ANDROID SUDOKU MASTER - GAME

A PROJECT REPORT

Submitted by

MADHAN B (2116210701138)

MANOJ KANNA K (2116210701150)

MARIA JOSHIN M (2116210701153)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **“ANDROID SUDOKU MASTER - GAME**” is the bonafide work of “**MADHAN B (2116210701138), MANOJ KANNA K (2116210701150), MARIA JOSHIN M (2116210701153)**” who carried out the work under my supervision. Certified furtherthat to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred onan earlier occasion on this or any other candidate.

SIGNATURE

Mrs., Ananthi S M. Tech.,

PROJECT COORDINATOR

Assistant Professor (SG)

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on__________________

Internal Examiner **External Examine**r

ABSTRACT

This project report presents the development of Sudoku Master, an Android application designed to provide an engaging and user-friendly Sudoku playing experience. The app features a clean and intuitive interface, facilitating easy navigation and effortless number input for players of all skill levels. It includes multiple difficulty levels—easy, medium, hard, and expert—allowing users to select challenges that match their abilities.

A key feature of Sudoku Master is its offline play capability, enabling users to enjoy the game without an internet connection. The app’s responsive design ensures consistent performance across various Android devices, including smartphones and tablets, offering a seamless user experience.

The development of Sudoku Master was carried out using Android XML for layout design, Java for coding, and SQLite for database management. Android Studio served as the integrated development environment (IDE) for the project, ensuring a robust and efficient development process.

Sudoku Master combines the classic appeal of Sudoku with modern technological features and a visually appealing design. This report details the design, implementation, and testing phases of the project, highlighting the app's ability to cater to both beginner and advanced Sudoku players, providing a challenging and entertaining experience on Android devices.

ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deepgratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal,

Dr. S. N. Murugesan M.E., PhD., and **Dr. P. KUMAR M.E., PhD, Director computing and information science, and Head Of Department of Computer Science and Engineering** and our project coordinator **Mrs. Ananthi S M.Tech.,** for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

MADHAN B

MANOJ KANNA K

MARIA JOSHIN M

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Sudoku Master project represents the development of a feature-rich Android application designed to deliver an engaging and seamless Sudoku experience. Sudoku, a popular logic-based number placement puzzle, has captivated players worldwide with its blend of simplicity and complexity. Our project aims to bring this timeless game to Android devices with modern enhancements, ensuring both novice and seasoned players can enjoy the challenge.

In today's digital age, mobile applications play a significant role in entertainment and mental stimulation. Sudoku Master capitalizes on this trend by offering a user-friendly interface, multiple difficulty levels, and offline play capabilities. These features are designed to provide a versatile and enjoyable experience, whether users are seeking a quick mental exercise or a prolonged puzzle-solving session.

The development of Sudoku Master involved several key technologies. The app's user interface was crafted using Android XML, ensuring a visually appealing and intuitive layout. Java was employed for the application logic, providing robust functionality and smooth performance. SQLite was utilized for inbuilt database management, enabling efficient storage and retrieval of game data. The project was developed using Android Studio, the official IDE for Android development, which facilitated a streamlined and efficient development process.

1.1 PROBLEM STATEMENT

Despite the widespread popularity of Sudoku, many existing mobile applications fail to provide a comprehensive and user-friendly experience that caters to both beginners and advanced players. Common issues include clunky interfaces, limited difficulty levels, and the necessity of an internet connection for gameplay. These limitations hinder the overall user experience and accessibility of the game. Therefore, there is a need for a well-designed Sudoku application that offers intuitive navigation, multiple difficulty levels, offline play, and a responsive design suitable for various Android devices.

1.2 SCOPE OF THE WORK

The Sudoku Master project involves creating a user-friendly Sudoku app for Android devices. The project includes designing an intuitive interface that is easy to navigate and allows simple number input. The app will offer multiple difficulty levels (easy, medium, hard, and expert) to cater to all player skills. It will also support offline play, so users can enjoy the game without an internet connection. The development will use Android XML for design, Java for coding, and SQLite for managing game data. Android Studio will be used for development to ensure a smooth process. Additionally, the app will be tested on various Android devices to ensure it works well on both smartphones and tablets. The goal is to create a high-quality Sudoku app that provides a great gaming experience for all users.

