

Vizualizace výsledků matematického modelování sklářských pecí

Mathematical modelling results visualization of glass furnaces

Michal Gerža

Bakalářská práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal GERŽA**
Osobní číslo: **A07119**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Vizualizace výsledků matematického modelování
sklářských pecí**

Zásady pro vypracování:

1. Vytvořte literární rešerši na zadané téma.
2. Seznamte se simulací, distribucí a optimalizací teplot ve sklářské peci.
3. Provedte návrh řešení vizualizace výsledků matematického modelování sklářských pecí.
4. Na základě návrhu proveďte programovou realizaci vizualizace geometrie modelu, teplot a proudění.
5. Při realizaci využijte jazyka C++ a grafické knihovny OpenGL.
6. Vytvořte výstupní datové soubory, které budou prezentovat možnosti Vámi vytvořeného programu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. PRATA, Stephen. Mistrovství v C++. 1. vyd. Praha : Computer Press, [1991?]. 966 s. ISBN 80-7226-339-0.
2. KRUGLINSKI, David J. Mistrovství ve Visual C++. Praha : Computer Press, 1999. 854 s. ISBN 80-7226-132-0.
3. SHREINER, Dave. OpenGL Programming Guide – Seventh Edition. Boston : Pearson Education, Inc., 2009. 1019 s. ISBN 978-0-321-55262-4.
4. SEGAL, Mark, AKELEY, Kurt. The OpenGL Graphics System: A Specification : Version 2.0. Jon Leech and Pat Brown. [s.l.] : [s.n.], 2004. 382 s.
5. OpenGL [online]. Khronos Group, c1997–2010 [cit. 2010-01-21]. Dostupný z WWW: www.opengl.org.

Vedoucí bakalářské práce:

Ing. Pavel Pokorný, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

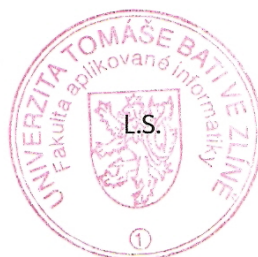
5. března 2010

Termín odevzdání bakalářské práce:

1. června 2010

Ve Zlíně dne 5. března 2010


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Hlavním cílem této bakalářské práce bylo vytvořit počítačový program, který dovede vizualizovat výsledné datové soubory ze specifických matematických aplikací v oblasti modelování sklářských pecí. Hlavní funkcí programu je vizualizace modelu pece v trojrozměrném prostoru, tedy zobrazení jednotlivých geometrických částí pece a volitelné zobrazení rozložení tepla nebo proudění skloviny ve vaně. Program je vytvořen v jazyce C++ tak, aby umožnil výběr datové modelu načtením ze souborů a jeho zobrazení pomocí knihovny OpenGL v daném promítání a osvětlení. Aplikace má jednoduché uživatelské rozhraní vytvořené prostřednictvím knihovny MFC, kde je umožněn výběr vstupního datového souboru, uložení výsledné scény, nastavení polohy modelu, úprava promítání a změna různých parametrů jednotlivých komponent modelu pece na vymezené scéně.

Klíčová slova: vizualizace, promítání, geometrie, model, data, OpenGL, MFC, C++

ABSTRACT

It is an objective of the bachelor thesis to build a computer program that can visualize data files results come from specific mathematical applications in the field of glass furnace modelling. The main feature of the program is a visualization of glass furnace model in a three-dimensional space so it displays particular furnace geometric parts and an optional display of temperature distributions or velocity flows in a basin. The program is realized in C++ language so that provides a data model selection loaded from files and its display through the OpenGL library in a given projection and lightening. The application has simple user interface created by the MFC library, where there is possible to choose input data file, to save a final scene, to set a model position, to adjust a projection and to change various parameters for particular furnace model components at the delimited scene.

Keywords: visualization, projection, geometry, model, data, OpenGL, MFC, C++

Poděkování:

Děkuji Ing. Pavlu Pokornému, Ph.D. za velmi cenné rady, které mi pomohly při realizaci této práce a také za čas věnovaný mé práci.

Motto:

„Každý program, který běží bez závad, je zastaralý.“

Murphyho počítačové zákony

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- § že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- § že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST.....	11
1 TECHNOLOGIE SKLA.....	12
1.1 DEFINICE SKLA.....	12
1.2 DEFINICE TAVENÍ.....	12
1.3 SUROVINY PRO VÝROBU SKLA.....	13
1.4 DRUHY SKLA.....	14
2 SIMULACE NÁVRHU A PROCESŮ V PEČI.....	15
2.1 DISTRIBUCE A OPTIMALIZACE TEPLoty V PEČI.....	15
3 VIZUALIZACE.....	19
3.1 PROMÍTÁNÍ.....	19
3.2 TRANSFORMAČNÍ MATICE.....	21
3.2.1 Rovnoběžné promítání.....	22
3.2.1.1 Pravouhlé promítání.....	22
3.2.2 Kosoúhlé promítání.....	23
3.2.2.1 Kabinetní promítání.....	23
3.2.2.2 Kavalířské promítání.....	23
3.2.2.3 Axonometrické promítání.....	24
3.2.3 Perspektivní promítání.....	25
3.2.3.1 Jednobodová perspektiva.....	26
3.2.3.2 Dvoubodová perspektiva.....	26
3.2.3.3 Třibodová perspektiva.....	27
3.3 GEOMETRIE.....	27
3.3.1 Definice geometrického útvaru.....	28
3.3.2 Rozdělení geometrických útvarů.....	28
3.3.3 Vzájemné polohy geometrických útvarů.....	28
3.3.4 Metrické vlastnosti geometrických útvarů.....	29
3.3.5 Geometrická zobrazení.....	29
3.3.5.1 Klasifikace geometrických zobrazení.....	29
4 APLIKAČNÍ TECHNOLOGIE.....	31
4.1 PROGRAMOVACÍ JAZYK C++.....	31
4.2 KNIHOVNA MFC.....	32
4.3 KNIHOVNA OPENGL.....	33
II PRAKTICKÁ ČÁST.....	36
5 SKLÁŘSKÁ PEČ.....	37
5.1 SPALOVACÍ PROSTOR.....	38

5.1.1	Palivový systém.....	38
5.1.2	Spalinový systém.....	38
5.1.3	Regenerátory.....	38
5.2	VÝTOKOVÉ KANÁLY	39
5.2.1	Míchadla	39
5.2.2	Výpust'.....	39
5.3	TAVÍCÍ VANA	39
5.3.1	Obvodové zdi a dno	40
5.3.2	Zakladač vsázky.....	40
5.3.3	Elektrický přihřev.....	40
5.3.4	Průtokový kanál.....	41
5.3.5	Probublávač	41
5.3.6	Bariéra	42
5.3.7	Izolace	42
6	VIZUALIZAČNÍ PROGRAM	43
6.1	POPIS VÝVOJOVÉHO PROSTŘEDÍ	43
6.2	POPIS PROGRAMU Z POHLEDU PROGRAMÁTORA.....	43
6.2.1	Konstrukce základního okna.....	43
6.2.1.1	Popis implementovaných tříd.....	44
6.2.2	Konstrukce dialogových oken	48
6.2.2.1	Popis implementovaných tříd.....	49
6.2.3	Konstrukce geometrie pece	50
6.2.3.1	Datové soubory.....	50
6.2.3.2	Popis implementovaných tříd.....	51
6.2.4	Konstrukce komponent modelu.....	55
6.2.4.1	Komponenta skalárních a vektorových řezů.....	55
6.2.4.2	Komponenta geometrie modelu.....	56
6.2.4.3	Renderování komponent.....	57
6.2.5	Konstrukce obecných šablon	59
6.3	POPIS PROGRAMU Z POHLEDU UŽIVATELE.....	59
6.3.1	Základní rozhraní programu	59
6.3.2	Zobrazení modelu pece.....	60
6.3.2.1	Dialogové okno Components	60
6.3.2.2	Dialogové okno Model Caption.....	65
6.3.2.3	Dialogové okno Background Color	65
6.3.2.4	Dialogové okno Head-up Light	65
6.3.2.5	Dialogové okno Fixed Light	65
6.3.2.6	Dialogové okno About	66
6.3.3	Rozhraní pro manipulaci se soubory	66
6.3.4	Rozhraní pro nastavení vzhledu programu	66
6.3.5	Rozhraní pro nastavení zobrazeného modelu	67
6.3.6	Ovládání zobrazeného modelu pece.....	67
6.3.7	Datový soubor vytvořené scény.....	68
6.4	VÝSTUPY Z VIZUALIZAČNÍHO PROGRAMU	68

ZÁVĚR.....	71
ZÁVĚR V ANGLIČTINĚ	72
SEZNAM POUŽITÉ LITERATURY	73
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	75
SEZNAM OBRÁZKŮ	78
SEZNAM PŘÍLOH	80

ÚVOD

Již delší dobu je trendem v průmyslové oblasti využívat matematické modely před samotnou realizací stavby nebo optimalizaci technologického procesu. Tyto modely jsou v současnosti velmi sofistikované, založené na mnoha typech algoritmů, výsledky řešení se mnohdy blíží realitě. Ve sklářství to platí také, výsledky matematického modelování umožňují týmu inženýrů diagnostikovat a analyzovat provoz pece s velmi vysokou přesností. Modelování je podpořeno nejpokročilejšími technologiemi v oblasti laboratorních analýz, jako jsou hmotnostní spektroskopie, elektronová mikroskopie a plynová chromatografie, které umožňují identifikovat zdroj vad ve skle. [10] Velmi častým případem je výpočet modelu pece na základě vstupních parametrů a po analýze všech dostupných výsledků sklárna přistoupí ke stavbě nové pece, případně ke generální opravě stávající pece. Tento postup často významně sníží náklady a čas na realizaci projektu, navíc jsou odstraněny nedostatky při návrhu konstrukce pece a optimalizace tavení. Model tedy velmi přesně popíše distribuci tepla vycházející z hořákového systému, který je umístěn ve spalovacím prostoru pece. Na základě těchto informací se dá lokalizovat teplotní maximum v peci a další zóny, které jsou významné pro kvalitní tavbu skloviny.

Cílem této práce je vytvoření vizualizačního programu, který dovede převést datovou interpretaci výsledků z modelování do srozumitelnější grafické podoby, kdy technolog může vidět optimální návrh konstrukce pece a proces tavení skloviny. Tento proces se dá popsat rozložením teplot a rychlostním proděním ve vaně. Program byl vytvořen za účelem využití v komerční firmě pro interní potřebu modelovacího oddělení. Program bude tedy sloužit k nekomerčnímu využití, jen k internímu porovnání a ověření finálních výsledků.

V teoretické části práce je cílem nejprve popsat základní procesy ve sklářské peci, jako je tavení skla, čerění a homogenizace. Dále budou z hlediska vizualizace dat rozebráno promítání, geometrické útvary a transformační matice. Na závěr této části bude popsán jazyk C++ s využitím knihoven MFC a OpenGL, které jsou implementovány v návazné části.

V praktické části bude kladen důraz na popis sklářské pece, tedy na popis jednotlivých částí, poté bude vysvětlen i princip zakládání vsázky. V další části bude z hlediska implementace OpenGL proveden rozbor pole vrcholů, viditelnost hran, nastavení perspektivní kamery, vykreslování osvětleného tělesa a přidání světelných zdrojů. Nakonec bude podrobně rozebrán vizualizační program, jak z pohledu programátora, tak uživatele.

I. TEORETICKÁ ČÁST

1 TECHNOLOGIE SKLA

Sklo se vyrábí v technologickém zařízení nazývaném sklářská pec, ve kterém dochází za daných fyzikálních jevů k chemickým reakcím mezi látkami, které jsou určitého typu a poměru. Důsledkem této interakce je vznik procesu tavení, jehož výsledkem je sklovina, která má specifické vlastnosti, mezi které patří viskozita, barva, atd. Pece mají různou velikost, tvar a tonáž, vyskytují se v různých typech, velikostech a objemu výroby.

1.1 Definice skla

Sklo je homogenní amorfní, tuhý materiál, který má v dlouhodobém hledisku vlastnosti a chování kapaliny. Vyrábí se z viskózní skloviny roztavené ve sklářské peci. Čisté sklo je transparentní, relativně pevný materiál, odolný proti opotřebení, v podstatě inertní a biologicky neaktivní. Může být formováno do všech existujících tvarů. Tyto žádané vlastnosti jej předurčují k velkému množství použití ve většině oborů lidské činnosti. Sklo je však velmi křehké a rozbíjí se na ostré střepy. Tyto vlastnosti mohou být modifikovány nebo i úplně změněny přidáním jiných sloučenin nebo tepelným zpracováním. [19]

1.2 Definice tavení

Tavení je mísení jednotlivých surovin při vysoké teplotě a vytvoření skloviny, je hlavní fází výroby skla. Existuje řada způsobů tavení skla, které závisejí na požadovaném výrobku, jeho konečném použití, na provozu a na komerčních požadavcích. Podle nich se volí složení skla, suroviny, technika tavení, výběr paliva a velikost pece. Tavení je tedy kombinace chemických reakcí a fyzikálních procesů, lze ho rozdělit do několika fází, které vyžadují velmi pečlivou kontrolu, jsou to následujících fáze: [20]

- *Otápění* - konvenčním a nejběžnějším způsobem zavádění tepla do skloviny je spalování fosilních paliv nad lázní kmene, který se kontinuálně zavádí do pece a pak je v roztaveném stavu z pece odebírán. Teplota potřebná k tavení a čěření skla závisí na přesném předpisu, leží mezi 1300 a 1550°C. Při těchto teplotách převládá přenos zářením zvláště z klenby pece, která se zahřívá plamenem až na 1650°C, ale také z plamenů samotných.

- *Primární tavení* - v důsledku nízké tepelné vodivosti kmene je proces tavení zpočátku zcela pomalý, což poskytuje čas k proběhnutí četných chemických a fyzikálních procesů. Protože se materiály zahřívají, vypařuje se vlhkost, některé ze surovin se rozkládají a plyny uvolňující se v surovinách unikají. První reakce (dekarbonizace) probíhá kolem 500°C. Suroviny se začínají tavit mezi 750 a 1200°C. Nejprve se začíná vlivem tavení rozpouštět písek. Oxid křemičitý z písku se váže s oxidem sodným z kalcinované sody a s dalšími látkami z kmene a vznikají křemičitany. Zároveň unikají velká množství plynů z rozkladu hydrátů, uhličitánů, dusičnanů a síranů. Uvolňuje se voda, oxid uhličitý, oxidy dusíku a oxidy síry. Sklovina se nakonec stává transparentní a tavicí fáze je skončena.
- *Čeření a homogenizace* - čeření skla je termín, který označuje snahu odstranit bublinky ze skloviny. Než se sklovina může tvarovat na výrobky, musí být dokonale homogenizovaná a bez bublin, to znamená úplné rozpuštění, stejnoměrnou distribuci všech složek a odstranění všech bublin čeřením.
- *Sejití* - po prvotních etapách tavení a čeření následuje fáze sejití za nižších teplot. Během tohoto procesu se všechny zbývající rozpustné bubliny reabsorbují do taveniny. Zároveň tavenina pomalu chladne na pracovní teplotu mezi 900 a 1350°C. Při tavení kmene tyto kroky probíhají postupně, ale na kontinuálních pecích probíhají fáze tavení současně v různých místech vany. Kmen (vsázka) je zaváděn na jednom konci vany a protéká rozdílnými zónami vany a předpecí, kde probíhá primární tavení, čeření a sejití.

1.3 Suroviny pro výrobu skla

Základními surovinami jsou sklotvorné materiály, jako je křemičitý písek, střepy z výroby, střepy ze spotřebitelského sběru. Křemen má teplotu tání kolem 2000°C, proto se při výrobě přidávají alkalické látky, jako je soda a potaš, které snižují teplotu tání na asi 1000°C. Protože alkálie snižují odolnost skla vůči vodě, což je obvykle nežádoucí, přidává se také oxid vápenatý, který tuto odolnost zlepšuje. Nezbytnou součástí jsou i barviva a odbarviva, mezi ně patří chromitan železitý, oxid železa, oxid kobaltu a selen. [20]

1.4 Druhy skla

Z hlediska chemického složení se sklo dělí do kategorií, zde jsou uvedeny jen ty základní:

- *lehce tavitelné sodnaté* - vhodné na foukání,
- *tvrdé draselné* - vhodné na řezání a broušení, nazývá se křišťál,
- *měkké olovnaté* - vhodné na lití do forem a na výrobu optického skla.

Z pohledu typu výroby a využití se dá sklo ještě rozdělit na:

- *technické* - okenní tabule, obkládačky,
- *užitkové* - poháry, vázy, lustry, sklenice,
- *optické* - skla do brýlí a optických přístrojů,
- *umělecké* - dekorace.



Obrázek 1 Ukázka broušené vázy z olovnatého skla

2 SIMULACE NÁVRHU A PROCESŮ V PECI

Využití počítačové simulace ve sklářském průmyslu se stalo běžnou součástí života, přináší totiž mnoho technických a ekonomických výhod pro firmy, zabývající se stavbou nebo renovací pecí. Simulace také přináší i nesporné výhody technologům ve výrobním provozu, kde se starají o běžný chod pece a ručí za kvalitu roztaveného skla.

Simulace se dá využít pro následující oblasti při návrhu nebo opravy pece:

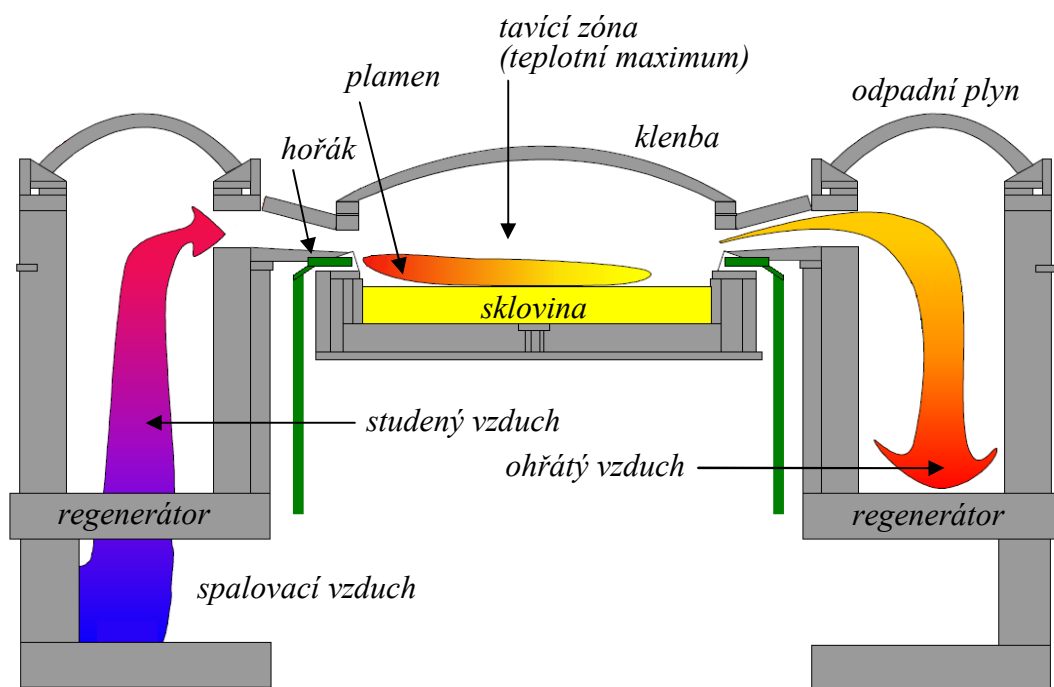
- *zlepšení kvality skloviny,*
- *prodloužení životnosti pece,*
- *optimalizování odběru skloviny,*
- *optimalizování elektrického přívěru,*
- *optimalizace hloubky bazénu a návrh tvaru dna,*
- *nastavení správné pozice probublávače a počet trysek,*
- *nastavení pozic hořáků, typ a jejich počet ve spalovacím prostoru,*
- *snížení spotřeby vstupních energií, tedy snížení nákladů na výrobu,*
- *snížení emise vyprodukovaných škodlivin.*

2.1 Distribuce a optimalizace teploty v peci

Základním fyzikálním jevem v peci je teplo, které vzniká ve spalovacím prostoru pece. Teplo je generováno soustavou hořáků, které jsou umístěny nad hladinou skloviny tak, aby bylo teplo optimálně rozprostřeno na hladinu a tímto docházelo k řádnému protavení hmoty. Pozice hořáků mohou být různé, záleží na typu skloviny a konstrukce pece. Velmi častým případem je příčně plamenné nebo u-plamenné hoření, každé rozložení má své specifické vlastnosti a výhody. Vytvořené teplo z velké části prostupuje do skloviny, tepelné ztráty jsou ovšem také nezanedbatelné, záleží na použitém žáromateriálu a izolaci, která zpravidla pokrývá stěny a klenbu pece, kde je prostup tepla největší. Typická pec je rozdělena do jednotlivých funkčních zón, tedy na oblast zakládání kmene, tavící oblast, kde dochází k protavení skloviny, homogenizaci a čerění. Další zónou je zpravidla bariera, které vrací neprotavenou sklovinu zpět do tavící oblasti. Pokud má sklovina požadované

vlastnosti, pak je odváděna do průtoku a ve finále do pracovní oblasti, kde dochází k ručnímu nebo strojnímu zpracování.

Mechanismů pro distribuci tepla v peci je celá řada. Velmi častým případem je využití odpadního tepla za pomoci regenerátorů. Tyto zařízení jsou určeny pro akumulaci tepla, které přichází ze spalovacího prostoru pece a při změně strany hoření (reverzaci) je toto teplo opět využito v tavicí zóně. Obrázek 2 ilustruje boční pohled na příčně plamennou regenerativní pec se střídavým směrem plamenů, je to názorná ukázka proudění vzduch pecí a hoření, jehož důsledkem dochází k distribuci tepla a tavení.



Obrázek 2 Spalovací systém v příčně plamenné regenerativní peci

Nedá se obecně přesně říci, kolik tepla jde do skla, velmi záleží na typu pece a palivu, navíc do jisté míry tento proces ovlivňuje i oxidér. Distribuce energie (přenos tepla) do skla se tedy pohybuje od 40 do 70%. V závislosti na teplotě hladiny zde mohou být přítomny tři mechanismy tepelného přenosu ve skle, které mohou být oddělené nebo v určité kombinaci.

