**Infoblatt**

Über ca. 3 Stunden beschäftigt sich dieses Modul mit dem Programmieren in der graphischen Programmiersprache *Scratch*. Anders als im Programmier-Modul für Klasse 6 gibt es diesmal keine Variante für die Programmiersprache *Pocket Code*.

Da in Klasse 7 für alle Schülerinnen und Schüler (SuS) einfache Grundlagen der Informatik als einstündiges Fach auf dem Stundenplan stehen, wird an sehr vielen Schulen mit Scratch gearbeitet. Daher bietet es sich an, Scratch auch in Mkid zu nutzen.

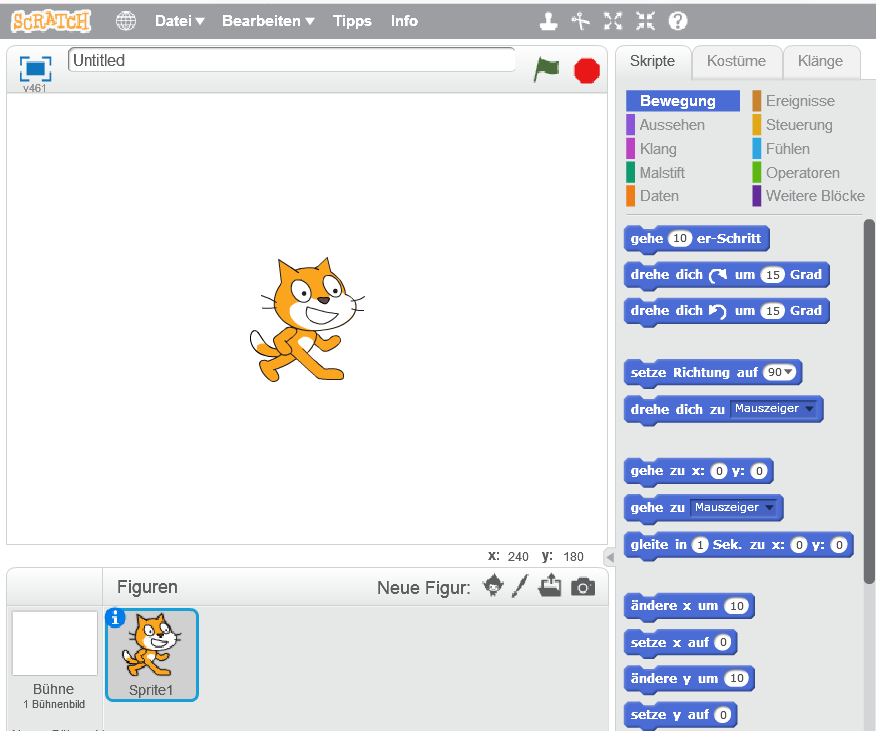
Die meisten Schulen werden Scratch auf ihren Servern bzw. Rechnern installiert haben, sodass Sie nur einen Computerraum oder einen Klassensatz Notebooks oder Tablets für die Mkid-Stunden buchen müssen. Klären Sie frühzeitig, was Sie benötigen und beachten müssen, um mit den SuS im Computerraum oder mit dem Notebooksatz arbeiten zu können. Sie müssen in den Stunden allen SuS zwei Dateien mit Spielfeldern zur Verfügung stellen. Fragen und testen Sie, wie das an Ihrer Schule am einfachsten geht (Tauschverzeichnis, Funktion Austeilen der Klassenraum-Software, …). Informieren Sie sich auch, wie und wo die SuS ihre Scratch-Programme speichern können.

Im letzten Schuljahr wurde in Mkid bereits ein Rechentrainer programmiert. Dieses Modul greift das Thema Programmieren wieder auf, ist aber so gestaltet, dass man ohne Vorkenntnisse aus Mkid der Klasse 6 oder dem Informatikunterricht der Klasse 7 zurechtkommt.

Bevor Sie mit Ihren SuS mit Scratch arbeiten, sollten Sie sich damit gründlich vertraut machen und alle Aufgaben aus den Arbeitsblättern selbst programmieren. Testen Sie auch die Verwendung von Scratch in Ihrer Schule (inkl. der Auswahl an Kostümen für die Figuren), da dort möglicherweise andere Pakete installiert sein könnten als auf Ihrem Computer.

**Hinweise zu Scratch:**

Scratch kann online genutzt werden <https://scratch.mit.edu/projects/editor/?tip_bar=home> oder kostenlos auf Geräten mit den Betriebssystemen Windows, macOS und Android installieren werden. Den dazu benötigten Offline-Editor finden Sie auf der Scratch Homepage <https://scratch.mit.edu> unter Support. Dort stehen neben der aktuellen auch ältere Versionen von Scratch zum Download zur Verfügung. An vielen Schulen sind noch ältere Versionen in Gebrauch, weswegen es sich vor dem Download empfiehlt zu klären, welche Version an der Schule installiert ist. Der Online-Editor nutzt ausschließlich die aktuelle Scratch-Version.

Um mit Scratch zu programmieren, werden einfach die gewünschten Programmblöcke in den Skriptbereich gezogen. Die Programmblöcke sind in verschiedene Bereiche wie Bewegung, Aussehen, Steuerung, Daten, Fühlen etc. gruppiert. Die Zuordnung ist nicht in allen Fällen intuitiv. Machen Sie sich daher mit den verschiedenen Bereichen vertraut.

Um einen Programmblock zu entfernen, schiebt man ihn einfach wieder zurück in den Block-Vorrat.

In Scratch können allen Figuren und Bühnen (so heißen die Hintergründe) Programm-Skripte zugeordnet werden. Damit man die Skripte, die z.B. zur Figur Katze gehören, sehen und verändern kann, muss man vorher im Bereich Figuren auf die Katze klicken. Sind die gesuchten Skripte nicht zu sehen, ist wahrscheinlich gerade eine andere Figur oder eine Bühne aktiviert.

**Tipps für den Unterricht:**

Besprechen und üben Sie bei erster Gelegenheit das Speichern eigener Scratch-Projekte an einer Stelle, an der die SuS das Programm auch Wochen später wiederfinden.

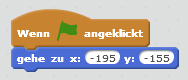
Das vorbereitet Spielfeld für das Labyrinth finden Sie in der Datei *Spielfeld1*, ein weiteres für Fortgeschrittene in der Datei *Spielfeld2*. Teilen Sie zunächst nur Spielfeld1 an die SuS aus.

Sie können die Lösungen zu den einzelnen Aufgaben an der Tafel aushängen, sodass die SuS bei kleinen Problemen oder Fragen selbstständig nachschauen können. Das kann Sie entlasten, weil so weniger Fragen auf einmal von Ihnen beantwortet werden müssen. Es ist sinnvoll vorzugeben, dass die Lösungen von den SuS nicht mit an den eigenen Platz genommen werden dürfen.

