

Unit-4 | Final Major Project



SolarGram's Fix It Shop

Anukriti Gupta [22018366]

with

Skylar Law, Katharina Trappe and Vladimir Chandran

Design Framework:

Construct the game's narrative with a strong emphasis on feedback. The story should react to the player's choices and actions, providing meaningful consequences and opportunities for player agency. Feedback mechanisms should be interwoven with the narrative to maintain engagement. Some techniques that can be applied follow:

1. **Continuous Player Feedback:** Implement a system that continually offers feedback to players regarding their decisions. This feedback should be delivered in real-time, allowing players to understand the impact of their choices as they progress through the game.
2. **Narrative-Driven NPCs Relationships:** Develop a supporting cast that acts as a feedback system within and in line with the narrative. This performs as the interface which informs the player about their narrative branch and the amount of agency, they have in influencing certain decisions, coupled with adding emotional expression and morality it can be a powerful tool in curbing player unpredictability.
3. **Structured Personalities with Feedback Variability:** Create companions with distinct, structured personalities that react dynamically to player choices. The feedback delivered by companions should have variability to keep players engaged and curious about the consequences of their actions.
4. **Narrative Quests Aligned with Feedback:** Design narrative quests that are closely aligned with the feedback system. These quests should both contribute to character development and provide players with insights into the consequences of their choices.
5. **Game-Defined Optimal Outcomes:** Clearly define optimal outcomes within the narrative, guiding player actions towards these goals. Ensure that feedback and narrative cues support the achievement of these outcomes.

Design Framework:

6. **Interactivity Progression:** Develop a structured progression of interactivity within the game. Start with scripted sequences to introduce players to the narrative, and then gradually increase interactivity as the story unfolds, allowing players to shape the outcome.
7. **Dynamic Feedforward Integration:** Integrate feedforward or affordances within narrative elements to help players anticipate the consequences of their choices. These feedforward cues should work cohesively with the feedback system to maintain a consistent player experience.
8. **Scoping Player Interaction:** Define the boundaries and scope of player interaction. While offering choices is crucial, set clear limits to maintain the narrative's coherence. Ensure that feedback reflects the scope of interaction and the consequences of choices made within these boundaries.
9. **Narrative-Driven Affordances:** Design affordances that align with the narrative structure. The divergence from the optimal path should match the narrative planning while flowing intuitively with the rest of the gameplay.
10. **Balancing Interactivity and Narrativity:** Continually balance interactivity and narrativity throughout the game, adapting feedback and feedforward to ensure players are engaged with the story while making meaningful choices.

This framework highlights the importance of feedback in narrative design and how it can be used to guide and scope player interaction. By incorporating feedback at every stage and ensuring it aligns with the narrative structure, game designers can strike a balance between player agency and coherent storytelling while keeping players engaged and informed about the consequences of their choices.

Personal Aims:

- Create a 3D Game.

Prior to this unit, I have yet to work in 3D Unity. Figuring out the art pipeline, how it can be optimised and how it can be used effectively is my primary goal with this aim.

- Use Inky to create a Branching Narrative.

Inky has been used for a lot of very successful narrative games (80 days, Where Water Tastes Like Wine), even though both games are mechanically light, their narrative interaction (not just the writing itself but the input taken by the system) has carried the games to critical acclaim.

- Create a believable and alive Solarpunk world.

The Solarpunk genre often lacks a realistic edge because a core part of solarpunk is optimism and any realistic adage would dull the shine and turn the narrative cynical. I want to balance a bit of realism with solarpunk and deliver a world that can be realistically imagined in 20-30 years time.

- Examine application of Design Framework on a very small scale project.

My design framework was based on research on very big budget AAA and even AA titles, I want to see how feasible it is to maintain my proposed philosophy of player agency reflecting on 4 levels, Visual, Ludic, Gameplay and Gameworld.

- Work with a team.

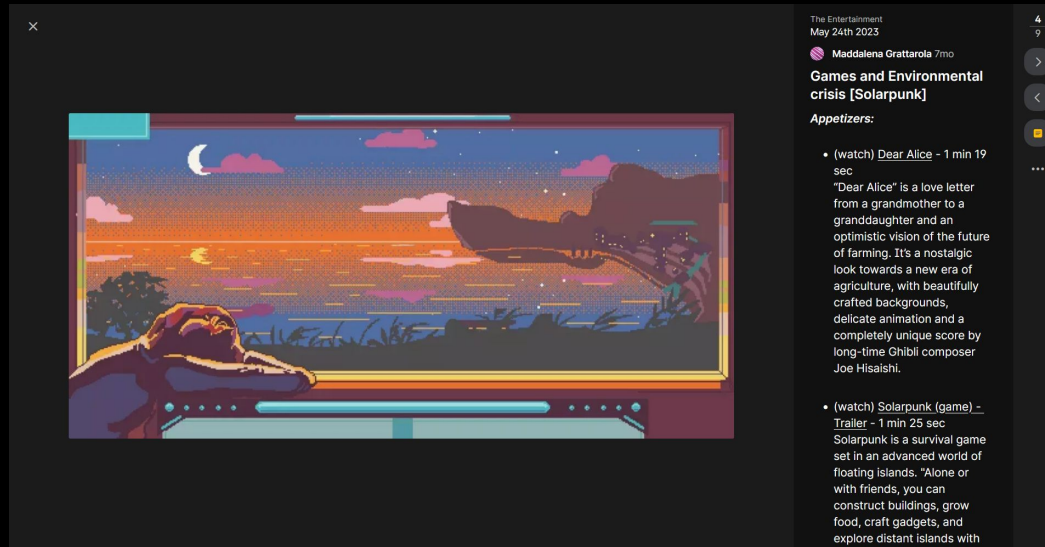
Making games on your own is hard, I want to see if the end result benefits from having multiple people involved in the Game Design process.

Tools Used:

- Unity, Game Engine.
- Inky, Branching Narrative Scripting
- Nomad Sculpt, 3D sculpting and Meshing.
- Procreate, Concept Art and UV Map colouring.
- ChatGPT, Code Support and Concept work.
- Miro, Narrative Planning and Collaborative Design Decisions.
- Discord, Communication with Team.
- Trello, Communication with Team.

Pre-Production:

The inspiration for this game came from the May edition of The Entertainment, our optional book club about video games and culture,



The PC Gamer article, “*Enough cyberpunk—it's solarpunk's time to shine*”, particularly highlights the lack of alternative point of views when it comes to the future of our planet and why the dystopian cyberpunk universe is more regularly adapted into Video Games. There is a notion that the inherent anti-capitalist principles of a solarpunk universe does not lend itself well to the typical, individual reward led, AAA Video Game development.

It is easier to motivate a player towards a narrative end goal when the focus is on overcoming trials and they are positioned as an outcast on the fringes of society rather than when the narrative end goal involves building community and getting the player to care about the game world, in this sense building community is a much more abstract and hard to represent goal and requires an emotional component to get the player behind the “why” of their in game actions.

The image of a solarpunk grandma was thus born, a helpful community figurehead who would fix the bumps and dents in your possessions for the small price of teaching her how to use her phone. Helping grandma would serve as the motivator and her repairs on your possessions would serve as the reward for completing your story goal. Whilst the idea is really simple and produced with minimal forethought, it was a positive indication that solarpunk goals can be adapted to fit the typical model of game development.

Over summer break, we each created a document outlining the things we wanted for our game. My initial ideation was as follows:

1. **Core Themes to explore:** Climate optimism, 4 Rs (Reuse, Renew, Recycle and Repair), Decolonial, Community Focused, Anti-Capitalist, Where technology fits in a society like this
2. **One-Line Description:** You are an engineer and you run a small repair company in post-climate disaster tech, but really you do everything from repairs to making grandma’s tea. Help your community and watch the town around you change.
3. **3 Cs:**
 - a. **Camera:** Isometric Top Down with free camera rotation, because that perspective gives the environment major focus rather than the player and would be good for environmental storytelling.
 - b. **Controls:** Mouse and Keyboard, Mouse to move the camera and left click to interact, WASD to move. Can be translated into controller.
 - c. **Character:** customisable Character, narratively a community helper but not a leader, a mirror for every person in real life that can help make small changes for the good of the community.

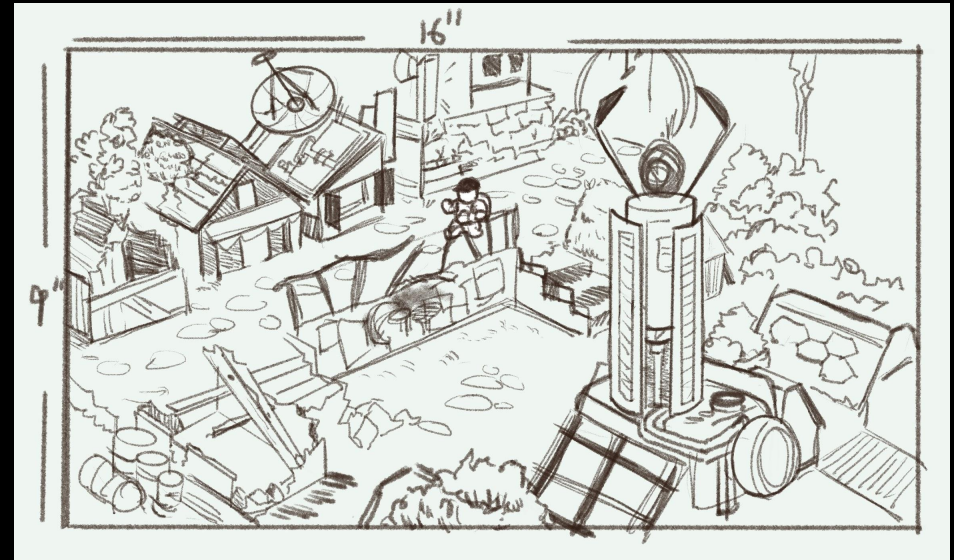
Core interaction:

1. Resource management + Strategy Based, All about reusing parts from random parts of the town to fix stuff, breaking down old machines to fix new ones, all about balance.
2. Basically, each machine can be a procedurally generated jigsaw puzzle with multiple solutions.
3. The number of machine types to be fixed can be determined.
4. You have a home base which you can run around and get familiar with, as you fix things around it the map reacts to those changes and grows.
5. Quest givers will be people in need of their machines getting fixed + other odd jobs.
6. As you help out around town, you will meet community leaders and potentially aid them in achieving their goals
 - a. Example, You can talk to the mayor and choose to provide information on people around the town who are hoarding resources or you can sympathise with them and not tell the mayor (to a detriment to your own operation and environment.

Ideas:

1. Applying the hero's journey to the environment around the player
2. Departure: The world goes through a massive change, some discovery, some disaster, an inciting incident changes the world from "normal" to "supernatural"
3. A world where climate disaster has already happened and humans are rebuilding. The whole message is about the power of group action
4. You are an engineer, you run a small repair company
 - a. Game mechanics can literally mirror reuse, repair, recycle
 - b. Choices come via dialogue
5. Or you are tech support in a solarpunk universe, you run around teaching old people how to use this new-world tech

Between 4 initial documents, we all agreed on the main game mechanic being fixing machines and running a repair shop. The initial ideation imagined the game as an open world, 2.5D game where the player could move from their shop into the town to do repairs, the repairs would work as a sandbox-ish mechanic where parts could be swapped out for unconventional materials. All of this was done to fully embody the 4 Rs (Reuse, Recycle, Renew & Repair) within the game's mechanics and narrative.



