**ĐỒ ÁN MÔN KỸ THUẬT LẬP TRÌNH: GAME 2048**

1. **TỔNG QUAN VỀ ĐỒ ÁN**
2. ***Giới thiệu đồ án***

* Đây là đồ án mô phỏng trò chơi 2048 được thiết kế bằng ngôn ngữ C++. Đồ án nhằm mục đích vận dụng các kiến thức trong môn học Kỹ thuật lập trình và Nhập môn lập trình.
* Đây là đồ án do cá nhân sinh viên thực hiện, ngoại trừ những phần đã ghi nhận, cảm ơn khi sử dụng công trình/công việc của người khác.

1. ***Tác giả***

* Họ và tên: Trần Anh Khoa
* Lớp: 23CTT2
* Mã số sinh viên: 23120135

1. ***Các cài đặt***

* Tải file zip chứa các tệp tin mã nguồn.
* Sau khi giải nén, trích xuất các tệp tin đó vào chung một thư mục.
* Ấn vào GAME2048.sln và chạy Game bằng Visual Studio.

1. **CHI TIẾT VỀ ĐỒ ÁN**
2. ***Về mã nguồn***

* **README.txt** là file hướng dẫn sử dụng
* **GAME2048.cpp** là file chứa hàm main của mã nguồn
* **Functions.h** là file chứa khai báo của tất cả các thư viện, struct, define các giá trị làm hằng số để mã nguồn trở nên dễ đọc hơn. Ngoài ra, file này cũng chứa khai báo của các hàm thực hiện chức năng trong game như: *processUp, processDown, processUndo, processRedo, …*
* **Model.h** là file chứa khai báo của các hàm để khởi tạo và thực hiện việc quản lý, lưu trữ và tải dữ liệu của trò chơi từ các tệp tin như: *initializeGame, updatelist, …*
* **Graphics.h** là file chứa khai báo của các hàm vẽ giao diện cho trò chơi như*: setColor, drawBox, …*
* **Functions.cpp** là file chứa định nghĩa của các hàm được khai báo trong file **Functions.h**
* **Model.cpp** là file chứa định nghĩa của các hàm được khai báo trong file **Model.h**
* **Graphics.cpp** là file chứa định nghĩa của các hàm được khai bái trong file **Graphics.h**
* **player.bin** là file lưu trữ thông tin của người chơi trong top 20.
* **listAccount.bin** là file lưu trữ thông tin của các người chơi đang được resume.
* Các file có dạng **AccountX.bin** *(1 <= X <= 5)* là các file lưu trữ trạng thái game đã được resume.
* Các file có dạng **UndoX.bin** *(1 <= X <= 5)* là các file lưu trữ dữ liệu của UndoStack để đảm bảo chức năng Undo vẫn được thực hiện khi chơi ở chế độ resume.
* Các file có dạng **RedoX.bin** *(1 <= X <= 5)* là các file lưu trữ dữ liệu của RedoStack để đảm bảo chức năng Redo vẫn được thực hiện khi chơi ở chế độ resume.

1. ***Các chức năng đã thực hiện được theo yêu cầu***

* Hiển thị được giao diện kèm chức năng của hai màn hình chính.
* Thực hiện được chức năng nhập tên và kiểm tra tính trùng lặp (có chuẩn hóa tên theo một form chung, cụ thể: tên người chơi dài tối đa 20 ký tự, giữa các chữ với nhau chỉ có một khoảng trắng và không có khoảng trắng dư thừa ở đầu và cuối tên).
* Thực hiện được chức năng cài đặt độ lớn của bàn cờ.
* Thực hiện được việc lưu trữ thông tin người chơi.
* Thực hiện được chức năng Resume và Resume cho 5 tài khoản. Việc chơi bằng một tài khoản đã được Resume trước đó vẫn đảm bảo sử dụng được các chức năng đã được kích hoạt trước đó.
* Thực hiện được chức năng Exit.
* Thực hiện được logic trong trò chơi: phát sinh ngẫu nhiên, di chuyển, hiển thị điểm, hiển thị trò chơi, ...
* Thực hiện được chức năng undo, undo và redo.
* Hiển thị được các thông báo và nội dung tương ứng.
* Tổ chức thành các file rõ ràng.
* Sử dụng phương thức đồ họa kèm giao diện thân thiện.

