



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
2020

SUMÁRIO

1. Introdução
2. Fundamentação teórica
3. Trabalhos correlatos
4. Proposta
5. Cronograma
6. 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

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

1.4 - Objetivos específicos

- Implementar e avaliar a linguagem para geração de modelos arquiteturais;
- 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 - *L-Systems*

2.3 - *Shape grammar*

2.4 - *Split grammar*

2.5 - *CGA Shape*

2.6 - *CGA++*

2.7 - *Selection Expressions (SELEX)*

2.8 - 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

2.1 - Geração procedural

- Exemplo:



Figura 1: Modelo de ambiente virtual criado com o *framework* SketchaWorld.

Fonte: (SMELIK et al., 2014)

2.1 - Geração procedural

- Exemplo:

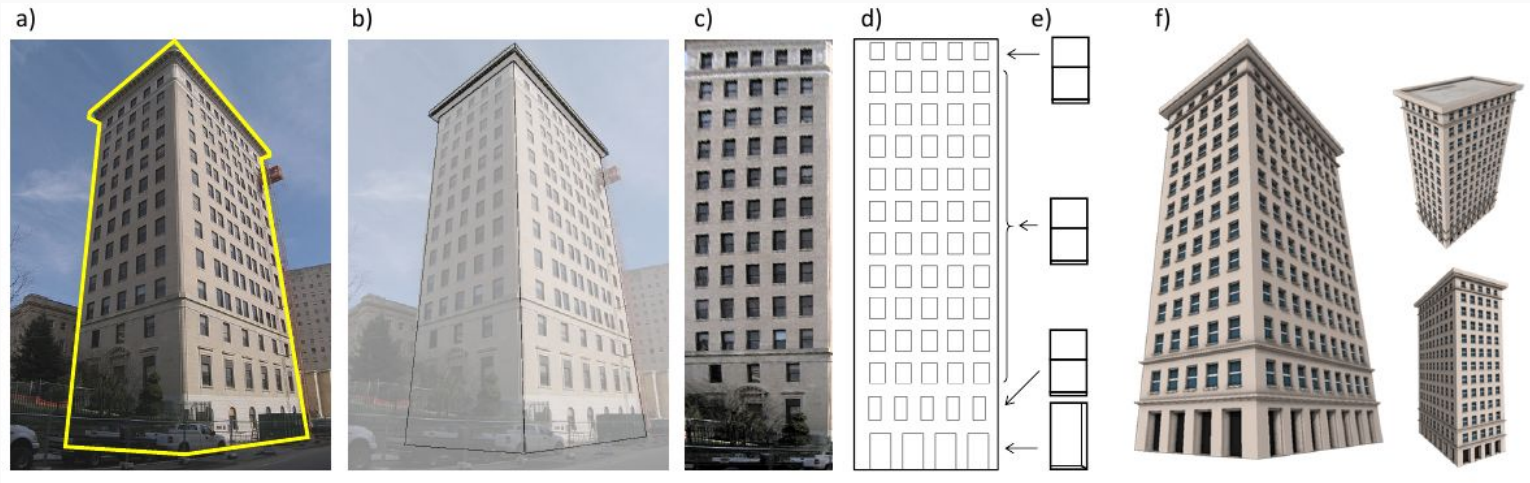
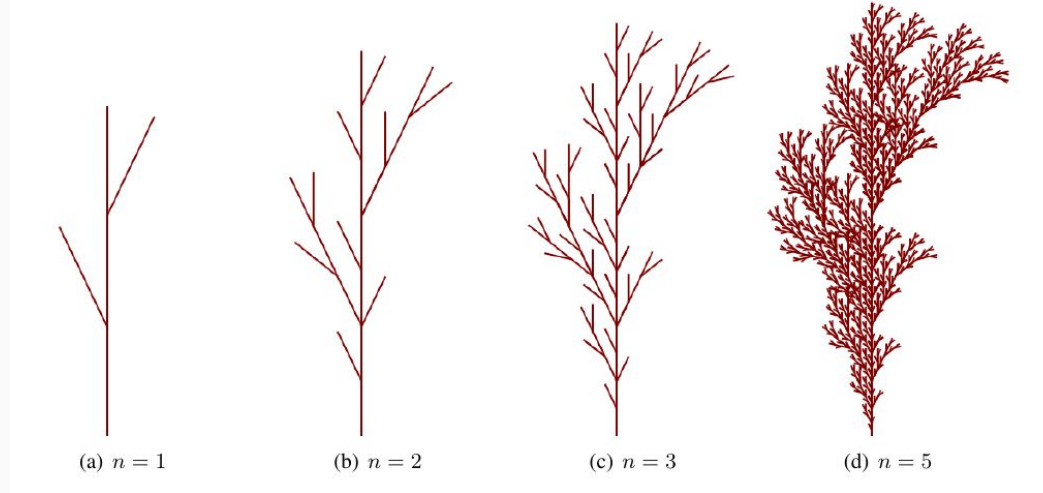


Figura 2: Modelagem procedural a partir de uma fotografia.

Fonte: (NISHIDA et al., 2018)

2.2 - L-Systems

Os *L-Systems* de *Lindenmayer* (1968) tem como ideia principal alterar partes de um objeto inicial por meio da aplicação de sucessivas regras de substituição, com objetivo de gerar objetos mais elaborados.



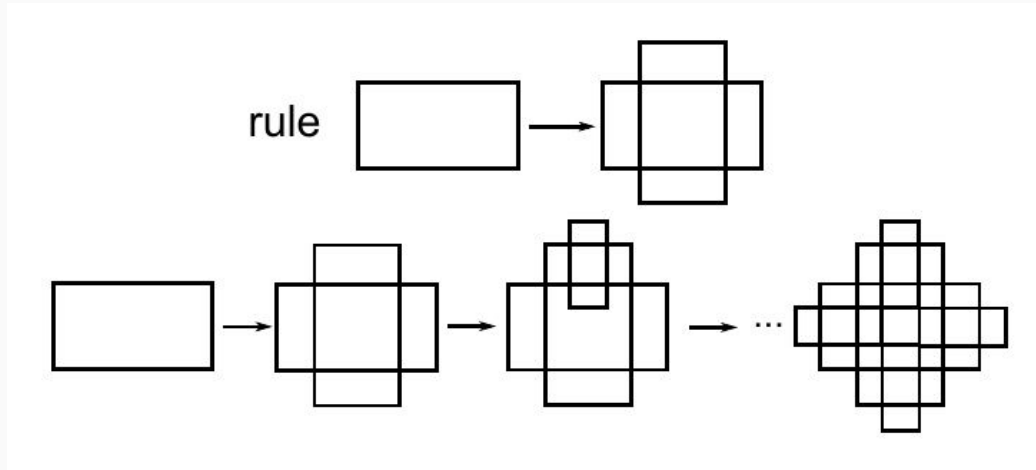
5

Figura 3: Geração de um modelo de alga utilizando *L-Systems*.

Fonte: (SIMON, 2011)

2.3 - Shape grammars

A ideia central das *shape grammars* para criação de modelos 3D é começar com uma forma básica, e modificá-la até que se pareça com o modelo desejado (HAUBENWALLNER, 2016).



6

Figura 4: Gramática para gerar planta de igreja na forma de cruz grega.
Fonte: (KNIGHT, 1995)

2.4 - Split grammar

As *split grammars* foram introduzidas por Wonka et al. (2003), sendo utilizadas para modelagem procedural de construções, geralmente, fachadas de edifícios.

