2024 – Interview practice task

Task 1 – JS Coding

This exercise will walk you through the most important areas of creating a glass cockpit for an aircraft. You will have to find ways to solve the required functionalities. To effectively get over the obstacles you will face in this task as well as in the future, you will need to utilize various resources. Firstly, you should refer to the official MSFS SDK documentation, which provides the essential details (Simulation Variables, Event IDs, etc.). In addition, you will have access to our internal tools, which are created to improve your productivity and minimize human error (or save you from a monotone task). Lastly, collaboration with your colleagues will be crucial. They bring valuable experience and insights that can help you to successfully complete the challenges. During this task **if you have issues feel free to ask Viktor Gyenes (viktor.gyenes@propairflight.com)**

The goal of this task is to create a multifunctional glass cockpit which will include different pages and functionalities. The main functionalities you need to create are the following:

- Pagination
- Formatted display of variables some of which may depend on other states
- Working with SVGs

Requirements and description

You will create two pages for the instrument. The first page is an attitude indicator which includes an artificial horizon, a bank and a slip indicator. The second page is a status page which will contain some indications about the aircraft. Those indications are flaps position and engines temperature. If there is no electricity the instrument must not show any indications.

Recommended tools

We recommend the use of the following tools:

- VScode
- Live Server (VScode extension)
- Glass cockpit testing (internal)
- Git

You can choose other tools, however, probably we won’t be able to support you on the same level.

Tech requirements

During the practice use HTML, CSS and vanilla JS.
Be aware that **the simulator uses chrome 49** as a user agent. (check caniuse.com frequently)

Getting Started

First you need to get familiar with the environment which we use for instrument development. Download the TEST_Instrument.zip file which contains the initial project you should use during the task. Extract the files and open the html file with Live Server, try out the CIRCUIT slider and see what happens. Based on the circuit slider you can add more sliders to mock more Simulation Variables that you need for the tasks. You should use the same variable name and unit here as you would use in the simulator.

Pagination

The objective is to create a pagination that will allow us to switch between the attitude indicator and the status page. These pages should be in one html inside the <div id="Electricity"> element. The architecture should work well for more pages as well while it's not overengineered. The instrument should only update the currently selected page. The main page is the attitude indicator page.

Attitude Indicator Page

On this page you will create an attitude indicator with artificial horizon, bank and slip indicator. Place the attitude indicator in a square area to the middle of the screen. The square should be the same height as the screen.

How an attitude indicator moves: Attitude indicator in action

Pitch Indicator

The pitch indicator has horizontal lines. The short lines indicate five degrees of pitch, and the long lines indicate ten degrees of pitch. At the ends of the long lines there is an indication for that line's pitch value e.g., 30° or 80°. The pitch indicator's range is from -100° to +100°.

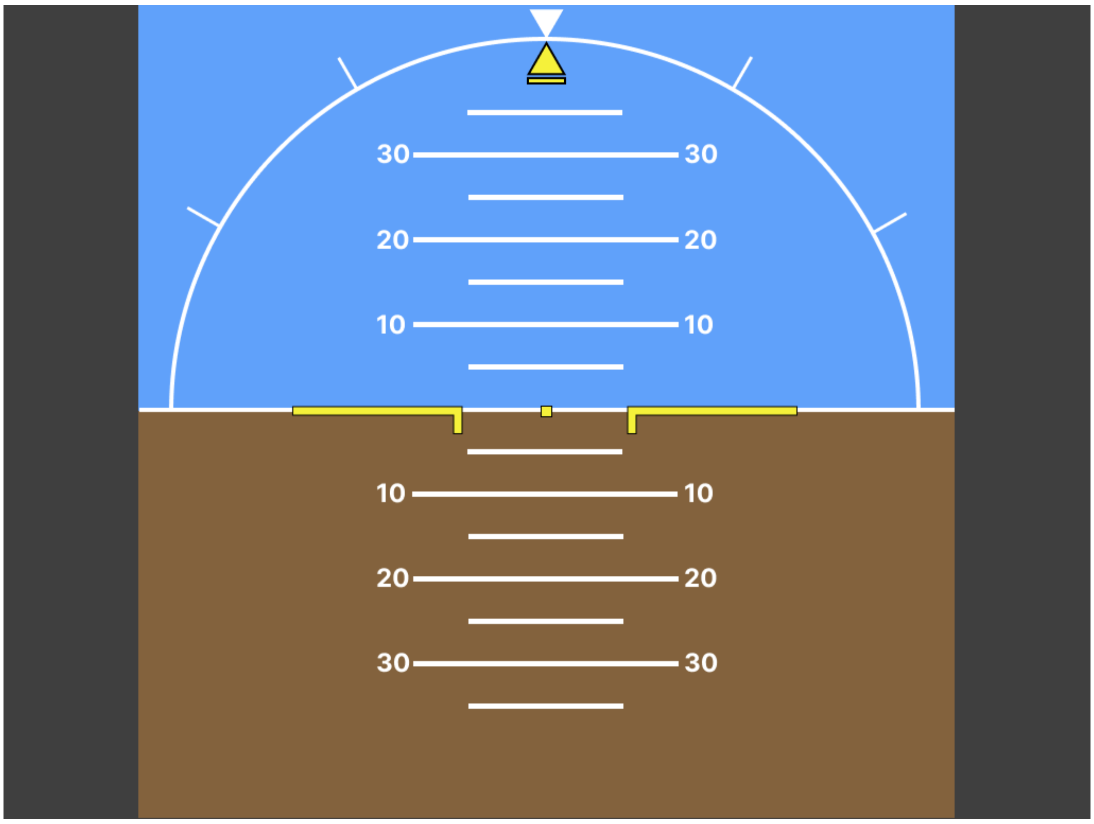
The pitch tape should not overlap with the turn and slip indicator.

