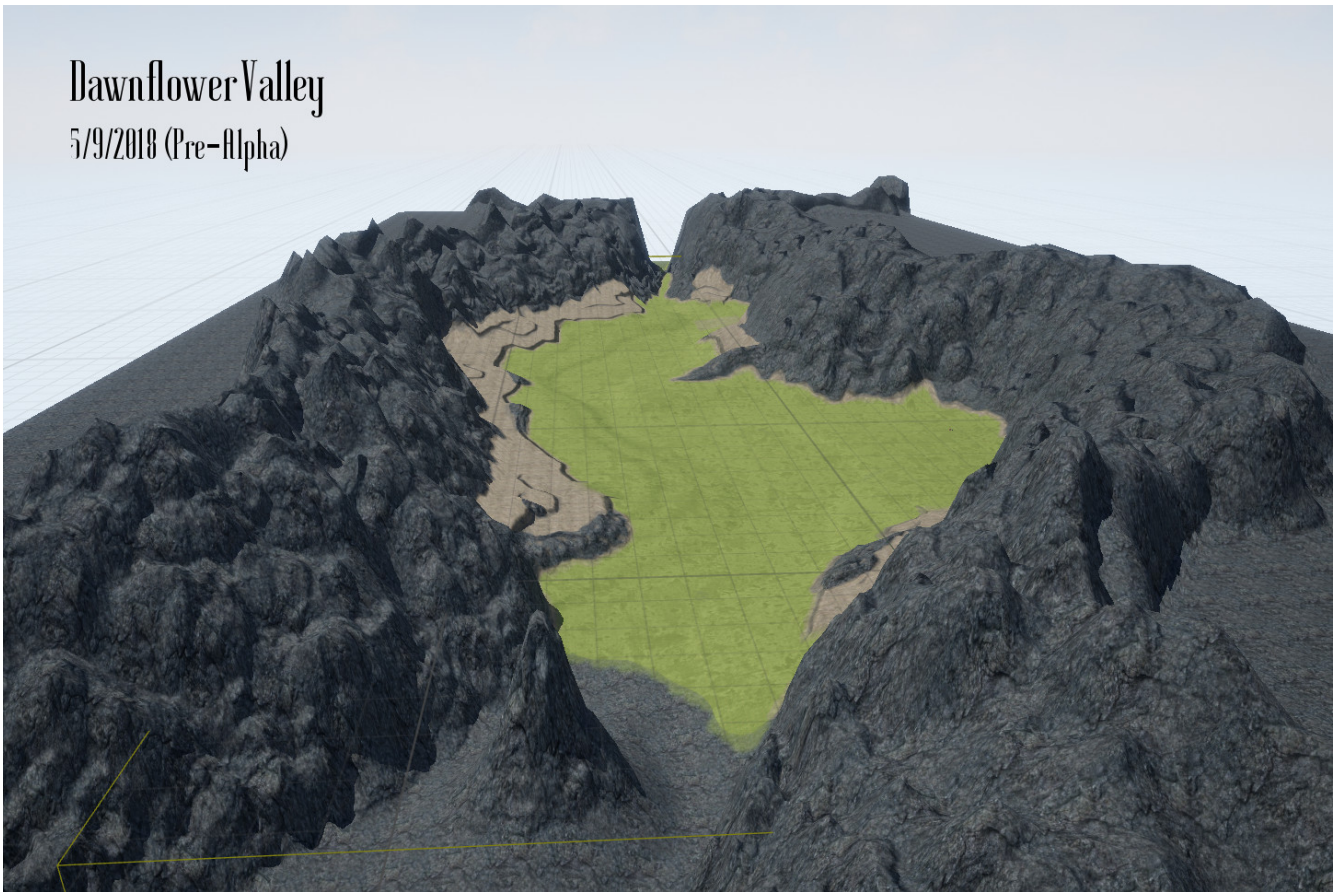
A town with NPCs that have lives and personality. Players come to understand the background and past of characters, and their relationship with other NPCs and the world. **The characters are the story** and they have flaws.

