GameProgramming

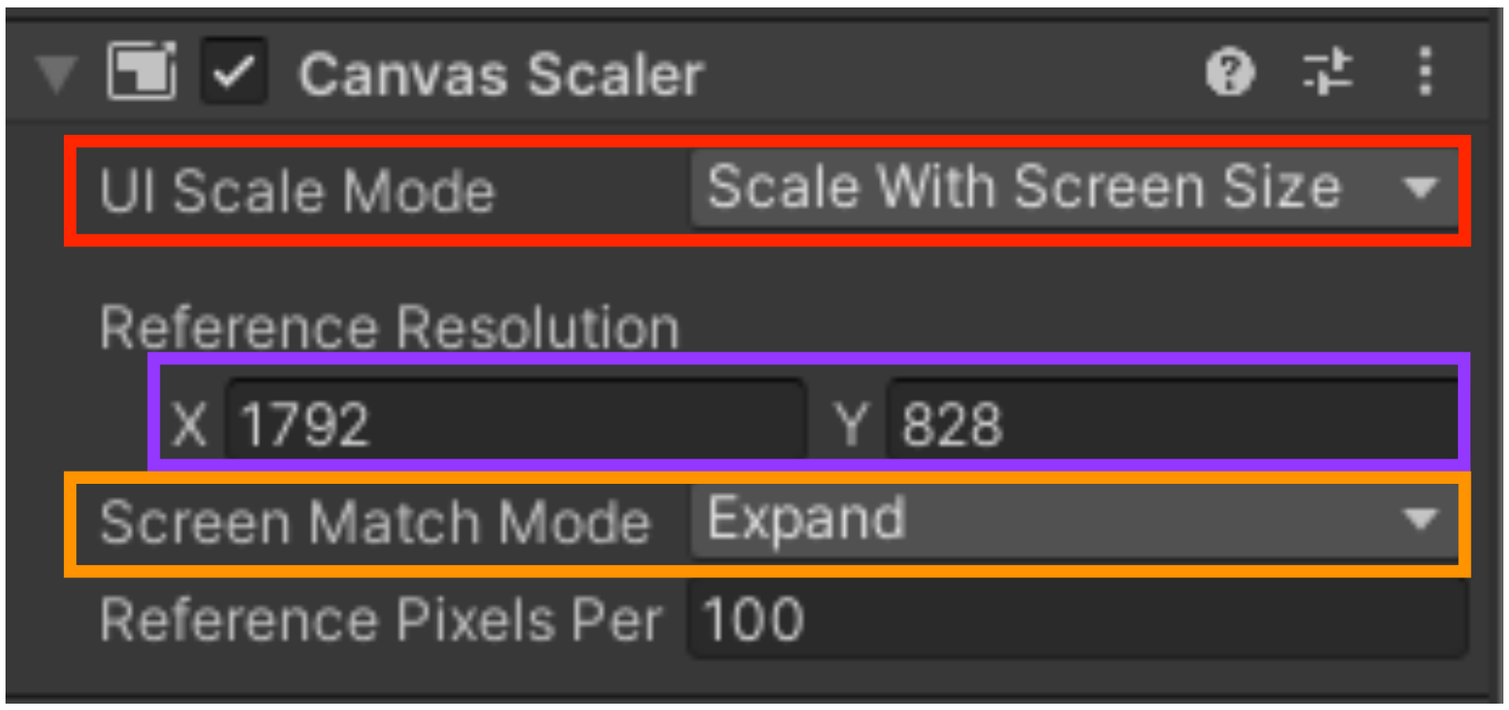
10 UI

配布データ

- 第10回配布データをダウンロード
  「GP10.unitypackage」
- 前回のUnityプロジェクトを開き,
  作ったゲームのシーンを開く
- ダウンロードしたデータをダブルクリックしてimport
- 追加ファイル「Fonts」「Textures」「Modelsにキャラ追加」
- Texturesフォルダ内のすべての画像を選択して,
  TexuteTypeを「Sprite（２D and UI）」に変更
  Applyを選択

