

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

**Mapování areálu ČZU s využitím vektorové grafiky
a jazyka SVG**

Bc. Petr Zdvořáček

© 2023 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Petr Zdvořáček

Informatika

Název práce

Mapování areálu ČZU s využitím vektorové grafiky a jazyka SVG

Název anglicky

Mapping the ČZU campus using vector graphics and the SVG language

Cíle práce

Cílem teoretické části práce je pomocí studia odborné literatury podrobně definovat vlastnosti 2D vektorové grafiky a značkovacího jazyka SVG. Popsat principy tvorby map pomocí jazyka SVG a vystihnout jeho úlohu ve webové kartografii. Pomocí dostupných zdrojů detailněji analyzovat webovou kartografii a nakonec popsat publikování (přenos) vektorových dat na web.

Cílem praktické části práce je zmapovat areál a jednotlivé fakulty České zemědělské univerzity v Praze pomocí vektorové grafiky a jejího formátu SVG. Následně navrhnout webovou prezentaci, sloužící k přehlednému procházení map jednotlivých fakult a areálu ČZU.

Metodika

Teoretická část práce bude vytvořena na základě čerpání z odborných knižních a webových zdrojů. Rešerše bude zaměřena na vektorovou grafiku, jazyk SVG a tvorbu map v tomto formátu. Bude vysvětlena definice webové kartografie a publikování vektorových dat na web.

Praktická část práce bude zaměřena na tvorbu vektorových map České zemědělské univerzity v Praze ve formátu SVG. Nástrojem pro zpracování map bude vhodný grafický editor pro tvorbu vektorové grafiky. Vytvořené mapy budou publikovány ve webové prezentaci, která bude sloužit pro zobrazení map a nalezení místností v areálu.

Vektorové mapy budou také volně přístupné k dalšímu možnému využití a úpravám, zejména kvůli formátu SVG.

Doporučený rozsah práce

60-80 stran

Klíčová slova

vektorová grafika, SVG, webová kartografie, vektorové mapy, areál ČZU, webová prezentace

Doporučené zdroje informací

NÉTEK, Rostislav, 2020. Webová kartografie – specifika tvorby interaktivních map na webu. Olomouc: Univerzita Palackého v Olomouci. ISBN 978-80-244-5827-4.

PETERSON, Michael P., 2003. Maps and the Internet. Omaha, USA: Pergamon. ISBN 9780080535791.

TRIPPE, Bill a Kate BINDER, 2002. SVG For Designers: Using Scalable Vector Graphics in Next-Generation Web Sites. New York, United States: McGraw-Hill/OsborneMedia. ISBN 978-0072225297.

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Dana Vyníkarová, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 4. 11. 2022

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 30. 11. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci „Mapování areálu ČZU s využitím vektorové grafiky a jazyka SVG“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30. listopadu 2023

Poděkování

Rád bych touto cestou poděkoval vedoucí mé práce Ing. Daně Vynikarové, Ph.D. za velkorysou pomoc, okamžité kontroly a neocenitelné rady při zpracování této práce.

Mapování areálu ČZU s využitím vektorové grafiky a jazyka SVG

Abstrakt

Diplomová práce zkoumá problematiku vektorové grafiky a jazyka SVG (Scalable Vector Graphics) se zaměřením na tvorbu vektorových map. Cílem práce je vytvoření vektorových map ve formátu SVG, které mapují podlaží fakultních budov v areálu České zemědělské univerzity v Praze (ČZU) za účelem sjednocení zmapování všech podlaží a učeben těchto budov do jednotného stylu a formátu souboru, který lze jednoduše přizpůsobovat, a hlavně publikovat na webu nebo v jiných online nástrojích pro univerzitní potřeby. Literární rešerše, která je podkladem pro vytvoření vektorových map se primárně zaměřuje na princip vektorové grafiky, charakteristiku formátu SVG, základy webové kartografie a publikování vektorových dat na webu.

Praktická část navazuje tvorbou vlastních vektorových map na základě teoretických východisek a vlastních zkušeností v oblasti vektorové grafiky. Nástrojem pro tvorbu je grafický editor Adobe Illustrator. V rámci práce dochází souběžně k deskripci postupu při tvorbě vektorových map. V poslední fázi je vytvořena webová stránka, která slouží k přehlednému zobrazení a procházení vektorových map. Výstupem diplomové práce je soubor vektorových map podlaží fakultních budov a webová stránka pro jejich prezentaci, která je dostupná na veřejné doméně www.dp.zdvoracek.eu.

Klíčová slova: vektorové mapy, SVG, webová stránka, areál ČZU v Praze, fakultní budovy, Adobe Illustrator

Mapping the ČZU campus using vector graphics and the SVG language

Abstract

The thesis deals with the issue of vector graphics and the SVG (Scalable Vector Graphics) language with a focus on the creation of vector maps. The aim of the thesis is to create vector maps in SVG format representing the floors of the faculties on the campus of the Czech University of Life Sciences in Prague (CZU), in order to unify the mapping of all the floors of the classrooms of the faculty buildings on the campus into a single style and file format that can be easily customised and, most importantly, published on the web or other online tools for university use. The literary research that is the basis for the creation of vector maps focuses mainly on the principles of vector graphics, the characteristics of the SVG format, the basics of web cartography and publishing vector data on the web.

The practical part is followed by the creation of vector maps based on the theoretical background and the author's experience in the field of vector graphics. The tool used for creation is the graphic editor Adobe Illustrator. The work includes a parallel description of the procedure for creating vector maps. In the last phase, a website is created, which is used to display and browse the vector maps. The result of the work is a set of vector maps of the floors of the faculty buildings and a website for their presentation, which is available in the public domain at www.dp.zdvoracek.eu.

Keywords: vector maps, SVG, website, CZU campus in Prague, faculty buildings, Adobe Illustrator

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	13
2.1 Cíl práce	13
2.2 Metodika	13
3 Teoretická východiska	14
3.1 Vektorová grafika.....	14
3.1.1 Princip vektorové grafiky	15
3.1.2 Výhody a využití.....	15
3.1.3 Bézierova křivka	16
3.1.4 Vektorizace	16
3.1.5 Formáty vektorové grafiky	17
3.1.5.1 AI (Adobe Illustrator).....	17
3.1.5.2 CDR (CorelDRAW)	17
3.1.5.3 EPS (Encapsulated PostScript).....	18
3.1.5.4 DWG (Drawing).....	18
3.1.5.5 PDF (Portable Document Format).....	19
3.1.5.6 SVG (Scalable Vector Graphics).....	19
3.2 SVG.....	19
3.2.1 Specifikace a vlastnosti formátu.....	19
3.2.2 Import SVG na webovou stránku	20
3.2.2.1 SVG jako vložený obrázek.....	20
3.2.2.2 SVG v elementu <i>object</i>	21
3.2.2.3 SVG přímo v dokumentu HTML	22
3.2.3 Struktura dokumentu.....	22
3.2.4 Geometrické elementy (tvary)	23
3.2.5 Souřadnice, transformace a zobrazení	27
3.2.6 Stylování a grafické efekty	28
3.2.7 Ořez a masky	28
3.2.8 Text	30
3.2.9 Animace	31
3.2.10 Přístupnost	32
3.2.11 Historie verzí formátu.....	33
3.2.12 Podpora ve webových prohlížečích	34
3.3 Softwary pro tvorbu SVG map	35
3.3.1 Adobe Illustrator	35

3.3.2	Inkscape	36
3.3.3	QGIS	37
3.3.4	Mapbox studio.....	37
3.3.5	Simplemaps a Scribble Maps	38
3.3.6	Zhodnocení a výběr vhodného softwaru	38
3.4	Webová kartografie a využití SVG	39
3.4.1	Interaktivita a animace v SVG mapách.....	40
3.4.2	Optimalizace SVG map na web	40
3.4.3	Budoucnost SVG map.....	41
3.5	Publikování vektorových dat na web	42
3.5.1	Přímá publikace do HTML	43
3.5.2	Webové mapové služby	43
3.5.2.1	WMS	43
3.5.2.2	WFS	43
3.5.3	Webmapping	44
3.5.4	Vektorové dlaždice	44
3.5.5	GeoJSON	44
4	Vlastní práce	46
4.1	Sběr podkladů.....	46
4.2	Grafická tvorba map.....	51
4.3	Export map do SVG	58
4.4	Optimalizace SVG.....	59
4.5	Návrh webu	60
4.5.1	Wireframe	61
4.5.2	Tvorba HTML.....	61
4.5.3	Finální webová stránka	62
5	Výsledky	64
6	Závěr.....	65
7	Seznam použitých zdrojů.....	67
8	Seznam obrázků a zkratk	69
8.1	Seznam obrázků	69
8.2	Seznam použitých zkratk.....	71
Přílohy		72

1 Úvod

Diplomová práce se zaměřuje na tvorbu vektorových map ve formátu SVG, které mapují podlaží fakult v areálu ČZU v Praze. V současné době je, při velkém rozvoji digitalizace kladen stále větší důraz na vizualizaci dat. To se dnes promítá ve všech oborech od samotného designu přes vývoj až po analýzu. Jasně, srozumitelné, jednoduché, ale zato účelné zobrazení dat je v dnešní době také velmi důležité i v akademickém prostředí. Vzdělávací instituce se zaměřují na poskytování komplexních a dostupných informací o svých kampusech všem studentům, pedagogům i novým návštěvníkům. Jako vhodný nástroj pro takovou tvorbu se v poslední době objevuje vektorová grafika s formátem a zároveň značkovacím jazykem SVG (Scalable Vector Graphics).

Všechny fakulty ČZU v Praze doposud nenabízely univerzální a jednotné mapy podlaží svých budov a učeben. Proto bylo hlavní motivací této práce sjednotit a zmapovat všechna podlaží učeben fakultních budov v areálu do jednotného stylu a formátu souboru, který lze jednoduše publikovat a přizpůsobovat.

Vektorový obrazový formát SVG nabízí výhodu škálovatelnosti beze ztráty kvality obrazu, což umožňuje tvořit mapy, které lze jednoduše přizpůsobovat podle změn v infrastruktuře kampusu. Využitím vlastností SVG grafiky se práce snaží přispět k rozvoji map v areálu univerzity publikovatelných zejména na webu a zlepšit celkový zážitek virtuálních i fyzických návštěvníků univerzitního areálu.

Webová kartografie se v současné době stále více zaměřuje k publikaci dat na webu, což bylo také důvodem vytvořit mapy, které lze jednoduše integrovat na web a do online platforem univerzity.

Cílem literární rešerše je nejprve analyzovat problematiku vektorové grafiky, její princip zobrazování dat, využití a formáty ve kterých je nejčastěji vytvářena či publikována. Další, nejrozšířenější kapitolou, je podrobnější analýza formátu SVG, konkrétně jeho specifikace a vlastnosti. Rešerše důkladně popisuje strukturu samotného SVG dokumentu, jeho jednotlivé prvky, geometrické elementy, stylování, grafické efekty, a nakonec se zaměřuje na přístupnost, import a podporu na webu a webových prohlížečích. Na kapitolu SVG navazuje analýza softwarů pro tvorbu vektorových map v SVG formátu, jak profesionálních, tak freeware i online nástrojů. Následující kapitola popisuje propojení a využití SVG map ve webové kartografii. Nejprve probíhá analýza webové kartografie a následně je zkoumáno využití SVG map, jejich optimalizace a budoucnost ve webové

kartografii. Poslední kapitolou je publikování vektorových dat na webu. Analýza popisuje možnosti publikace zejména formátu SVG na webu, které jsou důležité pro poslední fázi návrhu ve vlastní práci.

Cílem praktické části práce je samotná tvorba vektorových map ve formátu SVG, počínaje deskripcí sběru podkladů a metodologického popisu, tj. jak se postupovalo při samotné tvorbě v grafickém editoru Adobe Illustrator. Následuje proces exportu map do formátu SVG a jejich optimalizace pro efektivní publikaci na web. Nakonec je navržena webová prezentace těchto map sloužící k přehlednému procházení učeben v jednotlivých podlažích fakult v areálu univerzity. Vytvořené mapy mohou sloužit akademickému prostředí jako základ pro jakoukoliv další tvorbu, vývoj schémat nebo aplikací pro projekty ČZU v Praze.

Komplexně se práce zaměřuje na využití vektorové grafiky ve formátu SVG ke zmapování areálu ČZU v Praze. Prostřednictvím řešerše problematiky formátu SVG a sběru dat v podobě jednotlivých map fakultních budov v areálu si práce klade za cíl přispět ke zlepšení mapového řešení v areálu ČZU v Praze.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem práce je vytvoření vektorových map ve formátu SVG, které budou mapovat podlaží budov jednotlivých fakult v areálu ČZU v Praze.

Prvním dílčím cílem je na základě vlastností vektorové grafiky a formátu SVG demonstrovat potenciál SVG jako účinného nástroje pro tvorbu vektorových map snadno publikovatelných na webu. Druhým dílčím cílem je vytvoření návrhu webové stránky, dostupné na veřejné doméně pro přehledné zobrazení a procházení těchto map. Při naplnění těchto cílů bude výsledkem jednotné základní zmapování učeben fakultních budov v areálu ČZU v Praze.

2.2 Metodika

Teoretická část práce vychází z odborných webových a knižních zdrojů. Rešerše je zaměřena na vektorovou grafiku, její princip zobrazování dat, základní prvky, využití a formáty ve kterých může být publikována. Dále obsahuje podrobnou specifikaci značkovacího jazyka a grafického formátu SVG. Zaměřuje se na vlastnosti v celé struktuře formátu, jako jsou geometrické elementy, souřadnice, stylování a grafické efekty a také popisuje přístupnost a podporu na webu. Dále uvádí komparaci softwarů pro tvorbu SVG map a charakterizuje webovou kartografii a využití SVG. Nakonec analyzuje možnosti publikování těchto vektorových dat na webu.

Praktická část obsahuje postup samotné tvorby vektorových map jednotlivých podlaží fakultních budov v areálu ČZU v Praze ve formátu SVG. Mapy jsou tvořeny pomocí 2D vektorové grafiky ve vektorovém grafickém editoru Adobe Illustrator. Na základě sběru podkladů k výstavbě budov obsahují jednotlivé mapy přesné půdorysy podlaží a graficky znázorňují všechny oficiální učebny v dané budově fakulty. Mapy jsou v Adobe Illustratoru exportovány do formátu SVG a následně je SVG kód optimalizován zejména pro snížení datové velikosti souboru a publikaci na webu. Pro webovou prezentaci těchto map je vytvořena jednoduchá webová stránka, kde jsou mapy zobrazeny. Stránka je nahrána na webový server a pomocí vlastní veřejné domény je snadno přístupná z jakéhokoliv webového prohlížeče.

3 Teoretická východiska

3.1 Vektorová grafika

Vektorová grafika je oblast spadající pod obor Počítačová grafika. Počítačová grafika je obor výpočetní techniky, který se zabývá tvorbou a úpravou grafických informací. Využívá počítač k tvorbě grafických objektů a pro úpravu a následné zobrazení nasnímaných informací z reálného světa. (Žára, 2004)

U počítačové grafiky se lze setkat s pojmy 2D a 3D grafika. Tyto pojmy odlišují grafiku pracující s dvou nebo trojrozměrnými objekty.

Základní odlišnosti mezi 2D a 3D grafikou tedy spočívají v rozměru a prostorové hloubce, kterou představují ve vizuálním obsahu. 2D grafika se využívá pro situace, kdy má obraz pouze dva rozměry, šířku a výšku, zatímco 3D grafika využívá trojrozměrný obraz, tedy tři rozměry, šířku, výšku a hloubku. Často se lze také setkat s termínem „pseudo 3D“, který se využívá pro 2D grafiku nebo animaci, která se snaží simulovat vzhled 3D grafiky, vytváří se náznak třetího rozměru, který ale ve skutečnosti neexistuje. (2D a 3D grafika)

Samotná 2D počítačová grafika se dělí podle skladby a formy zobrazení na dvě základní části, grafiku rastrovou a vektorovou. Tyto dvě oblasti v praxi tvorby grafiky fungují společně a často se doplňují, i když vycházejí z odlišných základů. Rastrová grafika je složená z jednotlivých pixelů neboli obrazových bodů. Na počtu pixelů v konkrétním obraze je vždy závislá, to znamená, že obraz je možné bez ztráty kvality zvětšit pouze do jejího maximálního rozlišení. Rastrová grafika je nejvhodnější pro fotografie, avšak při změně její velikosti je náchylná k tzv. pixelizaci, což je kostičkovaný a zkraslený vzhled obrázku způsobený viditelností jednotlivých pixelů při zvětšení nad původní rozlišení. Vektorová grafika je vytvořena pomocí matematických rovnic a je nezávislá na rozlišení. Je vhodná zejména pro tvorbu log a jednoduchých ilustrací, je snadno škálovatelná a neztrácí kvalitu.

V souvislosti s vektorovou grafikou je nutno zmínit také 3D grafiku. Vektorová grafika a 3D grafika jsou sice dvě odlišné metody reprezentace vizuálních informací, ale v kontextu vytváření 3D grafiky mohou být příbuzné. 3D grafika je vektorové grafice příbuzná, protože také pracuje se souřadnicemi, křivkami a plochami. Vektorové prvky lze tedy integrovat do 3D prostředí, proto aby se zvýšila efektivita, škálovatelnost a některé aspekty celkové vizuální kvality 3D grafiky. Do 3D grafiky lze například integrovat různé

cesty a křivky, drátové kostry, vektorové textury nebo ostatní 2D vektorové prvky. (Žára, 2004)

Vektorová grafika pro zobrazení na digitální obrazovce nebo na fyzickém tisku je grafika v 2D prostoru, proto se tato práce zaměřuje primárně na informace o 2D grafice. Práce se konkrétně zaměřuje na tvorbu vektorové grafiky ve formátu SVG, tudíž jsou v následující kapitole uvedeny její základní prvky, výhody a nepoužívanější vektorové formáty.

3.1.1 Princip vektorové grafiky

Vektorová grafika se skládá z jednoduchých geometrických prvků, které ji reprezentují. Mezi tyto geometrické prvky patří body, křivky, přímky, polygony, mnohoúhelníky a další. Základní princip vektorové grafiky spočívá v matematice, která je klíčová pro její existenci. Oproti rastru, kde se obrázek skládá z jednotlivých pixelů, je vektorová grafika vytvořena pomocí křivek, které se nazývají vektory. Tyto křivky jsou tvořeny spojením kotevních bodů a mají také definovanou výplň. Díky tomuto způsobu zobrazení je vektorová grafika schopna definovat přesný popis tvarů a objektů s použitím matematických vzorců, což jí umožňuje držet škálovatelnost a zachovávat kvalitu i při změnách velikosti. (Horný, 2006)

3.1.2 Výhody a využití

Vektorová grafika má několik významných výhod právě oproti grafice rastrové. Nejvýznamnější výhodou spočívá v tom, že při zvětšování obrazu nedochází ke ztrátě jeho kvality. Vektorový obraz si tak udržuje svou kvalitu a informace i při libovolném zvětšení. Rozlišení tohoto obrazu je prakticky neomezené, uživatel ho může určovat dle svých potřeb. Velikost souboru při ukládání je oproti rastrové grafice daleko menší. Tato vlastnost souvisí s matematickým zápisem vektorového obrazu, který zajišťuje bezztrátovost vektorové grafiky, což znamená, že není třeba komprese jako u rastru.

Další výhodou vektorů je možnost rozdělení obrázku na jednotlivé objekty, které lze dále používat samostatně. Vektory také nabízí neomezenou možnost úprav křivek v obrázku, aniž by byla narušena kvalita, což umožňuje nekonečné modifikace bez omezení zachování kvality.

Obvykle se vektorová grafika využívá zejména u obrázků s vysokou přesností, jako jsou loga, vizitky, tiskoviny, grafy a diagramy. Z vektorů jsou také tvořeny všechny fonty.

Také se využívá při počítačovém modelování nebo konstrukci. Lze se s ní setkat při tvorbě ilustrací, schémat a počítačových animací. (Pászto, 2017)

3.1.3 Bézierova křivka

Bézierova křivka představuje základní prvek v oblasti vektorové grafiky. Jedná se o klíčový prvek pro většinu vektorových tvarů. Vytvořil ji francouzský inženýr Pierre Bézier, což je patrné již z názvu. Bézierova křivka sehrála klíčovou roli v rozvoji CAD (Computer Aided Design) softwarů a přispěla k populárnímu využití křivek v digitálním prostředí.

Bézierova křivka má úplný původ v matematice a její význam v oblasti počítačové grafiky se uplatnil až později. Vektorový obrázek se skládá z kotevních bodů a cest, každá tato cesta je nazývána Bézierova křivka. Jedná se tedy o zakřivený segment cesty mezi dvěma kotevními body. V jednom obrázku může být použito od jedné do tisíců Bézierových křivek. Designér a ilustrátor Von Glitschka popsal Bézierovu křivku ve své knize: „*Bézierova křivka je cesta, kterou můžete z jednoho či druhého konce ohnout pomocí táhel vyčnívajících z kotevních bodů na koncích této cesty.*“ (Glitschka, 2013)

3.1.4 Vektorizace

Často se v praxi lze setkat s nutností převádět rastrové obrázky do vektorové podoby, právě kvůli jednoduššímu přenosu, úpravě velikosti a také pro publikaci na web. K tomu většinou velmi dobře slouží proces vektorizace (angl. vectorization) neboli trasování (angl. image tracing), které představuje přeměnu rastrového obrázku na formát vektorové grafiky. Tato technika se používá v situacích, kdy je třeba z bitmapového základu vytvořit vektorovou grafiku, například u písem, naskenovaných kreseb, malých logotypů apod. Vektorizace se také využívá tam, kde se obrázek realizuje rytím nebo řezáním, například u gravírování či řezacích plotrů. Kvalita výsledné vektorizace je ovlivněna komplexností původního rastrového podkladu. S nejnáročnějšími procesy se lze setkat nejčastěji při transformaci fotografií z rastrového obrazu do vektorového formátu. (Glitschka, 2013)



Obrázek 1 – Vektorizace z formátu PNG do SVG (How to convert PNG to SVG, 2023)

3.1.5 Formáty vektorové grafiky

Ve vektorové grafice existuje několik speciálně navržených formátů souborů, které jsou určeny speciálně pro její použití. Postupně se vyvinulo mnoho variant těchto vektorových formátů, ale pouze některé z nich se staly standardy v této oblasti.

Vektorové formáty jsou velmi specifické. Nelze vždy popsat specifičnost nebo výhody každého formátu, to je způsobeno tím, že většina formátů je navržena specificky pro určitý grafický editor a je těsně provázána s daným softwarem. Každá společnost, která vyvíjí grafický software si většinou vytváří vlastní formát souboru, který je optimalizován zejména pro jejich program. Tímto způsobem se formát stává pevně závislým na konkrétním softwaru. (Navrátil, 2007)

Níže jsou pro přehlednost uvedeny nejpoužívanější vektorové formáty a jejich charakteristiky. Analýza slouží k jejich srovnání s vektorovým formátem SVG, kterému se práce bude věnovat ještě podrobněji.

3.1.5.1 AI (Adobe Illustrator)

Nativní formát AI představuje formát vytvořený profesionálním grafickým editorem Adobe Illustrator od společnosti Adobe Systems. V současné době je tento formát široce používán v oblasti profesionální grafiky a patří mezi nejběžnější formát souboru pro vektorovou grafiku mezi grafiky všeobecně. AI formát je známý pro svou vysokou preciznost a sofistikovanost, avšak jeho práce je často omezena na profesionální nástroje, není kompatibilní zcela se všemi amatérskými programy.

Formát AI podporuje barevný model CMYK (Cyan, Magenta, Yellow, black), což je důležité zejména pro tiskovou přípravu a profesionální tisk velkého rozlišení. AI ukládá různé vrstvy, tvary, texty, lze využít ztrátové i bezztrátové komprese, různé hloubky barev, průhlednost a také více barevných modelů. AI se stále vyvíjí a s každou verzí nabízí pokročilejší funkce. Může se tak stát, že starší verze grafického editoru nemusí novější verze formátu podporovat. (Pászto, 2017)

3.1.5.2 CDR (CorelDRAW)

Formát CDR je primárně používán softwarem CorelDRAW od společnosti Corel Corporation, jedná se o hlavní výstupní formát softwaru. Původně byl formát zcela proprietární. Od verze softwaru X4 byla k formátu přidána kompatibilita se softwaru Adobe

Illustrator a Inkscape. Vektory jsou v tomto formátu mapovány na stránku bod po bodu, což zajišťuje zachování původního tvaru obrázků bez deformací.

S každou novou verzí programu CorelDRAW se také formát postupně vyvíjí. Pro udržení zpětné kompatibility je možné soubory vytvořené v novějších verzích ukládat ve formátech specifických pro starší verze. Formát také umožňuje pracovat s rastrovými obrázky. (Boldt, 2023)

3.1.5.3 EPS (Encapsulated PostScript)

Vektorový formát EPS (Encapsulated PostScript) je jedním z historických formátů vektorové grafiky, původně byl vyvinutý společností Adobe Systems v roce 1985. Tento formát vychází z jazyka PostScript, a proto byl původně znám pod příponou souboru „.ps“, později se označení formátu vyvinulo do koncovky „.eps“.

Hlavní předností formátu EPS je jeho nezávislost na konkrétním zařízení. Zejména v minulosti byl často považován za standard u dražších tiskáren díky svým rozsáhlým schopnostem pro ukládání obrazových dat. Později se stal běžnou volbou pro ukládání různých typů obrázků. EPS umožňuje ukládání jak kombinovaných obrazů s vektorovou a rastrovou grafikou, tak i čistě rastrových obrázků. Byl využíván jako univerzální formát pro výměnu vektorových dat a podporuje barevný model CMYK, což je důležité pro tiskovou přípravu.

Naopak nevýhodou formátu EPS je, že není příliš přístupný pro práci v amatérských grafických programech. Byl spíše navržen pro profesionální grafickou tvorbu. V současné době se formát EPS postupně stává méně používaným a je nahrazován jedním z nejznámějších formátů, PDF. (Pászto, 2017)

3.1.5.4 DWG (Drawing)

V přehledu vektorových formátů by rozhodně neměl chybět formát DWG. Soubory formátu DWG jsou formátem pro ukládání výkresů a modelů využívaných v prostředí CAD. Jedná se o nativní formát softwaru AutoCAD, který je vyvíjen společností Autodesk. DWG umožňuje ale také zobrazení v ostatních vektorových programech, jako jsou CorelDRAW nebo Adobe Illustrator. Formát je široce používán v odvětvích jako je architektura, strojírenství, elektrotechnika apod, a umožňuje ukládání 2D a 3D grafiky, geometrických a popisných dat, rozměrů, vrstev a dalších informací souvisejících s návrhem. (Adobe, 2023)

3.1.5.5 PDF (Portable Document Format)

Formát PDF, který vyvinula společnost Adobe Systems, využívá základní principy značkovacího jazyka PostScript. Tento formát není nijak specializován a dokáže pracovat s rastrovou i vektorovou grafikou. V jednom souboru lze uložit text, rastrový obrázek, vektorový objekt. PDF formát je kompatibilní téměř na jakékoli platformě a umožňuje ukládání jednotlivých vrstev, kanálů, přímých barev, cest a dalších prvků.

K vytváření a úpravám PDF dokumentů slouží software Adobe Acrobat. Mnoho programů také umožňuje export do tohoto formátu, například Microsoft Word nebo Office. Pokud otevřeme soubor v rastrovém editoru, jako je Photoshop, nebo ve vektorovém editoru Illustrator, je v případě uložení s vrstvami, soubor graficky editovatelný. K prohlížení PDF dokumentů existuje mnoho různých volně dostupných prohlížečů pro různé platformy. Nejznámějším z nich je Adobe Reader. (Navrátil, 2007)

3.1.5.6 SVG (Scalable Vector Graphics)

Formát neboli jazyk SVG je nejmodernějším formátem vektorové grafiky. Vzhledem k tomu, že SVG je hlavní téma této práce, následující kapitola obsahuje detailnější analýzu tohoto formátu.

