



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

Semestrální práce z předmětu KKY/PP HTML + SVG + REST

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Vladimíra Kimlová

A20B0374P

Zimní semestr

23. 11. 2022

Obsah

1. Zadání	1
2. Vypracování	2
2.1. Použité technologie	2
2.2. Implementace	2
2.2.1. HTML.....	2
2.2.2. CSS.....	3
2.2.3. JavaScript	4
3. Uživatelský manuál	5
4. Závěr	6

1. Zadání

Úkolem semestrální práce je na základě přiloženého matematického modelu (*Quarter-car model*) vytvořit uživatelské rozhraní HMI pomocí technologií HTML, SVG a REST.

HMI bude obsahovat grafiku, která zobrazí aktuální stav kola na vozovce tak, aby si nezávislý uživatel dokázal představit, v jakém stavu se systém nachází.

HMI bude:

1. responzivní – umožní reakci na změnu velikosti okna prohlížeče
2. obsahovat hlavičku, navigační panel s ovládacími prvky a plochu pro zobrazení dat z modelu

Z hlediska uživatelského rozhraní bude HMI minimálně obsahovat:

1. Zobrazení výšky vozidla nad vozovkou (vizuálně i číselně včetně jednotek, zaokrouhleno na jedno desetinné místo, efekt zvětšit cca 10x, aby byl viditelný)
2. Zobrazení (animace) pružiny a tlumiče na základě veličiny `pos_susp_meas` a průměru kola `pos_wheel1` (vizuálně i číselně včetně jednotek, zaokrouhleno na jedno desetinné místo, efekt zvětšit cca 10x, aby byl viditelný)
3. Deformaci kola `pos_wheel1` (z kružnice do elipsy, efekt zvětšit cca 10x, aby byl viditelný)
4. Velikost zrychlení působící na řidiče v kolmém směru na vozovku (například pomocí sloupcového grafu)
5. Aktuální hmotnost v [kg] působící na pneumatiku – přepočet proměnné `F1_tire`
6. Zobrazení aktuálního stavu spojení se serverem. Pokud spojení nebude fungovat nepůjde zadávat hodnoty. Při obnovení spojení se provede inicializace zápisových komponent, aby odraželi stav v řídicím systému.

Ovládání aplikace:

1. Aktivace skokové poruchy `qcm.MP_NUDGE:BSTATE` a její intenzity `qcm.GAIN_NUDGE:k`
2. Aktivaci náhodné poruchy `qcm.SG_bumps:amp` a vhodné frekvence `qcm.SG_bumps:freq`
3. Zapnutí `qcm.CNB_RUN:YCN` a RESET modelu `qcm.MP_RESET:BSTATE`

Kromě samotné aplikace sepište semestrální práci, která bude obsahovat:

1. Zadání
2. Stručný popis použitých technologií
3. Popis postupu implementace vaší aplikace (vybrané ukázky zdrojového kódu – klíčové funkce)
4. Základní uživatelský manuál
5. Závěr

2. Vypracování

2.1. Použité technologie

HTML znamená **Hypertext Markup Language**, tedy hypertextový značkovací jazyk. HTML se vyvinul ze SGML a stal se používaným jazykem pro tvorbu webových stránek. Slovo **HyperText** zde vyjadřuje možnost vzájemně propojovat texty pomocí odkazů, **Markup** označuje schopnost jazyka HTML dávat významy jednotlivým blokům textu s pomocí speciálních značek tzv. elementů či tagů. Mezi základní tagy patří např. `html`, `head`, `body`, `div`, `span`, `p`, `h1`, ...

Kaskádové styly, zkratkovitě **CSS** (z anglického *Cascading Style Sheets*) jsou jazykem umožňujícím efektivní formátování stránek psaných ve značkovacích jazycích. Umožňují vytvářet vzhled jako druhou, na obsahu nezávislou vrstvu a různě ho měnit podle aktuálního kontextu.

JavaScript je objektově orientovaný, událostmi řízený programovací jazyk, využívaný při tvorbě webových stránek. Na straně klienta (tedy v prohlížeči) zajišťuje např. dynamické změny obsahu stránek, ovládání různých interaktivních prvků, či animace.

SVG (z anglického Scalable Vector Graphics) je vektorový formát souborů. Soubory **SVG** jsou napsány v jazyce XML. Na rozdíl od rastrových formátů souborů, které se skládají z obrazových bodů, uchovávají vektorové soubory obrázky pomocí matematických předpisů. To znamená, že lze výrazně měnit velikost, aniž by ztratily svou kvalitu.

