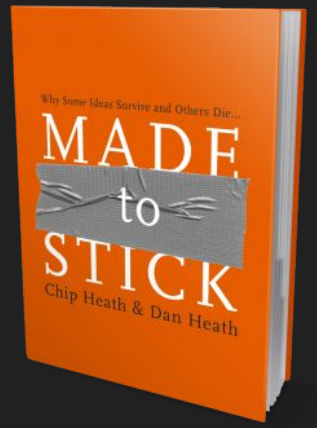
Rigidbodyを使おう

今回のプレゼンは・・・

ユニティ道場の動画「プロが教える脱初心者スクリプト術！」が元になっています。

ユニティテクノロジーズジャパンフィールドエンジニア、安原祐二さんの講演をお勧めしたいという事と自分の理解をプレゼンをする事で深めたいという2つの想いがあります。



1.Scriptを作ろう

「シンプルにキャラクタを動かす」

プリミティブからCubeを作成

SimplePlayer.csを作成

Input.GetAxisを使って上下左右の入力を取得しよう

加速度っぽい動きを除く

Time.deltatime
ゲームとは細かい処理の繰り返し

一つの処理を1フレームと呼び、1秒間に60回（フレーム）処理が行われる。

Δt をプログラムに組み込む

一般的にゲームは1秒間に60回画面を書き換える。モニタがそういう仕様だからそれに合わせて描き換える。

下記より引用

http://d.hatena.ne.jp/Nagise/20090824/1251117275

Rigidbodyを使おう！

リジッドボディを和訳すると剛体といい、変形することのない物体を表す。

2.振る舞いをつける

Cubeにコンポーネント「Rigidbody」を追加

グラビティをOFFにする

deltaTimeは使わない

force = new Vector3(hori, vert, 0f);

GetCompoinent<Rigidbody>().AddForce(force);

※hori, vertはver型でInput.GetAxisから型を取得

今回の重要テーマは3つ。

・Drag

・バネ

・回転

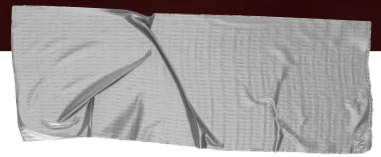
「Drag」とは動かしにくさ。

速度に比例した力が逆方向に働く。

一定の外力を加えるとある速度に収束する。

終端速度という。

```
void FixedUpdate()
{
        var hori = Input.GetAxis(“Horizontal”);
        var kmph = 100f;
        var mps = kmph * (1000f / 3600f);
        var target_velocity = mps * hori;
        var v = new Vector3(target_velocity, 0f, 0f);
        var rb = GetComponent<Rightbody>();
        var force = (rb.mass * rb.drag * v) /
                        (1f - rb.drag * Time.fixedDeltaTime);
        rb.AddForce(force);
        Debug.LogFormat(“velocity = {0} km/h”,
rb.velocity.magnitude*(3600f / 1000f);
}
```



終端速度

$$v' = \frac{F_0}{mk}(1 - k\Delta t)$$

終端速度を得る為の外力

m:質量

k:ドラッグ

Fo:外力

v':終端速度

$$F_0 = \frac{mkv'}{1 - k\Delta t}$$

Dragの値を変化させるとゲーム調整に使えそうだな。

オーバーシュートしないバネ係数

r：バネ係数
m：質量
k：ドラッグ

$$r < \frac{mk^2}{4}$$

「バネ」って

支点から近ければ弱く、遠ければ強くなる。

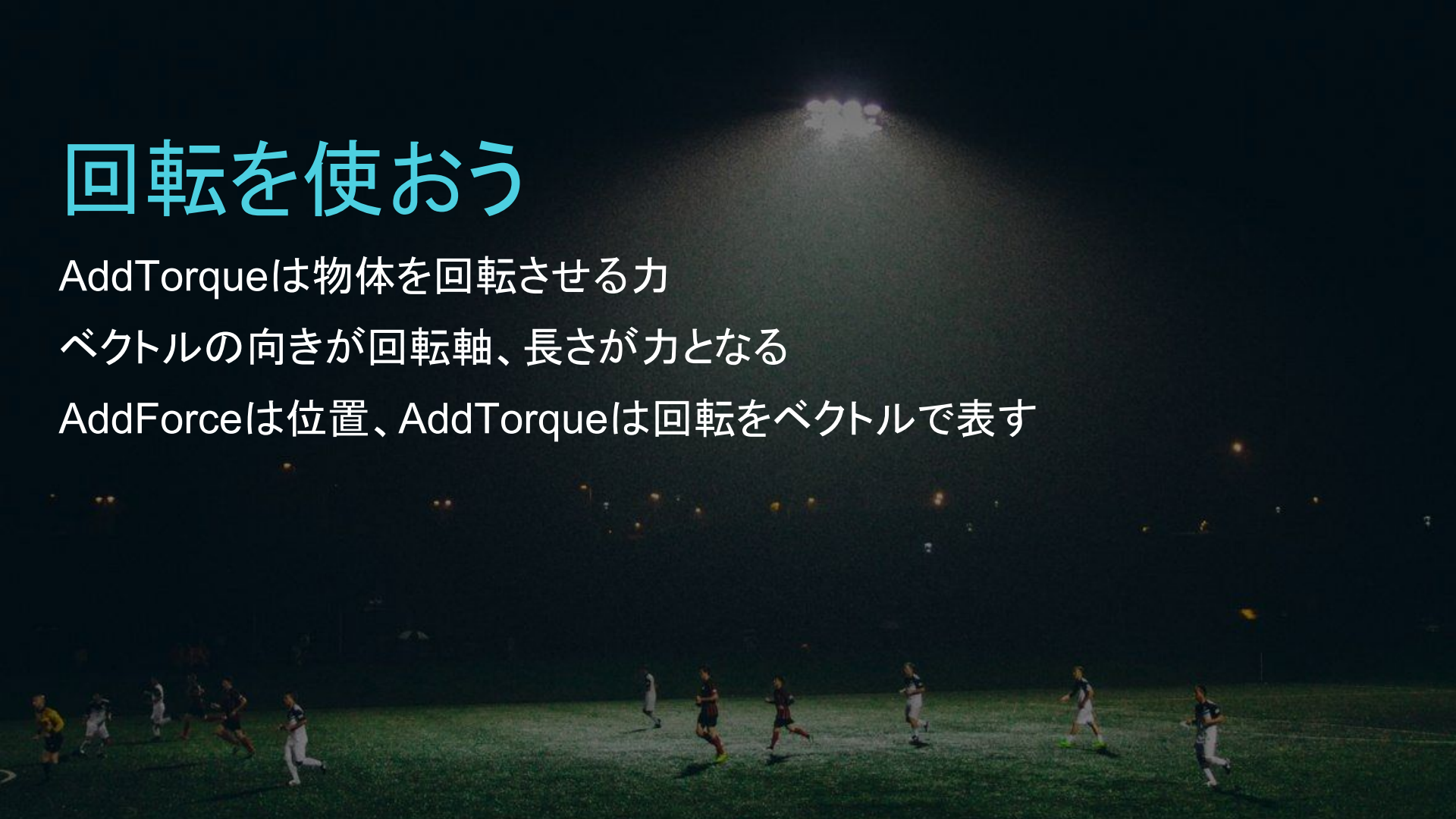
```
void AddSpringForce(Vector3 target_position, float r)

{

        var diff = target_position − transform.position;

        var force diff * r;

        rigidbody_.AddForce(force);

}
```

回転を使おう

AddTorqueは物体を回転させる力

ベクトルの向きが回転軸、長さが力となる

AddForceは位置、AddTorqueは回転をベクトルで表す



飛行機の操作を作ろう

AnglureDragを強くする

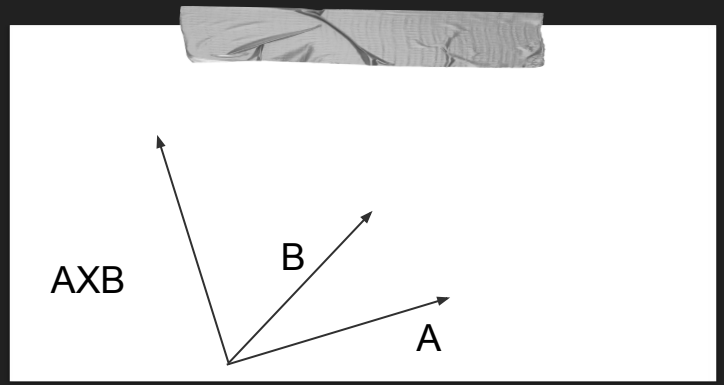
外積を使い、姿勢を水平に保つ

AddForceを与えて飛行機を進ませよう

ロールしてしまう問題に外積を使って対応する

現在の向きがAだとして、B方向に向きたいときのトルクは？

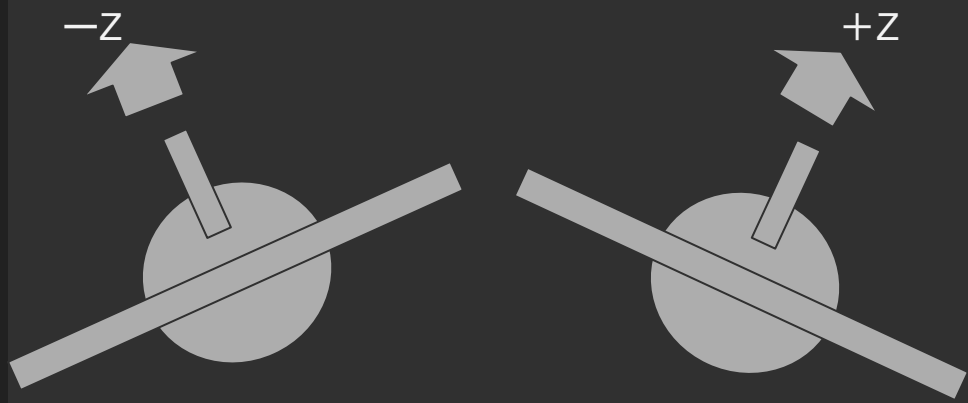
外積AXB（AとBの面に垂直、長さが平行四辺形の面積）で対応できる



飛行機なので機体を傾けたい！

Z軸に水平入力(Horizontal)を与える

あとはAddForceを使って機種の方向に
力を発生させればラジコン飛行機の様に
進む



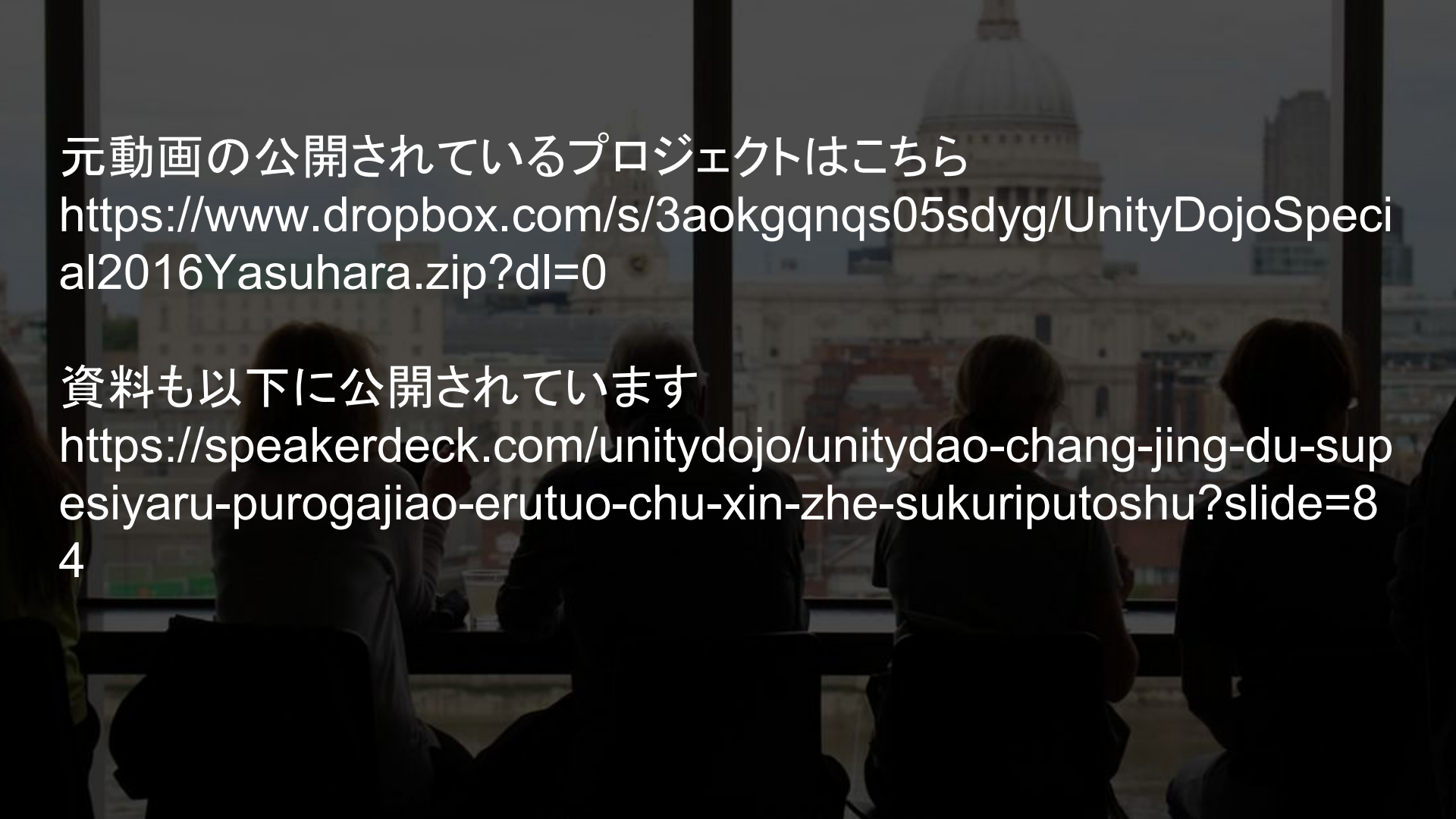
まとめ

物理エンジンを活用するには物理を学ぶのが良いと思います。

ゲームの世界を作るのに必要だと思えば楽しめて学ぶ事が出来そうな気がする。

終端速度を得るための外力

$$F_0 = \frac{mkv'}{1 - k\Delta t}$$



元動画の公開されているプロジェクトはこちら
https://www.dropbox.com/s/3aokgqnqs05sdyg/UnityDojoSpecial2016Yasuhara.zip?dl=0

資料も以下に公開されています
https://speakerdeck.com/unitydojo/unitydao-chang-jing-du-supesiyaru-purogajiao-erutuo-chu-xin-zhe-sukuriputoshu?slide=84

ありがとうございました