ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

KOMPONENTY PRO VÝUKOVÝ MATERIÁL-HTML ELEMENTY CANVAS A SVG

BAKALÁŘSKÁ PRÁCE

Lukáš Ledvina
Informatika se zaměřením na vzdělávání (Vt)

Vedoucí práce: PhDr. Tomáš Přibáň, Ph.D.

Plzeň 2019

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni dne 30. června 2019

.....

vlastnoruční podpis

PODĚKOVÁNÍ

Děkuji vedoucí bakalářské práce PhDr. Tomáši Přibáňovi, Ph.D. za čas, který jste obětoval a za připomínky, rady, a hlavně za vedení a podporu.

OBSAH

SEZNAM ZKRATEK

2-D	Dvoudimenzionální/Dvourozměrný
3-D	Trojdimenzionální/trojrozměrný
API	Application Programming Interface
CSS3	Cascading Style Sheets 3
DOM	Document Object Model
GNU	General Public Licence
HTML	Hypertext Markup Language
JPG	Joint Photographic Experts Group
LIFO	Last In First Out
SMIL	Synchronised Multimedia Integration Language
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 Úvod

Cílem této bakalářské práce je přiblížit grafický vektorový element SVG a bitmapový element Canvas. Možnosti obou technologií jsou velice rozsáhlé, a proto jsem v předkládané práci soustředil pozornost především na popis metod/elementů a problémů spojených s webovou grafikou. S grafickými prvky se na internetu setkáváme již velice dlouho. Neustálé zdokonalování internetových prohlížečů vede ke vzniku nových technologií usnadňujících programátorům jejich práci. Dříve se grafika na webu tvořila pouhým nahráním bitmapových obrázků. Tyto byly rozděleny do několika menších proto, aby uživatel nemusel čekat, než se nahraje celý velký bitmapový obrázek. Dnešní grafické formáty nabízejí nové technologie a snadnější možnosti, jak využívat grafiky na webových stránkách, a také uživatelé mají lepší internetové připojení. Jsme dokonce schopni vytvářet hry přímo v prohlížeči či vytvářet dynamickou grafiku pomocí elementu SVG. HTML5 Canvas a SVG jsou technologie sloužící k vytváření grafiky na internetové stránce, které se však od sebe velmi liší. SVG je vektorový grafický formát založený na XML. HTML5 Canvas je ECMAScriptové aplikační prostředí dovolující kódovat programové kreslící operace.

První kapitola je rozdělena na části zaměřující se na SVG a Canvas, v nichž budou popsány elementy SVG a metody ECMAScriptu pro Canvas. Tyto dvě kapitoly budou rozděleny do menších kapitol, jako jsou základní tvary, text, transformace atd. U každého popisu bude také zapsán kód.

Ve druhé kapitole bude popsáno, jakým způsobem je možné vytvářet animace pro webové stránky pomocí technologií Canvas a SVG.

Třetí kapitola bude věnována analýze ECMAScriptových a SVG knihoven.

Čtvrtou kapitolu bude tvořit popis elektronického kurzu a příkladů, které ukážou funkčnost všech metod a elementů popsaných v teoretické části. Elektronický kurz bude vytvořen v systému Moodle. S tímto řešením se mohou studenti připojit do kurzu pomoci STAGu.

Všechny příklady jsou dostupné na https://codepen.io/fpe-zcu-tutorial. Díky tomu mohou studenti vidět přímo kód v kurzu a upravovat jej podle potřeby. Při tvorbě příkladu bylo

počítáno se znalostí skriptovacího jazyka ECMAScript. Kurz je dostupný na adrese https://moodle.athos.zcu.cz/course/view.php?id=23.

1.1 HTML5

Jedná se o pátou verzi značkovacího jazyka HTML. V roce 2007 byl zahájen vývoj, přičemž finální specifikace byla vydána 28. října roku 2014. Vývojářský tým vydal již verzi 5.1 a 5.2, pracuje také na verzi HTML5.3. HTML5.3 by pak mělo být dokončeno kolem roku 2022. HTML je zkratkou pojmu hyper text markup language. S HTML5 také přišlo CSS3, kteréž slouží ke stylování HTML prvků (zaoblené rohy, stíny, animace, přechody atd.). Další novinkou je element Canvas, který umožňuje zobrazovat bitmapovou grafiku. Součástí elementu Canvas je plátno. Do plátna můžeme vkládat videa a obrázky. Kreslicí operace se využívají pomocí ECMAScriptu. HTML5 také obsahuje elementy audio a video. Element audio slouží k přehrávání zvukových stop. Nabízí možnost snadného vložení zvukové stopy na webové stránky. Element video nahrazuje složitý element object a umožňuje snadné vložení videa na webové stránky. Oba Element mají nativní podporu, což znamená, že není potřeba instalace plug-inů ani přehrávačů [9].

1.2 ECMAScript

ECMAScript je objektově orientovaný skriptovací jazyk, jehož vývoj začal v roce 1995. Výhodou ECMAScriptu je multiplatformita. Funguje na jakémkoliv zařízení, na němž je nainstalován internetový prohlížeč (počítač, mobil, hodinky aj.), na operačním systému tak nezáleží, aplikace fungují všude. ECMAScript běží na straně klienta, tedy všechny aplikace jsou spouštěny v prohlížeči až po stažení webové stránky klientem z internetu. Pomocí ECMAScriptu je možné vytvářet, mazat, kopírovat, upravovat HTML elementy. K přístupu k HTML elementům slouží DOM API. Rozhraní DOM je nezávislé na programovacím jazyku a obsahuje prvky webové stránky (atributy, texty, HTML elementy atd.). Každý prvek v DOM může být upravován pomocí ECMAScriptu, jako příklad můžeme uvést element Canvas [10,11].

1.3 CSS3

CSS se využívá pro grafickou úpravu webových stránek. Je to kolekce metod pro grafickou úpravu webových stránek. CSS3 je nejaktuálnější verzí CSS, která přináší možnost transformace objektů, kulaté rohy, stíny, selektory, ale hlavně tvorbu animací. Je to

nadstavba značkovacích jazyků HTML a XML. První verze CSS vyšla v roce 1996. CSS je spravováno a vyvíjeno mezinárodním konsorciem W3C [12].

1.4 STANDARD SMIL

Standard SMIL je značkovací jazyk, který přináší animační a interaktivní možnosti pro vektorovou grafiku. Umožňuje tak obejít zcela ECMAScript a vytvářet deklarativním způsobem interaktivní animované webové prezentace za použití pouze HTML a CSS. Síla standardu SMIL tkví ve využití jeho syntaxe v jiných XML jazycích a v jeho jednoduchosti. Tento standard navrhlo World Wide Web Consortium. Aktuální verze 3.0 byla vydána v roce 2008 [13].

2 CANVAS

Canvas je párový tag, který přišel spolu s HTML5. Element Canvas představuje bitmapu, jejíž pomocí je možné uvnitř plátna s definovanou šířkou a výškou provádět programové kreslicí operace prostřednictvím ECMAScriptu, a tak vytvářet ve webových stránkách dynamické grafické vykreslování obrázků, grafiky, her, grafů a podobně. Syntaxe vykreslování objektů je odlišná oproti SVG, protože využívá ECMAScriptové funkce. Pro vytváření grafiky pomocí Canvas potřebujeme mít na stránce minimálně jeden párový element Canvas. Prvek Canvas může být stylován jako každý jiný element (rámeček, okraj, pozadí atd.), tato pravidla nezmění vnitřek plátna. Prvek Canvas má dva atributy width a height. Oba jsou nepovinné. Atribut ID není spojen přímo u prvku Canvas, ale patří pod globální atribut HTML. Je důležité definovat atribut ID, jelikož se jeho pomocí budeme odkazovat na naše plátno [1, 5, 7].

<canvas id="plátno" width="200" height="200"></canvas>

2.1 ZÁKLADNÍ TVARY

Kreslení na Canvas je realizováno pomocí souřadnicové soustavy, která udává jednotlivé pixely Canvasu. Počáteční bod [0,0] se nachází v levém horním rohu a ostatní body vycházejí z počátku. Bod [5,10] se nachází 10 pixelů směrem doprava a 5 pixelů směrem dolů.

2.1.1 OBDÉLNÍK

Počátek (levý horní bod) obdélníku na plátně je určen pomocí souřadnic X a Y. Atribut width udává délku obdélníku v pixelech, height naopak výšku. Metoda stroke pak nakreslí požadovaný tvar na Canvas. Výchozí barva je černá [5].

rect(x, y, width, height)

2.1.2 ÚSEČKA

Metoda beginPath započne novou cestu, nebo vyresetuje aktuální. Metoda moveTo posunuje bod úsečky na specifický bod v Canvasu (X1 a Y1) bez vykreslení čáry. Metoda LineTo pak posunuje bod úsečky na určitý bod v Canvasu z původních souřadnic na X2 a Y2 a to bez vykreslení čáry. Metoda stroke vykreslí čáru z předem nadefinovaných hodnot z moveTo a lineTo metod [5].

beginPath()

moveTo(x1, y1)

lineTo(x2,y2)

stroke()

2.1.3 KRUH

Střed oblouku je na plátně určen pomocí X a Y, přičemž R je poloměr kružnice. Počátečním a konečným úhlem jsou sAngle a eAngle. Counterclockwise určuje směr vykreslení, hodnota může být true, nebo false. Hodnota true znamená, že kružnice bude vykreslena proti směru hodinových ručiček [5].

arc(x, y, r, sAngle, eAngle,counterclockwise)

2.1.4 KVADRATICKÁ KŘIVKA

Kvadratická křivka je tvořena pomocí metody moveTo, která značí začátek křivky. QuadraticCurveTo vykreslí křivku podle kontrolních X2 a Y2 konečných bodů a X3 a Y3 [5].

moveTo(x1,y1)

quadraticCurveTo(x2, y2, x3, y3)

2.1.5 BÉZIEROVA KŘIVKA

Bézierova křivka se vytváří velmi podobně jako křivka kvadratická, místo jednoho kontrolního bodu však existují dva [5].

bezierCurveTo(x1, y1, x2, y2,x3,y3)

2.2 VYBARVOVÁNÍ

Do proměnné BARVA je možné zapsat anglická slovíčka pro barvu (violet, blue, yellow atd.), rgb (0–255,0–255,0–255), rgba (0–255, 0–255, 0–255, průhlednost 0–1) i šestnáctkový zápis barvy (#000000). Výchozí barva je černá. Metoda fillStyle udává, která barva vykreslí vnitřek útvaru [5].

fillStyle = "barva“

2.2.1 PLNÝ ČTVEREC JEDNOU BARVOU

Metoda fillRect vykreslí vybarvený čtverec. Funguje na stejném principu jako metoda rect [5].

fillRect(x, y, width, height)

2.2.2 PŘECHOD BAREV

Proměnné X1 a Y1 udávají souřadnice počátku přechodu, X2 a Y2 pak udávají konec bodu přechodu. Metoda addColorStop může být volána několikrát. Proměnné X3 a X4 udávají pozici mezi začátkem a koncem barevného přechodu. Nabývají hodnot 0–1 [5].

createLinearGradient(x1, y1, x2, y2)

