

Веселе програмування для початківця

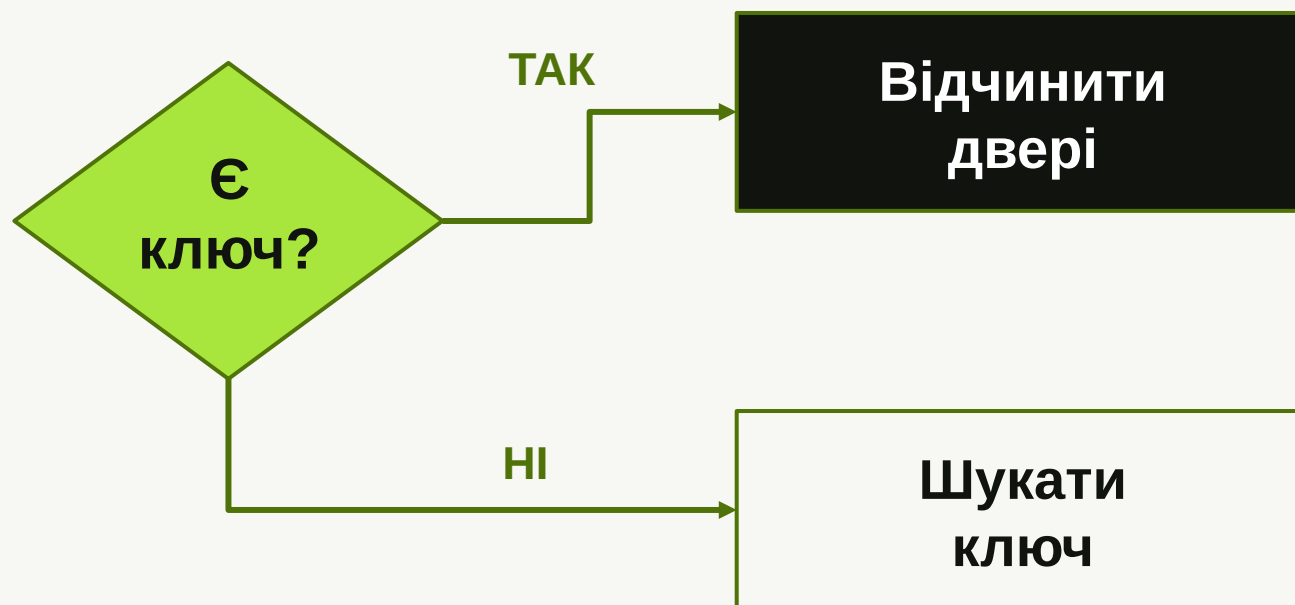
Олександр Коваль

Ігрові світи, власні правила
та перші програми на Python



Що відчинить двері?

У грі ми задаємо умови.
Програма перевіряє їх
і обирає наступну дію.



А якщо двері потребують двох ключів?

Ігрові світи та власний код

01

Будуємо світ

Персонажі, простір
і мета гри

02

Задаємо правила

Події, умови
та повторення

03

Пишемо Python

Графіка, дані
та прості програми

Kodu Game Lab Construct Minecraft Education Roblox Studio

Глибину Python добираємо з урахуванням класу.

Світ, у якому діють твої правила



Світ

Де відбувається пригода?

