A.C.T.I.O.N. GAME DESIGN DOCUMENT

Game concept

A Cautionary Torment Imposed On the Nameless (*A.C.T.I.O.N.*) is an action, top-down game focus on resource recollection and management in an extreme environment.

Game loop

The core game loop consists of the player exploring the map, killing minor enemies, recollecting materials, and crafting traps with them, to put them to use against the area's boss. This cycle repeats multiple times per area, against the same boss.

Game mechanics

The player is able to:

- Walk: slow movement, value 180 pixels / second. Unlike running, it doesn't cause an audio to play, signaling that it triggers no sound events (see boss description).
- Run: fast movement, value 800 pixels / second. It causes a "sound" signal that can alert the area's boss. Running decreases the stamina bar.
- Attack: a fast animation, but an action only effective against minor enemies.
- Craft trap: the avatar instantly consumes the necessary resources to craft a trap.
- Set trap: the avatar stays in place during 1 second, setting a trap on the ground.
- Reduce stress: spend the resource "vegetal ropes", used to craft any trap, to reduce stres by 5%.

Game components

The game can be divided in three main actors:

Avatar

The avatar is an androgynous youngster who recollects materials in order to leave the hazardous environment they were thrown into at the start of their quest. Gameplay-wise, it is characterized by a series of parameters:

- Stamina bar: regulates how long the avatar can run. It depletes as the player runs, and starts building up the stress bar once it goes under 25% of its 100 units of capacity. It decreases at a rate of 40 units per second and is regained at a rate of half its decrease rate, making a total net (if running) of negative 20 units per second.

- Stress bar: regulates the effectiveness with which the player can run. It increases once the stamina bar goes under 25% and the player continues running, at a ratio of half the stamina decrease rate. Once it reaches 100% of its 100 units of capacity, it halves the player's speed, regardless of which state they are in.

Additionally, the player has an inventory from which they can see and use their resources to craft traps, or reduce their stress.

Enemies

The enemies in this game are the environment's wildlife, which can be divided in two: minor enemies, and bosses.

Minor enemies

Minor enemies are non-moving entities who can drop resources, whose only behavior is to do their "attack action" once the player enters their range, then enter a cooldown of 0.5 – 3.0s seconds before returning to their idle. There are three types of enemies, differentiated through their sprite and "attack action" (for reference, all enemies have an idle hitbox of approximately 128 pixels, and their range is calculated from the center):

- Light enemy: at a range of 128 pixels, they launch a beaked tentacle to strike the player, with a 150 milliseconds lingering hitbox.

- Sound enemy: at a range of 768 pixels, they generate a sound event (see the description of bosses) during roughly a second and a half. That event is the same sound as a “chase start” sound, to add tension to triggering a sound enemy.

- Mouse enemy: at a range of 128 pixels, they emit a noxious gas that fills the tile they are in during 0.8 seconds.

Enemies’ dropped resources are based on what we manually input, so we can use enemies as challenges without worrying about how that will affect the game’s resource balance.

Bosses

Bosses are the real juice of the action side of the game. They are entities who constantly go after the player, and once they catch up, cause instantaneous death.

The bosses’ basic behavior is a pathfinding algorithm towards the player, based on their “sense of smell”. Once they have reached the end point of their pathfinding, they will relocate the player, and renew their persecution. During that process, two things might happen:

- Hearing event: in a radius of 640 pixels, if any “sound” is detected, its origin’s position becomes the new objective of the boss, who enters a chase state.

- Sight event: in a radius of 1280 pixels, raytracing is used to see if there is any wall between the boss and the source of a “sight” event. If there is not, the boss enters a chase state towards the origin of the event. A chase initiated by a “sight” event will only change objective by another “sight” event, and always rewrites “sound” and “smell” events.

Every time a chase state triggers, a sound will be played so that the player knows, regardless of whether they can see the enemy or not, that they are in imminent danger.

Once the chase state has been entered, the player might be caught, triggering a reset of the level; the boss might lose sight of the player; or the boss’ attention might go to some other entity. Once their chase reaches the final destination, or is interrupted without the possibility to continue, the boss returns to their basic patrol behavior.

The bosses' walking speed is 300 pixels per second, and their running speed is 1000 pixels per second. Since those values are bigger than the player's, and the bosses are always moving towards the player's position, it is almost a guarantee that the player will be forced to interact with them to progress.

Traps

Traps are tiles that have been manipulated to cause a certain effect when triggered, which can be caused by the player or any boss stepping onto them.