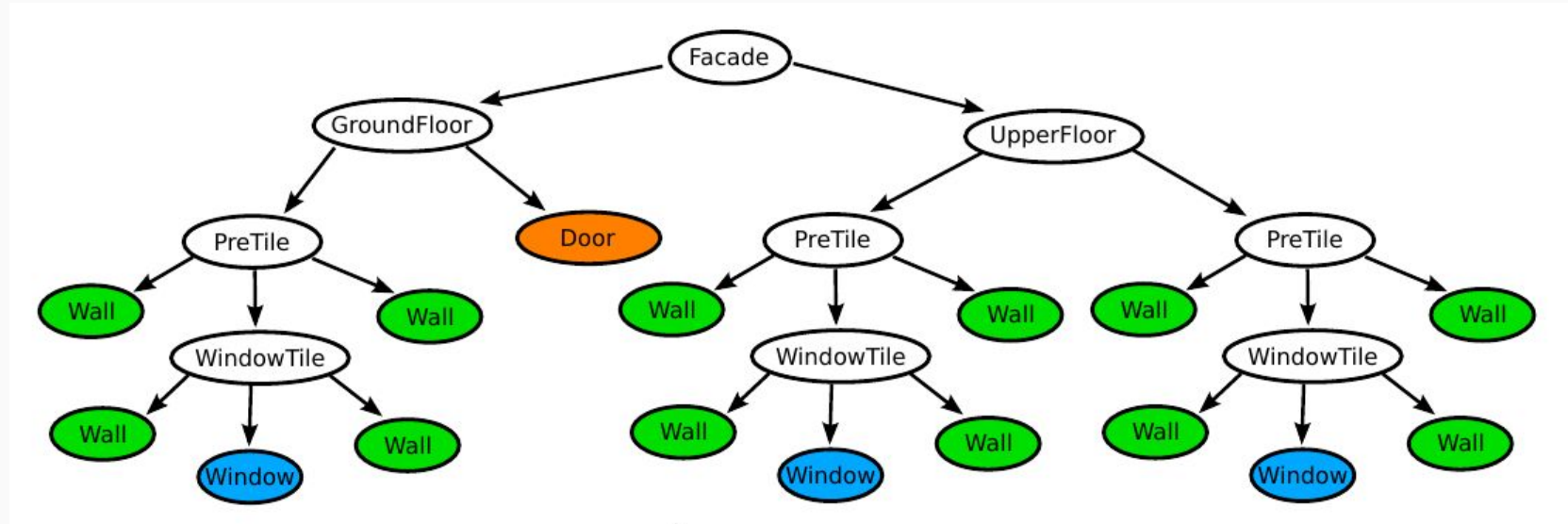


Figura 5: Árvore de derivação de uma fachada.

Fonte: (RODRIGUES, 2014)

2.4 - Split grammar

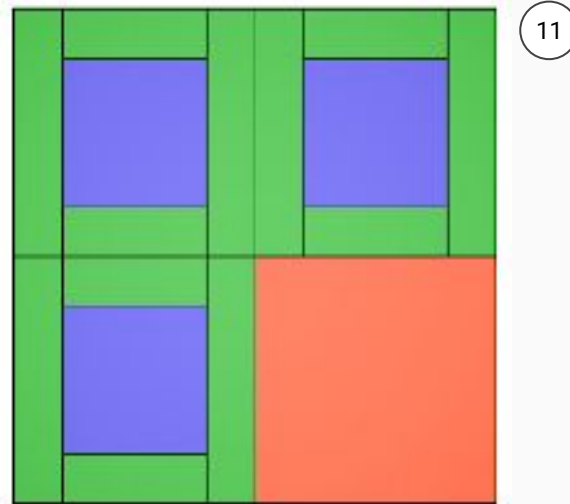


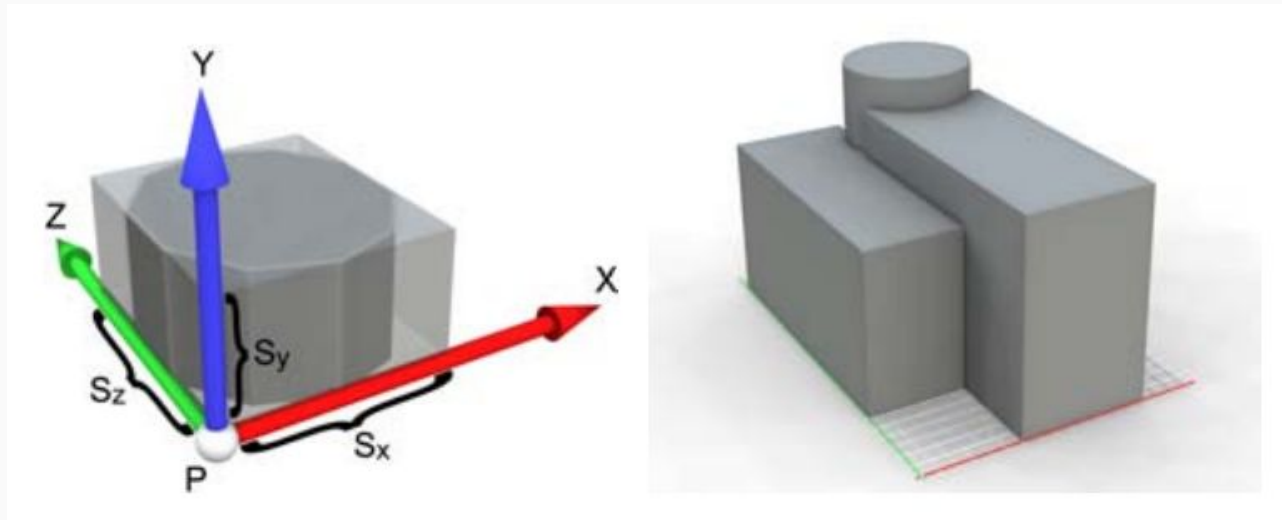
Figura 6: Fachada gerada a partir da árvore de derivação representada na Figura 5.
Fonte: (RODRIGUES, 2014)

2.5 - 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*. Suas especificações podem ser divididas em:

- Forma
- Regras de escopo
- Regras de divisão básica
- Regras de redimensionamento
- Regras de divisão com repetição
- Regras de divisão de componentes
- Processo de produção

2.5 - CGA Shape



12

Figura 7: Representação geométrica da *CGA Shape*.
Fonte: (MÜLLER et al., 2006)

2.5 - CGA Shape

1. *lot* $\rightsquigarrow S(1r, \text{building_height}, 1r)$
Subdiv("Z", *Scope.sz* * *rand*(0.3, 0.5), 1r) { *facades* | *sidewings* }
2. *sidewings* \rightsquigarrow
Subdiv("X", *Scope.sx* * *rand*(0.2, 0.6), 1r) { *sidewing* | ϵ }
Subdiv("X", 1r, *Scope.sx* * *rand*(0.2, 0.6)) { ϵ | *sidewing* }
3. *sidewing*
 $\rightsquigarrow S(1r, 1r, \text{Scope.sz} * \text{rand}(0.4, 1.0)) \text{ facades} : 0.5$
 $\rightsquigarrow S(1r, \text{Scope.sy} * \text{rand}(0.2, 0.9), \text{Scope.sz} * \text{rand}(0.4, 1.0)) \text{ facades} : 0.3$
 $\rightsquigarrow \epsilon : 0.2$
4. *facades* $\rightsquigarrow \text{Comp}(\text{"sidefaces"}) \{ \text{facade} \}$

