

Schritt-für-Schritt-Anleitung

Beispiel: Simulation einer Epidemie

Einordnung des Themas

Diese Simulation stellt dar, wie sich eine Epidemie innerhalb einer Population (z.B. in der Bevölkerung) ausbreitet. Man kann die Bevölkerungsdichte verändern und beobachten, wie dadurch die Ausbreitungsgeschwindigkeit der Krankheit beeinflusst wird. Zur besseren Veranschaulichung dieses Prozesses wird mit Squeak-Etoys in einer Liniengraphik die Anzahl der Infizierten im zeitlichen Ablauf dargestellt.

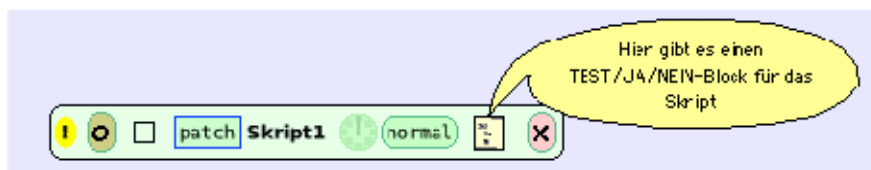
Dieses Beispiel unterstützt die Entwicklung unterschiedlicher Inhalts- und Prozesskompetenzen bei den Schülerinnen und Schülern (vgl. Grundsätze und Standards für die Informatik in der Schule). Sie analysieren und **strukturieren** den Prozess, so dass die umzusetzenden Modellkomponenten festgelegt werden können. Die Schüler **entwerfen und realisieren die erforderlichen Algorithmen**. Sie äußern Vermutungen über das mögliche Verhalten des Systems. Die Ergebnisse bei unterschiedlicher Bevölkerungsdichte **interpretieren** sie dann sachgerecht in Abhängigkeit der gewählten Abstraktion. Für die Darstellung der Ergebnisse werden verschiedene **Darstellungsformen** angeboten, aus denen sie dann die geeignete **auswählen** können. So lässt sich die Anzahl der infizierten Personen als Zahlenwert oder als Liniengraphik darstellen. Schließlich sollten die Ergebnisse der Simulation **begründet** und **bewertet** werden.

Erforderliche Vorkenntnisse

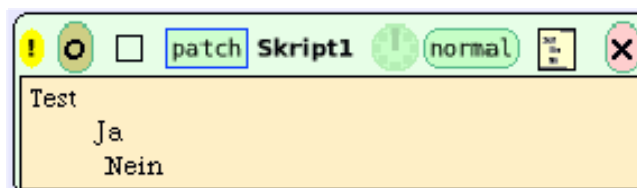
Für dieses Beispiel brauchen wir **Test-Blöcke** und **Variablen**. Beide Konzepte stammen aus in Squeak-Etoys und sollen hier kurz erklärt werden.

Testblöcke

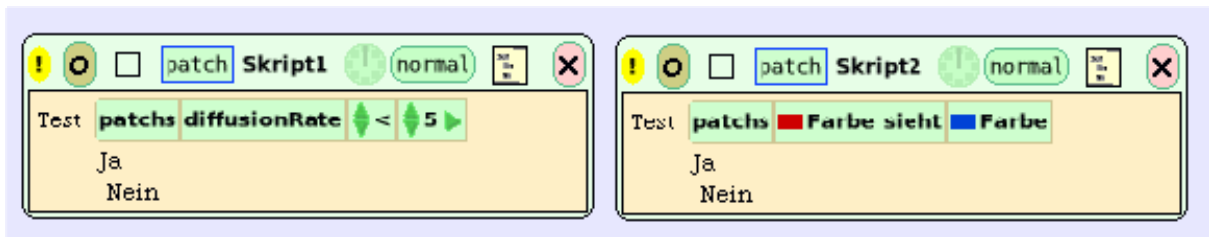
- werden in Skripten verwendet, um Alternativen zu modellieren



