

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Brachaczek** Jméno: **Petr** Osobní číslo: **434801**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**  
Studijní program: **Otevřená informatika**  
Studijní obor: **Počítačová grafika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Ilustrativní vizualizace primárního okruhu jaderné elektrárny**

Název diplomové práce anglicky:

**Illustrative visualization of the primary circuit of a nuclear power plant**

Pokyny pro vypracování:

Seznamte se s primárním okruhem jaderné elektrárny a ději, které v něm probíhají. Dále se seznámte s metodami pro nefotorealistické a ilustrativní vykreslování 3D objektů a s metodami usnadňujícími navigaci uživatele ve 3D scéně. Na základě analýzy navrhnete a implementujete aplikaci, která umožní vizualizovat prostorové uspořádání jednotlivých částí primárního okruhu jaderné elektrárny a fyzikální děje v tomto okruhu probíhající. V rámci vizualizace se zaměříte zejména na směřování pozornosti uživatele na důležité části primárního okruhu v závislosti na regionu zájmu uživatele (např. pokud bude regionem zájmu uživatele výparník, bude pozornost uživatele směřována na části primárního okruhu, které s výparníkem souvisí). Uživateli umožněte snadno měnit jejich region zájmu. Dále se zaměříte na jednoduchou navigaci ve 3D scéně. Uvažujte, že cílovou skupinou pro výslednou vizualizaci je laická veřejnost, která se skrze výslednou vizualizaci bude schopna s primárním okruhem reaktoru a v něm probíhajícími ději seznámit. Výslednou aplikaci otestujte pomocí testu použitelnosti s 6 až 10 uživateli.

Seznam doporučené literatury:

Čmolík L. Interactive Illustrative Visualization of 3D Models. PhD thesis, Czech Technical University in Prague, 2011.  
Bruckner S., S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving exploration of volume data. IEEE Transactions on Visualization and Computer Graphics, 12(6):1559–1569, 2006.  
de Moura Pinto, F., and C. M. D. S. Freitas. Illustrating volume data sets and layered models with importance-aware composition. The Visual Computer, 27(10):875–886, 2011.

Jméno a pracoviště vedoucí(ho) diplomové práce:

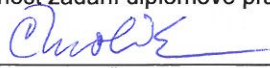
**Ing. Ladislav Čmolík, Ph.D., Katedra počítačové grafiky a interakce**

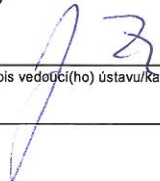
Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

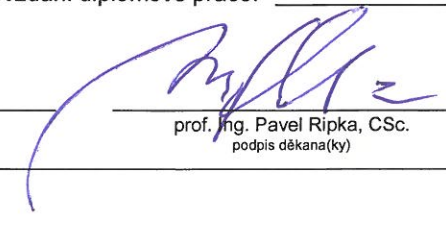
Datum zadání diplomové práce: **14.02.2019**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce: **20.09.2020**

  
Ing. Ladislav Čmolík, Ph.D.  
podpis vedoucí(ho) práce

  
podpis vedoucí(ho) ústavu/katedry

  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

7.5.2019

Datum převzetí zadání

Budka

Podpis studenta

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce

## Ilustrativní vizualizace primárního okruhu jaderné elektrárny

**Bc. Petr Brachaczek**

Vedoucí: Ing. Ladislav Čmolík, Ph.D.  
Studijní program: Otevřená informatika  
Obor: Počítačová grafika  
Květen 2019



## Poděkování

Děkuji vedoucímu diplomové práce Ing. Ladislavu Čmolíkovi, Ph.D., za odborné vedení spolu s ochotným přístupem. Také děkuji svým rodičům, kteří mě ve studiu po celou dobu podporovali.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze 25. května 2019

Podpis:.....

## Abstrakt

V této diplomové práci řeším komplexní metody ilustračních technik pro zobrazení složitých objektů sestavených z mnoha částí. Kombinuji různé ilustrační techniky, dynamicky umísťované popisky komponent spolu s vhodným způsobem navigace, abych vytvořil uživatelsky přívětivou aplikaci vizualizující primární okruh jaderné elektrárny. Metody implementuji tak, aby aplikace zobrazovala primární okruh v reálném čase.

**Klíčová slova:** ilustrativní vizualizace; jaderná elektrárna; zobrazení v reálném čase; Unity engine

**Vedoucí:** Ing. Ladislav Čmolík, Ph.D.

## Abstract

In this thesis I am using complex illustrative methods to achieve effective display of complex objects composed of many parts. I combine multiple of these illustrative effects, dynamic labeling with simple and easy to use navigation system to present user-friendly application, which is visualizing primary loop of nuclear powerplant. The illustrative methods are implemented in such way that the application is running in real-time.

**Keywords:** illustrative visualization; nuclear powerplant; real-time rendering; Unity engine

## Obsah

<b>1 Úvod</b>	<b>1</b>	5.2 Ovládání a navigace . . . . .	38
1.1 Související práce . . . . .	1	5.3 Textový popis a popisky komponent ve scéně . . . . .	39
<b>2 Jaderná elektrárna</b>	<b>3</b>	5.4 Vizualizace cirkulačních dějů . . .	40
2.1 Princip fungování jaderné elektrárny . . . . .	3	5.5 Ilustrační efekty vizualizace . . .	41
2.1.1 Reaktor . . . . .	4	<b>6 Uživatelské testování</b>	<b>43</b>
2.1.2 Parogenerátor . . . . .	4	6.1 Participanti . . . . .	43
2.1.3 Cirkulační čerpadlo . . . . .	5	6.2 Prostředí kvalitativního testování	43
2.1.4 Kompenzátor objemu . . . . .	5	6.3 Struktura testu . . . . .	44
<b>3 Analýza a návrh</b>	<b>7</b>	6.4 Výsledky testování . . . . .	45
3.1 Požadavky softwaru . . . . .	7	6.4.1 Praktická část . . . . .	46
3.2 Interaktivita . . . . .	8	6.4.2 Další poznatky v praktické části . . . . .	48
3.3 Navigace v modelu . . . . .	8	6.4.3 Post test dotazník . . . . .	49
3.3.1 Oblasti zájmu . . . . .	9	6.5 Nálezy . . . . .	52
3.4 Vizuální informace . . . . .	10	6.5.1 Pohyby pohledu kamery . . . .	52
3.4.1 Reprodukce tvarů . . . . .	10	6.5.2 Způsoby přechodu mezi detaily	52
3.4.2 Reprodukce struktur . . . . .	13	6.5.3 Propojení vizualizace s textem	53
3.4.3 Zvýraznění dalších informací	15	6.5.4 Zobrazení cirkulačních dějů .	53
3.4.4 Depth-peeling . . . . .	15	6.6 Zhodnocení testu . . . . .	54
3.4.5 Zvolené metody pro reprodukci vizuální informace . . . . .	17	<b>7 Závěr</b>	<b>55</b>
3.5 Verbální informace . . . . .	17	7.1 Budoucí práce . . . . .	56
3.5.1 Popisky a jejich umístování .	17	<b>A Literatura</b>	<b>57</b>
3.5.2 Zvolené metody pro reprodukci verbální informace . . . . .	23	<b>B Soubory přílohy</b>	<b>59</b>
3.6 Zobrazení cirkulačních dějů . . .	23		
3.6.1 Vizualizace skalárních dat . .	24		
3.6.2 Vizualizace vektorových dat .	27		
3.7 Souhrn zvolených technologií . .	28		
<b>4 Implementace</b>	<b>29</b>		
4.1 Nástroje a frameworky . . . . .	29		
4.2 Vytvoření modelu primárního okruhu . . . . .	29		
4.3 Použití Unity enginu . . . . .	30		
4.3.1 Převzetí kontroly nad vykreslováním . . . . .	30		
4.3.2 Implementace metody depth-peeling . . . . .	31		
4.3.3 Implementace popisků . . . . .	32		
4.3.4 Implementace pohyblivých glyphů . . . . .	33		
4.3.5 Výkon aplikace . . . . .	33		
<b>5 Výsledky</b>	<b>37</b>		
5.1 Rozvržení uživatelského rozhraní aplikace . . . . .	37		

## Obrázky

2.1 Schéma tlakovodního generátoru (převzato z [ČE16]) .....	4
2.2 Schéma reaktoru (převzato z [ČE16]) .....	4
2.3 Půdorys a bokorys parogenerátoru (převzato z [ČE16]) .....	5
3.1 Rozčlenění funkčních celků do hierarchické struktury. Každý funkční celek obsahuje svou oblast zájmu (detail modelu v aplikaci). Přechod mezi dvěma oblastmi zájmu je možné, jen pokud jsou přímo propojené ...	9
3.2 Demonstrace efektu cool-to-warm stínování. V červeném rámečku lze vidět dramatickou změnu ve zvýšení čitelnosti tvarů ve srovnání s Phongovým stínováním (Převzato z [GGSC99]) .....	10
3.3 Zvýraznění siluet pomocí skalárního součinu normály povrchu a vektoru ke kameře. Umocnění výsledku generuje různě silný efekt (zleva hodnoty mocnin: 0.5, 1 a 2). Vyznačování siluet pomocí stínování je horší u hranatých objektů .....	11
3.4 Efekt zvýraznění hran u oblých a hranatých objektů. Horní dva snímky ilustrují parogenerátor bez použití detekce hran (nalevo) a se zvýrazněním hran (napravo). Spodní čtyři výřezy jsou detailem oblasti vyznačené červeným rámečkem. <i>a</i> ) zobrazuje stav bez jakéhokoli zvýraznění hran, <i>b</i> ) zobrazuje detekci hran na základě diskontinuity hloubky, <i>c</i> ) demonstruje detekci diskontinuity v bufferu normál a konečně <i>d</i> ) zachycuje kombinaci detekce hran na základě normál i hloubky. Za zmínku stojí fakt, že teprve poslední výřez správně zvýrazní obě hrany vystouplého okraje světle šedého plechu .....	12
3.5 Ukázka zvýraznění křivosti povrchu průhledných těles pomocí neprůhledných linek v textuře poloprůhledného materiálu (převzato z [IFP96]) .....	12
3.6 Zvýraznění tvarů oblých poloprůhledných povrchů dosaženého pomocí spekulárních odrazů. Nalevo je zachycen sběrač páry u parogenerátoru bez odrazů, napravo s odrazy .....	13
3.7 Ukázka řešení kontextu obalu volumetrických dat (převzato z [BGKG05]) .....	14
3.8 Prolínání poloprůhledných objektů pomocí alfa kanálu (nalevo) a kombinací alfa kanálu a hodnoty důležitosti (napravo). Vhodným nastavením hodnot důležitostí lze dosáhnout daleko čitelnějších výsledků (převzato z [PF11]) .....	14
3.9 Metody exploded-view (napravo) a řezy objektem (nalevo) jakožto rozdílné přístupy k zobrazení vnitřních struktur celků složených z mnoha komponent (převzato z [LRA <sup>+</sup> 07] a [LACS08]) .....	15
3.10 Skladba poloprůhledných pixelů v rozdílném pořadí generuje rozdílné výsledky .....	16
3.11 Příklad situace, kdy seřazením tří objektů nelze vyřešit správné prolínání poloprůhledných ploch ..	16
3.12 Ukázka částí, ze kterých se skládá typický popis .....	18
3.13 Ukázka externích popisků ( <i>a</i> ) a interních popisků ( <i>b</i> ) .....	18
3.14 Silově založené rozmísťování popisků. Nalevo je případ, kdy se anotační texty překrývají. Působí na sebe odpudivými silami, které postupně dovedou anotační texty do pozic, ve kterých k překrytí nedochází .....	19



3.15 Vyřešení překryvu anotací pomocí jediného stupně volnosti (pohyb podél vodící čáry) . . . . .	20	4.2 Závislost snímkové frekvence na multiplikátoru rozlišení a počtu vykreslených vrstev metodou depth-peeling. Výsledky se vztahují k nativnímu rozlišení fullHD (1920x1080) . . . . .	35
3.16 Řešení kolizí anotačních textů pomocí jednoho stupně volnosti pohybu podél vodící čáry (nahore) a pomocí dalších dvou stupňů volnosti pohybu anotačního textu (dole) (Převzato z [TKGS14]) . . . . .	21	5.1 Rozvržení uživatelského rozhraní aplikace . . . . .	37
3.17 Na obrázcích A a B okamžitě vidíme, že červené prvky vystupují z řady. Stejný efekt se projeví i u obrázků C a D, u kterých si můžeme všimnout tvarových outlayerů. Kombinace tvarů a barev u obrázků E a F však již efekt značně zeslabují (převzato z [Mun14]) . . . . .	22	5.2 Po najetí kurzorem myši na popisek dojde ke zvýraznění popsané komponenty . . . . .	38
3.18 Ukázka situace, kdy není jednoznačné, ke které komponentě kotva popisku patří (a). Silný efekt zvýraznění po najetí na popisek víceznačnost kotev řeší (b) . . . . .	23	5.3 Po najetí kurzorem myši na popisek dojde ke zvýraznění popsané komponenty . . . . .	39
3.19 Ukázka různých barevných stupnic. B,C,D jsou stupnice sekvenční; E,F jsou stupnice divergující; A je stupnice cyklická (převzato z [Tel14]) . . . . .	25	5.4 Najetím kurzorem myši na vyznačený termín v textu se zvýrazní i související komponenta ve scéně . . . . .	40
3.20 Ukázka barevných prostorů RGB a HSL (převzato z [Sha08]) . . . . .	26	5.5 Porovnání klasického módu s módem zobrazení cirkulace chladiva v prvním cirkulačním okruhu . . . . .	40
3.21 Ukázka percepčně neuniformní a percepčně uniformní duhové barevné stupnice (převzato z [Mun14]) . . . . .	26	5.6 Reaktor je zpracován ve statickém řezu. Jeho plášť je zprůhledněn, vnitřní komponenty jsou díky ilustračním technikám dobře viditelné . . . . .	41
3.22 Vizualizace vektorového pole pomocí glyphů tvaru kužele (A) a šipek (B) (převzato z [Tel14]) . . . . .	27	5.7 Percepci hloubky napomáhá efekt mlhy, kdy jsou objekty za poloprůhlednou plochou zobrazeny s různou výrazností v závislosti na jejich vzdálenosti od poloprůhledné bariéry. V tomto příkladě se díváme na část parogenerátoru a reaktoru skrze poloprůhledný plášť kompenzátoru objemu. V červeném rámečku si všimněte, že ačkoliv jsou tvary parogenerátoru dobře čitelné, reaktor je za poloprůhledným pláštěm takřka neviditelný. Zajímavý efekt tvoří trubky vedoucí do reaktoru, které s rostoucí vzdáleností postupně mizí . . . . .	42
4.1 Nastavení grafické kvality řídí rozlišení bufferu, do kterého se vykresluje prostorová scéna. Textové popisky komponent a uživatelské rozhraní jsou vykresleny vždy v nativní kvalitě . . . . .	34	5.8 Efekt mizejících trub, které vedou mimo aktivní oblast zájmu . . . . .	42
		6.1 Odpovědi uživatelů na 1. otázku post test dotazníku . . . . .	49

## Tabulky

6.2 Odpovědi uživatelů na 2. otázku post test dotazníku .....	49
6.3 Odpovědi uživatelů na 3. otázku post test dotazníku .....	50
6.4 Odpovědi uživatelů na 4. otázku post test dotazníku .....	50
6.5 Odpovědi uživatelů na 5. otázku post test dotazníku .....	50
6.6 Odpovědi uživatelů na 6. otázku post test dotazníku .....	51
6.7 Odpovědi uživatelů na 7. otázku post test dotazníku .....	51
6.8 Odpovědi uživatelů na 8. otázku post test dotazníku .....	52
6.9 Lidé v evropských zemích prohlížíjí obrazovku specifickým způsobem (takzvaný z vzor), který je spojený též se zvyklostí čtení směrem zleva doprava (převzato z [Bab17])	54
6.1 Charakteristika testovaných uživatelů .....	43

# Kapitola 1

## Úvod

Ilustrace komplexních systémů, jako jsou například stroje, dopravní prostředky, ale i lidské tělo, dlouhou dobu patřila do domény uměleckých ilustrátorů, kteří tyto technické kresby tvořili. Hlavní nevýhoda takových ilustrací je to, že jsou statické. V dnešní době však do výuky čím dál více zasahují informační technologie a začínají se používat různé interaktivní aplikace. Obliba takových aplikací může velice snadno převyšovat klasické učebnice. Předností elektronického média jsou převážně nové možnosti interaktivity. Zobrazované ilustrace rovněž přestávají být svázané pouze se statickým pohledem.

Problém však nastává tehdy, kdy se chceme v prostoru ilustrace pohybovat, protože generovat dynamicky obrazy podobné těm v knížkách není triviální problém. Doposud byla vydána celá řada publikací, které se týkají rozličných metod, jež reprodukují určité prvky ruční ilustrace. Ty jsou však navrhovány samostatně a k jejich vzájemné kombinaci příliš nedochází. Cílem této práce je vybrat vhodné ilustrační techniky a zkombinovat je tak, aby co nejlépe reprodukovaly ilustrované objekty.

Tato práce ilustrační přístup demonstruje na komplexní vizualizaci primárního okruhu jaderné elektrárny. V kapitole 2 si nejdříve přiblížíme teorii ohledně fungování jaderné elektrárny a zaměříme se na popis komponent, ze kterých se skládá primární okruh. Kapitola 3 se věnuje analýze požadavků aplikace a řeší vizualizační metody, které budou v aplikaci použity. Následuje kapitola 4, ve které popíšu implementační detaily vizualizace. Kapitola 5 shrnuje výsledky implementace a obrazově demonstruje funkce aplikace. Uživatelské testování a jeho výsledky pak popisuje kapitola 6.

### 1.1 Související práce

Nejen ilustrační vizualizační techniky shrnuje například Ladislav Čmolík ve své disertační práci [Čm11]. Práce však není vystavěna kolem vizualizace jednoho složitěho systému objektů za použití kombinace více metod. Naopak ČEZ, a.s., vydal interaktivní naučnou aplikaci o jaderné elektrárně Temelín [ČE16], ta podobně jako tato práce míří do sféry populárně naučné. Vizualizace ale postrádá větší utilizaci ilustračních prvků, které by vjem obrazové stránky aplikace zefektivnily.



## Kapitola 2

### Jaderná elektrárna

V této kapitole se věnuji teoretickým základům fungování jaderné elektrárny, které jsou nezbytné pro pochopení potřeb vizualizace. Jména komponent uvedené v této kapitole budou občasné dále zmiňovány v textu.

#### 2.1 Princip fungování jaderné elektrárny

Podstatou jaderné energetiky je štěpení prvků těžkých kovů, jako jsou například izotopy uranu. Štěpná reakce je silně exotermická (uvolňuje velké množství tepla). Vzniklé teplo je pak dále potřeba transportovat z oblasti reaktoru do oblasti elektrického generátoru, kde je tepelná energie přeměněna na energii elektrickou. Odvodem tepla se také reaktor chladí. [ČE16]

Na základě typu chlazení, paliva a moderátoru<sup>1</sup> se pak rozlišuje několik typů reaktorů. V této práci se budeme výhradně zabývat jedním z nejrozšířenějších typů - takzvaným reaktorem tlakovodním. Ten se vyznačuje tím, že jako moderátor využívá vodu v kapalném skupenství. Vzhledem k vysokým teplotám je potřeba v oběžním vodním okruhu udržovat vysoký tlak, aby nedocházelo k varu a odparu moderátoru. [ČE16]

Jak už bylo řečeno, teplo vzniká při štěpné reakci v reaktoru. Z něho je pak teplo odváděno moderátorem do oblasti parogenerátoru, který slouží k výměně tepla mezi vodou v primárním a vodou v sekundárním oběhu. Z parogenerátoru pak ochlazený moderátor proudí přes cirkulační čerpadlo zpět do nádoby reaktoru. Systém je rozdělen na dva okruhy právě z toho důvodu, aby bylo možné v primárním okruhu udržet vysoký tlak a zachovat tak efektivitu chlazení reaktoru. V sekundárním okruhu již voda není držena pod tlakem, a tak může být teplem odpařena.

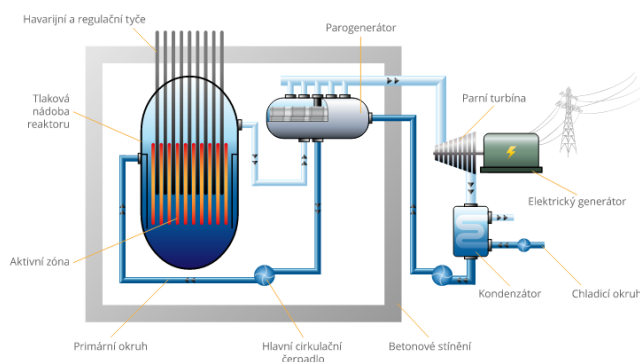
Sekundární okruh začíná v parogenerátoru, kde je sekundární voda odpařena. Horká pára pak proudí do parních turbín, které jsou napojeny na elektrický generátor. Později je pak pára zpátky zkapalněna v kondenzátoru a vedena zpět do parogenerátoru, kde se celý cyklus opakuje.