CANVAS

- Canvasオブジェクトの作り方は２種（どちらでもOK）
  - 上部メニュー：GameObject → UI →Canvas
  - Hierarchyウィンドウ：Create→UI →Canvas
- Gameウィンドウを16:9に変更
- CanvasのInspector：Canvas Scalerを以下のように設定



画面サイズに合わせて
スケーリング

*iPhone 11*の解像度

上記解像度領域の内容が
全て映るように

背景の作成

- イメージオブジェクトの作成

  上部GameObject＞UI＞Imageで新規作成

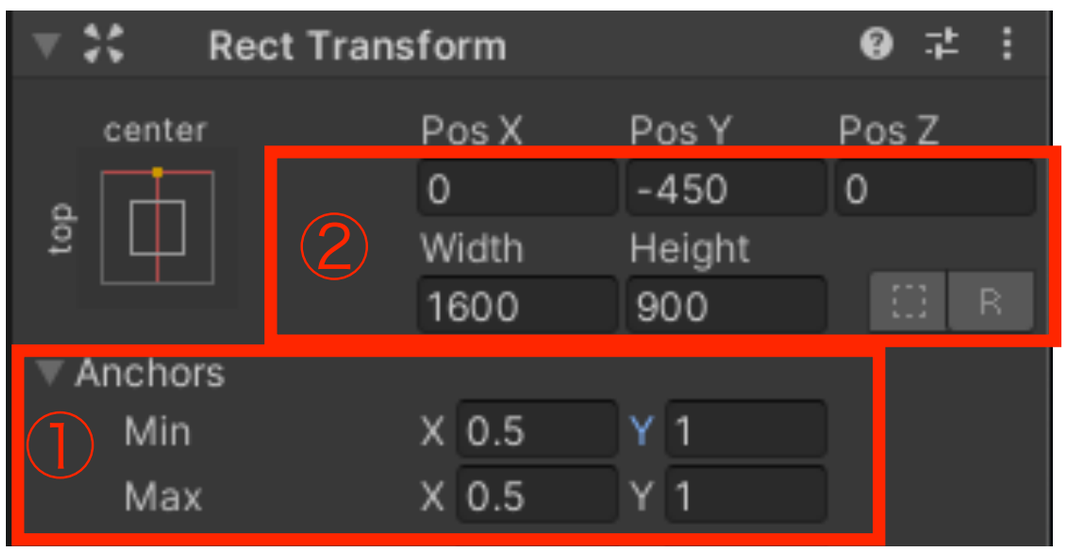
  ImageのInspectorを変更

  名前をBackground Imageに変更

  ①Anchorsを図のように設定（x中央，y上揃え）

  ②PosX: 0, PosY: -450, Width: 1600, Height: 900

  SourceImageにTextures内のbackgroundをD&D



