

# PyBots 1.0

<b>PYBOTS 1.0</b>	<b>1</b>
1.0 ZUSAMMENFASSUNG	2
2.0 DAS SPIELFELD	3
2.1 Ein Feld	5
3.0 ROBOTER	6
3.1 Bewegungsmuster eines Roboters	7
3.2 Damagesystem	8
4.0 DIE K.I.	9
4.1 K.I. Funktionen	9
5.0 HANDICAP	10

---

## 1.0 Zusammenfassung

In PyBots kämpfen zwei gegnerische KI's auf einer 9x9 Map. Das Spiel ist ein turn-based-game, das heißt, beide Roboter ziehen nacheinander. Rot zieht zuerst.

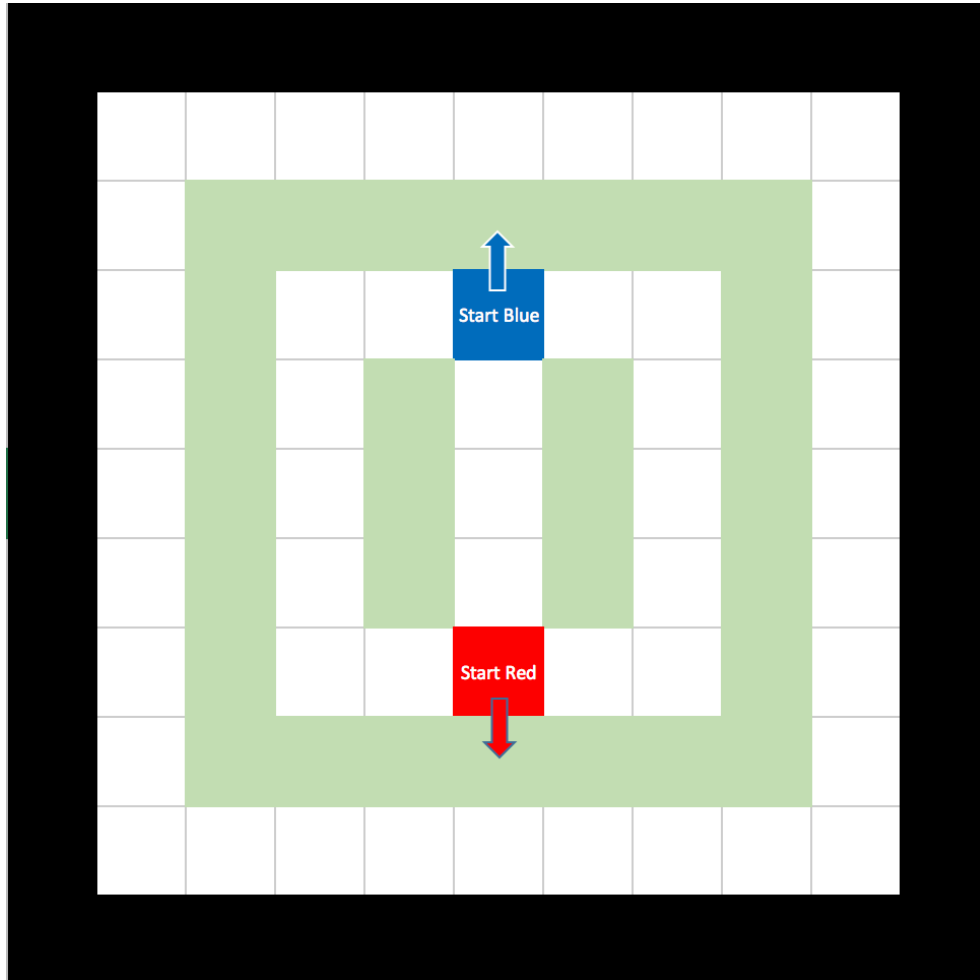
Gewonnen hat Spieler A wenn:

- Roboter B keine Leben mehr hat
- Roboter A nach x Zügen mehr Felder besitzt
- Spieler B aufgibt

Remis gibt es bei einer 9x9 Map grundsätzlich nicht (81-2 Felder), es sei denn, beide Spieler einigen sich auf ein Remis.

## 2.0 Das Spielfeld

Das Spielfeld ist ein 9x9 Gitter:

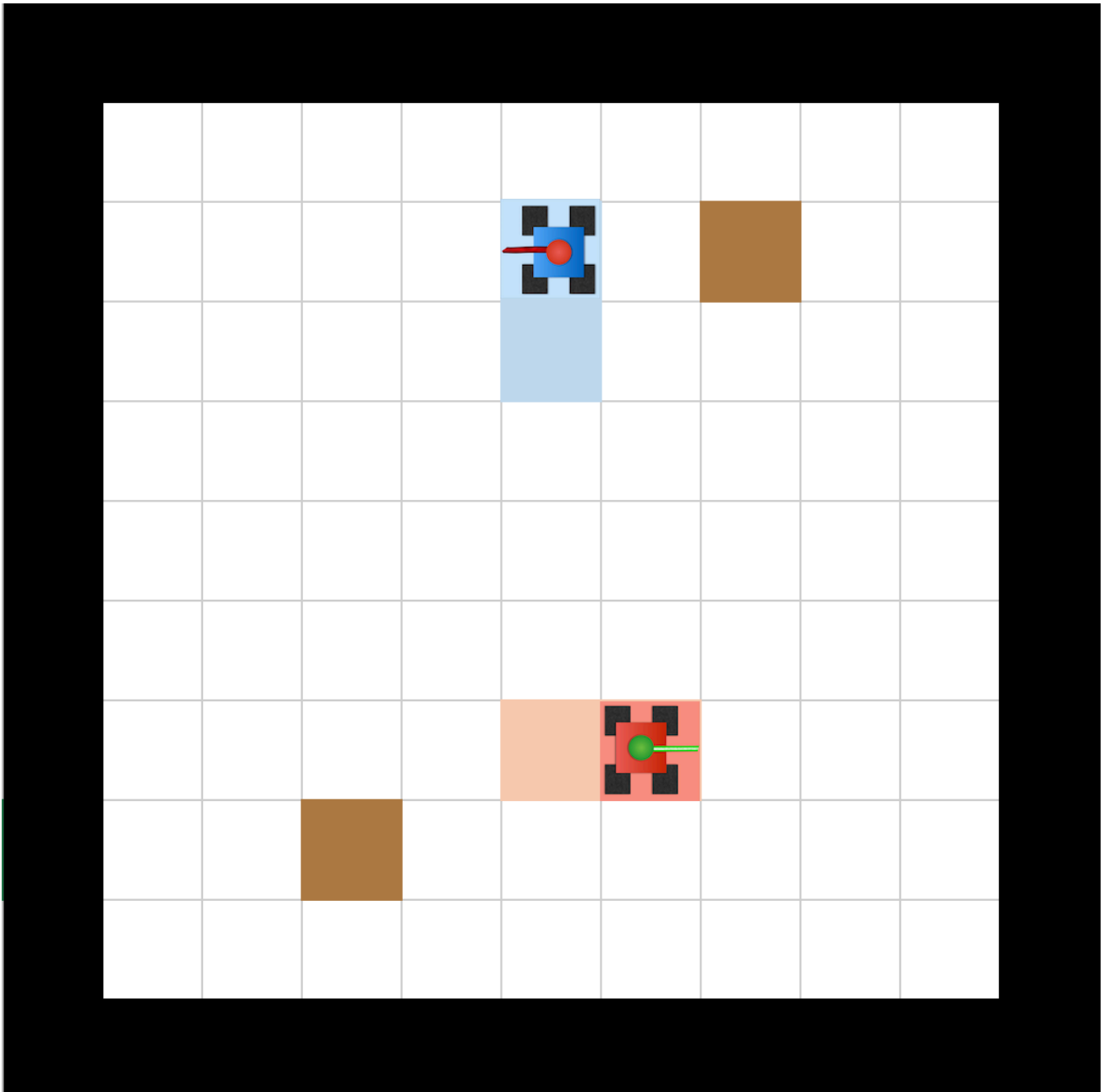


- 
- Beide Roboter starten gegenüber in entgegengesetzte Richtungen<sup>1</sup>
- Entlang der grünen Felder werden zwei unbegehbare Gebiete punktsymmetrisch zur Mitte in jedem Spiel zufällig neu erstellt
- Bei einer Bewegung beansprucht jeder Roboter das Feld, auf dem er steht, für sich (dargestellt in einer helleren Version der Teamfarbe). Dies gehört so lange zu ihm, bis ein anderer Roboter das Feld betritt.

---

<sup>1</sup> So wird ein sofortiges Aufeinanderzustürmen verhindert

Ein Beispiel, wie das Spielfeld nach 3 Zügen aussehen könnte:



(Züge Spieler Blau: Bewegung, Bewegung, Rotation links; Züge Spieler Rot: Rotation links, Bewegung, Bewegung)

## 2.1 Ein Feld

Ein Feld hat immer 4 Attribute.

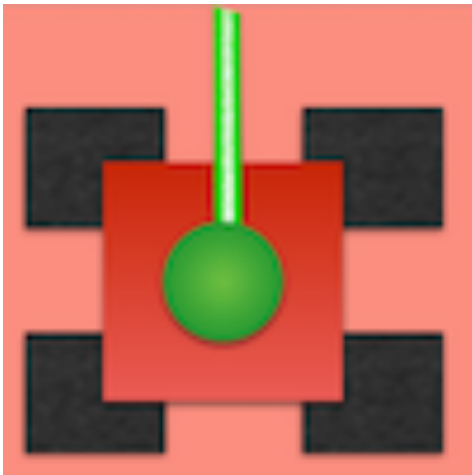
Attribute	Values	Beschreibung
<b>.passable</b>	True, False	Gibt an, ob ein Roboter das Feld betreten kann
<b>.type</b>	„field“, „wall“, „obstacle“	Gibt die Art des Feldes an
<b>.team</b>	„zero“, „blue“, „red“	Gibt die Teamzugehörigkeit des Feldes an
<b>.occupied</b>	„zero“, „roboter_blue“, „roboter_red“	Gibt an, ob ein Roboter auf diesem Feld steht

