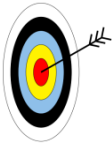


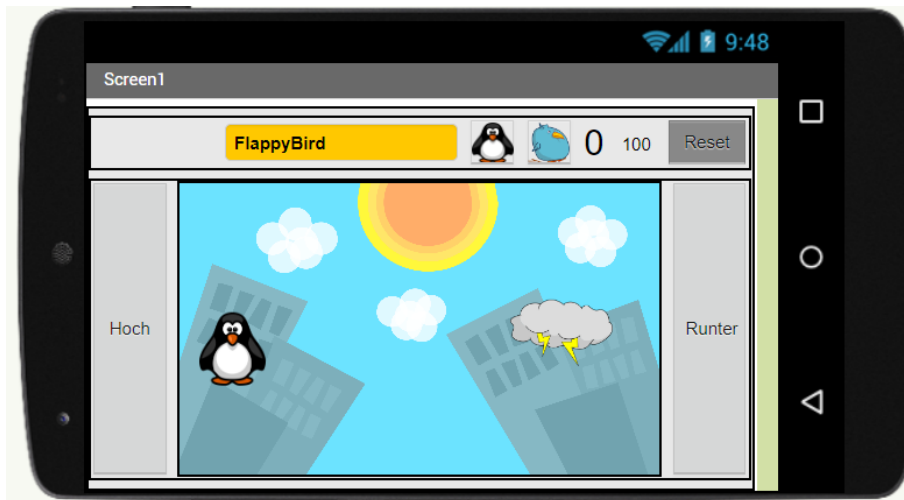
2c FlappyBird



Ihr habt euch für **FlappyBird** entschieden: ein lustiges Spiel, bei dem es darum geht, mit einem **Vogel** [1] **Gewitterwolken** [2] auszuweichen. Dieses Arbeitsblatt wird euch helfen, eine App zu erstellen, die...

- × einen **Vogel** auf einem **Spielfeld** [3] erscheinen lässt.
- × es dem Spieler bzw. der Spielerin ermöglicht, einen **Vogel** zu **steuern**.
- × **Wolken** erzeugt, die sich auf den Vogel zu **bewegen**.
- × **Punkte** hochzählt, wenn der Spieler bzw. die Spielerin es schafft, einer Wolke auszuweichen.




Schwierigkeitsgrad: ★★ ★



[4]



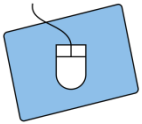
Während des gesamten Moduls geben euch die Arbeitsblätter Hinweise zur Umsetzung. Achtet dabei einfach auf die folgenden Symbole, die...

- × euer Arbeiten strukturieren und Teilziele aufzeigen, 
- × euch Hilfen geben, Wichtiges, Schwieriges, etc. kennzeichnen und 
- × die Arbeitsaufträge und Aktionen beinhalten. 

2c FlappyBird

Anlegen des Projekts

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



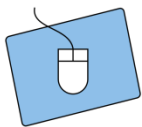
1. Erstellt euer Projekt, und gebt ihm einen passenden Namen.
2. Verbindet den App Inventor wie auf dem ersten Arbeitsblatt beschrieben mit dem Tablet/Smartphone.

Der Aufbau eurer App

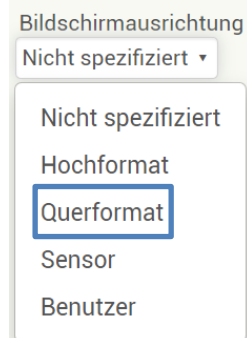
Als Nächstes müsst ihr euch Gedanken darüber machen, wie eure App aussehen soll. Eure App braucht Platz für folgende Dinge (die ihr der Reihe nach einbauen werdet):

- eine **Spielfläche**, auf der der Vogel sich bewegen kann,
- eine **Reset-Taste**, mit der ihr das Spiel neustartet,
- eine **Hoch-** und eine **Runter-Taste**, mit der man den Vogel nach oben und unten bewegen kann, und
- eine **Punkteanzeige**, die eure aktuellen Punkte hochzählt.

