PROCESS DOCUMENT:

2D MONSTER FARM

Written by

Timur Lüdemann (49931458)
Hai Lam Dang (129129858)

Course

Game as Research

During

WiSe 21/22

Lecturer

Slawa Deisling

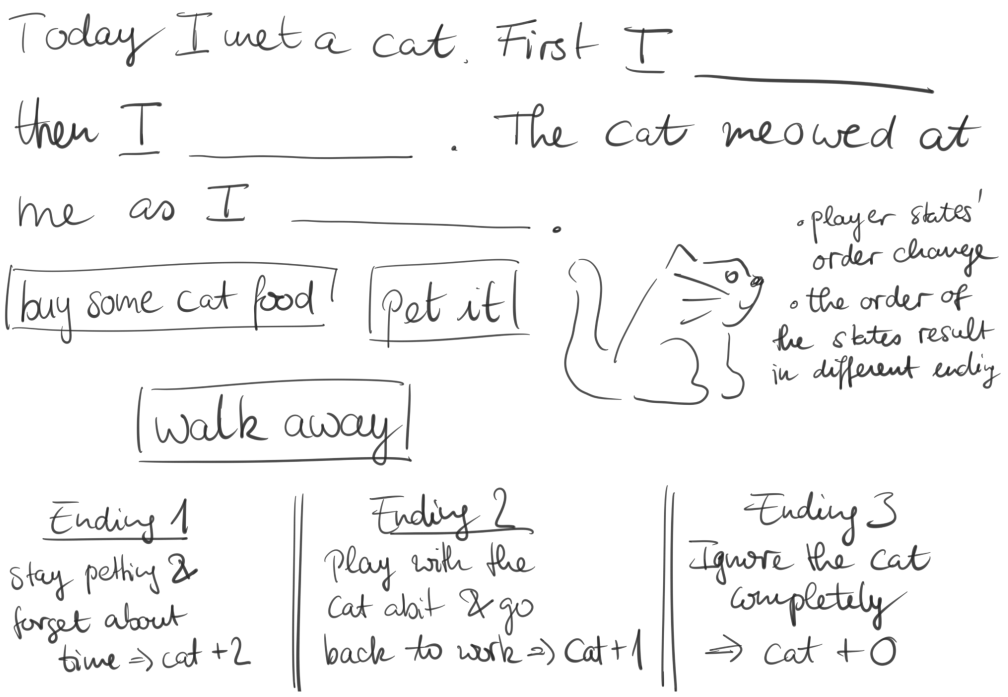
Table of content

2D Monster Farm (Introduction to game)

2D Monster Farm is an idle farming simulator which is ideally played passively, where the player can plant and raise monsters to hatch and sell them when they reach their mature stage. With the collected money, the player is able to buy expansion for the farm with different plots and even cooler monsters. The goal of the game is to expand the farm to the fullest and collect as much money as possible.

1. Our approach to the work:

It was not hard to divide our tasks, as it was clear who does what. Instead of each of us trying something out of our comfort field like Timur doing 2D arts and Hai Lam doing programming, we choose our strong point and work on it. So Timur is the team coder and Hai Lam is the artist. Our first meeting was brainstorming on the theme given by our lecturer Slawa. The theme was “Programming Pattern” and the game should have a more passive gameplay. An example Slawa gave us was a soda machine and the game Paper Please. With those as our base, we sketch out our ideas. Hai Lam presented her idea of a more text based game using State Machine.


Today I met a cat. First I ________
then I ________ . The cat meowed at
me as I ________ .
buy some cat food
pet it
walk away
◦ player states' order change
◦ the order of the states result in different ending
Ending 1
stay petting & forget about time => cat +2
Ending 2
Play with the cat abit & go back to work => Cat+1
Ending 3
Ignore the cat completely => cat + 0

The idea was based on the “Fill in the blank” using state. But after re-thinking, this idea was too complicated for us to make, so we looked elsewhere. Timur presented his idea, which was inspired from a Game Jam title, of an isometric farming simulator. We both think this idea is very fitting, as we both like idle games. Not to mention, this type of game is very fitting to our more passive gameplay as part of the performance requirement.

After we have our basic idea in place, we look for more inspiration for such a game. We looked for farming sim games like Stardew Valley, the Farming Simulator franchise, etc. But we all think farming crops was a bit boring, so we tweaked the concepts into a monster farm instead.

The player in our game was supposed to be a more mechanical observer who controls the monster farm through buttons and panels. But this was unrealistic in scope, so we removed this “observer” filter and let the player interact with the farm directly instead through buttons presses like the initial plan.

2. The theme is „Programming Patterns“ - what did you come up with? Why? Did your initial idea change over the weeks? Why? What player feedback was useful for you, which wasn‘t?

At the beginning we didn't know what to really think of the theme “Programming Patterns”, it was hard to imagine a game out of it. The last projects had a real “strict” theme, for example “Movement”. With this theme, you really knew what the boundaries are and what the main part of the game should be, which wasn't the same with this project.