Jedná se o následující mechanismy:

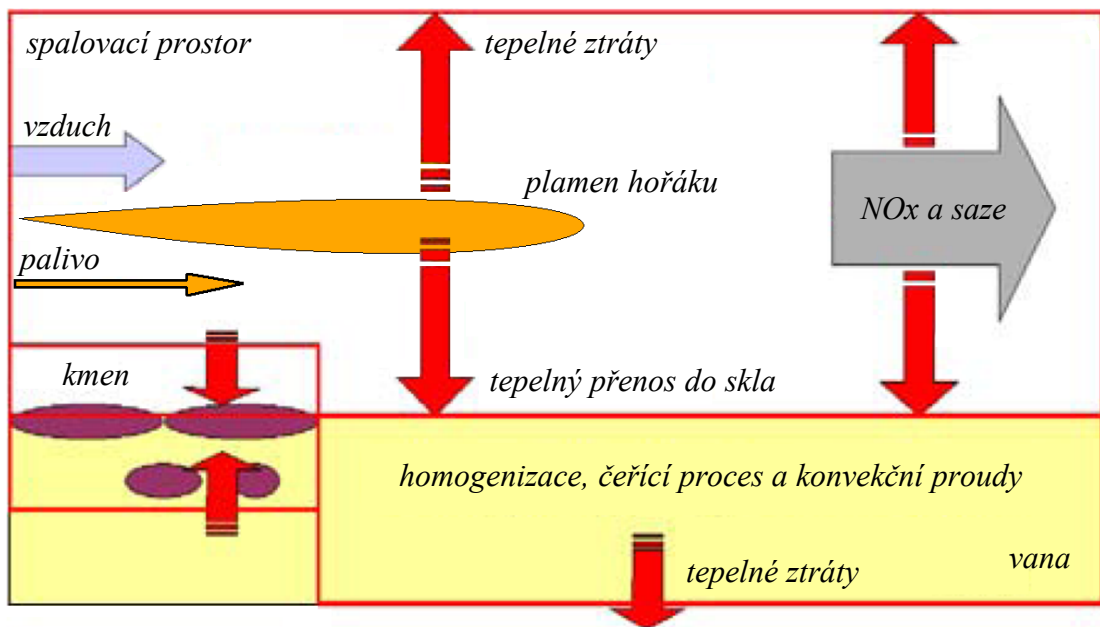
- 1) *Vodivost* - tepelný přenos vodivostí je způsobený tepelnou vibrační energií iontů, které jsou transferovány do jejich sousedních iontů.

- 2) *Radiace* - při teplotě nad 300 °C se tepelná vodivost radiací stává více důležitou a od 500 do 800°C je převládající a stává se více a více důležitou s rostoucí teplotou. Mechanismus radiální výměny je založen na vyzařování sálavého tepla, které je opět částečně pohlceno určitými zónami ve skle.
- 3) *Konvekce* - přenos tepla ve sklovině vlivem proudů. Konvekce má významný vliv na teplotní distribuci ve sklovině, na promíchávání a dobu zdržení.

Kombinace těchto mechanismů v závislosti na teplotě mohou být:

- do 300°C - důležitá je jen vodivost,
- od 300°C a výše - přítomná je kombinace vodivosti a radiace,
- od 800°C a výše - kombinace radiace a konvekce, vodivost je méně důležitá.

Proudění i teplotní distribuce mohou být popsány fyzikálními zákony zachování. Obrázek 3 popisuje distribuci tepla ve spalovací části pece a tavící vaně.

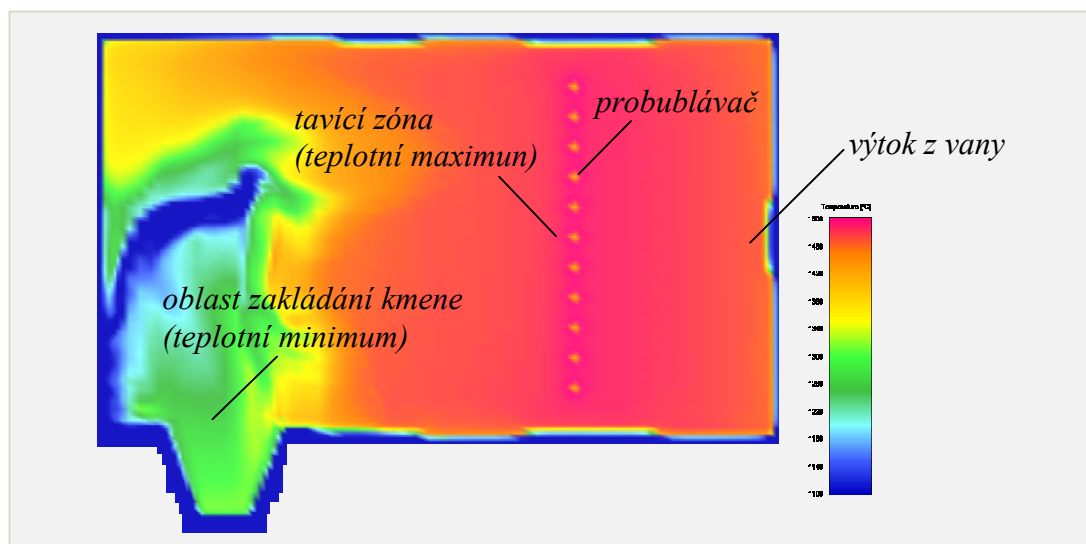


Obrázek 3 Distribuce tepla ve spalovacím prostoru a v tavící vaně

Z pohledu matematického modelu pece jsou teplota a rychlost proudění sklovin vyjádřeny ve formě diskretních trojrozměrných polí, které se řídí zákony zachování a mohou být vyjádřeny v podobě parciálních diferenciálních rovnic. Model aproximuje parciální diferenciální rovnice numerickou konečnou diferenční metodou.

Výsledný model dokáže označit nestabilní nebo kritická místa v peci, které lze poté optimalizovat z pohledu teplotního zatížení, rychlosti a směru proudění skloviny, kvality tavení, odběru, atd. Významná je teplotní optimalizace, která má dopad na kvalitu skla a náklady na provoz pece, proto se této oblasti věnuje náležitá pozornost. Z toho plyne, že je zbytečné zahřívat sklovinu více, než je nutné, správně využívat odpadní teplo v podobě vracení zpět do tavící části. Pece v současnosti používají moderní, velmi sofistikované, řídicí systémy, které dokážou na základě definovaných stavů a parametrů vyhodnotit aktuální situaci a optimalizovat teplotu v peci v podobě úpravy hoření. Na nestabilitu teploty v peci má vliv mnoho faktorů, nejčastějším je množství zakládaného kmene, odběr skloviny z pece a spaliny. Pokud řídicí systém zaregistruje snížení nebo zvýšení teploty, pak provede korekci v řádu desetin stupňů, v případě větších teplotních změn, poté v řádu stupňů. Mechanismus nastavení a měření je zpravidla realizován PID regulací a servoventily, které ovládají průtok plynu a vzduchu, směřující do spalovacího systému.

Obrázek 4 popisuje konkrétní modelovou studii, kde je vidět rozložení teplot jen několik centimetrů pod hladinou skloviny. Teplotní profil je umístěn v rovině XY, pec je tedy vidět z půdorysného pohledu, kde modrá barva značí nejnižší teplotu a červená nejvyšší. Model odkrývá podrobnosti o teplotách, na základě těchto informací lze provést poměrně přesnou optimalizaci teplot tak, aby na výstupy byla požadovaná kvalita skloviny.



Obrázek 4 Příklad rozložení teplot ve vaně

3 VIZUALIZACE

Vizualizací se rozumí zobrazování skutečnosti, jejichž výsledky jsou znázorněny vnímané prostřednictvím zrakových receptorů. Vizualizace úzce souvisí s uplatňováním zásady názornosti. S vizualizací se setkáváme v mnoha oblastech - stavebnictví, technice, strojírenství, atd. Je při tom využíváno moderních metod - počítačového modelování. [19]

3.1 Promítání

Aby bylo možné prostorové objekty vizualizovat, je nutné vyřešit promítání objektu na dvourozměrných zobrazovacích zařízeních, jako je monitor, televizor, plátno, atd. Převedení trojrozměrných objektů do podoby dvourozměrné je řešeno mechanismy promítáním.

Studiem promítacích metod a aplikacemi se zabývá vědní obor *deskriptivní geometrie*. Ten rozlišuje celou řadu postupů určených k tomu, abychom i z dvourozměrného obrazu získaného promítáním mohli zpětně odvodit prostorové vztahy, mezi základní patří vzdálenosti a úhly. V počítačové grafice však vystačíme s omezenou množinou promítacích metod, což je rovnoběžné a středové promítání na jednu průmětnu.

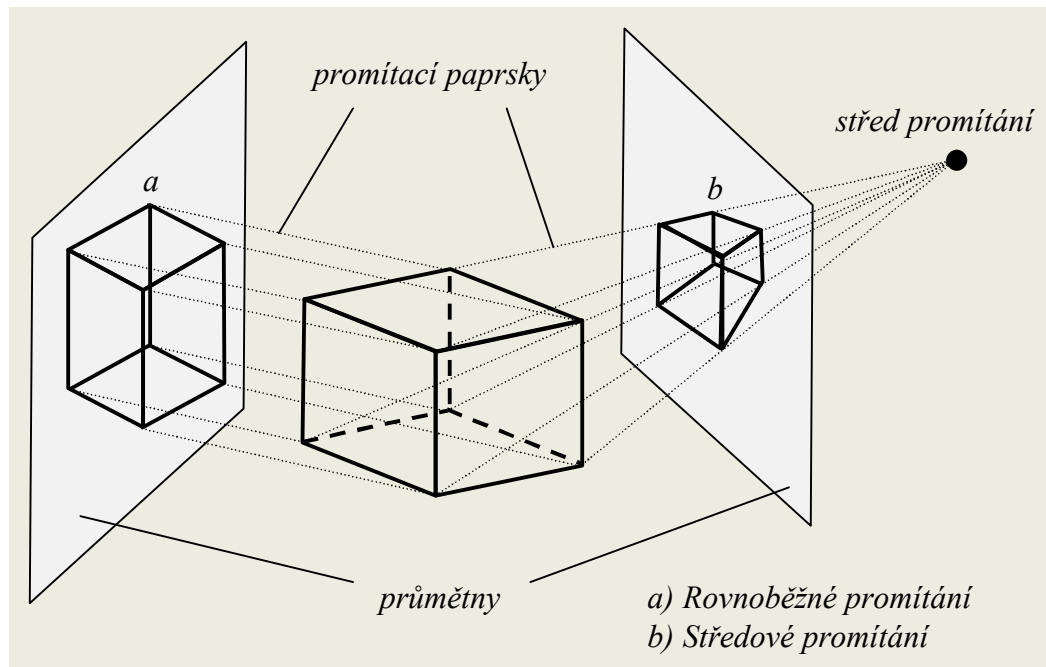
Promítání definuje dva následující pojmy:

- *Promítací paprsek* - přímka vedoucí promítaným bodem, jejíž směr závisí na dané promítací metodě,
- *Průmětna* - plocha v prostoru, na kterou dopadají promítací paprsky. V místě dopadu vytvářejí průmět, což je výsledný obraz na této ploše.

Při zobrazování prostorových objektů následuje po promítání další zpracování dat, jako je nalezení zakrytých a viditelných částí objektů, vyhodnocení jejich barvy, nanesení textur, průhlednost, atd. Z důvodu snazšího zpracování těchto kroků se stalo zvykem používat jako průmětnu rovinu xy , resp. jakoukoliv rovinu kolmou na osu z . Budeme tedy metody promítání uvádět s ohledem na průmětnu rovnoběžnou s rovinou xy . Promítání do jiné průmětny můžeme vždy převést pomocí transformací (otočením, posunutím, atd.) prostorových objektů na uvedený případ. [9]

Promítání je děleno na *rovnoběžné* a *středové*, podle rovnoběžnosti, resp. různoběžnosti promítacích paprsků. Na následujícím obrázku jsou vidět názorné ukázky těchto promítání. Z objektu, který se má zobrazit, jsou směrem doleva vedeny rovnoběžné promítací paprsky.

Jejich průsečíky s průmětnou jsou body výsledného obrazu. K průmětně vpravo vedou promítací paprsky směřující ke středu promítání. V tomto bodě se všechny tyto paprsky stýkají. Na průmětně se opět zobrazí objekt tvořený body, které vznikly protnutím promítacích paprsků s průmětnou. Při tomto druhu promítání tedy závisí tvar a velikost zobrazeného objektu na umístění vzhledem k průmětně.



Obrázek 5 Příklad rovnoběžného a středového promítání

Pokud je průmětna rovinná, prostorové promítací paprsky se promítají do úseček v rovině, nejsou tedy zakřiveny. U prostorových objektů se nemusí provádět promítání všemi body. V případě úseček, které tvoří hrany těles, stačí podrobit promítání pouze koncové body a ty spojit rasterizací v průmětně.

Po promítání následuje další důležité zpracování dat, jedná se hlavně o nalezení viditelných částí objektů, vyhodnocení jejich barvy a nanesení textur. Pro zjednodušení se často jako průmětna používá rovina xy , která je kolmá na osu z .

3.2 Transformační matice

Maticí se obecně nazývá speciálně zvolená tabulka reálných nebo komplexních čísel

$$\begin{pmatrix} a(1,1) & a(1,2) & \dots & a(1,n) \\ a(2,1) & a(2,2) & \dots & a(2,n) \\ \dots & \dots & \dots & \dots \\ a(m,1) & a(m,2) & \dots & a(m,n) \end{pmatrix}.$$

Číslo $a(i,j)$ je prvek matice, která obsahuje n sloupců a m řádků, pro které platí, že $1 \leq i \leq m$ a $1 \leq j \leq n$. Rozměrem matice je $m \times n$, což se zapisuje jako $A(m \times n)$, kde index i ukazuje, na kterém řádku se prvek nachází a index j ukazuje na sloupec matice. [3]

V prostorové počítačové grafice se používají transformační matice o rozměrech 4×4 , které nesou informace v trojrozměrné dimenzi. Jako vzor si uvedeme matici

$$\begin{pmatrix} RotXx & RotYx & RotZx & 0 \\ RotXy & RotYy & RotZy & 0 \\ RotXz & RotYz & RotZz & 0 \\ MoveX & MoveY & MoveZ & 1 \end{pmatrix}.$$

Jak je patrné z definice, poslední řádek (mimo poslední sloupec, tj. prvky s názvem obsahujícím *Move*) slouží k určení hodnot posunutí v jednotlivých osách. Složitější je to však s prvky pro rotaci. Prvky v této submatici 3×3 (v definici označené částí názvu *Rot*) definují rotaci původní osy ve směru osy jiné (např. v prvku označeném *RotXy* určíme hodnotu pootočení osy x na osu y). Poslední sloupec matice je zde nejen proto, aby se daly jednotlivé transformační matice mezi sebou násobit, resp. aby se bylo možné na sebe skládat jednotlivé transformace, ale také pomocí nich můžeme nastavit perspektivu. [8]

Díky transformačním maticím můžeme jednoduchým způsobem definovat a používat nejrůznější geometrické transformace, jako je posunutí, otočení, změna měřítka, souměrnost a zkosení a to ve všech osách prostoru. Tyto matice stačí v určitém pořadí vynásobit se stávající projekční maticí a získáme výsledné promítání.

3.2.1 Rovnoběžné promítání

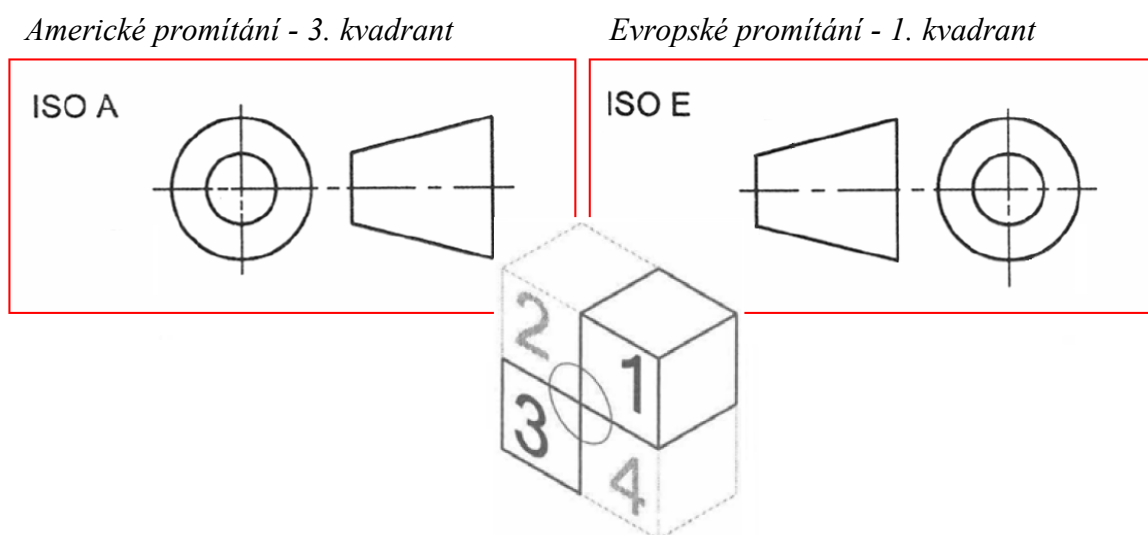
Při tomto způsobu promítání jsou všechny promítací paprsky rovnoběžné. Podle toho, jaký svírají úhel s průmětnou, dělíme rovnoběžné promítání na *pravoúhlé* pro úhel 90° a *kosouhlé* pro jiné úhly, než je 90° .

Rovnoběžné promítání je typické pro technické aplikace, neboť zachovává rovnoběžnost. Vzdálenost průmětny od promítacích objektů neovlivňuje velikost průmětů. [9]

3.2.1.1 Pravoúhlé promítání

Pravoúhlé (ortografické) promítání se velmi často používá v technických výkresech, ale i v oblasti počítačové grafiky.

Pravoúhlé promítání se v technické praxi dělí podle mezinárodních norem ISO na dvě metody, které se liší umístěním objektu vůči pozorovateli a průmětnám. Jejich název je odvozen podle umístění v soustavě navzájem kolmých rovin. Soustava rovin je rozdělena na čtyři kvadranty. Pro promítání se využívá pravidlo prvního a třetího kvadrantu. Každý kvadrant umožňuje promítání celkem na šest rovin. Promítání v prvním kvadrantu se používá v evropských zemích, a proto se nazývá jako *evropské*, promítání ve třetím kvadrantu se nazývá *americké*. [1]



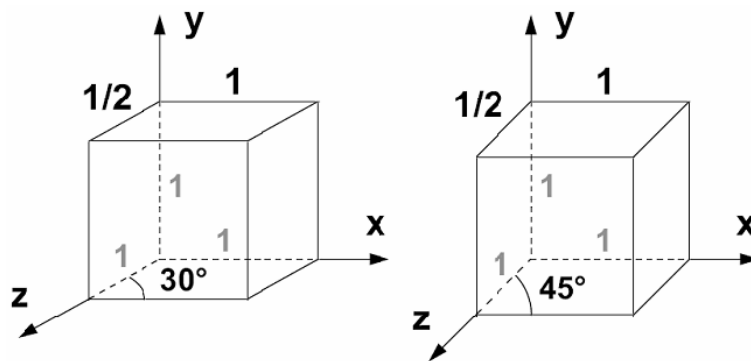
Obrázek 6 Orientace promítání v prvním a třetím kvadrantu.

3.2.2 Kosoúhlé promítání

U tohoto druhu promítání jsou promítací paprsky navzájem rovnoběžné, nejsou však kolmé k průmětně. Průmětna zůstává rovnoběžná s některou z hlavních rovin, např. s rovinou xy , yz nebo xz . Dva nejčastěji používané druhy kosoúhlého promítání se nazývají *kabinetní* a *kavalířské promítání*.

3.2.2.1 Kabinetní promítání

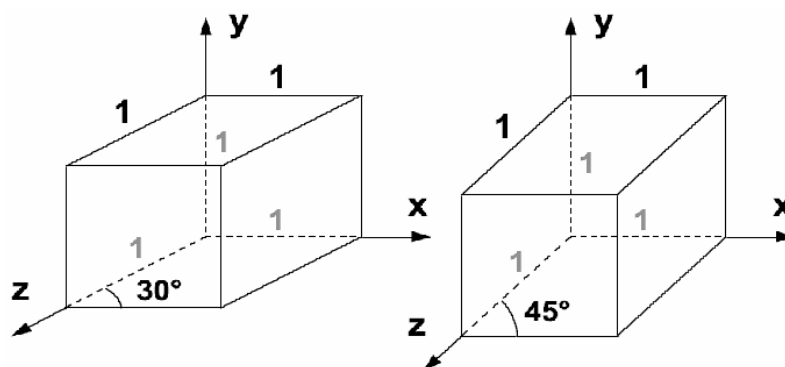
Směr projekce s průmětnou svírá $63,4^\circ$, úsečky kolmé na průmětnu se zkracují na polovinu.



Obrázek 7 Příklad kabinetního promítání

3.2.2.2 Kavalířské promítání

Směr této projekce svírá s průmětnou 45° , úsečky rovnoběžné s průmětnou i kolmé na ní mají tedy stejnou délku na obrazovce.



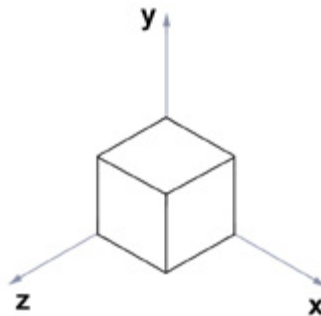
Obrázek 8 Příklad kavalířského promítání

3.2.2.3 Axonometrické promítání

Axonometrické promítání používá projekční roviny, které nejsou rovnoběžné s hlavními osami. Typy tohoto promítání jsou *izometrie*, *dimetrie*, *trimetrie* a *pravoúhlá axonometrie*.

3.2.2.3.1 Izometrie

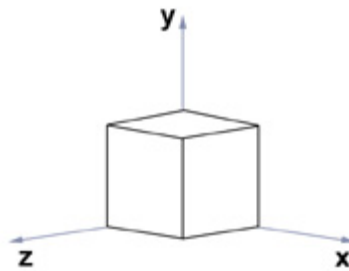
Izometrie je pseudo trojrozměrný metrický systém z ptačího pohledu. V podstatě se jedná o trojrozměrný systém bez perspektivy. [16]



Obrázek 9 Příklad izometrie

3.2.2.3.2 Dimetrie

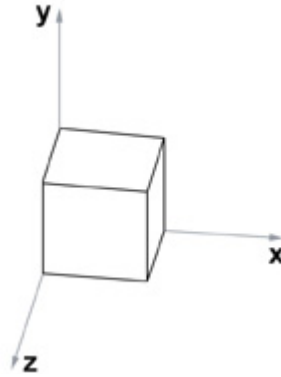
Je to promítání, které se podle ISO nazývá kabinetní axonometrie. Dva rozměry se vynášejí nezkráceně a třetí, v závislosti na orientaci souřadného systému, zkrácený na polovinu. [17]



Obrázek 10 Příklad dimetrie

3.2.2.3.3 Trimetrie

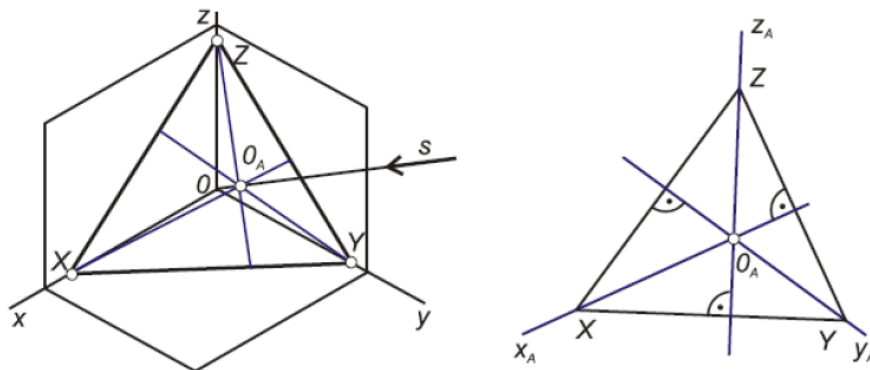
Z pohledu trimetrie platí, že v každém směru promítnuté osy je potřeba při porovnání uplatnit odlišné měřítko.



