CSCI3180 – Principles of Programming Languages – Spring 2012

Course Project Phase 3 — Extended Jungle Chess

Phase 3 and Bonus Deadline: April 8, 2012 (Sunday) 23:59

1 Introduction

Jungle Chess or Dou Shou Qi is a traditional Chinese board game with a 9x7 game board. It is a 2-player game where each player has eight chess pieces, each representing a different animal. The goal is to either get into the den of the opponent's side or capture all of enemy's pieces.

In Phase 3, you are required to extend the Jungle Chess you have implemented in Phase 2, both in C and Smalltalk. The extended Jungle Chess, as shown in Figure 1, is a 4-player game with larger game board and different components added. **Please note that otherwise specified, every requirement in Phase 2 are still valid and you should follow strictly in developing the extended Jungle Chess.** The details of the extended Jungle Chess is covered in later sections. In addition, you may optionally choose to implement the bonus part of the extended Jungle Chess to obtain extra marks.

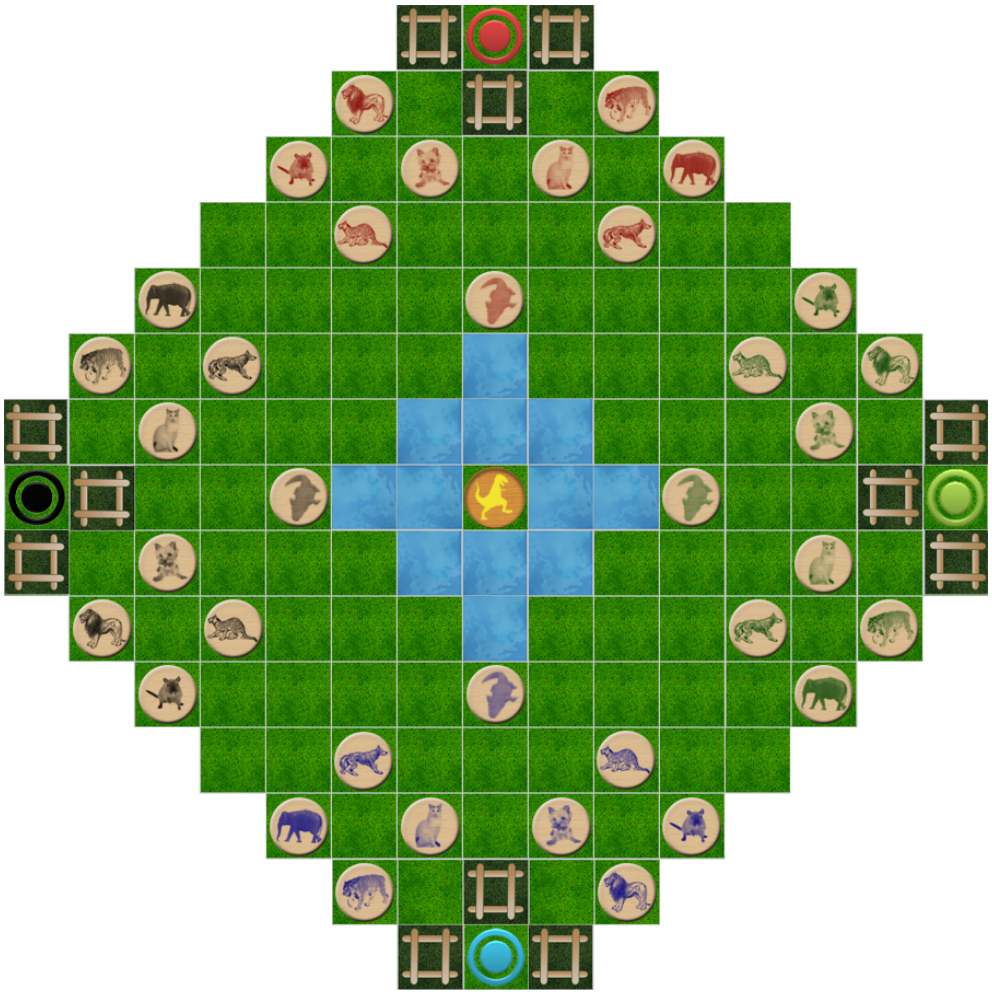

Figure 1: The extended Jungle Chess

2 Game Details

Some game details are changed or added on top of those defined in Section 2 of the Phase 1 and 2 specification. Please note that except the changes we mentioned in this section, other rules or details remain the same as before. Here is the outline of the major extensions:

- The game board is enlarged and becomes diamond-shaped.
- Number of players is extended to 4.
- Cat evolves to SuperCat.
- Each player has one more chess piece: Crocodile.
- A computer-controlled Monster is added.

2.1 Players

The number of players is increased to four: **Player Blue**, **Player Black**, **Player Red**, and **Player Green**. Each player has its own chess pieces and den in the corresponding color, e.g., Player Green with green chess pieces and green den. In addition, Player Blue, Player Black, Player Red, and Player Green occupy the bottom, left, top, and right parts of the chess board respectively, as shown in Figure 1.

The four players play in turn in the following order: **Player Blue->Player Black->Player Red->Player Green**. If a player loses, this player's turn is skipped. For example, if Player Black loses, the order of turns will become Player Blue->Player Red->Player Green.

2.2 Goal

The goal is to survive! The last surviving player **wins** the game. A player **loses** if 1) den is occupied by an opponent's chess piece, or 2) all chess pieces are captured. **When a player loses, all the chess pieces of this player will disappear from the chess board, and the den and traps of this player will become normal grounds**. For example, if Player Red, Player Black and Player Blue lose, Player Green is the winner.

2.3 Chess Board

Besides the number of players, the chess board is also expanded. As shown in Figure 1, it is a diamond-shaped chess board with 15 rows and 15 columns of squares. As before, there are four ground types: **ground, river, trap** and **den**.

The river becomes diamond-shaped and is located in the center of the chess board. The center square of the chess board is a ground square. We still have three traps for each players and they surround the den of the player.

2.4 Chess Pieces

Each player has nine chess pieces instead of eight in the extended Jungle Chess. There is one added chess piece: **Crocodile** (Figure 2a). Besides, the **Cat** has evolved to become the **Super-Cat** (Figure 2b). Other animals' appearances and behaviors remain unchanged.

Apart from fighting against one another, the players have to be aware of the **Monster** (Figure 2c). Monster is a newly added chess piece which does not belong to any player. You can regard it as a chess piece controlled by a fifth player: computer player.

Before the game starts, all chess pieces of each player and the Monster are placed on their **predefined positions**, as shown in Figure 1. The **ranking** in the extended Jungle Chess among the nine animals and the Monster is defined as follows (from strongest to weakest):

`Monster > Crocodile > Elephant > Lion > Tiger > Jaguar > Wolf > Dog > SuperCat > Rat`
and `Rat > Elephant`

(a) Crocodile (b) SuperCat (c) Monster

Figure 2: New chess pieces in the extended Jungle Chess

2.5 Game Rules

The four players take turns to move their own chess pieces and they must move one chess piece in each turn. Besides, after each of the four players makes a move, the Monster will move once. The sequence of moving is:

`Player Blue->Player Black->Player Red->Player Green>Monster`

The game ends when one of the players wins. Following is the capturing and movement of the newly added components in the extended Jungle Chess. Any component which is not listed below is governed by the rules listed in Section 2 of the Phase 1 and 2 specification.

Capturing

Animals can capture any opponents' animals with the same or lower ranking.

