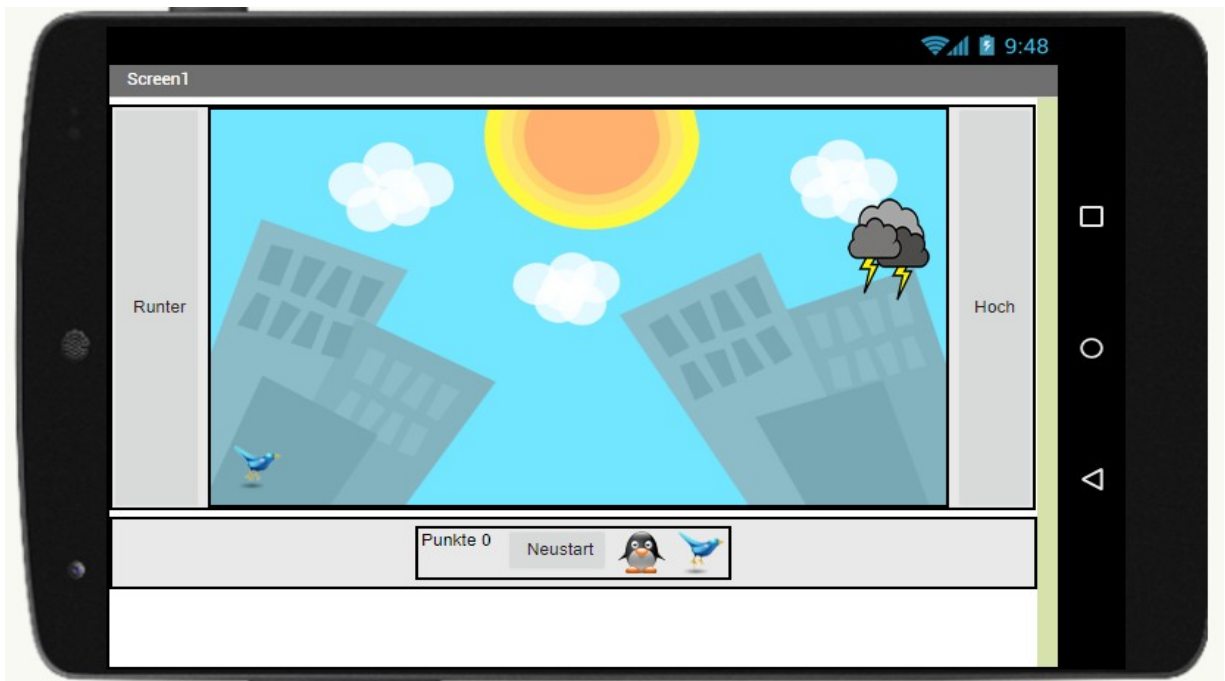





Spielprogrammierung als App


– FlappyBird –




HINWEIS

Ihr habt euch also für **FlappyBird** entschieden. Dies ist ein lustiges Spiel, bei dem es darum geht, mit eurem **Vogel**  den **Gewitterwolken** auszuweichen. Dieses Arbeitsblatt wird euch dabei helfen, eine App zu erstellen, die...

... euch einen  steuern lässt,

... **Wolken**  erzeugt, die sich auf euch zu bewegen und

... die **Punkte** hoch zählt, wenn ihr es schafft, einer Wolke auszuweichen.

Schwierigkeitsgrad: 

HINWEIS ANLEGEN DES PROJEKTS

Wie bei jeder neuen App müsst ihr zunächst ein neues Projekt anlegen.

Erstellt jetzt euer Projekt und gebt ihm einen ansprechenden Namen
Verbindet den App Inventor wieder wie auf dem ersten Arbeitsblatt
beschrieben mit dem Handy.

ACHTUNG DER AUFBAU EURER APP

Als nächstes müsst ihr euch Gedanken darüber machen, wie eure App aussehen soll. Eure App braucht Platz für folgende Dinge (die ihr in den folgenden Abschnitten der Reihe nach einbauen werdet):

- Eine **Spielfläche**, auf der sich der Vogel bewegen kann,
- einen **Reset-Knopf**, mit dem ihr das Spiel neu anfangen könnt,
- einen **Hoch-Knopf** und einen **Runter-Knopf** um den Vogel nach oben und unten zu bewegen und
- eine **Punkteanzeige**, die eure aktuellen Punkte hoch zählt.

