**Final Project – Minesweeper Game**

* ***Name: Phan Tuấn Kiệt***
* ***Class : 23TT2***
* ***ID : 23125062***

1. **Danh sách chức năng (Hoàn thành được tất cả các yêu cầu + Bổ sung vài tính năng):**

* **Cơ bản (được yêu cầu) – Hoàn thành:**
* Làm toàn bộ game trên console, chỉ dùng những thư viện cơ bản của C++ để tạo đồ họa, vẽ bảng, tô màu, tạo chiến thuật, lưu game (Hoàn thành).
* Vẽ bảng (bao gồm ô, bom và cờ) (Hoàn thành).
* Thay đổi trạng thái của ô khi được chọn (Hoàn thành).
* Tính thời gian kể từ lúc chơi (Timer) (Hoàn thành).
* Tạo các functions: New game, Continue, Score, Information, Exit. Thay đổi chế độ chơi (board dimensions). Chọn một chức năng từ màn hình console (Hoàn thành).
* Tạo ngẫu nhiên bom (Hoàn thành).
* Nhận tín hiệu từ bàn phím và thay đổi con trỏ tới vị trí mới (Hoàn thành).
* Lưu game đang chơi và có thể chơi tiếp (continue) trong lần chơi tiếp theo (Hoàn thành).
* Lưu điểm cao nhất + tên theo từng chế độ chơi (Hoàn thành).
* **Other features (được yêu cầu có làm hoặc không) – Hoàn thành:**
* Sử dụng bàn phím để chọn và thay đổi vị trí con trỏ, chọn ô (Hoàn thành).
* Tạo hình và đồ họa trên console (Hoàn thành).
* Sử dụng cấu trúc Struct để tạo và quản lí con trỏ. Sử dụng cấp phát động và binary files để lưu mảng, quản lí ô. (Hoàn thành).
* **Các chức năng thêm của game (không có trong mục yêu cầu, người làm game tự tạo thêm):**

1. **Trước khi vào game:**

* Trước khi vào chơi có giới thiệu, yêu cầu người dùng bật full màn hình. Nếu người dung muốn chơi thì bấm YES (hoặc yes). Nếu không thì game sẽ tự động thoát ra khi người dùng nhập NO.
* Khi người dùng đã bấm YES, hiện sang một trang mới có đưa ra 4 lưu ý quan trọng nhất khi chơi game, bao gồm các hướng dẫn cơ bản (những thứ cơ bản nhất để chơi game như nút, phím bấm). Mục đích giúp người chơi có thể chơi được liền luôn mà không cần phải đọc mục Information quá nhiều. Nếu người chơi đồng ý với các lưu ý được nêu trên thì nhấn OK, sẽ được vô game chơi. Còn nếu người chơi bỗng không muốn chơi thì có thể không nhấn OK, game sẽ tự động thoát khi gặp bất kì dòng nào khác.

1. **Khi đã vô game (Màn hình chính):**

* Màn hình chính gồm 5 functions: New game, Continue, Scores, Information, Exit.
* Luôn hiển thị “Last time warning” để người dùng có thể tự biết thoát ra vô lại khi game gặp lỗi hoặc lag.
* Tạo đồ họa thanh cầu vòng giúp game đẹp hơn trên màn hình console.

1. **Khi bấm vô mục New game:**

* Có 3 mức độ chơi bao gồm: Easy, Medium, Hard.
* Khi vào chơi sẽ tính thời gian. Ở 2 bên góc màn hình luôn có hướng dẫn chơi từ bàn phím để tránh lúc người dùng quên.
* Trên màn hình chơi sẽ hiển thị các mục: Tên người tạo game, Mapsize, Number of bombs, Time và Status.
* Số lượng bom sẽ giảm khi người chơi đặt cờ.
* Thanh trạng thái Status sẽ đổi sang Win hoặc Lose tương ứng lúc Thắng – Thua.
* Đang chơi nếu thoát (ESC) và bấm SAVE thì game sẽ được lưu lại. Người chơi có thể vô chơi lại lần chơi tiếp theo mà không bị mất dữ liệu hay thời gian rest.

1. **Khi bấm vô mục Continue:**

* Tiếp tục chơi ở lần lưu gần nhất.

1. **Mục Scores:**

* Lưu tên người chơi có điểm cao nhất (nhanh nhất) theo từng chế độ.
* Game được thiết kế theo kiểu hồi hộp, tức là chỉ khi nào người chơi thắng kỉ lục cao nhất trong Scores thì bảng lưu tên mới hiện lên, còn không thì sẽ được chúc may mắn lần sau.

1. **Mục Information:**

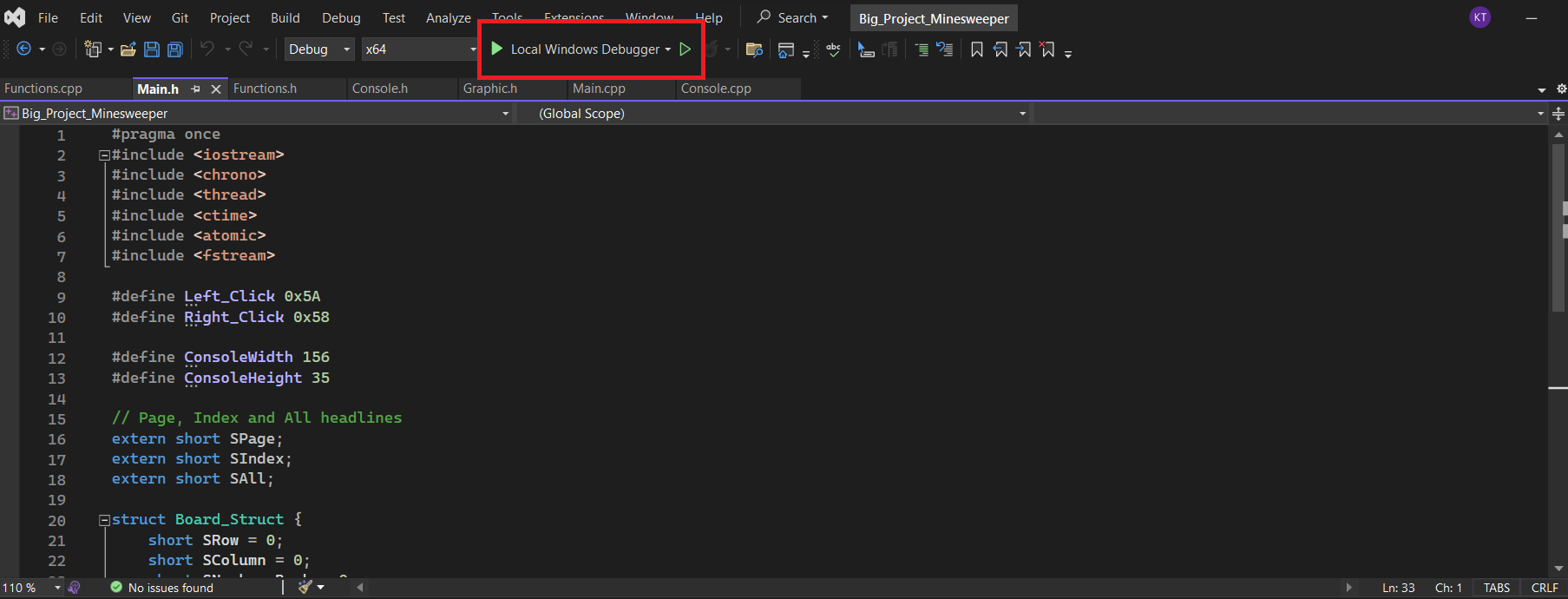
* Bao gồm Introduction (Giới thiệu bản thân), Instructions (Chi tiết cách chơi) và thông tin sơ lược của bản thân (để liên lạc).

1. **Mục exit:**

* Khi bấm exit sẽ hiển thị lên một trang mới gửi lời cảm ơn, chúc ngày mới tốt lành, sau đó game sẽ tự thoát.
* Ở bất kì trang nào, bấm ESC sẽ quay lại trang trước đó hoặc là thoát ra hẳn game (nếu ở trang đầu), hoặc hiện mục lưu (lúc đang chơi game).
* Có tính năng tự động lưu khi bấm (Exit) mà quên lưu (trong trang chơi game).
* Lúc thắng nếu cao điểm hơn người chơi nhanh nhất, người chơi sẽ được thông báo rằng mình đã phá vỡ kỉ lục và được yêu cầu để nhập tên. Nếu không cao điểm hơn người chơi nhanh nhất thì được thông báo là bạn đã chơi rất hay rồi, cố gắng them lần sau!
* Khi bấm continue, để tránh việc người chơi quên mất là mình đang chơi dở dang thì hệ thống sẽ tự lưu lần chơi gần nhất của người chơi, dù cho người chơi có bấm vô chơi mới nhưng nhớ ra rằng mình có lưu thì vẫn có thể exit ra và continue để chơi tiếp. Còn nếu đã chơi mới mà bấm lưu game mới thì phần continue sẽ lưu theo lần chơi vừa rồi.
* Game được thiết kế theo hướng kiểu auto solve and save (Tự động xử lí các lỗi sai của người chơi hoặc các chức năng sẽ giúp người chơi hạn chế tối đa lỗi kĩ thuật)

1. **Hướng dẫn chơi game**

**B1) Chạy game:**



**B2) Bật full màn hình rồi nhấn YES (Phải bật full màn hình để không bị lỗi do khác máy):**

A screenshot of a computer

Description automatically generated

**B3) Đọc kĩ hướng dẫn chơi game rồi nhấn OK**

A screenshot of a computer

Description automatically generated

**B4) Vô game và trải nghiệm các tính năng:**

A screenshot of a computer

Description automatically generated

1. **Cấu trúc mã nguồn**
2. **Danh sách các hàm**
3. ***Vận hành game***

* void make\_2DArray(): Tạo ma trận 2 chiều, dùng con trỏ và cấp phát động để quản lí. Ở mỗi hàng có một con trỏ quản lí con trỏ phụ ở mỗi cột. Trong mỗi cột, cấp phát động một mảng quản lí từng ô.
* void delete\_2DArray(): Xóa mảng khi không cần dùng nữa, tránh dư bộ nhớ.
* void initial(short SRow, short SColumn, short SNumber\_Bomb): Khởi tạo ma trận với số hàng – số cột và số bom cho trước.
* void drawCell(short SX, short SY, short SKieu): Vẽ ô tới vị trí x,y trên ma trận
* void drawBoard(): Vẽ bảng.
* void randomBomb(): Tạo bom ngẫu nhiên.
* void produceBomb(): Đặt bom tới vị trí ngẫu nhiên.
* void handleEvent(): Xử lý tín hiệu nhận vào từ bàn phím.
* void win(): Xử lí điều kiện khi thắng.
* void lose(): Xử lí điều kiện khi thua.
* void drawDescript(): Vẽ chữ design và thông báo cho người chơi lúc mới bắt đầu bấm vào trò chơi.
* void drawNotice(): Vẽ điều cần chú ý khi chơi game.
* void drawWarn(): Vẽ điều cần chú ý và cần thực hiện khi có lỗi xảy ra.
* void desCript(): Tạo introduction, instructions và about.
* void TextColor(int x): Tạo màu cho chữ.
* void gotoxy(int x, int y): Di chuyển con trỏ tới vị trí x, y trên ma trận.
* void Time(int total\_time): Hàm vẽ thời gian, chuyển thời gian sang dạng hh:mm:ss
* void saveGame(): Lưu trò chơi khi exit.
* void loadGame(): Khởi tạo trò chơi từ lần chơi gần nhất (chơi tiếp).
* void Score(): Vẽ trang score.
* void saveGameScore(): Lưu điểm
* void loadGameScore(): Load điểm chơi.
* void saveGameTime(): Lưu điểm cho hạng mục easy.
* void loadGameTime(): Load điểm cho hạng mục easy.
* void Time3(): Tạo thời gian cho hạng mục easy.
* void Time4(): Tạo thời gian cho hạng mục medium.
* void Time5(): Tạo thời gian cho hạng mục hard.
* void saveGameTime4(): Lưu điểm cho hạng mục medium.
* void loadGameTime4(): Load điểm cho hạng mục medium.
* void saveGameTime5(): Lưu điểm cho hạng mục hard.
* void loadGameTime5(): Load điểm cho hạng mục hard.

1. ***Một số hàm vẽ game + trang trí***

* void drawTitle(): Vẽ title cho game.
* void drawStatus(short Status, short SMode, short SIndex): Vẽ status cho game (Map, Bombs, Win or Lose, Save, Play Again, Exit, …)
* void drawMainMenu(short SIndex): Vẽ menu chính.
* void drawLevelMenu(short SIndex): Vẽ menu chọn level.

1. **Các struct khai báo**

* **Board\_Struct:** Quản lí số hàng, số cột, số bom, số ô đã mở, số lượng cờ đã đặt.
* **Cell\_Struct:** Quản lí ô (ô có bom, ô đã mở, ô có cờ, số bom xung quanh 1 ô).

1. **Cấu trúc file**

* ***Bốn file .h:***

**+** Console.h: Thư viện quản lí các hàm cơ bản trên console của C++.

**+** Functions.h: Quản lí các hàm tạo game chính.

**+** Graphic.h: Một số hàm đồ họa cơ bản của C++.

+ Main.h: Khai báo dài và rộng của console, quản lí board struct và cell struct.

* ***Ba file .cpp:***

+ Console.cpp: Các functions cơ bản trên console của C++.

+ Functions.cpp: Các hàm chính phục vụ game.

+ Main.cpp: Trang chính quản lí cũng như gọi các hàm của game.

1. **Luồng chạy**

* Trong hàm main, bắt đầu gọi hàm drawDescript() để chào người chơi và hỏi xem liệu có sẵn sàng không. Sau đó nếu người chơi nhập Yes thì gọi hàm drawNotice(). Nếu người chơi tiếp tục nhấn OK thì bắt đầu gọi hàm drawTitle, drawMainMenu, và gọi hàm xử lí sự kiện đầu vào từ bàn phím handleEvent().
* Trong handleEvent() sẽ lần lượt xử lí các sự kiện nhận vào từ bàn phím. Nếu một sự kiện nhận thỏa một hàm cho trước nào đó thì sẽ gọi hàm đó để xử lí song song với việc nhận thông tin từ bàn phím (nhưng vẫn tách biệt 2 cái theo thứ tự - gọi hàm rồi tiếp tục chờ nhận lệnh input từ bàn phím).