



PROGRAMAÇÃO ORIENTADA A OBJETOS PARA DINAMIZAR A CRIAÇÃO DE DIAGRAMAS EM ARQUITETURA

Palavras-Chave: Realidade Aumentada Móvel, Programação Orientada a Objetos,
Interface Humano-Computador, Computação Gráfica, Diagramas 3D

Autores(as):

Aluno: Lucca Ferreira Paiva - Unicamp

Orientador: Prof.^a Dr.^a Daniel de Carvalho Moreira FECFAU - Unicamp

Co-orientadora: Larissa Negris de Souza FECFAU - Unicamp

INTRODUÇÃO:

A inserção da terceira dimensão nas representações gráficas de arquitetura envolveu a mudança de percepção do espaço representado, mas também permitiu a inserção de uma nova camada de informações que representações bidimensionais não comportam. Nesta pesquisa, a Realidade Aumentada (RA) é trabalhada como ferramenta para a visualização de formas geométricas em 3D, adicionando interações, em tempo real, com elas, permitindo também que se inter-relacionem. Este trabalho de Iniciação Científica (IC), intitulado “Programação orientada a objetos para dinamizar a criação de diagramas em arquitetura”, dá continuidade a um projeto anterior, ambos financiados pelo PIBIC/SAE da UNICAMP. O projeto atual é associado a uma pesquisa de Doutorado em progresso na Faculdade de Engenharia Civil, Arquitetura e Urbanismo da Universidade Estadual de Campinas (FECFAU/UNICAMP).

As formas geométricas definidas se relacionam aos Parâmetros de Projeto (PPs) para arquitetura escolar, que são representados por meio de diagramas, os quais possuem suas composições baseadas em regras de combinação de Elementos Básicos (EBs). O conceito de “parâmetros” tem sentido abstrato, representando uma possível solução para problemas recorrentes em arquitetura, embora sem possuir uma forma definida. Assim sendo, com estes parâmetros, é possível criar representações gráficas, que fazem o uso de elementos visuais para demonstrar relações de proximidade, suas possíveis conexões, e que indicam também a variação das formas, suas proporções e geometrias, aspectos tipicamente relacionados a um contexto projetual de arquitetura.

Dentro desse cenário, o objetivo geral da pesquisa é desenvolver e expandir interações entre objetos virtuais e suas regras, bem como a documentação deste processo, culminando em uma base para uma aplicação em RA para dispositivos móveis. A solução criada deve permitir ao usuário inserir e modificar formas, que representam os EBs, na interface, mudando a sua escala, cor, posição, rotação e textura. A proposta é desenvolvida com o uso da Unity Engine associado ao Vuforia SDK, para criar experiências de Realidade Aumentada. O trabalho a nível de códigos é em C#, aprofundando a dinâmica de interação entre e com os objetos virtuais criados. Para a implementação

da tecnologia serão utilizadas práticas de programação orientada a objetos (POO) com a intenção de representar os elementos gráficos gerados e sua conexão. Conceitos de interface humano-computador (IHC) serão aplicados para tornar o produto de fácil usabilidade, em especial no que diz respeito às técnicas voltadas para o design centrado no usuário (DCU).

METODOLOGIA:

Essa pesquisa é delineada pela metodologia de *Design Science Research*, que se caracteriza por estabelecer pontes entre o prático e o teórico e tem como um dos principais o desenvolvimento de artefatos, físicos, virtuais ou metodológicos, que podem ser compreendidos como algo desenvolvido por humanos e caracterizado por meio de suas funções, objetivos, adaptações (DRESCHÉ; LACERDA; ANTUNES JR., 2015).

