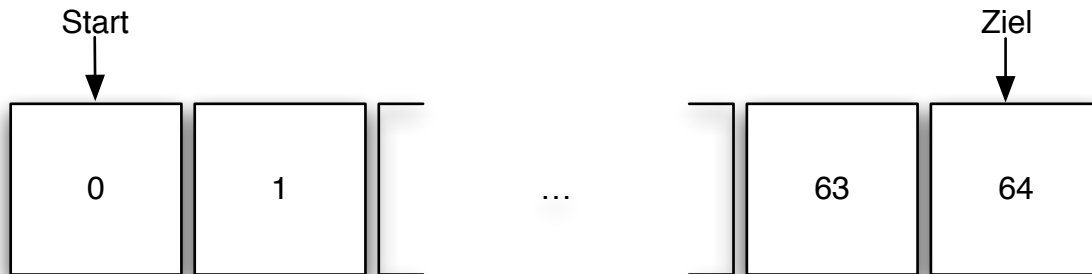


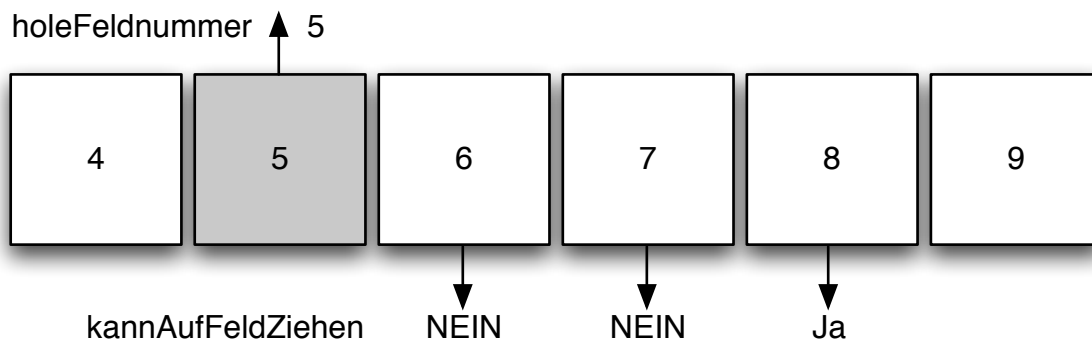
Software-Challenge 2010

Befehlsübersicht für Spieler

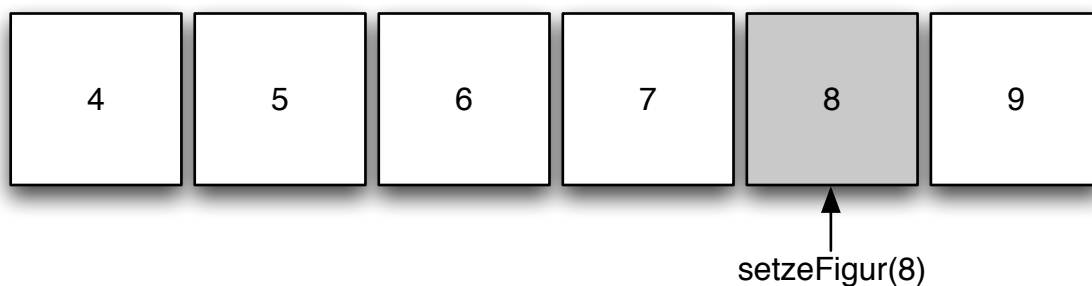
Bewegung



Das Spielfeld von Hase und Igel ist durchgehend von 0 bis 64 nummeriert, wobei 0 den Startpunkt und 64 das Ziel beschreibt.



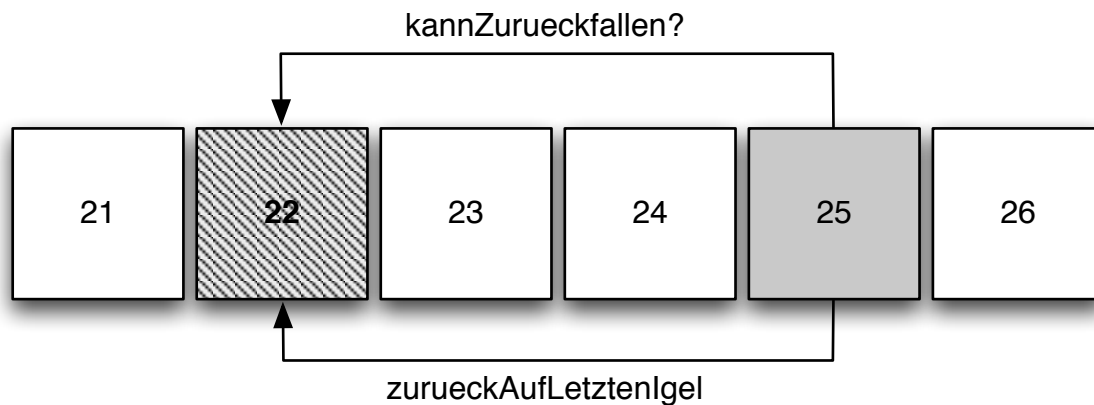
`holeFeldnummer()` gibt die Position der eigenen Spielfigur zurück. In dem Beispiel oben steht der Hase auf Feld 5. Mit `kannAufFeldZiehen(feldNummer)` überprüft man, ob die Spielfigur genügend Karotten hat, um ein Feld zu erreichen, und ob der Zug regelkonform ist. Im Beispiel ist nur das Feld 8 begehbar.