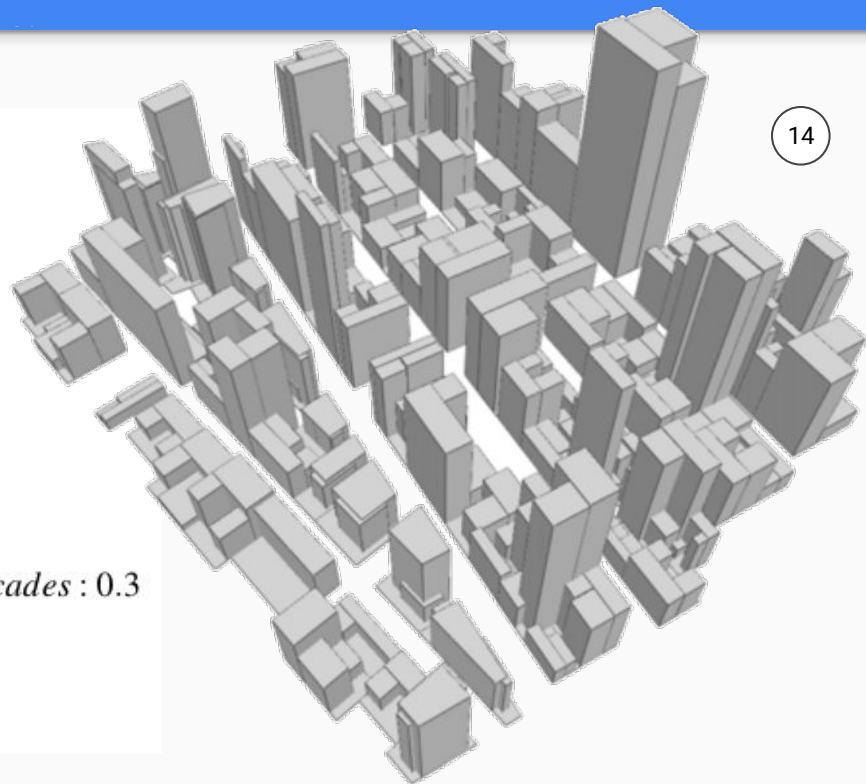


Figura 8: Variações estocásticas de modelos de edifícios.
Fonte: (MÜLLER et al., 2006)

2.6 - CGA++

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

- Impossibilidade de realizar operações envolvendo múltiplas formas;
- Falta de informações contextuais para determinados objetivos de modelagem;
- Incapacidade de gerar uma derivação de dentro de outra derivação e consultar ou incorporar o resultado.

2.6 - CGA++

- Especificações:
 - Consultar árvore de formas
 - Construir novas formas
 - Funções
- Linguagem:
 - Objetos: *booleanos*, números, *strings*, listas e tuplas
 - Argumentos: iteráveis, implícitos e de expressão
 - Variáveis auxiliares

2.6 - CGA++

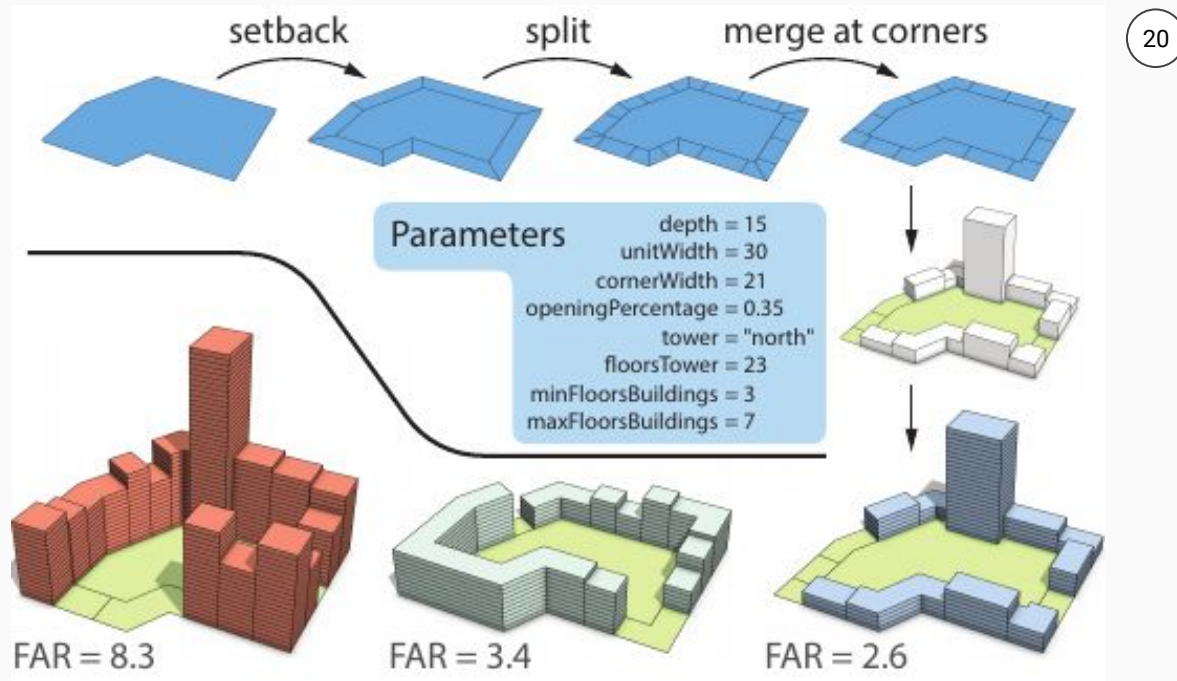


Figura 9: Representação de bloco perimetral.
Fonte: Adaptado de (SCHWARZ; MÜLLER, 2015)

2.7 - 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.

2.7 - Selection Expressions (SELEX)

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

2.7 - 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()*

2.7 - Selection Expressions (SELEX)

- Especificações:

- Exemplo:

$\langle [\text{label} = \text{"A"}] / [\text{label} = \text{"C"}][h \geq 2] \rangle$

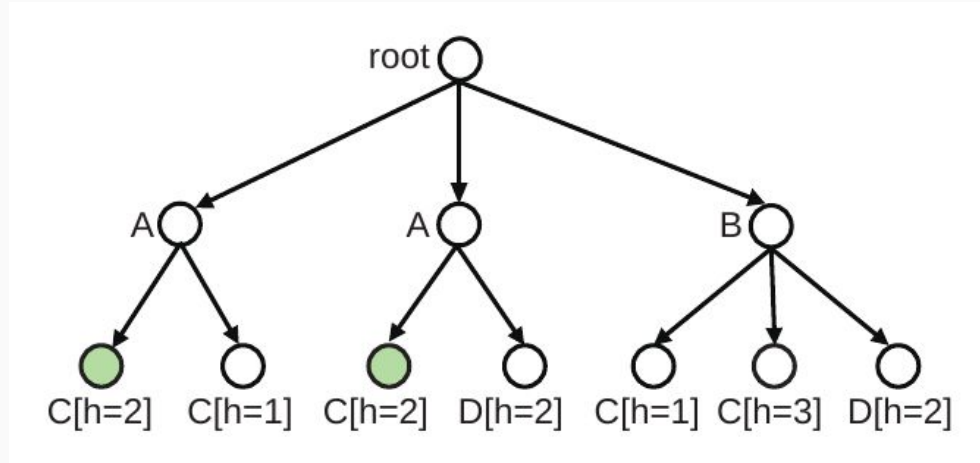


Figura 10: Grafo abstrato da seleção de nós.

