

Universidad Politecnica de Tecamac

Alumnos:

Estrada Mejía Raúl

Paz Martínez Eduardo

Matriculas:

1319104647

1319104658

Grupo: 1922IS

Materia: programación para móviles

Profesor: Emanuel Torres Servín

Contenido

| | |
|---|----|
| Explica que es el modelo de acceso a datos | 3 |
| Explica que es y cómo funciona la manipulación de datos móviles..... | 3 |
| Describe lo que es y conlleva la persistencia de datos en los dispositivos móviles..... | 3 |
| Explica que y cuales son los mecanismos de tolerancia a fallos | 3 |
| caso resuelto | 4 |
| resultado | 13 |
| Link del repositorio..... | 13 |

Explica que es el modelo de acceso a datos

Es una serie de objetos que le permiten tener acceso y manipular datos mediante programación en bases de datos locales o remotas.

Los objetos de acceso a datos son un patrón de diseño y considerados una buena práctica, la ventaja de usar objetos de accesos a datos es que cualquier objeto de negocio no requiere conocimiento directo del destino final de la información que manipula.

Explica que es y cómo funciona la manipulación de datos móviles

Proporciona una sintaxis no especializada para las instrucciones de INSERT, UPDATE o DELETE cuando se modifican datos de columnas de tipo definidos por el usuario. Las funciones CAST o CONVERT .se utilizan para convertir tipos de datos nativos al tipo definido por el usuario.

Describe lo que es y conlleva la persistencia de datos en los dispositivos móviles

Es una aplicación Android o de cualquier otro tipo consiste en que los datos manipulados por la aplicación sobrevivan a la ejecución de esta en el tiempo, en otras palabras, consiste en almacenar los datos en un medio secundario no volátil para posterior reconstrucción y utilización por lo que tanto son independientes en el tiempo del proceso que los creo.

Preferencias compartidas:

Podemos almacenar y recuperar en el formato clave – valor información como texto, booleanos y números. Lo que lo convierte en potencial para almacenar configuraciones del usuario como: estilo, preferencias, etc.

Almacenar archivos en memoria:

Son soportados por la mayoría de los lenguajes de programación aparte de java, consiste en guardar y recuperar la información en archivos, permite escribir y leer archivos que se encuentren ubicados en la propia memoria interna del dispositivo.

SQLite:

Es un sistema gestor de bases de datos relacional contenida en una relativamente pequeña biblioteca escrita, no es un proceso independiente con el que el programa principal se comuniquen, en lugar de eso, la biblioteca se enlaza con el programa pasando a ser parte integral del mismo.

Explica que y cuáles son los mecanismos de tolerancia a fallos

Las técnicas para conseguir que los sistemas continúen funcionando correctamente, a pesar de fallos en su hardware o errores de software, este tipo de sistemas denomina sistemas tolerantes a fallos.

Conceptos fundamentales en el ámbito de la tolerancia a fallos:

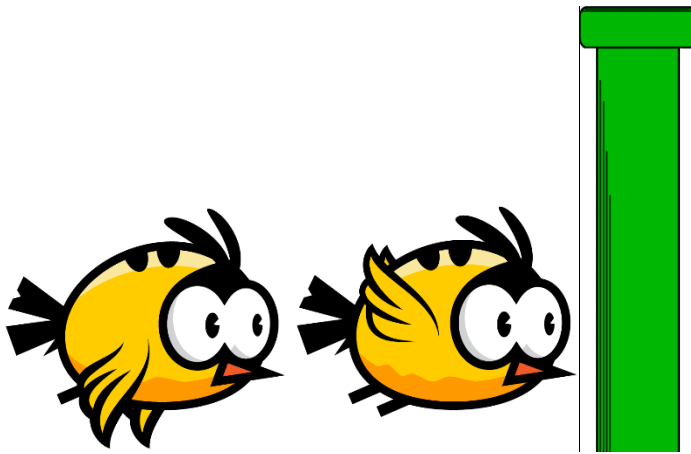
- Llamaremos fallo a cualquier defecto, físico o lógico, en cualquier componente, hardware o software, de un sistema.

