IT ШКОЛА SAMSUNG

DRUM HERO

Город: Москва

Площадка: Москва ТОЦ-15

Учащийся: Магомедов Шамиль

Преподаватель: Ильин Владимир Владимирович

Цели

- Создать игру, связанную с музыкой.
- Изучить JAVA и Android, получить опыт в ООП.
- Расширить возможности игр музыкального жанра.
- Реализовать работу с МИДИ-файлами на андроид 4.x и выше.
- Сделать приложение работоспособным на экранах с различным разрешением и плотностью.

Задачи

- Создать алгоритм запуска нот в нужную секунду.
- Создать “мозги” игры.
- Создать необходимое количество дополнительных классов.
- Придумать оформление.
- Создать конструктор, который из файла получит код песни для игры.
- Добавить некоторое количество песен по умолчанию.

IT ШКОЛА SAMSUNG

Похожие игры

Жанр моей игры очень популярен и имеет большое количество фанатов во всем мире. Игр этого жанра тоже не мало, но у них всех есть очень важный недостаток : пользователь не имеет возможности добавлять свои песни!! В некоторых приложениях есть данная функция, но совпадение нот с ритмом – нулевое.

IT ШКОЛА SAMSUNG

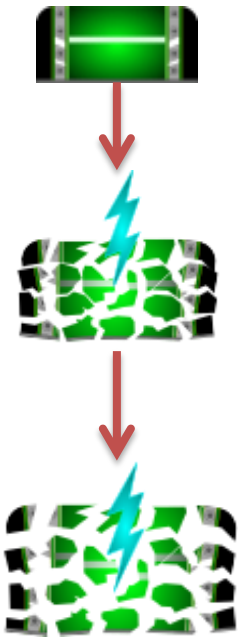
Ноты БД

Конструктор

```
public Note(int numberOfLine, int startSecond)
```

База данных

	NUMBEROFLINE	STARTSECOND
	Фильтр	Фильтр
79	2	17.80
80	4	17.80
81	2	18.05
82	1	18.30
83	4	18.30
84	2	18.80
85	4	18.80
86	1	19.30
87	4	19.30
88	2	19.80
89	4	19.80
90	2	20.05
91	1	20.30
92	4	20.30
93	2	20.85



```
public DBManager(Context context) {
    this.context = context;
    db = context.openOrCreateDatabase(DB_NAME, Context.MODE_PRIVATE, null);
    TABLE_NAME = SongsActivity.activeSong.getTableName();
}

public ArrayList<Note> getAllNotes() {
    ArrayList<Note> dbNotes = new ArrayList<Note>();
    Cursor cursor = db
            .query(TABLE_NAME, null, null, null, null, null, null);
    boolean hasMoreData = cursor.moveToFirst();

    while (hasMoreData) {
        int numberOfLine = cursor.getInt(cursor
                .getColumnIndex("NUMBEROFLINE"));
        int startSecond = (int) (Double.parseDouble(cursor.getString(cursor
                .getColumnIndex("STARTSECOND"))) * 1000);
        System.out.println(Double.parseDouble(cursor.getString(cursor
                .getColumnIndex("STARTSECOND"))));
        dbNotes.add(new Note(numberOfLine, startSecond));
        hasMoreData = cursor.moveToNext();
    }
    return dbNotes;
}
```

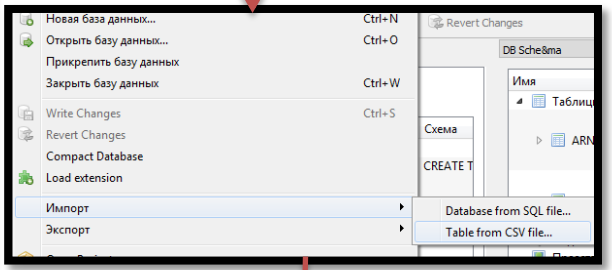
Решение

Вдохновленный этим недостатком, я начал искать решение данной проблемы, и наткнулся на MIDI. MIDI – это именно то, что мне нужно! Благодаря разделению композиции на дорожки и находящиеся на них команды, я получил возможность полностью разбирать песню, в моем случае : получать информацию о каждом бите ритма (тип бита, и в какую секунду он звучит).

IT ШКОЛА SAMSUNG

Первая музыка

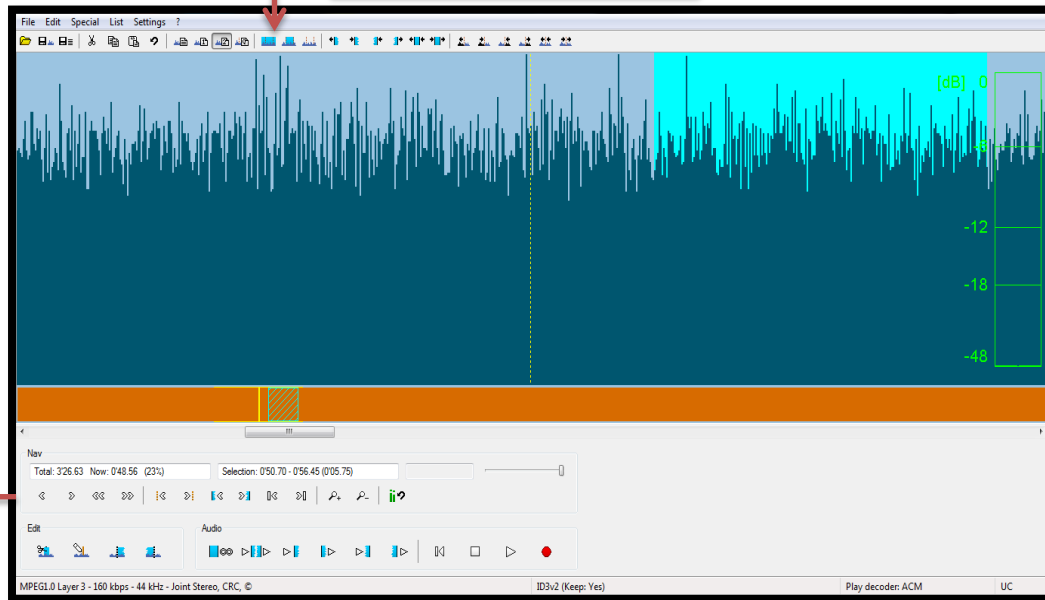
```
2 40.88
1 41.40
2 41.90
2 42.18
1 42.40

2 42.87
2 43.13
1 43.39
2 43.65
2 44.15
1 44.41

2 44.90
2 45.15
1 45.44
2 45.90
2 46.16
1 46.45

2 46.96
3 47.18
2 47.31
3 47.41
2 47.54
```



NumberOfLine

MIDI

(На слух)



Seconds

MIDI

MIDI — стандарт цифровой звукозаписи на формат обмена данными между электронными музыкальными инструментами. Его используют профессиональные музыканты для разных целей (управление разными инструментами с одной клавиатуры, получение графического изображения нот и тд). Благодаря данному формату музыкант может записывать композиции или их части, микшировать без потери качества звука и использования сложных студийных магнитофонов.

Проблема

К сожалению в андроид API есть стандартные средства для работы с миди только после 23 уровня, поэтому пришлось разбираться. В дальнейшем проблема была решена и допустимый api заметно снизился (api приложения = 14). В конечном итоге я могу получать игру с точным соответствием визуального ряда музыке.

