

Automação de partícula de matéria programável com o uso de agentes em um ambiente virtual

Lucas André Gonçalves, Rafael Ballottin Martins

Laboratório de Inteligência Aplicada – Centro de Ciências Tecnológicas da Terra e do Mar – Universidade do Vale do Itajaí (UNIVALI)
Rua Uruguai, 458 - 88302-202 – Itajaí– SC – Brasil

lucassmart@edu.univali.br, ballottin@univali.br

***Abstract.** Although major breakthroughs have been made in research on programmable matter the communication and automation of the particles that compose it was little explored. This research proposed the development of a communication system using agents to obtain the desired form of particles of programmable matter autonomously. For this purpose, a virtual environment was created, using the Unity engine to perform the particle coupling simulation. The validation test of the particles of programmable matter in the virtual environment showed a success percentage of 80% in obtaining the desired shape with an average of 6500 movements and an average time of 6 minutes.*

***Resumo.** Embora grandes avanços tenham sido realizados nas pesquisas sobre matéria programável a comunicação e automação das partículas que a compõem foi pouco explorada. Este trabalho propôs o desenvolvimento de um sistema de comunicação com o uso de agentes para obtenção da forma desejada de partículas de matéria programável de maneira autônoma. Para tanto, foi criado um ambiente virtual, usando a engine Unity para a realização da simulação de acoplagem das partículas. A validação do funcionamento das partículas de matéria programável no ambiente virtual, apresentou um percentual de sucesso de 80% na obtenção da forma desejada com uma média de 6500 movimentos e um tempo médio de 6 minutos.*

1. Introdução

A matéria programável baseia-se na matéria comum, sendo composta de várias partículas, ou seja, os robôs modulares para gerar uma determinada forma. Portanto, a matéria programável tem como principal característica a capacidade de organizar-se para tomar a forma do objeto que se deseja. Diferentemente da matéria comum a matéria programável nos permite ter total controle sobre ela.

O controle das partículas se dá por intermédio dos robôs modulares, componentes da matéria programável, estas partículas individuais, que podem mover-se em uma dimensão 3D em relação às outras. Com a junção de tais partículas gera-se a forma desejada, a qual se mantém até segunda ordem. Cabe expressar que tal possibilidade é viável pelo fato de que cada partícula é uma unidade autossuficiente e detentora de componentes eletrônicos e peças específicas, variando de tamanhos ou formas [Goldstein et al. 2005, Kurokawa et al. 2006, Romanishin et al. 2013].

Assim sendo, toda partícula de matéria programável já nasce com a capacidade de processamento individual ou em conjunto se necessário. Isto possibilita a automação das partículas [Toffoli e Margolus 1991].

Os recentes trabalhos em matéria programável, tais como o CUBLI [Gajamohan et al. 2012], e o M-block [Romanishin et al. 2013], focam principalmente na parte física da partícula (o design do robô modular e as peças do robô modular), sendo pouco explorada a parte de comunicação e inteligência (script/método de montagem). Outros trabalhos que se destacam com foco maior em aspectos ligados a interação das partículas e, portanto, mais similares a proposta desta pesquisa, são, o Enxame de mil Robôs [Rubenstein, Cornejo e Nagpal 2014] e o Claytronics [Goldstein e Mowry 2004, Kirby et al. 2007], apesar de suas implementações (comunicação/método de montagem da partícula) também serem realizadas diretamente no robô modular.

Com base nos trabalhos da área, esta pesquisa propôs a criação de um simulador para desenvolvimento e testes de partículas de matéria programável em um ambiente virtual.

2. Simulador de Partícula de Matéria Programável (SPMP)

O "Simulador de Partículas de Matéria Programável" (SPMP), foi criado para ser o mais flexível e adaptável possível, podendo ser extensível pelo seu usuário. Ele vem com sua configuração padrão, a qual dispõe de algumas opções de escolha pré-definidas, permitindo, porém, a elaboração e criação de outras escolhas no futuro.