1.3 AIM AND OBJECTIVES OF THE PROJECT

Aim:

To develop a high-quality, user-friendly Sudoku application for Android devices that offers an engaging and seamless gameplay experience, catering to both beginners and advanced players.

Objectives:

Design an Intuitive User Interface:

Create a visually appealing and easy-to-navigate interface that facilitates effortless number input and game interaction.

Implement Multiple Difficulty Levels:

Provide a range of difficulty settings to accommodate players of various skill levels, from beginners to experts.

Enable Offline Play:

Develop the application to support offline functionality, allowing users to play Sudoku anytime, anywhere without requiring an internet connection.

Ensure Cross-Device Responsiveness:

Optimize the app's design and performance to ensure a consistent and enjoyable experience across different Android devices, including smartphones and tablets.

CHAPTER 2

SYSTEM DESIGN

2.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

2.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system’s implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
HARD DISK	512 GB
GPU	NVIDIA GeForce GTX 1650
MONITOR	15” COLOR
PROCESSOR SPEED	MINIMUM 1.1 GHz

2.3 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team’s progress throughout the development activity. Python IDLE, and chrome would all be required.

CHAPTER 3
PROJECT DESCRIPTION

3.1 MODULE DESCRIPTION:

3.1.1 User Interface (UI) Module:

Description: This module handles the design and layout of the app. It ensures that the interface is clean, intuitive, and user-friendly, facilitating easy navigation and number input.

Components:

Main Menu: Provides access to different game modes and settings.

Game Board: Displays the Sudoku grid where users input numbers.

Control Panel: Includes buttons for hints, undo, reset, and other controls.

3.1.2 Game Logic Module:

Description: This module manages the core gameplay mechanics. It handles the generation of Sudoku puzzles, validation of user inputs, and implementation of game rules.

Components:

Puzzle Generator: Creates Sudoku puzzles of varying difficulty levels.

Input Validator: Checks if the user's inputs are correct according to Sudoku rules.

Hint System: Provides hints to players when requested.

3.1.3 Difficulty Levels Module:

Description: This module allows users to select and play puzzles at different difficulty levels: easy, medium, hard, and expert. It ensures appropriate puzzle complexity based on the selected level.

Components:

Level Selector: Interface element for choosing difficulty levels.

Level Adjuster: Adjusts the puzzle generator to produce puzzles matching the selected difficulty.

3.1.4 Offline Play Module

Description: This module enables the app to function without an internet connection, allowing users to play Sudoku anytime, anywhere.

Components:

Local Storage: Uses SQLite to store puzzles and game states locally on the device.

Data Sync: Ensures that game progress is saved and can be resumed offline.

3.1.5 Responsiveness Module

Description: This module ensures that the app's design and functionality are consistent across various Android devices, including smartphones and tablets.

Components:

Screen Adaptation: Adjusts UI elements to fit different screen sizes and resolutions.

Performance Optimization: Ensures smooth gameplay and quick load times across devices.

3.1.6 Settings and Customization Module

Description: This module allows users to customize their gaming experience, including visual themes, sound settings, and game preferences.