Celý oběh dobře ilustruje obrázek 2.1. Cirkulačních primárních okruhů může být na reaktor připojeno více. V modelu, který vznikne v souvislosti

---

<sup>1</sup>Moderátorem nazýváme látku, která transportuje vzniklé teplo do dalších částí elektrárny. Obvykle se používá jako moderátor například obyčejná voda. Moderátor může plnit zároveň i funkci chladiwa. [ČE16]

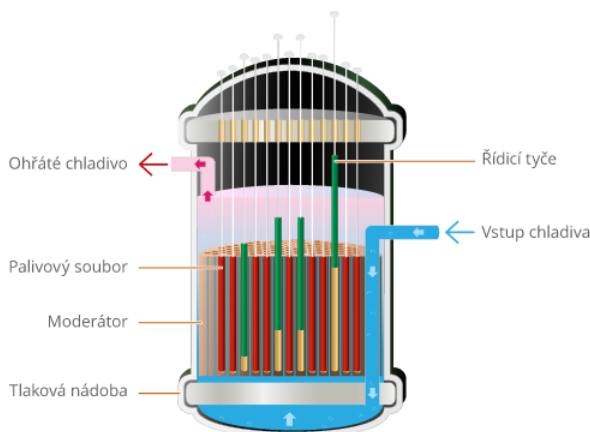
s touto prací, budou celkem čtyři cirkulační smyčky.



**Obrázek 2.1:** Schéma tlakovodního generátoru (převzato z [ČE16])

### 2.1.1 Reaktor

Reaktorem rozumíme nádobu, ve které probíhá štěpná reakce. Samotný reaktor se pak dělí na několik částí. Studený moderátor vstupuje do reaktoru ze strany, dále pak proudí po okraji nádoby směrem dolů. Dole se jeho směr změní a skrze děrované dno pak moderátor proudí přes takzvanou aktivní zónu. Aktivní zóna je oblast, ve které se nachází palivové články a regulační tyče, které štěpnou reakci řídí - zasouvají se mezi palivové články a tlumí řetězové štěpení mezi palivovými tyčemi. V aktivní zóně se moderátor ohřeje vzniklým teplem reakce a nad aktivní zónou pak ohřátý moderátor proudí z nádoby ven. Jednotlivé součástky dobře ilustruje obrázek 2.2.



**Obrázek 2.2:** Schéma reaktoru (převzato z [ČE16])

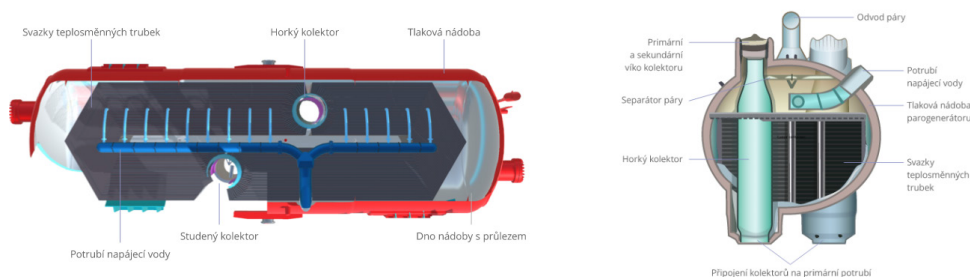
### 2.1.2 Parogenerátor

Součástka parogenerátoru řídí tepelnou výměnu mezi ohřátou primární a sekundární vodou. Primární voda vstupuje horkým kolektorem a dále

prochází svazkem mnoha teplosměnných trubek, které jsou zvnějšku oplachovány vodou sekundární. Ze svazku pak primární voda proudí do studeného kolektoru, kde z parogenerátoru odtéká.

Sekundární voda zvenčí volně oplachuje teplosměnné trubky a za intenzivního varu se odpařuje. Vzniklá pára je pak odvodnými trubkami vedena do dalších částí sekundárního okruhu.

Vnitřní uspořádání si můžete prohlédnout na obrázku 2.3.



**Obrázek 2.3:** Půdorys a bokorys parogenerátoru (převzato z [ČE16])

### 2.1.3 Cirkulační čerpadlo

Cirkulační čerpadlo je do primárního oběhu zařazeno mezi výstupem z parogenerátoru a vstupem do reaktoru. Jeho úkolem je udržovat vhodný průtok chladicího média smyčkou. Jeho konstrukce je poměrně složitá, protože je potřeba eliminovat riziko úniku kapaliny primárního okruhu ze smyčky (chladiivo se kontaktem s palivovými tyčemi stává radioaktivním).

### 2.1.4 Kompenzátor objemu

V primárním okruhu je chladicí voda vedena pod velmi vysokým tlakem, aby nedocházelo k jejímu vypařování. Vzhledem k teplotní roztažnosti však nemůže být smyčka napevno uzavřená, protože celkový objem chladiva v okruhu se stále mění na základě aktuálních hodnot teploty. Kompenzátor objemu je součástka, která se stará o řízení tlaku v okruhu. Nádobu kompenzátoru objemu pracuje na bázi rovnováhy parního polštáře a vodní hladiny. Pokud se v okruhu tlak zvýší, začne přesycená pára kondenzovat, čímž opět vyrovná požadovanou tlakovou hladinu. Pokud se naopak tlak ve smyčce sníží, začne se voda v kompenzátoru objemu odpařovat a tlak parního polštáře pak celkový tlak v oběhu opět vyrovná [ČE16].





## Kapitola 3

### Analýza a návrh

V této kapitole postupně nastíním účel vizualizace, ze kterého vyplynou požadavky softwaru. Na základě sestaveného seznamu požadavků zanalyzuji metody, které mi je pomohou co nejlépe naplnit.

#### 3.1 Požadavky softwaru

Vizualizace primárního okruhu jaderné elektrárny cílí na laickou veřejnost. Nejde tedy o uživatele problematiky znalé. Z tohoto důvodu je potřeba aplikací prezentovat spíše základní principy než detailní odborné informace. Program může běžet například v prezentačním kiosku v areálu jaderné elektrárny, kde může být k dispozici návštěvníkům před exkurzí.

Je nutné brát v potaz, že by měla aplikace uživatele upoutat a prezentovat informace záživným způsobem. Klíčem k upoutání pozornosti uživatele pak může být důraz na interaktivitu a obrazový výstup aplikace. Vzhled by však měl plnit také funkční roli, aby informace o primárním okruhu vhodně podpořil, nikoli zastínil.

Řešením bude vytvoření aplikace ovládané jedním uživatelem, který bude mít možnost volně prozkoumávat model primárního okruhu elektrárny. Aplikace bude poskytovat vedle vizuální informace také informace verbální (psané či mluvené).

Aplikace by měla:

1. být interaktivní, aby byl uživatel schopen směřovat svou pozornost na zvolené části primárního okruhu,
2. běžet v reálném čase (předpokládá se použití FullHD rozlišení), aby udržela pozornost uživatele a podtrhla tak interaktivitu aplikace,
3. názorně zobrazovat všechny důležité komponenty primárního okruhu a části, ze kterých se skládají,
4. vhodně reprodukovat tvar, strukturu a umístění komponent,
5. zobrazovat cirkulační děje v primárním okruhu,
6. být snadno ovladatelná, aby s ní mohl interagovat i uživatel, který není tak zblhlý v práci na počítači.

## 3.2 Interaktivita

Samotný požadavek interaktivity jde ruku v ruce s během aplikace v reálném čase. Bude důležité zvolit metody, které budou z pohledu výkonnosti vhodné. Měřítkem může být i paralelizace, protože existují například vizualizační techniky, které lze snadno implementovat na grafických kartách.

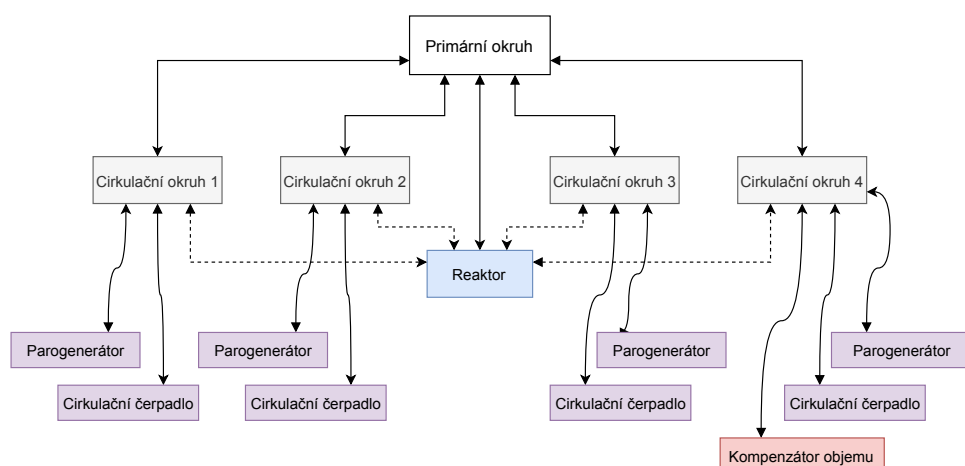
Obrazový výstup aplikace také nemusí být počítán v plném rozlišení (FullHD). Celý obraz (nebo jeho část) může být počítán v nižším rozlišení a namapován na původní velikost obrazu. Protože nebude ze strany uživatele potřeba rychle reagovat na žádné náhlé události, může se snímková frekvence pohybovat kolem třiceti hertzů, aniž by tím zhoršila ovladatelnost aplikace.

## 3.3 Navigace v modelu

Pracujeme-li s přiměřeným množstvím dat, můžeme je bez problému zobrazit všechna najednou. Pokud však zobrazujeme komplexní a rozsáhlé systémy, nemusí být už tento přístup vůbec efektivní, ba naopak hrozí nebezpečí, že zobrazením přílišného množství dat pozorovatele zahltneme, čímž mu čtení dat ztížíme.

Ben Shneiderman [Shn96] ve svém článku uvádí velice důležitou „mantru“ vizualizace: Overview first, zoom and filter, then details-on-demand (nejdříve přehled, přiblížení a filtrování, až nakonec detaily na vyžádání). V první řadě je potřeba pozorovateli ukázat zjednodušený přehled celého systému (v mém konkrétním případě oddálený pohled na celý primární okruh). Dále pozorovatel prostor filtruje na oblast, která ho zajímá (například oblast reaktoru). V neposlední řadě si pak vyžádá potřebné informace, které hledá (například konkrétnější informace o jedné součástce).

Model primárního okruhu je komplexní systém, který se skládá z mnoha funkčních celků. Ve spojitosti s vizualizační mantrou je tedy vhodné funkční celky od sebe oddělit a hierarchicky uspořádat. V mém případě bude model rozdělen do těchto celků manuálně. Existují publikace [LAS04], které popisují automatické rozdělení systému objektů na logické celky a automatizované vygenerování jejich hierarchické závislosti. V případě uvedeného zdroje se díly (například motoru) uspořádají podle jakési logiky posloupnosti montáže. Tento přístup je však pro náš případ užití vizualizace na primárním okruhu nepoužitelný a vzhledem k malé hloubce hierarchie také nepraktický.



**Obrázek 3.1:** Rozčlenění funkčních celků do hierarchické struktury. Každý funkční celek obsahuje svou oblast zájmu (detail modelu v aplikaci). Přejít mezi dvěma oblastmi zájmu je možné, jen pokud jsou přímo propojené

Ve chvíli, kdy je systém hierarchicky uspořádán, může uživatel volně přecházet mezi funkčními celky a hierarchií tak prozkoumávat. Navržené hierarchické rozdělení systému ilustruje diagram na obrázku 3.1. Kořenem je celý primární okruh. Ten se skládá ze čtyř cirkulačních smyček, z nichž každá se skládá z parogenerátoru a cirkulačního čerpadla. Čtvrtá cirkulační smyčka se od zbylých tří odlišuje tím, že zahrnuje i kompenzátor objemu. Uzel reaktoru se v hierarchii vymyká stromovému charakteru. Průtok chladiva každou smyčkou interaguje také s reaktorem. Zároveň je možné brát reaktor vedle cirkulačních smyček také jako samostatnou stavební jednotku primárního okruhu. Z tohoto důvodu je reaktor propojen také s každou ze čtyř cirkulačních smyček. Propojení mezi uzly pak vyznačuje možnou změnu oblasti zájmu (kterým se věnuje následující sekce). Bude-li se uživatel nacházet v první cirkulační smyčce, může přejít do oblasti reaktoru, parogenerátoru či cirkulačního čerpadla první smyčky, ale už nemůže přímo přejít do jiné cirkulační smyčky či jejích podřazených součástí (jako je třeba kompenzátor objemu).

### 3.3.1 Oblasti zájmu

Základní stavební jednotkou bude v primárním okruhu takzvaná oblast zájmu. Oblasti zájmu budeme nazývat detail, který bude disponovat vlastní úrovní konkrétnosti zobrazení. Pokud se budeme například nacházet v oblasti zájmu celého primárního okruhu, nebude potřeba popisovat vnitřní komponenty reaktoru, ale bude vhodné zobrazit alespoň plášť reaktoru a dalších okolních komponent, aby uživatel pochopil kontext celého systému.

V praxi není vhodné uživateli umožnit úplnou kontrolu nad kamerou. Často se v takových případech může stát, že uživatel nechtěně dostane kameru do takové pozice, kdy v záběru nemá žádný objekt. V této situaci je pak uživatel ztracen, protože nemá žádné vodítko, kterým směrem se s kamerou vydat, aby opětovně našel zobrazovaný model. Jednoduchým řešením může být fixace bodu, do kterého kamera vždy směřuje. Uživateli je následně povoleno

rotovat kolem tohoto bodu a kameru k němu přibližovat či od něho oddalovat. Umístím-li tento pevný střed pozornosti doprostřed zobrazované oblasti zájmu, mám zaručeno, že vždy bude v každém okamžiku zobrazena alespoň část modelu.

## 3.4 Vizualní informace

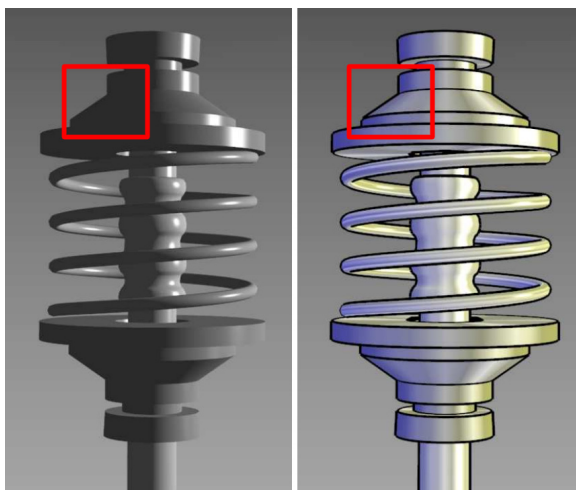
Silným médiem pro přenos informace je bezesporu obraz. Jeho správné použití pro prezentaci potřebných informací je stěžejní. Vizualní kanál je vhodný pro přenos informace o tvarech, umístění a vzájemné orientaci komponent primárního okruhu.

Zobrazovat modely realisticky nemusí být vždy nejlepší volbou. Díky tomu, že zobrazujeme virtuální scénu s objekty, si můžeme dovoluť vnést do obrazu i ilustrační prvky a obraz pomocí nich stylizovat. Můžeme zprůhledňovat povrchy, abstraktně je obarvovat, řezat či deformovat. Tato sekce se věnuje mimo jiné ilustračním postupům jasného vyobrazení prostorových modelů.

### 3.4.1 Reprodukce tvarů

Jednou z důležitých vlastností jasně čitelného obrazu je správná práce s tvary a siluetami.

K percepci tvarů nám může pomoci také vhodně zvolený osvětlovací model. Pokud například nasvítíme objekt jednobarevným ambientním světlem, zcela potlačíme jakýkoli vjem tvarů. Gooch a spol. ve své publikaci [GGSC99] například doporučuje takzvané cool-to-warm stínování, které používá k nasvícení modelu přechod od studených k teplým odstínům barev. Tento barevný přechod zvětšuje rozsah možných hodnot zastínění. Obrázek 3.2 dobře demonstruje efekt použitého stínování a porovnává ho s Phongovým osvětlovacím modelem.

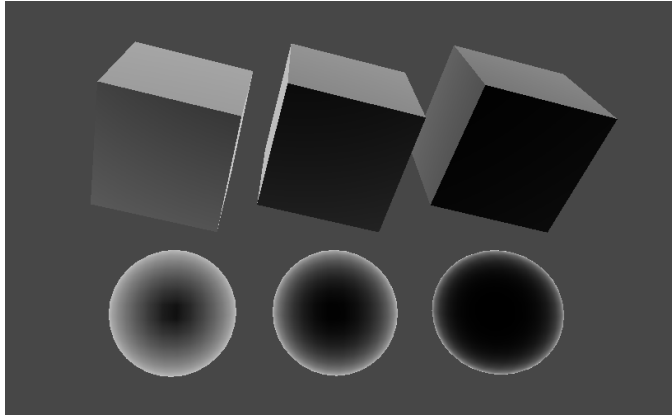


**Obrázek 3.2:** Demonstrace efektu cool-to-warm stínování. V červeném rámečku lze vidět dramatickou změnu ve zvýšení čitelnosti tvarů ve srovnání s Phongovým stínováním (Převzato z [GGSC99])

Pro zvýraznění siluet lze využít i funkce, které jsou závislé na pohledu kamery a normále zobrazovaného povrchu. Příkladem může být hodnota skalárního součinu mezi normalizovaným vektorem pohledu kamery  $\vec{V}$  a normálou  $\vec{N}$ :

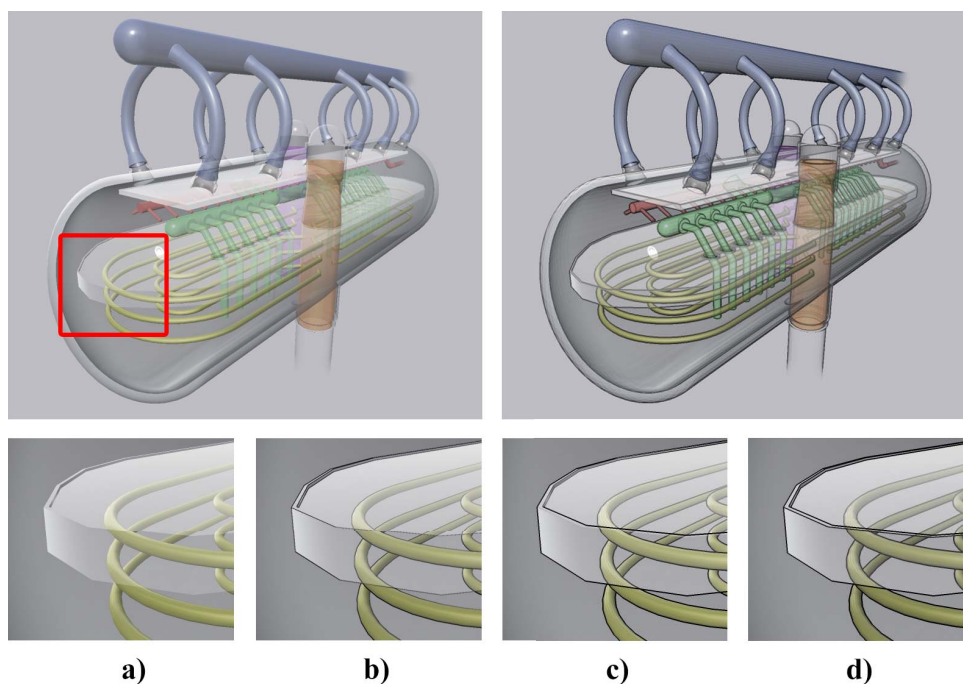
$$silueta = (1 - (\vec{V} \cdot \vec{N}))^p \quad (3.1)$$

Tato funkce vrací v absolutní hodnotě čísla v intervalu od 0 do 1. Umocněním výsledku mocnitelem  $p$  pak můžeme vhodně modulovat jemnost efektu. Vyšší mocniny generují ostřejší stínování siluet (viz obrázek 3.3). Lze však použít i komplexnější funkce, jako je například Schlickova aproximace Fresnelova efektu [Sch94].



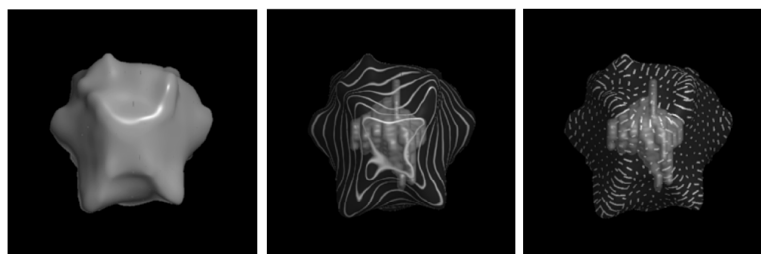
**Obrázek 3.3:** Zvýraznění siluet pomocí skalárního součinu normály povrchu a vektoru ke kameře. Umocnění výsledku generuje různě silný efekt (zleva hodnoty mocnin: 0.5, 1 a 2). Vyznačování siluet pomocí stínování je horší u hranatých objektů

Řízení nasvětlení a zvýrazňování siluet zmíněnými funkcemi se dobře chová k oblým a organickým tvarům (viz obrázek 3.3). Pro zvýraznění hranatých tvarů, které se často vizualizují například v motorech nebo jiných technických součástkách, však uvedené metody nefungují tak dobře. K tomu nám může pomoci detekce diskontinuit v hloubkovém bufferu, které dobře zvýrazní přechod mezi objektem a pozadím. K detekci hran v rámci jednoho celistvého objektu lze použít i normálový buffer. Kombinace obou metod naplňuje naše požadavky pro detekci důležitých hran v obraze. Implementaci detekce hran lze jednoduše uskutečnit konvolucí nad hloubkovým a normálovým bufferem. V praxi nám k detekci stačí u zpracovávaného fragmentu pouze hodnoty sousedních pixelů, díky tomu je konvoluční jádro o svém rozměru 3krát 3 pixely malé, a není tedy potřeba konvoluci nikterak akcelarovat, aby běžela v reálném čase. Zvýraznění hran za pomoci diskontinuity hloubky, normály či jejich kombinace ilustruje obrázek 3.4.