- Un error es la manifestación o el resultado de un fallo. Es la consecuencia de un fallo desde el punto de vista de la información.
- Si un error causa un funcionamiento incorrecto del sistema desde el punto de vista externo, es decir, si las consecuencias del fallo trascienden al exterior del sistema, diremos que se ha producido una avería. Las averías se producen en el universo del usuario.

caso resuelto

para este proyecto se desarrolló el caso del juego popular (flappy bird), donde se hizo uso de los anteriores conceptos dentro del software.

- 1- Como primer paso es preparar las imágenes



- 2- Obtener fondo para el juego



```
package com.example.flappybirdbasic;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Display;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    public static TextView txtScore, txtScoreFin;
    public static RelativeLayout rlFin;
    private GameView gv;
    public static Button btnIniciar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        DisplayMetrics dm = new DisplayMetrics();
        this.getWindowManager().getDefaultDisplay().getMetrics(dm);
        Constraints.SCREEN_HEIGHT = dm.heightPixels;
        Constraints.SCREEN_WIDTH = dm.widthPixels;
        setContentView(R.layout.activity_main);
        txtScore = findViewById(R.id.txtScore);
        txtScoreFin = findViewById(R.id.txtScoreFin);
        btnIniciar = findViewById(R.id.btnIniciar);
        rlFin = findViewById(R.id.rlFin);
        gv = findViewById(R.id.gv);
        btnIniciar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                gv.setIniciar(true);
                txtScore.setVisibility(View.VISIBLE);
                btnIniciar.setVisibility(View.INVISIBLE);
            }
        });
        rlFin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                btnIniciar.setVisibility(View.VISIBLE);
                rlFin.setVisibility(View.INVISIBLE);
                gv.setIniciar(false);
                gv.reset();
            }
        });
    }
}
```