In the very early stages we thought about just doing a vending machine simulator as this was an idea from Slawa that he talked about with the whole course. But with the project coming to a start, we decided on an idle game. We got the idea from a game jam video, which we thought was awesome. The game for us fulfilled the theme by having many ways of implementing different programming patterns we have learned. Also, they are visible while playing. Our initial idea didn't change much over the weeks except one point, this was the idea of having a console where the player had to write commands in to make something happen. For example, “plant [monster] on b2”. We thought this idea would be really cool but decided fast against it because we thought this would be very hard to implement and make it not frustrating for the players.

So much of the player feedback was very valuable for us and helped us to develop the game even faster and polished. As we are only two people, play testing and thinking about everything is really hard, so the weekly feedback sessions really helped to get bugs fixed and implement new ideas that would increase the fun of playing and the overall user experience. Some of the best feedback was: Instead of showing the monster that is hatching from the egg in the story, show the egg in the store, so it will be a surprise for the player what will hatch out of the egg. This will increase the curiosity and the will to buy this new egg. Also, the feedback of adding visual clues to the game was really helpful too because without the visual feedback this game really felt boring and clunky which was eliminated by the visual feedback, which also made the game fun again.

We used different programming patterns like : Flyweight for optimization, Singleton for Soundmanager, (Event System) for the NPC and Sound and states for the shop but not with a state machine. First we wanted to use state machines for the monsters and later the tiles, but we decided not to because then we needed to come up with 3 more state machines, and we just didn't have content that would need them. So we didn't want to forcefully add new features for state machines which don't really fit into our game.

3. What was good? What was bad?

Team

Good: Our scope was very clear and well-defined. We wanted to create a simple farming simulator with monsters. As the scope was simple in nature, we fulfilled it very naturally without having to scale it down like we usually do with other projects. In fact, we managed to add a bit more to our game outside our scope such as the NPC, the animated and gradient background and the little world-building which can be found within the NPC’s dialog.

Bad: Despite the implementation of our core gameplay loop, we sadly had to scrap a few of our secondary game features which would improve part of the games. One of such features was the idea of a season system, where specific monsters would receive a bonus hatch rate in the correct season. Another planned system was something similar to the stock market. Here it was planned that the value of the monster can be changed depending on the stock market. If the market for a specific monster was in high demand, that monster would be worth more than usual. Of course, the opposite was possible too. If the market was overflowed with a monster, that monster’s value was then lower than normal.

In our play test for the version v.0.08 (3.1.22) a suggestion was made to us to add a preview system when we select to buy a monster. With this planned system, the player could preview how the monster would look on a specific tile. However, the idea was suggested too late. We tried to come up with a preview system, but it didn’t work and only caused more problems. It didn’t fit well with our existing mechanics and their codes, so we decided to drop it in favor of a stable game.

Timur

Good: I think overall the project was awesome, I had fun making this game, and also I have learned much. I'm happy with the results and how we got to them. As me and Hai Lam haven't had any issues deciding on a game idea, it was awesome to work on a game I enjoy, and my partner is also enjoying. We both clearly had our strong areas and committed to them, which made working together so much easier. I really liked the challenge of creating the game idea from zero while also trying to make the game as planned. I'm really happy with what I did in this project, of course there were some ups and downs, but I overcame most of the coding problems, and we got everything done we wanted to have done. What went good the most, was the core game loop, I set up the system to plant and harvest monsters really fast, and it worked as intended.

Also, I worked with sound for the first time and that went really well too, I didn't expect that at all because I always thought it was much harder. I had many areas where I needed to read myself into (audio, isometric tiles, shop system, the power of scriptable objects and programming patterns) and those helped me to get way better at developing a game. I feel like through this project I learned so much more than in the first two semesters, I feel way more skilled after this semester than after the first two semesters, which is for me a huge success.

Bad: Afterwards I now know how I could have done a lot of things better but the main things that went bad during the development of the game were that the tiles should have been a state machine, so that every tile has a unique state and the tile would change to the specific tile that you bought from the shop. This way it could have easily adapted to the variables that this tile should have and the effects. I figured this way too late in the development and changing it would have been too troublesome because I would need to change a lot of the system. The effects for the tiles are now very unintuitive for future updates, this wouldn't be a problem if they would have a state machine.

Secondly I should have started to implement “saving” earlier in the game, we didn't really think about it until really late in the development and with that we had a lot of problems to develop a working save system which not only saves the money instead it should save the whole game state (monster stage, bought tiles, food on the tiles, bought monsters). Now we only save the volume and the money, which really sucks for players.

Hai Lam

Good: Overall, I would say this project was a good experience for me. There were many things I’m very proud of, such as the monster designs. Before, I didn’t have the chance to draw unique looking monsters. While I’m not exactly a huge die-hard fan of famous games which are famous for their monsters, like Pokémon or Monster Hunter, I always appreciate all the unique looks of those monsters. If there was more time, I would love to create even more to not only flesh out the game but also to train myself.