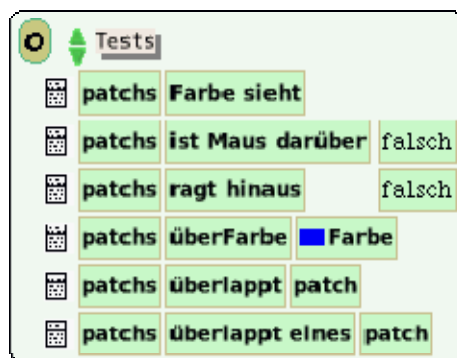
- es wird eine Bedingung formuliert, die entweder wahr oder falsch ist



- Bedingung ist numerischer Vergleich oder Kachel aus der Test-Kategorie eines Objektes



- Kacheln ermöglichen visuelle Vergleiche (ob sich eine Farbe über einer anderen Farbe befindet, ob sich der Mauszeiger gerade über dem Objekt befindet)



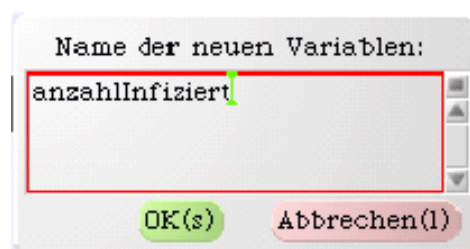
verfügbare Testkacheln für patch-Objekt

Variablen

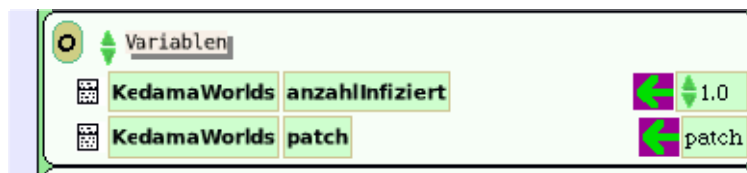
- Variablen werden im Betrachter des Objektes erzeugt, dem sie zugeordnet werden sollte
- in der Kopfzeile des Betrachters kleines "v"-Symbol anklicken



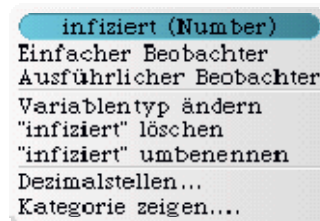
- Name kann im folgenden Dialog eingegeben werden



- Datentyp standardmäßig *number*, kann über Menü geändert werden



- kleines Menüsymbol vor der Kachel mit der neuen Variablen mit folgendem Inhalt zu finden:



- Ein Beobachter ist eine Kachel mit dem aktuellen Wert bzw. zusätzlich dem Namen der Variablen. Damit kann man während der Simulation die Veränderungen der Werte beobachten.
- Variablen können prinzipiell in beliebigen Objekten erstellt werden, es ist aber besser, sie jeweils sinnvoll zuzuordnen

Anleitung Simulationsbeispiel

Problemstellung:

Wie schnell können sich ansteckende Krankheiten ausbreiten?

Dafür sind verschiedene Einflussfaktoren verantwortlich. Ein wichtiger Punkt ist, auf welchem Weg die Krankheiten übertragen werden, ob ein einfacher Kontakt mit einer infizierten Person ausreicht oder man dem Infekt über einen längeren Zeitraum ausgesetzt sein muss. Außerdem kommt es darauf an, wie oft man jemanden trifft, der infiziert ist. In einer Stadt mit großer Bevölkerungsdichte ist dies häufiger der Fall als auf dem Land. Im Beispiel gehen wir von folgenden Vorgaben aus:

- die Simulation startet mit einer infizierten Person
- jeder direkte Kontakt einer gesunden mit einer infizierten Person führt zur Ansteckung
- alle Personen bewegen sich zufällig im betrachteten Umfeld, so dass ein Kontakt zufällig stattfindet
- die Bevölkerungsdichte ist veränderbar

Analyse:

- beim Start wird ein **Turtle infiziert** - Ausgangszustand
- **Ansteckung** findet statt, wenn gesundes Turtle auf Zelle trifft, auf der auch ein infiziertes Turtle steht
- Bevölkerung ist eine **Turtle**-Art (turtle breed), hat Zustand **gesund** oder **infiziert**

- Darstellung der gesunden und infizierten Turtle durch verschiedene Farben
- Zählen der infizierten Personen

Erweiterung:

- Darstellen der Anzahl der infizierten Turtle als Liniengraphik

Umsetzung:

- eine **Kedama-Welt**
- ein **patch** (Gitterebene) zum Speichern, ob gerade ein infiziertes Turtle darauf steht.
- eine **turtle**-Art
- **im Kedama-Bundle verfügbare Komponenten sind für dieses Beispiel ausreichend**
- **Skript** zum Herstellen des **Ausgangszustandes**
- **Skript** zum **Infizieren** eines Turtles
- **Skript** zur Modellierung der **Ansteckung**
- **Variable *infiziert*** für Turtle, die speichert, ob es infiziert ist oder nicht
- **Variable *anzahlInfiziert***, die die Anzahl der infizierten Personen speichert

Erweiterung:

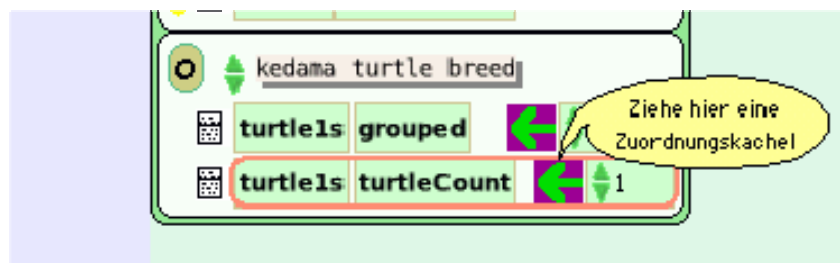
- **Liniengraphik** erstellen wie in Anleitung „Bewegungen plotten“ beschrieben

Anleitung

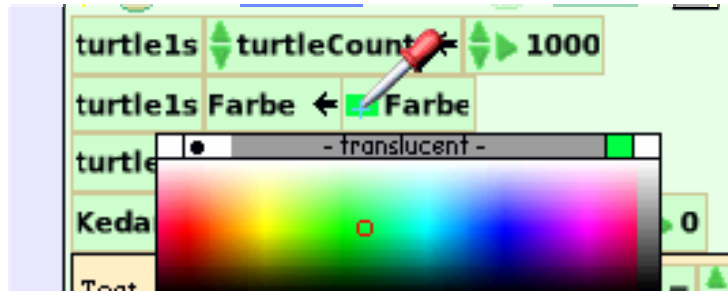
1. **KedamaBundle** öffnen wie in der Einführung erklärt

Anleitung für Skript Ausgangszustand

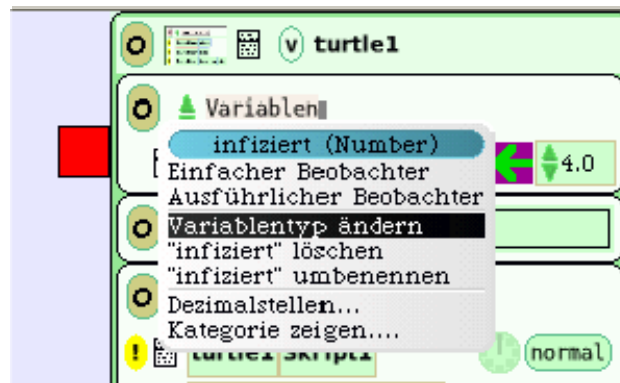
2. **Betrachter** für **turtle** öffnen
3. leeres Skript auf die Welt ziehen
4. Bevölkerungsdichte festlegen – in Kategorie *kedama turtle breed* die Kachel `turtle1s` am grünen Zuweisungspfeil ins Skript ziehen, Anzahl auf 1000 ändern



