|  |
| --- |
| C.U.M Inc. |
| Cookie Clicker |
| Wiki |

|  |
| --- |
| Ansgar, Cristian, Linus, Timo & Wesley  5.2.2023 |

# Spiel Konzept

"Cookie Clicker" ist ein sogenannter "Idle Game" oder auch "Clicker Game", bei dem man virtuell Kekse produzieren muss, indem man auf einen Button klickt. Durch das Sammeln von Keksen kann man im Spiel Upgrades und neue Gegenstände freischalten, um die Keksproduktion zu beschleunigen.

# Upgrades

"Cookie Clicker" bietet die Möglichkeit mit den gesammelten Keksen Upgrades zu kaufen. Es gibt zwei Typen von Upgrades.

## Cursors

Der Kauf eines neuen Cursors erhöht die Anzahl an Keksen, die erzeugt werden, wenn man auf den Keks klickt. Je teurer der Cursor, desto mehr Kekse werden pro Klick erzeugt. Der zusätzliche Boost wird zu dem bestehenden Boost addiert.

|  |  |  |  |
| --- | --- | --- | --- |
| Cursor | Bedingung | Kosten | CPH |
|  | - | - | 1 |
|  | 5x Erste Karte in Besitz | 100 | 2 |
|  | 5x Zweite Karte in Besitz | 500 | 8 |
|  | 5x Dritte Karte in Besitz | 10.000 | 69 |
|  | 5x Vierte Karte in Besitz | 1.000.000 | 112 |
|  | 5x Fünfte Karte in Besitz | 10.000.000 | 1000 |

## Cards

Der Kauf einer neuen Card erhöht die Anzahl an Keksen, die pro Sekunde automatisch erzeugt werden. Je teurer die Card, desto mehr Kekse werden pro Sekunde erzeugt. Der zusätzliche Boost wird zu dem bestehenden Boost addiert.

|  |  |  |  |
| --- | --- | --- | --- |
| Card | Kosten | Kostenerhöhung pro Kauf | CPS |
|  | 15 | 1.15 | 0.1 |
|  | 100 | 1.15 | 1 |
|  | 1.100 | 1.15 | 8 |
|  | 12.000 | 1.15 | 47 |
|  | 130.000 | 1.15 | 260 |
|  | 1.400.000 | 1.15 | 1.400 |
|  | 20.000.000 | 1.15 | 7.800 |
|  | 330.000.000 | 1.15 | 44.000 |
|  | 5.100.000.000 | 1.15 | 260.000 |
|  | 75.000.000.000 | 1.15 | 5.000.000 |

# Spiel Erklärung

Cookie Clicker ist objektorientiert programmiert und besteht aus Klassen und Objekten.

## Klassen

Klassen sind der Blueprint, von dem ein Objekt erstellt werden kann. Klassen bestehen aus Attributen und Methoden.

### **MainCookie()**

Darstellung und Definition des großen klickbaren Cookies

### **CookieHitAnimation()**

Darstellung einer Animation, wenn man den MainCookie klickt

### **ScoreDisplay()**

Darstellung des Cookie Scores und der Cookies pro Sekunde

### **Card()**

Darstellung und Definition der im Shop kaufbaren Karten, welche die Cookies pro Sekunde erhöhen

### **Cursor()**

Darstellung und Definition der im Shop kaufbaren Cursor, welche die Cookies pro Klick erhöhen

### **Debuff()**

Darstellung und Definition der Debuff-Karten. Durch einen NPC wird eine zufällige Debuff-Karte gezogen, welche entweder die Anzahl der Cookies, den Wert für Cookies pro Sekunde oder den Wert für Cookies pro Klick verringert.

### **Player()**

Definiert die Spieler Werte für Cookies, Cookies pro Sekunde und Cookies pro klick

### **Music()**

Definiert einen Hintergrundmusik Start und Stopp Knopf

### **NPC()**

Darstellung und Definition eines nicht Spieler Charakters welcher in Kombination mit der Debuff Klasse Spielerwerte verringert

### **Jumpscare()**

Darstellung eines Jumpscares

### **GoldenCookie()**

Darstellung und Definition von einem von oben fallenden goldenen Cookie welcher, wenn rechtzeitig angeklickt, ein Event auslöst, welches für begrenzte Zeit den Wert für Cookies pro klick verdoppelt

## Objekte

Objekte werden als Instanz von einer Klasse erstellt. Sie besitzen jeweils dieselben Methoden und Attribute.

## Funktionen

Funktionen sind Codeabschnitte, die im laufe des Programms beliebig oft abgerufen werden können, um dieselbe Aktion auszuführen und um Code-Duplikation zu vermeiden.

## Main-Loop

Attribute, Methoden und Funktionen werden in einem „WHILE-Loop“ mithilfe von Event-Listenern (inputs) und Variablen ausgelöst.