

# Snake Exercises

1. Add more apples (so that the screen contains 4 apples).
  - a. Hint: see SnakeView.java and find the use of the function addRandomApple().
2. Play a sound when an apple is hit
  - a. Hint: Identify the location in "SnakeView.java" where a collision is detected (search for the comment "// Look for apples"). A sound is produced e.g. as follows. First import the MediaPlayer class:

```
import android.media.MediaPlayer; // for monster hit sound
```

- b. You can insert it after:

```
import android.view.KeyEvent;  
import android.view.View;  
import android.widget.TextView;
```

- c. You can play a sound with:

```
MediaPlayer mp = new MediaPlayer();  
mp = MediaPlayer.create(this.getContext(),R.raw.hit);  
mp.start();
```

- d. You need to copy a sound file "hit.wav" into the "raw" directory. You can do this e.g. with drag-and-drop (from you desktop into the Eclipse).
3. Replace the "apples" to be eaten by the snake with a different own image to be displayed. We call the own image "Monster" image.
    - a. Include the Monster:

```
private static final int RED_STAR = 1;  
private static final int YELLOW_STAR = 2;  
private static final int GREEN_STAR = 3;  
private static final int MONSTER=4;
```

- b. Update the code to also load the monster image

```
resetTiles(5); // changed for Monster from 4 to 5+
```

```
loadTile(RED_STAR, r.getDrawable(R.drawable.redstar));
loadTile(YELLOW_STAR, r.getDrawable(R.drawable.yellowstar));
loadTile(GREEN_STAR, r.getDrawable(R.drawable.greenstar));
loadTile(MONSTER, r.getDrawable(R.drawable.monster2));
```

- c. Include the monster2.png image into the folder "res/drawable"
- d. Make sure that the Monster is used instead of the apple:

```
private void updateApples() {
    for (Coordinate c : mAppleList) {
        //setTile(YELLOW_STAR, c.x, c.y);
        setTile(MONSTER,c.x,c.y);
    }
}
```

4. Upon a hit, there should be many (or a random number of) new "Monsters" generated.
  - a. Implement a function that calls the function "addRandomApple()" multiple times (with an integer argument that either decides how many times or the maximum number of calls in case of a resulting random number of calls) and make sure, that this function is used upon a hit instead of the function "addRandomApple()".
  - b. Also make sure that ALL old apples/monsters are deleted upon a hit:

```
//mAppleList.remove(c);
//instead now remove ALL apples:
mAppleList.clear();
addMultipleApples(3);
```

5. Modify the function "UpdateApples()" such that "Monsters" and colored "Apples" are both generated.
  - a. You can do this for instance with the "switch" command:

```
private void updateApples() {
    /*for (Coordinate c : mAppleList) {
        //setTile(YELLOW_STAR, c.x, c.y);
        setTile(MONSTER,c.x,c.y);
    }*/
    // use apples and monsters:
    int applecount = mAppleList.size();
    for (int appleindex = 0; appleindex < applecount; appleindex++) {
        Coordinate c = mAppleList.get(appleindex);
        switch(appleindex) {
            case 0:setTile(MONSTER,c.x,c.y);break;
            case 1:setTile(YELLOW_STAR,c.x,c.y);break;
        }
    }
}
```

```
        default:
            setTile(MONSTER,c.x,c.y);break;
        }
    }
}
```

6. Add random single element wall fragments, if the snake hits them he should die.
  - a. Hint: Use the addRandomApple(), updateApples() and add a check for collision in updateSnake() in SnakeView.java
7. Implement a high score “Activity” and switch to it when the player loses.
  - a. Hint: Add a new class to the project called “HighScore” e.g. by right-clicking the project package and selecting “new->class”
  - b. Hint: Make sure the class extends android.app.Activity
  - c. Hint: Create an XML layout file, right-click layout folder “new->file”, name the layout file after the Activity e.g. highscore\_laout.xml
  - d. Hint: In the UI editor add some UI elements
  - e. Hint: Implement the onCreate function in the HighScore Activity

```
package com.example.android.snake;

import android.app.Activity;
import android.os.Bundle;

public class HighScore extends Activity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.highscore_layout);
    }
}
```

