

App Inventor – Quiz Tutorial

In diesem Tutorial wirst du ein kleines Quiz mit dem App Inventor erstellen. Dabei handelt es sich aber erst einmal nur um eine grobe Vorlage, die du im Anschluss noch erweitern kannst.

Unsere Quiz-App wird direkt mit der ersten Frage starten und bei richtiger Antwort mit der nächsten Frage weitermachen. Nach richtiger Beantwortung der zweiten Frage kann dann schon der Punktestand erscheinen.

Hinweis

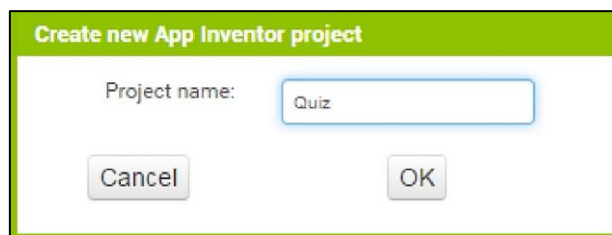
Es wird davon ausgegangen, dass du bereits eine kurze Einführung in den App Inventor hattest und weißt ...

- wo und wie du die Entwicklungsumgebung aufrufst,
- was die Design- und Block-Ansicht ist und wie sie aufgebaut sind.

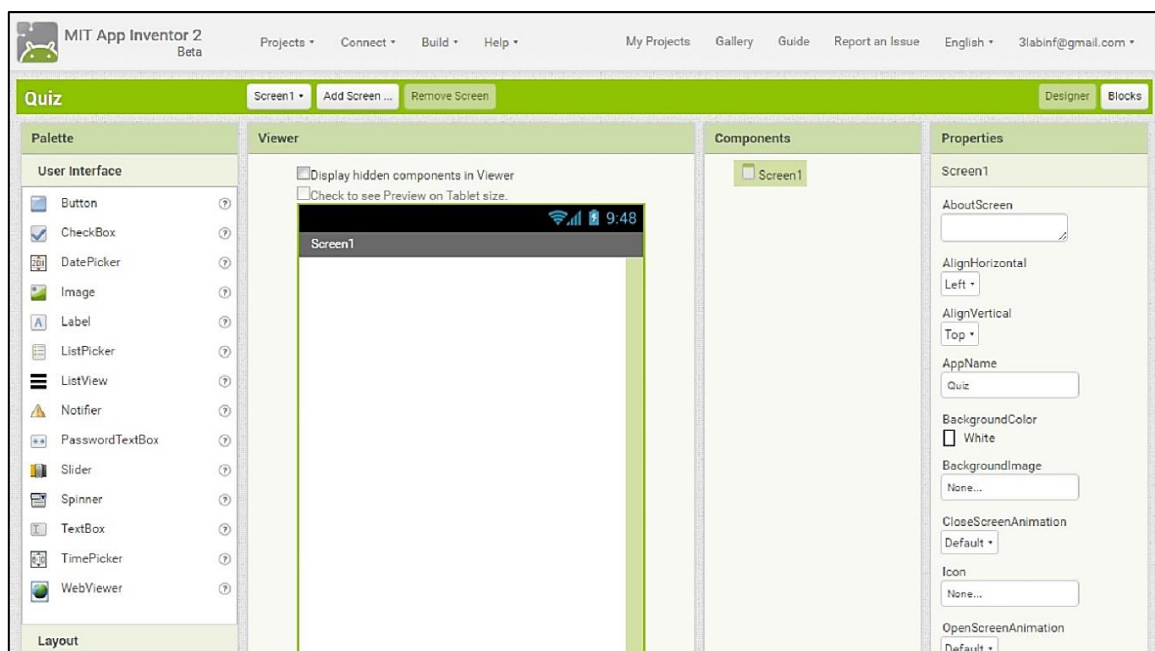
Wenn du bisher keine Einführung hattest, dann lies bitte die Einführung B7.1 genau durch bevor du hier weiter machst.

Schritt 1

Rufe die Webseite von App Inventor auf und starte die Entwicklungsumgebung. Rufe dann die Übersicht deiner Projekte auf und erstelle ein neues Projekt mit dem Namen *Quiz*.

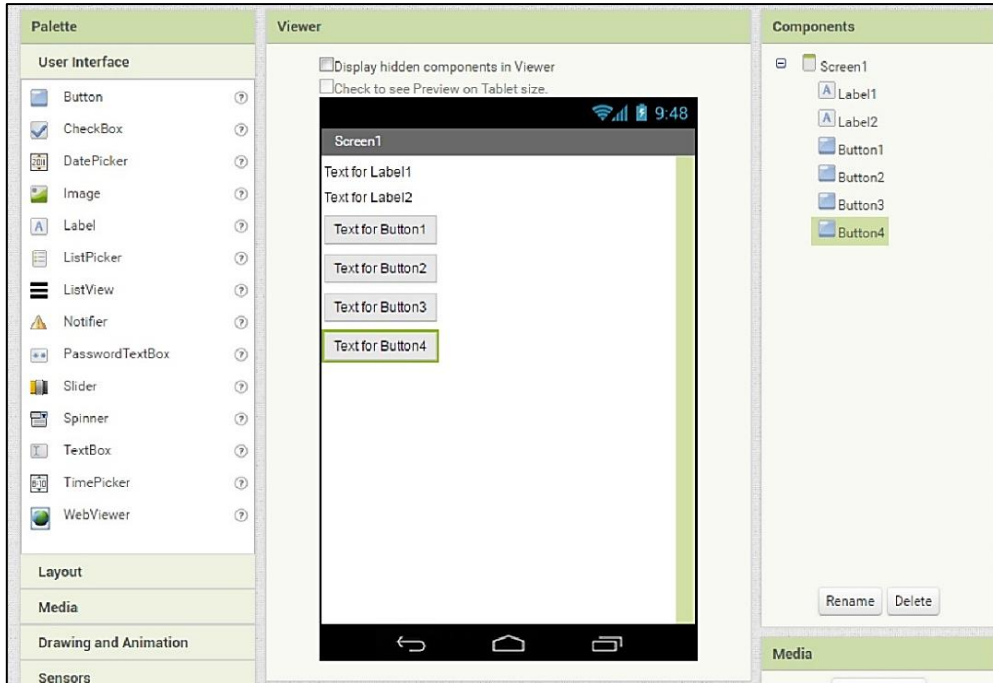


Anschließend erhältst du die gewohnte Ansicht eines neuen Projekts.



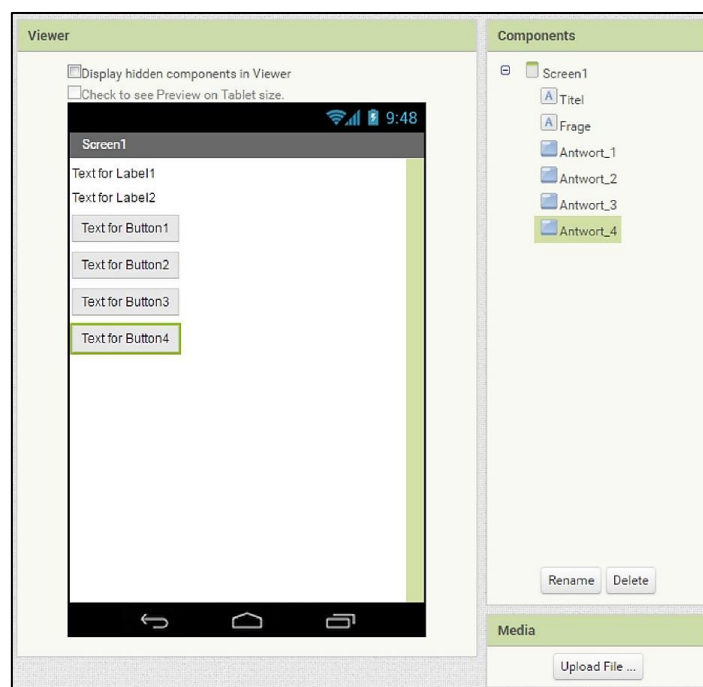
Schritt 2

Für die erste Frage des Quiz benötigen wir zwei *Labels* und vier *Buttons*, die du von der *Palette* in den *Viewer* ziehen musst. Achte darauf, dass die Komponenten wie auf dem folgenden Bild angeordnet sind, d.h. dass die zwei *Labels* ganz oben sind und daran die vier *Buttons*.



