

# KIT-NAP

## INTRODUCTION (1): Pablo

For this assignment we had to make a videogame. For most of us, it is going to be our first videogame and we would have to make a videogame from scratch and learn as we went on.

The first decision we had to make was: What is it going to be about? What game could we create that would be fun to play? As we developed different ideas, we finally sat on the concept of a Roguelike game. In a roguelike game, you normally have to traverse a dungeon full of monsters and defeat a boss in order to beat it, but with a handicap: if you die, at any point in the run, you start over from scratch and the layout of the dungeon has changed, so you have to use your previous knowledge to adapt to the new environment. It is a very challenging experience, but also a really fun one and due to its repeating loop, it is also very given to experimentation and ingenuity.

For that endeavour, we have assigned roles to each of us:

- Sandra Alegría is going to be the Enemy and Plan designer.
- Matías Araya is the one in charge of the Final Boss planning and development.
- Oscar Castro and Javier Mier are going to be the Map Design and construction Logistics head developers.
- Pablo González is the Item Balance and Player Gameplay programmer.

## WHAT THE GAME IS ABOUT (2): Pablo y Oscar

STORY (Oscar):

Kit-Nap is a 2D, top-down-view, roguelike game where the player has to traverse a series of dungeons, defeat enemies, find new weapons and uncover a secret conspiracy to save his cat from certain doom.

In the story, the protagonist is a man that wakes up and finds his cat missing and the door wide open, so he goes out to search for it. After making his way through the forest, he finds the cat's collar next to car tracks and a vehicle off in the distance and decides to follow them to the city. While exploring he finds clues of his cat entering the tallest building in the city and decides to go there to investigate. Traversing the building basement, our protagonist finds a passage to the final level, the sewers, and after making his way through hordes of enemies, he finds a door to a secret laboratory, where a mad scientist has kidnapped not only his cat, but also his wife, his daughter, and his cat, and it is his duty to defeat him to save them all.

MECHANICS (Pablo):

Gameplay wise, the player would be able to move in the 4 directions of space, through a zenithal perspective 2D world, similar to other titles like *The Binding of Isaac*, *the Legend of Zelda* or *Enter the Gungeon*.

The player would also be able to perform a somersault or hop that will dash him forward and make him temporally invulnerable, during which he won't be able to attack.

The goal of the game is to reach the end of the journey and defeat the final boss in one go, without dying. However, if the player dies, he will wake up in his cabin and the previous attempt would be a dream. Since it was a dream, the image the protagonist had of the environment was unreal and distorted, and when he goes out to find his cat again, he would find out everything has changed. This way, the layout of the "dungeon" would be randomized every run.

Through the 4 levels, the player will encounter enemies that grow stronger as he progresses. To face them, he would rely on the weapons and *game modifiers* he finds on his journey.

The player attacks with the weapon he has equipped, which can be ranged or melee. If the player doesn't have any, he would punch. Like enemies, weapons get more powerful the further into the dungeon.

The player can also find game modifiers hidden in chests or in the dungeon shop. Some game modifiers will be positive, buffing the player and reducing the difficulty (more damage dealt, fewer enemies spawning...) and some will be negative, making the experience more challenging (stronger enemies, player is poisoned...). Since the effect of the modifier is determined randomly when picked up.

Enemies drop coins or hearts on kill. The player can use the coins in the dungeon shop to purchase temporary upgrades and weapons, or save them for the outer shop, located in his starting cabin, to buy permanent, more expensive, ones.

## **GAME PRESENTATION (2):** Javi y Matías

In this part we'll be talking mainly about the visual style that our game will have.

Most of the assets we've gotten are from Kenney Game Assets, while the rest we needed to complete the game were public domain.

Our game will be based on pixel art sprites, following a 16 by 16 structure, as most of the assets we found had that style, we found it simpler to work with and, in case any of them needed some rework, it would be easier for us.

(Referring to the images in the first slide) Here we have some sprites for the possible main character and in this other image we have a collection of cat sprites that we will use as a base of inspiration for our own asset, because we couldn't find any online that we really liked.

(Referring to the images in the second slide) In this slide we have some examples to show how the levels our colleagues talked about earlier would look like. This would be the style of the forest levels (referring to the image on the left) and this would be the style for the city levels.

For some of the game elements, we haven't reached a conclusive decision as to what they'll look like but rest assured that they'll follow the same style as the rest of the game.

As a final note, we would like to clarify that the assets shown here may not be the assets the final version of our game will be using, as some of them are still subject to change.

## **TOOLS AND DEVELOPEMENT PLAN (1):** Sandra

For the game making process, we are going to use *Unity Game Engine* as we found it is the easiest to use when learning to make your first game, mainly because of all the support and tutorials available.

Since the visual style of Kit-Nap is 2D pixel art, we are going to use pixel art specialized image editing software such as Aseprite, which also provides support for animations. We are still deciding on a music software. For the visual effects of the game, we are planning on using Unity's built-in particle system.

For managing version control and coordinating the development, we are going to use GitHub repositories, and for making the Development Plan and managing deadlines, we are using ProjectLibre.

Finally, for the Game Development Plan, we have split the process in 2 phases:

First, the programming phase, during which we will make the systems and the behind-the-scenes tweaks necessary to make a very ugly but functional version of the game.

During this phase, we will have to code the player and final boss, as well as the templates for adding enemies, items and *game modifiers* and the level creating logic (referring to the table). The first person that finishes its part, will begin working on the interfaces.

When all the programming tasks are done, excepting the final boss and the interface design, we will move to the Design phase, during which we will incorporate the animations, music, visual and sound effects into the game.

In the event that an unexpected problem occurs, we are giving ourselves a great margin on the deadlines, so that we have enough time to fix it without affecting the development.

Meanwhile, we are using GitHub's remote repository system to keep track of the versions of the project in the cloud and use them as backup in case information is lost in our computers.

In addition, we will only be using Unity's 2022.3.10f1 install version to account for problems regarding changes in the product behaviour.