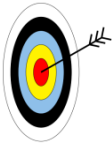




2d MoleMash

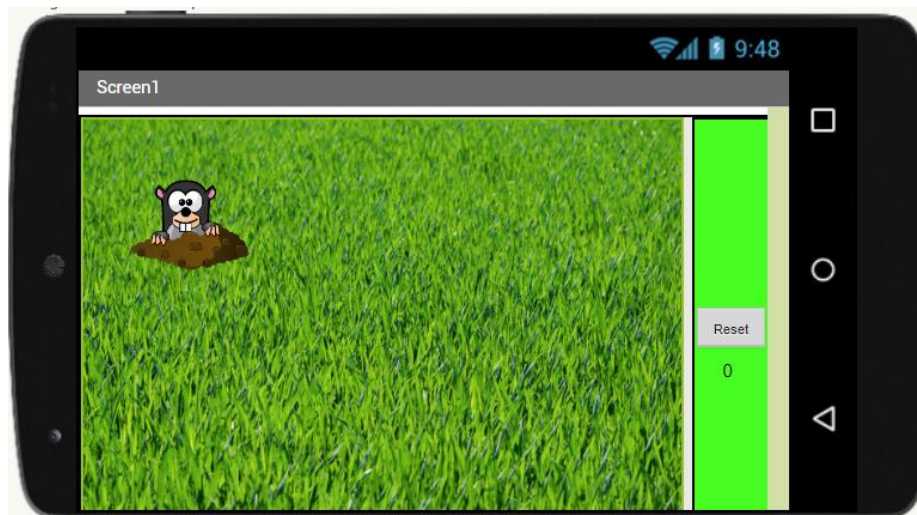


Ihr habt euch für **MoleMash** entschieden: ein lustiges Spiel, bei dem es darum geht, den

Maulwurf  [1] zu fangen, der an verschiedenen Positionen des Bildschirms erscheint. Dieses Arbeitsblatt wird euch helfen, eine App zu erstellen, die...

- × einen **Maulwurf** auf einer **Spielfläche**  [2] erscheinen lässt.
- × es dem Spieler bzw. der Spielerin ermöglicht, den Maulwurf durch Anklicken zu **fangen**.
- × **Punkte** hochzählt, wenn der Maulwurf gefangen wurde.




Schwierigkeitsgrad: ★★



[3]



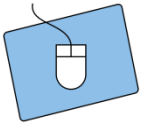
Während des gesamten Moduls geben euch die Arbeitsblätter Hinweise zur Umsetzung. Achtet dabei einfach auf die folgenden Symbole, die...

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

2d MoleMash

Anlegen des Projekts

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



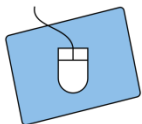
1. Erstellt euer Projekt, und gebt ihm einen passenden Namen.
2. Verbindet den App Inventor wie auf dem ersten Arbeitsblatt beschrieben mit dem Tablet/Smartphone.

Der Aufbau eurer App

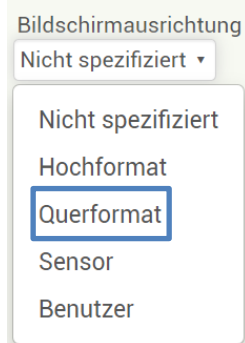
Als Nächstes müsst ihr euch Gedanken darüber machen, wie eure App aussehen soll. Eure App braucht Platz für folgende Dinge (die ihr der Reihe nach einbauen werdet):

- eine **Spielfläche**, auf der der Maulwurf sich bewegen kann,
- eine **Reset-Taste**, mit der ihr das Spiel neustartet 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**.



[4]

Die Bestandteile eurer App

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

- Für die **Spielfläche** braucht ihr eine **Zeichenfläche**; diese findet ihr unter **Zeichnen und Animation**.

2

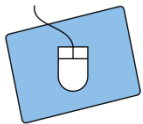
Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (<http://creativecommons.org/licenses/by-sa/4.0/>). Die Lizenz einzelner Materialien kann wie gekennzeichnet abweichen (siehe Quellenverzeichnis). Ausgenommen von dieser Lizenz sind außerdem alle Logos, insbesondere das RWTH- sowie das InfoSphere-Logo.



2d MoleMash

Eine Zeichenfläche könnt ihr euch wie eine Leinwand vorstellen, auf der sich Objekte wie der Maulwurf bewegen können.

- Für die **Reset-Taste** braucht ihr eine **Taste** mit dem Text „Reset“.
- 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**, **Taste**, **Bezeichnung**) hinzu.

Der erste Test

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



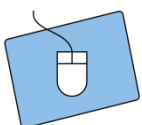
1. Startet dazu die App auf dem Tablet/Smartphone, wenn ihr dies noch nicht gemacht habt.
2. Gefällt euch die Anordnung? Wenn ja, dann könnt ihr weitermachen. Ansonsten überarbeitet sie einfach nochmal.



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.



Gebt den Bestandteilen eurer App sinnvolle Namen.

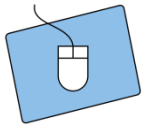
2d MoleMash

Ein Hintergrund für eure Spielfläche

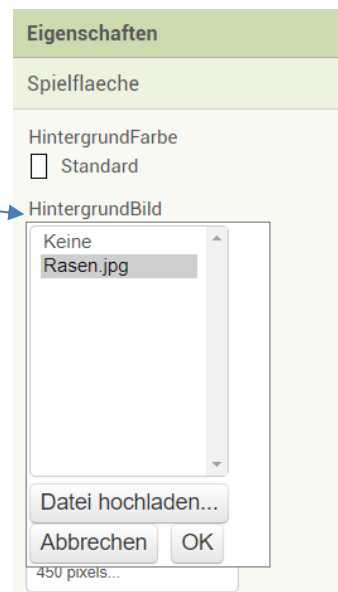
Damit der Maulwurf sich auf der Spielfläche wohlfühlt, soll diese einen Hintergrund bekommen. Ladet dazu einen der Hintergründe aus dem folgenden Ordner hoch:

Desktop / InfoSphere goes Android / MoleMash

Im Anschluss müsst ihr eurer Zeichenfläche den Hintergrund noch zuweisen:



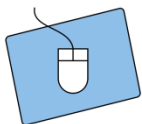
1. Wählt unter den **Komponenten** die **Zeichenfläche** aus.
2. Klickt unter den **Eigenschaften** auf **Hintergrundbild**, und weist der Zeichenfläche einen der Hintergründe zu.



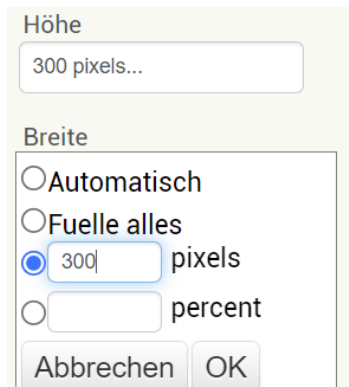
[5]



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 Spielflaeche 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.



[6]

Auf der nächsten Seite geht es weiter mit dem Maulwurf.

2d MoleMash

Der Maulwurf

Als Nächstes soll der Maulwurf auf die Wiese.

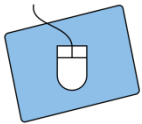


1. Wechselt in der Palette zu **Zeichnen und Animation**.
2. Zieht eine **ZeichenAnimation** direkt in den Hintergrund.
3. Ändert den Namen der ZeichenAnimation in Maulwurf.
4. Ladet nun aus dem Ordner, aus dem ihr auch den Hintergrund habt (S. 4), den Maulwurf hoch, und weist ihn der ZeichenAnimation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



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

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



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

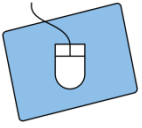
Eine Uhr für den Maulwurf

Als letzten Bestandteil eurer App braucht ihr eine Uhr. Diese braucht ihr, damit der Maulwurf sich bewegen kann.



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. Das ist hilfreich, wenn sich der Maulwurf nach einer gewissen Zeit an eine neue Position bewegen soll.

2d MoleMash



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

nicht sichtbare Komponenten

Timer

Komponenten

- Screen1
 - HorizontalArrangement1
 - Spieleflaeche
 - Maulwurf
 - VerticalArrangement1
 - ResetTaste
 - Anzeige_Punkte
- Timer

Umbenennen Löschen

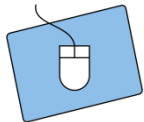
[7]

2d MoleMash

Die Bewegung des Maulwurfs

Als Erstes soll sich der Maulwurf zufällig über die Spielfläche bewegen. Dazu müssen zwei Dinge umgesetzt werden:

1. die Bewegung des Maulwurfs an eine zufällige Position.
2. das Zeitintervall, das vorgibt, wann diese Bewegung ausgelöst wird.



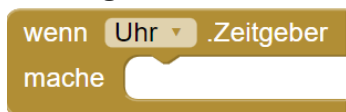
1. Wechselt in den Blöcke-Editor.
2. Wählt bei dem Maulwurf den Block **aufrufen-Maulwurf.BewegeZu** aus.



[8]

Dieser Block bewegt euren Maulwurf an neue X- und Y-Koordinaten, die ihr später an diesen Block anhängen werdet.

3. Um die Bewegung auszulösen, braucht ihr die Uhr. Benutzt deren Funktion **wenn-Uhr.Zeitgeber**, die alle 500 ms (also so, wie ihr es eingestellt habt) aufgerufen wird.

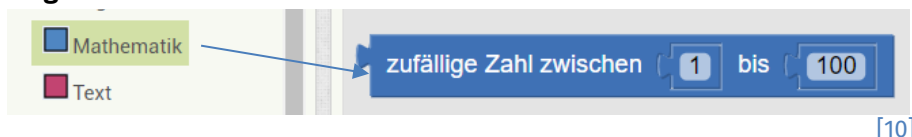


[9]

4. Kombiniert beide Blöcke.

Vorsicht: Nicht ausprobieren! Ihr habt noch offene Puzzle-Teile. Damit es richtig funktioniert, fehlen euch nämlich noch die **X- und Y-Koordinaten** der neuen Position. Diese fügt ihr im nächsten Schritt hinzu.

Die neue Position (x, y) des Maulwurfs soll 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**



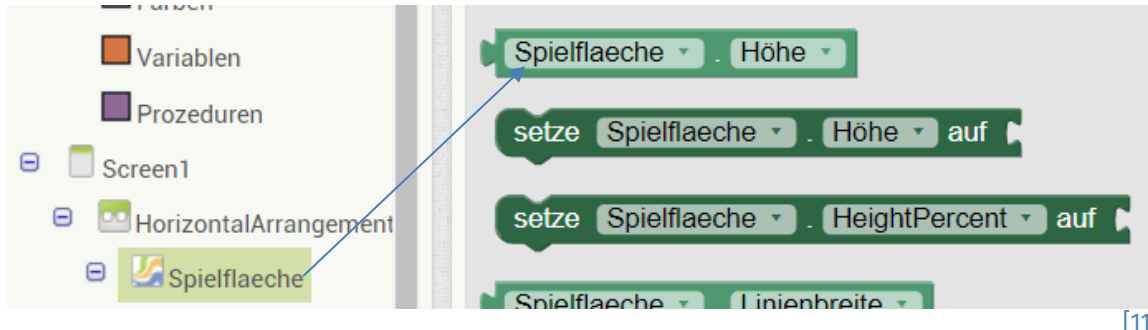
[10]

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

2d MoleMash

Der Bereich der Zufallszahlen

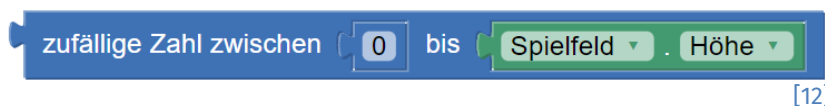
Euer Bereich für die Zahlen soll so groß sein wie die Wiese, also wie die Spielfläche. Die Größe der Wiese erhaltet ihr folgendermaßen:



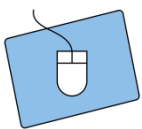
[11]

Spielflaeche.Höhe gibt euch die Höhe (Y-Koordinate) zurück, und **Spielflaeche.Breite** die Breite (X-Koordinate).

