Initially, I decided to minimize the use of third-party assets 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 an interactive experience rather than a full-fledged "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 constructed a basic room environment and implemented character animation and movement. Subsequently, I devised the concept for the collectible item, settling on using a pentagram to collect souls, which would serve as the in-game currency.

I developed the logic for interactables, which I also used for the vendor and other elements, 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 and enable players to purchase items. Additionally, I designed a user interface to display the soul count, with the pentagram adding souls to the player's inventory.

Before proceeding to develop the vendor, I worked on designing the inventory UI and implemented the inventory logic to allow players to equip items. I then crafted the vendor system, introducing a new game state for item vending and implemented tooltips to display item prices (for both buying and selling) above inventory slots. In the item vending state, clicking directly on items would either purchase or sell them, bypassing the need for equipping.

The final significant step was to display equipped items on the player character. Since I used separate sprites for each item and ensured they aligned perfectly with character frames, I devised a system to detect frame changes and update equipment accordingly, without relying on an animator. While it would have been ideal to implement a logic for directly changing player frames without using an animator (or utilizing an asset from a friend capable of showcasing asset changes as "animations" in the editor), I opted for a quicker and less error-prone approach.

After completing and testing the functionality, 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 and enemy 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. I decided to reserve such endeavors for future