

Digitalwerkstatt: 5. Hausaufgabe

Erinnerung P5.js

Klickt um mit P5.js zu nutzen auf diesen Link: [p5.js Web Editor](#)

Die Bibliothek bietet eine Reihe von vorgefertigten Funktionen und Methoden, die den Einstieg in die Programmierung erleichtern. Ihr könnt sie jederzeit hier nachschlagen: [reference | p5.js](#)

⚠ Achtung

Wir haben das Feedback bekommen, dass es manchmal für euch schwer ist zu überlegen, was in die Lücken von dem Code kommt, den wir euch in den letzten Hausaufgaben zur Verfügung gestellt haben.

Vielleicht ist es einfacher, wenn ihr selbst den ganzen Code schreibt. Das probieren wir dieses Mal. Wir haben die unten stehenden Aufgaben einmal selbst implementiert und der Code ist nicht lang, allerdings werdet ihr euch wahrscheinlich mehr von der [p5.js Dokumentation](#) durchlesen, damit ihr alles richtig programmiert. Wir haben alle Funktionen aus dem Blatt für euch verlinkt.

Deswegen geben wir diesen **Mittwoch, den 9. April** um 17:30 ein Online Tutorial zum Code debuggen und ihr habt die Möglichkeit uns Fragen zu den Aufgaben zu stellen. Hier geht es zum Online-Raum: [FörderInfor | Lernplattform Bauhaus-Universität Weimar](#)

(1) Wärmetod des Universums

🕒 Aufgabe 1

In der ersten Aufgabe recherchiert ihr das Ende des Universums. Natürlich wissen wir nicht, ob und wie das Universum enden wird. Aber wir haben physikalische Modelle entwickelt, mit denen wir versuchen das Ende vorherzusagen. Das folgende Video von kurzgesagt erklärt euch drei mögliche Szenarien:

1. Big Rip
2. Big Freeze
3. Big Bounce

Beantwortet die folgende Frage mit Begründung: Welches Ende des Universums würdet ihr euch wünschen?

- Link zum Video: [Three Ways to Destroy the Universe - Kurzgesagt](#)

(2) Entropie in der Thermodynamik

🕒 Aufgabe 2

Kreiere eine Simulation, wie sich Teilchen in einem Raum durch ihre Eigenbewegung ausbreiten (Diffusion).

🕒 Aufgabe 2 (a)

Um das Mischen vieler Teilchen (Diffusion) zu simulieren, müssen wir erst die Bewegung eines einzelnen Teilchen simulieren ([Brownsche Bewegung](#) oder auch [Random Walk](#))

Implementiere die zufällige Bewegung eines einzelnen Teilchens.

💡 Tipp Vektor

Erstelle einen [Vector](#) um darin x und y Koordinaten eines Teilchen zu speichern, dass du *simulieren* möchtest. Mit der Funktion [circle\(\)](#) kannst du das Teilchen als Kreis auf dem [canvas](#) zeichnen. In der Referenz zum [Vector](#) steht auch, wie ihr dafür an die gespeicherten x und y Koordinaten kommt.

💡 Bewegung

Wie bewegen sich eigentlich kleine Teilchen? Robert Brown hat 1827 unter dem Mikroskop entdeckt, dass Teilchen sich unregelmäßig und ruckartig bewegen. Dieses zufällige Bewegungsverhalten können wir auch simulieren. Dazu wird bei jedem Aufruf der [draw\(\)](#) Methode ein zufälliger x-Wert und ein zufälliger y-Wert auf das Teilchen addiert. Zufällige Werte erhalten wir mit der [random\(\)](#) Methode.

Beachtet, dass nur positive zufällige Werte das Teilchen nur nach unten oder rechts auf dem [canvas](#) bewegen würde. Wir brauchen also auch negative Werte.

🕒 Aufgabe 2 (b)

Füge jetzt 250 Teilchen in den Sketch ein, die sich alle unabhängig voneinander zufällig bewegen.

💡 Tipp Liste

Fülle eine leere Liste durch eine [for -Schleife](#) mit den Startvektoren der Teilchen. Nun müssen wir die Teilchen noch zeichnen. Nutze dafür eine zweite [for -Schleife](#) um durch die Liste zu *iterieren* und rufe innerhalb der Schleife die Funktion [circle\(\)](#) um die Teilchen zu zeichnen und [random\(\)](#) um sie zu bewegen.

📖 Erkenntnis

Der Random Walk von Teilchen erhöht die Entropie im System. Entropie ist ein Maß für die Unordnung oder Wahrscheinlichkeit von Zuständen in einem System. Beim Random Walk bewegen sich Teilchen zufällig in alle Richtungen, wodurch sie sich im Raum verteilen. Dadurch steigt die Unordnung des Systems und die Entropie nimmt zu.

(3) Hitzetod (Fortgeschritten)

🕒 Aufgabe 3

Das "Große Gefrieren" oder auch "Hitzetod" des Universums folgt aus dem dritten Grundsatz der Thermodynamik. Wir können diesen Grundsatz selbst mit einer Simulation erforschen.

Bearbeite deinen Sketch, sodass jedes Teilchen eine "Temperatur" erhält. Die "Temperatur" ist eine Variable die bestimmt, wie weit sich ein Teilchen in einem Schritt entfernen darf. Je höher die Temperatur umso größer ist die mögliche zufällige Bewegung des Teilchens.

Die Temperatur berechnet sich daraus, wieviele andere Teilchen gerade im direkten Umfeld des Teilchens ist.

💡 Tipp

Erstelle dazu am besten eine neue Funktion mit zB. dem Namen

`getTemperature(meinteilchen, teilchenliste)` die das aktuelle Teilchen und die Liste aller Teilchen erhält. Jetzt kann in der Funktion durch die Liste iteriert werden und zB. die Anzahl aller Teilchen zurückgegeben werden, deren Abstand zum aktuellen Teilchen kleiner als eine von euch gewählte Grenze ist.