I was also very glad monsters were not the only thing I had the chance to draw. As the team sole artist, I had to create other assets such as the tiles and background. Isometric drawing was something I never did besides in a math class. So to create isometric drawings while making them look visually interesting was a challenge. I thought of how I could create a uniqueness for each tile and the process was very quick. I simply make each tile represent an element: TreePlot = Wood; SkyPlot = Air; LavaPlot = Fire. Furthermore, I’m very glad this process was very easy instead of me overthinking. I learned that sometimes simplicity is the best. Simple designs make the players understand faster than overcomplicated one.

One of my highlights of this project was certainly my NPC. He was something I spontaneously created while Timur was sick. Neither of us even discuss having an NPC in the game or not. So I was very happy to hear the very positive feedback this NPC brought to our game: Not only does he make the screen less empty, he also gives this idle game a bit more character by giving some comments about each monster or the world in general. Funny enough, he also helps us fulfill our performance requirement in terms of the amount of Event Systems we have to have in our project.

Bad: While it’s not exactly a bad point when I was designing the monsters, it was surprisingly to come up with original-looking monsters. I tend to fall back to the classic “using a real animal as base and spice them up” like Pokémon. But I also learned that fully original ideas are very hard or next to impossible in the creative field. What I should learn was to let all these existing monsters from other games or real life creatures influence me, and put my own spin on them. This would make them look less copy, while still familiar enough for the players to associate this game’s monsters with something similar they have seen before.

I did want to create an idle animation for each stage of the monster to make the game less static. But due to time, I instead only animated the NPC and the floating background. It certainly gives the game some motions, but there is an obvious contrast when looking at the static monsters in comparison to the background.

4. Team meeting:

Team meeting:
(https://itch.io/dashboard/game/1288056/devlog)

We had a meeting each week when we talked about feedback, ideas and how the project is going in general. In those meetings, we mostly talked about the feedback and if we would implement it or not. Those meetings were held in discord voice, in discord we also chatted most of the time every day, if we had smaller things that didn't really need to be communicated by voice. Every meeting was like 30-60 minutes, depending on how much feedback we got and what we wanted to implement for the weekly version. Those meetings also were useful for helping each other how to maybe implement things or giving the other ideas about what to do.

In the "Who did what and how" section, there are the most things listed we talked about in every meeting for the specific weeks + in our changelogs on itch.io is also everything listed we talked about and then implemented for the week's version.

5. Who did what and how

1. v.0.01 (Prototype)

Hai Lam: My first idea for the game once our lecturer Slawa announced was actually more text based. Something similar to "WILL: A Wonderful World" [1] where the player can shuffle the event and a new result will come out. Of course, as this is a group work, Timur and I had to talk to each other. He gave some good feedback on my pitch as it might be a bit too "passive". He thought it would be better if there was more interaction without being "too much". So our next idea is some kind of simulation game. The game should also be modular, so we can build as many State-Machine or Event Systems as we need to fulfill the performance requirement. Timur then presented to me the idea of a farming simulator, which was inspired by a game jam's entry called "maelstrom.exe[2]". I think the idea is indeed better than my original idea. Also, it's great that each of our strong points are completely opposite of each other. So I decided to take this opportunity to be more creative in art and learn more about arts in Unity. For the prototype, I made a decent-looking tile and our first monster, so Timur can set up the rest.

Timur: For the prototype, I've implemented one tile that Hai Lam made and showed how our game will work. It was able to plant and hatch one monster on this one tile. We wanted to get our idea out and see how Slawa and the other students react to the idea. We did not implement more because we first wanted to really get our plan and scope down before we commit to doing more things, and for that the prototype was perfect.

2. v.0.02

Hai Lam: As it's only the first version, a lot of things are yet to be implemented. At this current state all I can do is have some placeholder arts which I will definitely change in the future. I did a new monster type: Lizard. At the current stage, I sadly can't do much but wait for the ground works to be done.

Timur: For this version, I needed to expand our game idea to get the game core loop going and also make visible what kind of game this will be. I implemented a menu and some tutorial text, so the players know what to do because for the earlier versions there are of course missing visual clues to lead the game flow. Besides that, the shop has added another crucial point of the game.

In the prototype you just clicked on the tile and the monster was hatching but for the future we know that we will have added more monsters so there needs to be a proper way to choose which monster to plant + the monsters should cost and give money, so a shop was very important for the early stages. Additionally, I've added the feature to zoom in/out with the mouse wheel and to drag the camera with left click. As we knew we would expand the farm, this feature is a very nice quality of life change. The shop was kind of a struggle for me, as I never really worked this “in-depth” with the unity UI / Text Mesh Pro system.

3. v.0.03

Hai Lam: We received a lot of good feedback from the course, which we try to fix and implement for the next update. Our ground works are practically done by now, and I can start working on more unique monsters as opposed to the placeholder "Dragon" and "Lizard". For the third monster, I thought about something indescribable. More info on my design process is in my design journal. But overall, I'm very satisfied with the final design. This monster gave me some ideas for future designs as well.

Timur: In this version, we took the feedback we got from the last week and implemented most of the things, such as a progression bar to visualize the monster growth and additional information about the monsters in the shop.

Also, some small quality of life changes like being able to open the shop with "B" instead of having to click the button. Besides, I've worked on many bugs that needed to be fixed like: Selecting two monsters at once in shop, selling monsters only while selecting them in shop, wrong color of the progression bar and the player being able to open the shop in the pause menu and vice versa.

4. v.0.04

Hai Lam: I notice that the scene is very empty when you open the shop. On the right side, you have your shop menu, yet on the left side is actually very empty besides looking at your plots. I did notice I could add something to the left side when the player opened up the shop. It's given that the space to look at your plots will be small but in a way, it helps the player focus on the plot more as it's right in the middle. As stated in my goal for this project and art journal, I want to be as creative and artistic as possible, since my partner Timur is an amazing programmer. I've never tried animating something, so this is an opportunity for me to try to animate something.

I noticed a lot of games have an NPC for the shop, so I decided to use this too. Not only that, but I thought it would be fun if the NPC would react or comment on what a player will buy. Simply to give them more character. More design process on how I create my "Monster Shopkeeper" NPC is in my design journal.

Timur: In this version, I did not work on the project because I had to visit the hospital and needed to rest the whole week. This was communicated with Slawa.

5. v.0.05

Hai Lam: I managed to fix a rather obvious bug from v.0.04 which was that the NPC will move with the camera. It was rather confusing as the solution was to have two cameras. One will render non-UI elements, while the other will only render UI elements. The solution was rather obvious in hindsight. Another problem I ran into when I was testing out the game, the “welcome” text of the NPC didn’t reset properly after you close it. Like in real life, the shopkeeper should greet you again when you visit them again. But in v.0.04 the text stays at whatever text the player last clicks. So it’s most likely a comment about the monster on display. I have to figure out how to reset my text. I already laid the groundwork from the last update, but sadly it didn’t work. I pick up where I left off: My general idea was simply when I close the shop, the currently displayed Scriptable Object (whatever it is) will be reset to a so-called “DisplayText_Data.Reset”. First, I thought I can't just reference my entire DisplayText system to the shop button so that when I close the shop it’s reset.

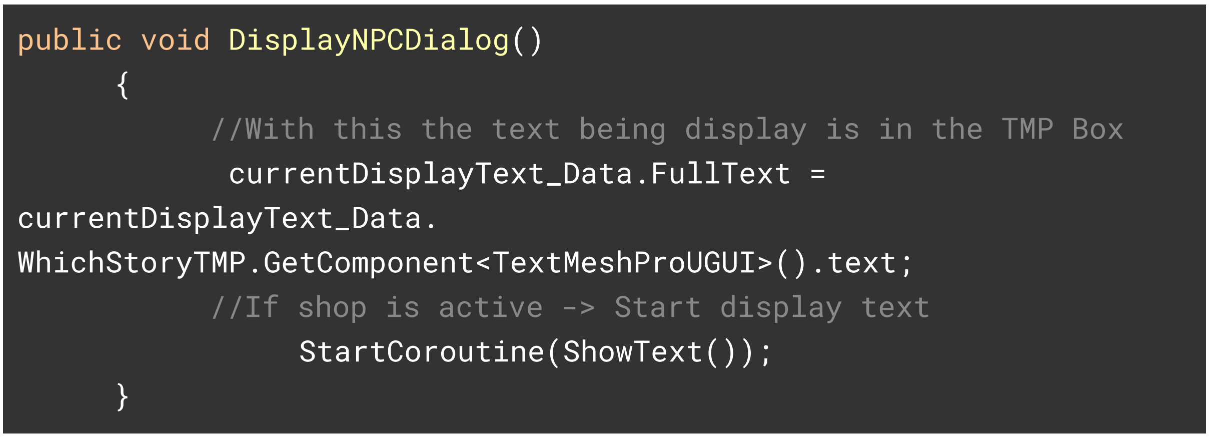
Problem 1: What happens if the shop is closed and the player clicks the button to OPEN it? NullReferenceException error because the object I referenced is not active due to Timur’s shop button’s logic before. That means my challenge is not to change his code, but to work as decoupled as possible.

My second idea is to have a separate game object which will “copy” a local Scriptable Object in itself and paste it on the displayed Scriptable Object when the shop is closed. That idea kind of works if I look at the inspector. I managed to reset the DisplayText_Data. Yet a new problem arises…

Problem 2: The text on the screen is NOT reset and the animation of displaying the test isn’t played. So my idea works on the inspector but not on the actual game. Admittedly, it feels really disheartening, but I have to rethink my strategy.