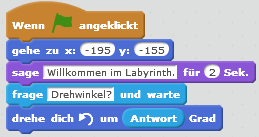
Wie schnell die SuS vorankommen, hängt auch von den Vorkenntnissen ab und wird variieren. Daher sind am Ende der Einheit zusätzliche Aufgaben für die schnelleren SuS vorgesehen.

**Lösungen zu den Aufgaben:**

**Aufgabe 1**

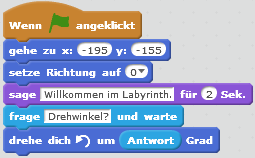
**Aufgabe 2**

Ungefähre Werte für die Koordinaten.

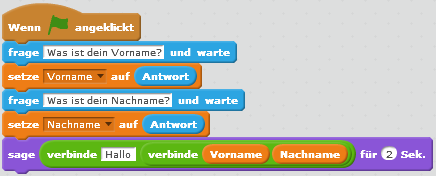
**Aufgabe 3**

´

Soll die Drehrichtung geändert werden (Drehung nach rechts), so kann man ein negatives Vorzeichen eingeben, z.B. „-90“. Das ist beim Spielen einfacher als die Alternative „270“.

**Aufgabe 4**

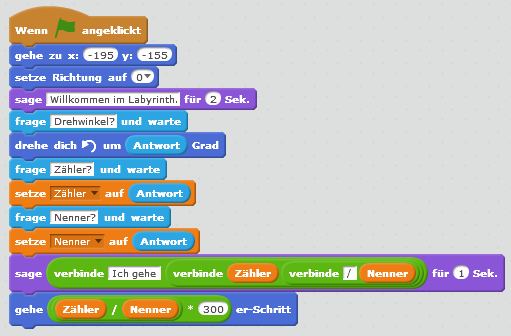
**Aufgabe 5**

c) d)

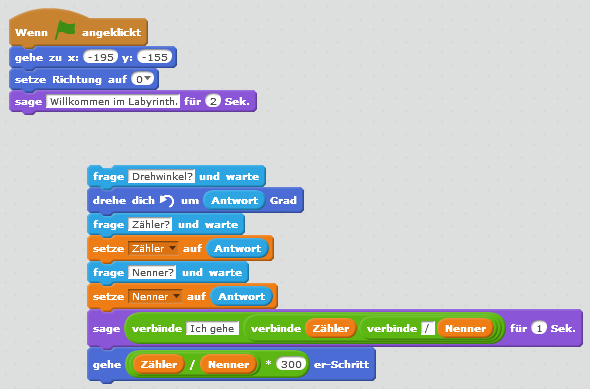
**Aufgabe 6**

a) b)

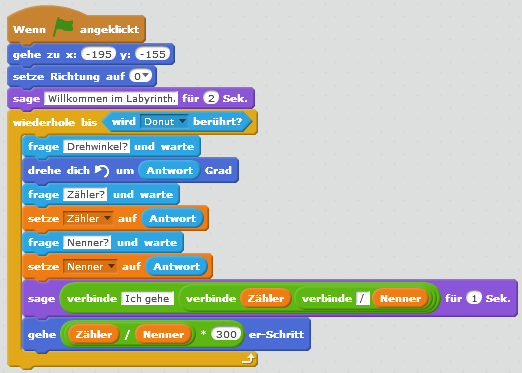
**Aufgabe 7**



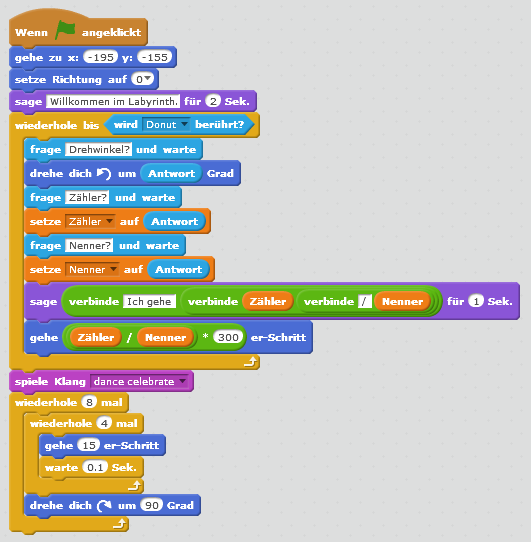
**Aufgabe 8**



**Aufgabe 10**

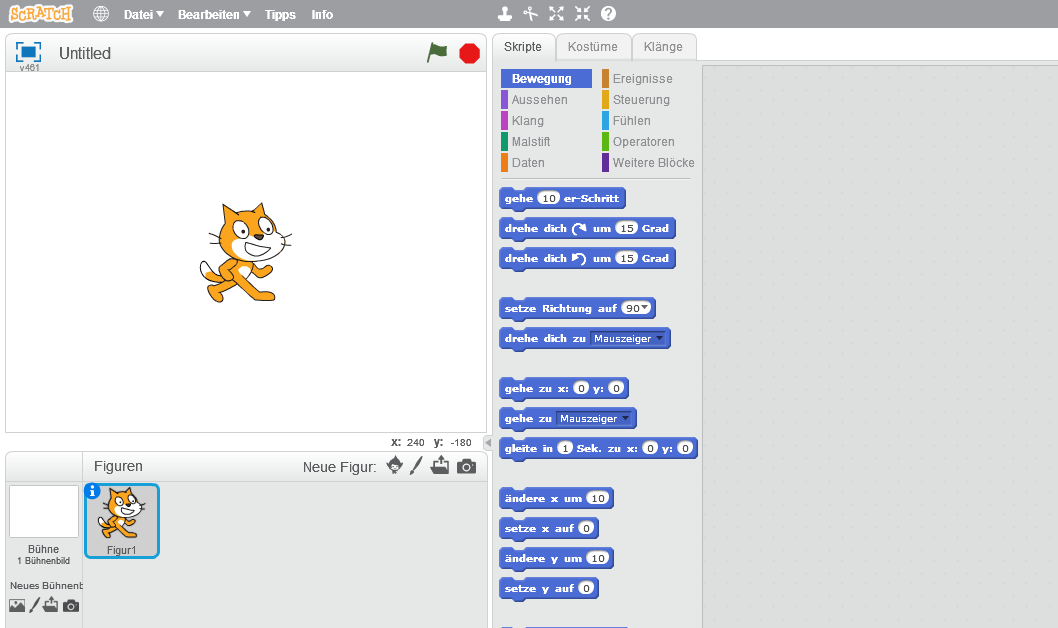


**Aufgabe 11**



**Informationen zu Scratch 2.0 und Vorbereitungen**

**Mit Scratch 2.0 programmieren**

Der Bildschirm ist bei Scratch in drei Teile aufgeteilt. Rechts befindet sich der **Skript- Bereich**, in dem die Programme (auch Skripte genannt) erstellt werden.

Skript-Bereich

Spielfeld

Links ist oben das **Spielfeld**, in dem später die Programme ablaufen. Du kannst die Programme mit der grünen Fahne starten und mit dem roten Stopp-Zeichen anhalten.

