

Comic erstellen mit Snap

 Vorbereitung	Besorge dir die nötigen Materialien <i>Wir werden in dieser Stunde einen Comic programmieren. Dafür benötigst du einiges an Materialien (Bilder und Sounds). Diese findest du im Ordner Comic in den Vorlagen. Speichere diesen Ordner auf deinem USB-Stick ab.</i>
--	--

1. Hintergrund der Zeichenfläche ändern

Du hast in Snap! die Möglichkeit, für deine Zeichenfläche einen **Hintergrund** zu wählen.

Dazu wählt man die Bühne durch Klicken auf **Stage** aus.

Wähle anschließend im Hauptmenü **Import**. Es erscheint der Dateibrowser, in dem du das Bild, das du verwenden möchtest, auswählen kannst.



 Aufgabe 1	Hintergrund <i>Im Ordner Comic findest du das Bild Hintergrund.png. Weiße dieses wie oben beschrieben deiner Zeichenfläche als Hintergrund zu.</i>
---	---

2. Ein eigenes Bild als Aussehen benutzen

Du weißt bereits, wie man in Snap! das Aussehen einer Figur ändert. Bislang stehen dir dazu jedoch nur vom Programm zur Verfügung gestellte Bilder zur Verfügung. Du hast jedoch auch die Möglichkeit, **eigene Bilder** zu verwenden.

Dies funktioniert genauso wie das Ändern des Hintergrunds der Zeichenfläche. Du musst lediglich anstatt der Zeichenfläche die entsprechende Figur auswählen.

 Aufgabe 2	Figuren <i>Für den Comic benötigen wir insgesamt drei Figuren: Kevin, Stuart und die Spritze. Erzeuge also insgesamt drei Figuren und benenne sie entsprechend. Weiße ihnen anschließend - wie oben beschrieben - das entsprechende Aussehen zu. Dazu findest du im Ordner Comic die Bilder Kevin-normal.png, Stuart-normal.png und Spritze-Bild.png. Positioniere deine Figuren abschließend richtig auf der Zeichenfläche.</i>
---	---

3. Kommunikation von Figuren miteinander

Wie im Video-Clip sollen unsere Figuren nun kurz miteinander sprechen. Dazu müssen sie miteinander **kommunizieren** – eine Figur muss der anderen sagen, wenn sie an der Reihe ist zu sprechen. Dazu benutzt man den Baustein **broadcast**.



Dieser schickt eine **Meldung mit eindeutigem Name** an alle anderen Figuren. Diese können diese **Meldung empfangen** und mit Hilfe des folgenden Bausteins entsprechend darauf reagieren:



Dabei müssen die Namen der gesendeten Meldung (also der Teil hinter broadcast) und der Meldung die empfangen werden soll (also der Teil hinter receive) übereinstimmen.

 <p>Aufgabe 3</p>	<p>Unterhaltung zwischen Kevin und Stuart</p> <p>Stelle mit Hilfe der oben erklärten Technik folgende Unterhaltung zwischen Kevin und Stuart nach:</p> <p>Stuart: „Kevin?!“</p> <p>Kevin: „Stuart?!“</p> <p>„Hallo!“</p> <p>Stuart: „Hallo!“</p>
---	---

4. Sichtbarkeit von Figuren

Figuren können in Snap sichtbar oder unsichtbar sein. Benutze dazu folgende Bausteine:

-  → macht die Figur **sichtbar**
-  → macht die Figur **unsichtbar**

 <p>Aufgabe 4</p>	<p>Spritze - Sichtbarkeit</p>  <p>Zu Beginn des Comics soll die Spritze unsichtbar sein. Klicke dazu einfach einmalig auf den entsprechenden Baustein.</p> <p>Wenn das Gespräch der beiden Minions aus Aufgabe 3 beendet ist, soll die Spritze kurz sichtbar werden. Kurz darauf soll sie dann wieder verschwinden.</p> <p>Tipp: Die Figur, die als letztes spricht, muss der Spritze eine Meldung schicken, auf die sie reagieren kann. Vergleiche dazu Punkt 3.</p>
---	--

5. Sounds

Natürlich kannst du in Snap! auch eigene **Sounds** verwenden. Wähle dazu die Figur aus, die diesen Sound wiedergeben soll und gehe genauso vor, wie beim Importieren eines eigenen Bildes. Wähle im Dateibrowser keine Bilddatei, sondern eine Sounddatei aus.

Einen importierten Sound kannst du dann mithilfe des folgenden Bausteins abspielen:

play sound until done

 <p>Aufgabe 5</p>	<p>Spritze - Sound</p> <p><i>Das Erscheinen der Spritze soll auch das entsprechende Geräusch verursachen. Dazu findest du im Ordner Comic die Datei Spritze-Sound.mp3.</i></p> <p><i>Importiere sie wie oben beschrieben und spiele diesen Sound gleich nach Erscheinen der Spritze ab.</i></p>
---	--

6. Aussehen einer Figur im Laufe des Programms ändern

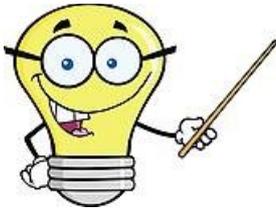
Möchtest du im Laufe eines Programms das **Aussehen deiner Figur ändern**, so muss das gewünschte Aussehen erst wie in Punkt 2 beschrieben **importiert** werden.

Anschließend kann das Aussehen einer Figur mit Hilfe des folgenden Bausteins während des Programmablaufs geändert werden:

switch to costume

 <p>Aufgabe 6</p>	<p>Verwandle Stuart in einen bösen Minion</p> <p><i>Nachdem die Spritze wieder verschwunden ist, soll sich Stuart in einen bösen, lila Minion verwandeln. Dazu findest du im Ordner Comic das Bild Stuart-boese.png.</i></p> <p><i>Ändere das Aussehen von Stuart wie oben beschrieben.</i></p> <p><i>Mache im Programm eine kleine Pause und lass Stuart dann „bäh“ schreien. Dazu findest du im Ordner Comic die Datei Bah.mp3.</i></p>  <p>Tipp: Die Spritze muss nach ihrem Erscheinen Stuart eine Meldung schicken, auf die er reagieren kann. Vergleiche dazu Punkt 3.</p>
---	---

 <p>Hausaufgabe</p>	<p>Kevins Gelächter</p>  <p><i>Nachdem Stuart in einen bösen lila Minion verwandelt wurde, soll Kevin in Gelächter ausbrechen. Dazu findest du die Bilddatei Kevin-lachend.png und die Sound-Datei Lachen.mp3 im Ordner Comic.</i></p> <p><i>Ändere Kevins Aussehen nachdem Stuart verwandelt wurde. Lass anschließend das Lachen ertönen.</i></p>
---	---



Tipps und Tricks

- Bei der Kommunikation von Figuren untereinander ist es nützlich, den einzelnen Meldungen **sprechende Namen** zu geben – d.h. Namen, die bereits beschreiben, was passieren soll. So behältst du einen besseren Überblick über die verschiedenen Meldungen.
- Falls dein Sound nicht abgespielt werden kann, obwohl er richtig importiert worden ist, so liegt es oft am Internetexplorer. Verwende dann zum Programmieren in Snap! GoogleChrome oder Firefox.
- Unter dem Punkt **Costumes** kannst du bei jeder Figur per Mausklick zwischen den einzelnen möglichen Aussehen hin- und herschalten.



**für die
Schnellen...**



Anfangsbedingungen für Video wiederherstellen

Nach Beendigung unseres Videos sehen Stuart und Kevin nun anders aus, als zuvor. Wenn wir das Video nochmal abspielen möchten, müssen wir ihr Aussehen erst per Hand wieder in den ursprünglichen Zustand zurückversetzen.

Im Folgenden soll dies automatisch geschehen – und zwar, wenn die Leertaste gedrückt wird.

Tip: Verwende folgenden Baustein:

