

# Biblioteca para Explorar Algoritmos de Planejamento de Movimento Utilizando Campos Potenciais no SDK do iOS 4

Marcelo Roberto Ferrari – Acadêmico  
Prof. Dalton Solano dos Reis - Orientador



# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Desenvolvimento
- Resultados e discussão
- Conclusão, Extensões e Demonstração



# Introdução

- Planejar o movimento de um agente autônomo é basicamente planejar um caminho contínuo e livre de colisões com obstáculos, a partir de uma configuração inicial, para chegar até um objetivo
- Caminhos alternativos para agentes autônomos através de campos potenciais
- Existem várias maneiras de resolver o problema de mínimos locais. Uma delas é a utilização de Problema de Valor de Contorno



# Objetivos

- Desenvolver uma biblioteca para explorar algoritmos de planejamento de movimento para o SDK da plataforma iOS 4
- Desenvolver um algoritmo para planejamento de movimentos de agentes autônomos utilizando a técnica de campos potenciais
- Eliminar o problema de mínimos locais através de Problemas de valor de contorno



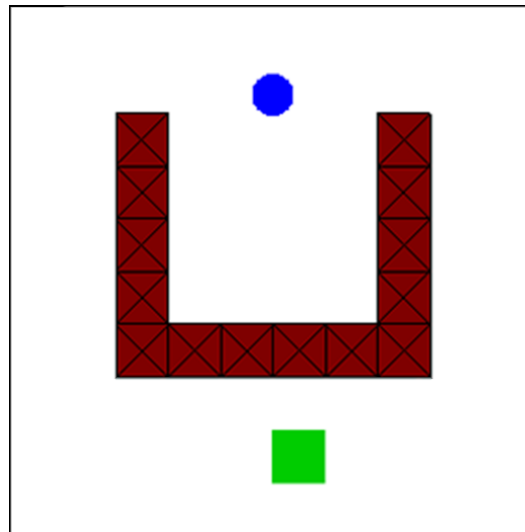
# Fundamentação teórica

- Planejamento de movimento
  - O planejamento de movimento consiste em encontrar um caminho livre de colisões com obstáculos, para movimentar um robô rígido ou articulado entre uma posição inicial e uma final
  - Principais métodos:
    - planejamento baseado em *scripts*
    - planejamento baseado em grades de células
    - planejamento baseado em *roadmaps*
    - planejamento baseado em campos potenciais



# Fundamentação teórica

- Campos potenciais
  - Khatib (1980) propôs utilizar campos potenciais para planejar caminhos.
  - Nesta técnica é gerado um campo potencial, que é criado a partir de forças repulsivas, geradas pelos obstáculos, e por forças atrativas, geradas pelos objetivos
  - A utilização desta técnica pode levar a mínimos locais



# Fundamentação teórica

- Problema de valor de contorno
  - São equações diferenciais ordinárias ou parciais
  - Utilizam funções harmônicas que satisfazem a equação de Laplace

$$\nabla^2 p(r) = \sum_i \frac{\partial^2 p(r)}{\partial x_i^2} = 0$$

- Divisão do cenário em células de mesmo tamanho
- Cada célula (i, j) recebe um valor potencial



# Fundamentação teórica

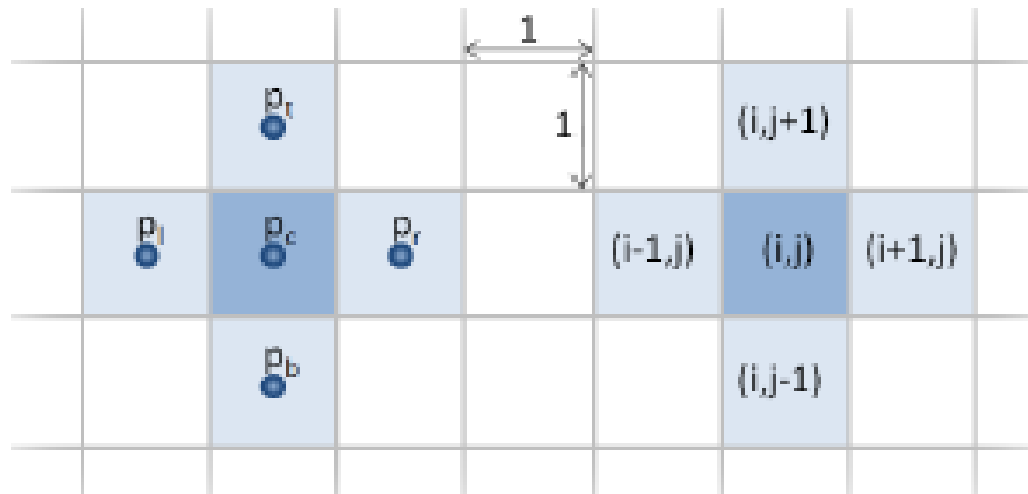
- Problema de valor de contorno

- Método de Gauss-Seidel

- Consiste em substituir o valor do potencial de cada célula livre pela média simples de seus vizinhos simultaneamente

$$p_c = \frac{p_t + p_b + p_l + p_r}{4} + \frac{\epsilon}{8} \left( (p_r - p_l)v_x + (p_b - p_t)v_y \right)$$

onde  $p_c = p_{i,j}^{t+1}$ ,  $p_t = p_{i,j-1}^{t+1}$ ,  $p_b = p_{i,j+1}^t$ ,  $p_l = p_{i-1,j}^{t+1}$ ,  $p_r = p_{i+1,j}^t$ ,  $v = (v_x, v_y)$ , sendo  $v$  um fator de ajuste e  $\epsilon \in (-2, 2)$ .



