

CARNEGIE MELLON UNIVERSITY

ROBOTICS CAPSTONE PROJECT

# Performance Validation Demonstration

*Friction Force Explorers:*

*Don Zheng*

*Neil Jassal*

*Yichu Jin*

*Rachel Holladay*

supervised by  
Dr. Cameron RIVIERE

Version 1.0  
May 11, 2017

# Contents

<b>1</b>	<b>System Overview</b>	<b>3</b>
<b>2</b>	<b>Risk Management</b>	<b>3</b>
<b>3</b>	<b>Future Work and Commercialization Considerations</b>	<b>6</b>
	<b>Appendices</b>	<b>7</b>
<b>A</b>	<b>Testing and Evaluation Plan</b>	<b>7</b>
A.1	Design Verification . . . . .	7
A.2	Writing Implement . . . . .	7
A.2.1	Performance Test: Loading . . . . .	7
A.2.2	Performance Test: Writing Quality . . . . .	7
A.2.3	Functional Test: Simultaneous Driving and Writing . . . . .	7
A.2.4	Functional Test: Marking . . . . .	7
A.3	Locomotion . . . . .	8
A.3.1	Performance Test: Accuracy . . . . .	8
A.3.2	Functional Test: Speed . . . . .	8
A.3.3	Functional Test: Omnidirectional . . . . .	8
A.4	Localization . . . . .	8
A.4.1	Performance Test: Robot Position Accuracy . . . . .	8
A.4.2	Performance Test: Bounds Accuracy . . . . .	8
A.4.3	Functional Test: Robot Position . . . . .	8
A.4.4	Functional Test: Bounds . . . . .	9
A.5	Input Processing . . . . .	9
A.5.1	Functional Test: Return Data . . . . .	9
A.5.2	Functional Test: Reject Improper Input . . . . .	9
A.6	Work Scheduling, Distribution and Planning . . . . .	9
A.6.1	Performance Test: Executable Plans . . . . .	9
A.6.2	Performance Test: Execution Distribution . . . . .	9
A.6.3	Performance Test: Drawing Distribution . . . . .	9
A.6.4	Performance Test: Speedup . . . . .	10
A.6.5	Functional Test: Collision Free . . . . .	10
A.6.6	Functional Test: Autonomy . . . . .	10
A.7	Communication . . . . .	10
A.7.1	Performance Test: Uptime . . . . .	10
A.7.2	Functional Test: Sending and Receiving Data . . . . .	10
A.7.3	Functional Test: Data Parsing . . . . .	10
A.8	User Interface . . . . .	11
A.8.1	Performance Test: Emergency Stop Speed . . . . .	11
A.8.2	Performance Test: Error Reporting Delay . . . . .	11
A.8.3	Performance Test: Error Understandability . . . . .	11
A.8.4	Functional Test: Emergency Stop . . . . .	11
A.8.5	Functional Test: Error Reporting . . . . .	11
A.9	Power System . . . . .	12
A.9.1	Performance Test: Battery Life . . . . .	12
A.10	Full System Validation . . . . .	12
<b>B</b>	<b>Full System Validation</b>	<b>12</b>
B.1	Performance Test: Painting Accuracy . . . . .	12
B.2	Performance Test: Reliability . . . . .	12
B.3	Functional Test: Size . . . . .	12
B.4	Functional Test: Weight . . . . .	12
B.5	Functional Test: Budget . . . . .	13
B.6	Functional Test: Safety . . . . .	13
B.7	Functional Test: Documentation . . . . .	13

**List of Figures**

1	Risk 1: Defective Parts . . . . .	3
2	Risk 2: Unavailable Member . . . . .	3
3	Risk 3: Breaking Parts . . . . .	4
4	Risk 4: Mecanum Drive Too Unstable . . . . .	4
5	Risk 5: Localization not precise enough . . . . .	5
6	Risk 6: Unexpected Budget Overruns . . . . .	5
7	Planner Test Cases. . . . .	14

# 1 System Overview

**RH: describe project**

## 2 Risk Management

**RH: DON PLEASE UPDATE**

In this section, we revisit the risks defined by our previous document. The risk tables have been updated with the actions we've taken to address the risks.

