TECHNISCHE UNIVERSITÄT DRESDEN

Spieleprogrammierung als App

– Angry BloB –

Professur für Didaktik der Informatik

HINWEIS

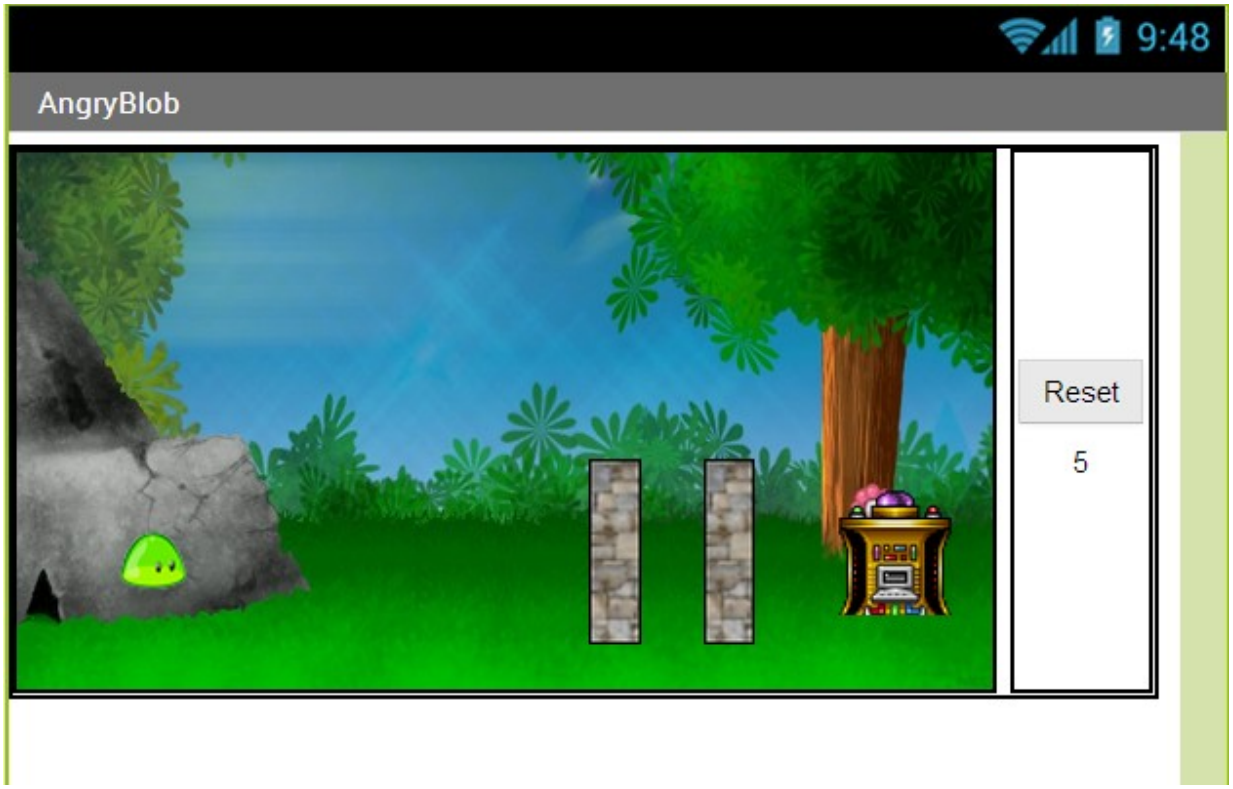
Ihr habt euch also für **AngryBlob** entschieden. Dies ist ein lustiges Spiel, bei dem es darum geht, den **Blob** zu werfen, um den Supercomputer zu zerstören.

Dieses Arbeitsblatt wird euch dabei helfen, eine App zu erstellen, die…

…einen auf einem **Spielfeld** erscheinen lässt.
…es dem **Spieler** ermöglicht, den durch **Wischen** zu werfen und
…die Verteidigung des Supercomputers bzw. den Supercomputer selbst zerstört, wenn er getroffen wird.

