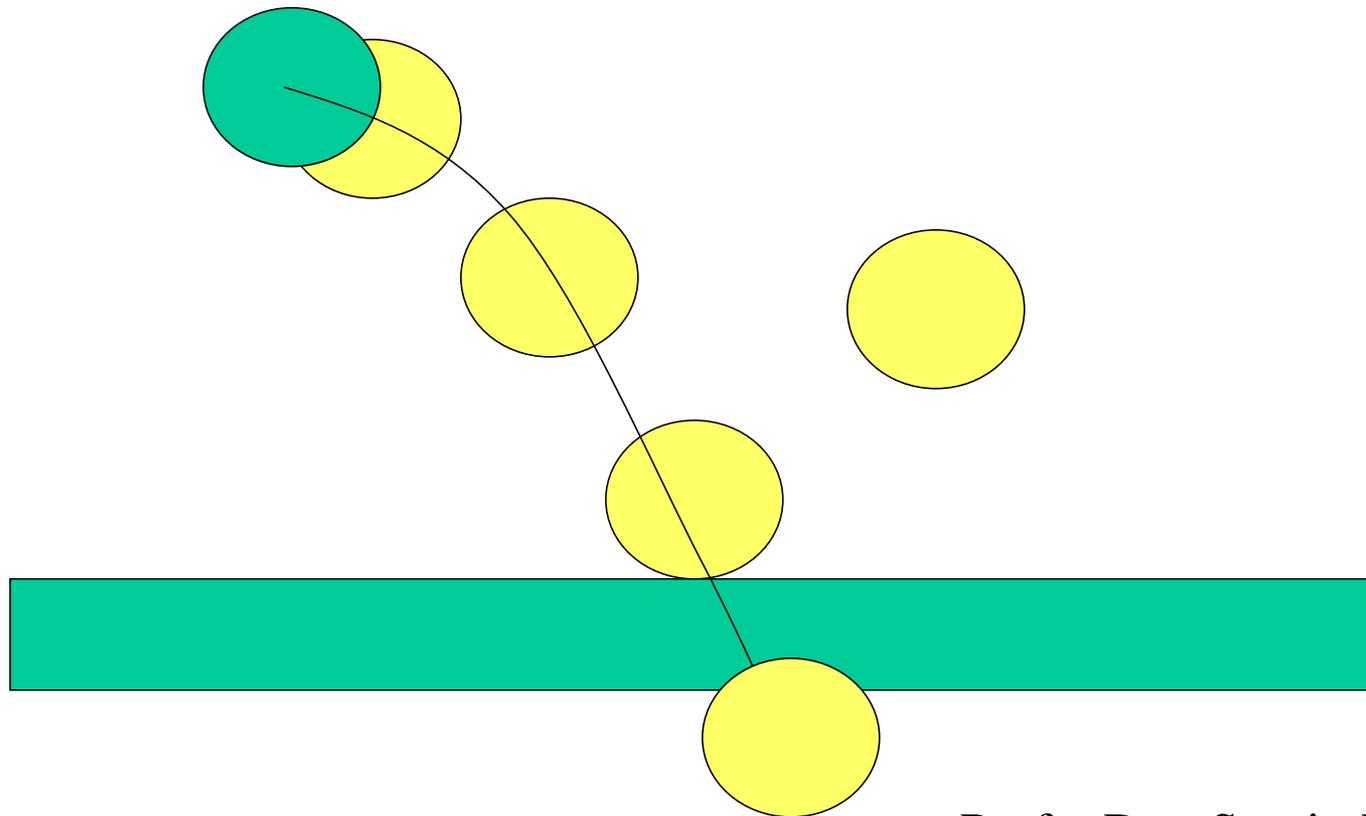
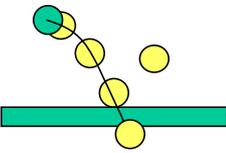


O Problema da Colisão

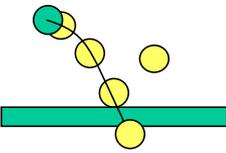


Profa. Dra. Soraia Raupp Musse



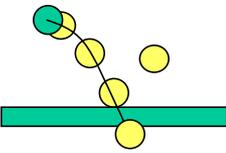
Porque tratar a colisão?

- Necessidade de repetir fenômenos físicos prevendo penetrações geométricas



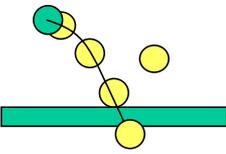
Objetivo

- Encontrar as interferências geométricas entre objetos:
 - Contatos
 - Interpenetrações
 - Proximidades



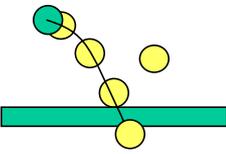
Aplicações

- Robótica
- CAD
- Motion planning
- Simulações mecânicas
- Animação



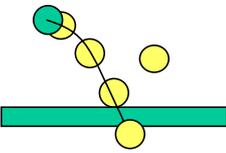
Aplicações: Robótica

- Verificar se os movimentos são compatíveis com obstáculos imóveis e móveis



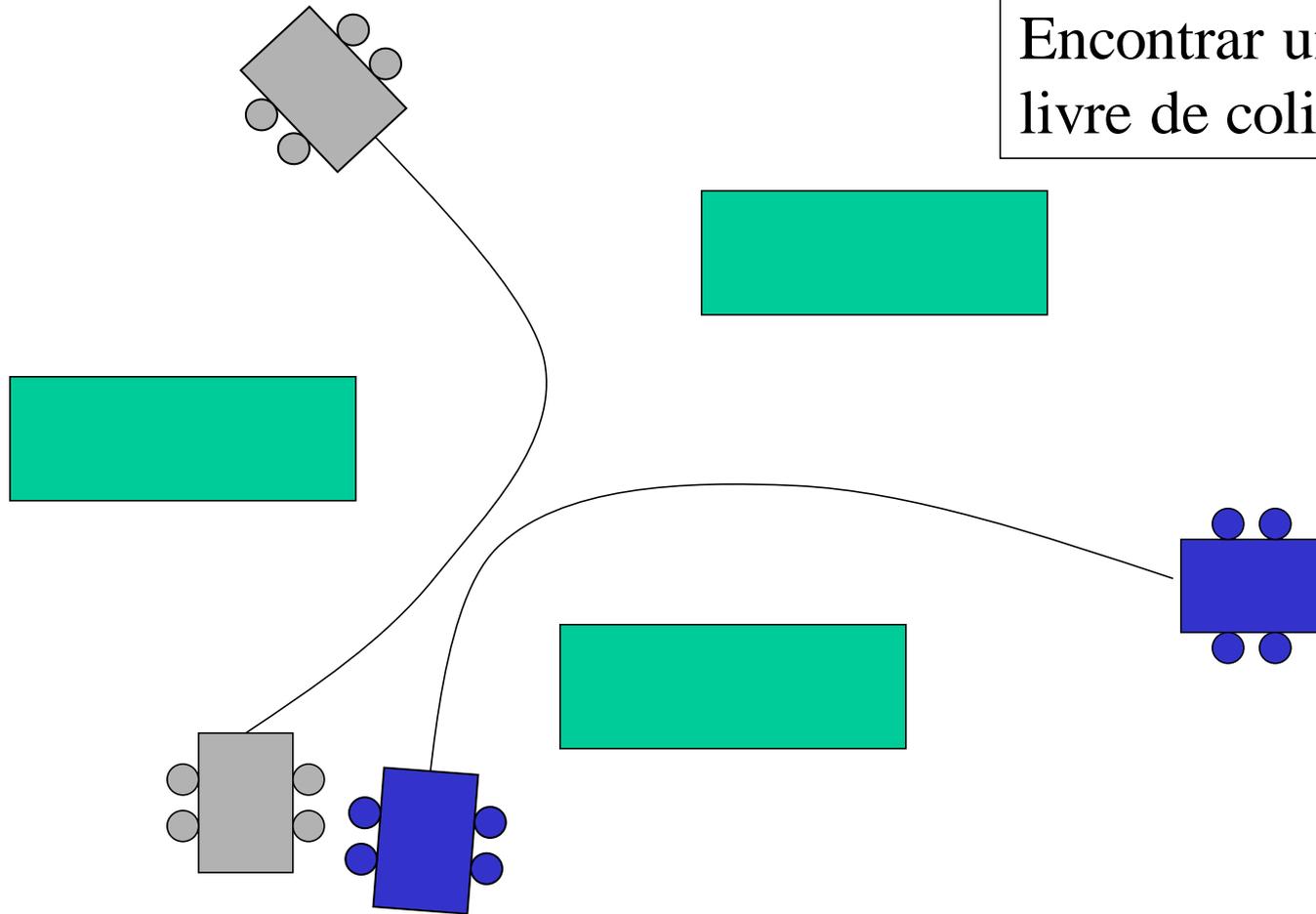
Aplicações: CAD

- Arquitetura
- Design mecânico

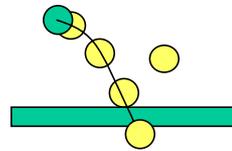


Aplicações: Motion Planning

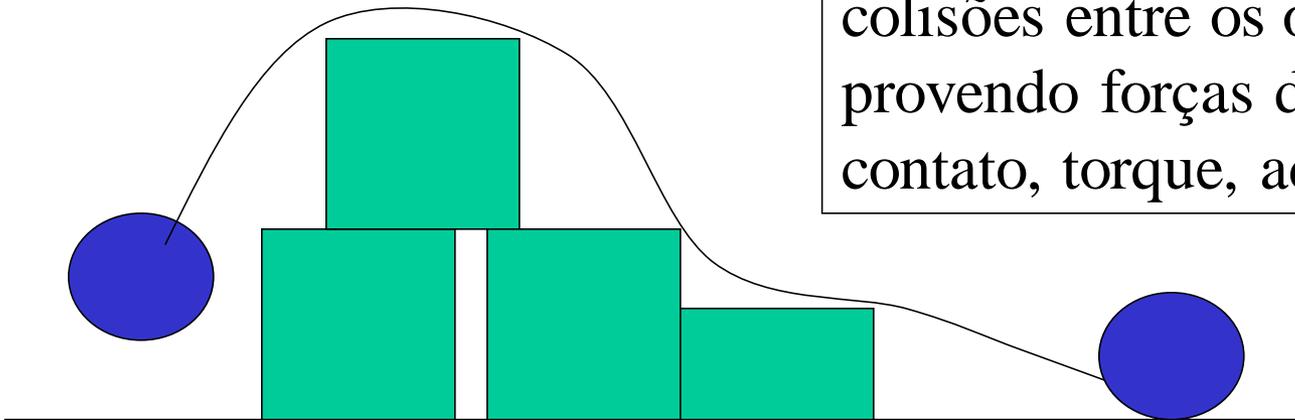
Encontrar um caminho livre de colisões

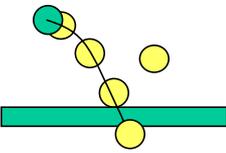


Aplicações: simulações mecânicas



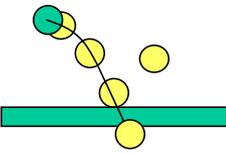
Calcular as restrições geométricas e colisões entre os objetos simulados provendo forças de reação resultantes: contato, torque, acelerações, etc





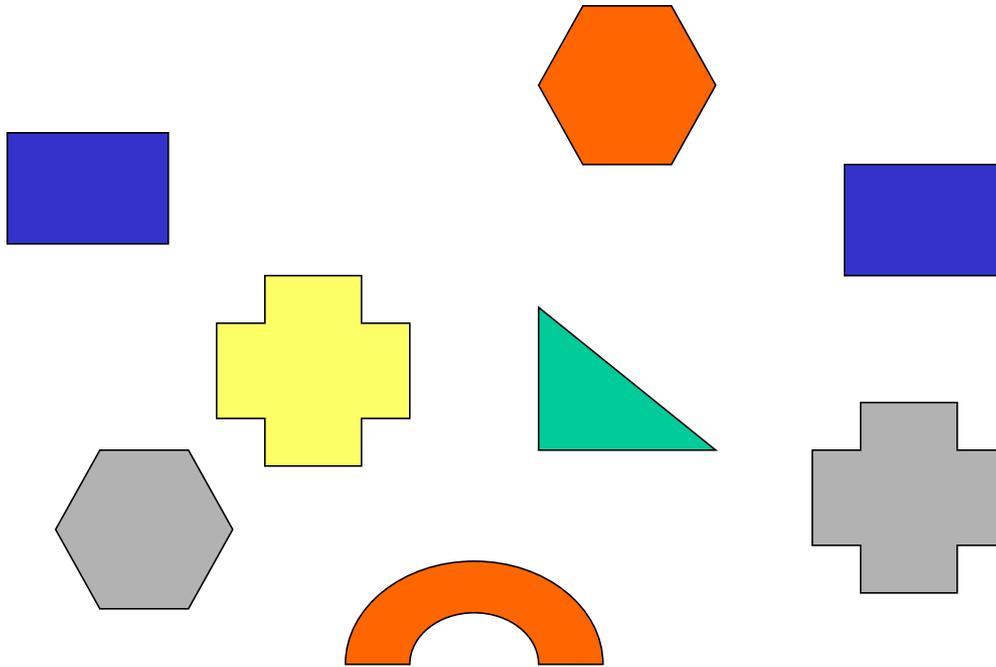
Aplicações: Animação

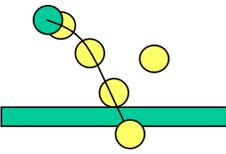
- Ferramentas para design de cenas realísticas
- Possibilitar agentes autônomos que evoluem num ambiente realista
- Prover interações com objetos (grasping)



Problemas

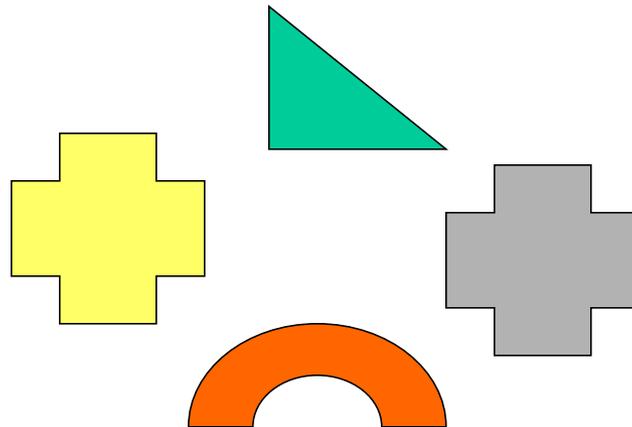
- Grande número de objetos

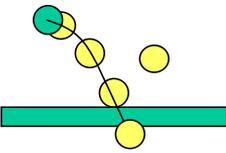




Problemas

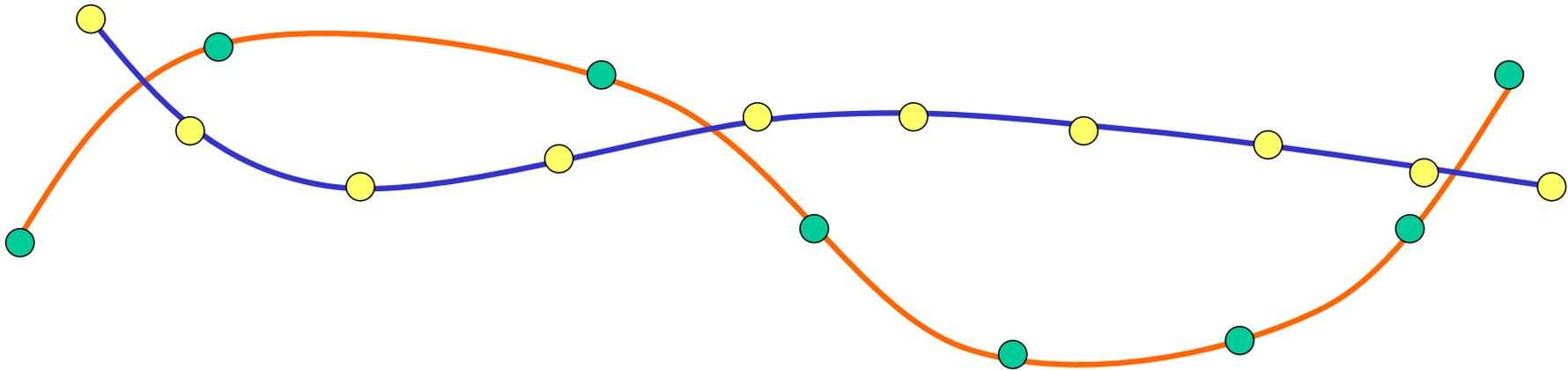
- Grande número de objetos
- Objetos complexos

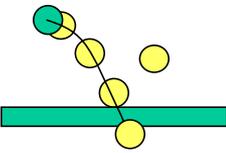




Problemas

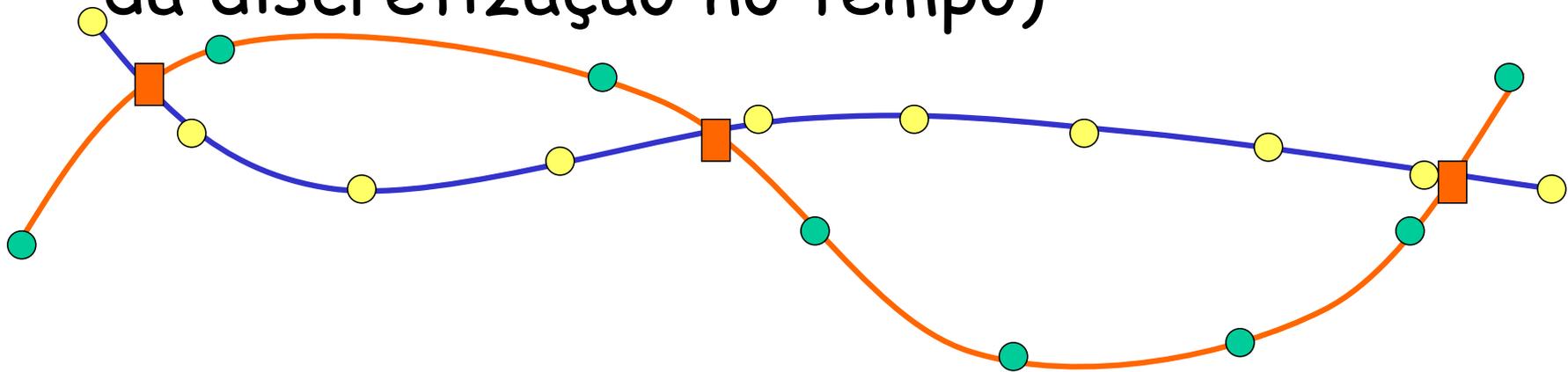
- Grande número de objetos
- Objetos complexos
- Colisão entre primitivas

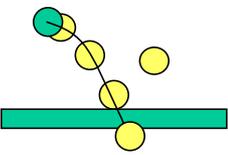




Problemas

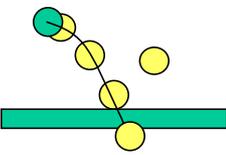
- Grande número de objetos
- Objetos complexos
- Colisão entre primitivas (dependendo da discretização no tempo)





3 níveis de resolução do problema

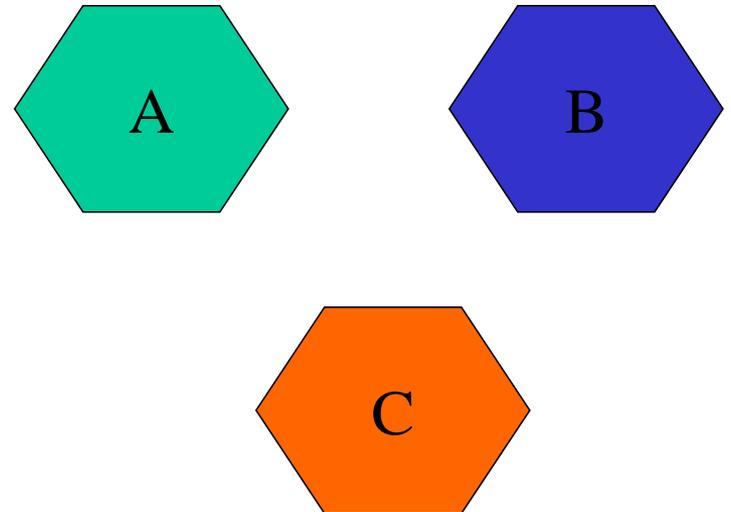
- Dado C o conjunto de objetos da cena
 - Quais deles vão colidir?
- Sendo os objetos O um conjunto complexo de primitivas
 - Quais primitivas vão colidir?
- Dada a colisão
 - Como deverá ser tratada a colisão?



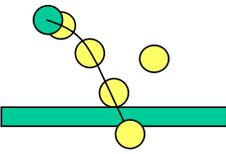
Análise de complexidade

- Fácil de implementar

- testar A com B
- Testar A com C
- Testar B com A
- Testar B com C
- Testar C com A
- Testar C com B

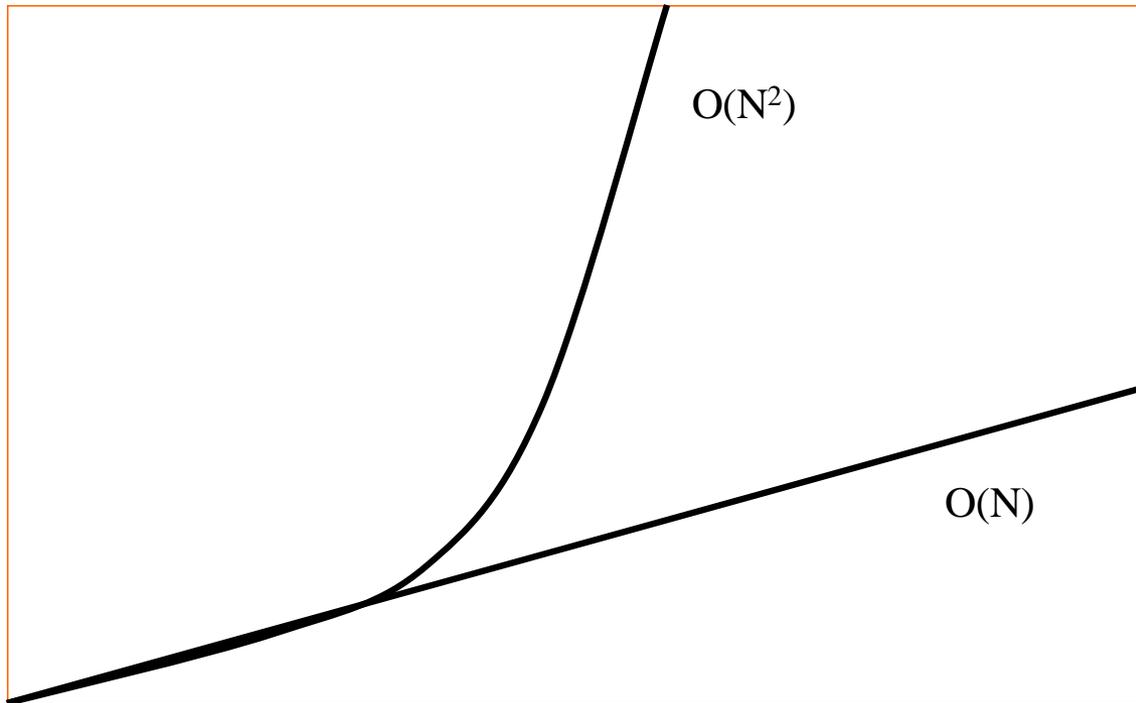


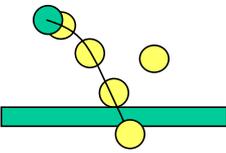
- Complexidade na $O(N^2)$



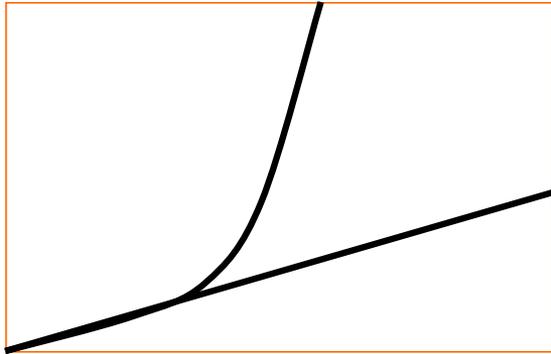
Análise de complexidade

- Muitos objetos, muitas primitivas...