Obrázek 11 Příklad trimetrie

3.2.2.3.4 Pravoúhlá axonometrie

Půdorysna, nárysna a bokorysna se protne rovinou ρ , která neprochází počátkem a protíná všechny tři osy. Průsečíky roviny ρ s osami označíme X , Y , Z . Trojúhelník XYZ je vždy ostroúhlý. Rovina ρ je *axonometrickou průmětnou*, do které budeme pravoúhle promítat. Trojúhelníku XYZ říkáme *axonometrický trojúhelník*. Tento trojúhelník se zobrazuje vždy ve skutečné velikosti, neboť leží v axonometrické průmětně. [18]



Obrázek 12 Příklad pravoúhlé axonometrie

3.2.3 Perspektivní promítání

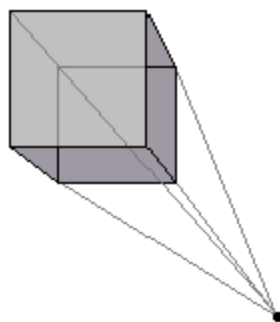
Pokud je střed promítání vlastní, mluvíme o perspektivním promítání a promítací přímky procházejí tímto středem promítání (promítací paprsky vycházejí z jediného bodu). Perspektivní promítání respektuje optický model, který vyjadřuje lidské vidění reálného světa. Modeluje proporcionální změnu předmětů při vzrůstající vzdálenosti od pozorovatele. Poskytuje dobrý prostorový vjem na rovinné průmětně. [19]

Charakteristickým rysem středového promítání je to, že nezachovává rovnoběžnost úseček. Průměty úseček, rovnoběžných v trojrozměrném prostoru, jsou obecně mimoběžné a střetávají se ve středu promítání. Výjimkou jsou prostorové úsečky, ležící v rovině rovnoběžné s průmětnou. [9]

Průmětna může mít libovolnou polohu, ale se rozlišují tři případy perspektivy, které odpovídají orientaci průmětny vůči osám souřadnicového systému, je to *perspektiva jednobodová*, *dvoubodová* a *trojbodová*.

3.2.3.1 Jednobodová perspektiva

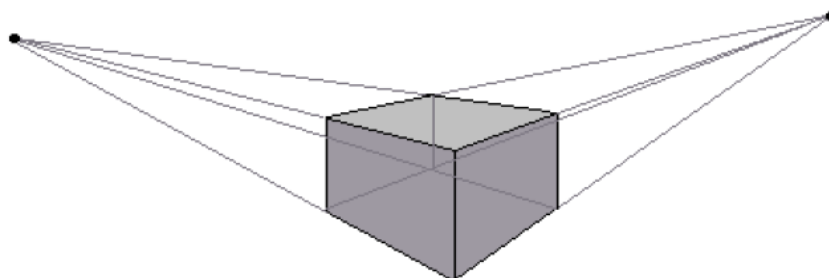
Jednobodová perspektiva vzniká, když průmětna protíná jedinou souřadnicovou osu. Všechny úsečky kolmé na průmětnu míří do jediného bodu, který se nazývá *hlavní úběžník*. Taktéž je tato perspektiva označována jako tzv. *jednouběžníková* či *jednostředová perspektiva*. [3]



Obrázek 13 Jednobodová perspektiva

3.2.3.2 Dvoubodová perspektiva

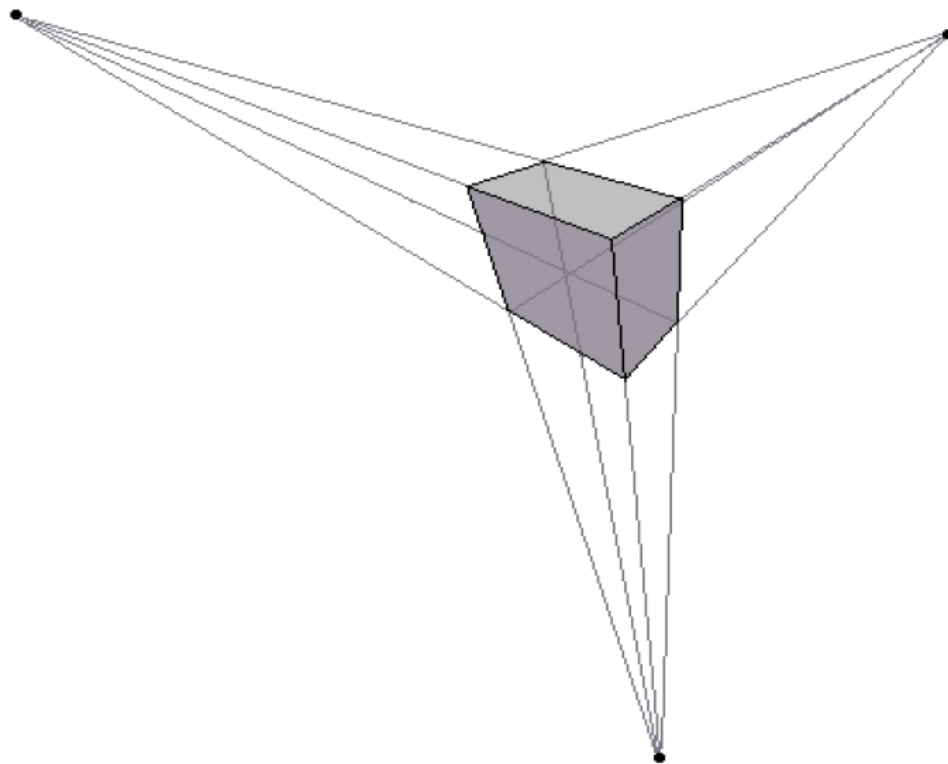
Tato perspektiva vzniká, pokud průmětna protíná dvě ze souřadnicových os. Hrany osově orientovaných kvádrů směřují do dvou hlavních úběžníků. Tento druh promítání bývá nazýván také jako tzv. *dvouúběžníková* či *dvoustředová perspektiva*. [3]



Obrázek 14 Dvoubodová perspektiva

3.2.3.3 Třibodová perspektiva

Tato projekce je nejobecnější případ, který vzniká, pokud průmětna promítá všechny tři souřadnicové osy. Protážením hran osově orientovaných kvádrů můžeme vysledovat tři hlavní úběžníky. [3]



Obrázek 15 Třibodová perspektiva

3.3 Geometrie

Geometrie je jedna z matematických věd, která se původně zabývala vlastnostmi typu tvar a velikost, a vzájemnými vztahy mezi geometrickými útvary prostorových těles, ploch, bodů, přímek a rovin. Geometrie bývá považována za jeden z prvních matematických oborů vůbec. Přestože je geometrie nejstarší oblastí matematiky, dodnes se vyvíjí. V modernějším pojetí se geometrie zabývá vlastnostmi prostoru, různými algebraickými strukturami na topologických objektech. [19] Jeden ze základních pojmů, které tato matematická věda definuje, je pojem *geometrický útvar*.

3.3.1 Definice geometrického útvaru

Geometrický útvar je množina bodů na přímce, v rovině nebo v prostoru. Jeho základní vlastností je tvar, ale jeho velikost není podstatná. Uzavřenou oblast v rovině nazýváme obrazcem, uzavřenou oblast v prostoru nazýváme tělesem. Hranici obrazce označujeme jako obvod, hranici tělesa jako povrch. [19]

3.3.2 Rozdělení geometrických útvarů

Geometrické útvary lze dělit podle různých vlastností, jsou to: [19]

- *Základní geometrické útvary* - bod, přímka, rovina a prostor
- *Lineární geometrické útvary* - přímka, polopřímka a úsečka
- *Rovinné geometrické útvary* - obrazce jako je polorovina, mnohoúhelníky, kružnice, kruh, kuželosečky, křivky a útvary vymezené křivkami
- *Prostorové geometrické útvary* - tělesa to této skupiny patří hranoly (krychle, kvádr), válec, jehlan, kužel, koule a další
- *Množiny všech bodů dané vlastnosti*

3.3.3 Vzájemné polohy geometrických útvarů

Vzájemné polohy geometrických útvarů se vyjadřují analyticky, touto formou se nabývá *analytická geometrie*. Vzájemná poloha dvou nebo více geometrických útvarů se určuje řešením soustavy rovnic.

Mezi nejdůležitější případy vzájemné polohy dvou geometrických útvarů patří: [19]

- *Vzájemná poloha dvou bodů* - vzdálenost
- *Vzájemná poloha bodu a přímky*
- *Vzájemná poloha bodu a kružnice*
- *Vzájemná poloha dvou přímek* - rovnoběžky, různoběžky, mimoběžky a další
- *Vzájemná poloha dvou kružnic* - vnější dotyk, vnitřní dotyk a další
- *Vzájemná poloha přímky a kružnice* - tečna, sečna a další
- *Vzájemná poloha přímky a roviny*

- *Vzájemná poloha bodu a roviny*
- *Vzájemná poloha rovin*

3.3.4 Metrické vlastnosti geometrických útvarů

Metrické vlastnosti jsou určovány definicí míry daného geometrického útvaru. Mírou v geometrii nazýváme zobrazení množiny bodů, které tvoří určitý geometrický útvar, na množinu \mathbf{R}_0^+ načme-li daný geometrický útvar jako X , pak jeho míru lze zapsat jako $\mu(X)$.

Platí, že podobné geometrické útvary mají stejné míry. Pokud dva geometrické útvary X a Y nemají žádný společný vnitřní bod, pak platí

$$\mu(X \cup Y) = \mu(X) + \mu(Y)$$

Jednotky jsou dány volbou geometrického útvaru, kterému přiřadíme jednotkovou velikost. Základními definovanými mírami v geometrii jsou: [19]

- *Délka* - délka úsečky, vzdálenost útvarů
Obvod obrazce - obvod trojúhelníku, čtverce, kruhu
- *Velikost úhlu, rovinný úhel a prostorový úhel*
- *Obsah obrazce* - obsah trojúhelníku, čtverce, kruhu
Povrch tělesa - povrch krychle, válce, koule
- *Objem tělesa* - objem krychle, válce, koule
- *Trigonometrie* - goniometrické funkce, věta sinová, věta kosinová

3.3.5 Geometrická zobrazení

Geometrickým zobrazením je zobrazení, které každému bodu A útvaru U přiřazuje právě jeden bod A' útvaru U' . Bod A je tzv. *vzor* a bod A' se označuje jako *obraz*.

3.3.5.1 Klasifikace geometrických zobrazení

Zobrazení se klasifikují *podle zachovávaných se vlastností a podle dimenze prostoru*.

3.3.5.1.1 Podle zachovávaných se vlastností

Podle toho, které vlastnosti se při geometrickém zobrazení zachovávají a které se mění, lze geometrická zobrazení rozdělit na: [19]

- *podobné zobrazení*, kam řadíme např. stejnoolehlost - podobná zobrazení lze považovat za speciální případ shodných zobrazení,
- *shodné zobrazení*, kam patří např. posunutí, rotace a další - shodná zobrazení lze považovat za speciální případ afinních zobrazení,
- *afinní zobrazení* - zobrazení zachovávající rovnoběžnost přímek,
- *topologické zobrazení* - zachovává se pouze příslušnost bodu k dané křivce.

3.3.5.1.2 Podle dimenze prostoru

Geometrická zobrazení lze rozdělit podle dimenze transformovaného prostoru a podle toho, zda vzor i obraz mají stejnou dimenzi. [19]

- *Dimenze vzoru i obrazu jsou stejné:*
 - *lineární* - např. posunutí bodu po přímce,
 - *rovinné* - např. rotace kolem bodu,
 - *prostorové*,
 - *víceměrné*.
- *Dimenze vzoru a obrazu jsou různé:*
 - *projektivní zobrazení* - např. rovnoběžné promítání, axonometrie, perspektiva

4 APLIKAČNÍ TECHNOLOGIE

Tato kapitola se zaměří z teoretického pohledu na zvolené technologie, které budou použity v praktické části práce a implementovány v programu.

4.1 Programovací jazyk C++

C++ je objektově orientovaný programovací jazyk, který vyvinul Bjarne Stroustrup a další v Bellových laboratořích AT&T rozšířením jazyka C. C++ podporuje několik programovacích stylů (paradigmat) jako je procedurální programování, objektově orientované programování a generické programování, není tedy jazykem čistě objektovým. V současné době patří C++ mezi nejrozšířenější programovací jazyky. [19]

Mezi hlavní charakteristické rysy tohoto jazyka patří:

- *Objekty* - jsou pojaty jako přirozené rozšíření datových struktur jazyka C o možnost vkládání členských funkcí. C++ umožňuje řídit viditelnost složek objektů pro ostatní části programu. Pro objekty je možná vícenásobná dědičnost.
- *Dědičnost* - hlavní myšlenka dědičnosti je znovupoužitelnost, to znamená, že můžeme vytvářet nové třídy založené na třídě, která již byla definována, místo toho abychom museli znovu psát již jednou napsaný kód jen s jinými typy proměnných. Díky dědičnosti je možné napsat kód jednou pro obecnější typ a poté ho používat pro všechny jeho potomky.
- *Asociace* - jestliže třída A asociuje třídu B, potom instance třídy A nějakým způsobem musejí vědět o instancích třídy B. Pokud zná objekt nějaký jiný objekt, znamená to, že jej může používat, zasílat mu zprávy a podobně.
- *Agregace* - znamená obsažení, jestliže třída A agreguje třídu B, potom instance třídy A v sobě obsahují instance třídy B.
- *Šablony* - rozšiřují znovupoužitelnost kódu, neboť umožňují napsat kód se zcela obecným datovým typem, proto jsou vhodným nástrojem pro vývoj aplikace. Jsou užitečné především pro základní typy, které v C++ nejsou objekty.

4.2 Knihovna MFC

Microsoft Foundation Class Library (Microsoft Foundation Classes nebo MFC) je knihovna, která zabaluje části Windows API do ucelených C++ tříd, které zajišťují použití a plnou kompatibilitu s většinou platformem OS Windows. Třídy spravují většinu objektů ve Windows, a tedy předdefinovávají například okna nebo kontrolní panely.

Knihovna MFC byla poprvé představena v roce 1992 společně s kompilátorem C/C++ 7.0 pro 16-bitové verze OS Windows. Jazyk C++ postupem času nahradil C pro vývoj komerčního software a C/C++ 7.0 byl první kompilátor společnosti Microsoft, který jazyk C++ plně podporoval.

MFC používá *Afx* jako předponu mnoha funkcí, maker a standardních předkompilovaných hlavičkových souborů, např. *stdafx.h*. Během krátkého období se knihovna MFC stala populární a mnoho programátorů jí začalo zařazovat jako Application Framework Extensions (Rozšíření pracovního prostředí) neboli ve zkratce AFX. Avšak název Microsoft Foundation Class (MFC) byl přijat později, a proto nezasáhl do názvů funkcí, maker a hlavičkových souborů.

Knihovna MFC 8.0 byla vydána společně s Visual Studiem 2005. MFC není součástí volné edice Visual C++ 2005 verze Express.

Význam MFC byl ovšem oslaben s příchodem .NET Framework. Alternativou k MFC je WTL (Windows Template Library). Kompilátor Visual C++ Express dokáže zkompileovat MFC aplikace, avšak nezahrnuje podporu IDE (Integrated Development Environment) profesionálních edicí.

Mezi stěžejní pojmy, které prostředí MFC definuje, patří následující:

Framework - je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API (Application Programming Interface), návrhové vzory nebo doporučené postupy při vývoji. Cílem frameworku je převzetí typických problémů dané oblasti, čímž se usnadní vývoj tak, aby se návrháři a vývojáři mohli soustředit pouze na své zadání.

Device context - je používán k definování atributů textu a obrázků, které jsou zobrazovány na obrazovce nebo tisknuty tiskárnou. Aktuální kontext udržuje GDI (Graphics Device Interface). Device context (DC), které nastavuje strukturu, je získáno před výstupem a je

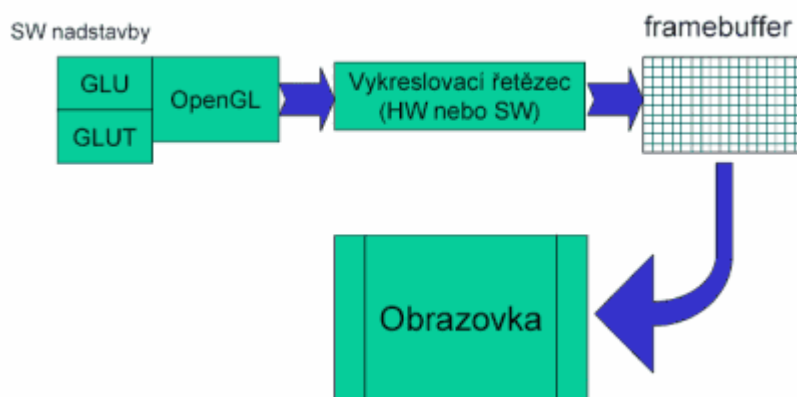
zapisován a uvolňován po částech, tak jak byly zapsány. DC, jako většina GDI objektů, je nepropustný, nepřístupuje k datům přímo, můžeme je předat funkcím, které tyto data zpracují, buď ke kreslení něčeho, ke zjištění informace o něčem, nebo ke změně objektu nějakým způsobem. [19]

4.3 Knihovna OpenGL

OpenGL (Open Graphics Library) je průmyslový standard specifikující multiplatformní rozhraní API pro tvorbu aplikací počítačové grafiky. Používá se při tvorbě počítačových her, CAD programů, aplikací virtuální reality či vědeckotechnické vizualizace apod.

Implementace OpenGL existují pro prakticky všechny počítačové platformy, na kterých je možno vykreslovat grafiku. Kromě implementací vestavěných v grafickém hardware (na grafické kartě) existují také softwarové implementace, které umožňují používat OpenGL i na hardwaru, který ho sám o sobě nepodporuje, ale obvykle nabízejí nižší výkon. Příkladem takové implementace je open source knihovna Mesa 3D, která ovšem z licenčních důvodů nemůže být označena jako implementace OpenGL, ale pouze jako implementace API, které je velmi blízké OpenGL.

Základní funkcí OpenGL je vykreslování do obrazového rámce (framebufferu). Umožňuje vykreslování různých základních primitiv (bodů, úseček, mnohoúhelníků a obdélníků pixelů) v několika různých režimech. Z framebufferu lze získat pouze barevnou informaci a tu je možné následně zobrazit na obrazovce, obrázek 16 popisuje tento proces.



Obrázek 16 Proces vykreslování přes framebuffer

Veškerá činnost OpenGL se řídí vydáváním příkazů pomocí volání funkcí a procedur, kterých OpenGL definuje cca 250. V OpenGL se nepoužívá objektově orientované programování. Jednotlivá primitiva jsou definována pomocí vrcholů, každý z nich definuje bod, koncový bod hrany nebo vrchol mnohoúhelníku. Každý vrchol má přiřazena data (obsahující souřadnice umístění bodu, barvy, normály a texturovací souřadnice). Rozhraní OpenGL je založeno na architektuře klient-server, program (klient) vydává příkazy, které grafický adaptér (server) vykonává. Díky této architektuře je možné, aby program fyzicky běžel na jiném počítači než na tom, na kterém se příkazy vykonávají, a příkazy se předávaly prostřednictvím počítačové sítě. [19]

Prostředí této knihovny zahrnuje mnoho různých specifických pojmů z oblasti grafiky, zde jsou uvedeny jen základní, se kterými může čtenář přijít do kontaktu.

Transformace - je v OpenGL vnímána jako transformace souřadnic, je reprezentována maticí 4x4. Základní matice jsou dvě: GL_MODELVIEW (určuje posun a rotaci objektu v prostorových souřadnicích) a matice GL_PROJECTION (slouží k definici projekce - transformace bodu podle zvoleného promítání). Jsou zde však i další matice, např. GL_TEXTURE (pro mapování textur). Existuje obecný mechanismus pro úpravy matic - funkcí *glMatrixMode* se zvolí aktuálně zpracovávaná matice, funkce *glLoadMatrix(M)* ji nahradí maticí *M*, *glLoadIdentity* ji nahradí identitou, *glMultMatrix(M)* ji vynásobí maticí *M* zprava (složí transformace). *glPushMatrix* uloží aktuální matici na stack (zásobník) a *glPopMatrix* ji z ní vyzvedne. Existují zde užitečné funkce *glTranslate*, *glRotate* a *glScale*, které přenásobí GL_MODELVIEW matici odpovídající příslušné transformaci, a funkce *glOrtho* a *glFrustum*, které matici GL_PROJECTION nastaví na hodnotu odpovídající příslušnému ortogonálnímu nebo perspektivnímu promítání. Pokud je nutné natáčet kameru, pak se na začátek nastaví opačné transformace do GL_MODELVIEW. Na závěr se ještě výhledové okno definované maticí GL_PROJECTION namapuje na skutečnou oblast OpenGL drawing contextu pomocí hodnot nastavených funkcí *glViewport*, která akceptuje souřadnice *x*, *y*, *w*, *h* udávající polohu a rozměr výhledu (již reprezentován v pixelech) v daném drawing contextu aplikace. [14]

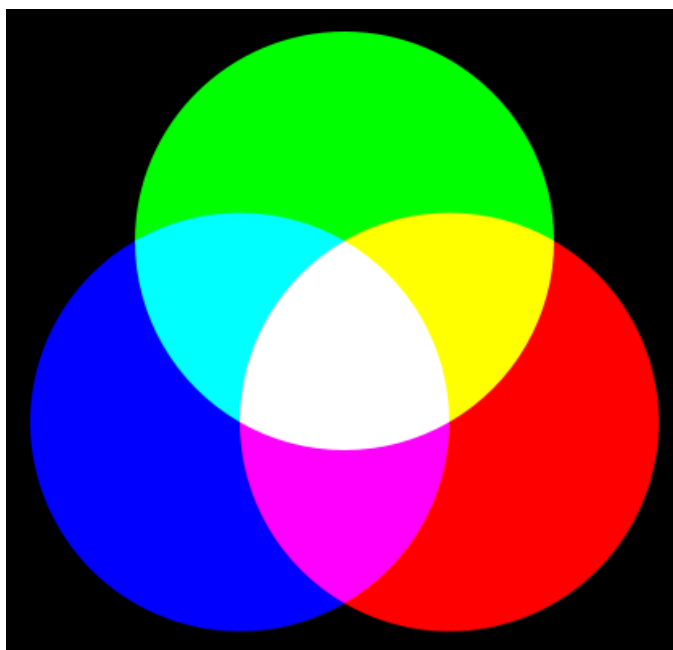
Osvětlení - je definovaný zdroj světla do určitého místa na scéně, v OpenGL je umožněno definovat maximálně 8 různých světelných zdrojů, přičemž pro každý z nich lze nastavit mnoho parametrů, jako barvu, intenzitu, druh, umístění nebo tvar světelného kužele. Při

použití osvětlení je potřeba pro každý vrchol zadat normálový vektor, využívá se především pro výpočet stínování tělesa.

Renderování - je proces, při němž ze zadaných dat vzniká cílový obraz. V případě grafického editoru je výstupem 3D obrázek, v případě webového prohlížeče je to vysázená webová stránka. [15]

Vertex - bod v prostoru, patří do množiny základních grafických primitiv. Je definován souřadnicí x , y a z . Vzhledem k jeho singularitě (nepřítomnost rozměrů) nemá smysl u něho kromě souřadnic zaznamenávat další informace, jako je rotace v každé ze základních os, faktor zvětšení nebo zmenšení, atd.

RGB - je to barevný model složený z barev *červená-zelená-modrá*. Je to aditivní způsob míchání barev používaný ve všech monitorech a projektorech, jde o míchání vyzařovaného světla, nepotřebuje tedy vnější světlo, proto monitor zobrazuje i v naprosté tmě. Obrázek 17 reprezentuje aditivní míchání barev ze třech barevných složek. [19]

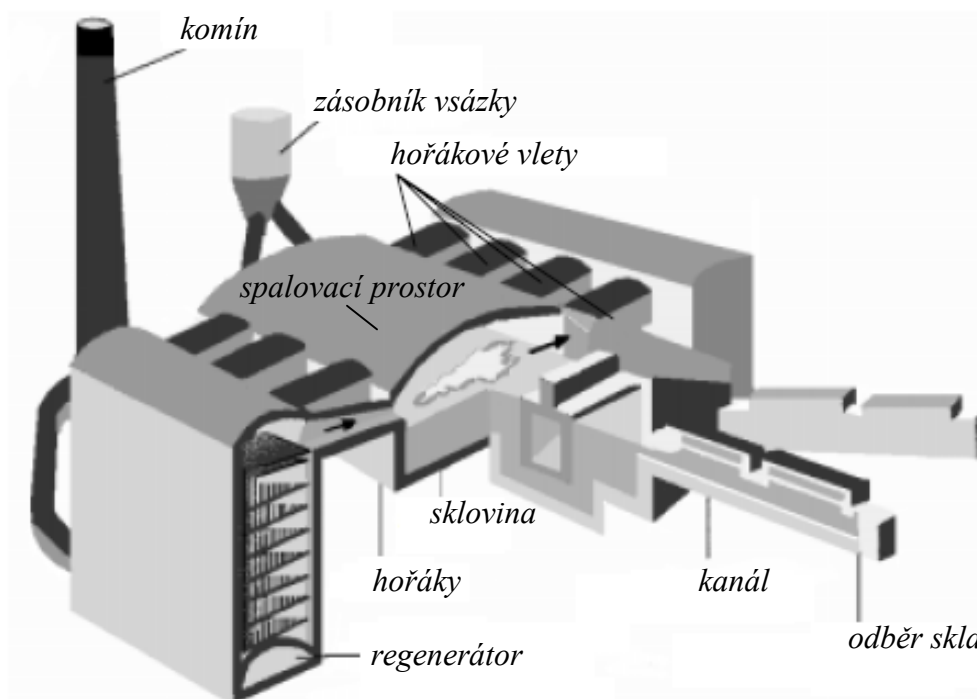


Obrázek 17 Ukázka aditivního barevného míchání ze tří složek

II. PRAKTICKÁ ČÁST

5 SKLÁŘSKÁ PEC

V teoretické části této práce již bylo řečeno, co je to sklářská pec, jaké typy skloviny z hlediska chemického složení se v ní taví, a byly zde i popsány základní druhy produkce. V této praktické části bude výklad zaměřen na popis jednotlivých částí pece, a jelikož je typů pecí celá řada a není možné zde všechny popsat, tak bude výklad obsahovat jen popis jednoho typu, který patří mezi základní, jak z pohledu velikosti a tvaru konstrukce pece, tak z pohledu výroby skla a následného zpracování. V běžném životě se všichni setkávají s lahvemi, které mají určitý tvar, velikost a barvu, výklad tedy bude zaměřen na pec produkující lahvové sklo. Obrázek 18 znázorňuje a popisuje model takové pece, včetně okolních zařízení, které jsou nezbytné např. pro spalovací a spalinový systém, následnou úpravu a zpracování skloviny v automatizovaných strojích. [7]



Obrázek 18 Model regenerativní pece na výrobu lahvového skla

Pec je zpravidla velká, technologicky složitější, obsahující mnoho okolních zařízení pro obsluhu vsázky, paliva, vzduch, spalin, atd. Denní odběr skloviny z pece je např. 100 tun, zpracování je strojové, kde na koncích výtokových kanálů jsou umístěny vyfukovací stroje, které paralelně každou minutu produkují desítky nových lahví, směřující do chladících pecí a k následnému dalšímu zpracování.

Bylo již řečeno, že pec se skládá z mnoha zařízení, předmětem této práce je jen podrobnější popis vany, tedy části pece, kde dochází k tavení skloviny. Další části zde budou zmíněny jen ve stručnosti, ne proto, že by byly méně důležité, ale z důvodu, že by tato práce narostla do větších rozměrů.

Ještě před detailním popisem tavící vany, zde bude ve stručnosti proveden výčet dalších částí pece a zařízení, bez kterých by pec nemohla optimálně fungovat.

5.1 Spalovací prostor

Spalovací prostor v peci je oblast, kde dochází k hoření paliva, jehož důsledkem je uvolňování tepelné energie, která přestupuje to tavené skloviny.

5.1.1 Palivový systém

Palivový systém je zpravidla složitý mechanismus, jehož cílem je dodat požadovanou tepelnou energii do skloviny. Systém se skládá ze soustavy hořáků, jejichž pozice jsou předně dány tak, aby prostup do hladiny skloviny byl optimální. Hořáky jsou ovládány řídicím systémem, který hlídá teplotu v určitých místech pece, a podle zpětné vazby nastavuje dodávané palivo. Hořáky mohou být plynové, olejové, případně v různých kombinacích, hlavním kritériem je cena dané komodity a typ tavené skloviny.

5.1.2 Spalinový systém

Pec musí zahrnovat spalinové hospodářství, k tomuto účelu slouží systém, monitorující množství vyprodukovaných spalin a provádí jejich filtraci ve speciálních zařízeních. Na systém filtrace je v současnosti kladen velký důraz a to především z ekologického hlediska.

5.1.3 Regenerátory

Regenerátor je tepelný výměník, který se skládá z velké komory a mřížoví ze žáruvzdorných cihel. Regenerativní pec má zpravidla dva takové regenerátory, viz obrázek 16, to z důvodu střídavého hoření, tedy levé nebo pravé strany. Mechanismus tohoto typu hoření se nazývá reverzace, hlavní výhodou je ve využívání odpadního tepla k tavení.

5.2 Výtokové kanály

Výtokové kanály slouží k odvádění protavené skloviny, splňující požadované vlastnosti, do automatických strojů, v kterých dochází k formování lahví pod určitým tlakem.

5.2.1 Míchadla

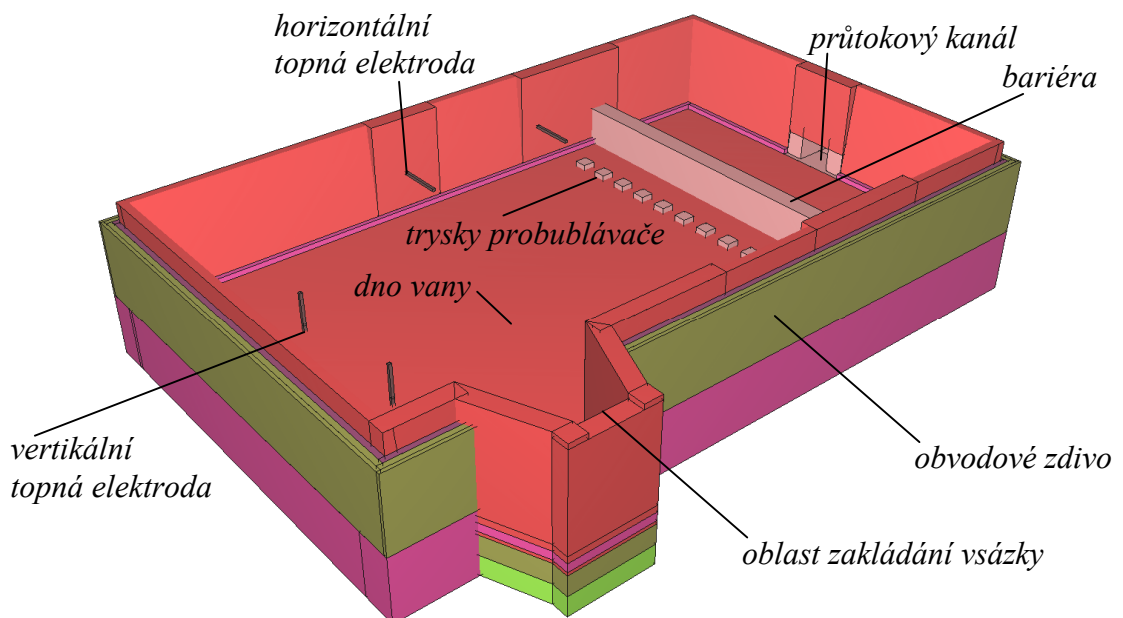
Jelikož výtokové kanály mají topné zóny, a v určitých případech i chladicí zóny, tak je někdy důležité sklovinu promíchávat. V kanále tedy mohou být nainstalovány i míchadla, jejichž cílem je sklovinu homogenizovat.

5.2.2 Výpusť

Výpusť je umístěna na konci kanálu, má daný průměr a sklovina do ní vstupuje při určité teplotě a rychlosti. Tento proces je důležitý, výsledná kapka skla musí mít přesnou váhu, aby láhev splňovala předepsaná kritéria.

5.3 Tavící vana

Tavící vana je oblast v peci, kde dochází k samotnému tavení skloviny, a aby byl tento proces optimální, tedy požadovaná kvalita za minimální náklady na provoz, tak jsou ve vaně instalovány různé prvky. Obrázek 19 popisuje rozložení těchto prvků ve vaně.



Obrázek 19 Tavící vana regenerativní pece

5.3.1 Obvodové zdi a dno

Obvodové zdi a dno tvoří základní konstrukci celé vany. Jelikož pec běží v nepřetržitém provozu minimálně 5 let, musí mít žáruvzdorné cihly požadované vlastnosti, tedy pevnost, tepelnou odolnost, atd. Vlivem chemické reakce skloviny s povrchem žáruvzdorného materiálu dochází časem k jeho úbytku. Životnost pece tedy určuje kvalita použitého materiálu, typ tavené skloviny a denní výtěžnost.

5.3.2 Zakladač vsázky

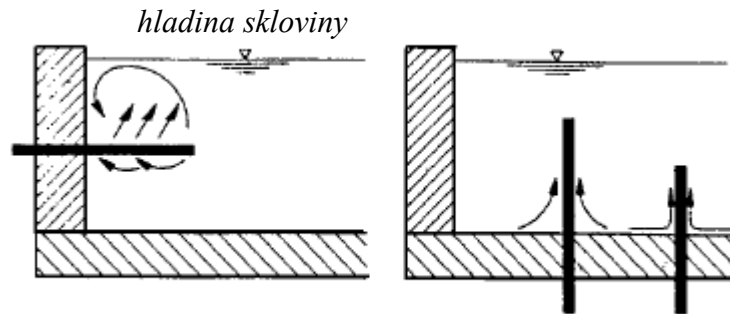
Zakládací mechanismus je zpravidla složité zařízení, to z důvodu, že vykonává práci v daných časových intervalech a v různých směrech. Vsázka (kmen) je uložen v zásobníku poblíž pece, zakladač ho postupně odebírá a posouvá do pece.



Obrázek 20 Ukázka zakladače vsázky

5.3.3 Elektrický přihřev

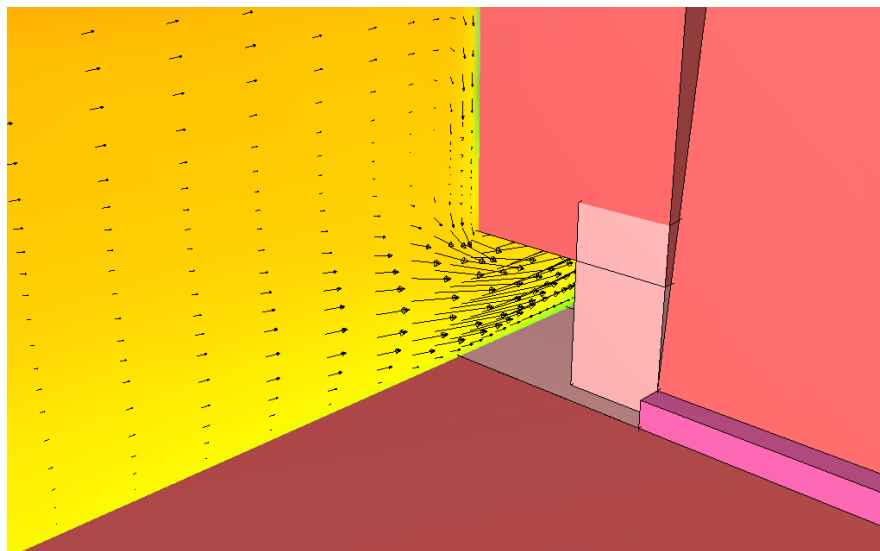
Elektrický přihřev je soustava přesně rozmístněných elektrod ve dně a v některých případech ve zdech vany. Tento systém má mnoho výhod, mezi hlavní patří lepšímu protavení skloviny, optimalizace proudění, tedy zvýšení, snížení rychlosti proudění nebo i vychýlení. Obrázek 21 ukazuje možné polohy elektrod a směr proudění skloviny. [7]



Obrázek 21 Ukázka rozmístění elektrod ve vaně

5.3.4 Průtokový kanál

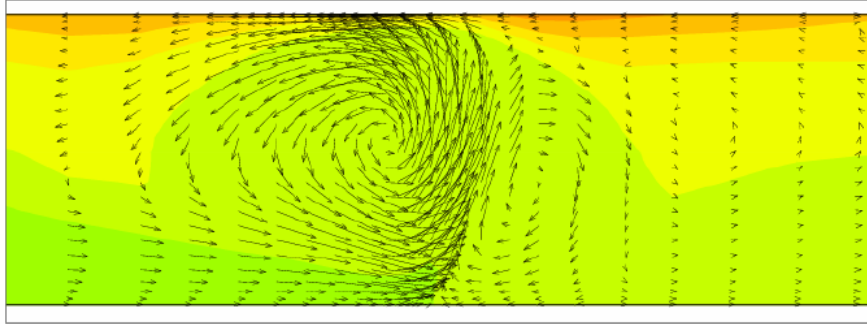
Průtokový kanál je místo na konci vany, kde sklo je již homogenizováno, má požadovanou kvalitu a je už připraveno ke vstupu do rozvodných kanálů. Obrázek 22 ukazuje detail v oblasti tohoto kanálu v podobě simulace proudění skloviny.



Obrázek 22 Ukázka simulace proudění v průtokovém kanálu

5.3.5 Probublávač

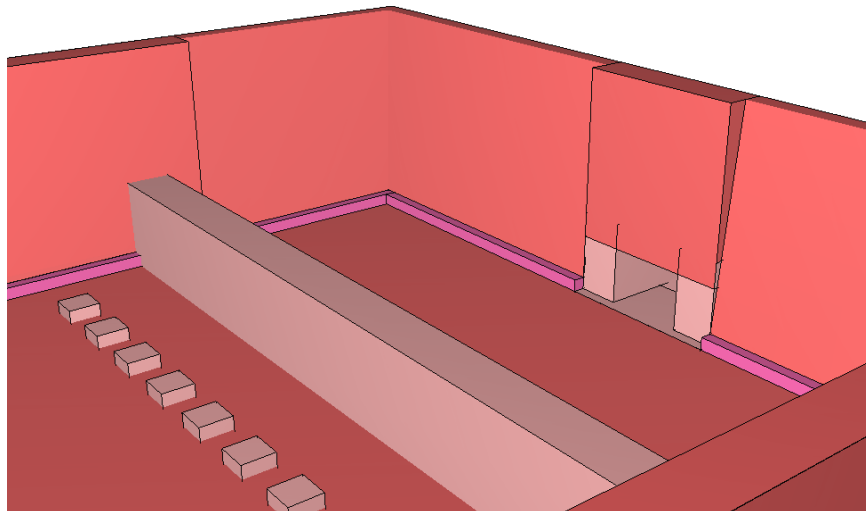
Probublávač je zařízení, které se skládá ze soustavy trysek, umístěných v tavicí části vany v oblasti teplotního maxima. Tryskami ze dna proudí vzduch určitou rychlostí, důsledkem je lepší proces homogenizace, také lépe dochází k tepelnému přenosu mezi studenější sklovinou u dna a teplejší u hladiny. Obrázek 23 ilustruje simulaci probublávače v řezu roviny XZ, kde vzduch prostupuje ze dna k hladině skloviny.



Obrázek 23 Ukázka simulace proudění v oblasti probublávače

5.3.6 Bariéra

Funkce bariéry spočívá v tom, aby se studenější sklovina u dna dostala blíže k hladině, a byla opět protavena před opuštěním tavící části. Obrázek 24 ukazuje detail typické bariéry, která je umístěna před průtokem, kde sklovina proudí z vany do dalších částí.



Obrázek 24 Pohled na umístění bariéry ve vaně

5.3.7 Izolace

Tepelná izolace je materiál, který zabraňuje průniku tepla z teplotně rozdílných oblastí. Jelikož jsou v peci tepelné ztráty poměrně velké, především v místech klenby spalovacího prostoru, obvodových zdí a rozvodných kanálů, tak se používá vhodná tepelná izolace, která částečně snižuje tepelný prostup těmito plochami. Častým typem izolace je skelná vata, je pokládána přímo na povrch žáruvzdorných cihel.

6 VIZUALIZAČNÍ PROGRAM

Tato kapitola se zaměří na vizualizační program, nejprve z pohledu programátora, budou tedy popsány hlavní třídy (objekty) a metody pro zobrazení aplikačního okna, nastavení pohledu a vizualizace datových polí. Dále bude z pohledu uživatele popsáno prostředí, parametry a funkce programu, jako je načtení vstupního souboru z disku, zobrazení geometrie modelu pece, teplotních nebo vektorových řezů, změna nebo přidání osvětlení, nastavení barevných stupnic, rotace, posuv a další.

6.1 Popis vývojového prostředí

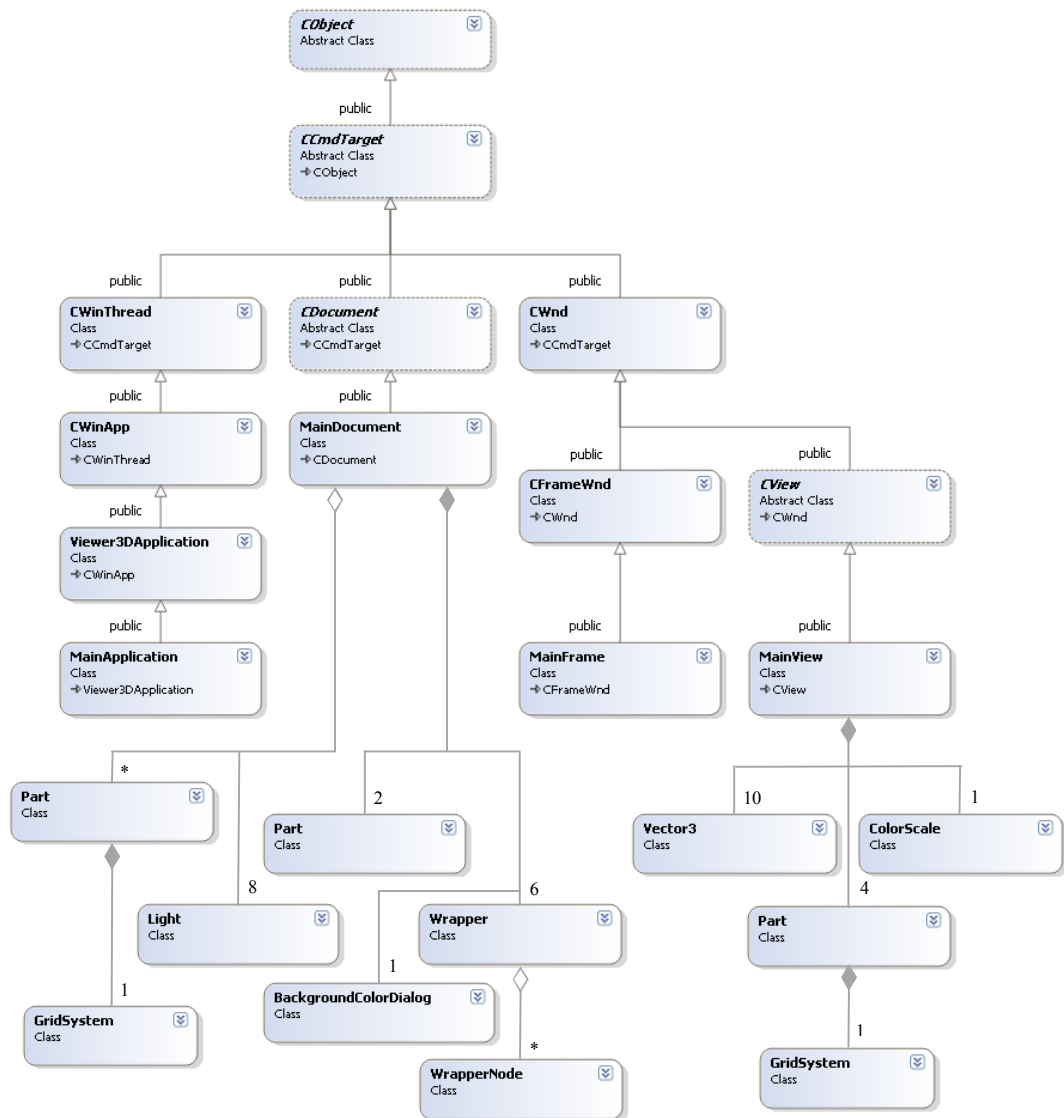
Vizualizační program se jmenuje *Viewer3D*, byl napsán v jazyce C++ ve formě OOP (Objektově Orientované Programování), za pomoci rozsáhlých knihoven MFC a OpenGL. Implementace proběhla ve vývojovém prostředí MS Visual Studio Professional 2008. Program je vytvořen a zkompileován pro platformu Microsoft Windows.

