The Jungle Game

A command-line-based program

Software Requirement Specification

Group 32
LI Muyuan: 20084337d
XU Jialu: 20075114d
YAN Qiuwen: 20058307d
YIN Wei: 20074787d

Contents

1. Preface

The Jungle Game is a two-person puzzle game that originated in the 1960s. The original version of the jungle game only has simple text and grids. Up to now, it can be played online. Our team set up a new version of it with various new features added.

2. Introduction

2.1 Purpose

The software aims to provide users with offline coliseum game services, allowing users to experience the fun of coliseum anytime and anywhere.

2.2 Project scope

The jungle game is a game beneficial to brain development. Players must brainstorm to win, which helps users to strengthen the speed of brain operation. In addition, this is a two-player game, so there is also a chance to recognize different people during the game to expand the user's social circle. Moreover, the rules of the game are based on the world rules of strong survival and weak elimination of animals, so users can have a preliminary understanding of the biological chain. Therefore, this will be a game conducive to the brain and social development of users.

2.3 Intended audience

The target audience of this software is primary and junior high school students who are interested in Jungle Game. Since the jungle game is a childhood game for

people in the 1970s and 1980s, people with nostalgia for the jungle game are also potential targets.

2.4 Intended use

In the game of Coliseum, users follow the following rules to play chess:
Rule 1: Each side has eight red and blue pieces and its own nest. The user's game goal is to eat all the other side's pieces or let one of his pieces walk to the other side's nest.
Rule 2: The special terrain in the map has traps and rivers. Any enemy piece entering our trap will lose its original ranking and can be eaten by any of our pieces. In principle, no pieces can enter the river area.
Rule 3: Users need to judge the behavior of "eating" according to the ranking of animals. In principle, animals with a high ranking can eat animals with a ranking smaller than or equal to their own.
(Ranking: Elephant > Lion > Tiger > Leopard > Wolf > Dog > Cat > Rat)
Rule 4: In principle, the user can only move one of his pieces one space horizontally or vertically in each action.
Rule 5: Special circumstances:

1. Although the rat is the lowest-ranked animal, it can directly eat the highest-ranked elephant
2. Only the rat can enter the river. But it may not capture the elephant or another rat on land directly from a water square.
3. Rats can eat each other's rats in the river.
4. Lions and tigers can jump rivers to land on the other side and eat animals on the other side in the process.
5. But if there are rats in the river (either from the enemy or from us), neither the lion nor the tiger will be able to jump the river.

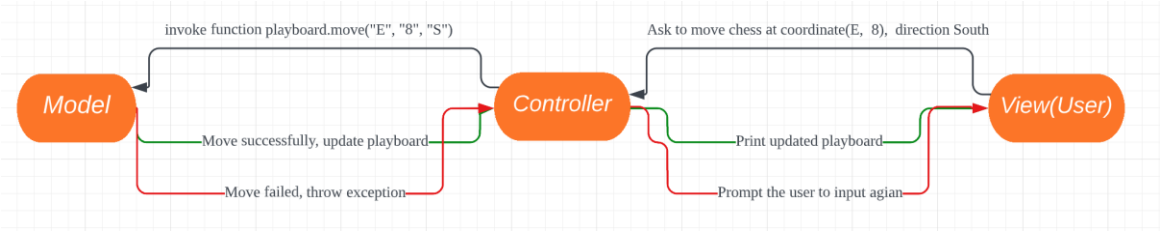
3. User requirements

3.1 The game software should be simple, clear and easy to operate.
3.2 The appearance should also be rationalized. The server program interface should be simple to operate and easy to manage.
3.3 The main interface of the application should provide the user with an introduction to the rules of the game.

3.4 After the start of the game, both users can take turns playing chess according to the rules.
3.5 In a match, the Colosseum software shall have a comprehensive error step to guide the user to play properly.
3.6 In the game, it is also necessary to meet the needs of users to pause and surrender, and the number of pauses will be set at most twice and each pause time is less than 30 seconds to ensure the game experience of both sides.
3.7 After finishing the game, users should be able to view their match history on the main screen.
3.8 This game product has good user friendliness and good stability, and the game has good portability and should be able to develop a mobile terminal in the future.

4. System architecture

The system architecture adopted in this game is Model-View-Controller (MVC) pattern.



