**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ……………………………………..…..…5

1. АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ…………..….……
   1. Загальні вимоги до програми…………………………………………….
   2. Призначення та область застосування………………………….…….…
   3. Функціональні вимоги……………………………………………………
2. ОПИС ПРОГРАМИ…………………………………………………….……
   1. Структура програми………………………………………………..……
   2. Опис методів програми…………………………………………….……
   3. Програмні засоби………………………………………………...………
   4. Опис користувацького інтерфейсу………………………………..……

ВИСНОВКИ……………………………………………………………...………

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……………………………....….……

ДОДАТКИ………………………………………………………………..………

Додаток А. Скролінг (текст) програми………………………………….….

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ОС – операційна система.

**якщо Ваша робота не містить мало поширені умовні позначення, символи, одиниці, скорочення та терміни, тоді даний розділ у курсовий проект не включати !!!**

**1. АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ ПОКАЗНИКИ**

**1.1 Загальні вимоги до програми**

**1.1.1 Вимоги до графічного інтерфейсу користувача**

1. Робоча мова інтерфейсу – українська.
2. Необхідно сконструювати такі вікна:
   1. вікно головного меню, в якому передбачити кнопки для доступу до головних вікон гри («Грати», «Досягнення», «Правила гри», «Допомога», «Про гру»).
   2. вікно з правилами гри, в якому передбачити кнопку повернення до головного меню.
   3. вікно допомоги, в якому передбачити кнопку повернення до головного меню.
   4. вікно з таблицею результатів проведених ігор, в якому передбачити кнопку повернення до головного меню.
   5. вікно з ігровим полем, в якому передбачити поле кнопок для основної гри, кнопку повороту для рандомно згенерованої труби, кнопку для запуску нової гри та кнопку виклику викидного меню.
   6. викидне меню, в якому передбачити кнопку повернення до гри зі збереженням всіх попередніх дій гравця та таймером, кнопки виходу в головне меню та кнопку початку нової гри.
   7. вікно з формою вводу імені, коли гру виграно, в якому передбачити текстове поле та кнопку «ОК», яка закриває вікно та повертає гравця до попереднього вікна.
3. Додатково необхідно передбачити управління з клавіатури, де клавіша Esc буде закривати поточне меню, клавіша F1 буде викликати вікно з правилами до гри або викидне меню, коли гру почато, клавіша Enter буде підтверджувати вибір користувача або завершувати ввід даних, клавіша Tab буде переходити до наступної кнопки в вікні, а комбінація клавіш Tab+Shift – повертатися до попередньої кнопки, доступної в вікні.
4. Поява відповідного повідомлення при програші гри, виборі кнопки «Про гру» в головному меню, виборі недоступної кнопки для того, щоб зробити хід, в вікні ігрового поля та при вичерпанні 30 секунд гри, щоб попередити учасника про кінець гри через 15 секунд.
5. Кольорова гама (палітра) за замовчуванням:

* фон – темно-синій;
* кольори фігур – див. Табл.1;
* кнопки керування повинні бути жовтого кольору з нанесеними на них піктограмами, що однозначно пояснюють їх призначення.

1. У програмі необхідно використати шрифт Arial Narrow звичайного, напівжирного накреслень та курсив.

**1.1.2 Вимоги до архітектури програми**

1. Використання стандартних контейнерних класів, які реалізують:

* вибір режимів роботи в програмі;
* завантаження нової гри;
* вибір режиму роботи «адміністратор»;
* зчитування / запис конфігурації гри;
* механізм потрапляння / оновлення списку Топ-10;
* завантаження звукових файлів;
* роботу зі звуком (відтворення);
* роботу із БД.