6.2 Popis programu z pohledu programátora

Pokud je program rozsáhlejší, složitější a je rozdělen do více částí, pak je vhodné použít nějaký standardní popisovací model, aby se v programu zorientoval i člověk, který se přímo neúčastní vývoje dané aplikace. Jedním z častých modelů je popis pomocí diagramů v jazyce UML (Unified Modeling Language), kde se definuje program z pohledu hierarchie tříd. Z diagramu jsou patrné báze a odvozené třídy, kompozice a agregace. Jelikož je program složitější, obsahuje mnoho tříd, je rozdělen do jednotlivých částí podle významu.

6.2.1 Konstrukce základního okna

Program implementuje SDI (Single Document Interface), z této definice tedy plyne, že každá instance programu může zobrazit jen jeden dokument uvnitř hlavního rámce. Knihovna MFC nabízí třídy pro vytvoření základní konstrukce programu, jako je vytvoření základního rámce aplikace, dokumentu a pohledu. Obrázek 25 reprezentuje UML diagram hlavního okna programu, kde jsou rozvrženy jednotlivé třídy a vztahy mezi nimi. Třídy začínající velkým písmenem C jsou součástí knihovny MFC, jsou použity jako báze třídy, které implementovala šablona pro vygenerování základní konstrukce programu.



Obrázek 25 Diagram základní konstrukce programu

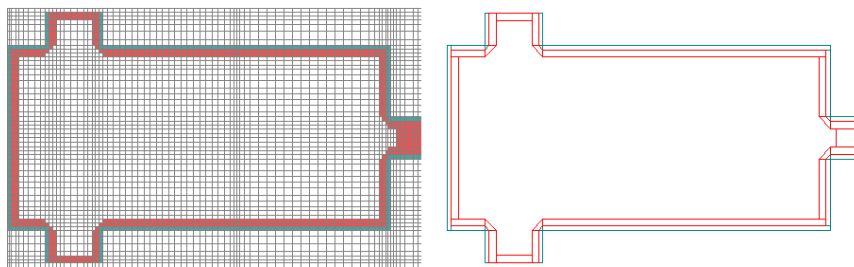
6.2.1.1 Popis implementovaných tříd

Každý definovaný objekt přináší do programu určitou funkcionalitu a vlastnosti, důsledkem zapouzdření, dědičnosti a agregace lze využít instance těchto tříd v různých vrstvách programu. Následuje popis jednotlivých tříd z hlediska jejich významu:

- *CObject* - je kořenová abstraktní básová třída pro většinu tříd MFC. Tato třída nabízí mnoho užitečných vlastností, které jsou efektivně využity i v tomto programu, jako je např. podpora serializace, informace o běhu tříd (run-time class information) a diagnostika objektů (object diagnostic output).

- *CCmdTarget* - je abstraktní básová třída, definující architekturu mapování zpráv (message-map architecture). Mapa zpráv směřuje jednotlivé generované příkazy a zprávy členských metodám, které byly napsány programátorem za účelem jejich zpracování. Třída je zde využita ke zpracování příkazů, přicházející např. z menu, tlačítek nebo akceleračních kláves.
- *CWinThread* - třída způsobí to, že aplikace poběží ve vlastním vlákně. Každá další instance třídy *CWinThread* vytvoří vlastní běžící vlákno v rámci jedné aplikace. Tato třída podporuje dva typy vláken, jsou to pracovní vlákna (worker threads) a vlákna uživatelského rozhraní (user-interface threads).
- *CWinApp* - je básová třída, z které se odvozuje aplikační objekt Windows. Tato třída poskytuje členské metody pro inicializaci aplikace (a každé její další instance) a pro běh samotné aplikace. MFC dovoluje v rámci jedné aplikace použít jen jeden objekt, který je odvozen od třídy *CWinApp*.
- *Viewer3DApplication* - je odvozená třída od básové *CWinApp*, vytvářející ve svém konstruktoru instanci programu *Viewer3D*, ve které pak dochází k vytvoření finální podoby a nastavení aplikace.
- *MainApplication* - inicializuje program v oblasti haldy (dynamická paměť), vytváří zde instanci třídy *CSingleDocTemplate*, do které přes konstruktorem jako argument vstupuje identifikátor *IDR_MAINFRAME*, třída *MainDocument*, *MainFrame* a *MainView*. Zmíněné třídy budou popsány v další části této práce. *CSingleDocTemplate* je šablona, inicializující hlavní menu, dokument, rámec typu SDI a pohled, tedy oblast pro renderování výsledného modelu. V poslední fázi inicializace dochází k zobrazení programu a nastavení implicitních hodnot.
- *CDocument* - je abstraktní básová třída, která poskytuje základní funkcionalitu pro uživatelsky definované třídy dokumentu. Jedná se podporu operací typu vytvoření dokumentu, jeho zavedení do paměti a uložení obsahu do souboru.
- *MainDocument* - třída je odvozena od básové *CDocument*, definující hlavní dokument. V programu je využito této třídy v podobě zpracování příkazů z menu a akceleračních kláves, načtení dat se souboru, uložení výsledné scény na disk, vytvoření nového dokumentu a vymazání pohledu za účelem zobrazení nového modelu.

- *GridSystem* - třída generuje grid systém (diskrétní geometrii) modelu, což je základní prostorová síť uzlů, na které se pak mapují teploty nebo výchozí body vektorů, určující směr a rychlost skloviny. Členská metoda *read* načte z datového souboru *ggri.d* do pole *grid* jednotlivé difference mezi uzly v každé dimenzi (ve směru osy *x*, *y* a *z*). Metoda *update_abs_grid* vypočítá absolutní vzdálenost uzlů ve všech směrech. Obrázek 26 ilustruje půdorysný pohled na základní geometrii, která je důležitá především pro zobrazení modelu vany, a diskrétní geometrii, sloužící pro výpočet a zobrazení fyzikálních jevů v peci.



Obrázek 26 Ukázka diskrétní a základní geometrie modelu

- *Part* - je třída, která je implementována v jichých třídách aplikace, formou agregace a kompozice (speciální typ agregace). Instance typu *Part* popisují jednotlivé komponenty modelu. Aplikace nabízí dva typy zmíněných komponent, první je zobrazení geometrie modelu tavicí vany a skalárního/vektorového pole (teplota/proudění skloviny) v podobě řezu ve zvolené rovině. Uživatelské rozhraní dovoluje vytvořit libovolné množství instancí těchto komponent. Každá taková instance má definované vlastnosti, které se nastavují prostřednictvím tříd, odvozených od základní třídy *CDialog*.
- *WrapperNode* - objekt této třídy uchovává libovolné hodnoty, jako např. souřadnice zobrazeného modelu, natočení, zoom a míchání barev. Tato třída je použita v objektu typu *Wrapper*, v deklarovaném objektu *nodes* třídy *CArray*, jako typ objektu a reference na tento objekt, který je uložen někde v paměti.
- *Wrapper* - definuje členská data a tři metody, jejichž je využíváno v mnoha dalších třídách. Metoda *get* získává libovolné hodnoty z pole *nodes* typu *CArray*, metoda *set* naopak hodnoty do pole ukládá. Třída dále definuje metodu *Serialize* a proměnné *range1*, *range2*, sloužící k uchování libovolných hodnot pro nastavení některých objektů, např. rozsahu hodnot pro smíchání dvou barev.

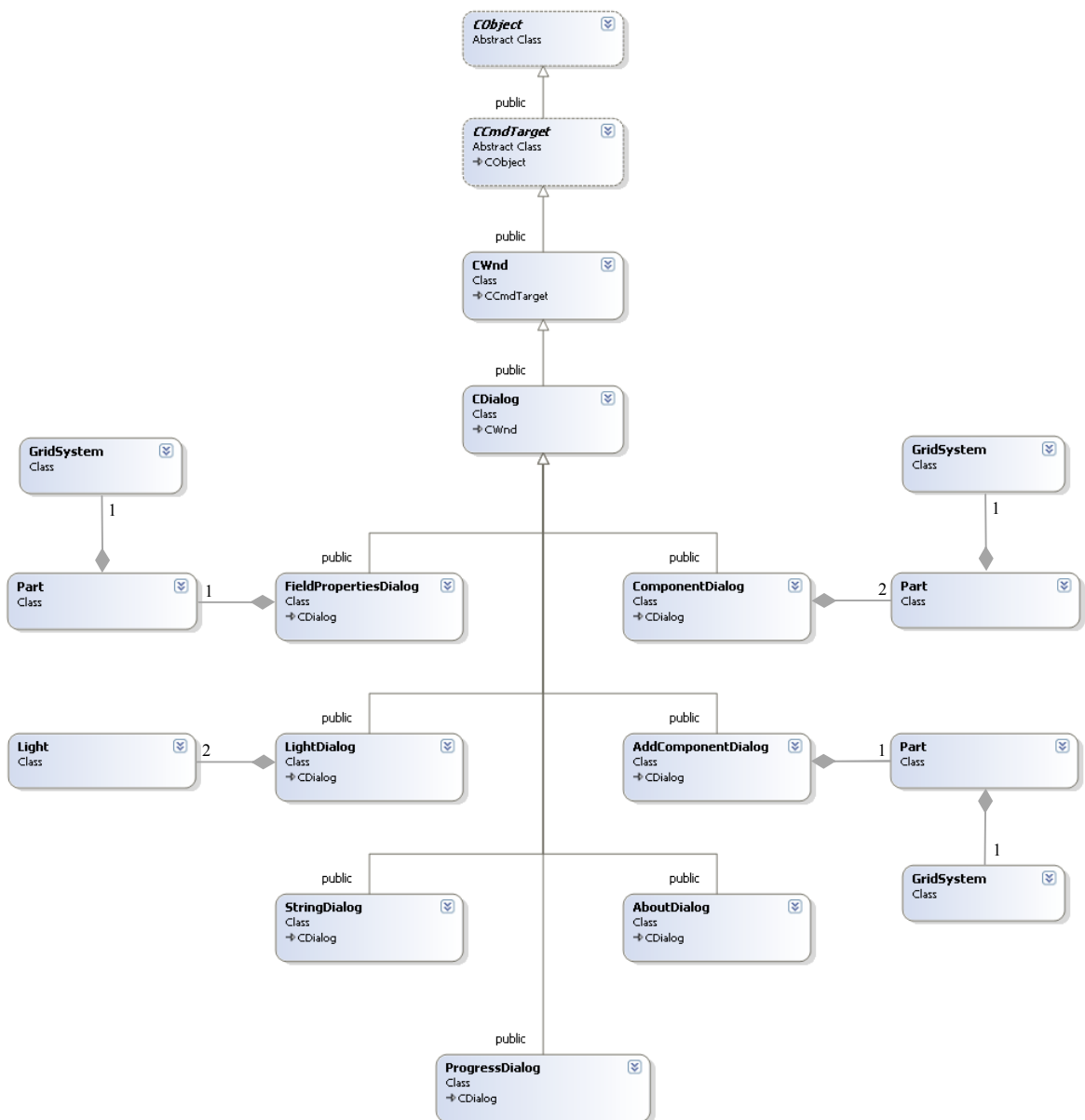
- *Light* - třída zajišťuje informace o pozici v prostoru a barvě světelných zdrojů, které jsou volitelně aplikovány na zobrazený model pece. Světelné zdroje jsou realizovány použitím funkce *glLightfv* z knihovny OpenGL, byl zvolen difuzní typ osvětlení (GL_DIFFUSE). Zdroje jsou celkem tři, první je ve výši očí (head-up light), nelze tedy měnit jeho pozici osvětlení, a další dvě lze libovolně nastavit, tyto zdroje se nazývají jako pevné (fixed lights).
- *BackgroundColorDialog* - tato třída byla vytvořena za účelem nastavení barvy pozadí zobrazovací plochy. V členské metodě *edit* se generuje instance třídy *CColorDialog*, (systémová paleta barev), kde lze vybrat přednastavenou barvu, případně namíchat vlastní barvu z palety nebo nastavením RGB hodnot.
- *CWnd* - třída poskytuje základní funkcionalitu pro všechny třídy typu *window classes* v knihovně MFC. Windows okno je datová struktura, která je vytvářena v *CWnd* metodou *Create* a mazána z paměti pomocí virtuálního destruktora.
- *CFrameWnd* - poskytuje funkcionalitu pro *Windows* okna s jedním rámcem (Windows single document interface) a pop-up okna (pop-up frame windows), včetně metod pro ovládání oken. Pokud je definována aplikace jako SDI, pak je třída základního rámce odvozena od základové třídy *CFrameWnd*.
- *MainFrame* - třída základního okna (rámce) aplikace, kde se inicializuje vzhled okna, hlavní menu (IDR_MAINFRAME), panel nástrojů (ToolBar), panel zpráv (StatusBar) a další části aplikace. V panelu zpráv jsou průběžně zobrazovány informace o poloze modelu (coordinates), natočení (heading a pitch), perspektivě (zoom) a podrobné informace o položkách, které nabízí aplikační menu.
- *CView* - třída poskytuje funkcionalitu pro třídy typu *view classes*. Pohled (view) je připojen k dokumentu a má roli zprostředkovatele mezi dokumentem a uživatelem. *CView* renderuje vzhled dokumentu na obrazovce monitoru nebo tiskárně a interpretuje uživatelský vstup jako operace v dokumentu.
- *MainView* - je třída, kde probíhá hlavní proces renderování modelu, je tedy stežejní pro celou aplikaci. V konstruktoru se inicializují důležité objekty a členské proměnné, např. instance *client_dc* třídy *CClientDC*, která definuje atributy grafických elementů. Dále zde figurují proměnné *mouse_cross_x* a *mouse_cross_y* pro ovládání modelu pomocí

myši, proměnná pro indikaci posuvu modelu *moving*, zamčení pohledu *lock_aimpt*, nastavení pozadí scény *bk_color*, zobrazení drátěného modelu *always_wireframe*, inicializaci proměnných *near_plane* (blízká ořezávací rovina), *far_plane* (vzdálená ořezávací rovina), *heading* (směr), *pitch* (úhel) a *zoom* (perspektiva). Nastavení pozice modelu v prostoru se realizuje přes instanci *viewpt* třídy *Vector3*, kde se zapisuje do členských proměnných *x*, *y* a *z*. Framework v prostředí MFC volá metodu *OnDraw*, která renderuje výsledný dokument, jako argument zde vstupuje ukazatel na instanci třídy *CDC* (Class Device Context). Inicializace OpenGL probíhá v metodě *Init* a ve vnořené *InitGL*, aktualizaci dané scény vykonává *OnUpdate*, kde jsou obsaženy metody jako je *update_gl* (aktualizace polohy modelu), *DrawScene* (renderování výsledného modelu), *gdi_stuff* (překreslení textu ve scéně) a další. Metoda *draw_scale* vytváří teplotní stupnici, *look_at* nastavuje model do dané pozice v prostoru a *update_aimpt* aktualizuje polohu modelu. Nastavení polohy a rotace modelu je implementováno do již popsané metody *DrawScene*, použitím transformační matice typu *GL_MODELVIEW* a pro modifikaci perspektivního promítání je aplikována matice *GL_PROJECTION*. Třída také zpracovává příkazy, přicházející z uživatelského rozhraní, a to v metodách jako je např. *OnRButtonDown* nebo *OnRButtonUp*.

- *ColorScale* - tato třída konstruuje teplotní stupnice, která může být volitelně zobrazena ve scéně. Barevnou paletu vytváří členská metoda *create_cell*, jednotlivé barvy jsou zde namíchány ze složek RGB.
- *Vector3* - instance této třídy obsahuje informace o poloze jednoho bodu vektoru, jenž např. reprezentuje velikost a směr proudící skloviny. Členské proměnné *x*, *y*, *z* udržují aktuální souřadnici v každém směru. Tyto proměnné jsou především aplikovány ve funkcích OpenGL, v podobě vstupních argumentů, příkladem je často vyskytující se funkce *glVertex3f*, která specifikuje vrchol v prostoru.

6.2.2 Konstrukce dialogových oken

Cílem této podkapitoly je popis jednotlivých dialogových oken, jejichž základem jsou třídy odvozené od báze třídy *CDialog*. Obrázek 27 popisuje rozvržení všech tříd z hlediska dědičnosti a kompozice. Některé objekty mají báze třídy, které již byly rozebrány v předchozí podkapitole, proto o nich zde nebude zmínka.



Obrázek 27 Diagram implementovaných dialogových oken

6.2.2.1 Popis implementovaných tříd

- *CDialog* - je bázevá třída, sloužící ke konstrukci a zobrazení dialogových oken na obrazovce. Třída definuje dva typy oken, modální (modal) a nemoďální (modeless). V této aplikaci jsou použity jen modální okna, to znamená, že překrývají okno hlavního dokumentu (blokují přístup), a dokud není okno uzavřeno, není k dalším oknům aplikace přístup, včetně hlavního dokumentu.
- *ComponentDialog* - je odvozená třída, která ve svém konstruktoru implementuje vzhled dialogového okna pod identifikátorem `IDD_COMPONENTS`. Cílem instance

této třídy je akumulovat vytvořené a zobrazené komponenty modelu na scéně, např. geometrie modelu nebo teplotní řezy v dané rovině. Dialog dovoluje nastavit vlastnosti zvolené komponenty, definovat její obarvení nebo odstanění ze seznamu.

- *AddComponentDialog* - je třída, která v konstruktoru implementuje identifikátor vzhledu `IDD_COMPTYPES`. Dialog umožňuje vybrat požadovanou komponentu ze seznamu, případně i opakovaně. Komponenta je po potvrzení v podobě instance zařazena do seznamu v objektu třídy *ComponentDialog*.
- *FieldPropertiesDialog* - implementuje `IDD_FIELDS`, cílem instance této třídy je podat uživateli informace o aktuálně zvoleném datovém souboru, např. o *gt.dir* (pole teplot). Dále dialog dovoluje nastavit meze jednotlivých stupnic a jejich zobrazení na scéně.
- *LightDialog* - implementuje `IDD_LIGHTPROPS`, tato třída zajišťuje pro uživatele rozhraní z hlediska nastavení osvětlení scény. Aplikace dovoluje vytvořit až tři instance této třídy, tedy tři zdroje osvětlení. První (primární) zdroj má pevnou polohu a je definován jako *head-up light* a zbylé dva lze libovolně nastavit, nazývají se *fixed lights*.
- *ProgressDialog* - implementuje `CG_IDD_PROGRESS`, instance této třídy vznikají v instancích třídy *Field*, v metodách *read_2dxy*, *read* a *update_stats*. Informační dialog je zobrazen v okamžiku, kdy se datové soubory z disku načítají delší dobu. Instance monitoruje běžící proces čtení a interpretuje ho v podobě procentuálního vyjádření.
- *StringDialog* - implementuje `IDD_STRINGDIALOG`, instance této třídy dovoluje definovat a zobrazit textový popis ve scéně v levém horním rohu.
- *AboutDialog* - implementuje `IDD_ABOUTBOX`, instance třídy zobrazuje informace o programu, autorovi a verzi.

6.2.3 Konstrukce geometrie pece

Tato podkapitola se věnuje problematice geometrie pece z hlediska struktury datových souborů, implementace tříd a funkcí z knihovny OpenGL.

6.2.3.1 Datové soubory

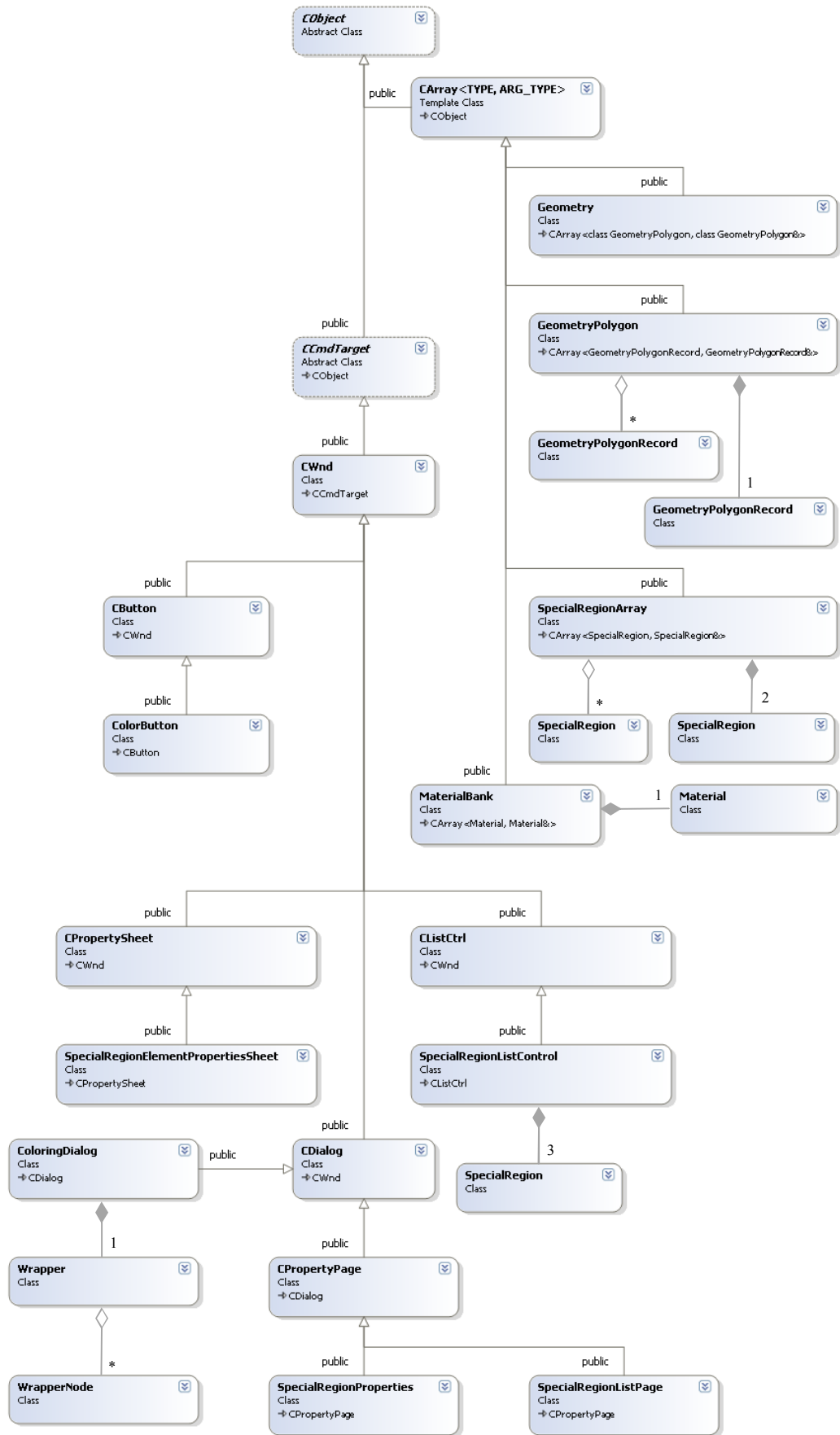
Geometrie modelu pece je rozdělena do několika datových souborů, které generuje specifická CAD aplikace, určena pro vytvoření grid systému a návrh speciálních oblastí.

Viewer3D požaduje pro zobrazení geometrie pece následující datové soubory:

- *ggri.d* - základní informace o modelu, soubor je načten uživatelem v dialogovém okně aplikace Viewer3D, v *ggri.d* jsou i reference na další důležité datové soubory, potřebné pro renderování geometrie, soubor dále pro každý rozměr (XYZ) obsahuje difference DX(I), DY(J) a DZ(K), které jsou mezi jednotlivými uzly grid systému (grid spacings),
- *gdbm.d* - databanka materiálů, obsahuje druhy materiálů, typy teplotních funkcí, koeficienty, teplotní minima a maxima, referenční teploty a materiálové vlastnosti (tepelná vodivost, elektrická vodivost, specifické teplo, zářivost, hustota, atd.),
- *ggeo.d* - soubor, v které se nachází geometrie modelu pece, je zde zaznamenán počet polygonů, každý definovaný polygon se skládá z počtu vrcholů, barvy, identifikačního čísla polygonu a materiálu,
- *grsp.d* - datový soubor obsahující parametry všech speciálních oblastí, jako je typ oblastí, sériová čísla materiálů v databance, geometrické druhy oblastí, rozsah oblastí (pro pravoúhlé oblasti) a další,
- *gspm.dir* - datový soubor obsahující identifikace speciálních oblastí v pozicích uzlů.

6.2.3.2 Popis implementovaných tříd

Tato část je zaměřena na rozbor tříd, které jsou zodpovědné za sestavení a zobrazení geometrie modelu na vymezené místo v okně aplikace. Když je *Viewer3D* spuštěn, je nejprve požadováno načtení hlavičkového datového souboru *ggri.d* nebo scény ze souboru s příponou *V3D*. Pokud je načten *ggri.d*, je geometrie modelu automaticky zobrazen do výchozí bokorysné polohy. Konstrukci modelu pece a vykreslení provede instance třídy *SpecialRegionElement*. Model pece je složen z polygonů, které mají zpravidla osm vrcholů, jsou načteny v daných velikostech ze souboru *ggeo.d*. Nejprve dochází k vykreslení drátěného modelu (wireframe) a poté z jednotlivých polygonů v metodě *draw_subdiv_quad*, kde se zobrazují za pomoci funkcí *glVertex3f* v režimu útvaru *GL_QUAD_STRIP* (grafické primitivum, je podobné pásu trojúhelníků). V případě načtení souboru *V3D*, nemusí být geometrie modelu zobrazena, scéna tedy může např. obsahovat jen teplotní nebo vektorový profil. Obrázek 28 názorně popisuje hierarchii tříd, z hlediska načtení dat, nastavení parametrů a výsledného renderování modelu pece na scéně.



Obrázek 28 Diagram tříd geometrie modelu pece

- *SpecialRegion* - je třída popisující jednu speciální oblast, která je charakterizována např. typem oblasti (EMPTY, ELECTRODE, WALL, BATCH, STIRRER, atd.), typem použitého materiálu a identifikačním číslem.
- *SpecialRegionArray* - je třída odvozena od *CArray*, její význam je udržovat seznam všech speciálních oblastí, využití je především v instanci třídy *Part*.
- *Material* - tato třída definuje strukturu materiálu, jenž je použit při renderování polygonů. Instance této třídy zahrnuje členské proměnné, jako je typ a jméno materiálu, jeho barva a další. V metodě *read* se načte materiál z daného souboru.
- *MaterialBank* - v instanci této třídy se ukládají všechny definované materiály do pole typu *CArray*, kde jsou k dispozici pro instanci třídy *Part*. Seznam materiálů je postupně získán v metodě *read* z instance třídy *Material*.
- *GeometryPolygonRecord* - instance této třídy uchovává základní informace o jednom vrcholu daného polygonu, např. jeho souřadnice v prostoru.
- *GeometryPolygon* - tato třída popisuje jeden polygon, jenž je základním prvkem při konstrukci modelu pece, má zpravidla osm vrcholů. V instanci této třídy se udržují informace o objektu, jako je číslo použitého materiálu, číslo a barva polygonu, atd.
- *Geometry* - instance této třídy udržuje seznam všech polygonů v poli typu *CArray*, z které je tato třída odvozena. Polygony jsou pak v objektu typu *SpecialRegionElement* zobrazeny na scéně. Ve starší verzi datového modelu se zobrazovaly jen speciální oblasti, tedy pravoúhlá tělesa typu kvádr nebo krychle. Nový datový model dovoluje definovat i tělesa, které mohou mít šikmé stěny, proto se na scéně prezentuje geometrie modelu jen v podobě těchto polygonů, více to odpovídá realitě.
- *CArray* - je šablona kolekce (collection template), která dovoluje uložit jakýkoliv datový typ do pole. Kolekce sama roste nebo se zmenšuje na základě potřeby uživatele.
- *CListCtrl* - básová třída, která zapouduje funkčnost z vzhled ovládacího prvku typu seznam (list view control), tuto třídu dědí objekt typu *SpecialRegionListControl*.
- *CPropertyPage* - básová třída reprezentující individuální stránky typu záložka vlastností, známé jako dialogové okno se záložkami (tab dialog box). Tuto třídu dědí

objekty typu *SpecialRegionProperties* a *SpecialRegionListPage*, jenž definují vlastnosti geometrie modelu pece z hlediska zobrazení na scéně.

- *CPropertySheet* - bazová třída, která reprezentuje list uživatelských ovládacích prvků, taky je často nazýván jako dialogové okno se záložkami. Třidu dědí objekt typu *SpecialRegionElementPropertiesSheet*.
- *SpecialRegionProperties* - třída poskytující aktivaci/deaktivaci zobrazení obrysových čar, slouží pro zvýraznění jednotlivých částí pece (Outline faces), a nastavení výplně jednotlivých polygonů (Fill faces).
- *SpecialRegionListPage* - třída definující ovládací prvky pro nastavení jednotlivých oblastí (polygonů), jedná se o skrytí/zobrazení oblastí v generovaném seznamu.
- *SpecialRegionElementPropertiesSheet* - je třída definující jednotlivé záložky v okně obsahuje instanci třídy *SpecialRegionProperties* a *SpecialRegionListPage*, které jsou v konstruktoru přidány do seznamu záložek pomocí metody *AddPage*.
- *SpecialRegionListControl* - je třída odvozená od bazové *CListCtrl*, cílem instance této třídy je udržovat seznam všech částí (polygonů) modelu, včetně názvu oblastí, informací o polohách v prostoru, viditelnost a další. Tento objekt je využit ve třídě *SpecialRegionListPage*, jenž definuje okno s ovládacími prvky.
- *ColoringDialog* - je třída odvozená od *CDialog*, její význam je v implementaci barvy nebo pole barev do dané komponenty, např. obarvení jednotlivých stěn polygonů na základě materiálové banky, kde ve třídě *Material*, členská proměnná *color* udává barvu žádaného materiálu, nebo obarvení stěn dle daných teplot v místech nejbližších uzlů v diskretní síti (grid system). Dále dialog dovoluje nastavení smíchání uživatelské s původní barvou materiálu formou procentuálního vyjádření, jenž má na starosti funkce *glColor4fv*. Poslední volbou je aktivace/deaktivace reflexního osvětlení a jeho intenzita, je vyjádřena číslem, implicitně hodnotou 100. Tento vizuální efekt je v kompetenci funkce *glMaterialf*, při použití argumentů *GL_FRONT_AND_BACK* a *GL_SHININESS*.
- *CButton* - je bazová třída poskytující funkčnost a vzhled systémového tlačítka (Windows button control). Tato třída je zděděna v objektu typu *ColorButton*.
- *ColorButton* - v této odvozené třídě je vytvořen vzhled tlačítka, jenž má místo textu v poli *caption*, definován obdelník s výplní, indikující aktuální barvu. Instance této třídy

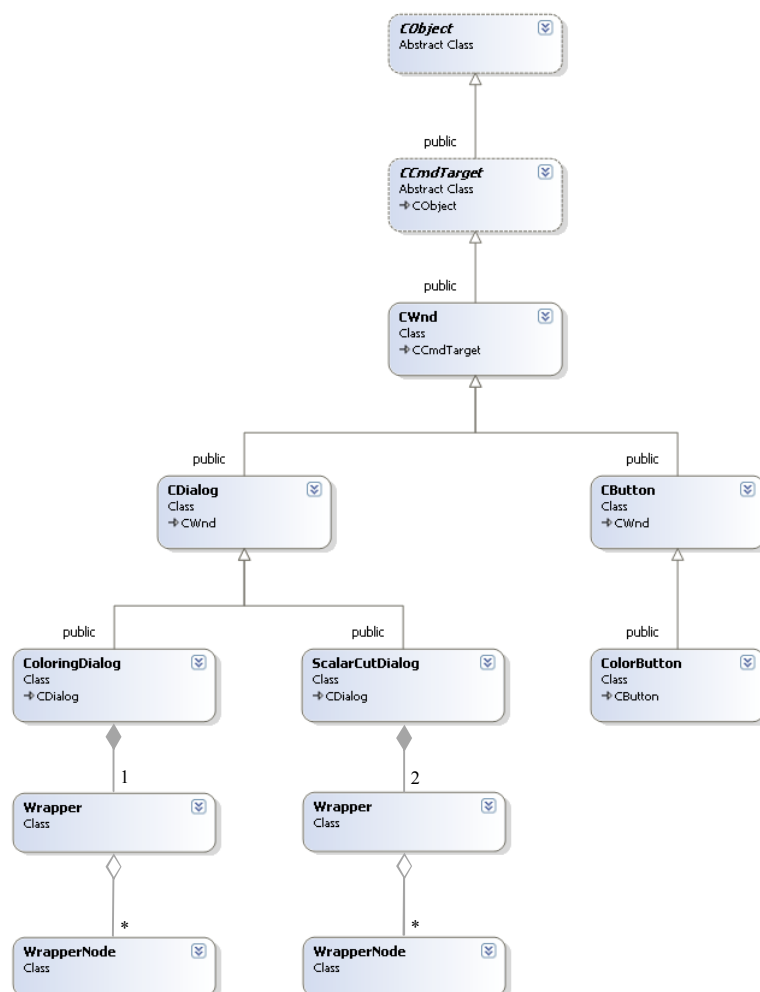
je použita v objektu typu *ColoringDialog*, kde jejím prostřednictvím je nastavena jednotná barva všech polygonů, a v objektu *LightDialog*, kde uživatel vybírá barvu jednoho ze tří zdrojů osvětlení geometrie modelu pece.

6.2.4 Konstrukce komponent modelu

Program dovoluje vygenerovat a zobrazit dva typy komponent, které mohou být vytvořeny v libovolném počtu instancí s různými hodnotami vstupních parametrů.

6.2.4.1 Komponenta skalárních a vektorových řezů

V této podkapitole je popsána komponenta skalárních a vektorových řezů (Scalar and vector slices). Komponenta je zastoupená třídou *ScalarCutDialog*. Obrázek 29 podrobně popisuje strukturu tříd, jenž jsou potřebné pro nastavení parametrů řezných rovin.



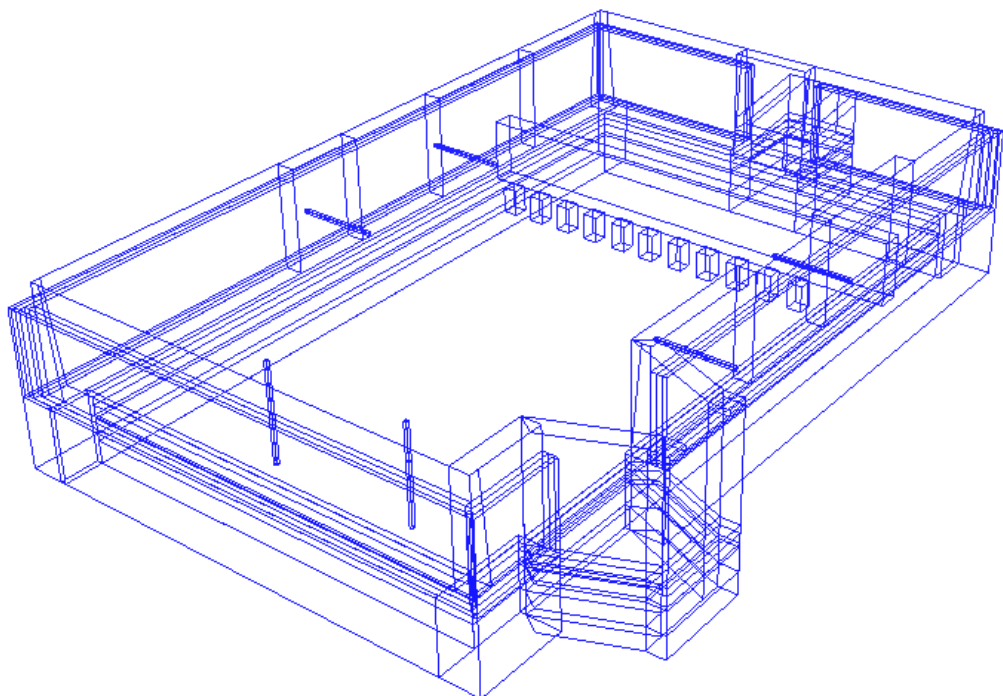
Obrázek 29 Diagram tříd pro parametrizaci řezných rovin

Nejsou zde popsány třídy, jako je *Wrapper*, *WrapperNode*, *ColoringDialog*, *ColorButton*, které jsou již zmíněny v předchozích kapitolách, mají totiž stejnou funkci.

- *ScalarCutDialog* - je třída odvozená od báze *CDialog*, má na starosti veškeré parametry z hlediska zobrazení komponenty, jako je např. volba typu řezné roviny, aktivace/deaktivace skalárního/vektorového řezu, případně více řezů, nastavení počtů řezů a difference mezi nimi. Tato třída úzce souvisí s třídou *CutElement*, ve které dochází k samotnému renderování komponenty na scénu.

6.2.4.2 Komponenta geometrie modelu

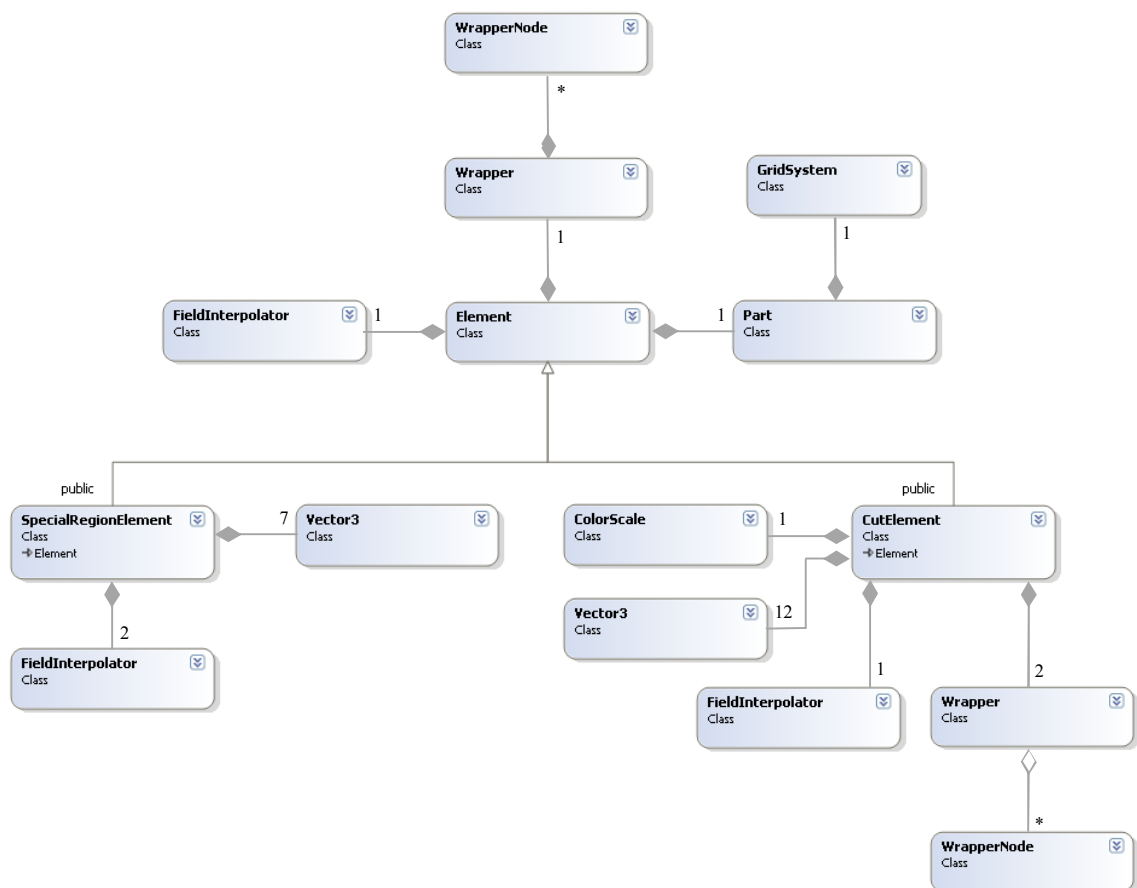
Tato komponenta (Mode geometry), generující a renderující soustavu polygonů, již byla dostatečně popsána v kapitole 6.2.3, proto jen krátce z hlediska popisu. Komponenta je zastoupena objektem typu *SpecialRegionElementPropertiesSheet*, která dovoluje uživateli parametrizovat vzhled geometrie modelu. S tímto objektem je úzce spojena třída typu *SpecialRegionElement*, její činností je renderování geometrie. Obrázek 30 prezentuje geometrii v podobě drátěného modelu, polygony tedy mají jen obrysové čáry.



Obrázek 30 Ukázka geometrie v podobě drátěného modelu

6.2.4.3 Renderování komponent

V této části bude pohled zaměřen na popis obou komponent z hlediska jejich renderování. Hlavními aktéry této činnosti jsou objekty typu *SpecialRegionElement* a *CutElement*. V podkapitole 6.2.3.2 bylo vysvětleno, odkud se čtou data a jak se zobrazuje geometrie modelu. V této části budou k tomuto tématu dodány další informace z hlediska přípravy dat a samotného renderování, jak geometrie, tak řezných rovin. Obrázek 31 nabízí popis hierarchické struktury tříd, které jsou zapojeny do procesu renderování komponent.



Obrázek 31 Diagram tříd pro renderování jednotlivých komponent

Opět zde nebudou rozebrány třídy zmíněné v předchozích kapitolách, jedná se např. o *Wrapper*, *WrapperNode*, *GridSystem*, *ColorScale*, *Vector3* nebo *Part*.

- *Element* - je bázevá třída, z které jsou odvozeny *SpecialRegionElement* a *CutElement*. V konstruktoru této třídy se inicializují proměnné, jako je např. *shininess* (reflexní osvětlení) nebo instance třídy *Part* (komponenta), anebo *Wrapper*, jejíž instance *own_color_blend* nese informaci o poměru původní a uživatelské barvy.

- *FieldInterpolator* - je pomocná třída, jejím významem je interpolace hodnot skalárního a vektorového pole. V případě, že uživatel nastaví řeznou rovinu, která neprochází přímo uzly diskrétního systému, jsou výsledné hodnoty interpolovány ze sousedních uzlů. Pokud je tedy nastavena rovina XY a řez neprochází uzly, pak je hodnota skaláru v daném místě interpolována ze sousedního uzlu nad o pod řezem, výpočet realizuje metoda *interpolate_scalar*. V případě vektorového pole, jenž má tři složky (U,V,W), udávající směr, je princip stejný, výpočet zajišťuje metoda *interpolate_velocity*. Pokud je nastavena volba zobrazení teplotního pole v polygonech geometrického modelu, pak dochází také k výpočtu interpolovaných hodnot, protože polygony mohou mít různé tvary a často plochy neprocházejí uzly diskrétní sítě modelu.
- *SpecialRegionElement* - již v úvodu podkapitoly 6.2.3.2 byl zmíněn význam instance této třídy, jenž generuje a renderuje geometrii modelu na obrazovce. Třída je odvozena od bazového objektu typu *Element*. Hlavní metoda již byla popsána v úvodu, další důležitou částí, jenž definuje tato třída skrze metodu *properties*, je zobrazení a nastavení vlastností jednotlivých oblastí (polygonů) na scéně. Jelikož je model osvětlen, je nutné vypočítat i normály ve vrcholech povrchu polygonů. Tyto informace jsou poté využity pro výpočet difúzní světelné složky a odlesky. Výpočet se realizuje v bazové třídě *Element*, v metodě *calc_normal* před vlastním vykreslováním.
- *CutElement* - tato třída je také odvozena od bazové třídy *Element*, sdílí tedy stejné metody jako *SpecialRegionElement*, např. výpočet normál pro osvětlení. Třída definuje několik důležitých metod, jednou z nich je *field_quad*, jenž vymezuje řeznou rovinu v daném směru, do které se poté mapuje teplotní pole. Další metodou je *vector*, kde dochází k realizaci výpočtu jednotlivých vektorů, reprezentující rychlost a směr proudění skloviny. Počátečním bodem vektoru je interpolovaný uzel z diskrétní sítě a koncovým je interpolovaný prostorový bod získaný ze složek U, V a W (velikosti ve směrech X, Y a Z). Metoda *draw_vec_head* vypočítá velikost šipky na koncové straně vektoru. V metodě *properties* je vytvořena instance třídy *ScalarCutDialog*, odkud uživatel parametrizuje řeznou rovinu. Informace o aktuálních parametrech zobrazené komponenty, jenž je uložena v seznamu objektu typu *ComponentDialog*, definuje metoda *update_name*. V záznamu této komponenty se nachází informace o počtu zobrazených řezů, typu požadované roviny a počáteční pozice řezu.

6.2.5 Konstrukce obecných šablon

Tato podkapitola je doplňková, popisuje šablony, které jsou použity na některých místech programu. Z významu šablony je jasné, že cílem je znovupoužitelnost kódu, jenž je napsán obecně, to značně ulehčuje implementaci některých algoritmů.

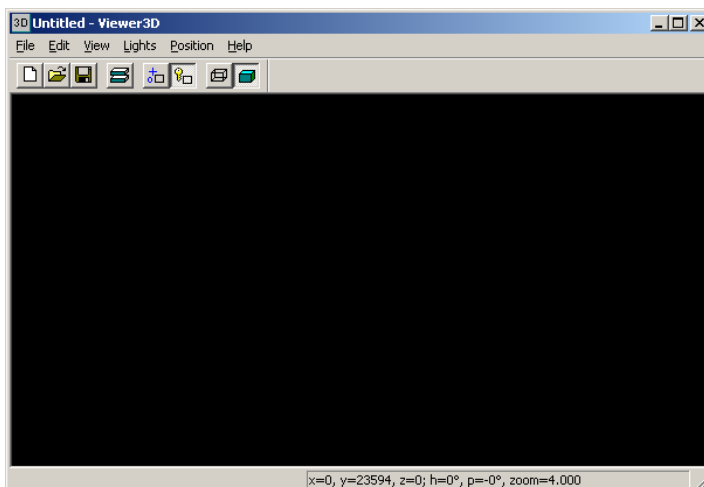
- `template <class T> class VersatileListNode` - je jednoduchý propojený seznam uzlů (obsahující data), kde je deklarován ukazatel na předchozí a následující uzel.
- `template <class T> class VersatileList` - je složitější propojený seznam, dovolující pomocí deklarovaných metod jednotlivé prvky (datové uzly) vkládat, mazat, nastavovat aktuální pozici, zjistit velikost seznamu a další užitečné funkce.

