

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



# **BÀI TẬP LỚN**

## **HỌC PHẦN: LÝ THUYẾT VÀ NGÔN NGỮ HƯỚNG ĐỐI TƯỢNG**

### **Xây dựng game xếp hình Tetris**

Sinh viên thực hiện:	Dương Minh Phúc	20194648
	Lê Xuân Thịnh	20176880
	Bùi Minh Anh	20194618
	Ngô Thành Đạt	20194499
	Nguyễn Văn Quang	20194655
	Trần Thế Luật	20170217

Nhóm:	5
Mã lớp:	130927

Giáo viên hướng dẫn:	ThS. Nguyễn Mạnh Tuấn
----------------------	-----------------------

*Hà Nội, tháng 7 năm 2022*

## MỤC LỤC

LỜI NÓI ĐẦU .....	3
PHÂN CÔNG THÀNH VIÊN TRONG NHÓM .....	4
CHƯƠNG 1. YÊU CẦU ĐỀ TÀI .....	5
CHƯƠNG 2. BIỂU ĐỒ USE CASE VÀ BIỂU ĐỒ LỚP CHI TIẾT .....	6
2.1. Biểu đồ use case. ....	6
2.2. Biểu đồ lớp chi tiết .....	7
CHƯƠNG 3. THUẬT TOÁN VÀ CÔNG NGHỆ SỬ DỤNG .....	8
3.1. Thuật toán được sử dụng: .....	8
3.2. Công nghệ sử dụng.....	9
CHƯƠNG 4. CHƯƠNG TRÌNH MINH HOẠ .....	10
4.1. Giao diện chương trình khi khởi chạy game .....	10
4.2. Giao diện chức năng tính điểm .....	11
4.3. Giao diện chức năng .....	13
4.4. Giao diện chức năng pause game: .....	14
4.5. Giao diện kết thúc game .....	15
4.6. Kiểm thử chức năng.....	16
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	17
5.1. Ưu điểm và nhược điểm. ....	17
5.2. Hướng phát triển trong tương lai.....	17
TÀI LIỆU THAM KHẢO.....	18

## **LỜI NÓI ĐẦU**

Lý do chọn và tầm quan trọng của đề tài

Tetris là game huyền thoại. Từ những năm 2000, những máy chơi game đen trắng bùng nổ, xuất hiện khắp mọi nơi, hầu như trẻ em thời điểm đấy ai cũng master thể loại Game này. Tetris còn được gọi là trò chơi xếp gạch, nó có thể chơi online trên mạng Internet hoặc offline trên máy tính cá nhân và cả điện thoại, nó không yêu cầu cấu hình quá cao. Tetris có thể xây dựng được bằng nhiều ngôn ngữ khác nhau như Java, C++, ...

Trong trò chơi này, chúng ta có bảy hình dạng khác nhau được gọi là tetrominoes. Hình chữ S, hình chữ Z, hình chữ T, hình chữ L, hình đường thẳng, hình nhân bản L và hình vuông. Mỗi hình dạng này được tạo thành với bốn hình vuông. Các hình dạng đang rơi xuống bảng. Mục tiêu của trò chơi Tetris là di chuyển và xoay các hình dạng sao cho chúng vừa vặn nhất có thể. Nếu chúng ta cố gắng tạo thành một hàng, hàng đó bị phá hủy và chúng ta ghi điểm.

Tuy đây là một game đã xuất hiện từ lâu trên thế giới, nhưng bọn em vẫn lựa chọn xây dựng lại tựa game này, vì nó hoàn toàn phù hợp với lượng kiến thức được học và đảm bảo được các yêu cầu về đề tài Project.

