

Connect6 Programs on Mobile Devices

Ji-Hong Zheng¹, Chia-Yun Hu¹, I-Chen Wu¹, Wen-Jie Tseng¹, Ching-Hsuan Wei¹, Hung-Hsuan Lin¹,
Chieh-Min Chang¹, Hao-Hua Kang¹, Hsiu-Chuan Lin^{1,2}

¹Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{hellogg1,shelley,icwu,wenjie,lano,stanleylin,aup,kangbb,sam}@java.csie.nctu.edu.tw

²ThinkNewIdea Inc., Hsinchu, Taiwan
sam@ethink4u.com

Abstract

In this demonstration, we port our current Connect6 program, NCTU6, to mobile devices, such as iOS devices, Android devices. However, computing powers of mobile devices are also relatively weaker when compared with high-end machines. For example, the CPU speeds of mobile devices are about 1/5 to 1/8 of those of desktops; and the memory of mobile devices is usually limited to several megabytes. In this demonstration, we tune the program NCTU6 to fit in mobile devices, while maintaining reasonable strength. The program also won the silver medal in TCGA 2012. In addition, we also support Connect6 puzzles in the program.

Keyword: Connect6, mobile, NCTU6, iOS, Android.

1. Introduction

Connect6 [14][15] is a kind of six-in-a-row game that was introduced by Wu et al. In this paper, we port our current Connect6 program, NCTU6, to mobile devices. The program NCTU6 won the champions of computer Connect6 tournaments and man-machine Connect6 tournaments several times from 2006 to 2011 [9][13][16][17][18].

Mobile devices normally have limited resources in memory and computing. In order to port NCTU6 to mobile devices while keeping the strength of NCTU6, we need carefully reduce the memory and computing consumption. For example, in NCTU6, we implemented a line pattern table with size 19 Megabytes. Unfortunately, in most mobile devices, such as iPhone, iPad, Android smartphones, Android tablets, the memory sizes are better not over 20 Megabytes [11]. In addition, NCTU6 also consumes a huge amount of memory for maintaining the data of search trees. In this paper, we redesign the system to reduce the memory greatly to fit in mobile devices.

Computing powers of mobile devices are also relatively weaker when compared with high-end machines. In our experiments, the CPU speeds of mobile devices, e.g., ASUS Transformer TF101, are about 1/5 to 1/8 of those of Intel® Core™ i5-2400 CPU 3.10GHz machines. Besides, the reduction of memory also incurs extra computing overhead. Thus, the search size should be carefully controlled in the mobile version of NCTU6.

We also implemented Connect6 puzzles (詰棋) and

some other functionalities, such as openings, hint lines, auxiliary lines. We collected games from an Internet server, named Littlegolem [10], and used NCTU6 to help find about 6000 interesting puzzles. Among these puzzles, about 530 puzzles were selected by Wang, a champion of NCTU Connect6 Open. Then, like Angry birds [3], our puzzles are arranged in several stages, currently 8 stages, each of which has 12 puzzles to be solved. In the last 4 stages, puzzles are selected at random.

Other functionalities include openings, hint lines, auxiliary lines. We added openings to make games more versatile, so that the players are not sick and tired of playing the same openings. We also added hints for those lines with the patterns, such as threats, live twos, live threes, dead threes, etc., to help beginners find these patterns. We also added auxiliary lines, since many mobile devices are too small to click at the right spots by fingers. Using auxiliary lines, users can identify the locations to place stones.

All of the above are integrated into a mobile program in two versions, one in Android [1], and the other in iOS [8]. The program, named *mobile6*, won the silver medal in the Connect6 tournament of TCGA 2012 [6], and also won the first prize in an expert tournament, named Bi-Lao-Da-13, held in the Littlegolem website [10]. This shows that we successfully squeezed our program NCTU6 into mobile devices while keeping a reasonable strength.

2. System Description

In this section, we investigate one important issue for memory size in Subsection 2.1, and describe our design framework in Subsection 2.2.

2.1 Game Editing Module

First, we discuss how to reduce the memory. In NCTU6, we maintain a big line pattern table with size 19 Megabytes. More specifically, since the current Connect6 board is 19x19, the line pattern table needs to cover all cases in a line of size 19, that is, 2^{19} entries in this table. For each entry, we need 19 data to maintain the information of the 19 locations. Thus, the size of the table is at least 19×2^{19} . In real case, we used about 19 Megabytes.

```

+-----+
. . . @ @ @ @ . @ . . . . . . . . .
+-----+

```

Figure 1. Decomposing a line of size 19 into two lines of size 15. (@ is a piece, and . is empty.)

In order to reduce the size, we decompose a line with 19 into two sub-lines with 15, and thus reduce the size to about 1 Megabytes. For example, the line in Figure 1 is decomposed into two. Since the left one apparently includes the most significant data, we can use the left one to represent the values of each empty. In fact, this research work needs to solve many exceptional cases, but the further details are omitted due to the size limit of this paper.

2.2 System Framework

We design a framework in order to develop Connect6 or other similar board games on mobile devices easily. Figure 2 (below) shows the framework. This framework includes the following major modules, GUI module, control module, an interface to AI programs, and knowledge base module.

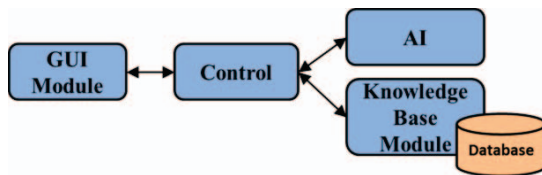


Figure 2. Framework of mobile Connect6.

The control module is responsible for transmitting messages among all the other modules. In the future, we will add the network module and let the control module to access network via it.

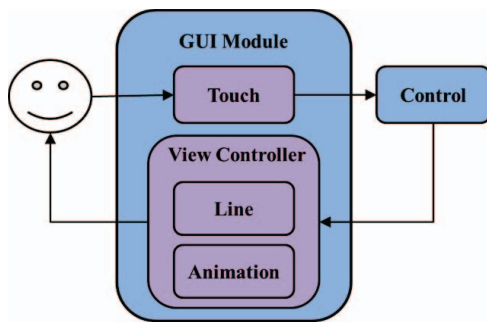


Figure 3. GUI module.

In the GUI module, as shown in Figure 3, we need to support touch events from users, and also support the messages from the control module to display figures in the window. A touch component is used to analyze the touch events, mainly for gestures, such as tap, swipe, drag, pinch out, and pinch in.

For displaying figures in the window, two major

components are designed for displaying hint/auxiliary lines and piece moving animations, respectively. The animation component is used to show the animation of images, especially for piece moving. In iOS, rendering for animation is well supported in the class UIView. Rendering for animation is supported in Android 3.0 or above, but not below. Therefore, it took more efforts in rendering animation for the old Android version.



Figure 4. AI interface.

In this framework, we define an interface to access AI programs as shown in Figure 4, normally written in C/C++. In this way, we can directly use the AI program developed by AI programmers. In Android, we use Android-NDK [1] to help compile, while in iOS we used LLVM [5]. The AI program was modified from NCTU6 as mentioned in Subsection 2.1. The program is currently divided into four ranks with different strengths. These ranks are subject to changes. Currently, Ranks 1 and 2 are purely based on move evaluation without involving in complex tree search, so the response time is fast. Rank 1 chooses weaker moves sometimes, so it is supposed to be weaker than Rank 2. Rank 4 is the strongest one. The program with Rank 4, named *mobile6*, won the silver medal in the Connect6 tournament of TCGA 2012 [6], and also won the first prize in an expert tournament, named Bi-Lao-Da-13, held in the Littlegolem website [10].

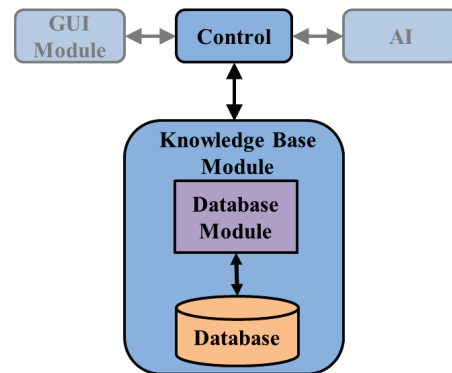


Figure 5. Knowledge base module.

To access puzzles and openings, we support knowledge base module in the framework as shown in Figure 5. We stored the collection of connect6 puzzles and openings to the database. The database module is used to access the database via SQLite.

3. System Demonstration

In this section, we demonstrate the operations of our mobile programs. Although we implemented on

both iOS and Android, this section displays the GUI of the iOS version on iPad. We show 12 typical screenshots for the program as follows.

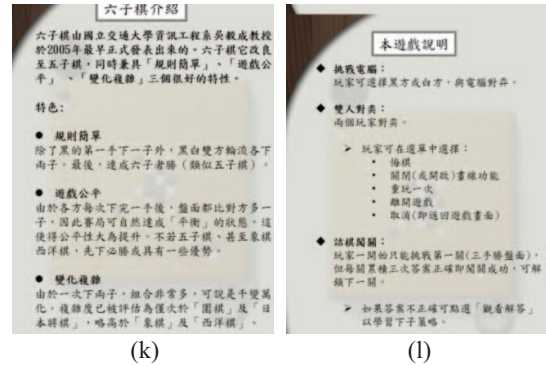
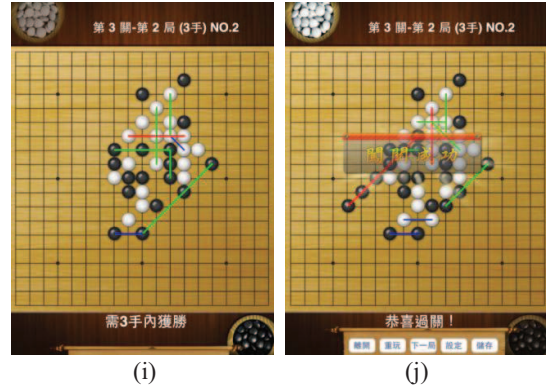
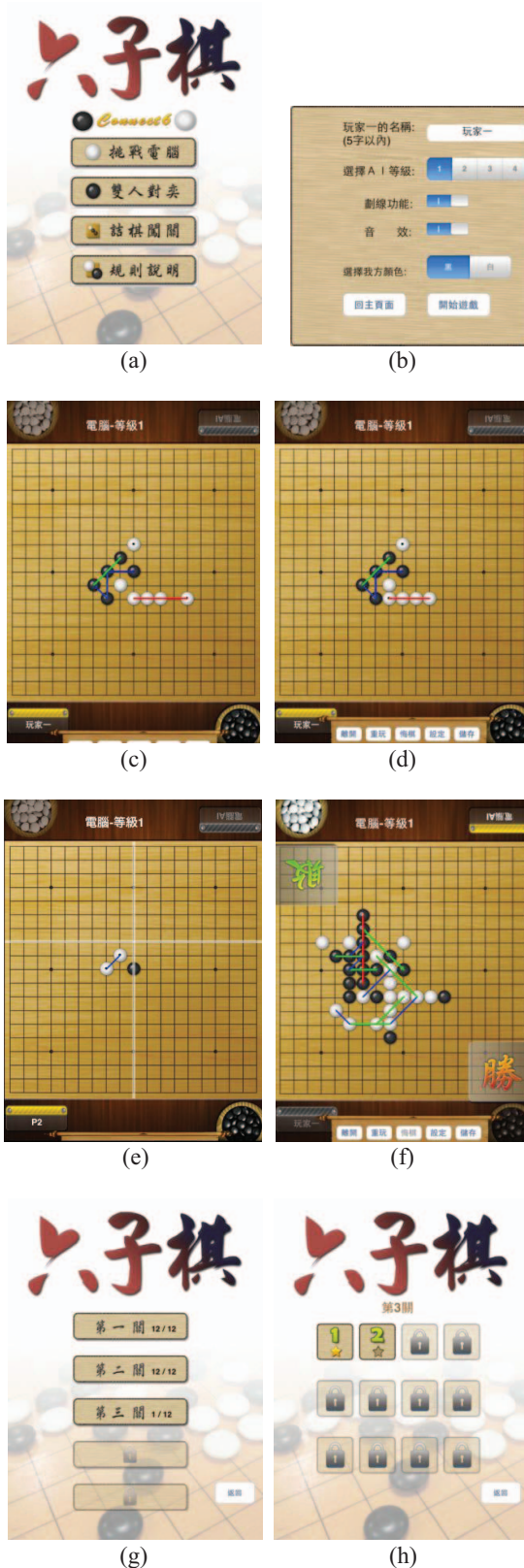


Figure 6. Screenshots.

The 12 screenshots in Figure 6 are respectively described as follows.

- (a) There are four buttons on the cover page. The first three buttons represent three different game modes, such as one-player game (playing against computer), two-player game and puzzle games. When users press the first button (挑戰電腦), the setting dialog in one-player game will be displayed as screenshot (b).
- (b) The setting dialog for playing against computer. Users can set the player name, select an AI rank (from four ranks as described above), turn on/off the hint line effect and sound effect, and choose the color of pieces.
- (c) A game board with hint lines in one-player game.
- (d) Menu bar for quitting games, replaying, withdrawing moves, settings, saving moves.
- (e) Auxiliary lines to help players identify locations.
- (f) A winning display. In this screenshot, black wins the game, and then the menu bar is popped up automatically.
- (g) Stages of puzzle games. There are currently 8 stages (subject to changes) to be selected. If users selected a stage, a list of puzzles will be displayed as screenshot (h).
- (h) A list of puzzles. There are currently 12 puzzles to choose in a stage.
- (i) A display for a puzzle. Users have to solve the puzzle in a limited number of moves. When the

- number exceeds, a failure message is popped up.
- (j) A display when a puzzle is solved. When users solve a puzzle, the menu bar will be popped up automatically, and then users can press the “next puzzle” button (下一局) to play the next puzzle.
 - (k) An introduction to Connect6.
 - (l) The user guide for playing this mobile game.

In general, the operation flow is subject to changes, e.g., rank number, stage numbers, puzzle numbers in one stage.

4. Installation and Links to the Software

Currently, the program has been written in both Android and iOS. They have been ported and tested in ASUS Transformer TF101, HTC Flyer, HTC Desire HD (above are in Android), iPad2, iPhone 4 (above are in iOS). It is planned that they will be uploaded to Apple App Store [4] and Google Play [7] before the conference is held.

In our experiments, the memory consumed during running is about 20 Megabytes and that for AI is about 10 Megabytes. As for the response time, Ranks 1 and 2 respond immediately, while Rank 4 (the strongest one) is over one minute in most cases. However, Rank 4 has a winning rate 38% against the original NCTU6 program and won the silver medal in TCGA 2012.

Acknowledgement

The authors would like to thank the National Science Council of the Republic of China (Taiwan) for financial support of this research under contract numbers NSC 95-2221-E-009-122-MY2 and NSC 97-2221-E-009-126-MY3, ThinkNewIdea Internet Technology for their support for graphic user interfaces, Chi-Gong Wang for helping select puzzles, and B.J. Wu for his valuable comments.

References

- [1] Android. Available at <http://www.android.com/>
- [2] Android NDK. Available at <http://developer.android.com/tools/sdk/ndk/index.html>
- [3] Angry Birds. Available from <http://www.rovio.com/en/our-work/games/view/1/angry-birds>

- [4] Apple App Store. Available at <http://itunes.apple.com/gb/app/apple-store/id375380948?mt=8>
- [5] Apple LLVM Compiler. Available at <https://developer.apple.com/technologies/tools/w-hats-new.html#llvm-compiler>
- [6] Connect6 tournament of TCGA 2012. Available at <http://tcga.ndhu.edu.tw/TCGA2012/>
- [7] Google Play. Available at <http://play.google.com/intl/zh-TW/about/index.html>
- [8] iOS. Available at <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>
- [9] Lin, H.-H., Sun, D.-J., Wu, I.-C. and Yen, S.-J., The 2010 TAAI Computer-Game Tournaments, ICGA Journal, Vol. 34(1), March 2011.
- [10] Littlegolem. Available at <http://www.littlegolem.net/jsp/index.jsp>
- [11] Stackoverflow.com, Memory uses limit on iPhone, 2010.
- [12] SQLite, Available from <http://www.sqlite.org/>
- [13] Wu, I.-C., Lin, H.-H., Lin, P.-H., Sun, D.-J., Chan, Y.-C., and Chen, B.-T, Job-Level Proof Number Search for Connect6. In the International Conference on Computers and Games (CG 2010), Kanazawa, Japan, 2010.
- [14] Wu, I.-C., Huang, D.-Y., and Chang, H.-C., Connect6. ICGA Journal, Vol. 28(4), pp. 234-242, 2006.
- [15] Wu, I.-C., and Huang, D.-Y., A New Family of k-in-a-row Games. The 11th Advances in Computer Games Conference (ACG'11), pp. 180-194, Taipei, Taiwan, 2005.
- [16] Wu, I.-C., and Lin, P.-H., NCTU6-Lite Wins Connect6 Tournament, ICGA Journal, Vol. 31(4), 2008.
- [17] Wu, I.-C. and Lin, P.-H., Relevance-Zone-Oriented Proof Search for Connect6, the IEEE Transactions on Computational Intelligence and AI in Games (SCI), Vol. 2(3), pp. 191-207, September 2010.
- [18] Wu, I.-C., and Yen, S.-J., NCTU6 Wins Connect6 Tournament, ICGA Journal, Vol. 29(3), pp. 157-158, September 2006.