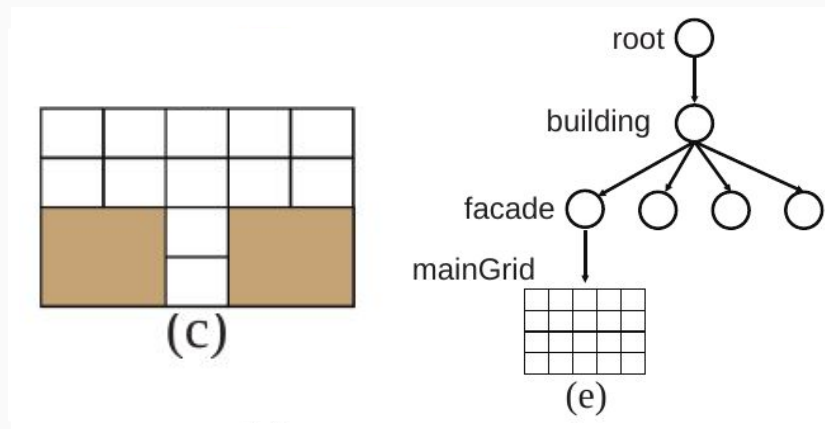
Fonte: (Jiang et al., 2018)

2.7 - Selection Expressions (SELEX)

- Especificações:

- Exemplo:

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



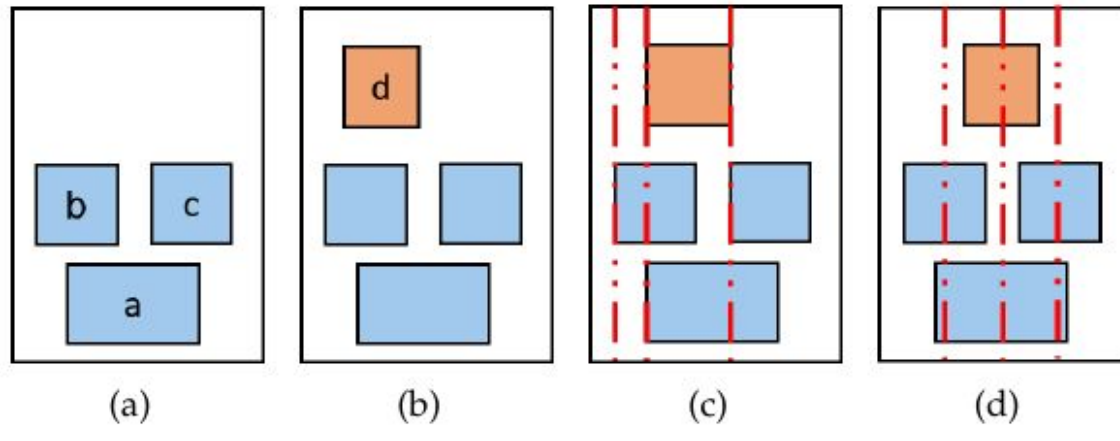
28

Figura 11: Exemplo da utilização de seletores.

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

2.7 - Selection Expressions (SELEX)

- Especificações:
 - Ações: *addShape*, *attachShape*, *coverShape*, *connectShape*
 - Funções de restrição: *left*, *right*, *top*, *bottom*, *center-x*, *center-y*



30

Figura 12: Comparação do ajuste com e sem o uso de rótulos.

Fonte: (Jiang et al., 2018)

2.7 - Selection Expressions (SELEX)

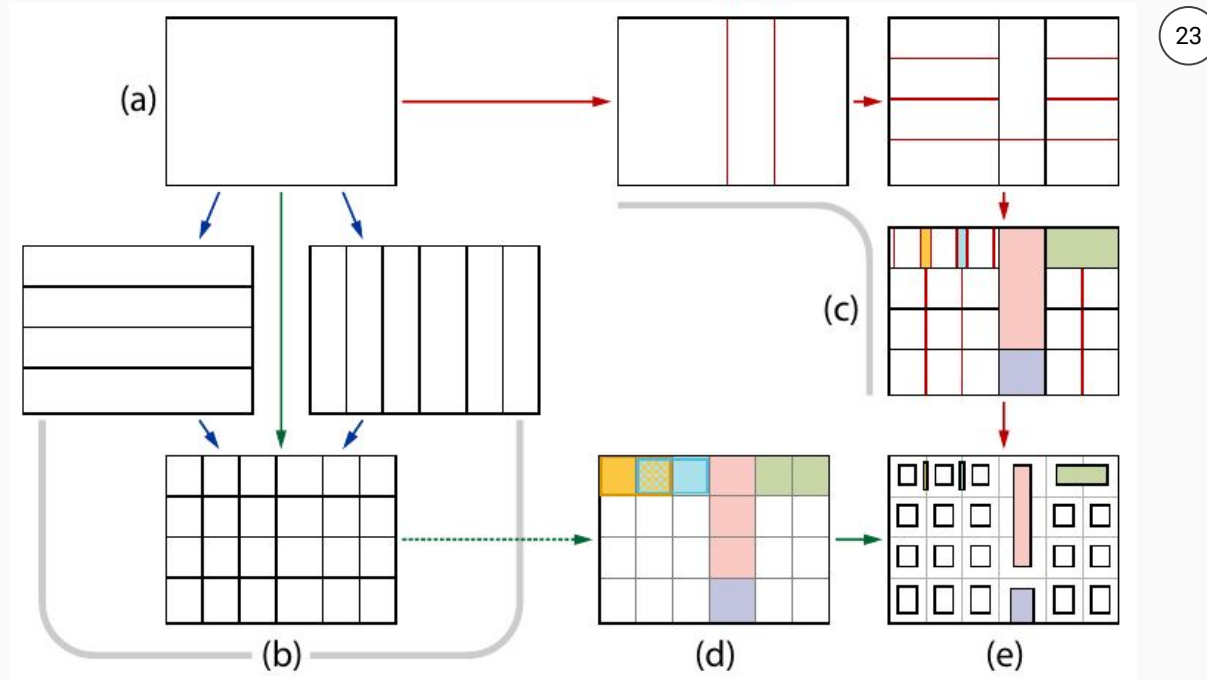


Figura 13: Comparação dos paradigmas de modelagem da *SELEX* e da *CGA Shape*.
Fonte: (Jiang et al., 2018)

2.7 - Selection Expressions (SELEX)

32

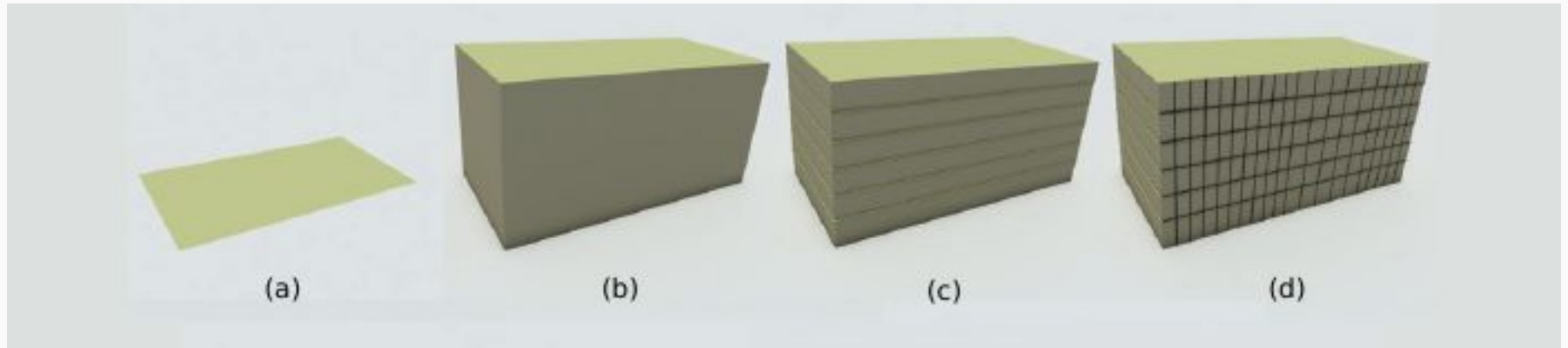


Figura 14.1: Exemplo de modelagem utilizando *SELEX*.

