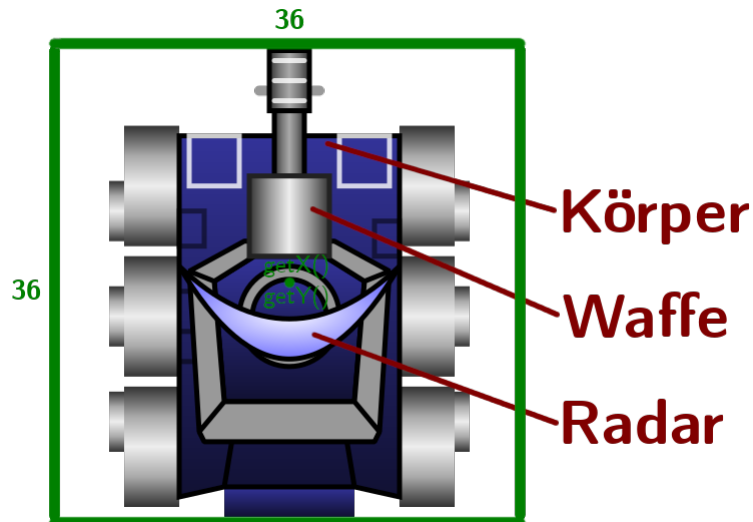


Robocode: Spielphysik – CheatSheet

1 Anatomie eines Roboters

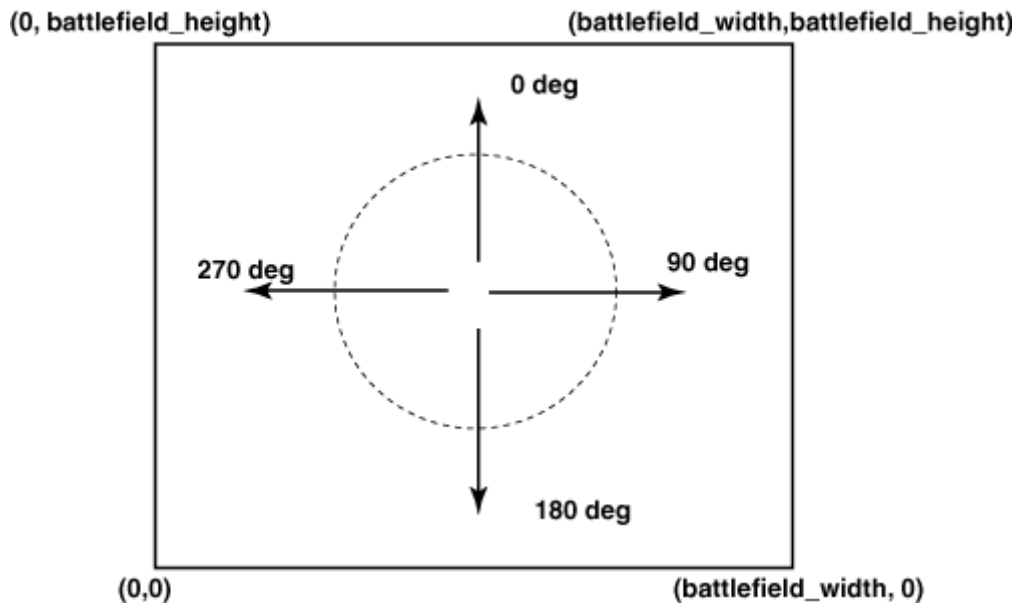


- Ein Roboter ist 36×36 pixel groß (*bounding box*)
- Für **SimpleRobot**: Obwohl das Radar auf der Waffe und die Waffe auf dem Körper angebracht sind, lassen sich alle 3 Teile unabhängig voneinander drehen. `turnGun(...)` bewegt also den Radar nicht mit – die Ausrichtung des Radars bleibt unverändert

2 Energie

- Jeder Roboter hat am Anfang 100 Energie
- Wenn ein Roboter 0 Energie hat, wird er *deaktiviert* (der Roboter wird bewegungs- und schussunfähig und wird besiegt, sobald er getroffen wird oder eine bestimmte Zeit verstrichen ist). Ein Roboter kann Energie zurückgewinnen, falls eine seiner (vorher) geschossenen Kugeln einen Gegner trifft
- Schießen kostet Energie
- Treffen eines anderen Roboters gibt Energie
- Getroffen werden von einem anderen Roboter kostet Energie
- Rammen eines anderen Roboters kostet Energie
- Rammen einer Wand kostet Energie

3 Geometrie des Spielfelds



4 Zeit

Die Zeit wird in *ticks* gemessen.