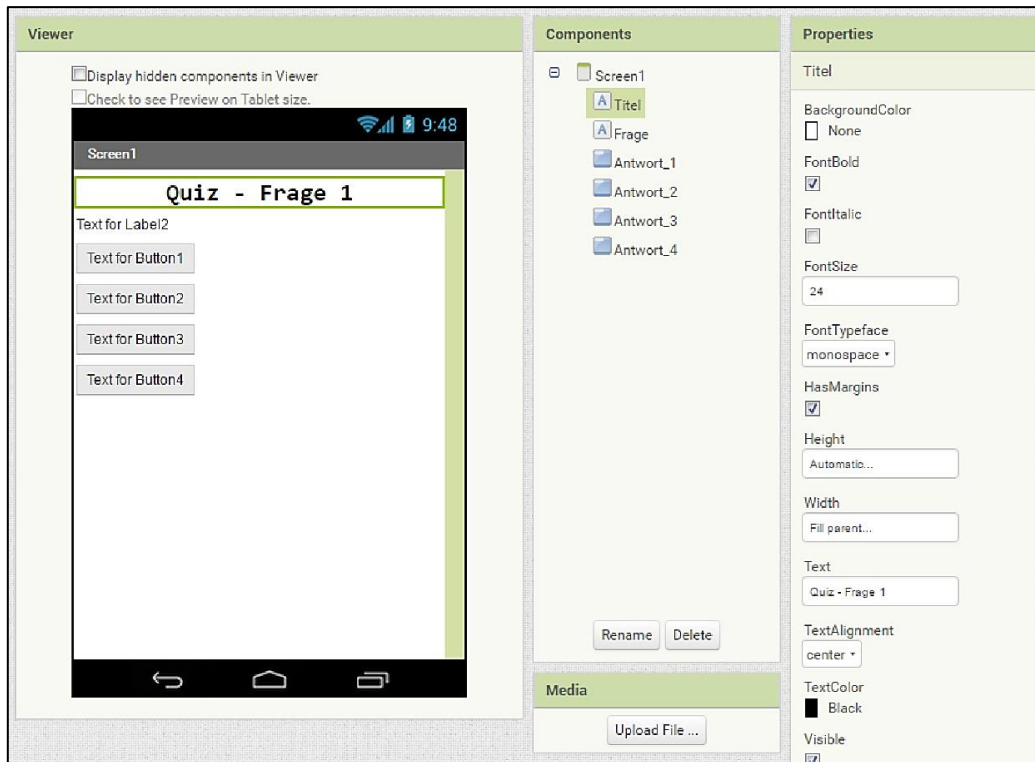
Schritt 3

Damit wir später die *Labels* und *Buttons* besser unterscheiden können, kannst du die Komponenten umbenennen. Das erste *Label* sollte der Titel sein, das zweite *Label* enthält die Frage und die vier *Buttons* werden die Antwortmöglichkeiten. Benenne die Komponenten wie beschrieben, du kannst dich an folgendem Bild orientieren.



Schritt 4

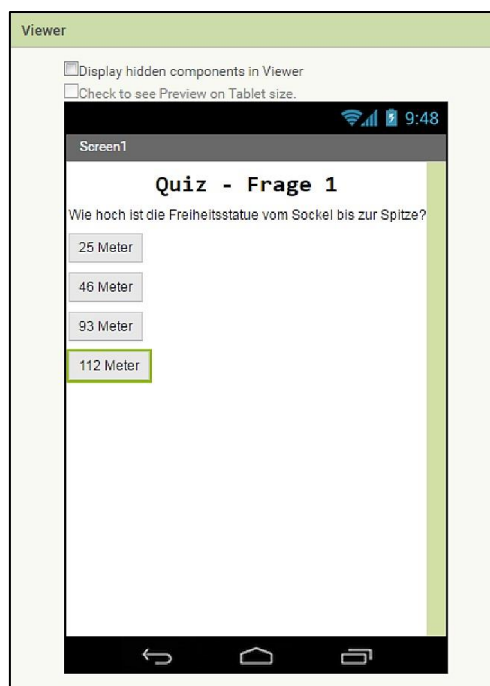
Anschließend kannst du den Titel ändern. Orientiere dich am besten an dem folgenden Bild.



Wichtige Eigenschaften (Properties) sind hier: FontBold (Schrift dick), FontSize (Schriftgröße), FontTypeface (Schriftart), Width (Breite des Labels), Text und TextAlignment (Textausrichtung).

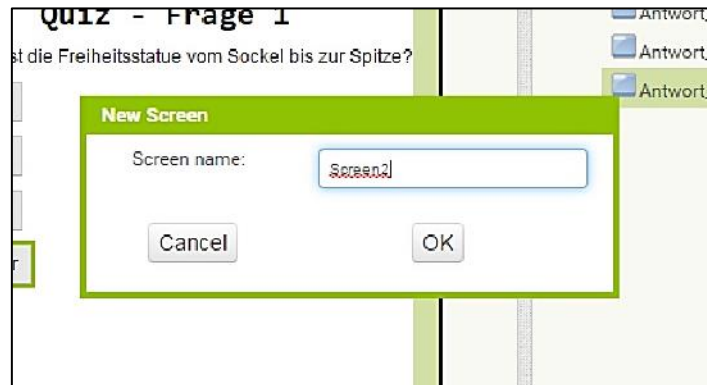
Schritt 5

Nachdem du den Titel angepasst hast, kannst du nun auch die anderen Komponenten bearbeiten. Orientiere dich auch hier am besten an dem folgenden Bild.



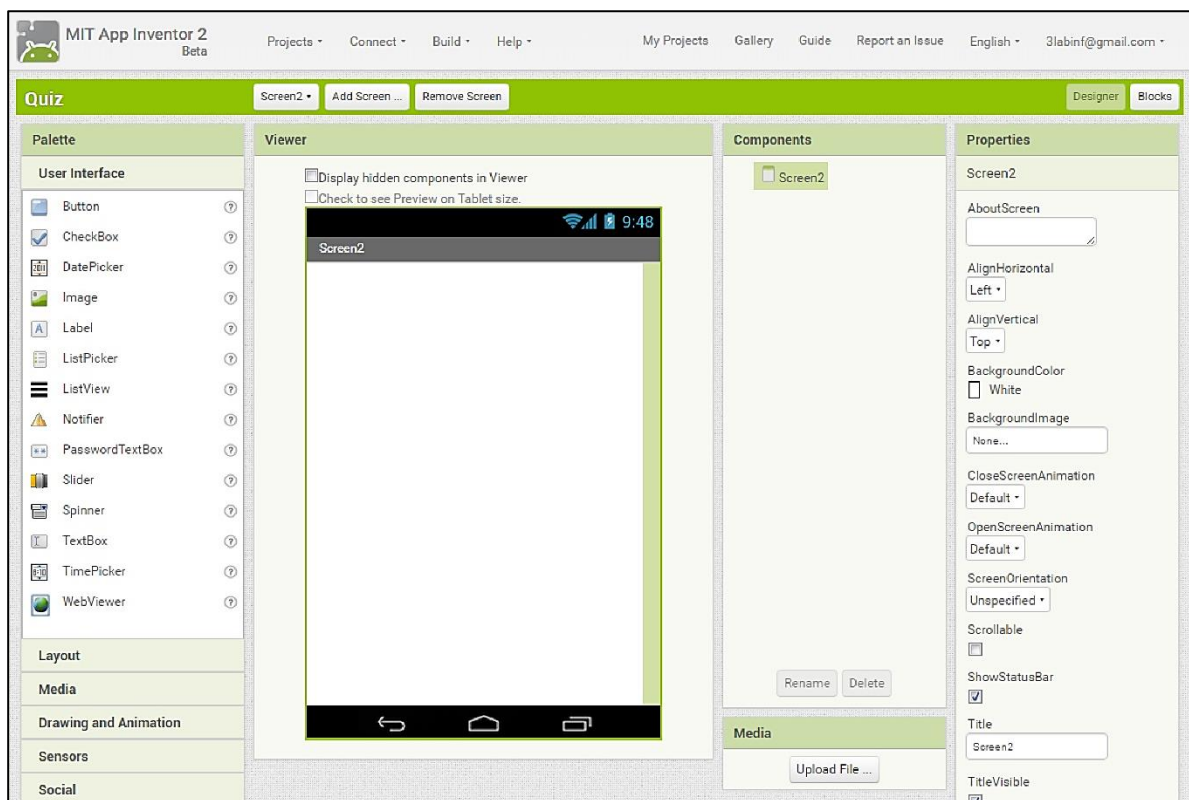
Schritt 6

Damit wäre die Grundlage für die erste Frage geschaffen. Um nun eine weitere Frage hinzuzufügen, brauchen wir einen neuen *Screen*. Klicke dazu auf *Add Screen* und lass den Namen am besten auf *Screen2* und drücke auf *OK*.

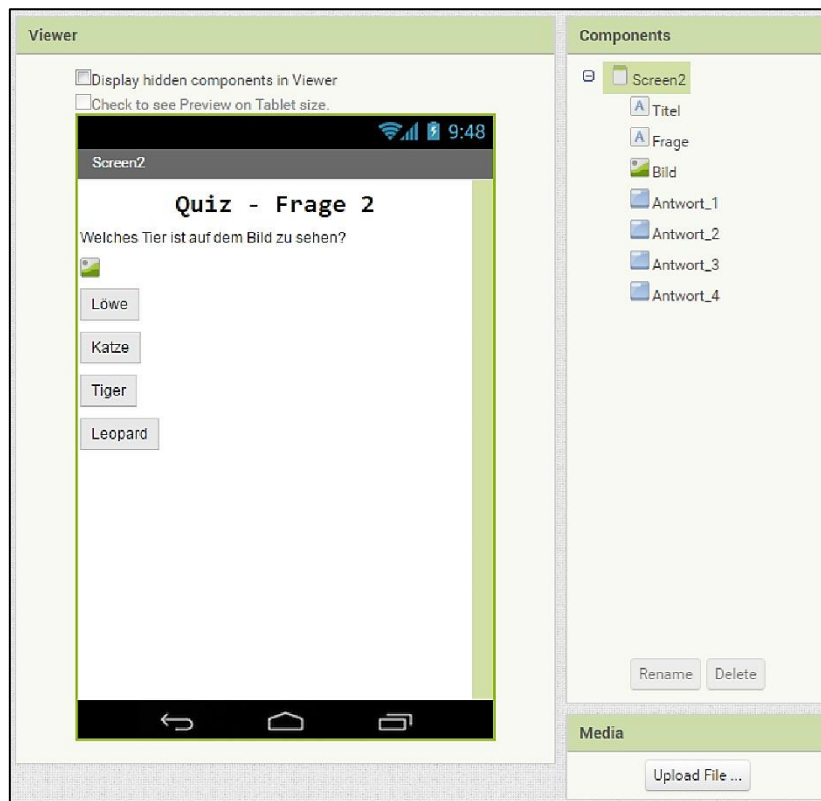


Schritt 7

Nachdem du auf *OK* geklickt hast, wird nun der neue *Screen* angezeigt.

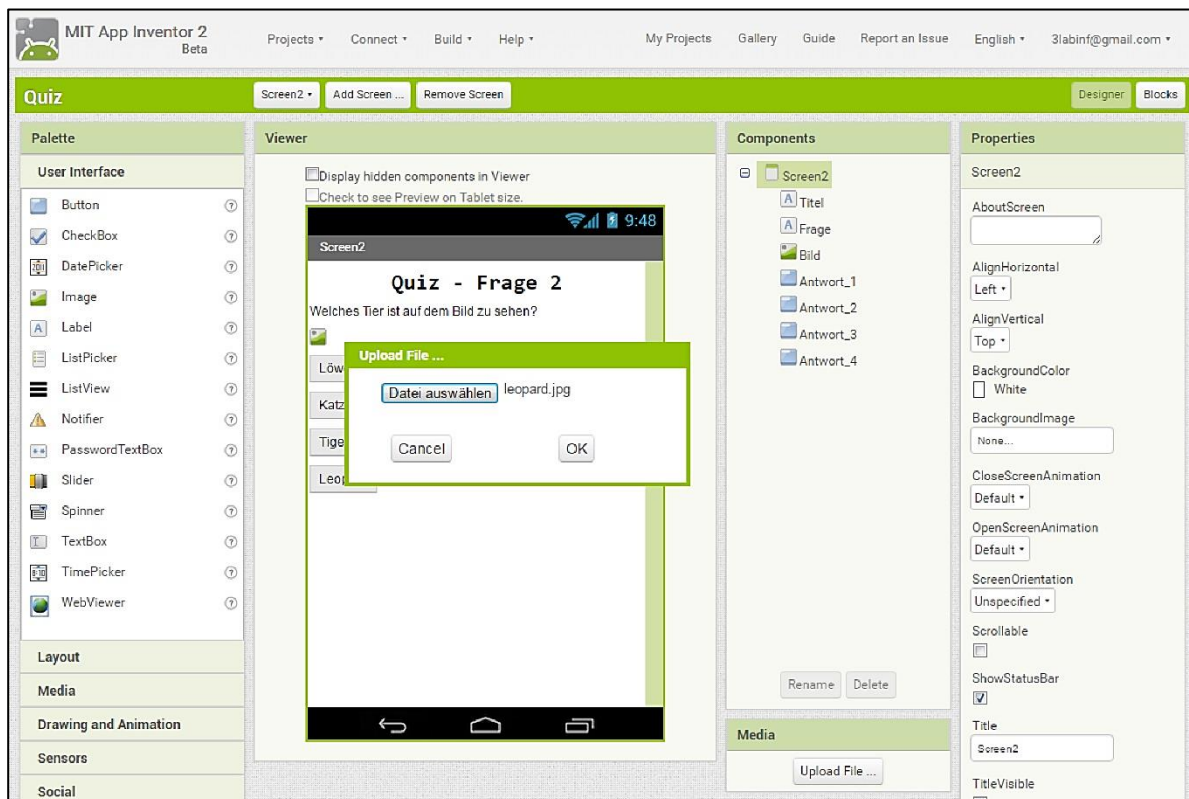


Du siehst, dass im *Viewer* und unter *Components* keines der vorherigen *Labels* und *Buttons* angezeigt wird. Daher musst du erneut zwei *Labels* und vier *Buttons* von der *Palette* in den *Viewer* des zweiten *Screens* ziehen. Ziehe anschließend aber zusätzlich auch ein *Image* (Bild) zwischen das zweite *Label* und den ersten *Button*. Dies wollen wir für die zweite Frage nutzen. Benenne die Komponenten dann am besten wieder so um, wie du es im ersten *Screen* gemacht hast und ändere auch deren Eigenschaften. Insgesamt sollte das dann wie auf dem folgenden Bild aussehen.



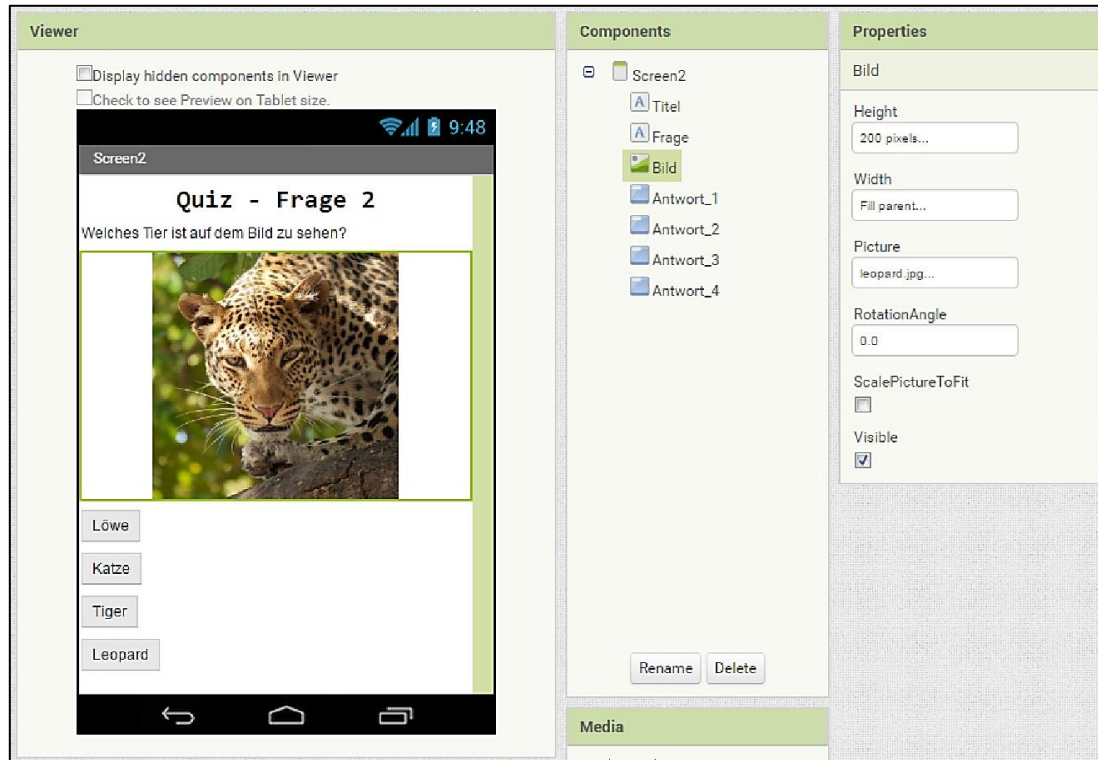
Schritt 8

Damit ein Bild angezeigt wird und die Frage somit auch beantwortet werden kann, musst du ein Bild im App Inventor hochladen. Such dazu erst einmal ein Bild eines Leopards im Internet und klicke anschließend auf *Upload File ...* um dieses Bild hochzuladen.



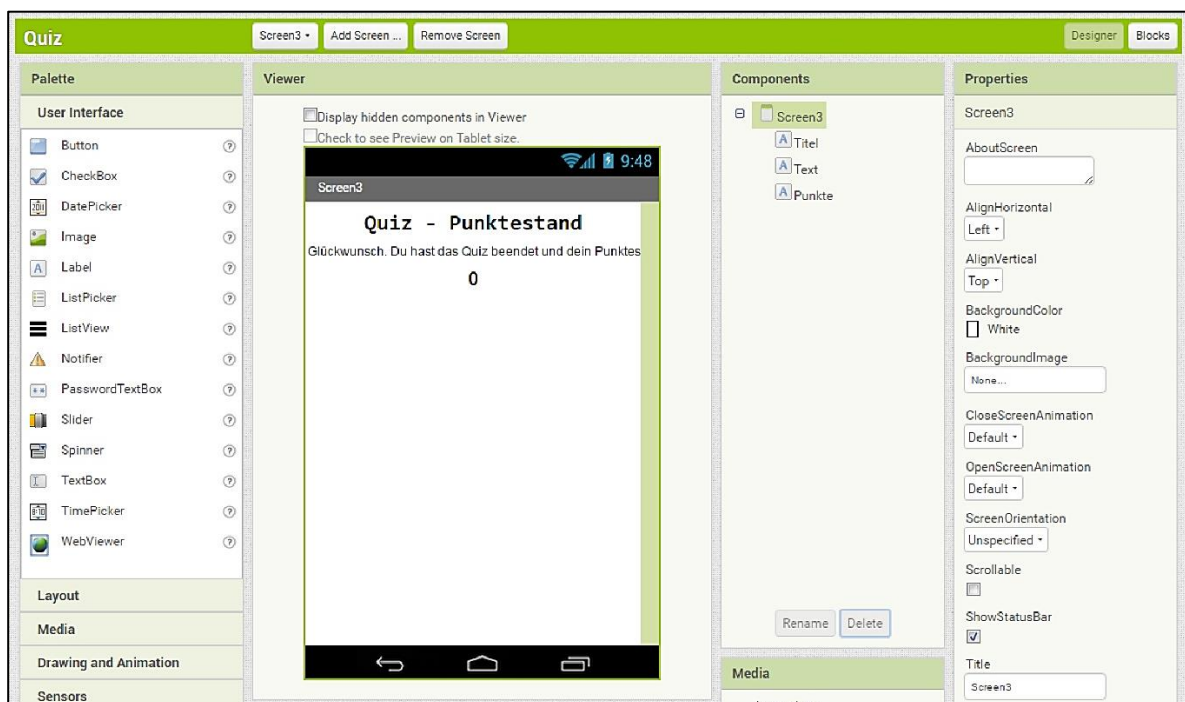
Schritt 9

Wähle dann die Image-Komponente aus und ändere die Eigenschaft *Picture*. Anschließend sollte das Ergebnis ungefähr so aussehen:



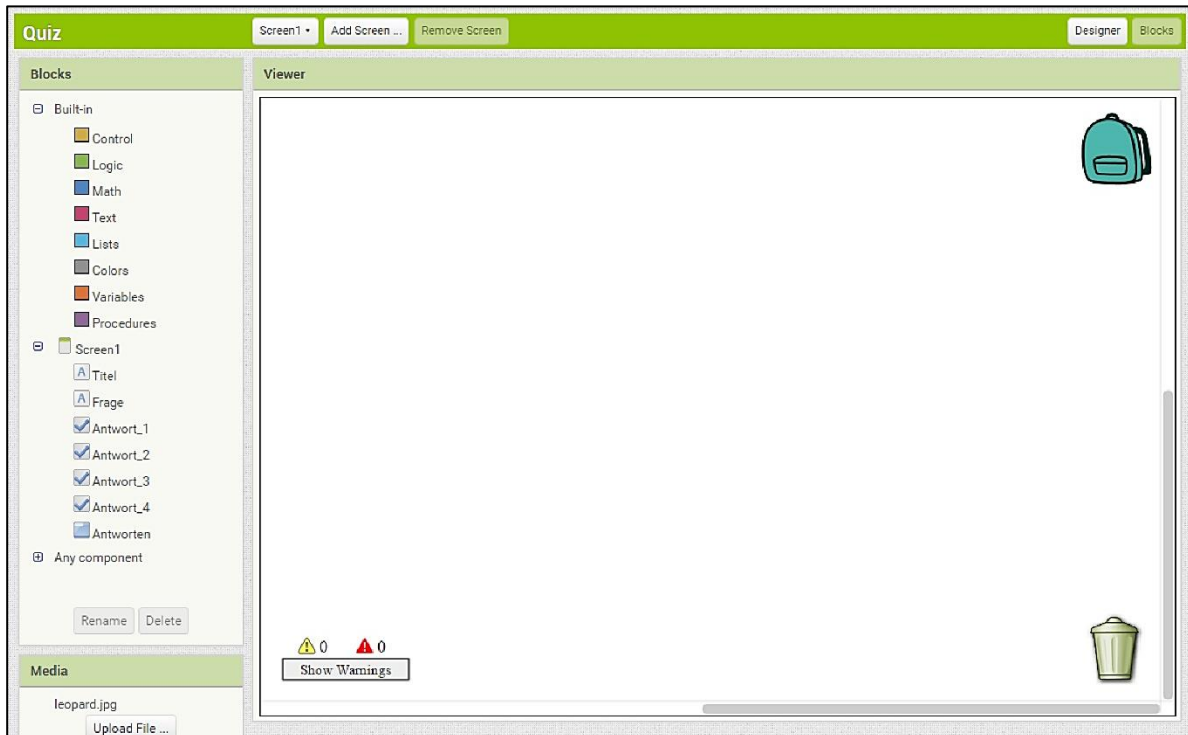
Schritt 10

Nun brauchen wir einen dritten und letzten *Screen* für den Punktestand des Quiz. Klicke dazu wieder auf *Add Screen* und füge einen neuen *Screen* mit Namen *Screen3* hinzu. Füge, sobald der neue *Screen* angezeigt wird, drei *Labels* hinzu und ändere deren Eigenschaften so, dass du folgendes Ergebnis erhältst:



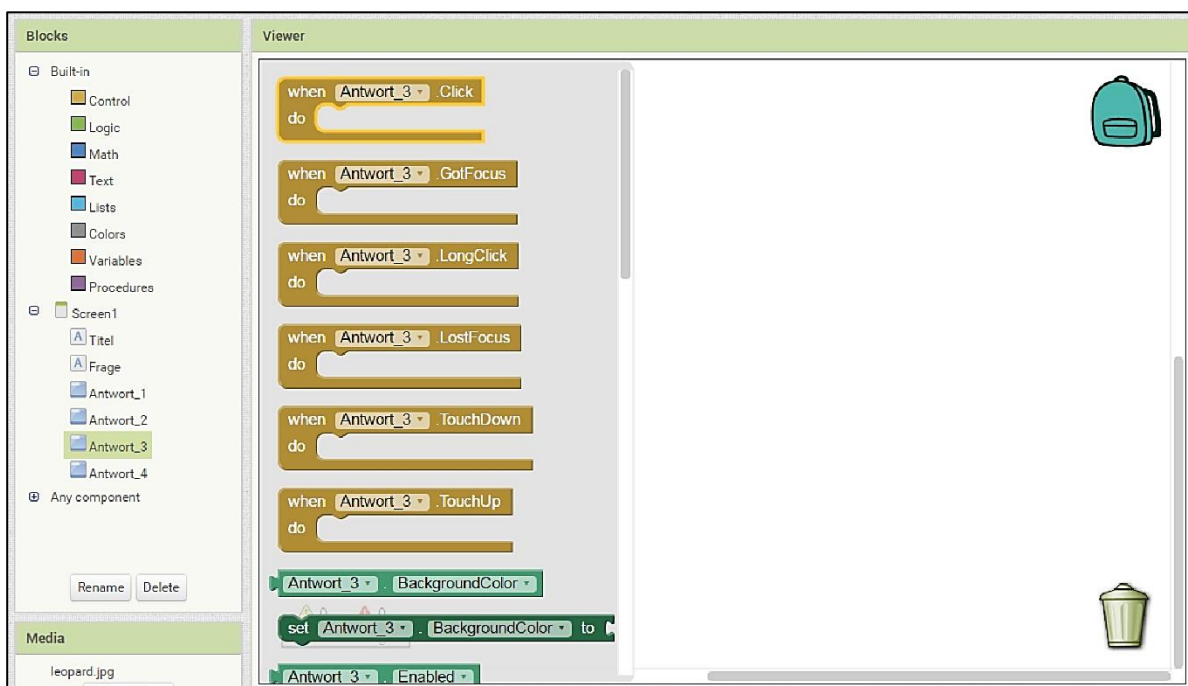
Schritt 12

Damit ist die grafische Vorlage des Quiz fertig und du kannst dich an die Programmierung machen. Wähle dazu *Screen1* aus dem Dropdown-Menü aus und wechsle dann in die *Blocks*-Ansicht.



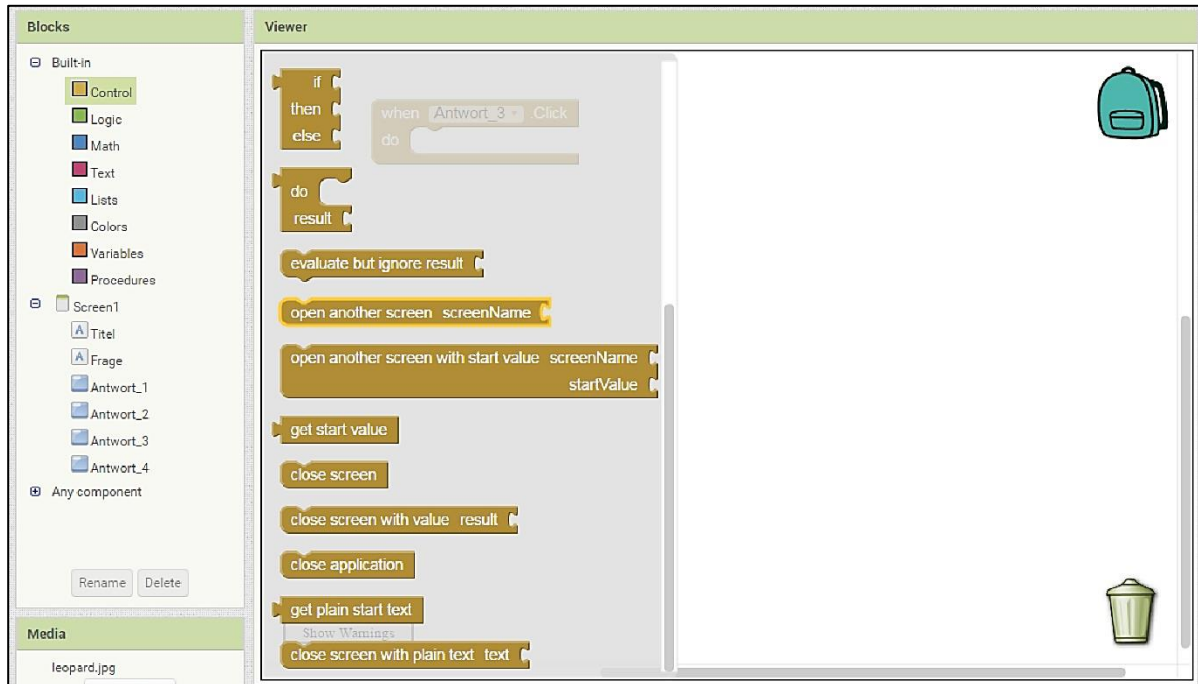
Schritt 13

Zunächst soll man beim Drücken der richtigen Antwort zur nächsten Frage kommen. Dazu brauchst du den Baustein, welcher für das Drücken von *Antwort_3* (das ist hier richtige Antwort) zuständig ist. Klicke unter *Blocks* auf *Antwort_3* und ziehe den Baustein *when Antwort_3.Click* in den Viewer.

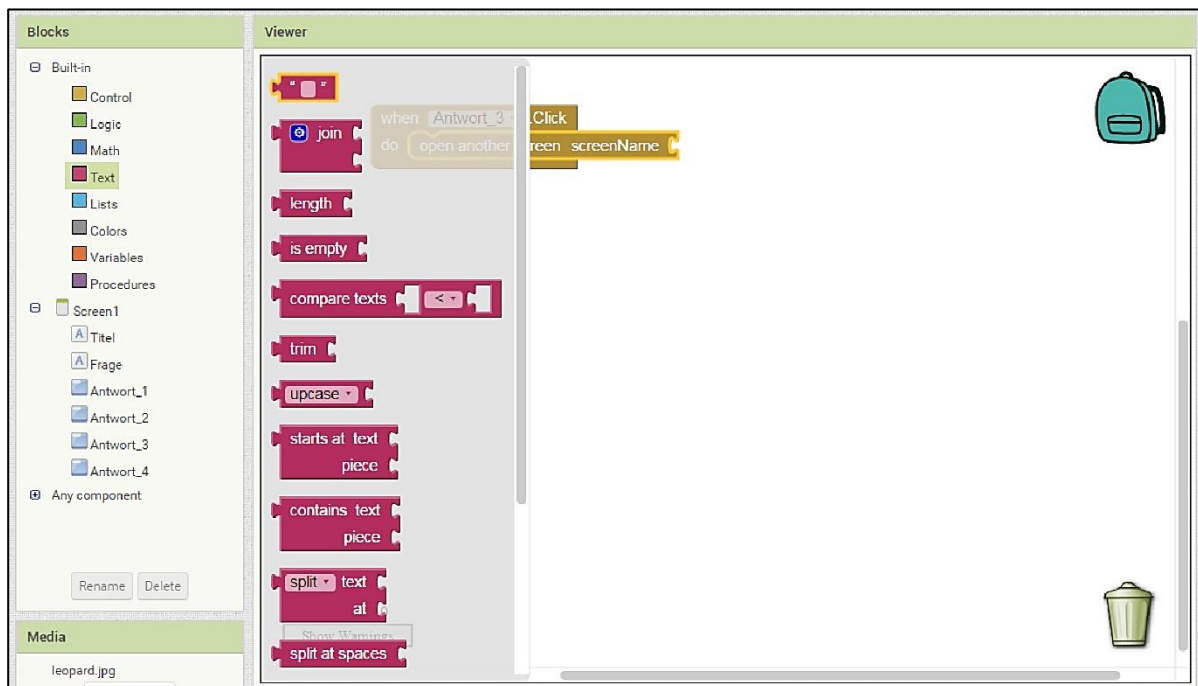


Schritt 14

Damit nun auch auf den zweiten Screen gewechselt wird, benötigst du noch den Baustein *open another screen*, den du unter Control findest.



Damit die App weiß, welcher Screen geöffnet werden soll, brauchst du noch den Baustein “ ” für eine *String* (Zeichenkette), welchen du unter *Text* findest.



Setze nun die Bausteine zusammen und ändere den Inhalt des *String*-Bausteines in *Screen2*. Das Ergebnis sollte so aussehen:



Schritt 15

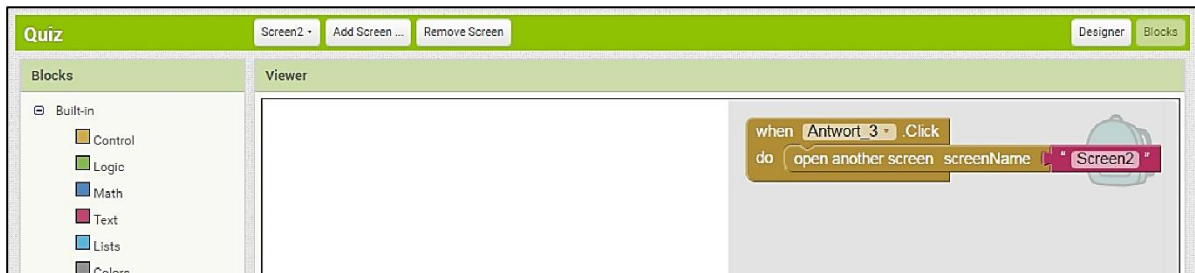
Nun kommt der Wechsel auf den dritten *Screen* bei richtiger Beantwortung der zweiten Frage. Damit die Arbeit aber einfacher wird, solltest du das Skript (die fertig verketteten Bausteine aus dem letzten Schritt) in den *Rucksack* in der oberen rechten Ecke des *Viewers* ziehen. Dies hat den Vorteil, dass du das *Skript* auch in anderen *Screens* aus dem *Rucksack* herausziehen kannst und dir damit Arbeit sparst.

Nachdem du das *Skript* in den *Rucksack* gezogen hast, solltest du folgendes Ergebnis haben.



Schritt 16

Wechsle nun mittels des Dropdown-Menüs in den *Screen2*. Du siehst, dass dieser *Screen* bisher keine Bausteine im *Viewer* besitzt. Klickst du aber auf den *Rucksack*, dann siehst du das *Skript*, welches du vorhin hineingezogen hast. Ziehe das Skript aus dem *Rucksack* in den *Viewer*.



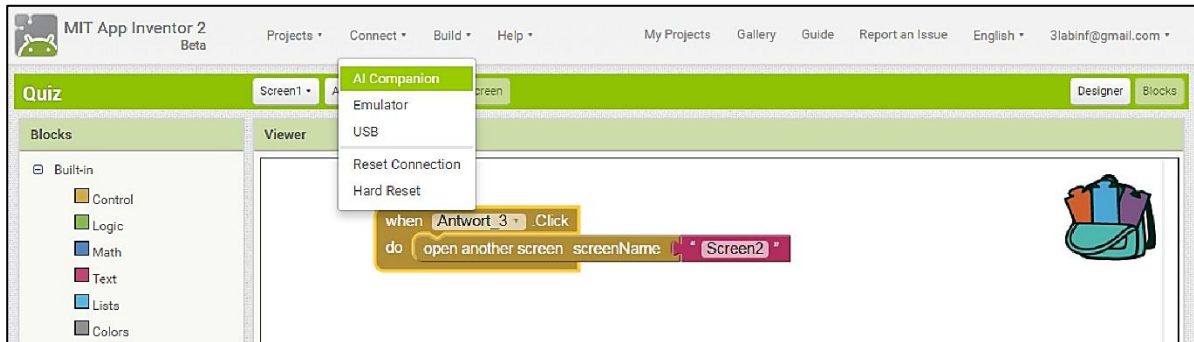
Schritt 17

Da bei der zweiten Frage die vierte Antwortmöglichkeit die Richtige ist, musst du das *Skript* etwas abändern. Ändere *Antwort_3* zu *Antwort_4* und *Screen2* zu *Screen3*. Es sollte danach so aussehen:



Schritt 18

Es wird Zeit für einen ersten Test der App. Wechsel dazu in den *Screen1* und klicke oben auf *Connect* und wähle *AI Companion* aus.

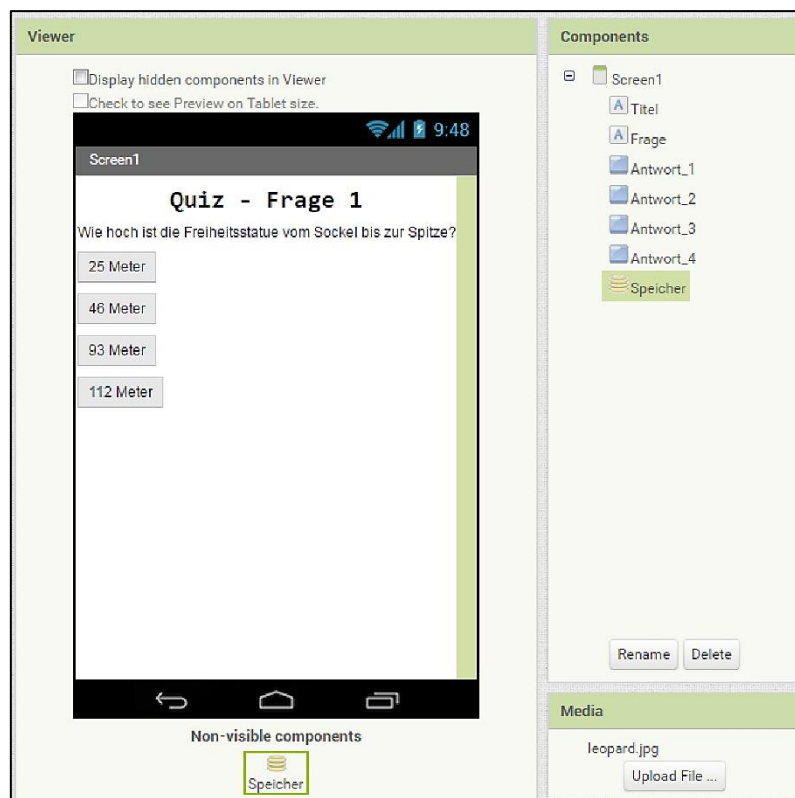


Es erscheint nun eine Mitteilung mit einem QR-Code. Scanne den QR-Code mit deinem Smartphone und teste deine App.

Schritt 19

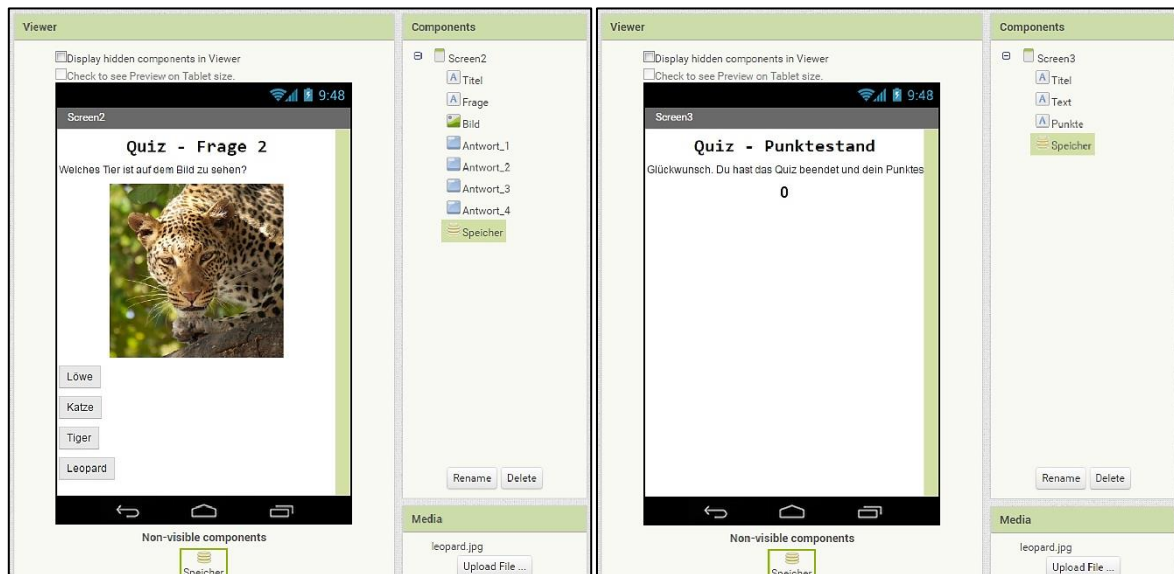
Deine App sollte nun eigentlich schon recht gut funktionieren. Es fehlt nun nur ein Punktestand und ein Hinweis, dass eine falsche Antwort gewählt wurde. Kümmere dich erst einmal um die Punkte.

Dazu brauchst du eine neue Komponente, weshalb du in den *Designer*-Ansicht wechseln musst. Wähle dann *Screen1* über das Dropdown-Menü aus, sofern dieses nicht bereits offen ist. Anschließend musst du *TinyDB*, welche du in der *Palette* und *Storage* findest, in den *Viewer* ziehen. Benenne diese Komponente dann in *Speicher* um. Das sollte insgesamt dann so aussehen:



Schritt 20

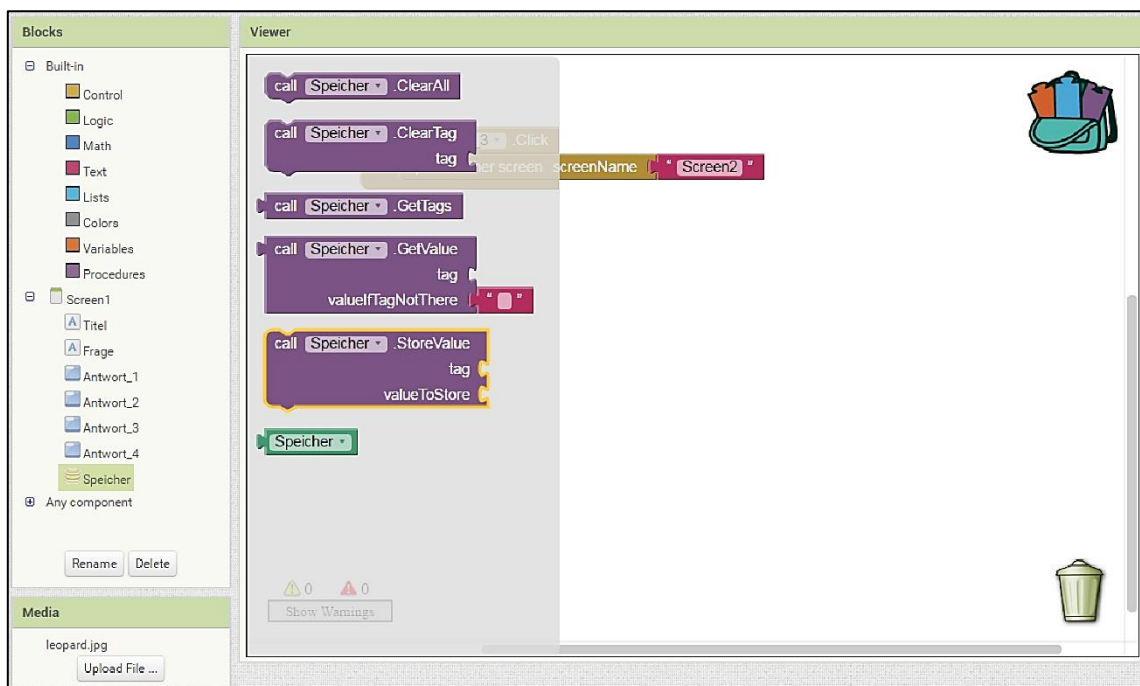
Wenn du im Dropdown-Menü nun den *Screen2* auswählst, wirst du sehen, dass dort die Komponente *TinyDB* nicht vorhanden ist. Wiederhole also das Hinzufügen der Komponente und das Umbenennen in *Speicher* auch für den *Screen2* und *Screen3*.



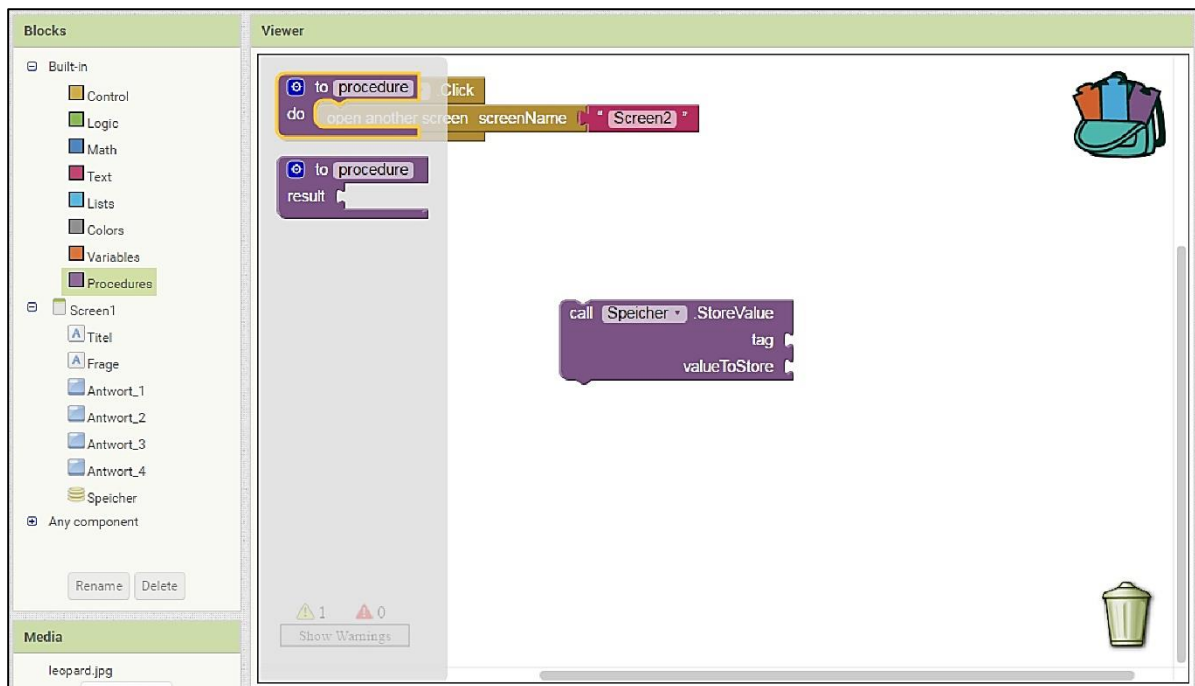
Schritt 21

Nun wechsele wieder in die *Blocks*-Ansicht von *Screen1* und es geht weiter mit der Programmierung.

Damit du dir wieder ein wenig Arbeit sparst, baust du nun eine Prozedur (Methode), die die Änderungen am Punktestand speichert, wofür du drei Bausteine brauchst. Der erste Baustein heißt *call Speicher.StoreValue* und befindet sich unter Block in der Komponente *Speicher*.



Der zweite notwendige Baustein heißt *to procedure do* und befindet sich unter *Procedures*.

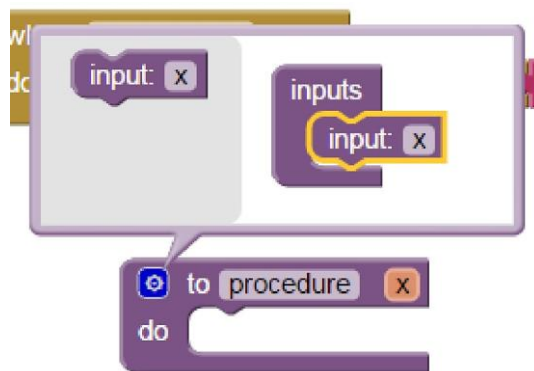


Der dritte und letzte notwendige Baustein ist *call Speicher.GetValue* und befindet sich auch bei der Komponente *Speicher*.

Schritt 22

Wenn du nun die drei oben erwähnten Bausteine in den Viewer gezogen hast, kannst du anfangen deine Prozedur und damit einen eigenen Baustein zusammenzusetzen.

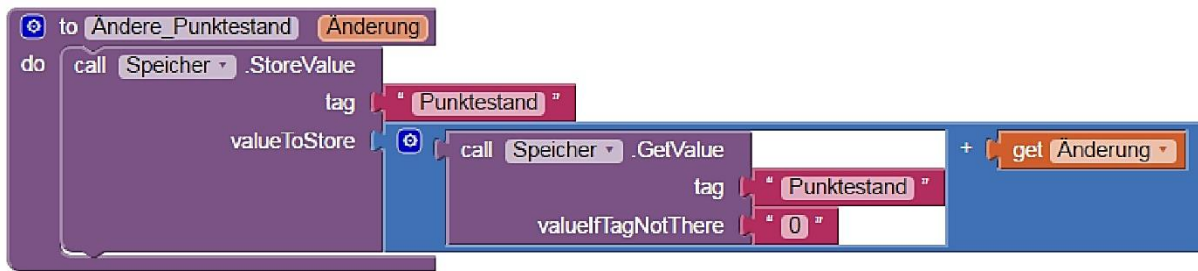
Wichtig ist hierfür der zweite Baustein *to procedure do*. Du hast bereits gesehen, dass bestimmte Bausteine bestimmte Werte angehängt werden können. Dies muss unser neuer Baustein auch können. Damit das geht, musst du auf das Zahrad klicken und in dem Fenster, das sich öffnet, den *input* Baustein einmal mit dem *inputs* Baustein verknüpfen (siehe Bild).



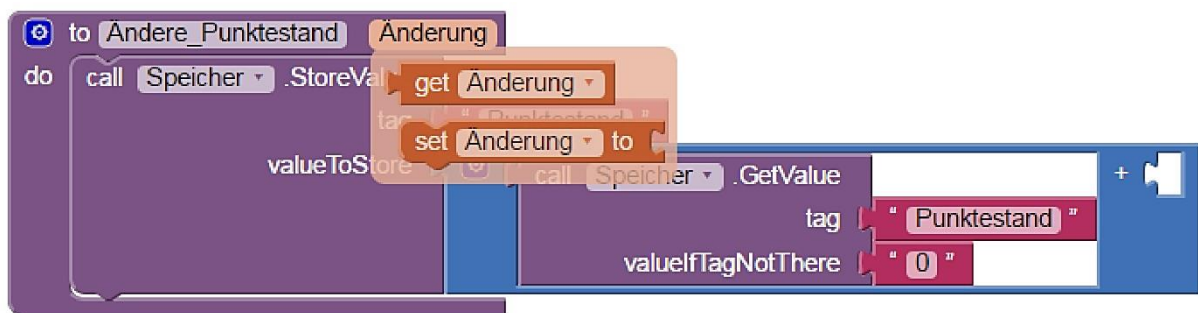
Anschließend kannst du den Text *procedure* und *x* des Bausteins in *Ändere_Punkttestand* und *Änderung* ändern, so wie auf dem folgenden Bild gezeigt.



Damit hast du deinen eigenen Baustein erstellt, der in der Kategorie *Procedures* nun auffindbar ist. Doch damit der Baustein auch etwas macht, musst du ihn noch mit den anderen Bausteinen aus dem vorherigen Schritt verknüpfen. Mache es wie auf dem folgenden Bild.

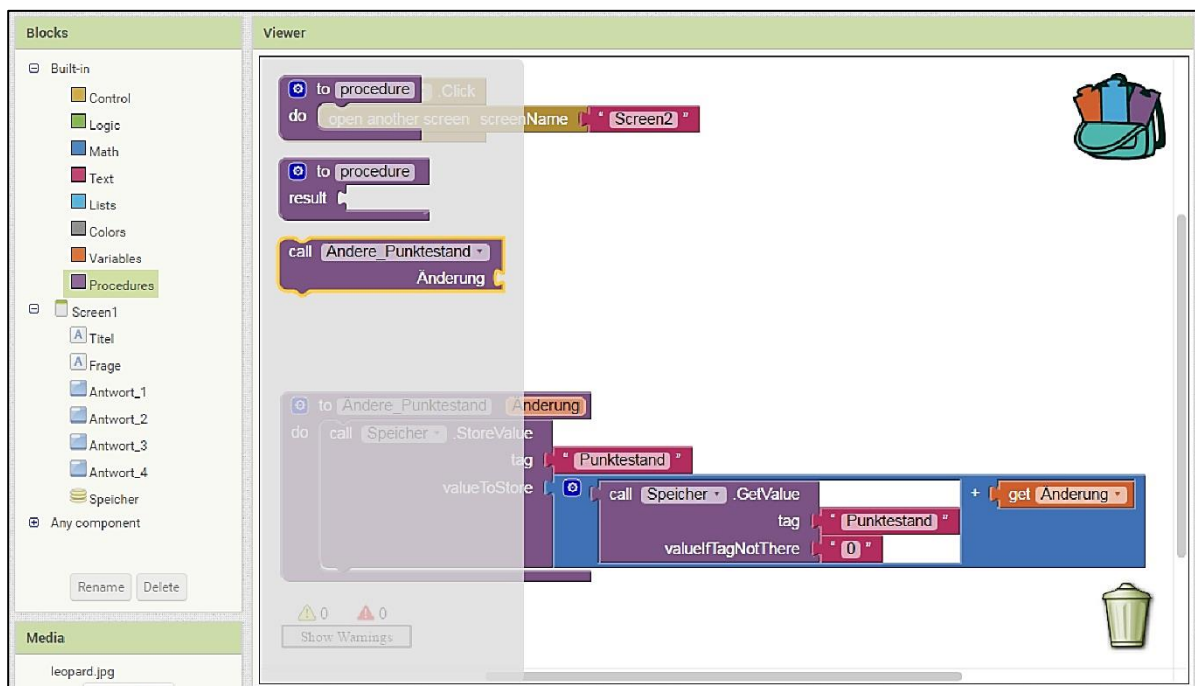


Dir fällt sicher auf, dass du noch zwei *String* Bausteine brauchst, die du unter Text findest. Außerdem brauchst du auch noch den *get Änderung* Baustein, den du bekommst, indem du auf *to Ändere_Punktstand Änderung* Baustein auf *Änderung* klickst (siehe Bild).

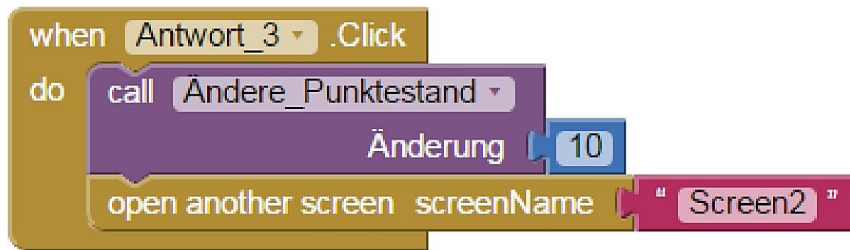


Schritt 23

Wenn du damit fertig bist, fehlt noch die Punkteänderung, wenn man auf die richtige Antwort klickt. Füge dazu den Baustein *call Ändere_Punktstand* aus der Kategorie *Procedures* in den Viewer.



Binde diesen Baustein und den für eine Zahl in den when Antwort_3.Click Baustein ein, so wie auf dem folgenden Bild gezeigt.



Schritt 24

Sobald nun *Antwort_3* auf dem *Screen1* angeklickt wird, wird der Punktestand um 10 erhöht. Damit bei einer falschen Antwort der Punktestand verringert wird und der Spieler erkennt, dass er eine falsche Antwort gegeben hat, braucht es noch etwas Arbeit von dir.

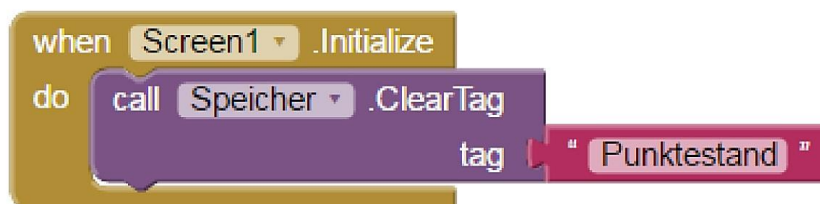
Füge nun Bausteine für die anderen Antworten auf *Screen1* hinzu und Sorge dafür, dass wenn diese Angeklickt werden, sich der Punktestand um 5 verringert, der *Button* nicht mehr aktiv ist (*Enabled* ist auf *false* gesetzt) und die Hintergrundfarbe des *Buttons* rot ist. Du kannst dich an dem folgenden Bild orientieren. Dort ist gezeigt, welche Bausteine du für *Antwort_1* verwenden musst.



Wenn du fertig bist, kannst du die Skripte für die richtige Antwort und für eine falsche Antwort (egal welche) und das Skript zu *Ändere_Punktestand* Änderung in den Rucksack ziehen, um dir für *Screen2* wieder etwas Arbeit zu sparen. Du bist nun mit *Screen1* fast fertig, es fehlt nur noch eine Kleinigkeit.

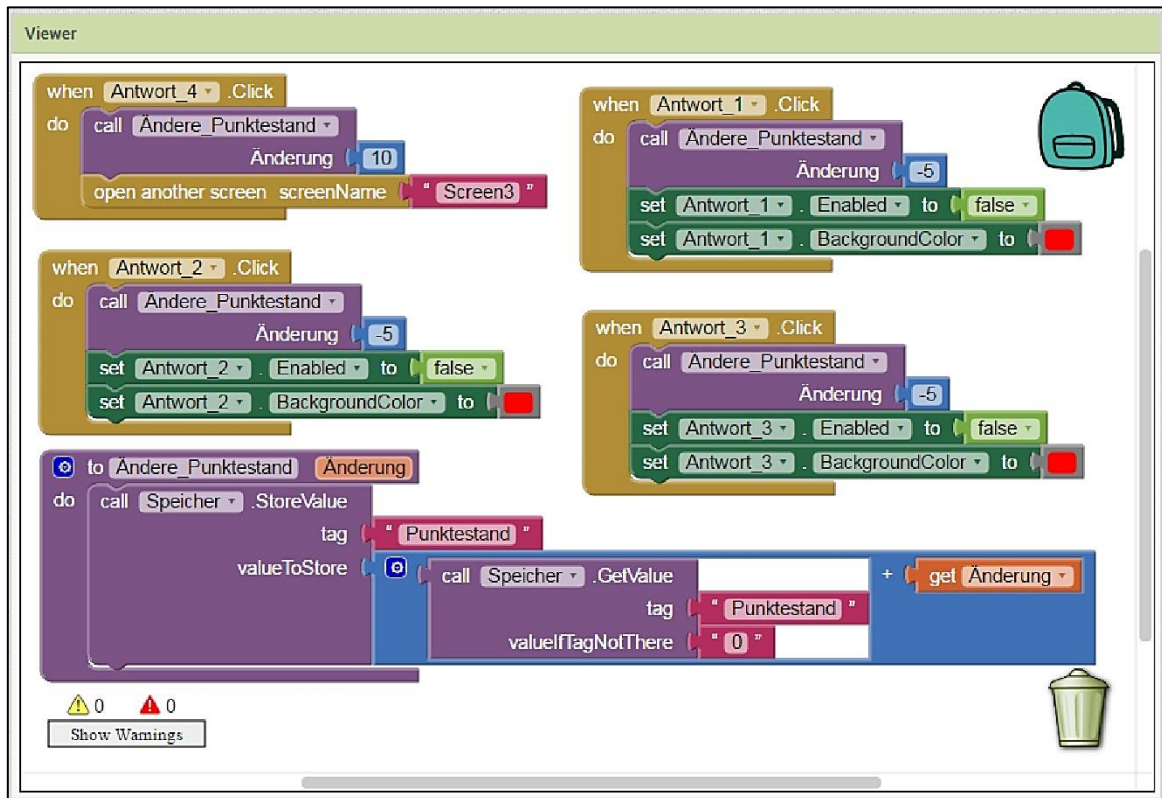
Schritt 25

Damit der Punktestand auch immer wieder bei 0 anfängt, sobald man das Quiz beginnt, brauchen wir nochmal drei Bausteine. Der erste Baustein ist *when Screen1.Initialize* und befindet sich unter *Screen1*. Der zweite Baustein ist *call Speicher.ClearTag* und befindet sich unter *Speicher*. Der dritte Baustein ist wieder mal der bereits bekannte *String* Baustein. Verknüpfe diese drei Bausteine wie auf dem folgenden Bild gezeigt.



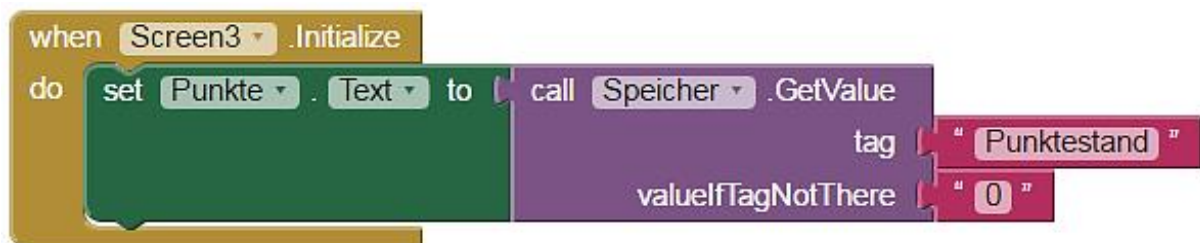
Schritt 26

Du bist nun mit *Screen1* fertig und kannst in der *Block-Ansicht* in den *Screen2* wechseln. Wenn du wie in Schritt 24 gesagt, die Bausteine in den Rucksack gezogen hast, kannst du diese Bausteine nun aus dem Rucksack in den Viewer ziehen. Anschließend musst du die entsprechenden Bausteine für den *Screen2* anpassen. Wichtig ist, dass hier *Antwort_4* für die richtige Antwortmöglichkeit steht. Wenn du fertig bist, dann sollte der Inhalt des Viewers ungefähr so wie auf dem folgenden Bild aussehen.



Schritt 27

Damit bist du auch schon mit *Screen2* fertig und es bleibt nur noch das Anzeigen des Punktestandes auf *Screen3*. Dafür musst du nur das Skript auf dem folgenden Bild nachbauen. Du findest die Bausteine bei den jeweiligen Komponenten *Screen3*, *Punkte* und *Speicher*.



Ende

Glückwunsch, du bist nun fertig mit deinem Quiz. Wie wäre es mit einem Test? Solltest du bereits die App auf dem Smartphone mittels *AI Companion* getestet haben und nun merken, dass sie sich nicht mit der im App Inventor aktualisiert hat, dann empfiehlt es sich die Verbindung neu aufzubauen. Wähle dazu auf *Reset Connection* im Reiter *Connect* und anschließend wieder auf *AI Companion*.