Brin Žvan - brzv@itu.dk

Course code: KGMAGAM1KU

# Chicken Defense Postmortem

## 1. Introduction

Chicken Defense is a game about protecting an ever moving flock of chickens from being stolen by jealous farmers. With action packed gameplay it's a spin on tower defence genre where the player is going to be constantly on the run and constantly moving.

Through the development process I acted as a Game Design Lead while also producing the graphic assets. I was primarily the person that made design documents and tried holding us to the pillars of the game we set up. Levering my previous experience I also programmed, mostly taking care of implementing the assets, juice and sound.
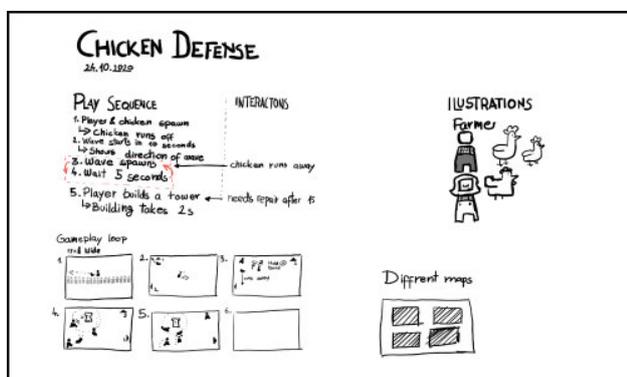


*Fig. 1: An example of a design diagram*

My roles had a "back and forth" connection to the rest of the team. When creating new designs I would bring them up to the team, we would discuss them and based on the discussion I would go back and revise the assets, design documents, animations ... The design documents are worth mentioning as they represented a north-star in our development and directly affected every planning and coding decision made in the team. These were created either as drawn out diagrams for the topics we care about more or were collected in a google document that we intentionally kept as abridged as possible to allow for quick iteration.

## 2. What we did right

At the start of the project we knew that we need to facilitate a world where the chicken would make sense and I think we have successfully done that through the theme we choose to use in the game.

### 2.1 A Sensible Chicken

I am a strong believer that in games things have to make sense. Why can the player do what they are doing? Why are they doing it in the first place? Even if the focus of the game is not HOW the player can build these scarecrows and WHY they are defending the chicken from jealous farmers it has to make sense. This mentality was inspired by Rami's and JW's talk of Sensible Nonsense [1] and Johny Ive's mantra of *"finishing the back of the drawer"* [2].

So to make sure things make sense we created a world where protecting a chicken would be logical, we designed a story of a farmer attending a chicken show that tours the country every day. This allowed us to change the landscape of the levels and make sure that "it made sense". We envisioned this farmer being highly clever, able to modify regular scarecrows into instruments of defense.
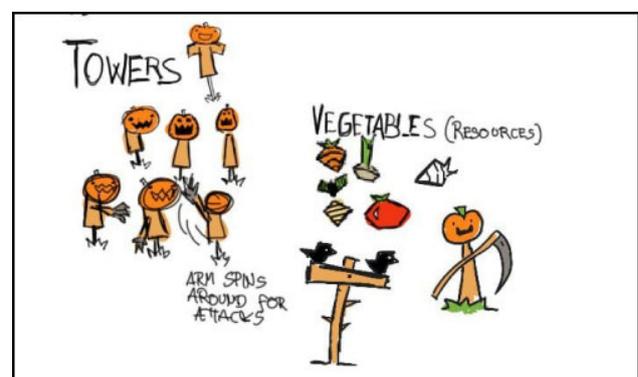


*Fig. 2: Scarecrow concept art*

So the theme was not only tailored to the gameplay but also played into what the game actually is. The game is a small, lunch-break getaway. It is not a grand adventure, it is not something you come back to every day. It is supposed to be something that's fun for a short amount of time because it turns a genre on its head. So having a theme that just screams unapologetic ridiculousness is honest. Being honest with our design is what helps us set the expectation of what a player is getting into. Honest design is good design [3].