O ambiente tem uma interface em que o usuário monta a imagem desejada com as partículas, uma a uma, então o SPMP que contém a interface de entrada, irá comunicar-se com as partículas sobre a forma desejada.

2.1 Interface de entrada

Para desenhar o formato desejado devem ser colocadas e posicionadas cada uma das partículas. Para a realização da tarefa pode se usar os botões de ações: *add*, *remove*, *initial*, *move* e *lock*.

- *Add*: Adiciona uma partícula ao desenho, em um clique com o botão esquerdo do mouse;
- *Remove*: Exclui uma partícula do desenho, em um clique com o botão esquerdo do mouse na direção da partícula que se deseja remover;
- *Move*: Permite movimentar a partícula desejada;
- *Lock*: Trava uma partícula, impedindo que ela seja alterada enquanto estiver no seu estado de travada;
- *Initial*: Define manualmente uma partícula como inicial;

O botão "-->" (*Run*) inicia a simulação, que rodará com as configurações padrão se não for previamente alterada. Como demonstra a Figura 1.

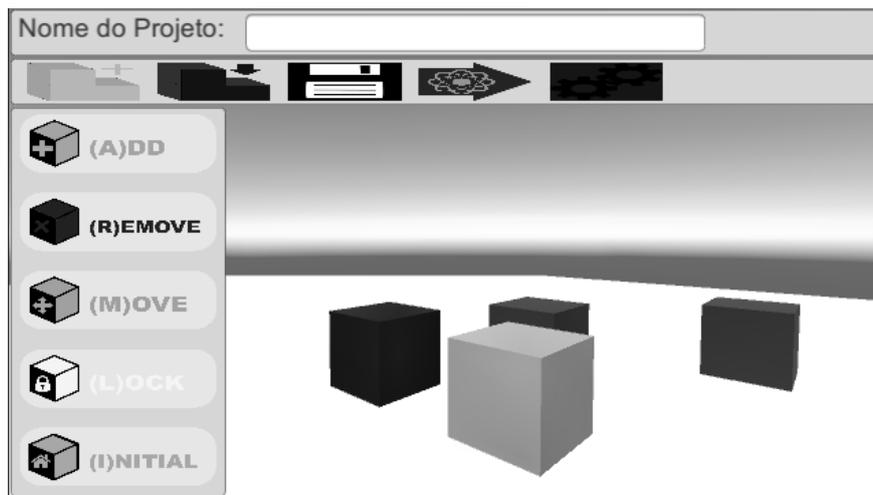


Figura 1. Interface de entrada

2.2 Ambiente de Simulação

O ambiente virtual em que ocorre a simulação da acoplagem das partículas, é inicialmente uma sala vazia, apenas com o chão e as paredes com espaço suficiente para que as quatro partículas possam se locomover e se acoplar.

A posição inicial das partículas é distribuída aleatoriamente, desta forma é possível testar melhor a automação das partículas, já que elas devem funcionar independentemente de onde estejam inicialmente. A distribuição padrão é centralizada, que distribui aleatoriamente a partir do centro da sala.

Ao iniciar a simulação, a câmera começa na posição em que ela estava no ambiente de edição. Os objetivos, onde as partículas devem chegar, ficam semitransparentes. As partículas então são inseridas de acordo com o tipo de distribuição e na sequência elas começam a se comunicar e movimentar-se de acordo com o script selecionado. Quando as partículas atingem o objetivo a simulação encerra.

2.3 Agentes

Foram desenvolvidos apenas dois tipos de agentes: o controlador, que é o software de entrada (SPMP) e as partículas. Por isso, a maioria dos agentes utiliza uma arquitetura reativa, pois se encaixa melhor com as características de se ter vários agentes mais superficiais.

O controlador, que é o agente cognitivo, tem a função de comunicar o objetivo e de manter a comunicação com as partículas.

