2002年度IS課題

VC++6.0による自立型迷路探索ロボットシミュレータの開発

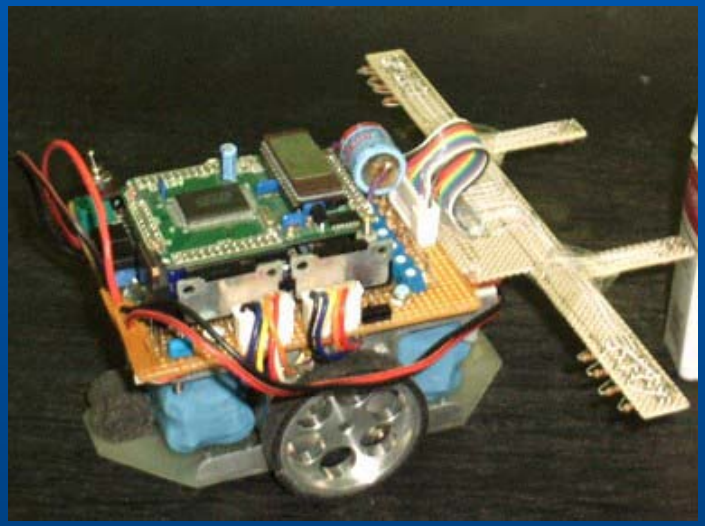
情報工学科3回生　藤川雅男（R00H0390）

研究の目的

- 迷路探索ロボット「マイクロマウス」の開発に当たり、機械的要素の制約を離れ、純粋に迷路探索アルゴリズムの研究を進められるようにすること。
- 将来開発予定の、「実機との通信による仮想迷路プログラム」開発への準備。
- VC++6.0によるWindowsプログラム開発の技能向上。

マイクロマウスとは

- マイコンを利用し、迷路探索を行う自立型ロボット（人は操作できない）
- ゴールまでの時間を競うという形で競技化されており、全国大会も行われている



実際のロボットの外観例

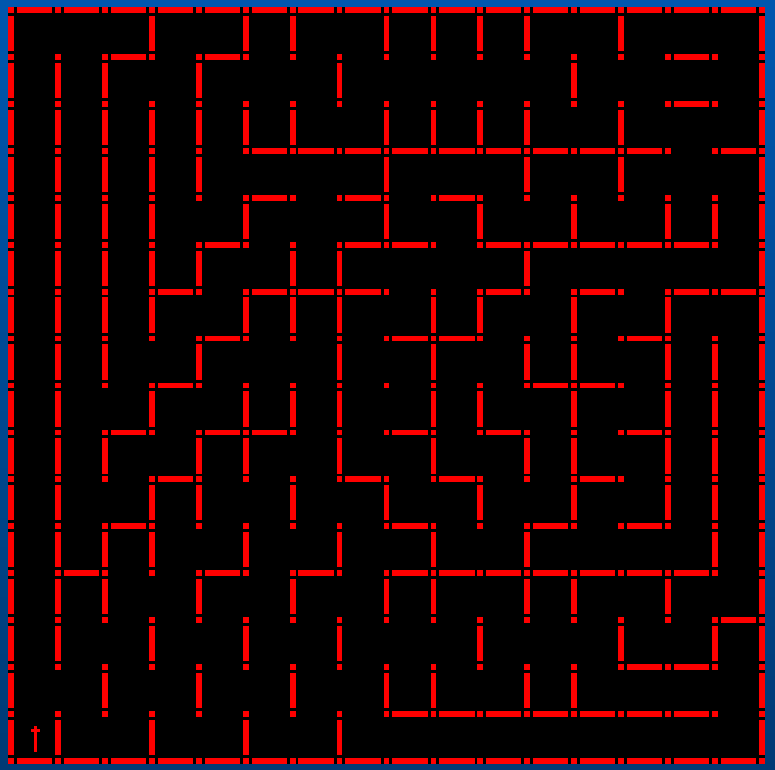
（2001年藤川作）

詳細については…

ニューテクノロジー振興財団マイクロマウス委員会
（http://www.bekkoame.ne.jp/~ntf/mouse/mouse.html）

マイクロマウスの迷路

- 16×16区画で、左端手前がスタート、中央がゴール
- 事前に迷路情報を入力してはならない



迷路の例

競技における制約

- 7分間に5回のトライアル可
- その間に迷路情報を覚え、最短経路を計算する
- 大きさ制限があり、ロボット自身には迷路全体を見渡す能力は（通常）ない

実際の競技の様子（左：藤川）

上部構造は下部構造に依存する

- 最短経路計算
- 迷路探索
- ==========
- センサとモータによる軌道修正
- センサの動作
- モータの加減速
- モータの回転
- 筐体・電池・マイコンなどの準備

昨年度はここまでで

タイムアップ！

何とか完走はできたが・・・

そこで、シミュレータを

- 下部構造については大体問題がなくなった

- では、今年は上部構造の研究をしよう

- むしろハードウエアの存在が重荷に（電池の消耗、タイヤのすべり、・・・）

シミュレータの利点

- ハードウエア的な制約を気にせず、純粋にアルゴリズムのテストをすることができる。
- メモリの内容を、画面上に表示させて確かめることができる。

シミュレータに求められる機能

- 自由な迷路デザインをマウスを用いてグラフィカルに編集できる。
- 編集した迷路デザインを別ファイルに保存することができる。(迷路エディタ)
- 作成した迷路上を仮想のロボットが探索アルゴリズムに従って走ることができる。(仮想ロボット)

顧問・開発環境など

- 顧問：　楠本先生（洛北校）
- 研究場所：　洛北校A教室ほか
- 開発環境：　日立FLORAほか
- OS:　Windows2000、98SE
- 開発ソフト：　VisualC++6.0
- 設計・開発：　藤川雅男