Bank and Slip Indicator

The bank angle is indicated with the yellow triangle which is in a fixed position and the white arc which turns opposite to the bank angle. The arc has markings at ±30° and ±60°.

The yellow line under this triangle is the slip indicator which can move to left and right based on the slip.

The horizon and the pitch tape turn together with the arc.



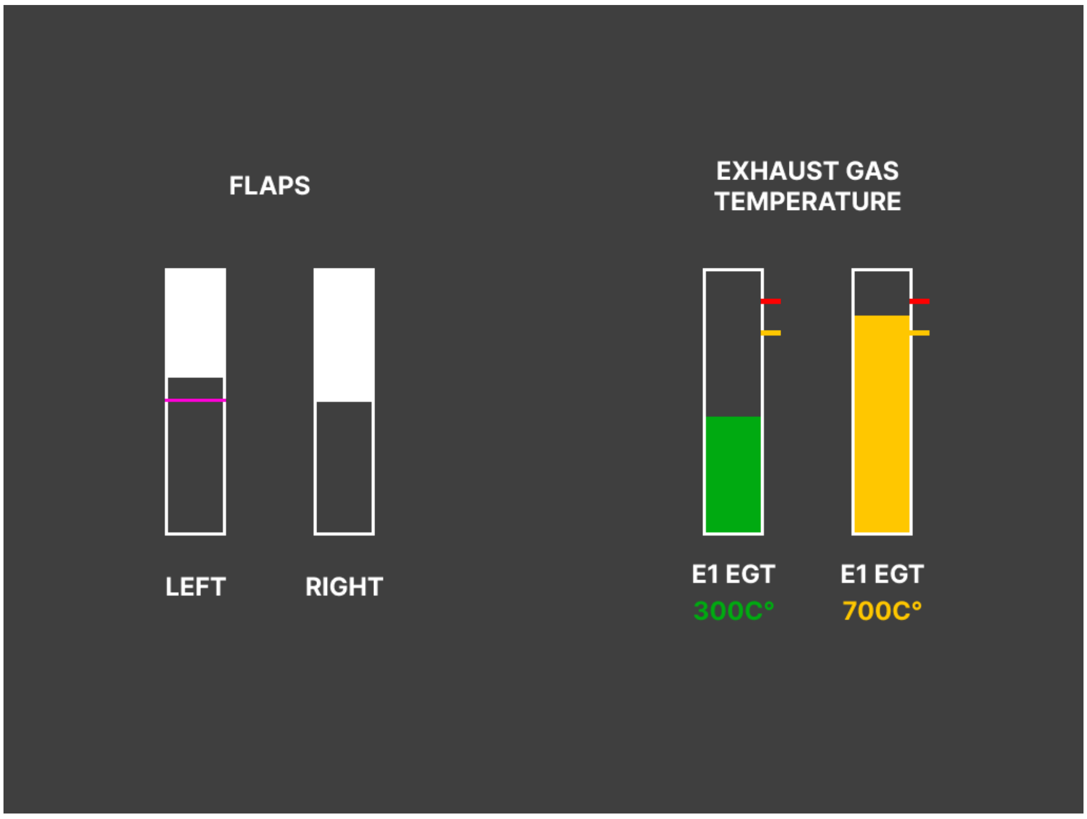
Status Page

The status page contains a flaps indication and an engine temperature indication. The flaps indication is on the left side of the screen and the engine temp indication is on the right side.

Flaps Indicator

The flaps indicator shows the left and right flaps independently on two separate bars. The deflection of the flaps is shown with a white indication in the bar starting from the top. A magenta line crosses the bar where the selected flap will be after moving into position. This line disappears when the flap is in position (with a small threshold). This indicator disappears when there is no flap deflection, and the selected flap position is the retracted position.