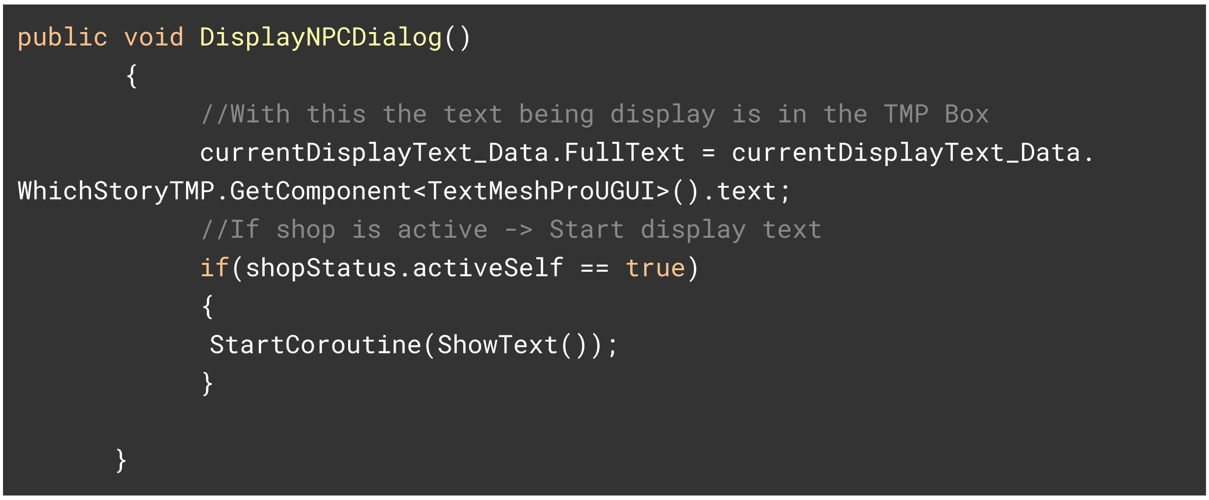
So it strikes me: I can use the power of the shop button Timur made. Instead of thinking about [closing the shop], I should change my strategy to [opening the shop]. Basically, my idea before still applies, but this time it activates when the shop is being open instead of closed! So I change the code inside my DisplayDialog.cs: First, I ask for the active status of the shop (which I reference in a variable). If the shop is open and only then, can this script start working? I added a simple if-clause to this script:

From:



```
public void DisplayNPCDialog()
      {
            //With this the text being display is in the TMP Box
             currentDisplayText_Data.FullText =
currentDisplayText_Data.
WhichStoryTMP.GetComponent<TextMeshProUGUI>().text;
            //If shop is active -> Start display text
                 StartCoroutine(ShowText());
      }
```

To:



```
public void DisplayNPCDialog()
       {
            //With this the text being display is in the TMP Box
            currentDisplayText_Data.FullText = currentDisplayText_Data.
WhichStoryTMP.GetComponent<TextMeshProUGUI>().text;
            //If shop is active -> Start display text
            if(shopStatus.activeSelf == true)
            {
             StartCoroutine(ShowText());
            }

      }
```

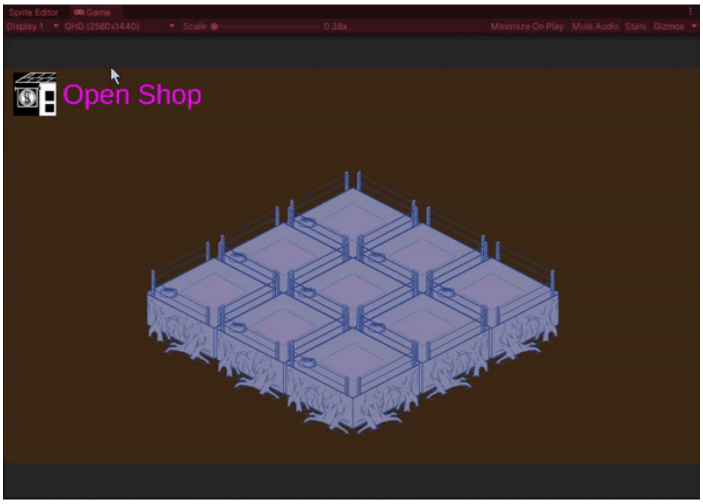
So a simple if-clause solves my problem of NullReferenceException when I click the shop button to close the shop. Sometimes the best solution is the simplest one.

With this down, my system is rather complete and modular enough to add new comments about new monsters, and it will always be reset back to “Welcome” text when the player returns. But then I think to myself, maybe I can improve the system further: I can build in a small bool check if this is the first time the player visits the shop. If it is, the shopkeeper will greet them like a new customer. If they are not first-timers, the shopkeeper will greet them like an old customer. I simply add the variable:

```
private bool isFirstTimeVisit = true;
```

Once the first shop is closed, the variable will change to false because it is not the first time visiting anymore. With that the “DisplayText_Data.Reset” will be changed to “DisplayText_Data.ReturnWelcome”. This welcome text is slightly different from the normal “DisplayText_Data.Welcome”. This feature is simply to make the NPC have more personality than before.

There is a potential bug or feature which Timur and I discuss, which is that the [Buy] button reset after the player closes the shop. As seen in this GIF:



My first thought was it’s a bug as I personally used to all the state of the button reset if I close the overall shop. But Timur made a very compelling argument that once we make the tiles so big and wide, this will be a QoL (Quality of Life) feature as it saves the player a few clicks. Especially when we will have many monsters and the player is thinking many steps ahead. So for now this would stay.

