NOTEBOOK ANALYSIS:

1) **Obstacle and Rock Sample Identification**
   To identify rocks and obstacles, I modify the given color_thresh function. I added a parameter called text which can take 3 strings: terrain, obstacles and rocks. In the function I added some if statements to define threshold for obstacles and rocks.
   For Rocks: (110,110,5) – (255,255,90)
   For obstacles: < (100,100,100)
2) **Process_image function modification**
   - Source and destination points are defined which are required for perspective transform
   - Applying perspective transform
   - Calling color_thresh function for detecting navigable terrain, obstacles and rocks
   - Converting to rover centric coordinates
   - Converting to world coordinated
   - World map is updated by adding obstacles , terrain and rocks pixels to different plane
   - Finally, output image consists of image, warped image, threshed image(Green for terrain and Red for Obstacles),and world map

AUTONOMOUS NAVIGATION AND MAPPING:

1) **Perception and Decision Step Modifications:**
   Perception step is quite similar to process_image function of jupyter notebook. Only Rover parameters are used instead of using parameters from .csv file created after training data. I also added two parameters for Rock picking – Rover.rock_angle and Rover.rock_dist which is quite similar to nav_angles and nav_dists. These parameters are also being updated in Perception step.

   In Decision Step, I included a separate function for picking of rocks. When a rock is in sight, using the mean of the rock_angle parameter it will go straight toward the rock. For navigation and mapping purposes, I added a condition of checking Rover.nav_dists. Then, I used the indexes of it at which the value is less than 120 to find mean of Nav_angles. I am only considering nearby points and then taking mean of those angles.

2) **Results explaination:**
   FPS: 29
   Graphics Quality: Good
   Resolution: 1366x768
   Fidelity is mostly around 65% with area coverage 40%. It drops to 40-50% when coverage is 80%. I might improve decision code by incorporating algorithms so that rover don’t trace its path again once it already mapped.