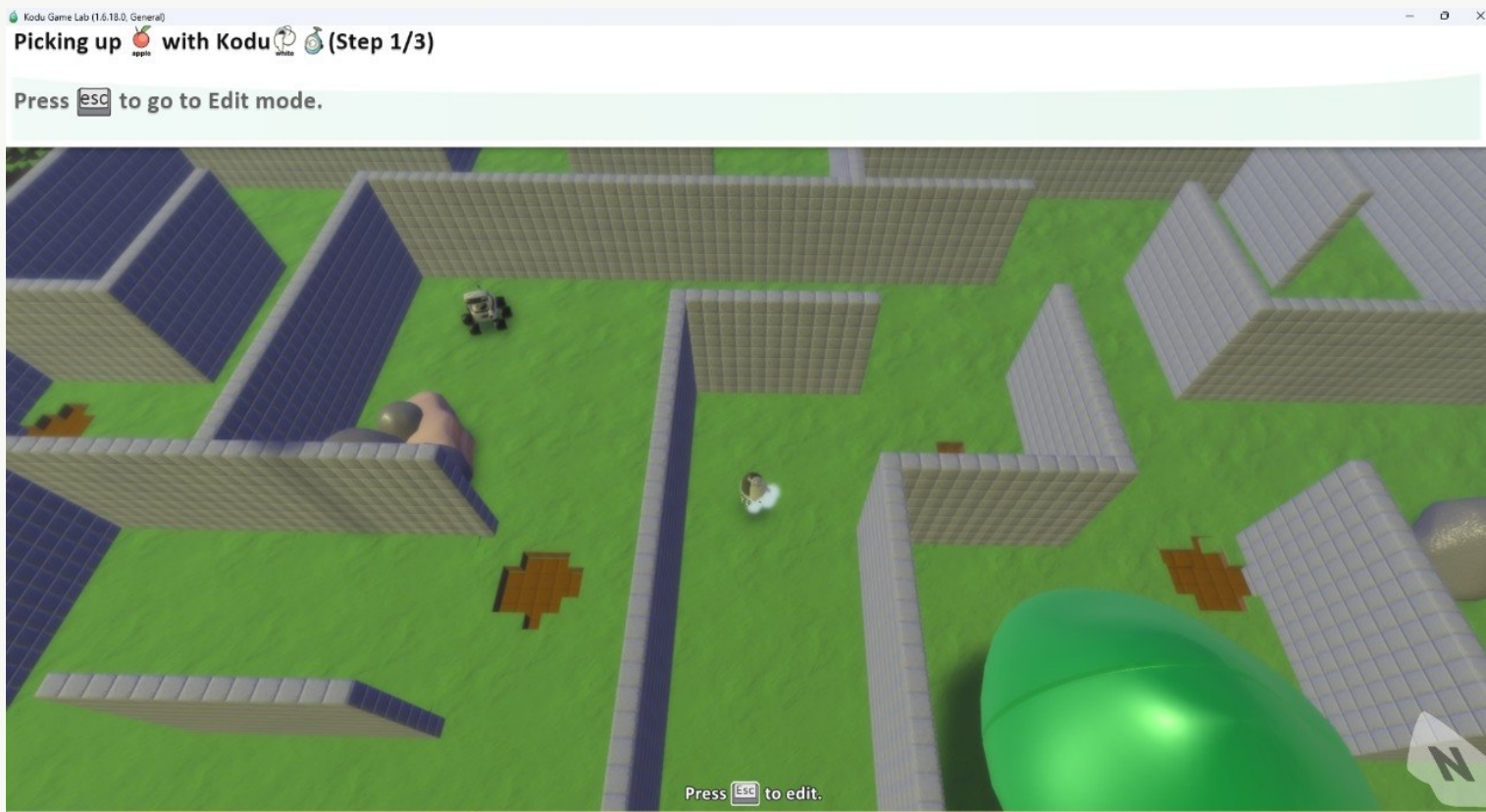
Герой

Що він має зробити?

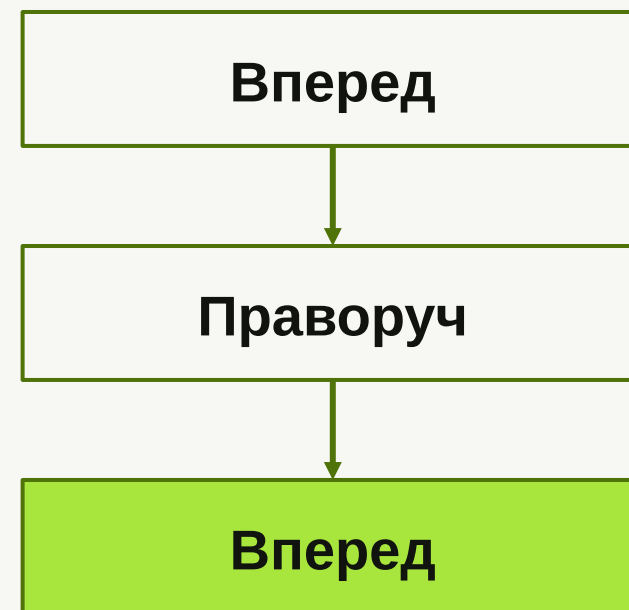
Правило

Коли герой рухається або зупиняється?