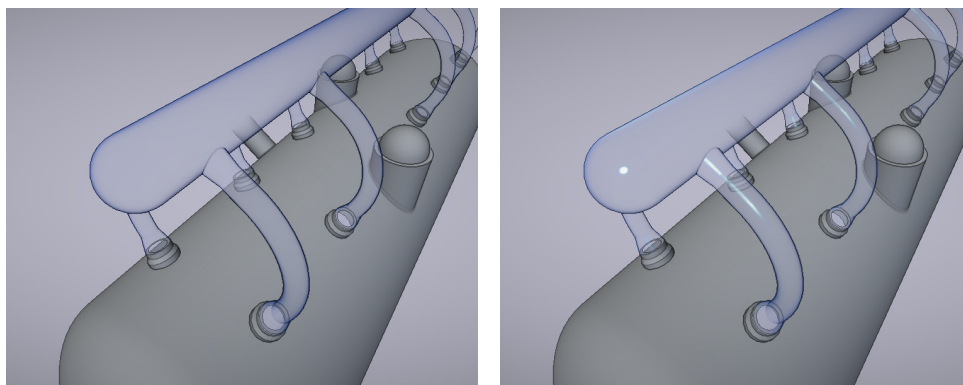
**Obrázek 3.4:** Efekt zvýraznění hran u oblých a hranatých objektů. Horní dva snímky ilustrují parogenerátor bez použití detekce hran (nalevo) a se zvýrazněním hran (napravo). Spodní čtyři výřezy jsou detailem oblasti vyznačené červeným rámečkem. *a)* zobrazuje stav bez jakéhokoli zvýraznění hran, *b)* zobrazuje detekci hran na základě diskontinuity hloubky, *c)* demonstruje detekci diskontinuity v bufferu normál a konečně *d)* zachycuje kombinaci detekce hran na základě normál i hloubky. Za zmínku stojí fakt, že teprve poslední výřez správně zvýrazní obě hrany vystouplého okraje světle šedého plechu

Problematickou skupinou jsou také objekty poloprůhledné, u nichž klesá výraznost tvaru objektu, pokud se zvyšuje jejich průhlednost. Techniky založené na stínování začínají být pro zobrazení takových objektů nedostačující. Interrante ve své publikaci [IFP96] tvarovou charakteristiku objektu zvýrazňuje méně průhlednými nebo zcela neprůhlednými čarami, které kopírují charakter průhledného povrchu. Výsledný efekt ilustruje obrázek 3.5. Metoda sice dobře reprodukuje tvary obalu, ve výsledné vizualizaci technického rázu, jako je tomu v našem případě, však může působit rušivě.



**Obrázek 3.5:** Ukázka zvýraznění křivosti povrchu průhledných těles pomocí neprůhledných linek v textuře poloprůhledného materiálu (převzato z [IFP96])

Dalším efektem, který napomáhá uživateli k percepci tvaru zprůhledněných ploch, mohou být spekulární odrazy na lesklých površích. Takové zvýraznění působí velmi přirozeně. Jejich výhodou je, že pokud zakrývají nějakou část, kterou chce uživatel vidět, drobným posunem kamery posune i tento odraz. Dopad spekulárních odrazů na percepci tvaru si lze prohlédnout na obrázku 3.6.



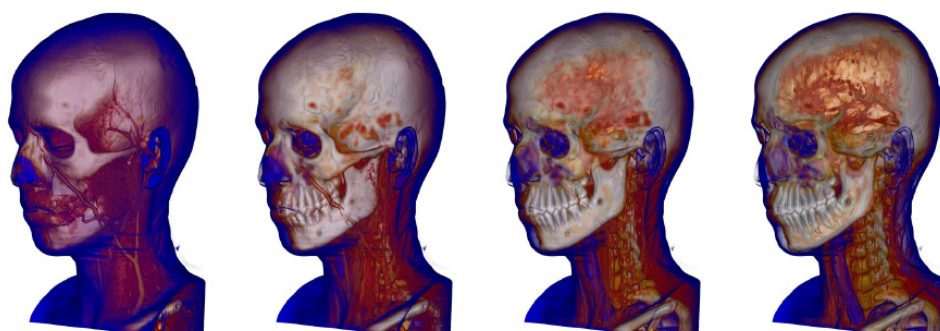
**Obrázek 3.6:** Zvýraznění tvarů oblých poloprůhledných povrchů dosaženého pomocí spekulárních odrazů. Nalevo je zachycen sběrač páry u parogenerátoru bez odrazů, napravo s odrazy

### 3.4.2 Reprodukce struktur

Zobrazování kontextu v trojrozměrném prostoru může být obtížné. Pokud například chceme zobrazit vnitřní součástky automobilu, vykreslení vnější karoserie jako kontextu je problematické, protože její vyobrazení by všechny vnitřní součástky zakrylo. Existují však techniky, které nám i takové vykreslení kontextu umožní. Důležitým krokem je rozdělit vizualizovanou scénu na části, které jsou pro nás technicky zajímavé (v příkladu s autem to může být třeba motor), a na části, které napomáhají k percepci kontextu (karoserie). Cílem je vykreslit kontext tak, aby nezastiňoval důležité části.

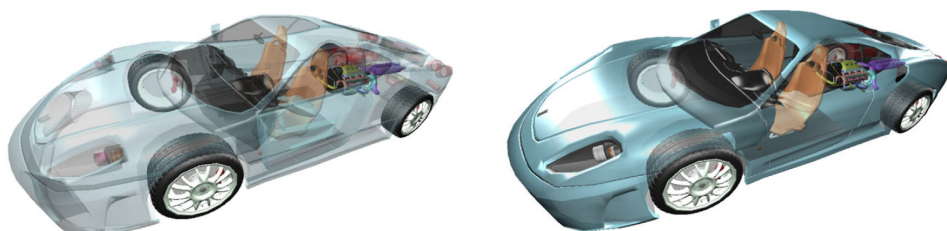
Jednou z možností je modulace alfa kanálu ploch. Karoserie se vykreslí s určitou úrovní průhlednosti, takže vnitřní uspořádání sice překryje, ale viditelné zůstanou. Bruckner ve svém článku [BGKG05] řídí průhlednost v závislosti na orientaci bodu obalu vzhledem k pozici kamery. Funkce je velmi podobná té, kterou jsme použili v předchozí kapitole pro zvýraznění siluet. Výsledek si můžeme prohlédnout na obrázku 3.7. Siluety zpravidla zaujmají okraje poloprůhledných obalů. Jejich menší průhlednost tedy výrazně nezhoršuje viditelnost vnitřních součástí a zároveň napomáhá k lepší percepci tvarů obalu.





**Obrázek 3.7:** Ukázka řešení kontextu obalu volumetrických dat (převzato z [BGKG05])

Velice povedenou metodu popisuje Pinto a Freidas ve své publikaci [PF11], ve které zavádí vedle alfa kanálu další veličinu, pomocí níž řídí způsob sestavování pixelů z poloprůhledných vrstev v obraze. Skladba poloprůhledných vrstev je založena na alfa kanálu a hodnotě důležitosti v daném pixelu v rozmezí od 0 do 1, kde vyšší hodnota značí důležitější objekt. Díky tomu jsme schopni zviditelnit i objekty, které leží za neprůhlednými bariérami, pokud dosáhnou vyšší důležitosti než popředí. Výhodou této metody je také to, že ji autoři demonstrovali nejen na volumetrických datech, ale také na polygonálních modelech. Jedná se o poměrně nenáročnou metodu, jejímu zakomponování do vizualizace tedy nestojí nic v cestě. Veličinu důležitosti lze také vhodně kombinovat s již uvedenými algoritmy z předchozí kapitoly. Můžeme například zvýšit důležitost hran či siluet. Obrázek 3.8 demonstruje rozdíl mezi klasickým použitím alfa kanálu a popisované metody s využitím hodnot důležitosti.

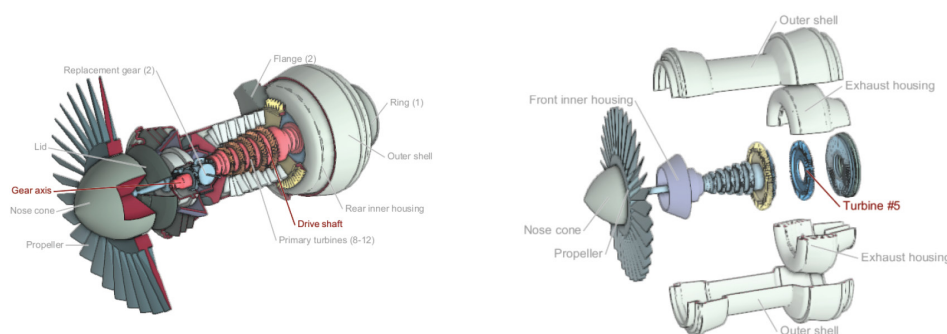


**Obrázek 3.8:** Prolínání poloprůhledných objektů pomocí alfa kanálu (nalevo) a kombinací alfa kanálu a hodnoty důležitosti (napravo). Vhodným nastavením hodnot důležitostí lze dosáhnout daleko čitelnějších výsledků (převzato z [PF11])

Další možností, jak zobrazit součástky, ze kterých je složen komplexnější objekt, je takzvaný exploded view. V tomto případě se nechá objekt rozpadnout na součástky, které mohou následovat například směr montáže. Metodou se zabýval třeba Li a spol. ve své publikaci [LACS08]. V mém případě, kdy zobrazuji primární okruh, však metoda příliš vhodná není. Tuto metodu bychom mohli využít například pro zobrazení reaktoru, ten by ale v rozloženém stavu zaujímal příliš velký prostor a naopak by se svým množstvím součástek čitelnost vizualizace zhoršil.



Zviditelnit vnitřní komponenty bez použití poloprůhlednosti lze i pomocí řezů. Plášť objektu je ve vhodném místě zcela odstraněn (uřezán) a nestíní tak vnitřní strukturám. Mírný problém může nastat, pokud nabídneme uživateli větší volnost v pohybu kamery. V takovém případě je vhodné řez pláštěm generovat automaticky, aby byly vnitřní komponenty viditelné v jakémkoli natočení pohledu. Dynamicky generuje řez například Li a spol. ve své publikaci [LRA<sup>+</sup>07]. Dalším možným řešením je připravit několik řezů, které pak lze přepínat v závislosti na pohledu uživatele. Metody řezů a exploded view si lze prohlédnout na obrázku 3.9.



**Obrázek 3.9:** Metody exploded-view (napravo) a řezy objektem (nalevo) jakožto rozdílné přístupy k zobrazení vnitřních struktur celků složených z mnoha komponent (převzato z [LRA<sup>+</sup>07] a [LACS08])

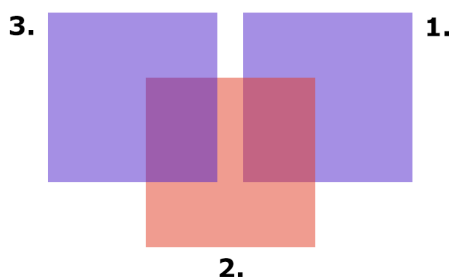
### 3.4.3 Zvýraznění dalších informací

Kromě tvarů a struktur nesou důležitou informaci také souvislosti mezi funkčními celky, jejich vzájemná pozice či orientace. Pohybujeme-li se kupříkladu v oblasti zájmu reaktoru, je vhodné naznačit, že součástka není plně izolovaný celek, ale souvisí i s dalšími součástkami v systému, například s parogenerátorem. Ten může být zobrazen méně výrazně, protože není v oblasti zájmu. Vzdálenější objekty lze také ponořit do mlhy, a tak snížit jejich výraznost, aby nenarušovaly zvolenou oblast zájmu.

Souvislosti mezi funkčními celky často tvoří hierarchii, kterou lze propojit s význačnými oblastmi zájmu z kapitoly 3.3.1. Existují publikace metod, které tyto hierarchie umí generovat automaticky (jak již bylo nastíněno v kapitole 3.3), jejich použití je ale nad rámec diplomové práce a svou povahou nevhodné pro systém primárního okruhu. Veškeré hierarchie a souvislosti budou tedy modelovány ručně.

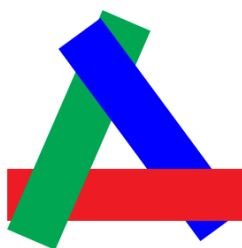
### 3.4.4 Depth-peeling

V souvislosti s popisovanými metodami pro zvýraznění kontextu se dostáváme k problematice vykreslování poloprůhledných objektů. Při kompozici poloprůhledných pixelů pomocí alfa kanálu je důležité respektovat jejich správné pořadí. Jak ukazuje obrázek 3.10, různě seřazené poloprůhledné plochy generují rozdílnou barvu složených pixelů.



**Obrázek 3.10:** Skladba poloprůhledných pixelů v rozdílném pořadí generuje rozdílné výsledky

Pokud vykreslujeme scénu metodou vrhání paprsku, nevzniká nám z tohoto pohledu žádný problém. V našem případě však požadujeme běh v reálném čase, proto bude potřeba použít vykreslování pomocí rasterizace. Při prolínání poloprůhledných ploch však záleží na pořadí. V teoretické rovině jsme sice schopni seřadit trojúhelníky podle vzdálenosti jejich středu od kamery, jak to obdobně dělá například malířův algoritmus, ale i tak mohou vzniknout situace, které jsou seřazením neřešitelné (viz obrázek 3.11).



**Obrázek 3.11:** Příklad situace, kdy seřazením tří objektů nelze vyřešit správné prolínání poloprůhledných ploch

Everitt popisuje metodu takzvaného depth-peeling [Eve01], kdy je scéna postupně vykreslena několika průchody do vrstev, které se na konci prolnou. Díky tomu můžeme dosáhnout správného pořadí prolínání na úrovni pixelu. Metoda je použitelná pro rasterizační vykreslovací přístup a je robustní, na rozdíl od řazení vykreslovaných trojúhelníků.

Algoritmus začíná prvním průchodem, kdy je scéna klasicky vykreslena bez použití průhlednosti, tím vznikne první obrazová vrstva. Vygenerovaný hloubkový buffer pak slouží v dalším průchodu k zahození fragmentů o stejné nebo menší hloubce, tím je vykreslena druhá vrstva. Rekursivním opakováním pak lze oddělit všechny vrstvy obrazu. Vykreslování vrstev je zastaveno ve chvíli, kdy posledním průchodem nebyl vykreslen žádný fragment.

Problémem algoritmu je vyšší výpočetní i paměťová náročnost, která roste s počtem vrstev, které je potřeba pro danou scénu vykreslit. Čím více objektů se směrem od kamery překrývá, tím více průchodů si algoritmus vyžádá. V praxi je však možné zvolit maximální počet vrstev.

Počet vrstev, které je potřeba k vykreslení celé scény lze snížit i jiným způsobem. Zobrazujeme-li nekonvexní těleso, může jeho vykreslení zasáhnout ve stejném pixelu do více vrstev. V praxi tomu však lze předejít. Vhodným způsobem, jak si držet informaci o vykreslených objektech, je bitová maska v každém pixelu obrazovky. Vykreslovaný objekt pak nese unikátní id, kterému náleží příslušný bit v masce. Buffer, který masku nese, může využít jednoho kanálu barevné textury (podpora až 32 objektů) nebo lze masku rozdělit do více kanálů s podporou až 128 bitů (možno vést záznam až o 128 objektech).

### ■ 3.4.5 Zvolené metody pro reprodukci vizuální informace

Pro zvýraznění siluet bude použita funkce popsaná vzorcem 3.1. Hranaté tvary budou zvýrazněny pomocí detekce hran, která bude realizována konvolucí nad normálami a hloubkou. K zobrazení struktur použijí průhlednost dle článku autorů Pinto a Freidas [PF11]. U reaktoru bude navíc použit jeden statický řez středem nádoby.

## ■ 3.5 Verbální informace

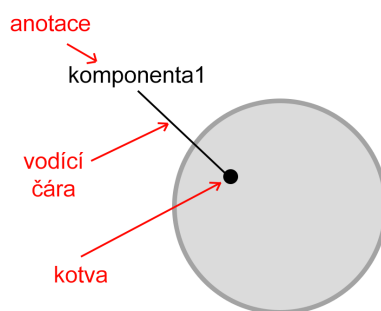
Ačkoliv je obraz pro člověka silným vjemovým kanálem, není možné přes něj efektivně některé informace přenést. Vizuálním kanálem jsme například schopni sdělit informaci o velikosti, barvě, natočení či vzájemném postavení komponent, jejich funkci či název však často nelze zcela jasně vysvětlit. Naopak pouze verbální popis komponent silně spoléhá na schopnost uživatele představit si popisované objekty. Je tedy vhodné, aby šly vizuální a verbální informace ruku v ruce a vzájemně se doplňovaly.

Verbální informace může mít podobu textu nebo zvuku. Vhodným pojátkem mezi obrazem a textovou informací jsou dynamicky umístěné popisky komponent.

Pro svou aplikaci jsem zvolil jako verbální informaci psaný text, ve kterém budou vyznačeny odborné termíny částí zobrazené komponenty.

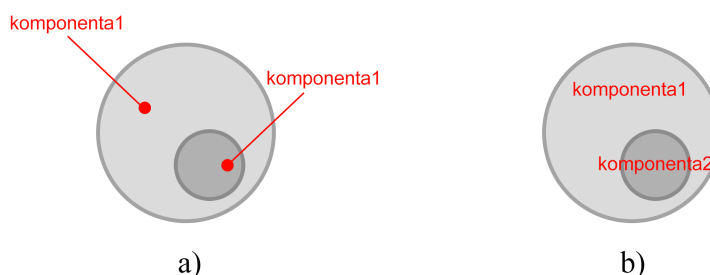
### ■ 3.5.1 Popisky a jejich umístování

Anotace částí zobrazovaného modelu je ve vizualizaci poměrně rozsáhlá a netriviální disciplína. Možnosti rozmístování popisků se dotýkají mnoha různých konstrukčních algoritmů a geometrických problémů. Popisek se může skládat celkem ze tří komponent: kotvy, vodící čáry a anotačního textu. Kotva popisku je umístěna na popisované komponentě, vodící čára pak spojuje anotační text s kotvou. Uvedené části popisků jsou ilustrovány na obrázku 3.12. Podle typu umístění anotačního textu dělíme typy popisků na interní a externí.



**Obrázek 3.12:** Ukázka částí, ze kterých se skládá typický popis

Interní popisky umísťují anotační text přímo na popisovaný objekt. Z tohoto důvodu není zapotřebí ani kotvy, ani vodící čáry. Ukázku interních a externích popisků si lze prohlédnout na obrázku 3.13.



**Obrázek 3.13:** Ukázka externích popisků (a) a interních popisků (b)

Externí popisky mají naopak anotační text umístěn mimo popisovanou komponentu. Text je umístěn optimálně tak, aby nezakrýval anotovaný objekt. Nejvhodnější pozice se tedy nachází mimo projekci objektu z pohledu kamery. U vodících čar je velmi důležité, aby se mezi sebou neprotínaly. Rovněž je z pochopitelných důvodů nežádoucí, aby se mezi sebou překrývaly také anotace.

Existuje celá řada publikací, které se věnují umísťování popisků, jež jsou interního nebo externího typu. Existují dokonce zdroje, které uvádějí kombinované užití jak externích, tak interních popisků zároveň. Při hledání vhodného algoritmu pro tuto práci hraje roli hlavně výpočetní náročnost, protože vyžadujeme běh v reálném čase. Dalším důležitým měřítkem je takzvaná časová koherence popisků, tedy že při pohybu kamery popisky neskáčou po obraze, ale mění svou pozici plynule. V neposlední řadě hraje roli také náročnost implementace, jelikož textové popisky nejsou hlavní oblastí zájmu této diplomové práce.

V následujících sekcích popíši pouze metody, které splňují popsání kritéria jednoduché implementace, běhu v reálném čase a dobré časové koherence.

## Kombinované metody

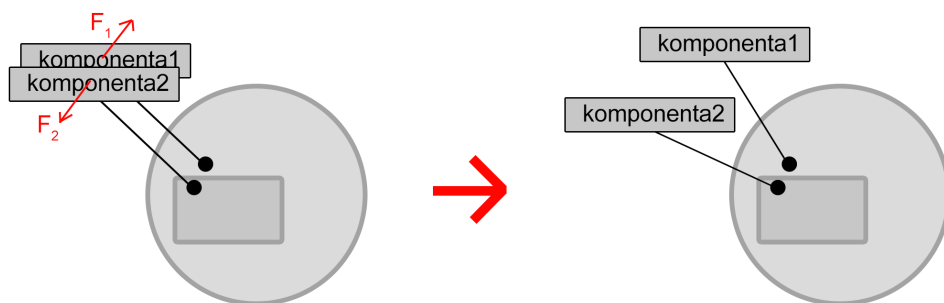
Kombinované metody umísťování popisků využívají jak interních, tak externích anotací. Götzelmann a spol. ve své publikaci [GAHS05] popisují způsob takového umísťování. Algoritmus pracuje v prostoru obrazu (projekci), ve kterém pomocí id mapy analyzuje vhodná místa pro pozici anotací. Samotné umísťování je pak řízeno heuristikami. Algoritmus má dobrou časovou složitost, jeho implementace na grafické kartě by však byla složitá.

## Interní popisky

Jak již bylo řečeno, interní popisky, na rozdíl od těch externích, postrádají kotvu, a tedy i vodící čáru. Text je pak umístěn přímo přes anotované objekty. Příkladem takového umísťování může být publikace autorů Kouřil, Čmolík a dalších [KmK<sup>+</sup>19], která interních popisků využívá pro anotaci chaotických a objekty přeplněných scén, jako mohou být například modely buněčných struktur. Metoda umísťuje popisky hierarchicky. Jestliže je popisovaná skupina objektů v projekci příliš malá, jejich popis zmizí. Algoritmus nabízí také slušnou časovou koherenci. Pro naše potřeby, kdy jsou součástky primárního okruhu jasně definované a jejich počet je poměrně malý, však nemá použití této metody příliš velký smysl.