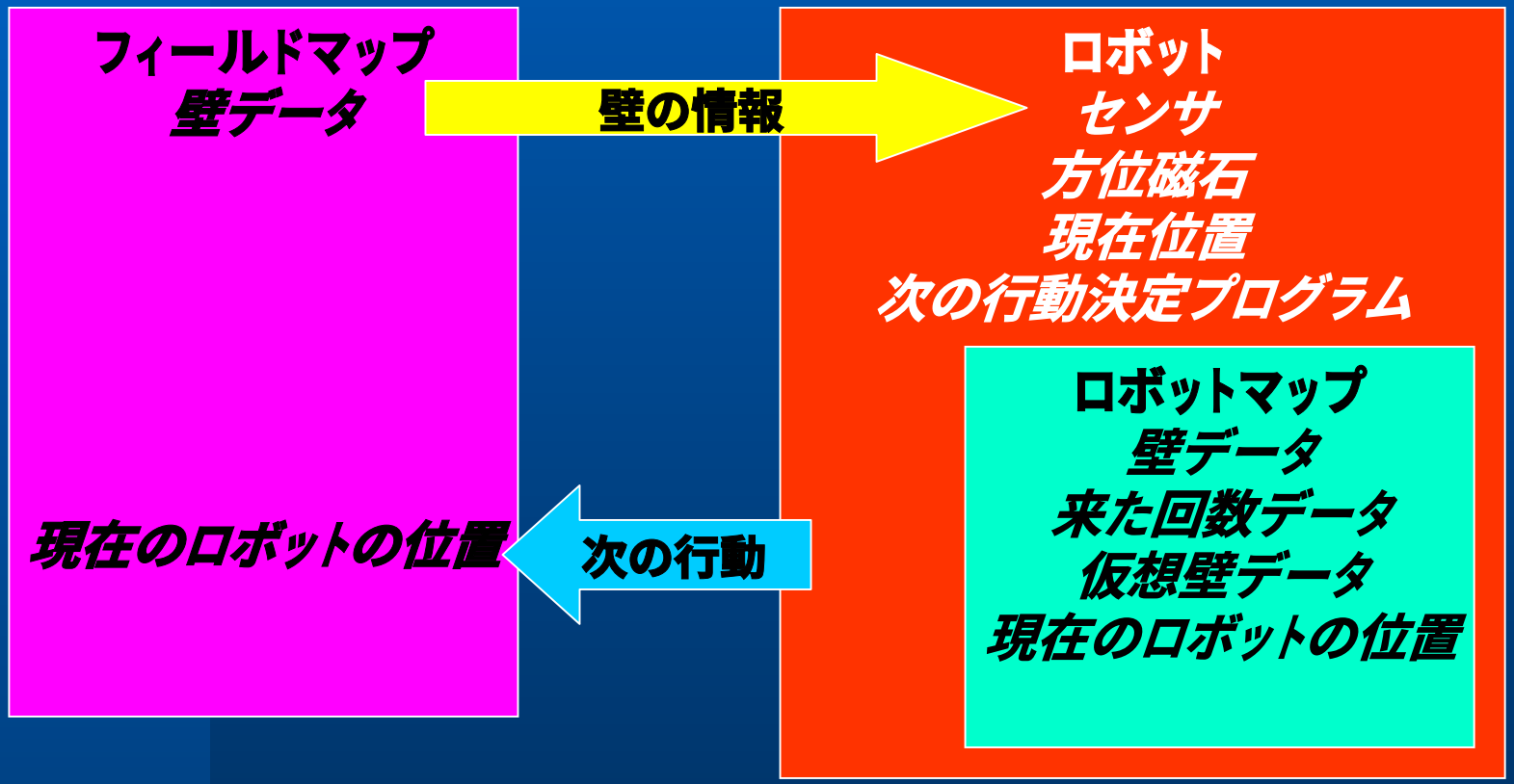
開発の経緯

- 3月　Windowsプログラムについて研究開始
- 4月　大体の構想固まる
- 5月　C言語でプロトタイプ作成開始
- 6月　プロトタイプ完成。コードが煩雑化したためC++の採用を検討。C++の研究開始。オブジェクト指向による全面的な設計のやり直し
- 7月　C++版、表示部分ほぼ完成。

オブジェクトの設計（1）

- 表現すべきもの：「マップ」と「ロボット」
- 「マップ」：実際の迷路（「フィールドマップ」）とロボットの記憶上の迷路（「ロボットマップ」）
- 「フィールドマップ」から「ロボット」が壁の情報をセンサを通して受け取り、「ロボットマップ」に書き込む。この情報からロボットは次の行動を決め、「フィールドマップ」に知らせる。「フィールドマップ」はそれにしたがって新しい位置にロボットを表示する。

オブジェクトの設計(2)


フィールドマップ
壁データ
現在のロボットの位置
壁の情報
次の行動
ロボット
センサ
方位磁石
現在位置
次の行動決定プログラム
ロボットマップ
壁データ
来た回数データ
仮想壁データ
現在のロボットの位置

オブジェクトの設計(3)