Schwierigkeitsgrad: ★★★★

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



AUFGABE

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

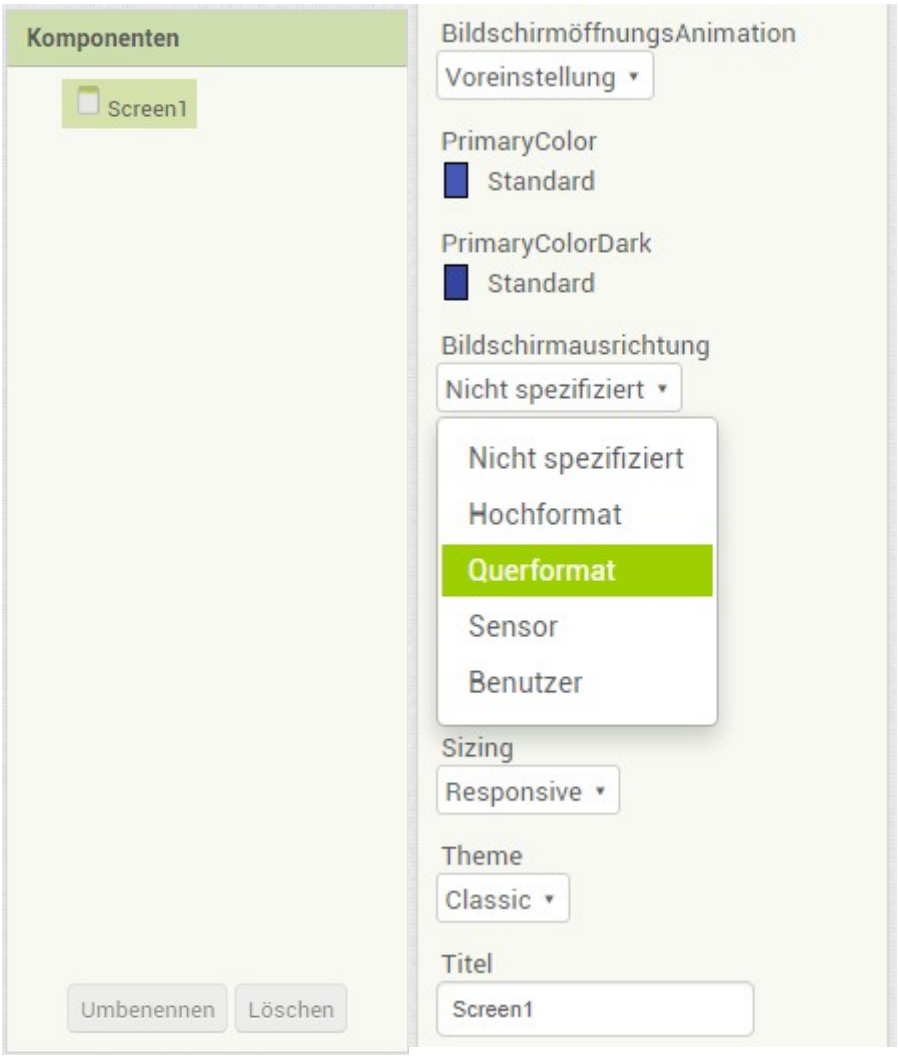
Als nächstes müsst ihr euch Gedanken darüber machen, wie eure App aussehen soll. Eure App braucht Platz für folgende Dinge (die ihr der Reihe nach einbauen werdet):
→ Eine ***Spielfläche,*** auf der sich euer ***Blob*** bewegen kann,
→ einen ***Reset-Knopf,*** mit dem ihr das Spiel neu anfangen könnt und
→ eine ***Lebensanzeige,*** die nach jedem Versuch ein Leben runterzählt.
Das sind schon recht viele Sachen.
Damit diese alle Platz haben, sollt ihr zunächst die Ausrichtung eures Screens (Bildschirmes) von **Nicht Spezifiziert** 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.*