Mit dem Befehl `setzeFigur(feldNummer)` bewegt man den Hasen auf ein neues Feld.

Das Beispiel in Java:

```
int position = eigenerSpieler.holeFeldNummer(); // Hase auf 5
int naechstePosition = position;
while (!eigenerSpieler.kannAufFeldZiehen(naechstePosition)) {
    naechstePosition++; // 5 nicht, 6 nicht, 7 nicht, aber...
}
eigenerSpieler.setzeFigur(naechstePosition); // 8 ist frei!
```



Zu jedem Zeitpunkt kann der Hase mit `kannZurueckfallen()` überprüfen, ob der letzte Igel hinter ihm frei ist. Ist der Igel unbesetzt, kann man sich mit `zurueckAufLetztenIgel()` zurückfallen lassen.

Es gibt Situationen, in denen man keinen Zug ausspielen kann; dies wird mit `mussAussetzen()` überprüft. Manchmal hilft halt kein Weg drum herum! `setzeAus()` hilft.

Das Beispiele in Java:

```
if (eigenerSpieler.kannZurueckfallen()) {
    eigenerSpieler.zurueckAufLetztenIgel();
}
// ...
if (eigenerSpieler.mussAussetzen()) {
    eigenerSpieler.setzeAus();
}
```

Beachte: Auf welcher Feldnummer du nach dem zurückfallen landest, musst du selbst herausfinden!

Ernährung

Das Ziel darf nur betreten, wer ordentlich gefressen hat!

`holeSalatAnzahl()` gibt Aufschluss darüber, wie viele Salate noch verspeist werden müssen. Ob man gerade einen Salat fressen kann, überprüft man mit `kannSalatFressen()`, und mit `frissSalat()` schreitet man sodann zur Tat.

Das Beispiel in Java:

```
if (eigenerSpieler.holeSalatAnzahl() > 0 &&
    eigenerSpieler.kannSalatFressen())
    eigenerSpieler.frissSalat();
// Wir müssen noch 3 Salate fressen, stehen aber auf einem Salatfeld!
// *mampf* - Jetscht' sind's nur noch zwei!
```

Karotten sind für die Bewegung des Spielers wichtig, allerdings darf man zum Ende hin nur noch wenige Karotten auf der Hand haben. Deswegen kann man auf Karottenfeldern sowohl Karotten abgeben, als auch aufnehmen:

```
int karotten = eigenerSpieler.holeKarottenAnzahl(); // 23
// ...
if (eigenerSpieler.kannKarottenAbgeben()) {
    eigenerSpieler.gibKarottenAb(); // 13 Karotten übrig
} // oder...
if (eigenerSpieler.kannKarottenAufnehmen()) {
    eigenerSpieler.nimmKarottenAuf(); // 33 Karotten übrig
}
```

Beachte: Karotten werden immer in 10er Paketen gehandelt!

Der Joker

Mit `holeHasenjoker()` bekommt man alle Hasenjoker aufgelistet, die der Spieler zur Zeit auf der Hand hält. Wenn man einen dieser Joker ausspielen kann, gibt `kannHasenjokerSpielen(joker)` `TRUE` zurück; mit `spieleHasenjoker(joker)` spielt man den Joker dann aus.

Ob man einen speziellen Hasenjoker noch auf der Hand hat, kann man mit `hatHasenjokerTyp(joker)` überprüfen.

Beispiel I:

```
List<Hasenjoker> joker = eigenerSpieler.holeHasenjoker();
for (int i = 0; i < joker.size(); i++) {
    if (eigenerSpieler.kannHasenjokerSpielen(joker.get(i)) {
        eigenerSpieler.spieleHasenjoker(joker.get(i));
        break;
    }
}
```

Beispiel II:

```
if (eigenerSpieler.hatHasenjokerTyp(Hasenjoker.RUECKE_VOR)) {
    eigenerSpieler.spieleHasenjoker(Hasenjoker.RUECKE_VOR);
} else if (eigenerSpieler.hatHasenjokerTyp(Hasenjoker.NIMM_ODER_GIB_20_KAROTTEN)) {
    eigenerSpieler.spieleHasenjoker(Hasenjoker.NIMM_ODER_GIB_20_KAROTTEN);
} else if (eigenerSpieler.hatHasenjokerTyp(Hasenjoker.FALLE_ZURUECK)) {
    eigenerSpieler.spieleHasenjoker(Hasenjoker.FALLE_ZURUECK);
} else if (eigenerSpieler.hatHasenjokerTyp(Hasenjoker.FRISS_SALAT)) {
    eigenerSpieler.spieleHasenjoker(Hasenjoker.FRISS_SALAT);
}
```

Sonstiges

Hase und Igel unterscheidet zwischen dem roten und den blauen Spieler. Rot hat den ersten Zug. Welche Farbe man selbst besitzt kann über `holeSpielerfarbe()` herausgefunden werden.

Beispiel:

```
if (Spielerfarbe.ROT.equals(eigenerSpieler.holeSpielerfarbe())) {
    // TODO Moon Walk auf dem Spielfeld!
}
```