Initial Concept sketches, Open World Design

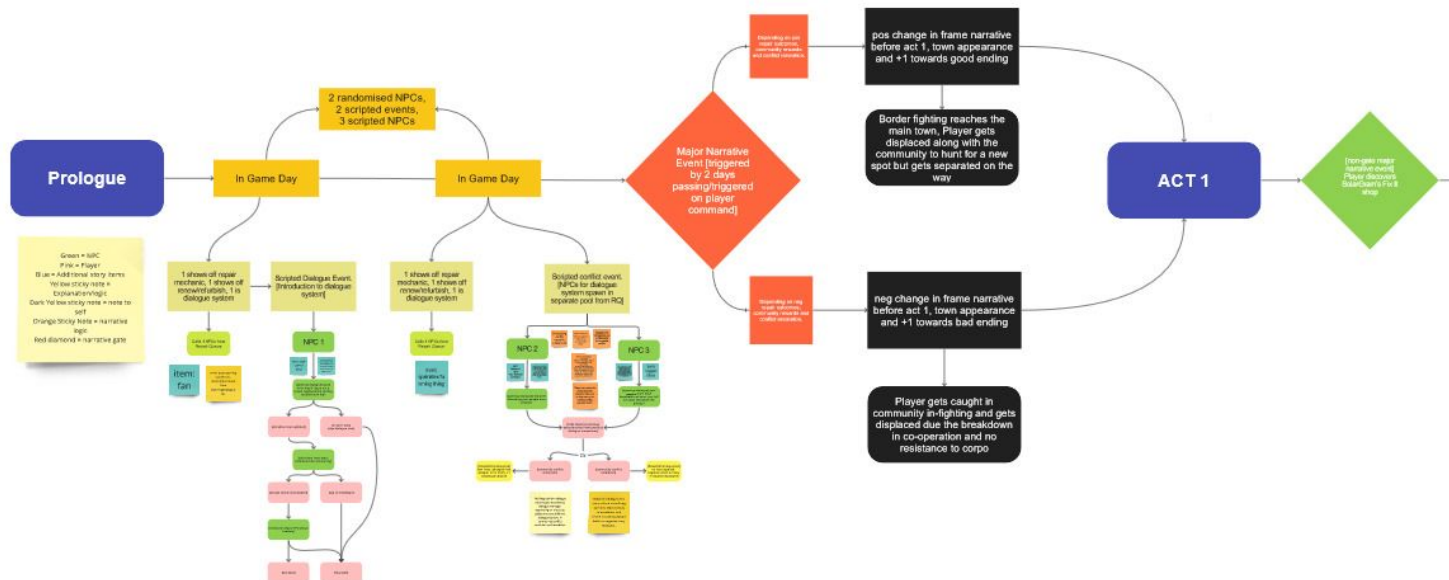
Further Ideation:

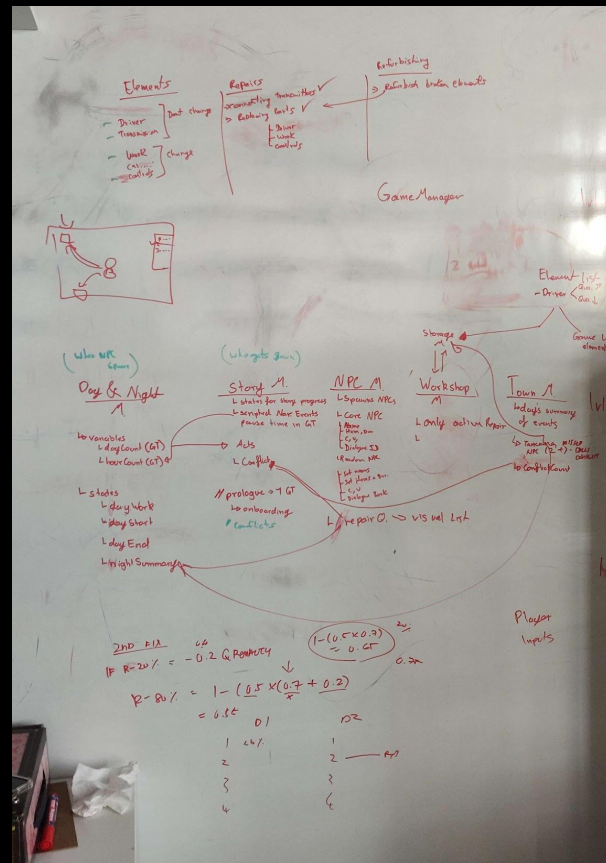
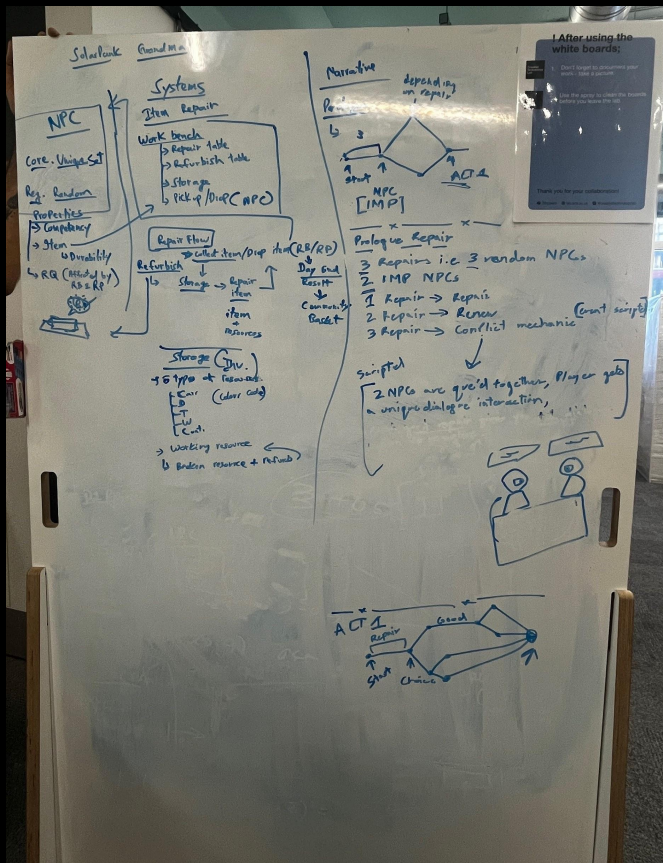
- Each “R” out of the 4rs represented via an unique mini game
- Inventory management system. Special backpack.
- Scavenge and find materials, blueprints and narrative vistas.
- Collect gossip as you fix machines.
- Day cycle, talk to your grandma at the end of the day.
- Give grandma the gossip or don’t.
- Grandma acts like a end of day report + the plants around her can linked to a parametric system which change according to gossip choices.
- Conflict: instead of human vs human conflict, you have human vs environment conflict

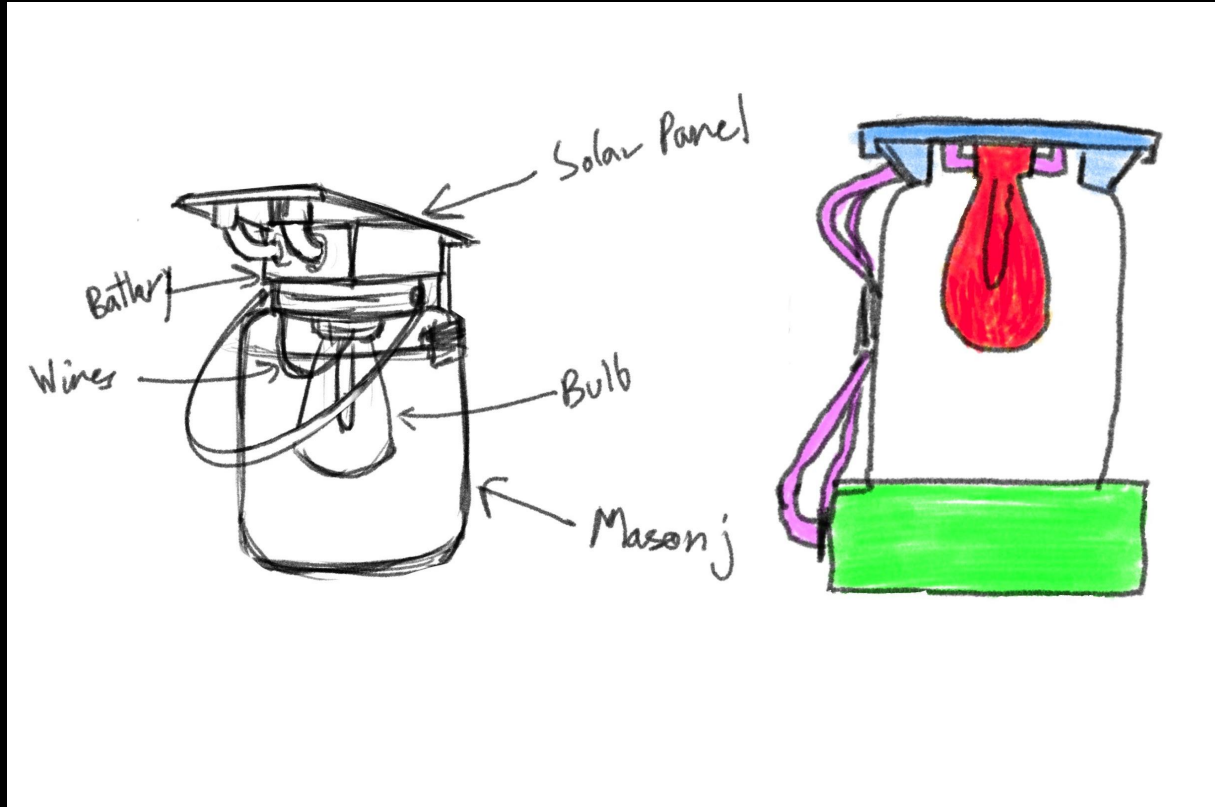
Proposed Game Loop:

- Player wakes up, checks laptop for jobs and then gets to work
- Has a travelling workbench and can fix everything on the spot
- Goes to people’s places on the small town map to fix their things
- As you go between job to job you can run around picking up stuff and gathering materials
- Mechanic UI is first person, as fixing happens conversation bubbles fly by you can choose to capture that information by pressing a button or let them go
- Machines to be fixed can be individual bespoke puzzles i.e each can be an individual level. If we can finalise the amount of levels we want from the start we can control the length of the game and the scope of what we are submitting.
- Grandma is a influential person and her words travels all over so the gossip you tell her will make its way back to you later down the game
 - The gossip is a mix of mundane, political, anti-social and event based.
 - You can bring up each topic with grandma, have some further discussions on it. Here is where the dialogue system records the players stance on these things and they can push the game into a more community focused environment or a individual focused environment.
 - System will record the kind of choices you make as parameters that cause changes on the plants in Grandma’s room.

