Instituto Tecnológico y de Estudios Superiores de Occidente.

ITESO

Universidad Jesuita de Guadalajara

Implementar manejo de estados de uno o mas widgets utilizando BloC

Materia: PROGRAMACION DE DISPOSITIVOS MOVILES

Profesor: Francisco Javier Camacho Gil

Fecha: 3 de marzo de 2021

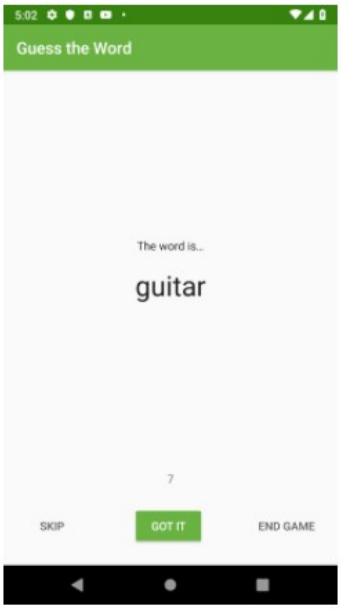
Autor(es): Armando Emmanuel Correa Amorelli

Introducción

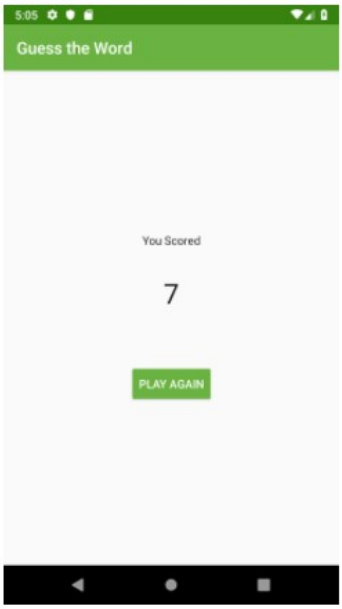
El objetivo del ejercicio realizar una app que consiste en una sola pantalla donde se debe tratar de adivinar una palabra en base a una sugerencia y se lleva la cuenta de aciertos misma que se va mostrando. Utilizando BLoC para manejar los estados y separar la lógica de la vista.

Screenshot 1	1. Dos textos centrados y en forma de columna. 2. Un botón verde con texto blanco centrado en una fila casi hasta abajo de la pantalla.
Screenshot 2	1. Mismos dos textos centrados y en forma de columna. 2. Un texto que indica el puntaje de palabras adivinadas en forma de columna con la fila que tiene los botones. 3. Un botón verde con texto blanco y dos botones blancos separados dentro de una fila casi hasta abajo de la pantalla.
Screenshot 3	1. Mismos dos textos centrados y en forma de columna y un botón verde con texto blanco con padding hacia arriba.

Title screen



Game screen



Score screen

Desarrollo

Para el desarrollo de esta actividad, se partío de un trabajo previamente ralizado en clase, el cual ya contaba con la mayoría de los elementos graficos y lógicos para el correcto funcionamiento.

Las adiciones que se ralizaron a la logica fueron las siguientes:

- adición de RestartEvent:
  Este evento se añadió para cumplir el requerimiento de reiniciar contadores y listas al momento de presionar el boton de “play again”.

```
if (event is RestartEvent) {
      listSize = list1.length;
      count = 0;
      index = 0;
      list1.shuffle();
      list2.shuffle();
      yield HomeInitial();
    }
```

Por otro lado, los cambiós realizados en la clase Home, que es nuestra unica ventana, fue el definir las adiciones de eventos como funciones. Esto con el proposito de pasarlas como argumentos a los widgets externos creados para cada ventana.

```
Function _skip(BuildContext context) {
  BlocProvider.of<HomeBloc>(context).add(SkipEvent());
}
Function _gotIt(BuildContext context) {
  BlocProvider.of<HomeBloc>(context).add(GotEvent());
}
Function _endGame(BuildContext context) {
  BlocProvider.of<HomeBloc>(context).add(EndEvent());
}
Function _reStart(BuildContext context) {
  BlocProvider.of<HomeBloc>(context).add(RestartEvent());
}
Function _startGame(BuildContext context) {
  BlocProvider.of<HomeBloc>(context).add(StartEvent());
}
```

Ya definidas estas funciones, se crearon 3 widgets distintos con los modelos con los que se contaba. Estos fueron los siguentes:

- GameOverDisplay
  Recibe una funcion, un string, un int y el context.
  Este consta de un stack alineado al centro. El cual muestra el un texto seguido de un contador, encapsulados en un elemento column, dentro de un center. A su vez cuenta con un Material button encapsulado en un elemento positioned.
- InGameDisplay
  Recibe 3 funciones, 2 string, un int y el context.
  Al igual que el anterior widget tambien está conformado por un stack, con encapsula una columna centrada donde se despliegan los 2 strings recibidos, el int recibido, está en un elemento positioned, igual que los 3 botones que hacen uso de las funciones recibidas.
- StartDisplay
  Recibe una funcion y el context.
  Este consta de un stack con un elemento align que muestra el texto de bienvenida y un elemento positioned que encapsuala un button.

Conclusiones

Mediante esta tarea se puede reforzar el uso de BloC, al igual que la posible utilizacion de la clase Function, con la finalidad de pasar funciones dependientes de elementos externos al widget personalizado que se desea implementar. Tambien se puede obsrvar que al realizar esto se debe pasar el context como argumento, con la finalidad de no ejecutar las funciones creadas cuando se ejecute el código donde se encontraba.

Observaciones

Posiblemente exista una manera distinta de pasar “funciones” a widgets personalizados apra que estos ejecuten cierta tarea al ser presionados.

Repositorio
Video

Bibliografía

[1]"How to pass functions to child widgets in Flutter - Kindacode", *Kindacode*, 2020. [Online]. Available: https://www.kindacode.com/article/how-to-pass-functions-to-child-widgets-in-flutter/. [Accessed: 04- Mar- 2021].

[2]V. Vijayan, "BLoc Pattern in Flutter explained with Real Example", *Medium*, 2020. [Online]. Available: https://vipinvijayannair.medium.com/bloc-pattern-in-flutter-explained-with-real-example-f1af2568d32d. [Accessed: 04- Mar- 2021].

Criterio de evaluación ponderado.

Criterio	**Puntos obtenibles**	**Mis puntos obtenidos**	**Observaciones**
Toda la lógica y procesamiento se lleva a cabo en el BLoC*	40	40	
La vista muestra widgets en base a los estados del BLoC que contienen datos necesarios para mostrar	20	20	
El flujo de la app se sigue correctamente gracias a la implementación del BLoC con sus estados y eventos	20	20	
Reporte de tareas incluye link a repo de git	10	10	
Se incluye link a vídeo de app funcionando	10	10	