- The Monster is the strongest creature in the extended Jungle Chess. **The Monster can capture any animal of any player but no animal can capture the Monster in any situation.** For example, even if the Monster is in a trap of a player, no animals can capture the Monster.
- **Crocodile is also very strong and it can capture all other animals (except the Monster)**. For example, a Crocodile can capture another Crocodile and Elephant. However, the moveable area of Crocodile is very limited.
- **SuperCat's capturing ability is the same as before**, i.e. being able to capture only SuperCat and Rat. However, its movement characteristic is changed.
- **When a chess piece moves into the trap of an opponent's side, it becomes super weak and any other opponents' animals can capture the piece in the trap. For example, if the Lion of Player Blue enters the trap of Player Red, the chess pieces of all other three players (Player Red, Player Black and Player Green) can capture the Blue Lion in the trap.**
- **When a chess piece moves into its own trap, it becomes super strong and no chess piece of any opponent can capture the piece in the trap.**

Movement

Same as before, each chess piece can be moved one step **vertically or horizontally**. No chess pieces can be moved into their own den. Besides, there are some special rules for the newly added chess pieces:

- **The Monster can move one step vertically, horizontally or diagonally in each turn.** That means, it can move to any one of the eight surrounding squares. Besides, it can move to any type of squares, i.e. Ground, River, Trap or Den. **The Monster steps**

on any type of squares as normal grounds. For example, even if the Monster steps on Player Red's den, Player Red will not lose. Also, nothing will happen to the Monster and other chess pieces if the Monster walks into the traps of players. In addition, the Monster moves semi-intelligently. **If there are animals in the eight squares surrounding the Monster, it must randomly capture one animal that surrounds it. Otherwise, it makes a random move (to one of the eight surrounding squares).** Since the Monster is controlled by the computer, it should move automatically after each player has moved in each round.

- **Crocodile can only move in the river and the ground squares immediately next to the river (Figure 3). It can move one step vertically or horizontally.** It can capture any opponents' animals within its moveable area (except the Monster).
- The movement of SuperCat is the same as Cat, except that SuperCat can move also diagonally. That means, **SuperCat can move one step vertically, horizontally or diagonally.**
- Lions and Tigers still have the ability to jump over the river as defined in Phase 2. **The only difference in Phase 3 is that they cannot jump over the river if there is any obstacle (possibly Rat, Crocodile or Monster) on their intended path for jumping.**

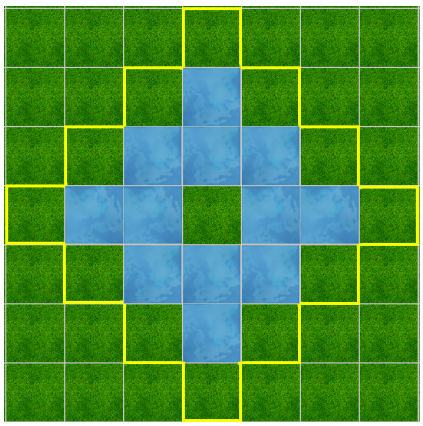

Figure 3: Crocodile's moveable areas: river and grounds beside river (the area with yellow border)

3 C Implementation

You have to extend your C program in Phase 2 in order to include the extra requirements in the extension in Phase 3. The Input and Output methods are the same as specified in Section 4.1 of the Phase 1 and 2 specification. You have to use Ncurses to make the text user interface.

Although the I/O methods are not changed, the game board of the Jungle Chess is extended, as shown in Figure 4. Please follow the game board and the **predefined position of the chess pieces** strictly according to Figure 4 in your C implementation. **The initial selected square is the Crocodile of Player Blue, as shown in Figure 4**. Besides, there are some newly added or modified chess pieces. We define these chess pieces with the first letters of their names, as shown in Table 1. Please note that Cat becomes SuperCat so it has name 'S' instead of 'C'.

