How did you structure your main game loop or update cycle in Unity?

In Start(), I snap the player to the ground with a raycast and set vertical speed. In Update(), I read Input System actions (Move, Sprint), make a camera-relative move vector, apply gravity and ground stick, and move with CharacterController.Move(). I rotate toward the move direction. I also set Animator values: Speed (0–1) and IsSprinting.

What tools or methods did you use to design and texture your environment (e.g., terrain tools, tilemaps, layers)?

I used Unity Terrain tools to sculpt hills and smooth them. I painted multiple textures (grass and dirt for now). I added a capsule collider for the tree prefab, and will add scripts to it later for Chopping. I added a skybox and a directional light for basic lighting. For trees and grass terrain, I used a free asset from Unity Store and itch.io. The current terrain is still on a small scale since I want to set the basic features first before expanding them.

How did you implement player controls (movement, jumping, camera/view rotation, etc.)?

Movement is WASD using the Input System (“Player/Move”). Sprint is Left Shift (“Player/Sprint”) with a speed multiplier. Gravity is applied each frame; the ground stick keeps the player grounded. The camera is fixed at a set angle and follows the player (no player look input). The Animator switches Idle/Run/Sprint using the Speed float and IsSprinting bool. Jumping is not available for now since I feel like there is no need for one. The asset that I’m currently using (Blink; free asset from Unity Store) contains all animations, so I just need to do a simple animation controller.