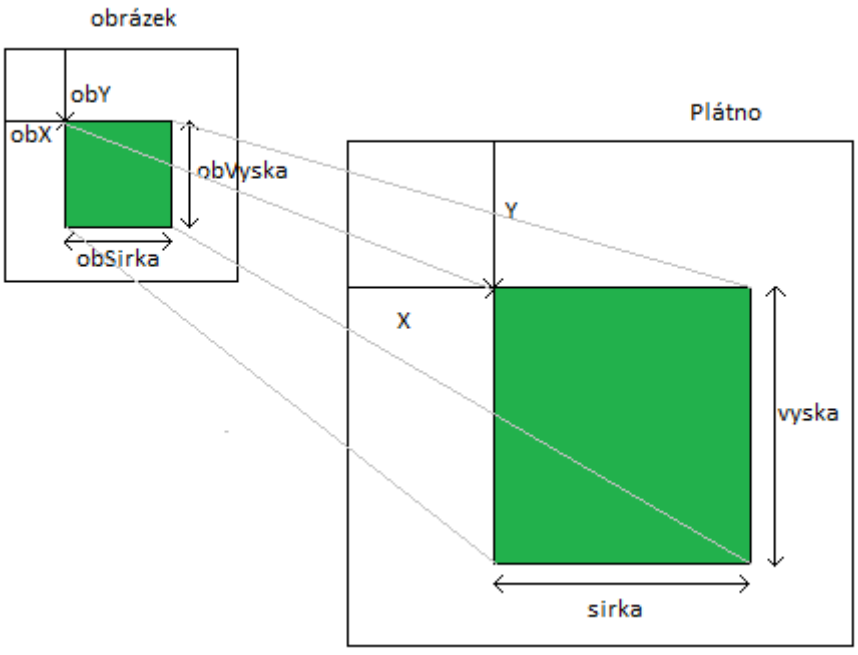
addColorStop(x3, "barva 1")

addColorStop(x4, "barva 2")

2.2.3 VLOŽENÍ OBRÁZKU

Metoda drawImage slouží k vložení obrázku do Canvasu. Proměnné X1 a Y1 udávají souřadnice počátku obrázku, levý horní roh. Proměnné X2 a Y2 udávají šířku a výšku v případě, že chceme změnit rozměry původního obrázku. Tyto dvě proměnné nejsou povinné [5].

drawImage(obrazek, x1, y1, x2, y2)



Obrázek 1: DrawImage

Proměnné obY, obX, obVyska a obSirka slouží k vystřihnutí části obrázku, pokud chceme změnit rozměry původního obrázku. Proměnné X a Y udávají souřadnice počátku obrázku na Canvasu. Proměnné šířka a výška udávají šířku a výšku odstřihnutého obrázku [5].

context.drawImage(obrazek, obX, obY, obSirka, obVyska, X, Y, sirka, vyska)

2.3 TEXT

Podobně jako u SVG můžeme i na Canvas kreslit text, avšak takový text zůstává stále pouze obrázkem z pixelů.

2.3.1 ŘETĚZEC

Proměnná text určuje text vypsaný na Canvas. X a Y jsou souřadnice počátku textu. Proměnná délka je nepovinná, určuje délku textu v pixelech. Metoda fillText slouží k vykreslení požadovaného textu na Canvas [5].

fillText (“text“, x, y, délka)

2.3.2 FONT, VELIKOST PÍSMA, ŘEZ PÍSMA

Proměnná X udává řez písma (normal, italic, bold, 100, 600 atd.). Proměnná Y udává velikost textu v pixelech (30px, 200px atd.) Proměnná Z udává font písma (Arial, verdana, serif atd.). Výchozí nastavení je sans-serif, 10px. Font tedy nastavuje charakteristiku písma (tloušťku, sklon, šířku atd.) [5].

font = "x y z"

2.3.3 BARVA TEXTU

Funguje stejně jako fillStyle u vybarvování obdélníku. Fillstyle tedy nastavuje barvu textu [5].

fillStyle = "barva"

2.3.4 OHRANIČENÍ TEXTU

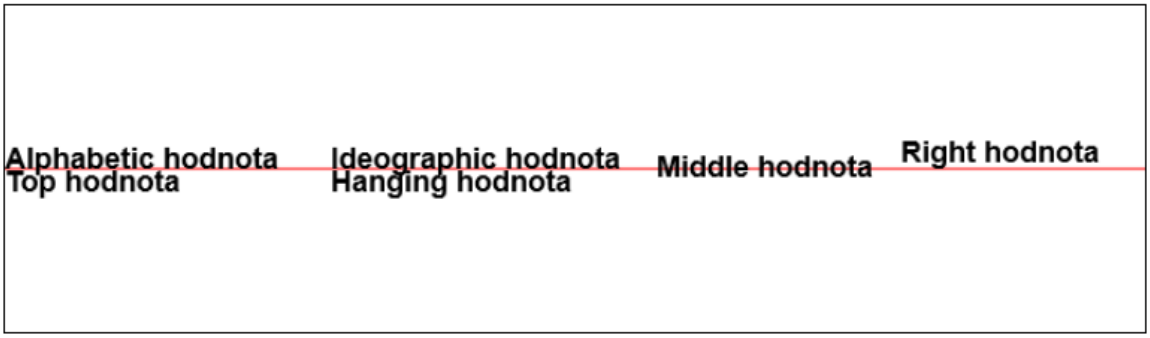
Ohraničení textu funguje na stejném principu jako metoda fillText. Slouží k ohraničení textu [5].

strokeText(“text“,x,y,délka)

2.3.5 VERTIKÁLNÍ ZAROVNÁNÍ TEXTU

Proměnná X může nabývat hodnot top, hanging, middle, alphabetic, ideographic a bottom. Výchozí nastavení je alphabetic. Textbaseline se využívá k vertikálnímu zarovnání textu, viz Obrázek 2: Vertikální zarovnání [7].

textBaseline = "X"

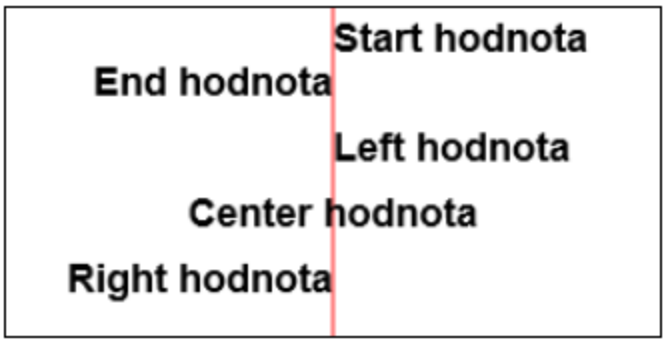


Obrázek 2: Vertikální zarovnání textu

2.3.6 HORIZONTÁLNÍ ZAROVNÁNÍ TEXTU

Proměnná x může nabývat hodnot start, end, left, right a center. Výchozí nastavení je start. TextAlign se využívá k horizontálnímu zarovnání textu, viz Obrázek 3: Horizontální zarovnáni textu [7].

textAlign = "X"



Obrázek 3: Horizontální zarovnáni textu.

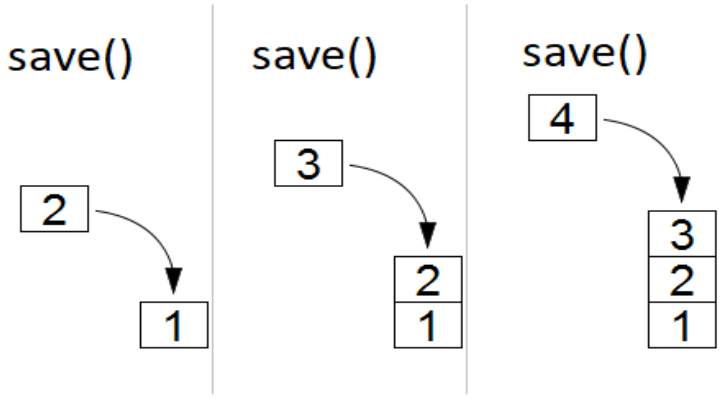
2.4 TRANSFORMACE

Transformace u Canvasu slouží k úpravě plátna. Je možné měnit výchozí nastavení plátna podle vlastních potřeb. Translate, rotate, scale nám umožňují změnit počátek, rotaci a škálování.

2.4.1 UKLÁDÁNÍ A OBNOVA

Před úpravou plátna u Canvasu je zapotřebí nejprve vysvětlit, jak se ukládá a obnovuje nastavení Canvasu. K tomuto slouží metody restore a save. Metoda save uloží celý stav Canvasu do zásobníku. Pokaždé když je metoda volaná, aktuální stav Canvasu se uloží do zásobníku. Metoda save pak ukládá transformace translate, rotate a scale. Dále také ukládá atributy: strokeStyle, fillStyle, globalAlpha, lineWidth, lineCap, lineJoin, miterLimit, lineDashOffset, shadowOffsetX, shadowOffsetY, shadowBlur, shadowColor, globalCompositeOperation, font, textAlign, textBaseline, direction, imageSmoothingEnabled. Zásobník funguje na řazení typu LIFO (poslední dovnitř, první ven), viz Obrázek 4: Metoda save() [5].

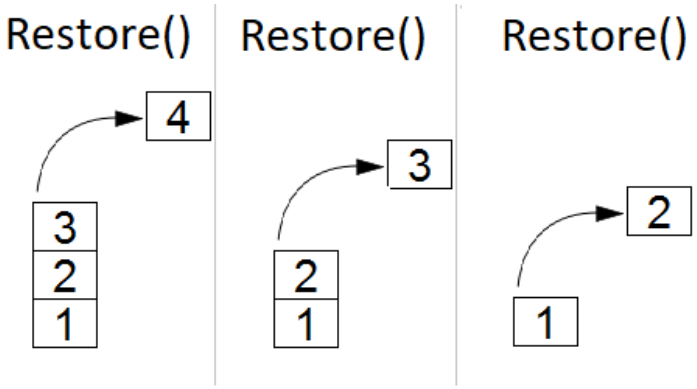
save()



Obrázek 4: Metoda save()

Metoda restore obnoví poslední uložený stav Canvasu, viz Obrázek 5: Metoda restore(). Pokud v zásobníku není uložený ani jeden stav, metoda neudělá nic [5].

restore()



Obrázek 5: Metoda restore()

2.4.2 POSUN

Metoda translate slouží k posunu počátku Canvasu na požadované souřadnice, což nám dává volnost umisťovat útvary na Canvas bez změny jejich souřadnic. Proměnná X udává horizontální posun, Y určuje vertikální posun [7].

translate(x, y)

2.4.3 ŠKÁLOVÁNÍ

Metoda scale se používá při škálování plochy Canvasu. Proměnná X škáluje horizontálně, proměnná Y vertikálně. Do obou proměnných se dosazují reálná čísla. Dosazené hodnoty, které jsou menší než 1, zmenšují velikost a hodnoty vetší než 1 zvětšují. Záporná čísla nám umožnují zrcadlit. Defaultní nastavení je 1 [7].

scale(x, y)