This originally served as an internal way to explain our decisions, we decided to sprinkle parts of explanations into the final game. This took form in with flavourful text, presented on a piece of paper between the levels, just enough to spark the imagination of the player, but minute enough to let the player's mind fill in the blanks. Player's reacted positively as the off-the-wall text hinted towards what is happening in the next level and played into reinforcing the comical nature of the game.
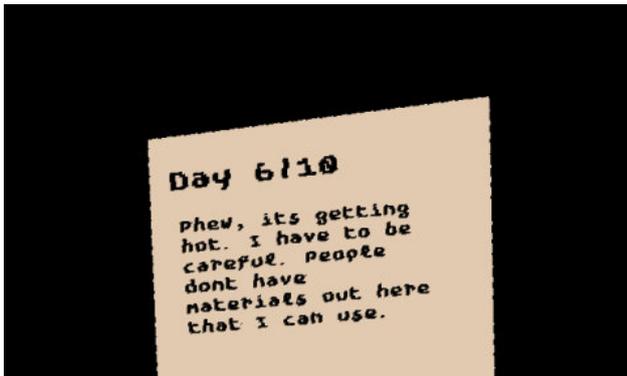
Fig. 3: "Diary" pages that deliver flavour

Lastly, we also needed to make sure that the game would feel as chaotic as we are trying to represent it with the theme. We ensured this through visual silly animations. The characters don't have a walk cycle, they wobble on a sine wave. The chicken's kick up a storm of dust whenever they run from their enemies while scarecrows squash and stretch generously as they attack the envious farmers. Every animation focuses on exaggerating [4] what it's representing, again playing into the overall theme.

# 3. What we did wrong

Looking back at the game I see the biggest issues being in the engine we decided to choose and in our process of tackling game difficulty through the development of the project.

## 3.1 ENGINE
I consider the choice of our engine a design choice and agree with Aral Balkan's idea that *"Design is how it works"* [5]. While it worked for me, I do not think it worked for us as a team. The engine we worked in was Construct 3. It is an event driven engine that is very visual and easy to use with a pick-and-choose approach to coding instead of regular code writing. This approachable nature came back to haunt us as the code is saved in .json files which is not very readable. This became problematic as we ran into issues when merging code and having a hard time reading what changed. Additionally the programmers did not end up enjoy coding in Construct, creating a situation where they worked in a framework they considered unfun - this does not help creating a motivating surrounding.
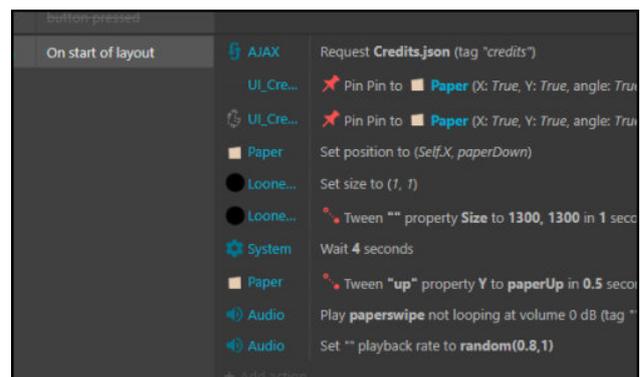
Fig. 4: Construct's event driven programming

## 3.1 DIFFICULTY
Our initial goal was to scale the difficulty by throwing more enemies at the player and giving the players more chickens to take care of. Making the game more chaotic.

I see our game as having 3 pain points that resulted in me marking difficulty as a failure.

### 3.1a Lack of strategic depth
The difficulty of the game increases as new gameplay elements get added. For example, in level 2 the player plays with a different scarecrow than level 1. Level 4 introduces new enemies, while level 5 adds an extra chicken. Later levels focus on introducing new environments such as caging the chicken in player-in-accessible areas in level 10 and removing resource drops in level 7 and 8.

However, we rarely use these elements to create puzzles. Adding them to the gameplay does create difficulty. This difficulty is shallow

and does not last for a long time. If we draw parallels with the world of UX design, we are essentially creating noise [6] that the player needs to scan through to find the right answer.

Users during tests have responded to this noise by trying out new strategies, however, the sad truth of the our level design is that this reaction is a façade. Usually the best strategy is to place as many range towers as possible in the middle of the map, no matter what level you play. This non-changing answer relegates our new elements just down to noise that needs to be scanned through and eliminated out of the equation. I believe that the elements in a game should require the player to reconsider what they are doing and force them into a different playstyle. We don't do this consistently and that is why the game lacks strategic depth.
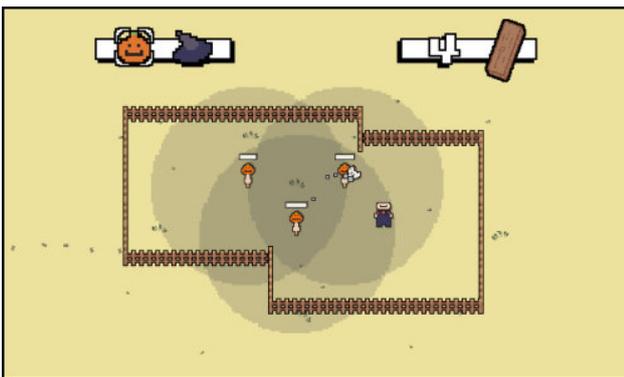


Fig. 5: The anwser to most levels

## 3.1b Resources

Using resources as a way to allow the player to build scarecrows seemed natural, we are making a tower defense game after all. But their purpose was to limit the player, force them to make deliberate choices and with that, create difficulty. However, this caused more issues than it was worth and for a game where we wanted to turn the tower defense genre on its head we were way too stubborn on keeping them in game.

