**Image Selection**

The first part of the project is to select part of an Image so that it can be refined and placed into another image. From our initial research creating a UI in MATLAB was not feasible for what we wanted in our application so we created a user interface using java. The code for the java program can be found in Paint.java and DrawArea.java. Paint.java is the main application that runs and DrawArea.java is a helper class. The overall application is around 500 lines of code and uses the java swing library for creating the user interface. The program starts by prompting the user to input two images. The first image will be the one you select from and the second will be the image that you place into. This can be seen in figure 1 below.

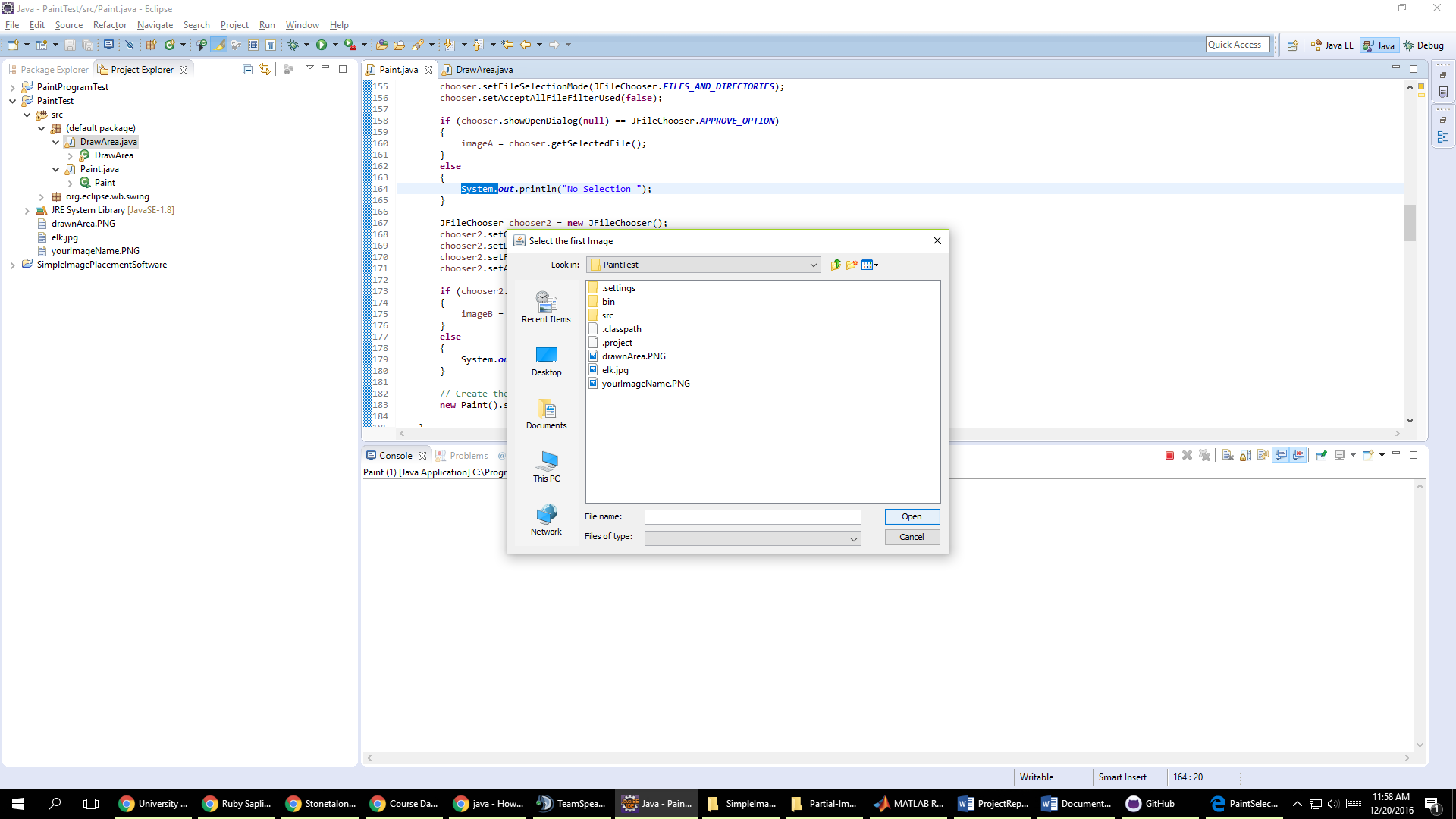


Figure 1: Selecting an image

After the user selects the two images that they want to use a new window pops up for the main user interface as shown in figure 2. This interface is how the user will select portions of the image. Due to difficulty and complications we were not able to integrate the java application directly into the MATLAB functions. The java application does output all the needed information for the MATLAB functions through the command line.