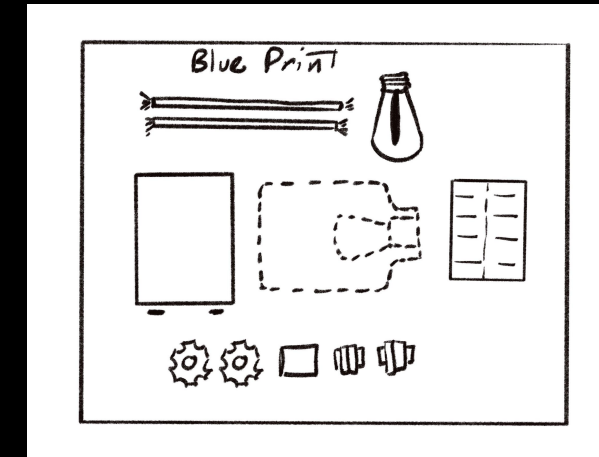
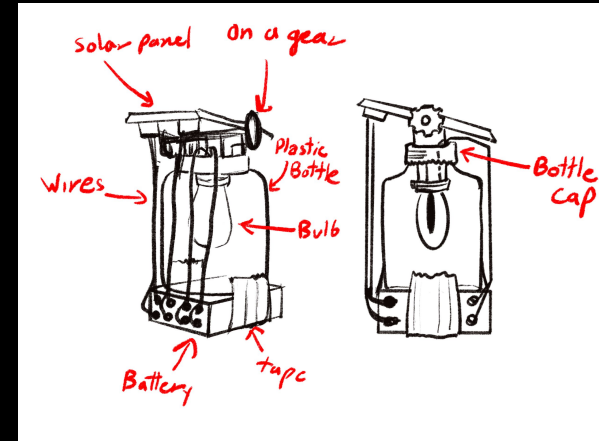
Once the proposed Gameloop was established, we all could separate our parts of the game and scope them accordingly. As narrative designer, my first task was establishing a narrative structure and how progression takes place within the game. After discussing the gameloop with our Course Tutors, the proposed gossip system which would record player choices and manifest as visual change through a shift in flowers/plants around Grandma seemed too disconnected from the main mechanic of the game and was not within the scope of what our development time frame allowed, so I scaled it back and decided to implement a standard dialogue system in place of it. Player choices would still be recorded but in a much more subtle and controlled way, below you can see how dialogue is distributed within one In-Game Day and planning of how narrative gate events will be approached once a gameloop is built and completed.







Solar Lamp, Repair Item Concept Art



Final Proposed Gameloop:

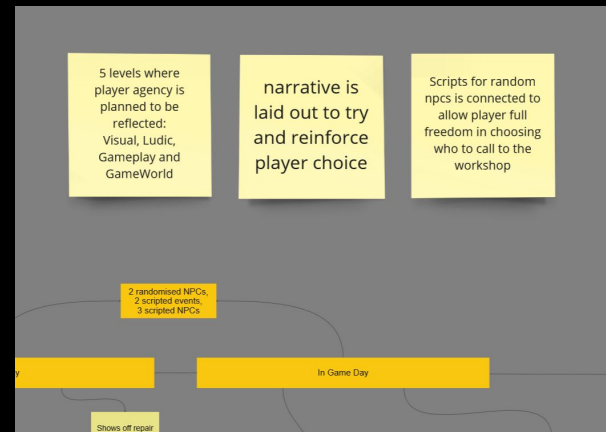
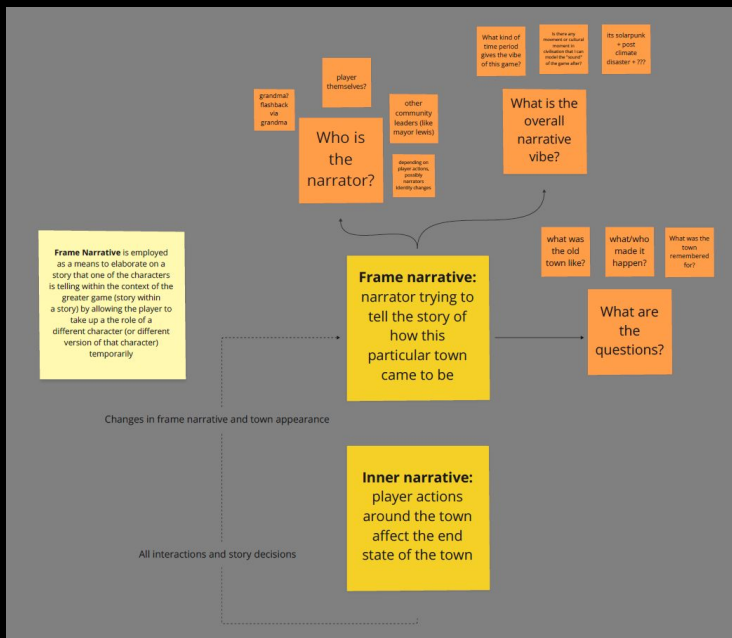
1. Player wakes up, checks computer for repair requests.
2. Player is confined to workshop and NPCs come to them.
3. Repair requests get sorted according to urgency of repair.
4. All parts required for repairs are either salvaged from prior repairs or received as community rewards.
5. Machines to be fixed can be individual bespoke puzzles i.e each can be an individual level.
6. NPCs called by the repair requests perform simple barks
7. Narrative NPCs are forced interactions at the end of each day.
 - a. The dialogue received is in reference to the upcoming major narrative gate.
 - b. The system will record the kind of reaction the player has to narrative NPCs.

Design Decisions for Changes in Gameloop:

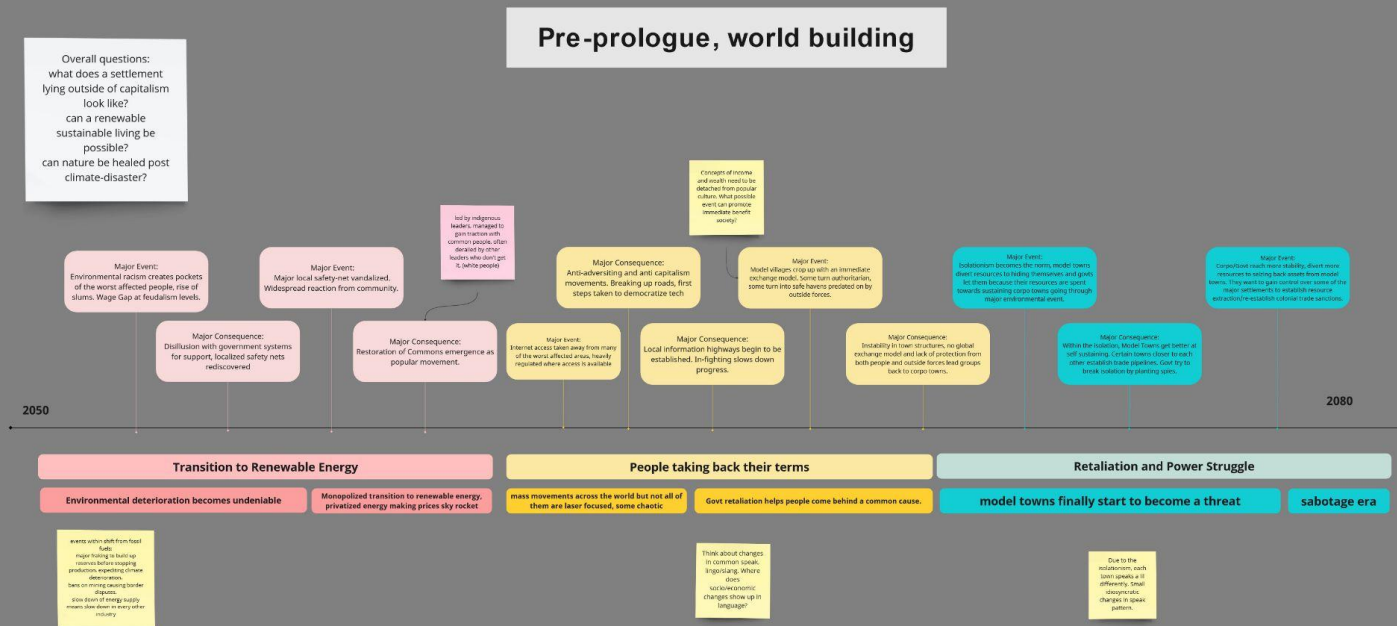
1. Player is confined to workshop instead of open world to scope game properly and execute multiple complex systems on a smaller scale before expanding.
2. As a side effect, scavenging parts is no longer possible from around the world so parts are given to the player/player is expected to sustainable in their repair practices.
3. Gossip system changed out for a simple dialogue system to add more narrative structure to the game, with a more solid narrative game events can be controlled and scripted where needed.

Narrative Planning:

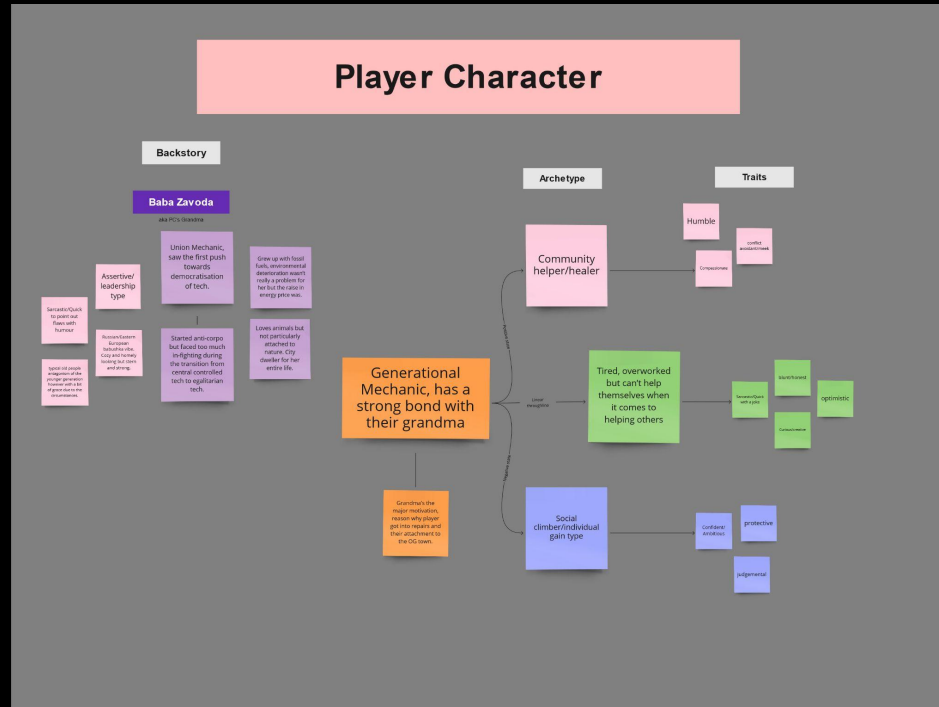
During the development of this game, the narrative has gone through multiple iterations and reordering but the goals have remained the same, to deliver a solarpunk experience without bogging down the gameplay with too many words. Applying all parts of my design framework was out of the scope of this project but I wanted to keep feedback at the top of my mind whilst designing the narrative which meant I had to find a way to reflect changes triggered by player dialogue choices on five levels within the scope o, the most efficient way to do that would be to create a frame narrative to encapsulate the player's in game action.

