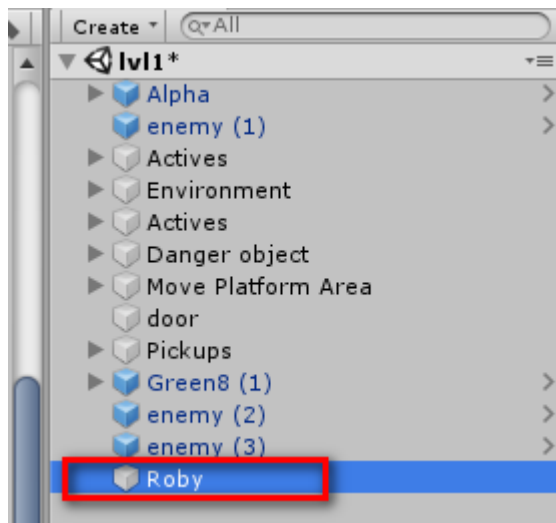


ОСНОВНІ КОМПОНЕНТИ

Перейменуйте спрайт нового персонажа на бажану назву:



Додайте персонажу 2 box collider 2d та 1 circle collider 2d.

Внутрішній box collider визначає точку дотику робота з фізичними об'єктами у грі.

Circle collider відповідає за плавне переміщення по похилим поверхням.

Зовнішній box collider має бути тригерним та відповідає за правильну обробку збору ігрових предметів.

Як приклад, дивіться на налаштування цих колайдерів у Альфи.

Вийти має приблизно такий результат:





Тепер додайте роботу компонент RigidBody2D і заморозьте його обертання по осі Z. Решта налаштувань доведеться зробити пізніше, коли буде написано скрипт для переміщення робота.

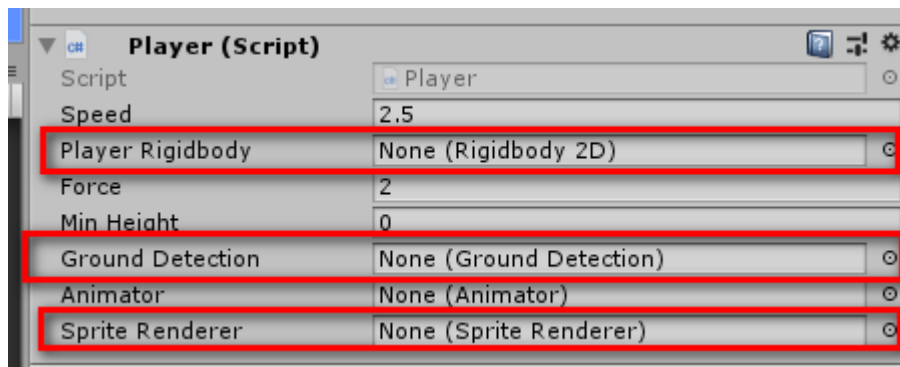
СКРИПТ PLAYER І УПРАВЛІННЯ ПЕРСОНАЖЕМ

Створіть скрипт Player, застосуйте його до роботи та відкрийте. Нам знадобляться такі змінні:

```
public class Player : MonoBehaviour
{
    public float speed = 2.5f;           //Скорость передвижения игрока
    public Rigidbody2D playerRigidbody; //Переменная-ссылка на компонент Rigidbody2D
    public float force = 2.0f;           //Сила прыжка
    public float minHeight;              //Высота, на которой нужно вернуть игрока назад
    public GroundDetection groundDetection; //Для контроля соприкосновения с землей
    Vector3 direction;                   //Направление движения персонажа на каждом кадре
    public Animator animator;            //Для настройки машины состояния анимаций
    public SpriteRenderer spriteRenderer; //Для доступа к изображению объекта
}
```

Додайте на робота раніше написаний скрипт GroundDetection та заповніть наступні поля скрипта Player