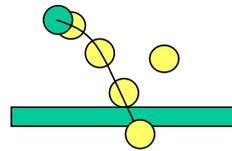


Análise de complexidade

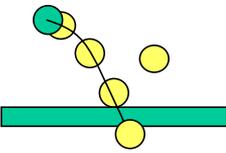


Técnicas avançadas de detecção de colisão devem prover tempo de processamento na $O(n)$ (proporcional ao número de objetos)

Técnicas Básicas de Detecção de Colisão

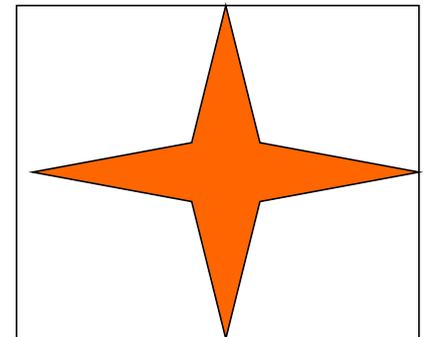
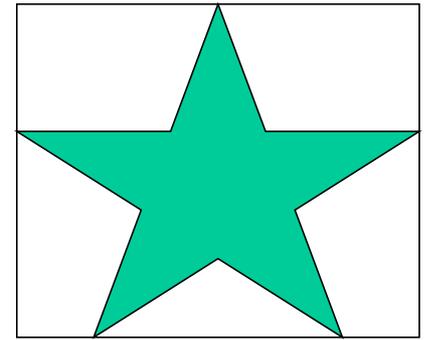


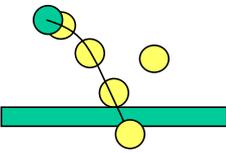
- Bounding volumes
- Animação
- Métodos incrementais



Técnicas Básicas : Bounding Volumes

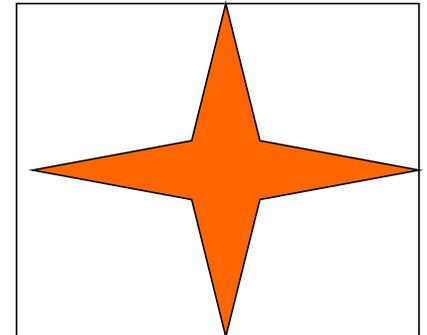
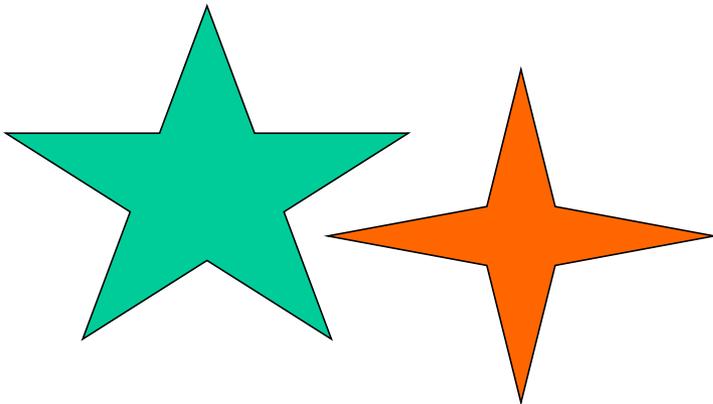
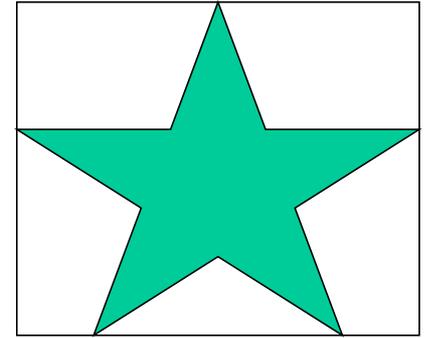
- Maneira eficiente de simplificar o problema da colisão

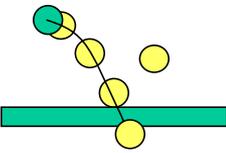




Técnicas Básicas : Bounding Volumes

- Maneira eficiente de simplificar o problema da colisão
- Se os volumes colidem
 - tratar a colisão e/ou a resposta





Técnicas Básicas : Bounding Volumes

- Volumes mais comuns:

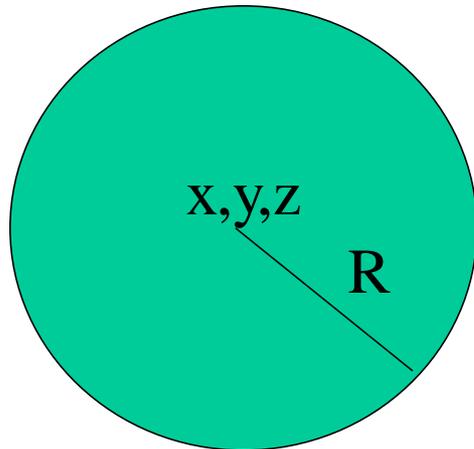
- Box

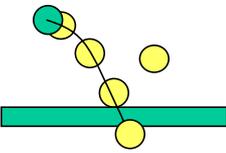
x,y,z

- Esferas



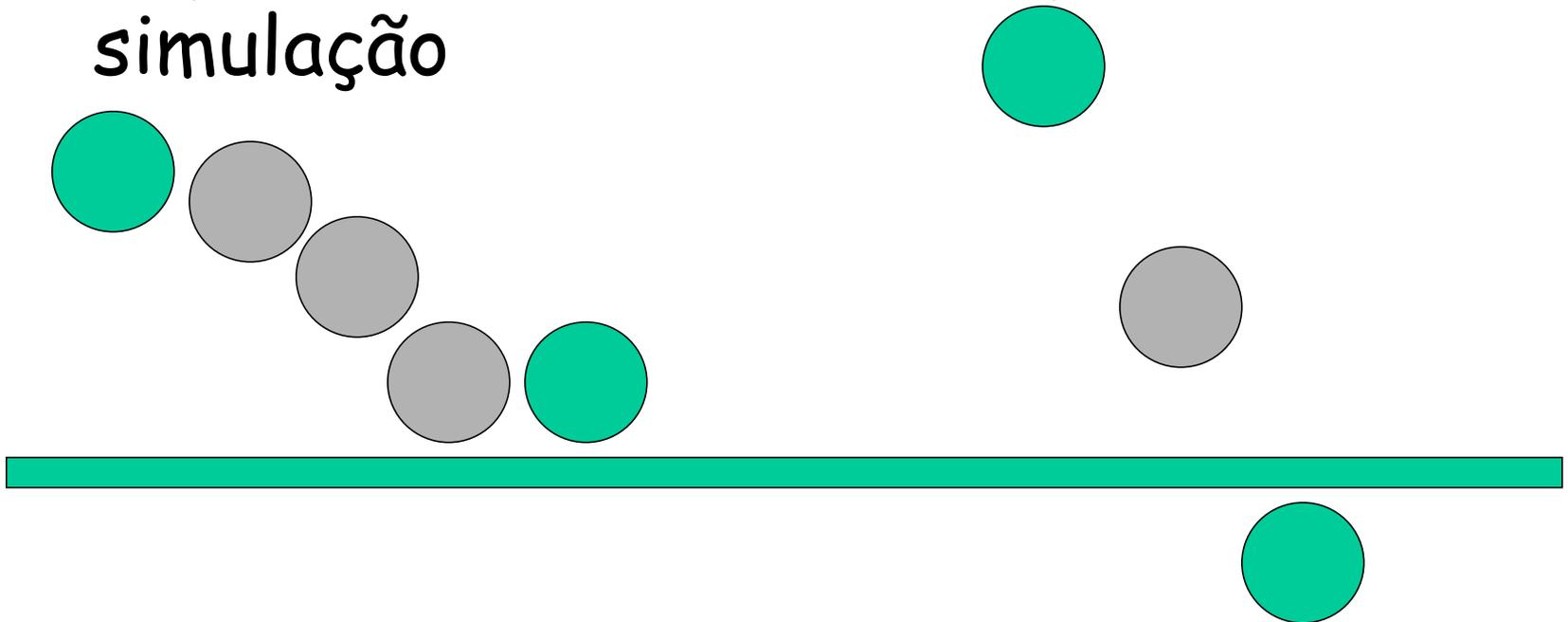
x_m, y_m, z_m

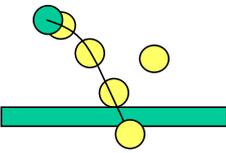




Técnicas Básicas : Animação

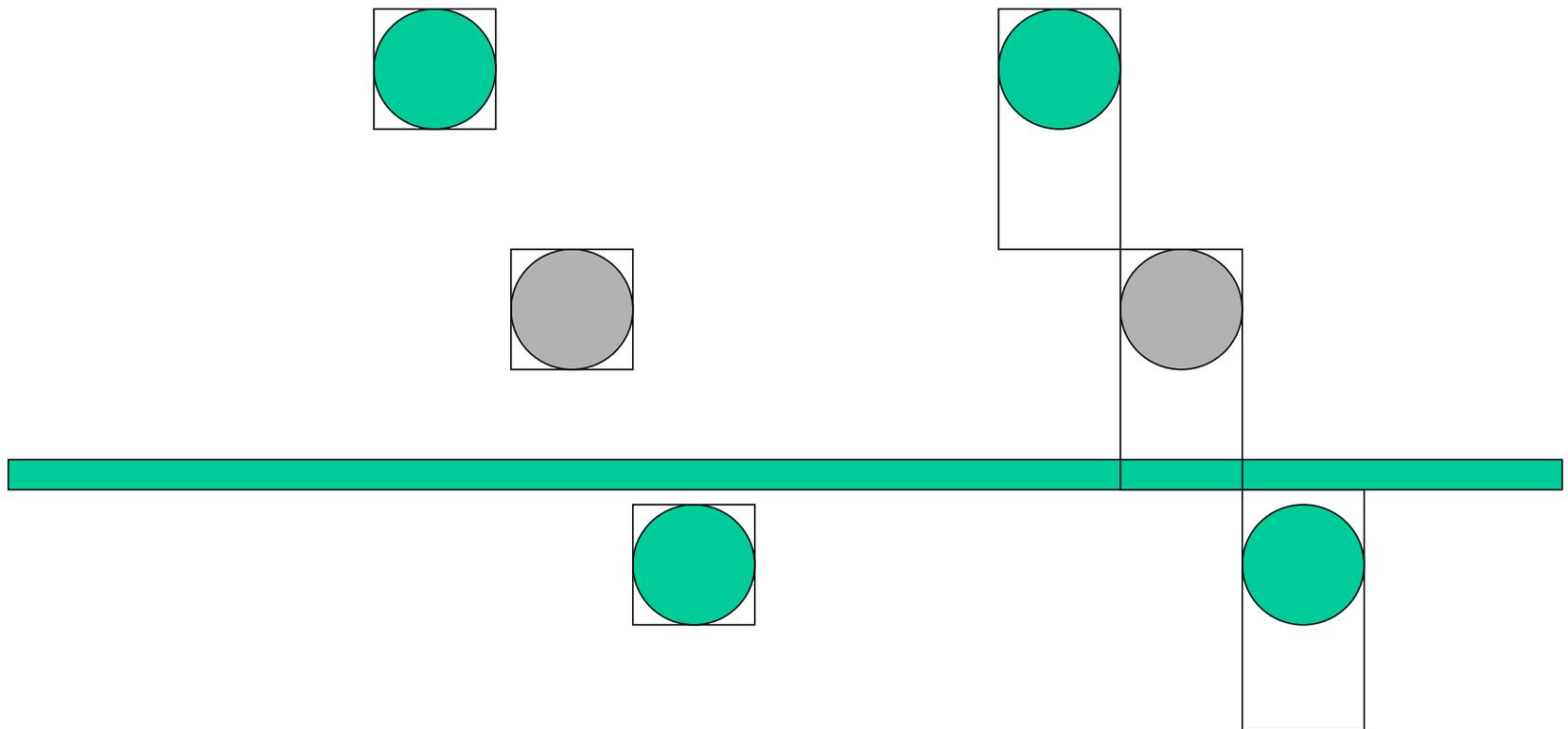
- Momento da colisão pode ser perdido dependendo do delta tempo da simulação