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

2.7 - Selection Expressions (SELEX)

32

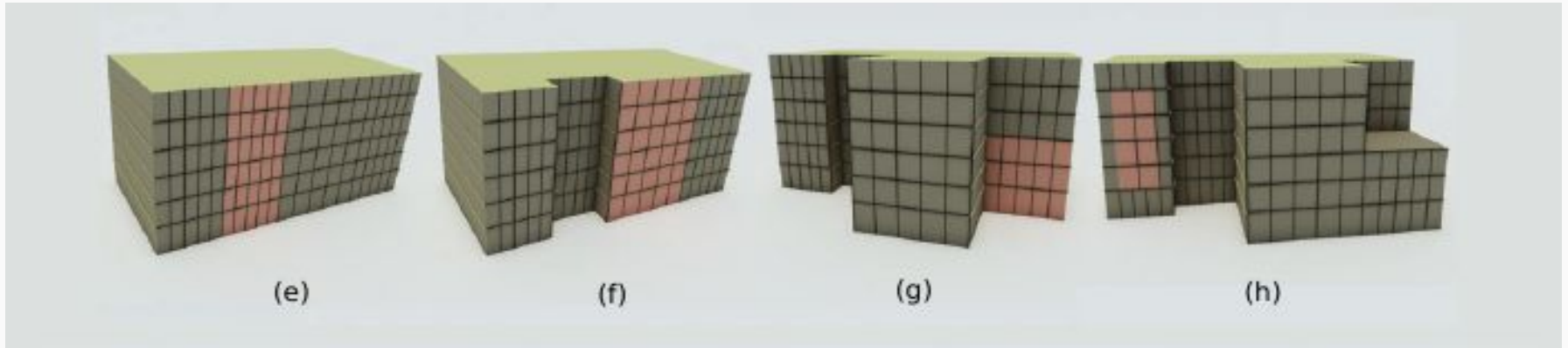


Figura 14.2: Exemplo de modelagem utilizando *SELEX*.

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

2.7 - Selection Expressions (SELEX)

32

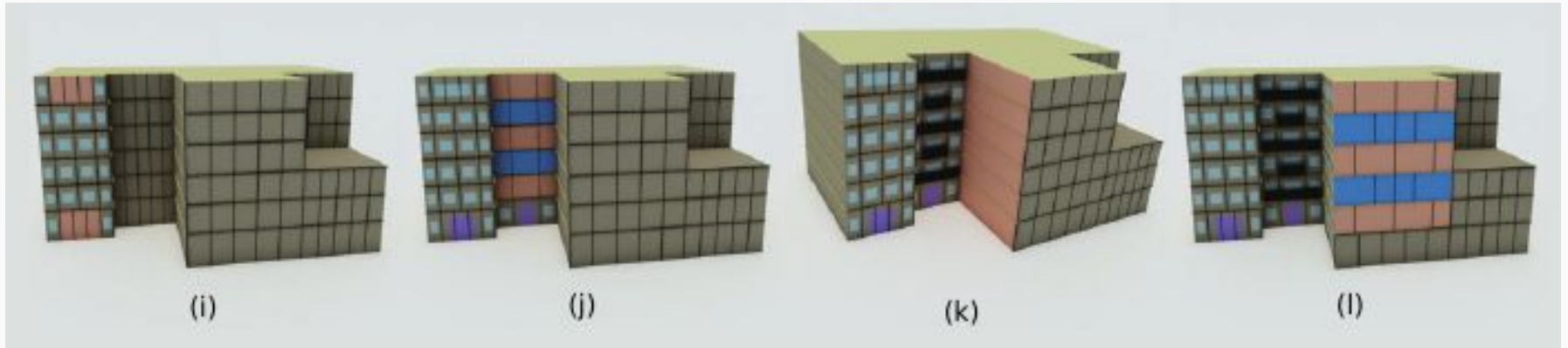


Figura 14.3: Exemplo de modelagem utilizando *SELEX*.

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

2.7 - Selection Expressions (SELEX)

32

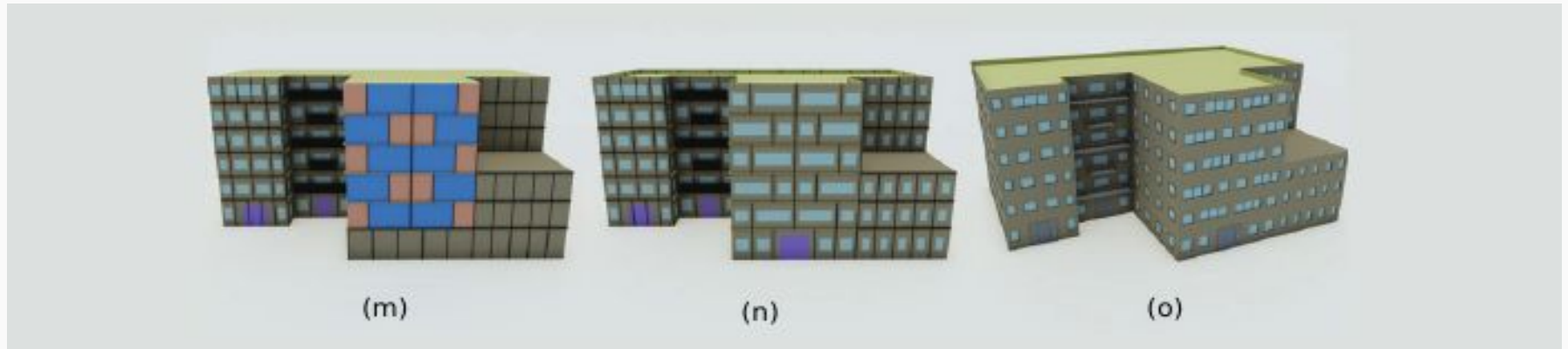


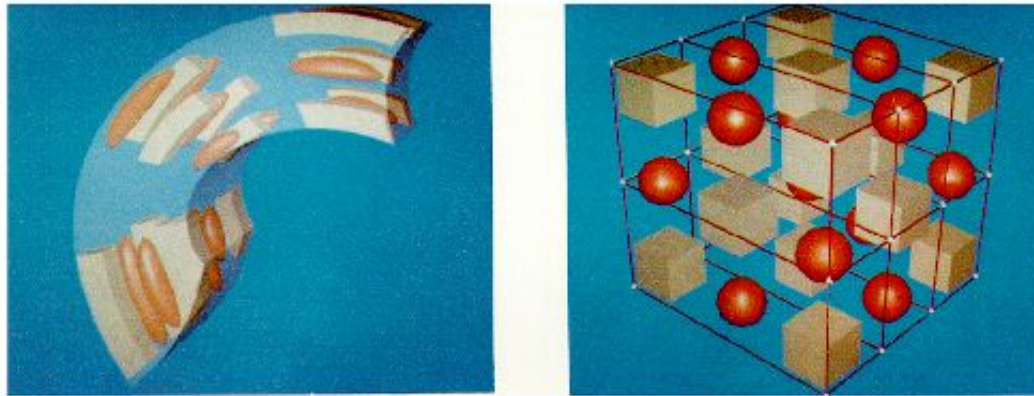
Figura 14.4: Exemplo de modelagem utilizando *SELEX*.
Fonte: Adaptado de (Jiang et al., 2018)

2.8 - 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*).

2.8 - 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.



33

Figura 15: Exemplo de deformação de forma livre.

Fonte: (SEDERBERG; PARRY, 1986)

2.8 - Deformação

Uma evolução do trabalho de *Sederberg e Parry (1986)* é trazida por *Jin e Li (2000)*, onde é apresentado um método de deformação tridimensional utilizando coordenadas polares direcionais.