Links unten ist der **Figuren-Bereich.** Dort kannst du beim Programmieren verschie­denen Figuren und Bühnen auswählen.

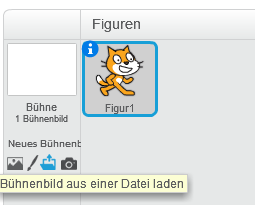
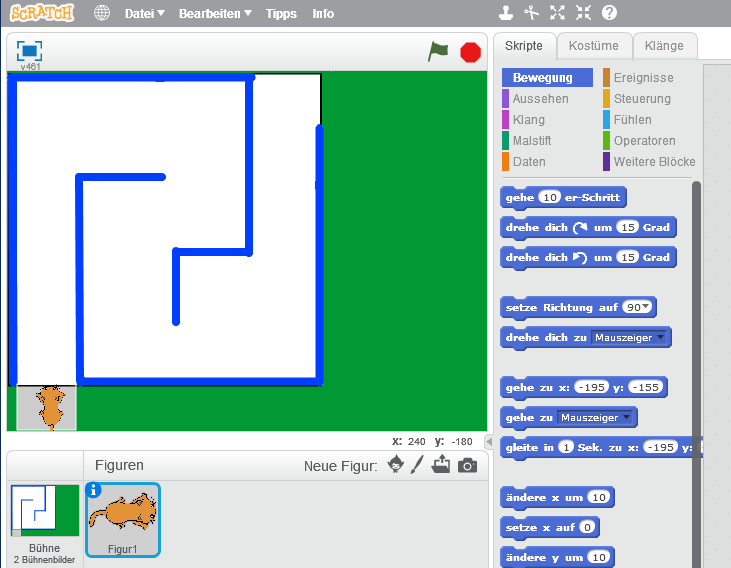
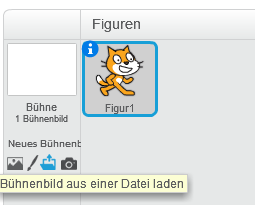
Figuren-Bereich

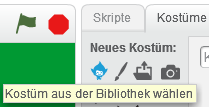
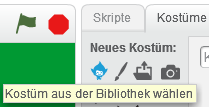
Um ein Programm zu schreiben, zieht man die farbigen Blöcke aus dem Vorrat nach rechts in den Skript-Bereich.

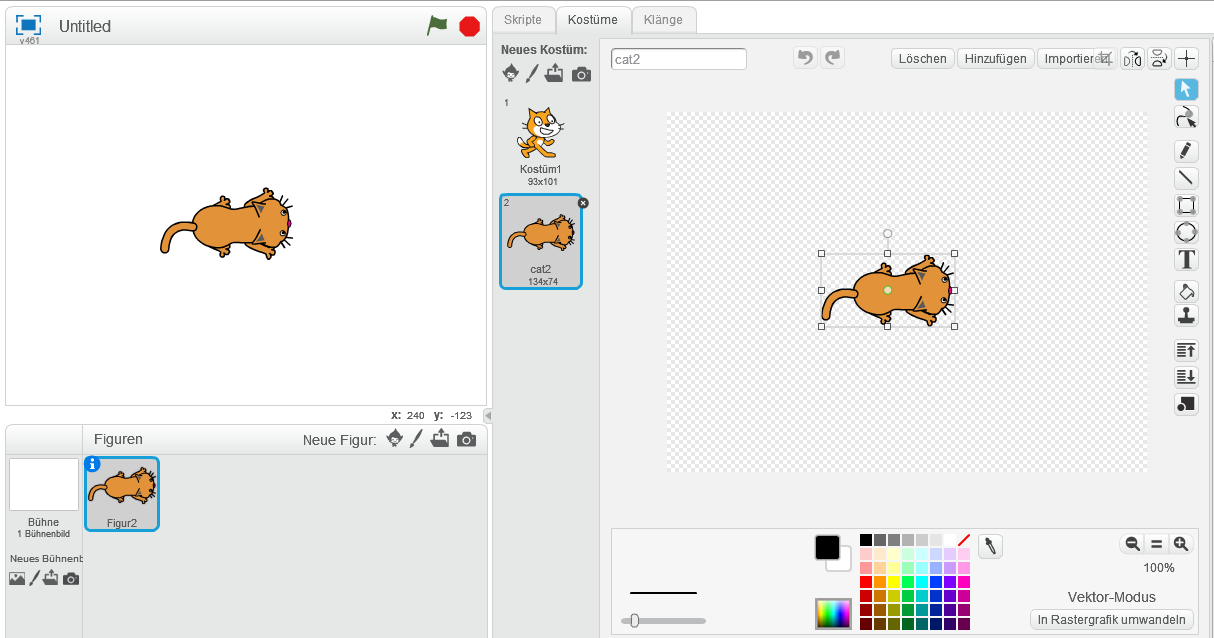
Probiere das kurz aus: Benutze aus dem Bereich  den „Wenn grüne Fahne ange­klickt“-Block und aus dem Bereich  den „gehe 10-er Schritt“-Block, um ein Programm zu schreiben, bei dem die Katze ein Stück nach rechts geht. Du musst dabei die Blöcke aneinanderhängen – wie bei Puzzle­teilen. Um das Programm zu starten, klicke auf die grüne Fahne  oberhalb des Spielfelds.

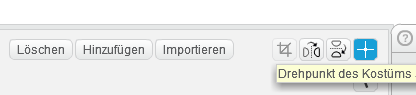
Um nicht mehr benötigte Blöcke zu löschen, kannst du sie einfach wieder in den Vorrat schieben.

**Vorbereitungen für das Labyrinth-Programm:**

a) Für das Labyrinth gibt es bereits einen passenden Hintergrund (in Scratch „Bühne“ genannt). Er befindet sich in der Datei **Spielfeld\_1**. Klicke im Figuren-Bereich unten links unter Neues Bühnenbild auf  . Du erfährst von deiner Lehrerin bzw. deinem Lehrer, wie du die Datei findest. Wähle die Datei aus und klicke auf Öffnen. Nun solltest du im Spielfeld-Bereich oben links das gleiche Labyrinth wie im folgenden Bild sehen.

b) Klicke nun im Figuren-Bereich auf die „Figur 1“ und dann auf „Kostüme“ (oben im Scratch-Bildschirm). Wähle unter „Neues Kostüm“ das linke Symbol  . Es öffnet sich ein Fenster, in dem du mit einem Doppelklick das Kostüm „cat2“ auswählst, das die Katze von oben zeigt.