Since their initial state uses the same sprite for all types, the player won't know what trap each is until it is triggered, or unless they were the ones to plant it.

There are three types of traps:

- Light trap: its effects last 6s. It requires 1 vegetal rope and 1 incandescent bulb.

- Sound trap: its effects last 2s. It requires 1 vegetal rope and 1 pressurized gland.

- Movement trap: it requires 2 vegetal ropes and 1 unconscious mouse. By triggering it, a mouse spawns and starts running towards the opposite direction at which the area boss is, emitting vision events. Once it hits a wall, the mouse disappears.

Depending on who does so, the two first traps have distinct effects:

- Player-trigger case:

    - Light trap: the screen goes totally white through a 0.5s transition. The player can still move, but will hardly be able to tell where are they going.

    - Sound trap: while the screen shakes, the player isn't impeded in any way.

- Boss-trigger case:

    - Light trap: an animation appears at the traps position, causing the boss to be unable to receive any sense-based events. Their

pathfinding, however, is not deleted, so they will continue moving to their current objective.

- Sound trap: the boss is shaken by the sound, and is kept immobile.

- Movement trap: the boss will prioritize “sight” events from movement traps over player “sight” events.

Progression

The player begins the game in a safe zone that connects the final boss area with the rest of the world. Since going into the boss arena will most likely result in a quick death, the player will end up in the regular levels; if they do go to the boss, it becomes a memorable return point, and if they don’t, they will have to realize it is their final destination, since it’s the only deviation available at the end of the level gauntlet.

The regular levels are a linear sequence of levels, progressively increasing in difficulty, pitting the player against the harsh environment, presenting the mechanics and dynamics of the game, forcing the use of resources and entity interaction.

The transition between each level will trigger a save game, so the player can retry each challenge relatively punishment-free; while this might be a bit against the essence of the experience, it was deemed the correct compromise between what the game could be, and what the development state could afford.

Level design

Considering the simplistic gameplay, and the lack of diversity in enemy behavior, the level design is the element doing most of the heavy-lifting to regulate difficulty.

There are five elements that were worth paying notice to, after the internal playtests were carried out, as mechanisms to control difficulty:

- Corners: due to the pathfinding’s traversal, the enemy loses a bit of distance relative to the player when turning corner, even with their superior speed. Thus, the more corners, the more opportunities a payer will have to skillfully maneuver away of their persecutors.

- Width: if being reached by the boss is an inevitability, the player will have to use traps. To trigger them, they will have to make the bosses stand on them. However, they always follow behind the avatar, and since traps trigger when the avatar step on them, too, the player is forced to make a turn to trick the boss. In summary, the wider a space, the easier for the player to make that turn, although the more distance they will have to cover back to align the enemies with the trap.

- Openness: the vision of bosses is limited by walls. The more walls, then, the safer it is to move, as the avatar will not trigger the chase behavior of enemies, as long as they don't run.

- Trap opportunities: spaces where adding a trap is especially useful, where it makes sense for the player to invest resources. Based on those, we will define the amount and types of resource drops, so players have an adequate amount of materials to make mistakes and compensate any skill issue, as long as they play smartly (after all, we want them to succeed at the cognitive challenges over the mechanic execution).

- Reaction distance: having implemented a fog of war effect to increase tension by hiding parts of the screen, any distances outside that scope is non-reachable information. This can be balanced through audio design, but it still is a powerful tool to makes situations more unnerving.

With all those elements in mind, and considering that we will not make a linear game, the design pipeline will be the following:

1. Establish a difficulty level in relation to other areas

2. Pick a skill theme (similar to CCST's)

3. Define a series of interesting challenges revolving that theme; include minor enemies if necessary

4. Distribute them throughout a series of possible paths, prioritizing increasing and equally distributed difficulty in direct relation to level progression, unless stated otherwise

5. Adapt level to ensure that it remains challenging both during the first traversal, and as the player returns to go back to the lobby

6. Model the geometry of the level to connect challenges and regulate difficulty following the previously listed factors

7. Define the drops of each enemy so that creating traps is neither too cheap nor too punishing for careful players

Once all levels are designed, and the amount of rewards in each level have been computed, we will design the final map's challenges based on the assumption that the player completed the easiest half of the game optimally. To compensate the deviation of having played more than that, or harder parts of the game, we will add additional loot to each level as we deem necessary.

