

# unity

## Lesson 4



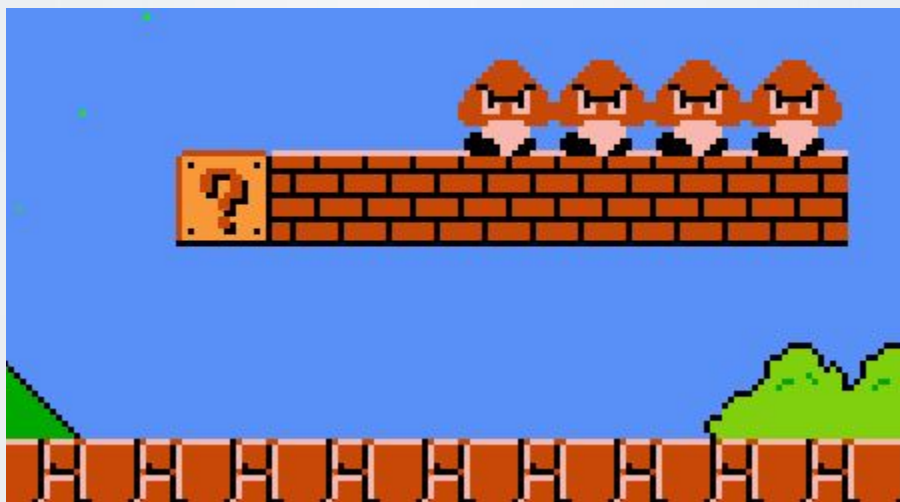
**Robo**  **Code**

# Plan:

- 1 Что такое “Платформер”?
- 2 Движение объекта
- 3 Добавляем прыжки
- 4 Лучший алгоритм для прыжка

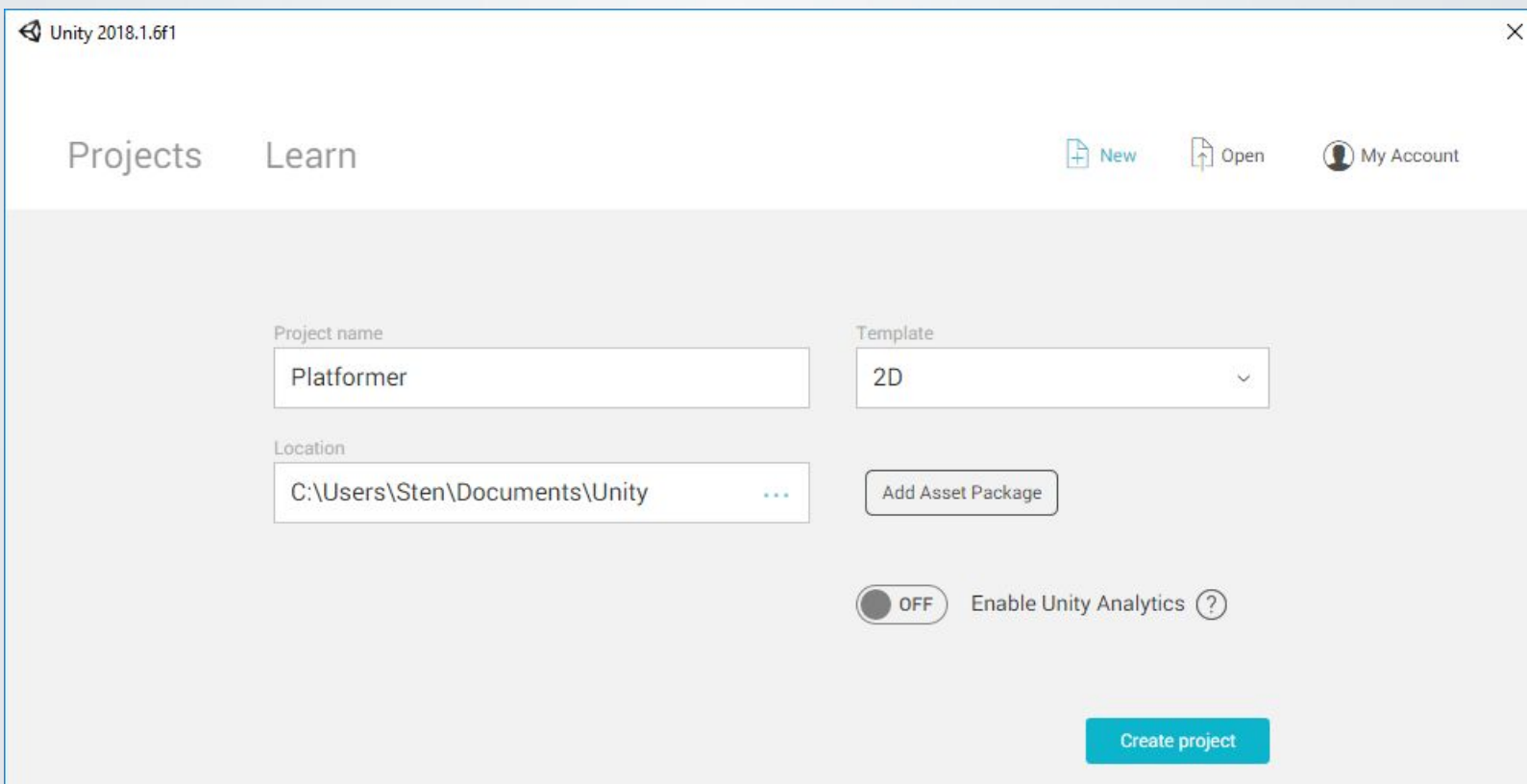
# Что такое “Платформер”?

Платформер - жанр компьютерных игр, в которых основной чертой игрового процесса является прыгание по платформам, лазанье по лестницам, собирание предметов, обычно необходимых для завершения уровня..



# Как создать проект?

Для платформера нам понадобится 2D:



Unity 2018.1.6f1

Projects Learn

New Open My Account

Project name  
Platformer

Template  
2D

Location  
C:\Users\Sten\Documents\Unity

Add Asset Package

OFF Enable Unity Analytics ?

Create project

\*отключите Unity Analytics, мы поговорим об этом позже.

# Движение объекта

Первая и очень важная часть нашего платформера - его перемещение и платформы. Для перемещения в Unity мы используем `transform.position`. Это НЕ физическое перемещение, мы просто меняем координаты объекта. Таким образом, чтобы изменить позицию, мы должны передать параметры типа `Vector` этому методу.

Например:

```
transform.position = new Vector2 (transform.position.x + s  
pd, transform.position.y);
```

Но, лучше использовать физику. Поэтому, дальше поговорим о `Velocity`.



# Velocity

Скорость объекта - это скорость изменения его положения относительно системы отсчета. Так что это напрямую связано со скоростью и ускорением нашего игрока.

Как ее использовать?

1. Для начала необходимо добавить Rigidbody2D игроку.
2. Далее написать следующий код для движения вправо:

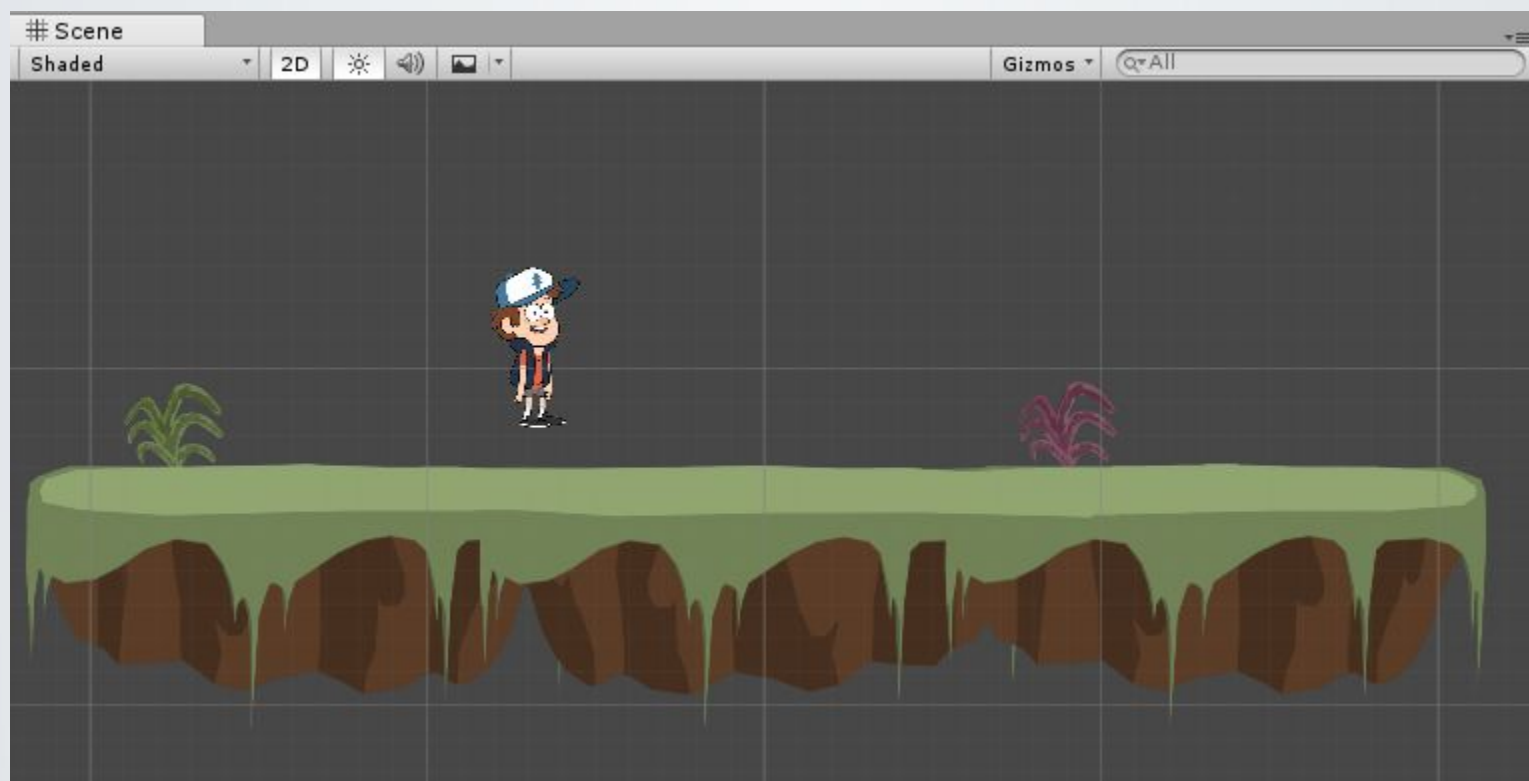
```
if (Input.GetKey(KeyCode.DownArrow))  
{  
    rb.velocity = new Vector2(0.0f, 1.0f);  
}
```



```
Rigidbody2D rb = gameObject.GetComponent<RigidBody2D>();
```

# Платформа

Теперь давайте создадим платформу:



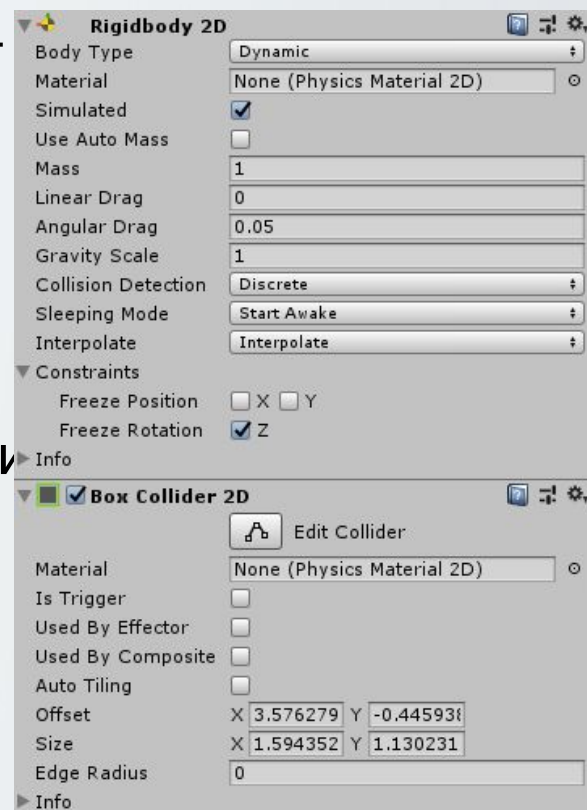
# Свойства объектов

Нам необходимы два свойства для игрока: Rigidbody2D и Collider 2D. Нам нужно это для того, чтобы соединить 2 объекта, чтобы сделать их “физическими”.

Вот как это выглядит. Rigidbody2D – дает нам гравитацию, тип объекта и тд.

Collider позволяет менять область соприкосновения с другими объектами.

Платформе выставляем только collider, так как мы не нуждаемся в ее физических свойствах.





# Добавляем прыжок

Для прыжка мы тоже можем использовать `transform.position` но в таком случае у нас не будет физики. Так, для решения этой задачи мы будем использовать `AddForce`

```
player.AddForce (new Vector2(0f, jumpForce));
```

Как это работает? Метод `AddForce` принимает `Vector2` или `Vector 3`, которые указывают насколько сильна должна быть сила и в каком направлении она будет применяться. Это может быть `(x,y)` или `(x,y,z)`.

Это будет работать только с объектами `Rigidbody` (в нашем случае `Rigidbody2D`, потому что у нас 2D проект).

Чтобы легче было подбирать силу прыжка, можно создать переменную `jumpForce` – это переменная типа `float` которая указывает силу прыжка.

# Кодим

Сейчас нам надо написать скрипт для игрока, который заставит его двигаться, используя свойство *Velocity*, а также прыгать, используя метод `AddForce()`;  
Для платформі скрипт не требуется (ура!).

# Практика

**Now, lets code!**

- Создать проект - “Платформер”
- Добавить движения, используя velocity
- Добавить возможность прыгать



\*Если все сделано => +5 exp



Bonus exp:

- Запретить прыжки в воздухе (+3 exp)
- Первый, кто выполнит основное задание (+1 exp)
- Создать маленький мир из платформ (+2 exp)



После практики у тебя будет 30-ти минутный тест с 10-ю вопросами.

Каждый тест имеет только один правильный ответ.