Arbeiten mit TWINE

Die Grundlagen

Start

www.twinery.org -> Use it online -> oben in der Leiste auf +Neu für neues Dokument oder auf Aus Datei importieren klicken, um an einer gespeicherten Geschichte weiterzuarbeiten

Einen neuen Abschnitt hinzufügen

Oben links im Menü auf +Neu klicken.

Verlinkung zu einem anderen Abschnitt

Der Text, der später anklickbar sein soll, wird in jeweils zwei eckige Klammern gepackt.

[[*Dieser Text wird ein Link!*]]

In diesem Satz ist nur [[*dieses*]] *Wort ein Link.*

Twine erstellt automatisch einen neuen Abschnitt samt Verlinkung, wenn Text in eckige Klammern geschrieben wird.

Um den Überblick zu behalten, empfiehlt es sich, den Abschnitten prägnante Namen zu geben. Dafür schreibt man hinter den Text, aber in die eckigen Klammern, einen Pfeil und den gewünschten Abschnittsnamen.

[[*Das hier ist der sichtbare Text.*->*Ziel-Abschnittsname*]]

Bilder einbetten

<img src="*http://www.direkter-link-auf-das-bild.de/foto.jpg*" width="*500*" height="*300*">

Eingefügt werden muss der direkte Bildlink – erkennbar daran, dass er z. B. auf .jpg oder .png endet. Auf vielen Websites lässt sich dieser Link mit einem Rechtsklick auf das Bild > *Grafikadresse kopieren* finden.

Die Zahlenwerte hinter *width* und *height* geben an, wie groß das Bild angezeigt werden soll. Kann auch weggelassen werden.

Insbesondere bei geplanter Veröffentlichung auf Copyright-Bestimmungen achten!

Videos, GIFs etc. einbetten

Das Einfügen von Videos ist in den letzten Jahren komplizierter geworden, weil viele Dienste (z. B. YouTube) das Einbetten auf anderen Seiten teilweise unterdrücken. Eine gute Alternative ist das Einbetten von GIFs – kurze Animationen, die einfach wie ein Bild mit <img> eingebunden werden können.

Sounds einbetten

<audio src="*http://www.direkter-link-auf-die-sounddatei.de/sound.wav*" autoplay>

MP3-Dateien werden beim Testen nicht abgespielt, sondern erst, wenn die Datei gespeichert (*Als Datei veröffentlichen)* und anschließend im Browser geöffnet wird.

WAV-Dateien sind auch schon im Test-Modus hörbar. Größere WAV-Dateien benötigen manchmal etwas Zeit zum Laden, daher kann es ggf. zu Verzögerungen beim Abspielen kommen.

Den Text stylen

Fett: ***fetter Text***

Kursiv*: *kursiver Text**

Unterstrichen*: <u>unterstrichener Text</u>*

Text verkleinern*: ^^verkleinerter Text^^*

Über das Stylesheet lassen sich grundlegende Einstellungen wie Schriftfarbe oder Schriftgröße verändern. Hier einige Beispiele:

Hintergrund*: Body {background-color: white;}*

Links*: tw-link {color: red;}*

Text*: tw-passage {color: black;}*

Links, die bereits angeklickt wurden*: tw-link.visited {color: green;}*

Schriftgröße*: Body, tw-story {font-size: 40px}*

Statt der Farbbezeichnungen können auch sechsstellige RGB-Hex-Werte verwendet werden. Zum Beispiel Kornblumenblau:

color: #6495ED;

Informationen von den Spieler:innen abfragen mit der prompt-Funktion

Am Anfang des Spiels (aber auch später noch) können Informationen von den Spieler:innen eingeholt werden, z. B. der Name oder das Alter.

Dazu fügt man in den Abschnitt, wo die Abfrage erscheinen soll, folgende Code-Zeile ein:

(set: $name to (prompt: "Wie heißt du?", ""))

In dem ersten ""-Paar steht der Text, der dann über dem Eingabefeld steht. In das zweite ""-Paar kommt der Text, der später im Eingabefeld stehen soll – in diesem Beispiel bleibt das Feld leer.

Die Variable $name (oder wie auch immer man sie benennt) kann dann jederzeit im Text eingebaut werden, z. B. so:

Plötzlich triffst du $name auf der Straße.

Boolesche Variablen

Boolesche Variablen können nur zwei unterschiedliche Werte speichern (z.B. true oder false). Ähnlich wie z. B. ein Lichtschalter können sie von einem Zustand (Licht aus) auf einen anderen (Licht an) umschalten.

Wenn der Zustand verändert werden soll, schreibt man z. B. folgendes in einen Abschnitt:

Du schaltest das Licht ein.

(set: $licht to true)

Dabei auf Groß- und Kleinschreibung achten! Variablennamen (hier also $licht) müssen an allen Stellen der Story gleich geschrieben werden. Die Begriffe true und false sind feststehende Befehle und müssen immer kleingeschrieben werden.

Ist die Variable einmal definiert (mit set:), kann ihr aktueller Zustand in allen darauffolgenden Abschnitten abgefragt werden.

(if: $licht is false)[Das Licht ist aus.]

(else:)[Das Licht ist an.]

Hier prüft der Code in der ersten Zeile, welchen Zustand die Variable $licht gerade hat. WENN (if) sie false ist, wird der Text „Das Licht ist aus“ ausgegeben. ANDERNFALLS (else) wird der Text „Das Licht ist an“ ausgegeben. Was in diesem Abschnitt geschrieben wird, hängt also davon ab, ob die Spielerin vorher schon am Lichtschalter vorbeigekommen ist (in diesem Fall nicht).

Wenn der Text noch zusätzlich auf den nächsten Abschnitt verlinken soll, müssen noch die zusätzlichen eckigen Klammern dazu, also z. B. so:

(if: $licht is false)[[[Das Licht ist aus.->Dunkelheit]]]

Integer-Variablen – Spaß mit Zahlen

Integer-Variablen funktionieren ganz ähnlich wie Boolesche, aber sie können deutlich mehr als zwei Zustände speichern, nämlich alle ganzen Zahlen.

So lassen sich zum Beispiel Minuten, Versuche oder Gegenstände zählen – der Fantasie sind keine Grenzen gesetzt. In diesem Beispiel hat eine Spielerin drei Versuche, um eine Frage richtig zu beantworten.

Im Start-Abschnitt wird die Variable definiert:

(set: $versuche to 3)

In dem Abschnitt, in dem die falsche Antwort ausgewertet wird, wird dann jeweils ein Versuch abgezogen, indem dieser Code hinzugefügt wird:

(set: $versuche to $versuche-1)

Wenn das Programm diese Code-Zeile gelesen hat, wird ein neuer Wert der Variablen $versuche gespeichert – nämlich 2 statt vorher 3. Das sollten wir auch der Spielerin mitteilen, indem wir schreiben:

Du hast noch $versuche Versuche!

Wenn die Versuche aufgebraucht sind, sollte das Spiel natürlich darauf reagieren. Mit if und else werden je nachdem, ob es noch Versuche gibt oder nicht, unterschiedliche Sätze ausgegeben:

(if: $versuche is 0)[Deine Versuche sind aufgebraucht. Du hast verloren!]

(else:)[Du hast noch $versuche Versuche!]

Das bedeutet: WENN (if) die Anzahl der Versuche (=die in der Variablen $versuche gespeicherte Zahl) gleich null ist, wird der erste Satz ausgegeben. ANDERNFALLS (else) kriegt die Spielerin den unteren Satz zu lesen.