Players will be invested in the story through their interaction with characters. **Players will get to know, date, and even marry NPCs.** They can celebrate NPC birthdays (and NPCs will celebrate their birthday!), solve problems and attend social events/festivals. Players will experience a sense of community.

Pillar 3: World Exploration and Discovery (Wonder)

In addition to social exploration, which can be considered as general world-building as NPCs are part of the world, players also explore the natural world outside of the town. Players will learn how to move around the (world) map and establish a sense of direction. Players will discover the locations of natural resources like stumps, wildflowers, seasonal wild fruit and mushrooms in addition to where they can fish, where NPCs spend their time at different times of day, different days of the week.

Players will explore one (or more) mines, in traditional floor-based fashion and can discover the secrets of certain floors and learn just how far down it goes. Players will explore a mysterious labyrinth-like section of a forest which changes day to day. An area of true exploration, this labyrinth can contain special or out of season resources within its unique, procedurally generated layouts. This forest labyrinth also provides ample opportunities for hidden secrets for players to discover.



Pillar 4: Player Progression (Satisfaction)

Create a feeling of accomplishment for most actions. Water crops so they get bigger, harvest and sell when ripe. Care for animals as they produce resources and win festivals. Speak with NPCs to improve their level of affection/etc towards player, progressing their storyline. Leveling up player stats to farm and interact with NPCs more effectively. Expanding/upgrading player house and farm buildings. Upgrading tools to use them more effectively and efficiently.

As **an optimization game at heart** with numerous RPG elements, **players should constantly be feeling more powerful** (as a farmer/member of the in-game community). Incremental rewards to bigger payoffs. This includes in part the reward system, which involves festival participation and their prizes,

character dialogue for reaching certain social levels, upgrading the town, and obtaining objects/items/loot itself. Item drop RNG is manipulated to make sure players can find what they’re looking for.

General Plans & Features

Crops

The following is the initial list of planned crops. Specific crop details such as the period where they have flowers, their growth and harvest patterns can be obtained in a supplementary document. With the genre's Japanese roots, I feel it is essential to properly include flowering periods when possible, especially for plum, peach, and cherry trees.

Flowers

1. Snowdrop
2. Snapdragon
3. Pansy
4. Marguerite
5. Morning Glory
6. Sunflower
7. Rose
8. Forget-me-not
9. Garden Phlox

Ground Crops

1. Potato
2. Turnip
3. Strawberry
4. Cucumber
5. Corn
6. Cabbage
7. Radish
8. Onion
9. Melon
10. Tomato
11. Squash
12. Peppers
13. Peas
14. Eggplant
15. Sweet Potato
16. Carrot
17. Grape

Tree Crops

1. Plum
2. Cherry
3. Pear
4. Orange
5. Peach
6. Apple
7. Lemon

Animals

1. Horse
2. Chicken
3. Cow
4. Sheep

Tools

Tools are upgraded through a combination of money and raw, natural resources. As a convention of the genre, we have an upgrade path of copper, silver, gold and possibly more afterwards. Upgrading them typically modifies stamina consumption and increases their area of effect. Our upgrade tree, based on the mining prototype created earlier, could look like this:

1. Copper
2. Iron
3. Silver
4. Gold
5. Steel
6. Electrum
7. Mithril
8. Fabled

Fabled-tier tools aren't upgraded to, so it's possible for players to have say a mithril watering can and a fabled watering can. Fabled tools should be found or earned in some way, possibly given to the player by an NPC, and should have a unique, special and powerful effect when used. For example, a fabled watering can could make it rain for 1/3 of the day, watering every crop, but also actually making it rain.

Fabled tools could initially be very mundane and require appropriate level-thresholds or quest-like conditions to be completed before they become powered. Possibly the fabled watering can may only be filled up at a/the goddess spring. Maybe a certain fabled tool only appears in a particular season? These are some of the hidden secret endgame gameplay items.