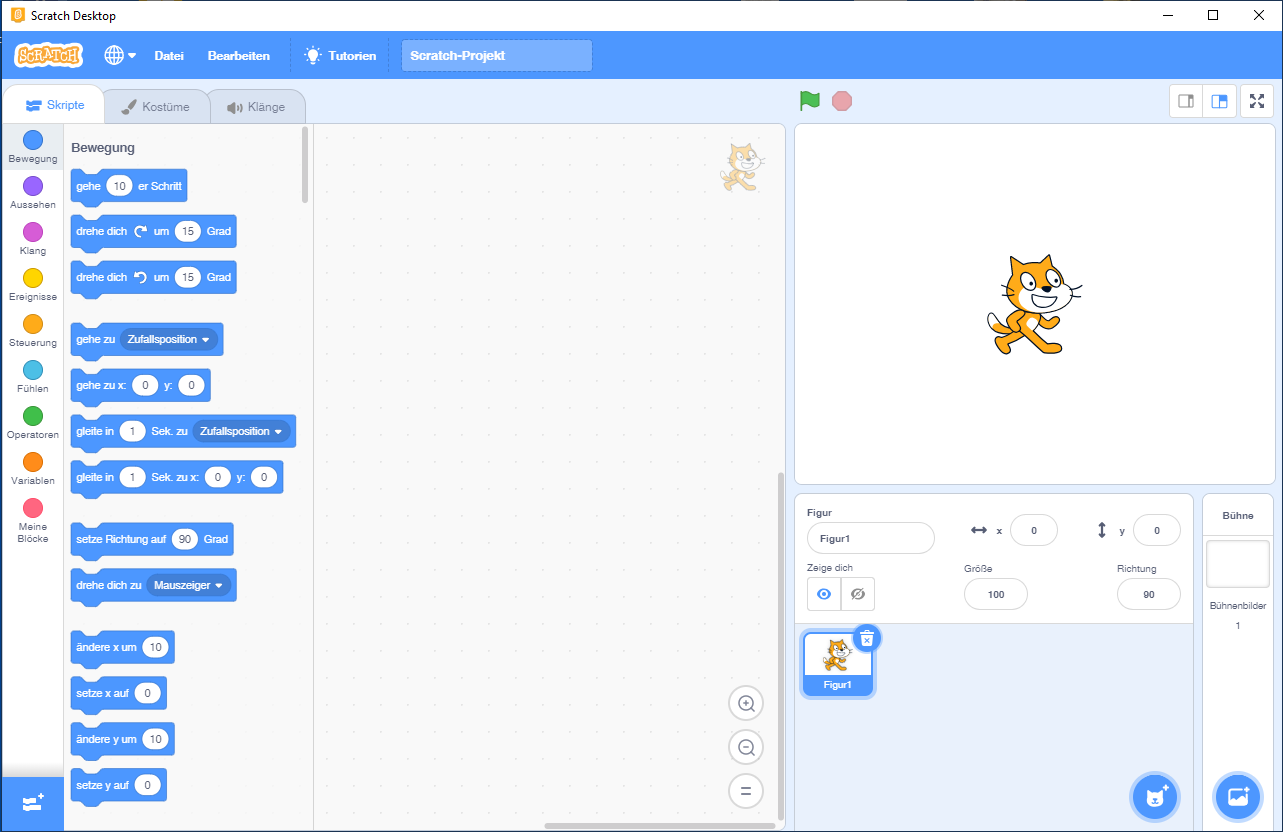
c) Die Katze ist für unser Labyrinth zu groß. Klicke zuerst ganz rechts unten auf „in Vektorgrafik umwandeln“.   
Ziehe mit der Computer-Maus ein Fenster um Katze im rechten Teil des Scratch-Bildschirms. Achte darauf, dass die ganze Katze in diesem Fenster enthalten ist.  
Klicke dann eine Ecke des Fensters an und verkleinre die Katze auf ungefähr die Hälfte.  
Dabei hat sich die Katze wahrscheinlich ein wenig verschoben. Schiebe die Katze daher wieder in die Mitte des Bereichs mit dem karierten Hintergrund.

d) Teste nun, ob die Katze klein genug ist. Klicke im Spielfeld (linker Teil des Scratch-Bildschirms) auf die Katze und schiebe sie in den grauen Startbereich. Die Katze muss etwas kleiner sein als der Startbereich. Falls nicht, verkleinere die Katze im rechten Teil des Scratch-Bildschirms noch ein wenig mehr.

e) Stelle nun den Drehpunkt ein, um den sich die Katzen-Figur später drehen wird. Klicke dazu oben rechts auf das Drehpunkt-Symbol.  
Anschließend schiebst du das schwarze Kreuz möglichst genau auf die Mitte der Katze.

Jetzt ist alles vorbereitet, um später das Labyrinth-Spiel zu programmieren. Gehe nun auf „Datei“ und dann auf „Speichern unter“ und **speichere** alles unter dem Namen *Labyrinth\_1*.  
Halte dich dabei an die Hinweise deiner Lehrerin bzw. deines Lehrers, in welchem Ordner du deine Programme speichern sollst.

**Informationen zu Scratch 3.0 und Vorbereitungen**

**Mit Scratch 3.0 programmieren**

Der Bildschirm ist bei Scratch 3.0 in vier Teile aufgeteilt. Links befindet sich der **Skript- Bereich**, in dem die Programme (auch Skripte genannt) erstellt werden.

Spielfeld

Rechts ist oben das **Spielfeld**, in dem später die Programme ablaufen. Du kannst die Programme mit der grünen Fahne starten und mit dem roten Stopp-Symbol anhalten.

Figuren-Bereich

Skript-Bereich

Rechts unten ist der **Figuren-Bereich.** Dort kannst du beim Programmieren verschie­denen Figuren und Bühnen auswählen. Ganz unten rechts findest du neben dem Figuren-Bereich den **Bühnen-Bereich**.

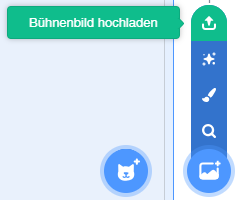
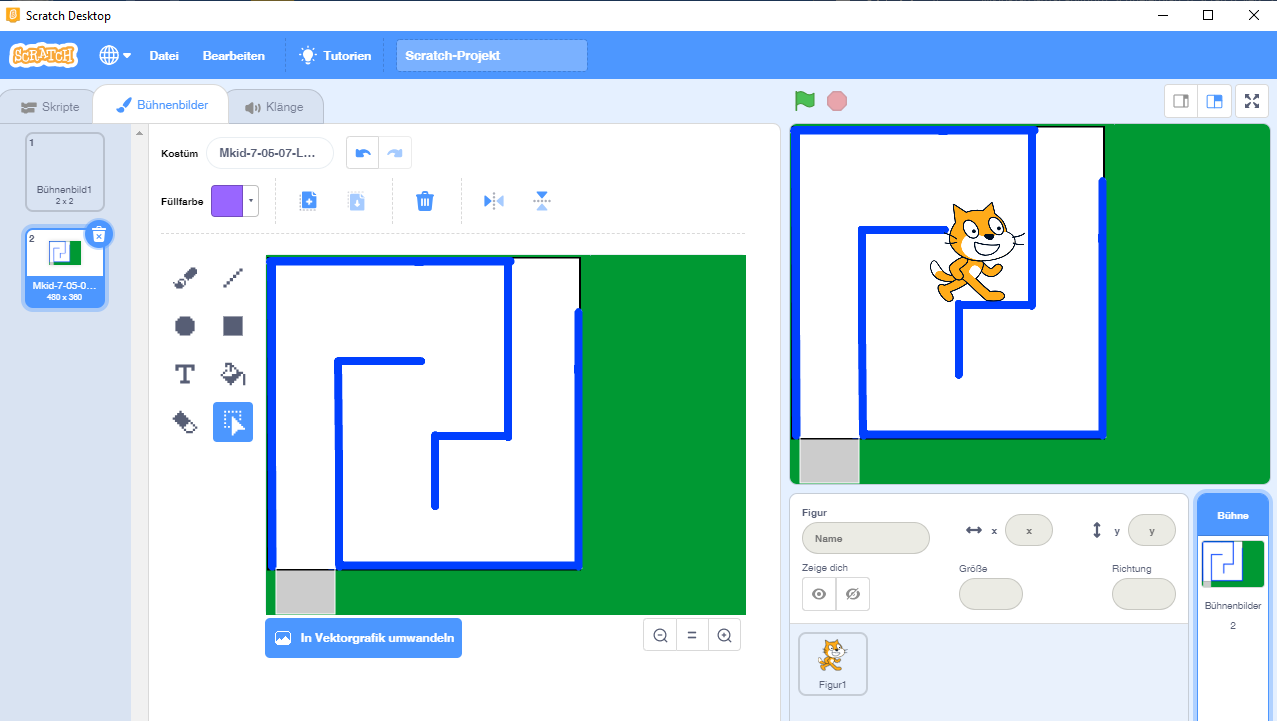
  
Um ein Programm zu schreiben, zieht man die farbigen Blöcke aus dem Vorrat nach rechts in den Skript-Bereich.

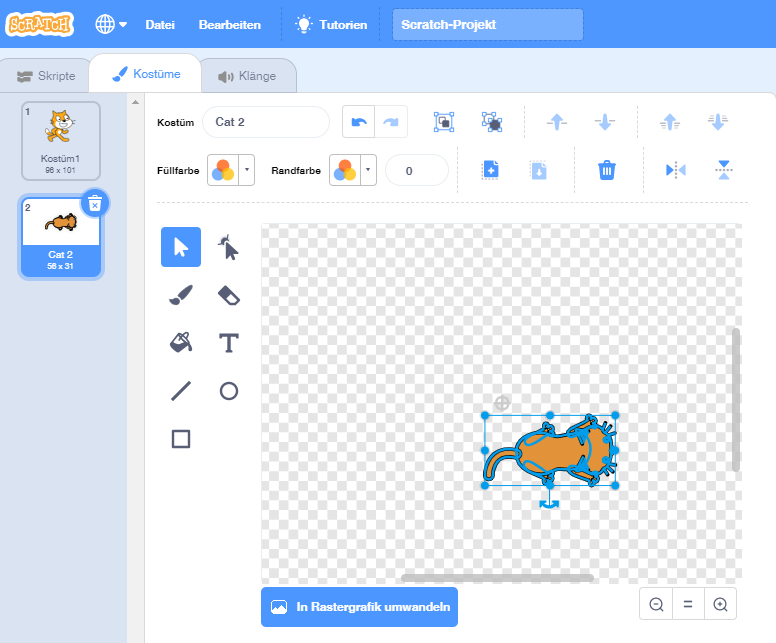
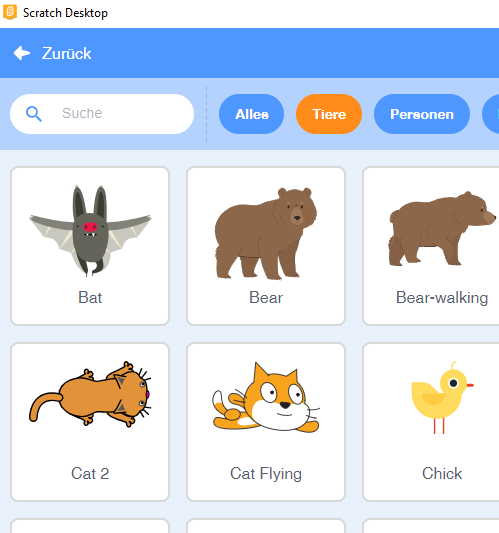
Probiere das kurz aus: Benutze aus dem Bereich den „Wenn grüne Fahne ange­klickt“-Block und aus dem Bereich den „gehe 10-er Schritt“-Block, um ein Programm zu schreiben, bei dem die

Katze ein Stück nach rechts geht. Du musst dabei die Blöcke aneinanderhängen – wie bei Puzzle­teilen. Um das Programm zu starten, klicke auf die grüne Fahne  oberhalb des Spielfelds.

Um nicht mehr benötigte Blöcke zu löschen, kannst du sie einfach wieder in den Vorrat schieben.

**Vorbereitungen für das Labyrinth-Programm:**

a) Für das Labyrinth gibt es bereits einen passenden Hintergrund (in Scratch „Bühne“ genannt). Er befindet sich in der Datei *Spielfeld\_1*. Klicke im Bühnen-Bereich auf „Bühnenbild hochladen“. (Um das in der Abbildung rechts gezeigte Menü zu sehen, musst du mit der Maus ohne zu klicken auf das Bühnenbild-Wählen-Symbol fahren.)   
Du erfährst von deiner Lehrerin bzw. deinem Lehrer, wie du die Datei findest. Wähle die Datei aus und klicke auf Öffnen. Nun solltest du im Spielfeld-Bereich oben rechts das gleiche Labyrinth wie im folgenden Bild sehen.

****b) Klicke nun im Figuren-Bereich auf die „Figur 1“und in der linken Hälfte des Scratch-Bildschirms oben auf „Kostüme“.Nun siehst du statt des Skript-Bereichs den „Kostüme-Bereich“. Klicke ganz links unten im Kostüme-Bereich auf das Kostüm-Wählen-Symbol . Es öffnet sich ein Fenster mit vielen Kostümen. Klicke oben auf Tiere und wählen dann das Kostüm „cat2“, welches die Katze von oben zeigt.

Kostüme-Bereich

c) Die Katze ist für unser Labyrinth zu groß. Ziehe im Kostüme-Bereich mit der Computer-Maus ein Fenster um die Katze. Achte darauf, dass die *ganze Katze* in diesem Fenster enthalten ist. Klicke dann eine Ecke des Fensters an und verkleinere die Katze auf ungefähr die Hälfte.  
Dabei hat sich die Katze wahrscheinlich ein wenig verschoben. Schiebe die Katze wieder zurück auf die Markierung in der Mitte des Kostüme-Bereichs. Der Bauch der Katze sollte möglichst genau auf der Markierung liegen. Denn die Markierung ist der Drehpunkt, um den sich die Katze später in deinem Programm drehen wird.

d) Teste nun, ob die Katze klein genug ist: Klicke im Spielfeld (rechts oben im Scratch-Fenster) auf die Katze und schiebe sie in den grauen Startbereich unter dem Labyrinth. Die Katze muss etwas kleiner sein als der graue Startbereich. Falls nicht, verkleinere im Kostüme-Bereich die Katze noch ein wenig mehr.

e) Jetzt ist alles vorbereitet, um später das Labyrinth-Spiel zu programmieren. Damit du nachher beginnen kannst, ein Programm für die Katze zu schreiben, klicke oben links auf „Skripte“. Jetzt solltest du wieder den Skript-Bereich sehen. Gehe nun auf „Datei“ und dann auf „Auf deinem Computer speichern“ und **speichere** alles unter *Labyrinth\_1*. Halte dich dabei an die Hinweise deiner Lehrerin bzw. deines Lehrers, in welchem Ordner du deine Programme speichern sollst.