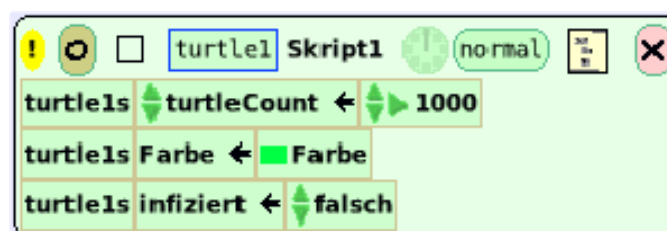
- Farben festlegen für gesunde Turtle, Kachel aus Kategorie *kedama turtle* ins Skript ziehen und Farbe wählen durch Anklicken des kleinen farbigen Feldes



- Variable für Turtle erstellen, zum Speichern, ob sie infiziert sind oder nicht. Die Variable soll "infiziert" heißen und vom Typ "Boolean" sein.



- Am Anfang soll diese Variable den Wert "falsch" haben, die Turtle sind ja nicht infiziert. Wir ziehen dazu eine Zuweisungskachel für die Variable in das Skript.



- Um zu sehen, wie viele Turtle sich gerade infiziert haben, erzeugen wir uns eine Variable "anzahlInfiziert". Diese Variable erzeugen wir im Betrachter der Kadamawelt, weil wir die Anzahl infizierter Turtle in der Welt zählen wollen.

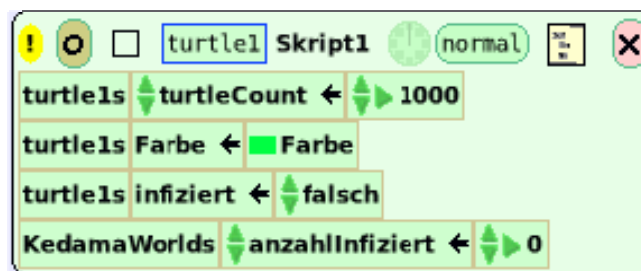


9. Wenn die Variable im Betrachter steht, holen wir uns aus dem Menü neben der Variablen einen "ausführlichen Beobachter" und platzieren ihn auf dem Bildschirm über der Kedamawelt.

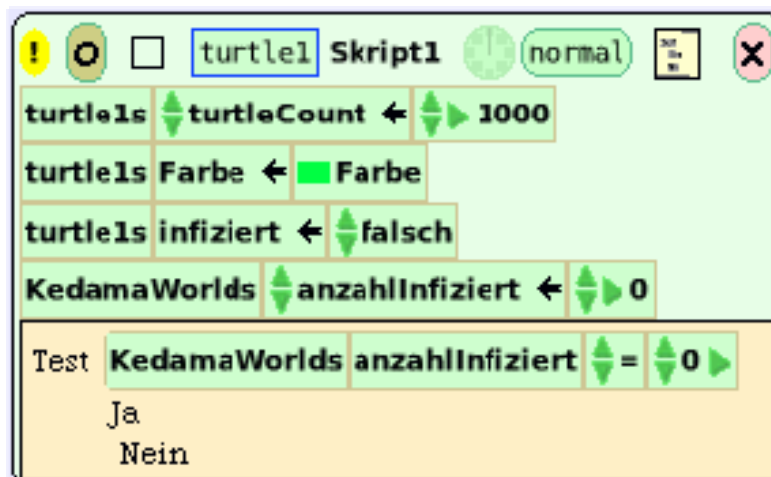


KedamaWorlds anzahlInfiziert = 0.0

10. Am Anfang gibt es keine infizierten Törtel, deshalb weisen wir der Variable "anzahlInfiziert" den Wert "0" zu.



11. Zuletzt wollen wir genau ein turtle infizieren. Dafür verwenden wir einen Testblock. Wenn die Anzahl der infizierten Turtle gleich Null ist, soll genau ein Turtle infiziert werden. Aus dem Kopf des Skriptes holen wir einen Testblock Dann wird eine Kachel mit dem Wert der Variablen "anzahlInfiziert" hinter "Test" gezogen und die Bedingung "=0" durch Anklicken der kleinen grünen Pfeile eingestellt. Diese Bedingung ist nur direkt beim Start der Simulation wahr.



12. Die Bedingung zum Infizieren des ersten Turtles haben wir schon formuliert, aber um im Ja-Zweig das passende Skript aufzurufen, müssen wir das erstmal schreiben.

