

TÁC GIẢ: DEV ALEX

MINECRAFT PYTHON VERSION 1

GIỚI THIỆU

Tài liệu này sẽ phân tách đoạn code demo Python sử dụng thư viện `ursina` ra thành các file riêng biệt, giúp tổ chức và quản lý code tốt hơn. Mỗi phần của ứng dụng sẽ được đặt trong các file tương ứng và được import vào file chính để chạy ứng dụng.

CẤU TRÚC THƯ MỤC

Cấu trúc thư mục của dự án sẽ như sau:

```
project/
├─ main.py
├─ src/
│   ├─ resources/
│   │   └─ textures.py
│   ├─ events/
│   │   └─ update.py
│   └─ scene/
│       ├─ voxel.py
│       ├─ sky.py
│       └─ hand.py
└─ settings/
```

< Dev Alex />

NỘI DUNG CÁC FILE

FILE SRC/RESOURCES/TEXTURES.PY

File này sẽ chứa các lệnh tải các tài nguyên:



< Dev Alex />

```
from ursina import load_texture, Audio
from src.settings import TEXTURE_PATH, SFX_PATH

# Tải các texture
grass_texture = load_texture(str(TEXTURE_PATH /
    "Grass_Block.png"))
stone_texture = load_texture(str(TEXTURE_PATH /
    "Stone_Block.png"))
brick_texture = load_texture(str(TEXTURE_PATH /
    "Brick_Block.png"))
dirt_texture = load_texture(str(TEXTURE_PATH /
    "Dirt_Block.png"))
wood_texture = load_texture(str(TEXTURE_PATH /
    "Wood_Block.png"))
sky_texture = load_texture(str(TEXTURE_PATH / "Skybox.png"))
arm_texture = load_texture(str(TEXTURE_PATH /
    "Arm_Texture.png"))

# Tải âm thanh
punch_sound = Audio(str(SFX_PATH / "Punch_Sound.wav"),
    loop=False, autoplay=False)
```

FILE SRC/EVENTS/UPDATE.PY

File này sẽ chứa hàm `update`:

< Dev Alex />

```
from ursina import held_keys
from src.scene.hand import hand

block_pick = 1

def update():
    global block_pick

    if held_keys["left mouse"] or held_keys["right mouse"]:
        hand.active()
    else:
        hand.passive()

    if held_keys["1"]: block_pick = 1
    if held_keys["2"]: block_pick = 2
    if held_keys["3"]: block_pick = 3
    if held_keys["4"]: block_pick = 4
    if held_keys["5"]: block_pick = 5
```

FILE SRC/SCENE/VOXEL.PY

File này sẽ chứa định nghĩa lớp **Voxel**:

< Dev Alex />

```
from ursina import Button, mouse, destroy, scene, color
from src.resources.textures import grass_texture, stone_texture,
brick_texture, dirt_texture, wood_texture, punch_sound
import random
from src.events.update import block_pick
from src.settings import MODEL_PATH
```

```
class Voxel(Button):
    def __init__(self, position=(0, 0, 0),
texture=grass_texture):
        super().__init__(
            parent=scene,
            position=position,
            model=str(MODEL_PATH / "Block"),
            origin_y=0.5,
            texture=texture,
            color=color.color(0, 0, random.uniform(0.9, 1)),
            highlight_color=color.light_gray,
            scale=0.5
        )

    def input(self, key):
        if self.hovered:
            if key == "left mouse down":
                punch_sound.play()
                if block_pick == 1: voxel =
Voxel(position=self.position + mouse.normal,
texture=grass_texture)
                if block_pick == 2: voxel =
Voxel(position=self.position + mouse.normal,
texture=stone_texture)
                if block_pick == 3: voxel =
Voxel(position=self.position + mouse.normal,
texture=brick_texture)
                if block_pick == 4: voxel =
Voxel(position=self.position + mouse.normal,
texture=dirt_texture)
                if block_pick == 5: voxel =
Voxel(position=self.position + mouse.normal,
texture=wood_texture)

            if key == "right mouse down":
                punch_sound.play()
                destroy(self)
```


FILE SRC/SCENE/SKY.PY

File này sẽ chứa định nghĩa lớp **Sky**:



< Dev Alex />

```
from ursina import Entity, scene
from src.resources.textures import sky_texture

class Sky(Entity):
    def __init__(self):
        super().__init__(
            parent=scene,
            model="Sphere",
            texture=sky_texture,
            scale=150,
            double_sided=True
        )
```

FILE SRC/SCENE/HAND.PY

File này sẽ chứa định nghĩa lớp **Hand**:



< Dev Alex />

```
from ursina import Entity, camera, Vec3, Vec2
from src.resources.textures import arm_texture
from src.settings import MODEL_PATH

class Hand(Entity):
    def __init__(self):
        super().__init__(
            parent=camera.ui,
            model=str(MODEL_PATH / "Arm"),
```

```
        texture=arm_texture,  
        scale=0.2,  
        rotation=Vec3(150, -10, 0),  
        position=Vec2(0.4, -0.6)  
    )
```

```
def active(self):  
    self.position = Vec2(0.3, -0.5)
```

```
def passive(self):  
    self.position = Vec2(0.4, -0.6)
```

```
hand = Hand()
```

FILE SETTINGS.PY

File này sẽ chứa các thiết lập đường dẫn:



< Dev Alex />

```
from pathlib import Path  
  
BASE_DIR = Path(__file__).resolve().parent  
  
MODEL_PATH = BASE_DIR / 'assets' / 'models'  
SFX_PATH = BASE_DIR / 'assets' / 'sfx'  
TEXTURE_PATH = BASE_DIR / 'assets' / 'textures'
```

FILE MAIN.PY

File chính sẽ import các module và chạy ứng dụng:



< Dev Alex />

```
from turtle import *
from ursina import *
from ursina.prefabs.first_person_controller import
FirstPersonController

app = Ursina()

# Tải các tài nguyên
from src.resources.textures import grass_texture, stone_texture,
brick_texture, dirt_texture, wood_texture, sky_texture,
arm_texture, punch_sound

# Tắt nút thoát
window.exit_button.visible = False

# Cập nhật mỗi khung hình
from src.events.update import update, block_pick

# Import các lớp trong scene
from src.scene.voxel import Voxel
from src.scene.sky import Sky
from src.scene.hand import hand

# Tạo các khối ban đầu
for z in range(50):
    for x in range(50):
        voxel = Voxel(position=(x, 0, z))

# Khởi tạo các thực thể khác
player = FirstPersonController()
sky = Sky()

# Chạy ứng dụng
app.run()
```