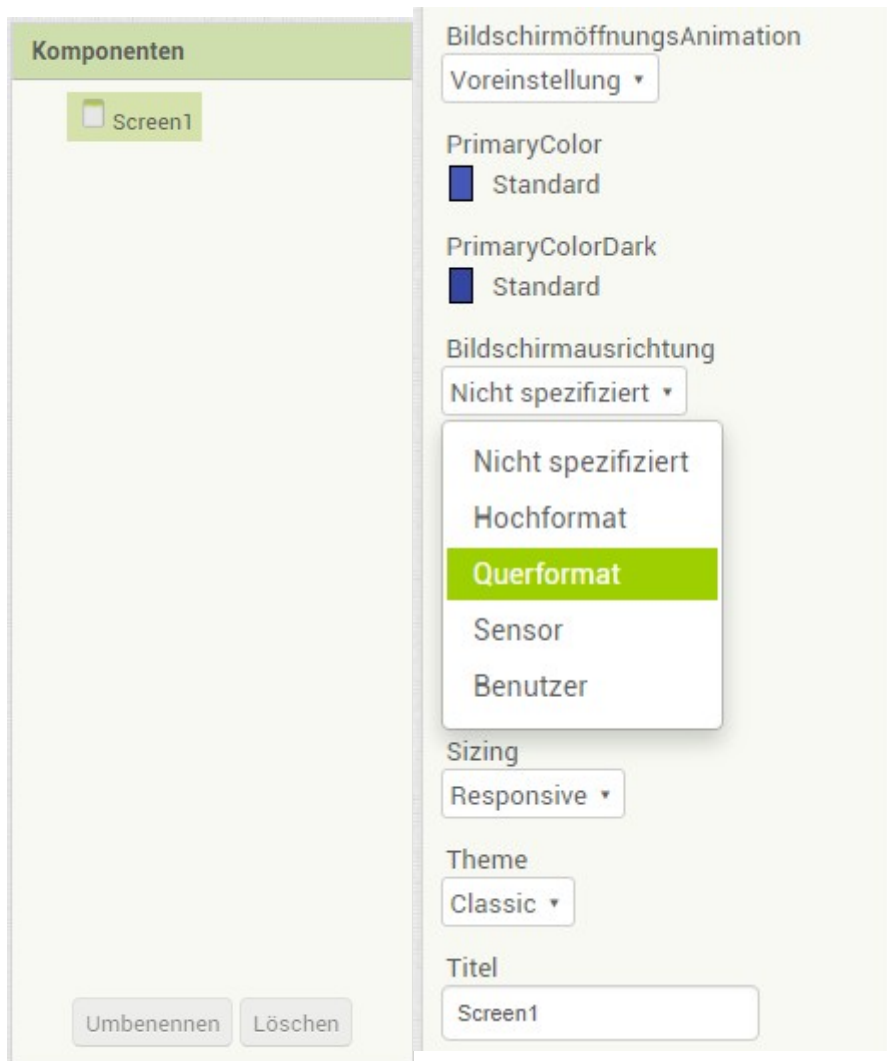
Das sind schon recht viele Sachen. Damit diese alle Platz haben, solltet ihr zunächst die Ausrichtung eurer *Bildschirm* von **NichtSpezifiziert** in **Querformat** ändern.

ACHTUNG

Mit **Querformat** ist bei Apps eine bestimmte Ausrichtung des Bildschirms gemeint, nämlich die horizontale Ausrichtung. Möchte man eine vertikale Ausrichtung haben, dann wählt man **Hochformat**.

AUFGABE

- 1.) Wählt euren *Screen1* unter **Komponenten** aus.
- 2.) Sucht unter **Eigenschaften** den Eintrag *Bildschirmorientierung*.
- 3.) Ändert den Wert von *Nicht Spezifiziert* in *Querformat*.



DIE BESTANDTEILE EURER APP

Nun könnt ihr anfangen, die einzelnen Bestandteile eurer App hinzuzufügen und diese mit Hilfe der vorgestellten **Bildschirmordnung** sinnvoll anzuordnen.

Für die **Spielfläche** braucht ihr ein **Zeichenfläche**.

Die *Zeichenfläche* findet ihr in der *Palette* im Unterpunkt *Zeichnen und Animation*.

HINWEIS

Eine Zeichenfläche ist eine Leinwand, auf der sich Objekte wie euer Vogel bewegen kann.

ACHTUNG

Für den **Reset-Knopf** braucht ihr eine **Taste** mit dem Text „Reset“.

- Für die **Bewegung** braucht ihr zwei **Tasten**. Einen beschriftet ihr mit „Hoch“ und die andere mit „Runter“.

- Und für die **Punkteanzeige** braucht ihr eine **Bezeichnung** mit dem Text „0“.

AUFGABEN

Macht euch Gedanken darüber, wie ihr sie positionieren wollt.

Zieht die *Bildschirmordnungen* (*diese findet ihr in Anordnungen*), die ihr für euer Layout braucht, hinein.

Fügt die Bestandteile (Zeichenfläche, Taste und Bezeichnung) eurer App hinzu.

