Applicant Name: Mohamed Elmejrab

Date: 23/Dec/2022

Company: Blue Gravity Studio

Role applied for: Unity Developer

Development brief:

The task requested was clear and straightforward, creating a clothing shop in the style of "Stardew Valley" along with any functionality needed to make it work.

To achieve this goal I deconstructed the task to two main sections, first is the logic of handling world interactions. Second is the presentation of the world with the required feel/look.

The logic implementation needed to allow for the player to hold, acquire, and equip items, as well as interact with shopkeepers. With sustainability in mind I based this on an inventory system that hold the equipment the player has while keeping it open-ended for adding consumables or key items in the future. Units in the game have triggers and events to handle proximity which in turns starts dialog and subsequently buying/selling items, funds were integrated with checks on whether an item is affordable or not, once the player has acquired items, an equipment system handles which of these are equipped or not and displays them on the player sprite accordingly. The submission uses Lists to keep track of items and equipment which of course would not work in production and is used for demonstration only since data persistence was not a requirement on the test, however, a persistent solution like serializable classes along with network infrastructure can easily be used instead in a production environment.

The presentation side of things was easier to implement as the top-down style of Stardew Valley is well known and documented, Tilemaps with sorting layers were used to recreate the feel and create obstacles around the map, and pixel art assets were used to replicate that feel for both units and environment. To display equipment on a player in this view a "Paper-Doll" basic implementation was done, however, unfortunately no free assets for clothing items were found, so it may look horrible but again in a production environment with artists producing sprites and animations for these items, it would be easily achievable. The UI view have all the functionality needed, again it may not be the best in terms of style but it demonstrates clearly its functions. While not required, an audio system was added in the game to bring it to life, with background music and sfx for UI and dialogs.

Finally, on how well I feel I did, I would rate my work on this test as good/very good, I managed to implement all the functionality requested in the style set out by the team, naming conventions and code styles were adhered to in order to make scripts easily readable, and comments were used only if necessary. Since this was done in 4 days there is a huge room for improvement in efficiency and bug fixing, and depending on the code base this could be integrated into, refactoring some classes and functions in a certain way may have been better suited to it.

Regardless, I really enjoyed taking the test and I hope you find it of the required quality.