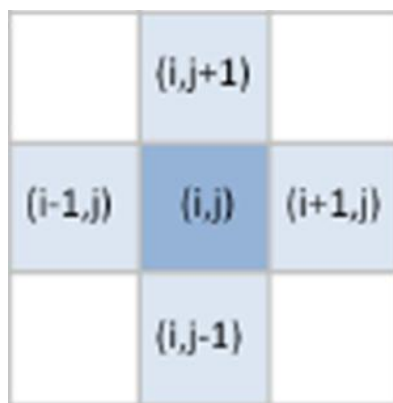


# Fundamentação teórica

- Movimentação do agente
  - É definida através do gradiente descendente da posição atual

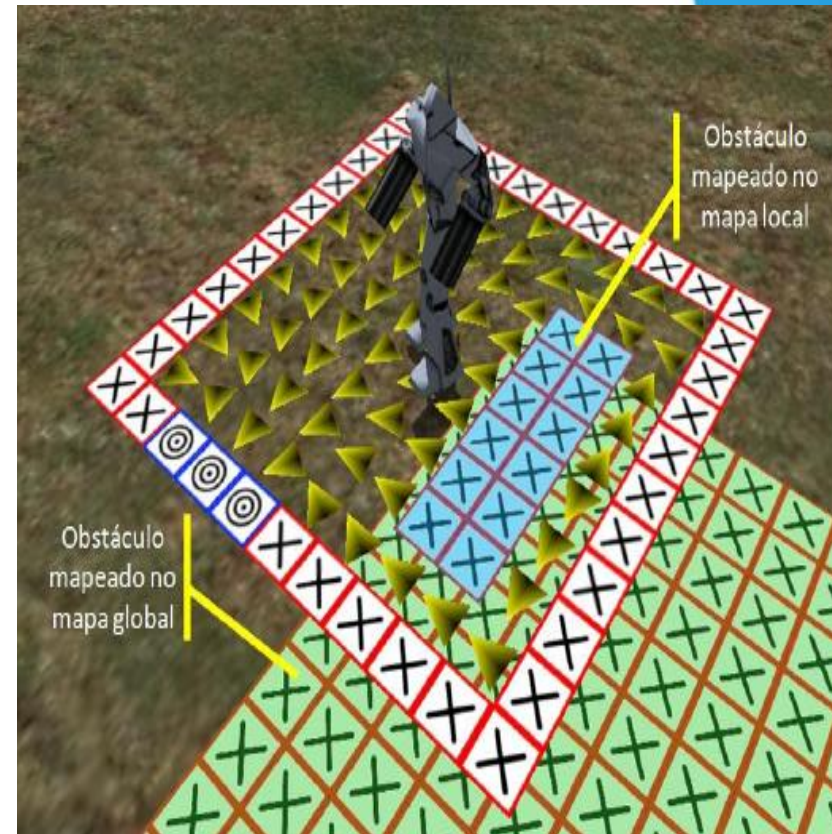
$$(\nabla p)_{(i,j)} = \left( \frac{p_{i+1,j} - p_{i-1,j}}{2}, \frac{p_{i,j+1} - p_{i,j-1}}{2} \right)$$

sendo o tamanho da discretização 1 unidade



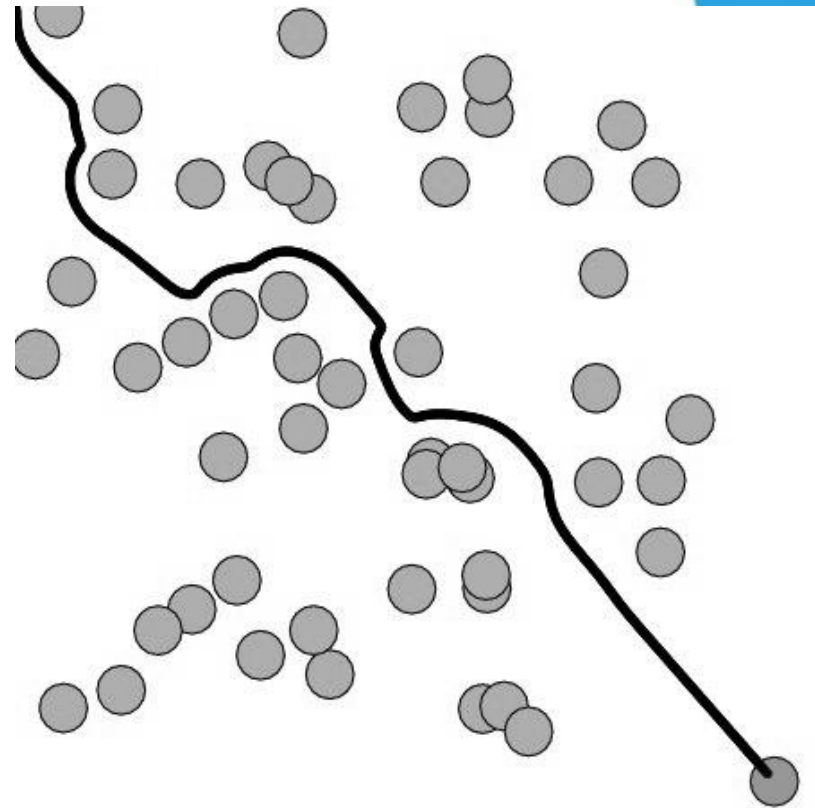
# Fundamentação teórica

- Trabalhos correlatos
  - Fischer (2008)
    - Planeja movimento de vários agentes
    - Possui vários objetivos sequenciais
    - Mapa global e local
    - Evita mínimos locais utilizando o método de Gauss-Seidel



