

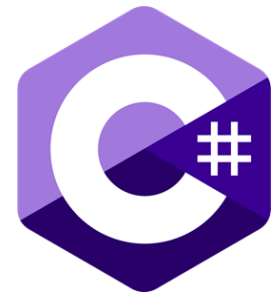
放物運動を用いたUnityのゲーム開発をしてみよう

数物情報科学科3年 Noir・可燃、・ゆきしまむ・ザベスちゃん・つかち

1.目的

ゲーム開発ツールUnityを用いてストラックアウトゲームを作り、
その中で以下2点について学ぶ

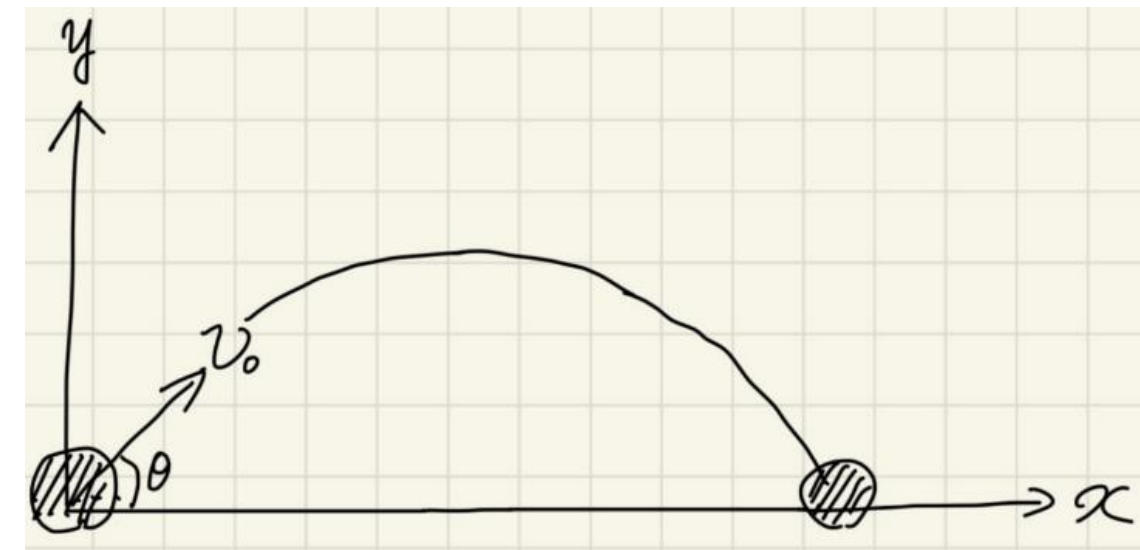
- 1.放物運動をする物体の振る舞いの理解とシミュレーションの実装方法
2. ゲーム開発におけるノウハウの習得とシステム構築
上記を通してシステム開発の楽しさに触れることを目的とする。



2.背景

■ 放物運動について

x を地面と平行の方向成分、
 y を地面と垂直方向の成分として
 m の物体を初速度 v_0 で投げた。
運動方程式の解は以下の通りになる。



$$m \frac{d^2x}{dt^2} = 0 \quad m \frac{d^2y}{dt^2} = -mg \quad (1)$$

$$\vec{v} = (v_0 \cos \theta, v_0 \sin \theta - gt) \quad \vec{r} = \left(v_0 \cos \theta t, v_0 \sin \theta t - \frac{1}{2}gt^2 \right) \quad (2)$$

$y = 0$ の時、地面に物体が到着する時間で解は $t = 0, \frac{2v_0^2 \sin \theta}{g}$ である。

$t = \frac{2v_0^2 \sin \theta}{g}$ を式(2)の x 成分に代入すると $x = \frac{v_0^2 \sin 2\theta}{g}$ である。

$$\theta = 45^\circ \text{ の時 } x \text{ が最大なので、 } l_{\max} = \frac{v_0^2}{g} \quad (3)$$