## Externí popisky - silově založený přístup

Hartmann, Strothotte a Ali ve své publikaci [HAS04] používají k umístění popisků síly působící v prostoru obrazu. Silové pole je tvořeno jak samotnými texty popisků, tak také projekcemi objektů. Tento přístup je v základu poměrně jednoduchý, jeho časová koherence však není nejlepší. Při větších pohybech kamery se mohou popisky poměrně dlouho pohybovat, dokud nedokonvergují do rovnovážného stavu. Během pohybu pak mohou být popisky matoucí a jejich čitelnost zhoršená. Ve zmíněné publikaci také vznikaly situace, kdy se vodící čáry překrývaly. Modelovou situaci, kdy působení sil mezi anotačními texty popisků vyřeší jejich překrytí si lze prohlédnout na obrázku 3.14.



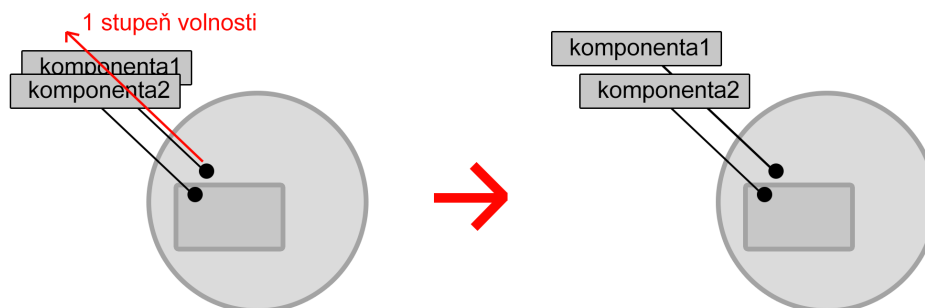
**Obrázek 3.14:** Silově založené rozmísťování popisků. Nalevo je případ, kdy se anotační texty překrývají. Působí na sebe odpudivými silami, které postupně dovedou anotační texty do pozic, ve kterých k překrytí nedochází

## ■ Externí popisky - hedgehog labeling

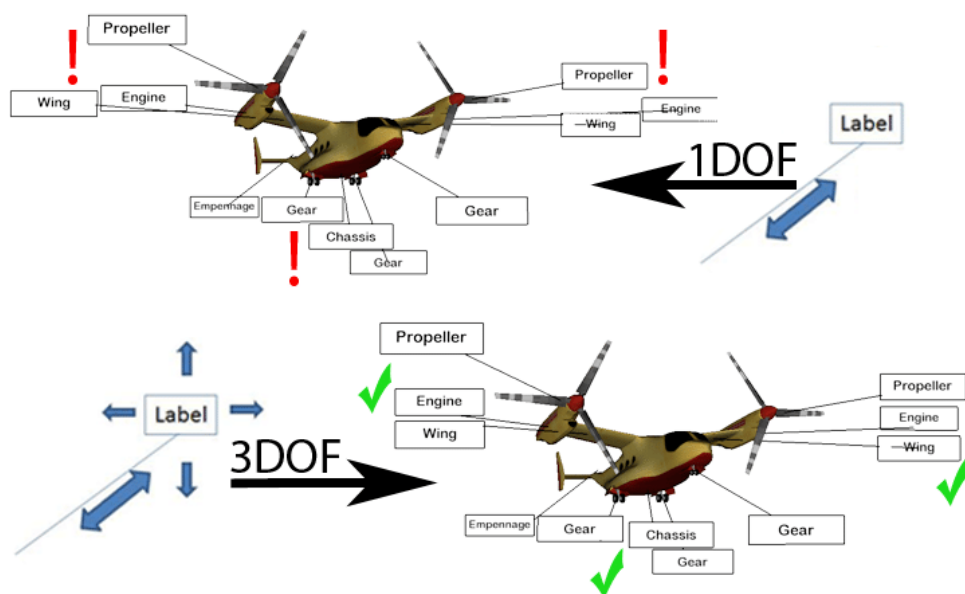
Příkladem poměrně jednoduchého algoritmu, který umísťování popisků řeší, je takzvaný hedgehog labeling, který pracuje s popisky umístěnými přímo v prostoru scény. Způsob umísťování popsali Tatzgern a další ve své publikaci [TKGS14]. Základem popisku je jeho kotva a směr vodicí čáry. Kotva je umístěna na pevně dané pozici a směr vodicí čáry je vhodně určen tak, aby nekřížil jiné čáry. Toho se dá jednoduše docílit například tím, že vodicí čáry leží na přímkce mezi společným definovaným středem a svou kotvou. Tento způsob určení směru čar se výborně chová v jakémkoli natočení kamery, ale nemusí být zcela vhodný pro všechny tvary anotovaných objektů. Příkladem mohou být podlouhlé objekty, u kterých by měly vodicí čáry optimálně směřovat k jejich ose.

Metoda hedgehog labeling provádí samotné umístění popisků ve dvou fázích. V první fázi dojde k umístění kotvy, vodicí čáry a anotace do výchozí pozice tak, aby byla anotace umístěna mimo projekci objektu. V druhé fázi se pak řeší kolize mezi anotačními texty.

Hedgehog labeling zavádí několik stupňů volnosti, kterými lze řešit kolize anotačních textů. Tím nejjednodušším je prodloužení konce vodicí čáry do prostoru, kde anotace nekoliduje s jinou (použití ilustruje obrázek 3.15). Tento přístup řeší kolize velice efektivně, v určitých případech však kupí anotace za sebou směrem od středu objektu, čímž zhoršuje percepce příslušnosti popisku k popisovanému objektu. Z tohoto důvodu algoritmus zavádí další dva stupně volnosti. Anotacímu textu je povoleno se ve své rovině pohnout libovolným směrem, ale pouze tak, aby nedošlo k odpojení anotace od vodicí čáry. Pohyb je tedy omezen na polovinu rozměru anotace v každé ose. Použití si lze prohlédnout na obrázku 3.16.



**Obrázek 3.15:** Vyřešení překryvu anotací pomocí jediného stupně volnosti (pohyb podél vodicí čáry)



**Obrázek 3.16:** Řešení kolizí anotačních textů pomocí jednoho stupně volnosti pohybu podél vodící čáry (nahore) a pomocí dalších dvou stupňů volnosti pohybu anotačního textu (dole) (Převzato z [TKGS14])

Popsaná metoda umísťování popisků se také dobře chová v proměnlivém prostředí, protože kontinuita pozice popisku vzhledem k pohybu kamery je mnohem lepší než kupříkladu metody postavené na sílovém poli. Díky dvěma fázím algoritmu je však možné pozastavit řešení kolizních situací během pohybu kamery, dokud se pozice pohledu neustálí. Riskujeme sice, že se během pohybu začnou anotace překrývat, zvýšíme však plynulost změny pozice samotných popisků.

### ■ Nejednoznačnost kotev externích popisků

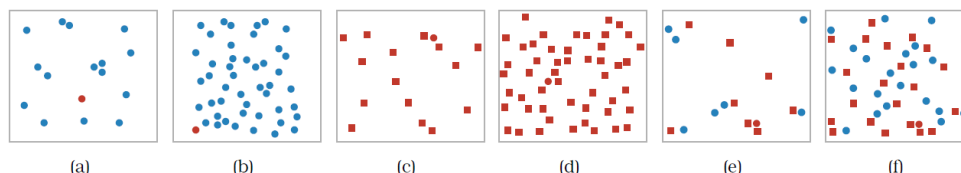
Popisování prostorových modelů, které se skládají z mnoha poloprůhledných vrstev, je problémový případ, protože uživateli nemusí být jasné, na kterém objektu kotva leží. Z tohoto důvodu je potřeba najít způsob, jak umožnit uživateli tuto víceznačnost vyřešit.

Vizuální systém člověka má unikátní schopnosti zpracovávat data o vysoké kvantitě. Náš mozek zpracovává mnoho vizuálních vjemů paralelně, díky tomu je schopen takto velké množství informací pojmout. Vizualizace využívá právě zrakový informační kanál, protože například sluch či jiné smysly nedokáží přenést stejně široké pásmo dat [Mun14].

Ve vizualizaci se můžeme setkat například s pojmem vizuální popup (nebo též preemptivní zpracování). To je efekt, kdy jsme schopni za zlomek sekundy rozhodnout, zda se například v zobrazované množině dat vyskytují prvky, které vystupují z řady. Tuto informaci jsme schopni zpracovat okamžitě, aniž bychom se snažili nad obrazem hluboce přemýšlet [Mun14].

Obrázky 3.17 znázorňují několik případů, kdy se efekt preempce dostaví. Můžeme vidět, že odlišení významných prvků pomocí barvy nebo tvaru nám

velmi zjednodušuje jejich nalezení v prezentovaném data setu. Pokud však tyto kanály začneme kombinovat, efekt velmi zeslabíme, nebo dokonce úplně odstraníme.

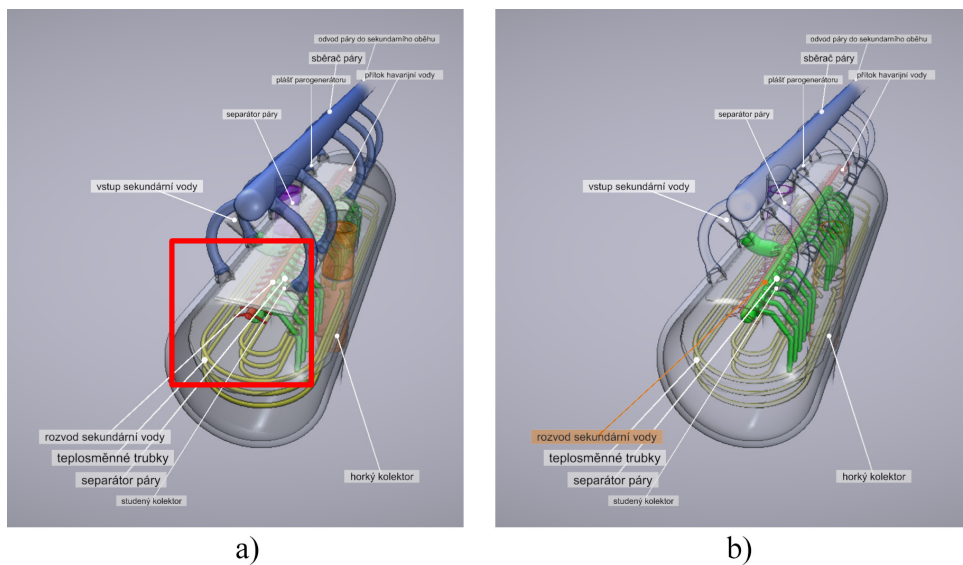


**Obrázek 3.17:** Na obrázcích A a B okamžitě vidíme, že červené prvky vystupují z řady. Stejný efekt se projeví i u obrázků C a D, u kterých si můžeme všimnout tvarových outlierů. Kombinace tvarů a barev u obrázků E a F však již efekt značně zeslabují (převzato z [Mun14])

Z uvedených příkladů je patrné, že existují kanály, které je možné lidským zrakem velmi dobře a efektivně zpracovat. Kromě barvy a tvarů jsou to například odlišení pomocí náklonu, velikosti, směru stínu nebo těsnějšího seskupení více prvků v uniformní distribuci [Mun14]. Mezi nepreemptivní kanály řadíme mimo jiné text. Efektivita vizualizace silně závisí na kvalitě využití možných kanálů a jejich kombinací. Často můžeme jen kombinací dvou preemptivních kanálů vytvořit nepreemptivní vizualizaci, jako tomu bylo například na obrázku 3.17. Je proto důležité se při návrhu vizualizace takovým případům co nejvíce vyhýbat.

A právě efektu vizuálního popupu budeme chtít využít, když bude potřeba zvýraznit konkrétní součástku. Nejsnáze efektu docílíme tak, že silně odlišíme zvýrazněnou komponentu od zbytku zobrazovaných objektů. Vhodně se jeví například zvýšení průhlednosti komponenty a zároveň snížení důležitosti a průhlednosti ostatních částí. Ostatní komponenty tedy začnou splývat s pozadím. Efekt pak aktivujeme při najetí kurzoru na textovou anotaci popisku nebo na zvýrazněný termín v textu, který je s popiskem ve scéně svázán. Dosažený vysoký kontrast mezi zvýrazněnou komponentou a okolím, které splývá s pozadím, vytvoří popisovaný popup efekt. Demonstraci popsáního problému a jeho řešení si lze prohlédnout na obrázku 3.18.





**Obrázek 3.18:** Ukázka situace, kdy není jednoznačné, ke které komponentě kotva popisku patří (a). Silný efekt zvýraznění po najetí na popisek víceznačnost kotev řeší (b)

### 3.5.2 Zvolené metody pro reprodukci verbální informace

Nositelem verbální informace bude psaný text, který bude slovně popisovat funkce zaměřené komponenty. V textu budou zvýrazněny odborné názvy dílčích částí zobrazované komponenty. Části komponent budou popsány externími popisky, které budou umístěny přímo ve scéně pomocí přístupu hedgehog labeling [TKGS14]. Odborné názvy v textu budou akčně propojeny s popisky ve scéně.

## 3.6 Zobrazení cirkulačních dějů

Abychom mohli navrhnout efektivní vizualizaci pro daná data, musíme jejich povaze nejdříve porozumět. Například vizualizace, která bude vhodná pro zobrazení teploty v různých městech České republiky, nemusí být vhodná pro vizualizaci nejrozšířenější značky autobusů, které v daných městech jezdí.

V základu rozdělujeme atributy na [Mun14]:

- kategoričké - nedají se seřadit (například druhy ovoce),
- ordinální - dají se seřadit, ale nedefinují rozestupy mezi jednotlivými typy (například velikosti oblečení - S, M, L, XL...),
- kvantitativní - dají se řadit a jsou jasně definované rozdíly mezi typy (například hodnoty teplot na různých místech).

Ordinální a kvantitativní data pak můžeme ještě charakterizovat způsobem, jakým se řadí [Mun14]:

- sekvenční - mají definované minimum (například délka),
- divergující - mají definovanou nulu a mohou nabývat záporných hodnot (například teplota ve stupních Celsia),
- cyklická - hodnoty se cyklicky opakují (například hodnoty úhlů).

V neposlední řadě pak mohou data nabývat různé dimenzionality [Tel14]:

- skalární - atribut obsahuje jednu hodnotu (například teplota),
- vektorová - atribut obsahuje 2 a více hodnot (například směr větru).

Výše uvedené rozdělení je pouze náhledem do základní klasifikace dat. Pro potřeby této práce byl však výčet z důvodu přehlednosti zkrácen.

Cirkulace chladiva v systému spadá do dat vektorových. Je však vhodné kromě směru cirkulace také vizualizovat teplotu chladiva, protože právě přenos tepelné energie z místa jejího vzniku do dalších komponent je hlavním účelem primárního okruhu. Teplotu lze klasifikovat jako skalární veličinu. Bude tedy potřeba zobrazovat kombinaci vektorových a skalárních hodnot.

Z pohledu vizualizace průtoku využijeme směr vektorových dat. Velikost vektorů v našem případě odpovídá rychlosti toku, z pohledu laické veřejnosti ale tato informace není důležitá, proto budu velikost vektorů dále ignorovat.

### 3.6.1 Vizualizace skalárních dat

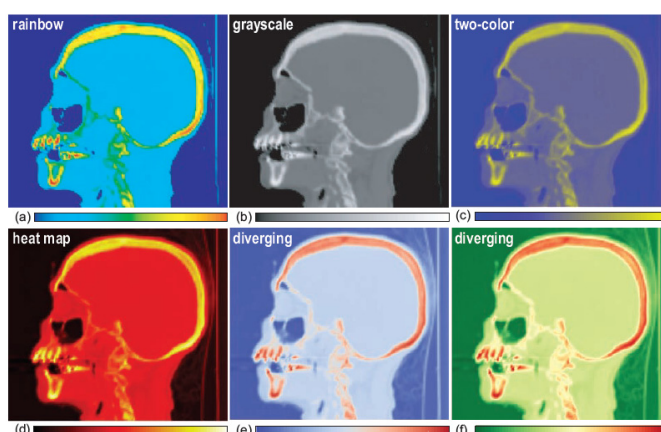
Nejvhodnějším způsobem, který se pro vizualizaci teploty jakožto skalární veličiny nabízí, je mapování hodnot na barvu. Prvním krokem je zvolení vhodné barevné stupnice, těch je několik typů.

Prvním typem je sekvenční stupnice, která nám mapuje určitý rozsah hodnot zcela lineárně. Příklad takové stupnice můžete vidět na obrázku 3.19.

Druhým typem je takzvaná divergující stupnice. Ta se vyznačuje zejména tím, že v ní existuje význačný bod, který můžeme vnímat jako nulu. Ukázku divergující stupnice můžeme vidět na obrázku 3.19. Divergující stupnice jsou vhodné zejména pro vizualizaci divergujících atributů (například teplota ve stupních Celsia). Použijeme ji tehdy, pokud nás zajímá směr od význačného bodu (například jestli mrzne, nebo je teplota nad nulou).

Posledním typem stupnice je stupnice cyklická. V takové stupnici nemůžeme přesně určit, které hodnoty jsou největší a které nejmenší. Příklad cyklické stupnice najdete na obrázku 3.19. Pomocí cyklické stupnice lze například zobrazovat čas během dne na mapě světa.

Teplota ve stupních Celsia je se svým význačným bodem  $0^{\circ}\text{C}$  divergující veličinou. Z pohledu použití barevné stupnice však také záleží na tom, jakou informaci uživateli chceme vizualizovat. V mém případě bychom chtěli odlišit oblasti s chladným a teplým moderátorem, zajímají nás tedy relativní rozdíly mezi hodnotami, nikoliv kladné a záporné hodnoty. Teplotní hodnoty v primárním okruhu se navíc pohybují řádově ve stovkách stupňů Celsia, proto použití divergující stupnice nedává smysl. Pro vyznačení teplotních hodnot použiji jednoduchou lineární stupnici.



**Obrázek 3.19:** Ukázka různých barevných stupnic. B,C,D jsou stupnice sekvenční; E,F jsou stupnice divergující; A je stupnice cyklická (převzato z [Tel14])

### ■ Barevné prostory a percepční uniformita

Barvy jsou v oboru vizualizace velmi užitečný nástroj. Teorie, která se barev týká, však rozhodně není triviální. V této sekci si proto pouze nastíníme nejnutnější minimum, které budeme k práci potřebovat.

Lidské oko používá pro percepci světla čípky a tyčinky. Tyčinky jsou velmi citlivé a využíváme jich obecně za nedostatečných světelných podmínek. Tyčinky nám však umožňují vidět jen monochromatický obraz. Naopak čípky nejsou tak citlivé jako tyčinky. Využíváme jich za normálních světelných podmínek. V oku máme celkem 3 druhy čípků, každý z nich je nejcitlivější na jinou vlnovou délku světla. Díky čípkům dokážeme vidět barvy [Mun14].

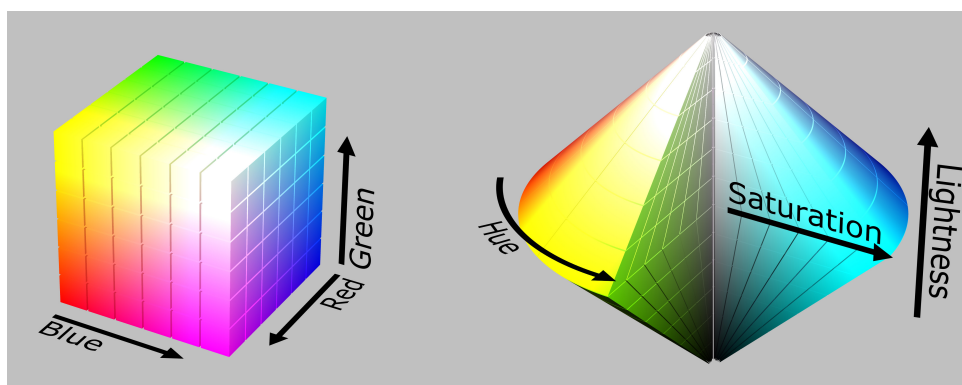
Abychom byli schopni reprodukovat barvy, zavádíme takzvané barevné prostory. Ty jsou definovány převážně alespoň třemi souřadnicemi (paralela ke třem druhům očních čípků). Způsob definice takového prostoru však může mít dopad na práci s barvami.

Barevný prostor RGB patří mezi nejznámější. Zkratka RGB vznikla ze složek, které barevný prostor tvoří: červená (red), modrá (blue) a zelená (green). Různou kombinací těchto složek dokážeme namíchat mnoho různých barev. Ačkoliv je tento přístup výhodný z pohledu výroby hardwaru (například pixely monitorů běžně obsahují tři složky uvedených základních barev), není příliš vhodný pro naše intuitivní chápání. Pokud například uvidíme dva barevné vektory prostoru RGB v číselné podobě, nejsme schopni určit, která z barev bude světlejší.

