ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

**---------------------- 🙠🕮🙢 ----------------------**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: CÔNG NGHỆ JAVA**

**Giảng viên:**

**Nhóm môn học:** 03

**Nhóm BTL:** 13

**Thành viên:** 221230771 – Phạm Tiến Dũng

*Hà Nội, 20 tháng 03 năm 2024*

Hà Nội năm 2022

**Hà Nội, Tháng 12/2016**

1. **Giới thiệu**

* Sản phẩm: Advanture Time
* Thể loại: Game Puzzle (giải đố)
* Lối chơi:
  + Game có nhiều màn chơi, trong mỗi màn chơi sẽ là một bản đồ với nhân vật nhất định (các nhân vật chỉ có thể di chuyển trên mặt đất)
  + Nhiệm vụ của người chơi là đi tìm những chiếc chìa khóa được bố trí khắp bản đồ, từ những chìa khóa tìm được người chơi sẽ mở được những cánh cửa bị khóa từ đó sẽ tìm được cái cầu thang giúp người chơi di chuyển đến màn tiếp theo
  + Để chiến thắng được trò chơi người chơi cần tìm đường để đi đến hòn đảo chứa rương kho báu

A screenshot of a video game

Description automatically generated

1. **Khảo sát**

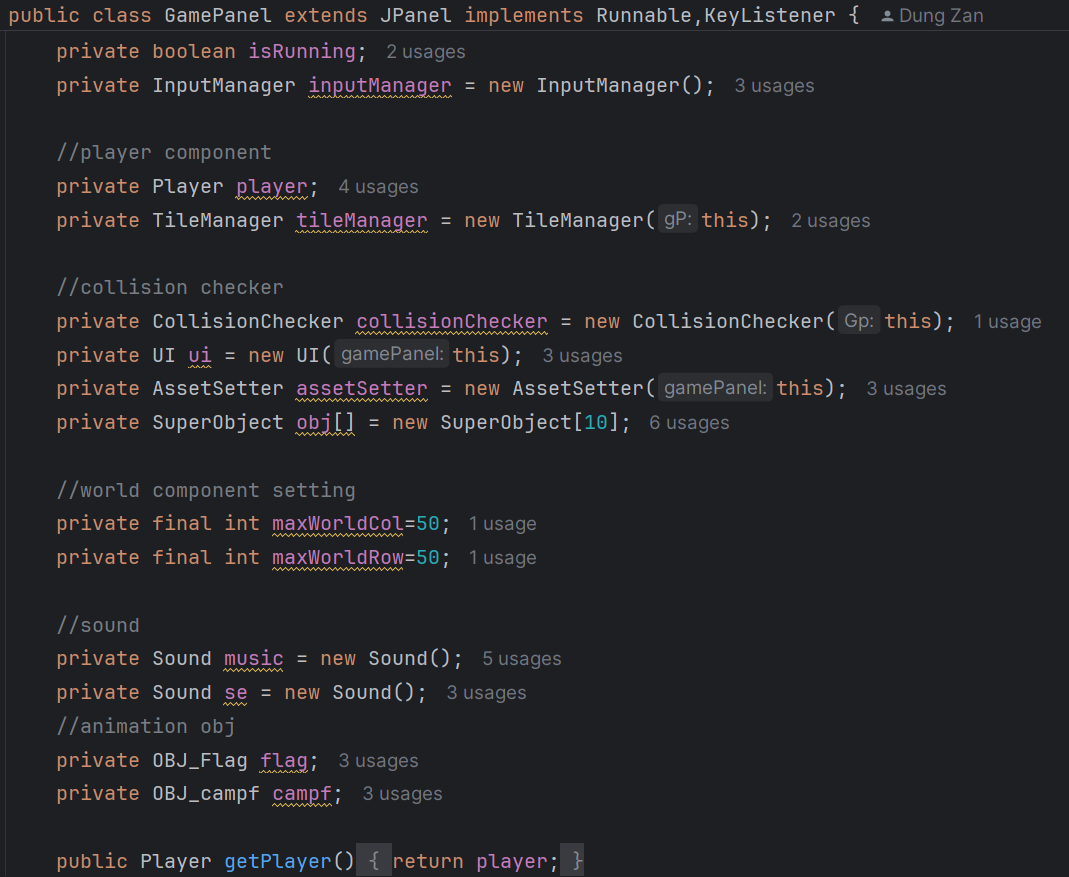
* Qua tìm hiểu ban đầu, nhận thấy game là một sản phẩm không quá khó
* Về thể loại, game puzzle rất dễ dàng tiếp cận với nhiều người, tập trung chủ yếu về tư duy giải đố, có khả năng kích thích sự hứng thú của người chơi.
* Khảo sát trải nghiệm người dùng:
  + Gameplay đơn giản, chỉ cần thao tác với một vài nút mũi tên trên bàn phím.
  + Màu sắc hợp lý (game với tông màu tươi sáng kết hợp với trầm tối).
  + Hình ảnh đơn giản và ưa nhìn
  + Âm thanh nhẹ nhàng, vui tươi.
  + Game đi từ dễ tới khó, chơi qua những màn khó thấy rất thú vị.

1. **Các module chính**
2. manager

A screenshot of a computer program

Description automatically generated

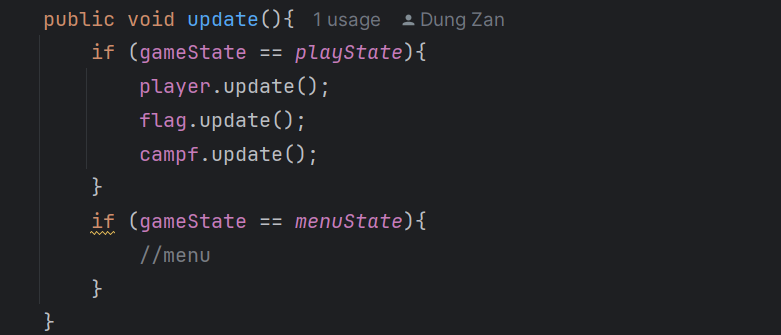
* + **GameFrame**: Chứa các thuộc tính của cửa sổ như width, height, tile size và chạy chương trình
  + **GamePanel**: được implements Runnable và KeyListener
* Class GamePanel thì quản lý và thực hiện việc update những đối tượng chính của game, chứa các thuộc tính quan trọng như CollisionChecker, InputManager, Sound, UI các thuộc tính này thực hiện những chức năng chính của game như lấy sự kiện từ bàn phím, kiểm tra va chạm, vẽ giao diện và điều khiển âm thanh
* Hỗ trợ 1 số chức năng: kiểm tra xem có phải vừa mới ấn nút, vừa mới thả nút, hay đang nhấn giữ nút nào.



* Hàm run() của class GamePanel sẽ là gameloop, nó tính toán deltaTime theo FPS mình target, update(), repaint() mọi thứ theo FPS đó.

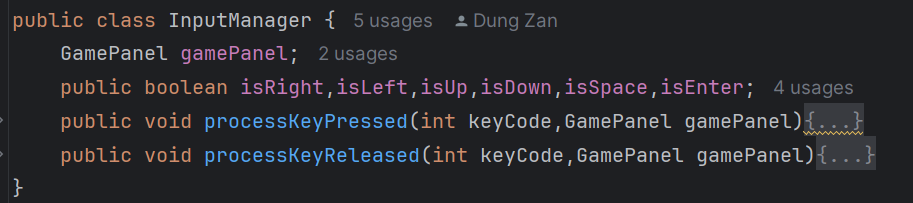


* Update() của Game sẽ là nơi đầu tiên được xử lý (nguồn), sau đó đi update tất cả mọi thứ nhỏ nhỏ hơn nó.

