



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SIMULÁTOR 3D TISKÁRNY

SIMPLE 3D PRINTER SIMULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VLADIMÍR ČILLO

VEDOUCÍ PRÁCE

SUPERVISOR

Dr. Ing. PETR PERINGER

BRNO 2015

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací simulátoru 3D tiskárny v jazyce C++ s využitím knihovny Qt. Simulátor předpokládá tiskárnu založenou na nejrozšířenější metodě 3D tisku – Fused Deposition Modeling. Vstupem simulátoru je 3D model ve formátu STL. Pro daný 3D model simulátor poskytuje odhad celkové doby tisku. Součástí simulace je také vizualizace procesu 3D tisku. Výsledkem práce je přenositelný program testovaný pod operačními systémy Linux a Windows. Funkčnost simulátoru byla experimentálně ověřena na příkladech.

Abstract

This thesis deals with design and implementation of 3D printer simulator in C++ using Qt. The simulator assumes a printer based on widely used method of 3D print – Fused Deposition Modeling. An input of simulator is a 3D model in STL file format. For the 3D model simulator provides an estimate of the total print time. Simulation also includes visualization of 3D print. The result is a cross-platform program tested under Linux and Windows. Functionality of simulator has been tested by several experiments.

Klíčová slova

3D tisk, GCode, simulace, C++, 3D vizualizace, animace

Keywords

3D print, GCode, simulation, C++, 3D visualization, animation

Citace

Vladimír Čillo: Simulátor 3D tiskárny, bakalářská práce, Brno, FIT VUT v Brně, 2015

Simulátor 3D tiskárny

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Dr. Ing. Petra Peringerera a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Vladimír Čillo
19. mája 2015

Poděkování

Ďakujem vedúcemu tejto bakalárskej práce, Dr. Ing. Petrovi Peringerovi, za množstvo cenných rád a odbornú pomoc pri riešení práce.

© Vladimír Čillo, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Princípy 3D tlačiarňí a ich simulácie	3
2.1 3D tlač	3
2.2 Projekt RepRap	5
2.3 Technológie 3D tlače	5
2.4 Prevod 3D modelu na GCode	6
2.5 G-Code	7
2.6 Simulácia 3D tlače	8
2.7 Existujúce simulátory	9
3 Návrh simulátoru	11
3.1 Požiadavky na simulátor	11
3.2 Abstraktný model 3D tlačiarne	12
3.3 Návrh vizualizácie	15
3.4 Grafické užívateľské rozhranie	15
3.5 Objektový návrh simulátoru	16
4 Implementácia simulátoru a overenie jeho funkčnosti	21
4.1 Implementačné prostredie	21
4.2 Implementácia	21
4.3 Časová a pamäťová zložitosť simulácie	24
4.4 Porovnanie s existujúcimi riešeniami	25
4.5 Overenie validity modelu 3D tlačiarne	26
5 Záver	30
A Obsah CD	32

Kapitola 1

Úvod

Technológia 3D tlače dnes nachádza uplatnenie v mnohých oblastiach vedy a techniky a na základe aktuálneho dopytu a rýchlosti vývoja sa v budúcnosti predpokladá jej veľký rozvoj. Praktické aplikácie tejto technológie v súčasnosti siahajú od rýchleho prototypovania až po tlač rodinných domov. Prirodzene tak vzniká tlak na zlepšovanie parametrov tlače, najmä pokiaľ ide o rýchlosť a presnosť. Z toho pramení potreba softwarových nástrojov akými sú simulátory, vizualizátory a pod., ktoré sú dnes samozrejmom súčasťou výskumu a vývoja v takmer každom odvetví technickej praxe.

Napriek tomu, že počiatky 3D tlače siahajú až do roku 1986, v súčasnosti sa počet 3D simulátorov 3D tlačiarň pohybuje rádovo v jednotkách. Naopak, veľmi rozšírené sú 2D simulátory, ktoré neumožňujú sledovať priebeh tlače z ľubovoľného uhla pohľadu a väčšinou zobrazujú len prierez práve tlačeného modelu. Súčasný stav je podmienený okrem iného tým, že existuje množstvo typov 3D tlačiarní, ktoré sa líšia nielen množinou inštrukcií, ale často aj samotnou technológiou tlače. Je preto náročné vytvoriť simulátor, ktorý by dokonale simuloval správanie viacerých konkrétnych modelov 3D tlačiarní súčasne.

Cieľom práce je navrhnúť a implementovať simulátor vybraného typu 3D tlačiarne s podporou 3D vizualizácie procesu tlače. Dôležitým predpokladom pre vytvorenie akéhokoľvek simulátoru je dôkladné porozumenie simulovanému systému, ktorým je v tomto prípade 3D tlačiareň. V druhej kapitole sa preto práca zaoberá základnými princípmi fungovania 3D tlačiarní najmä na logickej úrovni, hardwarové komponenty 3D tlačiarne sú zmienené len v nutnej miere. Súčasťou druhej kapitoly je aj stručný prehľad existujúcich riešení. Tretia kapitola popisuje objektovo-orientovaný návrh simulátoru a jeho reprezentáciu pomocou jazyka UML a následne pojednáva o metóde vizualizácie procesu tlače, ktorá je použitá v simulátore. Štvrtá kapitola sa zaoberá popisom funkčnosti simulátoru a jej overeniu na niekoľkých príkladoch s následným zhodnotením validity modelu tlačiarne.

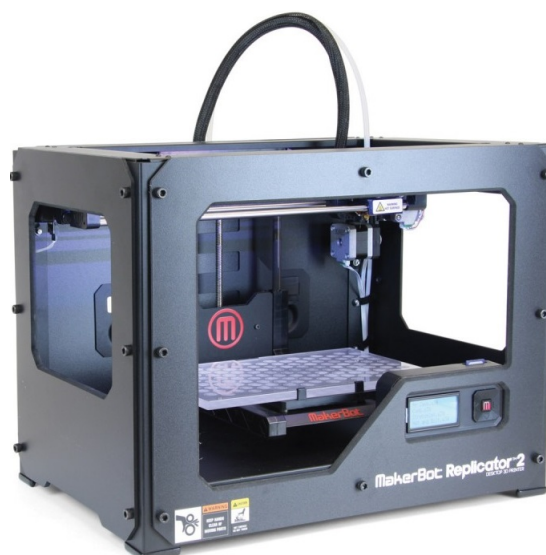
Kapitola 2

Princípy 3D tlačiarní a ich simulácie

Cieľom tejto kapitoly je poskytnúť úvod do problematiky 3D tlačiarní a vysvetliť základné pojmy a princípy používané v oblasti 3D tlače. Záver kapitoly je venovaný teoretickým východiskám simulácie a vizualizácie 3D tlače.

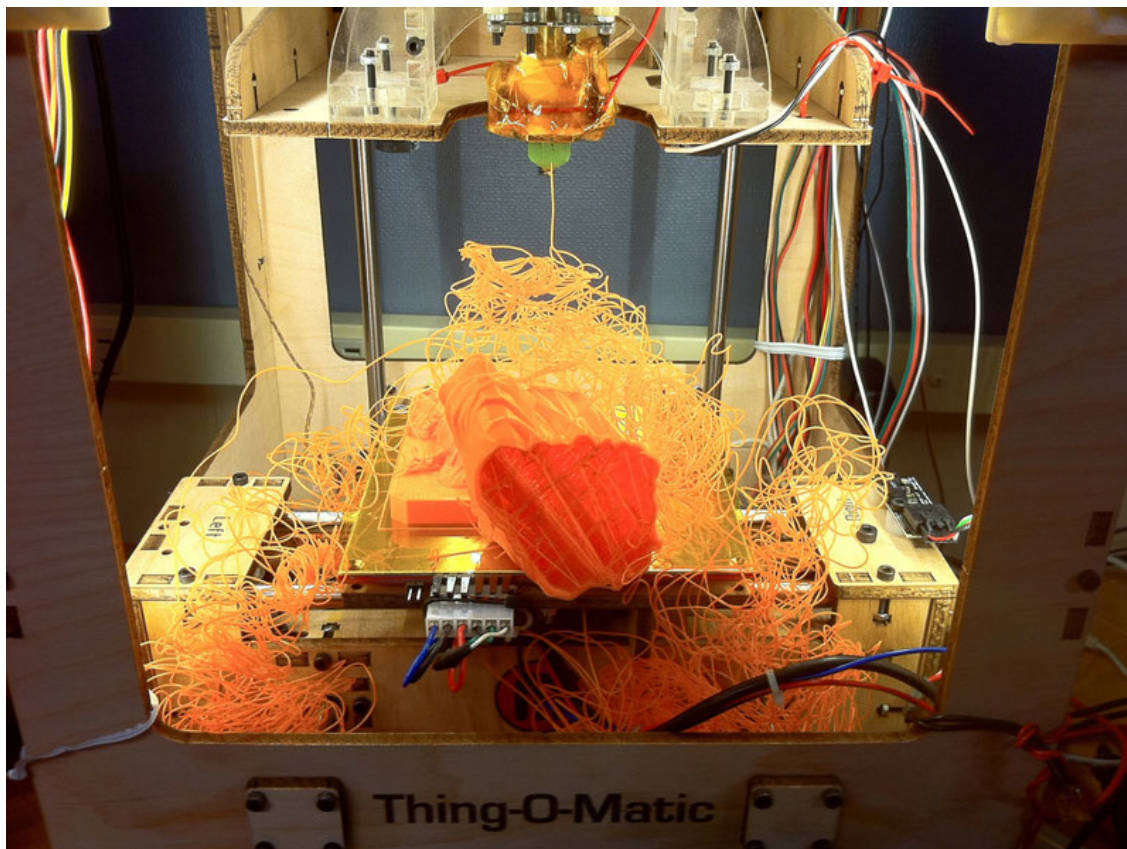
2.1 3D tlač

3D tlač je odpoveďou na otázku, ako automaticky vytvoríť 3D objekt na základe digitálnych dát. Obrovský potenciál 3D tlače podnietil rýchly vývoj tejto technológie, výsledkom čoho je široké uplatnenie 3D tlačiarní v praxi, kde sa môžeme stretnúť s modelmi určenými na priemyselné využitie, ktoré veľkosťou zvyčajne niekoľkonásobne prevyšujú klasické spotrebiťské modely a častokrát využívajú osobitnú technológiu výroby. Príkladom je technológia BAAM, ktorá umožňuje vyrábať objekty o veľkosti viac ako 30 metrov, pre letecký a iný priemysel.^[4]



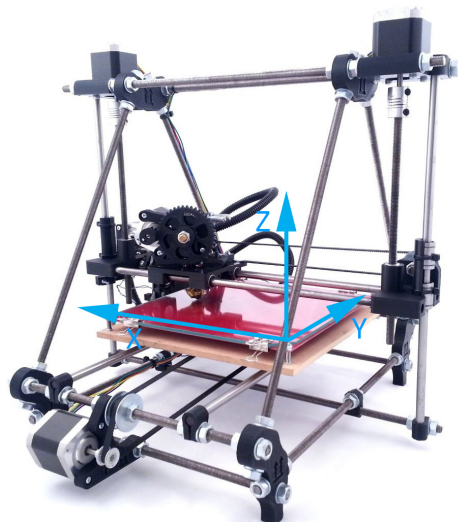
Obr. 2.1: Makerbot

Využitie 3D tlačne má však stále bližšie k prototypovaniu ako k masovej výrobe, dôvodom čoho je stále pomalá rýchlosť tlačne. Nie je ničím neobvyklým, keď výroba pomerne malého predmetu, na ktorý sa spotrebuje 400cm^3 materiálu, trvá viac ako 20 hodín. Technológia 3D tlačne teda ešte určite nie je na svojom vrchole a poskytuje veľký priestor pre inovácie.



Obr. 2.2: Úkáža chyby pri 3D tlačni

V rámci tejto práce sa budeme zameriavať výhradne na spotrebiteľské modely 3D tlačni, ktoré sa stávajú čoraz populárnejšie aj napriek svojej relatívne vysokej cene. Pravdepodobne najznámejším modelom v tejto kategórii je MakerBot Replicator 2 (obr. 2.1). Rozširovaniu 3D tlačni medzi bežných užívateľov v súčasnosti stále bráni aj ich pomerne zložitá konfigurácia a kalibrácia. Nesprávne nastavenie môže viesť k chybovým stavom a úplnému zničeniu vyrábaného modelu (viz napr. obr. 2.2). Okrem toho, proces 3D tlačne postráda automatizáciu, na ktorú sme zvyknutí u obyčajných tlačni.



Obr. 2.3: RepRap Mendel

Typická 3D tlačiareň pozostáva z niekoľkých častí. Hlava 3D tlačiarne, tzv. **extrudér** (niekedy aj hot end) je časť zodpovedná za roztavenie surového materiálu a jeho formovanie do súvislého profilu. Tlačiareň ďalej pozostáva najmä z častí, ktoré umožňujú pohyb extruderu v smere osí X, Y a Z tak, ako to je znázornené na obrázku 2.3. Pohyb je realizovaný prostredníctvom motorov, ktoré riadi príslušná elektronika. Princípy, na ktorých je založený tento pohyb boli prevzaté z technológie CNC strojov, kde sa s problematikou pohybu v 3D priestore stretávame v súvislosti s obrábaním výrobku. Technická realizácia pohybu sa u jednotlivých modelov môže líšiť, dôkazom je napríklad model RepRap Delta-Bot (obr.2.4). Dôležitou súčasťou každej 3D tlačiarne je v neposlednom rade podložka, t.j. plocha na ktorej vzniká výsledný model.

2.2 Projekt RepRap

K rozšíreniu 3D tlačiarňí významným spôsobom prispel projekt RepRap. RepRap je prvý projekt open source 3D tlačiarne. Open source v tomto prípade znamená, že návod, ako túto 3D tlačiareň poskladať, je možné zdarma stiahnuť na internete. V súčasnosti v rámci projektu RepRap existuje viacero modelov 3D tlačiarňí, ich kompletný zoznam je možné nájsť na wiki-stránke tohto projektu [8]. Tlačiarne RepRap sú zložené z plastových súčiastok, takže je teoreticky možné vytlačiť inú 3D tlačiareň, RepRap býva preto označovaný za samoreplikujúci sa stroj. V praxi to nie je úplne pravda, pretože súčasťou 3D tlačiarne sú aj časti umožňujúce pohyb tlačovej hlavy (elektronika, motory), ktoré RepRap tlačiarne nevyrobajú.

2.3 Technológia 3D tlače

V súčasnosti sa v rámci 3D tlače stretávame s viacerými spôsobmi vytvárania objektov, v podstate sa však vždy jedná o aditívny spôsob výroby, t.j. postupné pridávanie materiálu k vznikajúcemu modelu. Medzi najznámejšie technológie 3D tlače sa radí Stereolitografia (SLA), Selective Laser Sintering (SLS) a Fused Deposition Modeling (FDM). Kompletný



Obr. 2.4: 3D tlačiareň RepRap Deltabot

popis jednotlivých technológií je nad rámec tejto práce, a keďže väčšina 3D tlačiarňí - hlavne modely orientované na amatérskych priaznivcov 3D tlače — používa FDM [1], bude práve táto technológia predmetom implementácie simulátoru.

2.3.1 Princíp metódy FDM

Materiál — najčastejšie termoplastický polymér — vo forme vlákna sa v extrudéri roztaví, pričom je nanášaný najprv na podložku a potom na takto vzniknuté vrstvy. Teplota extrudéry je udržiavaná blízko teploty topenia použitého materiálu, čím sa materiál dostáva do polotekutého stavu. Keďže teplota okolitého prostredia je výrazne nižšia, materiál, ktorý opúšťa tlačovú hlavu, rýchlo tuhne [12]. Vrstvu po vrstve tak vzniká výsledný model.

2.4 Prevod 3D modelu na GCode

Napriek tomu, že jednotlivé modely 3D tlačiarňí sa od seba typicky odlišujú — na úrovni hardware, aj firmware — logický proces výroby 3D modelu je spoločný pre všetky zariadenia. Na vstupe tohto procesu je 3D model v digitálnom formáte, ktorý si buď vytvorí sám užívateľ, alebo je možné takýto model stiahnuť z jednej z online databází modelov. Následne je potrebné daný model konvertovať na tzv. **G-Code** [3] — jazyk inštrukcií, ktorým je riadená tlač. Vygenerovaný G-Code je priamym vstupom príslušnej 3D tlačiarne, tzn. firmware danej tlačiarne interpretuje jednotlivé inštrukcie zapísané vo vstupnom G-Code súbore. Interpretáciou v tomto prípade rozumieme vykonanie konkrétnej akcie, napr. pohyb extrudéry na dané miesto a pod.

Proces transformácie modelu na G-Code sa nazýva **Slicing** (z angl. slice — nakrájať,

narezať). Hlavná myšlienka je vygenerovať inštrukcie tak, aby sa pri ich interpretácii pohyboval extrudér buď horizontálne — v jednotlivých vrstvách, alebo vertikálne, pri prechode z jednej vrstvy na vyššiu. Generovanie inštrukcií zahŕňa výpočet optimálnej orientácie modelu, generovanie podporného materiálu (aby žiadna časť modelu nebola počas tlače vo vzduchu), generovanie ciest pre vyplnenie jednotlivých vrstiev a výpočet potrebného materiálu. Celý proces je algoritmizovateľný a v súčasnosti je dostupných viacero komerčných, ale aj opensource nástrojov. Výstupom celého procesu je textový súbor s inštrukciami.

2.5 G-Code

G-code je jazyk definovaný štandardom ISO 6983 [3]. Umožňuje definovať postupnosť operácií, ktoré sú priamo realizovateľné hardwarom 3D tlačiarne. Pre ilustráciu toho, ako G-Code vyzerá, uvádzam krátku ukážku:

```
M109 S200
G90
G92 E0
M82
G1 F1800.000 E-1.00000
G92 E0
G1 Z0.350 F7800.000
G1 X47.333 Y44.253 F7800.000
G1 E1.00000 F1800.000
G1 X49.226 Y42.949 E1.07480 F1080.000
G1 X52.000 Y42.343 E1.16718
```

Uvedený súbor inštrukcií bol vybraný zo súboru, ktorý má viac ako 200 000 riadkov. Počet inštrukcií je prirodzene priamo úmerný zložitosti modelu, zvyčajne sa pohybuje rádovo v stovkách tisíc.

2.5.1 Syntax a sémantika inštrukcií

G-Code je svojou syntaxou veľmi jednoduchý. Každá inštrukcia, vrátane všetkých parametrov, sa nachádza na samostatnom riadku. Prvý identifikátor definuje typ inštrukcie. Podľa typu inštrukcie potom nasledujú prípadné parametre. Keďže Reprap definuje viac ako 250 rôznych inštrukcií [2], je nad rámec tejto práce venovať sa podrobne každej z nich. Avšak viac ako 99% z celkového počtu inštrukcií v priemernom G-Code súbore tvoria inštrukcie typu G1, preto na príklade vysvetlím sémantiku tejto inštrukcie. Príklad inštrukcie G1:

```
G1 X102.754 Y106.854 E10.40321 F3600.000
```

G1 je inštrukcia Move — teda pohyb. Pri vykonaní tejto inštrukcie sa extrudér pohne zo svojej aktuálnej polohy do polohy danej jednotlivými parametrami. Parametre **X**, **Y** a **Z** definujú súradnice bodu, do ktorého sa má extrudér dostať. Ak niektorý z týchto parametrov chýba (v tomto prípade **Z**), znamená to, že jeho hodnota sa nemení — tzn. ostáva rovnaká ako v aktuálnej polohe. Zaujímavosťou je, že tieto tri parametre sa nikdy nevyskytujú všetky naraz. V súlade s filozofiou slicing (2.4) sa totiž extrudér pohybuje buď len horizontálne, t.j. v určitej vrstve, alebo len vertikálne - pri prechode na vyššiu

vrstvu. Parameter **E** udáva množstvo materiálu, ktoré sa má pri danom pohybe použiť. Parameter **F** definuje rýchlosť, pri ktorej sa má tento pohyb uskutočniť. Typicky sa udáva v milimetroch za minútu. Hodnota tohto parametra však nemusí znamenať, že pohyb sa reálne uskutoční pri tejto rýchlosti. Môže sa napríklad stať, že vzdialenosť od aktuálnej do cieľovej polohy v kombinácii s maximálnym zrýchlením nie je dostatočná na dosiahnutie požadovanej rýchlosti. Podobne je možná situácia, kedy sa extrudér v dôsledku predošlých inštrukcií pohybuje tak rýchlo, že na požadovanú rýchlosť nestihne pri danej vzdialenosti spomaliť.

2.6 Simulácia 3D tlače

Doteraz sme sa v tejto kapitole zaoberali princípmi 3D tlačiarňí. Pre vytvorenie teoretického základu pre simulátor 3D tlačiarne je potrebné tieto princípy spojiť s poznatkami z oblasti simulácii. Vo fáze návrhu simulátoru bude totiž v centre pozornosti vytváranie abstraktného modelu. V rámci tejto práce sa pri vytváraní abstraktného modelu 3D tlačiarne budem zameriavať najmä na matematický popis pohybu extrudéru, ktorý by mal poskytnúť pomerne presné informácie o polohe extrudéru v konkrétnom čase a teda aj o celkovej dobe tlačovej úlohy. Na základe toho bude možné vytvoriť program — simulačný model, ktorý bude pomocou 3D grafiky a animácií zobrazovať priebeh tlačovej úlohy s prípadnou možnosťou zrýchlenia, pozastavenia a zobrazovania práve spracovávaných inštrukcií. Takto vytvorený abstraktný model bude samozrejme do značnej miery zjednodušený oproti skutočnej 3D tlačiarňi. Diskusia o dopade tohto zjednodušenia na validitu modelu sa nachádza v sekcii 4.5.

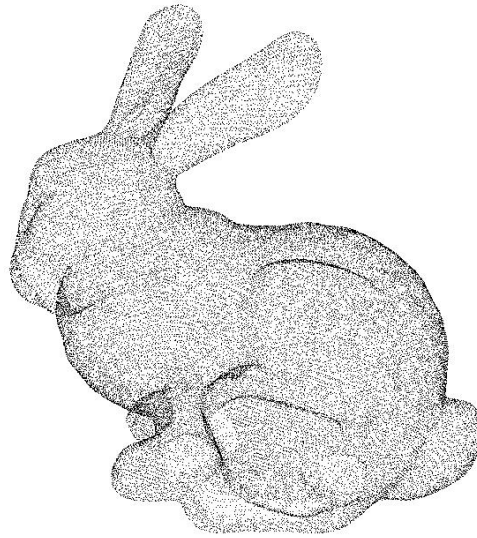
2.6.1 Simulačný model

Z programátorského hľadiska je simulátor 3D tlačiarne v podstate interpret G-Code, tzn. na úrovni spracovania G-Code súboru zahŕňa lexikálnu a syntaktickú analýzu. Samotná interpretácia potom spočíva v sekvenčnom vykonaní všetkých inštrukcií, pričom konkrétna akcia je daná typom inštrukcie a jej parametrami. Keďže počet inštrukcií je dopredu známy (po spracovaní zdrojového súboru). Algoritmus riadenia simulácie je cyklus, kde počet iterácií je daný počtom inštrukcií. V prípade, že výsledkom interpretácie danej inštrukcie je pohyb extrudéru, je potrebné pri simulácii vypočítať a zohľadniť dobu trvania daného pohybu.

2.6.2 Vizualizácia 3D tlače

Vizualizácia je akákoľvek technika vytvárania obrázkov, diagramov a animácií s cieľom prezentovať určitú informáciu. V kontexte počítačovej 3D grafiky nie je vizualizácia novou oblasťou, avšak, ako sa technológia vyvíja, vizualizácia sa stáva veľmi silným nástrojom vo viacerých disciplínach nielen v oblasti vedy. Vo svete 3D tlače sa s 3D vizualizáciou stretávame hlavne v súvislosti s vytváraním modelov, to však nie je predmetom tejto práce.

Vizualizácia 3D tlače sa od klasickej práce s 3D modelmi odlišuje tým, že model na grafickej scéne vzniká postupne, tzn. dopredu nevieme, kde ležia jednotlivé vrcholy a nepoznáme ani normály k povrchu. Vytvárať pri interpretácii G-Code priamo polygonálny model je preto v zásade zložitá úloha. Inštrukcie neposkytujú žiadne *priame* informácie o tom, ako by mal výsledný model vyzeráť, spracovaním G-Code je však možné získať tzv. mračno bodov (point cloud — viz obr. 2.5). Mračno bodov poskytuje pomerne dobrú predstavu o tom, ako daný model vyzerá, avšak z hľadiska 3D grafiky sa nejedná o reprezentáciu



Obr. 2.5: 3D model reprezentovaný mračnom bodov [5]

modelu, ktorá by počítala s osvetľovacím modelom a preto je typicky žiadúce konvertovať mračno na polygonálny model.

Vizualizácia 3D tlače je teda vo svojej podstate animácia, ktorá začína s prázdnu scénou, na ktorej postupne vzniká výsledný model. Vizualizácia procesu vzniku 3D modelu by mala zohľadňovať použitú metódu 3D tlače a vychádzať z pozorovaní skutočných 3D tlačiarňí. Potenciálny zdroj inšpirácie predstavujú aj existujúce simulátory.

2.7 Existujúce simulátory

V súčasnosti existuje viacero riešení, ktoré sa venujú problematike simulácií 3D tlače. Uvádzam prehľad nástrojov, ktoré sú z hľadiska tejto práce zaujímavé svojou funkčnosťou a prístupom k vizualizácii 3D tlače.

2.7.1 GCodeSimulator

GCodeSimulator sa stal populárnym vďaka aplikácii pre Android, ktorá umožňuje jednoducho ovládať 3D tlačiareň z mobilných zariadení. Počas simulácie zobrazuje zostávajúci čas, umožňuje simuláciu zrýchliť. Pri vizualizácii tlače zobrazuje každú vrstvu inou farbou, samotná vizualizácia je však 2D. K dispozícii je pohľad zhora, spredu a zo strany. Verzia pre PC je napísaná v Jave, zdrojové kódy nie sú dostupné.

Domovská stránka: <http://gcodeprintr.dietzm.de/#GCodeSimulator>

2.7.2 CNC Simulator Pro

Okrem základnej funkčnosti poskytuje tento simulátor aj prepracovanú 3D vizualizáciu procesu tlače, počas ktorej vypisuje práve vykonávané inštrukcie. Bohužiaľ, nástroj je komerčný a nemám ho k dispozícii. V príspevku na oficiálnom blogu [11] ho autori označili za vôbec prvý simulátor 3D tlačiarne. Súčasťou príspevku je aj video, na ktorom je vidieť veľmi presvedčivú 3D vizualizáciu.

Domovská stránka: <http://cn simulator.info/>

2.7.3 gCodeViewer

gCodeViewer je online nástroj, ktorý funguje v rámci prehliadača. Nejedná sa o kompletný simulátor 3D tlače, avšak program umožňuje užívateľovi nahrať G-Code súbor, na základe ktorého vytvára jednoduchú 3D vizualizáciu príslušného objektu. Zároveň poskytuje informácie o predpokladanej dobe tlače a množstve spotrebovaného materiálu. Nástroj je dostupný na svojej domovskej stránke.

Domovská stránka: <http://gcode.ws/>

2.7.4 GCode Viewer

Ďalší nástroj z kategórie online, avšak tentokrát pôsobí 3D vizualizácia prepracovanejšie a zobrazuje aj postupné vykonávanie jednotlivých inštrukcií. Extrudér nie je súčasťou vizualizácie, ale postupne pribúdajúci materiál je vizualizovaný rôznymi farbami.

Domovská stránka: <http://jherrm.com/gcode-viewer/>

2.7.5 Willit 3D Print

Willit 3D Print je najkomplexnejší nástroj, na ktorý som narazil. Jeho vstupom je priamo 3D model vo formáte stl, ktorý si program interne konvertuje na GCode. Užívateľ má možnosť zvoliť si typ tlačiarne, typ použitého materiálu a hrúbku vrstvy. Samozrejmosťou je detailná vizualizácia 3D modelu, avšak je možné, že táto vizualizácia je založená priamo na .stl súbore a nie na GCode, keďže zdrojové kódy programu nie sú voľne dostupné, túto informáciu nie je možné overiť. Výstupom programu je predpokladaný čas tlače vstupného modelu a zároveň informácia o tom, či model neobsahuje chyby a je možné ho bez problémov vyrobiť. Aj tento nástroj patrí do kategórie online.

Domovská stránka: <http://www.willit3dprint.com/>

Kapitola 3

Návrh simulátoru

Táto kapitola popisuje návrh modelu a simulátoru 3D tlačiarne s využitím poznatkov z kapitoly 2. Štruktúra kapitoly odpovedá návrhu metódou zhora dolu. V úvode je definovaná požadovaná funkčnosť simulátoru, následne predkladám návrh modelu 3D tlačiarne, ktorý umožní splniť definované požiadavky. Súčasťou kapitoly je aj popis návrhu grafického užívateľského rozhrania. V závere kapitoly popisujem návrh metódy vizualizácie procesu tlače. Výsledkom návrhu je diagram tried, z ktorých pozostáva implementovaný simulátor, uvedený v sekcii 3.5.

3.1 Požiadavky na simulátor

Pod pojmom simulátor obvykle myslíme niečo, čo umožňuje vytvoriť nejakým spôsobom realistické napodobenie určitej činnosti. V kontexte 3D tlačiarne môžu byť požiadavky na simulátor pomerne rozsiahle. Základná funkcionálna zahrňa vizualizáciu procesu tlače, informácie o dobe trvania tlačovej úlohy (celkový čas, zostávajúci čas), informácie o množstve potrebného materiálu a zobrazenie výsledného modelu po dokončení úlohy s prípadnou možnosťou náhľadu pre rôzne materiály. Ďalšou z požiadaviek môže byť podpora viacerých extrudérov, čo by umožnilo pri vizualizácii tlače farebne odlišiť jednotlivé materiály.

Simulátor môže byť doplnený o modelovanie termodynamických dejov, čo by sa dalo využiť napríklad pri realistickej vizualizácii procesu tlače (simulácia topenia materiálu). Z praktického hľadiska by bolo ideálne mať k dispozícii simulátor, ktorý by dokázal priamo zo sekvencie vykonávaných inštrukcií detekovať prípadne chyby a poruchy, ku ktorým občas pri 3D tlači dochádza (viz obr. 2.2). Zoznam vlastností, ktoré by bolo možné v rámci 3D tlače simulovať, by mohol pokračovať. Je zrejmé, že ideálny simulátor svojou zložitou presahuje úroveň bakalárskej práce. Výsledkom návrhu požiadaviek na simulátor je preto nasledovná funkcionálna:

- 3D vizualizácia procesu tlače
- Zobrazovanie informácie o celkovej dobe trvania danej úlohy
- Pozastavenie a zrýchlenie simulácie
- Zobrazovanie informácie o množstve potrebného materiálu
- Zobrazovanie práve vykonávaných inštrukcií
- Zobrazenie výsledného modelu

3.2 Abstraktný model 3D tlačiarne

Ako som naznačil v časti 2.6, úlohou tohto modelu je definovať správanie, ktoré čo najrealistickejšie simuluje vykonávanie jednotlivých inštrukcií. Vzhľadom na požiadavky definované v časti 3.1 je potrebné spracovávať a interpretovať inštrukcie týkajúce sa pohybu extrudéru. Simulácia pohybu extrudéru tvorí jadro celého simulátoru. Ostatné inštrukcie, týkajúce sa napríklad zmeny teploty, sú ignorované. V prípade niektorých inštrukcií je obtiažne určiť dobu trvania takejto inštrukcie a preto sa táto doba nezohľadňuje.

Abstraktný model predpokladá model 3D tlačiarne, ktorý má pohyb v jednotlivých osiach nezávislý, t.j. tlačiareň má 3 nezávislé motory, ktoré umožňujú pohyb v smere osí X, Y a Z. Navrhovaný simulátor je teda kompatibilný s akýmkoľvek modelom 3D tlačiarne ktorý spĺňa túto podmienku, dodržiava štandard [3] (G-Code). Pre konkrétny model je však potrebné odpovedajúco nastaviť maximálne hodnoty zrýchlenia v smere jednotlivých osí. Prednastavené hodnoty zrýchlení extrudéru v navrhovanom modeli 3D tlačiarne sú 500 mm/s^2 pre pohyb v smere osi X a Y a 20 mm/s^2 pre pohyb v smere osi Z. Tieto hodnoty odpovedajú prednastaveným hodnotám v implementácii RepRap firmware [9].

3.2.1 Pohyb extrudéru

Model 3D tlačiarne definuje maximálne zrýchlenie extrudéru v smere jednotlivých osí. Spracovávané inštrukcie poskytujú informáciu o jednotlivých bodoch, ktorými extrudér postupne prechádza, a o požadovanej rýchlosti, ktorou sa má tento pohyb uskutočniť. Z týchto informácií je možné vypočítať, ako dlho bude daný pohyb trvať. Súčtom časov všetkých pohybov získame celkovú dobu tlače.

Keďže zrýchlenia sú definované len v smere jednotlivých osí¹, v prípade pohybu v rovine XY je potrebné vypočítať maximálne zrýchlenie, ktoré odpovedá smeru daného pohybu. Pre toto zrýchlenie platí vzťah:

$$\vec{a}_{v_{max}} = \vec{a}_{x_{max}} \cos \alpha + \vec{a}_{y_{max}} \sin \alpha \quad (3.1)$$

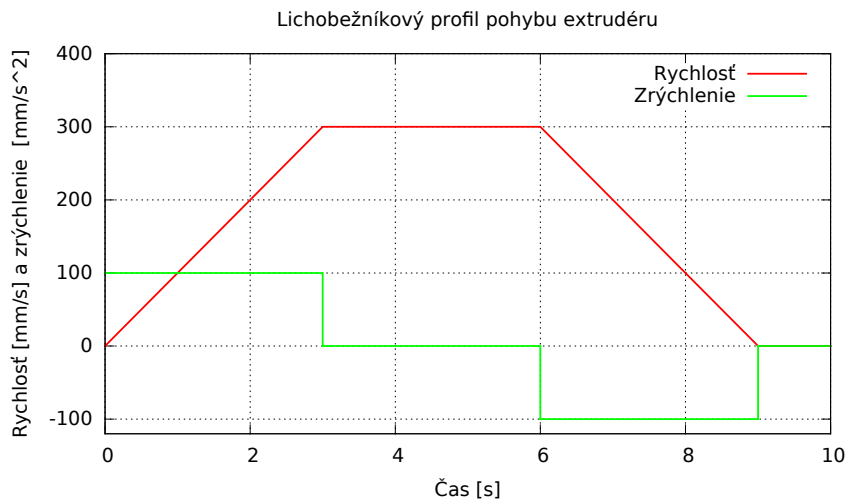
kde $\vec{a}_{x_{max}}$ je zrýchlenie v smere osi X, α je uhol, ktorý zvierá smerový vektor pohybu s osou X, $\vec{a}_{y_{max}}$ je zrýchlenie v smere osi Y a $\vec{a}_{v_{max}}$ je výsledné maximálne zrýchlenie v smere pohybu.

Rýchlosť, ktorou sa má uskutočniť pohyb, buď udáva daná inštrukcia G1, alebo sa použije posledná nastavená hodnota — viz 2.5.1. Na základe aktuálnej a cieľovej polohy extrudéru sa vypočíta vzdialenosť, ktorú je potrebné prejsť. Vektor medzi počiatočným a koncovým bodom udáva smer pohybu, na základe ktorého sa určí maximálne zrýchlenie pre daný smer podľa rovnice 3.1.

Z informácií o celkovej vzdialenosti, aktuálnej rýchlosti a zrýchlení, ktoré máme k dispozícii pre daný pohyb, vieme zistiť, aký *profil* bude mať daný pohyb. Lichobežníkový profil sa uplatní v prípade, že počas daného pohybu extrudér stihne zrýchliť na požadovanú rýchlosť, určitý úsek pohybu prejde požadovanou rýchlosťou a nakoniec spomalí na koncovú rýchlosť. Trojuholníkový profil je charakteristický pre pohyb, pri ktorom extrudér stihne na danej vzdialenosti dosiahnuť požadovanú rýchlosť, ale hneď nato musí začať spomaľovať na

¹Uvažujeme 3D tlačiareň s dvomi motorami pre horizontálny pohyb — jeden motor pre pohyb v smere osi X a druhý pre pohyb v smere osi Y.

koncovú rýchlosť. V tomto prípade počas prvej fázy pohybu extrudér zrýchľuje (spomaľuje), počas druhej spomaľuje (zrýchľuje).

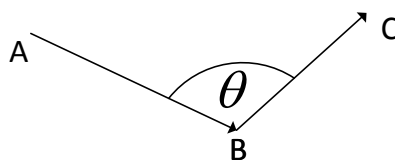


Obr. 3.1: Vzťah rýchlosti a zrýchlenia pri pohybe s lichobežníkovým profilom (hodnoty sú ilustračné)

Obr. 3.2 znázorňuje pohyb z bodu A do C cez bod B. Rýchlosť extrudéru v bode B medzi závisí na uhle, ktorý zvierajú jeho smerový vektor so smerovým vektorom nasledujúceho pohybu. Ak je tento uhol z intervalu $(\frac{\pi}{2}, \frac{3\pi}{2})$ pohyby môžu byť spojené — extrudér nemusí medzi pohybmi zastať. Rýchlosť extrudéru v bode B pri pohybe z bodu A do bodu B (a teda počiatočná rýchlosť pohybu z bodu B do C) je potom daná vzťahom:

$$v = \begin{cases} \vec{v}_{max} |\cos \theta| & \text{pre } \theta \in (0, \frac{\pi}{2}) \cup (\frac{3\pi}{2}, 0) \\ 0 & \text{inak} \end{cases} \quad [9]$$

kde \vec{v}_{max} je maximálna rýchlosť dosiahnutá počas pohybu z A do B a θ je uhol medzi vektormi \vec{AB} a \vec{BC} . Praktický aspekt tohto vzťahu je spomalenie pri zmene smeru pohybu. Jedná sa o jednoduché riešenie pomerne zložitého problému, komplexnejšie riešenie je možné nájsť v literatúre [13]. V prípade, že uhol θ je z intervalu $(\frac{3\pi}{2}, \frac{\pi}{2})$, musí extrudér v bode B zastaviť a teda rýchlosť v tomto bode je nulová.



Obr. 3.2: Pohyb extrudéru z bodu A do B a následne do C

Presný výpočet času jednotlivých pohybov by vyžadoval zohľadnenie zotrvačnosti extrudéru, s ktorou v rámci tejto práce nepočítam, pretože by to vyžadovalo znalosť hmotnosti

extrudéru, čím by vznikla závislosť na konkrétnom modeli 3D tlačiarne. Navyše, potenciálny užívateľ simulátoru by musel túto hmotnosť poznať a zadať ju do simulátoru, čo by komplikovalo použitie výsledneho programu. Základné vzťahy použité pre výpočet času sú:

$$s = \vec{v}_0 t + \frac{1}{2} \vec{a} t^2 \quad (3.2)$$

$$\vec{v} = \vec{v}_0 + \vec{a} t \quad (3.3)$$

kde s je celková dráha pohybu, \vec{v}_0 je počiatočná rýchlosť, \vec{a} je konštantné zrýchlenie v smere pohybu a t je doba trvania pohybu [7].

Vyjadrením t z rovnice 3.3:

$$t = \frac{\vec{v} - \vec{v}_0}{\vec{a}} \quad (3.4)$$

a dosadením do rovnice 3.2 po úprave dostávame:

$$s = \frac{\vec{v}^2 - \vec{v}_0^2}{2\vec{a}} \quad (3.5)$$

kde s je dráha, ktorú extrudér prejde pri zrýchľovaní z rýchlosti \vec{v}_0 na rýchlosť \vec{v} .

V prípade lichobežníkového profilu pre celkový čas t platí:

$$t_1 = \sqrt{\frac{2s_1}{\vec{a}}} \quad (3.6)$$

$$t_2 = \frac{s_2}{\vec{v}} \quad (3.7)$$

$$t_3 = \sqrt{\frac{2s_3}{\vec{a}}} \quad (3.8)$$

$$t = t_1 + t_2 + t_3 \quad (3.9)$$

kde s_1 je dráha prejdená pri zrýchľovaní (spomaľovaní) na požadovanú rýchlosť za čas t_1 , s_2 je dráha prejdená rovnomerným pohybom za čas t_2 a s_3 je dráha prejdená pri zrýchľovaní (spomaľovaní) na koncovú rýchlosť za čas t_3 .

V prípade trojuholníkového profilu je potrebné najprv vypočítať dráhu d_1 , ktorá odpovedá prvej fáze:

$$d_1 = \frac{1}{2} \left(\frac{\vec{v}^2 - \vec{v}_0^2}{2\vec{a} + s} \right) \quad (3.10)$$

kde s je celková dráha pohybu. Je zrejmé, že dráha druhej fázy pohybu je $d_2 = s - d_1$. Celkový čas pohybu je daný súčtom času t_1 , za ktorý extrudér prejde dráhu d_1 a t_2 , za ktorý extrudér prejde dráhu d_2 . Počas prvej fázy pohybu extrudér dosiahne rýchlosť \vec{v}_{max} pre ktorú platí:

$$\vec{v}_{max} = \vec{v}_0 + t_1 \vec{a} \quad (3.11)$$

Časy t_1 a t_2 je možné vypočítať z rovnice 3.2. Keďže sa jedná o kvadratickú rovnicu a počítame čas, vezmeme kladný koreň:

$$t = \frac{-\vec{v}_0 + \sqrt{\vec{v}_0^2 + 8\vec{a}s}}{2\vec{a}} \quad (3.12)$$

Pre výpočet času t_1 do tejto rovnice za \vec{v}_0 dosadíme počiatočnú rýchlosť pohybu a za s dráhu d_1 vypočítanú vzťahom 3.10, pre čas t_2 dosadíme za \vec{v}_0 hodnotu \vec{v}_{max} a za s celkovú dráhu pohybu skrátenu o d_1 .

V určitých situáciach sa môže stať, že nie je možné na danej vzdialenosti dosiahnuť koncovú rýchlosť. V tomto prípade sa jedná o špeciálny prípad trojuholníkového profilu, pričom dráha druhej fázy je nulová. Prakticky teda extrudér buď len zrýchľuje, alebo len spomaľuje. Pre celkový čas pohybu potom platí rovnica 3.12.

Na základe uvedených vzťahov je možné do určitej miery presne simulovať pohyb extrudéru 3D tlačiarne. Znalosť doby trvania jednotlivých pohybov je hlavným predpokladom pre vytvorenie realistickej animácie pohybu extrudéru.

3.3 Návrh vizualizácie

Súčasťou realistickej vizualizácie procesu tlače metódou FDM (2.3) je okrem presného modelu pohybu extrudéru aj vizualizácia správania materiálu — najmä topenia a tuhnutia v závislosti na teplote. Počítať v reálnom čase odpovedajúci matematický model by pri súčasnom výkone počítačov určite problém nebol. Po grafickej stránke, kde sa počíta s použitím knižnice Qt, sa však jedná o príliš zložitú úlohu. Navrhovaná vizualizácia preto správanie materiálu nezahŕňa, avšak aby bolo možné naplniť základnú myšlienku vizualizácie 3D tlače — postupné vytváranie modelu na scéne — je potrebné pribúdanie materiálu nejakým spôsobom zohľadniť.

Informácia o množstve použitého materiálu je súčasťou E parametra inštrukcie G1. V prípade, že tento parameter chýba, extrudér sa na dané miesto presunie, ale cestou nepoužije žiadny materiál. Rozlišujeme teda akési dva režimy — režim tlače a režim presunu. Ak chceme získať správnu podobu výsledného objektu, je potrebné s touto informáciou korektne pracovať.

Základná myšlienka navrhovanej vizualizácie je pomerne jednoduchá — každý bod, cez ktorý extrudér prejde v režime tlače sa uloží do výslednej datovej štruktúry vznikajúceho modelu. Pri zobrazení sa tieto body vykreslia spojené čiarami. Pri zobrazení s každým pohybom extrudéru vznikne dojem, že extrudér za sebou zanecháva stopu — materiál.

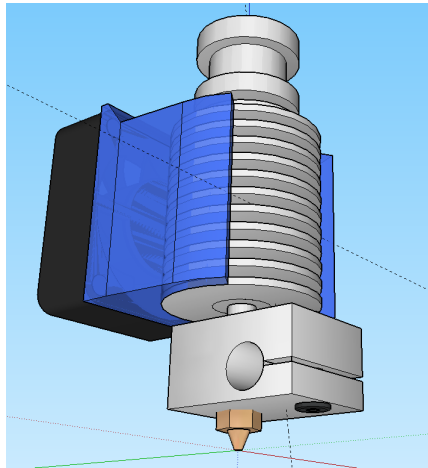
Hlavnou nevýhodou tohto prístupu je interná reprezentácia modelu vo forme mračna bodov (viz obr. 2.5). Vykreslenie čiar medzi jednotlivými bodmi síce poskytne dobrú predstavu o tom, ako model vyzerá, ale nemáme možnosť počítať na danom modeli žiadny osvetľovací model, čo značným spôsobom ubera vizualizácii na realistikosti.

3.3.1 Vizualizácia pohybu extrudéru

Model extrudéru (viz obr. 3.3) sa zobrazí vo svojej polohe, pri spustení vizualizácie je to bod $[0,0,0]$. Na základe vzťahov uvedených v časti 3.2.1 sa pri vykonaní inštrukcie G1 (pohyb) vypočíta čas, za ktorý sa extrudér dostane do svojej cieľovej pozície. Presun je vizualizovaný animáciou pohybu modelu extrudéru z jeho aktuálnej pozície po lineárnej trajektorii do cieľovej polohy. Doba trvania tejto animácie odpovedá vypočítanému času a je závislá na nastavenej rýchlosti simulácie.

3.4 Grafické užívateľské rozhranie

Grafické užívateľské rozhranie (obr. 3.4) bolo navrhnuté s ohľadom na požiadavky definované v časti 3.1. Z veľkej časti ho tvorí plocha, na ktorej je zobrazená grafická scéna



Obr. 3.3: Použitý model extrudéru (Prevzatý z [10])

slúžiaca na vizualizáciu tlače. Práva časť okna je určená na zobrazovanie práve vykonávaných inštrukcií. V spodnej časti programu sú umiestnené tlačidlá pre ovládanie simulácie. Pri spustení sú aktívne len tlačidlá na vloženie modelu a vyčistenie scény. Po načítaní modelu je možné spustiť simuláciu.

Po spustení simulácie program v pravej spodnej časti okna zobrazí celkový odhadovaný skutočný čas tlače daného modelu. Pri spustení programu je nastavená rýchlosť simulácie na $1\times$, čo predstavuje simuláciu v reálnom čase. Simuláciu je možné zrýchliť $10\times$, $100\times$ alebo na maximálnu možnú rýchlosť, pri ktorej sú animácie pohybu extrudéru vypnuté a celá simulácia trvá rádovo jednotky sekúnd. Pri tejto rýchlosti je vypisovanie vykonávaných inštrukcií zakázané, pretože typicky sa vykonáva viac ako 10000 inštrukcií za sekundu.

Simuláciu je možné kedykoľvek pozastaviť (Pause) a znovu spustiť (Resume), prípadne úplne zastaviť (Stop). O priebehu každej akcie, ktorú program vykonáva je užívateľ informovaný prostredníctvom progres-indikátoru v pravej dolnej časti. V prípade spustenej simulácie je v tejto časti zobrazovaný jej priebeh.

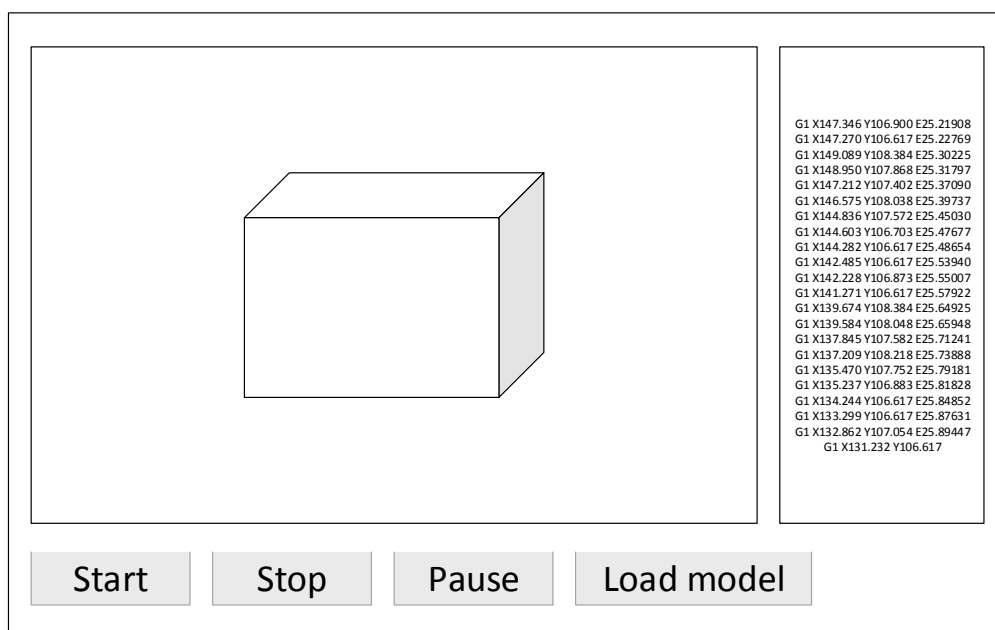
Pre lepšiu orientáciu v priestore program zobrazuje jednotlivé súradnicové osi. Os X červenou, os Y zelenou a os Z modrou farbou. Túto funkčnosť je možné vypnúť v ľavej dolnej časti okna, rovnako je možné vypnúť zobrazovanie vykonávaných inštrukcií.

3.5 Objektový návrh simulátoru

Cieľom objektového návrhu je identifikovať v rámci problému triedy, ktorým bude rozdelená zodpovednosť za jednotlivé časti simulátoru. Jednou z úloh návrhu je oddeliť model 3D tlačiarne od zvyšku programu, aby mohol byť v budúcnosti nahradený iným, presnejším modelom alebo modelom inej 3D tlačiarne, ktorá bude mať iný koncept pohybu extrudéru.

3.5.1 Trieda Scanner

Trieda Scanner je jediná trieda ktorá má prístup priamo k súboru s inštrukciami a má za úlohu jeho prvotné spracovanie — lexikálnu analýzu. Rozhranie triedy tvorí metóda `get_token()`, ktorá vracia nasledujúcu inštrukciu.



Obr. 3.4: Základné rozloženie prvkov v grafickom užívateľskom rozhraní

3.5.2 Trieda Model

Ako názov napovedá, trieda Model zapúzdruje model 3D tlačiarne a teda definuje akcie, ktoré odpovedajú jednotlivým inštrukciám. Aby mohol model správne reagovať na prichádzajúce správy, je potrebné aby si vnútorne uchovával stav. Stav modelu je definovaný aktuálnou pozíciou a rýchlosťou extrudéru, požadovanou rýchlosťou pre aktuálny pohyb a informáciou o tom, koľko výplne (materiálu) bolo doposiaľ použitej na vykonané pohyby. Je zrejmé, že stav modelu sa mení s každou inštrukciou. Rozhranie modelu pre pohyb extrudéru v horizontálnej rovine tvorí metóda `move()`, v ktorej model dostáva ako parameter pozíciu, do ktorej sa má extruder presunúť a uhol, ktorý zvierá smerový vektor tohto pohybu so smerovým vektorom nasledujúceho pohybu. Metóda `move()` implementuje správanie definované v časti 3.2.1. Výsledkom vykonania tejto metódy (návratovou hodnotou) je dobrá trvania pohybu v sekundách a odpovedajúca zmena vnútorného stavu modelu. Rozhranie modelu pre vertikálny pohyb je metóda `lift_head()`, ktorá extrudér presunie na ďalšiu vrstvu. Poslednou metódou, ktorá umožňuje meniť stav modelu zvonku, je metóda `extrude()`, ktorá simuluje vytlačanie materiálu. Trieda model môže byť v prípade potreby nahradená inou triedou, ktorá dodržiava toto rozhranie.

3.5.3 Trieda LookAhead

V zodpovednosti tejto triedy je správa pohybov. V rámci inicializácie simulácie, po spracovaní G-Code súboru, dostane objekt triedy `lookAhead` k dispozícii zoznam všetkých inštrukcií, ktoré budú v rámci simulácie vykonané. Ako bolo uvedené v časti 3.2.1, koncová rýchlosť pohybu je závislá na jeho uhle s nasledujúcim pohybom, ako to znázorňuje obr. 3.2. Aby bolo pri vykonávaní pohybu z bodu A do B známe, aký je uhol s nasledujúcim pohybom, je potrebné toto dopredu predpočítať. Úlohou objektu triedy `lookAhead` je teda

vytvoriť zoznam pohybov a pridať ku každému pohybu informáciu o tom, akým smerom sa bude extrudér hýbať pri nasledujúcom pohybe. Zároveň si tento objekt pamätá, ktorý pohyb nasleduje ako ďalší. Rozhranie triedy tvoria metódy `do_lookAhead()`, ktorej vstupom je zoznam inštrukcií a `next_move()`, ktorá vracia pohyb, ktorý nasleduje ako ďalší.

3.5.4 Trieda Controller

Trieda Controller zapúzdruje aplikačnú logiku celého programu. Prostredníctvom volania funkcie `simulation_start()` je možné spustiť simuláciu, pričom stav simulácie je udržiavaný práve v triede Controller, ktorá je zároveň jedinou triedou komunikujúcou s modelom 3D tlačiarne (pozri obr. 3.5). Súčasťou rozhrania tejto triedy sú aj funkcie `simulation_stop()`, `simulation_pause()` a `simulation_resume()`, ktorých názvy dostatočne napovedajú o svojom účele.

3.5.5 Trieda Interface

Aby Controller nebol pevne spojený s grafickým užívateľským rozhraním (ďalej len GUI), návrh zahŕňa triedu Interface, ktorá tvorí rozhranie medzi GUI a zvyškom programu. Akcie, ktoré užívateľ v GUI vykoná sú delegované objektu triedy Interface a ten ich predáva ďalej — objektu triedy Controller. Tok správ funguje aj opačne — o udalostiach, ktoré sa dejú v simulátore je GUI informované prostredníctvom objektu triedy Interface.

3.5.6 Trieda View

Trieda View spravuje grafickú scénu a jej obsah. K jej zodpovednosti patrí zobrazovanie extúderu a animovanie jeho pohybu a zobrazovanie vznikajúceho modelu. Práve táto trieda prijíma a spracováva informácie, ktoré poskytuje Interface a na základe nich zobrazuje stav modelu. Väčšina metód tejto triedy pracuje na úrovni OpenGL.

3.5.7 Trieda MainWindow

MainWindow je trieda oddedená od triedy QWidget, ktorú poskytuje knižnica Qt. Jej úlohou je správa GUI — obsluha akcií, ktoré vykoná užívateľ a ich propagovanie do triedy Controller prostredníctvom triedy Interface.

3.5.8 Trieda OptionsWindow

Táto trieda definuje vzhľad a správanie okna s nastaveniami simulátora. Zmeny v nastaveniach vykonané v okne triedy OptionsWindow budú prostredníctvom triedy Interface propagované triede Controller v rámci metódy `on_submit()`. Metóda `on_cancel()` zruší vykonané zmeny a zatvorí okno s nastaveniami.

3.5.9 Trieda Semaphore

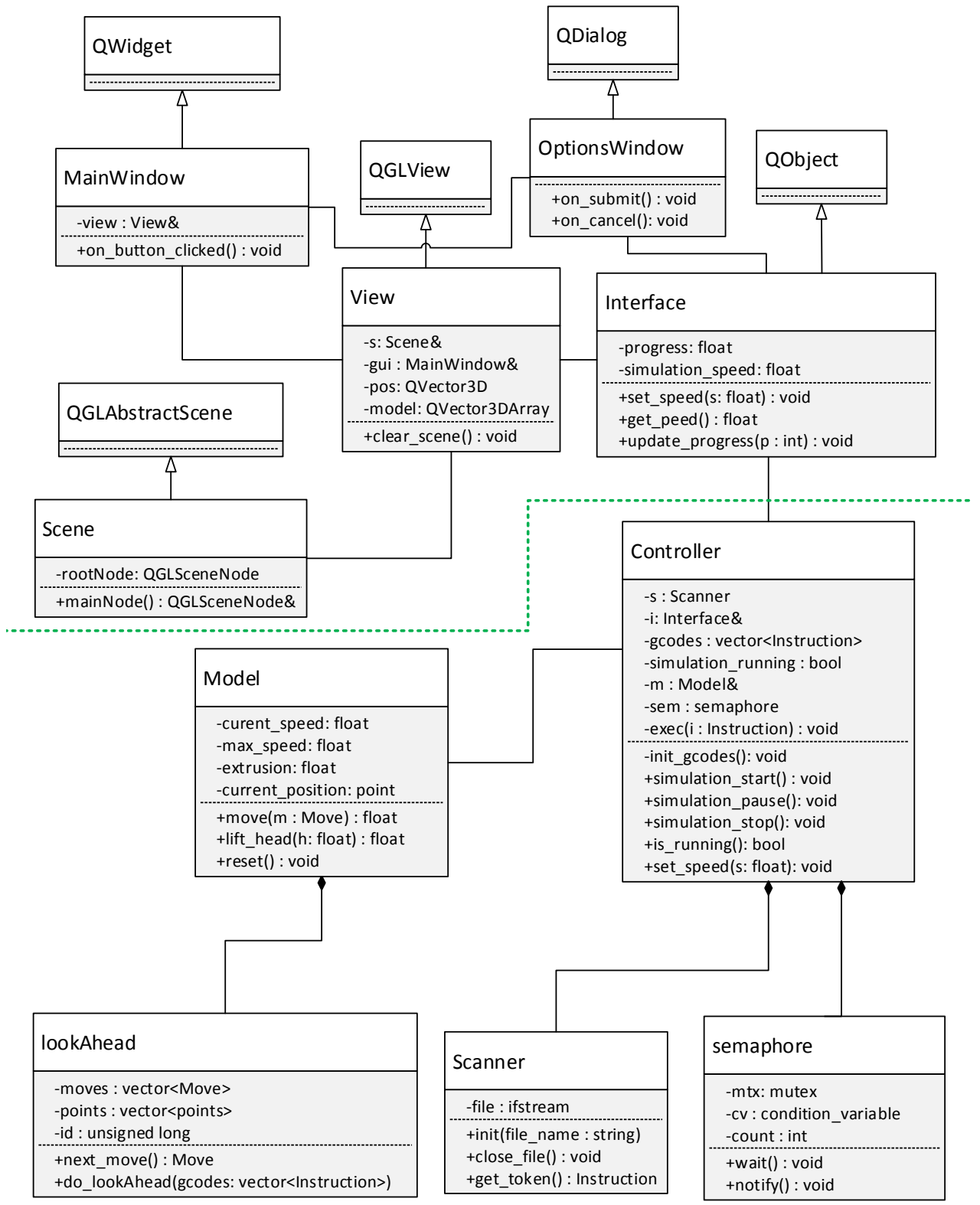
Úlohou tejto triedy je synchronizácia procesu simulácie s procesom obsluhujúcim užívateľské rozhranie.