HINWEIS

Die Bestandteile eurer App

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

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

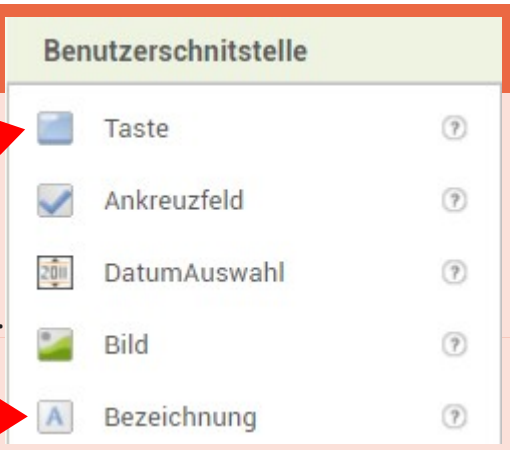
Die ***Zeichenfläche*** findet ihr in der *Palette* im Unterpunkt *Zeichnen und Animation.*

Eine ***Zeichenfläche*** ist eine Leinwand, auf der sich Objekte wie euer Blob bewegen können.

ACHTUNG

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

Und für die ***Lebensanzeige*** braucht ihr ein ***Bezeichnung (Label)*** mit dem Text „5“ bzw. der Anzahl an Leben, die euer ***Blob*** haben soll.



HINWEIS

→ Macht euch Gedanken darüber, wie ihr die Elemente positionieren wollt.
→ Zieht diejenige *Bildschirmanordnung*, 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.

- 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.

ACHTUNG

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

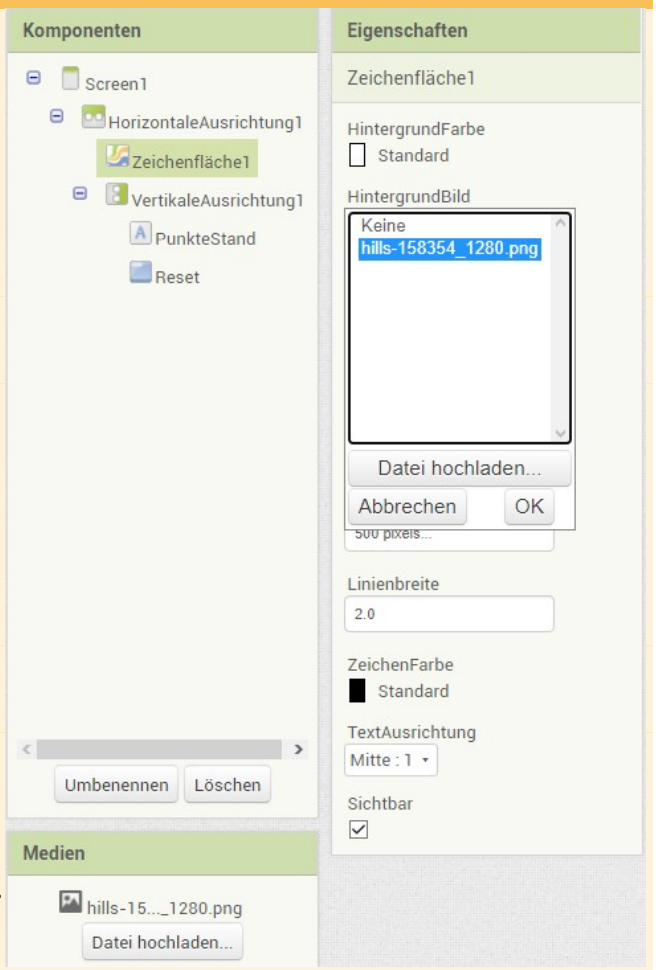
Umbenennen nicht vergessen

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.

