

As responsible of the code, my work has consisted to programming a lot of types of mechanics and algorithms for the game. Now I' m gonna explain a part of my work.

Entities System:

I have implemented a basic system to create and use entities. This system has evolved during the project by the types of entities created by the different members of the team.

I designed different types of interaction between entities, so the player could interact with them.

- **Triggers:** Points in the map created by realize one action.

- Dialog Trigger:



- Door:



- Exit Portal:

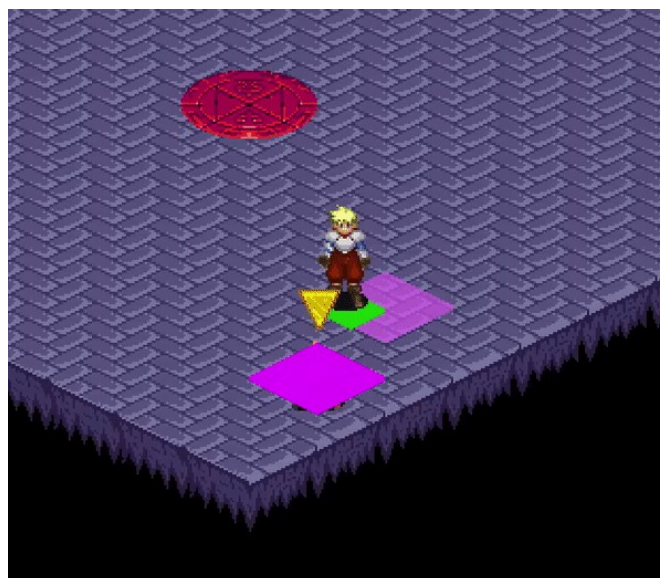
- LobbyPortal:

- Portal:

- NoWalkableTrigger:



- SaveTrigger:



- **Buff Manager:**

When the character attacks or is attacked by other entities, the buff manager is the responsible of calculating the damage. Buff use the objects for calculate damage.

Buff apply damage using two different methods, damage dealt/recived.



Create and Spawn Objects:

I have done this task with QA manager. I have implemented the creation of the item and the interaction with these. We separated the items in:

- consumables
 - potions:



- phoenix tail:



- equipables

- sword:



- bow:



- 

- 

- 

- 

-
- A screenshot from the video game Final Fantasy IX showing the Flower Boss battle. The boss is a large, multi-headed plant-like creature with many small, red, flower-like heads. It is surrounded by a large number of smaller, red, flower-like enemies. The battle takes place on a dark, checkered floor. In the top left corner, the player's status is shown, including a portrait of Zidane, a level indicator (LVL 1), and a health bar (100). In the top right corner, the boss's name "FLOWER BOSS" is displayed above its health bar. The bottom of the screen shows the player's command menu with various icons for actions like attack, magic, and items. The overall scene is set in a dark, dungeon-like environment with some green plants and bones scattered on the floor.

- bossArrow:
- emitterArows:



Sprite Ordering:

I adapted the sprite ordering created by Christian Martinez <https://github.com/christt105>

camera culling with Quad Tree:

I have created optimized draw system using Quad Tree.



Load/save Game:

I have implemented a Save, Load and Game Loop.

**Transision Manager:**

I adapted the transition manager created by Marc Guillen <https://github.com/Marcgs96>.