Another suggestion is to remove the text “Close/Open Shop” for either new icons (1 open shop icon and the other is closed shop icon” and the other idea is simply to have a border around the icon which changes color based on the state of the shop. We will have to think about it until the next update.

Timur: In this version, I have implemented being able to give the monsters' food, for faster hatching speed. This made me rethink the shop system because I needed to add another category because the food doesn't fit into the monster section. I came up with a small row for “misc” things that you can buy, such as food and tiles (in the future). Also, due to feedback from last week,

I changed the progress bar behavior. In the last version they did fill on each monster stage and for this version they continuously fill with each second. Also, Hai Lam added a new Plot, the sky plot, and we wanted the sky plot to give the bird egg a faster hatching time. To make this change work, I gave the monsters a speed variable that multiplies with the hatching time and set its default to one. Now, if the food is active, that variable is set +1. If the bird egg is active on the sky plot also +1 is added. So Skyplot and food gives the sky egg 3x the hatching speed, which is a huge bonus.

6. v.0.06

Hai Lam: One of the biggest features I want to include is a background. It might not sound much, but a good background can really enhance the game, especially in an idle game like 2D Monster Farm. But without any background, I want to make an animated one. As I notice, I won’t have time to animate every single monster in the game, so I want to make the game visually more “active”.
My first step to any design process is finding a few keywords. I asked Timur for his opinions when I threw a bunch of keywords for the background. He gave me his feedback by saying “I think this is good” or “It sounds rather boring”, etc. In the end, I came up with 3 keywords:

- Fantasy
- Lab/ Sci-fi
- Peaceful-like (semi-contradictory)

It reminded me of my first thought when we came up with this Monster Farm game: in a way, the whole idea was rather unethical. Similar to Pokémon, imagine you capture living beings and make them fight. But Pokémon is still packaged as a rather kid-friendly game. I want to go in a similar direction but showcase a rather dystopian-looking landscape without saying much.

With these three keywords in place, as the next step, I consult my collection of art books (as I love to collect art books of games I like). After finding what I like, I make a small mood board to visualize how my background will look.

Multiple sketches were made and, as always, I asked Timur for feedback and improvement. Having a second look, especially by someone who was not well-versed in art, could be a great way to see flaws where I usually wouldn’t see due to my fixed view.

Sadly, I couldn’t finish the complete background for this build, so only a very rough sketch is added. Instead, I added more NPC lines to give the game more personality.

Timur: I gave the main menu a new look so that the first impression of the game isn't as bad as before. Also added a visual clue for the tiles, they are now highlighted when you hover over. This was an idea from Slawa in last week's play test. And the biggest feature which was added to this version was that you can buy new tiles now. You start with fewer tiles in general, and now can buy more of the farm with your money. We didn't really know how we could implement it the best way but after sitting with Slawa he gave us some ideas and I chose to just place the tiles beforehand with a different sprite, that shows the plot can be bought.

If you now buy the plot the sprite will change and with the different sprite all the “logic” comes with it. On the not bought sprite you can't buy monsters on it and everything else, if it’s bought through the shop, a bool “isBought” will be true and if the bool is true everything is activated. Also, I did some bug fixing, each tile now has the correct bowl sprite. And the different effects on the tiles for the monsters work now too.

7. v.0.07

Hai Lam: The background was way more work than I expected. Especially due to how large the PSB file is. I also didn’t expect I had to animate each part of the background individually to create a random “floating feel” of the isles around. But I think the final result was worth it.

But I had lots of fun when animating the energy beam in the tube. Then an idea came to my mind: Since Timur added some sound effects, we can make each time the player harvest the monster, it turns into energy and beams up the tube.

A very annoying bug for this build was somehow the background is way too blurry. As I mentioned how big the PSB file was, it was actually 4k in size, so it really makes me wonder how such a big background can be so blurry. I didn’t have enough time to figure out where the problem lay for this build, so it was a high priority bug for the next build.

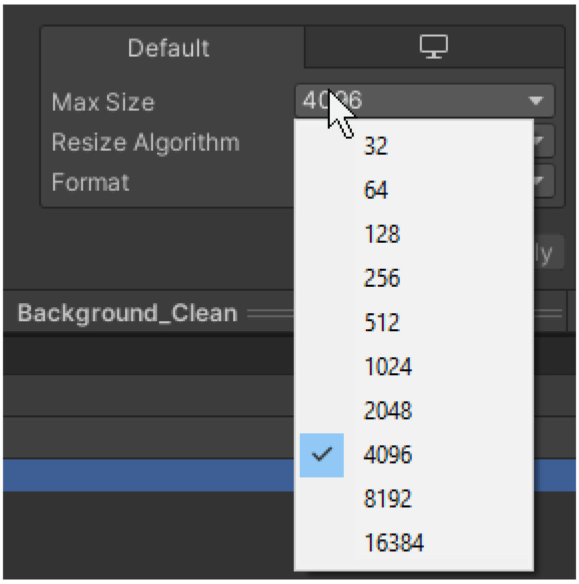
Timur: We decided to start polishing the game instead of adding new features because our next features are very big but not really needed because we already implemented everything we wanted for the core game and this was the biggest goal for us, do everything we had in our scope and then finish it to have a cool, little and playable game which was also good-looking & “polished”. We know that this wasn't the goal in general for this project, but because we got everything done we planned, we decided to not stress ourselves with things that weren't even planned to be in the game 100%.