AUFGABE
EIN HINTERGRUND FÜR EURE SPIELFLÄCHE

Damit sich der ***Blob*** auf der Spielfläche wohl fühlt, soll diese einen Hintergrund bekommen. Ladet dazu einen (von zwei möglichen) Hintergründen aus folgendem Ordner hoch:
Desktop / EduInf-AppInventor / AngryBlob

Anschließend müsst ihr eurer *Zeichenfläche* den Hintergrund nur noch zuweisen:



Wählt die Zeichenfläche unter Komponenten aus.

2.) Klickt unter *Eigenschaften* auf *HintergrundBild* und weist der *Zeichenfläche* ein Hintergrundbild zu.

HINWEIS

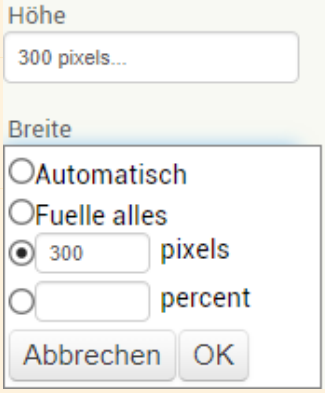
Wie wir euch bereits erklärt haben, müsst ihr beim App Inventor manchmal die Größe per Hand einstellen.

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.



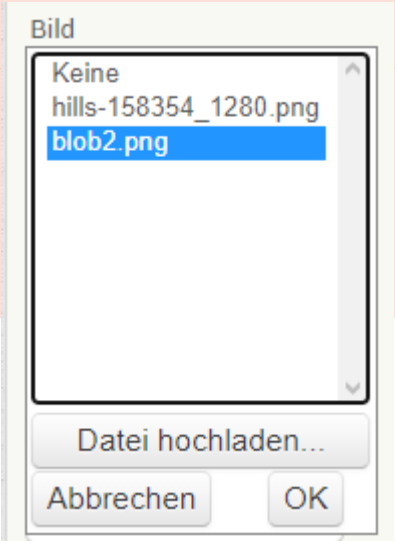
Als nächstes soll der ***Blob*** auf das Spielfeld.

HINWEIS
WECHSELT IN DER PALETTE ZU *„ZEICHNEN UND ANIMATION“.*

Zieht ein ***ZeichenAnimation*** *Objekt* direkt auf die Zeichenfläche.
Ändert den Namen des *ZeichenAnimation Objektes*.
Ladet nun aus demselben Bilderordner den ***Blob*** hoch und weist ihn dem ***ZeichenAnimation*** *Objekt* zu.

ACHTUNG

Eine ***ZeichenAnimation*** ist ein besonderes Bild. Dieses kann sich, im Gegensatz zu einem normalen Bild, auf einer Zeichenfläche bewegen.
Außerdem hat euer ***Blob*** eine Position auf dem Spielfeld. Diese könnt ihr in den *Eigenschaften* als ***X- und Y-Koordinaten*** ablesen bzw. verändern.



AUFGABE

Wählt den ***Blob*** unter *Komponenten* aus.
Ändert unter *Eigenschaften* die X- und Y-Koordinaten und sucht euch eine schöne Startposition für euren ***Blob*** aus.