2.4.4 ROTACE

Metoda rotate slouží k rotaci Canvasu. Počátek rotace se nachází v počátku Canvasu. Rotace probíhá ve směru hodinových ručiček, proměnná X je vyjádřena v radiánech [7].

rotate(x)

2.4.5 MATICE

Metoda transform slouží k manipulaci útvaru pomocí transformační matice. Proměnné A a D jsou určeny ke škálování, proměnné B a C pak ke kosení objektu. Pro rotaci objektu vyplníme B a C stejnou absolutní hodnotou. Proměnné E a F slouží pro přesun objektu [7].

transform(a, b, c, d, e, f)

Metoda setTransform funguje na stejném principu jako transform s rozdílem, že předchozí transformační matici zruší, a poté nastaví podle proměnných A B C D E F, to vše v jednom kroku [7].

setTransform(a, b, c, d, e, f)

Metoda resetTransform je užívá k obnově útvaru do původního vzhledu. Pro reset lze využít také setTransform(1, 0, 0, 1, 0, 0).

resetTransform()

2.5 CLIP

Slouží k vystřihnutí požadovaného útvaru z Canvasu. Odstřihnutý útvar může být vytvořen pouze z metody path, nelze jej vytvořit pouze z metody fillRect ani dalších [5].

beginPath()

arc()

clip()

2.6 STÍNY

V HTML5 Canvas je možné přidat stíny na čáru, text, obrázek, čtverec atd., čímž je možné vytvořit 3-D efekt. Shadowoffsetx a shadowoffsety slouží k určení vzdálenosti od objektu, kterému je nastavován stín. Hodnotami mohou být kladná a záporná čísla, která poslouží k určení směru stínu. Výchozí nastavení je 0. Proměnná X udává horizontální posun a Y vertikální posun. Shadowblur nastavuje intenzitu rozmazání stínu. Jako proměnná je nastaveno číslo, přičemž výchozí hodnota je 0 (žádné rozmazání). Shadowcolor určuje barvu stínu [5].

shadowOffsetX = "X"

shadowOffsetY = "Y"

shadowBlur = "Z"

shadowColor = "barva"

2.7 UDÁLOSTI

Události jsou důležitou složkou ECMAScriptu. Umožňují zaznamenat podněty uživatele. Událost může registrovat stisk myši, klávesnice nebo dotyk prstu.

addEventListener("událost", funkce)

Posluchač čeká na událost (kupř. stisk klávesy), poté zavolá funkci, přičemž funkce proběhne pokaždé, nastane-li událost [5].

2.7.1 KLÁVESNICE

Pro ovládání aplikace pomocí klávesnice existují události keydown, keypress, keyup, avšak nejsou podporovány ve všech prohlížečích. Proto doporučuji dopsat před události on [7].

- Událost onkeydown se provede při stisknutí tlačítka;

- Událost onkeypress se provede při delším stisknutí. Pro tlačítka (ALT, CTRL, SIFT, ESC) se nedetekuje událost, proto je lepší provést událost onkeydown;
- Událost onkeyup se provede po uvolnění tlačítka.

2.7.2 MYŠ

Pro ovládání aplikace pomocí myši existuje několik událostí. Základní události jsou kliknutí a pozice myši na aplikaci [1].

- Událost onclick se provede při stisknutí tlačítka na myši;
- Událost oncontextmenu se provede při stisknutí tlačítka na myši;
- Událost onmousemove se provede při přesunu myši na element;
- Událost mouseleave se provede při opuštění elementu myší;
- Někdy je třeba potřeba vědět pozice myši na Canvasu. To je možné zjistit pomocí clientX a clientY, které udávají pozici X a Y.

2.7.3 DOTYKOVÉ OBRAZOVKY

V případě ovládání aplikace pomocí dotykové obrazovky nastává nový problém. Při klasickém ovládání pomocí myši existuje jeden bod dotyku, kterým je kurzor myši. Avšak v případě chytrého telefonu může v jednom okamžiku existovat bodů dotyku hned několik. K tomu slouží touchEvent, touch a touchList [8]. TouchEvent obsahuje tyto události:

- Událost touchstart se provede při přiložení prstu na dotykovou obrazovku;
- Událost touchend se provede při odložení prstu z dotykové obrazovky, nebo při přejetí prstu z obrazovky;
- Událost touchmove se provede při pohybu prstem po dotykové obrazovce;
- Událost touchcancel se provede, pokud uživatel položil více prstů na obrazovku, než kolik je podporováno. Tímtou se první dotyk v touchList zruší. Další možností je vyjetí z okna Canvasu nebo například při spuštění.

Touch slouží k přidání hodnot k jednomu bodu dotyku a obsahuje tyto hodnoty:

- Identifier udává ID pro aktuální dotyk. ID zůstává po celou dobu pohybu po obrazovce;
- ScreenX udává souřadnice X vzhledem k obrazovce;
- ScreenY udává souřadnice Y vzhledem k obrazovce;
- ClientX udává souřadnice X vzhledem k viewportu;
- ClientY udává souřadnice Y vzhledem k viewportu;
- PageX udává souřadnice X vzhledem k celé stránce. Obsahuje horizontální posun;

- PageY udává souřadnice Y vzhledem k celé stránce. Obsahuje vertikální posun;
- Target udává počátek bodu, ve kterém se poprvé prst dotknul obrazovky.

TouchList slouží k zaznamenávání několika dotyků na obrazovce. Obsahuje tyto hodnoty:

- Touches udává seznam všech dotyků na obrazovce;
- TargetTouches udává seznam všech dotyků na elementu.

3 SVG

V roce 1999 vývoj SVG započal a o pár let později byl vydán (2001). Jedná se o vektorový grafický jazyk využívající populární formát pro výměnu dat XML a sloužící pro vykreslení 2-D vektorové grafiky. Tento formát se stal průmyslovým standardem pro přenos vektorové grafiky mezi různými platformami i aplikacemi. SVG neobsahuje žádné pixely, ale seznam svých elementů. Struktura XML umožňuje objekty dynamicky upravovat. Dříve SVG trpěl slabou podporou mezi prohlížeči, není třeba mít webové stránky pro různá rozlišení obrazovek. Díky HTML5 je možné přímo do HTML kódu stránky uložit SVG grafiku, a to pomocí elementu SVG, embed, object, iframe. Element SVG obsahuje několik atributů, přičemž mezi ty nejdůležitější patří viewport a viewbox. Je zapotřebí mít na stránce minimálně jeden párový element SVG. Stejně jako element Canvas, také SVG je párové, lze jej upravovat pomocí CSS atd. Tyto úpravy přitom nezmění vnitřek plátna. [14]

<svg width="200" height="200" viewbox="0 0 200 200">

3.1 SVG PLÁTNO

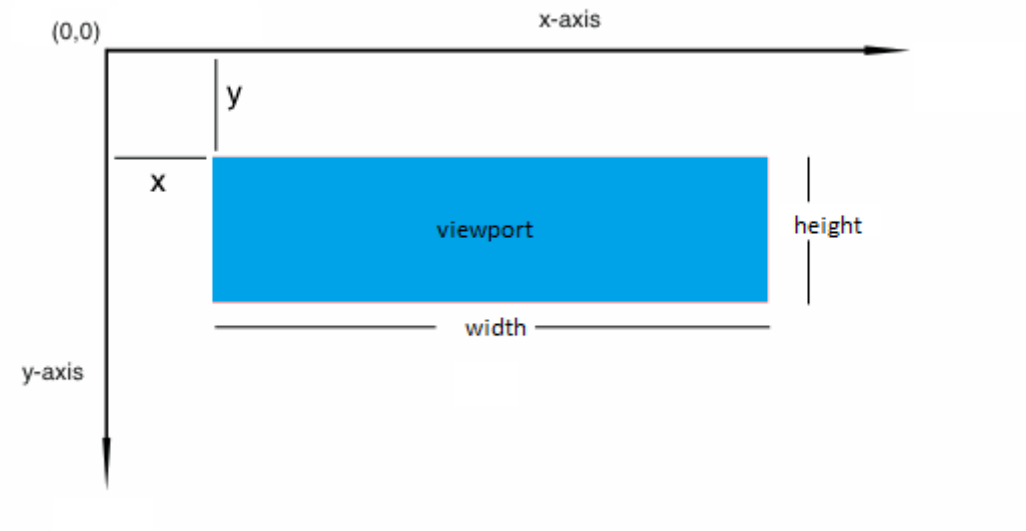
SVG plátno je oblast, kde se zobrazují SVG prvky. Plátno je nekonečné, avšak obrazovky nejsou, proto je nutné mít vyhrazené místo na webové stránce, k čemuž slouží atributy viewport a viewbox, které jsou součástí SVG elementu. Oba atributy jsou nepovinné. V případě, že SVG prvek překročí plátno, nebude zobrazen a bude uříznut. Viewbox nám vyváří nový systém souřadnic uvnitř plátna. Pod pojmem viewport si můžeme představit chytrý telefon, který zobrazuje webové stránky pro počítač v mobilním prohlížeči v omezeném rozlišení (výřez). Celá stránka je nekonečná a viewbox umožňuje přibližovat, oddalovat či posouvat stránku [3].

Viewport se skládá z width (šířka) a height (výška).

<svg width="200" height="200"></svg>

Viewbox má čtyři atributy. X a Y udává začátek viewboxu, X-AXIS udává šířku a Y-AXIS udává výšku.

<svg width="200" height="200" viewBox="x y x-axis y-axis"></svg>



Obrázek 6: Plátno SVG

3.2 Základní tvary

Technologie SVG je vhodná pro tvorbu 2-D grafiky. Základní jednotkou je matematický vektor. Pro rychlejší a jednodušší práci jsou v SVG předdefinovány základní tvary (obdélník, úsečka, kruh, elipsa, čára lomená a mnohoúhelník). Tyto základní tvary jsou ekvivalentní elementu path. Můžeme tak pomocí path nakreslit jakýkoliv tvar a díky této vlastnosti některé desktopové vektorové editory převádějí základní tvary na element path.

3.2.1 Obdélník

Počátek (levý horní bod) obdélníku na plátně se určuje pomocí souřadnic X a Y. Atribut width udává šířku obdélníku v pixelech a height výšku. Atributy RX a RY určují zaoblenost obdélníku [2].

<rect X="" Y="" width="" height="" RX="" RY=""/>

3.2.2 Úsečka

Atributy X1 a Y1 značí počáteční bod, X2 a Y2 pak konečný bod úsečky.

<line X1="" Y1="" X2="" Y2=""/>

3.2.3 Kruh

Střed oblouku je na plátně určen pomocí CX a CY, R je poloměr kruhu [2].

<circle CX="" CY="" R=""/>

3.2.4 Elipsa

Střed elipsy se na plátně určí pomocí CX a CY, RX a RY znamenají poloměr elipsy [2].

<ellipse CX="" CY="" RX="" RY=""/>

3.2.5 Čára lomená

Jedná se o útvar složený z několika úseček bez mezery. Do atributu points ukládáme souřadnice úseček, kterých může být mnoho [4].

<polyline points="x1,y1 x2,y2 x3,y3 x4,y4 " />

3.2.6 Mnohoúhelník

Útvar složený minimálně ze tří vrcholů úseček, které se ukládají do atributu points [4].