android.media.midi

Added in API level 23

Ноты из МИДИ

IT ШКОЛА

Конструктор

```
public Note(int numberOfLine, int startSecond)
```

MIDI

```
Game.notes = new ArrayList<Note>();
float coff = 1;
MidiFile midi = null;
try {
    midi = new MidiFile(new File(Environment.getExternalStorageDirectory().getPath().toString()
            + WebActivity.MUSICFOLDER + "/" + s.getFileName()));
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
long PPQ = midi.getResolution();
List<MidiTrack> tracks = midi.getTracks();
// get tempo
MidiTrack tempoTrack = tracks.get(0);
Iterator<MidiEvent> it = tempoTrack.getEvents().iterator();
float tempo = 1;

TreeSet<MidiEvent> drums = new TreeSet<MidiEvent>();

for (MidiTrack track : tracks) {
    List<MidiEvent> eventsToRemove = new ArrayList<MidiEvent>();
    Iterator<MidiEvent> iev = track.getEvents().iterator();
    while (iev.hasNext()) {

        MidiEvent event = iev.next();
        if (event instanceof Tempo) {
            drums.add(event);
        }
        if (event instanceof NoteOn) {
            NoteOn note = (NoteOn) event;
            if (note.getChannel() == 9 && note.getVelocity() > 50) {
                drums.add(note);
            } else {
                eventsToRemove.add(note);
            }
```

```
219
220                 File file = new File(Environment.getExternalStorageDirectory()
221                         .getPath() + "/mydrums.mid");
222                 for (MidiTrack track : midi.getTracks())
223                     track.insertEvent(new NoteOn(0, 9, 4 + 35, 60));
224                 midi.writeToFile(file);
225
226             } catch (Exception e) {
227                 e.printStackTrace();
228             }
229
230             Iterator<MidiEvent> iev = drums.iterator();
231             int k = 0;
232
233             Game.notes.add(new Note(1, 0));
234             Game.notes.add(new Note(2, 0));
235             Game.notes.add(new Note(3, 0));
236             Game.notes.add(new Note(4, 0));
237             while (iev.hasNext()) {
238                 MidiEvent ev = iev.next();
239                 if (ev instanceof Tempo) {
240                     coff = 60000 / (((Tempo) ev).getBpm() * PPQ);
241                     Log.d("COFF", coff + "");
242                     continue;
243                 }
244                 NoteOn event = (NoteOn) ev;
245                 if (event.getType() != event.NOTE_ON)
246                     continue;
247                 int startSecond = (int) (event.getTick() * coff);
248                 Log.d("NOTE", "" + event.getNoteValue() + ";" + startSecond
249                         + ";" + event.getVelocity());
250                 Note note = null;
251                 switch (event.getNoteValue() - 35) {
252                 case 4:
253                     note = new Note(1, startSecond);
254                     break;
```

IT ШКОЛА SAMSUNG

Музыка сейчас

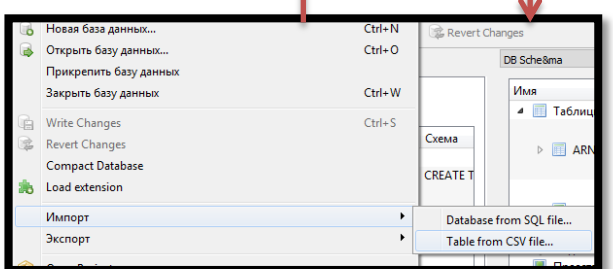
```
2 40.88
1 41.40
2 41.90
2 42.18
1 42.40

2 42.87
2 43.13
1 43.39
2 43.65
2 44.15
1 44.41

2 44.90
2 45.15
1 45.44
2 45.90
2 46.16
1 46.45

2 46.96
3 47.18
2 47.31
3 47.41
2 47.54
```