HINWEIS
DER SUPERCOMPUTER UND DIE WÄNDE

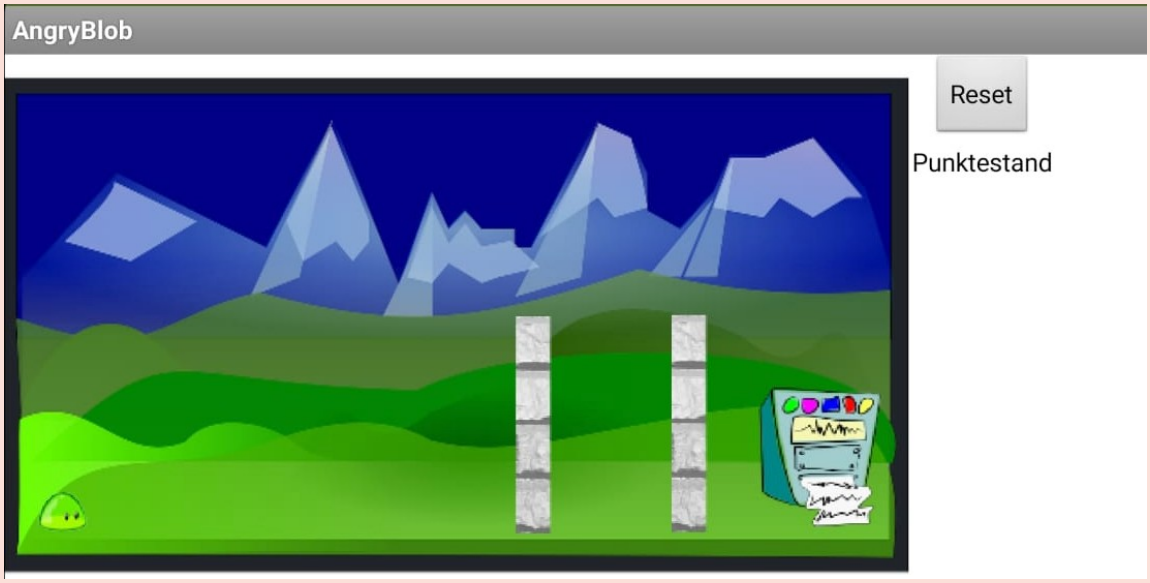
Neben dem ***Blob*** braucht ihr noch weitere ***ZeichenAnimation*** *Objekte*:
→ Ein ***ZeichenAnimation*** *Objekt* für den Supercomputer und
→ zwei *ZeichenAnimation* *Objekte* für die Wände.
Hinweis: Wenn ihr die Wände drehen wollt, müsst ihr den Wert „**Heading**“ in den **Properties** ändern. Allerdings wird euch das Ergebnis nur auf dem Smartphone angezeigt. (Probiert einmal die Werte 45 und 90 aus.)

AUFGABE

Verfahrt genau wie im vorigen Abschnitt und erstellt die ***ZeichenAnimation*** *Objekte* an einer sinnvollen Position.
Testet das Gesamtbild eurer App.
Die Bilder sind unter Desktop / EduInf-AppInventor / AngryBlob abgelegt.

ACHTUNG
EIN KLEINES ZWISCHENFAZIT

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



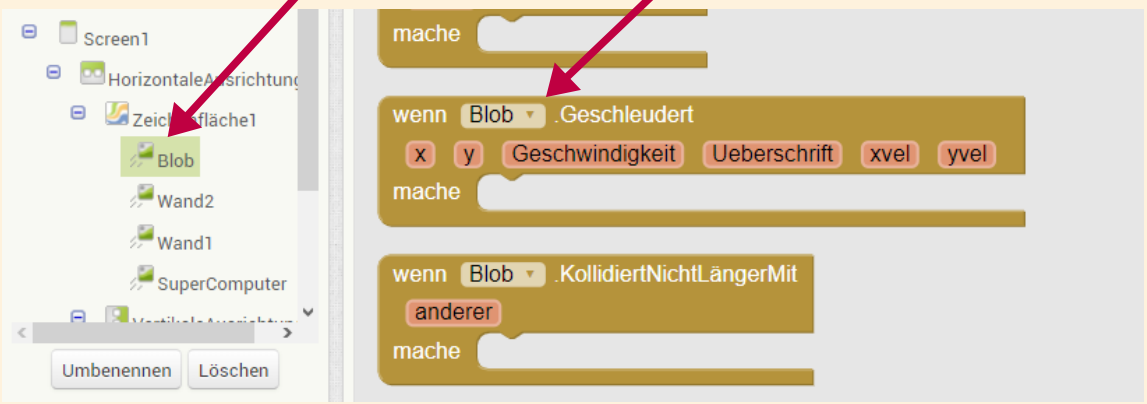
Die Bewegung des Blobs

Als erstes soll der *Blob* durch eure Wischbewegung losfliegen.

