

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ**

ĐỀ TÀI: PHÁT TRIỂN GAME GHÉP HÌNH THEO MẪU

**Môn học: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

SVTH MSSV

1. NGUYỄN MINH SƠN 20110713
2. NGUYỄN ĐỨC THÀNH 20110307

Ngành: CÔNG NGHỆ THÔNG TIN

GVHD: TS. LÊ VĨNH THỊNH

TP Hồ Chí Minh, tháng 12 năm 2021

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

---\*\*\*---

# **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên sinh viên: NGUYỄN MINH SƠN – MSSV: 20110713

Họ và tên sinh viên: NGUYỄN ĐỨC THÀNH – MSSV: 20110307

Ngành: Công nghệ thông tin

Tên đề tài: Phát triển game ghép hình theo mẫu

Họ và tên Giáo viên hướng dẫn: TS. Lê Vĩnh Thịnh

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm: (Bằng chữ……………………………………… )

Tp. *Hồ Chí Minh*, ngày tháng năm 20…

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

Nhóm em xin được phép gửi lời cảm ơn chân thành tới thầy Lê Vĩnh Thịnh, người đã giảng dạy và hướng dẫn tận tình cho chúng em trong suốt quá trình học tập môn học Lập trình hướng đối tượng. Nhờ việc giảng dạy và hướng dẫn nhiệt tình của thầy mà chúng em đã nắm chắc kiến thức cốt lõi của môn học, cũng như hiểu rõ hơn cách thức trình bày cũng như thực hiện đồ án. Vì thế mà nhóm em đã hoàn thành đúng tiến độ thầy giao và học hỏi được nhiều kinh nghiệm trong suốt một học kì.

Với đồ án này, nhóm em đã thực hiện trong vòng 3 tháng, tuy nhiên sai sót là điều không thể tránh khỏi, bởi có nhiều kiến thức mới mà chúng em chưa được tiếp cận kịp thời trong khoảng thời gian đó. Nhóm em mong nhận được ý kiến đóng góp của thầy để giúp hoàn thiện hơn cho đề tài cũng như có được nền tảng kiến thức vững chắc nhất.

Chúng em xin chân thành cảm ơn.

# **TÓM TẮT**

Trò chơi xếp hình là một trong những trò chơi nổi tiếng rộng rãi đối với tuổi thơ mỗi người. Để vận dung được kiến thức về Lập trình hướng đối tượng được học, nhóm đã phát triển trò chơi xếp hình theo mẫu, đồng thời thêm một số chức năng giúp cải thiện trò chơi. Giao diện của trò chơi được xây dựng trên nền tảng WinForm .NET framework, một nền tảng tuy cũ nhưng vẫn có thể đáp ứng đầy đủ các yêu cầu cơ bản. Đồng thời, nhóm cũng tìm hiểu và vận dụng được lập trình Socket trên môi trường .NET framework để có thể phát triển chức năng chơi hai người qua mạng LAN.

Trò chơi do nhóm phát triển cung cấp đủ các chức năng cần có của một game xếp hình như: kéo thả ảnh, trộn hình ảnh, chọn độ khó, chọn ảnh khác,… Đồng thời, nhóm cũng đã thêm một số chức năng như nghe nhạc, hẹn giờ tắt/khởi động lại máy, đặc biệt là chế độ chơi hai người qua mạng LAN nhằm tăng trải nghiệm người dùng khi chơi trò chơi. Mặc dù trong quá trình thực hiện đề tài, nhóm đã gặp không ít khó khăn, vì thế không tránh khỏi sai sót và các lỗi phát sinh trong quá trình chơi mà nhóm chưa phát hiện được, tuy là vậy nhưng nhóm cơ bản đã hoàn thành xong mục tiêu đặt ra ban đầu.

**MỤC LỤC**

[PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN i](#_Toc90317479)

[LỜI CẢM ƠN ii](#_Toc90317480)

[TÓM TẮT iii](#_Toc90317481)

[MỤC LỤC iv](#_Toc90317482)

[DANH SÁCH CÁC BẢNG BIỂU vi](#_Toc90317483)

[DANH SÁCH CÁC HÌNH ẢNH vii](#_Toc90317484)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc90317485)

[**1.1.** **Lý do chọn đề tài** 1](#_Toc90317486)

[**1.2.** **Mục tiêu của đề tài** 1](#_Toc90317487)

[**1.3.** **Công cụ thực hiện đề tài** 1](#_Toc90317488)

[**1.4.** **Phạm vi nghiên cứu** 2](#_Toc90317489)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc90317490)

[**2.1.** **Giới thiệu tổng quan về lập trình hướng đối tượng** 3](#_Toc90317491)

[**2.2.** **Giới thiệu về Winform .NET** 4](#_Toc90317492)

[**2.3.** **Giới thiệu về Socket Programming trong .NET** 5](#_Toc90317493)

[CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG 6](#_Toc90317494)

[**3.1.** **Mô tả yêu cầu đồ án** 6](#_Toc90317495)

[**3.2.** **Lược đồ Usecase** 6](#_Toc90317496)

[**3.3.** **Thiết kế giao diện** 12](#_Toc90317497)

[**3.3.1.** **Danh sách các Form** 12](#_Toc90317498)

[**3.3.2.** **Chi tiết các Form** 12](#_Toc90317499)

[**3.4.** **Thiết kế class** 24](#_Toc90317500)

[**3.4.1.** **Form chọn chế độ chơi** 24](#_Toc90317501)

[**3.4.2.** **Các class xây dựng khu vực chơi** 24](#_Toc90317502)

[**3.4.3.** **Các class Event (kế thừa từ class EventArgs)** 28](#_Toc90317503)

[**3.4.4.** **Các class chức năng chính** 29](#_Toc90317504)

[**3.4.5.** **Chế độ chơi đơn** 42](#_Toc90317505)

[**3.4.6.** **Chế độ chơi qua LAN** 46](#_Toc90317506)

[**3.5.** **Sơ đồ tuần tự các thao tác** 58](#_Toc90317507)

[**3.5.1.** **Thao tác thiết lập bàn chơi đơn** 58](#_Toc90317508)

[**3.5.2.** **Thao tác chức năng trộn hình ảnh** 58](#_Toc90317509)

[**3.5.3.** **Thao tác chức năng chọn độ khó** 59](#_Toc90317510)

[**3.5.4.** **Thao tác chức năng chọn ảnh** 59](#_Toc90317511)

[**3.5.5.** **Thao tác chức năng hẹn giờ tắt/khởi động lại máy** 60](#_Toc90317512)

[**3.5.6.** **Thao tác chức năng nghe nhạc** 60](#_Toc90317513)

[**3.5.7.** **Thao tác chức năng kết nối Host/Client** 61](#_Toc90317514)

[**3.5.8.** **Thao tác thiết lập chế độ chơi qua LAN** 61](#_Toc90317515)

[CHƯƠNG 4: KẾT LUẬN 63](#_Toc90317516)

# **DANH SÁCH CÁC BẢNG BIỂU**

[Bảng 3.1 Mô tả thành phần Form mở đầu 13](#_Toc90317379)

[Bảng 3.2 Mô tả thành phần Form chơi đơn 14](#_Toc90317380)

[Bảng 3.3 Mô tả các thành phần Form chọn độ khó 16](#_Toc90317381)

[Bảng 3.4 Mô tả thành phần Form chọn hình ảnh 17](#_Toc90317382)

[Bảng 3.5 Mô tả thành phần Form tuỳ chọn tắt/khởi động lại máy 18](#_Toc90317383)

[Bảng 3.6 Mô tả thành phần Form chọn nhạc 19](#_Toc90317384)

[Bảng 3.7 Mô tả thành phần Form nhập IP/Port 20](#_Toc90317385)

[Bảng 3.8 Mô tả các thành phần Form chơi qua LAN phía Host 21](#_Toc90317386)

[Bảng 3.9 Mô tả các thành phần Form chơi qua LAN phía Client 23](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317387)

[Bảng 3.10Mô tả các thuộc tính của class InitialGameForm 24](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317388)

[Bảng 3.11Mô tả các thuộc tính class Board 25](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317389)

[Bảng 3.12Mô tả các phương thức class Board 25](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317390)

[Bảng 3.13Mô tả các thuộc tính class BlankBoard 26](#_Toc90317391)

[Bảng 3.14Mô tả các phương thức class BlankBoard 26](#_Toc90317392)

[Bảng 3.15Mô tả các thuộc tính class ImageBoard 26](#_Toc90317393)

[Bảng 3.16Mô tả các phương thức class ImageBoard 26](#_Toc90317394)

[Bảng 3.17Mô tả các thuộc tính của class BoardManager 27](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317395)

[Bảng 3.18Mô tả các thuộc tính của event CloseChooseImageForm 28](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317396)

[Bảng 3.19Mô tả các thuộc tính của event CloseChooseLevelForm 28](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317397)

[Bảng 3.20Mô tả các thuộc tính của event FilledImageInBlankBoard 29](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317398)

[Bảng 3.21 Mô tả các thuộc tính của event PressPlayButtonMusicPlayer 29](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317399)

[Bảng 3.22Mô tả các thuộc tính của event ClosePowerModifierForm 29](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317400)

[Bảng 3.23Mô tả các thuộc tính của class ChooseImageForm 30](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317401)

[Bảng 3.24Mô tả các thuộc tính của class ImageFileManager 31](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317402)

[Bảng 3.25Mô tả các thuộc tính của class LevelManager 33](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317403)

[Bảng 3.26Mô tả các thuộc tính của class ChooseLevelForm 34](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317404)

[Bảng 3.27Mô tả các thuộc tính của class Song 35](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317405)

[Bảng 3.28Mô tả các thuộc tính của class MusicPlayerManager 35](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317406)

[Bảng 3.29Mô tả các thuộc tính của class MusicPlayerForm 37](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317407)

[Bảng 3.30Mô tả các sự kiện của class MusicPlayerForm 37](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317408)

[Bảng 3.31Mô tả các thuộc tính của class Power 38](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317409)

[Bảng 3.32Mô tả các thuộc tính của class PowerModifierForm 39](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317410)

[Bảng 3.33Mô tả các sự kiện của class PowerModifierForm 39](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317411)

[Bảng 3.34 Mô tả các thuộc tính của class GameFunction 40](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317412)

[Bảng 3.35Mô tả các sự kiện của class GameFunction 41](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317413)

[Bảng 3.36Mô tả các thuộc tính của class PuzzleGameForm 43](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317414)

[Bảng 3.37Mô tả các thuộc tính của class SocketManager 47](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317415)

[Bảng 3.38Mô tả các event của class SocketManager 47](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317416)

[Bảng 3.39Mô tả các thuộc tính của class SocketForm 48](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317417)

[Bảng 3.40Mô tả các thuộc tính của class MultiplayableData 50](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317418)

[Bảng 3.41 Mô tả các thuộc tính của class MultiplayManager 53](file:///E:\Learning\OOP\Report\GhepHinh.docx#_Toc90317419)

[Bảng 3.42 Mô tả các thuộc tính của class MultiplayForm 56](#_Toc90317420)

# **DANH SÁCH CÁC HÌNH ẢNH**

[Hình 3.1 Các thao tác trên Form PuzzleGameForm 6](#_Toc90317421)

[Hình 3.2 Các thao tác trên Form ChooseLevelForm 7](#_Toc90317422)

[Hình 3.3 Các thao tác trên Form ChooseImageForm 7](#_Toc90317423)

[Hình 3.4 Các thao tác trên Form PowerModifierForm 8](#_Toc90317424)

[Hình 3.5 Các thao tác trên Form MusicPlayerForm 9](#_Toc90317425)

[Hình 3.6 Các thao tác trên Form SocketForm 10](#_Toc90317426)

[Hình 3.7 Các thao tác phía bên Host trên Form MultiplayForm 11](#_Toc90317427)

[Hình 3.8 Các thao tác phía bên Client trên Form MultiplayForm 12](#_Toc90317428)

[Hình 3.9Giao diện Form mở đầu 13](#_Toc90317429)

[Hình 3.10Giao diện Form chơi đơn 14](#_Toc90317430)

[Hình 3.11Giao diện Form chọn độ khó 16](#_Toc90317431)

[Hình 3.12Giao diện Form chọn hình ảnh 17](#_Toc90317432)

[Hình 3.13Giao diện Form tuỳ chọn tắt/ khởi động lại máy 18](#_Toc90317433)

[Hình 3.14Giao diện Form chọn nhạc 19](#_Toc90317434)

[Hình 3.15Giao diện Form nhập IP/Port 20](#_Toc90317435)

[Hình 3.16Giao diện Form chơi qua LAN phía Host 21](#_Toc90317436)

[Hình 3.17Giao diện Form chơi qua LAN phía Client 22](#_Toc90317437)

[Hình 3.18Mô tả class InitialGameForm 24](#_Toc90317438)

[Hình 3.19Mô tả quan hệ giữa các class BlankBoard, Board, ImageBoard 25](#_Toc90317439)

[Hình 3.20Mô tả class BoardManager 27](#_Toc90317440)

[Hình 3.21Mô tả class event CloseChooseImageForm 28](#_Toc90317441)

[Hình 3.22Mô tả class event CloseChooseLevelForm 28](#_Toc90317442)

[Hình 3.23Mô tả class event FilledImageInBlankBoard 29](#_Toc90317443)

[Hình 3.24Mô tả class event PressPlayButtonMusicPlayer 29](#_Toc90317444)

[Hình 3.25Mô tả class event ClosePowerModifierForm 29](#_Toc90317445)

[Hình 3.26Mô tả class ChooseImageForm 30](#_Toc90317446)

[Hình 3.27Mô tả class ImageFileManager 31](#_Toc90317447)

[Hình 3.28Mô tả class LevelManager 33](#_Toc90317448)

[Hình 3.29Mô tả class ChooseLevelForm 34](#_Toc90317449)

[Hình 3.30Mô tả class Song 35](#_Toc90317450)

[Hình 3.31Mô tả class MusicPlayerManager 35](#_Toc90317451)

[Hình 3.32Mô tả class MusicPlayerForm 37](#_Toc90317452)

[Hình 3.33Mô tả class Power 38](#_Toc90317453)

[Hình 3.34Mô tả class PowerModifierForm 39](#_Toc90317454)

[Hình 3.35Mô tả class GameFunction 41](#_Toc90317455)

[Hình 3.36Mô tả Interface ICropImage 42](#_Toc90317456)

[Hình 3.37 Mô tả class PuzzleGameForm 43](#_Toc90317457)

[Hình 3.38 Mô tả class SocketManager 47](#_Toc90317458)

[Hình 3.39 Mô tả class SocketForm 48](#_Toc90317459)

[Hình 3.40Mô tả các thuộc tính 50](#_Toc90317460)

[Hình 3.41Mô tả constructor class MultiplayableData 51](#_Toc90317461)

[Hình 3.42Mô tả constructor class MultiplayableData 51](#_Toc90317462)

[Hình 3.43Mô tả constructor class MultiplayableData 51](#_Toc90317463)

[Hình 3.44Mô tả constructor class MultiplayableData 51](#_Toc90317464)

[Hình 3.45Mô tả constructor class MultiplayableData 52](#_Toc90317465)

[Hình 3.46 Mô tả constructor class MultiplayableData 52](#_Toc90317466)

[Hình 3.47Mô tả constructor class MultiplayableData 52](#_Toc90317467)

[Hình 3.48Mô tả constructor class MultiplayableData 53](#_Toc90317468)

[Hình 3.49Mô tả class MultiplayManager 53](#_Toc90317469)

[Hình 3.50Mô tả class MultiplayForm 55](#_Toc90317470)

[Hình 3.51Thao tác thiết lập bàn chơi đơn 58](#_Toc90317471)

[Hình 3.52Thao tác chức năng trộn ảnh 59](#_Toc90317472)

[Hình 3.53Thao tác chức năng chọn độ khó 59](#_Toc90317473)

[Hình 3.54Thao tác chức năng chọn hình ảnh 60](#_Toc90317474)

[Hình 3.55Thao tác chức năng tuỳ chọn tắt/khởi động lại máy 60](#_Toc90317475)

[Hình 3.56Thao tác chức năng nghe nhạc 61](#_Toc90317476)

[Hình 3.57Thao tác chức năng chọn Host/Client 61](#_Toc90317477)

[Hình 3.58Thao tác chức năng chơi LAN 62](#_Toc90317478)

# **CHƯƠNG 1: GIỚI THIỆU**

* 1. **Lý do chọn đề tài**

Môn lập trình hướng đối tượng (OOP) được xem như môn nền tảng trang bị cho sinh viên các kiến thức cơ bản về lập trình dựa trên các khái niệm lớp và đối tượng. Chính vì vậy nên ưu điểm của OOP so với phương thức lập trình cũ (lập trình hướng thủ tục) là rất lớn. Trải qua một học kì với sự hướng dẫn của thầy, chúng em đã tích lũy được nhiều kiến thức cũng như kinh nghiệm với lập trình hướng đối tượng thông qua công cụ là ngôn ngữ lập trình C#. Để cụ thể hóa những kiến thức đó vào các sản phẩm thực tế, nhóm chúng em đã lựa chọn đề tài “Game xếp hình” để làm đồ án cuối kì môn OOP. Chúng em nhận thấy đây là một trò chơi rất phổ biến và chắc hẳn ai cũng đã từng chơi qua. Đồng thời project cũng đáp ứng đủ các yêu cầu cần thiết đối với môn lập trình hướng đối tượng OOP.

Do mới tiếp cận với ngôn ngữ C# và việc xử lí các kĩ thuật trong lập trình hướng đối tượng còn hạn chế nên có thể project chưa được tối ưu cũng như vẫn còn tồn tại nhiều lỗi. Nhóm chúng em rất mong nhận được sự nhận xét, góp ý từ thầy về sản phẩm này để có thể khắc phục các hạn chế và tiếp tục làm tốt hơn trong các đồ án sau. Chúng em xin cảm ơn thầy !

* 1. **Mục tiêu của đề tài**
* Xây dựng các lớp và đối tượng thích hợp cho game xếp hình.
* Tìm hiểu và thực thi code trên nền tảng Winform .Net C# để xây dựng giao diện trò chơi.
* Tìm hiểu và triển khai thêm các chức năng nghe nhạc, hẹn giờ.
* Tìm hiểu về lập trình Socket trên C# để hoàn thiện chức năng chơi qua LAN.
* Vận dụng đầy đủ kiến thức về lập trình hướng đối tượng đã được học.
  1. **Công cụ thực hiện đề tài**
* Một trong những công cụ tuy cũ nhưng vẫn rất tốt cho việc học để lấy nền tảng cho sau này là WinForm .NET, được phát triển bởi Microsoft, viết bằng ngôn ngữ C# nên có nhiều tài liệu trên mạng để nhóm tìm hiểu và tham khảo.
* WinForm được dùng để xây dựng giao diện cơ bản.
  1. **Phạm vi nghiên cứu**
* Winform .NET framework
  + Xây dựng giao diện
  + Tạo event tương tác với các đối tượng
* Socket Programming
  + Tạo và thiết lập kết nối TCP giữa 2 máy qua mạng LAN
  + Truyền/nhận dữ liệu giữa 2 máy trong mạng LAN

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

* 1. **Giới thiệu tổng quan về lập trình hướng đối tượng**

OOP là tên viết tắt của **Object-oriented programming** – lập trình hướng đối tượng. Đây là một mẫu hình lập trình được phát triển dựa trên khái niệm về đối tượng. Trong đó, đối tượng chứa đựng các dữ liệu trên các trường thường được gọi là thuộc tính. Mã nguồn được tổ chức thành các phương thức giúp cho đối tượng có thể truy xuất, hiệu chỉnh các trường dữ liệu của đối tượng khác mà đối tượng hiện tại có tương tác.

Hiểu một cách đơn giản, lập trình hướng đối tượng có tác dụng tăng năng suất và đẩy nhanh tiến độ công việc. Từ đó cho phép các lập trình viên tác động vào các chủ thể như trong thực tế. Qua đó để sửa lỗi phần mềm, chủ động bảo trì, duy trì hệ thống. Ngôn ngữ lập trình hướng đối tượng rất đa dạng, phần lớn là các ngôn ngữ lập trình theo lớp. Mỗi đối tượng có khả năng nhận thông báo, xử lý dữ liệu và gửi trả lời đến các đối tượng khác hay đến môi trường.

Các đặc điểm của lập trình hướng đối tượng:

* **Encapsulation**: tính đóng gói

Tính đóng gói cho phép che giấu thông tin và những tính chất xử lý bên trong của đối tượng. Các đối tượng khác không thể tác động trực tiếp đến dữ liệu bên trong và làm thay đổi trạng thái của đối tượng mà bắt buộc phải thông qua các phương thức công khai do đối tượng đó cung cấp.

Tính chất này giúp tăng tính bảo mật cho đối tượng và tránh tình trạng dữ liệu bị hư hỏng ngoài ý muốn.

* **Abstraction**: tính trừu tượng

Tính trừu tượng giúp loại bỏ những thứ phức tạp, không cần thiết của đối tượng và chỉ tập trung vào bản chất, những gì cốt lõi, quan trọng.

* **Inheritance**: tính kế thừa

Đây là tính chất được sử dụng khá nhiều. Tính kế thừa cho phép xây dựng một lớp mới (lớp Con), kế thừa và tái sử dụng các thuộc tính, phương thức dựa trên lớp cũ (lớp Cha) đã có trước đó.

Các lớp Con kế thừa toàn bộ thành phần của lớp Cha và không cần phải định nghĩa lại. Lớp Con có thể mở rộng các thành phần kế thừa hoặc bổ sung những thành phần mới.

* **Polymorphism**: tính đa hình

Tính đa hình trong lập trình OOP cho phép các đối tượng khác nhau thực thi chức năng giống nhau theo những cách khác nhau.

* 1. **Giới thiệu về Winform .NET**

C# là ngôn ngữ lập trình hiện đại, theo lập trình hướng đối tượng được phát triển bởi đội ngũ kĩ sư của Microsoft vào năm 2000. Nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

WinForm là một thuật ngữ mô tả một ứng dụng được viết trên nền .NET Framework và có giao diện người dùng màn hình Windows.

Ưu điểm của WinForm là giao diện dễ sử dụng, ta có thể dễ dàng gắn các event cho button chỉ cần double click, lại hỗ trợ rất nhiều event như click, hover,… Việc viết code cũng trực quan hơn từ việc lấy text từ TextBox cho tới show dữ liệu bằng MessageBox hoặc dùng GribView để kết nối tới SQL. WinForm cũng rất phù hợp để làm các phần mềm quản lí, tính tiền, thống kê,…

* 1. **Giới thiệu về Socket Programming trong .NET**

Lập trình Socket là cách thức lập trình để kết nối 2 máy tính trong cùng một mạng (LAN) giúp chúng có thể giao tiếp được với nhau. Đây là thiết lập giữa một Client và một Server, Client có thể kết nối, gửi dữ liệu tới Server trong khi Server có thể lắng nghe kết nối tại một Port nào đó của địa chỉ IP, sau đó nhận dữ liệu, hiển thị chúng thông qua kết nối socket giữa 2 máy.

Bộ giao thức TCP/IP cung cấp các dịch vụ truyền thông như: TCP, UDP…. Giao thức TCP là giao thức **hướng kết nối** và **truyền dữ liệu tin cậy**, đảm bảo cho các chương trình đích có thể nhận được đúng dữ liệu theo thứ tự đã gửi ban đầu, không mất hoặc thừa byte nào. Trong khi, giao thức UDP là giao thức **phi liên kết** và **truyền dữ liệu không tin cậy**, khi một chương trình gửi dữ liệu thì giao thức UDP không đảm bảo dữ liệu sẽ đến đích cùng lúc đúng theo thứ tự ban đầu được gửi đi.

Trong đề tài này, nhóm em tập trung vào việc tạo kết nối TCP giữa 2 thiết bị, sử dụng các phương thức trong class Socket do .NET framework cung cấp sẵn. Ngoài ra, .NET framework còn hỗ trợ các lớp khác cho lập trình socket như: giao diện luồng dữ liệu trên mạng (NetStream), trình tự hoá và giải trình dữ liệu (Serialization/Deserialization), biến đổi dữ liệu (data -conversion)…

# **CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG**

1. **Mô tả yêu cầu đồ án**

* **Mô tả trò chơi:** Trò chơi xếp hình cho phép người chơi kéo thả các phần nhỏ của một hình ảnh ở khu vực này vào vị trí ở khu vực khác sao cho hoàn thành giống với hình ảnh ban đầu.
* **Các chức năng**: Kéo thả ảnh, trộn hình ảnh, chọn ảnh từ nguồn bên ngoài, chọn độ khó, nghe nhạc, hẹn giờ tắt máy, chơi 2 người qua mạng LAN

1. **Lược đồ Usecase**

**Diagram

Description automatically generated**

Hình 3. Các thao tác trên Form PuzzleGameForm

Diagram

Description automatically generated

Hình 3. Các thao tác trên Form ChooseLevelForm

Diagram

Description automatically generated

Hình 3. Các thao tác trên Form ChooseImageForm

Diagram

Description automatically generated

Hình 3. Các thao tác trên Form PowerModifierForm

Diagram

Description automatically generated

Hình 3. Các thao tác trên Form MusicPlayerForm

Diagram

Description automatically generated

Hình 3. Các thao tác trên Form SocketForm

Diagram

Description automatically generated

Hình 3. Các thao tác phía bên Host trên Form MultiplayForm

Diagram

Description automatically generated

Hình 3. Các thao tác phía bên Client trên Form MultiplayForm

1. **Thiết kế giao diện**
   * 1. **Danh sách các Form**

* Form chọn chế độ chơi (**InitialGameForm**)
* Form chế độ chơi đơn (**PuzzleGameForm**)
* Form chọn độ khó (**ChooseLevelForm**)
* Form chọn hình ảnh (**ChooseImageForm**)
* Form hẹn giờ tắt máy (**PowerModifierForm**)
* Form chọn nhạc để phát (**MusicPlayerForm**)
* Form nhập IP Address-Port (**SocketForm**)
* Form chế độ chơi qua LAN (**MultiplayForm**)
  + 1. **Chi tiết các Form**
       1. **Form chọn chế độ chơi (InitialGameForm)**

Chart

Description automatically generated

4

5

3

2

1

Hình 3.Giao diện Form mở đầu

Bảng 3. Mô tả thành phần Form mở đầu

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Label (Logo) | Logo của trò chơi |
| 2 | Button (Single Play) | Chọn chế độ chơi đơn |
| 3 | Button (Multi Play (LAN)) | Chọn chế độ chơi 2 người qua LAN |
| 4 | Button (About) | Thông tin về đồ án |
| 5 | Button (Quit) | Thoát trò chơi |

* + - 1. **Form chế độ chơi đơn (PuzzleGameForm)**

Graphical user interface

Description automatically generated with low confidence

8.5

8.4

8.3

8.2

8.1

2Graphical user interface

Description automatically generated with low confidence

3

10

9

8

7

6

5

4

1

Hình 3.Giao diện Form chơi đơn

Bảng 3. Mô tả thành phần Form chơi đơn

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | MenuStrip (Option)   |  | | --- | | MenuStripItem (F1) | | MenuStripItem (F2) | | MenuStripItem (F3) | | MenuStripItem (F4) | | MenuStripItem (F5) | | Chọn các chức năng   |  | | --- | | Trộn hình ảnh | | Chọn ảnh khác | | Chọn độ khó | | Hẹn giờ tắt/khởi động | | Phát nhạc | |
| 2 | Picture Box | Chứa hình ảnh được kéo thả |
| 3 | Picture Box List | Chứa tập các hình ảnh khi người chơi kéo thả |
| 4 | Picture Box | Chứa hình ảnh gốc để làm gợi ý cho người chơi |
| 5 | Label | Chứa chỉ dẫn cách bắt đầu trò chơi |
| 6 | Label | Chứa thời gian đếm ngược dành cho phần chơi với độ khó tương ứng |
| 7 | Group Box   |  | | --- | | Combo Box | | Bảng lưu thời gian hoàn thành sớm nhất   |  | | --- | | Chọn độ khó tương ứng để xem thành tích | |
| 8 | Panel   |  | | --- | | Label (1) | | Picture Box (2) | | Picture Box (3) | | Picture Box (4) | | Picture Box (5) | | Chứa các trình điểu khiển phát nhạc   |  | | --- | | Tên bài hát | | Tuỳ chọn phát bài hát trước | | Tuỳ chọn phát/dừng | | Tuỳ chọn lặp lại bài hát | | Tuỳ chọn phát bài hát sau | |
| 9 | Picture Box List | Chứa tập các hình ảnh ban đầu |
| 10 | Label List | Chứa tập các thông tin khi thực hiện tuỳ chọn hẹn giờ tắt máy |

* + - 1. **Form chọn độ khó (ChooseLevelForm)**

**Graphical user interface, text, application

Description automatically generated**

1

4

3

2

Hình 3.Giao diện Form chọn độ khó

Bảng 3. Mô tả các thành phần Form chọn độ khó

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Panel | Chứa các tuỳ chọn level |
| 2 | Radio Button | Chọn level tương ứng |
| 3 | Label | Logo Level |
| 4 | Button | Xác nhận level đã chọn |

* + - 1. **Form chọn hình ảnh (ChooseImageForm)**

**Graphical user interface, text, application, Word

Description automatically generated**

2

1

5

4

3

Hình 3.Giao diện Form chọn hình ảnh

Bảng 3. Mô tả thành phần Form chọn hình ảnh

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | List View | Chứa các hình ảnh được thêm vào |
| 2 | Image List Item | Chứa 1 hình ảnh |
| 3 | Button | Mở hộp thoại chọn file để thêm ảnh vào |
| 4 | Button | Xác nhận chọn ảnh |
| 5 | Button | Xoá ảnh đã chọn khỏi List View |

* + - 1. **Form hẹn giờ tắt/khởi động lại máy (PowerModifierForm)**

A screenshot of a computer

Description automatically generated

6

7

8

5

4

1

2

3

Hình 3.Giao diện Form tuỳ chọn tắt/ khởi động lại máy

Bảng 3. Mô tả thành phần Form tuỳ chọn tắt/khởi động lại máy

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Label | Logo Power |
| 2 | Rich Text Box | Chứa thời gian đếm ngược |
| 3 | Label | Biểu diễn thời gian đếm ngược để tắt/khởi động lại máy |
| 4 | Combo Box | Lựa chọn chế độ tắt hay khởi động lại |
| 5 | Date Time Picker | Chọn thời gian để hẹn giờ |
| 6 | Button | Thông tin về Form này |
| 7 | Button | Bắt đầu đếm ngược |
| 8 | Button | Huỷ đếm ngược |

* + - 1. **Form chọn nhạc (MusicPlayerForm)**

Graphical user interface, text, application

Description automatically generated

1

3

4

5

2

6

Hình 3.Giao diện Form chọn nhạc

Bảng 3. Mô tả thành phần Form chọn nhạc

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | List Box | Chứa danh sách bài hát để chọn |
| 2 | List Box Item | Chứa 1 bài hát |
| 3 | Button | Chọn file bài hát từ máy |
| 4 | Button | Xoá bài hát được chọn ra khỏi List Box |
| 5 | Button | Phát bài hát được chọn |
| 6 | Window Media Player(COM) | Biểu diễn tiến độ bài hát, dừng/phát nhạc, chỉnh âm lượng |