<polygon points="x1,y1 x2,y2 x3,y3"/>

3.2.7 Cesta

Element path slouží k vytváření složitějších grafických útvarů. Důležitý je atribut d, který udává styl cesty. Atribut d je case sensitive. Velká písmena značí absolutní souřadnice, malá písmena pak relativní souřadnice k předchozímu. Například při vytváření čáry jsou zapotřebí dvě souřadnice [5,5] a [10,10], přičemž v absolutních souřadnicích budou souřadnice [5,5] [10,10], v relativních [5,5] [15,15] (tedy relativní k předchozím souřadnicím) [2].

- M/m (Move To): Přesun na souřadnice;
- L/l (Line To): Vykreslení úsečky z aktuálních souřadnic k daným souřadnicím;
- H/h (Line To): Vykreslení úsečky horizontálně;
- V/v (Line To): Vykreslení úsečky vertikálně;
- C/c (Cubic Bézier Curve): Bézierova křivka;
- S/s (Cubic Bézier Curve): Bézierova křivka vycházející z předchozí Bézierovy křivky;
- Q/q (Quadratic Bézier Curve): Kvadratická křivka;
- T/t (Quadratic Bézier Curve): Kvadratická křivka vycházející z předchozí kvadratické křivky;
- A/a (Elliptical Arc Curve): Vykreslení oblouku z aktuálních souřadnic k daným souřadnicím;
- Z/z (ClosePath): Uzavření cesty.

<path d=" "/>

3.3 TEXT

SVG se využívá převážně ve webové grafice (tlačítka atd.), bylo by tedy záhodno umožnit vkládání textu přímo do grafiky. Bohužel tato funkce nenabízí mnoho možností. Kurzorem myši lze označit/vybrat a zkopírovat text, který vykreslujeme do SVG elementu.

3.3.1 ŘETĚZEC

Proměnná text určuje text. X a Y jsou souřadnicemi počátku prvního znaku v řetězci (levý dolní roh). Atributy X a Y jsou nepovinné. Výchozí nastavení je 0. Je možné nastavit souřadnice pro každé slovo v řetězci. V elementu text je definován také element tspan, který slouží k určení části řetězce, který chceme formátovat jinak [2].

<text x="" y="">text</text>

3.3.2 ŘETĚZEC NA KŘIVCE

SVG umožňuje vykreslit řetězec textu na jakoukoliv křivku (path). Tedy pro uchopení textu na křivku je potřeba vytvořit element path a určit mu ID, které slouží pro identifikaci křivky. Pro vykreslení se využívá prvek textPath, který je vnořen do textu. xlink:href="# " je povinný atribut a slouží k identifikaci křivky. Dále jsou nepovinné atributy startOffset = "" (udává vzdálenost od počátečního bodu), spacing = "" (udává mezery mezi písmeny), method = "" (udává přizpůsobení textu na křivku) [2].

<textPath xlink:href="# ">

Text na křivce

</textPath>

3.4 ELEMENT G

Párový element g slouží k seskupování grafických elementů. Důležitým atributem je ID. Díky němu můžeme upravovat celou skupinu elementů najednou [2].

<g id="">

<text x="" y="">text</text>

<polygon points="x1,y1 x2,y2 x3,y3"/>
<rect x="" y="" width="" height="" rx="" ry=""/>

</g>

3.5 VYBARVOVÁNÍ

SVG umožnuje grafickým elementům různé funkce. Je možné změnit průhlednost, barvy, vyplňovat texturou atd. Každý grafický element má dva atributy, jsou jimi fill (vnitřní výplň) a stroke (okraj).

3.5.1 VÝPLŇ JEDNOU BARVOU

Stejně jako u Canvasu můžeme za proměnou barva dosadit různé hodnoty (blue, white, #FF0000 atd.).

<rect x="" y="" width="" height="" rx="" ry="" style="fill:barva;stroke:barva"/>

3.5.2 PŘECHOD BAREV

Přechody se definují pomocí elementu linearGradient (horizontálně, vertikálně) a radialGradient (radiálně). Přechody se definují v elementu defs a odkazujeme na ně pomocí ID. Směr stupňující se barvy záleží na proměnných X1, Y1, X2, Y2. Element stop slouží k určování pozic barev a jejich prolínání. Pro vytvoření přechodu barev potřebujeme minimálně dvě barvy [2].

<linearGradient id="prechod0" x1="%" y1="%" x2="%" y2="%">

<stop offset="0%" style="stop-color: barva " />

<stop offset="100%" style="stop-color: barva" />

</linearGradient>

3.5.3 VLOŽENÍ VZORU

Pro vyplnění vzorem je zapotřebí deklarovat element pattern, do kterého vložíme prvek, který se bude opakovat, a určíme mu ID pro budoucí odkazování. Proměnné X a Y udávají polohu, atribut width určuje šířku, height pak výšku [2].

<pattern id="vzor" x="0" y="0" width="20" height="20" patternUnits="userSpaceOnUse">

<circle cx="10" cy="10" r="10" style="fill:#4bbee3;stroke:#4bbee3;"/>

</pattern>

3.6 TRANSFORMACE

Pomocí atributu transform, který může nabývat hodnot matix, translate, scale, skewX/Y, můžeme snadno a rychle upravovat souřadnice objektu (circle, ellipse, g, line, path atd.), není tedy potřeba náročné dopočítávání souřadnic u složitějších útvarů (path). Tyto transformace je možné řetězit za sebe a provést několik transformací najednou. Vhodné je nastavit si napřed všechny objekty na souřadnice [0;0], a poté dle potřeby transformovat. Změny, které provedeme, se provedou i na děti

3.6.1 POSUN

Jedná se o posunutí objektu na souřadnice X, Y. Proměnná Y není povinná, výchozí nastavení Y je 0 [2].

transform="translate(x y)"

3.6.2 ŠKÁLOVÁNÍ

Změna měřítka X a Y souřadnic. Proměnná Y není povinná, výchozí nastavení Y se rovná X [2].

transform="scale(x,y)"

3.6.3 ROTACE

Rotace objektu podle proměnné X. Proměnné A a B slouží k určení bodu rotace. Pokud nezadáme žádné hodnoty, dosadí se hodnoty počátku útvaru [2].

transform="rotate(x,a,b)"

3.6.4 ZKOSENÍ

Zkosení objektu podle směru osy x a y. Za proměnnou X se dosadí číslo ve stupních [2].

transform="skewX(X)"

transform="skewY(X)"

3.6.5 MATICE

Úprava objektu pomocí transformační matice. Proměnné B a C slouží ke kosení objektu. Pro rotaci objektu vyplníme B a C stejnou absolutní hodnotou. Proměnné E a F pak slouží pro přesun objektu. Proměnné A a D se užívají ke škálování [2].

transform="matrix(a b c d e f)"

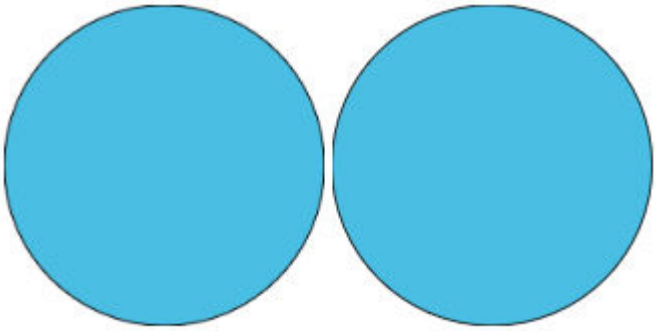
4 POROVNÁNÍ TECHNOLOGIÍ

Canvas poskytuje pouze místo, do kterého kreslíme, přičemž veškeré kreslicí funkce jsou prováděny pomocí ECMAScriptu. To znamená, že výsledný produkt je bitmapa, která se skládá z pixelů. Díky tomuto prohlížeč nedokáže přesně určit, co je na plátně zobrazeno, čímž se zhoršuje přístupnost webových stránek. Bitmapa Canvasu není součástí DOM, proto nelze upravovat vnitřek plátna pomocí HTML ani CSS. Do Canvas plátna je možné nahrát různé formáty bitmapových obrázků (JPG atd.), následně je upravovat pomocí různých filtrů (odstranění červených očí, ořezů aj.) a po úpravě je ukládat na server/desktop. Tímto způsobem je šetřen výkon serveru, jelikož veškeré úpravy probíhají u klienta a není tak nutné při každé změně přeposílat obrázek mezi serverem a uživatelem. To je výhodné v případě mobilních zařízení. Menší zátěž zde znamená menší spotřebu baterie. Jelikož plátno Canvasu je bitmapou, jeho výpočetní náročnost nenarůstá se složitostí nakreslených objektů. Příkladem můžou být různé topografické nebo všeobecně zeměpisné mapy, na kterých je zobrazeno velké množství tvarů (města, silnice a další). Canvas však není vhodné užívat pro věci, které vyžadují interaktivitu s uživatelem. Příkladem je uživatelské rozhraní, grafy atd.

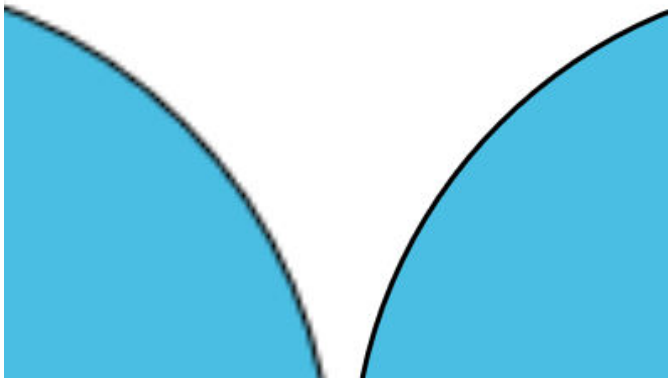
Jak již bylo zmíněno v předchozích kapitolách práce, SVG slouží k tvorbě 2-D grafiky v XML. Každý element, který vytvoříme pomocí SVG, je přístupný přes SVG DOM API, což znamená, že je upravitelný pomocí CSS, ECMAScriptu. Navíc lze také všechny elementy označkovat pro lepší přístupnost webových stránek. Podle stránek https://www.statista.com[1] bylo v letech 2018 a 2019 přes 50 % přístupů provedeno pomocí mobilních zařízení. Pokud je webová stránka pro návštěvníka nečitelná, ihned web opustí. Z toho vyplývá, že je nutné webové stránky optimalizovat pro více zařízení s rozdílným rozlišením (responzivní). SVG není závislé na rozlišení, viz obrázky číslo 2 a 3, čímž je lépe využito u multiplatformních uživatelských rozhraní, grafů, plánků budov atd. Další výhodou SVG je transformace. Není potřeba složité dopočítávání souřadnic u složitějších útvarů, jde pouze o drobné změny již vykreslených objektů. ři vytváření složitějších útvarů osobně doporučuji využít možnost desktopového editoru. K dispozici

[1] Mobile share of website visits worldwide 2018 | Statistic. *Statista – The Statistics Portal for Market Data, Market Research and Market Studies* [online]. Copyright © Statista 2019 [cit. 24. 03. 2019]. Dostupné z: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/

jsou volně dostupné programy, které sice nenabízí takové rozmanité funkce, avšak v určitých částech se těm placeným dozajista vyrovnají (Inkscape).