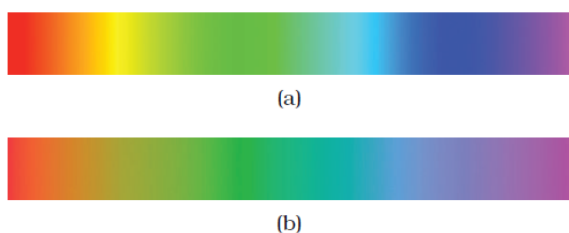
Naopak barevný prostor HSL je pro lidské chápání percepce barev mnohem jednodušší. Prostor charakterizuje barvu hodnotami pro 3 složky. Tentokrát se však odlišují od složek RGB: odstín (hue), sytost (saturation) a světlost (lightness). Barevné prostory RGB a HSL jsou ekvivalentní, ovšem jejich číselné vyjádření nám je mnohem bližší [Mun14]. Oproti prostoru RGB jsme již schopni v číselném tvaru bezpečně určit, která ze dvou barev bude světlejší, protože je hodnota světlosti oddělena v samostatném kanálu. Další výhodu přináší například z pohledu interpolace barev. Pokud bychom chtěli

například lineárně interpolovat ze žluté barvy do modré, uprostřed bychom v barevném prostoru RGB dostali šedou barvu. Naopak v prostoru HSL si během interpolace barva ponechá jak správnou světlost, tak i sytost. Měnit se pak bude pouze odstín. V praxi se můžeme často setkat s tím, že pracujeme v prostoru HSL a do prostoru RGB barvy transformujeme, až když chceme obraz vykreslit. Obrázek 3.20 ukazuje, jak si můžeme představit rozložení prostorů RGB a HSL.

Ačkoliv nám prostory RGB a HSL umožňují přesně definovat velké množství barev, stále mají své nedostatky. Z pohledu vizualizace je takzvaná percepční uniformita poměrně důležitá vlastnost. Představme si barevnou stupnici v prostoru HSL, která drží konstantní hodnoty sytosti a světlosti. Hodnota odstínu se bude postupně měnit v konstantních inkrementech. Ukázku vzniklé stupnice si můžeme prohlédnout na obrázku 3.21 (a). Vzniklá stupnice je však takzvaně percepčně neuniformní, protože například zóna žluté barvy se nám jeví jako mnohem menší než interval zelené barvy. Pokud bychom touto barevnou stupnicí vizualizovali číselný skalární atribut, mohli bychom chybně vnímat rozdíl mezi dvěma hodnotami zelené jako menší než rozdíl mezi hodnotou žluté a oranžové barvy.



**Obrázek 3.20:** Ukázka barevných prostorů RGB a HSL (převzato z [Sha08])



**Obrázek 3.21:** Ukázka percepčně neuniformní a percepčně uniformní duhové barevné stupnice (převzato z [Mun14])

Pokud bychom tedy chtěli použít duhovou barevnou stupnici, museli bychom ji vyjádřit v percepčně uniformním barevném prostoru [Mun14]. Příkladem percepčně uniformního prostoru je Lab.  $L$  označuje světlost,  $a$  je přechod od červené k zelené a konečně osa  $b$  označuje přechod mezi žlutou a modrou.

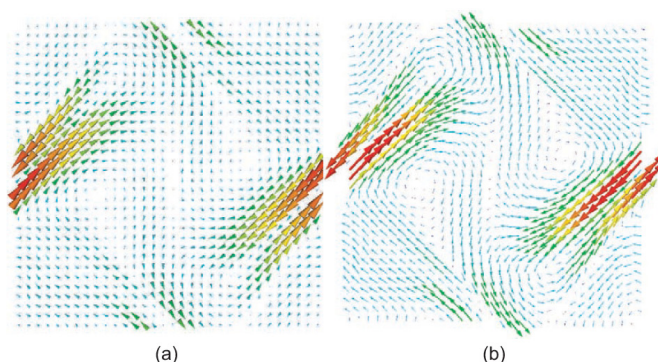
Druhým příkladem percepčně uniformního prostoru je HCL, který poměrně úzce souvisí s Lab. Jeho složka  $L$  je shodná se stejnojmennou složkou v prostoru Lab. Složky  $a$  a  $b$  však charakterizuje v jejich polárních souřadnicích jako  $H$  a  $C$ .

### 3.6.2 Vizualizace vektorových dat

U vektorových veličin, v našem případě konkrétně u průtoku, nás zajímá hlavně směr a délka vektoru. Směr lze vizualizovat více způsoby, tím nejzajímavějším z pohledu této práce je použití glyphů.

#### Glyphy

Glyphy si můžeme představit jako značky v prostoru, které jsou schopny vyjádřit hodnotu směru v daném bodě. Příkladem glyphů jsou úsečky, šipky a jejich obdoby. Informaci můžeme prezentovat u glyphů například jejich velikostí, délkou, barvou, orientací a podobně. Ukázku takové vizualizace můžeme vidět na obrázku 3.22, kde je vektorové pole vyznačeno pomocí glyphů ve tvaru kužele a šipek [Tel14].



**Obrázek 3.22:** Vizualizace vektorového pole pomocí glyphů tvaru kužele (A) a šipek (B) (převzato z [Tel14])

Výhodou glyphů je právě obsah informací, které dokážou nést. Jsou velice vhodné také pro kombinace více veličin. V našem konkrétním případě můžeme vizualizovat podle směru šipek nejen průtok, ale pomocí barvy glyphů i teplotu v daném bodě. Jsou tedy nejlepším řešením vizualizace cirkulačních dějů v primárním okruhu.

#### Data proměnná v čase

Mnoho statických vizualizací řeší proměnná data v čase různými triky. Můžeme kreslit trajektorie částic, které jsou vektorovým polem unášeny. Pracujeme-li však s interaktivní vizualizací, která běží v reálném čase, otevírají se nám dveře do světa animací, které mohou přispět k percepci pohybu. Je však potřeba zmínit, že ne v každé vizualizaci vektorového pole je pohyb vítaný. Pokud se v obraze pohybuje příliš mnoho objektů, může uživateli naopak

čtení informace ztížit. Z pohledu cirkulace chladiva v primárním okruhu je však proudění omezeno převážně na prostor v trubkách. Cílem navíc není uživateli reprodukovat víry, které mohou během toku v okruhu vznikat, ale spíše mu přiblížit obecný směr, kterým chladivo proudí. Z tohoto pohledu je tedy pohyb glyphů po proudu toku přínosným efektem.

### ■ 3.7 Souhrn zvolených technologií

Z projektu vyplynulo několik požadavků, ty budou naplněny konkrétními metodami, které byly zanalyzovány v této kapitole.

Model bude rozdělen do dílčích částí, které budou hierarchicky uspořádány. Uživateli bude umožněno mezi těmito dílčími částmi přecházet. Pohyb kamery bude omezen bodem, do kterého bude kamera vždy směřovat. Hierarchie objektů bude připravena manuálně.

V obrazu vizualizace použiji poloprůhledných ploch, díky kterým zviditelním vnitřní součástky celků. K prolínání použiji kromě alfa kanálu také hodnotu důležitosti. Pro zvýraznění siluet zvýším průhlednost na plochách, jejichž normály směřují blízko k pozici kamery, a naopak zneprůhledním ty plochy, které s pohledem kamery svírají téměř pravý úhel. K percepci tvaru použiji také detekci hran a spekulárních odlesků, které budou nezávislé na průhlednosti objektů. Ke zvýraznění vjemu hloubky se bude za poloprůhlednými plochami průhlednost postupně snižovat (vzdálenější objekty budou méně viditelné než objekty těsně za poloprůhlednou bariérou). Robustního systému prolínání bude dosaženo za pomoci metody depth-peeling.

Vizualizaci doplní textový popis funkcí a částí primárního okruhu. Součástky funkčních celků budou anotovány popisky, které budou umístěny dynamicky metodou hedgehog labeling.

Cirkulační děje v primárním okruhu budou zobrazovat glyphy, které se budou pohybovat ve směru toku. Barva glyphů bude reprodukovat teplotu chladiva pomocí lineární barevné stupnice.



## Kapitola 4

### Implementace

Požadavkem projektu je vytvoření 3D vizualizace, která bude sloužit pro výukové účely a poběží v reálném čase. Tyto požadavky jsou poměrně náročné, protože ne všechny vizualizační techniky lze efektivně používat v reálném čase. V této kapitole si přiblížíme nástroje, které jsem využil při realizaci projektu.

#### 4.1 Nástroje a frameworky

Postavit vlastní zobrazovací engine a implementovat si potřebné komponenty bylo pro tento projekt časově nerealizovatelné. Nabízí se však už hotová řešení v podobě herních engineů, které mají široké možnosti vykreslování, simulace fyziky nebo skriptování. Díky těmto engineům je možné přeskočit velkou část úmorného programování základních funkcí, jako jsou například: organizace scény, matematická komponenta pro výpočty lineární algebry, vykreslovací modul, načítání shaderů a podobně; a pustit se rovnou do implementace vizualizačních technik.

Na trhu je více herních engineů. Mezi ty nejrozšířenější běžně dostupné patří například Unity, Unreal Engine 4 nebo Cry Engine. O rozdílech mezi Unity a Unreal engineem pojednává práce [Šm17]. Obecně je však Unity považováno za jednodušší engine pro začátečníky a nabízí dostatečně široké možnosti. Unreal engine naopak dosahuje vyššího výkonu u rozsáhlých scén. Cry Engine je zase velmi silný nástroj pro vykreslování v reálném čase, jeho architektura je však velice složitá a ne tak dobře dokumentovaná jako již zmiňované protějšky. Pro tuto práci jsem si vybral Unity engine (verze 2018.2.6f1), se kterým mám již základní zkušenosti a který bude pro realizaci projektu bohatě postačovat.

#### 4.2 Vytvoření modelu primárního okruhu

Veškeré součásti primárního okruhu jaderné elektrárny byly vytvořeny jako polygonální modely. Při tvorbě bylo využito 3D studio max firmy Autodesk (verze 2017).

Vzhledem k tomu, že je vizualizace pojata více ilustračně nežli realisticky, je možné modely jednotlivých součástí primárního okruhu zjednodušit a vy-

nechat drobné detaily, jako jsou šrouby, rýhy ve spojích a podobně. Blok teplosměnných trubek byl rovněž pojat schematicky: místo tisíců drobných trubiček byla vymodelována pouze trojice trub s širšími průměry.

Díky zjednodušení objektů nebylo potřeba připravovat ani normálové mapy, které aproximují nerovnosti povrchů modelů. Z podobného důvodu nebylo potřeba ani vytvořit UV mapovací souřadnice pro textury (až na výjimky, které se týkají efektu mizejících trubek, které vedly mimo zobrazovaný systém).

Aby bylo možné model zobrazovat v reálném čase, bylo potřeba udržet množství trojúhelníků v rozumné míře. K výpočetně nejnáročnějším patřilo zobrazování samotného reaktoru. Ani v jeho kontextu však počet vykreslovaných trojúhelníků nepřesáhl tří set tisíc. Díky šetrnému přístupu k modelování jsem tedy nemusel připravovat ani zjednodušené verze modelů ve spojitosti s použitím technologie level of detail (LOD), která zobrazuje různé složité úrovně objektů podle vzdálenosti od kamery, čímž zrychluje vykreslování.

## 4.3 Použití Unity engineu

Pro implementaci skriptů v prostředí Unity engineu se používá jazyk C#. Unity nabízí pro skriptování poměrně bohaté a dobře dokumentované API.

Shadery jsou implementovány v jazyce CG. Unity disponuje také celou řadou užitečných funkcí, které se týkají nasvícení ze světelných zdrojů, aproximace ambientního nasvícení či různé konverzní funkce pro kódování datových typů float o vysoké bitové hloubce do vektorových typů a opačně. Použití těchto funkcí usnadnilo a urychlilo samotnou implementaci.

Skriptovací systém umožňuje vhodně propojit grafické rozhraní Unity editoru s naprogramovanými funkcemi. Každému hernímu objektu ve scéně lze přiřadit libovolný počet skriptů. Skripty, které je možné přiřadit k herním objektům, běžně implementují třídu *MonoBehaviour*, která nese základní členské funkce *Start()* a *Update()*, které Unity automaticky volá v rámci herní smyčky. Označením členských proměnných skriptů jako *public* dojde také k jejich zviditelnění v rámci grafického rozhraní v editoru Unity. To mi umožnilo jednoduše ladit hodnoty těchto členských proměnných přímo v editoru.

### 4.3.1 Převzetí kontroly nad vykreslováním

Unity engine disponuje v základu již připraveným řešením vykreslování. Díky skriptovacímu API jsme však schopni do tohoto daného řetězce částečně zasáhnout.

Abych byl schopen implementovat všechny zmíněné ilustrační techniky z předchozí kapitoly, bylo potřeba v první řadě realizovat techniku depth-peeling, která je pro celou práci stěžejní.

Základním vstupním bodem je call-back metoda *OnRenderImage()*, která je volána po vykreslení scény výchozím zobrazovacím řetězcem. Metoda je přetížena v hlavním skriptu připojeném přímo ke kameře.

```
void OnRenderImage(RenderTexture src, RenderTexture dst){
```



```
...
}
```

Parametrem funkce jsou 2 objekty typu *RenderTexture*. První z těchto textur nese scénu vykreslenou výchozím zobrazovacím řetězcem. Druhá textura pak plní úlohu výstupu. Metoda *OnRenderImage()* je vhodná například pro implementaci post-processing efektů. V našem případě však můžeme vstupní obrázek zahodit a do výstupní textury zapsat obraz, který bude vykreslen nezávisle na vstupu.

Abych byl schopen volat vykreslovací příkazy, je potřeba získat referenci na objekt kamery. Ta pak obsahuje pro nás důležité členské funkce *SetTargetBuffers()* a *RenderWithShader()*. První z nich nastavuje cílové buffery pro vykreslení, druhá pak přímo volá samotné vykreslování z pohledu zvolené kamery. Unity API nabízí pro vykreslení scény více verzí zobrazovacích funkcí, *RenderWithShader()* je však speciální v tom, že vykreslí každý objekt mnou specifikovaným shaderem. Stěžejní však je volat vykreslovací příkazy na kameře, která je deaktivována. Pokud bychom volali zmíněné metody na aktivovaném objektu kamery, začal by se rovněž volat call-back *OnRenderImage()* a skript by pravděpodobně havaroval.

V poslední řadě jsem využil funkci *Graphics.Blit()*, které umí překopírovat obsah jednoho bufferu do druhého a volitelně při této operaci aplikovat na každý pixel efekt definovaným shaderem.

```
void Blit(Texture src, RenderTexture dst, Material mat);
```

Prvním parametrem je vstupní textura, druhým je textura výstupní. Třetím parametrem je tentokrát instance materiálu, která byla vytvořena daným postprocess shaderem. Díky této funkci jsme schopni prolínat více vrstev bufferů do jednoho. Stačí nám v definovaném materiálu připojit na vstup další texture a v shaderu k ní přistoupit a použít ji do výpočtu výstupní hodnoty pixelu. Problém nastává ve chvíli, kdy vyžadují na výstupu více než jeden buffer. K těmto účelům jsem implementoval funkci *MultiTargetBlit()*, která vykreslí obdélník přes celou obrazovku a evokuje tím spuštění fragment shaderu na každý pixel. Díky tomu bylo možné nastavit jako cíl vykreslování více bufferů za použití funkce *SetTargetBuffers()*.

### 4.3.2 Implementace metody depth-peeling

Podstatou této metody je vykreslit scénu do oddělených vrstev. V první vrstvě je scéna vykreslena běžným způsobem. Pro vykreslení druhé vrstvy se použije informace hloubkového bufferu první vrstvy. Díky tomu je možné detekovat fragmenty vykreslené v první vrstvě a zahodit je. Do druhé vrstvy se tedy vykreslí fragmenty následující za první vrstvou. Další vrstvy jsou vykreslovány analogicky vždy za použití hloubkového bufferu z předchozího kroku.

Vykreslování vrstev je prováděno směrem od pozorovatele tak dlouho, dokud vykreslování vrací alespoň nějaký fragment. Je tak možné vykreslování zastavit, protože další vrstvy by byly již také prázdné. V praxi se používá

k této detekci čítač fragmentů, které prošly zobrazovacím řetězcem. Přístup k těmto čítačům zprostředkovávají takzvané *occlusion queries*. Unity API však v době implementace této práce neposkytovalo žádné funkce, které by mi umožnily tyto nízkourovňové dotazy volat. Z tohoto důvodu jsem byl nucen počet vrstev empiricky určit a zafixovat.

Abych snížil počet vrstev, které je potřeba vykreslit, implementoval jsem metodu, která zaručí, že vykreslovaný objekt bude v daném pixelu obsažen pouze v jedné vrstvě. Pozitivem je, že touto změnou nedochází ke snížení srozumitelnosti vykresleného obrazu. V mnoha případech má spíše opačný efekt, protože uživatel není nucen vnímat více vrstev stejného objektu. Celý obraz se tak stane přehlednějším.

Popsaného efektu jsem docílil pomocí bitové masky, která hlídá, zda již nebyl v předchozích vrstvách vykreslen objekt stejného materiálového id. Implementoval jsem masku, která podporuje 128 rozdílných id. Práci s maskou a převody mezi vektorovými a skalárními typy si lze prohlédnout v souboru *OpPeel.shader* v metodách *idToVector()* a *clipId()*.

Metoda depth-peelingu je implementována pomocí dvou shaderů: *OpPeel.shader*, který řídí zpracování scény a oddělení vrstev, a *OpBlend.shader*, který se stará o následné skládání jednotlivých oddělených vrstev.

Velmi užitečné byly funkce *DecodeFloatRGBA()* a *EncodeFloatRGBA()*, které dokázaly převádět hodnoty 32bitového typu float na vektor o čtyřech složkách a zpět. Díky tomu jsem byl schopen uložit hloubkovou mapu do čtyř kanálů textury s osmi bity na kanál. Při depth-peelingu jsem pak informaci hloubky používal při oddělování jednotlivých vrstev obrazu. K tomu jsem také použil funkci *clip(float)*, která zahodila daný fragment, pokud se na vstupu metody objevila hodnota menší než 0.

Celou sekvenci volání vykreslovacích příkazů a nastavování cílových bufferů si lze prohlédnout v metodě *renderSceneInto()* v souboru *ManualRenderer.cs*.

### 4.3.3 Implementace popisků

K umístění popisků jsem využil přístupu hedgehog labeling, který je detailněji popsán v kapitole 3.5.1. Text popisku je nejlepší umístit tak, aby nepřekrýval anotovaný objekt. Toho jsem docílil tak, že jsem popisec umístil do kotvy a vrhl z jeho pozice přímkou směrem do kamery. Dokud přímka protínala nějaký objekt ve scéně, posunoval jsem popisec ve směru anotace o pevně definovaný krok. Díky tomu jsem se vyhnul implementaci algoritmů výpočetní geometrie, které by dokázaly vhodné místo najít analyticky.

K vyřešení kolizních situací jsem použil však jen jeden stupeň volnosti - pohyb anotačního textu ve směru vodící čáry, dokud nepřestal překrývat ostatní popisky. Také k tomuto jsem využil vrhání paprsků, které v základu nabízí Unity API. Během testování jsem si potvrdil, že anotací nebylo ve vizualizaci tolik, aby se začalo ve větší míře projevovat řetězové kupení popisků (problém byl detailněji popsán v kapitole 3.5.1).

Nad rámec původního algoritmu jsem implementaci doplnil o systém vyhlazení pohybu popisků při změně pozice kamery. Skript popisku sice vypočte

pravou pozici anotačního textu dle výše uvedeného klíče, bere však tuto pozici jen jako cíl. V každém snímku vizualizace je aktuální pozice anotace interpolována směrem k vytyčenému cíli. Díky tomu po uklidnění pohybu kamery pozice popisku plynule konverguje k vypočtenému cíli.

Hedhehog labeling má však svá úskalí. Je-li pozice kotvy popisku neměnná, můžeme nevhodným natočením kamery vytvořit situace, kdy není jasné, ke které části modelu kotva patří. Tento problém jsem částečně eliminoval tím, že jsem vytvořil efekt, který po najetí na text popisku zvýrazní anotovanou komponentu tím způsobem, že jí nastaví nejvyšší důležitost. Ostatním objektům naopak nastaví nulovou důležitost a změní jejich alfa kanál na hodnotu 0. U objektů s nulovým alfa kanálem ale stále zůstávají čitelné hrany a siluety. Změna je dost výrazná na to, aby docílila popup efektu, který je detailně popsán v kapitole 3.5.1.

Implementaci popisků si lze prohlédnout ve skriptech: *HedhehogLabel.cs* (hlavní skript popisků, který implementuje umístění a řešení kolizí), *LabelSprite.cs* (skript textové anotace a detektor událostí jako najetí myši na popisek) a ve skriptech *LabelFocusChange.cs* a *LabelGlobalCenter* (drobné implementace středového bodu, od kterého popisky směřují pryč, respektive call-back změny detailu při kliknutí na anotaci popisku).

#### ■ 4.3.4 Implementace pohyblivých glyphů

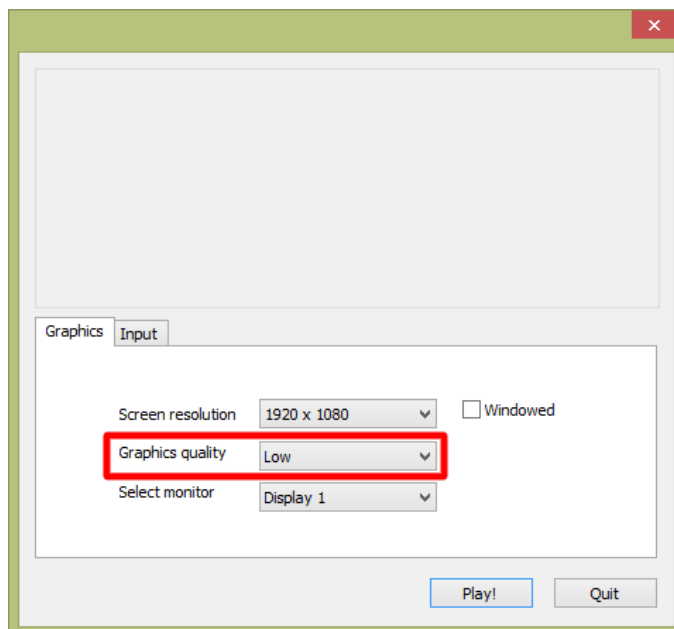
Pro zobrazení průtoků a teplot chladiva byly zvoleny glyphy ve tvaru šipek. Glyphy se pohybují po lomené trajektorii, která je vytyčena body. Body trajektorie jsou definovány jako prázdné objekty ve scéně se skriptem *GlyphVertex.cs*. Každému křivkovému bodu lze přiřadit barvu a velikost šipky. Glyph pohybující se podél trajektorie pak hodnoty velikosti a barvy respektuje lineární interpolací.