Clase principal diseño

```
package com.example.flappybirdbasic;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Display;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.RelativeLayout;
import android.widget.TextView;
//se preparan las imagenes que se encuentras en la carpeta drawble
//set background for game

public class MainActivity extends AppCompatActivity {
    public static TextView txtScore, txtScoreFin;
    public static RelativeLayout rlFin;
    private GameView gv;
    public static Button btnIniciar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,WindowManager.LayoutParams.FLAG_FULLSCREEN);
        DisplayMetrics dm = new DisplayMetrics();
        this.getWindowManager().getDefaultDisplay().getMetrics(dm);
        Constrants.SCREEN_HEIGHT = dm.heightPixels;
        Constrants.SCREEN_WIDTH = dm.widthPixels;
        setContentView(R.layout.activity_main);
        txtScore = findViewById(R.id.txtScore);
        txtScoreFin = findViewById(R.id.txtScoreFin);
        btnIniciar = findViewById(R.id.btnIniciar);
        rlFin = findViewById(R.id.rlFin);
        gv = findViewById(R.id.gv);
        btnIniciar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                gv.setIniciar(true);
                txtScore.setVisibility(View.VISIBLE);
                btnIniciar.setVisibility(View.INVISIBLE);
            }
        });

        rlFin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                btnIniciar.setVisibility(View.VISIBLE);
                rlFin.setVisibility(View.INVISIBLE);
                gv.setIniciar(false);
                gv.reset();
            }
        });
    }
}
```

```
package com.example.flappybirdbasic;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.os.Handler;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class GameView extends View {

    private Bird bird;
    private Handler handler;
    private Runnable r;
    private ArrayList<Tuberias> arrTuberias;
    private int sumaTuberia, distancia;
    private int score;
    private boolean iniciar = false;

    public GameView(Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        score = 0;
        iniciar = false;
        initBird();
        initTuberia();
        handler = new Handler();
        r = new Runnable() {
            @Override
            public void run() {
                invalidate();
            }
        };
    }

    //Posicionar tuberias
    private void initTuberia() {
        sumaTuberia = 6;
        distancia = 300*Constants.SCREEN_HEIGHT/1920;
        arrTuberias = new ArrayList<>();
        for (int i = 0; i < sumaTuberia; i++) {
            if(i < sumaTuberia/2) {
                this.arrTuberias.add(new
Tuberias(Constants.SCREEN_WIDTH+i*((Constants.SCREEN_WIDTH+200*Constan
ts.SCREEN_WIDTH/1080)/(sumaTuberia/2)),
                0, 200*Constants.SCREEN_WIDTH/1080,
Constants.SCREEN_HEIGHT/2));
                this.arrTuberias.get(this.arrTuberias.size()-
1).setBm(BitmapFactory.decodeResource(this.getResources(),
R.drawable.pipe2));
                this.arrTuberias.get(this.arrTuberias.size()-
```

```

1).aleatorioY();
    }else {
        this.arrTuberias.add(new Tuberias(this.arrTuberias.get(i-
sumaTuberia/2).getX(), this.arrTuberias.get(i-sumaTuberia/2).getY()
+this.arrTuberias.get(i-sumaTuberia/2).getHeight() +
this.distancia, 200*Constraints.SCREEN_WIDTH/1080,
Constraints.SCREEN_HEIGHT/2));
        this.arrTuberias.get(this.arrTuberias.size()-
1).setBm(BitmapFactory.decodeResource(this.getResources(),
R.drawable.pipel));
    }
}

private void initBird() {
    bird = new Bird();
    bird.setWidth(100*Constraints.SCREEN_WIDTH/1080);
    bird.setHeight(100*Constraints.SCREEN_HEIGHT/1920);
    bird.setX(100*Constraints.SCREEN_WIDTH/1080);
    bird.setY(Constraints.SCREEN_HEIGHT/2-bird.getHeight()/2);
    ArrayList<Bitmap> arrBms = new ArrayList<>();
    arrBms.add(BitmapFactory.decodeResource(this.getResources(),
R.drawable.bird1));
    arrBms.add(BitmapFactory.decodeResource(this.getResources(),
R.drawable.bird2));
    bird.setArrBms(arrBms);
}

public void draw(Canvas canvas) {
    super.draw(canvas);
    if (iniciar) {
        bird.draw(canvas);
        for (int i = 0; i < sumaTuberia; i++) {
            if(bird.getRect().intersect(arrTuberias.get(i).getRect())
|| bird.getY()-bird.getHeight() < 0 || bird.getY() >
Constraints.SCREEN_HEIGHT) {
                Tuberias.velocidad = 0;

MainActivity.txtScoreFin.setText(MainActivity.txtScore.getText());
                MainActivity.r1Fin.setVisibility(VISIBLE);
            }
            if (this.bird.getX()+this.bird.getWidth() >
arrTuberias.get(i).getX() + arrTuberias.get(i).getWidth()/2
                && this.bird.getX()+this.bird.getWidth() < -
arrTuberias.get(i).getX() + arrTuberias.get(i).getWidth()/2 +
Tuberias.velocidad
                && i < sumaTuberia/2) {
                    score++;
                    MainActivity.txtScore.setText(""+score);
            }
            if (this.arrTuberias.get(i).getX() < -
arrTuberias.get(i).getWidth()) {
this.arrTuberias.get(i).setX(Constraints.SCREEN_WIDTH);
                if (i < sumaTuberia/2) {
                    arrTuberias.get(i).aleatorioY();
                }else {

```



```
arrTuberias.get(i).setY(this.arrTuberias.get(i-  
sumaTuberia/2).getY()  
+this.arrTuberias.get(i-  
sumaTuberia/2).getHeight() + this.distancia);  
}  
this.arrTuberias.get(i).draw(canvas);  
}  
}else {  
    if (bird.getY() > Constrants.SCREEN_HEIGHT/2) {  
        bird.setDrop(-15*Constrants.SCREEN_HEIGHT/1920);  
    }  
    bird.draw(canvas);  
}  
handler.postDelayed(r, 10);  
}  
  
@Override  
public boolean onTouchEvent(MotionEvent event) {  
    if (event.getAction() == MotionEvent.ACTION_DOWN) {  
        bird.setDrop(-15);  
    }  
    return true;  
}  
  
public boolean isIniciar() {  
    return iniciar;  
}  
  
public void setIniciar(boolean iniciar) {  
    this.iniciar = iniciar;  
}  
  
public void reset() {  
    MainActivity.txtScore.setText("0");  
    score = 0;  
    initTuberia();  
    initBird();  
}  
}
```

