Technical Report

Multimedia Design and Applications - Assignment 1

CANDIDATE NO.: 118440

WORD COUNT: 1004

1. INTRODUCTION

This is a technical report that aims at giving a brief introduction to the implementation and design of a SVG JavaScript animated clock, which is an assignment for the Multimedia Design and Applications module. The report starts by introducing the media elements used, followed by explaining how and why the functionalities are implemented.

2. MEDIA ELEMENTS

2.1 VECTOR GRAPHICS

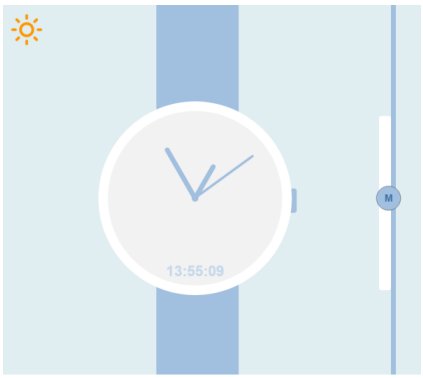

Figure 2.1, Outlook of the watch design.

A number of vector graphics are used to produce the clock graphic display, using the basic primitive shapes from Raphaël[1] library. As the watch is designed based on a simple and clean colour scheme, simple shades are used to create a clean outlook. The surface of the watch is created by concentric circle, and the strap is made by a single rectangle.

2.2 IMAGES

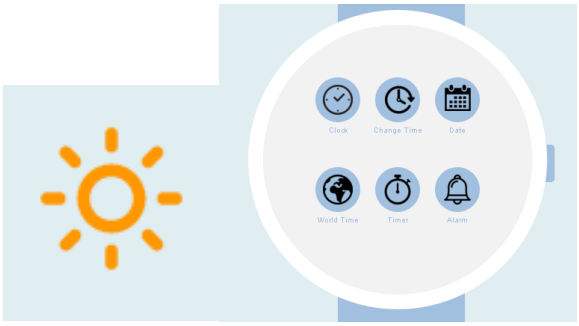

Figure 2.2, Icons in the watch. [2]

Due to the limitation of the primitive vector shapes provided by Raphaël library, more complicated shapes is hard to draw out. Instead of building shapes from the library, images are used to display icons in the watch. Figure 2.2 shows all images in use for the watch, including the orange sun logo on the top left of the watch application, and all the icons [2] indicating buttons in the main menu. Icons are used mainly to increase the ease of use and as a visual enhancement of the user-interface.

3.3 SOUND

In the alarm functionality of the clock which will be explained in the following section, an alarm sound track [3] is played when the alarm is triggered.

3. JS FUNCTIONALITIES

3.1 CLOCK FUNCTIONALITY

To make the clock updates every second, the **startTime()** function [4] is used with a **setTimeout()** function inside.

However, to enable the user to change the time in another functionality, the time need to be a global variable instead of a local variable inside the startTime function which is updated to get the time from machine.

In beginning of the main method, the current time is get once by the **Date()** function to set the default starting time of the clock. Then, the update function simply increment the time on its own. The current time is stored in a global variable which can be changed by another method easily.

After ensuring the correct time is calculated, the animation of clock tick is done by calculating the angle that the clock hand should be in, using the formulas as following:

$$\frac{\text{Tick angle of seconds hand}}{360} = \frac{actual\ seconds}{60}$$

$$\frac{\text{Tick angle of hours hand}}{360} = \frac{actual\ hours}{12}$$

3.2 ADDITIONAL FUNCTIONALITIES

3.2.1 DATE AND WEEK DAY DISPLAY

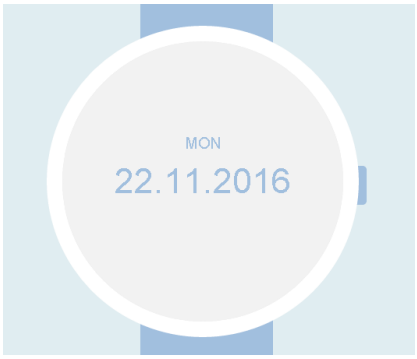

Figure 3.2.1, date display of the watch.

