



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS CRATEÚS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Geração procedural de modelos arquiteturais com geometria arredondada utilizando *Selection Expressions (SELEX)*

Daniel Henrique de Brito

Orientador: Arnaldo Barreto Vila Nova

Coorientador: Ítalo Mendes da Silva Ribeiro

Crateús
Abril - 2021

SUMÁRIO

1. Introdução
2. Fundamentação teórica
3. Trabalhos correlatos
4. Proposta
5. Resultados
6. Conclusão
7. Referências

1 - Introdução

Geração procedural de modelos arquiteturais com geometria arredondada utilizando *Selection Expressions* (SELEX)

1 - Introdução

1.1 - Contextualização

1.2 - Justificativa

1.3 - Objetivo geral

1.4 - Objetivos específicos

1.1 - Contextualização

- Modelagem procedural
 - Aplicações:
 - Indústria cinematográfica
 - Indústria de jogos
 - Simulações
 - Planejamento urbano
 - Logística

1.1 - Contextualização

- Modelagem procedural
 - Vantagens:
 - Geração de uma grande variedade de modelos baseados em diferentes parâmetros
 - Diminuição da carga de trabalho para geração dos modelos
 - Desvantagens:
 - Falta de intuitividade na utilização de alguns *frameworks*
 - Dificuldade na criação das gramáticas

1.2 - Justificativa

- Geração procedural de edifícios
- Motivação para escolha da *SELEX*:
 - Técnica relativamente recente
 - Evolução em relação à *CGA Shape* e *CGA++*
- Desafio:
 - Modelagem de arquiteturas arredondadas

1.3 - Objetivos geral

Produzir modelos arquiteturais com geometria arredondada utilizando *Selection Expressions*, por meio da especificação de uma nova operação de deformação.

Identificador da figura no trabalho:

67

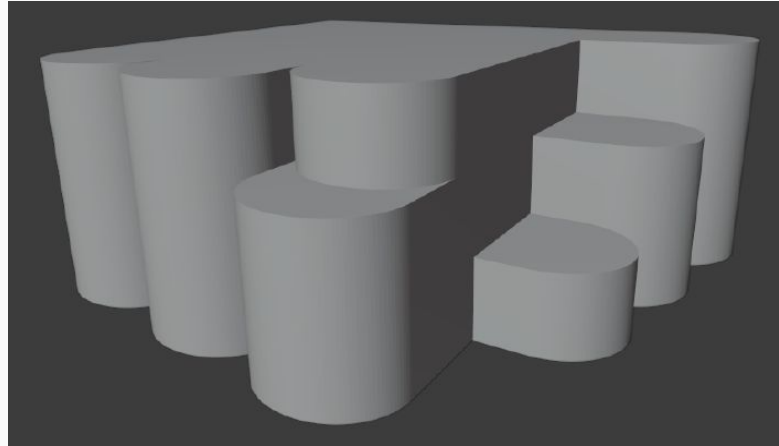


Figura 1: Modelo de massa gerado pela solução proposta.
Fonte: Próprio autor

1.4 - Objetivos específicos

- Implementar e avaliar a linguagem para geração de modelos de massa com arquitetura arredondada, ou seja, com bordas suavizadas;
- Integrar linguagem com ferramenta de modelagem 3D por meio de *scripts*;
- Avaliar a aplicação de técnicas de deformação para criação de modelos arquiteturais com geometria arredondada;
- Avaliar o resultado obtido frente a alguns exemplos do mundo real.

2 - Fundamentação teórica

2 - Fundamentação teórica

2.1 - Geração procedural

2.2 - Primórdios

2.3 - *CGA Shape*

2.4 - *CGA++*

2.5 - *Selection Expressions (SELEX)*

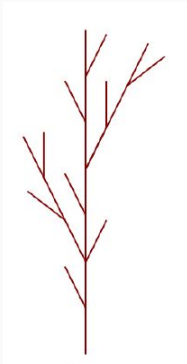
2.6 - Deformação

2.1 - Geração procedural

- Conceito:
 - Técnicas procedurais são segmentos de código que especificam algumas características de um modelo ou efeito gerado por computador (*EBERT et al., 2002*).
- Ferramentas:
 - Maya, Houdini, City Engine
- Técnicas

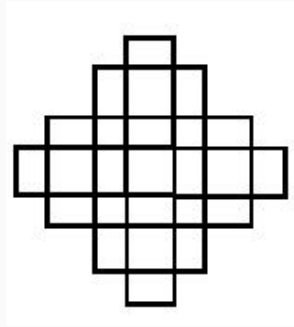
2.2 - Primórdios

- L-Systems
- Shape Grammars
- Split Grammars



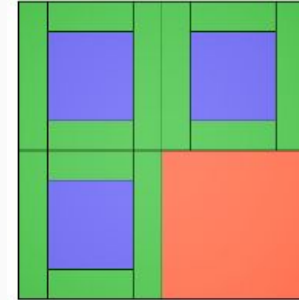
78

Figura 2: Modelo de alga.
Fonte: (SIMON, 2011)



80

Figura 3: Planta de igreja.
Fonte: (KNIGHT, 1995)



85

Figura 4: Modelo de fachada.
Fonte: (RODRIGUES, 2014)

2.3 - CGA Shape

A *CGA Shape* foi proposta por Müller et al. (2006) para geração procedural de modelos arquiteturais, trazendo melhorias em relação às *split grammars*.

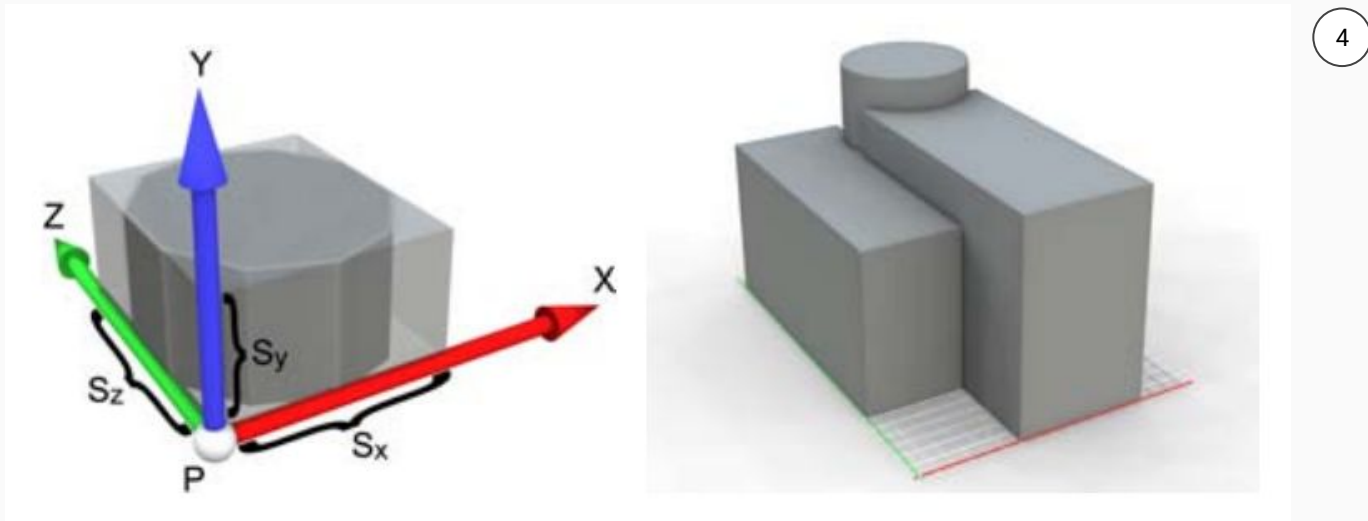


Figura 5: Representação geométrica da *CGA Shape*.
Fonte: (MÜLLER et al., 2006)

2.4 - CGA++

A *CGA++* foi introduzida por Schwarz e Müller (2015) como sendo uma evolução natural da *CGA Shape*, com o objetivo de superar algumas limitações existentes na modelagem procedural de arquiteturas.

2.4 - CGA++

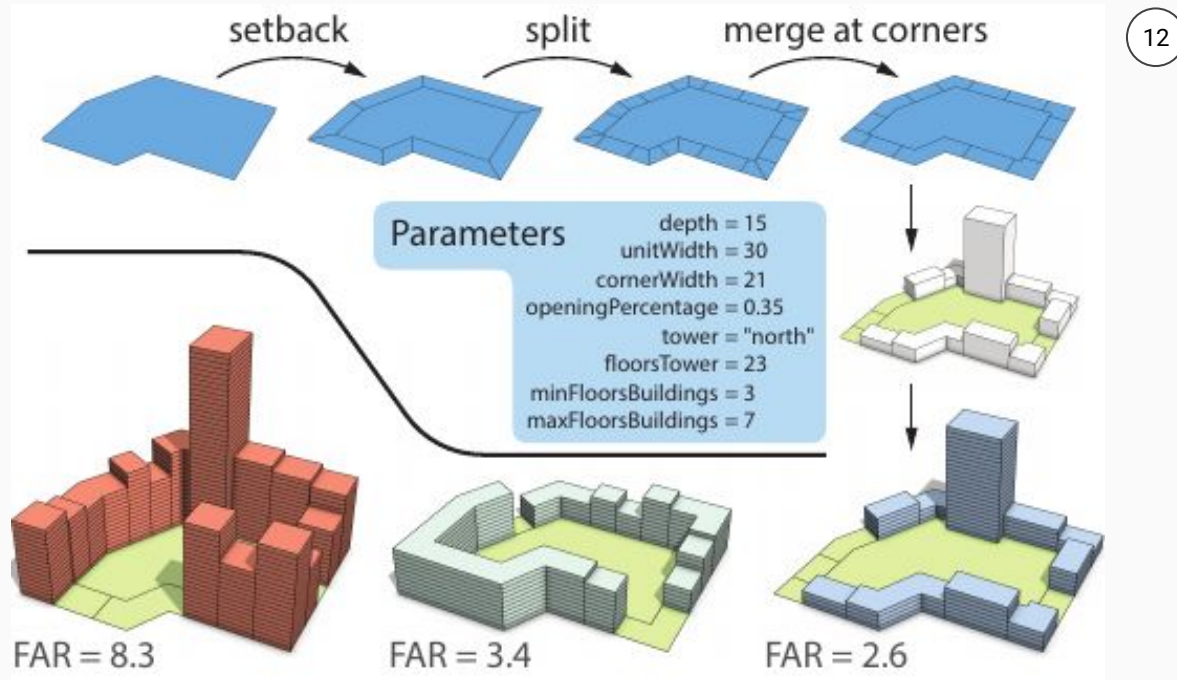


Figura 6: Representação de bloco perimetral.
Fonte: Adaptado de (SCHWARZ; MÜLLER, 2015)

2.5 - Selection Expressions (SELEX)

A *SELEX* é uma nova abordagem para geração procedural que foi introduzida por *Jiang et al. (2018)*, e tem como ideia principal selecionar um subconjunto de formas utilizando *selection-expressions*, visando melhorar dois problemas predominantes nas abordagens utilizadas na *CGA Shape* e *CGA++*:

- Limitação de coordenação dos elementos;
- Hierarquia de divisão dos elementos.

2.5 - Selection Expressions (SELEX)

- Especificações:
 - Conceito
 - Definições de forma:
 - Formas de construção
 - Formas virtuais
 - Linguagem:
 - Regra:
`selection-expression → action;`
 - Atribuição:
`identifier = expression;`

2.5 - Selection Expressions (SELEX)

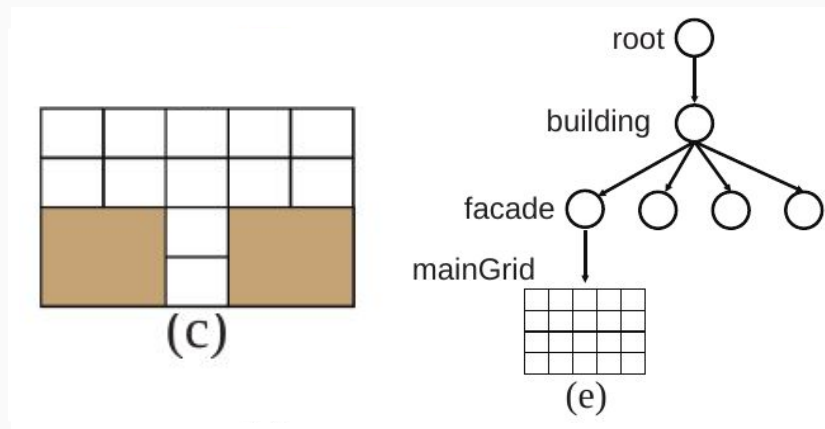
- Especificações:
 - Configuração de uma *selection-expression*:
`<[topoS] [attrS | groupS] * / [topoS] [attrS | groupS] * / . . . >`
 - Seletores:
 - Seletor de topologia: *child()*, *parent()*, *root()*, *neighbor()*
 - Seletor de atributo: *[label="facade"]*
 - Seletor de grupo: *groupRows()*, *groupCols()*, *groupRegions()*

2.5 - Selection Expressions (SELEX)

- Especificações:

- Exemplo:

c) `<descendant()[label=="facade"]/[label=="mainGrid"]/[type=="cell"]
[rowIdx in (3,4)][colIdx in (1,2,4,5)][::groupRegions()]>`