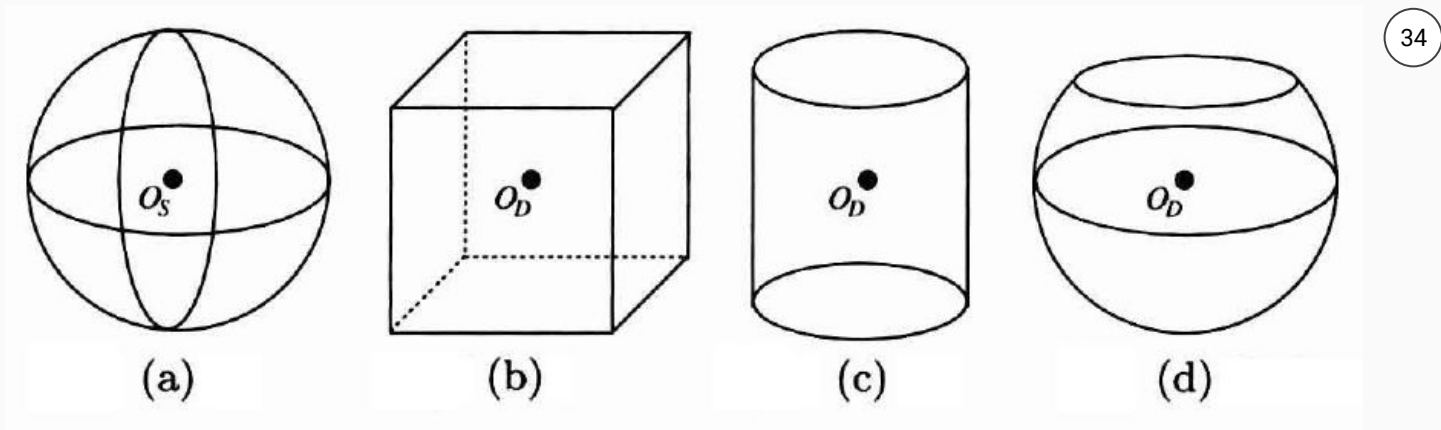
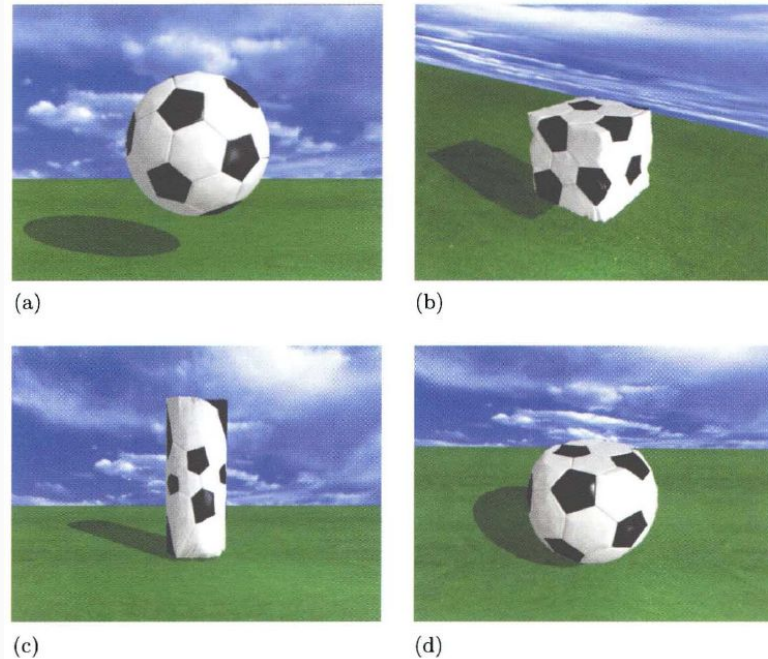


Figura 16: Objeto de controle de origem (a) e objetos de controle de destino (b), (c), (d).

Fonte: Adaptado de (JIN; LI, 2000)

2.8 - Deformação



35

Figura 17: Deformação de uma bola de futebol (a) com base nos objetos (b), (c) e (d) da Figura 16.
Fonte: Adaptado de (JIN; LI, 2000)

2.8 - Deformação

Outras duas técnicas são apresentadas por *Procházková (2017)*, o esquema de Sederberg, baseado em polinômios de Bernstein, e também o método NURBS (*Non-Uniform Rational B-Spline*).

36

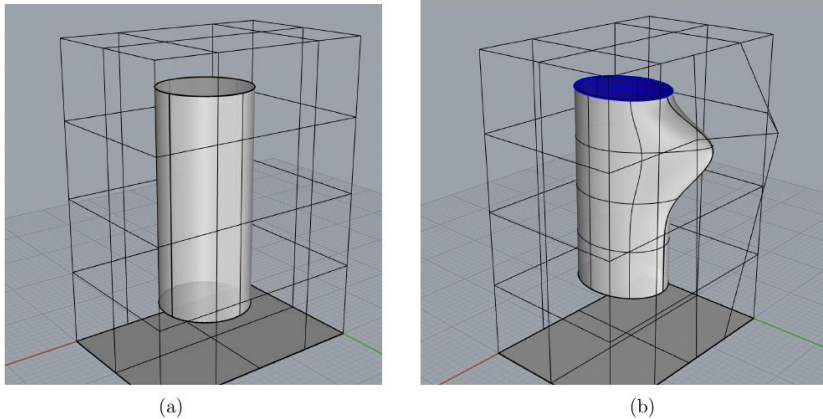


Figura 18: Deformação de forma livre de Sederberg.
Fonte: (*PROCHÁZKOVÁ, 2017*)

37

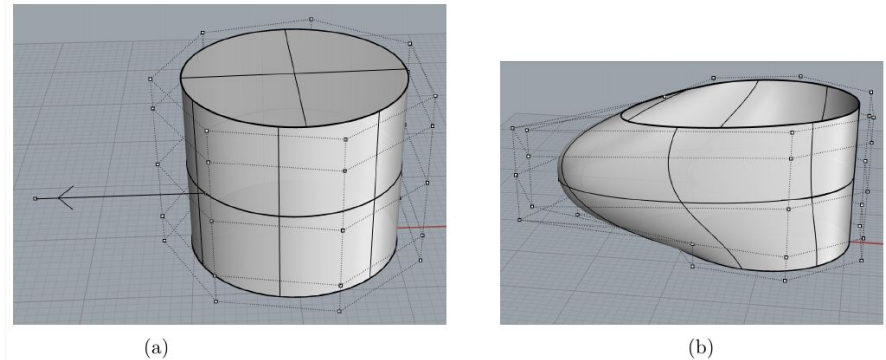


Figura 19: Deformação de forma livre utilizando NURBS.
Fonte: (*PROCHÁZKOVÁ, 2017*)

2.8 - Deformação

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



Figura 20: (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.

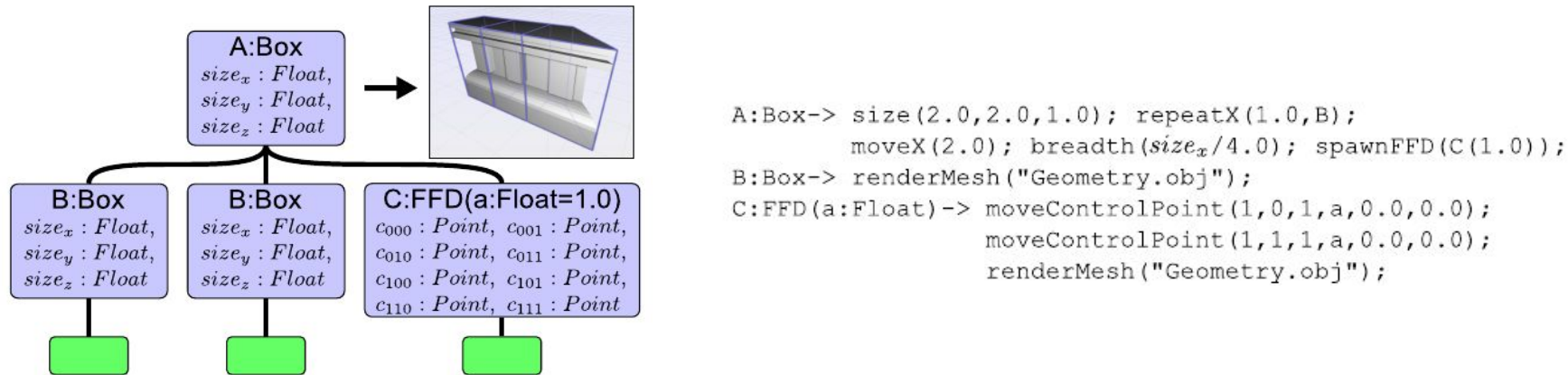


Figura 21: Aplicação de regras de modelagem da G^2 .

