Initially, I decided to minimize the use of third-party assets and scripts from other projects of mine as much as possible, except for the recommended art assets. While this decision slowed down the development process, I believed it would make it easier to showcase my work and its process. Consequently, I focused more on creating a simple experience rather than a game, as one would typically do in a game jam. To streamline development, I opted to contain everything within a single room, featuring interactive elements for resource collection and a gear vendor for purchasing items that could be equipped.

My process began with selecting character and environment assets and outlining interaction mechanics. I then set a basic room environment and implemented the input system, character animation and movement. Subsequently, after finding a “soul fragment” sprite and a pentagram, I decided to use souls for the collectible item, which would serve as the in-game currency.

I then developed the logic for interactables, which I would use for the vendor and the pentagram, and integrated DOTWEEN to create smoother animations for souls moving towards the pentagram. With the pentagram mechanics established, I created a player manager to track soul gains, set states and enable the player to purchase items and sell items. Additionally, I designed a user interface to display the soul count.

Before proceeding to develop the vendor, I worked on designing the inventory UI and implemented the inventory logic to allow the player to equip items. I then developed the vendor system, introducing a player state for item selling and implemented tooltips to display item prices (for both buying and selling) above inventory slots. In the item selling state, clicking directly on items would either purchase (if owned by the vendor) or sell them(if owned by the player).

The final significant step was to display equipped items on the player character. Since the pack I used had separate sprites for each item and they aligned perfectly in frame count, I created a script to detect the player character sprite frame changes during gameplay and update equipment frame accordingly, without relying on an animator for the items (which would be time consuming to create an animation for each item). While I believe it would have been ideal to not use an animator for the player character and implement a logic for directly changing the player frames (with a tween curve) and then change the equipments frame, I opted for a quicker and less error-prone approach.

After completing and testing the game, I proceeded with the build. While the outcome didn't fully resemble a traditional game, I achieved the intended goal of showcasing a simple and streamlined interaction involving resource collection and gear purchasing. I contemplated more ambitious ideas, such as incorporating multiple active pentagrams (without the need of being idle to enable them) and enemies or obstacles, with vendors available at the end of each round for enhanced defense gear. However, I deemed these ideas too risky, potentially resulting in an incomplete or buggy outcome rushed under time constraints.