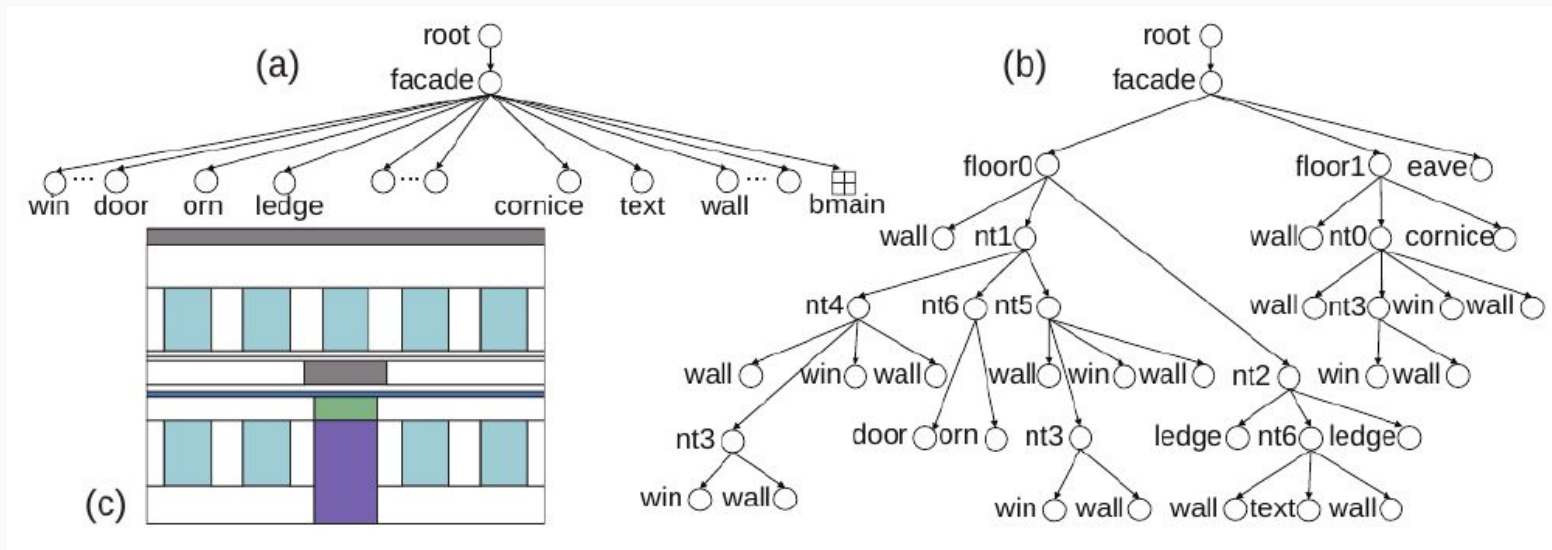


22

Figura 7: Exemplo da utilização de seletores.

Fonte: Adaptado de (Jiang et al., 2018)

2.5 - Selection Expressions (SELEX)



27

Figura 8: Exemplo da árvore de formas gerada pela SELEX (a), e pela CGA Shape (b), com base na fachada (c).
Fonte: (Jiang et al., 2018)

2.5 - Selection Expressions (SELEX)

29

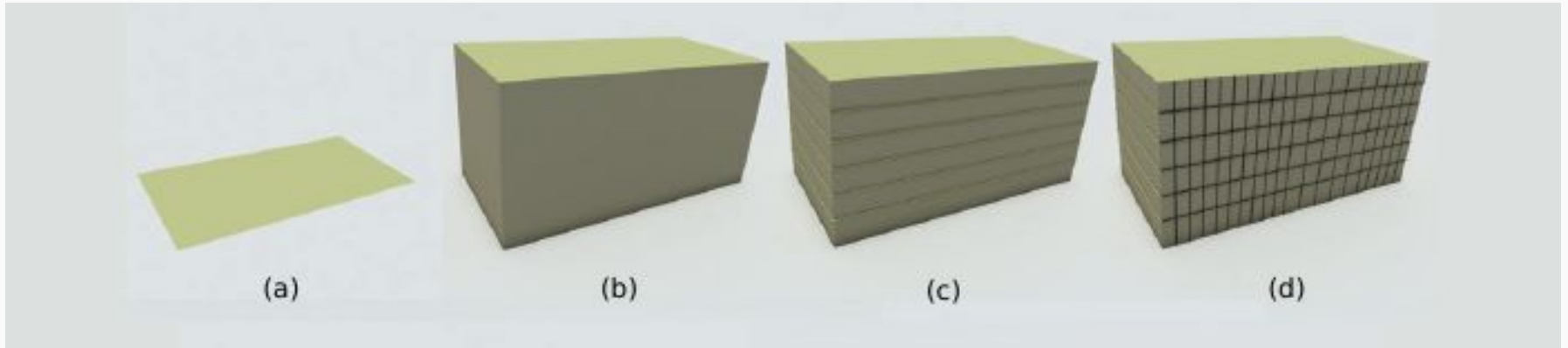


Figura 9.1: Exemplo de modelagem utilizando *SELEX*.

Fonte: Adaptado de (*Jiang et al., 2018*)

2.5 - Selection Expressions (SELEX)

29

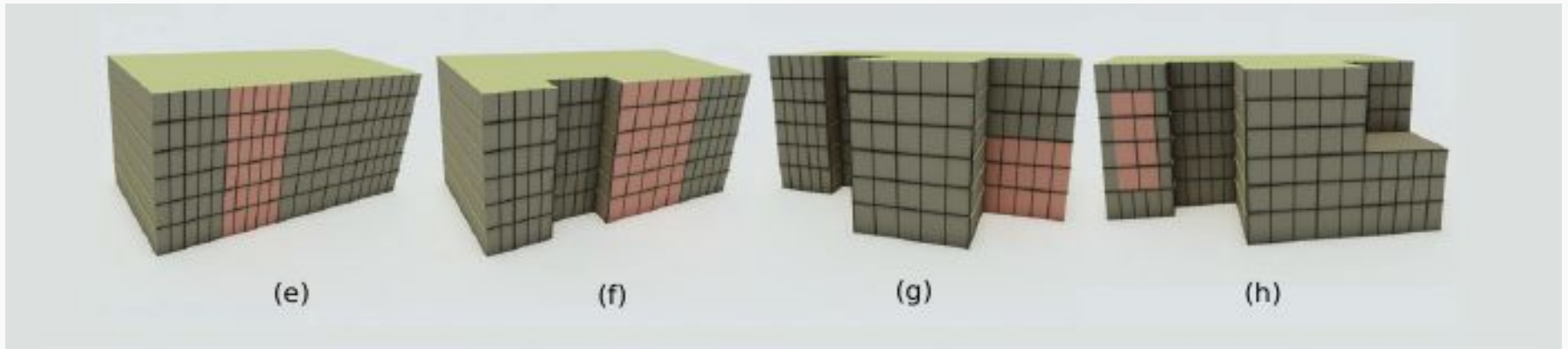


Figura 9.2: Exemplo de modelagem utilizando *SELEX*.

Fonte: Adaptado de (Jiang et al., 2018)

2.5 - Selection Expressions (SELEX)

29

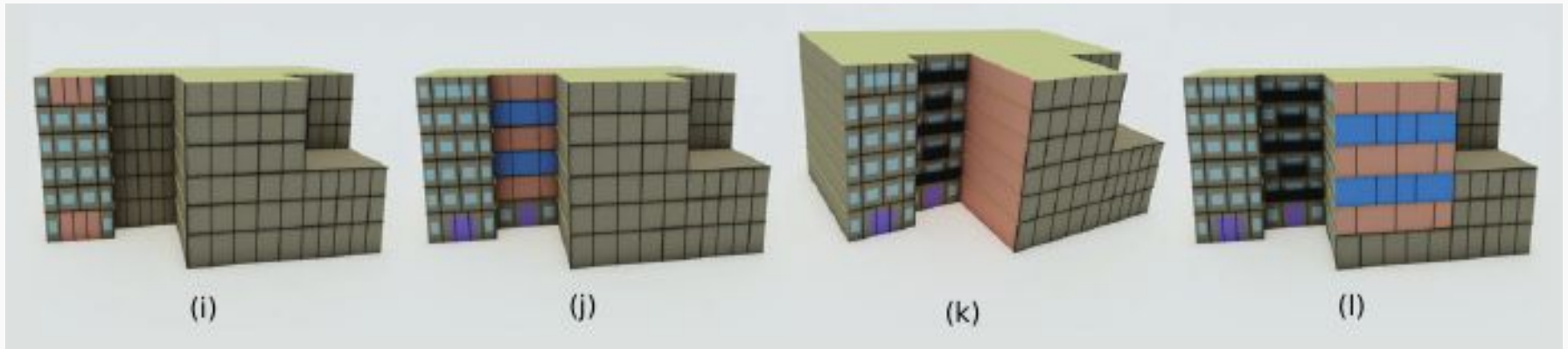


Figura 9.3: Exemplo de modelagem utilizando *SELEX*.

Fonte: Adaptado de (Jiang et al., 2018)

2.5 - Selection Expressions (SELEX)

29

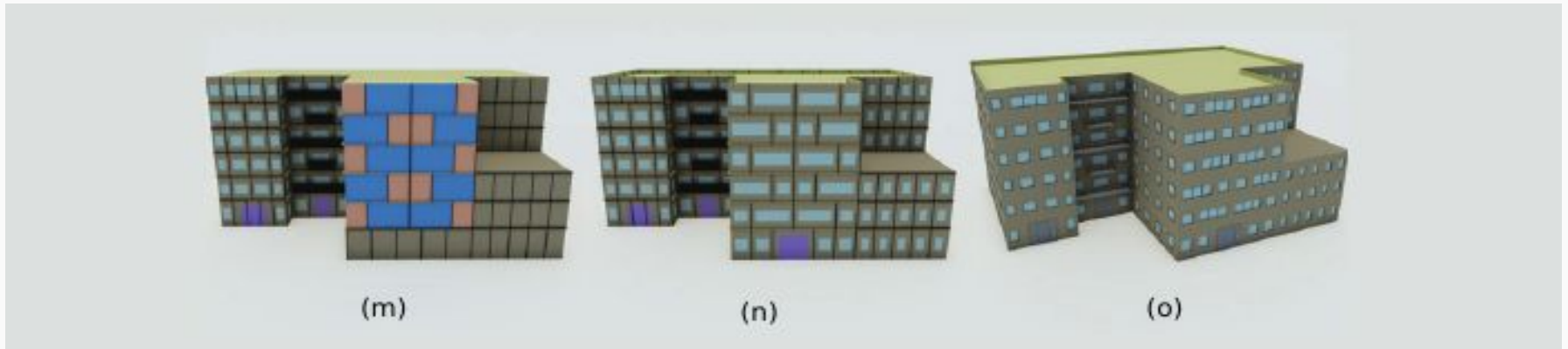


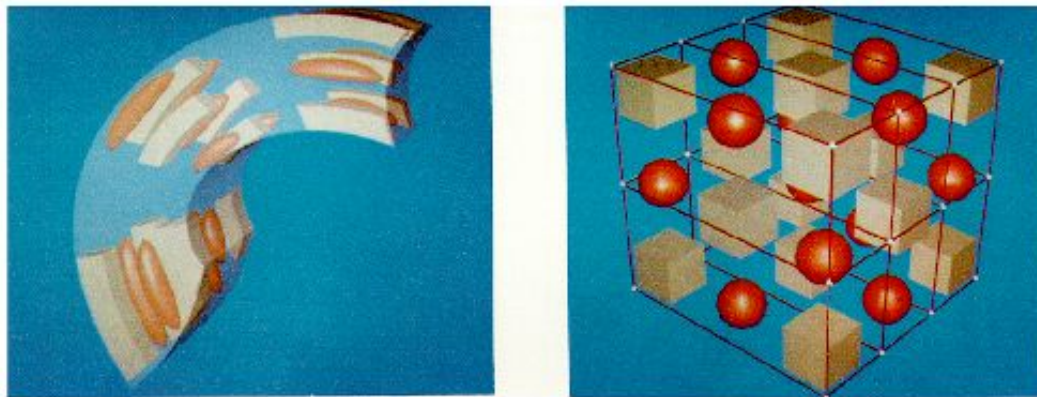
Figura 9.4: Exemplo de modelagem utilizando *SELEX*.
Fonte: Adaptado de (Jiang et al., 2018)

2.6 - Deformação

No campo da física, a deformação de uma estrutura é qualquer mudança da configuração geométrica do corpo que leve à uma variação da sua forma ou das suas dimensões após a aplicação de uma ação externa (*TRUESDALL; NOLL, 1992*).

2.6 - Deformação

Na deformação de forma livre, idealizada por *Sederberg e Parry (1986)*, é definida uma grade regular de pontos de controle. Ao deslocar esses pontos de controle, uma deformação do espaço é alcançada.



31

Figura 10: Exemplo de deformação de forma livre.

Fonte: (SEDERBERG; PARRY, 1986)

2.6 - Deformação

As *deformation grammars* foram introduzidas por Vimont et al. (2017), permitindo deformar livremente objetos complexos ou conjuntos de objetos, preservando sua consistência.



