

# Willkommen in deiner Bibliothek

Bei Roblox Story Games verschwinden Wörter, und die Spieler müssen die Lücken füllen! Programmiere in diesem einstündigen Projekt dein eigenes Story Game mit den kostenlosen Programmier- und Designwerkzeugen von Roblox. Stelle dein Programmier talent unter Beweis, um exklusive Abzeichen und Avatar-Gegenstände zu verdienen.

Schau dir das folgende Video an, um die Bibliothek zu sehen, in der du arbeiten wirst. Außerdem wird gezeigt, wie das Spiel abläuft.

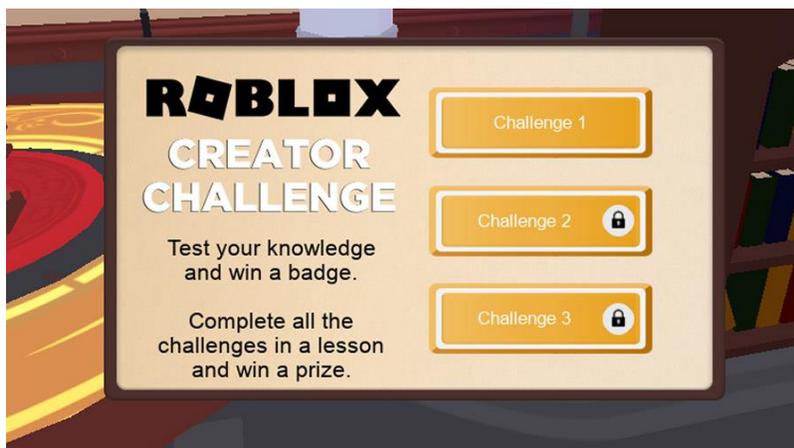


## Deine Reise durchs Programmieren

Um dein Story Game zu erschaffen, wirst du dich durch drei unterschiedliche Lektionen arbeiten. In jeder Lektion lernst du, wie du dein Spiel von Anfang bis Ende programmieren kannst.



Nachdem du eine Lektion abgeschlossen hast, kannst du Preise erhalten, indem du ein Quizspiel auf Roblox spielst. Wenn du deinen Preis erhalten hast, kannst du mit der nächsten Lektion weitermachen.



## Roblox Studio vorbereiten

Jedes Spiel, das du auf Roblox findest, wurde mit Roblox Studio hergestellt. Wenn du Roblox Studio bereits installiert hast, kannst du dich dort anmelden und unten auf dieser Seite auf „Weiter“ klicken, um fortzufahren.

1. Lade Roblox Studio herunter und installiere es.

[ROBLOX STUDIO HERUNTERLADEN](#)

2. Doppelklicke nach der Installation auf das Desktopsymbol (Windows) oder klicke auf das Docksymbol (Mac).

3. Gib im Anmeldefenster deinen Roblox-Benutzernamen und dein Passwort ein und klicke dann auf „Log In“, um dich anzumelden.

### Ein neues Konto erstellen

Wenn du Roblox noch nie verwendet hast, [klicke hier](#), um ein neues Konto zu erstellen. Denk an diese Tipps, um dein Konto zu schützen:

- **Teile niemals dein Passwort mit anderen**, nicht einmal mit Freunden, die du im echten Leben kennst.
- **Denke dir ein schwer zu erratendes Passwort aus** – Wenn dein Benutzername „bloxcool“ ist, sollte dein Passwort **auf keinen Fall** „bloxcool123“ lauten.
- **Roblox-Mitarbeiter werden dich niemals nach deinem Passwort fragen** – Sollte dich jemand danach fragen, melde diese Person über die Funktion [Verstoß melden](#).
- **Robux gibt es niemals gratis** – Traue keinen Spielern oder Internetseiten, die behaupten, dass sie Robux kostenlos beschaffen können!

Weitere Infos findest du auf der Hilfeseite für [Kontosicherheit](#).

## Die Story verfassen

Bevor du mit dem Programmieren anfängst, solltest du die Story schreiben, die du als Grundlage für dein Spiel verwenden möchtest. Du kannst dir ein beliebiges Thema aussuchen.

- Schreibe auf einem Stück Papier oder in einem Textprogramm. Deine Einleitung sollte aus 2 bis 3 Sätzen bestehen. Hier sind ein paar Ideen, um deine Fantasie anzuregen:
  - Der Traumurlaub deines besten Freundes
  - Was ein Zauberer zum Frühstück isst
  - Ein Superheld geht einkaufen

### Beispiel-Story

*In einem Baum auf einem Hügel lebt der mächtige Zauberer Nia. Zum Frühstück verspeist der Zauberer jeden Morgen eine riesige Schüssel voll honiggerösteter Echtenblätter.*

## Platzhalter erstellen

Um ein Wörterspiel aus der Story zu machen, löschst du Wörter und ersetzt diese mit Platzhaltern. Im Spiel werden die Spieler Fragen beantworten, um diese Platzhalter zu füllen. Nimm dir zu Beginn nur den ersten Satz vor.

1. Wähle im ersten Satz ein einzelnes Wort, das vom Spieler ersetzt werden soll. Dabei kann es sich um einen Namen, eine Handlung oder ein Hauptwort handeln.

### Wähle ein Wort

In einem Baum auf einem Hügel lebt der mächtige Zauberer Nia.

2. Ersetze das Wort mit einem Platzhalter für die Wortart, nach der du den Spieler fragen wirst.

Originalwort	Platzhalterwort
<i>In einem Baum auf einem Hügel lebt der mächtige Zauberer <b>Nia</b>.</i>	<i>In einem Baum auf einem Hügel lebt der mächtige Zauberer <b>name</b>.</i>

- Da du den Spieler eventuell nach mehr als nur einem Hauptwort oder Adjektiv fragen möchtest, solltest du die Platzhalter nummerieren.

Endgültiger Platzhalter
In einem Baum auf einem Hügel lebt der mächtige Zauberer name1.

Jetzt kannst du die Story in Programmiercode verwandeln!

## Template (Vorlage) öffnen

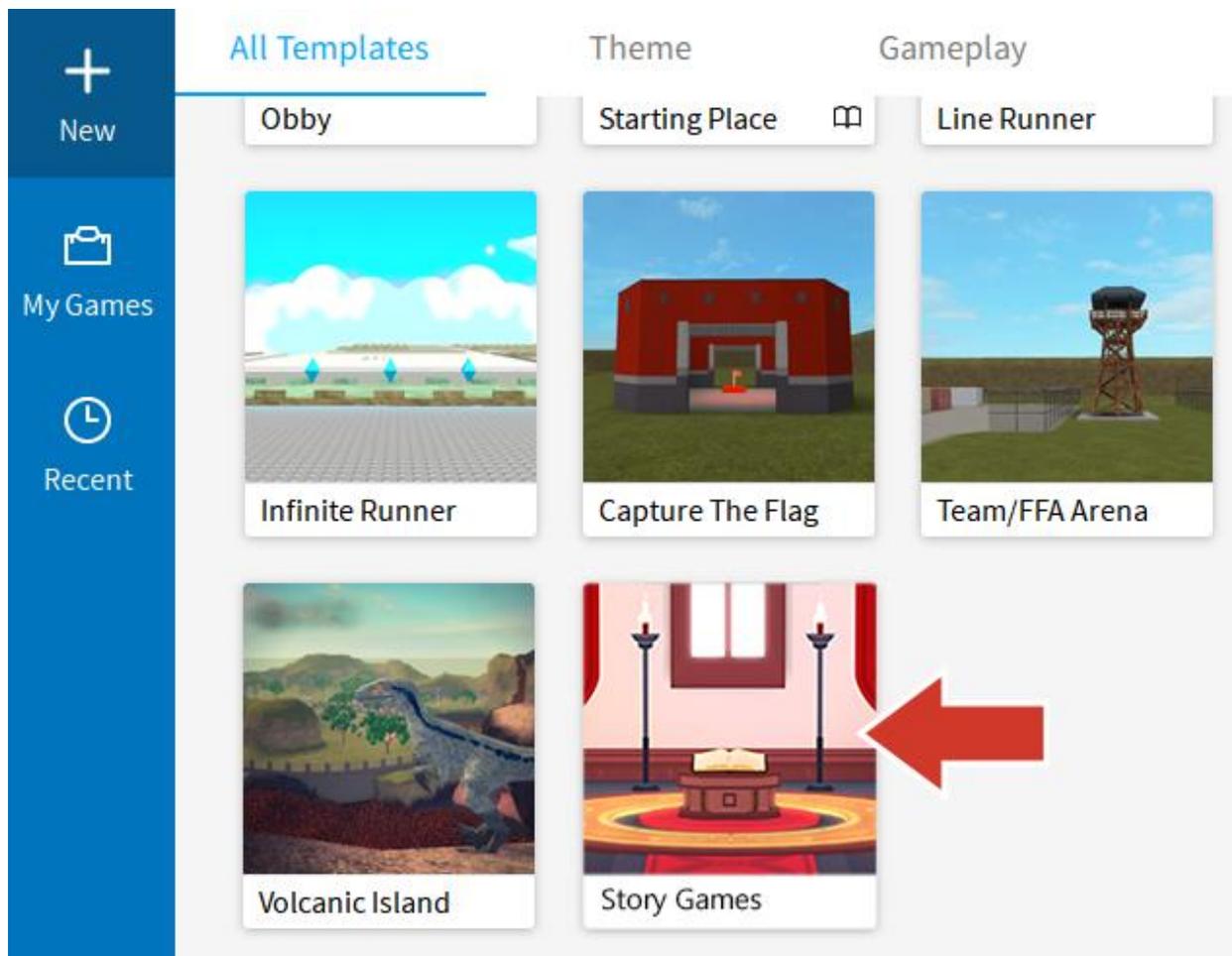
Das Spiel wird mit **Roblox Studio** erstellt. Es kann kostenlos verwendet werden, und du kannst Spiele umgehend für iPhone, Android, Xbox Live, PC, Mac und VR veröffentlichen.

- Öffne Roblox Studio**, indem du auf das blaue Desktopsymbol doppelklickst (Windows) oder indem du auf das Docksymbol klickst (Mac).
- Gib im Anmeldefenster deinen Roblox-Benutzernamen und dein Passwort ein und klicke dann auf **Log In**, um dich anzumelden.

## Öffne das Template

In Roblox Studio gibt es ein Template für **Story Games**, das bereits alles enthält, damit das Spiel funktionieren kann. Nur der Code für die tatsächliche Story muss von dir noch hinzugefügt werden. Templates sind vorgefertigte Welten, die du als Grundlagen für deine Spiele verwenden kannst.

- Klicke einmal auf das Template für **Story Games**.



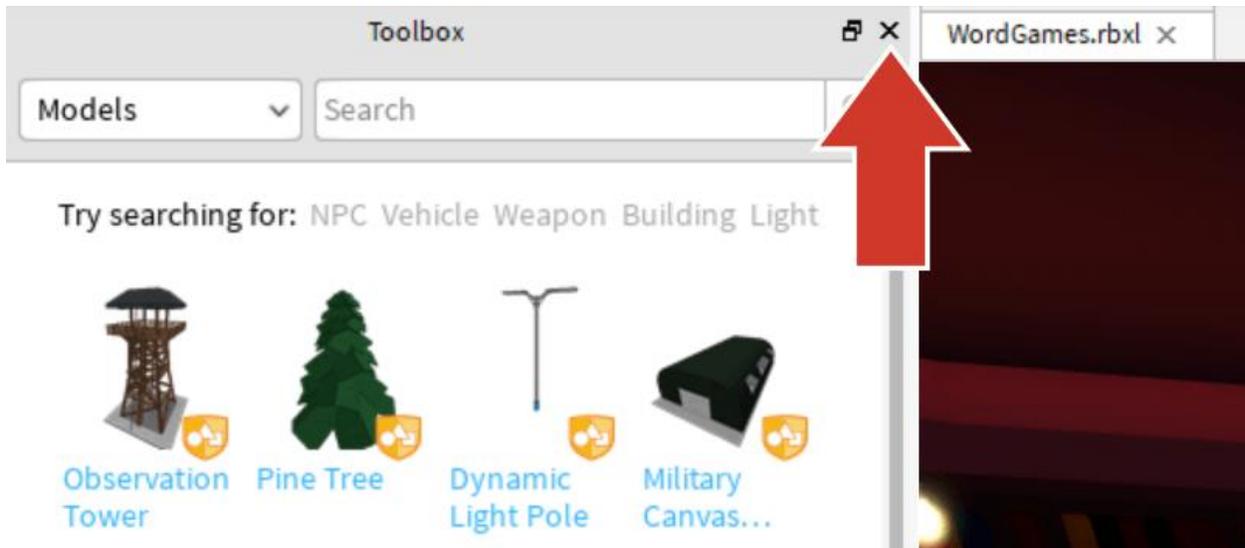
**Du siehst keine Templates?**

Solltest du keine Templates sehen, klicke auf **New** → **All Templates**.

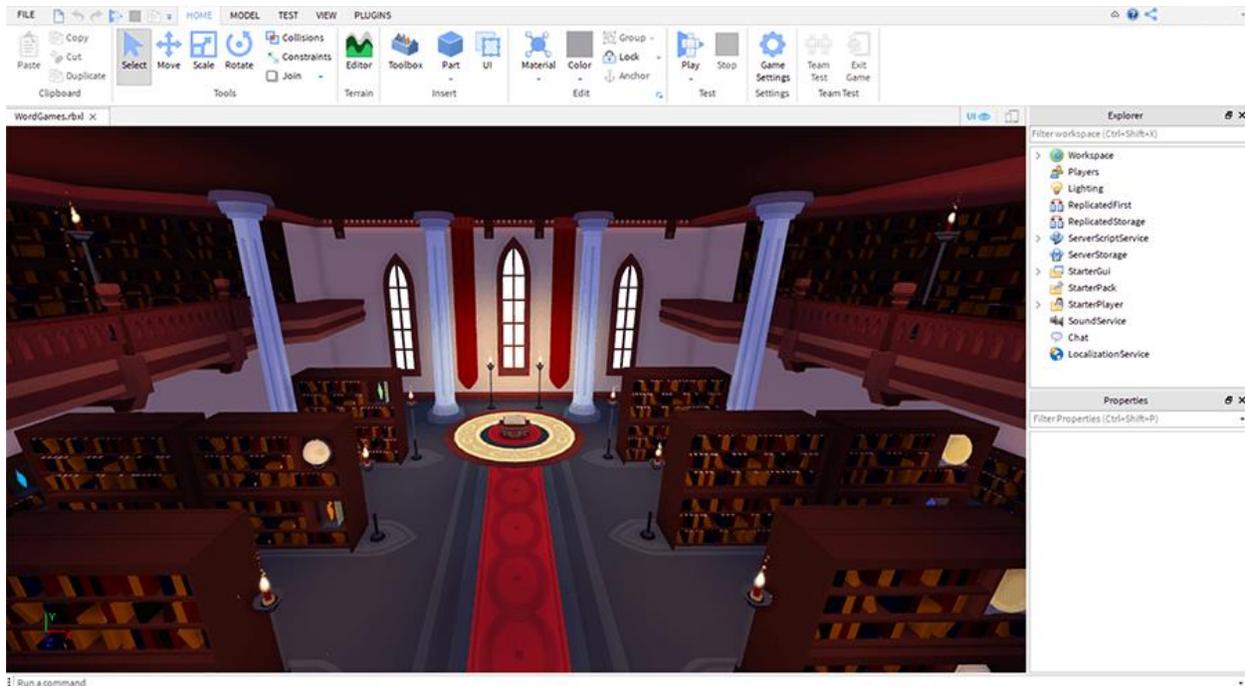
## Schließe nicht benötigte Fenster

Wenn du Roblox Studio zum ersten Mal startest, werden eventuell zusätzliche Fenster geöffnet, die du derzeit noch nicht benötigst. Wenn du diese Fenster schließt, hast du einen besseren Überblick über deine Arbeit.

1. Schließe alle Fenster auf der **linken** Seite von Studio, indem du auf **x** klickst.  
Sollte es keine Fenster geben, die geschlossen werden müssen, mache mit dem nächsten Schritt weiter.



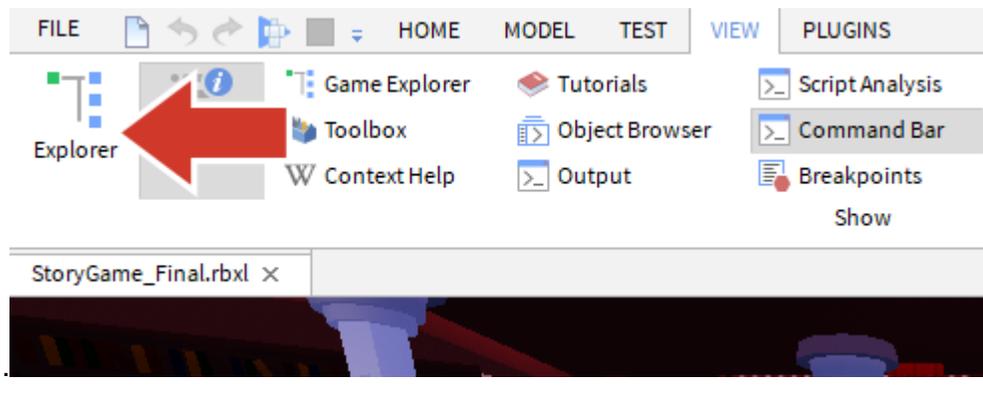
2. Lasse das Explorer-Fenster auf der rechten Seite geöffnet. Passe Studio so an, dass es wie unten gezeigt aussieht.



## Wenn du den Explorer nicht siehst

Sollte das Explorer-Fenster nicht zu sehen sein:

1. Wähle den Reiter **View** (Ansicht).
2. Klicke auf das **Explorer**-Symbol ganz links.

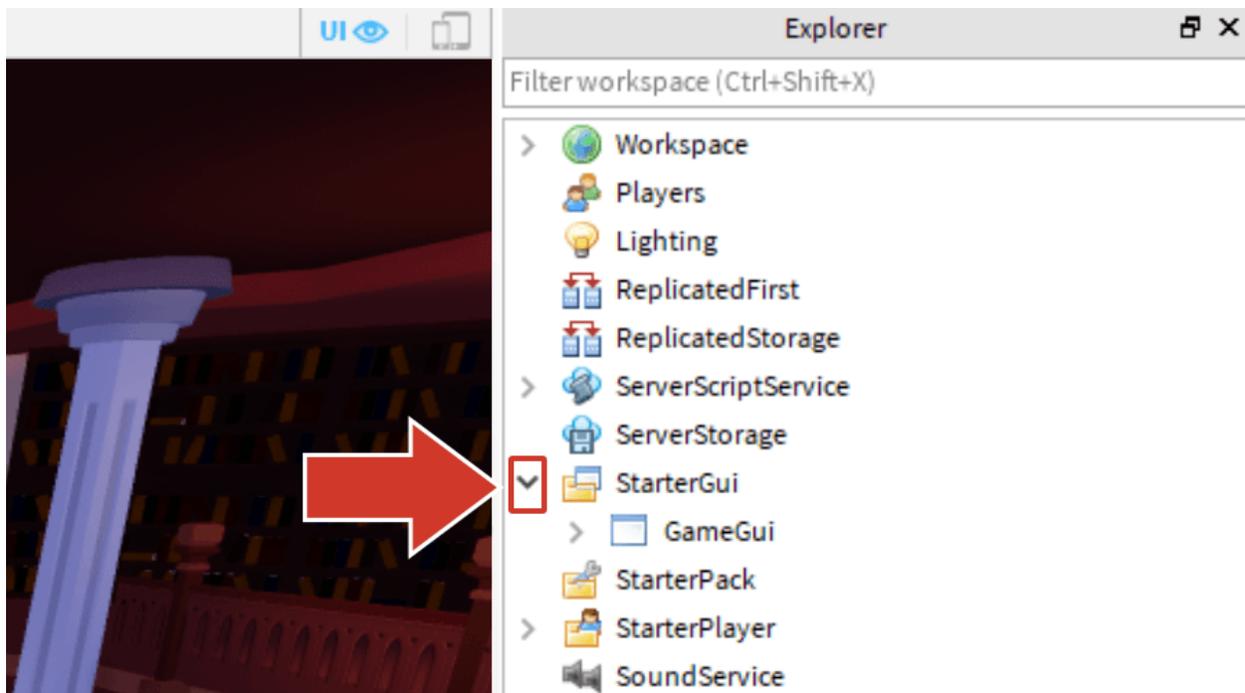


## Script öffnen

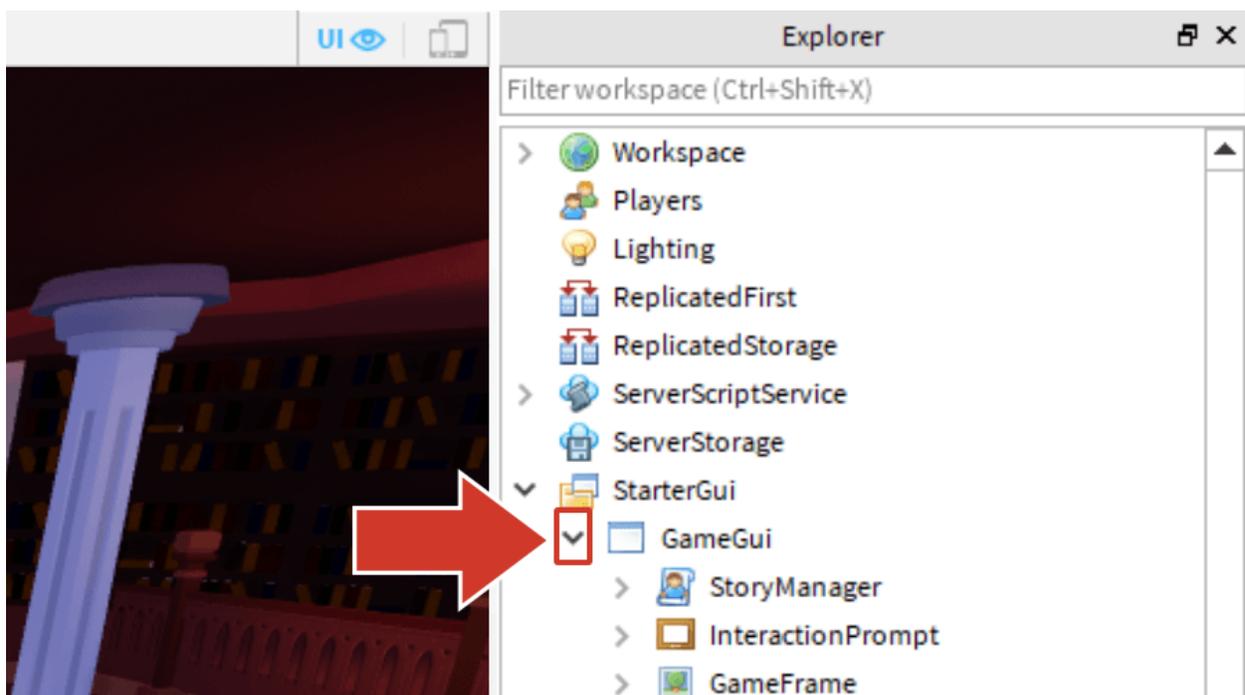
Bei Roblox wird Code mit der Programmiersprache **Lua** in **Scripts** getippt. Spiele verfügen oft über unterschiedliche Scripts für die verschiedenen Funktionen, die das Spiel ausführen soll. Im Bibliothek-Template gibt es bereits ein Script namens **StoryManager**. Diesem fügst du weiteren Code für dein Wörterspiel hinzu.

So findest du das **StoryManager**-Script:

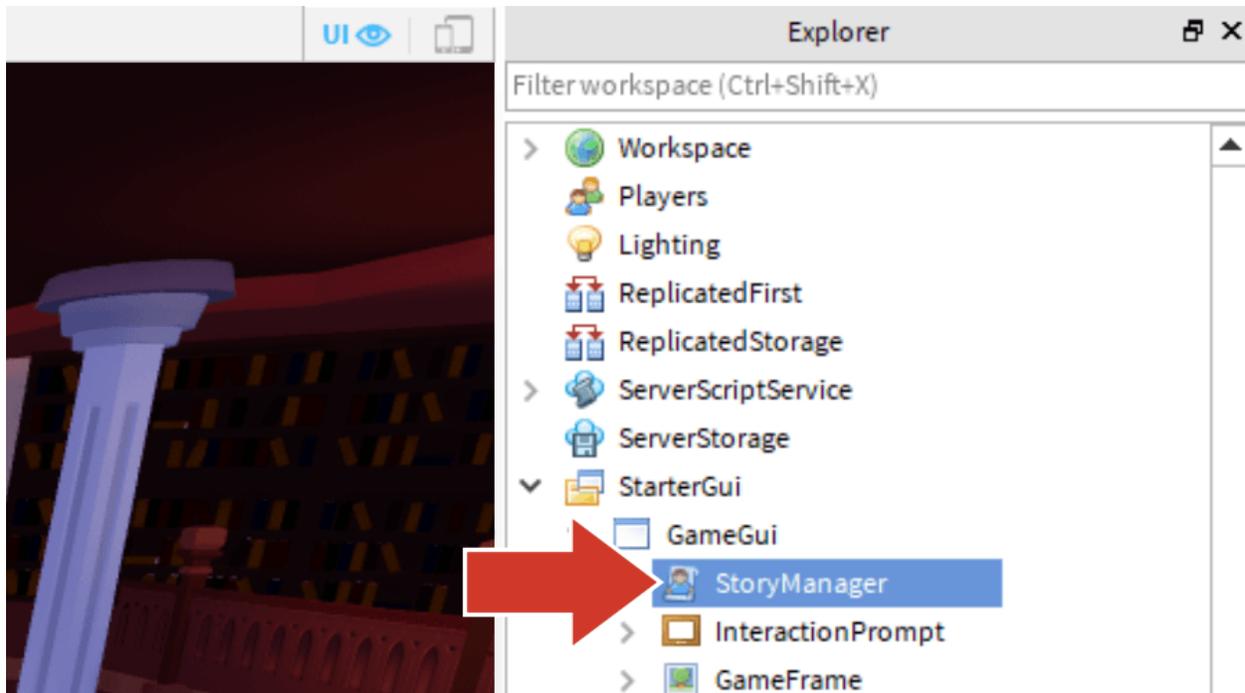
1. Klicke im Explorer-Fenster auf den Pfeil neben **StarterGUI**, um alle Elemente darunter anzuzeigen.



2. Klicke auf den Pfeil neben **GameGUI**, um diese Auswahl zu erweitern.



3. Doppelklicke auf das **StoryManager**-Script, um es zu öffnen.



## Das StoryManager-Script

Das Script enthält bereits Teile des Codes, der benötigt wird, um dem Spieler die fertige Story anzuzeigen. Sämtlicher Code, den du erstellst, wird unter der gestrichelten Doppellinie eingetippt.

```
--Spezieller Dank an @polarpanda16.  
  
-- GLOBALE VARIABLEN  
local storyMaker = require(script:WaitForChild("StoryMaker"))  
  
-- Code, der das Spiel steuert  
local playing = true  
  
while playing do  
    storyMaker:Reset()  
  
    -- Programmiere die Story zwischen den gestrichelten  
    Doppellinien
```

```
-- =====  
  
-- =====  
  
-- Füge die Story-Variable in die Klammer unten ein  
storyMaker:Write()  
  
-- Nochmal spielen?  
playing = storyMaker:PlayAgain()  
end
```

### Kommentare

Codezeilen, die mit -- beginnen, sind **Kommentare**. Damit kannst du Hinweise für dich selbst oder für andere Programmierer hinterlassen. Sie haben keinen Einfluss auf die Ausführung des Programms.

## Variablen programmieren

Zuerst musst du die Platzhalter für die Wörter programmieren, die vom Spieler eingefüllt werden sollen. In einem Script werden Platzhalter für Informationen **Variablen** genannt. Die Informationen, die sie enthalten, können nach Belieben geändert oder ersetzt werden.

## Erstelle die Variable

In der Programmiersprache Lua erstellst du Variablen, indem du zuerst `local` und dann den `namenDerVariablen` eingibst. Alle Variablennamen müssen aus einem oder mehreren aneinandergereihten Wörtern **ohne** Leerzeichen bestehen, genau wie ein Hashtag, und sie müssen mit einem Kleinbuchstaben beginnen. Mit einem aussagekräftigen Variablennamen weißt du immer, wofür die Variable steht.

Verwende den Platzhalternamen, den du vorhin aufgeschrieben hast, als den Namen deiner Variablen.

- Gib unter der ersten gestrichelten Doppellinie `local` und dann einen Namen für die Variable ein. Der Platzhalter in der Beispiel-Story war `name1`, die erste Variable ist also `local name1`.

Ursprünglicher Platzhalter
Vergiss nicht, <code>Workspace</code> großzuschreiben.
Platzhalter im Code
<pre>while playing do     storyMaker:Reset()      -- Programmiere die Story zwischen den gestrichelten     Doppellinien     -- =====      local name1</pre>



## Stell dich der Herausforderung!

Prüfe dein Wissen und verdiene den ersten von drei Preisen, indem du auf die folgende Schaltfläche klickst, um ein Roblox-Quizspiel zu öffnen. Wenn du deinen Preis verdient hast, komm hierher zurück und klicke auf die Schaltfläche **Weiter**, um zu erfahren, wie du Spielern Fragen stellen kannst.

[STELL DICH DER HERAUSFORDERUNG](#)

# Fragen stellen

Die Variable, die du gerade erstellt hast, wird dazu verwendet, Spielern Fragen zu stellen und ihre Antworten zu speichern. Die Fragen werden für den Spieler in der Bildschirmmitte angezeigt, nachdem er auf das große Buch im Eingangsbereich der Bibliothek geklickt hat.



## Füge eine Frage hinzu

### Informationen in der Variablen speichern

Was in Variablen gespeichert wird, kann durch Verwendung von = geändert werden.

Eine Variable kann sich in einem Script oder im Programmablauf oft ändern.

1. Gib nach dem Variablennamen wie z. B. local name1 das Symbol = ein.

```

while playing do
  storyMaker:Reset()

  -- Programmiere die Story zwischen den gestrichelten
Doppellinien
  -- =====

  local name1 =

```

Im nächsten Schritt musst du den Spielern eine Frage stellen und ihre Antwort in der Variablen speichern.

2. Gib zuerst = und dann storyMaker:GetInput("Frage") ein. Der Code muss genauso eingegeben werden, einschließlich der Groß-/Kleinschreibung.

```

while playing do
  storyMaker:Reset()

  -- Programmiere die Story zwischen den gestrichelten
Doppellinien
  -- =====

  local name1 = storyMaker:GetInput("Frage")

```

### storyMaker:GetInput()

Scripts können miteinander kommunizieren. Dieses Spiel verfügt über ein weiteres Script namens storyMaker, und darin befindet sich Code, um die Eingabe der Spieler zu erfassen. Die gerade erstellte Variable lässt den Code des storyMaker-Scripts ablaufen und speichert dann die Antwort des Spielers.

## String-Typen

Variablen können unterschiedliche Informationsarten speichern, z. B. niedrige Zahlen, Werte für wahr oder falsch sowie Strings (Zeichenketten). **String**-Variablen sind etwas Besonderes, weil sie ganze Sätze speichern können.

Sie sind leicht zu erkennen, weil sie immer in geraden Anführungszeichen stehen, "nämlich so".

- Ersetze "Frage" mit der tatsächlichen Frage, die du den Spielern stellen möchtest, um die nächste Zeile der Story abzuschließen. Vergiss die geraden Anführungszeichen ("" ) nicht.

```
while playing do
  storyMaker:Reset()

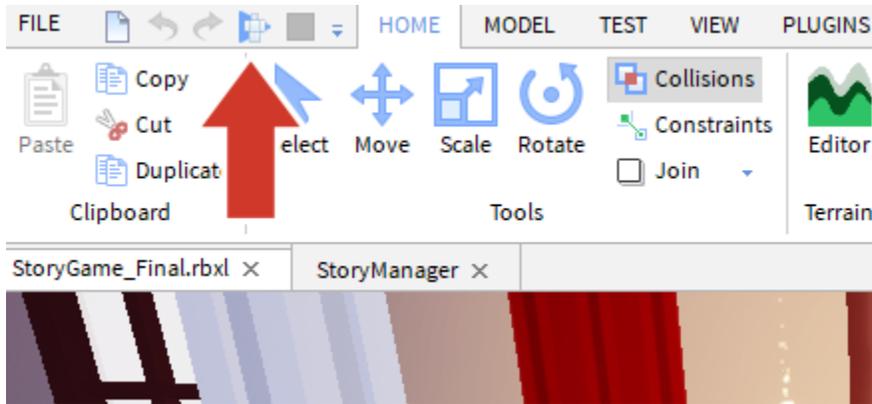
  -- Programmiere die Story zwischen den gestrichelten
  Doppellinien
  -- =====

  local name1 = storyMaker:GetInput("Was ist dein
  Lieblingsname?")
```

## Deinen Code testen

Jetzt solltest du deinen Code testen, damit du siehst, ob er auch funktioniert. Das solltest du häufig tun, damit dir schnell auffällt, ob du einen Fehler gemacht hast.

1. Teste deinen Code, indem du auf die Schaltfläche **Play** klickst.



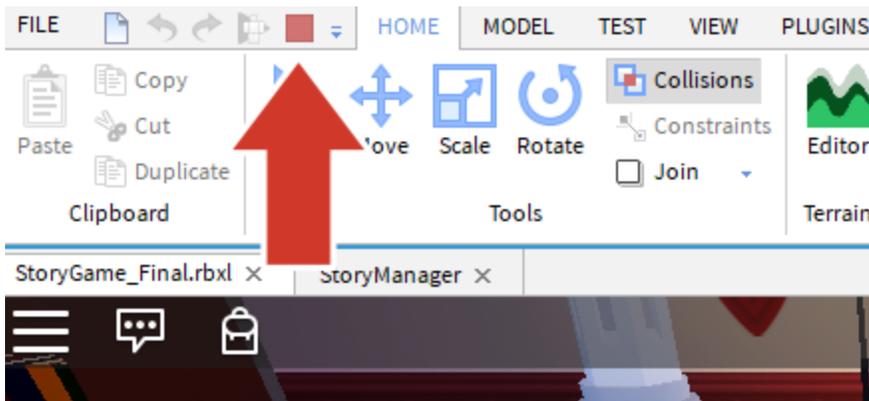
2. Bewege dich mit den Tasten W A S D auf deiner Tastatur zum Podest und drücke dann E.

Steuerung	Aktion
W A S D oder <b>Pfeiltasten</b>	Charakter bewegen
Leertaste	Springen
Rechte Maustaste (Taste halten und Maus bewegen)	Umsehen

3. Sieh nach, ob deine Frage angezeigt wird. Versuche nicht, die Frage zu beantworten oder auf die Schaltfläche „Submit“ (Senden) zu klicken. Diesen Teil der Story musst du erst noch programmieren.



#### 4. Beende den Spieltest.



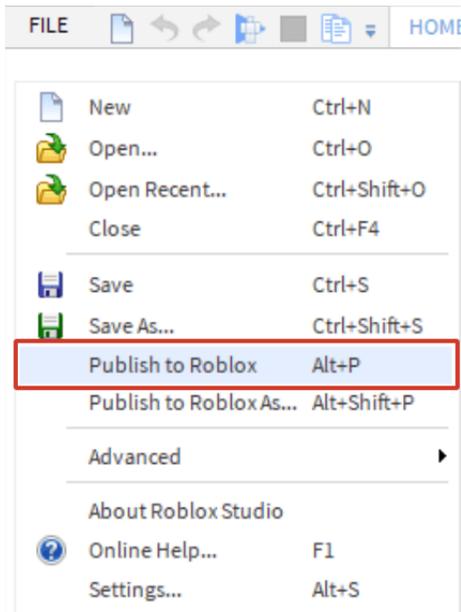
## Deinen Code speichern

Speichere deine Arbeit, indem du dein Spiel veröffentlichst, sodass es von Roblox online gesichert wird. Wir empfehlen, dein Spiel alle 10 bis 15 Minuten zu speichern.

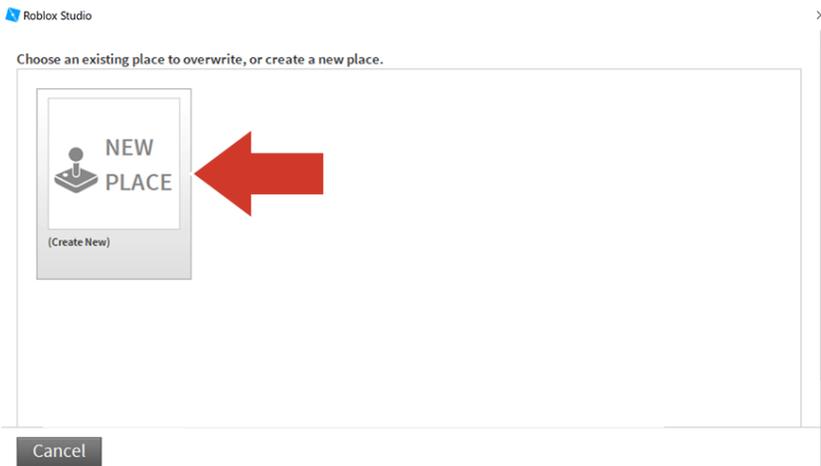
## Das Projekt veröffentlichen

1. Stelle sicher, dass der Spieltest beendet wurde.

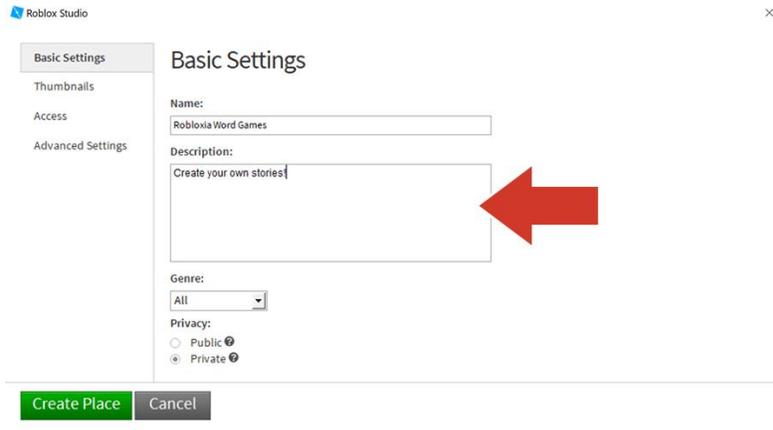
2. Wähle **File** → **Publish** to Roblox.



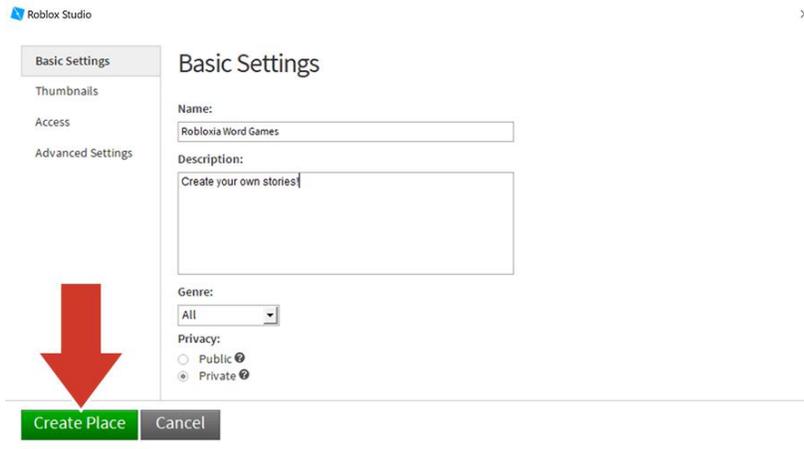
3. Klicke auf **New Place**, um einen neuen Ort zu erstellen.



4. Gib einen Namen und eine Beschreibung ein.



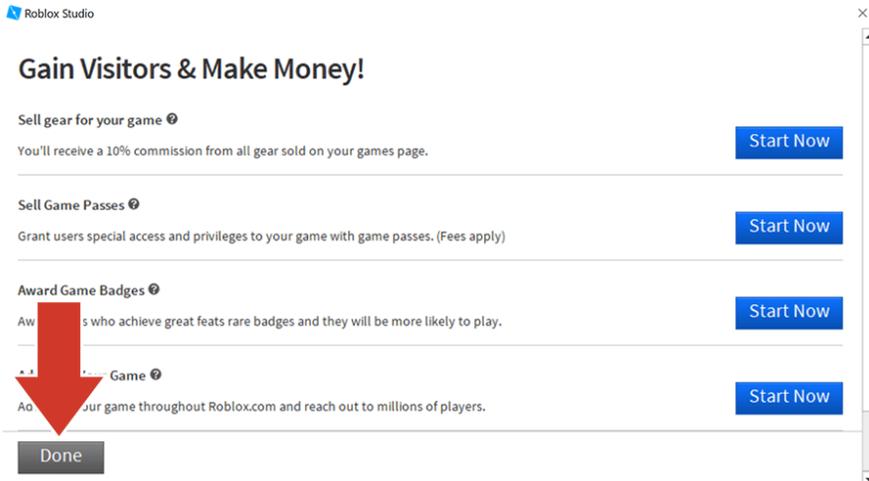
5. Klicke auf **Create Place**, um den Ort zu erstellen.



6. Klicke auf die blaue **Next**-Schaltfläche, wenn die Fortschrittsleiste voll ist und darunter „100% Completed“ steht.



7. Klicke auf **Done**, wenn du fertig bist.



## Stell dich der Herausforderung!

Prüfe dein Wissen und verdiene den zweiten von drei Preisen, indem du auf die folgende Schaltfläche klickst, um ein Roblox-Quizspiel zu öffnen. Wenn du deinen Preis verdient hast, komm hierher zurück und klicke auf die Schaltfläche **Weiter**, um zu erfahren, wie du deine Story schreiben kannst.

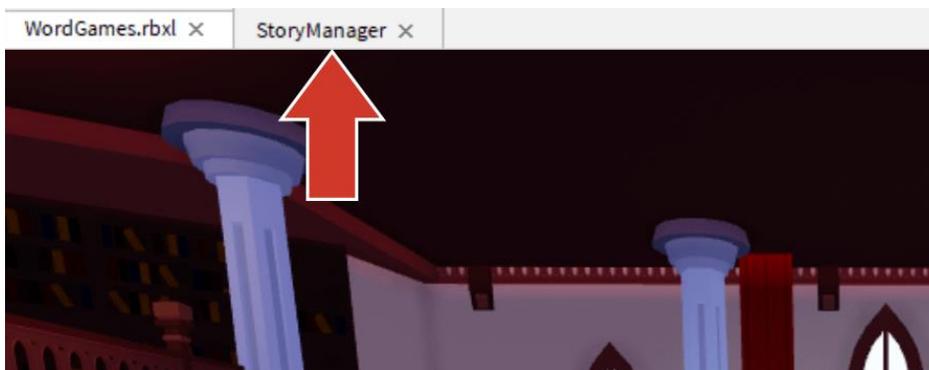
[STELL DICH DER HERAUSFORDERUNG](#)

# Verfasse die Story

Nachdem der Spieler alle Fragen beantwortet hat, sieht er seine Antworten in deiner Story. Die Story wird auch mithilfe von Strings in einer Variablen gespeichert und dann mit den Strings kombiniert, welche die Antworten des Spielers enthalten.

## Programmiere den ersten String

1. Stelle sicher, dass der Spieltest beendet wurde.
2. Gehe zurück zum Script, indem du auf den Reiter für das **StoryManager**-Script über dem Spieleditor klickst.



### Script wieder öffnen

Solltest du den Script-Editor-Reiter nicht sehen, kannst du ihn wieder öffnen, indem du im Explorer-Fenster auf **StoryManager** doppelklickst.

3. Erstelle unter deiner Frage eine neue Variable namens `story`. Stelle sicher, dass die Variable **kleingeschrieben** wird.

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien
-- =====

local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")

local story
```

4. Gehe zurück zur Originalstory, um den ersten String zu finden. Umrande oder markiere alles vor dem ersten Platzhalter. Es ist kein Problem, wenn deine Variable mitten in einem Satz steht – den Rest kannst du einfach später hinzufügen.

#### Ursprünglicher Platzhalter

In einem Baum auf einem Hügel lebt der mächtige Zauberer name1.

5. Bringe den String wie folgt in der Variablen story unter:

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien
-- =====

local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")

local story = "In einem Baum auf einem Hügel lebt der mächtige
Zauberer "
```

**Tipps zur Fehlerbehebung**  
**Fehlerbehebung bei Schleifen**

- Vergiss nicht, dass local story kleingeschrieben werden muss, da andere Teile des Spiels nach einer Variablen mit genau diesem Namen suchen.
- Der Code zeigt nur das an, was du programmiert hast. Füge also am Ende ein Leerzeichen wie im folgenden Beispiel ein, damit die Strings nicht aneinanderkleben.

"In einem Baum auf einem Hügel lebt der mächtige Zauberer "

## Die Story ausbauen

### Kombiniere die Story und die Antwort

Als Nächstes muss der erste String der Story mit der Antwort des Spielers kombiniert werden. Das Verbinden von Elementen nennt sich **Verkettung**. Zum Verketteten von zwei Strings verwendest du ..

1. Tippe in derselben Zeile wie die story-Variable ..

```
while playing do
  storyMaker:Reset()

  -- Programme die Story zwischen den gestrichelten
  Doppellinien
  -- =====

  local name1 = storyMaker:GetInput("Was ist dein
  Lieblingsname?")

  local story = "In einem Baum auf einem Hügel lebt der
  mächtige Zauberer " ..
```

2. Und in dieselbe Zeile schreibst du auch den Namen der Variablen, welche die Antwort des Spielers enthält.

```
while playing do
  storyMaker:Reset()

  -- Programmiere die Story zwischen den gestrichelten
Doppellinien
  -- =====

  local name1 = storyMaker:GetInput("Was ist dein
Lieblingsname?")

  local story = "In einem Baum auf einem Hügel lebt der
mächtige Zauberer " .. name1
```

## Story schreiben und Spiel testen

Nachdem du die Story nun eingegeben hast, muss sie dem Spieler angezeigt werden.

1. Suche unter der zweiten gestrichelten Doppellinie nach `storyMaker:Write()`. Gib zwischen `()` die Variable `story` ein. Dies teilt dem Programm mit, die Story im Spiel zu zeigen.

```
while playing do
  storyMaker:Reset()

  -- Programmiere die Story zwischen den gestrichelten
Doppellinien
  -- =====

  local name1 = storyMaker:GetInput("Was ist dein
Lieblingsname?")

  local story = "In einem Baum auf einem Hügel lebt der
mächtige Zauberer " .. name1

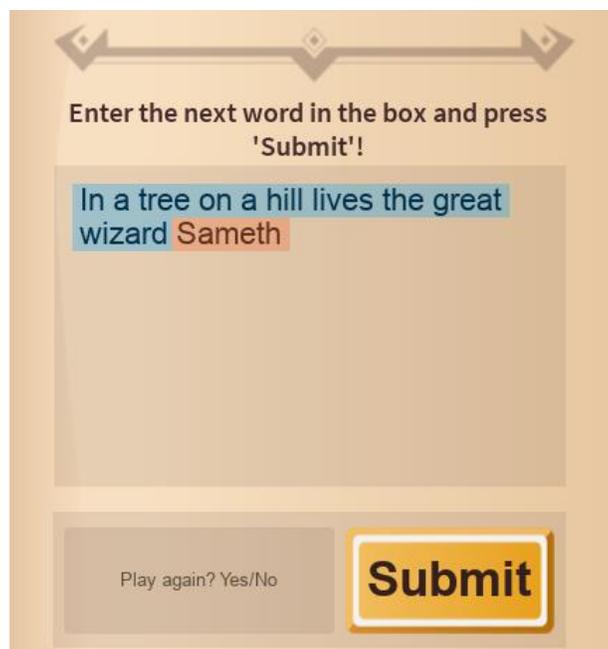
  -- =====
```

```
-- Füge die Story-Variable in die Klammer unten ein  
storyMaker:Write(story)
```

### Überprüfe deine Arbeit

Stelle sicher, dass du story in die Zeile storyMaker:Write(story) eingefügt hast und dass die Groß-/Kleinschreibung genau der Codebox entspricht. Ohne diesen Schritt wird die Story im Spieltest nicht erscheinen.

2. Führe einen **Spieltest** durch. Die beiden Strings sollten jetzt zusammen angezeigt werden.



### Tipps zur Fehlerbehebung Fehlerbehebung bei Schleifen

Sollte die Frage nicht gestellt werden, überprüfe Folgendes:

- Befindet sich die Frage innerhalb von geraden Anführungszeichen?

**Sollte die Story nicht zusammengefügt werden, überprüfe**

**Folgendes:**

- Befindet sich der erste Teil der Story innerhalb von geraden Anführungszeichen?
- Ist die Schreibung der Variablen für die Antwort des Spielers immer identisch? Dieselbe Groß-/Kleinschreibung ist ausschlaggebend!
- Befindet sich der Variablenname für die Antwort des Spielers **nicht** innerhalb von geraden Anführungszeichen?
- Werden die beiden Strings durch .. getrennt?

**Sollte deine Story nicht angezeigt werden:**

- Stelle sicher, dass du die story-Variable in die () in storyMaker:Write() eingefügt hast.

## Schließe deinen Satz ab

Füge einen weiteren String hinzu, um mehr Wörter oder ein Satzzeichen zu deinem Satz hinzuzufügen.

1. Füge in derselben Zeile wie die Story-Variable ein weiteres .. hinzu.
2. Füge einen weiteren String mit dem Rest deines Satzes oder einfach nur einen Punkt hinzu. Vergiss nicht, am Ende des Satzes ein zusätzliches Leerzeichen einzufügen.

```
-- =====  
  
local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")  
  
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". "
```

## Füge eine zweite Frage hinzu

Um eine zweite Frage zu stellen, verfasse eine neue Frage und erweitere damit die Variable, welche die Story enthält.

1. Entscheide dich für ein Wort, das du aus dem zweiten Satz deiner Story entfernen möchtest.

### Ursprünglicher Platzhalter

In einem Baum auf einem Hügel lebt der mächtige Zauberer name1.  
Zum Frühstück verspeist der Zauberer jeden Morgen eine riesige  
Schüssel voll honiggerösteter food1.