Risk ID:	Risk Title:	Risk Owner:	Actions taken:					
	1 Defective Parts	Don						
Description:								
Parts that we ordered arrived defective or do not perform to specifications			No issues Persistent: Only ordering reliable parts					
Consequences:	Risk Type:			Consequence				
We need to reorder parts, expending extra time and budget	Parts			1	2	3	4	5
Risk Reduction Plan:	Expected Outcome							
We will order only parts that have been extensively reviewed, or we have experience with, and order extra parts	We will be able to properly deal with any parts that break during the development process							
			Likelihood					

Figure 1: Risk 1: Defective Parts

Risk ID:	Risk Title:	Risk Owner:	Actions taken:				
Unavailable Group 2 Member		All					
Description:							
A group member becomes unavailable for work due to travel, sickness, or other emergencies							
		No issues Persistent: Only ordering reliable parts					
Consequences:	Risk Type:		Consequence				
Work that would have been distributed to that group member needs to be reassigned			1	2	3	4	5
	Logistical						5
Risk Reduction Plan:	Expected Outcome						4
We will ensure that every group member is always on the same page about progress so we don't lose too much progress					X		3
	If a member becomes unavailable, it will only be for a short time and can be easily dealt with						2
		Likelihood					1

Figure 2: Risk 2: Unavailable Member

Risk ID:	Risk Title:	Risk Owner:	Actions taken:				
3	Breaking parts	Eric					
Description:							
Parts unexpectedly break as a result of accidents or improper use			3/28/17: Motor shaft broken, reprinted 3/30/17: Motor encoder magnet flawed, needed realignment Persistent: Careful handling of parts				
Consequences:	Risk Type:		Consequence				
We need to reorder parts, expending extra time and budget	Parts		1	2	3	4	5
							5
Risk Reduction Plan:	Expected Outcome						4
							3
We will practice safe procedures when working with parts and order extras in case	Few parts will break, and even if they do we will have extras on hand				X		2
		Likelihood					1

Figure 3: Risk 3: Breaking Parts

Risk ID:	Risk Title:	Risk Owner:	Actions taken:				
4	Mecanum Drive Too Unstable	Eric					
Description:							
The drive mechanism for the robot proves too be too unstable or unreliable for our purposes			4/1/17: Tested mecanum wheels, no issues present Persistent: Built extra time into schedule				
Consequences:	Risk Type:		Consequence				
We will need to redesign the drive mechanism, expending considerable time and effort	Design flaw		1	2	3	4	5
							5
Risk Reduction Plan:	Expected Outcome						4
							3
We will build enough time in our schedule to deal with it if necessary, and will use suspension	The instability resulting from the wheels will be manageable				X		2
		Likelihood					1

Figure 4: Risk 4: Mecanum Drive Too Unstable

Risk ID:	Risk Title:	Risk Owner:	Actions taken:				
5	Localization not precise enough	Neil					
Description:							
Our localization system is not precise enough to ensure that the drawings are accurate representations of input			4/8/17: Preliminary test of localization, appears robust enough Persistent: None				
Consequences:	Risk Type:		Consequence				
We will need to redesign the localization system or redefine drawing requirements	Design flaw		1	2	3	4	5
							5
Risk Reduction Plan:	Expected Outcome						4
							3
We will test the localization system early on in order to catch any design flaws within the system	Localization will work well enough for our purposes			X			2
		Likelihood					1

Figure 5: Risk 5: Localization not precise enough

Risk ID:	Risk Title:	Risk Owner:	Actions taken:				
6	Unexpected Budget Overruns	Rachel					
Description:							
We unexpectedly run out of budget, because parts cost more than expected or other parties reduce our budget			No issues Persistent: We have a significant buffer in our budget				
Consequences:	Risk Type:		Consequence				
We need to scale down our project, or possibly even acquire funds through other means	Logistical		1	2	3	4	5
							5
Risk Reduction Plan:	Expected Outcome						4
							3
We will leave a significant buffer in our budget in case unexpected situations occur	We will have a large enough buffer that essential components will be acquired			X			2
		Likelihood					1

Figure 6: Risk 6: Unexpected Budget Overruns

### 3 Future Work and Commercialization Considerations

RH: grab image from class

# Appendices

## A Testing and Evaluation Plan

**RH: THIS NEED TO BE REDONE**

### A.1 Design Verification

This section details our test verification plan. This includes the questions and tests we intend to address, and how we plan to measure and achieve success with respect to these tests.

### A.2 Writing Implement

#### A.2.1 Performance Test: Loading

**Test Question:** Is a human operator able to reload the writing tool within the required time limit of 10s, how long does it take?

**Operational Procedure:** With a writing tool already installed in the writing assembly, a human test subject will perform 3 reload tests which are separately timed.

**Metric:** Average duration of the reload time.

**Acceptance Criteria:** The average reload time is under 10s.

**Requirement(s) Verified:** NFR14, FR5, FR6, FR7

#### A.2.2 Performance Test: Writing Quality

**Test Question:** Is the drawing produced by the robot of acceptable quality?

**Operational Procedure:** Using a fully loaded writing tool, the robot attempts to draw along a route with at least 4 ft. of travel distance and 3 turns exceeding 50 degrees. Verify that the resulting drawing is of acceptable quality.

**Metric:** Percent thickness of the route at its narrowest point compared to the maximum thickness of a line created with the writing tool. Boolean on whether or not there are complete breaks in the line.

**Acceptance Criteria:** The percent thickness is at least 70%, and there are no complete breaks in the line.

**Requirement(s) Verified:** NFR6

#### A.2.3 Functional Test: Simultaneous Driving and Writing

**Test Question:** Can the writing tool make a mark while driving?

**Operational Procedure:** With a fully loaded writing implement, the robot will mark a 1 ft. line on the writing surface.

**Metric:** Whether or not any discernible mark is made and the full distance is covered.

**Acceptance Criteria:** A discernible mark must be made and the full distance must be travelled without the robot becoming stuck or breaking.

**Requirement(s) Verified:** NFR6

#### A.2.4 Functional Test: Marking

**Test Question:** Does the writing tool make a mark when pushed down?

**Operational Procedure:** The robot will press down on a writing surface with a fully loaded writing implement. Robot must mark with pressure necessary to mark the surface without damaging the surface or writing tool.

**Metric:** Whether or not a any discerible mark is made.

**Acceptance Criteria:** A discernible mark must be made.

**Requirement(s) Verified:** NFR6



## A.3 Locomotion

### A.3.1 Performance Test: Accuracy

**Test Question:** Is the robot able to drive with positional and rotational accuracy?

**Operational Procedure:** The robot drives along a predetermined testing route consisting of driving forward at least 3 feet and driving to its left at least 1 feet.

**Metric:** The difference of the robot's final position and orientation from the intended position and orientation.

**Acceptance Criteria:** During the forward driving test, the robot's position must be less than 1 inch away from the intended position. During the left driving test, its orientation must be less 10 degrees from the commended perpendicular line. This result must be achieved 90 percent of the time.

**Requirement(s) Verified:** NFR12, NFR13

### A.3.2 Functional Test: Speed

**Test Question:** Can the robot reach a desired speed?

**Operational Procedure:** The robot will drive along a straight line for 5 ft. during a timed trial.

**Metric:** The time required for the robot to reach the end of the line.

**Acceptance Criteria:** The robot must reach the end of the 5 ft. testing course in 20 seconds. This test must be repeatable 90 percent of the time.

**Requirement(s) Verified:** NFR5

### A.3.3 Functional Test: Omnidirectional

**Test Question:** Can the robot drive omnidirectionally?

**Operational Procedure:** The robot will drive along a rectangular path that is 10 inches by 10 inches.

**Metric:** Whether or not the robot can complete the course with only linear motion.

**Acceptance Criteria:** The robot must be able to complete the course with linear motion only. **Requirement(s) Verified:** FR1, FR9

## A.4 Localization

### A.4.1 Performance Test: Robot Position Accuracy

**Test Question:** Is the localization system able to accurately determine the position of each robot?

**Operational Procedure:** Both robots sit stationary within the working bounds. The localization system then attempts to determine their locations.

**Metric:** The difference of the robot's actual position from the position reported by the localization system.

**Acceptance Criteria:** The reported position must be within 1/10 in. of the actual position.

**Requirement(s) Verified:** NF6, FR4

### A.4.2 Performance Test: Bounds Accuracy

**Test Question:** Is the localization system able to accurately determine the boundaries of the workspace?

**Operational Procedure:** The localization system attempts to determine the bounds of the workspace.

**Metric:** The total difference in distance between the reported corners of the workspace and the distance between the actual corners.

**Acceptance Criteria:** The total difference must not exceed 1 in.

**Requirement(s) Verified:** FR4

### A.4.3 Functional Test: Robot Position

**Test Question:** Can the localization system find the robot?

**Operational Procedure:** With a single robot within the working bounds, the localization system attempts to determine the robot's location. Robot operation out of bounds is also considered out of scope, and is undefined behavior.

**Metric:** Whether or not a location is returned by the localization system.

**Acceptance Criteria:** The system must return a location for the robot.

**Requirement(s) Verified:** FR4, NFR6, FR3, NFR13

#### A.4.4 Functional Test: Bounds

**Test Question:** Can the localization system find the working bounds?

**Operational Procedure:** The localization system attempts to find all four corners of the working bounds while they are all in its field of view.

**Metric:** Whether or not locations are returned for all four corners.

**Acceptance Criteria:** The system must return locations for all four corners of the working bounds.

**Requirement(s) Verified:** FR4, NFR6, FR3

### A.5 Input Processing

#### A.5.1 Functional Test: Return Data

**Test Question:** Does the image processor return data usable by the planner?

**Operational Procedure:** A user inputs valid drawings via the UI and the system attempts to generate motion commands.

**Metric:** Whether or not usable motion output is produced.

**Acceptance Criteria:** The system must be able to produce a series of legal motion commands.

**Requirement(s) Verified:** FR11

#### A.5.2 Functional Test: Reject Improper Input

**Test Question:** Is the input processor able to detect and reject improper input from the UI?

**Operational Procedure:** The system takes in an invalid input. For example, lines on top of each other.

**Metric:** Whether or not input is rejected.

**Acceptance Criteria:** The invalid input must be rejected.

**Requirement(s) Verified:** FR4, FR11

### A.6 Work Scheduling, Distribution and Planning

#### A.6.1 Performance Test: Executable Plans

**Test Question:** How consistent is the planner at generating executable plans, ie those that avoid collision and stay within bounds?

**Operational Procedure:** Using the example input set (Appendix C), run each input and check the plan for potential robot-robot collisions and out of bounds driving.

**Metric:** Ratio of number of unacceptable plans, those that would involve collision or driving out of bounds, over the total number of plans.

**Acceptance Criteria:** Almost all, 99% of plans would not involve collision or out-of-bounds if executed.

**Requirement(s) Verified:** FR4, NFR11

#### A.6.2 Performance Test: Execution Distribution

**Test Question:** How efficiently is execution time, i.e. the total time robots spend moving, distributed?

**Operational Procedure:** Using the example input set (Appendix C), run each input and record the total time each robot spends moving.

**Metric:** We define execution efficiency as  $\frac{\min(execution(R_0), execution(R_1))}{\max(execution(R_0), execution(R_1))}$  where  $execution(R_0)$  refers to the execution time of robot 0 and  $execution(R_1)$  refers to the execution time of robot 1

**Acceptance Criteria:** Execution efficiency of 0.75.

**Requirement(s) Verified:** FR8, NFR5

#### A.6.3 Performance Test: Drawing Distribution

**Test Question:** How efficiently is drawing time, i.e. the total time robots spend drawing, distributed?

**Operational Procedure:** Using the example input set (Appendix C), run each input and record the total time each robot spends drawing.

**Metric:** We define drawing efficiency as  $\frac{\min(\text{draw}(R_0), \text{draw}(R_1))}{\max(\text{draw}(R_0), \text{draw}(R_1))}$  where  $\text{draw}(R_0)$  refers to the drawing time of robot 0 and  $\text{draw}(R_1)$  refers to the drawing time of robot 1

**Acceptance Criteria:** Drawing efficiency of 0.75.

**Requirement(s) Verified:** FR8, NFR5

#### A.6.4 Performance Test: Speedup

**Test Question:** What speedup is achieved by using two robots instead of one?

**Operational Procedure:** Using the example input set (Appendix C), run each input first with one robot and then with two. Time the execution time of each variant.

**Metric:** The comparison of duration, i.e.  $\frac{\text{executiontimewith2robots}}{\text{executiontimewith1robot}}$ .

**Acceptance Criteria:** According to our requirements we expect a speedup of 2x.

**Requirement(s) Verified:** NFR5

#### A.6.5 Functional Test: Collision Free

**Test Question:** Does the planner and executor generate collision free plans?

**Operational Procedure:** Using the example input set (Appendix C), run each input and check for any robot-robot collisions during execution.

**Metric:** Boolean across each plans on whether a collision occurred.

**Acceptance Criteria:** We only accept if collisions were avoided on 95% of our test cases.

**Requirement(s) Verified:** NFR 11

#### A.6.6 Functional Test: Autonomy

**Test Question:** Does the system require no user input beyond adding the image to be drawn (except for error handling)?

**Operational Procedure:** After having input a plan, press "Run" on the system and observe if the system requires user input to finish the drawing.

**Metric:** Boolean on whether user input was required, excluding input relating to errors.

**Acceptance Criteria:** Accept only if no input was required.

**Requirement(s) Verified:** FR 2

### A.7 Communication

#### A.7.1 Performance Test: Uptime

**Test Question:** What is the uptime on our ability to communicate data between the robots and the offboard system?

**Operational Procedure:** Run the system for a significant period of time (several hours) and record any communication downtime or data loss during communication.

**Metric:** Time duration of down communication and packet loss.

**Acceptance Criteria:** Operational 95% of the time.

**Requirement(s) Verified:** FR12, NFR8, NFR9

#### A.7.2 Functional Test: Sending and Receiving Data

**Test Question:** Can the robot send and receive data to and from the offboard device and can the offboard device send and receive data to and from the robot?

**Operational Procedure:** Send data from the off-board device to the robot and verify the robot received it. Send data from the robot to the off-board device and verify the off-board device received it.

**Metric:** Four booleans on whether the data is successfully sent and recieved on both ends.

**Acceptance Criteria:** We must succeed on all four accounts.

**Requirement(s) Verified:** NFR9

#### A.7.3 Functional Test: Data Parsing

**Test Question:** Can the data on each side (robot, offboard device) be parsed by each system?

**Operational Procedure:** Send data from the offboard device to the robot and verify the robot received it and can execute and process the appropriate information. Send data from the robot to the offboard

system and verify the offboard device received and can respond to it.

**Metric:** Check whether the data was successfully parsed on all sides.

**Acceptance Criteria:** We require all data be parsable.

**Requirement(s) Verified:** NFR9

## A.8 User Interface

### A.8.1 Performance Test: Emergency Stop Speed

**Test Question:** How fast does the emergency stop shut down the system?

**Operational Procedure:** While the system is in use, press the emergency stop button and time how long it takes for everything to completely shut down.

**Metric:** Elapsed time.

**Acceptance Criteria:** It is vital to safety that our emergency stop shuts everything down within a second.

**Requirement(s) Verified:** FR13, NFR11

### A.8.2 Performance Test: Error Reporting Delay

**Test Question:** What is the delay between an error occurring and that error being reported to the user?

**Operational Procedure:** Given a list of known operational errors, intentionally trigger each error within the system and report the time between causing the error and it being reported to the user.

**Metric:** Averaged elapsed time across error reporting.

**Acceptance Criteria:** The average time to detect and report an error should be within 3 seconds.

**Requirement(s) Verified:** NFR2, NFR11

### A.8.3 Performance Test: Error Understandability

**Test Question:** How understandable and informative are error messages?

**Operational Procedure:** Given a list of known operational errors, intentionally trigger each error while a non-developer user is using the system (while masking the error cause) and evaluate how well the user can determine the error. For example, while the system is drawing, the user could be in a different room with only the error reporting device, making the user unable to see what errors the robots are facing.

**Metric:** Determine if the user can determine the error and knows how to react to or correct the error.

**Acceptance Criteria:** The user should be able to determine and react effectively for 90% of the errors.

**Requirement(s) Verified:** NFR1, NFR2, FR14

### A.8.4 Functional Test: Emergency Stop

**Test Question:** Does the emergency stop fully stop the system?

**Operational Procedure:** While the system is in use, press the emergency stop button and check if all systems halt their operation.

**Metric:** Boolean on whether every subsystem stops or not.

**Acceptance Criteria:** It is only successful if the boolean metric is true.

**Requirement(s) Verified:** FR13

### A.8.5 Functional Test: Error Reporting

**Test Question:** Is each operational error reported to the user?

**Operational Procedure:** Given a list of known operational errors, intentionally trigger each error within the system and report whether the error caused it reported to the user.

**Metric:** Each error must be reported correctly. Hence we can divide the number of correctly reported errors by the number of total errors caused to determine an error-reporting score.

**Acceptance Criteria:** Considering error handling is critical to performance, our system should have an error-reporting score of 90%.

**Requirement(s) Verified:** NFR1, NFR2, FR14

## A.9 Power System

### A.9.1 Performance Test: Battery Life

**Test Question:** How long can an individual robot run for on a single battery charge?

**Operational Procedure:** Charge a robot fully. Command the robot to complete an input drawing repeatedly until the robot is fully drained of power. Time how long this takes.

**Metric:** The duration of operational time given one charge

**Acceptance Criteria:** We accept this if the operational time exceeds the necessary duration time of 90% of our test drawing inputs.

**Requirement(s) Verified:** NFR7

## A.10 Full System Validation

## B Full System Validation

### B.1 Performance Test: Painting Accuracy

**Test Question:** How closely does the drawn image resemble the original input?

**Operational Procedure:** Using example input sets for the system to complete. After completion, overlap the input with the image of final drawing captured from overhead camera. Rescale the two images so that they are in the same size. Evaluate the coherence of the two images.

**Metric:** The percentage of drawn lines that were within 3 pixels of difference compared to those of the original image.

**Acceptance Criteria:** The system must successfully and accurately draw 95% of the lines in the original input.

**Requirement(s) Verified:** NFR6

### B.2 Performance Test: Reliability

**Test Question:** How reliable is the system in terms of successfully complete a series of drawing tasks?

**Operational Procedure:** Command the system to finish a series of drawing tasks. Measure the number of consecutive successful completion. Successful completion is defined as the system autonomously finishes painting and the painting process is free of errors including but not limited to localization breakdown, motor breakdown, or painting mechanism breakdown. Calling human interference with switching battery and drawing utility does not count as unsuccessful run.

**Metric:** Number of consecutive painting completion.

**Acceptance Criteria:** The minimum acceptable number of consecutive completion is 5.

**Requirement(s) Verified:** NFR8

### B.3 Functional Test: Size

**Test Question:** Is the robot agent too big to be portable, i.e. carry the robot through a standard door?

**Operational Procedure:** Measure the physical dimensions of the robot in terms of width, length, and height or in terms of diameter and height.

**Metric:** Numeric value of each length measurement; robot footprint; robot volume.

**Acceptance Criteria:** Must be less than 80 in. x 36 in. x 36 in.

**Requirement(s) Verified:** NFR4

### B.4 Functional Test: Weight

**Test Question:** Is the robot agent too heavy to be portable, i.e. able to be lifted by a normal person?

**Operational Procedure:** Measure the mass of the robot.

**Metric:** Numeric value of robot mass.

**Acceptance Criteria:** Must be less than 50 pounds.

**Requirement(s) Verified:** NFR3

## B.5 Functional Test: Budget

**Test Question:** Does the cost for developing this robotic system exceed our budget?

**Operational Procedure:** Document total amount of money spent for designing and constructing this robot system. This includes machining expense, part cost, and etc.

**Metric:** Total amount of money spent.

**Acceptance Criteria:** Total developing expense has to be less than \$2500.

**Requirement(s) Verified:** NFR10

## B.6 Functional Test: Safety

**Test Question:** Is the robot safe during operation? Specifically, when collision happens, will the robot harm other robots, external environment, or human?

**Operational Procedure:** Count the number of sharp edges on the exterior of the robot. Also, measure the time it takes from the overhead camera detects collision to robot agent stops moving motors. Intermediate steps involved are: camera sends collision signal to system controller and system controller commands involved robot agent to stop its current action.

**Metric:** Number of sharp edges (angles less than 90 degrees); amount of time takes from detection to action.

**Acceptance Criteria:** Values for these two metrics need to be as small as possible. Zero sharp edges can be on the exterior - any edges from, for example, a rectangular chassis, should be rounded. The maximum amount of time is 1.5 seconds.

**Requirement(s) Verified:** NFR11

## B.7 Functional Test: Documentation

**Test Question:** Is the documentation for the developing process comprehensive and replicable?

**Operational Procedure:** Give the full documentation to another design group or stakeholder and inquiry if they can duplicate the project with those documents.

**Metric:** Boolean on whether or not reviewers can replicate the system development process.

**Acceptance Criteria:** Reviewers are confident to replicate system development process based on the documentation.

**Requirement(s) Verified:** NFR1

# C Planner Inputs

The following are a set of sample drawing inputs that will be used to test the system. Some test inputs have been randomly generated while others were designed to stress test a particular feature.

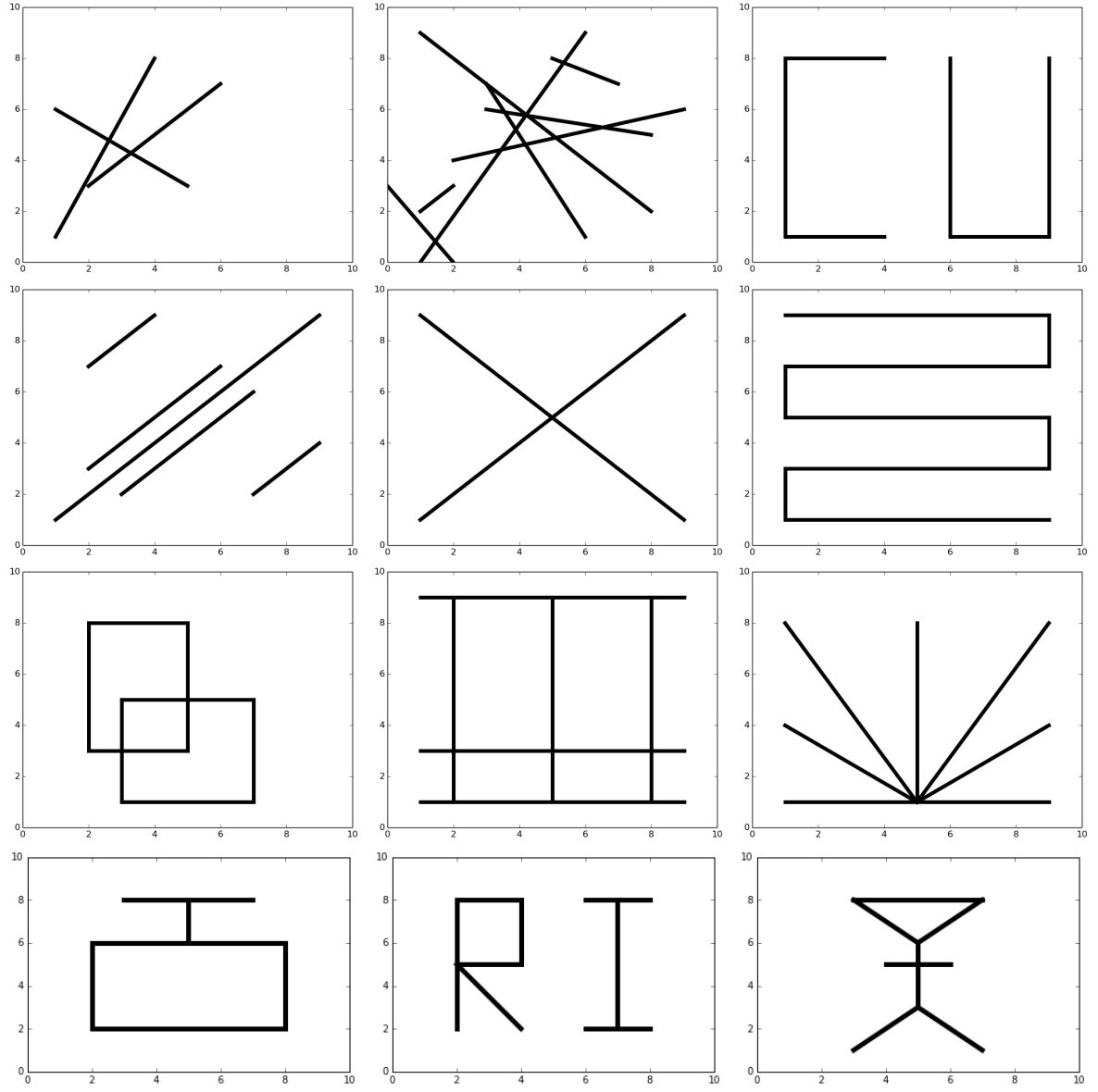


Figure 7: Planner Test Cases.