3.2 SVG

Scalable Vector Graphics je jazyk, který se používá pro reprezentaci dvourozměrných vektorových obrázků, podporuje základní kreslení a také animace a interakci. Jazyk SVG byl vytvořen ze značkovacího jazyka XML (Extensible Markup language), to znamená, že vektorový obraz vytvořený v SVG může mít také podobu textového řetězce. SVG je vyvíjeno od roku 1999 společností W3C (World Wide Web Consortium). Jedná se o open-source standard, který W3C doporučuje k užívání vektorové grafiky na webu, právě pro jeho jednoduchý zápis pomocí textového řetězce. SVG byl vytvořen W3C právě k zjednodušení zobrazení vektorové grafiky na webu. Specifikace, vlastnosti, možnosti, podpora nebo přístupnost formátu SVG je uvedena v následujících podkapitolách. (Trippe, 2002)

3.2.1 Specifikace a vlastnosti formátu

SVG lze označit za značkovací jazyk, který vychází z XML. Současně lze Scalable Vector Graphics označit za vektorový formát souboru, který můžeme vytvořit v grafickém editoru, výsledkem ale zároveň také může být textový soubor s popisem tohoto grafického

souboru. SVG se tedy využívá hlavně k integraci a popisu dvourozměrné grafiky a textu. Vektorová grafika v tomto formátu může být velmi bohatá na sémantiku, a zároveň také značně strukturovaná. Strukturovanost vzniká především seskupováním, stylizováním nebo skládáním jednotlivých objektů. SVG také disponuje interaktivitu, skriptování, animace což vše dohromady splňuje požadavky nejvyšší grafické kvality. (Eisenberg, 2014)

Aktuální verze jazyka SVG je SVG 2, jedná se o poslední update ze září roku 2021. Momentálně poslední verzi SVG podporují téměř všechny používané webové prohlížeče. Standard HTML5 (Hypertext Markup Language) také jako SVG vychází z jazyka XML, což umožňuje vkládat SVG obrázek do HTML dokumentu. (SVG Working Group, 2021)

U jazyka SVG se lze setkat s několika základními prvky, jedná se o grafické elementy jako jsou křivky, tvary, úsečky a rastrové obrázky a text. Všechny prvky lze editovat, například je možné měnit jejich velikost, barvu, typ písma nebo barvu výplně, pro elementy lze nastavit tloušťku čáry nebo barvu čar. V SVG lze na všechny prvky také aplikovat 2D transformace a animace. Také disponuje širokou škálou filtrů a efektů jako je rozmazávání, stínování, vykreslování apod. (Eisenberg, 2014)

3.2.2 Import SVG na webovou stránku

Vytvořenou grafiku ve formátu SVG lze zobrazit samostatně v podporovaných webových nebo obrázkových prohlížečích či grafických editorech. To není ovšem cílem vzniku formátu SVG. Cílem SVG je jeho integrace do jiného dokumentu nejčastěji na web, kvůli čemu byl W3C konsorciem vytvořen, tedy do HTML souboru na webové stránce. Způsobů integrace SVG do HTML je více, níže jsou uvedeny základní možnosti.

3.2.2.1 SVG jako vložený obrázek

Protože SVG soubor je formát grafického obrazu lze jej do HTML dokumentu zahrnout i stejnou formou jako jakýkoliv jiný obrázek. To lze udělat dvěma způsoby.

První způsob je vložení obrázku pomocí tagu *img*. Element *img* vyhrazuje obrázku prostor v dokumentu HTML, kde se má obrázek vykreslit. Obrázek se na stránku integruje pomocí odkazu na webový server, kde je uložený. Odkaz se vkládá do atributu v tomto elementu zvaném *src* (source). Tento způsob se doporučuje zejména v případě toho, že obrázek má být základní součástí stránky.

Druhým způsobem je vložení obrázku přímo do CSS (Cascading Style Sheets) jako vlastnost stylu obrázku. Pro to slouží některé vlastnosti, které tento způsob podporují.

Nejvíce se využívá vlastnost `background-image`. Tato vlastnost vykresluje obrázek za textový obsah prvku. Velikost zde lze nastavit pomocí vlastnosti `background-size` a pozici pomocí `background-position`.

Při využití těchto způsobů lze ovšem narazit na určitá omezení. Obrázek je vždy vykreslen odděleně od webové stránky, tudíž není možná žádná komunikace. Následně může dojít, v závislosti na prohlížeči, k převodu na rastrový obrázek a tento obrázek pak ztrácí vlastnosti vektorových dat. Pokud obrázek obsahuje text nelze jej editovat pomocí HTML dokumentu. (Eisenberg, 2014)



Obrázek 2 – SVG v HTML jako vložený obrázek (Eisenberg, 2014)

3.2.2.2 SVG v elementu *object*

SVG lze do HTML také vložit pomocí elementu *object*. Tento způsob integruje externí soubor do HTML bez jakýchkoliv omezení. *Object* element je univerzální způsob, jak vkládat externí soubory libovolného typu do dokumentů HTML. Prohlížeč je může zobrazit, pokud má aplikaci pro interpretaci tohoto typu souboru, nejčastěji se jedná o nějaký zásuvný modul. Umístění souboru je určeno atributem *data*, který element *object* obsahuje. V elementu lze definovat obsah, který se vykreslí v případě, že SVG obraz nevykreslí. To lze využít například pro zadání textu, že prohlížeč neobsahuje aplikaci pro interpretaci SVG. (Eisenberg, 2014)



Obrázek 3 – SVG v HTML jako element object (Eisenberg, 2014)

3.2.2.3 SVG přímo v dokumentu HTML

Další metoda umožňuje kód SVG vložit přímo do souboru HTML. Přímé vložení může urychlit načítání webové stránky, protože není potřeba stahovat samostatný soubor. Důležité tedy je, že HTML při používání skriptů a CSS stylů zachází s SVG souborem i HTML jako s jedním objektem.

První možný způsob uplatnění této metody je využití elementu *foreignObject*. Pomocí tohoto elementu si HTML vyhradí obdélníkovou oblast, do které vloží obsah v jakémkoliv XML formátu, v tomto případě tedy SVG kódu.

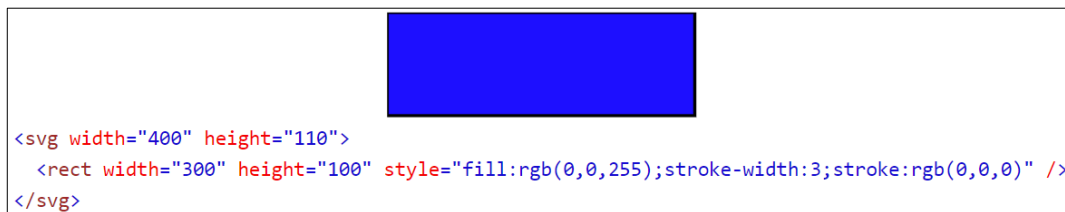
Další způsob se jmenuje Inline SVG a jedná se o přímé začlenění elementu SVG přímo do dokumentu XHTML (Extensible Hypertext Markup Language). V tomto případě při přechodu z HTML kódu do SVG kódu je potřeba definovat všechny prvky SVG ve jmenném prostoru tagu SVG, která je uvnitř HTML. (Eisenberg, 2014)

3.2.3 Struktura dokumentu

SVG dokument je obdobně jako XML postaven na struktuře dokumentu. Tvoří několik hlavních elementů a dalších atributů. Skládá se z hlavičky (*header*), kořenového elementu (*root*) a několika dalších tzv. “child” elementů. Využívá také značně atributy.

Jednotlivě vytvořené entity mohou být předem definovány a následně opakovaně využívány ve zbytku kódu. Toho se může využít pro konstanty nebo definici CSS (kaskádových stylů). Jiným možným způsobem znovupoužití elementů je využití odkazů pomocí tagu *<xlink:href>*. S použitím tagu *<defs>* lze zase v sekci centrálně definovat znovupoužitelné objekty. Elementy v SVG mají implicitní vykreslování grafického obrazu, to znamená, že první element v textovém dokumentu se zobrazí na obrazovce jako první. Další elementy se vykreslují postupně s ohledem na průhlednost vůči předešlým elementům. Většina webových prohlížečů je implementována tak aby nabídla maximální grafickou kvalitu souborů v SVG. Využívají průhlednost, vyhlazování, správu barev apod.

Níže uvedený obrázek ukazuje validní kód dokumentu SVG, který zobrazuje obdélník a kružnici. Kód lze libovolně a snadno editovat. Obrázek zobrazuje kód a jeho obrazový ekvivalent ve formátu SVG. (Peterson, 2003)



Obrázek 4 – SVG – Obdélník (SVG Rectangle, 2023)

3.2.4 Geometrické elementy (tvary)

SVG obsahuje několik geometrických tvarů, které jsou zavedeny jako elementy. Každý tvar umožňuje zobrazovat jiný objekt s jinými vlastnostmi. Podrobnější analýza základních elementů je popsána níže.

- **Path**

- Element *path* neboli cesta je jeden z nejsilnějších elementů SVG. Jsou to v podstatě polygony, které se mohou různě spojovat, oddělovat a obsahovat i soubor polygonů. Také mohou obsahovat spojení přímk, oblouky a různé křivky. Pomocí elementu *path* lze vytvořit v podstatě všechny ostatní geometrické útvary. Jedná se ale o nežádoucí praxi, která se využívá při tvorbě SVG v grafických editorech. Složitost a velikost SVG kódu se pak zbytečně zvětšuje. (Peterson, 2003)

- **Line**

- Nejzákladnějším prvkem v geometrii je přímka. Přímku lze v SVG vykreslit pomocí elementu *line*, jako atributy se zadávají koncové body přímky x a y. Element *line* se vypisuje po vzoru následující ukázky:

```
<line x1="start-x" y1="start-y" x2="end-x" y2="end-y" />
```

(Eisenberg, 2014)

- **Rectangle**

- *Rectangle* neboli obdélník představuje základní tvar mezi geometrickými útvary. Definiuje se zadáním souřadnic x a y levého horního rohu a jeho výškou a šířkou. Vnitřek obdélníku lze vyplnit libovolnou barvou pomocí atributu *style* a určení *fill* (výplň) a *stroke* (tah obrysu), také lze styl ponechat

prázdný pomocí údaje *'none'*. Dále lze definovat průhlednost (*opacity*) výplně a obrysu.

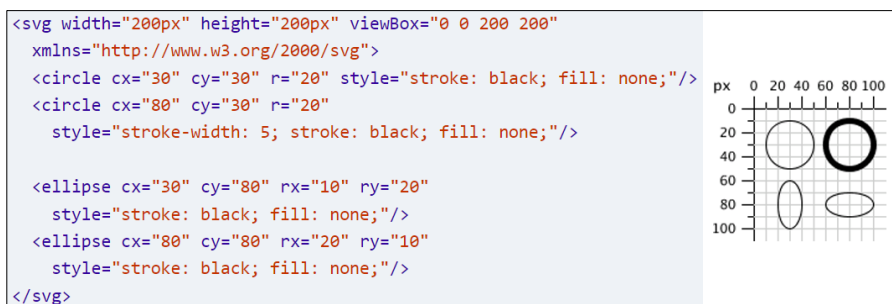
- *Rectangle* dále umožňuje vytvořit obdélník se zaoblenými rohy (*rounded rectangle*), to umožňují atributy *rx* a *ry*, které definují velikost zaoblení rohů. Při zadání pouze jednoho údaje SVG předpokládá, že druhý údaj je stejný. (Eisenberg, 2014)



Obrázek 5 – SVG – Rectangle element (Eisenberg, 2014)

• Circle a Ellipse

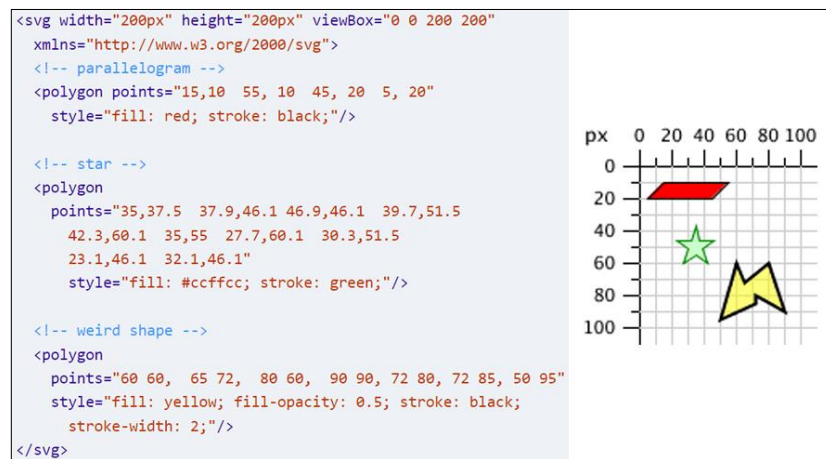
- Pro vykreslení kružnice se využívá element *circle*, který obsahuje atributy souřadnice, tedy zadání *x* a *y* a následně poloměrem *r*, souřadnice poloměru *r* se definují pomocí *cx* a *cy*. Výchozí nastavení výplně a obrysu kružnice je černá barva a bez obrysu.
- Elipsa se definuje elementem *ellipse* a je příbuzná kružnici, ale kromě středových souřadnic také vyžaduje zadání dvou poloměrů, jedná se o atributy *rx* a *ry*. Pokud jsou atributy středu vynechány automaticky se doplní nula a žádný tvar nebude vykreslen. (Eisenberg, 2014)



Obrázek 6 – SVG – Circle a ellipse elementy (Eisenberg, 2014)

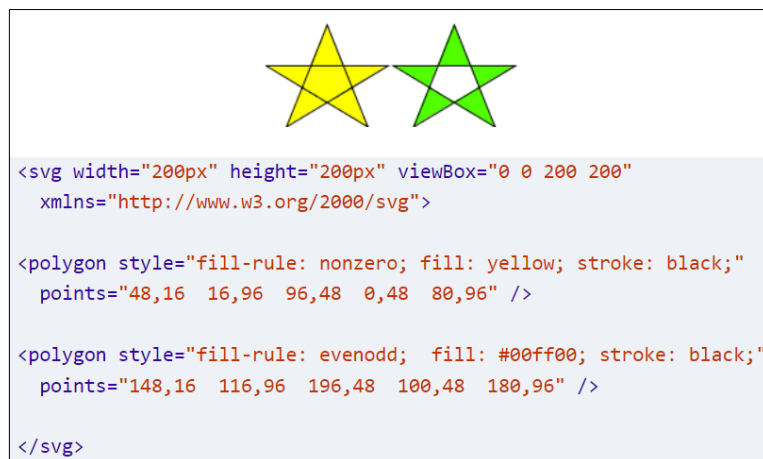
- **Polygon**

- SVG umožňuje vykreslovat více než jen obdélníky, kružnice a elipsy, daleko mocnější nástroj, který nabízí je element *polygon*. Polygon umožňuje vytvořit řadu bodů, které popisují jakýkoliv nepravidelný geometrický tvar, který má být vykreslen. Lze tak jednoduše tvořit například šestiúhelníky, osmiúhelníky, hvězdy nebo jakékoliv libovolné uzavřené tvary. Tyto body se vypíšu opět v souřadnicovém systému s hodnotami x a y do tohoto atributu. Na konci řady se nemusí opakovat první vypsany bod, tvar se uzavře automaticky. Jedinou podmínkou je sudý počet vypsanych bodů, v tomto případě by tvar mohl vytvořit mnohoúhelník. Na následujícím obrázku je zobrazen příklad rovnoběžníku, hvězdy a jiného nepravidelného tvaru. (Eisenberg, 2014)



Obrázek 7 – SVG – Polygon element (Eisenberg, 2014)

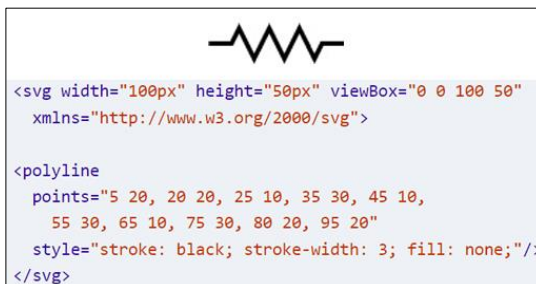
- Speciálním typem polygonů jsou polygony s protínajícími čarami. Výplně polygonů jsou komplikovanější, protože není jednoznačně určeno, která část se nachází uvnitř polygonu a která vně. Jedním pravidlem, které lze použít je *fill-rule*, to má hodnotu buď nenulovou nebo sudou. Nejčastěji se pravidlo používá pro vyplnění hvězdy. Pro přesnější pochopení slouží obrázek níže. (Eisenberg, 2014)



Obrázek 8 – SVG – Efekt rozdílných pravidel výplně polygonu (Eisenberg, 2014)

- **Polyline**

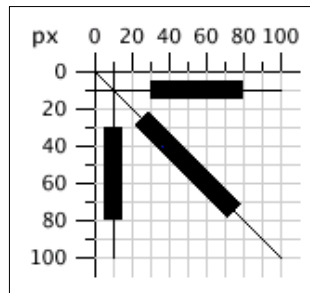
- Při návrhu může nastat situace, kdy je potřeba vytvořit více nepravidelných přímek (*line*), které společně tvoří neuzavřený tvar. Tento tvar lze vytvořit více přímkami, ale to je velice složité a také neefektivní. Pro tyto potřeby slouží element *polyline*. Jedná se v podstatě o podobný element jako *polygon*, ale netvoří uzavřený tvar.



Obrázek 9 – SVG – Polyline element (Eisenberg, 2014)

- **Stroke**

- *Stroke* neboli tah je vlastnost, která se využívá ve všech elementech uvedených výše. Tato vlastnost je definována v atributu *style*. Vždy lze definovat velikost, barva a typ stylu tahu.
- ***Stroke-width*** je vlastnost neboli šířka tahu, která se udává v atributu *style*, kde definuje, jaký bude design daného tahu. Při navrhování stylu tahu je vždy nutno pochopit jak se přímkou nebo geometrický objekt bude vyobrazovat na podkladu. Plátno, na kterém se tvoří obraz v SVG formátu se skládá z podkladové mřížky, každá vytvořená přímkou (*line*) vždy leží uprostřed tahu. Vše je znázorněno na obrázku níže.



Obrázek 10 – SVG – stroke-width (Eisenberg, 2014)

- **Stroke-color** je vlastnost v atributu *style*, která určuje barvu tahu. Barvu lze zadat různými způsoby. Nejjednodušší způsob je zadání názvu barvy do atributu, v tomto případě je na výběr pouze užší paleta barev. Pro zadání přesné barvy slouží hexadecimální zápis kódu barvy v barevném formátu RGB (Red, Green, Blue), lze jej vypsát šestimístním kódem ve formátu *#rrggb* nebo třímístním kódem ve formátu *#rgb*. Dále lze RGB také definovat slovně ve formátu *rgb(red-value, green-value, blue-value)*. Každou hodnotu lze zapsat číslem 0–255 nebo 0 % – 100 %.
- **Stroke-opacity** je vlastnost v atributu *style*, která určuje průhlednost. Průhlednost tahu lze modifikovat tak, že se zadá hodnota průhlednosti (*stroke-opacity*) od 0,0 do 1,0, kde 0 je zcela průhledná a 1 je zcela neprůhledná. V případě zadání hodnoty menší než 0 se změní na 0 a v případě zadání hodnoty větší než 1 se změní na 1.
- **Stroke-dasharray** je vlastnost v atributu *style*, která určuje typ stylu tahu. Tah může být například přerušovaný, nepřerušovaný nebo tečkovaný. Hodnota tohoto atributu se zapisuje seznamem čísel, který určuje délku pomlčky a mezery.

3.2.5 Souřadnice, transformace a zobrazení

SVG je formát dvourozměrné grafiky, tudíž je zřejmé, že je založen na souřadnicovém systému, konkrétně na kartézském souřadnicovém systému, který začíná v levém horním rohu. Díky možnosti vnořování souřadnic lze v SVG využít rovněž souřadnicové systémy z odlišných zařízení a také souřadnice reálného světa.

Možnost využití „real world“ souřadnic je velmi zajímavá vlastnost, která umožňuje snadnou integraci různých datových zdrojů a také snadnou manipulaci s objekty.

SVG také umožňuje mnoho transformací, jako jsou translace, rotace, zkosení, škálování a maticové transformace. Všechny lze společně kombinovat a vnořovat. (Peterson, 2003)

3.2.6 Stylování a grafické efekty

SVG umožňuje specifikaci různých grafických vlastností jako je *stroke-width*, *stroke-color*, *stroke-opacity*, *dashing*, *line-caps* a *mitter-joins*. Elementy line mohou obsahovat připojené značky na svých koncových bodech a mohou být rozděleny do jednotlivých objektů.

Pro vytváření složitějších stylů je možné propojovat a znovu použít již existující *line* elementy, což umožňuje nastavení různých stylů na elementech *line* ve stejném objektu. Díky tomu je ve vektorové grafice možné vytvářet velmi detailní a komplexní kresby.

Při vyplňování daných elementů lze volit mezi jednotlivými barvami, gradienty, vektorovými a rastrovými vzory. Centrálně definované vzory mohou obsahovat SVG kód obsahující animace. To může sloužit například pro vytváření animovaných map nebo různých grafických jevů s nimi spojených.

Důležité je také to, že všechny grafické atributy, jako je *fill*, *stroke* a *text*, lze definovat jako styly. SVG umožňuje využívat styly založené na CSS i XML. Styly mohou být využity v rámci *xhtml/html/svg* a dalších webových standardů, které jsou kompatibilní s CSS nebo XSL (Extensible Stylesheet Language) styly.

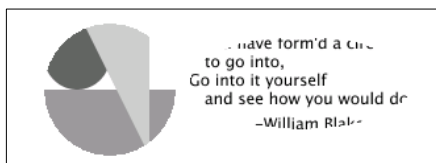
Velká jedinečnost SVG spočívá v možnosti aplikace filtrů, což nenabízí jiné vektorové webové standardy. SVG umožňuje filtrovat nejen rastrové obrázky, ale i všechny objekty. Filtry mohou být aplikovány na libovolný objekt a mohou obsahovat různé efekty, jako je například *lightning*, *shadow*, *blurring*, *blending*, *compositing*, *flooding*, *merging*, *turbulence*, *erosion* apod. Tyto filtry mohou být kombinovány a ovlivňovat pouze určité části objektů. (Peterson, 2003)

3.2.7 Ořez a masky

Grafici se často setkávají se situacemi, kdy není žádoucí vidět celý obrázek. Ke změně nebo zamaskování jakéhokoliv objektu, cesty, textu nebo geometrického tvaru v SVG dobře slouží nástroj ořezu a masky. *Masking* neboli maskování využívá principu průhlednosti (*opacity*), jednoduše lze objekty překrývat a s různým procentem průhlednosti zviditelnit či

zneviditelnit. *Clipping* neboli ořezávání cest lze využít k uzpůsobení jednotlivých tvarů. (Peterson, 2003)

Vlastní ořezová oblast se vytváří pomocí elementu *clipPath*. Nejprve je do kódu vložen obrázek *minicat.svg*. Samotný ořez se provádí v elementu *clipPath*, kde se nachází další element *rect*, který určuje, že se obraz bude ořezávat do tvaru obdélníku. Tento obdélník se nezobrazuje, ale pouze nám ukazuje souřadnice ořezu. Na obrázek, který se ořezává se přidává vlastnost *clip-path style*, jejíž hodnota odkazuje na prvek *clipPath*.



Obrázek 11 – SVG – Použití ořezu (Eisenberg, 2014)

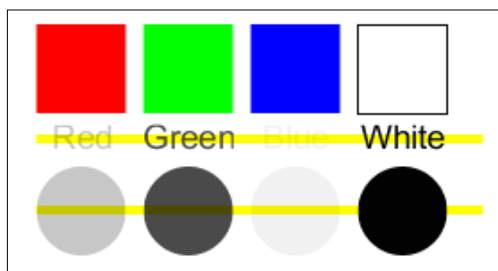
Maskování v SVG přenáší průhlednost na objekt, který maskuje. V místech neprůhlednosti masky jsou neprůhledné všechny pixely maskovaného objektu. V místech, kde je průhlednost větší je vidět i objekt.

Maska se vytváří pomocí elementu *mask*. Rozměry určují jako obvykle atributy *x* a *y*, což symbolizuje šířku a výšku. V elementu lze použít všechny základní tvary, které jsme si již vyjmenovali (cesty, geometrické tvary, text a jiné objekty). Výchozí nastavení souřadnic těchto elementů je ve výchozím uživatelském souřadnicovém systému.

Samotná průhlednost se v SVG určuje pomocí barevného modelu RGBA, tedy i s Alfa průhledností. Odlišností je, že SVG používá pro výpočet průhlednosti všechny dostupné informace, konkrétně využívá tento vzorec:

$$\text{Alfa} = (0.2125 * \text{red value} + 0.7154 * \text{green val.} + 0.0721 * \text{blue val.}) * \text{opacity val.}$$

Na obrázku níže lze zpozorovat, jak se výsledná hodnota alfa zmenšuje úměrně s tmavnutím barvy. Výsledkem je méně průhledný maskovaný objekt.



Obrázek 12 – Efekt průhlednosti maskování (Eisenberg, 2014)

3.2.8 Text

Text je jeden z nejdůležitějších grafických prvků všeobecně. SVG podporuje všechny vlastnosti textu, které jsou dostupné v DTP (Desktop publishing), např. fonty, styly, varianty, velikosti, roztažení atd.

V typografii všeobecně a také v tomto elementu *text* se lze setkat s pojmy *character* (znak), *glyph* (glyf) a *font*. *Character* neboli znak v dokumentu XML nebo SVG značí znak, bajt nebo bajty s číselnou hodnotou podle standardu Unicode. Například písmeno g, je znak s hodnotou Unicode 103, jednoduše řešeno nazývá se tak jeden jedinečný znak. *Glyf* je viditelná reprezentace znaku nebo znaků. Jeden znak může mít mnoho různých glyfů, které ho reprezentují. Častěji lze setkat s pojmem *font*, což je kolekce glyfů, který reprezentuje určitou sadu znaků.

Text v SVG lze vkládat pomocí elementu *text*, který může popisovat buď celou sadu písem, nebo jen potřebné glyfy. Každý jednotlivý glyf je vytvořen a popsán pomocí prvků vektorové grafiky tedy SVG (cest, křivek, bodů apod.). To umožňuje každému grafikovi snadno si vytvořit vlastní font. SVG podporuje tzv. internacionalizaci, což umožňuje Unicode, obousměrný text a text zprava doleva. Pomocí elementu *switch* lze kontrolovat nastavení jazyka prohlížeče a zobrazovat příslušný jazyk. (Peterson, 2003)

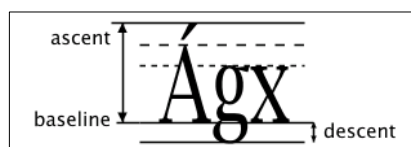
Element *text* nabízí prakticky stejné atributy jako u kaskádových stylů (CSS). Jedná se o atributy *font-family*, *font-size*, *font-weight*, *font-style*, *text-decoration*, *word-spacing*, *letter-spacing* apod. Ukázka elementu *text* s různými atributy je na obrázku níže.



Obrázek 13 – SVG – *text* element (*weight*, *style*, *decoration*) (Eisenberg, 2014)

Při tvorbě glyfů a fontů se návrháři a grafici vždy opírají o základní charakteristiky, které využívá každý font. Jedná o linie, které se vytvoří v grafickém návrhu glyfu. Základní přímkou neboli *baseline*, tvoří spodní základ glyfu. Výška k nejvyššímu (glyfu) písmenu ve fontu se nazývá *ascent*. Zbylá část od *baseline* k nejspodnějšímu okraji glyfu je *descent*. Celková délka od spodu glyfu tedy od linie *descent* až k horní části linie *ascent* se nazývá *em-height*. *Em-height* se dále dělá na *cap-height*, což označuje délku velkých písmen ve

fontu, a *ex-height*, což označuje délku malých písmen ve fontu. Obrázek níže graficky popisuje glyf a jeho linie. (Eisenberg, 2014)



Obrázek 14 – Míry glyfu (Eisenberg, 2014)

3.2.9 Animace

Všechny prozatím analyzované vlastnosti formátu SVG byly ve statickém obraze. Díky animaci lze statické obrazy rozpohybovat čímž se tato SVG vektorová grafika stává ještě zajímavější. SVG nabízí dvě metody animací.

První přístup k animaci lze implementovat pomocí metody DOM (Document Object Model) a skriptování. Tento přístup je oproti následujícímu složitější, ale za to více flexibilní. Jednodušeji lze kontrolovat metodu interpolace a časové kroky.

Deklarativní neboli popisná animace je přístup, který pracuje s příkazy jazyka SMIL3 (Synchronized Multimedia Integration Language Level 3), což je jazyk, který je vytvořen konsorciem World Wide Web a založen na XML a slouží pro popis, integraci a synchronizaci animací. Tento přístup je vcelku jednoduchý ale přitom dostatečně výkonný, aby se dal ve většině případů implementovat.

Ve SMIL se zadávají počáteční a koncové hodnoty atributu, barvy, pohybu nebo transformace, které chceme animovat. Následně se zvolí čas, kdy má animace začít, a doba trvání animace. (Peterson, 2003)

SMIL umožňuje animovat jak grafické, tak i negrafické atributy. Pro různé způsoby animací zahrnuje SMIL různé grafické elementy:

- ***animate***
 - Element *animate* umožňuje animovat každý atribut prostřednictvím lineární *keySpline* a diskrétní *interpolace*. Tím lze animovat například pohyb, změny velikosti, barvy apod.
 - Element *animate* může také obsahovat atribut *keyTimes*, který uchovává zlomky trvání celé animace a atribut *keyValues*, který popisuje hodnotu v čase. Atributy *from* a *to* umožňují definovat počáteční a koncové hodnoty a atribut *by* určuje relativní hodnoty posunu animace.

- ***set***
 - Element *set* interpoluje nečíselné hodnoty, jako jsou boolean hodnoty nebo řetězce. Lze jej použít k nastavení atributů na konkrétní hodnoty v určitých časových okamžicích.
- ***animateMotion***
 - Element *animateMotion* popisuje pohyb elementu podél trajektorie pohybu. Lze jej využít pro animaci pohybu objektu po předem definované dráze.
- ***animateColor***
 - Element *animateColor* interpoluje a animuje barvy. Pomocí něj lze vytvářet animace změn barev v průběhu času.
- ***animateTransform***
 - Element *animateTransform* umožňuje animovat transformace, což zahrnuje změny velikosti, rotace, zkosení a další transformace. (Peterson, 2003)

V animaci často vyčnívá pojem *interpolace*. Interpolace je původně pojem z numerické matematiky, který hledá přibližnou hodnotu v určitém intervalu s předpokladem, že její hodnota je obsažena v jiných bodech daného intervalu. SMIL používá různé interpolační mechanismy. Nejčastěji se jedná o diskrétní a lineární interpolaci, interpolaci podle tempa a interpolace spline. Použití interpolace keyspline je složitější, ale umožňuje simulovat zrychlení, zpomalení a gravitační efekty.

