Robotik - Übung 03

Pfadplanung

Christoph Zinnen

WS17/18

LABORROBOTIK
Informatik - Computer Science

Informatik
Hauptcampus

HOCHSCHULE TRIER

Gliederung

Map

- Map laden
- ArMap verwenden

Pfadplanung

- ArPathPlanningTask

Aufgaben

- Kartenerstellung
- Raumaufteilung
- Pfadsuche
- Pfadplanung

Gliederung

Map laden

Damit aus dem C++ Code heraus auf die Inhalte der Map zugegriffen werden kann, muss diese via Kommandozeilenargument an den `RobotServer` übergeben werden.

- `-map path/to/map/mapName.map`

Pfade können absolut, oder relativ zum *working directory* angegeben werden.

ArMap verwenden

ArMapObject Typen:

- `Goal`, `GoalWithHeading`
- `Dock`
- `ForbiddenLine`, `ForbiddenArea`
- `RobotHome`

Zugriff auf Map-Objekte:

```
std::list<ArMapObject*> findMapObjectsOfType(
            const char *type,
            bool isIncludeWithHeading=false)
```

Zugriff auf Hindernisse:

```
std::vector<ArLineSegment>* getLines()
```

Gliederung

ArPathPlanningTask I

- ARNL bietet mit der Klasse `ArPathPlanningTask` eine Implementierung der Potentialfeldmethode.
- Starten der Pfadplanung:

```
bool pathPlanToPose(ArPose goal, bool headFlag,
        bool printFlag = true)
```

- Abfragen des Zustandes:

```
PathPlanningState getState()
```

  - NOT_INITIALIZED
  - PLANNING_PATH
  - MOVING_TO_GOAL
  - REACHED_GOAL
  - FAILED_PLAN
  - FAILED_MOVE
  - ABORTED_PATHPLAN
  - INVALID

ArPathPlanningTask II

Hinweis:

Bei Verwendung des Path Planning Tasks werden alle Behaviours (ArActions) deaktiviert.

- Implementierung als *user task*.
- Aus `ArFunctor` ableiten.
- void ArFunctor::invoke() überschreiben
- Dem Roboter hinzufügen:

```
bool ArRobot::addUserTask(const char* name,
        int position,
        ArFunctor* functor,
        ArTaskState::State* state=NULL)
```

Gliederung

Kartenerstellung

- Erstellen Sie eine Karte und laden Sie diese in StudIP hoch.
- Ihnen stehen ~26 Wandstücke a 1m und ~5 a 0,5m zur Verfügung.
- Eine Wand mit 5,5m verbraucht fünf 1m und ein 0,5m Stücke.
- Außenwände zählen nicht mit in die Kalkulation.
- Fahrtwege sollten überall ~1m Breit sein.

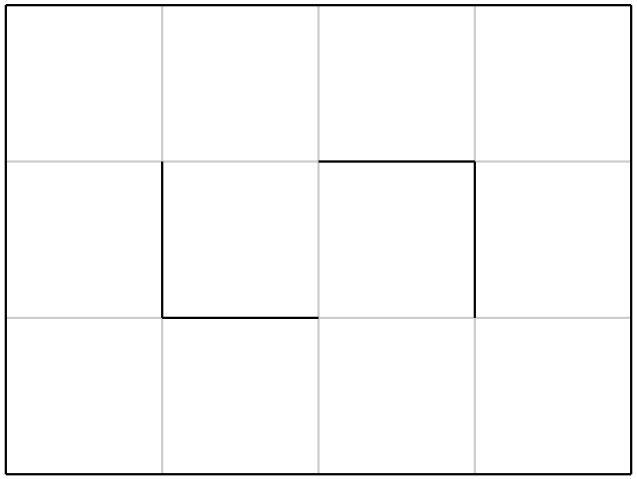

Abbildung 1: Beispiel Map

Raumaufteilung

- Implementieren Sie eine Raumaufteilung Ihrer Wahl.
- Erzeugen Sie in jedem so erzeugten Feld einen Wegpunkt.
- Testen Sie Ihre Raumaufteilung mit Ihrer Karte und den Karten Ihrer Kommilitonen.

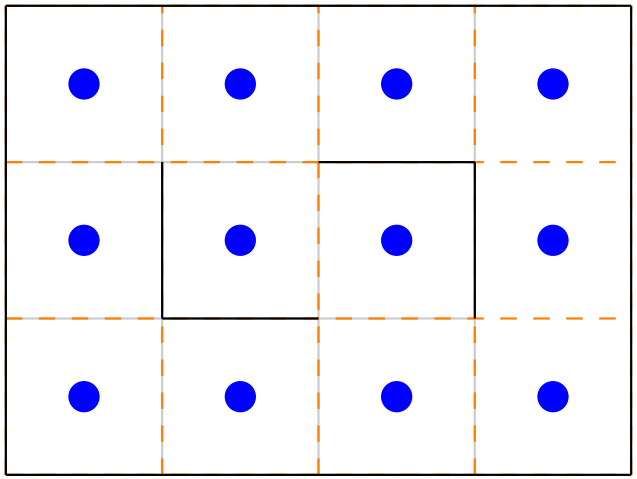

Abbildung 2: Raumaufteilung

Pfadsuche

- Erzeugen Sie einen Navigationsgraphen auf Basis Ihrer Raumaufteilung.

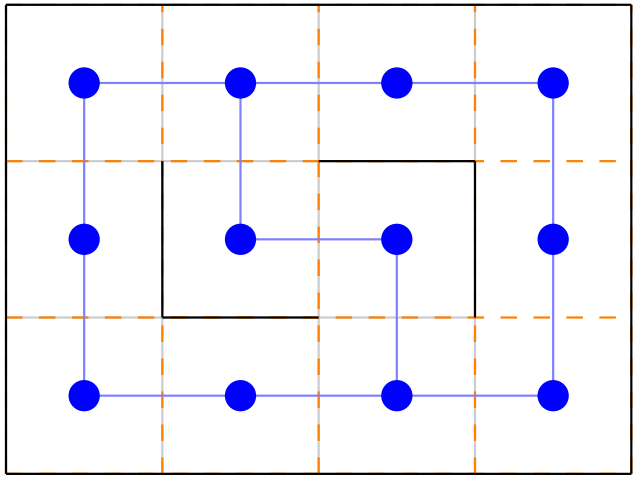

Abbildung 3: Ergebnisgraph

- Implementieren Sie einen Algorithmus zur Pfadsuche auf Ihrem Graphen.

Pfadplanung

- Implementieren Sie ein Verhalten, das den Roboter zu einer beliebigen Position fahren lässt.
- Verwenden Sie das hierarchische Kontrollparadigma.
- Berücksichtigen Sie dynamische Hindernisse.