▷ 放物運動かどうかの判断方法

重力加速度 g はUnityの物理エンジンでも9.81m/s²である。

今回は放物運動の正確性に関する指標として、(3)式を変形した、

$g = \frac{v_0^2}{l_{\max}} = 9.81$ の式をもとにUnityで放物運動のモデルを作成した。

■ Unityについて

UnityはUnity Technology社が開発している世界的に人気なゲーム制作ツールである。

主な特徴として以下の3点があげられる。

- ・物理エンジンをゲーム内で使用できる
- ・「C#」というプログラミング言語を学べる
- ・創造力や論理的思考力が養われる。

3.開発 I

■ 開発の手順



▷上記の段階に沿って開発をおこなった。その他に

- ・週1回の定例会を実施。
- ・開発した内容をドキュメントにまとめて全体に共有する作業をした。

■ C#上での放物運動の実装

▷ 放物運動をC#で実装してみる

C#で初速度を表現するために

`rb.AddForce(dir * rb.mass, ForceMode.Impulse);`を使用した。

上記のコードは静止している物体に瞬間的な力を与えた

衝突の運動方程式を利用している。静止した物体に力を与えると衝突の式は

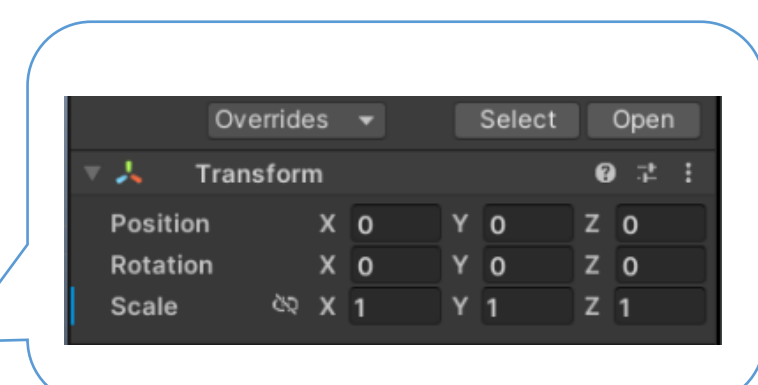
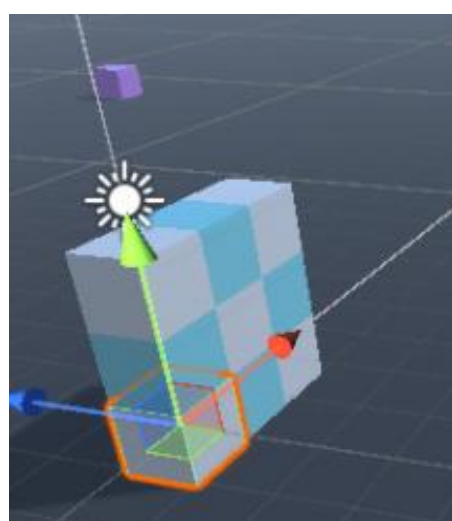
$$mv_0 - 0 = \int F dt \quad (4)$$

(4)式を変形すると、

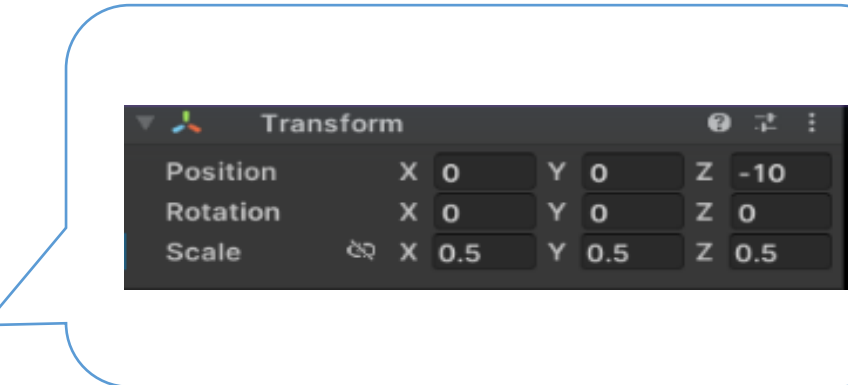
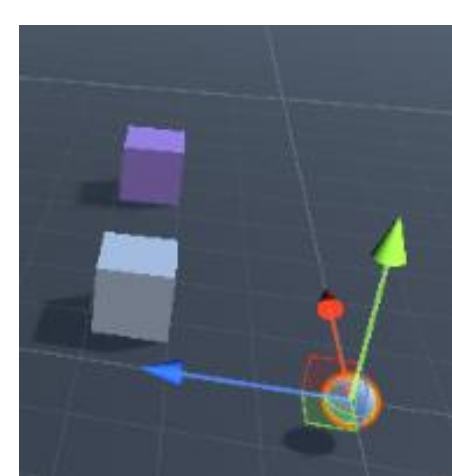
$$v_0 = \frac{F}{m} * \Delta t$$

