**Báo cáo đồ án môn Lập trình Hướng đối tượng**

**GAME 2048**

**MSSV: 1612647**

**Menu**

[1. Hướng dẫn sử dụng: 2](#_Toc501200790)

[1.1. Giao diện Menu chính: 2](#_Toc501200791)

[1.2. New Game: 2](#_Toc501200792)

[1.3. Giao diện game: 3](#_Toc501200793)

[1.3.1. Cách chơi: 3](#_Toc501200794)

[1.3.2. Thắng cuộc: 4](#_Toc501200795)

[1.3.3. Thua cuộc: 4](#_Toc501200796)

[1.4. Load Game: 5](#_Toc501200797)

[1.5. Score Board: (Xem top 10 người chơi điểm cao nhất) 5](#_Toc501200798)

[1.6. Giao diện Pause menu: 6](#_Toc501200799)

[2. Mô tả source code: 7](#_Toc501200800)

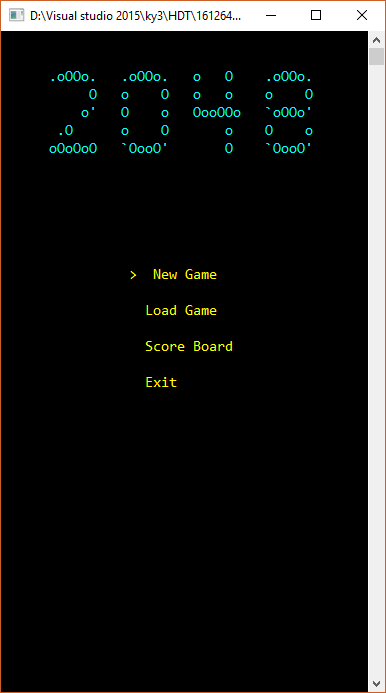
[2.2. Class Grid 7](#_Toc501200801)

[2.3. Class SroreBoard 8](#_Toc501200802)

[2.4. Class Game 8](#_Toc501200803)

[2.5. Class GameUI 8](#_Toc501200804)

[2.6. Các hàm đồ hoạ hỗ trợ 8](#_Toc501200805)

1. **Hướng dẫn sử dụng**:
   1. **Giao diện Menu chính**:

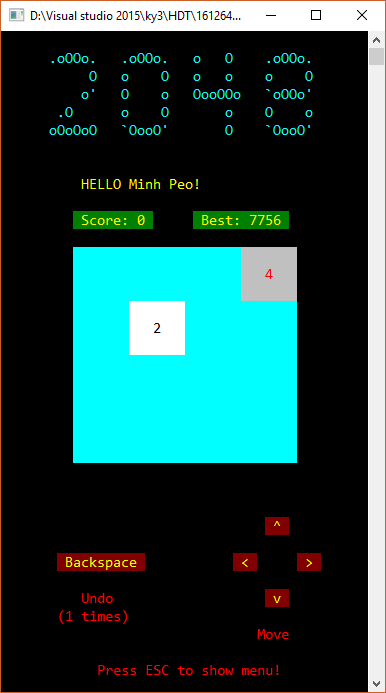
Nhấn **phím mũi tên** để **di chuyển** giữa các mục và nhấn **Enter** để **chọn**.

Chọn **Exit** để thoát khỏi trò chơi.

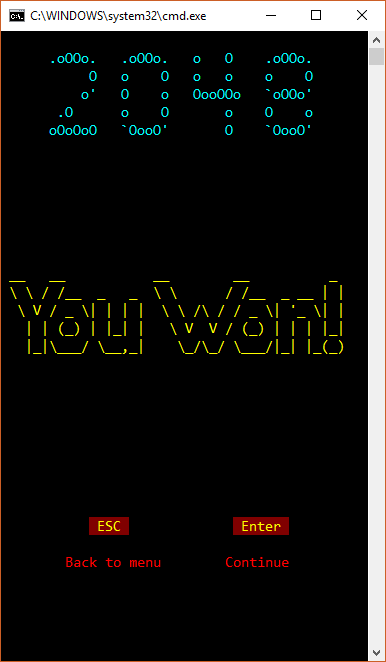
* 1. **New Game**:

****Bắt đầu màn chơi mới.

Sau khi chọn Chế độ New Game. Người chơi cần nhập Tên người dùng để bắt đầu chơi.

* 1. **Giao diện game:**
* **Score**: Hiển trị **số điểm hiện tại** của người chơi.
* **Best**: Hiển thị **số điểm cao nhất** mà người chơi đạt được.
* **(… times):** Hiển thị **số lần undo** còn lại.
  + 1. **Cách chơi:**
* Nhấn các **phím mũi tên** để **dồn** các khối vuông về cùng 1 phía. Nếu có **2 ô liên tục điểm bằng nhau** thì 2 ô sẽ được **gộp** và người chơi sẽ được cộng số điểm **bằng giá trị của ô kết quả**.
* **Undo**: Người chơi có thể quay lại lượt chơi trước đó **1 lần**, nếu đã thực hiện undo thì người chơi phải **chơi tiếp** **ít nhất 1 lượt** mới lại có thể thực hiện Undo.
* Nhấn **phím** **ESC** để **pause game** và **mở menu**.
  + 1. **Thắng cuộc:**

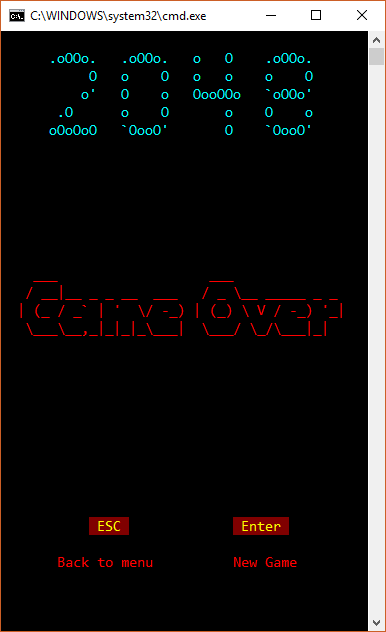
Người chơi sẽ giành chiến thắng khi tạo được ô có giá trị **2048**.



Giai diện khi giành chiến thắng

* Nhấn **ESC** để quay về **menu chính**.
* Nhấn **Enter** để **tiếp tục chơi tiếp** và gia tăng số điểm hiện có.
  + 1. **Thua cuộc:**

Người chơi sẽ thua cuộc khi **không thể di chuyển** được các ô bằng phím mũi tên nữa.

* Nhấn **ESC** để quay về **menu chính**.

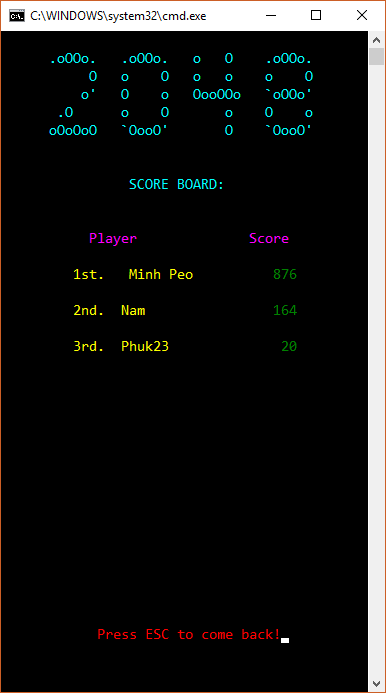
Giao diện Game Over

* Nhấn **Enter** để **bắt đầu lại** màn chơi mới.
  1. **Load Game:**

Tiếp tục chơi với dữ liệu đã được lưu trước đó. Nếu lần trước GameOver thì game được tạo mới tương tự New Game.

* 1. **Score Board:** (Xem top 10 người chơi điểm cao nhất)

Chương trình sẽ **hiển thị tên và số điểm** của **tối đa 10 người chơi** có **điểm số cao nhất**.



* 1. **Giao diện Pause menu:**
* **Restart:** Bắt đầu game mới giống chức năng New Game.
* **Resume:** Tiếp tục chơi
* **Score Board:** Xem top 10 người chơi điểm cao nhất.
* **Exit:** Thoát khỏi trò chơi.



1. **Mô tả source code**:

**2.1. Class Block**

- Các thuộc tính: x, y (vị trí ô trong console) và value (giá trị của ô).

- Các phương thức tiêu biểu:

+ Hàm dựng.

+ GetValue, SetValue, GetX, GetY.

+ IsEmpty, SetEmpty: kiểm tra là ô trống, Set ô trống (giá trị 0).

+ Show: Hiển thị ô ra màn hình.

+ Move: Di chuyển ô đến vị trí khác.

+ Merge: Gộp 1 ô khác.

## **2.2. Class Grid**

(Lưới 4x4 hiển thị các ô vuông)

- Các thuộc tính: x,y (vị trí hiển thị trong console) và mảng 2 chiều các Block.

- Các phương thức tiêu biểu:

+ Hàm dựng mặc định: Phát sinh lưới mới với ngẫu nhiên 2 ô có giá trị.

+ MoveUp, MoveDown, MoveLeft, MoveRight: Tìm ô trống đầu tiên và di chuyển ô hiện tại đến vị trí đó.

+ SlideUp, SlideDown, SlideLeft, SlideRight: Trượt tất các các ô lên trên hoặc xuống dưới, sang trái, sang phải.

+ IsFull, IsOver, IsWin: kiểm tra lưới đầy, kiểm tra thua cuộc, thắng cuộc.

+ AddNewBlock: Thêm ô trống mới có giá trị ngẫu nhiên tại vị trí ngẫu nhiên.

+ Save: Lưu lại lưới vào file.

+ Load: Đọc thông tin lưới từ file.

## **2.3. Class SroreBoard**

(Lưu thông tin bảng điểm top 10 người chơi)

- Các thuộc tính: size (số lượng người chơi), name (tên), score(số điểm).

- Các phương thức tiêu biểu:

+ Hàm dựng.

+ Input, Output: Đọc thông tin từ file và ghi thông tin vào file.

+ GetBestScore: Tìm điểm số cao nhất.

+ AddNewUser: Thêm người chơi với điều kiện điểm số thuộc top 10.

+ DescendingScore: Sắp xếp giảm dần theo điểm số.

+ GetScoreBoard: Lấy thông tin score board.

## **2.4. Class Game**

(Kế thừa từ lớp GameUI)

- Các thuộc tính: x,y (vị trí hiển thị trong console) và mảng 2 chiều các Block.

- Các phương thức tiêu biểu:

+ Hàm dựng: Tạo dữ liệu cho lần chơi mới.

+ ShowLogo, Win, GameOver, ScoreBoard, MainMenu, PauseMenu: Các chức năng hiển thị Logo, màn hình thắng cuộc , thua cuộc, bảng điểm, menu chính và pause menu.

+ Show: Chức năng hiển thị giao diện game.

+ InputUserName: Chức năng nhập tên ngời chơi.

+ Save, Load: Chức năng lưu, load dữ liệu.

+ Play: Điều khiển các thao tác chính của trò chơi.

## **2.5. Class GameUI**

Chứa các phương thức thuần ảo.

## **2.6. Các hàm đồ hoạ hỗ trợ**

Các hàm đồ hoạ nằm trong cặp file MyGraphics.h và MyGraphics.cpp (Source có tham khảo từ google):

* **ResizeConsole:** Thay đổi kích thước console.
* **TextColor:** Thay đổi màu chữ.
* **GoToXY:** Đến vị trí bất kỳ trên console.
* **Clear Screen:** Xoá màn hình
* **Print:** In ra chuỗi tại vị trí, màu sắc cho trước.