



2.1 ПОЗИЦІОНУВАННЯ ГРАВЦЯ

План

- Крок 1: Створіть новий проект для Prototype 2
- Крок 2: Додайте гравця, тварин та їжу
- Крок 3: Отримайте дані переміщення
- Крок 4: Перемістіть героя у відповідному напрямку
- Крок 5: Створіть границі для гравця
- Крок 6: Оптимізуйте свій код

Фінальний результат



Час: 60 хвилин

Короткий опис: Ви почнете цей блок із створення нового проекту для вашого другого Prototype. Спочатку вам потрібно обрати головного вигляд персонажа, тварин яких би ви хотіли нагодувати та якою їжею. Ви запрограмуєте гравця на рух з боку в бік, як у Prototype 1, тільки тепер вам знадобляться оператор if, щоб утримати гравця в обмеженнях.

Результат: Гравець зможе рухатися вліво і вправо по екрану на основі натискання лівою та правою клавішею, але він не зможе залишити ігрову зону по обидві сторони.

Навчальні цілі: Наприкінці цього уроку ви зможете:

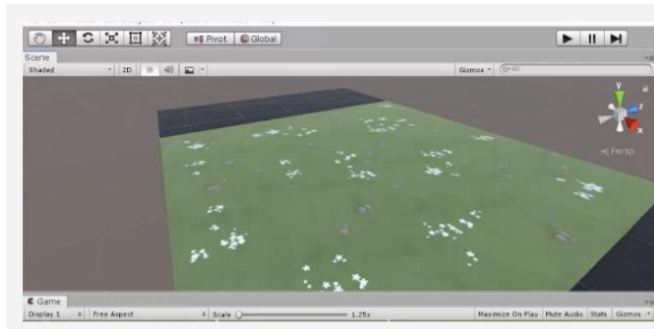
- Налаштовувати масштаб об'єкта
- Більш зручно використовувати функцію `GetInput()`
- Працювати з оператором `if`, щоб реалізувати базову логіку у вашому проекті, включаючи використання операторів порівняння більше (`>`) і менше (`<`)
- Використовувати коментарі та автоматичне форматування, щоб зробити наш код більш чистим і читабельним для інших програмістів

КРОК 1: СТВОРІТЬ НОВИЙ ПРОЕКТ ДЛЯ PROTOTYPE 2

Перше, що нам потрібно зробити, це створити новий проект та імпортувати пограткові файли Prototype 2.

1. Відкрийте **Unity Hub** і створіть **новий** проект з назвою “**Prototype 2**” у своїй папці
2. Перейдіть за **посиланням**, завантажте та **імпортуйте** потрібні нам файли до Unity
3. Відкрийте **сцену Prototype 2** і **видаліть** SampleScene без збереження
4. У верхньому правому куті редактора Unity змініть макет розташування вікон від **ТИПОВОГО** на власний

- **Не хвилюйтеся:** unitypackage 2 має набагато більше ассетів, ніж unitypackage1, тому розпаковка може зайняти деякий час.



КРОК 2: ДОДАЙТЕ ГРАВЦЯ, ТВАРИН ТА ЇЖУ

Давайте розмістимо всі наші об'єкти на сцені, включаючи головного героя, тварин та їжу.

1. Якщо ви хочете, перетягніть інші **матеріали** із *Course Library > Materials* до об'єкта Ground
2. Перетягніть 1 **людину**, 3 **тварин** та 1 **шматочок** їжі до ієрархії
3. Перейменуйте людину «**Player**», а потім змініть **позицію** інших доданих об'єктів, щоб ви їх роздивитись
4. Відрегулюйте шкалу XYZ у **scale**, щоб ви могли легко побачити всі об'єкти згори.