So sehen die Blöcke für die Höhe kombiniert aus:



[12]

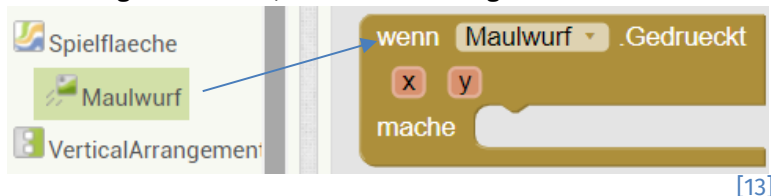


1. Erstellt nun die zufällige Bewegung für die Höhe.
2. Benutzt das Wissen aus den vorherigen Schritten, um die Funktion mit den fehlenden **X- und Y-Koordinaten** zu erweitern.
3. Testet die Bewegung!

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

Den Maulwurf fangen

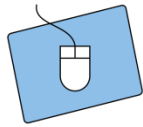
Um den Maulwurf zu fangen, muss man ihn antippen (bzw. drücken), bevor er sich wieder bewegt hat. Ob man das geschafft hat, lässt sich mit folgender Funktion feststellen:



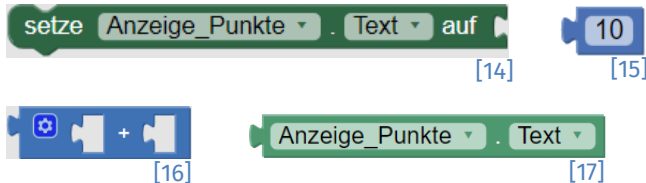
[13]

Wenn der Maulwurf gedrückt wurde, dann werden die Befehle ausgeführt, die in dem mache-Bereich stehen. Das Ziel ist, den Maulwurf zu treffen und dadurch beispielsweise 10 Punkte hochzuzählen.

2d MoleMash



1. Sucht dazu die folgenden Bausteine heraus:



2. Macht euch Gedanken, wie ihr sie sinnvoll kombinieren könnt.
3. Fügt sie in die Funktion **wenn-Maulwurf.Gedrueckt** ein, und testet euer Programm.

Die Reset-Taste

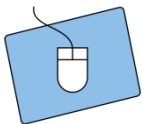
Als Letztes müsst ihr dafür sorgen, dass das Spiel zurückgesetzt werden kann. Dazu braucht ihr die Reset-Taste. Wenn die Reset-Taste gedrückt wird, dann sollen die Grundeinstellungen wiederhergestellt werden.



Die Grundeinstellungen

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

- Die Punkteanzeige hat eine Grundeinstellung.
- Die Position des Maulwurfs kann (muss aber nicht) auf den Startwert zurückgesetzt 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 MoleMash-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.*

2d MoleMash

Erweiterungen

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

Mehrere Maulwürfe 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.

```
setze Maulwurf . Bild auf Maulwurf.png [19]
```

```
setze Spielflaeche . HintergrundBild auf Rasen.jpg [20]
```

Ein zweiter Maulwurf auf der Spielfläche

Wie wäre es mit einem zweiten Maulwurf, der negative Punkte gibt, wenn man ihn anklickt? Dazu braucht ihr eine neue ZeichenAnimation. Außerdem müsst ihr anschließend den zweiten Maulwurf genau wie den ersten Maulwurf zufällig bewegen und bei einem Treffer 10 Punkte abziehen.

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 oder die Spielerin erreicht hat.

```
aufrufen Spielflaeche .ZeichneText
  Text
  x 100
  y 100
  verbinde " Du hast "
  verbinde Anzeige_Punkte . Text
  Punkte erreicht! [21]

wenn Spielzeit . Text = 0
dann [22]
```

Außerdem wäre es wichtig, den Zeitgeber auszuschalten, damit der Maulwurf stehen bleibt.

```
setze Timer . ZeitgeberAktiv auf falsch [23]
```

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

```
aufrufen Spielflaeche .Lösche [24]
```

2d MoleMash

Quellenverzeichnis:

Abb. 1 – Quelle: ClipSafari, Autor: Magnus (<https://www.clipsafari.com/clips/o168496-mole-with-buck-teeth>), CC0 1.0 (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 14.03.2023.

Abb. 2 – Quelle: pxhere (<https://pxhere.com/de/photo/984937>), CC0 1.0 (<https://creativecommons.org/publicdomain/zero/1.0/>), abgerufen am: 14.03.2023.

Abb. 3 bis 24 – 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/>).