Jagex Macroing Detection
From the RS forums

Our macro detection system is highly advanced, reminiscent of our efforts to curb rule breakers prior to the removal of unbalanced trade. Admittedly, we devote more of our energies towards introducing actual game content.

The use of macros is nevertheless unacceptable, and contrary to popular belief we have not ignored the recent outbreak of bot usage. Modern macros however are quite advanced, rarely interact with the client, and issue commands the same way players point and click, rotate the camera, etc... We presume this sudden leap of sophistication to be due to the fear of being caught with well developed accounts, as most macro users seldom use throwaway accounts. Many recent macro users actually have comparable gameplay hours to legitimate players.

[name removed], I don't actually know the very specific details. It's likely that nobody knows the entire system to the letter, given the vastness of all of the software involved.

Firstly, we only issue bans if we can beyond a reasonable that the individual in question has committed a violation of the rules. We have stated this to the public quite a bit, and it really is true. I wouldn't lose sleep over the bans we've issued at all.

Our actual detection system monitors player behavior autonomously. As most bot programmers have unfortunately and conveniently guessed, it works on a comparison of expected human behavior to the observed behavior. Thus, macros that follow a set series of commands would be incredibly easy to catch, if they would even be operational. Random camera shifts usually thwart these, and finding them is a very rare occurrence.

We regularly dismantle macros, and the most common approach taken is object recognition with a series of programmed responses. One of our very first attempts to counter this was random color changes, usually not very noticeable. This turned out to be a failure as object recognition technology proved to be far ahead of this eventuality."

"Unless the appearance of the interface was changed sufficiently enough to make it unusable to even a regular player, it was not practical to make the game unrecognizable to a macro program.
All that we could do was analyze rather than inhibit, so to speak. By determining sets of behaviors that are not characteristically human, in "criteria", we would get the alarm from the system if all were met.
The first criteria is quite a bit similar to that of most first person shooter (FPS) games who take steps to detect "aimbots." This would be simple movement. Inhumanly fast cursor movements are highly suspicious, and consistent patterns of such movements (to discern from some player who gets bored and plays with the mouse .) I don't know the actual numbers involved, sorry. Some exceptionally skilled players frequently do trigger this, which is why multiple criteria exist.
The second criteria is the "accuracy" of the clicks. The interface determines exactly where any specific click was made, and is further divided into the actual buttons such as the inventory, worn

equipment, etc...

Primitive programs typically recognize the object, but click the same spot (often the same pixel.) This immediately raises the alarm if it is done consistently. Most programs do not do this, though, but rather have an "area of uncertainty" where the click will randomly fall in.

This is still easy to detect. There is accuracy, and precision. These programs do not replicate human precision. Consider the dartboard as an example:

The goal is to simply "hit it." A human will still aim for the bullseye, to increase the chance of hitting the dartboard. Thus hits will be concentrated towards the bullseye, but there will still be hits nowhere close, and some misses too.

A macro does not have human flaws, and can always hit the dartboard. However, if randomized in the "area of uncertainty" there is no such pattern."

This will also raise the alarm. More advanced yet, some macros will actually form concentric regions of where the clicks will land. Still, we're always ahead of rulebreakers. Our final line of defense in this criteria is to analyze the regions most commonly clicked.

A macro tends to click in a perfect square, circle, or sometimes the shape of an interface icon. Sometimes even the region is randomized. Human behavior is not truly random however. The region most commonly clicked is related to placement, which is all I know about this.

Highly advanced bots have made an attempt to replicate even this, but we have not seen anything convincing yet. Admittedly, we're disadvantaged in this aspect, so we cannot fully rely on simple movement and accuracy to detect a macro program.

The next criteria is the actual sequence of actions. A player that decides to cut some trees for logs, and sell them for profit actually has a longer list of actions to perform than you'd think. The efficiency of which this is completed can be analyzed, and much more easily compared to the behavior of a macro programs' attempt to replicate it.

If each step is merely randomized with the same bounds of uncertainty like virtually all bots seem to do today, the actual time it takes to execute the actions in all converges to a specific value! It's similar to flipping a fair coin multiple times, and converging to a frequency of outcomes (namely 50%.) Humans are generally much less consistent, and once the consistency falls within a certain range, the alarm is raised.

One more of many criteria is the effect of total gameplay time. As expected, exhaustion will set in for most players and slow down their actions. This is not a completely reliable method, since some players have been able to keep up their concentration for remarkably long periods of time."

"The last criteria I really know of is probably the most obscure and advanced. Human players tend to have much better object recognition capabilities than a macro program, and the limitations of the procedure can be exploited. While a human may be able to recognize a tree at a very unusual and discrete angle, many macros lack this capability, and thus keep the camera angle within a certain range. Some fail to change it entirely.

Macros tend to locate objects that are far out of reach by going close to the area in question, and then finding it. Normal players tend to look ahead first.

There are probably many more techniques employed, and the genius and innovative capability of my coworkers is simply amazing. I am very confident in their ability to enforce the rules, and keep RS a fair, and equal environment. The detection system is definitely conducive to such, but that is not our only method.

The next is abuse reports. There are many new and better bots, just written, that can avoid at a few of the criteria, and not be flagged. The intuition of a human being can never be surpassed, and abuse reports give us an all seeing eye that brought many rule breakers to justice. Abuse reports are also helpful for locating the archives, and comparing the new behavior of the new macro that was previously able to avoid detection. We can update accordingly.

We highly encourage abuse reporting, and we will not punish players who send in reports with a reasonable cause to suspect the player in question is using a macro.

Circumstances (such as the usual range of playing time) is taken into account as well.