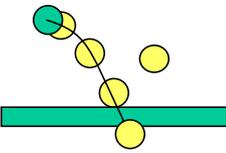




Técnicas Básicas : Animação

- Solução: Time-extended bounding boxes

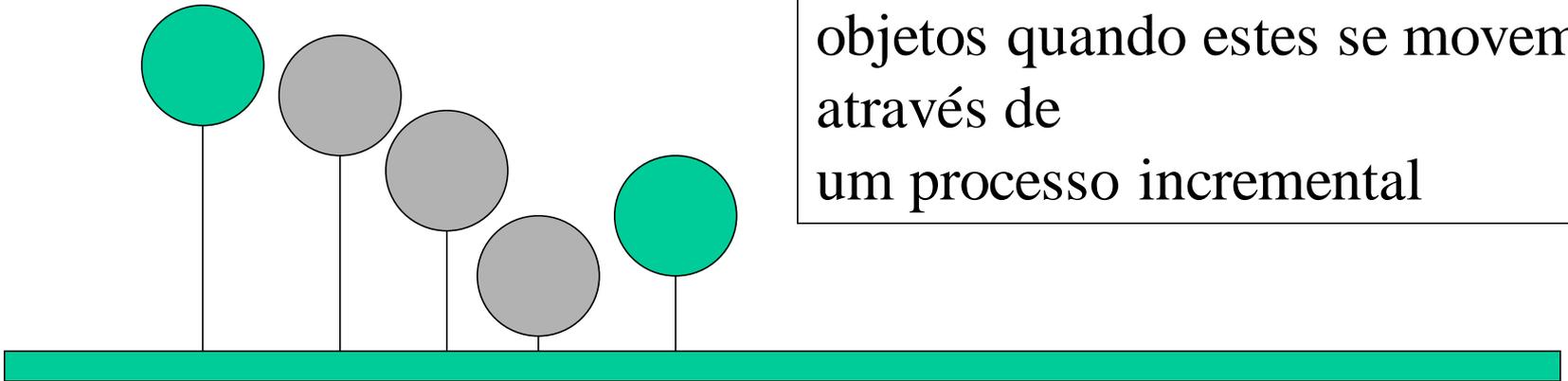


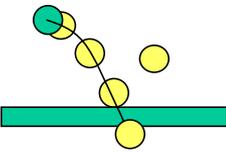


Técnicas Básicas : Incrementais

- Utilizar a semelhança de frames como vantagem para o processamento

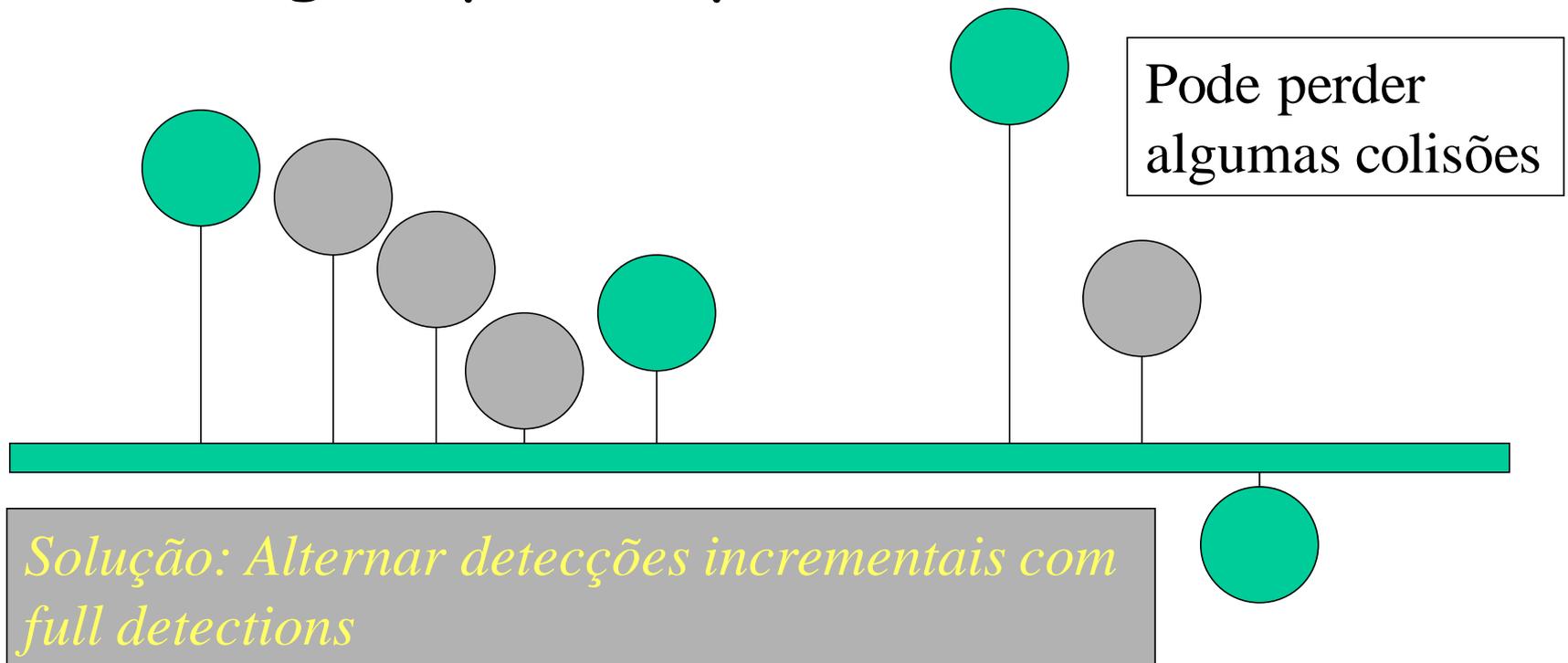
Recalcula as distâncias dos objetos quando estes se movem através de um processo incremental

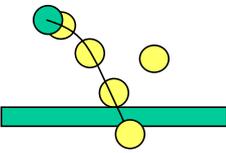




Técnicas Básicas : Incrementais

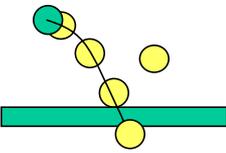
- Utilizar a semelhança de frames como vantagem para o processamento





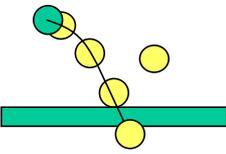
Detecção de colisão entre múltiplos objetos

- Métodos de subdivisão espacial:
 - Voxelização
 - Octree
 - Rasterização
- Métodos Hierárquicos
- Métodos Incrementais

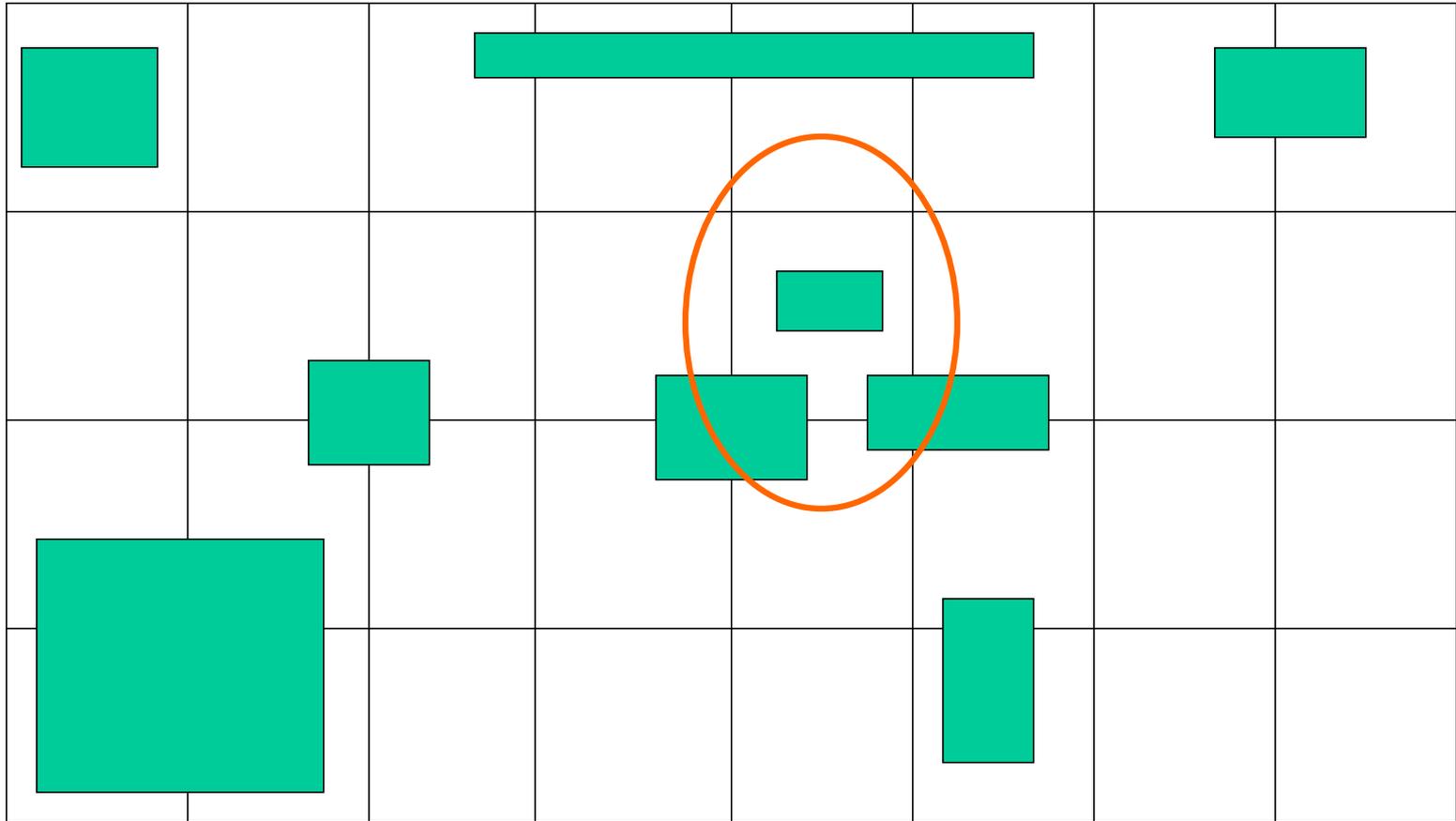


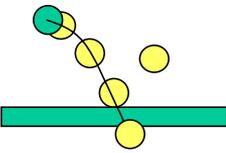
Subdivisão Espacial

- Dividir o espaço em regiões contendo poucos objetos de forma a proceder uma rápida avaliação