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.

47

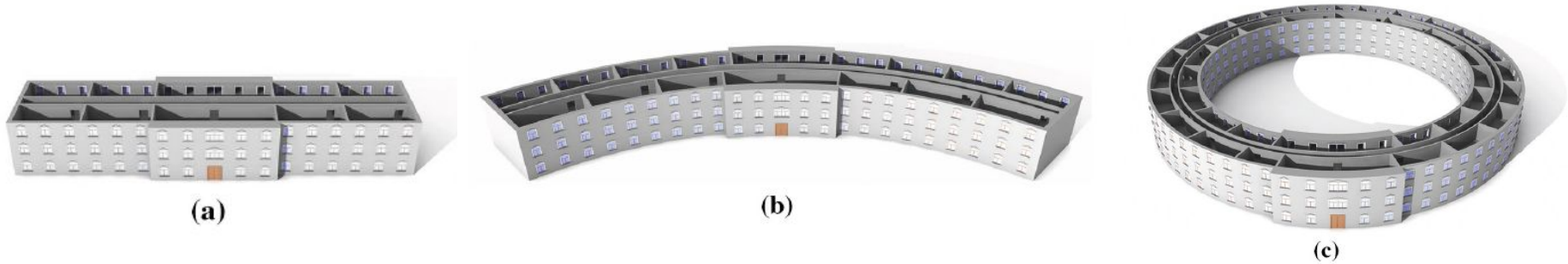


Figura 22: Prédio comercial com estrutura arredondada.
Fonte: Adaptado de (ZMUGG et al., 2014)

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.

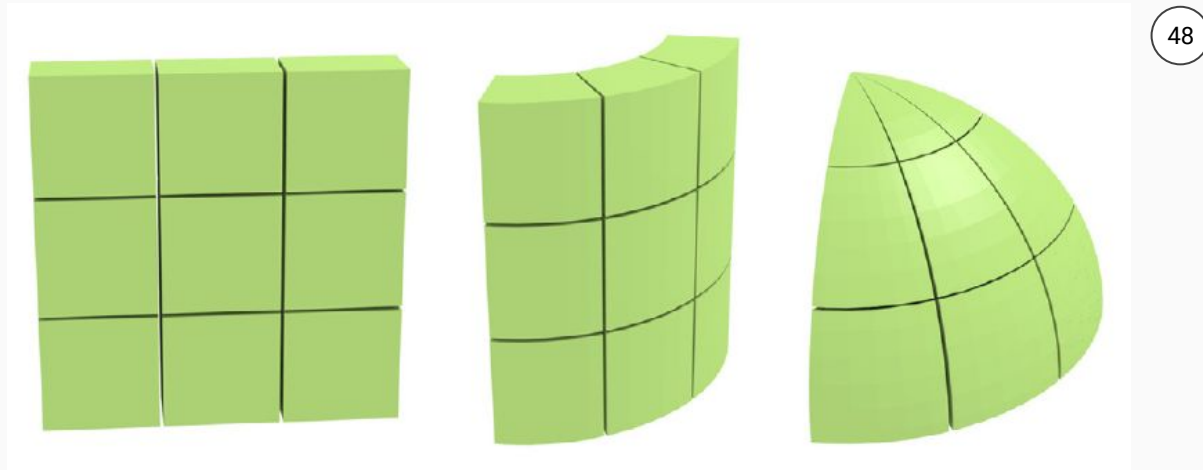


Figura 23: Divisão de formas com diferentes sistemas de coordenadas (cartesiana, cilíndrica e esférica).
Fonte: (EDELSBRUNNER et al., 2017)

4 - Proposta

4 - Proposta

4.1 - Problema

4.2 - Implementação e integração

4.3 - Representação das formas e operações

4.4 - Aplicação de deformação nos modelos

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



50

Figura 24: Exemplo que está além da capacidade de modelagem da *SELEX*.
Fonte: (Jiang et al., 2018)

4.2 - Implementação e integração

- Implementação do interpretador para a *SELEX*, por meio da utilização da biblioteca *pyparsing*;
- Implementação dos *scripts* para integração com o *software* de modelagem Blender (versão 2.83);
- Utilização das bibliotecas *bpy* e *bmesh* para modelagem.

4.3 - Representação das formas e operações

As formas de construção poderão ser representadas pelos objetos padrões presentes no Blender, como plano e cubo. Entretanto, um ponto importante de definição é a representação das formas virtuais, uma vez que elas ditam como são realizadas as subdivisões para operações de *design*.

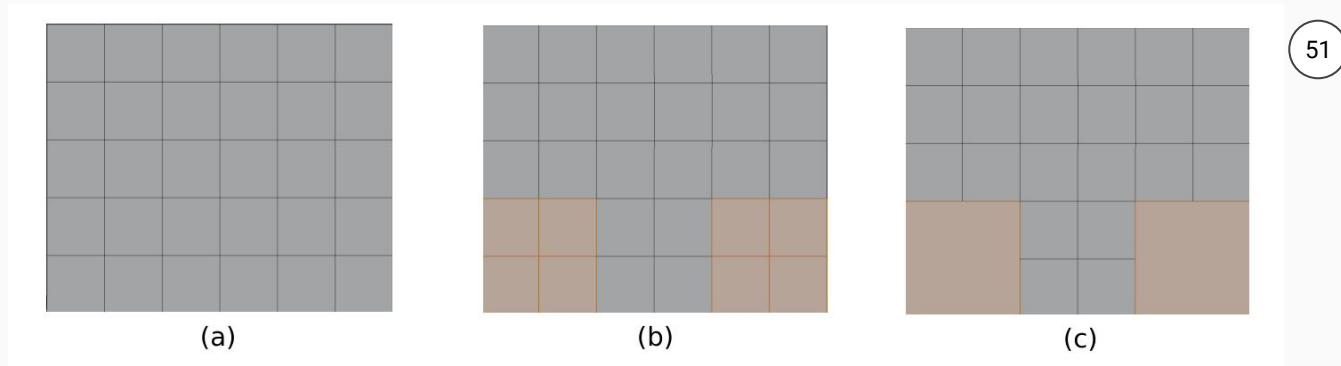
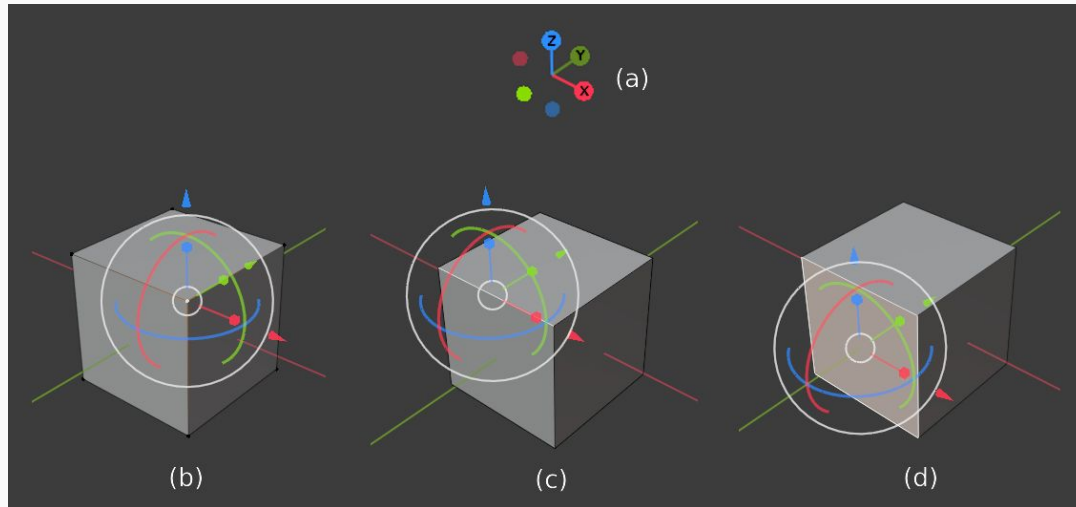