4 Smalltalk Implementation

Similar to the C implementation, the Input and Output methods remain unchanged in the Smalltalk implementation. You have to draw the game board and the initial chess pieces by strictly following

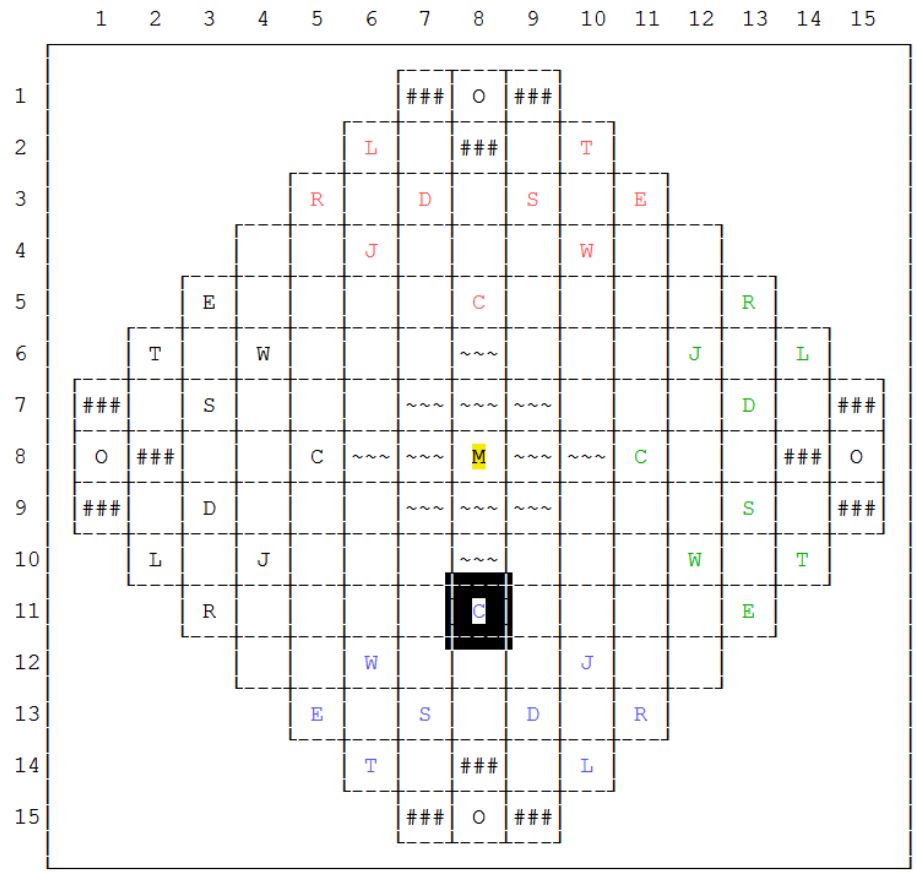

Figure 4: C implementation of the extended Jungle Chess

Figure 1.

5 OO Design and Report

Since you must change your OO design in order to incorporate the extension in Phase 3, you have to update your UML class diagram of Phase 2 and submit the class diagram for the extended Jungle Chess. Same as before, you have to write up a **report** to answer the following questions

1. Compare the advantages and disadvantages in extending the Jungle Chess in Phase 3 using C and Smalltalk. You may discuss it with respect to the five specific extension tasks.

2. Briefly explain what changes you make on your OO design of Phase 2 to incorporate the extension in Phase 3 and why.

6 Bonus

You are free to design and expand your the extended Jungle Chess implemented in Phase 3. **Please note that the bonus part is only needed for the Smalltalk implementation.** You do not have to implement your bonus part in C. You can get **at most 20 extra points** on top of Phase 3. **If you decide to implement the bonus part, please do it on a new copy of your Phase 3 implementation so that you can keep your Phase 3 implementation clean for submission.** Besides, you do not need to draw the OO design for your bonus part. The bonus

Animal	Chess Piece
Monster	M
Crocodile	C
Elephant	E
Lion	L
Tiger	T
Jaguar	J
Wolf	W
Dog	D
SuperCat	S
Rat	R

Table 1: Chess Pieces Representation of the extended Jungle Chess in C implementation

marks will be based on creativity, rationality, difficulties and completeness of the added functionalities. Here are some suggested features:

- **Undo Function:** Before the next player makes his/her move, the previous player can choose to undo his/her previous move. For example, if Player Red just made a move and the control is passed to the next player, Player Green, Player Red can click the undo button to undo his/her move before Player Green makes his/her move. If Player Red's move is undone, the control is passed back to Player Red again. At this point, before Player Red makes again his/her move, Player Black can undo his/her previous move also. **That means, the undoing can continue until the initial state of the game is reached. One point to note is that when Player Green clicks the undo button, the previous move of the Monster and that of Player Green will both be undone sequentially.**
- **Revival:** There are some Revival Fruits randomly spread on the grounds. When a chess piece moves to a square with a piece of Revival Fruit, it eats the fruit and one of its captured comrades can revive at a random, empty and valid square.
- **Move Countdown:** In each player's turn, there is a time limit (e.g. 10 seconds) for the player to make the move. If time out, one chess piece of this player is chosen randomly for a random but valid move. You should clearly show the countdown clock in the GUI.
- **Computer Player:** Create computer players to play the game intelligently. Users can choose the number of computers players before the game start. The computer players should make rational moves but not just random moves.

You should provide necessary prompts or messages to show the changes made by the functionalities. For example, messages showing which player's move is undone or which chess piece is revived at which position. Please note that user-friendliness is also counted. For your reference, each of the complete implementation of the first three functions can get at most 10 points and the fourth one (Computer Player) can obtain at most 20 points. For example, if you implement the first two (Undo Function and Revival), you can get at most 20 (10+10) points. If you design your own function, we will mark it relatively according to the four criteria mentioned above.

7 Development Environment

Please test your programs under the following environments before you submit it. You take your own risk if you just test your program within other development environments.

7.1 C Implementation

Make sure you can compile and run your C program without any problem with the gcc compiler on CSE **linux** machines. You should be able to print the borders of the game board with the use of

Ncurses. One point to note is about displaying the extended characters (for example, the corners of the game board). We suggest you to remote access you linux account using PuTTY which can display the extended characters without any problems.

7.2 Smalltalk Implementation

You are required to use **VisualWorks version 7.8** to develop your Smalltalk implementation. The download link for VisualWorks is provided in the course homepage. You should create a new package to do your project and export the whole package for submission. You can export your package in the System Browser in VisualWorks by right-clicking your package, selecting "File Out -> Package" and saving your package as "JungleChess.st". In addition, you have to export your workspace that is used to start up your program. In your workspace, click "File -> Save As" and save it as "JungleChessWS.ws". If you decided to implement the bonus part, please create a new package for it and keep a clean copy for your extended Jungle Chess. Please name the files of the package and workspace of the bonus part by "BonusJungleChess.st" and "BonusJungleChessWS.ws" respectively.

8 Other Requirements

You are required to follow the requirements below throughout the project.

1. **Division of Labour**
   You should divide the work in a way such that both of you participate in all phases, including the OO design and the implementation of both C and Smalltalk. In this way, both of you can learn and participate in different phases of constructing a real application and be able to tell the differences and difficulties in developing the game with the two languages.

2. **Error Handling**
   The programs should handle possible errors gracefully by printing meaningful error messages to standard output. For example, failure to open a non-existing file or input with wrong format.

3. **Good Programming Style**
   A good programming style not only improves your grade but also helps you a lot when debugging. Poor programming style will receive marks deduction. Construct your program with good readability and modularity. Provide enough documentation in your codes by commenting your codes properly but not redundantly. Divide up your programs into subroutines instead of clogging the main program. The main section of your program should only handle the basic file manipulation such as file opening and closing, and subroutine calling. The main purpose of programming is not just to make it right but also make it good.

4. **Other Notes**
   You are **NOT** allowed to implement your program in another language (e.g. Assembly/C++/Java) and then initiate system calls or external library calls in C and Smalltalk. Your source codes will be compiled and PERUSED, and the object code tested!

   In the C implementation, DO NOT implement your programs in multiple source files. Although C does allow you to build a project with subroutines scattered among multiple source files, you should only submit one source file for the C language.

