Yang Liu

Development of Mobile Phone Game Based on Java ME

Bachelor's Thesis
Information Technology

May 2011

MIKKELIN AMMATTIKORKEAKOULU
Mikkeli University of Applied Sciences

DESCRIPTION

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Date of the bachelor's thesis** 15th May, 2011
Author(s) Yang Liu	**Degree programme and option** Information Technology

Name of the bachelor's thesis

Development of Mobile Phone Game Based on Java ME

Abstract

Recently, mobile phones have become more and more widespread in more than one aspect. Meanwhile, a large number of advanced features have also been applied into mobile devices. As we know, mobile phone game is one of them. In this final thesis, I develop a Chinese Chess game.

Chinese Chess, also called Xiang Qi, is one of the most popular and oldest board games worldwide, which is more or less similar to Western Chess related to the appearance and regulations. In order to spread China culture and make individuals realize how fun and easy this game is, I introduce this Chinese Chess game as the topic in terms of my final thesis.

In this final thesis, I use API (JSR 118) to build a user interface so as to set the board and pieces in the first place. Thereafter, some relevant basic rules are drawn up through logical control. This project is designated to be run on Java ME platform and Java SDK simulation software.

Subject headings, (keywords)

Java ME, JDK, Java SDK, MIDP, CLDC, API, Chinese Chess

Pages 63 p.+app.28	**Language** English	**URN**

Remarks, notes on appendices

Tutor Matti Koivisto	**Employer of the bachelor's thesis** Mikkeli University of Applied Sciences

ACKNOWLEDGEMENT

In the first place, I would like to represent my greatest appreciation to my supervisor Mr. Matti Koivisto. He gives me plenty of sincere advice related to my final report and the huge project. Without his constant supports, I cannot complete them in such a short time successfully. Meanwhile, I also want to express my gratitude to Ms. Elina Kiviranta, who checks the grammar errors for me.

What is more, I would like to give my thanks to my partner Mr. Shuaiyi Lv in terms of my final thesis during this period. With his help, we can finish this joint project together successfully. More importantly, we always encourage ourselves whenever we feel frustrated and lonely.

In addition, I want to thank my teachers who have taught me in not only Mikkeli University of Applied Sciences but also Beijing University of Technology. I would like to express my appreciation to them to give me an opportunity to study in Finland and make more familiar with European study style and life style.

Last but not least, I would dedicate this final thesis to my dear parents who gave me everything. They are always like my real friends who can always stand by me, give me a listening ear, a warm shoulder, and a helping hand no matter what.

LIST OF ABBREVIATIONS

API	Application Programming Interface
BD-J	Blu-ray Disc Java
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
FP	Foundation Profile
GPS	Global Positioning Systems
IDE	Integrated Development Environment
IMP	Information Module Profile
JAVA EE	Java Enterprise Edition
JAVA ME	Java Micro Edition
JAVA SE	Java Standard Edition
JAR	Java Archive File
JSR	Java Specification Requests
JDK	Java Development Kit
JVM	Java Virtual Machine
MIDP	Mobile Information Device Profile
PBP	Personal Basis Profile
PP	Personal Profile
PDA	Personal Digital Assistant
RDA	Remote Device Access
SDK	Software Development Kit
SMS	Short Message Service
UI	User Interface
WAP	Wireless Application Protocol

LIST OF FIGURES

LIST OF CODE EXAMPLES

TABLE OF CONTENTS

1 INTRODUCTION

Nowadays, mobile phones have drawn not only adults' but also teenagers' attention in more than one aspect. They have also become more and more significant tools in our daily life not only for their essential functions related to transmitting and receiving phone calls but also for their great number of advanced functions. Individuals can use their mobile phones to surf the internet, play the phone games based on Java, take photos by using built-in camera, transmit information via Bluetooth, or even listen to the music and watch TV online. With an increasing number of features embedded into the mobile phones, their appearances have also changed from the brick-like box to the suitable device inside the pocket handily.

Recently, there are a multitude of new features provided to make people more convenient and familiar to use their mobile phones like GPS, built-in camera, Infrared, Bluetooth, and WLAN connections. As it is well-known, Java APIs are available to realize these features. In this research area, many different kinds of applications can be implemented as well.

At the same time, mobile phone games are becoming more and more popular among youngsters. They are widely used on different kinds of devices including mobile phones, PDAs, and personal computers. There are also numerous types of mobile games like adventure, roles play, puzzle, and so on, which can be played not only on the multi-player mode but also on the single-player mode.

The aim of my final thesis is to become more familiar with some applications of mobile phone games by using Java Micro Edition, which is one of the most popular platforms for mobile phone development. As we know, Chinese traditional culture is well-known and profound all over the world. So in my research area, I mainly emphasize a Chinese Chess game which I designed briefly to implement my project.

The structure of the thesis is as follows:

In Chapter 2, I introduce a comprehensive view of several types of games between different mobile phones, and analyze the similarities and differences among all the games. Lastly, the main idea of my project is displayed.

In Chapter 3, I demonstrate a general introduction of programming environment we usually use, briefly discuss the language in the first place. Thereafter, I illustrate the related issues regarding how to build the proper programming environment based on the Java Micro Edition, which is an edition I use to develop my application.

In Chapter 4, I explain some different kinds of development tools which are widely used among individuals, and compare their differences and relationships with each other. More specifically, NetBeans IDE 6.9.1 is the platform which I use to develop my study.

In Chapter 5, I present a Chinese Chess game which I came up with, and describe the main idea regarding to my project briefly. More specifically, some principles and regulations are demonstrated in order to make everybody understand clearly.

In Chapter 6, I explain the distribution with my partner, Mr. Shuaiyi Lv, on this joint programming project related to this thesis. I am responsible for the interaction for human-machine interface, and he is in charge of the logic control for response and process. As is known to all, which is a critical part of this project, and I emphasize in detail on this section.

In Chapter 7, I introduce the application concerning this project and build a proper programming environment step by step to illustrate the main idea, and then test my project in practice.

In Chapter 8, I draw a conclusion related to my final thesis. What is more, I also argue the challenges and difficulties I faced and the trouble-shooting with regard to my project. Last but not least, I discuss some prospects and further study which can be done in the future.

2 TYPES OF MOBILE PHONE GAMES

2.1 Overview

Nowadays, mobile phone games are becoming more and more popular in individuals' life. We can witness portability and mobility anywhere and anytime among all the different types of mobile phone games. One good reason that contributes to the success of mobile games is because of theirs mobility, which means players can play games anytime and anywhere. A survey on user preferences among students of Institute of Higher Learning (IHL) has been done to gather useful information and relevant data to support my study. The survey showed that an overwhelming majority (60%) of the respondents prefer to play games on mobile phone. (Hashim & Perkins 2008, 1) For this reason, we can see that mobile phone games have a broad marketing field. As we know, the first game that was pre-installed into a mobile phone was 'Snake' in selected Nokia models in 1997. (Forum Nokia [referred 13.03.2011]) Snake and its variants have since become the most popular game on the market with a great number of people having played. In this chapter, I specifically emphasize numerous kinds of mobile phone games and compare the similarities and differences between them briefly. Finally, I present my main idea of the study.

According to the method of usage, we can divide mobile phone games into two main types, one is called multi-player mobile phone games, and the other is called single-player mobile phone games. However, in terms of the form of representation, which can be classified into text-mode games and graphic-mode games as well. Thus, I explain these two main categories respectively as follows.

2.2 Classification Based on Method of Usage

2.2.1 Multi-player Mobile Phone Games

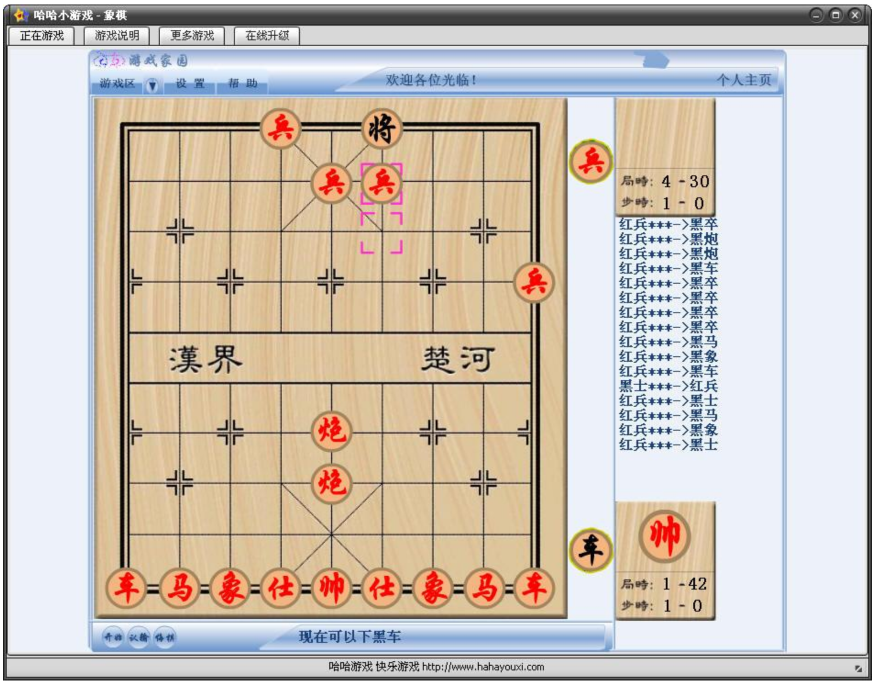
Recently, multi-player mobile games are always applied into the fields of PAN, LAN, MAN, or WAN by using several different wireless technologies. We can experience convenience in everything from cordless mice and keyboards, to satellite phones and global positioning systems (GPS). Generally speaking, some simple mobile phone games are usually used for WPAN, which is short for Wireless Personal Area Network. WPAN can be used for communication among the personal devices between a few metres. What is more, a WPAN can also be made possible with wireless network technologies such as IrDA, Bluetooth, UWB, Z-Wave and ZigBee. (Personal area network [referred 14.03.2011])

In these several years, the Bluetooth wireless technology is spreading rapidly. The number of Bluetooth chipsets shipped per year has doubled from 2009 to a total of 69 million chipsets in 2010, and manufactured Bluetooth-enabled equipment will still be led by mobile phones currently. (Pilato [referred 14.03.2011]) The following Figure 2.1 exhibits the logo of Bluetooth.

Figure 2.1: Bluetooth logo (Bluetooth.svg [referred 14.03.2011])

2.2.2 Single-player Mobile Phone Games

When it comes into single-player mobile phone games, also called Console games, I think which are more significant in individuals' life although multi-player mobile phone games are getting more and more popular in more than one aspect. Players can play the games flexibly without internet access, unlike a game with multiple players competing with or against each other to reach the game's goal, a one-player game is a battle solely against an element of the environment (an artificial opponent), against one's own skills, against time, or against chance. And many games described as "single-player" may be termed actually puzzles or recreations. (Single-player games [referred 14.03.2011]) Figure 2.2 below represents an example of single-mode mobile phone games.



红兵***->黑卒
红兵***->黑炮
红兵***->黑炮
红兵***->黑车
红兵***->黑卒
红兵***->黑卒
红兵***->黑卒
红兵***->黑卒
红兵***->黑马
红兵***->黑象
红兵***->黑车
黑士***->红兵
红兵***->黑士
红兵***->黑马
红兵***->黑象
红兵***->黑士

Figure 2.2: Chinese Chess game (Baidu pictures [referred 14.03.2011])

2.3 Classification Based on Form of Representation

2.3.1 Text-mode Games

Through the appearance of mobile phone interface, we can classify two main items, one is called text-mode games, and the other is called graphic-mode games.

Speaking of text-mode games, which is a form of games through swapping characteristics. We can play the games by replying the information according to the tips from mobile games. Furthermore, there are two basic approaches to achieve text games, SMS games and WAP games.

As we know, SMS (Short Message Service) games mean that we can play the games by transmitting and receiving the short messages between web servers and players. However, WAP (Wireless Application Protocol) games are similar to SMS games more or less. The games can be running based on the information shown on the WAP browers. Figure 2.3 below depicts the working process regarding to WAP games.

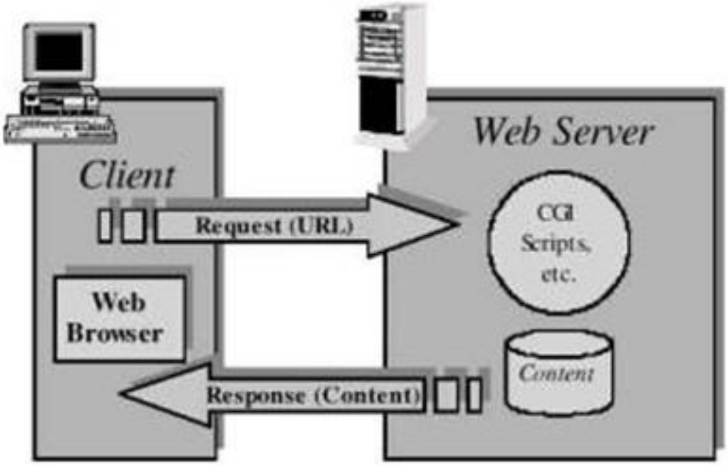


Figure 2.3: Working process of WAP games (Mobile games [referred 19.03.2011])

2.3.2 Graphic-mode Games

As far as graphic-mode games, which are the electronic games that involve interaction with a user interface to generate graphic feedback on the mobile phones. They are more like animations. More specifically, they can be divided into several areas like embedded games, J2ME games and some others.

Generally speaking, the games inherent existing in the mobile phones are usually called embedded games because they cannot be modified or deleted manually. Figure 2.4 below describes an example related to embedded games.

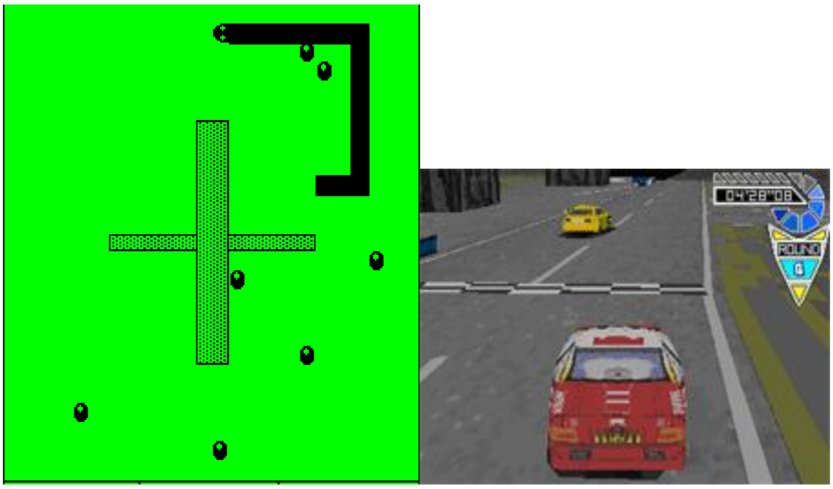


Figure 2.4: Classic game “Snake” and “Racing” (Mobile games [referred 19.03.2011])

In terms of J2ME games, they are based on Java Micro Edition platform. We can add, delete or modify some information to optimize the performance through Java language programming. Figure 2.5 below illustrates the screenshot of a J2ME game, we can witness clearly that the effect and performance may be better than in embedded games.

Figure 2.5: Performance of J2ME games (Mobile games [referred 19.03.2011])

In my study, I mainly focus on the project regarding to a Chinese Chess game. This project is actually a single-player mobile phone game based on Java Micro Edition platform.

3 MOBILE PROGRAMMING ENVIRONMENT

3.1 Java Programming Language

3.1.1 Introduction

With the rapid development of society, Java technology has become one of the most significant parts in our daily lives. From mobile phones to laptops, navigation systems to games, we can use Java technology to make them more functional, entertaining and convenient. The number of Java enabled mobile phones worldwide is over 250 million according to a press release from Sun titled: “Java technology is everywhere, surpasses 1.5 billion devices worldwide”, (Java technology [referred 24.03.2011]) and the number will continue to increase considerably.

In the early 1990s, Java was developed by a team led by James Gosling at Sun Microsystems, which was original called Oak and designed in 1991 for use in embedded chips in consumer electronic applications. But in 1995, it was renamed Java and redesigned for developing internet applications. (Mynttinen, 2011) Figure 3.1 below depicts the logo of Java.

Figure 3.1: Logo of Java (Java.svg [referred 24.03.2011])

3.1.2 Characteristics

Java is an object-oriented, simple, distributed, secure, dynamic and high performance programming language developed by Sun Microsystems, a company best known for its high-end Unix workstations. But now, it is a part of Oracle Corporation. Java language is designed to be small, simple, and portable across platforms and operating systems which derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. (Lemay 1996, 4)

Generally speaking, Java standard is defined by Java language specification and Java API. The Java language specification is a technical definition of the language that includes the syntax and semantics of the Java programming language. The application programming interface (API) contains many predefined classes and interfaces for developing Java programs. As we know, the Java language specification is stable, but the API is still expanding. (Mynttinen, 2011)

3.1.3 Classification

Recently, Java technology has become more and more popular due to its high qualities, and owing to its extendibility feature, we can applied it into various kinds of fields. There are several Java platforms in correspondence with different goals of development for programmers, offering them numerous proper resolutions they require. Figure 3.2 below demonstrates an overview of the components of different Java platforms and the areas of various applications.

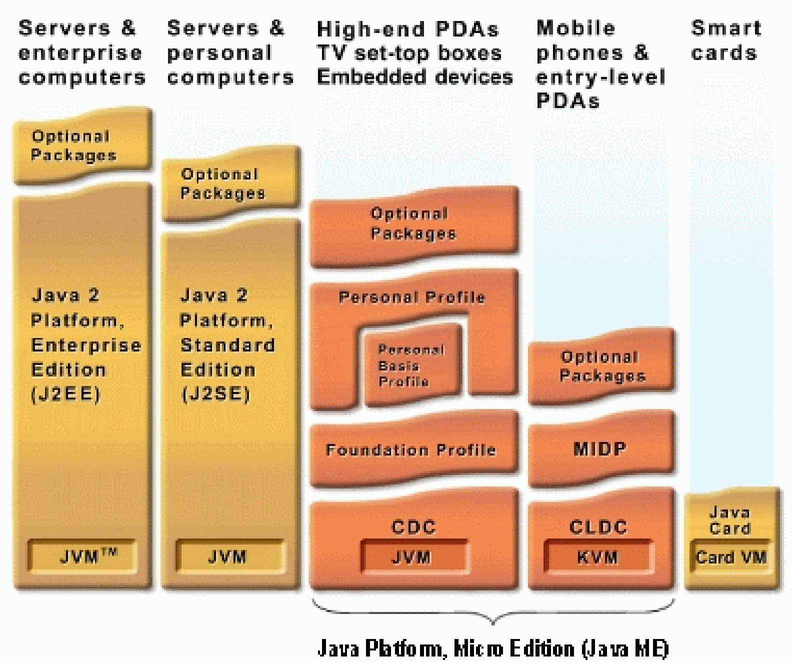

Servers & enterprise computers
Servers & personal computers
High-end PDAs TV set-top boxes Embedded devices
Mobile phones & entry-level PDAs
Smart cards
Optional Packages
Optional Packages
Optional Packages
Java 2 Platform, Enterprise Edition (J2EE)
Java 2 Platform, Standard Edition (J2SE)
Personal Profile
Personal Basis Profile
Optional Packages
Foundation Profile
MIDP
CDC
CLDC
Java Card
JVM™
JVM
JVM
KVM
Card VM
Java Platform, Micro Edition (Java ME)

Figure 3.2: Different Java platforms (Java platforms [referred 24.03.2011])

As can be seen from the graph, Java can be divided into three types of edition: Java Standard Edition (Java SE), Java Micro Edition (Java ME), and Java Enterprise Edition (Java EE).

Java SE can be always used to develop client-side standalone applications or applets, which is mainly designed for personal computing platforms like Sun Solaris, Linux and Microsoft Windows. It can help us to develop and deploy Java applications on desktops and servers, as well as today's demanding Embedded and Real-Time environments. (Java SE [referred 24.03.2011])

Java ME can be usually used to develop applications or MIDlets across many mobile devices, such as mobile phones, PDAs, PSPs, and some other wireless devices. This is a version of Java platforms targeted at devices which have limited processing power and storage capabilities and intermittent or fairly low-bandwidth network connections. (J2ME Tutorial [referred 24.03.2011])

Java EE is always applied into develop server-side applications, which is quite necessary for companies with great deployment needs and is the industry standard for enterprise Java computing. Developers will benefit from productivity improvements with more annotations, more POJOs, simplified packaging, and less XML configuration. (Java EE [referred 24.03.2011])

This final thesis will mainly deal with Java ME, so I will take more details in the next section about Java Micro Edition.

3.2 Java Micro Edition (Java ME)

3.2.1 Introduction

As we know, mobile phones are mobile devices with restricted hardware resources available, which means they cannot run complex applications as computers can because of the limited power capability and the small size of memory. In order to overcome those drawbacks, Java ME is carried out.

Java ME provides a robust, flexible environment for applications running on mobile and other embedded devices like mobile phones, personal digital assistants (PDAs), TV set-top boxes, and printers. Java ME includes flexible user interfaces, steady security, built-in network protocols, and support for networked and offline applications that can be downloaded dynamically. (Java ME [referred 26.03.2011])

3.2.2 Architecture

Generally speaking, there are three key elements in terms of Java ME technology: configuration, profile and optional packages. Thereafter, I will emphasize these several items in detail.

3.2.3 Configuration

Java ME can be divided into two types of configurations based on storage and process capacity which are CDC and CLDC respectively. CDC means Connected Device Configuration while CLDC is short for Connected Device Limited Device Configuration. Apparently, CDC is usually used for unrestricted mobile devices like set-top boxes while CLDC is always implemented into limited embedded devices like PDAs.

After that, I will demonstrate these two kinds of configurations and compare the similarities and differences with each other.

3.2.4 Connected Device Configuration (CDC)

As we know, this configuration is almost applied into some larger devices with more capacity and with a network-connection, like high-end personal digital assistants, and set-top boxes. The aim of the CDC configuration is to support the functions of a broad range of connected devices while fitting within their resource constraints. (Connected Device Configuration [referred 26.03.2011]) Figure 3.3 below depicts the structure of the CDC platform.

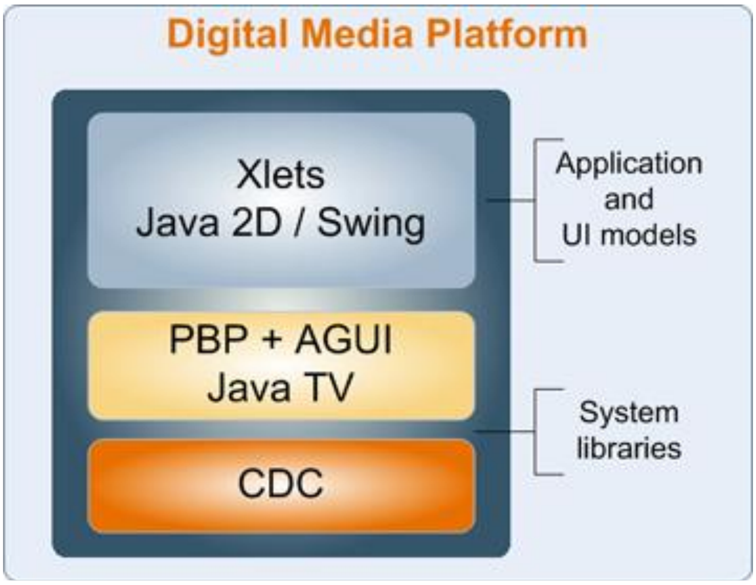


Figure 3.3: Structure of Connected Device Configuration (Digital Media Platform [referred 26.03.2011])

3.2.5 Connected Limited Device Configuration (CLDC)

CLDC is short for Connected Limited Device Configuration, which is designed to meet the requirements for a Java ME to run on devices with limited memory, processing and computing capabilities. What is more, Java ME defines a number of profiles related to CLDC in order to build an appropriate application. A widely adopted example is to combine the CLDC with the Mobile Information Device Profile (MIDP) to provide a complete Java application environment for mobile phones and other devices with similar capabilities. (Connected Limited Device Configuration [referred 26.03.2011]) As can be seen from Figure 3.4, which is the structure of the CLDC platform.

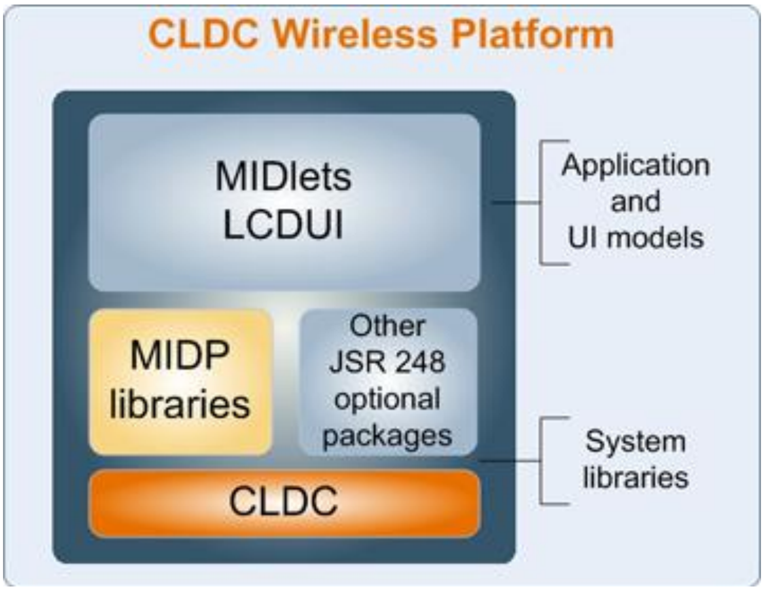


Figure 3.4: Structure of Connected Limited Device Configuration (CLDC Wireless Platform [referred 26.04.2011])

3.2.6 Profile

Profile forms a complete programming environment with Configuration I mentioned above, which consists of several Application Programming Interfaces (APIs) such as Foundation Profile (FP), Personal Basis Profile (PBP) and Personal Profile (PP) related to CDC, and Mobile Information Device Profile (MIDP) and Information Module Profile (IMP) related to CLDC.

In fact, MIDP is a type of profile I used in my study to build a complete Java programming environment for wireless devices based on CLDC.

3.2.7 Introduction to MIDP

As we know, MIDP is short for Mobile Information Device Profile, which is a specification published for the use of Java on embedded devices. MIDP is a key element of the Java ME framework and sits on top of Connected Limited Device Configuration. (Mobile Information Device Profile [referred 26.03.2011]) We can develop and deploy the applications designed by MIDP. Thus, CLDC and MIDP provide the core application functionality required by mobile applications, in the form of a standardized Java runtime environment and a rich set of Java APIs. Here are numerous advantages with regard to it. (MIDP Overview [referred 26.03.2011])

- Rich User Interface Capabilities
- Extensive Connectivity
- Multimedia and Game Functionality
- Over-the-Air-Provisioning
- End-to-End Scurity

3.2.8 Specification to MIDP

There are two specifications of MIDP, one is called MIDP 1.0 (JSR 37) and the other is called MIDP 2.0 (JSR 118). When it comes to MIDP 1.0, this is the initial version and may achieve some essential functions like user interface and Java specification requirement. In terms of MIDP 2.0, it is a modified version based on MIDP 1.0, which may provide more advanced features related to Java such as enhanced user interface and TCP/IP support.

3.2.9 MIDlet

MIDlet is short for Mobile Information Devices applet, which is actually a Java application embedded into the mobile devices, and only utilizes various APIs which are defined by MIDP and CLDC. The application I designed is practically a MIDlet.

MIDlets are all derived from the abstract base class javax.microedition.midlet.MIDlet, which contains methods that the MIDP platform calls to control the MIDlet’s life-cycle, as well as methods that the MIDlet itself can use to request a change in its state. A MIDlet must have a public default constructor which can be supplied by the programmer when initialization or there are no explicit constructors. (Koivisto, 2011) The empty default constructor can be demonstrated in Code example 3.1 as follow.

Code example 3.1:

```
public class MyMIDlet extends MIDlet {

   /* Optional constructor */
   MyMIDlet( ) {
   }
   protected void startApp( ) throws MIDletStateChangeException {
   }

   protected void pauseApp( ) {
   }

   protected void destroyApp(Boolean unconditional)
                        throws MIDletStateChangedException {
   }
}
```

As can be seen from the code above, there are three possible states in a MIDlet's life-cycle. Figure 3.5 demonstrates the MIDlet's life-cycle.

- Paused – The MIDlet instance has been constructed and is inactive.
- Active – The MIDlet is active.
- Destroyed – TheMIDlet has been terminated and is ready for reclamation by the garbage collector. (MIDlet [referred 26.03.2011])

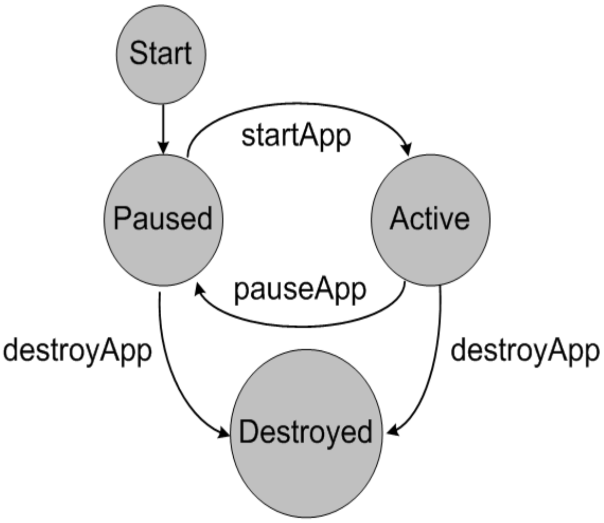


Figure 3.5: MIDlet's life-cycle (Life-cycle [referred 26.03.2011])

3.3.0 Optional Package

Generally speaking, optional packages provide some additional functions without modifying Profile, which can be regarded as the extension of Profile. In other words, when we need add a new API for Profile, we should firstly define it as an optional package. With time runs, if this package is widely used in many areas, it may be a specific part of Profile.

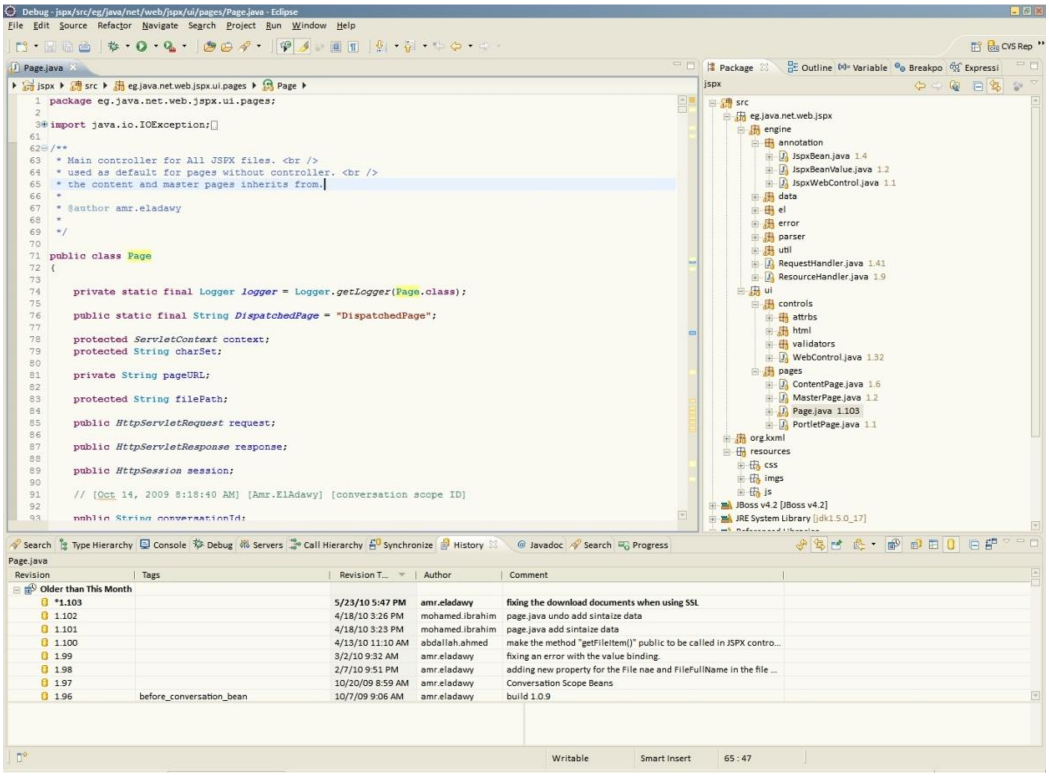
4 JAVA DEVELOPMENT TOOLS

4.1 Overview

Java ME can be implemented on a variety of platforms, but there are two main development tools that a number of people would like to use, one is called Eclipse and the other is called NetBeans. I illustrate these two development environments briefly and manly focus on NetBeans which I use to implement my study I designed.

4.2 Eclipse

Eclipse is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible system. (Eclipse (software) [referred 02.04.2011]) Apart from Java, some other programming languages like C, C++, and PHP can also be compiled by Eclipse. Figure 4.1 demonstrates the development platform of Eclipse.



```
package eg.java.net.web.jspx.ui.pages;

import java.io.IOException;

/**
 * Main controller for All JSPX files. <br />
 * used as default for pages without controller. <br />
 * the content and master pages inherits from.
 *
 * @author amr.eladawy
 *
 */

public class Page
{

    private static final Logger logger = Logger.getLogger(Page.class);

    public static final String DispatchedPage = "DispatchedPage";

    protected ServletContext context;
    protected String charSet;

    private String pageURL;

    protected String filePath;

    public HttpServletRequest request;

    public HttpServletResponse response;

    public HttpSession session;

    // [Oct 14, 2009 8:18:40 AM] [Amr.ElAdawy] [conversation scope ID]

    public String conversationId;
```

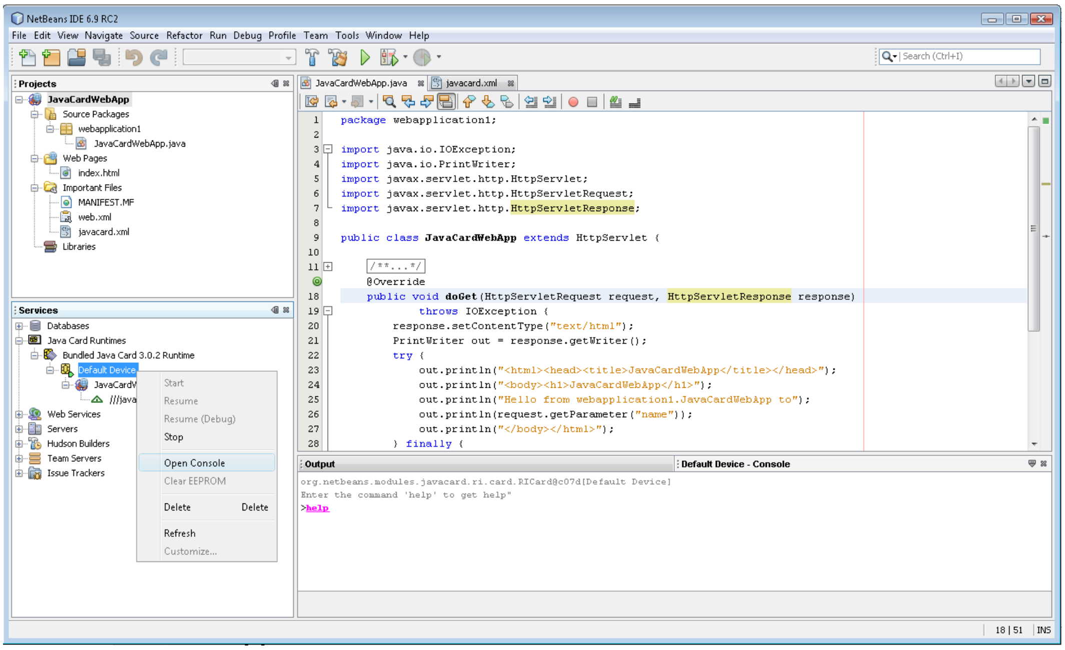
Page.java

Revision	Tags	Revision T...	Author	Comment
Older than This Month				
*1.103		5/23/10 5:47 PM	amr.eladawy	fixing the download documents when using SSL
1.102		4/18/10 3:26 PM	mohamed.ibrahim	page.java undo add sintaize data
1.101		4/18/10 3:23 PM	mohamed.ibrahim	page.java add sintaize data
1.100		4/13/10 11:10 AM	abdallah.ahmed	make the method "getFileItem()" public to be called in JSPX contro...
1.99		3/2/10 9:32 AM	amr.eladawy	fixing an error with the value binding.
1.98		2/7/10 9:51 PM	amr.eladawy	adding new property for the File nae and FileFullName in the file ...
1.97		10/20/09 8:59 AM	amr.eladawy	Conversation Scope Beans
1.96	before_conversation_bean	10/7/09 9:06 AM	amr.eladawy	build 1.0.9

Figure 4.1: Eclipse development platform (Eclipse IDE [referred 02.04.2011])

4.3 NetBeans

NetBeans is a platform for Java desktop applications and also an integrated development environment (IDE) for developing with not only Java but also many other languages. The NetBeans IDE is written in Java and can run anywhere a Java Virtual Machine (JVM) is installed, including Windows, Mac OS, Linux, and Solaris. A Java Development Kit (JDK) is also required for Java development functionality. (NetBeans [referred 05.04.2011]) Figure 4.2 illustrates the development platform of NetBeans.



```java
package webapplication1;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class JavaCardWebApp extends HttpServlet {

    /**...*/
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html><head><title>JavaCardWebApp</title></head>");
            out.println("<body><h1>JavaCardWebApp</h1>");
            out.println("Hello from webapplication1.JavaCardWebApp to");
            out.println(request.getParameter("name"));
            out.println("</body></html>");
        } finally {
```

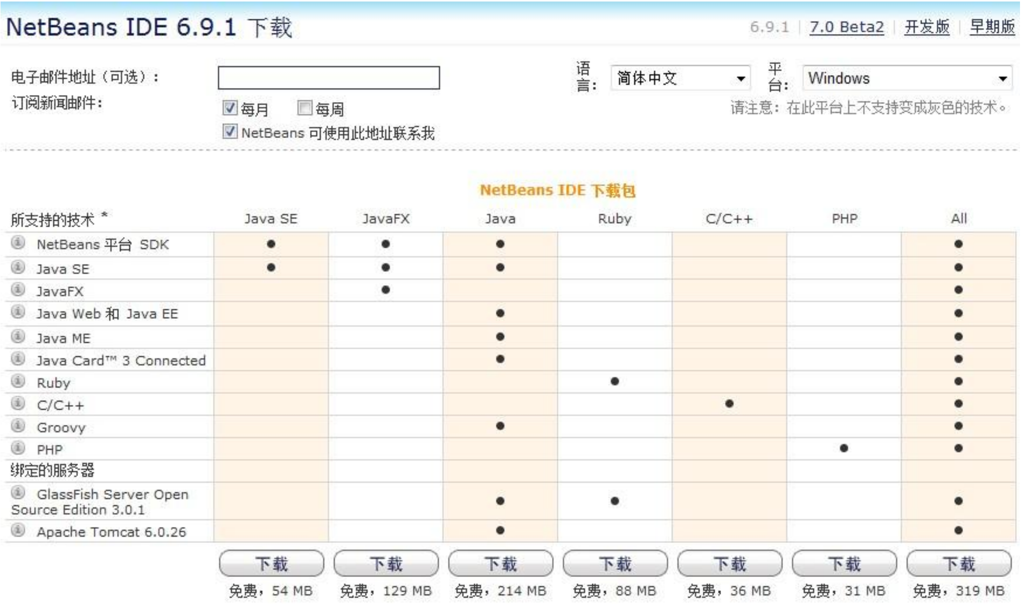
```
org.netbeans.modules.javacard.ri.card.RICard@c07d[Default Device]
Enter the command 'help' to get help"
>help
```

Figure 4.2: NetBeans development platform (NetBeans IDE [referred 05.04.2011])

In terms of my study, I use NetBeans IDE 6.9.1 as my development tool, which is the newest released version and very suitable for beginners to study gradually.

4.4 Installation

Generally speaking, we can download NetBeans IDE 6.9.1 for free from the internet, which is the newest version and it is available from the website http://netbeans.org/downloads/index.html. Then we may install this software step by step. Figure 4.3 below depicts the downloading package of NetBeans IDE 6.9.1.



NetBeans IDE 6.9.1 下载

6.9.1 | 7.0 Beta2 | 开发版 | 早期版

电子邮件地址（可选）：

订阅新闻邮件： ☑每月 □每周

☑NetBeans 可使用此地址联系我

语言： 简体中文

平台： Windows

请注意：在此平台上不支持变成灰色的技术。

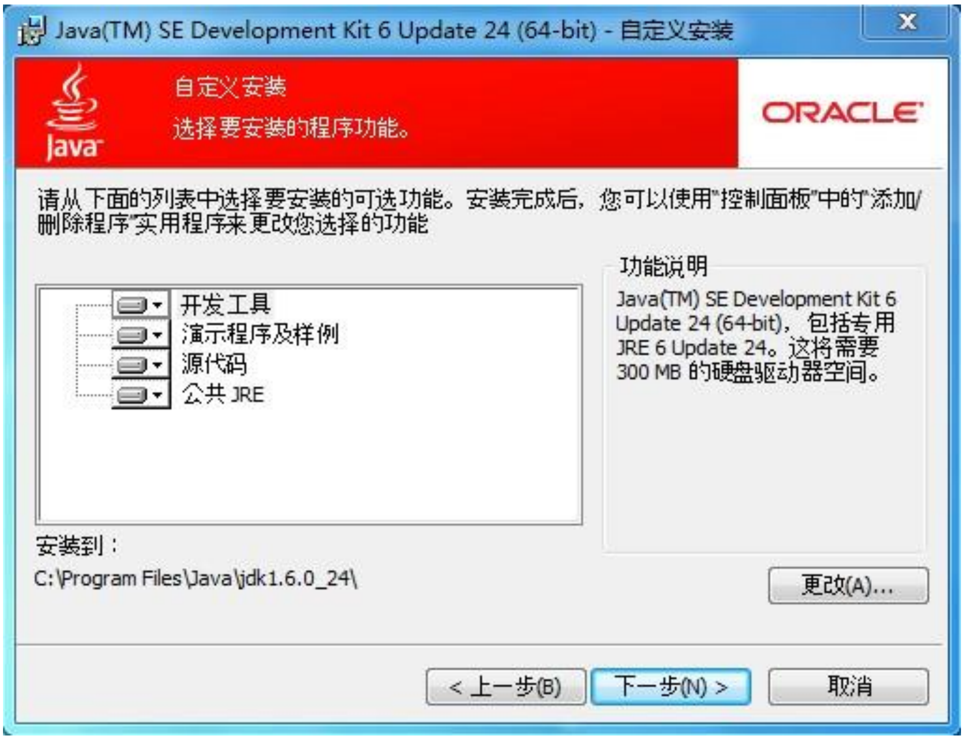
NetBeans IDE 下载包

所支持的技术 *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
NetBeans 平台 SDK	●	●	●				●
Java SE	●	●	●				●
JavaFX		●					●
Java Web 和 Java EE			●				●
Java ME			●				●
Java Card™ 3 Connected			●				●
Ruby				●			●
C/C++					●		●
Groovy			●				●
PHP						●	●
绑定的服务器							
GlassFish Server Open Source Edition 3.0.1			●	●			●
Apache Tomcat 6.0.26			●				●
	下载 免费，54 MB	下载 免费，129 MB	下载 免费，214 MB	下载 免费，88 MB	下载 免费，36 MB	下载 免费，31 MB	下载 免费，319 MB

Figure 4.3: NetBeans IDE downloading package (NetBeans Downloads [referred 05.04.2011])

Before we install NetBeans IDE, Java Development Kit (JDK) is required for Java ME platform. It can help us to create the Java Archive Files (jar.exe) automatically in terms of the process of Java compiler and applet. We may witness the installation process of JDK 6u24 as follows:

First of all, we must access into Custom Installation Wizard without any altering, and then choose Next Step button. This step is illustrated in Figure 4.4.



Java(TM) SE Development Kit 6 Update 24 (64-bit) - 自定义安装

自定义安装
选择要安装的程序功能。

请从下面的列表中选择要安装的可选功能。安装完成后，您可以使用"控制面板"中的"添加/删除程序"实用程序来更改您选择的功能

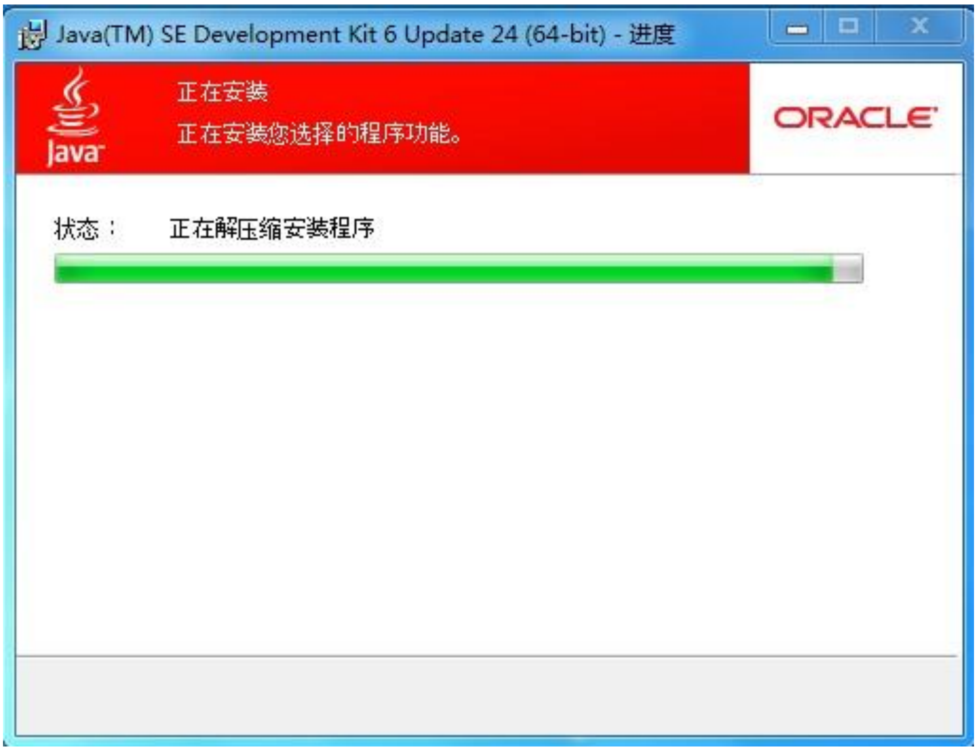
- 开发工具
- 演示程序及样例
- 源代码
- 公共 JRE

功能说明
Java(TM) SE Development Kit 6 Update 24 (64-bit)，包括专用 JRE 6 Update 24。这将需要 300 MB 的硬盘驱动器空间。

安装到：
C:\Program Files\Java\jdk1.6.0_24\

Figure 4.4: Custom Installation Wizard

Next, we should select an approximate route for installation. Thereafter, the package will be compressed and released. Figure 4.5 demonstrates the process of installing JDK 6u24.



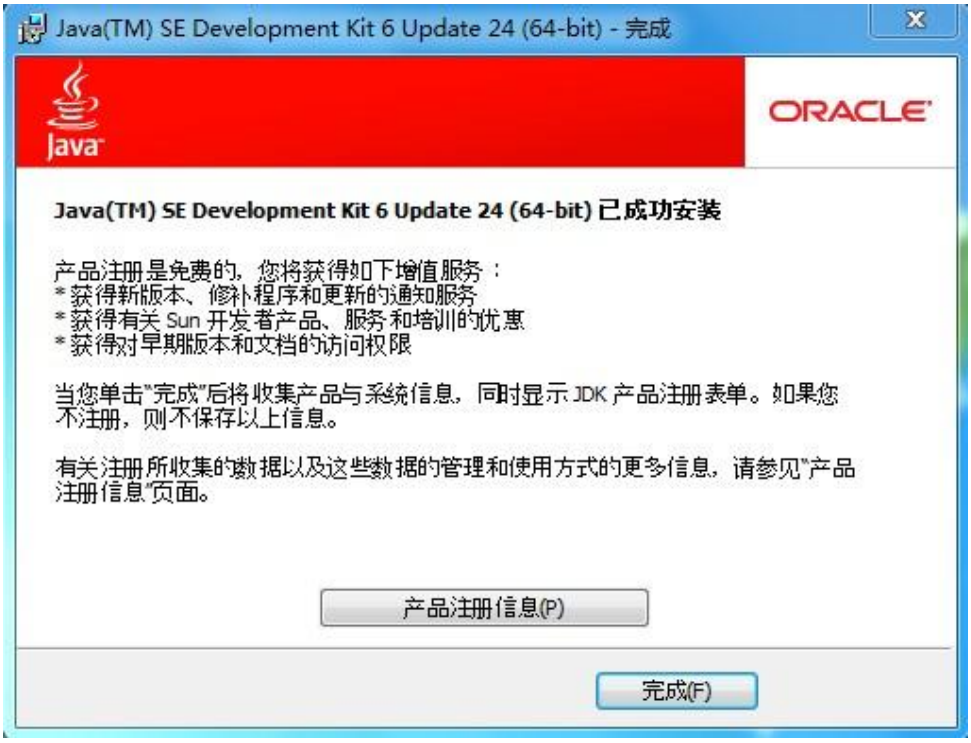
Java(TM) SE Development Kit 6 Update 24 (64-bit) - 进度

正在安装
正在安装您选择的程序功能。

状态： 正在解压缩安装程序

Figure 4.5: Process of JDK installation

Finally, JDK 6u24 Installation has been completed successfully as we can see from Figure 4.6, so we can just click the Finish button in order to install NetBeans IDE 6.9.1 for the next sectiom.



Java(TM) SE Development Kit 6 Update 24 (64-bit) - 完成

Java(TM) SE Development Kit 6 Update 24 (64-bit) 已成功安装

产品注册是免费的，您将获得如下增值服务：
* 获得新版本、修补程序和更新的通知服务
* 获得有关 Sun 开发者产品、服务和培训的优惠
* 获得对早期版本和文档的访问权限

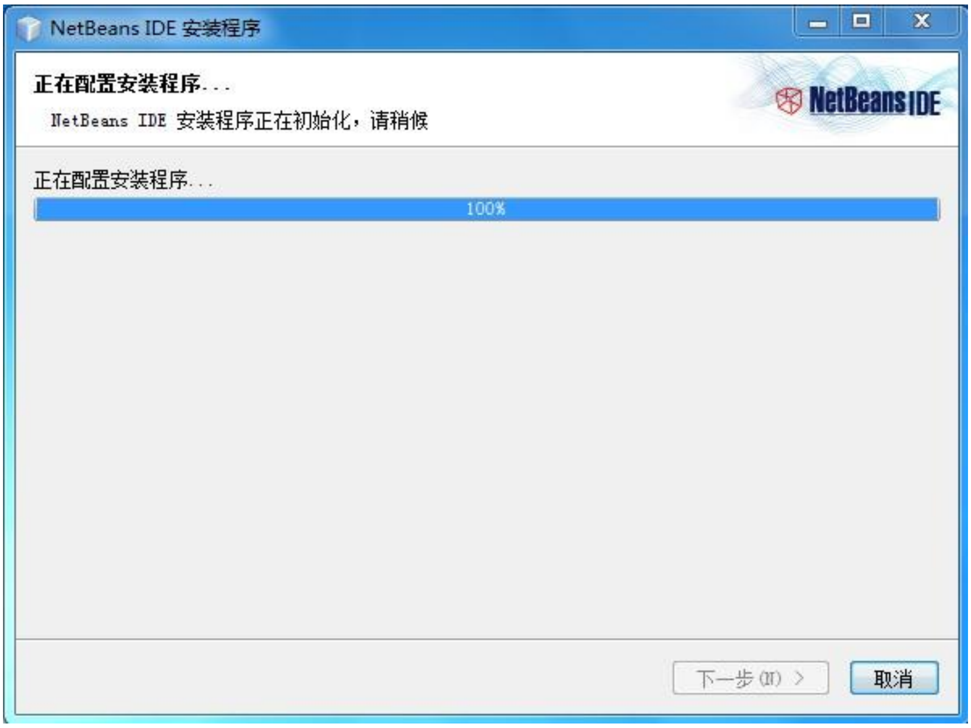
当您单击"完成"后将收集产品与系统信息，同时显示 JDK 产品注册表单。如果您不注册，则不保存以上信息。

有关注册所收集的数据以及这些数据的管理和使用方式的更多信息，请参见"产品注册信息"页面。

Figure 4.6: JDK installation is completed

Once the installation of JDK 6u24 is completed, NetBeans IDE 6.9.1 can be installed step by step so as to develop Java applications and applets. So we can witness the installation process of NetBeans IDE as follows:

In the first step, NetaBeans IDE 6.9.1 initialization must be configured at start. Figure 4.7 below shows the configuration of NetBeans initialization.



NetBeans IDE 安装程序

正在配置安装程序...

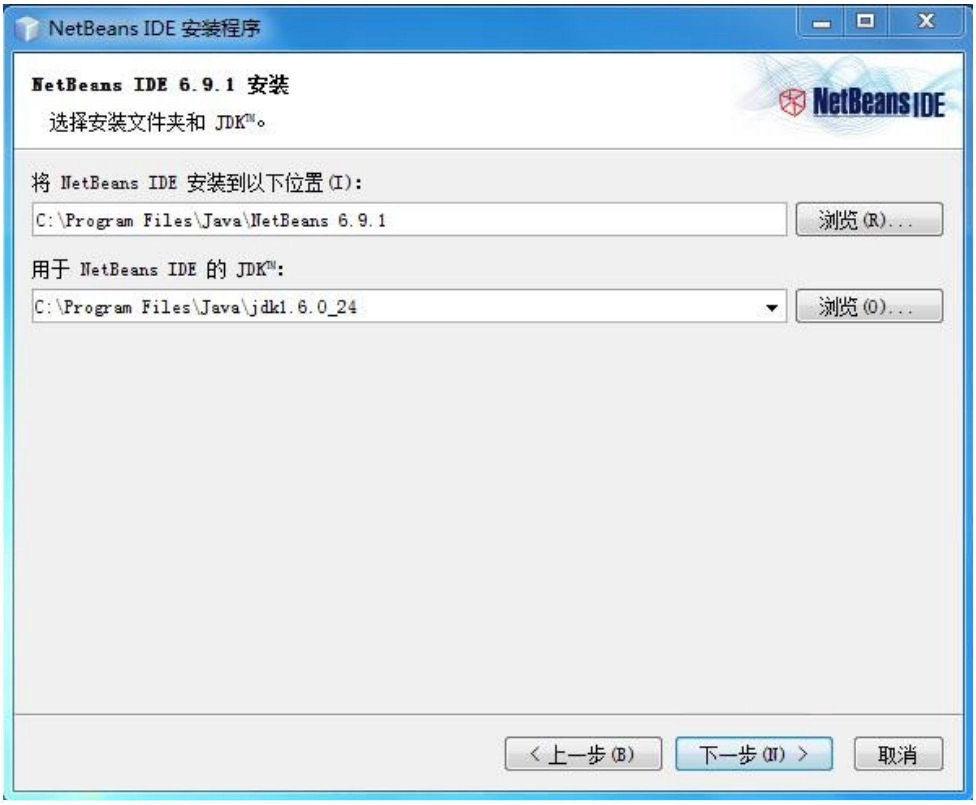
NetBeans IDE 安装程序正在初始化，请稍候

正在配置安装程序...

100%

Figure 4.7: NetBeans IDE initialization

Secondly, we must select some proper routes for NetBeans IDE 6.9.1 and JDK 6u24, which is described in Figure 4.8. Then, choose Next Step button to install the programming environment.



NetBeans IDE 安装程序

NetBeans IDE 6.9.1 安装

选择安装文件夹和 JDK™。

将 NetBeans IDE 安装到以下位置(I):

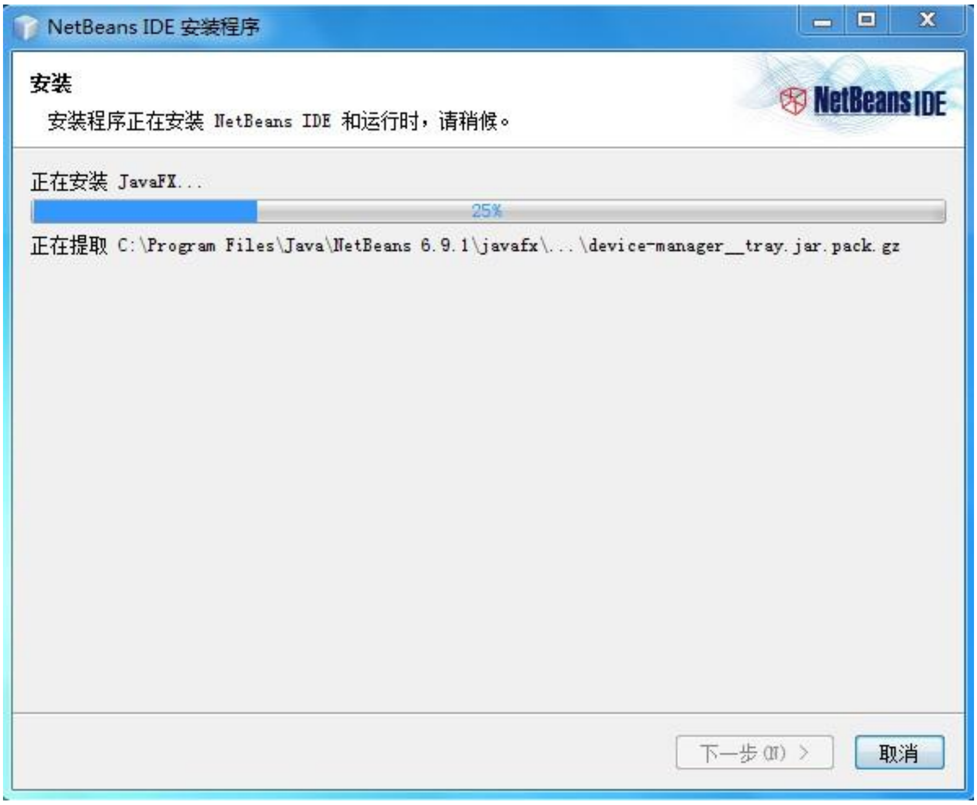
C:\Program Files\Java\NetBeans 6.9.1

用于 NetBeans IDE 的 JDK™:

C:\Program Files\Java\jdk1.6.0_24

Figure 4.8: Specifying their routes

In this step, we can experience the process of NetBeans installation as Figure 4.9 shown. It requires us to wait this process running for a little bit longer time.



安装

安装程序正在安装 NetBeans IDE 和运行时，请稍候。

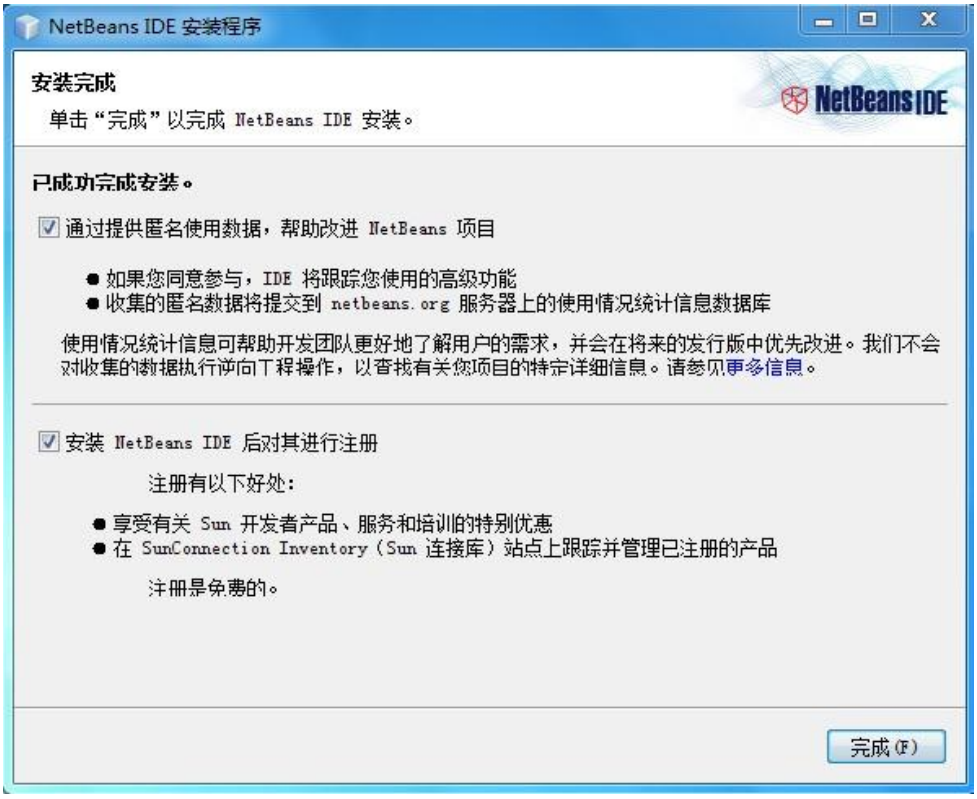
正在安装 JavaFX...

25%

正在提取 C:\Program Files\Java\NetBeans 6.9.1\javafx\...\device-manager__tray.jar.pack.gz

Figure 4.9: Process of NetBeans installation

At last, NetBeans IDE 6.9.1 installation has been completed successfully as can be seen from Figure 4.10. Thereafter, we may just select the Finish button so as to run NetBeans IDE 6.9.1 automatically.



安装完成

单击“完成”以完成 NetBeans IDE 安装。

已成功完成安装。

☑ 通过提供匿名使用数据，帮助改进 NetBeans 项目

- 如果您同意参与，IDE 将跟踪您使用的高级功能
- 收集的匿名数据将提交到 netbeans.org 服务器上的使用情况统计信息数据库

使用情况统计信息可帮助开发团队更好地了解用户的需求，并会在将来的发行版中优先改进。我们不会对收集的数据执行逆向工程操作，以查找有关您项目的特定详细信息。请参见更多信息。

☑ 安装 NetBeans IDE 后对其进行注册

注册有以下好处:

- 享受有关 Sun 开发者产品、服务和培训的特别优惠
- 在 SunConnection Inventory（Sun 连接库）站点上跟踪并管理已注册的产品

注册是免费的。

Figure 4.10: NetBeans installation is completed

5 WORKING PRINCIPLE OF THE GAME

5.1 Description of Project

In this chapter, I mainly introduce the project I designed, and some regulations and principles are also explained so as to make individuals understand clearly.

Chinese Chess game, also called Xiang Qi, is an extremely popular game in Asia with a long history, especially in China. Actually, Chinese Chess is quite similar to Western Chess, the general idea is identical. Each player controls an army of pieces, moves one piece at a time, and tries to get the opponent's "general" piece. It differs from Western Chess mainly in the board and the pieces. (Xiangqi [referred 10.04.2011])

5.1.1 Introduction to Board

Traditional Chinese Chess is played on a board that is a grid of nine lines wide and ten lines long. In a manner similar to the game Go, the pieces are played on the intersections, and the vertical lines are interrupted by the River between the fifth and sixth horizontal lines. Although the River provides a visual division between the two sides, only a few pieces are affected by its presence: "soldier" pieces have an enhanced move after crossing the river, while "elephant" pieces cannot cross. (Xiangqi_board [referred 10.04.2011]) There are also two Palaces at each side which is located by an X-shaped cross connecting its four corner intersections, which means the "general" and "advisors" should be stagnated in this area. Figure 5.1 below illustrates the board of Chinese Chess.

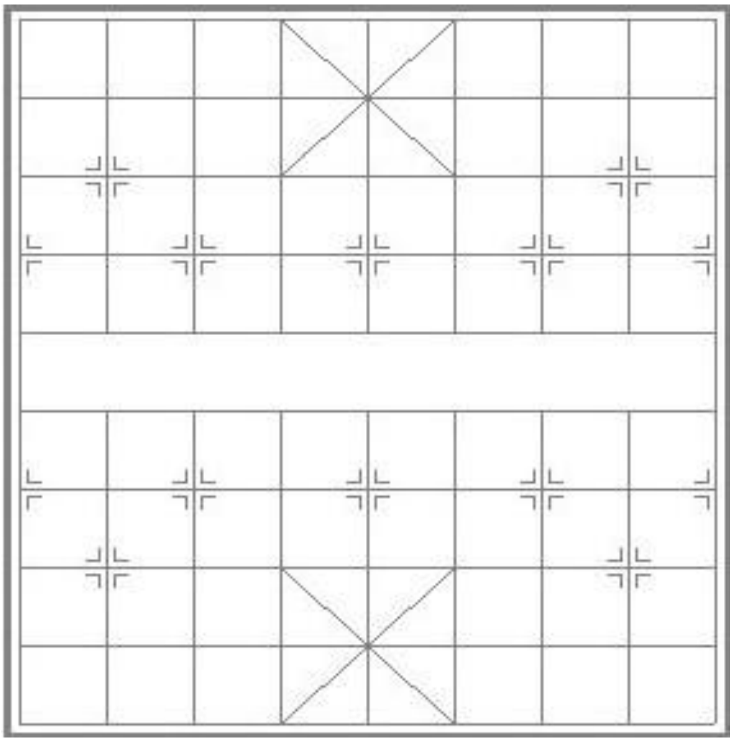


Figure 5.1: Board of Chinese Chess (Learning how to Play Chineses Chess [referred 10.04.2011])

5.1.2 Introduction to Pieces

Chinese Chess involves seven kinds of pieces, each side has one General, two Advisors, two Elephants, two Horses, two Chariots, two Cannons, and five Soldiers respectively. (Yen & Chen & Yang & Hsu [referred 15.04.2011] Next, I explain their location and principle in detail.

5.1.3 Chariots

The chariots are represented "車" for black and "俥" for red, which can move vertically and horizontally anywhere, but cannot jump over any pieces. The chariots are located at the four corners of the board initially.

5.1.4 Horses

The horses are stand for "馬" for black and "傌" for red, which are situated next to the chariots at the start. They can move vertically or horizontally one point and then diagonally away from its previous position. Also, the horses cannot jump over any pieces. We should be reminded that if there is a piece located one point away from the horse, the horse cannot move directly because the path is blocked. Figure 5.2 demonstrates the various situations on the horse's movements.

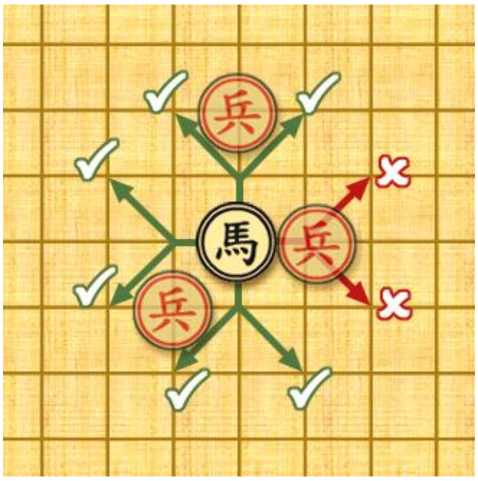


Figure 5.2: Movements of Horse in different situations (Horse movement [referred 15.04.2011])

5.1.5 Elephants

The elephants are labeled "象" for black and "相" for red, which are located next to the horses firstly. They can move two points diagonally like the format of 田. In addition, the elephants cannot jump over any pieces, and more critically, they must not cross over the river.

5.1.6 Advisors

The advisors are known as "士" for black and "仕" for red, which are mainly used for protecting the general from getting any threats. The advisors begin at the both sides of the general, they can only move diagonally within the palaces on both sides.

5.1.7 Generals

As we know, the generals are the royal pieces on the board. They are on behalf of "將" for black and "帥" for red, which are located in the middle of the board bottom at the beginning, each player has only one general. They can only move vertically or horizontally within the palaces on both sides. They are the most significant pieces on the board.

5.1.8 Cannons

The cannons stand for “炮” for black and “砲” for red, which are initially situated on the row behind the solders while in front of the horses, each side has only two the cannons. They can move vertically and horizontally anywhere like the chariots, while the only difference is that they must move by jumping over one piece anyway. Figure 5.3 shows the regulation of the cannons movement.

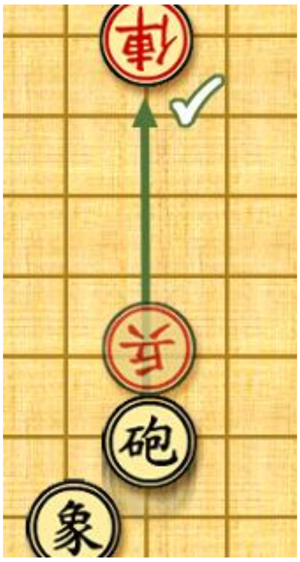


Figure 5.3: Rules of Cannons movement (Cannon movement [referred 15.04.2011])

5.1.9 Soldiers

The soldiers are usually called “卒” for black and “兵” for red, there are five pieces at each side located in one row behind the river. They can only move forward at own side. However, when they cross over the river, they are more functional, which can move virtually and horizontally without any restrictions.

Figure 5.4 below illustrates the pieces’ location on a board at the beginning.

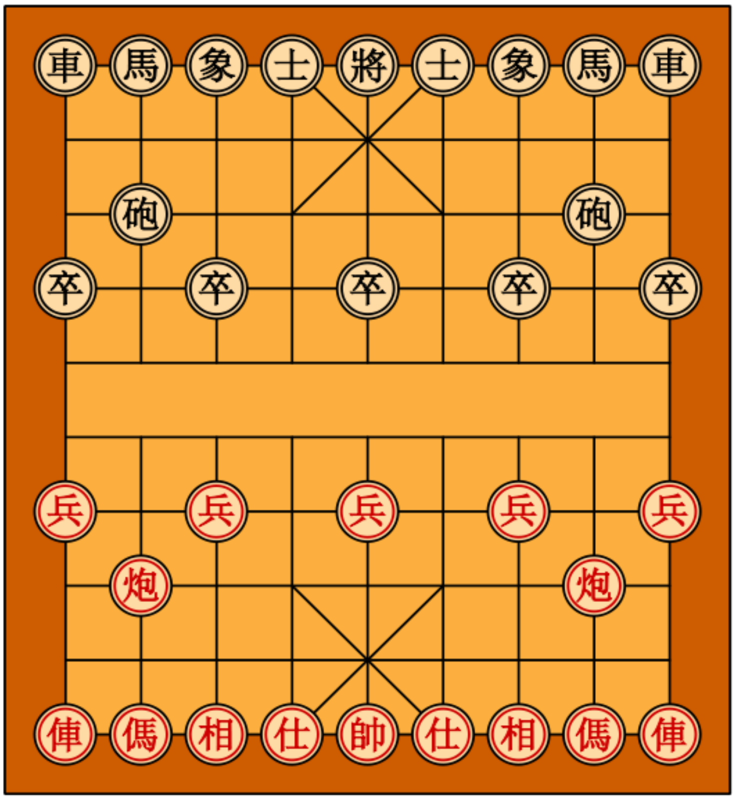


Figure 5.4: Pieces of Chinese Chess in their starting positions [38]

5.2 Structure of Project

This is a Chinese Chess game. When the game starts, the chess board may be appeared on the touch screen, as well as the chess pieces, which can be moved anywhere we want based on the principle. As the flow chart (Figure 5.5) shows, the structure of the project can be divided into three modules – Main module, Game module, and Result module.

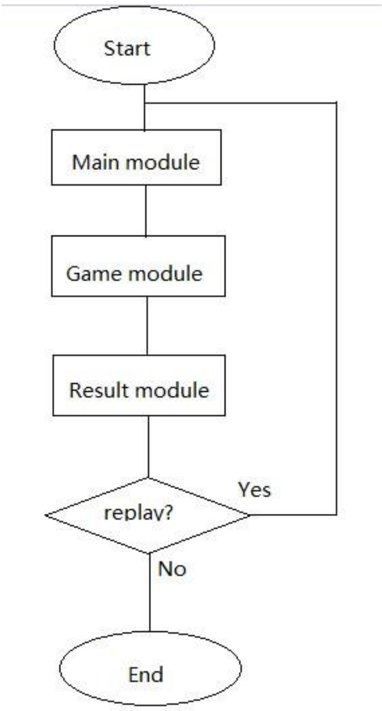


Figure 5.5: Flow chart of whole project

The main idea of my project has been illustrated as this flow chart described, like any other Java mobile phone games, this Chinese Chess game may be also separated into two main partitions: user interface and flow control. Basing on this principle, we focus on our own part in this joint project respectively. In the next chapter, I mainly emphasize our allocation and my key research field.

6 DIVISION OF THE PROJECT

First of all, I briefly point out the logical control for response and process in the project. Because Chinese Chess game has many regulations and principles. Developers should have a strong logical thinking ability. For instance, the rules of all the pieces which I have mentioned earlier, and how to choose the piece step by step, have become big challenges for us.

Some specific programming description is illustrated by my partner in his thesis, mainly because his strong logical capability. In the next section, I just introduce my key study field. The user interface components in this Chinese Chess game.

6.1 Introduction

In this section, I briefly emphasize the interaction for human-machine interface, which contains some user interface components. Names of these components look very familiar to AWT (Abstract Window Toolkit) programmer. There are also various APIs defined by MIDP 2.0 and CLDC 1.1. Next, I mainly explain the structure and description of Java APIs for User Interface (JSR118).

6.2 Structure of UI Package

In terms of the UI API, it can provide an approach to display the pictures or characteristics on the mobile phone screens, which can be logically composed of two kinds of APIs: high-level user interface and low-level user interface. As we know, the UI API affords a series of functions for implementation of user interfaces for the MIDlets, and the javax.microedition.lcdui packet consists of several public interfaces and a number of classes as well. In the next section, I mainly demonstrate some important interfaces and classes. Furthermore, I will illustrate the principles and regulations of these significant components in detail.

6.3 Interfaces and Classes of UI Package

The UI package is javax.microedition.lcdui, and there are four public interfaces related to this packet: Choice, CommandListener, ItemCommandListener, and ItemStateListener. (User

Interface Package [referred 20.04.2011]) In terms of this project, I use CommandListener as the interface to implement my MIDlet.

Furthermore, the UI package can also be divided into several public classes such as Canvas, Command, Display, Displayable, Form, Graphics, Screen, List, etc. (User Interface Package [referred 20.04.2011]) In terms of low-level UI APIs, the classes can be separated into Canvas and Graphics. However, when it comes to this project, I use GameCanvas and Graphics as the main classes to implement my MIDlet, and GameCanvas is inherited from Canvas which is defined by MIDP 1.0.

6.3.1 Interface CommandListener

In my partition, I use CommandListener to implement the MIDlet. This public interface is used by applications which need to receive high-level events from the implementation. (Interface CommandListener [referred 20.04.2011]) It contains a method called commandAction(Command command, Displayable displayable), which means when we press the key on Displayable d, this command will be invoked.

6.3.2 Class GameCanvas

First of all, we should know that the GameCanvas class is a subclass of lcdui's Canvas, and provides the basic 'screen' functionality for the user interface. (Game API Package [referred 20.04.2011]) It belongs to MIDP 2.0, yet Canvas class belongs to MIDP 1.0. The functions of them are all displaying pictures or characteristics on the mobile phone screens. In addition to the features inherited from Canvas, GameCanvas can also provide game-specific capabilities such as an off-screen graphics buffer and the ability to query key status. (Class GameCanvas [referred 20.04.2011]) After that, I could illustrate several critical components with regard to it.

GameCanvas can be divided into a range of fields, like DOWN PRESSED, GAME A PRESSED, LEFT PRESSED, and so on. They all stand for some specified keys on the mobile phones.

The constructor of GameCanvas class is GameCanvas(boolean suppressKeyEvents). It represents a framework which can help us to create a new example of a GameCanvas.

There are also various methods related to GameCanvas, such as paint(Graphics g), addCommand(Command command), keyPressed(int keyCode), pointerDragged(int x, int y), etc. In the next phase, I briefly describe some of them to make individuals understand with reference to my project.

public void *setCommandListener(CommandListener 1):*

This method sets a listener for Commands to the Displayable, replacing any previous CommandLiatener.

public void *addCommand(Command command):*

This method may let us add a command to the Displayable.

public void *Paint (Graphics g):*

This method can help us to paint the GameCanvas by default.

protected void *keyPressed(int keyCode):*

This method might be called when a key is pressed.

protected void *pointerDragged(int x, int y):*

This method would be called when the pointer is dragged.

6.3.3 Class Graphics

Generally speaking, public class Graphics affords a simple two dimensional geometric rendering capability, which can also be applied into a variety of fields, like BASELINE, BOTTOM, LEFT, VCENTER, etc. What is more, Graphics class contains numerous kinds of methods as well, such as drawRect(int x, int y, int width, int height), fillRect(int x, int y, int width, int height), setColor(int RGB), drawString(String string, int x, int y, int anchor), and so on. (Class Graphics [referred 20.04.2011]) Next, I generally demonstrate some of these methods to make individuals understand with regard to my project.

public void *drawLine(int x1, int y1, int x2, int y2):*

This method help us to draw the specified line between the coordinates (x1,y1) and (x2,y2) through the current color and style.

public void *setColor(int red, int green, int blue):*
This method sets the current color to some designated RGB values.

public void *fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):*
This method can fill a circular or elliptical arc covering the specified rectangle.

public void *setFont(Font, font):*
This method may set the font for all the contiguous text rendering operations.

public void *drawstring(String string, int x, int y, int anchor):*
This method can help us to draw the specified String using the current font and color. The x, y position is the position of the anchor point.

6.4 Description of User Interface API

In this section, I illustrate my design briefly and explain it step by step, the main codes and some descriptions could be demonstrated as well.

6.4.1 Game. java

In my project, I use GameCanvas to draw the board and pieces on the mobile phone instead of invoking pictures, which requires a strongly logical programming capability but can save the space, as well as make the algorithm simpler. First of all, I build a main class Game, and define the parameters and life-cycle. Some explanations have been mentioned in the previous chapter. The structure of the programme is demonstrated in Code example 6.1. This code is also a part of my whole application.

Code example 6.1:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Game extends MIDlet {
    GameCanvas game; // Define the object from class GameCanvas
    public Game() {
        super();
        game=new GameCanvas(this); // Generate the object from class GameCanvas
    }
    protected void startApp() {
        Display.getDisplay(this).setCurrent(game); // Draw the user interface on the screen
    }
    protected void pauseApp() {
    }
    protected void destroyApp(boolean unconditional) {
    }
}
```

6.4.2 GameCanvas. java

Thereafter, I could get into the key field --- class GameCanvas, which is the most significant section on the user interface design. Through using a two-dimensional array, I can implement the initialization of board and pieces on the phone screen. More specifically, I initiate the intersections to non-zero in which the pieces are occupied, while initiate the intersections to zero in which the pieces are not occupied. As we can see from the code in Appendices, the graphic outputs and pieces movements are all based on this two-dimensional array.

In this case, I guess it is necessary to introduce some items in terms of Arrays. As we know, an array is a collection of variables of the same types, which is always used for a variety of purposes because it can offer a convenient means of grouping together related variables. (Mynttinen, 2011) To declare a one-dimensional array, we may use this general form:

```
type [ ] arrayName = new type [size];
```

For example, the following creates an *int* array of *10* elements and links it to an array reference variable named *sample*:

```
int [ ] sample = new int [10];
```

In fact, when we represent a one-dimensional array, sometimes we do not have to list all the values one by one. There is an easier approach to initialize an array:

```
type [ ] arrayName = {val1, val2, val3, ……};
```

When it comes to my design, I use the simplest form of the multidimensional array, two-dimensional array. The format is more or less similar to one-dimensional array, which can be indicated as follow:

```
type [ ][ ] arrayName = new type [size][size];
```

Similarly, to declare a two-dimensional *integer* array *details* of size *10, 20* (10 rows and 20 columns), we can use this form to indicate it:

```
int [ ] [] details = new int [10] [20];
```

However, if we initiate the board by using this general format related to my application, it would be a little bit longer and more complicated. So we can use a shorter and simpler way to initiate this int two-dimensional array just like the easier approach in one-dimensional array which I referred previously. The following Code example 6.2 is the initialization of this two-dimensional array with a simpler way which I came up with.

Code example 6.2:

```
protected int point [ ][ ] = { {1,2,3,4,5,6,7,8,9}, // Initiate an integer array
                               {0,0,0,0,0,0,0,0,0},
                               {0,10,0,0,0,0,0,11,0},
                               {12,0,13,0,14,0,15,0,16},
                               {0,0,0,0,0,0,0,0,0},
                               {0,0,0,0,0,0,0,0,0},
                               {28,0,29,0,30,0,31,0,32},
                               {0,26,0,0,0,0,0,27,0},
                               {0,0,0,0,0,0,0,0,0},
                               {17,18,19,20,21,22,23,24,25} }
```

Now that completing the initialization of this two-dimensional array related to the board, I could begin to design the initialization of all the pieces. Through executing if statements based on this two-dimensional array, I can implement the initialization of these characters as Code example 6.3 shown.

Code example 6.3:

```
for (i=0; i<10; i++) // Initiate all the characters
{
    for (j=0; j<9; j++)
    {
        if (i==0) { if (j==0) {word [i] [j] = "车";}
                    if (j==1) {word [i] [j] = "马";}
                    if (j==2) {word [i] [j] = "相";}
                    if (j==3) {word [i] [j] = "士";}
                    if (j==4) {word [i] [j] = "帅";}
                    if (j==8) {word [i] [j] = "车";}
                    if (j==7) {word [i] [j] = "马";}
                    if (j==6) {word [i] [j] = "相";}
                    if (j==5) {word [i] [j] = "士";} }
        if (i==1) {word [i] [j] = "空";}
        if (i==2) { if ((j!=1) & (j!=7)) {word [i] [j] = "空";}
                    if (j==1) {word [i] [j] ="炮";}
                    if (j==7) {word [i] [j] = "炮";} }
        if (i==3) { if (j%2==0) {word [i] [j] = "卒";}
                    if (j%2==1) {word [i] [j] = "空";} }
        if (i==4) {word [i] [j] = "空";}
        if (i==5) {word [i] [j] = "空";}
        if (i==6) { if (j%2==0) {word [i] [j] = "卒";}
                    if (j%2==1) {word [i] [j] = "空";} }
        if (i==7) { if ((j!=1) & (j!=7)) {word [i] [j] = "空";}
                    if (j==1) {word [i] [j] = "炮";}
                    if (j==7) {word [i] [j] ="炮";} }
        if (i==8) {word [i] [j] = "空";}
        if (i==9) { if (j==0) {word [i] [j] = "车";}
```

```
                if (j==1) {word [i] [j] = "马";}
                if (j==2) {word [i] [j] = "相";}
                if (j==3) {word [i] [j] = "士";}
                if (j==4) {word [i] [j] = "帅";}
                if (j==8) {word [i] [j] = "车";}
                if (j==7) {word [i] [j] = "马";}
                if (j==6) {word [i] [j] = "相";}
                if (j==5) {word [i] [j] = "士";} }
    }
}
```

After the initialization about the board and pieces, I could paint this GameCanvas step by step. First of all, the top half of the board, can be drawn in Code example 6.4 as follow. In this code, many of methods are used to configure the board, and the functions of all these methods can also be found in this chapter above. Similarly, the bottom half of the board can be set in the same way, I just need to modify some of the parameters.

Code example 6.4:

```
protected void paintMapa (Graphics g) // Draw the top half of board
{
    for (int q=0; q<4; q++)
    {
        for (int w=0; w<8; w++)
        {
            g.setColor (128, 128, 128);
            g.drawRect (x+w*cellWidth, x+q*cellWidth, cellWidth, cellWidth);
        }
    }
    g.setColor (128, 128, 128);
    g.drawLine (x+3*cellWidth, x, x+5*cellWidth, x+2*cellWidth);
    g.drawLine (x+5*cellWidth, x, x+3*cellWidth, x+2*cellWidth);
}
```

Furthermore, the River ”楚河”, ”汉界” should be painted by using *setColor, drawRect, setFont, and drawString* these four methods respectively. The following Code example 6.5 implements the River. Additionally, we should definitely know that the River interrupts the vertical lines between the fifth and sixth horizontal lines on the board.

Code example 6.5:

```
protected void paintMapb (Graphics g) // Paint the River ”楚河”, ”汉界”
{
        g.setColor (128, 128, 128);
        g.drawRect (x,x+4*cellWidth,mapWidth,cellWidth);
        g.setFont (Font.getFont (Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
                      Font.SIZE_LARGE));
        g.drawString (”楚河  汉界”, getWidth()/2, x+4*cellWidth+cellWidth*3/4, Graphics.
                            HCENTER|Graphics.BASELINE);
}
```

Last but not least, all the pieces could be put onto the board. Actually, the board is a grid of nine lines wide and ten lines long, and all the pieces are played on the specified intersections as the regulations. The Code example 6.6 is illustrated as follow.

Code example 6.6:

```
protected void paintAllChess (Graphics g) // Draw all the pieces
{
        for (i=0; i<10; i++)
        {
                for (j=0; j<9; j++)
                {
                    if (point [i] [j] != 0) {
                          if(point [i] [j] < 17) {g.setColor (255, 0, 0);}
                          else{g.setColor (255, 255, 255);}
                          g.fillArc (x-chessR+j*cellWidth, x-chessR+i*cellWidth, 2*chessR,
                                    2*chessR, 0, 360);
                          g.setColor (0x00000000);
                          g.setFont (Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
                                      Font.SIZE_LARGE));
                    g.drawString (word[i][j], x+j*cellWidth, x+chessR+i*cellWidth,
                            Graphics.HCENTER|Graphics.BOTTOM);
          }
        }
    }
}
```

In conclusion, all the Java programming of the board and pieces can be demonstrated by using User Interface API. Thereafter, as far as the basic control in terms of the regulations and principles for the developer, my partner would point it out in his final thesis specifically.

7 IMPLEMENTING THE PROGRAMMING

7.1 Building a Programming Environment

In this case, NetBeans IDE 6.9.1, the newest version, can be used as my programming framework, which can be downloaded for free from www.NetBeans.org. Because my operating system language is Chinese by default, I can only select Simplified Chinese as my operating language.

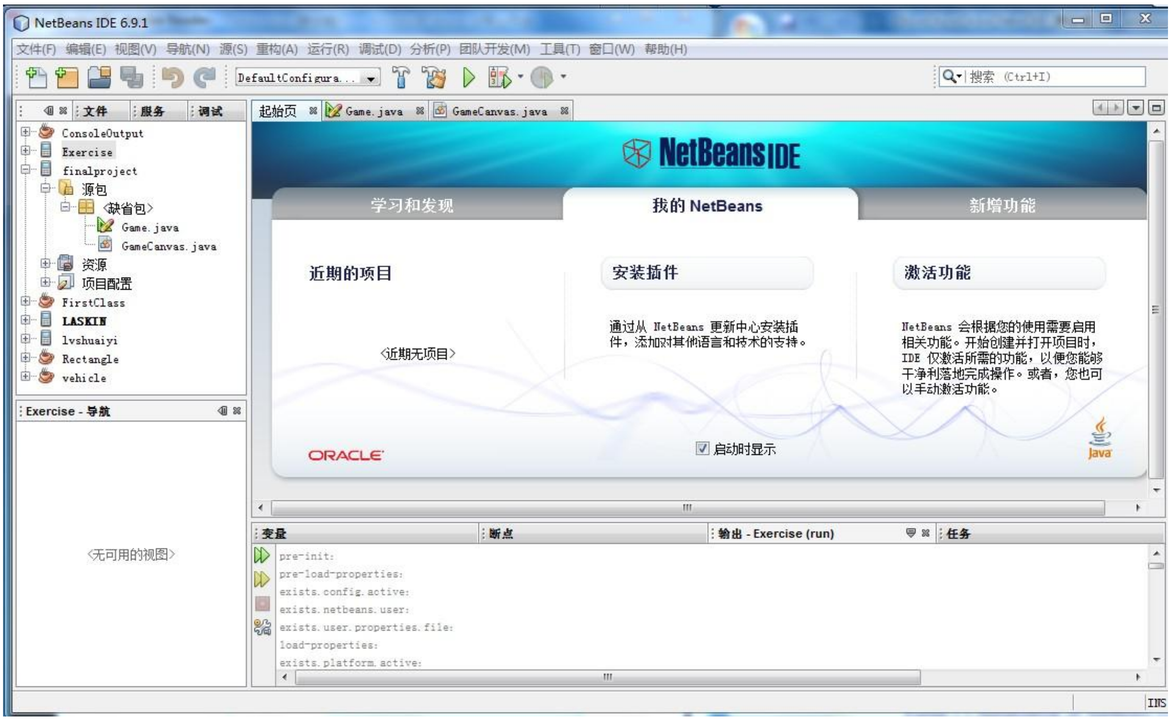
As I mentioned previously, Java Development Kit (JDK) is very essential for my application as well, the installation process has also been demonstrated in Chapter 4. Only after I complete the installation of JDK, I can begin to work with my NetBeans IDE.

Now we can run our project flexibly under NetBeans programming environment, but if we want to simulate or debug my applications in the project, Java Software Development Kit (SDK) is indeed required. In Windows, Java ME SDK 3.0 is the successor to the popular Java Wireless Toolkit 2.5.2 and Java Toolkit 1.0 for CDC. It integrates CLDC, CDC and Blu-ray Disc Java (BD-J) technology into one SDK.(Java ME Downloads [referred 25.04.2011]) So Java SDK 3.0 consists of all the APIs we need, we can use the emulator and debug the MIDlets ahead of my downloading this applications to my mobile phone.

After the installation of JDK, NetBeans IDE, and Java SDK, the entire programming and test environment are built successfully.

7.2 Creating a Project

First of all, I initiate NetBeans IDE 6.9.1 in order to perform my application and get access into NetBeans development platform. Figure 7.1 illustrates the initial layout of NetBeans IDE 6.9.1.



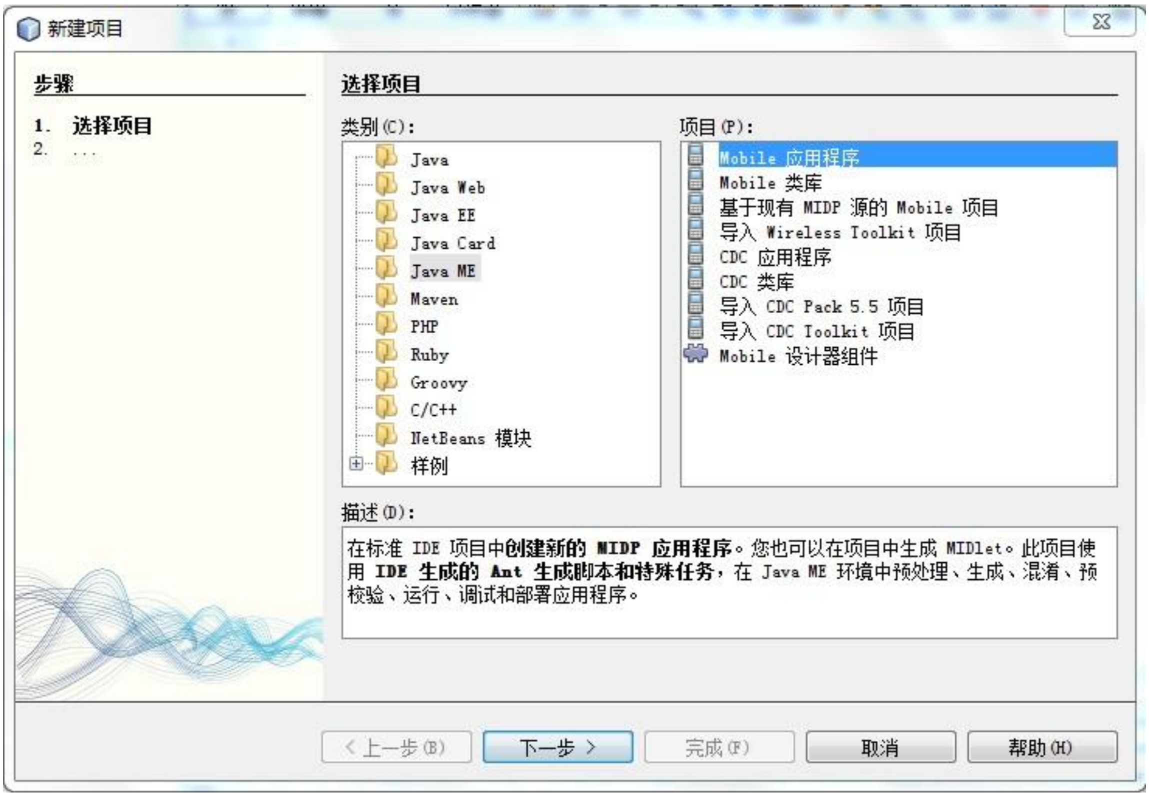
通过从 NetBeans 更新中心安装插件，添加对其他语言和技术的支持。

NetBeans 会根据您的使用需要启用相关功能。开始创建并打开项目时，IDE 仅激活所需的功能，以便您能够干净利落地完成操作。或者，您也可以手动激活功能。

```
pre-init:
pre-load-properties:
exists.config.active:
exists.netbeans.user:
exists.user.properties.file:
load-properties:
exists.platform.active:
```

Figure 7.1: Initialization of NetBeans IDE

Thereafter, I need to create a project. Under NetBeans environment, I click the File button and then select the order of New Project - Java ME - Mobile application. Figure 7.2 below demonstrates the sequence what I described.

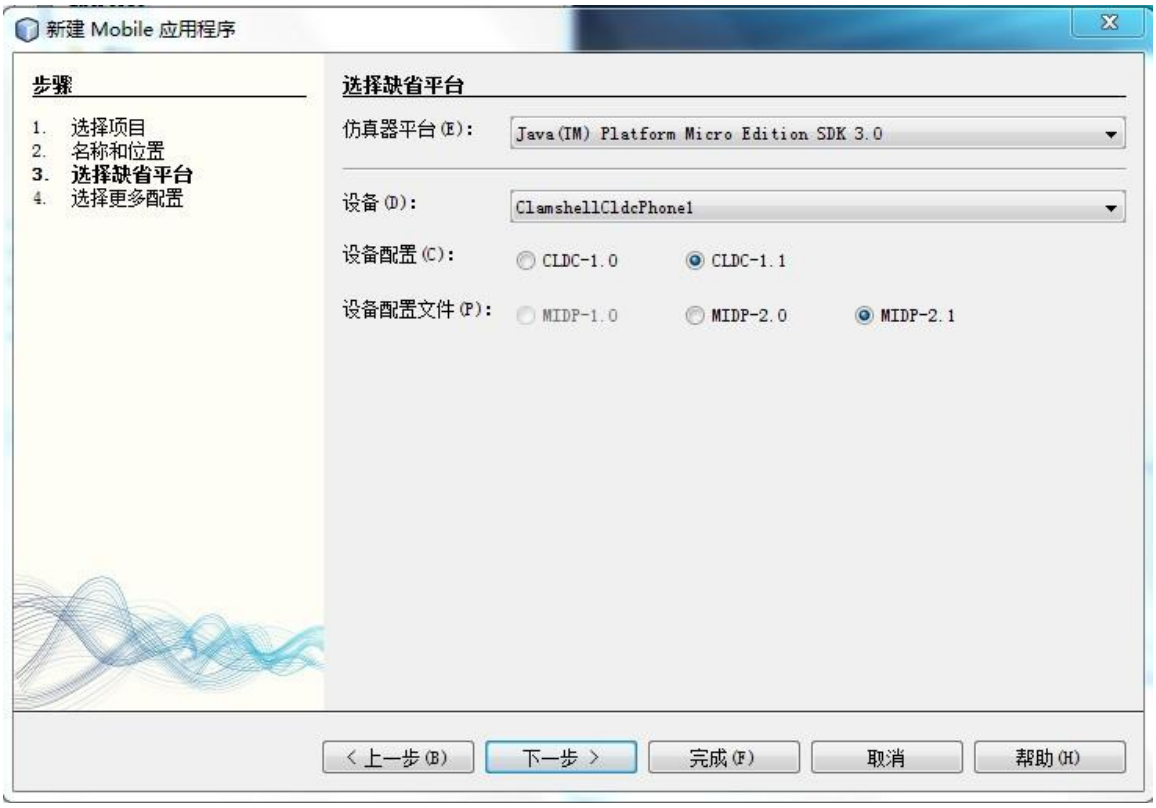


在标准 IDE 项目中**创建新的 MIDP 应用程序**。您也可以在项目中生成 MIDlet。此项目使用 **IDE 生成的 Ant 生成脚本和特殊任务**，在 Java ME 环境中预处理、生成、混淆、预校验、运行、调试和部署应用程序。

Figure 7.2: Creating a new project

Thirdly, I name the project as “Hello”, and also choose to create a Hello MIDlet in order to help me to get more familiar with this development platform.

For the next step, I choose the simulation platform and a series of default configuration. As can be seen from Figure 7.3, the simulation platform is regarded as Java SDK 3.0, the device configuration would be CLDC-1.1, and the device configuration profile could be MIDP-2.1 respectively in order to be more compatible with the real device.



新建 Mobile 应用程序

步骤

1. 选择项目
2. 名称和位置
3. **选择缺省平台**
4. 选择更多配置

选择缺省平台

仿真器平台(E): Java(TM) Platform Micro Edition SDK 3.0

设备(D): ClamshellCldcPhone1

设备配置(C): ○ CLDC-1.0 ● CLDC-1.1

设备配置文件(P): ○ MIDP-1.0 ○ MIDP-2.0 ● MIDP-2.1

Figure 7.3: Default configuration of the device

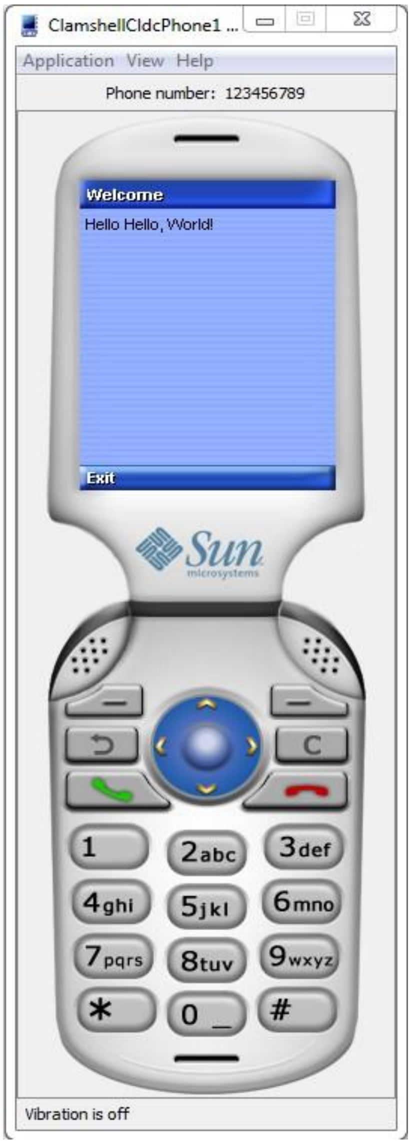
Finally, I click Finish button, and then the source code of “Hello” MIDlet has been illustrated in Code example 7.1 as follow:

Code example 7.1:

```
package hello;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloMIDlet extends MIDlet implements CommandListener {
```

```
private boolean midletPaused = false;

/* The HelloMIDlet constructor. */
public HelloMIDlet() {
}

/* Returns the display instance. */
public Display getDisplay () {
return Display.getDisplay(this);
}

/* Exits MIDlet. */
public void exitMIDlet() {
switchDisplayable (null, null);
destroyApp(true);
notifyDestroyed();
}

/* Called when MIDlet is started. */
public void startApp() {
        if (midletPaused){
        resumeMIDlet (); }
        else{
        initialize ();
        startMIDlet (); }
midletPaused = false;
}

/* Called when MIDlet is paused. */
public void pauseApp() {
midletPaused = true;
}

/* Called to signal the MIDlet to terminate. */
public void destroyApp(boolean unconditional) {
        }
}
```

Figure 7.4 below depicts the performance of this Hello MIDlet.



Phone number: 123456789

Welcome

Hello Hello, World!

Exit

Vibration is off

Figure 7.4: Performance on the emulator

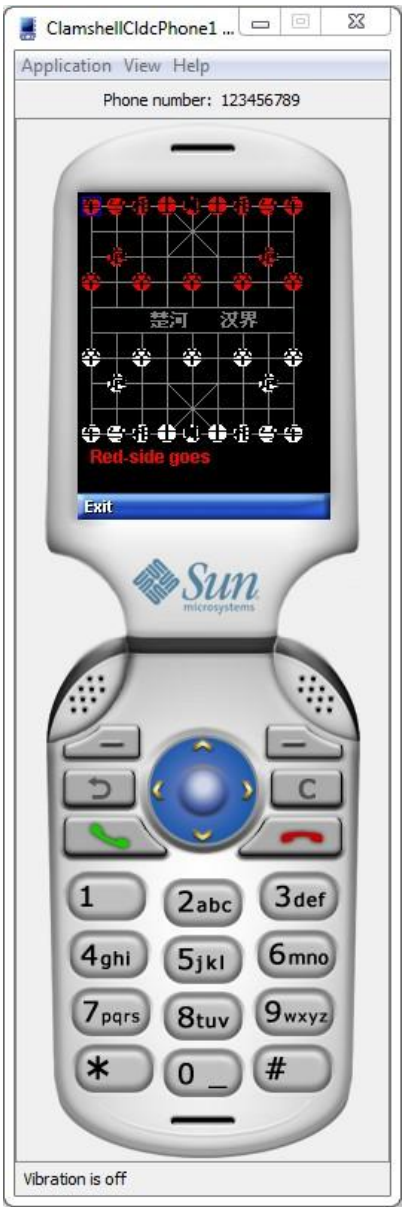
To summarize, this MIDlet can be performed normally and successfully, and also can be simulated on the emulator by setting the configuration and designing the source code. Next, I will put the implementation of my application.

7.3 Implementation of Project

In the first place, there is a start screen on the mobile phone, which is illustrated in Figure 7.5. In order to let individuals grasp and master this game faster, to realize how fun and easy this game is, and to distinguish both sides clearly, I decide to divide these two sides into Red background and White backgroud on each piece. Meanwhile, I make sure both sides of the pieces have the same characters correspondently. More specifically, ”车” represents Chariot, ”马” stands for Horse, ”相” represents Elephant, ”士” stands for Advisor, ”帅” represents General, ”炮” stands for Cannon, and ”卒” represents Soldier. All I mentioned above have also been explained in my programming codes in the previous chapter.

7.3.1 Start Module

As we can see from Figure 7.5, the top half of the board is Red-side, the bottom half of the board is White-side, and the middle of the board is the River ”楚河”, ”汉界”. According to my programming, Red-side player could choose anyone of the pieces initially. He can press left, right, up, and down key to move the choice box, and then press ”OK” button which is situated in the middle of the keyboard to confirm this choice. Next, I would control two sides respectively to play this game step by step so as to implement this project which my partner and me designed with each other.



Phone number: 123456789

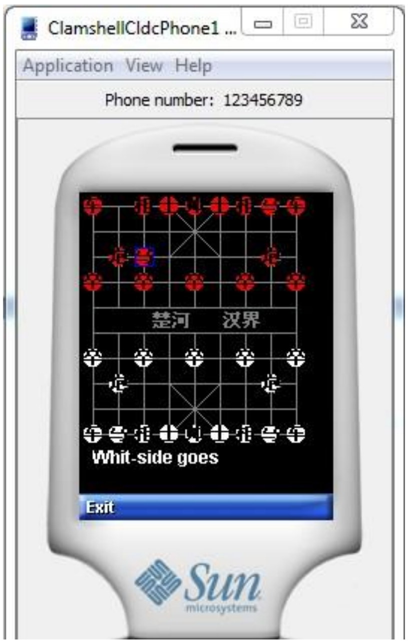
Red-side goes

Vibration is off

Figure 7.5: Start screen of the game

7.3.2 Game Module

According to the regulations of the pieces, Red-side's player firstly chooses "马" (Horse) which is located at the upper-left on the board, moves it vertically one point and diagonally away from its previous position. Figure 7.6 demonstrates the first step on Red-side from the emulator.

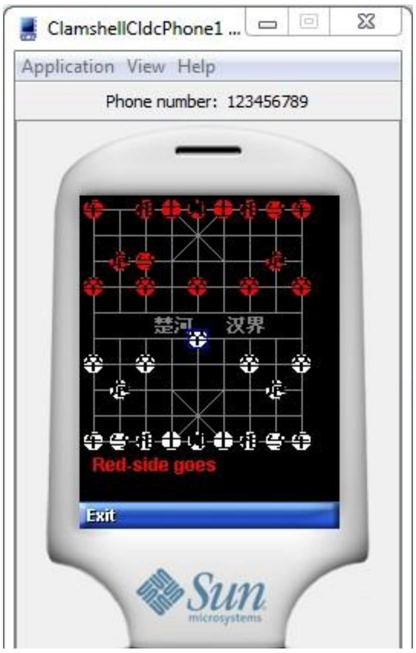


Phone number: 123456789

Whit-side goes

Figure 7.6: Step one on Red-side

In terms of the White-side, player would choose ”卒” (Soldier) which is situated in the middle of all the Soldiers in the bottom half of board to move forward one place based to the regulations. Figure 7.7 depicts the first step on White-side player.

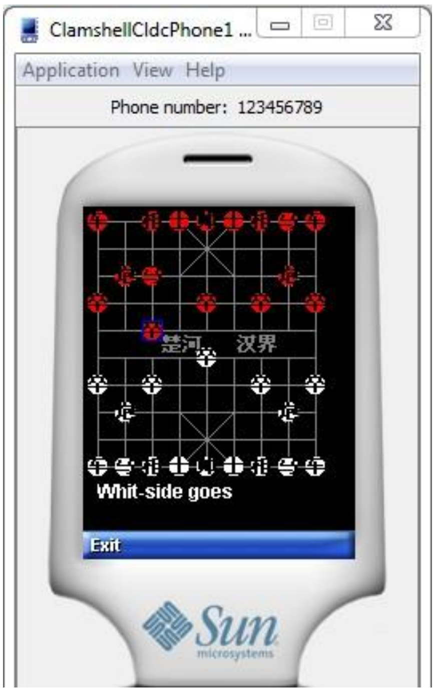


Phone number: 123456789

Red-side goes

Figure 7.7: Step one on White-side

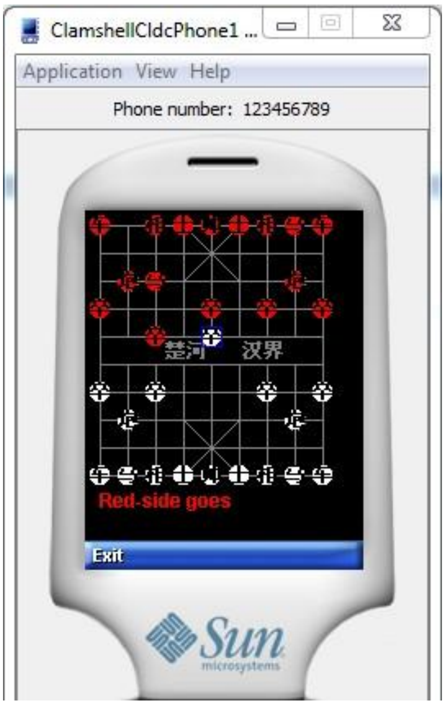
When it comes into Red-side again, player would also like to choose ”卒” (Soldier) which is located at the second order from the upper-left Soldiers in the top half of board to move forward one point. Figure 7.8 illustrates the second step on Red-side in the simulation.



Whit-side goes

Figure 7.8: Step two on Red-side

Talking about the second step on White-side, player makes the previous Soldier to move forward one point again to cross the River so as to be more functional and flexible. The output is displayed on the screen in Figure 7.9.



Red-side goes

Figure 7.9: Step two on White-side

As far as the Red-side in the next step, user chooses the former Horse to jump one path once more, and be ready to cross the River, which is described in Figure 7.10.

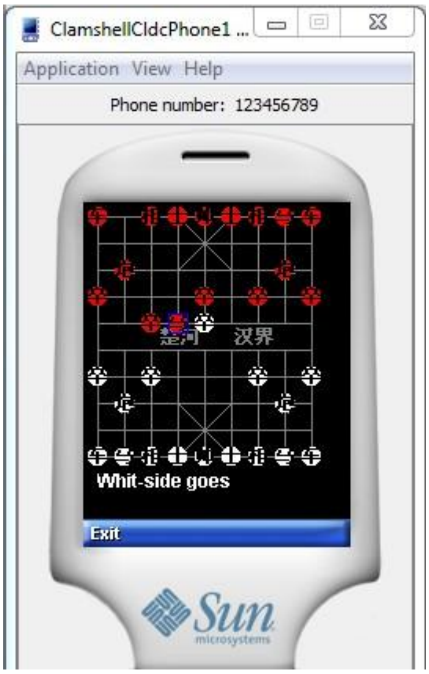


Figure 7.10: Step three on Red-side

Speaking of the third step, White-side's player selects previous Soldier to move forward more again. Meanwhile, this White-side's Soldier eliminates one of Red-side's Soldiers which is in front of it. Figure 7.11 represents the result on the mobile phone screen.

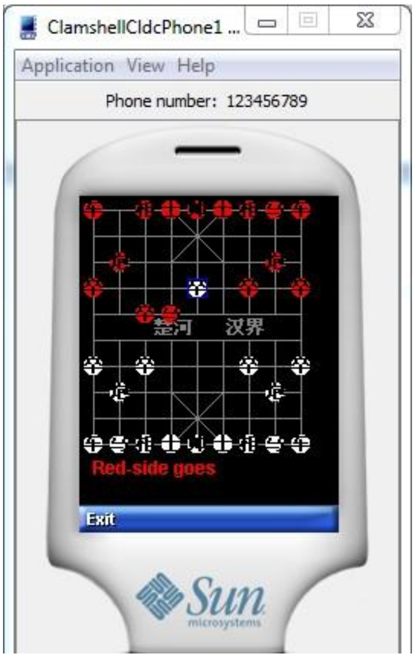
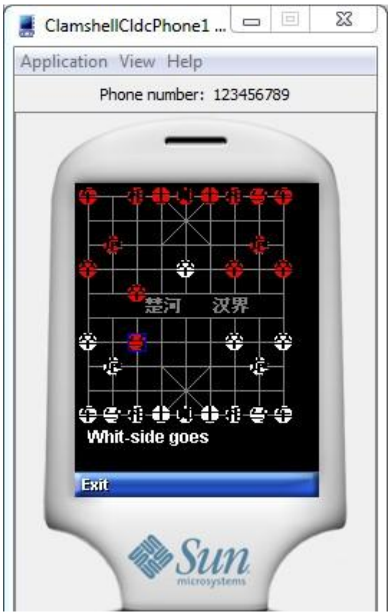


Figure 7.11: Step three on White-side

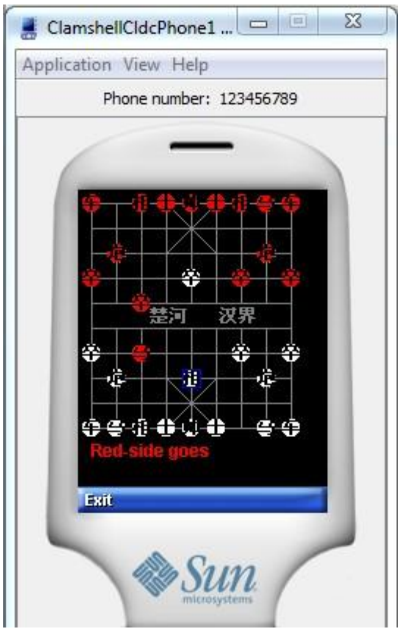
As Figure 7.12 shows, as soon as the White-side's player completes the third step, the former Horse of Red-side will jump one route to cross the River, and removes one of White-side's Soldiers which is located on the second order of lower-left Soldiers.



Whit-side goes

Figure 7.12: Step four on Red-side

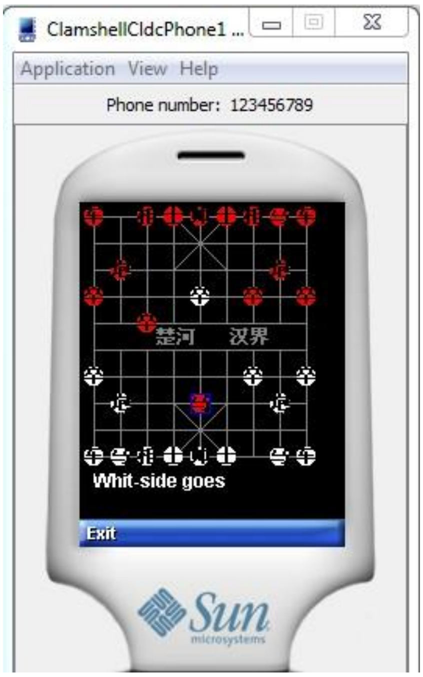
In order to prevent Red-side pieces from attacking General, player on White-side chooses "相" (Elephant) which is situated in the lower-right on the screen to move two points diagonally. Figure 7.13 draws the fourth step on White-side from the emulator.



Red-side goes

Figure 7.13: Step four on White-side

However, for the next step on Red-side, player uses the previous Horse to attack White-side's Elephant according to the principle. The Elephant of White-side is eliminated as Figure 7.14 illustrated.

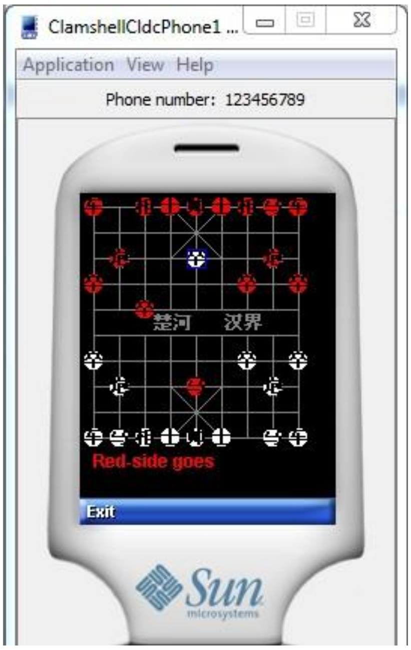


Phone number: 123456789

Whit-side goes

Figure 7.14: Step five on Red-side

In terms of the fifth step, the Soldier of White-side in the opposite side moves forward again in order to be as more possible as close to Red-side's General. Figure 7.15 presents the output of this step on the screen.



Phone number: 123456789

Red-side goes

Figure 7.15: Step-five on White-side

Now that the opposite Soldier gets access into the General closer and closer, Red-side’s player has to selects ”士” (Advisor) which is located at the upper-right onto the board to prevent General from removing, but it is more likely to sacrifice itself. Figure 7.16 demonstrates the process of Advisor movement, which can be only placed diagonally within its own palace.

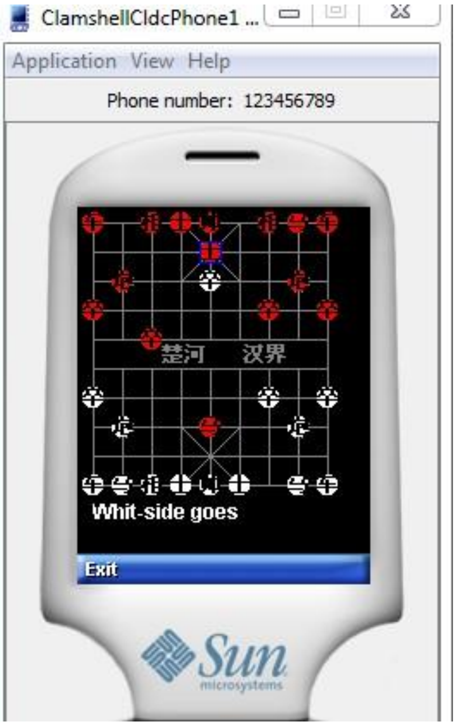
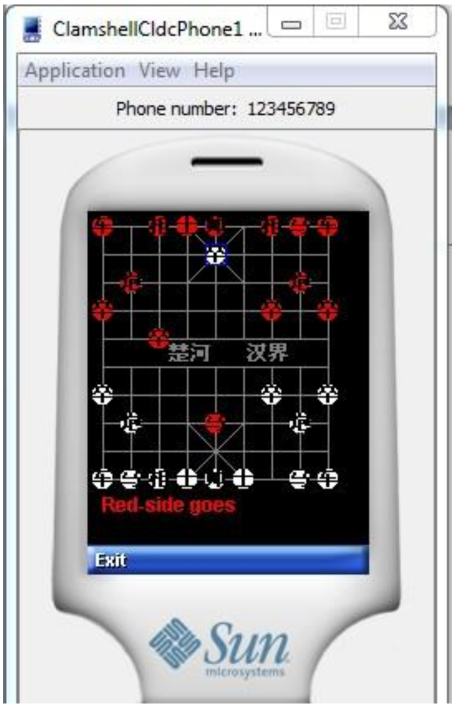


Figure 7.16: Step six on Red-side

Unfortunately, the previous Soldier of White-side clears up the Red-side’s Advisor eventually as Figure 7.17 shown. This can be seen that Soldier is more or less stronger than we thought, sometimes it is a useful weapon to attack opposite pieces.

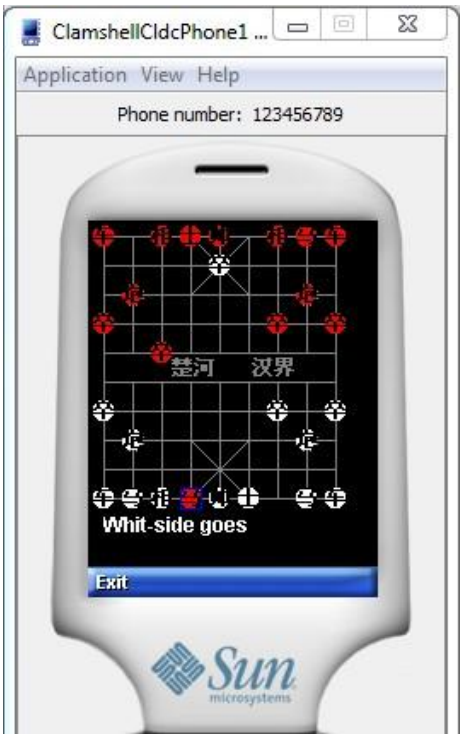


Phone number: 123456789

Red-side goes

Figure 7.17: Step six on White-side

In order to reverse this negative situation, Red-side's player have to attack even more fierce. So player uses the previous Horse to move one path to eliminate the Advisor of White-side. Figure 7.18 explains the seventh step in Red-side onto the screen.



Phone number: 123456789

Whit-side goes

Figure 7.18: Step seven on Red-side

When it comes to White-side, I guess it is time to use ”帅” (General) to remove opposite Horse. User could just choose his General which is located in the middle of bottom line to move one point horizontally within its palace, and then Red-side’s Horse is eliminated. Figure 7.19 demonstrates the result of this step.

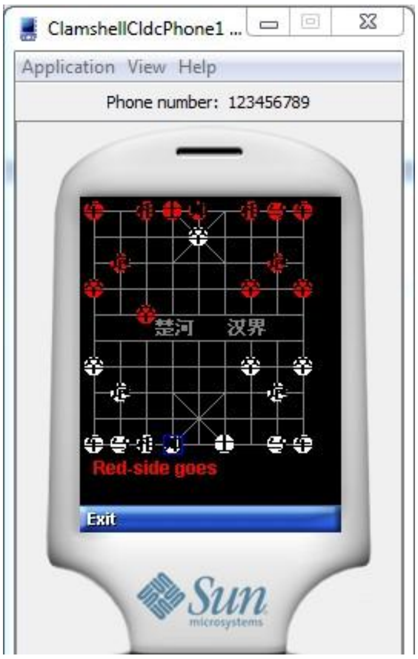


Figure 7.19: Step seven on White-side

As far as the Red-side in the eighth step, player selects the upper-right Elephant to move two points diagonally to avoid lost in this game, which can be illustrated in Figure 7.20 as follow.

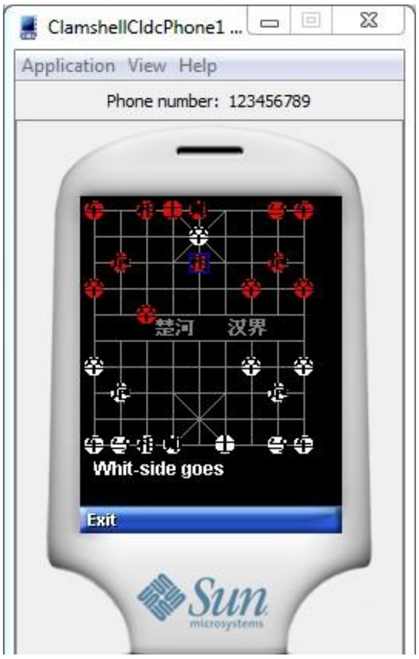
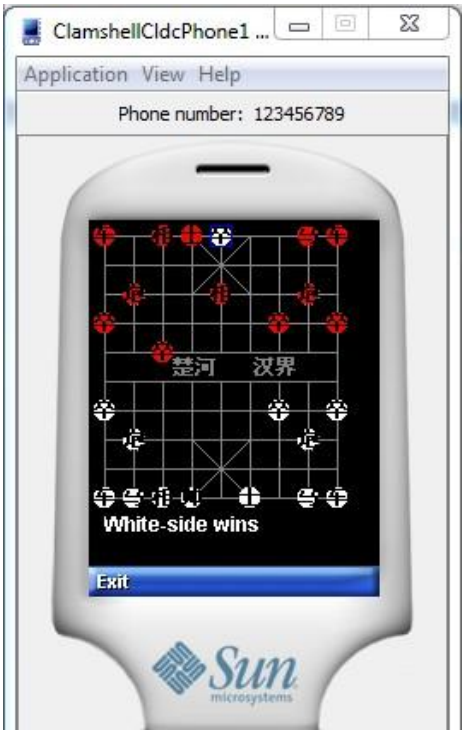


Figure 7.20: Step eight on Red-side

7.3.3 Result Module

Now, with the game progressing, we can access into the Result module. In terms of the eighth step on White-side, player can just select the previous Soldier to make it move virtually one point, and evenually eliminates the General of Red-side. Lastly, a text ”White-side wins” is displayed on the screen as Figure 7.21 shown.



White-side wins

Figure 7.21: Step eight on White-side

8 CONCLUSION

8.1 Purpose

Taking into account of all the aspects which are mentioned in previous chapters, the conclusion can be drawn is obviously that I have successfully developed a Chinese Chess game and tested it on the emulator. Actually, the ultimate purpose of my final thesis is to get more familiar with Java programming, build an application of mobile phone game by using Java Micro Edition, and utilize it to implement a Chinese Chess game.

8.2 Challenges

In this section, I would demonstrate some challenges and difficulties I faced, and the trouble-shootings with regard to my study. The critical part of my thesis is to figure out how to build a user interface by using UI API, and make all the pieces work on Java ME platform. Eventually, this Chinese Chess game can be displayed on the emulator successfully. However, when I download this application to my real mobile device in order to test it by using Nokia PC suite, there are some challenges related to my mobile phone that my application cannot be displayed on my mobile phone screen, which means there is an error message demonstrated onto the screen. I guess the main reason is that my mobile phone is not compatible with this application. As we know, Nokia RDA is a service provided by Nokia to remotely control various mobile devices from different manufacturers. I think if I choose a proper type of device through the Nokia Remote Device Access (RDA) platform firstly, maybe I could utilize that suitable and available device to test my application successfully. So the test result on a real mobile device still remains to be examined.

There is also another challenge which accounts for plenty of time during the study. That is the programming from my partner’s part of this joint project, especially the regulations of Chariots and Cannons. Unfortunately, these two kinds of pieces still cannot move forward so as to meet their requirements although we have modified and adjusted the programming codes many times.

Anyway, I think this huge project makes me more familiar with Java programming and helps me to grasp an essential skill for my Master degree and even my career although there are some troubles with regard to it.

8.3 Future Development

Last but not least, I need to discuss some prospects and further work which should be done in the near and far future related to my final thesis. Nowadays, Bluetooth communication has become more and more popular in more than one aspect. If it is possible, I guess it is necessary to apply this Chinese Chess game into multi-player mode by using Bluetooth or IrDA techniques based on real-time communication. I believe this type of mobile phone games will have a wide marketing perspective in the near future.

BIBLIOGRAPHY

Books

Hashim, Hafizullah Amin 2008. Analysis on Types of Mobile Games, First Edition, Printed in Malaysia, University of Malaya

Lemay, Laura and Perkins, Charles L. 1996. Teach Yourself JAVA in 21 Days, First Edition, Printed in the United States of American. George Bond

Course material

Mynttinen, Timo. 2011. Java Programming. Course Slides. Mikkeli University of Applied Sciences.

Koivisto, Matti. 2011. MIDLETS. Course Materials. Mikkeli University of Applied Sciences.

Electronic sources

Forum Nokia. [referred 13.03.2011]. Available in www.document:
<URL
http://www.nokia.com/about-nokia/company/story-of-nokia/mobile/revolution/snake-game>

Personal area network. [referred 14.03.2011]. Available in www.document:
<URL http://en.wikipedia.org/wiki/Personal_area_network>

Pilato, Fabrizio 2004. Bluetooth still coming on strong. [referred 14.04.2011]. Available in www-document:
<URL http://www.mobilemag.com/2004/05/04/bluetooth-still-coming-on-strong>

Bluetooth.svg. [referred 14.03.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/File:Bluetooth.svg>

Single-player games. [referred 14.03.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/Single-player_game#Single-player_games>

Baidu pictures. [referred 14.03.2011]. Avaiable in www-document:

<URL http://image.baidu.com/i?ct=503316480&z=&tn=baiduimagedetail&word=%B5%A5%BB%FA%CF%F3%C6%E5%D3%CE%CF%B7&in=455&cl=2&lm=-1&pn=8&rn=1&di=1656485040&ln=2000&fr=&fmq=&ic=0&s=0&se=1&sme=0&tab=&width=&height=&face=0&is=&istype=2>

Mobile games. [referred 19.04.2011]. Available in www-document:
<URL http://games.sina.com.cn/mobilegames/2004/05/052121865.shtml>

Java technology. [reference 24.03.2011]. Available in www-document:
<URL http://www.sun.com/smi/press/sunflash/2004-02>

Java.svg. [referred 24.03.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/File:Java_logo.svg>

Java platforms. [referred 24.03.2011]. Available in www-document:
<URL http://java.sun.com/javame/img/javame_components.gif>

Java SE. [referred 24.03.2011]. Available in www-document:
<URL http://www.oracle.com/technetwork/java/javase/overview/index.html>

J2ME Tutorial. [referred 24.03.2011]. Available in www-document:
<URL http://www.uberthings.com/mobile/>

Java EE. [referred 24.03.2011]. Available in www-document:
<URL http://www.oracle.com/technetwork/java/javaee/overview/index.html>

Java ME. [referred 26.03.2011]. Available in www-document:
<URL http://www.oracle.com/technetwork/java/javame/overview/index.html>

Connected Device Configuration. [referred 26.03.2011]. Available in www-document:
<URL http://www.oracle.com/technetwork/java/javame/tech/technology-139316.html#cdc>

Digital Media Platform. [referred 26.03.2011]. Available in www-document:
<URL http://java.sun.com/javame/img/digital_media_platform.jpg>

Connected Limited Device Configuration. [referred 26.03.2011]. Available in www-document:
<URL http://www.oracle.com/technetwork/java/javame/tech/technology-139316.html#cldc>

CLDC Wireless Platform. [referred 26.03.2011]. Available in www-document:
<URL http://java.sun.com/javame/img/wireless_platform.jpg>

Mobile Information Device Profile. [referred 26.03.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/Midp>

MIDP Overview. [referred 26.03.2011]. Available in www-document:
<URL http://java.sun.com/products/midp/overview.html>

MIDlet. [referred 26.03.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/Midlet>

Life-cycle. [referred 26.03.2011]. Avaiable in www-document:
<URL http://developers.sun.com/mobility/midp/articles/fsm/fsm_fig3.gif>

Eclipse (software). [referred 02.04.2011]. Availbale in www-document:
<URL http://en.wikipedia.org/wiki/Eclipse_(software)>

Eclipse IDE. [referred 02.04.2011]. Available in www-document:
<URL http://upload.wikimedia.org/wikipedia/commons/a/a2/Eclipse_3.6_Helios.jpg>

NetBeans. [referred 05.04.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/NetBeans>

NetBeans IDE. [referred 05.04.2011]. Available in www-document:
<URL http://netbeans.org/images_www/v6/9/screenshots/java-card.png>

NetBeans Downloads. [referred 05.04.2011]. Available in www-document:
<URL http://netbeans.org/downloads/index.html>

Xiangqi. [referred 10.04.2011]. Available in www-document:
<URL http://www.chessvariants.com/xiangqi.html>

Xiangqi_board. [referred 10.04.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/Xiangqi#Board>

Learning how to Play Chineses Chess. [referred 10.04.2011]. Available in www-document:
<URL http://web.pdx.edu/~trietn/CAFE%20CHINESE%20CHESS.htm>

Yen, Shi-Jim and Chen, Jr-Chang and Yang Tai-Ning and Hsu Shun-Chin 2004. Computer Chinese Chess [referred 15.04.2011]. Available in pdf-document

Horse movement. [referred 15.04.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/File:MovementOfHorsePiece.png>

Cannon movement. [referred 15.04.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/File:XiangqiCannonJumpToCapture.png>

Xiangqi Board.svg. [referred 15.04.2011]. Available in www-document:
<URL http://en.wikipedia.org/wiki/File:Xiangqi_Board.svg>

User Interface Package. [referred 20.04.2011]. Available in www-document:
<URL
http://download.oracle.com/javame/config/cldc/ref-impl/midp2.0/jsr118/javax/microedition/lcdui/package-summary.html>

Interface CommandListener. [referred 20.04.2011]. Available in www-document:
<URL
http://download.oracle.com/javame/config/cldc/ref-impl/midp2.0/jsr118/javax/microedition/lcdui/CommandListener.html>

Game API Package. [referred 20.04.2011]. Available in www-document:
<URL
http://download.oracle.com/javame/config/cldc/ref-impl/midp2.0/jsr118/javax/microedition/lcdui/game/package-summary.html>

Class GameCanvas. [referred 20.04.2011]. Available in www-document: <URL http://download.oracle.com/javame/config/cldc/ref-impl/midp2.0/jsr118/javax/microedition/lcdui/game/GameCanvas.html>

Class Graphics. [referred 20.04.2011]. Available in www-document: <URL http://download.oracle.com/javame/config/cldc/ref-impl/midp2.0/jsr118/javax/microedition/lcdui/Graphics.html>

Java ME Downloads. [referred 25.04.2011]. Available in www-document: <URL http://www.oracle.com/technetwork/java/javame/downloads/sdk30-jsp-139759.html>

APPENDICES

Appendix 1

```
/*Class Game*/

import java x.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Game extends MIDlet {
   GameCanvas game; // Define the object from class GameCanvas
   private boolean midletPaused = false;

public Game() {
   super();
   game=new GameCanvas(this); // Create the object from classGameCanvas
   }

private void initialize() {
   }

public void startMIDlet() {
   }

public void resumeMIDlet() {
   }

public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
   Display display = getDisplay();
   if (alert == null) {
      display.setCurrent(nextDisplayable);
      } else {
      display.setCurrent(alert, nextDisplayable);
      }
   }
public Display getDisplay () {
    return Display.getDisplay(this);
    }

public void exitMIDlet() {
    switchDisplayable (null, null);
    destroyApp(true);
    notifyDestroyed();
    }

protected void startApp() {
    Display.getDisplay(this).setCurrent(game); // Draw the user interface on the screen
    if (midletPaused) {
        resumeMIDlet ();
        } else {
        initialize ();
        startMIDlet ();
        }
        midletPaused = false;
    }

protected void pauseApp() {
    midletPaused = true;
    }

protected void destroyApp(boolean unconditional) {
    }

}
```

Appendix 2

```
/*Class GameCanvas*/

import javax.microedition.lcdui.*;

public class GameCanvas extends Canvas implements CommandListener {
   protected Game game;
   protected int empty; // Space on the right of screen
   protected int x; // Coordinate output on the Board
   protected int cellWidth; // Side length on each grid
   protected int mapWidth,canvasW; // Board’s width and Canvas’s width
   protected int a,b,c,d; // The broken lines under Cannons
   protected int chessR; // The radius of Pieces
   protected int selectedX,selectedY; // Choice box location on the Board
   protected static int i,j;
   protected int m,n,p;
   // Remember selectedX, selectedY, and point[selectedX][selectedY] at the beginning
   protected String q; // Remember word[selectedX][selectedY]
   protected int guard,guard1,guard2,g,g1;
   // Tag the times when FIRE presses, g is used to judge whether there are some pieces
   between former and latter pieces on this line
   protected static int g2,isRedWin,isWhiteWin; // g2 represents who will go ahead
   protected Command exitCmd;
   protected int point[][]={{1,2,3,4,5,6,7,8,9}, // Initiate an integer array
                                        {0,0,0,0,0,0,0,0,0},
                                        {0,10,0,0,0,0,0,11,0},
                                        {12,0,13,0,14,0,15,0,16},
                                        {0,0,0,0,0,0,0,0,0},
                                        {0,0,0,0,0,0,0,0,0},
                                        {28,0,29,0,30,0,31,0,32},
                                        {0,26,0,0,0,0,0,27,0},
                                        {0,0,0,0,0,0,0,0,0},
                                        {17,18,19,20,21,22,23,24,25}};
   protected String[][] word;
```

```
public GameCanvas() {
   try {
      setCommandListener(this);
      addCommand(new Command("Exit", Command.EXIT, 1));
      } catch(Exception e) {
      e.printStackTrace();
      }
   };

public GameCanvas(Game game) // Constructor
{
   this.game=game;
   empty=getWidth()/6;
   x=empty*1/3;
   canvasW=getWidth()-empty;
   mapWidth=canvasW-canvasW%8;
   cellWidth=mapWidth/8;
   a=cellWidth*2/5;
   b=cellWidth/8;
   c=cellWidth-a;
   d=cellWidth-b;
   chessR=cellWidth*2/5;
   selectedX=0;
   selectedY=0;
   guard=0;
   guard1=selectedX;guard2=selectedY;
   m=guard1;n=guard2;
   word=new String[10][9];
   g2=1;
   for(i=0;i<10;i++) // Initiate all the characters
   {
      for(j=0;j<9;j++)
      {
         if(i==0)
```

```
{
   if(j==0){word[i][j]="车";}
   if(j==1){word[i][j]="马";}
   if(j==2){word[i][j]="相";}
   if(j==3){word[i][j]="士";}
   if(j==4){word[i][j]="帅";}
   if(j==8){word[i][j]="车";}
   if(j==7){word[i][j]="马";}
   if(j==6){word[i][j]="相";}
   if(j==5){word[i][j]="士";}
}
if(i==1){word[i][j]="空";}
if(i==2)
{
   if((j!=1)&(j!=7)){word[i][j]="空";}
   if(j==1){word[i][j]="炮";}
   if(j==7){word[i][j]="炮";}
}
if(i==3)
{
   if(j%2==0){word[i][j]="卒";}
   if(j%2==1){word[i][j]="空";}
}
if(i==4){word[i][j]="空";}
if(i==5){word[i][j]="空";}
if(i==6)
{
   if(j%2==0){word[i][j]="卒";}
   if(j%2==1){word[i][j]="空";}
}
if(i==7)
{
   if((j!=1)&(j!=7)){word[i][j]="空";}
```

```
                if(j==1){word[i][j]="炮";}
                if(j==7){word[i][j]="炮";}
            }
            if(i==8){word[i][j]="空";}
            if(i==9)
            {
                if(j==0){word[i][j]="车";}
                if(j==1){word[i][j]="马";}
                if(j==2){word[i][j]="相";}
                if(j==3){word[i][j]="士";}
                if(j==4){word[i][j]="帅";}
                if(j==8){word[i][j]="车";}
                if(j==7){word[i][j]="马";}
                if(j==6){word[i][j]="相";}
                if(j==5){word[i][j]="士";}
            }
        }
    }
    exitCmd = new Command("Exit", Command.EXIT, 0);
    addCommand(exitCmd);
    setCommandListener(this);
}

protected void paintMapa(Graphics g) // Draw the top half of Board
{
    for(int q=0;q<4;q++)
    {
        for(int w=0;w<8;w++)
        {
            g.setColor(128,128,128);
            g.drawRect(x+w*cellWidth,x+q*cellWidth,cellWidth,cellWidth);
        }
    }
    g.setColor(128,128,128);
    g.drawLine(x+3*cellWidth,x,x+5*cellWidth,x+2*cellWidth);
    g.drawLine(x+5*cellWidth,x,x+3*cellWidth,x+2*cellWidth);
    /*Draw the upper-left Cannon*/
    g.drawLine(x+d,x+cellWidth+c,x+d,x+cellWidth+d); // Upper-left vertical
    g.drawLine(x+c,x+cellWidth+d,x+d,x+cellWidth+d); // Upper-left horizontal
    g.drawLine(x+d+2*b,x+cellWidth+c,x+d+2*b,x+cellWidth+d); // Upper-right vertical
    g.drawLine(x+cellWidth+b,x+cellWidth+d,x+cellWidth+a,x+cellWidth+d);
    // Upper-right horizontal
    g.drawLine(x+d,x+2*cellWidth+b,x+d,x+2*cellWidth+a); // Lower-left vertical
    g.drawLine(x+c,x+cellWidth+d+2*b,x+d,x+cellWidth+d+2*b); // Lower-left horizontal
    g.drawLine(x+d+2*b,x+2*cellWidth+b,x+d+2*b,x+2*cellWidth+a);
    // Lower-right vertical
    g.drawLine(x+cellWidth+b,x+cellWidth+d+2*b,x+cellWidth+a,x+cellWidth+d+2*b);
    // Lower-right horizontal
    /*Draw the upper-right Cannon*/
    g.drawLine(x+d+6*cellWidth,x+cellWidth+c,x+d+6*cellWidth,x+cellWidth+d);
    g.drawLine(x+c+6*cellWidth,x+cellWidth+d,x+d+6*cellWidth,x+cellWidth+d);
    g.drawLine(x+d+2*b+6*cellWidth,x+cellWidth+c, x+d+2*b+6*cellWidth,
    x+cellWidth+13+9);
    g.drawLine(x+cellWidth+b+6*cellWidth,x+cellWidth+d,x+cellWidth+a+6*cellWidth,
    x+cellWidth+d);
    g.drawLine(x+d+6*cellWidth,x+2*cellWidth+b,x+d+6*cellWidth,
    x+2*cellWidth+a);
    g.drawLine(x+c+6*cellWidth,x+cellWidth+d+2*b,x+d+6*cellWidth,
    x+cellWidth+d+2*b);
    g.drawLine(x+d+2*b+6*cellWidth,x+2*cellWidth+b,x+d+2*b+6*cellWidth,
    x+2*cellWidth+a);
    g.drawLine(x+cellWidth+b+6*cellWidth,x+cellWidth+d+2*b,
    x+cellWidth+a+6*cellWidth,x+cellWidth+d+2*b);
}

protected void paintMapb(Graphics g) // Draw the River "楚河　汉界"
{
    g.setColor(128,128,128);
    g.drawRect(x,x+4*cellWidth,mapWidth,cellWidth);
    g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD
    ,Font.SIZE_LARGE));
    g.drawString("楚河            汉界",getWidth()/2,x+4*cellWidth+
    cellWidth*3/4,Graphics.HCENTER| Graphics.BASELINE);
}

protected void paintMapc(Graphics g) // Draw the bottom half of Board
{
    for(int q=0;q<4;q++)
    {
        for(int w=0;w<8;w++)
        {
            g.setColor(128,128,128);
            g.drawRect(x+w*cellWidth,x+(q+5)*cellWidth,cellWidth,cellWidth);
        }
    }
    g.setColor(128,128,128);
    g.drawLine(x+3*cellWidth,x+7*cellWidth,x+5*cellWidth,x+9*cellWidth);
    g.drawLine(x+5*cellWidth,x+7*cellWidth,x+3*cellWidth,x+9*cellWidth);
    /* Draw the lower-left Cannon */
    g.drawLine(x+d,x+6*cellWidth+c,x+d,x+6*cellWidth+d); // Upper-left vertical
    g.drawLine(x+c,x+6*cellWidth+d,x+d,x+6*cellWidth+d); // Upper-left horizontal
    g.drawLine(x+d+2*b,x+6*cellWidth+c,x+d+2*b,x+6*cellWidth+d);
    // Upper-right vertical
    g.drawLine(x+cellWidth+b,x+6*cellWidth+d,x+cellWidth+a,x+6*cellWidth+d);
    // Upper-right horizontal
    g.drawLine(x+d,x+7*cellWidth+b,x+d,x+7*cellWidth+a); // Lower-left vertical
    g.drawLine(x+c,x+6*cellWidth+d+2*b,x+d,x+6*cellWidth+d+2*b);
    // Lower-left horizontal
    g.drawLine(x+d+2*b,x+7*cellWidth+b,x+d+2*b,x+7*cellWidth+a);
    // Lower-right vertical
    g.drawLine(x+cellWidth+b,x+6*cellWidth+d+2*b,x+cellWidth+a,
    x+6*cellWidth+d+2*b); // Lower-right horizontal
    /*Draw the lower-right Cannon*/
    g.drawLine(x+d+6*cellWidth,x+6*cellWidth+c,x+d+6*cellWidth,
    x+6*cellWidth+d);
    g.drawLine(x+c+6*cellWidth,x+6*cellWidth+d,x+d+6*cellWidth,
    x+6*cellWidth+d);
    g.drawLine(x+d+2*b+6*cellWidth,x+6*cellWidth+c,x+d+2*b+6*cellWidth,
    x+6*cellWidth+d);
    g.drawLine(x+cellWidth+b+6*cellWidth,x+6*cellWidth+d,
    x+cellWidth+a+6*cellWidth,x+6*cellWidth+d);
    g.drawLine(x+d+6*cellWidth,x+7*cellWidth+b,x+d+6*cellWidth,
    x+7*cellWidth+a);
    g.drawLine(x+c+6*cellWidth,x+6*cellWidth+d+2*b,x+d+6*cellWidth,
    x+6*cellWidth+d+2*b);
    g.drawLine(x+d+2*b+6*cellWidth,x+7*cellWidth+b,x+d+2*b+6*cellWidth,
    x+7*cellWidth+a);
    g.drawLine(x+cellWidth+b+6*cellWidth,x+6*cellWidth+d+2*b,
    x+cellWidth+a+6*cellWidth,x+6*cellWidth+d+2*b);
}

protected void paintAllChess(Graphics g) // Draw all the pieces
{
    for(i=0;i<10;i++)
    {
        for(j=0;j<9;j++)
        {
            if(point[i][j]!=0)
            {if(point[i][j]<17){g.setColor(255,0,0);}
              else{g.setColor(255,255,255);}
              g.fillArc(x-chessR+j*cellWidth,x-chessR+i*cellWidth,2*chessR,2*chessR,0,360);
              g.setColor(0x00000000);
              g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD,
              Font.SIZE_LARGE));
              g.drawString(word[i][j],x+j*cellWidth,x+chessR+i*cellWidth,Graphics.
              HCENTER|Graphics.BOTTOM);
            }
        }
    }
```

```
}

protected void chooseChess(Graphics g) // Choose one piece and go ahead
{
   m=guard1;n=guard2;
   if(point[guard2][guard1]!=0)
   {
      if(g2%2==1)
      {
         if(point[guard2][guard1]<=16)
         {
            g.setColor(255,255,0);
            g.fillArc(x-chessR+guard1*cellWidth,x-chessR+guard2*cellWidth,
            2*chessR,2*chessR,0,360);
            g.setColor(0x00000000);
            g.setFont(Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
            Font.SIZE_LARGE));
            g.drawString(word[guard2][guard1],x+guard1*cellWidth,
            x+chessR+guard2*cellWidth,Graphics.HCENTER|Graphics.BOTTOM);
         }
      }
      if(g2%2==0)
      {
         if(point[guard2][guard1]>16)
         {
            g.setColor(0,255,0);
            g.fillArc(x-chessR+guard1*cellWidth,x-chessR+guard2*cellWidth,
            2*chessR,2*chessR,0,360);
            g.setColor(0x00000000);
            g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD,
            Font.SIZE_LARGE));
            g.drawString(word[guard2][guard1],x+guard1*cellWidth,
            x+chessR+guard2*cellWidth,Graphics.HCENTER|Graphics.BOTTOM);
         }
      }
```

```
    }
}

protected void whoIsGoing(Graphics g) // Judge who will go ahead
{
    checkWin();
    g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD,
    Font.SIZE_LARGE));
    if(isRedWin!=0)
    {
        if(g2%2==1)
        {
        g.setColor(255,0,0);
        g.drawString("Red-side goes",x,x+chessR+10*cellWidth,Graphics.
        LEFT|Graphics.BOTTOM);
        }
    }
    else
    {
    g.setColor(255,255,255);
    g.drawString("White-side wins",x,x+chessR+10*cellWidth,Graphics.
    LEFT|Graphics.BOTTOM);
    }
    if(isWhiteWin!=0)
    {
        if(g2%2==0)
        {
        g.setColor(255,255,255);
        g.drawString("Whit-side goes",x,x+chessR+10*cellWidth,Graphics.
        LEFT|Graphics.BOTTOM);
        }
    }
    else
    {g.setColor(255,0,0);
    g.drawString("Red-side wins",x,x+chessR+10*cellWidth,Graphics.
    LEFT|Graphics.BOTTOM);
    }
}

protected void checkWin() // Judge winning or losing
{
    isRedWin=0;isWhiteWin=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(point[0+i][3+j]==5){isRedWin++;}
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(point[7+i][3+j]==21){isWhiteWin++;}
        }
    }
}

protected void paintSelected(Graphics g) // Draw the choice box
{
    g.setColor(0,0,255);
    g.drawRect(x-chessR+selectedX*cellWidth,x-chessR+selectedY*cellWidth,
    2*chessR,2*chessR);
}

/*Paint*/
protected void paint(Graphics g)
{
    g.setColor(0x00000000);
    g.fillRect(0, 0, getWidth(), getHeight());
    paintMapa(g);
    paintMapb(g);
    paintMapc(g);
    paintAllChess(g);
    if(guard%2==1)
    {
        chooseChess(g);
    }
    paintSelected(g);
    whoIsGoing(g);
}

protected void changTwoChessNum(int m,int n,int selectedX,int selectedY)
// Change the value of two pieces
{
    g2++;
    p=point[selectedY][selectedX];
    point[selectedY][selectedX]=point[n][m];
    point[n][m]=0;
    q=word[selectedY][selectedX];
    word[selectedY][selectedX]=word[n][m];
    word[n][m]="空";
}
protected void theRuleOfChe(int m,int n,int selectedX,int selectedY) // The rule of Chariots
{
    g=0;
    if(m==selectedX)
    {
        if(n>selectedY)
        {
            for(i=1;i<10;i++)
            {
                if(point[selectedY+i][m]!=0){g++;}
            }
        }
    else
    {
      for(i=1;i<10;i++)
      {
        if(point[n+i][m]!=0){g++;}
      }
    }
    if(g==0)
    {
      changTwoChessNum(m,n,selectedX,selectedY);
    }
 }
 if(n==selectedY)
   {
     if(m>selectedX)
     {
       for(i=1;i<10;i++)
       {
         if(point[n][i+selectedX]!=0){g++;}
       }
    }
    else
    {
      for(i=1;i<10;i++)
      {
        if(point[n][m+i]!=0){g++;}
      }
    }
    if(g==0)
    {
      changTwoChessNum(m,n,selectedX,selectedY);
    }
  }
}
```

```
protected void theRuleOfMa(int m,int n,int selectedX,int selectedY) // The rule of Horses
{
    if(n<9)
    {
        if(point[n+1][m]==0)
        {
            if(selectedX-m==1){if(selectedY-n==2) {
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(n>0)
    {
        if(point[n-1][m]==0)
        {
            if(m-selectedX==1){if(n-selectedY==2){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(n<9)
    {
        if(point[n+1][m]==0)
        {
            if(selectedX-m==-1){if(selectedY-n==2){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(n>0)
    {
        if(point[n-1][m]==0)
        {
            if(m-selectedX==-1){if(n-selectedY==2){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(m<8)
    {
        if(point[n][m+1]==0)
        {
            if(selectedX-m==2){if(selectedY-n==1){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(m>0)
    {
        if(point[n][m-1]==0)
        {
            if(m-selectedX==2){if(n-selectedY==1){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(m<8)
    {
        if(point[n][m+1]==0)
        {
            if(selectedX-m==2){if(selectedY-n==-1){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
    if(m>0)
    {
        if(point[n][m-1]==0)
        {
            if(m-selectedX==2){if(n-selectedY==-1){
            changTwoChessNum(m,n,selectedX,selectedY);}}
        }
    }
}

protected void theRuleOfPao(int m,int n,int selectedX,int selectedY,int g1)
// The rule of Cannons
```

```
{
   g=0;
   if(m==selectedX)
   {
      if(n>selectedY)
      {
         for(i=1;i<10;i++)
         {
            if(point[selectedY+i][m]!=0){g++;}
         }
      }
      else
      {
         for(i=1;i<10;i++)
         {
            if(point[n+i][m]!=0){g++;}
         }
      }
      if(g==g1){changTwoChessNum(m,n,selectedX,selectedY);}
   }
   if(n==selectedY)
   {
      if(m>selectedX)
      {
         for(i=1;i<10;i++)
         {
            if(point[n][i+selectedX]!=0){g++;}
         }
      }
      else
      {
         for(i=1;i<10;i++)
         {
            if(point[n][m+i]!=0){g++;}
         }
```

```
        }
        if(g==g1){changTwoChessNum(m,n,selectedX,selectedY);}
    }
}

protected void theRuleOfXiang(int m,int n,int selectedX,int selectedY)
// The rule of Elephants
{
    if(n<9&m<8)
    {
        if(point[n+1][m+1]==0)
        {
            if((selectedX-m==2)&(selectedY-n==2))
            {
                changTwoChessNum(m,n,selectedX,selectedY);
            }
        }
    }

    if(n>0&m<8)
    {
        if(point[n-1][m+1]==0)
        {
            if((selectedX-m==2)&(selectedY-n==-2))
            {
                changTwoChessNum(m,n,selectedX,selectedY);
            }
        }
    }

    if(n<9&m>0)
    {
        if(point[n+1][m-1]==0)
        {
            if((selectedX-m==-2)&(selectedY-n==2))
            {
                changTwoChessNum(m,n,selectedX,selectedY);
            }
        }
    }

    if(n>0&m>0)
    {
        if(point[n-1][m-1]==0)
        {
            if((selectedX-m==-2)&(selectedY-n==-2))
            {
                changTwoChessNum(m,n,selectedX,selectedY);
            }
        }
    }
}

protected void theRuleOfShi(int m,int n,int selectedX,int selectedY) // The rule of Advisor
{
    if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=7&n<=9)
    &(selectedY>=7&selectedY<=9))
    {
        if((selectedX-m==1)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==1)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);}
        }
    if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=0&n<3)
    &(selectedY>=0&selectedY<3))
    {
        if((selectedX-m==1)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==1)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
    }
}

protected void theRuleOfShuai(int m,int n,int selectedX,int selectedY)
// The rule of Generals
{
    if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=7&n<=9)
    &(selectedY>=7&selectedY<=9))
    {
        if((selectedX-m==1)&(selectedY-n==0))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==0))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==0)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==0)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
    }
    if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=0&n<3)
    &(selectedY>=0&selectedY<3))
    {
        if((selectedX-m==1)&(selectedY-n==0))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==-1)&(selectedY-n==0))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==0)&(selectedY-n==1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
        if((selectedX-m==0)&(selectedY-n==-1))
        {
            changTwoChessNum(m,n,selectedX,selectedY);
        }
    }
}
```

```
protected void theRuleOfZu(int m,int n,int selectedX,int selectedY) // The rule of Soldiers
{
   if(point[n][m]<17)
   {
      if(selectedY>=n)
      {
         if(n<5)
         {
            if((selectedY-n==1)&(selectedX-m==0))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
         }
         else
         {
            if((selectedY-n==1)&(selectedX-m==0))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
            if((selectedY-n==0)&(selectedX-m==1))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
            if((selectedY-n==0)&(selectedX-m==-1))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
         }
      }
   }
   else
   {
      if(selectedY<=n)
      {
         if(n>4)
         {
            if((selectedY-n==-1)&(selectedX-m==0))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
           }
         }
         else
         {
            if((selectedY-n==-1)&(selectedX-m==0))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
            if((selectedY-n==0)&(selectedX-m==1))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
            if((selectedY-n==0)&(selectedX-m==-1))
            {
               changTwoChessNum(m,n,selectedX,selectedY);
            }
         }
      }
   }
}

/*Called when a key is pressed*/
protected void keyPressed(int keyCode) // Handle the buttons
{
   int action = getGameAction(keyCode);
   if (action == Canvas.LEFT )
   {
      selectedX=(--selectedX+8+1)%(8+1);
   }
   else if (action == Canvas.RIGHT)
```

```
{
   selectedX=(++selectedX)%(8+1);
}
else if (action == Canvas.UP)
{
   selectedY=(--selectedY+9+1)%(9+1);
}
else if (action == Canvas.DOWN)
{
   selectedY=(++selectedY)%(9+1);
}
else if (action == Canvas.FIRE)
{
   guard=guard+1;
   if(guard%2==1)
   {
      if(point[selectedY][selectedX]!=0)
      {
         guard1=selectedX;
         guard2=selectedY;
      }
   }
   if(guard%2==0)
   {
      if(point[selectedY][selectedX]!=point[n][m])
      {
         if((point[n][m]==1)|(point[n][m]==9)|(point[n][m]==17)|(point[n][m]==25))
         {
            if(point[selectedY][selectedX]==0)
            {
               theRuleOfChe(m,n,selectedX,selectedY);
            }
            else
            {
               if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
```

```
              {
                theRuleOfChe(m,n,selectedX,selectedY);
              }
            }
          }
          if((point[n][m]==2)|(point[n][m]==8)|(point[n][m]==18)|(point[n][m]==24))
          {
            if(point[selectedY][selectedX]==0)
            {
              theRuleOfMa(m,n,selectedX,selectedY);
            }
            else
            {
              if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
              {
                theRuleOfMa(m,n,selectedX,selectedY);
              }
            }
          }

if((point[n][m]==10)|(point[n][m]==11)|(point[n][m]==26)|(point[n][m]==27))
          {
            if(point[selectedY][selectedX]==0)
            {
              g1=0;
              theRuleOfPao(m,n,selectedX,selectedY,g1);
            }
            else
            {
            g1=1;
            if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
            {
              theRuleOfPao(m,n,selectedX,selectedY,g1);
            }
          }
```

```
}
if((point[n][m]==3)|(point[n][m]==7)|(point[n][m]==19)|(point[n][m]==23))
{
   if(point[selectedY][selectedX]==0)
   {
      theRuleOfXiang(m,n,selectedX,selectedY);
   }
   else
   {
      if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
      {
         theRuleOfXiang(m,n,selectedX,selectedY);
      }
   }
}
if((point[n][m]==4)|(point[n][m]==6)|(point[n][m]==20)|(point[n][m]==22))
{
   if(point[selectedY][selectedX]==0)
   {
      theRuleOfShi(m,n,selectedX,selectedY);
   }
   else
   {
      if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
      {
         theRuleOfShi(m,n,selectedX,selectedY);
      }
   }
}
if((point[n][m]==5)|(point[n][m]==21))
{
   if(point[selectedY][selectedX]==0)
   {
      theRuleOfShuai(m,n,selectedX,selectedY);
   }
```

```
    else
    {
        if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
        {
            theRuleOfShuai(m,n,selectedX,selectedY);
        }
    }
}
if((point[n][m]>11&point[n][m]<17))
{
    if(point[selectedY][selectedX]==0)
    {
        theRuleOfZu(m,n,selectedX,selectedY);
    }
    else
    {
        if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
        {
            theRuleOfZu(m,n,selectedX,selectedY);
        }
    }
}
if(point[n][m]>27)
{
    if(point[selectedY][selectedX]==0)
    {
        theRuleOfZu(m,n,selectedX,selectedY);
    }
    else
    {
        if((point[selectedY][selectedX]/17)!=(point[n][m]/17))
        {
            theRuleOfZu(m,n,selectedX,selectedY);
        }
    }
```

```
                }
            }
        }
    }
    repaint();
}

/*Called when a key is released*/
protected void keyReleased(int keyCode) {
}

/*Called when a key is repeated (held down)*/
protected void keyRepeated(int keyCode) {
}

/*Called when the pointer is dragged*/
protected void pointerDragged(int x, int y) {
}

/*Called when the pointer is pressed*/
protected void pointerPressed(int x, int y) {
}

/*Called when the pointer is released*/
protected void pointerReleased(int x, int y) {
}

/*Called when action should be handled*/
public void commandAction(Command c, Displayable d) {
    if (c == exitCmd) {
        game.destroyApp(false);
        game.notifyDestroyed();
        }
    }
}
```