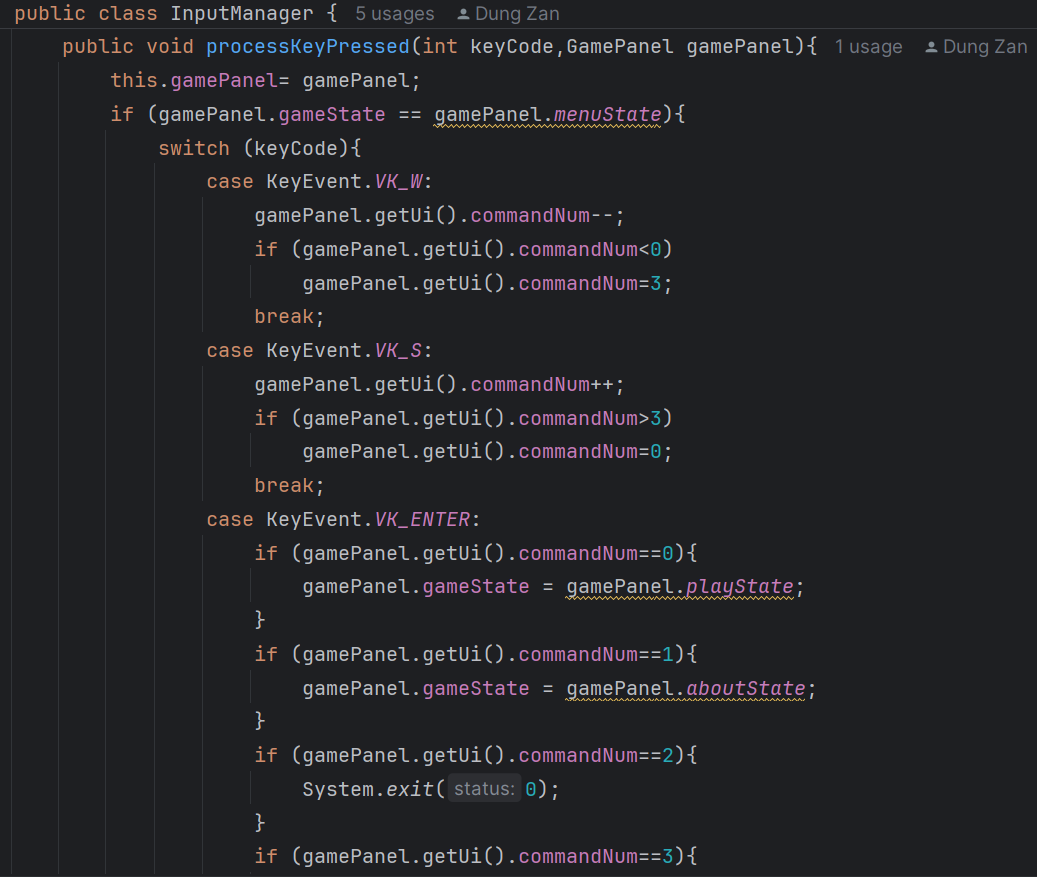




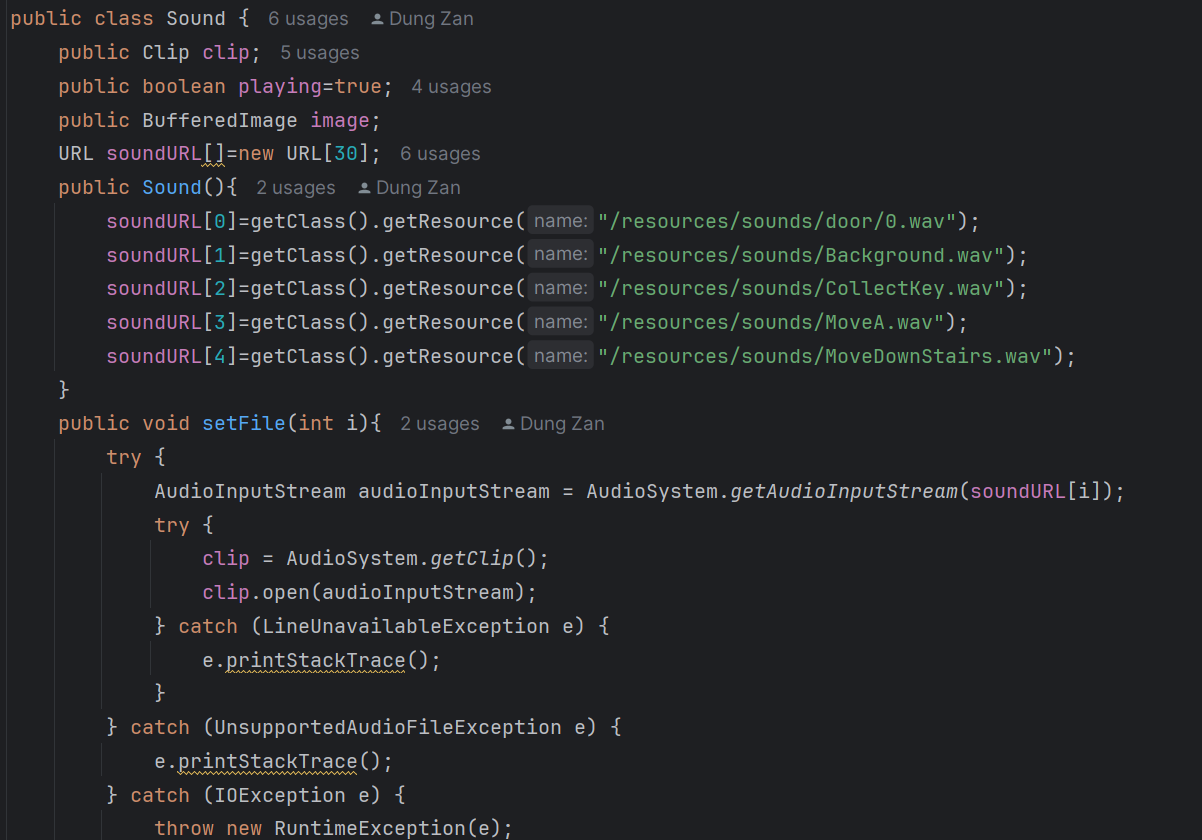
1. **input**
   * **InputManager**: Gồm các thuộc tính dùng để xử lý sự kiện bàn phím



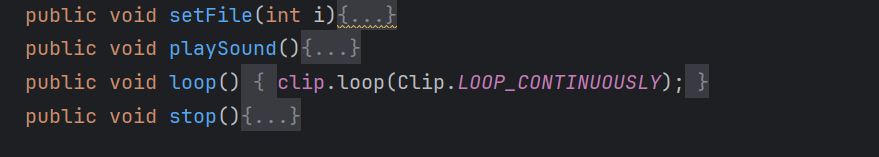
* isRight dùng để check xem Player có di chuyển sang phải hay không và tương tự những thuộc tính khác
* Hàm processKeyPressed xử lý sự kiện nhấn từ bàn phím lấy vào một KeyCode và từ đó làm thay đổi tọa độ nhân vật, chuyển đổi giữa các state game và play Sound



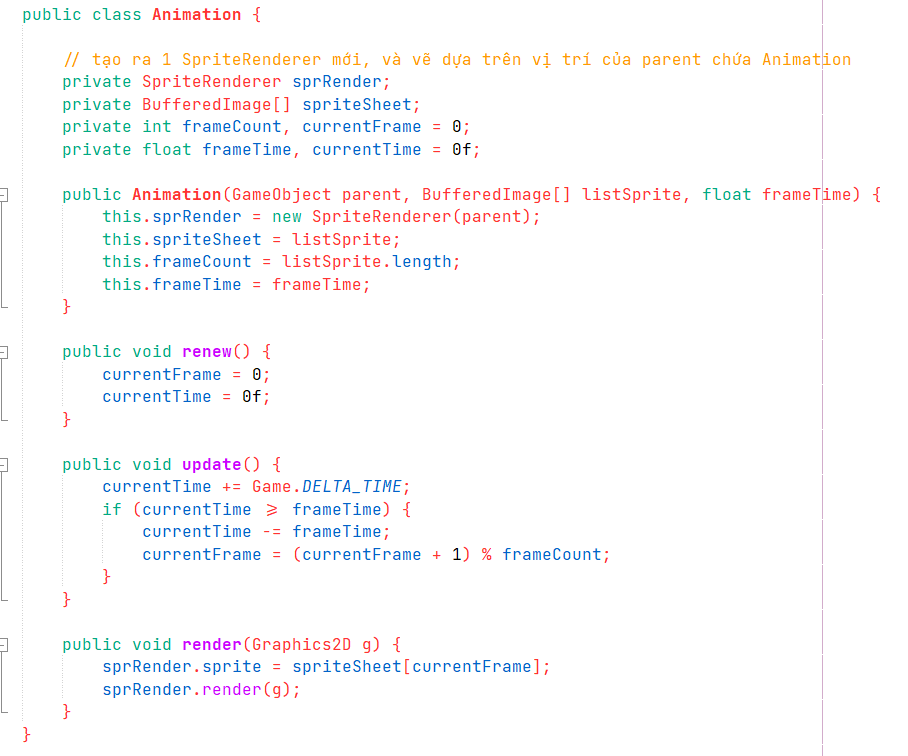
1. **Sound**
   * **Sound:** điều khiển các thuộc tính của âm thanh như pause, play và lấy nguồn âm thanh từ thư mục resources