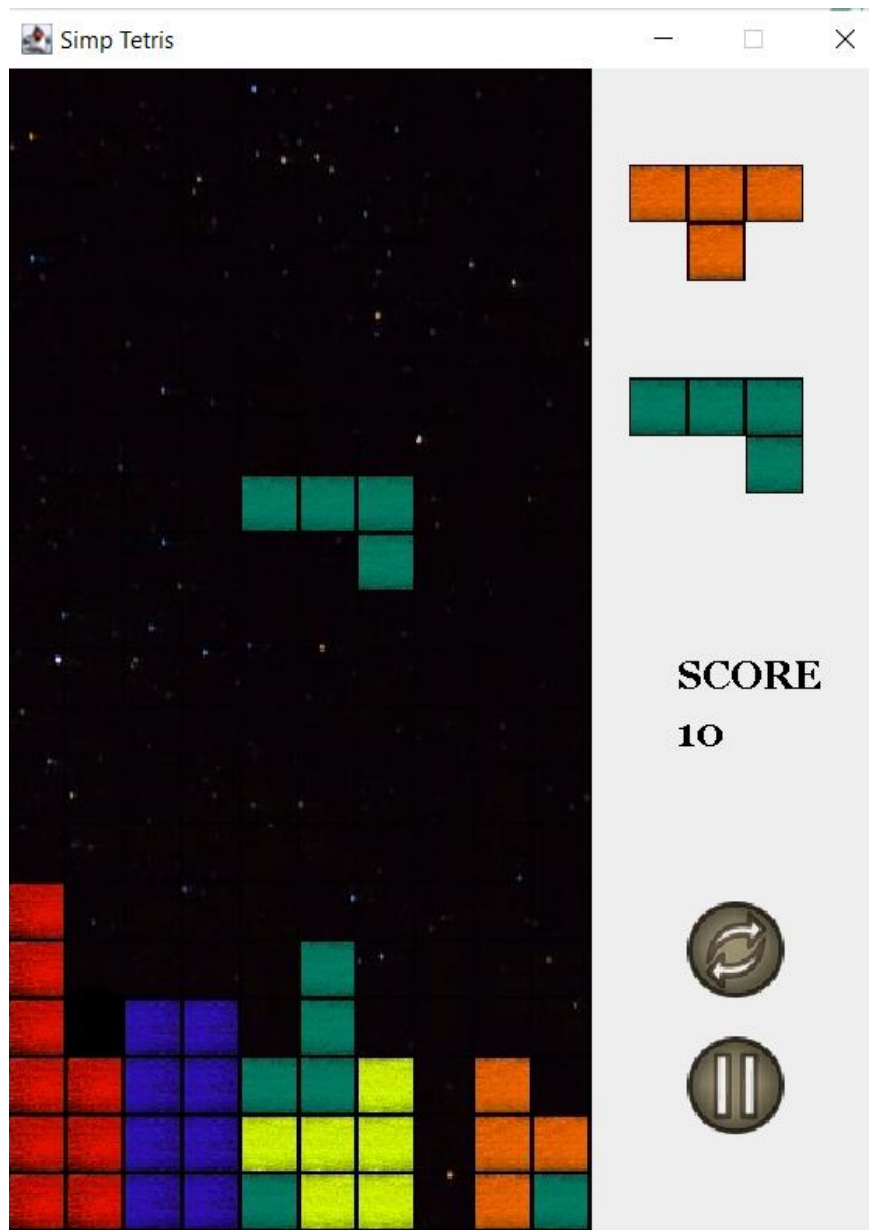
Bởi do mới làm quen với ngôn ngữ Java cũng như việc xử lý trong lập trình hướng đối tượng nên trong chương trình của bọn em không tránh khỏi những khuyết điểm và thiếu sót. Vậy nên nhóm chúng em mong có được những nhận xét, đóng góp, góp ý từ phía thầy để sản phẩm được hoàn thiện hơn.

## **PHÂN CÔNG THÀNH VIÊN TRONG NHÓM**

Họ và Tên	Công việc thực hiện	Đánh giá
Dương Minh Phúc	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Giao diện hiển thị</li><li>- Thuyết trình</li></ul>	Đạt
Bùi Anh Minh	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Score,destroy row,</li><li>- Kiểm thử</li></ul>	Đạt
Lê Xuân Thịnh	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Move block ,rotate block ,</li><li>- Vẽ Use case , Class diagram</li></ul>	Đạt
Ngô Thành Đạt	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Create blocks</li><li>- Silde</li></ul>	Đạt
Trần Thế Luật	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Giao diện hiển thị</li><li>- Báo cáo</li></ul>	Đạt
Nguyễn Văn Quang	<ul style="list-style-type: none"><li>- Lên ý tưởng</li><li>- Báo cáo</li></ul>	Đạt

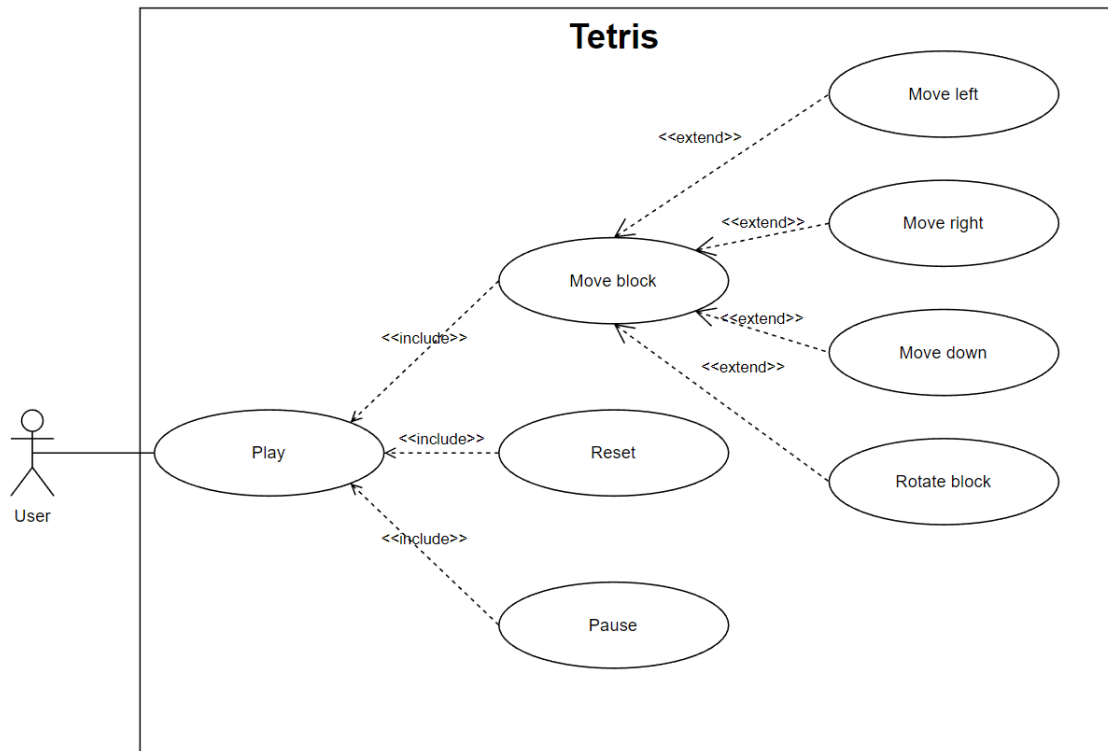
## CHƯƠNG 1. YÊU CẦU ĐỀ TÀI

- Khởi tạo cơ bản về hình dạng ( chữ S , chữ Z , Chữ T , chữ L ,..) , màu sắc của các viên gạch.
- Di chuyển (sang trái, sang phải, xuống dưới) và xoay các hình dạng của viên gạch.
- Khi chúng ta xếp thành 1 hàng , hàng đó bị phá hủy và chúng ta ghi điểm.



## CHƯƠNG 2. BIỂU ĐỒ USE CASE VÀ BIỂU ĐỒ LỚP CHI TIẾT

### 2.1. Biểu đồ use case.



- Danh sách Actor

Tên Actor	Ý nghĩa
User	Người chơi game

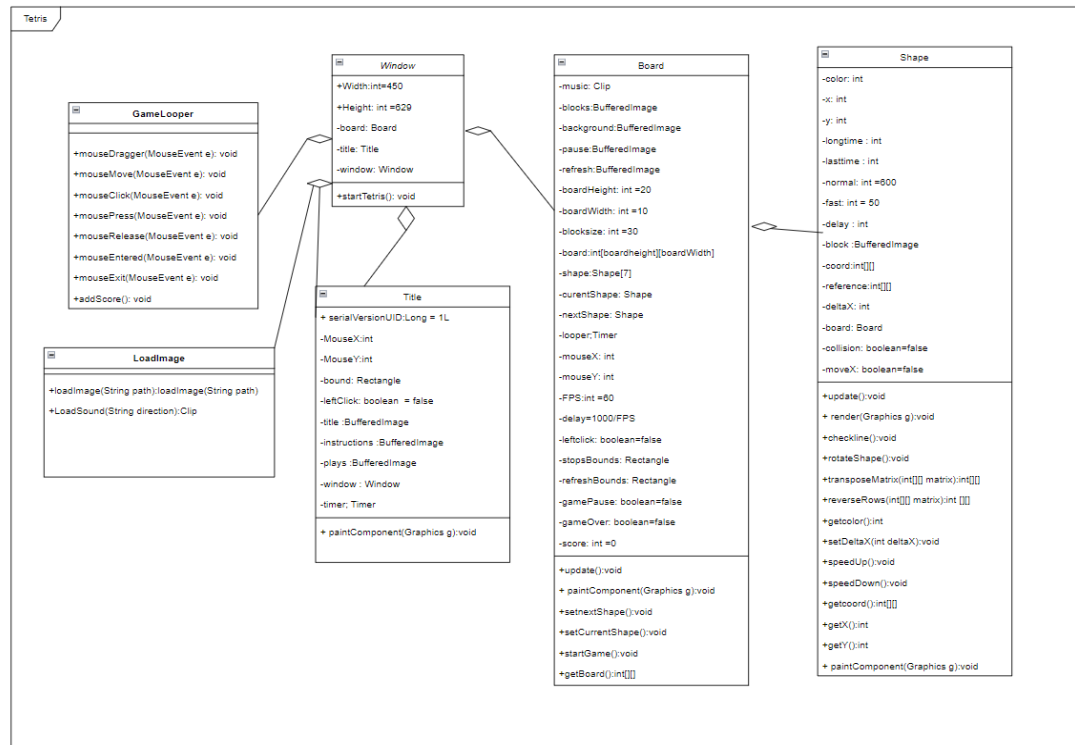
- Danh sách Use case

Tên Use Case	Ý nghĩa
Play	Bắt đầu trò chơi
Pause	Dừng (tiếp tục khi đã dừng) trò chơi
Reset	Chơi mới
Move block	Di chuyển khối