Obrázek 7: Porovnáni Canvasu(v levo) a SVG (v pravo)



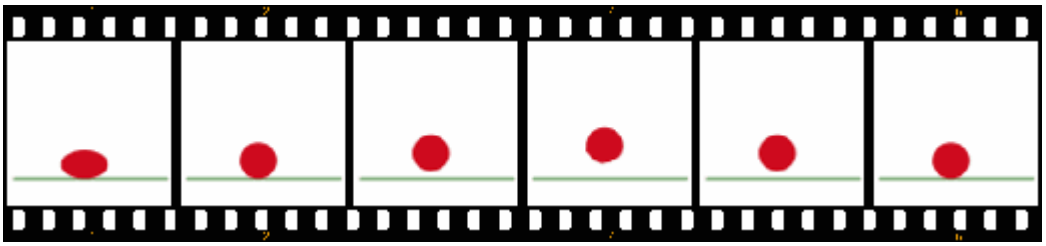
Obrázek 8: Porovnáni Canvasu (vlevo) a SVG (vpravo) v zoomu

5 ANIMACE

V následujících kapitolách bude popsáno, jakým způsobem je možné vytvářet animace pro webové stránky pomocí technologií Canvas a SVG. Dnešním uživatelům webových stránek stačí pár vteřin na rozhodnutí, jestli chtějí zůstat na stránce nebo odejít. Spousta uživatelů je také velice netrpělivá, a proto je velmi vhodné jim během čekání na odezvu ze serveru ukázat indikátor průběhu (přesýpací hodiny, načítací panel atd.). V současné době se animace hojně používají při tvorbě internetových reklam (např. rolovací bannery). Princip animace představuje záznam sekvence na sebe navazujících snímků, které jsou každý sám o sobě statické a drobně se od sebe liší. Při rychlém zobrazování těchto snímků postupně za sebou vzniká díky setrvačnosti lidského oka dojem plynulého pohybu. Snímky se však musí přehrávat takovou rychlostí, kterou oko nepostřehne.[2]

5.1 SPRITE

U dvourozměrné animace je možné využít sprite, viz Obrázek 5. Sprite je malý obrázek povětšinou s prázdným pozadím, který je poté vložen do plátna. Sada více spritů, která popisuje animaci, se nazývá sprite sheet. Animace dosáhneme rychlým promítáním jednotlivých obrázků. Tento princip využívaly starší počítačové hry. Pro animaci charakteru bylo použito několik spritů na jednotlivé části těla. I dnes se tento styl používá při tvorbě webových her, případně pro zrychlení webové stránky, kdy se všechny obrázky vloží do jednoho společného, díky čemuž se webová stránka urychlí [5, 7].



Obrázek 9: Části animace zdroj

5.2 CANVAS

Tato kapitola bude věnována popisu metod animací u Canvasu. Canvas se animuje pomocí ECMAScriptu. Díky tomu je možné jistým způsobem si hrát s fyzikou, kolizí atd. Animace se u Canvasu nazývají frame-based, což znamená, že daná animace je složená z několika snímků spuštěných rychle za sebou.

[2] Animace – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Animace

5.2.1 SMYČKA

Jak již bylo popsáno dříve, animace je záznamem množství snímků na sebe navazujících. Docílit toho můžeme pomocí neustálého vykreslování požadovaného útvaru. Pohybu útvaru pak dosáhneme tak, že změníme jeho souřadnice při vykreslování. Při tomto animování však nastává problém, kdy je neustále překreslováno původní plátno, takže je zapotřebí metoda na vyčištění plátna. Vyčištění plátna se dá vyřešit několika způsoby. Nejjednodušší způsob je celý Canvas smazat (přepsat například bílou barvou) a znovu vykreslit útvar. V případě, že se jedná o příliš složité pozadí, je výhodnější určit si malou oblast na Canvasu, kterou budeme mazat a následně vykreslovat znovu. Poslední možností, jak tento problém vyřešit, je použití blittingu. Na blitting vytvoříme nový Canvas. Pozadí, které vykreslíme na první Canvas, uložíme do sekundárního Canvasu. Pomocí sekundárního Canvasu pak budeme překreslovat původní Canvas. To nám ušetří zbytečné vykreslování. Nejlepší způsob je však tyto metody kombinovat [7].

Základní struktura animační smyčky u Canvasu vypadá následovně:

- Vykreslení útvaru na Canvas;
- Smazání Canvasu.

5.2.2 KOLIZE

Kolize a její detekce je důležitou částí programu. Při vývoji programu bude zapotřebí, aby objekty mezi sebou pracovaly, při aplikování gravitace nepropadly mimo Canvas atd. Je tedy nutné zjistit, zda se dva tvary překrývají. Ve 2-D prostředí se využívá testování kolize krajů. -Toho dosáhneme testováním pozic krajů před provedením pohybu, tedy netestujeme aktuální pozici tvaru. Také je nutné dávat si pozor na příliš rychlé pohyby objektu, jelikož může nastat situace, při které proměnné přeskočí překážku, což způsobí chybné chování [7].

5.2.3 FRAME RATE

Animace je sekvence unikátních obrázků, které jsou postupně za sebou zobrazovány. Animace tvoří dojem pohybu příslušnou rychlostí. Počet takových obrázků udává frame rate, který se měří ve snímcích za sekundu (fps). Minimální hodnotou pro plynulou grafiku je alespoň 24 fps. Dostačující hodnota je 60 fps [5].

5.2.4 INTERVAL

Metoda setInterval opakovaně volá anonymní funkci po uplynutí času t a vrací identifikační číslo. Proměnná t udává čas v ms [1].

setInterval(funkce(){tělo funkce }, t)

Metoda clearInterval zastaví běžící interval. Metodu setInterval uložíme do proměnné, kterou poté voláme v metodě clearInterval [1].

clearInterval(proměnná)

5.2.5 TIMEOUT

Metoda setTimeout funguje na podobném principu jako metoda setInterval, avšak spustí anonymní funkci jednou po uplynutí času T [1].

setTimeout(funkce(){tělo funkce }, t)

Metoda clearTimeout zastaví bežící interval pro spuštění anonymní funkce. Metodu setTimeout uložíme do proměnné, kterou poté voláme v metodě clearTimeout.

Metody setInterval a setTimeout běží nezávisle na hlavním programu [1].

clearTimeout(proměnná)

5.2.6 REQUESTANIMATIONFRAME

Metoda requestAnimationFrame opakovaně volá anonymní funkci šedesátkrát za vteřinu. Animace se přestanou provádět v neaktivních barech prohlížeče, tedy narozdíl od metody setInterval. Díky tomu je šetřena spotřeba v mobilních zařízeních [7].

requestAnimationFrame(tělo funkce)

Metoda cancelAnimationFrame zastaví běžící interval. Metodu requestAnimationFrame uložíme do proměnné, kterou poté voláme v metodě cancelAnimationFrame [7].

cancelAnimationFrame(proměnná)

5.2.7 HODNOCENÍ CANVAS

Pro:

- Můžeme vytvořit velice složité animace bez zvýšení náročnosti na zařízení;
- Tvorba her.

Proti:

- Nutná znalost ECMAScriptu;
- Není možné přesně určit, co je zobrazeno na plátně, čímž se zhoršuje přístupnost webových stránek;
- Složitější přizpůsobení animace pro responzivní web;
- Složitější kontrola chyb.

5.3 SVG

V této kapitole se budu věnovat popisu metod animací u SVG. Animace u SVG se nazývají deklarativní animace, což znamená, že můžeme snadno animovat grafické prvky, aniž bychom potřebovali využívat scripty, avšak je také možné animovat SVG pomocí ECMAScriptu. Tento způsob se nazývá time-base Díky tomu můžeme podle potřeby (okamžitě, po určité době) změnit jednotlivé parametry prvků (průhlednost, polohu atd.). Další důležitou vlastností deklarativních animací je nezávislost. Tento způsob animace je kompatibilní s CSS3 a standardem SMIL.

5.3.1 ANIMATE

Atribut attributeName slouží k určení elementu, který chceme animovat. Atribut attributeType určuje jmenný prostor (CSS, XML). Výchozí hodnota je auto (testuje se, jestli není atribut používán v CSS nebo XML). Tyto dva atributy, kromě elementu animate, využívají také elementy animateTransform a set. Atribut from slouží k určení počátečního bodu animace. Atribut to značí koncový bod animace. Atribut dur slouží k určení trvání animace, atribut repeatCount slouží k určení počtu opakování animace. Hodnotou, kterou můžeme do atributu dur nebo repeatCount dosadit, je buď číslo, nebo slovo indefinite (nekonečné trvání) [2].

<animate attributeName="x" attributeType="XML" from="" to="" dur="" repeatCount=""/>

5.3.2 ANIMATEMOTION

Element animateMotion slouží k pohybu elementu podél předdefinované křivky. Křivka může být pouze element path. Atributy dur a repeatCount již byly popsány výše. Pomocí atributu keyTimes je možné upravovat rychlost elementu. Atribut rotate slouží k rotaci elementu. Hodnoty, které můžeme dosadit, jsou auto (element se bude otáčet automaticky podle směru path), auto-reverse (totéž, avšak element je otočený o 180°),

číslo (značí úhel oproti ose x). Výchozí nastavení je 0. Atribut calcMode určuje interpolaci [2]. Hodnoty, které nabývá, jsou:

- discrete: Nepoužije se žádná interpolace při přechodu hodnot;
- linear: Použije se lineární interpolace při přechodu hodnot. Výchozí hodnota pro elementy animate a animateTransform;
- paced: Použije se interpolace konstantním krokem pouze tehdy, pokud lze vypočítat vzdálenost mezi body. Atributy keyTimes a keySplines jsou ignorovány. Výchozí hodnota pro element animateMotion;
- spline: Průběh je určen pomocí Bézierovy křivky. Body jsou definovány v atributu keyTimes, keySplines definuje kontrolní body pro každý interval.

5.3.3 ANIMATETRANSFORM

Element animateTransform umožňuje animovat atributy, čímž je možné provádět rotace, zkosení, změny velikosti. Atribut attributeName musí být pokaždé nastaven s hodnotou transform. Atribut type slouží k určení transformace (rotate, scale, skewx, skewy, translate). Atributy from a to již byly popsány výše [4].

<animateTransform attributeName = "" attributeType = "" type = "" from = "" to = "" dur = "" repeatCount = ""/>

5.3.4 SET

Element set je jednodušší verzí elementu animate. Slouží k nastavení hodnot v definovaném čase. Například pro skrytí viditelnosti, změnu textu atd. Atribut to slouží k určení hodnoty, která bude nastavena po dobu animace. Atributy from jsou zde ignorovány [4].

5.3.5 BEGIN

Atribut begin definuje začátek animace. Nabývá hodnot (offset-value, event-value atd.) oddělených středníky. Jedna animace tedy může být spouštěna hned několika událostmi najednou [4].

