

Práctica 3 – Gymkana

Características necesarias del proyecto

Este proyecto se ha realizado con **Android Studio**. Es compatible en dispositivos con **Android 4.3** o superior.

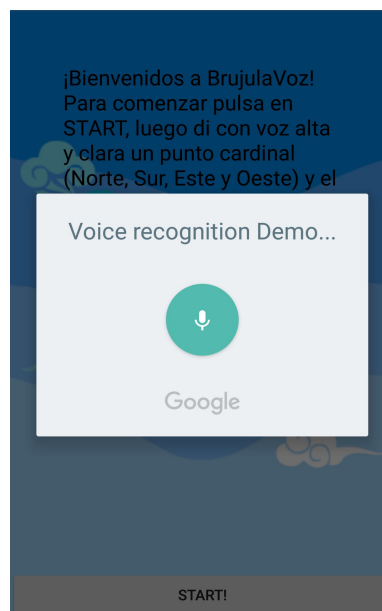
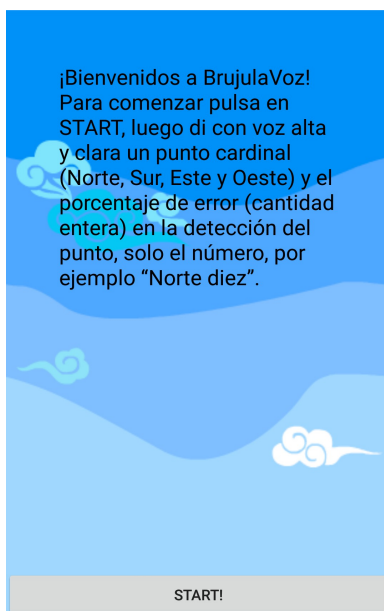
1. BrujulaVoz

Introducción

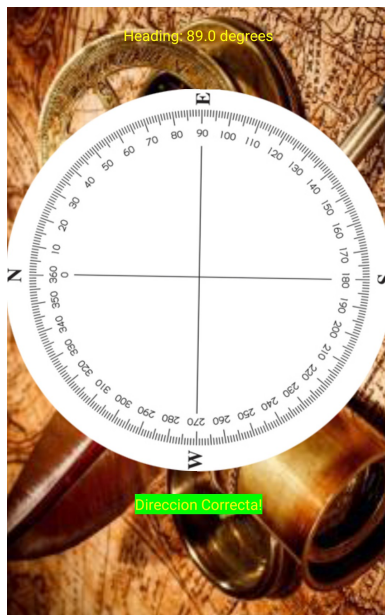
Con la **appFotoVoz** se debe identificar mediante interfaz vocal un punto cardinal (Norte, Sur, Este y Oeste) y el porcentaje de error (cantidad entera) en la detección del punto, solo el número, por ejemplo: “Norte diez”. Una vez reconocido el patrón vocal se debe mostrar una brújula para que el usuario se oriente en la dirección indicada, cuando lo realice se le debe indicar que ya se ha conseguido.

Desarrollo de la aplicación

Para realizar esta aplicación se ha añadido a la actividad principal del proyecto un botón **START** donde al pulsarlo nos aparecerá un pop-up para que digamos algo y lo reconozca, para posteriormente decirle el punto cardinal y el porcentaje de error:



Una vez que le hayamos dicho el punto cardinal y el porcentaje de error se nos abrirá una nueva actividad donde hay una brújula. En la parte de arriba, en heading, nos muestra los grados en los que nos encontramos. En las siguientes figuras se ve puede ver un ejemplo de lo que ocurre si nos encontramos de forma correcta en la dirección que hemos indicado anteriormente (foto de la izquierda) o si no nos hemos colocado bien (foto de la derecha):



Una vez llegado a la posición se le indica al usuario que lo ha conseguido.

2. Gesto de foto

Introducción

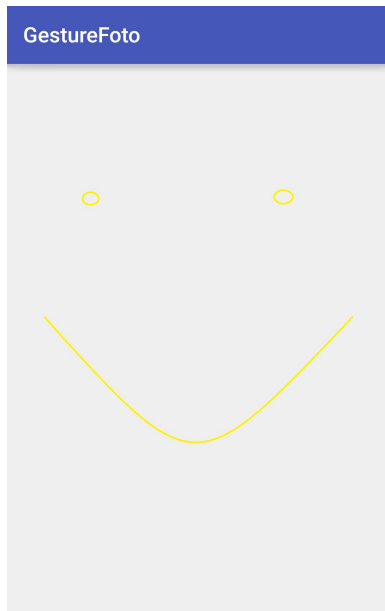
Se pedirá una **appGestosFoto** que reconozca un patrón de gestos sobre pantalla de vuestra elección. Una vez detectado el patrón se debe ejecutar la aplicación de cámara y realizar automáticamente una foto transcurridos 3 segundos. Los gestos a detectar son similares a los que se realizan con la pantalla de bloqueo Android, aquí podéis ver algunos ejemplos, cuanto más complejo sea el que se detecta mejor puntuación tendrá.

Desarrollo de la aplicación

Esta aplicación tendrá una pantalla principal con un botón, START, que se pulsará para comenzar a realizar el gesto:

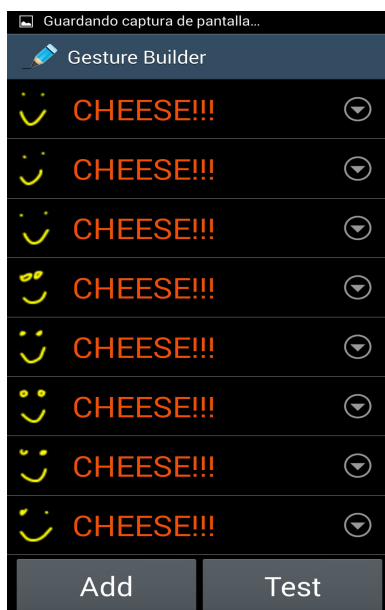


Se abrirá una nueva actividad en blanco donde podremos realizar un dibujo, en este caso para poder comenzar a realizar la foto, hay que dibujar una cara sonriente, mas o menos como la que se ve en amarillo en la siguiente imagen:



Realizado esto se abrirá automáticamente la cámara frontal y comenzará el temporizador en 3 segundos para realizar la foto.

Para establecer el patrón de selección de gesto hemos usado la aplicación Gesture Builder:



3. PuntoGPSQR

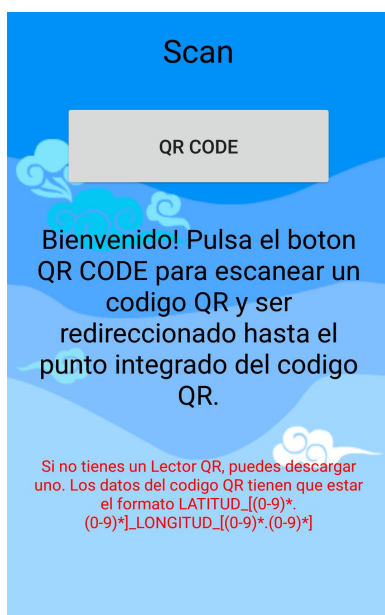
Introducción

Se facilitarán las coordenadas de una serie de puntos GPS (latitud/longitud) mediante códigos QR a la **appGPSQR** que iniciará automáticamente la navegación GPS hacia dicho punto, debe guardar el recorrido realizado y mostrarlo en un mapa al finalizar el mismo.

