

GTDM Clicker

TRABAJO FINAL

Aplicaciones y Usabilidad



Daniel Cabañero Martínez

Marcos Rome García

Javier Ruiz Ferrandis

Sara Oliete López

02 de Julio 2023

ÍNDICE

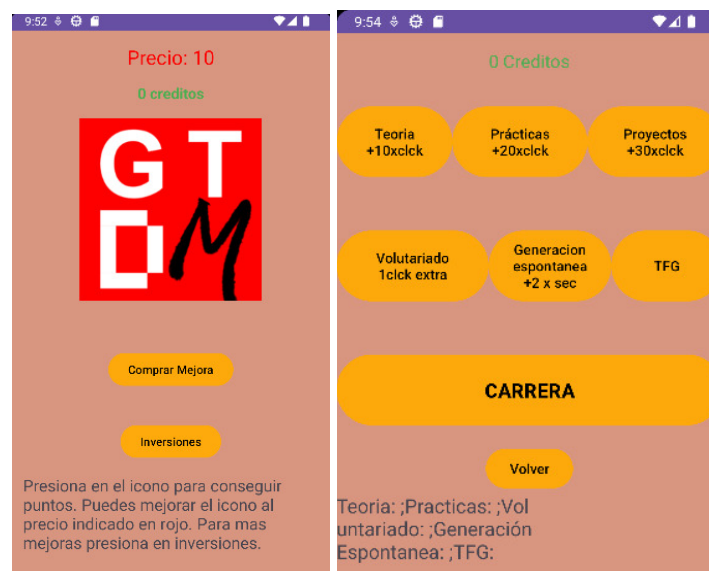
Manual de la App.....	2
Interfaz.....	2
Código.....	2
MainActivity.....	2
Upgrade Activity.....	4
Envío de datos.....	5
Conclusión.....	5

Manual de la App

Para el desarrollo de este proyecto hemos decidido diseñar una aplicación inspirada en los juegos “Clicker”. En los cuales, el jugador tiene que aumentar su puntuación apretando un objeto en la pantalla. Con dicha puntuación, el jugador puede adquirir mejoras que le ayuden a conseguir los puntos en los clicks.

Interfaz

La interfaz la hemos dividido en dos ventanas. Una donde está el logotipo de GTDM (el cual habrá que presionar) y dos botones: la mejora del icono y la tienda que en este caso se denomina “Inversiones”. En la tienda, el jugador podrá acceder a esas mejoras que ayudarán a que el jugador pueda aumentar sus puntos.



Código

MainActivity

La clase MainActivity almacena los clicks que el usuario consigue con el pulsador. También imprime en dos TextView la puntuación (los clicks) y el mínimo de puntos que el jugador necesita para poder conseguir la mejora del multiplicador. Para ello hemos definido tres funciones: `calcularCostoMejora()`, `actualizarContador()` y `actualizarPuntos()`.

```

// Calcula el costo de la mejora actual
private int calcularCostoMejora() {
    return 10 * (int) Math.pow(2, contador);
}

// Actualiza el texto del contador
private void actualizarContador() {
    textoContador.setText(String.valueOf(mejora));
}

// Actualiza el texto de los puntos
private void actualizarPuntos() {
    textoPuntos.setText(String.valueOf(puntos));
}

```

Por lo que hemos necesitado definir otros tres eventos: el primero, `botonClick()` el cual se encarga de ejecutar las funciones de actualizar puntos y contador. El segundo, un `.setOnClickListener()` para asignar directamente el evento al botón de la mejora. El cual comprueba los puntos que necesitas para obtener dicha mejora y los clicks que el jugador tiene acumulados. Si esos clicks acumulados son igual o mayor que los puntos necesarios, se aplica la mejora, se modifican esos puntos necesarios y se actualiza el contador de clicks. El tercer evento es el que iniciará la actividad de `UpgradeActivity.java`, que es la tienda.

```

botonMejora.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Comprueba si hay suficientes puntos para comprar la mejora
        if (puntos >= calcularCostoMejora()) {
            // Resta los puntos necesarios
            puntos -= calcularCostoMejora();
            // Incrementa la mejora en 1
            contador++;
            mejora=calcularCostoMejora();
            // Actualiza el contador y los puntos
            actualizarContador();

            actualizarPuntos();
        }
    }
});

