# Quarantine Gang Test Plan and Results

## Overall Test Plan

In general, our tests will be split into three categories: unit tests, performance tests, and functionality tests. The unit and performance tests will be the simplest to implement, at least in terms of their framework. For the unit tests, we'll simply be ensuring that our methods return the expected values for a variety of inputs. The performance tests will work in basically the same way, except that instead of checking the output for correctness, we'll be checking execution time for speed.

The functionality tests will be a bit more complex. These will require us to simulate actual user interaction with the interface. This may require us to use external software to emulate mouse and keyboard actions. These tests will be designed to validate that our application functions properly for end users.

## Test Case Descriptions

DS1.1 **Drill Solver Test 1 : Passed**
DS1.2 This test ensures that static sets are solved correctly
DS1.3 This test will determine if 2 identical, but scrambled, sets are solved correctly. This is the simplest drill move.
DS1.4 Inputs: 2 sets of 25 marchers arranged in a 5x5 4 step grid. The 2nd set will be scrambled.
DS1.5 Output: a set identical to the initial set
DS1.6 Normal
DS1.7 Blackbox
DS1.8 Functional
DS1.9 Unit

DS2.1 **Drill Solver Test 2 : Passed**
DS2.2 This test ensures that translated sets are solved correctly
DS2.3 This test will determine if translated and then scrambled sets are solved correctly. This is a simple drill move.
DS2.4 Inputs: 2 sets of 25 marchers arranged in a 5x5 4 step grid. The second set will be translated 8 steps to the right and then scrambled.
DS2.5 Output: A set identical to both the initial set translated 8 steps to the right and the pre-scrambled second set.
DS2.6 Normal
DS2.7 Blackbox
DS2.8 Functional
DS2.9 Unit

DS3.1 **Drill Solver Test 3 : Passed**
DS3.2 This test ensures that expanded sets are solved correctly

DS3.3 This test will determine if expanded and then scrambled sets are solved correctly. This is a simple drill move.
DS3.4 Inputs: 2 sets of 25 marchers arranged in a 5x5 grid. The first set will be a 4 step grid and the second will be an 8 step grid, with both grids centered at the same point. The second set is then scrambled
DS3.5 Output: A set identical to both the initial set expanded to an 8 step grid and the pre-scrambled second set.
DS3.6 Normal
DS3.7 Blackbox
DS3.8 Functional
DS3.9 Unit

DS4.1 **Drill Solver Test 4 : Passed**
DS4.2 This test ensures that split sets are solved correctly
DS4.3 This test will determine if split sets are solved correctly. This is a somewhat complex drill move.
DS4.4 Inputs: 1 set of 25 marchers arranged in a 5x5 4 step grid and a second set with the first 3 columns of set 1 moved 8 steps to the left and the last 2 columns moved 8 steps to the right. The second set is then scrambled.
DS4.5 Outputs: A set identical to the pre-scrambled second set.
DS4.6 Normal
DS4.7 Blackbox
DS4.8 Functional
DS4.9 Unit

DS5.1 **Drill Solver Test 5 : Passed**
DS5.2 Benchmark the performance of the drill solver algorithm
DS5.3 This test run multiple drill solving tests with 300 marchers to make sure the algorithm can efficiently find solutions.
DS5.4 The inputs are the drill sets
DS5.5 The expected outputs are the solved sets, and the time taken solving.
DS5.6 Boundary cases would be sets that are particularly easy or hard to solve
DS5.7 Whitebox
DS5.8 Performance
DS5.9 Unit test

UI1.1 **User Interface Test 1 : Passed**
UI1.2 Test the band size/instrumentation features

UI1.3 This test will verify that users can set and save information about the size of the band and the instruments being used. This will involve changing these settings and verifying the effects they have on how the solution is computed.
UI1.4 The inputs would be the mouse/keyboard and UI inputs for setting and saving the parameters and what the parameters are exactly
UI1.5 We should see the changes reflected accurately in the data processing
UI1.6 Abnormal inputs would be invalid inputs
UI1.7 blackbox
UI1.8 functional
UI1.9 Integration test

UI2.1 **User Interface Test 2 : Passed**
UI2.2 Testing the ability to load shows
UI2.3 This test will simulate load a file into the application and make sure that the application is able to successfully load the data from the file
UI2.4 input will be a saved drill set
UI2.5 output will be the data from the loaded drill set
UI2.6 normal
UI2.7 whitebox
UI2.8 functional
UI2.9 unit test

UI2.1 **User Interface Test 3 : Passed**
UI2.2 Testing the ability to save shows
UI2.3 This test will simulate a save with a known drill set and make sure the set is saved in a format the application can read
UI2.4 The input will be a drill set
UI2.5 The output will be a save file with the the drill set data included
UI2.6 normal
UI2.7 whitebox
UI2.8 functional
UI2.9 unit test

UI3.1 **User Interface Test 4 : Passed**
UI3.2 Testing the ability to graphically view each set and count in a loaded show
UI3.3 This test will load a drill set and take a screenshot of the loaded show and then check that the image matches a screenshot of the expected display.
UI3.4 The input will be a saved drill set
UI3.5 The output screenshot of the user interface and a comparison with a known image
UI3.6 normal