So why were resources bad? They made the levels and chaos much harder to scale. The more enemy farmers the scarecrows defeated the more resources you would get, so in essence throwing more enemy farmers at the player would create more opportunities for a tower to generate resources. Having access to more resources allows for placing more towers making the game easier. So trying to make the game harder, made the game easier. This could be fixed by adjusting tower costs, requiring more enemies to be defeated

using a single tower but this would destroy the opening levels with less enemies et cetera.

This un-intuitiveness created by the resources ties back to the first point. It made designing interesting levels really hard. The intuitive answer to ramp up difficulty was not intuitive anymore and while it did prompt us to create some interesting levels (level 7 & 9), they felt like gimmicks compared to the rest of the game.

The only difficulty the resources created was punishing a player harshly for a misplacement of an early tower. This issue is not uncommon [7] among tower defense games and something we ended up band aid fixing by making towers "recycle" themselves.

Resources had too many ripple effects and they did not serve their purpose, they forced us to band aid fixing and made level design unnecessarily hard. I am sure we could have found a better solution for limiting how many scarecrows you can build.
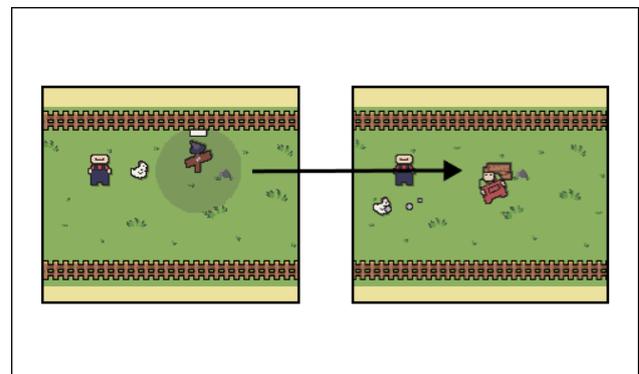


Fig. 6: Scarecrow "recyclying"

## 3.1c Bad (analysis of) data

I think the difficulty problem came to be from the way we analyzed data. As I alluded to in the first point, our testers responded to new elements being introduced by trying out different strategies. This led me to believe that the level designs were achieving what they were supposed to achieve.

The mistake here was not considering how many variables are at play when testing the difficulty of a game, or as Tom Francis puts it - how random difficulty is [8]. When someone played the game by simply using range towers in the middle of the map, we were baffled, unsure how to "fix" this. This issue would not happen had we thought of how we want the levels to be solved more and not just what are the things we want to introduce in a level.

# 4. Takeaways

These takeaways should serve as reminders of what to do in the next projects:

*1. Build a world not just for the player but also for yourself*

*2. A comfortable development environment goes a long way, make tool choices that facilitate this for everyone*

*3. If something is giving you too much trouble, try replacing it instead of fixing it*

# 5. Summary

The finished game has a strong and honest theme that creates appropriate player expectations and makes the game feel logical. The systems work together to create a feeling of confusion and the player is required to care of multiple elements.

Most of the difficulty is manifested in this confusion creating a difficulty curve that exists but it exists for the wrong reasons. The game is far from perfect and the development process has taught us a lot, however I would be lying if I didn't say that I am quite happy with how the game turned out.

# 6. References

(1) Nijman, J.W., & Ismael, R. (2012). Sensible nonsense https://www.youtube.com/watch?v=vk94HoI_tCo

(2) Goldberger, P. (2013). Designing Men https://www.vanityfair.com/news/business/2013/11/jony-ive-marc-newson-design-auction

(3) Good Design principles https://en.wikipedia.org/wiki/Dieter_Rams#%22Good_design%22_principles

(4) Twelve basic principles of animation - https://en.wikipedia.org/wiki/Twelve_basic_principles_of_animation#Exaggeration

(5) Balkan, A. (2012). Design is not veneer https://ar.al/notes/design-is-not-veneer/

(6) Krug, S. Don't make me think Revisited (4th ed.). New Riders

(7) Nijman, J.W. (2020) FFFLOOD postmortem - https://vlambeer.itch.io/ffflood

(8) Francis, T. (2019) Difficulty is random https://www.youtube.com/watch?v=-8lYPAPGo40