A case to illustrate the system's MVC architecture

4.1 View

The View is the part of the application that represents the data. It is created by the data collected from the model data and requests the model to give information to present the output to the user. To be more specific, the View transfers the user message to the Controller and transfers the information back from Controller to the user.

4.2 Controller

The general structure is considered to develop under the basic function of Controller. The Controller should be able to receive the message from the user and transform the message into instructions to continue to control the game. There are several functions contained in the Controller, such as playNewGame(), pauseGame(), resumeGame() and restartGame().

4.3 Model

The Model component stores the data and its related logic. It represents data transferred from Controller and other logic. It responds to the request from the View and responds to the instructions from the Controller. It is the lowest pattern level responsible for maintaining data.

5. System Requirements specification

5.1 Functional:

1. The system must display the playboard correctly and clearly.
2. The game implementation must strictly adhere to the game rules.
3. The system must be able to handle users' misinput properly and return corresponding feedback to advise the correct input.
4. The system's structure must be layered well-arranged. Which methods and property can be used by the controller must be clearly defined.

5.2 Non-functional:

1. Security: The software should make sure the identifier of each method and properties so that the data cannot be casually revised.
2. Portability: The program should be able to be executed on both Windows and Mac operating systems.
3. Reliability and availability: The program should decrease the possibility of errors. The program should be easily searched by users.
4. Maintainability: The program should be convenient for revision and scale-up.
5. Performance: The code should be concise and consistent.
6. Compatibility: see appendices.

6. Appendices

6.1 Minimal configurations(sample):

1. Operating system: Windows 7
2. Processor: Intel Pentium D or AMD Athlon 64 (K8) 2.6 GHz
3. Memory: 1 GB RAM
4. Graphics cards: Intel HD Graphics or AMD (Former) Radeon HD Graphics with OpenGL 2.1
5. Storage space: 100 MB free space is required

6.2 Optimal configurations(sample):

1. Operating system: Windows 10 Home 64-bit
2. Processor: Intel Core ™ I7 2700 k | AMD Ryzen 7 2700x
3. Memory: 16 GB RAM
4. Video: Nvidia GeForce ™ GTX (1.5 GB) | AMD Radeon 580 ™ RX 560 (4 GB)
5. DirectX version: 11
6. Network: Broadband Internet connection
7. Storage space: 4 GB free space is required

6.3 Game glossary

Board	The board consists of seven columns and nine rows of square spaces.
Capture	Each piece can capture any enemy piece by moving to their position. A piece can capture any enemy piece with the same or lower ranking. A rat may capture an elephant but not from a water square. A piece may capture any enemy pieces in the trap square regardless of the rank.
Den	The dens are the center of the boundary rows of the board with the label in Chinese characters 獸穴. There is only one den for each side, while each den only contains one square space.
Land	The lands are the remaining square space on the board except for dens, traps and rivers without any special labels. They are located at the edge of the board and down the middle between the rivers. Each land space is an ordinary square space.
Movement	Players alternate moves. A player must move during their turn. Each piece moves one square horizontally or vertically (not diagonally). A piece may not move to its den. The rat is the only type of animal that goes into rivers, while the others must stay on the lands. The lion and tiger can jump over a river by moving horizontally and vertically.
Objective	The goal of the game is to move one piece to the den of the opposite side

	of the board or capture all the pieces of the enemy.
Piece	There are eight pieces representing different animals on each side.
Rank	The rank defines the order of capture. Eight animals on each side represent different animals with different ranks. Higher ranking pieces can capture all pieces of the same or lower ranking. The ranking from high to low is shown as: Elephant, Lion, Tiger, Leopard, Wolf, Dog, Cat and rat. There is an exception where a rat may capture an elephant while an elephant may not capture a rat.
River	The rivers are also known as water areas in the center of the board with the label in Chinese characters 小河. There are two rivers in the board in total, while each one comprises six square spaces in a 2*3 rectangle square.
Square Space	A square space is where a piece stays and moves. Each square can only contain a piece. There are several kinds of square spaces, such as dens, traps, rivers (water areas) and lands.
Trap	The traps are located to each side and in front of and next to the dens with the label in Chinese characters 陷阱. There are three traps for each side, while each trap only contains one square space.