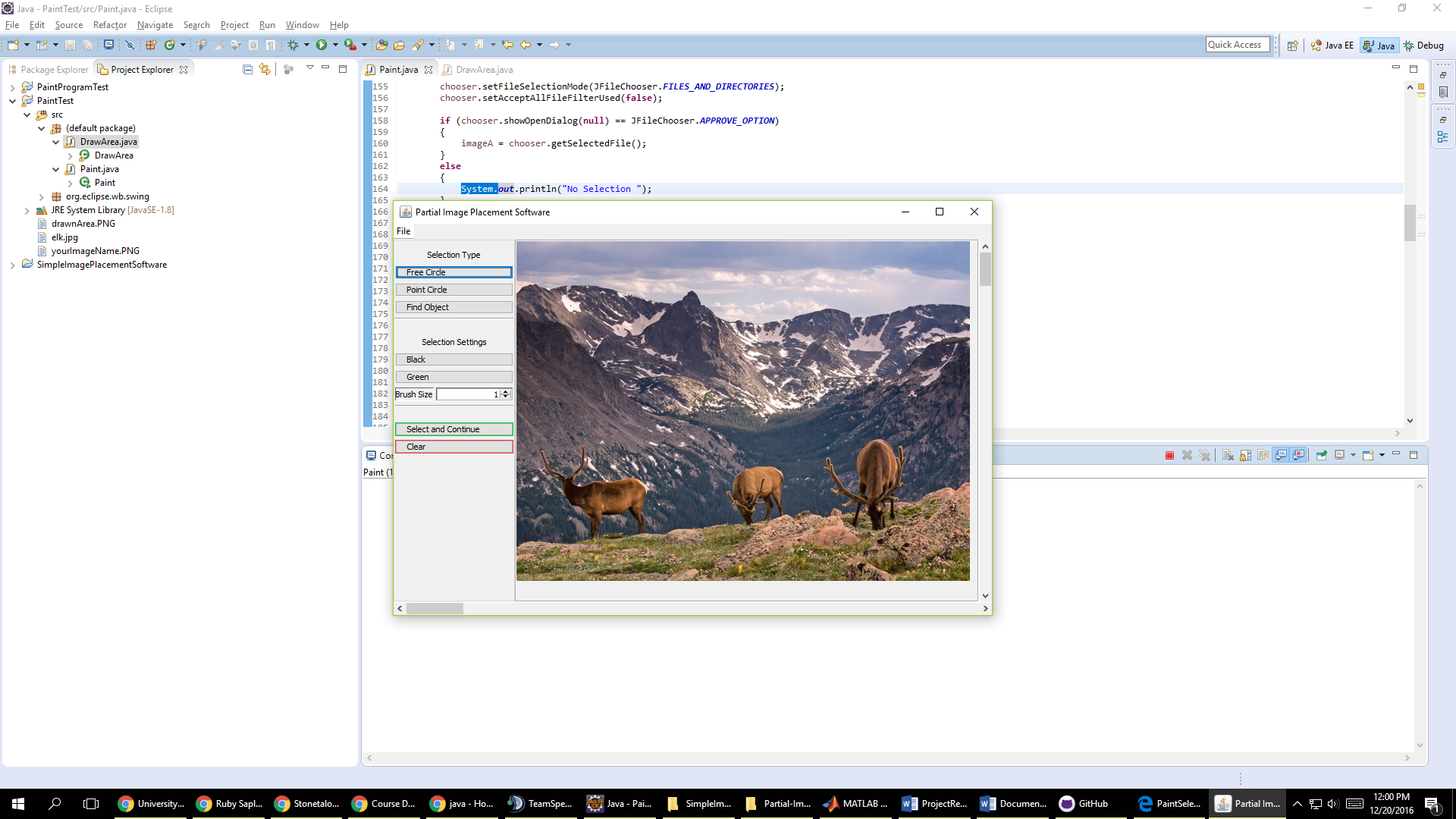


Figure 2: The main user interface

Our initial goals for image selection were to implement lazy snapping or a paint selection method. Lazy snapping, as outlined in a paper by Li, Yin, et al called Lazy Snapping published in ACM Transactions on Graphics, allows the user to draw a line around an object and have the line snap to the object. The second method we wanted to implement was paint selection, as outlined in a paper by Jiangyu Liu, et al, which allows a user to draw inside of objects that they wanted to select. However due to time constraints and difficulties in working with images in java we moved to allowing the user to select an area with a box that will later be refined by the automatic cut out algorithm. The box can be seen in figure 3 and the ability to freely draw around images can be seen in figure 4.