1. ***Các chức năng sinh viên tự đề xuất***

* Chức năng **New Game**: Chức năng này mặc định xuất hiện trong lúc người chơi vào màn hình chính để chơi game. Nó cho phép người chơi bắt đầu lại một Game mới theo 2 lựa chọn:

+ Lựa chọn 1: Bắt đầu lại từ màn hình bắt đầu để nhập lại tên hoặc điều chỉnh chỉnh lại phần Settings mà không thực hiện thao tác save game như việc ấn nút Exit.

+ Lựa chọn 2: Chỉ tạo lại bảng game mới, tên và các cài đặt trước đó trong phần Settings sẽ không bị thay đổi.

* Chức năng **RemoveBlock**: Chức năng này có thể tắt hoặc bật tùy thuộc vào người chơi. Để điều chỉnh chức năng này, người chơi vào mục *"Advanced Function"* trong giao diện màn hình bắt đầu. Chức năng này cho phép người chơi loại bỏ một ô bất kì trong bảng game. Tuy nhiên, mỗi lần thực hiện thì sẽ bị trừ 100 điểm. Tất nhiên, không thể thực hiện chức năng này khi số điểm hiện tại ít hơn 100 điểm.
* Chế độ **Random Color**: Chế độ này có thể tắt hoặc bật tùy thuộc vào nguời chơi. Người chơi sẽ điều chình chế độ này trong mục *"Advanced Function"* trong giao diện màn hình bắt đầu. Chế độ này cho phép bảng game thay đổi màu sắc bất kì sau mỗi lần di chuyển hoặc thậm chí là undo, redo, hay removeBlock...

1. **TỰ ĐÁNH GIÁ VÀ NHẬN XÉT**
2. ***Ưu điểm***

* Xây dựng thành công tựa game 2048 dựa trên màn hình Console, không gặp lỗi trong quá trình chạy.
* Cài đặt được hầu hết các chức năng theo yêu cầu của Giảng viên.
* Tự đề xuất thêm chức năng ngoài yêu cầu của đồ án.

1. ***Nhược điểm***

* Code tuy chạy được nhưng chưa thật sự tối ưu, clean code chưa thật sự triệt để.
* Chưa thực hiện được chức năng mã hóa dữ liệu.

1. **CÁC NGUỒN THAM KHẢO**
2. ***Nguồn tham khảo về cách xây dựng và sử dụng hàm gotoxy trong việc thiết lập đồ họa:***

<https://www.youtube.com/watch?v=9gWIbhOudEk&t=1275s>

1. ***Nguồn tham khảo về cách ẩn/hiện con trỏ trên màn hình console:***

<https://daynhauhoc.com/t/an-con-tro-chuot-tren-man-hinh-console/34411>

1. ***Nguồn tham khảo về cách sử dụng màu sắc trong C++:***

<https://gist.github.com/fnky/458719343aabd01cfb17a3a4f7296797>

1. ***Nguồn tham khảo trong việc vẽ logo:***

<https://patorjk.com/software/taag/#p=display&f=3D%20Diagonal&t=Type%20Something%20>

1. **VIDEO MÔ PHỎNG CHO TRÒ CHƠI**

Các chi tiết về chức năng, cách chơi, giao diện các màn hình và việc kiểm tra tính đúng đắn của các chức năng cũng như là demo game sẽ được thể hiện ở video được truy cập thông qua đường dẫn sau đây:

<https://drive.google.com/drive/folders/1RSUZZLhHDZnHm6QOyVXgccRXdGtSBGh7?usp=drive_link>