Test Scenario Number:	002	**Tested By:**	Ryan Bomalaski
Sprint Number:	1	**Application:**	main.py
Tracker ID:	ST-002	**Time Estimation:**	30 Minutes
Module:	N/A	**Type:**	Stepwise

Test Scenario and Requirements Description: This is an expansion of ST-001. After setting up the initial simulator, this will continue and test the simulator.

Prerequisites:

- User has Collision Avoidance folder
- User has Python 3.X installed with stock IDLE 3 IDE
- User has SQLite3 Installed
- User has run scenario test ST-001

Scenario Title: Run Simulator for 40 Steps

Scenario Procedure:

Using the provided scripts, the user will import the test airplanes to the python algorithm. Then the user will run the simulator for 40 steps.

Scenario Steps:	Validation:
Create Airplane Test Database: 1. Open New Terminal 2. Navigate to …/collision_avoidance/src/python 3. Run command: 1. sqlite3 airwaves.db	The SQLite program will start in the terminal, opening up the airwaves.db. If no database exists, it will create it.
Implement Starting Data: 1. While in SQLite3 run the command: 1. .read db_update.sql 2. While in SQLite3 run the command: 1. .schema 3. While in SQLite3 run the command: 1. **.exit**	The schema for the tables airwaves and stage should appear. Then the exit command will bring the user back to the linux terminal.* * - Note: If this is the first set up of the table, two errors will appear with the db_update.sql script
Open main.py in terminal: 1. In the same terminal as above, run the following: 1. python3 -i main.py	Will open the python terminal (Denoted with the “>>>”).
Create Simulator object and populate with Airplanes: 1. Create a new simulator object with step count of 40 by typing the following command: 1. sim = Simulator(40) 2. Populate the simulator with aircraft by running: 1. sim.createAirplanes() 3. Confirm that two airplanes were created	A list of two airplane objects with the address in memory will appear.

by running: 1. sim.airplanes	
Run Simulator: 1. In python environment, run the following command: 1. sim.run_sim() 2. When the simulator is complete, run: 1. exit()	The simulator will step through 40 steps, giving outputs for both airplanes. Upon exit, the user will be back at the linux terminal.