3.2.10 Přístupnost

Přístupnost je jeden z velmi důležitých aspektů celého World Wide Webu a často důležitý faktor při tvorbě HTML. Není tomu jinak u SVG. Je důležité zajistit, aby obsah SVG grafiky byl použitelný a srozumitelný pro jedince s postižením, včetně těch, kteří využívají čtečky obrazovky a asistenční technologie. U SVG souborů je důležité, aby SVG kód byl vždy sémantický a zahrnoval funkce a elementy pro zlepšení uživatelské přístupnosti.

K pochopení grafických prvků velmi napomůže uvedení smysluplného textového obsahu do textových elementů *text* a *tspan*, které sdělují stejné informace jako grafické prvky. Nápomocné může být také využití popisných názvů, jako jsou elementy *titles* a *labels*. Při použití *title* elementu uvnitř *svg* elementu lze podrobně popsat titul pro obsah celého SVG. U elementu *label* je také důležité popsáním přiřadit daný label k specifickému elementu v SVG. To může být žádoucí například při tvorbě grafů.

Dále lze využít element *use* k organizaci a uskupení souvisejících grafických prvků. Při potřebě uvést delší popis obsahu SVG lze zase využít element *desc*.

Základním pravidlem je užití dostatečného barevného kontrastu mezi popředím a pozadím. Důležité je vždy použít barvu písma dostatečně kontrastní k jejímu pozadí. Pro elementy, které nejsou textového charakteru je samozřejmostí využít atribut *alt* s alternativním textem, který popisuje daný prvek.

Dalším prvkem je možnost využití klávesové navigace. Zejména pro interaktivní prvky je důležité, aby byly přístupné pomocí klávesnice.

Do celého dokumentu je pak nutné na konci uvést a okomentovat funkce přístupnosti, které v SVG dokumentu byli implementovány.

Na obrázku níže je jeden z možných vzorů implementace přístupnosti. Obrázek reflektuje elementy *alt* a *role* využitě ke zlepšení přístupnosti daného prvku. (Eisenberg, 2014)



Obrázek 15 – SVG elementy *alt* a *role* (Accessible SVGs, 2021)

3.2.11 Historie verzí formátu

World Wide Web Consortium vyvíjí formát SVG již od roku 1999. Po delší snaze vyvinout jazyk pro vektorově orientovanou grafiku. SVG Working Group, skupina, která byla konsorciem vytvořena, se rozhodla vytvořit nový značkovací jazyk, který nebude komerční a bude na předešlých značkovacích jazycích.

První verzí, kterou SVG Working Group vytvořila byla verze SVG 1.0 ze 4. září 2001. Původní verze obsahovala základní podporu pro vektorovou grafiku a jednoduchou interaktivitu. Následovala verze SVG Tiny 1.1 z roku 2003 cílila na mobilní zařízení a jiná zařízení s omezenými zdroji. V roce 2008 přichází další verze SVG Tiny 1.2, která vyšla jako koncept a finálně byla doporučena konsorciem až v různých jejích podobách SVG 1.2 Tiny, SVG 1.2 Basic a SVG 1.2 Full. Verze Tiny byla vyvíjena pro mobilní zařízení

a zlepšená byla zejména podpora pro mobilní vyhledávače. Verze Basic přinesla více funkcí oproti verzi 1.1 a byla vhodná pro prostředí s omezenějšími zdroji. Verze Full, jak napovídá název, obsahovala širší spektrum funkcí a možností a byla celkově komplexnější.

Nejnovější generací se stala verze SVG 2, která zrušila některé funkce z verze SVG 1 a přidala funkce HTML5. Cílem druhé verze bylo přinést významná vylepšení a zdokonalení formátu, podpory responzivního designu, rozšířit možnosti skriptování a přidat další geometrické útvary. SVG odebrala některé elementy a jiné zase byly přidány. SVG Working Group přijala SVG 2 v roce 2016 a její další upgrady byly zveřejněny v roce 2018. Poslední oficiální verze dle webu W3C je verze z října roku 2018, ovšem poslední úpravy a neoficiální verze jsou z roku 2021. (SVG Working Group, 2021)

3.2.12 Podpora ve webových prohlížečích

Pro existenci formátu SVG je klíčová jeho podpora na webu, tedy na webových prohlížečích. SVG je velmi dobře podporován všemi moderními webovými prohlížeči. Poté co se SVG formát široce rozšířil díky jeho univerzálnosti, rozšířila se samozřejmě i jeho podpora ve webových prohlížečích a kompatibilita s webovými technologiemi.

Nejpoužívanějším prohlížečem je Google Chrome, který má podporu SVG již od jeho první verze. Nabízí velmi silnou podporu formátu a je považován za nejlepší prohlížeč renderující SVG grafiku. Podporuje tedy jak zobrazení SVG souborů, tak také jejich další vlastnosti animaci a interaktivitu.

Na podobné úrovni podpory je také Mozilla Firefox, která nabízí také přesné renderování SVG obrazu a velmi dobrou funkci vykreslování.

Safari od společnosti Apple má také velmi dobrou podporu SVG formátu, pouze v některých filtrech lze najít menší problémy, v celkovém výsledku jsou ale nepodstatné.

Jelikož Opera je postavena na podobných základech jako Google Chrome je na srovnatelné úrovni podpory. Nemá tudíž s formátem sebemenší problém.

Většina těchto prohlížečů má také svoji mobilní verzi, zde je podpora SVG také na velmi dobré úrovni napříč všemi platformami. Podpora je na vysoké úrovni jak u iOS Safari, tak také u defaultních Android's WebView.

Jediným zádrhelem mohou být starší verze prohlížeče Internet Explorer, které mají pouze limitovanou podporu SVG. Jedná se však pouze o verze IE 8 a nižší, které se v dnešní době již nevyužívají. Ve verzi IE 9 je již podpora vylepšená. Internet Explorer ovšem

v dnešní době není preferovaný vyhledávač ani u Microsoftu, jeho prioritou je prohlížeč Microsoft Edge.

Edge již nabízí bezproblémovou podporu formátu a každou další verzí ji zlepšuje, podpora interaktivity a animací tedy samozřejmě nechybí.

Pro zajištění nejlepší kompatibility a nejrychlejšího vykreslování je ovšem také velmi důležité dodržovat nejlepší praktiky při tvorbě SVG souboru, tedy zajistit dobře strukturovaný SVG kód a vyhnout se užíváním zastaralých atributů a elementů.

Celkově je podpora SVG na webu na nejvyšší úrovni a i z tohoto hlediska lze konstatovat, že budoucnost SVG na webu je více než jistá. (Meghan, 2021)

3.3 Softwary pro tvorbu SVG map

SVG formát je zcela jedinečný i právě proto, že jej lze vytvořit v grafických editorech. Tento formát grafického souboru totiž lze vygenerovat po vytvoření vektorového obrazu ve vektorové grafice. Grafik tak může jednoduše vytvořit libovolnou grafiku ve svém grafickém editoru a pak obraz vygenerovat ve formátu SVG a získá tak grafický návrh a zároveň jeho popis v jazyce SVG, který lze pak jednoduše zobrazit na webu.

Podkapitoly níže obsahují analýzu nejčastěji užívaných softwarových nástrojů pro editaci SVG souborů. Softwary se liší zejména z hlediska funkce, složitosti a také jejich ceny. Výběr vždy závisí na specifických potřebách každého grafika. Například pro tvorbu komplexních map vyhovují nejvíce Adobe Illustrator nebo Inkscape, ale pro jednodušší a snadno stvořitelné mapy zase nejlépe vyhovují online nástroje, jako jsou Simplemaps nebo Scribble Mapps. (Trippe, 2002)

3.3.1 Adobe Illustrator

Adobe Illustrator je vektorový grafický editor, který se běžně využívá k tvorbě a editaci různých vektorově orientovaných grafických prací, což zahrnuje zejména tvorbu log, ilustrací, různých objektů a ikon, ale také map. Podporuje formát SVG se všemi jeho vlastnostmi. Nabízí se tak jako výkonný nástroj pro práci s SVG soubory.

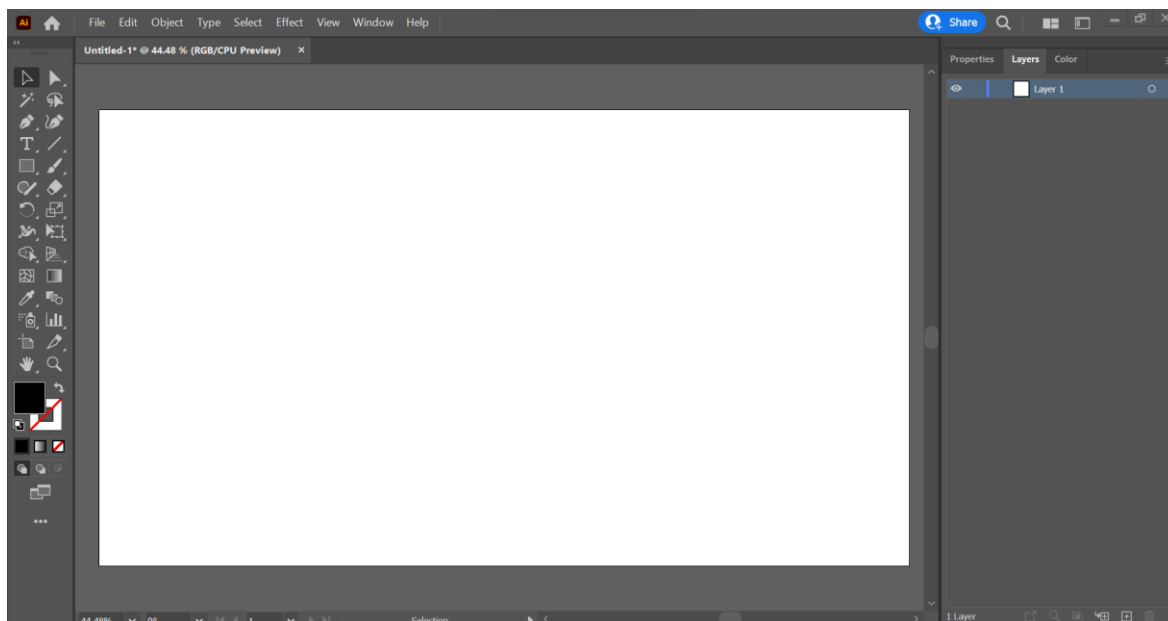
Illustrator obsahuje nativní podporu importu i exportu SVG souborů. Lze tedy vytvořit nový soubor v SVG nebo otevřít a editovat již vytvořený soubor. SVG formáty jsou nezávislé na rozlišení, což znamená, že mohou být editovány v jakémkoliv měřítku bez ztráty kvality, což zajišťuje, že SVG grafika zůstane ostrá a čistá bez ohledu na to na jakém zařízení bude zobrazována.

Dále umožňuje organizaci grafického obrazu s využitím vrstev. Lze jednoduše rozdělit jednotlivé elementy do odlišných vrstev, což zjednodušuje manipulaci a tvorbu s vrstvami, jednotlivými objekty a ve výsledku s celkovým návrhem.

Rovněž nabízí nástroje pro stylování jednotlivých SVG elementů, lze v něm jednoduše aplikovat tahy, výplně a gradienty a také stíny a textury. Pro aplikování těchto stylů je samozřejmě potřeba soubor uložit jako formát SVG.

Pro tvorbu SVG map v tomto editoru Illustrator nabízí velmi dobré textové nástroje. Text lze tvořit a formátovat přímo v mapě, což lze využít zejména pro přidání labelů, legendy a dalších textových popisků.

Po vytvoření návrhu lze jednoduše návrh exportovat do SVG souboru. Illustrator nabízí více možností exportu souboru do SVG. SVG návrh lze optimalizovat pro užití na webu, upravovat přesnost cest nebo spravovat vkládání písem. (Adobe Illustrator, 2023)

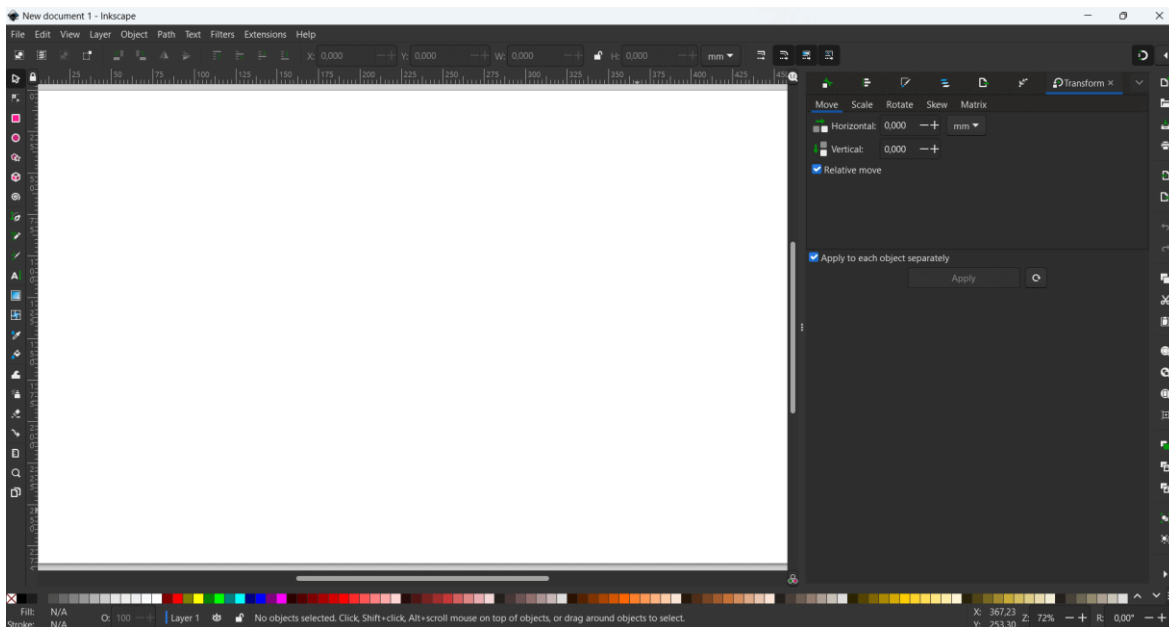


Obrázek 16 – Software Adobe Illustrator

3.3.2 Inkscape

Inkscape je open-source vektorový grafický editor, který je zcela zdarma a je vhodný pro tvorbu SVG map. Je to také velmi oblíbená varianta pro kartografy, grafické designéry nebo pro kohokoliv kdo potřebuje tvořit detailní a škálovatelné mapy bez potřeby profesionálního softwaru. Má nativní podporu SVG formátu, umožňuje, jak vytvoření nového souboru, tak editaci již vytvořeného souboru. Nabízí také širokou škálu nástrojů včetně tvorby cest a různých tvarů, které lze uložit jako SVG grafické elementy.

Při exportu souboru umožňuje optimalizaci SVG souboru pro web. To lze provést úpravou kódu manuálně nebo také pomocí externích nástrojů, které Inkscape nabízí, lze jimi například snížit velikost souboru nebo snížit loading time. (Inkscape Website Developers, 2023)



Obrázek 17 – Software Inkscape

3.3.3 QGIS

QGIS neboli Quantum Geographic Information System je jeden z populárnějších GIS (Geographic Information System) softwarů. Využívá se pro práci s geografickými a prostorovými daty, zejména slouží pro tvorbu map. Nabízí user-friendly interface, které je velmi intuitivní, a proto je přístupný jak pro začátečníky, tak pro pokročilé uživatele. Podporuje většinu datových formátů. Nabízí mnoho funkcí včetně tvorby a designování map, což zahrnuje také možnost přizpůsobovat symboly, labely nebo celé layouts.

QGIS umožňuje integraci s dalšími GIS nástroji, včetně webových nástrojů, což umožňuje přístup k datům z různých zdrojů. V QGIS lze také editovat přímo geografická data. (Discover QGIS, 2023)

3.3.4 Mapbox studio

Mapbox je široce užívaná platforma pro tvorbu map. Využívá se také pro integraci map na web a do mobilních aplikací. Software nabízí speciální nástroje pro práci s údaji o poloze. Poskytuje webové prostředí pro design map, kde lze tvořit stylovat a upravovat mapy se všemi specifickými potřebami.

Nabízí velkou škálu mapových stylů, které lze využít jako vzor nebo základ pro tvorbu konkrétnější mapy. To zahrnuje základní, satelitní, outdoorové, terénní nebo dopravní mapy.

Software také obsahuje geokódování, které umožňuje konvertovat adresy nebo názvy míst do geografických souřadnic (zeměpisná šířka a délka), což se může využít při mapovém nebo lokačním vyhledávání. Dále lze velmi dobře vizualizovat všechna data přidáním různých vrstev do map.

Mapbox je využíván zejména v ride-sharingových aplikacích, logistice, rekreacích v přírodě a dalších. Jedná se o všestrannou platformu pro vytváření a integrování map do různých platform a jiných projektů. (Mapbox Studio, 2023)

3.3.5 Simplemaps a Scribble Maps

Mezi nejznámější tzv. *web-based mapping tools* (webové softwary pro tvorbu map) patří Simplemaps a Scribble Maps, které nabízejí svoji free verzi. Jedná se tedy o snadno spustitelný software přímo z webu, který je zdarma.

Simplemaps je software, který je nejlepší pro tvorbu vlastních map. Obsahuje také knihovnu již předpřipravených map, která zahrnuje i velkou databázi map specifických území. Je znám především pro jeho user-friendly uživatelské rozhraní, je navržen proto, aby v něm mohli mapy tvořit i opravdoví začátečníci s minimálními zkušenostmi. Nabízí free verzi s omezenou funkcionalitou, v placené verzi jsou k dispozici pokročilejší funkce. (About the Online Map Customization Tool, 2023)

Scribble Maps software je ideální pro kreslení a anotování map. Podporuje spolupráci v týmu, na jednom projektu může současně pracovat více uživatelů. Do map lze jednoduše přidávat vlastní trasy, tvary a texty. Lze integrovat mapy z jiných GIS nástrojů, které lze následně upravovat a exportovat v jakýchkoliv jiných formátech. Do map lze také integrovat náhledy z Google Street View. (Unleash your data and imagination with Scribble Maps, 2023)

3.3.6 Zhodnocení a výběr vhodného softwaru

Ze zmíněné analýzy vyplývá, že geografický informační systém QGIS, je vhodný spíše pro komplexnější geografické mapy a není primárně orientován na web. Software Mapbox studio je vhodný pro tvorbu nových kartografických map. Softwary Simplemaps a Scribble Maps jsou online nástroje vhodné pro tvorbu vlastních map k osobnímu využití, kreslení

a anotování. Jako vhodné nástroje pro tvorbu praktické části této práce se tedy jeví Adobe Illustrator a software Inkscape, které slouží pro tvorbu primárně vektorové grafiky.

Po analýze a porovnání všech grafických editorů byl pro praktickou tvorbu této práce vybrán profesionální software Adobe Illustrator, protože nabízí intuitivní uživatelské prostředí, nejvíce funkcí a nejlepší podporu SVG. Illustrator nabízí velké množství funkcí pro tvorbu a úpravu vektorové grafiky a zároveň výborně podporuje formát SVG, jeho export a možnosti publikace, což je právě v této práci velmi žádoucí.

3.4 Webová kartografie a využití SVG

Webová kartografie je podobor kartografie, která se zabývá tvorbou map. Zaměřuje se na tvorbu, návrh, implementaci a šíření map na internetu skrz web a webové aplikace. Tento proces často vyžaduje kooperaci mnoha digitálních technologií, zejména se propojují oblasti geografických informačních systémů (GIS) a nástroje pro rozvoj world wide webu, což ve výsledku umožňuje produkovat a distribuovat interaktivní a dynamické mapy na webu.

Webová kartografie zahrnuje zejména tvorbu interaktivních map, které umožňují uživateli interagovat s geografickými informacemi přímo v dané webové aplikaci, což může zahrnovat například zoom, tedy přiblížení nebo oddálení mapy, interaktivně si zobrazovat jednotlivé informace ze štítků v mapách. V neposlední řadě je velkou výhodou interaktivních map možnost překrývat mapu různými vrstvami.

Mapy na webu také často umožňují zobrazovat tzv. real-time data, tedy aktuální informace například o dopravní situaci, počasí nebo up-to-date aktuality ze sociálních médií. Tyto elementy se zobrazují v podobě interaktivní legendy mapy, vyskakovacích oken nebo dynamických štítků.

Mapy na webu jsou také designovány tak aby mohly sloužit na různých platformách, to zajišťuje kompatibilita v odlišných webových prohlížečích nebo zařízeních, což přináší daleko větší spektrum uživatelů.

S využitím webové kartografie se lze setkat v mnoha oborech. Nejznámější jsou GPS (Global Positioning System) navigace, což jsou zejména online mapy, které zaměřují naši lokaci a hledají směr a zájmové body. Využití lze najít také u výzkumníků, v podnicích a vládních organizacích, které využívají webovou kartografii k vizualizaci a přenosu dat. V turismu se lze setkat s webovou kartografií při plánování výletů, vizualizací turistických destinací apod. Dále například v aplikacích pro plánování tras v městské dopravě, pro

monitorování životního prostředí nebo přírodních zdrojů a v neposlední řadě ve vzdělávání pro jakékoliv akademické účely. (Nétek, 2020)

3.4.1 Interaktivita a animace v SVG mapách

Jednou z výhod, které přinesl formát SVG je možnost animace a interaktivity. Právě tyto vlastnosti lze dobře využít v SVG mapách.

Interaktivitu můžeme docílit mnoha způsoby. SVG mapy lze jednoduše zoomovat, tedy přibližovat, oddalovat a také obsahuje funkcionality tzv. panningu. Panning je vlastnost mapy, která umožňuje snadné posouvání a plynulou navigaci po mapě. Tzv. klikací elementy pak umožňují lépe interagovat s uživatelem. Jedná se většinou o značky, body zájmu nebo rovnou různé regiony. Odtud na uživatele může vyskočit vyskakovací okno nebo se odkáže přímo na jinou webovou stránku. Pro nezahlcení mapy textem lze uživatelům poskytnout textové informace, které se zobrazí po najetí myši na určitý element v mapě. Další výhodou je přepínání mezi určitými vrstvami mapy. Na to v mapě slouží obvykle tlačítko zvané toogle, pomocí kterého lze například přepínat v dopravních mapách na zobrazení pouze železnic nebo silnic. Uživatel se tak lépe může soustředit právě jen na informace, které zrovna potřebuje. Velmi užitečné je také vyhledávání a filtrování v mapách, které interaktivita nabízí. Uživatel může jednoduše vyfiltrovat pouze informace, které ho zajímají na základě určitých kritérií nebo kategorií.

Animace v SVG mapách si lze všimnout například při různých přechodech při přepínání vrstev nebo filtrování, přechody tak můžou tento proces více vizuálně zatraktivnit. Vizualizace dat lze měnit v čase, například lze mapu, která ukazuje populační růst měnit v animaci a ukázat tak postupný růst a snížení v různých regionech atd. Dynamické změny se projeví nejlépe na real-time změnách, které nejlépe symbolizují počasí v reálném čase. Speciálně formát SVG umožňuje animovat jednotlivé cesty vektorového obrazu, což může napodobit pohyb objektu na mapě, jako například trajektorii letu, jízdu auta nebo hiking trail. (Nétek, 2020)

3.4.2 Optimalizace SVG map na web

Při integraci SVG map na web k zobrazení geografických dat na webu často nastává také spousta problémů, kterým je potřeba se dopředu vyvarovat. Tak jako všeobecně pro formát dat SVG je důležitým aspektem kompatibilita daného prohlížeče. V dnešní době mohou narušit kompatibilitu pouze některé staré verze prohlížeče Internet Explorer, přesto

je však důležité nepodcenit testování vytvořených map a vyzkoušet je na různých verzích mnoha prohlížečů.

Optimalizace výkonů je také důležitý faktor pro zajištění správné integrace. S každým dalším elementem se v SVG mapě zvyšuje komplexita mapy. Zjednodušováním cest a geometrických útvarů lze minimalizovat zatížení při vykreslování map. Zjednodušení SVG kódu může pomoci k zvýšení výkonu. Odebrání bílých znaků, komentářů, a dalších nepotřebných dat, jednoduše řečeno menší soubory vedou k zrychlení času vykreslování. Efektivní využití CSS stylů je také velmi důležité. Při využití nadměrných nebo složitých CSS stylů se vykreslování může mnohonásobně zpomalit. Využití externích CSS souborů a optimalizace selektorů CSS může výkonu výrazně pomoci. Při použití map s velkou datovou náročností se vyplatí využít asynchronního načítání pro zlepšení počáteční doby načítání stránky.

Velmi důležitá je přístupnost, která SVG mapy přivádí k širšímu spektru uživatelů a zvyšuje standardy přístupnosti. Několik příkladů nám může pomoci vylepšit uživatelskou přístupnost. U netextových elementů v SVG jako jsou ikony nebo symboly, je důležité uvést tzv. *alt text*, který umožňuje tyto prvky přečíst uživatelům, kteří používají čtečky obrazovky. Standardem je v SVG využití sémantických elementů jako jsou *title* nebo *desc*, které přidávají další popis obsahu. Interaktivní prvky jako jsou tlačítka nebo odkazy by měly být navigovatelné také pomocí kláves a měly by zobrazovat indikátory zaostření. Zajištění úměrného kontrastu mezi textem a grafikou zajišťuje možnost správně číst obraz i uživatelům se zrakovými vadami. Správná přístupnost map se zajistí pravidelným testováním pomocí asistenčních technologií, například čteček obrazovky.

Při zajištění všech těchto základních překážek, které mohou nastat lze SVG mapy efektivně integrovat do jakýchkoliv webových aplikací a zajistit tak přívětivý uživatelský zážitek pro všechny uživatele bez ohledu na schopnosti nebo zařízení, které využívají. (Murray, 2013)

3.4.3 Budoucnost SVG map

Význam SVG map ve webové kartografii se stává stále zásadnějším, a proto se SVG mapy budou vyvíjet kontinuálně s vývojem oblasti celé webové kartografie a vizualizace map. Způsob, jakým se v budoucnu bude pracovat s geografickými daty se bude vyvíjet mnoha směry, každopádně formát SVG v tom bude mít zásadní roli.

Mapy by se pravděpodobně mohly zaměřit na ještě realističtější technologie. Zejména se bude jednat o zapojení 3D prostoru. Tedy ve formě 3D mapování, vykreslování 3D terénů, budov a krajiny přímo v prohlížečích. Také interaktivita by měla v budoucnu stále hrát zásadní roli, očekává se větší zapojení uživatelů a větší pozornost se bude ubírat hlavně na funkce ovládání gesty, hlasovými příkazy, samozřejmě vše bude ovlivňovat také umělá inteligence.

Se stále se rychle vyvíjecími obory rozšířené a virtuální reality je pravděpodobné, že zde budou SVG mapy hrát taky důležitou roli. Bude se jednat zejména o aplikace jako jsou navigace, hry a interaktivní průvodci. V oboru IoT (Internet of Things) by mapy do budoucna mohly sloužit jako společné rozhraní pro analýzu dat v reálném čase a vizualizaci, to může pomoci zejména v oblasti monitorování životního prostředí, inteligentních měst a logistiky.

Dále se SVG mapy budou stále více integrovat do analýzy dat a vylepšených technik vizualizace. Větší platformní podpora bude také velmi žádoucí, SVG by mělo plynule přecházet z desktop zařízení do mobilů nebo hodinek, což bude opět zvyšovat uživatelskou zkušenost. S uživatelskou zkušeností také souvisí přístupnost, která bude kopírovat mezinárodní standardy přístupnosti. SVG mapy mohou také velmi pomoci ve vizualizaci životního prostředí a klimatických dat, což může přispět k celkové udržitelnosti životního prostředí a predikci klimatických změn a environmentálních problémů. SVG mapy budou schopny být součástí tzv. kolaborativního mapování, tzn. že uživatelé budou moci sami zasahovat do map a vyplňovat je aktuálními informacemi, což lze provádět výhradně online a nejčastěji právě u map na webu. (Peterson, 2003)

3.5 Publikování vektorových dat na web

Publikování dat na web je v dnešní době nejdůležitější typ publikace dat. Publikování vektorových dat zahrnuje zpřístupnění samotné mapy ale také geoprostorových informací uživatelům ve webových aplikacích nebo na webových stránkách. Vektorová data jsou v oblasti webové kartografie velmi často využívána, a proto způsobů, jak je publikovat na web je hned několik. Výběr správného způsobu publikace dat se odvíjí v závislosti na konkrétních potřebách a preferencích a závisí na více faktorech jako je velikost souboru, komplexita mapy a složitost tvorby. Níže je uvedeno několik nejčastějších způsobů. (Nétek, 2020)

3.5.1 Přímá publikace do HTML

Základní metodou publikace jednodušších vektorových map na web je vložení vektorových dat přímo do HTML souboru. Výhodou této základní metody je, že vyžaduje pouze základní technologie. Na straně klienta je potřebný pouze webový prohlížeč a na straně serveru webový server. Další metody vyžadují již pokročilejší technologie. (Nétek, 2020)

3.5.2 Webové mapové služby

Často využívanými službami v oblasti Webové kartografie jsou služby webové mapové služby podle standardů Open Geospatial Consortium (OGC). OGC je mezinárodní organizace která zahrnuje většinu vládních, komerčních, nevýdělečných nebo vědeckých organizací za účelem spolupráce při vývoji standardů pro geoprostorová data a služby, které mají za cíl umožňovat sdílení těchto dat a funkcionalit skrz aplikace na webu.