**Das mathematische Labyrinth – Teil 1**

**Aufgabe 1:**

Wähle  **Ereignisse**  aus und ziehe den „Wenn grüne Fahne angeklickt“ Block in den Skript-Bereich.

Wähle nun Bewegungen aus und ziehe den -Block direkt unter den ersten Block. Beide Blöcke müssen dabei wie Puzzleteile einrasten. In diesem „gehe zu“-Block kannst du x- und y-Koordinaten eintragen. Tippe in dem Block für x den Wert -180 und für y den Wert -120 ein. Starte dann das Programm, indem du auf die grüne Fahne klickst, und beobachte, was passiert.

Um herauszufinden, welche Koordinaten eine Stelle auf dem Spielfeld hat, kannst du mit Maus dorthin fahren und die Koordinaten am unteren Spielfeldrand ablesen. Versuche mit der Maus in die Nähe der Stelle x = 0, y = 0 und in die Nähe der Stelle x = +180, y = -120 zu fahren.

Nun soll die Katze ganz nach rechts laufen. Ziehe dazu den Block  direkt unter die ersten beiden. In diesem Block kann man eingeben, wie weit sich die Figur bewegen soll. Finde heraus, wie groß der Schritt sein muss.

**Speichere** dein fertiges Programm unter dem Namen Aufgabe1. Klicke dazu oben links auf „Datei“, dann auf „Speichern unter“. Halte dich an die Hinweise deiner Lehrerin bzw. deines Lehrers, in welchem Ordner du deine Programme speichern sollst.

**Zusatzaufgabe:**  
Schreibe ein Programm, in dem die Katze zweimal hintereinander nach rechts und wieder zurück läuft.

**Aufgabe 2:**

Die Katze soll am Anfang des Spiels immer im grauen Startbereich sein, egal wohin sie vorher gelaufen ist oder verschoben worden ist.

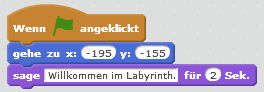
Klicke im Figuren-Bereich auf die **Figur 1** (das ist unsere Katze, die man von oben sieht). Im **Skripte**-Bereich kannst du jetzt ein Programm für die Figur 1 schreiben.

Ziehe wie in Aufgabe 1 einen „Wenn grüne Fahne angeklickt“-Block nach rechts. Ergänze das Programm so, dass die Katze beim Klicken auf die grüne Fahne in den grauen Startbereich gesetzt wird.

Teste das Programm, indem du zuerst die Katze von Hand irgendwo hin verschiebst und dann auf die grüne Fahne klickst.

**Merke dir: Um für eine Figur ein Programm zu schreiben oder zu ändern, muss man erst die Figur auswählen. Die gewählte Figur erkennst du am blauen Rahmen. Das Auswählen der Figur ist immer dann wichtig, wenn vorher eine Bühne oder eine andere Figur ausgewählt worden war.**

**Aufgabe 3:**

Im Labyrinth-Spiel soll der Spieler bei jedem Schritt gefragt werde, wie sich die Katze drehen soll, und dann wie weit sie geradeaus gehen soll.

Zunächst soll der Spieler begrüßt werden. Ziehe aus dem Bereich Aussehen den Block “sage … für … Sek.“ in das Programm. Tippe „Willkommen im Labyrinth ein.“

Ergänze das Programm um einen -Block, den du unter Fühlen findest. Schreibe in den weißen Bereich statt „What’s your name?“ die Frage „Drehwinkel?“.

Starte das Programm und gib als Antwort z.B. 45 ein (Eingabe mit Klicke auf den Haken oder mit der Enter-Taste beenden).   
Nun soll sich die Katze auch drehen. Ergänze dazu im Programm den Block . Ersetze nun die „15“ im weißen Feld durch das Element  aus dem Bereich Fühlen.

Teste das Programm. Wie erhältst du eine Drehung um 90° nach links? Wie eine Drehung nach rechts?

**Merke dir: Elemente mit runden Ecken können in weiße Felder eingesetzt werden.**

**Aufgabe 4:**

Startet man das Programm mehrmals hintereinander und dreht man darin jeweils die Katze, schaut sie zu Beginn des nächsten Programmstarts in eine andere Richtung als zuvor.

Versuche mit einem passenden Block aus dem Bereich Bewegungen, dafür zu sorgen, dass die Katze immer nach oben schaut.

**Speichere** dein Programm wieder unter dem Namen *Labyrinth\_1*.

**Zusatzaufgabe:**

Schreibe ein neues Programm, bei dem die Katze entlang der erlaubten Wege bis zur Mitte des Labyrinths läuft.

**Das mathematische Labyrinth – Teil 2**

**Aufgabe 5:**

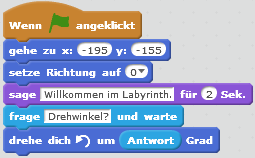
Im Labyrinth-Spiel muss die Spielerin bzw. der Spieler später mehrere Informationen eingeben, die sich das Programm merken muss. Darum geht es in dieser Aufgabe.

a) Schreibe ein Programm, das nach deinem Vornamen fragt.

b) Die Antwort soll sich das Programm nun merken. Gehe dazu in den Bereich Variablen (oder Daten), klicke auf  und gib „Vorname“ ein und klicke auf OK.   
Mit Hilfe des Blocks  kann sich das Programm z.B. die Antwort auf die Frage nach dem Vornamen merken. Dazu muss die  in das weiße Feld gezogen werden: .

c) Schreibe ein Programm, das dich erst nach dem Vornamen fragt und sich deinen Vornamen merkt und das danach nach deinem Nachnamen fragt und sich diesen auch merkt. Verstelle dazu wie in b) zwei Variablen, die Vorname und Nachname heißen sollen.

d) Nun soll die Katze dich mit Vor- und Nachnamen begrüßen. Ergänze dazu in deinem Programm den Block  aus dem Bereich Aussehen. Setze für „Hello!“ aus dem Bereich Operatoren das „verbinde“-Element  ein. Ziehe ein zweites „verbinde“-Element in dein Programm uns setzte es statt „world“ in das erste „verbinde“-Element ein. Das Ergebnis sollte so aussehen: .  
Jetzt kannst du in das erste weiße Feld „Hallo “ schreiben, in das zweite  und in das dritte .  
Teste das Programm. Du solltest nun mit Hallo und deinem Vor- und Nachnamen begrüßt werden.  
Speichere das Programm unter dem Namen Aufgabe5.



**Aufgabe 6:**

Öffne jetzt dein Programm Labyrith1 vom letzten Mal: Klicke dazu auf „Datei“ und dann auf „Öffnen“ und suche die Datei *Labyrinth1*. Klicke dann auf „Öffnen“. Du solltest das Labyrinth sehen und dein Programm sollte so aussehen wie rechts gezeigt.

In Labyrinth-Spiel soll die Spieler in bzw. der Spieler bei jedem Zug zuerst einen Drehwinkel eingeben und dann den Zähler und den Nenner einer Bruchzahl. Die Katze soll sich dann um den eingegebenen Bruchteil der Spielfeldlänge geradeaus bewegen. Gibt man z.B. den Zähler 1 und den Nenner 2 ein, so läuft die Katze  Spielfeldlänge geradeaus.

a) Ergänze das Programm so, dass es nach dem Zähler fragt und sich diesen merkt und danach nach dem Nenner fragt, den es sich auch merkt.

b) Gibt man z.B. für den Zähler 3 und den Nenner 4 ein, soll die Katze antworten „Ich gehe 3/4“. Ergänze das Programm, so dass dies funktioniert. Tipp: Du musst diesmal sogar drei „verbinde“-Elemente einsetzten und ähnlich wie in Aufgabe 5d) verschachteln.

c) Um zu verhindern, dass der Wert für den Zähler und der Wert für den Nenner die ganze Zeit auf dem Spielfeld angezeigt werden, gehe in den Bereich Variablen (oder Daten) und entferne vor allen Variablen die Häkchen.

d) Teste das Programm und **speichere** es, wenn es fertig ist unter dem neuen Namen *Labyrinth2*.

**Aufgabe 7:**

Es fehlt nun nur noch die Vorwärtsbewegung der Katze und wir haben den ersten Spielzug programmiert. Die Seitenlänge des Spielfelds beträgt 300. Wenn ein Spieler für die Vorwärtsbewegung den Bruch eingegeben hat, soll die Katze also 150 weit nach vorne gehen. Im Spiel muss der Computer diesen Wert aus dem Zähler, dem Nenner und der Spielfeldlänge selbst ausrechnen.

a) Notiere die Rechnung, die der Computer machen muss.

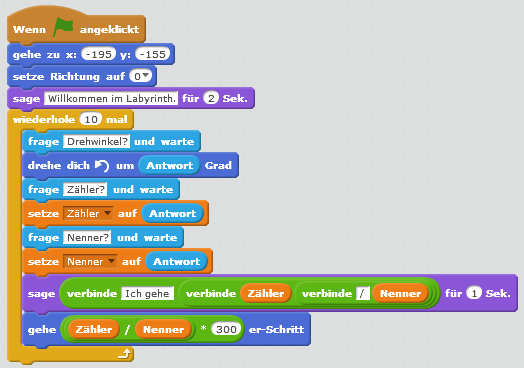
b) Im Bereich Operatoren findest du die Rechen-Elemente und  für das Multiplizieren und Dividieren. Wie schon beim „verbinde“-Element kann man diese Elemente ineinander einsetzten.   
Probiere das einfach einmal aus: Ziehe das Dividieren-Element in das erste weiße Feld des Multiplizieren-Elements und du erhältst . Dabei wird zuerst das erste Feld durch das zweite dividiert und anschließend das Ergebnis mit dem dritten Feld multipliziert. Die gerade erstellte Rechnung würde man mit Klammern so aufschreiben (  :  ) ⋅ .   
  
Benutze diese Operatoren, um die Katze passend zur Eingabe des Spielers vorwärts gehen zu lassen. Erinnerung: Die Variablen und  , mit denen sich das Programm die eigegebenen Werte merkt, findest du im Bereich Variablen (oder Daten).  
   
Teste dein Programm mit verschiedenen Brüchen.

**Das mathematische Labyrinth – Teil 3**

Im Labyrinth-Spiel besteht ein Zug daraus, dass der Spieler einen Drehwinkel eingibt und danach Zähler und Nenner für den Vorwärtslaufen der Katze. Bisher kann der Spieler nur einen Zug machen. Um das zu ändern, verwenden wir ein **Wiederhole**-Element. Wiederhole-Elemente kann man benutzen, um einen Programmteil mehrmals zu wiederholen.

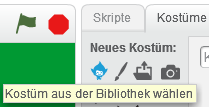
**Aufgabe 8:**

Öffne das Programm *Labyrinth2*, das du im Teil 2 geschrieben hast. Suche die Blöcke, die zu dem oben beschriebenen Spielzug gehören. Wenn du den obersten davon anpackst, kannst du alle diese Blöcke zusammen ein wenig von den ersten Blöcken wegziehen. Lasse sie aber im Skript-Bereich, da du sie noch brauchst.

**Aufgabe 9:**   
Im Bereich Steuerung findest du den Block „wiederhole … mal“ ****. Ziehe diesen direkt unter den Block mit der Begrüßung. Danach ziehst du die Blöcke für einen einzelnen Zug in die Mitte des Wiederhole-Blocks. Das Ergebnis sollte so aussehen:  
  
  
  
  
  
  
  
  
  
  
  
  
Starte das Programm und versuche in den 10 Zügen möglichst weit durch das Labyrinth zu kommen, ohne die blauen Wände zu berühren.

**Aufgabe 10:**

In Aufgabe 8 hat eine Spielerin bzw. ein Spieler genau 10 Züge, obwohl er entweder mehr oder weniger Züge bräuchte. Um das Problem zu beheben, benutzen wir einen anderen Wiederhole-Block, der so lange wiederholt wird, bis die Katze eine andere Figur erreicht hat, die wir am Ziel platzieren.

a) Klicke im Figuren-Bereich unter neue Figur auf das Figur Wählen-Symbol  (oder ) und wähle den Donut aus. Schiebe den Donut auf die grüne Fläche hinter dem Ziel. Die Katze  
muss nun den Donut berühren, um das Spiel zu beenden. Wenn du magst kannst   
du den Donut auch verkleinern, siehe das Blatt „Informationen zu Scratch“.

b) Um nun das Programm für die Katze zu ergänzen, denke daran, erst auf die Katze (Figur 1) zu klicken. Ersetze im Programm den „wiederhole … mal“-Block durch den „wiederhole bis …“-Block aus dem Bereich Steuerung. Ziehe dazu erst die Blöcke im Inneren des alten Wiederhole-Blocks zur Seite, entferne den alten Wiederhole-Block und füge den neuen ein. Ergänze danach aus dem Bereich Fühlen die Bedingung  und stelle   
dort Donut ein.

