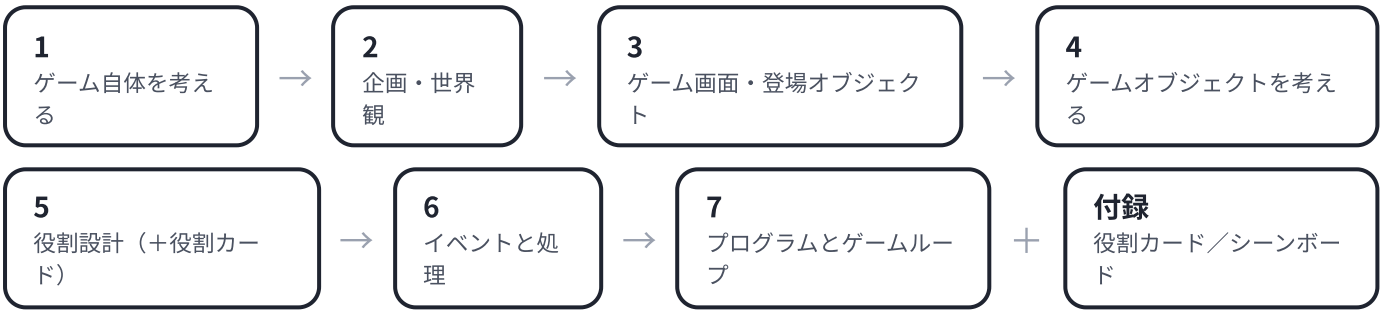
three.js ゲーム制作ワークシート

全7枚＋付録
ゲームを分解して考えるワークシート

JavaScript と three.js でゲームを作る初心者向けのワークシート、全7枚と付録のセットです。
遊びの芯から始めて、企画、オブジェクト、部品、イベント、スクリプトへ。
ゲームを分解して考える流れが、紙の上で順にたどれるようになっています。



共通のお題（全シート共通）

- ターゲットはスタートからゴールまで移動する
- プレイヤーはアイテムを配置して、ターゲットの移動を妨害する
- 5分間で目的地に到達しなければゲームクリア！

自分のお題（先生が書き込む欄）

お題を変えたら、各シートのお手本の行も、自分のお題に合わせて読み替えてください。

印刷と使い方

- ワークシートは、1人1部ずつ配ります
- お題は差し替えられます。変えた場合は、各シートのお手本（ターゲット／ゴール／妨害アイテム）も読み替えてください
- 役割カードとシーンボードは、使う人数分をコピーしてください（カードは少し厚い紙だと切りやすいです）
- 何度も使うクラスでは、カードをラミネートすると繰り返し使えます

名前：＿＿＿＿＿＿＿＿　クラス：＿＿＿＿＿＿＿＿

ワークシート 1

ゲーム自体を考える

このシートの目的： 世界観の前に、「遊びの芯」を決める

名前 _____

1 遊ぶ人と、プレイヤーの動き

遊ぶ人はだれ？（年齢や、好きそうなもの）

プレイヤーは何をする？（動詞で3つ書く： 集める／逃げる／守る／当てる など）

2 目的と、終わり方

目的（何のために動く？）

勝ち（クリア）はどうなったら？

負け（ゲームオーバー）は？

3 コアループ（遊びのくり返し）

① 操作する → ② 画面が返してくる → ③ 数や状態が変わる → ④ またやりたくなる理由

4 面白さを一文で

5 作る範囲

作るもの

作らないもの（欲張らない）

考えてみよう！

1. 「④またやりたくなる理由」が書けないとき、遊びのどこが弱いと思う？
2. 終わり方が決まると、作るものはどう変わる？
3. 一文で面白さを言えないとき、芯がぶれているのはどこだろう？

共通のお題

ターゲットはゴールへ移動する。プレイヤーはアイテムで妨害する。5分守ればクリア。

お題は同じでも、「遊びの芯」の決め方でゲームは変わります。

自分のお題（先生用）

ヒント

- 世界観より先に「動詞」を決めると、遊びが決まる
- 勝ち負けが決まると、必要なものが見えてくる
- コアループは「短く回る」ほど強い
- 作らないものを決めるのも設計のうち

このあと

決めた芯を持って、ワークシート2「企画・世界観」へ進みます。世界観は、芯に合うものを選びます。

ワークシート 2

ゲーム企画・世界観

このシートの目的： 自分のゲームを具体化する（オブジェクト洗い出しの準備）

名前 _____

タイトル

ゲーム概要

世界観

ターゲット（何がゴールへ向かう？）

ゴール（どこへ向かう？）

妨害アイテム（何で邪魔をする？）

プレイヤー（あなたは何者？）

ゲームイメージ

自分のゲームの画面を、自由に大きく描く。

考えてみよう！

1. あなたの世界のターゲットは、なぜゴールへ向かうの？
2. プレイヤーは、なぜそれを止めたいの？
3. 世界観が変わると、ゲームの見た目はどう変わる？（ルールは同じまま）