似たようなものは一つにまとめてしまいたい→継承を使うと楽

「マップ」クラス
256個のますの壁データ
ロボットの位置
ーーーーーーーーーーーーーー
壁データを読み書きする
壁を描画する
ロボットを描画する

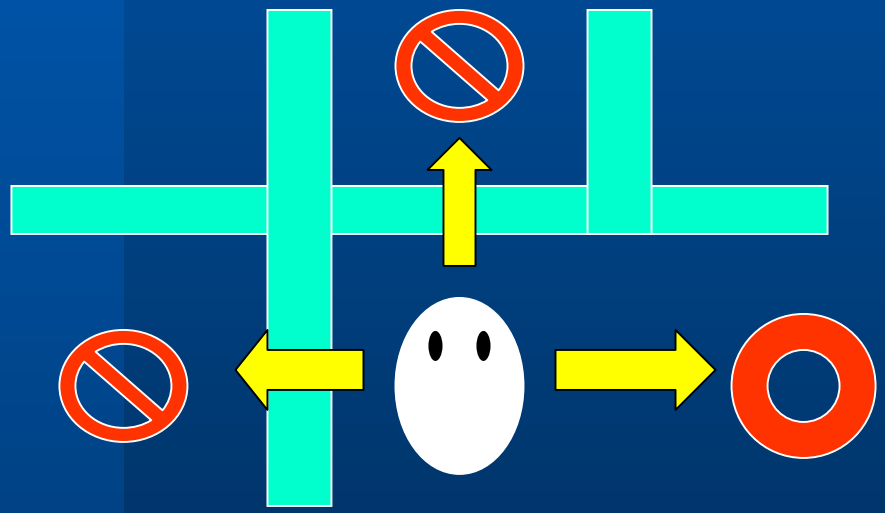
「フィールドマップ」クラス
データをロボットに渡す

「ロボットマップ」クラス
オプションデータ(来た回数など)

継承

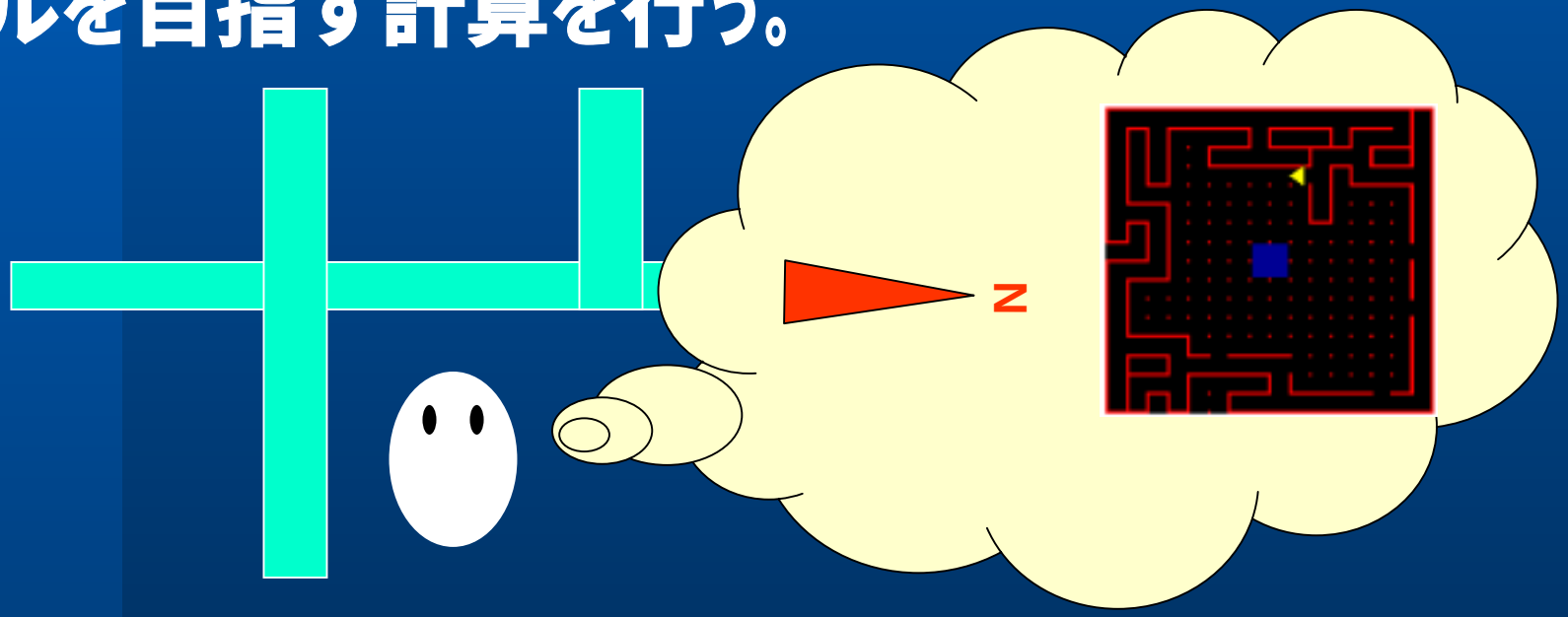
迷路探索

- ロボットは自分のいる区画の前･右･左の壁の有無を認識できるが、ほかの区画を見ることはできない。



迷路探索

- 絶対方向（東西南北）や現在位置・迷路全体の情報などを、メモリ内に保持して、ゴールを目指す計算を行う。

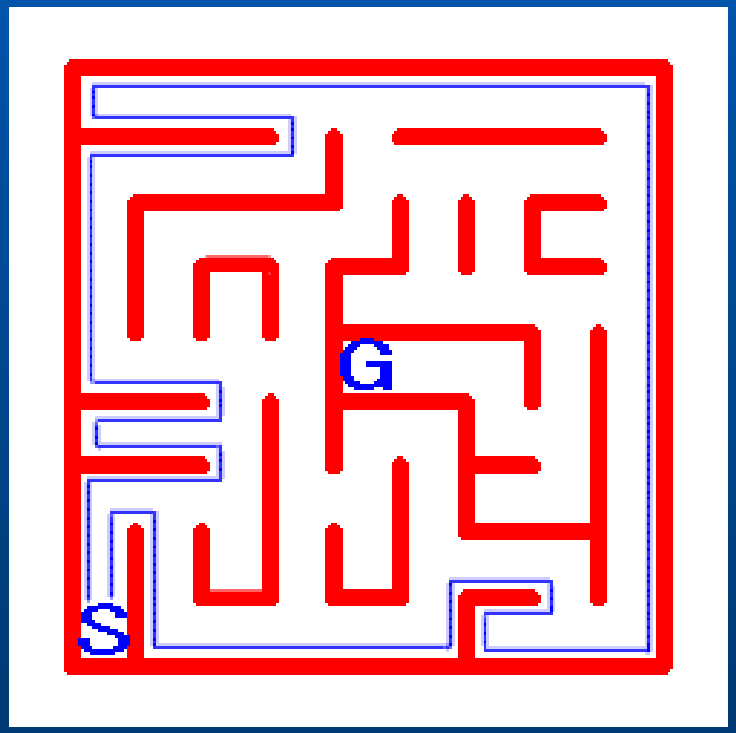


迷路探索

- 探索方法には右手法･求心法･足立法などがある。

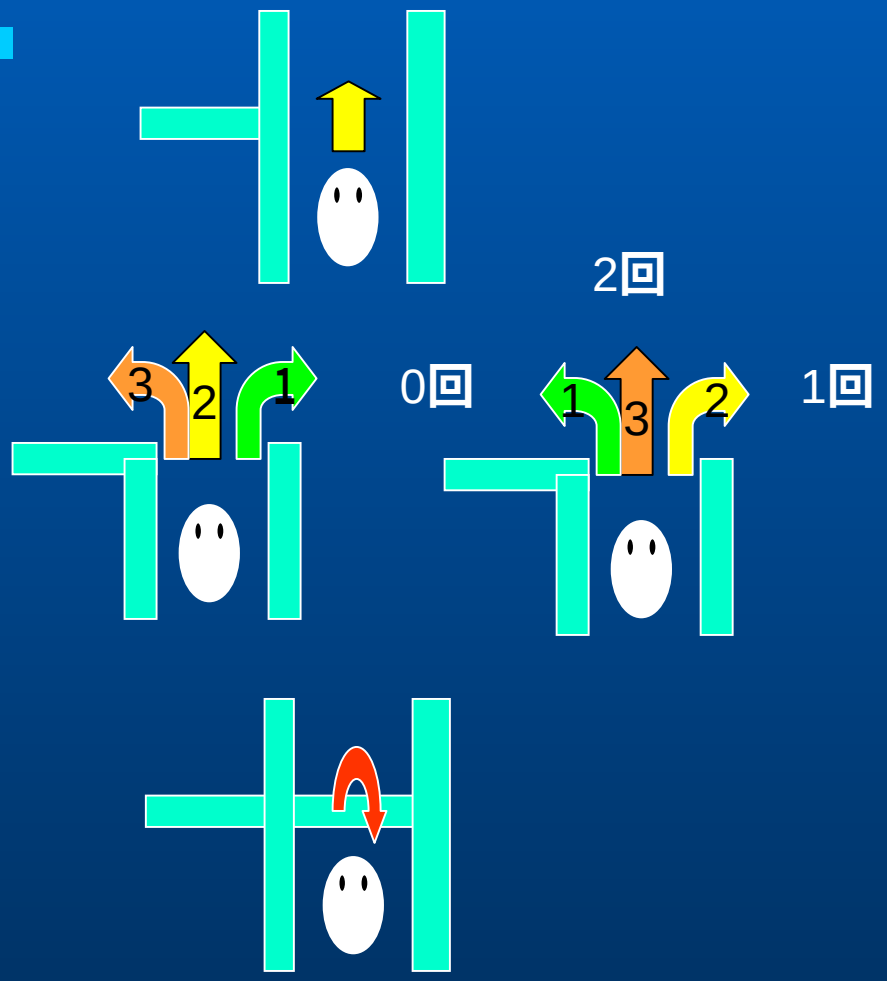
迷路探索アルゴリズム (1)