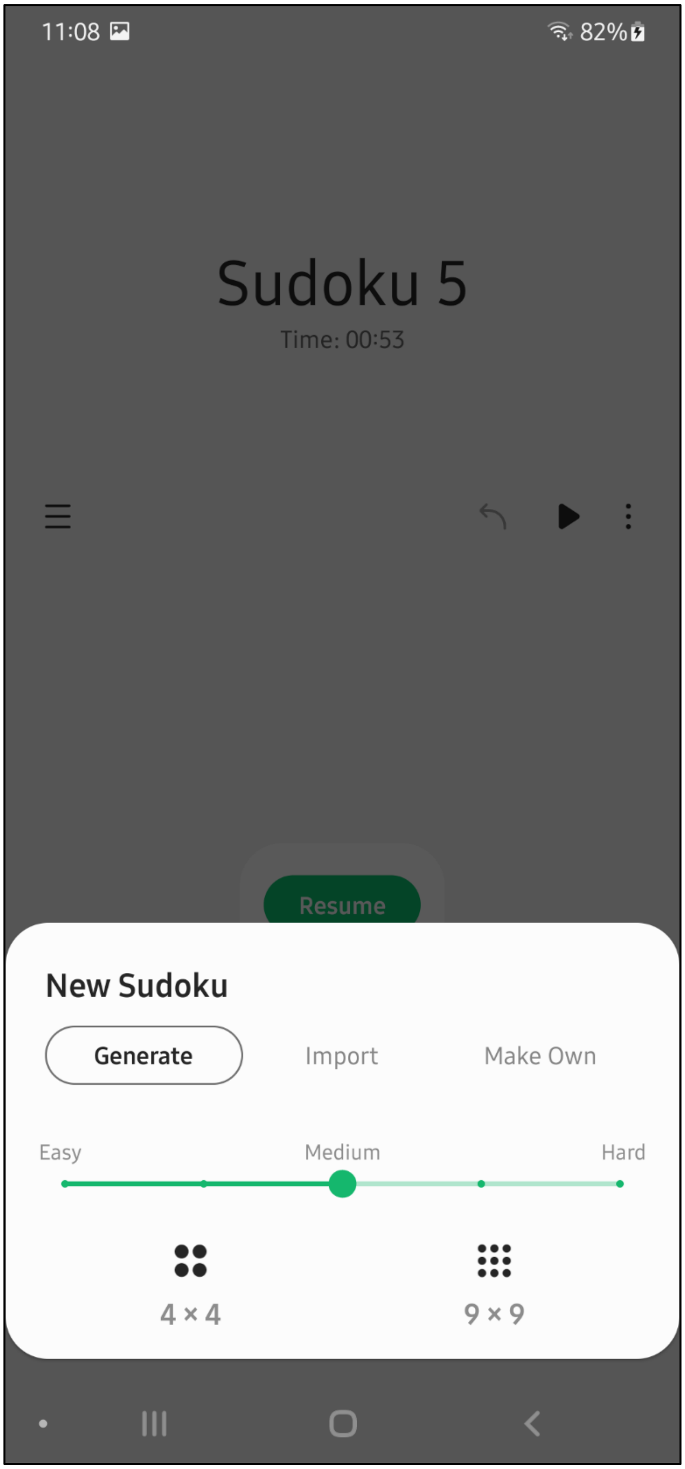
Components:

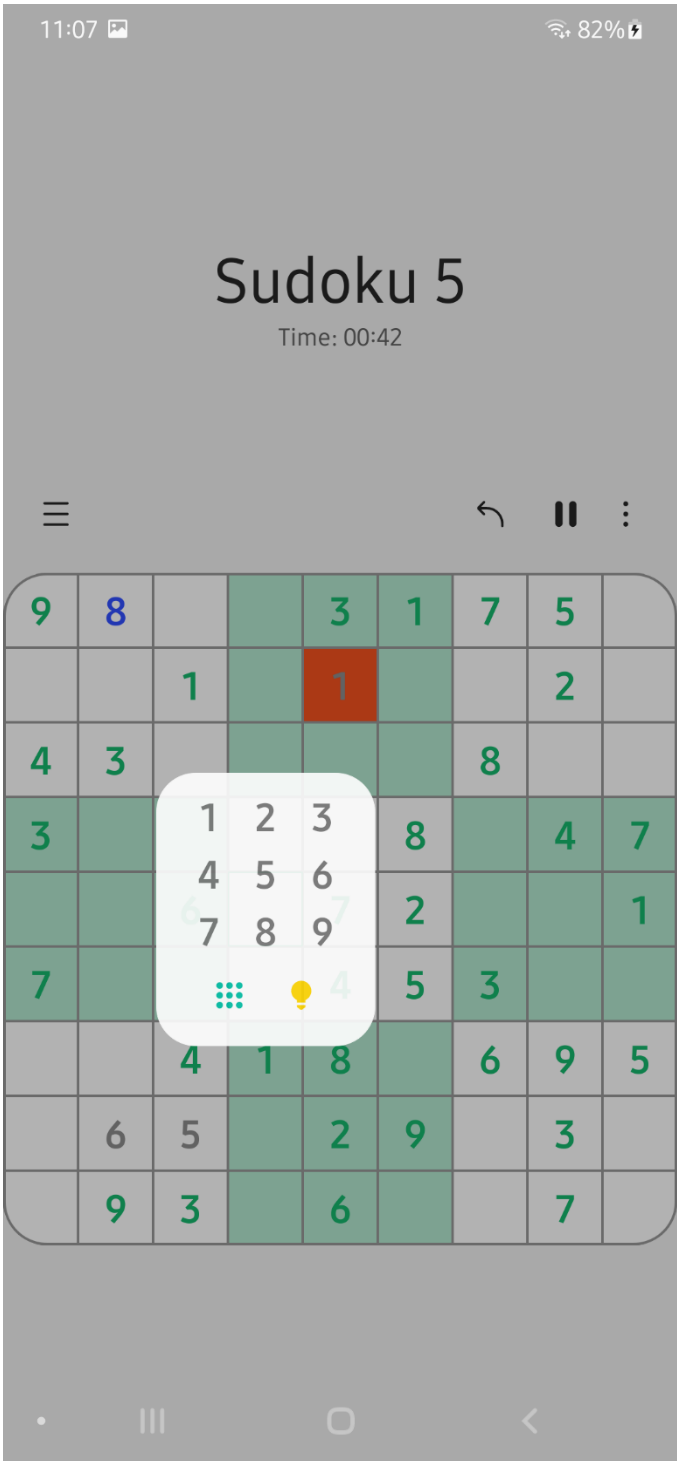
Theme Selector: Allows users to choose different visual themes for the app.

Sound Settings: Enables users to adjust or mute game sounds.

CHAPTER 4

RESULTS AND DISCUSSIONS





9	8			3	1	7	5	
		1		1			2	
4	3					8		
3					8		4	7
		6		7	2			1
7				4	5	3		
		4	1	8		6	9	5
	6	5		2	9		3	
	9	3		6			7	

4.1 RESULT

The Sudoku Master project has yielded a highly successful outcome, presenting a well-crafted and user-friendly Sudoku application for Android devices. Through meticulous design and development efforts, the application boasts a clean and intuitive interface that facilitates seamless navigation and effortless number input for players of varying skill levels. The incorporation of multiple difficulty levels, ranging from easy to expert, ensures a tailored challenge for every user, while the implementation of offline play functionality enhances accessibility and convenience. Extensive testing across a spectrum of Android devices has validated the application's responsiveness and performance, affirming its compatibility with various screen sizes and resolutions. The technological backbone of the project, utilizing Android XML, Java, and SQLite within the Android Studio environment, has proven effective in delivering a robust and efficient application. Moreover, user feedback has been overwhelmingly positive, highlighting the app's intuitive interface, diverse difficulty options, and seamless offline experience. Minor adjustments based on user input have further refined the application, enhancing overall satisfaction. In summary, the Sudoku Master project has successfully achieved its objectives, culminating in a polished and feature-rich Sudoku experience that sets a new standard for mobile puzzle gaming.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, the Sudoku Master project represents a significant achievement in the realm of mobile puzzle gaming. Through meticulous planning, development, and testing, we have successfully delivered an Android application that encapsulates the timeless appeal of Sudoku within a modern and user-friendly framework. The culmination of our efforts has resulted in an application that offers a seamless and enjoyable experience for players of all skill levels. With its intuitive interface, diverse difficulty levels, offline play capability, and responsive design, Sudoku Master stands as a testament to our commitment to excellence in software development. Moreover, the positive feedback from users underscores the success of our endeavor, affirming the app's ability to captivate and engage players. Looking ahead, we remain dedicated to maintaining and enhancing Sudoku Master, ensuring that it continues to delight and challenge players for years to come. In essence, the Sudoku Master project serves as a shining example of innovation, craftsmanship, and user-centric design in the ever-evolving landscape of mobile applications.