# Fundamentação teórica

- Trabalhos correlatos
  - Ferrari (2009)
    - Planeja movimento de um agente
    - Possui um objetivo
    - Evita mínimos locais utilizando campos potenciais parametrizáveis

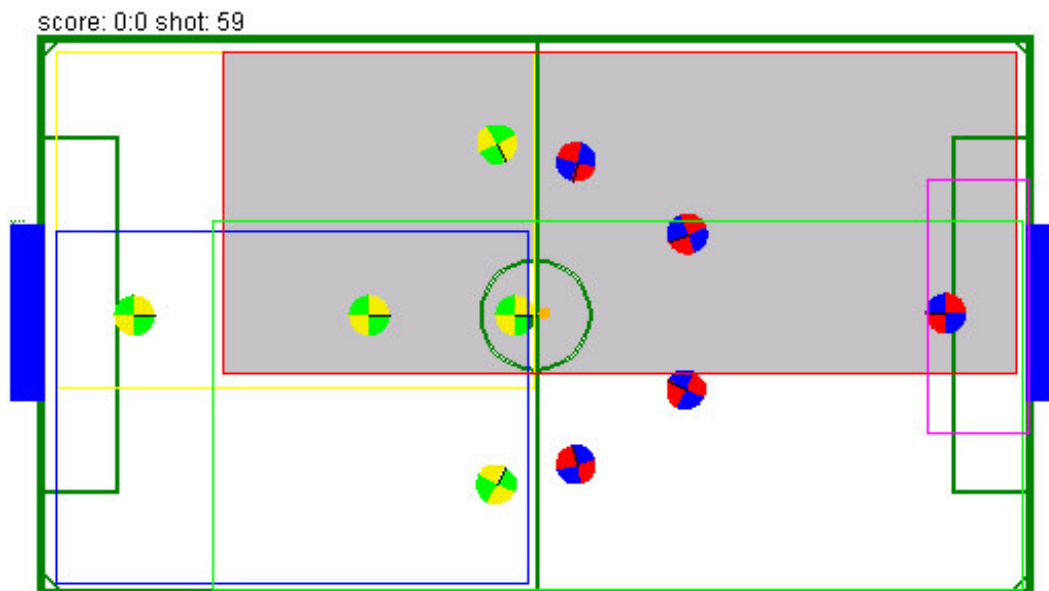


# Fundamentação teórica

- Trabalhos correlatos

- Mafra (2004)

- Protótipo de times de futebol de robôs
    - Possui um objetivo
    - Campo potencial gerado a partir da área de atuação do robô dentro do campo



# Desenvolvimento

- Requisitos funcionais
  - permitir que seja informado um ponto de saída
  - permitir que sejam informados um ou mais pontos de chegada
  - disponibilizar um aplicativo exemplo para permitir a execução de testes na biblioteca
  - permitir que sejam visualizados os mapas virtuais e a movimentação dos agentes autônomos