- Đặc tả Case Move block

Tên Use Case	Ý nghĩa
Move left	Di chuyển khối sang trái
Move right	Di chuyển khối sang phải
Move down	Di chuyển khối xuống dưới
Rotate block	Xoay hình dạng của khối

## 2.2. Biểu đồ lớp chi tiết



- Classloadimage: input hình ảnh
- ClassTitle: giao diện vào game
- Class Window: tạo giao diện cửa sổ game
- Class Shape: Tạo hình dạng các khối (lưu dưới dạng ma trận), xoay các khối ...
- Class GameLooper: các thao tác với chuột
- Class Board: load file hình ảnh 7 màu vào và chia thành 7 phần để tạo khối, các tính năng pause, reset, stopgame (game over), override các thao tác với phím

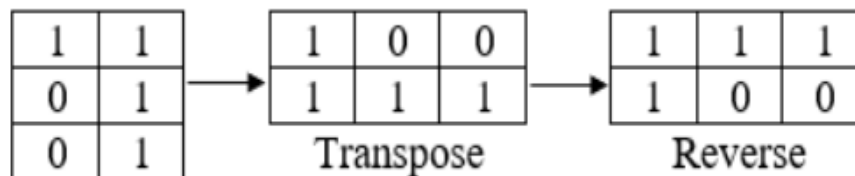
## CHƯƠNG 3. THUẬT TOÁN VÀ CÔNG NGHỆ SỬ DỤNG

### 3.1. Thuật toán được sử dụng:

Xử lý Shape:

Sau khi bắt đầu chơi, giao diện chính và shape xuất hiện, ta sẽ xử lý các sự kiện:

- Các shape sẽ tự động rơi ngẫu nhiên từ trên xuống, việc của người chơi là di chuyển và xoay các shape.
- Các shape có thể di chuyển sang trái hoặc phải bằng các nút trái phải và có thể tăng tốc độ rơi xuống bằng nút mũi tên xuống. Nếu như shape đang ở sát biên thì không thể di chuyển sang bên biên đó nữa và khi đã chạm biên dưới thì dừng lại.
- Tiếp theo là xử lý xoay Shape:



Ta tạo thao tác phím lên dùng để xoay Shape. Như trên, mỗi Shape sẽ được định nghĩa là một ma trận gồm các phần tử 0 và 1. Do đó, Shape sẽ được xoay theo công thức chuyển vị ma trận (Transpose) sau đó ta đảo ngược ma trận này (Reverse).

```
public void rotateShape() {
    int[][] rotatedShape = null;
    rotatedShape = transposeMatrix(coords);
    rotatedShape = reverseRows(rotatedShape);
    if ((x + rotatedShape[0].length > 10) || (y + rotatedShape.length > 20)) {
        return;
    }
    for (int row = 0; row < rotatedShape.length; row++) {
        for (int col = 0; col < rotatedShape[row].length; col++) {
            if (rotatedShape[row][col] != 0) {
                if (board.getBoard()[y + row][x + col] != 0) {
                    return;
                }
            }
        }
    }
    coords = rotatedShape;
}

private int[][] transposeMatrix(int[][] matrix) {
    int[][] temp = new int[matrix[0].length][matrix.length];
    for (int i = 0; i < matrix.length; i++)
        for (int j = 0; j < matrix[0].length; j++)
            temp[j][i] = matrix[i][j];
    return temp;
}

private int[][] reverseRows(int[][] matrix) {
    int middle = matrix.length / 2;
    for (int i = 0; i < middle; i++) {
        int[] temp = matrix[i];
        matrix[i] = matrix[matrix.length - i - 1];
        matrix[matrix.length - i - 1] = temp;
    }
    return matrix;
}
```



- Sau khi đã khởi tạo được shape đầu tiên, việc tiếp theo là ta sẽ tạo ra 1 shape khác bằng hàm random, shape mới sau khi xuất hiện sẽ va chạm với các shape cũ và việc ta cần làm tiếp theo là xử lý va chạm giữa các shape.
- Nếu như các shape va chạm mà shape mới vẫn xoay được thì rất vô lý nên sau khi shape mới va chạm vào shape cũ ta sẽ chặn chức năng xoay của shape mới đi
- Cách ghi điểm: sau khi điều khiển và sắp xếp được các shape, nếu như hàng nào được điền đầy thì ta sẽ cho hàng đó biến mất đi và đẩy các hàng ở trên xuống dưới.

```
private void checkLine(){
    int size = board.getBoard().length - 1;

    for(int i = board.getBoard().length - 1; i > 0; i--){
        int count = 0;
        for(int j = 0; j < board.getBoard()[0].length; j++){
            if(board.getBoard()[i][j] != 0){
                count++;
            }else {
                board.getBoard()[size][j] = board.getBoard()[i][j];
            }
        }
        if(count < board.getBoard()[0].length)
            size--;
    }
    if(size==1) {checkscore=size;}
}
```

- Game kết thúc khi không thể tạo ra shape mới có nghĩa là shape mới đè lên shape cũ thì sẽ thua.

### 3.2. Công nghệ sử dụng

- Ngôn ngữ được sử dụng Java.
- Giao diện được tạo với Java Swing.
- IDE sử dụng Eclipse.

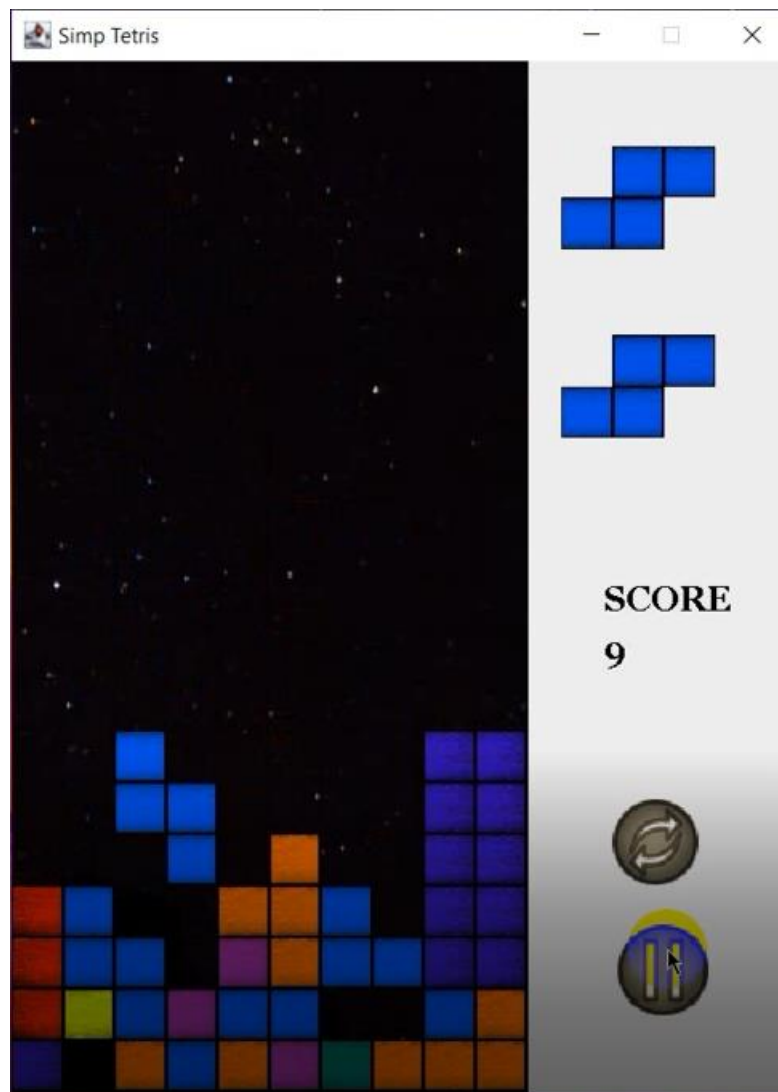
## CHƯƠNG 4. CHƯƠNG TRÌNH MINH HOẠ

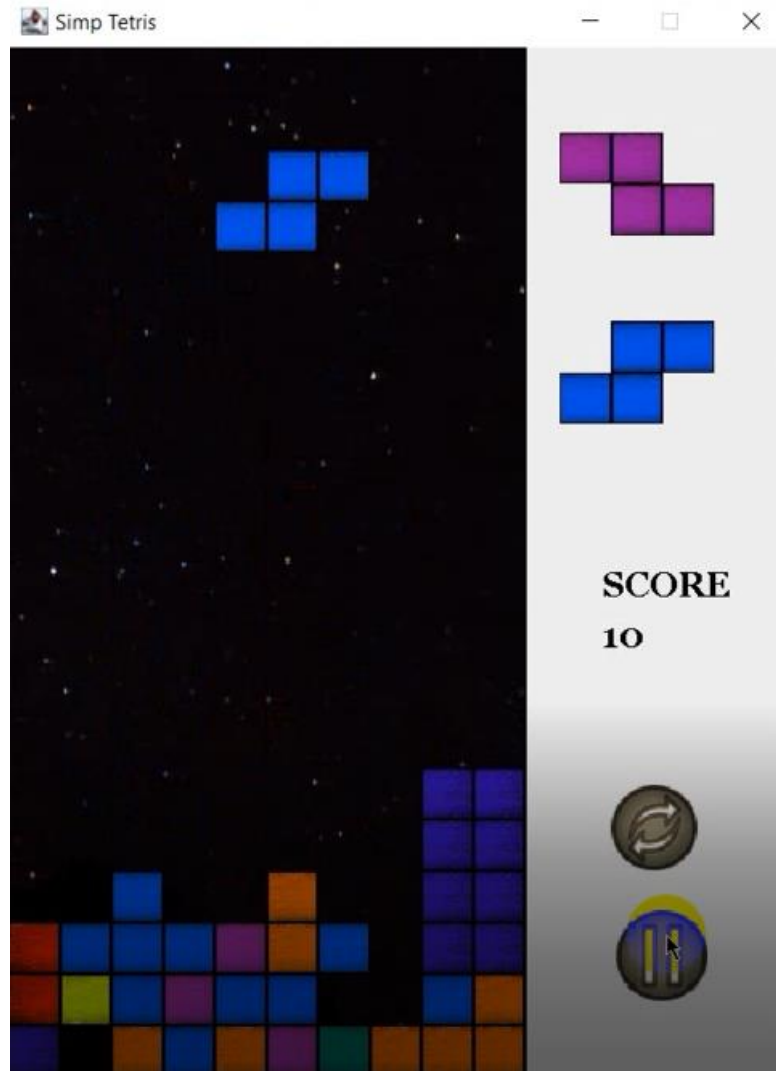
### 4.1. Giao diện chương trình khi khởi chạy game



#### 4.2. Giao diện chức năng tính điểm

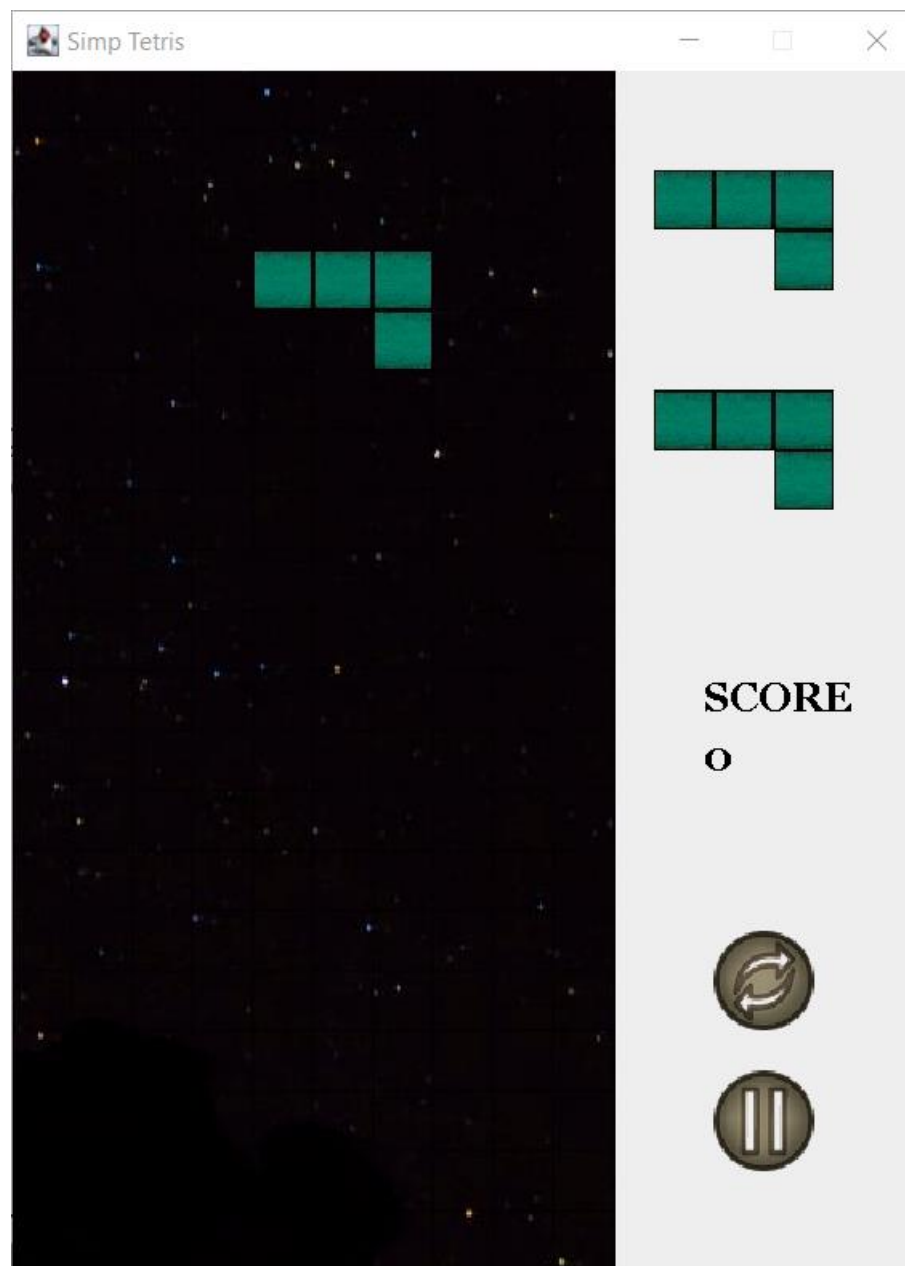
Khi chúng ta di chuyển trái phải , xoay khối và xếp khối gạch thành một hàng , hàng đó sẽ biến mất và điểm cộng thêm 1.



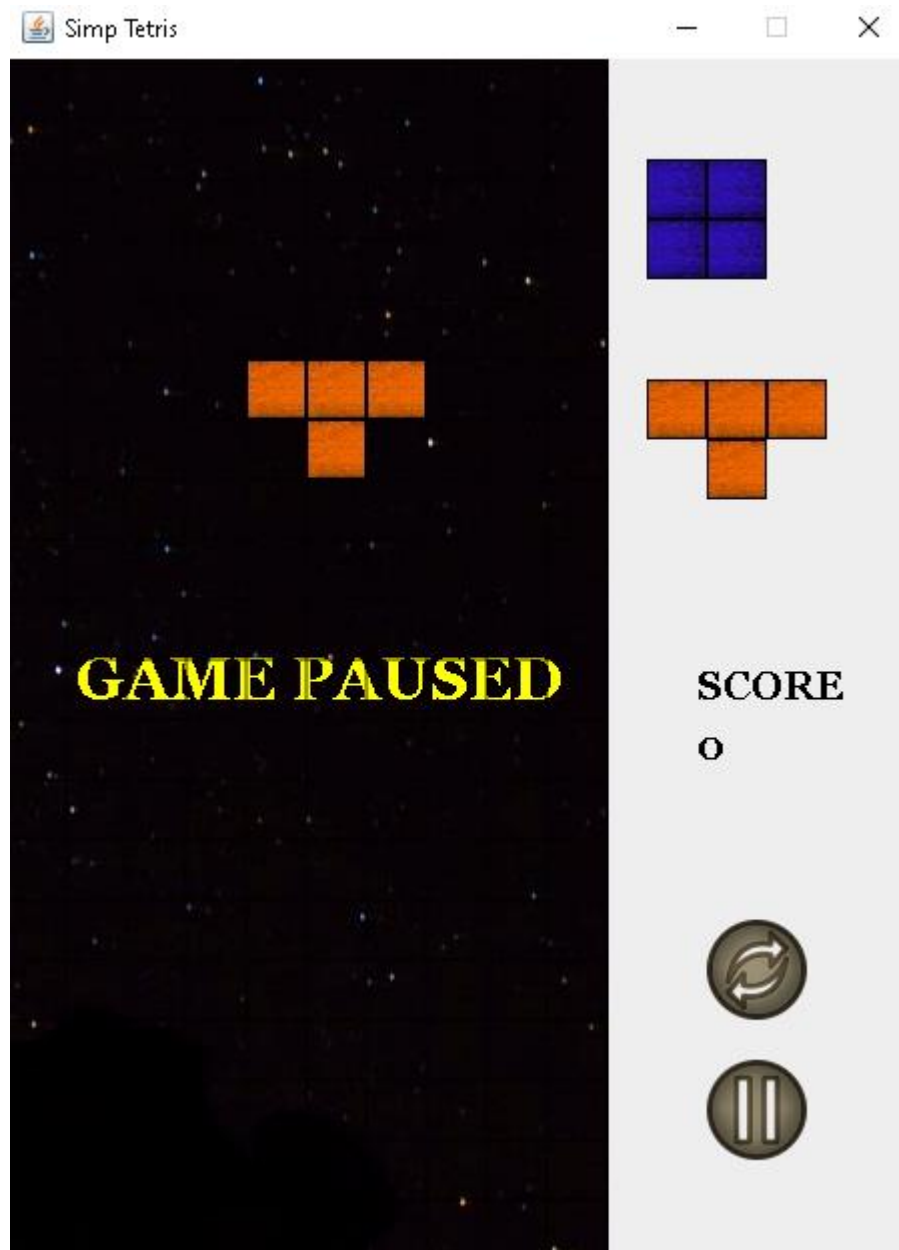


### 4.3. Giao diện chức năng

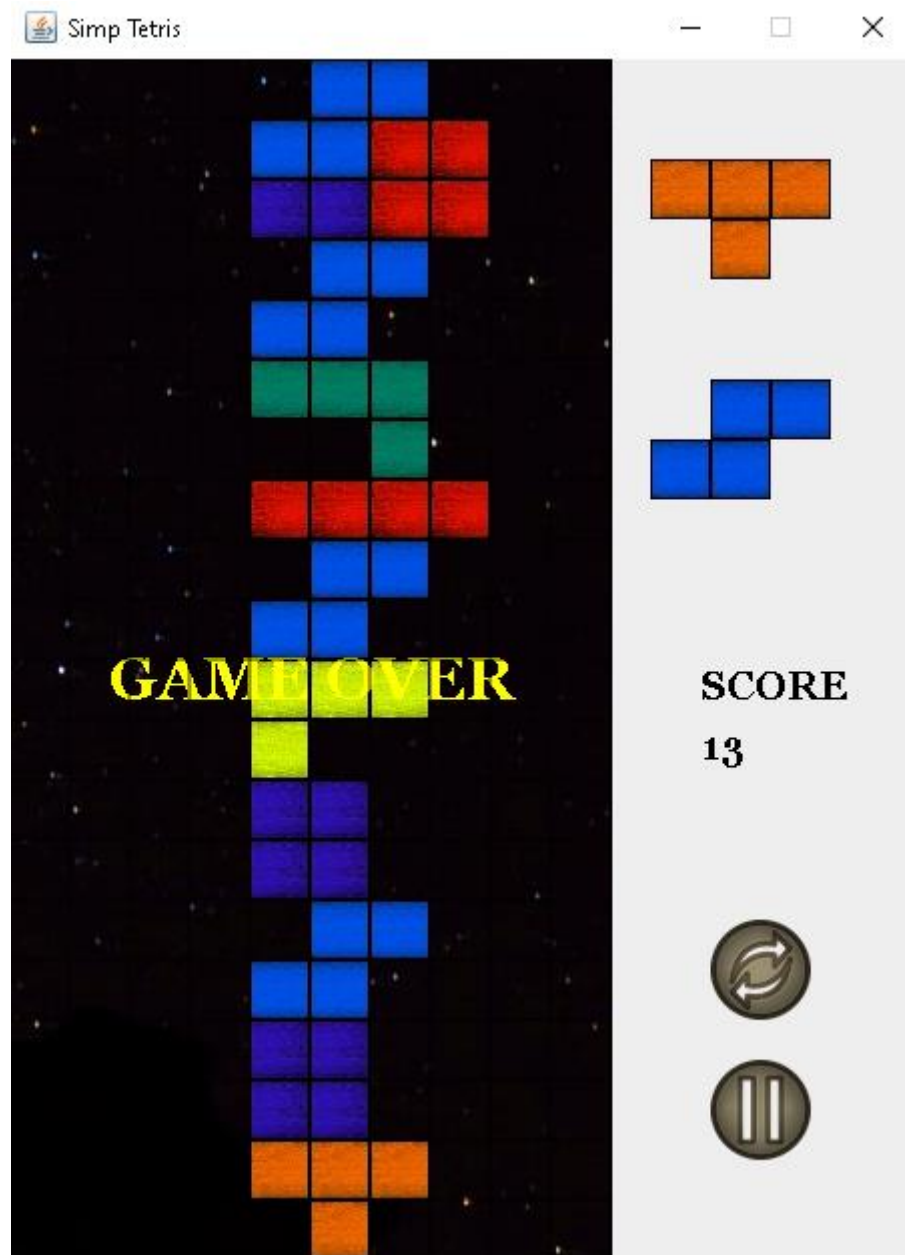
Khi click vào reset chúng ta sẽ bắt đầu lại từ đầu



4.4. Giao diện chức năng pause game:



#### 4.5. Giao diện kết thúc game



**4.6. Kiểm thử chức năng**

STT	Chức năng	Input	Output	Result
1	Play game	Click Play	Đưa người chơi bắt đầu chơi game	Đạt
2	Dịch chuyển khối	Sử dụng phím mũi trên (trái , phải , bên dưới)	Khối gạch có dịch chuyển với phím mũi tên đang sử dụng	Đạt
3	Xoay khối	Sử dụng phím mũi tên lên trên	Khối gạch sẽ xoay sau mỗi lần ấn phím mũi tên lên trên	Đạt
4	Tính điểm	Xếp một hàng gạch không có lỗ trống	Hàng gạch đó sẽ bị xoá và điểm tăng lên 1	Đạt
5	Tạm dừng game	Click vào 	Game sẽ tạm dừng khi click vào lần đầu và tiếp tục vào click tiếp theo	Đạt
6	Reset game	Click vào 	Game được reset , người chơi chơi lại từ đầu	Đạt



## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Ưu điểm và nhược điểm.

Chương trình đã hoàn thành được những yêu cầu cơ bản của bài toán.

- Ưu điểm:
  - Chạy mượt, không có lỗi logic
  - Giao diện dễ nhìn
  - Nhẹ, tốn ít tài nguyên
- Nhược điểm:
  - Một số hình ảnh trong game chưa thực sự đẹp
  - Vẫn chưa có 1 số cài đặt như âm thanh, thông tin người chơi,...

### 5.2. Hướng phát triển trong tương lai.

- Hướng phát triển cho đề tài:
  - Tìm kiếm thêm hình ảnh để giao diện game trau chuốt hơn
  - Thêm hướng dẫn chơi để người chơi dễ hiểu
  - Thêm các tính năng như âm thanh, thông tin người chơi, .

Mặc dù còn một số thiếu sót, tuy nhiên những gì thu được từ bài tập lớn lần này giúp chúng em tự tin hơn để tiếp tục đam mê trong lập trình.

Chúng em rất mong nhận được những đánh giá của thầy.

Chúng em xin chân thành cảm ơn!

## **TÀI LIỆU THAM KHẢO**

1. Java Tetris  
<https://zetcode.com/javagames/tetris/>
2. Codelearn  
<https://codelearn.io/sharing/game-tetris-lap-tinh-bang-java>