- offset-value: Slouží ke spuštění animace po načtení DOMready události. Za hodnotu dosazujeme číslo, které udává počet vteřin k aktivování animace;
- syncbase-value: Umožňuje synchronizovat dvě animace. Pokud jedna animace skončí či začne, pak se po námi určené době spustí další animace. Důležité animační elementy jsou begin nebo end. Například begin="animace.start+5s";
- event-value: Spouští animaci po určené události (activate, click, mousedown atd.) Například begin="animace.click";

- repeat-value: Cyklické spouštění jedné animace. Můžeme určit počet opakování. Například begin="animace.repeat(5)";
- accessKey-value: Spouští animaci po stisku klávesy. Například begin="animace.accesKey('a')";
- wallclock-sync-value: Spuštění animace v přesně daný čas, a to dle formátu ISO8601.

5.3.6 RESTART

Atribut restart udává, kdy může být animace spuštěna od začátku.

- Always: Animace může být spuštěna od začátku pokaždé. Tato hodnota je výchozí;
- whenNotActive: Animace může být spuštěna od začátku, pokud není aktivní;
- Never: Animace nemůže být spuštěna od začátku.

5.3.7 HODNOCENÍ SVG

Pro:

- Snadné přizpůsobení pro responzivní web;
- Není nutná znalost ECMAScriptu;
- Snadná tvorba animací pro web;
- Prvky jsou součástí DOM.

Proti:

- Náročnější na výpočet v případě složitějších animací.

6 KNIHOVNY

Knihovna je soubor, který se skládá z různých metod, které spolu úzce souvisí. Cílem knihoven je usnadnění tvorby zdrojového kódu, díky tomu je vývoj aplikací rychlejší, tvorba grafů i počítačových her snazší atd. Jak již bylo zmíněno, knihovna obsahuje metody, které může aplikace volat k vykonání určitého úkolu. Právě tento princip urychluje vývoj aplikací, poněvadž programátor nemusí vytvářet potřebné metody znovu od začátku, ale může rovnou použít již vytvořená řešení.

6.1 KRITÉRIA PŘI VÝBĚRU KNIHOVEN

Jelikož si může knihovnu vytvořit každý jedinec, rozhodl jsem se výběr knihoven snížit pomocí kritérií, jimiž jsou vizualizace, dokumentace, příklady, návody a počet hvězdiček na GitHubu.

6.1.1 VIZUALIZACE

Hlavním kritériem jsem určil vizualizaci, a to vzhledem k tématu bakalářské práce.

6.1.2 DOKUMENTACE /PŘÍKLADY/NÁVODY

Knihovna obsahuje metody, které může aplikace volat k vykonání určitého úkolu, proto je důležité, aby i ostatní programátoři pochopili, jak funguje daná knihovna a viděli také, co dokáže. K tomuto slouží dokumentace, příklady nebo návody.

6.1.3 ROZSAH KNIHOVNY

Důležitým faktorem pro programátory je, jaké daná knihovna umožňuje funkce. Někomu stačí pouze jednoduché grafy, jiný potřebuje složitější druhy.

6.1.4 POČET HVĚZDIČEK NA GITHUBU

GitHub je webová služba, která podporuje vývoj programů za pomocí verzovacího nástroje Git. Git je distribuovaný systém správy verzí. GitHub nabízí zdarma webhosting pro open source projekty. Díky tomu jsem určil také kritérium pro volně dostupné ECMAScriptové knihovny. Jedno z kritérií, na které se programátor zaměří, je počet stažení. To jsem u knihoven zjistil právě pomocí získaných hvězd na GitHubu. Hvězdy jsou veřejně k vidění u každého projektu. Hvězdu projektu může dát každý registrovaný uživatel. Počet hvězd se vztahuje ke dni 24. 6. 2019 [15].

6.2 SVG KNIHOVNY

6.2.1 D3

Knihovna slouží k vizualizaci dat. Úkolem této knihovny je tedy urychlit práci s daty. Knihovna se stále vyvíjí, avšak byla vydána v roce 2011. Knihovna umožňuje vložit data do DOM, tedy vložit čísla z pole do HTML tabulky, nebo ta samá data použít k tvorbě SVG baru. Dokumentace je psaná v angličtině, ale je také neoficiálně přeložena do několika jazyků, například do ruštiny, turečtiny, bohužel však do češtiny přeložena není. Dokumentace se skládá z několika kapitol (introduction, API reference, gallery atd.) a je velmi detailně popsaná. Příklady na webových stránkách knihovny d3 jsou velice rozsáhlé, obsahují různé grafy, mapy. Návody pro knihovnu d3 na webových stránkách jsou také bohaté, čítají několik desítek návodů, včetně několika knížek, online kurzů a videí [16]. Počet hvězd: 85,483.

6.2.2 BONSAIJS

BonsaiJS je ECMAScriptová knihovna s intuitivním grafickým API. Knihovna umožňuje tvorbu grafů, manipulaci s elementem path, přidání zvuků a videí, a také tvorbu animací. Dokumentace je v angličtině, obsahuje několik kapitol (overview, API modules/classes/mixins) a je detailně popsaná. Ani jeden příklad však na webových stránkách nefungoval. Počet hvězd: 1,971.

6.2.3 RAPHAËL

Raphaël slouží k tvorbě SVG grafů, rotaci, odstřihnutí, škálování. Využívá SVG pro většinu prohlížečů, pro starší prohlížeče pak VML. Raphaël je využíván na více než 3 000 webových stránkách. Dokumentace je velice rozsáhlá a psaná pouze v angličtině. Příkladů na webových stránkách je několik, bohužel nejsou detailně popsány [17]. Počet hvězd: 10,255.

6.2.4 PAPER

Paper slouží k vytváření SVG grafiky převážné spojené s Bézierovou křivkou a její následnou animaci. Dále umí konvertovat bitmapové obrázky na SVG. Dokumentace je velmi nepřehledná a nekompletní. Chybí detailnější popis funkcí. Příklady, které jsou na webových stránkách, jsou detailně popsané [18]. Počet hvězd: 10,643.

6.2.5 HIGHCHARTS

Knihovna je založena na SVG se zpětnou podporou VML a Canvas pro starší verze prohlížečů. Byla vydána v roce 2009. Knihovna obsahuje hightcharts (tvorba grafů), highstock (tvorba grafů pro akcie), highmaps (tvorba interaktivních map), highcharts cloud (sdílení grafů pomocí cloudu). Příklady a dokumentace jsou detailně popsány. Dema příkladů je možné otevřít v CodePenu nebo JSFiddle [19]. Počet hvězd: 8,765.

6.3 CANVAS KNIHOVNY

6.3.1 CHART

Chart je komunitní knihovna, která umožňuje osm typů grafů (spojnicový graf, výsečový graf, paprskový graf atd.). Všechny grafy jsou animovatelné. Úkolem této knihovny je tedy usnadnit práci pro vytváření grafů pro element Canvas. Dokumentace je podrobně vysvětlená s příkladem. Počet hvězd: 44,060.

6.3.2 JAVASCRIPT INFOVIS TOOLKIT

Tato knihovna slouží převážně k vytváření pokročilejších animací stromových map, výsečového grafu, hyperbolického zobrazení. Několik vizualizací umožňuje vizualizovat vztahy mezi různými uzly hierarchických dat. Dokumentace chybí. Ukázky příkladů mají základní popis v komentářích kódu [21]. Počet hvězd: 1,463.

6.3.3 PIXIJS

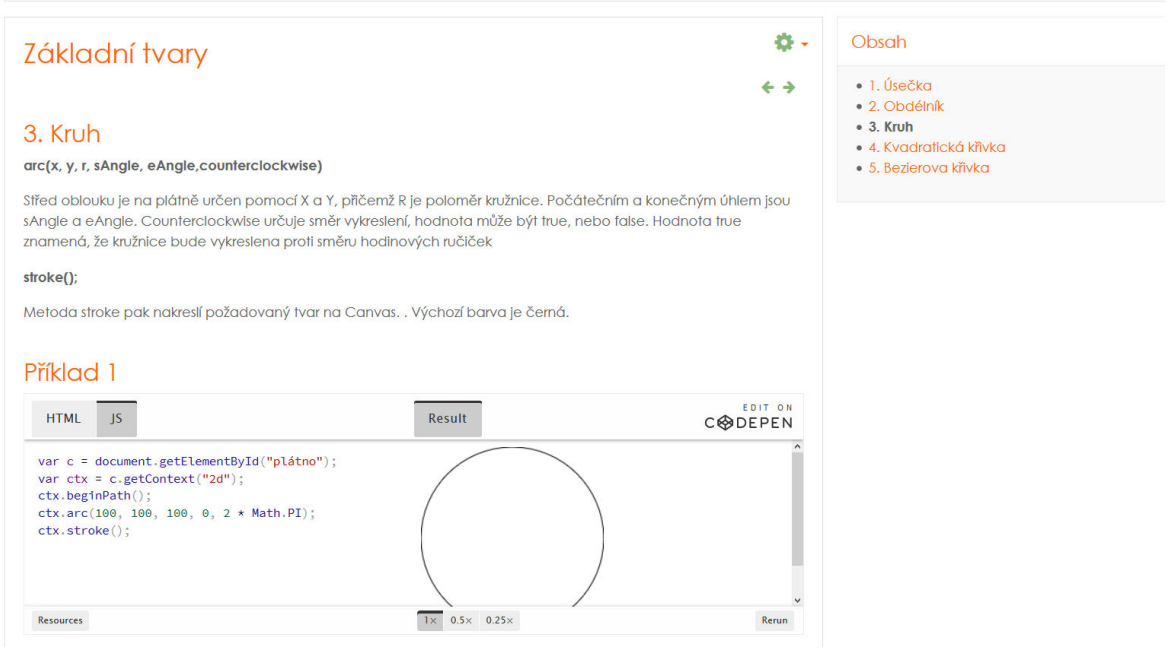
PixiJS nabízí vysokoúrovňové API k vykreslování 2-D grafiky na Canvas pomocí WebGL s fallbackem na 2-D Canvas. Vykreslování pomocí knihovny je velice rychlé. Dále k vykreslování obsahuje animaci Spine, hierarchický model vykreslování spritů, filtry. Dokumentace je detailně popsaná s příklady [22]. Počet hvězd: 24,975.

6.3.4 PLOTLY

Plotly nabízí tvorbu grafů. Grafy mají propracovaný vzhled a působí profesionálním dojmem. Návodů a příkladů je nespočet a všechny jsou detailně rozpracovány [23]. Počet hvězd: 10,368.

7 PRAKTICKÁ ČÁST

Tato kapitola je věnována představení příkladů vytvořených pro online kurz a popis e-kurzu. Hlavním důvodem pro vytvoření příkladu na webové stránce https://codepen.io/ je snadná integrace do learning management systému Moodle, editace a okamžitá viditelnost výsledného programu.



Obrázek 10: Ukázka kurzu (základní tvary, kruh)

7.1 POPIS KURZU

Tento elektronický kurz ukazuje studentům pohled do základů dvourozměrné grafiky, jak vektorové, tak bitmapové, a její následné animace. Při tvorbě příkladu bylo počítáno se znalostí skriptovacího jazyka ECMAScript.

7.2 CODEPEN

CodePen je vývojářské sociální prostředí. Umožňuje psát kód přímo v prohlížeči a zároveň rovnou vidět výsledek. CodePen je zaměřený převážně na HTML, CSS, ECMAScript. Přibližně je zde registrováno 330 000 uživatelů a měsíčně je zaznamenáno zhruba 16,9 milionů návštěvníků. Největší výhodou použití CodePenu je využití funkce Embed. Embed funkce slouží k vyexportování HTML, CSS a ECMAScriptu do HTML kódu, který je následně možné vložit do Moodle.

7.3 MOODLE

Moodle je akronymem pro Modular Object-Oriented Dynamic Learning Environment (modulární objektově orientované dynamické prostředí pro výuku). Moodle slouží coby softwarový balík pro podporu výuky. Tento balík je poskytován bezplatně pod licencí GNU. V prostředí Moodlu je k dispozici celá řada modulů, díky nimž vytváříme podporu pro obsah. Do online kurzu tedy můžeme vkládat ankety, fóra, složky, HTML příkazy, úkoly atd. Online kurz je provozován v univerzitním prostředí Moodle. S tímto řešením se mohou studenti připojit do kurzu pomocí STAGu.

7.4 KAPITOLY KURZU

Celý kurz je uspořádán do čtyř témat, které tvoří další podkapitoly.

7.5 POUŽITÉ MODULY

- Stránka: Modul Stránka umožňuje učiteli vytvářet webové stránky pomocí textového editoru. Stránka může zobrazit text, obrázky, zvuk, video, webové odkazy a vložený kód, jako jsou například Google mapy;
- Fórum: Tento modul umožňuje komunikaci student-student a učitel-student. Mohou zde být řešeny dotazy, sdílena vylepšení pro kurz atd.;
- Složka: Modul slouží k zobrazení několika souborů v jedné složce, čímž pomáhá snížit potřebu rolování na hlavní stránce kurzu. Lze nahrát větší množství souborů pomocí ZIPu. V e-kurzu toho využívám k ukládání a archivaci obrázků;
- Kniha: Umožňuje vytvořit vícestránkový studijní materiál, ve kterém se nacházejí kapitoly a podkapitoly s obsahem. Tento modul je využíván nejčastěji. Lze si také celou knihu vytisknout.

7.6 CÍLE KURZU

- Vytvořit si přesnější představu o možnostech SVG a Canvasu;
- Seznámit se s hlavními principy, pojmy v oblasti webové grafiky a její následné animace.

7.7 GENERAL

První téma nazvané General slouží ke stručnému popsání e-kurzu a jeho cílů. Toto téma obsahuje úvod, použité zdroje.

7.8 CANVAS

Téma slouží k pochopení základních kreslicích funkcí elementu Canvas a následné animace. Téma obsahuje sedm knih (Základní tvary, Vybarvování, Práce s textem, Transformace, Animace, Události, Webové toolbary) a jednu stránku (Plátno).

7.8.1 PLÁTNO

Kapitola plátno tvoří stránka, která se věnuje popisu elementu Canvas. Aby měl student lepší představu o velikosti plátna, ohraničil jsem jej pomocí CSS.

7.8.2 ZÁKLADNÍ TVARY

Kniha Základní tvary obsahuje 5 kapitol (Úsečka, Obdélník, Kruh, Kvadratická křivka, Bézierova křivka).

- V kapitole Úsečka se věnuji popisu metod beginPath, moveTo, lineTo a stroke. Příklad 1 je ukázkou vykreslení úsečky. Příklad 2 je ukázkou metody lineTo. Příklad 3 se věnuje vykreslení více než jedné úsečky s přerušením;
- Kapitola Obdélník je zaměřena na popis metod rect a stroke. Příklad 1 ukazuje vykreslení čtverce;
- V kapitole Kruh se také věnuji popisu metod arc a stroke. Příklad 1 ukazuje vykreslení kružnice;
- Kapitole Kvadratická křivka obsahuje popis metod moveTo a quadraticCurveTo. Příklad 1 ukazuje vykreslení kvadratické křivky;
- V kapitole Bézierova křivka se věnuji popisu metod moveTo a bezierCurveTo. Příklad 1 ukazuje vykreslení Bézierovy křivky.

7.8.3 VYBARVOVÁNÍ

Kniha Vybarvování se skládá ze čtyř stránek (Plný útvar jednou barvou, Přechod barev, Vložení obrázku a Stíny), které popisují, jak změnit výchozí barvu útvarů.

- V kapitole Plný útvar jednou barvou se věnuji popisu metody fillStyle. Příklad 1 ukazuje vykreslení čtverce žlutou barvou. Příklad 2 ukazuje kruh vyplněný barvou;
- V kapitole Přechod barev se zaměřuji na popis metod createLinearGradient a addColorStop. Příklad 1 ukazuje čtverec vyplněný přechodem barev. Příklad 2 ukazuje přechod barev v kruhu;
- V kapitole Vložení obrázku se zaobírám popisem metody drawImage a různými jejími variacemi. Příklad 1 ukazuje možnost nahrát na Canvas obrázek z odkazu. Druhý příklad rozšiřuje ten první o možnost změny rozměrů obrázku, který nahráváme na Canvas. Třetí příklad ukazuje konečné možnosti metody drawImage;
- V kapitole Stíny se věnuji popisu metod shadowOffsetX/Y, shadowBlur a shadowColor. Příklad 1 ukazuje možnost přidání stínu.

7.8.4 PRÁCE S TEXTEM

Kniha Práce s textem se skládá ze šesti kapitol (Řetězec, Barva textu, Font, velikost písma, řez písma, Ohraničení textu, Horizontální/Vertikální zarovnání textu).

- V kapitole Řetězec se věnuji popisu metod fillText. Příklad 1 ukazuje, jak vykreslit text na Canvas;
- V kapitole Barva textu popisuji metodu fillStyle. Příklad 1 ukazuje možnost vybarvit text barvou;
- V kapitole Font, velikost písma, řez písma přibližuji metodu font. Příklad 1 ukazuje možnost změny fontu, velikosti písma atd.;
- V kapitole Ohraničení textu se věnuji popisu metody strokeText. Příklad 1 ukazuje, jak ohraničit text;
- V kapitole Horizontální zarovnání textu se zaobírám popisem metody textAlign. Příklad 1 ukazuje možnosti horizontálního zarovnání textu. Pro lepší pochopení této metody jsem vytvořil čáru, podle níž zarovnávám text;
- V kapitole Vertikální zarovnání textu se pak věnuji popisu metody textBaseline. Příklad 1 ukazuje možnosti vertikálního zarovnání textu. Pro lepší pochopení této metody jsem vytvořil čáru a podle té zarovnávám text.

7.8.5 TRANSFORMACE

Kniha Transformace se skládá z pěti kapitol (Save a restore, Posun, Škálování, Rotace, Matice). V těchto kapitolách přibližuji transformaci Canvasu.

- V kapitole Save a restore se věnuji popisu metod save a restore. V příkladu 1 ukazuji, jak pomocí save a restore mohu změnit barvy útvaru. V druhém příkladu navíc ukazuji také škálování a následnou obnovu;
- V kapitole Posun popisuji metodu translate. Příklad 1 ukazuje posun Canvasu;
- V kapitole Škálování se věnuji popisu metody scale. Příklad 1 ukazuje škálování Canvasu;
- V kapitole Rotace se zaměřuji na metodu rotate. Příklad 1 ukazuje rotaci Canvasu;
- V kapitole Matice se věnuji popisu metody transform, setTransform a resetTransform. Příklad 1 ukazuje, jak transformovat útvar.

7.8.6 ANIMACE

Knihu Animace tvoří šest kapitol (Interval, Timeout, Request animation frame, Stopky, Sprite, Měsíc a slunce). V těchto kapitolách se věnuji popisu animací v Canvasu.

- V kapitole Interval se věnuji popisu metod setInterval a clearInterval. Příklad 1 ukazuje opakované vykreslování čtverce se změnou v souřadnicích;
- V kapitole Timeout se zaměřuji na popis metody setTimeout a clearTimeout. Příklad 1 ukazuje vykreslení čtverce po uplynulé době. Druhý příklad ukazuje postupné vykreslování loga metodou arc. Tento příklad jsem řešil objektově;
- V kapitole Request animation frame se věnuji popisu metod requestAnimationFrame a cancelAnimationFrame. Příklad 1 ukazuje opakované vykreslování čtverce se změnou v souřadnicích;

- Kapitola Stopky ukazuje možnost vytvoření stopek v Canvasu za pomocí arc, filltext a setInterval;
- V kapitole Sprite se věnuji animaci pomocí sprite sheetu. Příklad 1 ukazuje praktické využití pro animace pomocí metod drawImage a requestAnimationFrame;
- Kapitola Měsíc a slunce ukazuje možnost vytvoření pohybujícího se objektu podle osy pomocí metody arc a setInterval.

7.8.7 UDÁLOSTI

Knihu Události tvoří dvě kapitoly (Klávesnice, Myš). V těchto kapitolách se věnuji popisu událostí v Canvasu.

- V kapitole Klávesnice se věnuji popisu událostí onkeydown, onkeypress a onkeyup. Příklad 1 ukazuje možnost ovládání sprite pomocí klávesnice v intervalu. Příklad 2 ukazuje možnost ovládání sprite obrázku pomocí klávesnice;
- V kapitole Myš se věnuji popisu událostí onclick, oncontextmenu, onmousemove a mouseleave. Oba příklady ukazují možnost kreslení na Canvas pomocí myši.

7.8.8 WEBOVÉ TOOLBARY

Knihu Webové toolbary tvoří jedna kapitola (Posuvník). V této kapitole se věnuji popisu metody drawImage a posuvníku.

- Vytvořený program funguje jako slider pro obrázky. Využil jsem metody drawImage a časové metody setInterval. Důležité je mít všechny obrázky ve stejném rozlišení.

7.9 SVG

Toto téma slouží k pochopení základních kreslicích funkcí elementu SVG a jejich následné animace. Téma obsahuje pět knih a jednu stránku.

7.9.1 PLÁTNO

Kapitola Plátno se skládá ze stránky, která se věnuje popisu rozdílu viewboxu a viewportu. V příkladu 1 vykresluji na dva SVG elementy (přičemž jeden má nastavený viewbox) stejný čtverec.

7.9.2 ZÁKLADNÍ TVARY

Tato kapitola se skládá z osmi stránek (Úsečka, Obdélník, Kruh, Elipsa, Čára lomená, Mnohoúhelník, Cesta, Element g), které popisují vytváření základních grafických útvarů pro element SVG.

- V kapitole Úsečka se věnuji popisu elementu line. Příklad 1 ukazuje, jak vykreslit úsečku;
- V kapitole Obdélník přibližuji element rect. Příklad 1 ukazuje, jak vykreslit obdélník s oblou hranou;
- V kapitole Kruh se zaměřuji na popis elementu circle. Příklad 1 ukazuje, jak vykreslit kruh;
- V kapitole Elipsa se věnuji popisu elementu ellipse. Příklad 1 ukazuje, jak vykreslit elipsu;
- V kapitole Čára lomená popisuji element polyline. Příklad 1 ukazuje, jak vykreslit čáru lomenou;
- V kapitole Mnohoúhelník se věnuji popisu elementu polygon. Příklad 1 ukazuje, jak vykreslit mnohoúhelník;
- V kapitole Cesta se věnuji popisu elementu path. Příklad 1 ukazuje, jak vykreslit pomocí elementu path dva obdélníky. Příklad 2 ukazuje, jak vykreslit pomocí elementu path kruh;
- V kapitole Element g popisuji element g. Příklad 1 ukazuje, jak sjednotit dva obdélníky.

