

タイトル画面

no5

シーンを新しく作りましょう。プロジェクトウィンドウで右クリックからCreate

> Sceneで作成しましょう。



名前をTitleにしましょう。



ダブルクリックしてシーンを開きましょう。今回はボタンを一つ配置して、それ を押したらゲームをスタートできるようにしましょう。ボタンはunity上ではUI として扱われます。ヒエラルキーで右クリックからUI > Buttonでボタンを作成 してください。

	Cut					
	Copy					
	Paste					
	Paste As Child					
	Rename					
	Duplicate					
	Delete					
	Select Children					
	Set as Default Pare	ent				
	Create Empty					
	2D Object		>			
	3D Object		>			
	Effects		>			
	Light		>			
	Audio		>			
·	Video		> >			
	UI		>	Image		
5	UI Toolkit		>	Text - TextMeshPro		
e	Camera			Raw Image		
	Visual Scripting Sc	ene Variables		Panel		
led				Toggle		
anspar	ent			Slider		
Tilema	ip			Scrollbar		
abs				Scroll View		
nes				Button - TextMeshPro		
aes i				Dropdown - TextMeshPro	,	
				Input Field - TextMeshPro)	
				Canvas		
				Event System		

また、その後のポップアップは上を選択してください。読み込みが始まると思う ので少し待ってください。

Material Raycast Target	None (Materia
Raycast Target	
	•
Raycast Padding	
Maskable	~
Image Type	Sliced
: 🗆 🗙	~
	er 1
urces to your project t of your project in the	~
	Color Lint
	≊ Button (Ima
e discovering and be placed in the same	
	Maskable Image Type : X slinces to your project t of your project in the a discovering and be placed in the same

🔻 🛠 Title* 🕴
😚 Main Camera
🔻 🔂 Canvas
🔻 🕎 Button
Text (TMP)
🔂 EventSystem

no5

Cull Transparent Me

🔻 T 🗹 TextMeshPro - Text ((UI) 🛛 🖓 ᅷ :
Text Input	Enable RTL Editor
Start	
Text Style	Normal
Main Settings	
Font Asset	ELiberationSans SDF (TMP_Font Asset) 💿
Material Preset	LiberationSans SDF Material 👻
Font Style	B I U S ab AB SC

画像のようにヒエラルキーからText(TMP)を開いて、インスペクターウィンドウ からテキスト内容をStartに変更してください。次はこのボタンが押されたとき の挙動を作りましょう。



今までと同じようにStartButtonControllerを作りましょう。ここで新しくボタ ンが押されたときに実行される関数を作りましょう。ここでするべき処理は今ま で作ってきたMainSceneにシーンを遷移させることです。シーン遷移を実行する ためには一つ前準備が必要です。左上からFile > Build settingを選択して、 Scenes in buildの部分に今まで作成しているシーンをすべてドラッグアンドド ロップしてください。



これでシーンを変えられます。StartButtonControllerを開いてください。そし て新しく次の関数を書き込んでください。

```
public void OnPush()
{
```

}

この関数をボタンに設定しましょう。まず、StartButtonControllerをButton

につけましょう。

Pixels Per Unit Multiplier	1		
🔻 🖲 🖌 Button		0 ‡	
Interactable	~		
Transition	Color Tint		
Target Graphic	🖾 Button (Image)		
Normal Color			ø
Highlighted Color			ø
Pressed Color			ø
Selected Color			ø
Disabled Color			ø
Color Multiplier	•	- 1	
Fade Duration	0.1		
Navigation	Automatic		
	Visualize		
On Click ()			
List is Empty			
		. _	
🔻 🛱 🖌 Start Button Control	ler (Script)	9 ‡	
	StartButtonController		
Default UI Material I	(Material)	e	• •

その後、インスペクターウィンドウのButtonの下の方にある、On Click()という 部分を探してください。ここの+ボタンを押します。そうしたらこんな風になる と思います。

Or	n Click ()				
	Runtime Only None (Object)	•			
			4	-	

そうしたらNone(Object)となっているところに、すぐ下にある StartButtonControllerをドラッグアンドドロップして、No Functionとなって いるところを画像のようにOnPush関数に変更してください。



プレビューを確認して、ボタンを押してみるとMainSceneに遷移できたと思います。

リザルト画面

次は同じ要領で、ゴール後の画面も作りましょう。また、同じように新しくシー ンを作成して名前をResultSceneにしましょう。



とりあえず画面にゲームクリアの

文字とタイトルに戻るためのボタンを設置しましょう。UIのテキストを作成しま しょう。また、追加でボタンも設置しましょう。

	Paste Paste As Child Rename Duplicate Delete Select Children Set as Default Parent Create Empty 2D Object 3D Object Effects Light Audio	> > > > > > > > > > > > > > > > > > > >		
le	UI UI Toolkit Camera Visual Scripting Scene Variable	> > >	Image Text - TextMeshPro Raw Image Panel	
olatformer	MainScene ResultSce	Title	Toggle Slider Scrollbar Scroll View Button - TextMeshPro Dropdown - TextMeshPro Input Field - TextMeshPro Canvas Event System	
	G	ame C	Clear Title	

画像のように設定しましょう。ではタイトルに戻るボタンを作りましょう。 BackButtonControllerとしましょう。ここはさっきのスタートボタンとほぼ同じ なのでそのまま貼ります。また、On Clickの部分も設定しておいてください。

using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.SceneManagement;

public class BackButtonController : MonoBehaviour

```
no5
```

{

}

```
// Start is called before the first frame update
void Start()
{
    // Update is called once per frame
void Update()
{
    public void OnPush()
    {
        SceneManager.LoadScene("Title");
    }
}
```



クリア後とリスタート

クリアしたらリザルト画面に行くようにしましょう。これもさっきまでとほぼ同 じです。PlayerControllerにクリア後にシーン遷移するようにしましょう。シー ン遷移の為にBuildsettingsにresultSceneを追加しましょう。



できたら次のように書き足してください。

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if(collision.gameObject.tag == "Enemy")
    {
        ContactPoint2D contact_face = collision.contacts[0];
        Debug.Log(contact_face.normal.y);
        if (contact face.normal.y >= 0.7f)
```

no5

```
{
           Destroy(collision.gameObject);
        } else
        {
           Destroy(gameObject);
        }
    } else if(collision.gameObject.tag == "Goal")
    {
        //ゴール検知
        SceneManager.LoadScene("ResultScene");
    }
    if(collision.gameObject.tag == "map")
    {
        is_field = true;
    }
}
```

これで一旦プレビューを確認して、問題無ければ最後に敵に倒されたあと、 リスタートできるようにしましょう。今回はRキーを押したらリスタートという 形にしましょう。ここで、PlayerControllerにその処理を書き込みたくなる所で すが、敵に倒された後ということは、Playerはすでに破壊されているので、 PlayerControllerは動作しません。ので、ここでは新しくオブジェクトとスクリ プトを作りましょう。

ヒエラルキーからCreateEmptyで名前をGameManagerにしましょう。



また、ファイルも新しくつくって、名前をGameManagerにしましょう。歯車マー



そして、ファイルを開いて次のように書き足しましょう。

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.R))
```

	{	
		<pre>SceneManager.LoadScene("MainScene");</pre>
	}	
}		

最後にこのファイルをGameManagerにつけましょう。 これでRキーを押したら もう一度メインシーンが読み込まれて、ゲームが始めからリスタートできます。

カメラ移動

no5

最後にカメラがプレイヤーを追従するようにしましょう。これは、カメラの位置 をプレイヤーと同じになるようにしたらいいです。とりあえずスクリプトを作成 し、名前をCameraControllerとしましょう。そして、PlayerのtagをPlayerに設 定しましょう。

0 Ins	specto	or				
\bigcirc	~	Playe	r			
•	Tag	Play	er			Lay
	-	······				

次にCameraControllerを編集しましょう。内容を次のように書き換えてください。

```
public class CameraController : MonoBehaviour
{
    GameObject player;
    void Start()
    {
        player = GameObject.FindWithTag("Player");
    }
    void Update()
    {
        this.transform.position = new Vector3( player.transform.position.x+3, player.transform.position.y,-10);
    }
}
```

これは、シーン内のPlayerタグを持つオブジェクトを変数playerに代入し、 そのオブジェクトの位置をカメラの位置にするものです。これをMainCameraにつ けることでカメラが追従します。+3の部分はカメラが見やすくなるための調整の 値です。

ステージ作成

これでゲーム自体は完成です。最後にステージを作りましょう。左上のWindowからTilePaletteを開いて地面を作り、敵を設置しましょう。敵はプロジェクトウィンドウからプレハブの元をシーンウィンドウにドラッグアンドドロップすることでコピーを設置できます。



これで各々ステージを作成してください。ゴールはいい感じの所に場所を変えま しょう。

ビルド

no5

これで完成って言われたらなんか違いますよね。ので最後にこのゲームを実行フ ァイルに落とし込みましょう。

ビルドする前に設定しましょう。先ほどと同じようにbuild settingを開いて、 scenes in build内で、titleシーンが一番上に来るように順番を変えましょう。

Scenes/Title
 Scenes/ResultScene
 Scenes/MainScene

ビルドの設定はプラットフォーム以外変えなくてokです。windowsの人はwindows をmacの人はmacを選択してBuildを選択してください。



保存場所は各々で決めてください。ビルドは少し時間がかかるので待ちましょ

う。ビルドが終わったら実行してみましょう。.exeのファイルを実行してくださ

名前 ^	更新日時	種類	サイズ
joken_sample_BurstDebugInformation_D	2025/03/16 2:18	ファイル フォルダー	
📒 joken_sample_Data	2025/03/16 2:19	ファイル フォルダー	
MonoBleedingEdge	2025/03/16 2:19	ファイル フォルダー	
🥥 joken_sample.exe	2025/03/16 2:19	アプリケーション	63
UnityCrashHandler64.exe	2025/03/16 2:19	アプリケーション	1,09
JunityPlayer.dll	2025/03/16 2:19	アプリケーション拡張	28,68

実行したらゲームが開始されます。これにて一応完成です。ですがこれは本当に ギリギリゲームと呼べるレベルの物なので改造して自身の向上に利用していただ けるとありがたいです。