---

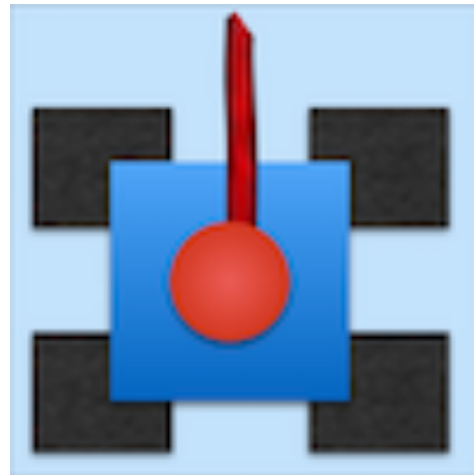
## 3.0 Roboter

Jeder Spieler steuert einen Roboter mit einer von ihm entwickelten KI.

Die beiden Roboter sehen so aus:



Roboter Red



Roboter Blue

Der Laser deutet immer in Fahrtrichtung.

### 3.1 Bewegungsmuster eines Roboters

Bei jedem Zug hat ein Roboter 5 Möglichkeiten:

Function	Aktion	Beschreibung
<b>.move_forward()</b>	Roboter bewegt sich	Der Roboter zieht in Fahrtrichtung um 1 Feld nach vorne
<b>.turn_left()</b>	Roboter dreht sich	Der Roboter dreht sich in Fahrtrichtung nach links
<b>.turn_right()</b>	Roboter dreht sich	Gibt die Teamzugehörigkeit des Feldes an
<b>.move_back()</b>	Roboter bewegt sich	Der Roboter zieht in Fahrtrichtung um 1 Feld nach hinten
<b>.attack()</b>	Roboter attackiert	Wenn der gegnerische Roboter direkt (in Fahrtrichtung) vor diesem Roboter steht, attackiert er ihn

Wenn eine Bewegung nicht ausführbar ist (z.B. `field.passable == False`), bewegt sich der Roboter in diesem Zug nicht

## 3.2 Damagesystem

Jeder Roboter hat zu Spielbeginn 100 Leben. Folgende Damagepunkte werden bei einer Attacke erhalten:

Richtung	Damage	Erklärung
<b>Frontal</b>	10	Frontales Aufprallen ist nicht sehr effektiv. Es gewinnt dabei erstens schlussendlich eh der Spieler, der dabei den ersten Treffer landet, und es ist auch kein guter Stil
<b>Seitlich</b>	20	Seitlich zu treffen, dazu gehört mehr Geschick. Deswegen macht dies moderate 20 damage
<b>Von Hinten</b>	50	Um einen Roboter von hinten zu treffen, muss die andere KI einen Fehler machen, und die eigene KI muss diesen erkennen und ausnutzen. Daher wird dies mit viel Damage belohnt



---

## 4.0 Die K.I.

Jeder Spieler muss seine eigene K.I. schreiben. Die K.I. wird als Modul in die API geladen und dort ausgeführt. Während eines Spieles hat der Spieler keinen Zugriff auf seine K.I.

### 4.1 K.I. Funktionen

In dem Modul kann der Spieler (abgesehen von der Bewegungsfunktion) diese Funktionen ausführen):

Function	Beschreibung
<b>.lookAtField(x, y).args</b>	Die K.I. erhält den Wert eines Attributes des Feldes [x, y]
<b>self.get_robot_position()</b>	Die K.I. erhält die Position des Feldes, auf dem der Roboter gerade steht
<b>self.get_robot_rotation ()</b>	Die K.I. erhält die Rotation des eigenen Roboters

Es ist nicht möglich, die Rotation des gegnerischen Roboters mit Hilfe einer Funktion zu erhalten. Man kann die Position erfahren, in dem mit `lookAtField(x, y).occupied` solange herumprobiert wird, bis man „robot\_red“ bzw. „robot\_blue“ erhält

---

## 5.0 Handicap

Das Ziel von PyBot ist, eine gewitzte KI zu schreiben, nicht, durch ausprobieren die besten Züge festzustellen. Daher werden folgende Maßnahmen empfohlen

- Jede KI darf nicht mehr als 200 Zeilen umfassen
- Die Berechnungszeit eines Zuges darf 5,0s nicht überschreiten
- Nach jedem Turnier muss ein funktionstüchtiger Programmcode den anderen Mitspielern überlassen werden, es ist allerdings gestattet, Kommentare zu entfernen und Variablen umzubenennen