2. Erstelle unter der ersten Variablen eine neue Variable, die als Platzhalter dient.

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien  
  
-- =====  
  
local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")  
local food1  
  
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". "
```

### Zur Erinnerung: Variablennamen

Vergiss nicht, dass Namen von Variablen kleingeschrieben werden müssen und keine Leerzeichen enthalten dürfen.

3. Verwende `storyMaker:GetInput("")`, um dem Spieler eine Frage zu stellen und seine Antwort zu speichern.

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien
-- =====

local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")
local food1 = storyMaker:GetInput("Was ist dein Lieblingsessen?")
```

4. Verkette den nächsten Story-String in der Story-Variablen mit `..`. Vergiss nicht, nach dem Satzende ein Leerzeichen einzufügen.

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien
-- =====

local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")
local food1 = storyMaker:GetInput("Was ist dein Lieblingsessen?")

local story = "In einem Baum auf einem Hügel lebt der mächtige
Zauberer " .. name1 .. ". " .. "Zum Frühstück verspeist der
Zauberer jeden Morgen eine riesige Schüssel voll honiggerösteter
"
```

5. Nach dem neuen Story-String verkettest du die Antwort auf die **zweite** Frage und schließt dann mit einem Satzzeichen ab.

```
-- Programmiere die Story zwischen den gestrichelten Doppellinien
```

```
-- =====  
  
local name1 = storyMaker:GetInput("Was ist dein Lieblingsname?")  
local food1 = storyMaker:GetInput("Was ist dein Lieblingsessen?")  
  
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". " .. "Zum Frühstück verspeist der  
Zauberer jeden Morgen eine riesige Schüssel voll honiggerösteter  
" .. food1 .. ". "
```

## Die Story abschließen

Erweitere dein Spiel, indem du mehr Variablen und Fragen hinzufügst und Strings verkettest.

1. Füge deiner Story weitere Zeilen hinzu.
2. Führe regelmäßig Spieltests durch, nachdem du neue Variablen und Strings hinzugefügt hast.

## Tipps und Tricks

### Variablen mehr als einmal verwenden

Variablen können mehr als nur einmal verwendet werden – verkette dafür einfach Strings, zwischen denen du das Wort oder die Wörter einsetzen möchtest.

#### Beispiel-Code

```
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". " .. "Zum Frühstück verspeist " .. name1 .. "  
jeden Morgen eine riesige Schüssel voll honiggerösteter " .. food1 .. ". "
```

#### Ergebnis

In einem Baum auf einem Hügel lebt der mächtige Zauberer Sameth.  
Zum Frühstück verspeist Sameth jeden Morgen eine riesige Schüssel  
voll honiggerösteter Kartoffeln.

## Zeilenumbrüche hinzufügen

Füge Zeilenumbrüche hinzu, indem du `\n` in einen String einfügst.

### Mit einem Zeilenumbruch

#### Beispiel-Code

```
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". " .. "\nZum Frühstück verspeist der Zauberer  
jeden Morgen eine riesige Schüssel voll honiggerösteter " .. food1 .. ". "
```

#### Ergebnis

In einem Baum auf einem Hügel lebt der mächtige Zauberer Sameth.  
Zum Frühstück verspeist der Zauberer jeden Morgen eine riesige  
Schüssel voll honiggerösteter Kartoffeln.

### Mit zwei Zeilenumbrüchen in Folge

#### Beispiel-Code

```
local story = "In einem Baum auf einem Hügel lebt der mächtige  
Zauberer " .. name1 .. ". " .. "\n\nZum Frühstück verspeist der Zauberer  
jeden Morgen eine riesige Schüssel voll honiggerösteter " .. food1 .. ". "
```

#### Ergebnis

In einem Baum auf einem Hügel lebt der mächtige Zauberer Sameth.  
Zum Frühstück verspeist der Zauberer jeden Morgen eine riesige  
Schüssel voll honiggerösteter Kartoffeln.



## Stell dich der Herausforderung!

Prüfe dein Wissen und verdiene den letzten Preis, indem du auf die folgende Schaltfläche klickst, um ein Roblox-Quizspiel zu öffnen. Wenn du deinen Preis verdient hast, komm hierher zurück und klicke auf die Schaltfläche **Weiter**, um zu erfahren, wie du die Entwickler-Herausforderung abschließen kannst.

[STELL DICH DER HERAUSFORDERUNG](#)

# Die Herausforderung abschließen



Herzlichen Glückwunsch, du hast soeben die Entwickler-Herausforderung von Roblox abgeschlossen! Nachdem du nun dein Spiel erstellt hast, findest du hier einige Tipps, um dein Spiel weiterzuentwickeln und neue Dinge zu lernen.

## Teile dein Spiel

Bisher wurde dein Spiel in der Cloud gespeichert, und nur du allein kannst es sehen.

Bei Roblox kann jeder seine Projekte mit Freunden und Spielern aus aller Welt teilen!

[SO FUNKTIONIERT DAS TEILEN](#)

## Erweitere deine Story

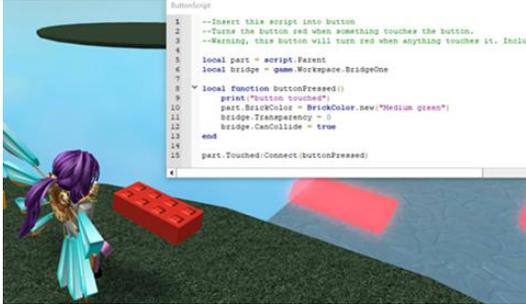
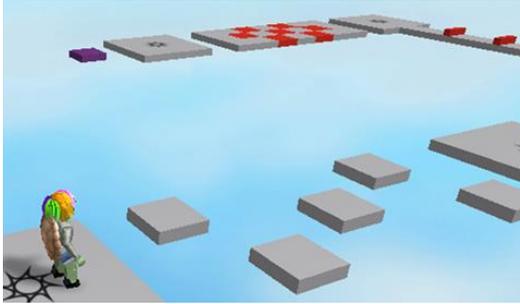
Du hast die Herausforderung zwar abgeschlossen, aber du kannst deine Story jederzeit erweitern und verbessern. Hier sind ein paar Vorschläge:

- **Füge mehr Charaktere hinzu** – Denke dir einen oder sogar zwei neue Charaktere für deine Story aus. Vergiss nicht, für jeden Charakter eine neue Variable zu erstellen, z. B. name2 oder name3.

- **Füge mehr Zeilen hinzu** – Schreibe einen neuen Satz für deine Story und wähle ein Objekt, das ersetzt werden kann.

## Lerne weiter

Möchtest du mehr darüber lernen, wie du deine eigenen Spiele erstellen kannst? Du könntest zum Beispiel einen Hindernisparcours entwerfen oder ein Power-up programmieren, das Charakteren einen Temposchub verleiht! Sieh dir diese Lektionen im [Entwicklerzentrum](#) an. Auf dieser Seite gibt es jede Menge Tutorials zum Erstellen von Roblox-Spielen.

 <p><b>Programmierlektionen</b> Lerne in diesen Schritt-für-Schritt-Lektionen mehr über das Programmieren. <a href="#">Ansehen</a></p>	 <p><b>Baulektionen</b> Baue einen Hindernisparcours und passe ihn individuell an. <a href="#">Ansehen</a></p>
--	---

These documents are licensed by Roblox Corporation under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). Roblox, Powering Imagination, and Robux are trademarks of Roblox Corporation, registered in the United States and other countries. The Hour of Code™ is a nationwide initiative by [Computer Science Education Week](#) and [Code.org](#) to introduce millions of students to one hour of computer science and computer programming.