**COMPSCI5018 MSc IT+ Project**

Essay: An investigation into the viability of developing a botting application in Runescape for the MSc IT+ Project

By

Paul Smith

Student ID: 3070332s

Student E-mail: [3070332s@student.gla.ac.uk](mailto:3070332s@student.gla.ac.uk)



Example screenshot of OSRS bot. Source: <https://joshuagornall.medium.com/how-i-made-a-runescape-bot-90248acae34>

Demonstration: [https://miro.medium.com/v2/resize:fit:640/format:webp/0\*v1QgOqzxiJPRfuTe.gif](https://miro.medium.com/v2/resize:fit:640/format:webp/0*v1QgOqzxiJPRfuTe.gif)

|  |
| --- |
| **Project topic:**  Self-education on Computer Vision (CV) and automation through developing a game botting application for Runescape for the MSc IT+ Project |
| **Aims (ranked):**   1. Self-education on automation 2. Self-education on botting 3. Develop a botting application to play the game hosted locally 4. Botting application can play the game hosted remotely 5. Botting application can play the game hosted on live versions 6. Gather and analyse data on bot performance 7. Develop anti-detection measures for bots |
| **Measurable Goals:**   1. Botting application can add new scripts and accounts. 2. Botting application can run a script without human intervention. 3. Botting application can safely terminate botting script. 4. Botting application can collect data on bot performance. 5. Botting application botscript can play game hosted locally 6. Botting application can run multiple bots on game hosted locally. 7. Botting application botscript can play game hosted remotely (LostCity/OSRS). 8. Botting application can run multiple bots on game hosted remotely (LostCity/OSRS). 9. Data analysis and visualisation on bot performance for each stage (goals 5-8). 10. Creation and publication of botting application and bot performance benchmarks. 11. Creation and publication of an API for botting. |
| **Timeline:** |
| **Requirements & Constraints:**  **Requirements**  **Personas:**   1. Botters 2. Bot developers   **Use cases:**   1. Botters will interact with the system via downloading it and their client, adding details of bot accounts to use, running a specific script and coming back to results. 2. Bot developers can interact with the system through analysing data collected, using this to inform how they will develop their own bot scripts and applications. 3. Bot developers can use the system to benchmark their own colour bots. 4. Bot developers can interact with the system’s API to benchmark a colour bot API. 5. Bot developers can interact with the system’s API to develop their own colour bots.   **System component interaction:**   1. Botters will interact with the system via the botting application GUI. 2. Botters will interact with the bot performance through reviewing their bots’ data. 3. Bot developers can interact with the system through analysing source code behind the botting application GUI, sample botscripts, and sample data. 4. The botting application will interact with the game through users running the application botscripts through the botting application’s GUI. 5. The botting application’s botscripts will interact with the game client using colours presented in the game client to identify current gamestate data, where the bot is, whether the bot is stuck and what tasks should or can be performed next.   **Data to support use cases:**   1. The first iteration will generate data on a bot’s performance. 2. The second iteration will generate data on multiple bots’ performance. 3. The third iteration will generate data on a bot’s performance on remote server (Lost City). 4. The fourth iteration will generate data on multiple bots’ performance on remote servers (Lost City). 5. The fifth iteration will generate data on a bot’s performance on game with anti-botting measures. 6. The sixth iteration will generate data on multiple bots’ performance on game with anti-botting measures.   **Constraints**  **Technology stack:**   1. Java – Must Have 2. Python – Must Have 3. WASPScripts – Would Like to Have 4. SIMBA – Would Like to Have 5. SRL – Would Like to Have   **Performance:**   1. Bots must be able to play the game. 2. The botting application must be able to interact with the game frequently enough such that they are not timed out.   **Security:**   1. Botters must be able to add account details without leaking usernames & passwords. 2. Botters must be able to review their botting data with the option of anonymising it.   **Usability**   1. Botters must be able to interact with the botting application GUI to run a botscript. 2. Botters must be able to add details of accounts they wish to bot on. 3. Botters must be able to run the game either through a client or web browser 4. Bot developers must be able to use the API to develop a botscript. |
| **Benchmarks**  **Botting platforms**  RuneMate– Client-based botting platform: reflection/injection.  DreamBot – Client-based botting platform: reflection/injection.  OSBot – Client-based botting platform: reflection/injection.  TriBot – Client-based botting platform: reflection/injection.  WASP Scripts – Pixel/Colour-based botting platform. Operates on SIMBA & SRL.  EpicBot – Client-based botting platform: reflection/injection.  OSMB – Pixel/Colour-based botting platform. Mobile Botting Niche.  **Botscript benchmarks**  Runescape Mining Bot: <https://github.com/joshuagornall/Runescape-Mining-Bot>  Parabot: <https://github.com/Parabot/Parabot>  WASPScripts Free colour bots: <https://waspscripts.com/scripts?type=free>  Slyautomation: <https://github.com/slyautomation/osrs_basic_botting_functions>  **Botting API benchmarks**  **Colour bot API benchmarks:**  SIMBA: <https://github.com/Villavu/Simba>   * Pascal-like language for image/colour recognition bots in Runescape.   SRL: <https://github.com/SRL/SRL>   * Simba Runescape Library   WASPScripts: <https://github.com/torwent/wasp-webapp>   * Built on SRL * Active community |
| **Initial problem analysis**  High Level problems:   * I want to learn about botting. * I want to learn about Computer Vision. * I used to play this game, but don’t have time anymore. * Game requires large time commitment. * Although accounts can be banned for botting, the game is rife with bots. * Botting has commercial viability aspects: botting to sell accounts for real money, botting for in-game currency to trade for real-world money (known as RWT: Real-world Trading). * I intend to use my MSc Project to learn about how to use Computer Vision (Colour/image recognition) for botting within this game. * I will create a benchmark for botting performance due to a lack of publicly available ones. |

**MoSCoW High-level Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| High-level Requirements | Status | Stage | MoSCoW |
| Bot(s) can run without human intervention. |  | MVP | Must Have |
| Can host game locally. | 18/06/2025 | MVP | Must Have |
| Bot(s) can perform multiple in-game activities. |  | MVP | Must Have |
| Bot application collects data |  | MVP | Must Have |
| Framework for analysing data |  | MVP | Must Have |
| Develop performance metrics |  | MVP | Should Have |
| Image recognition |  | MVP | Must Have |
| Botting Application GUI |  | MVP | Must Have |
| GUI: add account |  | MVP | Must Have |
| GUI: monitor bot(s) status |  | MVP | Must Have |
| GUI: show progress |  | MVP | Must Have |
| GUI: stop bot(s) |  | MVP | Must Have |
| Use MVP to develop API |  | API | Should Have |
| Create API to streamline botscript development |  | API | Should Have |
| Use API to develop botscripts |  | API | Should Have |
| Bot(s) can be ran on Lost City |  | LC | Could Have |
| Botting application can be ran on a Virtual Machine |  | LC | Could Have |
| Bot(s) can be ran on OSRS |  | OSRS | Could Have |
| Botting application can be ran on multiple Virtual Machines |  | OSRS | Would Like to Have |

A diagram of a computer program

AI-generated content may be incorrect.**Minimum Viable Product (botting application):**

Produce a botting application which contains at least one script. Scripts must use image recognition for input. Scripts must provide output in the form of game-interactions. The botting application must allow the player to bot on at least one account. The botting application must at least work on a locally hosted server of Lost City. The botting application must provide users with a User-Interface to: add a game account to user for botting, monitor status of any currently running bots, show user how long bots have been running for, show user what activities any currently running bots are performing, show user how much progress any currently running bots have made, allow users to stop a given bot or all bots. The botting application must also collect data on the bots for data analysis. Data collected includes: total number of bots currently running, what activities each bot has performed, how long each bot has been running, how much progress each bot has made (e.g. total experience gained, items collected, monsters killed, etc), does the bot appear to be stuck (after a certain period of time elapsed has no further progress been recorded).

**Project Initial Planned Phases**

This project is divided into phases where each phase represents a new iteration. Each iteration will have challenges which are unique to that phase. It is expected that each phase will become progressively more difficult. Challenges and difficulties will present useful learning opportunities. The main focus is on how I can use this project and the challenges discovered to learn. A benefit of an iterative process is that enables adaptation, hence nothing planned is set in stone. It is worth noting some challenges may be so significant that they require an additional phase, or changing an existing one.

Phases 1 and 2 will be concerned with botting on a locally hosted game server. This is important because this game server will not have anti-botting measures implement, unlike other versions of the game. Phases 3 and 4 will be playing the game on a live version of the game which is purported to have anti-botting measures in place. Accordingly, bans are expected. Analysing ban data should be useful in helping develop a framework for improving bots’ anti-detection.

Phases 5 and 6 will be playing the game on a live version which is known to have anti-botting measures in place. This version will be Oldschool Runescape, hosted by Jagex who have an active anti-cheat team. These last two phases are expected to be very difficult.

Phase 1

This phase will consist of getting a version (LostCity) of the game (Runescape) running locally, creating an initial botting application which is able to run simple scripts for botting in-game activities, generating data on bot performance, generating feedback for the developer to potentially create an API.

Phase 2

This phase will be concerned with exploring how the botting application can potentially run multiple bots and how to best manage computer resources.

Phase 3

This phase will be concerned with exploring how well the botting application functions on a version of the game with anti-botting measures, how well it performs in-game activities and what can be inferred from any ban data. Challenges expected: IP banned, detection of bot behaviour, account bans.

Phase 4

This phase will focus on exploring how best to run multiple bots

Phase 5

Phase 6