CODEV

TEXT HANDLER

MEng Year 3

Department of Electronics
University of York

Software Engineering Group 4

Contents

1. Text Handler

1.1 Requirements

The service provider will design an extension of a text view that can be passed variables. The view will have the capability to scroll both vertically and horizontally if the size of the displayed text exceeds the boundaries of the view. The extension should match the parent constraint sizing. Features should be included to manipulate the size, weight, and colour of the text, in addition to the background colour of the view (with transparent as a defaulted setting) and the option to customise the font of the text displayed in the view.

1.2 Text Handler Specification

The text handler developed by CoDev extends the functionality of the standard TextView included in the default Android library. It incorporates a parent ScrollView which holds the container for the TextView.

The text handler allows use of any fonts found through the google fonts API to be downloaded and displayed on the fly, with the option to change the weight, font size and font colour at any point.

The text handler also allows the dimensions and position of the containing box to be set, with the background of the container being transparent unless a colour is specified.

2. Setting up the Text Handler

To use the new text handler, simply add the PresentationTextView class to the application and call it within an activity like you would with a standard Android TextView. The height and width of the containing slide will need to be specified during creation so that the position and dimensions of the view are scaled to the slide.

If the PresentationTextView class is being used within a parent scroll view, lines 56 to 63 (below) of the class will need to be uncommented due to the way scroll views interact with each other.

```
setOnTouchListener(new OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
       v.performClick();
       v.getParent().requestDisallowInterceptTouchEvent(true);
       return false;
    }
});
```