To implement a week day display on the digital clock, the JS function **Date().getDay();** is used to generate the current day in a week. The output of this function produce a integer within 0-6 which can then be used in a switch case to determine which day of the week should be displayed from Sunday to Saturday.

3.2.2 WORLD TIME

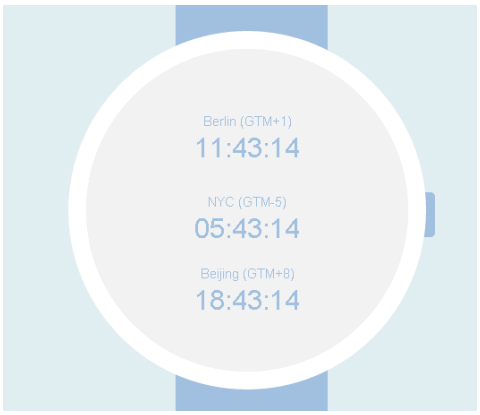

Figure 3.2.2, world time functionality of the watch.

Although this application is created originally in the UK, the world time functionality need to ensure that the world time works correctly even when used in other time zones. To do this, the program need to first find the current UTC time based on the time zone of the computer in use. **today.getTimezoneOffset()** function returns the different between the current time zone and the UTC time. Then, the UTC time could be easily calculate times and translated to other time zone.

Another concern is whether the time is displayed within the 0-24 hours range. For example, when UK time (GTM) is 20:00, Beijing time (GTM+8) is 28:00 based on the calculation, which will be converted to 04:00 for display. This have been carefully taken care of by the functionality.

3.2.3 TIMER

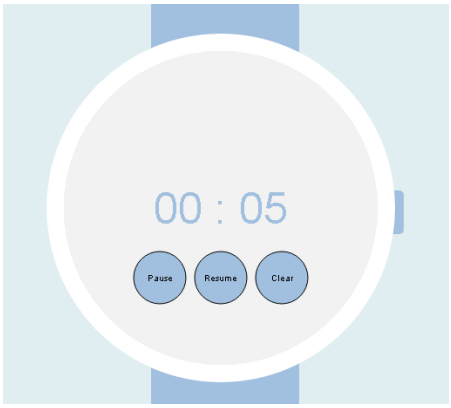

Figure 3.2.3, date display of the watch.

The timer starts automatically when the functionalities is pressed for quicker and convenient use. The current time of the timer is displayed by text in the middle and 3 buttons: pause, resume and clear is displayed for controlling the timer. When timer starts, **setTimeOut()** function used on timer function to start the timer running. **ClearTimeOut()** function is set to clear the timer.

However to prevent the user clicking the resume button more than once and triggering multiple timer running at the same time, a global variable **isPaused** is used to store the current state of the timer, allowing the timer to resume only when it is paused.

3.2.4 ALARM

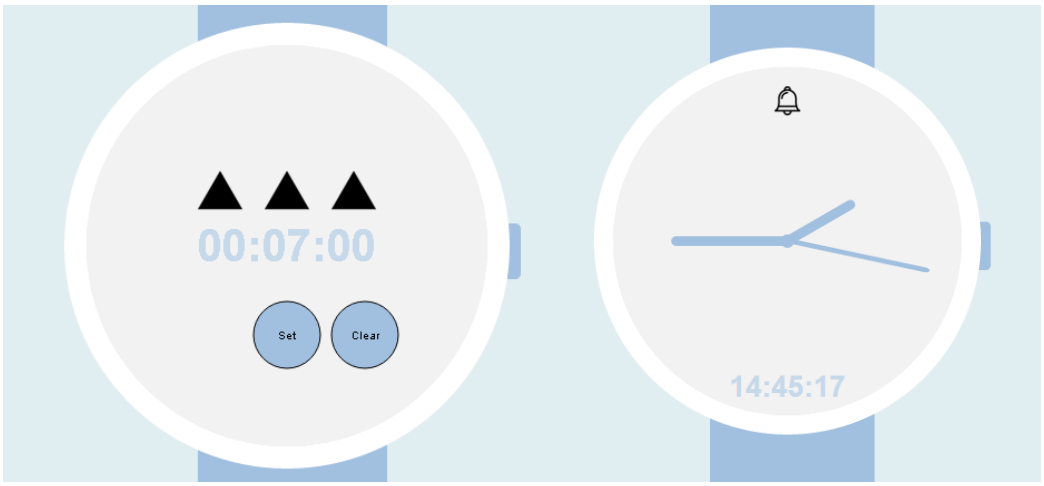

Figure 3.2.4, alarm of the watch.