Já as partículas, que são reativas, tem a função de se comunicar com o controlador para possibilitar, com o uso de sua base de dados, o trabalho em conjunto para a execução da tarefa até a conclusão da forma desejada. Inicialmente os agentes partícula teriam uma característica um pouco diferente, porque teriam a capacidade de serem promovidos à líderes e serem destituídos pelo controlador. Mas, devido a conflitos nas tarefas dos líderes a funcionalidade foi modificada.

2.3.1 Comunicação

A comunicação se dá por troca de mensagens e seriam usadas as bibliotecas de multiplayer e network da Unity.

No início da pesquisa, pensava-se em utilizar o netwokView, mas devido a falhas na comunicação, houve a necessidade de se refazer totalmente a parte de network e boa parte do projeto, foi então pensado sobre a possibilidade de usar o networkIdentity. Embora exista documentação para a networkIdentity, a substituta da netwokView, nos fóruns de Unity oficial e de terceiros, percebeu-se que há muitos tópicos de erros de network.

Seguindo como estratégia na tentativa de solucionar o problema foi criado um método que simula uma conexão, utilizando-se de uma função do Unity, o PlayerPrefs. Foram criadas 3 palavras-chave, que por sua vez serviam como destinatário, mais o assunto, remetente e conteúdo. São @MensagemSTR (para dados do tipo strings, assunto), @MensagemINT (para dados do tipo int, remetente), @MensagemFLOAT (para dados do tipo float, conteúdo), em alguns casos não era necessário passar algum conteúdo pois o próprio assunto já dizia o que se desejava, como por exemplo: Conexão, que já indica que a partícula está solicitando uma conexão ao controlador.

Para identificar o destinatário, cada partícula e o controlador tinham essas 3 palavras-chave como suas caixas de mensagem, sendo o número de id, adicionado ao final das palavras-chave, no caso das partículas. Nesse caso, sempre que fosse necessário mandar uma mensagem para o outro, a tarefa deveria ser realizada através destas 3 palavras. Conforme exemplo demonstrado na Figura 2

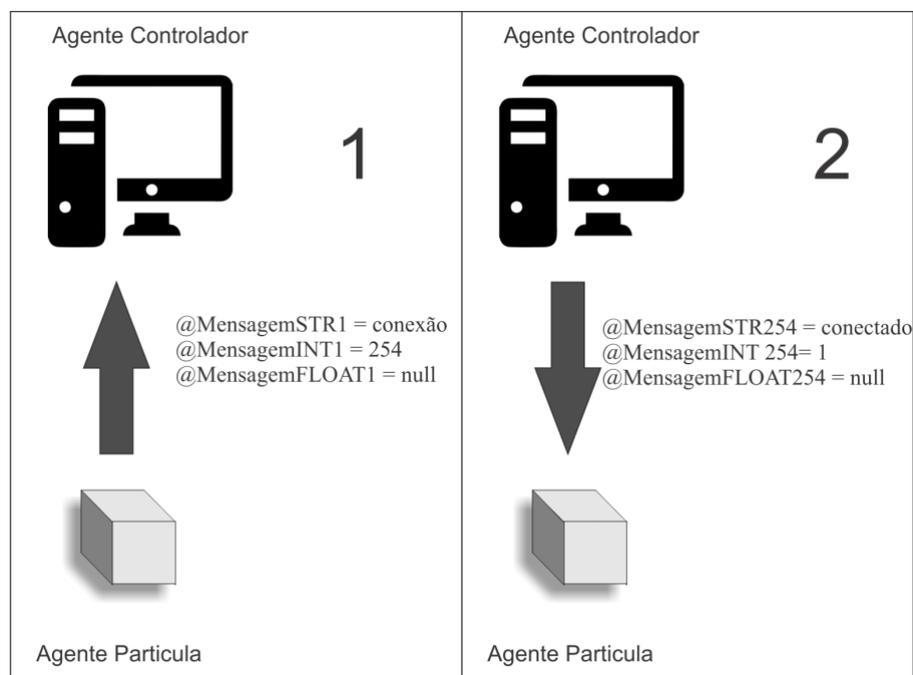


Figura 2. Comunicação entre partículas e controlador

2.4. Script

Inicialmente havia se pensado em um script de posicionamento de vizinhos em que as partículas começariam coladas umas nas outras, mas durante a execução do projeto

percebeu-se que isso não retrataria a realidade nem permitiria que as partículas fossem totalmente autônomas, já que no ambiente real se fosse aplicado, uma pessoa iria ter que inicialmente, posicioná-las manualmente.

Então optou-se por usar um script de alinhamento de eixos com base nas coordenadas das partículas, o que permitiria que elas fossem espalhadas aleatoriamente no ambiente, atingindo assim um dos objetivos do trabalho que é a total automação das partículas sem qualquer interferência humana em sua montagem.

O script funciona alinhando cada eixo das partículas, primeiramente se alinhando com o eixo X, depois se igualando no eixo Z e por fim sendo nivelada no eixo Y se necessário. Por se tratar de uso de coordenadas, poderia ser usada a precisão total, mas com o intuito de realizar a simulação o mais próximo possível da realidade, a precisão foi ajustada. O método de triangulação de sinal, que permite o uso de coordenadas no mundo real, não é tão preciso e foi adicionada uma tolerância, que pode ser configurada em metros. A Figura 3 exibe os alinhamentos e em que ordem eles ocorrem.

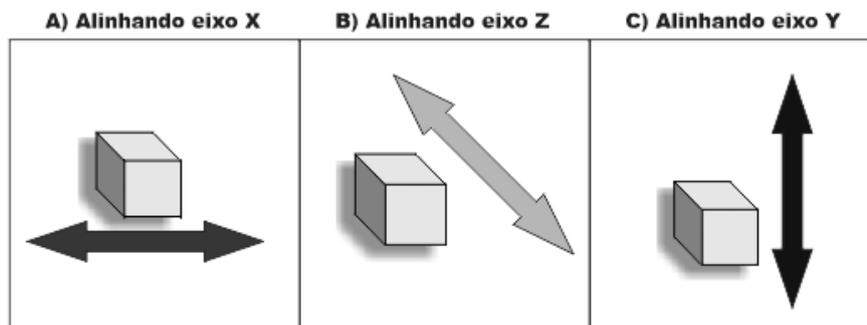


Figura 3. Alinhamentos da partícula

Durante os testes de produção foi possível observar que as partículas às vezes colidiam e ficavam colidindo quando seus objetivos ficavam em lados opostos ou na mesma posição em um dos eixos.

Então se buscou uma solução reativa, usando ciclo de alinhamento, buscando eliminar ou minimizar esse comportamento, ou seja, após elas se chocarem elas reagiriam de acordo com a colisão, resolvendo a colisão no próximo eixo, por exemplo: uma partícula A colide com seu lado direito (eixo X) com a partícula B no seu lado esquerdo (eixo X), então a partícula A se move para trás (eixo Z) e a partícula B para frente (eixo Z). A Figura 4 retrata o comportamento de reação após a colisão no eixo X.

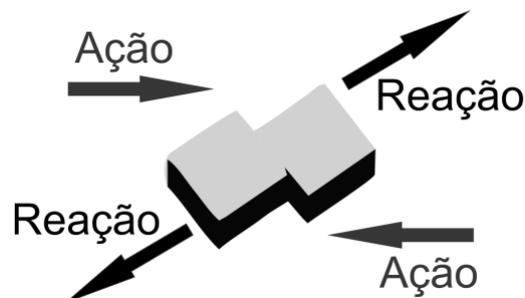


Figura 4. Colisão e reação da partícula

No eixo Z, inicialmente havia se pensado que se houvesse colisão na frente, a partícula saltaria, e atrás ficaria parada já que já estava no chão e não teria como descer. Mas durante os testes na fase de produção foi percebido que se a partícula ficasse parada ou tenta-se ir para baixo, caso o objetivo fosse no segundo nível em Y, e ela não se alinhasse antes em Z do que a de nível 1 ela não chegaria ao seu objetivo. Então decidiu-se fazer ela pular também e para garantir que fosse resolvido o impasse foi decidido que o fator que desempataria essa condição seria a força usada. A partícula que colidiu atrás usa mais força para saltar do que a partícula que colidiu na frente ao saltar, assim evitando ou minimizando o problema.

3. Testes e Validação

Após a conclusão do programa foram feitos alguns testes para observar o comportamento do mesmo e para se decidir os parâmetros a serem usados para os testes de validação. Nesses testes foram usados valores aleatórios para ajuste do seu comportamento, alterando-se os valores até que chegassem a valores equivalentes ao de interesse para os testes, que era o de simular o mais precisamente o ambiente real.

3.1. Parâmetros

Na Tabela 1 são indicados os valores dos parâmetros usados para as simulações em seus respectivos cenários.

Tabela 1. Parâmetros usados nos Cenários

Parâmetros	Cenário A	Cenário B
Força Básica: usado no cálculo da adição de força.	2	2
Força de Desempate Mínima: um dos parâmetros aplicados na força para desempatar ou/e resolver conflitos.	8	8
Força de Desempate Máxima: um dos parâmetros aplicados na força para desempatar ou/e resolver conflitos.	10	10
Propulsão: força para ir contra a gravidade e atingir objetivo em Y.	1.1	1.1
Força Breack: força limite de uma acoplagem.	12	12
Força Fixo: força inicial usada para quando as partículas se acoplam.	0	0
Tolerância: é o valor proporcional (relativo ao tamanho da partícula) permitido de diferença entre a posição real da partícula e do seu objetivo	2%	10%
Tempo Ação: é o tempo em unidade entre a reação e ação da partícula.	1	1
Tempo Reação: é o tempo em unidade de espera para reagir após a última colisão.	20	20
Tempo Máximo de Espera: é o limite de espera para reagir	60	60
Distancia: é a distância a partir do centro da partícula até sua borda.	0.026	0.026
Testes Totais	35	70
Maquinas	A	B

Na Tabela 2 são indicados os valores dos parâmetros usados para as simulações em seus respectivos cenários.

Tabela 2. Parâmetros usados nos Cenários

Configuração	Máquina A	Máquina B
Tipo	Notebook	Computador
Processador	Intel core i5-2450m 2.5Ghz com turbo <i>boost</i> para 3.1Ghz	AMD fx-8370E 8 núcleos 3.3 Ghz
Memória Ram	8GB	8GB
Placa de vídeo	Intel HD <i>Graphics</i> 3000 com 802 MB.	Gigabyte <i>Radeon</i> R7 240 2Gb DDR 3
Média de Movimentos por Segundo	17,30	32,22

3.2 Condições e Critérios

O teste tem como entrada: todas as 7 formas não isomórficas possíveis com o uso de 4 partículas em um ambiente 3D [SALVI et al. 2013]. Todos os testes foram executados com os parâmetros do sistema SPMP padrão. Buscando validar as simulações foram observados, quantas vezes cada partícula se moveu, a quantidade total de movimentos da simulação, a média de movimentos por partícula na simulação, quanto tempo durou a simulação e se foi possível se obter a forma desejada. Para este último critério foram definidos quatro resultados:

- Concluído: Todas as partículas chegaram no objetivo, respeitando a tolerância e consequentemente formando a imagem desejada;
- Forma Concluída: A imagem foi concluída, mas uma ou mais partículas que compõem a forma desejada estão fora da tolerância;
- Incompleto: Mais da metade das partículas chegaram ao seu objetivo e respeitaram a tolerância, mas não formaram a forma desejada;
- Não concluído: nenhum dos casos anteriores foi atingido.

Foram realizados um total de 105 simulações que foram compostas de 15 execuções para cada posição das 7 formas não isomórficas possíveis com o uso de 4 partículas em um ambiente 3D e foi estipulado como tempo máximo 10 minutos em ambos os cenários. Todos os testes foram gravados, observados com auxílio de um cronometro e a informação de movimentos realizados foi obtida através do retorno de cada partícula, o teste foi executado dentro da própria Unity, pois possibilitava o acesso a duas câmeras (uma do SPMP e outra do Unity) permitindo uma melhor observação dos testes.

Um teste era considerado concluído ao chegar ao tempo máximo ou se todas as partículas não se movessem mais. É considerado um movimento sempre que uma partícula usa a função de movimento e aplica uma força em qualquer direção.

4. Resultados

Nesta seção são apresentadas duas tabelas com resultados de testes em dois cenários ('A' e 'B') com a forma, a quantidade de testes feitos (QTD), quantos foram conclusivos

(concluídos (C), forma concluída (F), incompleta (I), não concluída (N)), média de movimentos totais (MMT), média de movimentos por partícula (MP) e a média de tempo (em minutos (T(m)) e em segundos (T(S))). Os resultados dos testes no cenário A são apresentados na Tabela 3 e do cenário B na Tabela 4.

Tabela 3. Resultados dos testes Cenário A

Forma	Qtd	C	F	I	N	MMT	MP	T (m)	T (s)
1	5	0	4	0	1	10849,4	2712,35	10:00	600
2	5	2	2	0	1	8027	2006,75	06:33	393,2
3	5	4	0	0	1	4204	1051	04:52	292,2
4	5	4	0	1	0	4222,4	1055,6	04:14	254,6
5	5	4	1	0	0	6242,2	1560,55	04:52	292,8
6	5	2	2	0	1	7381	1845,25	06:35	395
7	5	2	1	2	0	5100,2	1275,05	07:12	432,8
Total	35	18	10	3	4	6575,171	1643,7928	06:19	380,085

Através da Tabela 3 pode-se observar que 51,4% dos testes tiveram resultados positivos e que 28,6% dos resultados formaram a forma desejada, mas não estavam dentro do limite da tolerância de 2% do tamanho do cubo de 5cm³, 8,6% dos testes ficaram incompletos e 11,4% não concluíram. A média de movimentos totais foi de 6500 movimentos, tendo uma média de 1643 movimentos por partícula e uma execução média de 6 minutos e 19 segundos para conclusão das formas.

Tabela 4. Resultados dos testes Cenário B

Forma	Qtd	C	F	I	N	MMT	MP	T (m)	T (s)
1	10	6	2	1	1	15209,5	3802,375	07:05:42	425,7
2	10	9	0	1	0	4274,3	1068,575	1:49:48	109,8
3	10	10	0	0	0	2148,7	537,175	1:01:36	73,6
4	10	7	0	3	0	10369,5	2592,375	5:17:18	317,3
5	10	6	2	2	0	9480	2370	5:52:12	352,2
6	10	10	0	0	0	1705,5	426,375	0:47:24	47,4
7	10	5	0	5	0	11254,6	2813,65	6:03:42	363,7
Total	70	53	4	12	1	7777,442857	1944,360714	03:59:40	241,3857143

Através da Tabela 4 podemos observar que houve uma mudança nos resultados já que houve um aumento da tolerância no Cenário B. Com a alteração ocorreu uma mudança na porcentagem de ocorrências de cada resultado, em que é possível observar que 75,7% dos testes tiveram resultados positivos o que foi um aumento de 24,3% em relação ao cenário A, que 5,7% dos resultados formaram a forma desejada, mas não estavam dentro do limite da tolerância de 10% do tamanho do cubo de 5cm³, o que representa uma queda de 22,9% se comparado ao cenário A, 17,1% dos testes ficaram

incompletos onde ocorreu um aumento de 8,5% em relação ao cenário A e 1,4% não concluíram, o que representou uma queda de 10% se comparado ao cenário A.

Na Figura 5 é demonstrado graficamente os resultados finais da pesquisa, nos cenários A e B.



Figura 5. Resultado final para os dois cenários

De maneira geral (cenário A e B) é possível observar que 67,6% dos testes foram Concluídos, 13,3% Forma concluída, 14,3% Incompleto e 4,8% Não concluído. Ou seja, 80,9% dos resultados formaram as formas desejadas (dentro e fora da tolerância) e 19,1% não conseguiram tomar a forma desejada.

5. Conclusão

Os resultados obtidos no cenário A foram positivos com 80% formando as formas desejadas, dentro e fora da tolerância, e 20% não conseguiram tomar a forma desejada, com uma média de 6500 movimentos e um tempo médio aproximado de 6 minutos. No cenário B os resultados também foram positivos com 81,4%, formando as formas desejadas, dentro e fora da tolerância, e 19,6% não conseguiu tomar a forma desejada, com uma média de 7700 movimentos e um tempo médio de 4 minutos.

Durante a execução dos testes também foi possível observar que as chances de sucessos são maiores se forem concluídas primeiro as posições centrais e de baixo para cima.

Enfim, como o ambiente de simulação foi desenvolvido para permitir a criação de partículas através de diferentes técnicas, futuros trabalhos podem propor a implementação de scripts mais avançados ou técnicas de inteligência artificial. Outras possibilidades incluem o aprimoramento na evolução do sistema de comunicação e automação, a expansão do sistema através da criação de novos ambientes de testes, novas partículas de matéria programável, novas formas de locomoção de partícula, novas formas de localização, a gamificação da ferramenta para uso na área educacional, novas físicas para serem testadas e simuladas em ambientes diferentes, como se estivessem submersos em líquidos ou no espaço, além de novos estudos sobre matéria programável.

Referências

- GAJAMOHAN, Mohanarajah et al. The cubli: A cube that can jump up and balance. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012. p. 3722-3727
- GOLDSTEIN, Seth C.; MOWRY, Todd C. Claytronics: A Scalable Basis For Future Robots, RoboSphere, 2004.
- GOLDSTEIN, Seth Copen; CAMPBELL, Jason D.; MOWRY, Todd C. Programmable matter. Computer, v. 38, n. 6, p. 99-101, 2005.
- KIRBY, Brian T. et al. A modular robotic system using magnetic force effectors. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2007. p. 2787-2793.
- KUROKAWA, Haruhisa et al. Distributed Self-reconfiguration control of an M-TRAN System. In: Robitics: Science and Systems Workshop on Self-reconfigurable Modular Robots, Philadelphia. 2006.
- ROMANISHIN, John W.; GILPIN, Kyle; RUS, Daniela. M-blocks: Momentum-driven, magnetic modular robots. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013. p. 4288-4295.
- RUBENSTEIN, Michael; CORNEJO, Alejandro; NAGPAL, Radhika. Programmable self-assembly in a thousand-robot swarm. Science, v. 345, n. 6198, p. 795-799, 2014
- SALVI, Anelize Zomkowski et al. Novos métodos para enumeração de configurações não isomorfias de robôs metamórficos com módulos quadrados. 2013.
- TOFFOLI, Tommaso; MARGOLUS, Norman. Programmable matter: concepts and realization Proceedings of the NATO advanced research workshop on Lattice gas methods for PDE's : theory, applications and hardware: theory, applications and hardware North-Holland Publishing Co. Amsterdam, The Netherlands, The Netherlands P. 263-272, 1991