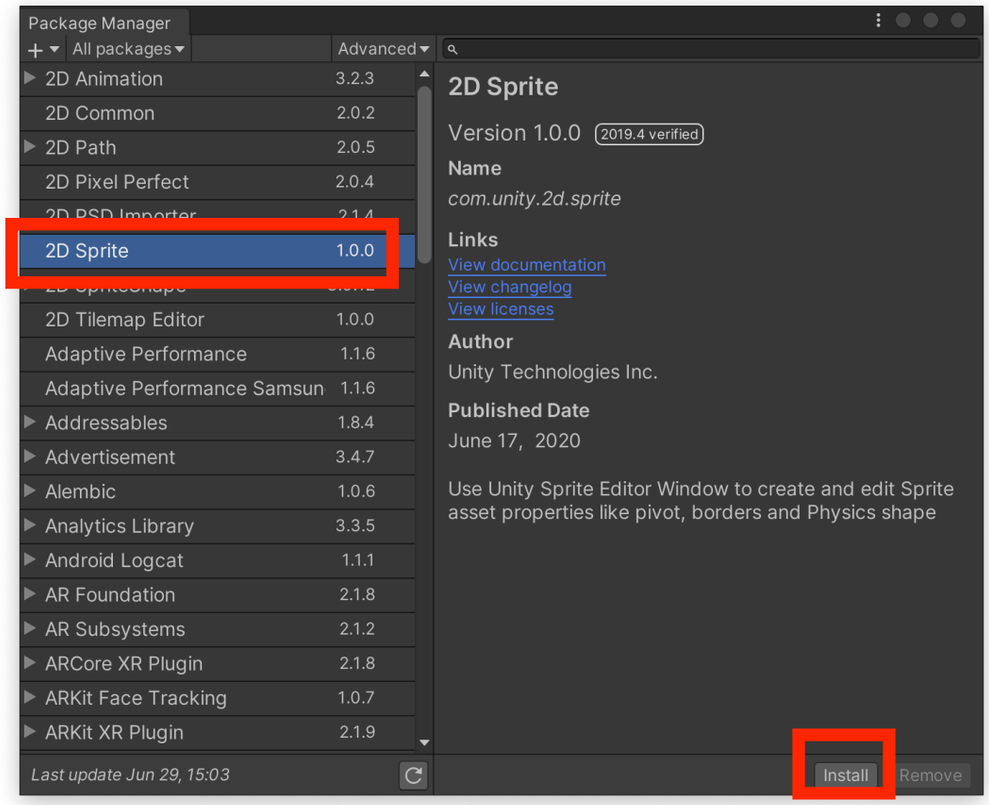
背景の作成

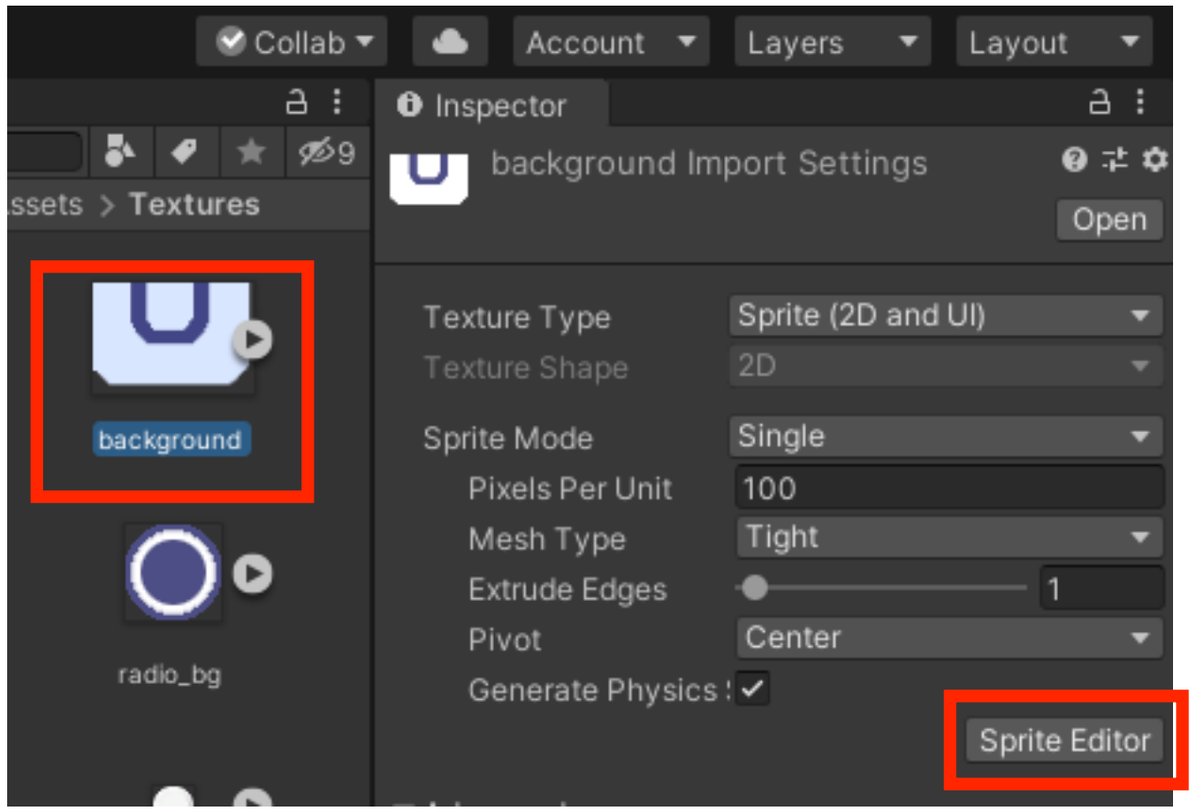
- スライス設定

上部メニュー　Window>Package Managerから

2D Spriteのパッケージをインストール

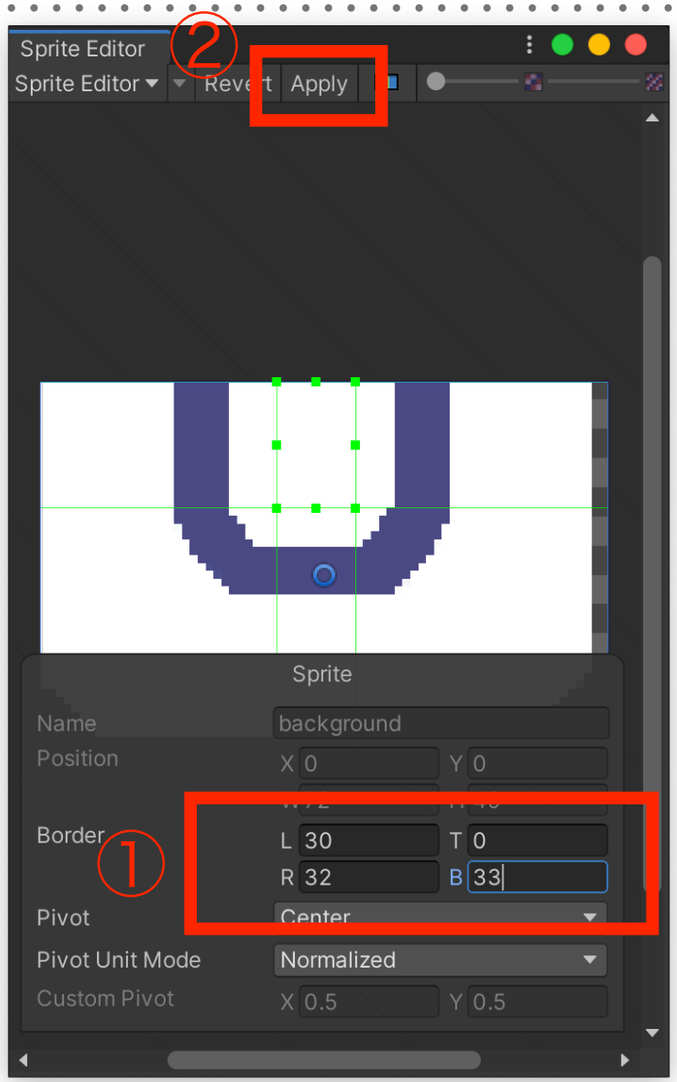
TexturesのbackgroundのInspectorでSpriteEditorを選択