**1.1.3 Вимоги до функціональності додатка**

1. Початок гри з вибору режиму роботи програми (режиму гравця або адміністратора) з відображенням функціоналу, який відповідатиме вибраному користувачем режиму роботи.
2. Зчитування файлу конфігурації ПЗ.
3. Генерація чергової фігури, яка з’являтиметься у верхній частині ігрового поля.
4. Виконання перевірки заповнення ігрового поля і досягнення його границь.
5. Очікування керуючих дій від користувача. Якщо таких дій не буде, програма пересуватиме поточну фігуру на одну клітинку вниз та перевірятиме, чи не досягнуто кінця ігрового поля або фігур, розташованих нижче.
6. При натисненні користувачем однієї з керуючих кнопок («ліворуч», «праворуч», «скидання» або «обертання»), програма перевірятиме, чи не досягнуто границь поля, та пересуватиме фігуру на одну клітинку у відповідний бік, скидатиме донизу або обертатиме її.
7. Якщо після відповідних маніпуляцій користувача з’явиться хоча б одна повна лінія, програма знищуватиме її, зсуваючи усі вищі лінії вниз, а гравцю нараховуватиметься 10 балів.
8. Генерація наступної фігури, яка з’являється у верхній частині ігрового поля, після нарахування балів гравцю.
9. Продовження гри до тих пір, допоки повністю не заповниться хоча би один стовпчик ігрового поля.
10. Запуск режиму оновлення таблиці рекордів, якщо гравець набрав більше балів, ніж останній гравець з Топ-10.
11. Зчитування з файлу Топ-10 даних найкращих гравців та порівняння кількості балів останнього з кількістю балів поточного гравця. Якщо останнє значення більше за рекорд №10, гра запитуватиме ім’я гравця та разом з його кількістю балів записуватиме до Топ-10 рекордів.
12. Відтворення поліфонічної мелодію «We are the Champions», яка завантажується з файлу *midi*-формату.
13. Проектування гри у вертикальному форматі без можливості переходу в горизонтальний при повороті смартфона.
14. Читання всіх даних з файлу бази та їх відображення при порівнянні кількості балів із списку Топ-10.
15. Додавання елементу даних до файлу бази при потраплянні до списку Топ-10.
16. Оновлення будь-якого елемента даних у файлі бази при оновленні списку Топ-10.
17. Видалення будь-якого елемента даних з файлу бази при програші гравця.
18. Сортування інформації, що відображується в графічному інтерфейсі користувача, за різними реквізитами (ПІБ, кількість балів, дата проходження гри).
19. Отримання та відображення підсумкової інформації по завершенню гри.

**1.2 Призначення та область застосування**

**Мета роботи** полягає у розробці інтерактивної гри «Тетріс» для смартфонів, які працюють під керуванням ОС Android.

# Реалізована версія гри «Тетріс» призначена для розважання користувачів смартфонів, що працюють під управлінням ОС Android, при проведенні дозвілля, тренуванні просторової уяви, реакції та прогнозування своїх дій.

# Область застосування – програмне забезпечення для недорогих смартфонів та планшетів з обмеженими обчислювальними ресурсами.

**1.3 Функціональні вимоги**

До програмного забезпечення висуваються такі функціональні вимоги:

1. Програмне забезпечення повинно забезпечувати роботу користувача у двох режимах:

* режим гравця;
* режим адміністратора.

Перехід користувача до ролі адміністратора повинен відбуватися без додаткової аутентифікації.

1. Фігури, які повинна забезпечувати гра:

Таблиця 1 – Параметри фігур гри

|  |  |
| --- | --- |
| **Тип фігури** | **Кількість клітинок та колір** |
|  | 4 клітинки по горизонталі – тип І (колір зелений). |
|  | Тип J (колір синій). |
|  | Тип L (колір коричневий). |
|  | Тип Т (колір фіолетовий). |
|  | Тип О (колір жовтий). |

* 1. Програма повинна забезпечувати зміну та налаштування кількості фігур, їх структури, кольору та кількості клітинок, яку займає фігура в режимі адміністратора.
  2. Поле гри – прямокутник висотою 20 клітин, шириною 10 клітин, розмір яких такий самий, як і у фігур.
  3. Якщо довільна лінія ігрового поля повністю заповнюється фігурами (або їх частинами), вона повинна зникнути з екрану. При цьому всі лінії, розташовані вище за неї, зсуваються вниз, а гравцеві нараховується 10 балів.
  4. Мета гри – набрати якомога більше балів.
  5. Гра закінчується, коли хоча би один стовпчик досяг висоти у 20 клітин. Гра повинна забезпечувати звуковий сигнал та повідомлення про закінчення гри.
  6. Гра повинна забезпечувати можливість продовження гри спочатку або її повне завершення з виходом до операційної системи.