   NO PLAGIARISM! You are free to design your own algorithm and code your own implementation, but you should not "steal" codes from your classmates or any other people. If you use an algorithm or code snippet that is publicly available or use codes from your classmates or friends, be sure to cite it in the comments of your program. Failure to comply will be considered as plagiarism.

9 Submission Guidelines

In Phase 1, you are required to submit your programs for the exercise, the object-oriented design, the report for the OO design and the receipt from VeriGuide. You are required to prepare your UML class diagrams and your report in PDF files. **In Phases 2 and 3, you are required to submit your code and report.** In addition, if you use your design with any modifications, you are required to submit your modified design also. Please read the guidelines CAREFULLY. If you fail to meet the deadline because of submission problem on your side, marks will still be deducted. So please start your work early! **Only one submission is needed for each group.**

1. In the following, **SUPPOSE**

   your group number is 1,
   your name is *Chan Tai Man*,
   your student ID is *1155234567*,
   your username is *tmchan*, and
   your email address is *tmchan@cse.cuhk.edu.hk*.

2. In each phase, the report should be submitted to VeriGuide, which will generate a submission receipt. The report and receipt should be submitted together with your source files in the same ZIP archive.

3. The OO design diagram should have the filename "design.pdf" and the report with "report.pdf". The VeriGuide receipt of report should have the filename "receipt.pdf". All file naming should be followed strictly and without the quotes.

4. In Phase 1, the C source should have the filename "VigenereCipher.c" and the Smalltalk source should have the filename "VigenereCipher.st". In Phase 2 and Phase 3, the C source should have the filename "JungleChess.c" and the Smalltalk source with filename "JungleChess.st" for your package and "JungleChessWS.ws" for the workspace. If you have implemented the bonus part in Phase 3, you should name your source files with "BonusJungleChess.st" and "BonusJungleChessWS.ws".

5. For Phase 1, `tar` your source files to `group[num].tar` by

   ```
   tar cvf group01.tar VigenereCipher.c VigenereCipher.st \
   design.pdf report.pdf receipt.pdf
   ```

   For Phase 2, if you have not changed your design, `tar` your source files to `group[num].tar` by

   ```
   tar cvf group01.tar JungleChess.c JungleChess.st JungleChessWS.ws \
   report.pdf receipt.pdf
   ```

   Otherwise, `tar` your source files to `group[num].tar` by

   ```
   tar cvf group01.tar JungleChess.c JungleChess.st JungleChessWS.ws \
   design.pdf report.pdf receipt.pdf
   ```

   For Phase 3, if you have done the bonus part, `tar` your source files to `group[num].tar` by

   ```
   tar cvf group01.tar JungleChess.c JungleChess.st JungleChessWS.ws \
   BonusJungleChess.st BonusJungleChessWS.ws design.pdf report.pdf receipt.pdf
   ```

   Otherwise, `tar` your source files to `group[num].tar` by

   ```
   tar cvf group01.tar JungleChess.c JungleChess.st JungleChessWS.ws \
   design.pdf report.pdf receipt.pdf
   ```

6. `Gzip` the `tarred` file to `group[num].tar.gz` by

   ```
   gzip group01.tar
   ```

7. `Uuencode` the `gzipped` file and send it to the course account with the email title "Project Group *your Group Number*" by

```
uuencode group01.tar.gz group01.tar.gz \
| mailx -s "Project Group01" csci3180@cse.cuhk.edu.hk
```

8. Please submit your project using your Unix accounts.

9. An acknowledgement email will be sent to you if your project is received. **DO NOT** delete or modify the acknowledgement email. You should contact your TAs for help if you do not receive the acknowledgement email within 5 minutes after your submission. **DO NOT** re-submit just because you do not receive the acknowledgement email.

10. You can check your submission status at

    `http://www.cse.cuhk.edu.hk/csci3180/submit/project.html`.

11. You can re-submit your project, but we will only grade the latest submission.

12. Enjoy your work :>