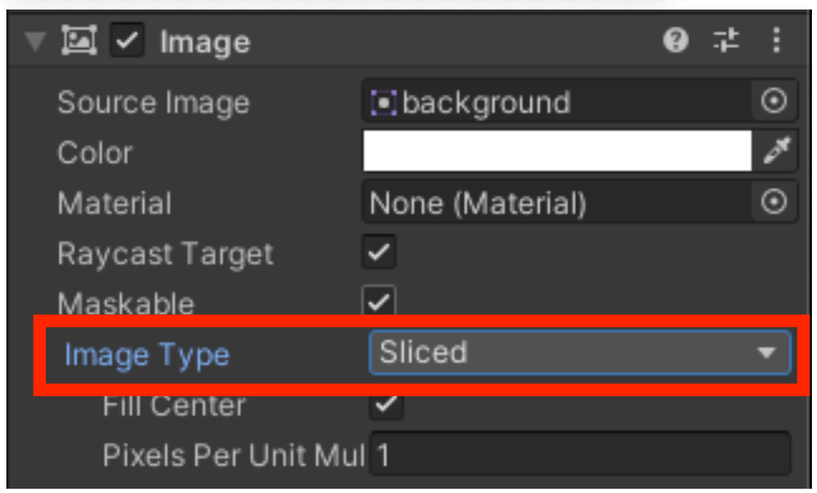


背景の作成

- SpriteEditorで
  Borderの値を
  L: 30, T: 0, R: 32, B:33に変更



- スライス設定の有効化
  Hierarckyウィンドウの
  Canvas内Background Imageを選択
  Inspector＞ImageTypeをSlicedに変更



キャラクター画像の配置

- イメージオブジェクトの作成　上部　GameObject＞UI＞Imageで新規作成

  名前をMain Imageに変更

  SourceImageにTextures内の「大鳥こはく.png」をD&D

  Set Native Sizeをクリック Scaleをx：0.5, y：0.5, z：1に変更

  PosX:0, PosY:60に変更

- マスク設定　上部GameObject＞UI＞Image で新規作成

  名前をMask Imageに変更

  PosX: 0, PosY: 60, Width: 600, Height:400
  ColorをR:128, G:128, B:128, A:128（好きな色でok）に変更