Figura 11: (a) O modelo inicial de uma casa (b) é deformado pelo usuário.

Fonte: (VIMONT et al., 2017)

3 - Trabalhos correlatos

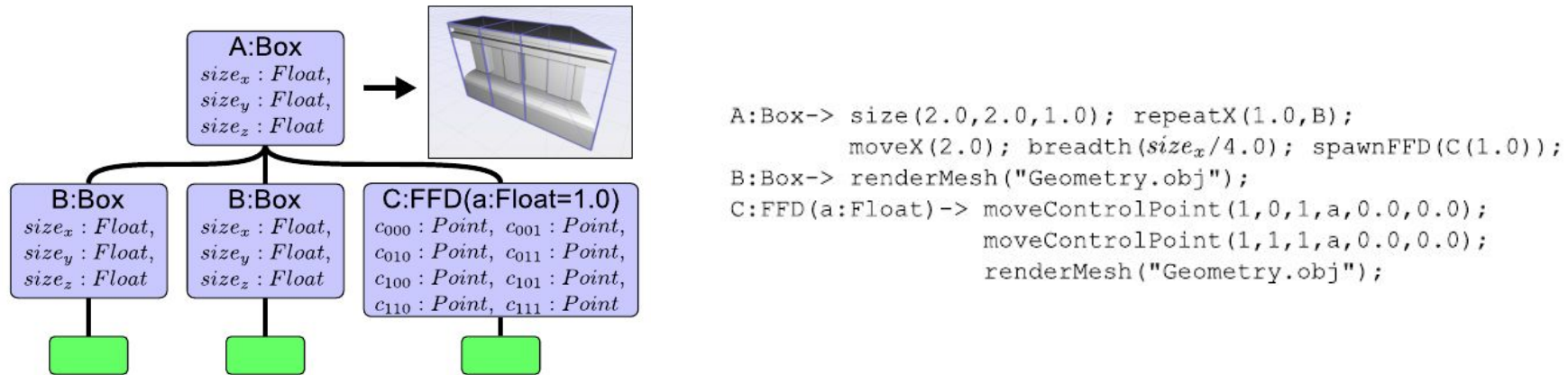
Geração procedural de modelos arquiteturais com geometria arredondada utilizando *Selection Expressions* (SELEX)

3 - Trabalhos correlatos

- 3.1 - *Generalized Use of Non-Terminal Symbols for Procedural Modeling*
(KRECKLAU et al., 2010)
- 3.2 - *Procedural architecture using deformation-aware split grammars*
(ZMUGGet al., 2014)
- 3.3 - *Procedural modeling of architecture with round geometry*
(EDELSEBRUNNER et al., 2017)

3.1 - Generalized Use of Non-Terminal Symbols for Procedural Modeling

A abordagem de Krecklau et al. (2010) introduz a linguagem de modelagem procedural G^2 , que utiliza deformação de forma livre como um objeto não-terminal alternativo para superar a desvantagem da criação de objetos arredondados.



38

Figura 12: Aplicação de regras de modelagem da G^2 .

Fonte: Adaptado de (KRECKLAU et al., 2010)

3.2 - Procedural architecture using deformation-aware split grammars

Uma extensão às *split grammars* é apresentada por Zmugg et al. (2014), permitindo a criação de arquiteturas curvadas através da integração de deformações de forma livre em qualquer nível de uma gramática.

45

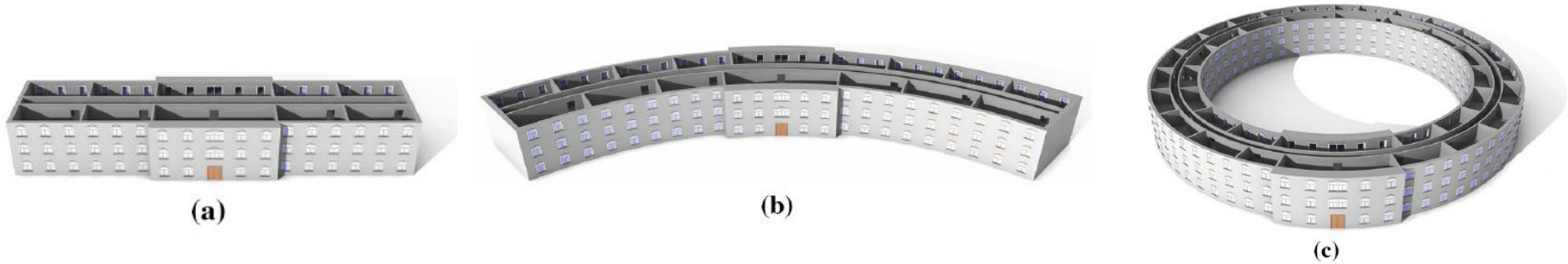


Figura 13: Prédio comercial com estrutura arredondada.
Fonte: Adaptado de (ZMUGG et al., 2014)

3.3 - Procedural modeling of architecture with round geometry

No trabalho de *Edelsbrunner et al. (2017)* são especificados sistemas de coordenadas personalizados na *split grammar* definida pelo usuário.

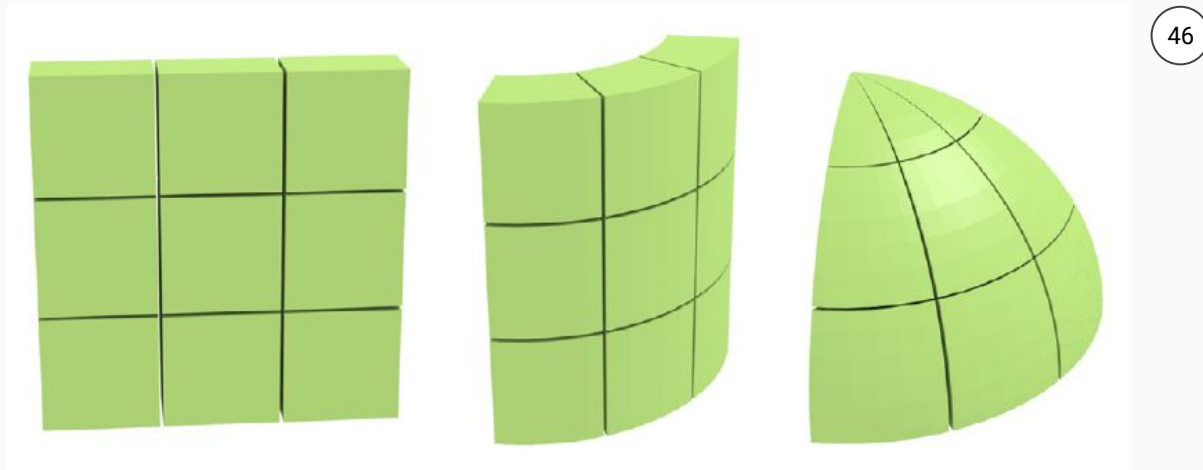


Figura 14: Divisão de formas com diferentes sistemas de coordenadas (cartesiana, cilíndrica e esférica).
Fonte: (EDELSBRUNNER et al., 2017)

4 - Proposta

4 - Proposta

4.1 - Problema

4.2 - Ferramentas de modelagem

4.3 - Fluxograma de modelagem

4.4 - Módulos de implementação

4.1 - Problema

Conforme mencionado por *Jiang et al. (2018)*, uma das limitações de implementação da *SELEX* é a incapacidade de modelar estruturas arredondadas diretamente, trabalhando apenas por meio de sua importação, como complementos, o que impossibilita a modelagem de fachadas curvadas.

4.1 - Problema



49

Figura 15: Exemplo que está além da capacidade de modelagem da *SELEX*.

Fonte: (Jiang et al., 2018)

4.2 - Ferramentas de modelagem

- **Blender:** Software *open source* de modelagem 3D.
- **Python:** Linguagem de programação.

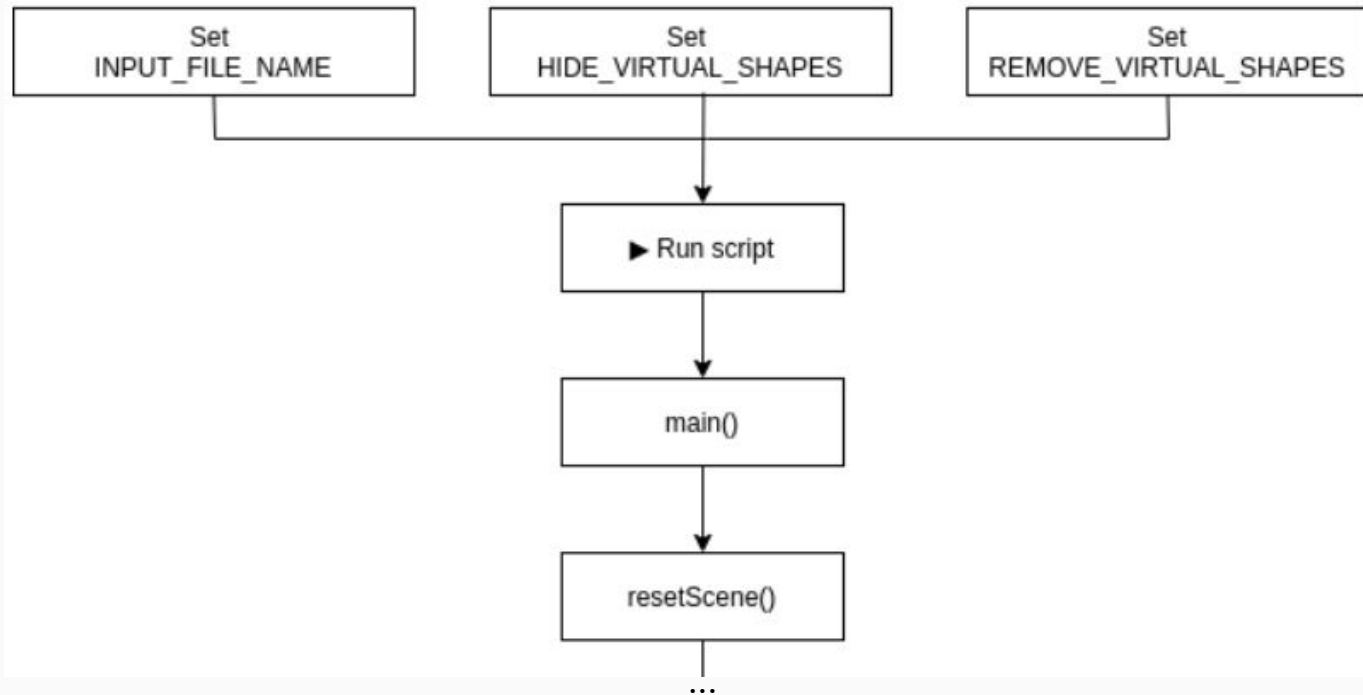


Figura 16: Blender logo.
Fonte: (blender.org, 2021)



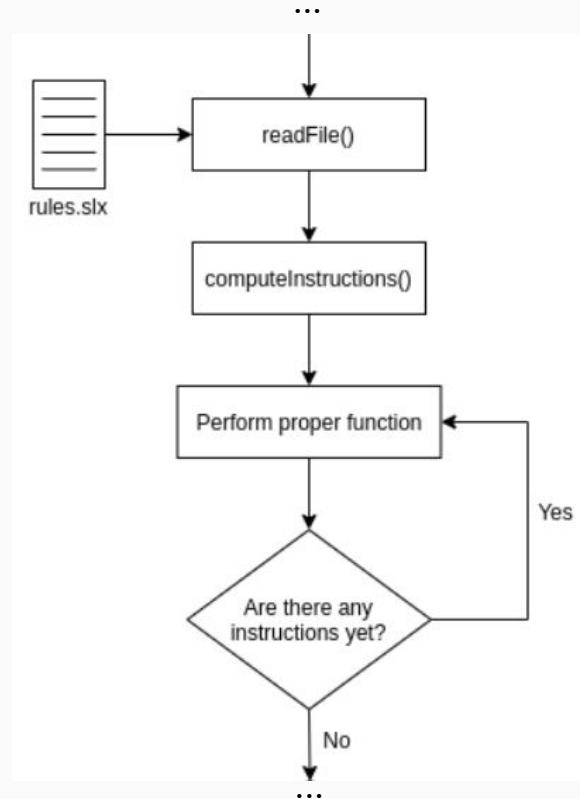
Figura 17: Python logo.
Fonte: (python.org, 2021)

4.3 - Fluxograma de modelagem



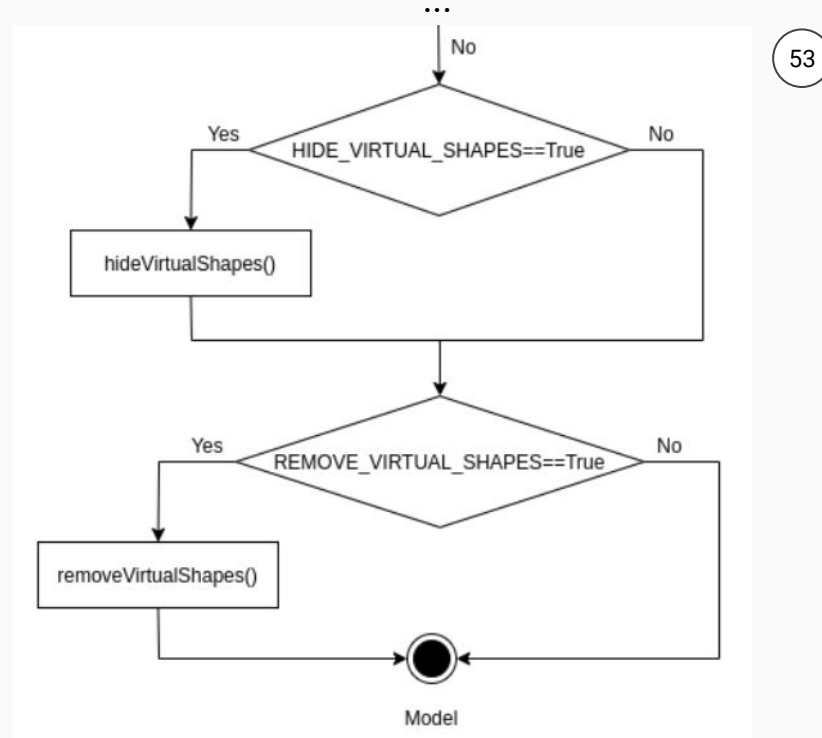
53

4.3 - Fluxograma de modelagem



53

4.3 - Fluxograma de modelagem



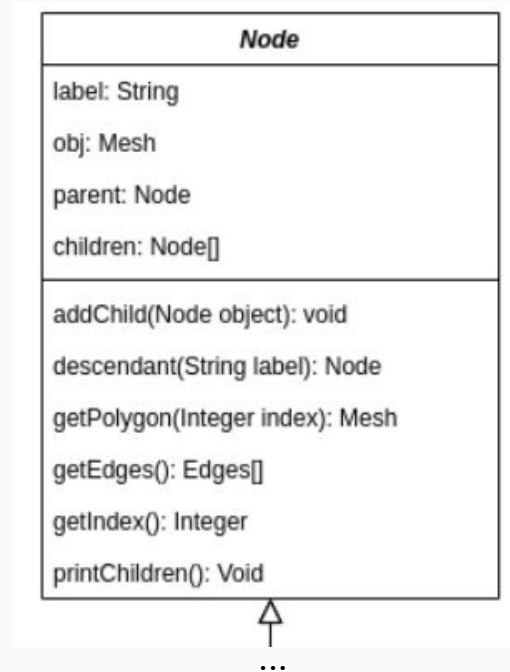
Fonte: Próprio autor

4.4 - Módulos de implementação

- ***Imported Libraries:***
 - **bpy:** Fornece acesso aos dados, classes e funções do Blender.
 - **math:** Fornece acesso às funções matemáticas.
 - **os:** Fornece acesso a recursos do sistema operacional.
 - **re:** Fornece operações para correspondência de expressões regulares.

4.4 - Módulos de implementação

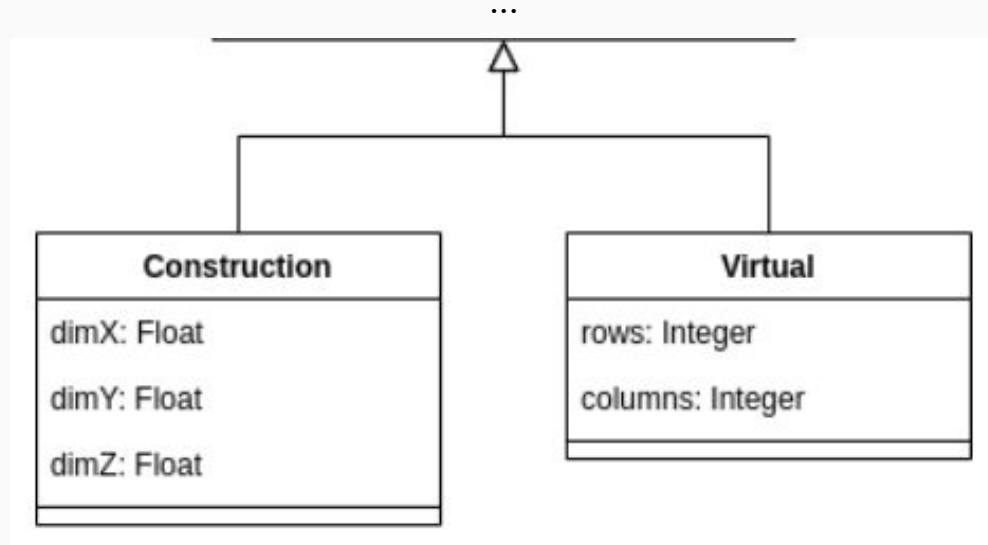
- **Classes module:**



54

4.4 - Módulos de implementação

- **Classes module:**



54

Figura 18: Diagrama de classes.

Fonte: Próprio autor

4.4 - Módulos de implementação

- ***Utility Functions Module:***
 - *create3DMass*
 - *vert e face*
 - *selectNode*
 - *placeVirtualShape*
 - *selectToBeVolume e duplicateShape*

4.4 - Módulos de implementação

- ***Actions Module:***

- Definição das principais funções que executam as actions responsáveis pela geração e manipulação das formas virtuais e de construção, com base nas especificações da *SELEX*.
- Exemplo:

#C1: Initial settings

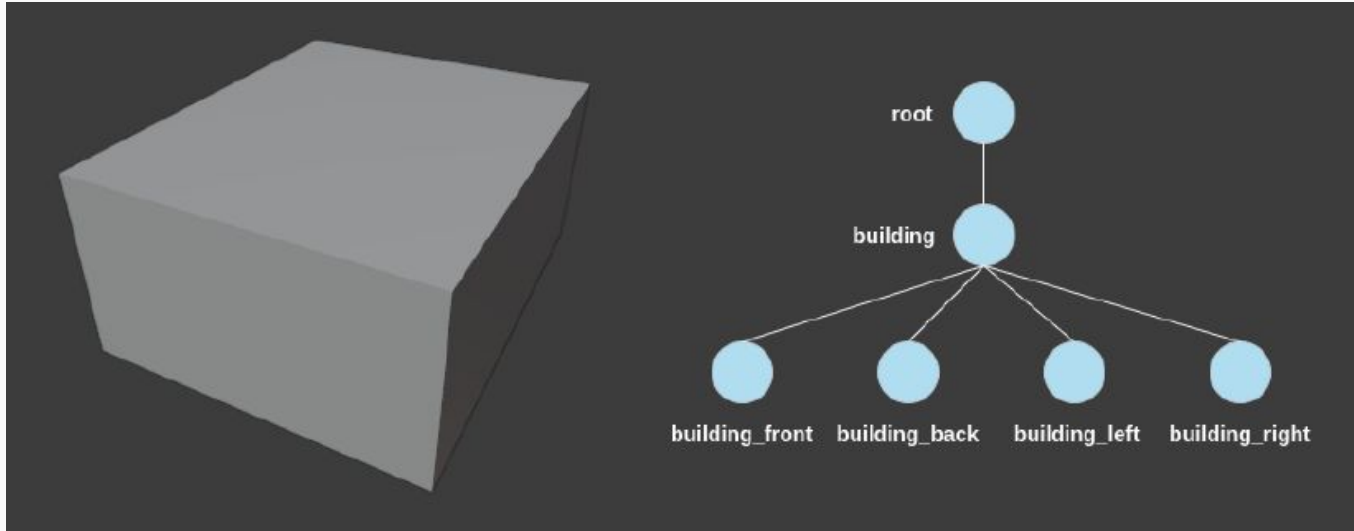
label = "building"; width = 9; depth = 11; height = 5;

4.4 - Módulos de implementação

#C2: Generating mass model

{<> -> **createShape**(label, width, depth, height)};

4.4 - Módulos de implementação



55

Figura 19: Modelo de massa inicial.
Fonte: Próprio autor

4.4 - Módulos de implementação

#C3: Adding virtual shape

```
{<descendant()[label=="building"]/ [label=="building_front"]>  
-> createGrid("main_front_grid", 3, 7)};
```

4.4 - Módulos de implementação

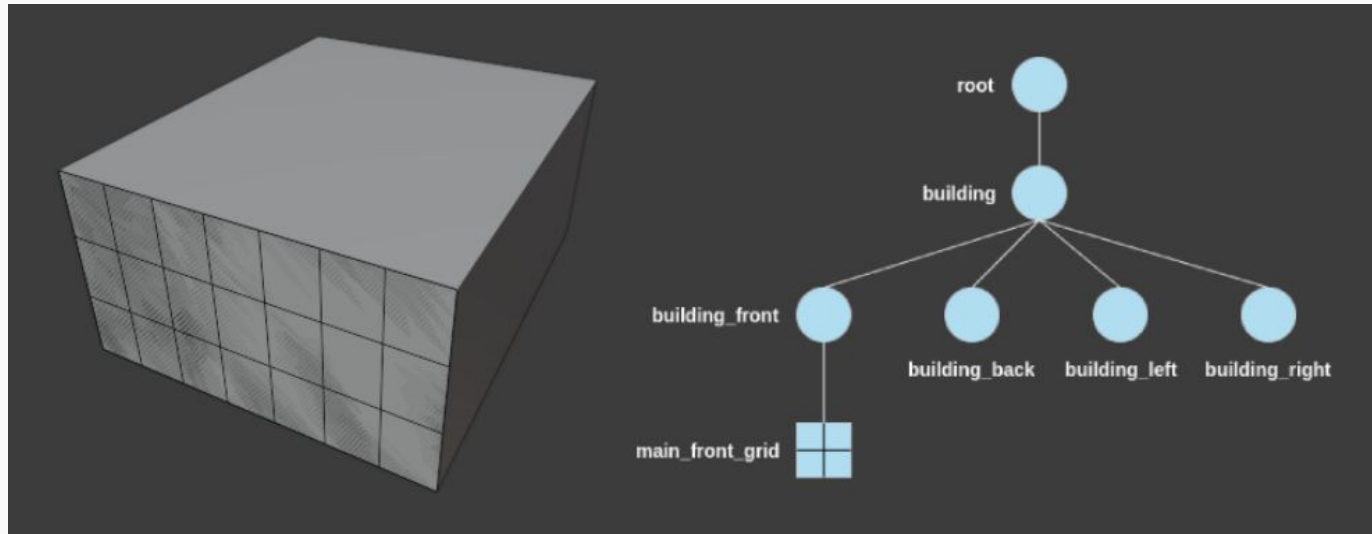


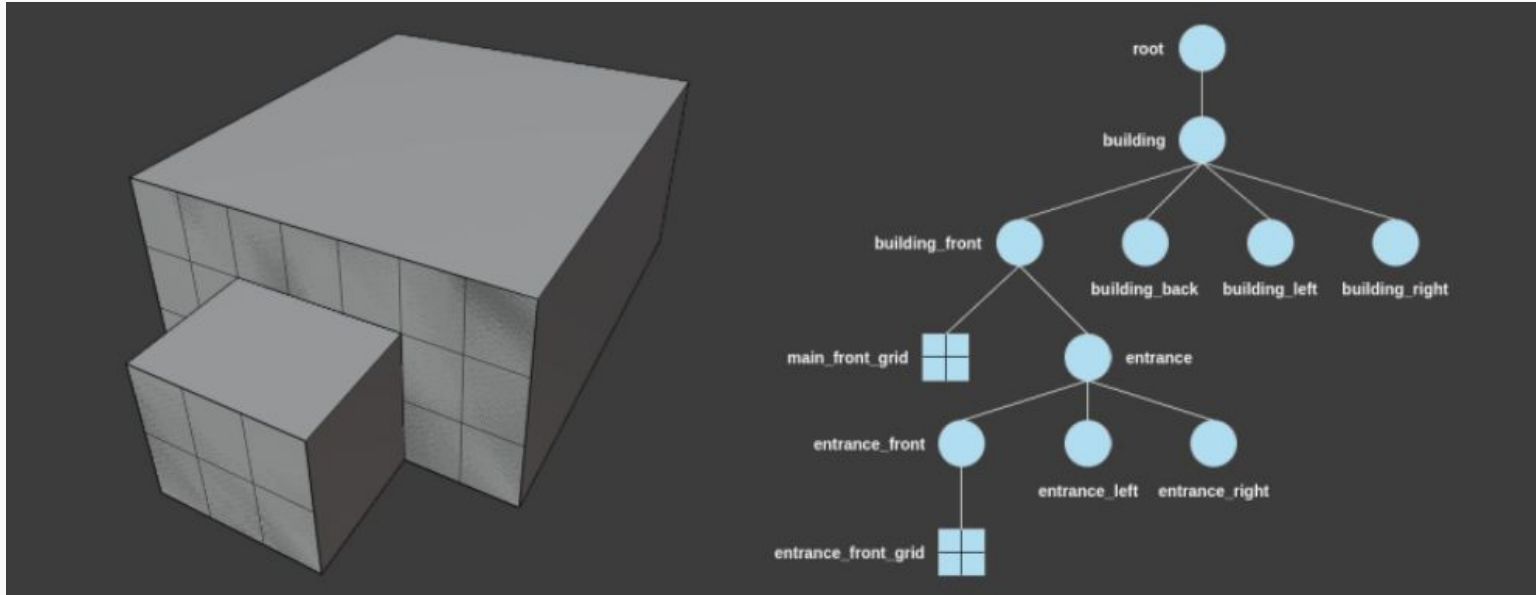
Figura 20: Inclusão de forma virtual na face frontal.
Fonte: Próprio autor

