|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Current Issue** | **Input Data (if applicable)** | **Solution/Cause** |
| 1 | Chunks are rendering all quads and blocks even if the camera has no way of seeing them. | N/A | By creating a new array of Boolean values, I set all but the surface blocks to false and then iterate through this array to only render the true values. |
| 2 | The structure of the blocks is fine but the normal are not being rendered properly. | N/A | This was down to Unities way of handling Quads and triangles; Unity will automatically try to smooth out any vertices and normal generated through code to save GPU usage on most meshes however I want perfect cubes. To fix this each point in a cube actually has 3 points for each possible side it connects too, rather than the 1 I had before, this is required for my desired result. |
| 3 | Chunks that aren’t even in view of the player are still being drawn, causing a heavy load on both the CPU and the GPU. | N/A | Similar to the 1st solution except I use an array of Chunk objects and check the bool isActive. I then modify and change these chunks every time the camera moves to a new chunk. When in a new chunk it will update 2 lists, one for the currently active chunks and one for the previously active chunks. Any chunks left in the previously active chunks list will be disabled but will remain in the chunk array, therefore if I revisit those chunks I check if they are already in the chunk array and just reactivate them rather than rebuild them. |
| 4 | World generation is very slow. | N/A | By changing some of the subroutines that do not return a value to coroutines, I have managed to temporarily reduce the amount of time it takes to generate chunks. I have also streamlined the block class by switching most of the lists for arrays or just removing them altogether as this was causing a lot of unnecessary memory usage. |
| 5 | While in block break and place mode, the block selection doesn’t work for any blocks on the -x or -z axis. | Input “Shift” key | Through a series of If statements that use the checkForVoxel subroutine, I now check all directions at the mouse cursor for a block. |
| 6 | World generation is still quite slow when generating completely new chunks | N/A | By separating my Data from my rendering processes, I have added threading into the chunk scripts. All data for chunks are calculated in their own threads and will end while queueing those chunks to be rendered through the main thread. |
| 7 | The blocks would still remain once it had been broken | Input and hold “Shift” key and then click “Left mouse button” on a block |  |
| 8 | World generation was quick but would easily reach thread cap for most CPU’s which would cause very glitchy world generation and missing chunks. | N/A | I have averted this issue by running a thread on a while loop and calculating all my data for my chunks on that thread instead of starting a new thread for every chunk. This thread ends when the player stops the game. All chunks are still rendered through the main thread. |
| 9 | When breaking the base wooden block on a tree the tree should fall down, however it currently only works for the first tree broken in a chunk, any other trees in the same chunk do not break. | Input and hold “Shift” key and then click “Left mouse button” on the base of a tree | This was caused because I never subtracted from the variable structuresToBuild, therefore doubling the tree’s and never breaking them. To fix this I just iterated back down structuresToBuild. |
| 10 | Unity’s in built navmeshing will not generate properly in accordance to my procedural world. | N/A | This is because of Unity’s in-built system limitations. I will fix this by creating my own pathfinding system so I will not require navmeshing or any Unity based scripts. |
| 11 | When selecting another colonist while the previous one is still moving, the pathfinding clears but the animations do not change. | Input “Left mouse button” on a worker and then click a block, then click on another worker whilst he is still moving. | Fixed by changing the moveWorker subroutine to move the parent of the script rather than the selected worker. |
| 12 | If the journey is short (approx 1 or 2 blocks) then the colonist does not switch back from the IdleToMove bridge animation and will just freeze in a walk pose. | Input “Left mouse button” click on a worker and then click a block | The If statements in the updateMovement subroutine were not in parenthesis so only 1 of the bool changes was passing, causing the animation to not update properly. |
| 13 | The game completely freezes when there is an obstacle in the way of the colonist (such as a wall) | Input “Left mouse button” click on a worker and then click a block while there is a wall in the direct path towards the block | Fixed by adding a subroutine that will automatically start when there is a wall in the way. This subroutine will be a keep left system, the colonist will hug the wall, keeping left of it until it reaches the same X or Z coordinate (depending on the direction of the wall). Once this coordinate is reached it will revert back to the old beelining and continue until it reaches this point or there is another wall. |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 |  |  |  |
| 21 |  |  |  |
| 22 |  |  |  |
| 23 |  |  |  |
| 24 |  |  |  |
| 25 |  |  |  |
| 26 |  |  |  |
| 27 |  |  |  |
| 28 |  |  |  |