

◆ Use Auto Mass

チェックボックスをオンにすることで設定されたコライダー（このあとで説明する「当たり」の設定）の大きさから物体の質量を自動計算します。

◆ Mass

物体の質量を設定します。「Use Auto Mass」を選択している場合は設定できません。

◆ Linear Drag

動いているものがどれくらいの時間で止まるのかを決める数値です。数値が大きいほど移動しにくく、止まりやすくなります。

◆ Angular Drag

回転しているものがどれくらいの時間で回るのかを決める数値です。数値が大きいほど回転しにくくなります。

◆ Gravity Scale

重力を設定できます。1を設定すると、「1G」つまり地球表面と同じ重力となります。

◆ Collision Detection

衝突を検知する方法を設定します。

- Discrete：物体が速く動いた場合、重なったり、すり抜けたりすることができます
- Continuous：重なったりすり抜けたりせず、最初の衝突検知した物体と接触します

◆ Sleeping Mode

コンピューターの計算を軽くするため、一定時間動かない物体は休止状態となり、接触判定などが行われなくなります。ここで休止状態を設定できます。

- Never Sleep：常に物理計算をしている状態で、大きな負荷がかかることがあります
- Start Awake：起動時に休止状態を解除します
- Start Asleep：初期状態で物理計算をしません、衝突により物理計算が開始されます

◆ Interpolate

物理計算の更新での動作補間を選択します。

- None : 補完しません
- Interpolate : 前のフレームにもとづいて動作補完をします
- Extrapolate : 次のフレームにもとづいて動作補完をします

◆ Constraints

動きの制限を設定します。

- Freeze Position : X 軸と Y 軸の動きを制限します。縦横に動かなくなります
- Freeze Rotation : Z 軸で選択して停止します。回転しなくなります

動作を確認しよう



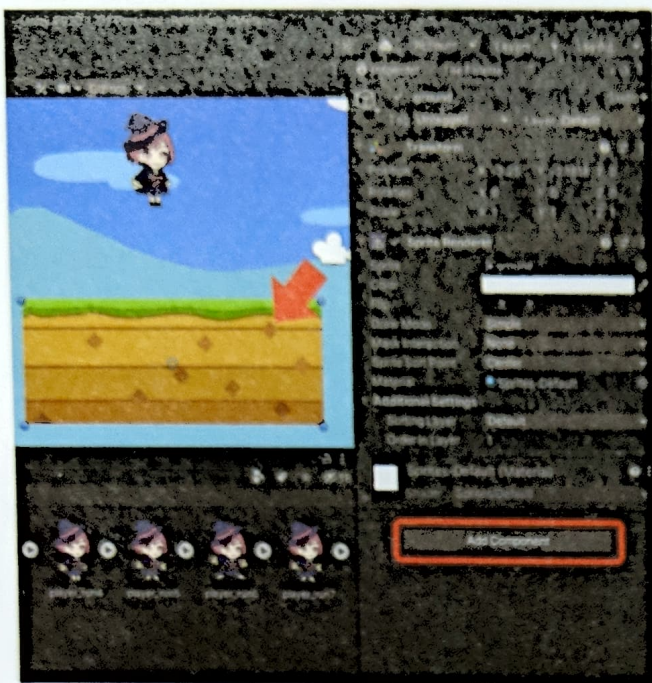
この状態でツールバーの実行ボタンをクリックしてゲームを開始してみてください。

キャラクターが画面下に落下してきましたね。これで今まで動きのなかった画面に何やら動きらしいものがありました。

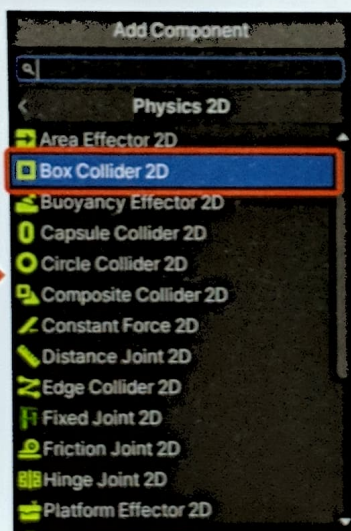
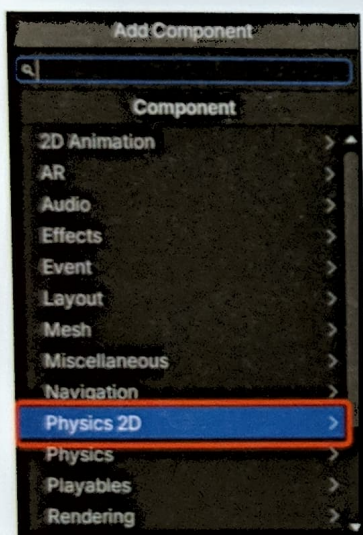
キャラクターは画面下に落ちていってしまいましたが、消えてなくなってしまうわけではありません。ヒエラルキービューを見てください。まだPlayerは存在していますね。そのPlayerを選択して、インスペクタービューのTransformを見てください。

PositionのYの値がマイナスにどんどん大きくなっていっているのがわかります。Yマイナスは下向きの方ですね。つまりキャラクターは延々と下方向に落ち続けているのです。

ゲームオブジェクトに当たりを付けよう



このまま落ちっぱなしではゲームにはなりません。次はプレイヤーキャラクターが地面の上に乗っかれるように当たりを作りましょう。地面画像をキャラクターの真下に来るように調整してみてください。配置された地面のゲームオブジェクトを選択したら、先ほどと同じように、Add Component ボタンをクリックします。



[Physics 2D] → [Box Collider 2D] とメニューをたどって選択してください。

インスペクタービューに **Box Collider 2D** が追加されました。**Collider** (コライダー) はゲームオブジェクトに物理的な接触や接触判定を発生させるコンポーネントです。Box Collider 2D はその名前のとおり、四角形で箱型の当たりをゲームオブジェクトに付けてくれます。

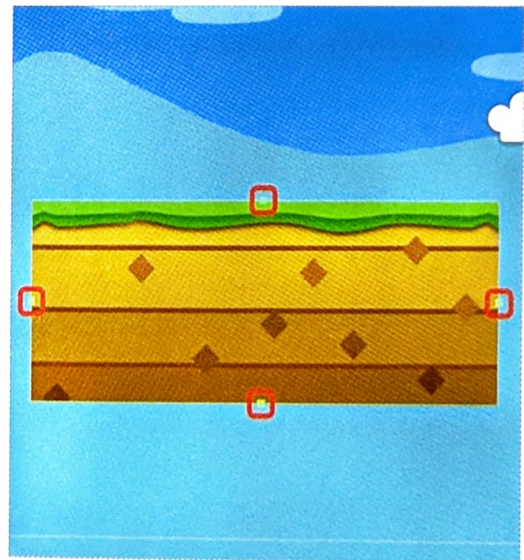
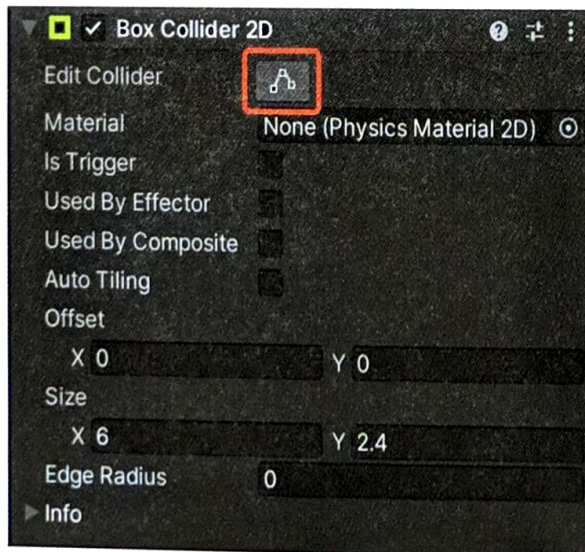


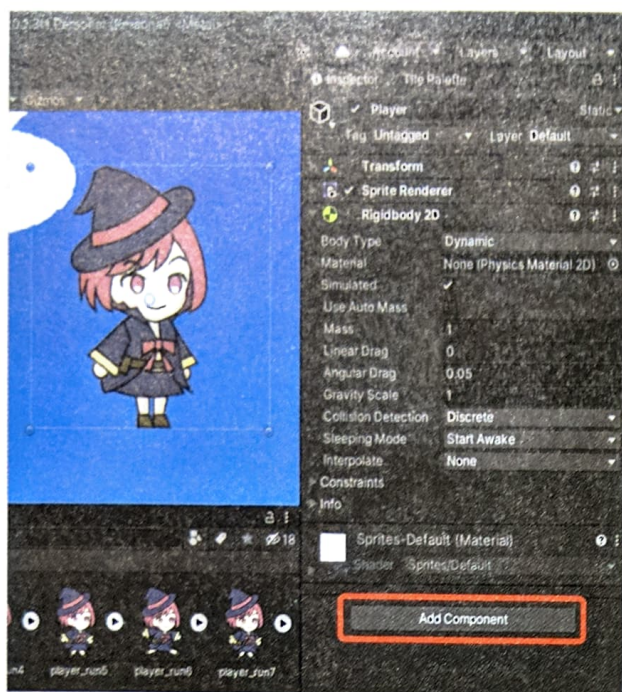
当たり判定とCollider (コライダー)

ゲーム内でゲームオブジェクトを他のゲームオブジェクトに接触させる場合は、このようなColliderというコンポーネントを付ける必要があります。後ほど紹介するように、Colliderは円や楕円、四角形などいろいろな形があります。これらColliderの外枠どうしがぶつかることで当たり判定が行われます。

負荷なども考慮に入れつつ、ゲームオブジェクトをどう使うかによって適切な形を設定しましょう。

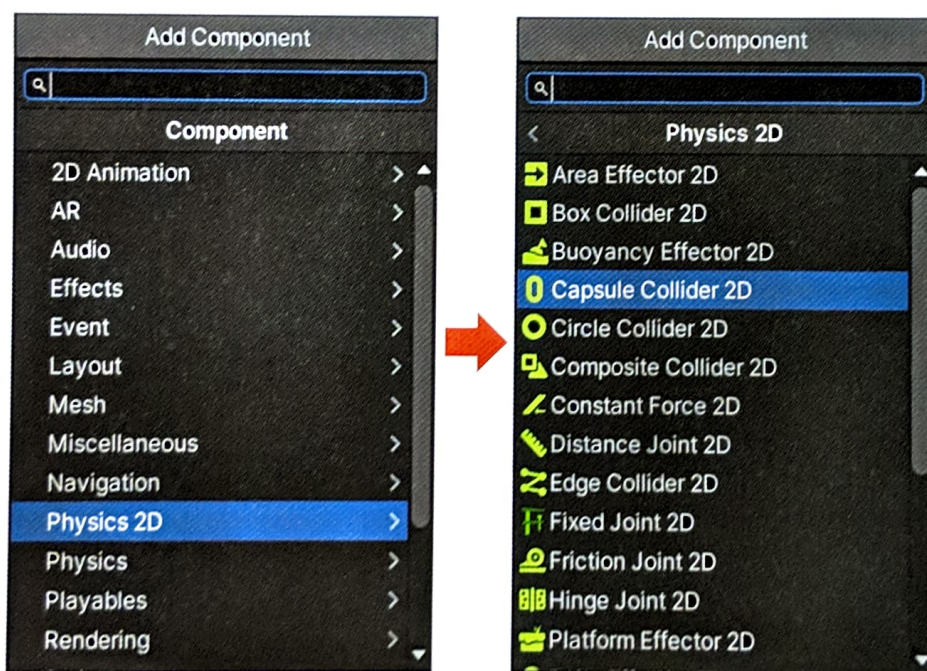
当たりの範囲は [Edit Collider] ボタンで変更できます。ボタンを押すと当たりを表す緑色の枠に四角いハンドルが表示されます。それが当たりの範囲です。このハンドルを操作することで当たりの範囲を変更することができます。

