





Und Drachen gibt es doch...

Willkommen zu Alice!!!

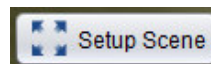
Hier könnt ihr eure eigene 3D-Welt programmieren und eurer Fantasie freien Lauf lassen. Doch damit euch das auch gelingt, solltet ihr euch erstmal unser kleines Einstiegsprojekt anschauen. Da werdet ihr viele Funktionen kennenlernen und ausprobieren, die euch nachher helfen werden. In den grünen Kästen findet ihr eure ToDos zu den einzelnen Aufgaben. Wenn ihr Hilfe braucht, oder eure Lösung überprüfen wollt, schaut auf die entsprechenden Hinweiskärtchen 

Unsere kleine Welt ist ein Sumpf, in dem eine Hexe  haust. Diese Hexe versucht sich einen Baby-Drachen  herbei zu zaubern. Mit dem möchte sie losfliegen um die Welt zu retten. Mal sehen, ob ihr das gelingt.

Doch genug gelangweilt. Jetzt dürft ihr ran...

Öffnet das Programm Alice  und wählt als Landschaft den Sumpf (*swamp*) aus.

Wechselt in der Ansicht auf *Setup Scene*.



Überlegt zuerst zu welchem Thema die Hexe gehört.

Dann sucht sie, fügt sie mittig ins Bild ein und benennt sie sinnvoll (z.B. Hexe).

Jetzt fügt noch ihren Hexenkessel (cauldron) ein, sodass er links vor der Hexe steht und benennt ihn (z.B. Kessel).



Super! Der erste Teil eurer Welt ist jetzt fertig. Und vergesst nicht zu speichern ;-)

Damit der Kamera-Ausschnitt nachher auch gut aussieht und man zum Beispiel auch den Zaubertrank im Kessel sehen kann, solltet ihr jetzt noch den Blickwinkel ändern.

Wie das geht und welche Tricks es gibt, erfahrt ihr auf der nächsten Seite...

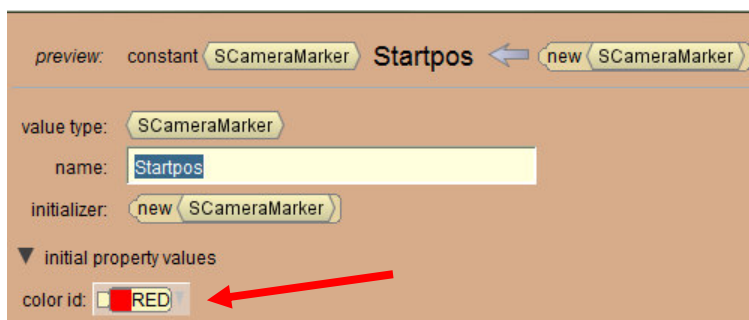
Und Drachen gibt es doch...

Bevor ihr anfangt den Blickwinkel zu verändern zuerst ein kleiner Trick, damit ihr im Sumpf nicht verloren geht. Ihr könnt einen *Camera Marker* setzen, der euch immer wieder zur Ausgangsposition der Kamera zurücksetzt.

Öffnet dazu auf der rechten Seite ganz unten den Reiter *Camera Markers*.

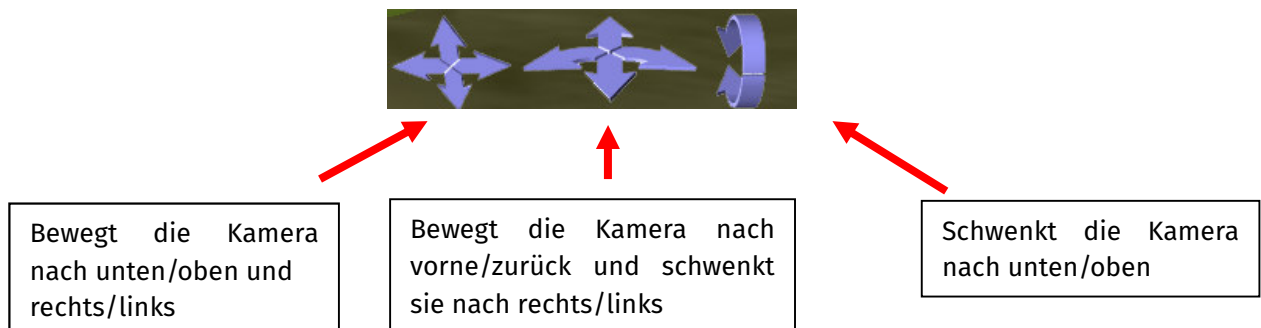
Klickt auf *Add Camera Marker*.

Gebt ihm im Pop-up-Fenster einen sinnvollen Namen (z.B. Startpos).




Achtet auf die Farbe, die euer Camera Marker hat.


Im unteren Teil des Sumpfes seht ihr drei blaue Pfeile.



Probiert einmal alle Pfeile aus und seht wie sie funktionieren.

Jetzt testet euren *Camera Marker*.  (schwarz ist die akt. Position, die farbige der entspr. Marker)

Dann ändert den Blickwinkel der Kamera so, dass man in den Kessel blicken kann.

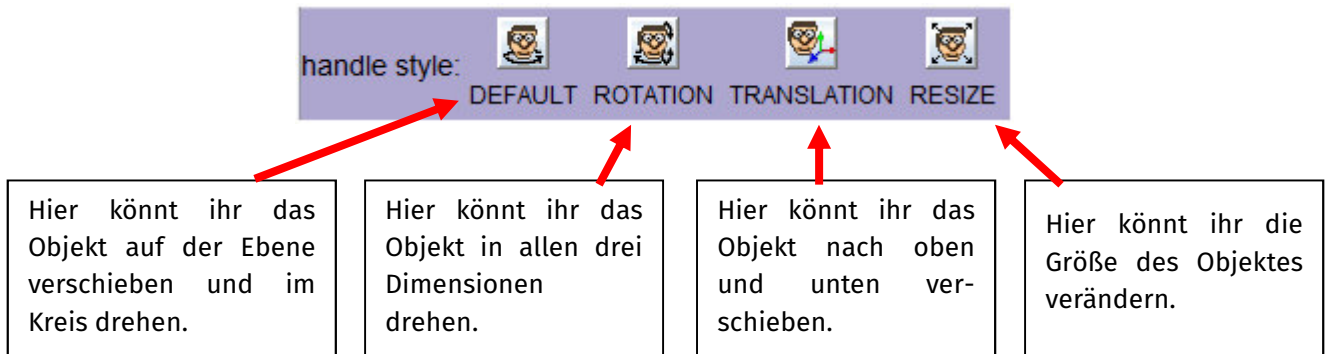
Um den *Camera Marker* nun auf die neue Position umzustellen, müsst ihr nur noch den  -Button drücken.

Sehr gut. Nun beherrscht ihr die Kameraführung. Auf zur nächsten Aufgabe...

Und Drachen gibt es doch...

Nun solltet ihr euch einmal genauer mit der Ausrichtung und Positionierung von Gegenständen beschäftigen. Dazu soll ein großer Holzlöffel so in dem Kessel positioniert werden, dass es realistisch aussieht.

Um dies zu erreichen schaut euch einmal auf der rechten Seite ganz oben den *handle style* an.



Sucht den Löffel (*MagicSpoon*) und zieht ihn in das Bild. Denkt ans Umbenennen.

Wählt *Rotation* und richtet ihn soweit auf, wie es aussähe, wenn er im Kessel steckte.

Wählt *Translation* (*Verschiebung*) und verschiebt ihn so weit nach oben, dass er aus dem Kessel rausgucken würde.

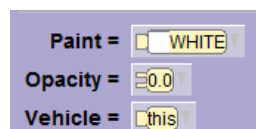
Wählt *Default* und zieht den Löffel in den Kessel.



Last but not least: Sichtbarkeit und Unsichtbarkeit

Wählt einen Baby-Drachen aus und zieht ihn in die rechte Bildhälfte.

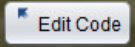
Wählt auf der rechten Seite die Eigenschaft *Opacity* (*Sichtbarkeit*) und setzt sie auf 0.



Sehr schön. Die graphische Gestaltung eurer Welt ist erst einmal abgeschlossen. Auf zur Programmierung...

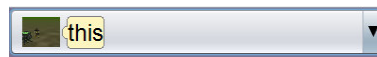
Und Drachen gibt es doch...

Denkt daran, dass ihr zum Programmieren in den *Edit Code* Bereich wechseln müsst

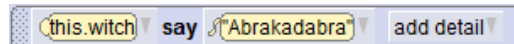
. Achtet darauf, dass ihr in *myFirstMethod* arbeitet.

Als erstes solltet ihr die Hexe zaubern lassen, dazu muss sie einen Zauberspruch aufsagen.

Wählt also links im DropDown-Menü die Hexe aus.



Zieht die Prozedur *say text* in die Programmierfläche.



Wählt im Pop-up-Fenster *Costum TextString* aus und gebt euren Zauberspruch ein.

Klickt auf *run* und testet euer erstes kleines Programm.

Um euer Programm etwas interessanter zu gestalten sorgt dafür, dass euer Löffel sich im Kessel dreht, also den Zaubertrank durchrührt.

Wählt im DropDown-Menü den Löffel aus.

Sucht die Methode *turn* (unter *orientation*) und zieht sie in die Programmieroberfläche.


Gebt als Richtung (*direction*) rechts an und als Anzahl der Drehungen (*Amount*) 2.0 an.

Testet mit *run*.

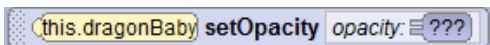
Naja, so ganz macht der Löffel ja noch nicht, was er soll...

Klickt auf *addDetails* in der *turn*-Methode. Wählt unter „as seen by“ *this.Kessel* aus.

Wenn ihr nun das Programm testet, werdet ihr feststellen, dass sich der Löffel viel zu schnell bewegt.

Kickt erneut auf *addDetails* und stellt die Dauer (*duration*) unter *CostumDecimalNumber* so ein, dass er sich so schnell dreht wie ihr wollt. 

Im nächsten Schritt soll nun euer Drache erscheinen.

Sucht die Methode  und setzt die Sichtbarkeit auf 1.0.

Lasst euren Baby-Drache denken: „Mama?“

Prozedur:

Eine Prozedur ist eine Methode zu einer Klasse, ohne Rückgabewert. Sie beinhaltet eine Abfolge von Befehlen und hat einen Namen. Mit diesem Namen kann sie aufgerufen werden.

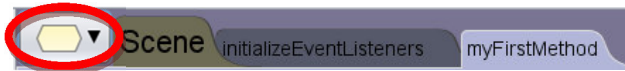
Das wäre geschafft! Und wie sich das nach getaner Arbeit gehört: Party!!!

Als nächstes werdet ihr eure Hexe feiern lassen...

Und Drachen gibt es doch...

Um eure Hexe feiern zu lassen ist es sinnvoll eine neue Prozedur zu erstellen, damit euer Programm übersichtlich bleibt.

Klickt im Reiter oben links auf das Menü.



Wählt dann unter *Witch* „Add Witch Procedure“ aus und gebt ihr einen Namen.

Super, ihr habt eure erste eigene Prozedur erstellt. Jetzt muss diese nur noch mit Inhalt gefüllt werden.

Wie ihr seht, hat sich ein neuer Reiter geöffnet, hier programmiert ihr eure Prozedur. Eure Hexe soll nun sagen: „Juhu, ich bin unbesiegbar!“ und anschließend einen Rückwärtssalto machen. Der Salto besteht aus drei Bewegungen: Springen, Drehen, Landen

Lasst eure Hexe „Juhu, ich bin unbesiegbar!“ sagen.

Benutzt die *move*-Methode um die Hexe 1.0 Meter hüpfen zu lassen.

Benutzt die *turn*-Methode um eure Hexe 1.0 Rückwärtssalto machen zu lassen.

Lasst eure Hexe landen.

Sehr gut, schon ist eure Party-Prozedur fertig. Aber bisher wird sie noch nicht ausgeführt. Dazu muss sie in *myFirstMethod* aufgerufen werden.

Wechselt den Reiter auf „*myFirstMethod*“ und wählt die Hexe aus.

Fügt eure Prozedur an das bisherige Programm an, und testet. 

Ein weiterer Vorteil von Prozeduren ist, dass man damit eine Abfolge von Befehlen öfter hintereinander ausführen lassen kann. Und das ist auch noch ganz einfach, ihr müsst nur eure Prozedur mehrmals hintereinander aufrufen.

Lasst eure Hexe nach der Beschwörung des Drachen dreimal feiern und testet euer Programm.

Und Drachen gibt es doch...

Sehr gut. Jetzt ist eure Hexe richtig gut drauf. Wenn ihr etwas wiederholen wollt, gibt es ein Konzept, das euch dabei hilft: Die Schleife

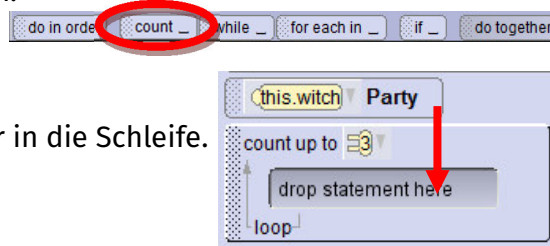
Löscht die letzten beiden Party-Methoden. (Rechtsklick ▶Delete)

Wählt in der Kontrollzeile unter der Programmierfläche `count` aus, zieht es in euer Programm.

Setzt den Zähler auf 3.

Zieht eure Party-Prozedur in die Schleife.

Testet das Programm.



Schleifen:

Eine Schleife ist ein Programmierkonzept, das euch erlaubt eine Abfolge von Befehlen mehrmals hintereinander ausführen zu lassen.

Die Count up-Schleife erhöht dabei eine Zählvariable bei jedem Durchlauf um 1, so lange bis der angegebene Wert erreicht ist.

Das Gute an Schleifen ist, dass ihr etwas beliebig oft wiederholen könnt, indem ihr einfach den Zähler ändert.

Sehr gut, jetzt habt ihr schon viele Funktionen von Alice kennengelernt und ausprobiert. Doch was ist eine Welt schon, ohne dass man in sie eingreifen kann?

Deshalb lernt ihr nun noch, wie ihr den User (Zuschauer) in eure Welt mit einbeziehen könnt. Dazu soll eure Hexe nun auf dem Drachen davonfliegen und zwar, wenn ihr die Taste „F“ (wie Fliegen) drückt. Damit das Programm auf einen Tastendruck reagieren kann, braucht ihr einen *EventListener*, der abfragt, ob ein bestimmtes Ereignis z.B. das Drücken einer Taste eintritt.

Doch als erstes solltet ihr eine neue Prozedur für den Baby-Drachen schreiben, damit er fliegen kann.

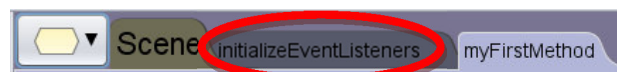
Erstellt für den Babydrachen eine entsprechende Prozedur und benennt sie.

Lasst den Drachen 10.0 Meter nach oben steigen.

EventListener:

Ein EventListener überprüft vom Programmstart (run) an, ob ein bestimmtes Ereignis (z.B. ein Tastendruck) eintritt. Es können mehrere EventListener gleichzeitig implementiert sein, also werden mehrere Ereignisse abgefragt.

Die *EventListeners* findet ihr hier:



Wenn ihr den Reiter anklickt, seht ihr, dass es schon einen *EventListener* gibt. Dieser führt *myFirstMethod* aus, wenn die Szene aktiviert wird (run).



Und Drachen gibt es doch...

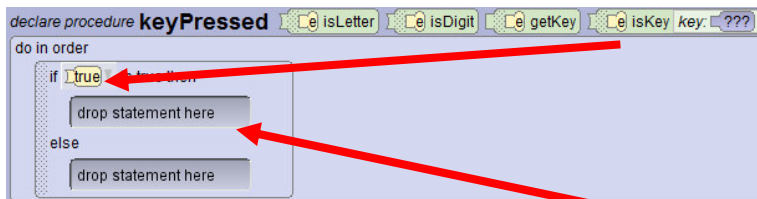
Doch nun wollen wir den kleinen Drachen fliegen lassen.

Klickt auf „Add Event Listener“ und wählt unter *keyboard*  aus.

Jetzt habt ihr einen *EventListener* erstellt, der abfragt, ob eine Taste gedrückt wurde. Damit er auf eine bestimmte Taste reagiert, braucht ihr eine *if-Abfrage*.

Zieht ein  in den *EventListener* und wählt *true* aus.

Ersetzt *true* durch „*e isKey*“ und wählt unter *Letters* „*F*“ aus.



Lasst jetzt den Baby-Drachen fliegen, wenn „*F*“ gedrückt wird. 

Testet euer Programm.

if-Abfrage:

Eine *if-Abfrage* ist eine Kontrollstruktur, die es ermöglicht auf eine bestimmte Bedingung zu reagieren. Ihr könnt angeben, was passieren soll, wenn diese erfüllt (*true*) ist und was passieren soll, wenn diese nicht erfüllt ist (*else*).

Den *else*-Bereich könnt ihr einfach leer lassen, denn bei jeder anderen Taste soll ja nichts passieren.

Jetzt fliegt euer Babydrache los, wenn ihr auf „*F*“ drückt. Dabei ist es egal, ob er schon sichtbar ist oder nicht.

Super! Ihr könnt euren Drachen fliegen lassen. Doch was ist mit der Hexe?

Auf zum Endspurt...

Und Drachen gibt es doch...

Eure Hexe möchte nun mit ihrem Drachen losfliegen um die Welt zu retten. Dazu geht sie zu dem Drachen und sagt währenddessen „Lass uns die Welt retten, Kleiner!“.

Damit sie dann auch tatsächlich mit abhebt, wenn ihr „F“ drückt, solltet ihr der Hexe noch den Drachen als Gefährt (*vehicle*) zuweisen. Das bedeutet, dass immer wenn sich das Drachen-Baby bewegt, sich auch die Hexe mit bewegt. Diese drei Befehle sollen zusammen (*together*) ausgeführt werden, also gleichzeitig.

Denkt daran in *myFirstMethod* zu wechseln.

Zieht aus der Kontrollzeile ein *do together* in die Programmieroberfläche.

Benutzt die *move to*-Methode um die Hexe zum Drachen gehen zu lassen.

Lasst sie gleichzeitig ihren Spruch aufsagen.

Sucht  und setzt als *vehicle* das Drachen-Baby ein.

Sehr gut, jetzt seid ihr fast fertig. H3

Doch woher soll der User nun wissen, dass er „F“ drücken soll?

Lasst den Baby-Drachen anschließend (also nach dem *together*) etwas Entsprechendes denken.

Speichert und testet eure Welt!!!

Jetzt könnt ihr eure Welt noch ein bisschen schöner und kreativer gestalten. Ihr könnt der Hexe z.B. eine Höhle als Bleibe geben und alles noch etwas mit Pilzen und Trankfläschchen dekorieren. Schaut euch mal die *FantasyGallery* an und tobt euch aus.



Super!!!!

Nun seid ihr bereit eure eigene Welt zu programmieren...

Quellenverzeichnis

Alle Abbildungen sind Screenshots der Alice-Software (von der Carnegie Mellon University entwickelt und unter der BSD-Lizenz veröffentlicht)

Do together:

„Do together“ ist eine Kontrollstruktur, die die Befehle gleichzeitig ausführt. Im Gegensatz dazu werden bei „do in order“ die Befehle der Reihe nach ausgeführt.