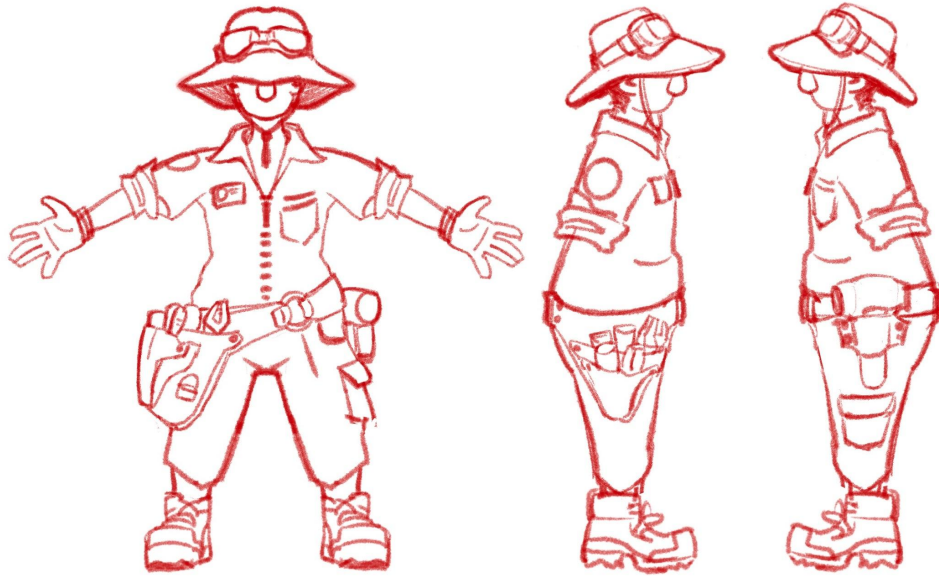


The idea of the frame narrative is to introduce the player first to a shiny, optimistic bastion of a solarpunk city and then through the game reveal how it came to be. In order to deliver a believable solarpunk world, I knew I had to order out the events that took place before the prologue happens and then slowly reveal these old world events through NPC interaction.

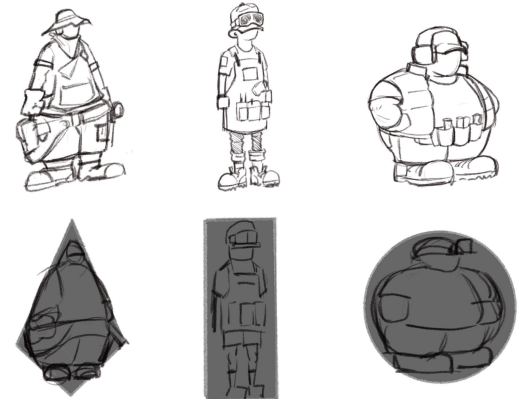


To further loop the player into the world, the PC's grandmother is narratively made the motivation of why the player carries out their in game actions. Baba Zavoda is inspired by the strong russian Babushka archetype and is meant to be connected to pre-prologue events in order to give the player a more personal connection to not only the in game world but also it's history. Baba calls the player Pomo, short for pomocnik which means Assistant in Slovenian. The PC is purposefully designed to look as gender neutral as possible and is never referred in third person, this is to protect player immersion and allow them to project themselves on to the PC without the narrative causing them dissonance.





Pomo, Player Character Concept Art





Player Character 3D Model, Created in Nomad Sculpt



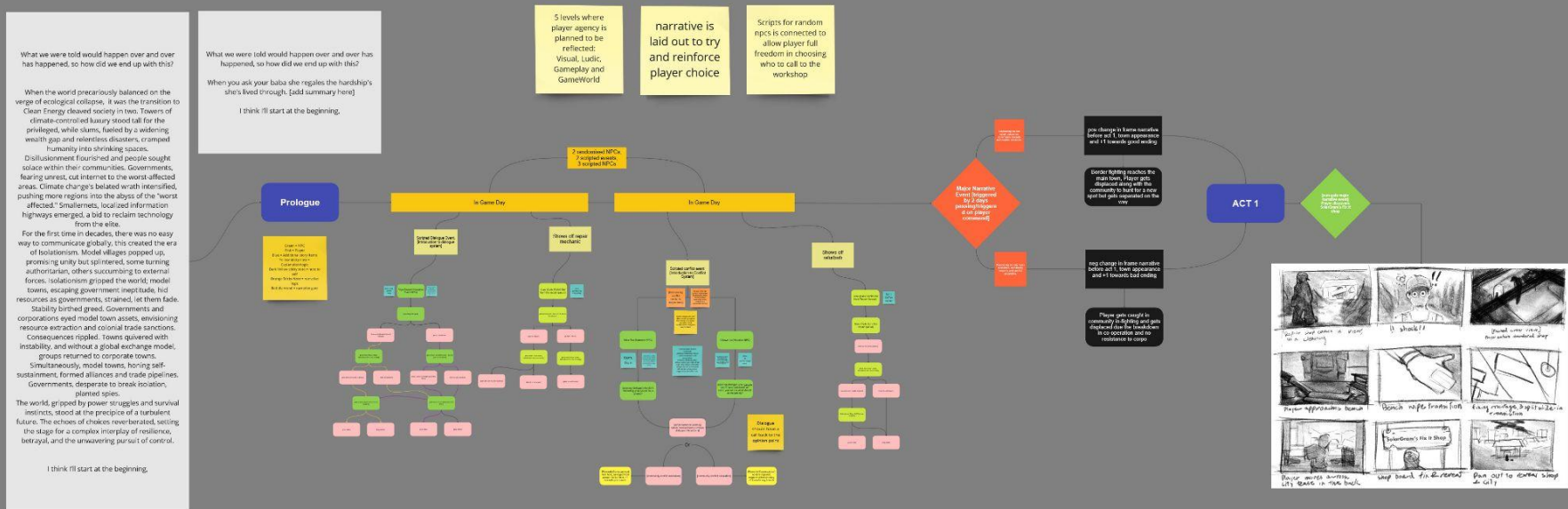


Baba Zavoda Character Concept and In -Game Icon

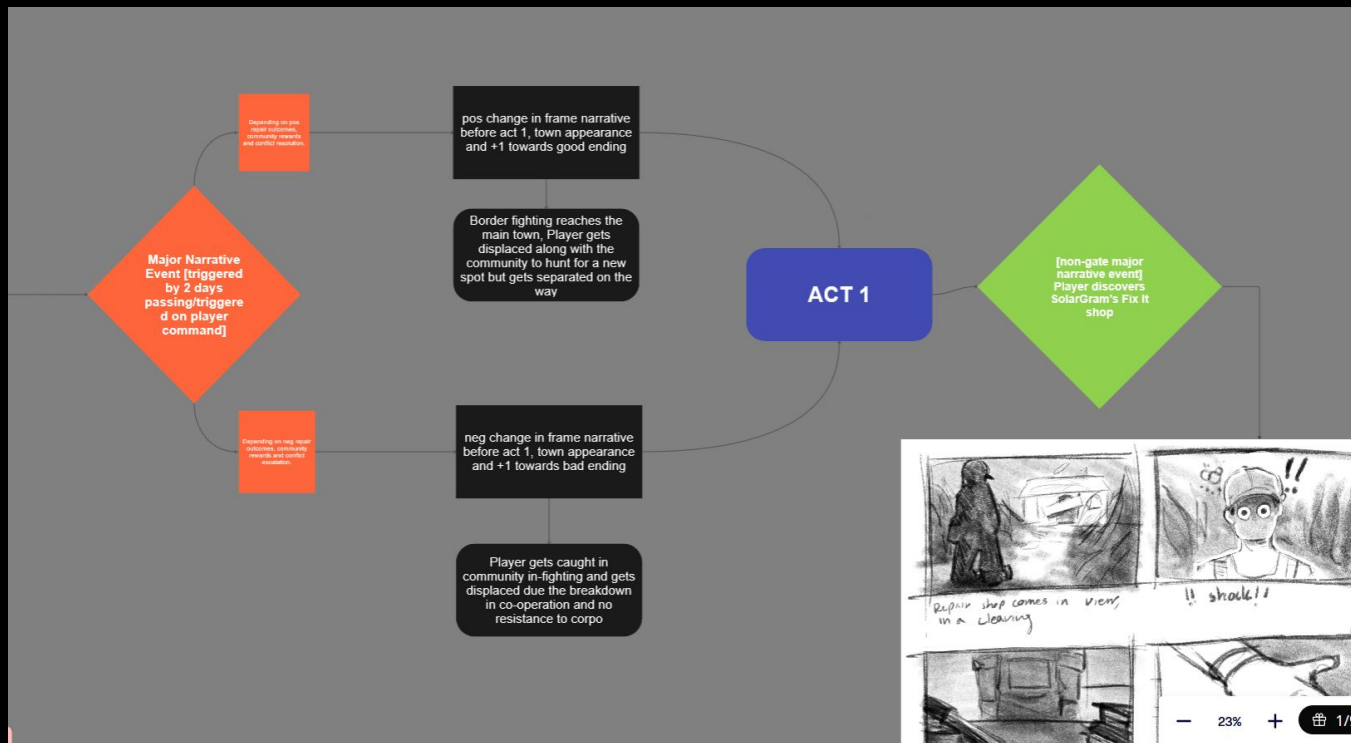
Narrative Planning:

Unit-4 | Final Major Project

The narrative is planned in such a way that player actions and player choices affect their ability to fix community conflicts, If the player is kind, compassionate and understanding with the NPCs that come to their shop (positive branch of narrative) they can promote conflict resolution between the final two narrative NPCs which in turn gives them the positive branch ending to the prologue.