Тепер реалізуємо у скрипті рух нового персонажа. Створіть метод `FixedUpdate`. Ви можете видалити стандартні методи!

```
private void FixedUpdate()
{
    direction = Vector3.zero; //Это нужно, т.к. мы не обрабатываем отжатие кнопки
                              //Без этого игрок будет перемещаться на огромные расстояния
                              //(0,0)

    //Если нажата кнопка с кодом A
    Код для проверка нажатия кнопки A
    {
        direction = Vector3.left; //(-1,0) делаем direction равным константе left
    }

    //Если нажата кнопка с кодом D
    Код для проверка нажатия кнопки D
    {
        Сохраняем в direction вектор right //(1,0) делаем direction равным константе right
    }

    direction *= speed; //учитываем скорость перемещения

    //Т.к. перемещение по оси X, то в Y у direction 0
    //тогда как у Player в данный момент в Y значение равно силе притяжения
    //умноженной на массу объекта
    //Чтобы не испортить это значение влиянием на velocity сохраняем его
    direction.y = playerRigidbody.velocity.y;

    //Задаем новое ускорение для объекта
    playerRigidbody.velocity = direction;
}
```

Тепер запрограмуємо стрибок. Стрибок повинен відбуватися, якщо натиснуто кнопку «Пробіл». Так само стрибок має бути лише тоді, коли персонаж стоїть на землі. Для цього вам доведеться звернутися до вашого скрипту `GrounDetection` і перевірити в ній стан змінної `isGrounded`.

Сам стрибок реалізуємо за допомогою імпульсу нагору (`Vector2.up`) з урахуванням сили стрибка.



```
//Прыжок на space
```

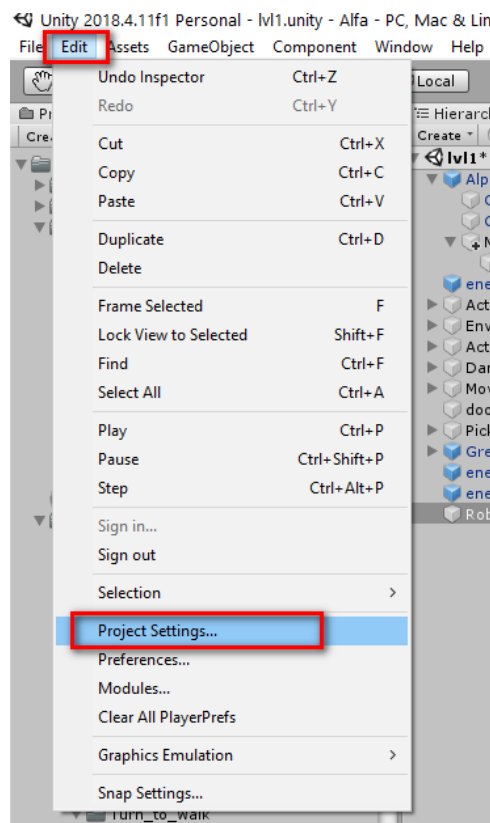
Условие нажатия кнопки "Пробел" и одновременно с этим проверка того, что персонаж стоит на земле

```
{  
    playerRigidbody.AddForce(Vector2.up * force, ForceMode2D.Impulse);  
}
```

Запустіть гру та переконайтеся, що стрибок працює! Зауважте, що Альфа визначав дотик до землі не так, як це запрограмовано у нас. У нашому скрипті проводиться перевірка дотику до об'єкта з тегом «platform». Тому вам необхідно встановити цей тег у всіх префабах, на яких робот має вміти стрибати!

