

TRƯỜNG ĐẠI HỌC NHA TRANG

KHOA CÔNG NGHỆ THÔNG TIN

**THỰC TẬP CƠ SỞ**

Đề Tài

**Chủ đề: Lập trình ứng dụng Winform**

**Ứng dụng: Lập trình Game “Ai là triệu phú”**

**** Giáo viên hướng dẫn: **Mai Cường Thọ**.

* Thực hiện đề tài: Nhóm 1.

***Thành viên nhóm:***

1. **Võ Ngọc Hiện**

2. **Võ Lương Sĩ Hoàng**

3. **Lê Xuân Đại**

**Lớp: 60.CNTT-3**

Năm Học: 2020 - 2021

**MỤC LỤC**

[**NỘI DUNG** 3](#_Toc61969007)

[**I. Nêu bài toán:** 3](#_Toc61969008)

[1. Tạo game : “Ai là triệu phú” trên giao diện Winform: 3](#_Toc61969009)

[2. Yêu cầu thực hiện: 3](#_Toc61969010)

[**II. Cách giải quyết:** 4](#_Toc61969011)

[1.Hướng giải quyết 4](#_Toc61969012)

[2. Quy trình thiết kế ứng dụng 5](#_Toc61969013)

[B1: Thiết kế giao diện cơ bản: 5](#_Toc61969014)

[B2: Lập trình tính năng xử lí cơ bản: 8](#_Toc61969015)

[B3: Lập trình tính năng xử lí bổ sung: Phần này để phát triển thêm cho game có nhiều tính năng thú vị, hấp dẫn hơn. Ví dụ như: 12](#_Toc61969016)

[B4: Hoàn tất thiết kế giao diện: 12](#_Toc61969017)

[**III. Cài đặt thực hiện:** 12](#_Toc61969018)

[3.1 Mã nguồn: 7 cửa sổ frmBatDau.cs, frmDienthoai\_Nguoithan.cs, frmLuatchoi.cs, frmMain.cs, frmThongTin.cs, frmToTuVan.cs, frmUpdateCauHoi.cs 12](#_Toc61969019)

[3.2 Giao diện : 12](#_Toc61969020)

[**IV. Tài liệu :** 14](#_Toc61969021)

[**V. Phân công:** 14](#_Toc61969022)

[1. Ý tưởng về thiết kế giao diện, chức năng: 14](#_Toc61969023)

[2. Giải quyết bài toán: 14](#_Toc61969024)

[3. Tiến hành code mã nguồn: 14](#_Toc61969025)

[4. Tiến hành kiểm thử: Cả nhóm 14](#_Toc61969026)

[5. Viết báo cáo và tổng hợp: 14](#_Toc61969027)

# **NỘI DUNG**

# **I. Nêu bài toán:**

## 1. Tạo game : “Ai là triệu phú” trên giao diện Winform:

* Game “Ai là triệu phú” là gì?

* Ý tưởng: Game “Ai là triệu phú” được chuyển thể từ phiên bản gameshow phát trên VTV3 tối thứ 3 hàng tuần và trở nên quen thuộc với tất cả mọi người. Cơ hội để ngồi trên chiếc ghế nóng của VTV3 tại trường quay S9 là rất khó cho hầu hết mọi người, tựa game này ra đời đã đáp ứng được yêu cầu đó.

Đây là một thể loại game chơi trên chính sự hiểu biết của bạn về tất cả mọi vấn đề từ cổ chí kim, nhiều lĩnh vực trong cuộc sống với hình thức trả lời câu hỏi trắc nghiệm khách quan 4 lựa chọn với 3 quyền trợ giúp: Gọi điện thoại cho người thân, hỏi ý kiến khán giả trong trường quay và quyền ngưng cuộc chơi ở mốc bất kỳ để bảo toàn số điểm thưởng đang có.

Với giao diện đồ họa sống động, âm thanh đa phương tiện với lời dẫn chương trình thật của MC Lại Văn Sâm, đây thật sự là một phần mềm vừa chơi vừa học thú vị nhất là cho các bạn học sinh – sinh viên dùng cho ôn luyện kiến thức.

