

Pathfinding

Interface

Allgemein

- Neuen Unterordner *pathfinding*
- Interface *INextPath* erstellt
- *RandomPathfinding* (bestehendes Ausgelagert) und *SpiralPathfinding* hinzugefügt
- *NextScheduler* erstellt

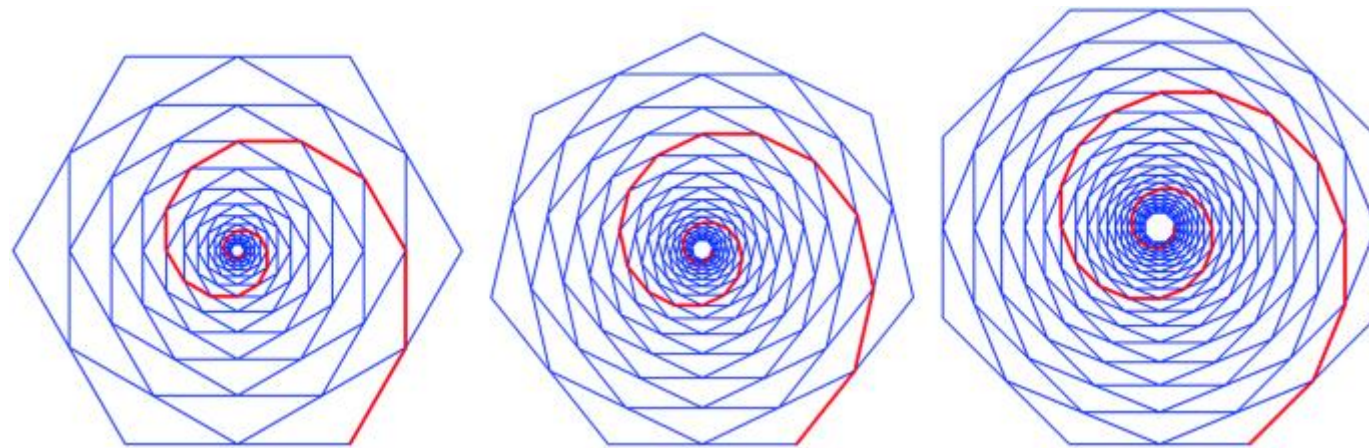
Interface INextPath

- Möglichkeit verschiedene Arten des Pathfindings einzubinden
 - Möglichkeiten Methoden vorzugeben, die alle Pathfindings gleich haben

```
1 package massim.javaagents.pathfinding;
2
3 import eis.iilang.Action;
4
5
6 /**
7  * Interface fuer Pathfinding
8  * @author miwo
9  *
10 */
11 public interface INextPath {
12
13     public Action GenerateNextMove();
14 }
15
```

NextSpiralPath

- Der Agent beginnt und läuft in einer Spirale im Uhrzeigersinn die ganze Map ab. Es ist ziemlich primitiv und sicherlich nicht die beste Strategie, aber zum Karten erkunden nicht schlecht.



NextScheduler

- Erweiterung um eine *PathFindingConfig*.
 - Derweil ist sie mal im NextScheduler und in der javaagentconfig (Ticket erstellt)

```
public class PathFindingConfig {  
    private int pathSteps;  
    private String pathStartDirection;  
  
    PathFindingConfig(int pathSteps, String pathStartDirection) {  
        this.pathSteps = pathSteps;  
        this.pathStartDirection = pathStartDirection;  
    }  
  
    public int GetPathSteps()  
    {  
        return this.pathSteps;  
    }  
  
    public String getPathStartDirection()  
    {  
        return this.pathStartDirection;  
    }  
}
```

Sonstiges

- Try-Catch im *NextPerceptReader* umgebaut

```
//WARNING: ConcurrentModificationException workaround! based on FitBUT
synchronized (percepts) {

    try {
        for (Percept percept : percepts) {

            switch (Constants.EPercepts.valueOf(percept.getName())) {
                // - SimulationStart Messages
                case simStart:
                    simStatus.SetFlagSimulationIsStarted();
                    break;
                case name:
                    agentStatus.SetName(percept.getParameters().get(0).toPrelog());
                    break;
                case team:
                    agentStatus.SetTeam(percept.getParameters().get(0).toPrelog());
                    break;
                case teamSize:
                    simStatus.SetTeamSize(Integer.parseInt(percept.getParameters().get(0).toPrelog()));
                    break;
                case steps:
                    simStatus.SetTotalSteps(Integer.parseInt(percept.getParameters().get(0).toPrelog()));
                    break;
                case role:

```

Sonstiges (Next Scheduler)

```
Agent agent = null;
switch (agentConf.className) {
    case "BasicAgent":
        agent = new BasicAgent(agentConf.name, mailService);
        break;
    case "NextAgent":
        agent = new NextAgent(agentConf.name, mailService, agentConf.pathFindingConfigurations);
        break;
    case "SebastianAgent":
        agent = new NextAgent(agentConf.name, mailService, agentConf.pathFindingConfigurations);
        break;
    // [add further types here]
    default:
        System.out.println("Unknown agent type/class " + agentConf.className);
}
```

Wenn die gleiche Konfiguration in beiden Fällen genutzt wird (hier also NextAgent und SebastianAgent) kann das untereinander geschrieben, und somit doppelte Implementierung gespart werden

```
Agent agent = null;
switch (agentConf.className) {
    case "BasicAgent":
        agent = new BasicAgent(agentConf.name, mailService);
        break;
    case "NextAgent":
    case "SebastianAgent":
        agent = new NextAgent(agentConf.name, mailService, agentConf.pathFindingConfigurations);
        break;
    // [add further types here]
    default:
        System.out.println("Unknown agent type/class " + agentConf.className);
}
```

Fragen / Weiteres

- NextAgent > generatePossibleAction
 - Hier weiter vorgehen, wenn keine Action gefunden wurde oder lieber abbrechen und einen neuen step gehen?
- Anderen „dummen“ Algorithmus oder gleich weiter mit einem „schlaueren“?
- Sollen wir hier auch den NextScheduler verwenden oder den bestehenden erweitern? Wird dieser hier geupdatet?
- Mergen?