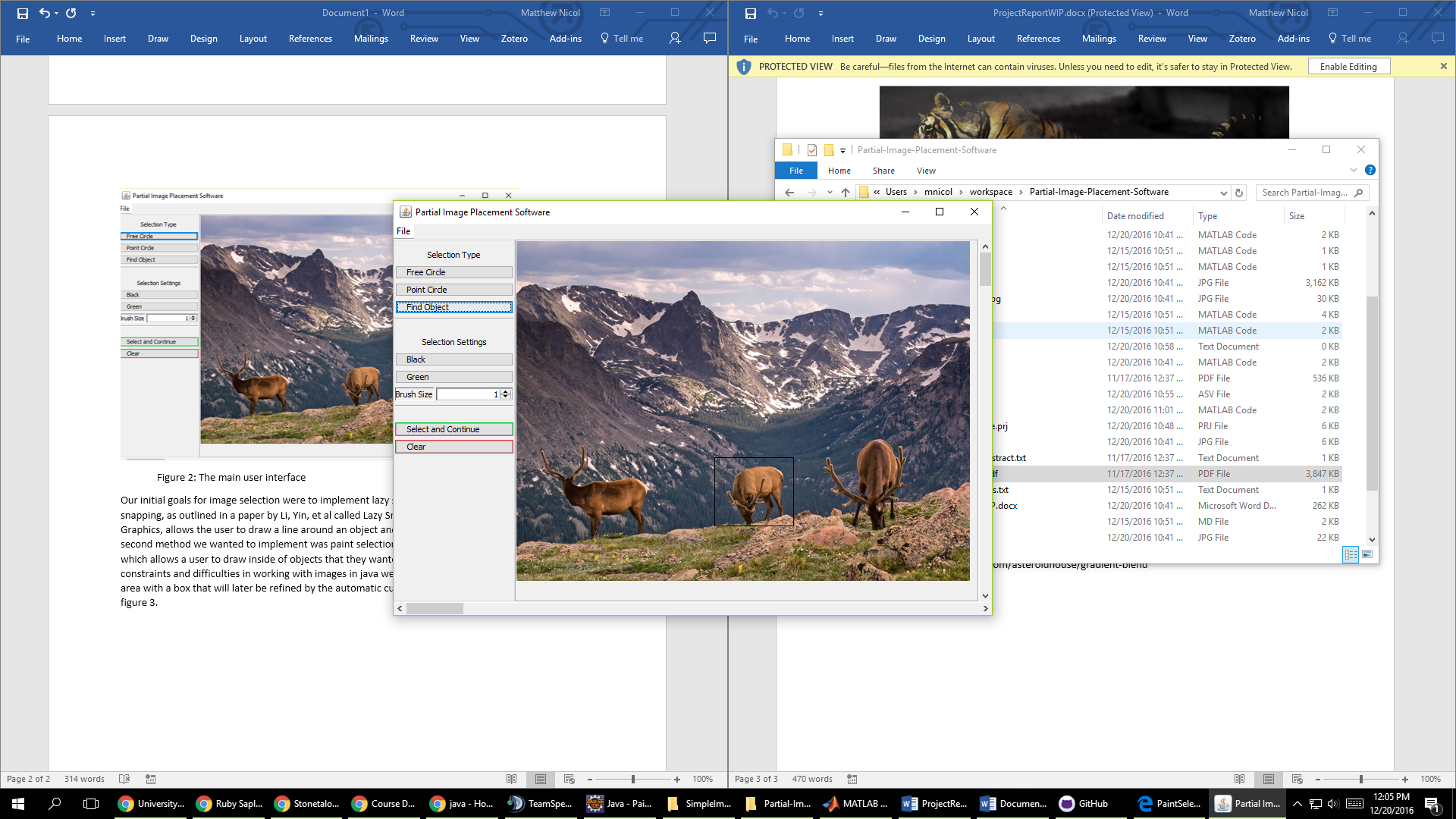


Figure 3: User placed box around an object

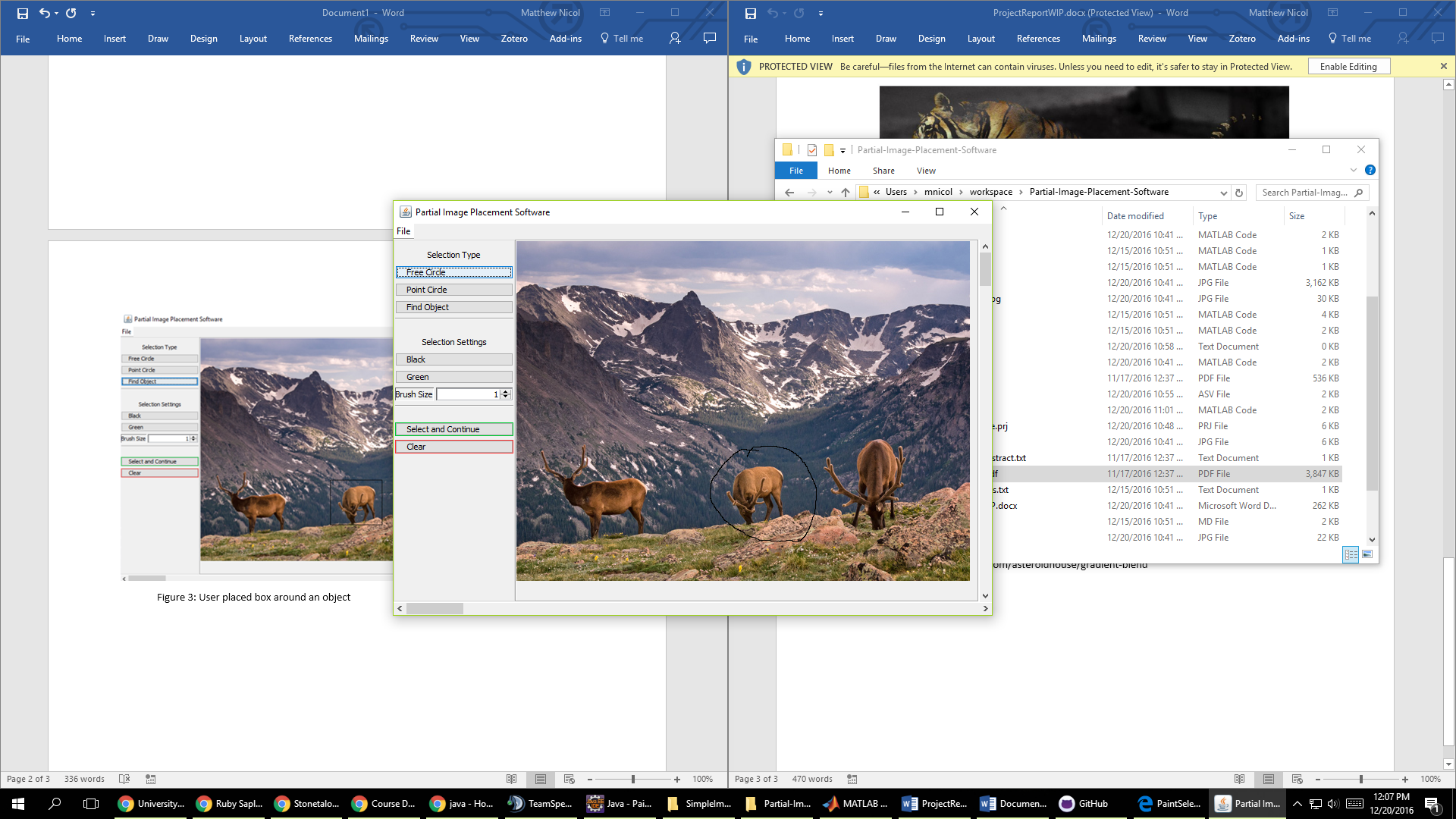


Figure 4: Freely drawn circle around object

The resulting information for these actions can be found as outputs in the command line as shown in figure 5. The information that is output is where the user clicked which will be needed for placing the image, and the upper left corner and bottom right corner of the rectangle to select the area for cutting. You can also save the drawn area of the image with a blank background to be processed later as seen in figure 6.

