

GAME TECHNOLOGY

*** ゲームテクノロジー ***

目次

はじめに	1
ランチャーの起動	2
そもそも UnrealEngine4 って何?	4
ゲーム業界の UE4 ニーズについて	6
C++で UE4 の何が難しいのか	6
じゃあブループリントは簡単かって言うと...	7
というわけで授業の流れ(コマシラバスともいう)	8
はじめての UE4	9
UE4 起動	9
エディタを操作	9
材質を変更しましょう	10
ファイアー!	13
敵(マト)を作ろう	18
当たり判定を追加しよう	18

はじめに

はい、ゲームテクノロジーなどと言うとりますけれども、普通に UE4(Unreal Engine4)…アンリアルエンジン4の授業です。

「基本的に何やってもいいし」とお任せされている授業なのですが、3 年生相手だし、今までやってない内容の話がいろいろだろうという事で、そう考えると他の先生がやってない事…ってなると DirectX12 か UE4 しかないわけです。

とはいえ、DirectX12 なんぞ全員向けにやると死人続出なのは目に見えているので、まあ…UE4 じゃろということで UE4 やります。

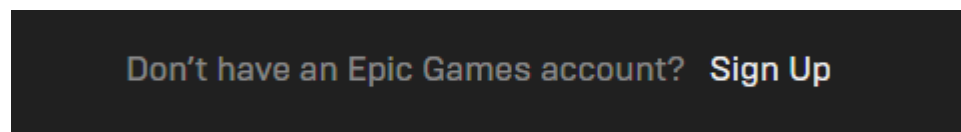
ちなみに皆さんのPCに既にUE4がインストールされてれば話が早いのですが、そうじゃない場合はガチで今からインストールする羽目になります。なお、UE4のランチャー(Epic Launcher)は入ってると思いますので、まずは UnrealEngine で検索して、自分のアカウントを作ってください。

<https://www.unrealengine.com/en-US/>

に行って



SIGN IN 押して



Sign Up 押して、自分のアカウントを作ってください。なお、既に持っている人は普通にサインインすればいいです。

で、アカウント作ったら最初にも言いましたが、既にランチャーは入ってるはずなので、ランチャーを立ち上げて、ログインしましょう。

ランチャーの起動

ウィンドウズマーク+Q を押した状態で"Epic"と打ち込んでください。Epic Games Launcher が見えますので、それを起動してください(Q 押さなくても OK かも)

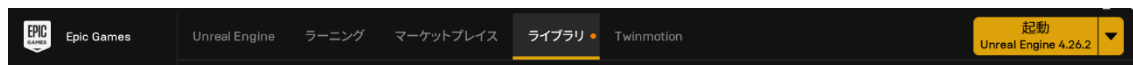
このランチャーじたいは Steam みたいなもんですね。買ってるゲームがあればこのランチャーから遊べるし、その中のツールとして UnrealEngine4 があるって感じです。Unity とはちょっとスタンスが違います。

はい、そして大抵の場合は UnrealEngine の本体がインストールされてないと思いますので、バージョンは何でも構いませんが、インストールします。

ちなみに、いつまたリモート授業になるかわかりませんので、今日家に帰ったら同じようにインストールしてください。

なお、UE4 一個で 4GB~6GB くらいあるので、PC の容量がない人は空けるか外付け HDD(SSD?) にインストールしておいてください。

インストール出来たら



起動ボタンが付きますし、『ライブラリ』に合わせてもらえればこんな風に



バージョンごとの UE4 が出てきます。なおバージョンごとに 4GB は食うので、余裕のない人は古いバージョンは捨てましょう。

あとですね、これを何故授業の最初に話したかということ、学校の回線が本当にクソクソのクソ過ぎて、もっと言うと無線が死ぬほどクソ過ぎるので 90 分の授業内でインストールが終わらない可能性が高いからです。

恐らく初日はインストールだけで終わるでしょう。なお、ちよつとこの後の予定は知りませんが、この授業内でインストールが終わらない場合は、次の時限まで持ち越しとなるので、もしその PC が次の時限で使えないなら、『一時停止』してログオフ状態にして次の人に渡して、くれぐれもシャットダウンしないようにと頼んで、PC 戻ってきたらまたインストールを『再開』してください。

まあ皆で一斉に同じ時間にやるから重いので、とりあえずこの時間中にはインストールできないと割り切って、放課後にゆっくりインストールするのもアリですが…。

と、まあインストールのハードルすら結構高い UE4 ですが、まあ頑張って作っていきましょう。正直家でやった方が速いと思いますので、早くリモートになってほしい。

あ、家の PC が遅い場合は UE4 けっこう辛いじょうきょうになりますのでご了承ください。基本的に授業としては『軽量版』で作っていきませんが、それでも結構重いので…。

そもそも UnrealEngine4 って何？

ゲーム学科でまったく知らない人はいないとは思いますが、インストールに時間もかかるはずなのでお話しします。

UnrealEngine4 というのは、UE4 と略され、最近様々なところで注目を浴びているゲームエンジンです。

これはフリーのゲームエンジンで様々なゲームがこれまで作られてきています。

しかもオープンソースなので、改造もバンバンやっています。ゲーム開発者にとってはうれしいエンジンです。

日本での活用事例の有名どころではロリポップチェーンソーやデビルメイクライやギルティギア Xrd などです。



このソフトは海外で評価を受けてきたソフトですので、海外での採用事例に比べるとまだまだ日本では少ないのが現状です。

本来はリアルな表現を得意としており、Bioshock Infinite などでは美しいグラフィックス表現が

評価されています。



昔のゲームだと、ミラズエッジや Postal2 などでも使用されていたようです。

日本での採用事例は海外に比べれば少ないです。

理由としては、リアルな表現は得意ですが、セルシェード(アニメ風表現)に関してはそれほど得意ではないため、日本で人気のゲームなどでは採用されづらい傾向にあるようです。

『えっ?でもギルティギア Xrd なんてもるアニメじゃーん!!』って思うかもしれませんが、そこは製作者さんの血と汗と涙とその他何かの結晶であそこまでアニメ風にできているのです。

<http://www.4gamer.net/games/216/G021678/20140703095/>

<http://www.4gamer.net/games/216/G021678/20150317055/>

<http://fox.2ch.net/test/read.cgi/poverty/1403442136/>

これらの記事を読めばわかりますが、アメリカの技術の粋を集めた UnrealEngine4 の開発者たちもびっくりするレベルの技術です。

これはもう作り手のこだわりということですね。

というわけで、最初に知っておいてほしいのは、現段階での UE4 はリアル表現に向いてるけど、アニメ風表現をしようとするたびつ大変なことです。

在野のゲームプログラマさんも、一生懸命頑張っているようですが

<http://qiita.com/junyash/items/0e528e1f291a710cf04f>

やはりギルティみたいな表現はそうそう作れないようです。

そういう意味で、日本的なアニメ寄りの表現はまだまだ Unity に軍配があがってそうです。

http://qiita.com/MuRo_CG/items/c417ef6d6cbeed3dd42b

ただ、本当にリアル表現に関しては現状 UE4 に適うエンジンはないので、Steam が流行っている昨今の状況を考えてか、ゲーム系各社が導入をどんどん始めているようです(日本市場狙いだけだとゲーム開発は採算が合わなくなっているため)。

実際にはプログラマはこのゲームエンジン自体の改造も含めてですが、このゲームエンジンを使いこなすことが要求されてきます。

ゲーム業界の UE4 ニーズについて

ゲーム業界における UE4 のニーズは高まってきていますが、基本的にコンシューマ向け開発の会社ばかりです。これも今後はどうなっていくのかは分かりませんが、まだまだ VR 業界やスマホゲー業界では Unity が優勢のようです。

そのコンシューマ業界においても、UE4 を採用している企業でさえ『UE4 やるのはいいけど C++ やれる人がいいです』とのこと。UE4 は C++ でも開発することができ、現場で BP 使ってゴリゴリやってる企業はまだまだ少ないようです。

ただ、BP だけでも相当な作り込みができていれば当然のように企業からは評価されますので…っていうか、BP 理解して、ある程度 C++ がわかれば移行はそれほど難しくない(嘘…結構大変)ので、UE4 は BP でもいいけど C++ できる人が欲しいって所ですね。

ちなみに UE4 は最初の選択で C++ で開発することができますが初心者にはお勧めできない。クラスで 1,2 を争うような学生さんでも挫折した経験あり。僕も正直よくわからん。

C++ で UE4 の何が難しいのか

まずですね、普通の C++ とちよつと違うんですよ。

時間があればいずれ話しますが、C++ の言語機能の中で使っちゃいけないものがあつたり(この制約は年々マシになってるので、いずれ普通に使えるようになるかも)と、

あとは UE4 特有の変な文法(C#のときの Attribute みたいなもの)があつて、ただでさえ C++ ややこしいのにそれを覚えなければならず、

あと UE4 じたいが持つてる機能も把握してないととても使える状態ではないということで、覚えるというか把握しておかなきゃいけないことがズゴーン!と増えて、そこをおろそかにして C++ できるからヘーキヘーキって手を出しちゃうと結構つらいです。

まとめると

- 標準の C++ と UE4 の C++ とはちょっと違う
- 覚えることが多い
- 予想外の所でバグる
- UE4 の構造を把握してないとマジツライ

というのがあるので正直いきなり UE4C++ はお勧めしません。ということでロードマップとしては

UE4 のブループリント(以降 BP と書く)を使って、テンプレートも使って一通りゲームっぽいものを作る→BP によるテクニックを初歩の一部分だけでもひととおり把握する(一通りと言っても実はここがかなりキツイ)→一部を C++ で書いてみる→フル C++ で書いてみる(プロジェクトを C++ で作る)。

てな感じが適切かなと思います。

じゃあブループリントは簡単かって言うと…

ただ自分でブループリントを使いこなせるようになるためには普通に

- UE4 エディタじたいの基本操作
- ブループリントの基本操作
- イベントドリブンの仕組み
- アクターの種類の把握(ここ結構難しいかも)
- マテリアルエディタを使いこなす(シェータ的理解が必要)
- アニメーション(僕個人的にはここがキツイ)
- ブラックボードだのビヘイビアツリーだの…

このくらいの事が何となくわかってる必要があって twitter とか見てるとなんかデザイナーさんとかでもみんなホイホイ使ってて、簡単そうに見えますがきちんとしたゲームを作るとなると意外と、意外と難しいです。

まあ慣れさえすれば、ゲームのアイデアを手っ取り早く形にできてしまうので、便利なツールであるとは思いますが。あと見た目が凄くいいので、テンションも上がりますし。

全部一人で作る人はこれ全部把握してるので凄いなーって思います。なお、チームでやる場合は

- イベント BP 系
- アニメーション BP 系
- UI ウィジェット BP 系
- マテリアル系

とかで分担した方がいいんじゃないかなと思います。全部自分で完璧にするのは大変(やってる奴はいるけどさ)

というわけで授業の流れ(コマシラバスともいう)

1. イントロダクション(授業の方針の説明と UE4 の概要)
2. テンプレ使って簡単になんか作ってみよう①TPS
3. テンプレで簡単に②FPS
4. テンプレ(ry③サイドスクローラー
5. 色々なマテリアルを作ってみよう
6. マテリアル弄って珍しいエフェクト作ってみよう
7. ちょっと珍しい BP 効果使ってみよう①スタティックメッシュ切断
8. 珍しい BP②ディスメンバーでマネキン君をバラバラに
9. ポストエフェクトをかけてみよう
10. 一部だけ C++化してみよう
11. C++でプロジェクト作って、本当に簡単な動きをやってみよう

1コマで終わるのもあれば2~3コマ使うものもあるので、何となくこんな感じということを知ってもらってればいいんじゃないかなと思います。

はじめてのUE4

まあ全然使ったことがない人もいますので、初めての前提でかいていきます。

UE4 起動

インストール出来たら起動ボタンを押してください。

起動を押すと、しばらくして、ウィンドウが立ち上がりますので、『新規プロジェクト』タブを選択して、『ブループリント』タブを選択した状態で、『First Person』を選択して、『プロジェクトを作成』ボタンを押してください。



ちょっと時間がかかりますが、固まったわけではないので、しばし待たれよ。

そうすると漸くアンリアルエディタが立ち上がります。

エディタを操作

ひとまず、これがどういうものを体験してもらいましょう。

上部メニューにある▲ボタンを押してみてください。

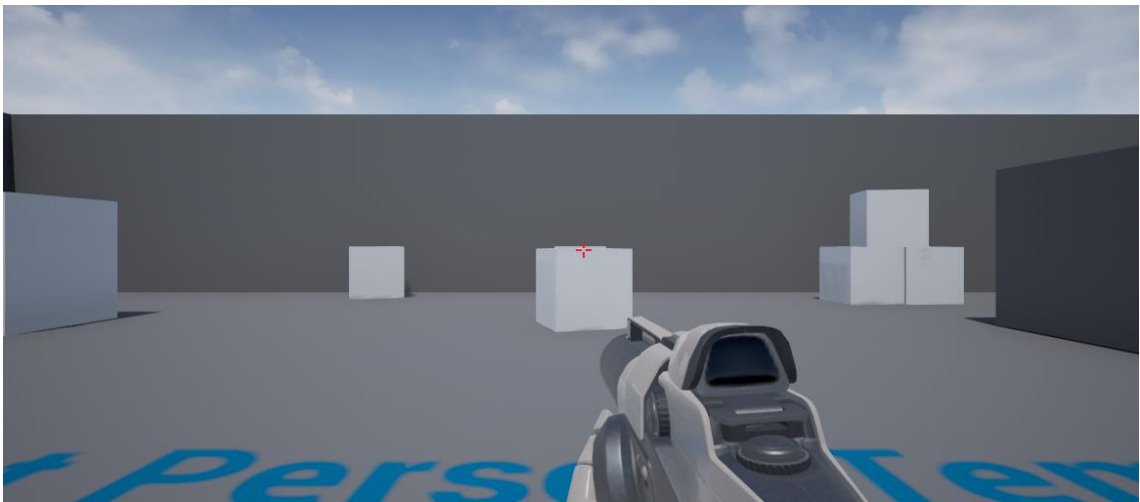


海外ゲームなどではおなじみのFPSゲームのような画面になります。

操作法は洋ゲーやる人ならわかると思いますが、WASD で前後左右に移動して、マウスを上下左右にふる事によってカメラの向きが変わります。スペースキーでジャンプします。

そして左クリックをすると『ボール』を飛ばすことができます。

しかし画面がこの通り…



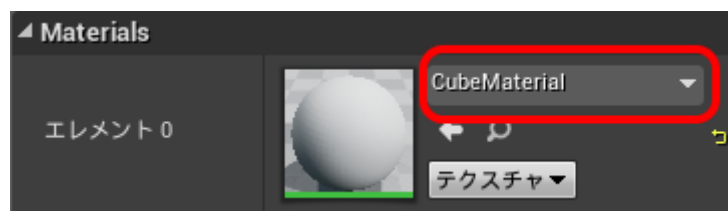
無機質ですねえ。

なお、実行をやめるには ESC キーを押します。それでもやめられない場合は、■の『停止』ボタンを押します。

材質を変更しましょう

そこで地面に土のぬくもりを与えましょう。

地面を選択して『マテリアル』を変更します。地面をクリックしてください。



そして右側の Materials の▼を押してください。



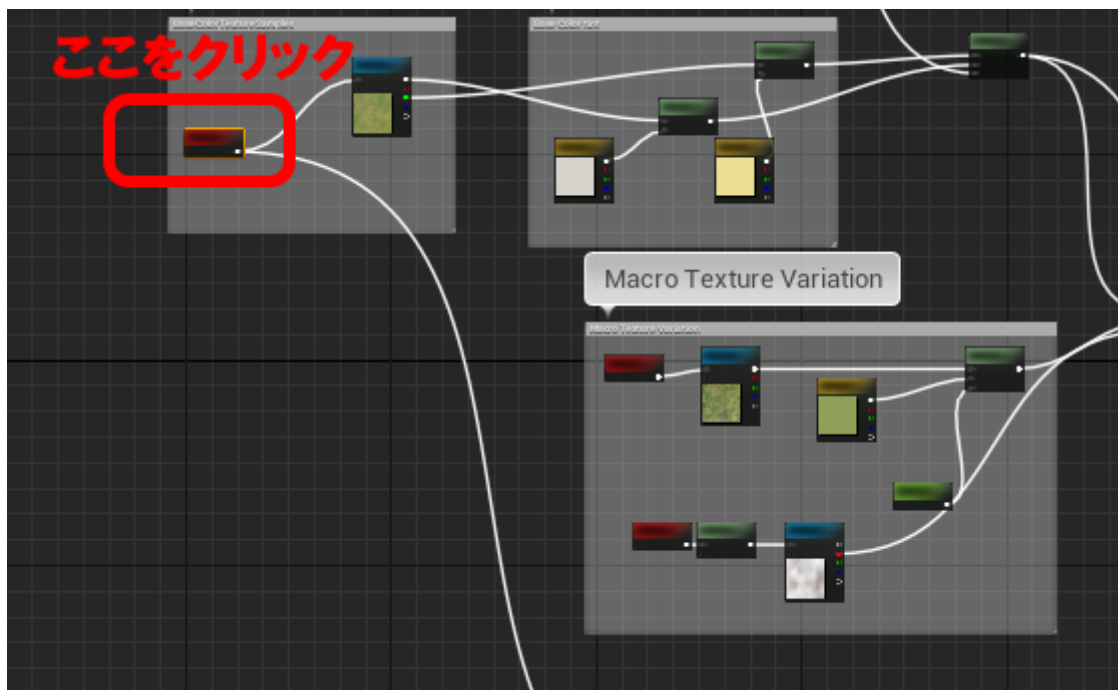
その中から『M_Ground_Grass』を選択してください。
しばらくすると地面の材質が変わります。待ちましょう。



う〜ん…
粗いですね。
調整しましょう。



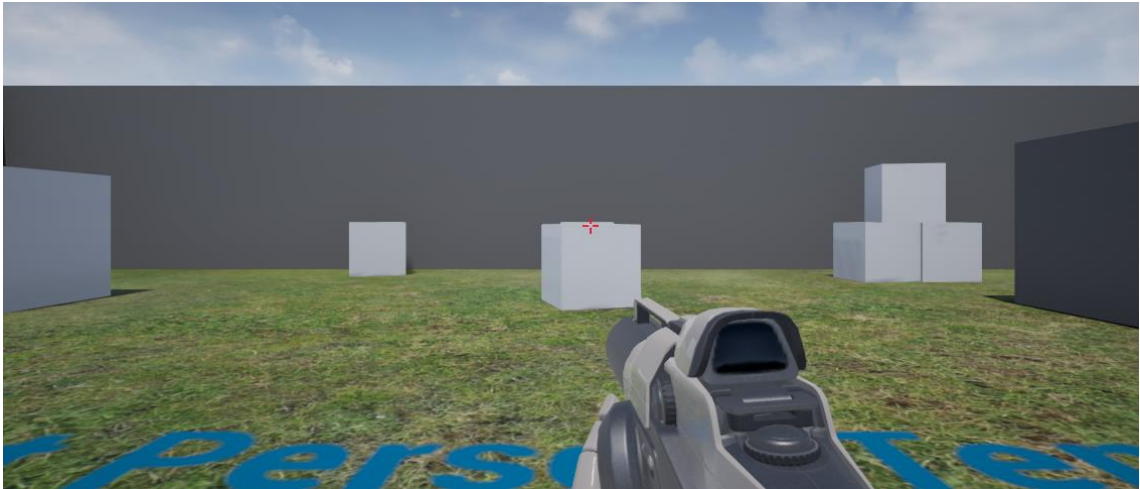
マテリアルをダブルクリックしてください。そうすると……UE4 の醍醐味の『ロードエディタ』が立ち上がります。



左上にある TexCoord(0)をクリックしてください。
 そうすると左側の『詳細』の部分の UTiling と VTiling が 0.5 だと思いますので、両方 5 くらいにしてください。



そうすると



ぬくもりですねえ…。

ファイアー!

しかし…弾がシヨボいですね。



ピンポン玉かよ!!!

でもね…これ…弾はどこから出てるんでしょう…?

これはねえ…探すの結構面倒なんですけど、

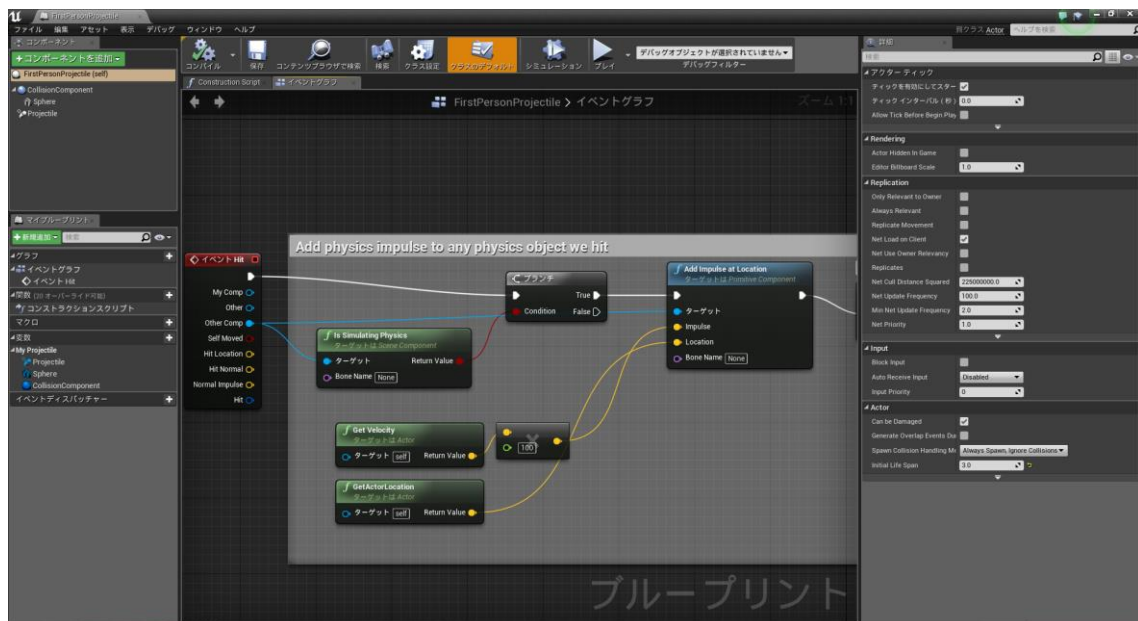
下のコンテンツブラウザを見て『フィルタ▼』の横に『FirstPersonProjectile』と入力してください。

そうすると



こんな感じで3つの弾が出てくると思います。
一番左のやつをダブルクリックしてください。こいつはタダのボールではなく『意思を持ったボール』なのです。

ここがプログラムっぽい…かな？

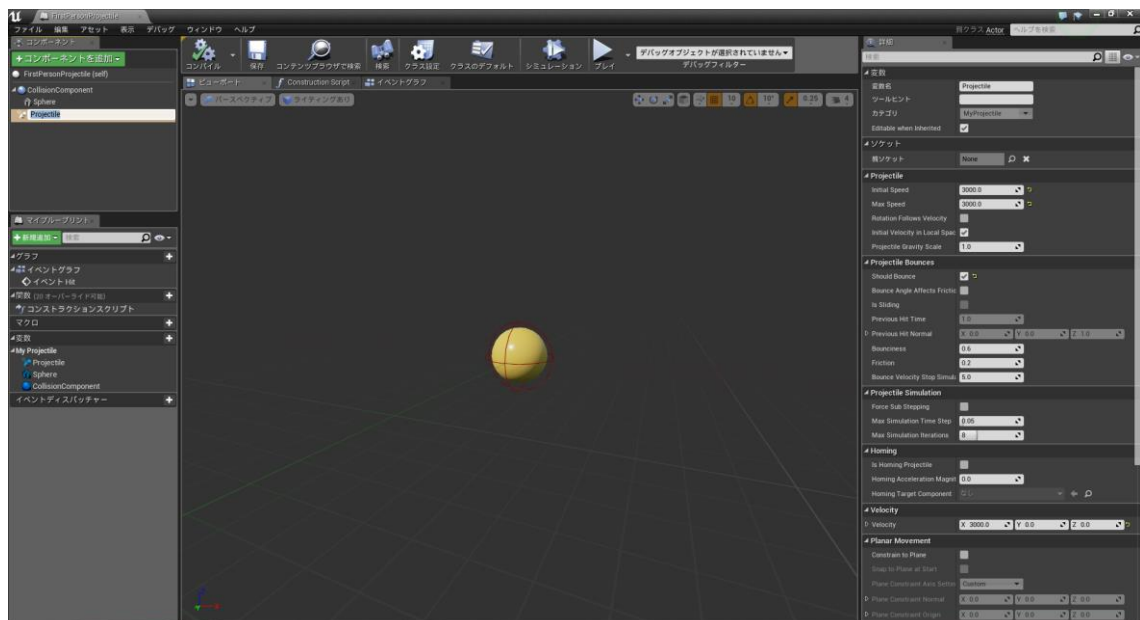


でもここはいじらないのだ。
左の『コンポーネント』にある『Projectile』をダブルクリックしてください。

ちなみに『Projectile』って英語のイメージは『ばびゅーん!』って飛んでいくモノってイメージです。



日本語にちょうどいい言葉がないんですが、そういうイメージです。ともかくダブルクリックすると



こんな感じになります。ピンポン玉じゃなくて鉄球みたいにしましょう。
Sphere って書かれてるところをクリックすると右側にマテリアルが出てきますので

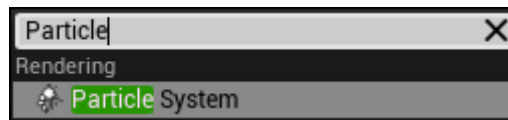
『M_Metal_Brushed_Nickel』を選んでください。

鉄球っぽくなると思います。

でもなんかつまらないですね。

ファイヤー! しましょう。

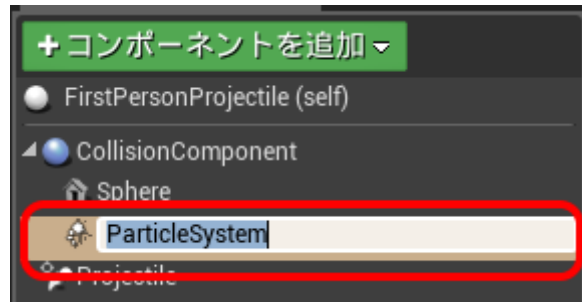
『コンポーネントを追加』を押してください。



そこで『Particle』と入力してください。

Particle っていうのは日本語で言うと、炎やら水やらのエフェクトに使用するものです。

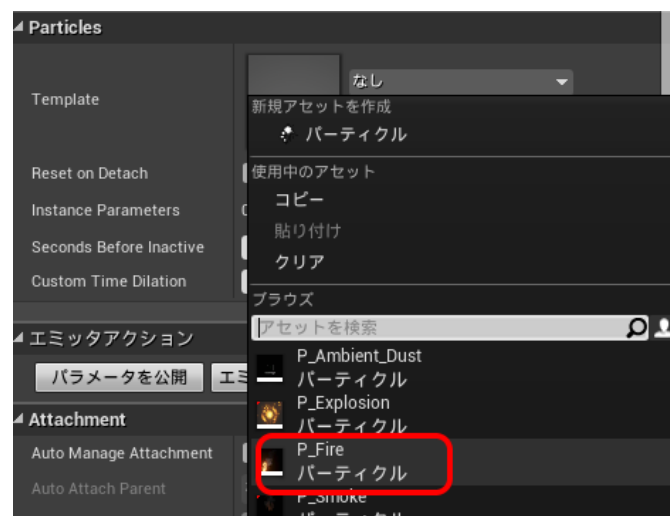
ParticleSystem っていうのが出てくるので、そいつを選択してください。



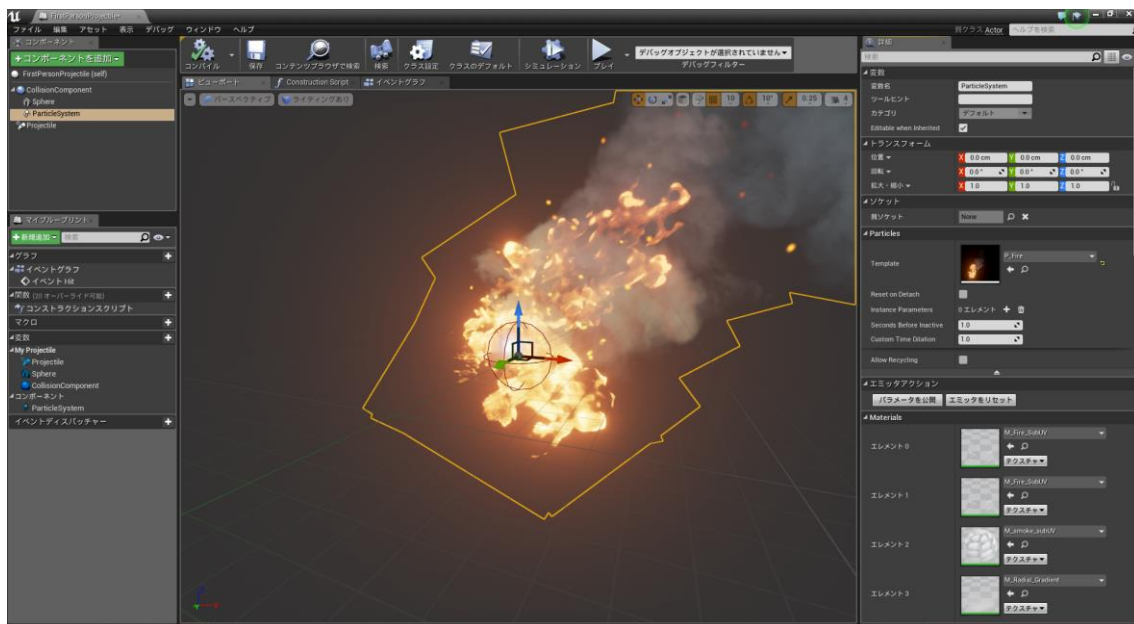
こんな感じになります。

さあ、ファイアーしましょう!!!

右側に Particles っていうのが有ると思いますので、その Template っていうところのドロップボックスを選択して、P_Fire を選択してください。



そうすれば…



ファイアー!!!

じゃあ、実際のゲーム画面を見てみましょう。この画面を最小化して、元の画面を表示して、実行ボタンを押してみてください。そして弾を発射しましょう。



おお!!ファイアー出とる!!!

でもなんかショボいんだ。

というわけで、また先程のパーティクルを開いて、スケーリングを3倍にしてください。もうちょっと強い感じにはなります。

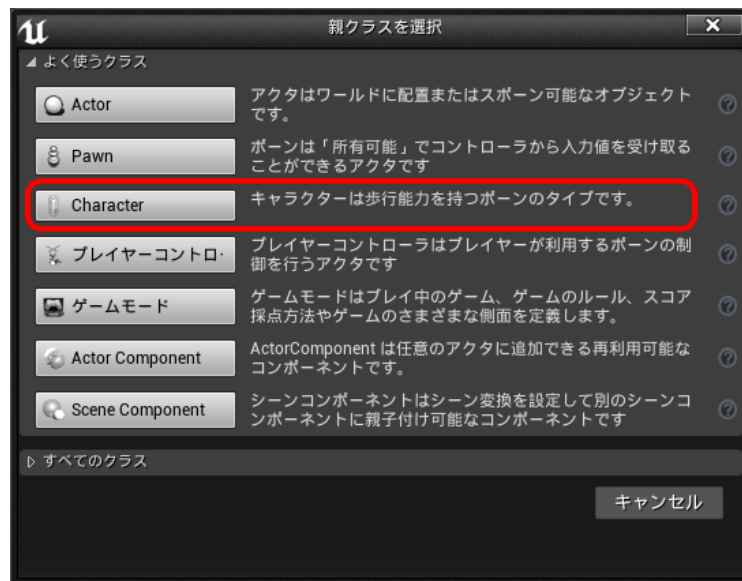
それでもちょっと弱かったら複製してください。

ParticleSystem を選択して『Duplicate』もしくは『複製』を選択してください。

かなり強くなると思います。

敵(マト)を作ろう

コンテンツブラウザ→新規追加→ブループリントをクリックしてください。



その中の Character をクリックしましょう。

NewBlueprint ってのができるので、名前を enemyBP と名前を変更してください。

次はその enemyBP をダブルクリックしてください。

enemyBP の中に入って、コンポーネントを追加。Cylinder を追加してください。円柱になると思います。

あとはこれに当たり判定を追加していきます。

当たり判定を追加しよう

まずは弾に当たり判定を追加します。

さきほど扱った『FirstPersonProjectile』をクリックしてください。

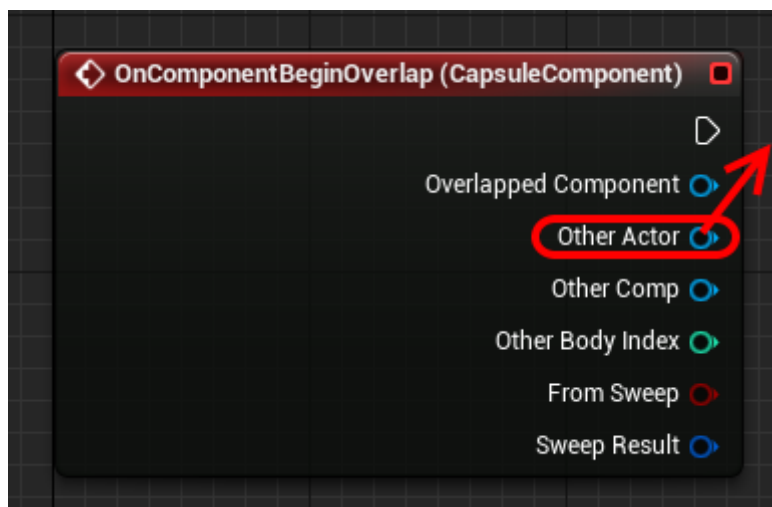
これに当たり判定を追加します。

コンポーネントを追加→SphereCollision を選択してください。

そうすると、なにやら当たり判定球のようなものができたと思います。
これが当たり判定です。

こっちはこれで終わり。
次は敵(マト)側を変更していきます。

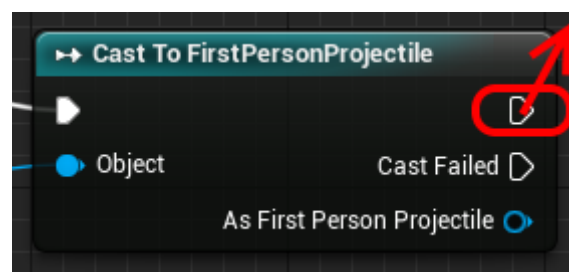
CapsuleComponent というのが有ると思いますので、そこで右クリック
AddEvent→OnComponentBeginOverlap を選択してください。



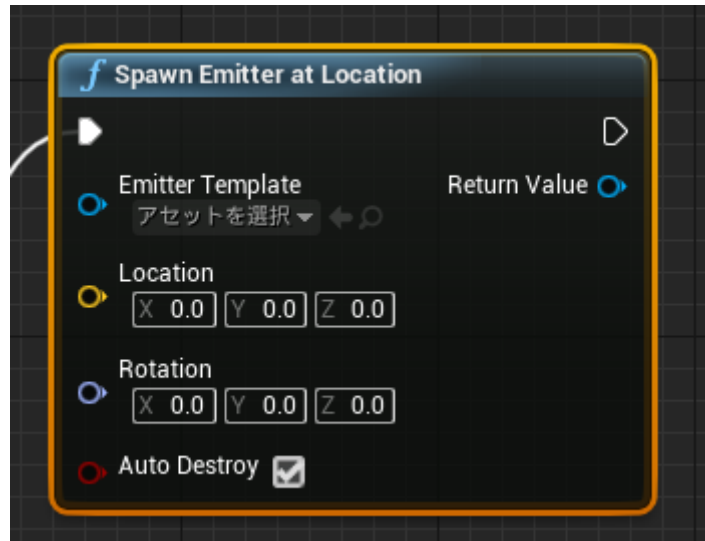
こんなのが出てくるといいますので OtherActor からドラッグ・アンド・ドロップして、『First PersonProjectile』を選択してください。

これで『弾に当たったら〜』という処理ができました。

今から弾に当たったら『〜どうするのか』という部分を設定していきます。



今度は CastToFirstPersonProjectile の右のこれ…五角形の矢印っぽいのからドラッグ・アンド・ドロップして、SpawnEmitterAtLocation を選択してください。



これはヒットエフェクトを出すためのものです。

で、↑のEmitterTemplateの『アセットを選択▼』を押してください。そこでP_Explosionを選択。で、これ、どこでもエフェクト出て良いわけじゃないので、どこにエフェクトを出すか指定します。

なにもないところで右クリックして、『Get Actor Transform』を選択してください。

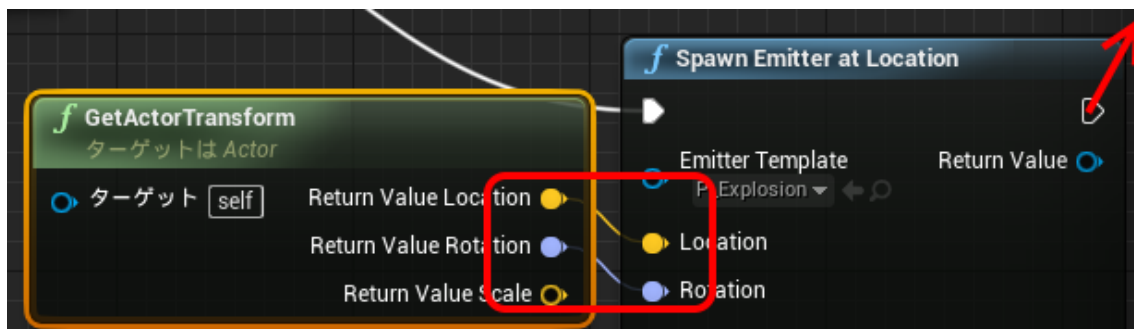


こういうのが出てきますので、Return Value で右クリック。今度は『構造体ピンを分割』としてください。



こういうふうになるので、Return Value Location, Return Value Rotation 双方からドラッグして、SpawnEmitterAtLocationにつなぎます。

こんな感じ。



これだけではまだまだ…。爆発するだけでは敵を倒した感がないので、さらに追加します。

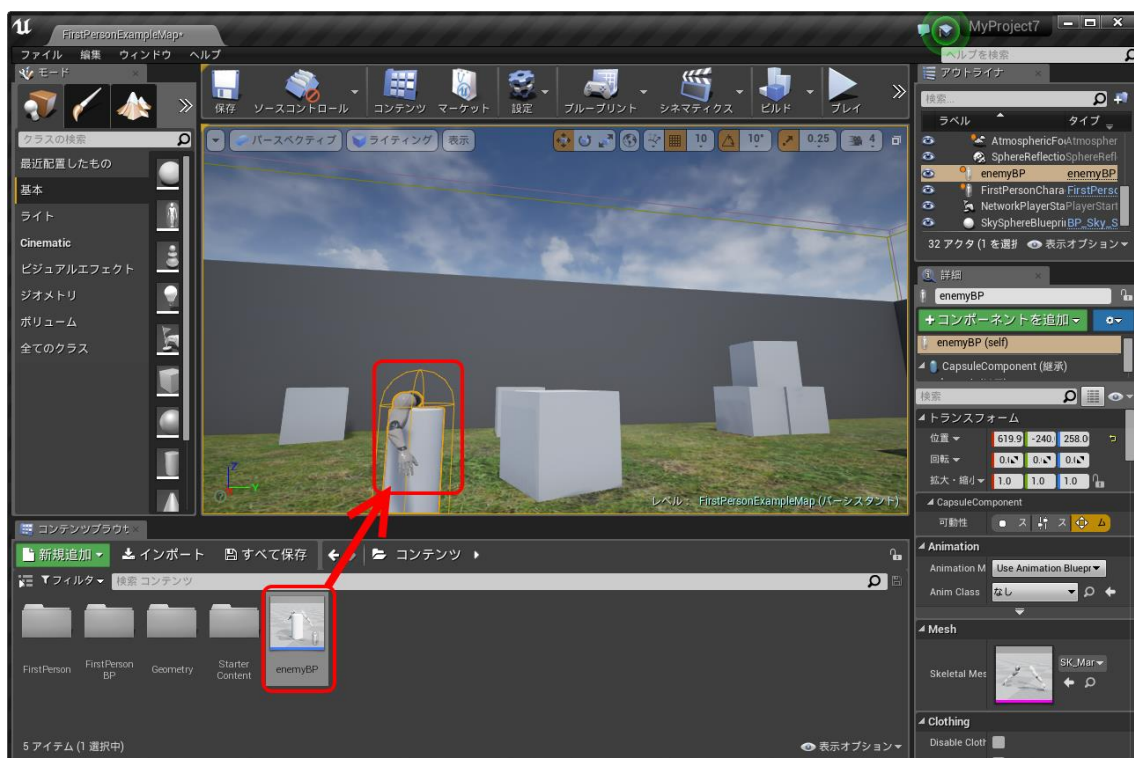
SpawnEmitterAtLocation からドラッグして、

「SetLifeSpan」

を選択。

「InLifeSpan」に 0.3 くらい…。これが死ぬまでの時間になります。

これで仕組みはできましたので、画面に配置します。

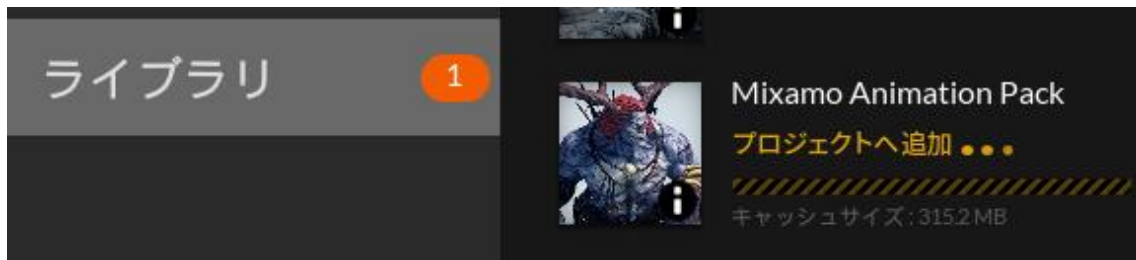


ドラッグアンドドロップで配置できます。

さあ、こいつをいくつか配置して、弾を当ててみましょう!!

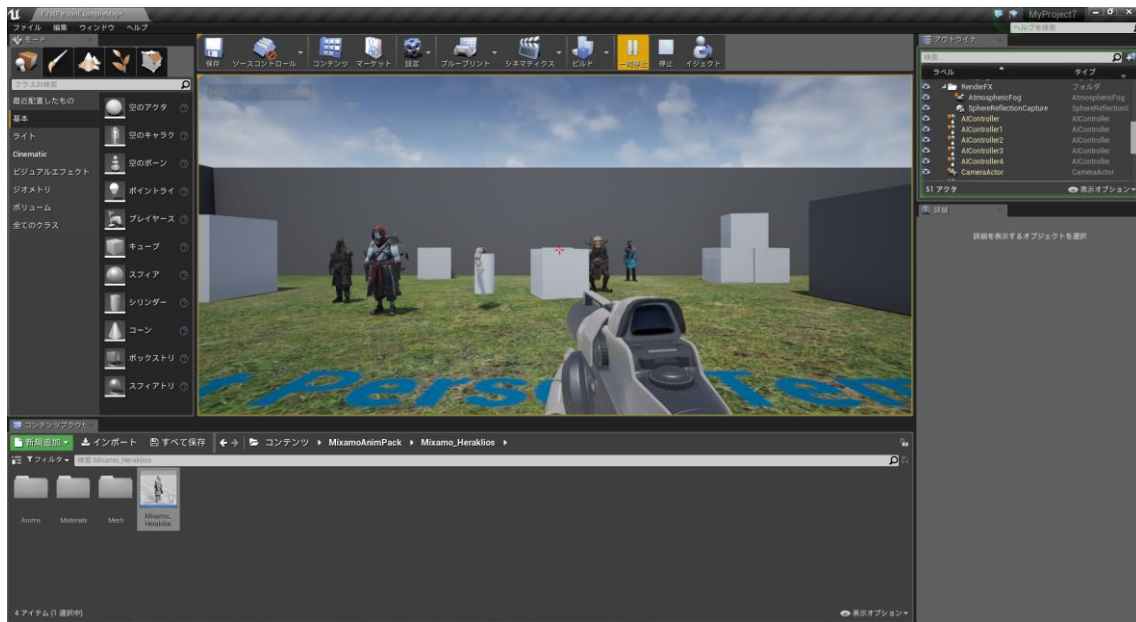
爆発して消えるところまで確認できたら、しばらくプレイして遊んでみてください。

実際にはこんな円柱ではなく
追加アセットとして



のようなものがありますが、有料だったり、組み込みに時間がかかったりしますので、今回はこの円柱でやってみました。

展開の時間があれば



色々なキャラクターを配置できますがそれはまた次回の機会に。