Der erste Test

Nachdem ihr die Bestandteile sinnvoll angeordnet habt, könnt ihr direkt einmal testen, wie das Ganze auf eurem Smartphone aussieht.

AUFGABE

Startet dazu die App auf dem Smartphone wenn ihr dies noch nicht gemacht habt.

Gefällt euch die Anordnung? Wenn ja, könnt ihr weitermachen, ansonsten überarbeitet sie einfach nochmal.

AUFGABEN

Lasst die App einfach auf dem Smartphone laufen.

Der App Inventor aktualisiert alle Änderungen, die ihr vornehmt und zeigt sie euch direkt an, ohne dass ihr die App neu starten müsst.

HINWEIS

UMBENENNEN NICHT VERGESSEN

Da ihr jetzt die ersten Bestandteile eurer App fertig habt, ist es an der Zeit, diese wieder mit sinnvollen Namen zu versehen, damit ihr sie im Blöcke Editor besser unterscheiden könnt.

HINWEIS

EIN HINTERGRUND FÜR EURE SPIELFLÄCHE

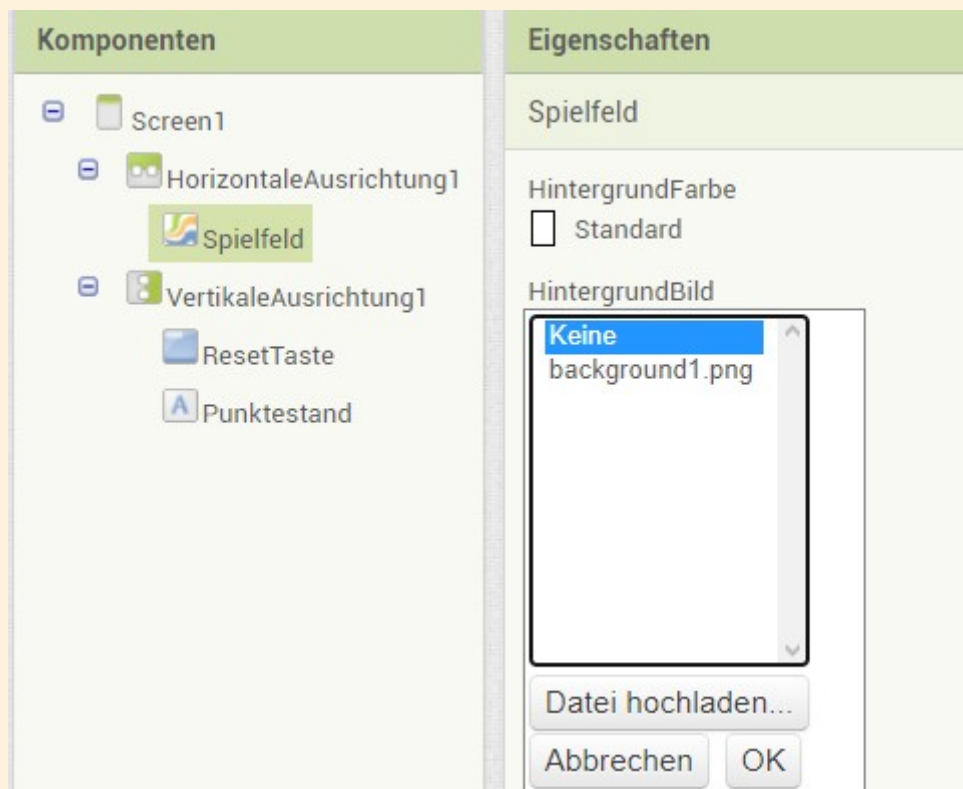
Damit der Vogel sich auf der Spielfläche wohl fühlt, soll diese einen schönen Hintergrund bekommen. Ladet dazu einen von zwei Hintergründen aus folgendem Ordner hoch: Desktop/EduInf-AppInventor

Anschließend müsst ihr eurer Zeichenfläche den Hintergrund nur noch zuweisen:

AUFGABE

Wählt die Zeichenfläche unter *Komponenten* aus.

Klickt unter *Eigenschaften* auf *HintergrundBild* und weist der Zeichenfläche das **background.png** als Hintergrundbild zu.



HINWEIS

Wie wir euch bereits erklärt haben, müsst ihr beim App Inventor manchmal die Größe per Hand einstellen.

AUFGABE

Ihr habt immer noch eure Zeichenfläche ausgewählt.

Sucht in den Eigenschaften nach Breite und Höhe und setzt diese jeweils auf 300 Pixel.

Testet jetzt eure App und ändert gegebenenfalls die Anzahl der Pixel, damit es auf dem Smartphone perfekt aussieht.

Höhe
300 pixels...

