Department of Electronic Engineering

FINAL YEAR PROJECT REPORT

BEngIE-2008/09-LMP-10

G6 Chinese Input Anywhere in MS-Windows Mobile

Student Name: Yau Pak Yu
Student ID: 
Supervisor: Dr. PO, L M
Assessor: Dr PAO, Derek C W

Bachelor of Engineering (Honours) in Information Engineering
Student Final Year Project Declaration

I have read the student handbook and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I declare that the work submitted for the final year project does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Student Handbook.

Project Title : G6 Chinese Input Anywhere in MS-Windows Mobile

Student Name : Yau Pak Yu                      Student ID:

Signature: ________________________________       Date: 24th April, 2009
# Table of contents

List of Figures........................................................................................................ vi
Abstract ............................................................................................................... vii
1. Objectives ........................................................................................................ 1
   1.1 Implementation of basic functions.............................................................. 1
   1.2 Addition of Special Features .................................................................... 2
   1.3 Creation of a Installation Package............................................................ 3
2. Introduction .................................................................................................... 4
   2.1 What is G6? ............................................................................................ 4
   2.2 Specification ........................................................................................... 5
   2.3 Background Information ..................................................................... 6
      2.3.1 Keystroke Assignment................................................................. 6
      2.3.2 Inputting Rule .............................................................................. 7
      2.3.3 Information Given.......................................................................... 8
      2.3.4 Development Tools ....................................................................... 8
      2.3.5 Graphical User Interface Design............................................... 10
         2.3.5.1 1st G6 GUI Design .............................................................. 10
         2.3.5.2 2nd G6 GUI Design .............................................................. 12
         2.3.5.3 Final Version of G6 GUI .................................................... 13
3. Methodology .................................................................................................. 15
   3.1 Input Method Development on Windows Mobile............................... 15
   3.2 Starting with a Hello World for Implementation.................................. 17
   3.3 Deploying the Program with an Emulator .......................................... 18
   3.4 Storage of Character Information......................................................... 19
   3.5 Searching Algorithm on Outputting Characters ............................... 21
   3.6 Listing of Associate Candidates........................................................ 22
   3.7 Dictionary Function ............................................................................ 23
   3.8 Installation Package Building............................................................... 24
4. Product Usage .............................................................................................. 26
   4.1 System Requirements .......................................................................... 26
   4.2 Installation and Deployment .................................................................. 27
   4.3 Fundamental Functions ...................................................................... 29
List of Figures

Figure 2.1 1st G6 GUI Design
Figure 2.2 2nd G6 GUI Design
Figure 2.3 Final Version of G6 GUI
Figure 3.1 SIP Application Development
Figure 3.2 Windows Mobile 6 Professional Emulator
Figure 3.3 basic illustration of the searching algorithm
Figure 3.4 WinCE CAB Manager
Figure 4.1 Before Installation of G6
Figure 4.2 G6 is successfully installed
Figure 4.3 G6 is allowed to be chosen
Figure 4.4 While G6 is Selected
Figure 4.5 Basic Usage in Single-character Mode
Figure 4.6 Basic Usage in Two-character Mode
Figure 4.7 Basic Usage in Three-character Mode
Figure 4.8 Basic Usage in Multi-character Mode
Figure 4.9 Simplified Chinese Mode
Figure 4.10 Flexibility of Input Sequence
Figure 4.11 One-click Characters in G6 Mode
Figure 4.12 One-click Characters in Roman Unshifted Mode
Figure 4.13 One-click Characters in Roman Shifted Mode
Figure 4.14 One-click Characters in Numeric Mode
Figure 4.15 One-click Characters in Symbolic Mode
Figure 4.16 Help Button in G6
Figure 4.17 Help Menu Page Contents
Figure 4.18 Wildcard Key Usage
Figure 4.19 Associate Candidate List and Consecutive Associate candidate List
Figure 4.20 Key Code Dictionary Usage
Figure 4.21 Input Method Options
Figure 4.22 Theme Changing
Abstract

Nowadays, communication is very important among people. SMS and email are the common communication tools through electronic devices. The PocketPC with MS-Windows Mobile can also send out and receive messages via the network. However, what we concern is whether the user is familiar with the input method, especially Chinese input method.

With this kind of concern to users, we developed G6 (diGit 6) Bihua Chinese Input Method (六碼筆劃輸入法). To enhance the mobility, we are going to put G6 onto the MS-Windows Mobile platform. It may be the easiest Chinese input method all over the world: easy to learn, easy to use, and easy to remember.

