The game is a simple 2D top-down adventure game, where the player has to collect items, combine them to get more items and finally solve the challenges presented to him/her.

It has:

- A main character, that can move and interact with the world.
- An NPC, that serves as a tutorial, giving items to the player and also consuming from him.
- An inventory system with features like adding, removing, moving, swapping, combining and using items.
    - It also saves and loads which items the player has, on which slot.
    - It's presented on a clear and intuitive User Interface, with a tooltip appearing on mouse hover.
- Interactable objects that give items to the player.

The player's actions (movement, idling, interacting) are made using a design pattern called *State*, in which there is a Finite State Machine structure, making sure that the player will always be exactly in only one state at a time. This makes the code easier to scale and maintain, as the quantity of mechanics grow, because each mechanic is isolated on its own state file, and the transitions between states are defined in a very clear manner.

I also used the design pattern *Singleton*, to create the Inventory and the Tooltip systems.

My thought process followed a Game Jam process: I created the design first, aiming at a small scope. Then, I implemented the core features (player actions, inventory, items) and then other important features like the NPC, Tiles, challenges and the tutorial.

I liked my overall performance, but I feel that it could have more good-to-have features, like more challenges, sounds, VFX and a better storyline. But with the little time available, I was happy to have been able to deliver a solid experience, containing the essential features that were requested plus some other cool features.