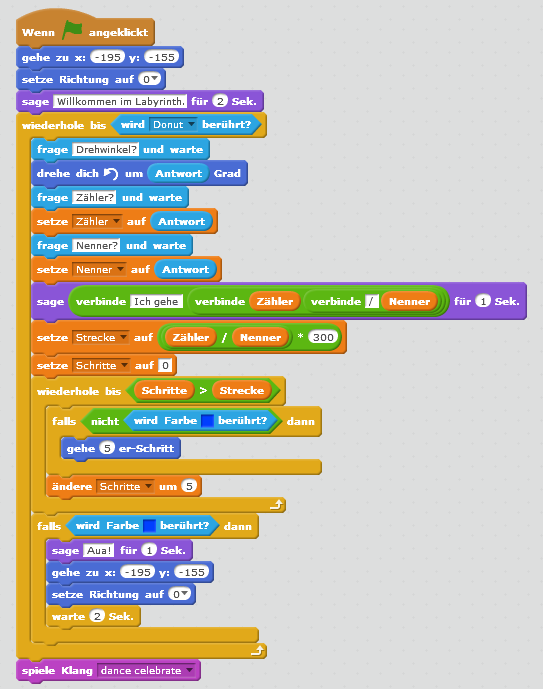
**Aufgabe 11:**

Ergänze dein Programm um einen kleinen Freudentanz, den Katze aufführt, nachdem sie den Donut erreicht hat. Probiere dabei gerne auch neue Blöcke aus dem Bereich Bewegung aus. Du kannst auch eine Schleife verwenden, die die Bewegung ein paar Mal wiederholt.   
Teste dein Programm und **speichere** es unter dem Namen Labyrinth3.

**Jetzt kannst du dein selbst programmiertes Labyrinth-Spiel spielen!**

**\*Aufgabe 12:**

Ein Nachteil beim bisherigen Programm ist, dass man die Katze ungestraft durch die Wände hindurch steuern kann. Das wurde im folgenden Programm geändert.



Schaue das Programm genau durch. Markiere die Stellen, an denen es von deinem abweicht.

Ändere dein Programm so ab, dass es den hier gezeigten entspricht. **Speichere es unter dem neuen Namen** *Labyrinth4*.

Spielt das Spiel und beschreibt möglichst genau die Unterschiede zu eurem Programm aus Aufgabe 11.

\*\* Versuche deiner Nebensitzerin bzw. deinem Nebensitzer zu erklären, wie die Änderungen im Programm-Skript funktionieren. Das ist eine wirklich schwere Aufgabe. Wenn du nicht weiterkommst, frage deine Lehrerin bzw. deinen Lehrer.

|  |
| --- |
| Spielfeld 1 |
|  |
| Spielfeld 2 |
|  |

**Verlaufsplan**

SuS … Schülerinnen und Schüler L … Lehrerin bzw. Lehrer

EA … Einzelarbeit PA … Partnerinnen- bzw. Partnerarbeit UG … Unterrichtsgespräch

Die Zeitangaben dienen nur zur groben Orientierung! Das Modul kann auch nur 2 statt 3 Stunden dauern. Wie viel Zeit benötigt wird, hängt stark von den Vorkenntnissen und vom Unterrichtsverlauf ab.

|  |  |  |
| --- | --- | --- |
| **Phase / Zeit** | **L / SuS** | **Medien** |
|  | **1. Stunde** |  |
| **1. Einstieg**  Demo und UG  10 min | L stellt Scratch vor und fragt nach Vorkenntnissen.  SuS oder L erklären die Scratch-Programmierung an einem Beispiel.  L benennt das Labyrinth-Spiel als Ziel des Moduls. | Computer und Beamer |
| **2. Erarbeiten, Üben**  PA / EA 10 min | SuS bearbeiten die Aufgabe 1 des Arbeitsblattes 1  L gibt individuelle Hilfen und lobt | Arbeits­blatt 1,  Computer bzw. Tablets |
| **3. Vorbereitung Labyrinth**  Demo und Anleitung  15 min | L geht mit den SuS die Vorbereitung für das Labyrinth-Programm schrittweise gemeinsam durch und demonstriert die Schritte über den Beamer. L gibt Hinweise zum Speichern am richtigen Speicherort und zum Öffnen von Dateien.  Kurze Zusammenfassung an der Tafel | Arbeits­blatt 1,  Beamer  Tafel |
| **4. Erarbeitung**  EA / PA 15 min | SuS bearbeiten Arbeitsblatt 1, Aufgaben 2 bis 4.  L gibt Hilfen und lobt. | Arbeits­blatt 1,  Computer bzw. Tablets |
| **5. Besprechung** Plenum10 min | Lösungen werden vorgestellt | Beamer |
|  | **2. Stunde** |  |
| **1. Einstieg**  UG 10 min | Erkenntnisse und Ergebnisstand der letzten Stunden wiederholen  L erläutert Spielablauf beim Labyrinth-Spiel genauer | Computer und Beamer |
| **2. Erarbeitung**  EA / PA 20 min | SuS bearbeiten Arbeitsblatt 2, Aufgaben 5 und 6.  L gibt Hilfen und lobt. | Arbeits­blatt 2,  Computer bzw. Tablets |
| **3. Besprechung**  Plenum10 min | Lösungen werden vorgestellt  L erinnert an Speicherort | Beamer |
| **4. Erarbeitung**  EA / PA 10 min | SuS bearbeiten Arbeitsblatt 2, Aufgabe 7.  L gibt Hilfen und lobt. | Arbeits­blatt 2,  Computer bzw. Tablets |
| **5. Besprechung** Plenum10 min | Lösungen werden vorgestellt | Beamer |

|  |  |  |
| --- | --- | --- |
|  | **3. Stunde** |  |
| **1. Einstieg**  UG 5 min | Erkenntnisse und Ergebnisstand der letzten Stunden wiederholen  L wiederholt Infos zum Speichern/ Öffnen von Dateien | Computer und Beamer |
| **2. Erarbeitung**  EA/PA 10 min | SuS bearbeiten Arbeitsblatt 3, Aufgaben 8 bis 11.  L gibt Hilfen und lobt. | Arbeits­blatt 3,  Computer bzw. Tablets |
| **3. Vorstellung**  Plenum5 min | Die fertigen Programme werden vorgestellt. | Computer und Beamer |
| **4. Üben**  EA / PA 10 min | SuS spielen ihr selbst erstelltes Labyrinth-Spiel. | Computer bzw. Tablets |
| **4. Erarbeitung**  EA / PA 20 min | SuS bearbeiten Arbeitsblatt 3, Aufgabe 12.  L gibt Hilfen und lobt. | Arbeits­blatt 3,  Computer bzw. Tablets |
| **5. Besprechung** Plenum10 min | Die Programmvariante aus Aufgabe 12 wird vorgestellt und Unterschiede werden diskutiert. | Computer und Beamer |
| **Erweiterungen** | SuS ergänzen ihre Programme mit eigenen Ideen.  SuS spielen ihre Programme auf dem anspruchsvolleren *Spielfeld2.*  SuS schreiben das Programm um in eine Variante mit  2 Levels (entsprechend den beiden Spielfeldern).  SuS erstellen weitere eigenen Spielfelder.  … | Computer und Beamer |