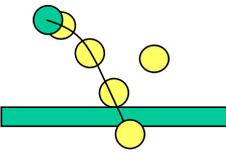
Subdivisão Espacial: Voxel





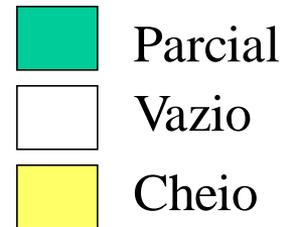
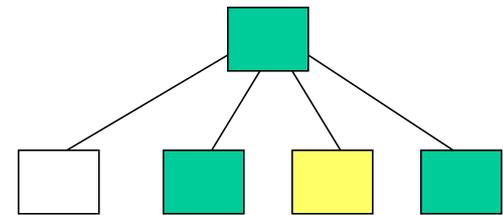
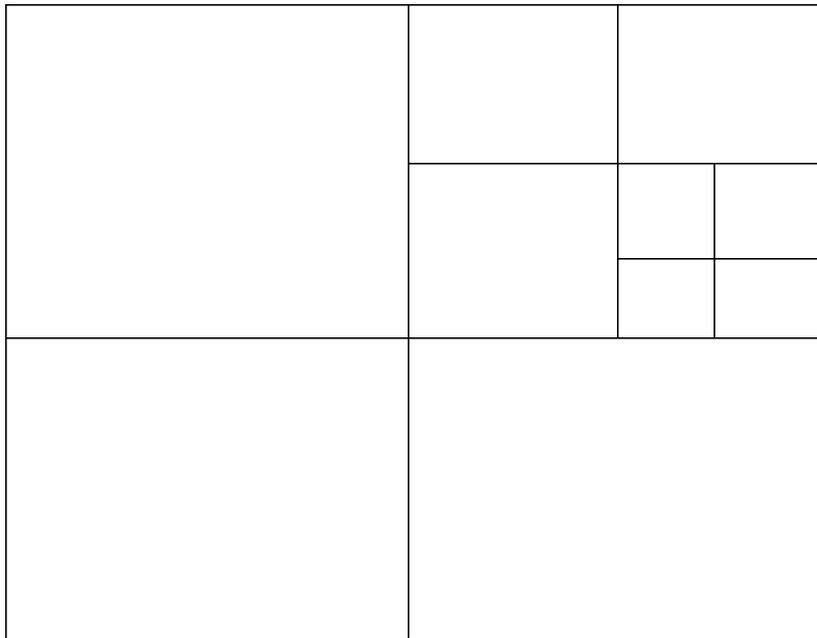
Subdivisão Espacial: Voxel

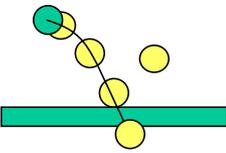
- Estratégia para tamanho da célula:
 - Depende do número de objetos
 - Do tempo de processamento de colisões
 - Da memória disponível



Subdivisão Espacial: Quadtree ou Octree

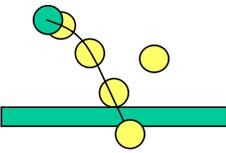
- Subdivisão recursiva do espaço dependendo da sua ocupação





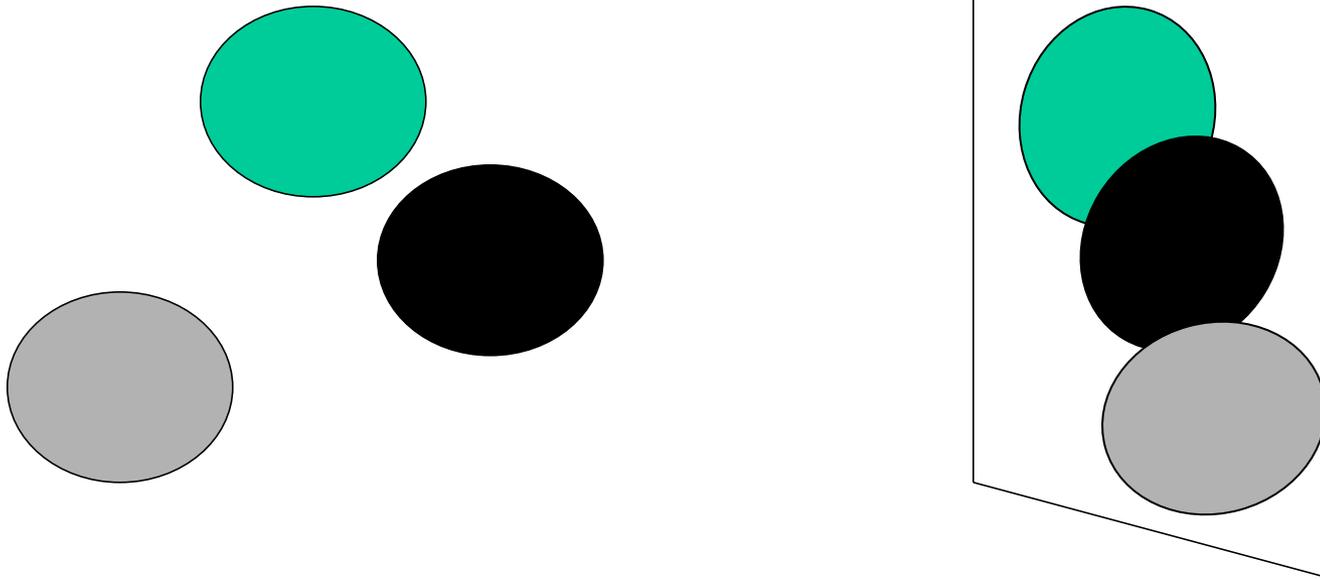
Subdivisão Espacial: Quadtree ou Octree

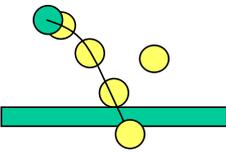
- Reconstruir a quadtree/octree
- Testar as colisões de objetos que compartilham a mesma célula da estrutura
- Mais complexo para implementar
- Resolução adaptativa
- Avaliação multi-colisão



Subdivisão Espacial: Rasterização

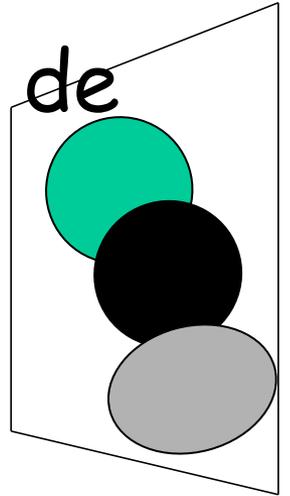
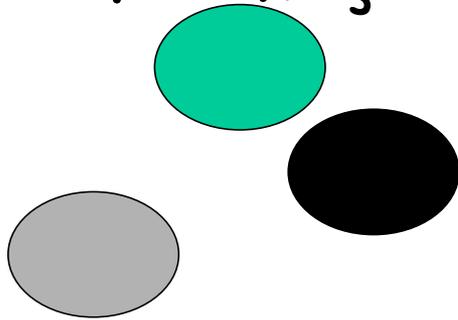
- Projeção de Objetos 3D num frame buffer 2D

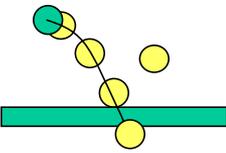




Subdivisão Espacial: Rasterização

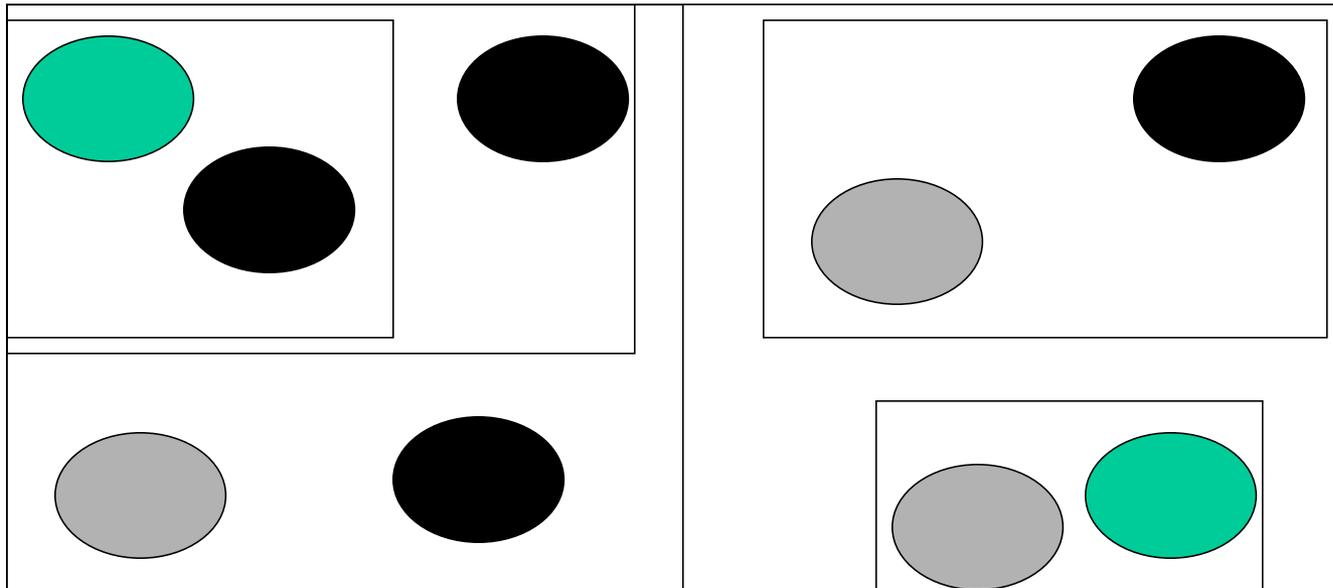
- Projeção de Objetos 3D num frame buffer 2D
- Objetos podem colidir se pixels na projeção se interpenetram
- Z-buffer provê informação extra de colisão

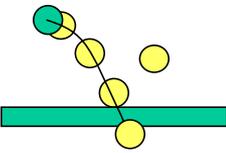




Métodos Hierárquicos

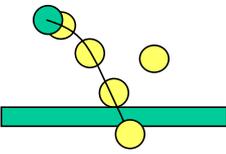
- Objetos são agrupados recursivamente usando um critério de proximidade





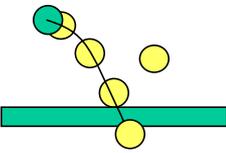
Métodos Hierárquicos

- Objetos são agrupados recursivamente usando um critério de proximidade
- Construir uma árvore com a hierarquia
- Processamento:
 - Processar todos os subnodos de um nodo
 - Verificar colisões entre pares de subnodos



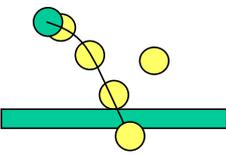
Métodos Hierárquicos

- Enquanto os objetos movem a hierarquia é alterada



Escolhendo o bom método

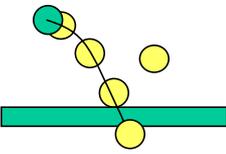
- Critérios:
 - Número de objetos
 - Distribuição dos objetos
 - Forma do objetos
 - Tamanho dos objetos
 - Movimentos dos objetos



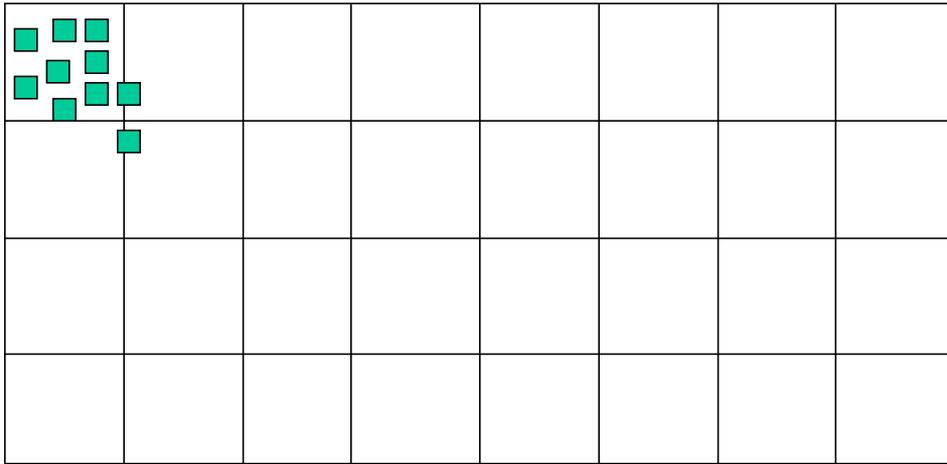
Escolhendo o bom método

Geralmente....