- 単純右手法 (単純左手法)
- 条件1：左右に壁があるときは直進する。
- 条件2：分岐点にきたときは、右・前・左の順に優先順位をつける。
- 条件3：袋小路はUターンする。
- 通常の迷路ならこれで解けるが、マイクロマウスでは無理 (一周してしまう)
- 左手法では優先順位が左・前・右の順



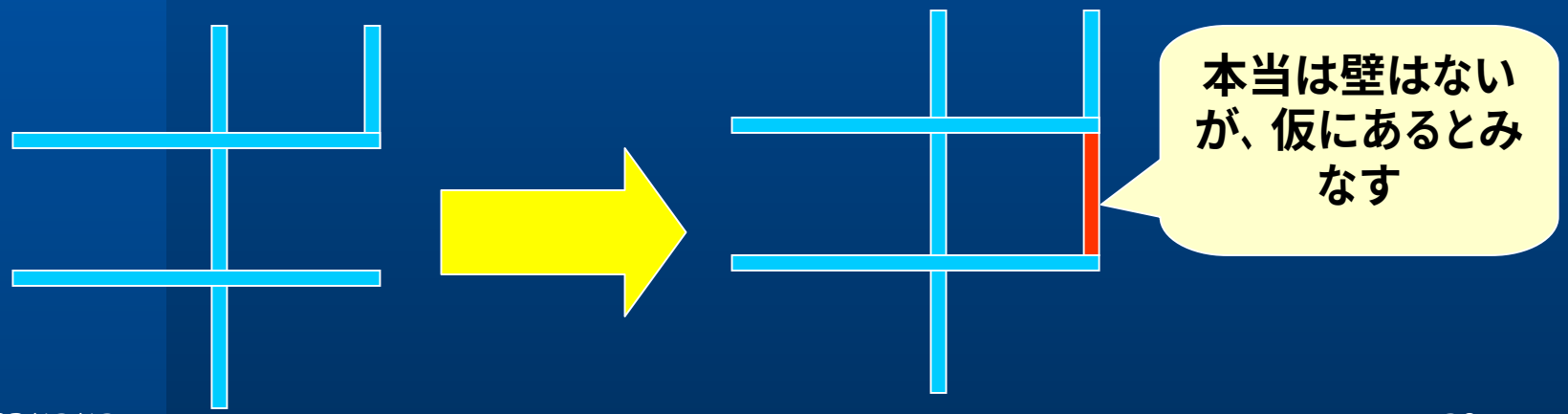
探索アルゴリズム(2)

- 拡張した右手法(左手法)
- 条件1:左右に壁があるときは直進
- 条件2:初めての分岐点にきたときは、右・前・左の順に優先順位をつける。(左手法は逆)
- 条件3:1回以上通過したことのある分岐点にきたときは通過回数の少ないほうへ進む。通過回数がすべて同じであった場合には条件2による。
- 条件4:袋小路はUターン