Certain upgrades can be moderately but not severely time or affection-gated. It's possible that the town blacksmith cannot make an electrum alloy at the beginning of the game and needs the player's assistance to improve their forge/foundry in order to do so as part of one of their social events. It shouldn't be towards the end of their social events, as we want players that are invested in upgrading their tools but not invested in the blacksmith character to be able to do what they want.

Planting & Layout

Planting of crops should be grid based. Genre conventions dictate that sowing seeds apply to a 3 by 3 square area of the grid. A single tree also takes up the same area. This grid should be visible to the player, not in general but when appropriate in context: when using tools or items that interact with or affect objects in one particular area of the grid. This prevents players from planting seeds in the wrong place, accidentally cutting down their crops, failing to hit a boulder or stump as they were not properly aligned with it.

It might be worth considering making the game very organic feeling by using completely circular object collision, but it might be more trouble than it is worth. Such a design pattern would limit the ways we can store and access data about world objects as there are no longer a fixed number of possible object locations. Furthermore, the grid-based planting style is tried and tested in the genre; I would need a good reason to change it.

Depending on crop variety, players may be able to walk through crops. Obviously, players cannot walk through completely solid farm buildings. But, if we allow players the opportunity to construct farm buildings and choose their placement, we necessarily need to allow players to demolish farm buildings when they feel they need to. Players should be able to move their buildings, not needing to demolish and reconstruct them in order to do so. Given the absolute choice of where to place buildings, which is not a feature of earlier games in the genre *at all*, they are likely sharing a space with the general crop-planting area. As is traditional, players will construct fences to prevent animals from escaping and eating their crops. But of course, players can place fences and farm features decoratively.

Farm Upgrades

As is established design in the genre, players expect to increase the size of their house and construct a variety of additional farm buildings. Housing expansions can include features such as a larger/walk-in **bathroom**, tool/construction **shed**, **kitchen**, and expansion of existing areas providing space for a **larger bed**, necessary for certain social milestones among other furniture options.

These upgrades should be used more or less as follows. The bathroom restores some stamina, perhaps with a daily limit. The shed allows for creation and storage of farming machinery such as a device that creates seeds or turns milk into cheese; the shed should be accessible from outside and might not necessarily be connected to the house at all. The kitchen is used for food storage and general cooking activities; prior to obtaining a kitchen, players do not have specific locations to store fruit, vegetables, or typically-perishable animal products. A larger bed, specifically is a requirement typically tied to NPC marriage acceptance but does not prevent players from enjoying having a larger bed for its own sake if they do not have an NPC spouse.

Non-housing farm buildings should include barns and possibly stable(s), coops, essentially all standard shelter options for the farm animals the game supports. The aforementioned shed, wells/watering locations, and semi-functional decorative elements such as waterwheels in a river should also be considered. Waterwheels could be used to provide power, or heating or be some variety of progression block that enables one or more other upgrades.

Cooking

Cooking should provide a variety of benefits not limited to restoring stamina. While stamina restoration is a necessity of food in this genre, it isn't a compelling system when you have dozens of recipes that only restore different amounts of the same resource.

Looking at RPGs with alchemy or cooking systems that produce consumables like the Elder Scrolls and Monster Hunter series, in addition to the baseline effects seen in Stardew Valley and recent Bokujou games, inspiration begins to become apparent. Much like the systems in those action RPGs, the strength of the benefits of food items should come from the quality of the ingredients and perhaps the style of cooking. The types of effects should be dependent on recipe, which of course includes the style of cooking and the variety of ingredients themselves. Essentially, all cheese and potato dishes will be similar, with minor differences depending on whether or not it's a "fried" dish, or a "pot" dish, or a "grilled" dish, as examples.