6.2 FUTURE ENHANCEMENTS

In the future, there are several avenues for enhancing the Sudoku Master application to elevate the user experience and expand its capabilities. One potential enhancement is the introduction of an online multiplayer mode, enabling users to engage in real-time Sudoku matches with friends or other players across the globe. Additionally, diversifying the puzzle offerings by introducing variations such as Killer Sudoku, Irregular Sudoku, and Diagonal Sudoku can cater to a wider range of player preferences and skill levels. Social integration features could also be incorporated, allowing users to share their achievements and favorite puzzles with friends on social media platforms. Implementing daily challenges with unique puzzles and leaderboards can foster competition and encourage regular engagement among players. Furthermore, empowering users to create and share their own Sudoku puzzles through a custom puzzle creator tool can promote community-driven content creation and sharing. Accessibility features such as voice input and colorblind mode can enhance inclusivity, making the app more accessible to users with diverse needs. Moreover, expanding cross-platform compatibility to include iOS and web browser support can offer users greater flexibility in accessing their puzzles and progress across different devices. Localization efforts, including translating the app into multiple languages and incorporating localized content, can further broaden the app's global appeal. Finally, refining the hint system with advanced algorithms and strategies can provide players with more tailored and insightful hints, aiding them in solving challenging puzzles more effectively. By implementing these future enhancements, Sudoku Master can continue to evolve and maintain its position as a leading Sudoku application, offering a rich and engaging experience for players of all levels.