Figura 25: Exemplo prévio de subdivisão por meio da utilização de *scripts* no Blender.

Fonte: Próprio autor

4.4 - Aplicação de deformação nos modelos

O Blender dispõe de vários recursos para manipulação de vértices, faces e arestas, em relação a diferentes eixos.



52

Figura 26: Representação do sistema de coordenadas do Blender: (a) eixos, (b) vértice, (c) aresta, (d) face.

Fonte: Próprio autor

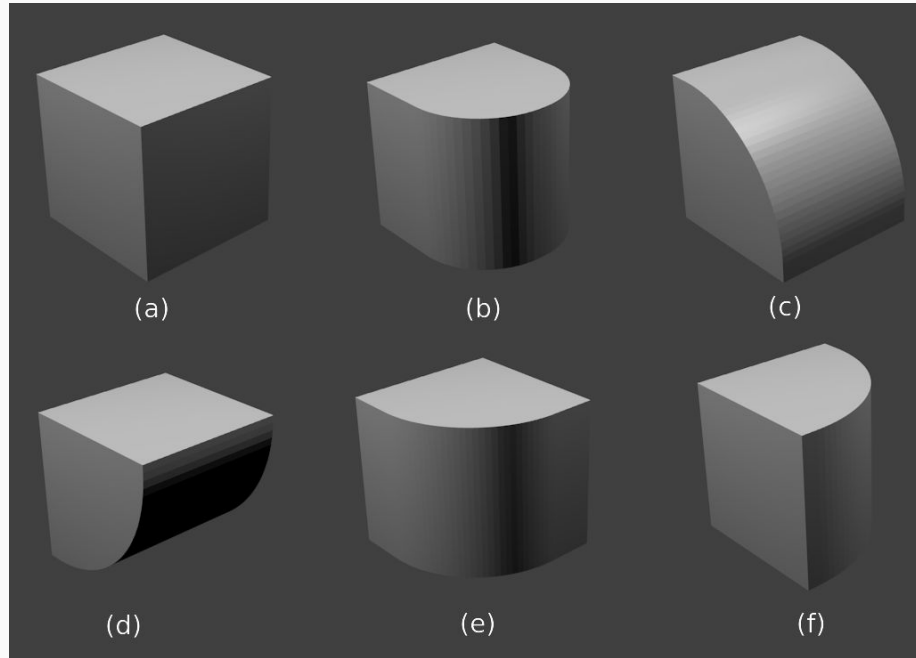
4.4 - Aplicação de deformação nos modelos

A proposta consiste na criação da operação *roundShape*, que recebe alguns parâmetros, como tipo de deformação, direção, nome da forma, e outros a serem especificados, podendo ser exemplificada da seguinte maneira:

- (i) *roundShape("roundFront", "outside", "cube", ...)*
- (ii) *roundShape("roundTop", "outside", "cube", ...)*
- (iii) *roundShape("roundBottom", "outside", "cube", ...)*
- (iv) *roundShape("roundLeft", "outside", "cube", ...)*
- (v) *roundShape("roundRight", "outside", "cube", ...)*

```
{<[label=="facade"]> -> roundShape("front", "outside", ...);}
```

4.4 - Aplicação de deformação nos modelos

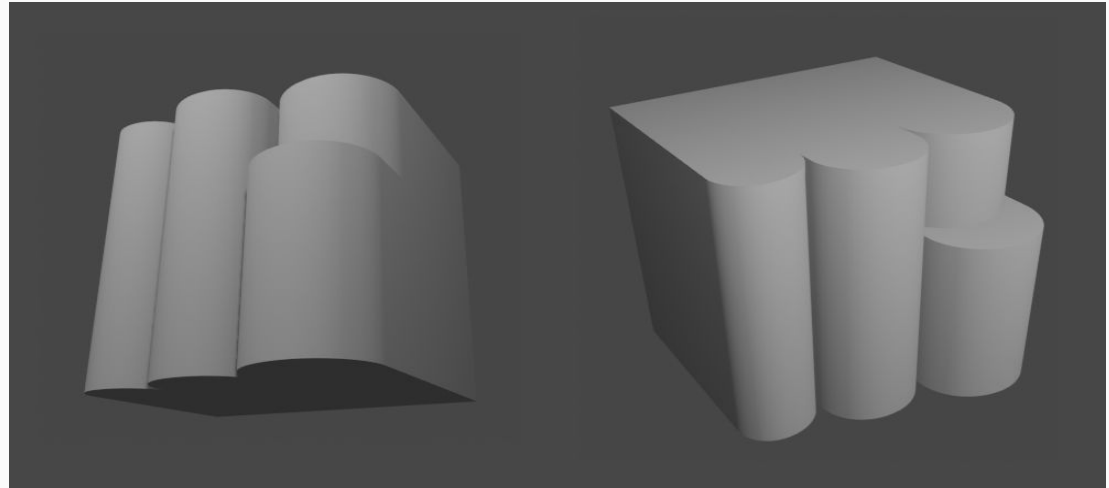


53

Figura 27: (a) Forma original, (b) *roundFront*, (c) *roundTop*, (d) *roundBottom*, (e) *roundLeft*, (f) *roundRight*.
Fonte: Próprio autor

4.4 - Aplicação de deformação nos modelos

- Exemplo do resultado esperado após formalização da operação *roundShape*:



54

Figura 28: Modelagem inicial e experimental da arquitetura da Figura 24 (ao lado).
Fonte: Próprio autor

5 - Cronograma

5 - Planejamento e próximas atividades

Atividades	2020.1										2020.2		
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar
Pesquisa do tema	X	X	X	X	X	X							
Definição do tema							X	X					
Pesquisa bibliográfica	X	X	X	X	X	X	X	X	X	X	X	X	X
Elaboração do projeto									X	X	X	X	X
Implementação do projeto										X	X	X	X
Apresentação e discussão do resultado												X	X
Entrega do projeto													X

Tabela 1: Cronograma com planejamento do trabalho.

Fonte: Próprio autor

6 - Referências

6 - 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.

6 - Referências

RODRIGUES, F. C. M. *Evolução de split grammars para construções procedurais*. 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.

6 - 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.

JIN, X.; LI, Y. F. *Three-dimensional deformation using directional polar coordinates*. J. Graph. Tools, A. K. Peters, Ltd., USA, v. 5, n. 2, p. 15–24, fev. 2000. ISSN 1086-7651.

PROCHÁZKOVÁ, J. *Free form deformation methods – the theory and practice*. 02 2017.

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.

6 - Referências

KRECKLAU, L.; PAVIC, D.; KOBBELT, 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.

6 - 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.

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



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
2020