The interface of the alarm setting page consist of the alarm time, 3 triangle button for adjusting time, and 2 extra button for setting and clearing the alarm.

When an alarm is set, a trigger is set up to updated a function which updates itself using **setTimeOut()** function every seconds to check if the alarm time is reached. When the alarm time is reached, the watch will jump to the alarm page, and a sound will be played to alarm the user. To stop the alarm and sound, the "clear" button need to be pressed.

Also, in case the user forgets about an alarm being set, an alarm logo will appear on the watch screen whenever an alarm is active, which can be seen in the left image of figure 3.2.4.

3.2.5 DAY/ NIGHT TIME THEMES

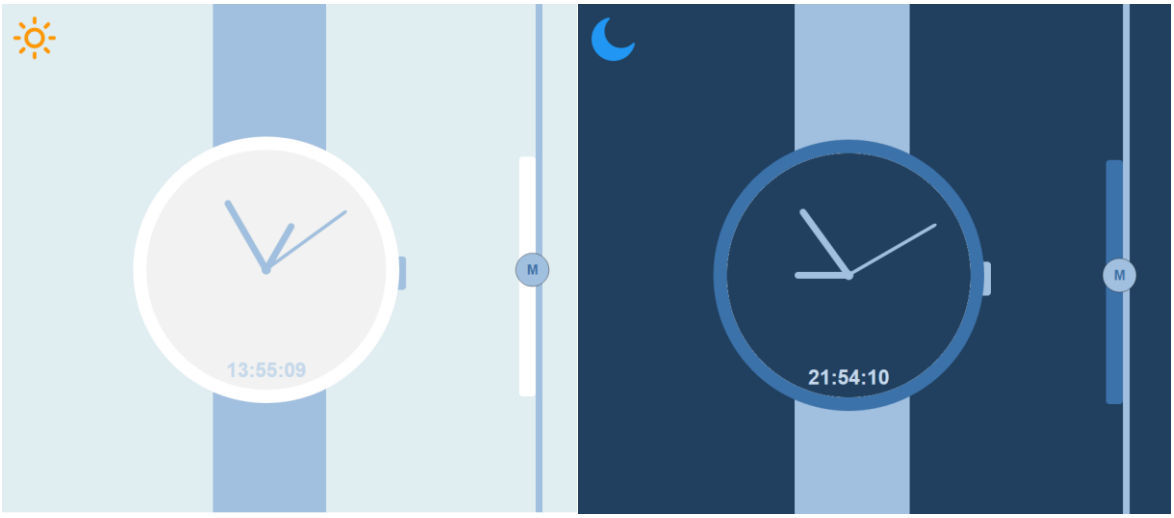

Figure 3.2.5, day and night theme of the watch.

To enhance the user experience, the colour theme of the clock is changed according to the current time of the clock to remind the user whether it is day or night time. Icon image is also used to indicate the theme.

4. CONCLUSION

In summary, the watch application has been successful. Future development could focus on enhancing the visual design of the watch and also adding more functionalities, for example, stop watch.

5. REFERENCE

[1] Raphaël(2008)Raphaël. Available at:http://raphaeljs.com/ (Accessed: 10/02/2016)

[2] Image Icons(2016)FlatIcon. Available at:www.flaticon.com (Accessed: 10/02/2016)

[3] Alarm sound track(2014)Alarm Clock Digital. Available at: https://www.freesound.org/people/zanox/sounds/233645/ (Accessed: 15/02/2016)

[4] startTime()Function (2016)'2D vector graphics and animation.', University of Sussex.