

レベルデザイン（通路）の中を歩きます。

Unity がこういった操作体系であるか？を体感する為の簡単なサンプルコンテンツを制作します。

作業のみであれば、10分ほどで終わってしまうボリュームなので、組み立てや完成だけを急がずに、陥りやすい失敗や修得するためのコツ・概念などのレクチャーに耳を傾け、全体像を把握する心掛けで挑んで下さい。

今回のコンテンツは、単純な通路を端から端まで歩くだけのコンテンツとなります。



シーンの構成

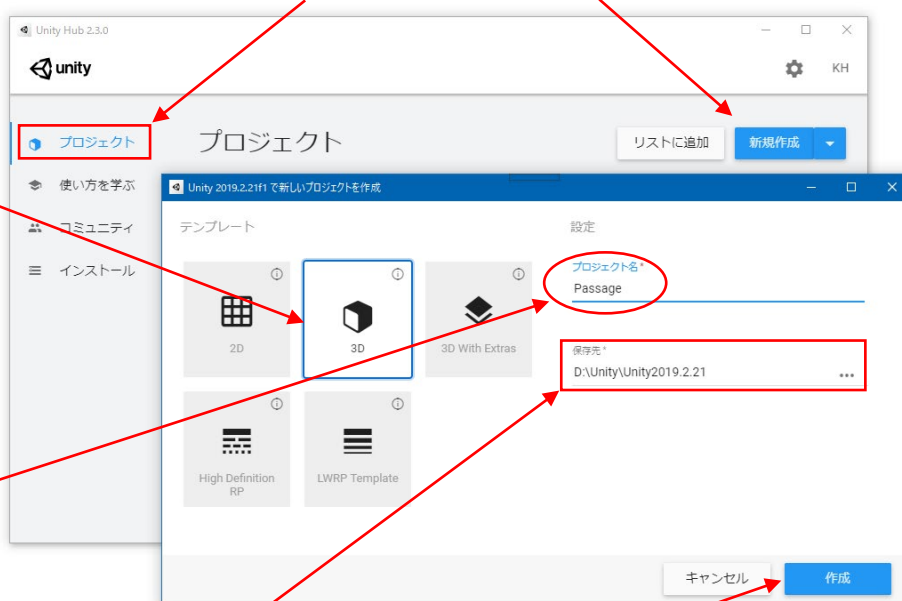
【STEP1】プロジェクトの新規作成

- Unity を起動します。ところが、Unity の前に、UnityHUB(ハブ)と呼ぶ玄関パネルが起動します。
- 学ぶコーナーもありますが、プロジェクトを作りたいので**プロジェクト**を選び、**新規作成**を押下します。

テンプレートでは **3D** を選んでおきます。

半角英数文字でプロジェクト名 **Passage**(パッセージ: 通路)を入力します。

保存場所 **Location**(ロケーション)を決めます。
教室の環境によって指定します。最後に**ボタン作成**を押下します。
ここでやっと Unity 本体が起動します。



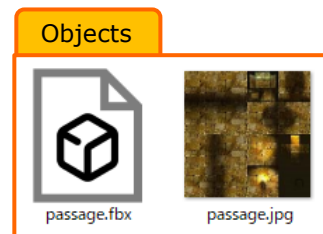
【STEP2】 外部素材を読み込む①

Unity はその名の通り統合化ソフトですので、各種グラフィックソフト、プログラム記述専門ソフト、サウンド生成ソフトなど、様々なソフトから出力されたデータを統合化する為に、外部素材として読み込む必要があります。

この読み込み方におおまかに2系統があるのです。

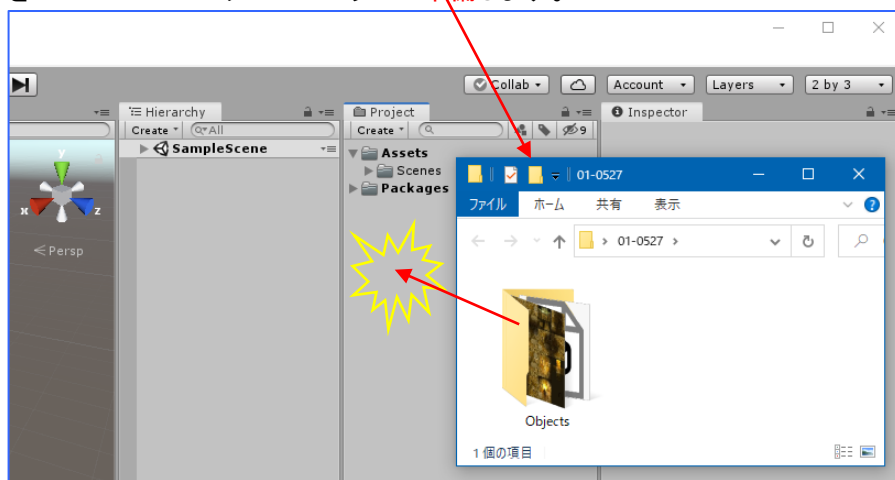
- ① 作られたデータを、そのままの形式で読み込む。
- ② Unity でパッケージ化された形式で読み込む。

まずは、①のそのままの形式で読み込む方法からです。今回は2個のファイルを1つのフォルダにまとめた物を配布していますので、これを Unity のプロジェクトに読み込む方法を演習します。

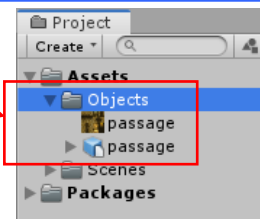


- 読み込ませたいデータ(フォルダ)を Windows のエクスプローラーで準備します。
- このフォルダを、Unity のプロジェクト欄にドラッグ & ドロップします。

- 形状データ(fbx 形式)
passage.fbx
- 画像データ(jpg 形式)
passage.jpg



このようにプロジェクト内に読み込まれたことが視認できます。



今後、他者とのコラボレーションも増えて来て、グラフィック担当者が作った2D・3Dの素材ファイルを読み込む機会が出てきます。自身でグラフィック素材を作成した場合はもちろんのことです。

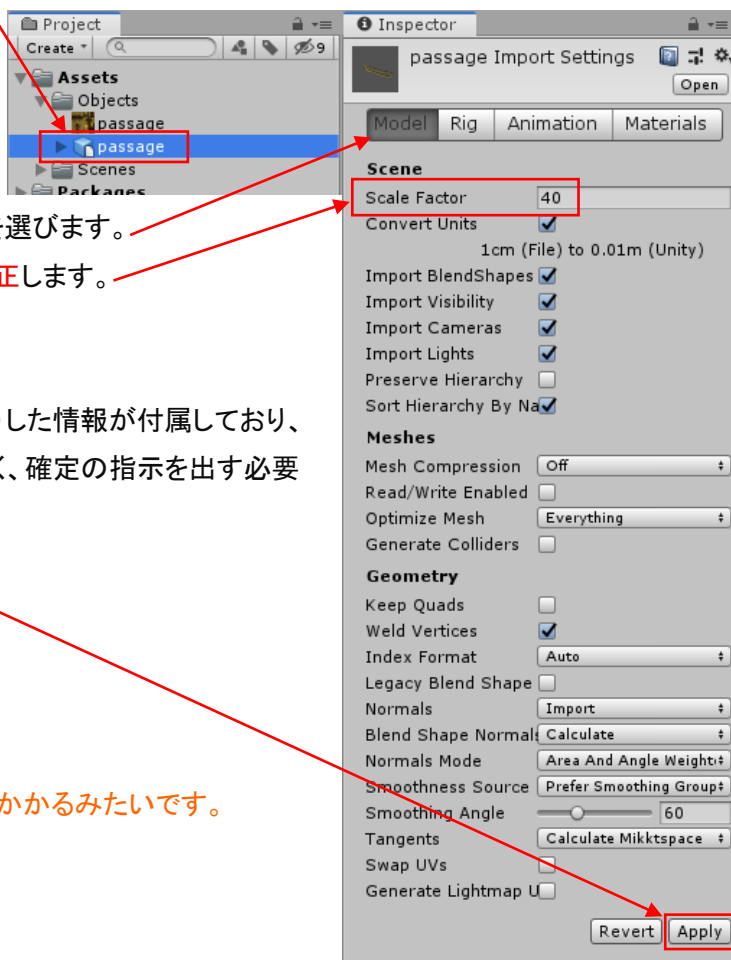
こういった機会でのデータ読み込み方法が、①そのままの形式での読み込みです。

【STEP3】 ゲームオブジェクトを設置する

今回はソフト Maya で学生が作ったオブジェクトを配布しています。少しサイズが小さかったので調整します。

- プロジェクト欄の Objects > **passage** を選択します。

画像と形状データが同じ名前なので、アイコンで区別します。



- インスペクタに着目し、4つのタブから **Model** を選びます。

Scene の中の項目 Scale Factor を **40** に修正します。

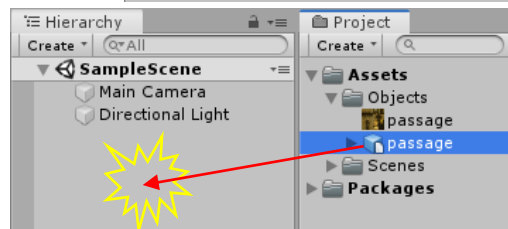
外部から読み込んだ素材にはインポート(輸入)した情報が付属しており、ここではそれを修正したので、仮の設定ではなく、確定の指示を出す必要があります。

ボタン **Apply**(アプライ)を押下します。

大きさの変換作業が行われます。

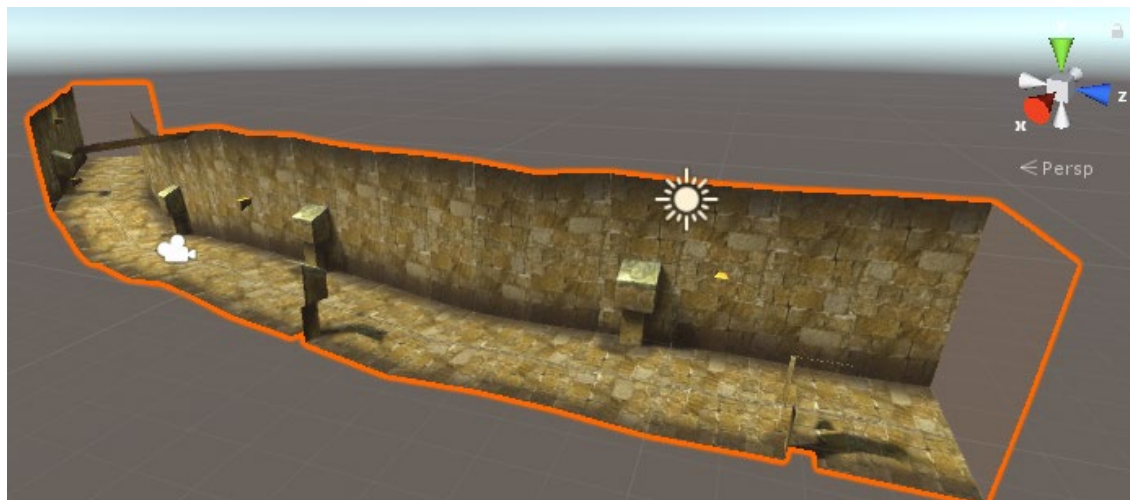
学生さんが作ったデータのせいか、少し時間がかかるみたいです。

- プロジェクト欄のオブジェクト **passage** をヒエラルキー欄の中にドラッグ & ドロップします。



- シーン内に通路(地下ダンジョンの一部かな?)のオブジェクトが表示されます。

何も仕掛けがないゲームのステージオブジェクトですが、このようにしてゲームの内容に沿ってアイテムを置いたり、イベントを仕掛けたりし、ゲームを作っていきます。



● ここで、シーン画面内の視野操作をマスターしてみましょう。

- 視点の回り込み : [Alt] + 左マウスボタン押下 + ドラッグ操作
- 視点の上下左右移動 : [Alt] + 中央マウスボタン押下 + ドラッグ操作
- 視点の前後進移動 : [Alt] + 右マウスボタン押下 + ドラッグ操作 (スクロールホイールを操作)

【STEP4】 外部素材を読み込む②

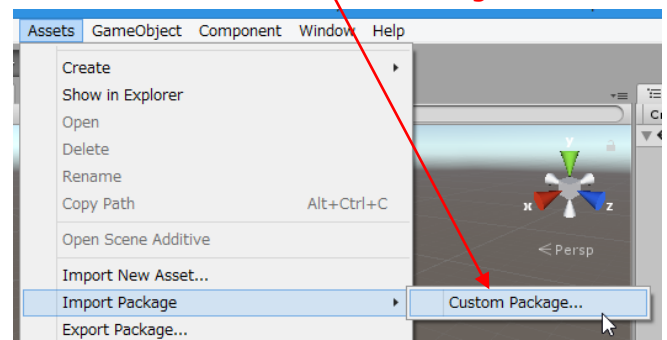
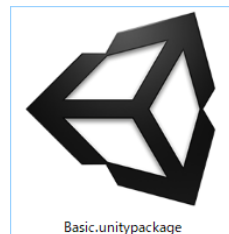
もう一方の読み込み方法を演習します。生のデータのまま読み込むのではなく、一旦 Unity の中で調整されて配布可能になったものをパッケージにして輸出(エクスポート)したデータ形式です。

当然、読み込む際は、その逆の作業ですから輸入(インポート)の作業となります。

配布形式は、導入に際して最適化された **パッケージファイル形式** としています。

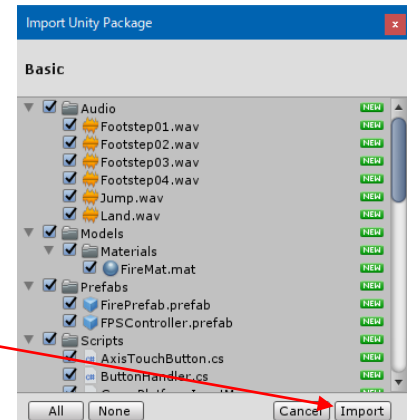
● メニューAssets(アセット)から Import Package(インポートパッケージ) > **Custom Package(カスタムパッケージ)** を選択します。

今回配布しているファイル **Basic.unitypackage** を選びます。



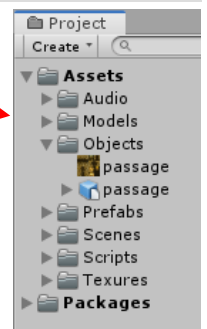
内容一覧が表示されます。この時点では、まだ読み込まれていません。

必要な物のみを選択したい場合があるので、一旦、圧縮状況のリストを表示しているにすぎません。



ボタン **Import(インポート)** を押下します。

非常に数多くの素材が読み込まれたことが判ります。



プレイヤーの設定

【STEP5】 1人称キャラクター(カメラ)の設置

Unity では、定番的なゲーム構成部品なのに、全世界の人々が同じものを作っている状況をナンセンスとみなして、数多くを準備して無料配布しています。

今回は、キャラクターの役柄(ロール)に扮して、世界を探索(プレイ)するゲームを作る典型的な構成部品をインポートしてあります。

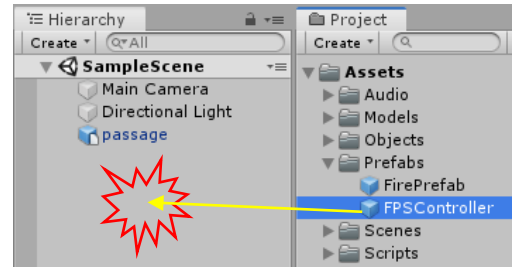
その中から FPS(ファースト・パーソン・シューティング)を構築するモジュールを設置します。

- プロジェクト欄のフォルダを▼マークを押しながら下層へたどっていきます。

Assets(アセット)

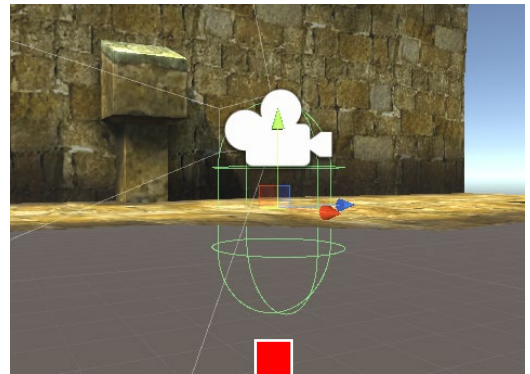
> Prefabs(プレハブス)

> **FPSController(FPSコントローラー)**をヒエラルキー欄にドラッグ&ドロップします。



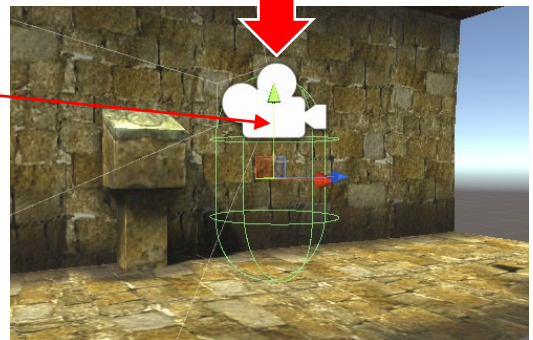
キャラクターの接触形状はカプセル型で示されます。

初期状態では、床で切腹したような位置に設置されるハズです。



- このまま緑の矢印をマウスで掴んで、上に引っ張り上げてやり、床からほんの少し浮いた位置になるようにします。

緑の矢印が出ていない場合は、キーボードのWキーを押下すると表示されます。



- この状態で一度、**プレイボタン**を押して動作確認してみます。



一人称のカメラが床をすり抜けて落ちていくのが判ると思います。

ゲームプレイの停止には、キーボードのエスケープキー(Esc)を押下してマウスカーソルを復帰させ、もう一度**プレイボタン**を押下します。

【注意事項】

この時に、プレイを押下したまま、つまりプレイ中のまま、制作作業を続けてしまって、プレイを停めた時にはその間の作業が全て消えてしまう！という事態に至るミスが多々あります。

制作作業に戻る際は、もう一度プレイボタンを押下して(黒に戻してから)作業するように注意しましょう。



衝突の設定

【STEP6】 接触形状(物理演算機能)の追加

3DCG ソフトで作ったオブジェクトを読み込んだだけなので、物体に接触演算する指定が抜けていた為に、カメラが落ちたのです。

通路のオブジェクトに物理演算の接触形状を持たせます。

- ヒエラルキー欄の通路 **passage** をクリックします。
- インспекタの最下部にあるボタン **Add Component**(アッド コンポーネント:コンポーネント追加)を押下します。

Physics(フィジックス) のグループの中の **Mesh Collider**(メッシュコライダー)を選択します。

このように取り付けられます。

ここでパラメータを設定することが多いですが、今回は何も変更しません。

- プレイボタンを押下します。
- カメラが passage のコライダーに接触するので、落ちないことが判ります。
- マウスでカメラの向きが変わり、キーボードの方向キーで通路の中を歩けます。スペースキーの押下でジャンプすることもできます。



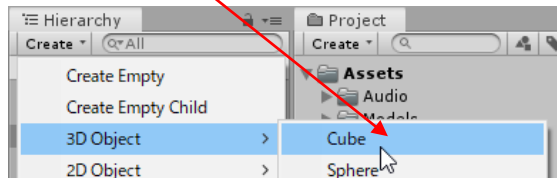
いろいろと歩いたりジャンプしたりしてみてください。端から落ちないようにね。

【STEP7】 見えない接触形状

このままだと、一人称カメラが通路の両端から落ちてしまいます。

先と同様に接触判定するオブジェクトを設置しますが、ここでは見えないオブジェクトを取り付けます。(実際のゲームの中でもプレイヤーには見えない壁が数多く用意されています。)

- ヒエラルキー欄の **Create(クリエイト)** から **3D Object(オブジェクト) > Cube** を選択し、立方体をシーン内に登場させます。

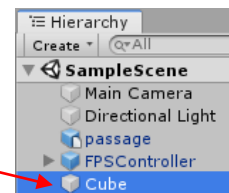


変なところに作られることもあって、どこに設置されたのか？

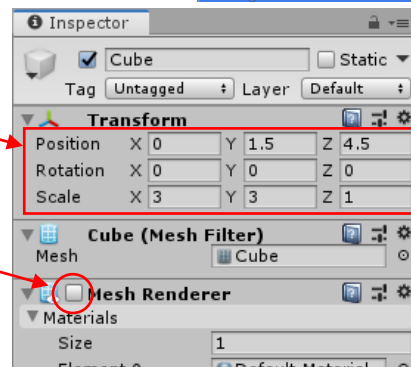
が判らない時は、ヒエラルキー欄の **Cube** をダブルクリックし

ます。すると、画面中央にフィットするかのように調整がなされます。

- ヒエラルキー欄の **Cube** を選択します。

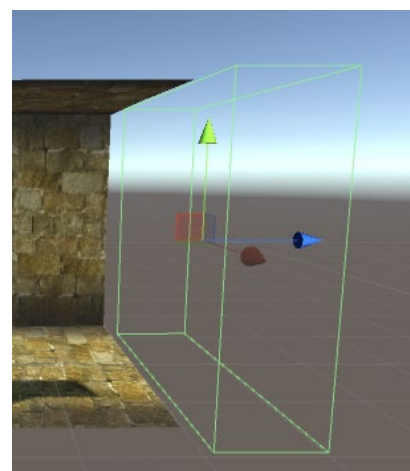


インスペクタでパラメータを設定します。



また、**Mesh Renderer(メッシュレンダラー: 形状描画)**のコンポーネントを**オフ**にします。

当たり判定はあるけれど、見えない物体になります。



- プレイボタンを押下します。



通路の端から落ちないことを確認してください。

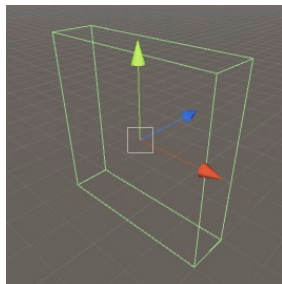
【STEP8】 物体の複製

このままでは、反対側の端から落ちてしまいます。

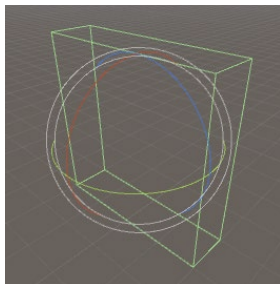
ここにも設置するのですが、新規作成するのではなく、先の完成した見えない壁を、複製して反対側に設置します。

- ヒエラルキー欄の Cube を選択し、複製 (Ctrl + D) を行って下さい。
選択していたオブジェクトが同じ場所にもう一つ存在する状態になり、今回の場合 Cube(1) が作成されます。
- この Cube を選択している状態で、 Gizmo を切り替えてみます。Gizmo は3種類が準備されていて、キーボードのショートカット「W」「E」「R」で切り替えます。見えますか？

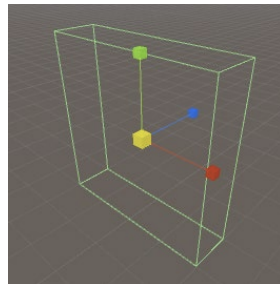
移動Gizmo: ショートカット[W]



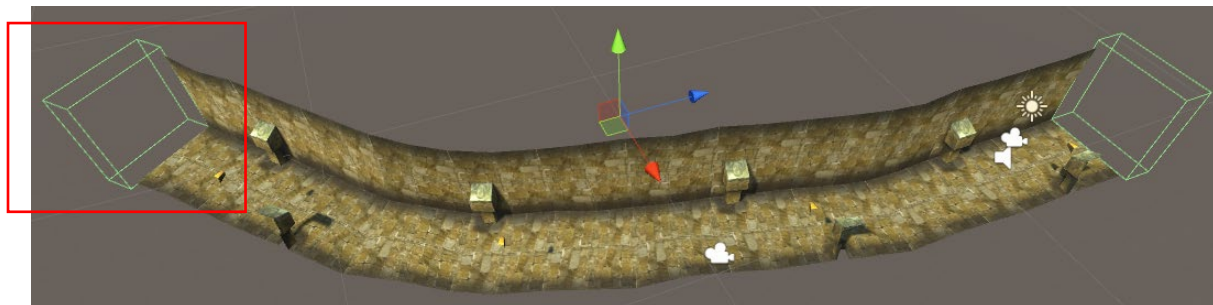
回転Gizmo: ショートカット[E]




拡大縮小Gizmo: ショートカット[R]



- 【命題】 この複製された Cube(1) をヒエラルキー欄で選択し、通路の反対側に設置して下さい。
まずは、移動のGizmo (W) に切り替えて、おおよその反対側に持って行きます。
その後、回転のGizmo (R) に切り替えて、通路の形状に合わせます。(下図参照)



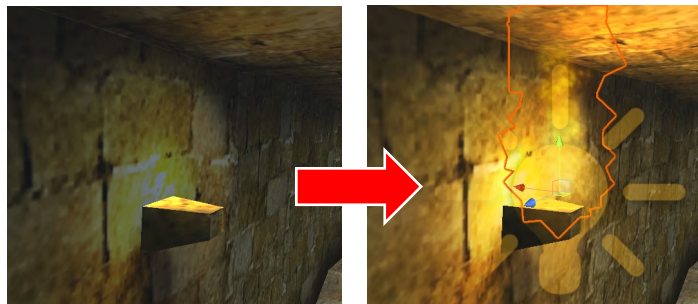
- プレイボタンを押下します。
通路の両方の端から落ちないことを確認してください。

演出効果の設定

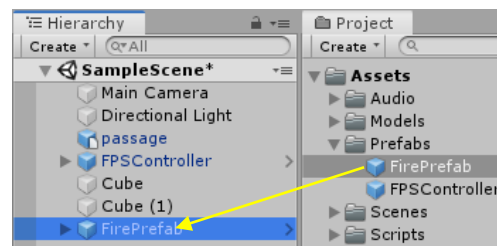
【STEP9】 松明(たいまつ)の設置

この通路内の壁面には、松明を設置するブランチが幾つかあります。ゲーム内でのエフェクト(効果)は、実はポリゴンなどで表現され、錯視とも言える手法で実現されています。

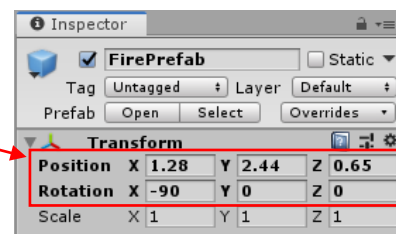
実際に松明のエフェクトを作って来があるので、取り付けてみましょう。




- プロジェクト欄の Prefabs(プレハブ) > **FirePrefab** をヒエラルキー欄にドラッグ & ドロップします。



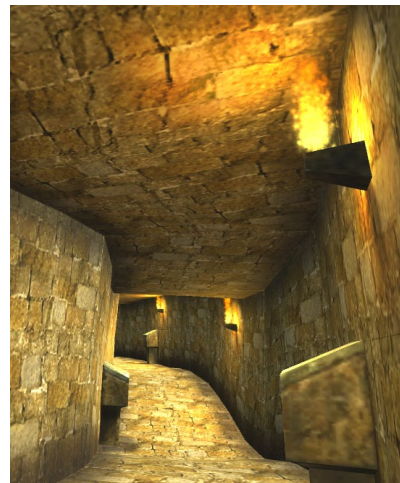
インスペクタでパラメータを設定します。



- プレイボタンを押下します。
松明を見上げると、まずまずの感じで燃えているように見えることが判ります。



- 【命題】 この松明を、先と同様の方法で複製して下さい。そして妥当な位置に移動させる作業を、残り3カ所について行い、合計4つの松明が灯った状態にして下さい。



以上