

Greenfoot - Lunar Lockout

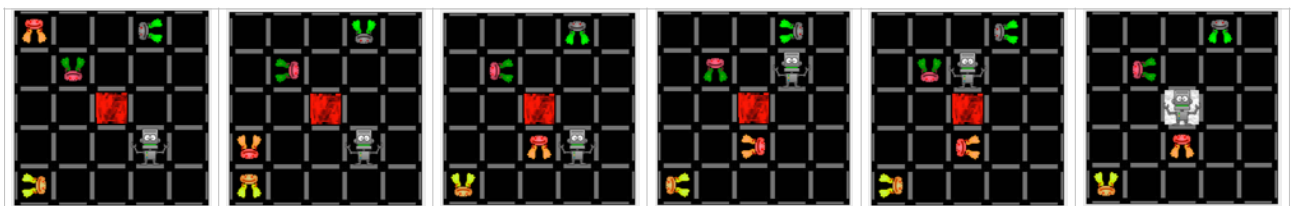
Beschreibung:

Lunar Lockout ist ein Denkspiel, in dem man versuchen muss, eine Spielfigur (hier der graue LunarBot auf Feld 4,4) auf das rote Zielfeld (in der Mitte) zu bewegen. Dabei darf der LunarBot immer nur in eine Richtung (right, left, up oder down) bewegt werden, in der sich in Verlängerung mindestens ein Roboter befindet, der den LunarBot abbremst. Dabei muss der LunarBot genau soweit bewegt werden, bis er direkt vor dem Roboter steht. Der Spielfeldrand dient dabei nicht als Begrenzung! Die Grafik illustriert die dementsprechend einzig mögliche Lösung für die Anfangssituation:



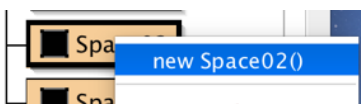
Im Lunar Lockout Spiel dürfen zudem mit derselben Regel auch die Roboter bewegt werden, so dass ein anspruchsvolles taktisches Spiel entsteht, in dem die Reihenfolge der Aktionen den Erfolg oder Misserfolg der Taktik bestimmt. Während der LunarBot mit den Pfeiltasten gesteuert werden soll (und von dir später programmiert werden soll), können die Roboter bereits jetzt schon während des Spiels per Drag & Drop bewegt werden ... jedoch nur in gültige Richtungen, ansonsten springen sie zur Ausgangsposition zurück.

Ein weiteres Beispiel klärt den Spielgedanken und das Regelwerk:



Der orangefarbene Roboter oben links wird nach unten und dann nach rechts gezogen. Der LunarBot kann danach nach oben, links und schließlich nach unten ins Zielfeld gezogen werden.

Es stehen jetzt bereits schon einige Level zur Verfügung, die Du mit Rechtsklick auf die Welt (z.B. Space02) laden kannst.



Aufgabe:

Vervollständige das Greenfoot-Szenario so, dass das Spiel entsprechend den obigen Angaben umgesetzt wird. Sprich: Der LunarBot muss in der Lage sein, über Tastatursteuerung ans Ziel zu gelangen ... mit folgenden Auflagen:

- Ein Klick auf RUN startet das Spiel
- Die Pfeiltasten steuern, wohin er sich bewegt (right, left, up oder down)
- Wird eine Pfeiltaste gedrückt, darf der LunarBot nur in eine Richtung ziehen, in der er von einem Roboter „gebremst“ wird (nicht vom Spielfeldrand). Er muss in einem Schritt bis unmittelbar vor den Roboter ziehen und stehen bleiben.

Dazu ist nur die Methode act() des LunarBots anzupassen bzw. umzuschreiben! Achte darauf, Deinen Quelltext zu kommentieren!

Material: Methoden des LunarBots:

public boolean isNextToRobotOnLeft()	// true, wenn auf linkem Nachbarfeld ein Roboter ist
public boolean isNextToRobotOnRight()	// true, wenn auf rechtem Nachbarfeld ein Roboter ist
public boolean isNextToRobotOnUp()	// true, wenn auf oberem Nachbarfeld ein Roboter ist
public boolean isNextToRobotOnDown()	// true, wenn auf unterem Nachbarfeld ein Roboter ist
public boolean isAnyRobotOnDown()	// true, wenn irgendwo nach unten noch ein Roboter ist
public boolean isAnyRobotOnUp()	// true, wenn irgendwo nach oben noch ein Roboter ist
public boolean isAnyRobotOnLeft()	// true, wenn irgendwo nach links noch ein Roboter ist
public boolean isAnyRobotOnRight()	// true, wenn irgendwo nach rechts noch ein Roboter ist
public void moveOneStepLeft()	// gehe ein Feld nach links
public void moveOneStepRight()	// gehe ein Feld nach rechts
public void moveOneStepUp()	// gehe ein Feld nach oben
public void moveOneStepDown()	// gehe ein Feld nach unten

weitere Methoden, die Du benötigen wirst:

public boolean Greenfoot.isKeyDown("left")	// true, wenn linke Pfeiltaste gedrückt wurde
public boolean Greenfoot.isKeyDown("right")	// true, wenn rechte Pfeiltaste gedrückt wurde
public boolean Greenfoot.isKeyDown("up")	// true, wenn Pfeiltaste nach oben gedrückt wurde
public boolean Greenfoot.isKeyDown("down")	// true, wenn Pfeiltaste nach unten gedrückt wurde

Zusatzaufgabe:

Schaue Dir an, wie die Level erstellt wurden. Lege neue Level an, indem Du alte Level kopierst und entsprechend anpasst. Denke daran, die Klassen umzubenennen.