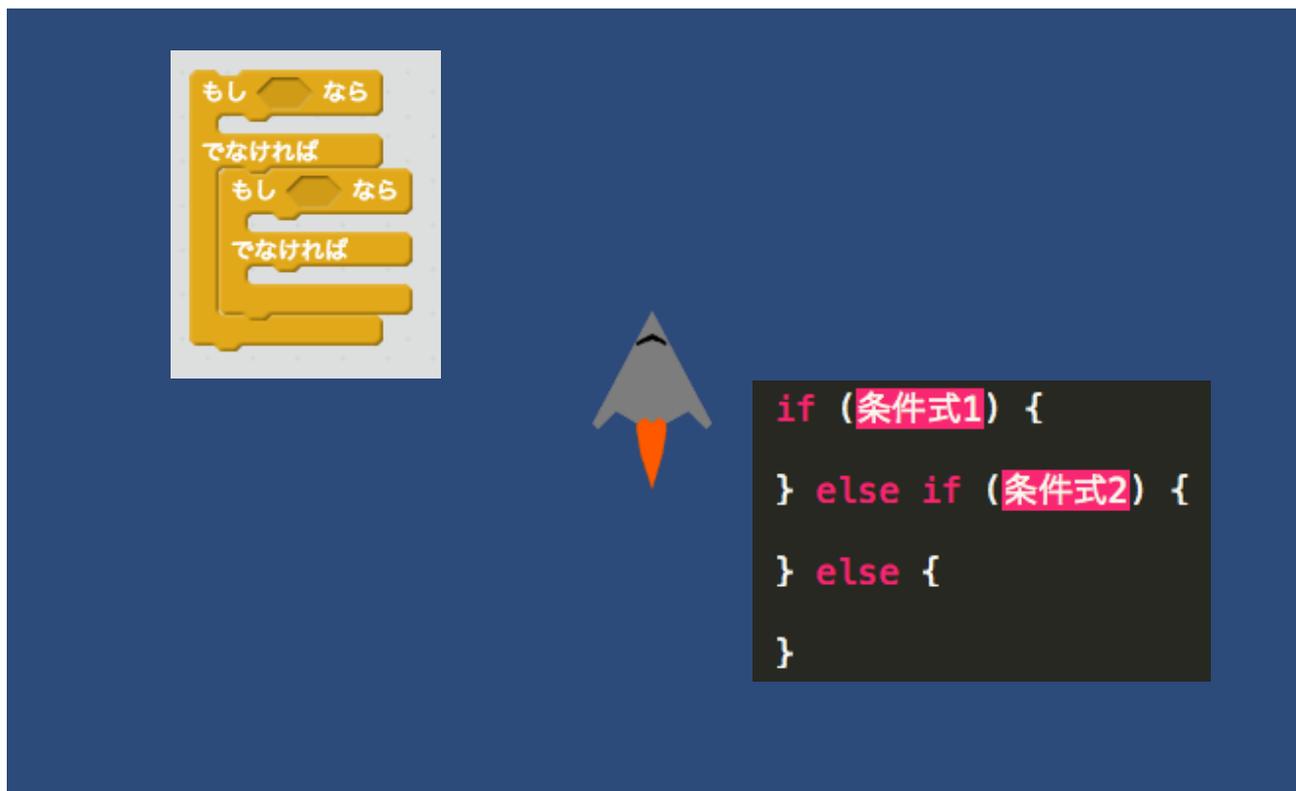


ZeroOne.Scrath2Unity



ZER◇◇NE studio

スクラッチからユニティへの架け橋！

Scratch to Unity誕生

10歩動かす

10 歩動かす

```
transform.position += transform.right * 0.1f;
```

または

```
transform.Translate(Vector3.right * 0.1f);
```

15度回す

15 度回す

```
transform.Rotate (0f, 0f, -15f);
```

15 度回す

```
transform.Rotate (0f, 0f, 15f);
```

90度に向ける

90 度に向ける

```
transform.eulerAngles = new Vector3 (0f, 0f, 0f);
```

-90 度に向ける

```
transform.eulerAngles = new Vector3 (0f, 0f, 180f);
```

0 度に向ける

```
transform.eulerAngles = new Vector3 (0f, 0f, 90f);
```

180 度に向ける

```
transform.eulerAngles = new Vector3 (0f, 0f, 270f);
```

x座標を10、y座標を20にする

x座標を 10 、y座標を 20 にする

```
transform.position = new Vector2 (10f, 20f);
```

または

```
transform.position = new Vector3 (10f, 20f, 0f);
```

x座標を10ずつ変える

x座標を 10 ずつ変える

```
transform.Translate (0.1f, 0f, 0f);
```

x座標を10にする

x座標を 10 にする

```
transform.position = new Vector2 (10f, transform.position.y);
```

y座標を10ずつ変える

y座標を 10 ずつ変える

```
transform.Translate (0f, 0.1f, 0f);
```

y座標を10にする

y座標を 10 にする

```
transform.position = new Vector2 (transform.position.x, 10f);
```

x座標

x座標

```
transform.position.x
```

y座標

y座標

```
transform.position.y
```

表示する

表示する

```
GetComponent<SpriteRenderer> ().enabled = true;
```

隠す

隠す

```
GetComponent<SpriteRenderer> ().enabled = false;
```

大きさを200%にする

大きさを 200 % にする

```
transform.localScale = new Vector2 (2f, 2f);
```

変数を作る



整数の入る変数aを作る

```
int a;
```

小数の入る変数bを作る

```
float b;
```

文字列の入る変数cを作る

```
string c;
```

true(真)またはfalse(偽)の入る変数dを作る

```
bool d;
```

10回繰り返す



```
for (int i = 0; i < 10; i++) {
}
```

ずっと



```
void Update () {
}
```

※最初から記述されているUpdate関数を使います。

もし なら



```
if (条件式) {
}
```

もし なら でなければ



```
if (条件式) {
} else {
}
```

まで繰り返す



```
while (条件式) {  
}
```

Spriteに触れた

Sprite ▾ に触れた

```
void OnCollisionEnter2D(Collision2D col) {
}
```

※衝突を判定する2つのオブジェクトにコライダー（当たり判定）をアタッチします。
2つのオブジェクトの一方に、Rigidbody2Dをアタッチします。

スペースキーが押された

スペース ▾ キーが押された

```
Input.GetKey(KeyCode.Space)
```

※押されている間ずっとtrueになります。

```
Input.GetKeyDown(KeyCode.Space)
```

※押された時に一回だけtrueになります。

マウスが押された

マウスが押された

```
Input.GetMouseButton(0)
```

※押されている間ずっとtrueになります。

```
Input.GetMouseButtonDown(0)
```

※押された時に一回だけtrueになります。

タイマー

タイマー

```
float timer;
```

①変数timerを宣言します。

```
void Update () {
    timer += Time.deltaTime;
}
```

②Updateの中で、Time.deltaTimeを加算し続けます。

タイマーをリセット

タイマーをリセット

```
timer = 0f;
```

x座標 (Sprite)

x座標 (Sprite)

```
GameObject.Find("Sprite").transform.position.x
```

または

①GameObject型の変数「sprite」を作ります。

```
GameObject sprite;
```

②スタート関数で、ヒエラルキーのSpriteオブジェクトの情報を取り出し変数spriteに入れます。

```
void Start () {  
    sprite = GameObject.Find ("Sprite");  
}
```

③x座標は次のようになります。

```
sprite.transform.position.x
```

※y座標も同様になります。

演算

	<code>a = 5 + 6;</code>
	<code>a = 5 - 6;</code>
	<code>a = 5 * 6;</code>
	<code>a = 5 / 6;</code>
	<code>a++;</code>
	<code>a += 5;</code>

1から10までの乱数

	<code>Random.Range (1, 10)</code>
--	-----------------------------------

条件式

	<code>a < 5</code>
	<code>a == 5</code>
	<code>a > 5</code>
	<code>a > 5 && b < 6</code>
	<code>a > 5 b < 6</code>
	<code>a <= 5</code>
	<code>a >= 5</code>
	<code>a != 5</code>

余り

a を b で割った余り

```
a % b
```

リジッドボディーでx方向へ動かす

```
GetComponent<Rigidbody2D> ().velocity = new Vector2 (1f, 0f);
```

リジッドボディーでy方向へ動かす

```
GetComponent<Rigidbody2D> ().velocity = new Vector2 (0f, 1f);
```

クローンを作る

Sprite のクローンを作る

- ①クローンしたいオブジェクトをプレハブ化します。
- ②Materを作り、プレハブを入れる変数cを作ります。

```
public GameObject c;
```

- ③位置(0,0)に、オブジェクトをクローン（インスタンス化：実体化）します。

```
Instantiate (c, new Vector3 (0f, 0f), Quaternion.identity);
```

※クローンしたオブジェクトを操作したい場合

- ②クローンしたオブジェクトを入れる変数objを作っておきます。

```
public GameObject c;
GameObject obj;
```

- ③クローンしたオブジェクトを代入します。

```
obj = Instantiate (c, new Vector3 (0f, 0f), Quaternion.identity);
```

クローンされたとき

クローンされたとき

上のプレハブにスクリプトをアタッチします。

このクローンを削除する

このクローンを削除する

```
Destroy (obj)
```