As seguintes etapas compuseram o processo:

1. Levantamento de Requisitos para o sistema
2. Implementação:
 - Seleção de múltiplos objetos
 - Porte para dispositivos móveis
 - Sistema de carregamento e salvamento
 - Matriz de adjacência e exportação
3. Teste intensivo, revisão e refatoração (reescrita do código) das funcionalidades implementadas;
4. Sistematização e análise de resultados utilizando métodos de avaliação de interação, especialmente a observação participante;
5. Resultados e conclusão;

RESULTADOS E DISCUSSÃO:

Foi desenvolvida uma aplicação por meio da Unity Engine com suporte de scripts em C#. Para elaborar a solução foram utilizadas práticas de POO. A aplicação pode ser entendida por meio de componentes, códigos em C#, eles foram divididos em três grandes grupos (divisão apenas lógica); o primeiro, o das formas, que contêm os componentes presentes em cada objeto geométrico. Depois, o dos managers, que realizam tarefas mais globais de controle de fluxo. O último, o dos comandos, responsáveis por realizar as ações desejadas pelo usuário e a comunicação com inputs, ou entradas de comando, dele.

A atual fase da aplicação desenvolvida permite ao usuário realizar uma série de funcionalidades com as formas geométricas, que serão explicitadas a seguir. Deve-se lembrar que todas as regras de interação foram criadas baseadas nas regras estabelecidas na pesquisa de Doutorado citada, entendendo a relação entre elementos básicos para formar os parâmetros de projeto da arquitetura escolar. Não é do escopo dessa pesquisa apresentar cada detalhe por trás do significado das formas, mas sim os aspectos técnicos de sua criação e funcionamento.

Na aplicação proveniente da pesquisa anterior era possível criar formas, cubos ou esferas, interagir com elas, mudando sua posição, escala, rotação e cor. O artefato também possuía controles básicos, que permitiam realizar os comandos descritos previamente, mas eles eram limitados. A adição da Realidade Aumentada também foi implementada de forma inicial naquele momento.

No período desta pesquisa foi desenvolvido primeiramente uma forma de interagir com múltiplos objetos simultaneamente, que permite ao usuário selecionar vários objetos e realizar a mesma operação em todos eles simultaneamente. Exemplificando, é possível selecionar dois cubos e, ao mover um deles, o outro vai sofrer o mesmo deslocamento. Interações com mais de um objeto são válidas para as operações de translação, rotação e escala uniforme. Sendo assim, quando se está nesse modo, as opções de mudança de cor e de dimensão por eixo ficam desabilitadas. Isto pode ser visto na Imagem 1, na qual o contorno verde indica que o objeto está selecionado, logo, ao mover o cubo menor (com contorno verde), a mesma mudança de posição é aplicada no outro elemento selecionado. Vale ressaltar que formas não selecionadas permanecem inalteradas.

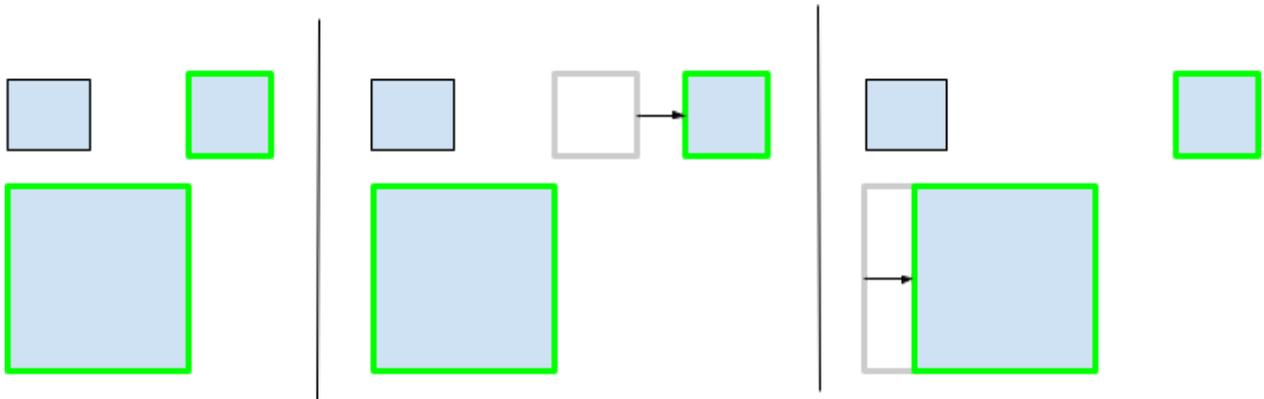


Imagem 1: Deslocamento de múltiplos objetos selecionados

Outra capacidade da aplicação é a de gerar a matriz de adjacência, de forma a mostrar para o usuário um relatório com as interações entre cada elemento. Durante o uso da aplicação esta matriz pode ser vista pela interface da Unity, Imagem 2, apenas disponível para os desenvolvedores. Desta forma, foi necessário criar uma maneira de deixar que o usuário acessasse esses dados utilizando um sistema que escreve essa informação em um arquivo CSV, o que permite manter as características da matriz.

