Developer Manual of Monopoly

Written by YEUNG Tsz Lok 22076383d

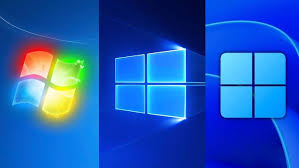
Brief Introduction:

This is Daniel from Group43. In this Manual, we will go through the expected specifications for the software to be run at, i.e. depiction of the required Operating System (OS) platform, programming language, the language package version (Dev Kit) , intended Integrated Development Environment (IDE) used for the Monopoly game.

By any means, here’s a brief Table of Contents.

1. OS -----------------------------------------------pg.2
2. Programming Language----------------------pg.3
3. Development Kit-------------------------------pg.4
4. IDE------------------------------------------------pg.4
5. Simple setup (Lab computer example)----pg.5-8
6. **Operating System**

Our software is designed to run on the Windows operating system, specifically versions 10 and above. Our choice allows for broad compatibility with a range of hardware configurations, but most importantly commonly used in development and gaming environments, and general purposes. Windows just simply has a user-friendly interface and stable support for Java applications, making sure the Monopoly game can be run at ease.



1. **Programming Language**

The Monopoly game is developed using the Java programming language. The language's strong adherence to object-oriented programming (OOP) principles promotes modular design, we are enabled to create reusable code components that can be easily maintained and extended.

Java’s rich set of libraries and frameworks further enhances development efficiency, which support some pre-built functionalities that helping our coding process faster. Java as well displays strong type-checking and exception handling capabilities contribute to building a more reliable and error-free application. Overall, Java's performance, combined with its OOP features, makes us decide to choose it.

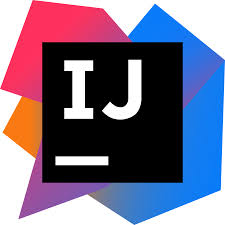


1. **Java package/Development Kit version**

We are running JDK-22. It has high compatibility and language syntax can be written cleaner and more concise. We also introduced some advanced functionalities incorporated in our software.

1. **Integrated Development Environment**

IntelliJ IDEA is the IDE we selected. We chose IntelliJ due to its powerful code assistance features, fair code layout and analysis, with refactoring tools.



1. Setup for running the software / in debugging mode

(to be done in Monday, going to lab)