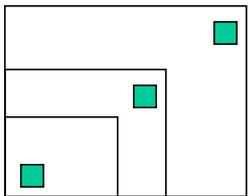
- Múltiplos objetos de tamanho pequeno
 - Voxelização
- Objetos com maior volume
 - Octree
- Animações
 - Métodos hierárquicos



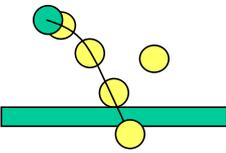
Piores casos



Voxel

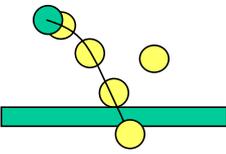


Hierarquia

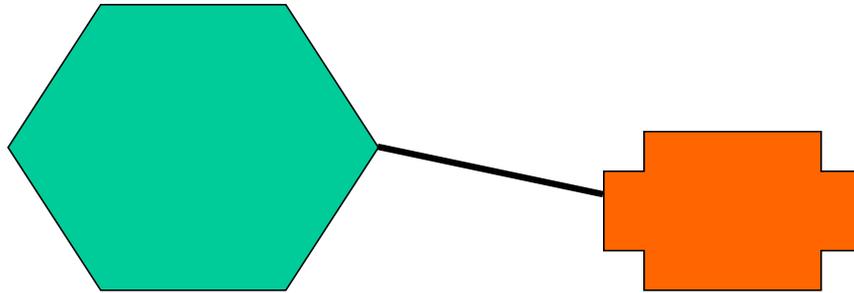


Performance

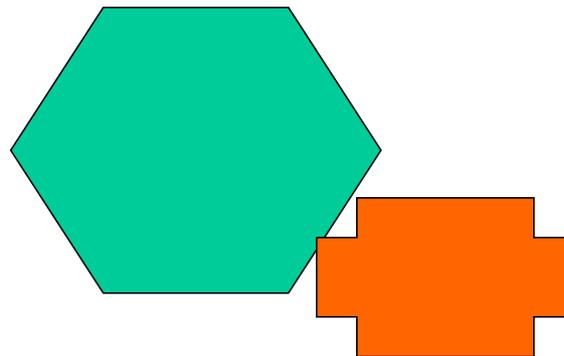
- Average time for N objetos, n colisões
 - Algoritmos simples $O(N^2)$
 - Full detection $O(N \log N)$
 - Método incremental $O(N)$



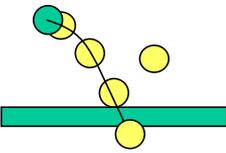
Tipos de colisão



Proximidade

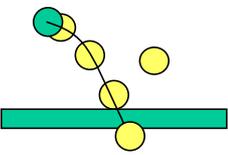


Intersecção



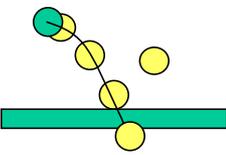
Resposta à colisão

- Robótica, CAD, Motion planning
 - Evitar contato entre objetos e/ou manter uma distância mínima dos obstáculos
- Contato
 - Evitar situação inconsistente
- Proximidade
 - Verificar mínima distância



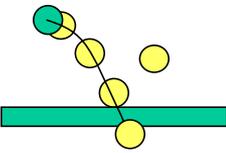
Resposta à colisão

- Simulação mecânica
 - A colisão deve acontecer e ser tratada de maneira a repetir a colisão na vida real
 - Normalmente tratada como uma descontinuidade no movimento
- Forças de reação
 - geradas durante a colisão



Resposta à colisão

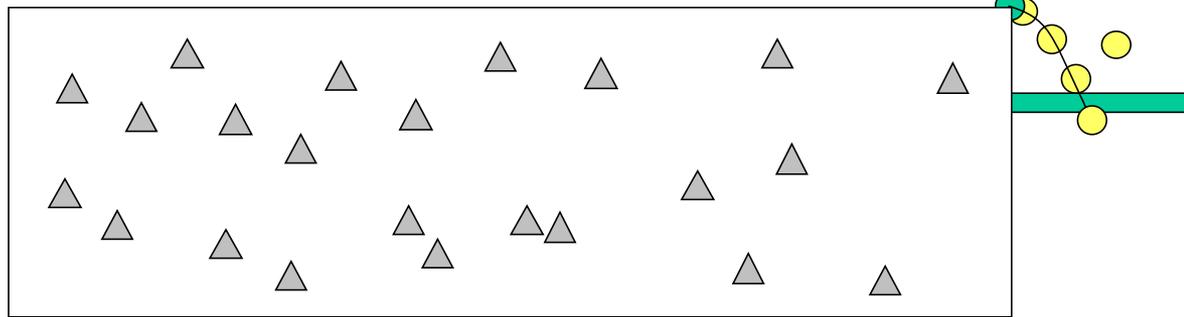
- Resposta depende da aplicação
- Tem que refletir corretamente uma situação lógica ou numérica
- Método tem que ser robusto para manipular várias colisões



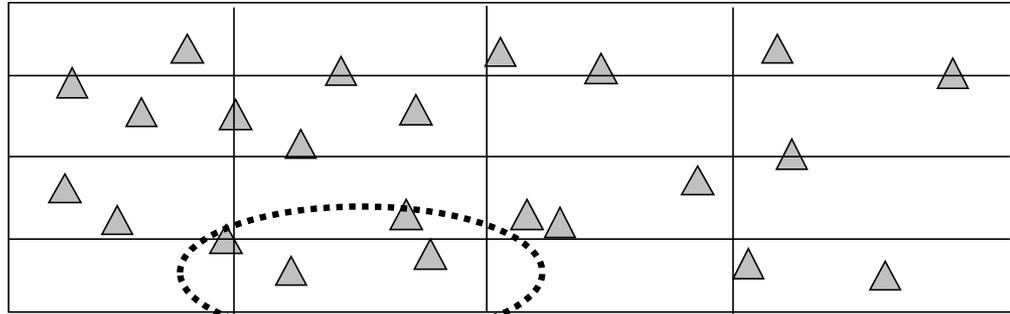
Resposta à colisão

- Quando a resposta deve ser evitar colisão?
- E quando a colisão deve ser evitada durante uma trajetória?
- E quando tem múltiplas entidades?

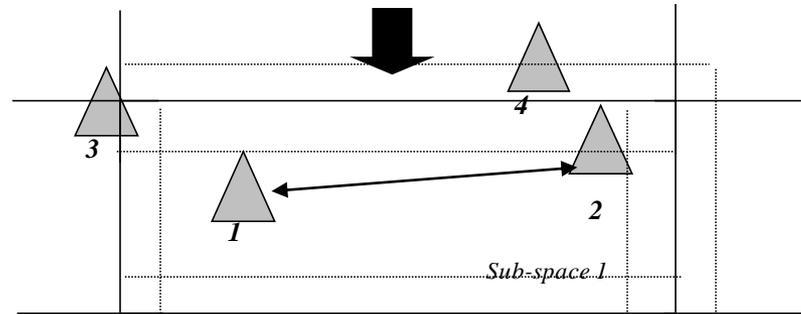
a) População distribuída no universo



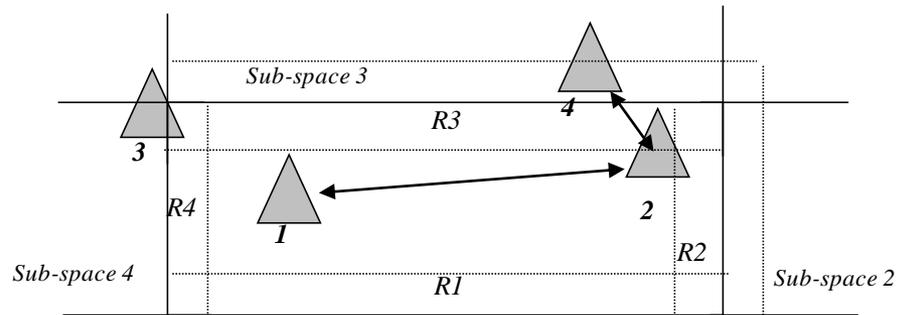
b) Universo subdividido em células (voxelização)

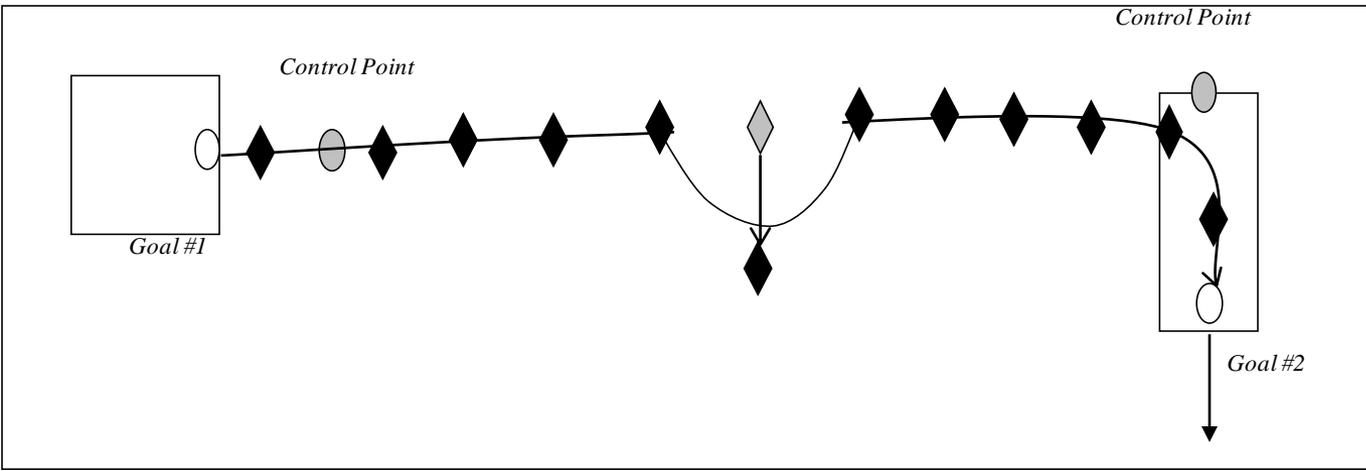
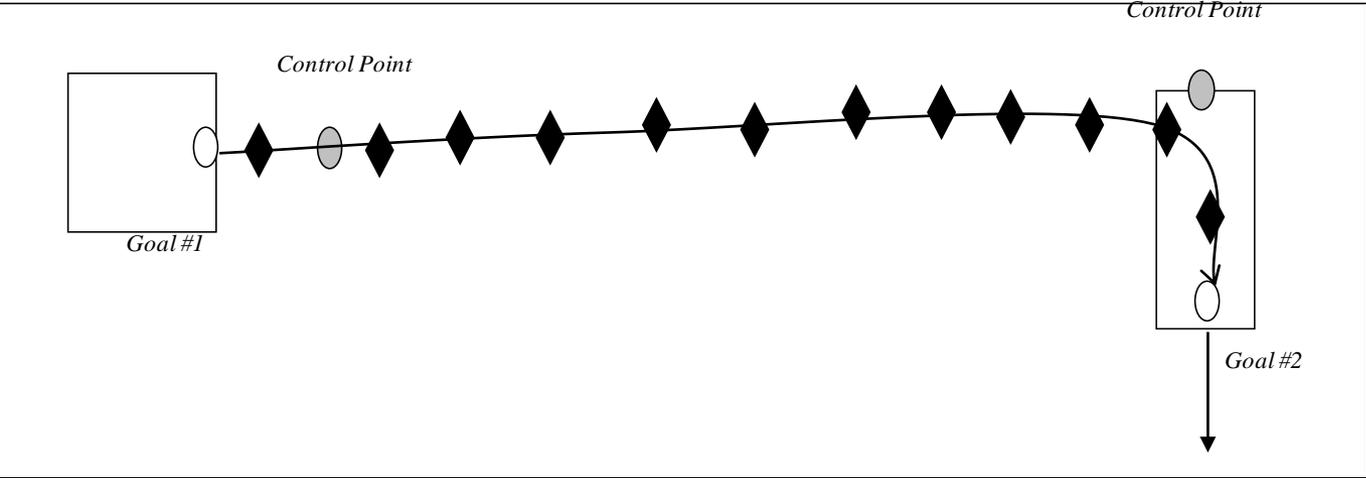
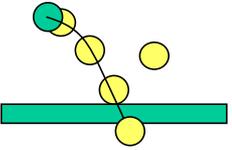