Breite
 Automatisch
 Fülle alles
 300 pixels
 percent
Abbrechen OK

DER VOGEL

Als nächstes soll der Vogel auf die Spielfläche.

AUFGABE

Wechselt in der Palette zu *Zeichen und Animation*.

Zieht ein **ZeichnenAnimation** Objekt direkt auf die *Zeichenfläche*.

Ändert den Namen des **ZeichnenAnimation** Objektes.

Ladet nun aus demselben Ordner einen der zwei Vögel hoch und weist das Bild der **ZeichnenAnimation** zu.

HINWEIS

Ein **ZeichnenAnimation** Objekt ist ein besonderes Bild. Dieses kann sich, im Gegensatz zu einem normalen Bild, auf einer Zeichenfläche bewegen.

Außerdem hat euer Vogel eine Position auf der Wiese. Diese könnt ihr in den Eigenschaften als **X- und Y-Koordinaten** ablesen bzw. verändern.

AUFGABE

Wählt den Vogel unter Komponenten aus.

Ändert unter Eigenschaften die X- und Y-Koordinaten und sucht euch eine schöne Position für euren Vogel aus.

Eine Wolke

Als nächstes braucht ihr eine Wolke auf der Spielfläche.

Wiederholt dazu die gleichen Schritte wie beim Erstellen des Vogels.

Denkt an das Bild der Wolke und an die Position.

HINWEIS EIN TIMER FÜR EURE WOLKEN

Als letzten Bestandteil eurer App benötigt ihr eine **Uhr**. Diese braucht ihr, um die Wolke zu bewegen.

Die *Uhr* findet ihr in der Palette unter *Sensoren*.

ACHTUNG

Eine **Uhr** ist für viele Sachen gut. Zum Beispiel könnt ihr ein **ZeitgeberIntervall** einstellen.

Dieses löst nach Ablauf der eingestellten Zeit immer wieder die Funktion **Uhr.Zeitgeber** aus.

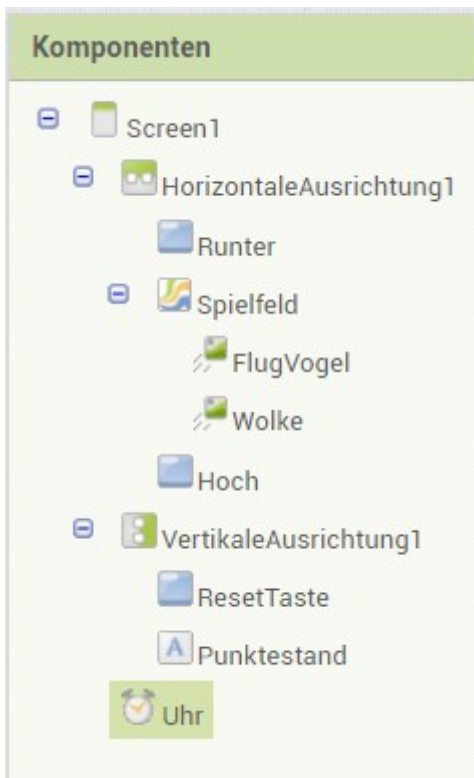
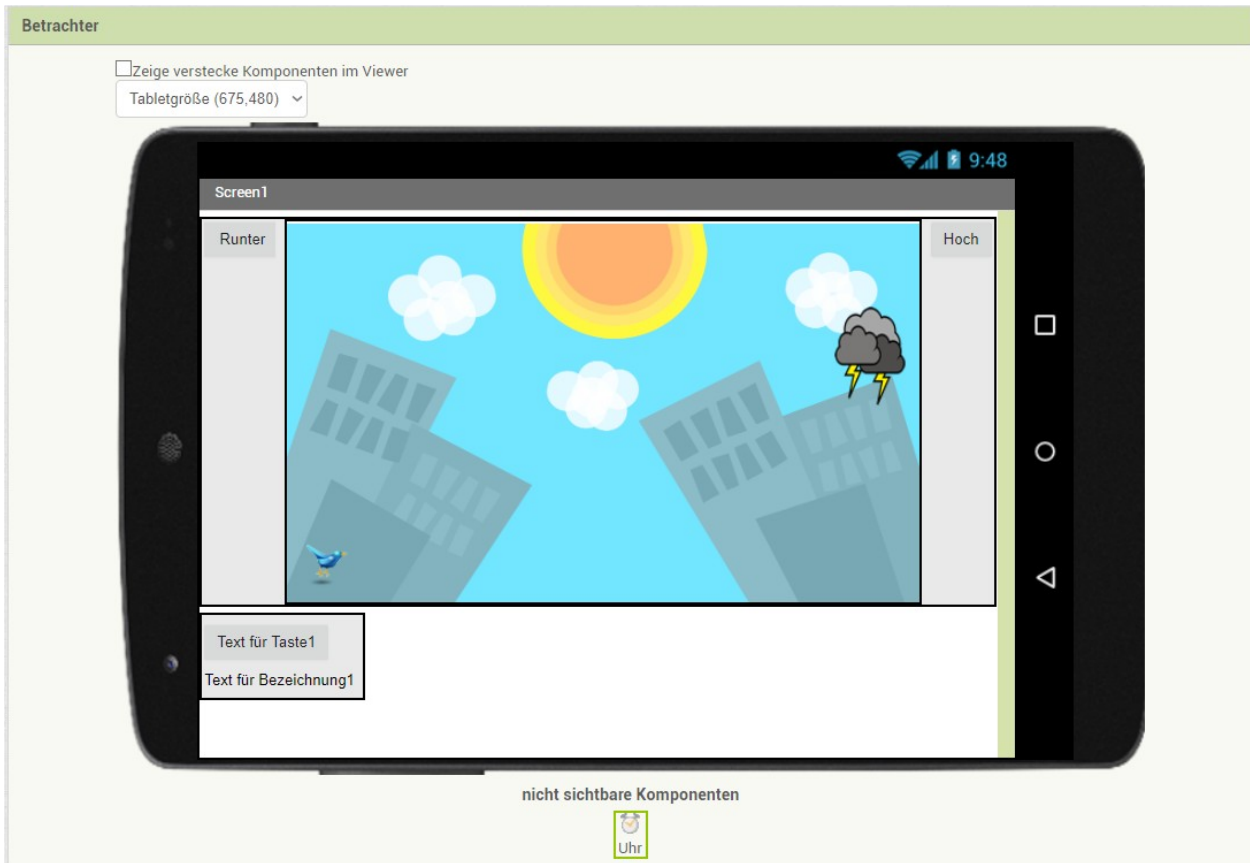
AUFGABE

Zieht eine **Uhr** in eure App.

Wählt die *Uhr* unter *Komponenten* aus und ändert den Wert für *ZeitgeberIntervall* in 500. (500 Millisekunden entspricht 0,5 Sekunden)

Ein kleines Zwischenfazit

So, oder so ähnlich sollte eure App jetzt aussehen. Falls ihr noch Fragen habt, spricht einfach kurz mit einem Betreuer.



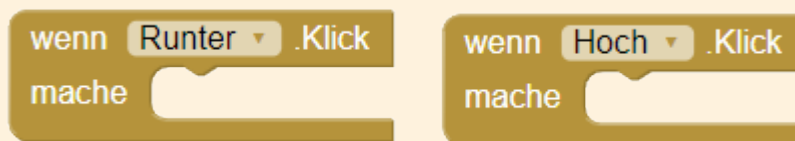
Die Bewegung des Vogels

Der Vogel kann sich in diesem Spiel nach oben oder unten bewegen. Daher müsst ihr diese beiden Bewegungen einbauen.

AUFGABE

Wechselt in den Blöcke Editor.

Wählt bei euren beiden Tasten **Hoch** und **Runter** die Funktion **wenn Hoch.klick / Runter.Klick** aus.

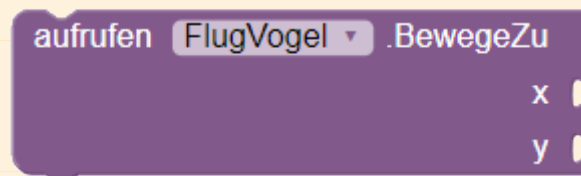


Wenn ihr den Vogel nach oben oder unten bewegt, dann ändert ihr dadurch die **y-Koordinate** eures Vogels. Der Ursprung (0,0) eures Koordinatensystems ist oben links in der Ecke der **Zeichenfläche**. Das heißt:

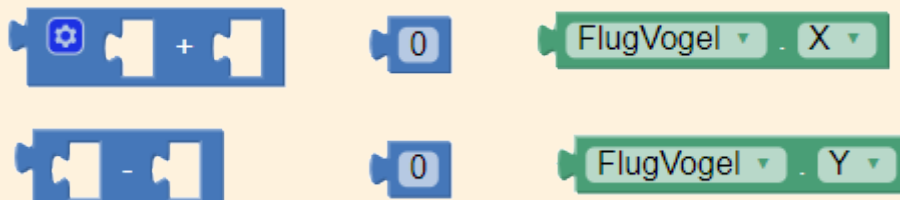
Wenn ihr den Vogel nach oben bewegen wollt, muss die y-Koordinate **kleiner** werden.

Wenn ihr den Vogel nach unten bewegen wollt, muss die y-Koordinate **größer** werden.

Jetzt braucht ihr die Funktion:



Mit diesem Block könnt ihr den Vogel an eine neue Position (x,y) bewegen. Damit kennt ihr jetzt alles, was ihr braucht, um den Vogel nach oben und unten zu bewegen. Außerdem benötigt ihr noch folgende Blöcke:



Versucht einmal selbständig mit diesen Blöcken die Bewegung nach oben und unten einzubauen. Zögert nicht, einen der Betreuer zu fragen.

Denkt daran, immer fleißig zu testen!

Die Bewegung der Wolken

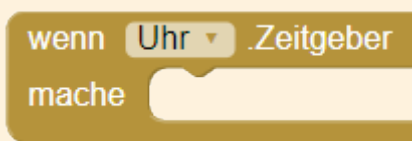
Für die Bewegung der Wolken braucht ihr die Uhr bzw. das von euch eingestellte **ZeitgeberIntervall**

Jedes Mal, wenn dieses ausgelöst wird, soll sich die Wolke etwas nach links bewegen.

AUFGABE

Wechselt in den Blöcke Editor.

Wählt die Funktion **wenn Uhr.Zeitgeber** aus



Ab jetzt ist der Vorgang identisch zu der Bewegung eures Vogels:

Die y-Koordinate soll **gleich** bleiben.

Die x-Koordinate soll **kleiner** werden, da sich die Wolke nach links bewegen soll.

Versucht einmal selbständig die Bewegung der Wolke einzubauen. Und auch hier gilt: „Zögert nicht, einen der Betreuer zu fragen.“

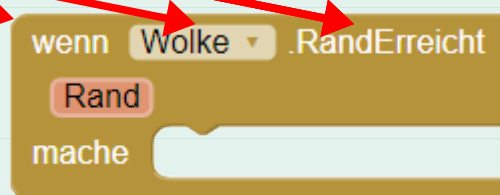
Tipp: Wenn eure Wolke sich zu langsam bewegt, müsst ihr das **TimerIntervall** im Design-Editor verändern.

HINWEIS

NEUE WOLKEN ERSCHEINEN LASSEN

Eine neue Wolke soll dann erscheinen, wenn die aktuelle Wolke die **linke Wand** erreicht hat. Dazu braucht ihr folgenden Block:

Wenn die **Wolke** die **Kante/Rand** erreicht hat, dann kann etwas **gemacht** werden:



Diesen Block könnt ihr schon einmal **auf die Arbeitsfläche ziehen**.

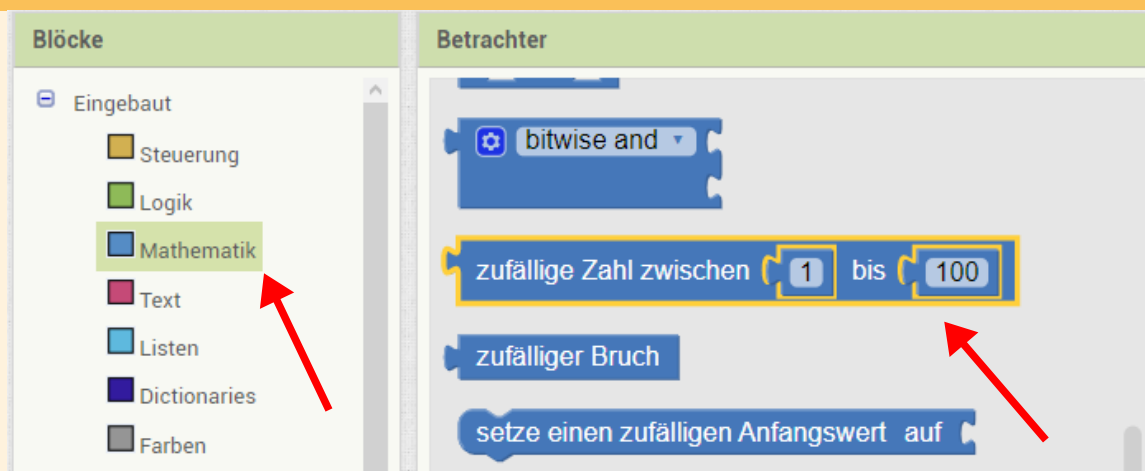
Neue Wolken

Um nun eine **neue Wolke** erscheinen zu lassen werdet ihr einen kleinen Trick anwenden:

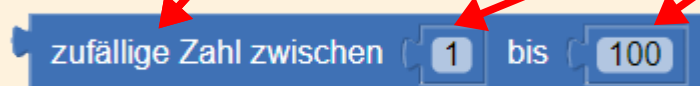
- Ihr wählt einfach eine neue zufällige Position für die Wolke im rechten Teil des Bildschirms aus.
- Dadurch sieht es für den Spieler so aus, als wäre eine neue Wolke rechts erschienen und die alte Wolke verschwunden.
- Denkt daran, dass nur die **Höhe** der Wolke zufällig sein soll. Das heißt, dass die **x-Koordinate** immer dieselbe sein soll aber die **y-Koordinate** jedes Mal zufällig.

Die neue Position (x,y) der Wolke soll also **zufällig** sein. Um das zu erreichen, besitzt der App Inventor die Möglichkeit, eine **Zufallszahl** zu generieren. Diese findet ihr im **Blöcke Editor** unter **Eingebaut** -> **Mathe**.

AUFGABE



Die Funktion wählt eine **zufällige ganze Zahl** im Bereich von **a** bis **b** aus, hier von 1-100.



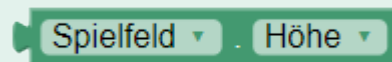
Der Bereich der Zufallszahlen

Euer Bereich für die Zahlen soll so groß sein wie die **Höhe des Spielfelds**.

Diese erhaltet ihr folgendermaßen:

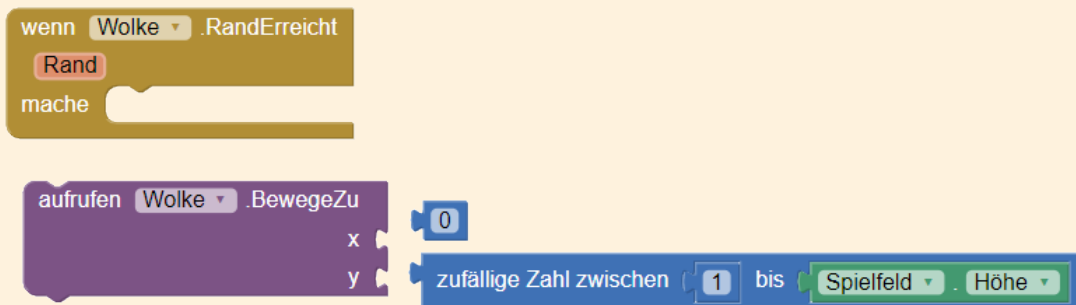
HINWEIS

Spielfeld.Höhe gibt euch die Höhe (Y-Koordinate) zurück.



AUFGABE

Erstellt nun die zufällige neue Position für die Wolke.
Benutzt dazu folgende Blöcke:



Testet eure App!

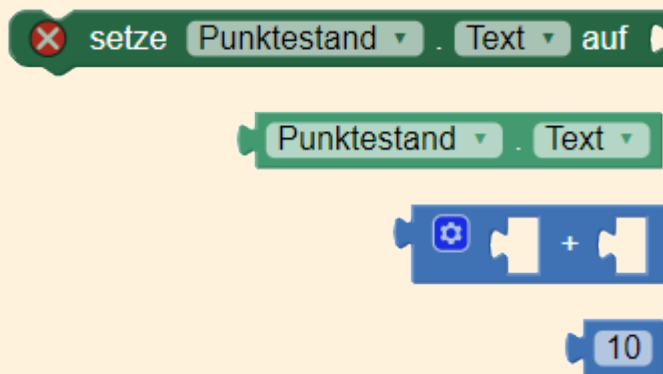
*Super! Bis hier ist es schon ein ganzes Stück Arbeit gewesen.
Jetzt kommt der Endspurt!*

Die Punkte hochzählen

Jedes Mal, wenn eine Wolke den linken Rand erreicht hat, sollen Punkte hochgezählt werden. Diese Funktionalität der App soll im folgenden Abschnitt implementiert werden.

AUFGABE

Sucht folgende Bausteine raus:



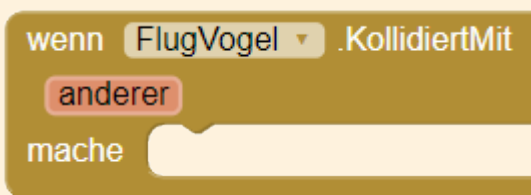
Macht euch Gedanken, wie ihr sie kombinieren könnt.
Fügt sie in die Funktion **wenn Wolke.RandErreicht** ein.
Testet euer Programm.

Die Kollision mit der Wolke

Als nächstes soll die Kollision mit der Wolke eingebaut werden. Diese ist gleichbedeutend mit dem **Spielende**. Die nötigen Schritte sind folgende:

AUFGABE

Sucht euch die Funktion **FlugVogel.KollidiertMit** heraus, diese wird bei der Kollision mit der Wolke ausgelöst.



Schreibt alles in diese Funktion hinein, was bei einer Kollision passieren soll, also:

Die Geschwindigkeit der Wolke auf 0 setzen.

