Kyle Singer, Joseph Grosso, Andrew Farley, Andrew Fryer

Team Aggasiz

Brick Builder

## Our Goal:

The goal of the challenge was to unscramble clients’ pixel models into their correct positions as outlined by the files given by our boss. The main challenge with this was to optimize the time with which the drone fixed the broken piece. Clients expect quick fixes to their challenges so our main objective was to fix their products as quickly as possible.

## Against the Grain:

One of the main focuses of our team was how we can develop a solution that not only optimized time, but that also works in untraditional ways. Rather than using a very standard algorithm of moving block sorting stack by stack, we created a process that is much more versatile and that can be built off and applied to other projects.

With the ECL, one of the main things that resonated with our team was deliberating values. We decided to enforce values of sustainability, creativity, and effects on future development rather than strictly usability for specific examples.

## Algorithm:

The goal of this algorithm was to optimize the time with which it takes to reconstruct the blocks into their correct positions. The algorithm works as follows: first our drone does a full sweep over the entire matrix so we get a two dimensional overview. In this sweep, the drone recognizes which blocks in the z=0 with nothing stacked on top are in their correct locations as well as where the empty spots are. In this case, the algorithm also takes in the number of colours that exist amongst the blocks.

The algorithm uses any blocks in the z=0 states that are correct and any blank spots as places to stack other blocks for futures use of the same colour. This way we have more capabilities to store block in addition to knowing where different colour blocks are.