UI3.7 blackbox
UI3.8 functional
UI3.9 integration

IC1.1 **Image Conversion Test 1 : Passed**
IC1.2 Testing the ability to convert images to a drill set
IC1.3 This test will load images of basic shapes into the images converter and check if the outputed drill set coordinates match up with an expected result of the basic shape.
IC1.4 The input will be an images of a shape (square, circle, triangle)
IC1.5 The output will be a numpy array with the coordinates of the dots that are equidistant to each other across shaded parts of the shape
IC1.6 normal
IC1.7 whitebox
IC1.8 functional
IC1.9 unit

IC2.1 **Image Conversion Test 2 : Passed**
IC2.2 Testing the ability reject files that are not images
IC2.3 This test will load a file that is not an image and check if an error message is given back to the user.
IC2.4 The input will be a text file
IC2.5 The output an error message explaining detailing the accepted file types that can be used
IC2.6 abnormal
IC2.7 blackbox
IC2.8 functional
IC2.9 unit

DE1.1 **Drill Editor Test 1 : Passed**
DE1.2 Testing the ability to edit the drills in the drill editor
DE1.3 This test will check if user is able to complete basic edits on a loaded drill set
DE1.4 The input will be a saved drill set and mouse and keyboard clicks
DE1.5 The output will a drill set with the correct elements moved from the edits
DE1.6 Normal
DE1.7 Blackbox
DE1.8 Functional
DE1.9 Integration

DE2.1 **Drill Editor Test 2 : Passed**
DE2.2 Testing that drill editor notifies the user of potential collision after an edit and after a drillset creation via image conversion

DE2.3 This test will focus on the intelligence of the drill editor, specifically on ensuring minimal collisions in drill. This is part of testing input validation.
DE2.4 Two drillsets which will cause certain collisions
DE2.5 The drill editor should create a pop-up showing the collision issue and asking how to resolve it.
DE2.6 Normal
DE2.7 Blackbox
DE2.8 Functional
DE2.9 Unit

DE3.1 **Drill Editor Test 3 : Passed**
DE3.2 Testing that drill editor notifies the user of unreasonable step counts after an edit and after a drillset creation via image conversion
DE3.3 This test will focus on the intelligence of the drill editor, specifically on ensuring reasonable step size. This is part of testing input validation.
DE3.4 An image with dots mostly on the left end of the field followed by an image with dots mostly on the right end of the field. Also, moving one marcher's dot manually with the editor to an unreachable spot
DE3.5 In both cases, the drill editor should have a pop-up showing the step size issue and asking how to resolve it.
DE3.6 Normal
DE3.7 Blackbox
DE3.8 Functional
DE3.9 Unit

DE4.1 **Drill Editor Test 4 : Passed**
DE4.2 Testing the ability to convert moves between floats, flanks, and follow the leader drill moves.
DE4.3 This will test that drill moves can have their move type edited and saved out properly. This will also test marcher selection.
DE4.4 Any given drill move should be able to be converted between these move types.
DE4.5 Switching the type of move should change the path for every selected marcher to match the new type.
DE4.6 Normal
DE4.7 Blackbox
DE4.8 Functional
DE4.9 Unit

DE5.1 **Drill Editor Test 5 : Passed**

DE5.2 Test the ability to convert moves between halftime, singletime, and doubletime steps, as well as scatters.
DE5.3 This will test that drill moves can have their step speed changed and saved out properly. This will also test marcher selection.
DE5.4 Any given drill move should be able to be converted between these step speeds.
DE5.5 Step type should be visible when selecting a certain marcher or group of marchers.
DE5.6 Normal
DE5.7 Blackbox
DE5.8 Functional
DE5.9 Unit

DE6.1 **Drill Editor Test 6 : Passed**
DE6.2 Test import and export features.
DE6.3 Shows will need to be saved and loaded from multiple file types. This test will ensure that this program can read and write these file types properly.
DE6.4 Pyware drill file, drillcreator drill file
DE6.5 Should be able to successfully give visual output on loading either file, make edits, and save to both file types for each file.
DE6.6 Normal
DE6.7 Blackbox
DE6.8 Functional
DE6.9 Unit

**Test Case Matrix**

| | **Normal/ Abnormal** | **Blackbox/ Whitebox** | **Functional/ Performance** | **Unit/ Integration** | **Pass/Fail** |
|---|---|---|---|---|---|
| **DS1** | Normal | Blackbox | Functional | Unit | P |
| **DS2** | Normal | Blackbox | Functional | Unit | P |
| **DS3** | Normal | Blackbox | Functional | Unit | P |
| **DS4** | Normal | Blackbox | Functional | Unit | P |
| **DS5** | Boundary | Whitebox | Performance | Unit | P |
| **UI1** | Abnormal | Blackbox | Functional | Integration | P |
| **UI2** | Normal | Whitebox | Functional | Unit | P |
| **UI3** | Normal | Whitebox | Functional | Unit | P |

| | | | | | |
|---|---|---|---|---|---|
| **UI4** | Normal | Blackbox | Functional | Integration | |
| **IC1** | Normal | Whitebox | Functional | Unit | |
| **IC2** | Normal | Blackbox | Functional | Unit | |
| **DE1** | Normal | Blackbox | Functional | Integration | |
| **DE2** | Normal | Blackbox | Functional | Unit | |
| **DE3** | Normal | Blackbox | Functional | Unit | |
| **DE4** | Normal | Blackbox | Functional | Unit | |
| **DE5** | Normal | Blackbox | Functional | Unit | |
| **DE6** | Normal | Blackbox | Functional | Unit | |