



College of Engineering,  
Forestry, and Natural Sciences

Tanner Massahos, Joseph Remy, Julian Bell, Tyler Boice, Chase Mosteller

Title of Project: roBOTically efficient

[GitHub](#)

D.5 Release 1 – Due: 2018-03-30

CS386 – Software Engineering – Spring 2018

Dr. Marco Gerosa

## 1. Introduction

- i. In its current state, *roBOTically efficient* has a GUI that a user can utilize to activate a woodcutting bot and exit out of the bot whenever they would like. Its current features as of now is that it is an executable (easily distributable), very easy to use, acts almost like a real player when it comes to woodcutting, reliable, and easy to understand. It also works quickly and can in a way correct most errors thus far. It allows a user to be away from the computer, while still making gains within the woodcutting skill. We plan on adding more features and more bots as time goes on.
- ii. **GitHub:** [https://github.com/J1411/NAU\\_CS386](https://github.com/J1411/NAU_CS386)
- iii. **Trello:** <https://trello.com/b/4hvAsbj3/naucs386opdingo>

## 2. Implemented requirements

- i. As we stated in D.2\_Requirements, for our first release we stated, “we are going to create the GUI with just one bot for one skill, and make sure that it satisfies both the functional and non-functional requirements.” That is almost exactly what we did, we created one bot for one skill that could be managed through a GUI. We also implemented a large portion of our functional and non-functional requirements which can be seen below.
- ii. **Create a GUI for the application** – Implemented by Tanner and Joseph using Koda which was built into SciTE Script Editor.
- iii. **Create a bot for woodcutting** – All individuals within the group helped with this portion.
- iv. **Create bots that are human like** – Worked on by mostly Julian and Tanner, tried to randomize things, we are going to do chat later.
- v. **Reliability** – After we have tested it for a few hours, it seems to run just fine.
- vi. **Undetectability** – Have yet to be banned on any of the accounts we have utilized the bot on.
- vii. **Availability** – It would be easy to distribute as it is packed up in a nice executable.
- viii. **Usability** – All a user must do is click to start the bot next to some trees and the script handles the rest.
- ix. **Scalable** – This bot can be utilized on many different types of trees in many differing locations.

## 3. Adopted technologies

- i. **AutoIt** which is a free BASIC-like scripting language. We utilized this because it is what we had the most exposure to and there was a ton of online sources, it also just looked to be the easiest to quickly learn.
- ii. **SciTE Editor** was pretty much our only option for a script editor as it is what comes alongside AutoIt, it is also loaded full of awesome utilities.

- iii. **Open Broadcaster Software** which is what we used to record the video, we used this because it was the easiest for us to use, as most of us knew how to work the software.
- iv. **Koda** which is a GUI Designer, that also works alongside Autolt, it just made it very easy to create a basic GUI so we saw no reason not to use it.

#### 4. Licensing

- i. **GNU General Public License v3 (GPL-3)** is the license of our project (see our Github for the full license).
- ii. **Summary:** People may use, copy, distribute, and modify our software if the modifications, changes, and dates are kept track of. Any modifications or derivative works must use the same license as ours along with compile and run instructions.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	ⓘ License and copyright notice
✓ Modification	✗ Warranty	ⓘ State changes
✓ Distribution		ⓘ Disclose source
✓ Patent use		ⓘ Same license
✓ Private use		

#### 5. Look & feel

- i. The user interface is extremely basic. Utilizing the Koda GUI designer, comes with Autolt, we created a GUI menu with two buttons so far: the “Woodcutting bot” and a standard “EXIT” button. Once the skill/bot is selected, it confirms with the user on how to stop the bot software (hit Escape). The rest of the system is automated and currently does NOT have a graphical user interface.

ii. Screenshots below:

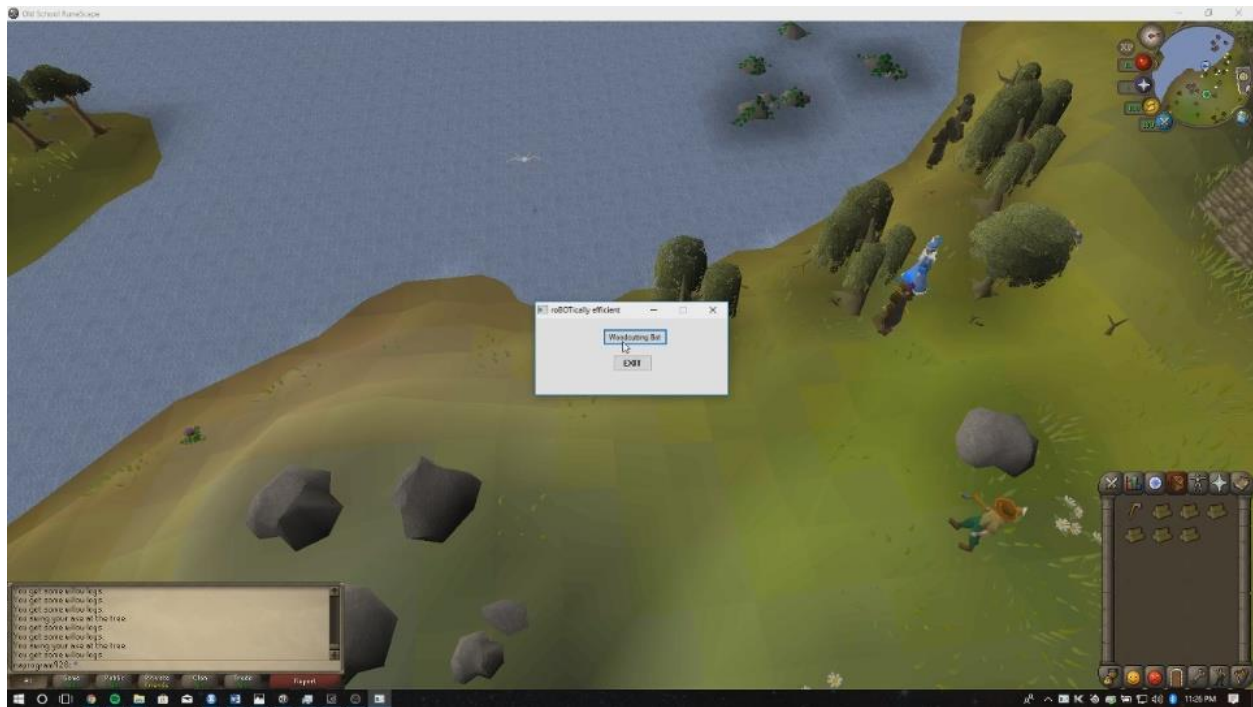


Figure 1: Shows the main menu with all possible bots/scripts for training skills, automating tasks, etc.



Figure 2: Simple confirmation dialog with instructions on how to stop the bot.

## 6. Learning/training

- i. Overall, most of the development was pretty straight forward. AutoIt was simple to use and the language behind it (Pascal) is extremely simple and similar to BASIC.
- ii. Information was primarily gathered via Internet resources such as YouTube videos, forum posts, etc. Since botting has “trade secrets,” because people can make money off of this service, most of the knowledge is kept secret and not disclosed usually.
- iii. The setup of the environment was not difficult and was quick to set up and start running.

## 7. Lessons learned

- i. In retrospect, our team would like to keep tinkering with the script. This is our first attempt and is merely a proof of concept in our eyes. Primarily, this release is showing us how automation can be accomplished, how we can improve, what needs to be fixed, and what else we need to learn.
- ii. Overall, we want to expand our bot options (other skills and whatnot) and continue to update the user interface to make it super easy to set up and use. Currently only works for Windows as well, which may or may not change.

## 8. Demo

- i. The link to the video is here :
  - i. YouTube video: [https://youtu.be/3w9i9tHuR\\_s](https://youtu.be/3w9i9tHuR_s)
  - ii. Google Drive back-up: <https://drive.google.com/open?id=1Zkos0uhbqAQJtunEuQvqdEW7Ibpqt0Zp>

## 9. Group participation

- i. Tanner Massahos – Updated the Trello page and lead the team when it came to developing our first bot. (25%)
- ii. Joseph Remy – Did a lot of research when it came to what software we could use when it came to scripting and creating bots, also assisted in creating the bot, reading over the final document, and commenting the code. (18.77%)
- iii. Julian Bell – Did most the revising for the deliverable, also handled and came up with solutions when we ran into any problems during development. (18.76%)
- iv. Tyler Boice – Tested the bot to verify it ran correctly. Also, wrote the learning/training section. (18.75%)
- v. Chase Mosteller – Tested the bot to verify it ran correctly. Also wrote the Lessons learned section. (18.72%)