Anleitung für Infizieren-Skript

13. Wie infizieren wir jetzt ein Turtle? Dazu benötigen wir ein neues Skript, das drei Aktionen ausführen muss:

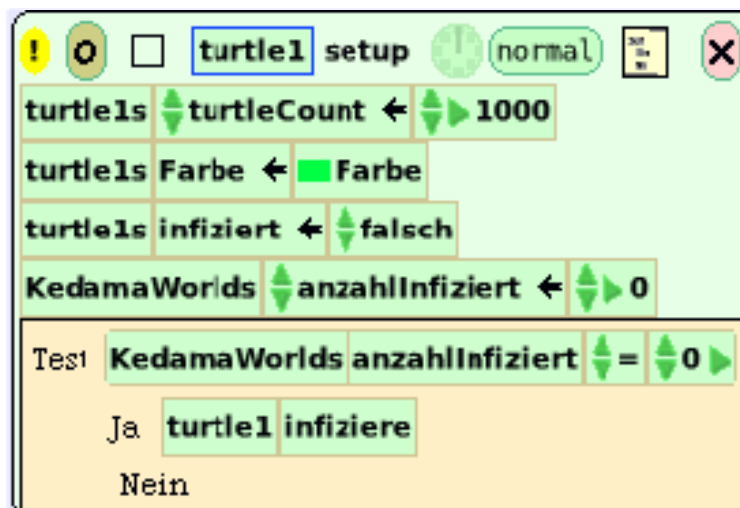
1. den Wert der **Variable** *infiziert* für das Turtle auf *wahr* ändern,
2. die Farbe des Turtles vom gesunden grün in ein krankes rot ändern,
3. den Wert der **Variable** *anzahlInfiziert* um **1 erhöhen**.

Das Skript dazu sieht folgendermaßen aus:



14. Den Namen eines Skriptes ändert man, indem man ihn einfach anklickt und dann schreibt. Dieses Skript heißt jetzt *infiziere*.

15. Nun können wir eine Kachel des Skriptes aus dem Betrachter in Skript 1 ziehen. Dieses benennen wir auch gleich um in *setup*.



16. Dieses Skript soll nicht in einer Endlosschleife laufen, sondern genau einmal. Das erreicht man entweder durch Klicke auf das gelbe Ausrufezeichen oder durch einen Knopf zum Anklicken. Damit das Skript auch aufgerufen werden kann, wenn es geschlossen ist, holen wir uns aus dem Menü einen **Knopf um dieses Skript auszuführen** und platzieren ihn auf dem Bildschirm.