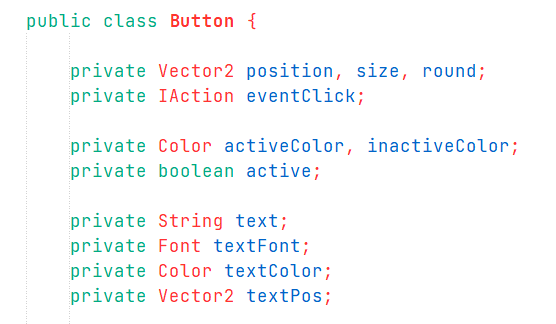
- Gồm các phương thức như sau:



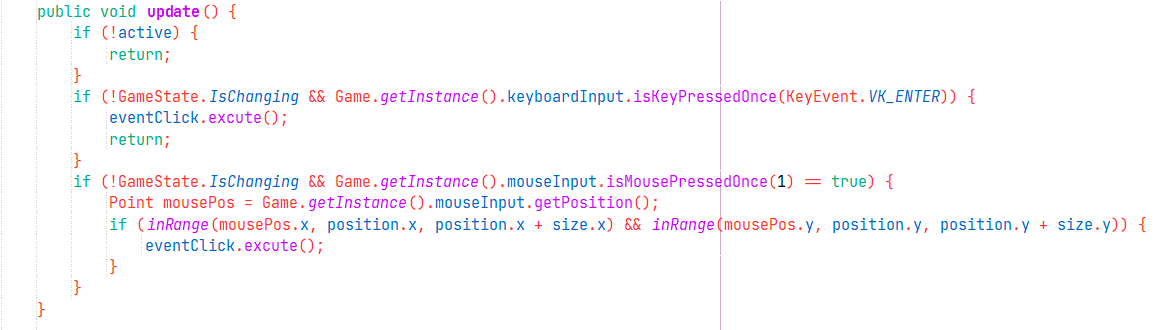
* + **Animation**: sẽ gồm 1 danh sách các frame, mỗi frame cho hiển thị trong khoảng thời gian nhỏ nhỏ (vd tầm 0.1s) , như vậy sau mỗi 0.1s nó sẽ chuyển sang vẽ frame tiếp theo, cho tới khi vẽ xong frame cuối cùng thì chuyển về vẽ frame đầu tiên.



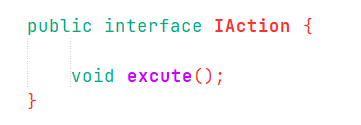
* + **Button**: đây là class nhóm em tự code lại (do khi dùng JButton của java thì không thấy hiển thị trên game và chưa khắc phục được)



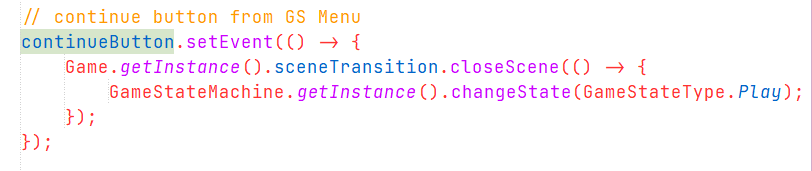
* Trong hàm update() của button: nó nhận và kiểm tra các sự kiện như nhấn phím Enter, click chuột sau đó đi vào thực thi eventClick.



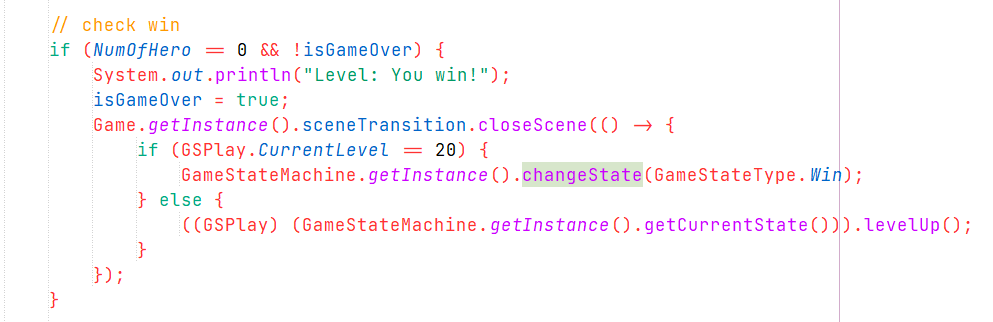
* + IAction: là một interface dùng để thực hiện 1 công việc gì đó khi có sự kiện xảy ra (bấm button, sau khi hoàn thành sceneTransition)



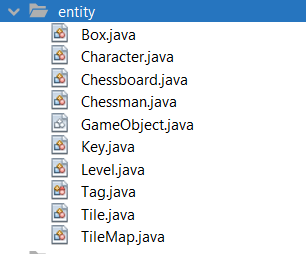
* Button:



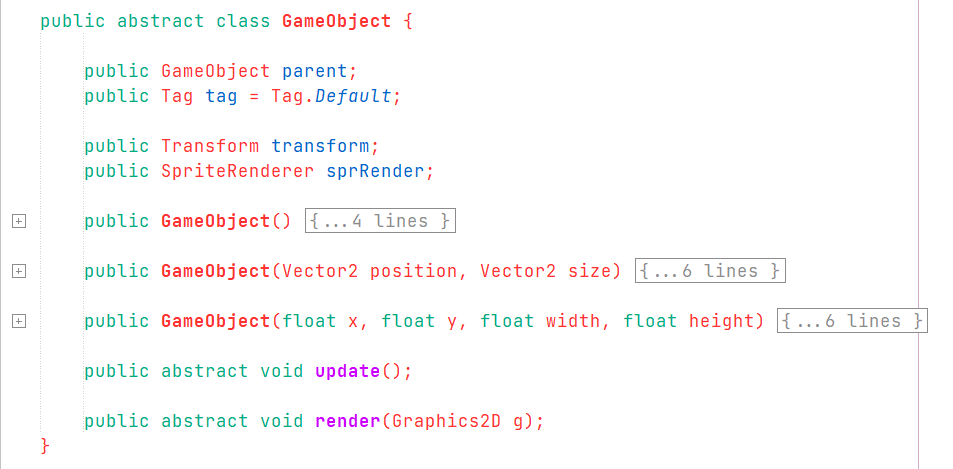
* SceneTransition: làm tối dần màn hình, khi đã tối đen hoàn toàn thì sẽ thực hiện 1 Action gì đó