* + - 1. **Form nhập IP Address-Port (SocketForm)**

Graphical user interface, application

Description automatically generated

1

2

3

4

6

5

7

Hình 3.Giao diện Form nhập IP/Port

Bảng 3. Mô tả thành phần Form nhập IP/Port

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Label | Tên tương ứng các thành phần |
| 2 | Text Box | Nhập tên người chơi |
| 3 | Button | Chọn vai trò là Client |
| 4 | Button | Chọn vai trò là Server |
| 5 | Text Box | Nhập IP Address nếu bạn là Client |
| 6 | Text Box | Nhập Port nếu bạn là Client |
| 7 | Button | Kết nối tới server nếu bạn là Client |

* + - 1. **Form chế đội chơi qua LAN (MultiplayForm)**
* **Server**

Graphical user interface, application, website

Description automatically generated

4

16

15

14

12

13

10

11

8

9

17

6

5

7

1

2

3

Hình 3.Giao diện Form chơi qua LAN phía Host

Bảng 3. Mô tả các thành phần Form chơi qua LAN phía Host

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Label | Hiển thị Port để Client có thể vào |
| 2 | Picture Box List | Chứa tập các hình ảnh khi người chơi kéo thả |
| 3 | Picture Box | Chứa hình ảnh khi người chơi kéo thả |
| 4 | Picture Box List | Chứa tập ảnh ban đầu của phía người chơi được cắt ra |
| 5 | Picture Box List | Chứa tập các hình ảnh khi đối phương kéo thả |
| 6 | Picture Box List | Chứa tập ảnh ban đầu của phía đối phương được cắt ra |
| 7 | Picture Box | Chứa hình ảnh khi đối phương kéo thả |
| 8 | Button | Dùng để chọn độ khó (chỉ có bên phía server) |
| 9 | Button | Dùng để trộn tập các hình ảnh |
| 10 | Button | Dùng để chọn hình ảnh khác |
| 11 | Button | Bắt đầu trò chơi (chỉ bên phía server) |
| 12 | Picture Box | Chứa hình ảnh ban đầu làm gợi ý |
| 13 | Label | Đếm ngược thời gian khi server nhấn Start |
| 14 | Panel | Bảng ghi điểm khi mỗi người hoàn thành |
| 15 | Label | Điểm của phía người chơi |
| 16 | Label | Điểm của phía đối phương |
| 17 | Button | Thông tin về trò chơi |

* **Client**

Graphical user interface, application, website

Description automatically generated

8

15

14

13

12

10

11

9

4

6

5

7

3

2

1

Hình 3.Giao diện Form chơi qua LAN phía Client

Bảng 3. Mô tả các thành phần Form chơi qua LAN phía Client

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Ý nghĩa** |
| 1 | Label | Hiển thị Client |
| 2 | Picture Box List | Chứa tập các hình ảnh khi người chơi kéo thả |
| 3 | Picture Box | Chứa hình ảnh khi người chơi kéo thả |
| 4 | Picture Box List | Chứa tập ảnh ban đầu được cắt ra của phía người chơi |
| 5 | Picture Box List | Chứa tập các hình ảnh khi đối phương kéo thả |
| 6 | Picture Box List | Chứa tập ảnh ban đầu được cắt ra của phía đối phương |
| 7 | Picture Box | Chứa hình ảnh khi đối phương kéo thả |
| 8 | Picture Box | Chứa hình ảnh ban đầu làm gợi ý |
| 9 | Button | Dùng để trộn tập các hình ảnh |
| 10 | Button | Dùng để chọn hình ảnh khác |
| 11 | Button | Thông tin về trò chơi |
| 12 | Label | Đếm ngược thời gian khi server nhấn Start |
| 13 | Panel | Bảng ghi điểm khi mỗi người hoàn thành |
| 14 | Label | Điểm của phía người chơi |
| 15 | Label | Điểm của phía đối phương |

* 1. **Thiết kế class**
     1. **Form chọn chế độ chơi**

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **PuzzleGameForm** GameFm | Đối tượng Form “**PuzzleGameForm**” |
| **SocketForm** socketForm | Đối tượng Form  “**SocketForm**” |

Bảng 3.Mô tả các thuộc tính của class InitialGameForm

Graphical user interface

Description automatically generated with medium confidence

Hình 3.Mô tả class InitialGameForm

**Method**

* + **aboutBtn\_Click**: gọi khi click chuột vào button “**About**”, để hiển thị thông tin về đồ án.
  + **InitialGameForm** (constructor): để khởi tạo Form.
  + **multiplayBtn\_Click**: gọi khi click chuột vào button **“Multi Play (LAN)”**, hiển thị Form **“SockerForm”** cho phép nhập IP – Port và ẩn Form **“InitialGameForm”** .
  + **quitBtn\_Click**: gọi khi click vào button **Quit**, thoát Form **“InitialGameForm”.**
  + **singleplayBtn\_Click**: gọi khi click vào Button **“Single Play”**, hiển thị Form **“PuzzleGameForm”** để vào chế độ chơi đơn và ẩn Form **“InitialGameForm”.**
    1. **Các class xây dựng khu vực chơi**
       1. **Class Board, BlankBoard, ImageBoard**

Graphical user interface, application

Description automatically generated

Hình 3.Mô tả quan hệ giữa các class BlankBoard, Board, ImageBoard

* **Class Board**: abstract class gom nhóm các thành phần có cùng bản chất là khu vực chơi để kéo thả ảnh.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Int** width | chiều rộng của 1 board |
| **Int** height | chiểu cao của 1 board |
| **Int** indexPtrb | đánh chỉ số cho các PictureBox |
| **Int** numberOfRowBoard | số lượng hàng trong 1 board |
| **Int** numberOfColBoard | số lượng cột trong 1 board |
| **Point** position | vị trí của board trên Form |

Bảng 3.Mô tả các thuộc tính class Board

|  |  |
| --- | --- |
| **Method** | **Ý nghĩa** |
| DrawBoard | abstract method để vẽ bàn chơi |

Bảng 3.Mô tả các phương thức class Board

* **Class BlankBoard**: kế thừa từ abstract class Board biểu diễn bàn chơi còn trống ban đầu.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Panel** panelBlank | panel chứa các PictureBox không có ảnh ban đầu |

Bảng 3.Mô tả các thuộc tính class BlankBoard

|  |  |
| --- | --- |
| **Method** | **Ý nghĩa** |
| DrawBoard | override từ abstract method của lớp Board để vẽ bàn chơi trống ban đầu |

Bảng 3.Mô tả các phương thức class BlankBoard

* **Class ImageBoard**: kế thừa từ abstract class Board biểu diễn bàn chơi có danh sách các ảnh đã được cắt ra từ ban đầu.

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **List<Image>** imgs | chứa danh sách các ảnh được cắt ra ban đầu |
| **Panel** panelImgList | panel chứa các Picture Box có sẵn ảnh ban đầu |

Bảng 3.Mô tả các thuộc tính class ImageBoard

|  |  |
| --- | --- |
| **Method** | **Ý nghĩa** |
| DrawBoard | **override** từ abstract method của class **Board** để vẽ bàn chơi chứa các ảnh được cắt ra ban đầu |

Bảng 3.Mô tả các phương thức class ImageBoard

* + - 1. **Class BoardManager**

Gồm các phương thức quản lý các thao tác trên bàn chơi như kéo thả, kiểm tra thắng thua, vẽ bàn chơi ….

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **BlankBoard** blankBoard | Đối tượng class **BlankBoard** |
| **Bool** droppedImg | Đánh dấu ảnh đã được thả hay chưa |
| **Image** holdImg | Giữ hình ảnh cần được swap |
| **ImageBoard** imgBoard | Đối tượng class **ImageBoard** |
| **List<Picture Box>** ptrbList | Sau khi ảnh đã thả đầy các PictureBox trong PanelBlank thì lưu lại các PictureBox đó |

Bảng 3.Mô tả các thuộc tính của class BoardManager

Hình 3.Mô tả class BoardManager

**Method**

* + **AddImageToPictureBox**: dùng để thêm ảnh vào các Picture Box ở bên imageBoard.
  + **AddPanelToForm**: dùng để thêm Panel chứa các Picture Box vào Form “PuzzleGameForm”.
  + **AddPictureBoxToPanel**: thêm lần lượt các Picture Box vào Panel và gán các event tương ứng với các thao tác kéo thả trên Picture Box.
  + **BoardManager** (constructor): khởi tạo đối tượng BoardManager.
  + **CheckFilledPanel**: kiểm tra xem các Picture Box trong PanelBlank đã được kéo thả đầy đủ hình ảnh hay chưa.
  + **CurrPicBlank\_Paint**: tô màu border.
  + **Ptrb\_MouseDown**: sự kiện khi nhấn chuột xuống để thực hiện việc kéo ảnh, bắt đầu đếm ngược thời gian chơi và thực hiện kiểm tra bằng method CheckFilledPanel() liên tục để xác định việc hoàn tất trò chơi hay chưa.
  + **Ptrb\_MouseEnter, PtrbMouseLeave, Ptrb\_Paint**: chuỗi event thực hiện việc tô đỏ border khi người chơi rê chuột vào.
  + **PtrbDropped\_DragDrop** và **PtrbDropped\_DragEnter**: chuỗi event thực hiện việc thả ảnh đang được giữ vào 1 Picture Box.
  + **Reset**: Đặt lại các thuộc tính đang có khi bắt đầu phương thức DrawBoard.
  + **StartDraw**: thực hiện việc vẽ các bàn chơi bằng cách gọi các method DrawBoard tương ứng.
    1. **Các class Event (kế thừa từ class EventArgs)**

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** path | để lưu đường dẫn file ảnh và phát event khi đóng Form “**ChooseImageForm**” |

Bảng 3.Mô tả các thuộc tính của event CloseChooseImageForm

Hình 3.Mô tả class event CloseChooseImageForm

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** level | để lưu độ khó người chơi chọn và phát event khi đóng Form “**ChooseLevelForm**” |

Bảng 3.Mô tả các thuộc tính của event CloseChooseLevelForm

Hình 3.Mô tả class event CloseChooseLevelForm

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **List<PictureBox> ptrbList** | chứa danh sách các PictureBox khi đã được kéo thả đủ hình ảnh, sau đó phát event để xác định việc hoàn tất trò chơi hay chưa? |

Bảng 3.Mô tả các thuộc tính của event FilledImageInBlankBoard

Hình 3.Mô tả class event FilledImageInBlankBoard

Graphical user interface, text, application, chat or text message

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** name | chứa tên bài hát được phát và khi đóng Form “**MusicPlayer**” sẽ gửi event để hiển thị tên bài hát đang phát trên Form “**PuzzleGameForm**” |

Bảng 3. Mô tả các thuộc tính của event PressPlayButtonMusicPlayer

Hình 3.Mô tả class event PressPlayButtonMusicPlayer

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** hour, minute, second | Chứa thời gian còn lại cho đến khi thực hiện tắt/khởi động lại máy. |
| **String** mode | chế độ mà người dùng chọn |
| **bool** isDisplay | kiểm tra xem các thông tin trên có được phép hiển thị ra Form “**PuzzleGameForm**” hay không? |

Bảng 3.Mô tả các thuộc tính của event ClosePowerModifierForm

Hình 3.Mô tả class event ClosePowerModifierForm

* + 1. **Các class chức năng chính**
       1. **Chọn ảnh**
* **Class “ChooseImageForm”**

**Graphical user interface, text, application

Description automatically generated**

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **ImageFileManager** imageFileManager | Tạo đối tượng class **ImageFileManager** để quản lý việc thêm, xoá, chọn các hình ảnh từ ListView |
| **Int** imgIndexChoosen | Lưu chỉ số của ảnh khi chọn từ ListView |
| **ImageList** imageList | Đối tượng **ImageList** để lưu danh sách các hình ảnh |
| **OpenFileDialog** openFileDlg | Đối tượng của class **OpenFileDialog** để mở hộp thoại chọn file |
| **ListView** imageLsView | Đối tượng của class **ListView** để lưu và hiện thị các hình ảnh đang có trong imageList |

Bảng 3.Mô tả các thuộc tính của class ChooseImageForm

Hình 3.Mô tả class ChooseImageForm

**Method**

* + **addImgFileBtn\_Click**: sự kiện click Button “**Add Img File**” để mở hộp thoại “**openFileDlg**” chọn ảnh và sau đó thêm ảnh được chọn vào **ListView** trên Form.
  + **ChooseImageForm\_Load**: sự kiện khi Form được khởi động lần đầu, thiết lập các thuộc tính cần thiết và cập nhật các hình ảnh đã được người chơi thêm vào qua các lần chơi trước.
  + **LoadImage**: đọc các file ảnh từ thư mục chứa ảnh được lưu từ các lần chơi trước và cập nhật các hình ảnh đó lên **ListView.**
  + **DelBtn\_Click**: sự kiện khi nhấn Button “**Delete**” để xoá một ảnh khỏi **ListView** cũng như xoá ảnh đó trong thư mục lưu trữ và cập nhật lại **ListView**.
  + **OkBtn\_Click**: sự kiện khi nhấn Button “**OK**” để chọn một ảnh và hiển thị ngoài khu vực chơi bằng cách phát event để thông báo cho “**PuzzleGameForm**” .
  + **imageLsview\_SelectedIndexChanged**: Sự kiện khi chọn một ảnh trong **ListView** và cập nhật index của ảnh đó vào field **imgIndexChoosen** đồng thời enable và gán event cho 2 Button “**OK**” và “**Delete**”.
  + **imageLsview\_DragDrop** và **imageLsview\_DragEnter**: các event dùng để bắt sự kiện khi người chơi kéo thả file ảnh từ ngoài vào trong **ListView**, sau đó cập nhật ảnh đó vào ListView và lưu lại vào thư mục.
* **Class “ImageFileManager” :** gồm các thuộc tính và phương thức để quản lý việc thêm, xoá hình ảnh từ **ListView** và các thao tác làm việc với file, thư mục.

**Graphical user interface, application

Description automatically generated**

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Int** countImage | Số thứ tự để đặt lại tên các ảnh và lưu vào trong thư mục |
| **Int** indexImage | Đánh chỉ số cho từng ảnh trong ListView |
| **List<string>** listPath | Chứa danh sách các path của file ảnh |

Bảng 3.Mô tả các thuộc tính của class ImageFileManager

Hình 3.Mô tả class ImageFileManager

**Method:**

* + **ImageFileManager** (constructor): khởi tạo đối tượng .
  + **AddImageToFolderAndGetPath**: Đặt lại tên ảnh của người dùng chọn thêm vào, lưu file ảnh đó vào thư mục “**Image**” cố định trong Project đồng thời lấy đường dẫn mới của ảnh đó.
  + **AddImageToListView**: Đọc đường dẫn file ảnh mà người dùng chọn thêm vào để hiển thị ra **ListView** đồng thời cũng lưu vào thư mục “**Image**” cố đinh trong Project.
  + **GetImageFromFile**: Lấy ảnh từ đường dẫn người dùng chọn và sửa lại kích thước cho phù hợp, thông báo lỗi nếu đó không phải file ảnh.
  + **GetListPathImageFromFolder**: lấy danh sách các đường dẫn các file ảnh trong folder “**Image**” đã lưu các ảnh ở lần chơi trước đó.
  + **GetNumberImageCurrentFromFolder**: Lấy file ảnh có tên là số lớn nhất rồi trả về giá trị đó để việc đặt tên tiếp theo không bị trùng.
  + **InitialImageList**: Nhận listPath từ method **GetListPathImageFromFolder** và hiển thị ảnh lên **ListView** trong lần đầu tiên khởi động Form.
    - 1. **Chọn độ khó**
* **Class “LevelManager”**

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Button** btnOkForm2 | Button “**OK**” của Form “**ChooseLevelForm**” |
| **String** lvChoose | Nhận giá trị là độ khó hiện tại người chơi chọn |
| **Int** lvChoosing | Nhận giá trị độ khó hiện tại đang chơi để đánh dấu cho **RadioButton** tương ứng |
| **Panel** pnlForm2 | Panel chứa các đối tượng **RadioButton** biểu thị độ khó 3->10 |

Bảng 3.Mô tả các thuộc tính của class LevelManager

Hình 3.Mô tả class LevelManager

**Method**

* + **LevelManager**(constructor): khởi tạo đối tượng với thuộc tính tương ứng.
  + **Item\_CheckedChanged:** bắt sự kiện khi chọn 1 **RadioButton** để hiển thị Button “**OK**”.
  + **LevelChoosen**: Kiểm tra xem đối tượng **RadioButton** có đang được chọn hay không, nếu có sẽ lấy độ khó tương ứng.
  + **SetlevelChoosing**: Để set độ khó đang chơi vào **RadioButton** tương ứng khi khởi động Form “**ChooseLevelForm**”.
* **Class “ChooseLevelForm”**

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **LevelManager** lvManager | Đối tượng class LevelManager để quản lí việc chọn level |

Bảng 3.Mô tả các thuộc tính của class ChooseLevelForm

Hình 3.Mô tả class ChooseLevelForm

**Method**

* + **ChooseLevelForm** (constructor): khởi tạo đối tượng của Form “**ChooseLevelForm**”, và cập nhật **RadioButton** chứa độ khó đang chơi.
  + **btnOKForm2\_Click**: sự kiện click Button “**OK**” trên Form để xác nhận chọn độ khó và phát thông báo cho Form “**PuzzleGameForm**”.
    - 1. **Phát nhạc**
* **Class “Song”:** chứa thông tin một bài hát

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** name | Lưu tên bài hát |
| **String** path | Lưu đường dẫn hiện tại của bài hát |

Bảng 3.Mô tả các thuộc tính của class Song

Hình 3.Mô tả class Song

* **Class “MusicPlayerManager”:** chứa các phương thức xử lý việc thêm, xoá, phát một bài hát

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String** folderPath | Đường dẫn tới folder “**Music**” cố định trong Project |
| **Int** indexSongPlaying | Chứa chỉ mục của bài hát đang phát |
| **String** songPlaying | Chứa đường dẫn của bài hát đang hát |
| **String** songChoosen | Chứa đường dẫn của bài hát được chọn trong ListBox |
| **List<Song>** songs | Chứa danh sách các bài hát hiện tại |

Bảng 3.Mô tả các thuộc tính của class MusicPlayerManager

Hình 3.Mô tả class MusicPlayerManager

**Method**

* + **MusicPlayerManager** (constructor): khởi tạo đối tượng.
  + **AddSongToListBoxAndSave**: Thêm bài hát được chọn từ hộp thoại “**OpenFileDialog**” vào **ListBox** và lưu vào folder “**Music**”.
  + **GetSongFromFolder**: lấy toàn bộ đường dẫn của các bài hát trong folder “**Music**” để cập nhật vào **List<Song>** songs.
  + **LoadMusic**: Cập nhật các bài hát đang có trong folder “**Music**” lên **ListBox** trong lần đầu khởi động Form.
  + **SaveIntoFolder**: Lưu các bài hát hiện tại vào folder “**Music**”
  + **UpdateListBox**: cập nhật danh sách bài hát vào **ListBox** mỗi khi có thao tác thêm, xoá.
* **Class “MusicPlayerForm”**

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **MusicPlayerManager**  musicPlayerManager | Khởi tạo đối tượng để thực hiện các thao táo quản lý thêm, xoá, phát bài hát |

Bảng 3.Mô tả các thuộc tính của class MusicPlayerForm

|  |  |
| --- | --- |
| **Event** | **Ý nghĩa** |
| **OnEndedMusic** | Sự kiện khi bài hát kết thúc và phát ra thông báo |
| **OnPressPlay**  **ButtonMusicPlayer** | Sự kiện khi bắt đầu phát bài hát để thông báo cho “**PuzzleGameForm**” để hiển thị trình điều khiển nhạc. |

Bảng 3.Mô tả các sự kiện của class MusicPlayerForm

Hình 3.Mô tả class MusicPlayerForm

**Method**

* + **DelBtn\_Click**: Bắt sự kiện khi nhấn Button “**Delete**” để xoá một bài hát khỏi **ListBox** và folder “**Music**”.
  + **MusicPlayerForm\_Load**: sự kiện xảy ra khi Form được gọi lần đầu để load file nhạc từ folder “**Music**” ở các lần chơi truóc lên **ListBox.**
  + **openBtn\_Click**: bắt sự kiên khi nhấn Button “**Open**” để chọn file nhạc từ hộp thoại “**OpenFileDialog**”.
  + **PlayBtn\_Click**: bắt sự kiện nhấn Button “**Play**” để phát bài hát hiện tại đang chọn .
  + **songsLsbx\_SelectedIndexChanged:** bắt sự kiện khi chọn một bài hát trong **ListBox.**
  + **WMPlayer\_PlayStateChange:** bắt sự kiện khi trạng thái của bài hát thay đổi trong **Window Media Player** để xử lý các tình huống như lặp lại bài hát, phát bài tiếp theo…
    - 1. **Hẹn giờ tắt/khởi động lại**
* **Class “Power”:** Thực hiện việc tắt/khởi động lại máy

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **String processName** | Lưu tên tiến trình mặc định của Window “shutdown.exe” thực hiện việc shutdown/restart |
| **String typeName** | Tuỳ chọn shutdown/restart |
| **ProcessStartInfo process** | Đối tượng của ProcessStartInfo |

