**Individual Report : Md Meshkat Hossain Shoheb**

**Role : Student 1{Lego brick RB physics controller system(Rigid body Physics)}**

**Contribution Details:**

1**. Design and Implementation of PhysicsController Script:**

- Developed the core PhysicsController script, responsible for handling player input and translating it into movement for the block within the Unity environment.

- The script used Unity’s Rigidbody component to add physics-based movement forces in response to keyboard inputs (W, A, S, D) that allowed the player to control the block’s direction in the maze.

- The script was refined to ensure that movement was both responsive and precise, as initial tests showed that the block moved more than intended with a single keypress.

2**. Debugging and Enhancing Movement Control**:

- In initial testing, the block exhibited excessive movement and occasional mid-air flipping, which detracted from a realistic experience.

- To resolve this, Student 1 used the Rigid body’s \*\*Constraints\*\* settings to freeze rotation on the \*\*X\*\*, \*\*Y\*\*, and \*\*Z\*\* axes, ensuring the block stayed upright when colliding with maze walls.

- These adjustments allowed for smoother movement and prevented unwanted rotations, making gameplay more stable and predictable.

3**. Tagging and Collider Configuration for Walls**:

- I took the initiative to manage the tagging and collider setup for the maze walls, essential for proper wall-block interactions.

- By creating and assigning a custom "Wall" tag to the maze walls and adding Box Colliders, Student 1 ensured that the block would correctly detect and respond to collisions with walls. This setup provided the framework for detecting wall collisions programmatically.

4. **Testing and Quality Assurance**:

- I conducted testing of movement controls and wall interactions to ensure expected behaviour within the maze environment.

- During testing, adjustments were made to the Physics Controller script to balance force application, ensuring consistent movement speed and preventing unintended behaviours such as sliding or overshooting.

- Multiple test cases were conducted, including testing boundary collisions, response to consecutive keypresses, and overall block stability upon collision with walls.

**Challenges and Solutions**

1. **Excessive Block Movement**:

- Initial configurations resulted in the block moving further than intended upon single keypresses, requiring recalibration of the force application in the script.

- Solution: Modifying the `AddForce` parameters and using `ForceMode.Impulse` allowed me to apply a more controlled force to the block, resulting in better control.

2. **Unwanted Rotation and Flipping**:

- On wall collision, the block would rotate in unexpected directions, occasionally flipping over or getting lifted off the ground.

- Solution: Freezing rotation on all axes within the Rigid body’s Constraints prevented these unwanted behaviours, ensuring stable and realistic interactions.

3**. Collider and Tagging Setup**:

- Ensuring that each wall segment in the maze had appropriate tags and colliders was a critical part of the setup. Misconfigured walls could result in failed collision detection.

- Solution: I carefully assigned the custom \*\*"Wall"\*\* tag and checked that all walls were equipped with Box Colliders, which ensured proper detection within the PhysicsController script.

**Conclusion**:

My contributions focused on the PhysicsController’s functionality, ensuring precise player-controlled movement, handling wall collision mechanics, and enhancing stability. These contributions were instrumental in creating a smooth and realistic experience for navigating the maze game environment.