MIDI

NumberOfLine

Seconds

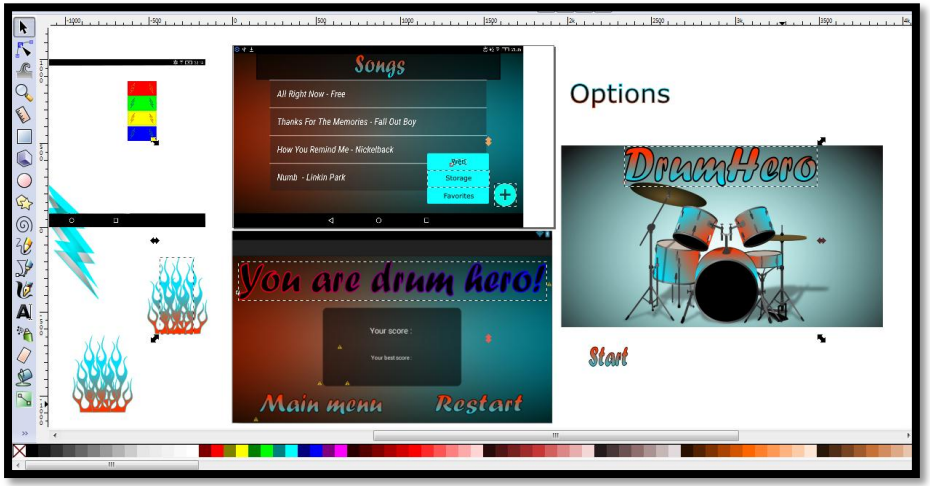
data

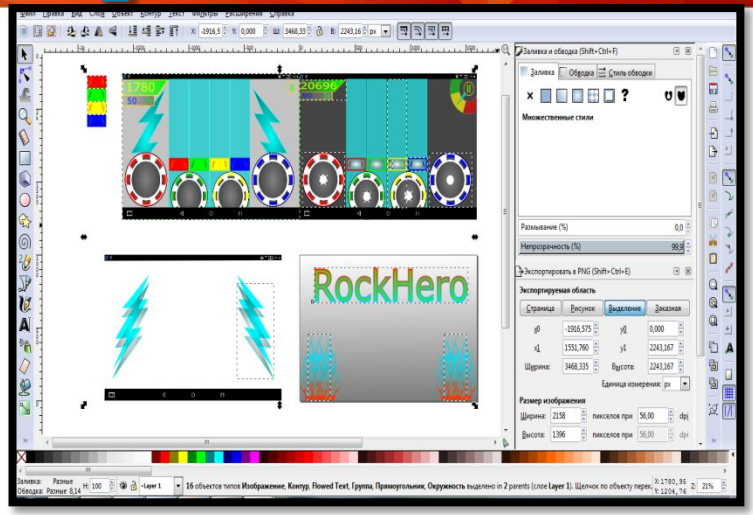


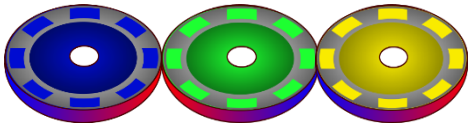
IT ШКОЛА SAMSUNG

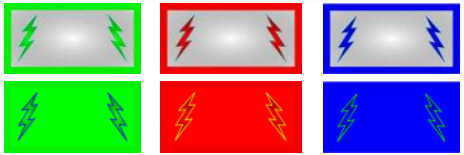
Оформление

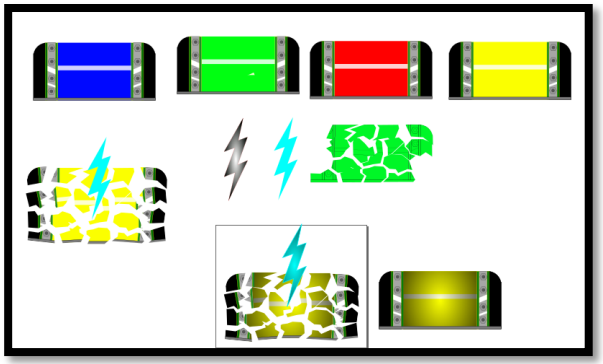
Все изображения рисовались в ручную в графическом векторном редакторе Inkscape.











- blue_crash1
- blue_crash2
- blue_desline
- blue_note2
- destapper3
- green_crash1
- green_crash2
- green_desline
- green_note2
- red_crash1
- red_crash2
- red_desline
- red_note2
- yellow_crash1
- yellow_crash2
- yellow_desline
- yellow_note2

Результат

В конце концов получилась красивая игра во многими любимом жанре и со своей уникальной и очень нужной возможностью – добавление своих песен! Это приложение будет интересно не только многочисленным фанатам музыкального жанра, но оно будет очень полезно тем, кто хочет улучшить чувство ритма или разобрать свою любимую песню по нотам ударной установки. Музыканты же могут использовать данное приложение как абсолютно точное графическое отображение битов на своих современных гаджетах.

*Игра протестирована на большом количестве устройств и корректно работает на экранах с различным разрешением и различной плотностью ☺

Что дальше?

- Добавить больше песен в базу данных.
- Добавить уровни сложностей.
- Улучшить дизайн.
- Добавить возможность выделения ритма для обучения.
- Добавить MIDI редактор для построения графически интересных композиций.
- Загрузить в Google Play.

IT ШКОЛА SAMSUNG

Спасибо за внимание!