Objekty bodů trajektorie jsou seskupeny jako potomci jediného prázdného objektu, který nese hlavní skript *GlyphSystem.cs*. Tento hlavní skript se stará o vytvoření definovaného počtu glyphů podél lomené trajektorie, je také zodpovědný za správný pohyb glyphů v závislosti na definované rychlosti toku a taktéž přetěžuje Unity funkci *OnDrawGizmos()*. V této funkci je pak implementován systém náhledu hustoty, umístění a barvy glyphů. Díky tomu lze tyto hodnoty pohodlně nastavit přímo v Unity editoru.

#### ■ 4.3.5 Výkon aplikace

Ilustrační techniky ve spojitosti s depth-peelingem se ukázaly jako poměrně náročné, aplikace však běží i na starším hardwaru. Na své sestavě (mobilní grafická karta Nvidia GT650m, intel i5) jsem dosahoval snímkové frekvence přibližně 20 FPS. Abych byl schopen aplikaci provozovat s vyšší snímkovou frekvencí, zabudoval jsem do aplikace řízení detailu zobrazení. Nastavení nižšího detailu způsobí, že je pohled do prostorové scény vykreslen v nižším rozlišení a posléze je snímek zvětšen na cílovou velikost obrazovky. Abych tímto přístupem neovlivnil kvalitu textových popisků a textu obecně, vykresluji v nižším rozlišení pouze model. Stejně jako lze rozlišení snímku 3D scény

snížit, lze ho také na výkonnějších počítačích zvýšit, a tak dosáhnout hladšího zobrazení. Rozlišení snímku je svázáno s nastavením grafické kvality v úvodním okně spouštěné Unity aplikace (viz obrázek 4.1).



**Obrázek 4.1:** Nastavení grafické kvality řídí rozlišení bufferu, do kterého se vykresluje prostorová scéna. Textové popisky komponent a uživatelské rozhraní jsou vykresleny vždy v nativní kvalitě

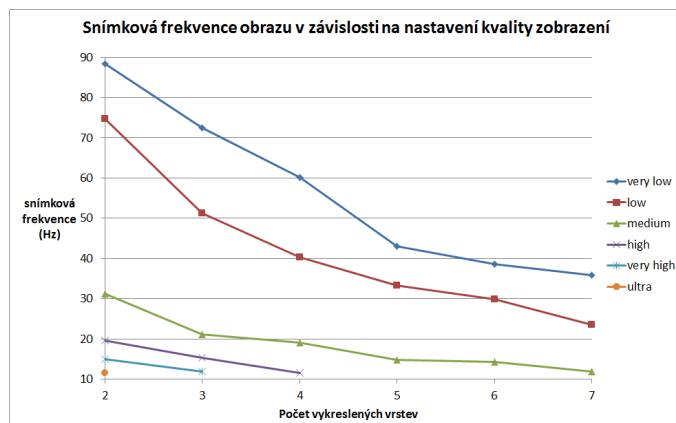
Multiplikátory rozlišení jsou s nastavením kvality grafiky svázány následovně:

- very low = 0.5 (poloviční rozlišení),
- low = 0.7,
- medium = 1.0 (nativní rozlišení),
- high = 1.3,
- very high = 1.5,
- ultra = 2.0 (dvojnásobné rozlišení).

Byl zdokumentován také vliv multiplikátoru rozlišení a počtu vykreslených vrstev na snímkovou frekvenci obrazu. Naměřené výsledky si lze prohlédnout v grafu 4.2. Hodnoty snímkových frekvencí byly naměřeny na sestavě těchto parametrů:

- procesor: i5-3210M (2,5 GHz),
- grafická karta: Nvidia 650M (2Gb GDDR5),

- množství RAM: 8 Gb,
- operační systém: Windows 8.1 64bit,
- monitor: fullHD (1920x1080).



**Obrázek 4.2:** Závislost snímkové frekvence na multiplikátoru rozlišení a počtu vykreslených vrstev metodou depth-peeling. Výsledky se vztahují k nativnímu rozlišení fullHD (1920x1080)

Snímkové frekvence byly měřeny v nejsložitější části vizualizace - pohledu na reaktor. Hraniční počet vrstev, kdy už nedocházelo k vykreslování dalších informací, se zastavil na hodnotě 5. Pro tuto hodnotu bylo dosaženo 30 a více hertzů snímkové frekvence jen za použití multiplikátoru rozlišení 0.5 a 0.7. Pro následný test s uživateli (více v kapitole 6) tedy bylo zvoleno nastavení pěti vykreslených vrstev s multiplikátorem 0.7.



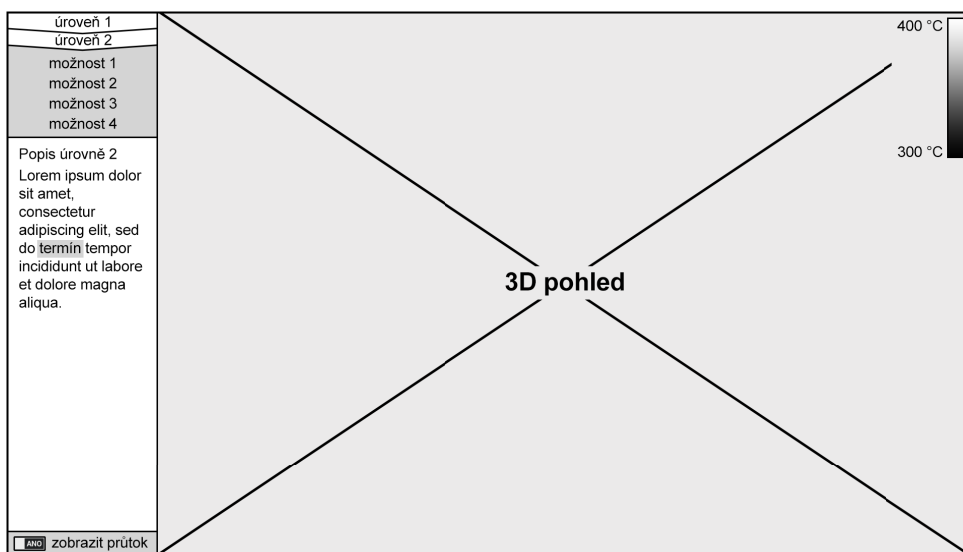
## Kapitola 5

### Výsledky

Výstupem práce je spustitelná aplikace v prostředí Microsoft Windows. Unity nabízí další možnosti sestavení projektu například pro platformy Linux, iOS a další, vyvíjena a testována však byla výhradně na verzi Windows 8.1 64bit.

#### 5.1 Rozvržení uživatelského rozhraní aplikace

Vizualizace je interaktivní a nabízí vedle prostorového náhledu na součástky primárního okruhu také navigační prvky nabídky menu, které jsou umístěny v levé liště. Pod menu se nachází doprovodný text, který se obměňuje na základě aktuálně zacíleného detailu. V textu jsou vyznačeny důležité termíny. Ve spodní části levé lišty je umístěn přepínač, kterým je možné přepínat módy zobrazení součástek a cirkulačních dějů. Návrh uživatelského rozhraní si lze prohlédnout na obrázku 5.1. Rozhraní je stavěno na rozlišení fullHD (1920x1080px), spuštěním aplikace v menších rozlišeních může být znemožněna viditelnost popisného textu v plné délce.

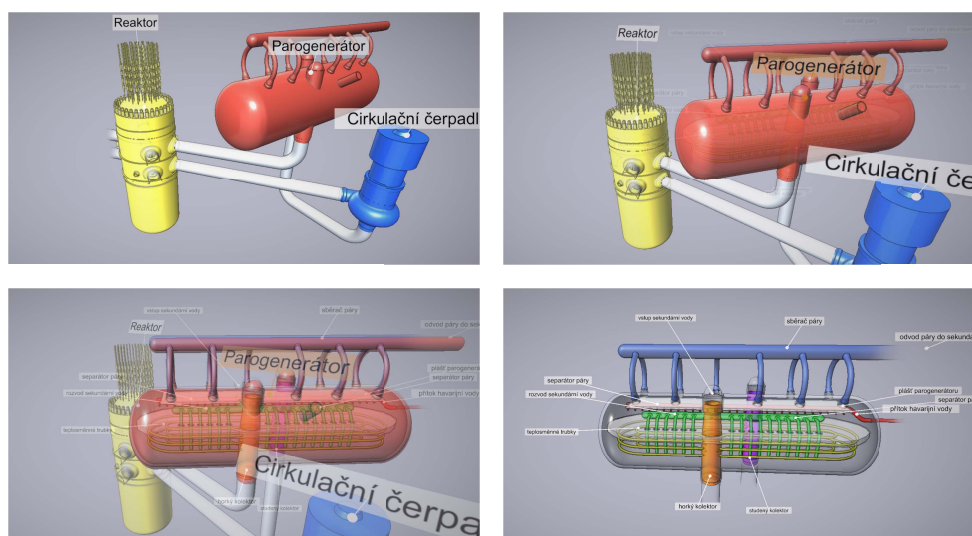


Obrázek 5.1: Rozvržení uživatelského rozhraní aplikace

## 5.2 Ovládání a navigace

Navigaci mezi kontexty jednotlivých součástí je možné provést více způsoby. Pokud je ve 3D pohledu viditelný popisek komponenty, která má vlastní kontext, je možné se na něj přepnout přímo kliknutím na název popisku. Druhým způsobem je navigace pomocí menu v levé liště, ze kterého je možné se pohybovat v hierarchii kontextů nejen směrem do hloubky, ale i směrem k nadřazeným komponentám. Uživatel tedy není nucen měnit kontext pomocí popisků, využívání levého navigačního menu je však pro některé změny nezbytné, neboť v popiscích chybí například odkazy na nadřazené komponenty. Položky v menu jsou graficky ztvárněny tak, aby uživateli napověděly, že se pohybuje v hierarchii (viz obrázek 5.1).

Při změně kontextu na jinou oblast zájmu je kamera plynule přemístěna do výchozího pohledu, který daná oblast zájmu přímo definuje. Tato výchozí nastavení jsou určena tak, aby byly pohodlně viditelné všechny důležité komponenty daného funkčního celku. Při změně úrovně detailu kontextu dochází také k plynulé změně zobrazení 3D modelů (viz obrázek 5.2).



**Obrázek 5.2:** Po najetí kurzorem myši na popisek dojde ke zvýraznění popsané komponenty

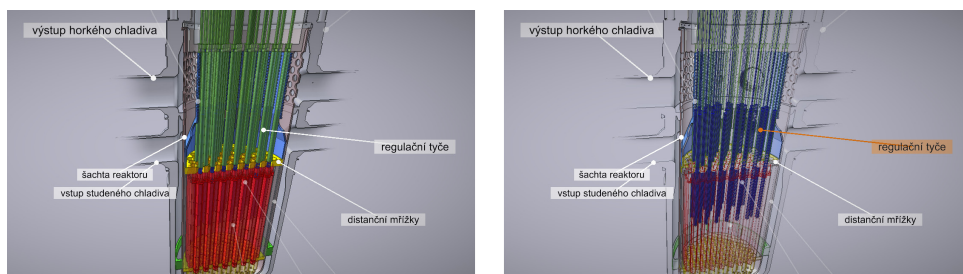
Pohyb kamery je fixován bodem, do kterého kamera neustále směřuje. Uživatel může kameru přemístit držením levého tlačítka myši a tažením ve směru kýženého pohybu. Rychlost pohybu lze měnit vzdáleností kurzoru od bodu, ve kterém došlo k prvotnímu levému kliknutí. Kolečkem myši lze měnit vzdálenost od centrálního bodu, do kterého kamera směřuje. Všechny pohyby jsou implementovány s mírnou setrvačností, pohyb je tedy plynulý a přirozený. Aplikaci lze ukončit stisknutím klávesy *ESC*.



## 5.3 Textový popis a popisky komponent ve scéně

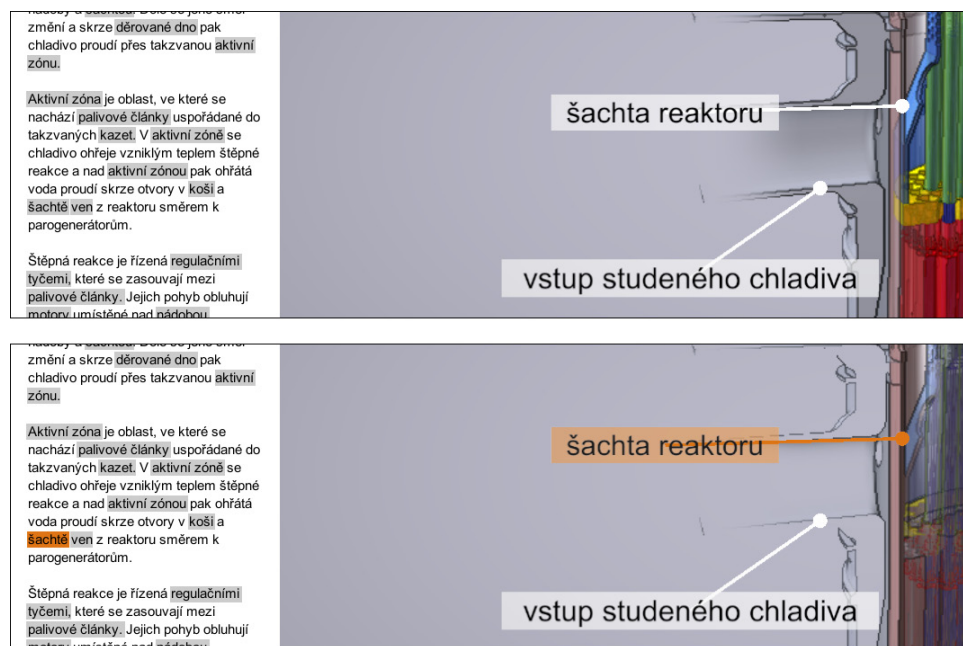
Každý funkční celek nese rozdílnou množinu popisků. Popisky se dynamicky pohybují v závislosti na pozici kamery. Ke změně jejich pozice dochází plynule spojitým pohybem. Vodicí čáry popisků se nikdy nekříží, stejně jako anotační texty, které se v ustálené pozici rovněž nepřekrývají. Pokud zasáhne anotační text popisku mimo záběr, bude vodicí čára spolu s kotvou zobrazena poloprůhledně.

Při najetí na popisek se zvýrazní popisovaná komponenta (viz obrázek 5.3), tím se řeší víceznačné situace, kdy nemusí být uživateli jasné, ke které komponentě popisek patří. Popisky funkčních celků, které v hierarchii představují vnitřní uzly grafu, reagují na klik myši a změni oblast zájmu kamery do popisovaného kontextového detailu.



**Obrázek 5.3:** Po najetí kurzorem myši na popisek dojde ke zvýraznění popsané komponenty

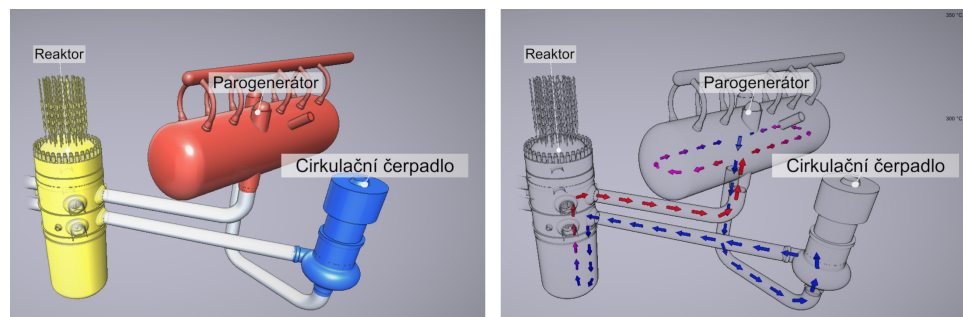
Vyznačené odborné termíny v textu levé lišty jsou propojeny s příslušnými popisky ve scéně. Díky tomu uživatel popisovanou komponentu snadno najde i v prostorovém modelu. Zvýrazňovací efekt je shodný s tím, který se aktivuje po přjetí myši na anotaci popisku. Propojení textu s popisky demonstruje dvojice obrázků 5.4.



**Obrázek 5.4:** Najetím kurzorem myši na vyznačený termín v textu se zvýrazní i související komponenta ve scéně

## 5.4 Vizualizace cirkulačních dějů

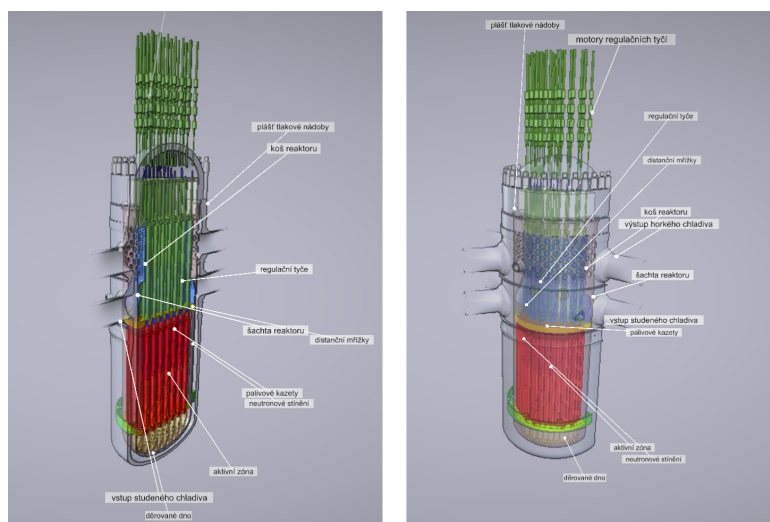
Zobrazení cirkulačních dějů je realizováno v samostatném pohledovém módu. Mód zobrazení lze přepínat ovladačem v levém dolním rohu. Spolu s glyphy se v pravé horní části zobrazí také teplotní stupnice, která dále vysvětluje teplotní rozsah hodnot. Barva a velikost glyphů se dynamicky mění v závislosti na nastavení bodů trajektorie, po kterých se pohybují. Porovnání obou módů zobrazení si lze prohlédnout na obrázku 5.5.



**Obrázek 5.5:** Porovnání klasického módu s módem zobrazení cirkulace chladiva v prvním cirkulačním okruhu

## 5.5 Ilustrační efekty vizualizace

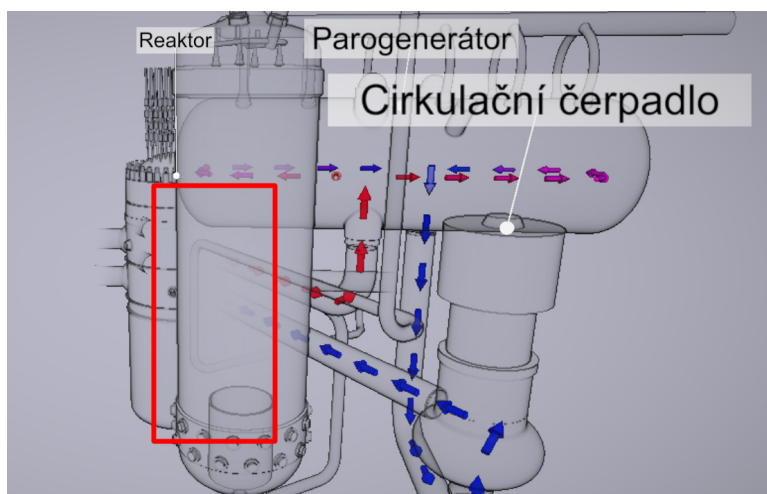
Vizualizace používá ilustračních efektů. Zobrazení vnitřních komponent je realizováno poloprůhlednými plochami vnějších plášťů. Vjem vylepšuje implementace metody, která více zviditelňuje objekty s vyšší důležitostí. U reaktoru je jeho vnitřní uspořádání zobrazeno ve statickém řezu (viz obrázek 5.6).



**Obrázek 5.6:** Reaktor je zpracován ve statickém řezu. Jeho plášť je zprůhledněn, vnitřní komponenty jsou díky ilustračním technikám dobře viditelné

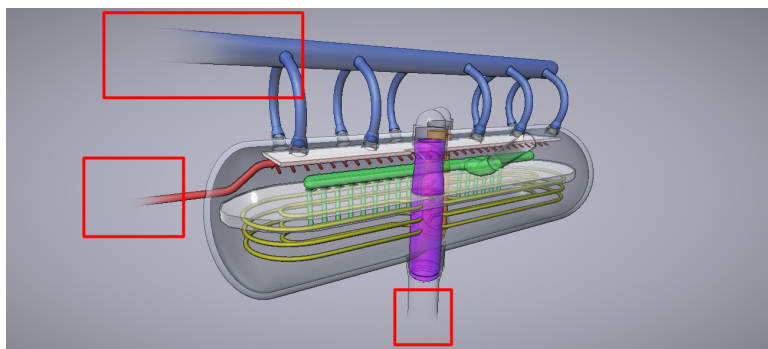
Plochy přikloněné směrem ke kameře jsou zprůhledněny více než plochy téměř rovnoběžné s vektorem pohledu kamery, díky tomu vynikly siluety zprůhledněných objektů bez zbytečného zastínění komponent, které se nacházejí za nimi. Význačné hrany jsou zvýrazněny linkami. Ke zvýraznění tvarů jsou pak dále použity spekulární odlesky.