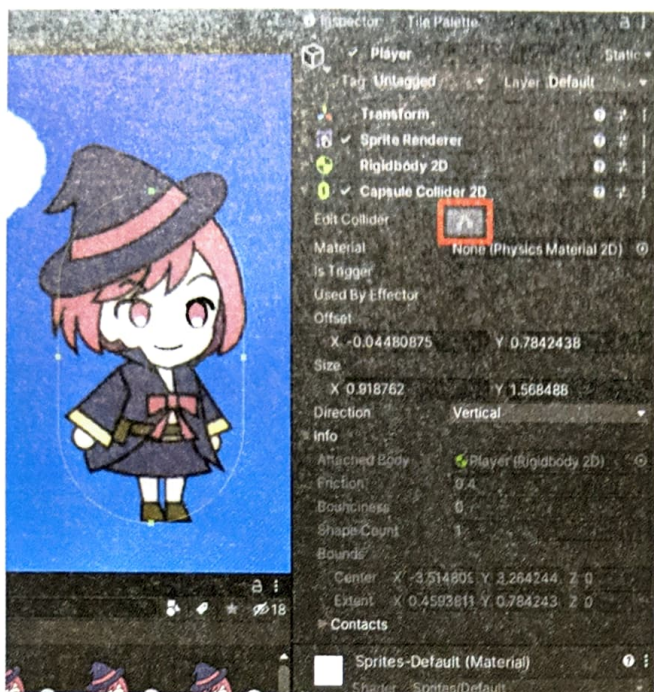




落下するキャラクターをこのブロックに乗せるには、キャラクター側にも当たりを付ける必要があります。同じようにしてキャラクターにも Collider を付けてみましょう。シーンビュー、またはヒエラルキービューで「Player」を選択して [Add Component] ボタンを押してください。

[Physics 2D] → [Capsule Collider 2D] とメニューをたどって選択してください。すると、インスペクタービューに **Capsule Collider 2D** が追加されます。





Edit Colliderボタンで当たりの範囲を調整して、キャラクター周囲に当たりが来るようにしてください。Capsule Collider 2Dは四角形の角が丸くなったカプセル型の当たりです。角がないので引っかかりにくく、移動するキャラクターの当たりに向いています。

ところで、他にもいくつか「○○ Collider 2D」というものがあったのに気が付きませんか？ ここで、Box ColliderやCapsule Collider以外のColliderについても紹介しておきましょう。

ちなみに「2D」と付いていないColliderは3Dゲーム用です。2Dゲームには使えません。

◆ Circle Collider 2D



円形の当たりを付けます。ゲームオブジェクトの中心からの半径で描かれる真円が当たりの範囲となります。比較的軽量の当たりなので、弾丸やたくさん出現する敵キャラなどに向いています。

◆ Edge Collider 2D



ラインを伸ばして引いていくように付けられる当たりです。斜めの坂道や階段、デコボコの地形など、ある程度複雑な形の当たりを作るのに向いています。負荷がそれなりにかかるので、あまり多用はしないほうがいいでしょう。

◆ Polygon Collider 2D



画像の不透明部分を囲むように多角形の当たりが作られます。かなり正確に画像を囲む当たりが作れますが、その分負荷も大きくなります。

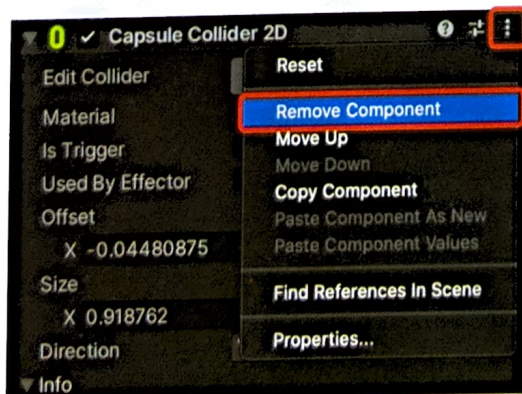
ゲームを実行しよう



ツールバーの実行ボタンを押してゲームを実行してみましょう。落下したキャラクターがその下にある地面に乗って停止しましたね。これでキャラクターと足元のブロックに当たりが付きしました。



アタッチしたコンポーネントの削除



アタッチされたコンポーネントを削除するには、各コンポーネントパネルの右上にあるアイコンから、「Remove Component」を選択します。