共通のお題

ターゲットはスタートからゴールへ移動する。プレイヤーはアイテムを配置して妨害する。5分間で到達しなければクリア！

自分のお題（先生用）

置き換えのヒント

ターゲット	ゾンビ、猫、勇者、泥棒
ゴール	街、家、宝箱
妨害アイテム	バリケード、罠、壁
プレイヤー	生存者、番人、村人

ルールはみんな同じ。変えていいのは世界観だけ！

ワークシート 3

ゲーム画面・登場オブジェクト

このシートの目的： three.js のシーンに入る「もの」を理解する名前：

オブジェクトの一覧

ワークシート2で描いた絵の中から、「もの」を全部抜き出して書く。three.js では、画面に出る「もの」はメッシュ（Mesh）としてシーンに入れます。

オブジェクト	役割
ターゲット	スタートからゴールへ移動するもの
ゴール	ターゲットが目指す場所
妨害アイテム	プレイヤーが置く、ジャマをするもの

シーン配置図



スタート、ゴール、ターゲット、妨害アイテムを置いて、フィールドを上から見た図を描く。カメラの位置にも印を付ける

考えてみよう！

1. 絵の中に「もの」はいくつあった？
2. メッシュにしにくいものはあった？（光や音はどうする？）
3. カメラは、どこに置くと見やすくなる？

共通のお題

ターゲットはスタートからゴールまで移動する。プレイヤーはアイテムを配置して、ターゲットの移動を妨害する。5分間で目的地に到達しなければクリア！

自分のお題（先生用）

数え方のヒント

- 画面に出る「もの」はメッシュ
- カメラとライトも、シーンに要る「もの」
- 画面に出ていないものは、まだ数えない。

ワークシート 4

ゲームオブジェクトを考える

このシートの目的： ワークシート3で数えた「もの」を、1つずつ深く考える　名前 _____

オブジェクト名　　役割

見た目　　動き

状態が変わるとき（例： 速くなる／止まる）　　持つデータ（数： 速さ／体力／個数など）

他のオブジェクトとの関係　　当たったとき／出てくるとき／消えるとき

オブジェクト名　　役割

見た目　　動き

状態が変わるとき（例： 速くなる／止まる）　　持つデータ（数： 速さ／体力／個数など）

他のオブジェクトとの関係　　当たったとき／出てくるとき／消えるとき

裏 裏方オブジェクト（画面に出ていない「もの」）

タイマー（時間をはかる）

スコアや残り時間の管理

出現の管理（何を・いつ・どこに出す）

考えてみよう！

1. 画面に出ていないのに、必要な「もの」はあった？
2. 同じ部品で作れそうなオブジェクトはどれとどれ？
3. 「関係」を線で結ぶと、ワークシート4のイベントが見えてくる。どの関係が一番大事？

使い方

ワークシート3の一覧から、大事なオブジェクトを2つ選んで深く書きます。残りは、同じ形式でノートに続けても大丈夫です。

ヒント

- 「状態」は、今どうなっているか（止まっている／動いている／壊れた）
- 「データ」は、数で持てるもの（速さ、体力、残り時間）
- 裏方も立派なオブジェクト。タイマーは目に見えないけれど、ゲームを動かしている

このあと

ここで書いた「動き」「状態」「データ」が、次のワークシート5で用意する役割の決め手になります。

ワークシート 5

役割設計

このシートの目的： three.js にはコンポーネントが無い。オブジェクトごとに「役割」を用意することを理解する名前：

オブジェクトごとに、要る役割にチェックを入れてみる。役割は、プログラムで自分で用意します。

ターゲット

☐メッシュ（見た目）☐移動の処理☐当たり判定☐データ（速さ）

ゴール

☐メッシュ（見た目）☐当たり判定☐データ（場所）

妨害アイテム

☐メッシュ☐当たり判定☐効果の処理

画面まわり

☐カメラ（視点）☐ライト（明かり）☐文字（UI）

自分のオブジェクト（ワークシート3・4から）

名前：

☐メッシュ☐移動の処理☐当たり判定☐データ

役割メモ

・メッシュ： 形と見た目
・移動の処理： 位置を変える
・当たり判定： 自分で判定
・データ： 数をプロパティで持つ

付録の使い方

付録の「役割カードシート」を切って、シーンボードに置くと、手で並べながら考えられます。

考えてみよう！

1. ゴールに移動の処理が要らないのはなぜだろう？
2. 当たり判定を用意しなかったら、ゲームはどうなる？
3. エンジンが部品を用意してくれる場合（Unity）と、自分で役割を用意する場合の違いはどこだろう？