今回、Force.Impulseでは Δt の演算を無視して力を加える時間を無視して
 v_0 を表現した。結果、放物運動における初速度を表現することができた。

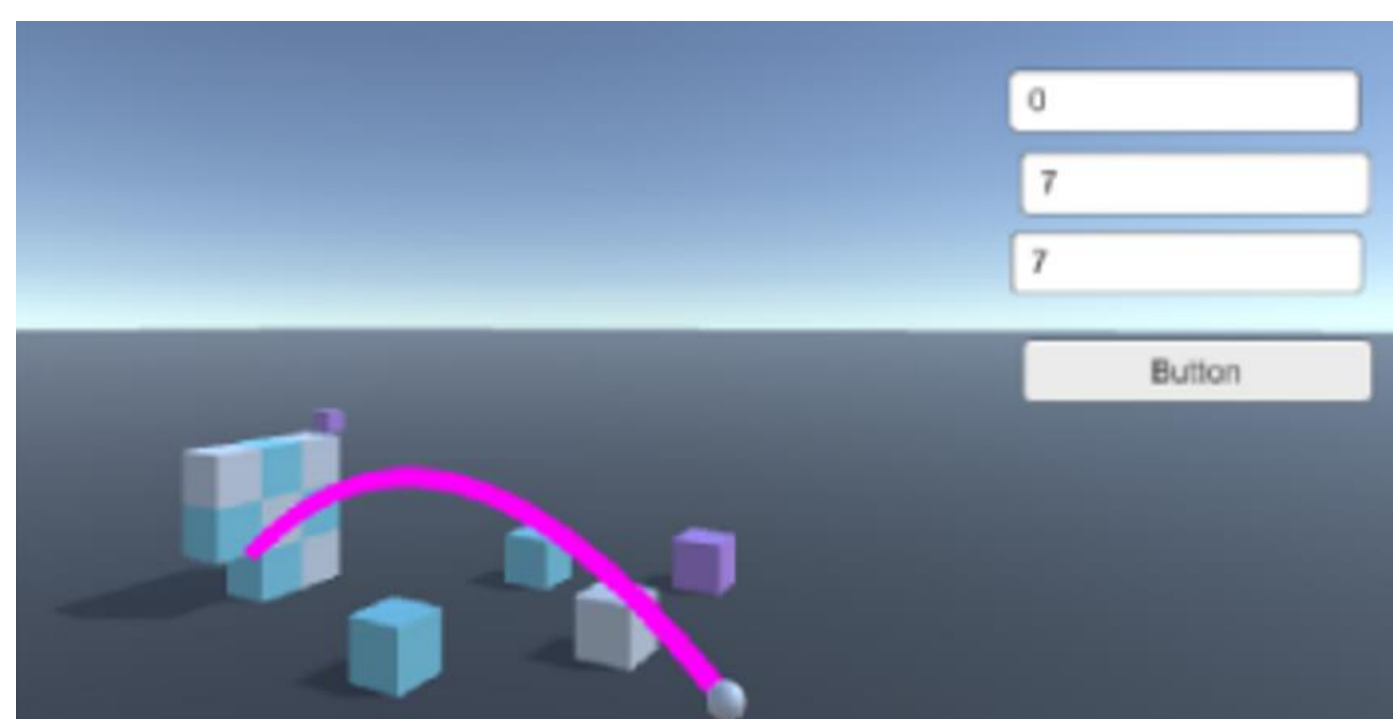
▷ 本当に放物運動ができているか検証してみる



的座標 (0, 0, 0)