Nejpoužívanějšími službami jsou služby WMS (Web Map Service) a WFS (Web Feature Service). Jedná se o webové služby, které slouží právě pro přenos geoprostorových dat na webu a umožňují mapy sdílet a integrovat do mnoha webových aplikací. Jedna z důležitých vlastností těchto služeb, je tzv. kaskádování. Při kaskádování jedna služba přebírá geoprostorová data od jiné, což umožní kombinování vrstev od různých poskytovatelů a tvorbu nových tematických map. (OGC Standards, 2023)

3.5.2.1 WMS

Web Map Service je služba, která umožňuje sdílení a vyhledávání geoprostorových dat a mapových snímků ze vzdálených serverů na internetu. Vstupní data jsou rastrová nebo vektorová, výstupem jsou primárně obrazová data v různých formátech nejčastěji rastrových, které mohou zobrazovat tematické mapy nebo mapové kompozice. To umožňuje uživatelům zobrazit mapy jako obrázky v různých formátech. WMS je nejčastěji užívána pro vizualizaci statických nebo dynamických map, které umožňují integrovat geoprostorová data napříč různými platformami. (OGC Standards, 2023)

3.5.2.2 WFS

Web Feature Service je standardizovaná služba pro přístup a publikování pouze vektorově orientovaných geoprostorových dat na internetu. Umožňuje uživatelům získávat a manipulovat s geografickými funkcemi, jako jsou body, linie, polygony apod. Výstupem

jsou oproti WMS pouze vektorová data ve formátu GML (Geography Markup Language). WFS velmi usnadňuje výměnu geoprostorových dat mezi různými aplikacemi a systémy. (OGC Standards, 2023)

3.5.3 Webmapping

Webmapping je metoda využívaná pro zobrazování geografických dat s využitím nějaké serverové aplikace. Při klasickém systému client/server stojí na straně serveru ještě jedna aplikace, která se často nazývá mapový server. Mapový server může být naprogramován v jakémkoliv jazyce a je schopný si se serverem předávat informace a vyžádaná data. Často je k tomu využíván CGI rozhraní (Common Gateway Interface), které zajišťuje právě komunikaci mezi webovým serverem a mapovým serverem.

Výsledná aplikace funguje tak, že na straně serveru jsou uložena geografická data a je spuštěn mapový a webový server. Na straně klienta je spuštěn pouze webový prohlížeč, který spustí základní HTML stránku s formulářem a vloženými skripty.

Webmapping se stal součástí spousty odvětví a aplikací, nejčastěji v navigacích, smart cities a monitoringu životního prostředí. (Cajthaml, 2005)

3.5.4 Vektorové dlaždice

Způsob publikování dat pomocí vektorových dlaždic vyvinula již zmíněná společnost Mapbox původně pro svůj software Mapbox Studio. Tento způsob ukládání dat se jmenuje MVT (Mapbox Vector Tiles). Tento formát se postupně rozšířil i do dalších aplikací, je oblíben zejména pro účinné ukládání a renderování vektorových dat.

Tento princip publikování dat pomocí vektorových dlaždic spočívá v zasílání balíčků vektorových souřadnic do klientských zařízení, které se poté vykreslují. Dlaždice pracují podle motivu pyramidy. Výhodou tohoto způsobu je, že lze přetvořit jen část mapy, protože lze přetvořit jen konkrétní dlaždice. Data v dlaždicích jsou od sebe oddělena. Tento přístup také usnadňuje vizuální efekty pro otáčení a zvětšování mapy. (Sterling, 2023)

3.5.5 GeoJSON

Prostřednictvím webových serverů nebo API rozhraní lze přímo poskytovat vektorová data ve formátu GeoJSON (Geographic JavaScript Object Notation), TopoJSON (Topographic JavaScript Object Notation) a dalších. Jedná se většinou o komplexnější formáty oproti SVG, které využívají spíše profesionálové u složitější kartografie, která

zahrnuje všechna prostorová data s komplexní geografickou charakteristikou. Oproti tomu výhodou SVG je jeho snadná rozšiřitelnost, kompatibilita a také jeho uživatelsky přívětivá tvorba a editace.

GeoJSON je formát dat a způsob publikování vektorových dat na web, který slouží pro kódování geografických dat. Jak již název vypovídá, jedná se o formát, který je založen na JSON a využívá se také pro webovou kartografii. Díky kompatibilitě s JavaScriptem se s ním občas také můžeme setkat i u neprofesionálů. Soubory GeoJSON lze obsluhovat přímo z webového severu. Jeho složitější verzí je TopoJSON, což je rozšíření, které optimalizuje ukládání geografických dat pomocí topologie. Snižuje redundanci a je kompaktnější s komplexní geometrií. (González, 2022)

4 Vlastní práce

Praktická část této diplomové práce se zaměřuje na tvorbu vektorových map ve výsledném formátu SVG. Vlastní práce je podložena nastudovanými poznatky z odborné literatury vektorové grafiky a zejména formátu SVG.

Hlavním cílem praktické části práce je vytvořit vektorové mapy v grafické editoru. Tvorba je zaměřena na jednotlivá podlaží fakultních budov ČZU v Praze. Cílem je zmapování všech oficiálních učeben univerzity.

Po vytvoření map v grafickém editoru následuje export map do formátu SVG. Následně je potřeba formát SVG optimalizovat pro co nejvhodnější použití na webu. Po optimalizaci jsou mapy připraveny pro následnou jednoduchou publikaci na webovou stránku.

Dílčím cílem je následně vytvořit webovou stránku pro přehledné zobrazení a prohlížení vytvořených map. Výsledná webová stránka slouží pro možnost hledání jednotlivých učeben a demonstraci vektorových SVG podkladů.

4.1 Sběr podkladů

Prvním krokem, ještě před grafickým návrhem map, bylo nasbírání podkladů pro samotnou tvorbu map. Sběr kvalitních podkladů byl velmi důležitý, protože pouze při pečlivém sjednocení všech zajištěných podkladů lze vytvořit kvalitní vektorové mapy. Způsob, jakým byla konkrétní mapa vytvořena, se odvíjel od podkladu, který se pro danou budovu fakulty a její jednotlivá podlaží podařilo zajistit.

Před samotným sběrem podkladů pro tvorbu map bylo důležité vytyčit a sjednotit cíl této části práce. Jako podklad pro zmapování učeben posloužil nově aktualizovaný seznam učeben v areálu ČZU v Praze z roku 2023, který je dostupný v Průvodci nového studenta 2023 na oficiálních webových stránkách ČZU v Praze. Oficiální seznam učeben lze najít v kapitole 8 – Seznam učeben na ČZU. Průvodce lze najít v digitální podobě nebo jej lze stáhnout v sekci Studium – Informace pro studenty – Průvodce studujícími. Tento seznam je zobrazen na obrázku níže.

Seznam učeben na ČZU										
PROVOZNĚ EKONOMICKÁ FAKULTA	VÝUKOVÝ PAVILON T	FAPEZ-B	B120	TECHNICKÁ FAKULTA	FAKULTA LESNICKÁ A DŘEVÁRSKÁ	FAKULTA ŽIVOTNÍHO PROSTŘEDÍ	FAKULTA TROPICKEHO ZEMĚDĚLSTVÍ	ŘP (PRAMA 5 - MALÁ CHUCHLE)	REKTORÁT	KOLEJ BCD
BUDOVA ZBRATILA	BUDOVA ZBRATILA	FAPEZ-B B121	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA	BUDOVA ZBRATILA
PEF E1	T II	FAPEZ-B B124	BK BKU	FLD L271	F7P Z1	FTZ 113 PC	IVP 1097	R IP	BCD S1	
PEF E11	T T125	FAPEZ-C C107	BK BKU	FLD L352	F7P Z8	FTZ 116	IVP 1098	R S16	BCD S6	
PEF E11	T T125	FAPEZ-C C111	BK BKU	FLD L366	F7P Z017	FTZ 115	IVP 1099	R S18		
PEF E1V	T T137	FAPEZ-C C112	TF MI	FLD L372	F7P Z019	FTZ 116	IVP 1095	R S19		
PEF E1V	T (bezoznat)	FAPEZ-C C118	TF MIH	FLD L201	F7P Z028	FTZ 122	IVP 1096	R Testovací centrum		KOLEJ EFG
PEF E119	T (bezoznat)	FAPEZ-C C121	TF MIH	FLD L301	F7P Z105	FTZ 123	IVP 1097			BUDOVA ZBRATILA
PEF E155					F7P Z107	FTZ 124	IVP 1098			BCD S1
PEF E225					F7P Z108	FTZ 125	IVP 1099			BCD S6
PEF E329					F7P Z109	FTZ 126	IVP 1095			
PEF E315					F7P Z110	FTZ 127	IVP 1096			
PEF E323					F7P Z225	FTZ 128	IVP 1097			
PEF E333					F7P Z229		IVP 1098			
PEF E429					F7P Z328		IVP 1099			
PEF E423					F7P Z330		IVP 1095			
PEF E431							IVP 1096			
PEF E439							IVP 1097			
PEF E435							IVP 1098			
PEF C11							IVP 1099			
PEF C12							IVP 1095			
PEF C13							IVP 1096			
PEF C14							IVP 1097			
PEF C21							IVP 1098			
PEF C22							IVP 1099			
PEF C24							IVP 1095			
PEF C27							IVP 1096			
PEF CEMS2							IVP 1097			
BUDOVA ZBRATILA							IVP 1098			
PEF-C C15							IVP 1099			
PEF-C C16							IVP 1095			
PEF-C C17							IVP 1096			
PEF-C C18							IVP 1097			
PEF-C C26							IVP 1098			
PEF-C C28							IVP 1099			
PEF-D D1							IVP 1095			
PEF-D D11							IVP 1096			
PEF-D D107							IVP 1097			
PEF-D D108							IVP 1098			
PEF-D D106							IVP 1099			
PEF-D D122							IVP 1095			
PEF-D D126							IVP 1096			
PEF-D D332							IVP 1097			
PEF-D D476							IVP 1098			
PEF-D D432							IVP 1099			
PEF-D D438							IVP 1095			
PEF-D D444							IVP 1096			

Obrázek 18 – Seznam učeben na ČZU (Průvodce nového studenta, 2023)

Tento seznam byl také nápomocný pro definici jednotlivých budov či částí budov a jeho rozdělení, které před samotnou tvorbou bylo velmi důležité. ČZU v Praze má celkem šest fakult. Počet fakult, ale neodpovídá přesně počtu budov všech učeben v areálu. Proto je důležité si určit a rozdělit i samotné budovy.

Na počet studentů je nejpočetnější fakultou Provozně ekonomická fakulta (PEF). Skládá se ze čtyř budov. Nejstarší je budova E, později vystavěné budovy CEMS a D a dále také budova T. Vzhledem k propojení budov E, CEMS a D lze konstatovat, že fakulta disponuje touto hlavní budovou a jedinou oddělenou budovou T.

Technická fakulta (TF) disponuje celkem čtyřmi pavilony a také pod ní spadá kruhová hala. Pavilony 1, 2 a 3 jsou propojeny v jednu větší budovu, tudíž lze konstatovat, že tyto tři pavilony tvoří hlavní budovu TF. Oddělenou budovou je pavilon 4, který obsahuje zejména dílny a třetí budovou je zmíněná Kruhová hala.

Fakulta agrobiologie, přírodních a potravinových zdrojů (FAPPZ) se nachází na největší ploše areálu, celkem se rozprostírá ve čtyřech budovách. Nejstarší budovou je budova A, hned vedle se nachází budova B a budova C. V zadní části se nachází jedna

z největších budov v areálu zvaná Mezifakultní centrum environmentálních věd (MCEV) neboli budova D. Tuto budovu FAPPZ společně sdílí také s Fakultou životního prostředí.

Fakulta životního prostředí (FŽP) disponuje ovšem také svojí vlastní budovou, která je napojena právě na zmíněnou budovu D. Komplex se někdy nazývá právě MCEV budovy I a II.

Fakulta lesnická a dřevařská (FLD) sídlí celkem ve třech budovách. V nejstarší hlavní budově, v dřevařském pavilonu a nejnovější budově High-tech technologickém pavilonu.

Poslední fakultou je Fakulta tropického zemědělství (FTZ). FTZ je nejmenší fakulta ČZU a sídlí v jedné, nejnovější budově na ČZU.

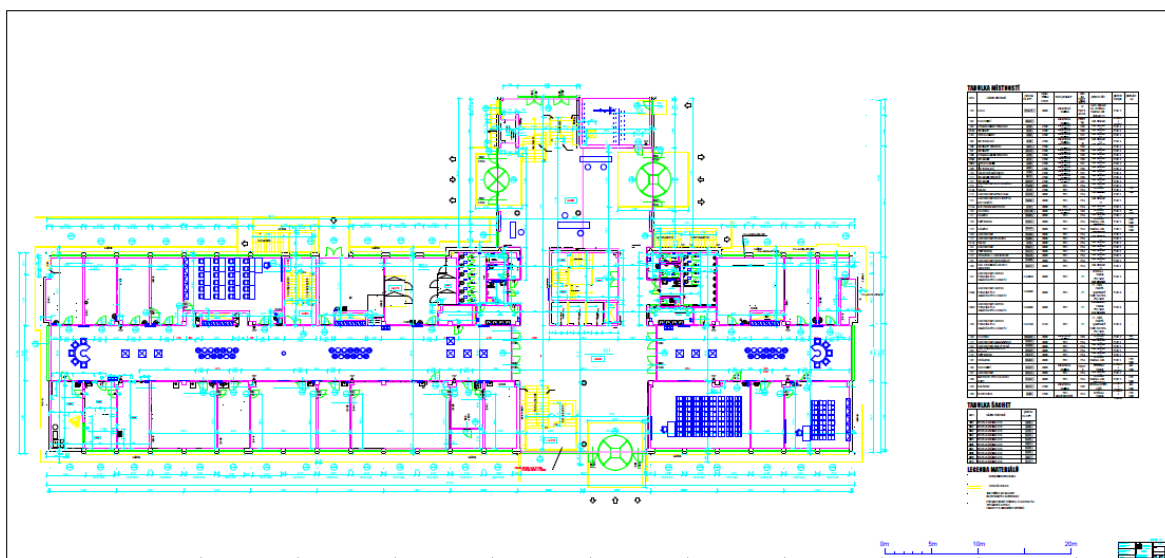
Pro tvorbu je tedy definován seznam budov učeben ČZU:

- PEF (4 podlaží)
- PEF_T (2 podlaží)
- TF (4 podlaží)
- TF_4 (1 podlaží)
- TF_RKU (1 podlaží)
- FAPPZ_A (6 podlaží)
- FAPPZ_B (4 podlaží)
- FAPPZ_C (2 podlaží)
- MCEV – FAPPZ_D + FŽP_D (6 podlaží)
- FŽP (4 podlaží)
- FLD (4 podlaží)
- FLD_DP (4 podlaží)
- FLD_HT (2 podlaží)
- FTZ (4 podlaží)

Po prvotním rozdělení budov bylo potřeba vyhledat vyměřené plány těchto budov, pro možnost vytvoření přesných map. Hlavním podkladem, který se podařilo zajistit byly stavební půdorysy budov. Jelikož byla většina areálu ČZU postavena nebo rekonstruována postupně v průběhu cca posledních dvaceti let, tak se některé tyto půdorysy budov liší kvalitou a způsobem zpracování. Proto bylo potřeba tyto materiály důkladně projít a zkompletovat. Postupně bylo potřeba soubory převést do editovatelného formátu, v kterém

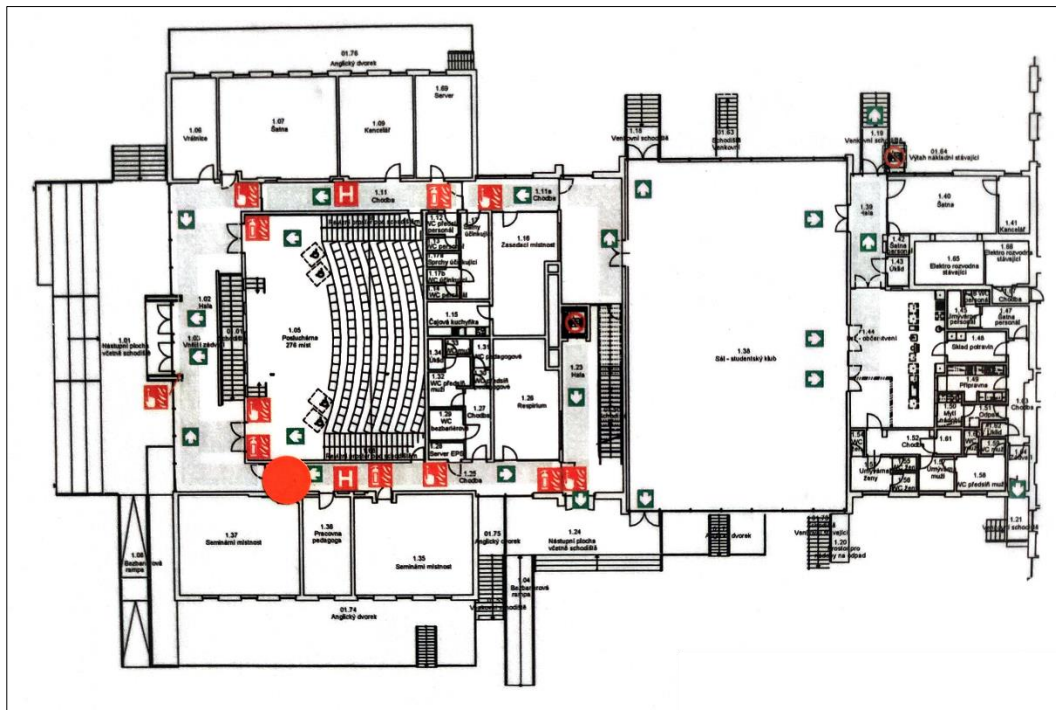
bylo potřeba upravit vrstvy jednotlivých podkladových půdorysů nebo některé vrstvy zcela dodělat.

Konkrétně byly tyto soubory k dispozici ve formátu DWG. DWG je nativní formát dvou a třírozměrné grafiky vytvořený pro CAD (Computer-Aided Design) softwary. Předpokladem tedy je že půdorysy byly vytvořeny v neznámějším softwaru AutoCAD, využívaném právě ve stavebnictví pro návrhy půdorysů budov apod. Druhým nalezeným formátem souborů byl univerzální PDF, zde obtížnost zpracování záležela na tom, zda byl soubor editovatelný ve vektorovém editoru či ne. Všechny tyto formáty naštěstí měly základ vektorové grafiky, tudíž byla možnost zpracovat jednotlivé křivky a polygony. Příklad podkladu půdorysu podlaží budovy MCEV ve formátu DWG lze vidět na obrázku níže.



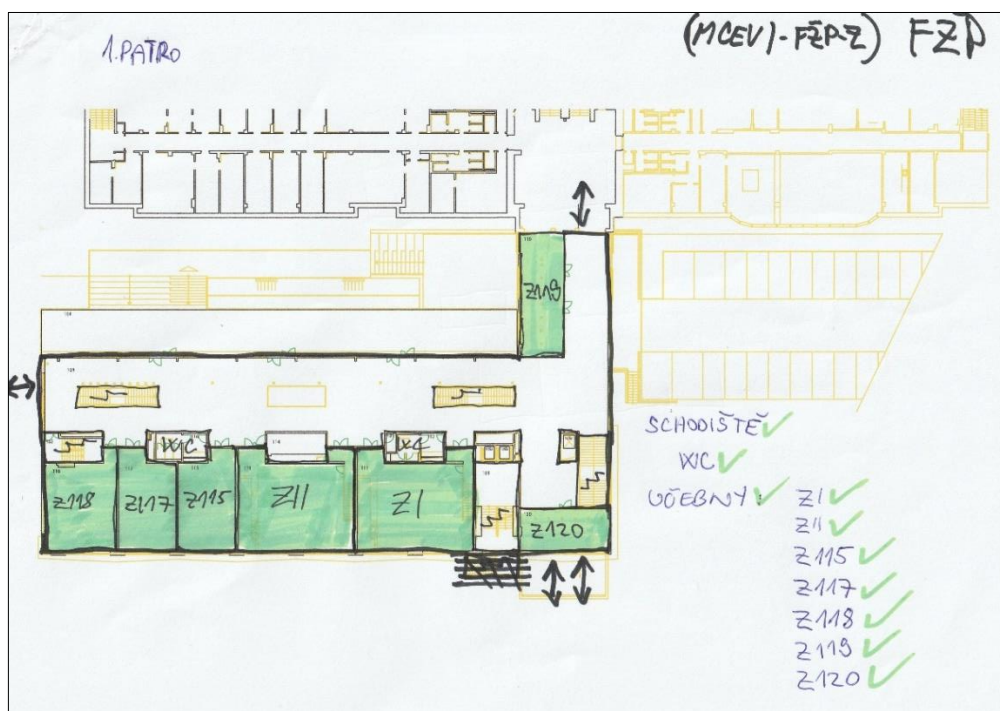
Obrázek 19 – Půdorys prvního podlaží budovy MCEV ve formátu DWG (INDESS, s.r.o., 2015)

Stavební půdorysy se nepodařilo zajistit u všech budov areálu ČZU, a tak bylo potřeba najít jinou alternativu pro vytvoření map učeben. Nakonec jako podklad v těchto případech byly využity požární evakuační plány budov, na základě kterých lze ve stejném měřítku také vytvořit mapy jednotlivých podlaží. Konkrétně se jednalo o tři budovy, největší budovu Fakulty tropického zemědělství, pavilon T dnes již Provozně ekonomické fakulty a rekonstruovanou hlavní budovu Fakulty lesnické a dřevařské. Bohužel k těmto budovám nebyly zajištěny vektorové podklady, ale pouze rastrové. Proto bylo potřeba tyto rastrové obrazy přepracovat na vektorové. Příklad tohoto rastrového podkladu podlaží budovy PEF_T lze vidět níže.



Obrázek 20 – Evakuační plán prvního podlaží pavilonu T Provozně ekonomické fakulty
(Odbor bezpečnosti ČZU v Praze, 2013)

Po zpracování digitálních podkladů nastala důležitá část přímo v areálu. Vzhledem k tomu, že většina budov měla tyto materiály půdorysů od prvopočátku stavby, spousta budov se mohla lehce měnit či v malých částech rekonstruovat. Nejen kvůli těmto změnám ale hlavně kvůli účelu této práce bylo potřeba rozvržení učeben v areálu kontrolovat fyzicky přímo na místě. Nastala tedy poslední fáze sběru podkladů a to fyzické procházení zmíněných budov a zápis učeben a interiérového zařízení do podkladových půdorysů, což výrazně dopomohlo k následné tvorbě vektorových map. Na obrázku níže lze vidět příklad upraveného půdorysu s vlastnoručními zápisky.



Obrázek 21 – Upravený půdorys 1. patra Fakulty životního prostředí s vlastnoručními zápisky

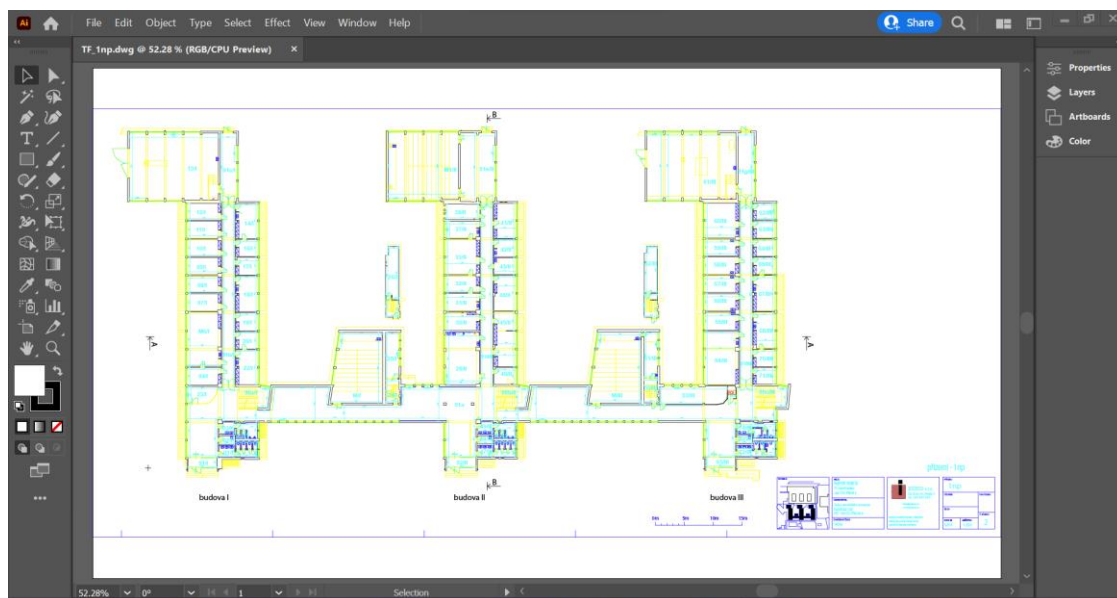
4.2 Grafická tvorba map

Po sběru grafických podkladů byl připraven základ pro samotnou grafickou tvorbu map. Jak již bylo zmíněno, pro vytvoření každého podlaží konkrétní budovy bylo potřeba nejprve zpracovat půdorys daného podlaží z původního formátu DWG do editovatelné podoby.

Jako vhodný grafický editor byl vybrán software Adobe Illustrator. Adobe Illustrator je současně nejvhodnějším profesionálním editorem pro tvorbu vektorové grafiky a také formátu SVG. Nabízí všechny funkce potřebné k tvorbě vektorových map a také byl velmi vhodný z hlediska jeho kompatibility s mnoha formáty souborů. Pro tuto práci bylo důležité, že Illustrator umožňuje editovat soubory DWG a PDF a také dokáže importovat do návrhové plochy rastrovou grafiku. Samozřejmě byla také klíčová možnost exportu finálního grafického návrhu do formátu SVG v mnoha verzích.

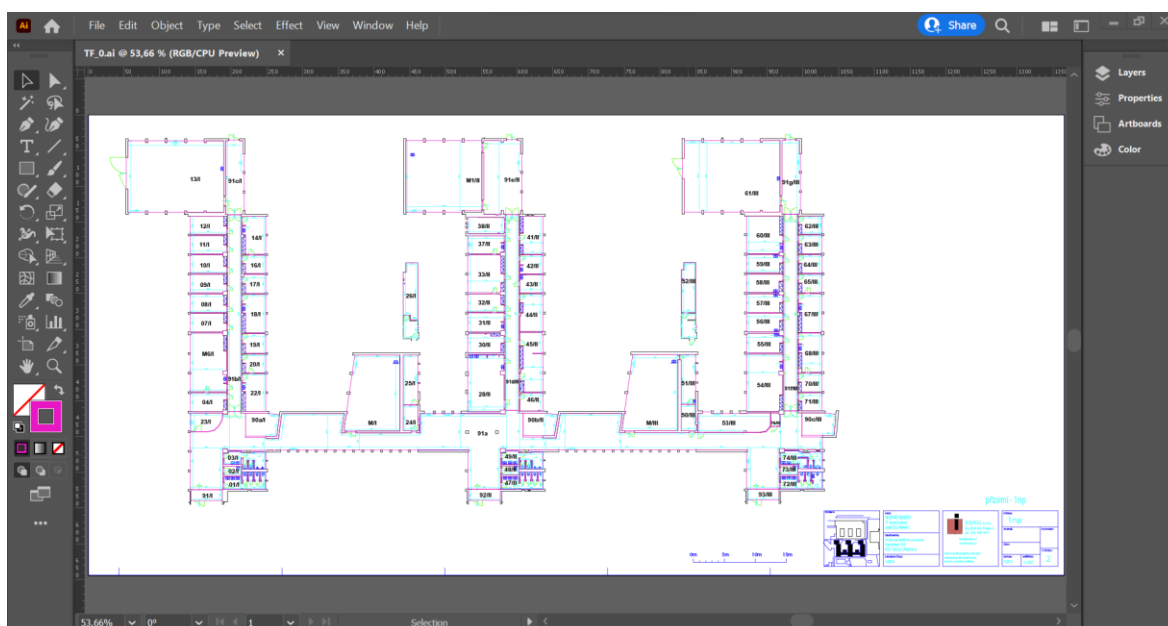
Proces samotné tvorby map spočíval v několika postupných fázích. Celá tvorba v jednotlivých stádiích úpravy je popsána a vyobrazena níže. Jako ukázka zde slouží podlaží Technické fakulty.

Prvotním podkladem je půdorys budovy ve formátu DWG vytvořen v původním softwaru AutoCAD.