Das sind schon recht viele Dinge. Damit diese alle Platz haben, solltet ihr zunächst die **Bildschirmausrichtung** (Screen1) von **Nicht spezifiziert** in **Querformat** ändern.



1. Wählt unter den **Komponenten Screen1** aus.
2. Sucht unter den **Eigenschaften** den Eintrag **Bildschirmausrichtung**.
3. Ändert den Wert von **Nicht spezifiziert** in **Querformat**.



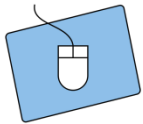
[5]

Die Bestandteile eurer App

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

2c FlappyBird

- Für die **Spielfläche** braucht ihr eine **Zeichenfläche**; diese findet ihr unter **Zeichnen und Animation**. Eine Zeichenfläche könnt ihr euch wie eine Leinwand vorstellen, auf der sich Objekte wie der Vogel bewegen können.
- Für die **Reset-Taste** braucht ihr eine **Taste** mit dem Text „Reset“.
- Für die **Bewegung** braucht ihr **zwei Tasten**. Eine beschriftet ihr mit „Hoch“, die andere mit „Runter“.
- Für die **Punkteanzeige** benötigt ihr eine **Bezeichnung** mit dem Text „0“.



1. Macht euch Gedanken darüber, wie ihr die Elemente positionieren wollt.
2. Zieht die **Anordnungen**, die ihr für euer Layout braucht, in den Betrachter hinein.
3. Fügt eurer App die oben beschriebenen Bestandteile (**Zeichenfläche, Tasten, Bezeichnung**) hinzu.

Der erste Test

Nachdem ihr die Bestandteile sinnvoll angeordnet habt, könnt ihr direkt testen, wie das Ganze auf dem Tablet/Smartphone aussieht.



1. Startet dazu die App auf dem Tablet/Smartphone, wenn ihr dies noch nicht gemacht habt.
2. Gefällt euch die Anordnung? Wenn ja, dann könnt ihr weitermachen. Ansonsten überarbeitet sie einfach nochmal.

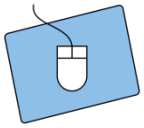


Lass eure App auf dem Tablet/Smartphone laufen. Der App Inventor aktualisiert alle Änderungen, die ihr vornehmt, und zeigt euch diese direkt an, ohne dass ihr die App neustarten müsst. Sollte dies einmal nicht der Fall sein, dann setzt die Verbindung zurück.

Umbenennen nicht vergessen 😊

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

2c FlappyBird



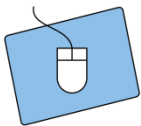
Gebt den Bestandteilen eurer App sinnvolle Namen.

Ein Hintergrund für eure Spielfläche

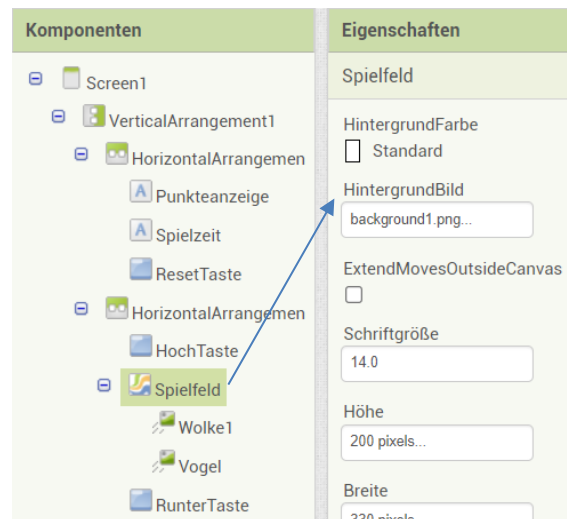
Damit der Vogel sich auf der Spielfläche wohlfühlt, soll diese einen Hintergrund bekommen. Ladet dazu einen der beiden Hintergründe aus dem folgenden Ordner hoch:

Desktop / InfoSphere goes Android / FlappyBird

Im Anschluss müsst ihr eurer Zeichenfläche den Hintergrund noch zuweisen:



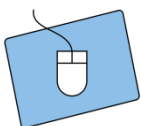
1. Wählt unter den **Komponenten** die **Zeichenfläche** aus.
2. Klickt unter den **Eigenschaften** auf **HintergrundBild**, und weist der Zeichenfläche einen der beiden Hintergründe zu.



