??????? UI

現状、ゲームが終わるとき、 `Debug.Log` でメッセージを表示しているが、その代わりに Canvas で「ゲームオーバー」を表示し、「さいかい」と「おわり」のボタンを実装しましょう!

??????

???????

まず、ブロックと同様に、画像を分解しないといけない。ただし、行列ではないので、きれいに縦横で分割できない。この場合は、マウスを使い、各スプライトの領域を定義しましょう:

3つのスクリプトを分割したら、「Apply」ボタンを押し、シーンを編集しましょう

Canvas???

すでに作ったゲーム用の Canvas は別目的なので、ゲームオーバー専用の Canvas を新しく作ろう。そして、Image 1つと Image + Button 2つを追加してください:

??????????

GameOver オブジェクトに Animator

コンポーネントを追加し、アニメーションを作成しましょう。ここは好きなように作成しても問題ない

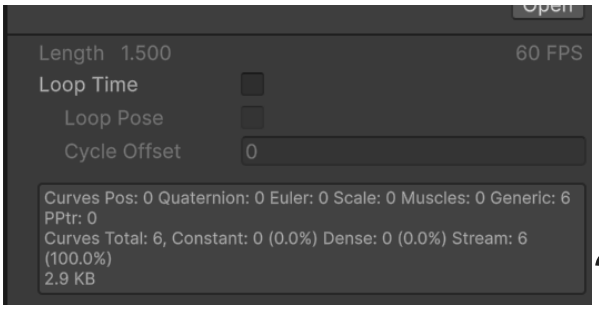
なお、アニメーションを作成した後、無限再生（ループ）を無効化しましょう。

フォルダーでアニメーションのファイルを探し、クリックしたら、

」のチェックボックスからチェックを外してください:

Project

Inspectorで「 Loop Time



?????

GameOverManager □□□□□□□□ Show□□□□□ Hide□□□□□□□□□□□ Show □□□□□□□□□□□□□□□□□□□□□□□

```
// GameOver□□□□□□□□□
public class GameOverManager : MonoBehaviour
{
    // □□□□□□□□□□□□□□□
    [SerializeField]
    private Animator animator;

    // □□
    public void Show()
    {
        // □□□□□□□□□□□□□□
        // □□□□□□□□□□□□□□
        gameObject.SetActive(true);
        animator.Play("ShowGameOver");
    }

    // □□□
    public void Hide()
    {
        animator.StopPlayback();
        gameObject.SetActive(false);
    }
}
```

□□□□ GameManager □□□ ...

```
// □□□□□
public class GameManager : MonoBehaviour
{
```

```
    // ボールを投げる処理
    [SerializeField]
    private BallShooter ballShooter;

    // ゲームの情報を表示できるように
    [SerializeField]
    private UIManager uiManager;

    // ゲームオーバーの表示・非表示用
    [SerializeField]
    private GameOverManager gameOverManager;

    // 後は同じ
}
```

で、ゲームが始まったら、ゲームオーバーを非表示し、ゲームオーバー（またはゲームクリア）の時はまた表示すれば良いでしょう:

```
private void Start()
{
    _lives = 3; // 3個で始まる
    uiManager.SetLives(_lives);

    // ゲームオーバー非表示
    gameOverManager.Hide();
}

// ボール廃止されたときに呼び出される
public void BallDestroyed()
{
    // 残機が減る
    _lives--;
    uiManager.SetLives(_lives);
    if (_lives == 0)
    {
        // ゲームオーバーを表示
        gameOverManager.Show();

        // スクリプトを無効化する
        ballShooter.enabled = false;
```

```
    }
}

// ブロック廃止のとき呼び出される
public void BlockDestroyed()
{
    // 100点を与える
    _score += 100;

    // UIを更新
    uiManager.SetScore(_score);

    _blockCount--;
    if (_blockCount == 0)
    {
        // ゲームオーバーを表示
        gameOverManager.Show();
        ballShooter.enabled = false;
    }
}
```

Unity???

GameOverManager のスクリプトを GameOver にアタッチし、GameManager との紐づけできてたら、実行して、確認しましょう!

??????

ボタン（Button）はを押したら、ゲーム監督 GameManager を呼び出し、ゲームを再開するのか、それともゲーム終了するなのかを実装しましょう。

Button から　GameManager を呼び出すので、メソッドは「　　public」にしましょう!　　それでは、GameManager に新しいメソッドを追加してください:

```
// ゲーム再開ボタンの処理
public void OnGameRestart()
{
    SceneManager.LoadScene("Game");
}

// ゲーム終了ボタンの処理
public void OnGameEnd()
{
    // 独立に実行ファイルの時
    Application.Quit();

    // Unityの中で実行する場合
    EditorApplication.ExitPlaymode();
}
```

Button???

Buttonコンポーネントを見ると「　　OnClick」という項目がある。この項目が「ボタンが押したら何を呼び出すか? 」という設定場所である。「+」ボタンを押し、新しい呼び出しを追加しましょう:

3つの設定できる項目がある:

①

ゲームオブジェクト: ボタンを押したら、このゲームオブジェクトを呼び出す。普通通りに、ヒエラルキーからドラッグドロップで設定ができる。

② 呼び出したいメソッド:　　①で選択したゲームオブジェクトのすべてのコンポーネントとスクリプトを表示され、呼び出したいメソッド

を選択ができる。
③ 引数：メソッドは引数があれば、ここで設定ができる。引数がない場合は何も表示されない。

今回は、「さいかいボタンを押したら、 GameManager オブジェクトの中にある GameManager.OnGameRestart() を呼び出したい」にしたいので、 OnClick はこうなる：

「おわり」ボタンは：

これでもう一度確認しましょう。

Revision #2
Created 2026-05-04 04:43:21 UTC by Menendez Francisco
Updated 2026-05-04 06:28:16 UTC by Menendez Francisco