MyPocketWorld - DevJournal

This document talks about the daily increments on the 7-day process of creating the prototype ‘MyPocketWorld’. It includes a summary of the project, its daily increments, what could have been improved and conclusion.

This is a very proof of concept project, and was focused on structuring a scalable project, still reaching the requirements.

Summary

The project includes a customization and shop system, which can be accessed through a session at Host/Client paradigms by using Unity NGO (Netcode for Game Objects). Players can start the game and immediately host a session signing in anonymously or join one with a provided code.

Once joined a session, players have very basic position and rotation variables synced during the session, and also their currently wearing equipment. Players can open store and customization panels by pressing the prompted keys on the screen (for the shop, it requires getting close to the merchant).

At the shop, players can buy/sell items by dragging and dropping them along inventory grids, making the transaction automatically if money allows.

At the customization panel, players can choose different categories to filter all items, and equip them individually. When done, they can hit the ‘Save Changes’ button, which will save everything and sync the visuals over client and host.

Daily Progress

Day 0 (night only)

The first day started alright at night, where I read all the requirements for the project. After that, I downloaded the required Unity version and set the basic foundation (Unity project and Git Repository for version control). Also downloaded some necessary or handy packages, like the multiplayer/netcode ones, the provided asset package, DoTween and Odin.
Took the rest of the day to elaborate an action plan on how to approach the task each day.

Day 1 (Friday, Full day)
Basic World Setup, Inventory System

Created basic world setup and decided to go for a component system, aiming for scalability. Added a basic movement component + player component, so all entities could still move if desired. Then moved to split the base component parameters from scriptable objects to each Entity's own data. This way, component scriptable objects act as blueprints which we can attach to what we need on each entity, where its data will belong to each instance. ensures we can have both npcs and multiple players handled separately.

Then created the first instance of 'inventory'. It's not a literal inventory like a backpack, but players can use the basic UI to drag and drop the data instances of what equipment they have. Splitting Data and Parameters (just like I did for components) allow instances of items having different values, like color for example.

Day 2 (Saturday, Full day)
Customization + UI

Created the first UI for customization. The UI also required accessing player's equipment data, to display the different options for customization. This was defined as another component, entityEquipment. Also created the base ui panel and made it display the player's model preview with a renderTexture and custom camera just for that.

Day 3 (Sunday, Full day)
Customization + Shop System
Continued on Customization, extending EntityEquipment, with a dictionary with item sockets and equipment. Also created the logic for updating the character model through the component, finally connected to the ui, which now contains buttons for updating the model preview visuals and the 'Save Changes' button.

Day 4 (Monday, night only)
Customization + Color
Added support for loading all player items, filtering by category, ui integration, and revamped item data so it can support customizing color as well in the future (was still in progress).

Day 5 (Tuesday, night only)
Multiplayer Structure: Reorganized the whole scene to accommodate the session manager, along the updated network objects. Created the Lobby with a simple UI to manage sessions signing and implemented a simple movement, connecting to the existing components.Finally, finished the support for syncing the player equipables, whose foundations were already started at the component setup at day 1.

Day 6 (Wednesday,)
Shop Integration: Setup the entire Shop UI, along drag and drop functions and management logic to load merchant and player data, process transactions, and validate when they are not possible.

Day 7 (Thursday, night only)
Bug Fixing, Polish, Organize and Wrap Documentation: Fixed minor unexpected behaviors and edge cases, which were related mostly to UI, and avoiding host/client confusions. Organized assets and reviewed some UI feedback. Finally finished documenting the whole process, which you are reading right now!

Potential improvements:
As I was still working full time,time was shorter than I anticipated. I feel like the project has a lot of space for improvement. I ended up resorting to easy solutions like singletons access more than I wished. Given the time, I would decouple it as much as possible through a more robust system based on events. There is also some confusion on the point of customization by equipping/unequipping items. Finally, I would have developed more the instance size of the data objects I created, so I could support things like persistence and instance data like colors.

Conclusion

It was a great challenge to manage different systems and integrate them all in a short time span, especially in areas which I haven't worked extensively like Unity Netcode. Regardless, I feel proud of what was achieved given the time, since I know my way on how to polish it to make it better. Thank you for taking the time to evaluate this project!