

Univerzita Karlova v Praze
Přírodovědecká fakulta
Katedra aplikované geoinformatiky a kartografie

Studijní program: Geografie

Studijní obor: Fyzická geografie a geoinformatika



Dominik Mazur

**Automatizace 3D modelování prostorových objektů s
vyžitím API vybraného software**

**Automation of spatial objects modeling in 3D using API of
selected modeling software**

Bakalářská práce

Vedoucí práce: Mgr. Lukáš Brůha, Ph. D.

Praha 2018

Vysoká škola: Univerzita Karlova v Praze

Fakulta: Přírodovědecká

Katedra: Aplikované geoinformatiky a kartografie

Školní rok: 2017/2018

Zadání bakalářské práce

pro Dominika Mazura

obor Fyzická geografie a geoinformatika

Název tématu:

Automatizace 3D modelování prostorových objektů s využitím API vybraného software

Hlavním cílem bakalářské práce je vytvoření generátoru budov, který umožní jednoduše a intuitivně generovat 3D modely budov a následně využít vygenerované modely v geoinformačních softwarech. Dílčí cíle práce jsou:

- zvýšení produktivity tvůrců 3D modelů za současného uchování kontroly nad generovanými výsledky,
- poskytnutí intuitivních prostředků 3D modelování pro uživatele bez speciální kvalifikace,
- tvorba takových 3D modelů, které budou využitelné v prostorových analýzách v geoinformačních softwarech.

Zásady pro vypracování

Rozsah grafických prací: cca 20 stran

Rozsah průvodní zprávy: cca 40 stran

Seznam odborné literatury:

MÜLLER, Pascal, Peter WONKA, Simon HAEGLER, Andreas ULMER a Luc VAN GOOL, 2006. Procedural modeling of buildings. *ACM Transactions on Graphics* [online]. **25**, 614–623. ISSN 07300301. Dostupné z: doi:10.1145/1179352.1141931

SMELIK, R. M., T. TUTENEL, K. J. DE KRAKER a R. BIDARRA, 2011. A declarative approach to procedural modeling of virtual worlds. *Computers and Graphics (Pergamon)* [online]. **35**(2), 352–363. ISSN 00978493. Dostupné z: doi:10.1016/j.cag.2010.11.011

SMELIK, Ruben M., Tim TUTENEL, Rafael BIDARRA a Bedrich BENES, 2014. A survey on procedural modelling for virtual worlds. *Computer Graphics Forum* [online]. **33**(6), 31–50. ISSN 14678659. Dostupné z: doi:10.1111/cgf.12276

WONKA, Peter, Michael WIMMER, François SILLION a William RIBARSKY, 2003. Instant architecture. *ACM Transactions on Graphics* [online]. **22**(3), 669. ISSN 07300301. Dostupné z: doi:10.1145/882262.882324

Vedoucí bakalářské práce: Mgr. Lukáš Brůha, Ph.D.

Konzultant bakalářské práce: -

Datum zadání bakalářské práce: 24. 01. 2018

Termín odevzdání bakalářské práce: 14. 5. 2018

Platnost tohoto zadání je po dobu jednoho akademického roku.

.....

Vedoucí bakalářské práce

.....

Vedoucí katedry

V Praze dne 9. 5. 2018

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

Jsem si vědom toho, že případné použití výsledků, získaných v této práci mimo Univerzitu Karlovu v Praze je možné pouze po písemném souhlasu této univerzity.

Svoluji k zapůjčení této práce pro studijní účely a souhlasím s tím, aby byla řádně vedena v evidenci vypůjčovatelů.

V Praze dne 9. května 2018

.....

Dominik Mazur

Abstrakt

Vzhledem k námaze, kterou vyžaduje manuální tvorba virtuálních světů, je příležitost vytvářet virtuální světy automaticky příliš lákavá na to, aby byla nevyužita.

Tato bakalářská práce představuje řešení, které automatizuje generování 3D modelů budov. Metodický základ je postaven na deklarativním přístupu k modelování a inspirován procedurálními modelovacími technikami. Prezentovaná práce stanovuje sadu pravidel a parametrů, pomocí kterých je navržen rychlejší a intuitivnější způsob navrhování modelů budov, zatímco je zachována flexibilita a kontrola, kterou uživatel má nad procesem generování.

Řešení je implementováno jako doplněk pro Blender s názvem „Generátor budov“. Doplněk je psaný v programovacím jazyce Python, který používá rozhraní API open source softwaru Blender a je k dispozici pod všeobecnou veřejnou licencí GNU.

Klíčová slova: Procedurální modelování, Blender, 3D budovy, automatizace modelování, CityEngine

Abstract

Considering the drudgery related to manual creation of virtual worlds, the opportunity of producing them automatically is too alluring to be dismissed.

This bachelor thesis introduces a solution that automates the generation of building's 3D model. The methodological foundation is based on declarative modeling approach and inspired by procedural modeling techniques. Presented work defines the set of rules and parameters. By these means, a faster and more intuitive way to design buildings' models, while preserving the flexibility and control of the user over the construction process, is proposed.

The solution is implemented as an add-on for Blender entitled „Buildings generator“. The Python scripting, which the add-on is based on, uses the API of open-source software Blender and is available under GNU GPL licence.

Keywords: Procedural modeling, Blender, 3D buildings, modeling automation, CityEngine

OBSAH

Seznam obrázků.....	8
1 Úvod.....	10
2 Modelování 3D objektů.....	13
2.1 Metody.....	13
2.2 Techniky procedurální modelování.....	14
2.2.1 L-systém.....	15
2.2.2 Shape Grammars.....	16
2.2.3 Split Grammars.....	17
2.3 Programy využívající procedurální modelování.....	18
2.3.1 ESRI CityEngine.....	18
3 Software Blender.....	20
3.1 Orientace v Blenderu.....	20
3.2 Základy pro manipulaci s objekty.....	22
3.3 Jednotky.....	24
3.4 Python v Blenderu.....	25
4 Generátor budov.....	28
4.1 Metoda.....	28
4.1.1 Pravidlo dělení.....	28
4.1.2 Pravidlo varianty.....	29
4.1.3 Pravidlo kopírování.....	30
4.2 Funkcionalita a implementace.....	30
4.2.1 Modelování detailních prvků budovy.....	31
4.2.2 Modelování střechy.....	32
4.2.3 Modelování výstupu na střechu a výtahové šachty.....	32
4.2.4 Modelování komínu.....	33
4.2.5 Modelování podlaží a říms.....	33
4.2.6 Modelování garáží.....	33
4.3 Instalace doplňku.....	33
4.4 Ovládání.....	35
4.4.1 Tvary půdorysů a typ objektu.....	35
4.4.2 Rozměry.....	37
4.4.3 Střechy.....	38

4.4.4 Okna	39
4.4.5 Balkony	39
4.4.6 Přízemí.....	40
4.4.7 Barvy.....	41
4.4.8 Garáže	42
4.5 Nahrávání vlastního půdorysu	42
4.5.1 Vlastní půdorys obrys	42
4.5.2 Vlastní půdorys čtverce	44
4.6 Doporučení k ovládní	45
4.7 Import do softwaru ArcGIS Desktop	47
4.8 Ukázky vygenerovaných budov	49
5 Diskuze.....	53
6 Závěr	55
Seznam použitých zdrojů informací	56
Seznam příloh	58

Seznam obrázků

Obr. 1 Strom vytvořený L-systémem s parametry uvedenými níže.....	15
Obr. 2 budovy vytvořené pomocí „CGA shape grammars“.....	17
Obr. 3 dělení části budovy pomocí „split grammars“.....	17
Obr. 4 město vygenerované pomocí softwaru CityEngine.....	19
Obr. 5 Logo Blenderu	20
Obr. 6 Typy oken	21
Obr. 7 výchozí grafické rozhraní Blenderu	21
Obr. 8 klávesové zkratky pro orientaci	22
Obr. 9 okno pro vložení objektů.....	23
Obr. 10 Nastavení jednotek v Blenderu	24
Obr. 11 „Scripting layout“	25
Obr. 12 ukázka nápovědy.....	26
Obr. 13 „Templates“	27
Obr. 14 ukázka dvou variant vchodu rodinného domu	29
Obr. 15 zmenšená strana, na které se vymodeloval balkon s oknem	31
Obr. 16 „User Preferences“.....	34
Obr. 17 ovládací panel	34
Obr. 18 parametr „Půdorysy“.....	35
Obr. 19 parametr „Typ Objektu“	36
Obr. 20 parametr „Tvar půdorysu“	36
Obr. 21 parametry „Počet podlaží“, „Rozměry“ a „Dodatečné rozměry“	37
Obr. 22 Ukázka rozměrů	37
Obr. 23 parametr „Typ střechy“	38
Obr. 24 ukázky typů střech, zleva valbová, sedlová, rovná (plochá).....	38
Obr. 25 parametr „Na střeše“ pro administrativní budovu.....	38
Obr. 26 parametr „Na střeše“ pro ostatní typy objektů.....	38
Obr. 27 skupina parametrů ovlivňující tvorbu oken.....	39
Obr. 28 skupina parametrů ovlivňující tvorbu balkonů	40
Obr. 29 parametr „vchod podle osy x“.....	40
Obr. 30 parametry pro rodinný dům.....	41
Obr. 31 parametry, pomocí kterých se nastavují barvy.....	41
Obr. 32 barevný vzorník.....	41
Obr. 33 ovládací panel pro garáže.....	42
Obr. 34 parametr „Načtení souboru“.....	43
Obr. 35 ukázka zápisu obrysu půdorysu, kde mají všechny strany délku 5	43

Obr. 36 ukázka zápisu půdorysu, který se skládá ze čtverce a dvou obdélníků	44
Obr. 37 ukázka dvou půdorysů, které porušují pravidla a jednoho, který pravidla splňuje	45
Obr. 38 strana složená ze tří různých stran	46
Obr. 39 složení okna	46
Obr. 40 možné formáty exportu	47
Obr. 41 „Marker Symbol“	47
Obr. 42 Multipart feature class ukázka	48
Obr. 43 použití nástroje „Replace With Model...“	49
Obr. 44 bytový dům	49
Obr. 45 rodinný dům s více bytovými jednotkami a sedlovou střechou	50
Obr. 46 budovy.....	50
Obr. 47 řadové garáže	51
Obr. 48 bytový dům 2	51
Obr. 49 administrativní budova.....	52

1 Úvod

V posledních padesáti letech zažilo lidstvo obrovský pokrok vědy a techniky. Každým dnem se objevují nové technické vymoženosti, které lidem usnadňují život. Velmi výrazně se projevuje technický pokrok v informatice. Virtuální prostředí, animace, vytváření her a filmů s reálnou grafikou, to vše je dnes běžnou součástí našich životů. Informatika se tímto dostává téměř do všech vědních oborů. Geografie je jeden z těchto vědních oborů, který v druhé polovině 20. století díky rozvoji informatiky sám zažil obrovský rozvoj.

Dnes již geografové běžně používají nejrůznější geoinformační systémy a pro sběr dat využívají družice, laserové skenování, GPS, radary apod. Mapy se tvoří převážně na počítačích a ruční tvorba téměř vymizela. Navzdory tomuto vývoji jsou mapy stále převážně tvořeny ve 2D, jelikož je to klasický způsob, na který jsou lidé zvyklí. Technický pokrok však způsobuje rozšíření 3D modelů i v oblasti geografie a kartografie. Orientace v běžné 2D mapě je pro mnoho lidí velmi složitá, ale 3D mapa jim orientaci pomáhá zjednodušit a je pro uživatele atraktivnější.

3D mapy jsou dnes již dostupné v různých internetových aplikacích. Jejich obsah pomáhá zpřesnit některé analýzy, jako například viditelnost či proudění vzduchu. Dále se 3D obsah využívá pro rekonstrukci zaniklých popřípadě historických budov a v oblastech územního plánování, dopravy, navigace apod. (Brůha 2017).

Tato práce se zaměřuje hlavně na 3D modely budov. Tvorba 3D modelů budov není jednoduchá záležitost a tudíž s sebou nese řadu problémů. Pro tvorbu 3D modelů budov jsou nejvíce využívány generativní techniky procedurálního modelování, které výrazně zvyšují variabilitu modelů a umožňují generovat velmi složité modely (Smelik et al. 2014). Na druhou stranu nastavování parametrů těchto technik je pro začínajícího uživatele dost složité a tyto parametry nejsou zcela intuitivní. Často vstupní parametry nemají takový impakt na výsledek, jak by uživatel čekal, jelikož pravidla generativních technik procedurálního modelování postrádají kontrolu od uživatele a jsou řízeny sadou parametrů, které nejsou vždy předvídatelné a znemožňují tak uživateli plnou kontrolu nad výsledkem (Smelik et al. 2011). Naprostá většina programů, které využívají procedurální modelování pro tvorbu 3D obsahu, nejsou open source softwary a je potřeba mít zakoupenou licenci pro práci s nimi. Odstranění výše uvedených

nedostatků, ale i přizpůsobení existujících řešení specifickým požadavkům uživatele, není v důsledku uzavřeného zdrojového kódu u proprietárních systémů možné.

Práce si proto klade za cíl přispět k vyřešení těchto problémů kombinací výhod manuálního a procedurálního přístupu k modelování. Obecně, přístup k řešení použitý v této práci umožní tvůrcům 3D modelů se více soustředit na to, *co* chtějí vytvořit, spíše než na to, *jak* toho dosáhnout.

Cílem je vytvoření generátoru budov, který umožní jednoduše a intuitivně generovat 3D modely budov a následně využít vygenerované modely v geoinformačních softwarech. Dílčí cíle práce jsou:

- zvýšení produktivity tvůrců 3D modelů za současného uchování kontroly nad generovanými výsledky,
- poskytnutí intuitivních prostředků 3D modelování pro uživatele bez speciální kvalifikace,
- tvorba takových 3D modelů, které budou využitelné v prostorových analýzách v geoinformačních softwarech.

Tyto dílčí cíle práce vyžadují, aby si uživatel mohl ponechat kontrolu nad geometrií modelů, mohl přizpůsobit výstupy specifickým potřebám konkrétních aplikací a v důsledku umožnil i analytické využití 3D modelů, nikoli pouhou vizualizaci.

Práce ukáže proveditelnost deklarativního přístupu k modelování s využitím open source softwaru. Využití řešení s otevřeným zdrojovým kódem je v tomto případě podmínkou nutnou proto, aby výsledná platforma byla přizpůsobitelná konkrétním požadavkům uživatelů.

Konkrétně, představené řešení má podobu doplňku do open source softwaru Blender. Ten umožňuje přístup k vlastnímu API (rozhraní pro programování aplikací) pomocí programovacího jazyku Python. Vlastní generátor tak bude vytvořen v jazyce Python s využitím knihovnic funkcí Blender API.

Osobní motivací a prvotním podnětem pro tuto práci byl nedostatek českých 3D modelů budov pro vizualizaci budov ve 3D v programu ArcGIS Desktop. Dalším podnětem bylo zjištění, že většina programů pro generování 3D modelů je licencovaná. Výsledky této práce mohou být použity pro vizualizaci budov ve 3D v geoinformačních

softwarech, které umožňují import 3D modelů, jako například ArcGIS Desktop. V budoucnu pak mohou umožnit i zvýšení analytických možností ve třetím rozměru.

Zbytek této práce má následující uspořádání. Ve druhé kapitole jsou představeny metody a generativní techniky procedurálního modelování, které se využívají pro tvorbu 3D modelů. Ve třetí kapitole je představen open source software Blender, který bude použit pro tvorbu generátoru budov. V kapitole čtvrté jsou navržena pravidla, která uživateli umožní rychlejší a intuitivnější tvorbu 3D modelů. Pravidla jsou následně implementována s využitím aplikačního rozhraní vybraného softwaru. Dále je prezentována funkcionality generátoru budov, ovládání generátoru budov a ukázka importu modelů do geoinformačního softwaru. Prezentace výsledných 3D modelů je uvedena v kapitole 4.8.

V kapitole 5 jsou diskutovány výhody a nevýhody řešení ve vztahu k již existujícímu softwaru CityEngine a možná rozšíření do budoucna. Práce je shrnuta v kapitole 6.

2 Modelování 3D objektů

3D modely reprezentují skutečné prostorové objekty ve virtuálním prostředí. Čím více se tyto modely podobají skutečným objektům, tím přirozenější jsou pro uživatele a dokáží navodit pocit reálnosti. Takové modely je ale těžké vytvořit, jelikož objekty v reálném světě jsou velmi variabilní a téměř každý objekt se od ostatních liší. Metody pro vytváření 3D modelů mají mezi sebou dvě hlavní odlišnosti, úroveň podobnosti se skutečnými objekty a efektivita při vytváření více modelů. Níže jsou uvedeny základní metody pro tvorbu 3D modelů.

2.1 Metody

Jedna z dnes již příliš nepoužívaných metod tvorby 3D modelů je manuální metoda. Uživatel musí manuálně zadat geometrický popis daného prostorového objektu, aby se objekt vymodeloval. Tato metoda je jak časově náročná, tak velmi neefektivní pro modelování více objektů.

Více využívaná metoda pro tvorbu 3D modelů je využití některého 3D modelovacího softwaru. Využití automatických či poloautomatických funkcí, které software nabízí, umožňuje uživateli vymodelovat velmi přesný 3D model. Tato metoda vyžaduje alespoň lehké umělecké cítění a zkušenosti s 3D softwary. Je nejvíce používána pro tvorbu přesného 3D modelu konkrétního prostorového objektu. Nevýhoda této metody spočívá v časové náročnosti.

Metoda využívající laserové skenování je další možnou metodou tvorby 3D objektů. Jedná se o metodu, jenž zpracuje mračno bodů, které vzniklo laserovým skenováním. Nejvíce se využívá pro rekonstrukci budov ve městech. Tato metoda je stále vyvíjena a každý rok se objevují nové metody, jak data zpracovat. Například již existuje metoda, která plně automatizovaně dokáže z prvotních dat laserového skenování vytvořit plášť budov, zatím bez výplní otvorů (Yi et al. 2017).

V poslední době se ale nejvíce využívá metoda procedurálního modelování. Tato metoda nemá jednotnou definici, ale zahrnuje širokou škálu generativních technik, které automatizují tvorbu určitého obsahu pomocí nastavených parametrů. Pro zjednodušení je možné si představit procedurální modelování jako sadu pravidel, která ovlivňují modelování 3D modelu. Největší výhodou procedurálního modelování je množství

modelů, které je možné vygenerovat. Stačí nastavit několik parametrů a je možné vygenerovat velké množství rozdílných modelů, navíc ve většině těchto pravidel figurují náhodná čísla, a proto je možné dojít k různým výsledkům z jednoho specifického nastavení parametrů (Smelik et al. 2014). Tato metoda šetří velké množství času, jelikož je možné s jejím využitím vymodelovat celé město na jedno nastavení parametrů. Nevýhodou generativních technik procedurálního modelování je omezená kontrola uživatele nad výsledkem, jelikož parametry, které ovlivňují generování, nejsou vždy zcela intuitivní a jsou složité na zápis. Pro reprezentaci skutečných objektů je tato metoda vhodná jen v některých oblastech, například generování stromů a skal. Při generování těchto objektů není nutné až na výjimečné případy zachytit přesnou podobu objektu ve skutečnosti. Při modelování měst je to ale jiné, jelikož každé město má specifické budovy, které se liší i podle kultury či geografické polohy. Pro vytvoření specifického města by bylo nutné vytvořit nová pravidla apod., což je samozřejmě možné, ale již více časově náročné. Procedurální modelování se neustále vyvíjí a je jen otázkou času, než bude téměř bez nevýhod.

2.2 Techniky procedurální modelování

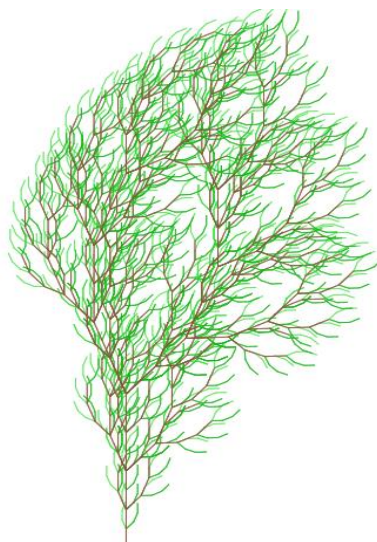
Základem procedurálních technik je princip gramatiky. Gramatika je slovník, který popisuje strukturu daného jazyka. Obsahuje sadu pravidel, která mají za úkol dvě věci. Mohou být použita pro vytváření sekvence symbolů v daném jazyce a mohou určovat, zdali je daný symbol součástí daného jazyka (Chomsky 1956). Tato pravidla popisují formování sekvence symbolů, které patří do abecedy daného jazyka. Je to tedy generátor daného jazyka. Generování symbolů funguje na principu přepisovacího systému. Tento systém má za úkol získávání nových symbolů z již napsaných symbolů podle daných pravidel. Systém začne získávat symboly z počátečního symbolu nebo počáteční sekvence symbolů. Přepisování se opakuje, dokud nejsou všechny symboly nahrazeny již nenahraditelnými symboly. Navíc mohou být nastaveny finální sekvence symbolů, které přepis ukončí (Van Leeuwen 1990).

Jednoduchý příklad může být: Jsou dána dvě pravidla „x“ → „xy“ a „y“ → „yy“, počáteční symbol je „x“. Při první iteraci se „x“ přepíše na „xy“ podle prvního pravidla. V druhé iteraci se „xy“ přepíše na „xyyy“, ve třetí na „xyyyyyyy“ atd.

Na tomto principu generování znaků podle pravidel fungují všechny procedurální techniky, které jsou uvedeny níže. Každá technika má jiné významy jednotlivých znaků. Význam těchto znaků slouží jako příkaz pro modelování. Například „x“ může znamenat vykreslení linie a „y“ vykreslení polygonu apod.

2.2.1 L-systém

Lindenmayerův systém neboli L-systém je obdobou již zmíněné gramatiky. Hlavní rozdíl mezi těmito mechanismy je v metodách aplikování přepisu symbolů. Zatímco v gramatice se přepis aplikuje postupně, v L-systému se aplikuje souběžně a všechny symboly jsou nahrazeny najednou. Tento rozdíl vznikl na základě buněčného dělení mnohobuněčných organismů, kdy může dojít k mnoha dělení současně (Prusinkiewicz a Lindenmayer 1990).



Obr. 1 Strom vytvořený L-systémem s parametry uvedenými níže

Zdroj: (<http://www.kevs3d.co.uk/dev/lsystems/>)

Nejlépe jde L-systém reprezentovat pomocí želví grafiky. Želva je virtuální kurzor, který se pohybuje v kartézském systému souřadnic a poslouchá příkazy, které mu program zadá. Příkazy jsou dány pravidly, viz příklad níže.

F → posuň se o vzdálenost d, + → otoč se doprava o úhel δ , - → otoč se doleva o úhel δ , [→ uložení pozice, na které se želva nachází do zásobníku (datová struktura),] → načtení pozice, která se nachází na vrchu zásobníku (Prusinkiewicz 1986).

- Počáteční bod: F

- Pravidla: $F \rightarrow FF+[+F-F-F]-[-F+F+F]$
- Úhel natočení: $\delta = 22,5^\circ$
- Vzdálenost d posunutí želvy: 5
- Počet iterací: 4

Pomocí symbolů „[“ a „]“ je želva schopná kreslit jednotlivé větve, jelikož si ukládá pozici, odkud vyšla.

L-systém je jednou z prvních procedurálních technik a nejvíce se využívá pro modelování rostlin a stromů. Dříve se L-systém využíval i pro generování budov například, když byl poprvé představen software CityEngine, využíval pro generování budov právě L-systém. L-systém modeluje budovy stejně jako rostliny. Symboly reprezentují příkazy pro vykreslení jednotlivých částí budovy. Pomocí pravidel jsou symboly nahrazovány a tím se mění části budovy, které se vykreslují (Parish a Müller 2001). Později byl ale L-systém nahrazen gramatikou tvarů, viz níže. Dnes se L-systém využívá převážně pro generování rostlin a cestních sítí.

2.2.2 Shape Grammars

„Shape Grammars“ neboli gramatika tvarů pracuje na stejném principu jako výše zmíněná gramatika. Gramatika tvarů je ale definována abecedou tvarů, kde každý tvar generuje další tvary. Tvar je uspořádání linií, ze kterých jsou tvary poskládány ve 2D. Finální tvar je výsledkem generování tvarů podle pravidel z počátečního tvaru. Proces generování je ukončen, pokud již není možné aplikovat žádná pravidla (Stiny 1975).

Gramatika tvarů je jednou z nejpoužívanějších technik procedurálního modelování. Gramatika tvarů pro architekturu počítačové grafiky neboli „CGA shape grammars“, kterou představili Müller et al. (2006), dokáže generovat budovy z několika půdorysů. Pravidla jsou nastavená tak, aby po vygenerování nedošlo k překrytí střech nebo vymodelování dveří a oken do strany, která sousedí s vedlejší budovou apod.

Nejdříve gramatika nahradí plášť budovy za půdorys, následně nahradí jednotlivá podlaží za plášť budovy, jednotlivá podlaží nahradí dalšími elementy atd. Takto se postupuje až k finálnímu vzhledu budovy. Po nalezení finální vzhledu se začnou modelovat jednotlivé objekty. Tento fakt má obrovskou výhodu při modelování, jelikož jen výsledná geometrie je modelována a tím se šetří potřebná paměť a čas pro modelování.

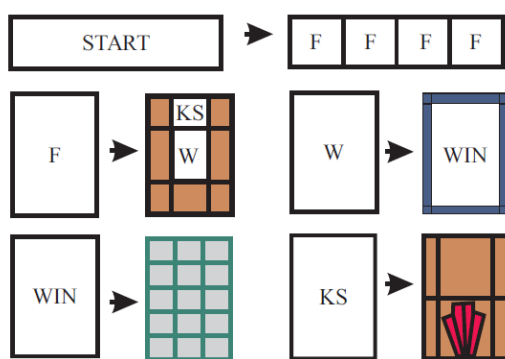


Obr. 2 budovy vytvořené pomocí „CGA shape grammars“

Zdroj:(Müller et al. 2006)

2.2.3 Split Grammars

„Split Grammars“ neboli gramatika dělení je další gramatika, pomocí které se dají efektivně generovat budovy. Jádrem této gramatiky je dělení částí budovy až na „basic shapes“ neboli základní objekty gramatiky dělení, což jsou jednoduše geometricky definovatelné objekty, jako kvádry, hranoly nebo válce. „Split grammars“ se dělí na tři základní mechanismy. Základem je gramatika dělení, která vytvoří ze základního tvaru



Obr. 3 dělení části budovy pomocí „split grammars“

Zdroj:(Wonka et al. 2003)

fasádu. Následně fasádu dělí na jednotlivé individuální elementy, jako jsou okna, dveře apod. Další důležitou část gramatiky tvoří „control grammar“, který přiřazuje jednotlivé atributy každému elementu a zároveň jednotlivým pravidlům. Poslední částí celé

gramatiky je atributový mechanismus, který vypočte nejlepší možné pravidlo, jenž se má použít na jednotlivý tvar na základě shody jejich atributů. Každý nový tvar zdědí atributy tvaru, ze kterého byl vygenerován. Uživatel si může atributy a pravidla různě nastavit a tím ovlivnit vzhled budovy (Wonka et al. 2003).

2.3 Programy využívající procedurální modelování

V této kapitole stručně představím oblasti, kde se využívá procedurální modelování a programy, ve kterých je možné generovat zmíněný obsah.

První oblastí, kde se využívá procedurální modelování je terén. Terén se nejlépe reprezentuje výškovými mapami a nadmořskou výšku terénu je možné jednoduše vymodelovat procedurálně. Dnešní techniky umožňují modelovat spolu s terénem i jeskyně, volné kameny, převisy, oblouky apod., dokonce je možné znázornit erozi. Terén je možné generovat v programech TerraGen, GeoControl, L3DT a dalších. Vegetace je další oblastí, ve které se využívá procedurální modelování. Modelování vegetace je možné dělit na modelování jednotlivých částí rostliny, celých rostlin a jednotlivých ekosystémů. Základem pro modelování vegetace je již zmíněný L-systém, který byl rozšířen, aby bylo možné modelovat rostliny i ve 3D. Některé programy pro modelování vegetace jsou například SpeedTree, XFrog nebo produkt softwaru e-on Plant Factory. Nejméně rozšířená oblast pro procedurální modelování jsou vodní útvary, jelikož je velmi těžké zajistit, aby byly modelované řeky hydrologicky korektní. Program, který generuje nejen vodní útvary, ale celé scenérie VUE je dalším produktem e-on software. Poslední oblastí jsou města, kde se uplatňuje procedurální modelování nejvíce. Do této oblasti spadají cestní sítě, rozložení města, budovy a interiér budov. Zde se využívají hlavně modelovací techniky zmíněné výše. Nejznámější programy pro modelování měst a budov jsou CityEngine a Houdini (Smelik et al. 2014).

2.3.1 ESRI CityEngine

CityEngine je pravděpodobně nejznámější 3D modelovací software v oblasti geografie, jelikož stejně jako ArcGIS Desktop je CityEngine produktem společnosti ESRI. Poskytuje grafické uživatelské rozhraní a také nabízí možnost využití programovacího jazyku Python pro ovládání. CityEngine poprvé představili Parish a Müller (2001). Prvotně CityEngine využíval k modelování budov a cestní sítě L-systémy, dnes využívá již zmíněnou gramatiku „CGA shape grammars“. CityEngine dokáže od začátku



Obr. 4 město vygenerované pomocí softwaru CityEngine

Zdroj:(<https://cehelp.esri.com/help/index.jsp>)

vygenerovat celé město na základě vstupních dat, jako terén či cestní síť. CityEngine nabízí možnost vybrat si několik předdefinovaných cestních sítí a typů měst pro rychlé modelování. Cestní síť je možné tvořit i manuálně. Modelování podoby objektů je již složitější, vzhled a umístění budovy se nastavuje pomocí pravidel CGA gramatiky. Používání pravidel je velice mocný nástroj, pomocí kterého je možné vygenerovat téměř nekonečné množství různých podob objektů. Na druhou stranu tato pravidla nemusí být zprvu intuitivní a nejsou zcela jednoduchá pro zapsání. CityEngine také nabízí manuální modelování budov a editování již vygenerovaných budov (Esri R&D Center Zurich 2013).

Možnosti modelování softwaru CityEngin jsou obrovské a není možné sepsat veškeré informace o tomto programu do této práce, více informací o softwaru CityEngine je možné najít na stránkách <https://cehelp.esri.com/help/index.jsp>.

3 Software Blender

Blender je open source software, který slouží k vytváření 3D obsahu. Má velmi rozsáhlé možnosti použití od modelování přes tvorbu animací, až po vytváření her. Blender také nabízí využití programovacího jazyka Python pro tvorbu různých nástrojů, funkcí, 3D modelů apod. Blender podporuje operační systémy Windows, Linux a Mac a je určen převážně pro malé společnosti nebo pro jednotlivce. Nachází využití v oblastech reklam, filmů, her, vědeckých výzkumů, umění a mnoha dalších.

Blender vyvíjí nezávislá nezisková veřejně prospěšná společnost Blender Foundation, která sídlí v Amsterdamu v Nizozemsku. Jejím účelem je vytvořit, udržovat a vylepšovat služby pro aktivní uživatele a vývojáře Blenderu a poskytnout Blender jednotlivcům a malým společnostem bezplatně a plnohodnotně. Společnost nabízí na stránkách mnoho tutoriálů a pořádá přednášky, aby zasvětila co nejvíce nových lidí (Blender Foundation 2016).



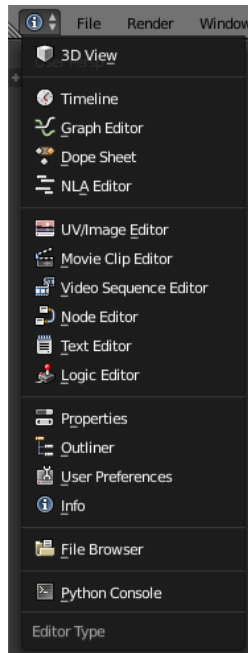
Obr. 5 Logo Blenderu

Zdroj:(<https://www.blender.org/about/logo>)

3.1 Orientace v Blenderu

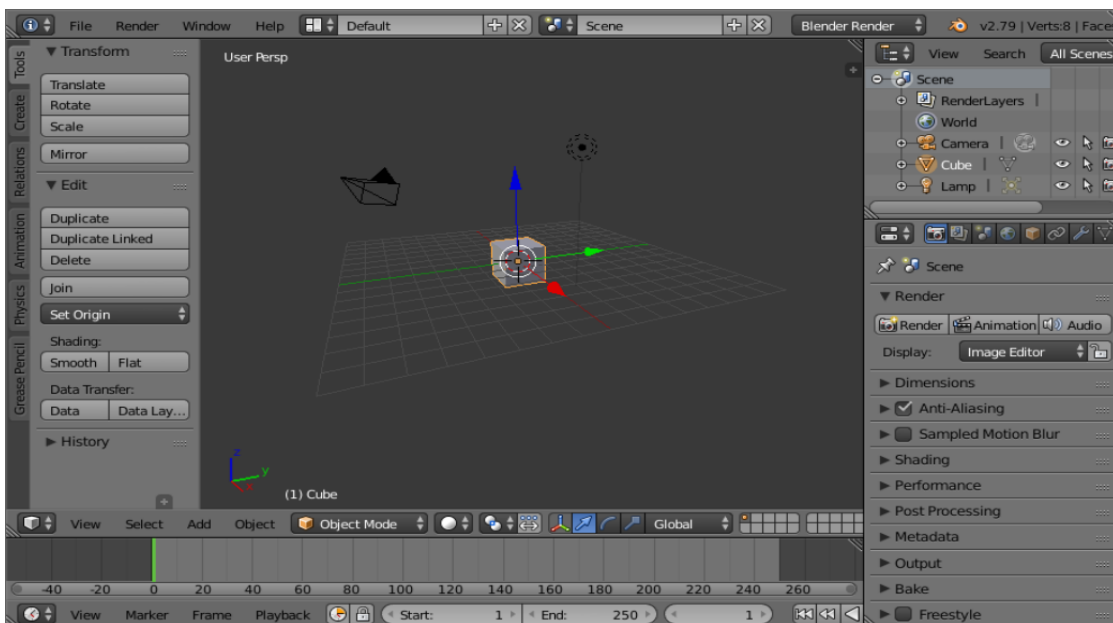
Po instalaci a úspěšném spuštění Blenderu se uživatel dostává do grafického rozhraní Blenderu. Zde vidí výchozí rozložení grafického rozhraní, které může zprvu působit nepřehledně. Na každý software si je možné zvyknout, ale Blender má tu výhodu, že si uživatel může rozhraní různě měnit. Případně si může pomocí Pythonu napsat své vlastní rozhraní, které mu bude vyhovovat. Každou změnu si je možné uložit.

Ve výchozím rozhraní se nachází několik typů oken, typy oken se dají měnit ikonou v levé části okna. Některá okna ji mají dole některá nahoře. Po kliknutí na ikonu se objeví nabídka všech typů oken, typy oken jsou vidět na *obrázku číslo 6*. Každá okna potřebují pro práci s nimi různou velikost, proto je možné okna jakkoliv zvětšovat či zmenšovat nebo měnit apod. (Pokorný 2009).



Obr. 6 Typy oken

Ve výchozím rozložení se nachází v pravém horním rohu seznam objektů ve scéně. Pod ním se nachází „properties panel“, ve kterém je možné najít záložky pracující s aktivním objektem a další užitečné záložky, jako osvětlení, render apod. Ve spodní části rozhraní se rozkládá menu časové osy pro animace, uprostřed se nachází největší okno s 3D prostředím a v horní části je vidět lišta hlavního menu. Výchozí grafické rozhraní je možné vidět na *obrázku číslo 7* (Blender Foundation 2016).



Obr. 7 výchozí grafické rozhraní Blenderu

Zdroj:(https://docs.blender.org/manual/en/dev/interface/window_system/introduction.html)

Nejdůležitější pro nového uživatele je orientace ve 3D prostředí, kde si uživatel prohlíží vytvořené objekty. Podržení kolečka na myši a následným pohybem myši může uživatel pohybovat s pohledem na scénu. Tento způsob je nejvíce efektivní pro obyčejné prohlížení, ale není vhodný pro nastavení pohledu přesně ze shora nebo přesně ze strany. Pro tyto případy je možné použít klávesové zkratky, které jsou patrné z *obrázku číslo 8*.

Basic Navigation

Orbit	Middle Click
Pan	Shift + Middle Click
Zoom	Scroll
Top View	Numpad 7
Front View	Numpad 1
Side View	Numpad 3
Camera View	Numpad 0
Go to Selected Object	Numpad .
Perspective/Orthographic	Numpad 5
Zoom to Show All Objects	Home
Fly Mode	Shift + F
Toolbar	T
Object Properties	N
Reset 3D cursor	Shift + C

Obr. 8 klávesové zkratky pro orientaci

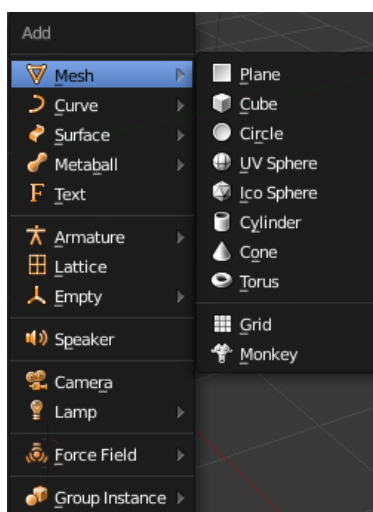
Zdroj: (<https://www.blenderguru.com/articles/free-blender-keyboard-shortcut-pdf>)

Pokud má uživatel notebook a nemá myš a numpad, může využít možnosti Blenderu a nastavit alternativní klávesy pro kolečko u myši a numpad. Nastavení alternativních kláves se nachází v okně „User Preferences“. Stačí kliknout na horní liště na „File“ → „User Preferences...“ → „Input“ → a zaškrtnout „Emulate numpad“ a „Emulated 3 Button Mouse“.

3.2 Základy pro manipulaci s objekty

Každý nový uživatel by měl znát základní funkce Blenderu. Úplný základ je přidání objektu do scény, stačí zmáčknout klávesy „shift + A“ a objeví se okno pro přidání objektu. Okno pro přidání objektu je možné vidět na *obrázku číslo 9*. Možnost „Mesh“ nabízí několik možných objektů pro vložení. Tato možnost je nejčastěji používaná, objekty jsou tvořeny z bodů, hran a ploch, které je možné libovolně upravovat v editačním módu. Objekty se vždy vloží na místo, kde je umístěný 3D kurzor, který je

možné nastavit levým tlačítkem (pokud je tlačítko přenastaveno, tak pravým tlačítkem) myši (Pokorný 2009).



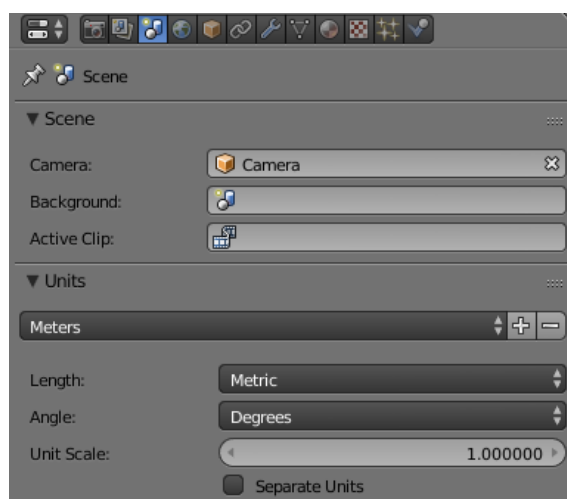
Obr. 9 okno pro vložení objektů

Základní funkce pomáhají při manipulaci s vytvořenými objekty a jsou naprostým základem pro nového uživatele. Například použitím nejrůznějších Blender doplňků si každý uživatel může vytvořit různé 3D modely. Pro export modelu nebo pro export obrázku scény je užitečné umět tyto objekty lehce natočit, zvětšit, zmenšit nebo rozestavit podle potřeby. Pro tyto účely slouží funkce „scale“ s klávesovou zkratkou „S“, „rotate“ s klávesovou zkratkou „R“ a „grab/move“ s klávesovou zkratkou „G“. Aby mohly být tyto funkce použity, je zapotřebí nejdříve označit objekt, na který mají být použity. Označení objektu se v Blenderu provádí pravým kliknutím myši. To je pro běžného uživatele velmi nepraktické, jelikož ve „všech“ programech se tento úkon provádí pomocí levého tlačítka myši. Doporučuji si proto změnit nastavení označení objektů z pravého tlačítka myši na levé. Toto nastavení se nachází na stejném místě jako předchozí možnost „Emulated 3 Button Mouse“. Po označení objektu je možné použít jednu z funkcí stisknutím její klávesové zkratky. Práce s objekty ve 3D nemusí být zprvu intuitivní, a proto výsledky použití funkcí pravděpodobně nebudou vypadat podle představy. Blender totiž pracuje v kartézském souřadnicovém systému se třemi osami „x“, „y“, „z“. Při použití některých z výše uvedených funkcí, Blender neví, jakým směrem se má objekt pohybovat, natáčet či zvětšovat, a proto se účinek funkcí promítá do všech tří os. Po zmáčknutí klávesové zkratky funkce, je tedy zapotřebí specifikovat osu, podle které se má objekt pohybovat, natáčet či zvětšovat. Pro specifikaci osy stačí stisknout klávesu dané osy.

Pro uživatele, kteří preferují čísla je zde klávesová zkratka „N“, která otevře vlastnosti označeného objektu. Zde je možné si všechny ze tří zmíněných funkcí nastavit pro jednotlivé osy za pomoci číselných hodnot. Zmíněné funkce pracují s celými objekty, pouze pokud má uživatel nastavený „Object mode“, který je v Blenderu nastavený jako výchozí. Mnoho dalších užitečných funkcí se již provádí v editačním módu. „Edit mode“ je již velmi obsáhlý a proto není možné ho zde podrobně rozebrat (Blender Foundation 2016).

3.3 Jednotky

Blender používá jako výchozí jednotky tzv. „Blender unit“ (dále jen BU). BU jsou abstraktní jednotky, které nemají ve skutečnosti žádný ekvivalent jednotek délky. Pro tvorbu budov je lepší pracovat s metry. Doporučuji si tedy představovat, že $1 \text{ BU} = 1 \text{ metr}$. Pro uživatele, kteří nemají takovou fantazii, nabízí Blender nastavení jednotek délky podle libosti. V záložce „Scene“ → „Units“ si může uživatel zvolit libovolné jednotky, například metry viz *obrázek číslo 10*. Na výsledek to ale nebude mít žádný vliv. Jediné, co se změní je souřadnicová síť Blenderu, na které se objekty modelují. Když si uživatel zvolí jednotky milimetry, souřadnicová síť se zvětší a objekty budou vypadat malé. Nastavené jednotky nemají na export objektu žádný vliv. Pro všechny příklady v této práci jsou použity BU.

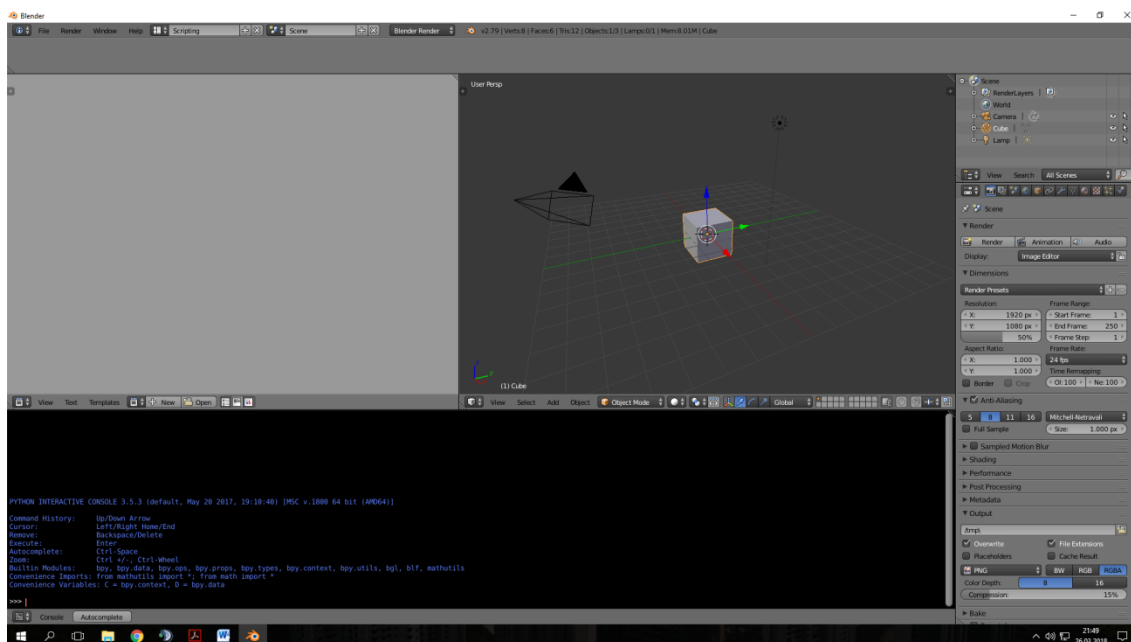


Obr.10 Nastavení jednotek v Blenderu

3.4 Python v Blenderu

Blender nabízí možnost využít při práci programovací jazyk Python. Dříve se pomocí Pythonu dalo provádět omezené množství úkonů, ale s příchodem Blenderu 2.5 se možnosti Pythonu výrazně zvýšily. V současnosti je možné pomocí Pythonu provádět téměř všechny operace, které je možné provádět v grafickém rozhraní. Pro velkou většinu uživatelů je tento rozsah dostačující. Blender se stále vyvíjí a je možné, že zanedlouho již bude možné v Pythonu provádět všechny operace. Novější verze Blenderu, tedy od 2.5 dále používají Python 3.x.

Zdrojový kód je možné psát přímo v samotném Blenderu. Stačí si na horní liště v možnosti „Choose Screen layout“ vybrat „Scripting“. Na pravé straně zůstává okno s 3D pohledem a „properties panel“, na straně levé se objevuje textový editor, ve kterém se píše samotný kód. Ve spodní části se nachází interaktivní konzole pro Python a pod



Obr.11 „Scripting layout“

horní lištou přibylo nové okno. Toto nové okno je jednou z obrovských výhod Blenderu. Každá funkce, kterou uživatel provede „ručně“, má svůj zápis v Pythonu. Tento zápis se po provedení kterékoliv funkce objevuje právě v novém horním okně. Z tohoto okna je pak možné příkaz zkopírovat. Zkopírovaný příkaz obsahuje většinou několik nepotřebných parametrů, které mohou být smazány. Rozložení grafického prostředí pro psaní zdrojového kódu je vidět na obrázku číslo 11. Další výhodou Blenderu jsou nápovědy, které se objevují na všech tlačítkách v Blenderu, když na ně

```
Duplicate selected objects and move them
Shortcut: Shift D
Python: bpy.ops.object.duplicate_move()
```

Obr.12 ukázka nápovědy

uživatel najede myší. V této nápovědě je vždy příkaz v Pythonu, který vykonává totéž, co samotné tlačítko. Tím uživateli napoví, jak zapsat danou funkci, kterou tlačítko vykonává, viz *obrázek číslo 12* (Blender Foundation 2016).

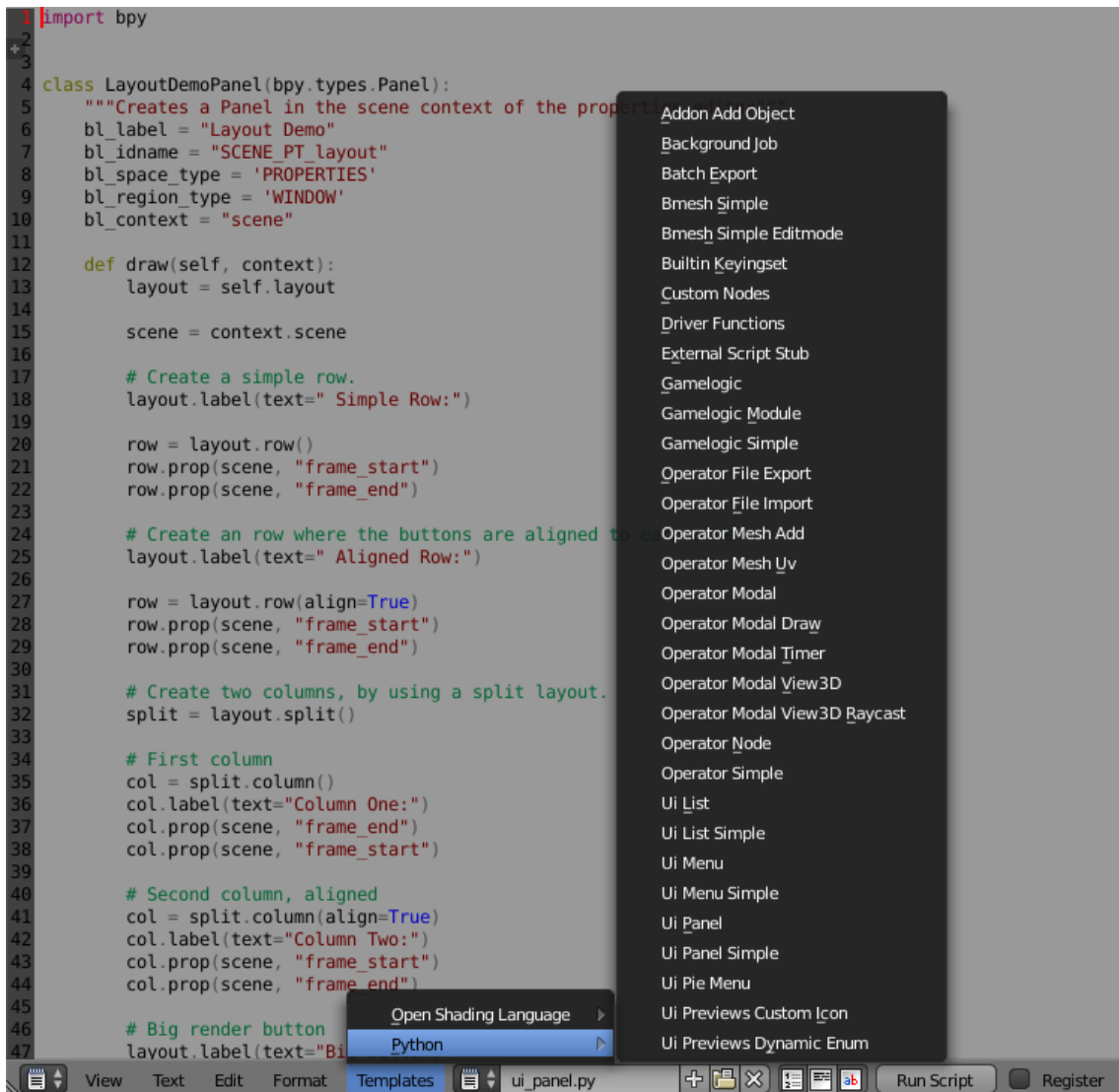
Blender má také jednu velkou nevýhodu a to je absence IDE (vývojového prostředí). Blender nabízí možnost psaní kódu v textovém editoru, který jen zvýrazňuje syntaxi. Nekontroluje tedy správnost syntaxe při psaní a hlavně zde není debugger (nástroj pro hledání chyb při ladění). Pro krátké skripty to není takový problém, ale pro velké projekty je to velký nedostatek. Na tento problém upozornil W. Jaworski (2011), když o tomto nedostatku napsal téměř celou knihu „Programming Add-Ons for Blender 2.5“. Ve své knize tento problém řeší využitím Eclipse IDE obohaceným o PyDev modul.

Podle mého názoru je tento způsob zbytečně složitý a úplně stačí pracovat s libovolným IDE pro Python. Nabízí se i další možnosti řešení tohoto nedostatku, jako například použití „pdb“ (interaktivní zdrojový kód pro debuggování v Pythonu) modulu (Python Software Foundation 2007).

Python v Blenderu má uplatnění i pro uživatele, kteří mají minimální znalosti programování. V Blenderu je možné využít tzv. „Templates“ neboli šablon, které si uživatel může zobrazit. V těchto šablonách se nachází zdrojové kódy napsané v Pythonu, které slouží jako ukázky jednotlivých operací. Těchto šablon je přímo v Blenderu celkem 32. Na *obrázku číslo 13* je vidět, jaké šablony Blender nabízí.

Kromě těchto šablon existují stránky, které slouží jako manuál Blenderu, například <https://wiki.blender.org/>, kde je možné najít nejrůznější tutoriály a ukázky programování v Pythonu pro Blender. Další užitečná stránka pro programování v Blenderu je oficiální stránka, která dokumentuje API (rozhraní pro programování aplikací) Blenderu <https://docs.blender.org/api/2.79/>. Zde je možné najít veškerou syntaxi pro Python v Blenderu a navíc je zde několik tutoriálů na ukázkou. Tyto stránky ale nepokryjí veškeré problémy, se kterými se uživatelé po celém světě setkávají. Proto doporučuji využít stránku <https://blender.stackexchange.com/>. Na tomto fóru je možné

najít snad „všechny“ problémy při programování, které mohou nového uživatele potkat. Na všechny tyto problémy již existují odpovědi a řešení od jiných zkušených uživatelů.



Obr.13 „Templates“

4 Generátor budov

Generátor budov je doplněk pro Blender, který jsem vytvořil za účelem generování 3D modelů budov pro 3D vizualizaci budov a pro případné využití modelů k prostorovým analýzám v geoinformačních softwarech. Některé programy nabízejí různé výchozí 3D modely budov, jako například ArcGIS Desktop. Velká část těchto symbolů jsou americké hotely či mrakodrapy. Pro vizualizaci měst a vesnic v České republice nebo jejich částí se tyto symboly nehodí. Proto jsem se rozhodl vytvořit tento doplněk, který umožňuje uživateli vygenerovat 3D model budovy a naimportovat ho do geoinformačních softwarů, které import modelů umožňují, například ArcGIS Desktop. Generované 3D modely budov jsou inspirovány českými budovami. Pro využívání tohoto doplňku není nutná licence a uživatel si může zdrojový kód upravit podle vlastní potřeby. Jednotlivé vygenerované 3D modely je možné velmi detailně editovat v Blenderu. Úpravou zdrojového kódu či editací si uživatel může vytvořit speciální typ 3D modelu budovy dle potřeby. Tyto vlastnosti odlišují generátor budov od již existujících softwarů, jako je CityEngine.

4.1 Metoda

Jak již bylo zmíněno v kapitole 2.1, generativní techniky procedurálního modelování neposkytují uživateli úplnou kontrolu nad výsledkem a nastavování parametrů není příliš uživatelsky přívětivé. Tyto techniky jsou navíc velice složité na implementaci. Mým cílem je vytvořit doplněk, který bude naopak jednoduchý na ovládání a umožní uživateli úplnou kontrolu nad výsledkem. Proto jsem nevyužil přímo generativní techniky procedurálního modelování, ale inspiroval jsem se jimi a před samotnou implementací jsem si stanovil jednoduchá pravidla, která zaručí větší variabilitu generovaných budov a umožní uživatelům jednoduše rozvíjet zdrojový kód.

4.1.1 Pravidlo dělení

První stanovené pravidlo, je pravidlo dělení. Toto pravidlo tvoří základ gramatiky „Split grammars“ popsané v kapitole 2.2.3.

Pravidlo jsem definoval tak, že jeho podstata spočívá v rozdělování budovy na již nedělitelné části. Tím je po vzniku všech podlaží budova rozdělena na jednotlivá podlaží a podlaží na jednotlivé strany. Strana je již nedělitelná část budovy, a jelikož

generátor budov pracuje pouze s pravoúhlými půdorysy, existují pouze dvě možnosti orientace této části, podle osy x nebo podle osy y. S využitím tohoto pravidla jsou orientace a umístění plochy jednoduše zjistitelné atributy. Pokud jsou tyto atributy známy, není problém na dané straně vymodelovat fasádu s detailními prvky, jako jsou okna, balkony apod. Toto pravidlo je také využito při tvorbě šikmých střech, když rozděluje polygony na jednotlivé čtverce či obdélníky, podrobněji je tento proces popsán v kapitole 4.2.2.

Bez pravidla dělení by nebylo možné jednoduše zaměřit část budovy, na které se má modelovat fasáda s detailními prvky. Toto pravidlo případně umožňuje uživateli rozšířit variabilitu fasád, jelikož stačí napsat funkci, která operuje se stranou podle vzoru ostatních funkcí, jenž modelují fasádu a přidat parametr pro tento typ fasády.

4.1.2 Pravidlo varianty

Pravidlo varianty zvyšuje enormně variabilitu fasády s detailními prvky na stranách budovy. Jeho podstata je velmi jednoduchá. Pro každý prvek, který se modeluje,



Obr. 14 ukázka dvou variant vchodu rodinného domu

existuje další jeho varianta. Jednotlivé varianty závisí na délkách jednotlivých stran. Pokud se na určité straně má modelovat vchod s okny, ale strana má příliš malé rozměry pro tuto variantu, vymodeluje se pouze vchod. Pokud je strana krátká i na vchod samotný, nevymodeluje se nic. Toto pravidlo platí pro všechny detailní prvky, které se modelují na stranách budovy. Pravidlo v praxi je možné vidět na *obrázku číslo 14*. Bez tohoto pravidla by mohlo při určitých rozměrech strany dojít k deformaci detailních prvků a variabilita fasády by byla o hodně nižší.

4.1.3 Pravidlo kopírování

Pravidlo vychází z předpokladu, že jednotlivé detailní prvky budovy, jako okna, balkony, vchody apod., se na budově opakují. Výjimku tvoří speciální typ budov například budovy historické. I kdyby tento předpoklad nebyl vždy pravdivý, velké bytové domy, kterých člověk vidí denně nespočet, vytváří dojem, že se vše pravidelně opakuje.

Princip pravidla je jednoduchý, vymodelované prvky se kopírují tak, aby rovnoměrně pokryly danou plochu, na které se modelují. Využití nachází toto pravidlo například u „výstupu na střechu a výtahové šachty“, kde se uplatňuje, pokud má plocha velké rozměry. Jednoduše se „výstup na střechu a výtahová šachta“ nakopíruje vícekrát, aby jejich počet odpovídal rozměrům. Například pro budovu ve tvaru obdélníku, která je dlouhá třicet BU, se vytvoří dva „výstupy na střechu a výtahové šachty“.

Pro využití plného potenciálu tohoto pravidla, je ale zapotřebí pracovat již s nedělitelnou částí budovy a aplikovat toto pravidlo na detailní prvky. Jak již bylo zmíněno, nedělitelná část budovy je strana, se kterou je možné operovat jednoduše právě díky pravidlu dělení. Kombinací pravidla varianty a pravidla kopírování dokáže program vytvářet několikrát vybranou variantu prvku na jedné straně budovy, ačkoli byl prvek vymodelován jen jednou. Bez pravidla kopírování, by modelování jednotlivých prvků muselo probíhat opakovaně a tím by se zvyšovaly nároky na operační paměť počítače. Jak přesně prvek program modeluje je popsáno v kapitole 4.2.1.

4.2 Funkcionalita a implementace

Doplněk je psaný v programovacím jazyce Python, který Blender podporuje. Pomocí Pythonu jsem přistoupil k sadě funkcí, které Blender nabízí a napsal jsem zdrojový kód pro vlastní generátor. Využil jsem převážně funkce pro manipulaci s geometrií jednotlivých objektů. Zdrojový kód jsem psal přímo v textovém editoru, který Blender nabízí. Největším problémem bylo zvládnutí syntaxe funkcí Blenderu, jelikož to pro mě bylo úplně něco nového. Generátor budov jsem naprogramoval v Blenderu 2.79. Tato verze je tedy ideální pro jeho používání. Doplněk by měl fungovat i na jiných verzích, tato možnost ale není otestována. Zdrojový kód je plně okomentován a je možné ho najít na přiloženém CD „Generátor budov.py“. Ve zdrojovém kódu lze najít veškeré

informace o implementaci jednotlivých částí generátoru, a proto zde nebudu rozebírat implementaci jednotlivých funkcí, ale popíši zde funkcionalitu programu.

Proces modelování budov začíná od půdorysu. Nejprve se vytvoří půdorys spojením bodů a hran. Pokud si uživatel vybral doplňky na střeše, jako je „komín“ nebo „výstup na střechu a výtahová šachta“, vymodelují se nejprve tyto doplňky a ty se umístí na zatím pomyslnou střechu budovy. Dále se z půdorysu modeluje jedna z šikmých střech. Pokud je vybrána, umístí se tak, aby přiléhala k poslednímu zatím nevytvořenému podlaží budovy. Po vymodelování střechy se znovu vytvoří půdorys a začnou se modelovat zatím prázdná podlaží. Pokud se modelují římsy nebo rovná střecha, modelují se po vymodelování všech podlaží. Následně je budova rozdělována na jednotlivá podlaží a podlaží na jednotlivé strany. Zde probíhá modelování detailních prvků, jako jsou okna, balkony, vchod, garáže apod. Před koncem modelovacího procesu se všechny části budovy spojí do jednoho objektu.

4.2.1 Modelování detailních prvků budovy

Detailní prvky se mohou začít modelovat až po rozdělení budovy na nedělitelné části. Nejprve program rozpozná orientaci strany pomocí její délky. Pokud je orientovaná podél osy x, nemá délku v ypsilonovém směru. Dále zjistí její délku a porovná, jestli je možné zde vymodelovat danou variantu prvku. Vždy začne porovnávat variantu daného prvku s největšími rozměry. Pokud jsou rozměry strany příliš malé, použije se postup zmíněný v kapitole 4.1.2. Pokud je možné daný prvek vymodelovat, program zjistí,



Obr. 15 zmenšená strana, na které se vymodeloval balkon s oknem

kolikrát se daný prvek do rozměrů stěny vejde. Přirozeně téměř nikdy není výsledek celé číslo. Program tedy dopočítá, o kolik by bylo nutné zmenšit mezery, aby se daný prvek na stranu vešel ještě jednou. Jsou-li modelována okna či balkony, program počítá

i s možností mezery zvětšit a vybere vhodnější variantu. Následně je celá strana zmenšena na přesné rozměry potřebné pro vytvoření jednoho prvku. Po vymodelování prvku se tato zmenšená strana zkopíruje tolikrát, aby spolu s originálem tvořily původní rozměry strany. Poté program přistoupí k další straně a proces se opakuje. *Obrázek číslo 15* zobrazuje stav strany po zmenšení a vymodelování prvku.

Například pokud bude mít strana délku 10 BU, okno šířku 2 BU a mezera šířku 0,7 BU. Program si sečte šířku mezery, okna a mezery, což je jeden prvek, podrobněji je to rozepsáno v kapitole 4.6 a provede výpočet $10 \text{ BU} / 3.4 \text{ BU} = 2.94 \text{ BU}$. Na základě toho program rozhodne, že budou mezery zmenšeny a vymodelují se okna tři.

4.2.2 Modelování střechy

Modelování rovné střechy není nutné komentovat, a proto přejdu přímo k modelování střech šikmých. Program vytváří střechy přímo z půdorysu. Nejprve rozloží půdorys na nedělitelné části. Dále se program zaměří na jeden polygon, který je rozdělen přes střed na osm částí jako „pizza“. Toto je důležité, aby mohlo po vyzvednutí střechy vzniknout úžlabí. Pro tvorbu střechy je nutné zajistit její návaznost mezi jednotlivými polygony. Je tedy nutné, znát vzájemnou polohu polygonů v prostoru. Program porovná x medián a y medián vybraného polygonu s mediány ostatních polygonů. Samotné porovnání mediánů nestačí pro zjištění souseda polygonu. Pokud se x nebo y mediány rovnají, je nutné porovnat i druhé mediány, aby se zjistilo, z jaké strany polygony sousedí. Dále je nutné ověřit, zda spolu sousedí přímo nebo je mezi nimi mezera. Tento proces program vykoná pro každý polygon. Při tvorbě valbové střechy je vyzdvižena prostřední část polygonu od středu směrem ke všem sousedícím polygonům. Jsou tedy rozlišovány čtyři směry. Vyzdvižení se provede pro všechny polygony. Pro střechu sedlovou se vyzdvižení nedělí na čtyři směry, ale jen na dva směry, směr vodorovný a směr svislý. Pokud existuje soused v jednom ze směrů, vyzvedne se prostřední část polygonu v daném směru. Tím se vytvoří štíty. Tento zdvih se provede pro všechny polygony. Dále se jednotlivé části střechy spojí v jeden objekt.

4.2.3 Modelování výstupu na střechu a výtahové šachty

Po vymodelování půdorysu, přejde program u plochých střech k modelování „výstupu na střechu a výtahové šachty“. Program nejprve rozdělí půdorys na jednotlivé nedělitelné části. Zaměří se na jeden polygon, a pokud jsou rozměry polygonu

dostatečné, vymodeluje „výstup na střechu a výtahovou šachtu“. Porovná, jestli je daný polygon delší či širší. Na základě toho se „výstup na střechu a výtahová šachta“ natočí směrem, který má větší rozměry. Následně je „výstup na střechu a výtahová šachta“ přemístěn na střed polygonu. Pokud je polygon dostatečně dlouhý, „výstup na střechu a výtahová šachta“ se nakopíruje podle velikosti většího rozměru daného polygonu. Kopírování probíhá zároveň s přesunováním, aby objekt „výstup na střechu a výtahová šachta“ zachoval symetrii vzhledem k polygonu. Celý proces se opakuje pro každý polygon, z nichž je tvořen půdorys.

4.2.4 Modelování komínu

Modelování „komínu“ probíhá velice jednoduše. „Komín“ se vymodeluje a přiřadí se prvnímu polygonu, který program najde po rozdělení půdorysu. „Komín“ se přemístí na střed polygonu, odkud je následně lehce posunut, aby neprocházel hřbetem a nárožím.

4.2.5 Modelování podlaží a říms

Jednotlivá podlaží jsou modelována přímo z půdorysu. Plocha půdorysu je postupně vytahována do výšky jednotlivých podlaží, čímž podlaží vznikají. Římsy se modelují ze „stropů“ daných podlaží. Podlaží jsou zvětšena od stropu směrem nahoru o výšku římsy. Této části podlaží se následně přiřadí velká tloušťka, čímž vznikne efekt římsy.

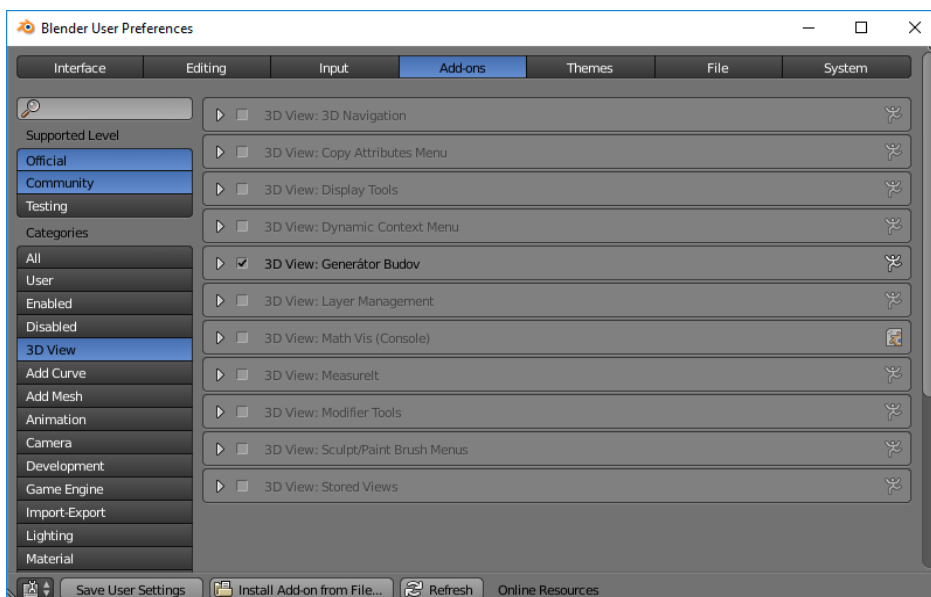
4.2.6 Modelování garáží

„Garáže“ jsou speciální typ objektu, ke kterému si nelze vybrat půdorys. Proto je jejich modelování velmi snadné. Program vymodeluje jednu garáž, která je následně kopírována, či různě natáčena.

4.3 Instalace doplňku

Doplňek, který jsem vytvořil, je nutné nejprve nainstalovat do Blenderu. Samotná instalace je celkem snadná. Po otevření příloženého CD, uživatel zkopíruje soubor „Generátor budov.py“ do libovolné složky v počítači. Spustí Blender a nainstaluje doplňek, klikne v horní liště vlevo na „File“ → „User Preferences...“ → „Add-ons“ → „Instal Add-on from a File...“, najde cestu k souboru „Generátor budov.py“ a potvrdí. Dále zaškrtně pole, čímž se doplňek zapne, viz *obrázek číslo 16*. Pokud by uživatel chtěl, aby byl doplňek automaticky zapnutý po každém spuštění Blenderu, vybere

možnost „Save User Settings“. Po instalaci doplňku se po levé straně v okně „Tool Shelf“ objeví záložka s ovládacím panelem Generátoru budov, viz *obrázek číslo 17*. Pokud okno „Tool Shelf“ není vidět, stiskne uživatel klávesovou zkratku „T“ a okno se objeví.



Obr 16. „User Preferences“



Obr. 17 ovládací panel

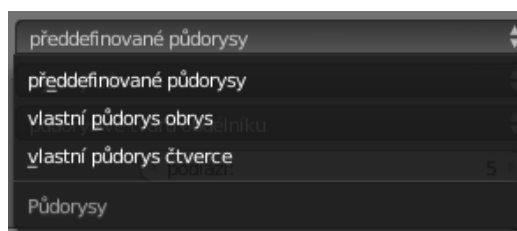
4.4 Ovládání

Generátor budov může generovat velké množství rozdílných budov a tudíž má mnoho parametrů, které si uživatel může nastavit. Každému parametru se při instalaci doplňku nastaví defaultní hodnoty, ze kterých může uživatel vycházet. Záložka s ovládacím panelem je naprogramovaná tak, aby uživateli nenabízela zbytečně možnosti navíc. Pokud si uživatel například vybere typ objektu „garáže“, záložka se změní a nabídne uživateli parametry, které je možné u tohoto typu objektu nastavit. Toto pravidlo platí pro všechny typy objektů. Pro „rodinný dům“ zmizí možnost „výstup na střechu a výtahová šachta“. Pokud uživatel nechce vymodelovat balkony, ovládací panel mu obratem zruší možnost nastavit parametry balkonů apod. Tím uživateli šetří čas a poskytuje rychlý přehled o parametrech daného objektu. Po nastavení parametrů je nutné kliknout na tlačítko „Generovat“, čímž se spustí proces modelování objektu. Ačkoli jsem se snažil vytvořit ovládací panel co nejvíce intuitivní, je nutné popsat parametry, aby mohl uživatel využít plný potenciál Generátoru budov.

4.4.1 Tvary půdorysů a typ objektu

Jako první parametr, se kterým se uživatel setká, je parametr „Půdorysy“, který je vidět na *obrázku číslo 18*. Zde si uživatel nastavuje půdorysy budovy. Vybrat si může půdorysy předdefinované nebo půdorys vlastní.

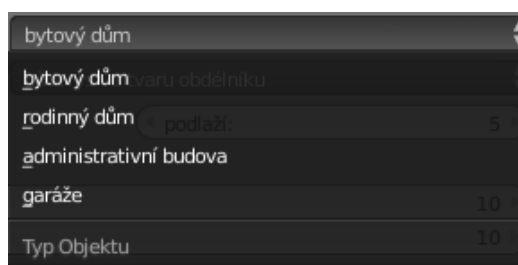
- „předdefinované půdorysy“ jsou nejpohodlnější možností, kde nemusí uživatel nic nahrávat, stačí si vybrat jeden z pěti předdefinovaných tvarů půdorysů.
- „vlastní půdorys obrys“ je možnost pro nahrání vlastního půdorysu splňující určitá pravidla, která jsou rozepsána v kapitole 4.5.1.
- „vlastní půdorys čtverce“ je složitější možností nahrání vlastního půdorysu splňující pravidla, která jsou podrobně popsána v kapitole 4.5.2.



Obr. 18 parametr „Půdorysy“

Po výběru jedné z možností si uživatel zvolí „Typ Objektu“ viz *obrázek číslo 19*. Uživatel si může vybrat ze tří respektive čtyř možností: „bytový dům“, „rodinný dům“, „administrativní budova“, a pokud uživatel nastavil parametr „Půdorysy“ na „předdefinované půdorysy“ má možnost vybrat si i „garáže“.

- „bytový dům“ je budova, která odpovídá svým vzhledem a parametry panelovým či cihlovým bytovým domům. První podlaží je tvořeno vchody s malými sklepními okénky.
- „rodinný dům“ se liší od bytového domu prvním podlažím, které je tvořeno vchodem, okny, vchodem na terasu, stěnami bez oken a volitelně garáží. Maximální počet podlaží je omezený na tři.
- „administrativní budova“ je charakteristická rovnou střechou, římsami a nemá balkony.
- „garáže“ jsou speciální objekt, který má zcela odlišné parametry, než ostatní budovy, podrobná ukázka se nachází v kapitole 4.4.8.



Obr. 19 parametr „Typ Objektu“

Parametr „Tvar půdorysu“ umožňuje uživateli vybrat tvar půdorysu z pěti předdefinovaných. Parametr je možné vidět na *obrázku číslo 20*.

- Půdorys ve tvaru obdélníku, písmene U, písmene H, písmene L a písmene T.



Obr. 20 parametr „Tvar půdorysu“

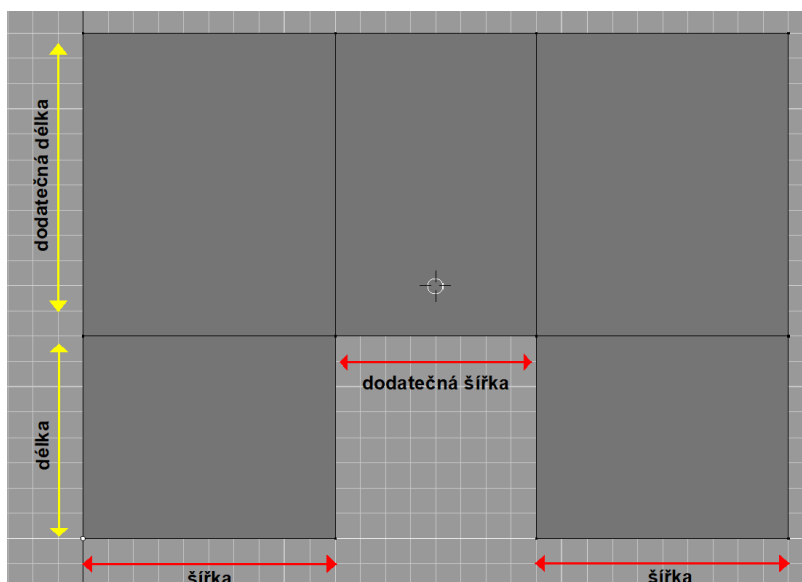
4.4.2 Rozměry

Parametr „počet podlaží“ je omezen minimální hodnotou 3 a maximální hodnotou 99 pro typ objektu „bytový dům“ a „administrativní budovu“ z důvodu náročného generování. Z mojí zkušenosti vím, že již padesát podlaží je pro počítač náročný úkol. Typ objektu „rodinný dům“ je omezena minimální hodnotou 1 a maximální hodnotou 3.

Počet podlaží:	podlaží: 5
Rozměry:	Dodatečné rozměry:
délka: 10	dodatečná délka strany: 10
šířka: 10	dodatečná šířka strany: 10

Obr. 21 parametry „Počet podlaží“, „Rozměry“ a „Dodatečné rozměry“

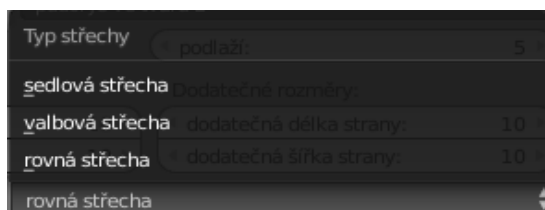
Pokud je vybraná možnost „předdefinované půdorysy“, v záložce se objeví parametr „Rozměry“, popřípadě „Dodatečné rozměry“, které slouží pro nastavení šířky a délky půdorysu. Parametry je možné vidět na *obrázku číslo 21*. Pokud si uživatel zvolí obdélník, je možné nastavit šířku a délku. U ostatních typů půdorysu je možné nastavit i „dodatečnou šířku“ a „dodatečnou délku“. „Dodatečná šířka“ a „dodatečná délka“ jsou rozměry, které určují velikost druhé části půdorysu, ze které se půdorys skládá. Nejlépe je to vidět na *obrázku číslo 22*.



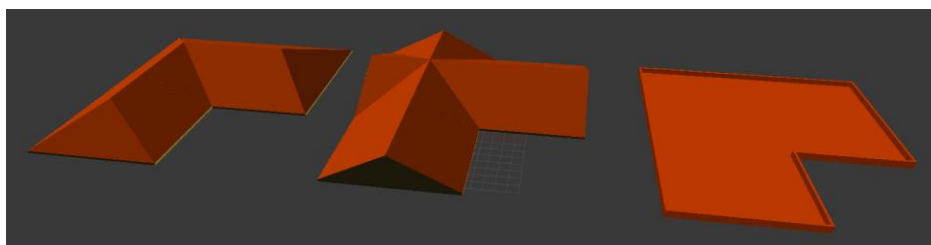
Obr. 22 Ukázka rozměrů

4.4.3 Střechy

Následně si uživatel pomocí parametru „Typ střechy“ vybírá ze tří možností střech: „rovná střecha“, „valbová střecha“ a „sedlová střecha“ viz *obrázek číslo 23*. Typy střech nejsou pro všechny objekty stejné. Administrativní budova může mít pouze střechu rovnou.



Obr. 23 parametr „Typ střechy“

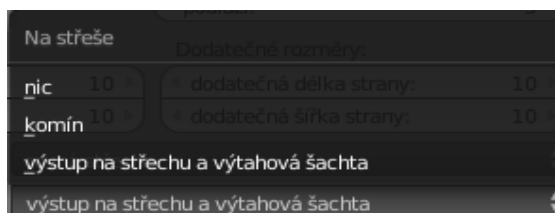


Obr. 24 ukázky typů střech, zleva valbová, sedlová, rovná(plochá)

Další parametr úzce souvisí s typem střechy. Jde o parametr „Na střeše“, který umožňuje uživateli výběr mezi možnostmi „výstup na střechu a výtahová šachta“, „komín“ a „nic“ viz *obrázek číslo 26*. Tento parametr je poměrně proměnlivý, protože ho ovlivňuje více ostatních parametrů. Například pokud není střecha rovná, nemůže si uživatel vybrat možnost „výstup na střechu a výtahová šachta“.



Obr. 25 parametr „Na střeše“ pro administrativní budovu

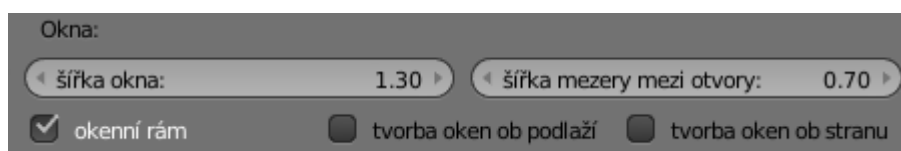


Obr. 26 parametr „Na střeše“ pro ostatní typy objektů

4.4.4 Okna

Několik dalších parametrů by se dalo zařadit do skupiny, která se týká oken a jejich rozmístění. Jsou to parametry, které mají velký vliv na vzhled budovy. Parametry je možné vidět na *obrázku číslo 27*.

- „šířka okna“ je zřejmý parametr, který udává šířku okna. Minimální hodnota je 1 a maximální hodnota 2.5.
- „šířka mezery mezi otvory“ udává šířku mezery mezi okny, dveřmi apod. Minimální hodnota je 0.3 a maximální hodnota 2. Výjimka je „administrativní budova“, kde je možné nastavit minimální hodnotu až na 0.1. U tohoto parametru je nutné vědět, že jedno okno se tvoří spojením mezery, okna a mezery podrobněji to vysvětlují v kapitole 4.6.
- „okenní rám“ je parametr pro tvorbu středního okenního rámu
- „tvorba oken ob podlaží“ znamená, že se okna vymodelují ob podlaží. Tudiž se budou střídát podlaží bez oken a podlaží s okny.
- „tvorba oken ob stranu“ znamená, že se okna vymodelují ob stranu na stejném principu jako parametr „Tvorba oken ob podlaží“. Při zaškrtnutí obou parametrů dojde na jednom podlaží ke střídání strany s okny a strany bez oken. Na druhém podlaží dojde ke stejnému střídání v opačném pořadí.



Obr. 27 skupina parametrů ovlivňující tvorbu oken

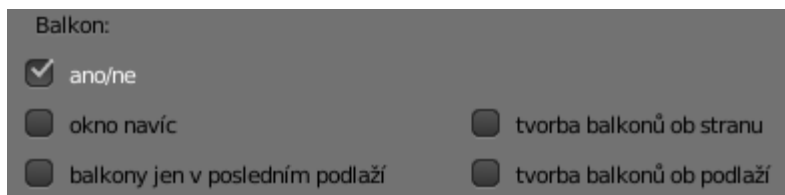
4.4.5 Balkony

Parametry, které mají také velký vliv na vzhled budovy, jsou kromě oken i balkony. Opět se jedná o pět parametrů, které se dají zahrnout do skupiny balkony a jsou zobrazeny na *obrázku číslo 28*.

- „ano/ne“ parametr jednoduše znamená vytvořit nebo nevytvořit balkony.
- „okno navíc“ je parametr, který vytvoří vedle balkonu ještě jedno okno. Výsledek na pohled působí více přirozeně.
- parametr „balkony jen v posledním podlaží“ modeluje balkony pouze v posledním podlaží

- „tvorba balkonů ob stranu“ funguje na stejném principu jako „tvorba oken ob stranu“, pouze se střídají strany s balkonem a strany s okny.
- „tvorba balkonů ob podlaží“ opět funguje obdobně jako „tvorba oken ob podlaží“, pouze se střídají podlaží s balkonem a podlaží s okny.

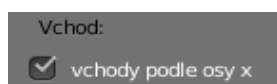
Při zaškrtnutí všech čtyřech parametrů, které mají v názvu „ob“ dojde ke střídání podlaží a stran s balkony a stran prázdných.



Obr. 28 skupina parametrů ovlivňující tvorbu balkonů

4.4.6 Přízemí

Pokud je vybrána možnost „bytový dům“ nebo „administrativní budova“ má uživatel přístup k parametru „vchod podle osy x“ viz *obrázek číslo 29*. Pokud je parametr zaškrtnutý, vchody se budou tvořit podle osy x s předpokladem, že se budova nachází v prvním kvadrantu kartézského souřadnicového systému. Pokud zaškrtnutý není, vchody se budou tvořit podle osy y.



Obr. 29 parametr „vchod podle osy x“

Pokud je ovšem vybrána možnost „rodinný dům“ parametr „vchod podle osy x“ není k dispozici a místo tohoto parametru se objeví dva nové parametry, které jsou vidět na *obrázku číslo 30*.

- parametr „garáž“, který po zaškrtnutí zajistí vymodelování minimálně jedné garáže (garážových vrat), pokud nebudou nastavené rozměry příliš malé.
- parametr „výměna stěn s okny za stěny bez oken a naopak“ je zvláštní parametr, který je dobré zaškrtnout jen v případě, pokud uživatel není spokojen s vymodelováním prvního podlaží. Parametr při příštím modelování vymění stěny s okny za stěny prázdné a naopak. Tímto může vzniknout příjemnější vzhled prvního podlaží.



Obr. 30 parametry pro rodinný dům

4.4.7 Barvy

Důležitou skupinou parametrů jsou barvy, které sice nemají vliv na samotné modelování, ale výrazně ovlivňují vzhled objektu. Uživatel si může vybrat možnost „předdefinované barvy“ nebo si nastavit své vlastní viz *obrázek číslo 31*. Části objektu, u kterých si uživatel může vybrat barvu, jsou střecha, fasáda, okna, okenní rám, komín, dveře, garáž, balkon a římsa. Pokud není parametr „předdefinované barvy“ zaškrtnut, objeví se parametry barev jednotlivých částí. Pokud uživatel chce zvolit jinou, například svou oblíbenou barvu a zná její RGB hodnoty, lze tuto barvu zvolit i pomocí těchto hodnot. Po kliknutí na barevný pruh se zobrazí barevný vzorník, který je možné vidět na *obrázku číslo 32*. Ze vzorníku si uživatel může vybrat jakoukoliv barvu, případně zadat její RGB hodnoty.



Obr. 31 parametry, pomocí kterých se nastavují barvy

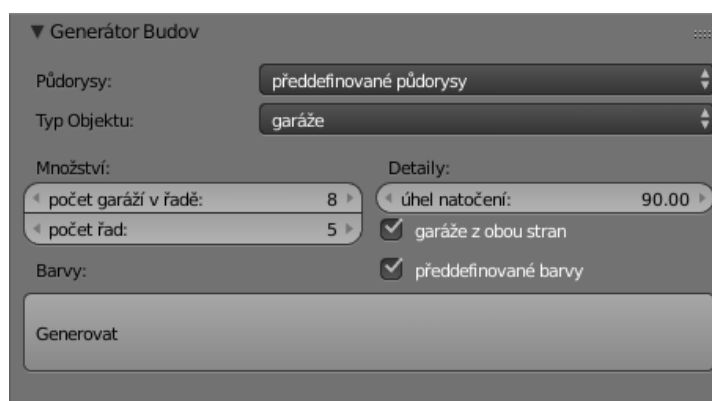


Obr. 32 barevný vzorník

4.4.8 Garáže

Posledními parametry jsou parametry již zmíněného objektu garáže. Kromě barev jsou zde čtyři parametry, které jsou vidět na *obrázku číslo 33*.

- „počet garáží v řadě“ určuje, kolik garáží se v jedné řadě vymodeluje.
- „počet řad“ určuje, kolik řad garáží se vymodeluje.
- „úhel natočení“ umožňuje garáže natočit vybraným směrem již při modelování.
- „garáže z obou stran“ je parametr, který zajišťuje modelování garáží v jedné řadě z obou stran. Garáže jsou tedy „zády“ k sobě.



Obr. 33 ovládací panel pro garáže

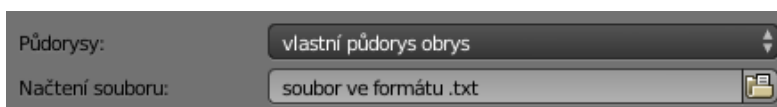
4.5 Nahrávání vlastního půdorysu

„Půdorysy předdefinované“ pokrývají díky možnosti nastavení rozměrů velkou škálu pravoúhlých půdorysů, ale nemohou pokrýt zdaleka všechny. Možnost nahrát si vlastní pravoúhlý půdorys tudíž výrazně rozšiřuje možnosti škály tvarů půdorysů, které je doplněk schopný vygenerovat. Teoreticky si uživatel může vygenerovat všechny pravoúhlé půdorysy, které ho napadnou. Generátor budov nabízí dvě možnosti, jak si nahrát svůj půdorys, mají ale určitá pravidla, kterými se musí uživatel řídit.

4.5.1 Vlastní půdorys obrys

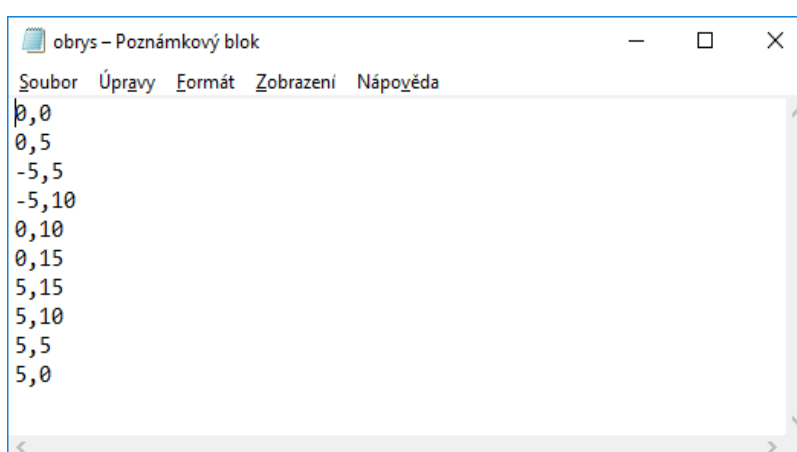
„Vlastní půdorys obrys“ je jednodušší varianta nahrání půdorysu. Uživatel si musí vytvořit textový soubor, do kterého zapíše souřadnice x a y jednotlivých bodů v kartézském souřadnicovém systému, ze kterých se půdorys skládá. Jak už jistě název parametru napověděl, jedná se o obrys půdorysu, tudíž body v textovém souboru musí tvořit pravoúhlý souvislý obrys. Body v souboru musí být řazeny postupně od prvního do posledního. Doporučuji začít prvním bodem se souřadnicemi 0,0. Nezáleží na tom,

jestli se body budou spojovat po směru nebo proti směru hodinových ručiček. Na každý řádek musí uživatel zapsat jeden bod ve tvaru: „souřadnice x“, „souřadnice y“. Pokud si uživatel zvolí tuto variantu, ovládací panel mu nabídne zadat cestu k souboru, viz *obrázek číslo 34*. Stačí kliknout na ikonu složky a program se přesune do adresáře, kde může uživatel soubor najít.



Obr. 34 parametr „Načtení souboru“

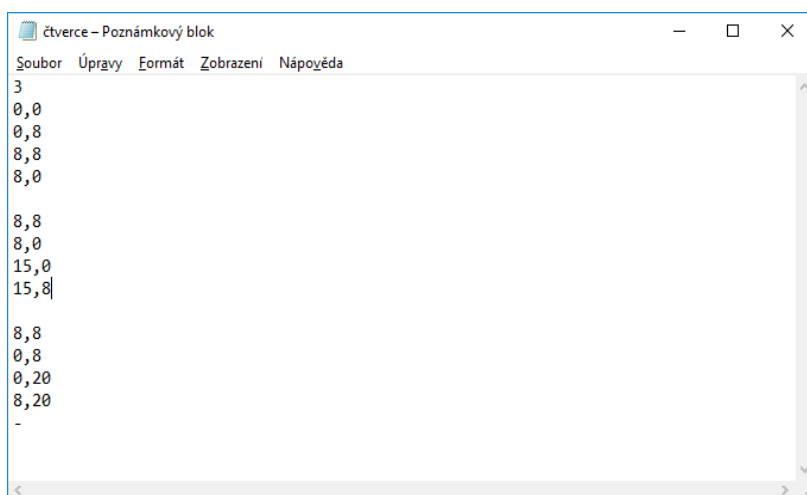
Tato varianta je poměrně jednoduchá, ale nese s sebou značné nevýhody. Při volbě této možnosti není možné zvolit „typ střechy“. Vždy se vytvoří „rovná střecha“ a nelze si vybrat její doplňky, jako je „komín“ či „výstup na střechu a výtahová šachta“. Poslední nevýhodou je, že nelze tvořit půdorysy s uzavřeným atriem, protože takový půdorys nemá souvislý obrys. Proč při této možnosti nelze zvolit typ střechy je popsáno v kapitole 4.2.2., kde jsem se věnoval modelování střech. Je dobré si uvědomit, že pokud je zadán vlastní půdorys, nelze si vybrat parametr „Rozměry“. Rozměry si uživatel sám nastavuje tím, jak daleko od sebe body definuje. Toto je výhoda oproti předdefinovaným půdorysům, jelikož si uživatel může nastavit každou stěnu budovy s různými rozměry. Stěna budovy je vzdálenost dvou po sobě jdoucích bodů. Ukázkový textový soubor se nachází na přiloženém CD „obrys.txt“.



Obr. 35 ukázka zápisu obrysu půdorysu, kde mají všechny strany délku 5

4.5.2 Vlastní půdorys čtverce

„Vlastní půdorys čtverce“ je složitější varianta nahrání souboru. Půdorys se skládá ze čtverců nebo obdélníků, ze kterých si uživatel poskládá výsledný pravoúhlý půdorys. Všechny čtverce či obdélníky musejí mít společné vrcholy se všemi sousedícími čtverci či obdélníky a nesmí existovat vrchol, který je společný právě pro čtyři čtverce či obdélníky. Toto je velice důležité pravidlo pro možnost vytvoření sedlové či valbové střechy, komínu apod., viz kapitola 4.2.2, kde jsem se věnoval procesu modelování střech. Pro využití této možnosti si uživatel musí taktéž vytvořit textový soubor se souřadnicemi x a y jednotlivých vrcholů, podobně jako v předchozí variantě, avšak jeho struktura je odlišná. Na prvním řádku musí být napsaný počet čtverců a obdélníků, ze kterých se půdorys tvoří. Další čtyři řádky jsou souřadnice vrcholů, které tvoří jeden čtverec či obdélník, ve tvaru: „souřadnice x“, „souřadnice y“ například 0,0. Pátý řádek bude volný, další čtyři řádky budou opět souřadnice vrcholů dalšího čtverce či obdélníku. Takto se musí zapsat všechny čtverce či obdélníky a na úplně posledním řádku (řádek za poslední souřadnicí) libovolný znak, například pomlčka viz *obrázek číslo 36*. Obrisy čtverců či obdélníků mohou být zapisovány po směru či proti směru hodinových ručiček. Tato možnost eliminuje nevýhody, které má možnost „Vlastní půdorys obrys“, avšak je náročnější na zápis. Je možné si zde vybrat mezi třemi „typy střech“, vytvořit půdorys s atriem a vymodelovat „komín“ či „výstup na střechu a výtahovou šachtu“. Stejně jako v předešlé možnosti i tady si rozměry definuje sám uživatel při zápisu souřadnic. Ukázkový textový soubor se nachází na příloženém CD „čtverce.txt“.



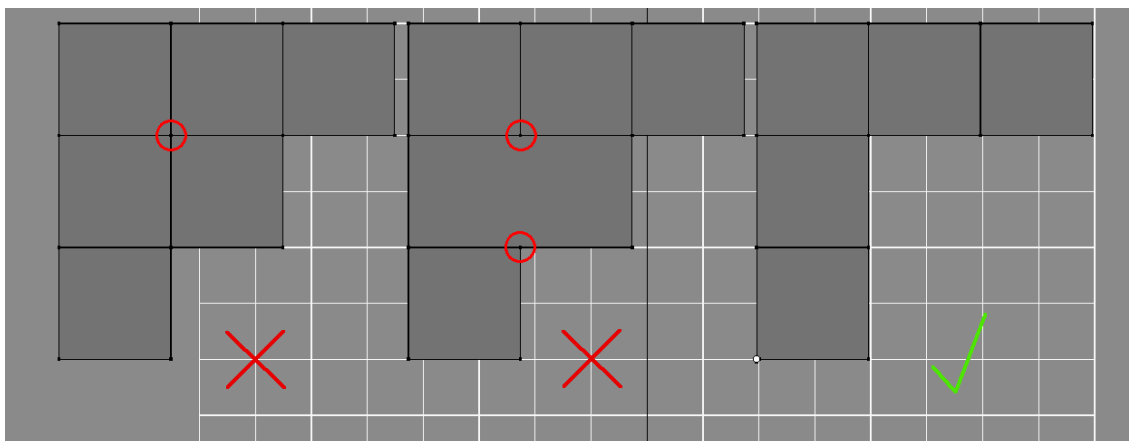
```
čtverce - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
3
0,0
0,8
8,8
8,0

8,8
8,0
15,0
15,8

8,8
0,8
0,20
8,20
-
```

Obr. 36 ukázka zápisu půdorysu, který se skládá ze čtverce a dvou obdélníků

Ani tato metoda ale nenabízí možnost realizace šikmých střeš pro veškeré tvary půdorysů. Již dříve zmíněné pravidlo neumožňuje tvorbu složitých půdorysů, jelikož nesmí žádný vrchol sousedit se čtyřmi čtverci či obdélníky, ale zároveň sousedící čtverce či obdélníky musí mít společné body. Nejlépe je to vidět na *obrázku číslo 37*.



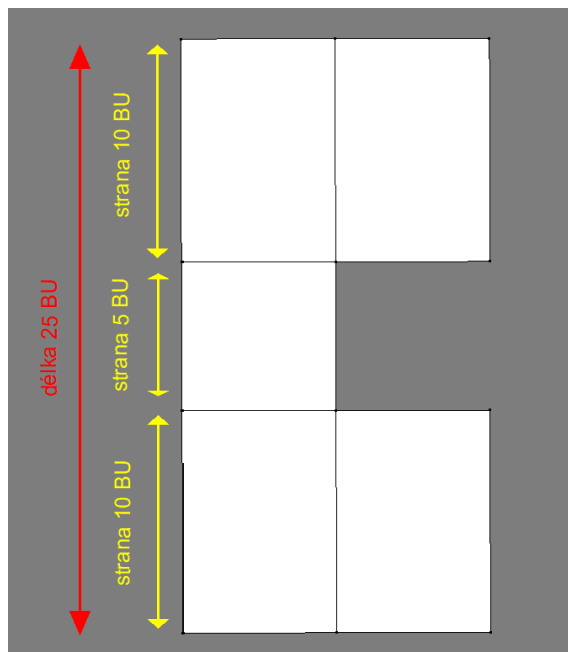
Obr. 37 ukázka dvou půdorysů, které porušují pravidla a jednoho, který pravidla splňuje

Pokud se stane, že nějaký vrchol sousedí se čtyřmi čtverci či obdélníky, tak v tom místě vznikne nesmyslné úžlabí ze všech směrů, což je samozřejmě špatně. Pro tyto půdorysy je proto vhodné zvolit rovnou střechu. U takto členitých půdorysů je velmi složité, či nereálné šikmou střechu vytvořit i ve skutečnosti. Jedno z možných řešení by mohlo být aplikování metody „Straight skeleton“, která uvnitř polygonu pomocí procesu smršťování hran polygonu najde osu. Každý bod, kterým je osa tvořena, je spojený přímkou se svým originálem, který tvoří hranice polygonu. Tím dochází k rozdělení polygonu na jednotlivé části střechy (Aichholzer et al. 1995). Toto řešení již není předmětem bakalářské práce, ale mohlo by se stát předmětem práce diplomové.

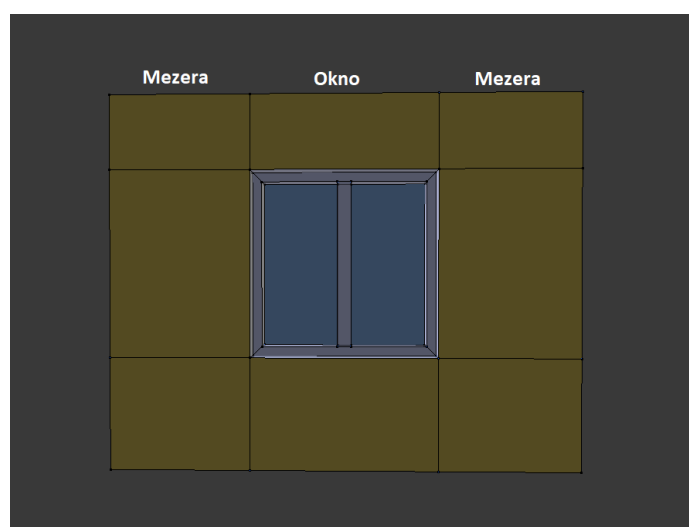
4.6 Doporučení k ovládání

Všechny půdorysy, kromě půdorysu vytvořeného pomocí možnosti „vlastní půdorys obrys“, si musí uživatel představovat jako polygon složený z obdélníku a čtverců. Tento fakt je důležitý, když si uživatel definuje nebo vybírá rozměry. Strana, která se může na pohled jevit jako jedna, se ve skutečnosti může skládat z několika stran (strany jednotlivých čtverců či obdélníků). Je tedy možné, že se po zadání parametrů nevyodelují prvky, které mají větší rozměry než jednotlivé strany. Nejlépe je to vidět na *obrázku číslo 38*.

Dále je dobré vědět, že rozměry detailních prvků si ke svému rozměru přičítají minimálně dvakrát hodnotu mezery. Pokud je šířka okna 2 BU a šířka mezery 1 BU, tak minimální velikost strany pro vymodelování okna je 4 BU (mezera + okno + mezera) viz *obrázek číslo 39*. To platí pro všechny detailní prvky, jako jsou balkony, dveře apod. Tento princip je využit, aby žádný detailní prvek nesousedil s koncem strany. Je tedy vyloučeno, aby otvory byly umístěny v rohu objektu.



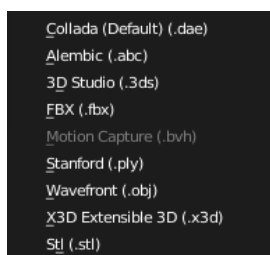
Obr. 38 strana složená ze tří různých stran



Obr. 39 složení okna

4.7 Import do softwaru ArcGIS Desktop

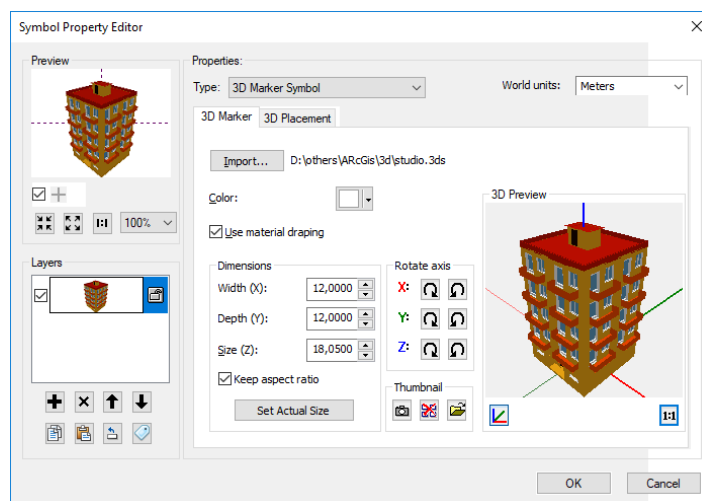
Po vygenerování objektu už jen zbývá exportovat vygenerovaný objekt. Blender nabízí několik možností exportu. Pro účely 3D vizualizace je potřeba pracovat v aplikaci ArcScene, kterou ArcGIS Desktop nabízí. Nejdůležitější formáty exportu jsou Collada, která má koncovku .dae a 3D Studio s koncovkou .3ds. Oba tyto formáty jsou kompatibilní s programem ArcGIS Desktop. Doporučuji exportovat ve formátu 3D Studia, jelikož mi přijde více kompatibilní než Collada. Z vlastní zkušenosti mi formát Collada občas v náhledu, který nabízí „Marker Symbol“ nevykresluje celou geometrii modelu. Exportovat objekt je v Blenderu možné pomocí ikony File → Export → 3D Studio (.3ds) popřípadě si uživatel může vybrat jiný formát.



Obr. 40 možné formáty exportu

Po úspěšném vyexportování objektu, je zapotřebí objekt naimportovat do aplikace ArcScene. Zobrazit objekt je možné několika způsoby.

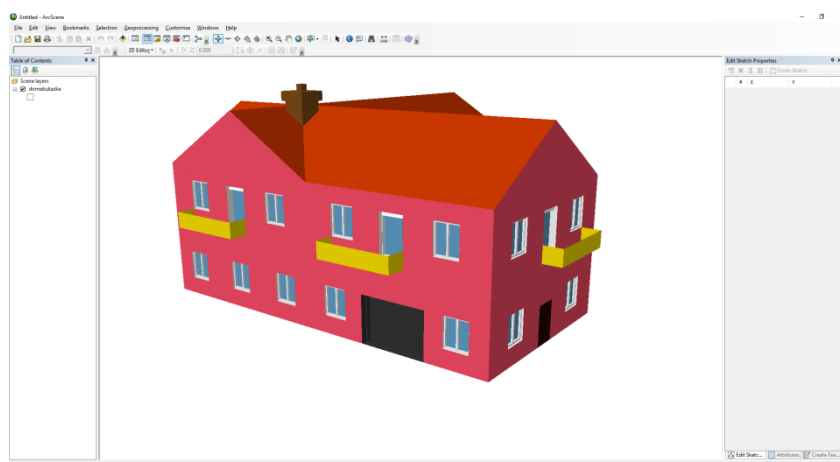
První způsob je nahrát objekt jako „Marker Symbol“ v okně „Style Manager“. Tím si uživatel vytvoří vlastní symbol, kterým může reprezentovat bod ve 3D viz *obrázek číslo 41*. Pokud si po nahrání uživatel vybírá symbol, který má reprezentovat bod ve 3D,



Obr. 41 „Marker Symbol“

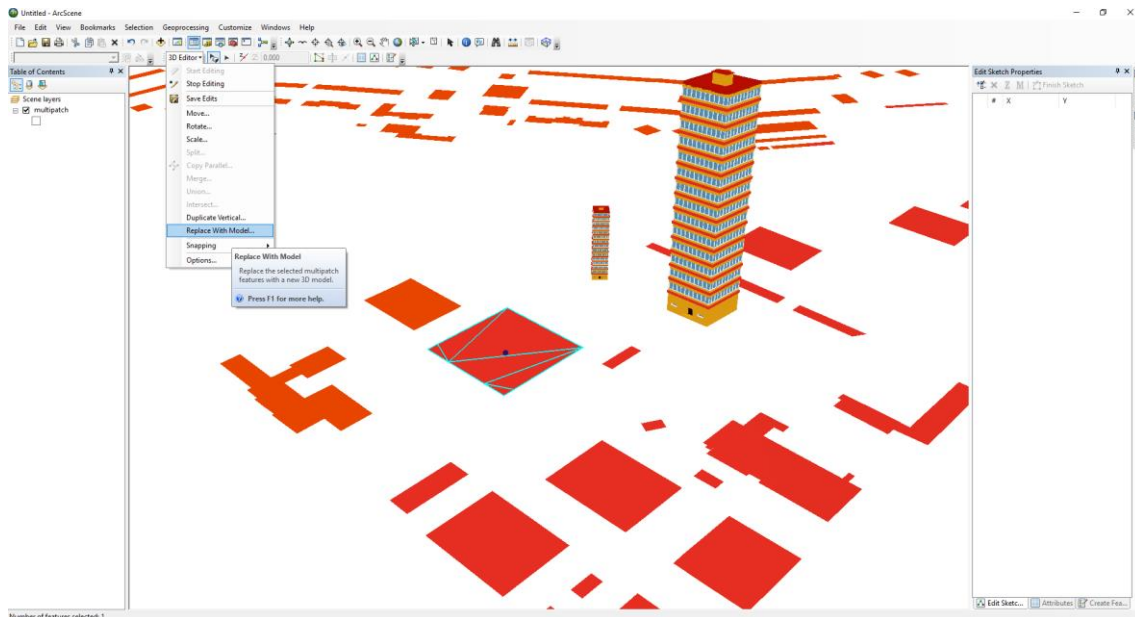
automaticky mu aplikace ArcScene nabídne jeho vytvořený symbol, v tomto případě exportovaný objekt z Blenderu.

Druhý způsob jak importovat objekt do aplikace ArcScene je využití funkce „Import 3D Files“, která vytvoří z nahraného 3D objektu „multipatch feature class“. Multipatch feature class reprezentuje 3D objekty, jak je možné vidět na *obrázku číslo 42*. Nevýhodou tohoto způsobu jsou souřadnice importovaného objektu. Jelikož Blender pracuje v kartézském souřadnicovém systému, při importu do aplikace ArcScene objekt nemá žádný souřadnicový systém.



Obr. 42 Multipart feature class ukázka

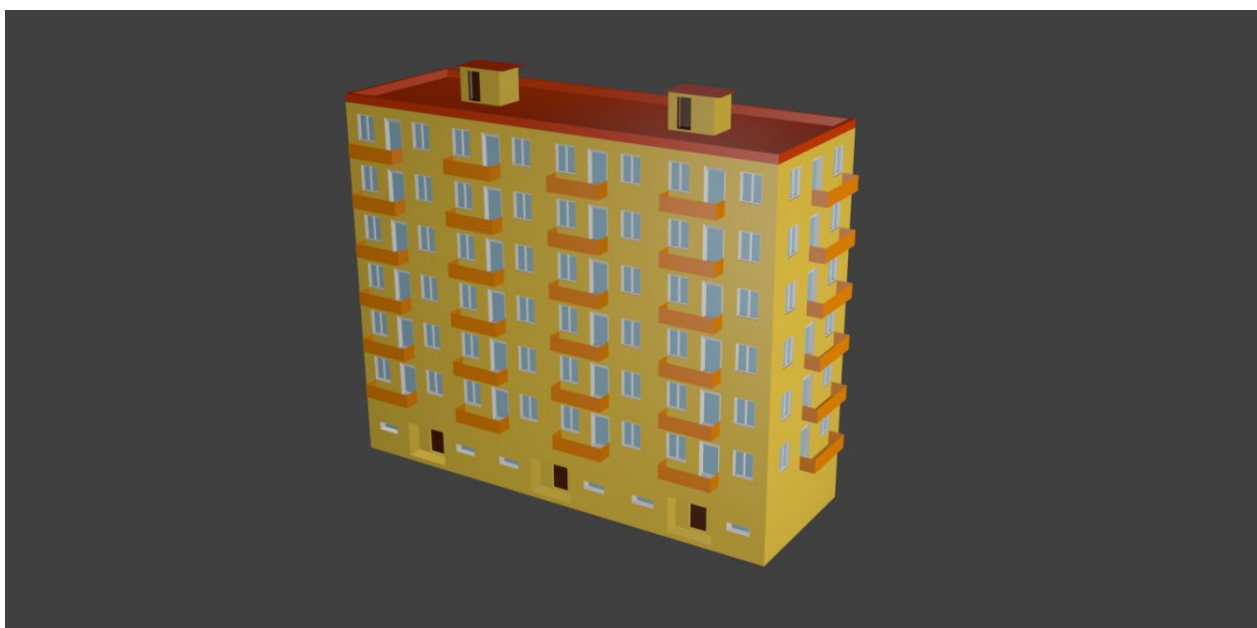
Třetí způsob vyžaduje již vytvořenou multipatch feature class. Multipatch feature class si uživatel vytvoří z polygonů, které chce nahradit vytvořenými objekty z Blenderu. Funkce „Interpolate Polygon to Multipatch“ převede polygonovou vrstvu na multipatch feature class. Dále je nutné zapnout „edit mode“ a označit polygon (který je součástí multipatch feature class), který si uživatel přeje nahradit. Poté stačí použít nástroj „Replace With Model...“ a vybrat objekt, který nahradí označený polygon. Dále je možné změnit aktuální velikost modelu pomocí nástroje v editoru „scale“ nebo model natáčet pomocí „rotate“. Použití nástrojů scale a rotate je vidět na *obrázku číslo 43*. V Blenderu je možné vymodelovat více budov, rozmístit si je podle potřeby a následně je vyexportovat. Tím vznikne jeden soubor, který v sobě má uloženo několik budov. Pokud bude soubor nahrán například jako „Marker Symbol“, zobrazí se všechny budovy, které si uživatel vymodeloval v Blenderu, jako reprezentace jednoho bodu. Tento fakt je možné využít k vymodelování města nebo větších obytných území.



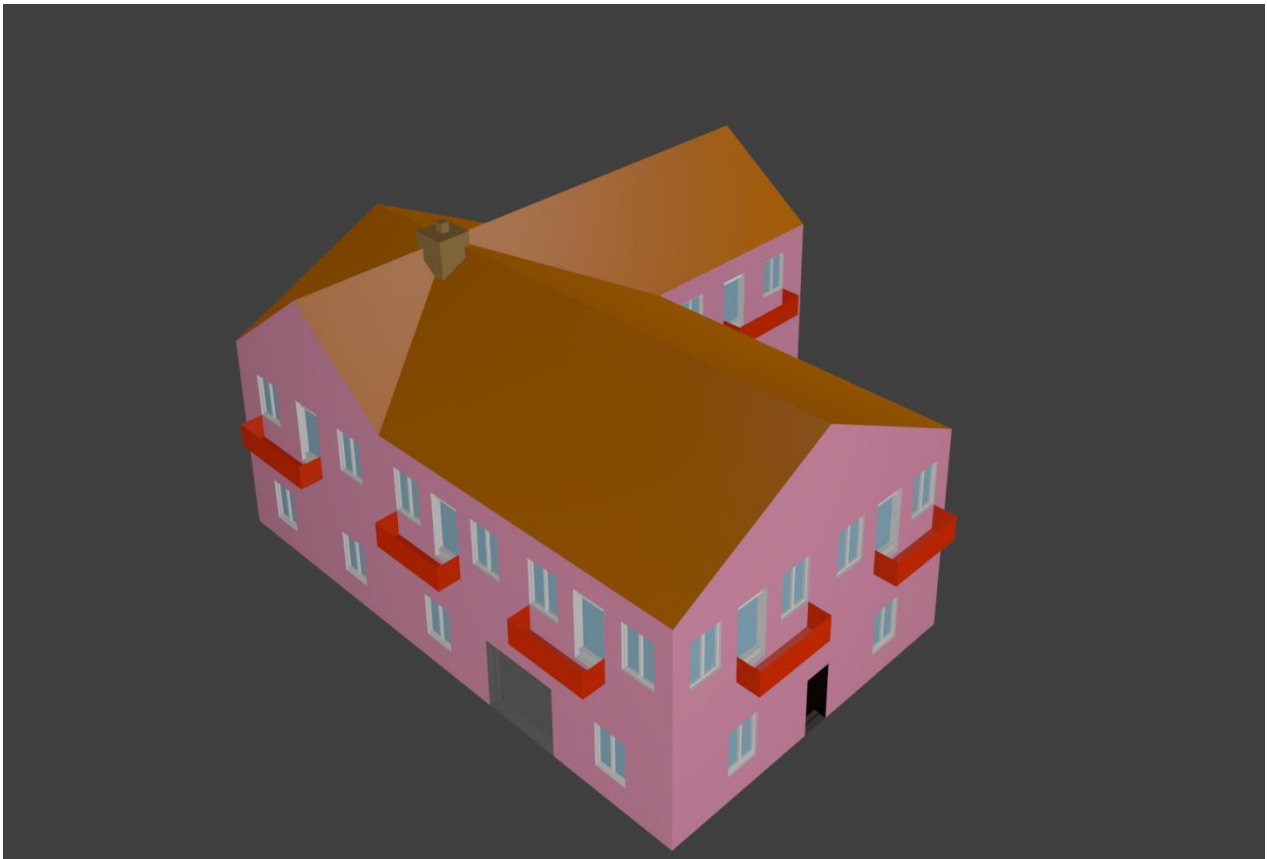
Obr. 43 použití nástroje „Replace With Model...“

4.8 Ukázky vygenerovaných budov

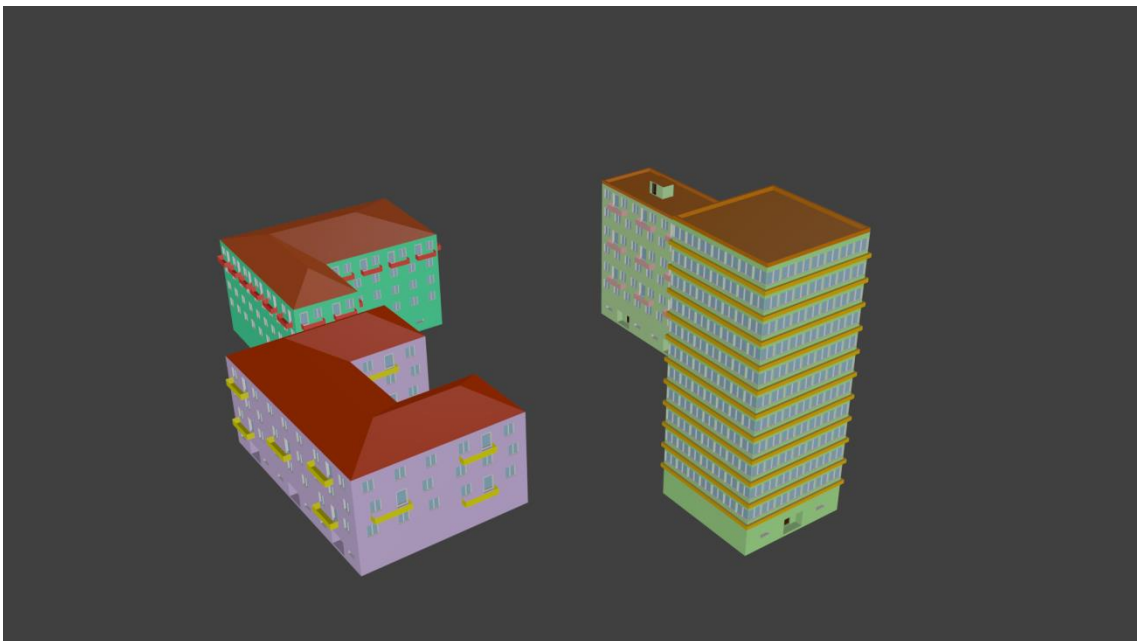
V této kapitole bych chtěl poukázat na různorodost budov, které se dají vymodelovat generátorem budov. Stačí si jen „pohrát“ s parametry. Výsledky je možné vidět níže. Další ukázky je možné najít v přílohách.



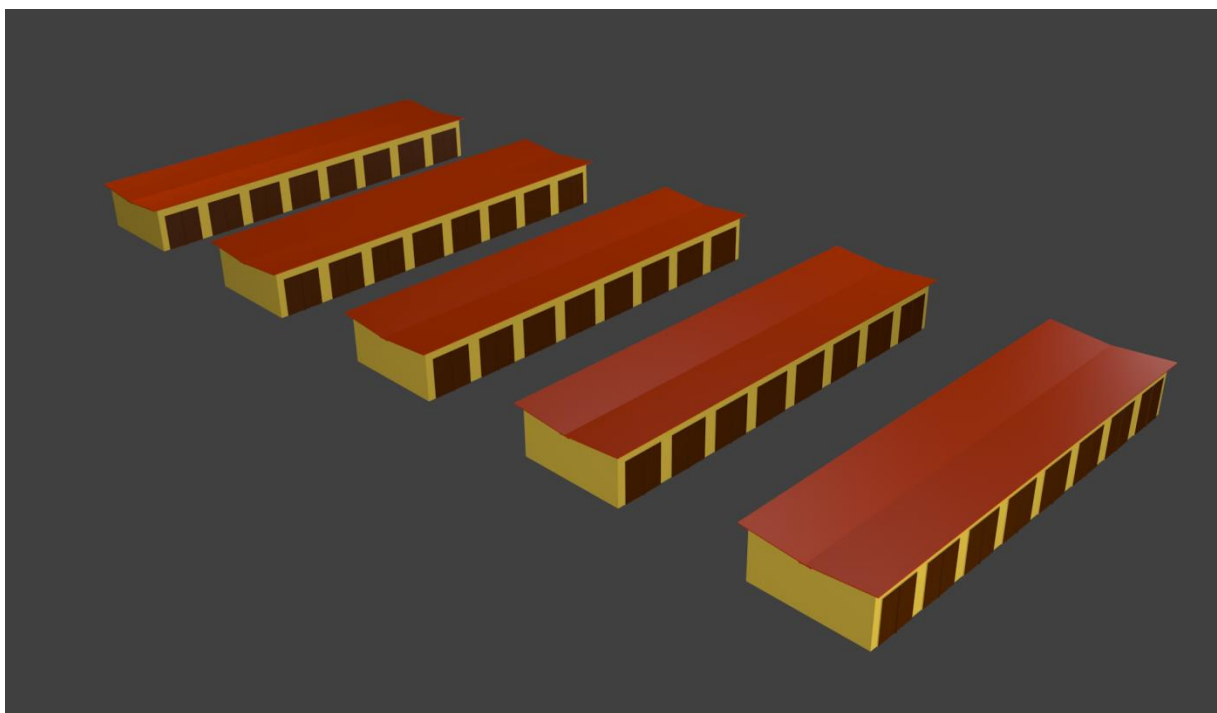
Obr. 44 bytový dům



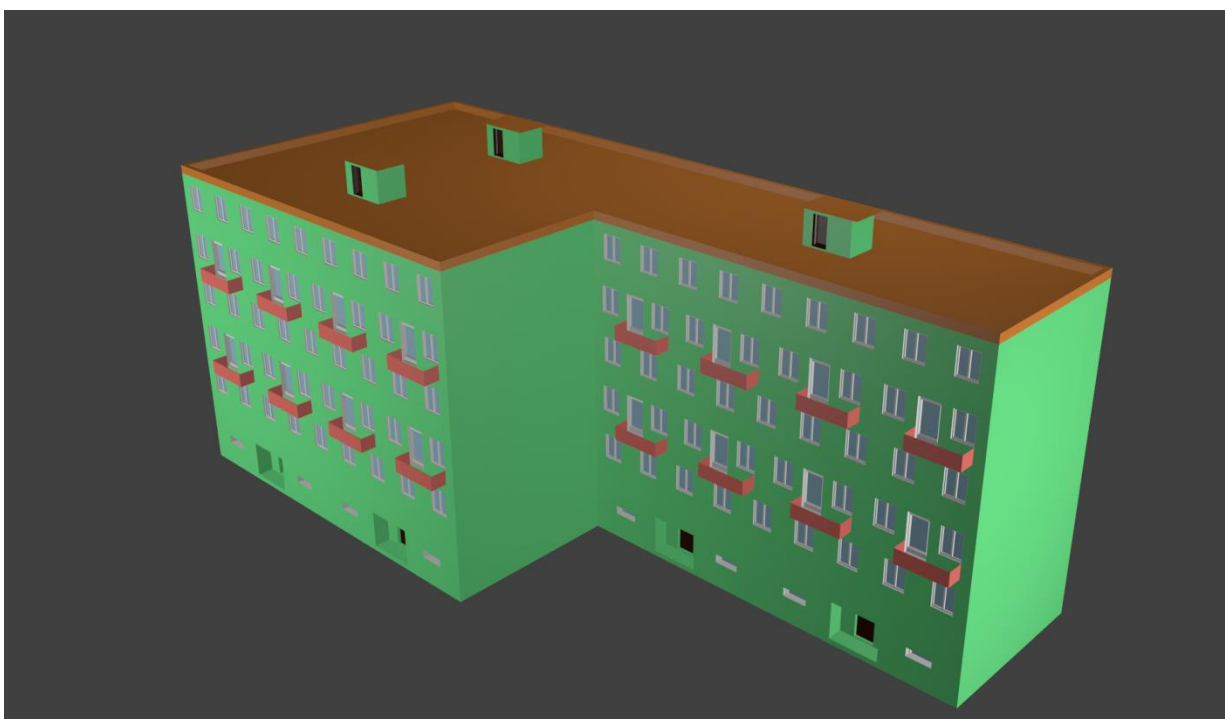
Obr. 45 rodinný dům s více bytovými jednotkami a sedlovou střechou



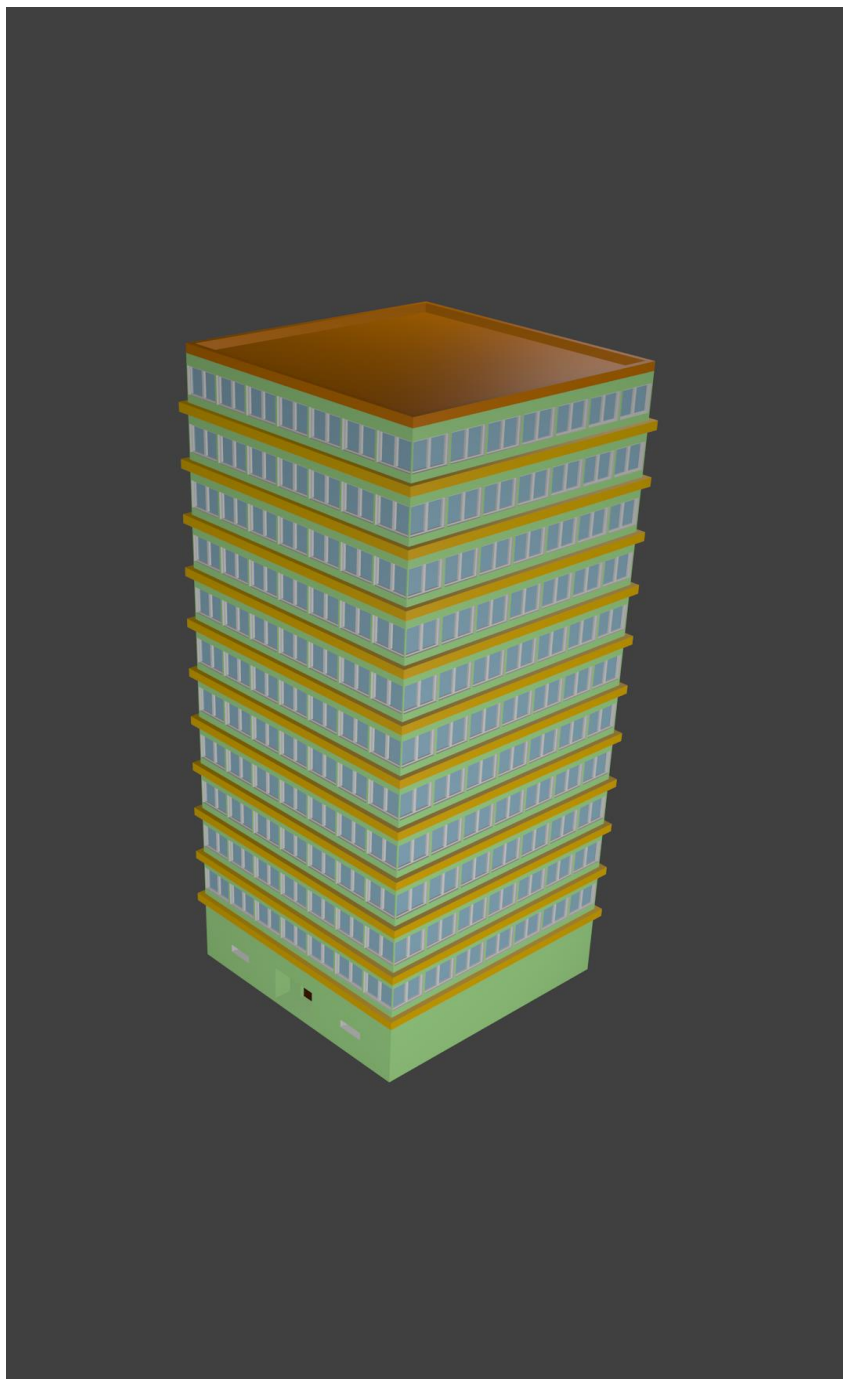
Obr. 46 budovy



Obr. 47 řadové garáže



Obr. 48 bytový dům 2



Obr. 49 administrativní budova

5 Diskuze

V této kapitole zhodnotím výsledky práce a srovnám je s již existujícím programem CityEngine, který jsem stručně představil v kapitole 2.3.1. 3D modelovací software CityEngine je v oblasti geografie jeden z nejznámějších a je velice propracovaný. Proto jsem se rozhodl generátor budov porovnat právě s tímto softwarem.

Nelze srovnávat najednou oba programy ve všech ohledech, proto vezmu postupně jednotlivé oblasti a pokusím se je porovnat. Jako první porovnáám dostupnost softwarů. Dostupnost je největší výhodou mého generátoru budov, jelikož je psaný v open source softwaru Blender, který si může každý stáhnout zdarma. Oproti tomu CityEngine je licencovaný software, který není zrovna nejlevnější. Podle anglické wikipedie stojí licence pro CityEngine cca 500\$. Pro malé firmy nebo pro jednotlivce je cena takové licence samozřejmě značný výdaj. Je tedy jasné, že můj generátor budov je dostupnější než software CityEngine.

Ovládání je další důležitou oblastí, která hraje roli hlavně pro nové uživatele. Složitost ovládání dokáže uživatele po neúspěchu lehce odradit. Generování budov v softwaru CityEngine je ovládáno sadou pravidel, které uživatel zapisuje v podobě jazyka CGA gramatiky. Pravidla CGA gramatiky jsou pro nového uživatele velmi neintuitivní a složitá na zápis, zatímco můj generátor používá k ovládání jednoduchý panel, kde si uživatel jen vybírá z možností parametrů nebo zaškrťává parametry. Na základě těchto argumentů si troufám tvrdit, že ovládání mého generátoru je jednodušší než ovládání softwaru CityEngine.

Generování 3D modelů je účelem těchto programů a oblastí, kde CityEngine velmi vyniká. CityEngine dokáže generovat celá města s ulicemi, parcelami, budovami i stromy. Variabilita budov, které CityEngine dokáže generovat je téměř nekonečná. Na druhou stranu můj generátor generuje jen modely budov a rozhodně nemají tak enormní variabilitu, jako budovy softwaru CityEngine. V této oblasti tedy CityEngine výrazně převyšuje můj generátor budov.

Malou nevýhodou softwaru CityEngine je generování specifického typu budovy, jelikož tento software je určený spíše pro generování celých měst a kvůli složitému ovládání není tato činnost jednoduchá. V manuálním modelování nabízí CityEngine velkou škálu editačních a modelovacích nástrojů, zatímco po vygenerování modelu již nenabízí

širokou možnost editace. Tudíž vygenerované modely si uživatel manuálně upraví jen málo. Zatímco generátor budov pracuje v Blenderu, a tudíž má uživatel po vygenerování budovy přístup k veškeré geometrii a lze si jí libovolně editovat. Tento fakt může sloužit uživateli, pokud si chce vytvořit specifickou budovu nebo upravit stávající model. Použije kostru vygenerované budovy a manuálně si dotvoří zbytek. Dále si uživatel může upravovat zdrojový kód a přidávat nové typy budov či fasád nebo si jen lehce upravit některé fasády. Upravovat zdrojový kód je pro uživatele samozřejmě náročné, ale tuto možnost lze využít pro větší projekty, jako například vytvoření modelů budov pro rekonstrukci zaniklých vesnic apod.

Výše zmíněné argumenty nahrávají ve prospěch mého generátoru budov. Nicméně generátor budov je postavený právě na nedostacích softwarů jako je CityEngine. Jestliže uživatel perfektně ovládá CityEngine, dokáže využít potenciál tohoto softwaru na maximum a vytvářet další obsah, který jsem v této práci nepopsal.

Generátor budov je dobrý pomocník při tvorbě 3D mapy části města či vesnice. Uživatel si může vygenerovat modely budov, zkombinovat je s modely, které nabízí vybraný geoinformační software a tím vytvořit věrohodné 3D prostředí.

Je možné, že v budoucnu v diplomové práci se ke generátoru budov vrátím a zkusím tuto aplikaci vylepšit. Možné vylepšení by mohlo být využití algoritmu „Straight skeleton“ pro tvorbu střech nebo rozšíření samotného generátoru budov pro konkrétní aplikaci.

6 Závěr

Cílem práce bylo vytvořit generátor budov, který umožní uživateli generovat 3D modely budov. Tyto modely je následně možné využít pro 3D vizualizaci budov v geoinformačních softwarech. Hlavními aspekty práce bylo využití open source softwaru pro tvorbu generátoru a vytvoření jednoduchého a intuitivního ovládání generátoru budov. Pro tvorbu jsem využil open source software Blender, který jsem ve své práci představil. Generátoru budov jsem věnoval větší část práce, kde jsem popsal teoretický základ, na kterém jsem generátor postavil, funkcionalitu programu a ovládání. Na závěr jsem demonstroval dosažené výsledky několika ukázkami a vytvořil jednoduchý návod, jak vytvořené 3D modely importovat do geoinformačního softwaru ArcGIS Desktop.

Cíle mé práce byly splněny a generátor budov může být volně využíván veřejností při 3D vizualizaci budov.

Seznam použitých zdrojů informací

- AICHHOLZER, Oswin, Franz AURENHAMMER, David ALBERTS a Bernd GÄRTNER, 1995. A novel type of skeleton for polygons. *Journal of Universal Computer Science* [online]. **1**(12), 752–761. Dostupné z: doi:10.3217/jucs-001-12-0752
- BLENDER FOUNDATION. *Blender.org - Home of the Blender project - Free and Open 3D Creation Software* [online]. 2016 [cit. 2018-04-21]. Dostupné z: <https://www.blender.org/>
- BLENDER FOUNDATION. *Blender Reference Manual - Blender Manual* [online]. 2016 [cit. 2018-04-21]. Dostupné z: <https://docs.blender.org/manual/en/dev/>
- BRŮHA, Lukáš, 2017. Modely prostorových dat v GIS : od 2D k 3D. *Geografické rozhledy*. 18–19. ISSN 1210-3004.
- ESRI R&D CENTER ZURICH. *CityEngine Help* [online]. 2013 [cit. 2018-04-23]. Dostupné z: <https://cehelp.esri.com/help/index.jsp>
- CHOMSKY, Noam, 1956. Three models for the description of natural language. *I. R. E. Transactions on Information Theory* [online]. **IT-2**, 113–124. ISSN 0018-9448. Dostupné z: doi:10.1109/TIT.1956.1056813
- JAWORSKI, Witold, 2011. *Programming Add-Ons for Blender 2.5*. ISBN 978-83-931754-2-0.
- MÜLLER, Pascal, Peter WONKA, Simon HAEGLER, Andreas ULMER a Luc VAN GOOL, 2006. Procedural modeling of buildings. *ACM Transactions on Graphics* [online]. **25**, 614–623. ISSN 07300301. Dostupné z: doi:10.1145/1179352.1141931
- PARISH, Yoav I. H. a Pascal MÜLLER, 2001. Procedural modeling of cities. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01* [online]. 301–308. Dostupné z: doi:10.1145/383259.383292
- POKORNÝ, Pavel, 2009. *Blender - Naučte se 3D grafiku*. 2. vyd. B.m.: BEN. ISBN 80-7300-244-2.
- PRUSINKIEWICZ, Przemyslaw, 1986. Graphical Applications of L-Systems. *Graphics Interface* [online]. (1986), 247--253. ISSN 07135424. Dostupné

z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.4242&rep=rep1&type=pdf>

PRUSINKIEWICZ, Przemysław a Aristid LINDENMAYER, 1990. *The algorithmic beauty of plants* [online]. B.m.: Springer. ISBN 9780387946764. Dostupné

z: <http://linkinghub.elsevier.com/retrieve/pii/S0168945296045268>

PYTHON SOFTWARE FOUNDATION. 26.2. *pdb — The Python Debugger* [online]. 2007 [cit. 2018-04-21]. Dostupné z: <https://docs.python.org/2/library/pdb.html>

SMELIK, R. M., T. TUTENEL, K. J. DE KRAKER a R. BIDARRA, 2011. A declarative approach to procedural modeling of virtual worlds. *Computers and Graphics (Pergamon)* [online]. **35**(2), 352–363. ISSN 00978493. Dostupné

z: doi:10.1016/j.cag.2010.11.011

SMELIK, Ruben M., Tim TUTENEL, Rafael BIDARRA a Bedrich BENES, 2014. A survey on procedural modelling for virtual worlds. *Computer Graphics Forum* [online].

33(6), 31–50. ISSN 14678659. Dostupné z: doi:10.1111/cgf.12276

STINY, George, 1975. *Pictorial and Formal Aspects of Shape and Shape Grammars: On Computer Generation of Aesthetic Objects* [online]. ISBN 978-3-7643-0803-2.

Dostupné z: doi:10.1007/978-3-0348-6879-2

VAN LEEUWEN, J, 1990. *Handbook of Theoretical Computer Science, Volume B Formal Models and Semantics*. B.m.: Elsevier. ISBN 9780262220392.

WONKA, Peter, Michael WIMMER, François SILLION a William RIBARSKY, 2003. Instant architecture. *ACM Transactions on Graphics* [online]. **22**(3), 669.

ISSN 07300301. Dostupné z: doi:10.1145/882262.882324

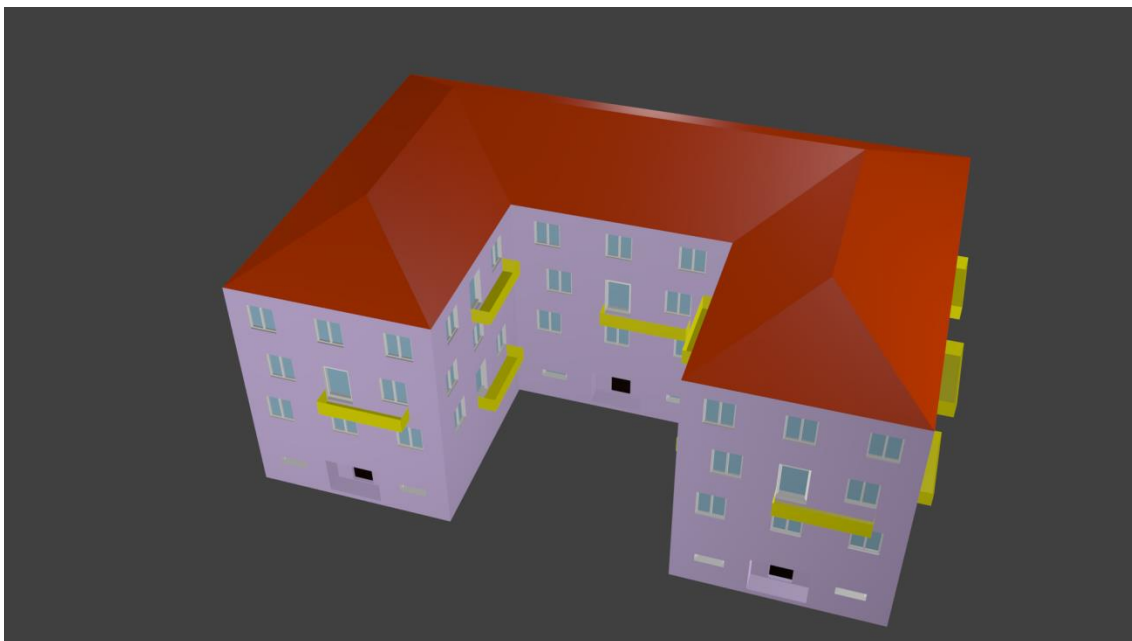
YI, Cheng, Yuan ZHANG, Qiaoyun WU, Yabin XU, Oussama REMIL, Mingqiang WEI a Jun WANG, 2017. Urban building reconstruction from raw LiDAR point data. *CAD Computer Aided Design* [online]. B.m.: Elsevier Ltd, **93**, 1–14. ISSN 00104485.

Dostupné z: doi:10.1016/j.cad.2017.07.005

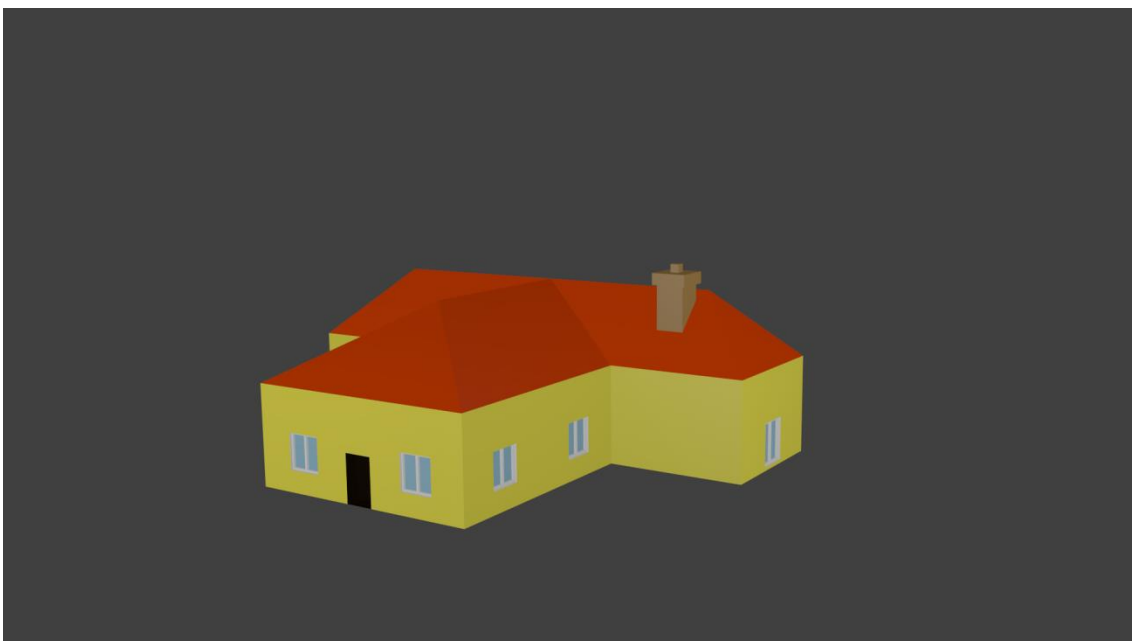
Seznam příloh

- Příloha 1 Ukázky vygenerovaných budov pomocí generátoru budov
- Příloha 2 CD s elektronickou verzí práce, se skriptem, který obsahuje generátor budov a se dvěma textovými soubory, které obsahují souřadnice pro nahrání půdorysu

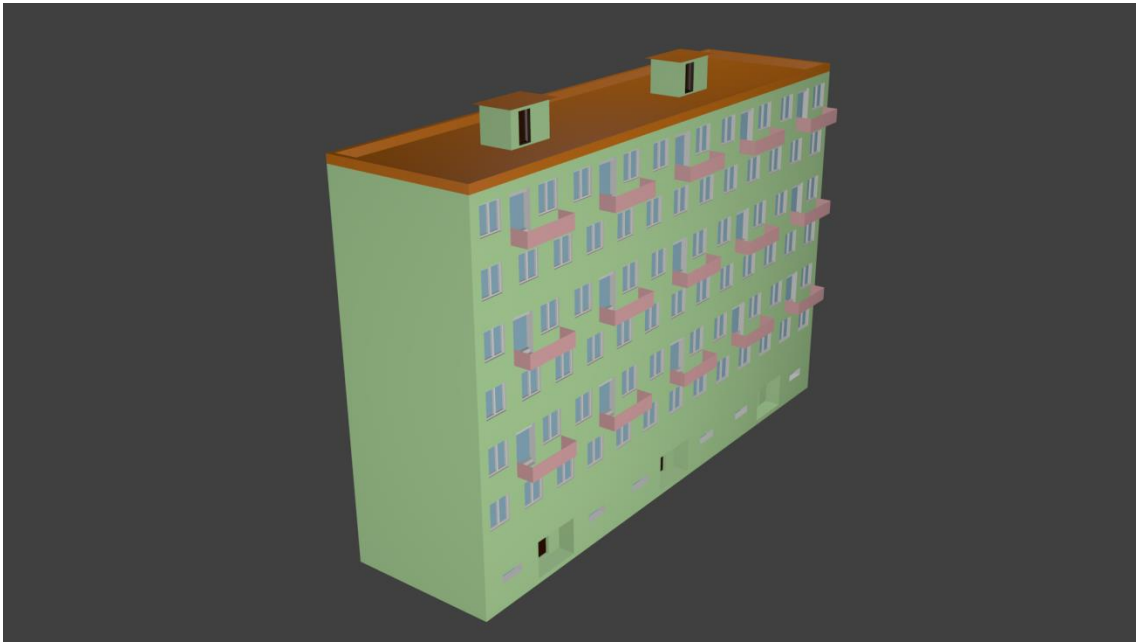
Příloha 1



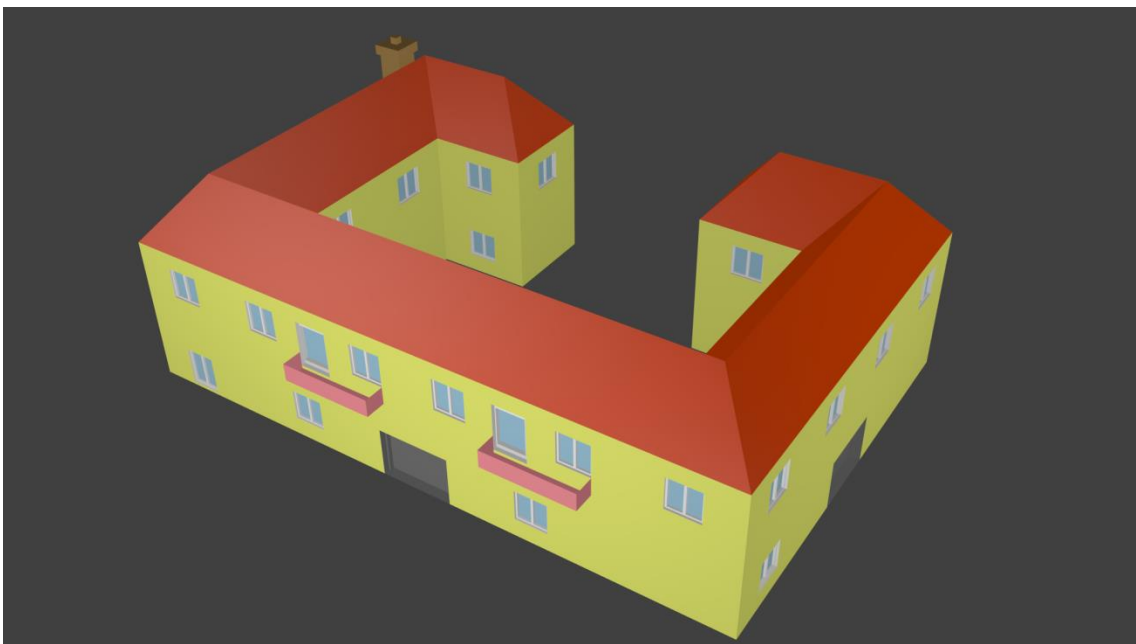
Bytový dům s půdorysem do U



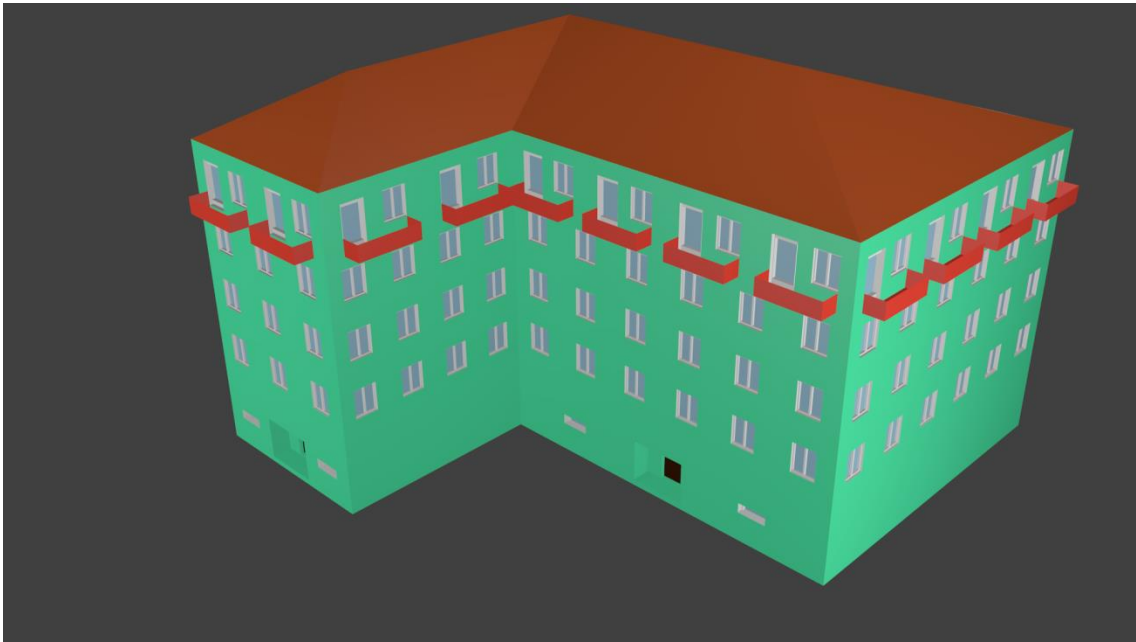
Rodinný dům jednopodlažní



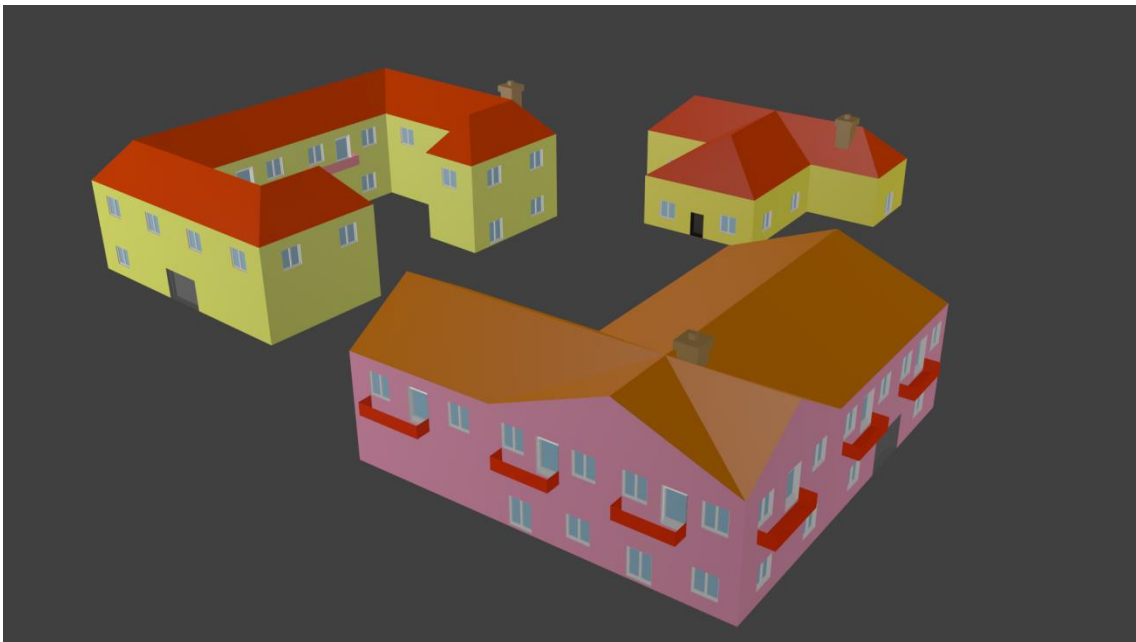
Panelový bytový dům



Budova se specifickým tvarem



Bytový dům s balkony jen v posledním patře



Ukázka více vygenerovaných budov