

Junioraufgabe 2

Lösungsidee

Es werden Objekte der Klasse *Lamp* in einem Gitter angeordnet, dessen Größe von dem Benutzer bestimmt werden kann. Nachdem die anfänglichen Einstellungen getroffen sind (Feldgröße, Lampengröße, Spielmodus) soll der Spieler in der Lage sein vor Beginn des eigentlichen Spiels Lampen an und aus zu schalten. Wenn der Spieler mit seiner Konfiguration der Lampen zufrieden ist drückt er einen Button und das Spiel beginnt. Je nachdem welchen Modus der Spieler gewählt hat werden entweder die Lampen die angeklickt werden und ihre Nachbarn (oben, unten, links, rechts) umgeschaltet, oder die Lampe und alle Lampen links und über ihr.

Eine Lampe hat eine Methode welche ihren Zustand umstellen kann, welche aber auch ein „passthrough“ parameter übergibt welches der Lampe sagt, ob sie den Zustandswechsel an ihre Nachbarn (oder eine andere Menge von Lampen, je nach Modus) übergeben soll oder nicht.

Sobald alle Lampen aktiviert wurden kann das Spiel (mit den jetzigen Einstellungen) wiederholt werden.

Umsetzung

Das Programm wurde in Java, mit Hilfe von Greenfoot geschrieben. Ich habe mir das Koordinatensystem von Greenfoot zu nutze gemacht um die `Actor#getNeighbours` Methode zu nutzen welche mir die benachbarten Lampen gibt, um deren Zustand zu wechseln. Im „Spezialmodus“ benutze ich zwei for-Schleifen um über alle Lampen in der jetzigen Welt zu gehen und die Lampen, die über und links von der angeklickten liegen umzuschalten.

Um den Zustand der Lampen zu speichern benutze ich ein Enum welches in der Lampe als Subclass vorhanden ist. Dieses Enum speichert außerdem das Farbschema der Lampe.

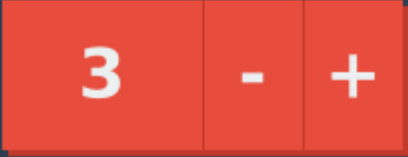
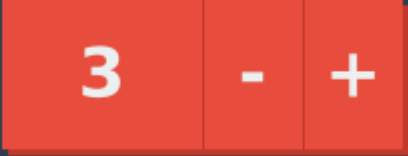



Des Weiteren ist ein Button am unteren Ende der Welt zu finden welcher, abhängig von der derzeitigen Spielphase das Spiel startet, oder zurücksetzt (entweder weil das Level zu schwer ist, oder weil man gewonnen hat). Dieser Zustand wird wieder einmal in einem Enum im Button gespeichert welcher dazu (wider) das Farbschema definiert.

Bevor aber das eigentliche Spiel beginnt wird man von einem Einstellungsmenü begrüßt welches einem die Möglichkeit gibt die Anzahl der Horizontalen und Vertikalen Lampen, und die Größe der Lampen zu definieren. Um das möglich zu machen benutze ich eine selbstgeschriebene Klasse namens `NumInput` welche einfach die derzeitig gewählte Zahl mit einem + und - Button daneben anzeigt. Danach wird das Spiel mit einem der zwei Buttons gestartet (welche den Modus bestimmen).

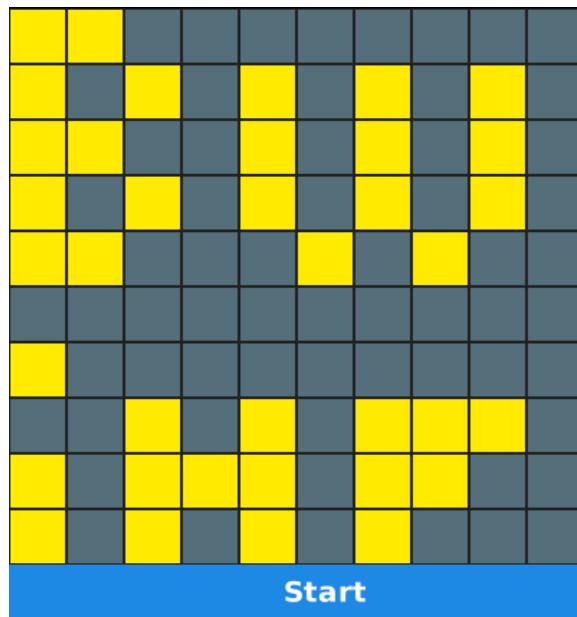
Für jedem dem es noch nicht aufgefallen ist: Alle Grafiken sind komplett generiert.

Beispiele

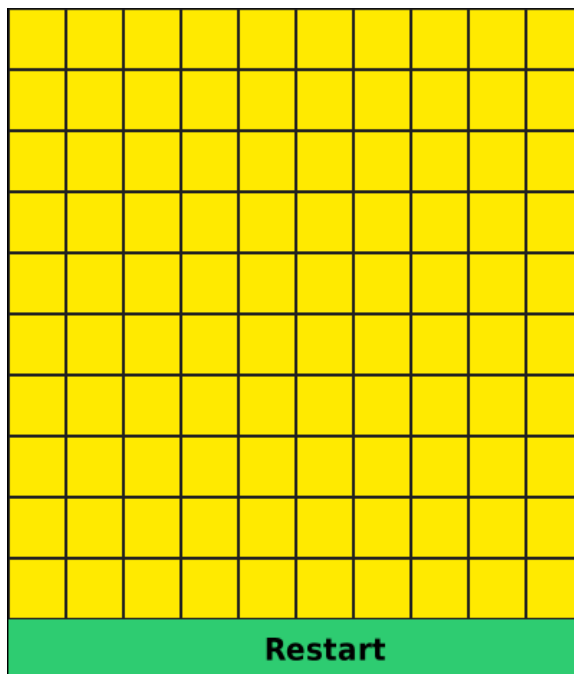
Das Menü

	Horizontale Anzahl an Lampen
	Vertikale Anzahl an Lampen
	Größe der Lampen
	Der „normale“ Modus
	Der „spezielle“ Modus

So könnte ein Level vor dem Start aussehen



So sieht ein Sieg aus



Wichtige Codebeispiele

Diese Methode wird benutzt um die Lampen umzuschalten.

```
public void updateLamp(boolean shouldTravel) {
    this.state = this.state == State.OFF ? State.ON : State.OFF;
    if(shouldTravel) {
        if(Game.special) {
            this.getWorld().getObjects(Lamp.class).forEach((lamp) -> {
                if(lamp.getX() <= this.getX() && lamp.getY() <= this.getY() && lamp != this) lamp.updateLamp(false);
            });
        } else {
            List<Lamp> neighbours = getNeighbours(1, false, Lamp.class);
            for(Lamp lamp : neighbours) {
                lamp.updateLamp(false);
            }
        }
    }
    updateImage();
}
```

Verwendung

WICHTIG: Dieses Projekt benutzt Funktionen aus Java 8 und ist deswegen nicht kompatibel mit Java 7 oder darunter!

Da dieses Programm Greenfoot verwendet kann einfach die fertige .jar Datei ausgeführt werden und das Spiel funktioniert. Im Menü repräsentieren, wie oben schon beschrieben, der erste NumInput die horizontale und der zweite die vertikale Anzahl an Lampen. Der dritte bestimmt die Größe. Dann einfach auf „Normal“ oder „Spezial“ klicken und der „Spaß“ kann los gehen!