First of, I started thinking about what I should do to accomplish the task:

- Simple character controller and camera follow;
- To find some tileset assets, and create a simple scene. Fortunately, I've found a free package at unity assets store (Everything inside "/Cainos" folder was from this package);
- Create a simple variation of outfits. I just draw a grey hat and shirt, and recolored it to have some variations. After this, I duplicated their Animations and Animator Controller to set everything;

From now, I had everything needed to start creating the system that would handle the inventory, storing items and changing the player's outfit. I created a Scriptable object to use as base to the objects list, and them 2 structs (InventorySystem.cs), that would be the lists of items the player has.

After that, I started creating an InventoryManager that would handle some basics inventory operations, like adding, or subtracting items (I did not finish the selling system, but almost everything is done), and an Item specific manager, to handle switching runtime animator controllers from the item slots (hats and shirts used in this gametest). This way I could handle creating over than 60000 items inside player's inventory without compromising game's performance, since it's always the same object instantiated, and what changes is it's animator controller. However, I recognize that it could (and must) be improved to achieve the best performance outcome. Since it clears and relaod the whole player's inventory everytime the dresser is opened. Better solutions could be determining specific moments where the player's inventory can be altered (like shoping/selling) and only reloading the inventory after conclusion (in fact, I could do it now, but I really need to sleep a little).

And last, but not less important (I guess), I created the canvas, for the shop, inventory and player controls (to make your playtest easier). The shop (again) could and must be done script-automated, searching for the items on HatList and ShirtList Scriptable Objects, but I was running out of time, and maybe risc some troubles right now could compromise the entire game. This ways I did (by setting manually item by item on shop) works fine for little amount of items.

The class KeyboradDetector is a class I'm used to do and should be used to manage the player Inputs and should be able to let the player choose its own keybindings, with proper UI settle. Due to time, Ijust left it unfinished (but can be easily referred and used by other classes).

To conclude, first of all, I woul like to thank you to invite me to participate in this test. I learned some new things and now I'm feeling great (and really tired) for this. I think I did a good job solving a comom problem that just having a list of game objects and instantiating/destroying them is not good, and to be honest, I achieved a better result than what I expected (about the game performance). I hope I reached your expectations.