K percepci hloubky napomáhá efekt proměnlivé hodnoty alfa kanálu. Objekty, které jsou bezprostředně za poloprůhlednou bariérou, jsou zobrazeny výrazněji než vzdálenější objekty (viz obrázek 5.7).



**Obrázek 5.7:** Percepci hloubky napomáhá efekt mlhy, kdy jsou objekty za poloprůhlednou plochou zobrazeny s různou výrazností v závislosti na jejich vzdálenosti od poloprůhledné bariéry. V tomto příkladě se díváme na část parogenerátoru a reaktoru skrze poloprůhledný plášť kompenzátoru objemu. V červeném rámečku si všimněte, že ačkoliv jsou tvary parogenerátoru dobře čitelné, reaktor je za poloprůhledným pláštěm takřka neviditelný. Zajímavý efekt tvoří trubky vedoucí do reaktoru, které s rostoucí vzdáleností postupně mizí

Zajímavým efektem byly zvýrazněny trubky, které pokračují do komponent mimo aktuální oblast zájmu. Podél trubky se její alfa kanál postupně interpoluje až k nulové hodnotě. Díky detekci hran je však vedení trubky o něco déle zvýrazněno tmavšími linkami, které mizejí až o něco později (viz obrázek 5.8).



**Obrázek 5.8:** Efekt mizejících trub, které vedou mimo aktivní oblast zájmu

## Kapitola 6

### Uživatelské testování

Aplikace bude otestována kvalitativním testem. Účelem testu je úvodní charakteristika prvotních problémů implementace. Sledovat budu kromě funkčních problémů také subjektivní pocity z aplikace, které se mohou týkat například poutavosti prezentace.

Od testu očekáváme odhalení možných hrubých chyb v navigaci, problémů uživatelského rozhraní či efektivity prezentace informací. Test rovněž buď potvrdí, nebo vyvrátí, zda byly v kapitole analýzy odhaleny všechny důležité požadavky na software.

#### 6.1 Participanti

Složení participantů bylo zvoleno s ohledem na znalost problematiky jaderné elektrárny, věk, zkušenost s ovládáním počítače a zkušenosti s ovládáním počítačových her. Všichni otestovaní participanté jsou charakterizováni v tabulce 6.1.

číslo participanta	věkové rozmezí	znalý jaderné elektrárny	zkušenost s počítačem	zkušenost s hrami
1	25-30	ne	průměrné	průměrné
2	30-35	ano	průměrné	vyšší
3	50-55	ano	vyšší	žádné
4	50-55	ne	nižší	žádné
5	20-25	ne	vyšší	vyšší
6	25-30	ne	nižší	nižší

Tabulka 6.1: Charakteristika testovaných uživatelů

#### 6.2 Prostředí kvalitativního testování

U uživatelského testu jsem byl osobně přítomen ve stejné místnosti jako participant. Uživatel pak pracoval se systémem těchto parametrů:

- procesor: i5-3210M (2,5 GHz),

- grafická karta: Nvidia 650M (2Gb GDDR5),
- množství RAM: 8 Gb,
- operační systém: Windows 8.1 64bit,
- monitor: uhlopříčka 15', fullHD (1920x1080), 60Hz,
- další hardware: myš, klávesnice.

Během uživatelského testu nedocházelo k bezprostřednímu psaní logů. Z testu však byl pořízen hlasový záznam participanta a videozáznam obrazovky, aby mohl být průběh testu detailněji zpracován později.

Všem participantům byl průběh před začátkem testu řádně vysvětlen. Všichni testovaní uživatelé souhlasili s pořízením hlasového a obrazového záznamu. Data jsou anonymizována a nejsou ani součástí příloh diplomové práce.

### 6.3 Struktura testu

Test uvádí participanta do smyšlené situace, kdy je na exkurzi v jaderné elektrárně a čeká na příchod průvodce a začátek prohlídky. V čekací místnosti ho upoutá kiosek se mnou vytvořenou vizualizací primárního okruhu.

Test byl rozdělen do dvou částí. V první části byl každý participant postaven před sedm úkolů:

1. Přejděte s kamerou do kontextu první cirkulační smyčky.
2. Přejděte do kontextu parogenerátoru.
3. Najděte, k čemu slouží teplosměnné trubky a kde se v parogenerátoru nacházejí.
4. Přejděte do kontextu reaktoru.
5. Zjistěte, jakým způsobem protéká chladivo reaktorem.
6. Najděte v primárním okruhu jaderné elektrárny kompenzátor objemu.
7. Prozkoumejte aplikaci a zjistěte další informace o fungování primárního okruhu.

Úkoly 1, 2 a 4 testují systém navigace mezi kontextovými oblastmi zájmu a schopnost uživatele ovládat kameru myší. Testuje také, zda si uživatel uvědomuje, že se pohybuje v hierarchické struktuře.

Úkol 3 testuje, zda je uživatel schopen vyhledat informace v popisném textu a zda si uvědomuje propojení vyznačených termínů v textu s konkrétními komponentami ve scéně. Úkol 3 rovněž testuje, zda je zvýraznění komponent ve scéně dostatečně výrazné a poutavé.

Úkol číslo 5 se zaměřuje na vizualizaci cirkulačních dějů za pomoci glyphů. Uživatel by měl být schopen najít důležitý přepínač v levém dolním rohu, kterým aktivuje mód zobrazení s glyphy.

Úkol 6 je zaměřen na schopnost uživatele efektivně a rychle vyhledat zvolenou komponentu. V této chvíli by měl být uživatel již navyklý na způsob navigace.

Poslední úkol pak nechává uživateli volnost pohybu v aplikaci. V tomto bodě se zaměřuji hlavně na poutavost aplikace, zda je pro uživatele zajímavá a v čem. Sleduji akce, které participant provádí, a jeho komentáře, zda ho něco zaujalo či překvapilo.

Před prováděním praktických bodů testu bylo uživatelům sděleno, že navigovat se lze pomocí klikání na popisky ve scéně a ovládací prvky v levé liště. Rovněž jim bylo řečeno, že lze s kamerou pohybovat, přesný způsob, jak pohyb uskutečnit, jim však sdělen nebyl.

Po provedení všech sedmi úkolů se test přesune do druhé části, ve které každý participant vyplní post test dotazník, který se skládá z celkem osmi tvrzení. Každé tvrzení uživatel ohodnotí dle Linkertovy škály. I v této fázi je však participantům nasloucháno a je žádoucí, aby své zatržené hodnocení vhodně odůvodnili. Všech osm tvrzení je uvedeno v následujícím seznamu.

1. Navigace mezi součástkami je snadná.
2. Při přechodech kamery mezi kontexty jsem neztratil/neztratila pojem o umístění součástek.
3. Najít součástku zmíněnou v textu bylo snadné.
4. Sledování směru toku chladiva bylo díky šipkám názorné.
5. Vizualizace průtoků mi umožnila vidět, kde je jaká teplota.
6. Při pohybu kamery byl pohyb popisků přirozený.
7. Interakcí s aplikací jsem byl/byla schopen/schopna pochopit primární okruh snáže, než kdybych se tyto informace učil/učila například z učebnice.
8. Použitý způsob vizualizace je vhodný k prezentaci složitých systémů, jako je jaderná elektrárna.

První dvě tvrzení cílí na názor uživatele týkající se navigace v hierarchické struktuře. 3., 4. a 5. tvrzení klasifikuje efektivitu prezentace informace. Koněčně 7. a 8. tvrzení se zaměřuje na celkový efekt vizualizace a na to, jak pozitivně vnímá participant zvolený způsob prezentace.

## 6.4 Výsledky testování

Účastníci vstoupili do testu s různými pocity. Participant 1 a 5 byli okamžitě rezervovaní vůči slovnímu popisu v levé části okna a popsali, že by si



pravděpodobně nic nepřčetli. Naopak ostatní participantů prvotní setkání s aplikací nikterak nekomentovali. Participant 1 navíc uvedl, že pro něj není téma vizualizace atraktivní: „*Mě to odrazuje už jen proto, že to je něco strašně technického.*“

### 6.4.1 Praktická část

První úkol změny zaměřeného detailu na první cirkulační smyčku všichni uživatelé splnili bez větších problémů. Ačkoliv jim byly popsány aktivní prvky aplikace, všichni uživatelé zvolili pro změnu kontextu klik na popisek primárního okruhu přímo ve scéně. Uživatel 5 však nevedl prvotní klik na popisek primární smyčky, ale na 3D objekt smyčky ve scéně. Bez zaváhání však obratem provedl klik na popisek. Participant 1 uvedl, že popisky nevypadají jako aktivní prvky, a pokoušel se přejít do detailu pohybem kamery a přiblížením „*Jsem si myslel, že přejdu k té věci pomocí přiblížení, tak jak jsem zvyklý třeba ze hry.*“

Změnu pohledu komentoval jen 1. a 5. participant. Uživatel číslo 5 okamžitě cítil nutkání vrátit se do předchozího stavu, aby si znovu prohlédl, na kterou část primárního okruhu zrovna zamířil. Byl překvapen, že kromě prvního cirkulačního okruhu všechny ostatní komponenty zmizely. Naopak participant 1 byl překvapen změnou pohledu kamery: „*Teď mi trochu vadí, jak jsem se kousek pootočil.*“

Druhý úkol přechodu do detailu parogenerátoru všichni splnili bez větších problémů. Participant 1 chvíli hledal pojem parogenerátor, ale během minuty dokázal v popisovaných komponentech parogenerátor najít.

S dalším úkolem, který se týkal nalezení teplosměnných trubek a jejich funkce, se popasovali účastníci úspěšně. Všichni participanté teplosměnné trubky našli na základě jejich popisku ve scéně. Každý uživatel si okamžitě všiml, že došlo ke zvýraznění součástky po najetí na popisek. Žádný participant neměl problém popsat barvu a tvar teplosměnných trubek. Každý uživatel však zkoušel na popisky v nejnižší úrovni hierarchie klikat. Uživatel 6 pouze zkoušel, jestli dojde k nějaké akci. Uživatelé 2 a 5 očekávali, že se po kliku objeví přímo ve scéně krátký popis aktivované komponenty: „*Čekám, že mi vyjede nějaká tabulka s textem.*“ Participant 1 a 3 čekali, že dojde také ke změně pozice pohledu a jeho zaměření na popsanou komponentu. Naopak uživatel 4 pouze očekával, že kliknutím zůstane komponenta zvýrazněná.

Uživatelé byli různě ochotni hledat informace v textu. Participant 3, 4 a 6 bez větších problémů s textem v levé části okna pracovali. Uživatelé 1 a 2 se ke čtení uchýlili až tehdy, když si ověřili, že funkci nezjistí jiným způsobem. 5. participant však úkol mírně odbyl se slovy: „*Tady bych to asi našel v tom textu,*“ a jal se pokračovat dalším úkolem. Uživatelé 1, 2 a 3 odhalili propojení mezi vyznačenými termíny v textu a zvýrazňováním 3D částí ve vizualizaci po najetí na tyto termíny. Uživatelé 4 a 6 na popisky najeli, ale zvýrazňování 3D komponent si nevšimli, nebo efekt nekomentovali. Uživatel 5 s textem nepracoval.

Následujícím úkolem bylo přejít pohledem do reaktoru. V této chvíli všichni participanté uvedli, že by se chtěli vrátit zpět. Participant 6 a 2 ihned použili



položky v levém navigačním menu. Uživatelé 1, 3 a 4 chvíli přemýšleli nad organizací levého menu, ale po chvíli se navigovali do nadřazených prvků v hierarchii. Participant 5 očekával, že se do předchozích pohledů dostane dostatečným oddálením pohledu.

V této fázi testu již všichni až na pátého participanta rozuměli hierarchickému uspořádání modelu a funkci levého menu. Pátý participant tápe v organizaci menu. Tvar položek nevnímá jako posloupnost. Po dotazu, jak chápe jednotlivé odkazy v menu, odpovídá: „*Myslím, že když na to kliknu, tak se to rozbalí a objeví se podřazené položky.*“ Žádný participant nevnímá šipkový tvar položek v menu jako funkční prvek, ale spíše jako prvek designový. Uživatel 4 však není nadšen ořezanými možnostmi pohybu v hierarchii: „*Chtěl bych tu mít vypsané všechny ty části, abych nemusel jít do primárního okruhu a až pak zase do jiné komponenty, ale mohl tam přejít rovnou.*“

Dalším úkolem bylo popsat proudění chladiva reaktorem. Nejlépe se úkolu zhostil uživatel číslo 3, který byl po celou dobu testu ze všech uživatelů nejaktivnější a nejvíce experimentoval. Přepínač, který zobrazuje proudění chladiva totiž náhodou odhalil již dříve, když se nebál zkoušet všechny aktivní prvky v uživatelském rozhraní. Právě proto využil této dříve nabyté znalosti a okamžitě mód zobrazení proudění aktivoval. Zbytek participantů však doposud o existenci speciálního módu zobrazení nevěděl. Uživatelé se snažili informaci vyčíst z textu, ze kterého byli schopni s různou úrovní úspěšnosti průběh průtoku svými slovy popsat. Uživatelé 1, 3 a 5 průběh průtoku nereprodukovali zcela správně. Všechny uživatele kromě druhého jsem pobídl, jestli nepřijdou na jiný způsob, jak informaci z vizualizace vyčíst. Uživatelé 1, 3 a 4 začali hledat klíčové slovo *průtok* v popisném textu, participant 6 hledal termín průtoku v popiscích komponent reaktoru. Participant 1, 5 a 6 ovladač našli po tom, co jsem je povzbudil k prozkoumání ovládacích prvků. Naopak uživatele 4 jsem musel výslovně přesvědčit, že ve vizualizaci existuje speciální mód zobrazení. Po této informaci ovládací prvek našel i participant 4.

Nejpozitivnější reakci měl uživatel číslo 5, který zjistil, že chladivo neproudí tak, jak si dříve myslel: „*Je potěšující, že mě byl ten program schopn opravit, a tím něco naučit.*“ Ostatní uživatelé však také reagovali kladně. Všichni uživatelé až na participanta 4 uvedli, že si ovladače všimli, ale nepřipisovali mu přílišnou důležitost. Uživatel 5 uvedl, že přepínač vnímal jen jako nějaké nastavení. Uživatelé 3 a 4 zase ovladač vnímali jako poznámku pod čarou: „*Já jsem viděl, že tam něco je, ale je to někde dole a působilo to na mě jen jako nějaká poznámka pod čarou.*“ Všichni uživatelé však bez problému dokázali průtok pomocí tohoto módu zobrazení popsat. Zajímavým poznatkem bylo, že po položení doplňující otázky: „*V jakém rozmezí hodnot se asi teplota chladiva pohybuje?*“ nebyli někteří uživatelé (4 a 5) schopni informaci najít, protože si nevšimli barevné stupnice, která je zobrazena v pravém horním rohu.

Test pokračoval šestým úkolem, ve kterém měl uživatel najít kompenzátor objemu. K úkolu uživatelé přistupovali odlišně, ale všichni začali hledání v kořeni hierarchie. Uživatelé 2, 3, 5 a 6 v poměrně krátké době našli popisek čtvrtého cirkulačního okruhu, který klíčové slovní spojení *kompenzátor objemu*

explicitně uvádí. První participant našel popisek až po delší chvíli. Čtvrtý participant kompenzátor objemu našel volným prozkoumáváním hierarchie cirkulačních okruhů.

V posledním úkolu měli uživatelé volnost aplikaci prozkoumat. Všichni byli ochotní si s vizualizací poměrně dlouhou dobu hrát.

#### 6.4.2 Další poznatky v praktické části

Všichni uživatelé komentovali způsob ovládání kamery. Uživatelé 2, 3, 4 a 5 uvedli, že pohyb pohledu v závislosti na pohybu kurzoru myši je invertovaný. Uživatelé 1 a 6 kritizovali přílišnou restrikcí pohybu kamery (pevný střed, do kterého kamera pohledem vždy směřuje). Participant 2 a 5 kritizovali přílišnou setrvačnost a pocitově pomalou reakci kamery na pohyb myši. Uživatel 3 se odlišoval tím, že kamerou pohyboval minimálně: „*Výchozí pohledy mi pro práci zcela stačí.*“

Více participantů (1, 2, 3 a 5) uvedlo, že by systém zvýrazňování komponent ve scéně na základě najetí kurzoru na vyznačený termín v textu uvítali také reverzním směrem. Pokud by se po najetí na popisek ve scéně také zvýraznil termín v textu, usnadnil by uživateli najít o zaměřené komponentě informace mnohem rychleji. Participant 4 nevnímal zvýrazněné termíny v textu jako aktivní prvky, zvýrazňování komponent odhalil náhodou až v pozdější části testu. Naopak uživatel 6 odhalil zvýrazňování téměř okamžitě, ale tuto funkcionalitu v průběhu testu vůbec nevyužíval.

Zajímavým poznatkem byl rozdílný přístup k barvám komponent. Participant 3 a 6 vnímali systém obarvení okruhů v primární smyčce, kde jsou okruhy 1, 2 a 3 obarveny různými odstíny modré a 4. cirkulační okruh s kompenzátořem objemu je obarven červeně. Na přímý dotaz však jen participant číslo 6 dokázal přesně popsat, proč je čtvrtý okruh obarven rozdílně: „*Asi protože to je jediný okruh, kde je ten kompenzátor objemu.*“ Uživatel číslo 5 k detailu kompenzátoru objemu uvedl, že si myslí, že modře obarvená sprcha bude ochlazovat obsah nádoby a že červené ohříváky budou naopak obsah nádoby ohřívát. Výslovně tak spojil barvu komponent s jejich funkcí, ačkoliv bylo takové obarvení komponent zvoleno náhodně bez jakéhokoli důrazu na jejich fungování.

Participant také rozdílně přistupovali k práci s textem. Participant číslo 5 na začátku prakticky odmítal s textem pracovat. V pozdější části ho však aplikace upoutala a motivovala ke čtení, ačkoliv stále text považoval za příliš dlouhý. Naopak participant 2 a 6 hodnotili délku textu jako přiměřenou. Ostatní participant 1, 4 a 5 délku textu nekomentovali.

Uživatelům 1, 4 a 5 chyběly ve vizualizaci dílčích částí náznaky okolních komponent. Když například bylo v textu reaktoru uvedeno, že chladivo přichází směrem od cirkulačního čerpadla, očekávali, že bude cirkulační čerpadlo někde alespoň naznačeno. Participant 6 uvedl, že by uvítal informaci o umístění primárního okruhu v areálu jaderné elektrárny.

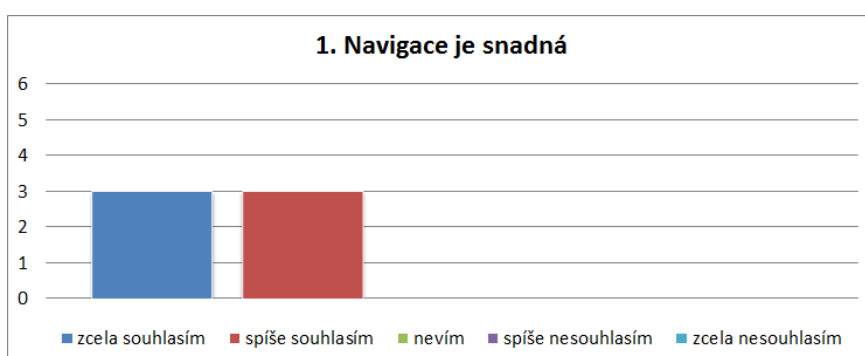
Participant 1, 2, 3 a 5 by uvítali větší možnosti interakce s funkcemi reaktoru. Naopak uživatel 6 výslovně uvedl, že takovou interakci nepostrádá. Mezi uživateli se objevily rozličné reakce na úroveň odbornosti aplikace.

Uživatelé 1, 2 a 5 vnímají aplikaci jako příliš technickou a odbornou, ačkoliv participant 2 dodává, že mu míra odbornosti vyhovuje. Ostatní participantů považují odbornost vizualizace za vyváženou.

Celkově uživatelé přistupovali k aplikaci pozitivně: „*Jo, mně se to líbí, myslím, že je to super.*“ Případně vyjádřili respekt k práci, kterou vynaložil autor aplikace.

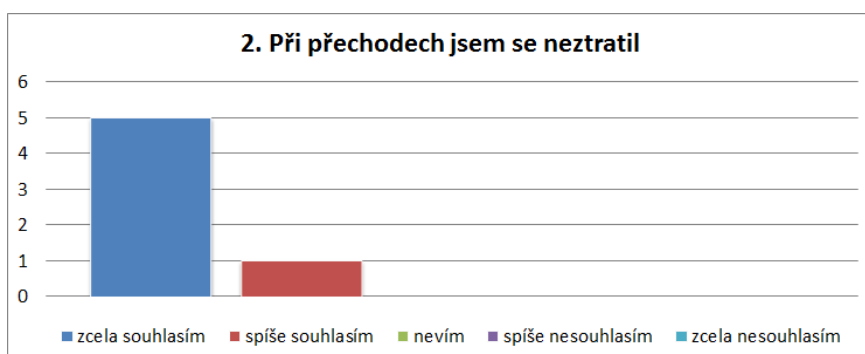
### 6.4.3 Post test dotazník

Test pokračoval osmi otázkami, ve kterých uživatelé odpovídali dle Linkertovy škály. Vzhledem k malému počtu účastníků nemá smysl výsledky statisticky zpracovávat. Grafy počtů odpovědí jsou pouze ilustrační a lze si je prohlédnout na obrázku. Účelem byl komentář uživatele k příslušnému tvrzení.