Clase base de datos

```
package com.example.flappybirdbasic;

import android.graphics.Bitmap;
import android.graphics.Rect;

public class BaseObject {
    protected float x, y;
    protected int width, height;
    protected Rect rect;
    protected Bitmap bm;

    public BaseObject() {
    }

    public BaseObject(float x, float y, int width, int height) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }

    public float getX() {
        return x;
    }

    public void setX(float x) {
        this.x = x;
    }

    public float getY() {
        return y;
    }

    public void setY(float y) {
        this.y = y;
    }

    public int getWidth() {
        return width;
    }

    public void setWidth(int width) {
        this.width = width;
    }

    public int getHeight() {
        return height;
    }

    public void setHeight(int height) {
        this.height = height;
    }

    public Bitmap getBm() {
        return bm;
    }
}
```

```
}

public void setBm(Bitmap bm) {
    this.bm = bm;
}

public Rect getRect() {
    return new Rect((int)this.x, (int)this.y, (int)this.x+this.width,
(int)this.y+this.height);
}

public void setRect(Rect rect) {
    this.rect = rect;
}
}
```

Clase pájaro

```
package com.example.flappybirdbasic;

import android.graphics.Bitmap;
import android.graphics.Rect;

public class BaseObject {
    protected float x, y;
    protected int width, height;
    protected Rect rect;
    protected Bitmap bm;

    public BaseObject() {
    }

    public BaseObject(float x, float y, int width, int height) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }

    public float getX() {
        return x;
    }

    public void setX(float x) {
        this.x = x;
    }

    public float getY() {
        return y;
    }

    public void setY(float y) {
        this.y = y;
    }

    public int getWidth() {
```

```
        return width;
    }

    public void setWidth(int width) {
        this.width = width;
    }

    public int getHeight() {
        return height;
    }

    public void setHeight(int height) {
        this.height = height;
    }

    public Bitmap getBm() {
        return bm;
    }

    public void setBm(Bitmap bm) {
        this.bm = bm;
    }

    public Rect getRect() {
        return new Rect((int)this.x, (int)this.y, (int)this.x+this.width,
(int)this.y+this.height);
    }

    public void setRect(Rect rect) {
        this.rect = rect;
    }
}
```

Clase restricciones

```
package com.example.flappybirdbasic;

public class Constrants {
    public static int SCREEN_WIDTH;
    public static int SCREEN_HEIGHT;
}
```

Clase tuberías

```
package com.example.flappybirdbasic;

import android.graphics.Bitmap;
import android.graphics.Canvas;

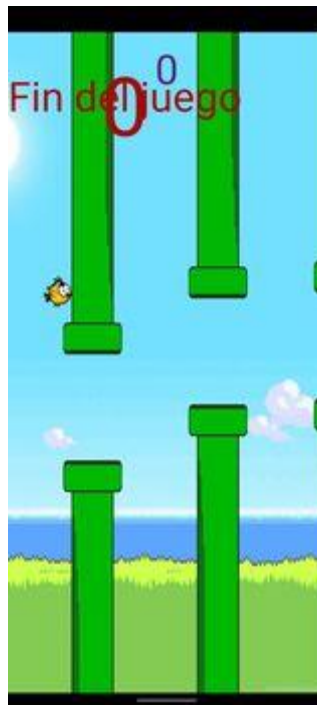
import java.util.Random;

public class Tuberias extends BaseObject {

    public static int velocidad;
```

```
public Tuberias(float x, float y, int width, int height) {  
    super(x, y, width, height);  
    velocidad = 10 * Constraints.SCREEN_WIDTH/1000;  
}  
  
public void draw(Canvas canvas) {  
    this.x-=velocidad;  
    canvas.drawBitmap(this.bm, this.x, this.y, null);  
}  
  
public void aleatorioY() {  
    Random r = new Random();  
    this.y = r.nextInt((0+this.height/4)+1)-this.height/4;  
}  
  
@Override  
public void setBm(Bitmap bm) {  
    this.bm = Bitmap.createScaledBitmap(bm, width, height, true);  
}  
}
```

Resultado



Link del repositorio

<https://github.com/neutroREM/flappyBirdBasic.git>