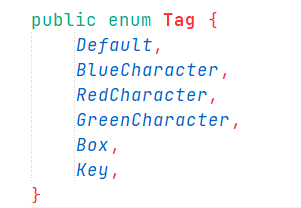
1. **Entity**



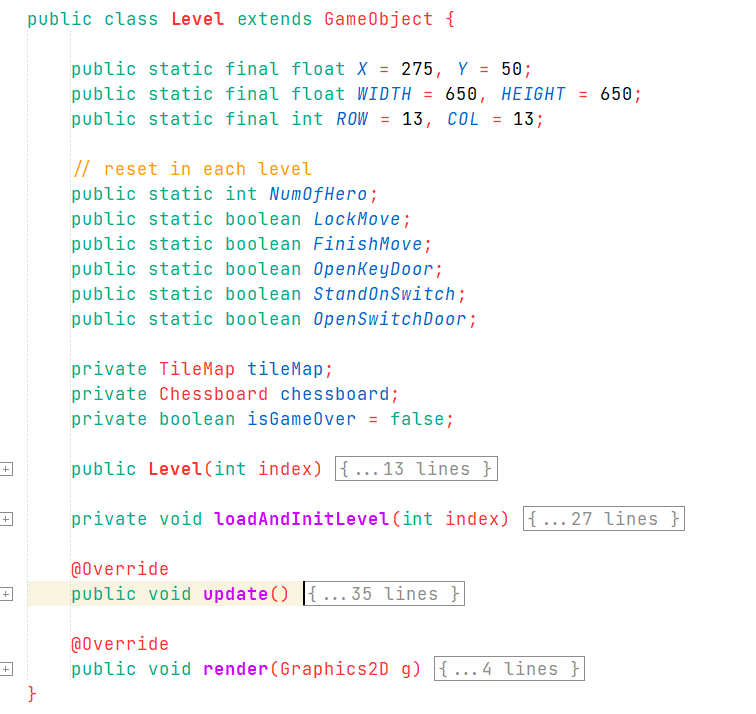
* + **GameObject**: abstract class, là class để cho hầu hết các đối tượng trong gameplay kế thừa và triển khai 2 phương thức chính là update() và render().



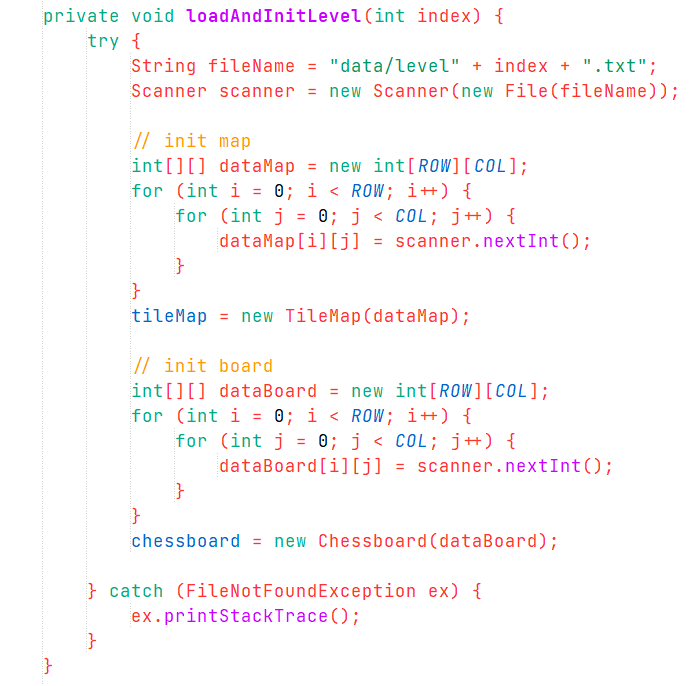
* Object Tag: Default là tag dành cho những gameObject hoặc là không tồn tại trong level đó, hoặc là không thể di chuyển, các object có tag còn lại thì có thể di chuyển được trên ma trận.



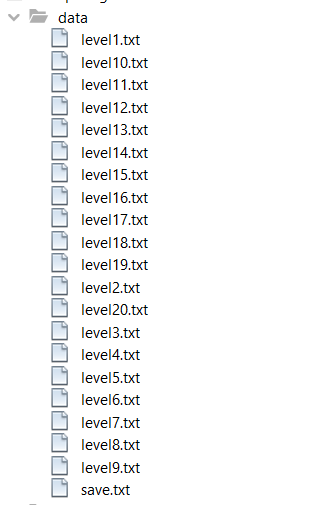
* + **Level**: GamePlay thì sẽ bao gồm level, nó sẽ nhận vào 1 số int là tên level và tiến hành đọc ghi file để lấy dữ liệu, rồi khởi tạo level đó

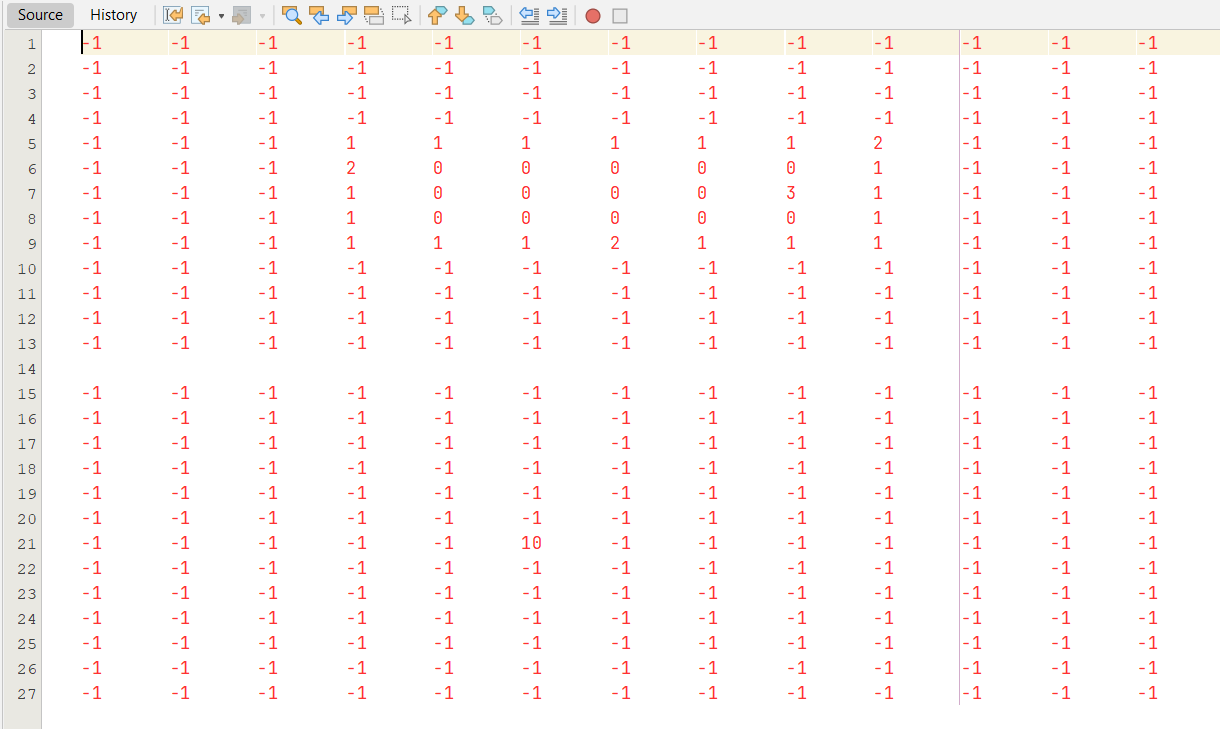


* Load data form file:

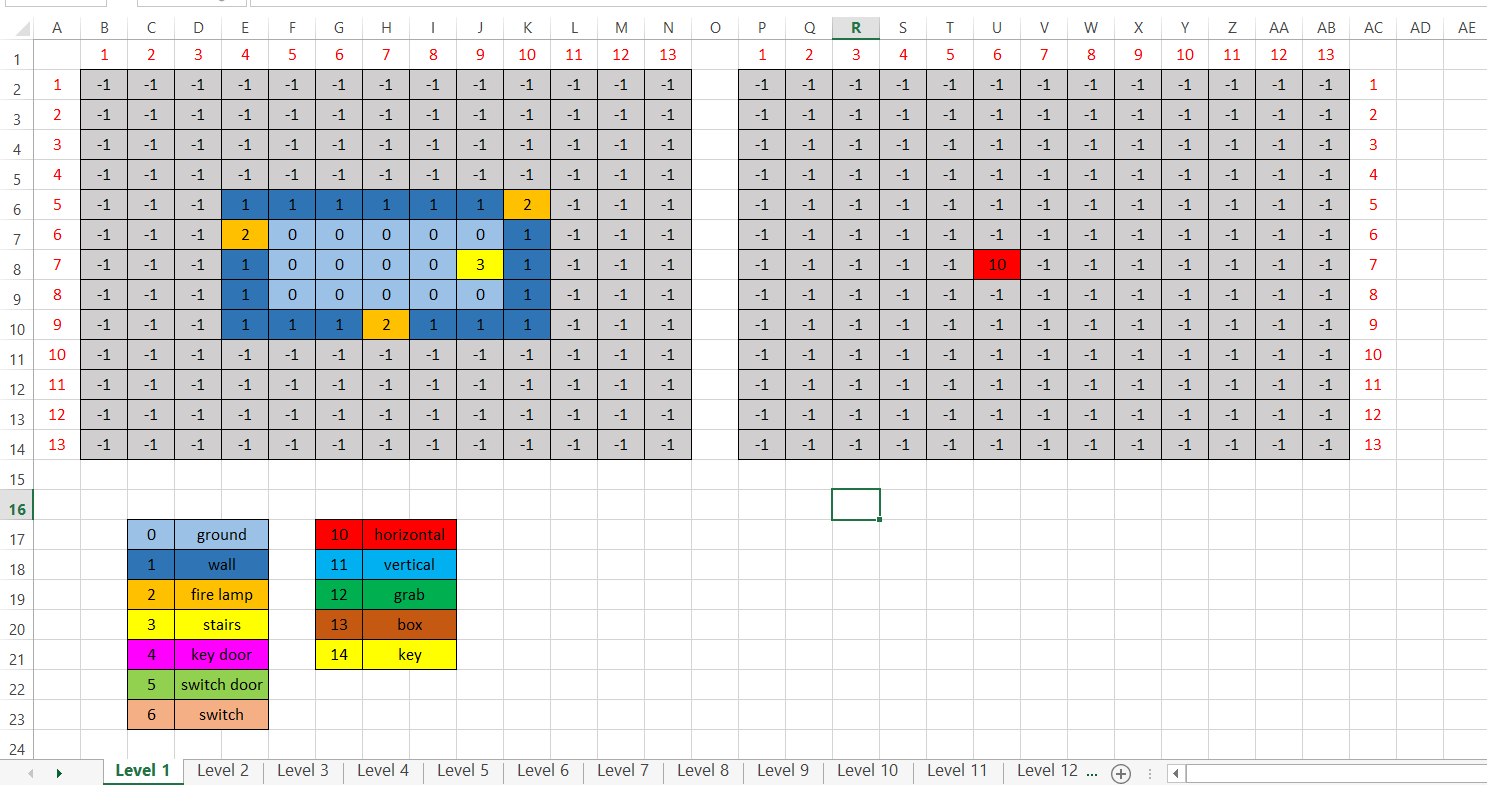


* Data được lưu file text như này:





* Giá trị của các phần tử trong ma trận đã được thiết kế và định nghĩa ở file Level.xlsx như này:

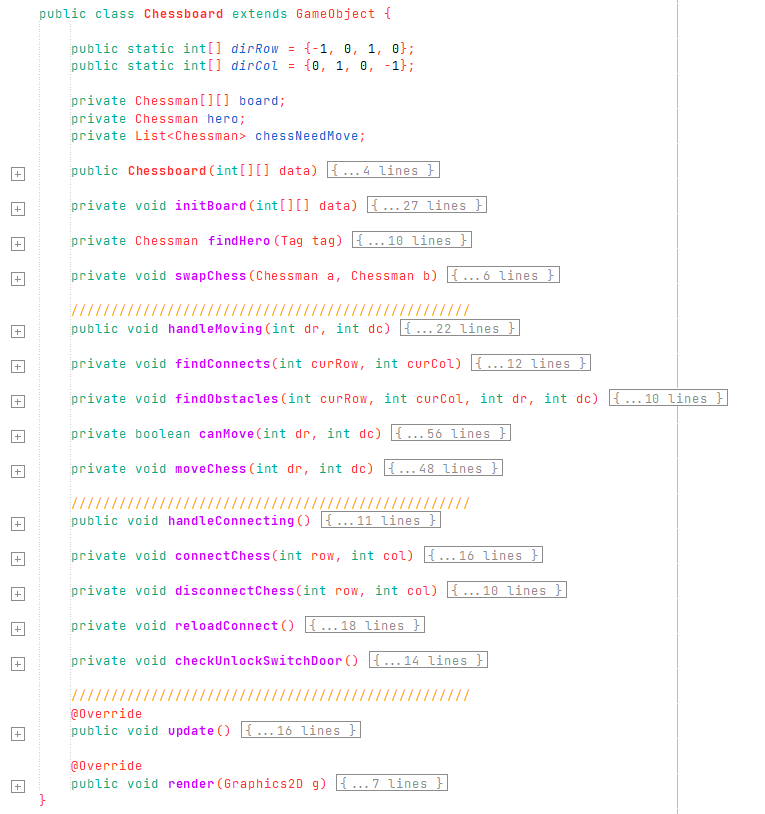


Ma trận bên trái (trong excel) cũng là ma trận thứ nhất (trong Level.txt) thực hiện việc vẽ map: vd -1 là không vẽ gì, 1 là khởi tạo 1 đối tượng wall (tường), 0 là khởi tạo đối tượng ground (đất).

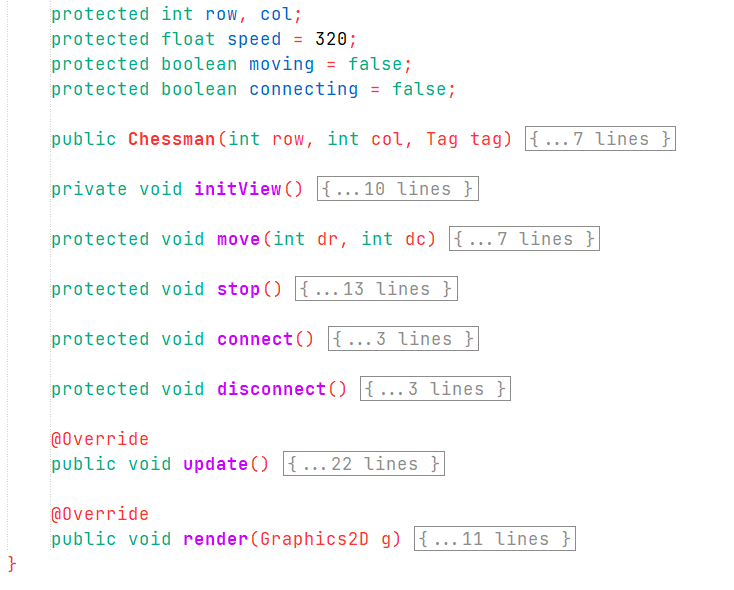
Ma trận bên phải (trong excel) và là ma trận thứ 2 (trong file.txt) thực hiện việc khởi tạo Các Object.

Level sẽ quản lý TileMap (ma trận trái) và Chessboard (ma trận phải), khi render() level, ta sẽ tiến hành vẽ tilemap trước, xong vẽ chessboard trùng lên trên như vậy thì gameobject có thể di chuyển trên mặt đất rồi.

* + **Chessboard**: quản lý vị trí (tính theo ma trận: hàng, cột) của các gameobject



* Khi update(): nó nhận vào các sự kiện bấm bàn phím, sau đó kiểm tra xem có thể di chuyển được không (không bị chặn bởi tường), thực hiện việc di chuyển, liên kết các nhân vật, ...
  + **Chessman**: đơn giản là 1 object nằm trong sự quản lý của chessboard (gameobject có thể di chuyển trong ma trận: nhân vật, hộp gỗ, chìa khoá, ..)



* Nó là class cha để các class con kế thừa, ta sẽ có:





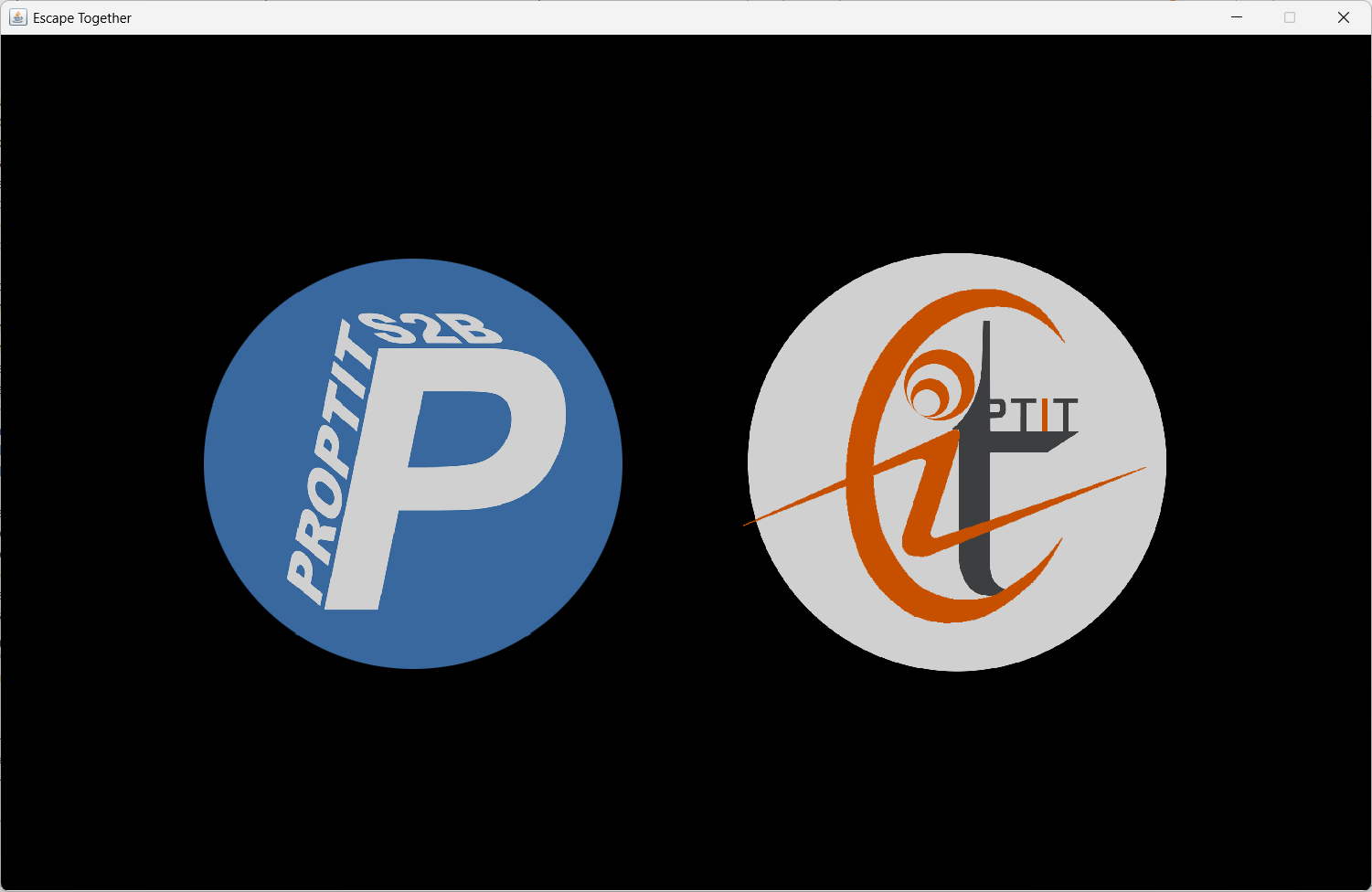


1. **Phương pháp – Ngôn ngữ - Công cụ**

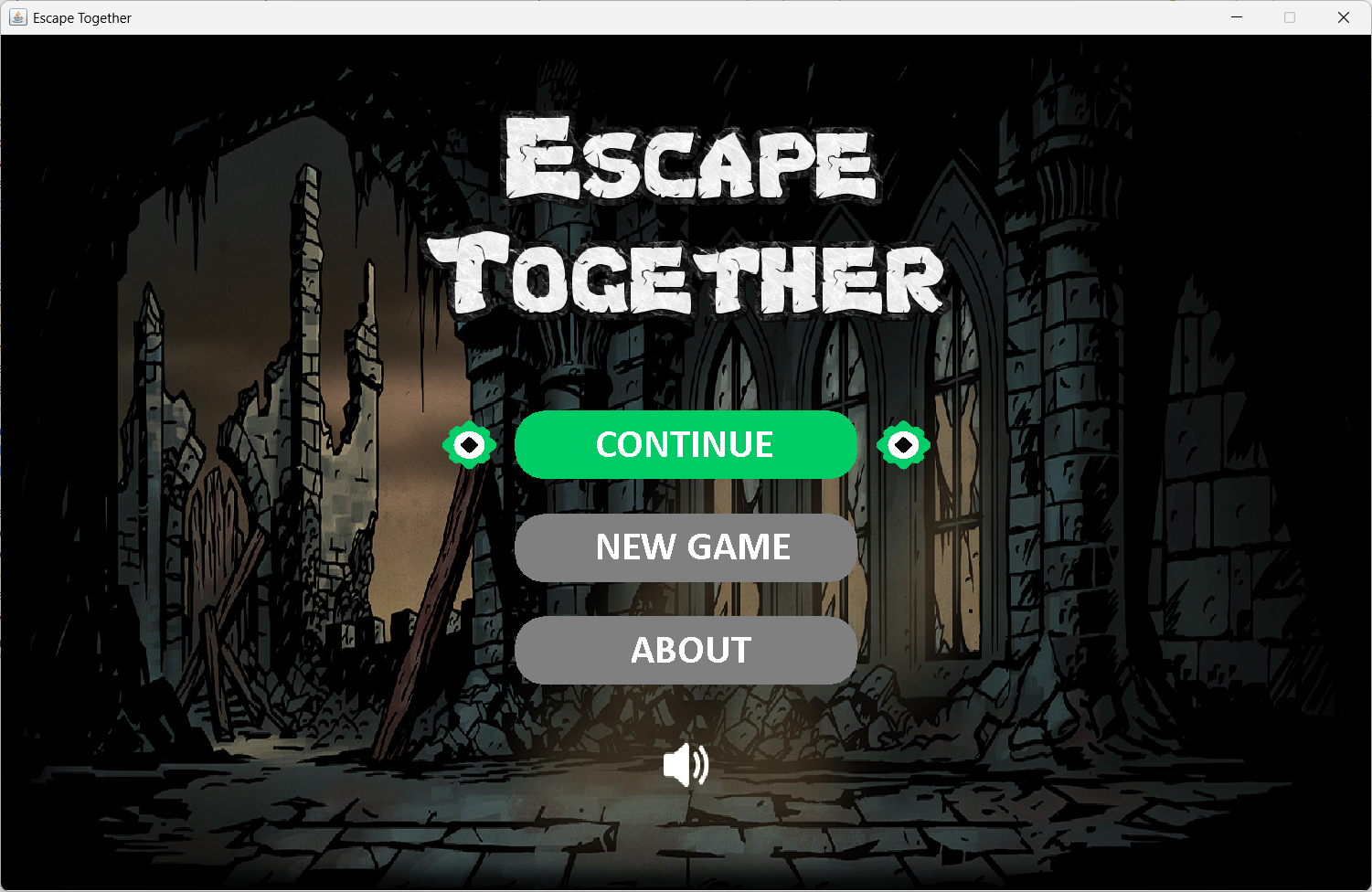
* Phương pháp:
  + Lập trình hướng đối tượng oop
  + Design pattern: singleton, factory method, state machine
* Ngôn ngữ: Java
* Công cụ:
  + Apache Netbeans IDE: code
  + Graphics2D cùng các thư viện khác của java.awt
  + Adobe Ilustator: thiết kế asset cho game
  + Photoshop: chỉnh sửa 1 số ảnh