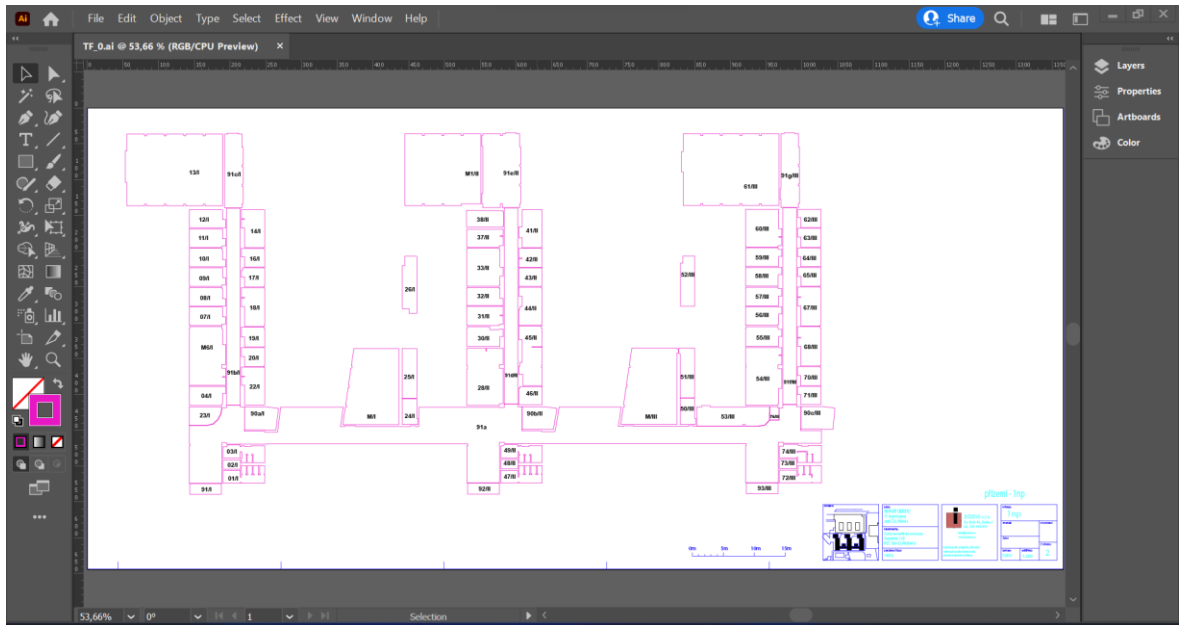
Obrázek 22 – Tvorba mapy v Illustratoru (půdorys přízemí Technické fakulty ve formátu DWG)

Nejprve bylo zapotřebí konvertovat daný soubor DWG do vektorového formátu AI, pro snadnou úpravu. AI je nativní formát souborů tvořených v Adobe Illustrator. První provedená úprava již v tomto formátu, která umožnila úpravu jednotlivých vrstev je zobrazena na obrázku níže.



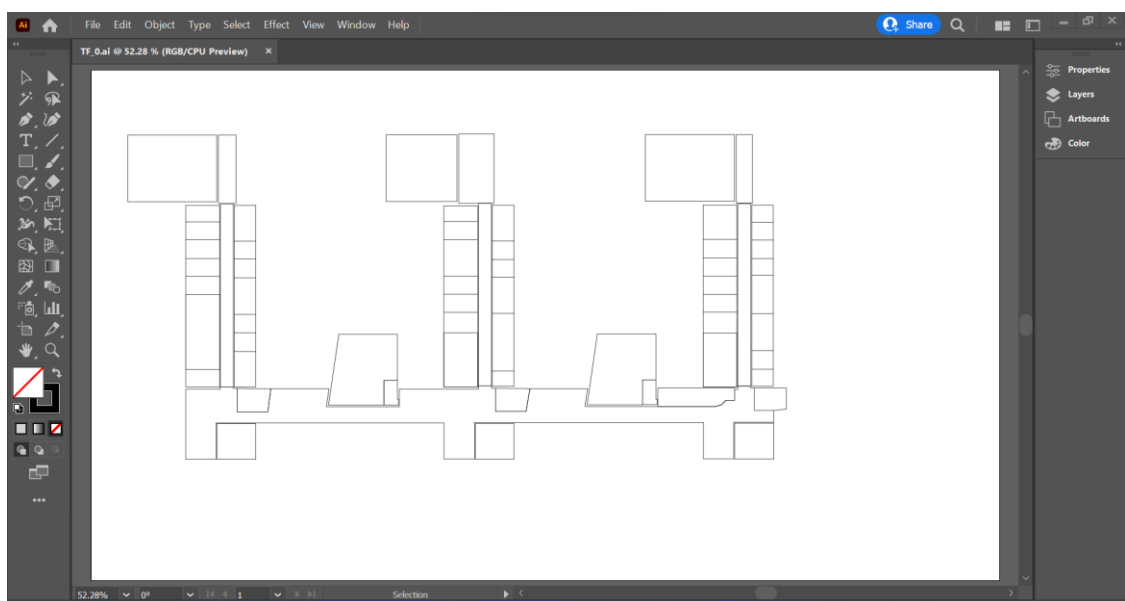
Obrázek 23 – Tvorba mapy v Illustratoru (převod a úpravy do formátu AI)

Dále bylo důležité provést komplexní úpravu jednotlivých vrstev při které došlo k redukci nepotřebných objektů, seskupení objektů, ořezu apod. Výsledná podoba je zobrazena na obrázku níže.



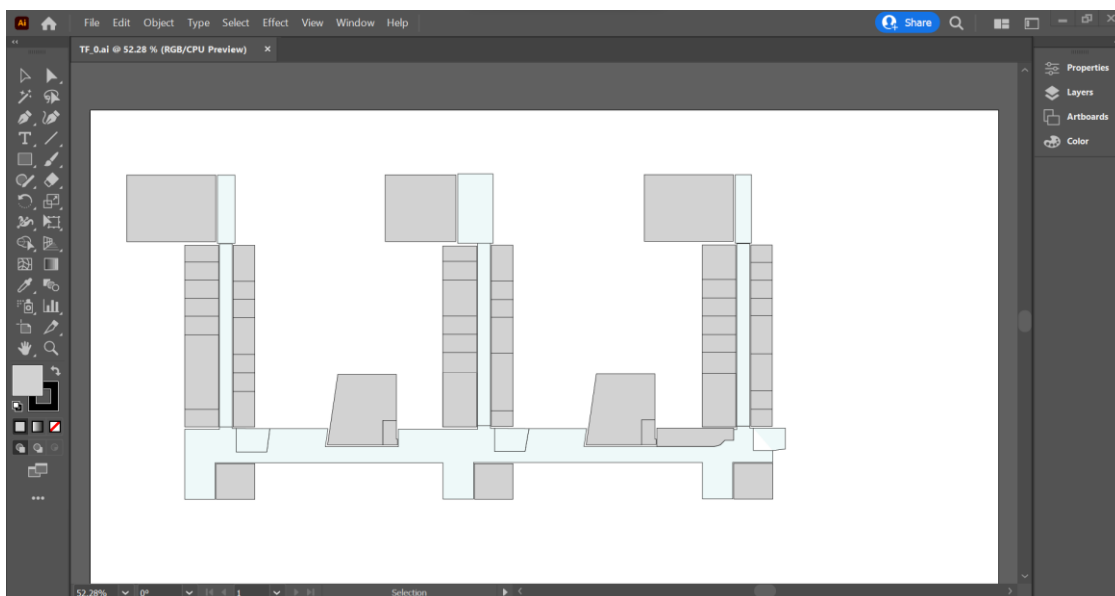
Obrázek 24 – Tvorba mapy v Illustratoru (redukce nepotřebných objektů, seskupení objektů, ořez)

Následný obrázek zobrazuje úpravy vedoucí k čistému podkladu podlaží. Zde bylo zapotřebí upravit vhodně již existující křivky nebo vytvořit pomocí křivek a polygonů nové objekty.



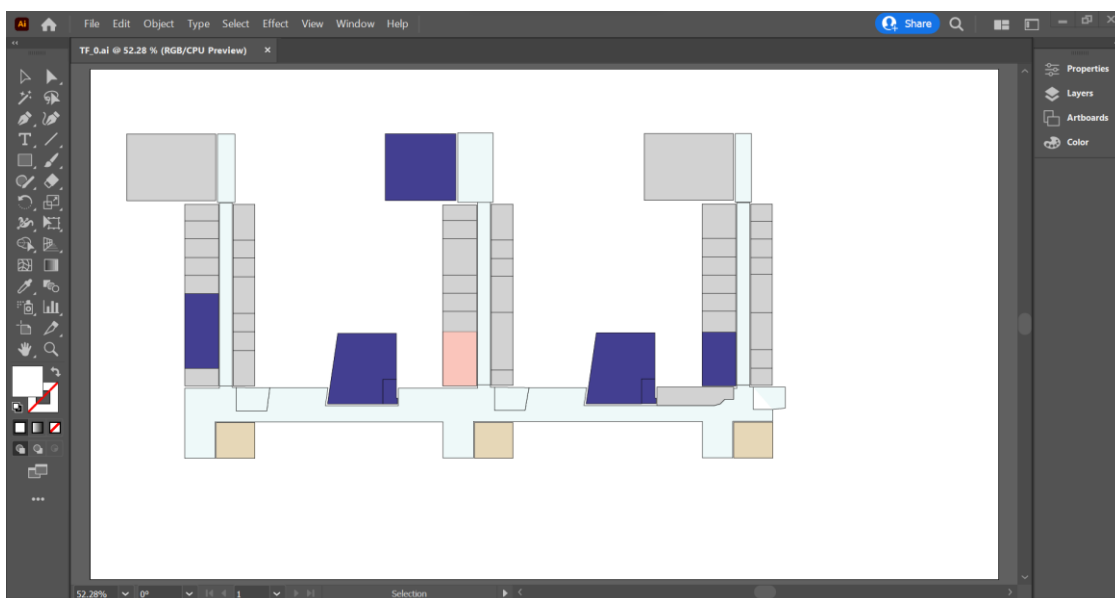
Obrázek 25 – Tvorba mapy v Illustratoru (tvorba a úprava křivek)

Následně byli barevně rozlišeny místnosti a všechny chodby a průchody v budovách, což zobrazuje obrázek níže.



Obrázek 26 – Tvorba mapy v Illustratoru (úprava barev)

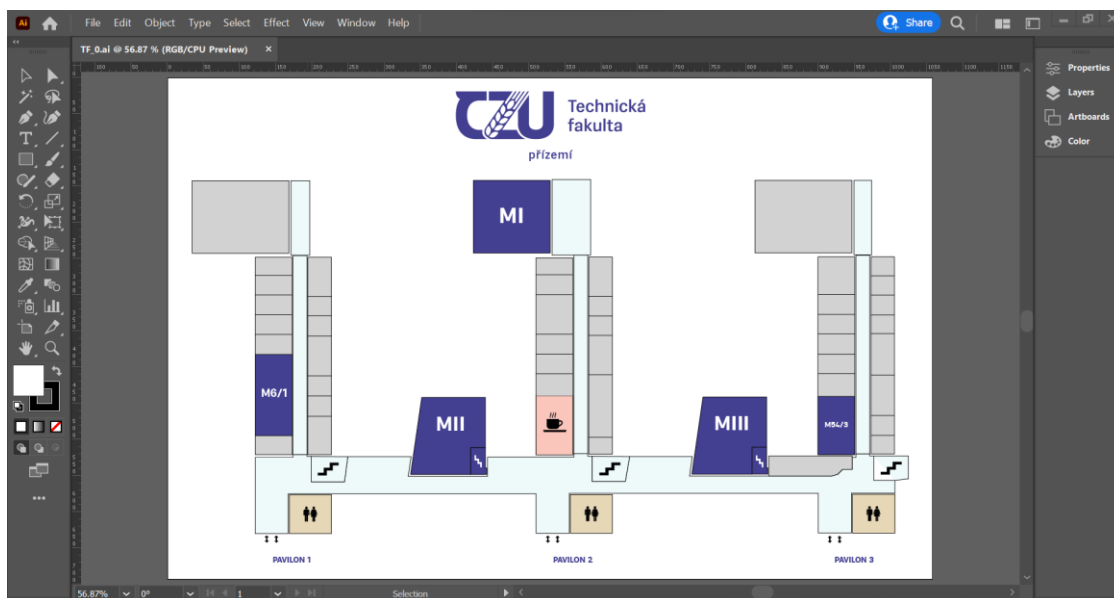
Předposlední fází bylo vytvoření místností a zanesení učeben do mapy. Vždy byly učebny vytvořeny v barvě dané fakulty. Také bylo potřeba vyznačit jednotnými barvami ostatní technické zařízení. Barevnou úpravu zobrazuje obrázek níže.



Obrázek 27 – Tvorba mapy v Illustratoru (tvorba místností)

K vytvoření finální mapy bylo ještě zapotřebí doplnit označení učeben a technického zařízení (značky schodů, průchodů, toalet, stravovacích zařízení apod.). Mapa byla poté

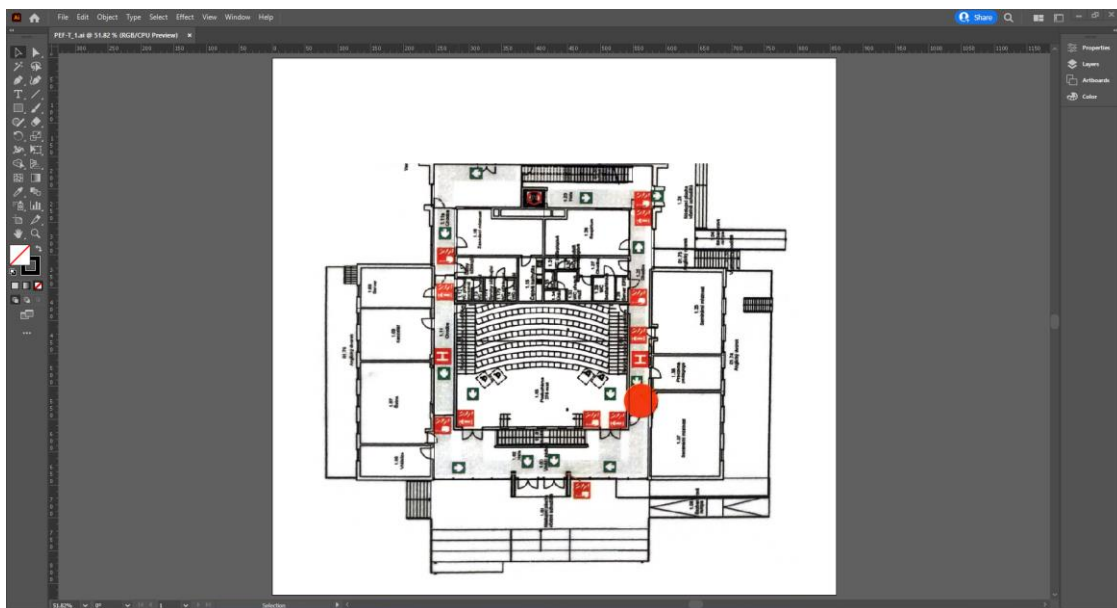
designována do barvy dané fakulty s jejím logem a popisem. Obrázek níže zobrazuje finální vektorový návrh přízemí Technické fakulty.



Obrázek 28 – Tvorba mapy v Illustratoru (finální mapa přízemí Technické fakulty)

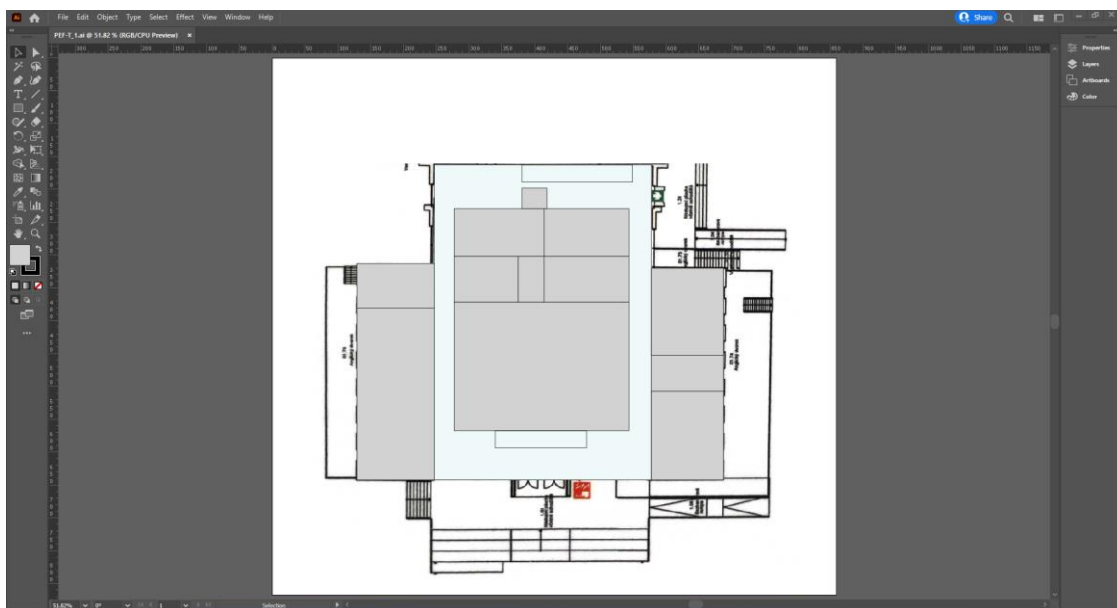
Pro mapy tvořené z rastrových podkladů požárních evakuačních plánů budov byl prvotní proces lehce odlišný. V tomto případě bylo nejprve zapotřebí převést rastrový obraz do digitální podoby a posléze soubor importovat do Adobe Illustrator. Jak bylo zmíněno v kapitole o Adobe Illustratoru, ačkoliv se jedná o vektorový grafický editor, umí také zobrazovat a lehce pracovat s rastrovými podklady, což bylo v tomto případě velice důležité. Rastrový podklad sloužil jako vzor, nad kterým bylo možné od nuly vytvářet vhodné křivky a objekty vedoucí k vytvoření finální mapy. Vzhledem k možnosti přesné tvorby vektorových přímek byla zachována velikost, rozměry a měřítka místností. Celý proces tvorby v jednotlivých stádiích úpravy je popsán a vyobrazen níže.

Nejprve bylo zapotřebí upravit rastrový podklad, tak aby bylo možné podle něj vytvářet vektorové přímky a objekty. Poté byl vytvořen základní obrys budovy a jednotlivých chodeb. Tento proces lze vidět na obrázku níže.



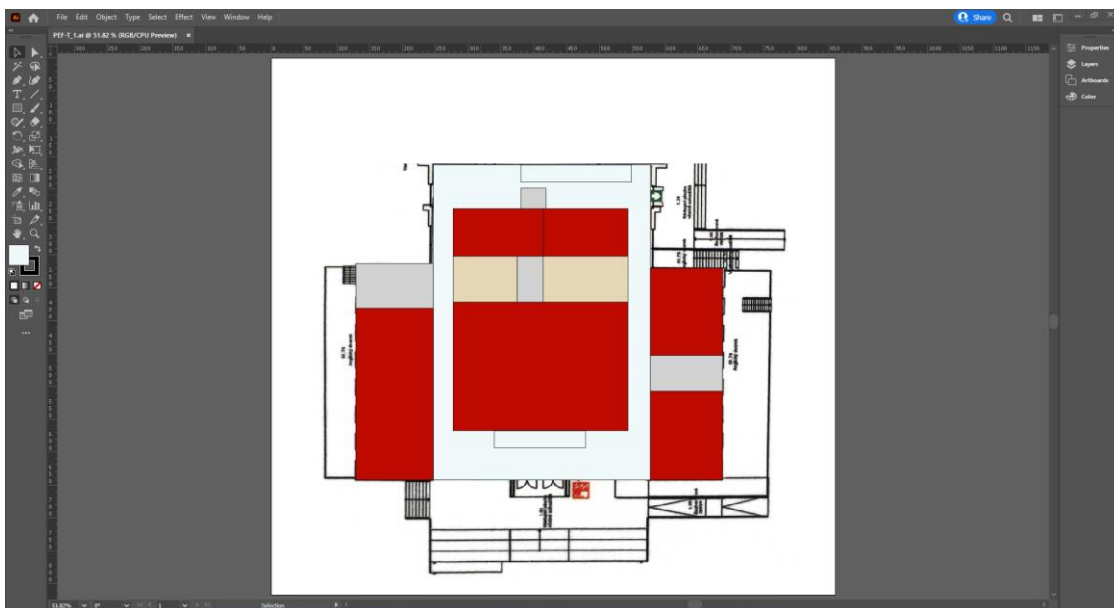
Obrázek 29 – Tvorba mapy v Illustratoru (úprava rastrového podkladu a tvorba obrysu)

Dalším krokem bylo vytvoření jednotlivých místností, průchodů a chodeb a jejich barevné rozlišení, což zobrazuje obrázek níže.



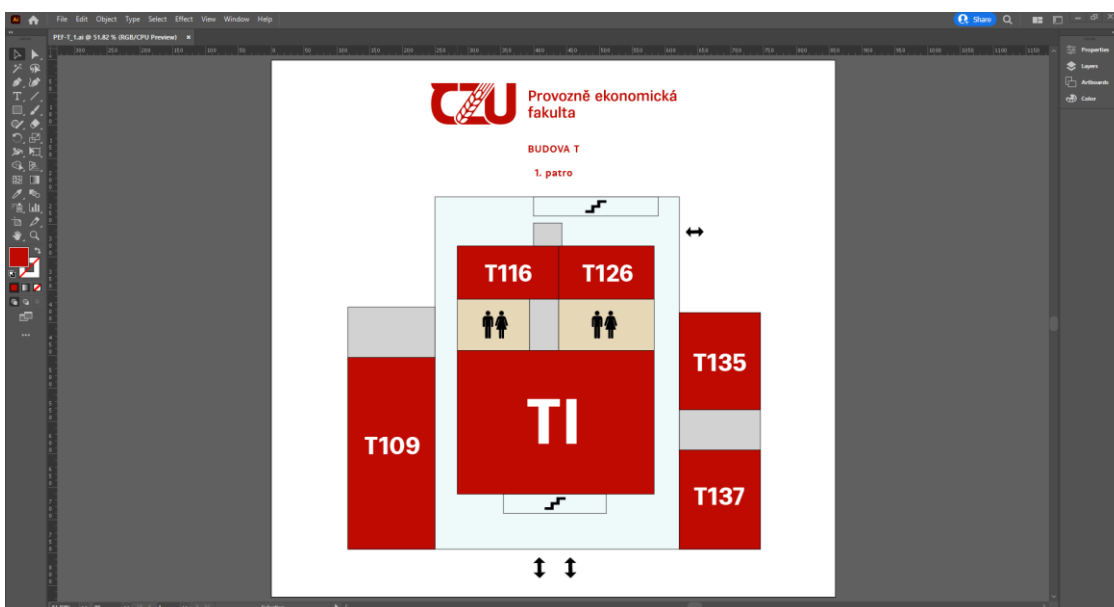
Obrázek 30 – Tvorba mapy v Illustratoru (úprava barev)

Předposlední fází bylo zanesení učeben a technického zařízení do mapy. Barevnou úpravu zobrazuje obrázek níže.



Obrázek 31 – Tvorba mapy v Illustratoru (tvorba místností)

Finální mapa Provozně ekonomické fakulty budovy T s vyznačenými učebnami a technickým zařízením je na obrázku níže.



Obrázek 32 – Tvorba mapy v Illustratoru (finální mapa 1. patra budovy T Provozně ekonomické fakulty)

Finální mapy jsou zachovány ve stejných rozměrech a měřítku jako jsou podklady, tudíž kopírují reálné velikosti učeben. Po vytvoření finálních map bylo potřeba ještě provést dvě poslední grafické úpravy. Prvně bylo nutné odstranit spoustu nežádoucích a nevyužitých

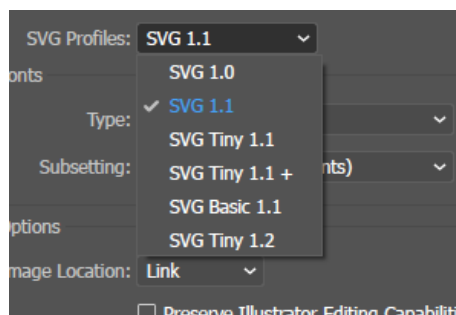
pozůstalých vrstev a objektů. To bylo velmi důležité pro zmenšení velikosti souboru, před odstraněním nevyužitých objektů totiž byla velikost souboru přibližně 3000 kB.

Grafický návrh se nyní téměř celý skládá z vektorových křivek a objektů. Posledním problémem jsou prvky ve formě textu, konkrétně jsou to názvy učeben a názvy budov. Tento problém je potřeba vyřešit pro pozdější vkládání SVG souboru do HTML stránky. Vzhledem k odlišné kompatibilitě různých webových prohlížečů není vždy jisté, že každý prohlížeč bude vždy podporovat zvolený oficiální font ČZU v Praze. Proto je důležité prvky textu přeměnit také na objekty. K tomu v Illustratoru slouží výchozí nástroj zvaný „create outlines“. Tento nástroj vytvoří přesné obrysy jednotlivých glyfů neboli znaků v podobě vektorových křivek. Tyto texty se poté budou zobrazovat ve formě křivek a objektů a budou vždy vykresleny stejně neohledně na zvolený prohlížeč.

4.3 Export map do SVG

Po vytvoření finálního vektorového návrhu přišla na řadu poslední část tvorby, převod a export grafiky do formátu SVG pro snadnou publikaci na webu. Proces transformace grafiky v Illustratoru z editovatelného formátu AI do formátu SVG probíhá automaticky. Illustrator nabízí jednoduchou konfiguraci pro vygenerování SVG kódu daného vektorového obrazu a uložení souboru ve formátu SVG.

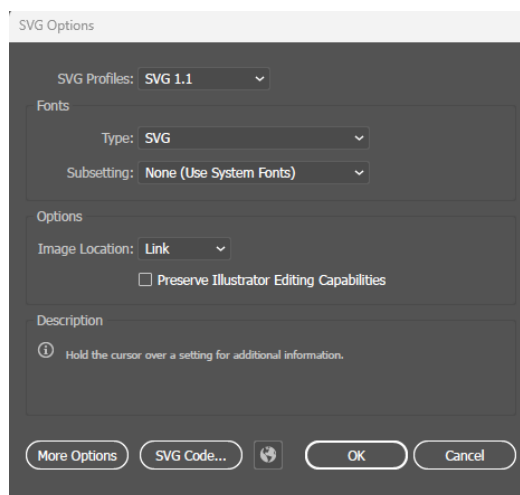
Při exportu Illustrator nabízí nastavení všech důležitých vlastností budoucího SVG souboru. V dialogovém okně lze nastavit profil SVG nebo verzi SVG, ve které lze soubor exportovat. Jak již bylo zmíněné SVG se stále vyvíjí, a proto se s ním lze setkat v různých verzích. Tato verze Adobe Illustratoru z roku 2022 nabízí export do různých verzí první generace SVG. Verze SVG dostupné v Illustratoru lze vidět na obrázku níže.



Obrázek 33 – Verze SVG v Illustratoru

Dále lze v možnostech SVG nastavit vlastnosti fontu, konkrétně jeho typ a nastavení, jaký font se v případě absence fontu v prohlížeči má využít. V případě vloženého externího obrazu v souboru lze nastavit, jakým způsobem bude obrázek odkazován. Ve spodní části

Lze vidět popis všech vlastností tohoto nastavení. Ve spodní části se nachází tlačítko SVG Code, pomocí kterého lze otevřít okno s textovým editorem, ve kterém lze samotný SVG kód ještě dopravit ručně. Všechny možnosti nastavení lze vidět na obrázku níže.



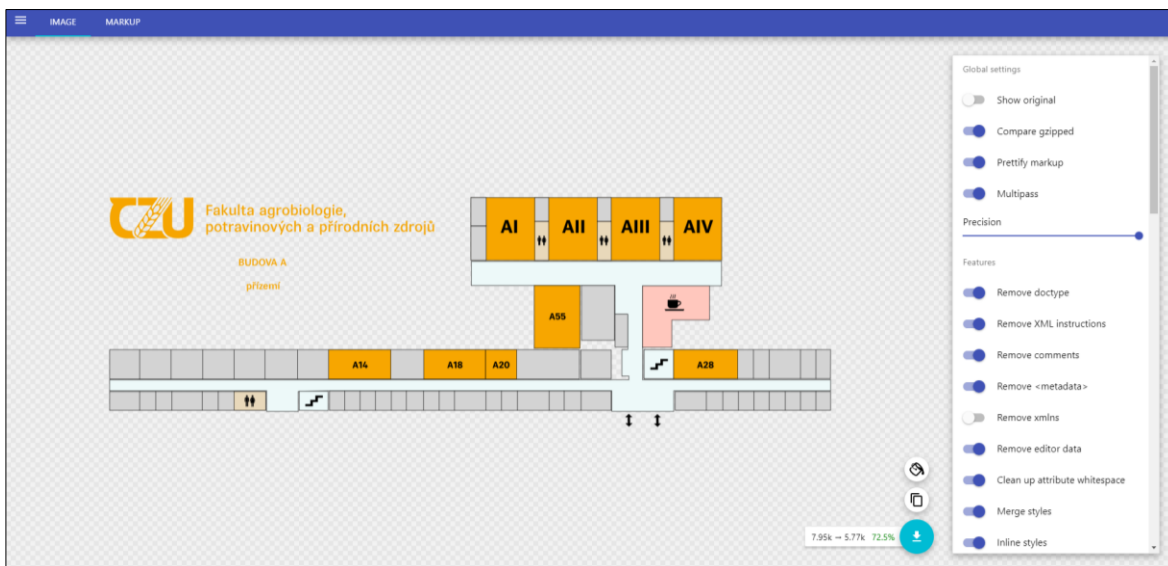
Obrázek 34 – Volba vlastností SVG souboru v Illustratoru

V tomto případě byla využita verze SVG 1.1 vzhledem k její největší kompatibilitě většiny současných prohlížečů. Nastavení fontu nebylo nijak měněno, vzhledem k již zmíněné konverzi textů na objekty je tento problém již vyřešen. Rastrové podklady jsou již ze souboru vymazány, tudíž u lokace obrázku také není potřeba nic měnit. Poslední úpravu samotného SVG kódu nebylo potřeba provádět ručně, ale byl na ni využit vhodný nástroj.