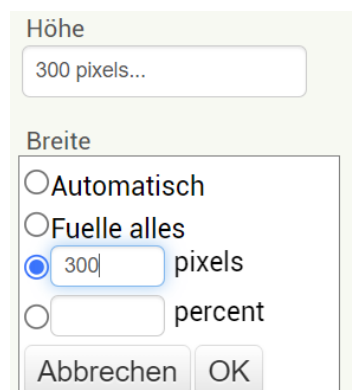
[6]



Wie ihr schon von Blatt 1 wisst, muss man die Größe beim App Inventor manchmal per Hand einstellen.



3. Ihr habt immer noch eure **Zeichenfläche** (die in der Abbildung schon in Spielfeld umbenannt ist) ausgewählt. Sucht in den **Eigenschaften** nach **Breite** und **Höhe**, und setzt beide auf 300 Pixel.
4. Testet jetzt eure App, und ändert gegebenenfalls die Anzahl der Pixel, damit es auf dem Tablet/Smartphone gut aussieht.



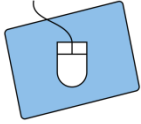
[7]

Auf der nächsten Seite geht es weiter mit dem Vogel.

2c FlappyBird

Der Vogel

Als Nächstes soll der Vogel auf das Spielfeld.

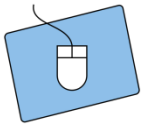


1. Wechselt in der Palette zu **Zeichnen und Animation**.
2. Zieht eine **ZeichenAnimation** direkt in den Hintergrund.
3. Ändert den Namen der ZeichenAnimation in Vogel.
4. Ladet nun aus dem Ordner, aus dem ihr auch den Hintergrund habt (S. 4), den Vogel hoch, und weist ihn der ZeichenAnimation zu.



Eine **ZeichenAnimation** ist ein besonderes Bild. Im Gegensatz zu einem normalen Bild kann die ZeichenAnimation sich auf dem Spielfeld bewegen.

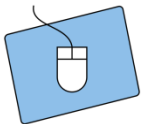
Außerdem hat euer Vogel eine **Position** auf dem Spielfeld. Diese könnt ihr in den Eigenschaften als **X- und Y-Koordinaten** ablesen bzw. verändern.



1. Wählt den Vogel unter den Komponenten aus.
2. Ändert unter den Eigenschaften die X- und Y-Koordinaten, und sucht euch eine schöne Startposition für den Vogel aus.

Eine (Gewitter-)Wolke

Als Nächstes braucht ihr eine Wolke auf der Spielfläche.



1. Verfährt genau wie bei dem Vogel, und erstellt die ZeichenAnimationen für die Wolke an einer sinnvollen Position.
2. Testet das Gesamtbild eurer App.

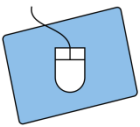
2c FlappyBird

Eine Uhr für die Wolke

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



Die Uhr findet ihr in der Palette unter **Sensoren**. Eine Uhr ist für viele Sachen gut, z. B. könnt ihr ein **Zeitgeberintervall** einstellen. Dieses löst nach Ablauf der eingestellten Zeit immer wieder die Funktion **Uhr.Zeitgeber** aus.



1. Zieht eine Uhr in eure App.
2. Wählt die Uhr unter den Komponenten aus, und ändert das Zeitgeberintervall in 500 (500 ms = entsprechen 0,5 Sekunden).

Ein Zwischenfazit

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

Betrachter

Zeige versteckte Komponenten im Viewer

Komponenten

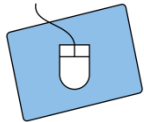
- Screen1
 - VerticalArrangement1
 - HorizontalArrangemen
 - Punkteanzeige
 - Spielzeit
 - ResetTaste
 - HorizontalArrangemen
 - HochTaste
 - Spielfeld
 - Wolke1
 - Vogel
 - RunterTaste
 - Uhr

[8]

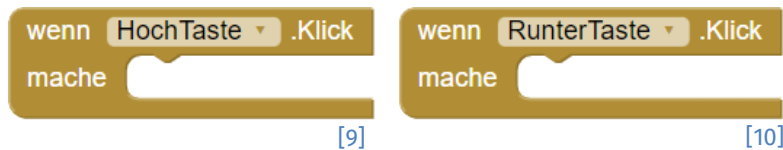
2c FlappyBird

Die Bewegung des Vogels

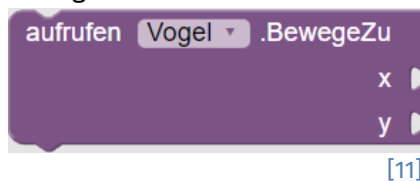
Der Vogel kann sich in dem Spiel hoch und runter bewegen. Daher müsst ihr diese beiden Bewegungen einbauen.



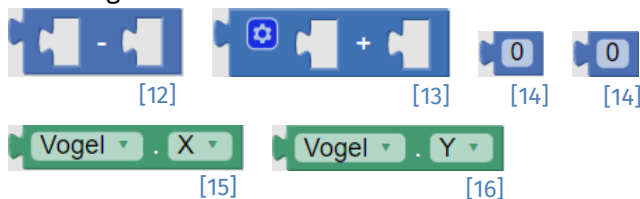
1. Wechselt in den Blöcke-Editor.
2. Wählt bei euren Hoch- und Runter-Tasten jeweils die **Wenn-XX.Klick-Funktion** aus:



3. Wenn ihr den Vogel hoch oder runter bewegt, dann ändert ihr dadurch die **Y-Koordinate** des Vogels. Der Ursprung (0, 0) des Koordinatensystems ist oben links in der Ecke der Spielfläche. Das heißt:
 - Wenn ihr den Vogel hoch bewegen wollt, muss die Y-Koordinate **kleiner** werden.
 - Wenn ihr den Vogel runter bewegen wollt, muss die Y-Koordinate **größer** werden.
4. Jetzt braucht ihr diese Funktion, mit der ihr den Vogel an eine neue Position (x, y) bewegen könnt:



Außerdem benötigt ihr noch die folgenden Blöcke, um den Vogel hoch und runter zu bewegen:

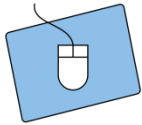


Versucht einmal selbstständig, mit diesen Blöcken die Bewegung nach oben und unten einzubauen. Zögert nicht, bei Fragen die Betreuer*innen anzusprechen. Vergesst nicht, immer wieder zu testen.

2c FlappyBird

Die Bewegung der Wolke

Für die Bewegung der Wolke braucht ihr die **Uhr** bzw. das **Zeitgeber.Intervall**. Jedes Mal, wenn dieses ausgelöst wird, soll sich die Wolke etwas nach links bewegen.



1. Ihr befindet euch immer noch im Blöcke-Editor. Wählt hier die Funktion **Wenn-Uhr.Zeitgeber** aus:



2. Ab jetzt ist der Vorgang identisch zur Bewegung des Vogels:

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

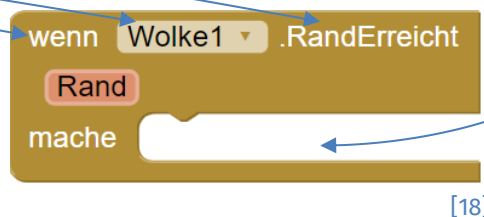
Versucht einmal selbstständig, die Bewegung der Wolke einzubauen. Auch hier gilt: Zögert nicht die Betreuer*innen zu fragen.

Tipp: Wenn die Wolke sich zu langsam bewegt, müsst ihr das **Zeitgeber.Intervall** im Designer-Editor verändern.

Neue Wolken erscheinen lassen

Eine neue Wolke soll dann erscheinen, wenn die aktuelle Wolke den linken Spielfeldrand erreicht hat. Dazu braucht ihr den folgenden Block:

Wenn die Wolke den Spielfeldrand erreicht, dann kann etwas **gemacht** werden.



