For the System test we added every kind of command block and filled out arguments for them. We reordered them and exported the script to verify the results.

Max: Tested deleting command blocks by adding multiple command blocks to the list, calling delete on them and then exporting the script to see if they had been removed properly. He tested editing blocks by entering different arguments into blocks of the same and different types, seeing if the values saved to the correct blocks and editing them multiple times to see if values continued to update. He tested interference of editing with other components by reordering or deleting blocks that were in the process of being edited and fixing problems that this caused.

Thomas: Tested drag and drop by reordering blocks with drag and drop. Then verified the backend mirroring of the reorder by exporting the script.

Wil: Tested backend calls by calling reorder, delete and add command functions independently of the frontend controls; maintained to-do list to track completion of acceptance criteria; fixed back end items and handed off most front end issues.

Rory: Most of my work was done on the back-end; this meant that I had full access to STDOUT and STDERR. Most of my testing consisted of print statements based on the toStrings I wrote for each of the back-end structures (namely: ScriptStruct and Command). These were used to check the status of the structures at any given time and scope. Similarly, I used back-tracking (scope-wise) print statements to root out bugs. The only other (read: more advanced) style of testing I utilized was IntelliJ's step-in/out/over debugging functions to watch the stack line-by-line. I am not a fan of this style, so I used to sparingly.