AUFGABE

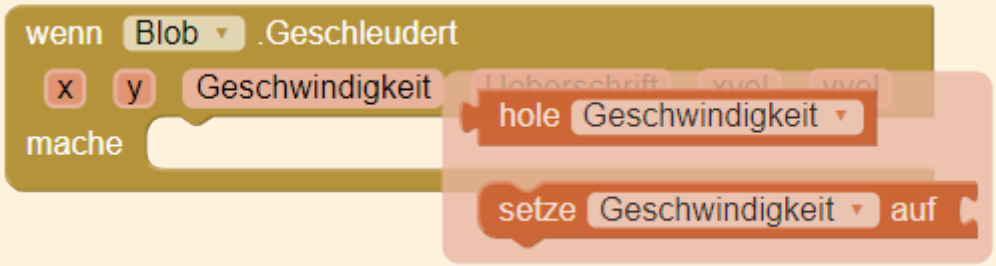
Wechselt in den Blöcke *Editor (rechts über den Eigenschaften)*.

Wählt bei eurem ***Blob*** den Block ***wenn Blob.Geschleudert*** aus.



Wenn der Blob geschleudert wird, dann kann etwas ge**macht** werden.

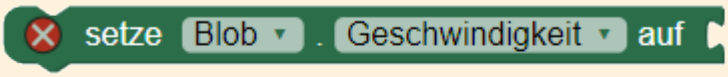
Außerdem werden durch diese Funktion noch Variablen erzeugt, die für euch wichtig sind.
Nämlich die *Geschwindigkeit* und die Richtung (hier *Ueberschrift*). Diese könnt ihr erhalten, wenn ihr den Mauszeiger auf den Namen der Variablen bewegt.



Jetzt müsst ihr eurem ***Blob*** nur noch die neuen Werte für die Geschwindigkeit und die Richtung (*Ueberschrift*) zuweisen.

Versucht, die Blöcke selbständig zu kombinieren und fragt einen Betreuer, falls ihr Hilfe benötigt.

Testet die Wischfunktion.



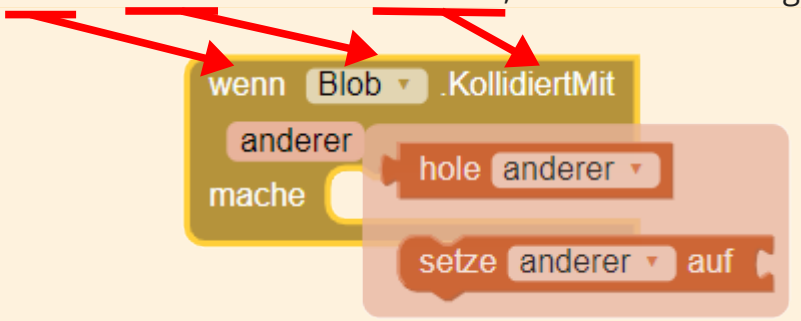
HINWEIS
AUSWERTUNG NACH JEDEM VERSUCH

Der Wurf eures ***Blobs*** ist beendet, wenn:
→ Er mit einer ***Wand*** kollidiert,
→ er mit dem ***Supercomputer*** kollidiert oder
→ er den Rand des ***Spielfeldes*** berührt.
Nach jedem Wurf müssen diese Fälle überprüft werden und anschließend muss entschieden werden, ob weitergespielt wird oder das Spiel zu Ende ist.

Diese Fälle werdet ihr in den folgenden Abschnitten einbauen.
Für die ersten beiden Fälle (**Wand**, **Supercomputer**) könnt ihr die Funktion ***wenn Blob.KolidiertMit*** benutzen:

AUFGABE

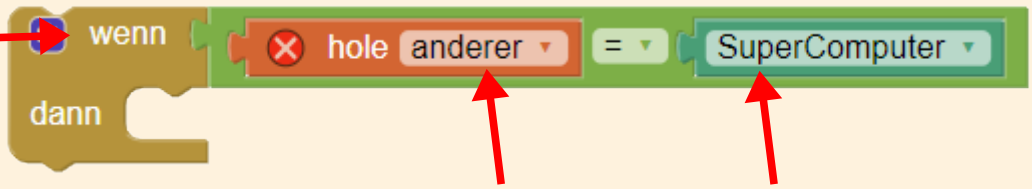
Wenn der ***Blob*** mit etwas kollidiert, dann kann etwas gemacht werden.



Zusätzlich erhaltet ihr die Komponente, mit der die Kollision stattgefunden hat. ***Anderer*** kann also eine ***Wand*** oder der ***Supercomputer*** sein.

AUFGABE

Im nächsten Schritt sollt ihr kontrollieren, welcher der drei Fälle eingetroffen ist. Ob der Supercomputer getroffen wurde, könnt ihr folgendermaßen überprüfen:



Wenn die getroffene Komponente (***anderer***) der ***Supercomputer*** ist, dann kann etwas gemacht werden.

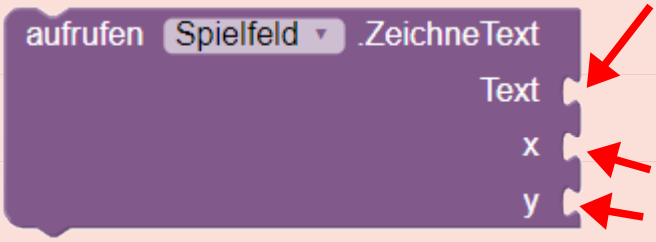
Erstellt das *wenn-dann*-Konstrukt für die Fälle: Supercomputer und die Wände.

TIPP

Die wenn-dann Struktur findet ihr unter Steuerung. Den hellgrünen Block (mit dem Gleichheitszeichen) findet ihr unter Mathematik.
Und den Supercomputer findet ihr ganz unten bei *SuperComputer.*
Falls ihr Hilfe braucht, zögert nicht zu fragen.

ACHTUNG
DER SUPERCOMPUTER WURDE GETROFFEN

Falls der Supercomputer getroffen wurde, müsst ihr dem Spieler sagen, dass er gewonnen hat.
Dazu könnt ihr einen Text mit folgender Konstruktion auf dem Bildschirm ausgeben:



Text, der ausgegeben werden soll

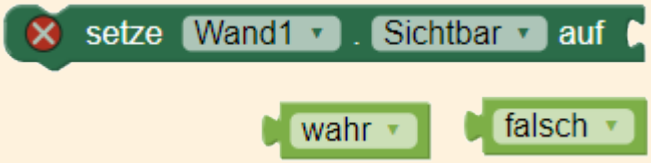
Der „Ort“ der Nachricht in X-Y-Koordinaten, die ausgegeben werden soll

ACHTUNG
EINE WAND WURDE GETROFFENEN

Falls eine Wand getroffen wurde, müsst ihr folgendes beachten:
Ihr müsst ***ein Leben*** abziehen. Dazu braucht ihr folgende Blöcke:

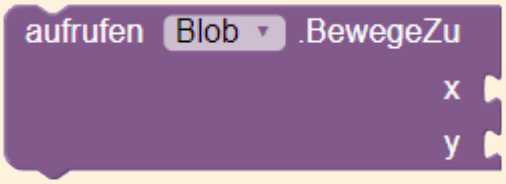
AUFGABE
MACHT EINE GETROFFENE WAND KAPUTT

Ihr müsst die getroffene Wand zerstören (unsichtbar machen):



AUFGABE
SETZT DEN BLOB WIEDER AUF SEINEN ANFANGSPUNKT

Ihr müsst den ***Blob*** an die ***Startposition*** zurücksetzen:



AUFGABE
HAT DER SPIELER VERLOREN

Und ihr müsst überprüfen, ob der Spieler ***verloren*** hat (Leben = 0).
Wenn das Leben gleich 0 ist, dann gebe einen Text aus,
der dem Spieler sagt, dass er verloren hat.