ワークシート 6

イベントと処理

このシートの目的：イベント駆動（「〜したら、〜する」）の考え方を理解する名前：

イベントと処理の表

お手本が4つ書いてある。自分のゲームではどうなるか、続きを書いてみる。

イベント（〜したら）	処理（〜する）
キーを押す・クリックする	アイテム設置
アイテムに触れる（当たり判定）	速度低下
ゴールに到達（判定）	ゲームオーバー
5分経過（タイマー）	ゲームクリア

考えてみよう！

1. あなたのゲームで一番大事なイベントはどれ？
2. 「アイテムに触れる」は、誰の仕事が知らせてくれる？（ワークシート5を見直そう）
3. 次に書いたイベントは、次のワークシート7でプログラムになる。どの処理の中に書く？

共通のお題

ターゲットはスタートからゴールまで移動する。プレイヤーはアイテムを配置して、ターゲットの移動を妨害する。5分間で目的地に到達しなければクリア！

自分のお題（先生用）

イベントの見つけ方

- プレイヤーが「操作」したとき（入力）
- もの同士が「当たった」とき（毎フレーム判定する）
- 時間が「経った」とき
- 数が「変わった」とき

「〜したら」と言える場面を探すのがコツ！

ワークシート 7

プログラムとゲームループ

このシートの目的： ゲームループを自分で回すことを理解する名前：

処理の流れを書く

ゲームは初期化のあと、同じ流れを繰り返して動いていく。流れを1つずつ書いてみる。
初期化（init）
入力
更新（update）
当たり判定
データ処理
描画（render）

プログラムを分ける

オブジェクト	プログラム（クラス・関数）
ターゲット	Target クラス
ゴール	Goal クラス
妨害アイテム	Item クラス
ゲーム管理	Game クラス

変数を決める

speed
time
itemCount

ゲームループ（全体）
初期化
↓
入力
↓
更新
↓
当たり判定
↓
データ処理
↓
描画
ループは requestAnimationFrame で自分で回します。描画のあと、ループの先頭へ戻ります。

共通のお題
ターゲットはスタートからゴールまで移動する。プレイヤーはアイテムを配置し、ターゲットの移動を妨害する。ターゲットが、5分以内に目的地に到達しなければゲームクリア！

自分のお題（先生用）

考えてみよう！

1. ループは1秒に約60回回る。speed が大きいと、動きはどう変わる？
2. 描画（render）を書き忘れると、画面はどうなる？
3. 流れの中で、勝敗が決まるのはどの段だろう？

付録

役割カードシート

ワークシート5「役割設計」で使う付録： ハサミで切って、シーンボードに置いて使う名前：
点線にそって切り、オブジェクトの横に置いて「どの役割が要るか」を決めます。カードの下は、その役割で決める値。

※ three.js にはコンポーネントが無いので、カードは「エンジンが用意した部品」ではなく「自分で用意する役割」です。位置（position）・回転・大きさは Object3D が持つ基本の値なので、カードには入れていません。

見た目
メッシュ（Mesh）
形と見た目（ジオメトリ＋マテリアル）
決める値： 形・色・大きさ

見た目
マテリアル（Material）
色・質感
決める値： 色・光り方

視点
カメラ（Camera）
見る位置と向き
決める値： 位置・向き・画角

明かり
ライト（Light）
場面を照らす
決める値： 種類・強さ・色

動き
移動（update）
毎フレーム位置を変える
決める値： 速さ・向き・止まる条件

判定
当たり判定
距離や箱で判定する
決める値： 判定の大きさ

入力
入力
キー・マウス・タッチ
決める値： どの操作で何をするか

処理
処理（関数）
ルールを書く
決める値： 何をするか

処理
タイマー
時間をはかる
決める値： 制限時間・数え方

処理
出現管理
ものを出す
決める値： 何を・いつ・どこに出す

音
音（Audio）
Web Audio で鳴らす
決める値： 音の種類・鳴る場面

データ
データ（プロパティ）
数を持たせる
決める値： 数の名前・最初の値

先生メモ： Unity版と並べて見ると、「用意されている部品」と「自分で用意する役割」の違いが見えます。当たり判定を自前にするところが、three.js版の山場です。

付録

シーンボード

役割カードとオブジェクトを置く台紙 ： ワークシート3・5で使う　　名前 _____

タイマー： 5:00　　スコア：　　アイテム残り：

ゴール

スタート

置いた部品のメモ（どのオブジェクトに、どのカードを置いたか）

置いてみて気づいたこと