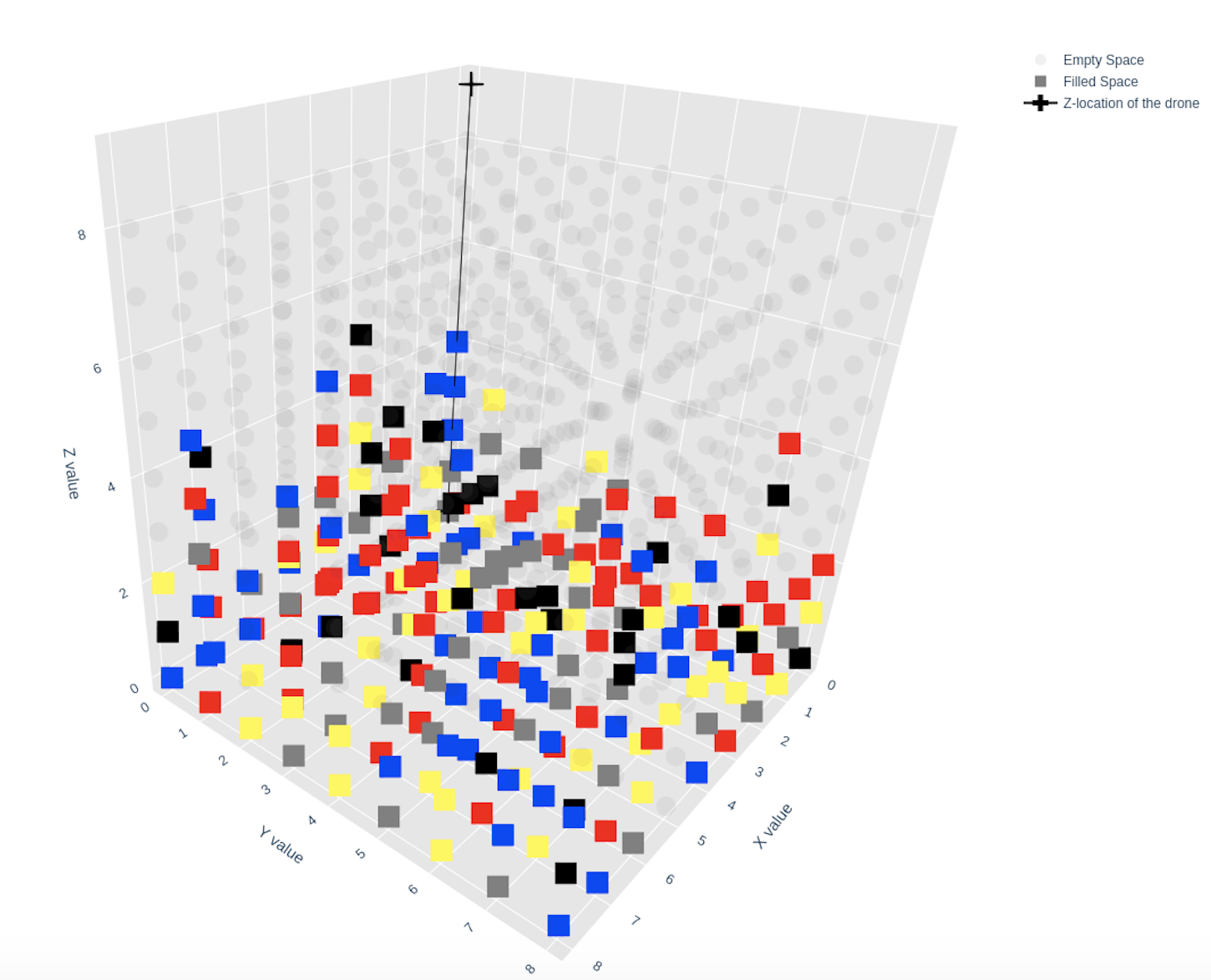
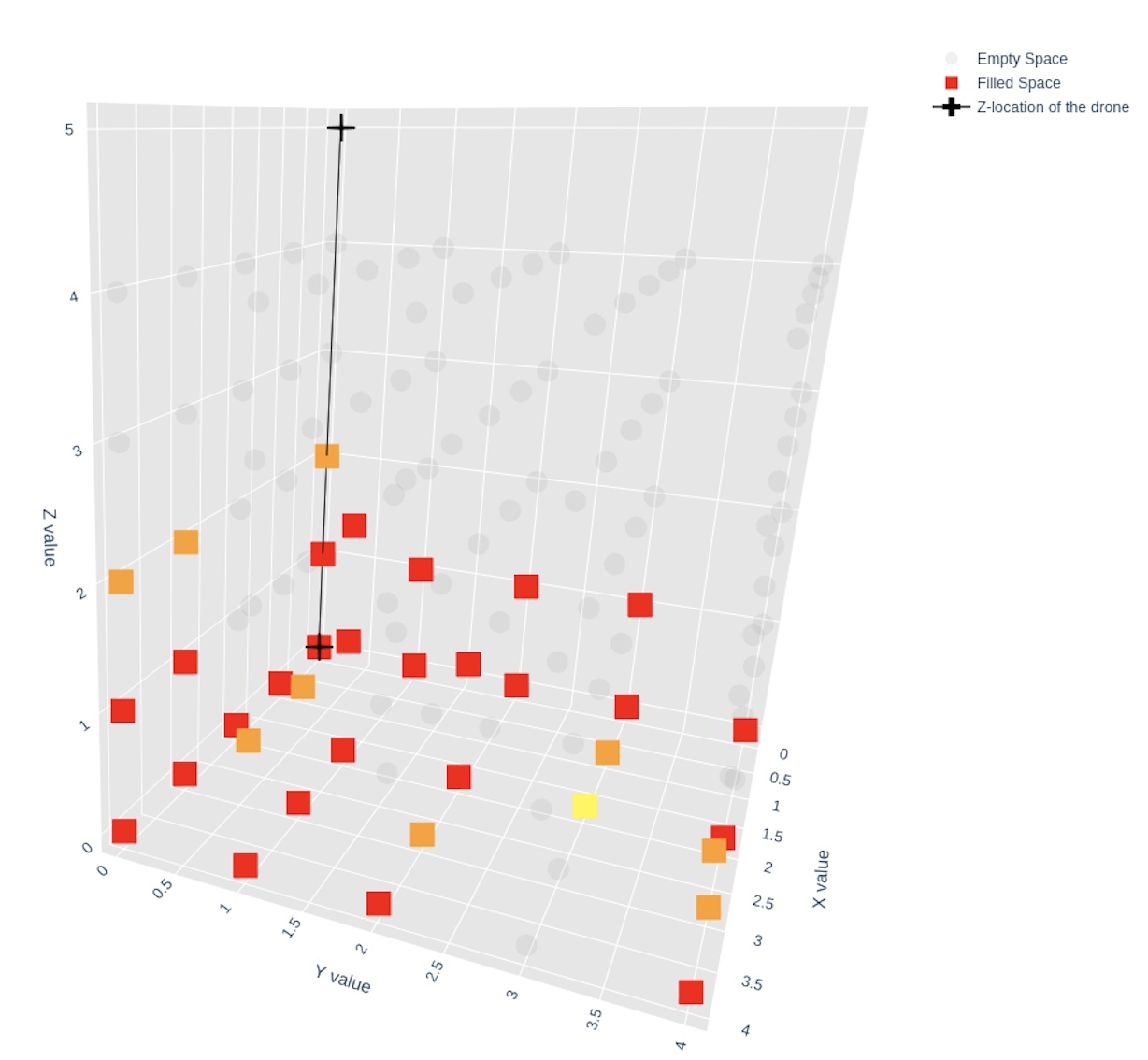
Once this documentation has been completed the drone begins to go over the blocks. Starting with blocks at z=0 with nothing on top, the drone begins to pick up the blocks and shift them into their correct positions with extra ones being stored on the verified stack locations outlined by z=0 blocks or blank spots in their correct positions.

