**Problem Statement**

I created a clipping window, where polygons that are drawn are clipped to the window. A viewport can be drawn by selecting two diagonal points of a rectangle, and then using a menu option the polygon within the clipper window can be displayed in the viewport. The viewport can be scaled using the bottom-right corner of the viewport as a reference, dragging the viewport and scaling the objects inside. Likewise, the same can be accomplished with the clipper window, achieving the zooming effect. Furthermore, the clipper window can be translated, achieving the panning effect in the viewport.

**Algorithm Design**

Polygons are drawn using a standard implementation of OpenGL and Glut. The polygon is “closed” by pressing the ‘P’ button. The polygon is clipped over the displayed clipping region using the Sutherland-Hodgman algorithm, modified to work with vectors of tuples, which is how I chose to contain the points of the polygon. The polygon can be filled in using the boundary fill algorithm. When you right-click on an area you want to fill and select the fill option, the boundary fill algorithm first creates a screenxXscreeny array, where each of the values in the vector contains a Color struct, which contains RGB values. These values are updated recursively according to the algorithm, with the first point being taken from the location where the menu was brought up. The polygon fills in red. The viewport can be created by pressing ‘O’ and then clicking on two points on the screen, which sets the two diagonal points of the viewport. Then, by a menu option, the polygon in the clipper is clipped and then transferred pointwise to the viewport. When the viewport is expanded, each of the values in the viewport are multiplied proportionally to the change in the size of the viewport, while when the clipper is changed in size, the values inside the clipper are remapped to the viewport each time. Same is done when the clipper is translated.

**How to Run the Code**

DISCLAIMER: The code has a few bugs. If you follow the instructions, it will run, and all of the parts of the assignment will be shown. Trying to test for out-of-bounds cases etc. will cause the program to crash.

--------------

I recommend trying to fill the polygon and then restarting the program for the other parts, because the fill algorithm can often crash the program.

--------------

To draw a polygon: Left-click to add a vertex to the polygon. When you are done, press ‘P’ to close the polygon.

To clip the polygon: Right-click anywhere and click the clip button

To fill the polygon: Right-click INSIDE THE POLYGON and select the fill option. If you click outside the polygon, it will try to fill the whole screen and the stack will overflow. *It takes about 10 to 15 seconds to fill the polygon*.

To create a viewport: Press ‘O’, and then click two points near the center of the screen. *If you try to draw the viewport too close to the edge of the screen, the array will go out-of-bounds and the viewport values will not work.*

To display polygons in the viewport: Right-click anywhere, and select the window-to-viewport option

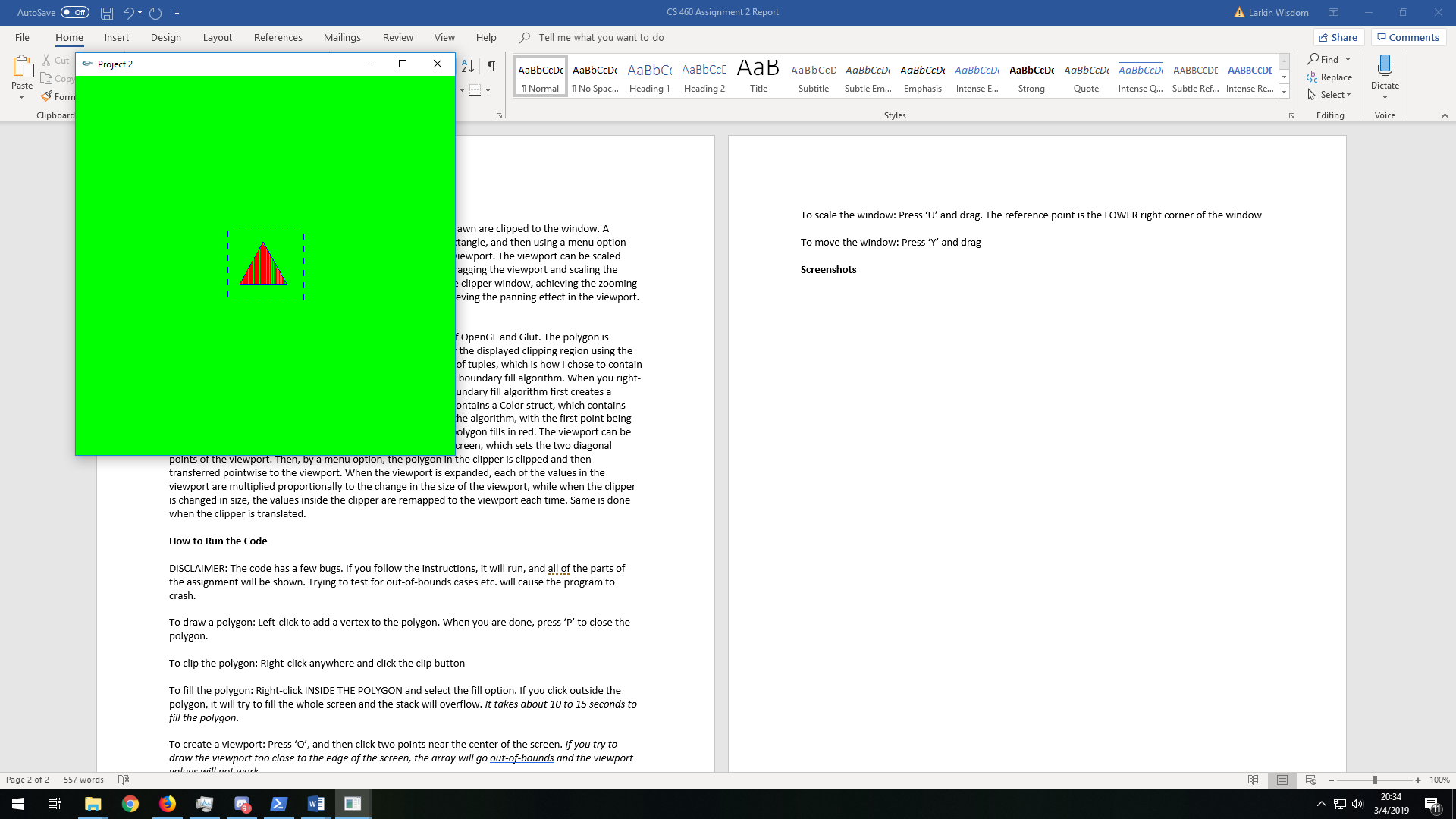
To scale the viewport: Press ‘I’ and drag along the screen. The reference point is the LOWER right corner of the viewport.

To scale the window: Press ‘U’ and drag. The reference point is the LOWER right corner of the window

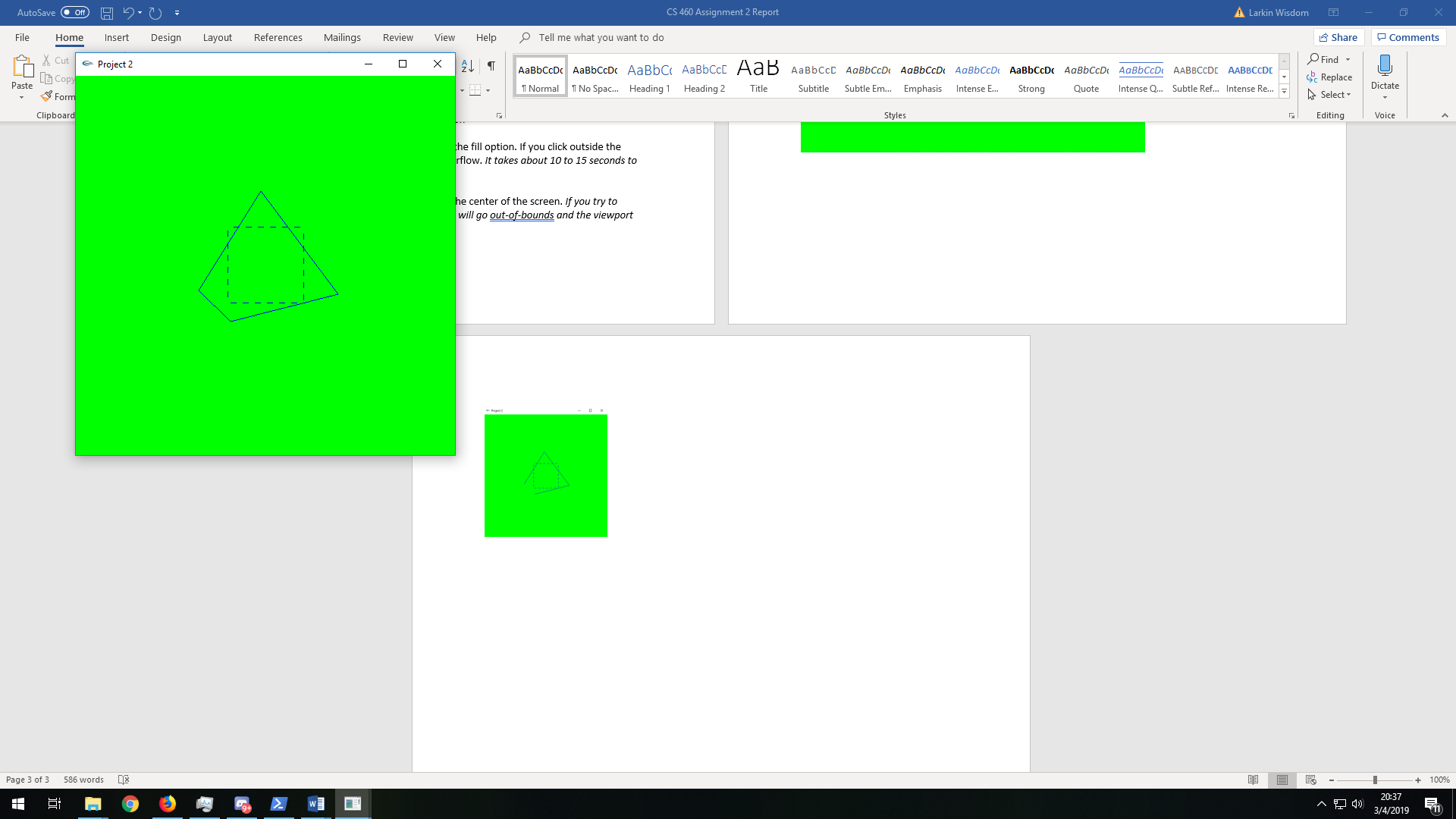
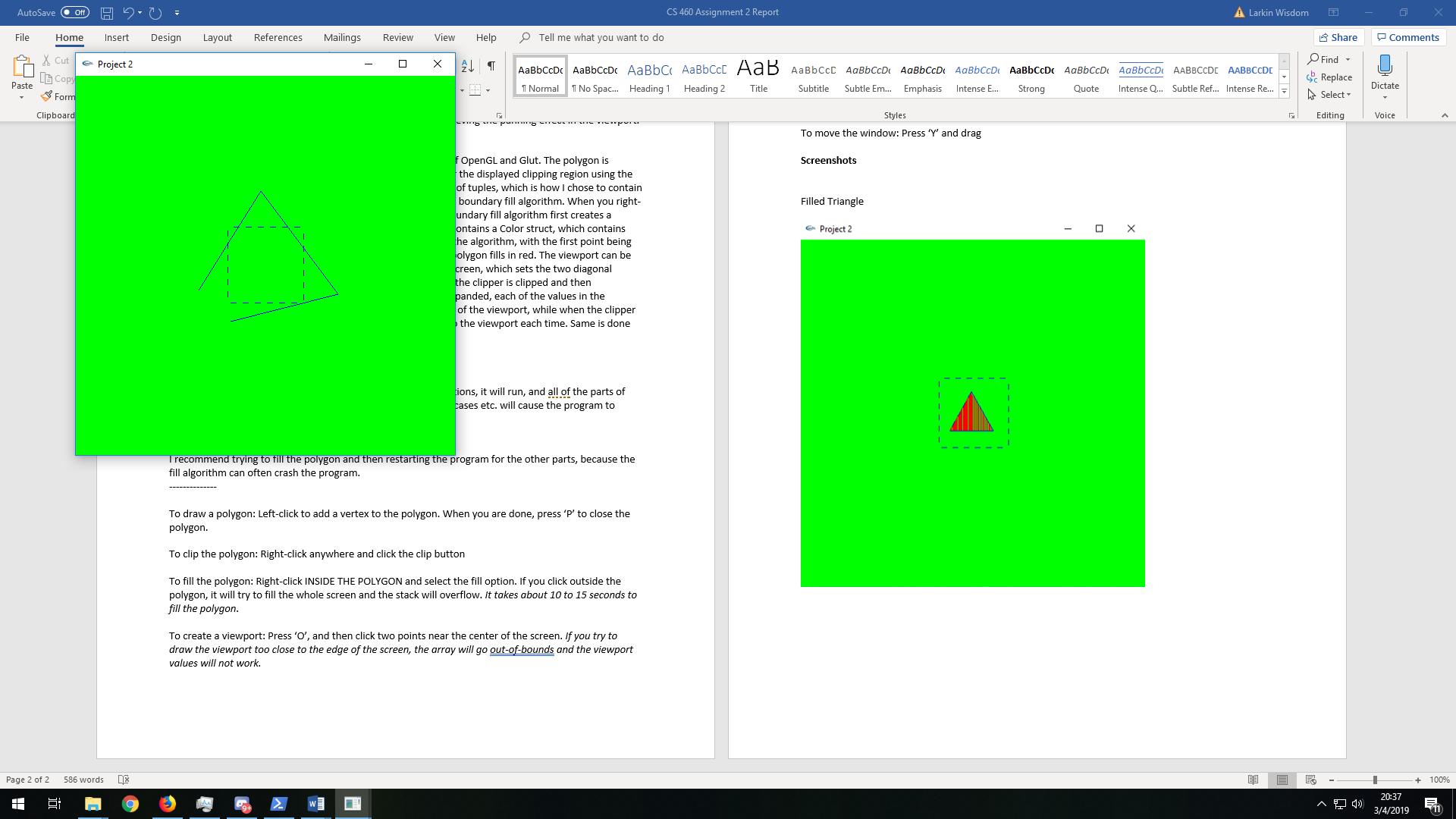
To move the window: Press ‘Y’ and drag

**Screenshots**

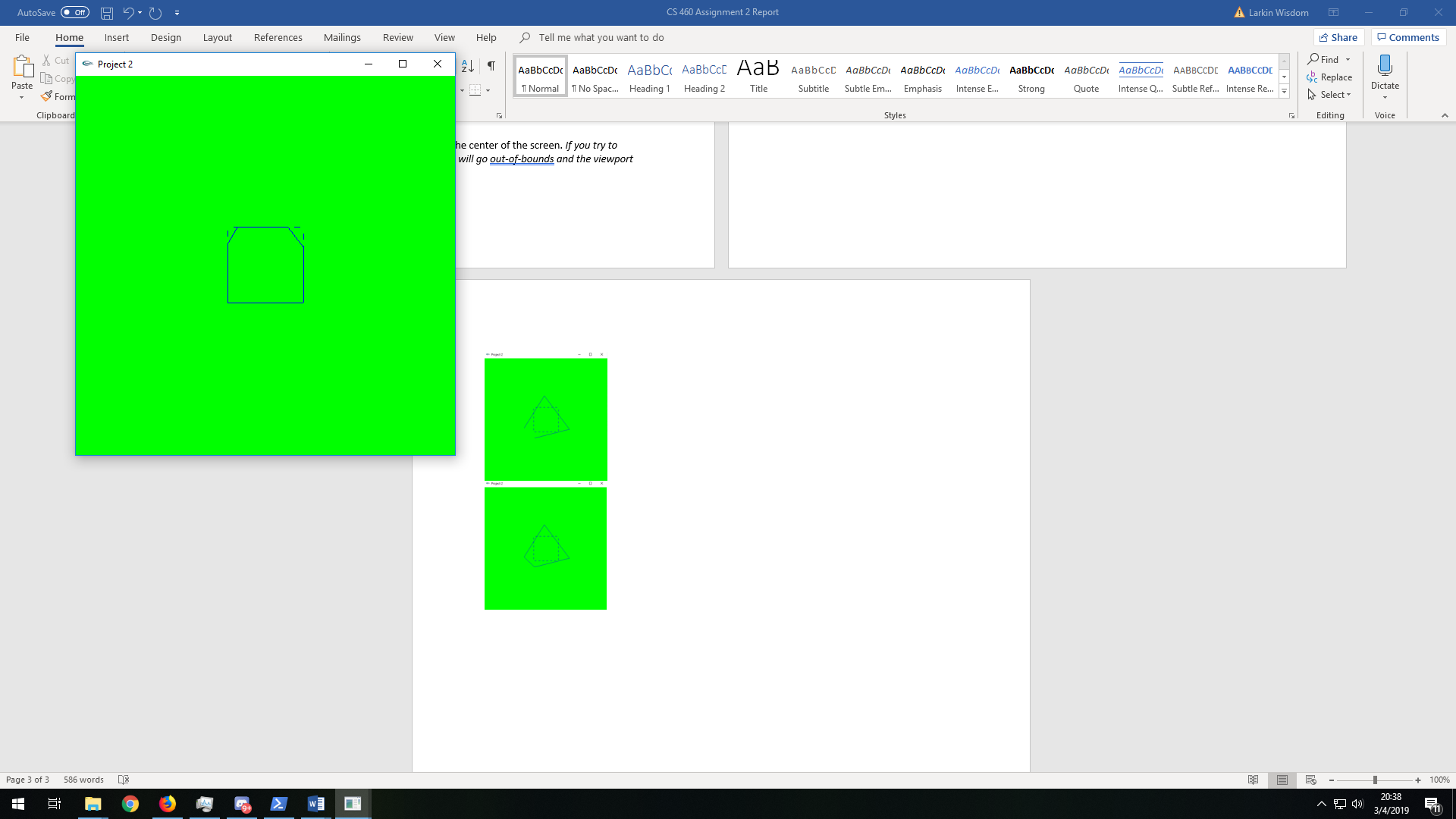
Filled Triangle



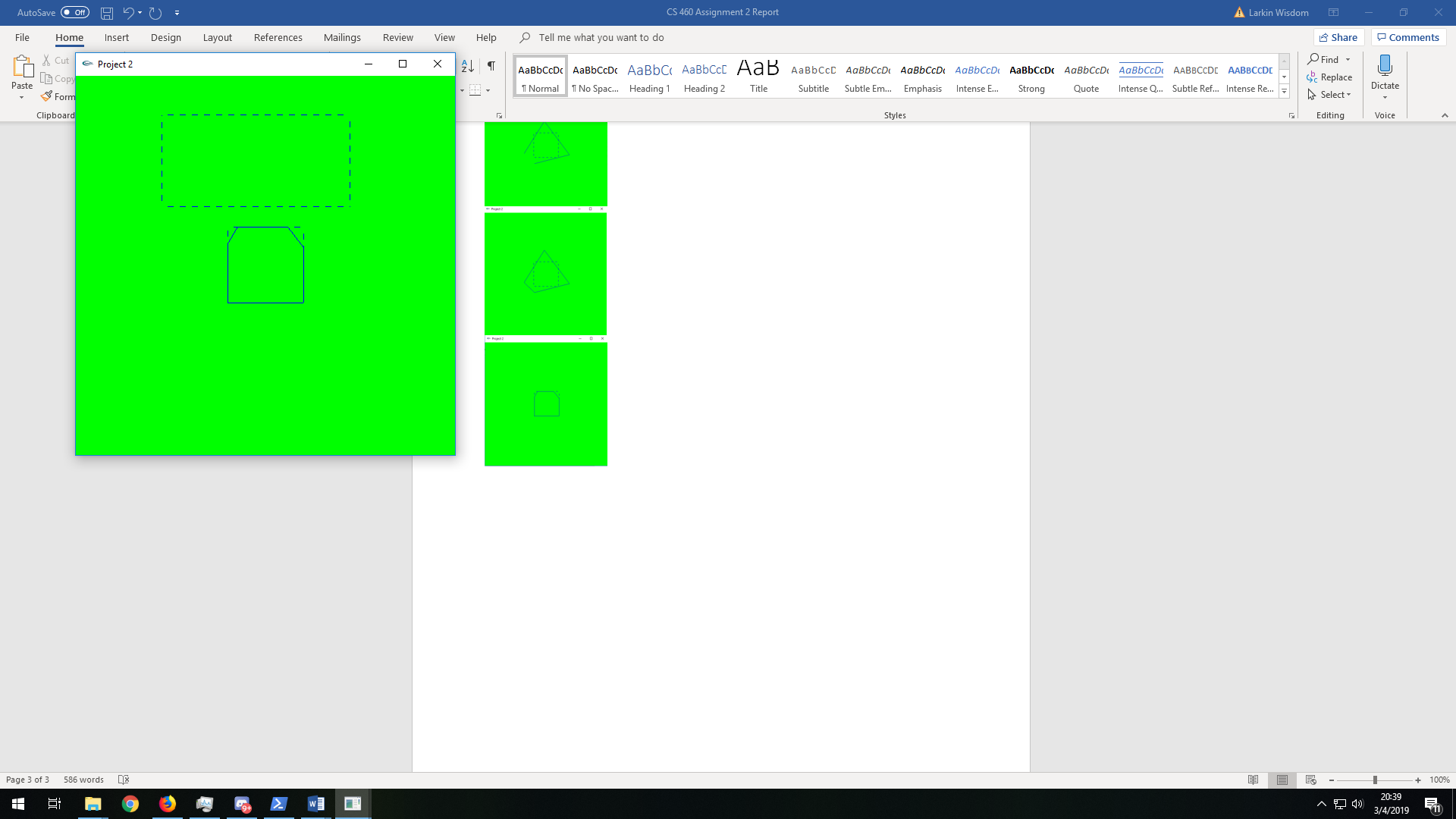
Drawing three sides of a polygon, and then closing it with ‘P’



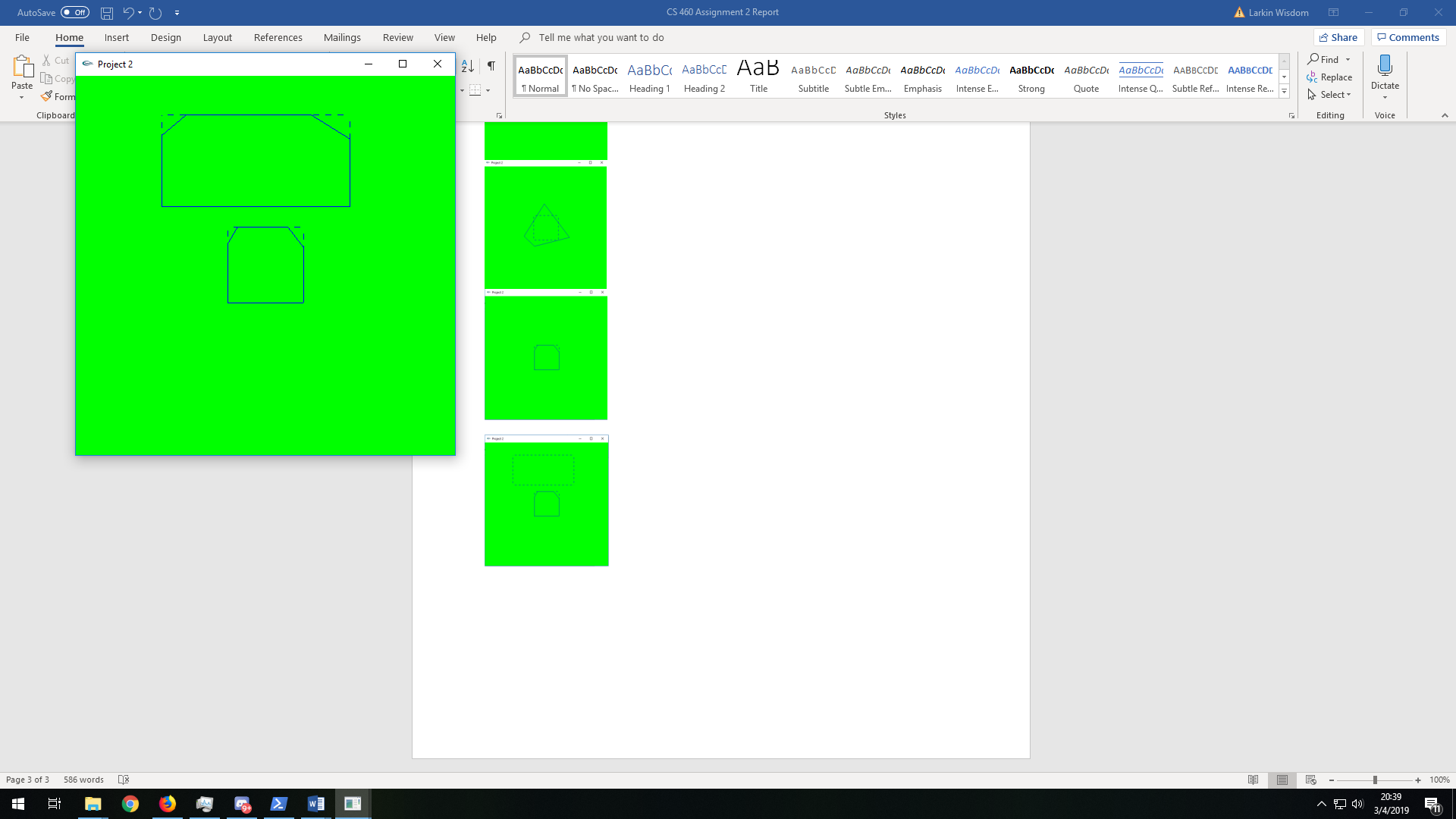
Clipping the polygon



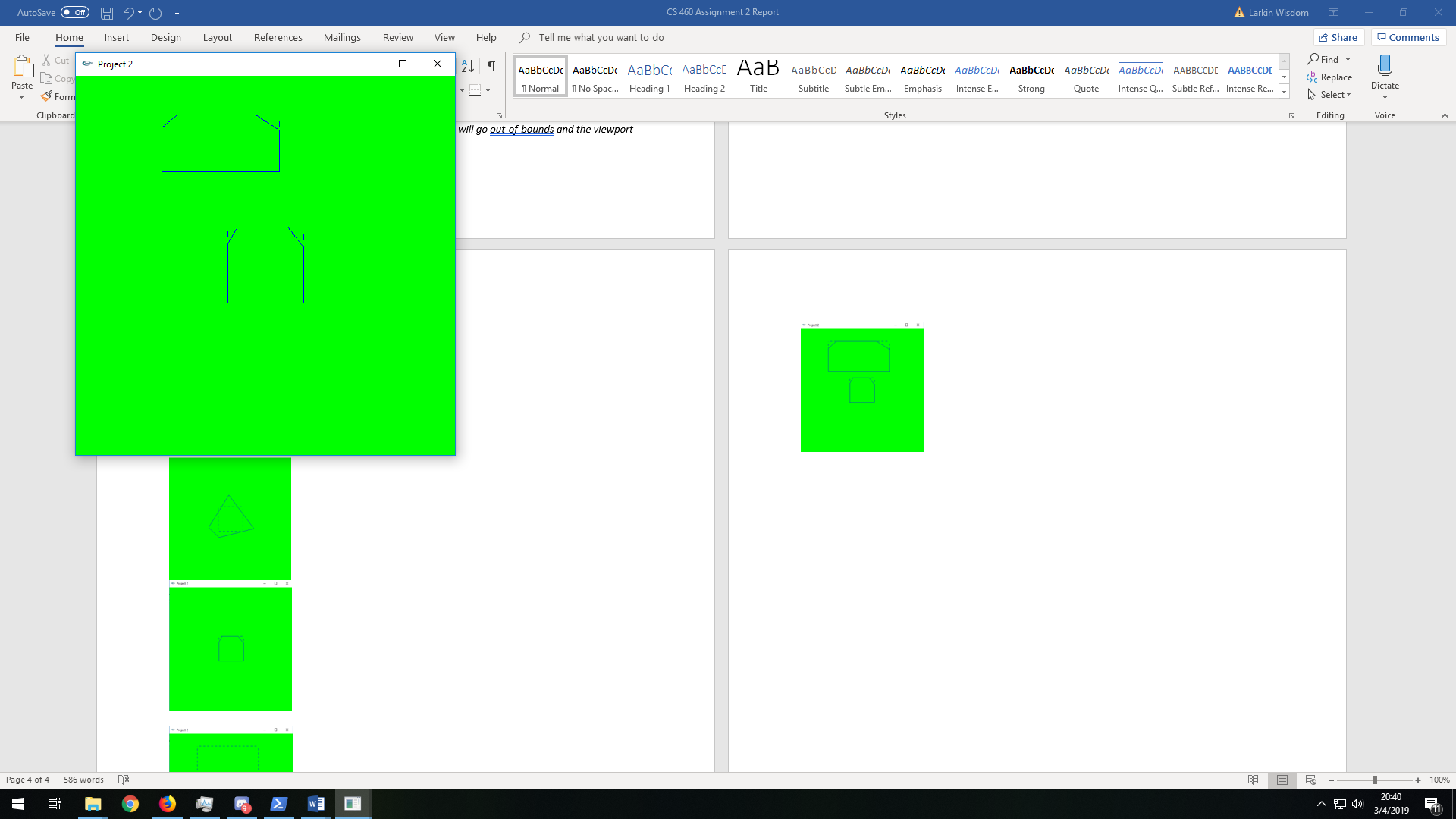
Drawing the viewport



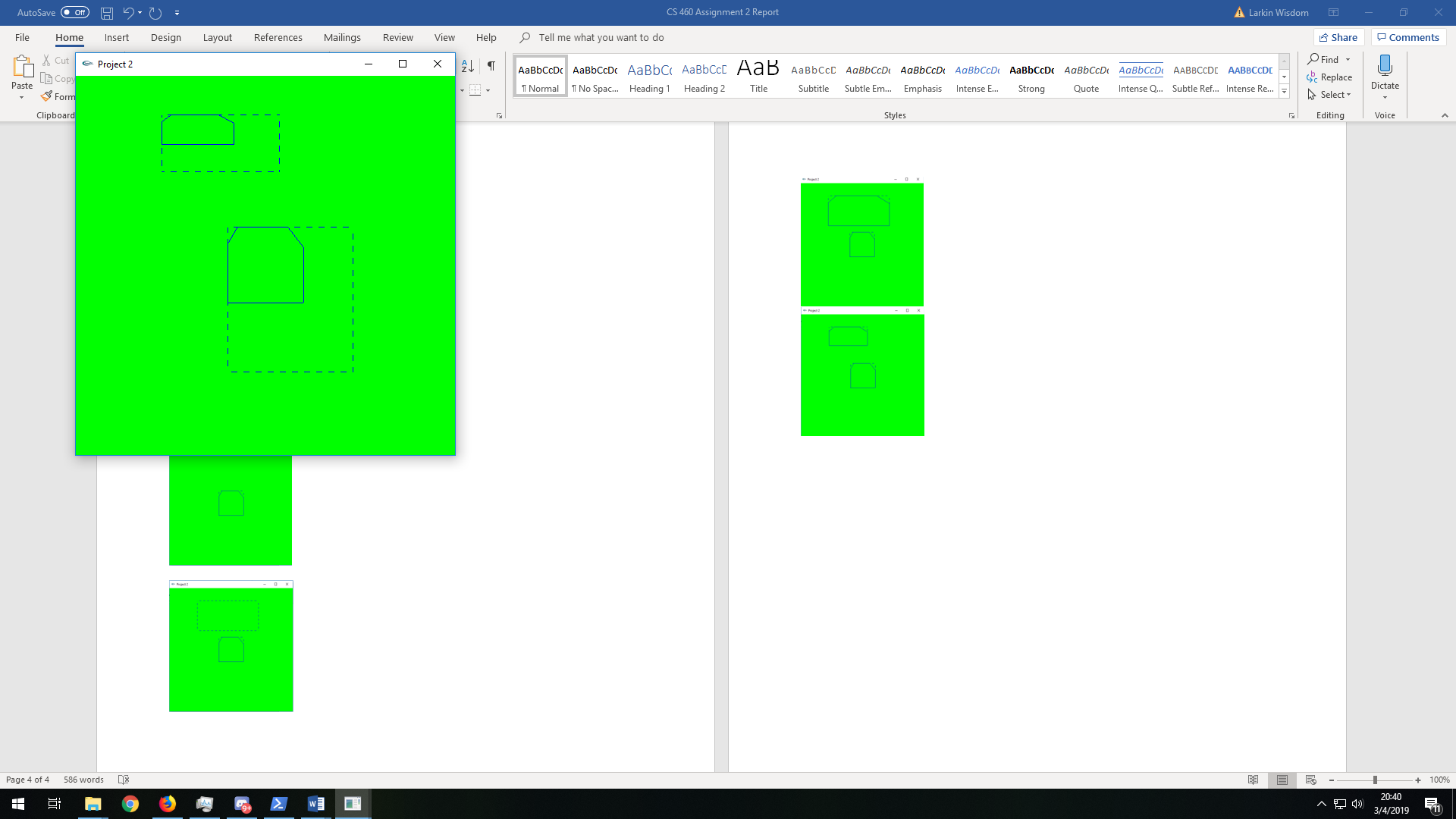
Transferring the polygon to the viewport



Scaling the viewport



Scaling the window, you can see the zooming effect



Moving the window, you can see the panning effect