Bảng 3.Mô tả các thuộc tính của class Power

Hình 3.Mô tả class Power

**Method**

* + **GetPowerType**: Chọn các tuỳ chọn thích hợp.
  + **PowerManager**: Thực hiện tuỳ chọn đã chọn trước đó.
* **Class “PowerModifierForm”**

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **GroupBox** powerGrpBox | Đối tượng class **GroupBox** nằm trên Form “**PuzzleGameForm”** thực hiện việc hiển thị thời gian còn lại cho đến khi tuỳ chọn được thực hiện. |

Bảng 3.Mô tả các thuộc tính của class PowerModifierForm

|  |  |
| --- | --- |
| **Event** | **Ý nghĩa** |
| **HandleTime**  **WhenClosing** | Hiển thị thời gian còn lại trên powerGrpBox khi đóng **“PowerModifierForm”** |
| **HandleTime**  **WhenPressCancel** | Huỷ thời gian đếm ngược trên cả **powerGrpBox** và “**PowerModifierForm**” khi nhất Button **Cancel** |
| **HandleTime**  **WhenPressStart** | Bắt đầu đếm thời gian trên cả **powerGrpBox** và **“PowerModifierForm”** khi nhấn Button **Start** |

Bảng 3.Mô tả các sự kiện của class PowerModifierForm

Hình 3.Mô tả class PowerModifierForm

**Method**

* + **cancelBtn\_Click**: bắt sự kiện nhấn Button “**Cancel**” để huỷ bộ đếm thời gian.
  + **GameFunction\_TimeEndEvent**: bắt sự kiện khi bộ đếm hết thời gian sẽ thực hiện các tuỳ chọn tương ứng.
  + **PowerModifier\_FormClosing**: sự kiện trước khi đóng Form sẽ phát thông báo có hiển thị thời gian còn lại lên “**PuzzleGameForm**” hay không?
  + **PowerModifier\_Load:** sự kiện khi khởi động Form, để cập nhật các thuộc tính và thời gian còn lại, chế độ đã chọn trên “**PowerModifierForm**” trước đó.
  + **startBtn\_Click**: bắt sự kiện khi nhấn Button “**Start**” và bộ đếm thời gian bắt đầu chạy.
  + **timePicker\_ValueChanged**: Sự kiện khi chọn thời gian trên TimePicker, sau đó hiển thị thời gian còn lại trên. **“PowerModifierForm”.**
    - 1. **Quản lý các chức năng**

Bảng 3. Mô tả các thuộc tính của class GameFunction

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **ChooseImgForm** chooseImgForm | Đối tượng Form “**ChooseImgForm**” |
| **ChooseLvForm**  chooseLvForm | Đối tượng Form “**ChooseLvForm**” |
| **MusicPlayerForm**  musicPlayerForm | Đối tượng Form “**MusicPlayerForm**” |
| **PowerModifierForm**  powerFm | Đối tượng Form “**PowerModifierForm**” |
| Bool isPowerStartClicked | Kiểm tra xem Button **Start** của “**PowerModifierForm**” được nhấn hay chưa? |
| **String** modePowerChosen | Chế độ Power đang được chọn |
| **List<Thread>** threadCountDown | Danh sách các Thread của bộ đếm thời gian trên Form ”**PuzzleGameForm**” |
| **List<Thread>** threadsOfPowerForm | Danh sách các Thread của bộ đếm thời gian trên Form ”**PowerModifierForm**” |
| **List<string>** timeList | Lưu thời gian đếm ngược trên Form “**PuzzleGameForm**” |
| **DateTime** timeSet | Lưu mốc thời gian đã chọn trên TimePicker của Form “**PowerModifierForm**” |

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Event** | **Ý nghĩa** |
| **TimeEndEvent** | Sự kiện được phát khi bộ đếm thời gian về 0 |

Bảng 3.Mô tả các sự kiện của class GameFunction

Hình 3.Mô tả class GameFunction

**Method**

* + **ChooseImage**: khởi chạy Form “**ChooseImageForm**”.
  + **ChooseLevel**: khởi chạy Form “**ChooseLevelForm**”.
  + **MusicPlayer**: khởi chạy Form “**MusicPlayerForm**”.
  + **PowerModify**: khởi chạy Form “**PowerModifierForm**”.
  + **ConvertTimeToSecond** và **ConvertSecondToTime**: để chuyển đổi qua lại giữa chuỗi thời gian và giây.
  + **HandleTime**: chuyển số nguyên 1 chữ số sang dạng giờ, phút, giây có 2 chữ số.
  + **Timer**: tìm thời gian trước đó 1 giây.
  + **TimeElapse**: tính thời gian đã trôi qua kể từ mốc thời gian cố định.
  + **TimeCount**: thực hiện việc đếm ngược thời gian liên tục trên một thread khác và lưu lại các Thread này.
  + **ImagesAreDifferent**: kiểm tra 2 ảnh có giống nhau hoàn toàn hay không.
  + **IsWin**: Kiểm tra hoàn thành trò chơi hay chưa.
  + **StopThreads**: ngừng các thread đang chạy, chủ yếu là bộ đếm thời gian khi có tác động nào đó.
  + **RandomNotRepeat**: random số ngẫu nhiên không tùng lặp để cho việc trộn các ảnh.
  + **Mix**: thực hiện việc trộn ảnh.
    - 1. **Interface IcropImage**

Cung cấp cho class khả năng cắt một ảnh thành nhiều ảnh nhỏ với các kích thước chỉ định.

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 3.Mô tả Interface ICropImage

* + 1. **Chế độ chơi đơn**

Class **“PuzzleGameForm”:** Form chính chứa các đối tượng thao tác khi chơi, triển khai Interface **IcropImage.**

Graphical user interface

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Int** maxLevel | Độ khó tối đa trong game |
| **String** path | Đường dẫn file ảnh khi thực hiện đổi ảnh |
| **Image** img | Lưu ảnh lấy được từ path |
| **List<Image>** imgList | Lưu các phần của ảnh đã được cắt ra |
| **List<List<int>>** records | Lưu 3 thời gian hoàn thành sớm nhất theo độ khó |
| **Size** resizedImg | Lưu kích thước ảnh muốn đổi sau khi ảnh được lấy được từ **path** |
| **InitialGameForm** startFm | Lưu lại Form “**InitialGameForm**” để có thể quay lại khi thoát game |
| **Bool** isPressRepeatBtn | Kiểm tra xem nút **repeat** trên **Music Controls** có được nhấn hay không? |

Bảng 3.Mô tả các thuộc tính của class PuzzleGameForm

Graphical user interface, application

Description automatically generated

Hình 3. Mô tả class PuzzleGameForm

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Int** height | Chiều cao của board |
| **Int** width | Chiều rộng của board |
| **Int** numeberOfCol | Số cột ban đầu trên board |
| **Int** numberOfRow | Số hàng ban đầu trên board |
| **BoardManager** boardManager | Đối tượng class **BoardManager** để thực hiện các thao tác trên các Board |
| **Int** currentLevel | Độ khó hiện tại đang chơi |

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

**Method**

* + **CropIntoListImgs**: triển khai từ interface **IcropImage** dùng để cắt nhỏ ảnh.
  + **PuzzleGameForm\_Load:** khởi tạo các đối tượng của các class cần thiết và gọi method **Start** để bắt đầu các thao tác như vẽ bàn chơi, hiển thị ảnh,..
  + **Start**: dùng để khởi tạo các giá trị ban đầu cho các field, vẽ bàn chơi, cập nhật ảnh…
  + **GetListImageFromCurrentListPtrb**: lấy các ảnh từ các PictureBox hiện có để phục vụ việc trộn ảnh.
  + **mixToolStripMenuItem1\_Click**: bắt sự kiện khi click vào lựa chọn “**Mix**” sau đó thực hiện việc trộn ảnh bằng cách gọi Method **Mix** của class **GameFunction.**
  + **chooseLevelToolStripMenuItem\_Click**: Bắt sự kiện khi click vào lựa chọn “**Choose Level”,** sau đó gọi Method “**ChooseLevel**” của class **GameFunction**, hiển thị Form “**ChooseLevelForm**” để chọn độ khó.
  + **ChooseLvForm\_closeChooseLevelEvent**: Bắt sự kiện khi đóng Form “**ChooseLevelForm**” sau đó thực hiện cập nhật độ khó của game cũng như vẽ lại Board tương ứng độ khó.
  + **chooseImageToolStripMenuItem1\_Click**: Bắt sự kiện khi chọn lựa chọn **Choose Image**, sau đó gọi Method **ChooseImage** từ class **GameFunction** để hiển thị Form **ChooseImageForm** để chọn ảnh.
  + **ChooseImgForm\_closeChooseImageEvent**: Bắt sự kiện khi đóng Form **ChooseImageForm** sau đó thực hiện cập nhật lại hình ảnh đã chọn bằng cách gọi method **Start.**
  + **SetTimeForPowerForm**: lấy thời gian còn lại trên **powerGrpBox** của **PuzzleGameForm** gán vào field **timeList** của class **GameFunction** để cập nhật thời gian còn lại cho **PowerModifierForm.**
  + **powerToolStripMenuItem\_Click**: Bắt sự kiện khi chọn lựa chọn **Power**, sau đó gọi method **PowerModify** từ class **GameFunction** để hiển thị Form **PowerModifierForm.**
  + **PowerFm\_closePowerFormWhenClosingEvent**: Có cho phép hiển thị thời gian còn lại cho đến khi shutdown/restart trên **PuzzleGameForm** hay không, sau khi đóng Form **PowerModifierForm.**
  + **PowerFm\_closePowerFormWhenPressCancelEvent**: Ngừng bộ đếm thời gian trên **PuzzleGameForm** khi click Button “**Cancel**” rồi thoát Form **PowerModifierForm.**
  + **PowerFm\_closePowerFormWhenPressStartEvent**: Hiển thị thời gian còn lại cho đến lúc shutdown/restart trên **PuzzleGameForm** saukhi nhấn Button **Start** và đóng Form **PowerModifierForm.**
  + **Time\_ChangeEvent**: kiểm tra bộ đếm thời gian của tuỳ chọn trên **PowerModifierForm** kết thúc chưa, nếu rồi thì thực hiện thao tác tương ứng là shutdown/restart.
  + **CountDownTimeEvent**: kiểm tra bộ đếm ngược thời gian chơi kết thúc chưa, nếu rồi mà vẫn chưa thắng được sẽ xử thua và đưa ra các thao tác khác.
  + **PuzzleGameForm\_FormClosed:** Sự kiện xảy ra khi đóng **PuzzleGameForm**, sau đó ngưng tất cả bộ đếm thời gian, ngưng bài hát đang phát, và hiển thị lại Form **InitialGameForm.**
  + **Brd\_OnFilledImageInBlankBoardEvent:** sự kiện xảy ra khi hình ảnh đã được kéo thả đủ trên **PanelBlank** và bắt đầu kiểm tra việc hoàn tất game, nếu thắng thì cập nhật thời gian hoàn thành vào bảng Records, ngừng thời gian đếm ngược, đưa ra các lựa chọn chơi tiếp hay chơi lại.
  + **BrdImage\_OnClickPictureBox:** Sự kiện xảy ra khi click vào 1 ô PictureBox bất kì trên bàn chơi, trò chơi bắt đầu và thời gian bắt đầu đếm ngược.
  + **UpdateRecordTable:** cập nhật top 3 thời gian hoàn thành game nhanh nhất theo từng độ khó tương ứng lên bảng **Records.**
  + **levelCbx\_SelectedIndexChanged:** sự kiện xảy ra khi chọn level tương ứng trong bảng **Records** để xem lại thành tích.
  + **musicPlayerToolStripMenuItem\_Click:** bắt sự kiện khi chọn lựa chọn **Music Player**, sau đó gọi method **MusicPlayer** của class **GameFunction** và khởi động Form “**Music PlayerForm**”.
  + **timePlayNextSong\_Tick** và **MusicPlayerForm\_OnEndedMusic:** bắt sự kiện khi bài nhạc kết thúc và tuỳ theo lựa chọn lặp lại hay phát bài tiếp theo mà thực hiện.
  + **MusicPlayerForm\_OnPressPlayButtonMusicPlayer:** bắt sự kiện khi bài hát được phát để cập nhật trạng thái trình điều khiển cũng như tên bài hát trên Form “**PuzzleGameForm**”.
  + **repeatBtn\_Click, pauseBtn\_Click, playBtn\_Click, nextBtn\_Click, prevBtn\_Click:** các tuỳ chọn lặp lại, dừng/phát, bài tiếp theo/trước đó nằm trong trình điều khiển trên Form “**PuzzleGameForm**”.
  + **UpdataMusic:** cập nhật bài hát mới khi nhấn next/previous.
    1. **Chế độ chơi qua LAN**

**Lưu ý**:

* + Mỗi người chơi sẽ nhìn thấy trên giao diện 2 khu vực chơi, bên trái là của người chơi (**ClientBoard**), bên phải là của đối phương (**ServerBoard**), vì thế nên **ClientBoard** của người chơi này sẽ được hiển thị trên **ServerBoard** của đối phương.
  + Chỉ có phía Host mới có chức năng chọn độ khó và bắt đầu trò chơi.
  + Khi một bên bất kì chọn chức năng thì bên kia phải đợi đối phương hoàn thành xong và cập nhật lại các trạng thái cần thiết thì mới có thể thực hiện chức năng đó để tránh một số lỗi.
    - 1. **Class SocketManager**

Dùng để quản lý các thao tác làm host hoặc kết nối tới host khác bằng IP Address và Port

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Random** rd | Để random 1 port ngẫu nhiên |
| **Int** port | Lưu port random được |
| **Bool** isClient | Đánh dấu người chơi đang là **Client** hay **Host** |
| **TcpClient** client | Dùng để thiết lập kết nối bằng giao thức **TCP** tới **Host** |
| **TcpListener** listener | Dùng để đợi kết nối từ **Client** tới **Host** |
| **NetworkStream** netStream | Lấy Stream sau khi **Client** kết nối tới **Host**, dùng để gửi/nhận dữ liệu |
| **MultiplayForm** fm | Lưu đối tượng của **MultiplayForm** – nới thực hiện trò chơi |
| **SocketForm** skForm | Lưu đối tượng của **SocketForm** – nới nhập tên, IP Address, Port |
| **String** playerName | Lưu tên người chơi được nhập vào từ **SocketForm** |

Bảng 3.Mô tả các thuộc tính của class SocketManager

|  |  |
| --- | --- |
| **Event** | **Ý nghĩa** |
| **OnClientConnected** | Phát sự kiện khi có Client kết nối tới Host |

Bảng 3.Mô tả các event của class SocketManager

Graphical user interface, text, application

Description automatically generated

Hình 3. Mô tả class SocketManager

**Method**

* + **ConnectToServer**: Dùng để kết nối tới **Host** khi người chơi là **Client.**
  + **WaitForClientConnect**: Dùng để đợi **Client** kết nối tới nếu người chơi là **Server** và phát ra thông báo cho Form **MultiplayForm** khi có thiết bị kết nối đến.
  + **GetLocalIpv4Address**: Lấy địa chỉ **IPv4** của máy người chơi.
  + **Send**, **Receive**: dùng để gửi/nhận dữ liệu dạng object.
  + **StartForm**: Khởi tạo Form **MultiplayForm** để chơi giữa 2 người qua LAN.
    - 1. **Class SocketForm**

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **SocketManager** sk | Đối tượng của class **SocketManager** để thực hiện các thao tác tương ứng |
| **InitialGameForm** startFm | Dùng để hiện thị **InitialGameForm** khi đóng **SocketForm** |

Bảng 3.Mô tả các thuộc tính của class SocketForm

Hình 3. Mô tả class SocketForm

**Method**

* + **clientBtn\_Click**: sự kiện khi người chơi click vào Button **Client**, lúc này người chơi sẽ là **Client** và cần điền **IP Address**, **Port** để kết nối tới **Host.**
  + **connectBtn\_Click**: sự kiện khi người chơi click bào Button **Connect**, lúc này sẽ thực hiện việc kết nối tới **Host**, nếu **IP Address** hoặc **Port** không đúng sẽ báo lỗi cho người chơi nhập lại.
  + **serverbtn\_Click**: sự kiện khi người chơi click vào Button **Server**, lúc này người chơi sẽ là **Host**, Form **MultiplayForm** sẽ được khởi tạo và người chơi sẽ phải chời người chơi khác kết nối vào mới thao tác được.
  + **nameTxb\_TextChanged**: sự kiện khi nhập tên vào ô **textbox**, kiểm tra nếu tên bắt đầu bằng số hoặc vượt quá 6 kí tự sẽ báo lỗi.
  + **SocketForm\_FormClosed**: sự kiện xảy ra khi đóng **SocketForm**, lúc này sẽ thực hiện việc đóng luôn **InitialGameForm** đang được **Hide** trước đó.
    - 1. **Class MultiplayableData**

Đây là class để tạo đối tượng chứa dữ liệu cần gửi của cả 2 bên Host và Client

A picture containing graphical user interface

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Image** desImg | Chứa ảnh tại vị trí thả |
| **Image** srcImg | Chứa ảnh tại vị trí kéo |
| **Int** desPos | Chứa index PictureBox tại vị trí thả |
| **Int** srcPos | Chứa index PictureBox tại vị trí kéo |
| **List<Image>** initialImgs | Chứa danh sách các ảnh đã được cắt nhỏ ban đầu để đồng bộ giữa **Client** và **Host** lúc mới kết nối, hoặc khi nhấn Button **Mix** |
| **List<Image>** mixedImgsClient | Chứa danh sách các ảnh được cắt nhỏ **và được trộn** của bên **Host** trên **ClientBoard** để đồng bộ 2 bên lúc chọn độ khó khác |
| **List<Image>** mixedImgsServer | Chứa danh sách các ảnh được cắt nhỏ **và được trộn** của **Host** trên **ServerBoard** để đồng bộ 2 bên lúc chọn độ khó khác |
| **List<Image>** initialImgsListClient | Chứa danh sách các ảnh được cắt nhỏ của Host trên **ClientBoard** để đồng bộ 2 bên lúc chọn độ khó khác |
| **List<Image>** initialImgsListServer | Chứa danh sách các ảnh được cắt nhỏ của Host trên **ServerBoard** để đồng bộ 2 bên lúc chọn độ khó khác |
| **Image** img | Chứa ảnh gốc để làm hướng dẫn và được gửi khi thực hiện chọn độ khó |
| **Int** level | Độ khó bên Host đã chọn |
| **String** hour, **String** minute, **String** second | Thời gian tương ứng khi chọn độ khó và sẽ được gửi đi sau khi **Host** chọn độ khó |
| **Bool** isPressButton | Đánh dấu xem cả 2 bên có đang nhấn Button nào không |
| **String** msg | Gửi thông báo đến người chơi khác nếu bên còn lại có nhấn 1 Button nào đó |
| **Bool** isPressStartButton | Đánh dấu xem bên Host có nhấn Button **Start** chưa |
| **String** status | Gửi thông báo các trạng thái cần thiết như đã trộn ảnh xong, đã chọn level xong… để thông báo cho bên còn lại |
| **PlayerInfo** player | Thông tin về người chơi: điểm, tên, thời gian hoàn thành… |
| **Bool** isWin, **Bool** isLose | Nếu một bên hoàn thành hoặc hết giờ thì gửi thông báo cũng như cập nhật điểm số bên còn lại |

Bảng 3.Mô tả các thuộc tính của class MultiplayableData

Hình 3.Mô tả các thuộc tính

class MultiplayableData

* Các **Constructor** của class **MultiplayableData**
  + Thông báo Win/Lose và cập nhật điểm của người chơi.

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Thông báo trạng thái đang thực hiện của 2 bên đã hoàn thành hay chưa?

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Thông báo cho bên Client biết khi bên **Host** nhấn **Start** và bắt đầu đếm ngược thời gian chơi.

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Thông báo cho bên còn lại mình đang thao tác gì, click Button nào, ví dụ: chọn level, chọn ảnh, trộn ảnh…

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Thông báo bên **Host** đang chọn độ khó cho cả 2, vô hiệu hoá chức năng bên **Client** cho đến khi cả 2 bên cùng cập nhật lại độ khó.

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Đồng bộ ảnh 2 bên khi nhấn Button **Mix.**

Text

Description automatically generated

Hình 3. Mô tả constructor class MultiplayableData

* + Đồng bộ ảnh 2 bên khi mới kết nối và cập nhật tên người chơi phía còn lại.

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + Đồng bộ hiển thị ảnh kéo thả ở 2 bên trong lúc đang chơi.

Text

Description automatically generated

Hình 3.Mô tả constructor class MultiplayableData

* + - 1. **Class MultiplayManager**

Graphical user interface, text, application

Description automatically generated

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **BoardManager** client | Quản lí các đối tượng trong khu vực chơi phía người chơi |
| **BoardManager** server | Quản lí các đối tượng trong khu vực chơi phía đối phương |
| **MultiplayableData** dataSend | Đối tượng class **MultiplayableData** chứa dữ liệu cần được gửi/nhận |
| **SocketManager** socket | Quản lý các chức năng nhận, gửi dữ liệu |
| **List<PictureBox>** ptrbs | Danh sách các **PictureBox** trên cả 2 bên để quản lý đồng bộ việc kéo, thả ảnh |
| **MultiplayForm** fm | Đối tượng của class **MultiplayForm** để truy xuất, cập nhật các thuộc tính khi có thay đổi |

Bảng 3. Mô tả các thuộc tính của class MultiplayManager

Hình 3.Mô tả class MultiplayManager

**Method**

* + **GetAllPtrb**: Lấy **List<PictureBox>** từ phía **ServerBoard** để thực hiện việc cập nhật **Board** khi việc kéo thả xảy ra bên phía còn lại.
  + **AddEventToEachPtrb**: Gán event cho phép ghi lại vị trí cũng như hình ảnh ở mỗi **PictureBox** mỗi khi kéo/thả, nhằm gửi cho bên còn lại để đồng bộ.
  + **SetEventToPtrb**: Lấy **List<PictureBox>** ở **ServerBoard** rồi gán event cho phép việc kéo thả ảnh ở cả **ClientBoard** và **ServerBoard** mỗi khi việc đồng bộ giữa 2 bên diễn ra.
  + **ListenDataReceived**: Lắng nghe liên tục việc nhận dữ liệu trên một luồng khác khi bên còn lại bắt đầu gửi, sau đó thực hiện các thao tác tương ứng, cập nhật lại giao diện để đồng bộ tiến trình chơi giữa 2 bên.
  + **UpdateLevel**: thực hiện việc cập nhật lại giao diện để đồng bộ tiến trình chơi của 2 bên sau khi **Host** thực hiện việc chọn độ khó.
  + **UpdateBoard**: thực hiện việc cập nhật ảnh kéo thả giữa 2 bên khi 1 trong 2 người chơi thực hiện việc kéo thả.
  + **Start**: Bắt đầu việc gán **Event** cho các **PictureBox** cả 2 bên **ClientBoard**, **ServerBoard** và lắng nghe liên tục dữ liệu nhận được khi đối phương gửi.
  + **Ptrb\_MouseDown** và **Ptrb\_DragDrop**: thực hiện việc kéo thả ảnh và đồng thời gửi dữ liệu về ảnh và vị trí kéo thả cho đối phương để cập nhật lại.
    - 1. **Class MultiplayForm**

Đây là Form chính chứa các khu vực chơi để cả 2 bên thao tác

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Hình 3.Mô tả class MultiplayForm

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **Int** widthBoard | Chiều rộng bàn chơi |
| **Int** heightBoard | Chiều cao bàn chơi |
| **Size** resizedImg | Ảnh gốc được thay đổi kích thước |
| **String** path | Đường dẫn file ảnh cần hiển thị |
| **Int** currentLevel | Độ khó hiện tại |
| **Int** numberOfRow | Số lượng hàng trên 1 Board |
| **Int**  numberOfCol | Số lượng cột trên 1 Board |
| **Bool** isClient | Đánh dấu xem người chơi đang là **Host** hay **Client** |
| **Image** img | Lưu ảnh đọc được từ đường dẫn |
| **List<Image>** imgListServer | Chứa danh sách ảnh được cắt ra trên **ServerBoard** |

Bảng 3. Mô tả các thuộc tính của class MultiplayForm

|  |  |
| --- | --- |
| **Field** | **Ý nghĩa** |
| **List<Image>** imgListClient | Chứa danh sách ảnh được cắt ra trên **ClientBoard** |
| **SocketManager** socket | Quản lý các thao tác gửi/nhận dữ liệu |
| **BoardManager** clientBoard | Quản lý phần **ClientBoard** phía người chơi |
| **BoardManager** serverBoard | Quản lý phần **ServerBoard** phía **Host** |
| **MultiplayManager** multiplayManager | Quản lý các thao tác cập nhật giao diện khi 2 người chơi thao tác |
| **Size** sizeOfPtrb | Kích thước của một **PictureBox** tuỳ vào độ khó được chọn |
| **Bool** isWin | Đánh dấu người chơi đã hoàn thành hay chưa |
| **Bool** isDisconnected | Đánh dấu đối phương đã ngắt kết nối hay chưa |

**Method**

* + **Start**: Dùng để vẽ khu vực để chơi của 2 bên, không cho phép chơi khi chưa đủ 2 người, hiển thị ảnh gợi ý, cập nhật độ khó hiện tại.
  + **MultiplayForm\_Load**: sự kiện xảy ra khi khởi động Form ban đầu, dùng để khởi tạo các đối tượng, thiết lập một số thuộc tính.
  + **Socket\_OnClientConnected**: bắt sự kiện khi **Client** kết nối tới **Host** để hiển thị tên 2 người chơi cũng như đồng bộ thiết lập giữa 2 bên.
  + **BoardManager\_OnFilledPictureBox**: Sự kiện xảy khi khi người chơi kéo thả đủ ảnh vào khu vực chơi và bắt đầu kiểm tra xem có hoàn thành trò chơi hay không, nếu có thì gửi thông báo đến đối phương về điểm số cũng như thời gian hoàn thành.
  + **SetAttribute**: thiết lập các thuộc tính như số hàng, số cột, hình ảnh và vẽ bàn chơi cho cả **Client** và **Host.**
  + **DisablePtrbs** và **ReleasePtrbs**: Dùng để khoá/mở khoá khu vực chơi khi đáp ứng đủ yêu cầu, ví dụ: có đủ 2 người chơi hoặc cả 2 cùng hoàn thành.
  + **GetListImageFromCurrentListPtrb**: Lấy danh sách các ảnh hiện tại trên khu vực chơi.
  + **mixBtn\_Click**: bắt sự kiện khi click Button **Mix**, trộn các hình ảnh phía người chơi và gửi dữ liệu để cập nhật nó ở phía đối phương.
  + **chooseImgBtn\_Click**: bắt sự kiện khi click Button **Choose Image**, lựa chọn ảnh sau đó gửi dữ liệu để cập nhật nó ở phía đối phương.
  + **ProcessWhenChooseImage**: xử lí việc cập nhật hình ảnh phía người chơi.
  + **ChooseImgForm\_closeChooseImageEvent**: bắt sự kiện khi đóng Form **ChooseImageForm**, thực hiện đồng bộ dữ liệu giữa 2 người chơi.
  + **chooseLvBtn\_Click**: bắt sự kiện khi click Button **Choose Level** ở phía **Host.**
  + **ProcessWhenChooseLevel**: sau khi chọn độ khó xong sẽ vẽ lại bàn chơi, cập nhật ảnh, thời gian bên **Host.**
  + **ChooseLvForm\_closeChooseLevelEvent:** Sự kiện xảy ra khi đóng Form “**ChooseLevelForm**”, để đồng bộ tiến trình chơi giữa 2 bên bằng cách gửi dữ liệu về ảnh, thời gian chơi, độ khó cho bên còn lại.
  + **MultiplayForm\_FormClosing:** Khi một trong 2 thoát khỏi trò chơi thì sẽ thông báo cho bên còn lại và tự động ngắt kết nối rồi thoát.
  + **start\_Btn\_Click:** Sự kiện xảy ra khi bên Host click Button **Start**, sau đó gửi thông báo cho bên còn lại và bắt đầu đếm giờ.
  + **GameFunction\_TimeEndEvent:** Sự kiện xảy ra khi thời gian đếm hết, nếu chưa hoàn thành trò chơi sẽ xử thua và thông báo cho bên còn lại.
  + **Label\_TextChanged:** Sự kiện xảy ra khi nội dung Label thay đổi, mục đích để canh thời gian sao cho khi cả 2 bên hoàn thành trò chơi thì mới ngừng đếm giờ.
  + **aboutBtn\_Click:** Hiện thông tin về chế độ chơi khi click Button About.
  1. **Sơ đồ tuần tự các thao tác**
     1. **Thao tác thiết lập bàn chơi đơn**

**Diagram

Description automatically generated**

Hình 3.Thao tác thiết lập bàn chơi đơn

* + 1. **Thao tác chức năng trộn hình ảnh**

**Diagram

Description automatically generated**

Hình 3.Thao tác chức năng trộn ảnh

* + 1. **Thao tác chức năng chọn độ khó**

**Chart

Description automatically generated with low confidence**

Hình 3.Thao tác chức năng chọn độ khó

* + 1. **Thao tác chức năng chọn ảnh**

**A picture containing chart

Description automatically generated**

Hình 3.Thao tác chức năng chọn hình ảnh

* + 1. **Thao tác chức năng hẹn giờ tắt/khởi động lại máy**

**Diagram

Description automatically generated with low confidence**

Hình 3.Thao tác chức năng tuỳ chọn tắt/khởi động lại máy

* + 1. **Thao tác chức năng nghe nhạc**

**Chart

Description automatically generated with medium confidence**

Hình 3.Thao tác chức năng nghe nhạc

* + 1. **Thao tác chức năng kết nối Host/Client**

**Diagram

Description automatically generated**

Hình 3.Thao tác chức năng chọn Host/Client

* + 1. **Thao tác thiết lập chế độ chơi qua LAN**

**Diagram

Description automatically generated with low confidence**

Hình 3.Thao tác chức năng chơi LAN

# **CHƯƠNG 4: KẾT LUẬN**

Về kết quả đạt được qua môn học OOP, chúng em đã tiếp thu được các kiến thức về tư duy trong lập trình hướng đối tượng; các thành phần chính của .NET framework; các thành phần cơ bản và cú pháp của ngôn ngữ C#; thành phần trong OOP: lớp đối tượng, thuộc tính, phương thức, constructor; các khái niệm và cách thức hoạt động của abstract class, interface,…Ngoài ra còn có các kiến thức về exception xử lí biệt lệ, truy xuất tệp tin,… Hơn hết, quan trọng nhất vẫn là 4 đặc tính cơ bản của OOP: sự đóng gói, sự kế thừa, sự đa hình, sự trừu tượng; cách thức xây dựng giao diện cơ bản winform: sử dụng button, text, label, picture box, group box,…

Sản phẩm project game xếp hình này là kết quả cuối cùng của chúng em sau một kì học môn lập trình hướng đối tượng. Chúng em đã cố gắng hoàn thiện đồ án tốt và chỉn chu nhất có thể. Tuy nhiên bên cạnh đó sản phẩm vẫn còn tồn tại một số khuyết điểm chưa lường hết. Trong quá trình code có thể một số lỗi chưa được phát hiện và xử lí triệt để, chưa tối ưu được hết code…Đồng thời giao diện game kéo thả không thực sự thân thiện với người dùng. Trong quá trình thực hiện đồ án, đôi lúc gặp nhiều khó khăn về ý tưởng cũng như xử lí code. Chúng em rất mong sự góp ý và nhận xét từ thầy để có thể cải thiện và hoàn thiện hơn ở các môn học sau. Chúng em xin trân trọng cảm ơn thầy!