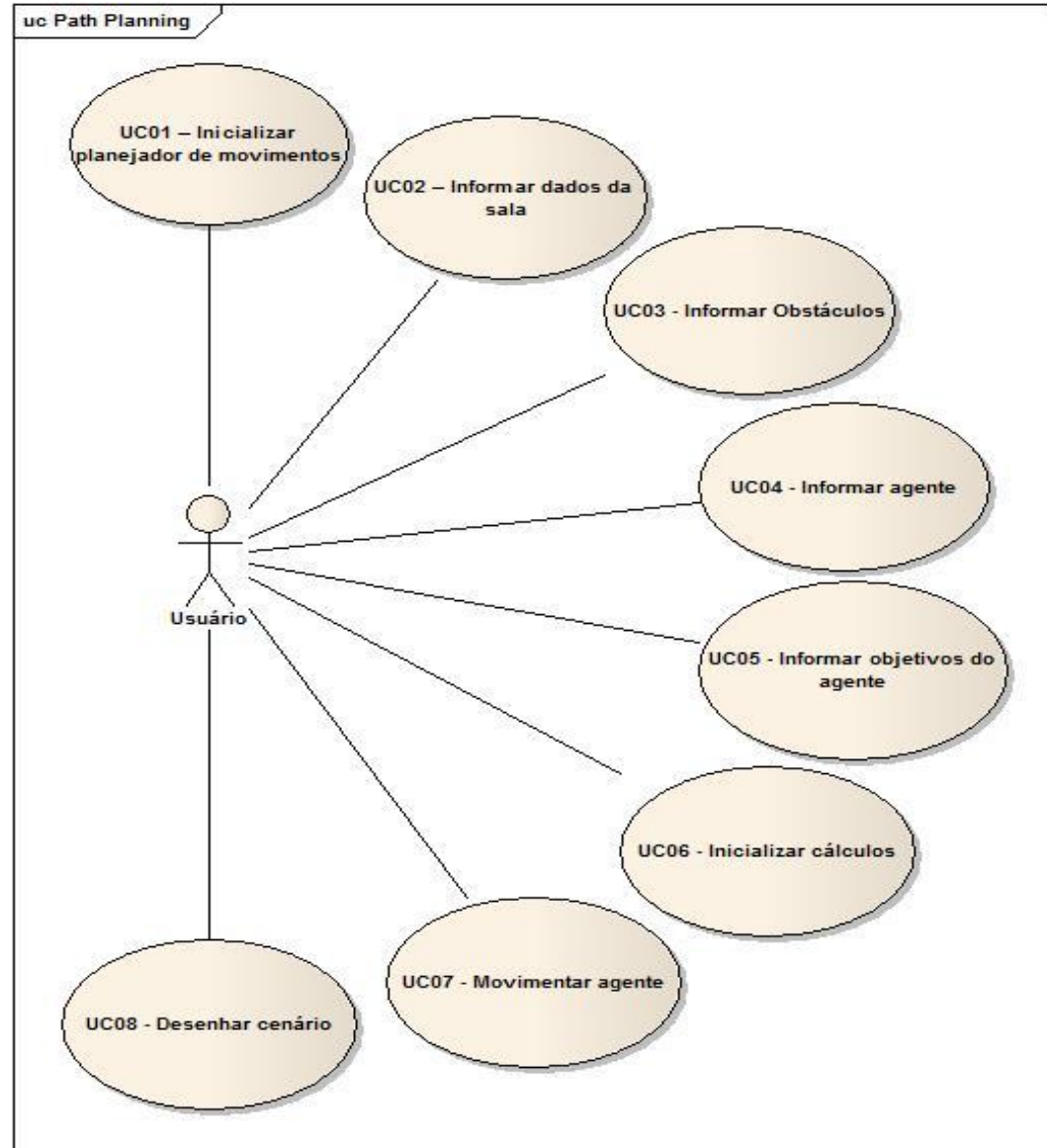
# Desenvolvimento

- Requisitos não-funcionais
  - ser implementado utilizando o ambiente de desenvolvimento Xcode
  - ser implementado utilizando a linguagem Objective-C
  - ser compatível com a plataforma móvel iOS 4
  - possuir documentação
  - ser implementado utilizando a técnica de campos potenciais
  - deverá evitar o problema de mínimos locais através da utilização da técnica de PVC