c) Sub-espacos de tamanho extrapolado

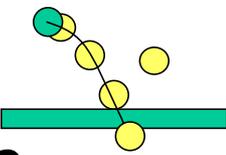


d) Colisão detectadas entre objetos de mesmo sub-espaco

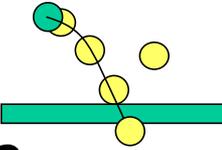




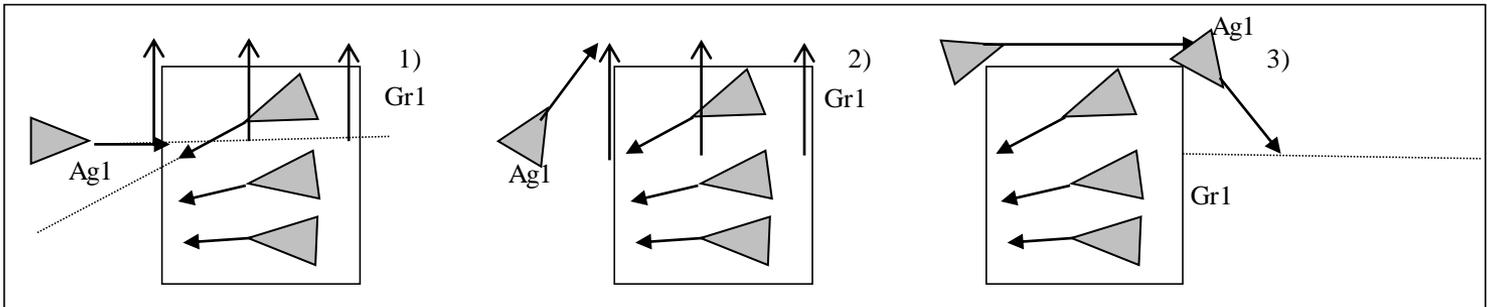
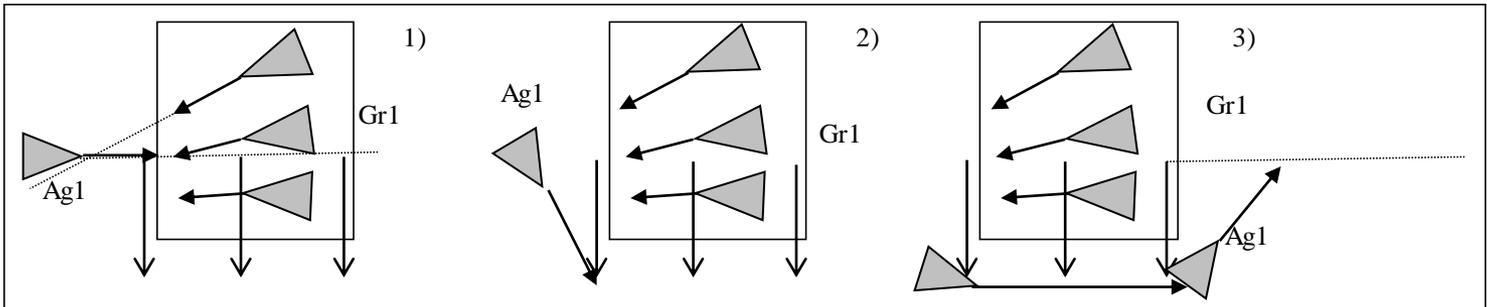
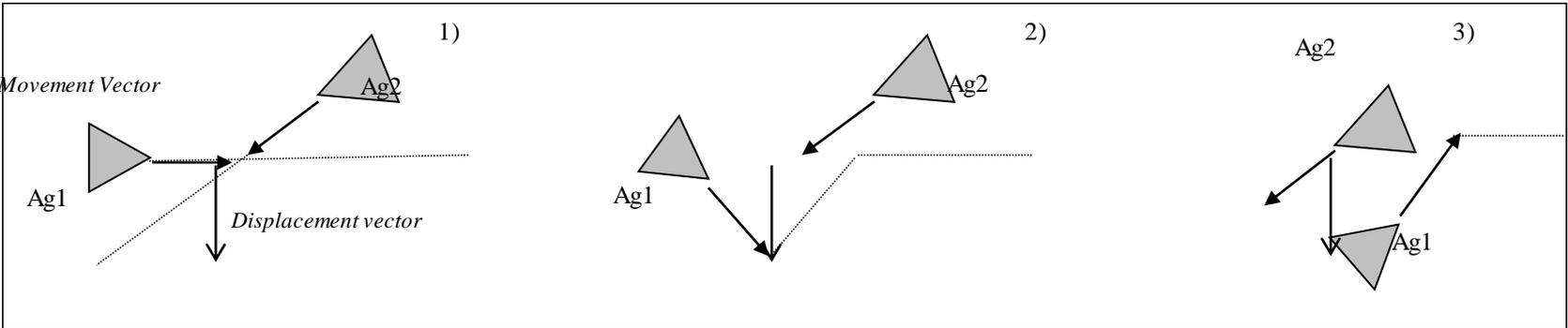
Estratégia para saber quem deve evitar?

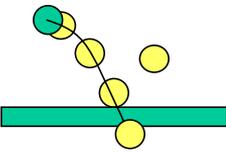


- Dependente da densidade das entidades envolvidas na situação (prioridade 1)
- Dependente da velocidade das entidades (prioridade 2)
- Se uma entidade é mais rápida e a outra é menos densa?
- Se duas entidades são igualmente rápidas e estão em espaços igualmente densos?



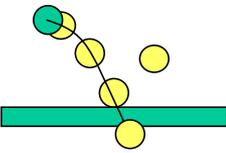
Estratégia para saber quem deve evitar?



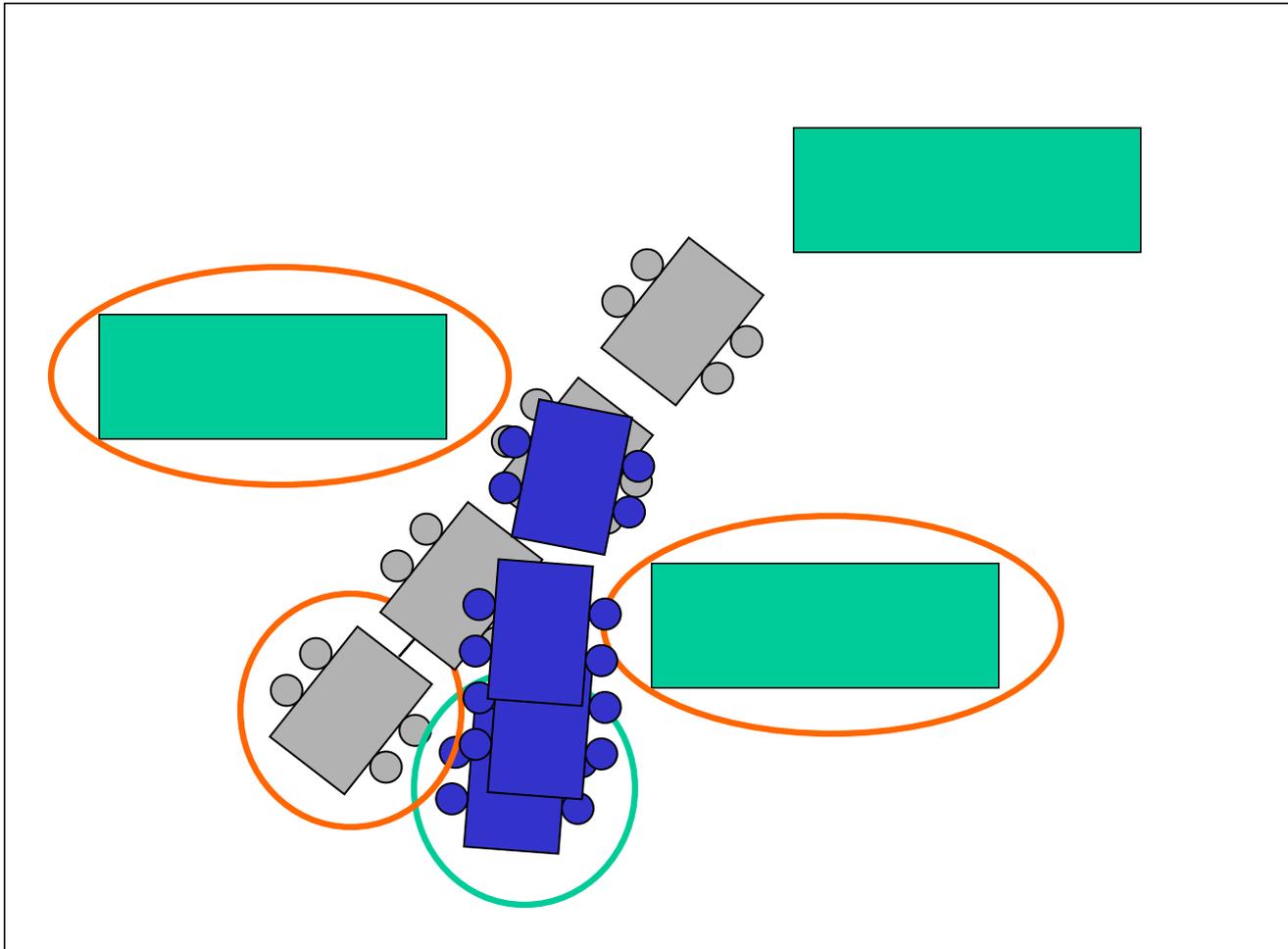


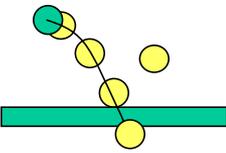
Métodos para evitar colisão

- Métodos globais (física das partículas, procedural etc)
- Métodos locais (física das partículas, procedural etc)
- Force fields

