Migrating to AdvancedRobot

RoboCode Special Interest Topic

Christian Riess, Eva Eibenberger

Pattern Recognition Lab (Computer Science Dep. 5)

Friedrich-Alexander-University Erlangen-Nuremberg

AdvancedRobot – Simultan Anweisungen geben

- Statt

```
public class RockHard extends Robot
public class RockHard extends AdvancedRobot
```

- Anweisungen werden mit set* gegeben, z.B.
  - **`setAhead()`**
  - **`setFire()`**
  - **`setTurnGunLeft()`**
  - **`setTurnRadarRight()`**, usw.

- Die Anweisungen werden parallel abgearbeitet, sobald **`execute()`** aufgerufen wird

- Wird **`execute()`** zu lange nicht aufgerufen, gilt der Roboter als inaktiv und hat verloren

- –> daher am besten ein **`execute()`** mit in die Hauptschleife in der **`run()`**-Methode schreiben!

AdvancedRobot – Abhängige Anweisungen

- Vorsicht bei abhängigen Anweisungen!
  - Beispiel:
    Angenommen, wir wollen die Kanone 10° nach rechts drehen und dann schießen
  - -> dann muss sichergestellt sein, dass die Kanone vollständig gedreht wurde, bevor die **`setFire()`**-Anweisung ausgeführt wird
  - Zwei Implementierungsmöglichkeiten:

Anderweitig beschäftigen, schießen wenn Kanone ausreichend weit gedreht:

```
setTurnGunRight(10);
boolean iWantToShoot = true;
// (..)
while (true) {
  if (iWantToShoot &&
      getGunTurnRemaining() < 0.001)
  {
    setFire(2);
  }
  // (..)
  execute();
}
```

Vorteil: Optimale Zeitnutzung
Nachteil: Notizen mitführen (**`iWantToShoot`**)

Explizit warten:

```
setTurnGunRight(10);
waitFor(new GunTurnCompleteCondition());
setFire(2);
execute();
```

Vorteil: Sehr übersichtlich
Nachteil: Nicht möglich, neue Anweisungen, z.B. Bewegung, zwischenzuschieben (außer in **`on*Event`**-Methoden)

Andere coole Features

- Dinge, die cool sind, wir aber (wohl) nicht brauchen werden:
  - Eigene Ereignisse definieren:

    Eine Unterklasse von **`CustomEvent`** mit eigener Bedingung (**`Condition`**) implementieren.
    Besonders wichtig: Implementierung der **`test()`**-Funktion in **`Condition`**
    Diese Ereignisse in **`onCustomEvent()`** fangen.

  - Kämpfe offline analysieren:
    Mit **`RoboCodeFileWriter`** bzw. **`RoboCodeOutputStream`** während dem Kampf Daten herausschreiben.

  - Ereignisprioritäten ändern:
    `setPriority()` – Ereignisse mit höherer Priorität unterbrechen Ereignisse mit niedrigerer Priorität.
    Nicht nötig für uns: Die Voreinstellungen sind recht gut.

- **Viel Spaß!**