Due to that I've added a lot of audio feedback to the game, such as different placement sounds, shop click sounds, menu sounds, food sounds and even more! This rounded the game up a lot more than we now had a lot of visual & audio feedback which makes the game feel a lot more finished. The feedback on it in general this week was great, the play testers liked it. Also, i have fixed the bug that you could spam infinite food on a tile even if the food bowl is filled, this leads to losing a lot of money.

Additionally, I’ve added a sound pitch, volume “randomizer” in the Soundmanager, each time a sound is played that will occur a lot, like placing tiles and food, the sound manager will give each sound play a different pitch in a certain range, so you don't are spammed with the same sound over and over. I got the idea from Mario, collecting coins back to back in a certain time plays the same sound over and over, but the sound doesn't sound the same because with each collect in a certain time a pitch is added to let it sound slightly different.

8. v.0.08

Hai Lam: The highest priority bug of mine was the blurry 4k background. Since I wasn't sure where the issue was, I had to play around with the setting and google. Luckily, it was an easy fix. When I imported the PSB file, Unity set “Max Size'' to 2048 as default. So that means my 4k image is being compressed down, losing lots of quality in the process. All I had to do was change max size to 4096 and the background is not blurry anymore (even when the player zoomed in).



Default

Max Size 4096

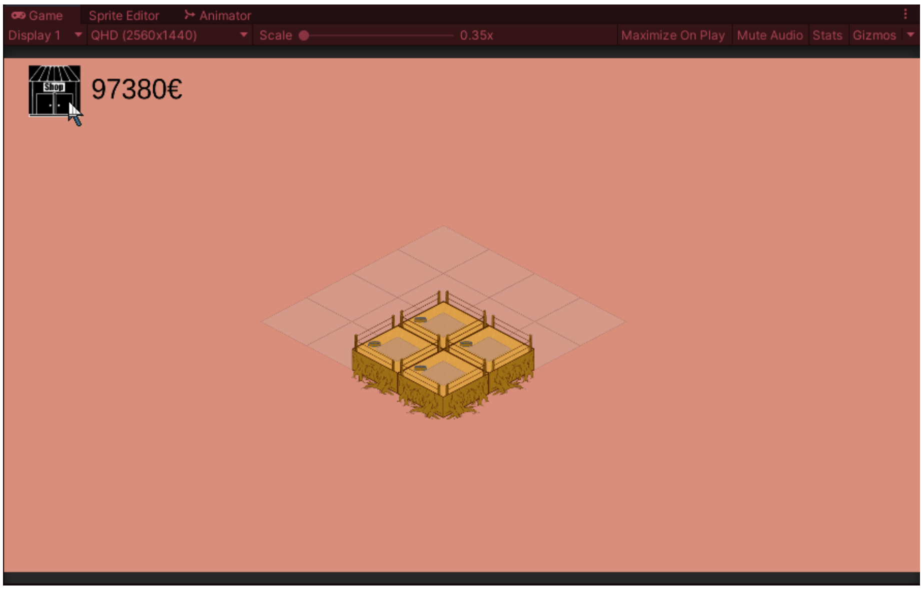
Resize Algorithm

Format

Background_Clean

I admitted we have been rather slacking in our Event System or State Machine. So I rewrite my NPC dialogue into an Event System instead of simply clicking the button. And yet, I think this is a huge improvement in terms of cleaner project and fewer bugs. By having the dialog system into an event system, I managed to try out a feature I always wanted to do, but it always has been a low priority.

I thought, “Hey, we have this cool NPC. Why don't we do more with him to make the game have a bit of personality?” And as a visual novel enthusiast, I test out a quick “interacting mechanic”. And it was a success! I can now make as many colliders and NPC's dialog as I want.



Of course, these texts were not polished. But the mechanic worked. If possible I would like to add more animation and each zone plays a different animation. But this is a low priority at the moment.

__Timur:__ In this version, I mainly focused on splitting the shop. We wanted to divide the monsters and misc things in different tabs. This worked like a charm because unity makes it really easy with the UI system. The reason why we wanted to create tabs is: the small row in the lower area of the shop was very bad for expansion. We could have added a scrollbar to the row too, but this would have looked weird + two scrollbars in one place isn't really UX friendly. So different tabs for the different kinds of shop items is the best way, the main reason being that you could expand them without cluttering the shop UI too much. Also, the player knows directly where he can buy what items.

And I worked on making the credit page, settings page look a bit better. Added some animation to it and changed the colors / alignment of the elements.

