

# Kalkulationsprogram ONI

Lastenheft nach DIN 69901-5

## Einführung

Das Programm "Kalkulationsprogram ONI" soll Spielern des Spiels Oxygen Not Included (ONI) bei Berechnungen helfen.

Diese werden genutzt, um die Effizienz von Spielprozessen zu maximieren.

## Beschreibung des Spiels:

Beschreibung auf Steam:

Oxygen Not Included is a space-colony simulation game. Deep inside an alien space rock your industrious crew will need to master science, overcome strange new lifeforms, and harness incredible space tech to survive, and possibly, thrive.

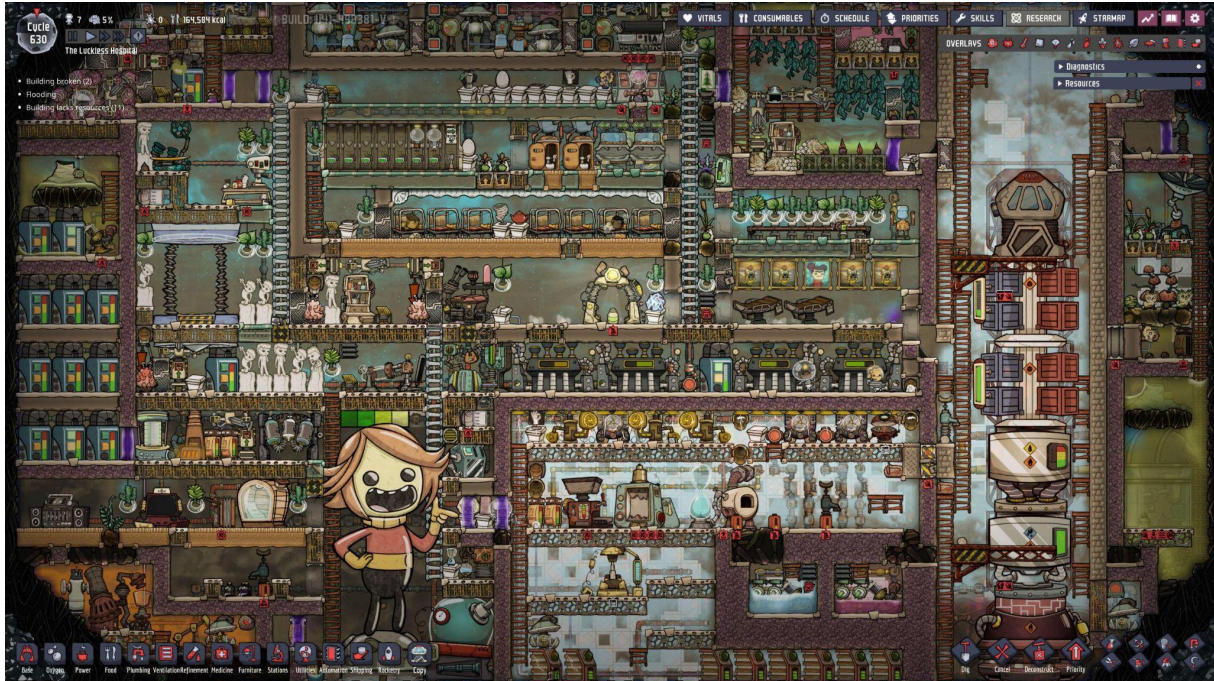
Beschreibung von ChatGPT:

Oxygen Not Included ist eine tiefgehende, technisch anspruchsvolle Kolonie-Simulation mit starkem Fokus auf physikalischen Prozessen und nachhaltigem Ressourcenmanagement. Erfolg beruht weniger auf schnellen Reaktionen als auf strategischer Planung und präzisiertem Systemdesign.

## Eine Starterbase:



## Eine Base nach vielen Stunden:



Folgende Webseite setzt unsere Berechnungen schon um:

<https://onicalc.com/>

Beispiel das im folgenden Kapitel erläutert wird:

<https://onicalc.com/buildings>

Weitere Informationen zum Spiel sind im Anhang zu finden.

## Beschreibung des Ist-Zustands

Die Anzahl von Produkten, die aus einem Prozess hervorgehen, müssen per Hand ausgerechnet werden. Im folgenden werden zwei mögliche Berechnungen beschrieben:

g: Gramm;      s: Sekunde

### Informationen die im Spiel verfügbar sind:

1x Terrarium verbraucht:      30 g Algen pro s  
   300 g Wasser pro s

und produziert:                      40 g Sauerstoff pro s  
   290 g verschmutztes Wasser pro s

1.