  Add Component＞UI＞Maskを選択

  Main ImageをMask Imageの子にする（Main ImageをMask ImageにD&D）

  Main Imageの位置調整をして顔を見えるようにする。

名前入力用テキストフィールド１

- Textオブジェクトの作成

  上部GameObject＞UI＞Text で新規作成

  名前をPlayer Name Textに変更

  PosX: 0, PosY: 400,
  Width: 300, Height: 40
  TextにPlayer Nameと入力

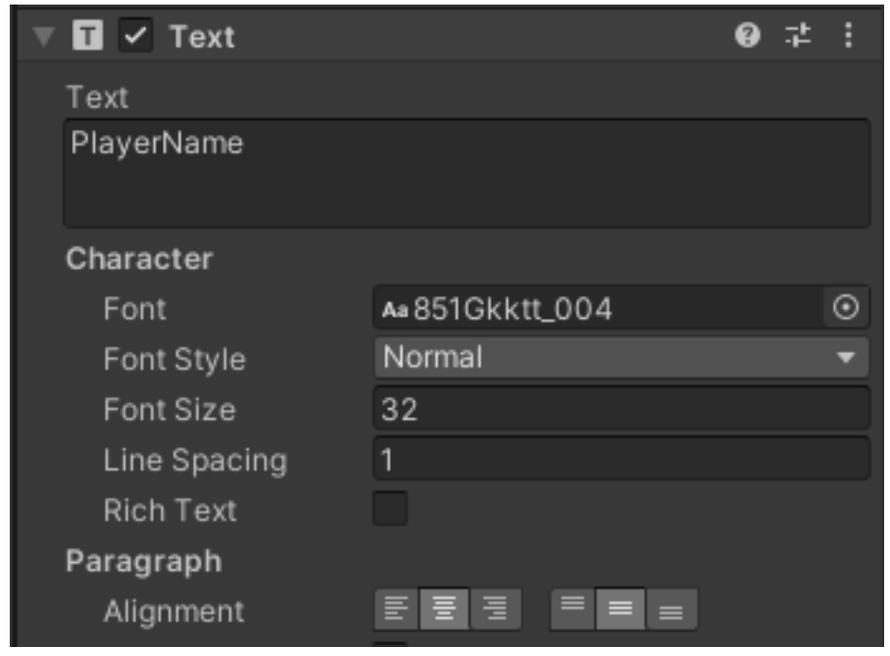
  FontにProjectウィンドウ内

  　Fontsフォルダ内のフォントを設定

  FontsSizeを32に変更

  RichTextのチェックを外す

  Alignmentを中央揃えにする

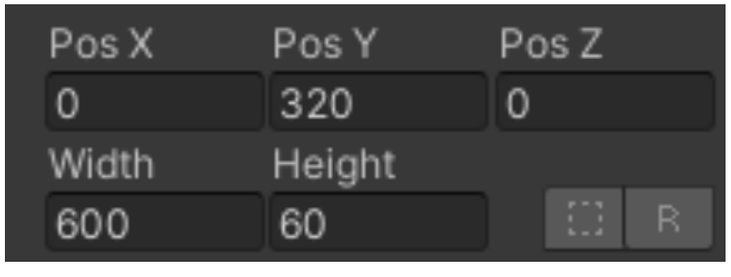


- 入力用テキストフィールドの作成

  Player Name Textの上で右クリック＞Duplicate(複製)

  名前をName Fieldに変更

  PosX: 0, PosY: 320, Width: 600, Height: 60
  FontsSizeを48に変更



名前入力用テキストフィールド２

- 入力用テキストフィールドの作成　続き

  NameFieldのInspector

  Add Component＞UI＞InputFieldを選択

  Interactiveにチェックを入れる

  TextComponentにName FieldをD&D

  Textに「Input Your Name」と入力

- 座布団の設置

  上部GameObject＞UI＞Imageで新規作成

  名前をName Field Area Imageに変更

  Name Field Area ImageをName Fieldの子にする(Name FieldにD&D)

  AnchorsのMin X:0, Y:0, Max X:1, Y:1

  Left: 0, Top 0, Right: 0 Bottom: 0
  ColorをR:200, G:200, B:200 A:64（好きな色でok）に変更

ラジオボタン（TOGGLE）

- Textオブジェクトの作成

  上部GameObject＞UI＞Textで新規作成

  名前をCharacter Textに変更

  PosX: 0, PosY: -200, Width: 300, Height: 40
  TextにCharacterと入力

  FontにProjectウィンドウ内Fontsフォルダ内のフォントを設定

  FontsSizeを32に変更　　RichTextのチェックを外す

  Alignmentを中央揃えにする

- Toggleグループの作成

  上部GameObject＞UI＞Image で新規作成

  名前をToggle Groupに変更

  PosX: 0, PosY: -300, Width: 300, Height: 30
  Image、Canvas Rendererの順に

  それぞれ右上の歯車＞RemoveComponentで削除

  Add Component＞UI＞Toggle Groupを選択

ラジオボタン（TOGGLE）

- ラジオボタンの作成

  上部GameObject＞UI＞Toggleで新規作成

  名前をToggle0に変更

  Toggle0をToggle Groupの子にする

  PosX: -200, PosY: 0, Width: 200, Height: 100
  Toggle0の子（下の階層）のBackground内のSourceImageに
  Texturesフォルダ内のradio_bgをD&D
  Set Native Sizeをクリック