AUFGABE

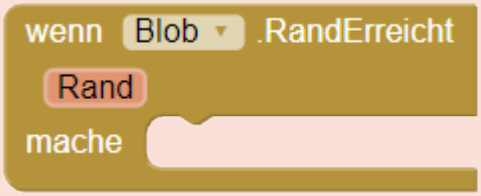
Falls ihr es noch nicht gemacht habt: Implementiert die Treffer für den Supercomputer und die Wände.

Testet eure App.

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

ACHTUNG
DER RAND WURDE GETROFFEN

Jetzt fehlt euch nur noch der Spielfeldrand. Eine Kollision mit diesem könnt ihr über folgende Funktion überprüfen:



Wenn der Rand getroffen wird, müsst ihr folgendes machen:
Den ***Blob*** auf die **Startposition** zurücksetzen.
Ein **Leben abziehen**.
Überprüfen, ob ihr **noch genug Leben** habt.

DER RESET-KNOPF

Als letztes müsst ihr dafür sorgen, dass das Spiel ***zurückgesetzt*** werden kann. Dazu braucht ihr den ***Reset-Knopf.***
Wenn der Reset-Knopf gedrückt wird, dann sollen die Grundeinstellungen wiederhergestellt werden.

DIE GRUNDEINSTELLUNGEN

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

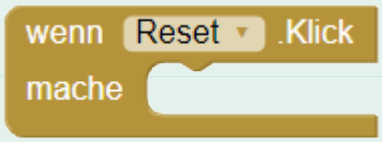
→ Die Lebensanzeige hat eine Grundeinstellung.

→ Die Position des ***Blobs*** muss auf **den Startwert zurückgesetzt** werden.

→ Zerstörte Komponenten (Wände u. Supercomputer) müssen alle **wieder sichtbar** gemacht werden.

Überlegt euch, was alles zurückgesetzt werden muss.

Setzt eure Ideen in der Funktion ***wenn ResetButton.Click*** um.



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 findet ihr einige Ideen um das Spiel noch zu erweitern:

Mehrere Wände

Ihr könnt mehrere Wände einbauen, die unterschiedliche Positionen haben.

Zusätzlich könnten die Leben erhöht werden, damit es nicht zu schwierig wird.

Bewegte Wände

Ihr könntet die Wände bewegen lassen. Dazu braucht ihr folgendes:

- Ihr müsst euch überlegen, in welchem Bereich sich die Wände bewegen sollen und wann sie ihre Richtung ändern.
- Beispielsweise könntet ihr die Wände immer bis zum Rand laufen lassen. Wenn dann eine Kollision mit dem Rand stattfindet, ändert ihr die Bewegungsrichtung um 180°.

Extra Leben

Ihr könntet einen zweiten ***Blob*** auf dem Spielfeld einfügen. Wenn man diesen trifft, bekommt man ein Extra-Leben geschenkt.

Quellenverzeichnis:

 *- Quelle: pixabay.com, Autor: OpenClipartVectors (CC0)*

 *- Quelle: pixabay.com, Autor: geralt (CC0)*

 *- Quelle: pixabay.com, Autor: OpenClipartVectors (CC0)*

 *- Quelle: openclipart.org, Autor: mi_brami (Unlimited Commercial Use)*

 *- Quelle: InfoSphere*

alle weiteren Grafiken sind Screenshots von App Inventor: http://appinventor.mit.edu/explore/