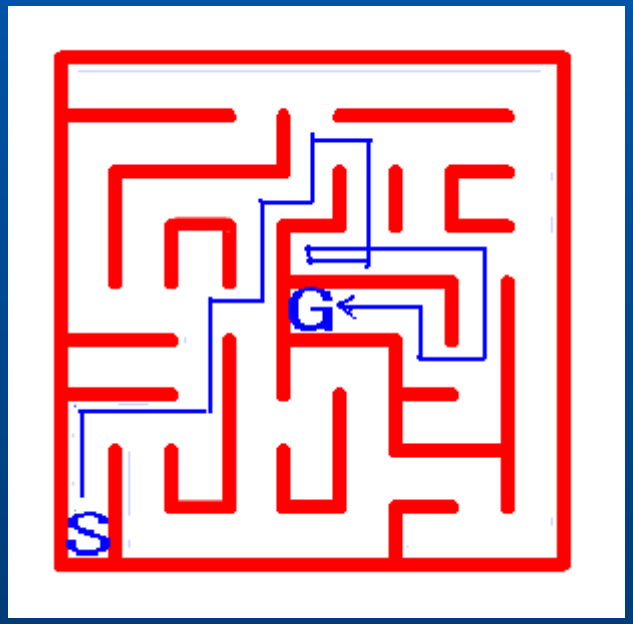
探索アルゴリズム (2)

- 拡張した右手法 (左手法)
- 来た回数を数えるのではなく、すでに来た区画に壁がなくても「仮想壁」を設けて「通れないこと」にする方法もある。



探索アルゴリズム (3)

- 求心法
- 迷路の中央 (ゴール) を目指して進みつづける方法。
- これも着た回数のカウントや仮想壁により無限ループに陥るのを防ぐ必要がある。
- 右手法 (左手法) よりは1回目は早く着くが、全面探索には向かない。



探索アルゴリズム (その他)

- 足立法
- 求心法の一種
- 認識していない壁を「ないもの」とみなし、現在位置からの最短経路を作成する。
- それにしたがって進み、壁にぶつかったら新しい最短経路を作る。
- 最短経路の計算が探索と同時に行える利点がある。

探索アルゴリズム（その他）

- トレモー法
- 迷路のすべての区画を回るのに有効な方法
- 規則1　左右とも壁がある場合は直進
- 規則2　未通過の角ではどの方向に進んでもよい（ここでは中央を目指すことにする）
- 規則3　新しい道を通って通過済の角にきた場合はUターン
- 規則4　通過済の道を通って角へきた場合は通過回数の少ない方へ（優先順位右・前・左）
- 規則5　袋小路ではUターン

今後の開発予定

- 探索アルゴリズムのいくつかが研究途中のため、この研究を進める。
- 「最短経路計算」についても調査を開始する。
- 「実機（H8マイコン使用）との通信による仮想迷路プログラム」の開発の準備を進める。

関連文書

- 実行プログラム
  - Mazeedit.exe,VirtualRobot.exe
- ソースプログラム
  - フォルダMazeC++2a内

参考文献

- プログラミング学習シリーズVisual C++①② (山本信夫著、翔泳社)
- 改訂新C言語入門シニア編 (林晴比古著、ソフトバンク)
- Cプログラマのための C++入門』(柴田望洋著、ソフトバンク)
- やさしいマイコン制御ロボットの製作 (横山直隆著、シータスク)
- ロボティクス入門 (ロボティクス研究会編、コロナ社)
- 初心者のためのロボットの作り方第六版 (マイクロマウス委員会東日本支部)
- 高速マイクロマウスの作り方 (浅野健一著、東京電機大学出版局)
- マイクロマウスの開発研究～マイクロマウス2001に出場して～ (藤川雅男、http://users.kyoto-kcg.ac.jp/~r00h0390/mouse/mm2001.htm)

ほか

以上