Einen Text ausgeben. Dazu könnt ihr die Funktion **Spielfeld.ZeichneText** nutzen.

Der Reset-Knopf

Als letztes müsst ihr dafür sorgen, dass das Spiel **zurückgesetzt** wird.

AUFGABE

Dazu braucht ihr den **Reset-Knopf**. Wenn der Reset-Knopf gedrückt wird, dann sollen die Grundeinstellungen wiederhergestellt werden.



Die Grundeinstellungen

HINWEIS

Jetzt müsst ihr euch kurz überlegen, was die Grundeinstellungen sind. Dazu habt ihr folgende Hinweise:

- Der Punktestand hat eine Grundeinstellung.
- Die Position des Vogels kann auf den Startwert zurückgesetzt werden.
- Die Wolke hat eine Startposition.

AUFGABE

Überlegt euch, was alles zurückgesetzt werden muss.
Setzt eure Ideen in der Funktion **wenn ResetTaste.Klick** um.

```
wenn ResetTaste .Klick  
mache
```



Gratulation!

Damit habt ihr ein voll funktionsfähiges AngryBlob-Spiel programmiert. Auf der nächsten Seite findet ihr noch Tipps, Hinweise und Anregungen, wie ihr das Spiel erweitern könnt. Falls ihr lieber ein neues Spiel programmieren wollt, dann wendet euch an die Betreuer.

Erweiterungen

Hier sind einige Ideen, um das Spiel noch zu erweitern:

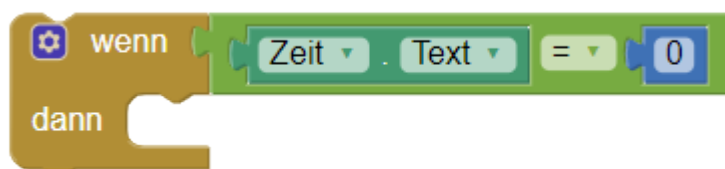
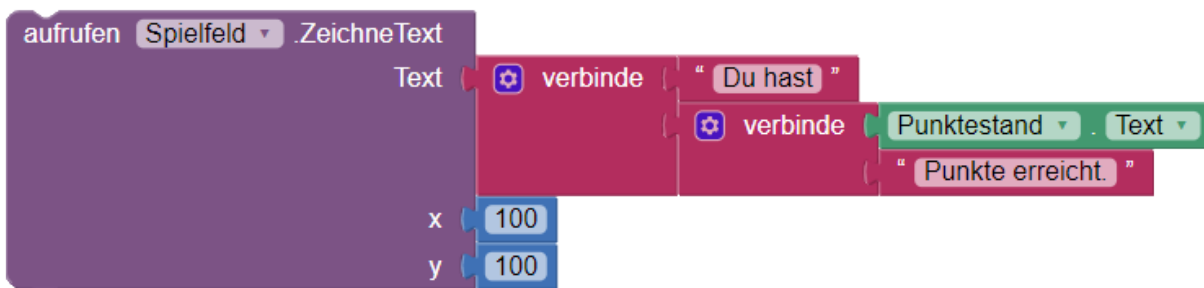
Mehrere Vögel und Hintergründe zur Auswahl

1. Um diesen Effekt zu erzielen, könnt ihr für jeden Hintergrund und jeden Vogel einen eigenen Button anlegen. Wenn ihr diesen Button drückt, soll sich das jeweilige Bild ändern:

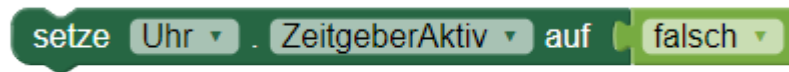


Ein Spielende

wäre noch sehr schön. Dazu könnt ihr ein neues Label hinzufügen, dass bei 100 anfängt. Jedes Mal, wenn der Timer auslöst, reduziert ihr den Wert um 1. Wenn der Wert 0 erreicht wird, könnt ihr bspw. das Spiel resetten oder einen Text ausgeben, der sagt, wie viele Punkte der Spieler bekommen hat.



Außerdem wäre es wichtig, den Timer auszuschalten, damit die Wolke stehenbleibt.



Tipp: Wenn ihr den Reset-Knopf drückt, muss der Timer wieder eingeschaltet werden und die Schrift vom Bildschirm entfernt werden.
Schaut euch dazu mal die Funktionen vom Spielfeld an, denn ihr wollt es **wischen**.

Quellenverzeichnis:





: <https://pixabay.com/de/pinguin-linux-klein-baby-vogel-48559/>



: <https://pixabay.com/de/wolke-gewitter-blitz-wetter-regen-149323/>



  : InfoSphere alle weiteren Grafiken sind Screenshots von App Inventor: (<http://appinventor.mit.edu/explore/>)