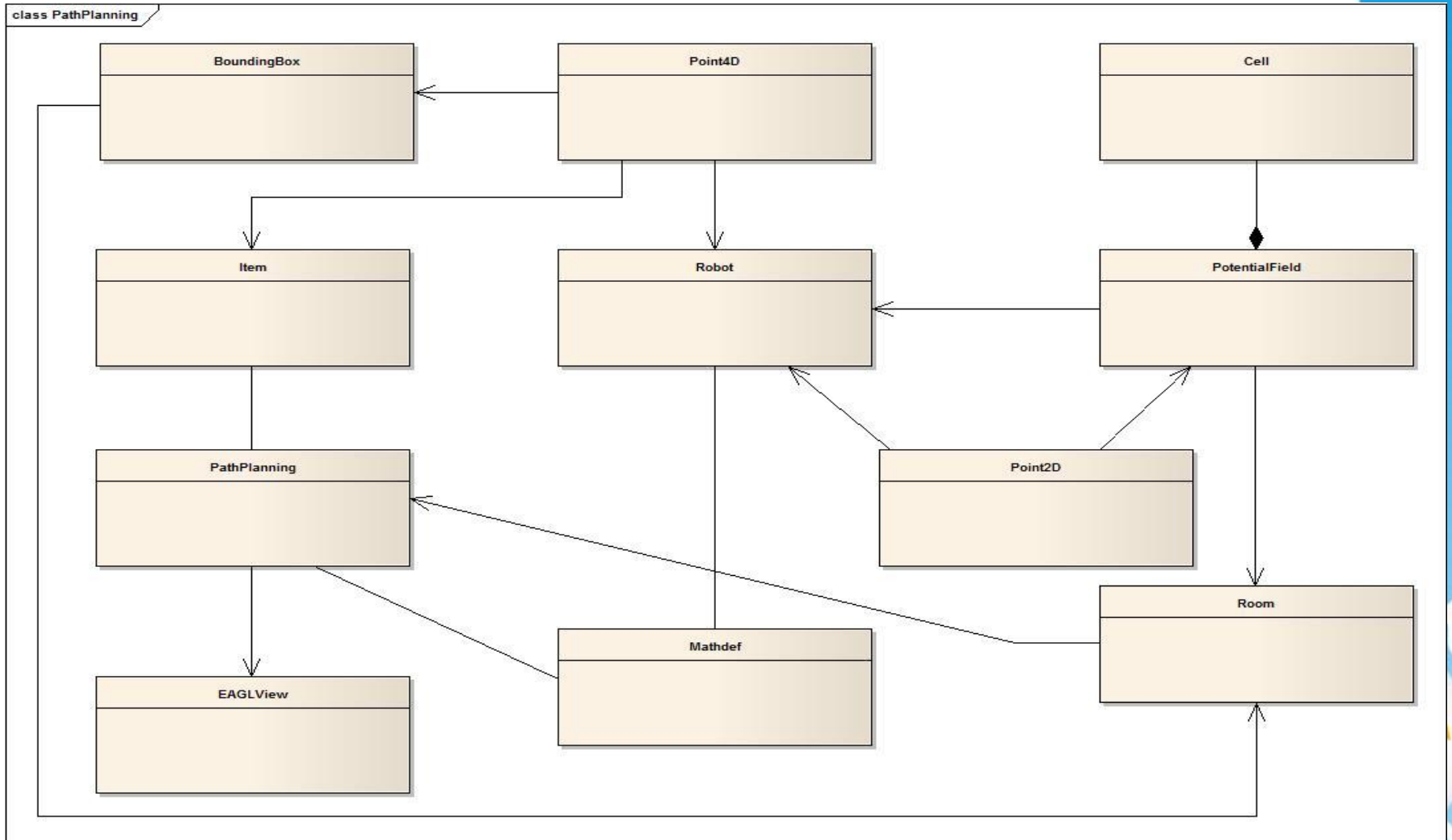
# Desenvolvimento

- Casos de uso



# Desenvolvimento

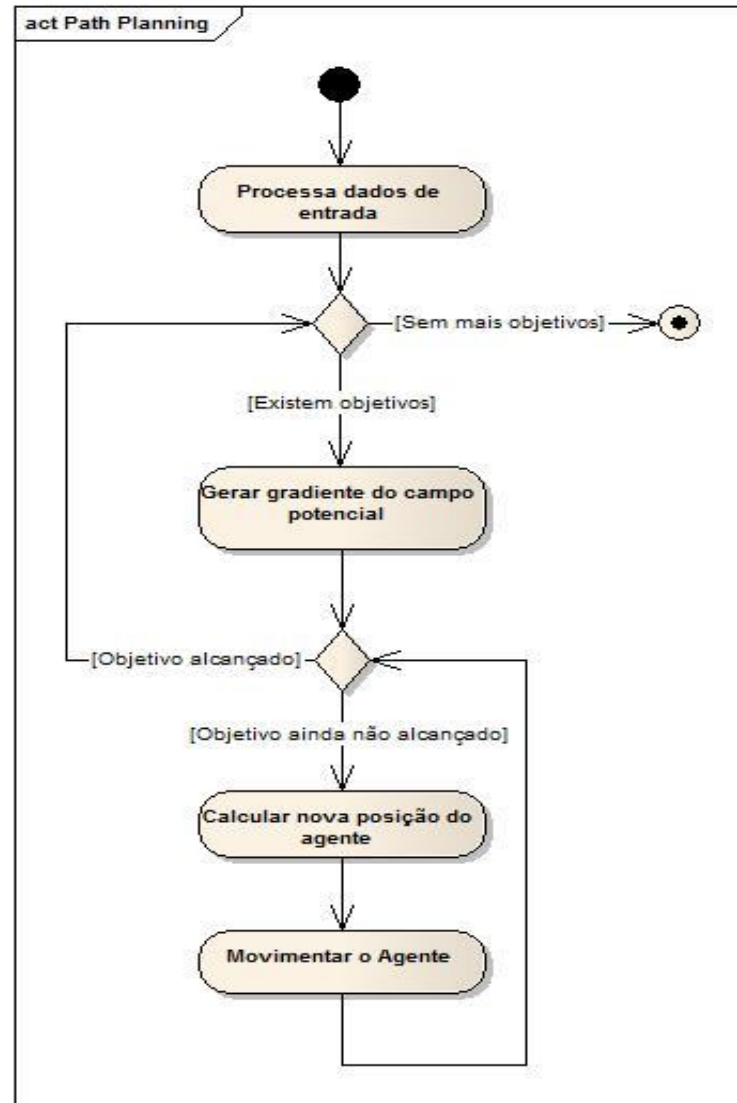
- Diagrama de classes





# Desenvolvimento

- Diagrama de atividades



# Desenvolvimento

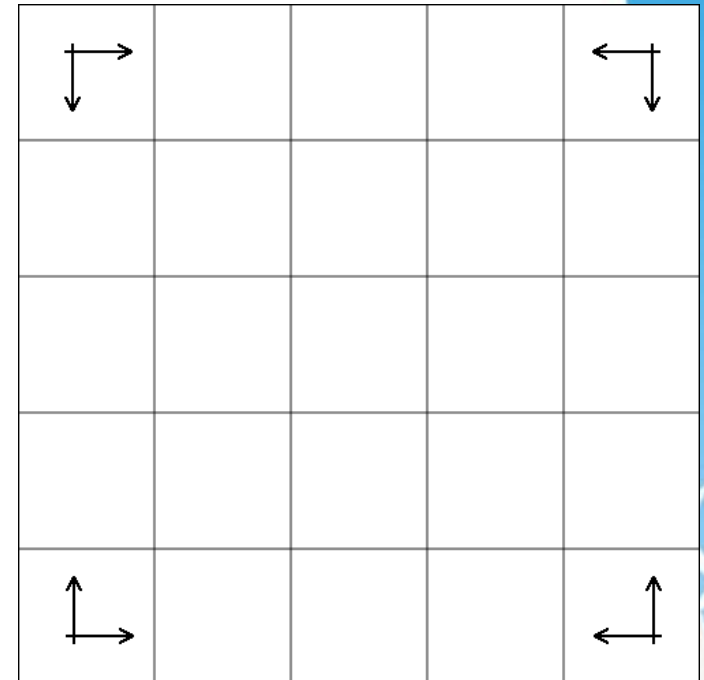
- Técnicas e ferramentas utilizadas
  - Xcode 4
  - Objective-C
  - iPhone Simulator
  - OpenGL ES
  - Doxygen



# Desenvolvimento

- Campo potencial
  - Código fonte do método relax:

```
- (void) relax
{
1   for (int v = 0; v < verticalSize; v++)
2   {
3       for (int h = 0; h < horizontalSize; h++)
4       {
5           [self relaxCellWithHorizontalSize: h
andVerticalSize: v];
6       }
7   }
      ...
25  for (int h = horizontalSize - 1; h > 0; h--)
26  {
27      for (int v = 0; v < verticalSize; v++)
28      {
29          [self relaxCellWithHorizontalSize: h
andVerticalSize: v];
30      }
31  }
}
```



# Desenvolvimento

- Problema de valor de contorno

- Código fonte do método

- getAveragePotentialWithHorizontalSize:andVerticalSize:

```
- (double) getAveragePotentialWithHorizontalSize: (int) horizontalSize_
andVerticalSize: (int) verticalSize_
{
    ...
5   // Busca o valor potencial das células vizinhas
6   if (horizontalSize_ > 0)
7       left = [self getCellPotentialWithHorizontalSize: (horizontalSize_ - 1)
andVerticalSize: verticalSize_];
8   else
9       left = 1.0;
10  if (horizontalSize_ < (horizontalSize - 1))
11      right = [self getCellPotentialWithHorizontalSize: (horizontalSize_ + 1)
andVerticalSize: verticalSize_];
12  else
13      right = 1.0;
    ...
25  // Calcula o valor potencial de acordo com a equação de Gauss-Seidel
26  double difference = (fatorRelax / 8) * ((top - bottom) + (right - left));
27  return (((top + bottom + right + left) / 4) + difference);
}
```

# Desenvolvimento

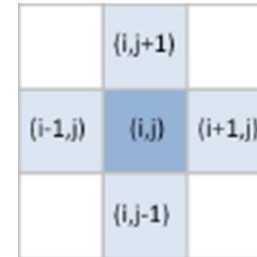
- Movimentação do agente
  - Código fonte do método moveWithRoomArea:

```
- (void) moveWithRoomArea: (BoundingBox *) roomArea
{
1  if (potentialField.wayPointPosition != nil)
2  {
3      Point4D * gradient = [potentialField getGradientWithPosition: position];
4      [gradient normalize];
5      [direction addX: gradient.x * stepMove];
6      [direction addY: gradient.y * stepMove];
7      position = [Mathdef getCellPositionWithX: direction.x -
roomArea.smallerPoint.x andY: direction.y - roomArea.smallerPoint.y
andHorizontalResolution: potentialField.horizontalResolution
andVerticalResolution: potentialField.verticalResolution andUp: NO];
8      [gradient release];
9  }
}
```

# Desenvolvimento

- Movimentação do agente

- Código fonte do método `getGradientWithPosition`:



```
1  - (Point4D *) getGradientWithPosition: (Point2D *) position_  
  {  
2    Point4D * point = [[Point4D alloc] init];  
    ...  
3    double x = [self getCellPotentialWithHorizontalSize: (position_.x - 1)  
andVerticalSize: position_.y] - [self getCellPotentialWithHorizontalSize:  
4    (position_.x + 1) andVerticalSize: position_.y];  
5    double y = [self getCellPotentialWithHorizontalSize: position_.x  
andVerticalSize: (position_.y - 1)] - [self getCellPotentialWithHorizontalSize:  
6    position_.x andVerticalSize: (position_.y + 1)];  
7    double mod = sqrt(x * x + y * y);  
8    double fatorMod = 0.00000005;  
9    if (fabs(mod) > fabs(fatorMod))  
10   {  
11     point.x = (x / mod);  
12     point.y = (y / mod);  
13   }  
14   else  
15   {  
16     point.x = 0;  
17     point.y = 0;  
18   }  
19   return point;  
20 }  
21 }  
22 }
```

# Desenvolvimento

- Operacionalidade

- Código fonte do método initWithCoder:

```
- (id) initWithCoder: (NSCoder *) coder
{
    ...
11     pathPlanning = [[PathPlanning alloc] initWithCoder: coder];
12     pathPlanning.verticalResolution = 40.0;
13     pathPlanning.horizontalResolution = 40.0;
14
15     [pathPlanning createRoomWithSmallerX: 0.0 andGreaterX: 640.0 andSmallerY: 0.0
andGreaterY: 960.0 andSmallerZ: 0.0 andGreaterZ: 0.0];
16
17     [pathPlanning addObstacleWithSmallerX: 200.0 andGreaterX: 400.0 andSmallerY:
500.0 andGreaterY: 600.0 andSmallerZ: 0.0 andGreaterZ: 0.0];
18
19     [pathPlanning addRobotWithX: 80.0 andY: 10.0 andZ: 0.0 andStepMove: 1.5
andIDRobot: 1];
20
21     [pathPlanning addWayPointWithX: 600.0 andY: 800.0 andIDRobot: 1];
22
23     [pathPlanning start];
    ...
}
```

# Desenvolvimento

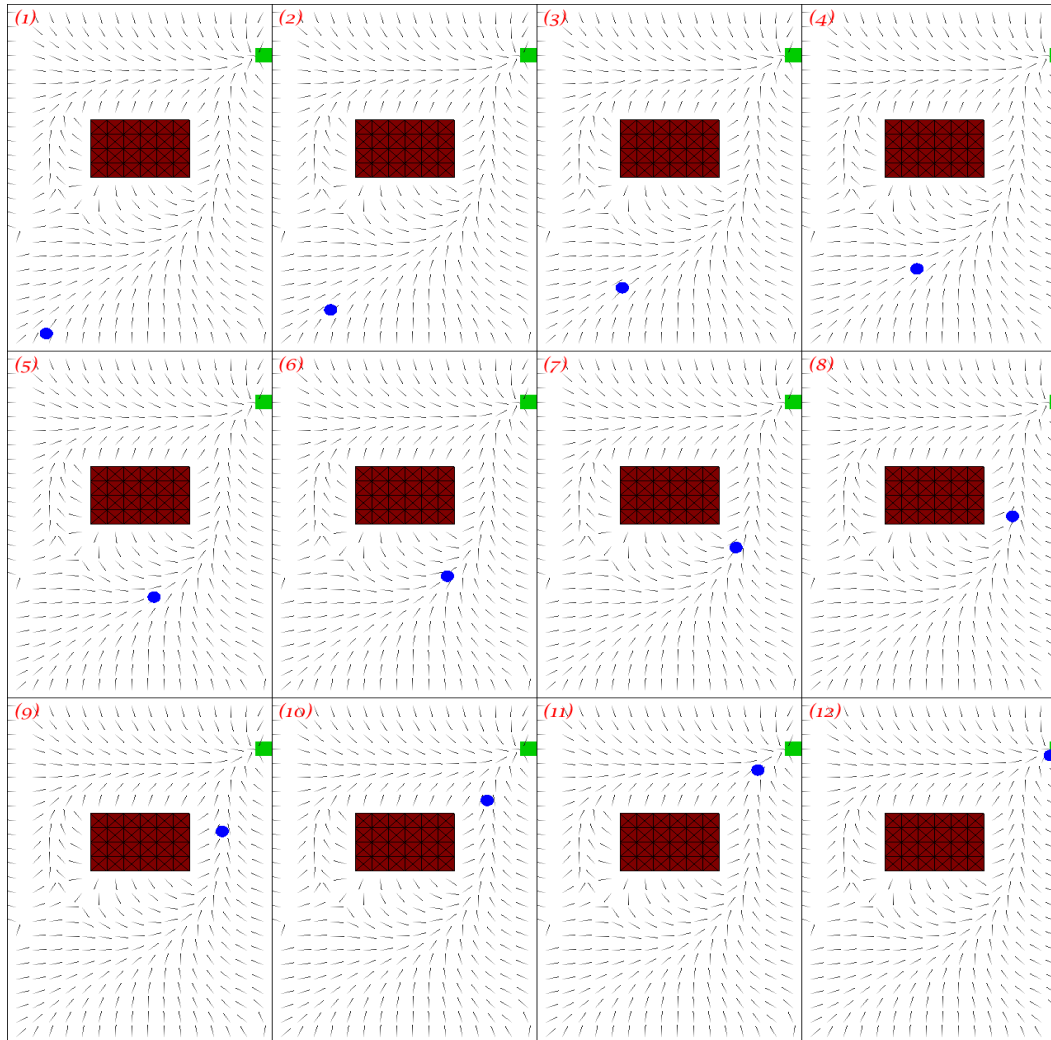
- Operacionalidade
  - Código fonte do método presentFramebuffer

```
- (BOOL) presentFramebuffer
{
    ...
3   if (context)
4   {
    ...
8   [pathPlanning movementRobots];
9   [pathPlanning drawGridRobot: 1];
    ...
12  }
    ...
}
```



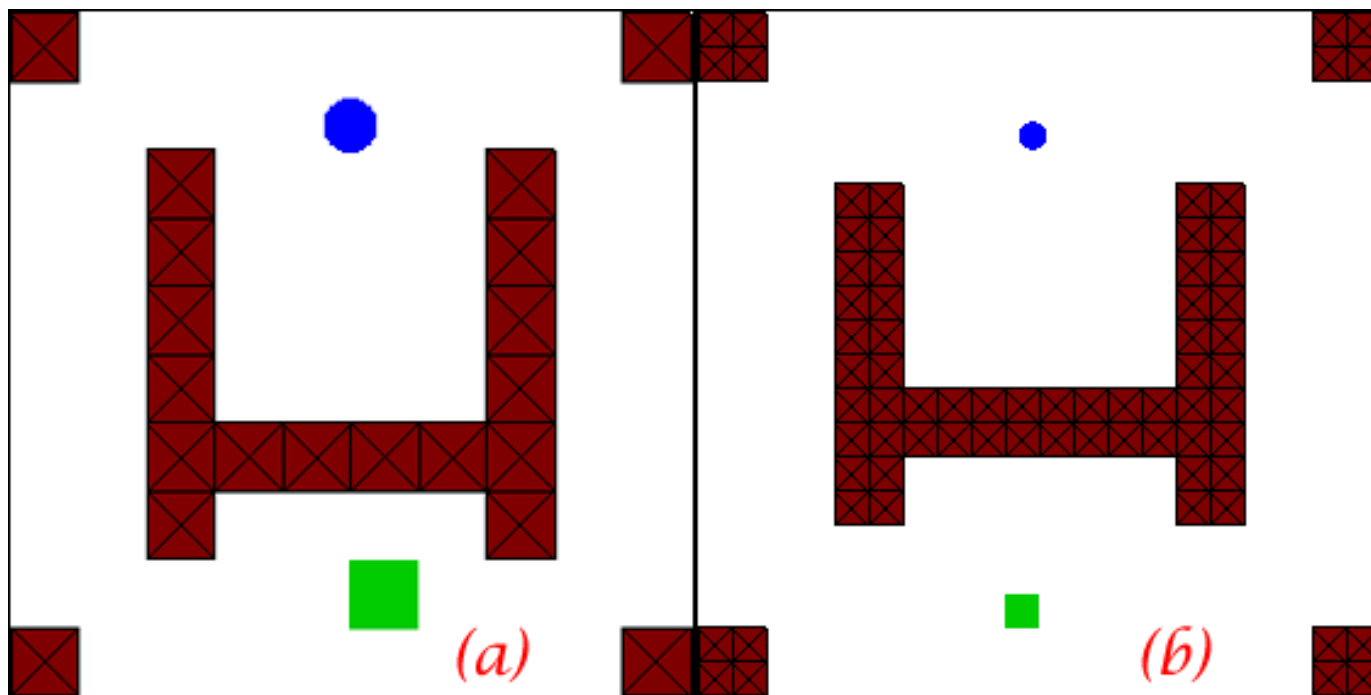
# Desenvolvimento

- Operacionalidade



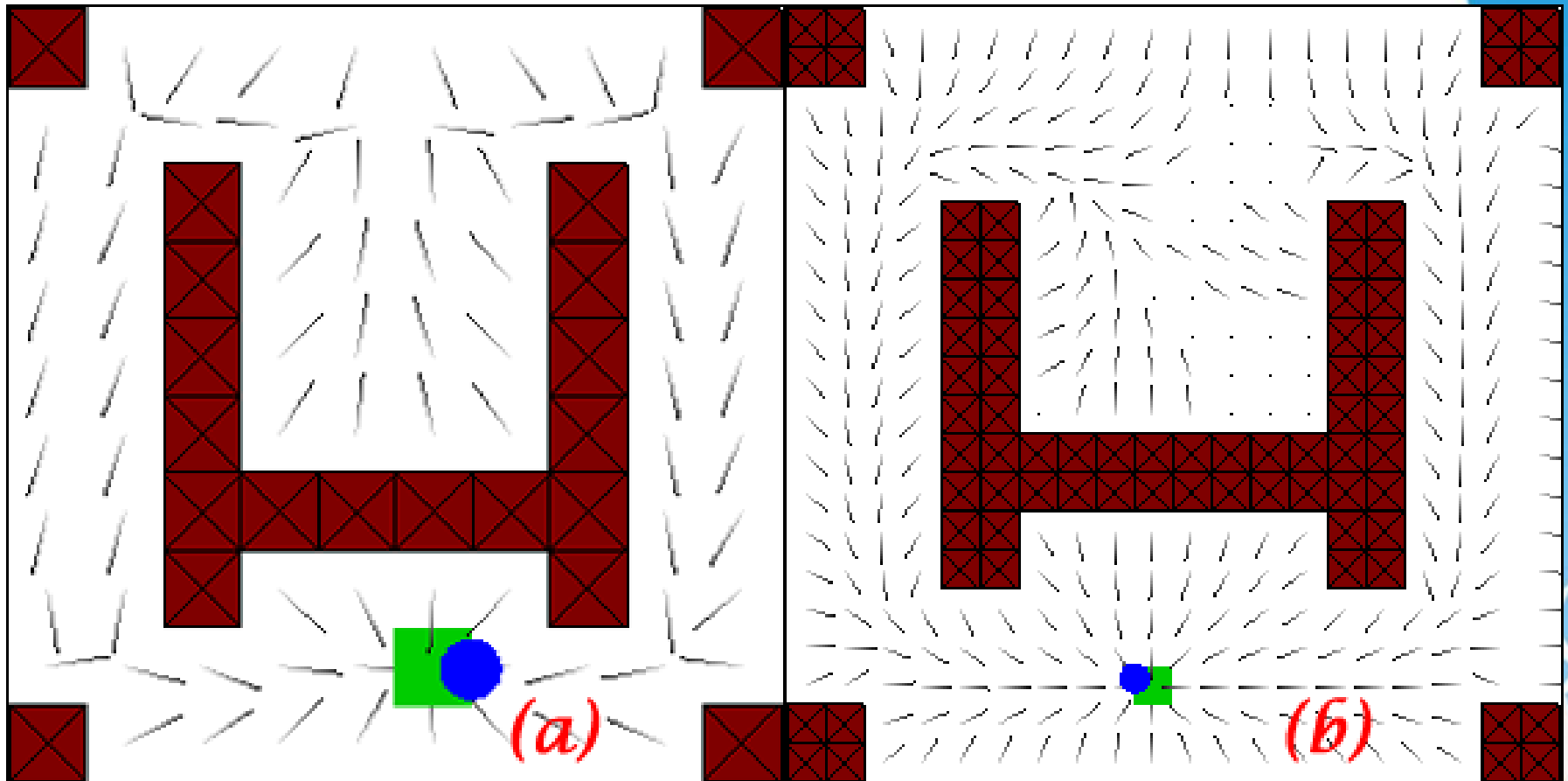
# Resultados e discussão

- Cenário de testes
  - Mínimo local
  - (a) tamanho 10 x 10 – (b) tamanho 20 x 20
  - 1, 10, 20, 30, 40 e 50 execuções do relaxamento



# Resultados e discussão

- Cenário de testes



# Resultados e discussão

- Cenário de testes

Quantidade de execuções do relaxamento	Grade com tamanho 10 x 10 células	Grade com tamanho 20 x 20 células
1	Não	Não
10	Não	Não
20	Sim	Não
30	Sim	Não
40	Sim	Não
50	Sim	Sim

# Resultados e discussão

- Trabalhos correlatos

	FISCHER 2008	FERRARI 2009	MAFRA 2004	Biblioteca Planejamento de Movimento
Suporte a mais de 1 agente	Sim	Não	Sim	Não
Suporte a mais de 1 objetivo sequencial	Sim	Não	Não	Sim
Movimentação próxima ao mundo real	Sim	Não	Não	Não
Evita mínimos locais	Sim	Sim	Não se aplica	Sim
Suporte a plataforma móvel iOS 4	Não	Não	Não	Sim

- Limitações

- Não planeja movimentos para mais de um agente simultaneamente
- Movimento não é próximo ao mundo real



# Conclusão

- Todos os objetivos foram alcançados
- Suporte a plataforma iOS4
- Deploy nos dispositivos
- Vários objetivos sequencialmente
- Planeja movimento de apenas um agente
- Movimentação não gera comportamentos realísticos



# Extensões

- Movimentação mais próxima do mundo real
- Suporte a mais de um agente simultaneamente
- Controle de grupos de agentes com formações
- Mapa global e mapa local



# Demonstração