Лабіринт починається з плану

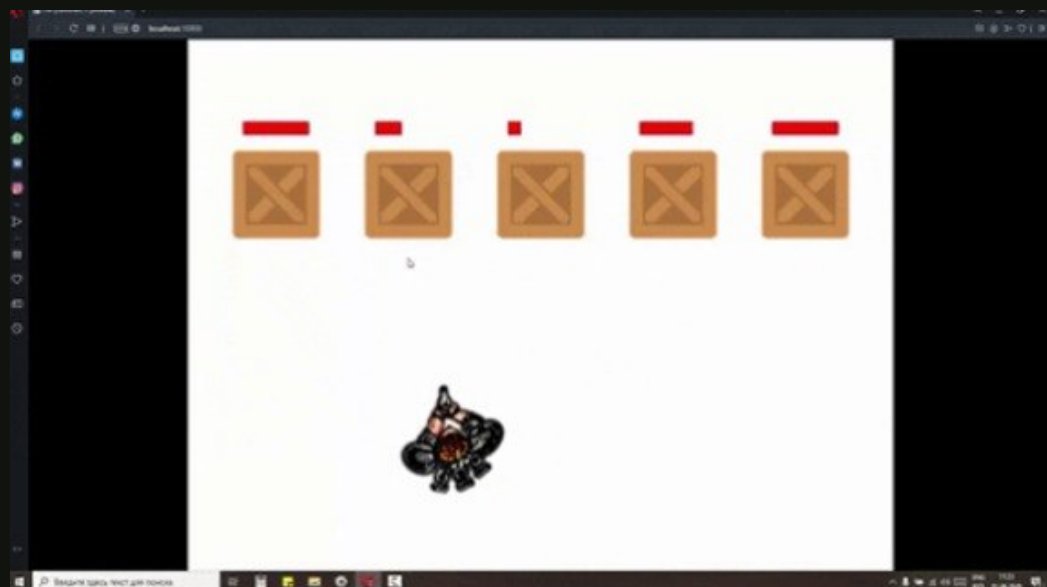


Приклад команд



Яка команда буде наступною?

Подія запускає дію



Подія

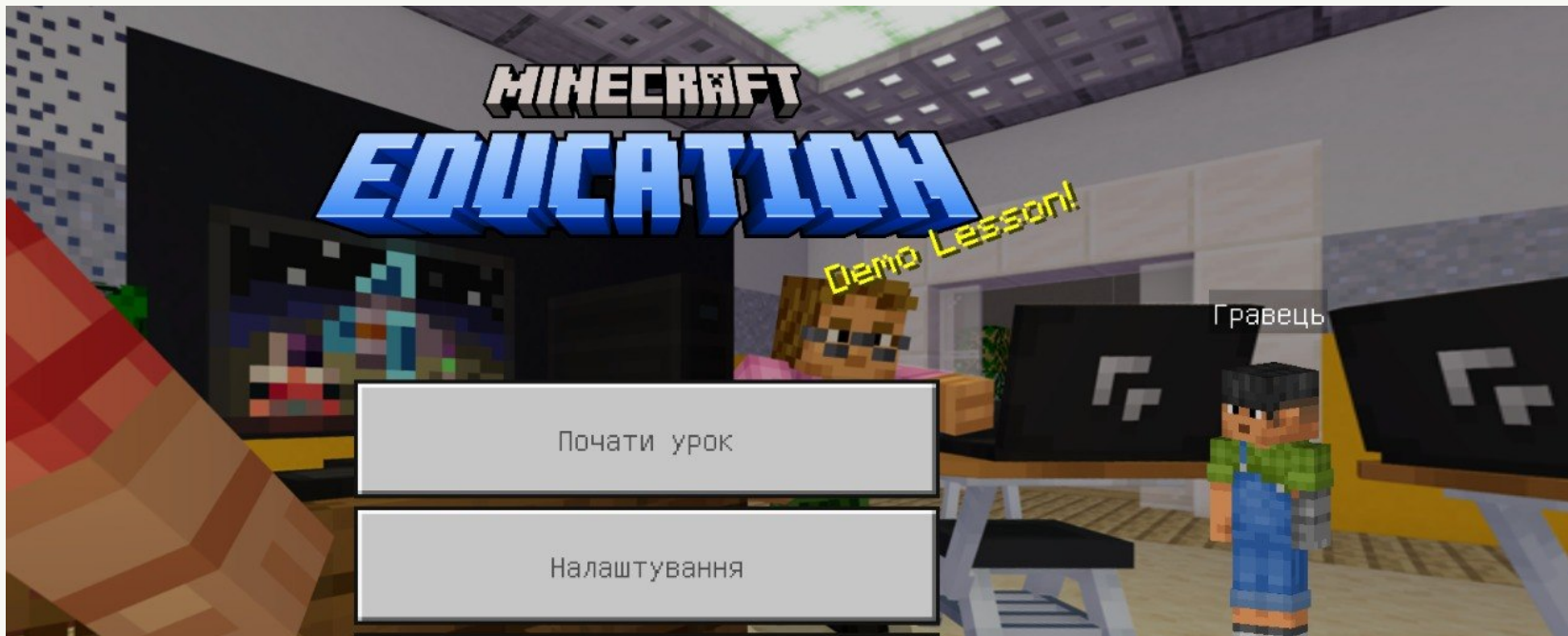
Натиснули клавішу

Дія

Герой рухається

Яку дію можна призначити іншій клавіші?

Знайомий світ, нова задача



Наше питання

Як пояснити
іншому гравцеві
свій план?

Велика задача стає зрозумілішою, коли поділити її на кроки.

Кожен світ ставить свою задачу

Вітаємо вас на відкритих уроках Minecraft Education! Запрошуємо вас переглянути ці уроки та дізнатися, як Minecraft розвиває навички точних наук, розкриває творчі здібності та залучає учнів до співпраці та вирішення проблем.



Bad Connection?

Автор: Minecraft Education



Година штучного інтелекту: Перша ніч

Автор: Minecraft Education



Фантастичні ярмарки

Автор: Minecraft Education



Година програмування: Покоління ШІ

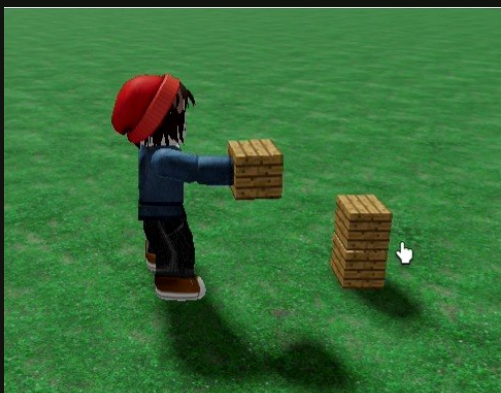
Автор: Minecraft Education

Що тут потрібно з'ясувати?

Як перевірити своє рішення?

Код змінює ігровий світ

```
script.Parent.Activated:Connect(function()  
    local handle = script.Parent.Handle --модель блока  
    local root = script.Parent.Parent.HumanoidRootPart --тіло гравця  
    local position = root.CFrame --позиція гравця  
  
    local block = handle:Clone() --клон блока  
    block.CFrame = position * CFrame.new(0,0,-5) --позиція блока перед гравцем  
    block.Parent = workspace --створюємо блок  
    block.CanCollide = true --можливість доторкатися блоку  
  
    wait(0.5)  
    block.Anchored = true --закріплюємо блок  
end)
```



Читаємо код через те,
що відбувається
на екрані.

01 Активуємо
інструмент

02 Створюємо
копію блока

03 Закріплюємо
блок

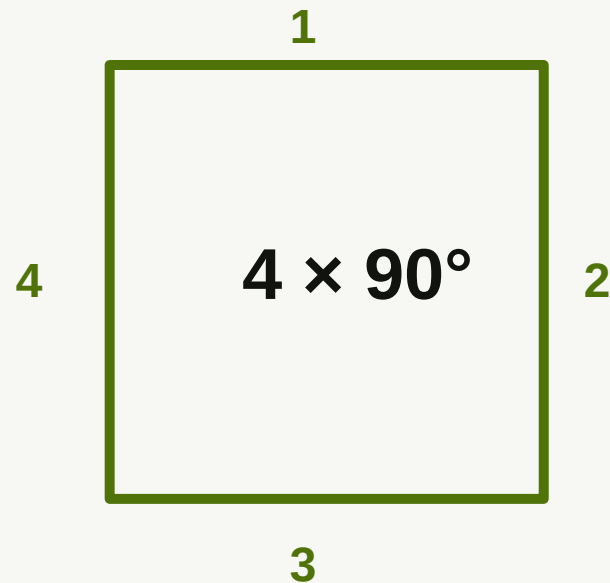
Повторення малює квадрат

Одна пара команд. Чотири повторення.

```
import turtle

for _ in range(4):
    turtle.forward(100)
    turtle.right(90)

turtle.done()
```



Що зміниться, якщо замість 100 написати 150?

