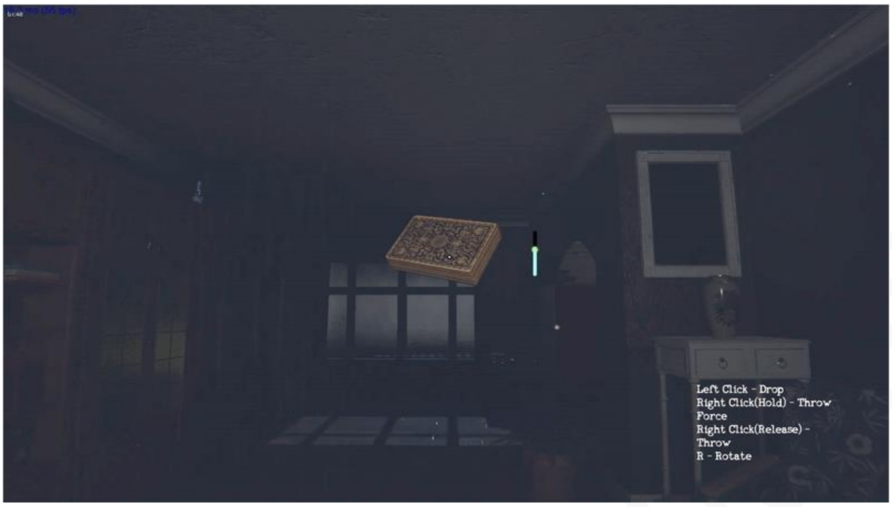
It Will Find You Interactivity System:

Goal: Create a flexible interaction system for the player to interact with objects throughout the environment, while allowing for variables based on the objects class.



Left Click - Drop
Right Click(Hold) - Throw Force
Right Click(Release) - Throw
R - Rotate

Definition: Throughout the game the player will encounter various objects in the environment that can be picked up and manipulated.

The player camera detects an object of the appropriate layer using a Raycast, altering the shader of the object so that the mesh is highlighted, identifying the object as selected to the player.

The player can then pickup the object and rotate it, before dropping or throwing the object with a variable charge defined by holding the secondary input. To allow for synchronised movement the object becomes a child of the main camera, and its Rigidbody is altered to ensure it remains in the correct position (as discussed in System flow).

The object is stored as a public variable for the system, so that even if the Raycast is no longer colliding with the object (e.g. if it has drifted due to environmental collisions), the object can still be thrown, dropped or interacted with while held.

While rotating the picked up object the player is locked in position, and the mouse movement rotates the object rather than the player camera. Variables (as listed below) can add additional functions to the interaction once an object has been picked up.

Variables:

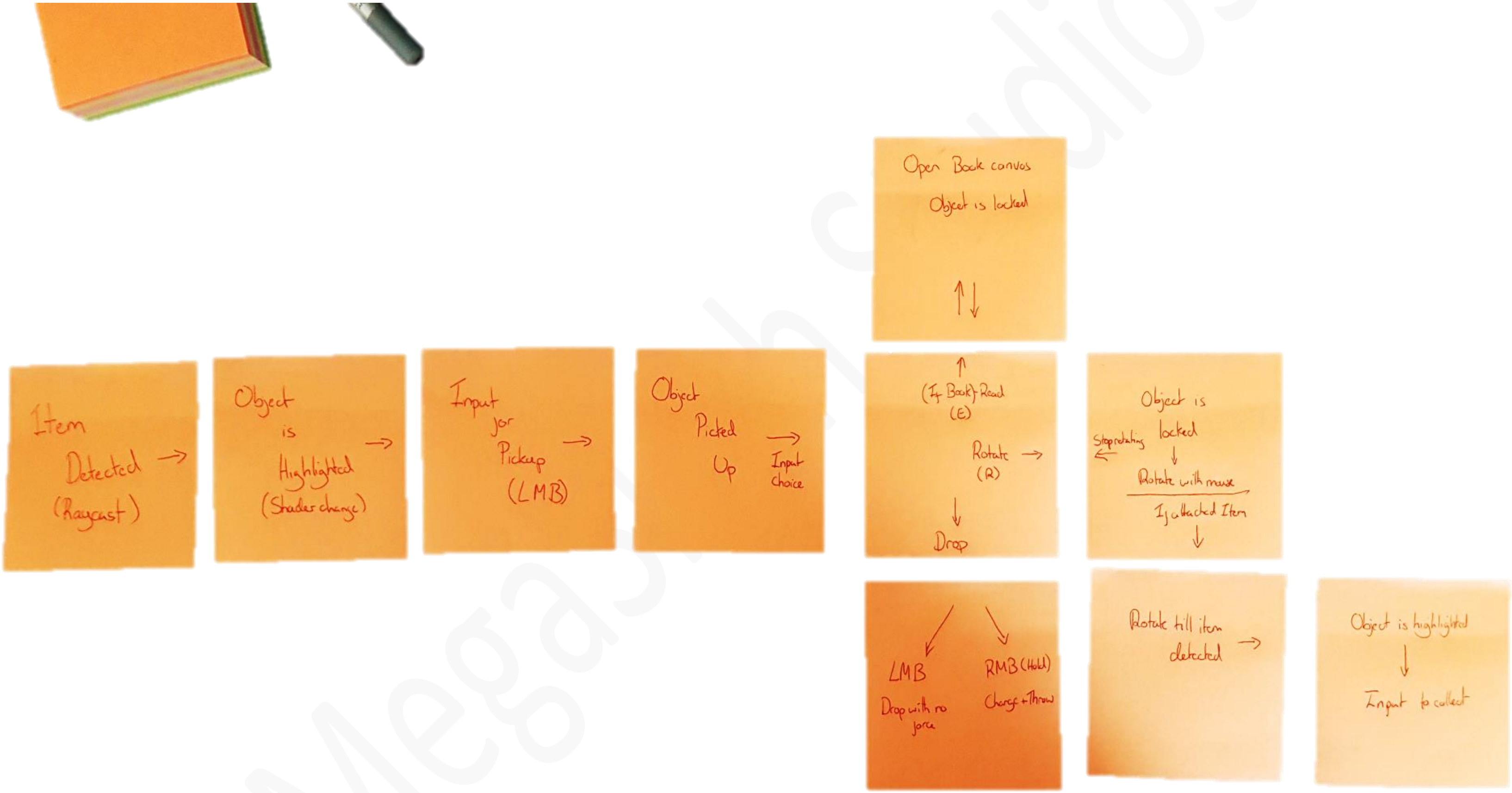
Standard Item – Can be picked up, rotated, dropped or thrown.