The challenge distribution and design will be done with the following metrics in mind:

- Player walking speed = 1.4 tiles per second (TpS)

- Player running speed = 6.25 TpS

- Enemy walking speed = 2.34 TpS

- Enemy running speed = 7.81 TpS

- Distance before player starts building stress (if stamina bar is full): 25 tiles

Note that we can afford high-difficulty challenges, and in fact it will be done purposefully at least a few times, because traps essentially trivialize every boss encounter, as it was noted in the internal playtesting.

Under this premise, forcing their use communicates to the player that their skill isn't as important as how adequately they distribute their resources, and it paints the illusion of scarcity. We call it an illusion, because the resources are balanced so that playing with an understanding of what the game is asking will always leave the player with a slight excess of resources.

However, through this perceptive trick, we can create a self-regulating system for gameplay difficulty that the player will, nonetheless, not consider as their default play due to its scarcity.

Lobby

The Level 0. A simple room where the player first gains control of the character. In it, there are two holes: one leads to the boss room; the other leads to Level 1, and the subsequent rest of the game's content. Additionally, the lobby will have 5 mouse enemies on its lower section.

This room fulfills multiple key functions. The first one is to let the player experience their moveset without external pressure. Secondly, it teaches the player the concept of the "hole" tile being the entrance and exit of the level. Finally, the enemies arranged as the "Q" letter serve three purposes:

- Give a safe space to practice combat
- Expose the player to the resource drop mechanics: the one closest to the spawning point drops an unconscious mouse and vegetal rope, to incentivize further interaction; the ones adjacent drop nothing, to subvert the expectations, focusing on the fact that drops are not consistent (or at least, not predictable by the player); and the other ones will drop vegetal ropes, to end the beat on a positive note.
- The Q visual pattern, while not very visually clear, is intended to remind the player to check their inventory (which opens with the "Q" key), where they will be able to see their items, and if they so wish, to reduce their stress with the newly obtained materials.

Starting the game with 50% stress not only lets the player put that mechanic into practice, but it also complicates that high-skill players can finish the game by going straight into the final level. It also serves as the main incentive for the player to explore.

Level 1

This is a big map, with very few enemies, very isolated between them, and in easily traversable areas. It's a map to get the player some free resources, introduce the rest of enemies and the first possibly lethal encounter with a boss, albeit in a space

that is very beneficial to the player's movement. So much so that it is the only map without fog of war, thus having the longest time to react to their approach.

The exit is positioned in a relatively easy to find location so that it will lure players into progressing without recollecting all the hidden resources. This is intentional so they don't get too many freebies without actually going out of their way to collect them; alternatively, it provides easy access to resources once they return from all the harder levels, becoming a positive surprise that encourages the player to face the final boss.

Additionally, the distance between the exit and the boss is relatively short, so returning to the lobby will force the player to use resources up, or barely escape the chase, an adequate climax for the last level section of the game.

Level 2

Finally starting with the juicier levels, Level 2 is centered around the skill theme of traps. Right at the beginning, the player will be faced by the level's boss, who will trigger a sound trap and become paralyzed, and example of mechanic "show, don't tell".

The concept here is that they traverse the level while making the boss fall into a bunch of traps; if the player gained enough advantage, they will have plenty of time to deal with the enemies at the end of the level, and maybe even get some resources from them. The reverse level is the same concept, but with less available traps, and with the boss coming towards the player, instead of after them.

A trend that those levels will tend towards is having an optional challenge around the middle part, so the player can't abuse the entrance / exits, making the challenge a step higher in difficulty to treat skilled players. Level 2 is not an exception, presenting the player with the possibility of taking the long path to get the kill on a mouse enemy, or risking stepping on a trap to leave towards the exit faster.

Level 3

Level 3 is centered around sound enemies, and how sound events influence boss behavior. We specially aim at making it clear that vision events have a priority over hearing events, but that those are prioritized over past vision events; this is specially clear in the hallway at the start of the level.

There are a series of resources at the end of the level positioned to bait the player to move away from the exit, into a hidden sound enemy, a situation that can elevate the climax of the area. Incidentally, realizing how to deal with those enemies when going back to the lobby, by using the exit close by, reverses that climax into a distinct, more advanced challenge.

This level also has a mid-level optional challenge that reinforces how the boss' pathfinding works, and rewards understanding it.

Level 4

Level 4 is a chase sequence that teaches the player to use the enemies' pushback on their favor, to move at higher speed. It was vital to introduce a level that taught this specific non-intuitive dynamic, because it would become part of future challenges, and without guaranteeing that knowledge, a part of the design's depth regarding risk – reward would be most likely lost.