REST API zajišťuje funkci „rozhraní“ pro předávání dat mezi dvěma či více aplikacemi. REST k tomuto obecnému mechanismu dodává sadu doporučení a omezení např. bezstavovost. **REST API** obsahuje dílčí endpointy, které jsou dostupné na definovaných URL adresách, které provolává klientská aplikace přes HTTP protokol (pomocí metod POST, GET, PUT...). Pro tělo požadavku se nejčastěji používá komunikační formát JSON.

2.2. Implementace

2.2.1. HTML

Webová stránka obsahuje hlavičku, hlavní část a patičku. Hlavní část je rozdělena na plátno pro vykreslování SVG, postranní panel a plochu pro vykreslení grafu.

Náčrtek auta je rozdělen na 3 části: karoserii auta, pružinu s tlumičem a kolo. Karoserie je vytvořena pomocí Bézierovy křivky a vodorovné čáry. Kolo se skládá ze tří kruhů – černá pneumatika, bílý vnitřek pneumatiky a černý střed kola. Pružina byla vytvořena pomocí svislé přerušované čáry, zubaté čáry představující pružinu, obdélníku představujícího tlumič a několika dalších čar. Pod samotným autem se ještě nachází zem – zde bylo využito `for` smyčky v JavaScriptu pro vytvoření šikmých čárek.

Postranní panel je tvořen pomocí `divů` a `spanů` pro zobrazení hodnot aktuálních výstupů a tlačítek, zaškrťovacích boxů a inputboxů pro ovládání simulace.

Graf je vytvořen pomocí Plotly – jeho implementace byla tedy provedena v JavaScriptu.

2.2.2. CSS

Vzhled webové stránky byl inspirován webovým rozhraním poskytnutého API. Hlavička i patička tedy obsahují stejný barevný gradient. Barvy z tohoto gradientu byly využity i pro nastýlování zbytku stránky, např. tlačítek, pružiny, některých textů, či obecně stínů u daných boxů.

Responzivita webové stránky byla zajištěna primárně pomocí flexboxů – ty se při malém rozlišení dají pod sebe místo vedle sebe.

Kaskádových stylů bylo také využito pro změnu vzhledu zaškrťovacího boxu tak, aby vypadal jako tlačítko.

```
.toggle {  
    visibility: hidden;  
}  
  
.toggle+label {  
    display: block;  
    position: relative;  
    cursor: pointer;  
    border-radius: 10px;  
    width: 120px;  
    margin-left: 20px;  
    margin-right: 20px;  
    font-size: 20px;  
    font-weight: bold;  
    box-shadow: 0px 2px 8px rgba(0, 200, 180, 0.8);  
    overflow: hidden;  
}  
  
input.toggle-round-yes-no+label:before {  
    content: "Off";  
    color: #00C8B4;  
    backface-visibility: hidden;  
    background-color: #fff;  
    text-indent: inherit;
```

```

}

input.toggle-round-yes-no+label:after {
    content: "On";
    color: #FFF;
    backface-visibility: hidden;
    transform: rotateY(180deg);
    background-color: #00C8B4;
    text-indent: inherit;
}

```

Ukázkový kód nastýlování jednotlivých stran zaškrťovacího boxu

2.2.3. JavaScript

V samotném JavaScriptu byly nejprve vytvořeny konstanty s URL adresami potřebných hodnot z API a proměnné s upravovanými HTML elementy.

Jako kostra celé aplikace byl využit kód ze cvičení, v němž již byly mimo jiné implementovány metody GET a POST, zajišťující komunikaci s REXYGENEM a metoda pro výpis chyby.

Metoda `initWriteDatapoints` byla upravena tak, aby při načtení webové stránky synchronizovala hodnoty poruch a tlačítka pro vypnutí/zapnutí modelu.

Metoda `readLoop` zajišťuje aktualizaci hodnot a animaci auta. Nejprve jsou načteny a zobrazeny hodnoty výstupů z modelu, dále je provedena animace a vykreslení grafu.