Effects that could possibly be included:

1. Stamina restoration
2. Reduced stamina use
3. Increased run speed
4. Reduced illness chance
5. Increased (rare)item find
6. *Slow down time*
7. Increased experience gain
8. Increased affection gain
9. Fishing (something)
10. Palette cleanser (clear buffs)

Time Flow

The overall time-length of the day should be appropriately balanced that players very often feel like they have enough time, but not that they're going to sleep at 4PM. However, this is an animation-speed balancing issue and it's very nice to be able to say that 1 second real-time is 1 minute game-time or something similar. That particular example leads to 1m→1hr, ~15m→1d. In *Stardew,* 45 real seconds makes an in-game hour and is a bit quicker, as *Stardew's* days are 18-20 hours long, days last between 13.5-15m. Concerned Ape in particular mentioned wanting the game to move faster, closer to a 1s is 2m ratio.

It's reasonable to stop time when players are indoors, in menus, or in dialogue. It might not be necessary with the inclusion of a time-slowing food effect. This also questions whether or not the flow of time in the mines or forest should be different. Other than maybe doing bizarre things with time in the forest, intuitively my answer is no.

Social events with NPCs should take a fixed amount of time, and if they make plans with you, should adjust their schedule such that they meet you in a certain place to begin the event. NPCs will possibly wait a short time, and an event could typically take 4 in-game hours as an arbitrary estimate. These events will be marked in a calendar/planner, and the player will be reminded of their commitments when they wake up that day. Further, players might be able to set an alarm clock; Early to bed, early to rise.

Dawnflower Labyrinth will test an open-timed festival, where players can necessarily come and go as they please. Certain events like contest judging and event entry will still have specified time periods that players need to be aware of, and most NPCs will spend time at the festival. However, players are largely free to continue their routine non-social activities throughout the day as they want. Players should, however, be given the choice of skipping ahead directly to the next event/stage of the festival, at least after it has already started.

Mines & Labyrinth

TBD

Characters

TBD

Festivals

TBD

Social Systems

TBD

Player Stats

It's worth looking into the way that *Persona 5* handles time and social systems. Players might enter the mine in a way that guarantees a certain amount of time spent, much like any event in *Persona 5* or specifically the metaverse uses half of a day's available time. Whether I restrict players to only enter the mine before a certain time so they can always exit in the evening, or perhaps automatically transport players home if they enter too late and take away their agency. It might be best to just maintain a normal or slightly slower time-flow in the mine and let players suffer a penalty if they stay too long rather than giving them "infinite" time but limited stamina.

Generally players are largely concerned with how much time they spend doing things, and as there isn't much to do indoors, time can reasonably be frozen while inside. Does this apply to the mine? Currently undecided.

The social stats business from earlier might not be terrible either, and I really enjoyed that system. I like the idea of training stamina by working/using stamina within a certain threshold of your maximum. If you exceed this threshold, maybe your body is a little sore and your maximum stamina is reduced the next day? This could affect how fast players run, how long they can run for, how much stamina is used by running or other activities, their maximum stamina and other such things. I don't need to necessarily include the same 5 stats, but adding some minor RPG elements like pretty much every other genre is doing these days might not be a bad thing.

In my opinion, its important to not screw up player placement of crops by having weird-shaped terrain. It bare-minimum messes with my mild perfectionist habits. As a potential solution, as it might bother players to have 1-2 squares extra on the end of one of their crop layouts, the grass border to their field/general farm can spread to adjacent tiles if they're untilled. This kind of thing might at first glance be better suited towards 2D games compared to 3D

ones and while not mechanically important in this situation, it could be an important feature aesthetically.

General Pattern Implementation

Stamina System

Stamina *experience* system, with different benefits at different level *thresholds*. Experience is gained at the end of a day, based on the quantity of stamina used during that day, *tracked via cumulative counter* over the course of that day. Aim to emulate real life in pushing boundaries of physical exertion; gain more experience towards leveling stamina with higher fractions of *general* maximum stamina used that day. For example, 2 experience points for using 50% of your maximum stamina, 3 for using 75% of your maximum stamina, and 4 or 5 experience for using 90% of your maximum stamina.

Players should be presented with a risk of using 100% maximum stamina, through character *collapsing* and awakening the next day with reduced maximum *but not general* stamina. Generally speaking, it should be optimal to reach the maximum experience threshold each day without suffering character collapse, and the reduced stamina the following day should create a sub-optimal environment which provides less experience than reaching 75% each day. This is a quantitative balance problem to be solved.

These thresholds, *50% etc*, are subject to change but are ideally flat numbers for easy player math if the systems are teased out by the game community. As food will be allowed to restore stamina as its primary purpose in the game, these thresholds could be reasonably moved to include numbers greater than 100% and it is likely that they will be adjusted in such a way. I expect players will circumvent the collapse penalty by consistently eating food the following day, although the maximum available stamina that day is capped such that the penalty isn't negated entirely by feasting that morning until the stamina value is fully recovered to its normal value.

Furthermore, as the player levels up stamina through the experience system, and there may be alternate ways to gain experience towards leveling up

the stamina player statistic, players are rewarded by being able to do more in the game. However, as this necessarily makes players take more actions to reach the earlier mentioned stamina thresholds for different daily experience gains, without increasing the amount of experience required to level up the number of player actions required to gain the requisite experience will increase. This needs to be taken into account during experience curve balancing for the stamina statistic. Whether the benefits of higher levels are action cost reduction or an increased maximum stamina value, the resulting effect is similar.

The Eternal Forest (Name Subject to Change)

The eternal woods represent a rogue-like exploration challenge to the players. Similar to the “Lost Woods” from various iterations of the *Legend of Zelda* series of games, the Eternal Forest should have seemingly shifting paths through it that may return players to it’s entrance. While not requiring the stamina consumption and general repetition of progressing through one of the two mines, rarer resources should be available in addition to an area of indeterminate size with objects such as large stumps and rocks that players may break, pick up, or otherwise harvest for essential building and farming materials (as well as a way to further consume stamina to fuel the stamina experience system from the prior implementation section in a way that players might find more fun and engaging than repeatedly tilling the same square of ground).

As the Eternal Forest would imply, time progress could be reduced to an extent to encourage players to come explore the area. As it has a greater degree of exploration than similar-looking mine floors and a lower focus on resource collection, the Eternal Forest is an excellent candidate for hiding secrets or easter eggs depending on in-game date or weather.

In support of player-community content, I think it would be best to generate the forest layout and contents based on (seeded) the in-game date. This allows for community-created guides with content such as “I got X item in the Eternal Forest on Y day in my game, you should go check it out! I went north, west, west, and then south and found it in front of a waterfall!” Eventually, I suppose

this would allow for consolidated information on the complete forest layout for every day of the game but it isn't worth fighting player information gathering over this. Having layouts tied to days allows for non PCG layouts on particular days of my choosing, which I can use for friend/family/supporter birthdays as a means of thanking them. This might also allow for specifically coded weather on certain dates rather than having a PCG state-based/Markov-chain modeled weather.

Daily Weather

A number of considerations need to be made for weather in the game. Avoiding overly complex systems like those that affect crops in a few iterations of the *Bokujou* series of games (Harvest Moon: Sunshine Islands, etc) and their "weather points" is a good idea. While players should be happy to have their crops watered on rainy days as is standard across the genre, it would be unfair towards players to have crops fail to grow from consistently bad or cloudy weather that results in a lower amount of sun. After all, certain crops can only be planted in winter and as such are suited to growing within the range of winter weather patterns, it makes sense that they would grow without issue.

At a minimum, I think we want bright/sunny weather, clear weather, cloudy weather and rainy weather. Rainy weather can become snowy weather at certain temperatures (which might be dependent on altitude if we're getting specific). It would also be consistent with the genre to have thunderstorms in summer or blizzards in winter where players cannot go outside. Certain festivals and specific days I want to tie to the Eternal Forest for easter egg purposes can require particular weather.

I dislike the idea of save-scumming to change weather, although I understand why players do it if they're overly penalized for thunderstorms/hurricanes or blizzards in the appropriate season. Animal care should be minimally impacted by missing a single day of care and attention to not penalize diligent players if it happens.