Since this is a mandatory section, many rewards are made easily accessible to compensate the player for the stress their character will inevitably build once completed. This should also be the point by which they have learnt how to reduce stress; if not, being forced to build it up should be a further incentive to put that mechanic to use.

The level's reverse is the opposite concept; since the boss is far, and thus won't receive vision nor hearing events, the player's best strategy is to carefully (but not slowly) tread the path towards the exit without building stress, and maybe even farming enemies to get resources before the boss catches up to them.

Level 5

The fifth level is the first instance of multiple bosses chasing the player at the same time. Returning to the original vision of the game, it presents a more open ended structure that divides the level traversal in two phases: exploration and planning.

By design, the level has two major paths, with their variations, which are essentially the riskier (straight to the left) and the safer path (a more roundabout one, by walking up).

Since it is not mandatory for the player to complete all levels, there is the possibility that they ignore the final levels. If that were to happen, returning to the lobby after

completing the fifth level serves to teach the player the usefulness of traps in open environments; if they had reach the end of the gauntlet, it reinforces that teaching.

Level 6

The sixth level is a very simple test: has the player learnt to spend their resources instead of fully relying on using the environment to their advantage? Or at the very least, if they have understood what they can potentially do with traps. If yes, it shouldn't be complicated for them to complete the level.

While the reverse of the level is the same, because it is symmetrical, it is also an opportunity to realize that no traps are actually needed if the player can use high-level movement strategies, thus saving on resources for the final level.

Level 6.1

Level 6.1 is a trivial level difficulty-wise, that just has a bunch of static enemies for resource acquisition, as a reward for completing level 6.

Level 6.2

For Level 6.2, we are iterating the concept of Level 6, and taking it a step further. By incrementing the boss count, the player will have to not only spend more resources (or not, depending on how keen their movement is), but also factor positioning and boss behavior after the first one is temporarily out of the picture.

Level 6.3

Level 6.3 follows the same logic as level 6.1, except this one doesn't have an exit, thus signaling to the player that they should go back.

Final level

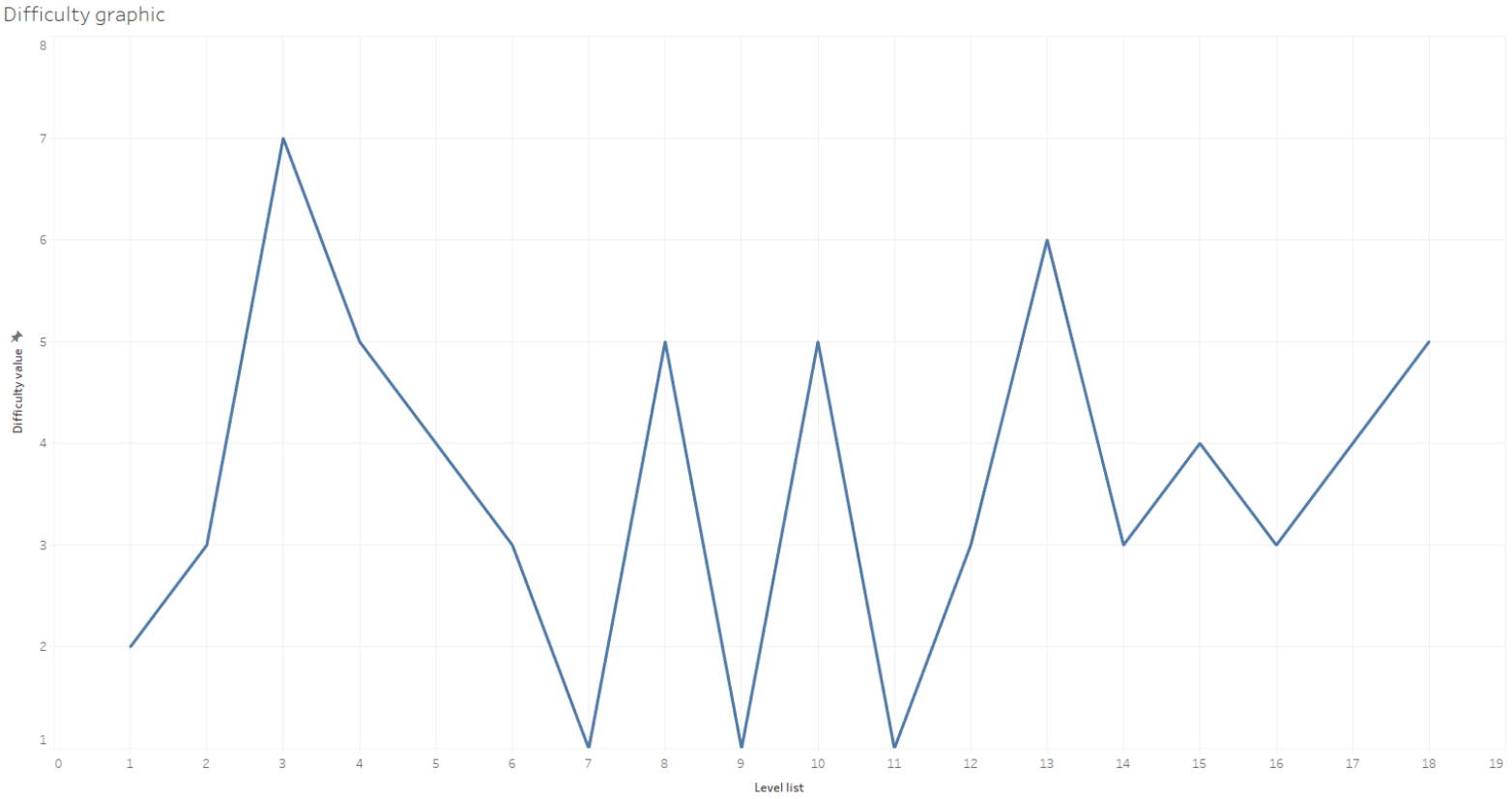
The final level replicates the open structure of Level 5, but adds two more bosses for a total of four. Once again, it traces two distinct paths, one to the left, and one to the right. Both demand resources to traverse the level safely, with the left path being more movement-driven, while the right one is easier to traverse by applying what the level gauntlet conveyed with its level design.