ボールの座標 (0, 0, -10)



▷ ゲーム上での検証

当たった物体が消滅する仕様にした。

距離10m、 $V_0^2 = 0^2 + 7^2 + 7^2 = 98$

に物体が当たるので、(3)式から

$$g = V_0^2 / X_{\max}$$

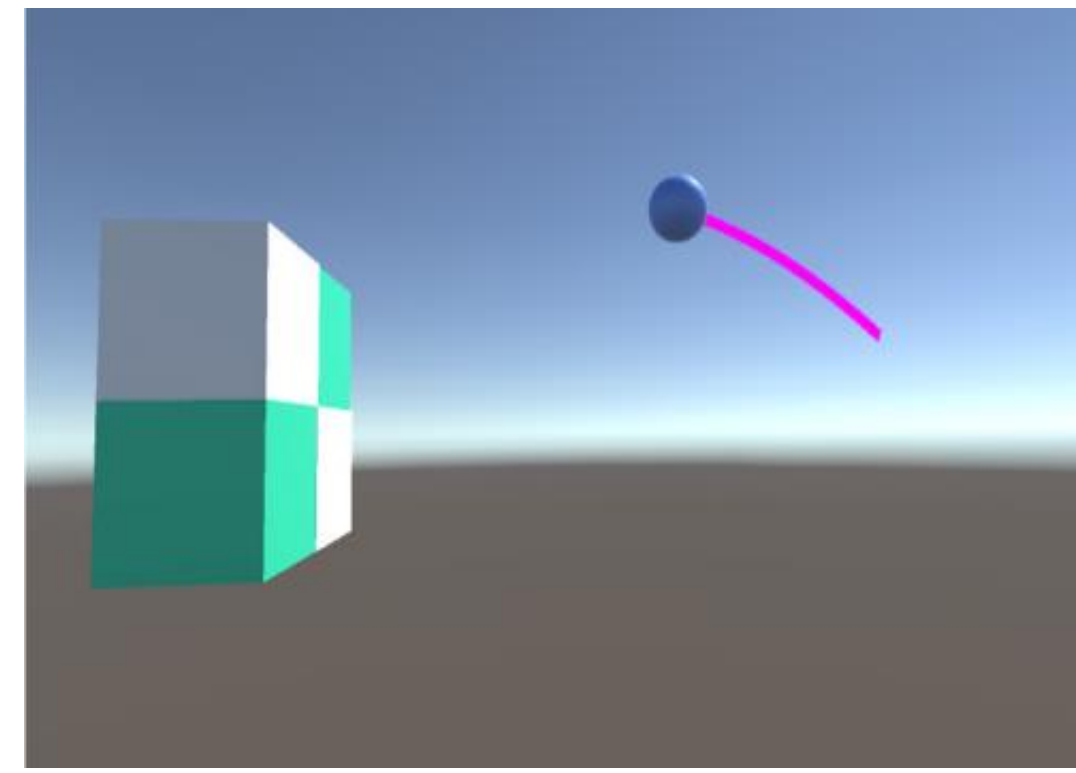