1. Функціональні вимоги до режиму гравця:
   1. Гра повинна надавати можливість обертати та пересувати активну фігуру за допомогою віртуальних кнопок смартфона.
   2. Напрям обертання не нормується.
   3. Кут обертання – 90 градусів за одне натискання на кнопку.
   4. Фігура повинна пересуватися на одну клітинку в горизонтальному напрямку (ліворуч або праворуч) залежно від натиснутої кнопки.
   5. Гра повинна забезпечувати можливість ведення рахунку гравця та таблиці рекордів десяти найкращих гравців.
   6. Гравець, який набрав за одну гру балів більше за останнього найкращого гравця з таблиці рекордів повинен потрапити до неї, витіснивши останнього.
   7. Таблиця рекордів повинна зберігатися у постійній пам’яті доти, поки її не буде очищено.
   8. Гра повинна забезпечувати можливість призупинення процесу гри з подальшим продовженням її з того самого місця (пауза).
   9. Гра повинна забезпечувати можливість показу наступної фігури у процесі падіння поточної.
2. Функціональні вимоги до режиму адміністратора:
   1. Адміністратор повинен мати можливість зміни параметрів фігур та налаштувань гри. До цього відноситься:

* зміна типу, кількості, кольору та інших можливих параметрів фігур;
* зміна кольорів інтерфейсу;
* зміна швидкості падіння фігури;
* очищення таблиці рекордів гри;
* вмикання/вимикання процесу скидання фігури, коли визначено її остаточну позицію;
* вмикання/вимикання показу наступної фігури;
* вмикання/вимикання режиму тиші (без звуків);
* зміна мови інтерфейсу програми.
  1. Адміністратор повинен мати можливість збереження налаштувань конфігурації гри у файлі з подальшим зчитуванням його програмним забезпеченням у режимі гравця. Структура файлу конфігурації не нормується.

**2. ОПИС ПРОГРАМИ**

**2.1 Структура програми**

**2.1.1 Модулі програми**

Робота розробленого програмного забезпечення реалізується наступними модулями:

1. EnterScoreActivity – клас, в якому реалізовано створення та збереження користувача.
2. GameActivity – клас, в якому реалізовано дії гри.
3. HighScoreActivity – клас, в якому реалізовано опис рекордів.
4. MainActivity – клас, в якому реалізовано опис кнопок.
5. SettingsActivity – клас, в якому реалізовано налаштування гри.
6. DatabaseBundler – клас, в якому реалізовано створення файлу для збереження БД.
7. Score – клас, в якому реалізовано опис рекордів всіх користувачів.
8. ScoreDBManager – клас, в якому реалізовано управління БД.
9. Block – клас, в якому реалізовано опис блоків.
10. Board – клас, в якому реалізовано перевірка розташування фігури.
11. GameManager – клас, в якому реалізовано управління всієї гри.
12. GameThread – клас, в якому реалізовано можливість контролювати розмір вікна, редагувати пікселі на поверхні.
13. GameView – клас, в якому реалізовано спеціальний малюнок поверхні.

**2.1.2 Алгоритми роботи програми**



Рисунок 1 – Узагальнена схема роботи гри



Рисунок 2 – Блок-схема режиму гравця

### 

### Рисунок 3 – Блок-схема оновлення таблиці рекордів



Рисунок 4 – Блок-схема режиму адміністратора

**2.2 Опис методів програми**

Список методів класу EnterScoreActivity та їх опис наведено в табл.2.

Таблиця 2 – Основні методи класу

|  |  |  |
| --- | --- | --- |
| № | Назва методу | Короткий опис |
| 1 | onCreate () | Створює нового користувача у грі. |
| 2 | btnEnterClick () | Зберігає введене користувачем ім’я та додає його у БД. |

***і так далі …*** для кожного класу описуємо його методи !!!

**2.3 Програмні засоби**

Розробка реалізована засобами середовища Android Studio на мові Java. Дане середовище є зручним у використанні для швидкого та якісного створення додатків на ОС Android.

Android забезпечує повний набір програмного забезпечення для мобільних пристроїв: операційну систему, сполучні програми і ключові мобільні додатки.

Android – це відкрита операційна система, яка була створена «з нуля», що дозволяє розробникам створювати програми, які можуть повною мірою скористатись функціями мобільного пристрою. Наприклад, програма може звернутися до будь-якої основної функції телефону (дзвінки, відправлення текстових повідомлень, використанням камери), що дозволяє розробникам створювати більш багаті і інтуїтивно зрозумілі для користувачів програми.

