

# TerraTactician Expandoria

[terratactician-expandoria/javabot: Official java bot CLI. - Codeberg.org](https://codeberg.org/terratactician-expandoria/javabot)

## Installation

- compile bot `javac -cp tte-bot.jar bots/ExampleBot.java`
  - start bot `java -jar tte-bot.jar --challenge ExampleBot`
  - `C:\\Program Files (x86)\\terratactician-expandoria\\terratactician-expandoria.exe`
  - `terratactician-expandoria --bot localhost:7738 --challenge`
- [Overview \(app API\)](#) [page.codeberg.terratactician\\_expandoria.bots \(app API\)](#)

## Tiles

Tile	Produces	Dependent on	Formula
Bienen	Nahrung	Wald (3) Weizen (5)	$x = \ln((f + 1) * (w + 1))$ <ul style="list-style-type: none"><li>• <math>f</math> : Anzahl Wald-Tiles in Radius 3</li><li>• <math>w</math> : Anzahl Weizen-Tiles in Radius 5</li><li>• <math>x</math> : Produktionsrate</li></ul>
Weizen	Nahrung	Weizen (v)	$x = e^{-\left(\frac{w-9}{s}\right)^2} * 2.4 + 1$ <ul style="list-style-type: none"><li>• <math>w</math> : Anzahl Weizen-Tiles die direkt oder indirekt über andere Wheat-Tiles mit diesem verbunden sind</li><li>• <math>s : \begin{cases} 4 &amp; w \leq 9 \\ -2.5 &amp; \text{sonst} \end{cases}</math></li><li>• <math>x</math> : Produktionsrate</li></ul>
Haus	Geld	Booster (3)	$x = 2 * b$ <ul style="list-style-type: none"><li>• <math>b</math> : Stärkster Boost durch ein Booster-Tile im Radius von 3</li><li>• <math>x</math> : Produktionsrate</li></ul>
Doppelhaus	Geld	Booster (3), Haus/Doppelhaus (1)	$x = \left(\frac{- h - 3  + 3}{3/2} + 2\right) * b$ <ul style="list-style-type: none"><li>• <math>h</math> : Summe aller Haus Tiles und Doppelhaus Tiles in der direkten Nachbarschaft</li><li>• <math>b</math> : Stärkster Boost durch ein Booster-Tile im Radius von 3</li><li>• <math>x</math> : Produktionsrate</li></ul>

Tile	Produces	Dependent on	Formula
Marktplatz	Geld	Haus (5) Doppelhaus (5) Nahrung (5) Materialiern (3)	$e = \min\{1, (h + d * 2) * 0.2 + 0.5\}$ $x = (f * f\_rate + m * m\_rate) * e$ $f\_new = -f * f\_rate$ $m\_new = -m * m\_rate$ <ul style="list-style-type: none"> <li>• <math>h</math> : Anzahl Haus-Tiles in einem Radius von 5</li> <li>• <math>d</math> : Anzahl Doppelhaus-Tiles in einem Radius von 5</li> <li>• <math>e</math> : Handelseffizienz des Marktplatzes</li> <li>• <math>f</math> : Summe der produzierten Nahrung in einem Radius von 5</li> <li>• <math>m</math> : Summe der produzierten Materialien in einem Radius von 3</li> <li>• <math>f\_rate</math> : Anteil Nahrung, der verkauft werden soll</li> <li>• <math>m\_rate</math> : Anteil Materialien, die verkauft werden sollen</li> <li>• <math>f\_new</math> : Abgezogenen Nahrung an der Produktionsrate in der Umgebung</li> <li>• <math>m\_new</math> : Abgezogenen Materialien an der Produktionsrate in der Umgebung</li> <li>• <math>x</math> : Produktionsrate</li> </ul>
Windmühle	Nahrung	Windmühlen (-3 v. Weizen)	$f\_reg = \min\{1, \frac{2}{m}\}$ $x = w * (1 - (f * 0.1))$ <ul style="list-style-type: none"> <li>• <math>f</math> : Anzahl Wald-Tiles in direkter Nachbarschaft</li> <li>• <math>m</math> : Anzahl Windmühlen in einem Radius von 3 für jedes Weizen Tile</li> <li>• <math>f\_reg</math> : Quotient über die Anzahl der Windmühlen in der Nachbarschaft, für jedes Weizen Tile</li> <li>• <math>w</math> : Summe aller <math>f\_reg</math> Werte in einem Radius von 3, der Weizen Tiles.</li> <li>• <math>x</math> : Produktionsrate</li> </ul>
Moai	Booster	einzigartige Tiles (2)	$x = 1 + 0.2 * \max\{0, d - 2\}$ <ul style="list-style-type: none"> <li>• <math>d</math> : Anzahl einzigartiger Tiles in einem Radius von 2</li> <li>• <math>x</math> : Stärke des Boosts</li> </ul>
Steinbruch	Materialien	Steinbruch-Level	$x = 5 * l$ $l \in \{0, 1, 2, 3\}$ <ul style="list-style-type: none"> <li>• <math>l</math> : Das Level des Stein-Tiles, dem der Steinbruch zugeordnet wurde</li> <li>• <math>x</math> : Produktionsrate</li> </ul>

Tile	Produces	Dependent on	Formula
Wald	Materialien	Wald (1)	$x = f * 0.5 + 1$ <ul style="list-style-type: none"> <li>• <math>f</math> : Anzahl Wälder in der direkten Nachbarschaft</li> <li>• <math>x</math> : Produktionsrate</li> </ul>
Gras			
Steine			

- hand completely random

## Strategy 1

- always place what gives most reward in least resource
- if target reached, switch to other resources
- if target can be reached with current rate switch to other resources

## Strategy 2

- ~~genetic algorithm for finding best policies~~
- ~~decision tree learning~~
- ~~reinforcement learning~~

## Actions

```

1  // page.codeberg.terratactician_expandoria.bots
2  // usually one action per tick
3  collectReward // always if possible
4
5  configureMarket // always if possible by reward
6
7  placeTile
8
9  redraw // always if possible? no costs resources / all tiles will be
    replaced
10
11  takeTile
12
13  // selectSlot // i dont think this is needed

```

# Environment

```
1  // page.codeberg.terratactician_expandoria.bots
2  getActionCount
3
4  getBuildArea
5
6  getHand
7
8  getMap
9
10 getRedrawCost
11
12 getRedrawTime
13
14 getResouces
15
16 getResourceRate
17
18 getRewards
19
20 getRoundTime
21
22 getTargetResources
```