$$= 98 / 10 = 9.8 \text{ となる。}$$

g が正しい値になったので

放物運動ができているといえる。

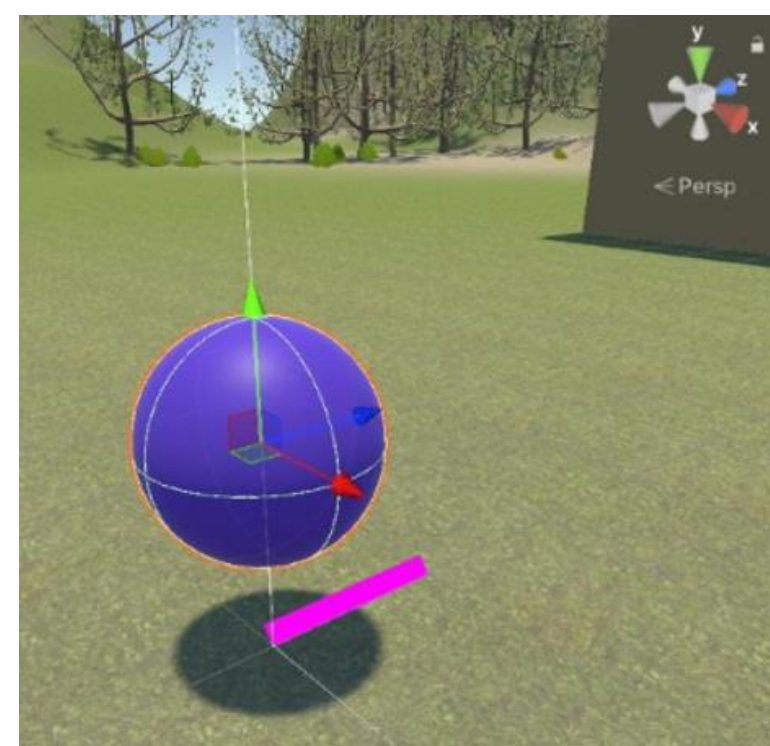
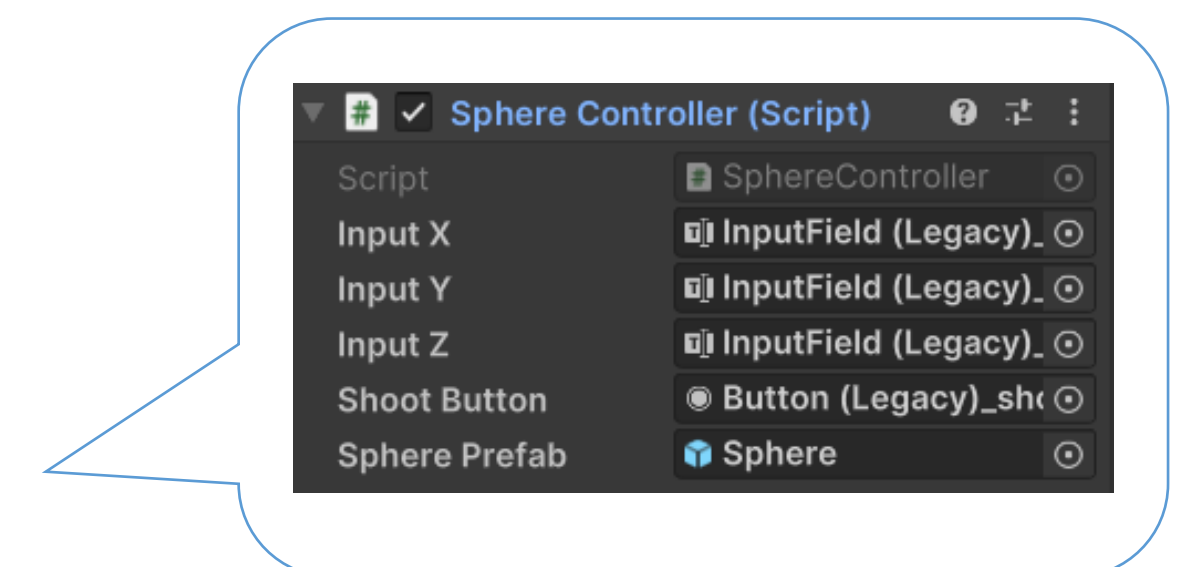
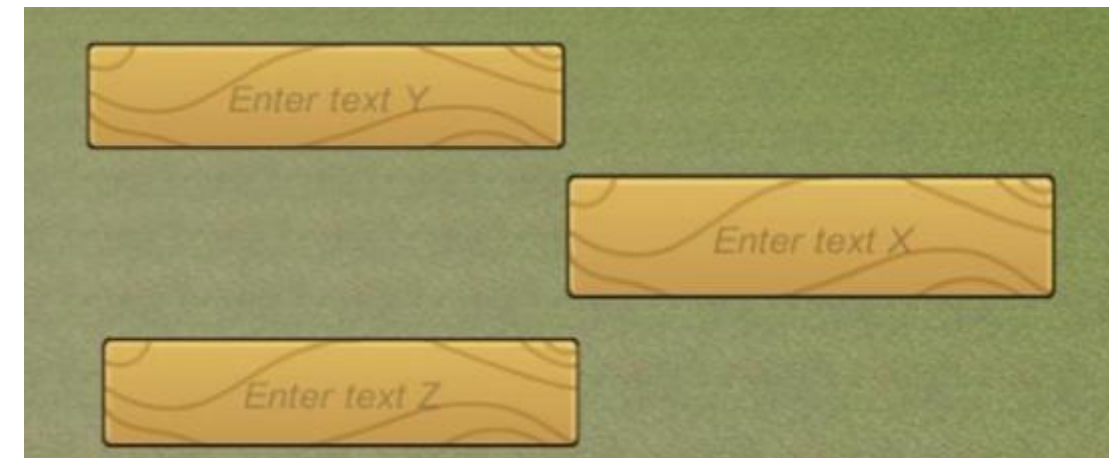
4.開発 II

■ C#上での放物軌道の実装



▷C#で軌道を表現するために
LineRendererを使用した。
実装結果は右図の
ようになり、軌道が表示されるように。

■ 入力・出力UIの実装



Input x : Unity上のx軸方向の初速度入力
Input y : Unity上のy軸方向の初速度入力
Input z : Unity上のz軸方向の初速度入力

▷初速度をユーザー側が入力した値が
ゲーム内で反映できるシステムを
Unity側とc#側で作成。

■ カメラの視点切り替え



正面から見たゲーム画面



真上から見たゲーム画面

▷入力した初速度に合わせてボールを発射できるようにした。
カメラをUnity上で複数台設置し、カメラの切り替えの実装をした。

■ ゲームクリア画面



GameClear画面切り替え



GameOver画面切り替え

▷ターゲットを全て消した時に「Gameclear」のUIを表示させた。
制限時間を超過した場合、「GameOver」のUIが表示させた。

5.感想

・Noir：見かけで判断するのではなく、理論的に事象を捉える力が物理モデルを作る上で初歩的な心構えであることを学びました。

・つかち：自分たちの生活している世界はモノを動かすのも視点を動かすのも簡単に行えるが、PC上でシミュレーションとして動かすのは難しかった。その反動で実験が楽しかった！

・ザベスちゃん：ゲームを作るのは本当にたくさんの工程が必要で制作の大変さを実感しましたが、共同作業は楽しかったです。

・ゆきしまむ：チーム開発をやってとても良い経験になりました。実際に放物運動をゲームとして落としこむ難しさやゲーム作りの大変さを実感しました。次こそUnityの共同作業ができるようにしたいです。

・可燃、：こんなに簡単なゲームを作るのに沢山の設定やコードが必要なことに驚かされました。普段私がプレイしているゲームを以前よりありがたく感じる事ができました。

6.参考文献

参考書：Unityの教科書 Unity 2022完全対応版

開発工程資料：

https://hackmd.io/@IBQURunJTziQ_jGTP6rCHg/ゲーム仕様の実装/edit

<https://hackmd.io/@Noiriron/放物運動の評価>

<https://hackmd.io/@kanengomi/ry2pVky1x/edit>

<https://hackmd.io/i2StCz7qSwm78S277eS2CA?view>

https://hackmd.io/e_dOz5cdSCCe_gIqxjo_jg?view