4.4 Optimalizace SVG

Poslední krok v procesu tvorby SVG map spočívá v optimalizaci SVG a finálních úpravách SVG kódu. Pro úpravu kódu není potřeba manuální přístup v textovém editoru. Existuje mnoho online nástrojů, které automaticky upraví SVG kód na jeho nejjednodušší podobu.

Zde je využit bezplatný online nástroj [svgomg.net](https://www.svgomg.net). Jedná se o nástroj pro optimalizaci a minimalizaci SVG. Po nahrání souboru SVG lze nastavit jaké všechny funkce je žádoucí provést. V tomto případě je vybrána většina. Po optimalizaci SVG souboru je vygenerována finální SVG mapa. V tomto případě byla datová komplexita souboru zmenšena z 32 KB na finální 21 KB. Nástroj [svgomg.net](https://www.svgomg.net) lze vidět na obrázku níže.



Obrázek 35 – Online nástroj pro optimalizaci SVG kódu (SVGOMG, 2023)

Jedním z dílčích cílů tvorby SVG a následné optimalizace SVG bylo snížit datovou velikost výsledného souboru na co nejmenší možnou. Toho bylo potřeba dosáhnout pro nejlepší využití map právě ve formátu SVG. Cílem je využití map na webu, při nahrávání map na web je často znát jejich datová náročnost. Celkový proces nejprve manuálního odstranění nepotřebných vrstev a objektů a následné využití nástroje pro finální optimalizaci SVG souboru nám pomohl zmenšit soubor z přibližně 3000 kB na pouhých 21 kB. Tato výsledná malá velikost souborů okolo 20 kB bude velice žádoucí při následné tvorbě webové HTML stránky.

Výsledkem celé grafické práce je zmapování čtrnácti budov a čtyřiceti osmi vytvořených map ve formátu SVG. Součet datové velikosti všech map je 824 kB a průměrná velikost mapy je přibližně 17 kB. Ukázkou map každé fakulty lze najít v tištěné příloze v této práci. Soubor se všemi mapami je dostupný jako digitální příloha této práce a je také k dispozici ke stažení na webové stránce.

4.5 Návrh webu

Po vytvoření vektorových map následuje poslední část praktické části práce, a to návrh webové prezentace těchto map. Vektorové mapy vyobrazují všechna podlaží fakultních budov a jejich učebny. Pro lepší zobrazení a demonstraci využití vlastností právě formátu SVG, ve kterém jsou mapy vytvořeny, jsou tyto mapy vloženy na webovou stránku.

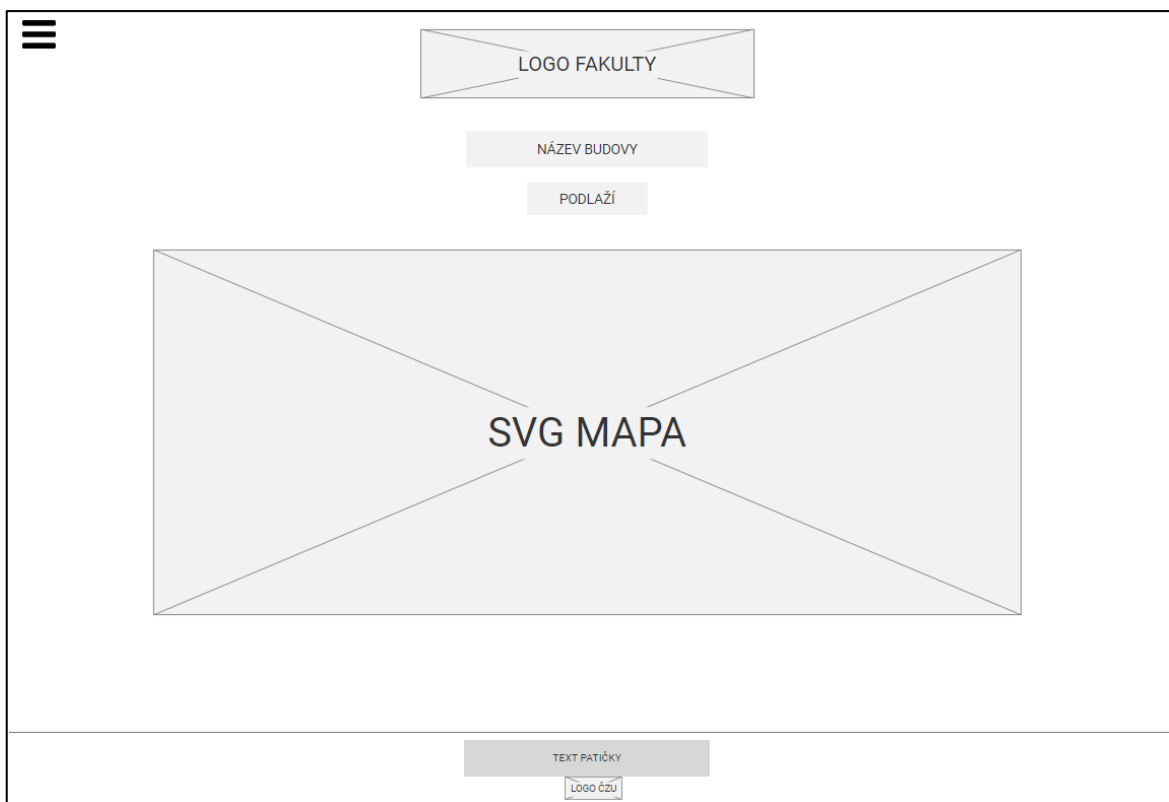
Jelikož návrh samotné webové stránky nebyl hlavním cílem práce, ale pouze dílčím, proces tvorby webu je k tomu adekvátně přizpůsoben. Pro demonstraci vytvořených

vektorových map je vytvořena jednostránková HTML stránka, na které lze přepínat náhled mezi jednotlivými budovami a jejich podlažními. Proces se skládá z vytvoření jednoduchého wireframu, následné tvorby HTML kódu, zřízení webhostingu a domény na webhostingové službě, a nakonec nahrání samotné stránky na web.

4.5.1 Wireframe

Před samotnou tvorbou HTML stránky byl vytvořen velmi jednoduchý návrh rozložení prvků. Ten byl vytvořen pomocí wireframu neboli drátového modelu. Vzhledem k tomu, že se jedná o jednostránkovou HTML stránku byl vytvořen pouze jeden wireframe pro základní rozložení prvků, s ohledem na možnosti tvorby v samotném HTML.

Pro tvorbu byl využit software Axure RP, což je softwarový nástroj právě pro tvorbu náhledu těchto řešení. Wireframe webové stránky lze vidět na obrázku níže. Je použit jak pro desktop zařízení, tak pro mobilní zařízení.



Obrázek 36 – Wireframe webové stránky

4.5.2 Tvorba HTML

Struktura webové stránky byla napsána pomocí značkovacího jazyka HTML5, design stránky pomocí CSS a interaktivní prvky pomocí JavaScriptu. Jako textový nástroj byl zvolen

software Microsoft Visual Studio Code, který je intuitivní a pomocí nápověd a automatických korekcí dopomáhá uživateli při tvorbě kódu.

Samotné tělo HTML kódu se skládá z jednotlivých částí. Obsahuje menu, úvod, element *object* pro načtení map, interaktivní JavaScriptovou část a patičku. Menu je vytvořeno pomocí klasických elementů seznamu *ul* a *li*. Interaktivita a funkčnost menu je vytvořena pomocí vloženého kódu v JavaScriptu. Je vytvořen tzv. toggle, pomocí kterého lze intuitivně přepínat jednotlivé mapy.

Vektorové mapy jsou vloženy do HTML stránky pomocí elementu *object*, který slouží pro integraci externího souboru do HTML bez jakýchkoliv omezení. To je zde žádoucí, protože vektorový obraz se poté vykreslí se všemi jeho vlastnostmi. Zobrazená mapa lze přibližovat i oddalovat bez ztráty kvality.

Dalšími částmi v těle dokumentu je úvodní stránka, která se zobrazí automaticky při načtení stránky a také patička celého dokumentu, která je zobrazena v dolní části stránky. Výsledkem je jeden HTML soubor a k tomu napojený CSS soubor.

Celý web se skládá pouze z jednoho HTML indexu. Webová stránka při prvotním otevření načte kód HTML a následně, při přepínání jednotlivých budov a podlaží, se pomocí odkazu načítají jednotlivé SVG soubory (vektorové mapy).

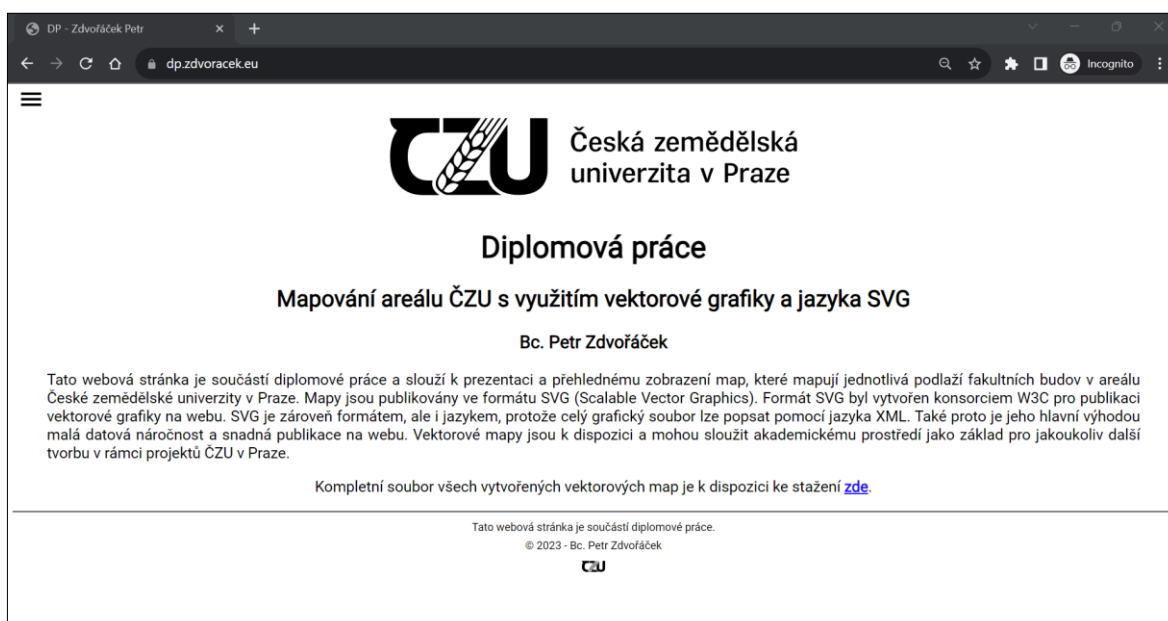
Jednou z výhod map vytvořených ve formátu SVG je rychlost jejich načtení na webu, což funguje zejména díky jejich menší datové velikosti. Demonstraci důležitosti datové velikosti map lze uvést u komparace s formátem JPEG (Joint Photographic Experts Group). Při exportu s maximálním rozlišením mapy ve formátu rastrové grafiky JPEG je datová velikost jedné mapy okolo 600 kB, zatím co stejná mapa ve formátu SVG má velikost 20 kB, nemluvě o tom, že mapa ve formátu JPEG nedisponuje vlastnostmi jako je nezávislost na rozlišení a další vlastnosti vektorové grafiky. Finální HTML kód stránky lze najít v příloze této diplomové práce.

4.5.3 Finální webová stránka

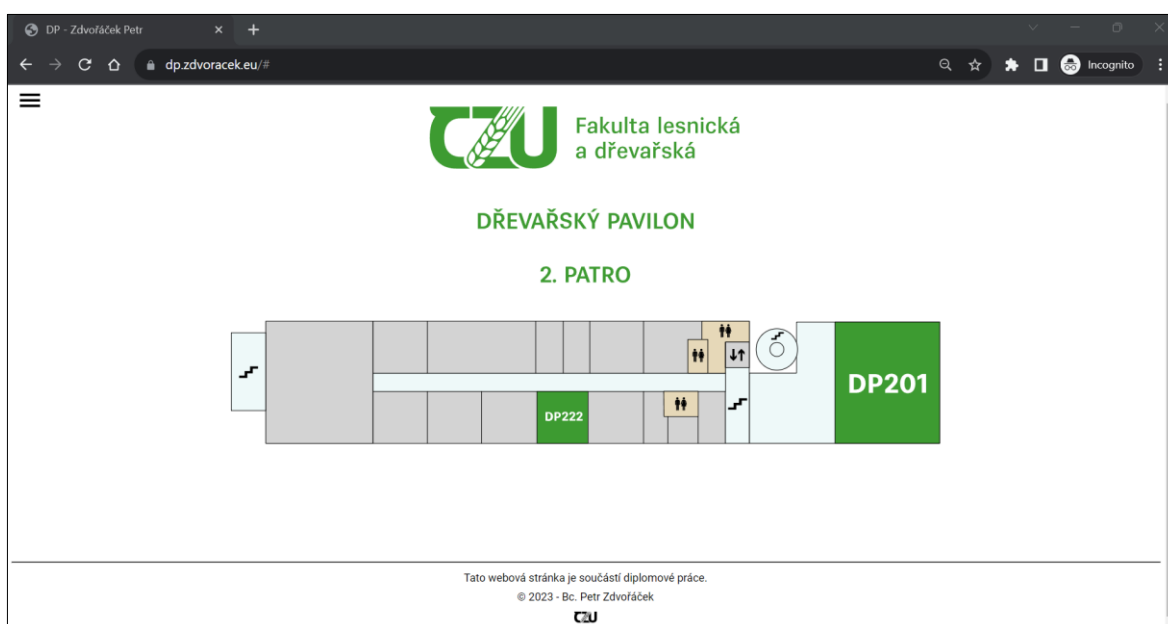
Po vytvoření HTML kódu bylo cílem nahrát stránku na web pro to, aby byla veřejně přístupná všem zájemcům. Pro tyto účely bylo potřeba nejprve si zřídit webhosting a následně si pronajmout vlastní doménu na webu. Vytvoření webhostingu i pronájem domény proběhl pomocí největší webhostingové firmy v ČR WEDOS Internet a.s. WEDOS nabízí služby jako je webhosting, domény, e-mailové řešení a servery.

Nejprve byl založen samotný webhosting a poté pronajata doména. Pro jednoduchou konfiguraci a bezpečný přenos souborů mezi lokálním a vzdáleným úložištěm byl využit WinSCP, open source klient a správce souborů pro operační systém Windows. Pomocí WinSCP byla na webhosting nahrána finální HTML stránka a všechny vektorové mapy ve formátu souboru SVG.

Finální webová stránka je pro zájemce veřejně dostupná na adrese www.dp.zdvoracek.eu. Níže je zobrazen náhled stránky po prvotním načtení ve webovém prohlížeči a následně zobrazená mapa dřevařského pavilonu na FLD.



Obrázek 37 – Webová stránka po úvodním načtení



Obrázek 38 – Webová stránka po zobrazení mapy

5 Výsledky

Výstupem diplomové práce je vytvořený soubor vektorových map jednotlivých podlaží všech fakultních budov areálu ČZU v Praze a webová stránka sloužící k jejich prezentaci, přehlednému zobrazení a procházení.

Soubor vektorových map obsahuje 48 vytvořených vektorových map, mapujících 14 budov v areálu, které vytváří zázemí všem šesti fakultám ČZU v Praze. Součet datové velikosti všech map je 824 kB a průměrná velikost jedné mapy je přibližně 17 kB. Vektorové mapy byly vytvořeny na základě sběru původních podkladů k výstavbě budov, tudíž jednotlivé mapy obsahují přesné půdorysy podlaží a přesné vymezení místností a chodeb společně s interiérovými zařízeními. Na základě podkladu oficiálního seznamu učeben na ČZU, jsou v mapách graficky vyznačeny všechny oficiální učebny v dané budově fakulty.

Dílním výstupem práce je webová stránka, která zobrazuje vytvořené vektorové mapy a slouží k jejich prezentaci. Jedná se o jednostránkový web, tedy o jednu HTML stránku, na které lze intuitivně přecházet z jednoho podlaží budovy na jiné. Vektorové mapy jsou vytvořeny ve formátu SVG a díky zákonitostem vektorové grafiky, tohoto formátu a k tomu provedené optimalizaci je jejich velikost velmi malá. To je velmi důležité právě pro jejich zobrazení a publikaci na webu, protože doba načítání je velmi krátká a pro webové prohlížeče není tak náročné mapy zobrazit. Webová stránka je dostupná na veřejné doméně www.dp.zdvoracek.eu. Na této stránce je k dispozici ke stažení také soubor všech vytvořených map.

Vytvořené vektorové mapy splňují účel, pro který byly vytvořeny. Mapují jednotlivá podlaží všech učeben areálu ČZU, sjednocují a vytvářejí jednotný styl jejich grafického zobrazení. Webová stránka dopomáhá k vizualizaci těchto map a jejich jednoduché prezentaci.

Výsledek této práce by mohl přispět k rozvoji map areálu univerzity publikovatelných na webu a zlepšit celkový zážitek virtuálních i fyzických návštěvníků univerzitního areálu. Také může sloužit jako mapový podklad pro integraci na web nebo do online platformem univerzity.

6 Závěr

Cílem této diplomové práce bylo vytvoření vektorových map ve formátu SVG, mapujících areál ČZU v Praze. Následným dílčím cílem bylo vytvoření webové stránky pro zobrazení těchto vektorových map, sloužící k jejich prezentaci. Vektorové mapy jsou k dispozici akademickému prostředí jako základ pro další tvorbu v rámci projektů univerzity nebo k jakémukoliv jinému využití.

Cílem teoretické části práce bylo popsat princip vektorové grafiky, její využití a formáty ve kterých je nejčastěji publikována. Současně také důkladně analyzovat značkovací jazyk a formát SVG, jeho specifikace, strukturu a hlavní vlastnosti. Po podrobnějším zkoumání hlavních témat následovala komparace dostupných softwarů pro tvorbu SVG vektorových map, která napomohla k výběru nástroje pro zpracování vlastní práce. Poslední části jsou zaměřeny na důležitou vlastnost formátu SVG, a to jeho možnost jednoduché publikace na webu. Dále byly nastíněny základy webové kartografie, možnosti využití právě tohoto formátu, a nakonec možnosti publikování vektorových dat na webu. Současná webová kartografie je primárně zaměřena na využití jiných formátů publikovaných na webu. Proto tato práce může mít přidanou hodnotu i v sumarizaci dosavadních studií a literárních podkladů, věnujících se využití formátu SVG v této specifické problematice. Na základě teoretických východisek a jejich aplikace ve vlastní práci došlo k naplnění stanoveného cíle práce.

Vektorové mapy byly vytvořeny v grafickém editoru Adobe Illustrator, který slouží primárně pro tvorbu vektorové grafiky. Po vytvoření map lze konstatovat, že volba tohoto softwaru byla správná. Jak vyplývá také z jeho analýzy provedené v teoretické části práce, jedná se o absolutní špičku na trhu. Klíčovou funkcí Illustratoru byly zejména možnosti exportu z grafického návrhu do formátu SVG.

Cílem praktické části práce bylo vytvoření vektorových map fakultních budov areálu ČZU v Praze. Dílčí částí bylo vytvoření webové prezentace těchto vektorových map, která samotná demonstruje splnění požadavků cíle vlastní práce. Úkolem map je sjednotit a zmapovat všechna podlaží učeben v areálu do jednotného stylu a přispět tak k rozvoji map v areálu univerzity.

Téma této práce se sice zaměřuje pouze na okrajovou část webové kartografie v podobě SVG formátu, ale za to důkladně mapuje jeho vlastnosti a výhody. S ohledem na rostoucí požadavky rychlosti přenosu webových stránek a aplikací, je formát SVG

v posledních letech všeobecně na internetu stále více využíván, a proto lze předpokládat, že má v blízké budoucnosti ve webové kartografii velký potenciál.

7 Seznam použitých zdrojů

- 2D a 3D grafika. In: *CZECH MULTIMEDIA INTERACTIVE* [online]. [cit. 2023-08-15]. Dostupné z: <https://czmi.cz/slovník/2d-3d-grafika/>
- About the Online Map Customization Tool, 2023. In: *Simplemaps.com* [online]. [cit. 2023-10-04]. Dostupné z: <https://simplemaps.com/docs/custom>
- Accessible SVGs: Perfect Patterns For Screen Reader Users, 2021. In: *Smashing Magazine* [online]. [cit. 2023-10-07]. Dostupné z: <https://www.smashingmagazine.com/2021/05/accessible-svg-patterns-comparison/>
- ADOBE, 2023. Soubory DWG. In: *Adobe* [online]. [cit. 2023-11-29]. Dostupné z: <https://www.adobe.com/cz/creativecloud/file-types/image/vector/dwg-file.html>
- Adobe Illustrator, 2023. In: *Adobe* [online]. [cit. 2023-10-02]. Dostupné z: <https://helpx.adobe.com/content/help/cz/cs/illustrator/using/svg.html>
- BOLDT, Thomas, 2023. CoreDRAW Review. In: *SoftwareHow* [online]. [cit. 2023-11-07]. Dostupné z: <https://www.softwarehow.com/coreldraw-graphics-suite-review/>
- CAJTHAML, Jiří, 2005. Využití webových mapových serverů. *16. kartografická konference: Mapa v informační společnosti-abstrakta*. (34).
- Discover QGIS, 2023. In: *QGIS* [online]. [cit. 2023-10-04]. Dostupné z: <https://www.qgis.org/en/site/about/index.html>
- EISENBERG, J.David a Amelia BELLAMY-ROYDS, 2014. *SVG Essentials: Producing Scalable Vector Graphics with XML*. 2nd edition. Sebastopol: O'Reilly Media. ISBN 978-1-4493-7435-8.
- GLITSCHKA, Von, 2013. *Vektory: základní výcvik*. Brno: Computer Press. ISBN 978-80-251-4129-8.
- GONZÁLEZ, Martín, 2022. Geospatial Data Representation: The GeoJSON Format. In: *Datascience.aero* [online]. [cit. 2023-11-29]. Dostupné z: <https://datascience.aero/geospatial-data-representation-the-geojson-format/>
- HORNÝ, Stanislav, 2006. *Počítačová grafika*. Praha: Vysoká škola ekonomická, Fakulta informatiky a statistiky. ISBN 80-245-1104-5.
- How to convert PNG to SVG, 2023. In: *Sticker Mule* [online]. [cit. 2023-08-16]. Dostupné z: <https://www.stickermule.com/blog/how-to-convert-png-to-svg>
- INDESS, S.R.O., 2015. *Půdorys budovy - Mezifakultní centrum env. věd*. 1:50. Praha.
- INKSCAPE WEBSITE DEVELOPERS, 2023. Inkscape. In: *Inkscape* [online]. [cit. 2023-10-04]. Dostupné z: <https://inkscape.org/about/>
- Mapbox Studio, 2023. In: *Mapbox* [online]. [cit. 2023-10-04]. Dostupné z: <https://docs.mapbox.com/studio-manual/guides/>

- MEGHAN, 2021. Testing SVG Support Across Web Browser Engines (Case Study). In: *Hongkiat* [online]. [cit. 2023-10-06]. Dostupné z: <https://www.hongkiat.com/blog/test-svg-format-browser-engines/>
- MURRAY, Scott, 2013. *Interactive Data Visualization for the Web*. 1st. O'Reilly Media, Inc. ISBN 9781449339739.
- NAVRÁTIL, Pavel, 2007. *Počítačová grafika a multimédia*. Kralice na Hané: Computer Media. ISBN 80-866-8677-9.
- NÉTEK, Rostislav, 2020. *Webová kartografie - specifika tvorby interaktivních map na webu*. Olomouc: Univerzita Palackého v Olomouci. ISBN 978-80-244-5827-4.
- ODBOR BEZPEČNOSTI ČZU V PRAZE, 2013. *Požární evakuační plán*. Praha.
- OGC Standards, 2023. In: *Open Geospatial Consortium* [online]. [cit. 2023-10-07]. Dostupné z: <https://www.ogc.org/standards/>
- PÁSZTO, Vít a Zdeňka KRIŠOVÁ, 2017. Počítačová grafika. In: *Moravská vysoká škola Olomouc, o. p. s.* [online]. Olomouc [cit. 2021-02-22]. Dostupné z: <https://www.mvso.cz/files/pocitacova-grafika.pdf>
- PETERSON, Michael P., 2003. *Maps and the Internet*. 1st. Omaha, USA: Pergamon. ISBN 9780080535791.
- Průvodce nového studenta, 2023. In: *ČZU* [online]. [cit. 2023-10-30]. Dostupné z: <https://www.czu.cz/cs/r-7213-studium/r-7256-informace-pro-studenty/r-8010-pruvodce-studujicich>
- STERLING, Quinn a Ryan BAXTER, 2023. Mapbox services and vector tiles. In: *GEOG 865: Open Web Mapping (The Pennsylvania State University)* [online]. [cit. 2023-10-07]. Dostupné z: <https://www.e-education.psu.edu/geog865/node/324>
- SVG Rectangle, 2023. In: *W3 schools* [online]. [cit. 2023-11-29]. Dostupné z: https://www.w3schools.com/graphics/svg_rect.asp
- SVG WORKING GROUP, 2021. Scalable Vector Graphics (SVG) 2. In: *W3C* [online]. [cit. 2023-08-17]. Dostupné z: <https://www.w3.org/TR/SVG2/>
- SVGOMG, 2023. In: *Optimize and minify SVG images* [online]. [cit. 2023-10-31]. Dostupné z: <https://svgomg.net/>
- TRIPPE, Bill a Kate BINDER, 2002. *SVG For Designers: Using Scalable Vector Graphics in Next-Generation Web Sites*. 1st. New York, United States: McGraw-Hill/OsborneMedia. ISBN 978-0072225297.
- Unleash your data and imagination with Scribble Maps, 2023. In: *Scribble Maps* [online]. [cit. 2023-10-04]. Dostupné z: <https://www.scribblemaps.com/>
- ŽÁRA, Jiří, 2004. *Moderní počítačová grafika*. Vyd 1. Brno: Computer Press. ISBN 9788025104545.