9. v.0.09

There wasn't much in this update as we were preparing for the final release. We're mainly focusing on bug fixing, polishing (especially the Menu) for this update as opposed to adding new features.

10. v.1.00

We mainly focus on polishing our games and organizing the Unity project for the final submission. Similar to v.0.09 no new features were added.

6. Communication and project management tools

Main communication tool: Discord

Our main communication tool was Discord. When something arises, or we have a question for another, we simply send each other a quick message in discord.

Our personal rule was to at least talk to each other and discuss what we want to do for next week at least 1 time per week. Since our vision and tasks were very clear, there wasn't any difficulty in dividing our task and what we want to add for the next build.

Each of us are also very open to discussion, feedback and ideas. Hai Lam, for example, would ask for sketch/design feedback and change accordingly. She will also report bugs and test out new features made by Timur. If there are questionable choices, we will discuss with each other if we want to keep this, change it or even remove it from the game.

Project management tool: Codecks and Miro

For our project management tools, we use Codecks to keep an eye on our tasks. Although as mentioned we are very clear what was on our bucket list, codecks were not so overly helpful. However, it was very helpful to see how many ideas we have to drop, as in the beginning we just threw ideas wildly to our codecks. It's very common to scale down the projects due to time and resources, and it was certainly interesting to see old ideas that sadly never come to fruition.

Miro is mainly used by Hai Lam to create some quick mood boards and ideas. Though, these will all be included in the Design Journal.

7. Self-reflection: What could be better?

Timur

As mentioned above in the “What was good? What was bad?” section, there are a couple of things that could have been better from a programming perspective. Afterwards I now learned much through the mistakes I made and for the next project I know how to handle such things was better. Generally speaking, there wasn't much where I would say it “could have been better” because I was very satisfied with how everything turned out and how we handled things. I think besides the things I have mentioned above, more features could have been better for the game, as we had many things in our minds that we didn't get to make because we decided to stick to our scope and finish that before we start “experimental” things. Looking back, it was the right decision after all, but maybe others won't see it that way when playing the game because it seems very simple and short. Those features that aren't added such as a stock market or a weather system where the effects change depending on the weather and season would have rounded up the game way more and possibly that would have been better for the general project instead of trying to polish our “main game loop”.

Also, I had many problems with the shop system and in the end I just wanted it to work and made some pretty “lazy” changes, to make it work. I should have taken more time to make it proper. The players don't see it, but I know how I have made it, even though I also know this wasn't the best way to make it. This leads back to the usual issue I have with many things. I should just take my time with it and do it properly instead of the sloppy way because that’s what it is about for me, to learn how to code as efficiently and best as I can.

Hai Lam

When we came up with this idea, I had a lot of vision of what the game should look like. As this was my first project where I had to care less about coding, I wanted to take this opportunity to experiment with what Unity and I personally can do. 2D animation is always what I want to try my hand on, and with the NPC and the background I think the result was satisfying. At first, I even want to animate each monster stage, make each sprite full color and shaded. But as the weeks go by and more and more work from other courses (especially Theorie) I notice there are a lot of things I won’t be able to do. So I had to scale down my own tasks. The monsters are now flat color and visually identifiable enough for the player.

One of the biggest annoyances for me is a mistake I made with GIT which caused me to backtrack many hours of work. I didn’t know GIT had an upload image. So when I add my 4k background and add animation for the floating rocks, it sadly wouldn’t let me push my commit. So I had to drop everything and first add the background, edit and polish it. After that, I can animate the rocks. However, I made a backup Unity project because I knew all this work of mine will be gone since GIT wouldn’t allow me to push this commit. Being able to reuse most of my animation is a blessing. And since most animations were simple animation, the one I couldn’t retrieve I simply made anew.

I wasn’t sure if a 2k image was enough, so I created a 4K resolution background for this game. And it was a gigantic mistake for me. As mentioned above, the domino effects of this choice made me backtrack a few hours of work I did. This time could have been used for more polishing, but I had to sacrifice them to fix the background and its file. It was a very big lesson for me.

I learn my lesson to scale my vision in time and don’t try to upload something so big as a 4K PSB file.

8. Division of tasks: Who does what?

Timur: Programming

Hai Lam: 2D Arts and Animation

It was very clear where our strong suits were: Timur takes on coding, while Hai Lam takes on 2D arts.

9. Conclusion

Overall it was a very satisfied end product despite a lot of short-coming internally. We learned a lot in this project. What was especially surprising was the size of the scope for the game. Usually, people tend to make the scope too big, causing problems down the development line. We luckily avoided this issue, making our end result very satisfying for the both of us. We both learn a lot from this project, both as individuals but also as a team. There were some shortcomings which could easily have been solved if we thought of it earlier. But those shortcomings were not game breaking or conflicting with our Performance Requirement. If we had more time, we would certainly add more contents such as the stock market, weather system, more tiles and monsters to the game.

---

[1] https://store.steampowered.com/app/588040/WILL_A_Wonderful_World__WILL/

[2] https://wintrmut3.itch.io/maelstromexe