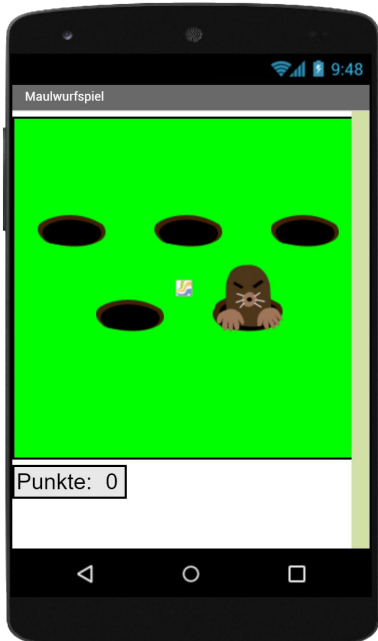




Non-visible components



Du brauchst diese Komponenten:

- 1 Canvas
- 5 ImageSprites ("Loch1" bis "Loch5")
- 1 ImageSprite ("Maulwurf")
- 1 HorizontalArrangement
- 2 Label ("PunkteText", "PunkteWert")
- 1 Clock
- 1 Sound

Hinweis

Hier findet ihr unsere Bilder:
appcamps.link/maulwurfspiel

Aufgabe

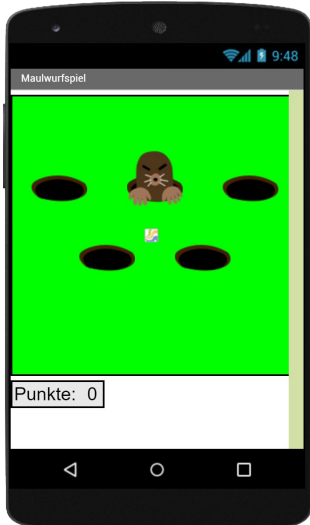
1. Gebt appinventor.mit.edu in den Browser ein und klickt oben links auf [Create App!](#).
2. Loggt euch ein und erstellt ein neues Projekt über "My Projects" → "Start new project".
3. Fügt alle benötigten Komponenten ein (siehe links).
4. Ladet die Bilder für die Löcher und den Maulwurf unter "Media" unten rechts hoch und ergänzt sie bei den ImageSprites als "Picture".
5. Setzt bei dem Maulwurf den Z-Wert auf 2.0, damit er immer vor den Löchern zu sehen ist.
6. Wechselt dann in die Programmierung.
7. Erstellt **zwei Variablen**:
 - "punkte": speichert die Anzahl der erwischten Maulwürfe
 - "loeche": enthält eine Liste alle Löcher, um ein zufälliges auszuwählen, aus dem der Maulwurf als nächstes schaut
8. Initialisiert die Variable *punkte* mit 0 und die Variable *loeche* mit einer Liste, die alle Löcher von 1 bis 5 enthält.

initialize global punkte to 0

initialize global loeche to make a list
Loch1
Loch2
Loch3
Loch4
Loch5

Nächste Aufgabe

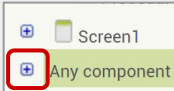
Erstellt eine Prozedur "bewegen". Diese soll dafür sorgen, dass der Maulwurf zu einem zufälligen Loch springt.



Hinweise

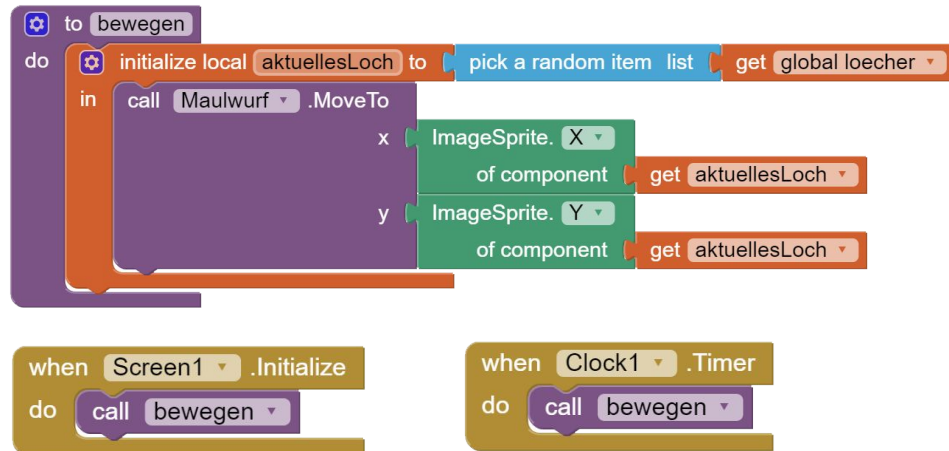
Eine lokale Variable kann nur von Anweisungen innerhalb eines bestimmten Rahmens genutzt werden. Dieser wird durch den orangenen Block markiert. Wird dieser Rahmen verlassen, existiert auch die Variable nicht mehr.

Für die grünen Blöcke klickt ihr links auf das Plus bei "Any component". Ihr findet die Blöcke dann unter "Any ImageSprite".