Level design data

To summarize the gameplay's experience, we can graph a series of properties. Disclaimer: level 6.1, 6.2 and 6.3 will be considered as 7, 8 and 9; the return of level 6.2 will be 10, from level 5 will be 13, and so on, making the final level the 18th.

Difficulty

We compute difficulty as a mix of each level's challenge, which locally considers distance to the boss, distance to escape the boss' chase, effective trap placement spots and enemy positioning.

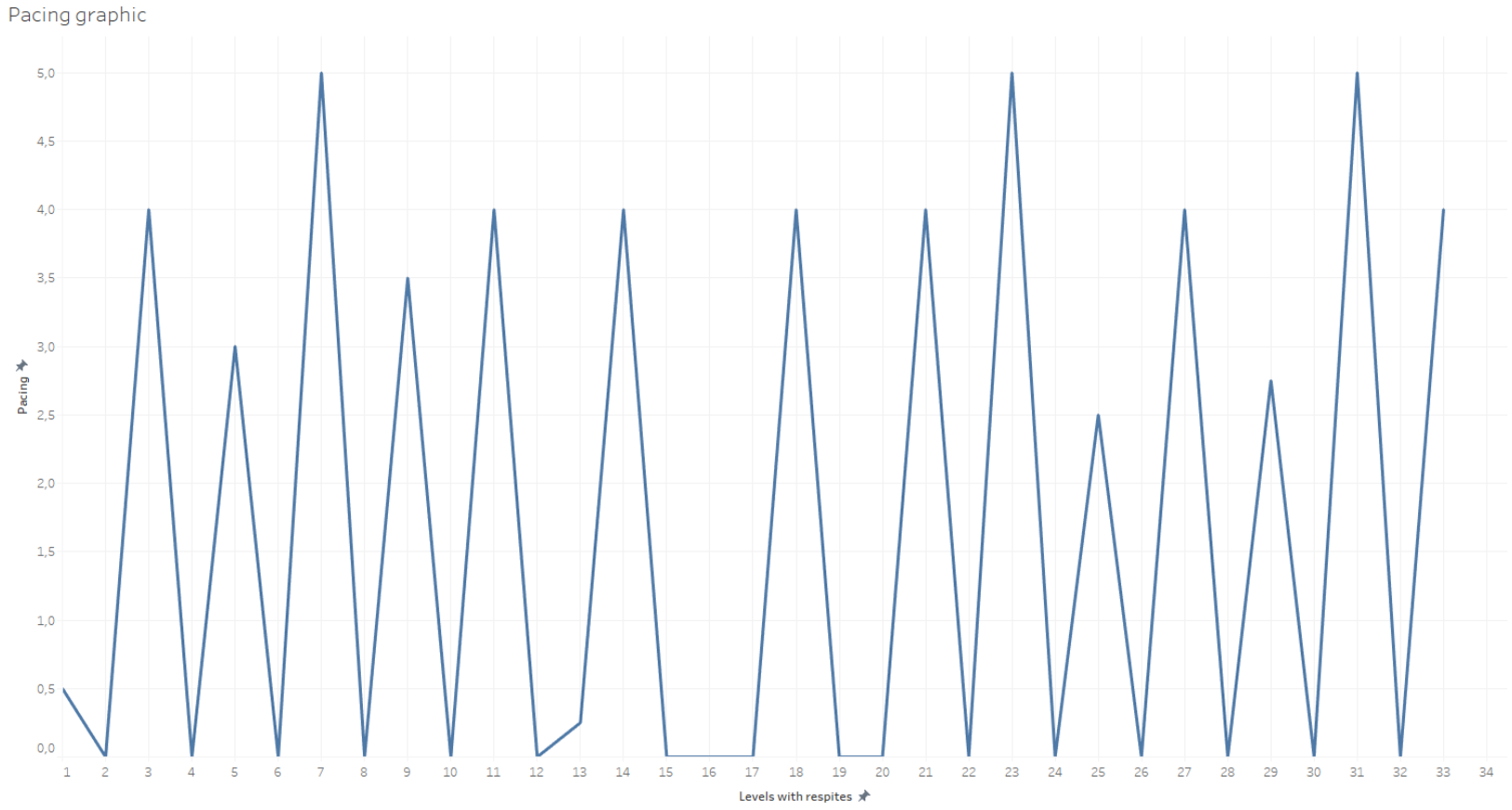


Pacing

The pacing curve is defined by the following formula:

pacing value = (–distance between challenges / nº of challenges) + [nº of challenges] ∑ [x = 0] (challenge(x)'s distance traversed / (1 + opportunities to stop))

To more adequately reflect the pacing of the experience, we will include the respites between levels, so instead of 18 entrances in the X axis, we will have 33.



Resource distribution

Regarding the resource distribution, the corresponding calculations can be found in the document "Resource distribution.xlsx".

The procedure followed has been to account for every challenge that was considered difficult to the average player, enough to cause a situation where the stress bar gets filled up. We noted how much stress it would build considering

distance and previously noted player and boss speed, giving a bit of alleyway to not demand perfect execution.

This was considered per level, so we could distribute the resources not only holistically, but give them out as the player needed them, to reinforce the tension of the game, and avoid premature consumption.

Additionally, we will also compute spots where we deem it “necessary”, even if it’s not, to use traps, which will be converted into resources, and added to the count.

Any stress derived from mistakes outside of the considered challenges means the player is losing on future benefits, which they can make up by better performing on other challenges. This is intended, and the line of performance we consider adequate to move the player towards the designed experience, in turn, how much we expect them to understand of the game before completing it.