3.5.10 Slicer

Napriek tomu, že nejde priamo o objekt alebo triedu, zmieňujem slicer na záver objektového návrhu, pretože sa jedná o autonómnu časť simulátoru s presne vymedzenou zodpovednosťou. Ako napovedá názov, úlohou sliceru je slicing — teda generovanie G-Code z vstupného modelu. Slicer je jediná časť simulátoru, ktorú neimplementujem vo vlastnej réžií. O slicing sa stará externý opensource program Slic3r², ktorý však z pohľadu GUI funguje úplne transparentne.

Spojením vyššie uvedených tried do jedného celku získavame funkčný simulátor, ktorému sa z pohľadu implementácie a overovania validity venuje nasledujúca kapitola. Triedy `QWidget`, `QGLAbstractScene`, `QGLView`, `QDialog` a `QObject` znázornené v diagrame 3.5 sú súčasťou knižnice Qt. Pre jednoduchosť boli atribúty týchto tried vynechané, ich presný popis je možné nájsť v dokumentácii knižnice Qt [6]. Čiarkovaná čiara v diagrame tried znázorňuje oddelenie grafického užívateľského rozhrania od aplikačnej logiky simulátoru.

²<http://slic3r.org>



Obr. 3.5: UML diagram tried simulátoru

Kapitola 4

Implementácia simulátoru a overenie jeho funkčnosti

Prvá časť tejto kapitoly je venovaná popisu použitých technológií a vybraných aspektov implementácie simulátoru. Druhá časť kapitoly obsahuje popis troch experimentov, ktorých úlohou bolo overiť správnu funkčnosť simulátoru.

4.1 Implementačné prostredie

Program bol implementovaný v jazyku C++ s využitím štandardu C++11. Pri vývoji bol použitý prekladač GCC verzie 4.9.2. Grafické užívateľské rozhranie vrátane vizualizácie procesu tlače a pohybu extrudéru bolo implementované s využitím knižnice Qt verzie 5.3. Zdrojové kódy sú dokumentované v štýle kompatibilnom s programom doxygen. Program bol vyvíjaný pod operačným systémom Linux Mint 17.1, pri vývoji však boli dodržované všetky zásady multiplatformového programovania, program by teda mal fungovať v zásade pod akýmkoľvek operačným systémom. Funkčnosť bola úspešne testovaná aj na systéme Windows 7. Potenciálnym zdrojom nekompatibility môže byť slicer, ktorý simulátor využíva na generovanie G-Code. Program Slic3r¹ je v súčasnosti dostupný pre platformy Linux, Windows a Mac OS. Verzia pre Mac OS nebola v spojení so simulátorom testovaná.

4.2 Implementácia

Implementácia vychádza z návrhu uvedeného v sekcii 3.5, riešenie má teda čisto objektový charakter. Riadenie toku programu je realizované predávaním správ medzi objektami. Spusteniu simulácie predchádza načítanie 3D modelu, ktorý chceme vytlačiť. O načítaní modelu je informovaný objekt triedy Controller (ďalej len Controller), ktorý v reakcii na túto správu spustí slicer (3.5.10), ktorý z daného modelu vygeneruje G-Code. G-Code je pri spustení simulácie spracovaný Scannerom a všetky inštrukcie sú načítané do pamäti. Následne Controller metódou `do_lookAhead()` (3.5.3) spustí analýzu inštrukcií. Po tejto fáze je simulácia pripravená na beh a v rámci metódy `simulation_run()` sa spustí algoritmus riadenia simulácie, ktorý postupne vykonáva jednotlivé inštrukcie. Pred skutočným spustením simulácie Controller najprv spustí simuláciu v režime emulácie, v ktorom celá simulácia v rýchlosti prebehne s cieľom zistiť, ako dlho bude trvať vytlačenie daného modelu. Táto informácia je potom pri spustení simulácie už zobrazená v grafickom užívateľskom rozhraní.

¹<http://slic3r.org>

Architektúra aplikácie je dvojláková – jedno vlákno obsluhuje udalosti, ktoré generuje užívateľ prostredníctvom GUI a tieto udalosti propaguje ďalej prostredníctvom objektu triedy Interface. Toto vlákno zároveň obsluhuje grafickú scénu, spúšťa animácie a obsluhuje správy, ktoré prichádzajú od Controlleru. Druhé vlákno má na starosti beh simulácie, najmä výpočty časov jednotlivých pohybov extrudéru, ktoré prebiehajú až za behu simulácie.

Výnimku z tohto pravidla tvorí generovanie G-Code, ktoré prebieha ako samostatný proces, keďže je realizované samostatnou aplikáciou. Jedná sa o pomerne dlhotrvajúcu akciu (typicky trvá 5 - 30 sekúnd, v závislosti na veľkosti modelu a výkone procesora). O priebehu generovania G-Code GUI informuje prostredníctvom progress-baru, ktorý znázorňuje percentuálny priebeh operácie. Keďže Slic3r (3.5.10) neposkytuje žiadne informácie o percentuálnom priebehu generovania kódu, musel som vymyslieť spôsob, ako odhadnúť trvanie generovania kódu pre daný 3D model. Implementovaná metóda dopredu odhadne čas trvania danej operácie na základe veľkosti súboru 3D modelu a postupne aktualizuje daný progress-bar v závislosti na uplynutom čase od začiatku operácie. Bohužiaľ, závislosť medzi veľkosťou súboru a dobou trvania generovania G-Code nie je lineárna, takže táto metóda nefunguje úplne presne.

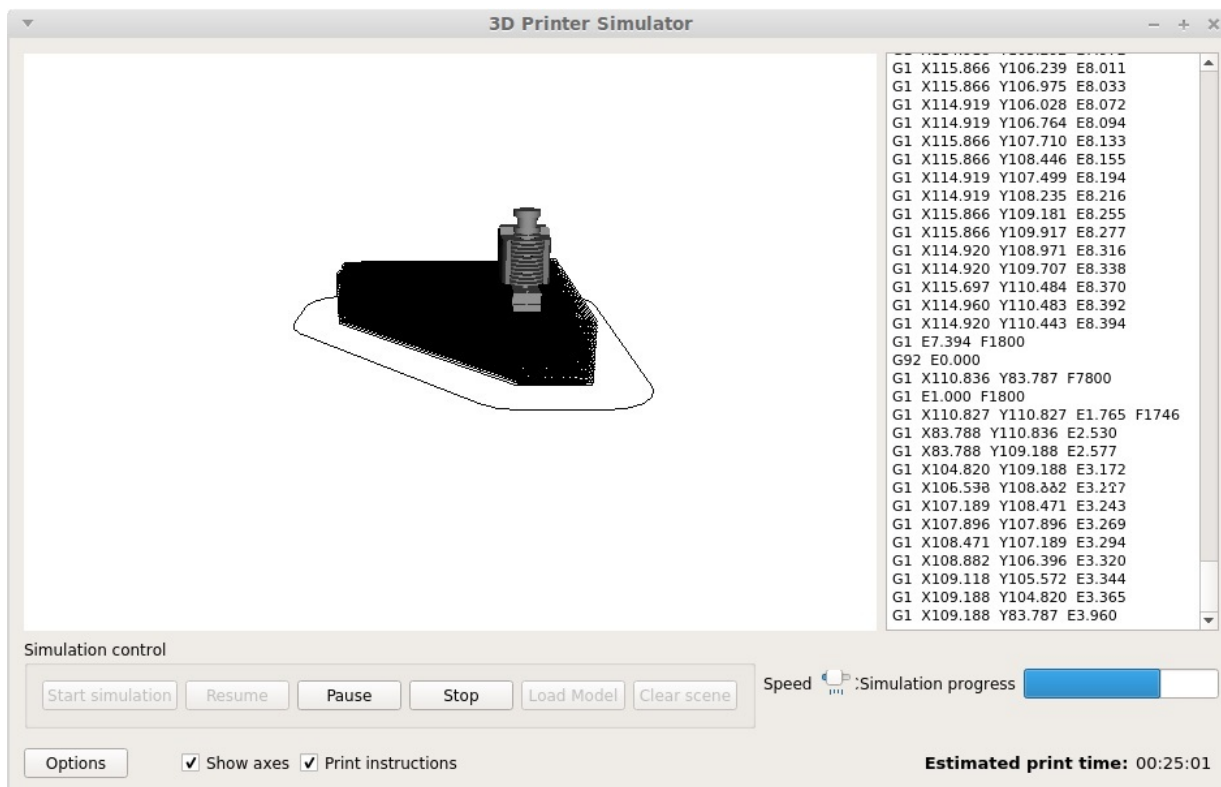
Implementácia obsahuje oproti návrhu (3.5) navyše triedu Logger, ktorá zaznamenáva činnosť simulátoru do súboru. Meno výstupného súboru je možné pri preklade nastaviť v súbore `configuration.h`. Trieda je implementovaná pomocou návrhového vzoru Singleton. Na činnosť simulátoru nemá Logger žiadny vplyv, avšak počas vývoja predstavoval značné zjednodušenie hľadania chýb v programe. Príklad výstupu triedy Logger:

```
12:55:17 Executing slicer
12:55:22 Slicer returned 0
12:55:29 Initializing gcodes...
12:55:29 Initializing Scanner...
12:55:29 Scanner initialized.
12:55:29 Total gcodes to proceed: 211883
12:55:29 GCodes initialized.
12:55:29 Executing lookahead.
12:55:30 Lookahead done.
12:55:30 Starting simulation...
```

Pre prípad budúceho rozširovania alebo upravovania programu bola táto trieda v implementácii ponechaná.

4.2.1 Nastavenia programu

Pre software v oblasti 3D tlačiarň je typická vysoká prispôsobivosť konkrétnemu modelu 3D tlačiarne prostredníctvom širokej škály nastavení. Implementovaný simulátor je možné do určitej miery konfigurovať po kliknutí na tlačidlo options, ktoré otvorí jednoduché okno s nastaveniami (obr. 4.2). Tu je možné definovať argumenty, ktoré budú predané sliceru pri jeho spustení a je pomocou nich možné špecifikovať veľké množstvo parametrov tlače. Kompletný zoznam podporovaných argumentov je možné získať spustením programu `slic3r` s parametrom `--help`.



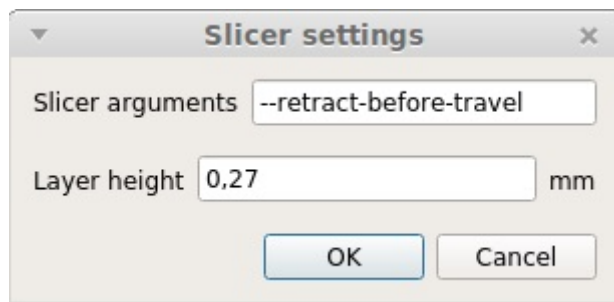
Obr. 4.1: Vzhľad výslednej aplikácie

4.2.2 Preklad a spustenie programu

Vizualizácia procesu tlače využíva modul Qt3D, ktorý poskytuje API pre prácu s OpenGL. Tento modul však v aktuálnej verzii knižnice Qt (5.4) chýba, jeho podpora je ale plánovaná už v ďalšom vydaní Qt. K prekladu programu je teda potrebné mať nainštalovanú verziu Qt 5.3, prípadne inú, ktorá obsahuje modul Qt3D. K programu je priložený makefile, takže pre preklad stačí spustiť príkaz `make`. Príkaz `make run` preloží a spustí program.

Príkazom `make doxygen` je možné automaticky vygenerovať dokumentáciu k projektu vo formáte `html`. Pre korektnú funkčnosť programu je potrebné v hlavičkovom súbore `configuration.h` nastaviť cestu k sliceru. Program bol testovaný so slicerom verzie 1.0.1, táto verzia je aj súčasťou odovzdaných súborov, ktorých popis sa nachádza v prílohe A. Zdrojové kódy pozostávajú z 22 súborov, ktoré celkovo obsahujú 2819 riadkov kódu vrátane komentárov.

Simulátor je teda plne funkčný a autorovi nie sú známe žiadne chyby, ktoré by viedli k nesprávnej funkčnosti alebo pádu programu. Počas implementácie sa vyskytli problémy pri práci s knižnicou Point Cloud Library (ďalej len PCL), ktorá slúži na prácu s mračnami bodov. V simulátore by bolo vhodné použiť túto knižnicu na konverziu mračna bodov, ktorým je interne reprezentovaný model, na polygonálny model. Túto funkčnosť sa mi bohužiaľ nepodarilo docieľiť, nakoľko po preložení a spustení vzorovej implementácie rozpoznávania modelu, ktorá je k dispozícii priamo na stánkach knižnice, program po dobu približne 30 sekúnd vyťažoval procesor na 100% a následne skončil s chybou `Segmentation fault`.



Obr. 4.2: Okno s nastaveniami sliceru

Príčinu chyby sa mi vzhľadom na zložitosť tejto knižnice nepodarilo odhaliť.

4.3 Časová a pamäťová zložitosť simulácie

Algoritmus riadenia simulácie sekvenčne spracováva jednotlivé GCode inštrukcie. Zložitosť tohto algoritmu je teda lineárna v závislosti na počte inštrukcií. V prípade interpretácie inštrukcie G1 sa počas behu simulácie počíta čas potrebný na vykonanie daného pohybu, ktorého výpočet zahŕňa niekoľko operácií s desatinnými číslami. Uhly medzi po sebe idúcimi pohybmi sa predpočítavajú v rámci inicializácie. Principiálne je možné všetky výpočty vykonať dopredu, ale keďže počas behu simulácie animácia jedného pohybu extrudéru trvá rádovo desiatky milisekúnd, na potrebné výpočty má program dostatok času. Výnimkou je stav, kedy je nastavená maximálna rýchlosť simulácie a animácie sú vypnuté. V tomto prípade by zrejme malo predvypočítanie časov pohybov na celkovú dobu simulácie vplyv, avšak pri tejto rýchlosti simulácia trvá rádovo jednotky sekúnd, čo považujem za dostatočujúce. Napríklad simulácia tlače modelu uvedeného v časti 4.5.2 trvá približne 10 sekúnd, pričom reálne by výroba tohto modelu mala trvať až 24 hodín.

Pamäťová zložitosť simulátoru je rovnako závislá na počte inštrukcií. Každá inštrukcia je reprezentovaná štruktúrou `Instruction`:

```
struct {  
    instruction_type type; // typ inštrukcie  
    float x, y, z; // súradnice novej polohy extrudéru  
    float f; // feedrate — rýchlosť pohybu extrudéru  
    float S; // parameter S  
    float E; // parameter E  
} Instruction;
```

Parameter S sa využíva len pri niektorých inštrukciách. Príkladom je inštrukcia M109, ktorá zohrieva extrudér na požadovanú teplotu. Zápis M109 S215 značí zohriatie extrudéru na 215°C. Hodnotou parametru S je v tomto prípade 215. Parameter E uchováva dĺžku materiálu, ktorý sa má pri tomto pohybe použiť.

Po inicializácii simulácie vznikne pre každý pohyb (každú inštrukciu G1) v pamäti štruktúra `Move`:

```
struct {  
    point start; // začiatkový bod  
    point end; // koncový bod
```

```

float cos_alpha; // kosínus uhlu s nasledujúcim pohybom
} Move;

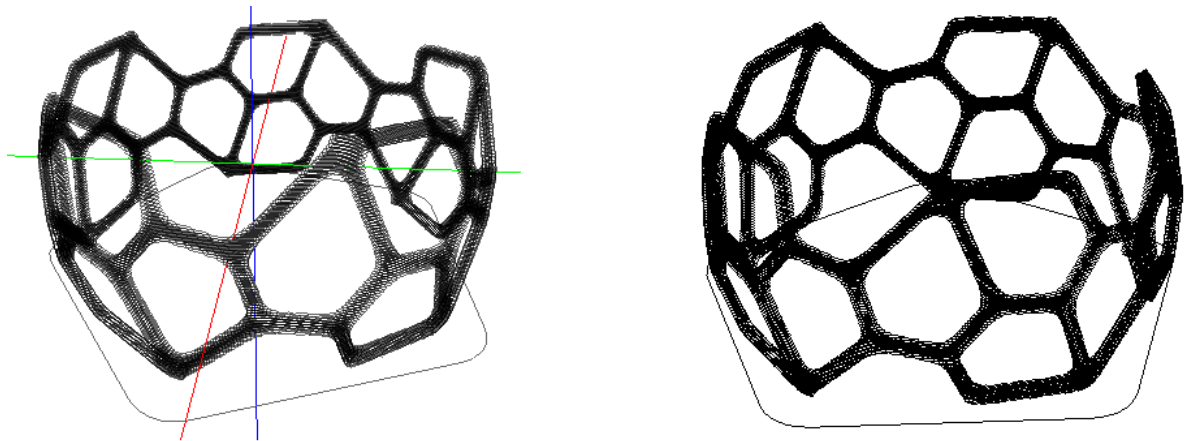
```

Vo vyššie uvedenom implementačnom prostredí je veľkosť štruktúry `Move` 40 Bajtov a veľkosť štruktúry `Instruction` 28 Bajtov. V prípade, že by GCode obsahoval 100000 inštrukcií G1, vyžada si ich uloženie približne 6,5 MB. V praxi však program využíva typicky okolo 170 MB pamäti, na čom má reprezentácia inštrukcii len malý podiel. Hlavný podiel ma grafické užívateľské rozhranie a vizualizačná časť implementovaná s využitím knižnice Qt.

4.4 Porovnanie s existujúcimi riešeniami

Z nástrojov uvedených v sekcii 2.7 sa mi podarilo spustiť `GCode Viewer`, `gCodeViewer` a `Willit 3D print`. Ani v jednom prípade sa bohužiaľ nejedná o kompletný simulátor, avšak z pohľadu tejto práce sú tieto nástroje zaujímavé kvôli vizualizácii 3D modelu a spracovaniu GCode všeobecne.

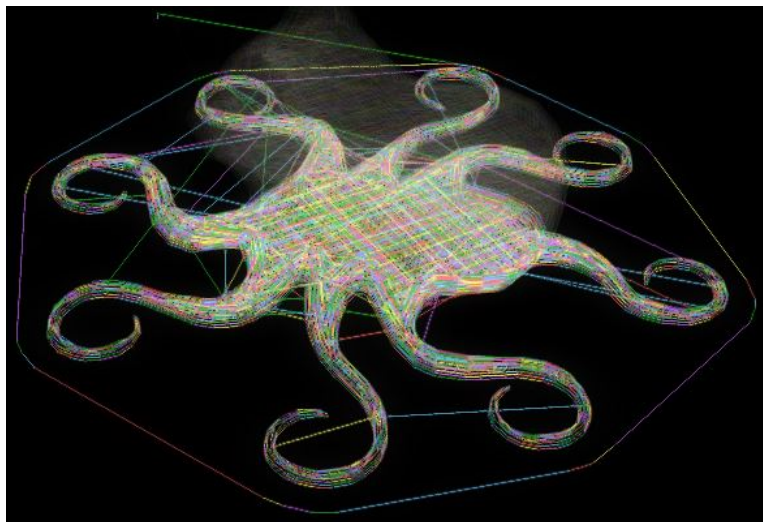
Na účely porovnania simulátoru s nástrojom `gCodeViewer` bol vybraný jednoduchý 3D model (pozri obr. 4.3). Program `gCodeViewer` používa zrejme rovnakú metódu vizualizácie ako implementovaný simulátor, pretože jednotlivé vizualizácie sú si nápadne podobné. Simulátor vytvorený v rámci tejto práce odhaduje dobu tlače tohto modelu na 43 minút, `gCodeViewer` na 39 minút. Po preskúmaní zdrojových kódov nástroja `gCodeViewer` som zistil, že pri počítaní času pohybu extrudéru tento program používa iba vzorec pre čas rovnomerného pohybu $t = s/v$. Je zrejme, že takýto výpočet je značne nepresný a pre zložitejšie modely by sa časy určite líšili významnejšie. V tomto prípade je teda čas vypočítaný simulátorom určite presnejší.



Obr. 4.3: Vizualizácia 3D modelu nástrojom `gCodeViewer` (vľavo) a ten istý model v simulátore (vpravo)

Druhý z uvedených nástrojov — `GCode Viewer` neposkytuje žiadne informácie o predpokladanej dobe tlače a slúži teda len na vizualizáciu GCode. Na obrázku 4.4 vidíme, ako vizualizácia v podaní tohto nástroja vyzerá. Na účely porovnania simulátoru s týmto prog-

ramom som stiahol rovnaký 3D model a nechal si ho odsimulovať. Výsledkom je zistenie, že GCode Viewer pri vizualizácii GCode nezohľadňuje presuny extrudéru bez použitia výplne, v dôsledku čoho vizualizácia obsahuje množstvo nepotrebných čiar, ktoré nie sú súčasťou modelu.



Obr. 4.4: Ukážka vizualizácie GCode pomocou nástroja GCode Viewer

Posledným skúmaným nástrojom je Willit 3D print. V tomto prípade nie sú zdrojové kódy programu dostupné, čo značne sťažuje celý proces porovnania. Principiálne poskytuje tento nástroj určitú podmnožinu funkcií simulátora — načítanie a spracovanie 3D modelu vo formáte STL, vizualizáciu 3D modelu a odhad celkovej doby tlače. Navyše program poskytuje aj určitý odhad ceny na základe množstva použitého materiálu. Samotná vizualizácia 3D modelu je veľmi kvalitná, mohli by sme jej vytknúť snáď len absenciu zoomu. Keďže zdrojové kódy nie sú dostupné, nepodarilo sa mi zistiť, akým spôsobom je táto vizualizácia realizovaná. Je možné, že vychádza priamo z daného STL súboru a nie z odpovedajúceho GCode.

Z porovnania simulátora s uvedenými nástrojmi vyplýva, že implementovaný simulátor využíva presnejšiu metódu vizualizácie ako GCode Viewer a presnejší model 3D tlačiarne ako gCodeViewer. V rámci tejto trojice je teda možné považovať implementovaný simulátor za najlepšie riešenie danej problematiky. Pokiaľ ide o nástroj Willit 3D print, tento nevizualizuje proces tlače, ale len hotový výsledok. Po tejto stránke by silnejším protikandidátom pre simulátor určite bol napríklad v časti 2.7 spomínaný CNC Simulator Pro. Bohužiaľ, tento nástroj nie je voľne dostupný a porovnanie preto nie je možné uskutočniť.

4.5 Overenie validity modelu 3D tlačiarne

Keďže som sa pri návrhu a implementácii simulátora zameriaval predovšetkým na správne modelovanie časovania pohybu extrudéru, aj pri skúmaní validity modelu 3D tlačiarne sa budem venovať práve tejto stránke simulátora.

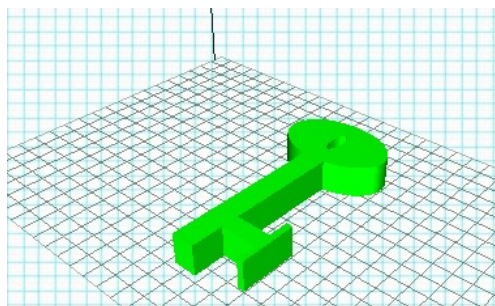
Model pohybu extrudéru je v porovnaní so skutočnosťou prirodzene značne zjednodušený. V praxi je pohyb typicky realizovaný prostredníctvom krokových motorov. Model krokového motora nie je súčasťou návrhu, čím vzniká prvá potenciálna nepresnosť modelu

3D tlačiarne. Za ďalší zdroj nepresnosti je možné považovať nezohľadnenie zotrvačnosti extrudéru pri počítaní dráhy prejdenej počas zrýchľovania a spomaľovania. Pri experimentoch sa pokúsim overiť, že dopad týchto zjednodušení na celkový výsledok simulácie nepresahuje rádovo jednotky percent.