- Jeder Roboter hat einen *turn* pro tick (in dem der Roboter die für ihn programmierten Aktionen ausführt). Sollte der Roboter seine Aktion (z.B. `ahead(13333337)`) aus Zeitgründen (s.u.) nicht in einem turn beenden können, wird diese Aktion pausiert und im nächsten turn fortgeführt. Falls im nächsten turn eine gleiche Aktion ausgeführt wird (z.B. `ahead(10)`), so wird die ursprüngliche Aktion überschrieben. In diesem Fall bewegt sich der Roboter in 3 turns also insgesamt $8 + (8 + 2)$ pixel vorwärts.
- In einem tick wird folgender *processing loop* (in dieser Reihenfolge) durchlaufen
 1. Spielfeld wird aktualisiert und neu gezeichnet
 2. Alle Roboter führen ihren Code aus, bis sie aktiv werden (also nicht nur interne Berechnungen machen, sondern z.B. eine Bewegung ausführen)
 3. Zeit wird aktualisiert
 4. Alle Kugeln bewegen sich
 5. Alle Roboter bewegen sich (Reihenfolge: Waffe, Radar, Richtung, Beschleunigung, Geschwindigkeit, Abstand; Ein `SimpleRobot` kann sich gleichzeitig (in einem turn) nach vorne bewegen und seinen Radar stehen)
 6. Alle Roboter führen scans (mit dem Radar) etc. aus
 7. Alle Roboter führen ihren Code weiter aus

Insbesondere bedeutet dies, dass Kugeln abgefeuert werden (4.), bevor die Waffe gedreht wird (5.)!

5 Raum

5.1 Bewegung

Distanzen werden in *pixel* angegeben.

- **Beschleunigung:** $a = a + 1$ pixel/turn. Die initiale Geschwindigkeit ist 0.
- **Bremsen:** $a = a - 2$ pixel/turn.
- **Geschwindigkeit:** $v = \max\{\text{Beschleunigung} \cdot \text{Zeit}, 8\}$. Die maximale Geschwindigkeit ist 8 pixel in einem turn.

5.2 Rotation

Winkel werden grundsätzlich in *Grad* ($^{\circ}$) angegeben. Es stehen Funktionen zur Konvertierung zum Bogenmaß bereit (s. Dokumentation zu **SimpleRobot**).

- **Körper:** Maximal $10 - 0.75 \cdot |\text{Geschwindigkeit}|$ $^{\circ}$ /turn. Je schneller man sich bewegt, desto langsamer dreht man sich also.
- **Waffe:** Maximal 20 $^{\circ}$ /turn.
- **Radar:** Maximal 45 $^{\circ}$ /turn.

5.3 Kugeln

Kugeln sind abhängig von der *firepower*, die bei Schießen einer Kugel mit angegeben wird.

- **Schaden:** $4 \cdot \text{firepower} + \max(0, 2 \cdot (\text{firepower} - 1))$. Ein Schuss mit $\text{firepower} > 1$ verursacht also zusätzlichen Schaden.
- **Geschwindigkeit:** $20 - 3 \cdot \text{firepower}$.
- **Energiegewinn bei Treffer:** $3 \cdot \text{firepower}$

5.4 Kollisionen

- **Mit anderem Roboter:** -0.6 Energie.
- **Mit Wand:** $\max(0, |v| \cdot 0.5 - 1)$

Insbesondere bedeutet also z.B. **ahead(42)** nicht, dass sich ein Roboter 42 Pixel in einem turn weiterbewegt – er wird stattdessen einige turns lang geradeaus fahren (wie lange, ist u.a. abhängig von der Geschwindigkeit).