Discarded content

The following is a list of content that was discarded during the second iteration, and the justification for it:

- Smell-regulating item: this was conceptualized as a way to regulate pacing, but since enemy speed > player speed, there will already be enough pressure to discourage excessively slow or conservative gameplay.

- Projectile weapons: they were mentioned as a mix of increasing the skill requirement to use traps, and to diversify the types of traps available, but the later won’t be implemented due to time constrains, and the prior can be done less cumbersomely through a jumping action.

- Enemy rage meter: a bar that would invalidate the effects of traps after a boss had triggered multiple of those. Due to their deadly nature, and how we visualized player interaction in the game, this would be too niche an interaction to spend time on it.

- Hide mechanic: this mechanic was discarded because it was very marginally useful: the boss’ “sense of smell” would cause the boss to find the player after a light trap is triggered, unless there was another event. The mouse trap already took the focus of the boss, even if the player didn’t hide. The sound trap paralyzes, and when that happens, the player wants to put distance, not stay there to get killed by smell event. The only situation it

would be decent in would be if there was a sound enemy near where a boss triggered a light trap, which is really specific.

- Trap effectivity probability (inversely proportional to stress): halving the player speed was chosen over the original “trap effectivity probability” because it was easier to silently convey, rather than what would seem like a random malfunctioning of a core mechanic. Not only this effect was easier to grasp, but also fulfilled essentially the same purpose as trap effectiveness, which was to force the player to play safely, and penalize them otherwise. Additionally, the decision that was ultimately taken meant not adding randomness to player decisions, which supports the strategy-based gameplay.