Overiť platnosť získaných výsledkov bez reálnej 3D tlačiarne je pomerne náročná úloha, nakoľko nemáme k dispozícii žiadne skutočné údaje o dobe trvania tlačových úloh pre konkrétne modely. V rámci skúmania validity modelu tlačiarne bude pri prvých dvoch experimentoch hodnotená zhoda vypočítanej doby trvania tlače s referenčnou hodnotou, ktorú získam z online nástroja *Willit 3D print*. Tento nástroj bol vybraný na základe svojho kvalitného prevedenia v porovnaní s ostatnými dostupnými nástrojmi. Po vložení súboru s 3D modelom nástroj *Willit 3D print* poskytuje informácie o predpokladanej dobe tlače a zároveň zobrazuje 3D vizualizáciu daného modelu. Predpokladaná doba tlače bude porovnávaná s hodnotou, ktorú vypočíta implementovaný simulátor. Zároveň budú porovnané jednotlivé vizualizácie.

4.5.1 Experiment 1

V rámci prvého experimentu bol zvolený jednoduchý model kľúča (obr. 4.5). GCode vygenerovaný pre tento model obsahuje približne 47000 inštrukcií. Nástroj *Willit 3D print* poskytuje pre tento model odhad doby tlače 1 hodina a 42 minút, simulátor udáva hodnotu 1 hodina a 44 minút. Odchýlka oproti referenčnej hodnote je teda približne 1.9%, čo je akceptovateľná hodnota.



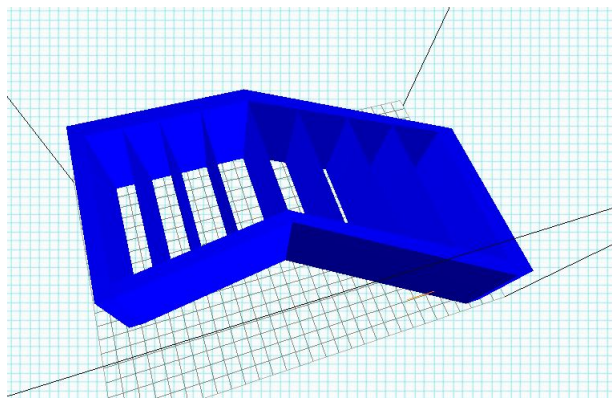
Obr. 4.5: 3D model použitý pri prvom experimente, vizualizácia nástrojom *Willit 3D print*

4.5.2 Experiment 2

Cieľom tohto experimentu je otestovať simulátor na 3D modeli, ktorému odpovedá veľké množstvo G-Code inštrukcií. Za týmto účelom bol vybraný model krájača na banány (obr. 4.6), vygenerovaný GCode obsahuje viac ako 1365000 inštrukcií. Nástroj *Willit 3D print* odhaduje dobu tlače tohto modelu na 24 hodín a 4 minúty. Simulátor pre tento model odhaduje dobu tlače na 23 hodín 39 minút, čo činí odchýlku približne 1.7%. Vzhľadom na veľkosť tohto 3D modelu je výsledok poskytnutý simulátorom možné považovať za zmysluplnú hodnotu.

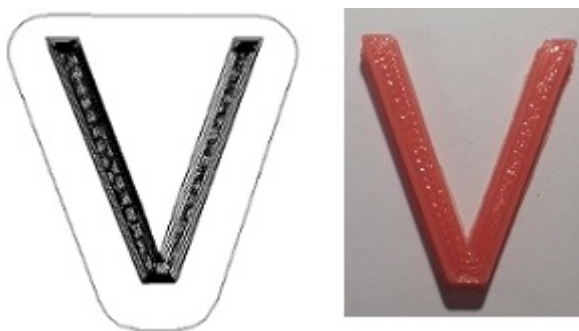
4.5.3 Experiment 3

Tento experiment bol vykonaný na reálnej 3D tlačiarňi *Easy 3D maker*, na ktorej bol tlačný jednoduchý model písmena V (obr. 4.7) o veľkosti 4 cm. V skutočnosti tlač trvala 12



Obr. 4.6: Model krájača na banány použitý pri druhom experimente

minút. Po korektnom nastavení sliceru v simulátore tak, aby generoval podkladovú vrstvu, odhadol program pre tento model dobu tlače na 12 minút 16 sekúnd. Odchýlka simulácie oproti skutočnosti je teda približne 2%.



Obr. 4.7: Model písmena V v simulátore (vľavo) a v skutočnosti

4.5.4 Zhrnutie experimentov

Napriek zjednodušeniam modelu 3D tlačiarne oproti realite, ktoré boli uvedené v úvode tejto sekcie, poskytuje simulátor uspokojivé výsledky. Otázka presnosti odhadov časov tlače jednotlivých 3D modelov zostáva svojim spôsobom nezodpovedaná, keďže nebolo vykonané dostatočné množstvo experimentov na skutočných 3D tlačiarňach. Okrem experimentov uvedených v tejto práci som simulátor testoval na väčšom množstve 3D modelov a v rámci zhrnutia uskutočnených experimentov uvádzam na záver ešte jedno dôležité pozorovanie.

Na výsledný odhad doby tlače má okrem modelu 3D tlačiarne zásadný vplyv nastavenie sliceru. V implementácii simulátoru slicer funguje v defaultnom nastavení, čo okrem iného znamená, že pri generovaní GCode predpokladá hrúbku jednej vrstvy 0.4 mm. Samotného modelu 3D tlačiarne sa to priamo netýka, ale na simuláciu má táto skutočnosť zásadný vplyv, pretože priamo ovplyvňuje množstvo vygenerovaných inštrukcií a teda aj celkový čas tlače. Validita výsledkov simulátoru je teda úzko zviazaná s nastavením sliceru. Potenciálny užívateľ simulátoru by sa preto mal určite v prvom rade zaujímať o nastavenie sliceru odpovedajúce jeho 3D tlačiarňi, ktoré je možné meniť, ako bolo uvedené v časti [4.2.1](#).

Na základe vykonaných experimentov samozrejme nie je možné prehlásiť model 3D tlačiarne za úplne validný, avšak neboli odhalené žiadne podstatné chyby, ani výrazné odchýlky od očakávaných výsledkov.

Kapitola 5

Záver

Cieľom práce bolo vytvoriť simulátor vybraného typu 3D tlačiarne. Pri práci som sa sústreďoval na tlačiarne typu RepRap, ale simulátor nakoniec nie je stavaný na jeden konkrétny model. Spoločné znaky 3D tlačiarní umožnili vytvoriť program, ktorý je schopný sa konkrétnemu modelu 3D tlačiarne prispôbiť najmä nastavením odpovedajúcich maximálnych zrýchlení extrudéru, ale aj nastavením slicera. Na prvý pohľad sa môže zdať, že škálovateľnosť riešenia bola do určitej miery dosiahnutá na úkor validity, keďže som sa pri práci sústreďoval hlavne na všeobecné princípy 3D tlačiarní a nie na špecifické znaky určitého modelu 3D tlačiarne. Pohyb extrudéru 3D tlačiarne je však riadený inštrukciami GCode, ktorého interpretácia je jednoznačne daná a všetky modely 3D tlačiarní ho teda musia interpretovať rovnako.

Pri experimentoch sa ukázalo, že simulátor funguje správne a poskytuje do značnej miery validné výsledky, čo bolo potvrdené viacerými experimentami (vrátane jedného praktického), ktorých výsledky sa od očakávaných neodlišovali o viac ako o jednotky percent. Simulátor však nie je úplne univerzálny a ak by sme chceli hovoriť o kompatibilitate s určitým modelom 3D tlačiarne, tak by tento model musel využívať metodu FDM a používať nezávislé motory pre pohyb v smere osí X, Y a Z. Pre čo najpresnejšiu simuláciu je potrebné nastaviť simulátor tak, aby počítal s hodnotami zrýchlení extrudéru, ktoré odpovedajú 3D tlačiarni, ktorú chceme simulovať.

Prínos práce spočíva mimo iné v open-source charakteri riešenia. Podobných simulátorov je navyše veľmi málo — pri riešení práce som nenarazil na žiadny iný simulátor 3D tlače napísaný v C++. Existujúce simulátory, s ktorými som sa pri tejto práci stretol, fungujú len v rámci prehliadača a výkonom sa samozrejme nemôžu rovnať programu napísanému v C++.

Vizualizácia 3D tlače je vyriešená pomerne jednoduchým spôsobom a v budúcnosti by bolo vhodné doplniť ju o rozpoznávanie objektu z mračna bodov, ktorým je vyrábaný model v simulátore reprezentovaný. To by umožnilo zobraziť model na scéne vrátane osvetľovacieho modelu, vďaka čomu by celá vizualizácia pôsobila realistickejšie. Ďalšou z možností pokračovania práce na simulátore je spresnenie vlastného modelu 3D tlačiarne, ktorý v súčasnosti využíva pomerne jednoduchú matematiku.

Literatúra

- [1] 3D Printing [online]. [cit. 2015-03-19]. Dostupné z: http://en.wikipedia.org/wiki/3D_printing.
- [2] G-code [online]. [cit. 2015-03-28]. Dostupné z: <http://reprap.org/wiki/G-code>.
- [3] ISO 6983: Automation systems and integration [online]. [cit. 2015-03-31]. Dostupné z: http://www.iso.org/iso/catalogue_detail.htm?csnumber=34608.
- [4] The Next Big Thing in 3-D Printing: Big Area Additive Manufacturing, or BAAM [online]. [cit. 2015-04-04]. Dostupné z: <http://www.fool.com/investing/general/2014/04/26/the-next-big-thing-in-3-d-printing-big-area-addi-2.aspx>.
- [5] Point Clouds. [online]. [cit. 2015-05-11]. Dostupné z: http://w3.impa.br/~boris/images1/otro_bun_portada.jpg.
- [6] Qt Documentation. [online]. [cit. 2015-04-19]. Dostupné z: <http://doc.qt.io/qt-5/classes.html>.
- [7] Real time stepper motor linear ramping just by addition and multiplication. [online]. [cit. 2015-04-21]. Dostupné z: <http://hwml.com/LeibRamp.pdf>.
- [8] RepRap Machines [online]. [cit. 2015-04-04]. Dostupné z: http://reprap.org/wiki/RepRap_Machines.
- [9] RepRapFirmware. [online]. [cit. 2015-04-05]. Dostupné z: <https://github.com/reprappro/RepRapFirmware>.
- [10] SketchUp 3D Warehouse - E3D V6 HotEnd. Dostupné z: <https://3dwarehouse.sketchup.com/model.html?id=u3d731367-dfd7-4c02-850a-af651da706cd>.
- [11] The world's first 3D Printer Simulator. [online]. 2012-12-18 [cit. 2015-04-06]. Dostupné z: <http://cn simulator.info/blog/?p=224>.
- [12] Boschetto, A.; Bottini, L.: Accuracy prediction in fused deposition modeling. *The International Journal of Advanced Manufacturing Technology*, ročník 73, 2014: s. 913–928.
- [13] Rymansaib, Z.; Iravani, P.; Sahinkaya, M.: Exponential Trajectory Generation for Point to Point Motions. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2013.

Dodatok A

Obsah CD

Na priloženom CD sa nachádzajú nasledujúce súbory a adresáre:

`slicer/`

`linux/` - slicer pre Linux

`windows/` - slicer pre Windows

`src` - zdrojové kódy programu

`Makefile` - Makefile pre preklad projektu a generovanie dokumentácie

`Doxyfile` - Konfiguračný subor pre Doxygen

`mainpage.dox` - Titulná strana dokumentácie

`bp-tex/` - zdrojové súbory k tomuto dokumentu

`xcillo00.tex`

`obsah.tex`

`prilohy.tex`

`bp_xcillo00.pdf` - Tento dokument