- f. Hint: Add the activity to the AndroidManifest.xml file
- g. Hint: In the manifest editor select the Application tab and add the activity in the “Application Nodes” box.
- h. Hint: Set the name of the activity to “HighScore”
- i. Hint: Edit the SnakeView.java file so that when we loose we switch to the HighScore activity.
- j. Hint: See the function setMode(...) function in SnakeView.java
- k. Hint: Possible code to switch activity

```
Intent i = new Intent(getContext(), HighScore.class);
getContext().startActivity(i);
```

- l. Hint: Edit the `highscore_layout.xml`, add a “back” Button and a `ListView` for the high scores.
- m. Hint: Add a `OnClickListener` to the “back” button so that we return to the Snake Activity in the `HighScore.java` file

```
package com.example.android.snake;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ListView;

public class HighScore extends Activity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.highscore_layout);

        Button button = (Button)findViewById(R.id.button_hs_back);
        ListView list = (ListView)findViewById(R.id.highscore_list);

        button.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                startActivity(new Intent(HighScore.this, Snake.class));
                finish();
            }
        });
    }
}
```

- n. Hint: Pass the snake score to the `HighScore` activity in the `setMode(...)` function in `SnakeView.java`

```
Intent i = new Intent(getApplicationContext(), HighScore.class);
i.putExtra("score", mScore);

getApplicationContext().startActivity(i);
```

- o. Hint: Retrieve the score in the `HighScore` activity in the `onCreate(...)` function in `HighScore.java`

```
// Get the score
Bundle b = this.getIntent().getExtras();
long score = b.getLong("score");

Log.i("highscore", "score " + score);
```

- p. Add a list item layout file (list\_item.xml) do define the look of the high score list items. Right-click layout folder “new->file”

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dp"
    android:textSize="16sp" >
</TextView>
```

- q. Add a list of scores and a list adapter to the HighScore.java

```
package com.example.android.snake;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;

public class HighScore extends Activity {

    ArrayList<String> listItems=new ArrayList<String>();
    ArrayAdapter<String> adapter;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.highscore_layout);

        Button button = (Button)findViewById(R.id.button_hs_back);
        ListView list = (ListView)findViewById(R.id.highscore_list);
```

```

// Get the score
Bundle b = this.getIntent().getExtras();
long score = b.getLong("score");

Log.i("highscore", "score " + score);

listItems.add("Score "+score);

adapter = new ArrayAdapter<String>(this, R.layout.list_item, listItems);
list.setAdapter(adapter);

button.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
    startActivity(new Intent(HighScore.this, Snake.class));
    finish();
}
});
}
}

```

r. Read and write the high scores to a file

```

package com.example.android.snake;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;

public class HighScore extends Activity {

    ArrayList<String> listItems=new ArrayList<String>();
    ArrayAdapter<String> adapter;

    static final String filename = "highscore.txt";

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.highscore_layout);

    Button button = (Button)findViewById(R.id.button_hs_back);
    ListView list = (ListView)findViewById(R.id.highscore_list);

    // Get the score
    Bundle b = this getIntent().getExtras();
    long score = b.getLong("score");

    Log.i("highscore", "score " + score);

    readHighscore();

    listItems.add(0, "Score "+score);

    writeHighscore();

    adapter = new ArrayAdapter<String>(this, R.layout.list_item, listItems);
    list.setAdapter(adapter);

    button.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            startActivity(new Intent(HighScore.this, Snake.class));
            finish();
        }
    });
}

private void readHighscore()
{
    try {
        FileInputStream fis;

        fis = openFileInput(filename);

        String s = "";
        int ch;
        while( (ch = fis.read()) != -1)
            s = s + ((char)ch);

        String scores[] = s.substring(1, s.length() - 1).split(", ");

        listItems.addAll(Arrays.asList(scores));
    }
}

```

```
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

private void writeHighscore()
{
    try {
        FileOutputStream fos;

        fos = openFileOutput(filename, this.MODE_PRIVATE);
        String s = listItems.toString();

        fos.write(s.getBytes());
        fos.close();

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}
```

s. Hint: Make the unbeatable high score :)