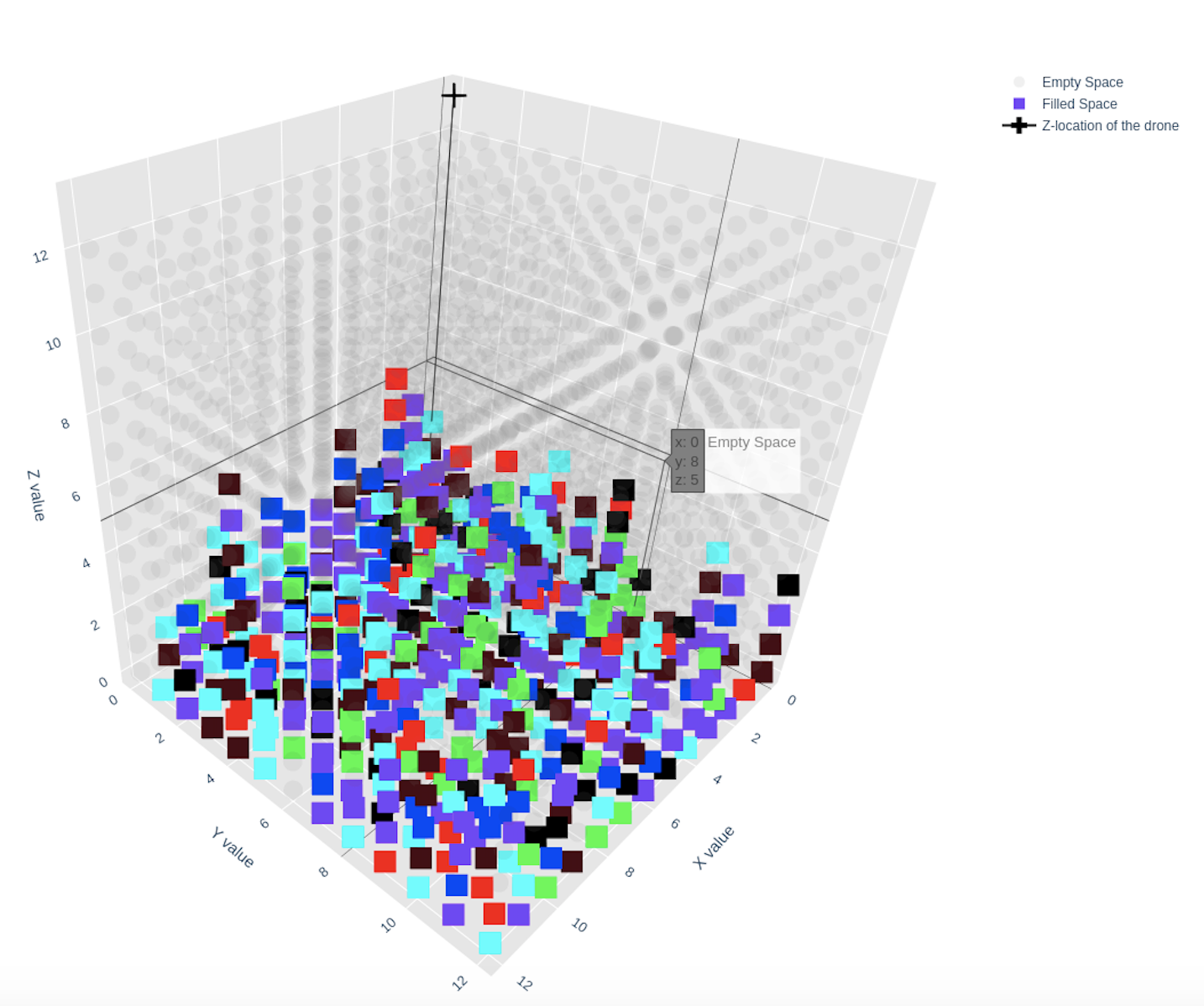
Once this step has been completed, the algorithm moves on to the z=n state where n represents all the possible dimensions between 0 and the limit n. Here, one by one we increase the z value stacks we are looking at (anything with a limit for the z-value we are specifically looking at i.e. z=1, z=2...z=n).

Now that we know where everything is and what is in the correct locations, the drone starts to pull from different stacks to build up the blocks into their correct locations. There is a simple function that takes in the distance from the drone to the varying storage stacks so we can optimize the travel time of the drone.

## Visualization

Scrambled graphs:





Unscrambled graphs:

A picture containing computer

Description automatically generatedA picture containing computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated