

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công nghệ thông tin và Truyền thông**



## **Báo cáo môn học: Project 2**

**Tên đề tài: Game Tetris**

**Sinh viên thực hiện: Đinh Trọng Đạt – 20173010**

**Giảng viên hướng dẫn: Nguyễn Tuấn Dũng**

Hà Nội, tháng 7 năm 2020

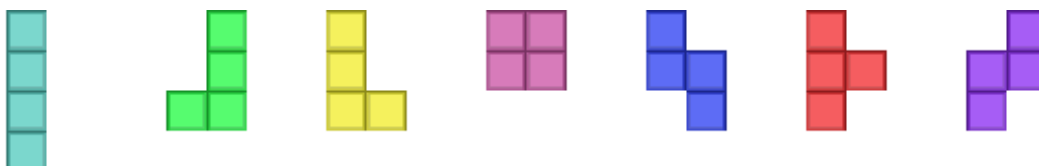
# Mục lục

Mục lục .....	1
I. Ý tưởng.....	2
II. Các bước xây dựng game .....	3
1. Xây dựng các Game Scene .....	3
2. Xây dựng Gameplay .....	4
2.1. Game Board .....	4
2.2. Tetromino .....	5
2.3. Moving .....	6
2.4. Rotating .....	7
2.5. Collision detection .....	8
2.6. Scoring .....	8
III. Tổng kết.....	9

# I. Ý tưởng

Ý tưởng xây dựng tựa game này của em bắt nguồn từ tựa game xếp hình kinh điển có cùng tên.

Trò chơi có 7 loại khối hình: I, J, L, O, S, T, Z. Mỗi khối hình được tạo thành từ 4 ô vuông nhỏ.



Người chơi sẽ phải tìm cách điều khiển các khối hình đang rơi từ từ xuống trong màn hình có kích thước 21 hàng x 10 cột sao cho các khối hình lấp đầy một hàng ngang và hàng ngang đó sẽ biến mất.

Người chơi có thể điều khiển các khối hình sang trái, sang phải, đi xuống, xoay các góc tương ứng  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ .

Trò chơi sẽ kết thúc khi các khối hình xếp cao quá màn hình và không thể rơi xuống được nữa.

Tựa game này sẽ được xây dựng bằng ngôn ngữ C++ trên Microsoft Visual Studio và dựa trên một framework có sẵn do Gameloft xây dựng.

Để hoàn thành được dự án này, cần phải nắm vững được các kiến thức về lập trình C++, lập trình hướng đối tượng và một vài kiến thức khác liên quan đến làm game.

## II. Các bước xây dựng game

### 1. Xây dựng các Game Scene

Việc đầu tiên là phải xây dựng được các màn hình hiển thị trong game, hay còn gọi là Scene. Những Scene cơ bản bao gồm Menu, Gameplay và Setting, trong tựa game này em có xây dựng thêm các Scene khác như Intro, PauseMenu và GameOver.

Khi bắt đầu mở game, Scene Intro sẽ được hiển thị đầu tiên. Sau khoảng vài giây thì sẽ chuyển sang Scene Menu, tại đây người chơi có thể lựa chọn chuyển sang Scene Gameplay để vào chơi game, chuyển sang Scene Setting để thay đổi các cài đặt hoặc thoát game.

Trong Scene Gameplay, ngoài việc chơi game thì người chơi cũng có các nút để lựa chọn chuyển về Scene Menu, Scene Setting hoặc tạm dừng game để chuyển sang Scene PauseMenu. Khi game kết thúc sẽ chuyển sang Scene GameOver.

Trong Scene Setting, người chơi có thể lựa chọn bật hoặc tắt nhạc nền và nhạc hiệu ứng.

Trong Scene PauseMenu, người chơi có thể tiếp tục game hoặc quay trở lại Menu chính.

Trong Scene GameOver, người chơi có thể xem số điểm đã đạt được và lựa chọn chơi game lại hoặc quay về Menu chính.

Các cách thức để tạo ra 1 Scene và chuyển đổi giữa các Scene sẽ do framework thực hiện và framework cũng cung cấp đầy đủ các phương thức để thực hiện những việc này nên em sẽ không nêu chi tiết.

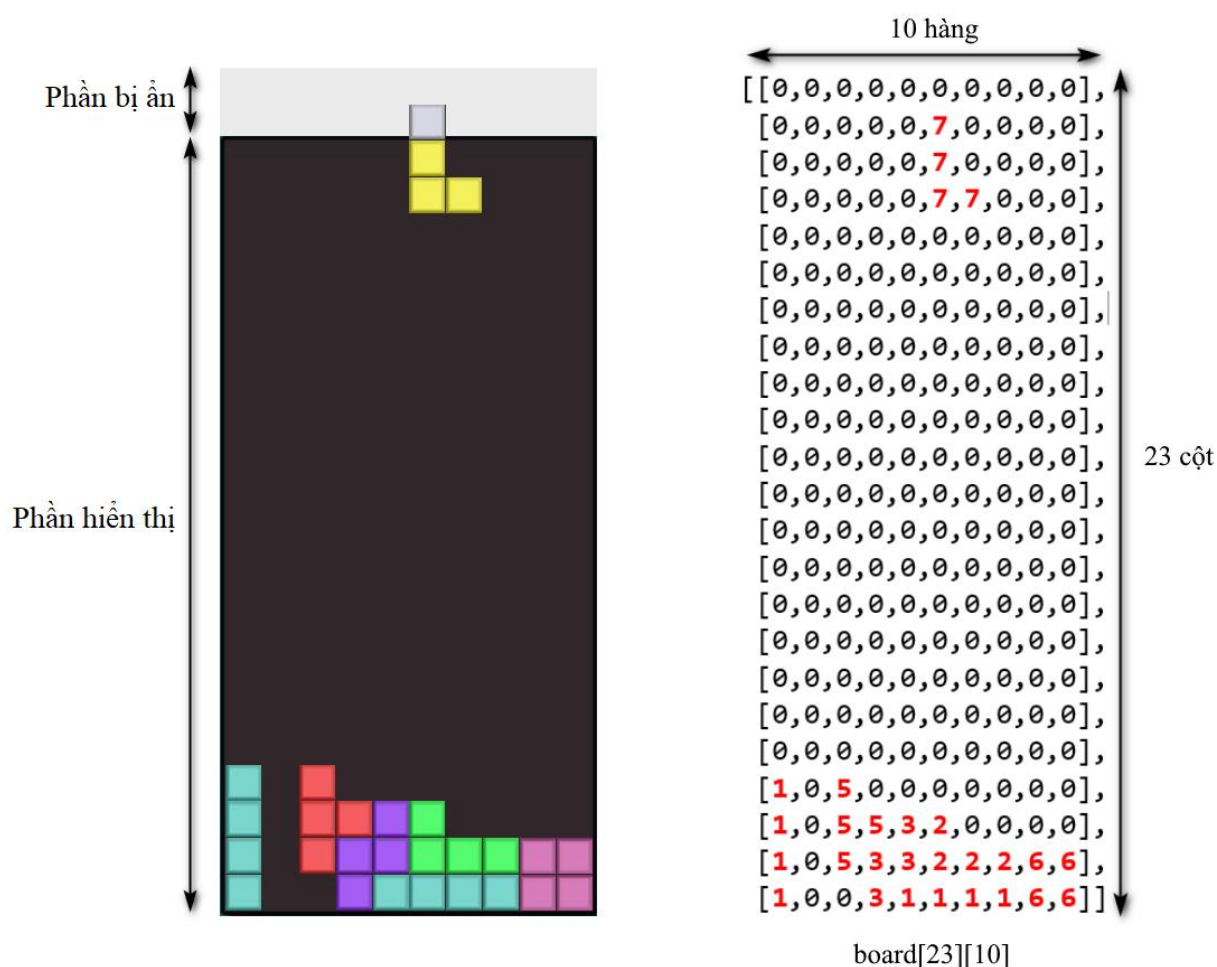
## 2. Xây dựng Gameplay

### 2.1. Game Board

Đây chính là nơi người chơi sẽ chơi game, tại màn hình Gameplay, người chơi sẽ thấy game board có kích thước 21 hàng x 10 cột, nhưng thực kích thước thực tế của game board sẽ là 23 hàng x 10 cột, 2 hàng trên cùng là 2 hàng sẽ bị ẩn khỏi giao diện game.

Chia game board thành 2 phần như thế sẽ khiến các khối hình được tạo ra ở trên cùng rồi rơi dần xuống làm người chơi có cảm giác các khối hình rơi từ bên ngoài màn hình vào.

Để biểu diễn và lưu trữ phần game board này thì em sử dụng 1 mảng 2 chiều có kích thước 23x10 tương ứng.



Những ô có giá trị 0 là những ô chưa có khối hình, những ô có giá trị từ 1 đến 7 là những ô có các khối hình tương ứng với 7 loại khối hình: I, J, Z, S, T, O, L.

## 2.2. Tetromino

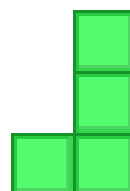
Để biểu diễn các khối hình, hay còn gọi là các Tetromino, em sử dụng các mảng 2 chiều 4x4 chứa các giá trị tương tự như ở game board.

- Khối chữ I:



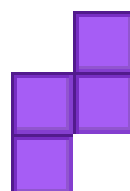
```
[[1,0,0,0],
 [1,0,0,0],
 [1,0,0,0],
 [1,0,0,0]]
```

- Khối chữ J:



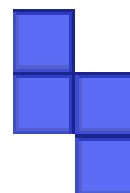
```
[[0,0,2,0],
 [0,0,2,0],
 [0,2,2,0],
 [0,0,0,0]]
```

- Khối chữ Z:



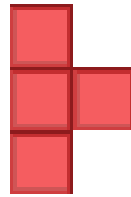
```
[[0,0,3,0],
 [0,3,3,0],
 [0,3,0,0],
 [0,0,0,0]]
```

- Khối chữ S:



```
[[0,4,0,0],
 [0,4,4,0],
 [0,0,4,0],
 [0,0,0,0]]
```

- Khối chữ T:



```
[[0, 5, 0, 0],
 [0, 5, 5, 0],
 [0, 5, 0, 0],
 [0, 0, 0, 0]]
```

- Khối chữ O:



```
[[0, 6, 6, 0],
 [0, 6, 6, 0],
 [0, 0, 0, 0],
 [0, 0, 0, 0]]
```

- Khối chữ L:



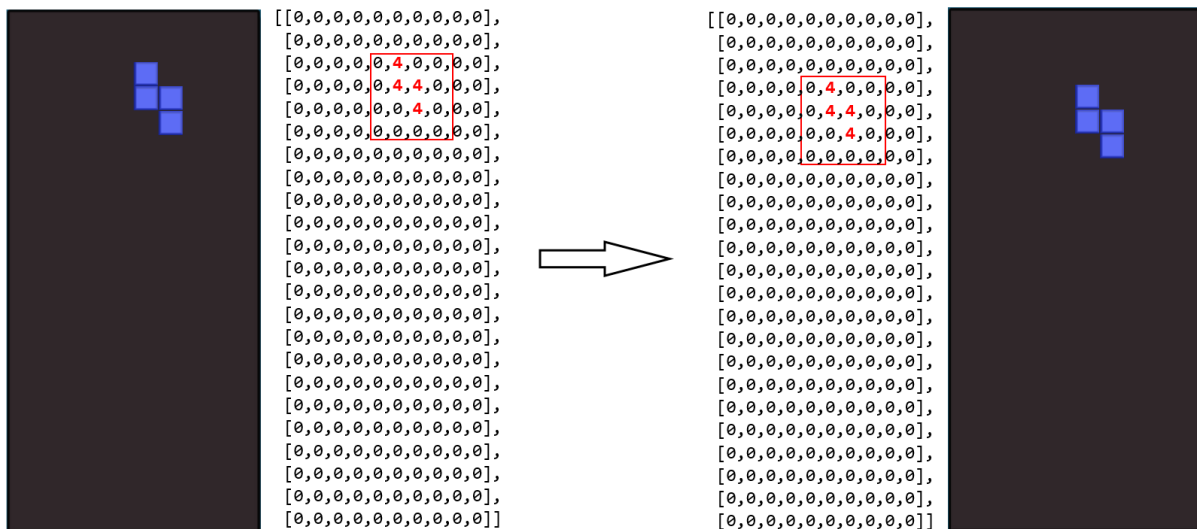
```
[[0, 7, 0, 0],
 [0, 7, 0, 0],
 [0, 7, 7, 0],
 [0, 0, 0, 0]]
```

Mục đích của việc đánh số khác nhau từ 1 đến 7 là để có thể phân biệt giữa các khối hình để có thể hiển thị ra các màu sắc khác nhau.

Để sinh ra một khối hình mới em sử dụng 1 hàm random() để sinh 1 số nguyên ngẫu nhiên từ 1 đến 7 rồi sinh ra khối hình tương ứng tại những hàng trên cùng của game board.

### 2.3. Moving

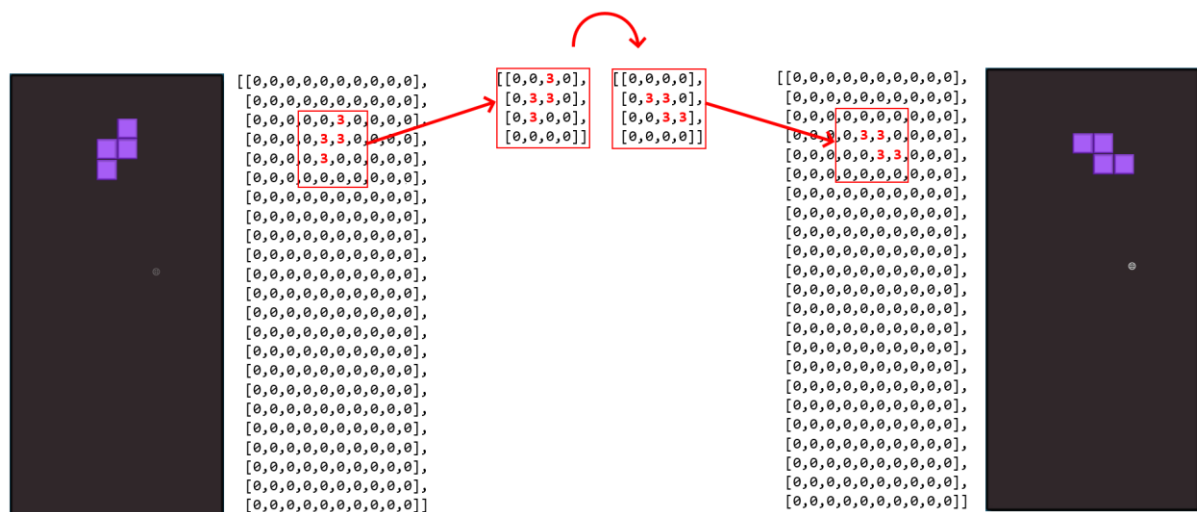
Để có thể di chuyển được khối hình sang phải, trái hay di chuyển xuống dưới thì em thay đổi vị trí của khối hình trong game board dịch chuyển 1 ô theo hướng tương ứng.



Để khiến cho khối hình tự động rơi xuống, sau 1 khoảng thời gian delta time cho khối hình di chuyển xuống 1 ô, delta time có thể thay đổi dựa trên số điểm hiện tại của người chơi để tăng tốc độ rơi của khối hình.

## 2.4. Rotating

Để xoay được khối hình, em dùng thêm 1 mảng 2 chiều block[4][4] để lưu trạng thái của khối hình hiện tại, thực hiện việc xoay khối hình trên mảng phụ này rồi gán lại giá trị vào mảng board[][] tại vị trí hiện tại của khối hình.





## **2.5. Collision detection**

Trước mỗi bước di chuyển hay xoay khối hình, em kiểm tra va chạm bằng cách kiểm tra xem vị trí mà khối hình sẽ di chuyển đến có nằm trong màn hình chơi game hay không và tại đó đã có ô vuông nào tồn tại chưa.

Nếu đã có các ô vuông khác nằm tại đó rồi hoặc ra ngoài màn hình chơi game thì khối hình sẽ không di chuyển hoặc xoay được nữa.

Trong trường hợp khối hình đã đi đến phía dưới cùng của màn hình hoặc rơi xuống chạm phải các khối hình khác nằm sẵn phía dưới thì khối hình sẽ dừng lại và khối hình tiếp theo sẽ rơi xuống.

## **2.6. Scoring**

Người chơi ghi điểm bằng cách xếp các khối hình lấp đầy hàng ngang của màn hình chơi game. Khi đó các hàng ngang đã được lấp đầy sẽ biến mất và những hàng ngang bên trên sẽ rơi xuống thế chỗ những hàng ngang đã biến mất.

VD: Nếu xóa được 2 hàng thì tất cả các hàng phía trên sẽ rơi xuống phía dưới 2 ô.

Người chơi sẽ được thêm điểm thưởng nếu xóa được nhiều hàng ngang cùng 1 lúc (tối đa 4 hàng).

### III. Tổng kết

Về cơ bản, chương trình đã hoàn thành được những yêu cầu cơ bản của 1 game như gameplay, âm nhạc, điểm số,...

Trong tương lai, nếu có cơ hội thì để hoàn thiện chương trình em sẽ phát triển thêm các tính năng:

- Bảng xếp hạng: cho phép người chơi có thể lưu lại điểm số để có thể chia sẻ và thi đấu với bạn bè và những người chơi khác.
- Thay đổi độ lớn của nhạc game (hiện tại mới chỉ có tính năng bật/tắt nhạc).
- Thêm hiệu ứng chuyển động khi di chuyển và xoay khối hình.
- Cải thiện giao diện game (hiện vẫn còn khá sơ sài).

Source code của chương trình:

<https://github.com/trongdat1512/Project2-Tetris>

Các tài liệu tham khảo::

- <https://daynhahoc.com/t/smart-pointer-trong-ngon-ngu-c/75278>
- <https://gpcoder.com/4164-gioi-thieu-design-patterns/>
- Framework:  
<https://drive.google.com/file/d/1OrhkNolLLeYYQ065KCUzYyqglb1a17mi/view>
- <https://topdev.vn/blog/cung-thu-viet-mot-game-xep-hinh-tetris-hoan-chinh-tu-con-so-0-phan-1-giao-dien-va-game-loop/>