

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Ondřej Chvíla**

Studijní program: Softwarové technologie a management
Obor: Web a multimedia

Název tématu: **Procedurální modelování objektů v Maya**

Pokyny pro vypracování:

Vytvořte sadu skriptů nebo plugin který umožní v modeláři Maya procedurálně vytvářet typizované 3D objekty. Pro každý typ objektu vyberte vhodné modifikační parametry a naleznete jejich maximální hodnoty, které mají pro genezi objektů ještě smysl. Objekty budou z domény nádobí, založené na reálných tvarech. Ověřte schopnost generování vybraných kusů porovnáním předlohy a vygenerovaného. Pomocí skriptů půjdou vytvořit minimálně tři sady nádobí o pěti kusech. Zhodnoťte kvalitu generovaných objektů (věrnost předloze), škálu variability a rozšiřitelnost vaší práce o nové typy objektů.

Pro použití skriptů navrhnete jednoduší UI, splňující zavedené požadavky na postup práce v modeláři Maya.

Vytvořte HTML stránku, kde budete prezentovat a distribuovat vámi vytvořené skripty/plugin.

Seznam odborné literatury:

Yumer, M. E., and Asente, P., and Mech, R., and Kara, L. B.. Procedural Modeling Using Autoencoder Networks. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), 2015.

Smelik, Ruben M. and Tutenel, Tim and Bidarra, Rafael and Benes, Bedrich. A survey on procedural modeling for virtual worlds. Computer Graphics Forum 6, vol. 33, 2014.

Vedoucí: Ing. David Sedláček, Ph.D.

Platnost zadání: do konce zimního semestru 2017/2018



prof. Ing. Jiří Žára, CSc.
vedoucí katedry

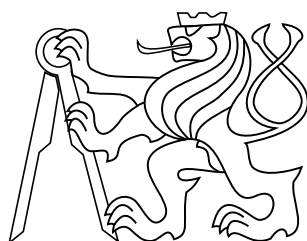
prof. Ing. Pavel Řípka, CSc.
děkan

V Praze dne 29. 2. 2016

Bakalářská práce

Procedurální modelování objektů v Maya

Ondřej Chvíla



2016

Vedoucí práce: Ing. David Sedláček, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra počítačové grafiky a interakce

Poděkování

Děkuji vedoucímu své práce Ing. Davidu Sedláčkovi, Ph.D. za náměty, připomínky a odborné rady, které mi dal a za čas, který mé práci věnoval. Dále děkuji své rodině a přátelům za podporu při studiu.

Prohlášení

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

V Praze dne 24. května 2016

Podpis autora

Abstrakt

Předmětem bakalářské práce je návrh a následná realizace sady skriptů s uživatelským rozhraním v jazyce MEL pro modelovací software Maya od společnosti Autodesk. Skripty umožní generovat pět kávových setů o třech designech založené na reálných tvarech pomocí procedurálního modelování. Dále se bakalářská práce zabývá procedurálním modelováním obecně, vlastnostmi NURBS křivek a ploch, které skripty generují a seznámí čtenáře s použitým jazykem MEL.

Klíčová slova

Procedurální modelování, Autodesk Maya, MEL skript, NURBS modelování

Abstract

The purpose of the bachelor thesis is the design and subsequent implementation of a set of scripts with user interface in the MEL scripting language for modeling software Autodesk Maya. The scripts allow you to generate with procedural modeling five types of coffee sets of three designs based on real shapes. Furthermore, the thesis will introduce reader procedural modeling in general, the properties of NURBS curves and surfaces, that the scripts generate and basics of MEL scripting language.

Keywords

Procedural modeling, Autodesk Maya, MEL script, NURBS modeling

Obsah

1 Úvod	1
1.1 Cíl bakalářské práce	2
2 Analýza	3
2.1 Procedurální modelování	3
2.1.1 Problematika procedurálního modelování	4
2.2 Autodesk Maya	4
2.2.1 Tvorba skriptu a pluginu v Maya	4
2.2.2 Tvorba 3D modelů v Maya	5
2.3 Jazyk MEL	5
2.3.1 Vlastnosti MEL	5
Výhody:	6
Nevýhody:	6
2.3.2 Typy proměnných	6
2.3.3 Procedury	6
2.3.4 Globální a lokální proměnné	7
2.3.5 Globální a lokální procedury	7
2.3.6 Grafické uživatelské rozhraní	7
Rozvržení	8
2.4 NURBS	10
2.4.1 Modelování pomocí NURBS	11
Výhody:	11
Nevýhody:	11
Způsoby tvorby NURBS povrchů	12
Historie povrchů	13
3 Návrh řešení	14
3.1 Možnosti řešení	14
3.2 Vstupní parametry	14
3.2.1 Typy šálek	14
3.2.2 Typy designu	15
3.2.3 Rozměrové parametry	16
3.2.4 Doplnkové parametry	16
3.3 Omezení vstupních parametrů	17
3.4 Návrh uživatelského rozhraní	18
4 Implementace	20
4.1 Charcoal Editor for Maya 2015	20
4.1.1 Funkce Charcoal Editoru:	20
4.2 Struktura skriptů	20
4.3 Princip modelování	20
4.3.1 Modelování primitiv	21
4.3.2 Modelování pomocí křivek	21
Revolve	21
Lofting	22
Extrude	23
4.4 Možnosti rozšíření	23

4.5	Grafické uživatelské rozhraní	24
4.6	Distribuce skriptů	25
5	Testování	26
5.1	Porovnání výsledků	26
5.2	Škála variability	28
6	Závěr	31
	Přílohy	
A	Spuštění skriptů	32
B	Obsah přiloženého DVD	33
	Literatura	34

Zkratky

Zkratky použité v tomto dokumentu:

MEL	Maya Embedded Language
NURBS	Non-Uniform Rational B-Splines
GUI	Graphical User Interface
CV	Control vertices
PM	Procedural modeling
API	Application Programming Interface
IDE	Integrated Development Environment

1 Úvod

Využití počítačové 3D grafiky je v dnešní době používáno v mnoha odvětvích jako například pro tvorbu filmů, počítačových her, ale i ve vědě a průmyslu. Stále častěji je potřeba vytvářet 3D modely. Ty můžeme získat při ručním modelování člověkem nebo 3D rekonstrukcí objektů z reálného světa. Většina z modelů, je ale vytvořena pomocí ručního modelování.

Vhodnou alternativou pro ruční modelování by mohlo být procedurální modelování, díky kterému můžeme generovat obsah na základě vstupních parametrů. Procedurální modelování slibuje jak časovou úsporu návrhářům a modelářům, tak širokou variabilitu výstupů. Programy na procedurální modelování budou hrát v budoucnu významnou roli při modelování a animaci. Na trhu už nalezneme mnoho skriptů a pluginů zabývajících se touto problematikou. Ať už od jednoduchých volně stáhnutelných skriptů, až po drahé profesionální skripty využívající se v mnoha herních a filmových studiích.

Motivací k vytváření těchto skriptů je nejenom velká časová úspora, ale také umožnit uživatelům generovat obsah na základě jejich představ. Takovým druhem modelů, u kterého vyžadujeme originální tvary a zároveň jednotnou funkčnost je kavárenské nádobí. Právě u kávových šálek nalezneme velikou tvarovou různorodost, přesto jejich tvary vycházejí ze vžitých konvencí. Díky těmto vlastnostem může skript na generování kavárenského nádobí oslovit širokou skupinu potenciálních uživatelů.

1.1 Cíl bakalářské práce

Cílem bakalářské práce je generovat kavářenské nádobí pomocí sady skriptů, viz Obrázek 1. Modely jsou generovány na základě přednastavených vstupních parametrů, které může uživatel měnit a tím změnit tvar, velikost a typ designu všech typů kávových šálek. Uživatel se může rozhodnout, zda-li chce generovat celý kávový set, šálek, podšálek, příp. víčko. Skripty by měli uživateli nabídnout možnost generovat i šálky bez ucha a různé způsoby generování podle užití.

Grafické uživatelské rozhraní by mělo být jednotné, přehledné a splňující požadavky na postup práce v modeláři Maya. U způsobu nastavování vstupních parametrů má být zřejmé, které hodnoty pro genezi jsou maximální. Rozhraní by mělo být samovysvětlující, aby i neznalý uživatel kávových šálek měl před generováním představu, jak výsledný objekt bude vypadat.

Další cíl bakalářské práce je seznámit čtenáře s procedurálním modelováním, použitým jazykem MEL a modelováním pomocí NURBS křivek a ploch v modeláři Maya.



Obrázek 1 Požadovaný výsledek - použité materiály *mia_material_x_passes* pro *mentalRay* render

2 Analýza

Kapitola seznámí čtenáře s procedurálním modelováním a její problematikou, s modelovacím softwarem Maya, skriptovacím jazykem MEL, jeho vlastnostmi a základními principy jazyka. Dále kapitola analyzuje NURBS křivky a modelování pomocí nich.

2.1 Procedurální modelování

PM (procedurální modelování) se zabývá automatickým či poloautomatickým generováním obsahu pomocí procedur na základě vstupních parametrů. Pomocí PM lze modelovat širokou škálu objektů různých kategorií, například organické tvary jako stromy nebo syntetické tvary jako jsou budovy, nábytek nebo šperky [1].

Malé množství vstupních parametrů nebo několik pravidel procedurálního modelování zajistí širokou variabilitu generovaných modelů. PM výrazně snižuje čas potřebný k modelování digitálního obsahu. Další výhodou je, že metody používané při PM jsou často stochastické, což umožňuje vytvořit unikátní tvar na základě stejných parametrů. Použití PM je velice atraktivní při generování virtuálního prostředí. Virtuální světy jsou důležité pro mnoho aplikací jako jsou počítačové hry a nejrůznější simulace. Ruční způsoby modelování 3D obsahu je velice pracné a nákladné, protože vyžadují vysokou úroveň detailu. Z tohoto důvodu byla v posledních letech věnována zvýšená pozornost právě na PM i z řad vědeckých obcí.



Obrázek 2 Kaňon s volnými kameny vytvořený PM pomocí frameworku Arches [2]

Zvyšující se požadavky a očekávání uživatelů, rychlý pokrok u zobrazovacích zařízeních a u počítačového hardwaru, snaha o rozsáhlejší a vizuálně přesvědčivější virtuální světy. Díky těmto faktorům se v následujících letech bude klást veliký tlak právě na vývoj PM. Potenciál totiž není zdaleka využit [2].

2.1.1 Problematika procedurálního modelování

Většina současných metod na PM, ale bohužel nenabízí vhodnou alternativu k ručnímu modelování a to i přes slíbený zisk produktivity a zdánlivě nekonečné změny obsahu. Hlavním důvodem je nedostatečná ovladatelnost. Vyžaduje se manipulace s komplikovanými pravidly a parametry, u kterých může uživatel výstup těžko předpovídat. Výsledkem je, že systémy generují modely s malou variabilitou, kde uživatel upravuje přednastavené vzory nebo náhodně mění parametry než dosáhne požadovaného výsledku [2].

2.2 Autodesk Maya

Autodesk, Inc., je lídrem v oblasti 3D návrhového, technického a animačního softwaru. Od uvedení programu AutoCAD v roce 1982 rozvíjí Autodesk nejširší portfolio 3D softwaru pro globální trhy [3].

Autodesk Maya je profesionální program pro tvorbu 3D grafiky. Často bývá používán ve filmu a televizním průmyslu pro vytváření 3D efektů, ale slouží i k tvorbě počítačových her. Maya představuje výkonně řešení, které integruje 3D modelování, animaci, vizuální efekty a renderování. Předností programu je konfigurovatelnost a značná rozšiřitelnost. Je multiplatformní - Windows, Mac OS X a Linux. Maya také nabízí skriptovací jazyk Python a svůj vlastní jazyk MEL [4].

2.2.1 Tvorba skriptu a pluginu v Maya

Modelář Maya nabízí tři různé způsoby, jak rozšířit jeho funkcionalitu pomocí skriptu či pluginu: MEL, Python a C++ API.

C++ API nabízí jediný způsob, jak psát nativní kód pro Maya. Nejedná se o skriptovací jazyk, tím pádem tvorba pluginu vyžaduje externí IDE a musí dojít ke kompilaci na rozdíl od jazyka Python nebo MEL. Může zde nastat problém u přenášení z různých platforem.

Jazyk MEL je speciálně určený pro práci v modeláři Maya, celé Maya GUI je vytvořeno pomocí jazyka MEL. Je multiplatformní a nabízí rychlé řešení pro jednoduché či složité úlohy [5]. Více o Jazyku MEL v kapitole 2.3.

Python je alternativa mezi C++ API a MEL. Jedná se o pokročilý skriptovací jazyk s velikou základnou uživatelů. Python je hybridní jazyk, to znamená, že nabízí jak programování objektově orientované, tak i procedurální.

Pro své účely jsem se rozhodl pro skriptovací jazyk MEL kvůli snadnému prototypování, interaktivnímu procesu implementace a především kvůli intuitivnosti, kdy všechny příkazy v Maya mají odezvu právě v jazyce MEL. Vlastnosti, výhody, nevýhody a základy jazyka MEL jsou popsány v kapitole 2.3.

2.2.2 Tvorba 3D modelů v Maya

V Autodesk Maya jsou tři možnosti, jak modelovat objekty. Jednou z metod je modelování pomocí NURBS, další metoda je polygonální modelování a poslední pomocí dělení povrchů (subdivision surfaces).

U polygonálního modelování jsou jednotlivé polygony vytvořeny z plošek (faces). Jednoduchá polygonální plocha je povrch tvořený třemi nebo více body, nazývanými vrcholy (vertices), které jsou vzájemně spojeny. Čára spojující jeden vrchol s druhým se nazývá hrana (edge). Polygonální modelování je poměrně oblíbené, protože výsledný model se skládá z jednoho mnohostranného geometrického kusu. Proto můžeme tyto modely bez obav deformovat, aniž hrozí vzájemné vzdalování švů, jak se to může stát v případě NURBS.

Modelování pomocí NURBS je založeno na matematice, která je složitější než v případě polygonů. Protože modelování pomocí NURBS vyžaduje více zpracování, je tato metoda obvykle používána v aplikacích, ve kterých vykreslování probíhá předem (nikoliv v reálném čase). Více o NURBS v kapitole 2.4.

Dělení povrchů spojuje to nejlepší z modelovací metody polygonální a NURBS. Doplňuje snadnost polygonální tvorby s hladkostí a organickým tvarem geometrie NURBS. Dělení povrchů obvykle začíná jako polygonální povrch. Poté můžeme k vyhlazení použít NURBS, kde drsné polygonální povrchy vyhladíme jejich dělením podle toho, jak vyžaduje vzhled našeho modelu. Nevýhodou dělených povrchů je to, že vyžadují více výpočetního času než modely NURBS [6].

2.3 Jazyk MEL

MEL je akronym pro **Maya Embedded Language**. Jedná se o programovací jazyk speciálně určený pro práci v modeláři Maya. Jazyk MEL je uživatelsky přístupný programovací jazyk, vycházející z jazyků Perl a Tcl. Aniž bychom to věděli už od otevření programu Maya byl MEL použit. Výběr objektu, či zobrazení dialogového okna je přímým důsledkem spuštění příkazu MEL. Ve skutečnosti celé grafické rozhraní Maya je tvořeno právě tímto jazykem.

2.3.1 Vlastnosti MEL

Vzhledem k tomu, že MEL je skriptovací jazyk můžeme jednoduše napsat příkaz a okamžitě jej spustit, nedochází u něj ke kroku kompilování a linkování jako například u jazyků C nebo C++. To znamená, že můžeme s příkazy experimentovat a vidíme okamžité výsledky. V případě, že skript nefunguje, je ho možné rychle změnit a znovu spustit. Celý interaktivní proces implementace a testování je mnohem rychlejší a jednodušší. Skripty mohou být psané jak pro základní úkoly, tak pro složité, jako například k vytvoření komplikovaného GUI. Schopnost řešit jednoduché i složité problémy, aniž bychom museli znát jiný programovací jazyk je jednou z největších předností jazyka MEL. S dobrou znalostí programování v MEL bude práce s modelářem Maya rychlejší a produktivnější při každodenních úkolech [5].

Výhody:

- Maya GUI je vytvořeno jazykem MEL.
- Pomáhá pochopit, jak Maya funguje.
- Je multiplatformní.
- Příkazy zobrazené jako odezva programu jsou v MEL.
- Rozsáhlá dokumentace s jednotlivými příklady.

Nevýhody:

- Nepatří mezi pokročilé skriptovací jazyky jako např. Python.
- MEL se moc nezměnil od doby svého vzniku.
- Nenabízí rozšiřující knihovny a podporu třetích stran.
- Malá základna uživatelů.

2.3.2 Typy proměnných

MEL poskytuje omezenou sadu typů proměnných: *int*, *float*, *string*, *vector* a *matrix*, nicméně tyto proměnné jsou dostačující pro skriptovací potřeby většiny uživatelů [5].

V mnoha jazycích, včetně MEL, můžeme vytvořit proměnné za běhu skriptu. Některé jiné jazyky vyžadují zadefinovat proměnné na začátku skriptu. Názvy proměnných jsou tvořeny znakem dolaru (\$) následovaný kombinací písmen, číslic a podtržíték. Použití prefixu \$ pro označení proměnných pomáhá zajistit, že proměnná není zaměnitelná s názvy příkazů či atributů [7].

Například pro definování proměnné s názvem \$rotation_value, typem float a číslem 4.0 nadefinujeme tímto řádkem:

```
float $rotation_value = 4.0;
```

Častokrát potřebujeme uložit pole proměnných o stejných typech. Pole (array) MEL umožňuje definovat i při neznámém počtu prvků a v případě potřeby pole automaticky zvětší. Například nadefinování pole s typem string:

```
string $names[] = {"Bill", "Bob", "Ollie"};
```

2.3.3 Procedury

Cílem strukturovaného programování je rozložit skript na menší funkční kousky, které mohou být znovu použity. MEL umožňuje výsledku dosáhnout pomocí **procedur**. Postup je podobný jako při zavolání příkazu. Procedura se zavolá a vykoná nějakou operaci. Procedura může obsahovat libovolný počet vstupních proměnných [5]. Jednoduchý příklad procedury na vypsání vašeho jména:

```
proc printName(){  
    print "My name is Ollie";  
}
```

2.3.4 Globální a lokální proměnné

Proměnné \$a a \$b jsou považovány za **lokální** pro jejich konkrétní procedury či bloky. Být lokální proměnná znamená, že jsou přístupné pouze v rámci svého bloku a případných vnitřních bloků, které mohou být definovány. Občas budeme chtít mít proměnnou, která může být sdílena mezi ostatními bloky. Tyto proměnné se nazývají **globální**. Po definování globální proměnné je tato proměnná přístupna po celém modeláři Maya. Globální proměnná se definuje pomocí klíčového slova - *global* [5].

```
proc printName() {
    global string $currentUserName "
}
```

2.3.5 Globální a lokální procedury

Stejně jako u proměnných, je možné definovat procedury jako **lokální** či **globální**. Není-li uvedeno jinak, procedury jsou nastaveny jako lokální. Lokální procedura je přístupna pouze v skriptu, ke kterému je definována.

```
global proc printName(string $name){
    print $name;
}
```

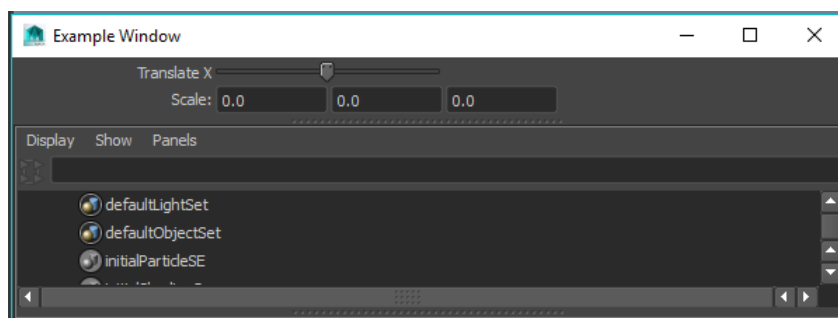
Tuto proceduru nyní lze zavolat odkudkoliv v modeláři Maya. Jako globální proměnné, tak i procedury sdílejí stejný obor názvů. Chceme-li snížit pravděpodobnost kolize s jiným skriptem vytváříme jedinečné názvy pro globální procedury. Mějme na paměti, že jakmile je globální procedura načtena, zůstává rezidentní v paměti Maya. Obecně platí, že se snažíme snížit počet globálních procedur v našem skriptu na minimum [5].

2.3.6 Grafické uživatelské rozhraní

MEL může být použit k vytvoření široké škály grafických uživatelských rozhraní. Jak již bylo v předchozí kapitole zmíněno, celé uživatelské rozhraní Maya je postaveno a řízeno právě pomocí MEL příkazů.

Je pravděpodobné, že funkcím vytvořeným v našem skriptu budeme muset poskytnout uživatelské rozhraní. V typickém vývojovém prostředí může být tvorba GUI zdoluhavý úkol. Zatímco Maya neposkytuje vizuální prostředky pro vytváření prvků pro uživatelské rozhraní, poskytuje bohatou sadu příkazů pro samotnou tvorbu a správu. Následující skript generuje okno, které je znázorněno na obrázku 3 [5].

```
window -title "Example Window";
    paneLayout -configuration "horizontal2";
    columnLayout;
    floatSliderGrp -label "Translate X" -min -100 -max 100;
    floatFieldGrp -label "Scale:" -numberOfFields 3;
    setParent .;
    frameLayout -labelVisible false;
    outlinerPanel ;
showWindow;
```



Obrázek 3 Zkušební okno

Všimněte si, jak takový složitý prvek GUI jako je *outliner panel* může být snadno použit do vlastních oken.

Rozvržení

MEL nabízí širokou škálu rozvržení - *layouts*. Každé rozvržení slouží ke společnému cíli: umístění jednotlivých prvků v rozhraní. Komplexní skripty mohou být vytvořeny vnořením různých typů rozvržení, ale každé okno musím mít alespoň jeden typ. V následujících kapitolách si představíme několik základních rozvržení [5].

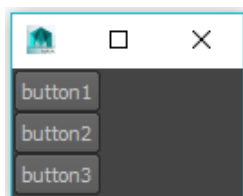
ColumnLayout

Nejjednodušší ze všech rozvržení je *columnLayout*. To jednoduše umístí všechny ovládací prvky ve vertikálním sloupci. Je zde možnost definovat šířku sloupce a výšku řádku.

```

window ;
    columnLayout ;
        button -label "button1 ";
        button -label "button2 ";
        button -label "button3 ";
showWindow ;

```

Obrázek 4 Ukázka pro rozvržení *columnLayout*

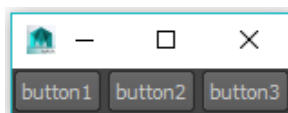
RowLayout

RowLayout umístí všechny ovládací prvky ve vodorovné řadě. Musíme určit počet sloupců, které mají být použity v řádku. Je možnost nastavení zarovnání a odsazení. V následujícím příkladu používáme tři sloupce.

```

window ;
    rowLayout -numberOfColumns 3 -columnWidth3 50 50 50;
        button -label "button1 ";
        button -label "button2 ";
        button -label "button3 ";
showWindow ;

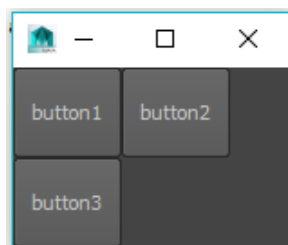
```

Obrázek 5 Ukázka pro rozvržení *rowLayout*

GridLayout

GridLayout umístí všechny své prvky do série řádků a sloupců. Je zde možnost nastavit počet řádků a sloupců v mřížce a výšku a šířku jednotlivých buněk.

```
window ;
  gridLayout -numberOfRowsColumns 2 2 -cellWidthHeight 60 50;
  button -label "button1 ";
  button -label "button2 ";
  button -label "button3 ";
showWindow ;
```

Obrázek 6 Ukázka pro rozvržení *gridLayout*

FormLayout

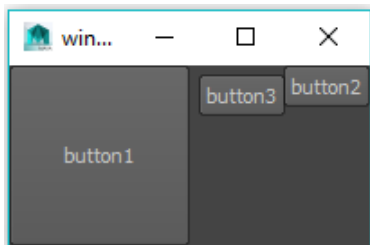
Jedná se o nejvíce flexibilní schéma rozvržení. Nabízí velmi mnoho možností zarovnání a ukotvení. Poskytuje absolutní a relativní polohování prvků. Po přidání prvku do rozvržení *formLayout*, musíme ještě specifikovat jeho polohu - není nastavena výchozí hodnota.

```
window ;
  string $form = 'formLayout -numberOfDivisions 100';
  string $but1 = 'button -label "button1" ';
  string $but2 = 'button -label "button2" ';
  string $but3 = 'button -label "button3" ';
  formLayout -edit
  // Button 1
  -attachForm $but1 "top" 0
  -attachForm $but1 "left" 0
  -attachForm $but1 "bottom" 0
  -attachPosition $but1 "right" 0 50
  // Button 2
  -attachForm $but2 "right" 0
  // Button 3
  -attachPosition $but3 "top" 0 5
  -attachControl $but3 "left" 5 $but1
  $form ;
showWindow ;
```

Příznak *-numberOfDivisions* definuje počet buněk ve virtuální mřížce, na kterou se umísťují jednotlivé elementy. Tento příznak může být nastaven na jakoukoliv hodnotu,

ale nejčastěji se používá hodnota 100, aby pozice mohla být udávána v procentech. To znamená, že nastavení horizontální polohy na 50 je jako uvedení 50% šířky okna.

Výsledné uživatelské rozhraní je ukázáno na obrázku 7. Do rozvržení *formLayout* je možné přidávat další rozvržení, včetně dalšího *formLayout*. Touto kombinací docílíme větší kontroly nad velikostí a umístěním každého jednotlivého prvku.

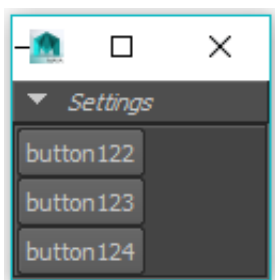


Obrázek 7 Ukázka pro rozvržení *formLayout*

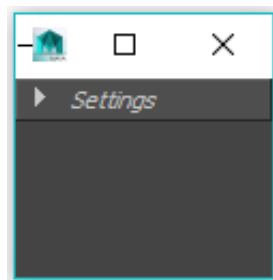
FrameLayout

Potřebujeme-li mít oblast rozhraní, kterou lze rozbalit nebo sbalit, použijeme *frameLayout*. U tohoto rozvržení můžeme měnit štítek, okraje a font. Je důležité si uvědomit, že *frameLayout* může obsahovat pouze jeden prvek, proto musí být přidáno další rozvržení pro ostatní prvky.

```
window ;
    frameLayout -label "Settings" -borderStyle "etchedIn"
    -font "obliqueLabelFont" -collapsable true ;
    columnLayout ;
        button ;
        button ;
        button ;
showWindow ;
```



a) Rozbalená nabídka



b) Sbalená nabídka

Obrázek 8 Ukázka pro rozvržení *frameLayout*

2.4 NURBS

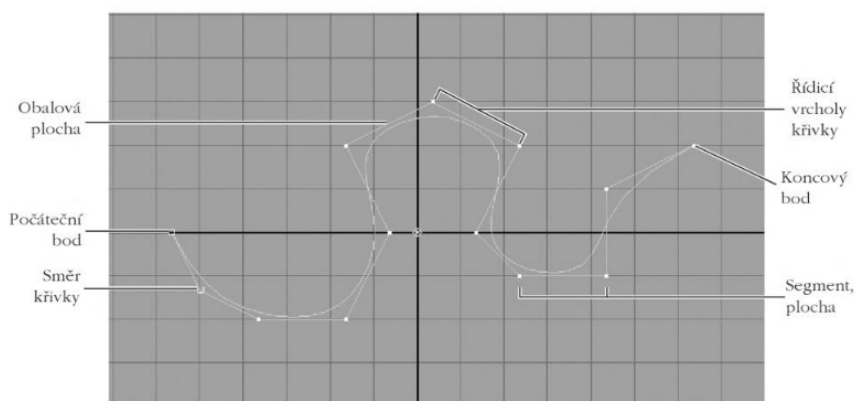
NURBS je zkratka pro **Non-Uniform Rational B-Splines** (Nepřavidelné racionální B-křivky). Geometrie NURBS je založena na Béziových křivkách, matematickém konceptu původně zavedeném francouzským inženýrem Pierrem Bézierem. Béziové křivky se vykreslují mezi **řídícími vrcholy** (control vertices, CV) založenými na rovnicích používajících kubické polynomy [6].

V podstatě lze říci, že Bézierovy křivky jsou tvořeny počátečním a koncovým řídicím vrcholem CV a většinou dvěma nebo více mezilehlými CV, které poskytují hladké zakřivení. Po vytvoření všech CV se křivka snaží procházet z předchozího na následující CV nejhladším možným způsobem [8].

2.4.1 Modelování pomocí NURBS

Modelování NURBS vyniká v tvorbě zakřivených tvarů a čar, takže se častěji využívá pro organické tvary, jako jsou zvířata a lidi i pro vysoce detailní tvary, jako jsou např. vozidla.

Jak je vidět na obrázku 9, CV řídí křivost. Obalové plochy (hulls) spojují tyto CV a jsou užitečné pro výběr více řad CV najednou. Počáteční CV se v Maye objevuje jako uzavřený čtvereček a druhý CV, který definuje směr zakřivení, je otevřeným čtverečkem, takže můžete jednoduše spatřit směr, ve kterém byla křivka vytvořena [6].



Obrázek 9 Bézierova křivka a její komponenty [6]

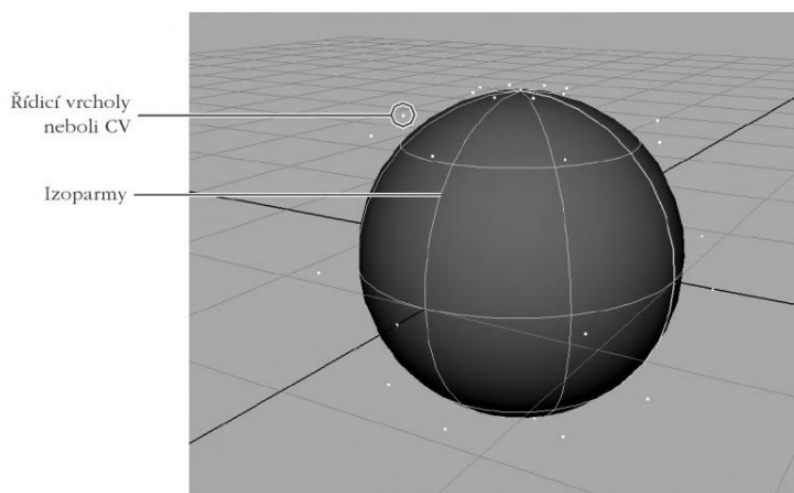
Tato křivka přirozeně končí koncový CV. Počáteční a koncový CV jsou pouze řídicími vrcholy, které jediné jsou na křivce vždy. Zatímco vrcholy CV řídí křivost Bézierovy křivky, povrchy NURBS jsou definovány křivkami zvanými **izoparmy** (isoparms), které jsou vytvořeny prostřednictvím vrcholů CV. Povrch je vymezen těmito izoparmami a vytváří **segmenty** (spans), které sledují povrchové zakřivení definované těmito izoparmami, jak je patrné na obrázku 10. Čím více segmentů, tím větší stupeň detailu a kontroly nad tímto povrchem [6].

Výhody:

- NURBS křivky a povrchy jsou vždy dokonale vyhlazené (nejsou aproximované).
- K vytvoření stejného tvaru je potřeba daleko méně informací (řídicích vrcholů) oproti polygonálnímu modelování.
- NURBS povrchy lze kdykoliv konvertovat na polygonální geometrii [9].

Nevýhody:

- Modelování nepravidelných a složitých objektů je složitější.
- Vykreslování NURBS objektů je výrazněji pomalejší (nepoužívá se v real-time počítačové grafice) [9].



Obrázek 10 Povrchy NURBS jsou vytvářeny mezi izoparmami. Můžete je modelovat pomocí změn poloh jejich řídicích vrcholů CV [6]

Způsoby tvorby NURBS povrchů

Povrch NURBS můžeme vytvořit různými způsoby. Nejjednodušší je vytvořit základní tvar NURBS. Tento základní tvar můžeme modelovat pohybováním jeho řídicích vrcholů CV, ale můžeme to učinit také tak, že vytvoříme různé ústřížky a záplaty (patches), pokud je nezbytné. Základní tvar nemusí zůstat nezměněn. Použitím nástrojů na úpravu povrchu můžeme oddělovat, ořezávat a připojovat kusy od a k základnímu tvaru, čímž docílíme požadovaného tvaru.

Povrchy můžeme také vytvořit několika způsoby, aniž bychom používali základní tvary. Všechny tyto metody zahrnují nejprve vytvoření nebo použití již existující křivky NURBS (nebo křivek na jiném povrchu). Jejich prostřednictvím definujeme část nebo části tohoto povrchu a pak použijeme jednu z následujících metod, kterou daný povrch vytvoříme [6].

Lofting

Nejběžnější povrchovou metodou je lofting, které používá minimálně dvě křivky a vytváří povrchovou plochu (pole) ležící mezi každou vybranou z nich. Definujeme-li více křivek pro lofting, Maya vytvoří složitější tvary. Čím více CV pro každou křivku použijeme, tím více izoparem máme a tím větší počet detailů je na tomto povrchu.

Lofting se nejlépe hodí tam, kde jsou křivky nakresleny jako průřezové vrstvy modelovaného objektu. Používá se k tvorbě rozličných povrchů, které mohou být jednoduché jako deska stolu nebo složité jako lidská tvář [6].

Revolve

Revolve vyžaduje pouze jedinou křivku, která se otáčí kolem osy a vytváří tak povrch. Nejprve definujeme profil požadovaného objektu tak, že nakreslíme **profilovou křivku** a pak necháme tuto křivku rotovat (od 0° do 360°).

Revolve je užitečný při vytváření objektů, jako jsou láhve, nohy nábytku, baseballové hole - zkrátka všeho, co je symetrické dle osy [6].

Extrude

Extrude používá dvě křivky - profilovou a směrovou. Profilová křivka slouží ke stanovení tvaru profilu požadovaného povrchu. Tento profil se pak natahuje z jednoho konce trasy ke druhému, což vytváří povrchovou plochu podél dráhy směrové křivky. Čím větší počet CV se nachází na každé křivce, tím větší počet podrobností na povrchu získáme.

Extrude se používá k tvorbě objektů jako jsou proudové (větrné) tunely, vinuté zahradní hadice, pružiny nebo závěsy [6].

Historie povrchů

Maya obsahuje možnost zachovat konstrukční historii při tvorbě objektů. Aktivace historie během vytváření povrchů NURBS umožňuje, aby se povrch aktualizoval, jakmile se změní části, z nichž se skládá. Například loft, který jsme vytvořili, se aktualizuje okamžitě, jakmile dva původní použité povrchy nebo křivky změní svůj tvar nebo se přemístí. Tato technika pracuje se všemi povrchovými technikami, nikoli pouze s loftingem. Historie je užitečná k provádění úprav a jemnému doladování povrchu a může být praktická, jestliže potřebujeme deformovat několik povrchů a chceme zachovat jejich spojení [6].

3 Návrh řešení

K dosažení cíle bakalářské práce lze dospět několika způsoby, které v následující kapitole popíši. Vyberu si jedno z řešení a v dalších kapitolách se zaměřím na jejich vlastnosti a výhody oproti ostatním.

3.1 Možnosti řešení

Pro bakalářskou práci jsem si vybral modelování pomocí NURBS, které jsou vždy dokonale vyhlazené. U jednoduchých tvarů potřebují daleko méně informací oproti polygonálnímu modelování a kdykoliv lze NURBS plochy konvertovat na polygonální geometrii. Vlastnosti NURBS jsou blíže specifikovány v kapitole 2.4.

Maya software podporuje tři jazyky pro psaní skriptů či pluginů. C++ API umožňuje psát pouze pluginy. Pomocí jazyka Python lze psát jak skripty, tak pluginy a u jazyka MEL pouze skripty. Rozhodl jsem se pro skripty, kvůli snadnému a rychlému prototypování a interaktivnímu procesu implementace. Pro mé účely jsem si zvolil skriptovací jazyk MEL, protože dobře popisuje vnitřní chování Maya, je jednoduchý a odezva jednotlivých příkazů Maya se projeví právě v jazyku MEL. Výhody, nevýhody, vlastnosti a jednotlivé příkazy jsou blíže specifikovány v kapitole 2.3.

Na základě problematiky procedurálního modelování viz kapitola 2.1.1 byla zvolena možnost, kdy skripty generují modely s nižší variabilitou, ale s očekávaným výstupem. Generované kávěnský nádobí musí vycházet z reálných tvarů, které se na nádobí používají. Není zde taková možnost variability, jako například u generování abstraktních tvarů nebo modelů složených z mnoha menších.

3.2 Vstupní parametry

Ve skriptu je možnost výběru z pěti různých typů šálek podle kávového nápoje - *espresso*, *cappuccino*, *caffè latte*, *irská káva* a *klasický hrnek*. Další možností je výběr ze třech různých designů - *minimalistický*, *kubistický* a *kelímky na kávu s sebou*. U každého šálku má uživatel možnost výběru, jestli chce generovat celý set - šálek i s podšálkem popř. víčkem, nebo pouze jednotlivé nádobí. Při generování je možnost změnit parametry *horního poloměru*, *dolního poloměru* a *výšky*. Mezi vstupní parametry dále patří způsob generování a možnost zachování kontrolních křivek.

3.2.1 Typy šáleků

Skripty generují pět tvarů šáleků. Každý tvar je založen na reálných a vžitých tvarech používané pro konkrétní kávový nápoj.

- Espresso - šálek na espresso je kónického tvaru o celkovém objemu 60 ml.
- Cappuccino - šálek je při spodním poloměru oblejšího tvaru než espresso a objem šálku je 2,5 krát větší.

- Caffè latte - tvar šálku je kónický s nízkou podstavou a s celkovým objemem 280 ml.
- Irská káva - šálek je na úzké nožce s podstavou a s celkovým objemem 230 ml.
- Klasický hrnek - poloměr hrnku je po celé své výšce stejný. Objem je 250 ml.

3.2.2 Typy designu

Mezi další parametry skriptů jsou jednotlivé typy designu. Skripty umožňují generovat tři typy designu.

Minimalistický

Design vycházející z klasického kavárenského nádobí. Půdorys šálků je kulatý a nevyskytují se u něj ostré hrany. Ideální materiál pro tento design je bílá keramika s glazurou a čiré sklo.



Obrázek 11 Reference pro *minimalistický* design¹

Kubistický

Design vycházející z výtvarného umění a architektury z počátku 20. století. Půdorys šálku je tvarem osmihran a hrany jsou zaoblené. Materiál pro tento design je ideální černá či bílá keramika s glazurou nebo čiré sklo.



Obrázek 12 Reference pro *kubistický* design²

¹ Obrázky byly použity z těchto zdrojů:
http://shop.illy.com/online/store/product_cups-glassware-us_illy-logo-cappuccino-cups_us
<http://www.ikea.com/gb/en/catalog/products/00284386/>
<http://www.foodlushblog.com/2011/03/the-best-irish-coffee-recipe-for-st-pattys-day.html>
<https://www.qualitylogoproducts.com/mug-personalized.htm>

² Obrázky byly použity z těchto zdrojů:
<http://kirbytent.com/product/octagon-china/>
<https://www.etsy.com/listing/119613730/vintage-black-arcoroc-octime-octagon>

S sebou

Design kelímků pro kávu s sebou. Půdorys kelímků je kulatý a tvar neobsahuje ostré hrany. Materiály pro tento design jsou ideální papír či plast pro kelímek, papírové madlo a plastové víčko.

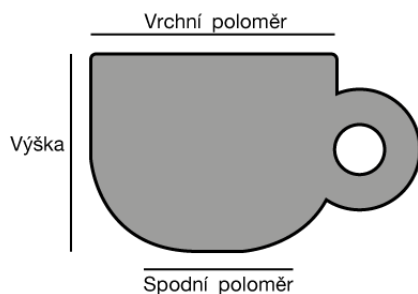


Obrázek 13 Reference pro design na kávu s sebou ³

3.2.3 Rozměrové parametry

Do skriptů vstupují parametry, které ovlivňují výšku, vrchní poloměr a spodní poloměr šálku. Pro každý typ je nastavená hodnota jednotlivých parametrů na referenční hodnoty. Uživatel má možnost jednotlivé parametry měnit z přednastavené škály k požadovanému výsledku.

- **Vrchní poloměr** - parametr, který udává vzdálenost vrchního okraje šálku od svislé osy.
- **Spodní poloměr** - parametr, který udává vzdálenost spodního okraje šálku od svislé osy.
- **Výška** - parametr udávající vzdálenost vrchní hrany šálku od základní mřížky.



Obrázek 14 Rozměrové parametry

3.2.4 Doplnkové parametry

Parametry, u kterých si uživatel může vybrat jestli se provedou. Jedná se o vygenerování ucha k šálku, zda se zachovávají modifikační křivky a zda se mohou objekty překrývat.

³ Obrázky byly použity z těchto zdrojů:
<http://a1coffee.co.uk/lavazza-4oz-cups-100.html>
<http://www.shutterstock.com/pic-148313015.html>
<http://www.coffeemanuk.co.uk/products/smoothie-cups-12oz-x1000>
<http://www.americanpartycups.com/red-plastic-cups/>

- **Generování ucha** - nabízí možnost vygenerovat ucho k vybranému šálku o daných parametrech. Ucho je ideálně velké a umístěné vzhledem k šálku.
- **Zachování kontrolních křivek** - možnost zachovat křivky, podle kterých se modeluje vybraný šálek. Je zachována historie a tím se při změně křivky změní i výsledný objekt.
- **Překrývání objektů** - možnost, kdy si uživatel vybírá, zda se mohou generované objekty překrývat.

3.3 Omezení vstupních parametrů

Pro každý šálek z každého typu designu jsou nastaveny minimální a maximální hodnoty, pro které má geneze objektů ještě smysl. Vstupní parametry - *vrchní* a *spodní poloměr* na sobě u některých typů šálků mohou být závislé. Zvláště kvůli estetickým vlastnostem a stabilitě. Parametr *výšky* je na ostatních nezávislý.

V následujících tabulkách jsou vypsány mezní hodnoty parametrů a omezení vztahů mezi nimi pro jednotlivé designy. Čísla v tabulkách jsou minimální a maximální násobky referenčních hodnot pro jednotlivé typy.

	minimální výška	maximální výška	omezení vztahu vrchního a spodního poloměru
espresso	0,7	1,3	$\text{abs}(V-S) < 0,4$
cappuccino	0,7	1,3	$\text{abs}(V-S) < 0,4$
caffè latte	0,7	1,3	$(V-S) < 0,2$
irská káva	0,9	1,1	$\text{abs}(V-S) < 0,2$
klasický hrnek	0,9	1,1	nemá spodní poloměr

Pozn.: minimální spodní a vrchní poloměr pro všechny typy šálků je 0,7

maximální spodní a vrchní poloměr pro všechny typy šálků je 1,3

V = vrchní poloměr, S = spodní poloměr

Tabulka 1 Omezení pro *minimalistický* design

	min. výška	max. výška	min. vrchní poloměr	max. vrchní poloměr	omezení vztahu vrchního a spodního poloměru
espresso	0,7	1,3	0,7	1,3	$(V-S) < 0,3$; $(S-V) < 0,2$
cappuccino	0,7	1,3	0,7	1,3	$(V-S) < 0,3$; $(S-V) < 0,2$
caffè latte	0,9	1,3	0,8	1,3	$\text{abs}(V-S) < 0,2$
irská káva	0,9	1,3	0,7	1,3	$\text{abs}(V-S) < 0,2$
klasický hrnek	0,9	1,1	0,7	1,3	nemá spodní poloměr

Pozn.: minimální spodní poloměr pro všechny typy šálků je 0,7

maximální spodní poloměr pro všechny typy šálků je 1,3

V = vrchní poloměr, S = spodní poloměr

Tabulka 2 Omezení pro *kubistický* design

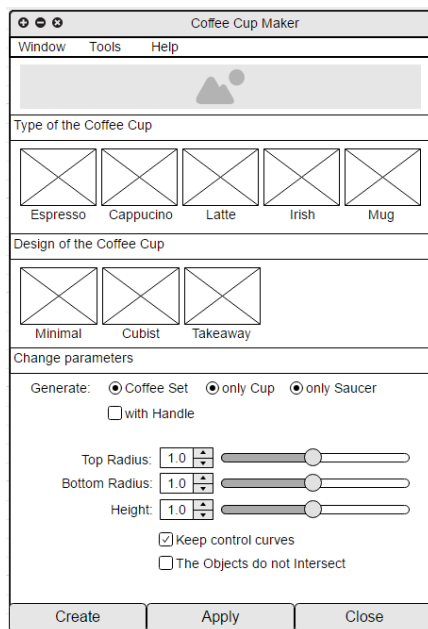
	minimální výška	maximální výška	omezení vztahu vrchního a spodního poloměru
espresso	0,7	1,3	$V > S$; $(V-S) < 0,3$
cappuccino	0,7	1,3	$V > S$; $(V-S) < 0,3$
caffè latte	0,7	1,3	$(V-S) < 0,3$; $(S-V) < 0,2$
irská káva	0,7	1,3	$(V-S) < 0,3$; $(S-V) < 0,2$
klasický hrnek	0,9	1,1	nemá spodní poloměr

Pozn.: minimální spodní a vrchní poloměr pro všechny typy šálek je 0,7
 maximální spodní a vrchní poloměr pro všechny typy šálek je 1,3
 V = vrchní poloměr, S = spodní poloměr

Tabulka 3 Omezení pro šálky s sebou

3.4 Návrh uživatelského rozhraní

Vzhled okna a jednotlivých prvků je konzistentní s uživatelským rozhraním Maya. Stejně akce mají stejný popis a chovají se stejně. Využívá se výchozího vzhledu se standardními ovládacími prvky. V každém stavu by měl uživatel vědět, v jakém stavu se nachází, jestli se čeká na zadání vstupu, či se provádí určitá operace. Skripty musí být efektivní a jednoduché na použití, zároveň obsahovat dostatečný počet voleb. Pokročilejším uživatelům by měly nabídnout efektivnější řešení. V GUI by se mělo zobrazovat co nejméně informací a voleb, tak aby práce byla co nejrychlejší a přehledná. Rozhraní by mělo být samovysvětlující, aby i neseznámený uživatel měl představu, co má očekávat za výsledek.



Obrázek 15 Návrh grafického uživatelského rozhraní

Menu pod hlavní lištou

Menu pod hlavní lištou okna se skládá ze tří záložek.

- **Window** - Nabízí funkci pro ukotvení okna skriptů do rozhraní modeláře Maya a funkci na jeho zavření.
- **Tools** - Obsahuje možnost pro otevření Hypershader, vytvoření klávesové zkratky pro snadné spuštění skriptů a vytvoření tlačítka na jeho opětovné spuštění.
- **Help** - Nabízí informace o skriptech a přesměrování na distribuční stránku skriptů.

Výběr typu šálku a designu

Výběr typu šálku a designu obsahuje jak název, tak piktogram. Tak i neznalý uživatel kavárenského nádobí bude mít představu, ze kterých typů si vybírá a jak bude výsledný šálek vypadat.

Změna parametrů

Část GUI, kde si uživatel může nastavit jak rozměrové, tak doplňkové parametry specifikované v kapitole 3.2.3 a 3.2.4. Uživatel by měl při výběru rozměrových parametrů na první pohled vidět, jaké jsou krajní hodnoty pro jednotlivý rozměr. U doplňkových parametrů by mělo být zřejmé, které možnosti jsou vybrány a které nejsou pro daný typ aktivní.

4 Implementace

Pro implementaci MEL skriptů byl použit Charcoal Editor, který je v této kapitole popsán. Následuje struktura skriptů, jak jsou jednotlivé části nádobí modelovány, jaká jsou omezení pro vstupní parametry, možnosti rozšíření a popis použitých funkcí pro vytvoření GUI.

4.1 Charcoal Editor for Maya 2015

Charcoal Editor je plugin pro Autodesk Maya, který byl při implementaci skriptů použit. Je navržený tak, aby nahradil standardní skript editor. Je plně integrován do Maya pomocí Qt knihovny a Maya API, je multiplatformní a obsahuje mnoho pokročilých funkcí. Charcoal Editor odstraňuje potřebu komplexního externího editoru [10].

4.1.1 Funkce Charcoal Editoru:

- Zvýraznění syntaxe (MEL a Python)
- Rozšířené hledání a nahrazování
- Automatické ukládání skriptu při spuštění
- Inteligentní osazování
- Našeptávání
- Zvýrazňování chyb a varování
- Nativní C++ plugin [11]

4.2 Struktura skriptů

V souboru *CoffeeCupMaker.mel* je implementována hlavní funkčnost skriptů a celé GUI. Z tohoto souboru volá uživatel hlavní proceduru *CoffeeCupCreate()*, která postupně volá procedury potřebné pro generování. Ostatní skripty se už nemusí volat při každém spuštění, ale jen v případě jejich výběru. Skripty jsou strukturované do adresářů s názvem designu. V každém adresáři je šestnáct souborů, patnáct z nich generují jednotlivou část sady kavárenského nádobí. Šestnáctý nastavuje modifikační parametry pro každý typ. Všechny názvy souborů mají přesně definovaný název pro případné rozšiřování skriptů o typy šálků a jejich designy. Součástí sady skriptů je i adresář s logem a piktogramy pro každý design sady a jednotlivé typy šálků.

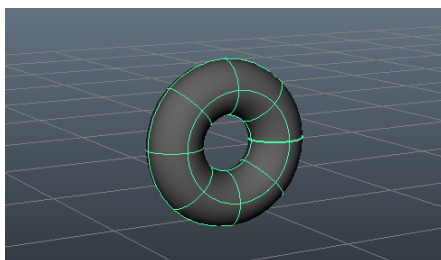
4.3 Princip modelování

Kavárenské nádobí se generuje za základě vstupních parametrů popsaných v kapitole 3.2.3 a jejich omezení nalezneme v kapitole 3.3. Pro každou část ze sady nádobí se využívá jiných modelovacích principů.

4.3.1 Modelování primitiv

Pro část modelů se využívá modelování pomocí primitiv. Maya nabízí vytvoření až osmi základních tvarů pomocí jednoduchých MEL příkazů. Ve skriptech je použito pouze vytvoření základního tvaru - prstence. Tento typ se využívá pro modelování uch šálků pro *espresso*, *cappuccino* a *caffè latte* u *minimalistického* a *kubistického* designu. Rozdíl mezi designy je ten, že u *minimalistického* designu je povrch ucha hladký a průřez je tvarem kruh, kdežto u *kubistického* je průřez uchem tvarem osmihran. Příkaz pro vytvoření ucha pro *minimalistické espresso*.

```
torus -n coffeeHandleSurface -ch on -po 0 -ax 1 0 0
      -p 0 (0.61*($height)) (-(($topRad)+(0.18 * $size)))
      -r (0.2 * $size) -nsp 8 -hr 0.5
```



Obrázek 16 Příklad pro modelování z primitiv

Na příkladu je vidět, že ucho může být pojmenováno (*-n*), můžeme se rozhodnout zda ponecháme konstrukční historii (*-ch*), zda bude povrch typu NURBS (*-po*), podle které osy bude prstenec vytvořen (*-ax*), kde nastavíme pozici pivotu prstence (*-p*), jaký bude poloměr (*-r*), počet polí které určují rozlišení povrchu (*-nsp*) a poměr výšky na šířce (*-hr*). Umístění pivotu prstence a jeho poloměr ovlivňují vstupní parametr *výšky* a *šířky*.

Ucho šálku na *caffè latte* je vytvořeno stejným principem, nevytváří se celý prstenec, ale pouze jeho polovina. Následně je ucho otočeno o úhel, který byl vypočítán ze zadaných parametrů pomocí goniometrické funkce, aby přesně navazoval na hranu šálku.

4.3.2 Modelování pomocí křivek

Ostatní modely jsou generovány pomocí povrchových metod, které jsou obecně popsány v kapitole 2.4.1. Jedná se o metody *revolve*, *lofting* a *extrude*.

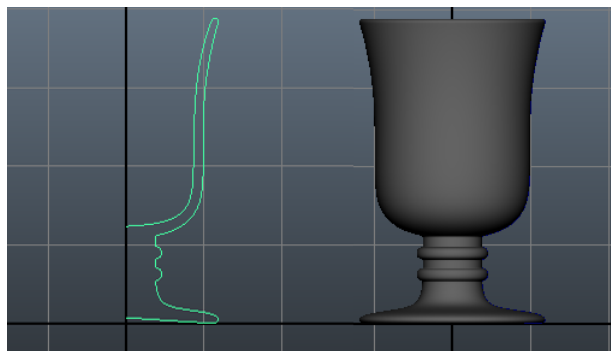
Revolve

Metoda *revolve* se používá u *minimalistických* typů šálků a podšálků a u kelímků na kávu *s sebou* včetně držátka a víčka na *espresso* a *caffè latte*.

Při vytváření příkazu pro modelovací křivku se vychází z reálného nádobí. Narýsuje se pomyslná křivka průřezem referenčního modelu. Tato křivka je překreslena do modeláře Maya a upravena podle reference. U rýsování křivky se dbá na co nejmenší počet řídicích bodů, bez ztráty detailu. U jednotlivých řídicích bodů je nutné zaměnit souřadnice za proměnné, které si může nastavit sám uživatel. Po vytvoření modelovací křivky se pak pomocí příkazu *revolve* modeluje povrch modelu.

4 Implementace

```
revolve -n coffeeCupSurface -ch 1 -po 0
      -ssw 0 -esw 360 -s 20 -ulp 1
      -ax 0 1 0 $ccmCupCurve;
```



Obrázek 17 Příklad pro modelování pomocí metody *revolve*

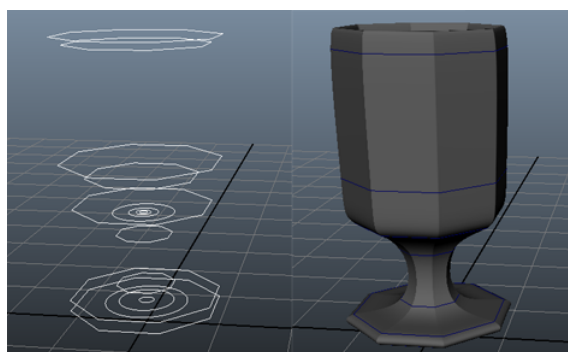
Na příkladu můžeme vidět, že výslednému objektu dáváme název (*-n*), aktivujeme konstrukční historii (*-ch*), vybíráme typ objektu (*-po*). Otáčíme křivku s názvem *\$ccmCupCurve* podle osy Y (*-ax*) o 360° (*-ssw*), (*-esw*). Dále nastavujeme počet sekcí (*-s*) a pivot umístíme do počátečního bodu osy otáčení (*-ulp*).

Lofting

Této metody se využívá u *kubistických* šálek a podšálek nebo u víček pro kávu *s sebou*, které nejsou symetrické dle osy.

Pro náš případ modelování není tato metoda příliš intuitivní. Implementace probíhala tak, že se podél hrany výšky referenčního obrázku umisťovali křivky ve vzdálenosti podle detailu. Tyto křivky jsou kolmé k ose výšky, jsou uzavřené a tvarem buď osmihran či kružnice. Poloha a poloměr těchto křivek je dán vstupními parametry. První a poslední vytvořená křivka je má nulový poloměr. Při výběru všech vytvořených křivek pak pomocí příkazu *loft* vytvoříme část nádobí.

```
loft -n coffeeCupSurface
     -ch 1 -ss 2 -po 0 -rsn 1
     $circle0[0] $circle1[0] $circle2[0] $circle3[0];
```



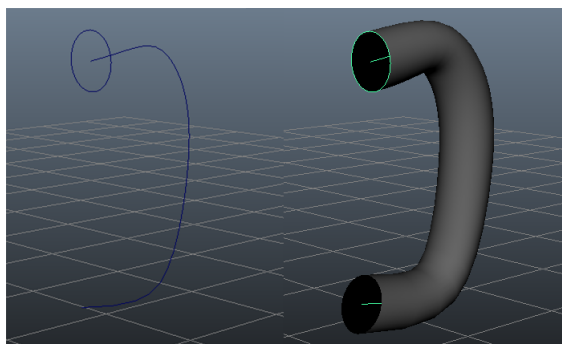
Obrázek 18 Příklad pro modelování pomocí metody *lofting*

Na příkladu vidíme, že vytváříme *lofting* pro křivky $\$circle0[0]$ až $\$circle3[0]$. Stejnými příznaky jako v předchozích příkladech objekt pojmenujeme $(-n)$, zapneme konstrukční historii $(-ch)$. Nastavíme počet ploch mezi po sobě jdoucími křivkami $(-ss)$, zvolíme typ objektu $(-po)$ a výsledný povrch objektu převrátíme $(-rsn)$.

Extrude

Metoda *extrude* se používá pro ucha šálek, které nebyly vytvořeny pomocí primitivních tvarů. Podle referenčních obrázků se vytvoří směrová a profilová křivka. Směrová křivka určí, jak detailní bude ucho šálku a profilová šířku a tvar ucha. Profilových křivek máme pro naši implementaci dva druhy. Jedna křivka je klasická kružnice, využívaná v *minimalistickém* designu a druhá křivka je tvarem osmihran, pro design *kubistický*.

```
extrude -n coffeeHandleSurface -ch 1
      -po 0 -et 2 -fpt 1 -rsp 1
      $circleHandle [0] $handleCurve;
```



Obrázek 19 Příklad pro modelování pomocí metody *extrude*

Příklad metody *extrude* ukazuje, že využívá profilové a směrové křivky $\$circleHandle[0]$ a $\$handleCurve$. Výsledný objekt pojmenujeme $(-n)$, zapneme konstrukční historii $(-ch)$, nastavíme výsledný objekt na typ NURBS $(-po)$. Tvar objektu bude trubice $(-et)$ s umístěním na dráze křivky $(-fpt)$ a převrátíme povrch objektu $(-rsp)$.

4.4 Možnosti rozšíření

Sada skriptů je programována tak, aby byl snadný a přehledný způsob, jak přidávat nové objekty ke generování. V hlavní proceduře *CoffeeCupCreate()* je inicializované pole *stringů* s názvy typů šálek a typů designu. Jednotlivé pole se prochází v cyklech. V těchto cyklech se díky jmenné konvenci volají generující a modifikační procedury, piktogramy a jednotlivé MEL soubory.

Nový design se přidá do skriptů inkrementací pole o nový název designu a přiložení adresáře s MEL soubory, které generují všechny části sady pro každý dosavadní typ. Do složky *CCM* je možné nahrát piktogram nového designu.

Pro rozšíření skriptů o nový typ šálku je nutné inkrementovat pole typů o nový název a přidat do každého adresáře s designy MEL soubory, které jsou potřeba ke generování nového typu šálku. Do složky *CCM* je též možné nahrát piktogram nového typu.

4.5 Grafické uživatelské rozhraní

GUI pro sady skriptů je složeno z několika jednotlivých částí.

Menu pod hlavní lištou

Menu je složeno ze třech záložek. Pro toto menu není potřeba volit rozvržení.

- **Window** - umožní volat proceduru *windowDock*, která ukotví okno skriptu do rozhraní modeláře Maya. Dále může volat příkaz *deleteUI*, který okno zavírá.
- **Tools** - umožní zavolat proceduru *HupershadeWindow*, která otevře Hypershader, ve kterém můžeme přidat vygenerovaným sadám materiály. Dále může zavolat metodu *ccmHotkey*, která otevře okno pro nastavení klávesové zkratky pro snadné spuštění skriptu. V této liště je další možnost vytvoření tlačítka, které se uloží do aktivní záložky v modeláři Maya. Po kliknutí tlačítka spustí stejný příkaz jako při prvním spuštění skriptů - *source CoffeeCupMaker.mel; CoffeeCupCreate()*;
- **Help** - otevírá okno o informacích ohledně skriptů. V tomto okně se využívá rozložení *scrollField*. Dále volá příkaz *system("\shell start...)*, který přesměruje uživatele na webové stránky skriptů.

Výběr typu šálku a designu

V této části GUI se využívá rozvržení *gridLayout* pro pět sloupců. Výběr typu je realizován pomocí příkazu *iconTextRadioButton*, který nám umožní přidat k přepínacímu tlačítku (radio button) piktogram. Tlačítka se přidávají pomocí cyklu, kvůli snadnému rozšíření a orientaci ve skriptech.

Změna parametrů

Jedná se o část GUI, kde se využívá rozvržení *formLayout*, kvůli snadné editaci rozmístění prvků. Příkazy *radioButtonGrp* a *checkBoxGrp* umožní vytvořit jednoduché přepínací a zaškrtačací tlačítka. V druhé polovině této části se používá rozvržení *gridLayout* pro jeden sloupec. Aby uživatel na první pohled viděl mezní hodnotky pro jednotlivé typy šálků, je zde použit příkaz *floatSliderGrp*, díky kterému se vytvoří posuvník s jinými hodnotami pro každý typ.

Aby skripty reagovali na změnu parametrů, je po každé změně volána procedura *ccmRedrawUI()*, která přenastaví GUI podle uživateli změny.

Tlačítka pro generování

po zmáčknutí tlačítka *Create* se zavolá procedura *ccmCreateCmd()*. Pomocí této procedury se načítají a spouštějí potřebné soubory a procedury, které generují kuchyňské nádoby na základě vstupních parametrů.

Tlačítko *Apply* smaže poslední vytvořenou sadu nádobí a znova spustí proceduru *ccmCreateCmd()* stejně jako po zmáčknutí tlačítka *Create*.

4.6 Distribuce skriptů

Vytvořené skripty jsou prezentovány a distribuovány na webových stránkách:

<https://www.creativecrash.com/maya/script/coffee-cups-maker>.

Tyto stránky jsou největší internetovou databází skriptů a pluginů pro 3D a 2D grafiku. Nachází se zde velké množství tutoriálů, fór, nabídek prací a novinek ze světa počítačové grafiky. Díky umístění na server s širokou základnou uživatelů je pravděpodobné, že si mnou vytvořené skripty stáhne a ohodnotí velké množství uživatelů.

5 Testování

V kapitole testování se porovnají vybrané modely, které byly pomocí skriptů vygenerovány, s referenčními obrázky. Dále se hodnotí škála variability.

K renderování všech modelů se využívá *mentalRay* render, materiály jsou použity *mia_material_x_passes*.

5.1 Porovnání výsledků



a) Reference



b) Vysledný render

Obrázek 20 Porovnání *minimalistického* šálku na *espresso*



a) Reference



b) Vysledný render

Obrázek 21 Porovnání *minimalistického* šálku na *caffè latte*



a) Reference



b) Vysledný render

Obrázek 22 Porovnání *minimalistického* šálku na *irskou kávu*



a) Reference



b) Vysledný render

Obrázek 23 Porovnání *minimalistického klasického hrnku*



a) Reference



b) Vysledný render

Obrázek 24 Porovnání *kubistického klasického hrnku s podšálkem*



a) Reference



b) Vysledný render

Obrázek 25 Porovnání *kubistického klasického hrnku*



a) Reference



b) Vysledný render

Obrázek 26 Porovnání kelímků na *cappuccino s sebou*



a) Reference



b) Vysledný render

Obrázek 27 Porovnání klasického kelímku na kávu s sebou

5.2 Škála variability

Skripty umožňují generovat pět typů šálků na kávový nápoj - *espresso*, *cappuccino*, *caffè latte*, *irskou kávu* a *klasický hrnek*. U těchto šálků lze vybrat ze tří druhů designu - *minimalistický*, *kubistický* a šálky na kávu s sebou.



Obrázek 28 Škála variability pro *minimalistický* design



Obrázek 29 Škála variability pro *kubistický* design

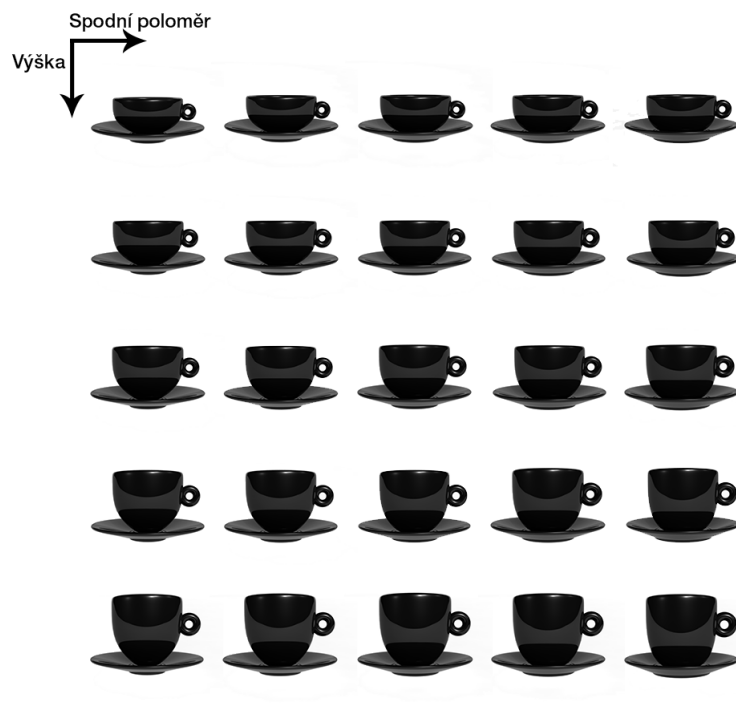


Obrázek 30 Škála variability pro šálky na kávu s sebou

U každého typu šálku je možnost nastavení rozměrových parametrů - *výška*, *vrchní* a *spodní poloměr*. Vstupní proměnné *vrchního* a *spodního poloměru* jsou na sobě závislé viz kapitola 3.3. Z tohoto důvodu u hodnot, které jsou omezeny není výsledný model šálku. Pro ukázkou variability použijeme šálek na *minimalistické espresso*.



Obrázek 31 Variabilita při změně *vrchního* a *spodního poloměru* pro *minimalistické espresso*



Obrázek 32 Variabilita při změně *spodního poloměru* a *výšky* pro *minimalistické espresso*



Obrázek 33 Variabilita při změně *výšky* a *vrchního poloměru* pro *minimalistické espresso*

6 Závěr

Výstupem bakalářské práce je dle zvoleného cíle vytvoření sady skriptů, které umožní procedurálně modelovat typy kávových šálek o různých tvarech na základě vstupních parametrů v modeláři Maya.

Písemná část práce obsahuje důležité teoretické základy pro realizaci podobných skriptů využívajících stejných funkcionalit. Popsány byly i hlavní modelovací techniky pro modelování z NURBS křivek, základy procedurálního modelování a seznámení se skriptovacím jazykem MEL.

Největším přínosem skriptů je časová úspora a zjednodušení práce pro umělce, kteří modelují interiéry. Skripty jim nabízí možnost rychlého vygenerování šálek podle jejich představ.

Skripty mohou být rozšiřovány o nové funkce, například přidání šálek pro jiné druhy kávy, či nové designy. Nabízí se i možnost rozšíření skriptů o druhově jiné nádobí než kavárenské. Z tohoto důvodu jsou skripty psány tak, aby následné rozšíření bylo co nejjednodušší a uživatel nemusel znát strukturu skriptů.

Zvolené metody pro tvorbu kavárenského nádobí pomocí procedurálního modelování se na základě testování jeví jako ideální. Při porovnání výsledných renderů a referencí je vidět, že jsou si velice podobné. Variabilita výsledného nádobí je postačující pro většinu kávových nápojů a pomocí rozměrových parametrů je možné měnit jejich tvar na požadovaný výsledek.

Na přiloženém DVD nosiči jsou umístěny samotné skripty a návod na jejich spuštění. Skripty jsou dále prezentovány a distribuovány na webových stránkách největší internetové databáze skriptů a pluginů pro práci s 3D a 2D grafikou. Odkaz na internetové stránky vytvořených skriptů: <https://www.creativecrash.com/maya/script/coffee-cups-maker>. Během prvních čtrnácti dnů (publikováno 1. 5. 2016) si skripty stáhlo 82 lidí, 4 lidé je ohodnotili a všichni kladně.

Příloha A

Spuštění skriptů

Návod na spuštění skriptů:

1. Ukončete program Maya.
2. Zkopírujte soubor *CoffeeCupMaker.mel* a adresáře *Minimal*, *Cubist* a *Takeaway* do `C:\Users\OWNER\Documents\maya\VERSION\scripts`.
3. Zkopírujte adresář *CCM* do `C:\Users\OWNER\Documents\maya\VERSION\prefs\icons`.
4. Spusťte program Maya.
5. Do příkazového řádku Maya (nebo do skript editoru) vložte následující příkazy a zmáčkněte Enter nebo klikněte na tlačítko Execute All.

```
source CoffeeCupMaker.mel; CoffeeCupCreate ();
```

Příloha B

Obsah přiloženého DVD

Na přiloženém DVD nosiči se nachází:

DVD/	
├─ ReadMe.txt	textový soubor se základními informacemi o skriptu a návodem na spuštění
├─ CCM/	adresář s piktogramy a logem skriptů
├─ Minimal/	adresář s MEL soubory pro generování všech objektů pro minimalistický design
├─ Cubist/	adresář s MEL soubory pro generování všech objektů pro kubistický design
├─ Takeaway/	adresář s MEL soubory pro generování všech objektů pro kávu s sebou
└─ CoffeeCupMaker.mel	hlavní MEL soubor

Literatura

- [1] M. E. Yumer et al. “Procedural Modeling Using Autoencoder Networks”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)* (2015).
- [2] Ruben M. Smelik et al. “A Survey on Procedural Modelling for Virtual Worlds”. In: *Comput. Graph. Forum* (2014).
- [3] Autodesk Inc. *About Autodesk*. [online]. [cit. 2016-05-21]. URL: <http://www.autodesk.com/company>.
- [4] Autodesk Academia Silicon Hill. *Autodesk Maya*. [online]. [cit. 2016-05-05]. URL: <http://autodesk.siliconhill.cz/produkty/>.
- [5] Gould David. *Complete Maya Programming: An Extensive Guide to MEL and C++ API*. San Francisco: Morgan Kaufmann Publishers, 2003. ISBN: 1-55860-835-4.
- [6] Dariush Derakhshani. *Maya : průvodce 3D grafikou*. Praha: Grada Publishing, 2006. ISBN: 80-247-1253-9.
- [7] Mark R. Wilkins, Chris Kazmier. *MEL scripting for Maya animators*. San Francisco: Morgan Kaufmann Publishers, 2005. ISBN: 0-12-088793-2.
- [8] Dariush Derakhshani. *Introducing Autodesk Maya 2014*. Canada: Neil Edde, 2013. ISBN: 978-1-118-57490-4.
- [9] Jim Lammers, Lee Gooding. *Maya 4 - kompletní průvodce*. Praha: SoftPress, 2002. ISBN: 80-86497-30-5.
- [10] Chris Zurbrigg. *Charcoal Editor*. [online]. [cit. 2016-05-05]. URL: <http://zurbrigg.com/charcoal-editor>.
- [11] Lester Banks. *Charcoal Editor for Maya replaces Maya’s Internal Script Editor Providing Modern IDE Features*. [online]. [cit. 2016-05-05]. URL: <http://lesterbanks.com/2012/11/charcoal-editor-for-maya-replaces-mayas-internal-script-editor-providing-modern-ide-features/>.