▼ Adjacency Matrix								
Formas	0	1	2	3	4	5	6	7
0	None	Intersecti	None	None	None	None	None	None
1	Intersecti	None	None	None	None	None	None	None
2	None	None	None	Parcial	None	None	None	None
3	None	None	Parcial	None	None	None	None	None
4	None	None	None	None	None	Volume	None	None
5	None	None	None	None	Volume	None	None	None
6	None	None	None	None	None	None	None	None
7	None	None	None	None	None	None	None	None

Imagem 2: Matriz de Adjacência na Unity

Adicionalmente, o software desenvolvido tem o recurso de um sistema para carregar e salvar o progresso entre sessões, que funciona por meio da escrita em arquivos. O primeiro passo foi determinar quais objetos de cada cena deveriam ser salvos, conclui-se, após análise, que os componentes *Transform* e *Form* precisam ser salvos. O primeiro se refere ao conjunto de posição, escala e rotação do objeto, o segundo é um componente criado para gerir as informações de cada

objeto, principalmente as relacionadas às especificidades da pesquisa. A etapa seguinte foi a de serialização, processo de converter em texto essas informações, para que pudessem ser escritas em arquivos. As classes como *GameData*, *Data Persistence Manager* e *File DataHandler*, junto da biblioteca *Newtonsoft Json*, foram criadas com essa finalidade. Uma vez que os dados são serializados, eles são escritos em um arquivo *JSON* (Javascript Object Notation). No Código 1 é possível ver uma versão cortada e simplificada do arquivo gerado para uma cena de exemplo.

```
JavaScript
{id": 0,
  "type": 0,
  "pp": 0,
  "pos": {
    "x": -8.106925,
    "y": 2.77075148,
    "z": 28.3922,
  "rot": {
    "x": 0.0,
    "y": 0.0,
    "z": 0.0,
    "w": 1.0,
  "scale": {
    "x": 2.0,
    "y": 2.0,
    "z": 2.0,
  }
}
```

Também foram feitas melhorias da interface, visando uma apresentação amigável e de fácil aprendizado. Isto é feito por meio de um menu inicial, em que o usuário pode selecionar entre as opções de ter um tutorial da aplicação se desejar. Outras possíveis opções são a de carregar uma sessão anterior ou ir para definições.



Imagem 3: Tela da aplicação

CONCLUSÕES:

O produto desenvolvido até o momento é capaz de cumprir os requisitos funcionais, levantados com base no livro “Mastering the Requirements Process Getting Requirements Right” (ROBERTSON; ROBERTSON, 2013), para a representação e interação com elementos básicos que compõem parâmetros de projeto. Estes requisitos incluem: ser capaz de criar as formas geométricas, selecionar a forma com a qual se deseja interagir, interagir com elas mudando posição, rotação, escala e cor, estabelecer formas de interagir entre os elementos criados, interagir com múltiplos objetos, salvar o progresso entre sessões e exportar os dados das formas criadas.

Os requisitos não funcionais também foram implementados, isto inclui a interface para dispositivos móveis, em especial voltadas para princípios como ser *User Friendly*, de fácil aprendizagem e intuitiva. Por último, a utilização da ferramenta permite ao usuário capacitado gerir e interagir com os parâmetros de projeto através de uma nova abordagem.

BIBLIOGRAFIA

- DRESCHE, A.; LACERDA, D. P.; ANTUNES JR., J. A. V. **Design Science Research: A Method for Science and Technology Advancement**. Porto Alegre: Springer, 2015.
- ROBERTSON, S.; ROBERTSON, J. Mastering the Requirements Process Getting Requirements Right. **Work**, v. 44, 2013