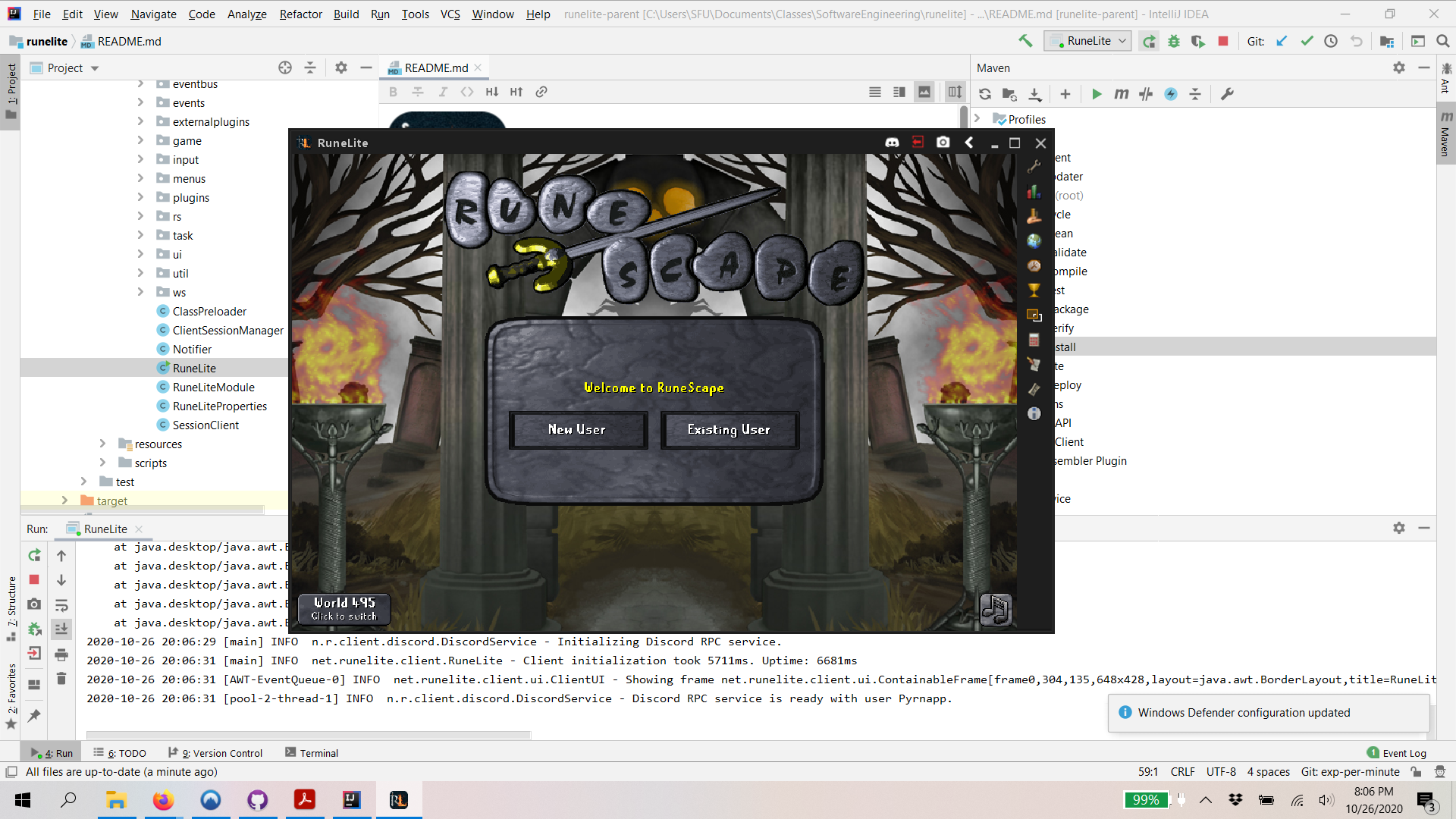
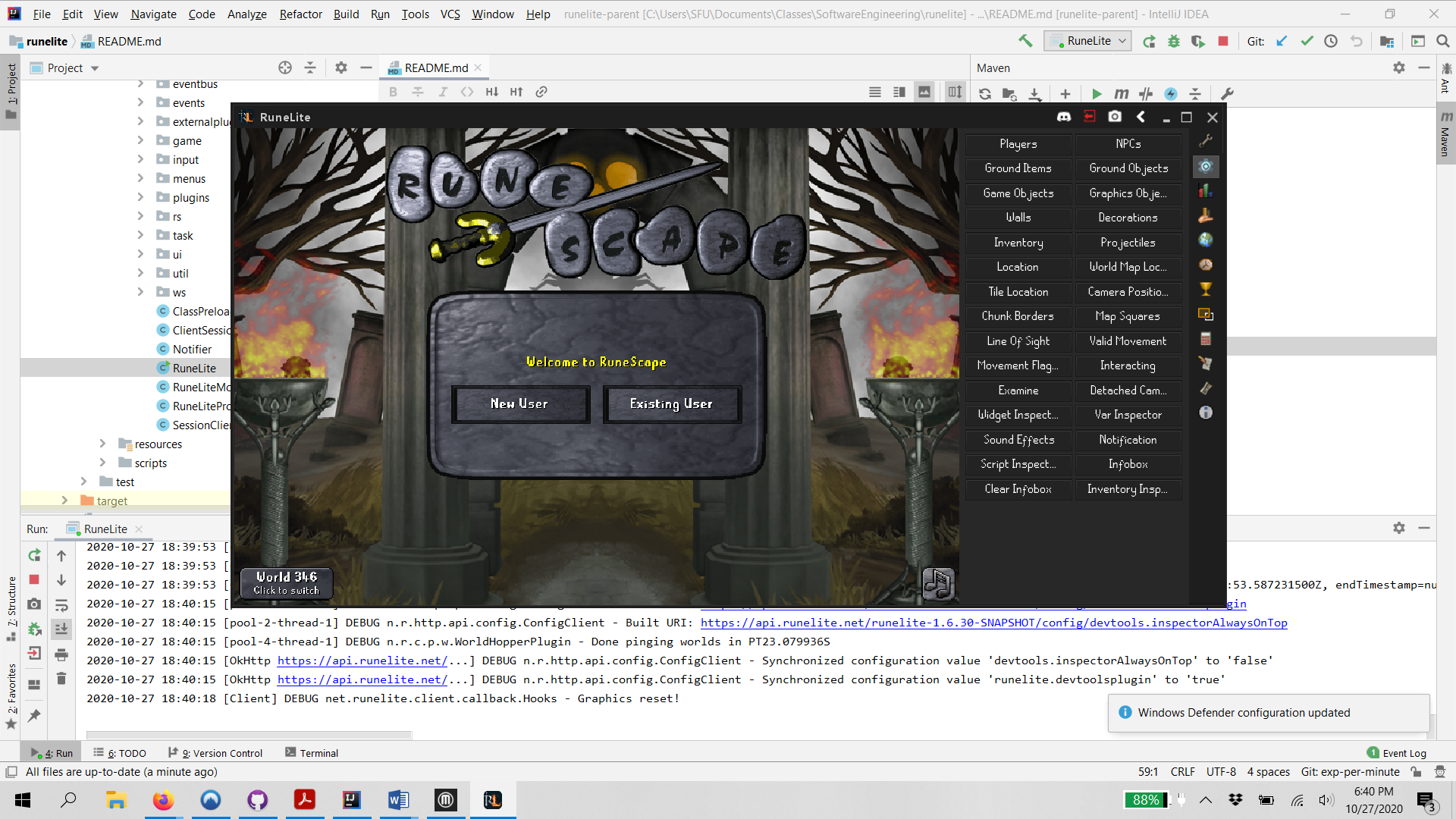
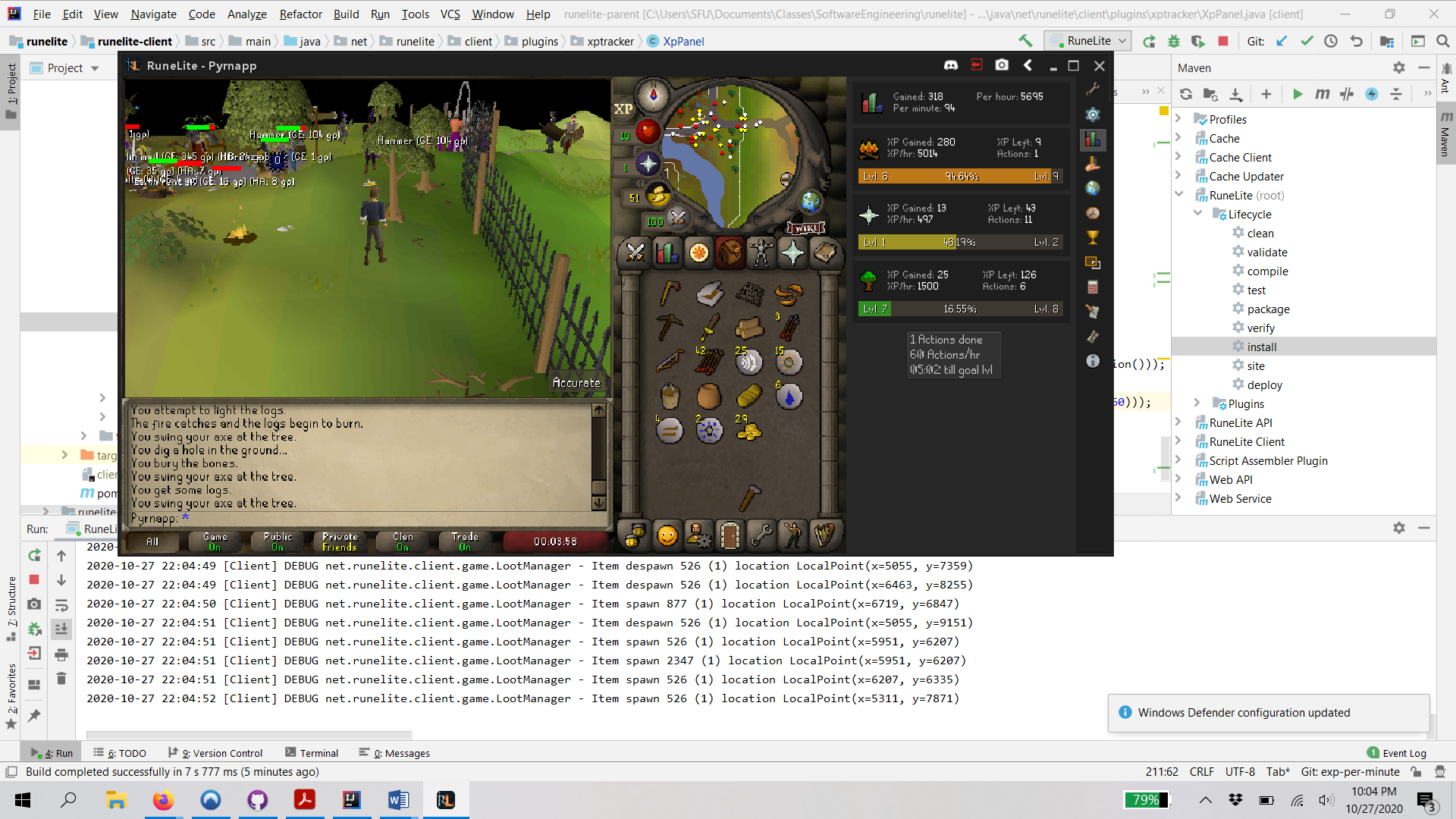
1: Get runelite build working: 

2: Enable Devtools



3: Implement Xp per minute counter



4: Code Review:

The “woodcutting” plugin manages the woodcutting mechanic in Old School RuneScape. I chose this program because I found myself frequently using woodcutting as a test for the “exp-per-minute” mod. This made woodcutting the skill I am most experienced with in RuneScape, and the logical choice for my code review.

The first thing I noticed about the woodcutting plugin that I did not expect is that it also appears to handle the trees. I expected the woodcutting plugin and tree management to be separate, but the woodcutting plugin handles the type of tree, their respective respawn times, and their durability. This seems to break the clean functions best practice indirectly – while the plugin in not a function per se, it is still doing more than it says on the tin. Perhaps this is essential to the functionality, or makes sense given the overall structure of the game, but I cannot see it yet. It would make sense to me to have respawn times for trees in some kind of world-builder plugin, which would manage respawn times, spawn rates, buildings, and terrain, but I can see why the developers might have chosen to put it in woodcutting. I am not sure if it was a good design choice, however.

Overall, while the code is not overly messy, I would not say it is clear either. The code is laid out in a way which would probably make sense to a developer used to the structure of the code, but as a total newcomer to the code, it is not terribly helpful for me. Given enough time to follow data through the structure and get a handle of how information is passed around, I’m sure I could figure it out. However, the point of clean code isn’t to be eventually understood, it is to be immediately understandable by a new developer. There are a lot of not-equal logic operations, which take longer to understand, and should be avoided.

Attempting to update this code would be a little bit of a challenge, as I would have to trace back the interconnections manually without documentation. This would be time consuming up front to get a full picture of how the moving parts connect. Once this is well understood however, I believe the code is well organized for an update. I can see structure and organization in the code, I just don’t understand the methodology yet. Everything is broken down into small functions with only a line or two in them, making sure that each function only does one thing. This is good coding practice and will make updating easier. Adding new features will only need to create new functions to support them, not dig through existing ones to figure out what you need to modify to make it happen. If you did need to modify an existing function for some reason, it would be fairly easy to find, as long as you knew what you were looking for.

If I were to develop my own plugin, I would want to make an automation assistant plugin. RuneScape seems to rely on grinding a bunch of repetitive tasks to advance, which is tedious and time consuming by default. I am not sure if such a tool exists already, as I am not very familiar with MMOs and RuneScape. To implement such a plugin, I would start with a simple activity such as woodcutting, where I know the object is fixed, and has a known respawn time. Since this plugin is designed for idling, efficiency is not an immediate concern. We can therefore assume that we can just wait for the same tree to respawn instead of having to pathfind between trees, although this could be added later. The plugin would cause a click event on the tree whenever it detected a valid tree object in front of the player character, initiating the woodcutting action automatically.

Much of the code for this plugin exists already, as most of the work is done by the vanilla game code. No additional play mechanics are being added, we ae just adding an automatic trigger for existing mechanics. The auto assistant only needs a UI, and a way to activate click events. The UI can be lifted from another plugin, but the auto-click may have to be written manually, depending on how user input is implemented. If a similar feature exists in RuneScape (as a new player, I do not know), it might be possible to re-use some of the code from that feature for the auto-clicker.