1. Intro

At first I had no Idea what I wanted to make, so I started with the scenario, to see if I would come up with something in the process. I chose 2D because I thought it would be faster to build the scenario with tilemaps. So I came across this Tiny RPC asset in the unity store and had the idea of building a house area, just to showcase the mechanics that had been asked for.

By the end of it I had the Idea of creating a tiny narrative, where the player wants to travel through the woods, but couldn't because it's too dense, and you had to ask for help for it.

2. Gameplay

After finishing it (and getting some sleep), I went for what was easier for me, which was character movement and animations.

For the movement, I always try to make it as responsive as it can be, since it's the base mechanic, the player should feel from the start that the character responds exactly to what they input. Used RigidBody2D.AddForce() to give the player movement, while clamping the linearVelocity to enforce the topSpeed.

The animations were really standard, I used an animator with an auxiliary script (AnimatorController) to set the parameters.

And at last the interactions, which are based on a trigger the player, and the interactables carry. There are 2 different interactions: pickup items and NPC dialogues.

3. Inventory Structure and UI Design

The idea here was to build an inventory script which would store and manage the items, and a separate script to manage the UI, while keeping them updated to each other. So after every inventory interaction, there would be an equivalent one to the UI, and vice-versa.

I spent more time than I wanted to find suitable UI assets, but I think that the ones I got are fine. Since every asset was pixel art so far, I needed to use pixelLab to pixelize some assets to fit the theme. All of the assets were obtained by unity asset store, itch.io free assets or, free to use fonts.

The inventory layout was designed to be simple, easy to use and fast to implement, while keeping the chill/medieval aesthetic, 8 slots and 2 text boxes for the item name and item description respectively.

To drag/drop/inspect/use items through the UI, PointerEventHandlers were used.

4. Save and Load System

Final but not least, the save/load system. It was implemented to save only the inventory items, and it keeps their original slot in it.

After every inventory interaction, it waits for a short delay (0.3s), so it won't save multiple times after successive changes, and loads every time the player boots the game.

A save file is automatically created when you first enter the game.

5. Final thoughts & Extras

I think it is mechanically fine, all of what was asked was implemented. Visually I think it is fitting, maybe if I had more time I would have messed around more with the scenario, and definitely put some music and sound.

Overall, I'm happy with it.

There are some hidden dialogues which I thought it would be funny to include, cause after all, that's what games are all about.