System test case - #2

Test case details

Tester:	Eetu Luoma
Date:	27.3.2025
Device:	Desktop computer, high-end, Windows 11 Home V.24H2 x64
Environment:	

- Most recent main branch of the project repository
- Most recent commit is 519a6946df6fc2cc716ce0d89dd38bfb8afa63f0
- Unity editor version 6000.0.35f1
- Generated runnable .exe via File > Build and run

Test details

This system test is exploratory in nature. The simulation is run via the built .exe with the purpose of moving around the environment, trying to break it somehow: walking and sprinting into objects, moving against them, jumping and crouching. The test is ended if the tester manages to break the simulation somehow, otherwise the tester decides when to stop exploring the application.
Point scanning is ignored in this test.

Note: At the time of performing this test, the natural boundaries for play area (https://gitlab.tuni.fi/cs/gamilidar/simulated_forest_scanning/-/issues/51) was not yet complete, so the user is very easily able to jump into the void and get outside of the play area. Therefore no testing related to the edge of the are is done.

Test steps and results

1. Open the project with Unity Editor, go File > Build Profiles, and Build the project for the Windows platform. The tester chooses to build it into a folder called 'BuildTarget' in the repository directory.
    a. Result: Build succeeded in 11 seconds. Simulated forest scanning.exe now exists in the directory where building was done to.

2. Run 'Simulated forest scanning.exe'
    a. Result: Simulation starts successfully

3. Explore the simulation
    a. Result: Discovered possibility of user clipping their camera into a rock, which causes user to see inside the rock and therefore points to be scannable from the terrain beneath the rock. We don't really think this is problematic, although it is not very realistic for a simulation.
    b. Result: Discovered that visualizing the AR mesh we simulate from scanning seems to affect performance adversely. However changes are incoming in which visualizing it is disabled by default, and a toggle button for it will be implemented. Created a new system test case / issue to explicitly explore this performance aspect.
    c. Result: Discovered that player currently collides with the simulated AR mesh. Fix incoming for this.

4. Terminate the simulation after playing around for 20 minutes
    a. Result: Program terminated.

Conclusions

Simulation seems fairly robust right now, ignoring the missing play area limits and the problems the recent AR mesh simulation feature brought. I was not able to break the world in any meaningful way, which is definitely good. In my opinion the simulation is usable at a sufficient level, and I consider this a successful system test run.