4.4 - Módulos de implementação

#C4: Selecting regions and performing extrusion

```
{<descendant()[label=="building"]/ [label=="building_front"]  
[label=="main_front_grid"]/ [type=="cell"]  
[rowIdx in (2, 3)] [colIdx in (3, 4, 5)][::groupRegions()]>  
-> addVolume("entrance", "building_front", 3,  
"entrance_front", "entrance_left", "entrance_right")});
```

4.4 - Módulos de implementação



57

Figura 21: Resultado obtido após a operação de extrusão.
Fonte: Próprio autor

4.4 - Módulos de implementação

roundShape(*type, direction, roundingDegree, segments, sideReference, axis, insideDegree*):



#C5: Applying roundShape deformation

```
{<descendant()[label=="building"]/[label=="building_front"]/  
[label=="entrance"]/[label=="entrance_front"]>  
-> roundShape("front", "outside", 0.42, 30, "main_front", "vertical");
```

4.4 - Módulos de implementação

58

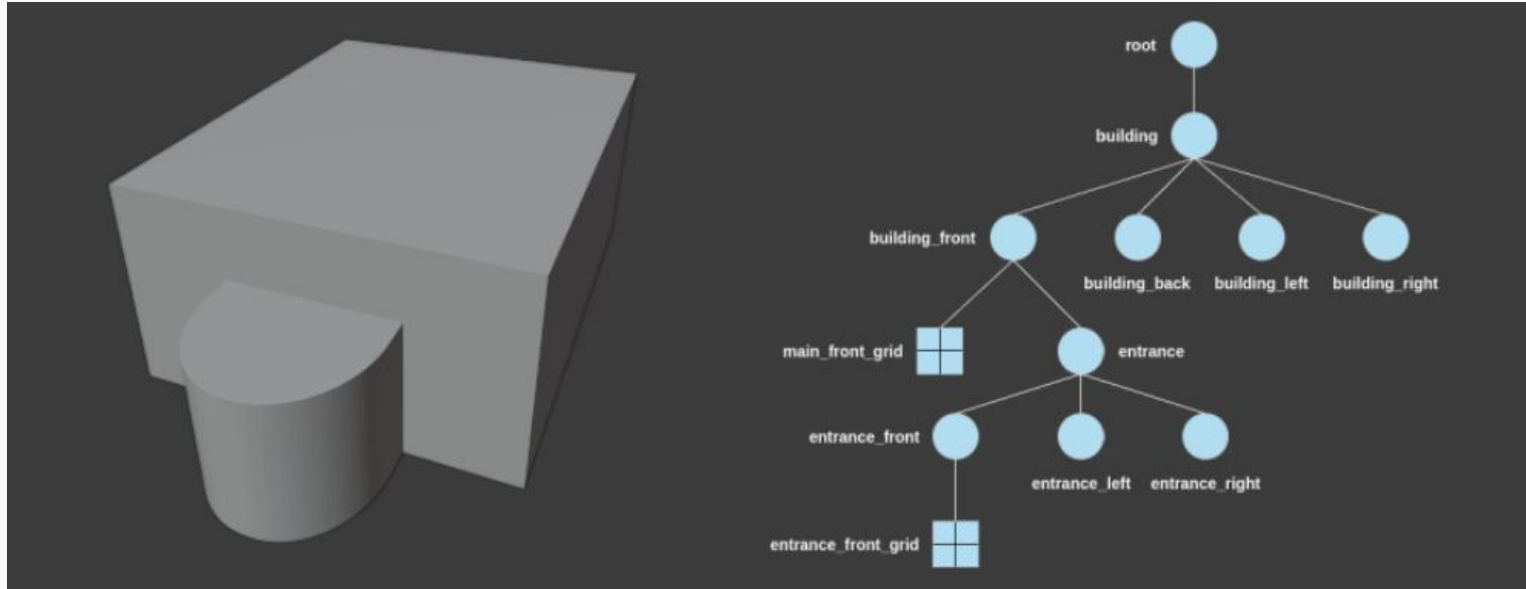


Figura 22: Modelo final com estrutura frontal arredondada.
Fonte: Próprio autor

4.4 - Módulos de implementação

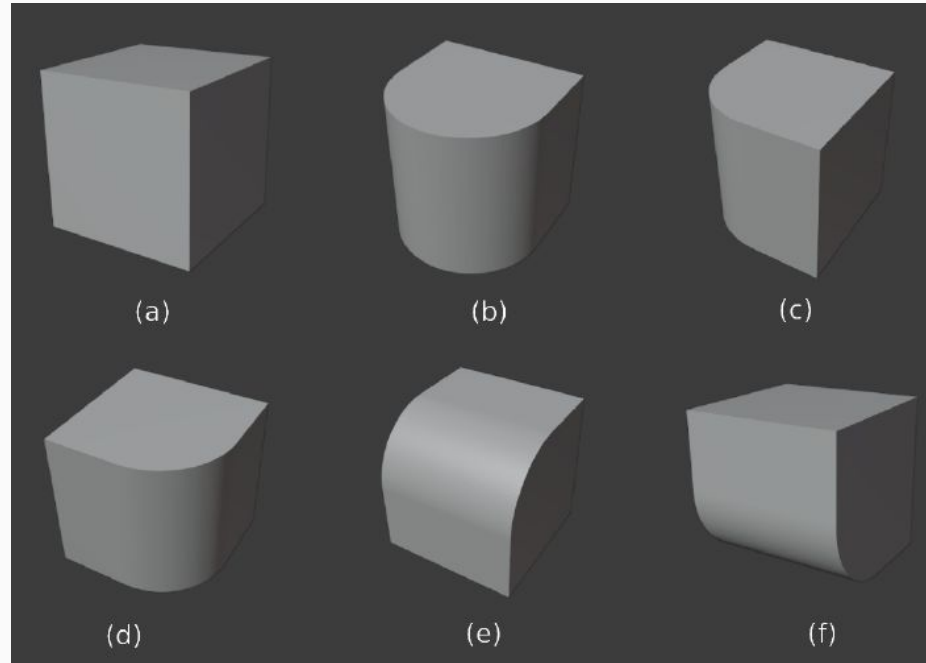
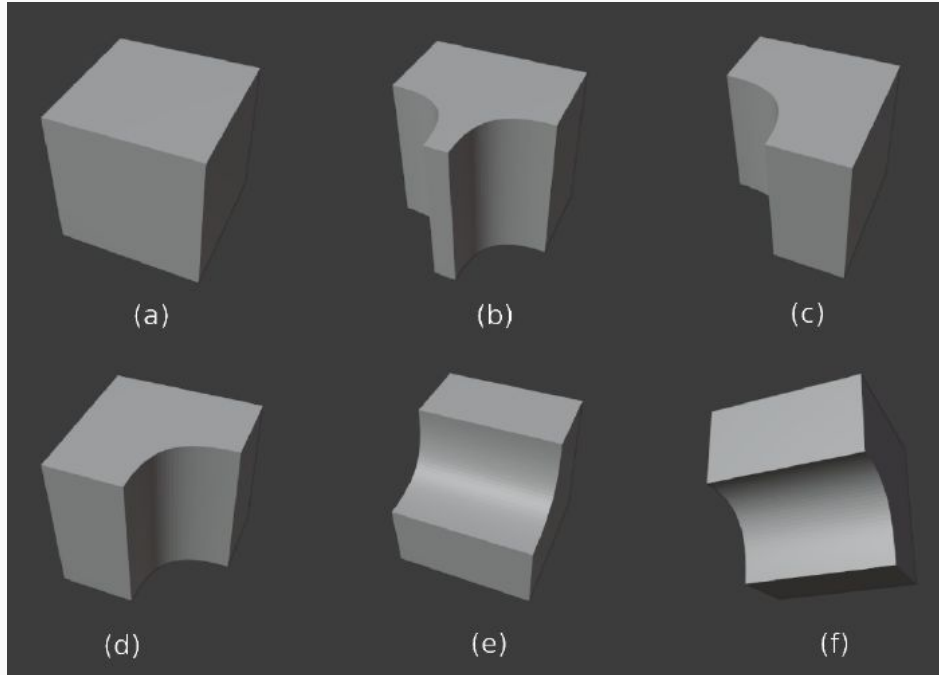


Figura 23: Exemplos de variação do parâmetro *type*, com *direction* valorado como *outside*: (a) Forma original, (b) *front*, (c) *left*, (d) *right*, (e) *top*, (f) *bottom*.

Fonte: Próprio autor

4.4 - Módulos de implementação

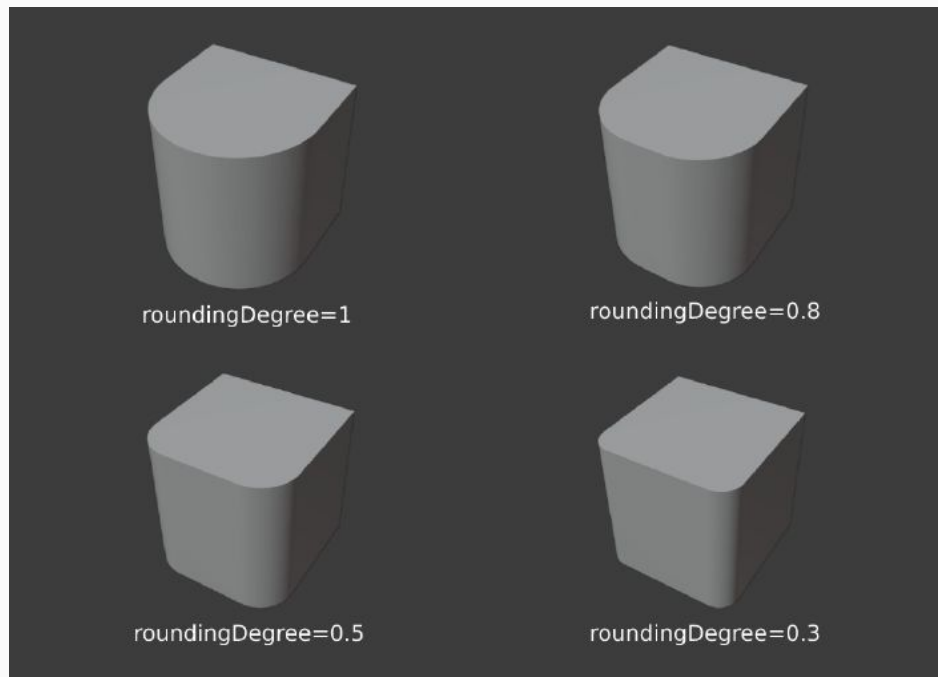


60

Figura 24: Exemplos de variação do parâmetro *type*, com *direction* valorado como *inside*: (a) Forma original, (b) *front*, (c) *left*, (d) *right*, (e) *top*, (f) *bottom*.

Fonte: Próprio autor

4.4 - Módulos de implementação

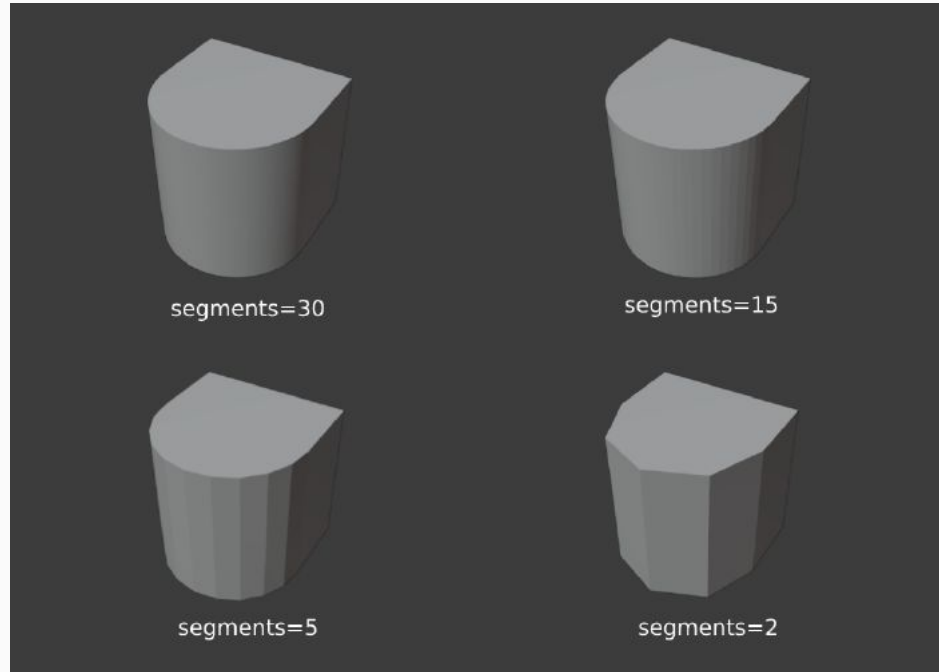


61

Figura 25: Deformação frontal de um modelo de massa em forma de cubo, possuindo uma grade virtual com apenas uma linha e uma coluna, a partir de diferentes valores de *roundingDegree*.