As is, the weather can be decided as a function of both precipitation and temperature. It's pretty true to real life and makes a good deal of sense. I don't

think having different temperatures based on altitude in game is a good idea, as the difference between rain and snow is a large and would affect players disproportionately based on their farm location. In addressing the issue of save manipulation or save-scumming which isn't fun at all as it largely feels like a waste of time, although user research might be done on this.

Preliminary research suggests that users save scum for particular rare resources when RNG plays a large part in whether they get them or not. Additionally, users will abuse saves when the difference between the desired RNG and other options is very large (when they need one result exclusively). Newer games in the *Bokujou* series have weather determined more than a week in advance which largely negates save abuse for weather manipulation. It is worth noting that while there are many complaints about the heavy amount of RNG in certain *Bokujou* titles that drive players to abuse saves, many players came to like rainy weather. This was the case when rainy weather caused the equivalent of watering crops twice rather than once as it had previously in the series. With a watering system that allows for watering twice, rainy weather should always create the best result. Because of the large number of reasons and complications surrounding save abuse in this genre of game, I will be adding a section to specifically discuss the design patterns that will discourage it.

As it is, weather will be a result of seasonal effects on temperature and precipitation. Ideally, there will be an average temperature at the start of the month and a target average temperature at the end of a month, resulting in a stable average that gradually changes over the course of the month. It gets warmer throughout spring, warmer throughout summer, colder throughout fall, and colder throughout winter for example. Some amount of noise will be applied to this slope such that the temperature remains *interesting* while still fitting the linear change in temperature of the month. The highest rates of precipitation happen in winter, followed by spring and fall equally. Weather should be generated based on previous day's weather in that rain should usually be preceded by cloudy weather. Because we would like to give players ample

warning of very bad weather, which shouldn't last longer than a day and should be preceded by rain or snow depending, a weather-related broadcast via radio or television should tell players that a large storm is approaching.

This means weather patterns need to be determined several days in advance such that we can have cloudy, then rainy, then thunderstormy weather where the player receives notice when it is cloudy or the day before. This requires recognizing certain patterns such that the information can be given to the player. Furthermore, as weather can sometimes change throughout the day, it wouldn't be unreasonable for a day to have 3 distinct periods of weather, where spring might have winter might have low medium low, summer might have medium high medium, spring might have medium, medium low, and fall might have low, medium, medium. This allows for it to get cold enough to start snowing in the evening where it has been raining all day, or for clouds to clear up or form at different parts of the day.

Certain things should be taken account when changing the weather throughout the course of a single day. This necessarily complicates weather forecasts, and I don't think that weather should be able to go from clear to cloudy to rainy in a single day. Players may complete their tasks for the day early and then go to sleep, leaving their animals out only to find that it rained that night and they became sick. Because of this, the weather forecast should be 100% accurate that it either will or will not rain the following day (at all), and should mention the current weather and the remaining weather pattern for the day: "It's currently slightly overcast and we're expecting rain later this evening" or something of the sort.

Having more rapidly changing weather requires three times the size of the data structure used to hold the upcoming weather patterns. For a single period of weather, we need a temperature difference from the average. As the average is determined based on the month as a whole, we need at least 4 bits for a signed positive or negative increase in temperature up to 9 degrees, which seems a reasonable difference in temperature. This would be 12 bits overall for a 3 period day. Then we would need the weather state of sunny, clear, cloudy,

precipitation or storm. This requires 3 bits with 5 states, or 2 bits if sunny is clear weather dependent on a certain temperature above the average for the month, where we then have clear, cloudy, precipitation, and stormy. That being the case, we have an additional 6 bits in a 3 period day for a total of 18 bits for the entire day's weather. This results in 324 bits for an entire month's weather pattern at a *minimum*. However, due to how a byte is the smallest reasonable data type, we're dealing with a minimum of a byte per weather pattern (3x per day) with custom modulus or division functions to read the first section of the byte compared to the last section of it. Reasonably we'd just use 6 bytes per day and worry less about needing to read individual bits, and we'd have either sbytes or bytes as data types in C#.

The question also remains as to when new weather patterns will be generated. I feel like the beginning of the previous season is a good place, as that threshold is when an old season's pattern becomes completely unused and can be reclaimed for use as the next season's pattern. How long the generation algorithm takes is another big one. You could reasonably generate it all during a scene transition that represents the change between seasons, or at least try to. If we're dealing with stored bytes, 3 bytes per day results in 90 bytes per season or 180 bytes for 2 seasons. This is a tiny amount of data, so we could reasonably increase weather data for two months to 360 bytes with data for temperature and weather-type taking up a full byte each.

Save/Load System

For ease of use and development, the initial user interface for saving and loading games uses simple Unreal Engine widgets (boxes with bound text widget children) that display Player Name, Farm Name, Playtime, the current Season, Day, and Year, as well as possibly in-game money for the save.

Because of UE4's workflow and because we want to display relevant save-identifying information without loading several potentially large saves in their entirety, we'll be using a meta save file (slot). Its save information, including the default slot number and name information most saves should have, is only an array of structs. These structs are simplified save files, containing the information displayed in the widgets as mentioned above, as well as the appropriate slot number/user index and slot name to link things up appropriately to be loaded and modified.

This design choice isn't without consequence. We pay some performance cost in needing to modify and save the meta save file whenever we modify and save any of the actual save slots. Additionally, as our UI and save data structures are arranged, players are limited in how many different save slots they may use. If you don't need to give players an infinite number of saves, slot information can be hard-coded and we don't need to worry about saving conflicts that might arise from a poor slot-naming system. I think the ideal situation is containing a number of save slots equal to the total number of eligible player marriage candidates (probably 10-12).

The current menu design doesn't support that many slots displayed at once, although a *scrollbox* could be used. The ideal, finished and polished menu design is completely different and supports many different save slots. Keeping to some early rough narrative concepts and the original working title *Dawnflower Chronicles*, the majority of the menus of the game, outside of dialogue, shop and possibly inventory could be done through a rendered book.

Difficult and unfamiliar as this is, preliminary research has shown it to be very possible. A book can be modeled in 3D and pages can be turned as an animation that can be looped some amount of times. This allows for a very

emergent and organic collection of save information to be displayed. As we're only showing one save at a time now, and using a larger portion of the screen to do so, we can include images (of the farm, possibly progression related diagrams), and text about what the player has done and accomplished in addition to the other information. This can absolutely be woven into fictional(ish) sentences using an alphabet (and possibly language) created for the project. This particular implementation goes above and beyond what is required for this variety of game but would possibly be one of the most impressive user interfaces seen in recent games.

Title Thoughts

This section will also discuss logo design, for which I currently like the font Ayres. The genre traditionally has been comprised of two word titles. Story of Seasons, Harvest Moon, Stardew Valley, Orange Season, Bokujou Monogatari. Keeping with convention, I've chosen Dawnflower Chronicles. For this choice, I feel that *Dawnflower* is essential.

The words used for titles in this genre all evoke farming or plants, many of the English-localized logos are in large, friendly block lettering. *Stardew Valley* is currently the exception to the rule as it avoids a wooden backdrop and maintains it's pixel-art-esque aesthetic. Meanwhile, the traditional Japanese logos have primarily had large, friendly, lettering with a brown wooden texture and a white outline. Subtitles for these games usually include a geographic word such as town, island, or valley.

After some amount of brainstorming, I've chosen *Dawnflower Chronicles*. "Chronicles" is too frequently used as a cool-sounding synonym for "story," though in this case I hope its use is acceptable. "Dawnflower" refers specifically to the morning glory flower, and maintains the game's title's ties to nature and agriculture. In Japan, the country the genre was created in, the morning glory has cultural significance as a symbol of summer.

Originally I chose *Dawnflower Labyrinth*, referring to the labyrinth of ancient trees I plan to implement in the game. However, in the future this restricts subsequent titles in the series to include that particular game feature, and complicates world-building if the geographic area shares the name as well.

Logo design isn't finalized but is in production. Additionally, with any plans of localization in Japan as a tribute to the *Bokujou* series, as lofty a dream as it is, the title will probably drop *Dawnflower* in favor of *Asagao* or precisely morning glory. It's presumptuous to say so, but I hope a translation to *Asagao no Nendaiki* or something similar could have fans affectionately shortening the title to the likes of *Asa-nen*.

Save Abuse and Player Motivation

Discussion of save abuse will be more anecdotal and less concrete as it deals directly with the subjective nature of player expectations and motivation. The topic is worthy of its own unique document. With simple user polling for a recent *Bokujou* game (*Story of Seasons*), 89% of the respondents said they abused saves. 3% of total respondents said they only reloaded saves in order to win public festivals. 22% reloaded saves for favorable resources from mining. 30% don't aim to do it or avoid it, seemingly having a neutral stance but not denying that they reload saves in this way. The largest percentage of respondents (34%) said they abused saves "for many/multiple aspects." This poll had a total of 128 respondents from a forum related to this genre of game exclusively.

While I won't go into detail here, rather saving it for a separate document, this player activity is rampant within the series. Players have particular motivations for reloading a save. It's a single player game and the reloading of a save doesn't affect the experience of any other person; it should be allowed. Rather than preventing players from reloading saves in this way, I should take efforts to reduce or eliminate certain motivations players have for abusing saves, because I at least don't view reloading a save and replaying an event multiple times until the desired result is achieved to be a compelling or interesting use of time or fun gameplay. In the context of this genre, this means reducing or eliminating randomness in festival results (infrequent events), and providing players the means of gently manipulating numbers for more frequent events like the result of breaking a rock. The undesirable results are not the result of player action, they are not "mistakes" but consequences of a game system players cannot control. As such, I am supportive of reloading when players make a mistake like leaving a cow outside in the rain to get sick, but not when a player loses a festival. Players should lose an event when they aren't good enough, not when they're good enough to win 20% of the time and got unlucky.

Document Changelog

v2018.01.06a – Document Creation

Added Table of Contents, Overall Goals, Current Design Standards, Pattern Criticisms, Specific Plans, & Stamina System (General Pattern Implementation)

v2018.01.08a – General Pattern Implementation and Title

Added The Eternal Forest & Daily Weather (to General Pattern Implementation), Title Thoughts (Settled on Dawnflower Labyrinth)

v2018.01.09a – Changelog and Refinement

Added the document changelog, trimmed sections down to remove “discussion,” changed game name to *Dawnflower Chronicles* and adjusted the Title Thoughts section to reflect it. General pattern implementation section largely still in need of work.

v2018.01.10a – Save Abuse and UX

Added section on save abuse and player motivations. Created a separate document to include user research regarding player motivations for reloading saves.

v2018.01.13a – Design Pillars

Added the beginnings of design pillar section to have a more concrete foundation of targeted player experiences to build towards. Did not update table of contents to reflect this. Modified plans to not have different farm locations.

v2018.01.14 – Pillars, Hook and Loop

“Finished” section on pillars, added hook and core game loop section. Need to adjust pattern implementation section.

v2018.01.15 – Hook Refinement and Loop Completion

Slightly adjusted table of contents, finished writing the hook and documenting via diagram the core game loop(s). Oddly, the loops reflected the four pillars.

v2018.01.19 – Big Changes

Completely changed “specific plans” to “general plans & features” with around a dozen subcategories. Adjusted table of contents to reflect some of the changes. Finished writing a survey targeting Stardew Valley community as they are a large portion of this genre’s fans now.

v2018.01.25 – Planting & Layout & Title

Completed an early version of the game’s title/logo and included it on the first page. Filled the subsection of “general plans & features” called “planting & layout.” Wrote “Farm Upgrades” and began to write “Cooking” sections.

v2018.01.27 – Cooking & Time Flow

Finished cooking section and also finished time flow section. Specific additions include description of planned/committed NPC social events and non-forced festival activities to be elaborated on in later sections.

v2018.04.09 – Save System Implementation & UI

Added a section under general design/pattern implementation for the save system and concept of a 3D-modelled book to be animated for most of the game’s menus.