6.3 Popis programu z pohledu uživatele

Program *Viewer3D* je pokročilá vizualizace, je to rozhraní, které dovoluje zobrazit datovou interpretaci výsledků z matematického modelování. Program nabízí uživateli pestrou paletu funkcí, které lze aplikovat na zobrazený model pece. Následující výklad bude zaměřen na popis prostředí programu, dialogových oken a manipulaci s modelem pece na scéně.

6.3.1 Základní rozhraní programu

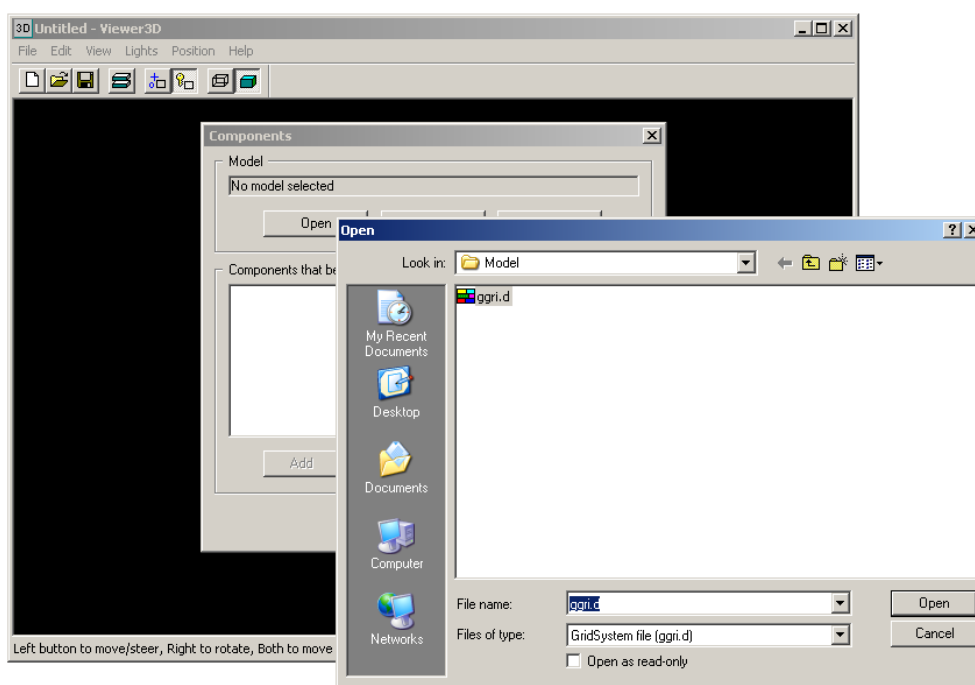
Vizualizační program nabízí uživatelské rozhraní v podobě aplikačního okna, které obsahuje menu s funkcemi, panel nástrojů (toolbar), informační panel (status bar) a scénu s výchozí černou barvou na pozadí. Zobrazovací plochou je scéna, hlavní činnost tedy probíhá v této vymezené oblasti. Obrázek 32 prezentuje základní okno popisovaného programu.



Obrázek 32 Základní okno vizualizačního programu

6.3.2 Zobrazení modelu pece

Data modelované pece jsou uloženy v několika souborech, podle jejich významu, popisu se věnuje podkapitola 6.2.3.1, proto t tomuto tématu jen krátce. Hlavním datovým souborem je *ggri.d*, který je požadován pro načtení celé struktury. Obrázek 33 popisuje způsob načtení tohoto souboru. Postup je následující, v menu kliknou na *Edit* → *Components*, zobrazí se dialogové okno *Components*, poté kliknout na tlačítko *Open*, otevře se okno s adresářovou strukturou, ve kterém se označí požadovaný *ggri.d* a na závěr potvrdit tlačítkem *Open*. Soubor bude načten a na scénu bude zobrazena geometrie modelu pece.

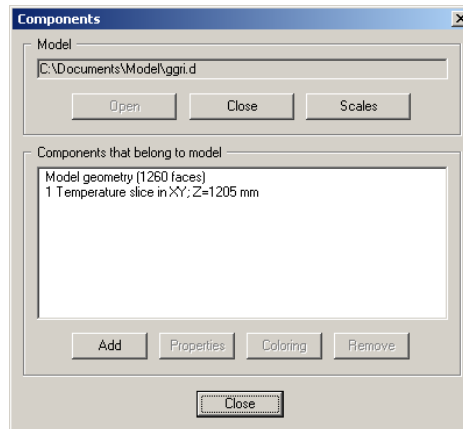


Obrázek 33 Ukázka otevření datového souboru v programu

Po načtení tohoto datového souboru, je do seznamu v okně *Components* přidán záznam s názvem *Model geometry (n faces)*, hodnota *n* indikuje počet ploch, které jsou na scéně zobrazeny. Pokud má jeden polygon 8 vrcholů, pak má 6 vnějších a 6 vnitřních ploch, celkem tedy 12 zobrazených ploch (faces).

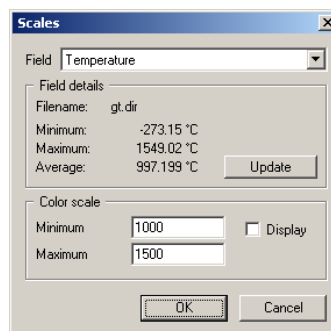
6.3.2.1 Dialogové okno *Components*

Dialogové okno *Components* nabízí uživateli mnoho užitečných funkcí. Z pohledu editace modelu, veškeré úpravy komponent probíhají zde v okně, obrázek 34 popisuje všechny možnosti, jako je přidání, smazání a modifikace komponenty, nastavení stupnice a další.



Obrázek 34 Ukázka dialogového okna Components

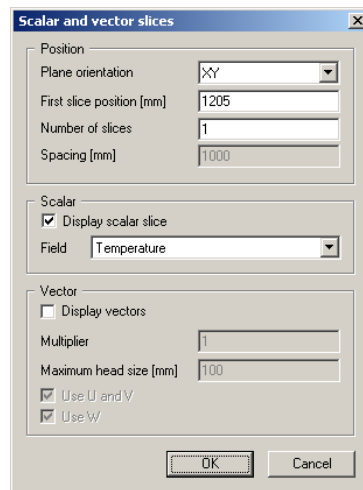
- *Model* - indikuje aktuální datový model na scéně této aplikace. Na tomto místě je pro lepší přehled zobrazena plná cesta k základnímu datovému souboru *ggri.d*.
- *Open* - zobrazí vstupní dialogové okno pro výběr požadovaného datového souboru.
- *Close* - odstraní aktuální model z aplikace, která poté přejde do výchozího nastavení.
- *Scales* - zobrazí dialogové okno *Scales* s informacemi o datových polích typu *gt.dir*, *gu.dir*, *gv.dir* a *gw.dir*. Dále se v okně nachází aktivace/deaktivace stupnic, včetně nastavení jejich dolních a horních mezí. Na scéně je možné zobrazit až tři stupnice, tedy podle výběru pole v položce *Field* a nastavení *Display*. Obrázek 35 ukazuje rozvržení jednotlivých informací a ovládacích prvků v tomto dialogovém okně.



Obrázek 35 Ukázka dialogového okna Scales

- *Components that belong to model* - udržuje aktuální seznam zobrazených komponent.

- *Add* - zobrazí dialogové okno *Add component*, kde se nachází nabídka dostupných komponent pro zobrazení na scéně. V seznamu *Available components* jsou uvedeny dva typy komponent, a to *Scalar and vector slices* (skalární a vektorové řezy) a *Model geometry* (geometrie modelu pece). Aplikace dovoluje vytvořit libovolné množství instancí těchto komponent, to z důvodu, aby bylo možné použít komponenty s různými parametry, např. rozdílné typy řezných rovin v jedné scéně.
- *Properties* - zobrazí dialogové okno vlastností komponenty, jenž je vybrána v seznamu *Components that belong to model*. Pokud je tedy zvolena komponenta typu *Scalar and vector slices*, pak je zobrazen stejnojmenný dialog. Obrázek 36 podrobně popisuje toto dialogové okno se všemi možnostmi nastavení řezných rovin.

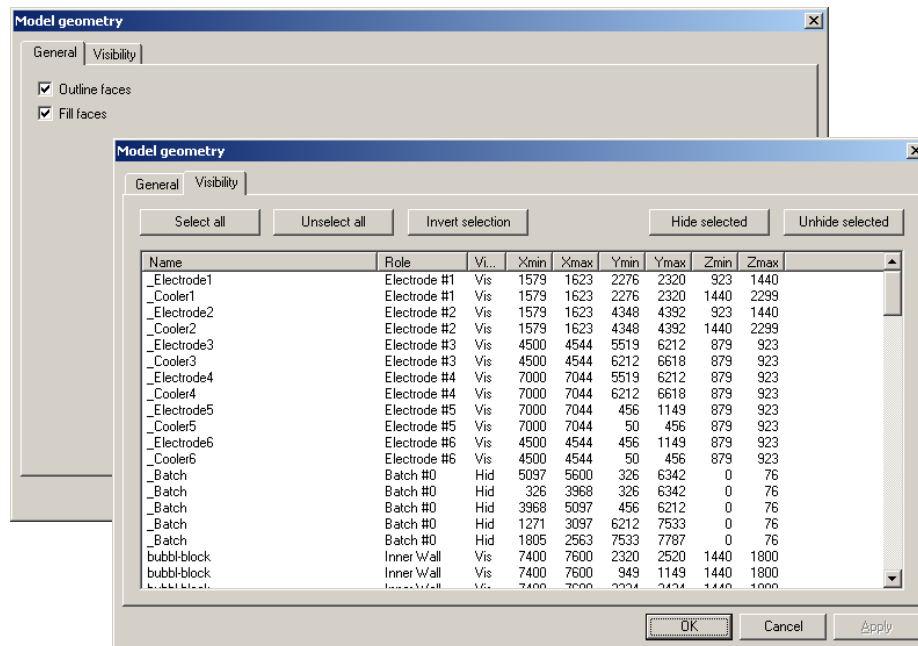


Obrázek 36 Ukázka dialogového okna *Scalar and vector slices*

- *Plane orientation* - nastaví typ řezné roviny (XY, XZ, YZ).
- *First slice position* - nastaví pozici prvního řezu, hodnota je uvedena v milimetrech.
- *Number of slices* - definuje počet řezů ve zvolené řezné rovině.
- *Spacing* - definuje konstantní diferenci mezi jednotlivými řezy.
- *Display scalar slice* - aktivuje/deaktivuje skalární řeznou rovinu na scéně.
- *Field* - definuje typ skalárního pole, výchozím profilem je teplota, volitelně aplikace dovoluje zobrazit i jednotlivé pole rychlostních složek U, V a W.

- *Display vector* - aktivuje/deaktivuje vektorovou řeznou rovinu na scéně.
- *Multiplier* - nastavuje násobitel vektorů, jenž pomáhá v lepší orientaci.
- *Maximum head size* - nastavuje velikost hlavičky (šipky) vektoru.
- *Use U and V* - aktivuje/deaktivuje složku U a V v řezné rovině. Pokud jsou složky deaktivovány, pak vektory nejsou vyjádřeny v osách X a Y.
- *Use W* - aktivuje/deaktivuje složku W, tedy velikost vektoru ve směru osy Z.

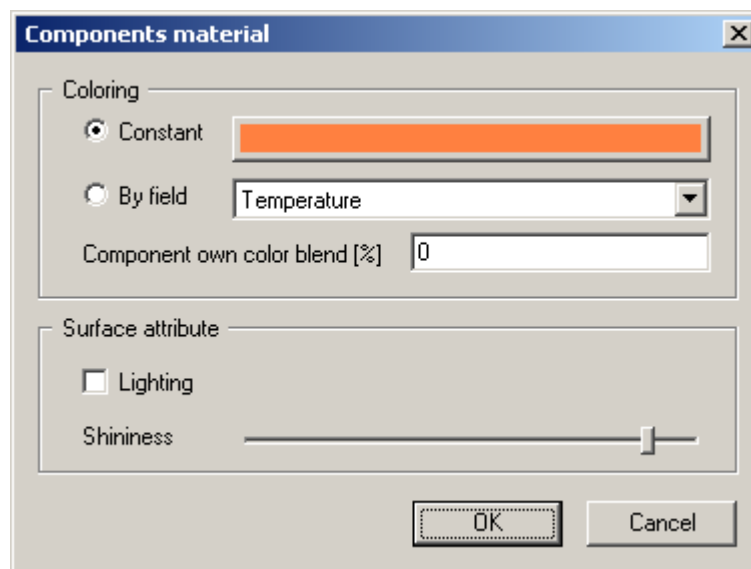
Pokud je ovšem zvolena komponenta typu *Model geometry*, pak je zobrazen stejnojmenný dialog se záložkami, se jmény *General* a *Visibility*. Obrázek 37 reprezentuje popisované dialogové okno, které nabízí uživateli mnoho užitečných funkcí.



Obrázek 37 Ukázka dialogového okna Model geometry

- *Outline faces* - aktivuje/deaktivuje v modelu obrysové čáry jednotlivých polygonů.
- *Fill faces* - aktivuje/deaktivuje výplň polygonů, v případě deaktivace těchto ploch, je na scéně zobrazen jen drátěný model pece.
- *Select all* - vybere všechny oblasti (polygony) v níže umístěném seznamu.
- *Unselect All* - zruší výběr všech oblastí v uvedeném seznamu.

- *Invert selection* - provede v seznamu inverzní výběr oblastí, to znamená, že oblasti, které nejsou vybrány, tak budou označeny a naopak.
- *Hide selected* - skryje označené oblasti, ve sloupci *Visibility* se objeví stav *Hid*.
- *Unhide selected* - zobrazí označené oblasti, ve sloupci *Visibility* se objeví stav *Vis*.
- *Coloring* - zobrazí dialogové okno *Components material*, které definuje RGB barvu, případně paletu barev (na základě hodnot skalárního pole) pro zvolenou komponentu v seznamu. Obrázek 38 popisuje výchozí dialogové okno pro správu barev a osvětlení.



Obrázek 38 Ukázka dialogového okna
Components material

- *Constant* - obarví zvolenou komponentu konstantní barvou. Barevné tlačítko v případě stisku zobrazí paletu, v které lze nastavit libovolnou barvu. Tato volba je platná jen pro obarvení geometrie modelu pece a vektorového řezu.
- *By field* - obarví komponentu podle zvoleného skalárního pole, výchozí položkou je teplota, ale volitelně lze nastavit i složky vektorového pole. Nastavení je opět platné jen pro geometrii a vektorové řezy. Obarvení dané komponenty je založeno na diskrétní síti, na jednotlivé uzly jsou mapovány skalární hodnoty (např. teplota) a interpolované hodnoty jsou poté dosazeny do stěn dané komponenty.
- *Lighting* - aktivace/deaktivace osvětlení dané komponenty.
- *Shininess* - nastavení intenzity odrazu světla.

- *Remove* - odstraní vybranou komponentu ze seznamu *Components that belong to model*. Výsledkem tohoto zásahu je smazání komponenty ze scény.

6.3.2.2 Dialogové okno *Model Caption*

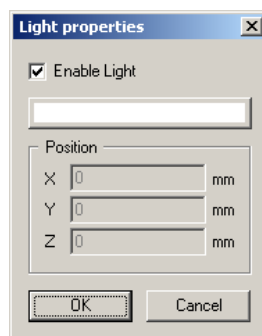
Dialogové okno *Model Caption* se nachází v menu *Edit* → *Model Caption*, dovoluje uživateli zobrazit libovolný text ve scéně v levém horním rohu obrazovky.

6.3.2.3 Dialogové okno *Background Color*

Cílem dialogového okna *Background Color* je zobrazení palety barev, výsledná hodnota barvy je pak použita pro vykreslení pozadí celé scény. V případě požadavku lze vytvořit i vlastní barvu. Dialog se nachází v menu *Edit* → *Background color*.

6.3.2.4 Dialogové okno *Head-up Light*

Výsledkem použití dialogu *Head-up Light* je nastavení osvětlení scény ve výši očí. Polohu světelného zdroje nelze měnit, pouze barvu z palety, která se nachází pod barevným tlačítkem. Nastavení tohoto druhu osvětlení (např. aktivace/deaktivace) se provádí v menu *Edit* → *Head-up Light*. Obrázek 39 reprezentuje typ popisovaného dialogového okna.



Obrázek 39 Ukázka dialogového okna *Head-up Light*

6.3.2.5 Dialogové okno *Fixed Light*

Dialogové okno *Fixed Light* dovoluje nastavit světelný zdroj do libovolného místa v prostoru. Dále také nabízí nastavení barvy osvětlení a případnou aktivaci/deaktivaci zdroje. Dialog je shodný s *Head-up Light* z předchozí části 6.3.2.4, jen navíc dovoluje nastavení

pozice. Dialog se nachází v *Edit* → *Fixed Light 1* a *Fixed Light 2*, z toho plyne, že uživatel může nastavit dva nezávislé světelné zdroje, např. s různou barvou osvětlení.

6.3.2.6 Dialogové okno *About*

Dialogové okno *About* poskytuje informace o aplikaci, nachází se v menu *Help* → *About*.

6.3.3 Rozhraní pro manipulaci se soubory

Vizualizační program nabízí v menu *File* standardní funkce pro manipulaci s datovými soubory typu *V3D*, jedná je o následující položky:

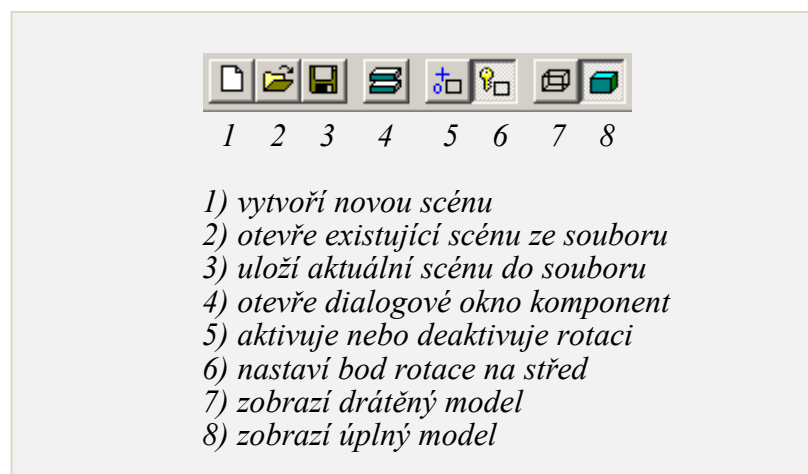
- *New* - vytvoří novou scénu.
- *Open* - otevře ze souboru vytvořenou scénu.
- *Save* - uloží vytvořenou scénu do aktuálního souboru na disk.
- *Save As* - uloží vytvořenou scénu do nově zadaného souboru na disk.

Menu *File* dále nabízí položku *Exit*, která ukončí běhu programu.

6.3.4 Rozhraní pro nastavení vzhledu programu

Vzhled vizualizačního program lze nastavit pomocí následujících funkcí:

- *Toolbar* - aktivuje/deaktivuje panel nástrojů, jenž je implicitně umístěn pod menu. Panel je možné přesunout myší kamkoliv na zobrazovací plochu monitoru. Na obrázku 40 je uveden podrobný popis všech funkcí, které zmíněný panel uživateli nabízí.



Obrázek 40 Ukázka funkcí v panelu nástrojů

- *Status Bar* - aktivuje/deaktivuje informační panel v zápatí okna programu, v kterém je umístěna kontextová nápověda. Dále jsou zde k dispozici aktuálně nastavené hodnoty o zobrazeném modelu na scéně, jako je poloha v prostoru, úhly natočení a perspektiva.

6.3.5 Rozhraní pro nastavení zobrazeného modelu

Program dovoluje model pece zobrazit ve dvou níže uvedených režimech:

- *Solid* - zobrazí model v plném režimu (vykreslí výplně všech ploch v modelu).
- *Wireframe* - zobrazí model v podobě drátěné konstrukce.

Rozhraní poskytuje uživateli i nastavení polohy modelu na scéně, jsou to funkce:

- *Set Centre* - nastaví bod rotace na střed geometrie modelu ve všech směrech prostoru.
- *Lock Aimpoint* - aktivuje/deaktivuje režim rotace, provede to tím způsobem, že ho zamkne. Pokud je režim deaktivován, pak je model na obrazovce jen posouván ve směrech x a y . Tímto pohybem je docíleno přesnějšího zobrazení požadovaného místa.

6.3.6 Ovládání zobrazeného modelu pece

Ovládání zobrazeného modelu na scéně je velmi snadné, celá tato činnost je realizována polohovacím zařízením, tedy počítačovou myší, v kombinaci s funkcí *Lock Aimpoint*, která je umístěna v aplikačním menu *Position* nebo případně v panelu nástrojů.

Kombinace zmíněné funkce a tlačítek myši jsou následující:

- Aktivní tlačítko *Lock Aimpoint*, současně stisknuté levé a pravé tlačítko myši, plus libovolný pohyb myši: dovolí režim rotace kolem daného bodu v prostoru.
- Aktivní tlačítko *Lock Aimpoint*, stisknuté levé tlačítko myši, plus libovolný pohyb myši: dovolí přibližování nebo vzdalování modelu po dráze mezi zamčeným bodem v prostoru a bodem pohledu ze strany uživatele.
- Neaktivní tlačítko *Lock Aimpoint*, současně stisknuté levé a pravé tlačítko myši, plus libovolný pohyb myši: dovolí uživateli režim plošného posuvu modelu na scéně.
- Neaktivní tlačítko *Lock Aimpoint*, stisknuté levé tlačítko myši, plus libovolný pohyb myši: dovolí přibližování nebo vzdalování modelu po dráze mezi neuzamčeným bodem (volně pohyblivým) v prostoru a bodem pohledu uživatele.