```

```

// Método para el botón de clic para aumentar el contador
public void botonClic(View view) {
    if (vol == true) {
        puntos += ((1 + contador * 2) + extra)*2; // Aumenta en 1 más 2 por cada 2 mejoras compradas
    } else {
        puntos += (1 + contador * 2) + extra; // Aumenta en 1 más 2 por cada 2 mejoras compradas
    }
    actualizarContador();
    actualizarPuntos();
}

```

Upgrade Activity

La clase UpgradeActivity es una actividad aparte que se puede acceder desde la actividad principal. Almacena los onclick que permiten al usuario mejorar su eficiencia al apretar la imagen.

En esta clase tenemos definidos ocho OnClick y dos TextView:

El primer OnClick está en el botón “Teoria”. Cuando este se presione, se consigue que el usuario gane +10 créditos cada vez que este apriete el pulsador.

El segundo, es el botón “Practicas”. El cual hace que, una vez haya sido presionado, el marcador suba +20 créditos cada vez que el usuario apriete el pulsador.

El botón “Proyecto” es el tercer OnClick. Y cuando es apretado hace que a partir de ese momento, cuando el usuario apriete el pulsador gane +30 créditos.

```
click1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(c1==false){  
            extra+=10;  
            sobras-=1;  
            actualizarPuntos();  
            c1=true;  
        }  
    }  
});
```

```
click2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(c2==false){  
            extra+=20;  
            sobras-=2;  
            actualizarPuntos();  
            c2=true;}  
    }  
});
```

```
click3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(c3==false){extra+=30;  
            sobras-=3;  
            actualizarPuntos();  
            c3=true;}  
    }  
});
```

El cuarto OnClick es el botón “Voluntariado”. Que al presionarlo bonifica al jugador multiplicando por dos la cantidad de puntos que gana cada vez que apriete el pulsador.

El quinto es el botón “GeneracionEspontanea”. Para que cada segundo el usuario gane 5 créditos.

```

v.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (vol==false) {
            vol = true;
        }
    }
});

```

```

gen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (GE==false) {
            GE = true;
        }
    }
});

```

El botón “TFG” es el sexto OnClick. Cuando este botón se aprieta, tendrás la posibilidad de acabar el juego.

```

T.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int c=40000;
        if (sobras>=c && TFG==false) {
            TFG = true;
            carrera=true;
            sobras-=c;
            actualizarPuntos();
        }
    }
});

```

El séptimo es el botón “Carrera”. Cuando es apretado el usuario es llevado a una pantalla de fin.

```

C.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (carrera==true) {
            setContentView(R.layout.win);
        }
    }
});

```

Por último, el octavo OnClick es el botón “botonVolver”. Y se encarga de llevar al usuario a la “MainActivity”.

```

botonVolver.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Vuelve a la actividad principal
        Intent intentoRetorno = new Intent();
        Bundle b = new Bundle();
        b.putInt("sobra", sobras);
        b.putBoolean("teoria", c1);
        b.putBoolean("practicas", c2);
        b.putBoolean("proyectos", c3);
        b.putInt("Extra", extra);
        intentoRetorno.putExtras(b);
        setResult( Activity.RESULT_OK, intentoRetorno );
        finish();
    }
});

```

En cuanto a los TextView, el primero (ubicado arriba en la pantalla) indica cuántos créditos tiene el usuario.

Mientras que en el segundo (ubicado debajo del anterior) explicamos cómo funciona esta pantalla.

Envío de datos

Para que ambas actividades se comuniquen correctamente hacemos uso del lanzamiento de actividad con espera de resultado.

En nuestro caso tenemos todas las variables compartidas definidas en ambas actividades. Luego MainActivity llama a UpgradeActivity y le envía todos los valores de las variables con sus alias. Upgrade actualiza sus variables con los valores recibidos en su onCreate.

Una vez terminamos en Upgrade, al presionar “volver”, todos los valores se le envían a MainActivity como resultado y esta actualiza los suyos.

Conclusión

Gracias a este proyecto hemos podido comprobar lo útil que es Android Studio para el diseño de aplicaciones y como hasta la aplicación más sencilla requiere una buena organización y bastante trabajo por detrás de lo que luego todos ven al abrir la app.