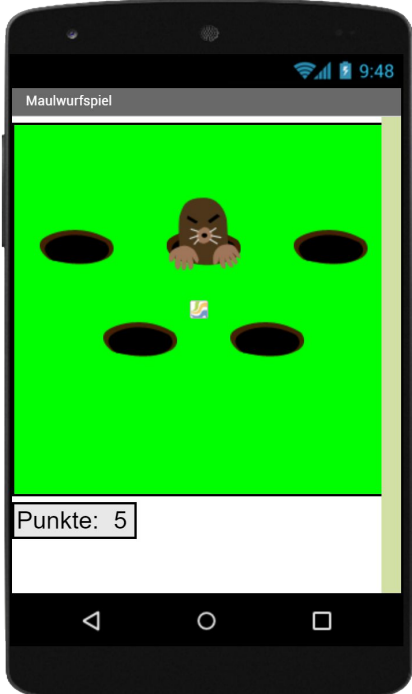
Aufgabe

1. Erstellt eine neue **Prozedur** und nennt sie "*bewegen*".
2. Initialisiert zu Beginn eine lokale Variable "aktuellesLoch" mit einem zufälligen Eintrag aus der Liste "*loecher*".
3. Bewegt den Maulwurf dann zur Position des ausgewählten Lochs. Nutzt die lokale Variable "*aktuellesLoch*", um die X- und Y-Koordinaten herauszufinden.
4. Ruft die Prozedur auf, sobald der Screen initialisiert wurde.
5. Ruft die Prozedur auch auf, sobald die Uhr abgelaufen ist. Die Zeit dafür könnt ihr in den Properties der Clock1 in Millisekunden einstellen ("TimerInterval"). Je kleiner die Zahl, desto schwieriger das Spiel. Standardmäßig eingestellt ist 1s also 1000ms. Probiert das am besten aus.



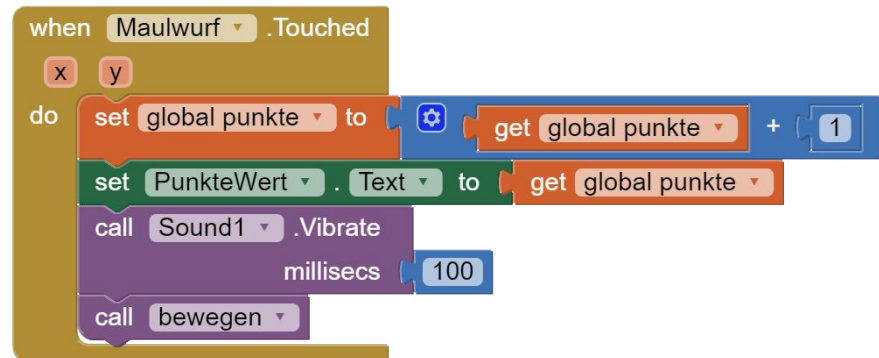
Nächste Aufgabe

Wenn der Maulwurf berührt wird, soll es einen Punkt geben und das Smartphone soll vibrieren.
Tipp: Schaut euch an, welche Komponente bisher noch nicht genutzt wird.



Aufgabe

1. Sobald der Maulwurf berührt wird, bekommt der Spieler einen Punkt. Erhöht also die Variable „punkte“ um 1.
2. Aktualisiert auch das Label mit dem Wert der Punkte.
3. Sorgt über die Sound-Komponente dafür, dass das Smartphone für 100 ms vibriert, wenn der Maulwurf getroffen wurde.
4. Ruft am Ende wieder die Prozedur „bewegen“ auf, damit der Maulwurf sich ein neues Loch sucht.



Nächste Aufgabe

Nehmt die Zeit anhand einer weiteren Clock. Ergänzt dafür auch zwei Labels wie bei den Punkten, um sie anzuzeigen.



Non-visible components
MaulwurfUhr Sound1 SpielUhr

Du brauchst diese Komponenten:

2 Label ("ZeitText", "ZeitWert")

1 Clock ("SpielUhr")

Aufgabe

1. Fügt eine weitere Clock "SpielUhr" ein und benennt die Clock1 um in "MaulwurfUhr". Fügt außerdem zwei weitere Labels in das HorizontalArrangement ein und benennt sie wie links.
2. Setzt das TimerInterval der neuen Uhr auf 1000ms, also 1s.
3. Wechselt in die Programmierung und erstellt eine neue Variable "zeit". Zu Beginn soll diese den Wert 0 haben.
4. Sobald die SpielUhr abgelaufen ist, also einmal die Sekunde, erhöht die "zeit" um 1.
5. Zeigt den Wert der Variablen danach im Label "ZeitWert" an.
6. Setzt zu Beginn des Spiels die Zeit und die Punkte zurück auf null und zeigt diese jeweils an. Erweitert dafür den Block "Screen1.Initialize" von Lernkarte 7.2.

initialize global zeit to 0

when SpielUhr.Timer
do
set global zeit to get global zeit + 1
set ZeitWert.Text to get global zeit

when Screen1.Initialize
do
set global punkte to 0
set PunkteWert.Text to get global punkte
set global zeit to 0
set ZeitWert.Text to get global zeit
call bewegen

Schon fertig?

Super! Ihr habt schon alles erledigt. Ihr könnt eure App natürlich noch erweitern. Zum Beispiel so:

- Baut einen Reset-Knopf ein, mit dem zwischendurch die Punkte *zurückgesetzt* werden.
- Ergänzt ein *Zeitlimit*, nachdem das Spiel beendet wird.
- Sorgt dafür, dass ein Punkt *abgezogen* wird, wenn der Nutzer daneben klickt.
- Lasst den Nutzer die *Schwierigkeit* selbst einstellen über einen Slider.

Tipp: Verändert damit das TimerInterval der MaulwurfUhr.

Testet eure App, indem ihr es auf euer Smartphone übertragt. Wie viele Punkte schafft ihr?