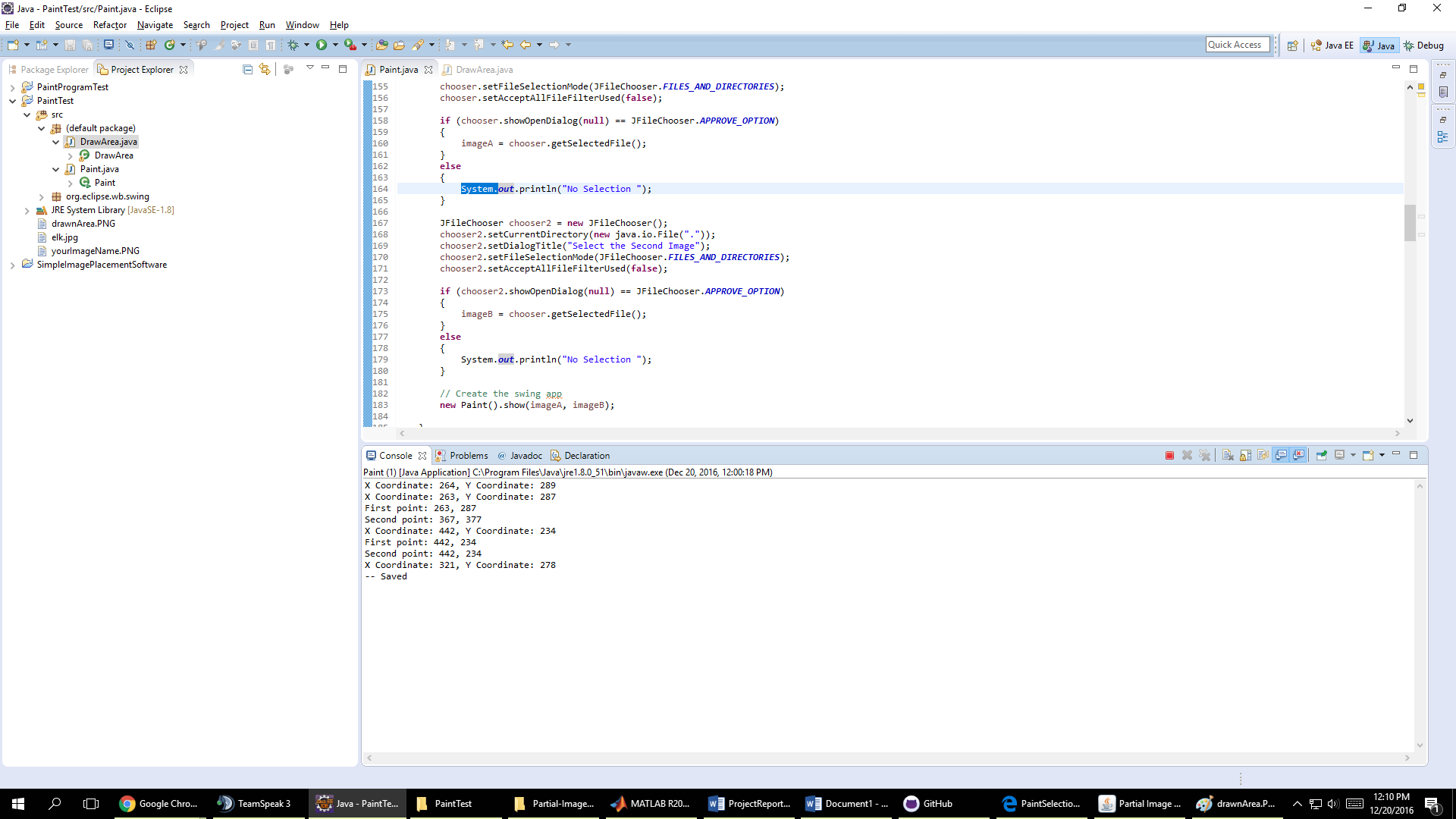


Figure 5: The coordinate outputs from using the program