8 Seznam obrázků a zkratk

8.1 Seznam obrázků

Obrázek 1 – Vektorizace z formátu PNG do SVG (How to convert PNG to SVG, 2023) .	16
Obrázek 2 – SVG v HTML jako vložený obrázek (Eisenberg, 2014).....	21
Obrázek 3 – SVG v HTML jako element object (Eisenberg, 2014).....	22
Obrázek 4 – SVG – Obdélník (SVG Rectangle, 2023)	23
Obrázek 5 – SVG – Rectangle element (Eisenberg, 2014).....	24
Obrázek 6 – SVG – Circle a ellipse elementy (Eisenberg, 2014).....	24
Obrázek 7 – SVG – Polygon element (Eisenberg, 2014)	25
Obrázek 8 – SVG – Efekt rozdílných pravidel výplně polygonu (Eisenberg, 2014)	26
Obrázek 9 – SVG – Polyline element (Eisenberg, 2014)	26
Obrázek 10 – SVG – stroke-width (Eisenberg, 2014)	27
Obrázek 11 – SVG – Použití ořezu (Eisenberg, 2014).....	29
Obrázek 12 – Efekt průhlednosti maskování (Eisenberg, 2014)	29
Obrázek 13 – SVG – text element (weight, style, decoration) (Eisenberg, 2014).....	30
Obrázek 14 – Míry glyfu (Eisenberg, 2014).....	31
Obrázek 15 – SVG elementy alt a role (Accessible SVGs, 2021).....	33
Obrázek 16 – Software Adobe Illustrator	36
Obrázek 17 – Software Inkscape	37
Obrázek 18 – Seznam učeben na ČZU (Průvodce nového studenta, 2023)	47
Obrázek 19 – Půdorys prvního podlaží budovy MCEV ve formátu DWG (INDESS, s.r.o., 2015)	49
Obrázek 20 – Evakuační plán prvního podlaží pavilonu T Provozně ekonomické fakulty (Odbor bezpečnosti ČZU v Praze, 2013).....	50
Obrázek 21 – Upravený půdorys 1. patra Fakulty životního prostředí s vlastnoručními zápisky	51
Obrázek 22 – Tvorba mapy v Illustratoru (půdorys přízemí Technické fakulty ve formátu DWG).....	52
Obrázek 23 – Tvorba mapy v Illustratoru (převod a úpravy do formátu AI).....	52
Obrázek 24 – Tvorba mapy v Illustratoru (redukce nepotřebných objektů, seskupení objektů, ořez)	53
Obrázek 25 – Tvorba mapy v Illustratoru (tvorba a úprava křivek).....	53

Obrázek 26 – Tvorba mapy v Illustratoru (úprava barev)	54
Obrázek 27 – Tvorba mapy v Illustratoru (tvorba místností)	54
Obrázek 28 – Tvorba mapy v Illustratoru (finální mapa přízemí Technické fakulty)	55
Obrázek 29 – Tvorba mapy v Illustratoru (úprava rastrového podkladu a tvorba obrysu)..	56
Obrázek 30 – Tvorba mapy v Illustratoru (úprava barev)	56
Obrázek 31 – Tvorba mapy v Illustratoru (tvorba místností)	57
Obrázek 32 – Tvorba mapy v Illustratoru (finální mapa 1. patra budovy T Provozně ekonomické fakulty)	57
Obrázek 33 – Verze SVG v Illustratoru	58
Obrázek 34 – Volba vlastností SVG souboru v Illustratoru	59
Obrázek 35 – Online nástroj pro optimalizaci SVG kódu (SVGOMG, 2023)	60
Obrázek 36 – Wireframe webové stránky	61
Obrázek 37 – Webová stránka po úvodním načtení	63
Obrázek 38 – Webová stránka po zobrazení mapy	63

8.2 Seznam použitých zkratk

ČZU (Česká zemědělská univerzita)

SVG (Scalable Vector Graphics)

HTML (Hypertext Markup Language)

CSS (Cascading Style Sheets)

XML (Extensible Markup Language)

W3C (World Wide Web Consortium)

AI (Adobe Illustrator)

CDR (Corel Draw)

EPS (Encapsulated PostScript)

DWG (Drawing)

PDF (Portable Document Format)

JPEG (Joint Photographic Experts Group)

CAD (Computer Aided Design)

DTP (Desktop Publishing)

CMYK (Cyan, Magenta, Yellow, black)

RGB (Red, Green, Blue)

SMIL (Synchronized Multimedia Integration Language)

GIS (Geographic Information System)

OGC (Open Geospatial Consortium)

WMS (Web Map Service)

WFS (Web Feature Service)

GML (Geography Markup Language)

GeoJSON (Geographic JavaScript Object Notation)

TopoJSON (Topographic JavaScript Object Notation)

