**Individual Report: Md Meshkat Hossain Shoheb**

**Contribution Details**

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

* Developed the core **PhysicsController** script to handle player input and translate it into movement for the Lego block in the Unity environment.
* Utilized Unity’s **Rigidbody** component to add physics-based movement forces in response to keyboard inputs (W, A, S, D), enabling directional control within the maze.
* Refined the script to ensure responsive and precise movement. Initial testing revealed excessive motion with single keypresses, which was optimized for better control.

**2. Debugging and Enhancing Movement Control**

* Addressed issues where the block exhibited excessive movement and occasional mid-air flipping, detracting from realism.
* Utilized **Constraints** in the Rigidbody component to freeze rotation on **X**, **Y**, and **Z** axes, ensuring stability and upright positioning during collisions.
* These adjustments resulted in smoother movement and prevented unwanted rotations, improving gameplay experience.

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

* Managed tagging and collider setup for maze walls to ensure proper interaction between the Lego block and the environment.
* Assigned a custom **"Wall"** tag to maze walls and added **Box Colliders** to ensure collision detection. This setup facilitated seamless wall-block interactions within the PhysicsController script.

**4. Testing and Quality Assurance**

* Conducted rigorous testing of movement controls and wall interactions to ensure expected behavior.
* Adjusted force application parameters in the PhysicsController script to balance movement speed and minimize sliding or overshooting.
* Tested various scenarios, including boundary collisions, consecutive keypresses, and stability upon collisions, ensuring consistent and realistic performance.

**Creating a Lego Minifigure in 3ds Max**

**Overview**

In addition to scripting the PhysicsController, I contributed by creating a **Lego Minifigure** model using **3ds Max**, which added an aesthetic and interactive component to the project. This involved modeling, texturing, and assembling individual parts for a realistic and customizable minifigure.

**Modeling Process**

1. **Reference Setup:**
   * Imported front, side, and top reference images of a Lego minifigure into 3ds Max as blueprints for accurate modeling.
   * Placed the reference images on planes in the viewport for guidance.
2. **Body and Head:**
   * Modelled the torso and legs using **Box** primitives and refined them with the **Edit Poly** modifier for proper dimensions and details.
   * Created the head with a **Cylinder**, adding edge loops to define the top and bottom. Chamfered edges for a smooth, rounded appearance.
3. **Arms and Hands:**
   * Crafted arms using **Cylinders**, shaping them to resemble the curved Lego arm.
   * Designed hands with a **Torus**, modified using the **Edit Poly** tool to create the iconic clamp shape for holding objects.
4. **Accessories and Details:**
   * Modelled additional elements like helmets or tools using primitives (e.g., Sphere or Cylinder) and modified them with boolean operations for precision.
   * Ensured proportions aligned with Lego minifigure standards.

**Texturing and Materials**

* Unwrapped the UVs using the **Unwrap UVW** modifier and applied custom textures to the torso and head.

**Assembly and Hierarchy**

* Assembled individual parts (head, torso, legs, arms, and accessories) into a hierarchical structure for easy manipulation and posing.
* Adjusted pivot points to enable proper rotation and animation if needed.

**Integration and Testing**

* Exported the finished model in a Unity-compatible format (.FBX) and integrated it into the maze environment.
* Verified the minifigure's compatibility with the physics system, ensuring smooth movement and accurate collision detection.

**Challenges and Solutions**

**1. Excessive Block Movement**

* **Challenge:** Initial settings caused the Lego block to move too far with a single keypress.
* **Solution:** Adjusted AddForce parameters, using ForceMode.Impulse for controlled force application.

**2. Unwanted Rotation and Flipping**

* **Challenge:** The block rotated unpredictably upon collision with walls.
* **Solution:** Freezing rotation on all axes within Rigidbody constraints resolved this issue.

**3. Minifigure Proportions and Details**

* **Challenge:** Ensuring accurate proportions and small details like hand clamps required precision.
* **Solution:** Repeated iterations using reference images ensured a realistic and proportionate model.

**Conclusion**

My contributions encompassed scripting the PhysicsController, refining movement mechanics, and creating a Lego minifigure using 3ds Max. These efforts combined technical and artistic elements to enhance the overall project, providing a realistic and engaging experience within the Unity maze environment.