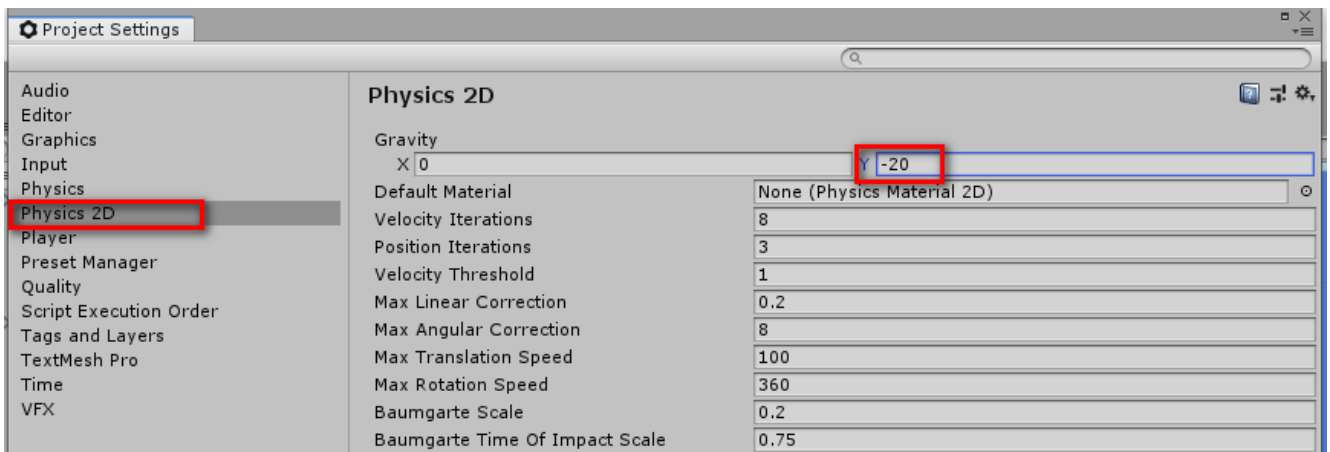
НАЛАШТУВАННЯ ГРАВІТАЦІЇ, СИЛИ СТИБКА І КАМЕРИ

Тепер відкоригуємо гравітацію. Зайдіть в налаштування проекту:



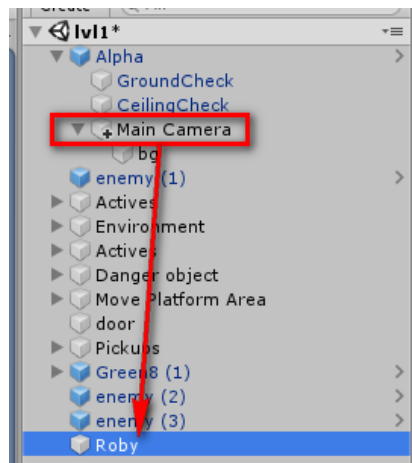
Змінюємо силу гравітації на -20:





Налаштуйте силу стрибка та швидкість руху гравця з урахуванням цих змін!

Тепер перенесіть камеру з Альфи до нового робота:

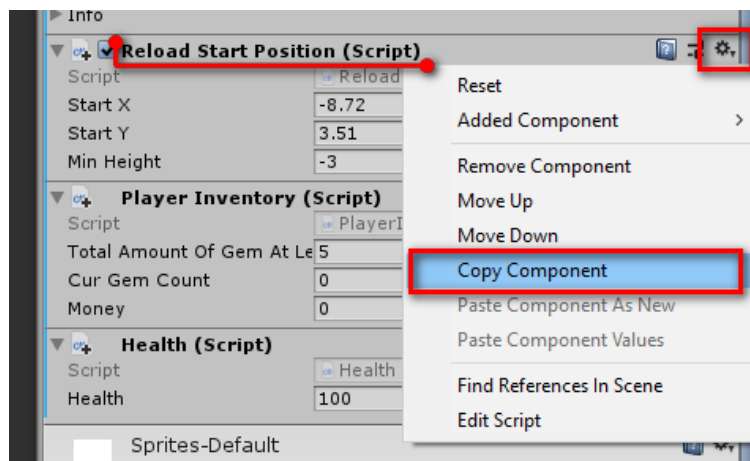


Не забудьте обнулити положення камери щодо нового персонажа по осях X та Y!

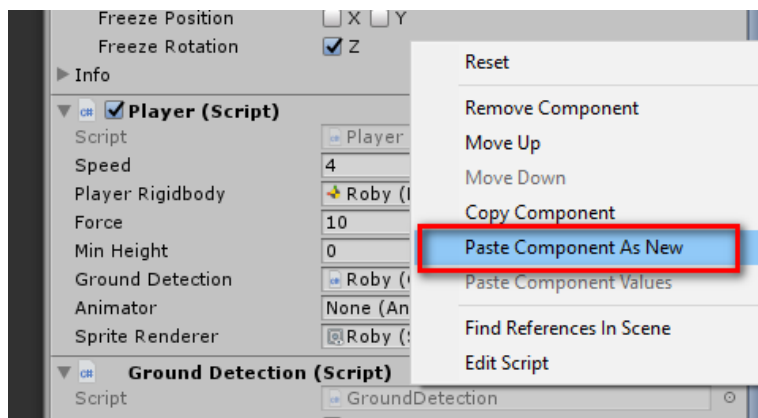
ЗДОРОВ'Я, УРОН І РЕСТАРТ ПОЗИЦІЇ

Тепер скопіюємо створені до цього моменту скрипти зі своїми значеннями. Для цього виділяємо Альфу та копіюємо наступний скрипт:

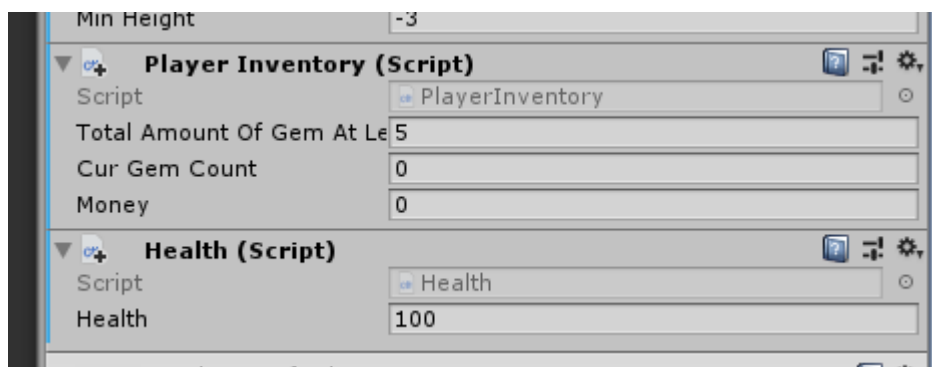




Виділяємо нового робота і натискаємо в інспекторі шестерню навпроти будь-якого компонента. Вибираємо:



Так само копіюємо компоненти:



Якщо запустити гру, то ви побачите, що небезпеки не завдають шкоди новому роботі. Це тому, що у нього не встановлено тег "Player". Виправте цей недолік!



Скрипт `ReloadStartPosition` також потребує невеликої корекції. Нам більше не потрібна перевірка падіння до певної висоти, тому що тепер ми вриваємось у воду. Видаліть у скрипті цей фрагмент:

```
public float startX;    //Переменная для стартовой позиции робота по оси X
public float startY;    //Переменная для стартовой позиции робота по оси Y

public float minHeight; //Положение по оси Y, при котором необходимо вернуть робота
                        //в стартовую позицию
```

```
Unity Message | 0 references
void Update()
{
    if (transform.position.y < minHeight)
    {
        ReloadPosition();
    }

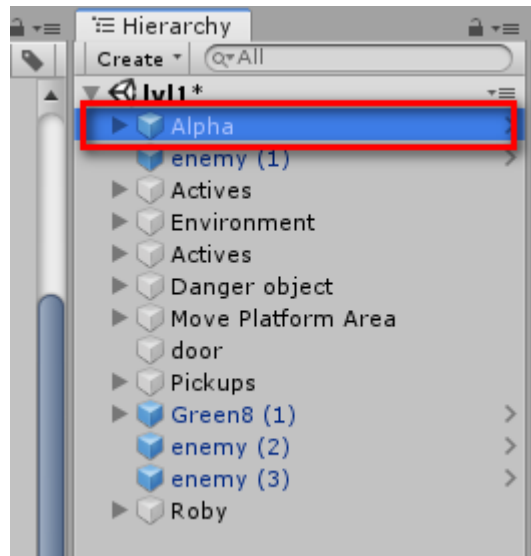
    if (Input.GetKeyDown(KeyCode.F9))
    {
        ReloadPosition();
    }
}
```

ВИДАЛЕННЯ СТАНДАРТНИХ АСЕТІВ

Перед виконанням цієї дії переконайтеся, що новий персонаж здатний виконувати ті самі дії, що виконував Альфа. Можливо, він буде робити це гірше. Трохи коровіше стрибати або по-іншому стикатися з об'єктами. Але технічно він повинен повністю повторювати функціонал Альфи.

Тепер видаліть зі сцени Альфу:





Переконайтеся, що ви випадково не зберегли папку Standard Assets свої файли та видаліть її з проекту:

