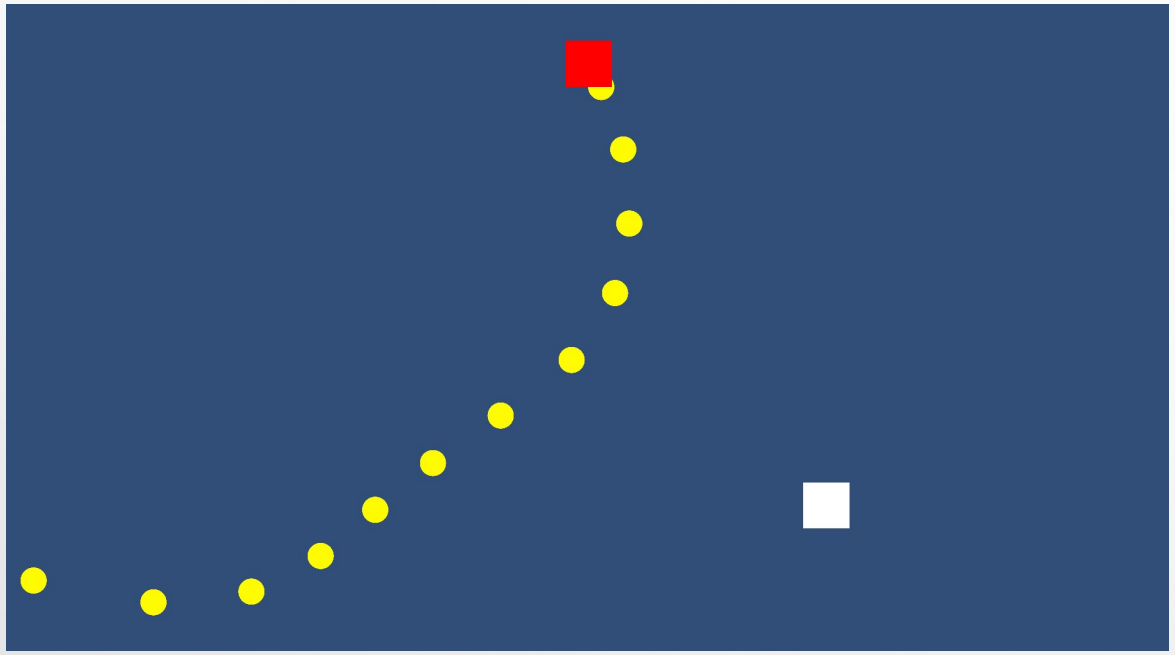
弾の発射と当たり判定

ゲーム数学

目次

１．UnityPackageを使用しての準備

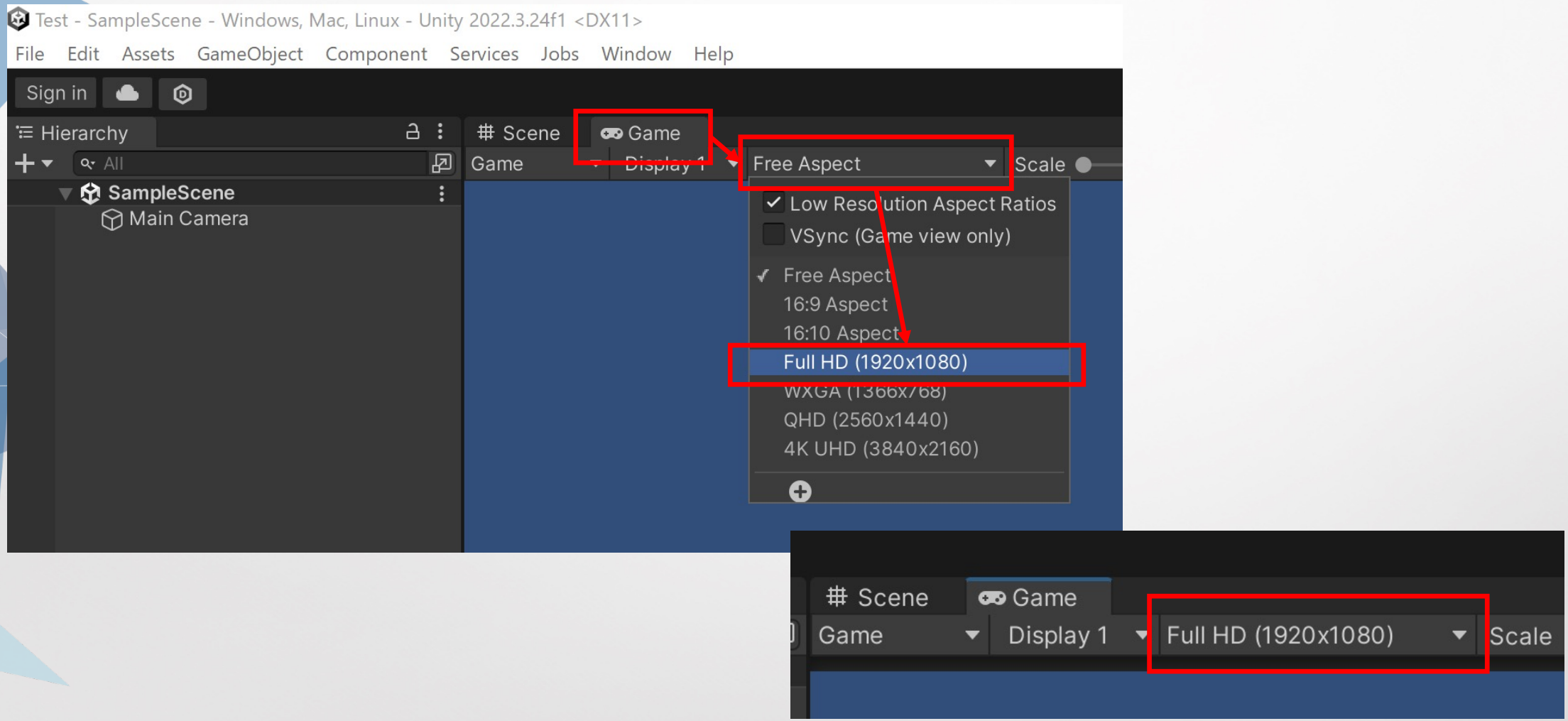
シューティングゲームで敵からプレイヤーに弾が発射されていますが、
これはどのような計算によって行われているか実際に試してみましょう！



それでは、まずはゲームのプロジェクトの設定をしていきたいので、
UnityHubを開いて2Dでプロジェクトを作成してください。
また、別途unitypackageのファイルを自分のPCにコピーしてください。

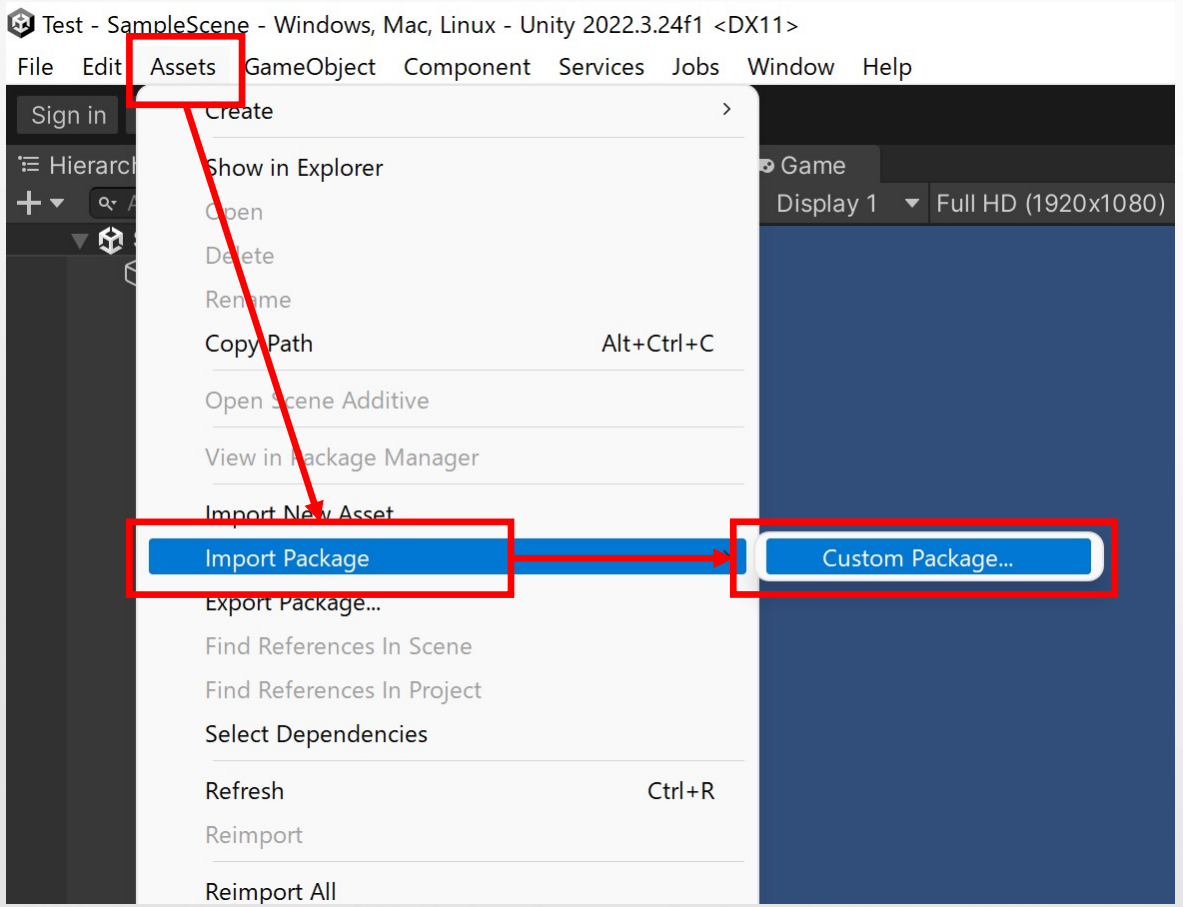
1．UnityPackageを使用しての準備

プロジェクト開いた人は、以下の操作をお願いします。(画面の解像度の設定)



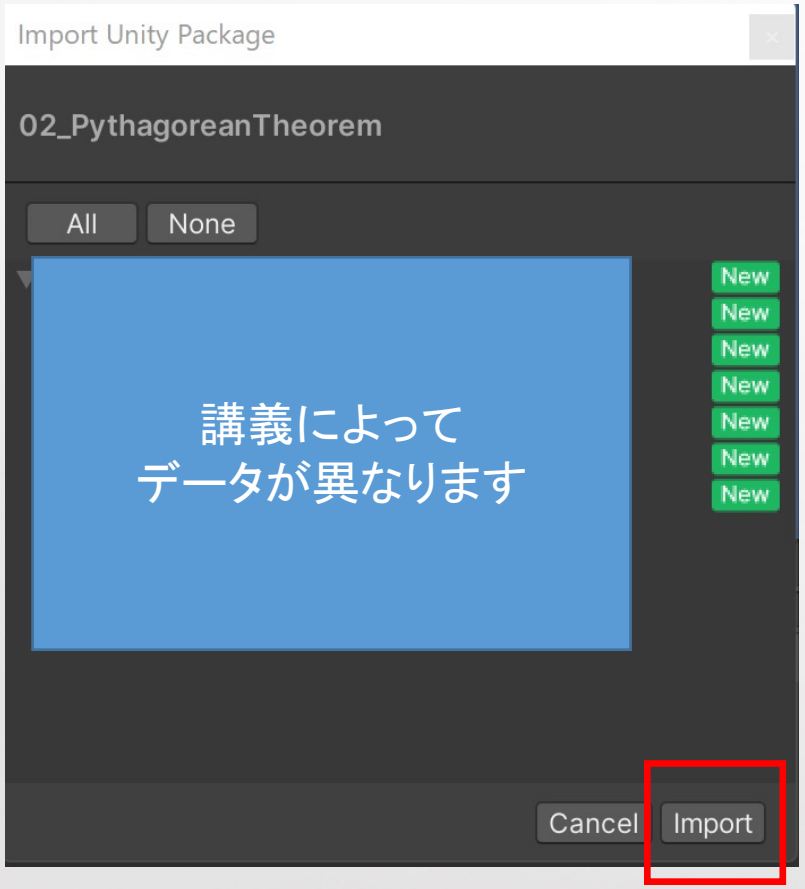
１．UnityPackageを使用しての準備

次に、unitypackageというパッケージファイル(データが詰め込まれたファイル)を読み込みます。



１．UnityPackageを使用しての準備

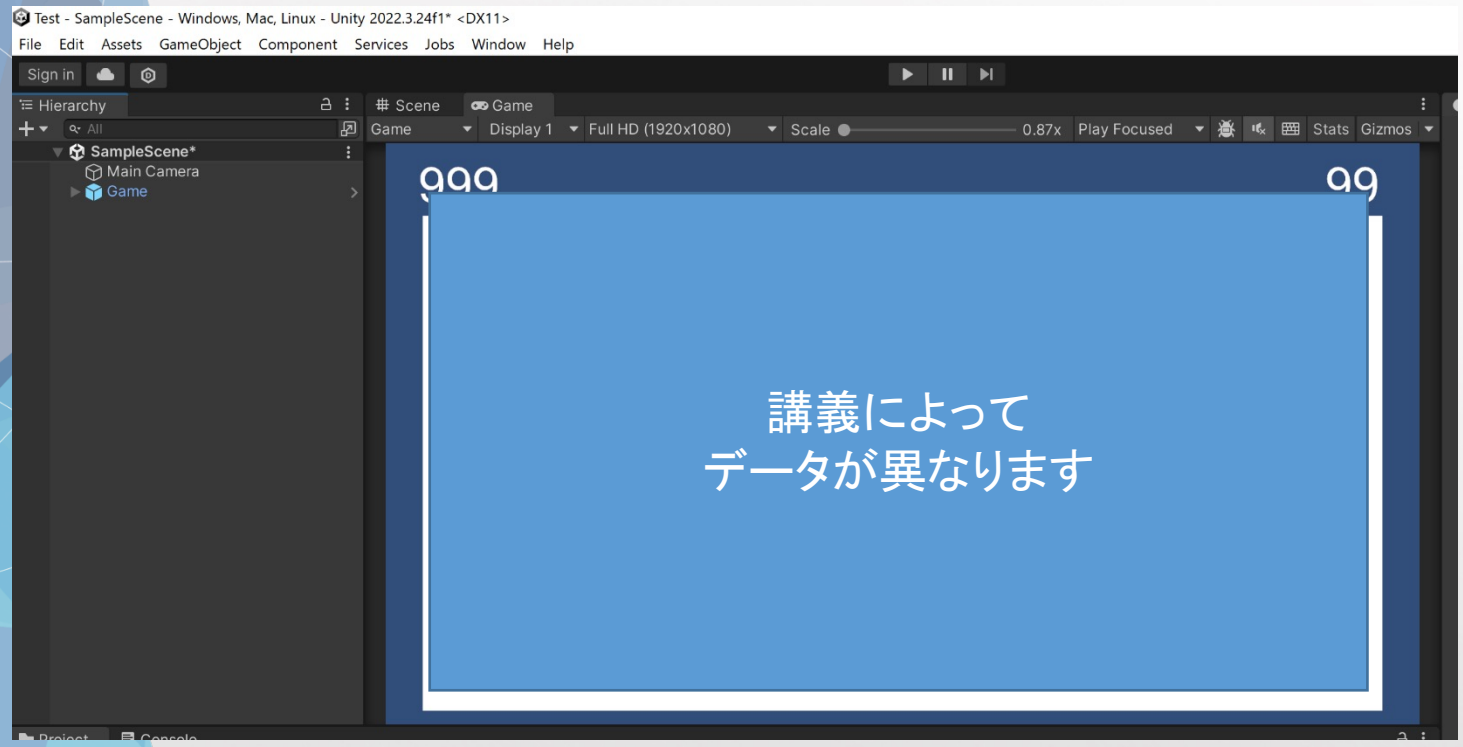
ファイルを指定し、以下の画面が出たらImportを押すと、ファイルが展開されます。

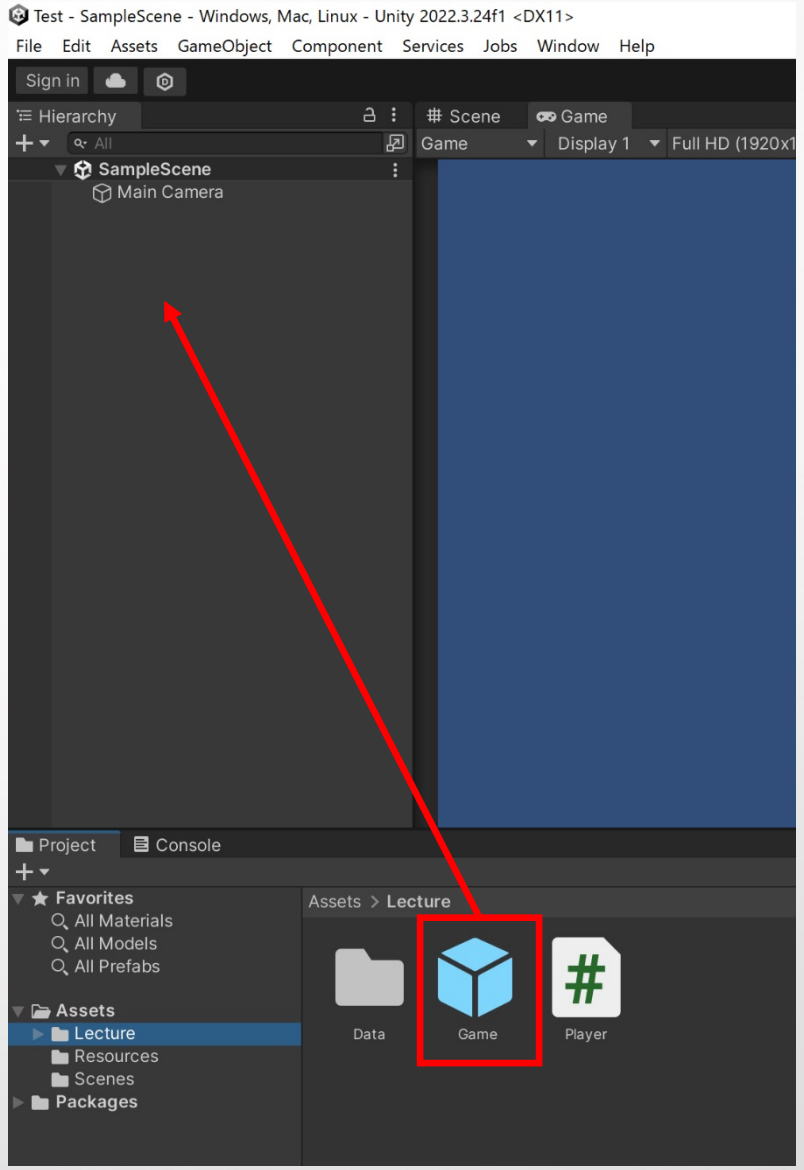


１．UnityPackageを使用しての準備

「Game」のファイルを、ヒエラルキー上に
ドラッグしてみてください。(右記参照)

以下のような画面になっていればOKです！





２．弾を飛ばす際の考え方

では、実際に敵(赤)からプレイヤー(白)に弾を飛ばすにはどうしたら良いか考えてみよう！

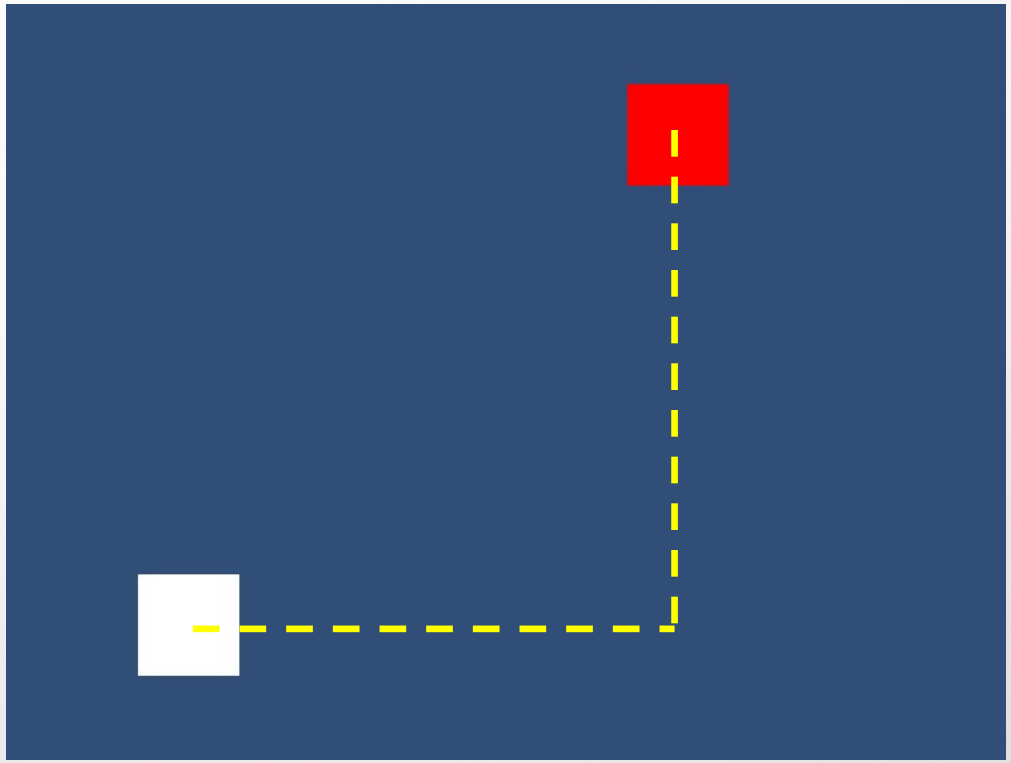
まずは分かっていることを整理！

・プレイヤーのX/Y座標

・敵のX/Y座標

この2つが分かれば、まずはプレイヤーと敵がX/Y座標ともに差分が取れると思います。

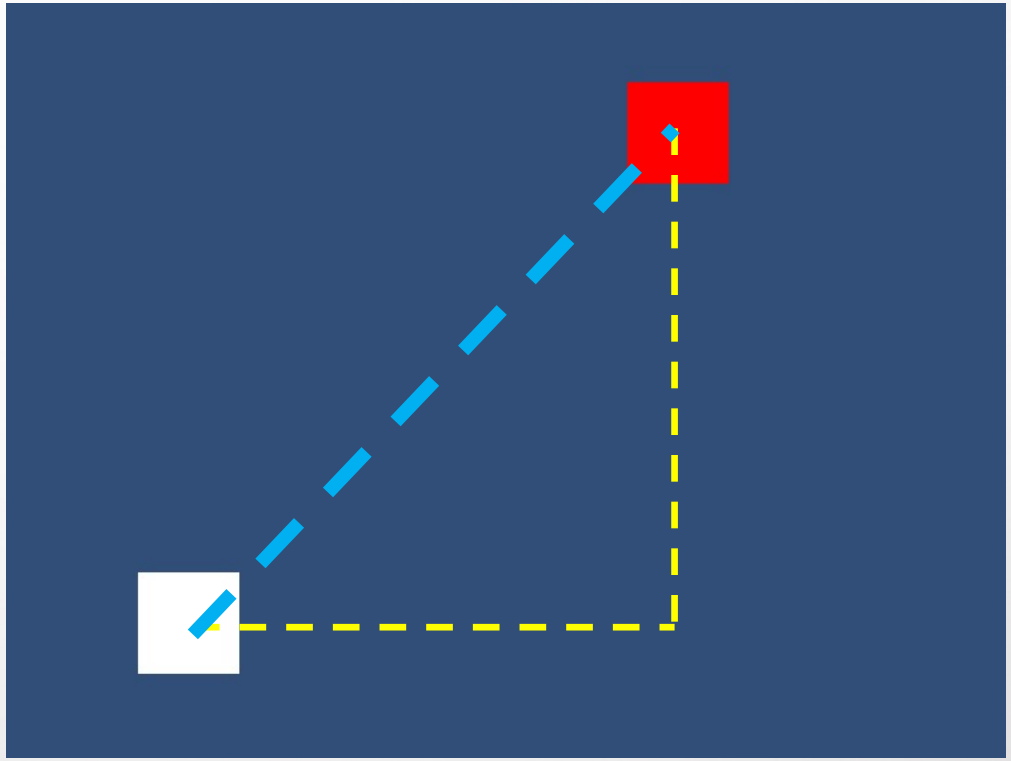
ここからは少し考えてみてください！



２．弾を飛ばす際の考え方

この2点が分かれば、あとは斜辺も分かると思います！

ここまで来たら、あとは移動量を
求めることが出来れば良いかと
思います。
敵からプレイヤーに真っすぐ弾を
飛ばしたいと思うのですが
Xをどれだけ、Yについてもどれだけ
移動させていけば良いでしょうか。



求めた斜辺を使って上手いことできないでしょうか。。

２．弾を飛ばす際の考え方

隣辺、対辺の長さを斜辺で除算するとX/Yの移動量が出てくると思います！

・隣辺 / 斜辺 ＝ Xの移動量　　・対辺 / 斜辺 ＝ Yの移動量

そのため、以下のようなことをやれば上手くできそうです。

１：プレイヤーと敵の斜辺を求める

２：斜辺から移動量を求める

３：移動量に沿って弾を動かす

３は実装済みのため、１と２についてプログラムを作ってみてください！

スクリプト：Bullet.cs(Start関数内)

プレイヤーのX座標：playerObj.transform.position.x

敵のX座標：enemyObj.transform.position.x

(Y座標は.xを.yに変更してください)

２．弾を飛ばす際の考え方

解答例

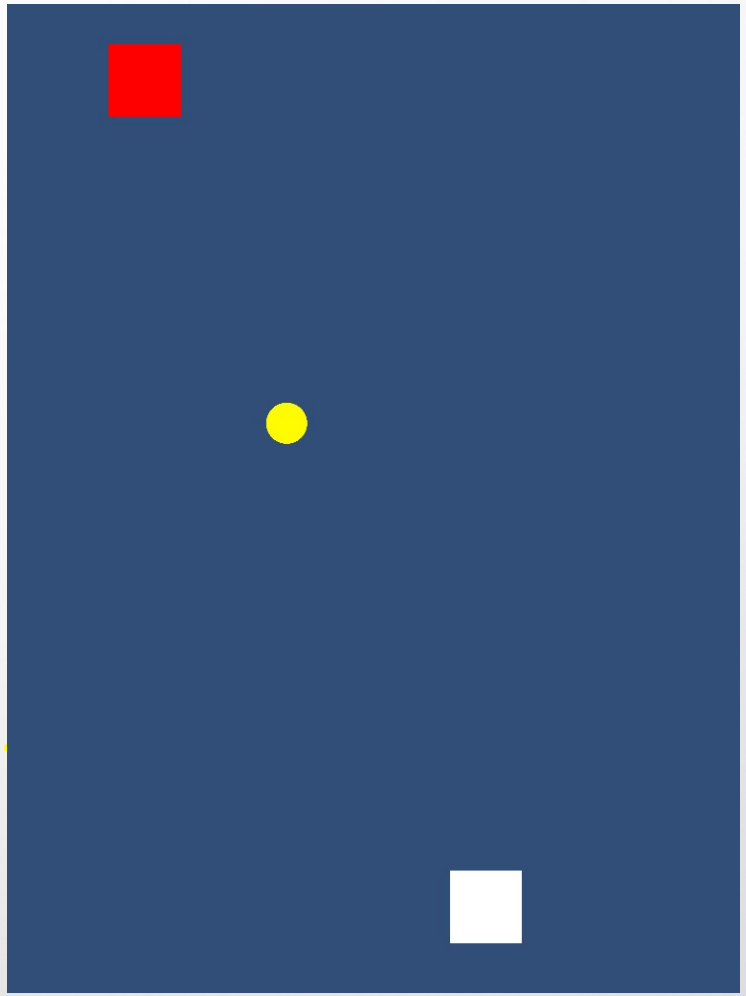
```
//プレイヤーと敵のオブジェクト取得
playerObj = GameObject.Find("Player");
enemyObj = GameObject.Find("Enemy");

//距離の差分を出し、斜辺を求める
float x = playerObj.transform.position.x - enemyObj.transform.position.x;
float y = playerObj.transform.position.y - enemyObj.transform.position.y;
float syahen = Mathf.Sqrt( (x * x) + (y * y) );

//斜辺からXとYの移動量を求める
//X座標の加速度をvec2.xに代入してください
//Y座標の加速度をvec2.yに代入してください
vec2.x = x / syahen;
vec2.y = y / syahen;
```

３．弾との当たり判定の考え方

当たり判定とは、物と物が衝突しているか判定を行うこと。

今回のプロジェクトでは黄色の球と白の
プレイヤーが重複しているかしていないかで
判定を行う。

Unityでは当たり判定をUnity側が行い
コールバック関数が呼ばれるが、今回は
自力で当たり判定を実装する。

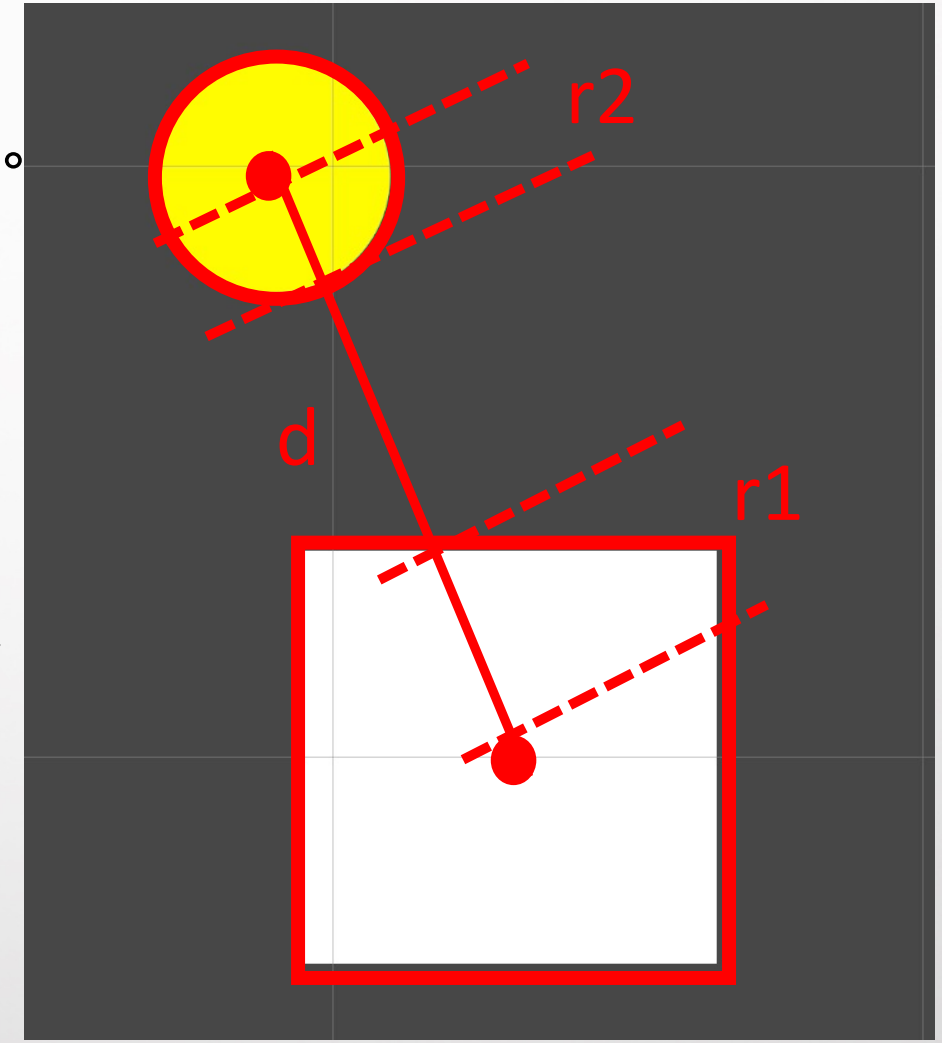


３．弾との当たり判定の考え方

実際にどのような考え方で実装するのかを解説する。

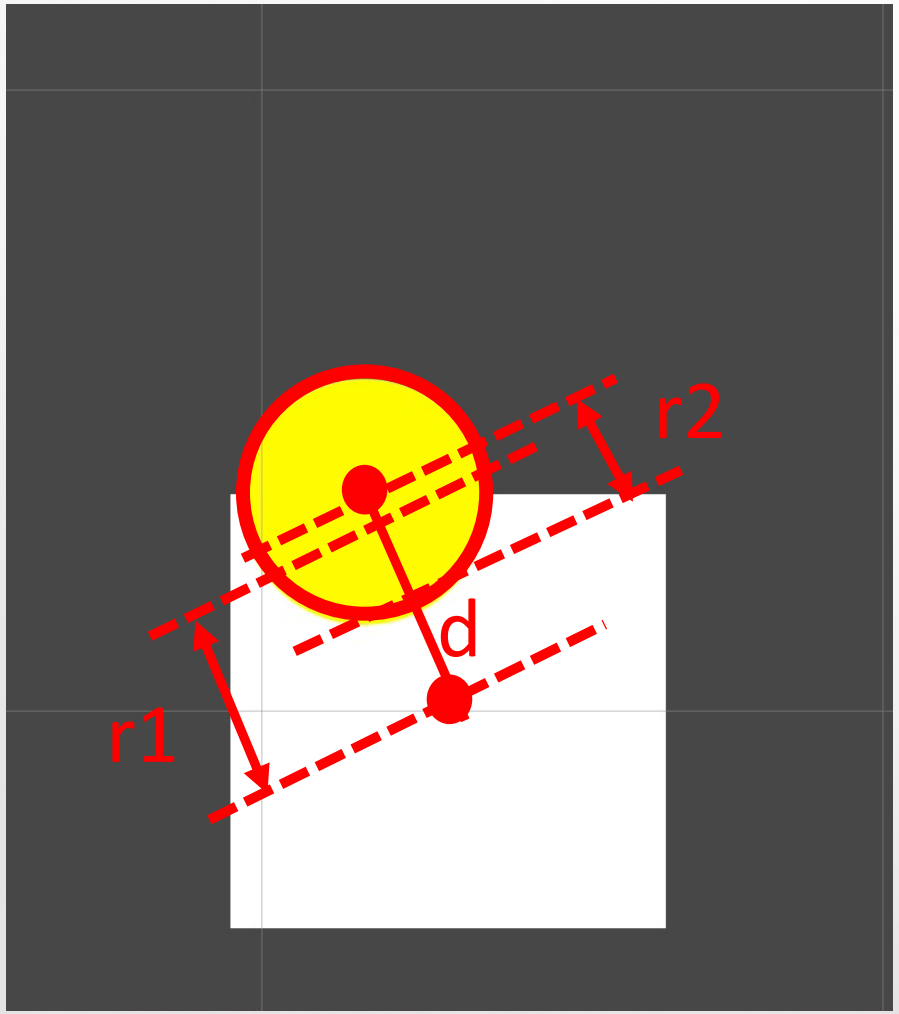
パラメータとしては以下のデータを使用する。
r1：プレイヤーのサイズ(0.35)
r2：弾のサイズ(0.2)
d：プレイヤーと弾の斜辺

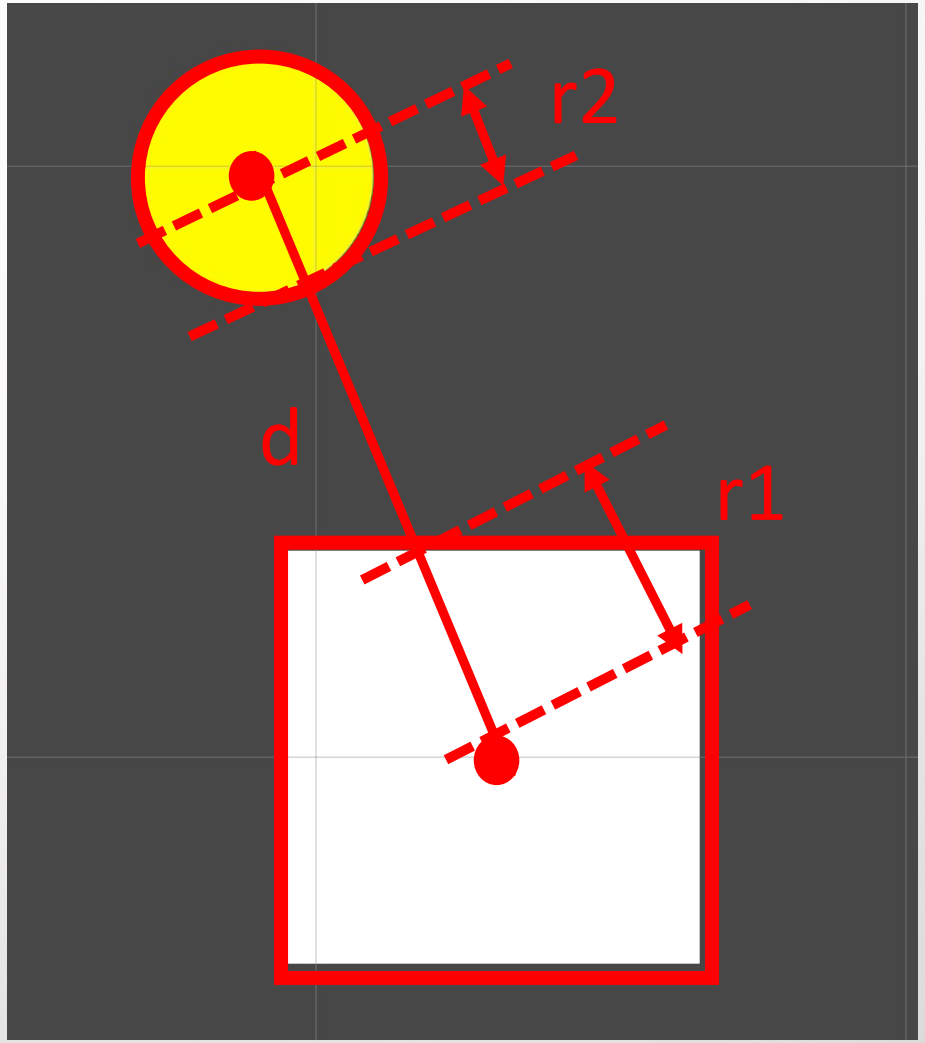
上記のパラメータの関連性から当たり判定を導くことは可能か。。。



３．弾との当たり判定の考え方

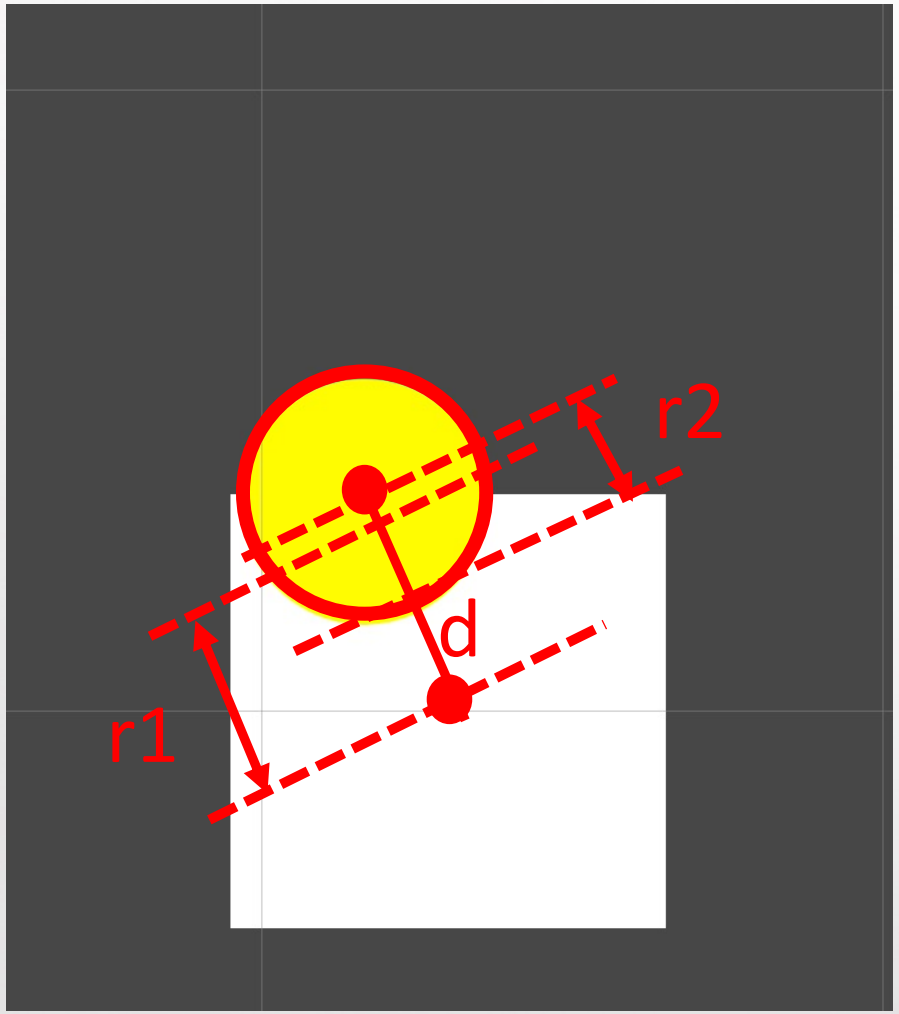
r1、r2、dで当たっている時と当たっていない時、何が違うか。



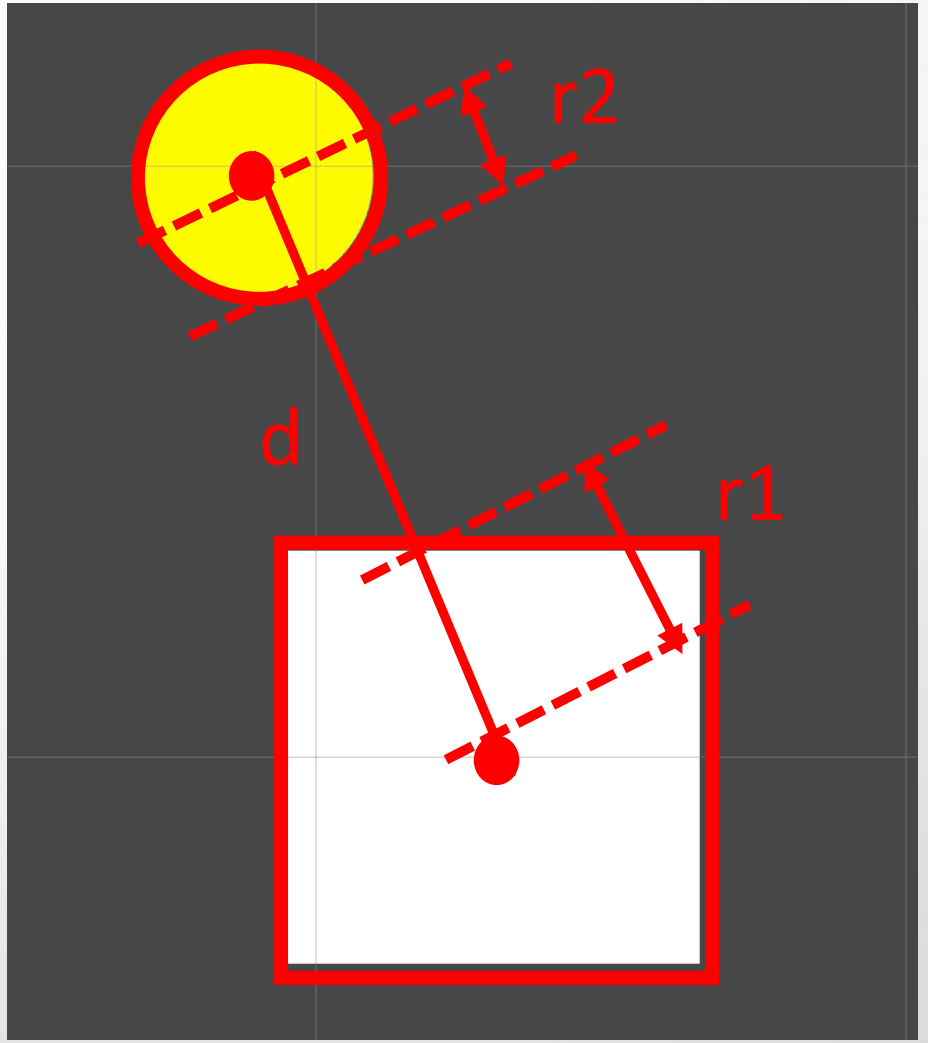


３．弾との当たり判定の考え方

当たっている時：r1 + r2 > d



当たっていない時：r1 + r2 < d



３．弾との当たり判定の考え方

上記の内容を踏まえて、自分でプログラムを考えてください。

スクリプト：Bullet.cs

関数：update内、移動処理の下

処理内容：

①プレイヤーと弾の斜辺を求める
(弾は、transform.position.x(y))

②r1は0.35、r2は0.2f固定とする

③①で求めた斜辺とr1、r2を足したものを比較する。
r1とr2を足したものが大きい場合、弾を消す。
(弾を消すのはDestroy(gameObject)

３．弾との当たり判定の考え方

解答例

```
void Update()
{
    //移動
    rb2.velocity = new Vector2(vec2.x * speedData, vec2.y * speedData);

    //当たり判定
    float x = playerObj.transform.position.x - transform.position.x;
    float y = playerObj.transform.position.y - transform.position.y;
    float d = Mathf.Sqrt( (x * x) + (y * y) );
    float r1 = 0.35f;
    float r2 = 0.2f;

    if (d < r1 + r2)
    {
        Destroy(gameObject);
    }
}
```