APPLICATION DOCUMENTATION

Jan 14, 2018

ABSTRACT

The documentation includes both the user guide and the coding documentation for the maze game Android application

Chen Wang

School of Software, Fudan University

CONTENTS

USER GUIDE

A. INTRODUCTION

This application is a simple android maze game. In this game, you can control the character in the screen to go through different level of maze to accomplish the task in this game.

B. MAIN MENU IN THE WELCOME PAGE

A. STORY MODE

In this mode, you will experience the full four levels of the game.

B. SANDBOX MODE

In this mode, you will be able to choose an appropriate level according to your preference, and you will exit from the game page after the selected level finishes.

C. RESTORE PROCESS

If you have stored a process before, you can continue your game with this function.

D. RANKING LIST

It lists all the players and the scores if they finished the game.

E. SETTINGS MENU

The settings menu is on the top right corner of the screen, denoted by three dots. You can configure the length of footprints and your username in the settings page.

C. EFFECTS

This game provide special sound effect when you kill a monster, pick a treasure, pass a level or are killed by a monster. In addition, the background music will be on in the full process a story mode game.

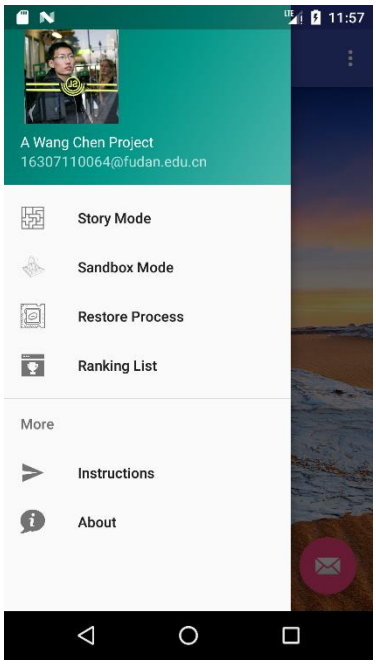
D. GAME RULES

Different level have different map and different scenes. The first level have no other elements other than the player. You can feel free to go anywhere in the maze. And in the first level, the crack button does not function. In addition, all your steps are stored once you start the game. Thus you are able to press the back button to go back. You can even go back to the previous level if you are in the story mode and in a higher level. If you go in the direction of wall, there will be a notification and the player cannot move. The start point in each level is in the upper-left corner and the finish point in each level is the bottom-right corner. The higher the level, the more complex the scene is. In the second level, there comes the treasure, which you can pick to increase your score. In the third level, there comes the monster, which you can kill to increase your score. The effective distance to pick treasure or attack a monster is one step and the crack button can perform both picking treasure and attacking monster. However, if you step onto the monster, you will go dead and your game finishes. In the fourth level, the monster will move in a random direction which can increase the difficulty of the

game. Whenever in the game, you can click the menu in the upper-right corner to quit with/without saving your process. If you finish the whole four levels, the game ends and your score will be updated in the ranking list automatically.

E. DEMONSTRATION WITH GAME USER INTERFACES EXAMPLES

A. THE WELCOME PAGE

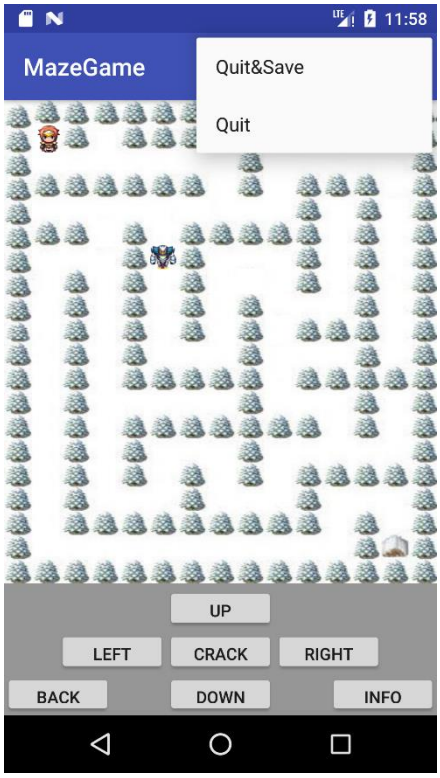


You can start the game by either press story mode or sandbox mode. It is recommended that you go to the settings page first.

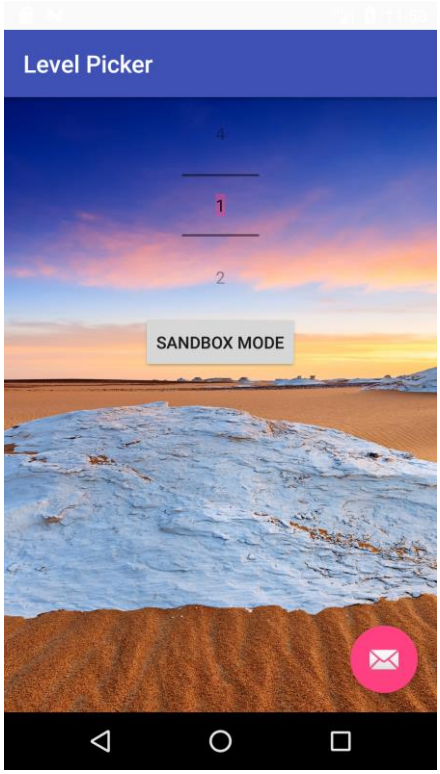
B. THE SETTINGS PAGE.

You can set the length of footprint and your username first. The username will be used in the ranking list.

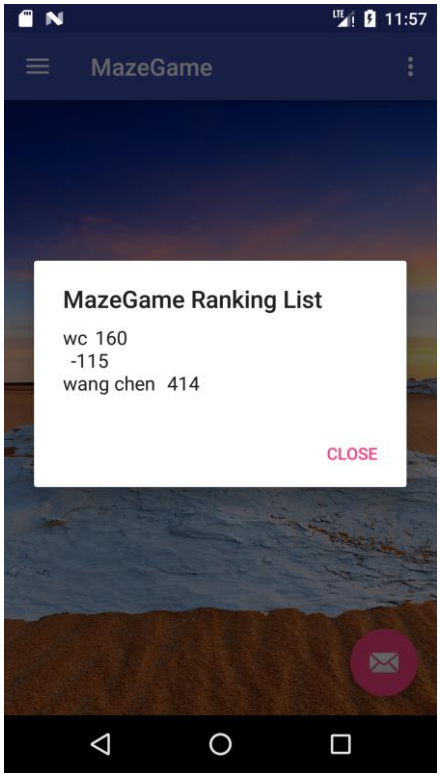
C. THE GAME VIEW PAGE



D. THE PICK LEVEL PAGE IN THE SANDBOX MODE.



E. THE RANKING LIST


11:57
MazeGame
MazeGame Ranking List
wc 160
-115
wang chen 414
CLOSE

APPLICATION CODING DOCUMENTATION

A. ENVIRONMENT DESCRIPTION

This application is written on Android Studio 3.0 with Android Development Kit 1.8 and built on Gradle 4.1. The Gradle repositories in this application adopts google () and jcenter (). The application is both compiled on and have a target SDK of API 26: Android 8.0 (Oreo), which will cover approximately less than 1% Android devices. The minimum API is 24: Android 7.0 (Nougat), which will cover approximately 8.1% Android devices. All the User Interfaces are written in Android XML layout and the classes and methods are all written in Java. The arrangement of project files adhere to the Android Studio project conventions.

B. ACTIVITIES

There are seven activities in this application in total and there are only seven public classes in this project corresponding to the activities. The entrance activity of this application is Main Activity. Main activity contains several menu options and a setting option for the users to choose what they want to do in this application. In the game view, there are four activities corresponding to the four levels of the game. The options and menus in the main activity can lead the user to game view activity, sandbox level picker activity or settings activity. Other options in the menu of the main activity give feedbacks in the form of an alert dialogue.

All the four game view activities have the relationship of inheritance and a lower level activity is the parent class of a higher level. It is designed to be like this because the higher level activity have all the function of a lower level activity plus additional functions.

C. OBJECTS

The application essentially designs three objects to achieve its function. The three objects are the main character in the game, that is, player, treasure and monster. To make the code cleaner and simpler, all the three elements extend the abstract class Element. The essential methods and data field is defined in the abstract class Element.

D. CONSTANTS

The class constants store all the four maps. The maps are defined as static private final, and there are getter method in this class.

E. LAYOUTS

All the four game view activity layout file share the buttons pane layout because all the four game views have the same control area. Frame layout is widely used in the game view layout and all the elements (i.e. the treasure, monster, and player) are added in the layout in an Image View form. All the layout files are linked to the corresponding java files via the URI.