PowerGrid documentation

An overview of the PowerGrid project, how it works, and what the differences are with other Runescape bot clients.



**Table of Contents**

[PowerGrid documentation 1](#_Toc376726397)

[Chapter 1 – Introduction 3](#_Toc376726398)

[Chapter 2 – Overview of PowerGrid 4](#_Toc376726399)

[2.1 – Communication with the Runescape client 4](#_Toc376726400)

[2.2 – Cross-platform Graphical User Interfaces 4](#_Toc376726401)

[2.3 – High-level task descriptions in favor of direct control 4](#_Toc376726402)

[Chapter 3 – Structural overview 5](#_Toc376726403)

[3.1 – Structural diagram of PowerGrid 5](#_Toc376726404)

[3.2 – Summary of each of the modules in PowerGrid 6](#_Toc376726405)

[3.3 – The process of performing tasks in PowerGrid 7](#_Toc376726406)

[Chapter 4 – Implementation 8](#_Toc376726407)

[Chapter 5 – Legal notes 9](#_Toc376726408)

# Chapter 1 – Introduction

Many bot clients already exist for the MMORPG Runescape. However, since the introduction of Runescape 3, most of these bot clients have been rendered useless, as using these bot clients may cause the player to be banned. As such, only few bot clients are able to provide the functionality that is desired from a bot client without the implied risk of being banned.

The above problems, together with observations of the points where other bot clients failed on, calls for a new approach to Runescape bot development. This project aims to fill the gap left behind by the other bot clients after the Runescape 3 update, and to provide the functionality requested by its users more than other bot clients did before. This project aims to do so by making the client completely open-source. Also, to prevent direct manipulation of the Runescape client, this new bot client will not perform any byte code injection on the running Runescape client. Instead, the core of the bot client is written in C++, an object-oriented language which is somewhat similar to Java. Because of this, most of the bot client’s botting functionality will happen completely separate from the Java Virtual Machine, and as such also invisible from the Runescape client.

Putting all the above together, we present you PowerGrid. A revolutionary, open-source bot client that intends to make life easier for everyone by providing functionality that automatically reads the data from the Runescape client and presents it as native C++ objects. Because of this, the bot can provide the same functionality as traditional injection bots, but without modifying the game data.

The final goal of PowerGrid is to provide users with a tool that can play Runescape completely by itself, automatically deciding on the tasks to perform based on changes in the environment. PowerGrid will even be able to perform abstract tasks like leveling a certain skill to a certain level, or get to a certain destination. PowerGrid should then automatically decide on the concrete tasks (what methods to use for quick travelling, or what method to use to train the requested skill), based on the efficiency of each task.

# Chapter 2 – Overview of PowerGrid

## 2.1 – Communication with the Runescape client

The main component of any Runescape bot client is getting the information from the Runescape client, and sending actions back to the client. To achieve this between the C++ bot and the Java client, it is required to use an intermediate system provided by the Java Virtual Machine called JNI (Java Native Interface). A drawback of the JNI is, that it is a quite basic system, not at all suitable for the intensive and complex operations that a bot client requires. Therefore, PowerGrid uses a library that extends JNI and provides a more natural way to communicate with the Java Virtual Machine. This library is called JACE, and it provides a way to communicate with Java classes and objects as if they were C++ classes and objects, making it easier to bridge the gap between the two languages.

## 2.2 – Cross-platform Graphical User Interfaces

PowerGrid is written in C++, and as such, it is platform-dependent. To make it easier to use PowerGrid on multiple platforms, such as Windows, Mac OS, and Linux, PowerGrid makes use of a sophisticated library that aims to remove all platform differences, allowing the same code to work on almost all popular platforms. This library (called Qt) also features ways to dynamically load plugins, something which will extend PowerGrid with additional features and functionality.

## 2.3 – High-level task descriptions in favor of direct control

PowerGrid is meant to be used with high level tasks, so for example “train mining skill to level 99”, instead of “mine this much of this type of ore at that place”. PowerGrid is then meant to automatically decide the best location and ore type between the options it knows about. This supposedly results in a more natural playing style, since PowerGrid can decide to change strategies midway and go to a different mining spot if that turns out better (for example, when the initial mining spot is very crowded). By giving high-level tasks to PowerGrid, it may be possible to use strategies that players would normally not think of, but are more efficient than conventional strategies, while still remaining more player-like due to this strategy-switching behavior.

# Chapter 3 – Structural overview

## 3.1 – Structural diagram of PowerGrid

PowerGrid consists of many small parts, but these can be grouped together in only a few large parts with distinct roles. The diagram below shows the way information is sent between the Runescape client and PowerGrid.



Figure 1 - Structural Diagram of PowerGrid

## 3.2 – Summary of each of the modules in PowerGrid

As seen in the diagram in chapter 3.1, PowerGrid consists of a few large modules. In this chapter, a summary of the tasks of each of these modules is given.

RS Client

The Runescape client, running in the Java Virtual Machine. It connects to the Runescape servers to allow player to play the game online. PowerGrid can read data from the Runescape client which it uses to decide its actions.

AWT

AWT stands for Advanced Windowing Toolkit. This component is part of Java itself and contains the required code to process input and to display output to the screen. Runescape uses AWT to draw the world, and PowerGrid uses AWT to simulate user input.

PG Loader

The PowerGrid client loader, which loads and starts the Runescape client. It is built with the focus on performance. As such, it is able to start the client faster than almost any other client, including the official Runescape Launcher. It also plays a part in the communication between Java and C++.

JACE

The library that is used to communicate with the Java Virtual Machine. Through JACE, the bot can read information from the client, and take actions based on this. It serves as the main component linking the Java Virtual Machine and the PowerGrid C++ client.

Event Handler

This component deals with constructing and queuing input events to be sent to the client. The Event Handler then sends the required input events to AWT through JACE. The Runescape client then accepts these events as user input. Basically, this component translates instructions like “Click the sprint button” to “Move the mouse to that location (x, y) and press the left mouse button”.

PowerGrid AI

The PowerGrid AI is the core of the bot. This component deals with executing tasks and performing the actual botting. It reads data using JACE and uses the Event Handler to execute actions. Users of PowerGrid can give instructions to the AI using the GUI.

PowerGrid GUI

The graphical user interface of PowerGrid. It allows users to select tasks for PowerGrid to perform, and have direct control over the PowerGrid client. The GUI is presented next to the Runescape client in a separate window. This is because Java (AWT/Swing) GUI parts and C++ (Qt) GUI parts cannot be mixed.

## 3.3 – The process of performing tasks in PowerGrid

To further illustrate what PowerGrid does and how it works, an example is given here of a user having PowerGrid execute a simple task.

Imagine the user wants to train the mining skill level of his/her character. And PowerGrid contains functionality to train this skill, then the user can select the appropriate skill in the list of skills that can be trained. PowerGrid will then look for tasks it knows that promise to train the skill (in this case: mining). PowerGrid will compare the skills based on efficiency and execute one or more of these tasks in sequence to achieve the goal the user has set (which can be in terms of experience gained, or reaching a certain level, amongst others).

To execute the task, PowerGrid may need to execute smaller subtasks (like moving to a specific spot, or collecting a specific item from the bank). In case of mining, PowerGrid would need to move the character to a mining spot, then repeatedly click minable rocks to collect ore, and (possibly) go to the bank and back when the character’s inventory is full. These are all subtasks, that are provided either by PowerGrid itself, or are included in one of the plugins loaded by PowerGrid. The task that involves moving to a mining spot may, in turn, also consist of smaller tasks, like using a lodestone, teleport, or just walking there. This way, large and complex tasks can be done by combining smaller and easier tasks.

Over time, PowerGrid may be able to support a large variety of tasks that help it to execute extremely complex tasks, like fighting against enemies in a group. PowerGrid may with time even be able to communicate with other bot clients and optimize task execution by telling each other about what they’re seeing. Consider the previous example of mining. Imagine that at a certain mining spot there are other users using PowerGrid. Then these clients may be able to tell that the mining spot they’re in is already very crowded. As such they may be able to pass on this information to other PowerGrid clients, so that these clients choose a different mining spot. This helps to prevent the PowerGrid client from navigating to a mining spot, only to find that it is too crowded when it gets there.

# Chapter 4 – Implementation

This chapter will go deeper into the implementation of all the parts of PowerGrid. It illustrates how the PowerGrid client does what it does. This chapter is mostly intended for programmers and developers, as it may contain terminology specific to programming or Runescape bot development.

<…>

# Chapter 5 – Legal notes

PowerGrid is completely free and open-source, and is licensed under the GPL license version 3. A copy of this license is available at <http://www.gnu.org/licenses/gpl.html>. A copy of this license is also shipped along with PowerGrid.

Furthermore, we, the authors of this document, and the main developers of PowerGrid, have no connection to Jagex Ltd. or Runescape whatsoever. Any and all trademarks and/or rights are property of their respective owners.

The libraries that are used for PowerGrid