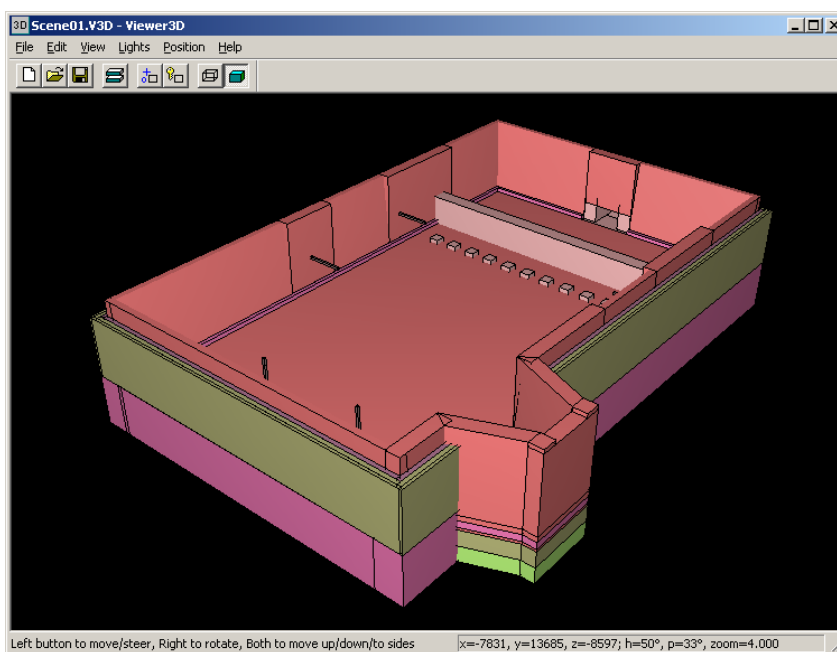
- Otáčení kolečkem myši: dovolí uživateli měnit perspektivní projekci zobrazeného modelu. Výchozí hodnota parametru projekce (zoom) je 4.000, tento údaj je pro lepší orientaci umístěn v informačním panelu. Pokud je kolečkem otáčeno směrem k uživateli, je hodnota snižována, v případě opačného točení, je hodnota zvyšována.

6.3.7 Datový soubor vytvořené scény

Vizualizační program dovoluje uživateli uložit výsledné nastavení jednotlivých komponent a polohu modelu pece na scéně do výstupního datového souboru. Je v binárním tvaru se specifickým formátem, s příponou *V3D* (Viewer 3 Dimensional). Obsah souboru začíná absolutní cestou k uloženým datům modelované pece, poté následuje cesta k souboru s názvem *ggri.d*, který je stěžejním pro samotné zobrazení. Dalšími hodnotami v pořadí jsou parametry pro vytvoření scény, nastavení geometrie a všech komponent řezných rovin. Hodnoty jsou postupně načteny z jednotlivých tříd do dialogových oken.

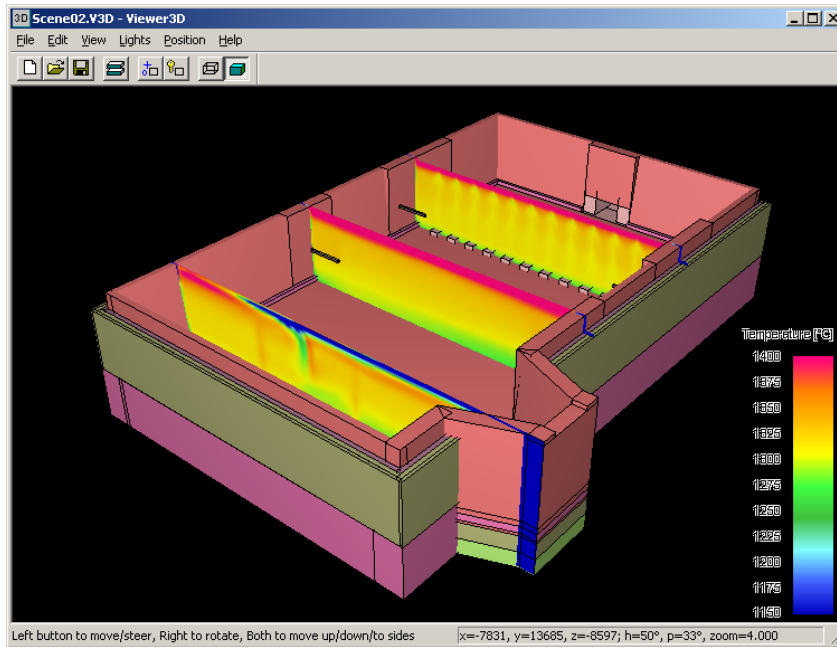
6.4 Výstupy z vizualizačního programu

V této podkapitole budou prezentovány grafické výstupy z programu. Na obrázku 41 je názorně vidět geometrie modelu pece, jedná se o tavící vanu, kde jednotlivé oblasti mají definovanou velikost, polohu a barvu, která odpovídá typu použitého materiálu z databáze.



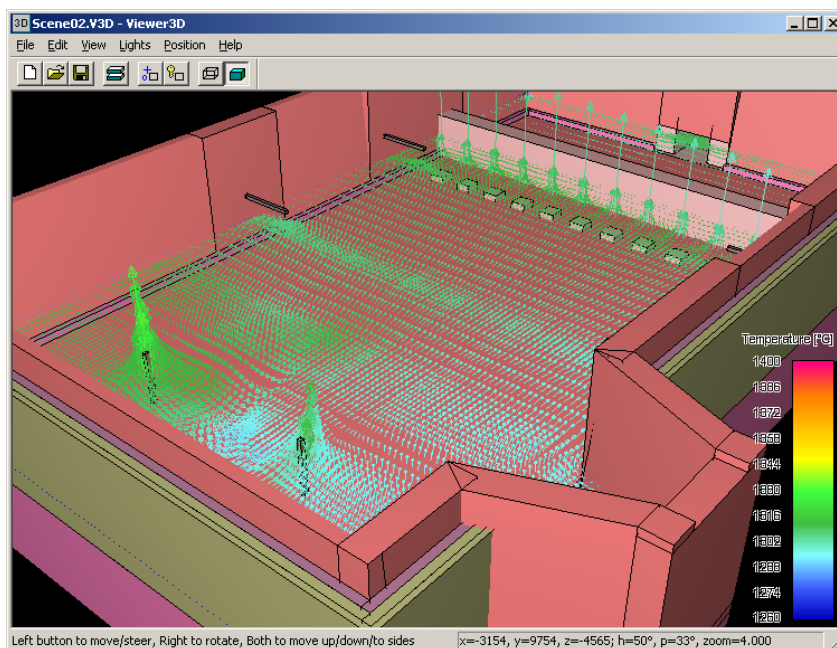
Obrázek 41 Ukázka zobrazení geometrie modelu pece

Obrázek 42 reprezentuje tři teplotní řezy v rovině YZ. První je nastaven v oblasti zakládání vsázky, druhý v místě tavení skloviny a třetí v pozici výstupních trysek probublávače.



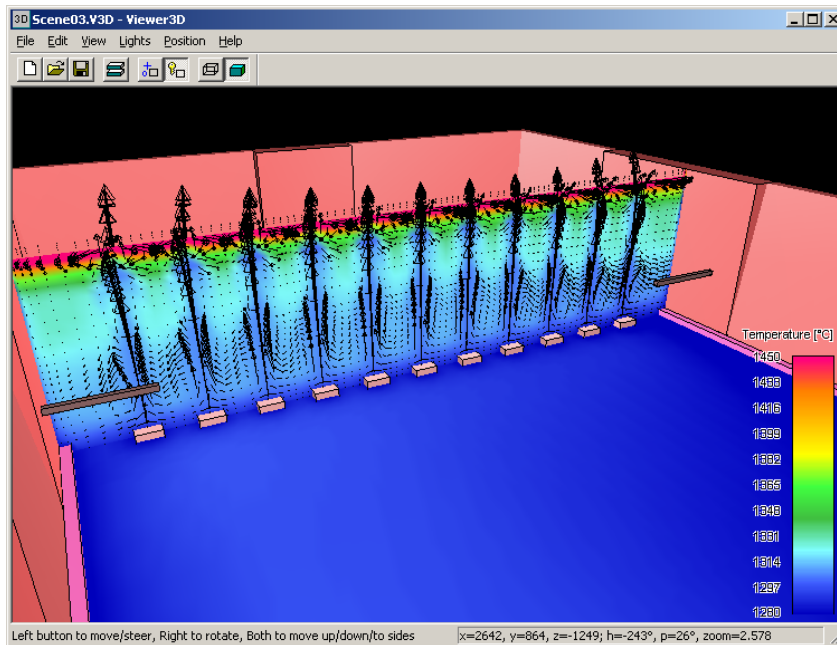
Obrázek 42 Ukázka zobrazení teplotních řezů ve vaně

Na následujícím obrázku 43 je vidět jeden vektorový řez v rovině XY, jenž popisuje směr a rychlost proudění roztavené skloviny ve vaně, výsledné vektory jsou obarveny teplotami.



Obrázek 43 Ukázka zobrazení vektorového řezu ve vaně

Obrázek 44 prezentuje kombinaci dvou teplotních řezů, první je příčný YZ v oblasti probublávače a druhý vodorovný XY nad dnem vany. Dále je zde zobrazen vektorový řez YZ v blízkosti teplotního příčného řezu. Cílem této sestavy komponent ve vaně bylo lépe vizualizovat požadovanou část modelu z hlediska technologického pochopení procesu.



Obrázek 44 Ukázka kombinace více různých komponent na scéně

Předchozí ukázky výstupů dokazují, že program dovoluje uživateli kombinovat různé typy komponent s požadovaným nastavením. Tato vizualizace může plně sloužit jako nástroj pro modeláře, kteří často potřebují znát průběžné výsledky jejich výpočetní práce, vycházející z matematických programů. Výsledné obrázky mohou být také použity jako podklad pro technickou diskuzi nebo pro prezentaci projektu u zákazníka.

ZÁVĚR

Tato bakalářská práce měla za cíl se seznámit s technologií tavení skla a vytvořit uživatelský program Viewer3D, který dokáže efektivně vizualizovat specifické výstupní datové soubory z matematických aplikací, řešící distribuci a optimalizaci tepla, proudění roztavené skloviny a další fyzikální procesy v modelu sklářské pece.

Teoretická část se zaměřila na popis procesu tavení skloviny, rozbor vstupních surovin a rozdělení skla z hlediska jeho využití. Dále byla rozebrána problematika projekce, transformace a geometrie, to z důvodu, že to jsou základní stavební kameny v oblasti vizualizace těles v trojrozměrném prostoru. Byly také probrány standardní technologie, jako jsou programovací jazyk C++, aplikační knihovna MFC a rozhraní knihovny OpenGL, které byly aplikovány pro návrh a realizaci programu.

Praktická část této práce zahrnuje detailní pohled na sklářskou pec, na její hlavní části a popis technologických zařízení, bez kterých by pec nemohla optimálně fungovat. V dalších kapitolách byl proveden rozbor vizualizačního programu z pohledu programátora, za pomoci jazyka UML a popisu jednotlivých tříd, hlavních metod a parametrů. Zde byl kladen důraz na popis zpracování datových souborů a následné zobrazení geometrie modelu a řezných rovin na scéně. V další části byl popsán program z pohledu uživatele, tedy modeláře nebo technologa. Zde byly vysvětleny jednotlivé možnosti, funkce a ovládání programu. Výklad zahrnuje pro lepší přehled i ukázky dialogových oken s návazným vysvětlením jejich vstupních parametrů. Na závěr této práce byl popsán soubor typu V3D a prezentovány grafické výstupy, jako názorné ukázky možností vizualizace výsledných dat.

Program byl testován na počítačích různých sestav, test se především zaměřil na grafické karty, které mohly způsobovat problémy při zobrazení. Výsledek byl pozitivní, program běžel na všech počítačích se standardní hardwarovou konfigurací.

Vizualizační program bude do budoucna rozšířen o další funkce. Nová verze bude tedy vylepšena o rozhraní pro zobrazení modelu spalovacího prostoru a nástroj, který dovládá uživateli interaktivní posuv řezů v libovolném směru. Další vývoj programu bude založen na požadavcích uživatelů, zabývající se návrhem, modelováním a optimalizací sklářských pecí.

ZÁVĚR V ANGLIČTINĚ

This bachelor thesis objective was to acquaint with a glass melting technology and to produce the Viewer3D program that can effectively visualize specific data files incoming from mathematical applications solving a temperature distribution and optimisation, velocity flow of glass melt and other physical processes in a glass furnace model.

In the theoretical part, the target was to describe a glass melting process, analysis of input raw materials and glass sorts in light of its usage. Furthermore, there were analyzed questions of projection, transformation and geometry because they are a base for the visualization of solids in a three-dimensional space. There was also thrashed out a question of standard technologies like the C++ programming language, MFC application library and OpenGL library interface that were applied for a design and realization of the program.

The practical part of this thesis includes a detailed view of glass furnace, a focus on its main parts and technological arrangements that are important for an optimal glass furnace operation. In the next chapters, an analysis was made for the visualization program in view of a programmer by the aid of the UML language and a description of particular classes, main methods and parameters. There was accentuated a characterization of data files processing and the subsequent visualization of model geometry and cutting planes in the scene. In the next part, there was described the program in view of an ordinary user as modeller and technologist. Herein, it specified particular possibilities, features and user control of the program. The commentary includes here, for a better understanding, a demonstration of dialog boxes with a consecutive explanation of their input parameters. At the conclusion of this thesis, there was described the file typed V3D and presented the graphics outputs as an illustration of visualization possibilities of resulting data.

The test was completed, the program ran on computers with different configurations, I especially focused on the graphics cards that could cause displaying problems. The result was positive, the program was able to operate on all computers with standard hardware.

The visualization program is going to be extended for additional features in the future. The new version will be improved for an interface to a visualization of the combustion space model and the tool that allows a user to interactively move cuts in any direction. A future program development will be based on requirements of users who are concerned with the desing, modelling and optimisation of glass furnaces.

SEZNAM POUŽITÉ LITERATURY

- [1] KLETEČKA, Jaroslav, FOŘT, Petr. *Technické kreslení*. 1. vyd. Praha : Computer Press, 1999. 193 s. ISBN 80-7226-192-4.
- [2] KRUGLINSKI, David J. *Mistrovství ve Visual C++*. Praha : Computer Press, 1999. 854 s. ISBN 80-7226-132-0.
- [3] POKORNÝ, Pavel. *Základy počítačové grafiky*. 1. vyd. Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta Technologická, 2004. 120 s. ISBN 80-7318-161-4.
- [4] PRATA, Stephen. *Mistrovství v C++*. 1. vyd. Praha : Computer Press, [1991?]. 966 s. ISBN 80-7226-339-0.
- [5] SEGAL, Mark, AKELEY, Kurt. *The OpenGL Graphics System: A Specification : Version 2.0*. Jon Leech and Pat Brown. [s.l.] : [s.n.], 2004. 382 s.
- [6] SHREINER, Dave. *OpenGL Programming Guide - Seventh Edition*. Boston : Pearson Education, Inc., 2009. 1019 s. ISBN 978-0-321-55262-4.
- [7] SIMONIS, Frank; BEERKENS, Ruud; DE WAAL, Henk. *Handbook for Glass Technologists : Course on Glass Manufacture* [online]. [s.l.] : TNO-TPD-Glass Technology, 1997 [cit. 2010-04-05]. Dostupné z WWW: <<http://www.tno.nl>>.
- [8] TUREK, Michal. *Vše o programování 3D grafiky s knihovnou OpenGL* [online]. CZ NeHe OpenGL, 2002 [cit. 2010-03-30]. Dostupné z WWW: <<http://nehe.ceske-hry.cz>>.
- [9] ŽÁRA, Jiří, BENEŠ, Bedřich, FELKEL, Petr. *Moderní počítačová grafika*. 1. vyd. Praha : Computer Press, 2005. 448 s. ISBN 80-7226-049-9.
- [10] *Glass Service, a.s.* [online]. SHARK publishing, 2006 [cit. 2010-03-29]. Dostupné z WWW: <<http://www.gsl.cz>>.
- [11] *Institut national polytechnique de Lorraine* [online]. LENSEM, 2009 [cit. 2010-03-30]. Glass Furnace. Dostupné z WWW: <<http://perso.ensem.inpl-nancy.fr/Pierre.Riedinger/publications/ISIE04.pdf>>.
- [12] *Maturita.cz* [online]. Internet Express, 1999 [cit. 2010-03-30]. Sklo a výroba skla. Dostupné z WWW: <<http://www.maturita.cz/referaty/referat.asp?id=6158>>.

- [13] *OpenGL : The Industry Foundation for High Performance Graphics* [online]. Khronos Group, 1997-2010 [cit. 2010-01-21]. Dostupný z WWW: <<http://www.opengl.org>>.
- [14] *OpenGL* [online]. Univerzita Karlova v Praze : Služby MFF, 1999, 31. 01. 2010 [cit. 2010-04-11]. Karlínský Lab. Dostupné z WWW: <<http://artax.karlin.mff.cuni.cz/~zajio1am/texts/opengl/opengl.html>>.
- [15] *ROOT.CZ : OpenGL* [online]. Internet Info, s.r.o, 1998 [cit. 2010-04-04]. Dostupné z WWW: <<http://www.root.cz>>.
- [16] *SKooDA wiki* [online]. Skooda, 2004, 2009/09/19 [cit. 2010-03-29]. Dostupné z WWW: <<http://wiki.skooda.org>>.
- [17] *Strojari.info* [online]. JakubSvestka.cz, 2009 [cit. 2010-03-30]. Technické zobrazování. Dostupné z WWW: <<http://www.strojari.info/clanek/6/technicke-kresleni/technicke-zobrazovani>>.
- [18] *TU v Liberci* [online]. Fakulta přírodovědně-humanitní a pedagogická, 2009 [cit. 2010-03-30]. Axonometrie. Dostupné z WWW: <http://www.fp.vslib.cz/kmd/lide/zackova/GE2/Pravouhla_axonometrie.pdf>.
- [19] *Wikipedie : Otevřená encyklopedie* [online]. MediaWiki, 2002 [cit. 2010-03-29]. Dostupné z WWW: <<http://cs.wikipedia.org>>.
- [20] *Západočeská univerzita* [online]. ZČU Plzeň, 2009 [cit. 2010-03-29]. Přednášky. Dostupné z WWW: <http://nemat.zcu.cz/MN/Prednasky/MN_6b_verze2009.doc>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

3D	Three Dimensional space.
AFX	Application Framework eXtensions.
API	Application Programming Interface.
C	Programovací jazyk ANSI C.
C++	Programovací jazyk ISO C++.
CAD	Computer Aided Design.
CDC	Class Device Context.
DC	Device Context.
DX	Délka v ose x.
DY	Délka v ose y.
DZ	Délka v ose z.
GDI	Graphics Device Interface.
GLU	OpenGL Utility Library.
GLUT	OpenGL Utility Toolkit.
HW	Hardwarové vybavení.
I	Jednotková vzdálenost v ose x.
IDE	Integrated Development Environment.
ISO	International Organization for Standardization.
J	Jednotková vzdálenost v ose y.
K	Jednotková vzdálenost v ose z.
MFC	Microsoft Foundation Class Library.
MoveX	Posunutí v ose x.
MoveY	Posunutí v ose y.
MoveZ	Posunutí v ose z.

OOP	Object Oriented Programming.
OpenGL	Open Graphics Library.
OS	Operating System.
PID	Regulátor, složený z Proporcionální, Integrovaná a Derivační části.
RGB	Barevný model popisující Red, Green, Blue.
RotXx	Rotace osy x ve směru osy x.
RotXy	Rotace osy x ve směru osy y.
RotXz	Rotace osy x ve směru osy z.
RotYx	Rotace osy y ve směru osy x.
RotYy	Rotace osy y ve směru osy y.
RotYz	Rotace osy y ve směru osy z.
RotZx	Rotace osy z ve směru osy x.
RotZy	Rotace osy z ve směru osy y.
RotZz	Rotace osy z ve směru osy z.
SDI	Single Document Interface.
SW	Softwarové vybavení.
U	Velikost vektoru v ose x.
UML	Unified Modeling Language.
V	Velikost vektoru v ose y.
V3D	Přípona souboru scény.
W	Velikost vektoru v ose z.
WTL	Windows Template Library.
X	Osa x světového souřadnicového systému.
XY	Řezná rovina ve vodorovném směru.
XYZ	Trojrozměrný prostor.

- XZ Řezná rovina v podélném směru.
- Y Osa y světového souřadnicového systému.
- YZ Řezná rovina v příčném směru.
- Z Osa z světového souřadnicového systému.

SEZNAM OBRÁZKŮ

Obrázek 1 Ukázka broušené vázy z olovnatého skla.....	14
Obrázek 2 Spalovací systém v příčně plamenné regenerativní peci.....	16
Obrázek 3 Distribuce tepla ve spalovacím prostoru a v tavící vaně	17
Obrázek 4 Příklad rozložení teplot ve vaně.....	18
Obrázek 5 Příklad rovnoběžného a středového promítání	20
Obrázek 6 Orientace promítání v prvním a třetím kvadrantu.	22
Obrázek 7 Příklad kabinetního promítání.....	23
Obrázek 8 Příklad kavalírského promítání	23
Obrázek 9 Příklad izometrie	24
Obrázek 10 Příklad dimetrie.....	24
Obrázek 11 Příklad trimetrie	25
Obrázek 12 Příklad pravoúhlé axonometrie	25
Obrázek 13 Jednobodová perspektiva	26
Obrázek 14 Dvoubodová perspektiva.....	26
Obrázek 15 Třibodová perspektiva.....	27
Obrázek 16 Proces vykreslování přes framebuffer	33
Obrázek 17 Ukázka aditivního barevného míchání ze tří složek	35
Obrázek 18 Model regenerativní pece na výrobu lahvového skla	37
Obrázek 19 Tavící vana regenerativní pece.....	39
Obrázek 20 Ukázka zakladače vsázky	40
Obrázek 21 Ukázka rozmístění elektrod ve vaně	41
Obrázek 22 Ukázka simulace proudění v průtokovém kanálu	41
Obrázek 23 Ukázka simulace proudění v oblasti probublávače	42
Obrázek 24 Pohled na umístění bariéry ve vaně	42
Obrázek 25 Diagram základní konstrukce programu	44
Obrázek 26 Ukázka diskrétní a základní geometrie modelu	46
Obrázek 27 Diagram implementovaných dialogových oken	49
Obrázek 28 Diagram tříd geometrie modelu pece	52
Obrázek 29 Diagram tříd pro parametrizaci řezných rovin.....	55
Obrázek 30 Ukázka geometrie v podobě drátěného modelu	56
Obrázek 31 Diagram tříd pro renderování jednotlivých komponent.....	57

Obrázek 32 Základní okno vizualizačního programu.....	59
Obrázek 33 Ukázka otevření datového souboru v programu.....	60
Obrázek 34 Ukázka dialogového okna Components.....	61
Obrázek 35 Ukázka dialogového okna Scales.....	61
Obrázek 36 Ukázka dialogového okna Scalar and vector slices	62
Obrázek 37 Ukázka dialogového okna Model geometry.....	63
Obrázek 38 Ukázka dialogového okna Components material.....	64
Obrázek 39 Ukázka dialogového okna Head-up Light.....	65
Obrázek 40 Ukázka funkcí v panelu nástrojů.....	66
Obrázek 41 Ukázka zobrazení geometrie modelu pece	68
Obrázek 42 Ukázka zobrazení teplotních řezů ve vaně	69
Obrázek 43 Ukázka zobrazení vektorového řezu ve vaně	69
Obrázek 44 Ukázka kombinace více různých komponent na scéně.....	70

SEZNAM PŘÍLOH

- P I Dokumentační CD obsahující v elektronické podobě zdrojové kódy a uživatelský manuál k programu Viewer3D, výsledný datový model sklářské pece, soubory s ukázkovými scénami a tuto bakalářskou práci.