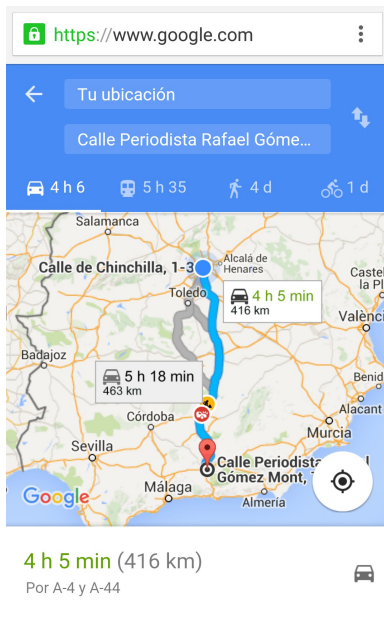
Desarrollo de la aplicación

Para realizar esta aplicación se ha añadido a la actividad principal del proyecto un botón donde al pulsarlo pueden ocurrir dos cosas:

- En caso de no tener ningún tipo de escáner de códigos QR nos llevará a una de éstas en Play Store para descargarla ya que es necesaria.
- En caso de si tener un escáner QR (o haberla instalado como se indica anteriormente), se ejecutará la aplicación a la cual tenemos que proporcionarle la imagen QR en formato `LATITUD_[(0-9)*.(0-9)*]_LONGITUD_[(0-9)*.(0-9)*]`



Tras haberle proporcionado un código QR que sea correcto se nos dará opción de elegir con que aplicación queremos abrir los datos de ubicación. Si elegimos un navegador veremos algo como esto (depende de la ubicación del usuario) :



4. Punto movimiento sonido

Introducción

Se pedirá una **appMovimientoSonido** que reconozca un patrón de movimientos de vuestra elección usando el giroscopio y/o el acelerómetro, una vez detectado el patrón, se reproducirá un sonido. En este caso se ha optado por un sable láser de Star Wars que suena de forma diferente en función de la velocidad del movimiento del dispositivo (despacio o fuerte).

Desarrollo de la aplicación

Para realizar esta aplicación primero nos encontramos en la pantalla principal donde pulsaremos el botón START para comenzar:



Se nos abrirá una nueva actividad donde veremos un sable con un botón. Para que funcione correctamente debemos pulsar el botón para que se encienda (incluye sonido de encendido). Una vez encendido si realizamos un movimiento suave hará el ruido correspondiente al de un sable láser pero no muy alto, en cambio si movemos más rápido el dispositivo, el sonido será diferente y en un tono mayor.



5. Punto Sorpresa

Introducción

La **appSorpresa** hará uso de otro sensor no cubierto en ninguna de las aplicaciones anteriores: NFC, Android wear...

Desarrollo de la aplicación

Al principio se inicia la aplicación y nos muestra la pantalla de inicio donde nos sugiere como comenzar a usar la aplicación:



Tras pulsar en el botón START nos aparecerá un botón, para poder efectuar el evento que tenemos asignado, en este caso un sonido, tenemos que acercar la mano a la pantalla para que el sensor de proximidad lo detecte a cierta distancia y se escuche el sonido:



Problemas encontrados

- En el PuntoGPSQR no se ha podido guardar el camino que se recorre ya que no se ha encontrado información suficiente para saber hacerlo.
- En GestureFoto al principio se intentó usar un intent pero no funcionaba con el temporizador por lo que se tuvo que usar la aplicación de la cámara y no el intent.
- En GestureFoto se iba a realizar el gesto para desbloquear la cámara como los que se realizan para desbloquear la pantalla de desbloqueo, por lo se realizó un gesto más libre, en este caso un sonrisa.

Referencias

[STACK OVERFLOW] <https://www.stackoverflow.com>

[How to implement voice recognition] <http://www.jameselsey.co.uk/blogs/techblog/android-how-to-implement-voice-recognition-a-nice-easy-tutorial/>

[How to Create a Compass] <https://www.codeofaninja.com/2013/08/android-compass-code-example.html>

[Android Asset Studio] <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html#foreground.space.trim=1&foreground.space.pad=0&foreColor=607d8b%2C0&crop=0&backgroundShape=square&backColor=ffffff%2C100&effects=none>

[Using Proximity Sensor] <http://karanbalkar.com/2014/07/using-proximity-sensor-in-android/>

[Implementing Android Custom Gesture Recognition]
http://www.techotopia.com/index.php/Implementing_Android_Custom_Gesture_Recognition_with_Android_Studio