Fonte: Próprio autor

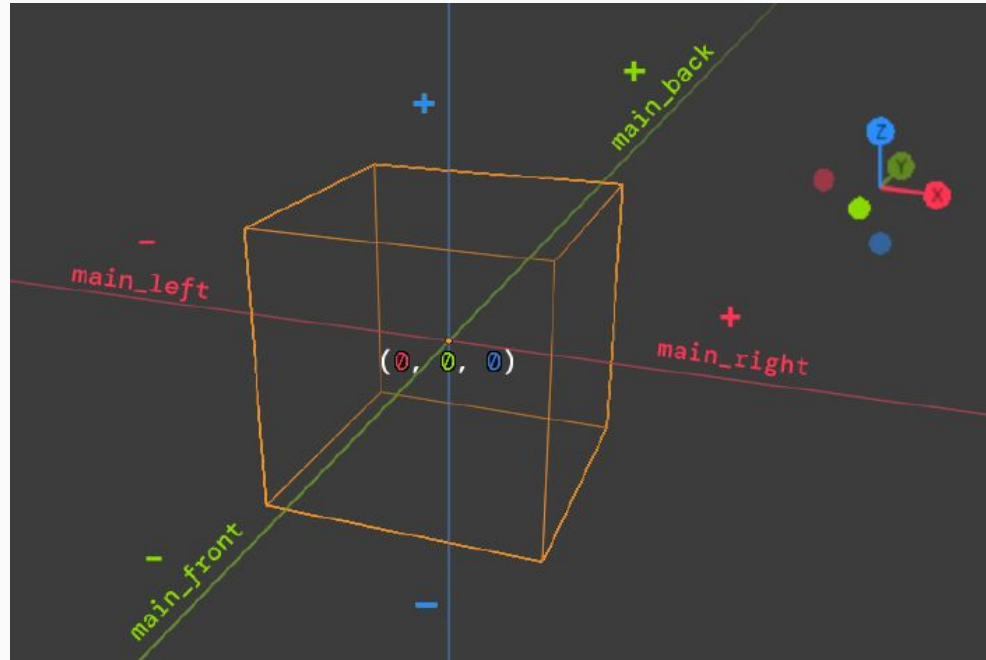
4.4 - Módulos de implementação



62

Figura 26: Deformação frontal de um modelo de massa em forma de cubo, a partir de diferentes valores de *segments*.
Fonte: Próprio autor

4.4 - Módulos de implementação



63

Figura 27: Representação gráfica dos diferentes valores de *sideReference*: *main_front*, *main_back*, *main_left* e *main_right*.
Fonte: Próprio autor

4.4 - Módulos de implementação

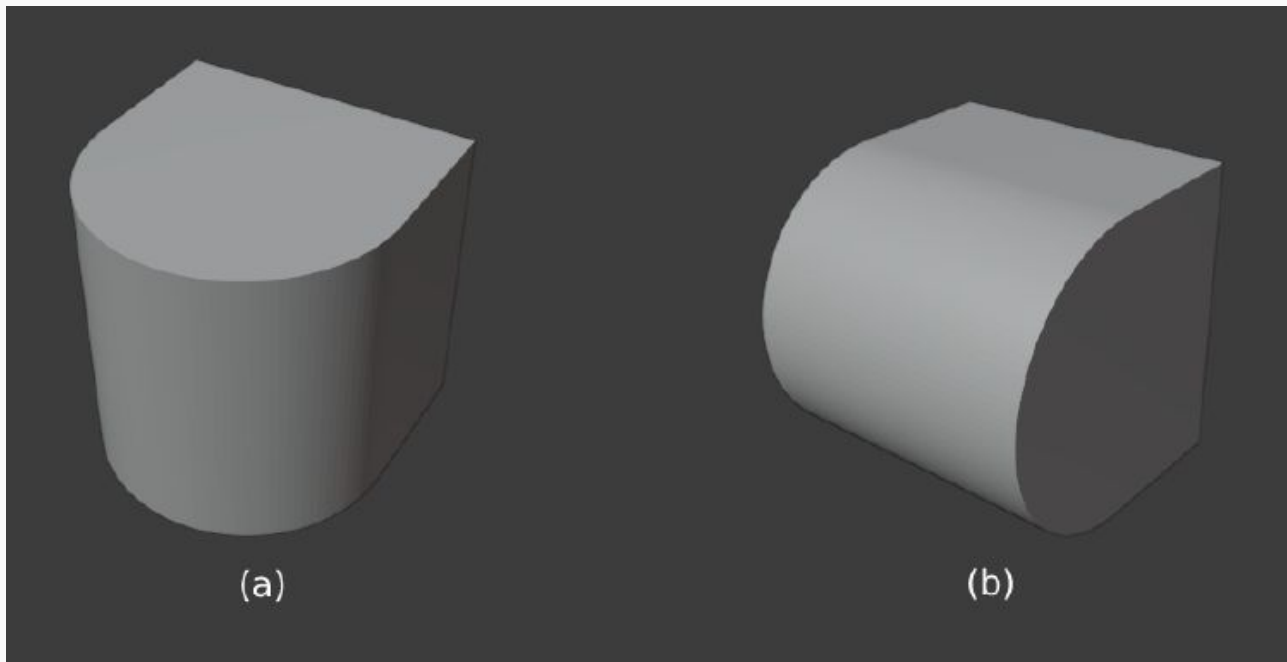
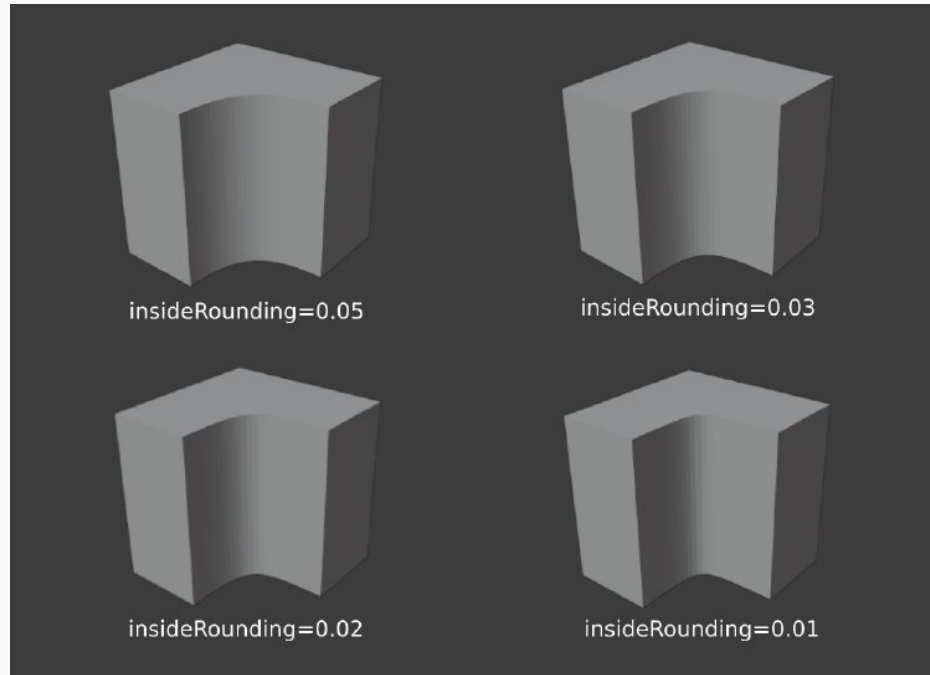


Figura 28: Deformação frontal de um modelo de massa em forma de cubo, para diferentes valores de *axis*:
(a) *vertical* e (b) *horizontal*.

Fonte: Próprio autor

4.4 - Módulos de implementação



65

Figura 29: Deformação lateral (direita) de um modelo de massa em forma de cubo, a partir de diferentes valores de *insideRounding*.

Fonte: Próprio autor

4.4 - Módulos de implementação

- **Outros módulos:**
 - *User Configuration Module*
 - *Input Stream Module*
 - *File Module*
 - *Execution Module*

5 - Resultados

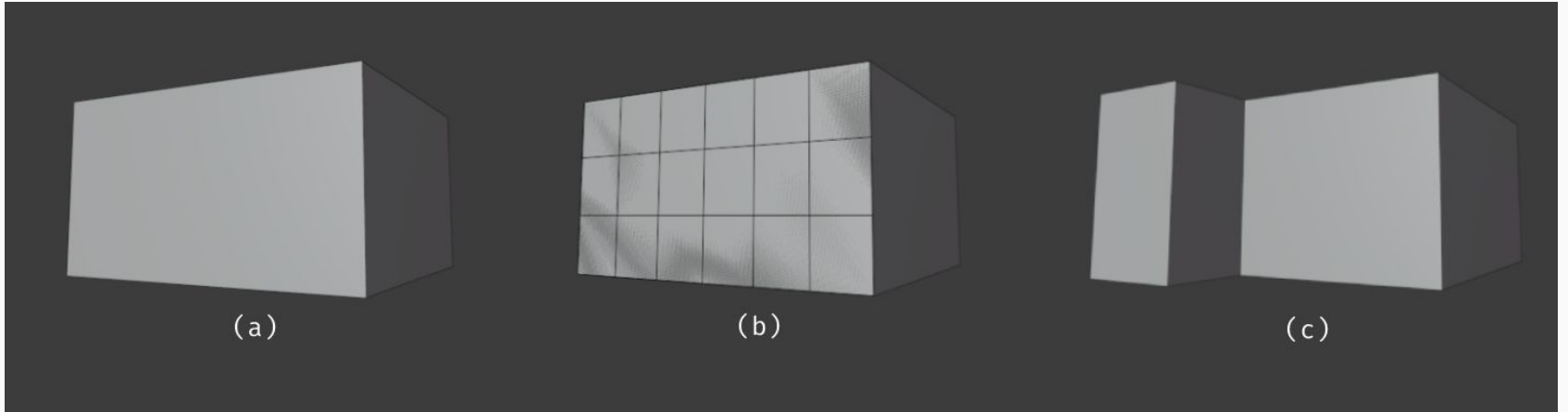
5 - Resultados

5.1 - Modelos gerados

5.2 - Desempenho

5.3 - Restrições

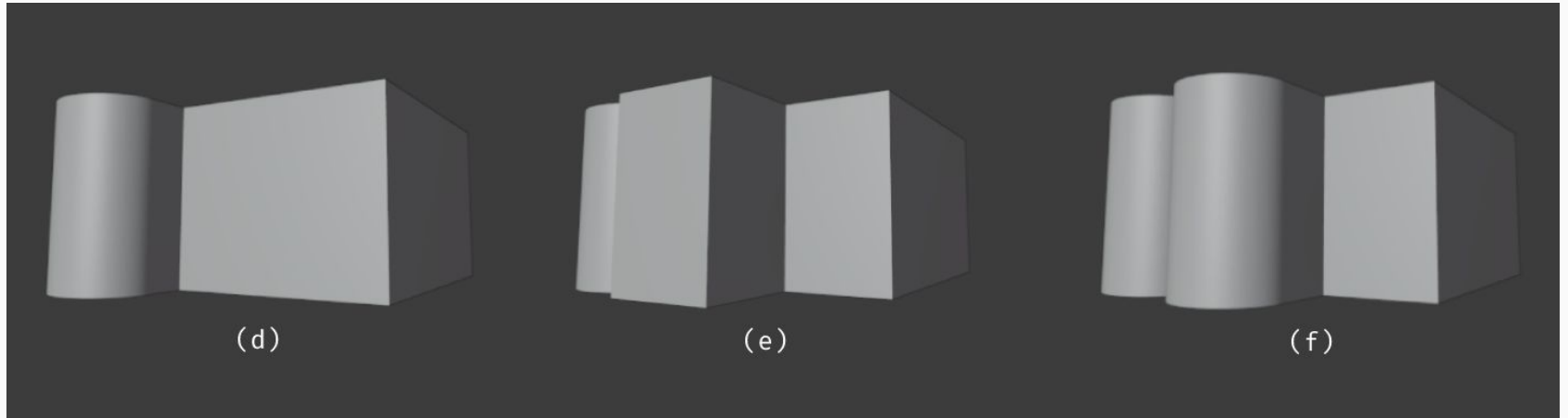
5.1 - Modelos gerados



66

Figura 30.1: Geração do modelo de massa do edifício da Figura 15.
Fonte: Próprio autor

