Index.js

import clock from "clock";

import document from "document";

import { preferences } from "user-settings";

import \* as util from "../common/utils";

import { HeartRateSensor } from "heart-rate"; // import HR reading from sensor (seel line 18)

import { battery } from "power";

import { vibration } from "haptics";

// Update the clock every minute

clock.granularity = "minutes";

// Get a handle on the <text> element

const myLabel = document.getElementById("myLabel");

const myMonth = document.getElementById("myMonth");

const myDay = document.getElementById("myDay");

const myHeartRate = document.getElementById("heartrateLabel")

const hrmLabel = document.getElementById("hrm-label");

const myBattery = document.getElementById("myBattery");

// Update the <text> element every tick with the current time

clock.ontick = (evt) => {

let today = evt.date;

let hours = today.getHours();

let monthnum = today.getMonth();

let day = today.getDate();

var month = new Array();

month[0] = "Jan";

month[1] = "Feb";

month[2] = "Mar";

month[3] = "Apr";

month[4] = "May";

month[5] = "Jun";

month[6] = "Jul";

month[7] = "Aug";

month[8] = "Sep";

month[9] = "Oct";

month[10] = "Nov";

month[11] = "Dec";

let monthname = month[monthnum];

if (preferences.clockDisplay === "12h") {

// 12h format

hours = hours % 12 || 12;

} else {

// 24h format

hours = util.zeroPad(hours);

}

let mins = util.zeroPad(today.getMinutes());

myLabel.text = `${hours}:${mins}`;

myMonth.text = `${monthname}`;

myDay.text = `${day}`;

}

let hrm = new HeartRateSensor();

hrm.onreading = function() {

myHeartRate.text = `${hrm.heartRate}`; // the measured HR is being sent to the heartrateHandle set at line 16

}

hrm.access\_heart\_rate

hrm.start();

myBattery.text = `${battery.chargeLevel}%`; // initialize on startup

battery.onchange = (charger, evt) => {

myBattery.text = `${battery.chargeLevel}%`;

}

var btnScoreA = document.getElementById("scoreA");

btnScoreA.onactivate = function(evt) {

vibration.start("nudge-max");

}

Utils.js

// Add zero in front of numbers < 10

export function zeroPad(i) {

if (i < 10) {

i = "0" + i;

}

return i;

}

Index.gui

<svg>

<image href="tesla watchface.png" />

<text id="myLabel"></text>

<text id="myMonth" />

<text id="myDay" />

<text id="heartrateLabel" />

<text id="hrm-label" class="sensor-label">BPM</text>

<text id="myBattery" class="topLeft smallFont"/>

<text id="myBattery" class="sensor-label">%</text>

<use id="scoreA" href="#square-button" x="200" y="10" width="100" fill="fb-white" >

<set href="#text" attributeName="text-buffer" to="haptic" />

</use>

</svg>

Styles.css

.background {

viewport-fill: red;

}

#myLabel {

font-size: 80;

font-family: System-Bold;

text-length: 32;

text-anchor: middle;

x: 40%;

y: 100%;

fill: white;

}

#myMonth {

font-size: 60;

font-family: System-Bold;

text-length: 32;

text-anchor: end;

x: 40%;

y: 1%+60;

fill: green;

}

#myDay {

font-size: 60;

font-family: System-Bold;

text-length: 32;

text-anchor: end;

x: 65%;

y: 1%+60;

fill: lightblue;

}

#heartrateLabel {

font-size: 50;

font-family: System-Bold;

text-length: 3;

text-anchor: middle;

x: 85%;

y: 90%;

fill: white;

}

.sensor-label {

font-family: System-Bold;

fill: white;

text-anchor: middle;

text-length: 32;

font-size: 20;

x: 85%;

y: 76%;

}

#myBattery {

font-size: 40;

font-family: System-Bold;

text-length: 32;

text-anchor: end;

x: 100%;

y: 100%;

fill: darkblue;

}

Widgets.gui

<svg>

<defs>

<link rel="stylesheet" href="styles.css" />

<link rel="import" href="/mnt/sysassets/widgets\_common.gui" />

<link rel="import" href="/mnt/sysassets/widgets/square\_button\_widget.gui" />

</defs>

</svg>

Import image name it ‘tesla watchface.png”