Animace jsou prováděny pomocí `style.transform`, a to konkrétně translace a škálování. Animace celého auta je pouhou translací na základě změny veličiny `pos_car_abs`. Deformace kola je provedena na základě změny veličiny `pos_wheel`. Je zde také využito transformace počátku a to tak, aby kolo zůstalo v kontaktu se zemí a nedeformovalo se pouze kolem středu.

```

wheel.style.transformBox = "fill-box";
wheel.style.transform = "scale(1, " + wheel_deformation + ")";
wheel.style.transformOrigin = "bottom";

```

Ukázkový kód deformace pneumatiky

Animace karoserie je obdobná jako u animace celého auta pouze s tím rozdílem, že posun je prováděn na základě změny veličiny `pos_susp_meas`. Pohyb pružiny je pak kombinací deformace kola a posunu karoserie auta. Pružina je nejprve posunuta stejně jako samotná karoserie a následně je natažena. Natažení

pružiny odpovídá 2 – natažení kola (neboť probíhá v opačném směru, např. z natažení 102 % se tak stane 98 % - tedy smrštění) a to celé mínus poměr translace karoserie.

```
var spring_deformation = 2 - wheel_deformation;
spring_deformation = spring_deformation-(car_translation / car_wheel_init);
spring.style.transform = "translateY(" + car_translation + "px)";
spring_dumper.style.transformBox = "fill-box";
spring_dumper.style.transform = "scale(1, " + spring_deformation + ")";
```

Ukázkový kód deformace pružiny

Dále je zajištěno uzamknutí zadávání vstupních hodnot a přepínání přepínačů v případě, kdy přestane fungovat spojení se serverem. Uzamknutí je provedeno nastavením atributu `disabled`, odemčení jeho odejmutím. Po ztracení spojení se program snaží znovu toto spojení navázat, pokud úspěje dojde k obnovení stránky a k aktivaci ovládacích prvků.

Zobrazení zrychlení působícího na řidiče pomocí grafu je zajištěno pomocí Plotly. Zde byl trochu problém s přeškálováním osy x, neboť hodnoty zrychlení se velice rychle mění, a navíc mění znaménko. Osa x tedy byla staticky zařazována nulou na střed a rozsah musel být dopočítán ručně. Rozsah je spočítán jako: rychlost zaokrouhlená na 4 desetinná místa – zaokrouhleno na 1. platnou cifru – vynásobeno 10 – vyděleno hodnotou 1. platné cifry (tj. z čísla 0.0102 se stane 0.1)

```
acceleration = Math.round(acceleration * 10000) / 10000;
// compute the x-axis range
var max_range = acceleration.toPrecision(1) * 10;
max_range = Math.abs(max_range);
var str_max_range = max_range.toString();
var last_digit = str_max_range.charAt(str_max_range.length - 1);
max_range = max_range / last_digit;
```

Ukázkový kód výpočtu rozsahu osy x

Plotly umožňuje i zajištění responzivity – při zmenšení webové stránky bude zmenšen i samotný graf.

JavaScript provádí také aktualizaci hodnot (tedy volání funkce `readLoop`) každých 100 milisekund – kompromis mezi plynulostí animace a schopností lidského oka číst hodnoty výstupů.

3. Uživatelský manuál

Pro běh webové stránky je nutné, aby byl lokálně spuštěn i program REXYGEN a v něm lokální server. Webová stránka je pak v prohlížeči dostupná na adrese <http://localhost:8008/hmi/index.html>.

Horní část postranního panelu představuje aktuální hodnoty výstupů modelu – na jejichž základě je prováděna animace, lze je ovlivnit vytvořením skokové nebo náhodné poruchy pomocí ovládacího panelu pod nimi. U skokové poruchy lze nastavit intenzitu a kliknutím na tlačítko jednorázově poruchu vygenerovat. U poruchy náhodné lze ovlivnit její amplitudu a frekvenci, poruchu lze aktivovat tlačítkem (switchboxem) On – lze ji pak opět deaktivovat tlačítkem Off. Dále je možno vypnutí/zapnutí modelu – tedy pozastavení simulace a RESET modelu. Pod ovládacím panelem je zobrazen status spojení se serverem, pokud bude spojení přerušeno, nebude možno simulaci ovládat.

4. Závěr

Úkolem semestrální práce bylo vytvořit simulaci levé přední části auta vybaveného aktivním tlumičem. Pro vizualizaci této simulace byla vytvořena webová stránka využívající technologie HTML, SVG a REST. Simulaci je možno ovládat pomocí postranního panelu, na němž je možno nastavit a aktivovat poruchy. Do budoucna by šla animace rozšířit o další prvky než jen kolo, karoserii a tlumič, případně by bylo možné webovou stránku doplnit např. o další grafy zobrazující například vývoj nějaké veličiny v čase nebo v závislosti na velikosti poruchy a podobně.