Diesen Block könnt ihr schonmal in die Arbeitsfläche ziehen.

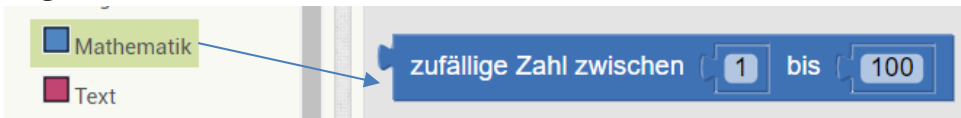
Um nun eine **neue Wolke** erscheinen zu lassen, werdet ihr einen kleinen Trick anwenden:

- Ihr wählt einfach eine neue zufällige Position für die schon vorhandene Wolke im rechten Teil des Bildschirms aus. Dadurch sieht es für den Spieler oder die Spielerin so aus, als wäre eine neue Wolke rechts erschienen und die alte Wolke verschwunden.
- Denkt daran, dass die Höhe der Wolke zufällig sein soll. Das bedeutet, dass die X-Koordinate immer

2c FlappyBird

dieselbe sein soll, aber die Y-Koordinate immer zufällig.

Die neue Position der Wolke soll also zufällig sein. Um das zu erreichen, besitzt der App Inventor die Möglichkeit, eine **Zufallszahl** zu generieren. Den entsprechenden Block findet ihr im Blöcke-Editor unter **Eingebaut → Mathematik**



[19]

Die Funktion wählt eine zufällige ganze Zahl im Bereich zwischen a (hier 1) und b (hier 100) aus.

Der Bereich der Zufallszahlen

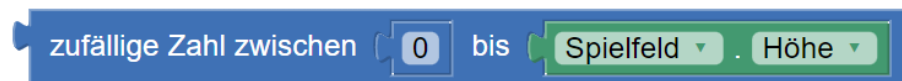
Euer Bereich für die Zahlen soll so groß sein wie die Höhe des Spielfelds. Diese erhaltet ihr folgendermaßen:



[20]

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

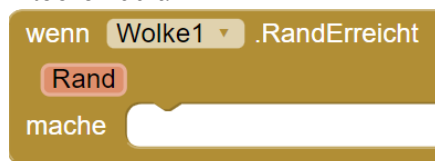
So sehen die Blöcke kombiniert aus:



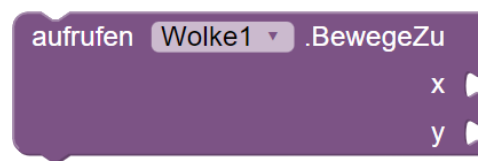
[21]



1. Erstellt nun die zufällige neue Position für die Wolke, indem ihr die folgenden Blöcke nutzt.



[18]



[22]



[21]

[14]

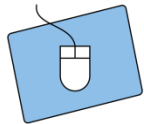
2. Testet eure App.

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

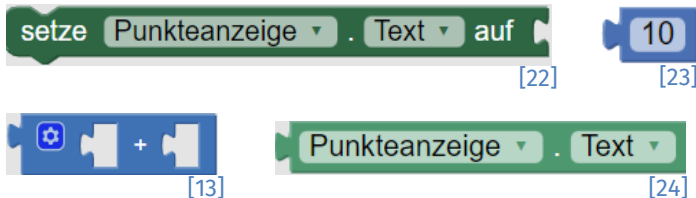
2c FlappyBird

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.



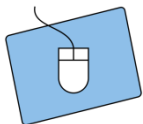
- Sucht die folgenden Bausteine heraus:



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

Die Kollision des Vogels mit der Wolke

Als Nächstes soll die Kollision mit der Wolke eingebaut werden. Diese ist gleichbedeutend mit dem Spielende. Die folgenden Schritte sind nötig:



- Sucht die Funktion **wenn-Vogel.KollidiertMit**. Diese wird bei der Kollision mit der Wolke ausgelöst.
- Schreibt in diese Funktion alles hinein, was bei der Kollision passieren soll, also:
 - Die Geschwindigkeit der Wolke wird auf 0 gesetzt.
 - Es wird ein Text ausgegeben. Dazu könnt ihr die Funktion **aufrufen-Spielfeld.ZeichneText** nutzen.

Die Reset-Taste

Als Letztes müsst ihr dafür sorgen, dass das Spiel zurückgesetzt werden kann. Dazu braucht ihr die Reset-Taste. Wenn die Reset-Taste gedrückt wird, dann sollen die Grundeinstellungen wiederhergestellt werden.



Die Grundeinstellungen

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

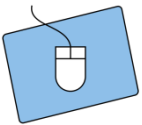
- Die Punkteanzeige hat eine Grundeinstellung.
- Die Position des Vogels kann auf den Startwert zurückgesetzt werden.

2c FlappyBird

- Die Wolke hat eine Startposition.
- Die Meldung, dass ein*e Spieler*in verloren hat, muss wieder von der Spielfläche gelöscht werden.



Mit `aufrufen Spielflaeche .Lösche` [26] kann ein Text auf der Spielfläche gelöscht werden.



1. Überlegt, was alles zurückgesetzt werden muss.
2. Setzt eure Überlegungen in der Funktion **wenn-ResetTaste.Klick** um.



*Gratulation 😊. Ihr habt ein voll funktionsfähiges FlappyBird-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 möchtet, dann wendet euch an die Betreuer*innen.*

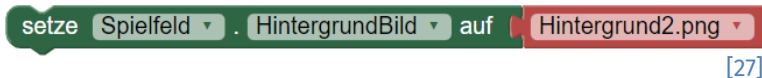
2c FlappyBird

Erweiterungen

Hier findet ihr einige Ideen, um das Spiel noch zu erweitern.

Mehrere Vögel und Hintergründe zur Auswahl

Um diesen Effekt zu erzielen, könnt ihr für jeden Hintergrund und jeden Vogel eine eigene Taste anlegen. Wenn ihr diese drückt, soll sich das jeweilige Bild ändern.

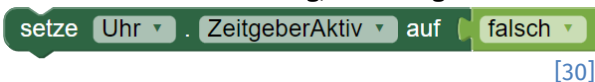


Ein Spielende

... wäre noch sehr schön. Dazu könnt ihr eine neue Bezeichnung hinzufügen, die bei 100 anfängt. Jedes Mal, wenn das ZeitgeberIntervall auslöst, reduziert ihr den Wert um 1. Wenn der Wert 0 erreicht wird, könnt ihr das Spiel beispielsweise resettet oder einen Text ausgeben, der sagt, wie viele Punkte der Spieler erreicht hat.



Außerdem wäre es wichtig, den Zeitgeber auszuschalten, damit die Wolke stehen bleibt.



Tipp: Wenn ihr die Reset-Taste drückt, muss der Zeitgeber wieder eingeschaltet werden und die Schrift vom Bildschirm entfernt werden. Schaut euch dazu die Funktionen der Spielfläche an, denn die wollt ihr löschen.

Quellenverzeichnis:

Abb. 1 – Quelle: ClipSafari, Autor: lemmling (<https://www.clipsafari.com/clips/o27563-cartoon-penguin>), CC0 1.0 (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.03.2023.

Abb. 2 – Quelle: ClipSafari, Autor: laobc (<https://www.clipsafari.com/clips/o61585-thunder-cloud>), CC0 1.0 (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.03.2023.

Abb. 3 – Quelle: ClipSafari, Autor: qubodup (<https://openclipart.org/detail/175652/cartoon-cities-background>), CC0 1.0 (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 06.03.2023.

Abb. 4 bis 30 – Quelle: Screenshot des MIT App Inventor 2 (<http://ai2.appinventor.mit.edu/?locale=de>), CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>), erstellt am: 06.03.2023.

– Quelle: InfoSphere, CC BY-SA 4.0 Attribution-ShareAlike 4.0 International (<https://creativecommons.org/licenses/by-sa/4.0/>).