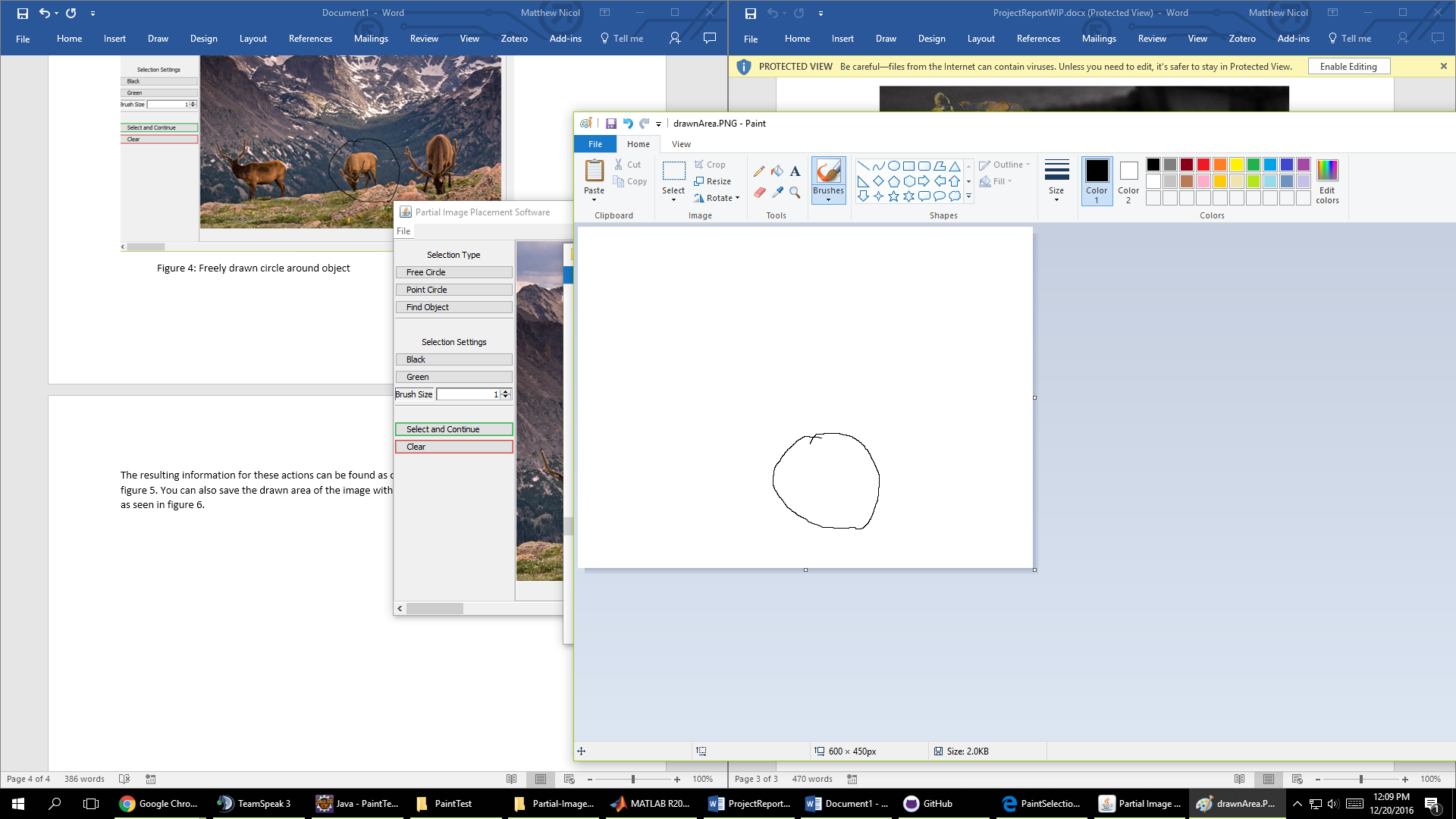


Figure 6: The drawn area from figure 4