7.9.3 VYBARVOVÁNÍ

Tato část se skládá ze tří stránek (Výplň jednou barvou, Přechod barev, Vložení vzoru), které popisují grafickou úpravu elementů SVG.

- V kapitole Výplň jednou barvou se věnuji popisu style=fill,stroke. Příklad 1 ukazuje, jak vybarvit kruh modrou barvou a nastavit ohraničení černou barvou;
- V kapitole Přechod barev popisuji elementy defs, linearGradient a radialGradient. Příklad 1 ukazuje, jak vybarvit kruh přechodem barev od modré k bílé;
- Kapitola Vložení vzoru se věnuje popisu elementu pattern. Příklad 1 ukazuje, jak vyplnit kruh malým kruhem. Příklad 2 ukazuje změnu seskupeným útvarům.

7.9.4 PRÁCE S TEXTEM

Kapitola se skládá ze dvou stránek (Řetězec a Řetězec na křivce). Popisuje vkládání textu do elementu SVG.

- V kapitole Řetězec se věnuji popisu elementu text. Příklad 1 ukazuje, jak vykreslit text do elementu SVG;
- V kapitole Řetězec na křivce popisuji element textPath. Příklad 1 ukazuje, jak vykreslit text na křivku.

7.9.5 TRANSFORMACE

Kapitolu tvoří pět stránek (Posun, Škálování, Rotace, Zkosení, Matice). Popisuje souřadnicovou úpravu SVG elementu.

- V kapitole Posun se věnuji popisu elementu transform=translate. Příklad 1 ukazuje, jak posunout čtverec;
- V kapitole Škálování se zaměřuji na popis elementu transform=scale. Příklad 1 ukazuje, jak zvětšit čtverec;
- V kapitole Rotace se věnuji popisu elementu transform=rotate. PříKlad 1 ukazuje, jak otočit čtverec;
- V kapitole Zkosení popisuji element transform=skewX/Y. PříKlad 1 ukazuje, jak zkosit čtverec;
- V kapitole Matice se věnuji popisu elementu transform=matrix. Příklad 1 ukazuje, jak změnit čtverec pomocí matice.

7.9.6 ANIMACE

Kniha Animace se skládá ze šesti kapitol (Animace elementu, Animace podél křivky, AnimateTransform, Gradient, Ikony, Logy). V těchto kapitolách se věnuji popisu animací elementu SVG.

- V kapitole Animace elementu se věnuji popisu elementu animate. Příklad 1 ukazuje, jak animovat čtverec pomocí elementu animate;
- V kapitole Animace podél křivky popisuji element animateMotion. Příklad 1 ukazuje, jak pomocí elementu animateMotion můžeme nastavit objektu trasu podle souřadnic path;
- V kapitole AnimateTransform se zaměřuji na popis elementu animateTransform. Příklad 1 ukazuje, jak se trojúhelník točí kolem určitého bodu;
- Kapitola Gradient ukazuje možnosti elementu linearGradient a animate;
- Kapitola Ikony ukazuje možnosti tvorby animovaných ikon za pomoci animateTransform;
- Kapitola Logo ukazuje možnosti animace ikon za pomoci animateTransform.

8 ZÁVĚR

Předkládaná bakalářská práce je zaměřena na představení HTML elementů Canvas a SVG a jejich následné animace. V kapitole Animace se blíže věnuji popisu animací pomocí SMIL a ECMAScriptu. V další kapitole pak analýze knihoven pro použití SVG a Canvas elementů. V rámci praktické části jsem vytvořil e-kurz a sadu ukázek, které přibližují jednotlivé metody a elementy.

Vhodnost použití technologií Canvas a SVG závisí na problému, který je potřeba vyřešit. Obě technologie jsou velice rozdílné, přičemž každá má co nabídnout. Za rozšířením SVG stojí především vzestup responsivních webů. SVG doporučuji použít k práci s logy, ikonami, prvky uživatelského rozhraní, interaktivními plánky budov, vizualizací dat (grafů). Výhodou je také možnost animovat pomocí ECMAScriptu, standartu SMIL nebo CSS. Usnadňujícím prvkem při práci zde je strukturovatelnost. Grafik může snadno identifikovat jednotlivé objekty, které následně může seskupovat/animovat podle potřeb.

Jak jsem již zmínil výše, do Canvas plátna je možné nahrát různé formáty bitmapových obrázků, a následně je pak upravovat pomocí filtrů. Je tak šetřen výkon serveru, poněvadž dochází k menší komunikaci mezi serverem a zařízením. To je výhodné například v případě mobilních zařazení, kdy Jsou šetřena data a dochází k menší spotřebě baterie. Jelikož je plátno Canvasu bitmapou, jeho výpočetní náročnost nenarůstá se složitostí nakreslených objektů, což je výhodné při vykreslování map či her.

RESUMÉ

Cílem této bakalářské práce je uvedení do problematiky a základů dvourozměrné grafiky, jak vektorové, tak bitmapové, a její následné animace ve webovém prostředí. Na začátku se práce věnuje definování základních tvarů elementů SVG a Canvas. Další část je věnována popisu animačních možností obou elementů. Třetí kapitola je zaměřena na analýzu volně dostupných ECMAScriptových knihoven, které je možné použít pro práci s oběma elementy. Hlavním kritériem byla určena vizualizace, a to především vzhledem k tématu bakalářské práce. V praktické části práce bylo důležité vytvoření sady příkladů (https://codepen.io/fpe-zcu-tutorial) a e-kurzu. Příklady jsou editovatelné, přičemž je zároveň vidět jejich výsledná podoba. K tomuto bylo užito vývojářské sociální prostředí CodePen. Online kurz je provozován v univerzitním prostředí Moodle.

S tímto řešením se mohou studenti připojit do kurzu pomocí STAGu (https://moodle.athos.zcu.cz/course/view.php?id=23).

The aim of my bachelor thesis is stating the basis of 2-D graphics, both vector and bitmap, and its subsequent animation on web. In the beginning the work deals with the definition of the basic forms of the SVG and Canvas elements. The next part is devoted to description of possibilities of animation of both elements. The third chapter is focused on the analysis of the accessible ECMA Script libraries, which can be used for work with both elements. The main criterion was the visualization, primarily due to the topic of the bachelor thesis. The important part of the practical section of the work was to create a set of examples (https://codepen.io/fpe-zcu-tutorial) and the e-course. The examples can be edited, while the final look can be viewed at the same time. This was done by the social development environment CodePen. The online course runs on the university environment Moodle. With this solution students can join the course through STAG (https://moodle.athos.zcu.cz/course/view.php?id=23).

SEZNAM LITERATURY

[1] CECCO, Raffaele. Supercharged JavaScript graphics. Sebastopol, CA: O'Reilly Media, c2011. ISBN 978-1449393632.

[2] BELLAMY-ROYDS, Amelia, Kurt CAGLE a Dudley STOREY. Using SVG with CSS3 and HTML5: vector graphics for web design. Sebastopol, CA: O'Reilly, [2018]. ISBN 978-1491921975.

[3] DRASNER, Sarah. SVG animations: from common UX implementations to complex responsive animation. Boston: O'Reilly, 2017. ISBN 978-1491939703.

[4] DAILEY, David., Jon. FROST a Domenico. STRAZZULLO. Building web applications with SVG. Sebastopol, Calif.: Microsoft Press, c2012. ISBN 978-0735660120.

[5] FULTON, Steve. a Jeff. FULTON. HTML5 canvas. Second edition. Farnham: O'Reilly, [2013]. ISBN 978-1449334987

[6] CAMPESATO, Oswald. HTML5 Canvas and CSS3 graphics primer. Dulles, Virginia: Mercury Learning and Information, [2013]. ISBN 978-1936420346.

[7] GEARY, David M. Core HTML5 canvas: graphics, animation, and game development. Upper Saddle River, NJ: Prentice Hall, c2012. ISBN 978-0132761611.

[8] Touch events - Web APIs | MDN. [online]. Copyright © 2005 [cit. 22.06.2019]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Touch_events

[9] HTML5 Introduction. *W3Schools Online Web Tutorials* [online]. Dostupné z: https://www.w3schools.com/html/html5_intro.asp

[10] ECMAScript Archives - Zdroják. *Zdroják - o tvorbě webových stránek a aplikací* [online]. Dostupné z: https://www.zdrojak.cz/n/ecmascript/

[11] Co je JavaScript? - Learning the Web | MDN. [online]. Copyright © 2005 [cit. 29.06.2019]. Dostupné z: https://developer.mozilla.org/cs/docs/Learn/JavaScript/First_steps/Co_je_JavaScript

[12] CSS3 - CSS: Cascading Style Sheets | MDN. [online]. Copyright © 2005 [cit. 29.06.2019]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3

[13] Synchronized Multimedia Integration Language (SMIL 3.0). *World Wide Web Consortium (W3C)* [online]. Copyright © 2008 [cit. 29.06.2019]. Dostupné z: https://www.w3.org/TR/SMIL3/

[14] Adding vector graphics to the Web - Learn web development | MDN. [online]. Copyright © 2005 [cit. 29.06.2019]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Adding_vector_graphics_to_the_Web

[15] GitHub Help. [online]. Copyright © 2019 GitHub, Inc. [cit. 29.06.2019]. Dostupné z: https://help.github.com/en

[16] D3.js - Data-Driven Documents. *D3.js - Data-Driven Documents* [online]. Dostupné z: https://d3js.org/

[17] BonsaiJS - A Graphics Library. *BonsaiJS - A Graphics Library* [online]. Copyright © 2012 [cit. 29.06.2019]. Dostupné z: https://bonsaijs.org/

[18] Paper.js. *Paper.js* [online]. Dostupné z: http://paperjs.org/

[19] Interactive JavaScript charts for your webpage | Highcharts. *Interactive JavaScript charts for your webpage | Highcharts* [online]. Copyright © 2019 Highcharts. All rights reserved. [cit. 29.06.2019]. Dostupné z: https://www.highcharts.com/

[20] Chart.js | Open source HTML5 Charts for your website. *Chart.js | Open source HTML5 Charts for your website* [online]. Dostupné z: https://www.chartjs.org/

[21] *Nicolas Garcia Belmonte* [online]. Copyright © 2013 [cit. 29.06.2019]. Dost upné z: https://philogb.github.io/jit/

[22] PixiJS. *PixiJS* [online]. Copyright © 2019 Goodboy Digital Ltd. All Rights Reserved [cit. 29.06.2019]. Dostupné z: https://www.pixijs.com/

[23] plotly.js | JavaScript Graphing Library. *Modern Analytic Apps for the Enterprise - Plotly* [online]. Copyright © 2019 Plotly. All rights reserved. [cit. 29.06.2019]. Dostupné z: https://plot.ly/javascript/

SEZNAM OBRÁZKŮ