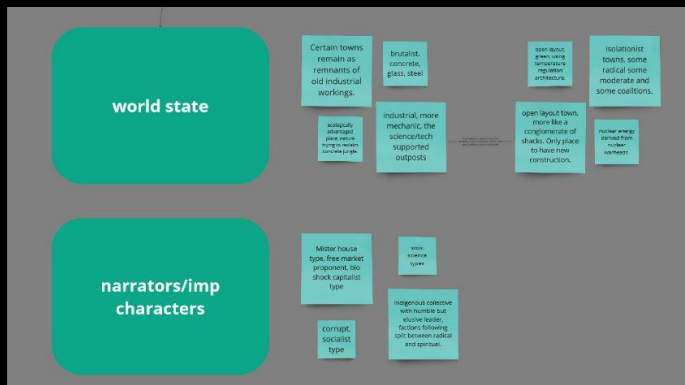
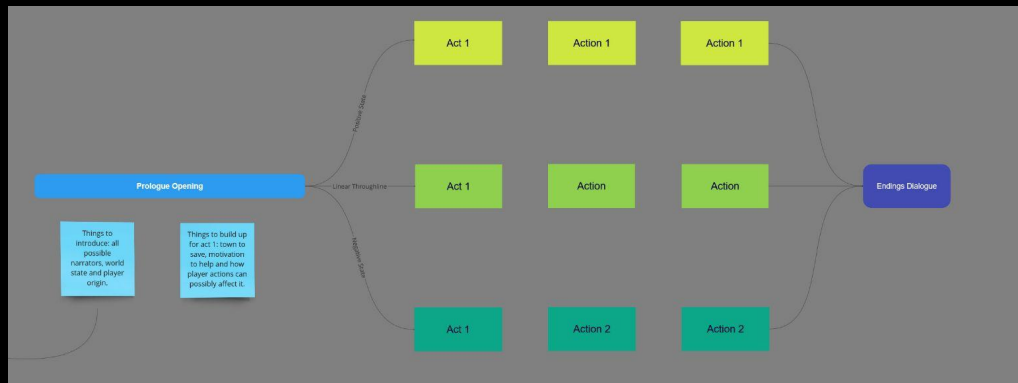


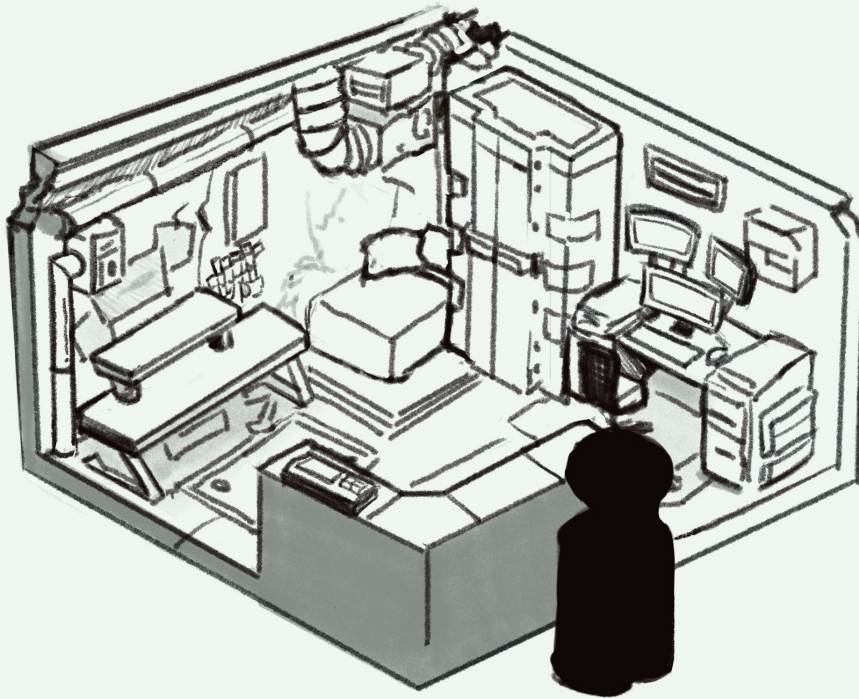
Though the positive or negative branch don't make much of a difference within the prologue, the player will always get displaced from their Corporate workshop and end up finding SolarGram's Fix It Shop, it does create a change within the frame narrative. The way the PC is remembered, as someone who honoured their Grandmother's legacy or one who completely destroyed it, is reflected in how the PC is talked about in the voiceover/narration between acts.



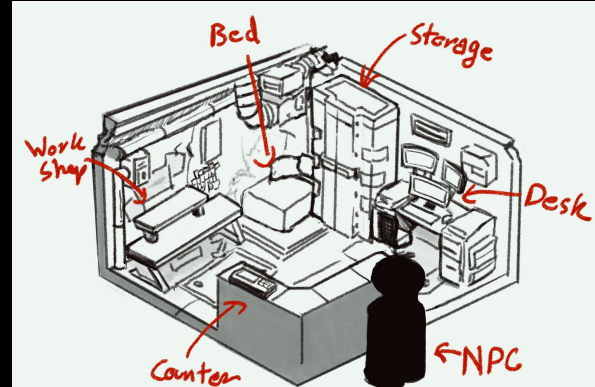
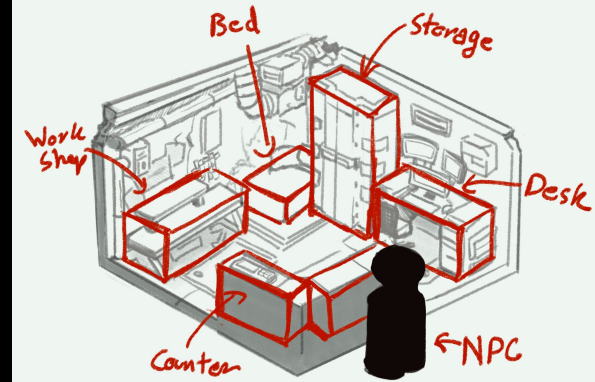
Unit-4 | Final Major Project

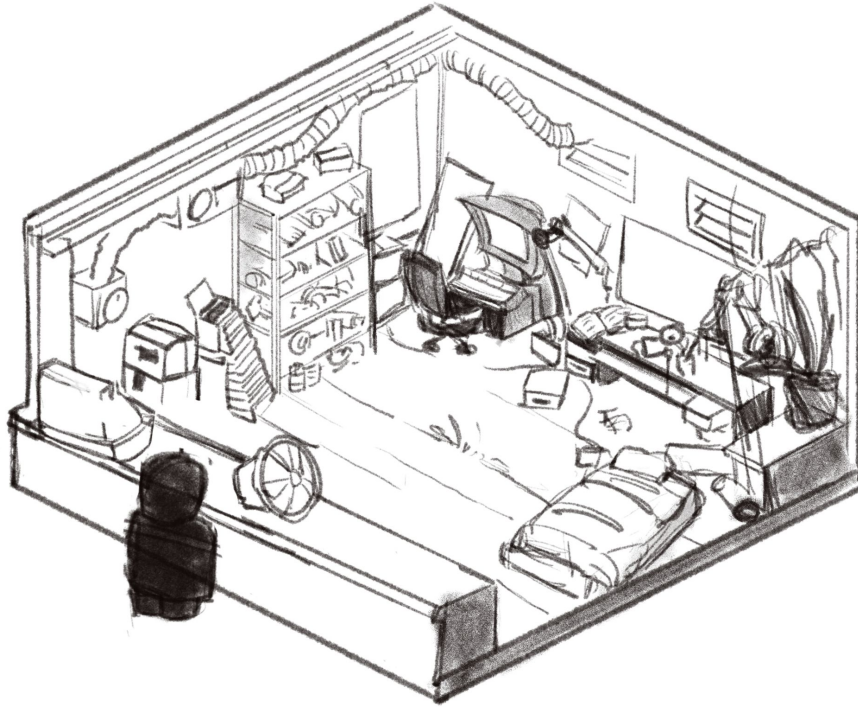
Each act is meant to answer a question about the town at the start of the game, Act 1: what was the old town like? Act 2: what/who made it happen? And Act 3: What was the town remembered for?, and the player's in game actions affect the the answer to each of these questions. My vision for the prologue was to see the player start in a fast paced, corporate model town to provide contrast to the quiet, rural temperature of when they find SolarGram's Fix It Shop.



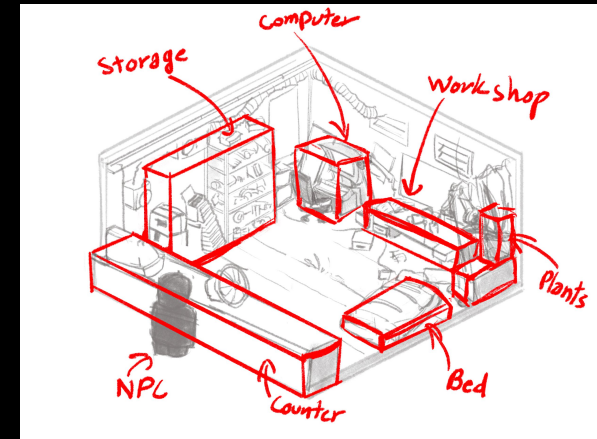
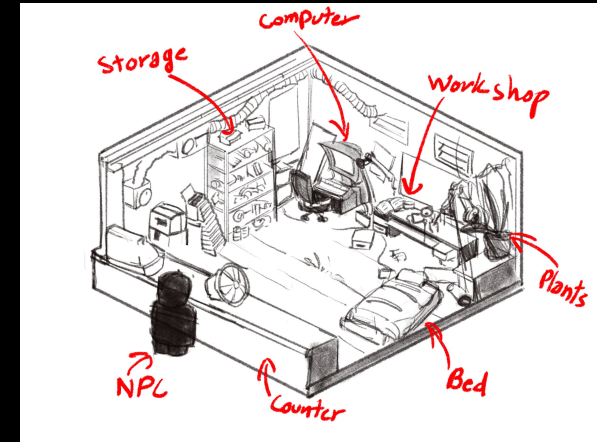


Concepts for workshop one, Corpo level





Concepts for workshop two, SolarGram's Fix It Shop





Proof of Concept, End of Prologue Cutscene, Player Discovers SolarGram's Fix It Shop

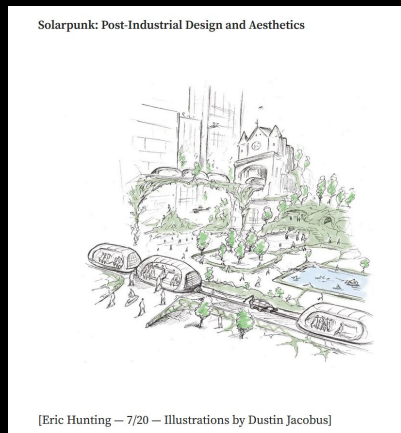


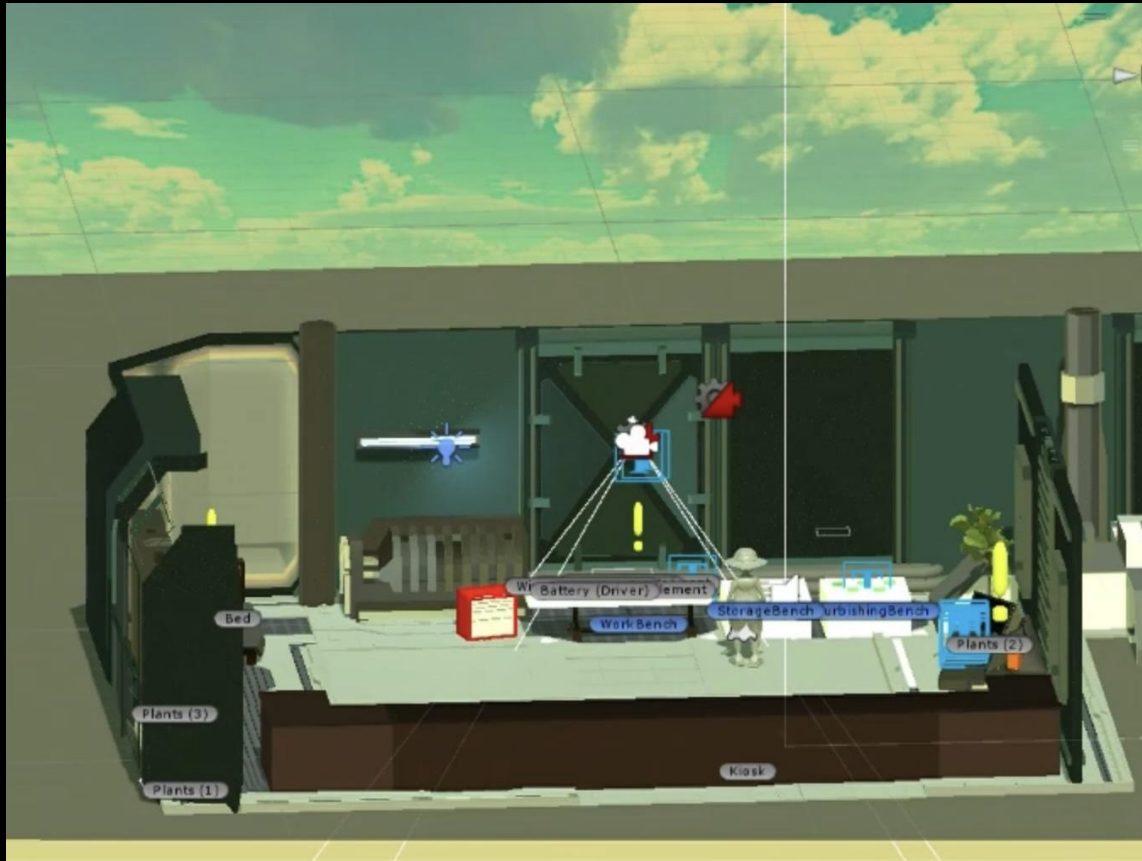
Environmental Storytelling:

As the development of the game progressed, I got more and more preoccupied with creating assets and at first I did not pay enough attention to the workshop environment. I was expecting the cutscenes to do the narrative heavy lifting and the in game world could be subpar as long as it gave a vague idea of my original workshop concept one.