    Backgroundの子（下の階層）のCheckmark内のSourceImageに

    Texturesフォルダ内のradio_checkをD&D

    Set Native Sizeをクリック

  Toggle0の子（下の階層）のLabelのTextに「大鳥こはく」と入力
  Labelの位置を移動ツールで修正

ラジオボタン（TOGGLE）

- ラジオボタンの複製

  Toggle0の上で右クリック＞Duplicateを2回実行

  それぞれ名前をToggle1、Toggle2に変更

  Toggle1の設定を以下にする

  PosX: 10
  Toggle1の子（下の階層）のLabelのTextに「神林ゆうこ」と入力

  Toggle2の設定を以下にする

  PosX: 220
  Toggle2の子（下の階層）のLabelのTextに「藤原みさき」と入力

- グループの初期設定

  Toggle0のIs Onにチェックを入れる　GroupにToggle GroupをD&D

  Toggle1のIs Onのチェックを外す　GroupにToggle GroupをD&D

  Toggle2のIs Onのチェックを外す　GroupにToggle GroupをD&D

プログラム

- ProjectsウィンドウでScriptsフォルダを作る

  Projectsウィンドウ上部Create＞Folder名前をScriptsに変更

- C#ファイルを作る

  Scriptsフォルダを選んだ状態で

  Projectsウィンドウ上部Create＞C#Script

  名前をGameManagerに変更

- ゲームオブジェクトに追加

  HierarchyのCreateでCreate Emptyを選択

  名前をGameManagerに変更

  ScriptsフォルダのGameManager.csをD&D

```
using UnityEngine;
using UnityEngine.UI;

public class GameManager: MonoBehaviour
{
    [SerializeField]
    private InputField nameField; //プレイヤーネーム保存用

    [SerializeField]
    private Image playerImage;  //表示用プレイヤー画像

    [SerializeField]
    private Sprite chara0_Sprite;//プレハブ画像読み込み用

    [SerializeField]
    private Sprite chara1_Sprite;//プレハブ画像読み込み用

    [SerializeField]
    private Sprite chara2_Sprite;//プレハブ画像読み込み用


    public void ChangePlayerType(int index)
    {
        switch (index)
        {
            case 0:
                playerImage.sprite = chara0_Sprite;
                break;
            case 1:
                playerImage.sprite = chara1_Sprite;
                break;
            case 2:
                playerImage.sprite = chara2_Sprite;
                break;
            default:
                break;

        }
    }
}
```

オブジェクトの設定

- HierarchyのGameManagerを選択

  NameField に Canvasの子のNameFieldをD&D

  PlayerImage に MaskImageの子のMainImageをD&D

  Chara0_Sprite に ProjectウィンドウのTexture内の大鳥こはくをD&D

  Chara1_Sprite に ProjectウィンドウのTexture内の神林ゆうこをD&D

  Chara2_Sprite に ProjectウィンドウのTexture内の藤原みさきをD&D

イベントの設定

- Toggle0のInspectorで

  Add Component>Event>EventTrigger
  Add New Event Type > Pointer Click
  List is Emptyと書いてある右下の＋をクリック

  Noneと書いてあるところに

  HierarchyのGameManagerをD&D

  No Functionと書いてあるところを

  GameManager＞ChangePlayerType(int)

  下の値を0に

Toggle1、Toggle2でも同様に行い、

最後の値のみToggle1は１に、Toggle2は２に設定

課題

- 名前を入力できるフィールド,
  UIのトグルボタン（ラジオボタン）を操作して
  キャラクターの画像が切り替わる機能をUIを用いて作成せよ
  (ここまでの授業内容)

- 自身の名前を入力,
  トグルボタン操作で３キャラクターの切り替えが
  確認できる動画を撮影し,
  ファイル名を「学籍番号_GP10」として提出せよ.

- 余裕がある人は次ページ以降の応用課題に挑戦してください
  応用課題に関しては資料以上の説明はしません.

応用課題

➤ 前回授業のプロジェクトと組み合わせて，以下を実装せよ

・授業中に作成したCanvasの中に「決定」ボタンを追加
決定ボタンを押した時の動作
- プレイヤーネームの保存（下記のTextに表示）
- 選んだキャラクターの3Dモデルの表示
- 授業中に作成したCanvasの非表示
- 「menuに戻る」ボタンの表示

・入力したプレイヤーネームを
ゲーム画面のどこかに常時表示するTextの追加

ボタンの作成

- 決定ボタンの作成　Add Component＞UI＞Buttonで新規作成
  名前をSubmit Buttonに変更 Width:160, Height:60に変更
  SourceImageにTextures内のradio_bg.pngをD&D
  radio_bg.pngをsprite editorで編集（LRTB全て17）

  Submit Buttonの子のText内のTextにSubmitと入力
  FontにProjectウィンドウ内Fontsフォルダ内のフォントを設定
  FontsSizeを32に変更
  Alignmentを中央揃えにする　RichTextのチェックを外す
  Submit Buttonを移動ツールで画面下部右側へ移動

- GameManager.csにpublic void Submit(){ボタンを押した際の実行内容}を追加

- SubmitButtonのInspector内のOn Click()の＋を選択
  Noneと書いてあるところにGameManagerをD&D
  No Functionと書いてあるところをGameManager＞Submit()