**Ich besitze drei Algenterrarien und möchte wissen, wieviel diese an Ressourcen verbrauchen und produzieren.**

Nun muss die Berechnung für 3x Terrarien umgestellt werden:

3x Terrarien verbrauchen:       $3 \cdot 30 \text{ g Algen pro s} = 90 \text{ g Algen pro s}$   
    $3 \cdot 300 \text{ g Wasser pro s} = 900 \text{ g Wasser pro s}$

und produzieren:                       $3 \cdot 40 \text{ g Sauerstoff pro s} = 120 \text{ g Sauerstoff pro s}$   
    $3 \cdot 290 \text{ g verschmutztes Wasser pro s}$   
    $= 870 \text{ g verschmutztes Wasser pro s}$

2.

**Ich brauche mindestens 2000 g Sauerstoff pro s und suche eine ganze Anzahl von Algenterrarien die ich mindestens brauche.**

x = gesuchte Anzahl an Terrarien

a = Menge von Sauerstoff die ein Terrarium produziert = 40 g Sauerstoff/s

b = Menge an gewünschtem Sauerstoff = 2000 g Sauerstoff/s

$x = a/b$

$$\rightarrow x \geq \frac{2000 \text{ g/s}}{40 \text{ g/s}}$$

$$\Leftrightarrow x \geq 49.8$$

Ich benötige mindestens 50 Algenterrarien um 2000 g Sauerstoff /s zu erreichen.

## Beschreibung des Soll-Konzepts

Für die beschriebenen Gleichungen soll ein Benutzer in der Lage sein,

1. eine von allen Maschinen, Tieren oder anderen Produzenten auswählen zu können,
2. auszuwählen, welche Variable er sucht (1. Menge von Verbrauch oder Produkten bei gegebener Anzahl der Maschinen oder 2. Anzahl der Maschinen bei gewollter Produktion),
3. gegebene Variablen einzugeben,
4. und das Ergebnis ausgegeben zu bekommen.

## Beschreibung von Schnittstellen

Mögliche Schnittstellen mit den Quellcode, um Informationen(\*1) der Umwandlung zu erhalten.

## Funktionale Anforderungen

Für die Umsetzung der Berechnung sind keine Vorgaben gegeben.

Bei der Grafik wird folgendes erwartet:

Für jede Ein- und Ausgabe gibt es ein Textfeld und eine Beschreibung des Feldes.

Bei der Eingabe der Maschine etc. ist aus einer Liste zu wählen, die nach Art kategorisiert und alphabetisch sortiert ist.

Ein graphisches Interface in einer Windows Form-App ist vorgegeben.

## Nichtfunktionale Anforderungen

### Benutzbarkeit

Die Software soll für Spieler nutzbar sein, die die Berechnungen kennen.

### Zuverlässigkeit

Die Software sollte durch Fehleingaben nicht zum Absturz gebracht werden können und Gründe für einen Fehler sollen angezeigt werden.

### Effizienz

Das Programm sollte nicht zu langsam sein.



**Änderbarkeit**

Die Software sollte gut dokumentiert sein, sodass auch andere Teams Änderungen vornehmen können oder die Teile wiederverwenden können.

**Übertragbarkeit**

Keine besonderen Vorgaben.

**Wartbarkeit**

Die Wartbarkeit soll durch gut strukturierten und kommentierten Code, sowie angemessene Dokumentation gegeben sein. Es soll einen Black Box-Test geben.

**Risikoakzeptanz**

Keine besonderen Vorgaben.

**Skizze des Entwicklungszyklus und der Systemarchitektur**

Die Software soll nach dem Wasserfallmodell entwickelt werden.

**Lieferumfang**

Geschriebener Code in C# in einem Visual Studio Projekt, Dokumentation, Testdokumentation.

**Abnahmekriterien**

Funktionsfähige Software, die den Anforderungen entspricht.

**Glossar**

ONI	Oxygen Not Included, Videospiel von Klei Entertainment
g	Gramm
s	Sekunde
Terrarium	Eine von vielen Maschinen, auf die die beschriebenen Berechnungen anwendbar sind.

## Referenzen

\*1 - siehe: (Seite 3) Informationen die im Spiel verfügbar sind

## Anhang

Das WIKI für das Spiel:

<https://oxygennotincluded.wiki.gg/>

Youtube Tutorials für das Spiel:

Playlist Oxygen Not Included - Tutorial Bites:

[https://www.youtube.com/watch?v=hKFcpmu5OP8&list=PLvfifJA8en\\_I5l4gedXfdFnca5o9igr2V&index=2](https://www.youtube.com/watch?v=hKFcpmu5OP8&list=PLvfifJA8en_I5l4gedXfdFnca5o9igr2V&index=2)