

また、本教材は「UnisideGame」という名前で以下の URL からダウンロードできるようになっています。ぜひお試しください。

- <https://www.shoehisha.co.jp/book/download/3602/read>

ダウンロードしたプロジェクトをUnityで開く方法はこちらを確認してください。

参照 → 「4.2 まずはサンプルゲームを実行してみよう」 90 ページ



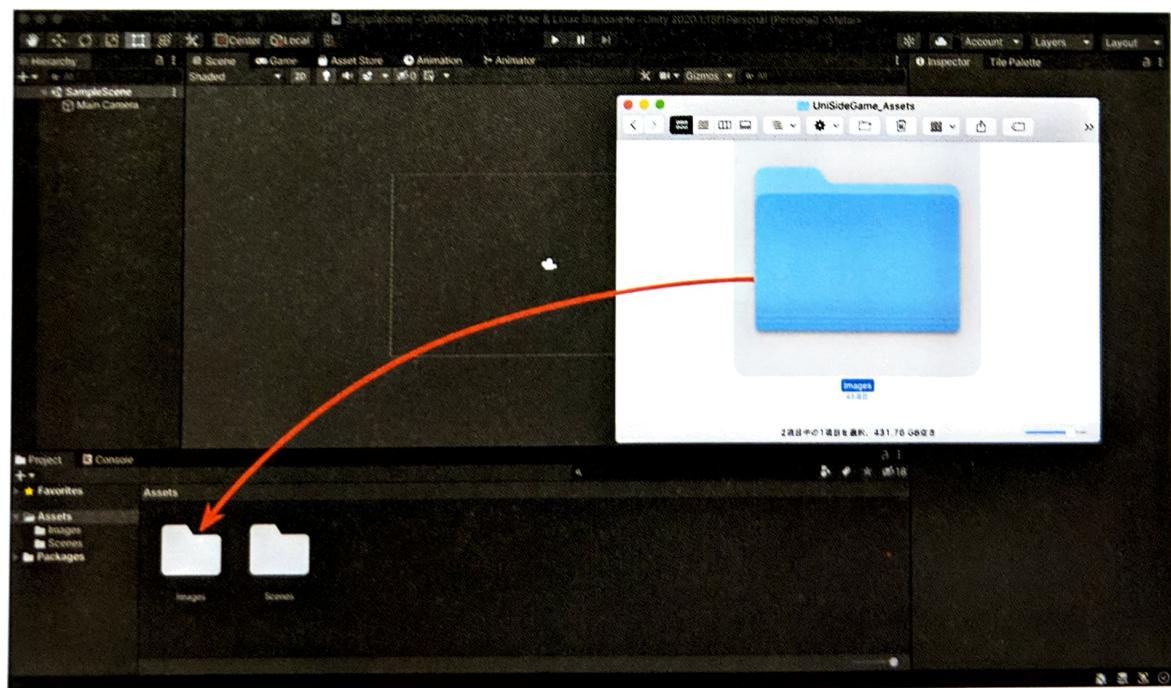
2.2

ゲーム画面を作ろう

画像アセットをプロジェクトに登録しよう

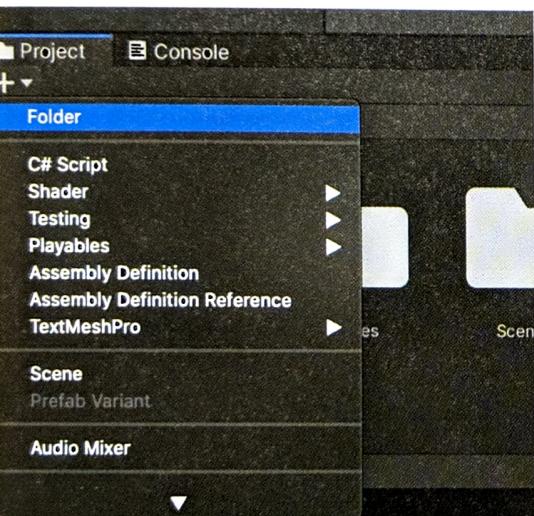
サイドビューゲーム用の画像をUnityに登録します。ダウンロードされた圧縮ファイルを解凍すると、UniSideGame_Assetsというフォルダーがあります。その中で画像の入ったImagesフォルダーを選択し、プロジェクトビューのAssetsフォルダーにドラッグ&ドロップしてください。画像が、Unityで使用可能な**画像アセット**として登録されます。

参照 → 「アセット (Asset)」 20 ページ



Unityでは、パソコンと同じように、プロジェクトビューの中にいくつもフォルダーを作りデータを整理することができます。これからいろいろなデータを増やしていくことになりますが、全部を同じフォルダーに入れてしまうとどこに何があるかわかりづらくなります。ゲームを作るうえではそれでも特に問題はないのですが、ここではサブフォルダーを作ってデータを整理していくことにしましょう。

Assets フォルダーの下に、データを整理するためのフォルダーを作りデータを分類しておきましょう。Assets フォルダーを選択して、[+] ボタンを押して、一番上の [Folder] を選択してください。すると、Assets の下にフォルダーができます。フォルダーナンバーは適当なものに変更しておきましょう。作ったフォルダーに各データをドラッグ&ドロップすることで移動できます。

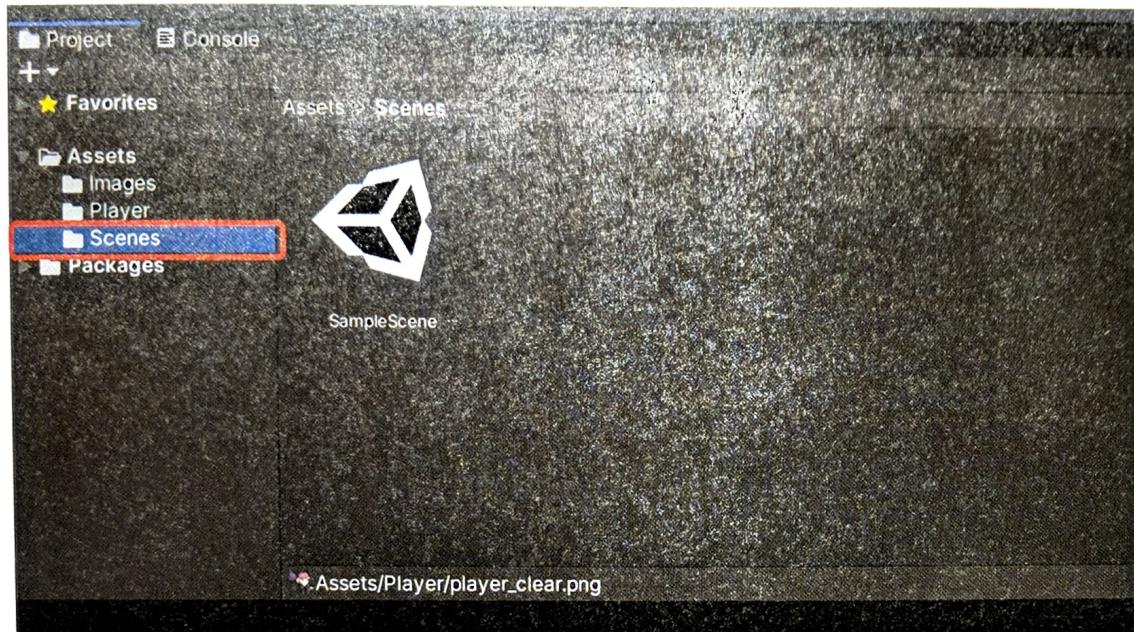


それでは、他にもフォルダーをいくつか作って、整理できるようにしておきましょう。もちろん「このようにしなければならない」ということではありませんので、自分のわかりやすいようにしてみてください。

まず Player フォルダーを作ります。プレイヤーキャラクターに関するデータをここに格納していきます。手始めにプレイヤーの画像を移動させておきましょう。今後プレイヤーに関連するデータは Player フォルダーに追加するようにしていきます。

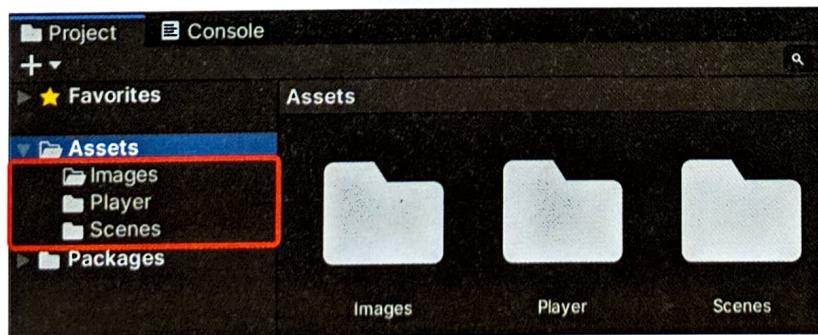


Scenes フォルダーは最初からあったフォルダーでしたね。ここにはこれから追加するシーンデータを入れていきます。なお、シーンについてはこのあと説明します。



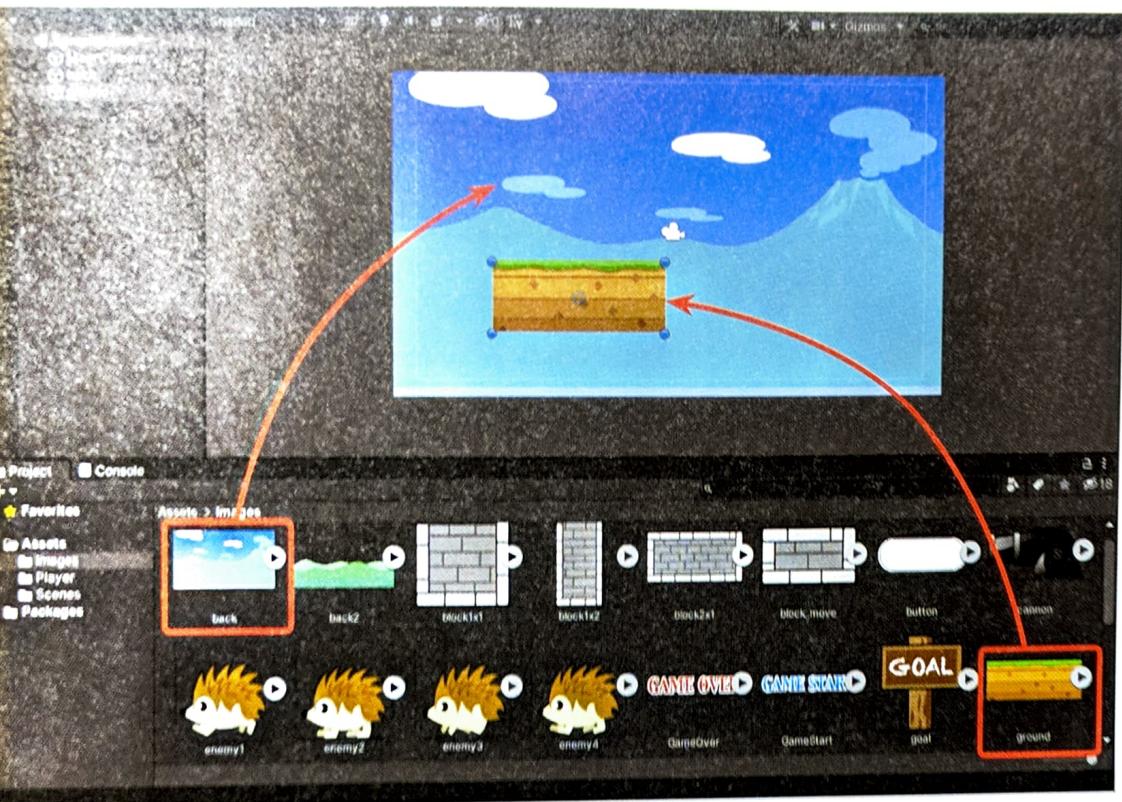
ここまでで、Assets フォルダーの下に以下のようなフォルダーができました。それぞれの使い方とあわせてまとめておきましょう。

- Images : ゲームで使用する画像（画像アセット）を入れるフォルダー
- Player : プレイヤーに関するファイルやデータを入れるフォルダー
- Scenes : シーンファイルを入れるフォルダー

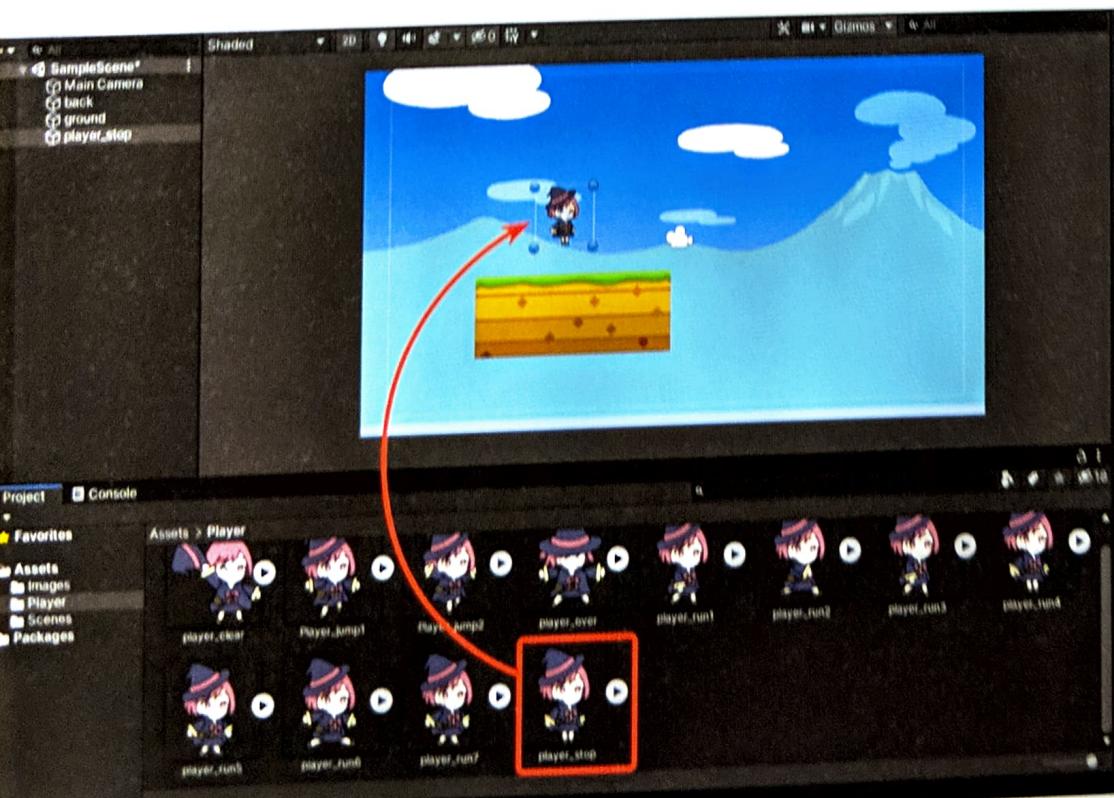


画像アセットを使ってゲーム画面を作ろう

では、さっそくゲーム画面を作ってみましょう。まずはプロジェクトビューにある背景画像「back」、地面画像「ground」をシーンビューにドラッグ＆ドロップしましょう。これでゲーム画面に地面の絵が配置されます。



次にプレイヤーキャラクターを配置します。「player_stop」をシーンビューにドラッグ＆ドロップしましょう。その際、それぞれの位置は適当に調整してみてください。



すると、図のよう画像がシーンビューに配置されます。またヒエラルキービューにも同じ名前の項目が追加されていますね。シーンビューとヒエラルキービューには常に同じものが存在していることになります。

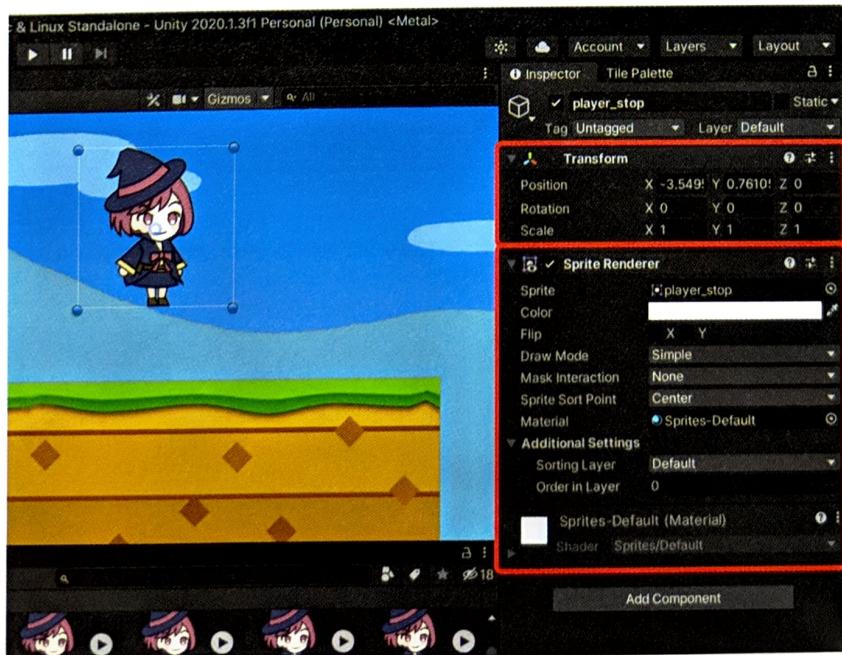
「Scene」を選択するのを忘れないように

ゲームビューが選択されると配置できません。ドラッグ&ドロップする前にシーンビューのタブが「Scene」になっているかを確認してください。

またもし、このときに地面やプレイヤーキャラが見えなくなってしまった場合、それは地面やプレイヤーキャラが背景よりも後ろに表示されてしまっているからです。その場合は、後ほど説明するように、インスペクタービューの [Sprite Renderer] にある「Order in Layer」という数字を0以上に設定してください。

参照 → 「表示の優先順位を知ろう」 [33 ページ](#)

シーンビューにゲームオブジェクトを配置して選択すると、インスペクタービューの表示が変わります。これがゲームオブジェクトにくっついているコンポーネントです。画像を配置した直後のゲームオブジェクトにはTransform(トランスフォーム)とSprite Renderer(スプライトレンダラー)という2つのコンポーネントがあります。



Transform (トランスフォーム)

Transform (トランスフォーム) は、位置、回転、拡大率を決めるコンポーネントです。以下ののようなパラメーターがあります。いずれもよく使うパラメーターです。覚えておいてください。

◆ Position

シーン上でのゲームオブジェクトの位置をX（横方向）、Y（縦方向）、Z（奥行き）の座標で表しています。ここを書き換えることでゲームオブジェクトの位置を変更することができます。

◆ Rotation

ゲームオブジェクトのXYZの各軸の方向を中心とした回転を表しています。2Dゲームの場合Z軸（手前から奥に向かう軸）方向を中心とした回転だけを意識すればいいでしょう。ここを書き換えることでゲームオブジェクトの回転角度を変更することができます。

◆ Scale

ゲームオブジェクトのX（横方向）、Y（縦方向）、Z（奥行き）の表示倍率を小数で表しています。ここを書き換えることでゲームオブジェクトの大きさを変更することができます。

Sprite Renderer

Sprite Renderer (スプライトレンダラー) は、画像を表示するコンポーネントです。以下ののようなパラメーターがあります。よく使うパラメーターについて説明しておきましょう。

◆ Sprite

Unityでは画像のことをスプライト(Sprite)と呼びます。つまりこのパラメーターは、ゲームオブジェクトに表示される画像アセットです。ここを変更することでゲームオブジェクトに表示される画像アセットを変更することができます。

◆ Color

描画する場合に利用される色です。基本となる色は白ですが、他の色に変更することで画像の色味を変えることができます。

◆ Flip

左右、上下に反転させることができます。

◆ Sorting Layer

ゲームオブジェクトをレイヤーというグループに分け、表示の優先をグループごとに決めます。

◆ Order in Layer

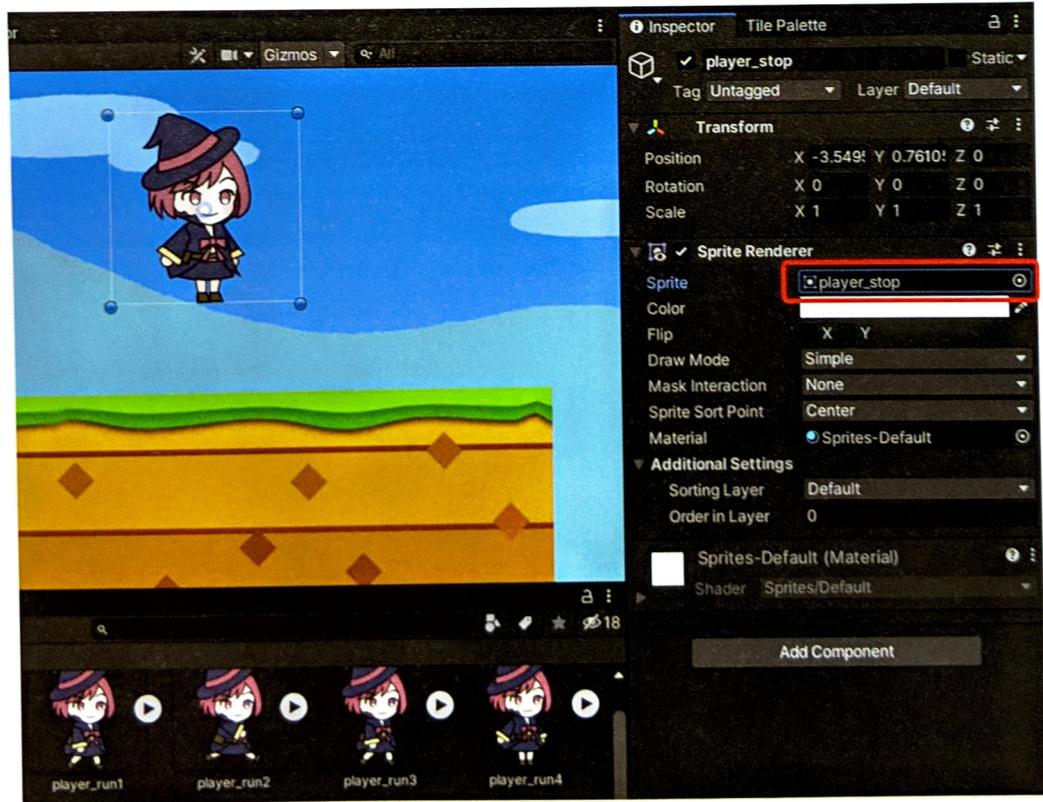
Sorting Layerで分けたレイヤーの中で、ゲームオブジェクトがシーンに表示される順番を決めます。数が大きいほど手前に表示されます。

ゲームオブジェクトの画像を変えてみよう

Sprite Rendererに「Sprite」という項目があります。ここには画像のファイル名が入っているのがわかるでしょうか？

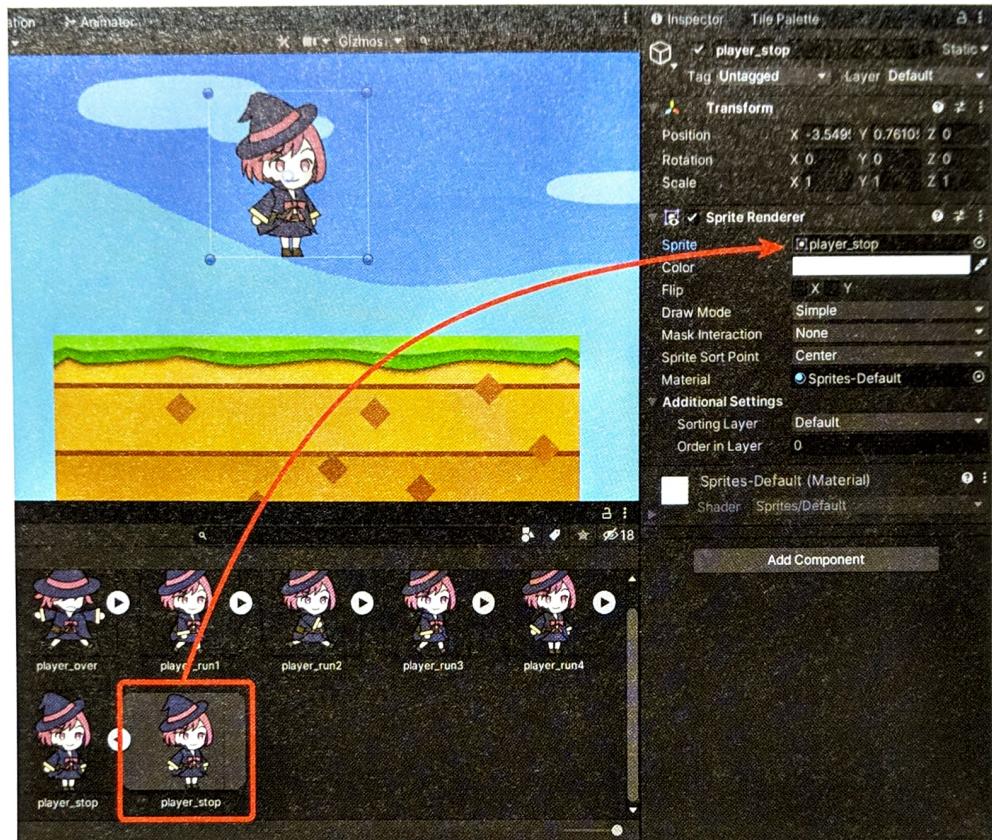
さっそくここをクリックしてみてください。プロジェクトビューで対応している画像アセットがハイライト表示されましたね。これでどのアセットが使われているのかがわかります。

次に選択後、[Delete] キーを押して削除してみてください。「None(Sprite)」（「スプライト（画像）がない」という意味です）と表示され、シーンビューでの表示も消えました。しかし、ヒエラルキービューを見るとゲームオブジェクトは存在しています。つまり、「表示されている画像だけが消えた」というわけです。



それでは、再び画像を設定してみましょう。ゲームオブジェクトを選択してください。画像が消えているので、ヒエラルキーから選択するのがいいかもしれません。

その状態でプロジェクトビューの画像を選択して、「Sprite」の右のテキストボックスにドラッグ&ドロップしてください。

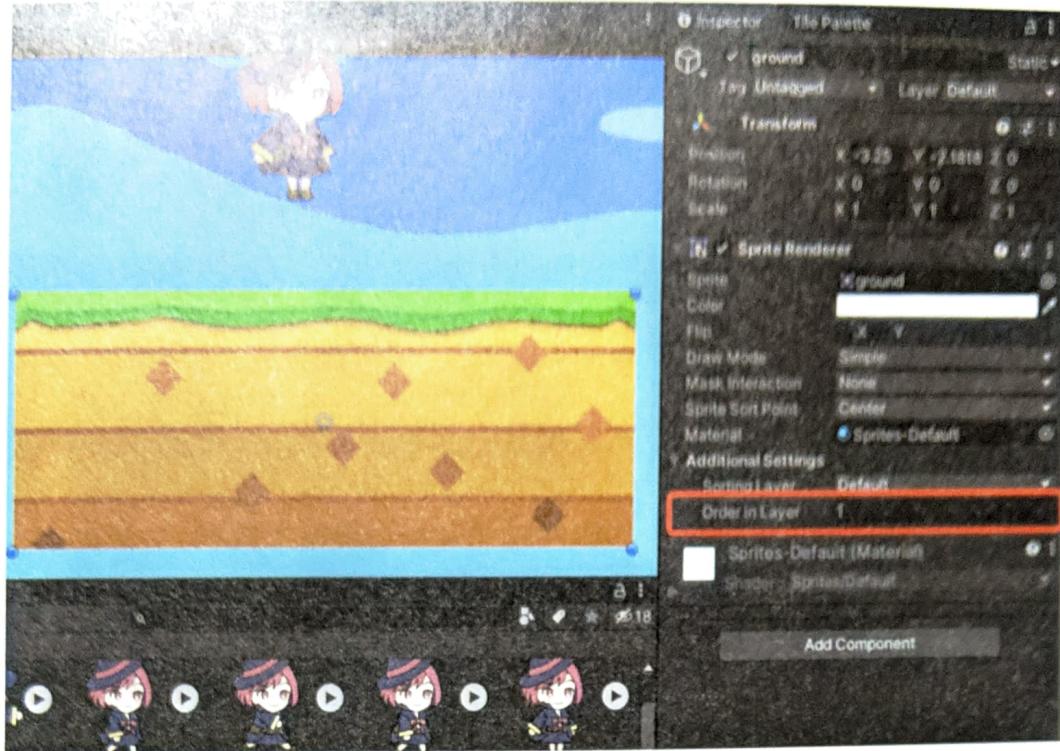


このとき、気を付けてほしいのは、一度選択したら画像を離さないことです。離してしまうとゲームオブジェクトではなく、その画像が選択されてしまい、インスペクタービューの表示が変わってしまうからです。慣れると簡単なのですが、最初のうちはなかなかうまくいかないかもしれません。この操作はUnityでは非常によく行う操作なので、確実にできるようにしておきましょう。

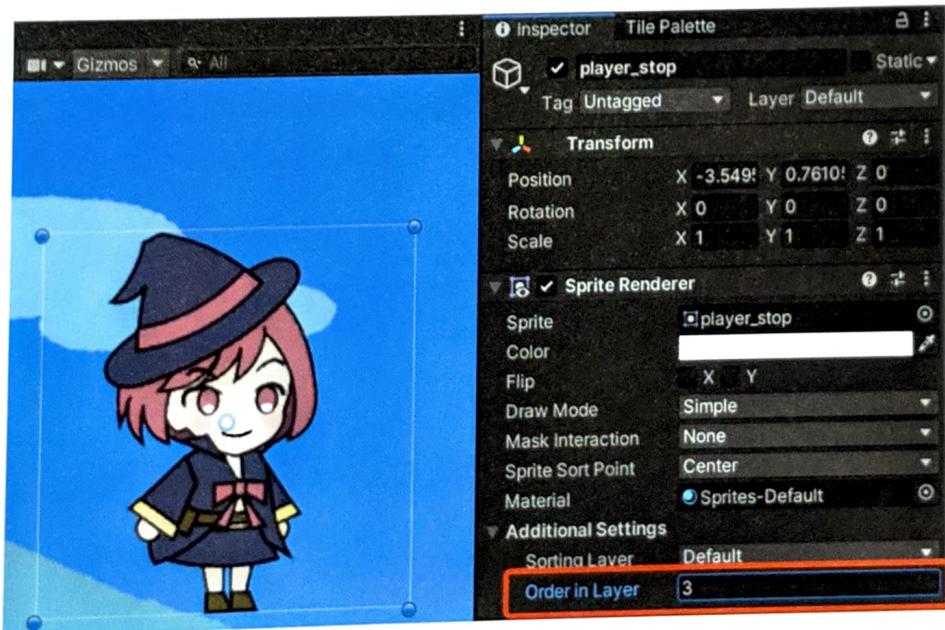
他にも、テキストボックス右端にある小さな丸いボタンを押せば画像アセットの選択リストが表示されるので、そこから選ぶこともできます。

表示の優先順位を知ろう

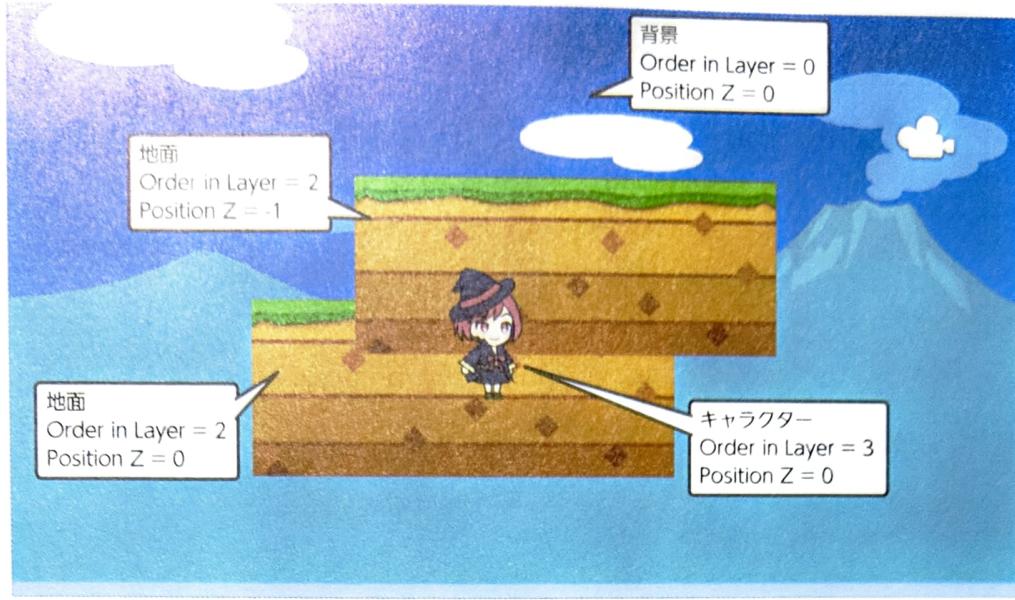
地面やキャラクターなどは必ず背景の上に表示されてほしいので、優先順位を意図的に変更する必要があります。それにはSprite Rendererコンポーネントの「Order in Layer」を大きい数字に変更します。



地面とプレイヤーキャラクターの表示優先を背景よりも上にするために、Sprite Rendererコンポーネントの「Order in Layer」に「0」よりも大きい数字を設定してください（ここでではプレイヤーキャラクターは「3」に、地面は「2」に設定しておきます）。



これで表示画面の優先は、「キャラクター > 地面 > 背景」の順番になりました。



「Order in Layer」の値が同じ場合、表示優先は「Position Z」の値（小さいほど手前に表示）で決まります。



ゲーム画面の位置を表す座標

ゲーム画面上の位置はXとYの座標で表されます。画面中央がX=0、Y=0です。そこから右方向にいくとXがプラスに、左方向だとマイナスになります。Yは上方向がプラス、下方向がマイナスです。

これはゲーム画面の位置と向きを表す座標とベクトルが関わっています。詳しくは以下のコラムを見てください。

参照 → 「座標とベクトル」 80 ページ

この状態で一度ゲームとして実行してみましょう。ツールバーの実行ボタンを押せば、シンビューアがゲームビューに切り替わり、ゲームが実行されます。



もう一度実行ボタンを押せばゲーム実行は終了してシーンビューに戻ります。まだこの段階では画像を置いているだけなので、画面に動きは何もありませんが、「画像を表示する」だけのゲームができ上がりました。

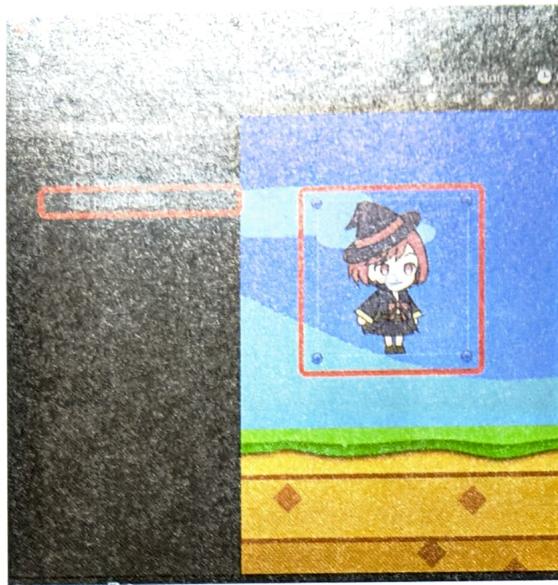
Unity 画面の操作方法を覚えよう

それではここで、シーン画面やゲームオブジェクトの操作方法を覚えておきましょう。

◆ ゲームオブジェクトを選択する



シーンに配置したゲームオブジェクトを選択するには2つの方法があります。



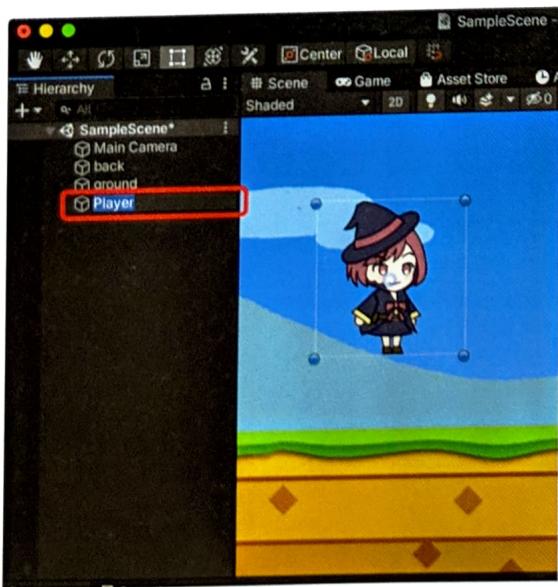
まず1つ目はシーンビューに見えているゲームオブジェクトを直接マウスで選択することです。もう1つはヒエラルキービューのリストから選択するやり方です。シーンに配置されているゲームオブジェクトが増えてくるとシーンビューからは選択しにくくなってしまいます。ヒエラルキービューなら確実に選択することができます。状況に合わせて使い分けましょう。

◆ ゲームオブジェクトの名前を変える

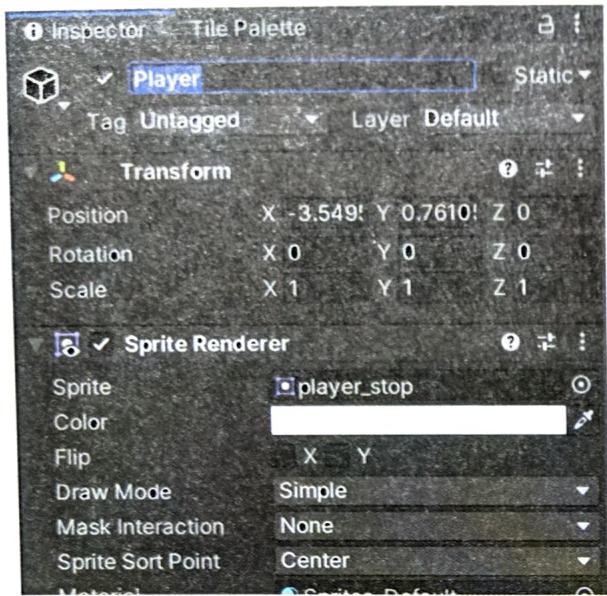


シーンに配置したゲームオブジェクトには名前が付きます。画像をドラッグ&ドロップしてゲームオブジェクトを作った場合、名前は画像ファイル名になっています。

シーン上でゲームオブジェクトは名前で区別されるため、名前はとても重要です。名前を変更する方法を覚えておきましょう。名前を変更するには2つのやり方があります。



ヒエラルキービューのリストから選択した状態で [Return] キーを押します。名前のテキストが選択されて編集状態になるので、名前を書き換えて [Return] キーを押すことで変更が終了します。



もう1つの方法も紹介しておきましょう。インスペクタービューを見ると、一番上に名前が表示されています。ここを書き換えることでも名前を変更することができます。

ここではプレイヤーキャラクターの名前を「Player」に変更しておいてください。



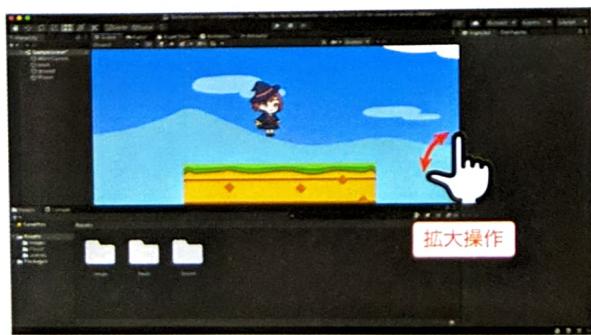
アルファベットの大文字と小文字

アルファベットの大文字と小文字は区別されます。ここでは最初の「P」は大文字で付けています。間違えずに「Player」と設定しておいてください。

画面を自由に移動しよう

ゲーム画面を編集していると、画面の一部を拡大したり表示する位置を変えたりする必要が出てきます。そのやり方を説明しましょう。

画面を拡大縮小して表示するにはマウスホイール（またはタッチパッド）を操作します。マウスホイールやタッチパッドを上に操作すると画面が縮小され、下に操作すると拡大されます（Macの場合はトラックパッドやマウスの設定で拡大と縮小が逆になることがあります）。



ツールバーアイコンでの操作を覚えよう

◆ ハンドツール



画面をドラッグすればシーン画面をスクロールすることができます。

他のツールを選択している場合でも、[Option] キー (Windows では [Alt] キー) を押せばその間はハンドツールに切り替わります。

シーン画面をスクロールするにはツールバーの **ハンドツール** を選択し

◆ 平行移動ツール



矢印をつかむと縦方向 (Y 軸方向) に、赤の矢印をつかむと横方向 (X 軸方向) に平行移動させることができます。また中央にマウスカーソルを持っていけば黄色い四角形が表示され、それをつかめば縦横に移動させることができます。

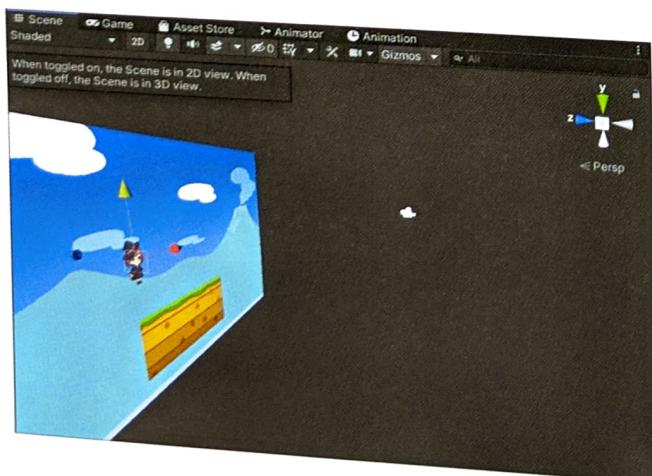
選択されたゲームオブジェクトに緑と赤の矢印が表示されます。緑の

◆ 回転ツール



重の球体が表示されます。この球をつかむことで各軸に対して回転させることができます。この本ではシーンを 2D 表示しているので、縦横のラインと円のように見えますが、シーンビュー上部にある「2D」をオフにしてみてください。シーンが 3D 表示されます。これで 3D 軸と球体が確認できると思います。

選択されたゲームオブジェクトに緑 (Y 軸)、赤 (X 軸)、青 (Z 軸) の三



回転と軸は慣れるまではわかりにくいかもしれませんね。ただ、2D ゲームの場合は Z 軸以外の回転を気にする必要はありません。

◆ スケール変更ツール



選択されたゲームオブジェクトの先端に四角形が付いた緑と赤のラインが表示されます。緑のラインをつかんで縦方向に動かすと縦方向(Y軸)に、赤のラインをつかんで横方向(X軸)に動かすとそれぞれの方向に拡大縮小させることができます。

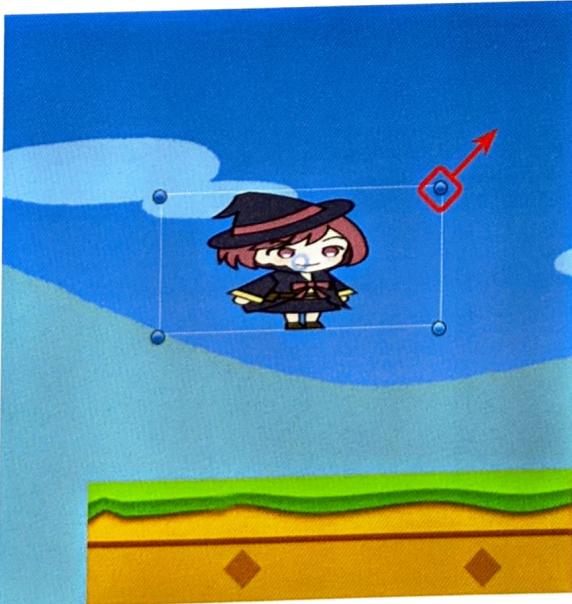


また中央に黄色い四角形が表示されており、それをつかめば縦横比を固定したまま拡大縮小させることができます。

◆ 矩形ツール



選択されたゲームオブジェクトに白い枠と、四隅に丸いハンドルが表示されます。枠内をつかむことで、ゲームオブジェクトを縦横に自由に移動させることができます(平行移動の黄色い四角形と同じです)。



四隅のハンドルをつかんでドラッグすることで、ゲームオブジェクトの矩形サイズを自由に変えることができます。

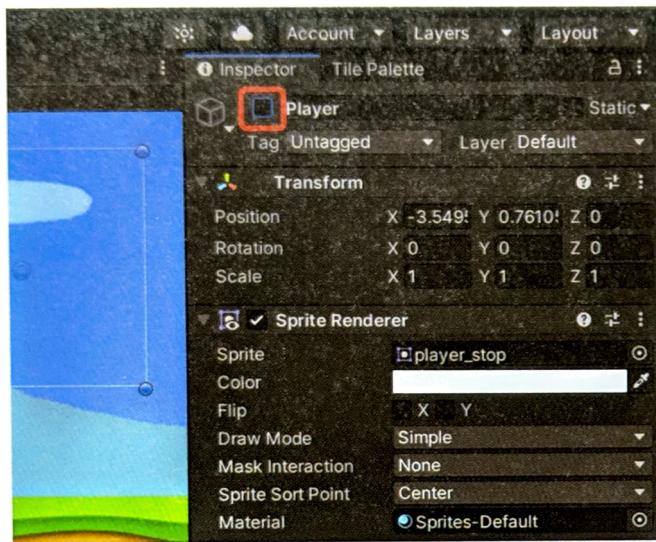
◆ 移動・スケール・回転ツール



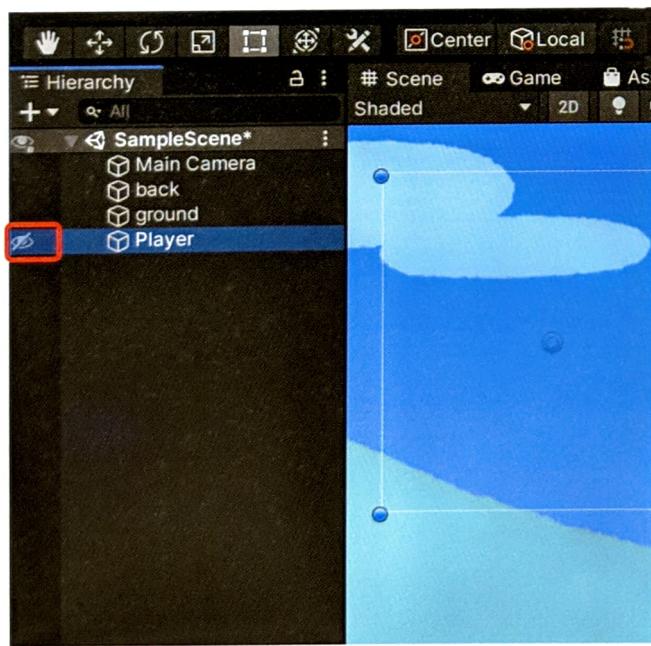
移動、回転、スケール変更を一度に行なうことができます。3つのツール

で使用した操作UIが表示されるので、それぞれを使ってゲームオブジェクトを操作することができます。

ゲームオブジェクトの無効化と非表示を使おう



インスペクタービューの名前の前にチェックボックスがありますが、ここを外しておくとそのゲームオブジェクトが無効になります。

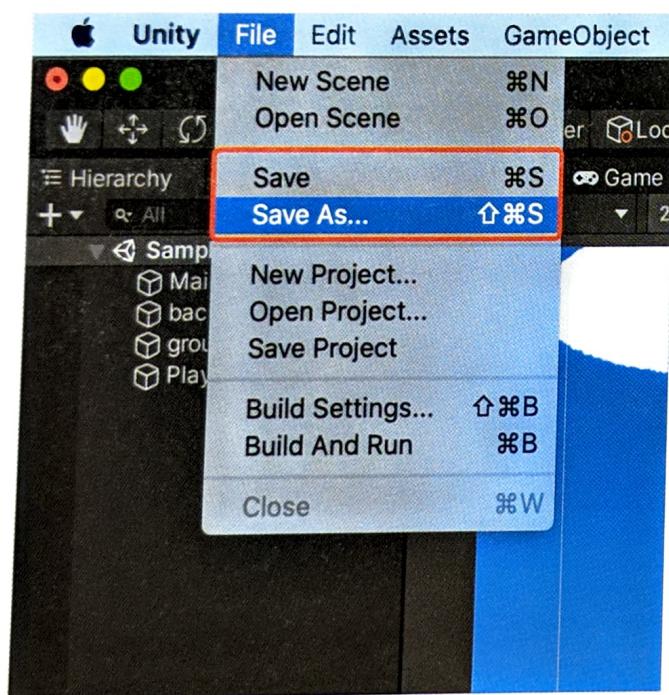


ヒエラルキービューの左端にマウスカーソルを合わせると目のマークが表示されます。これをクリックするとそのゲームオブジェクトがシーンから非表示になります。ここで隠したゲームオブジェクトはゲーム実行中には表示されるので、あくまで編集中に一時的に隠すときに使える機能です。

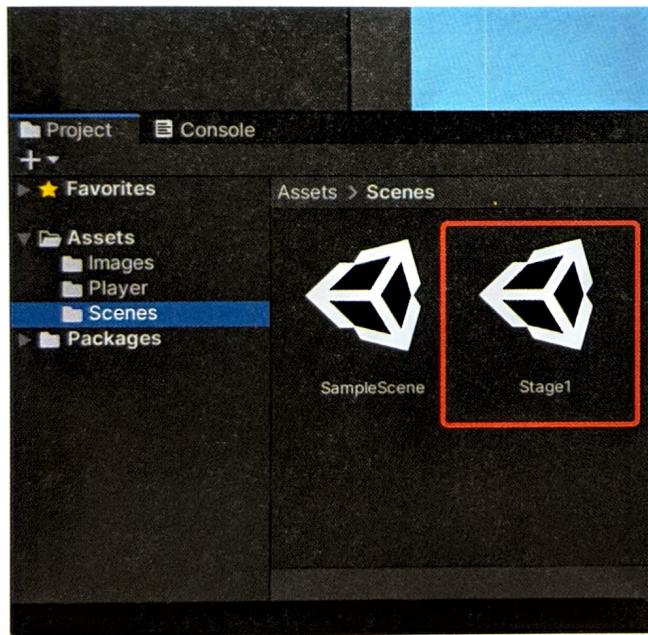
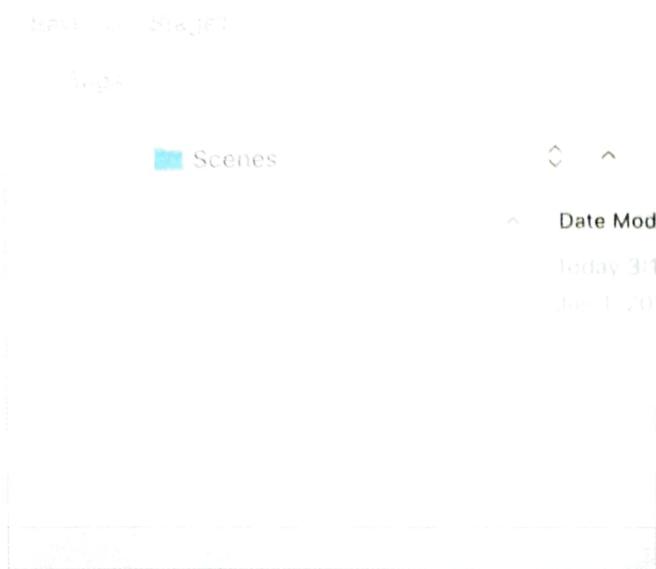


ヒエラルキーの左の、目のマーク右どなりにある指先マークをクリックするとそのゲームオブジェクトがシーン上で選択不可になります。表示されたまま選択不可になるので、重なり合ったゲームオブジェクトを操作するときに便利です。

ゲーム画面(シーン)を保存しよう



シーンを編集したあとは保存するようにしましょう。意識的に保存しないと、もし途中でUnityがクラッシュしてしまったときに、今までの作業が無駄になってしまいます。定期的に保存する癖を付けておきましょう。Macはキーボードの [コマンド + S]、Windowsは [Ctrl + S] で上書き保存できます。

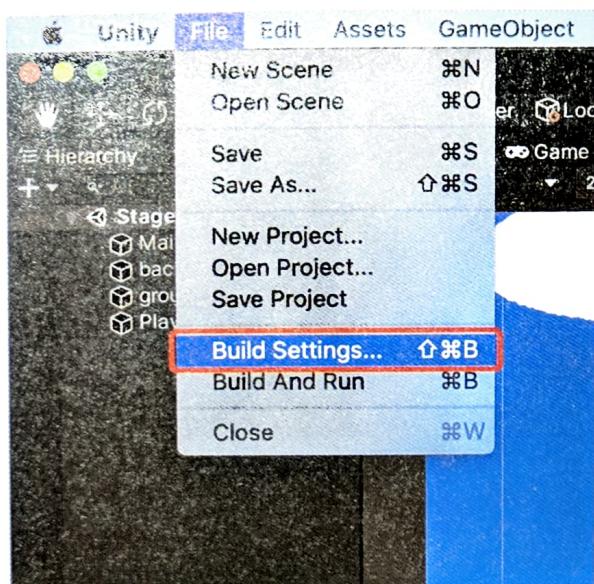


最初に作られたシーンは SampleScene という名前で、Assets フォルダーの中にある、Scenes フォルダーに保存されています。[File] メニューから [Save] を選択すれば 現在編集中のシーンが上書き保存されます。

ここではゲームの最初のステージを作る想定なので、[Save AS…]（別名で保存）を選択して、「Stage1」という名前で Scenes フォルダーに保存しておきましょう。

シーンを保存すると、図のように アイコンが表示されて、ファイル化されます。

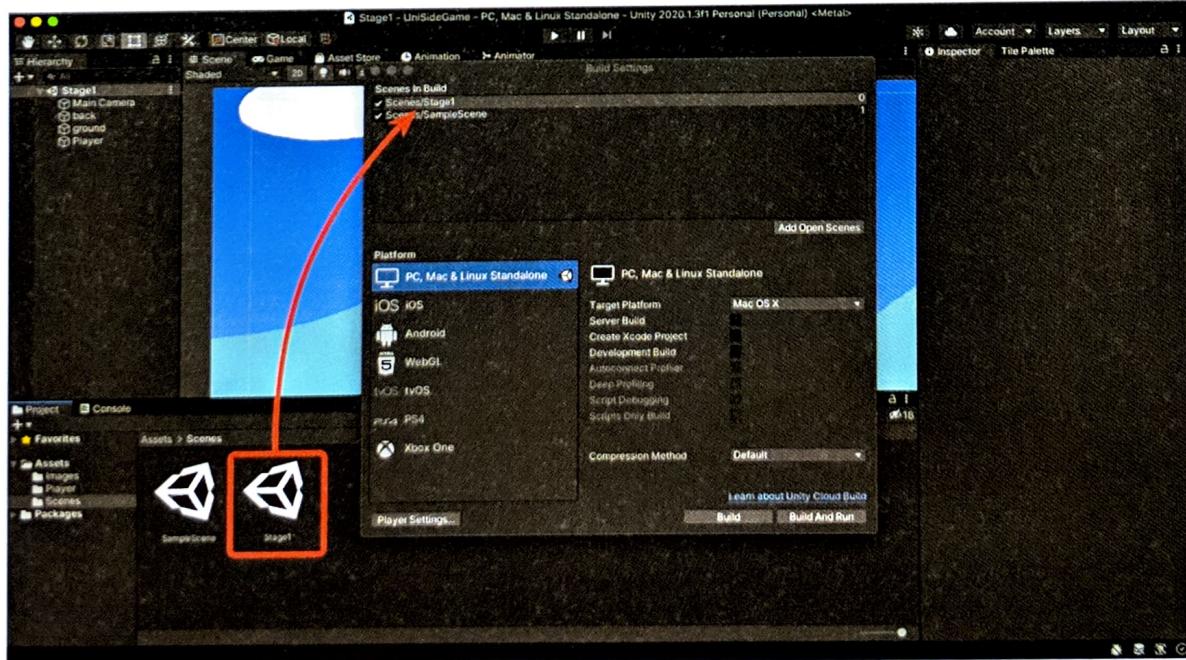
シーンを Scenes In Build に登録しよう



シーンはファイルとして追加しただけではまだ使うことができません。Unityで読み込むシーンはビルドに登録しておく必要があります。

[File] メニューから [Build Settings...] を選択してください。

「Build Settings」というウィンドウが開くので、保存したシーンファイルを「Scenes In Build」という枠内にドラッグ&ドロップして追加してください。これで新しいシーンが使えるようになります。