Přílohy

Příloha A – Ukázka vektorových map ve formátu SVG

Příloha B – Kód webové stránky www.dp.zdvoracek.eu

Příloha A

Vektorové mapy podlaží fakultních budov

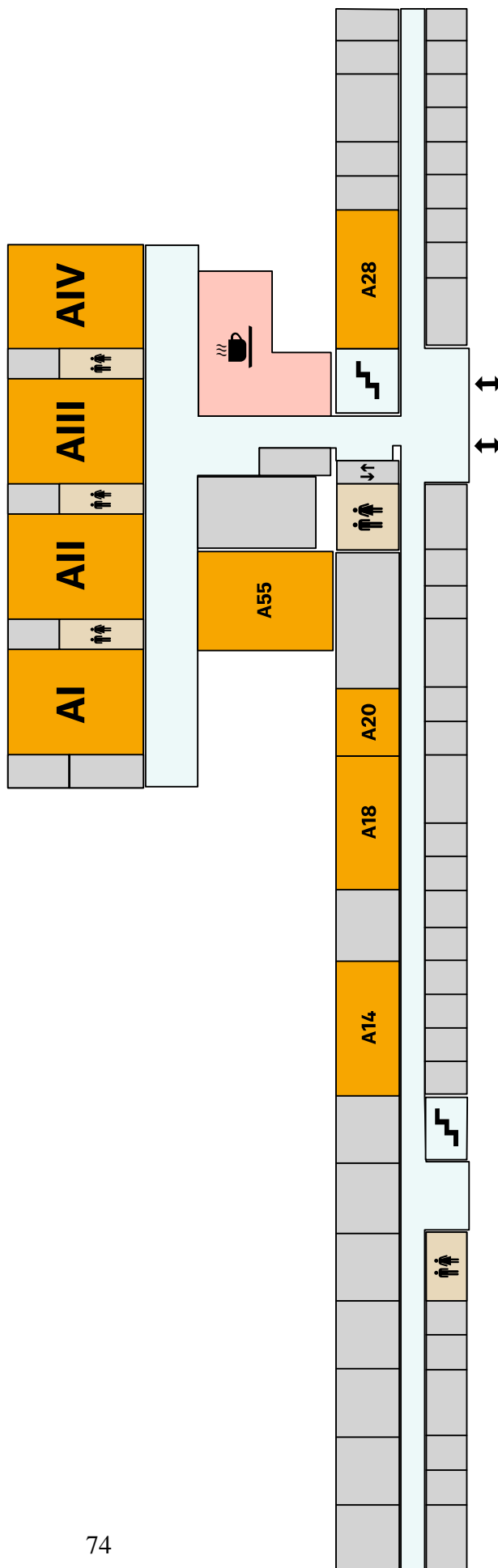
Česká zemědělská univerzita v Praze



Fakulta agrobiologie,
potravinových a přírodních zdrojů

BUDOVA A

PŘÍZEMÍ

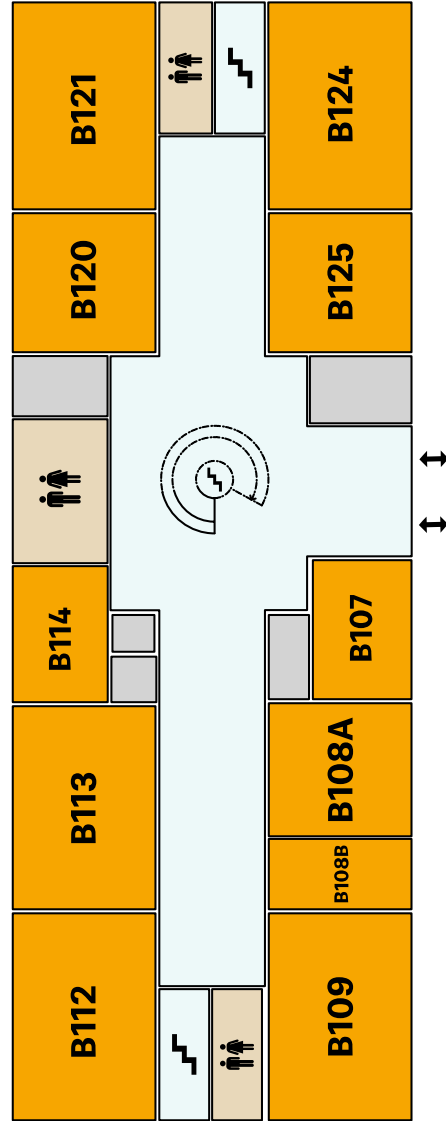




Fakulta agrobiologie,
potravinových a přírodních zdrojů

BUDOVA B

1. PATRO

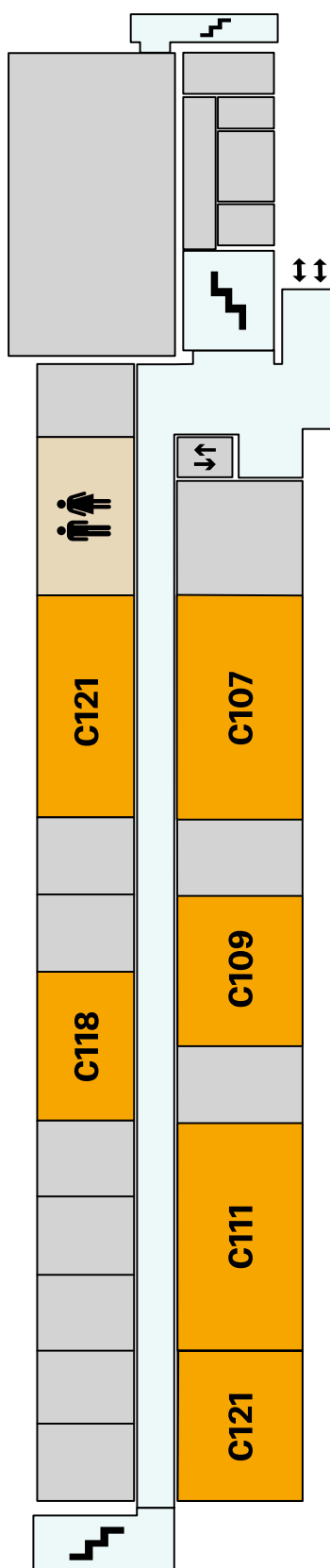




Fakulta agrobiologie,
potravinových a přírodních zdrojů

BUDOVA C

1. PATRO

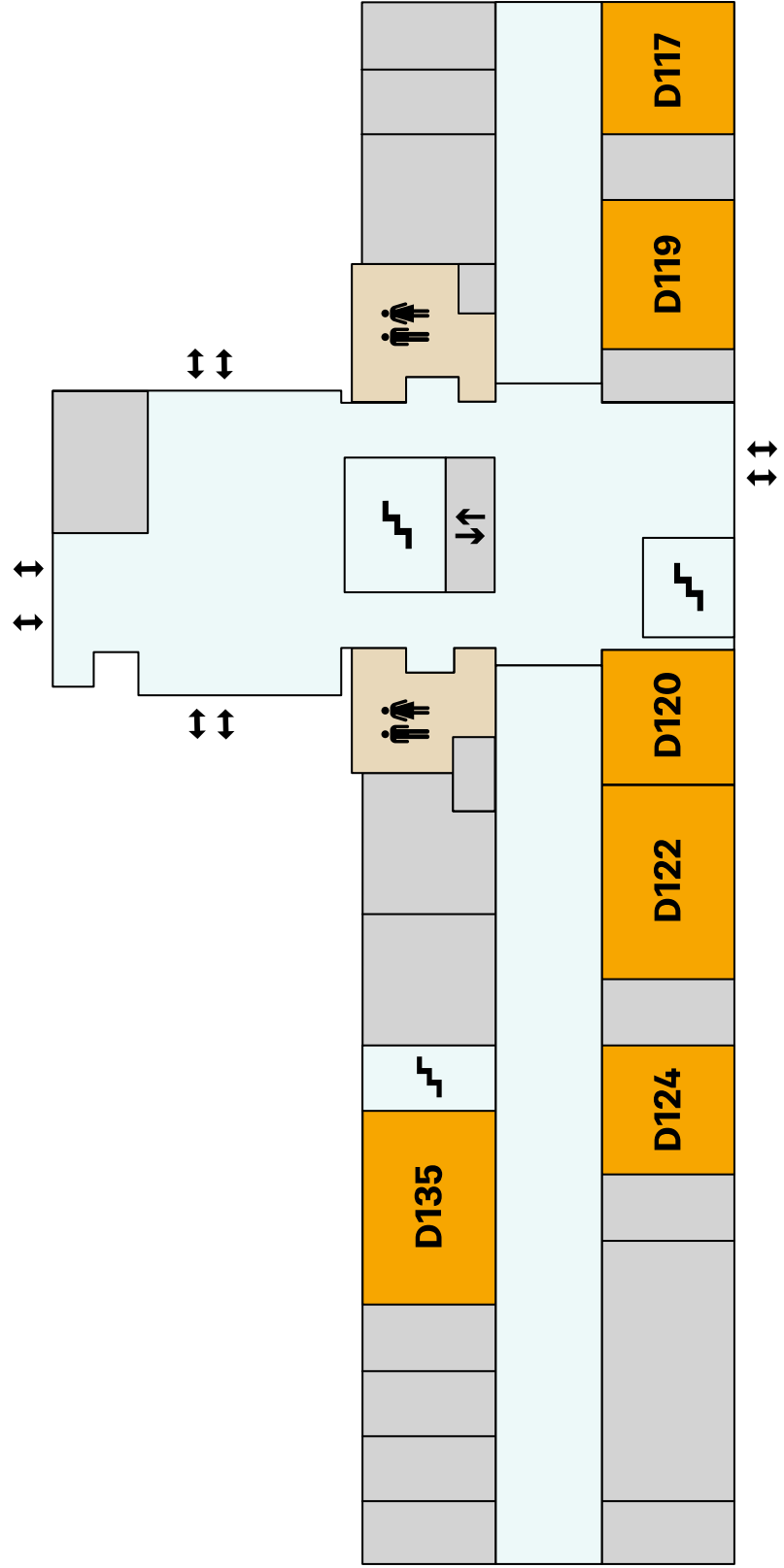




Fakulta agrobiologie,
potravinových a přírodních zdrojů

MCEV

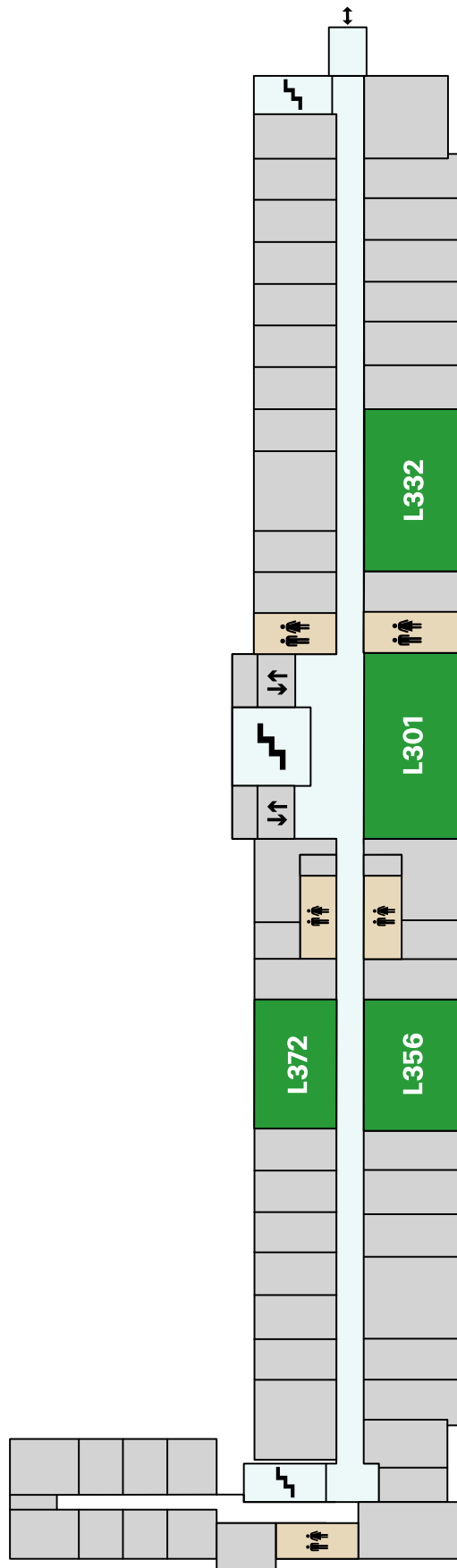
1. PATRO





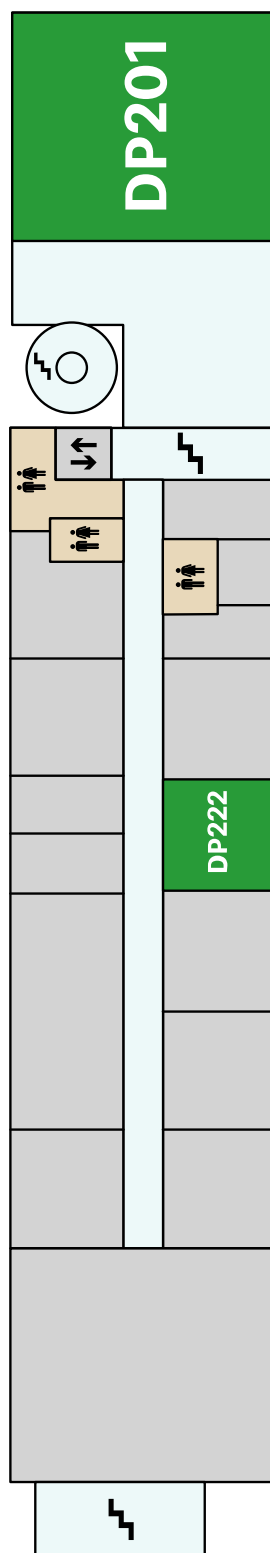
Fakulta lesnická
a dřevařská

3. PATRO



DŘEVAŘSKÝ PAVILON

2. PATRO

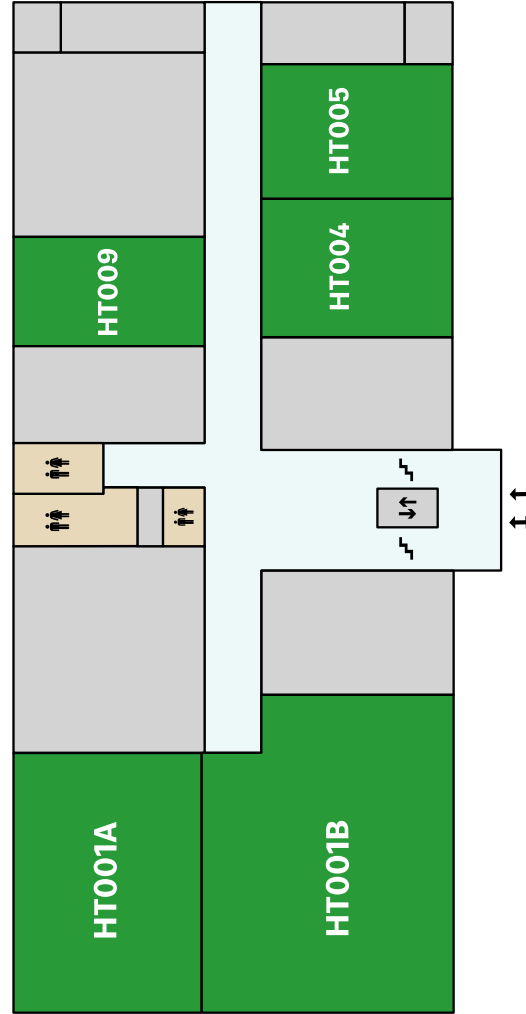




Fakulta lesnická
a dřevařská

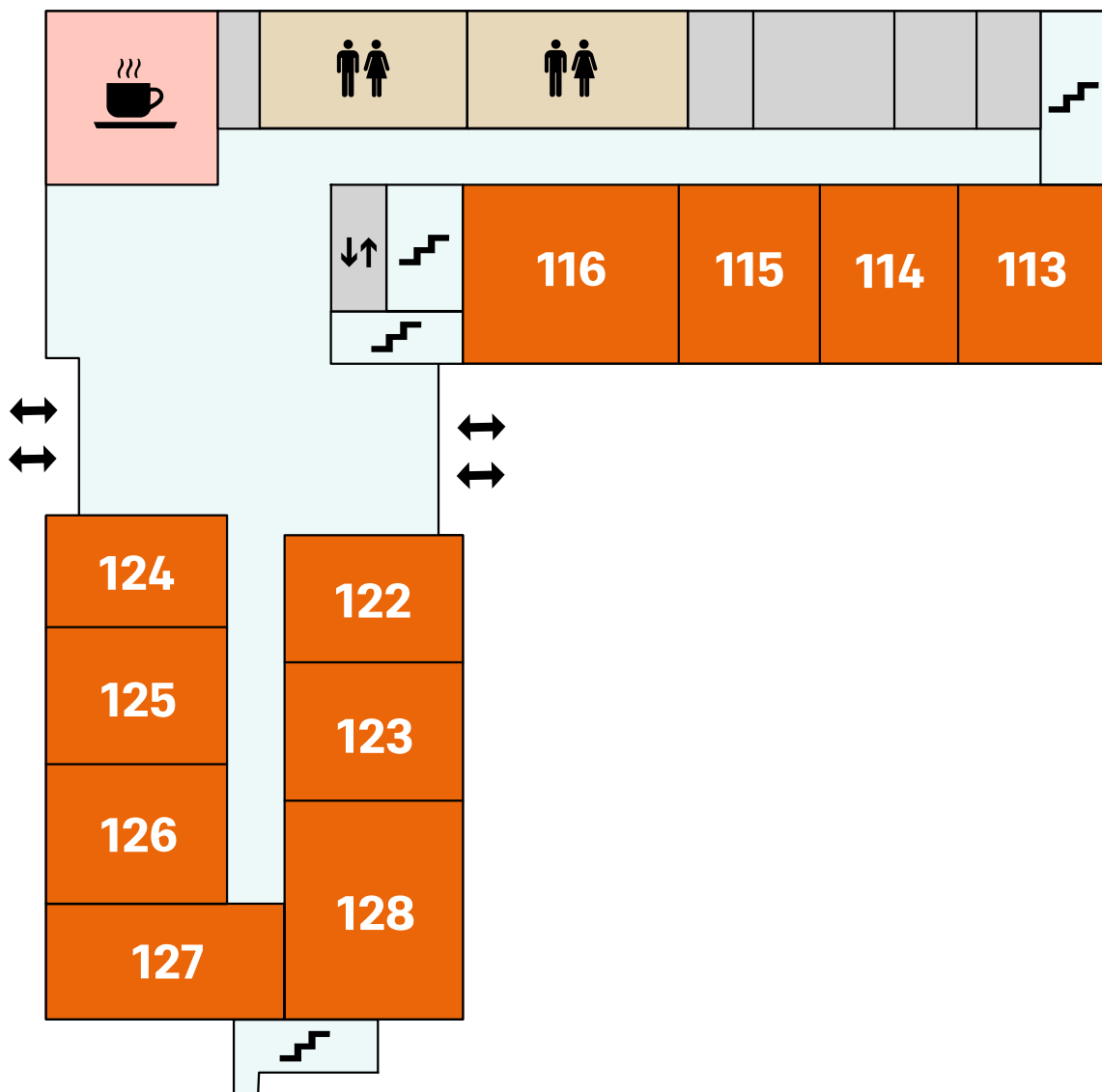
HIGH - TECH PAVILON

PŘÍZEMÍ





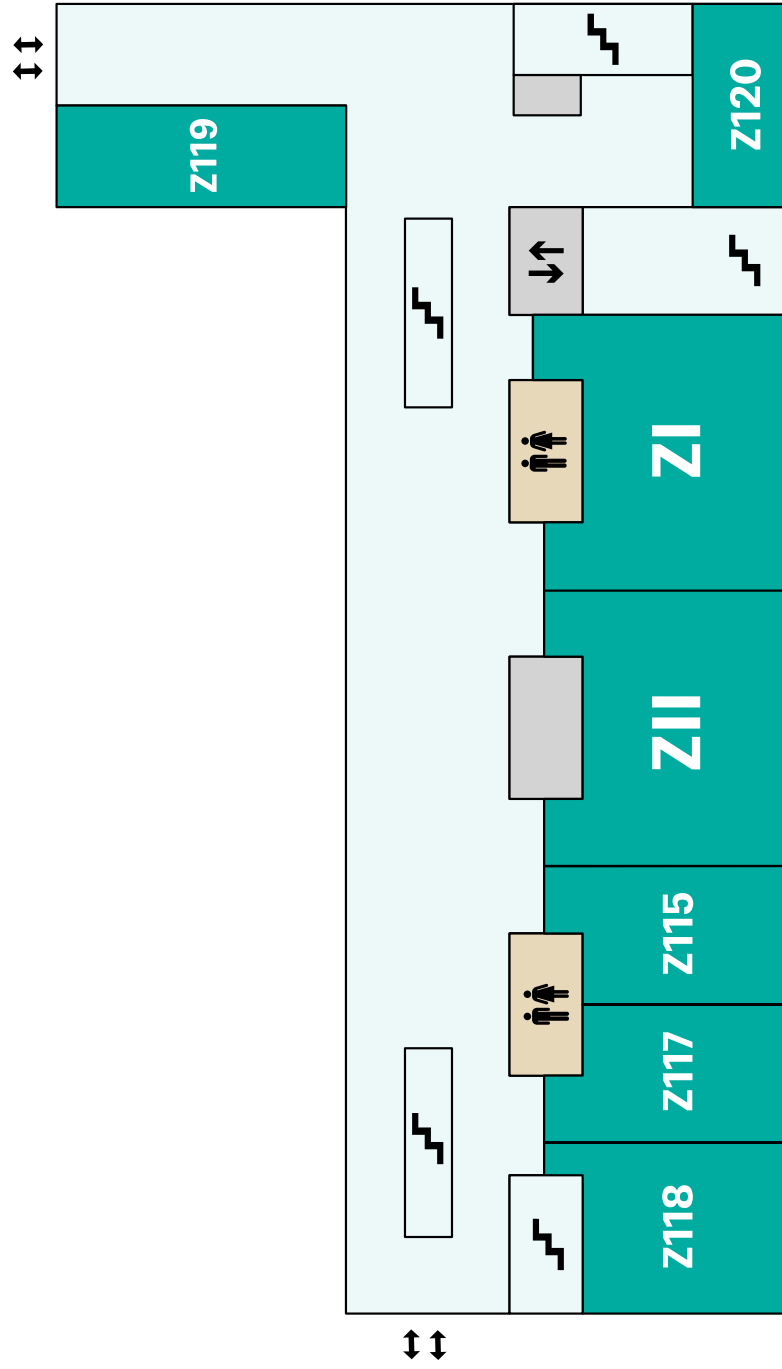
1. PATRO





Fakulta životního
prostředí

1. PATRO

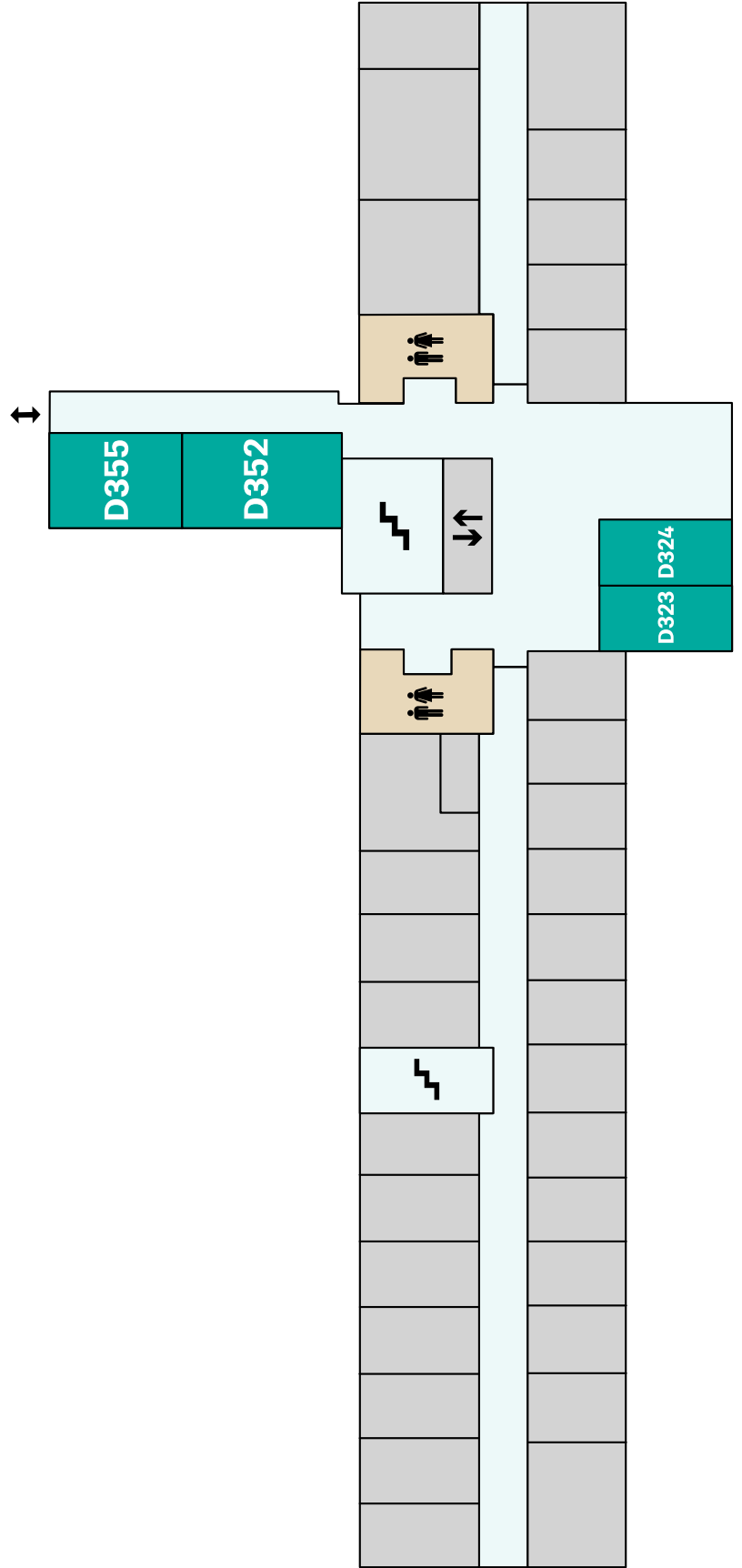




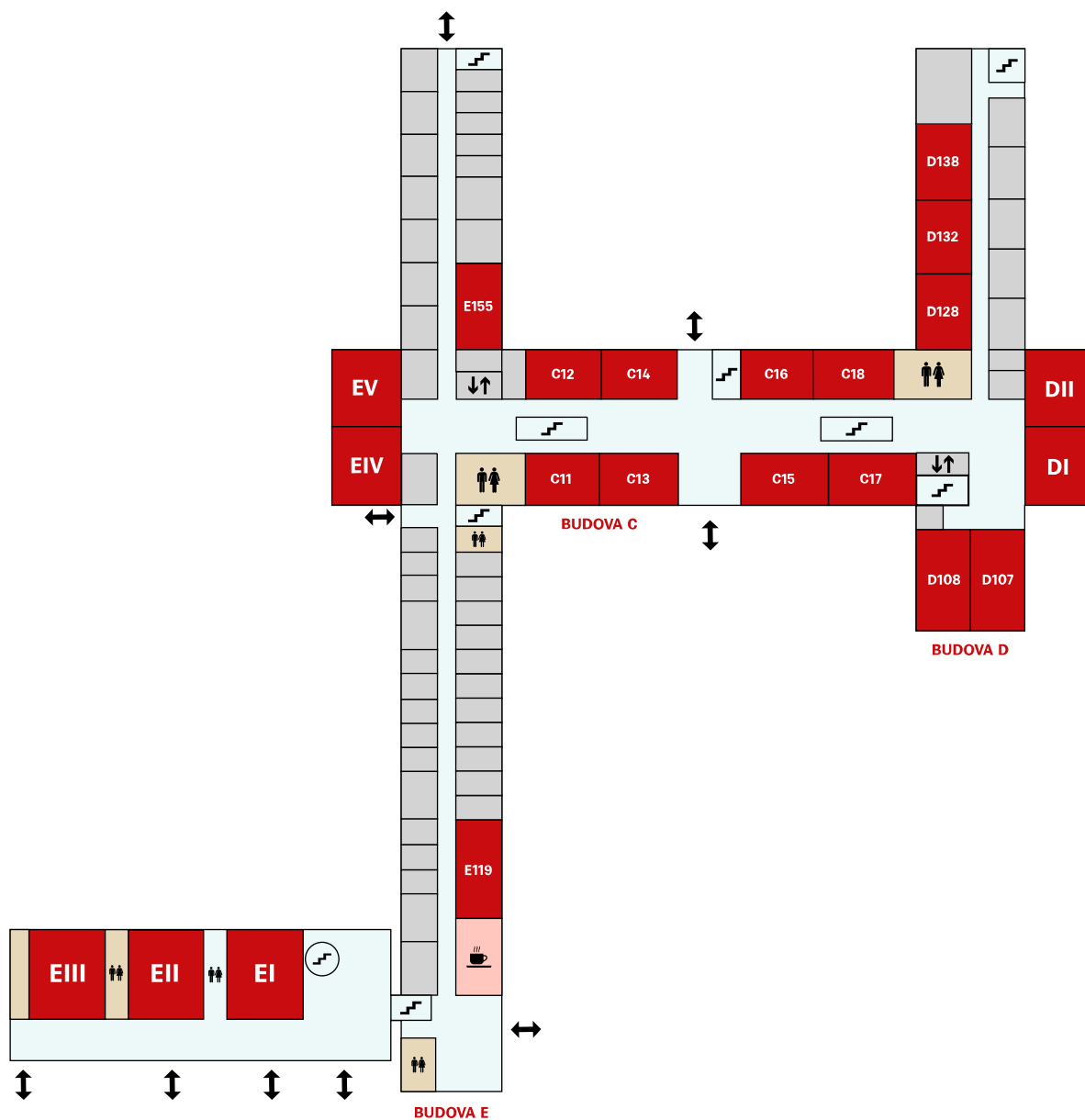
Fakulta životního
prostředí

MCEV

3. PATRO



1. PATRO

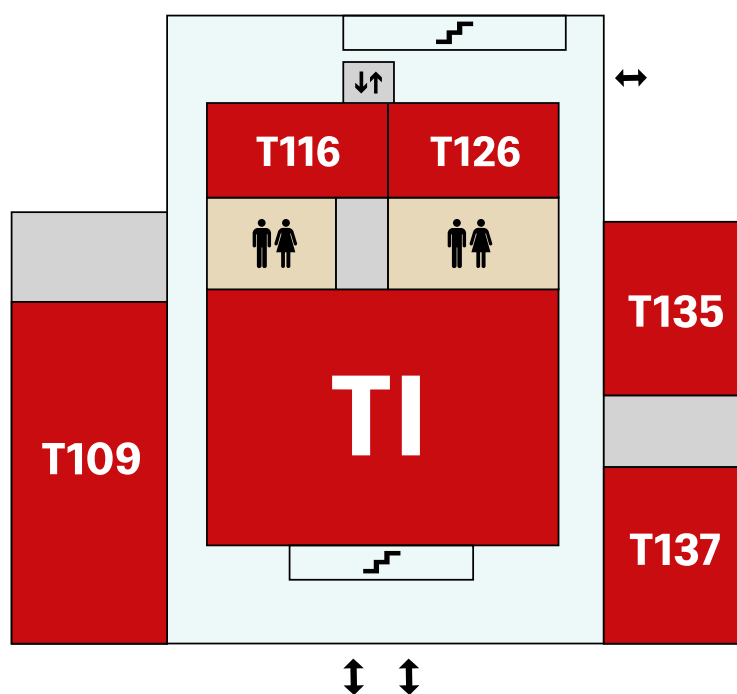




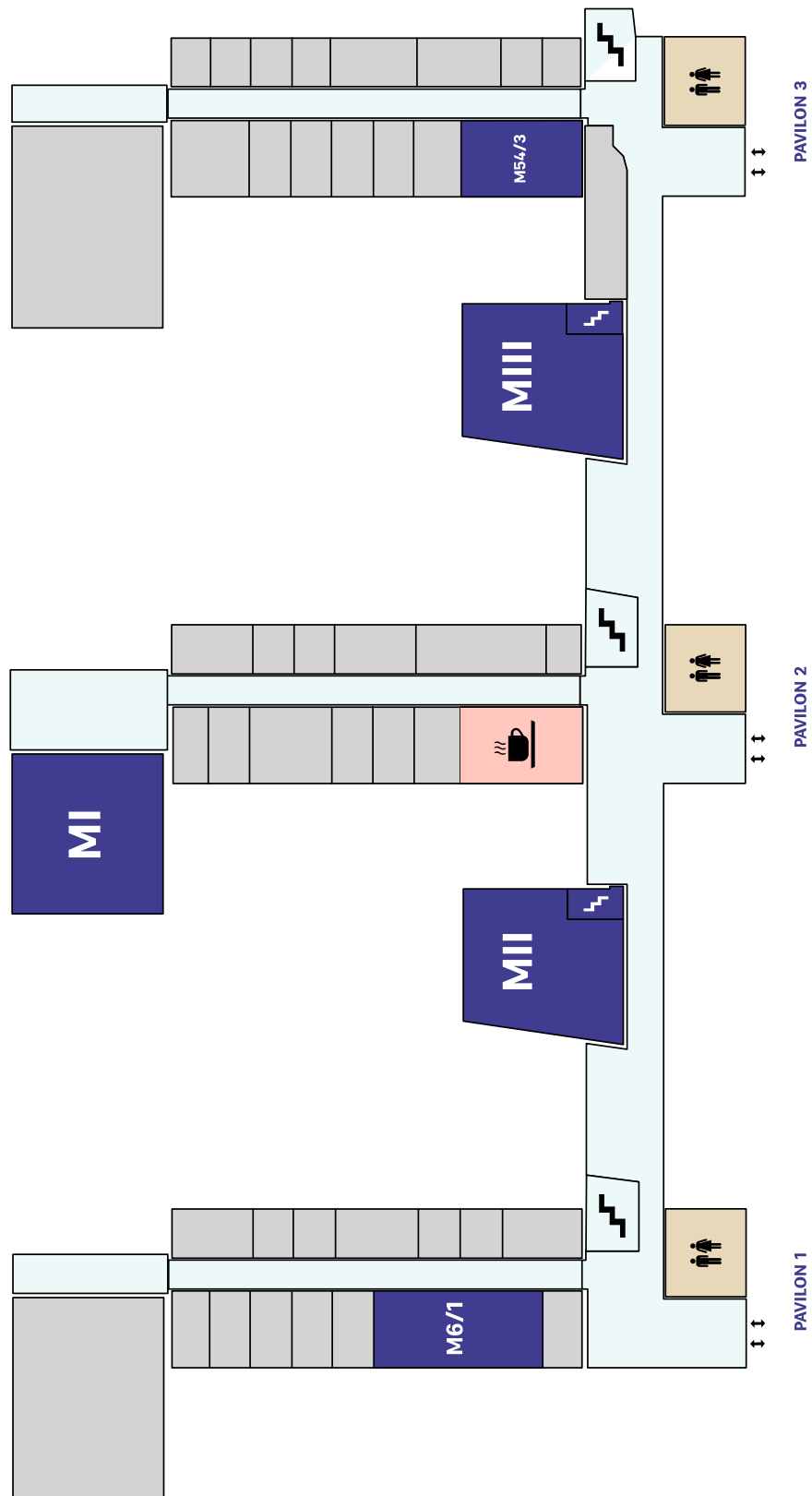
Provozně ekonomická
fakulta

BUDOVA T

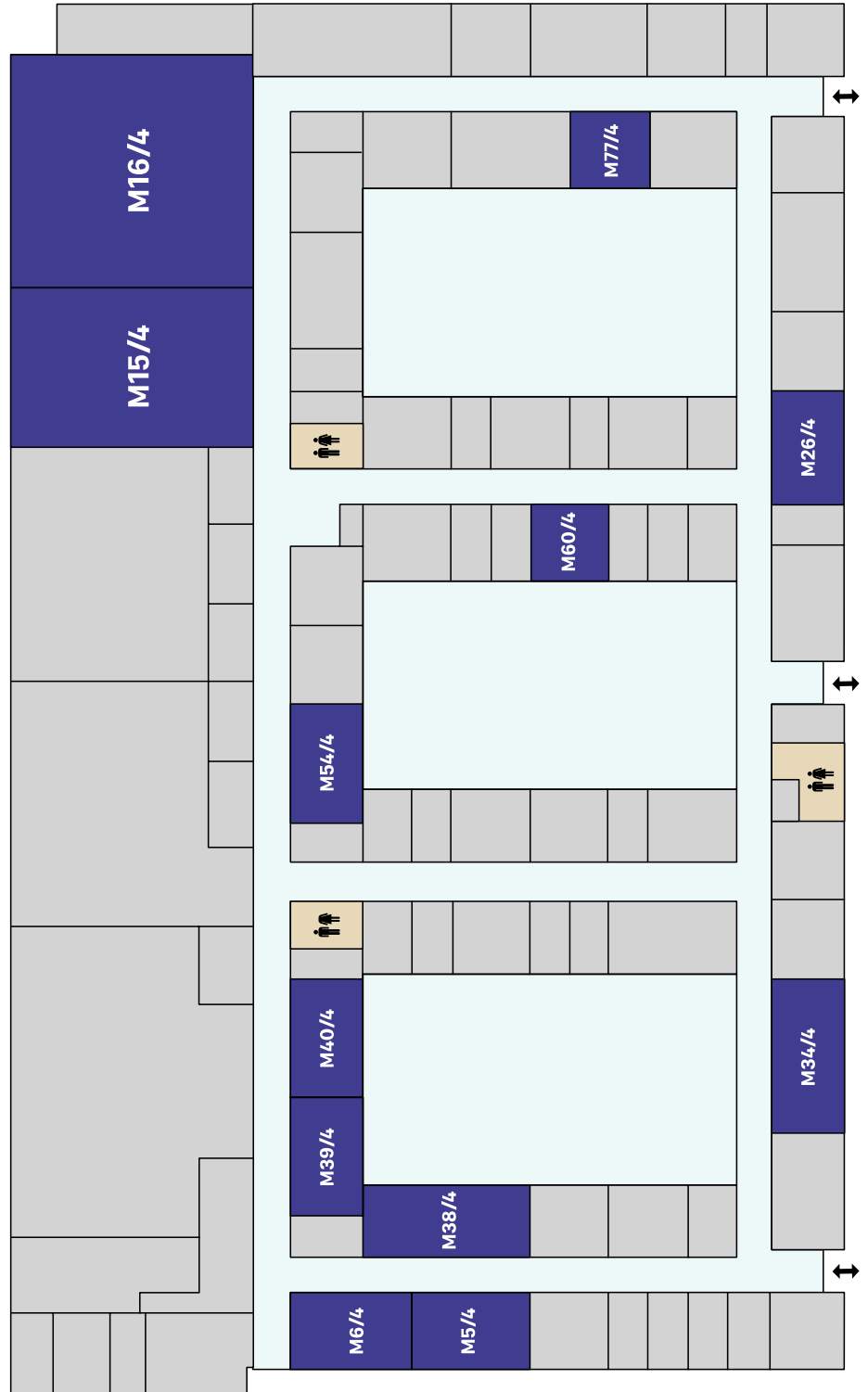
1. PATRO



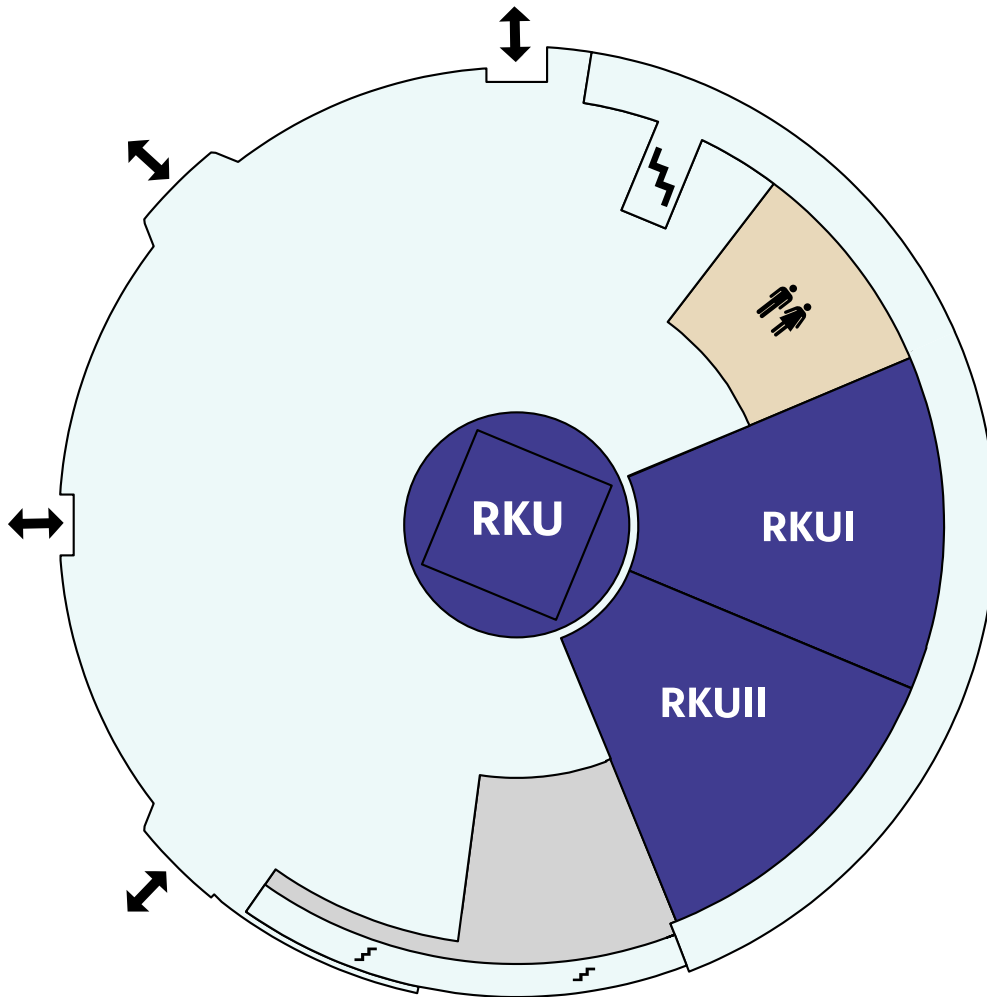
PŘÍZEMÍ



DÍLNY



KRUHOVÁ HALA



Příloha B

Webová stránka

www.dp.zdvoracek.eu

HTML, JavaScript a CSS kód

HTML a JavaScript kód

```
<!DOCTYPE html>
<html lang="cs">

<head>
  <meta charset="utf-8" />
  <title>DP - Zdvořáček Petr</title>
  <link rel="stylesheet" href="style.css" />
  <link href='https://fonts.googleapis.com/css?family=Roboto'
    rel='stylesheet'>
</head>

<body>
  <div class="hamburger-menu">
    <input id="menu_toggle" type="checkbox" />
    <label class="menu_btn" for="menu_toggle">
      <span></span>
    </label>
    <ul class="menu_box">
      <li><a onclick="GoHome();" class="menu_item"
        href="#">ÚVOD</a></li>

      <li><a onclick="ShowLv1(event, 'PEF');"
        class="menu_item" href="#">PEF</a></li>
      <ul class="menu_box2" id="PEF">
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF/PEF_1.svg');"
          class="menu_item" href="#">1. PATRO</a></li>
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF/PEF_2.svg');"
          class="menu_item" href="#">2. PATRO</a></li>
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF/PEF_3.svg');"
          class="menu_item" href="#">3. PATRO</a></li>
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF/PEF_4.svg');"
          class="menu_item" href="#">4. PATRO</a></li>
      <li><a onclick="ShowLv12(event, 'PEF-T');"
        class="menu_item" href="#">BUDOVA T</a></li>
      <ul class="menu_box2" id="PEF-T">
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF-T/PEF-T_1.svg');"
          class="menu_item" href="#">1. PATRO</a></li>
        <li><a onclick="ChangeMap.call
          (this, 'mapy/PEF-T/PEF-T_2.svg');"
          class="menu_item" href="#">2. PATRO</a></li>
      </ul>
    </ul>
  </div>
</body>
</html>
```

```

        </ul>
    </ul>
    <li><a onclick="ShowLv1(event, 'FAPPZ');" class="menu__item"
        href="#">FAPPZ</a></li>
    <ul class="menu__box2" id="FAPPZ">
        <li><a onclick="ShowLv12(event, 'FAPPZ-A');"
            class="menu__item" href="#">BUDOVA A</a></li>
        <ul class="menu__box2" id="FAPPZ-A">
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_00.svg');"
                class="menu__item" href="#">SUTERÉN</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_0.svg');"
                class="menu__item" href="#">PŘÍZEMÍ</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_3.svg');"
                class="menu__item" href="#">3. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-A/FAPPZ-A_4.svg');"
                class="menu__item" href="#">4. PATRO</a></li>
        </ul>
        <li><a onclick="ShowLv12(event, 'FAPPZ-B');"
            class="menu__item" href="#">BUDOVA B</a></li>
        <ul class="menu__box2" id="FAPPZ-B">
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-B/FAPPZ-B_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-B/FAPPZ-B_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-B/FAPPZ-B_3.svg');"
                class="menu__item" href="#">3. PATRO</a></li>
            <li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-B/FAPPZ-B_4.svg');"
                class="menu__item" href="#">4. PATRO</a></li>
        </ul>
        <li><a onclick="ShowLv12(event, 'FAPPZ-C');"
            class="menu__item" href="#">BUDOVA C</a></li>
        <ul class="menu__box2" id="FAPPZ-C">
            <li><a onclick="ChangeMap.call

```

```

                (this, 'mapy/FAPPZ-C/FAPPZ-C_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
<li><a onclick="ChangeMap.call
                (this, 'mapy/FAPPZ-C/FAPPZ-C_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
</ul>
<li><a onclick="ShowLv12(event, 'MCEV_FAPPZ');"
                class="menu__item" href="#">MCEV</a></li>
<ul class="menu__box2" id="MCEV_FAPPZ">
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_5.svg');"
                class="menu__item" href="#">5. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_6.svg');"
                class="menu__item" href="#">6. PATRO</a></li>
</ul>
</ul>
<li><a onclick="ShowLv1(event, 'TF');" class="menu__item"
                href="#">TF</a></li>
<ul class="menu__box2" id="TF">
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF/TF_00.svg');"
                class="menu__item" href="#">SUTERÉN</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF/TF_0.svg');"
                class="menu__item" href="#">PŘÍZEMÍ</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF/TF_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF/TF_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF-4/TF-4_0.svg');"
                class="menu__item" href="#">DÍLNY</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/TF-RK/TF-RK_0.svg');"
                class="menu__item" href="#">KRUHOVÁ HALA</a></li>
</ul>
<li><a onclick="ShowLv1(event, 'FLD');" class="menu__item"
                href="#">FLD</a></li>
<ul class="menu__box2" id="FLD">
    <li><a onclick="ChangeMap.call
                (this, 'mapy/FLD/FLD_0.svg');"

```

```

        class="menu_item" href="#">SUTERÉN</a></li>
<li><a onclick="ChangeMap.call
        (this, 'mapy/FLD/FLD_1.svg');"
        class="menu_item" href="#">1. PATRO</a></li>
<li><a onclick="ChangeMap.call
        (this, 'mapy/FLD/FLD_2.svg');"
        class="menu_item" href="#">2. PATRO</a></li>
<li><a onclick="ChangeMap.call
        (this, 'mapy/FLD/FLD_3.svg');"
        class="menu_item" href="#">3. PATRO</a></li>
<li><a onclick="ChangeMap.call
        (this, 'mapy/FLD/FLD_4.svg');"
        class="menu_item" href="#">4. PATRO</a></li>
<li><a onclick="ShowLv12(event, 'FLD-DP');"
        class="menu_item" href="#">DŘEVAŘSKÝ PAVILON</a></li>
<ul class="menu_box2" id="FLD-DP">
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-DP/FLD-DP_1.svg');"
        class="menu_item" href="#">1. PATRO</a></li>
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-DP/FLD-DP_2.svg');"
        class="menu_item" href="#">2. PATRO</a></li>
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-DP/FLD-DP_3.svg');"
        class="menu_item" href="#">3. PATRO</a></li>
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-DP/FLD-DP_4.svg');"
        class="menu_item" href="#">4. PATRO</a></li>
</ul>
<li><a onclick="ShowLv12(event, 'FLD-HT');"
        class="menu_item" href="#">HIGH-TECH PAVILON</a></li>
<ul class="menu_box2" id="FLD-HT">
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-HT/FLD-HT_0.svg');"
        class="menu_item" href="#">PŘÍZEMÍ</a></li>
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FLD-HT/FLD-HT_1.svg');"
        class="menu_item" href="#">1. PATRO</a></li>
</ul>
</ul>
<li><a onclick="ShowLv1(event, 'FZP');" class="menu_item"
        href="#">FŽP</a></li>
<ul class="menu_box2" id="FZP">
    <li><a onclick="ChangeMap.call
        (this, 'mapy/FZP/FZP_0.svg');"
        class="menu_item" href="#">PŘÍZEMÍ</a></li>
    <li><a onclick="ChangeMap.call

```

```

                (this, 'mapy/FZP/FZP_1.svg');"
                class="menu__item" href="#">1. PATRO</a></li>
<li><a onclick="ChangeMap.call
                (this, 'mapy/FZP/FZP_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
<li><a onclick="ChangeMap.call
                (this, 'mapy/FZP/FZP_3.svg');"
                class="menu__item" href="#">3. PATRO</a></li>
<li><a onclick="ShowLvl2(event, 'MCEV_FZP');"
                class="menu__item" href="#">MCEV</a></li>
<ul class="menu__box2" id="MCEV_FZP">
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_2.svg');"
                class="menu__item" href="#">2. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_3.svg');"
                class="menu__item" href="#">3. PATRO</a></li>
    <li><a onclick="ChangeMap.call
                (this, 'mapy/MCEV/MCEV_4.svg');"
                class="menu__item" href="#">4. PATRO</a></li>
</ul>
</ul>
<li><a onclick="ShowLvl1(event, 'FTZ');" class="menu__item"
    href="#">FTZ</a></li>
<ul class="menu__box2" id="FTZ">
    <li><a onclick="ChangeMap.call(this, 'mapy/FTZ/FTZ_1.svg');"
        class="menu__item" href="#">1. PATRO</a></li>
    <li><a onclick="ChangeMap.call(this, 'mapy/FTZ/FTZ_2.svg');"
        class="menu__item" href="#">2. PATRO</a></li>
    <li><a onclick="ChangeMap.call(this, 'mapy/FTZ/FTZ_3.svg');"
        class="menu__item" href="#">3. PATRO</a></li>
    <li><a onclick="ChangeMap.call(this, 'mapy/FTZ/FTZ_4.svg');"
        class="menu__item" href="#">4. PATRO</a></li>
</ul>
</ul>
</div>
<div class="home" id="text" style="display: block;">
    
    <p>
        <h1>Diplomová práce</h1>
        <h2>Mapování areálu ČZU s využitím vektorové grafiky
            a jazyka SVG</h2>
        <h3>Bc. Petr Zdvůřáček</h3>
    </p>
    <p class="homeContent">
        Tato webová stránka je součástí diplomové práce a slouží k
        prezentaci a přehlednému zobrazení map, které mapují jednotlivá

```

podlaží fakultních budov v areálu České zemědělské univerzity v Praze. Mapy jsou publikovány ve formátu SVG (Scalable Vector Graphics). Formát SVG byl vytvořen konsorciem W3C pro publikaci vektorové grafiky na webu. SVG je zároveň formátem, ale i jazykem, protože celý grafický soubor lze popsat pomocí jazyka XML, také proto je jeho hlavní výhodou malá datová náročnost a snadná publikace na webu.

```
</p>
```

```
<p>Kompletní soubor všech vytvořených vektorových map je také k dispozici ke stažení <strong><a href="mapy.zip"> zde</a></strong>.</p>
```

```
</div>
```

```
<div><object id="backg" data="img/mapa.svg" class="responsive" alt="mapa"></object></div>
```

```
<footer>
```

```
<hr>
```

```
<div class="footer">
```

```
<p>Tato webová stránka je součástí diplomové práce.</p>
```

```
<p>© 2023 - Bc. Petr Zdvoráček</p>
```

```

```

```
</div>
```

```
</footer>
```

```
<script>
```

```
function GoHome() {
```

```
document.getElementById('menu__toggle').checked = false;
```

```
document.getElementById('backg').style = "display: none";
```

```
document.getElementById('text').style = "display: block";
```

```
closeAllSubmenus();
```

```
document.body.style.overflow = "auto";
```

```
}
```

```
function closeAllSubmenus() {
```

```
var submenus = document.querySelectorAll('.menu__box2');
```

```
submenus.forEach(function(submenu) {
```

```
    submenu.style.display = 'none';
```

```
});
```

```
}
```

```
function ShowLv1(event, building) {
```

```
event.preventDefault();
```

```
var element = document.getElementById(building);
```

```
if (element.style.display === "block") {
```

```
    element.style.display = "none";
```

```
} else {
```

```
closeAllSubmenus();
```

```
element.style.display = "block";
```

```
}
```

```

}
function ShowLv12(event, building) {
    event.preventDefault();
    var element = document.getElementById(building);
    var parentMenuId = element.parentElement.id;
    if (element.style.display === "block") {
        element.style.display = "none";
    } else {
        document.querySelectorAll('.menu__box2')
            .forEach(function(submenu) {
                if (submenu.id !== parentMenuId) {
                    submenu.style.display = 'none';
                }
            });
        element.style.display = "block";
    }
}
function ChangeMap(map) {
    document.getElementById('backg').data = map;
    document.getElementById('backg').style = "display: block";
    document.getElementById('text').style = "display: none";
    document.getElementById('menu__toggle').checked = false;
    var items = document.querySelectorAll('.menu__item');
    items.forEach(function(item) {
        item.classList.remove('active-item');
    });
    this.classList.add('active-item');
}
</script>
</body>
</html>

```


CSS kód

```
body {
  font-family: 'Roboto';
  font-size: 22px;
}
.logo {
  width: 20%;
  height: auto;
  display: block;
  margin-left: auto;
  margin-right: auto;
}
.logoHome {
  width: 35%;
  height: auto;
  display: block;
  margin-left: auto;
  margin-right: auto;
  padding-bottom: 5px;
}
.logoFt {
  width: 2%;
  height: auto;
  display: block;
  margin-left: auto;
  margin-right: auto;
  padding-bottom: 5px;
}
.responsive {
  width: 100%;
  height: auto;
  display: none;
}
div.home {
  text-align: center;
}
div.home p.homeContent {
  margin-left: 50px;
  margin-right: 50px;
  text-align: justify;
}
div.footer {
  text-align: center;
  font-size: 16px;
}
div.footer p {
```

```

    line-height: 0.7;
}
hr {
    width: 100%;
    height: 1px;
    background-color: black;
}
#menu__toggle {
    opacity: 0;
}
#menu__toggle:checked + .menu__btn > span {
    transform: rotate(45deg);
}
#menu__toggle:checked + .menu__btn > span::before {
    top: 0;
    transform: rotate(0deg);
}
#menu__toggle:checked + .menu__btn > span::after {
    top: 0;
    transform: rotate(90deg);
}
#menu__toggle:checked ~ .menu__box {
    left: 0 !important;
}
.menu__btn {
    position: fixed;
    top: 20px;
    left: 20px;
    width: 30px;
    height: 30px;
    cursor: pointer;
    z-index: 100;
}
.menu__btn > span,
.menu__btn > span::before,
.menu__btn > span::after {
    display: block;
    position: absolute;
    width: 100%;
    height: 4px;
    background-color: #000000;
    transition-duration: .25s;
}
.menu__btn > span::before {
    content: '';
    top: -8px;
}

```

```

.menu__btn > span::after {
  content: '';
  top: 8px;
}
.menu__box {
  display: block;
  position: absolute;
  top: 0;
  left: -100%;
  width: 300px;
  height: 100vh;
  margin: 0;
  padding: 50px 0;
  list-style: none;
  background-color: #ECEFF1;
  box-shadow: 2px 2px 6px rgb(0, 0, 0);
  transition-duration: .25s;
  transition: left 0.25s;
  box-sizing: border-box;
  overflow-y: auto;
}
.menu__box2 {
  list-style: none;
  background-color: #a3a3a3;
  display: none;
}
.menu__item {
  display: block;
  padding: 12px 24px;
  color: #333;
  font-family: 'Roboto', sans-serif;
  font-size: 20px;
  font-weight: 600;
  text-decoration: none;
  transition-duration: .25s;
}
.menu__item:hover {
  background-color: #CFD8DC;
}
.active-menu {
  background-color: #f0f0f0;
}
.active-item {
  text-decoration: underline;
}

```