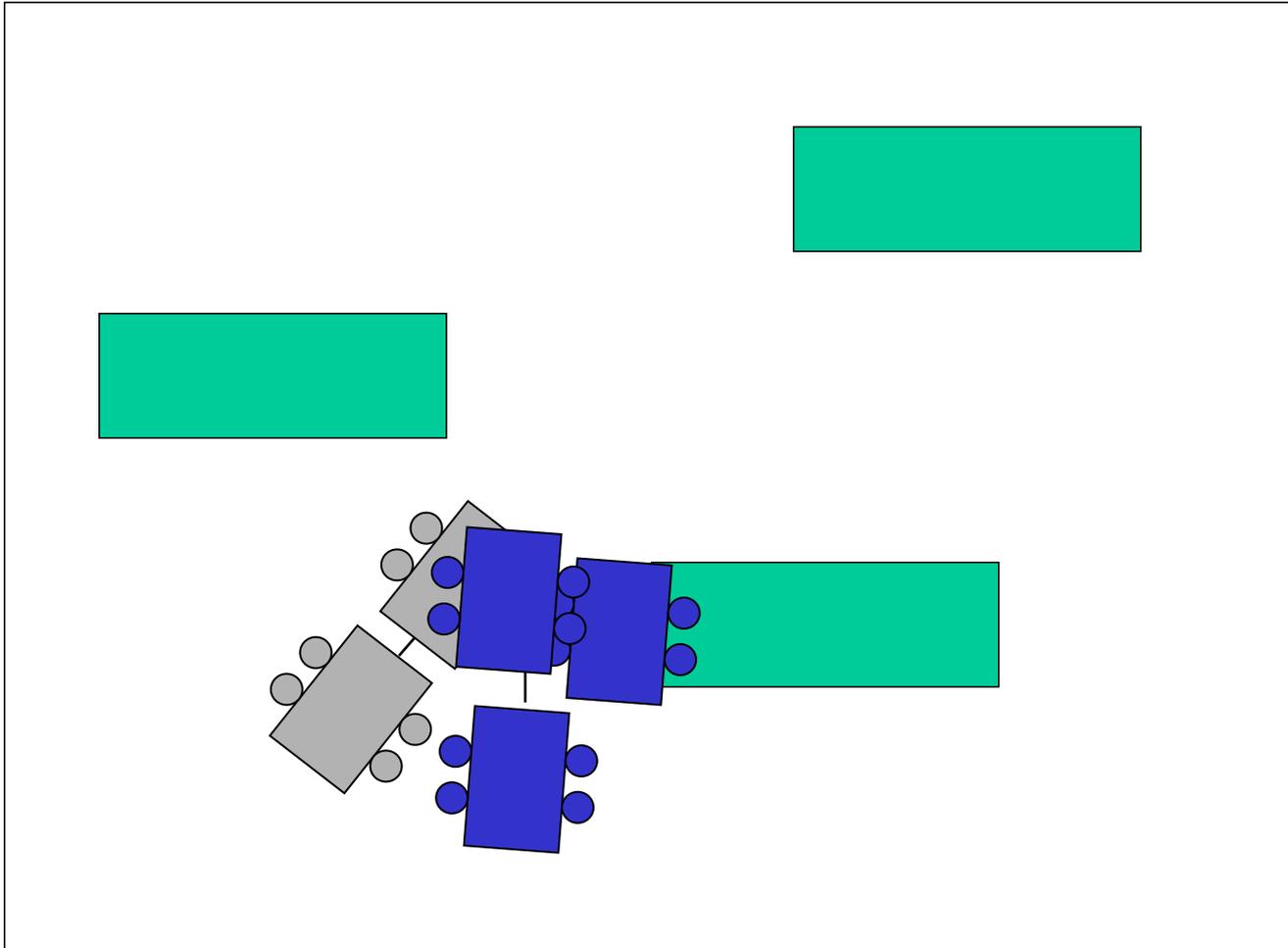


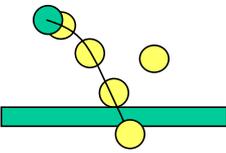
Métodos Globais



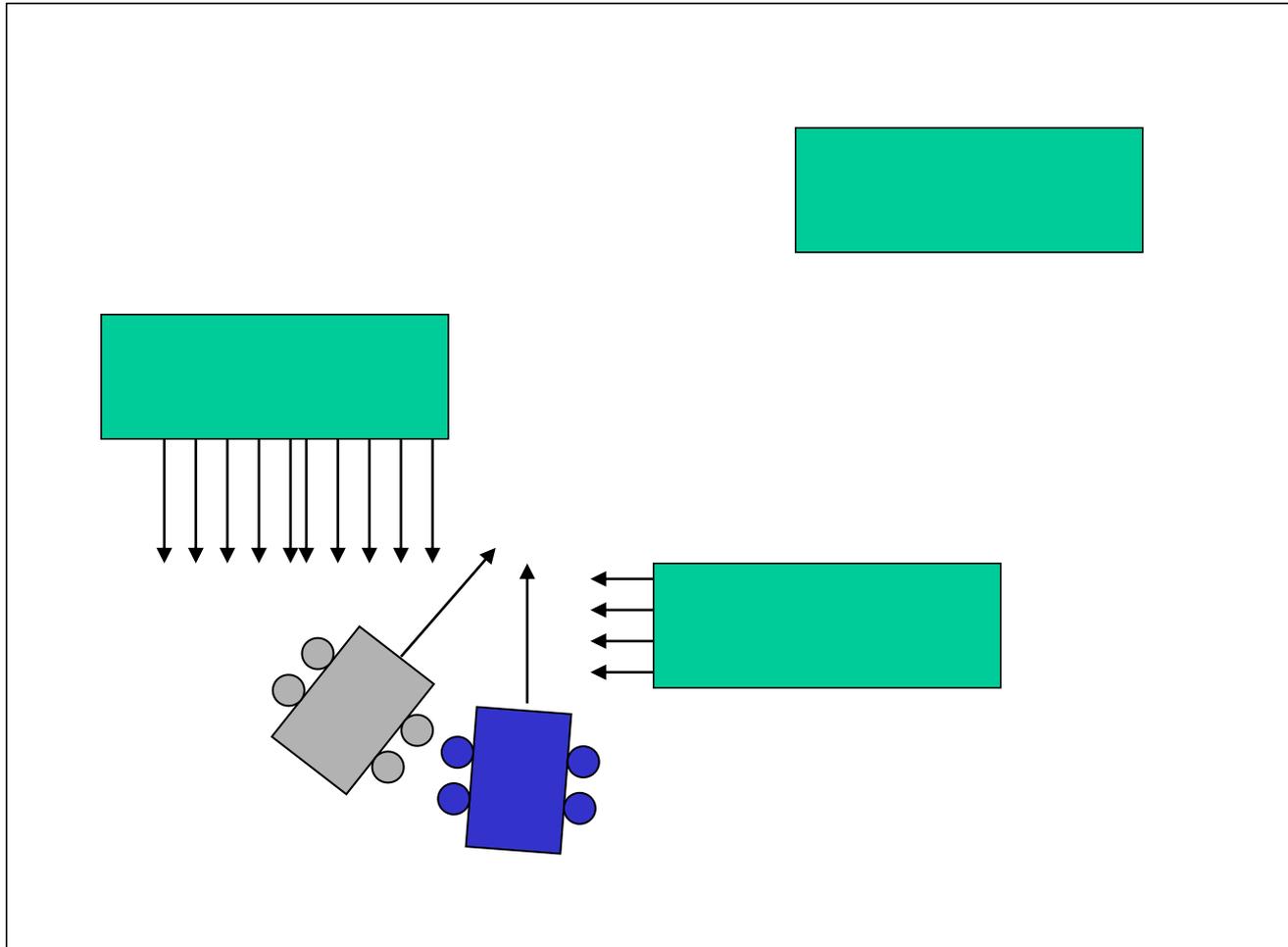


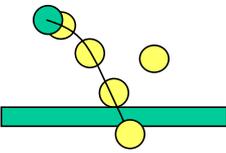
Métodos Locais



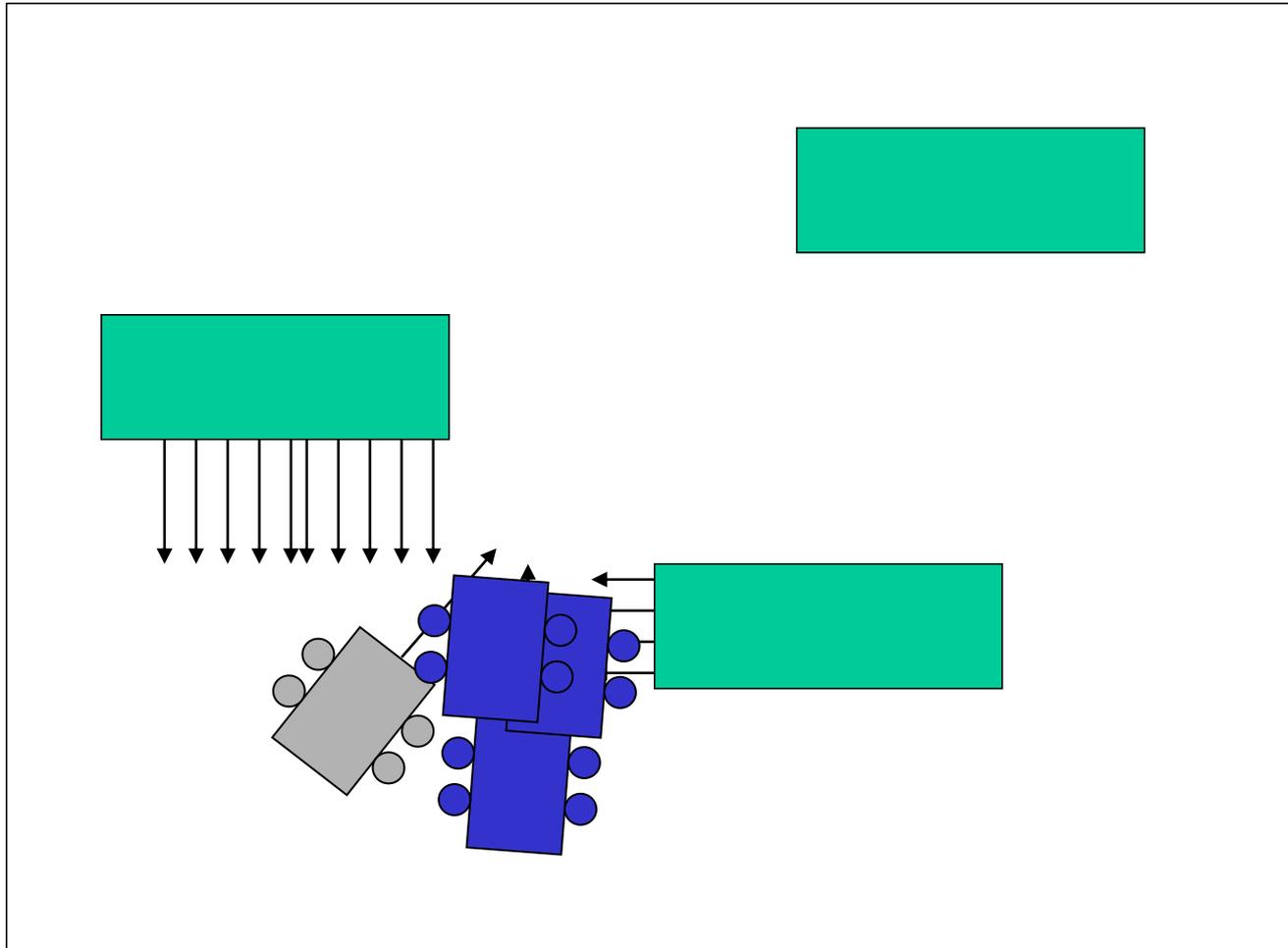


Force fields

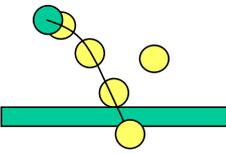




Force fields



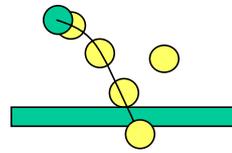
Instabilidade



O problema

- Método de colisão eficiente/robusto em multi-escala e em tempo real
 - Que trate objetos móveis
 - Que trate objetos imóveis
 - Que mantenha trajetórias coerentes e livres de colisão
 - Que se aproxime corretamente de objetos levando em consideração bounding box e as vezes primitivas

No escopo do trabalho de você



- Usar proximidade, localmente aos objetos
- Permitir o contato ($dist=0$) e calcular resposta a colisão
- Resposta deve ser geração de vetor de movimento para os objetos