**2.4 Опис користувацького інтерфейсу**

При запуску гри, гравець має можливість вибрати рівень складності такими кнопками: «Easy», «Normal», «Hard», а також кнопка «Settings» для налагодження звуків в грі. Після натискання, з’являється нове вікно на екрані, де розташовані такі компоненти як кнопки – «Вліво», «Вправо», «Вниз», «Пів-оберта вліво», «Пів-оберта вправо», які знаходиться внизу екрана даного смартфона, інший компонент – це прямокутний стакан висотою 10 і шириною 20 кліток, а також з лівого боку біля нього, можна побачити рівень та очки, які набрав гравець.

# Після закінчення гри появляється вікно з результатами, які користувач набрав.

Розглянемо дії користувача при роботі з програмою. Користувач запускає гру. На екрані смартфона з’являється головне вікно програми (рис.5).



Рисунок 5 – Головне вікно програми

Тут ми бачимо чотири основні кнопки, три для вибирання складності рівня і одна кнопка для налаштування, а саме:

* Easy (легко) – кнопка для легкого режиму гри.
* Normal (середньо) – кнопка для середнього режиму гри.
* Hard (тяжко) – кнопка для тяжкого режиму гри.
* Settings (налаштування) – кнопка для налагодження звуків гри.

# При натисненні на кнопку «Settings», нам відкривається вікно налаштувань (рис.6).

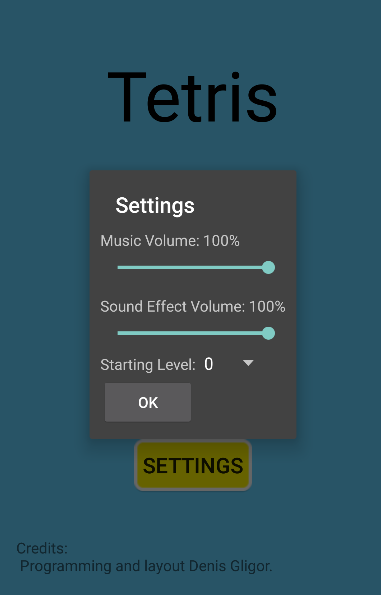


Рисунок 6 – Вікно налаштування

В даному вікні користувач може змінити рівень гучності, а також початковий рівень.

# При натисненні на кнопки «Easy», «Normal» або «Hard» нам відкривається вікно самої гри (рис.7).

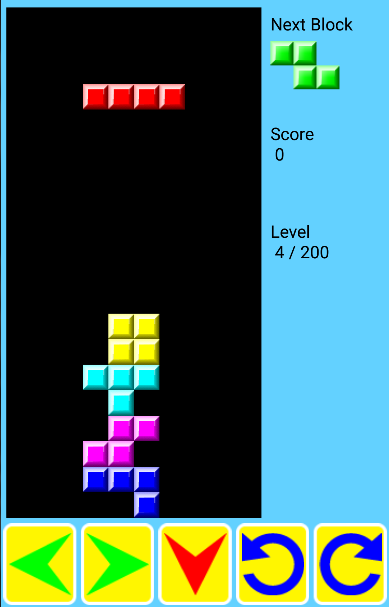


Рисунок 7 – Графічне відображення гри

## В даному вікні здійснивши маніпуляцію над кнопками, користувач сміло може грати.

# При закінченні гри вводимо ім’я для зберігання результату (рис.8). При натисненні на кнопку «Enter», зберігається дані користувача.

