

Cấu trúc dữ liệu trong Unity

Trong Unity, có một số cách để lưu trữ dữ liệu. Dưới đây là một số phương pháp phổ biến:

Mảng (Array): Mảng là một kiểu dữ liệu cơ bản trong Unity, có kích thước cố định và có thể được thay đổi kích thước bằng cách sử dụng hàm `System.Array.Resize`. Mảng là lựa chọn tốt cho hiệu suất, đặc biệt là khi bạn đang phát triển trò chơi cho iPhone. Để khai báo một mảng, bạn có thể sử dụng cú pháp sau:

```
//Khai báo
Type[] array = new Type[lengthOfArray];
//Khai báo một mảng với một tập hợp các số nguyên
int[] numbers = new int[10];
//Khai báo một mảng với một tập hợp các đối tượng trong trò chơi
GameObject[] objects = new GameObject[16];
//Lấy số lượng đối tượng trong mảng với chỉ số i
int count = myArray.Length;
//Gán một giá trị cho đối tượng có chỉ số i trong mảng
myArray[i] = newValue;
//Gán một giá trị mới cho chỉ số mảng
TheType thisValue = myArray[i];
//Thay đổi kích thước mảng
System.Array.Resize(ref myArray, size);
```

Generic List: Generic List là một cấu trúc dữ liệu có kích thước động và hỗ trợ thêm, lấy và xóa các phần tử. Generic List cho phép bạn chỉ định loại dữ liệu mà List sẽ chứa khi khai báo. Đây là một lựa chọn tiện lợi và có hiệu suất tốt hơn so với mảng. Để khai báo một Generic List, bạn có thể sử dụng cú pháp sau:

```
List<Type> listName = new List<Type>();//Khai báo một danh sách số nguyên
List<int> someNumbers = new List<int>();
//Khai báo một danh sách các đối tượng trong trò chơi
List<GameObject> enemies = new List<GameObject>();
//Thêm phần tử vào danh sách
myList.Add(theItem);
//Gán một phần tử trong danh sách
myList[i] = newItem;
//Gán giá trị của một phần tử trong danh sách
Type thisItem = List[i];
//Xóa phần tử trong danh sách
myList.RemoveAt(i);
```

Hashtable: Hashtable là một cấu trúc dữ liệu trong Unity cho phép bạn lưu trữ các cặp key-value. Key được sử dụng để tìm kiếm giá trị tương ứng trong Hashtable.

Đây là một lựa chọn tốt khi bạn cần tìm kiếm nhanh một phần tử từ một tập hợp dữ liệu. Để khai báo một Hashtable, bạn có thể sử dụng cú pháp sau:

```
//Khai báo một hashtable
Hashtable myHashtable = new Hashtable();
//Gán một giá trị với khóa vào hashtable
myHashtable[anyKey] = newValue;
//Lấy giá trị từ hashtable, cần ép kiểu (casting)
ValueType thisValue = (ValueType)myHashtable[theKey];
//Lấy số lượng các mục trong hashtable
int count = myHashtable.Count;
//Xóa mục trong hashtable
myHashtable.Remove(theKey);
```

Generic Dictionary: Generic Dictionary cung cấp một cấu trúc tương tự như Hashtable, nhưng bạn phải xác định kiểu dữ liệu cho cả key và value khi khai báo. Generic Dictionary có hiệu suất tốt hơn Hashtable và không yêu cầu phép ép kiểu (casting). Để khai báo một Generic Dictionary, bạn có thể sử dụng cú pháp sau:

```
//Khai báo một từ điển (dictionary) với chuỗi làm khóa (key) và giá trị (value)
Dictionary<string, string> myDictionary = new Dictionary<string, string>();
//Gán một giá trị vào từ điển với khóa (key)
myDictionary[anyKey] = newValue;
//Lấy giá trị từ từ điển với khóa (key)
string thisValue = myDictionary[theKey];
//Lấy số lượng các mục trong từ điển
int count = myDictionary.Count;
//Xóa mục khỏi từ điển
myDictionary.Remove(theKey);
```

Ngoài ra, Unity cũng cung cấp các công cụ lưu trữ dữ liệu khác như PlayerPrefs để lưu trữ các giá trị cơ bản như số nguyên, số thực và chuỗi trong quá trình chạy ứng dụng.

Quá trình chọn lựa giữa các phương pháp lưu trữ dữ liệu trong Unity phụ thuộc vào yêu cầu của trò chơi và tình huống cụ thể.

PlayerPrefs: PlayerPrefs cho phép lưu trữ và truy xuất các giá trị cơ bản như số nguyên, số thực và chuỗi. Dữ liệu được lưu trữ trong Registry (Windows) hoặc tệp tin cấu hình (macOS và Linux). Đây là một phương pháp đơn giản và thuận tiện, nhưng không phù hợp cho việc lưu trữ dữ liệu phức tạp hoặc lớn.

SQLite: Unity hỗ trợ thư viện SQLite để làm việc với cơ sở dữ liệu SQLite. Bằng cách sử dụng SQLite, bạn có thể tạo, truy vấn và cập nhật cơ sở dữ liệu phức tạp. Điều này rất hữu ích khi bạn cần lưu trữ và truy xuất dữ liệu đa dạng như trạng thái trò chơi, thành tích cao, hoặc dữ liệu người chơi.

JSON và XML: Bạn có thể sử dụng các thư viện JSON hoặc XML để lưu trữ dữ liệu dưới dạng văn bản. JSON (JavaScript Object Notation) và XML (eXtensible Markup Language) đều là định dạng phổ biến để lưu trữ và truyền dữ liệu dạng cấu trúc. Unity cung cấp các API để phân tích và tạo ra dữ liệu JSON hoặc XML, giúp bạn lưu trữ và truy xuất dữ liệu theo cách tùy ý.