- **Нова техніка:** регулювання масштабу

- **Попередження:** людей використовуйте тільки в ролі гравця, оскільки у них немає анімація для ходьби

- **Порада.** Пам'ятайте, перетягування об'єктів у ієрархія ставить їх на початок



КРОК 3: ОТРИМАЙТЕ ДАНІ ПЕРЕМІЩЕННЯ

Якщо ми хочемо перемістити програвач зліва направо, нам потрібна змінна, що відстежує дані з клавіатури

1. У папці **Assets** створіть папку “**Script**”, а в ній файл “**PlayerController.cs**”
2. **Прикріпіть** скрипт до Player та відкрийте його
3. Угорі PlayerController.cs оголошіть нову змінну **public float horizontalInput**
4. В **Update ()** встановіть **horizontalInput = Input.GetAxis (“Horizontal”)**, а потім протестуйте

- **Попередження:** Обов’язково створіть свою папку сценаріїв
- **Не хвилюйтеся:** ми збираємось детально знайомитись з цим процесом
- **Попередження:** якщо ви неправильно написали ім’я скрипту, просто видаліть його та створіть заново.

```
public float horizontalInput;

void Update()
{
    horizontalInput = Input.GetAxis("Horizontal");
}
```

КРОК 4: ПЕРЕМІСТІТЬ ГЕРОЯ У ВІДПОВІДНОМУ НАПРЯМКУ

Нам фактично потрібно використовувати вхідні дані для перекладу програвача вліво та вправо

1. Об’явіть нову змінну **public float speed= 10,0f;**
2. В **Update()** напишіть строку яка буде відповідати за рух по горизонталі нашого персонажа враховуючі **швидкість**

- **Порада.** Ви можете переглянути свої старі сценарії

```
public float horizontalInput;
public float speed = 10.0f;

void Update()
{
    horizontalInput = Input.GetAxis("Horizontal");
    transform.Translate(Vector3.right * horizontalInput * Time.deltaTime * speed);
}
```

КРОК 5: СТВОРІТЬ ГРАНИЦІ ДЛЯ ГРАВЦЯ

Ми повинні запобігти тому, щоб програвач не виходив за межі екрана за допомогою оператора **if**.

1. В **Update()** напишіть оператор **if**, для перевірки, чи є ліва позиція гравця **X менше** певного значення
2. В тілі оператора **if** задайте поточну позицію гравця, але з фіксованим розташуванням по **X**

- **Порада.** Перемістіть героя у режимі перегляду для знаходження крайньої точки
- **Нова концепція:** твердження «якщо-то»
- **Нова концепція:** оператори порівняння «<>» та «<>>»

```
void Update() {
    if (transform.position.x < -10) {
        transform.position = new Vector3(-10, transform.position.y, transform.position.z);
    }
}
```

КРОК 6: ОПТИМІЗУЙТЕ СВІЙ КОД

Нам також потрібно зробити це з правого боку, а потім очистити наш код

1. Повторіть цей процес для **правої сторони** екрану
2. Оголосіть нову змінну **xRange**, а потім замініть нею відповідні значення
3. Додайте **коментарі** до свого коду

- **Попередження**: кожного разу, коли ви бачите значення яке повторюється, спробуйте замінити його змінною

- **Попередження**: стежте за вашим значенням більше / менше!

```
public float xRange = 10;

void Update()
{
    // Keep the player in bounds
    if (transform.position.x < -10 -xRange)
    {
        transform.position = new Vector3(-10 -xRange, transform.position.y, transform.position.z);
    }
    if (transform.position.x > xRange)
    {
        transform.position = new Vector3(xRange, transform.position.y, transform.position.z);
    }
}
```

ПІДСУМОК УРОКУ

Досягнуті цілі

- Гравець може переміщатися вліво та вправо на основі керування лівої та правої клавіші клавіатури
- Гравець не зможе залишити ігрову зону з будь-якої сторони

Нові поняття

- Налаштування масштабу об'єкта

та навички

- оператор if
- оператори порівняння «>», «<»

Наступний урок

- Ми навчимося створювати і кидати нескінченну кількість їжі, щоб годувати своїх тварин!