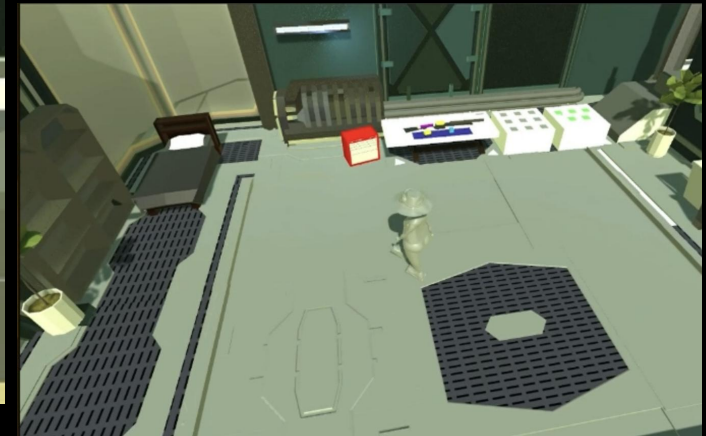
I quickly realised that a single snapshot of the game should be able to deliver the flavour of what the narrative entails. Eric Hunting's Post-Industrial Design and Aesthetics was a major jumping off point in imagining a SolarPunk model town that has shed its old industrial ways would look like. I knew I wanted the effects of climate change to be apparent on first look, the image of a flooded town with vegetation retaking concrete came to mind. Since the workshop is a small contained scene, I made sure to give it some openness by adding windows all along the left wall.

The windows act as a peak into the world, not big enough to over take the cluttered and homely aesthetics of the workshop but just big enough to allow players to look outside and bring their own interpretations to what happened to town which reinforces the questions at the heart of the frame narrative. This kind of environment building also informs my narrative, as after creating workshop two my teammate Sky asked how are people getting to each other if everything is flooded? Which promptly influenced the narrative into changing people coming into the workshop to sending their drones in instead, this aligns with the Scoping Player Interaction part of my design framework.





Workshop Version. 1





Workshop Version 2



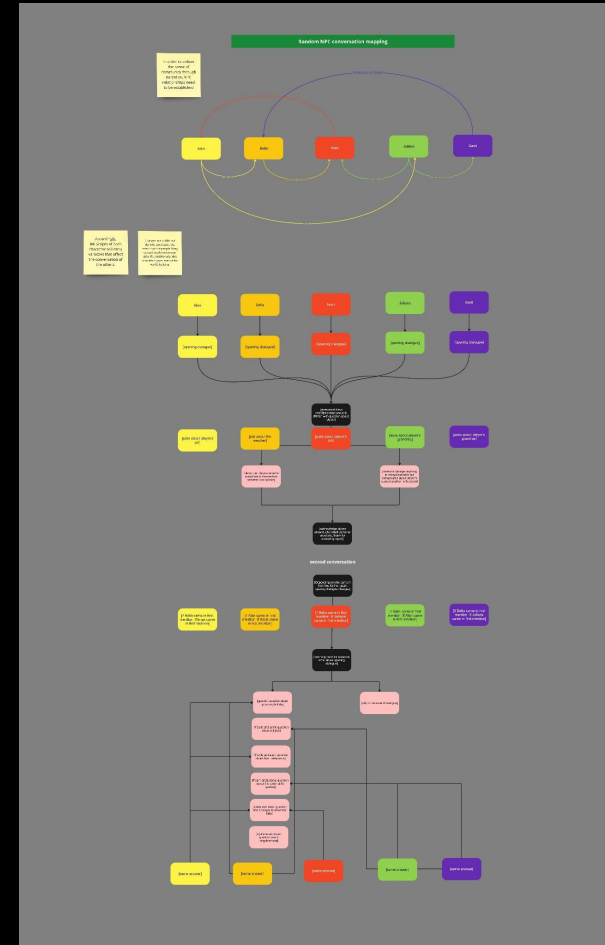
NPC Relationships:

Keeping in mind my design framework, Narrative Driven NPC relationships is a huge part of reinforcing player choices and consequences. My teammate Kathi, also has a framework focused on community building so it was very important for me to try and emulate that within my narrative.

The following graph shows off the relationships the NPCs have with each other and how that informs the dialogue branch the player receives from them. In order to truly sell the feeling of a isolated town and a close knit community, I made sure that each NPC has a unique relationship that they share.

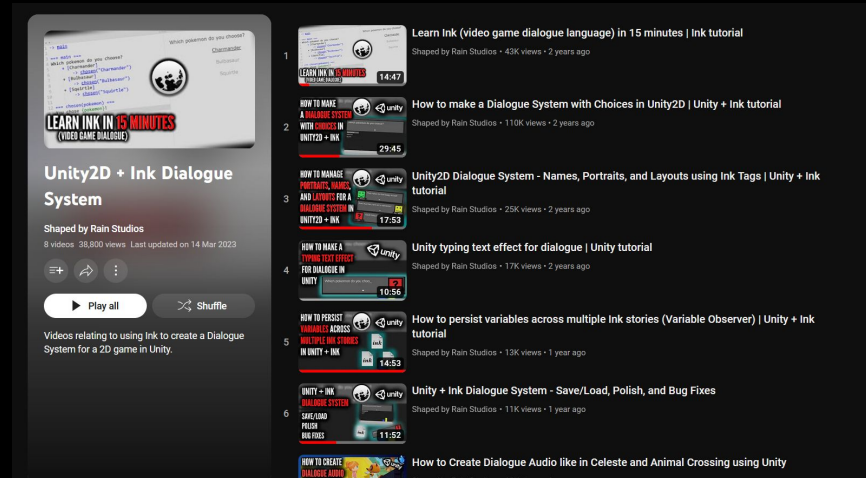
When the player meets two NPCs which have an established relationship, they can enquire further about the relationship, thus finding out something about the world building in general, and/or receive anecdotes about the town or about a specific entity that I want the player to be aware of. For example, if the player talks to Sami on first repair and then Manuel on second repair, they can find out about the community garden and also that their grandmother frequently visits this community garden.

The goal is to give the player characters they can attach to so that the NPCs can influence their actions within the game in turn giving me control over player unpredictability.



In Game Dialogue:

The in game dialogue is handled through Inky by Inkle Studios. I followed a series of tutorials by Shaped by Rain Studios to build our dialogue system.



The dialogue system works via 3 major scripts, DialogueManager, DialogueTrigger and DialogueVariables. The DialogueManager loads story via compiling Ink JSON text assets, allows the player to make a choice within the dialogue and parses Tags within the Ink script to animate NPC portraits and assign correct character name to the NPC. DialogueTrigger triggers the visual cue on top of NPCs and triggers dialogue on player interaction. DialogueVariables loads, stores and updates all variables across all NPC scripts via compiling a Global include file and updating a dictionary item called variablestoStory.

Ink itself is a rather simple yet powerful scripting language that allows you to write out your narrative like scripts and then apply logic to them. The following is an example of an NPC script, the format of which is uniform across all NPCs expect core narrative ones.

```

1 INCLUDE Globals_Dialogue.ink
2 [talk1k1_done == true] --> already_talked | -> talk_1one | #Speaker: Manuel #portrait: manuel_default
3 [already_talked == true && talk1k1_done == true] --> already_talked2 | -> talk_two
4 == talk_two ==
5 [manuel_talk1 == false && sam1_talk1 == true] --> talk_tuomannell1
6 [manuel_talk1 == false && alan_talk1 == true] --> talk_tuomannell2
7 [manuel_talk1 == true] --> talk_tuomannell3
8 Oh hello hello! Your Grandpa has told me so much about you, #Speaker: Manuel #portrait: manuel_default
9 This drone will hand you my thing. It should be a small fix, but this is my little helper so be nice.
10 -> generic_question
11 -> END
12 == generic_question ==
13 | How did you break this?
14 Ugh, it is kind of embarrassing so I'd rather not tell you. #Speaker: Manuel #portrait: manuel_sad
15 -> talk2k1_done == true
16 -> END
17 | Yes I'll get to repairing this.
18 Thank you! #portrait: manuel_happy
19 -> talk2k1_done == true
20 -> END
21 == talk_tuomannell ==
22 | Sam! raved about your place- and your hat oh wow- so when this thing broke, I knew you were the repair maestro I had to hit up. #Speaker: Manuel #portrait: manuel_default
23 [manuel_talk1 == true]
24 [manuel_talk1 == true && alan_talk1 == true] --> garden_question
25 -> END
26 == garden_question ==
27 | Grandpa mentioned you two! How's the Garden doing?
28 We've gone vertical with our planting, stacking those greens high to make the most of our space. #Speaker: Manuel #portrait: manuel_happy
29 It's like a lush skyscraper of veggies and flowers- who knew a garden could be so space-savvy?
30 And get this, we've set up special ozone absorption shields to protect our green buddies from the harsh rays. It's like our own little plant sunscreen and it has really made our yield so much better.
31 -> positive_state == positive_state + 1
32 -> END
33 | Sam! I got this art of saying more with fewer words. Maybe a hint of mystery suits you too?
34 Oh, #Speaker: Manuel #portrait: manuel_sad
35 I'm sorry.. I'll be quiet.
36 -> negative_state == negative_state + 1
37 -> END
38 == talk_tuomannell2 ==
39 | I mean really understands the art of silence-let's give it a shot..
40 Oh, #Speaker: Manuel #portrait: manuel_sad
41 I'm sorry.. I'll be quiet.
42 -> negative_state == negative_state + 1
43 -> END
44 == talk_tuomannell3 ==
45 | I mean really understands the art of silence-let's give it a shot..
46 Oh, #Speaker: Manuel #portrait: manuel_sad
47 I'm sorry.. I'll be quiet.
48 -> negative_state == negative_state + 1
49 -> END
50 == already_talked ==
51 | Thanks for doing this! #Speaker: Manuel #portrait: manuel_happy
52 -> END
53 == already_talked2 ==
54 | Thanks for doing this! #Speaker: Manuel #portrait: manuel_happy
55 -> END

```

```

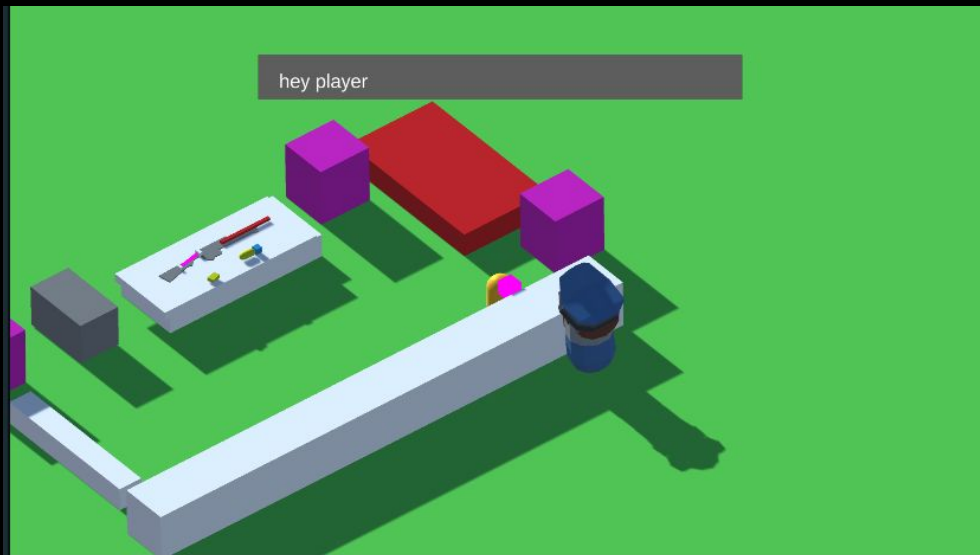
1 INCLUDE Globals_Dialogue.ink
2 [talk1k1_done == true] --> already_talked | -> talk_1one | #Speaker: Manuel #portrait: manuel_default
3 [already_talked == true && talk1k1_done == true] --> already_talked2 | -> talk_two
4 == talk_two ==
5 [manuel_talk1 == false && sam1_talk1 == true] --> talk_tuomannell1
6 [manuel_talk1 == false && alan_talk1 == true] --> talk_tuomannell2
7 [manuel_talk1 == true] --> talk_tuomannell3
8 Oh hello hello! Your Grandpa has told me so much about you, #Speaker: Manuel #portrait: manuel_default
9 This drone will hand you my thing. It should be a small fix, but this is my little helper so be nice.
10 -> generic_question
11 -> END
12 == generic_question ==
13 | How did you break this?
14 Ugh, it is kind of embarrassing so I'd rather not tell you. #Speaker: Manuel #portrait: manuel_sad
15 -> talk2k1_done == true
16 -> END
17 | Yes I'll get to repairing this.
18 Thank you! #portrait: manuel_happy
19 -> talk2k1_done == true
20 -> END
21 == talk_tuomannell ==
22 | Sam! raved about your place- and your hat oh wow- so when this thing broke, I knew you were the repair maestro I had to hit up. #Speaker: Manuel #portrait: manuel_default
23 [manuel_talk1 == true]
24 [manuel_talk1 == true && alan_talk1 == true] --> neighbor_question
25 -> END
26 == neighbor_question ==
27 | I'm glad I could help! How do you know I mean?
28 If I glance out my window, I can peek into her living room-it's lovely and cozy in there with those handmade rugs. #Speaker: Manuel #portrait: manuel_happy
29 But it just needs more green, you know? Living in a concrete box without nature, can you even imagine?
30 Even if I never had the garden, I would still never be able to live without Thoth at home.
31 -> positive_state == positive_state + 1
32 -> END
33 == talk_tuomannell2 ==
34 | I mean really understands the art of silence-let's give it a shot..
35 Oh, #Speaker: Manuel #portrait: manuel_sad
36 I'm sorry.. I'll be quiet.
37 -> negative_state == negative_state + 1
38 -> END
39 == talk_tuomannell3 ==
40 | I mean really understands the art of silence-let's give it a shot..
41 Oh, #Speaker: Manuel #portrait: manuel_sad
42 I'm sorry.. I'll be quiet.
43 -> negative_state == negative_state + 1
44 -> END
45 == already_talked ==
46 | Thanks for doing this! #Speaker: Manuel #portrait: manuel_happy
47 -> END
48 == already_talked2 ==
49 | Thanks for doing this! #Speaker: Manuel #portrait: manuel_happy
50 -> END

```

Ink's divert function allows me to structure dialogue in a way where I can code checks, with the use of variable bools, for what the player has seen or hasn't seen, for example, if a player calls upon Manuel twice in one day, he will not get the same conversation twice, instead the script recognises this is their second time and diverts the player to a new conversation knot.

Version 1:

- Ink dialogue system implemented
- Content NPCs given storeFrontBark Ink text asset,
 - Narrative reserved for NPCs spawned separately from repair list.
 - Done so to control narrative flow.
- Ink asset rotates through a set list of prompts
 - Can be used on all NPCs and quickly generate a variety of interactable NPCs.
- Simple check for when player comes back a second time to acknowledge that,
 - To create a sense of a more intelligent dialogue system.
- Dialogue box simple, minimal, with a text bubble in mind.
 - Giving myself limitations to keep dialogue word count low.



```

1  INCLUDE Globals_Dialogue.ink
2  {finished_start == true: -> already_talked | -> talk_start}
3  == talk_start ==
4  hey player <> # speaker: Random #layout: bark
5  {&{~I need you to fix this| can you fix this for me| I am an idiot I need help| I really need this fixed| can you help me?| this is so important to me}}
6  ~ finished_start = true
7  -> DONE
8  == already_talked ==
9  {&{~ I'm waiting | Thanks!| yeah just this thanks}} #layout: bark
10 -> DONE

```

Version 2:

- storeFrontBark assigned only to content NPCs
- Narrative NPC distinguished from content NPC via dialogue box layout.
 - Done so to create even clearer distinction between NPCs that are just for gameplay vs narrative
- Layout animator applied to dialogue box to keep track of speaker and switch portrait sides on Ink tag call.
 - Done so to distinguish when player is talking and when NPC is talking as they share the same dialogue box.
- Dialogue box updated, with speaker name, portrait frame and dialogue choices.
 - Previously proposed, narrative branch would be handled by the way the player repaired the item
 - Since repair states not ready, standard dialogue choices added to maintain narrative goal.
- Dialogue choices set to record player choice and accordingly,
- Update narrative conflict Ink file and the choices now available to the player when conflict happens.



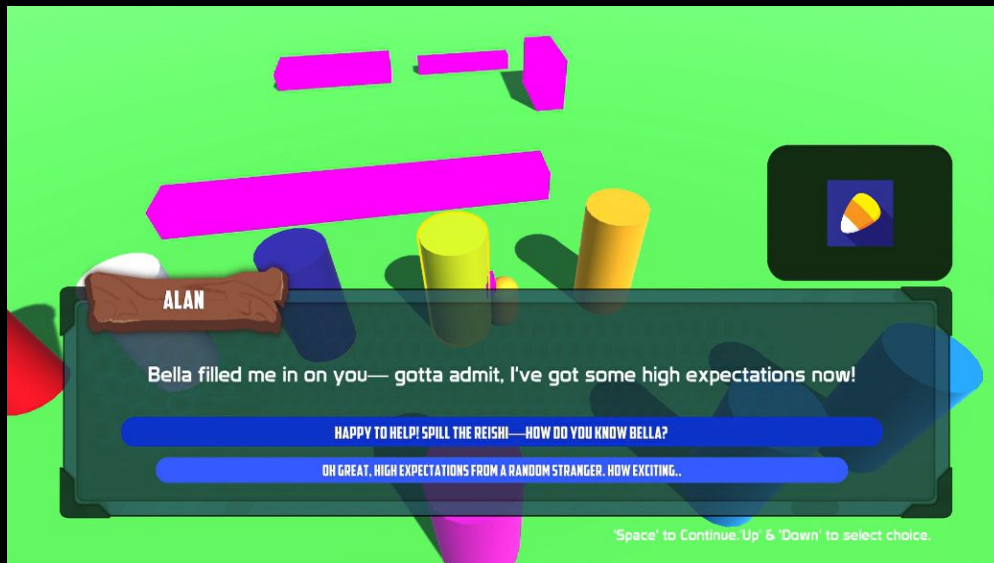
Version 3:

- All content NPCs now turned into narrative NPCs
 - The distinction between NPC was not serving the overall gamefeel
 - Made dialogue interactions too few and far between, making players lose interest.
- All five assigned unique script, Alan_Prologue, Bella_Prologue, Iman_Prologue, Sami_Prologue and Manuel_Prologue,
- With variables to update dialogue branch depending on who the player has talked to prior.
- Narrative NPC Priya turned into onboarding for dialogue system.
- Dialogue UI updated:
 - Dialogue Box, Speaker name and Portrait frame UI assets changed
 - Choices button now placed within dialogue box
 - Animator removed, layout now constant with only one speaker



Version 3.5:

- Dialogue UI updated:
 - Choice buttons now scale up on select to make selection more obvious
 - Dialogue box increased in size to fit all dialogue lines comfortably
- Testing dialogue only cues for hinting player choices will get remembered.
 - When players get to a choice that will be remembered, the dialogue will tell them exactly that.
- Redid all NPCs dialogue script to improve diverts and reduce delay between dialogue
- Added more unique dialogue when player meets a NPCs in established relationships.
- Reduced amount of dialogue required to get through before repair item is accepted to make faster, snappier gamefeel.



Version 4:

- Dialogue UI updated:
 - Choice buttons now work on mouse click
 - Character portraits added
 - Animator applied to character portraits.
- Dialogue choices that get remembered now have NPC reactions.
 - Just dialogue cues were not enough to hint, now with added portraits player can get added visual feedback on their choice.
- Second redo on all NPC scripts to improve diverts and decrease delay in retrieving dialogue.
- Added check to give feedback when repair is done and item is returned to player.



Critical Reflections:

- *Create a 3D Game.*

Prior to this unit, I have yet to work in 3D Unity. Figuring out the art pipeline, how it can be optimised and how it can be used effectively is my primary goal with this aim.

The artpipeline took me a solid 2 weeks of non stop work to get down properly. My fatal flaw here was drawing concepts by hand, redoing them and only then getting to the player character modeling and meshing. If I could go back and redo this project, I would definitely use either ready made concepts or concepts that are much looser in finish before getting into the sculpting, since my 2D artist brain couldn't really account for what details get lost in a 3D sculpt and what details are extraneous to add to a character design. Fortunately, now that I have done it once I can recreate the desired effects much faster, Unity 3D does have a learning curve but it is not too steep and you only get faster at it as you go. I believe we made a very good call by keeping this game in the low poly aesthetic as creating anything more realistic looking that this simple character would've added another week to the art pipeline. We already have so many 3D objects to worry about in any given scene, the load on the build would be crushing if we had gone for a more realistic aesthetic.

- *Use Inky to create a Branching Narrative.*

Inky has been used for a lot of very successful narrative games (80 days, Where Water Tastes Like Wine), even though both games are mechanically light, their narrative interaction (not just the writing itself but the input taken by the system) has carried the games to critical acclaim.

Inky proves to be an accessible language for quickly diving into the creation of narrative-driven games. Drawing from my experience with Twine and Bitsy, both capable tools for narrative game development, Inky stands out for its extensive logic conditions and customization possibilities. It excels in scenarios where narrative takes center stage from the project's inception. However, the simplicity of Inky becomes a challenge when attempting to integrate it seamlessly with other C# scripts that might play a crucial role in shaping the story. Unlike its counterparts, Inky lacks the flexibility to easily connect with external scripts, making it less ideal for projects where intricate interactions demand a seamless blend of narrative and functionality.

Yet, the constraints imposed by Inky present an opportunity for creative solutions. While this limitation may necessitate repetitive patterns when dealing with numerous NPCs, it also encourages inventive workarounds to achieve desired outcomes. It's worth noting that Inky's logic focuses primarily on making text interactable and advancing the narrative, rather than enhancing the text's presentation or formatting.

In conclusion, Inky shines in narrative-centric game development but falls short when intricate integrations with other scripts are a priority. The trade-off lies in the need for creative problem-solving within Inky's constraints, ultimately determining its suitability for a particular project's requirements.

- *Create a believable and alive Solarpunk world.*

The Solarpunk genre often lacks a realistic edge because a core part of solarpunk is optimism and any realistic adage would dull the shine and turn the narrative cynical. I want to balance a bit of realism with solarpunk and deliver a world that can be realistically imagined in 20-30 years time.

In this solarpunk narrative, the world undergoes a transition to clean energy amid ecological collapse, resulting in stark wealth disparities. Communities seek solace amidst isolation caused by a lack of global communication, giving rise to Smallernets. Model towns, emblematic of self-sustainability, face power struggles with governments and corporations eyeing their resources. Isolationism becomes both a unifying force and a source of internal conflict, setting the stage for a complex interplay of resilience, betrayal, and the unwavering pursuit of control. The narrative effectively explores themes of environmental consequences, economic tensions, and societal responses, and sets the stage for our players to enter our game and make the world more solarpunk, more sustainable and more closer knit.

I believe in my narrative planning I have managed to create a backdrop of a society on collapse and brought the players into the world with the goal to restore it. The implementation of this narrative is currently lacking within the game, without Acts 1-3, the progression from dilapidated town to a picturesque solarpunk world cannot be seen in the game as it stands, however, we the systems complete and running smoothly I can focus on creating, adding and polishing content very soon. The item designs currently lack visual polish and do not immediately invoke a solarpunk world. I believe the addition of cutscenes and the completion of the prologue from start to narrative gate will make this Solarpunk world even more believable.

- *Examine application of Design Framework on a very small scale project.*

My design framework was based on research on very big budget AAA and even AA titles, I want to see how feasible it is to maintain my proposed philosophy of player agency reflecting on 4 levels, Visual, Ludic, Gameplay and Gameworld.

Certain aspects of my framework are definitely much easier to adapt to a smaller scale.

1. *Interactivity Progression*
2. *Scoping Player Interaction*
3. *Narrative-Driven Affordances*
4. *Balancing Interactivity and Narrativity*
5. *Continuous Player Feedback*

These four aspects are incredibly helpful in doing the game design planning prior to actual building. Because I took Interactivity Progression and Scoping Player Interaction seriously before building the narrative, it translated into the game seamlessly. Narrative-Driven affordances are implemented in my narrative through the frame narrative. The player is allowed to interact with the world however they want, but the way they get remembered in the frame narrative should help define the optimal path desired for an ending in the game. Balancing Interactivity and Narrativity actually came a lot easier than I thought, and it was largely due to how adaptable small teams can be and how involved everyone is in the game design process. In a larger company, perhaps the narrative and interactive balance is harder won due to the lack of communication between departments. Continuous Player Feedback is also easily scalable, though our game falls short in its implementation as it only exists in the visual feedback space and not in the ludic, gameplay, or gameworld.

1. *Narrative-Driven NPCs Relationships*
2. *Structured Personalities with Feedback Variability*

These other four aspects are time-consuming and not very scalable to a smaller studio. Creating Narrative-driven NPC relationships can occur on a small scale, but it will not be as effective as in AAA games, as they can generate a more involved NPC personality. Structured personalities with feedback variability both require time, multiple iterations, and possibly a dedicated team to perform in a seamless and well-integrated fashion.

- *Examine application of Design Framework on a very small scale project.*

My design framework was based on research on very big budget AAA and even AA titles, I want to see how feasible it is to maintain my proposed philosophy of player agency reflecting on 4 levels, Visual, Ludic, Gameplay and Gameworld.

Certain aspects of my framework are definitely much easier to adapt to a smaller scale.

1. *Interactivity Progression*
2. *Scoping Player Interaction*
3. *Narrative-Driven Affordances*
4. *Balancing Interactivity and Narrativity*
5. *Continuous Player Feedback*

These four aspects are incredibly helpful in doing the game design planning prior to actual building. Because I took Interactivity Progression and Scoping Player Interaction seriously before building the narrative, it translated into the game seamlessly. Narrative-Driven affordances are implemented in my narrative through the frame narrative. The player is allowed to interact with the world however they want, but the way they get remembered in the frame narrative should help define the optimal path desired for an ending in the game. Balancing Interactivity and Narrativity actually came a lot easier than I thought, and it was largely due to how adaptable small teams can be and how involved everyone is in the game design process. In a larger company, perhaps the narrative and interactive balance is harder won due to the lack of communication between departments. Continuous Player Feedback is also easily scalable, though our game falls short in its implementation as it only exists in the visual feedback space and not in the ludic, gameplay, or gameworld.

1. *Narrative-Driven NPCs Relationships*
2. *Structured Personalities with Feedback Variability*

These other four aspects are time-consuming and not very scalable to a smaller studio. Creating Narrative-driven NPC relationships can occur on a small scale, but it will not be as effective as in AAA games, as they can generate a more involved NPC personality. Structured personalities with feedback variability both require time, multiple iterations, and possibly a dedicated team to perform in a seamless and well-integrated fashion.

- *Work with a team.*

Making games on your own is hard, I want to see if the end result benefits from having multiple people involved in the Game Design process.

Working with a team has been a wonderful experience, my teammates have been incredibly patient with my weaknesses. Supporting my struggles with coding and unity and any technical issues I ran into. They have also been incredibly supportive with my creative direction I took the narrative, any and all changes we embraced. I believe we got very lucky with how similar our game design vision ended up being when it came to this game and with experienced designers like Kathi and Vlad, I was able to scope down my artistic tendencies to do extravagant things that are impossible to in code. Sky's UI not only aligned with my design language but also provided inspiration for the color palette for the Grandma character.

Credits

- Low-Poly Simple Nature Pack by JustCreate
- (<https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153>)
- SimplePoly City - Low Poly Assets by VenCreations
- (<https://assetstore.unity.com/packages/3d/environments/simplepoly-city-low-poly-assets-58899>)
- Sci-Fi Styled Modular Pack by karboosx (<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>)
- Sci-fi Kit Bash Pack by tran.ha.anh.thu.99 on Sketchfab (<https://sketchfab.com/3d-models/sci-fi-kit-bash-pack-cfdd2f511afc417b969c7b7dc0e95a97>)
- Broken Car by Quaternius (<https://poly.pizza/m/Y67erogmR9>)
- Shelf Tall by Quaternius (<https://poly.pizza/m/TDgvluorcX>)
- Soldering Iron by J-Toastie [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/eLUe9OMQrj>)
- Drill by jeremy [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/93nEewogYE0>)
- Broom by Don Carson [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/cOMoz6gJFN>)
- Yughués Free Metal Materials | 2D Metals | Unity Asset Store Mixamo
- [QuickOutline] <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>
- Ink Integration for Unity | Integration | Unity Asset Store

- Vent by Quaternius (<https://poly.pizza/m/UDFenJ0U73>)
- Vines by Poly by Google [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/2jftIS8PMjZ>)
- Ranking plant by Jakob Hippe [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/2q-hEEe3Njb>)
- Cardboard Box Open by Kenney (<https://poly.pizza/m/i1lr4yJFY0>)
- ☆ friend maker ★ by Rowyn (https://picrew.me/en/image_maker/13228630)
- Sound Effect from Pixabay
- Music by Patrick A.
- from Pixabay
- Music by 23843807 from
- Pixabay
- Sound Effect from Pixabay
- Sound Effect by 666HeroHero
- > from Pixabay
- [Transition Sound]https://freesound.org/people/Vilkas_Sound/sounds/460475/

- Single bed credit Single Bed by Zsky [CC-BY] via Poly Pizza
- Bed by jeremy [CC-BY] via Poly Pizza
- Houseplant by jeremy [CC-BY] via Poly Pizza
- Fiddle leaf plant Fiddle-leaf Plant by Poly by Google [CC-BY] via Poly Pizza
- Scifi PC by Nick Slough [CC-BY] via Poly Pizza
- Table by Quaternius
- Counter Straight by Quaternius
- Black Smith Shop by Zsky [CC-BY] via Poly Pizza
- [Light Bulb 3D Model] <https://poly.pizza/m/eaqeYLhDkio>
- Butter Robot by Martin Calviello [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly
- Pizza (https://poly.pizza/m/9HIzs__db3k)
- Gardening Trowel by Pookage Hayes [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly
- Pizza (<https://poly.pizza/m/3Jrn0djxEEf>)
- Bamboo Garden Scaffolding by Ryan Korsak [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>)
- via Poly Pizza (https://poly.pizza/m/b_yuq_oDGQI)
- Plants - Assorted shelf plants by Jakers_H [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via
- Poly Pizza (<https://poly.pizza/m/5COCzyz489J>)
- Fold Out Ladder by Jarlan Perez [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly
- Pizza (<https://poly.pizza/m/0KCCeQA6caU>)
- Hanging Flower Pot by Zsky [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/qS7bDdlSw7>)
- Cardboard Boxes by Quaternius (<https://poly.pizza/m/V9KbWC8Vd6>)
-

- SF Street by Alan Zimmerman [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/cnTMgkFoTS0>)
- Robot Kit v.1 by Jarlan Perez [CC-BY] (<https://creativecommons.org/licenses/by/3.0/>) via Poly Pizza
- (<https://poly.pizza/m/9L2Lt-sxzdP>)
- Pipes by Quaternius (<https://poly.pizza/m/GB6AFkoiZb>)
- URP Stylized Water Shader - Proto Series by BitGem
- (<https://assetstore.unity.com/packages/vfx/shaders/urp-stylized-water-shader-proto-series-187485>)
- Fantasy Skybox Free by Render Knight ([https://assetstore.unity.com/packages/2d/textures-](https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353)
- [materials/sky/fantasy-skybox-free-18353](https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353))