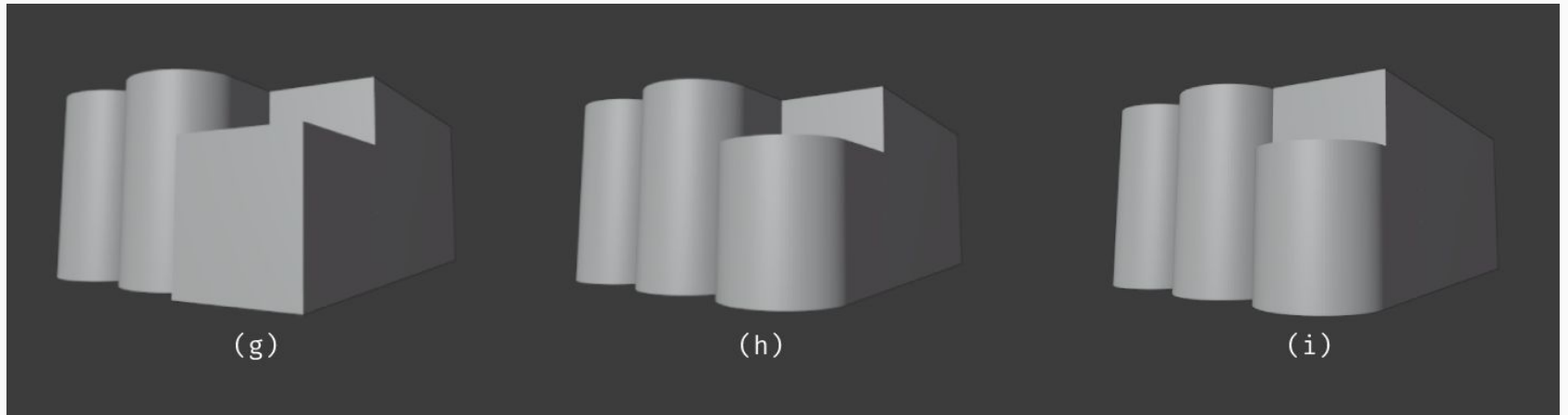
5.1 - Modelos gerados



66

Figura 30.2: Geração do modelo de massa do edifício da Figura 15.
Fonte: Próprio autor

5.1 - Modelos gerados



66

Figura 30.3: Geração do modelo de massa do edifício da Figura 15.
Fonte: Próprio autor

5.1 - Modelos gerados

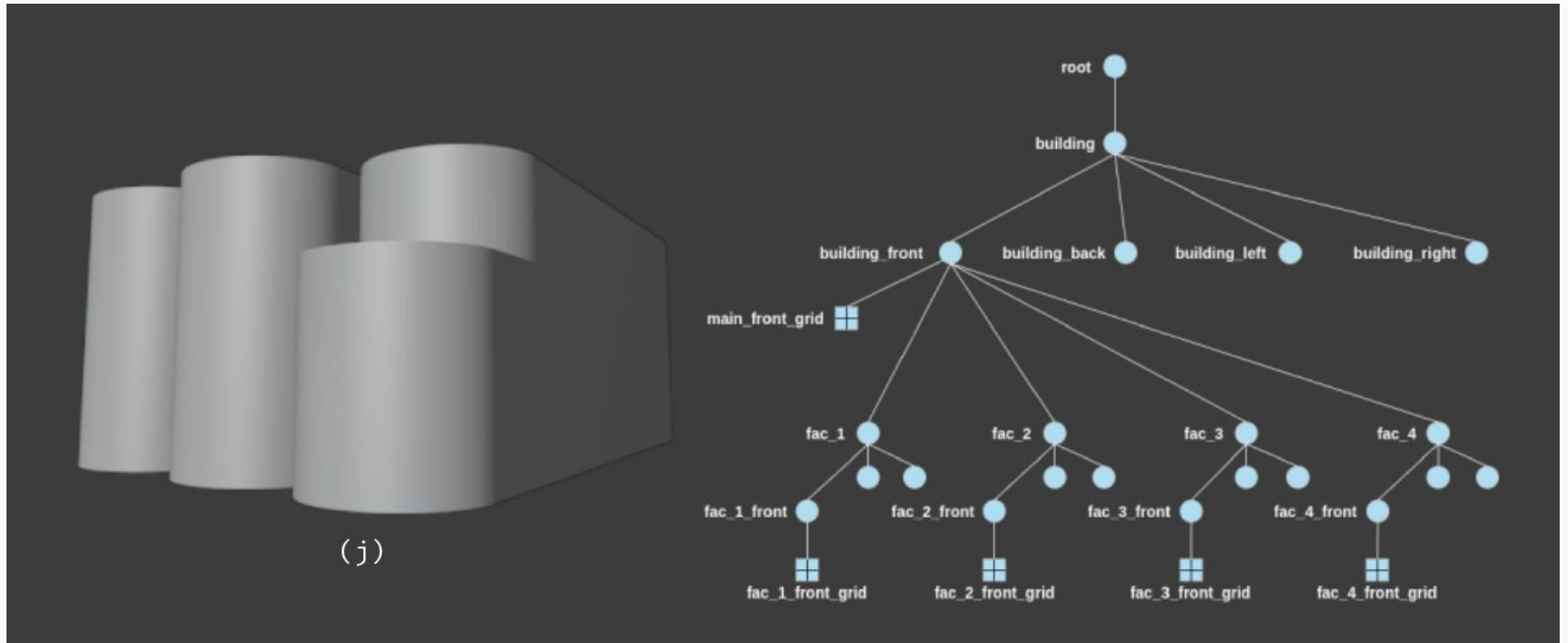
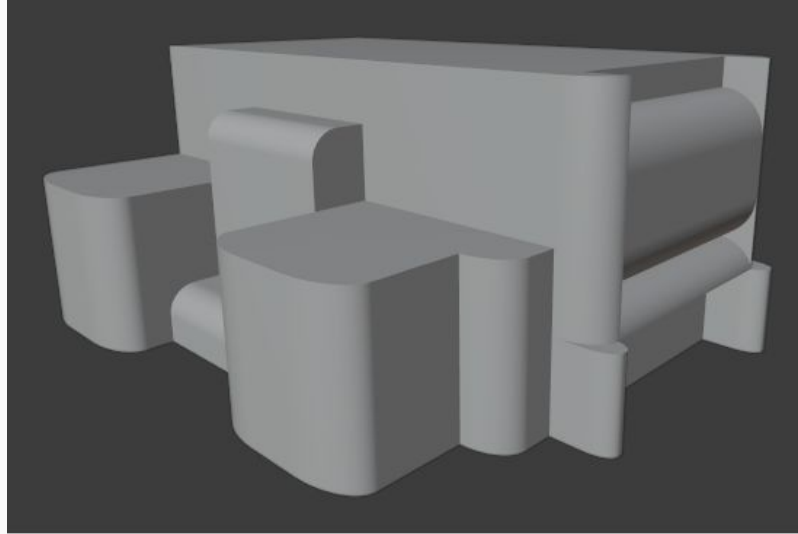


Figura 30.4: Geração do modelo de massa do edifício da Figura 15.

Fonte: Próprio autor

5.1 - Modelos gerados



(a)



(b)

Figura 31: Variação (a) do modelo (b) apresentado por (Jiang et al., 2018).
Fonte: Próprio autor e (Jiang et al., 2018)

5.1 - Modelos gerados

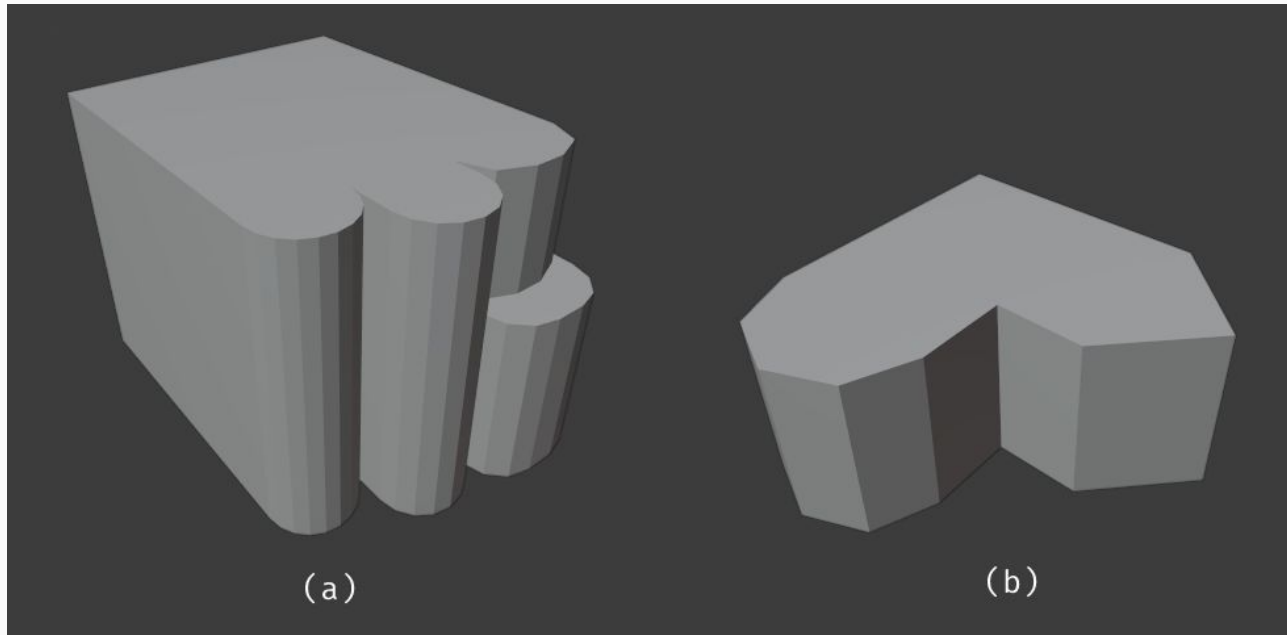


Figura 32: Variação do modelos apresentados por (a) (*Jiang et al., 2018*) e (b) (*VIMONT et al., 2017*).
Fonte: Próprio autor