There are three positions for the flaps handle which correspond to the following values: 0°, 20°, 42°.

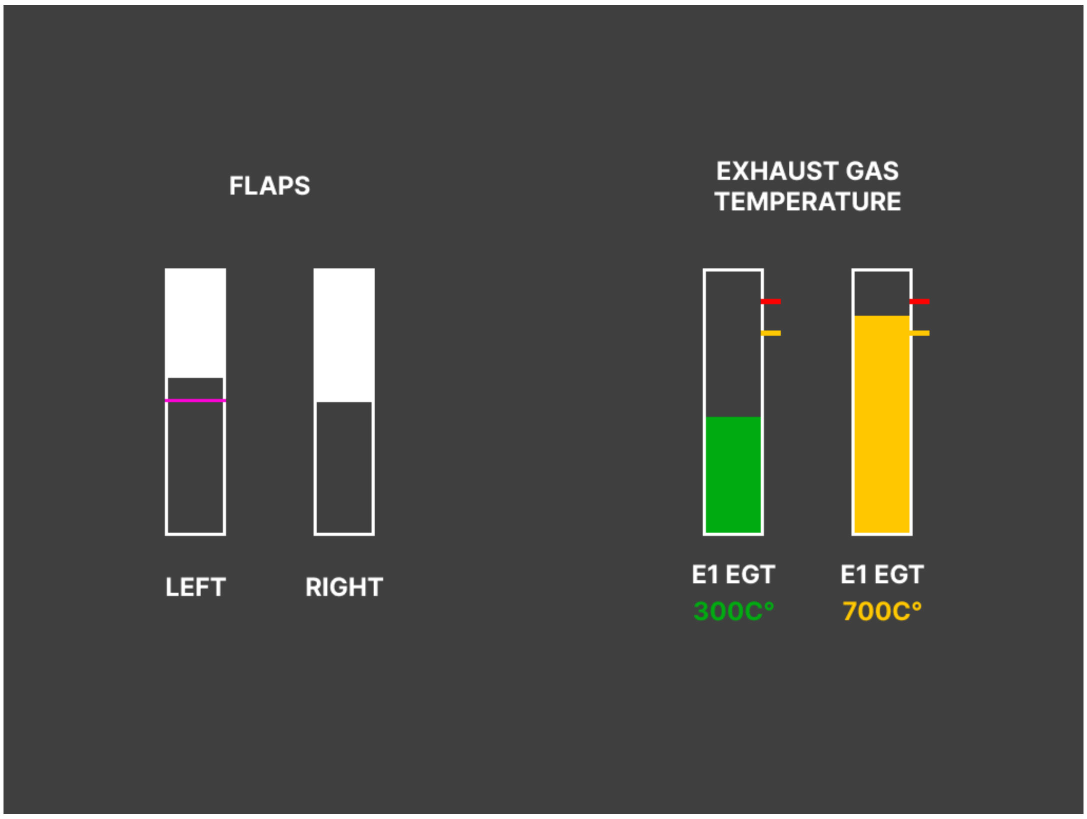


Exhaust Gas Temperature

The engine temperature indication shows the exhaust gas temperature of engine 1 and engine 2. The temperature is displayed in two places, one inside the bar and one as a label below the bar. The indication inside the bar and the text which indicates the temperature value change color based on the temperature. The indications are originally green, over 680C° EGT appear as amber and over 750C° those are red. The limits are also indicated on the right side of the EGT bars. The range of indication is from 100C° to 899C°.

You should create a solution that allows you to easily change the limitation values as well as the indicatable ranges if necessary.

(The illustration below will not be correct in aspect of the positions of the limits).



Tips for the development stage

DOs

Preferably get static html elements in the connectedCallback function and not in the Update.

In html, it is recommended to use custom attributes in cases where different states cannot be active at the same time e.g. off/loading/on (loading screen example is in the code generated with the build script)

In case the instrument contains multiple pages create those pages separate classes with update functions and only call the shown page's update functions with the variables When using the transform property (css/js), the following sequence should be followed: transform rotate scale

DONTs

You should avoid using clientWidth / clientHeight etc., due to the problems that come in when the component is not rendered in the DOM while initializing the inner elements (those variables will contain 0 instead of a desired value).

SVG

You should always specify the namespace for the svg elements (line, text, path, etc). Use the built-in document.createElementNS function.

You can easily align the text within the SVG with the text-anchor and the dominant-baseline

For easy positioning, it is recommended to leave the viewBox according to the width and height of the element

For more complex svg you should use tools like figma; gimp; PS; Illustrator etc...

You can change the line endings with the stroke-linecap to be rounded or box (box can be useful when you need to create corners or T shapes)

Test environment

You can add more test variables such as pitch or bank angle, just duplicate the slider of the circuit in the html and update the parameters as needed the last four parameters are the following: minimum, maximum, slider step and start value