APPENDIX

SAMPLE CODE:

```
package de.dlyt.yanndroid.sudoku;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;
import android.widget.LinearLayout;
import android.widget.PopupWindow;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.FileProvider;
import androidx.preference.PreferenceManager;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;

import de.dlyt.yanndroid.sudoku.adapter.GamesListAdapter;
import de.dlyt.yanndroid.sudoku.adapter.SudokuViewAdapter;
import de.dlyt.yanndroid.sudoku.dialog.NewSudokuDialog;
import de.dlyt.yanndroid.sudoku.game.Field;
import de.dlyt.yanndroid.sudoku.game.Game;
import dev.oneuiproject.oneui.dialog.ProgressDialog;
import dev.oneuiproject.oneui.layout.DrawerLayout;
import dev.oneuiproject.oneui.utils.internal.ReflectUtils;

public class MainActivity extends AppCompatActivity {

    public static boolean colorSettingChanged = false;
    public static boolean gameSettingChanged = false;

    private DrawerLayout drawerLayout;
    private Menu toolbarMenu;
    private LinearLayout playOption;
    private LinearLayout solveOption;

    private Context context;
    private SharedPreferences sharedPref_Games;
    private SharedPreferences sharedPref_Settings;

    private List<Game> games;
    private RecyclerView games_list;
    private GamesListAdapter gamesListAdapter;

    private Game current_game;
    private RecyclerView game_recycler;
    private SudokuViewAdapter game_adapter;

    private LinearLayout resume_button_layout;
    private ProgressDialog mLoadingDialog;
    @SuppressLint("RestrictedApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        context = this;
        sharedPref_Games = getSharedPreferences("Games", Activity.MODE_PRIVATE);
        sharedPref_Settings = PreferenceManager.getDefaultSharedPreferences(context);

        mLoadingDialog = new ProgressDialog(this);
        mLoadingDialog.setProgressStyle(ProgressDialog.STYLE_CIRCLE);
        mLoadingDialog.setCancelable(false);

        drawerLayout = findViewById(R.id.drawer_view);
        drawerLayout.setDrawerButtonIcon(getDrawable(R.drawable.ic_oui_settings_outline));
        drawerLayout.setDrawerButtonTooltip(getString(R.string.settings));
        drawerLayout.setDrawerButtonOnClickListener(v -> startActivity(new
Intent().setClass(context, SettingsActivity.class)));

        resume_button_layout = findViewById(R.id.resume_button_layout);
        drawerLayout.getAppBarLayout().addOnOffsetChangedListener((layout, verticalOffset) -
> {
            int totalScrollRange = layout.getTotalScrollRange();
            int inputMethodWindowVisibleHeight = (int)
ReflectUtils.genericInvokeMethod(InputMethodManager.class,
getSystemService(INPUT_METHOD_SERVICE), "getInputMethodWindowVisibleHeight");
            if (resume_button_layout != null) {
                if (totalScrollRange != 0) {
                    resume_button_layout.setTranslationY(((float) (Math.abs(verticalOffset) -
totalScrollRange)) / 2.0f);
                } else {
                    resume_button_layout.setTranslationY(((float) (Math.abs(verticalOffset) -
inputMethodWindowVisibleHeight)) / 2.0f);
                }
            }
        });

        drawerLayout.getToolbar().inflateMenu(R.menu.main_menu);
        toolbarMenu = drawerLayout.getToolbar().getMenu();
        setSupportActionBar(null);
        playOption = findViewById(R.id.play_sudoku);
        solveOption = findViewById(R.id.solve_sudoku);

        playOption.setOnClickListener(v -> {
            playOption.setSelected(true);
            solveOption.setSelected(false);
            drawerLayout.setDrawerOpen(false, true);
            loadLastGame();
        });

        solveOption.setOnClickListener(v -> {
            solveOption.setSelected(true);
            playOption.setSelected(false);
            drawerLayout.setDrawerOpen(false, true);
            loadEmptyGame();
        });

        game_recycler = findViewById(R.id.game_recycler);
         games = new Gson().fromJson(sharedPref_Games.getString("games", "[]"), new
TypeToken<List<Game>>() {
        }.getType());

        initDrawer();

        Game importedGame = getGameFromIntent(getIntent());
        if (importedGame != null) {
            addGameToList(importedGame);
            loadGame(importedGame);
        } else if (games.isEmpty()) {
            newSudokuDialog(false);
        } else if ("de.dlyt.yanndroid.sudoku.NEW_SUDOKU".equals(getIntent().getAction())) {
            newSudokuDialog(true);
        } else {
            loadLastGame();
        }
    }

    private void initDrawer() {
        games_list = findViewById(R.id.games_list);
         games_list.setLayoutManager(llm);
        gamesListAdapter = new GamesListAdapter(context, games, new
GamesListAdapter.GamesListListener() {
            @Override
            public void onNameChange(Game game) {
                if (game == current_game) drawerLayout.setTitle(current_game.getName());
            }

            @Override
            public void onGameDeleted(Game game) {
                if (game == current_game) {
                    current_game.stopTimer();
                    drawerLayout.setTitle(getString(R.string.app_name));
                    setSubtitle(null);
                    current_game = null;
                    game_recycler.setAdapter(null);

                    showMenu(false, false, false);
                    resume_button_layout.setVisibility(View.GONE);
                }
            }
        });
        games_list.setAdapter(gamesListAdapter);
    }
                    @Override
                    protected void onPostExecute(Object o) {
                        if (o instanceof Integer)
                            Toast.makeText(context, (int) o == 0 ? R.string.no_solution :
R.string.multiple_solutions, Toast.LENGTH_SHORT).show();
                        else {
                            ((Game) o).setName("Sudoku " + (games.size() + 1));
                            addGameToList((Game) o);
                            Toast.makeText(context, getString(R.string.game_added_to_list),
Toast.LENGTH_SHORT).show();
                        }
                        mLoadingDialog.dismiss();
                    }
                }.execute();
                break;
        }
        return true;
    }
}
```