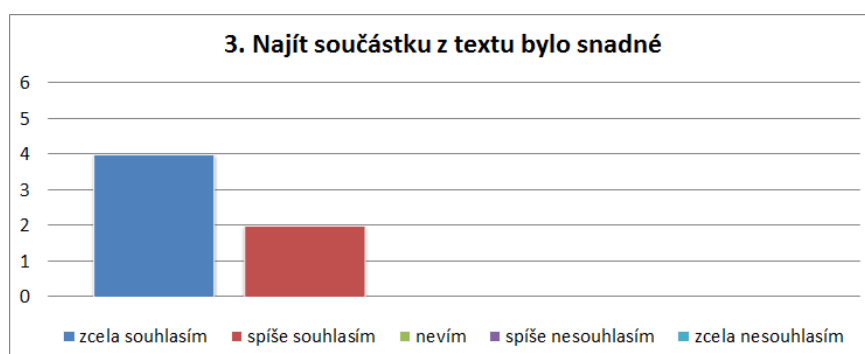
Obrázek 6.1: Odpovědi uživatelů na 1. otázku post test dotazníku

První polovina participantů odpověděla, že jim ovládání připadalo snadné. Uživatelé 2 a 5 zahrli možnost *spíše souhlasím*, protože nebyli zcela spokojeni s chováním kamery a cítili, že jim špatná ovladatelnost znemožňuje mít řízení pohledu pod kontrolou. Participant 1 uvádí, že popisky komponent nepovažuje na první pohled za ovládací prvky.



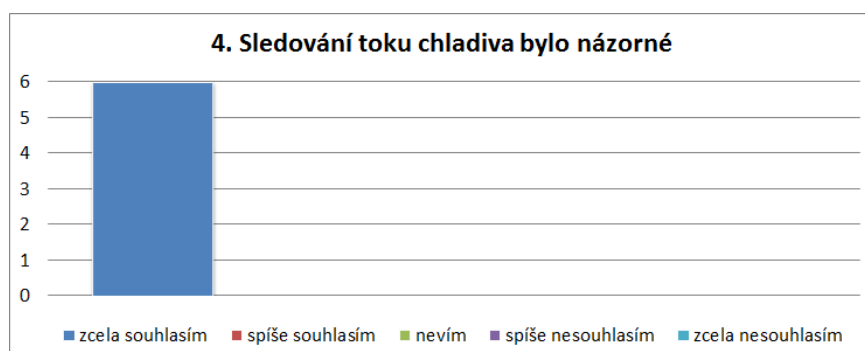
Obrázek 6.2: Odpovědi uživatelů na 2. otázku post test dotazníku

Všichni testovaní uživatelé kromě pátého uvedli, že neztratili orientaci při přechodech mezi detaily v hierarchii. Participant 5 ztratil orientaci jednou na začátku testu, kdy nebyl ještě zcela navyklý na způsob navigace.



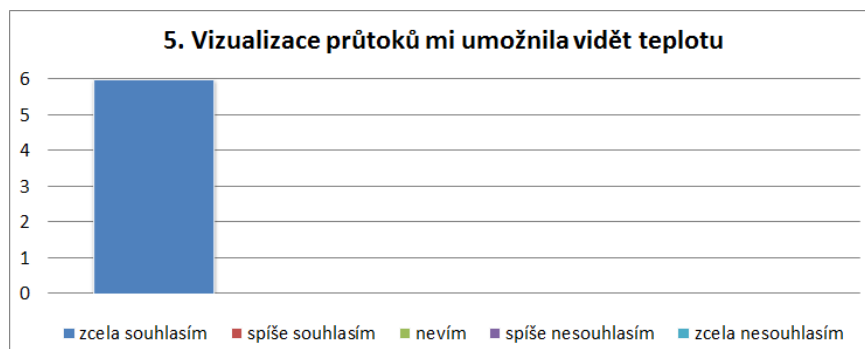
Obrázek 6.3: Odpovědi uživatelů na 3. otázku post test dotazníku

Většina uživatelů uvedla, že neměla problém nalézt ve 3D pohledu komponenty zvýrazněné v textu. Participant 4 se přiklonil k odpovědi *spíše souhlasím*, protože nevnímal způsob vyznačení termínů jako aktivní prvky. Stejnou možnost zvolil i participant číslo 2, protože nebyl spokojen s tím, že v určitých pozicích pohledu nejsou viditelné všechny popisky ve scéně. Z tohoto důvodu nebyl schopen lokalizovat objekty, u kterých nebyly popisky viditelné.



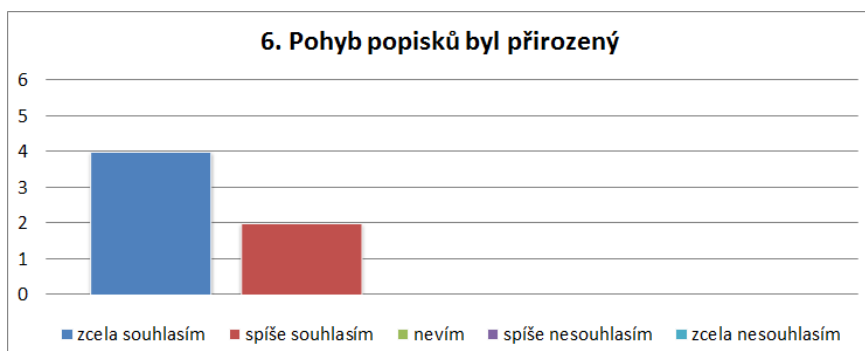
Obrázek 6.4: Odpovědi uživatelů na 4. otázku post test dotazníku

Všech 6 uživatelů jednoznačně uvedlo, že neměli problém s pochopením průtoku chladiva, když aplikaci přepnuli do příslušného módu, který jim tuto informaci zprostředkoval.



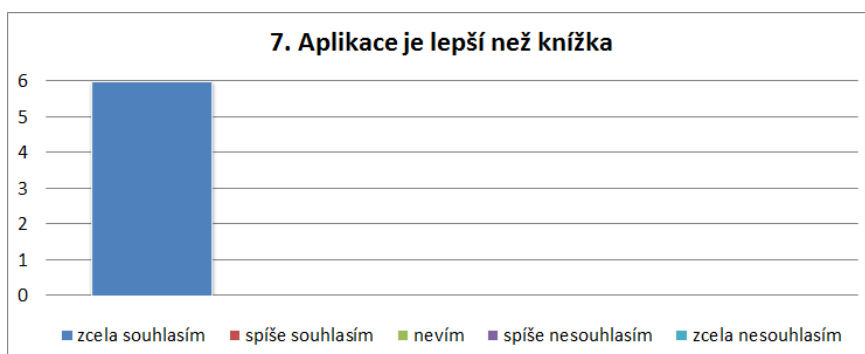
Obrázek 6.5: Odpovědi uživatelů na 5. otázku post test dotazníku

Podobně se testovaní uživatelé shodli na tom, že způsob obarvení šipek jim bez problému předalo informaci o teplotě chladiva v různých místech primárního okruhu.



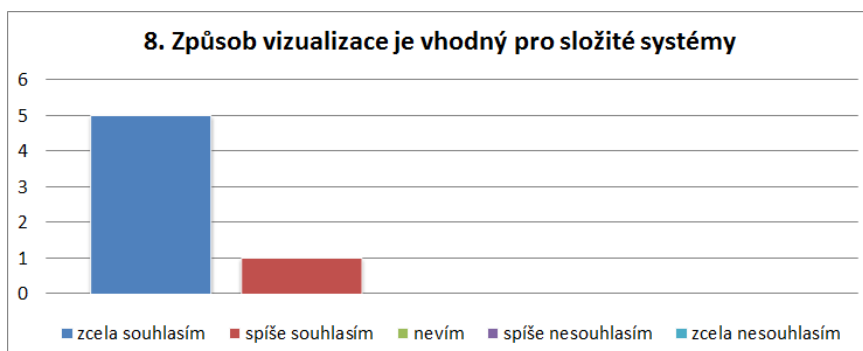
**Obrázek 6.6:** Odpovědi uživatelů na 6. otázku post test dotazníku

Také pohyb popisků hodnotili participanti kladně. Uživatel 2 uvedl, že mu vadily popisky vyjíždějící mimo obraz i v situacích, kdy je kolem anotované komponenty ještě mnoho místa. Participant 5 nazval při testu pohyb popisků divokým, ale uvedl, že při ustálení pohybu kamery se i pozice popisků velice rychle uklidní. Díky tomu ho jejich pohyb nikterak neomezoval v práci s vizualizací.



**Obrázek 6.7:** Odpovědi uživatelů na 7. otázku post test dotazníku

Všichni participanti se shodli na tom, že efektivita výuky při práci s interaktivní vizualizací je mnohem vyšší, než výuka s využitím statických obrázků v učebnici s textem. Participant 6 dokonce uvedl, že by primární okruh z knížky pravděpodobně ani nepochopil.



**Obrázek 6.8:** Odpovědi uživatelů na 8. otázku post test dotazníku

Většina uživatelů uvedla, že je použitý způsob vizualizace vhodný pro podobně složité systémy, jako jsou vedle jaderné elektrárny například dopravní prostředky, stroje, nebo dokonce i těla živočichů. Uživatel 5 uvedl, že ve stavu, v jakém se aplikace nachází, nepovažuje způsob prezentace textových informací za nejlepší.

## 6.5 Nálezy

Na základě problémů, které se objevily u jednotlivých testovaných uživatelů, jsem identifikoval jednotlivé nálezy.

### 6.5.1 Pohyby pohledu kamery

Velká část uživatelů vnímala pohyb navigace pomocí kurzoru myši jako invertovaný. Dva participantů uvedli, že odezva pohybu kamery je při pohybu myši pomalá, kamera se tak pohybuje pocitově ztěžka. Někteří uživatelé také kritizovali přílišnou restrikcí pohybu. Uvítali by kromě rotace a změny měřítka také posun bodu, do kterého kamera směřuje.

#### Navržená řešení:

- otočení os pohybu,
- vyšší akcelerace pohybu a nižší setrvačnost.

### 6.5.2 Způsoby přechodu mezi detaily

K přechodu mezi kontextovými detaily sloužily popisky ve scéně a levé navigační menu. Jeden participant uvedl, že popisky nevypadají jako akční prvek. Nenapadlo by ho na ně vůbec klikat. Jeden uživatel zkusil také klikat na samotné prostorové komponenty, na které ukazovaly akční popisky.

Uživatelé byli úspěšní v chápání hierarchického uspořádání scény. Šipkový tvar položek v menu ale chápali jen jako design. Jeden uživatel uvedl, že mu chybí více možností k pohybu hierarchií než jen o jednu úroveň níž a výš. Jeden participant hierarchické uspořádání vůbec nepochopil.

Několik uživatelů se snažilo pohybovat hierarchií přiblížením a oddálením pohledu kamery.

#### ■ Navržená řešení:

- změna podoby kurzoru po najetí na popisek ve scéně (uživatel jej začne vnímat jako aktivní prvek),
- možnost klikat kromě popisků také přímo na 3D komponenty,
- změna položek v menu a vytvoření vjemu hierarchie na základě odsazení položek od levého okraje,
- umožnit přechod mezi patry hierarchie scrollováním.

### ■ 6.5.3 Propojení vizualizace s textem

Všichni uživatelé odhalili efekt zvýrazněných termínů v textu, kdy se po jejich aktivaci přejetím myši zvýrazní i příslušná komponenta v 3D pohledu. Dále uvedli, že by očekávali stejný způsob propojení i opačným směrem, kdy by po najetí na popisek ve scéně došlo také ke zvýraznění termínu v textu. Jeden uživatel pak nevnímal zvýrazněné pojmy v textu jako interakční prvky.

K délce textu byli dva uživatelé mírně rezervovaní, ale oba se ke čtení v pozdějších fázích uchýlili.

Mnoho uživatelů odhalilo nekonzistenci mezi funkcemi popisků ve scéně. Ve vyšších úrovních hierarchie popisky sloužily k navigaci, na nejnižší úrovni ale žádnou akci nezprostředkovávaly. Participantů se lišili v tom, jaké akce od klikání očekávali, klikání však vyzkoušeli úplně všichni. Dva participantů očekávali zobrazení textového popisu přímo ve scéně, jiní změnu pohledu kamery a detail komponenty (jak k tomu docházelo ve vyšších úrovních).

#### ■ Navržená řešení:

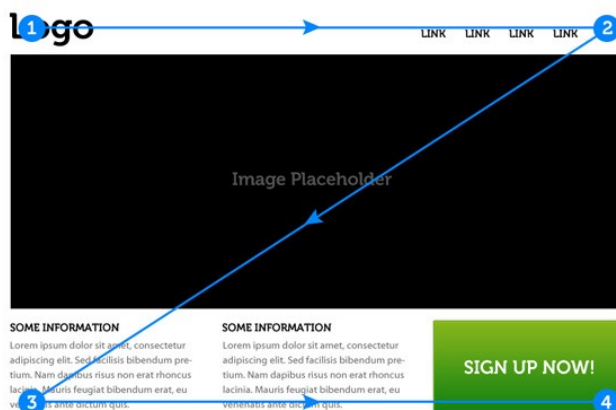
- propojení i v opačném směru (po najetí kurzorem na popisek dojde ke zvýraznění příslušného termínu v textu),
- vytvoření speciálních pohledů kamery po kliknutí na dílčí části komponent.

### ■ 6.5.4 Zobrazení cirkulačních dějů

Všichni uživatelé kromě jednoho měli problém najít ovladač pro zobrazení proudění chladiva. Přepínače v levém dolním rohu si někteří nevšimli, jiní mu nepřikládali důležitost či si jeho funkci nespojili se změnou zobrazení.

### ■ Navržená řešení:

- barevně zvýraznit přepínač módu zobrazení s cirkulačními ději,
- umožnit přepnutí pomocí vyznačeného termínu přímo v textu,
- přesunout ovladač na vhodnější místo v závislosti na běžném postupu čtení v takzvaném z vzoru (viz obrázek 6.9).



**Obrázek 6.9:** Lidé v evropských zemích prohlížíjí obrazovku specifickým způsobem (takzvaný z vzor), který je spojený též se zvyklostí čtení směrem zleva doprava (převzato z [Bab17])

## ■ 6.6 Zhodnocení testu

I přes problémy, na které uživatelé při testování narazili, se vyjadřovali o vizualizaci kladně. Hlavním pozitivem testu je fakt, že nálezy se netýkají zvolených vizuálních efektů, ale spíše navigace, podoby rozhraní a popisků komponent.

Někteří uživatelé měli po ukončení testu další zájem o informace týkající se chodu jaderné elektrárny. Těmto uživatelům jsem pak na místě vysvětlil funkci jaderné elektrárny a roli primárního okruhu. Při vysvětlování jsem efektivně využil aplikace, kterou jsem ale nechal ovládat testovaným uživatelem. Díky těmto malým rozhovorům jsem zjistil, že by aplikace byla velmi efektivní pomůckou i při klasické výuce.



## Kapitola 7

### Závěr

Tato diplomová práce se zabývala ilustrativní vizualizací primárního okruhu jaderné elektrárny. Nejprve jsem se seznámil s technickou problematikou primárního okruhu jaderné elektrárny.

Zanalyzoval jsem požadavky na software vizualizace dle praktického využití aplikace a cílové skupiny. Analýza vedla k interaktivní vizualizaci, která měla běžet v reálném čase.

Provedl jsem rešerši ilustračních metod, na jejichž základě jsem navrhl vhodné řešení celé aplikace. Jako vhodné metody pro použití v mém případě se jevily:

- vylepšené prolínání poloprůhledných ploch na základě důležitosti,
- zvýraznění siluet pomocí směru normály,
- zvýraznění význačných hran v modelu,
- řezy modelem pro zviditelnění vnitřních struktur,
- textový popis chování a dynamicky umístěné popisky komponent ve scéně pomocí techniky hedgehog labeling,
- zobrazení cirkulačních dějů a teplot pomocí animovaných prostorových glyphů.

Zvolené metody jsem implementoval v prostředí Unity engine v jazyce C# a CG. Vytvořil jsem prostorový model primárního okruhu v programu 3Ds max 2017 a aplikoval na něj implementované metody. Připravil jsem spustitelnou aplikaci v prostředí Microsoft Windows 8.1 a Windows 10.

Aplikaci jsem následně prověřil kvalitativním testem s šesti uživateli. Uživatelé reagovali na aplikaci kladně a pracovali s ní vcelku úspěšně. Identifikované problémy aplikace se týkaly převážně způsobu navigace a uživatelského rozhraní. Zvolené ilustrační techniky zobrazování fungovaly efektivně bez větších problémů.

## 7.1 Budoucí práce

Budoucí práce se pojí s výsledky uživatelského testu. Byly identifikovány problémy v aplikaci, které se týkaly navigace, uživatelského rozhraní či propojení textu s popisky ve scéně. Následně byly navrženy možnosti, jak by odhalené problémy mohly být eliminovány.

Další oblastí vývoje by mohlo být přepracování ovládání aplikace tak, aby byla použita místo myši dotyková obrazovka.

Celá práce je vytvořena s ohledem na využití konkrétního modelu primárního okruhu jaderné elektrárny. Implementace by však v budoucnu mohla být doplněna tak, aby bylo možné jednoduše vizualizovat jakýkoli model podobného charakteru bez větších zásahů do kódu.

## Příloha A

### Literatura

- [Bab17] Nick Babich, *Z-shaped pattern for reading web content*, <https://uxplanet.org/z-shaped-pattern-for-reading-web-content-ce1135f92f1c>, 2017.
- [BGKG05] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M. Eduard Gröller, *Illustrative context-preserving volume rendering*, Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization (Aire-la-Ville, Switzerland, Switzerland), EUROVIS'05, Eurographics Association, 2005, pp. 69–76.
- [ČE16] Simopt, s.r.o. & ČEZ, a. s., *aplikace jaderné elektrárny 3d*, <https://www.svetenergie.cz/cz/aplikace/svet-energie-3d-elektrarny>, 2016.
- [Eve01] Cass Everitt, *Interactive order-independent transparency*, NVIDIA Corporation **2** (2001).
- [GAHS05] Timo Götzelmann, Kamran Ali, Knut Hartmann, and Thomas Strothotte, *Form follows function: Aesthetic interactive labels.*, 01 2005, pp. 193–200.
- [GGSC99] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen, *A non-photorealistic lighting model for automatic technical illustration*, Proceedings of the ACM SIGGRAPH Conference on Computer Graphics (1999).
- [HAS04] Knut Hartmann, Kamran Ali, and Thomas Strothotte, *Floating labels: Applying dynamic potential fields for label layout*, Smart Graphics (Berlin, Heidelberg) (Andreas Butz, Antonio Krüger, and Patrick Olivier, eds.), Springer Berlin Heidelberg, 2004, pp. 101–113.
- [IFP96] V. Interrante, H. Fuchs, and S. Pizer, *Illustrating transparent surfaces with curvature-directed strokes*, Proceedings of Seventh Annual IEEE Visualization '96, Oct 1996, pp. 211–218.

- [KmK<sup>+</sup>19] David Kouřil, Ladislav Čmolík, Barbora Kozlíková, Hsiang-Yun Wu, Graham Johnson, David Goodsell, Arthur Olson, Meister Eduard Gröller, and Ivan Viola, *Labels on levels: Labeling of multi-scale multi-instance and crowded 3d biological environments*, IEEE Transactions on Visualization and Computer Graphics (2019), SciVis Best Paper Honorable Mention.
- [LACS08] Wilmot Li, Maneesh Agrawala, Brian Curless, and David Salesin, *Automated generation of interactive 3d exploded view diagrams*, ACM Trans. Graph. **27** (2008).
- [LAS04] Wilmot Li, Maneesh Agrawala, and David Salesin, *Interactive image-based exploded view diagrams.*, 01 2004, pp. 203–212.
- [LRA<sup>+</sup>07] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin, *Interactive cutaway illustrations of complex 3d models*, ACM Trans. Graph. **26** (2007), no. 3.
- [Čm11] Ladislav Čmolík, *Interactive illustrative rendering of 3d meshes*, Ph.D. thesis, 11 2011.
- [Šm17] Antonín Šmíd, *Comparison of unity and unreal engine*, 2017.
- [Mun14] Tamara Munzner, *Visualization analysis and design (ak peters visualization series)*, A K Peters/CRC Press, dec 2014.
- [PF11] Francisco Pinto and Carla Freitas, *Illustrating volume data sets and layered models with importance-aware composition*, The Visual Computer **27** (2011), 875–886.
- [Sch94] Christophe Schlick, *An inexpensive brdf model for physically-based rendering*, Computer Graphics Forum **13** (1994), no. 3, 233–246.
- [Sha08] SharkD, *Obrázek barevných prostorů*, [https://commons.wikimedia.org/wiki/File:Color\\_solid\\_comparison\\_hsl\\_hsv\\_rgb\\_cone\\_sphere\\_cube\\_cylinder.png](https://commons.wikimedia.org/wiki/File:Color_solid_comparison_hsl_hsv_rgb_cone_sphere_cube_cylinder.png), 2008.
- [Shn96] B. Shneiderman, *The eyes have it: a task by data type taxonomy for information visualizations*, Proceedings 1996 IEEE Symposium on Visual Languages, Sep. 1996, pp. 336–343.
- [Tel14] Alexandru C. Telea, *Data visualization: Principles and practice, second edition*, A K Peters/CRC Press, sep 2014.
- [TKGS14] Markus Tatzgern, Denis Kalkofen, Raphael Grasset, and Dieter Schmalstieg, *Hedgehog labeling: View management techniques for external labels in 3d space*, 04 2014.



## Příloha B

### Soubory přílohy

- src/ - Unity projekt se zdrojovými soubory
- doxygen/ - Vygenerovaná dokumentace zdrojových souborů
- latex/ - zdrojové soubory textu
- images/ - obrázky práce
- bin/ - spustitelná aplikace vizualizace
- text.pdf - text diplomové práce
- testovací arch.pdf - dokument, který byl využit v průběhu uživatelského testu
- readme.txt - popis struktury přílohy