Що ще можна створювати

Список предметів

`["ключ", "карта", "ліхтар"]`

Список зберігає кілька значень.

Власна команда

Функція `jump()`

Функція об'єднує потрібні дії.

Вікно з кнопкою

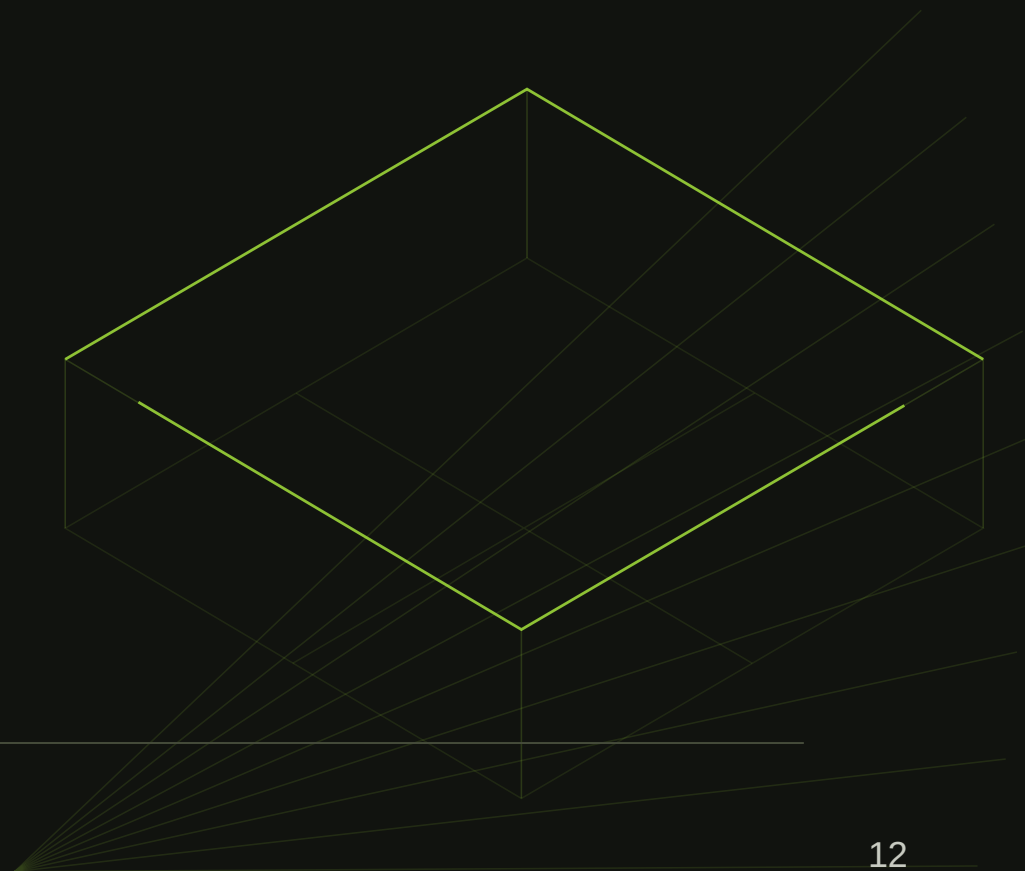
Натискаємо «Почати»

Програма реагує на команду.

Перша версія може бути простою

- 01** Обираємо посильну задачу
- 02** Створюємо першу версію
- 03** Перевіряємо й виправляємо
- 04** Показуємо та пояснюємо рішення

Помилка підказує, яке правило потрібно перевірити.



Якою буде твоя гра?

Мій герой...

Його мета...

Головне правило...

Наприклад: робот шукає вихід, а двері відчиняються лише з ключем.

Попередній формат навчання

6 місяців

Одне заняття на тиждень

Тривалість заняття: 1,5 години

1 300 грн

на місяць

Четвер: попередньо

Старт, час, місце та повні умови
уточнюємо перед записом.

ONOVY

Яку гру створиш ти?

Зв'язок із Олександром Ковалем

093 050 12 44

066 793 92 81

Сторінка курсу на onovu.com.ua

Програму можна переглянути разом із батьками.

Звернення подають батьки або законний представник.

Техніку й доступи погоджуємо до навчання. Звернення не бронює місце.

