

Kapitel 14.3: Sortierung – Tetris

In Tetris ist es so, dass eine Reihe gelöscht wird, sobald sie komplett mit Blöcken gefüllt ist. Darüber liegende Blöcke fallen dann automatisch eine Reihe nach unten. Achtung! Lücken bleiben bestehen! Das hört sich kompliziert an, aber wenn wir das Problem wieder in kleine Einzelprobleme zerlegen, dann ist es gar nicht so schwer!

Sortieren

Du brauchst erstmal eine Funktion `delete_full_rows()`, die du aufrufst, wenn der aktive Block inaktiv gesetzt wird, weil er entweder auf dem Boden oder einem anderen Block gelandet ist. In dieser neuen Funktion sortierst du zunächst die `inactive_blocks` nach ihrem y-Wert. Rückwärts! Die Blöcke ganz unten sollen in der Liste also ganz vorne stehen. Der x-Wert spielt hier keine große Rolle, aber wenn du möchtest, kannst du als 2. Sortierkriterium den x Wert nehmen. Stelle sicher, dass deine Sortierung auch funktioniert (Debug und/oder Print)!

Eine volle Reihe erkennen

Wie erkennt man, ob eine Reihe voll ist und keine Lücken mehr hat? Eigentlich ist das gar nicht so schwer! Du brauchst eine Methode `find_full_row()`, welche nach dem Sortieren der `inactive_blocks` in der Methode `delete_full_rows()` aufgerufen wird. Du weißt, wie viele Blöcke in eine Reihe passen: Das steht im Attribut `blocks_per_row` der Klasse `Board`. Wenn du jetzt durch die nach y-Werten sortierte Liste iterierst und dir merkst, wie viele Blöcke den gleichen y-Wert haben, dann kannst du den y-Wert der ersten gefundenen vollen Reihe zurückgeben, ansonsten -1. Du möchtest also erst einmal nur wissen, ob es mindestens eine volle Reihe gibt. Wenn du keine volle Reihe gefunden hast, bist du hier fertig!

Eine volle Reihe löschen

Wenn du eine volle Reihe gefunden hast, dann müssen die Blöcke aus dieser Reihe gelöscht werden. Da wir den y-Wert kennen, ist das ja nicht so schwer. Du schreibst eine neue Funktion `delete_row(y)`, der du den in `find_full_row()` gefundenen y-Wert als Parameter mitgibst. In der Funktion iterierst du über die Liste der inaktiven Blöcke und fügst alle Blöcke, mit dem übergebenen y-Wert in eine neue temporäre Liste.

Achtung: Du kannst die Blöcke nicht direkt aus der Liste löschen, weil die Liste durch das Löschen verändert wird und die for-Schleife dann nicht mehr richtig funktioniert.

Jetzt iterierst du über die temporäre Liste und löschst alle Blöcke aus dieser Liste in der Liste der inaktiven Blöcke.

Blöcke eine Reihe nach unten fallen lassen

Nachdem die Reihe jetzt gelöscht ist, müssen alle Blöcke, deren y-Wert kleiner ist als der gefundene y-Wert um `block_size` nach unten fallen. Implementiere das in einer Funktion `drop_inactive_blocks(y)`.

Alle vollen Reihen löschen

In der Funktion `delete_full_rows` müssen wir diese 3 Schritte so lange wiederholen, bis es keine vollen Reihen mehr gibt:

- eine volle Reihe finde
- die Blöcke in dieser Reihe lösche
- die darüber liegenden Blöcke um eine Reihe nach unten fallen lassen

Im Moment kann man natürlich nur jeweils eine Reihe voll machen, aber das wird sich schon bald ändern.

Super! So langsam sieht es doch fast schon aus, wie Tetris 😊.