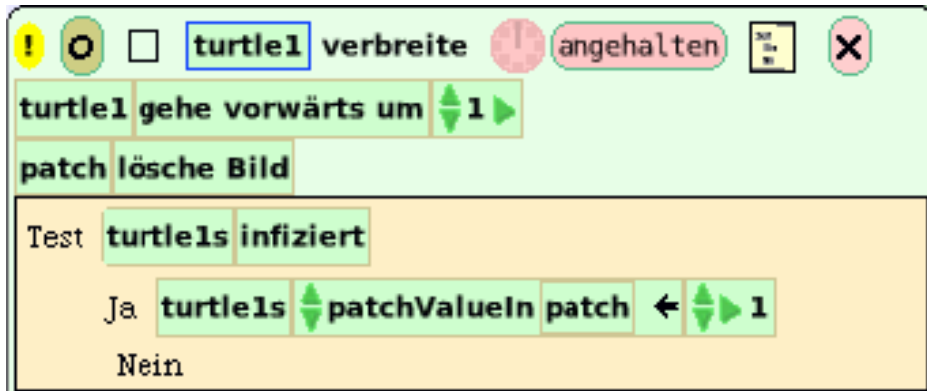
Anleitung für Verbreitungs-Skript

17. Wie verbreitet das infizierte Turtle die Krankheit? Wir brauchen ein Skript, das folgende Aktionen ausführen muss:

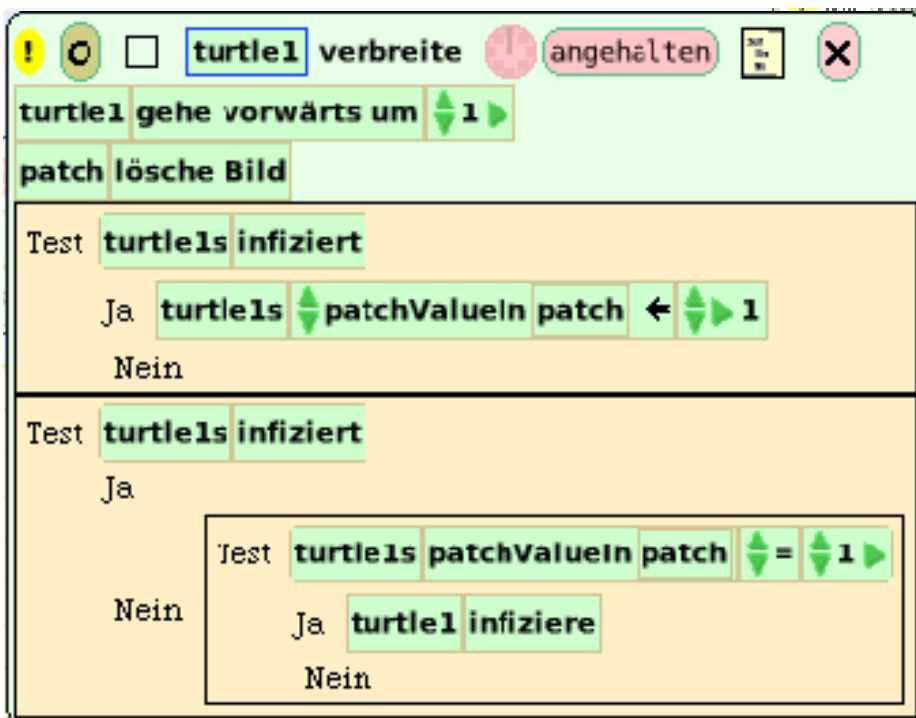
1. die **Turtle in Bewegung** setzen.
 2. dafür sorgen, dass immer, wenn sich ein infiziertes Turtle in der selben Zelle wie ein gesundes befindet, sich das gesunde Turtle **ansteckt**
18. Für die Bewegung holen wir uns die *gehe vorwärts*-Kachel für turtle1 und lassen das Turtle in jedem Skriptdurchlauf 1 Schritt vorwärts gehen.



19. Für die Überprüfung, ob ein gesundes und ein infiziertes Turtle in der selben Zelle stehen, brauchen wir ein Patch. Wir löschen zuerst die Patch-Werte aus dem vorhergehenden Schritt und ändern dann die Werte der Zellen mit infizierten Turtles auf den Wert „1“. Um herauszufinden, ob ein Turtle infiziert ist, testen wir den Wert seine *infiziert*-Variable. Zum Ändern des Wertes im Patch holen wir die Kachel "turtle1s patchValueIn" und weisen dieser den Wert „1“ zu. Damit haben wir jetzt quasi die betreffenden Zellen „infiziert“. In Kedama erfolgt eine zeilenweise Synchronisation, das bedeutet, dass jede Zeile immer für alle Turtle ausgeführt wird, ehe die nächste Zeile im Skript drankommt.