With considering the characteristics of the Chinese characters, we have found that the Chinese characters can be divided into 5 basic stroke types. To make use of this uniqueness other than the other language, Dr. Lai-Man PO invented (3+3) rule by using the first 3 strokes and the last 3 strokes for single character, and the first 3 strokes of the first and last characters for a phrase. As a result, each character or phrase should be encoded by at most six digits, which could significantly improve the input speed.
1. Objectives

1.1 Implementation of basic functions

As G6 has never been developed onto the WM platform, the first objective is to install the input method into the registry of the Windows. With adding the registry of the input method, users can select G6 to type Chinese characters wherever there is a text input. The second task is to define the graphical user interface of G6. Since the size of the screen of a PocketPC is limited, a clear GUI and a page of instruction should be included in the product. The size of the key strokes should be big enough to be seen and the instruction is act as a tutorial page to teach users how to use G6. After the user interface defined, key mapping and area mapping are the next step. As the G6 Bihua Chinese Input Method in PocketPC should be controlled by touch screen, there is a must to define which area of the screen is mapped to which input key. Basically there are at least 5 keys displayed to represent the 5 codes of keystroke type. Furthermore, some function keys should also appear in the GUI for user to select. The function keys include the wildcard key, symbol key, page up/down key, multi-character key, dictionary key, help key, etc. After that, the phase of implementation of program code begins. With selecting from the character database, after the character codes are entered, the results should be stored into a buffer and generated out on the screen. Users can choose the output character or phrase by selecting their choice from the result listbox. By the way, the sequence of the characters / phrases displayed should be ordered by the frequency of the usage.
1.2 Addition of Special Features

Other than the basic functions, some other special functions will also be implemented. In the text inputting mode, there will be a conversion button which can convert the G6 input method from Tradition Chinese character inputting to Simplified Chinese character inputting, and vice versa. This is convenient for users to switch the G6 input method by simply clicking on a button. With using the character database, the user can select the associate character(s) from his selected character. This feature is important for the end users as they can save time by choosing the associated characters listed without typing the complete codes of character or phrase. Besides, some modes will be defined for users to select, for example, by entering 2-character mode, the resulting records should be shown in 2 characters. This saves time to filter out the unwanted results and display the relevant records. To make some guidelines for users to use G6, a help page should be added. It gives a brief description on what is G6 and teaches user how to control. This helps users learning usage of G6 within a short period time. Finally, a dictionary function will be implemented for users to check the G6 code of any Chinese character / phrase. This is an additional feature for users who do not know how to decode a particular Chinese character in G6 code.
1.3 Creation of a Installation Package

To deploy G6 in Windows Mobile Professional Emulator, developer can simply deploy the project in the Visual Studio environment. But if the end users would like to install G6 in their PocketPC, a installation package should be created for them to install.

The objective of this part is to create G6 project as an installation package for user to put into the PocketPC folder. While user clicked on the installer, G6 will be auto-installed to the target device. This must be a simple process to reduce the complexity for user to install G6 Bihua Chinese Input Method on Windows Mobile.
2. Introduction

2.1 What is G6?

G6 in full name is “diGit 6 Bihua Chinese Input Method”, which is a new Chinese character and phrase input method invented by Dr. Lai-Man PO.

This Chinese input method has a unique code pattern “3+3” rule for every single character and the phrase. For the single character, we will use the first 3 and the last 3 strokes of the character. And for the phrase which contains multiple characters, we will use the first 3 strokes of the first and the last character. As a result, any character and phrase should be encoded with at most 6 digits. And this is the reason we call this “G6”.

As Pocket PC with MS-Windows Mobile operating system can also send out and receive messages via network, Chinese Input Method become more important. But we concern how to improve the efficiency in typing Chinese. Therefore we developed G6 in Windows Mobile version.
2.2 Specification

In this project, Microsoft Windows Mobile (WM) version 6 operating system is selected for deployment.

As my project title is given as G6 in Windows Mobile version, I had done a research on the WM platform. Although it does not dominate the market share, it still acts an important role on mobile operating system. (13.6% Market Share Sales Q3 2008) [1]. And there is a trend that more users try to use WM year by year since Microsoft keeps development new version on WM.

When I started to develop on the project, the latest version is Windows Mobile 6.1, which is a minor upgrade to the existing Windows Mobile 6 platform. [2] Considering the future usage of the OS, I decided to build G6 on Windows Mobile 6 platform. Besides, there were some significant change between WM5 and WM6. I therefore focus on the development on WM6.

There are 3 versions of WM6, which includes:

1. "Windows Mobile 6 Standard" for Smartphones (phones without touchscreens),
2. "Windows Mobile 6 Professional" for Pocket PCs with phone functionality

To select on a particular version, my decision based on the functionality. Since Window Mobile 6 Professional has the best functionality and will be a trend to become popular, it was selected to be my developing operating system.
2.3 Background Information

G6 Bihua Chinese Input Method was invented by my FYP supervisor. It has been developed about a year for researching on the method structure, character database, Windows XP implementation, etc. The main task in this project is to deploy the G6 Bihua Chinese Input Method on the PocketPC which uses WM6 Pro OS.

2.3.1 Keystroke Assignment

For the keystroke assignment, we had made lots of research and adjustment, and finally we decided to use 5 basic stroke types, which are Horizontal stroke [一], Vertical stroke [丨], Left-Falling stroke [丿], Dot stroke [丶], and Turning stroke [フ].

- Horizontal stroke [一]: 一
- Vertical stroke [丨]: 丨
- Left-Falling stroke [丿]: 丿
- Dot stroke [丶]: 丶
- Turning stroke [フ]: フ

The following examples show how the above keystrokes work.

1. Horizontal strokes [一]: 想 像 背 四 涼 森
2. Vertical stroke [ 丨 ] :

想肖胚凶楞松

3. Left-Falling stroke [ 丨 ] :

想肖胚凶楞松

4. Dot stroke [ 、 ] :

想肖胚凶楞松

5. Turning stroke [ フ ] :

想肖胚凶楞松

2.3.2 Inputting Rule

This Chinese input method has a unique code pattern “3+3” rule for every single character and the phrase. For the single character, we will use the first 3 and the last 3 strokes of the character. And for the phrase which contains multiple characters, we will use the first 3 strokes of the first and the last character. The following figures show how G6 works:

Example 1: (Normal character example)

流 ⟷ 流バーノフ

Example 2: (for character less than 6 strokes in total)

水 ⟷ 水フノ

Example 3: (Normal phrase example)

Example 4: (Extreme phrase example)

In example 4, it is one of the special characteristics of G6. In the traditional Chinese Input Method, more characters must contain more keystrokes to be input. However, in the above example, more strokes but uses less key codes, which violates the normal concept to enhance higher typing speed and efficiency.

2.3.3 Information Given

As G6 was developed for almost a year since the first concept appeared, some of the information were provided for developing my project on Windows Mobile.

Apart from the keystroke assignment and the inputting rule, the character and phrase information were given before starting on my project. The character and phrase information includes the frequency of usage, the full code of G6 keystroke, and the size of the database.

2.3.4 Development Tools

To develop G6 in Windows Mobile platform, 5 major softwares were used, which are:
1. Microsoft ActiveSync
2. .NET Compact Framework 2.0
4. Windows Mobile 6 Professional SDK
5. WinCE CAB Manager

Microsoft Visual Studio is the basic development platform for the input method implementation, where Windows Mobile 6 Professional SDK is the software development kit for developing G6.

The basic requirement to install Windows Mobile 6 Professional SDK is to install Microsoft ActiveSync, .NET Compact Framework 2.0 and Microsoft Visual Studio 2005 (or higher) before hand. Thus there should be a installation sequence to install these softwares.

ActiveSync is a data synchronization program developed by Microsoft for use with its Microsoft Windows line of operating systems. It can make the PocketPC synchronizing with the computer. Therefore during the development process, I can easily send any file from my PC to the emulated target mobile device through this synchronization software.

The WinCE CAB Manager is a utility for developers to make the released applications or DLL packed in a installation package. The extension of installation archive in Windows Mobile environment is .cab, which means the “cabinet file”. While the G6 project files were released, it can be packed in the .cab. By setting the
installation path, all the packed files can be installed to the target folder in WM and certain registry keys can be added.

2.3.5 Graphical User Interface Design

As there is no keyboard on the Windows Mobile Professional, all the inputs are controlled by touch-screen by using a stylus. As a result, graphical user interface (GUI) design is a very important part in my project. With an attractive and user-friendly design, users can use G6 in a more efficient way.

2.3.5.1 1st G6 GUI Design

The G6 GUI design phase was mainly divided into 3 stages. The following figure was my first GUI design of G6 in WM:

![1st G6 GUI Design](image)

**Figure 2.1** 1st G6 GUI Design
In figure 2.1, it is the working environment of G6 in WM. The main IM window is located at the bottom of the screen. 12 keys can be pressed including the keystroke keys and function keys. The input keystrokes are displayed beside the cursor.

Furthermore, the candidate list is shown on the left which occupied about 3/5 of the horizontal space.
2.3.5.2 2\textsuperscript{nd} G6 GUI Design

To make the GUI more attractive and functional, I had designed a modified graphical user interface in the second stage:

![Image of 2\textsuperscript{nd} G6 GUI Design]

**Figure 2.2** 2\textsuperscript{nd} G6 GUI Design

In this attempt, mode keys were added for user to select. With switching the mode, user can type both Chinese, Alphabetic, numeric or symbolic characters. Besides, the G6 keys were no longer black and white. Colors were filled on the key text to identity the key type. In this case, functions keys are in violet, keystroke keys are in green, and mode keys are in red. In addition, the input keystrokes display is placed just above the candidate list.
2.3.5.3 Final Version of G6 GUI

From the 2\textsuperscript{nd} to the 3\textsuperscript{rd} GUI design, I had made lots of minor changes and the adjustment was about 10 times. Finally the final version of the G6 GUI was confirmed:

![Final Version of G6 GUI](image)

**Figure 2.3** Final Version of G6 GUI

In the above figure, there are many differences between the 2\textsuperscript{nd} GUI and the final GUI. First, the height of the keyboard is increased from 80px to 100px, which make the buttons become larger and thus easier for user to click on a particular button. To make the graphical user interface became more attractive, I used deep background color for the buttons and light color for the button texts. Also, the buttons are filled with metallic effect. It gives a grand feeling to user. Key types are still identified by different colors as before. A wildcard key is included as a type of keystroke for user to input. More function keys are included such as dictionary
button and help button. Besides, Simplified Chinese key is added for switching between Simplified Chinese and Traditional Chinese. To make use of all the space on the screen, some frequently used punctuation marks are added in the G6 mode to fill up the screen. Font color of the candidate list is changed and the page indicator for the list is included. The major difference is the position of the candidate list. It is changed from the bottom of the screen to the middle. As a result, the G6 keyboard can take out all the space at the bottom. (240px * 100 px)
3. Methodology

3.1 Input Method Development on Windows Mobile

Unlike the traditional mobile phone, Windows Mobile Professional does not have keyboard and is controlled by touch-screen. Thus in the Windows Mobile development, there is a user interface called Software-based Input Panel (SIP). The SIP functionality in Windows Mobile allows target devices to simulate keyboard input by using an input method (IM). Therefore we can input what we wanted through the graphical user interface.

![SIP Application Development Diagram]

**Figure 3.1** SIP Application Development

The above figure shows how the input method communicated with the application. The SIP creates the IM through the IInputMethod interface. After the IM is created, the IM receives user input and passes this data to the SIP through the IIMCallback interface. After the SIP receives data from the IM, it passes the data to the Graphics, Windowing, and Events Subsystem (GWES) module. Usually, this data is a keyboard event. GWES passes the message to your application through the
standard delivery system for keyboard events. Finally the application display what is input by the user.

Through the SIP interface, an input method is created. Therefore, G6 in Windows Mobile version is not a standalone application. It is a real input method. Whenever there is a textfield for user to input, G6 can be selected for typing characters. This is the reason my project is entitled “G6 Anywhere”.
3.2 Starting with a Hello World for Implementation

To get started on implementing a new input method on the Windows Mobile platform, I used a sample template for the base of G6.

In the Microsoft Windows Mobile 6 Professional SDK, there is a samples directory which contains some fundamental program codes to teach developers how to develop applications on Windows Mobile 6. There is a sample folder “dvoraksip”, which contains a code sample named code sample is named “DvorakSIP” to demonstrate how to create a custom Soft Input Panel (SIP) as a COM object. It is intended to help users and service providers in foreign countries support different languages in the SIP.[5]

The DvorakSIP project includes some header and source files to deploy a Dvorak Input Method, which user can type English characters through this Input Method interface.

This input method does not contain any feature on functionality. The implementation of the Input Method is coded in the dvorak_implementation.cpp. I had to modify the sample code of this sample source file to implement a new input method. Therefore, G6 in Windows Mobile version is dvorak-based input method, just like building a program started with a Hello World sample.
3.3 Deploying the Program with an Emulator

For developing G6 in Windows Mobile platform, Microsoft Windows Mobile 6 Professional SDK is used rather than testing on a real PocketPC.

After installing the WM6 Pro SDK, the “Device Emulator Manager” is added to the “tools” in the MS Visual Studio 2008 working environment. Developer can select and connect a Windows Mobile 6 Professional Emulator for program testing and developing.

The following figure shows the WM6 Pro Emulator connected on a personal computer:

![Windows Mobile 6 Professional Emulator](image)

**Figure 3.2** Windows Mobile 6 Professional Emulator
3.4 Storage of Character Information

Unlike desktop or laptop, the processing power in mobile is not very strong. Considering this kind of concern, unnecessary information should be filtered out to reduce heavy loading of the mobile processor. Also, installing database tools will lower the performance in Windows Mobile. Therefore the information of characters is embedded in the .cpp source.

In my VC Project, 2 source files are used for storing the character information, which are tcdata.cpp for Traditional Chinese and scdata.cpp for Simplified Chinese. Each file stores the characters and the corresponding keystrokes in single-character mode, two-character mode, three-character mode, and multi-character mode.

As database tool is not used in my project, to reduce the complexity on sorting the output characters / phrases while user requesting the search results, the sorting process should be carried before loading the character information.

In the source data files, all the characters /phrases in the array are already sorted by the frequency of usage. The simplified the sorting process on displaying result candidate list. As the characters are sorted in order, the system will only need to select out the result candidates in a normal sequence. By the way, the frequency of usage of the characters and phrases is provided by the statistics from The Chinese University of Hong Kong (CUHK) and Peking University (PKU).

The storage of character information is in array structure, where the size of array is equal to the number of characters. In each character mode, 2 arrays are used to
store the character information. One is holding the character Unicode, and the other
one is holding the full code of keystrokes.

Moreover, all the characters are encoded in UTF16 hexcode. When the mobile
system loading the .cpp data source file, a conversion is processed from converting
the Big5 characters to Unicode. After the system read the Unicode of the characters,
it displays the Unicode Chinese character on the screen with using second
conversion. In other words, each character will be converted twice from reading in
data source to outputting. To reduce the loading time on conversion, I have put all
the characters in UTF16 instead of Big5. As a result, there is only one conversion
take place from converting the UTF16 to Unicode Chinese character.
3.5 Searching Algorithm on Outputting Characters

In the implementation of outputting result candidates, I have created 2 arrays for storing the character information in a mode. One is holding the character Unicode, and the other one is holding the full code of keystrokes. The following figure illustrates how these 2 arrays communicate:

![Figure 3.3 basic illustration of the searching algorithm](image)

The basic concept is to search on character / phrase by inputting the code of keystrokes, for example, a user input the keystrokes in G6. After he input the code, the system will immediately search on the keystroke array to check which indexes on the array are matched with the input keystroke. In figure 3.3, index #1 is found to be matched. The index is stored and reused on the character array for outputting the corresponding character. In this case, index #1 is then passed to the character array, and output the character with index #1. This is the basic searching algorithm in my project.
3.6 Listing of Associate Candidates

To display an associate candidate list, no extra data source is required. Since there are data sources to store the phrase information in two-character mode, three-character mode and multi-character mode, these source files can be reused on selecting the associate characters.

When the user has selected a single character, a buffer stored that particular character and the system will immediately search on the phrase array. To be specific, the system will only check on the first character in the phrase array (that is Index #0 of the phrase). If the first character in phrase is matched with the selected character, the phrase except the first character will be selected out as the associate candidate.

In addition, associate candidate list will appear only in the single character mode. It is not logical to display the associate characters / phrases when user selected more than one single character. Therefore, in my G6 implementation, consecutive associate candidate list will be displayed if user keeps selecting single characters.
3.7 Dictionary Function

To write a dictionary function for user to find the full code of keystrokes by giving the character or phrase, the system will only need to search on the character array.

When user input the character or phrase, the system will get the text content stored into the buffer. It will then count on the buffer to check how many characters are input by user. Having the number of characters, the system will then look up on the corresponding character / phrase array. If the character or whole phrase in the array is exact matched with the user input, the index of the array will be stored and used on the keystroke array, for example, index #8 of character array is found to be matched with the input, the key code in keystroke array with index #8 will be selected out and display on the screen. Hence, user can find out the full key code of any character / phrase.
3.8 Installation Package Building

In the release of the G6 project, the only file generated is the g6sip.dll, which is added to the Windows folder of Mobile Device to tell the system a new input method is installed. To make this file being auto-installed to a specific path and adding the registry keys, a cabinet file creation tool is used which is called “WinCE CAB Manager”.

![WinCE CAB Manager](image)

**Figure 3.4** WinCE CAB Manager

In figure 3.4, it is the working environment of WinCE CAB Manager. With using this software, developers can easily assign the properties of the cabinet file.

The first step to create a cabinet file is to type in the information of the .cab. The important information includes the application name, minimum OS version of Windows Mobile, unsupported platforms, etc.

Next step is to add all the installation files into the .cab. In my release of the project, only g6sip.dll is needed to be installed. Thus I only have to add this file to the
cabinet and set its installation path to be “%Windows%”, which is the Windows folder in Windows Mobile.

Finally step is setting up the registry keys’ value. To find out which registry keys should be added to the system, I have to look up the source file which contains the DllRegisterServer(void) implementation. (In my project, g6sip.cpp gives the registry information) This is an important step for the installation process. If the registry keys are not set or set incorrectly, user cannot fully uninstall G6 if they wish to remove this input method.

By clicking on the save button after the above steps are done, the .cab installation package is created and the end user can click on it to install G6 Bihua Chinese Input Method in the Windows Mobile environment.
4. Product Usage

4.1 System Requirements

To ensure “G6 Bihua Chinese Input Method” can be fully functional to install in the PocketPC, the following installation requirement should be achieved:

1. Operating system is recommended to be Windows Mobile Professional 6.0 or above.
2. Asian fonts should be installed to display Chinese characters.
3. Minimum 15 MB free space on the device
4.2 Installation and Deployment

As the end product of G6 is in a .cab cabinet file, it should be placed to any location of the device path. In the following figure, the installation package of G6 named G6SIP.cab is placed in the “//My Device/My Documents/” folder.

![Figure 4.1 Before Installation of G6](image)

Since .cab file in Windows Mobile environment is a Windows installer, user can simply click on G6SIP.cab to start the installation process. Without any annoying step, G6 Bihua Chinese Input Method will be auto-installed to the device. Figure 4.2 shows the situation on the installation process.
After G6 is successfully installed to the device, user can choose G6 whenever Input Method is allowed to be selected. By the way, the installation package can be deleted to save space.
4.3 Fundamental Functions

In Windows Mobile, while G6 is selected, the graphical user interface will be displayed as below:

![Figure 4.4 While G6 is Selected](image)

In figure 4.4, the button 「六」 and 「單字」 are highlighted. It means that the input method is in the G6 and single-character mode, which are the default setting when G6 is first launched. At this moment, user can start inputting the keystrokes to type any Chinese characters / phrases.
4.3.1 Candidate List Display

To select a target candidate, a result candidate list should be provided for user. In my implementation of G6, I have chosen the listbox for outputting the result candidates. Whenever user inputs a keystroke, the candidate list will be displayed to show the matched results from the input key code. At the same moment, the input keystroke(s) will also be listed for user to know what he has input. Above the listed input keystrokes, the current and total page number will be displayed to indicate user how many pages are in total and which page he is browsing. The PageUp and PageDown Keys help in page number increment and decrement respectively. By the way, user can still page-up even he is currently at page 1. The candidate list will be updated and showed the result candidates in the last page.

In addition, the list is designed to contain 6 result candidates per page. With the limitation the candidate list will not block the content of the text, showing the results in 6 rows is the optimum decision to use all of the entire space for display.

4.3.2 Single-character Mode

In the single-character mode, whenever user input a keystroke, the result candidate list is shown to display the matched results with the input keystroke(s). It is the dynamic function in implementation of G6 to avoid an annoying step which requires user to press an output key for showing the results. While inputting the keystrokes, if user pressed the backspace button, the latest input keystroke will be erased and the result candidate list will be updated. If all the keystrokes are removed, the candidate list will also be destroyed. The details of associate candidate list will be discussed later.
The following diagram shows the basic usage in single-character mode:

**Figure 4.5** Basic Usage in Single-character Mode
4.3.3 Two-character Mode

Similar to single-character mode, user can input keystrokes to display candidate list in two-character mode. Dynamic function is enabled just like inputting single character. The major difference from single-character is the result candidates will be shown in 2 characters. User can save time to input two characters with at most 6 keystrokes. But unlike single-character mode, the associate candidate list will not be displayed. The following diagram shows the basic usage in two-character mode:

![Diagram showing basic usage in two-character mode](image)

**Figure 4.6** Basic Usage in Two-character Mode
4.3.4 Three-character Mode

Usage in three-character mode is very similar to two-character mode. Whenever user input a keystroke, the result candidate list will display all the matched results in 3 characters. User can enter this mode and select their expected results if he wants to enter a three-character phrase. Same to two-character mode, associate candidate list will not be shown in this mode. Figure 4.7 shows the basic usage in three-character mode:

![Figure 4.7 Basic Usage in Three-character Mode](image-url)
4.3.5 Multi-character Mode

In multi-character mode, the number of candidate displayed is not fixed. When user input a keystroke, the result candidate list will show the matched results in at least 4 characters. The results consist less than 4 characters will be displayed only in single-character mode, two-character mode or three-character mode. With using this mode, user can select a long phrase up to 15 characters by inputting at most 6 keystrokes. Figure 4.8 shows the basic usage in multi-character mode:

![Figure 4.8 Basic Usage in Multi-character Mode](image-url)
4.4 Advanced Features

Other than input keystrokes and display the results list, there are some advanced features implemented in Windows Mobile G6.

4.4.1 Traditional / Simplified Chinese Switching

On the bottom-left corner of the G6 graphical user interface, there is a “Traditional / Simplified Chinese Switching” button for user to select inputting Traditional / Simplified Chinese.

The working environment in the Simplified Chinese mode is the same as that in Traditional Chinese mode. User can still enter the single-character mode, two-character mode, three-character mode or multi-character mode. In addition, the character database between Traditional Chinese and Simplified Chinese are different. Since some of the characters / phrases will not appear in both two modes, the result candidates displayed will not be the same in Traditional and Simplified Chinese modes with the same page number.

By default, G6 is in the Traditional Chinese mode. Whenever user input a keystroke, all the results will be displayed in Traditional Chinese Character. While clicking on the “Traditional / Simplified Chinese Switching” button, the Traditional / Simplified Chinese mode will be interchanged and user can type in Simplified Chinese characters / phrases. By the way, when user entering the Simplified Chinese mode, the “Traditional / Simplified Chinese Switching” button will be highlighted to indicate it is in the Simplified Chinese mode currently.
The following figure shows inputting keystrokes in the Simplified Chinese mode:

![Figure 4.9 Simplified Chinese Mode](image)

Figure 4.9  Simplified Chinese Mode
4.4.2 Flexible Input Sequence

In G6 Windows Mobile version, user has a flexibility to have his own input sequence. It allows user to switch the character mode even he is in the intermediate of the keystroke inputting. That is, either user first switch the character mode then input the keystrokes, or first input the keystrokes then switch the character mode, the results candidate list displays the identical result candidates. This reduce time wasted on re-type the keystrokes if user forgot to switch the character mode before inputting the key code. The same logic is also applied to switching the Traditional / Simplified Chinese, for example, if user wants to input a Simplified Chinese phrase, he can have the following input sequences which display the same results:

1. Switch to phrase → Switch to Simplified Chinese → Input keystroke
2. Switch to Simplified Chinese → Switch to phrase → Input keystroke
3. Input keystroke → Switch to phrase → Switch to Simplified Chinese
4. Switch to phrase → Input keystroke → Switch to Simplified Chinese
5. Switch to Simplified Chinese → Input keystroke → Switch to phrase
6. Input keystroke → Switch to Simplified Chinese → Switch to phrase

In other words, user can have any of the above sequences to input a Simplified Chinese phrase. (Except the input sequence of the keystrokes) In addition, whenever user switches the character mode or interchanges the Traditional / Simplified Chinese mode, the page number of the result candidate list will be reset to 1. That means if the user is browsing page 3 of the results in single-character mode, if he switched to multi-character mode, the candidate list will be updated to display the first page of the matched results.
The following figure illustrates the flexibility of the input sequence:

Figure 4.10  Flexibility of Input Sequence
4.4.3 Other Characters Selection

Other than inputting Chinese characters, G6 in Windows Mobile version also provides some common used characters for user to select. All of those characters are one-click input. In other words, user clicks on what he sees on the button, the corresponding character will be output.

To group those common used character keys, I divided it into 4 modes, which are G6 mode, Roman mode, Numeric mode and Symbolic mode.

4.4.3.1 G6 Mode

In G6 mode, the common used characters are the full-width punctuation marks of Chinese. The background color of the buttons is in gray by default. The one-click character buttons includes 「，」，「。」，「？」，etc. Also, a newline button is added for user to open a new paragraph while typing the text. Figure 4.11 shows the one-click characters in G6 mode:

![Figure 4.11 One-click Characters in G6 Mode](image)
4.4.3.2 Roman Mode

The Roman Mode of G6 is for typing Roman alphabets a ~ z and A ~ Z. Although the default input method in Windows Mobile includes these Roman characters output, it is not convenient for users to switch the input method when they want to insert a Roman character while using G6. Considering this kind of concern, I also implemented the Roman mode for user to input English.

Besides, Roman mode is further divided into 2 sub-modes, which are Shifted-character mode and Unshifted-character mode. By pressing the “CAP” button at the bottom-left corner in the Roman keyboard, user can interchange these 2 modes to type the Roman alphabets in uppercase or lowercase. Moreover, comma, full-stop, space and newline keys are also included in Roman mode.

Figure 4.12 and 4.13 shows the one-click characters in Roman mode:

![Figure 4.12 One-click Characters in Roman Unshifted Mode](image-url)
4.4.3.3 Numeric Mode

Numeric mode in Windows Mobile G6 is a simple mode to input numeric characters. The graphical user interface will only display Arabic numerals 0 ~ 9 for user to select. Other than the numbers, comma and decimal point are also included on the keyboard.

Figure 4.14 shows the one-click characters in Numeric mode:
4.3.4 Symbolic Mode

The last but not least is the Symbolic mode. In this mode, all the common used symbolic characters can be selected by one-click only, which includes “:”, “?”,”!” etc. Figure 4.15 shows the one-click characters in Symbolic mode:
4.5 Additional Features

4.5.1 Help Menu

If user is the beginner to use G6 in Windows Mobile version, probably he may not know what G6 is and how to control to input Chinese. To help user solving this kind of problems, a Help Menu is added to teach user deal with G6.

![Figure 4.16 Help Button in G6](image)

In figure 4.16, there is a Help button on the top-left corner of the keyboard. Whenever user clicked on this button, the Help Menu will be displayed in full-screen window size.

The Help Menu consists of 5 pages, including G6 introduction, characteristics of G6, graphical user interface description, developers in G6 Team, and contact method.
The following figure shows the page content in Help Menu:

Figure 4.17  Help Menu Page Contents
4.5.2 Wildcard Key

In some cases, if user does not know the full key code of a particular character or phrase, probably the result candidate list will not display what he wanted to input. To help user inputting the characters / phrases easily, a wildcard key is added to solve the uncertainty of the key code.

In my implementation of G6, a wildcard key means it can be any of the 5 keystroke types refer to its corresponding position, for example, I input 「一」 for the first stroke and press a wildcard key for the second stroke, the result candidates which are matched with 「一一」, 「一丷」, 「一丶」 or 「一フ」 will be shown. Also, this powerful key can be used in any times and at any position of the 6 key codes.

Figure 4.18 illustrates the scenario of using the wildcard key:

Figure 4.18 Wildcard Key Usage
4.5.3 Associate Candidate List

Associate candidate list in G6 is a very important feature for end users. User may select the associate candidate from the list to save time for inputting another new character / phrase.

Note that the associate candidate list will only appear when user selected a single character. As a result, while user is in two-character mode, three-character mode or multi-character mode, associate candidate list surely will not be displayed. In case, if the character selected will not have any associate character / phrase, the list will not be shown also.

When user selected a character from the list, the associate candidate list will be generated according to the selected character. The associate candidate list can be in two-character mode, three-character mode or multi-character mode. User can choose the number of associate character(s) by clicking on the button on the right-hand side of the list. Also, if the user selected a single character in the associate character list, the consecutive associate candidate results will be shown until user close the list window or select multi-character. By the way, to close the associate candidate list, user can simply click on the backspace button.

After the associate candidate list appears, user can ignore the list and continue to input the keystrokes. When user input the first keystroke, the associate candidate list will be destroyed and the normal result candidate list will be shown instead. User can have their own choice to choose whether selecting the associate candidate or generating a new character / phrase list by inputting the keystrokes.
Figure 4.19 illustrates how the associate candidate list works:

**Figure 4.19** Associate Candidate List and Consecutive Associate Candidate List
4.5.4 Key Code Dictionary

Although wildcard key help user a lot in typing the Chinese characters without knowing the full code of keystroke, there is an additional function in Windows Mobile G6 to find out the full key code of a particular character / phrase. This function is called Key Code Dictionary.

In the G6 mode, by pressing the Dictionary button, a popup window will be immediately display at the middle of the screen. User can first copy the character / phrase which they would like to search for its full key code. Then, press on the Paste button of the popup window, what the user just copied will be pasted on the textfield.

By pressing the Go button on the right-hand side of the textbox, the popup window will be enlarged and the full code of the keystrokes will be shown at the bottom of the window. The searched code results will be displayed in both Traditional Chinese mode and Simplified Chinese mode, which means user should enter the corresponding mode and type in the full key code to obtain the target candidate.

For practical usage, while user browsing on the webpage and want to find out some characters / phrases in full G6 key code, he can copy those characters / phrases and use the Dictionary function to decode.

In the following figure, it shows the usage of Key Code Dictionary with above example:
Figure 4.20  Key Code Dictionary Usage
4.5.5 Theme Selection

Maybe some of the users do not like the default theme of the graphical user interface, I have also designed another theme for user to choose in G6. For further development, numerous themes can be added to Windows Mobile G6 to ensure there must be a favorite theme for any user.

To change the theme setting, user should click on the input method logo and click “Options…”.

![Input Method Options](image)

**Figure 4.21** Input Method Options

In the dropdown list, select G6 Bihua Chinese Input Method and click on “Options…”, a popup dialog is shown for prompting user to select a theme set. User can click on the radio button to choose one of the themes he wanted to set. Besides, there is a preview of the selected theme next to the radio buttons. By clicking the “OK” button for confirmation, the selected theme will be set.
In figure 4.22, there are steps for changing the theme:

**Figure 4.22** Theme Changing
5. Discussion

In the implementation of G6 in Windows Mobile version, I had faced several difficulties on the technical parts.

The first part was to create the 4 modes in G6. The mode information could be stored by a variable, but the keyboard switching and the keys mapping parts were difficult to handle. Finally I created different keyboard arrays for stored the keyboard information on each mode, and luckily it worked.

The second part should be the handling of sticky key pressing. Sticky key included all the mode switching buttons. While pressing on those sticky buttons, whether the key flags should be set to F_UP or F_DOWN was not an easy task. As there were so many situations, I had to make a suitable solution to ensure there would not be any bug in sticky key handling. Finally, I decided to un-press all sticky keys in each mode when a new sticky key was being pressed. After that, the bug did not appear anymore.

The listing of the result candidates was also a difficult part for me. Although the selection logic was not very complicated, writing program code was not easy as writing pseudo code. Since I was not very familiar with the C++ code and the APIs in developing of Windows Mobile, I spent much time on browsing MSDN and learn from other people’s references. The data type handling was a key point in this project. Outputting a character from the unsigned long hexcode took me lots of time on try and error.
The bottleneck in my project was to select a target candidate from the listbox. I spent more than a month on solving this technical problem. Although I had found tones of reference on the Internet, none of it gave me a correct solution. I kept modifying the source code to test on the deployed product. Lastly, I found out the solution after million times of failure.

Besides, I spent the rest of the time on modifying the graphical user interface. Actually there were more than 10 versions of the amended GUI. To design the best one need to think of the view from end users. Up to now, the GUI is much better compared to the earlier versions.
6. Conclusion

To conclude, G6 Bihua Chinese Input Method in Windows Mobile version is an input method which can be selected anytime whenever there is a textfield for inputting. It has a powerful functionality to input any Chinese character / phrase in at most 6 digits of key code. The main characteristic of G6 is easy to learn. Even the user has not used G6 before, he can think of the key code easily provided that he knows the keystroke sequence of the Chinese character. The multi-character mode of G6 allows user to input a long phrase within a few seconds. This significantly reduces the time consumption on inputting the key code and selecting a target candidate. I wish all G6 users will enjoy this revolution on Chinese Input Method.
7. Reference

http://en.wikipedia.org/wiki/Mobile_operating_system


http://en.wikipedia.org/wiki/ActiveSync


8. Glossary

WM
Windows Mobile, an operating system for Pocket PCs

Emulator
Emulation refers to the ability of a computer program or electronic device to imitate another program or device.

SIP
Software-based Input Panel, the main input method on Pocket PC.

GWES
Graphics, Windowing, and Events Subsystem.

SDK
Software Development Kit, typically a set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform.

Dvorak
Dvorak Simplified Keyboard, a keyboard layout patented in 1936 by August Dvorak.
9. Appendix

9.1 Guideline to deploy the project

To deploy the source project for the developer, here are the steps:

1. Install ActiveSync, Microsoft Visual Studio and Windows Mobile 6 Professional SDK.

2. Open the project folder and double-click on g6sip.vcproj.

3. Select “Windows Mobile 6 Professional Emulator” as the target device.

4. Click on “Build”, “Deploy g6sip”.

5. Save a project solution, and G6 will be installed to the target emulator.