TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

Logo

Description automatically generated

**TIỂU LUẬN CUỐI KỲ**

**Môn học: Cấu trúc dữ liệu và giải thuật**

**Xây dựng trò chơi nối từ Tiếng Việt**

Giảng viên: PGS.TS. Hoàng Văn Dũng

**Danh sách sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **Mã số SV** | **Họ và tên** | **Tỷ lệ**  **đóng góp** |
| 20133079 | Lã Minh Phúc | 120% |
|  |  |  |
|  |  |  |

*TP. Hồ Chí Minh, tháng 12 năm 2021*

# MỤC LỤC

# MỞ ĐẦU

Ngày nay với sự phát triển của internet thì mạng xã hội đã trở thành một phần trong cuộc sống mỗi chúng ta. Sử dụng mạng xã hội trong mục các mục đích như giao tiếp, chia sẻ, giải trí giúp chúng ta luôn cập nhập thông tin một cách nhanh chóng. Tuy nhiên chính vì nhu cầu trao đổi thông tin nhanh chóng nên nhiều người dùng đã sử dụng các từ vay mượn từ tiếng nước ngoài trong quá trình chia sẻ thông tin nhằm rút ngắn thời gian nhập liệu. Điều này dẫn tới việc lạm dụng từ mượn làm mất đi vẻ đẹp và sự phong phú của Tiếng Việt khi ngày càng nhiều các từ Tiếng Việt dần bị thay thế như “tạm biệt”, “xin chào”, …

# CƠ SỞ LÝ THUYẾT

## Công cụ và môi trường để lập trình

### Xây dựng chương trình

* Visual Studio Community 2019, Desktop Development with C++
* Thư viện dpp (hỗ trợ kết nối với bot Discord)

### Xử lý, phân tích dữ liệu

* Python 3.8
* Anaconda, Jupyter Notebook
* Từ điển Việt Việt; Hồ Ngọc Đức

## Các phương pháp, kĩ thuật sử dụng

* Danh sách sử dụng mảng
* Danh sách sử dụng liên kết đơn
* Bảng băm
* Cây nhị phân tìm kiếm
* Tìm kiếm nhị phân

# PHÂN TÍCH, THIẾT KẾ GIẢI PHÁP

## Dữ liệu đầu vào của chương trình

### dictionary.txt

Là một tệp văn bản chứa các từ vựng có hai âm tiết của Tiếng Việt mỗi từ được ngăn cách nhau bằng dấu phẩy (kí hiệu: ,), các từ được sắp xếp tăng dần theo giá trị mã hoá từ của chương trình.

Các từ vựng trong trò chơi được xem là có nghĩa nếu tồn tại trong tệp này. Trong chương trình tệp sẽ được đọc và ghi dữ liệu vào một bảng băm.

Là kết quả của quá trình phân tích Từ điển tiếng Việt.

### start\_word.txt

Là một tệp văn bản chứa các âm để bắt đầu một từ trong Tiếng Việt. Dòng đầu tiên là số lượng phần tử các âm, dòng thứ hai là các âm để bắt đầu bằng một từ trong Tiếng Việt.

Các âm trong tệp được chương trình sử dụng làm âm ngẫu nhiên để bắt đầu trò chơi. Trong chương trình tệp sẽ được đọc và ghi vào một mảng.

Là kết quả của quá trình phân tích Từ điển tiếng Việt.

### encoder.txt

Là một tệp văn bản có cấu trúc được chương trình sử dụng để phân tách và mã hoá các âm trong Tiếng Việt.

Là kết quả của quá trình phân tích dữ liệu bằng Python.

*Cấu trúc tệp:*

* Dòng thứ nhất: Các số lượng phần tử của các dữ liệu được ngăn cách nhau bằng khoảng trắng, các dữ liệu tương ứng lần lượt là: các âm đặc biệt, các kí tự bắt đầu của một âm, các kí tự kết thúc của một âm, các kí tự có dấu trong Tiếng Việt.
* Dòng thứ hai: Các chuỗi được ngăn cách nhau bằng dấu phẩy, chứa các âm đặc biệt không thể phân tách và mã hoá theo cách thông thường của chương trình.
* Dòng thứ ba: Các chuỗi được ngăn cách nhau bằng dấu phẩy, chứa các kí tự bắt đầu của một âm trong Tiếng Việt, được sắp xếp theo mã hoá UTF8.
* Dòng thứ bốn: Các chuỗi được ngăn cách nhau bằng dấu phẩy, chứa các kí tự kết thúc của một âm trong Tiếng Việt, được sắp xếp theo mã hoá UTF8.
* Dòng thứ năm: Các chuỗi được ngăn cách nhau bằng dấu phẩy, chứa các kí tự có dấu trong Tiếng Việt.

## Các cấu trúc dữ liệu

### Sử dụng bảng băm để chứa các từ vựng (class Dictionary)

*Cấu trúc bảng băm:*

* Khoá (key): Âm bắt đầu.
* Giá trị (value): Một danh sách mảng (con trỏ) chứa các mã hoá của các âm kết thúc của các từ vựng tương ứng với giá trị băm của âm bắt đầu. Mảng được sắp xếp tăng dần theo giá trị mã hoá của âm.
* Hàm băm (hash function): Hàm mã hoá âm Tiếng Việt. Giá trị trong đoạn từ 0 đến 38877. Là hàm băm không có va chạm.

*Đọc tệp “dictionary.txt”:*

* Lấy lần lượt các từ vựng trong tệp
* Với một một khoá tạo một danh sách liên kết chứa các mã hoá của các âm kết thúc tương ứng
* Chuyển danh sách liên kết chứa các âm kết thúc thành con trỏ danh sách mảng chứa các âm kết thúc để lưu trữ và hỗ trợ thực hiện tìm kiếm nhị phân.

*Tra cứu tính có nghĩa của một từ có hai âm tiết:*

* Tách từ thành hai âm riêng biệt, băm âm bắt đầu của từ sẽ có được vị trí danh sách mảng chứa các mã hoá của các âm kết thúc của các từ vựng tương ứng với giá trị băm của âm bắt đầu.
* Sử dụng thuật toán tìm kiếm nhị phân giá trị mã hoá của âm kết thúc trên danh sách mảng tìm được. Nếu giá trị đó tồn tại trong mảng nghĩa là từ có nghĩa, ngược lại từ sẽ là vô nghĩa.

Graphical user interface, application

Description automatically generated

*Ví dụ tra cứu âm “gà” trong bảng băm*

Graphical user interface, Teams

Description automatically generated

*Lưu đồ khối kiểm tra tính có nghĩa của một từ có hai âm tiết trong Tiếng Việt*

*Các phương thức, thuộc tính:*

private:

static Dictionary\* instance; // Singleton Instance

Dictionary(); // Đọc tệp, khởi tạo dữ liệu cho bảng băm

MyArray<uint16\_t>\*\* buckets;

int hashFunction(const std::wstring& sound); // Hàm băm không có va chạm

int binarySearch(uint16\_t arr[], const int& numElems, const uint16\_t& value); // Tìm kiếm nhị phân trong mảng uint16\_t đã được sắp xếp tăng dần

public:

static Dictionary\* getInstance();

bool checkMeaning(const std::wstring& word); // Kiểm tra tính có nghĩa của một từ có 2 âm tiết

MyArray<uint16\_t>\* getLastSounds(const int& index); // Lấy danh sách mảng các encode của các âm kết thúc

### Sử dụng cây nhị phân tìm kiếm lưu trữ các từ đã sử dụng trong màn chơi (class History)

Kiểu dữ liệu lưu trữ: số nguyên không âm 32 bits (Giá trị mã hoá một từ Tiếng Việt 2 âm tiết).

Dữ liệu lưu trong cây nhị phân là duy nhất.

A picture containing timeline

Description automatically generated

*Lưu đồ khổi thuật toán thêm một phần tử vào cây*

*Các phương thức, thuộc tính:*

private:

int count; // Số lượng phần tử của cây

TreeNode<uint32\_t>\* root; // Gốc của cây nhị phân

void remove(TreeNode<uint32\_t>\*& node); // Xoá tất một node của cây, các node con của cây cũng sẽ bị xoá

public:

History();

~History();

bool isExisted(const uint32\_t& encode); // Kiểm tra từ đã tồn tại trong cây trước đó

bool add(const std::wstring& x); // Thêm giá trị encode của x vào cây

int getCount();

void removeAll(); // Xoá tất cả các node của cây

### Danh sách liên kết đơn (class ArrayBuilder)

Được sử dụng để lưu trữ dữ liệu sau đó có thể chuyển qua danh sách mảng khi số lượng phần tử không đổi nhằm tiết kiệm bộ nhớ và thực hiện tìm kiếm nhị phân.

*Các phương thức:*

void add(T x) // Thêm một phần tử vào đầu danh sách liên kết đơn

MyArray<T>\* toMyArray() // Chuyển thành danh sách sử dụng mảng (con trỏ), danh sách mảng trả về sẽ ngược với danh sách liên kết

### Danh sách mảng (class MyArray)

*Các phương thức:*

T& operator[](const int& index) // Truy cập đến một phần tử của mảng

int getSize() // Lấy số lượng phần tử của mảng

## Thuật toán

### Tách dấu một âm trong Tiếng Việt

*Đọc dữ liệu dấu từ tệp:*

Trong tệp *encoder.txt*, phần tử cuối của dòng đầu tiên sẽ là số lượng các kí tự có thể có dấu trong Tiếng Việt giá trị này được ghi vào biến *sizeTones* trong chương trình. Dòng cuối cùng của tệp là chuỗi các kí tự có dấu trong Tiếng Việt, các dấu lần lượt là thanh, sắc, huyển, hỏi, ngã, nặng, các chuỗi này được ghi vào mảng *tones* trong chương trình.

|  |
| --- |
| encoder.txt |
| 11 28 152 12  …  aáàảãạ,eéèẻẽẹ,iíìỉĩị,oóòỏõọ,uúùủũụ,yýỳỷỹỵ,âấầẩẫậ,êếềểễệ,ôốồổỗộ,ăắằẳẵặ,ơớờởỡợ,ưứừửữự |

*Dữ liệu để đọc mảng tones của tệp encoder.txt*

Thuật toán sẽ duyệt lần lượt các kí tự của chuỗi âm, khi đến kí tự có dấu sẽ thay thế bằng kí tự không dấu và trả về giá trị tương ứng với dấu của kí tự.

/// <summary>

/// Thay một kí tự có dấu trong âm bằng kí tự không dấu và trả về giá trị của dấu.

/// </summary>

/// <param name="sound">Âm cần tách dấu, sau khi tách sẽ mất dấu</param>

/// <returns>Giá trị của dấu</returns>

int removeTone(std::wstring& sound)

{

    int lenght = sound.size();

    int tone = 0;

    for (int i = lenght - 1; i >= 0; i--)

    {

        if (sound[i] >= 224) // Giá trị bé nhất của một kí tự có dấu

        {

            for (int j = 0; j < sizeTones; j++)

            {

// Duyệt các kí tự có dấu (trừ dấu thanh)

                for (int k = 1; k < 6; k++)

                {

                    if (sound[i] == tones[j][k])

                    {

                        sound[i] = tones[j][0];

                        tone = k;

                        return tone;

                    }

                }

            }

        }

    }

    return tone;

}

Code C++ hàm tách dấu của một âm Tiếng Việt

### Mã hoá một âm trong Tiếng Việt

Phân tách một âm trong Tiếng Việt thành 3 thành phần: phần bắt đầu, phần kết thúc và phần dấu.

Sử dụng dữ liệu *unsigned short (uint16\_t)* để lưu trữ 3 thành phần này, cụ thể:

* 8 bits đầu: Phần kết thúc
* 5 bits tiếp theo: Phần bắt đầu
* 3 bits còn lại: Phần dấu

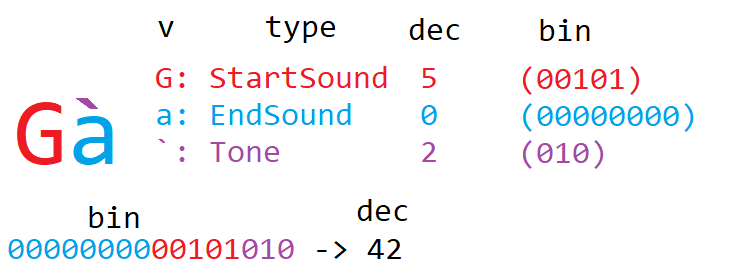
Bỏ qua phân biệt kí tự hoa, thường, vị trí đăt dấu trong âm. Ví dụ: âm “Hoá” sẽ có cùng mã hoá với âm “hóa”

Các âm đặc biệt được tính phần bắt đầu là 31, phần kết thúc là vị trí của âm trong mảng các âm đặc biệt.

A picture containing diagram

Description automatically generated

Lưu đồ khối thuật toán mã hoá một âm trong Tiếng Việt



*Hình Nal: Mô tả phân tách và mã hoá âm “gà”*

### Mã hoá một từ có hai âm tiết trong Tiếng Việt

Tách một từ thành hai âm tiết sử dụng 16 bits đầu để lưu trữ từ đầu tiên và 16 bits tiếp theo để lưu trữ từ thứ hai.

Graphical user interface

Description automatically generated

Lưu đồ khối thuật toán mã hoá một từ hai âm tiết

# THỰC NGHIỆM, ĐÁNH GIÁ, PHÂN TÍCH KẾT QUẢ

## Cách chơi

Chương trình sẽ tạo một âm ngẫu nhiên để bắt đầu trò chơi, ta cũng có thể cài đặt âm bắt đầu bằng lệnh “!set <StartSound>”.

Các người chơi sẽ lần lượt đưa ra đáp án phù hợp bằng cách nhắn tin đáp án. Một người chơi không thể trả lời liên tiếp nhiều lần.

Khi không còn đáp án phù hợp trong từ điển của chương trình, trò chơi sẽ kết thúc với người chiến thắng là người cuối cùng đưa ra đáp án.

Một đáp án được xem là phù hợp nếu:

* Là một từ Tiếng Việt có hai âm tiết.
* Âm cuối cùng của từ trước sẽ là âm bắt đầu của từ tiếp theo.
* Tồn tại trong từ điển của chương trình (có ý nghĩa).

Các lệnh chương trình: Bắt đầu bằng dấu chấm than (ký hiệu: !), sử dụng bằng cách nhắn tin như trả lời đáp án.

## Hướng dẫn thực thi phần mềm, kết quả giao diện phần mềm

!help: Hiển thị hướng dẫn chương trình

!set <StartSound>: Thiết lập âm <StartSound> thành âm để bắt đầu trò chơi

!info: Xem thông tin hiện tại của trò chơi

!reset: Bắt đầu lại trò chơi

!nocheckplayer: Bật/Tắt cho phép người chơi trả lời liên tiếp nhiều lần

Chuẩn bị một tài khoản discord, truy cập đường dẫn [tại đây](https://discord.gg/Hs62wHxEEh) để tham gia kênh chat đã chứa tài khoản bot mặc định.

Tài khoản bot và kênh chat mặc định:

* Token bot: OTA1ODM1ODQ5NzA5NjYyMjA5.YYP3YA.YRcUBf-yBWr\_yEgOhZUKKMvegNk
* Id channel: 905849241577074728

Ta cũng có thể tạo mới một kênh chat discord sau đó truy cập đường dẫn [tại dây](https://discordapp.com/api/oauth2/authorize?client_id=905835849709662209&scope=bot&permissions=1) để thêm tài khoản bot discord vào kênh chat hoặc tạo một tài khoản bot mới để chương trình kết nối.

Khởi động chương trình nhập lần lượt token bot và id channel, sau khi bot đăng nhập sẽ nhắn tin thông báo trò chơi đã khởi động. Chương trình console sẽ ghi lại và hiển thị các hoạt động chương trình đã thực hiện.

A screenshot of a computer

Description automatically generated with low confidence

Lưu ý: Luật chơi yêu cầu một người chơi không được trả lời liên tiếp nhiều lần, ta có thể bật/tắt điều kiện kiểm tra này bằng lệnh “!nocheckplayer” để thử nghiệm dễ dàng hơn

# PHẦN KẾT LUẬN

## Đánh giá những kết quả đã thực hiện được

## Định hướng phát triển

Nâng cấp xây dựng các tính năng như tính điểm, xếp hạng cho từng người chơi:

Kết nối với một cơ sở dữ liệu, lưu trữ các điểm số khi người chơi có đáp án đúng và xếp hạng người chơi.

Cải thiện chất lượng từ điển, xây dựng chức năng góp ý từ người chơi: Từ điển sử dụng từ “Dự án Từ điển tiếng Việt miễn phí” đã cũ nên chưa được đầy đủ các từ mới được bổ sung. Người chơi sẽ góp ý các từ vựng và tính năng để hoàn thiện chương trình.

# TÀI LIỆU THAM KHẢO

* Hồ Ngọc Đức (2004), Dự án Từ điển tiếng Việt miễn phí, <http://informatik.uni-leipzig.de/~duc/Dict/>. Dữ liệu từ điển của trò chơi
* TriviaBot, D++ Developer Documentation. <https://dpp.dev/index.html>. Kết nối chương trình với bot Discord
* Discord, Discord Official API Documentation, <https://discord.com/developers/docs/intro>. Tạo bot discord

# PHỤ LỤC

## Giá trị mã hoá các thành phần của âm

Bảng Nal: Giá trị mã hoá phần bắt đầu

Bảng Nal: Giá trị mã hoá phần kết thúc

|  |  |
| --- | --- |
| **Phần dấu** | **Giá trị** |
| Thanh | 0 |
| Sắc | 1 |
| Huyền | 2 |
| Hỏi | 3 |
| Ngã | 4 |
| Nặng | 5 |

Bảng Nal: Giá trị mã hoá phần dấu

Bảng Nal: Giá trị mã hoá âm đặc biệt