このBuild Settings ウィンドウは、Unityで作ったゲームをさまざまな環境に書き出す設定を行うためのウィンドウです。

ゲームの書き出しについては本の後半部分で説明します。



Build Settingsへの追加を忘れないように！

シーンを作ったらBuild Settingsに追加するようにしましょう。これを忘るとゲームの中でシーンが読み込めません。



2.3 プレイヤーキャラクターを作ろう

次に、シーンに配置したキャラクターの画像を「ゲームで操作できるプレイヤーキャラクター」にしていきましょう。

画像アセットからゲームキャラクターを作ろう

まずプレイヤーキャラクターを作っていきましょう。プレイヤーキャラクターとして以下のようないくつかのアニメーションパターンを用意しています。

◆ 待機



停止状態のパターンです。移動していないときはこの画像を表示します。先ほどはこの画像アセットを使ってキャラクターを配置しましたね。

◆ 移動



移動中にループで動くアニメーションパターンです。今回は7コマでアニメーションするようにサンプルのパターンを作っています。

◆ ジャンプ中



ジャンプしているときのパターンです。2コマのアニメーションとして作ります。

◆ ゴール



ゴールに到達したときのパターンです。1コマだけのアニメーションとして作ります。

◆ ゲームオーバー



ゲームオーバー時のパターンです。アニメーションとしては1コマだけ作りますが、ステージから消えていく動きをスクリプトで付けることにしましょう。



最初に、キャラクター画像にピボット(Pivot)を設定しましょう。ピボットとは画像の基準になる点で、通常は画像の中心が基準点になっています。画像をゲームオブジェクトとしてシーンに配置した場合、配置位置や拡大縮小、回転などの変形をさせた場合にピボットがその原点になります。

今回は画像の中央下をピボットにします。「キャラクターの足元を基準点にする」ということですね。これはジャンプさせるときの基準点を足元にしたいからです。詳しくは後ほど説明します。

プロジェクトビューの画像を選択するとインスペクタービューに情報が表示されるので、「Pivot」のプルダウンメニューから「Bottom」(中央下)を選択してください。右下の[Apply]ボタンを押せば、変更が反映されます。

ピボットは、キャラクター画像すべてに設定しておきましょう。



ピボットの設定と位置

ゲームオブジェクトの原点、つまりピボットは通常その中央にあります。ゲームオブジェクトはこの点を基準にして移動や変形が行われます。



ここで例として、次のように配置されたゲームオブジェクトがあったとします。



ピボットはCenter（中央）なので、このゲームオブジェクトを2倍に拡大するとこのようになります。画像の中央から大きくなっているので、地面にめり込んで見えますね。



そこでピボットをBottom（中央下）にしてゲームオブジェクトを2倍に拡大すると、このようになります。

地面に立っているキャラが巨大化するような場合はピボットはBottomのほうが適切に見えます。回転する場合はピボットはCenterのほうが適切に見えるかもしれません。ゲームオブジェクトの状況に合わせてピボットを設定しましょう。

コンポーネントを追加しよう



シーンに配置したゲームキャラクターにコンポーネントを付けて機能を追加してみましょう。この操作のことをアタッチといいます。

コンポーネントとは、ゲームオブジェクトにいろいろな機能を追加するものでしたね。画像をシーンに追加したときにはすでに、TransformとSprite Rendererというコンポーネントが付いていました。これらはUnityが自動的にアタッチしてくれたものです。

ゲームオブジェクトに重力 (Rigidbody 2D) を付けよう

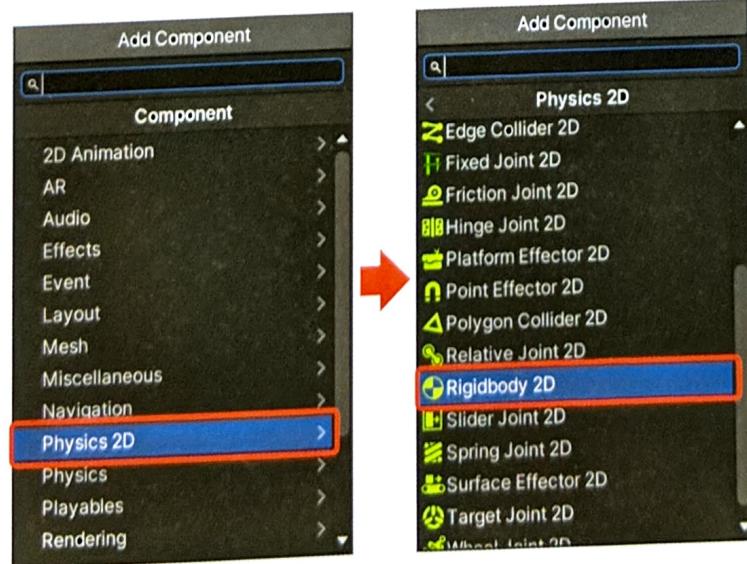
ここでは、まず第2部で作るサイドビューゲームの基礎を作っていきます。サイドビューゲームとは、ゲームの世界を真横から見たアングルのゲームのことです。キャラクターが左右に走るランゲームやジャンプアクションゲームなどがおなじみのジャンルですね。

先ほど、キャラクター用の画像をシーンに配置してゲームキャラの元を作りました。今まではただ画面に絵が貼られているだけです。これをサイドビューゲームのキャラクターにするために、まずは下方向に重力がきくようにしましょう。

重力をきかせるためにはどうすればいいのでしょうか。実はUnityには、重力を発生させるコンポーネントがあるのです。

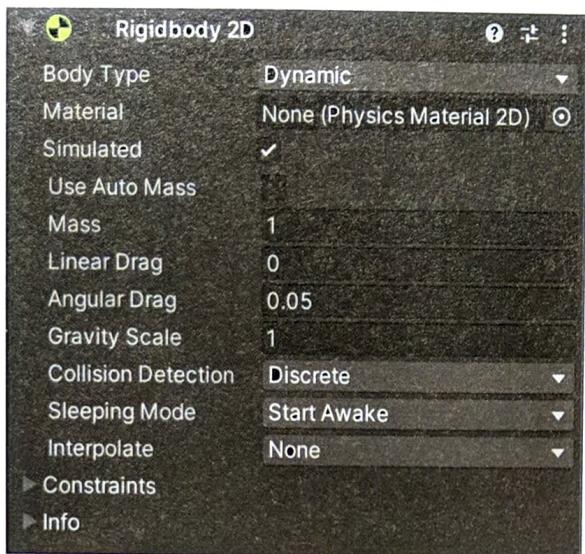


それでは、そのコンポーネントを追加してみましょう。キャラクターを選択したら、インスペクタービューの下にある「Add Component」ボタンをクリックしてください。



プルダウンメニューが開きます。そこから「Physics 2D」という項目を探してください。

選択するとさらにメニューが開きます。次に「Rigidbody 2D」という項目を探して選択してください。



インスペクタービューに「Rigidbody 2D」が追加されましたね。これがゲームオブジェクトに重力を働かせるようにするための、**Rigidbody 2Dコンポーネント**です。Rigidbody 2Dコンポーネントが付けられたゲームオブジェクトはゲーム内で物理法則に従って動くようになります。

なお、コンポーネントの左上にある三角ボタンで項目をタイトルだけ残して、展開／収縮させることができます。覚えておきましょう。

Rigidbody 2Dコンポーネントには以下のようなパラメーターがあります。よく使うものを解説しておきましょう。

◆ Body Type

以下の3つがあります。この選択によりこれ以下の設定が変わってきます。

- Dynamic : 物理シミュレーションの影響を受けて動作するようになります
- Kinematic : 重力や外的的な力の影響を受けません
- Static : 一切の物理シミュレーションの影響を受けなくなります



▶ 物理シミュレーション

コンピューターが計算により、ゲームの中のものを現実世界のもののように動かすことです。落下や衝突した後の動きなどをコンピューターが計算してリアルに動かします。

◆ Material

物体の材質を決定するコンポーネントを設定するパラメーターです。他のものにぶつかったときの跳ね返り具合や滑り具合を設定することができます。

参照 → 「ジャンプ動作を調整しよう (Physics Material 2D の追加)」 116 ページ

◆ Simulated

チェックボックスをオンにすることで物理シミュレーションを有効にします。