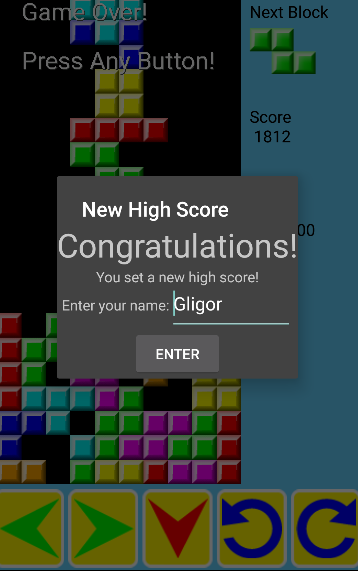


Рисунок 8 – Графічне відображення гри

# Таблиця рекордів даної гри представлена на рис.9.

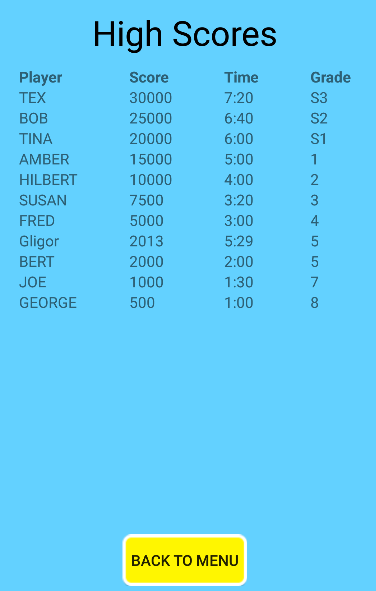


Рисунок 9 – Таблиця рекордів

# При натисненні на кнопку «Back to menu», користувач повертається на головне вікно (рис.5).

**ВИСНОВКИ**

В результаті виконання курсового проекту розроблена гра «Тетріс» під OC Android.

Розроблене програмне забезпечення має наступні можливості:

1. Рухати фігурку вліво-вправо по горизонталі.
2. Здійснити пів-оберта ліворуч та праворуч фігурки.
3. Можливість прискорення фігурки.
4. Введення ім’я гравця.
5. Зберігати та переглядати рекорди.
6. Можливість вибору рівня складності гри.

В даній роботі використовуються Android SDK Tools – це пакет інструментів для розробників, призначення якого полягає в тестуванні додатків з можливістю їх налагодження в реальному часі, що значно спрощує розробку.

Дана розробка у майбутньому може бути розширена із добавленням нового функціоналу і видозміненою логікою обробки.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

## Цехнер Марио. Программирование игр под Android. – СПб.: Питер, 2013. – 688с.

## Дейтел П., Дейтел Х., Дейтел Э. Android для разработчиков. – СПб.: Питер, 2015. – 384 с.

## Genymotion [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Genymotion © 2016. – Режим доступу: https://docs.genymotion.com/pdf/PDF\_Plugin\_for\_Android\_Studio/Plugin-for-Android-Studio-1.0.7-Guide.pdf (дата звернення 15.03.2016) – Назва з екрана.

**ДОДАТКИ**

**ДОДАТОК А. Скролінг (текст) програми**

**class EnterScoreActivity**

# package ca.denisgligor.texmobile.activities;

# import android.content.Intent;

# import android.os.Bundle;

# import android.support.v7.app.AppCompatActivity;

# import android.view.View;

# import android.widget.EditText;

# import ca.denisgligor.texmobile.R;

# public class EnterScoreActivity extends AppCompatActivity {

# private EditText txtName;

# @Override

# protected void onCreate(Bundle savedInstanceState) {

# super.onCreate(savedInstanceState);

# setContentView(R.layout.activity\_enter\_score);

# txtName = (EditText) findViewById(R.id.txtName);

# txtName.setText(getSharedPreferences("lastName", 0).getString("lastName", ""));

# }

# public void btnEnterClick(View v)

# {

# // Remember the name they enter so they don't have to type it again

# String name = txtName.getText().toString();

# getSharedPreferences("lastName", 0).edit().putString("lastName", name).commit();

# // Send the entered name back to the game activity

# Intent output = new Intent();

# output.putExtra("name", name);

# setResult(RESULT\_OK, output);

# finish();

# }

# }

**class GameActivity**

package ca.denisgligor.texmobile.activities;

import android.content.Intent;

import android.graphics.drawable.NinePatchDrawable;

import android.media.AudioManager;

import android.media.MediaPlayer;

import android.media.SoundPool;

import android.os.Bundle;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.AppCompatActivity;

import android.view.MotionEvent;

import android.view.View;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.Hashtable;

import java.util.List;

import ca.denisgligor.texmobile.game.GameManager;

import ca.denisgligor.texmobile.game.GameView;

import ca.denisgligor.texmobile.R;

import ca.denisgligor.texmobile.db.ScoreDBManager;

import ca.denisgligor.texmobile.game.Block;

public class GameActivity extends AppCompatActivity{

private GameView gameView;

private List<String> input;

private MediaPlayer mp;

private SoundPool sp;

private int[] soundEffects;

private float volume;

private ImageView imgNext;

private TextView txtScore;

private TextView txtLevel;

private Hashtable<Block.Shape, NinePatchDrawable> nextPieces;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_game);

input = new ArrayList<>();

gameView = (GameView) findViewById(R.id.svBoard);

txtScore = (TextView) findViewById(R.id.txtScore);

txtLevel = (TextView) findViewById(R.id.txtLevel);

imgNext = (ImageView) findViewById(R.id.imgNext);

***і так далі… наводите весь код***