1. **Giới thiệu hình ảnh kết quả**

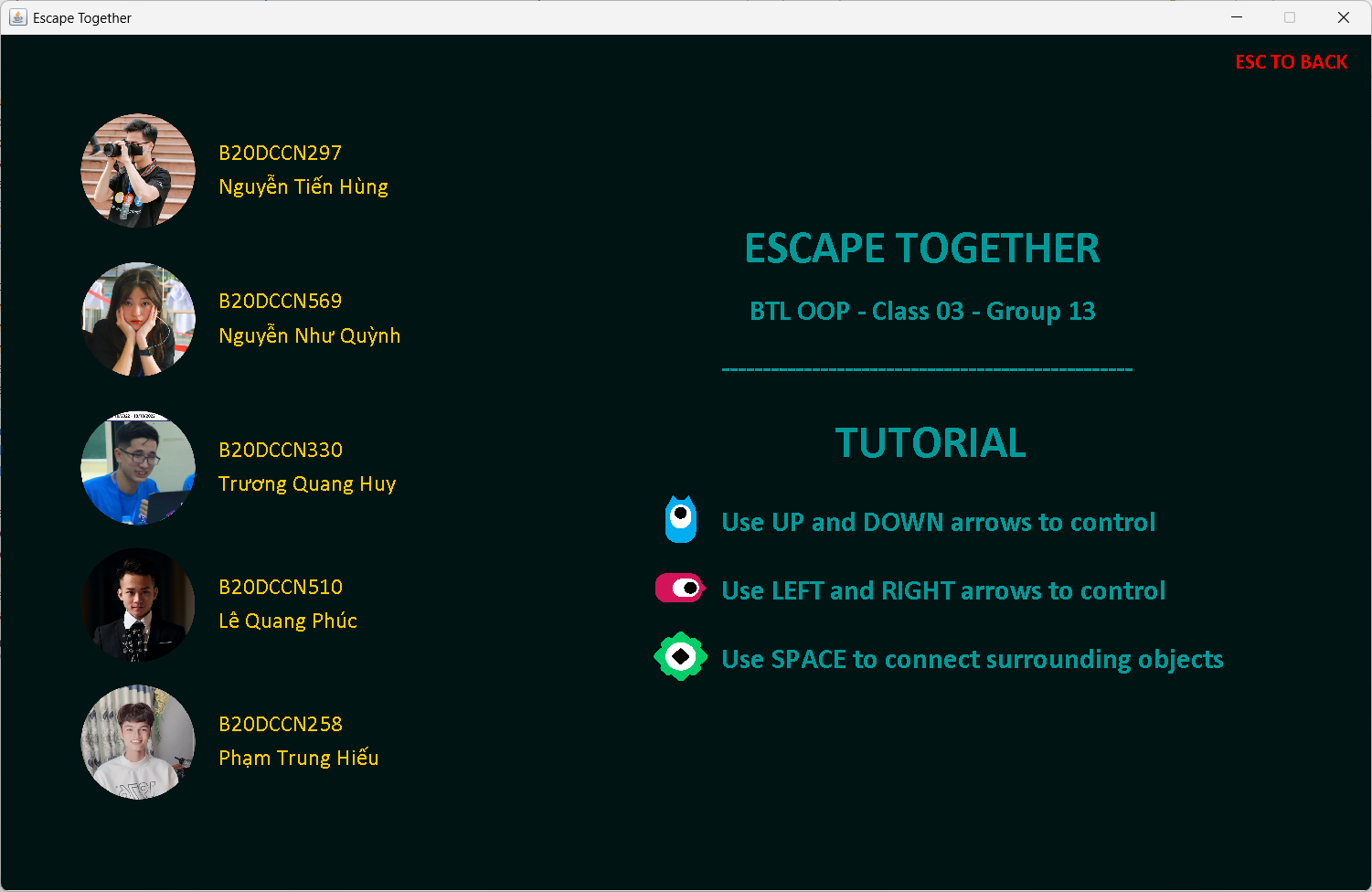
* Intro:



* Menu:

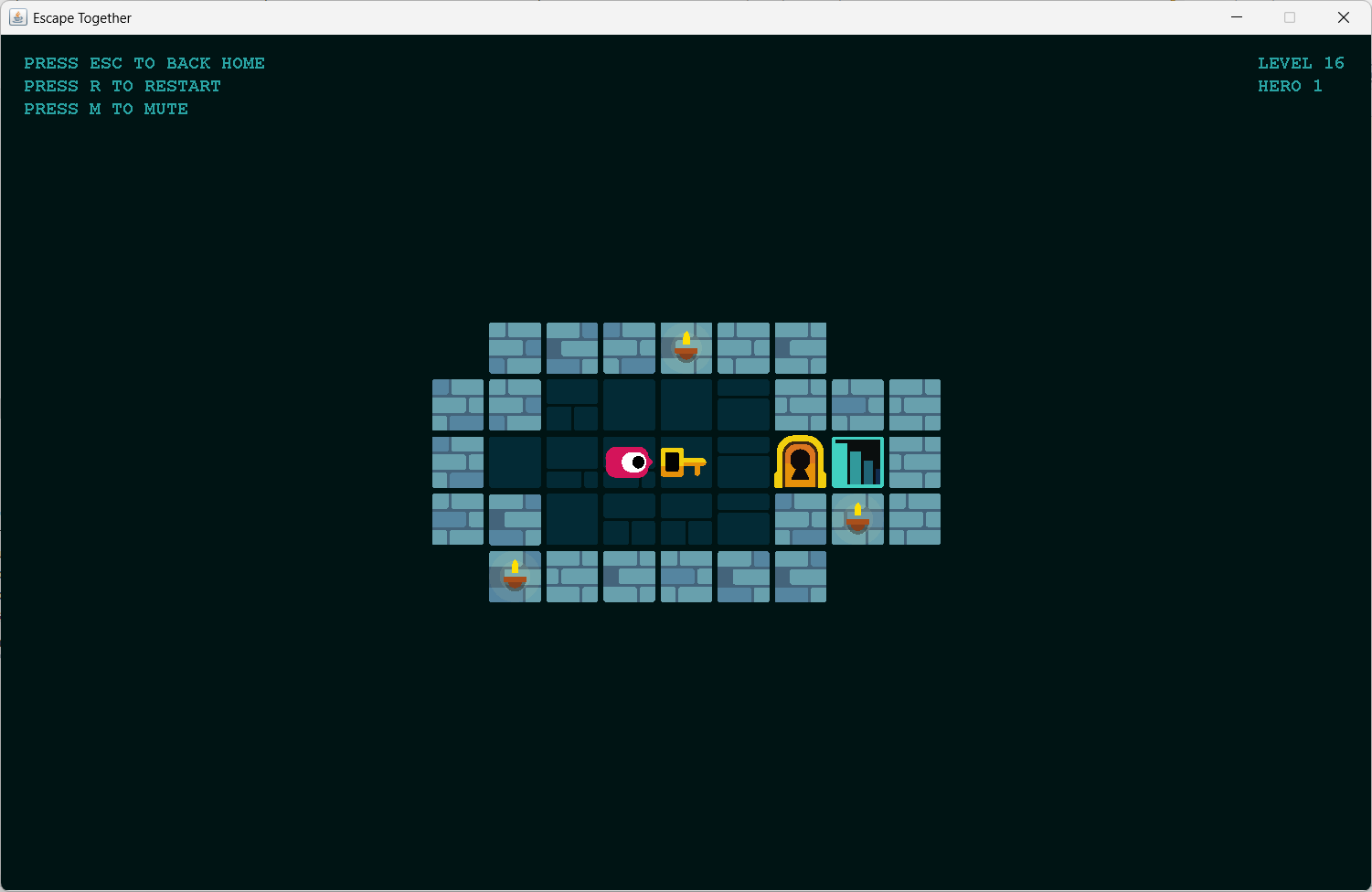


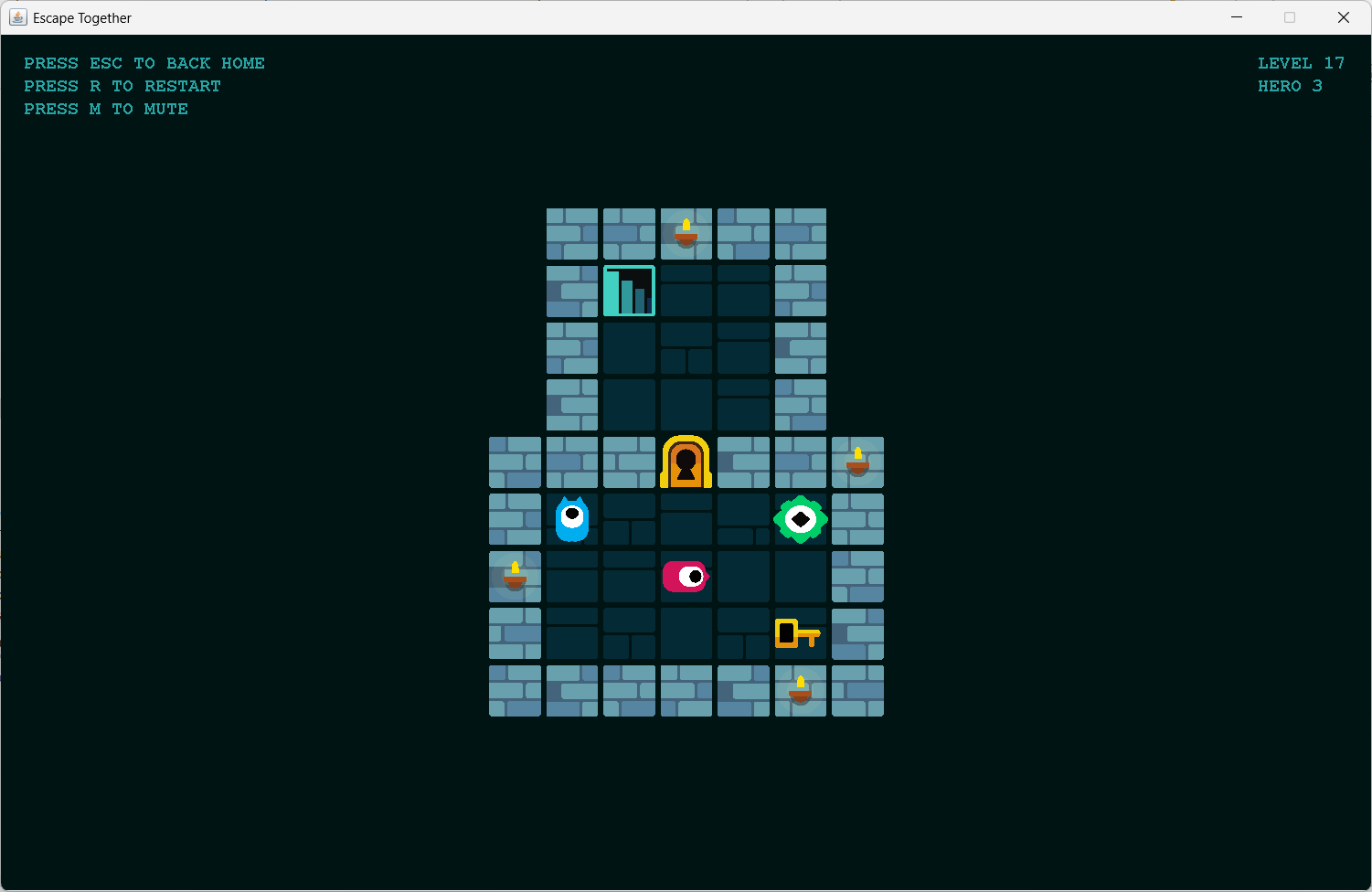
* About:



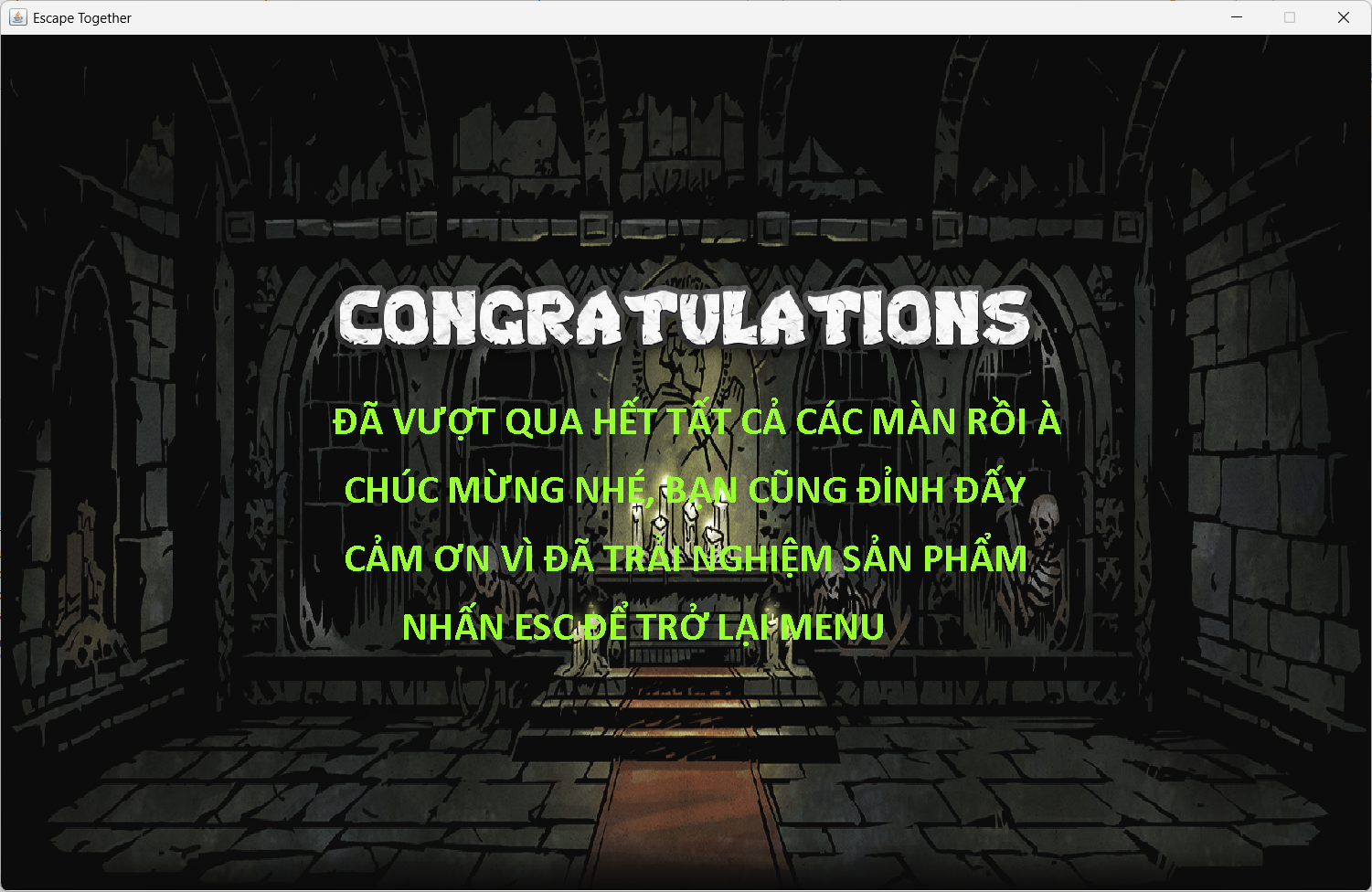
* Play:







* Win:



1. Các tài liệu tham khảo:

* Series lập trình game 2d với java: <https://youtu.be/om59cwR7psI>
* MouseInput pooling system: <https://youtu.be/_zNh-BJPrb4>
* Delta time handling: <https://youtu.be/p7X64g6cOgQ>
* Github tham khảo: <https://github.com/NgTienHungg/SimpleMaze>
* Github tham khảo: <https://github.com/Dynamicslvl/Beat-em-up>