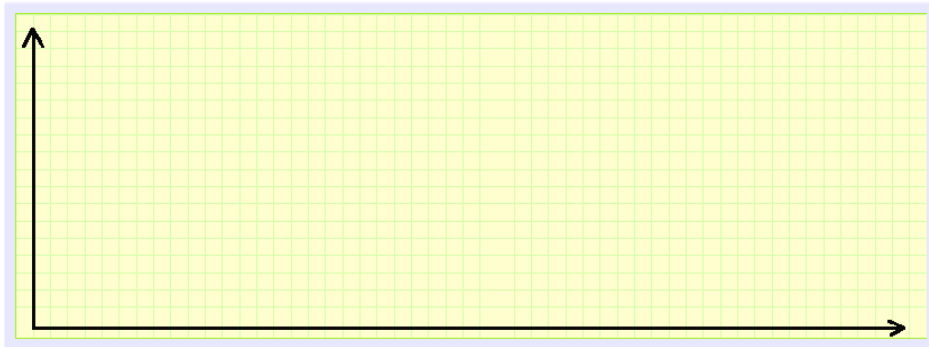


20. Schließlich müssen wir noch nach gesunden Turtles suchen, die sich anstecken. Dazu benötigen wir noch einen Testblock. Wir testen wieder für jedes Turtle, ob es infiziert ist. Wenn ja, dann müssen wir nichts tun, aber wenn nicht, dann testen wir, ob sich ein infiziertes Turtle in der gleichen Zelle befindet. Dazu fragen wir den Wert von *patchValueIn* der Turtle ab. wenn der „1“ ist, dann rufen wir das *infiziere*-Skript auf.



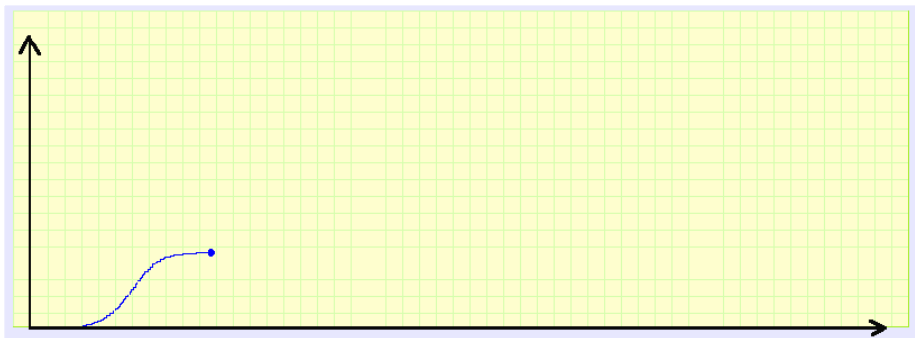
Anleitung für Erweiterung: Liniengraphik (siehe Handzettel „Bewegung plotten“)

- 21. Wir müssen kein Objekt zeichnen, da wir als Wert die Anzahl der infizierten Turtle graphisch darstellen wollen und nicht die Bewegung eines Objektes. Deshalb brauchen wir auch nur eine Spielweise.
- 22. Wir passen die Größe der Spielweise so an, dass in der Breite (x) die Zeit t und in der Höhe (y) die Anzahl abgetragen werden kann. Wir stellen auch den Millimeterpapier-Hintergrund ein (Anleitung 4 und 5). Mit dem Malwerkzeug in der Navigator-Klappe kann man auch ein Koordinatensystem zeichnen.



23. Dann zeichnen wir einen Stift (Anleitung 6 - 9).

24. Das Skript zum Zeichnen weicht von der Vorlage ab. In x-Richtung kann die Bewegung um 1 Pixel pro Schritt stehen bleiben. Die Bewegung in y-Richtung hängt nun aber vom Wert der *anzahlInfiziert*-Variable ab.



25. Damit der Stift eine Spur hinterlässt, muss noch der Wert für *Stift unten* gesetzt werden (Anleitung 12). Um die Position des Stiftes an das Koordinatensystem anzupassen, ist oftmals noch ein Offset zu addieren. Wie groß der Wert sein muss ist jeweils auszuprobieren.

26. Um nach jedem Durchlauf die Linie zu löschen und den Stift auf die Ausgangsposition zu setzen, empfiehlt sich entweder ein eigenes Setup-Skript oder eine Erweiterung des vorhandenen Skriptes.

27. Da jetzt zwei Skripte gleichzeitig laufen müssen (das verbreite-Skript und das zeichne-Skript) sollte man sich eine Skriptkontrolle aus dem Lager holen. Klickt man auf „go“, dann laufen alle Skripte los, deren Zustand „angehalten“ ist. Also muss man vorher die erforderlichen Skripte in diesen Zustand versetzen und kann diese dann mit einem Klick starten.