

# TakeASquare

## Einleitung

Take-a-Square („Zieh' ein Quadrat!“) ist ein mathematisches Strategiespiel für zwei SpielerInnen. Beginnend mit einer positiven, ganzen Zahl kann jede(r) SpielerIn davon abwechselnd das Quadrat einer ganzen Zahl abziehen, solange ein Wert größer gleich Null übrigbleibt. Normalerweise gewinnt jene(r) SpielerIn, der/die die letzte Subtraktion machen kann, also genau auf Null kommt. Hier soll aber die “misère”-Variante implementiert werden: Wer den letzten Zug machen muss, verliert!

## Aufgabe

Schreiben Sie ein Programm TakeASquare, das zwei SpielerInnen die Möglichkeit bietet, auf der Konsole gegeneinander anzutreten. Es muss folgenden Anforderungen genügen:

- Die Startzahl wird mittels Zufallsgenerator ermittelt (aus dem Bereich 0-200)
- Die Zahl wird grafisch mittels vertikaler Striche angezeigt (wie Streichhölzer). Die Darstellung muss exakt wie im Beispiel gezeigt erfolgen, also:
  - Jeweils fünf Striche werden durch ein Leerzeichen getrennt.
  - Die Anzahl wird in derselben Zeile ausgegeben.
- Das Programm muss folgende Falscheingaben abfangen:
  - Eine zu große Zahl soll abgezogen werden.
  - Die eingegebene Zahl ist kein Quadrat einer ganzen Zahl.
  - Es wurde eine negative Zahl eingegeben.
- Nach einer Falscheingabe wird der/die SpielerIn über das jeweilige Problem informiert (Was war genau falsch?) und der/die SpielerIn wird solange nach einer Eingabe gefragt, bis diese gültig ist.
- Der/die SpielerIn, welche(r) das letzte “Streichholz” ziehen muss, verliert!
- Ab dem Zeitpunkt, wo der Spielausgang eindeutig determiniert ist, soll das Spiel enden.
- Die Spieler werden über den Spielausgang informiert.

## Beispiele:

In diesem Beispiel gewinnt SpielerIn 2, da SpielerIn 1 das letzte Streichholz ziehen **muss**.

```
Willkommen!  
Zwei SpielerInnen ziehen abwechselnd Hoelzer von einem Haufen.  
Jene(r) SpielerIn, der/die das letzte Holz zieht, verliert.  
Die jeweils gezogene Zahl muss das Quadrat einer positiven ganzen Zahl  
sein (1, 4, 9, 16 ...)  
SpielerIn 1 beginnt!  
||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| || (42 Hoelzer uebrig)
```

generated by [haroopad](#)

```
SpielerIn 1: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{4}

||||| ||||| ||||| ||||| ||||| ||||| ||||| ||| (38 Hoelzer uebrig)
SpielerIn 2: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{4}

||||| ||||| ||||| ||||| ||||| ||||| |||| (34 Hoelzer uebrig)
SpielerIn 1: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{9}

||||| ||||| ||||| ||||| ||||| (25 Hoelzer uebrig)
SpielerIn 2: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{16}

||||| |||| (9 Hoelzer uebrig)
SpielerIn 1: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{4}

||||| (5 Hoelzer uebrig)
SpielerIn 2: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{4}

SpielerIn 2 gewinnt!
```

Dieses Beispiel ist kein vollständiges Spiel, aber zeigt die Behandlung von Falscheingaben.

```
||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| (40 Hoelzer uebrig)
SpielerIn 1: Wieviele Hoelzer moechtest Du wegnehmen? \stdin{-5}
Die Zahl muss groesser als 0 sein!
Neuer Versuch! \stdin{999}
Zu gross!
Neuer Versuch! \stdin{3}
Kein Quadrat!
Neuer Versuch! \stdin{6}
Kein Quadrat!
Neuer Versuch! \stdin{7}
Kein Quadrat!
Neuer Versuch! \stdin{9}

||||| ||||| ||||| ||||| ||||| ||||| ||| (31 Hoelzer uebrig)
```