5.1 - Modelos gerados

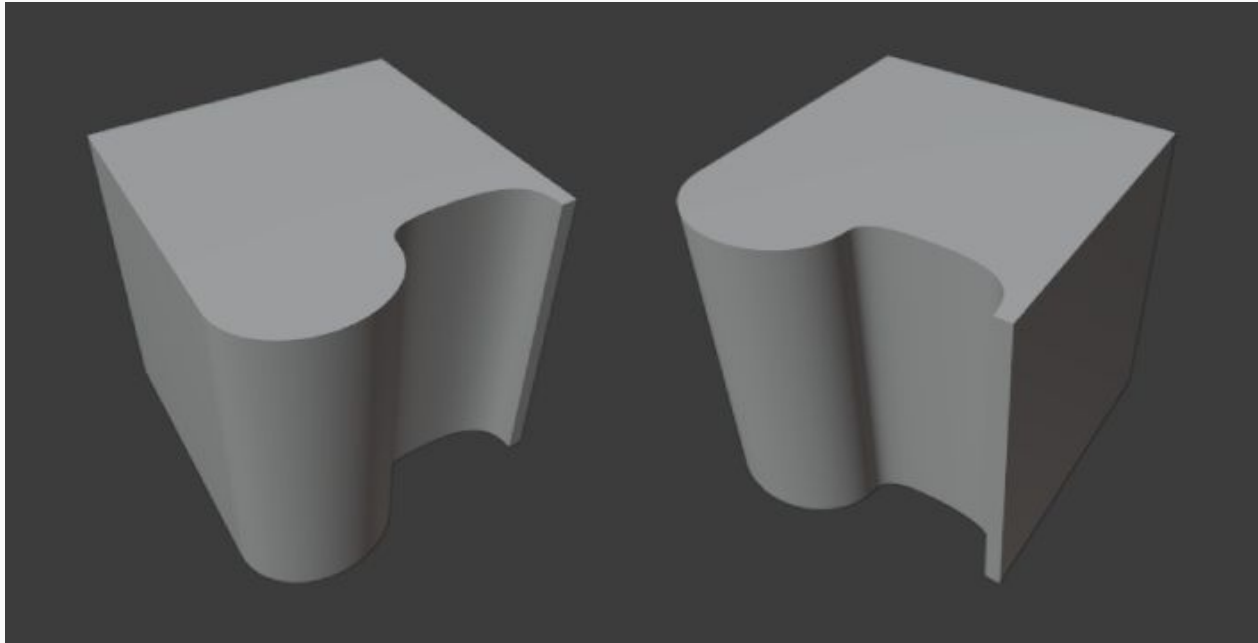


Figura 33: Combinação de arredondamento interno e externo.
Fonte: Próprio autor

5.1 - Modelos gerados

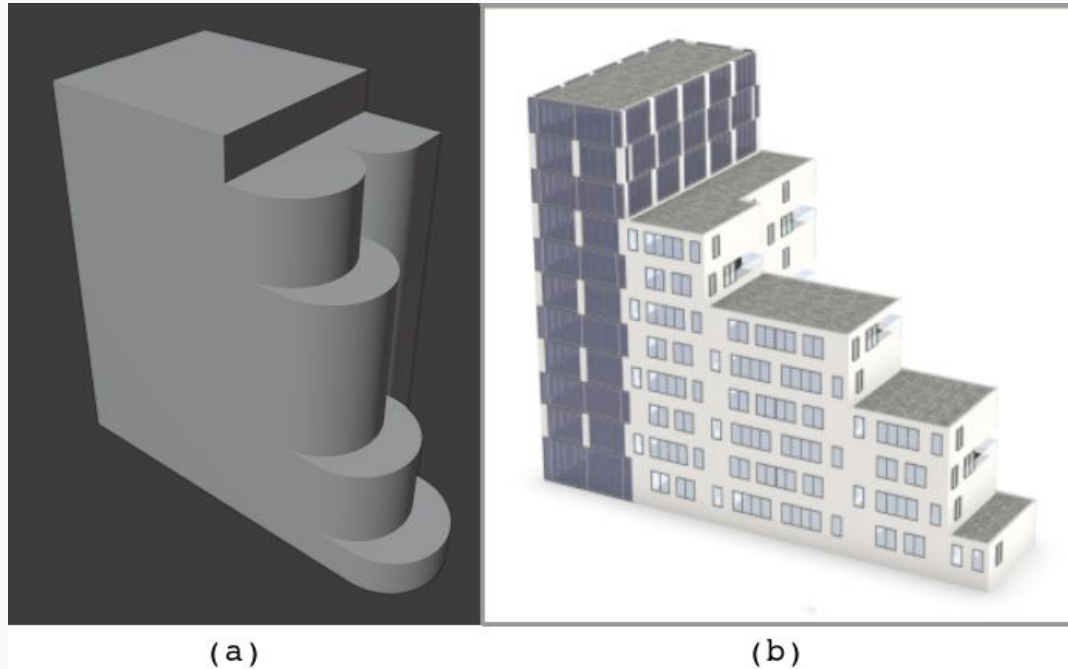
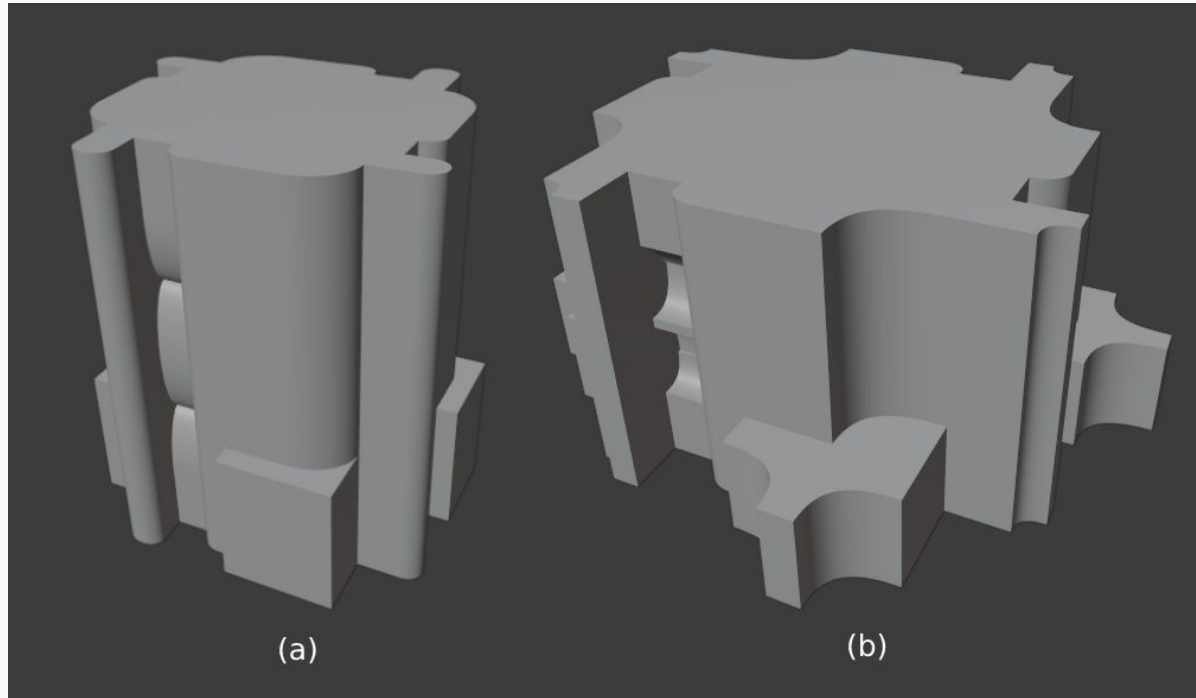


Figura 34: Variação (a) do modelo (b) apresentado por (MÜLLER et al., 2006).
Fonte: Próprio autor e (MÜLLER et al., 2006)

73

5.1 - Modelos gerados



75

Figura 35: Modelos com múltiplas variações de arredondamento (a) externo e (b) interno.
Fonte: Próprio autor

5.2 - Desempenho

Figura	Regras	Vértices	Arestas	Faces	Tamanho (MB)
66(f)	11	520	780	270	1,0
67	18	904	1356	468	1,1
68	8	212	319	114	1,0
69(a)	36	1576	2364	822	1,3
69(b)	31	1380	2070	720	1,3
69(c)	26	1056	1584	552	1,3
69(d)	24	892	1338	474	1,3
70(a)	11	96	144	58	1,1
70(b)	8	36	54	24	1,0
71	30	824	1236	438	1,2
72	9	272	408	144	1,1
73(a)	13	588	882	306	1,1
74(a)	12	520	780	270	1,1
75(a)	58	2392	3588	1254	1,5
75(b)	62	2152	3228	1134	1,5

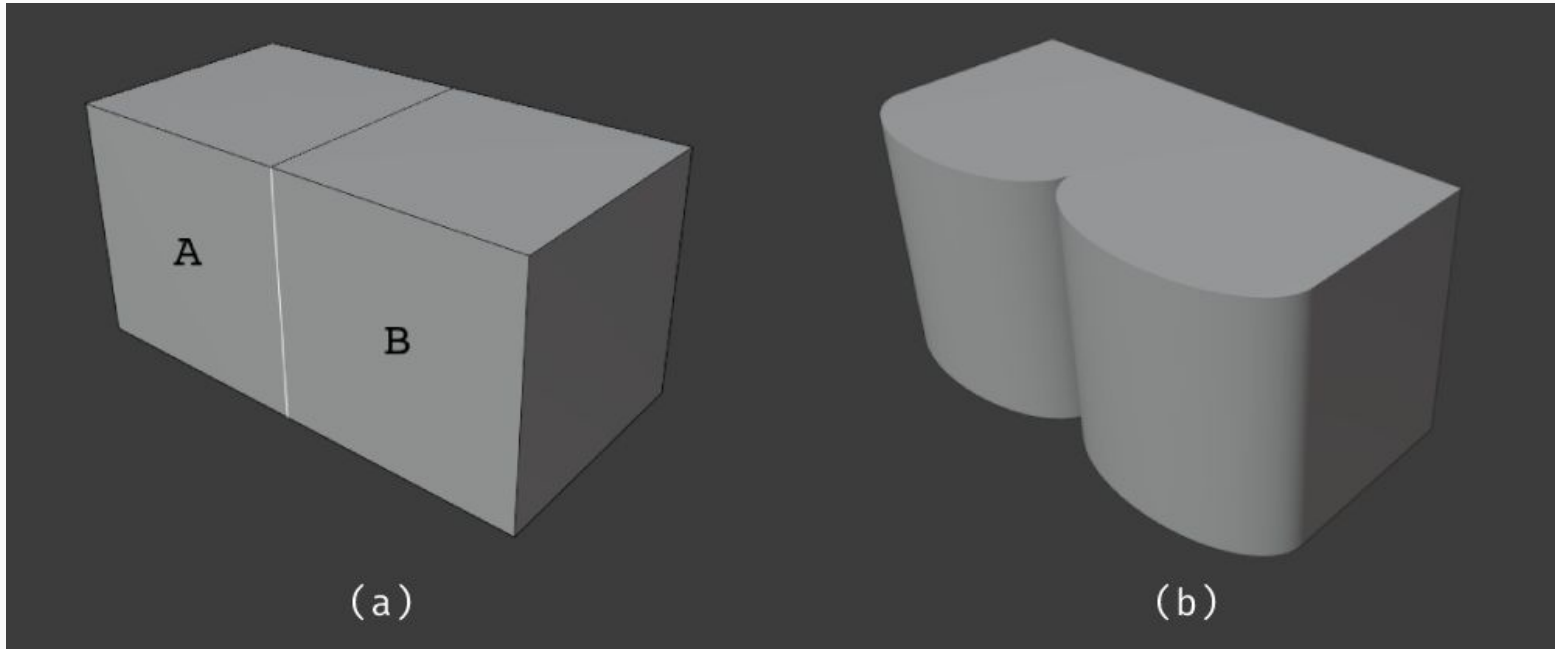
Tabela 1 - Dados estatísticos dos modelos gerados pela solução proposta.

Fonte: Próprio autor

5.3 - Restrições

- A *action* roundShape deve ser definida imediatamente após uma operação de extrusão.
- Não é realizada uma representação da região arredondada na árvore de formas.
- Não é possível realizar uma operação de arredondamento em faces paralelas que possuem aresta em comum. Contudo, isto pode ser contornado:

5.3 - Restrições



76

Figura 36: (a) Exemplo de faces que compartilham uma mesma aresta, e (b) solução para aplicar deformação.
Fonte: Próprio autor

6 - Conclusão

6 - Conclusão

6.1 - Considerações

6.2 - Trabalhos futuros

6.1 - Considerações

- Resultados remetem à geometria base de edifícios do mundo real.
- Geração de modelos no formato *low poly*.
- Possibilidade de edição por meio do Blender.
- Projeto *open source*.

6.2 - Trabalhos futuros

- Manipulação da forma virtual em relação à região arredondada, visando uma representação na árvore de formas.
- Disposição de elementos como janelas e portas, através das formas virtuais, bem como a inclusão de telhados.
- Geração estocástica de múltiplos modelos, a fim de produzir um ambiente urbano.
- Definição de atributos de *design*, como cor e textura.
- Criação de *add-on* para simplificar a leitura do arquivo que contém as regras.

7- Referências

7 - Referências

SMELIK, R. M.; TUTENEL, T.; BIDARRA, R.; BENES, B. *A survey on procedural modeling for virtual worlds*. Comput. Graph. Forum, The Eurographs Association & John Wiley & Sons, Ltd., Chichester, GBR, v. 33, n. 6, p. 31–50, set. 2014. ISSN 0167-7055.

NISHIDA, G.; BOUSSEAU, A.; ALIAGA, D. G. *Procedural Modeling of a Building from a Single Image*. Computer Graphics Forum, Wiley, v. 37, n. 2, 2018.

SIMON, L. *Procedural reconstruction of buildings : towards large scale automatic 3d modeling of urban environments*. 07 2011.

KNIGHT, T. W. *Transformations in Design: A Formal Approach to Stylistic Change and Innovation in the Visual Arts*. USA: Cambridge University Press, 1995. ISBN 0521384605.

7 - Referências

RODRIGUES, F. C. M. *Split grammar evolution for procedural modeling of virtual buildings*. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Ceará, Fortaleza, 2014.

MÜLLER, P.; WONKA, P.; HAEGLER, S.; ULMER, A.; GOOL, L. V. *Procedural modeling of buildings*. ACM Trans. Graph., Association for Computing Machinery, New York, NY, USA, v. 25, n. 3, p. 614–623, jul. 2006. ISSN 0730-0301.

SCHWARZ, M.; MÜLLER, P. *Advanced procedural modeling of architecture*. ACM Trans. Graph., Association for Computing Machinery, New York, NY, USA, v. 34, n. 4, jul. 2015. ISSN 0730-0301.

Jiang, H.; Yan, D.; Zhang, X.; Wonka, P. *Selection expressions for procedural modeling*. IEEE Transactions on Visualization and Computer Graphics, v. 26, n. 4, p. 1775–1788, 2018.

7 - Referências

SEDERBERG, T. W.; PARRY, S. R. *Free-form deformation of solid geometric models*. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques. New York, NY, USA: Association for Computing Machinery, 1986. (SIGGRAPH '86), p. 151–160. ISBN 0897911962.

VIMONT, U.; ROHMER, D.; BEGAULT, A.; CANI, M.-P. *Deformation grammars: Hierarchical constraint preservation under deformation*. Computer Graphics Forum, v. 36, n. 8, p. 429–443, 2017.

HAUBENWALLNER, K. *Procedural generation using grammar based modeling and genetic algorithms*. Graz University of Technology - Institute of Computer Graphics, Austria, 2016.

7 - Referências

KRECKLAU, L.; PAVIC, D.; KOBELT, L. *Generalized use of non-terminal symbols for procedural modeling*. Computer Graphics Forum, v. 29, n. 8, p. 2291–2303, 2010.

ZMUGG, R.; THALLER, W.; KRISPEL, U.; EDELSBRUNNER, J.; HAVEMANN, S.; FELLNER, D. *Procedural architecture using deformation-aware split grammars*. Vis. Comput., v. 30, p. 1009–1019, 09 2014.

EDELSBRUNNER, J.; HAVEMANN, S.; SOURIN, A.; FELLNER, D. W. *Procedural modeling of architecture with round geometry*. Computers & Graphics, v. 64, p. 14 – 25, 2017. ISSN 0097-8493. Cyberworlds 2016.

TRUESDALL, C.; NOLL, W. *The non-linear field theories of mechanics*. In: The Non-Linear Field Theories of Mechanics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. p. 1–579. ISBN 978-3-662-13183-1.

7 - Referências

EBERT, D. S.; MUSGRAVE, F. K.; PEACHEY, D.; PERLIN, K.; WORLEY, S. *Texturing and Modeling: A Procedural Approach*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. ISBN 1558608486.

LINDENMAYER, A. *Mathematical models for cellular interactions in development*. Journal of Theoretical Biology, v. 18, n. 3, p. 300 – 315, 1968. ISSN 0022-5193.



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS CRATEÚS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Geração procedural de modelos arquiteturais com geometria arredondada utilizando *Selection Expressions (SELEX)*

Daniel Henrique de Brito

Orientador: Arnaldo Barreto Vila Nova

Coorientador: Ítalo Mendes da Silva Ribeiro

Crateús
2021