Attached Items – A secondary object e.g. a key that is attached to the main object. Uses a secondary trigger so that when the player can see the attached item, it can be removed and collected via a specific input.

Books – Books defined as readable open a specific book canvas linked to the object that allows the player to read the books contents. This locks the player in position, using the mouse to interact with the canvas and change pages.

Static Items – Objects that need to be returned to a specific location. When dropped or thrown these reset their position to a predefined location.

Paper Prototype:



Initial System Flow:

1. Raycast hits interactable Object.
2. Object changes shader to ‘Highlighter’.
3. Player can then pick up object using LMB.
4. Object becomes a child of the main camera.
5. Object removes highlighter once picked up, changes Rigidbody to kinematic.
6. Player rotates object or accesses variable such as attached item or book.
7. Player drops or throws item.
8. Rigidbody returns to non-kinematic.

This system flow worked well as an early concept in engine but presented several issues with clipping and a lack of physical interaction between held objects and the world. To solve these issues two solutions were explored.

To aid in physical interactions, the Rigidbody was no longer made to be kinematic, but was instead set to ignore gravity when picked up. This allowed the object to collide with environmental elements but remain in the players grasp.

To allow for some movement of the object, a bool is toggled when the object is not colliding with another object. Whilst this bool is toggled, the object will try to lerp back to its original held position. This means that while colliding, the object can be moved out of the centre position in a realistic manner, but will return to the player’s central view as soon as it is freed, with a failsafe dropping the object if it is taken too far from its initial position.

To aid with clipping, a secondary orthographic camera was placed inside the main camera, and given the ability to see a ‘No Clip’ layer that was not visible to the main camera. When the object was picked up, it would change layers to ‘No Clip’, and the orthographic camera would place it in front of all other objects, meaning it always appeared in front of the environment, rather than clipping through it.

Refined System Flow:

1. Raycast hits interactable Object.
2. Object changes shader to ‘Highlighter’.
3. Player can then pick up object using LMB.
4. Object becomes a child of the main camera.
5. Object removes highlighter once picked up, and changes layer to ‘No Clip’ layer. Also sets the Rigidbody to ignore gravity.
6. Player rotates object or accesses variable such as attached item or book.
7. Player drops or throws item.
8. Item returns to Interactable layer, sets use gravity on Rigidbody to true.

Interruptions:

Naturally occurring events can interrupt this system, e.g. an environmental event such as an explosion, or an attack from the monster. To compensate for this a drop function was implemented to allow for the object to be released from the players grasp as needed and returned to its original state.

Conclusion:

The system is now implemented and functioning as desired, with its flexible nature now allowing for further expansion if required. Whilst there are still improvements to the core functionality that could be made, the system is consistent and effective, and meets the initial goal established when design began.