## 2. Yêu cầu thực hiện:

* Yêu cầu:
* Sinh viên bắt buộc phải tạo tài khoản trên Github để tổ chức lưu trữ và quản lý phiên bản mã nguồn (gửi Link cho GVHD), push kết quả định kỳ ½ ngày nếu có thay đổi mã nguồn. Sinh viên thực hiện trên nhóm 3 SV. Lập trình hướng đối tượng.
* Sinh viên viết chương trình ai là triệu phú.
* Phân tích bài toán game Ai là triệu phú, mô tả các chức năng, các dữ liệu cần tổ chức lưu trữ, ..
* Tổ chức cấu trúc dữ liệu phù hợp để lưu trữ ngân hàng câu hỏi và lập trình các chức năng cơ bản. Xây dựng ứng dụng với 2 phân hệ: Quản trị (để thêm/xóa/sửa đề thi, câu hỏi và các thông tin khác), phân hệ người chơi cho phép sử dụng các phím A, B, C, D để trả lời.
* Các chức năng và giao diện chương trình được tổ chức tốt, thân thiện, dễ dùng (sử dụng các ký tự ở phần mở rộng của bảng mã ASCII để trình bày, hoặc sử dụng lập trình đồ họa )
* Giới thiệu
  + Chương trình được viết trên ngôn ngữ C#

# **II. Cách giải quyết:**

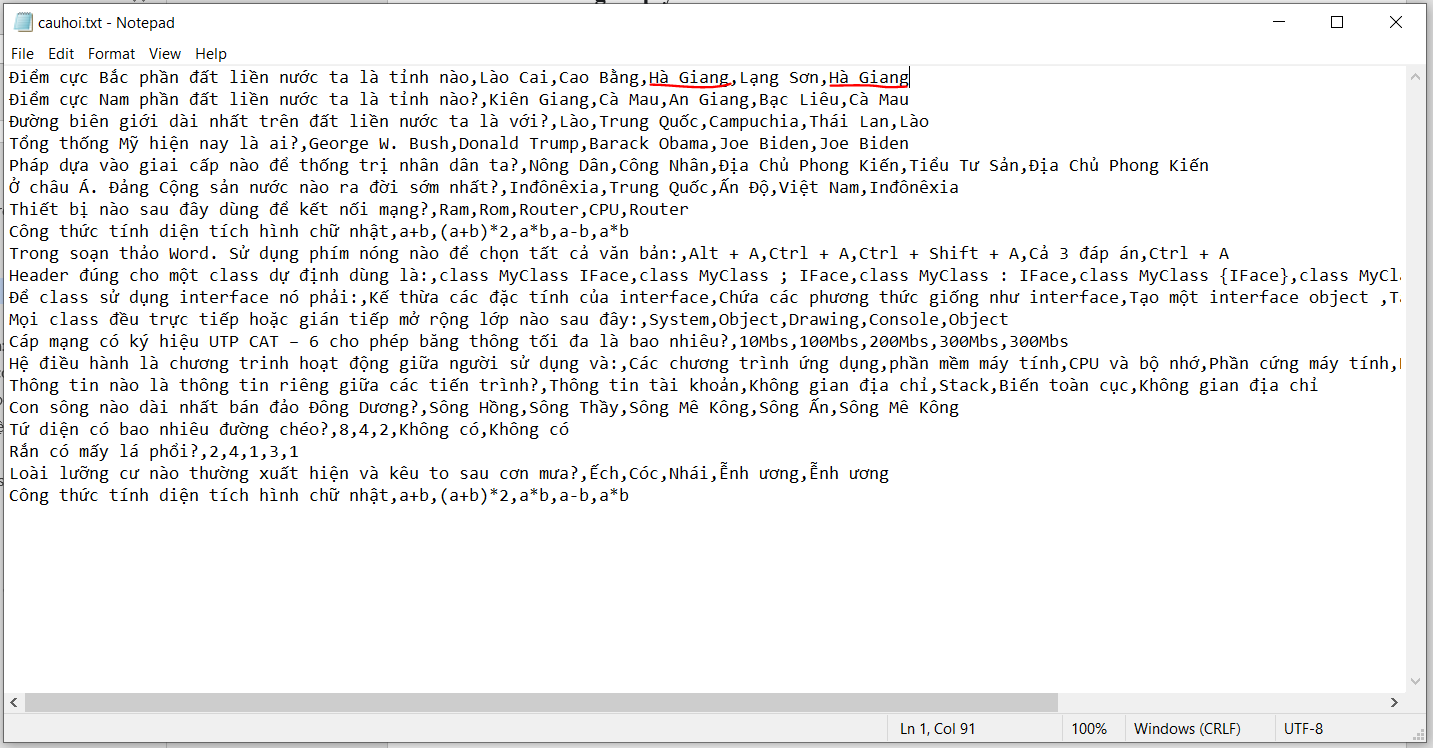
## 1.Hướng giải quyết

- Lựa chọn lập trình trên Winform của C# ( Visual studio): Tiết kiệm thời gian, dễ thao tác.

- Dựa vào yêu cầu thực hiện bài toán, ta xử lí như sau:

+ Mỗi câu hỏi sẽ có 4 phương án khác nhau trong đó có 1 phương án đúng.

=> Tạo sẵn một file cauhoi.txt gồm: câu hỏi, phương án a, phương án b, phương án c, phương án d, câu trả lời đúng.



Hình 1

+ Luôn cố định 1 danh sách câu hỏi và 4 phương án để ta chọn và sẽ có 1 phương án đúng như (hình 1)

=> Vì thế khi tạo ra game này với mỗi câu hỏi chúng ta đã có sẵn đáp án đúng và dựa theo đáp án đúng để tăng số tiền tương ứng với số câu trả lời đúng.

==========================

## 2. Quy trình thiết kế ứng dụng

B1. Thiết kế giao diện cơ bản

B2. Lập trình tính năng xử lý cơ bản

B3. Lập trình tính năng xử lý bổ sung

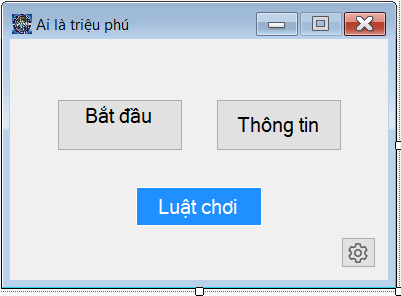
B4. Hoàn tất thiết kế giao diện

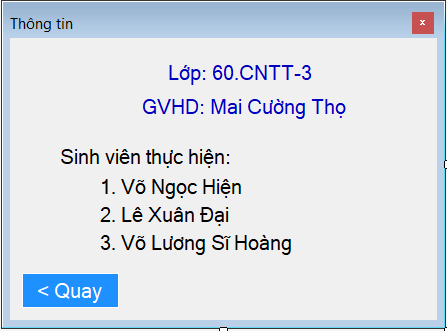
==========================

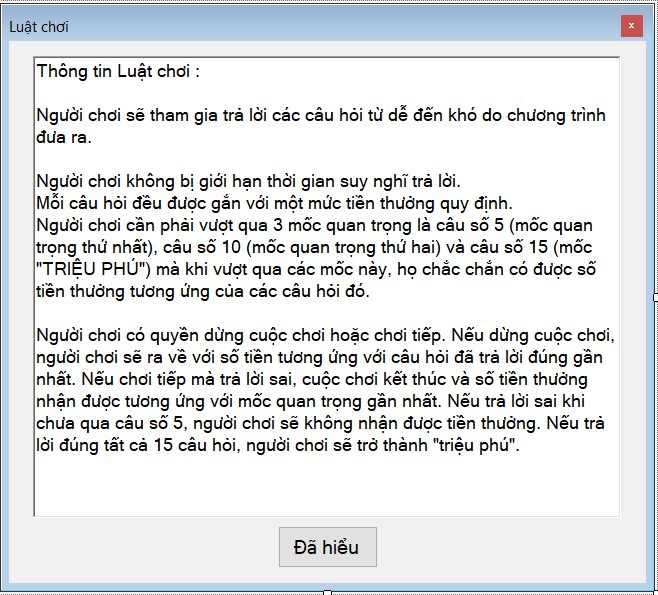
Quy trình thiết kế game:

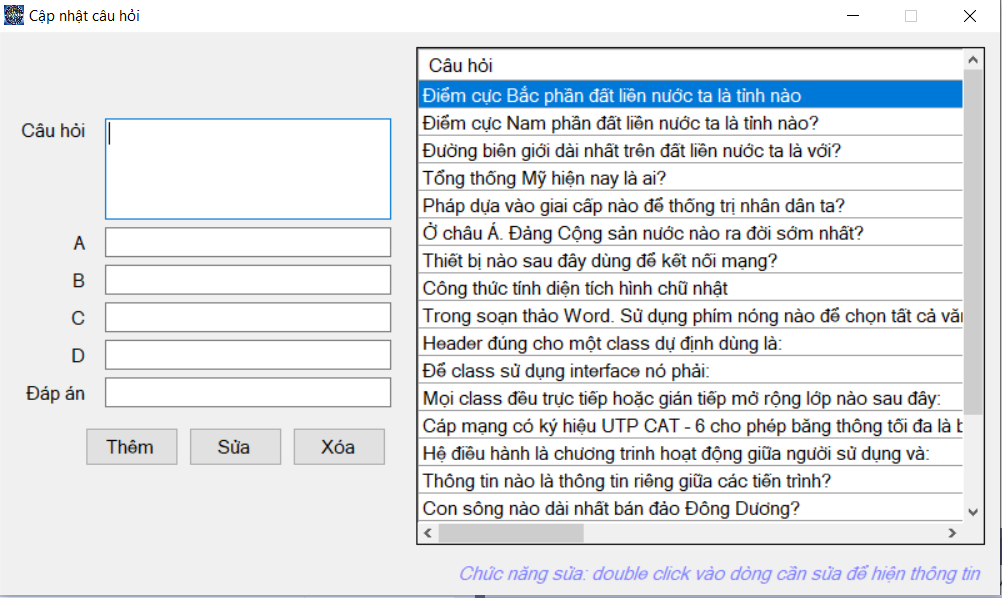
### B1: Thiết kế giao diện cơ bản:

1. Tạo form cơ bản:

* Chúng ta tạo form bắt đầu gồm các button đặt thành các nút: Bắt đầu, thông tin, luật chơi và setting. 
* Trong các button Bắt đầu, thông tin, luật chơi và setting sẽ chứa các form con như form main, form thông tin, form luật chơi, form updatecauhoi.
* Khi ta click vào nút bắt đầu sẽ xuất hiện giao diện người chơi. Trong giao diện người chơi sẽ có các: Button, label, dataGridveiw và pictureBox.
* Khi ta click vào nút Thông tin sẽ xuất hiện người làm chương trình.



* Khi ta click vào nút Luật chơi sẽ xuất hiện luật chơi để cho người chơi dễ chơi game hơn
* Khi ta click vào Nút setting sẽ xuất hiện thông tin các câu hỏi vào chức năng thêm, xoá, sửa câu hỏi.



### B2: Lập trình tính năng xử lí cơ bản:

1. Hiển thị câu hỏi trong file text lên form:

* Gán đường dẫn tới file cauhoi.txt và đọc tất cả dữ liệu có trong file cauhoi.txt sau đó ta cần tạo 1 public câu hỏi và khai báo 1 biến gọi là Datatable để chứa các dữ liệu của bảng câu hỏi trong file cauhoi.txt.

**Code:**

private DataTable \_data;

public string file = string.Format(@"{0}\data\cauhoi.txt", Application.StartupPath); // đường dẫn đến file cauhoi.txt

public CauHoi()

{

// khai báo DataTable chứa dữ liệu của bảng cauhoi trong csdl

\_data = new DataTable(); // tạo mới DataTable

\_data.Columns.Add("Cau", typeof(int)); // thêm cột cho DataTable, kèm theo kiểu dữ liệu cho cột

\_data.Columns.Add("CauHoi", typeof(string));

\_data.Columns.Add("A", typeof(string));

\_data.Columns.Add("B", typeof(string));

\_data.Columns.Add("C", typeof(string));

\_data.Columns.Add("D", typeof(string));

\_data.Columns.Add("DapAn", typeof(string));

string[] lines = File.ReadAllLines(file); // đọc tất cả dữ liệu trong cauhoi.txt

for (int i = 0; i < lines.Length; i++) // duyệt qua từng dòng

{

string[] array\_line = lines[i].Split(','); // cắt tại vị trí có dấu phẩy (,) => chuyển thành mảng

\_data.Rows.Add(new object[] {i+1, array\_line[0], array\_line[1], array\_line[2], array\_line[3], array\_line[4], array\_line[5] } ); // thêm mới dòng vào DataTable

}

BangCauHoi = \_data;

}

* Sau khi gán đường dẫn chúng ta tạo lệnh load câu hỏi sẽ lấy câu hỏi từ file text lên chương trình và nhận dữ liệu từ file text đổ lên dataGridveiw

public override void LoadCauHoi(DataGridView gv)

{

gv.Rows.Clear();

string[] lines = File.ReadAllLines(file); // đọc tất cả dữ liệu trong cauhoi.txt

for (int i = 0; i < lines.Length; i++) // duyệt qua từng dòng

{

string[] array\_line = lines[i].Split(','); // cắt tại vị trí có dấu phẩy (,) => chuyển thành mảng

gv.Rows.Add(array\_line); // thêm mới dòng vào DataGridView

}

}

* Mỗi lần trả lời hàm sẽ kiểm tra câu trả lời so với đáp án. Nếu đáp án đúng sẽ chuyển tới câu hỏi tiếp theo không bị lặp lại câu hỏi trước đó và gán số tiền thưởng cho người chơi trả lời đúng. Ngược lại nếu người chơi trả lời sai sẽ tự động dừng cuộc chơi, hoặc khi người chơi hoàn thành trò chơi bằng 15 câu trả lời đúng thì trò chơi sẽ tự động kết thúc và ra về với số tiền thưởng tương ứng với số câu trả lời đúng.

**Code:**

private void ChonDapAn(string cauTraLoi)

{

string cau = lbl\_number.Text.Replace(":", ""); // xóa bỏ dấu :

if (cauhoi.KiemTra(cau, cauTraLoi)) // kiểm tra câu trả lời so với đáp án

{

SoundPlayer sound = new SoundPlayer(string.Format(@"{0}\Data\dung.wav", Application.StartupPath)); // đường dẫn đến file âm thanh

sound.Play(); // phát âm thanh

MessageBox.Show("Xin chúc mừng");

int i = int.Parse(cau); // chuyển từ string => int

try

{

dataGridView1.Rows[14 - int.Parse(cau)].Selected = true; // di chuyển focus khi trả lời đúng

Next(i += 1); // sang câu kế tiếp

tienThuong = dataGridView1.Rows[15 - int.Parse(cau)].Cells[1].Value.ToString(); // gán số tiền thưởng khi trả lời đúng

}

catch

{

MessageBox.Show("Chúc mừng, Bạn là người thắng cuộc.!"); // trả hết 15 câu

btnDungCuocChoi.Enabled = false;

}

}

else

{

if (tienThuong != null)

{

MessageBox.Show(string.Format("Bạn ra về với số tiền thưởng: ${0}", tienThuong), "Rất tiếc, bạn trả lời sai.!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Bạn ra về với số tiền thưởng: $0", "Rất tiếc! Trả lời sai.!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

this.Close();

}

} }

1. Câu trả lời và đáp án, chuyển câu.

* Lấy câu trả lời của người chơi và xoá bỏ đi các ký tự thừa A. B. C. D. Sau đó so sánh câu trả lời của người chơi với đáp án chính xác của chương trình.

**Code:**

internal bool KiemTra(string cau, string cauTraLoi)

{

string traLoi = cauTraLoi.Substring(3); // Lấy câu trả lời, xóa bỏ ký tự thừa A. B. C. D.

DataRow[] row = \_data.Select(string.Format("[Cau] = '{0}' AND [DapAn] = '{1}'", cau, traLoi)); // so sánh câu trả lời với đáp án

if (row.Length == 1)

{

return true;

}

else

{

return false;

}

}

* Chuyển câu kế tiếp: Sau khi kiểm tra tính đúng/sai nếu câu trả lời sai thì thông báo “Rất tiêc! Trả lời sai” còn nếu câu trả lời đúng thì thông báo “Xin chúc mừng” đồng thời tự động chuyển sang câu tiếp theo lấy câu hỏi tiếp theo bằng hàm Next.

**Code:**

private void Next(int i = 1)

{

rows = cauhoi.BangCauHoi.Select(string.Format("Cau = {0}", i)); // lấy câu hỏi kế tiếp

// gán nội dung câu hỏi

lbl\_number.Text = string.Format("{0:}", i);

lbl\_cauHoi.Text = string.Format("{0}", rows[0]["CauHoi"]);

btnA.Text = string.Format("A. {0}", rows[0]["A"]);

btnB.Text = string.Format("B. {0}", rows[0]["B"]);

btnC.Text = string.Format("C. {0}", rows[0]["C"]);

btnD.Text = string.Format("D. {0}", rows[0]["D"]);

}

* Tô màu 3 cột mốc 5-10-15 và khởi tạo giải thưởng tương ứng từng mốc.

**Code:**

public override void Set\_GiaiThuong(DataGridView gv)

{

int[] tienThuong = { 200, 400, 600, 1000, 2000, 3000, 6000, 10000, 14000, 22000, 30000, 40000, 60000, 85000, 150000 }; // tạo mảng giá trị giải thưởng

for (int i = tienThuong.Length; i > 0; i--) // duyệt qua mảng

{

gv.Rows.Add(new object[] { i, tienThuong[i - 1] }); // thêm mới dòng vào DataGridView

}

gv.Rows[14].Selected = true;

gv.Rows[0].DefaultCellStyle.BackColor = ColorTranslator.FromHtml("#f2d9d9");

gv.Rows[5].DefaultCellStyle.BackColor = ColorTranslator.FromHtml("#f2d9d9");

gv.Rows[10].DefaultCellStyle.BackColor = ColorTranslator.FromHtml("#f2d9d9");

}

1. Chức năng đáp án bằng phím.

* Khi người chơi sử dụng các phím A, B, C, D sẽ chọn câu trả lời tương ứng từng phím

**Code:**

private void frmMain\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.A) // khi người chơi ấn phím A

{

ChonDapAn(btnA.Text);

}

if (e.KeyCode == Keys.B) // khi người chơi ấn phím B

{

ChonDapAn(btnB.Text);

}

if (e.KeyCode == Keys.C) // khi người chơi ấn phím C

{

ChonDapAn(btnC.Text);

}

if (e.KeyCode == Keys.D) // khi người chơi ấn phím D

{

ChonDapAn(btnD.Text);

}

}

### B3: Lập trình tính năng xử lí bổ sung: Phần này để phát triển thêm cho game có nhiều tính năng thú vị, hấp dẫn hơn. Ví dụ như:

* Luôn có điểm cho từng câu (điểm này có thể cố định hoặc tùy theo mức độ khó mỗi câu sẽ khác nhau) – Tùy theo điểm mà sẽ có số tiền tương ứng.

TH1: Điểm luôn cố định mỗi câu.

TH2: Điểm tùy theo cột mốc (15 câu sẽ có số tiền thưởng tương ứng).

* Chức năng hỏi ý kiến khán giả trong trường quay: Người chơi sẽ được khán giả trong trường quay gợi ý một ý kiến cho là đúng nhất.
* Chức gọi điện thoại cho người thân: Người chơi sẽ gọi cho 1 người thân để nhờ sự trợ giúp trong vòng 10s và đưa ra câu trả lời của mình.
* Chức năng thêm, xoá, sửa câu hỏi: Người lập trình có thể thêm, xoá hoặc sửa câu hỏi tuỳ ý khi sử dụng chương trình.
* Chức năng trả lời bằng phím: người chơi có thể sử dụng phím A,B,C,D để chọn phương án cho mình.

### B4: Hoàn tất thiết kế giao diện:

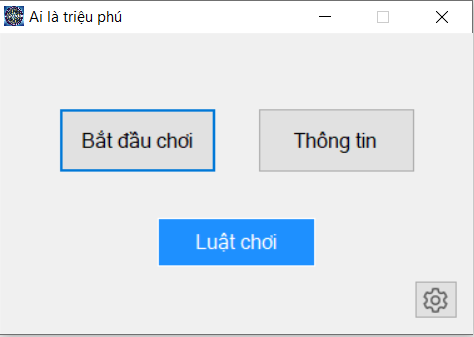
* Canh chỉnh lại vị trí các thành phần hiển thị trên form để cho ra 1 giao diện đẹp.
* Thêm các hiệu ứng nhấn phím để trả lời, click chuột để trả lời.
* Chỉnh sửa phông chữ, màu chữ, cỡ chữ...
* Thêm một số hình ảnh trang trí.

# **III. Cài đặt thực hiện:**

## 3.1 Mã nguồn: 7 cửa sổ frmBatDau.cs, frmDienthoai\_Nguoithan.cs, frmLuatchoi.cs, frmMain.cs, frmThongTin.cs, frmToTuVan.cs, frmUpdateCauHoi.cs

## 3.2 Giao diện :

• Màn hình chính (Chưa nhấn nút Bắt Đầu Chơi)



• Sau khi nhấn nút Bắt Đầu Chơi



# **IV. Tài liệu :**

- Kiến thức lập trình OOP with C# (Đã được học tại trường)

- Kiến thức chung về winform: Trang web tham khảo: Howkteam

<https://www.howkteam.vn/course/lap-trinh-winform-co-ban-27>

- Phân tích và hướng dẫn tự làm game: Ai là triệu phú

https://cinausp.org/huong-dan-lam-game-ai-la-trieu-phu-c

# **V. Phân công:**

## 1. Ý tưởng về thiết kế giao diện, chức năng:

Cả nhóm

## 2. Giải quyết bài toán:

## 3. Tiến hành code mã nguồn:

## 4. Tiến hành kiểm thử: Cả nhóm

## 5. Viết báo cáo và tổng hợp: