Materials Today: Proceedings xxx (xxxx) xxx

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Design and development of artificial intelligence (AI) based board game (Gobang) using android

L. Venkateswara Reddy [a,*], S. Saravana Kumar [b], S. Sugumaran [c], K. Lavanya [d]

[a] Dept. of IT, Sree Vidyanikethan Engineering College, Tirupati, AP, India
[b] Dept. of CSE, CMR University, Bangalore, KA, India
[c] Dept. of ECE, Vishnu Institute of Technology, Guntur, AP, India
[d] Dept. of CSE, School of Engineering, SPMVV University, Tirupati, AP, India

ARTICLE INFO

Article history:
Received 11 December 2020
Accepted 28 December 2020
Available online xxxx

Keywords:
Android
Operating system
Artificial intelligence
Gobang
Mobile device

ABSTRACT

In modern society, more and more cell phones come into the work and life of ordinary people. With the popularization and application of the smart-phone technology, the operating system of mobile devices in the use of mobile phones is very broad, like Android, Symbian and iPhone. Among them, Android is a very popular open source mobile operating system for mobile phone and tablet personal computer. As the rapid development of computer and smart-phone, intelligent games have become an integral part of the universal life, not only does it make people entertainment, but also to develop people's intelligence. Intelligent game is one of the important branches of Artificial Intelligence. The artificial intelligence techniques are applied to the Gobang in this article. An intelligent Gobang game is designed and realized in the game between human and computer in Android operating system.

Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

1. Introduction

As the rapid development of computer and smart-phone, intelligent games havebecome an integral part of the universal life, not only does it make people entertainment, but also to develop people's intelligence, as in this paper as the mainspoken welcome to play smart to tap people's talents and head of the clever level. Gobang is very popular in civil as a sport, and Android operating system is currently a very popular mobile operating system for mobile phones and tablets, to become acquainted with Android operating system and Android development techniques, artificial intelligence research. I decide to use the Java todevelop aintelligence Gobang game with human–machine war in Android operating system.

This article is talking about Android development tools and java development language to develop a custom interface Gobang game, the game program can achieve against human and computer playing chess, and the program can be able to intelligently determinethe game winning side and the end of the game or not, to achieve two sidegame. This game can be installed in the mobile-phone with Android operating systemand also can be installed in the Android simulator in PC.

1.1. Android OS

For mobile devices, Android is an operating system, middleware and core applications software stack. In 2005, Google Inc. acquired the original developer of the app, Android Inc. The mobile operating system for Android is based on a modified version of the kernel for Linux. Google and other Free Phone Alliance members have partnered on the creation and release of Android. The maintenance and further development of Android is the job of the Android-Open-Source-Project (AOSP). The Android operating system is the best-selling platform for mobile phones in the world.

Android has a wide community of applications ("apps") written by developers that extend the devices' functionality. For Android, there are currently over 150,000 applications available. The online app store operated by Google is the Android Market, while applications. It is also possible to download from third-party pages. Developers write primarily in the Java language and manage the framework via Google-built Java libraries [1].

The display interface of Android operating system is shown in Fig. 1.1.

* Corresponding author.
E-mail address: lakkireddy.v@gmail.com (L. Venkateswara Reddy).

https://doi.org/10.1016/j.matpr.2020.12.1144
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the Emerging Trends in Materials Science, Technology and Engineering.

Fig. 1.1. Android operating system.

1.2. Gobang game

Gobang is an abstract board strategy game and is known as Five-in-a-Row or Gomoku as well. It is typically played on a 15–15 board with black and white stones, but since pieces are not moved or removed from the board once put, Gobang can also be played as a game of paper and pencil. This game is recognized under various names in many countries.

The conventional Gobang rule is that black plays first, and players alternate by putting an empty intersection with a stone of their color. The winner is the first player in either direction to achieve an unbroken row of five pieces in a row. Gobang game model is shown in Fig. 1.2.

I had a course which is Artificial Intelligence in the 3rd semester and learnt the knowledge about the computer AI games algorithm. Android operating system is a very popular mobile operating system for mobile phone and tablet personal computer. So the AI board game applicationof Android is a valuable area.

To developa board game – Gobang (Five-in-a-row) asa application software in Android operating system. Learn the development tools of Android. Analysis and design the algorithm of the board game – Gobang. Use JAVA language for the development. Make sure the product can be used in Android operating system of mobile phone (Installed in simulator of computer for testing and presentation).

A product of board game – Gobang, as normal application software, which is installed in a Android mobile phone or Android simulator. Realize playing chess between human player and computer AI player. The Gobang game can be installed in the Android system, [1] as an application.

2. Literature survey

2.1. Artificial intelligence

One of the newest sciences is AI. Soon after World War II, work began in earnest, and the name itself was coined in 1956. In addition to molecular biology, scientists in other disciplines frequently quote AI as the ‘‘area in which I would most like to be”. AI currently encompasses a wide variety of subfields, ranging from specific fields including learning and understanding to personal hobbies such as playing games, proving mathematical concepts, writing poetry, and diagnosing illnesses. Intellectual activities are systematized and automated by AI and are thus theoretically applicable to every area of human intellectual activity. It is actually a universal realm in this sense [2].

AI achieved its biggest achievements in the 1990 s and early 21st century. Artificial intelligence is used in the technology industry For logistics, data mining, medical diagnosis, etc. The success was attributed to several factors: the growing computing capacity of computers, a greater focus on speed resolution [1].

Today, what can AI do? A succinct response is difficult, since in so many subfields, there are so many activities. Sample a game play application here: IBM’s Deep Blue Becoming the first software program to overcome the world champion in a chess match by defeating Garry Kasparov by either a score in an exhibition gameof 3.5 to 2.5. Kasparov said he sensed a “new kind of intellect” from him across the board. The match was described by Newsweek magazine as “The Brain’s Last Stand.”

2.1.1. Adversarial search and games

Competitive conditions in which the interests of the agents are in dispute offer rise to adversarial quest issues that are sometimes referred to as games. A branch of economics, mathematical game theory, considers any multi-agent situation as a game if the effect of each agent on the others is ‘‘important,” irrespective If they are positive and cooperative agents. In AI, what theoreticians call deterministic, turn-taking, two-player, zero-sum perfect knowledge games are usually perfect knowledge ‘‘games” of a very specialized type. This means deterministic, completely measurable conditions in our language. There are two agents whose actions must alternate and where the utility values at the end of the game are both equal and opposite. For instance, if one player wins a chess game (+1), the other player will necessarily lose (−1). This opposition between the utility functions of the agents is what renders the situation adversarial [4].

One of the first activities performed in AI was game playing. By 1950, Konrad Zuse (discoverer of the first programmable machine and the first computer program), Claude Shannon (innovator of the theory of knowledge), Norbert Wiener (father of modern systems engineering), and Norbert Wiener (father of modern control systems) tackled chess. and by Alan Turing almost as soon as computers were programmable. There has been steady improvement in the level of play since then, to the extent where computers have surpassed humans in checkers and Othello, defeated human champions in chess and backgammon (although not every time) and are competitive in many other sports. The primary exception is Go, where computers work at the level of amateurs.

2.2. Gobang

2.2.1. Board and stones

In Gobang, At the 225 crossings between 15 vertically and 15 horizontal lines, players played. The vertical lines are labelled ’a’

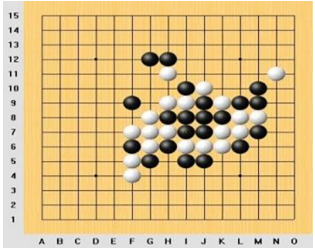

Fig. 1.2. Gobang game.

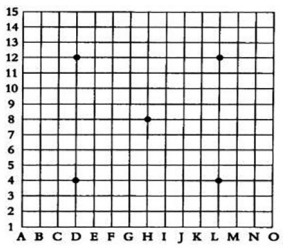

Fig. 2.1. Gobang board.

to 'o' from left to right; the horizontal lines are numbered from 1 to 15 from the bottom to the top. The board is shown in Fig. 2.1.

Gobang is Played on a board among two adversaries by making motions with Black and White men named stones. So there are two kinds of stone, black and white, totally 225 pieces, Black stones 113 and White stones 112.

2.2.1.1. The rules of Gobang. In Gobang, simple rules main to a highly compositegame, played on a board. Atfirst, the board is empty, as shown in Fig. 2.2.

Two teams, Black and White, in turn move by putting on an empty intersection a stone of their own color, henceforth called a square. The player starts the game with the black stones. The first step to the middle of the board has to be made. Instead of putting them in the centre of the circles, all motions are placed at the intersections of the arcs. After that, white moves, as shown in Fig. 2.3, to every empty position on the board.

The teams, putting the moves on the board, take turns. During the game, the stones once put on the board never move again or can be caught, meaning the stones already on the board should not be touched. Only stones can be attached by the players to the board. The player who first makes a line (horizontally, vertically or diagonally) of five consecutive stones of his color wins the game. Fig. 2.4 demonstrates an example game. Put numbers on the stones for better understanding, helping us understand the sequence of movements. Of course, there are no numbers on the stones in the actual game. By the 16th move, White won the game, forming the horizontal line of 5 white stones.

The Fig. 2.5 shows the examples of all possiblewinning lines. If the board is completely filled, and no onehas five-in-a-row, the game is drawn, nobody win [3] (Fig. 2.6 Fig. 2.7 Fig. 2.8 Fig. 2.9 Fig. 2.10 Fig. 2.11 Fig. 2.12 Fig. 2.13).

2.2.1.2. Terms and definitions. ROW:

A mixture of stones of the same color, either placed on a diagonal, vertical, or horizontal line, restricted by the edge of the panel, the opposing stone, or the free intersection, and where no opposing stones are placed between stones of their own.

UNBROKEN ROW:

A row where a free intersection does not occur between just any stones.

FIVE IN A ROW:

An unbroken row with five stones.

FOUR:

Live Four and Rush Four include a row of four stones to which you can add one more stone to reach Five in a Row.

LIVE FOUR:

The unbroken row of four stones ("four") to which you can add one more stone in two separate ways to reach Five in a Row.

RUSH FOUR:

In the same row the same color connected to four pieces, the different with Live Four is that one end of which the other pieces were blocked, it is only be formed Five in a Row at the other end.

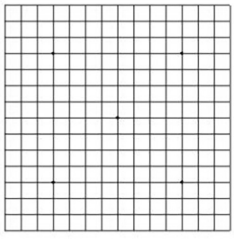

Fig. 2.2. Initial board.

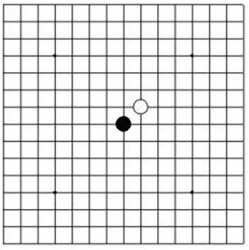

Fig. 2.3. Game start.

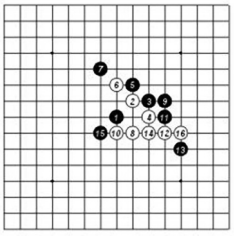

Fig. 2.4. White won.

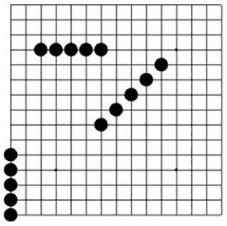

Fig. 2.5. State of win.

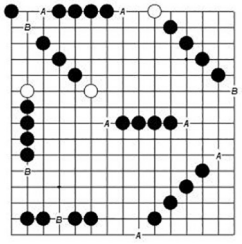

Fig. 2.6. Four.

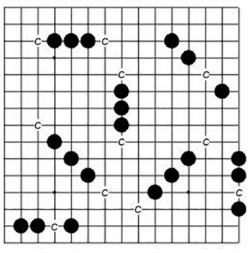

Fig. 2.7. Live three.

THREE:

A row of three stones to which you can add another stone to reach a Four, like Live Three and Sleep Three, without a Five in a Row being made at the same time.

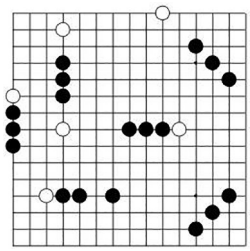

Fig. 2.8. Sleep three.

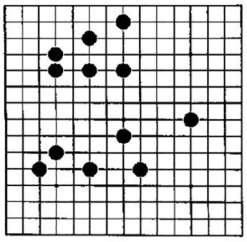

Fig. 2.9. Live two.

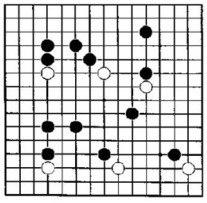

Fig. 2.10. Sleep two.

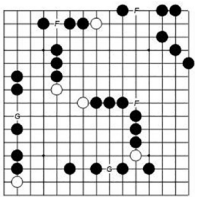

Fig. 2.11. Double-Four.

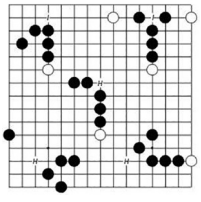

Fig. 2.12. Four-Three Forks.

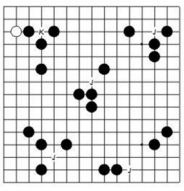

Fig. 2.13. Double-Three.

Live three:

A row with three stones can add one more stone to attain a LiveFour.

- SLEEP THREE: A row with three stones can add one more stone to attain a Rush Four.
- TWO: Two is the base of all the attack action of Gobang, can add one more stone to attain a Three, includes Live Two and Sleep Two.
- LIVE TWO: A kind of Two, can add one more stone to attain a Live Three.
- SLEEP TWO: A kind of Two, can add one more stone to attain a Sleep Three.
- DOUBLE-FOUR: Putting a stone on a junction that around the same period makes upwards of one Three in this alignment that pass one another.
- FOUR-THREE FORKS: Putting a stone on an intersection, which at the same time makes one Four and one Three that meet each other in this intersection.
- DOUBLE-THREE: Putting a stone on an intersection, which at the same time makes more than one Three that meet each other in this intersection?

2.2.1.3. The offense and defense of Gobang. Gobang is a high confrontational movement. When the game start, it begin to enter a bayonet charge, each other for the upper hand, either player cannot be taken lightly, as little as possible to make mistakes, even no mistakes, otherwise it will be lead to a quick lose.

As we know, who can connect five pieces first is winner. The state is from four to five, three to four, two to three, and so on. Therefore, in Gobang game, all the offense and defense begin from battle the‘‘two”and‘‘three”. ‘‘A good beginning is half of success”, this motto used in Gobang is fit [5].

In Gobang, the position selection is crucial. Gobang is actually the best placement by selecting with the right place order, occupy the various important positions step by step, and ultimately win.

The early of a game, the main focus point selection is to maintain contact of the pieces, for the future to create as many as opportunities of Three or Four percent, but also try to limit the good state of opponent. In the course of the game, control the placement through the other placement is possible, such as Live Three and Rush Four is predictable. It’s entirely possible to keep the opponent out of the state with the upper hand to control the opponent’s point until victory. Therefore, in the late of game, it is necessary in the accurate calculation of the premise, the offensive as soon as possible in order to obtain control of the game, or if it delays, opponent was the firstto launch an offensive, it will become a party who is controlled.

When there are multiple points of attack are available, select the point which has more next steps and will not be the first of the opponent.

Offensive divided into single attack and double attack. The single attack means a single line attack or single direction attack, including Three attack moves (Live Three, Four Fill) and Four attack moves (Rush Four, Five Fill). The double attack means the two lines attack or two direction attack, including Three-Three attack moves (Double Live Three, Double Four Fill, Live Three-Four Fill); Four-Three attack moves (Rush Four-Live Three, Rush Four-Four Fill, Five Fill-Live Three, Five Fill-Four Fill); Four-Four attack moves (Double Rush Four, Double Five Fill, Five Fill-Rush Four) [6].

Gobang ideas of win start from one piece, goal is to use various methods to develop Five in a Row as well as long connected on the board, then win. In the course of this development is bound to be the processes of one piece to two pieces, two pieces to three pieces, three pieces to four pieces. So, to say, remember all the states of all

the direction of Two, Three, Four structure. Live up to draw inferences about other cases from one instance, is the basic of learning Gobang, but also the base of flexibility to use the skills to win.

2.3. Preliminary plan

The preliminary plan of the work is shown in Fig. 2.14.

2.3.1. Gobang game algorithm

In general, the Gobang game algorithms include the evaluation function, searching algorithm and outcome judging and so on. As mentioned above, the BFS, DFS, the Minimax algorithm, Alpha-beta pruning and evaluation function are applied to the Gobang game. This paper designs a intelligent Gobang game with human–computer war. Human player play with computer player, The machine will determine how to react to the movement of the human player. To look for the best move, the computer uses searching techniques such as BFS, DFS. The board is large, so we can't scan the entire game space to find the best move, so there's a high computational complexity. Instead, to minimise the time, we need, as much as we can, to cut off the hunt by pruning those branches. The evaluation function is also very critical for good results since, due to the computational difficulty, we could only check for a few depths. The Gobang Game Flowchart Algorithm is shown in Fig. 2.15.

2.4. Architectural design

2.4.1. Architectural design

The architectural design of the paper is shown in Fig. 3.1.

2.4.1.1. Gobang system flowchart. The Gobang game system flowchart is shown in Fig. 3.2. First the game start, then initialization of the game or board. The main control is the system control that is player's turn or computer's turn, realize the game playing between human player and computer player. In every step, system will judge the outcome, if someone win, the game end, or the game draw, the game end.

2.5. Modular design

Modules of the Paper are:
1.) Game environment build Module
2.) Game control Module
3.) Game state Module
4.) AI Module
5.) Outcome judge Module

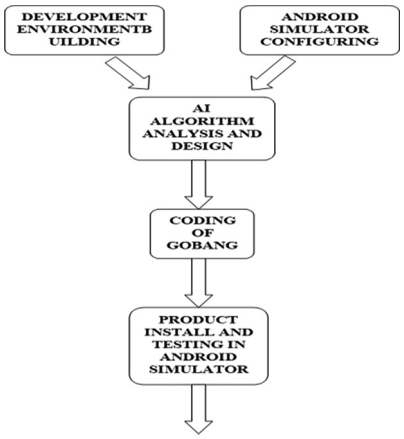

Fig. 2.14. Preliminary plan.

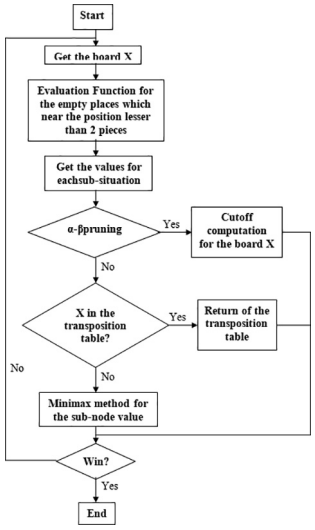

Fig. 2.15. Gobang algorithm flowchart.

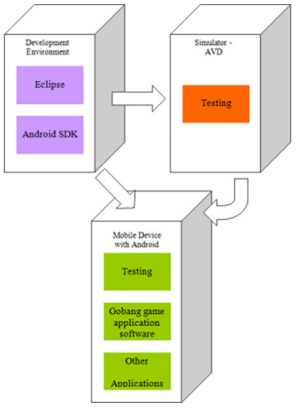

Fig. 3.1. Architectural design diagram.

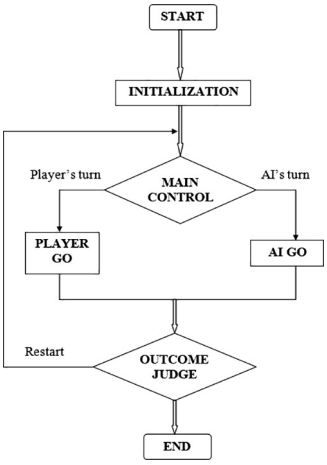

Fig. 3.2. Gobang game flowchart.

2.5.1. Game environment build Module

Game environment build Module includes three sub-modules, there are draw board, draw stone and play chess, as shown in Fig. 3.3.

The function of each module:

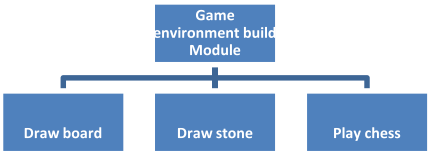

Fig. 3.3. Game environment build module.

Draw board: draw and build a board of game.
Draw stone: draw stone, or chess pieces of game.
Play chess: realize playing chess; put the chess pieces on the board. This function is on touch event in Android.

2.5.1.1. Game control Module. Game control Module includes three sub-modules, there are game start, game restart and quit game, as shown in Fig. 3.4.

The function of each module:
Game start: start a new Gobang game, initialization of game.
Game restart: restart a Gobang game when one game end.
Quit game: quit a Gobang game.

2.5.1.2. GameState Module. Game state Module includes two sub-modules, there are board state and whose turn, as shown in Fig. 3.5.

The function of each module:
Board state: record the board from, position value and position coordinate.
Whose turn: record the current turn of players.

2.5.1.3. AI Module. AI Module includes three sub-modules, there are AI player, search method and evaluation function, as shown in Fig. 3.6.

The function of each module:
AI player: the record of AI player for playing game.
Search method: the AI algorithm.
Evaluation function: the evaluation functions for the chess form, set values.

2.5.1.4. Outcome judge Module. Outcome judge Module includes two sub-modules, there are outcome judge and check full, as shown in Fig. 3.7.

The function of each module:
Outcome judge: judge the outcome, which side is winner and which is loser.
Check full: check whether the board is full. If the board is full, the game is draw.

3. Implementation

3.1. Implementation

At first, must create a project. Startup Eclipse, click "File"–"New"–"Android Project", as shown in Fig. 4.1.

Then begin coding for the Android project in JAVA language with Eclipse, as shown in Fig. 4.2 (Fig. 4.3).

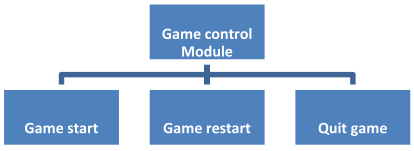

Fig. 3.4. Game control module.

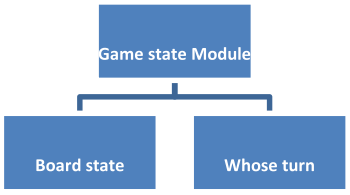

Fig. 3.5. Game state module.

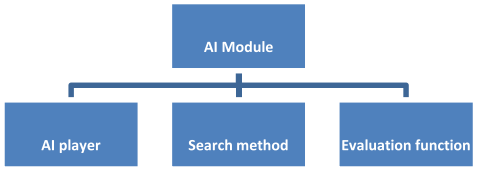

Fig. 3.6. AI module.

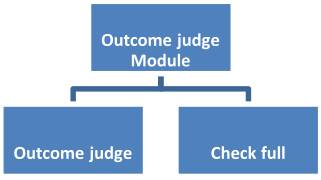

Fig. 3.7. Outcome judge module.

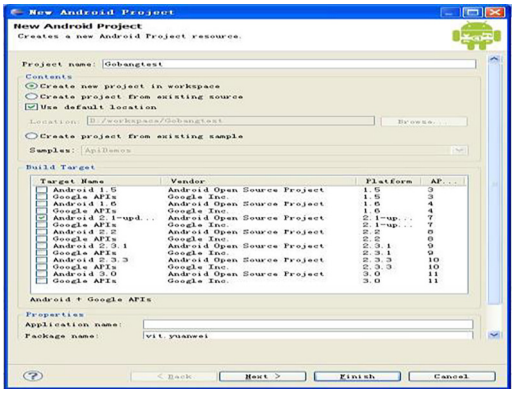

Fig. 4.1. New Android project.

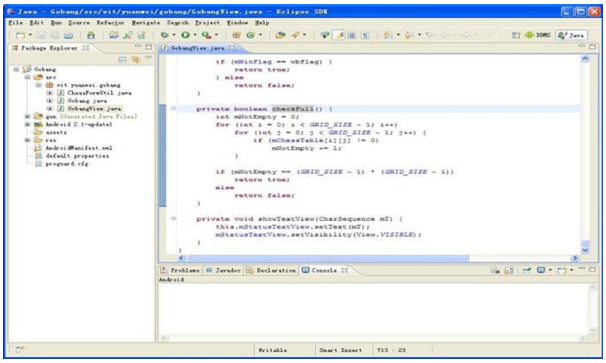

Fig. 4.2. Coding state.

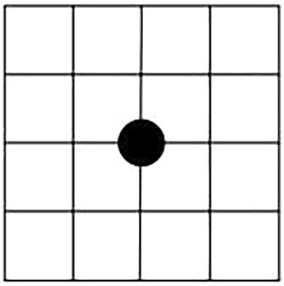

Fig. 4.3. Area of search.

3.2. Modules implementation

1.) Game environment build Module
2.) Game control Module
3.) Game state Module
4.) AI Module
5.) Outcome judge Module

3.2.1. Game environment build Module

First is the main program:

public class Gobang extends Activity {

// main class, inherited from Activity, implement the onCreate function

Draw board: draw and build a board of game.

public void onDraw(Canvas canvas)

protected void onSizeChanged(int w, int h, int oldw, int oldh)

//normal Bitmap drawing

// Bitmap + Canvas + Drawable

Draw stone: draw stone (or chess pieces) of game.

public void onDraw(Canvas canvas)

//through the picture to draw the stone

Play chess: realize playing chess, put the chess pieces on the board. This function is on touch event in Android.

public booleanonTouchEvent(MotionEvent event)

private void putChess(int ×, int y, int blackwhite) {

mChessTable[x][y] = blackwhite;

}

3.2.1.1. Game control Module. Game start: start a new Gobang game, initialization of game.

```
public void init() {//initialization of black and whiter's
    BitmapmGameState = 1;//set the game to the start
    statewhoTurn = WHITE;mWinFlag = 0;//clear words of
    winner or loserfor (int i = 0; i < CHESS_GRID; i++) {//clear all
    tablesfor (int j = 0; j < CHESS_GRID; j++) {mChessTable[i][j]
    = 0;for (int k = 0; k < CHECK_DIR; k++) {computerTable[i][j]
    [k] = 0;playerTable[i][j][k] = 0;}}}}
```

Game restart: restart a Gobang game when one game is end.

public booleanonKeyDown (int keyCode, KeyEvent msg)

//after the game, press the CENTER button to continue

3.2.1.2. GameState module. Board state and game state: record the board from, position value and position coordinate.

The class of the current chess piece point and score:

```
class ChessPoint {int x;int y;int score;}
```

Game state:

private int mGameState = GAMESTATE_RUN;

//game state, 0 = don't start, 1 = playing game, 2 = pause of game, 3 = game end.

private TextViewmStatusTextView;

//the words display about game state.

Whose turn: record the current turn of players.

private int whoTurn = WHITE;

//whiter's turn = 2, black's turn = 1.

3.2.1.3. AI module. Prepare an array to express the current board, the other two arrays were prepared to store the computer player's and human player's values and coordinates of empty points (chess form array). Each empty point has four values of four directions, which are horizontal, vertical, left oblique and right oblique.

```
private int[][] mChessTable = new int[CHESS_GRID][CHESS_G
    RID];//the gridprivate int[][][] computerTable = new int[C
    HESS_GRID][CHESS_GRID][CHECK_DIR];//computer table of
    chess gridprivate int[][][] playerTable = new int[CHESS_GR
    ID][CHESS_GRID][CHECK_DIR];//player table of chess grid
```

Each point can be put, that the values judge of four directions, to take the state in each direction around the current point moves five points on each side, and then analyze whether they constitute Connect Five, Live Four, Live Three, and so on. Each form set adifferent score.

If satisfy Connect Five, then no need Live Four, if satisfy Live Four, then no need Live Three, if satisfy Live Three, then no need Rush Four, and so on.

For example, Connect Five (Five in a Row) analysis:

```
public booleananalyzeConnectFive(int[] tmpChess, int isWho)
    {int count = 0;for (int i = 0; i < HALF_LEN; i++) {if (tmpChess
    [HALF_LEN - (i + 1)] == isWho) {count++;} else {break;}}for
    (int i = 0; i < HALF_LEN; i++) {if (tmpChess[HALF_LEN + i] ==
    isWho) {count++;} else {break;}}if (count == 4) {return
    true;}return false;}
```

For each form set a score as the evaluation function:

public static final int CONNECT_FIVE = 85;
public static final int LIVE_FOUR = 40;
public static final int LIVE_THREE = 15;
public static final int RUSH_FOUR = 6;
public static final int LIVE_TWO = 4;
public static final int SLEEP_THREE = 2;
public static final int SLEEP_TWO = 1;

Compare every element scores of chess form array between computer player and human player. Select the five largest put into a descending order array.

In this step:

private ChessPointfindBestPoint()

Calling the function:

insertBetterChessPoint(point);

And return:

analyzeBetterChess();

//private ChessPointanalyzeBetterChess()

Process the array in descending order. If the score of the first element is equal of greater than the score which is the condition of win, then return it direct. If less than the score which is the condition of win, then continue to process.

private int analyzeDir(int[] tmpChess, int isWho)

//analyze one direction

private void analyzeChessMater(int[][][] materChess, int isWho, int sx, int sy, int ex, int ey)

//analyze the chess form which is appointed

private ChessPointanalyzeBetterChess()

private void insertBetterChessPoint(ChessPoint cp)

//analyze the better chess point.

Each element of the array in descending order, assuming that element has put, determine the consequences, select the best one and return its value. This value is the point of computer player's next step.

When determine the consequences of each element, in fact only need to deal with the scope of its board places which it has an effect. As shown inFigure 4.3. It's unnecessary to search the whole board.

3.2.1.4. Outcome judge Module. Outcome judge: judge the outcome, check whether it's winner.

private booleancheckWin(int wbflag)

//check five connect of four directions, horizontal, vertical, left oblique, right oblique

For each direction:

```
for (int i = 0; i < GRID_SIZE – 1; i++)for (int j = 0; j < GRID_SIZE
  – 1; j++) {if (((i + 4) < (GRID_SIZE – 1))//check five connect
  of horizontal&& (mChessTable[i][j] == wbflag)&&
  (mChessTable[i + 1][j] == wbflag)&& (mChessTable[i + 2][j]
  == wbflag)&& (mChessTable[i + 3][j] == wbflag)&&
  (mChessTable[i + 4][j] == wbflag)) {mWinFlag = wbflag;}}if
  (mWinFlag == wbflag) {return true;} elsereturn false;
```

//and so on...

Someone win, show the words:

private void showTextView(CharSequencemT)

Check full: check whether the board is full. If the board is full, the game is draw.

private booleancheckFull()

3.3. Result discussion with snapshots

There are some snapshots of the intelligent Gobang game executed in Android simulator in Windows XP.

Fig. 5.1 shows the interface displayed when the application is initially executed.

Fig. 5.2 shows the stage of Human player (White) goes first, the computer player reply accordingly. Fig. 5.3 shows the status text

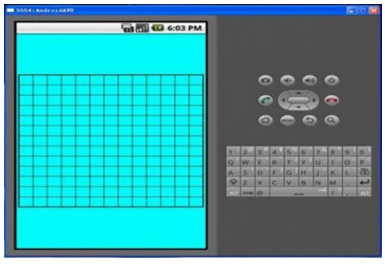

Fig. 5.1. Initially executed.

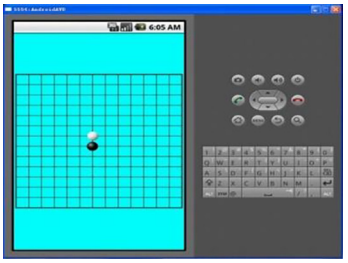

Fig. 5.2. Game start.

Fig. 5.3. Win state.

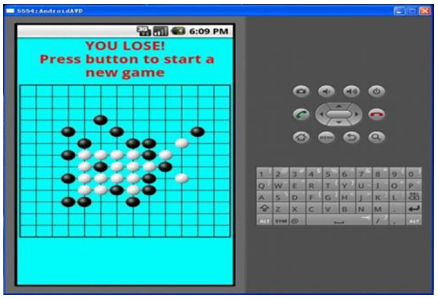

Fig. 5.4. Lose state.

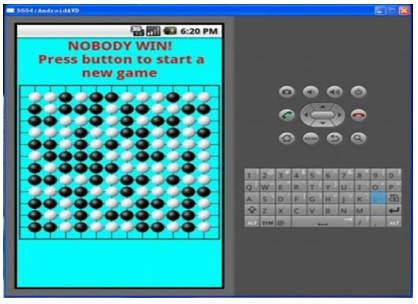

Fig. 5.5. Draw state.

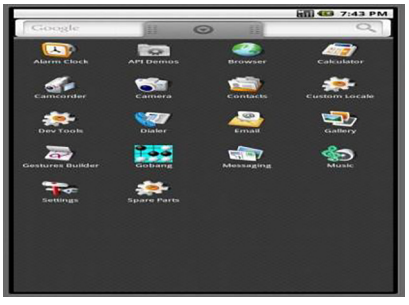

Fig. 5.6. Gobang application.

shows up when human player wins, press the Fire Key to start a new game. Fig. 5.4 shows the status text shows up when human player loses (computer player wins), press the Fire Key to start a new game.

Fig. 5.5 shows the status text shows up when the chess board is full, then equal, nobody wins, press the Fire Key to start a new game. Fig. 5.6 shows the Gobang icon in the applications, it is one of the Android applications. We can use the Gobang application when click it. The results of this work satisfy required functions, are the expected results. Because of there isn't real Android mobile phone, so that just in the simulator to realize.

4. Conclusions

The rapid development of the smart-phone, more and more mobile devices come into the work and life of people. This paper realizes an interactive entertaining Gobang game in Android operating system. It is a application software for mobile devices, using the artificial intelligence algorithm to analyze and design the intelligent Gobang system. Artificial intelligence is a very promising subject, it has become the development direction of new products and new equipment. With the new algorithms and theoretical research, artificial intelligence will be in people's lives, play an important role. Via coding of the Gobang game to make myself a deeper understanding of JAVA language, more understand and familiar with the Eclipse and Android development tools.

For this paper, more work need to be done for the evaluation function and search algorithm. In Gobang game, the forbidden rule of international rules can be applied to the game in the future.

With the continuous improvement of the level of hardware, the mobile devices will have a better CPU. More and more artificial intelligence search algorithm can be applied to mobile devices. It could be improved by testing more games and examine carefully when the program does not reply correctly to the human's play.

CRediT authorship contribution statement

L. Venkateswara Reddy: Conceptualization, Methodology, Software. **S. Saravana Kumar:** Data curation, Supervision, Software, Validation. **S. Sugumaran:** Visualization. **K. Lavanya:** Investigation, Writing - original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal system relationships that could have appeared to influence the work reported in this paper.

References

[1] Androidoperating system, http://en.wikipedia.org/wiki/Android_(operating_system), April 17, 2011.
[2] S. Russell, P. Norvig, Artificial Intelligence - A Modern Approach, 2nd ed., Pearson Education/Prentice Hall of India, 2004.
[3] L.V. Allis, H.J. van den Herik and M.P.H. Huntjens, Go-Moku and Threat-Space Search, March 7, 1994.
[4] W.u. Yafeng, S. Yina, Android Core Technology and Development, Publishing House of Electronics Industry, Beijing, 2010.
[5] C. Jie, Gobang algorithm to research M.S. thesis, University of Science and Technology of China, 2003.
[6] W. Changfei, C. Qiang, L. Haisheng, Design and implementation of intelligent gobang playgame, J. Syst. Simul. 21 (6) (2009) 1051–1054.